FLIR | The World's Sixth Sense®

AUTOMATION & INDUSTRIAL SAFETY TRAINING MANUAL

Contents

User guide ... 6

Getting started ... 9

How to set up your FLIR AX8 ... 10

Step 1: Download and install FLIR IP Config ... 11

Step 2: Connect your FLIR AX8 to a computer ... 13

Step 3: Configure the IP address of your FLIR AX8 ... 14

Step 4: Log in and access the web interface ... 18

How to set up your FLIR A310 ... 20

Step 1: Download and install FLIR IP Config and IR Monitor ... 21

Step 2: Connect your FLIR A310 to a computer ... 22

Step 3: Configure the IP address of your FLIR A310 ... 23

Step 4: Access your FLIR A310 in IR Monitor ... 23

How to set up your FLIR Ax5 ... 25

Step 1: Download and install FLIR GEV Demo ... 26

Step 2: Connect your FLIR Ax5 to a computer ... 28

Step 3: Access your FLIR Ax5 in FLIR GEV Demo ... 28

Additional interfaces and programs for FLIR Ax5 and other image streaming cameras .. 31

The IR image ... 32

Features walkthrough ... 33

Changing object parameters ... 35

Changing settings related to infrared images ... 36

Setting up analysis functions ... 40

Setting up an alarm ... 41

Camera control ... 41

Object parameters ... 43

Emissivity ... 44

A brief look into the concept of emissivity and temperature ... 44

Five factors that affect the emissivity ... 49

How can I determine emissivity? ... 56

How can I compensate for low emissivity? ... 59

Reflected temperature ... 63

Atmospheric temperature ... 63

Distance ... 63
Relative humidity ... 63
External IR window ... 66
Window transmissivity estimation, method 1 ... 66
Window transmissivity estimation, method 2 ... 69
Analytics and alarms ... 72
Analytics ... 72
Spot measurement ... 73
Box measurement ... 76
Delta Measurement ... 80
Isotherms ... 83
Iso-coverage ... 88
Mask ... 92
Schedule ... 95
Alarms ... 97
How do I set up an alarm? ... 98
Alarm options ... 100
Hands-on Exercises ... 106
The IR camera system ... 110
Optics ... 110
Field of view (FOV) ... 111
Spatial resolution (IFOV) ... 112
Focal length ... 116
Depth of field / Minimum focus distance ... 116
f-number ... 117
Detectors ... 120
Detector type ... 120
IR resolution ... 124
Thermal sensitivity (NETD) ... 124
Accuracy ... 125
Detector time constant ... 126
Spectral range ... 126
Image processing ... 130

Bit depth ... 130
Image streaming cameras ... 131
Smart sensor cameras ... 133
Sampling ... 135
Networks ... 140
IP address ... 141
IPv4 ... 143
Subnet mask ... 143
The length of the network ID - A simple rule ... 144
Configure a valid IP address for your IR camera. ... 144
New notation – CIDR ... 150
More on binary numbers ... 151
Gateway ... 156
A concrete example ... 156
Port ... 157
Port forwarding ... 158
Protocols ... 162
Some common protocols ... 162
Email protocols – SMTP and POP/IMAP ... 163
File Transfer Protocol (FTP) ... 164
Real Time Streaming Protocol (RTSP) ... 165
Network Time Protocol (NTP) ... 166
Hands-on exercises ... 167
Fieldbus ... 186
Modbus ... 188
EtherNet/IP ... 192
Machine Vision ... 194
GigE Vision ... 194
GenICam ... 194
Input and output ... 198
Analog signals vs. digital signals ... 199
Typical system overview ... 201
I/O cable connections ... 202

Power the FLIR AX8 with an external power supply 203
Input 204
An example using Digital in 205
Output 206
Set up an alarm in the FLIR AX8 web interface 206
Enable digital outputs with a power supply 207
Using Reversed Logic to detect camera or cable failure 208
Software 210
FLIR Tools 210
FLIR Atlas SDK 210
FLIR Products 212
Glossary 213
Index 221

User guide

I am very glad that you have found this learning material you have in front of you. I am certain of that you and I together can take us through the basics of IR cameras in automation. Before we start, I want to explain the outline of this learning material, and also give you some tips on how you can study this material in order to get the best learning outcomes.

My first tip to you is this:

Focus your studies on the parts you are most interested in or feel the most need to know.

Since both you and I are adults – we know that it is very difficult, almost impossible – to study what we are not interested in. I am not saying that I will not do my part – I will try my fullest to make the subjects of this learning material as interesting as possible. However, I am aware of the fact that all individuals are not interested in all things. This leads me to my second tip.

Don’t read every word of every chapter. Browse through the less interesting parts – or even skip them – and get back to them later. Asses your previous knowledge and focus less on the parts you already know.

Reading for the sake of reading sounds unnecessary to me. Some parts of this material you might not see the need for at first – but when you’ve gotten through other parts – might see the need for them. Maybe some parts seem too difficult at first, but hopefully they will clear as we go along through the material. My last tip to you is therefore this.

Try and think about the things you are reading – reflect on your knowledge and learning – but don’t think too long! If you get stuck – move on and return to it later. Things often become clearer after a pause from them.

Now, let me tell you about the outline of this learning material. Every chapter is presented with a set of objectives. The idea of these are partly that it should be easy for you to navigate through the material, and partly so that you may have these objectives in mind when studying this material. I've tried to pinpoint the most important take-aways from the chapters through the objectives.

Throughout the material you might stumble upon Box tips with the symbol

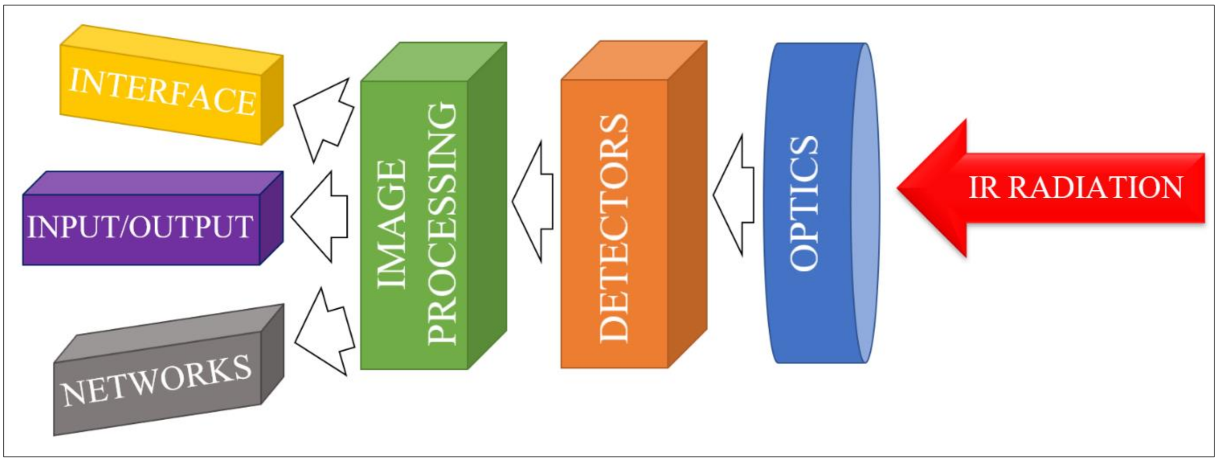
These are meant to give you tips on instructional YouTube clips, and miscellaneous tips.

You may also see these symbols

They mark the answer and the response of an SAQ. SAQ stands for Self-Assessment Question, and the meaning of them is to help you assess your learning and newly found knowledge. They are also meant to serve as an indicator of a short break, or maybe a reflective pause. The format of the SAQs is that a question will be posed in direct relation to the topic it is about. The correct answer and feedback are then given on the next page. I do urge you to give the SAQs a try before turning the page. I can promise you that your learning outcome will improve.

Another thing I recommend is for you to write and draw in this material – both where there are boxes for that purpose, but also anywhere you like. Writing while studying is shown to improve the learning, and I think that making this material your own – by writing in it – in some ways makes you in charge of your own learning. It can also be good for remembering your thoughts or measurement values in the future.

Many of the chapters in this learning material are based on hands-on activities or exercises. The best way to study through these parts is to do the activities and exercises with me. Partly because the aim of this learning material is to guide you to set up your IR camera so it matches your purposes, and partly because the saying *learning by doing* isn't all wrong.



The outline of this learning material is planned to follow the illustration above. We will walk alongside the IR radiation from the object, through the IR camera system – all the way until it meets our eyes in the form of a color. The order of the path we're walking may stray away from the order presented in the illustration – I hope that it won't get confusing. I'll show you the illustration every time I think that there is a risk for that.

My intention with this material is that the chapters should be completely stand-alone, but this may not be the case for all of them. Some chapters and exercises require you to have certain programs installed or an IR camera connected, and some contain references to other chapters in the material. Since it is a big area we are trying to cover – they will overlap – I hope that this will not pose a problem in your learning experience. After all, your learning experience is of utmost importance to me – and it is because of it this learning material exists.

I think now is as good time as any to dive into this learning experience. I am ready when you are.

Getting started

FLIR AX8

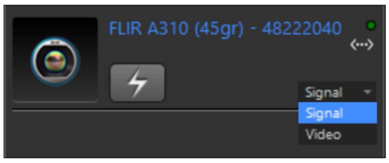
In the following chapters I will walk you through on how to set up your IR camera. I will cover the FLIR AX8, the FLIR A310 and the FLIR Ax5. I will start off with the FLIR AX8, and this chapter will be the most elaborative one – since it's the first, and many steps are similar or exactly the same for all IR cameras. The chapters on the other IR cameras will be shorter and refer to the FLIR AX8 chapter, when steps are similar or the same.

FLIR A310

There is a distinction to be made between two types of IR cameras: smart sensor camera and image streaming camera. We will delve deeper into the subject in chapter Image processing on page 130, but I'll mention a few key features here.

FLIR Ax5

Smart sensor cameras have their own built-in computer, which lets you set up analytics functions and alarms in a user interface. This also allows smart sensor cameras to operate without a connected computer, as long as they are supplied with power. Image streaming cameras need to be connected to a computer in order to operate, as they do not perform analytics and alarms by themselves. The analytics and alarms need to be programmed in a computer.



Signal or Video option for the FLIR A310 in FLIR Tools.

The FLIR AX8 is a smart sensor camera, and so is the FLIR A310 – only that the FLIR A310 can be set to stream like an image streaming camera in FLIR Tools. Setting the stream to signal in FLIR Tools will "turn" the FLIR A310 into an image streaming camera. Setting it to video will treat it as a smart sensor camera. For more on FLIR Tools, refer to chapter FLIR Tools on page 210.

The FLIR A615, FLIR A315 and FLIR Ax5 are image streaming cameras, and since their analytics and alarms need to be programmed in a software on a computer – their features will not be covered in chapter Features walkthrough. The Analytics and alarms chapter will be based on the FLIR AX8 and the FLIR A310, but the principles for analytics and alarms are the same – independent of IR camera type. For information on how to set analysis functions and alarms with image streaming cameras, refer to chapter Software – where you learn how to find software for your purposes.

Let's begin with perhaps the most important part – setting up your IR camera.

How to set up your FLIR AX8

In this first chapter I will walk you through how to set up your FLIR AX8.

Objectives

When you have worked through this part, my aim is that you will be able to answer these questions

> How do I connect my FLIR AX8 to my computer?
>
> How do I configure the IP address of my FLIR AX8?
>
> How do I access the web interface of my FLIR AX8?

If you feel that you already know the answer to these questions, feel free to just browse through – or even skip – this part. The important thing is that you know how to set up your FLIR AX8 and see the IR image in the web interface for us to be able to continue to the next chapter.

We will need to go through four steps in order to set up our FLIR AX8. They are:

> Step 1: Download and install FLIR IP Config
>
> Step 2: Connect your FLIR AX8 to a computer
>
> Step 3: Configure the IP address of your FLIR AX8
>
> Step 4: Log in and access the web interface

Box tip – If you prefer video instead of text

If you feel that it is easier with instructions on video instead of reading them in text, FLIR has made an instructional YouTube clip on how to connect your camera to your computer. You must, however, already have `FLIR IP Config` installed, so make sure that you don't miss Step 1 in this chapter.

Search for the video `FLIR AX8: Connecting to the PC` on YouTube.

Before we get started, make sure that you have the following gear:

FLIR AX8


Power over Ethernet (PoE) switch with power supply cord


Ethernet Connector M12, X-coded


Ethernet cable

Computer with an Ethernet RJ45 Connector

Internet connection to your computer

In the examples I will use a PC with Windows 10. If you are using any other type of computer or operating system, things may look a bit different.

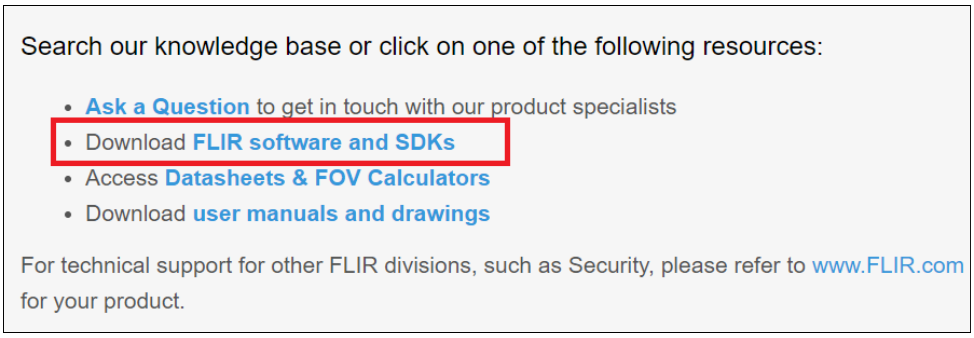
Step 1: Download and install FLIR IP Config

The FLIR AX8 uses Ethernet connection in order to communicate with the computer. This means that the IR camera has its own IP address. We have to make sure that the IP address of the camera is compatible with the computer. You can think of this as making sure that the computer and camera speak on the same phone line. We do this with the program

FLIR IP Config

First, we must find the program. **Go to** `flir.custhelp.com` and

Click on `Download FLIR software and SDKs`


Search our knowledge base or click on one of the following resources:

- **Ask a Question** to get in touch with our product specialists
- Download **FLIR software and SDKs**
- Access **Datasheets & FOV Calculators**
- Download **user manuals and drawings**

For technical support for other FLIR divisions, such as Security, please refer to www.FLIR.com for your product.

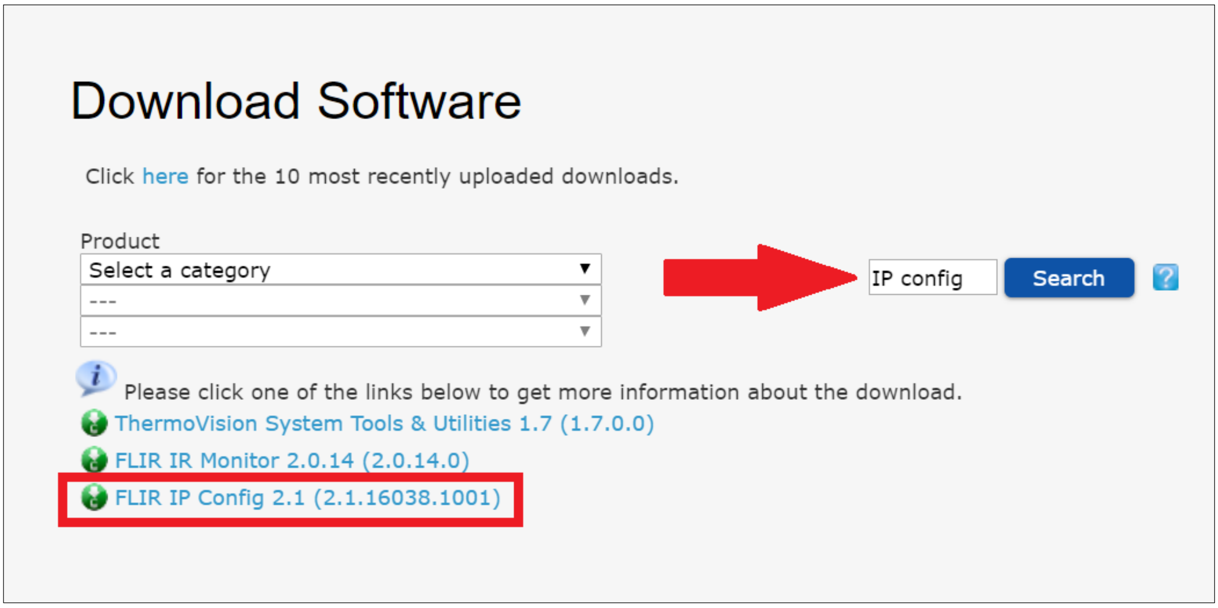
You will now be asked to log in.

If you already have an account, **log in** with your login details. Otherwise, **create** an account by clicking the Create a New Account button and follow the necessary steps.

Type `IP config` in the search field.

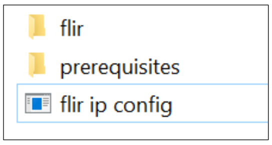
Click the file named `FLIR IP Config`.

Click on `Click to download as ZIP` to download the file as a ZIP file.

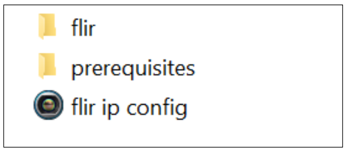


When you **open** the downloaded ZIP file, **double-click** on `flir ip config` and **select** `Extract all`. **Select** a destination for the files and **click** `Extract`.

The file content should then go from this



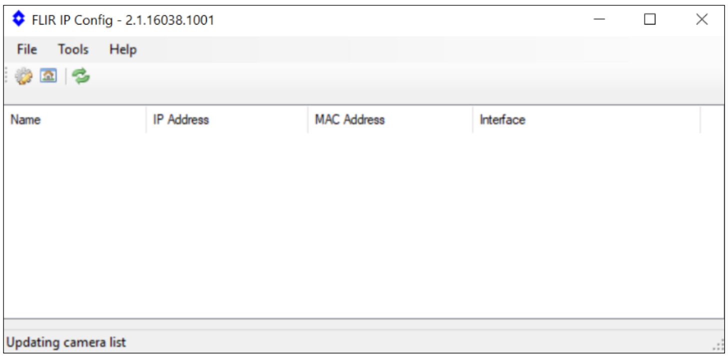
to this



Click on `flir ip config` to install the program. Your computer may have to restart after the installation.

Search for `FLIR IP Config` in your computer search menu.

Starting FLIR IP Config should look like the image to the right.



If it does, it means that you have successfully completed Step 1! Now, on to Step 2, where I'll walk you through on how to connect your FLIR AX8 to your computer, so that FLIR IP Config can detect it.

Step 2: Connect your FLIR AX8 to a computer

Now, it's time to use all the gear I mentioned earlier.

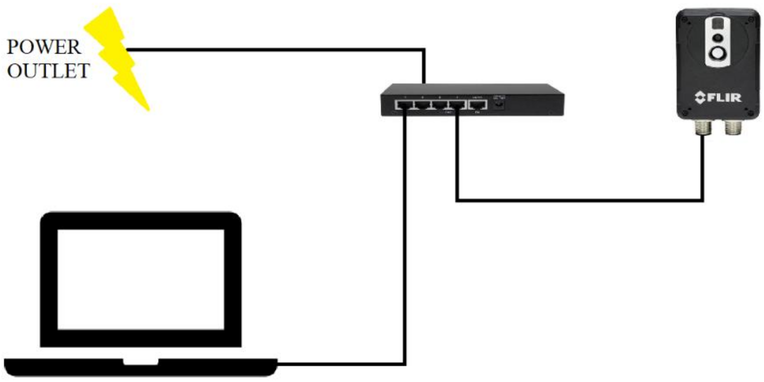
We'll start with the FLIR AX8. It has two connectors, but for now we are only going to use the left one on the FLIR AX8, which matches with your Ethernet Connector M12 cable. Don't worry, we'll get to the right connector and all the technical stuff further on (see chapter Input and output on page 198).

Connect the left connector on the FLIR AX8 with the connection on the PoE switch marked `PoE` using the Ethernet Connector M12 cable.

Connect the PoE switch to your computer using the Ethernet cable.

You can look at the diagram below to get a clearer picture on how to connect.

All you have to do now is **connect** the PoE switch to a power supply, and you have successfully completed Step 2 and connected your FLIR AX8 to your computer.



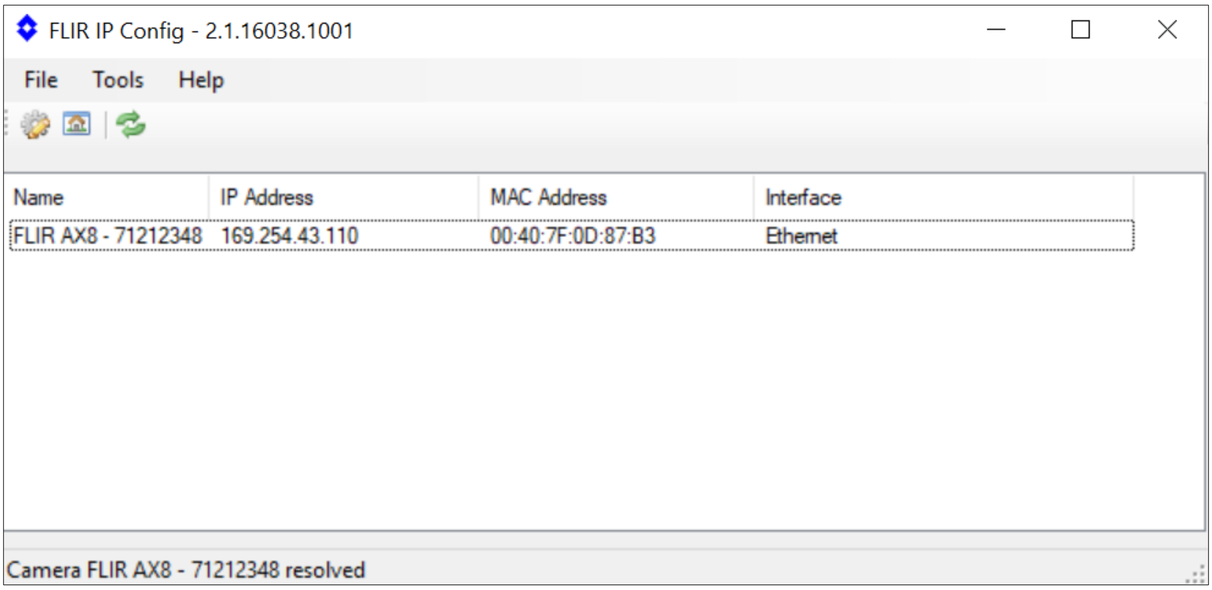
Box tip – Why is my camera making clicking sounds?

When you have connected your camera, you may hear clicking noises coming from your camera. Don't worry, your camera is not broken! This is perfectly normal, it's called the NUC (non-uniformity correction) of the camera. We will return to what it is and why it occurs (in chapter Non-Uniformity Correction (NUC) on page 123), but all you need to know right now is that there's nothing wrong with your camera.

Step 3: Configure the IP address of your FLIR AX8

Your FLIR AX8 is connected, and you've got FLIR IP Config up and running. The next step is to help your computer find your FLIR AX8.

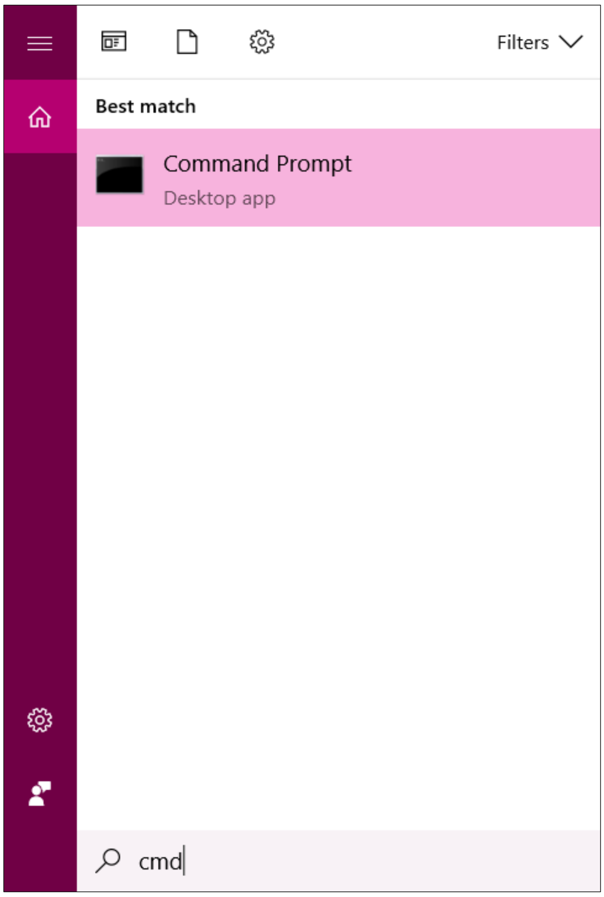
To do this, simply **click** this icon , and FLIR IP Config should be able to identify your FLIR AX8 and present it in a list.



If you have several cameras connected and wish to identify a specific one, you can match the MAC Address of your camera written on the side of the camera with the MAC address in the list in FILR IP Config.

For the computer and FLIR AX8 to properly communicate, you need to configure the IP address of your camera in FLIR IP Config. This depends on the IP address of your computer.

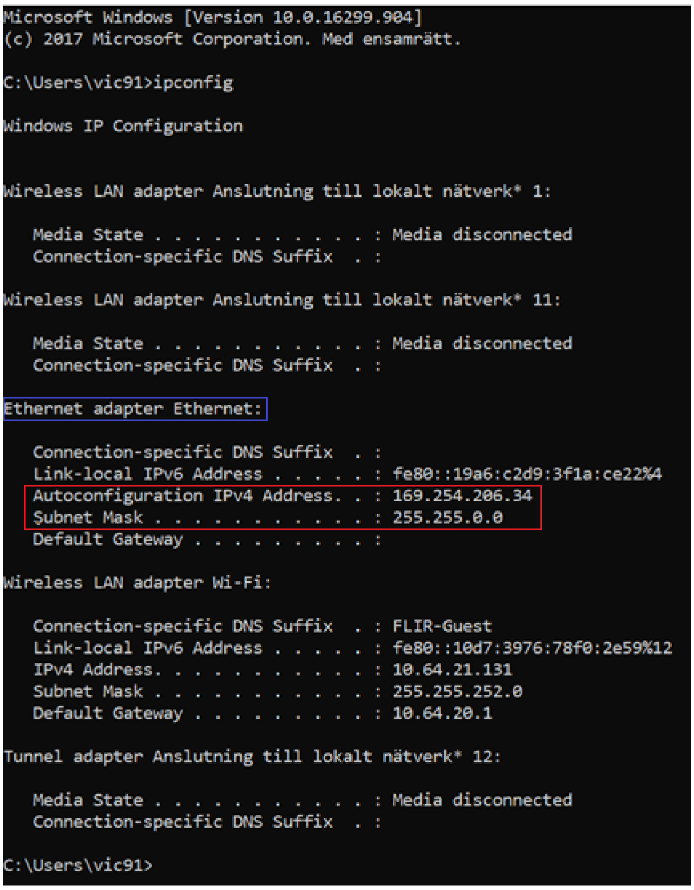
Therefore, you need to know the IPv4 address and Subnet Mask of your computer. At this time, you don't have to worry about what they are, simply follow my steps and we'll get to the details further on (see chapter Networks on page 140).



First, we are going to access the Command Prompt by **typing** `cmd` in our computer search menu and **press** `Enter`.

In the Command Prompt, **type** `ipconfig` and hit `Enter`.

We need two bits of information in the list presented to us.



First, we must **find** the text Ethernet adapter Ethernet in the list.

For me, it looks like the image to the left. I've marked what we are looking for with a blue rectangle.

In the red rectangle we find the information we need, namely Autoconfiguration IPv4 Address and Subnet Mask.

As you can see in the image, mine are 169.254.206.34 and 255.255.0.0

Since we are going to use these numbers later, I suggest that you **write** down *your* numbers.

IPv4 Address:

Subnet Mask:

Now that we have found the information needed to configure the IP addresses of our FLIR AX8, we return to FLIR IP Config.

In FLIR IP Config, **mark** your FLIR AX8 by clicking on it. Then **click** the  icon to configure the IP address of your FLIR AX8.

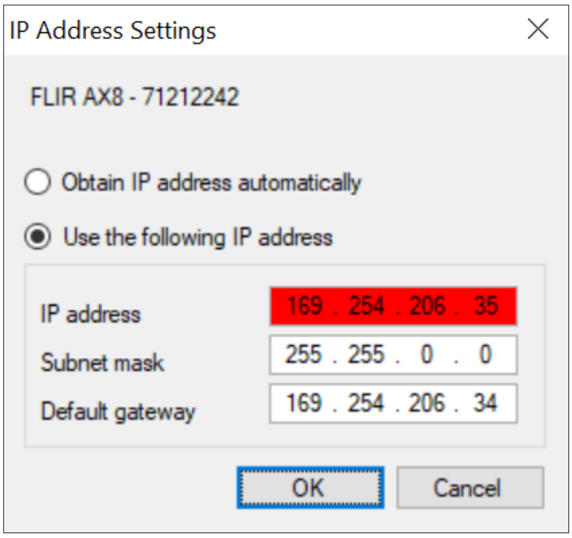
Click the `Use the following IP address` circle.

You will now need to type in IP address, Subnet mask and Default gateway. Once again, don't worry if you feel unsure about these terms. The goal right now is for you to set up your IR camera and see an infrared image. The goal is *not* that you should understand everything about IP addresses. We will cover this in the Networks chapter.

IP address

In the row for the IP address, the **first three fields** must be the **same** as your IPv4 address you wrote down earlier. The **last field** must be **different** from your IPv4 address. This is very important, since your computer and your FLIR AX8 need to speak on the same phone line – the first three fields – but not drown out each other – the last field.

This might seem tricky, but don't worry, I'll show you how I did.



My IP address of my computer was

IPv4: 169.254.206.**34**

I can therefore put the camera's

IP address to 169.254.206.**35**

Notice that the **first three** fields are the **same**, but the **last** field is **different**.

Subnet mask

To the right of Subnet mask, simply put the **same numbers** as your Subnet Mask that wrote down earlier. You can see in the image above that I put Subnet Mask: 255.255.0.0

Default gateway

In the field next to Default gateway, type in your IPv4 Address without changing any numbers. I put Default gateway: 169.254.206.34, since this was my IPv4 Address.

Now, all that's left is to **click** `OK` and then `OK` again on the pop-up message.

You have now successfully configured your camera's IP address, well done!

In the next step, you'll be able to access the web interface to actually see the IR image.

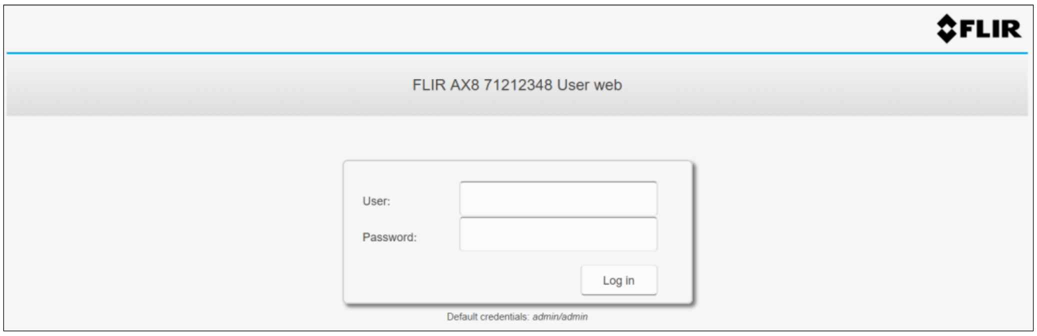
Step 4: Log in and access the web interface

We have just a couple of small steps left to get you all set up.

In FLIR IP Config, **mark** your FLIR AX8 by clicking on it. Then **click** the  icon.

This will open a new tab in your Internet browser. If FLIR IP config asks you to select a program to open the new tab with, choose your default browser.

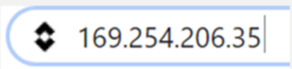
The tab opened in your Internet browser will look like the image below. Make sure you have the latest version of your web browser.



Box tip – Did the login page not show up?

If you were not directed to the login page window, don't worry.

Instead, **open** your web browser and type your **camera's IP address** in the address bar. This will lead you to the login page.



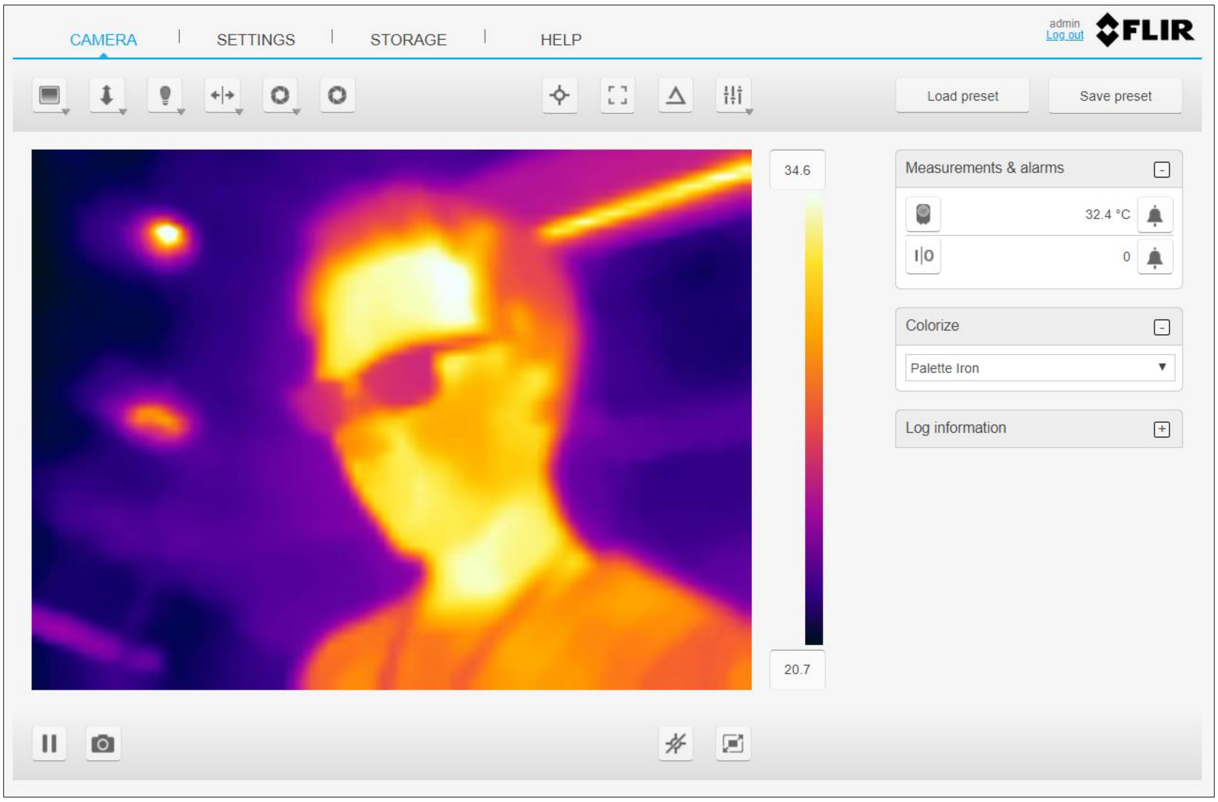
In my case, I type in 169.254.206.35, because that was the IP address which I configured in FLIR IP Config.

You can always access your FLIR AX8 by typing its IP address in the address bar, instead of going via FLIR IP Config.

Log in by **typing** `admin` as User, and `admin` as Password and **click** on `Log in`.

You should now see an interface like the image below.

If you do, well done! You have successfully configured your FLIR AX8. Now, let's move on to learn how to use the web interface!



Hey, that's me! But why are my glasses black? We'll figure that out in chapter A brief look into the concept of emissivity and temperature on page 44.

How to set up your FLIR A310

In this chapter I will walk you through how to set up your FLIR A310. Many of the steps in this chapter will be the same as in chapter How to set up your FLIR AX8 – I will therefore not repeat them – but refer you to the steps there.

Objectives

When you have worked through this part, my aim is that you will be able to answer these questions

> How do I connect my FLIR A310 to my computer?
>
> How do I configure the IP address of my FLIR A310?
>
> How do I access my FLIR A310 in IR Monitor?

If you feel that you already know the answer to these questions, feel free to just browse through – or even skip – this part. The important thing is that you know how to set up your FLIR A310 and see the IR image in IR Monitor for us to be able to continue to the next chapter.

We will need to go through four steps in order to set up our FLIR A310. They are:

> Step 1: Download and install FLIR IP Config and IR Monitor
>
> Step 2: Connect your FLIR A310 to a computer
>
> Step 3: Configure the IP address of your FLIR A310
>
> Step 4: Access your FLIR A310 in IR Monitor

Box tip – If you prefer video instead of text

If you feel that it is easier with instructions on video instead of reading them in text, FLIR has made an instructional YouTube clip on how to connect your camera to your computer.

Here's the link: `https://www.youtube.com/watch?v=-t07Nu9AWF4`

Or you can simply search for `Getting Started with the FLIR A310 Thermal Imaging Camera for Automation` on YouTube.

For a video clip on how to download the necessary software search for `Downloading Software for the FLIR A310 Thermal Imaging Camera for Automation` on YouTube.

On YouTube, or enter this web address: `https://youtu.be/k6dNzyTZRqY`

Before we get started, make sure that you have the following gear:


FLIR A310


Power supply T910922


Ethernet cable


Computer with an Ethernet RJ45 Connector


Internet connection to your computer

In the examples I will use a PC with Windows 10. If you are using any other type of computer or operating system, things may look a bit different.

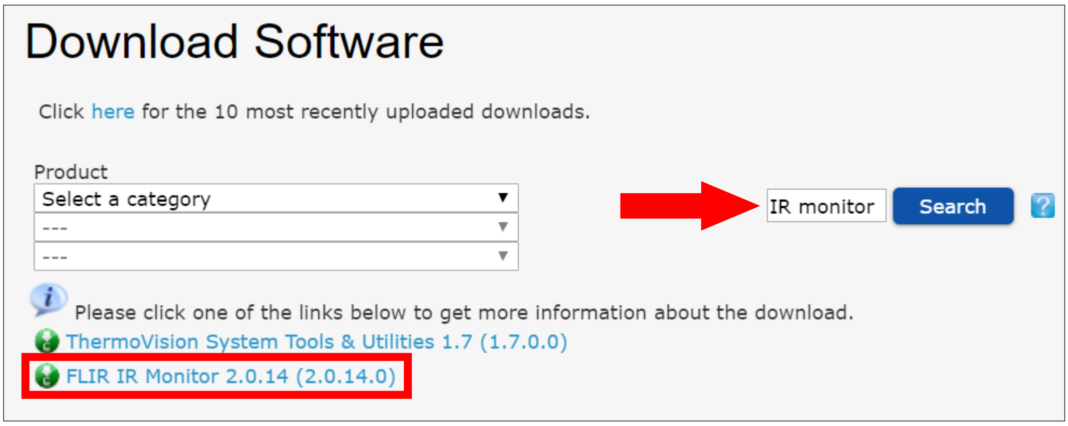
Step 1: Download and install FLIR IP Config and IR Monitor

The step to download FLIR IP Config is presented in chapter Step 1: Download and install FLIR IP Config – so please go through pages 11-13 – before continuing.

IR Monitor is found on the same place as FLIR IP Config, through the address

`http://flir.custhelp.com/app/account/fl_download_software`

Only this time you **search** for `IR monitor` in the search field.



When you've **downloaded** IR Monitor – go through the same steps as with FLIR IP Config (page 12) – extract the files and install the program.

Step 2: Connect your FLIR A310 to a computer

Connect the power supply T910922 to your FLIR A310, by using the connector marked *12/24 VDC*. **Connect** your FLIR A310 with your computer using the Ethernet cable in the connector marked *10/100*.

It is also possible to connect your FLIR A310 with a PoE switch.

Box tip – Why is my camera making clicking sounds?

When you have connected your camera, you may hear clicking noises coming from your camera. Don't worry, your camera is not broken! This is perfectly normal, it's called the NUC (non-uniformity correction) of the camera. We will return to what it is and why it occurs (in chapter Non-Uniformity Correction (NUC) on page 123), but all you need to know right now is that there's nothing wrong with your camera.

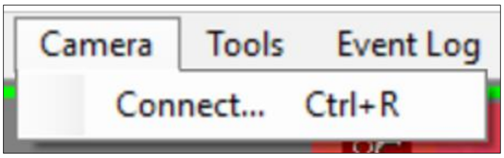
Step 3: Configure the IP address of your FLIR A310

This step is exactly the same as Step 3 on pages 13-18 – so I refer you to those pages instead of repeating myself.

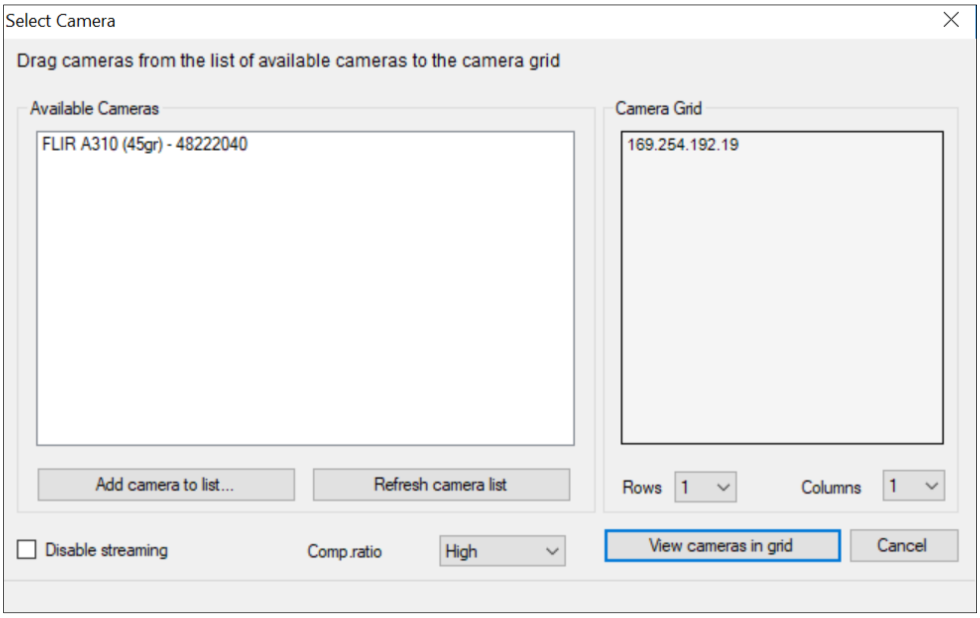
Step 4: Access your FLIR A310 in IR Monitor

Start IR Monitor.

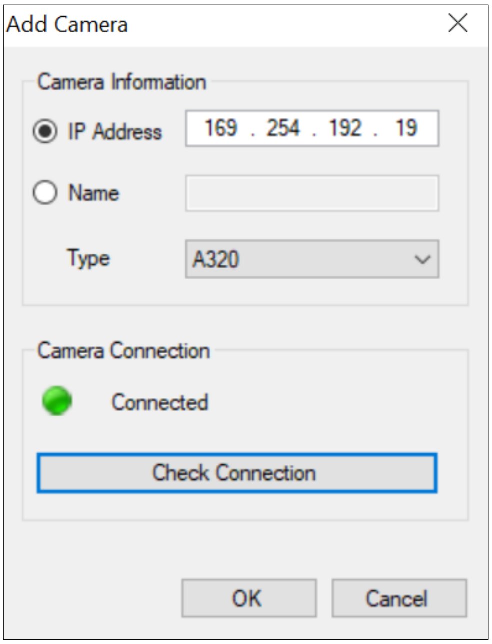
If IR Monitor doesn't identify your FLIR A310 directly you can find it by **clicking** `Camera >> Connect` in the main tab.



If your FLIR A310 shows up under *Available Cameras*, **drag** your camera to the box under *Camera Grid*. Then **click** `View cameras in grid`. (To view several cameras, you have to adjust the number of rows or columns before dragging your cameras to the Camera Grid).



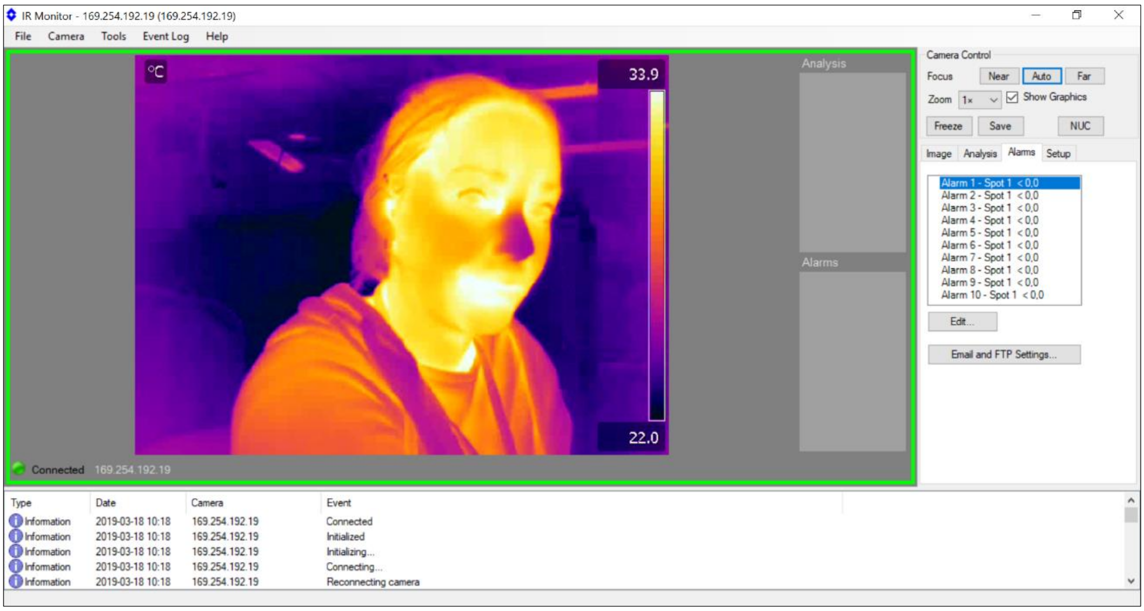
If your FLIR A310 does not show up – **click** `Add camera to list…` .



Fill in the IP Address of your FLIR A310 – the one you gave it in FLIR IP Config – and **click** `Check Connection`.

If you've filled in the correct IP Address, the Camera Connection light should be green, and you can **click** `OK`.

Lastly, **click** `View cameras in grid` and you should see the IR image of your FLIR A310.



How to set up your FLIR Ax5

In this first chapter I will walk you through how to set up your FLIR Ax5.

Objectives

When you have worked through this part, my aim is that you will be able to answer these questions

> How do I connect my FLIR Ax5 to my computer?
>
> How do I use FLIR GEV Demo?
>
> What additional programs and software are there for the FLIR Ax5?

If you feel that you already know the answer to these questions, feel free to just browse through – or even skip – this part. The important thing is that you know how to set up your FLIR Ax5 and see the IR image in FLIR GEV Demo for us to be able to continue to the next chapter.

We will need to go through three steps in order to set up our FLIR Ax5. They are:

> Step 1: Download and install FLIR GEV Demo
> Step 2: Connect your FLIR Ax5 to a computer
> Step 3: Access your FLIR Ax5 in FLIR GEV Demo

Lastly, I will briefly discuss additional interfaces and programs for image streaming cameras such as the FLIR Ax5.

Before we get started, make sure that you have the following gear:

FLIR Ax5

Power over Ethernet (PoE) switch with power supply cord

2 Ethernet cables

Computer with an Ethernet RJ45 Connector

Internet connection to your computer

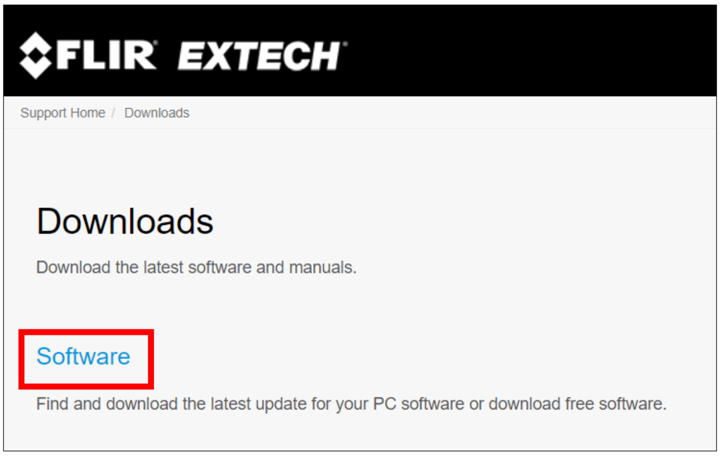
In the examples I will use a PC with Windows 10. If you are using any other type of computer or operating system, things may look a bit different.

Step 1: Download and install FLIR GEV Demo

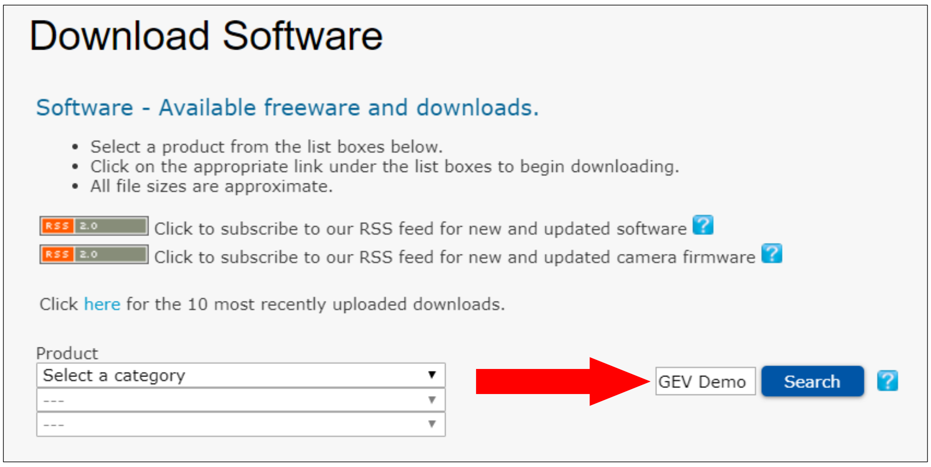
To find the program, type the following link in your web browser.

`https://flir.custhelp.com/app/account/fl_downloads`

You should then find your way to this page.

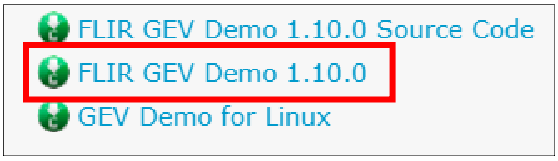


Click on `Software`.



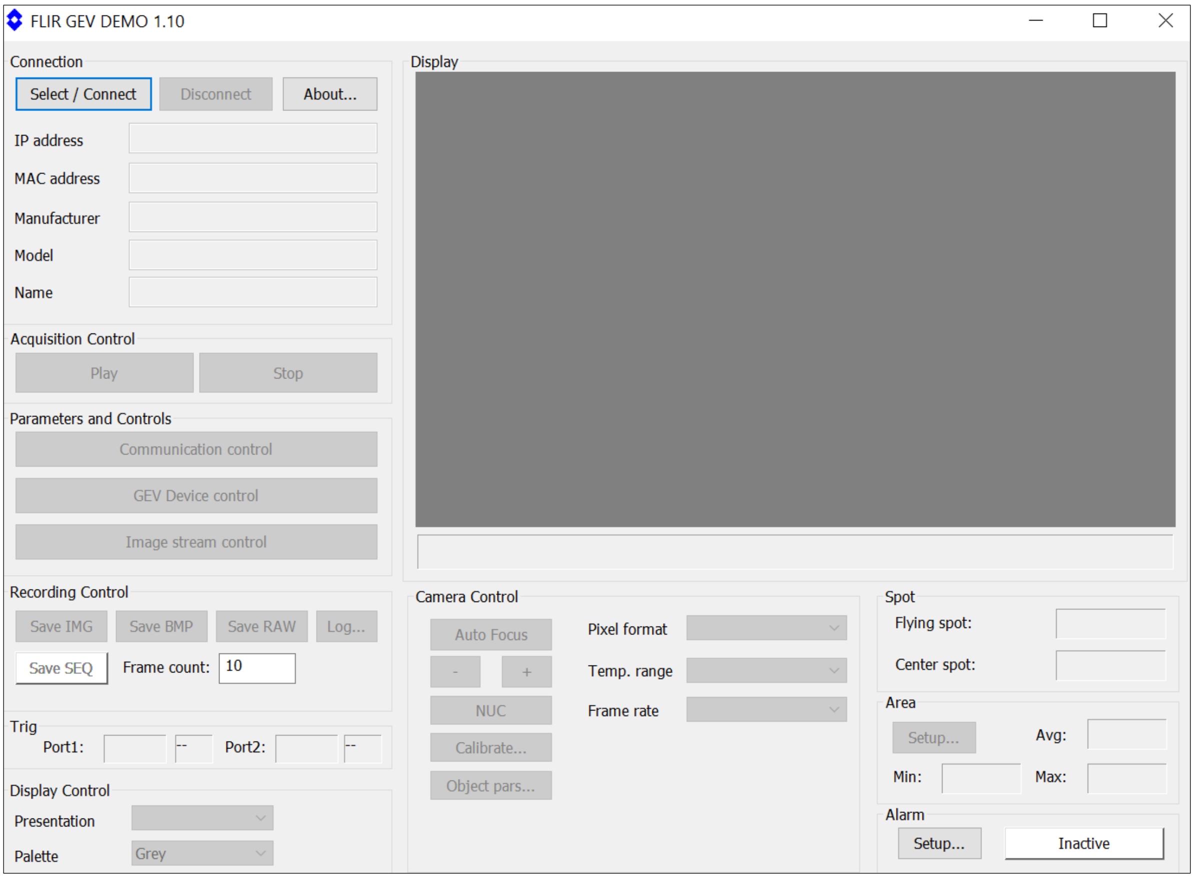
Enter `GEV Demo` in the search field.

Click `FLIR GEV Demo 1.10.0`, or the latest version available (or `GEV Demo for Linux`, if this is the system you're using).



Follow the same procedure as on page 12 for downloading FLIR IP config: **Download as zip**, **extract** and **install**.

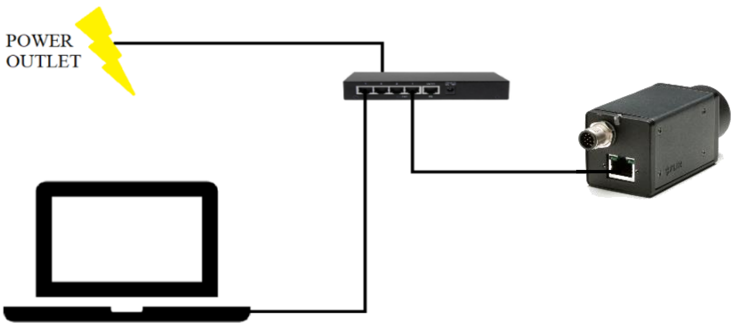
Starting the FLIR GEV Demo program should look like this.



If it does, it means that you have successfully completed Step 1! Now, on to Step 2, where I'll walk you through on how to connect your FLIR Ax5 to your computer, so that you can see the IR image.

Step 2: Connect your FLIR Ax5 to a computer

Connect your FLIR Ax5 with the PoE switch using an Ethernet cable. Make sure that you connect your FLIR Ax5 with the outlet on the PoE switch marked `PoE`. **Connect** the computer with the PoE switch using an Ethernet cable. You can look at the circuit diagram below to get a clearer picture of how to connect.



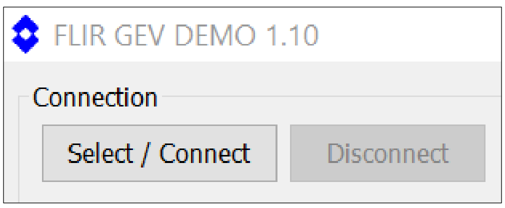
Now, all that's left is to **connect** the PoE switch with a power outlet – and you've completed Step 2.

> **Box tip – Why is my camera making clicking sounds?**
>
> When you have connected your camera, you may hear clicking noises coming from your camera. Don't worry, your camera is not broken! This is perfectly normal, it's called the NUC (non-uniformity correction) of the camera. We will return to what it is and why it occurs (in chapter Non-Uniformity Correction (NUC) on page 123), but all you need to know right now is that there's nothing wrong with your camera.

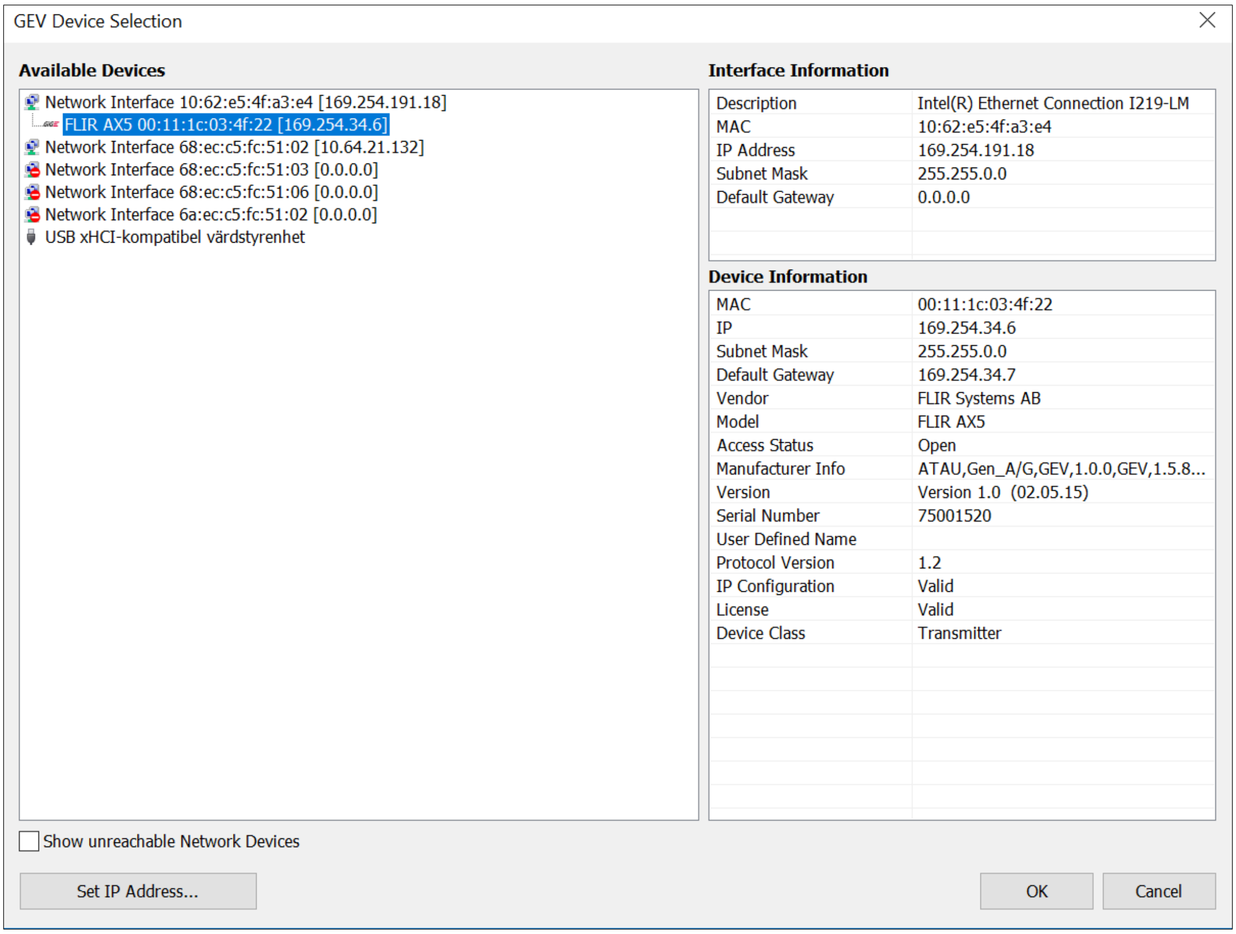
Step 3: Access your FLIR Ax5 in FLIR GEV Demo

Back to FLIR GEV Demo!

Under `Connection`, **click** `Select / Connect`.

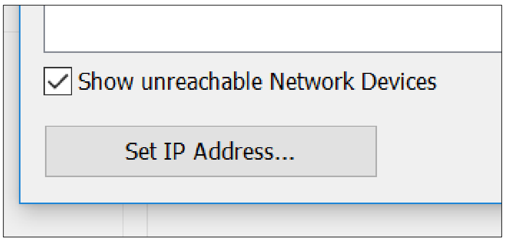


This will open a dialog box looking like this.



If your FLIR Ax5 is shown in the list under `Available Devices`, **mark** it and **click** `OK`.

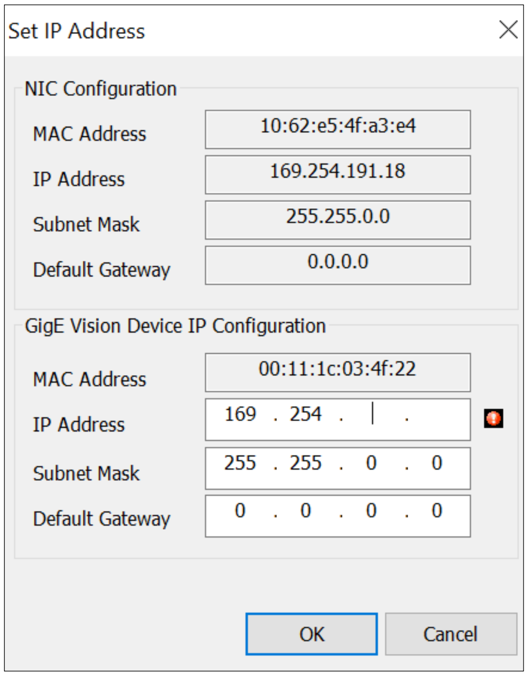
If your FLIR Ax5 does not show up in the list, **tick** the box `Show unreachable Network Devices`.



Your FLIR Ax5 should then show up in the list. You now have to set the IP address of your FLIR Ax5 manually, so that it works with your computer. **Mark** your FLIR Ax5 in the list and

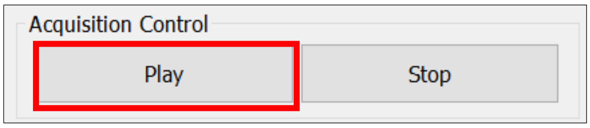
click `Set IP Address… .`

This will open a dialog box where you can manually set the IP address of your FLIR Ax5. The procedure of choosing the IP address, subnet mask and default gateway are thoroughly described on page 17. The IP address and subnet mask of your computer are displayed under `NIC Configuration`. This means that you do not have to retrieve this information via the Command Prompt.

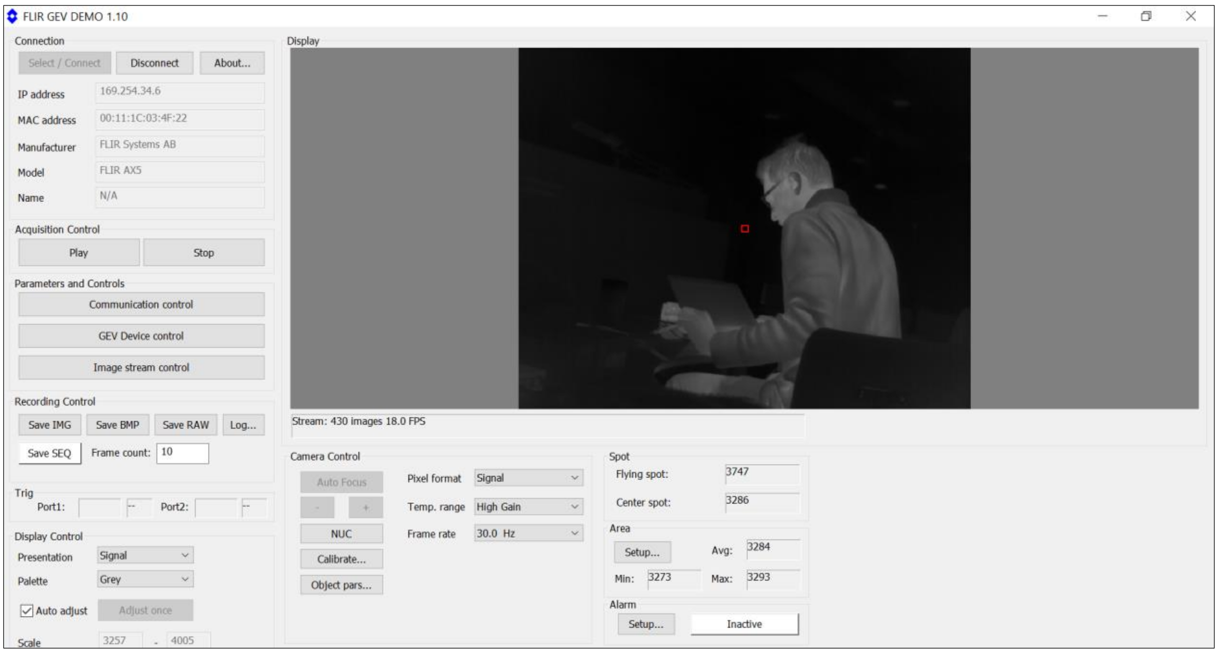


Click OK so that you are returned to the main panel of FLIR GEV Demo.

Under Acquisition Control, click Play.



You should now be able to see the IR image. Well done!



Additional interfaces and programs for FLIR Ax5 and other image streaming cameras

The FLIR Ax5 is an image streaming camera and not a smart sensor camera – that is, the FLIR Ax5 can stream radiometric images and videos (more on radiometric on page 133), but all the analytics and alarms are performed on the computer and not in the camera itself. I will cover this more thoroughly later on. The consequence of the image streaming camera not having analytics and alarm functions programmed in the camera is that you need to program these by yourself in the computer. There are some analytic functions in the FLIR GEV Demo, as you might have seen. These do not, however, use the full potential of the FLIR Ax5. To get the full potential, I would advise you to program the analytic functions by yourself. Fortunately, FLIR has developed Software Development Kits (SDKs) and additional programs you can download from their web site. For information on where you can find the additional programs, refer to chapter Software on page 210.

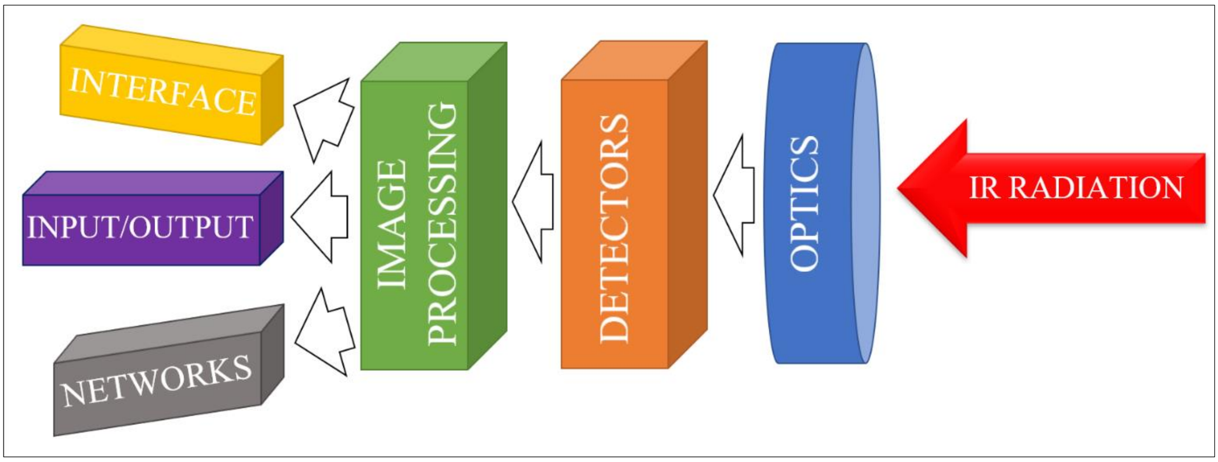
The consequence for the owner of an image streaming camera when it comes to this material is that I will not include images from the user interface of these cameras. This does not mean that the rest of the material is useless for you – the features and main principles are the same, independent of type of IR camera. It simply means that you – owner of an image streaming camera – have to see the connection between your user interface and the ones I will discuss.

The IR image

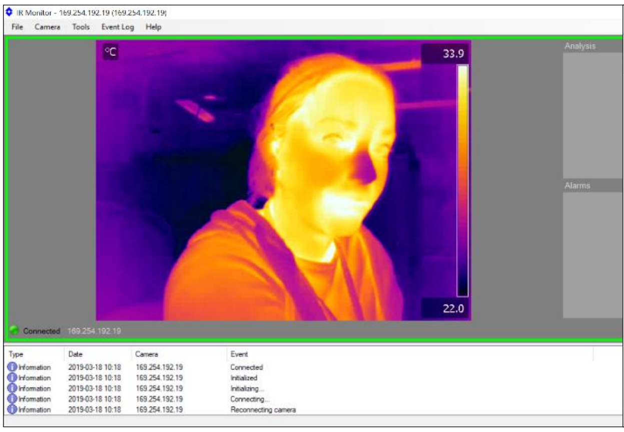
Now that we've gotten so far as to being able to see the IR image in an interface, we might wonder: what does the IR image depict? The IR image is composed of **pseudo colors** – or false colors – correlated to the **surface temperatures** of the scene. By pseudo colors I mean that the colors in the IR image is not the same as the colors we would get with a normal visual camera. Whatever your goal is with your IR camera, I would suspect that it has something to do with measuring temperatures. Before we go on to discover the different features and parameter settings of the IR camera, I would like to say a few words about the colors and the temperatures portrayed in the IR image. As you'll discover further on, the color scheme of the IR image can be changed by you – but the **color difference** between objects in the scene is *relative*. By that I mean that the color one object assumes depends on the objects surrounding it. White doesn't always mean that the object is hot or has a certain temperature. It simply means that the object whose color is white has a higher temperature than surrounding objects. The object may well have a temperature of -40 °C and be depicted as white, if the surroundings are colder.

The last thing I want to say before we move on is that the IR camera can only detect surface temperatures. That is – you cannot use the IR camera to see temperatures inside an object, only its surface.

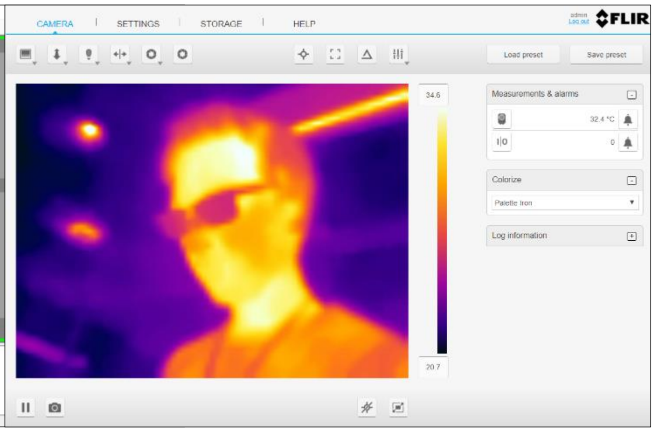
Features walkthrough



In this chapter, my aim is that you become more familiar with the different features in the IR camera interface. That is, what happens to the IR radiation after it has passed through the IR camera, but before it reaches your eyes. I will discuss where you can find different settings and refer you to the chapter that cover that specific setting. This chapter is thus aimed at very briefly guide you through the features, and letting you use it as a reference chapter. The features will be discussed from the interfaces of IR Monitor – used with the FLIR A310, and the web interface – used with the FLIR AX8.



IR Monitor



The web interface

Objectives

When you have worked through this part, my aim is that you will

- Know where to find the information you need in this material
- Feel comfortable in navigating the user interface
- Recognize symbols and concepts of the user interface

I've separated this chapter into five parts. They are:

- Changing object parameters
- Changing settings related to infrared images
- Setting up analysis functions
- Setting up an alarm
- Camera control

Box tip – Video tutorials

There are several good video tutorials on YouTube for both FLIR A310 (IR Monitor) and FLIR AX8 (the web interface)

Here are the links:

Setting up your FLIR A310 Thermal Imaging Camera using FLIR IR Monitor
`https://www.youtube.com/watch?v=G5IVOuV45Ig`

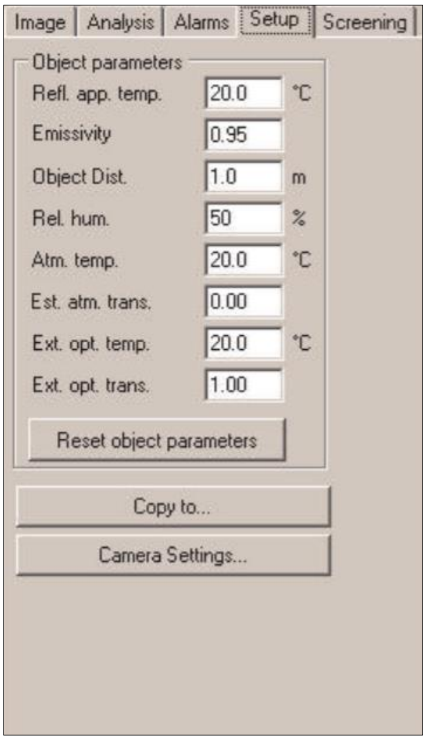
IR Monitor Tutorial - Analysis & Alarms
`https://www.youtube.com/watch?v=fC0UY3CX3LY`

IR Monitor - Camera Setup
`https://www.youtube.com/watch?v=0XSNW15Fgvo`

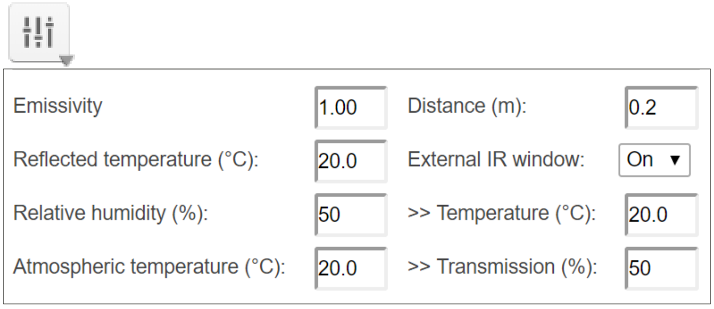
FLIR AX8: Features Walkthrough
`https://www.youtube.com/watch?v=gA-GXixDYrs`

FLIR AX8: Measurement Tools
`https://www.youtube.com/watch?v=ydyZKJpX20k`

Changing object parameters



FLIR IR Monitor



FLIR web Interface

The object parameters determine how your IR camera interprets the incoming IR radiation. They are important to set correctly, to get an accurate temperature measurement. I would therefore advice you to study them more thoroughly in the Object parameters chapter on page 43. The object parameters may go under different names, so I will try to clarify which is which in the table below.

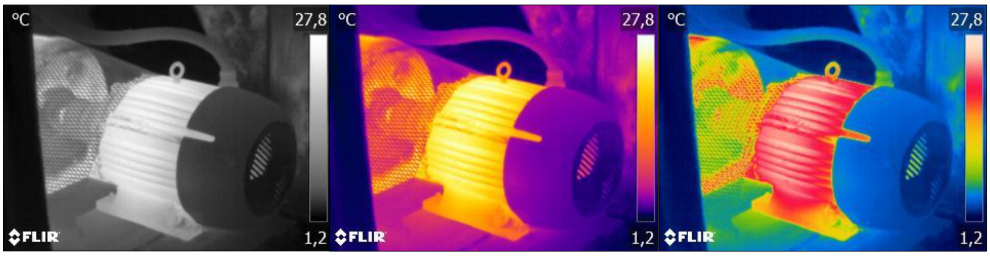
IR Monitor (FLIR A310)	Web interface (FLIR AX8)	Description and page reference
Refl. app. temp.	Reflected temperature	The temperature of the reflected IR radiation (page 63)
Emissivity	Emissivity	A measurement of how much of the IR radiation that originates from the object (page 44)
Object Dist.	Distance	The distance between your target and your IR camera (page 63)
Rel. hum.	Relative humidity	The relative humidity of the scene. Default is usually 50 % (page 63)
Atm. temp.	Atmospheric temperature	The temperature of the atmosphere between your target and your IR camera (page 63)
Est. atm. trans.	-	The estimated atmospheric transmittance. Simply put, an estimation of the atmosphere's ability to transmit IR

		radiation. It is not set by the user but calculated through other parameter settings.
Ext. opt. temp.	External IR window: on. Temperature	The temperature of the external IR window (page 64)
Ext. opt. trans.	External IR window: on. Transmissivity	The transmissivity of the external IR window (page 64)

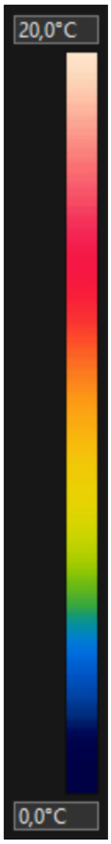
Some analysis functions allow you to set **local object parameters**. These settings will override the global object parameters within the analysis function, and the global object parameters will apply elsewhere.

Changing settings related to infrared images

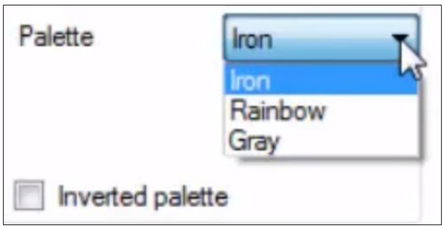
The interfaces of the FLIR AX8 and FLIR A310 differs in some aspects when it comes to the settings related to infrared images. Both have the options to change the palette used, the span and whether to have the overlay graphics visible or not. The palette is the color scheme of the IR image – it determines the colorization of the IR image.



Different palettes. From left: grey, iron, rainbow

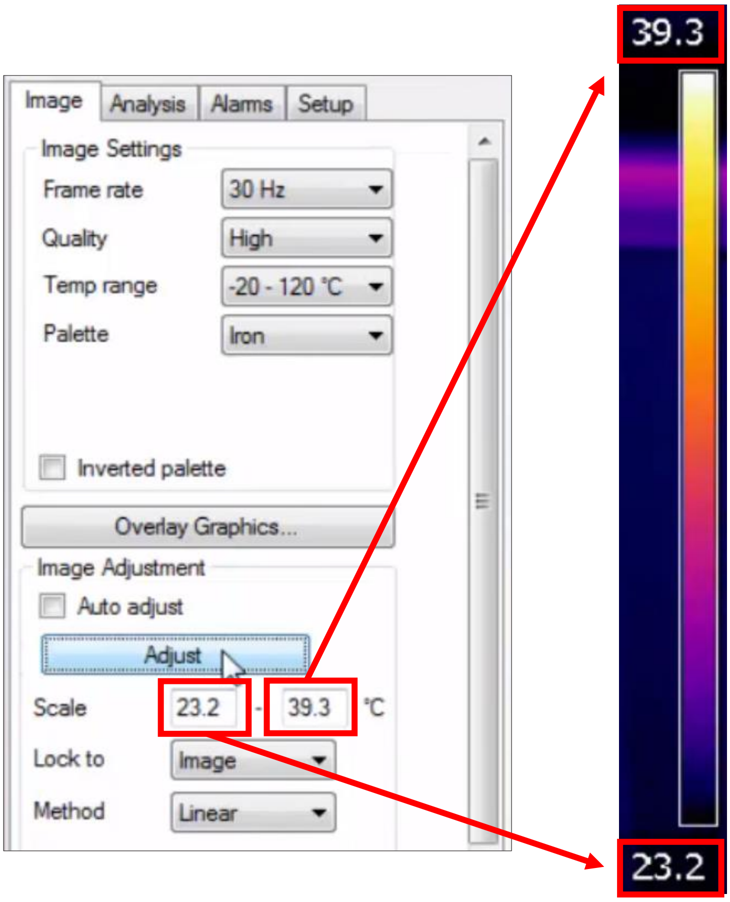


The span of the image is simply the temperature span in the image. You can see it on the color bar in the interface. In the color bar to the right, the span is 0 °C – 20 °C. The level denotes the temperature in the middle of the span. In the color bar to the right, the level is 10 °C.



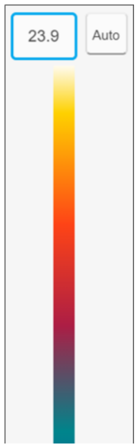
The palette option of IR monitor

The palette option of the web interface

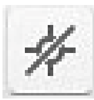


IR Monitor allows you to adjust the span or scale of the IR image. IR Monitor also allows you to lock the temperature scale to the image or to the temperature. When ticking the "Auto adjust" box, the scale will be adjusted to match the temperatures of the scene.

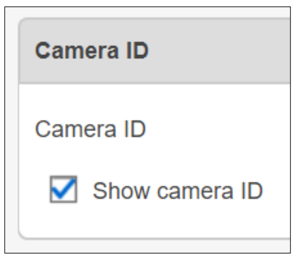
To adjust the scale in the web interface, you simply set the temperature limits in the blue squares, as shown in the image to the right.

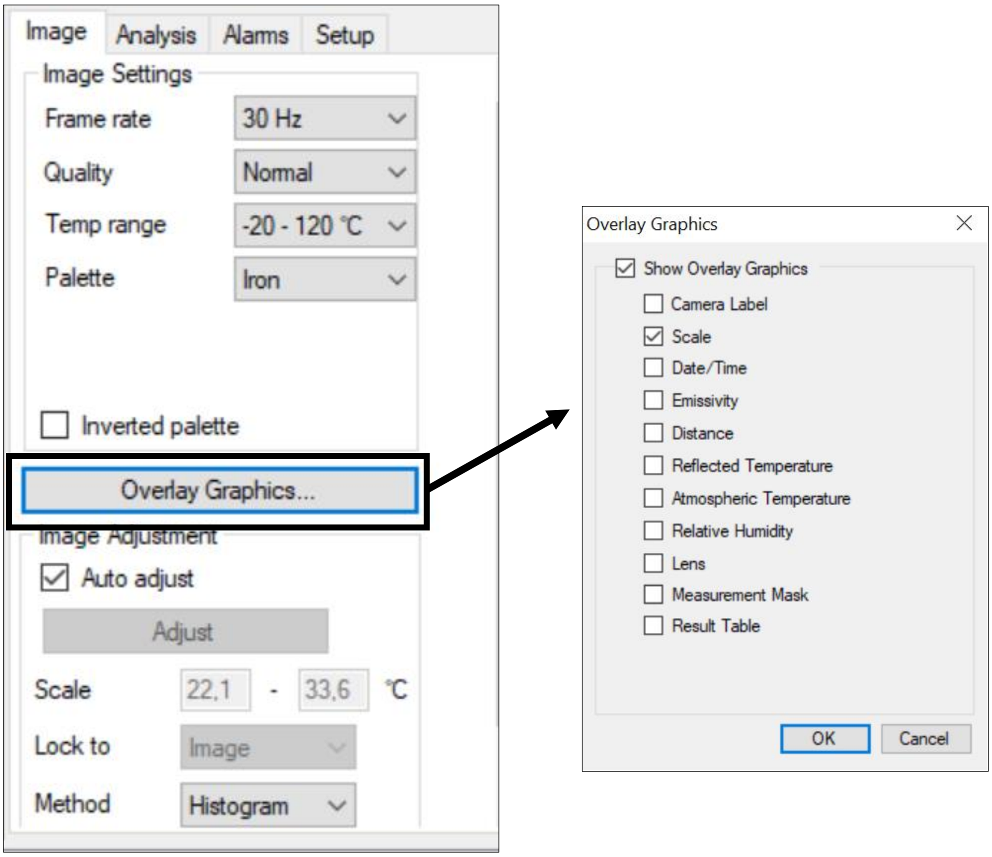


The overlay graphics are basically all the text information about the IR image. This can be the spots, areas and input settings, such as object parameters.

In the web interface, you can click the symbol `Hide overlay`  to make all analysis functions hidden in the live stream.

If you want your camera ID to be visible in snapshots or video clips, you can tick the box `Show camera ID` via `Settings >> Camera ID`.

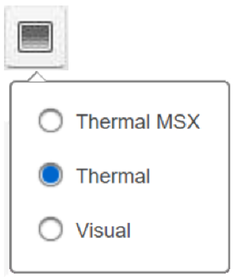




IR Monitor lets you choose which overlay graphics you want to be visible.

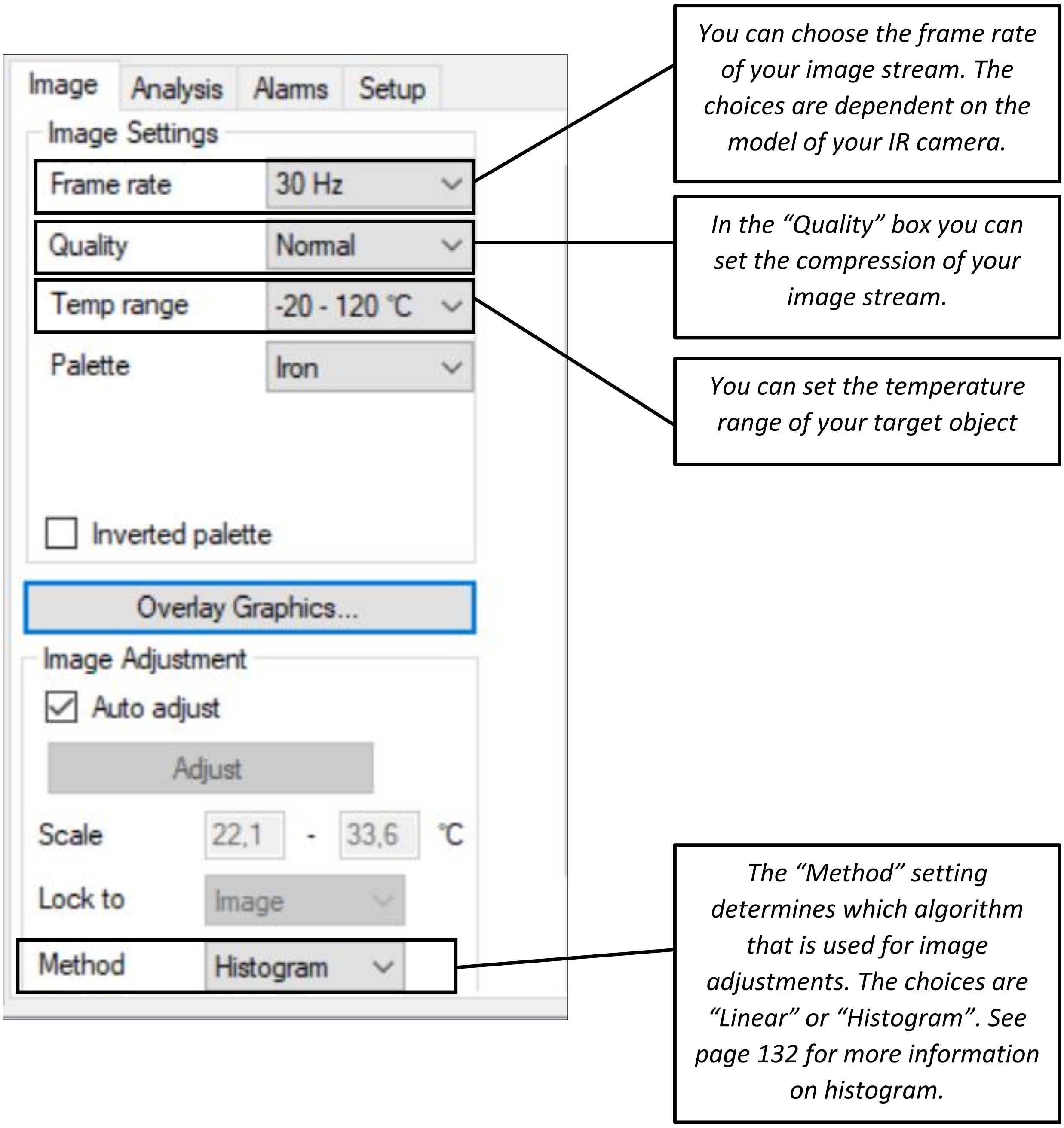
Other features of the web interface are that you can flip the image vertically and horizontally, 

and you can choose whether you want a thermal view, visual view, or a mix of both (Thermal MSX).



 When using the Thermal MSX – be sure to set the distance to your target with the symbol to the right. This is so that the thermal and visual image are aligned.

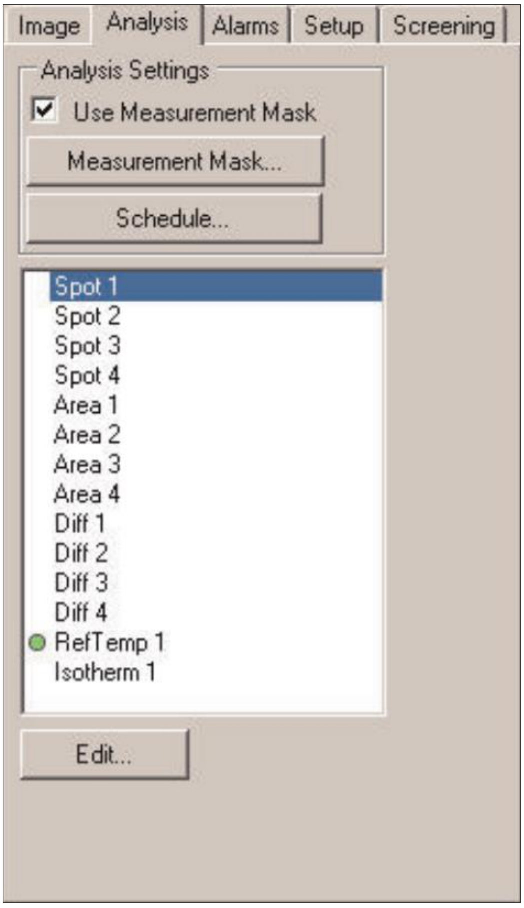
Additional features of IR Monitor are displayed in the image below.



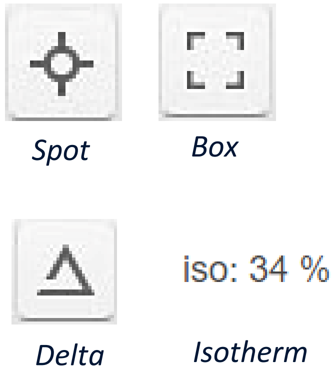
Setting up analysis functions

There are essentially four different analysis functions in the web interface, and they are: Spot, Box, Delta/Diff and Isotherm. IR Monitor has these four, and an additional type: Mask.

In the Analytics chapter on page 72, I will discuss these thoroughly. Here, I only present the interfaces and where to find the analysis functions.



Analysis functions in IR Monitor



Analysis functions in the web interface

IR Monitor also lets you set up a schedule for when image and video captions is to be sent to an email or FTP server (see page 95).

Setting up an alarm

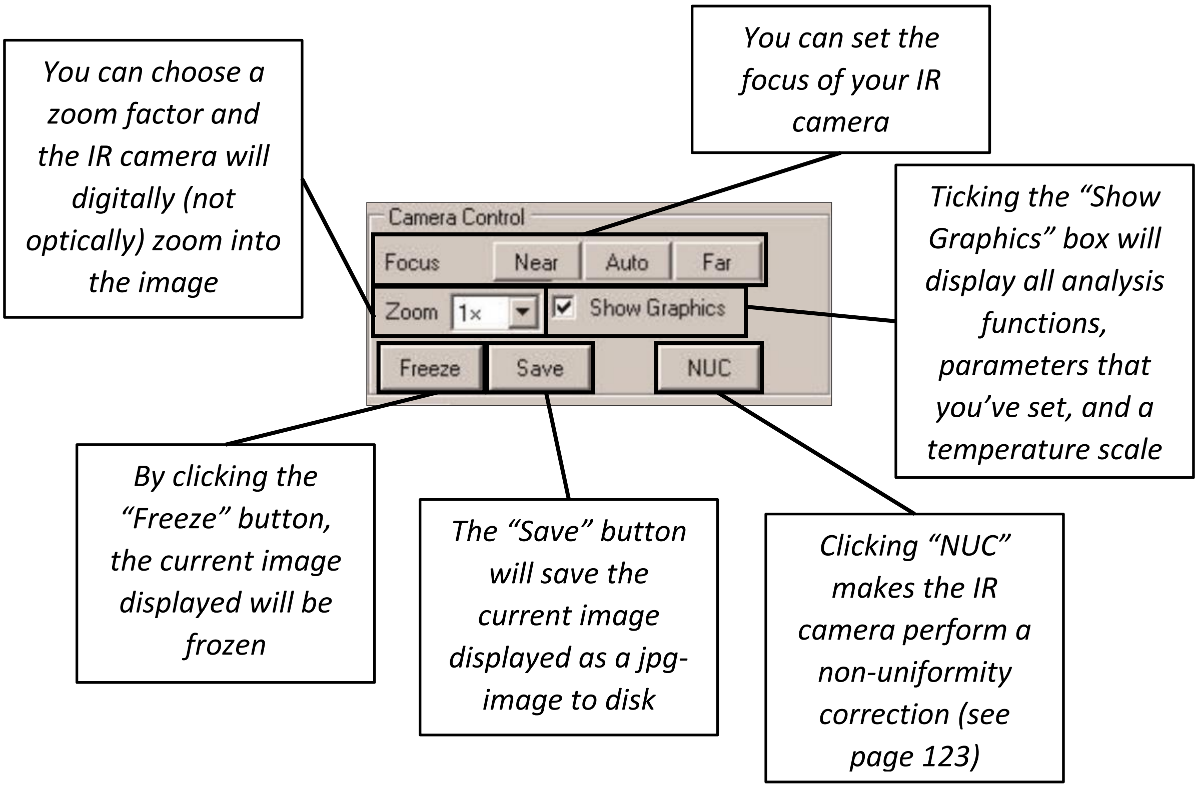
Alarms can be associated with all the analysis functions. This requires that you set conditions to be met for the alarm to go off. The conditions can be above, below or between set temperatures, or a percentage covered with an isotherm. I will go through this in more detail in the Alarms chapter on page 97.

There are different settings for the alarm action:

- A digital output, covered in the Output chapter on page 206
- An image or video capture sent to an email or FTP server, covered in chapter Protocols on page 162
- A beep, flash or disable NUC (page 123 on NUC, and page 103 on alarm actions)

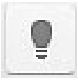
Camera control

The camera control functions for IR Monitor are displayed in the image below.

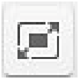


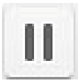
In the web interface, the camera control function symbols are the following.

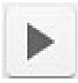
 Perform and set the NUC (see page 123)

 Turn on and off the light on the IR camera

 Take a snapshot of the current image displayed and saves it as a jpg-image in the `Storage` tab

 View the image in full screen mode

 Freeze the current image displayed

 Resume the live image stream

This chapter was meant to give you an overall view of the features in the web interface and IR Monitor and where you can learn more about them. One important feature is the object parameters as we will dive into next.

Object parameters

Your IR camera can make many adjustments and calculations, to provide you with accurate surface temperature measurements. The IR camera cannot, however, do everything by itself. It is you – the user – who need to adjust a few parameters regarding your object's surroundings so that the IR camera can measure and calculate the surface temperature. You do this by setting the Object Parameters.

Emissivity	1.00	Distance (m):	0.2
Reflected temperature (°C):	20.0	External IR window:	On ▼
Relative humidity (%):	50	>> Temperature (°C):	20.0
Atmospheric temperature (°C):	20.0	>> Transmission (%):	50

Objectives

When you have worked through this part, my aim is that you will be able to

- Explain the importance of the concept emissivity from an IR imaging perspective
- Identify factors that affect emissivity
- Set the object parameter settings accurately

For you to be able to fulfill the objectives, I will break down the emissivity chapter into four parts. They are:

- A brief look into the concept of emissivity and temperature
- Five factors that affect the emissivity
- How can I determine emissivity?
- How can I compensate for low emissivity?

I will just say a few words about Reflected temperature, Atmospheric temperature, Distance and Relative humidity. Lastly, I will discuss External IR windows a bit more thoroughly. So, if you do not intend to use an external IR window, you may just skip that part and move along to the next chapter.

Emissivity

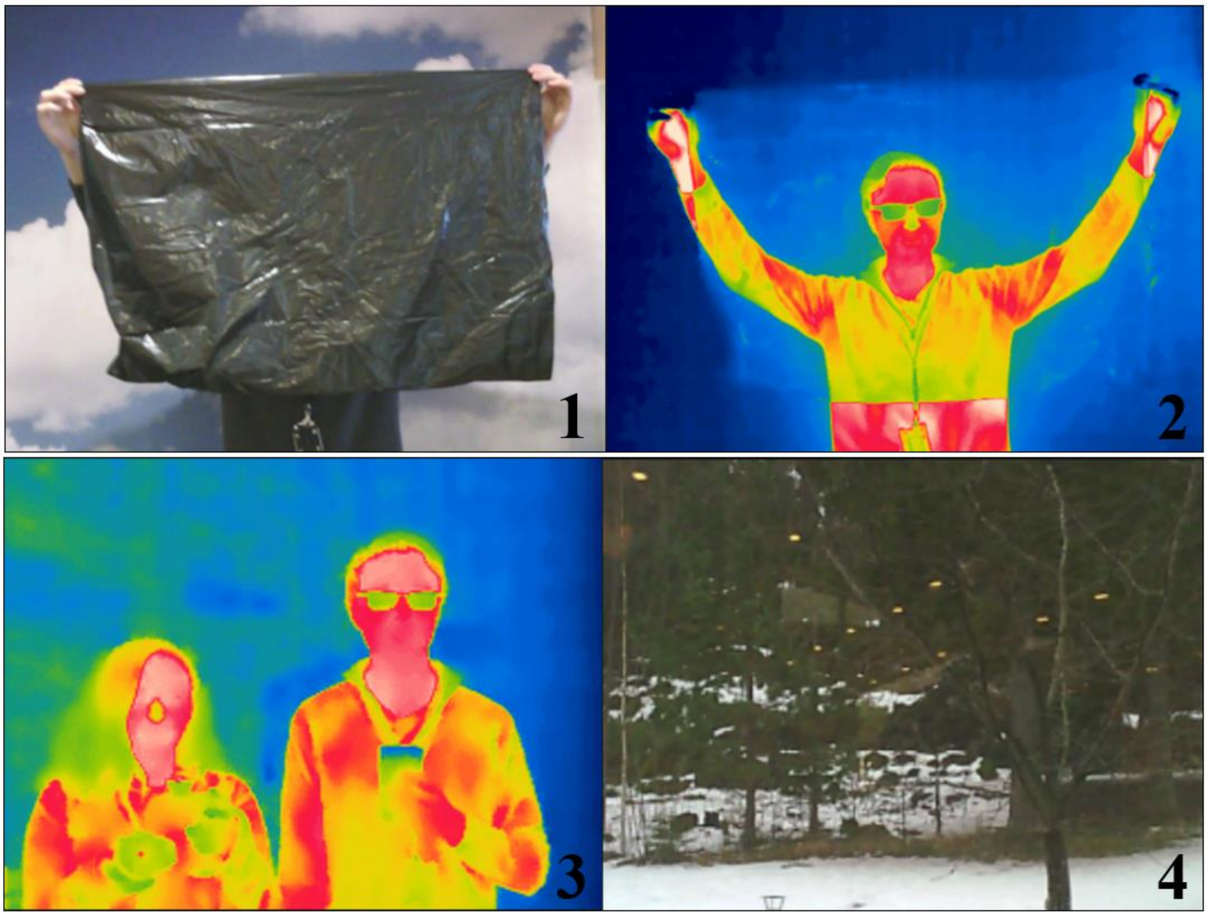
Emissivity is a key concept in IR imaging, and we will stumble upon it many times throughout this course. The full concept of emissivity would take a long time to learn, but my hope is that you will have a good idea about it after this chapter. It is one of the most important parameters to set correctly to get actual readings on the surface temperature of an object. I will therefore spend more time on discussing it. Emissivity is denoted by the Greek letter epsilon, ε.

> **Box tip – Webinar about emissivity**
>
> FLIR has a webinar on YouTube that briefly goes through the subject of emissivity from an IR imaging aspect.
>
> Here’s the link: https://www.youtube.com/watch?v=Seu97i_0nFE
>
> Or you can search for `FLIR webinar emissivity` on YouTube.

A brief look into the concept of emissivity and temperature

Every object emits thermal radiation. In other words, every object transfers heat to its surroundings. At the same time, all surrounding object transfer heat to it. There are two things that determine the amount of thermal radiation of an object. The first thing is the **surface temperature**, which is what we want to measure with IR cameras. The second thing is **emissivity**.

Infrared radiation behaves similarly to visible light in many aspects. It can be **reflected**, just as the light does when you are looking in a mirror. It can be **transmitted**, like light through a window. It can also be **emitted**. The thermal radiation that is emitted from an object tells us about the object’s surface temperature.



To give you an example of emission, transmission and reflection I have taken two photographs. As you probably see, there are four photographs above this text. Both images are shown in the visible spectrum and in the infrared spectrum. Image 1 and 2 are images of the same scene, although it may not look like it. The same goes for image 3 and 4. In the visible spectrum, a black plastic bag is **opaque**. By that, I mean that the visible light cannot pass through the plastic bag. In the infrared spectrum the black plastic bag is transparent. This allows you to see the person behind the plastic bag.

The scene in image 3 and 4 is snow and trees outside of a window. In the visible spectrum (image 4) a glass window is transparent. If you look carefully, however, you can see reflections in the window in image 4. In image 3, you can see that a glass window is opaque in the infrared spectrum. It is the photographers' reflection in the window that is visible in the infrared spectrum. They are quite lively in color; this is the way in which different temperature areas are represented with different colors. Since we cannot see in infrared – or see temperature – we must make it visible through different representations called palettes (see page 36 for more about palettes).

There is no direct way of measuring temperature. We can only measure the *effect* of temperature, not the temperature directly. When we use a regular thermometer to measure a temperature, we are really measuring the volume expansion of the liquid in the thermometer, not the temperature.

The same thing goes for the IR camera, but instead of a change in volume, we measure a change in the resistance as an effect of temperature. We will cover how the IR camera works in more detail in chapter The IR camera system on page 110.

Emissivity is a measure of how much of the thermal radiation from an object is emitted. If an object has emissivity of one ($\varepsilon = 1$), 100 % of the thermal radiation is emitted, and no part of it is reflected or transmitted. Emissivity can take values between 0 and 1, or equivalently between 0 % and 100 %.

The **apparent temperature** is the temperature reported to the IR camera, but this may not be the actual temperature. To see the true – or actual – temperature of an object, we need to know the object's emissivity.

We have now arrived at the first SAQ! Remember what I said in the user guide: Please take a moment and think about the answer before turning the page. I guarantee that the time you take on thinking about the SAQs will pay off in learning outcome in the end. Let's give it a try!

SAQ

Suppose that you want to measure an object's surface temperature as accurately as possible with an IR camera. Would you prefer the object to have **high** or **low** emissivity?

SAQ

Suppose that you want to measure an object's surface temperature as accurately as possible with an IR camera. Would you prefer the object to have **high** or **low** emissivity?

To get a measurement as accurately as possible, we would like a **high** emissivity.

This is because the higher the emissivity, the more of the thermal radiation reaching the IR camera originates from the object. The measured temperature will therefore be closer to the actual temperature.

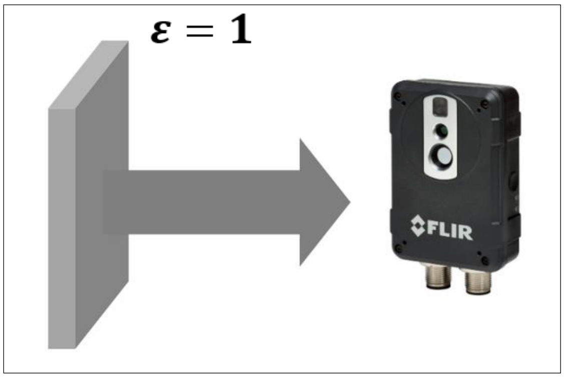
If this was your answer, well done!

If your answer was that you would prefer a **low** emissivity, you might have thought that the reflected or the transmitted radiation can tell us something about the object's surface temperature. This is not the case, since the reflected and the transmitted radiation carries information about the surrounding objects' temperatures. That is, the objects that the reflected and the transmitted radiation originates from. If your answer wasn't correct, don't worry. Most people find this hard at first. It will get clearer as we move along.

You may think of it like this: An object with low emissivity acts as a chameleon. The object camouflages itself to look like its surroundings. If the object is warmer than its surroundings, it will camouflage itself to look cooler. If it is colder than its surroundings, it will look warmer than it is.

Conclusion: Low Emissivity Lies!

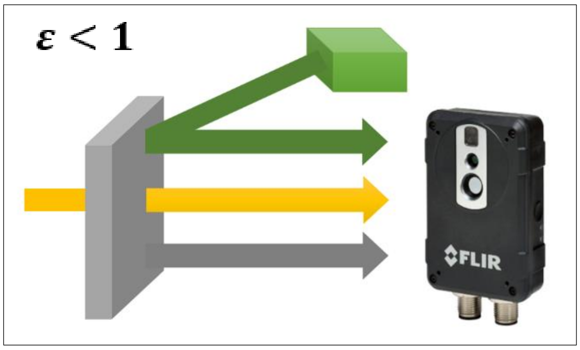
In the image below, the object has an emissivity of one. This means that all the thermal radiation detected by the camera is radiated from the object, and not from other sources. We will get a correct reading of the object's surface temperature if we set the emissivity parameter to one.



The object has an emissivity of one, which means that the apparent temperature is equal to the temperature of the object.

In everyday life, objects with an emissivity of one – called blackbodies – do not exist. No real object that we want to measure can radiate 100 % from themselves (although human skin is quite close, with emissivity from 0.97 to 0.98). If you want to delve deeper into the concept of blackbodies and Planck's law of radiation, refer to the ITC courses (`https://www.infraredtraining.com/`).

As I mentioned earlier, the thermal radiation from the object you're looking at may have been reflected or transmitted from another object. The IR camera cannot see which source the thermal radiation is coming from, it is up to you to deduce.



The apparent temperature of the object is measured partly from the emitted radiation (grey), partly from the transmitted radiation (yellow), and partly from the reflected radiation (green). The object in this image must have emissivity less than one.

Through the parameter setting, you can determine how the IR camera interprets the incoming radiation and present the correct temperature.

Five factors that affect the emissivity

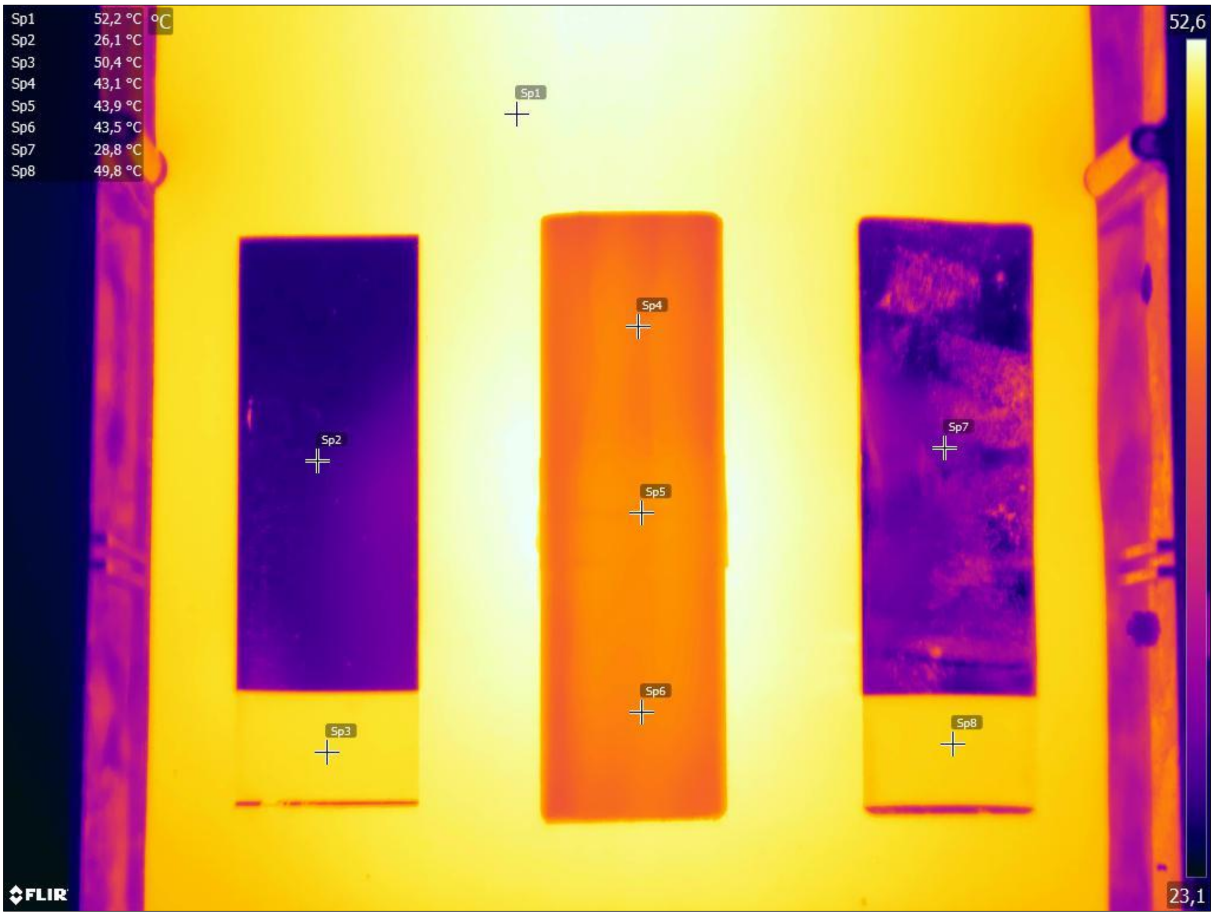
The emissivity of an object is not constant but depends on several factors. In this part, we shall look into a few of those, and see what can be done to compensate for low emissivity.

The five factors are: material, surface structure, temperature, viewing angle, and geometry. Let’s go through the factors one by one.

The perhaps most intuitive factor that affect an object’s emissivity is the **material**. Although there are plenty of materials, we can make a simple classification, namely metals and non-metals. Non-metals such as paper, paints, stones and concrete have quite high emissivity, often above 0.8. This means that these types of materials are easier to get accurate temperature readings from, since over 80 % of the thermal radiation is from the actual object and not reflected.

Metals, on the other hand, are much trickier. They have very low emissivity – especially polished metals – which makes them problematic from an IR imaging perspective. Metals can have an emissivity lower than 0.2. This means that most of the thermal radiation that the IR camera detects does not radiate from the object. This is why we want to avoid measuring the temperature of metals – but there are ways to compensate for it if we have to – as we’ll see shortly.

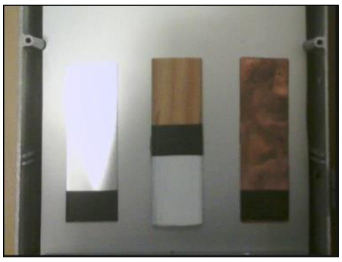
The second factor that affects the emissivity is the **surface structure**. This makes the matter even more tricky. Just because we know the material of our object, does not mean that we know the emissivity. Surface structure plays a major part in an object’s emissivity, especially for metals. The difference in emissivity between a polished and a roughened metal can be huge.



Different materials heated to the same temperature, 52.2 °C. To the left is a piece of aluminum with electrical tape. Spot 2 is on the aluminum and shows a temperature of 26.1 °C. Spot 3 is on the electrical tape and shows a temperature closer to the actual temperature, 50.4 °C.

In the middle is a piece of wood with electrical tape and paint. Spot 4 is on the wood, spot 5 is on the tape, and spot 6 is on the paint.

To the right is a piece of dirty copper with electrical tape, where spot 7 is on the dirty copper, and spot 8 is on the electrical tape.



Looking at iron and steel in the emissivity table below, we see that the difference between polished and rough surface is almost 0.9 in emissivity!

Table 33.1 T: Total spectrum; SW: 2–5 µm; LW: 8–14 µm, LLW: 6.5–20 µm; 1: Material; 2: Specification; 3:Temperature in °C; 4: Spectrum; 5: Emissivity: 6:Reference (continued)

Iron and steel	oxidized	100	T	0.74	4
Iron and steel	oxidized	100	T	0.74	1
Iron and steel	oxidized	1227	T	0.89	4
Iron and steel	oxidized	125–525	T	0.78–0.82	1
Iron and steel	oxidized	200	T	0.79	2
Iron and steel	oxidized	200–600	T	0.80	1
Iron and steel	oxidized strongly	50	T	0.88	1
Iron and steel	oxidized strongly	500	T	0.98	1
Iron and steel	polished	100	T	0.07	2
Iron and steel	polished	400–1000	T	0.14–0.38	1
Iron and steel	polished sheet	750–1050	T	0.52–0.56	1
Iron and steel	rolled sheet	50	T	0.56	1
Iron and steel	rolled, freshly	20	T	0.24	1
Iron and steel	rough, plane surface	50	T	0.95–0.98	1
Iron and steel	rusted red, sheet	22	T	0.69	4
Iron and steel	rusted, heavily	17	SW	0.96	5
Iron and steel	rusty, red	20	T	0.69	1

Emissivity table from the FLIR AX8 user manual

This means that if you roughen up your iron object, the amount of thermal radiation actually originating from it will increase by almost 90 percentage points!

Box tip – Emissivity tables

You can find an emissivity table in the user manual of your FLIR IR camera. It also contains lots of information about emissivity and other aspects that can help you in handling your IR camera.

A word of caution: Use the emissivity tables with care! The values there may differ from your situation, since emissivity is highly dependent on many factors, as we shall see. They do, however, provide a guideline.

To find the user manual for your IR camera, search for `FLIR USER MANUAL` and the name of your IR camera in your web browser.

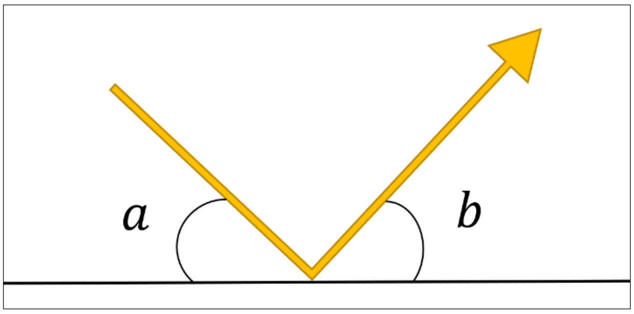
As we look further in the emissivity table above, we see that *oxidation* and *rust* also affects the emissivity value, and this goes for all metals. So, if you are monitoring an object with metal parts – such as bolts and nuts – their emissivity might change with time and the temperature reading will change.

Looking even further in the table, we see that the **temperature** also affects the emissivity. This may come as a surprise, since it's the temperature we want to measure. To know the temperature of an object, we need to know the emissivity. But to know the emissivity, we need to know the temperature. It is actually quite intuitive. You may think of emissivity as a property of an object, like color (in the visible spectrum). But the color of the object can change with temperature. Iron, for instance, glows red when heated. The main point to remember is that emissivity can change with temperature, so it is important to know in which temperature interval you're monitoring to get an accurate temperature reading.

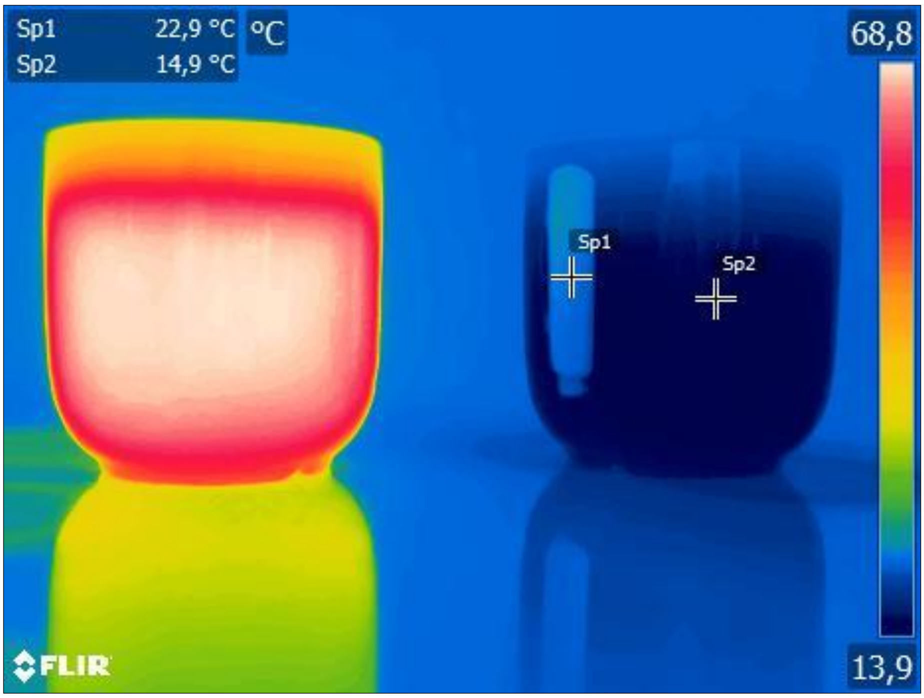
Earlier, we discussed that some of the thermal radiation entering the IR camera will be reflected radiation from other sources. This is why the **viewing angle** is important when measuring surface temperatures. If there is a source for thermal radiation nearby your object, it will probably reflect in your object. Being aware of this, and of the rule "angle in = angle out" can be of great help.

The Law of Reflection:

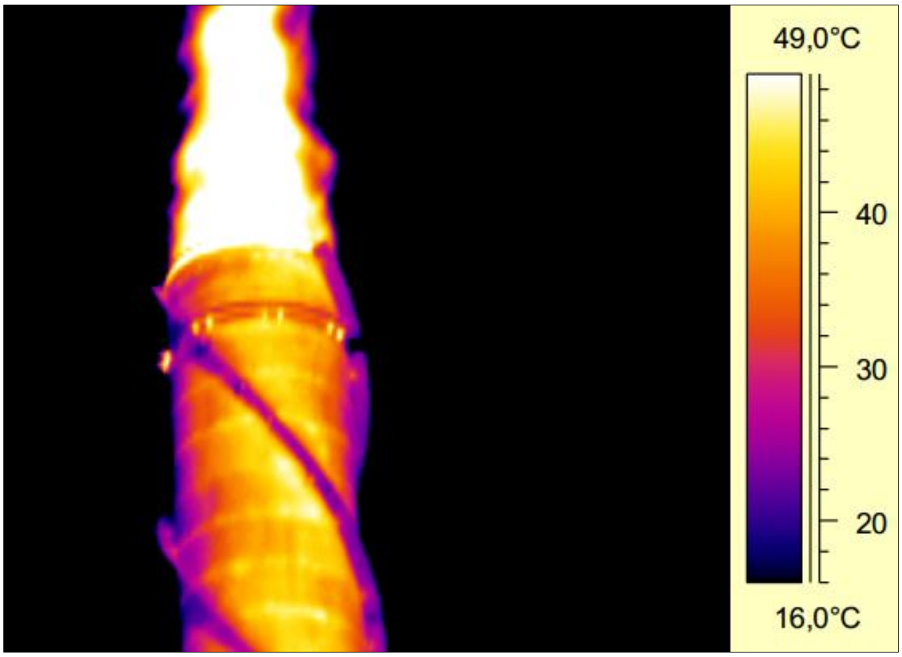
The angles formed by the incident and reflected radiation are the same. In other words: $a = b$.



By using the law of reflection, you can deduce where the sources of reflection are. Either you can move your IR camera, so that the reflection does not affect your image, or you can account for the reflection.

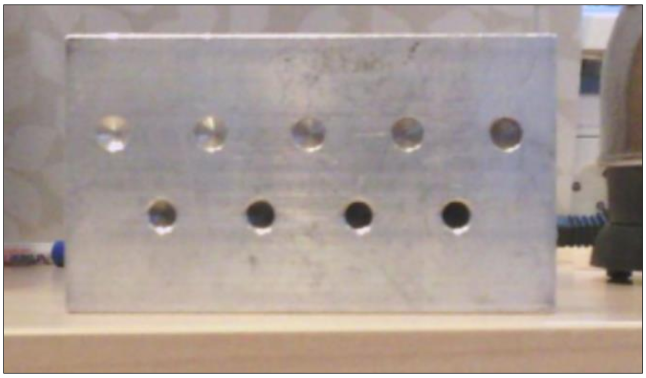


This is an IR image of two cups – one warm and one cold. If you look at the temperature readings on the cold cup, you can see that they differ. Spot 1 is directed at the reflection of the warm cup, so the most accurate measurement is probably spot 2.



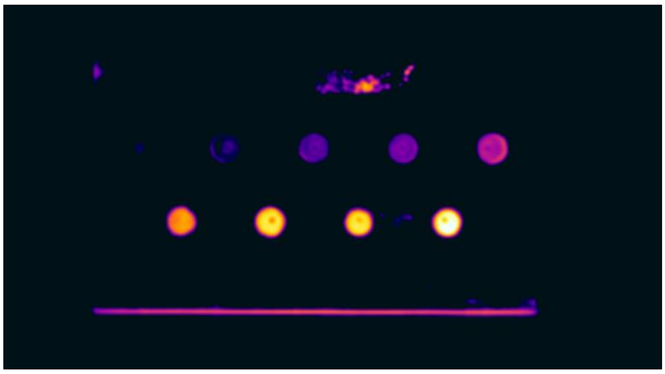
The geometry of the chimney makes it look colder around the edges than on the middle, because of the reflection angles. The areas that look colder are probably reflection from the sky.

If you suspect that a hot spot might be a reflection, try moving the IR camera around. If the hot spot moves, it is a reflection!



Visual image of a heated aluminum brick with holes of different depths.
The brick is evenly heated to approximately 130 °C.

When looking at an IR image of the 130 °C aluminum brick in the image above, we might expect it to show a glowing brick. Still, the IR image that we get is this.



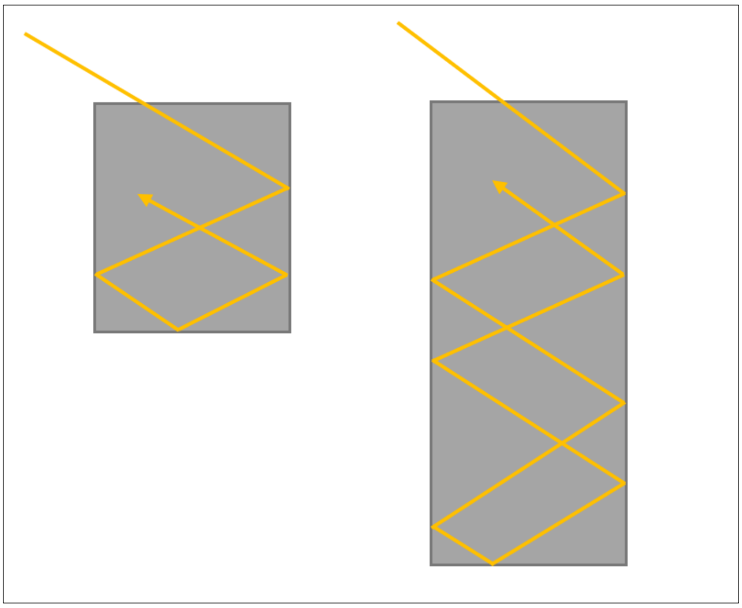
IR image of the aluminum brick.

The resulting IR image is because the holes in the aluminum brick has different depths. The **geometry** of the object also affects the emissivity. The deeper holes appear to be warmer than the shallower ones. We know that this is not true, since the aluminum brick is evenly heated.

Let's think about this for a moment. We saw earlier that the material and surface structure affect emissivity, and subsequently the apparent surface temperature. Aluminum is a metal and the surface on the visual image looks quite smooth, so an educated guess would be that the brick has low emissivity. If our guess is correct, then most of the thermal radiation reaching our IR camera is reflected from other sources, not from the aluminum brick. Casting a second glance on the IR image above – and ignoring the holes – we see that this is probably the case. The surface of the aluminum brick appears to have the same temperature as the surroundings. But the brick is 130 °C! A much higher temperature than room temperature. So, the measured temperature on the brick surface must be reflected.

Now to the holes, or cavities. You may think of it like this: Thermal radiation from other sources enters the cavity. The radiation is then reflected back and forth inside the cavity. For every hit against the cavity wall, one of two things can happen. Either the thermal radiation is reflected again, or it is absorbed by the material. If it is absorbed, it will be emitted. The deeper the cavity, the more hits on the cavity wall before the radiation can escape. The more hits on the cavity wall, the higher probability that it will be absorbed, and subsequently emitted.

SAQ



To the left you see two cavities of an object with same temperature, but different depths. You want to measure the temperature of the object with an IR camera.

Should you measure in the **deeper** or **shallower** cavity to get a more accurate temperature measurement of the object?

SAQ

Should you measure in the **deeper** or **shallower** cavity to get a more accurate temperature measurement of the object?

The correct answer is in the **deeper** cavity. Let's look at why.

Radiation will enter the cavity. Every time the radiation hits the cavity wall, one of two things can happen:

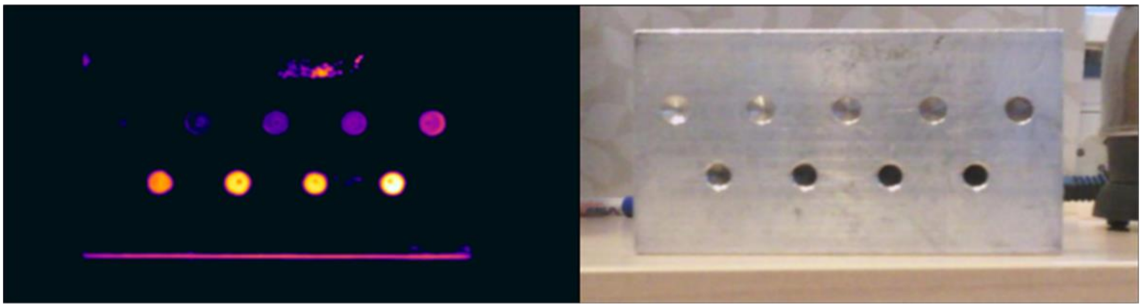
1. The radiation is **reflected** – it bounces off the wall.
2. The radiation is **absorbed** by the material – and then **emitted**.

The deeper the cavity is, the more times the radiation will hit the wall. This makes it **more likely** for the radiation to eventually get absorbed. Radiation that is **absorbed** will also be **emitted**.

A deeper hole then means that more radiation will be emitted from the **object itself**. In other words, the deeper hole has **higher emissivity** and a higher emissivity makes it easier for us to get an accurate temperature measurement with the IR camera.

This will allow our IR camera to make a more accurate measurement and the apparent temperature of the deeper hole will be close to the actual temperature of the object.

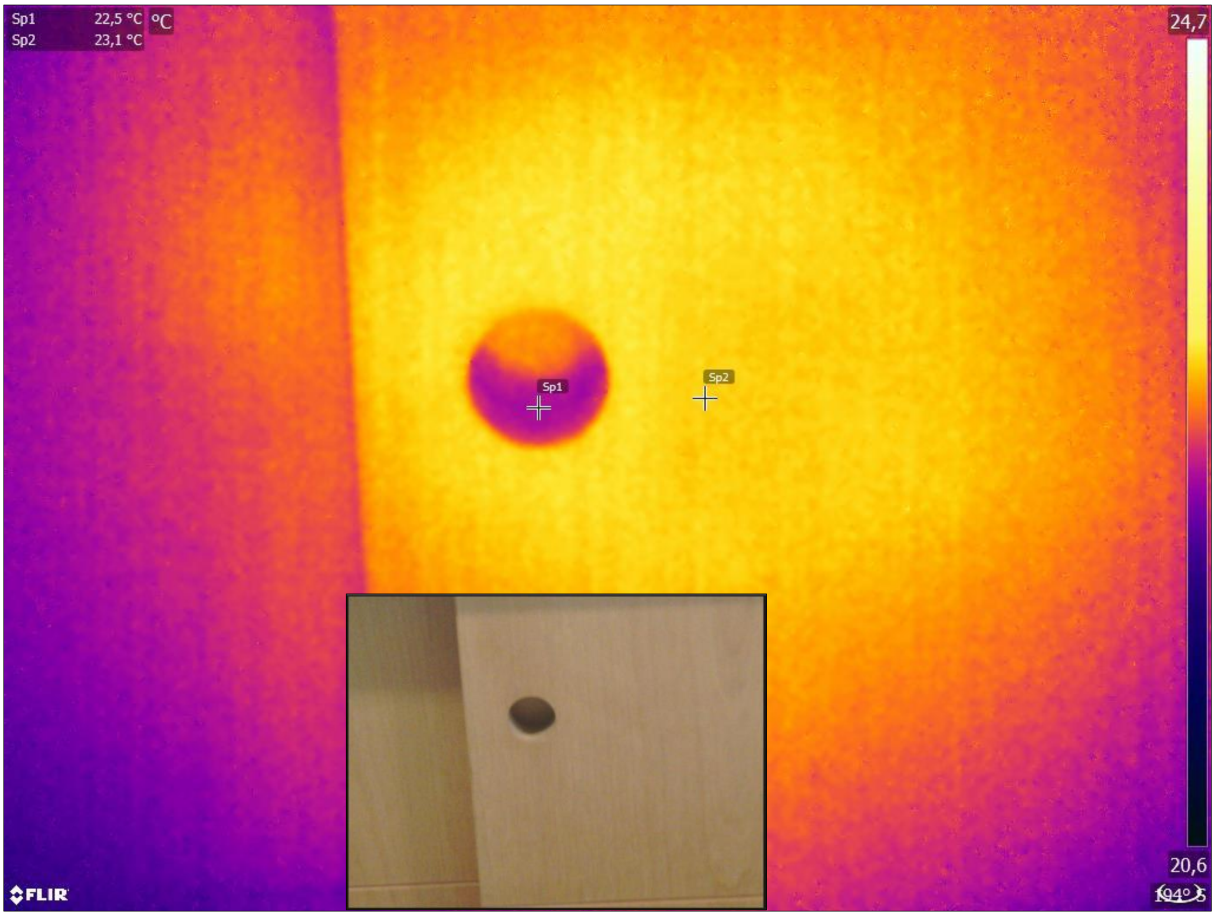
Let's connect this to the heated aluminum brick from earlier. The brick is quite hot – 130 $°C$. We can "see" that high temperature in the deeper cavities down to the right, while the shallower cavities look quite cold.



This gives us a way to measure the temperature of an object with normally low emissivity, drill a hole in it!

If your reasoning was in line with my response, that's fantastic! If not, fantastic! This is tricky business and by no means easy at first. However, the fact that you have gone through this response is great and I hope that it has made things a little clearer.

This phenomenon occurs in many different geometries, and not just holes. Emissivity in the corners of bolts, for instance, might be higher than the surroundings. It can therefore give a more accurate temperature reading there, than on a plain surface.



The shallow hole in a cabinet door shows different temperatures because of the slight cavity. The brighter area is my thermal reflection in the shiny wood surface.

How can I determine emissivity?

By now, I hope that I've made it quite clear that emissivity is a key concept in IR imaging. We've looked into some factors that affect the emissivity of an object, now, let's discuss how we can determine it.

You can turn to an emissivity table. These are not hard to find; you can search on the web or look in the manual of your IR camera. I repeat the caution from the Box tip earlier: These emissivity values may not be completely accurate in your situation! There are ways to determine emissivity by yourself, so that all conditions for your situation are correct.

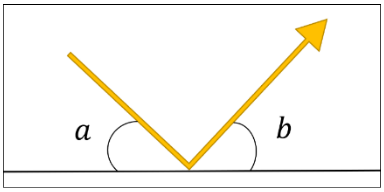
To determine the emissivity by yourself, you first need to determine the reflected temperature. There are two ways to do this: direct method or reflector method. Before we go through them, I want to give you a word of advice. Since the reflected temperature is highly dependent on the surroundings of the object, the best place to perform these methods is where the "real" measurements are going to be.

There is some gear you’re going to need, depending on which method you use.

- Tape or paint of known (and high) emissivity – for both methods
- Large piece of cardboard – for both methods
- Large piece of aluminum foil – for the reflector method

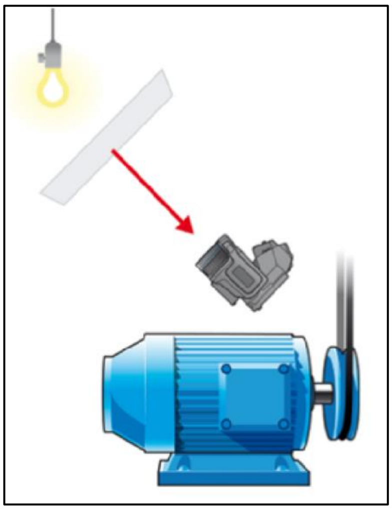
The direct method

First, you need to find possible reflection sources. The reflection law I mentioned earlier can be of help ($a = b$, in the image to the right).



If the reflection is from a spot source – like a light bulb – you can use a piece of cardboard to obstruct it.

In the settings of your IR camera, set the `Emissivity` to `1.0`, and the `Distance` to `0.0`, or as low as possible.



Aim the IR camera towards the source of reflection, or the cardboard, and write down its apparent temperature. We will use this as the reflected temperature later.

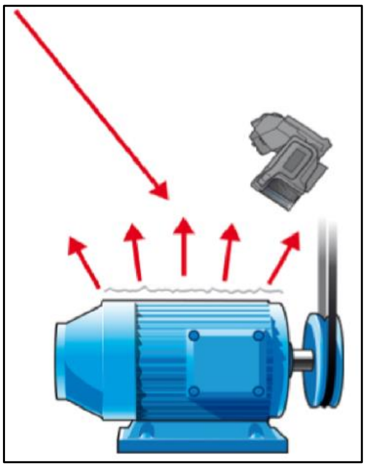
Reflected Temperature:

The reflector method

To use this method, you are going to need the aluminum foil and the cardboard.

First, crumble the aluminum foil. Then uncrumble it and attach it to the cardboard. Place the cardboard on your object, with the aluminum foil facing towards the camera.

In the settings of your IR camera, set the `Emissivity` to `1.0`, and the `Distance` to `0.0`, or as low as possible.



Aim the IR camera towards the aluminum foil and write down its apparent temperature. We will use this as the reflected temperature later.

Reflected Temperature:

Determining the emissivity

To determine the emissivity of your object, we are going to use an electrical tape of known emissivity. Attach the tape on your object, and make sure it has good thermal contact. This can be difficult if the surface of your object is very rough. In that case, you can use a paint of known emissivity and paint a part of your object. When you've attached the tape or painted part of your object, make sure that you wait a little while so that the tape or paint has the same temperature as your object.

Aim your IR camera towards the tape or paint, such that the frame is half filled with tape or paint and half filled with your object. Freeze the image, so that it is easier to work with. Set the reflected temperature of your IR camera to the one you measured earlier and the emissivity to that of the tape or paint. Measure the temperature of the tape, preferably with a spot or box average (see chapter Analytics on page 72).

Write down the temperature of the tape or paint.

Temperature of tape/paint:

Lastly, move the spot or box to your object and change the emissivity setting until the apparent temperature of your object is the same as the temperature of the tape or paint.

When the temperature matches, you will have the emissivity of your object. Well done!

ε : $\boxed{}$

How can I compensate for low emissivity?

There are different ways that you can compensate for low emissivity in order to get accurate temperature recordings. Some of them, we have covered earlier in this chapter, but I'll give you a quick run-through on them.

In the part about factors affecting emissivity, we started to discuss the material of the target object. I have two pieces of advice for you regarding this:

(1) Don't measure objects that are transparent, like plastic for instance. When a material lets thermal radiation through, it is almost impossible to determine where the sources are and the reflective temperature.

(2) Don't measure polished metals or any object with an emissivity lower than 0.5, as the error in the temperature reading is becomes larger for lower emissivities. This is precisely what we will look at in the next SAQ.

SAQ

The importance of knowing emissivity

This SAQ is thought to illustrate the importance of knowing the emissivity of the object and why we prefer an object to have a high emissivity.

I want you to create a spot and make 4 measurements with different emissivity settings in the camera. I will use my FLIR AX8 to do this.

Aim your camera at an object, preferably warmer than the surrounding. It could be a lamp, coffee cup or yourself. I am using a ceiling lamp.

Create a spot by clicking the  icon.

Drag the spot to your object.

Click on the 

icon. Select local parameters on.

Change the emissivity to 0.95, 0.90, 0.25 and 0.20. Write down the temperature for each emissivity in the table below. Then calculate the temperature difference for the high emissivites and low emissivites respectively.

Emissivity	Temperature in spot	Temperature Difference
0.95		
0.90		
0.25		
0.2		

Compare the temperature differences between high emissivity (0.90 and 0.95) and low emissivity (0.20 and 0.25). Which **temperature difference** is larger?

SAQ

I hope that you found that **low** emissivity gave the biggest temperature difference. If you did not come to this conclusion, it may have been due to your measurements. Perhaps you measured different parts of your object when changing the emissivity?

This is how I filled in my table.

Emissivity	Temperature in Spot (°C)	Temperature Difference (°C)
0.95	35.7	0.8
0.90	36.5	
0.25	71.4	10.7
0.2	82.1	

Notice that the **emissivity differences** between high and low are the **same**, $0.95 - 0.90 = 0.05$ and $0.25 - 0.20 = 0.05$. However, the **temperature difference** is much **larger** for the low emissivities (10.7 °C) compared to the high emissivities (0.8 °C). This means that the **inaccuracies** in temperature measurement become larger when the emissivity is low.

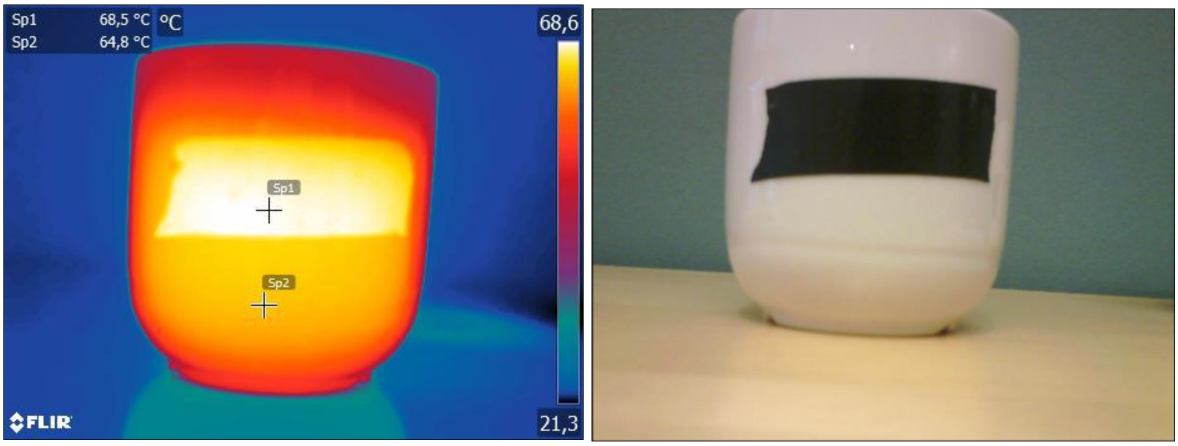
Here we see another reason why we prefer out object to have high emissivity. To summarize, if our object has low emissivity, then we have two problems:

- Only a small portion of the radiation reaching the camera will come from the object itself. This means that an object with low emissivity "hides" its true temperature to us. In other words: low emissivity lies!
- Large inaccuracies. A small error in the emissivity setting causes a large error in the temperature. This means that our temperature measurement becomes more uncertain the lower the emissivity is.

By now, you might have realized that it is very difficult to determine an object's emissivity precisely; we will always make at least a tiny error. Therefore, we want to measure on objects with high emissivity so that the error doesn't affect the temperature measurement as much.

If you must measure on a surface with low emissivity, there are ways to come around it. This is what we will cover next.

If you do want to measure an object with low emissivity, you can **roughen up the surface** or even **drill a hole** in it. This will give you higher emissivity and more accurate readings. If you don't want to change your object in such a drastic manner, you can use an **electrical tape** or **paint** with high emissivity and cover the spot you're measuring. If you use tape or paint, make sure that it has a good thermal connection to your object!



A warm cup with electrical tape. Notice the temperature difference in the measurement on the electrical tape and on the cup.

Reflected temperature

If you recall from the part Determining emissivity, you measured the reflected temperature. As you've probably already guessed, this parameter is used to compensate for the thermal radiation reflected in your object. I will not linger any more on the subject other than to mention that this parameter is more important to set accurate if your object has low emissivity and if the difference in temperature between your object and the reflection is large.

Atmospheric temperature

The atmospheric temperature – simply put – is the temperature of the atmosphere between your target object and the IR camera. In the visible spectrum, the atmosphere around us is fully transparent, unless it is a foggy day or the like. In the IR spectrum, the atmosphere must be treated as an object blocking the IR camera's view of your object. This means that the atmosphere can reflect, transmit and emit radiation, just like every other object. Fortunately for us, the atmosphere is mostly transparent, but it's still an important parameter to set fairly accurate. Set it to the room temperature where your object is being monitored and you'll be fine.

Distance

The distance parameter is pretty straightforward. It is the distance between your object and the IR camera. One thing to keep in mind is that if the distance is large, there will be more atmosphere in between the IR camera and the object. So, as the distance increase, the importance of setting the atmospheric temperature setting increases.

Relative humidity

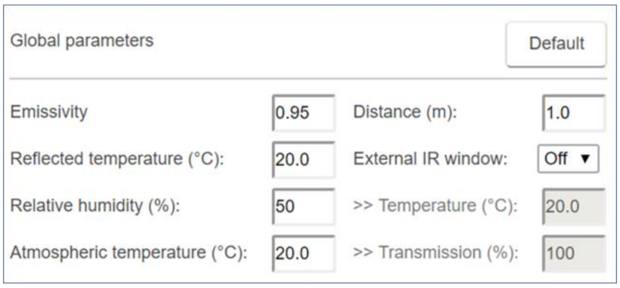
The IR camera can also compensate for the fact that the transmission through the atmosphere is dependent on the relative humidity. This parameter does not affect your measurement by much – unless you are measuring temperatures in a rain forest or in the desert – so you may leave it on the default setting. This is usually 50 %.

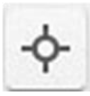
SAQ

The importance of different object parameters

In this chapter we cover the different object parameters you set in your IR camera, but is every parameter equally important? In this SAQ, I want you to investigate the importance of each object parameter.

The five parameters I want you to focus on are emissivity, reflected temperature, relative humidity, atmospheric temperature and distance. I will use my FLIR AX8 for this. In the FLIR AX8 interface, you change the parameters by clicking the Global measurement parameters icon .



To make your observation easier, choose an object and measure its temperature with a spot. I am measuring a ceiling lamp. Create a spot by clicking the  icon and then drag it to your object.

How will different parameter settings affect the apparent temperature displayed in the camera?

Play around with the object parameters and see how it affects the apparent temperature. Based on your observations, list the object parameters in order from most important to least important.

1.
2.
3.
4.
5.

SAQ

How will different parameter settings affect the apparent temperature displayed in the camera?

This is how I listed the object parameter in the order from most to least important.

1. Emissivity
2. Reflected temperature

3. Distance
4. Atmospheric temperature
5. Relative humidity

First, I want to say that there is no single correct answer here. It depends on your measurement and what different setting you tried. Thus, you should only view my list as a suggestion. However, I argue that the 2 **most important** parameters are **emissivity** and **reflected temperature** while distance, atmospheric temperature and relative humidity have less impact on the apparent temperature. Hence, the gap between 2 and 3 in my list above.

By now, I hope that I have conveyed that emissivity has a central role in thermal imaging. I would now like to comment some more on reflected temperature, since it is highly related to emissivity. As I said earlier in this chapter, the reflected temperature is the temperature of **another object** whose radiation is **reflected** by the object we're studying. This reflected radiation complicates temperature measurements, because the reflected radiation does not originate from our object. By knowing the temperature of the reflected object, the IR camera can compensate for this effect and give a more accurate temperature reading.

When you tried different values for the reflected temperature, you might have noticed both small and large impacts on the apparent temperature. This depended on what emissivity you had set. This is because **higher emissivity** means **less radiation** is **reflected**. So, the reflected temperature will have a lesser impact on the temperature if the emissivity is higher. You may test this by setting the emissivity to 1. This will make the IR camera to interpret that all radiation is originated from the object itself. In other words, no radiation will be reflected. This means that no matter what reflected temperature you set; the IR camera's temperature reading will not change. What I am saying is that if you set the emissivity to 1, then the reflected temperature setting won't matter. It could be as high as 1500 °C or as low as −200 °C, it won't change the apparent temperature. Don't believe me? Check it out yourself!

I hope that this SAQ has made you feel more comfortable with changing parameters in your IR camera as well as understand the impact that different object parameters have on the apparent temperature displayed in the IR camera.

External IR window

The external IR window can be used when the object you want to measure is in an enclosed space, or if the object you're measuring is at risk at injuring you – such as arc flash accidents. The IR window is a great tool for measuring objects that need to be behind walls, or other protective barriers, and it is easy to install.

The two parameters that need to be determined when using an external IR window is the **External IR window temperature** and the **External IR window transmissivity**.

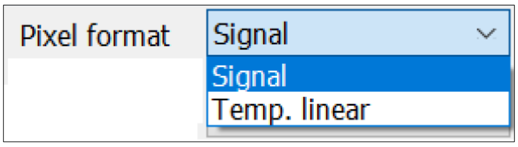
The temperature of the IR window can usually be set to the same as the atmospheric temperature, unless you have a contact thermometer to measure the IR window temperature directly.

There is no table for the transmissivity of the IR windows, since it may vary with the situation and surroundings. There are two ways for you to measure it, though – and I'll walk you through them. Remember that the transmissivity of the IR window must be determined *before* it is installed for the application.

Window transmissivity estimation, method 1

The first method requires you to have an IR camera whose output you can convert to counts or signal, or an image streaming camera (see chapter Image stream on page 131).

If you are working in FLIR GEV DEMO, set the `Pixel format` and `Presentation` to *Signal*.



Pixel format on the FLIR GEV DEMO interface

For this method, you are going to need the following equipment.

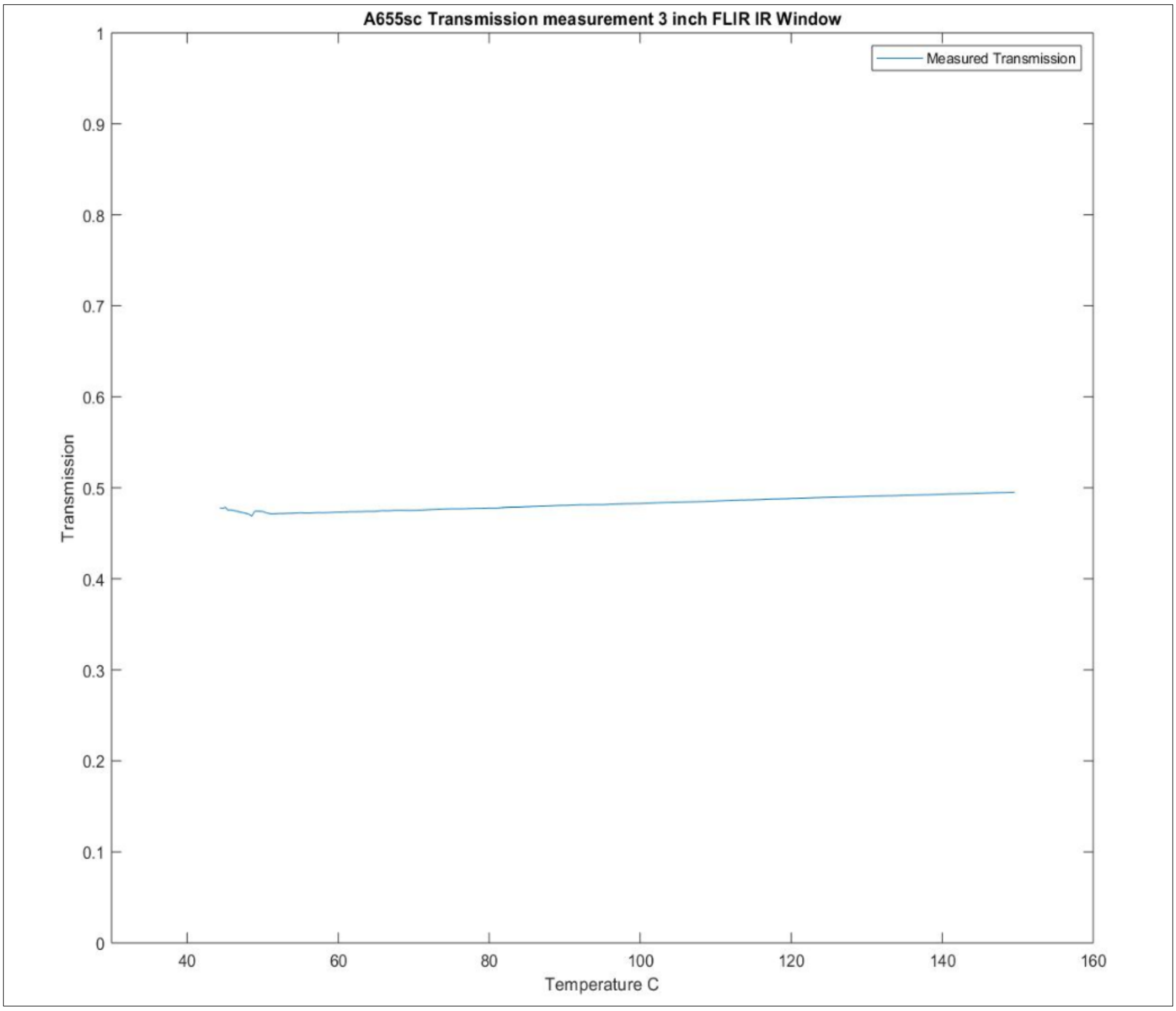
- A relatively large object that can be kept at an even temperature, preferably in the temperature interval you plan on measuring in. (If you happen to have a blackbody extender – use that. That would be optimal).

- An electrical tape with an emissivity that *differs* from that of the object described first in the list.

- Analytics software of some kind, for example FLIR Tools, FLIR ResearchIR or your own program. The analytics we are going to use are spot or box measurements.

When doing this estimation – the best option would be to test the IR window on several temperatures, but it's okay if you don't have that possibility. The difference in transmittance is not that large, when working in a not too big temperature interval.

Here's an example of the transmissivity of an IR window in the temperature interval 40 °C to 150 °C.



As you can see, the line is pretty straight around 0.47 – 0.48, corresponding to 47 % - 48 %.

Now, to the measurement. Make sure that your object has an even temperature, and also that the electrical tape has good thermal contact with the object.

Your setup should look something like this. The IR camera should have a good view of both tape area and object area, both through the IR window and outside.

The interface should look something like the image below.

Before you make your measurements: Make sure that the object and the tape has the same temperature and good thermal contact!

Place four spots like in the image above – or better yet four boxes, make sure that they only cover tape/object and not the edge of the IR window.

The formula for calculating the transmissivity (τ) of the window is

$$\tau = \frac{Cursor\ 3 - Cursor\ 4}{Cursor\ 2 - Cursor\ 1}$$

*Don't forget that your measurements should be in **counts/signal** and not temperature!*

Window transmissivity estimation, method 2

This method is not as accurate as the first, but it's easy and enough for standard users (not R&D). It also works for all IR cameras – smart sensor cameras as well – and not just image streaming cameras.

Here's what you do:

1. **Set** the emissivity to `1.00` and the distance to `0 m` (or as low as possible).
2. **Measure** and **set** the reflected temperature (see chapter How can I determine emissivity? on page 56).
3. With the emissivity still set to 1.00, **measure** the apparent temperature of a hot object.
   Write the apparent temperature down: $\boxed{}$
4. **Place** your IR window between the IR camera and the hot object, so that the IR camera looks through the IR window.
5. **Change** the emissivity parameter until the temperature reading is equal to the apparent temperature you wrote down.
6. The emissivity you get when the temperatures match is the transmissivity (τ) of your IR window.

$$\tau: \boxed{}$$

SUMMARY

In this part, we have discussed the image processing of an IR camera. I will give you a list with the main parts that we've covered. When you go through the list, try to reflect over what you have read.

> Every object emits thermal radiation and the amount of thermal radiation is determined by the object's surface temperature and emissivity.
>
> IR radiation can be reflected, transmitted and emitted. The emitted radiation from an object is what tells us of its surface temperature.
>
> Emissivity is a measurement of how much of the thermal radiation that originates from the object.
>
> Emissivity can take values between 0 and 1.
>
> The parameter settings in your IR camera tells the camera how it should interpret incoming thermal radiation.
>
> Factors that affect emissivity are:
>
> - material
> - surface structure
> - temperature
> - viewing angle
> - geometry
>
> There are ways, besides looking at emissivity tables, to determine emissivity.
>
> You can compensate for low emissivity by
>
> - roughening up the surface
> - drilling a hole
> - using electrical tape or paint

Reflected temperature is the parameter that compensates for the thermal radiation reflected in your target object.

Atmospheric temperature is the temperature of the atmosphere between your target object and the IR camera.

The distance parameter is the distance between the IR camera and your target object.

The relative humidity parameter is usually set to 50 %.

An external IR window can be used if the measurements are taken in an encapsulated place. There are ways of determining the external IR window’s transmission.

Analytics and alarms

The FLIR AX8 and the FLIR A310 comes with a variety of tools for analytics and alarms. Depending on the application, some tools are better suited than others. Thus, it's important for you to be familiar with the different measurement tools, to know which measurement is best suited for a particular application.

The main aim of this chapter is to provide you with possibilities to increase your *practical* knowledge of the different analytics and alarms features. So, while going through this chapter, make sure to have your IR camera connected and try the features yourself!

Analytics

Objectives

When you have worked through this unit, my aim is that you will be able to answer these questions

> Which measurement tools are available for analytics with the FLIR AX8 and the FLIR A310?
>
> How do I set these up?
>
> What are the differences?

The Analytics chapter can be divided into 7 features:

- Spot
- Box
- Delta
- Isotherm
- Iso-coverage
- Mask
- Schedule

Box tip – Supplemental material

There are plenty of YouTube videos that can be of help when you are discovering the features of your IR camera. I'll list the names of the clips below, and you can just search for them on YouTube or enter their web addresses.

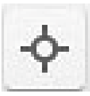
IR Monitor Tutorial - Analysis & Alarms
`https://www.youtube.com/watch?v=fC0UY3CX3LY`

FLIR AX8: Features Walkthrough
`https://www.youtube.com/watch?v=gA-GXixDYrs`

FLIR AX8: Measurement Tools
`https://www.youtube.com/watch?v=ydyZKJpX20k`

Don't forget that there is a Help tab in the web interface for the FLIR AX8 and an instructional PDF to download in IR Monitor for the FLIR A310 – just click the Help tab!

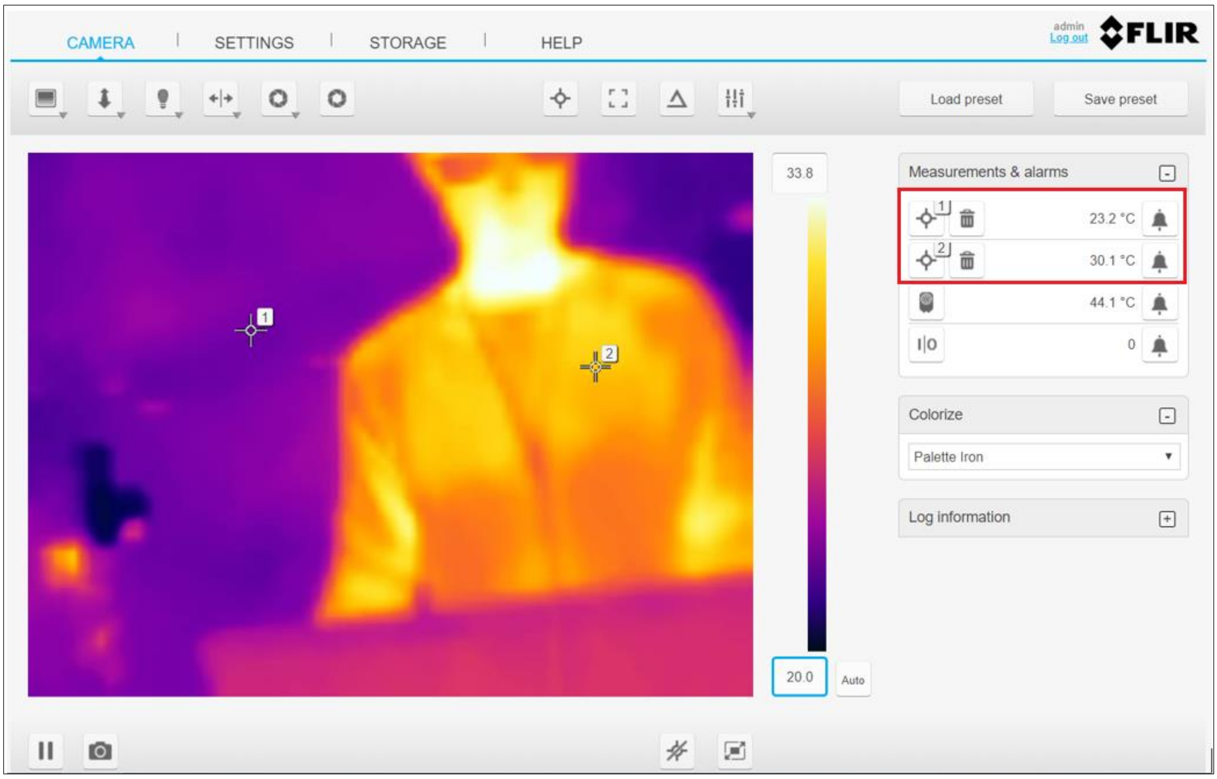
For each measurement tool, I will give a short description and then walk you through how to set it up. Let's start with perhaps the simplest analytics tool: The Spot.



Spot measurement

The spot measurement, or simply *spot*, displays the apparent temperature of a *single point* on the monitor. In the image below, I have created two spots. Notice that Spot 2 (30.1 °C) displays a higher temperature than Spot 1 (23.2 °C). This is because Spot 2 is placed at me while Spot 1 is placed in the background, and I'm warmer than the background. With the FLIR AX8, it's possible to add up to six spots.

It good to know that the spot measurement can only be accurate if the circle fits entirely within the object of interest. In other words, don't measure on objects that are too small.

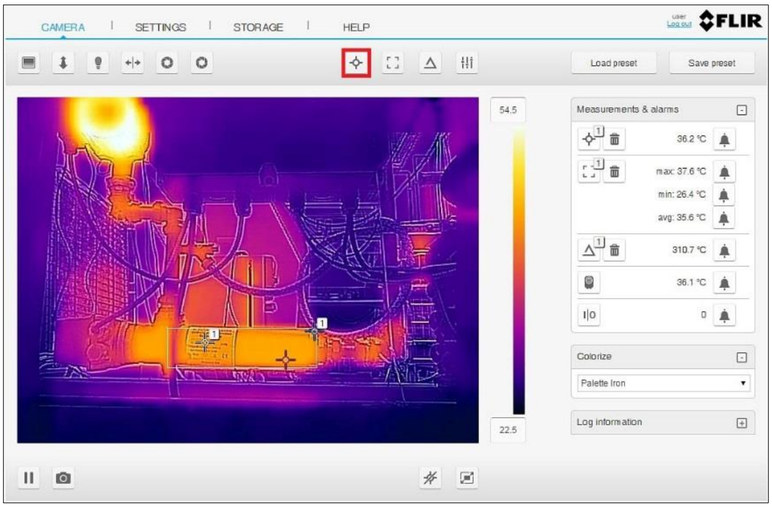


With the FLIR A310, it's possible to add up to ten spots.

How do I set up a spot measurement?

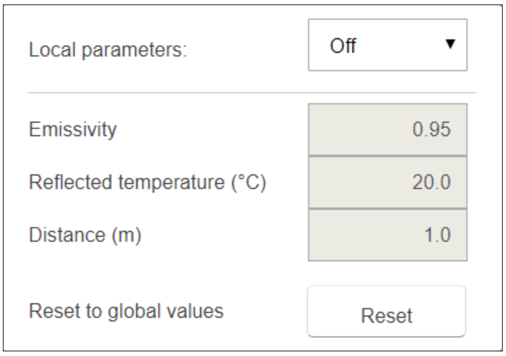
FLIR AX8

 On the upper toolbar, click the `Spot measurement` icon. This displays a spot in the image, labeled with a number. The spot tool is also displayed in the `Measurements & alarms` section.



To move the spot, click the spot in the image and drag it to the desired location.

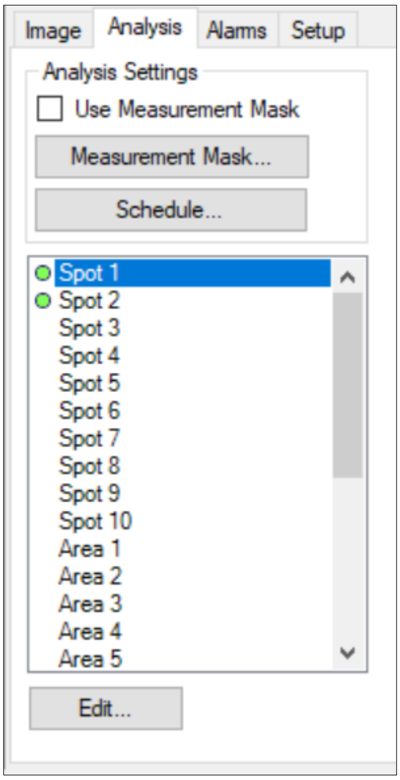
 In the `Measurements & alarms` section, click the `Spot` icon. This displays a dialog box where you can set `local parameters` for the spot, such as *Emissivity*, *Reflected temperature (°C)* and *Distance (m)*. To learn more about these *Object parameters*, refer to chapter Object parameters.



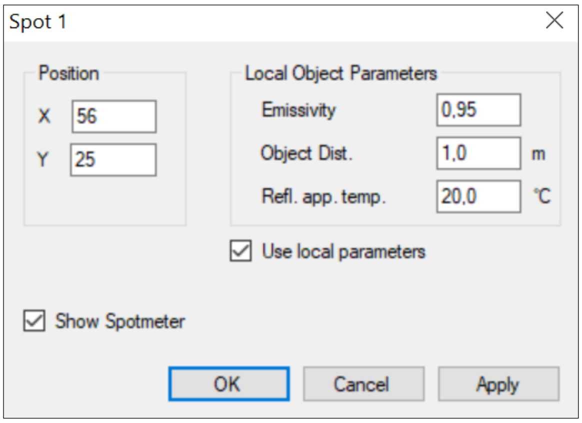
To associate an alarm with the spot, see section Alarms further down in this chapter.

 To remove the spot, click the `Delete` icon next to the tool in the `Measurements & alarms` menu.

FLIR A310



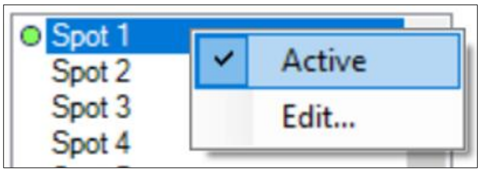
Mark `Spot 1` by clicking on it, then click `Edit…` . This displays a dialog box where you can set the `X` and `Y` position of your spot, measured in pixels from the top left corner of the IR image. If you don’t know the `X` and `Y` position of your spot, just click `OK` and then you can drag the spotmeter to the desired location in the IR image.

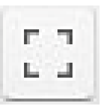


You can also choose to use *Local Object Parameters*. More on object parameters in chapter Object parameters (page 43).

Tick the `Show Spotmeter` box to see the spot in the interface.

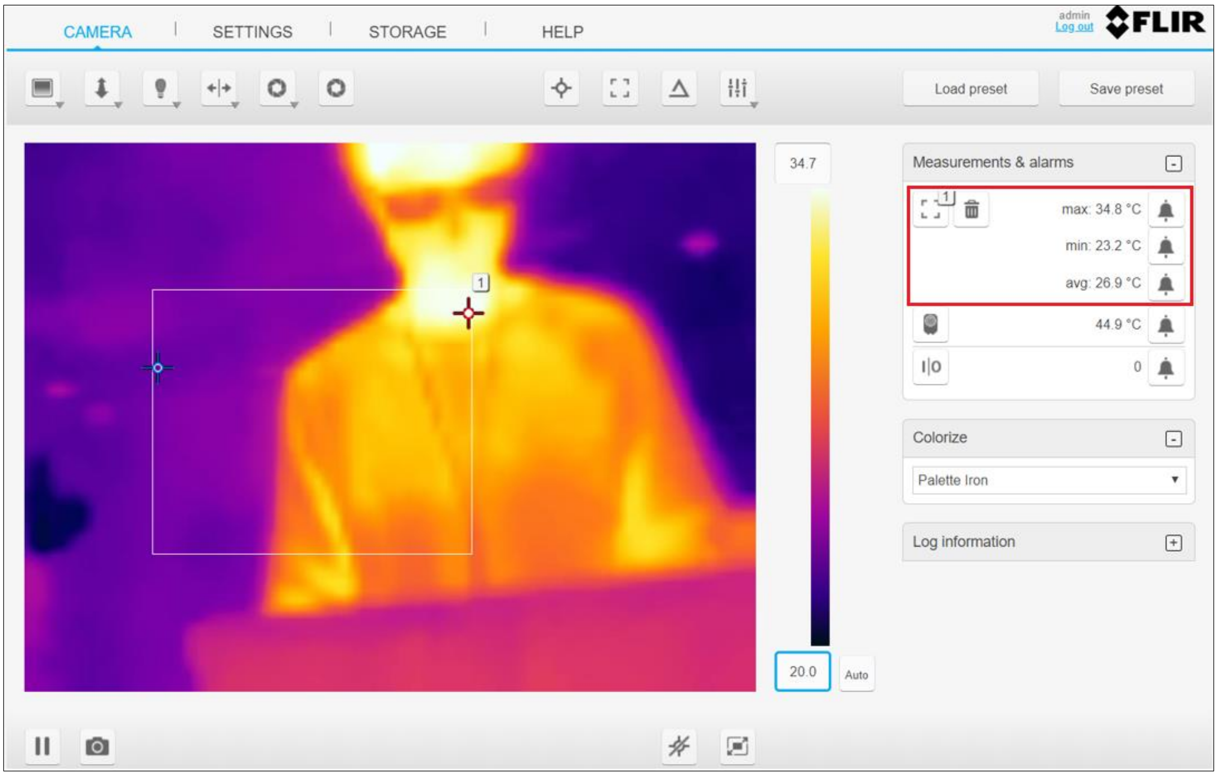
To remove a spot – right-click the spot in the analysis pane and click `Active`.





Box measurement

The box measurement displays the minimum, maximum and average apparent temperature within *a chosen area* of the image. In the image below, you can see that these are 34.8 °C, 23.2 °C and 26.9 °C respectively. The blue spot shows the minimum temperature in the box and the red spot shows the maximum temperature. With the FLIR AX8, you may add up to six boxes.

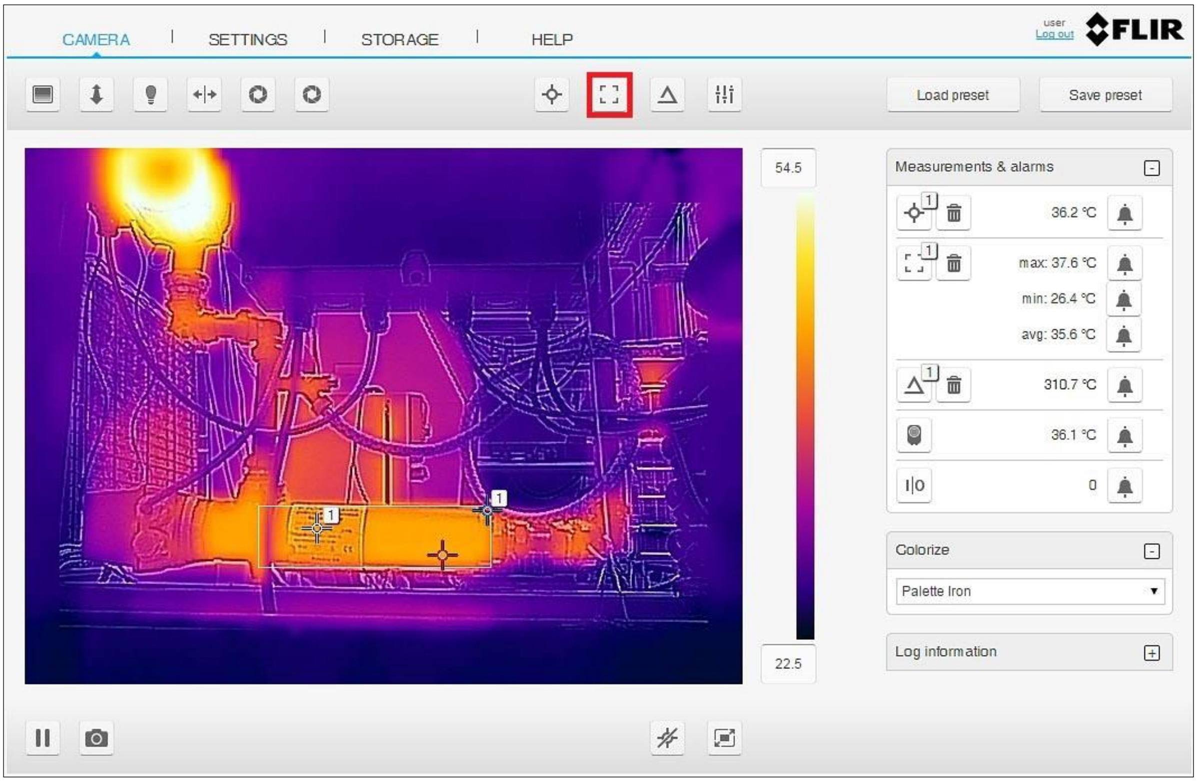


With the FLIR A310, you may add up to ten area measurements (box measurements).

How do I set up a box measurement?

FLIR AX8

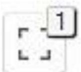
 On the upper toolbar, click the `Box measurement` icon. This displays a dialog box in the image, labeled with a number and including a hot spot and a cold spot. The box tool is also displayed in the `Measurement & alarms` section.



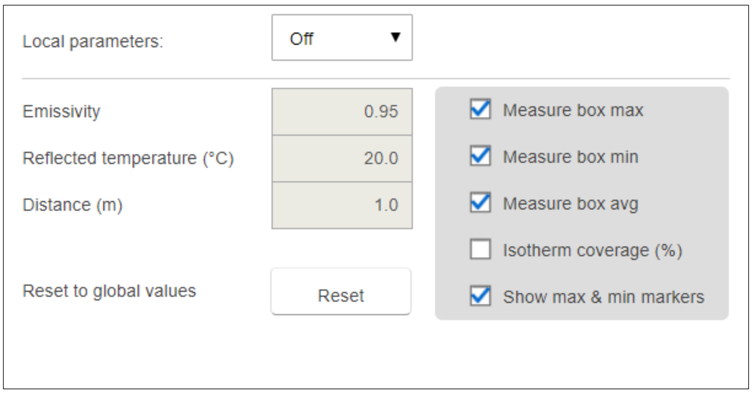
To move the box, click inside the box in the image and drag the box to the desired location.

To resize the box, click the border of the box in the image and drag the border to the desired size.

To configure settings for the box, follow these steps:

1.  In the `Measurements & alarms` section, click the `Box` icon. This displays a dialog box where you can configure the settings.

2. To set local parameters for the area, such as *Emissivity*, *Reflected temperature (°C)* and *Distance (m),* select `On` in the drop-down menu. To learn more about *Object parameters*, refer to Chapter Object parameters on page 43.



   a. To select what measurements results to display, use the check boxes `Measure box max`, `Measure box min`, and `Measure box avg`.

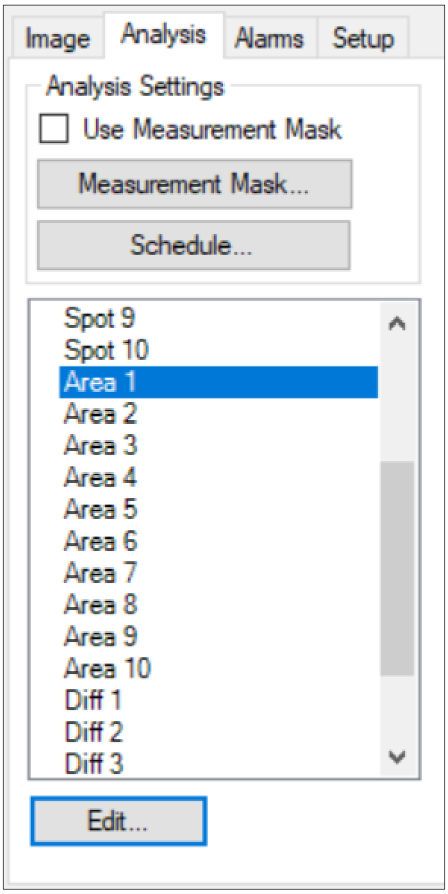
   b. To display how much of the box that is covered by an isotherm, tick the box `Isotherm coverage (%)`. This setting is only applicable if you have selected a color alarm (isotherm). Isotherms and Isotherm coverage is covered in detail further down this chapter on page 83.

   c. To show or hide the maximum and minimum markers (hot spot and cold spot) in the overlay graphics, select or deselect the check box `Show max & min markers`.

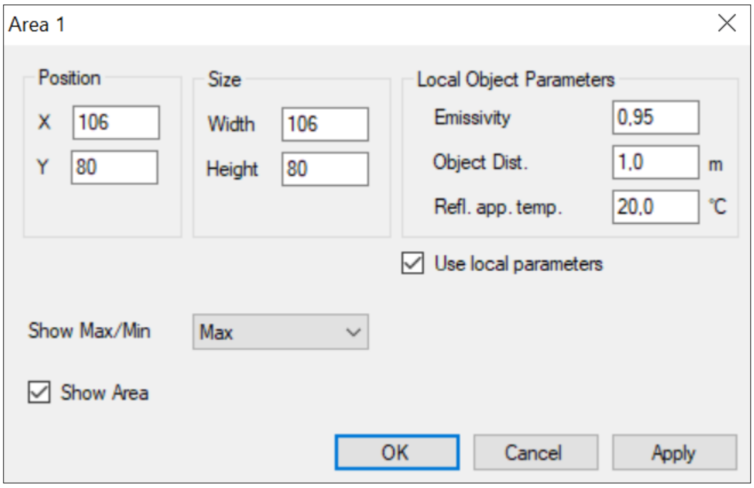
3. To associate an alarm with the box, see section Alarms further down in this chapter (page 97).

4. To remove the box, click the `Delete` icon  next to the tool in the `Measurements & alarms` menu.

FLIR A310



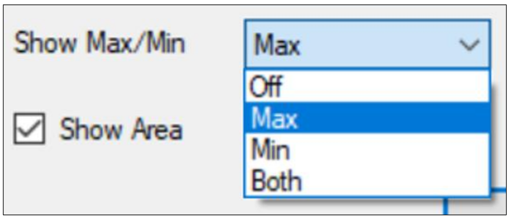
Mark `Area 1` by clicking on it, then click `Edit…` . This displays a dialog box where you can set the `X` and `Y` position of your area, measured in pixels from the top left corner of the IR image. The width and height of the area are also measured in pixels. If you don't know the `X` and `Y` position of your spot, just click `OK` and then you can drag the area to the desired location in the IR image. If you need to resize your area, mark it in the analysis pane and click `Edit…` .



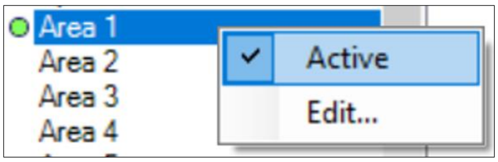
You can also choose to use *Local Object Parameters*, more on object parameters in chapter Object parameters (page 43).

Tick the `Show Area` box to see the spot in the interface.

In the drop-down menu by `Show Max/Min`, you can choose whether to show the maximum or minimum temperatures inside your area, or both.



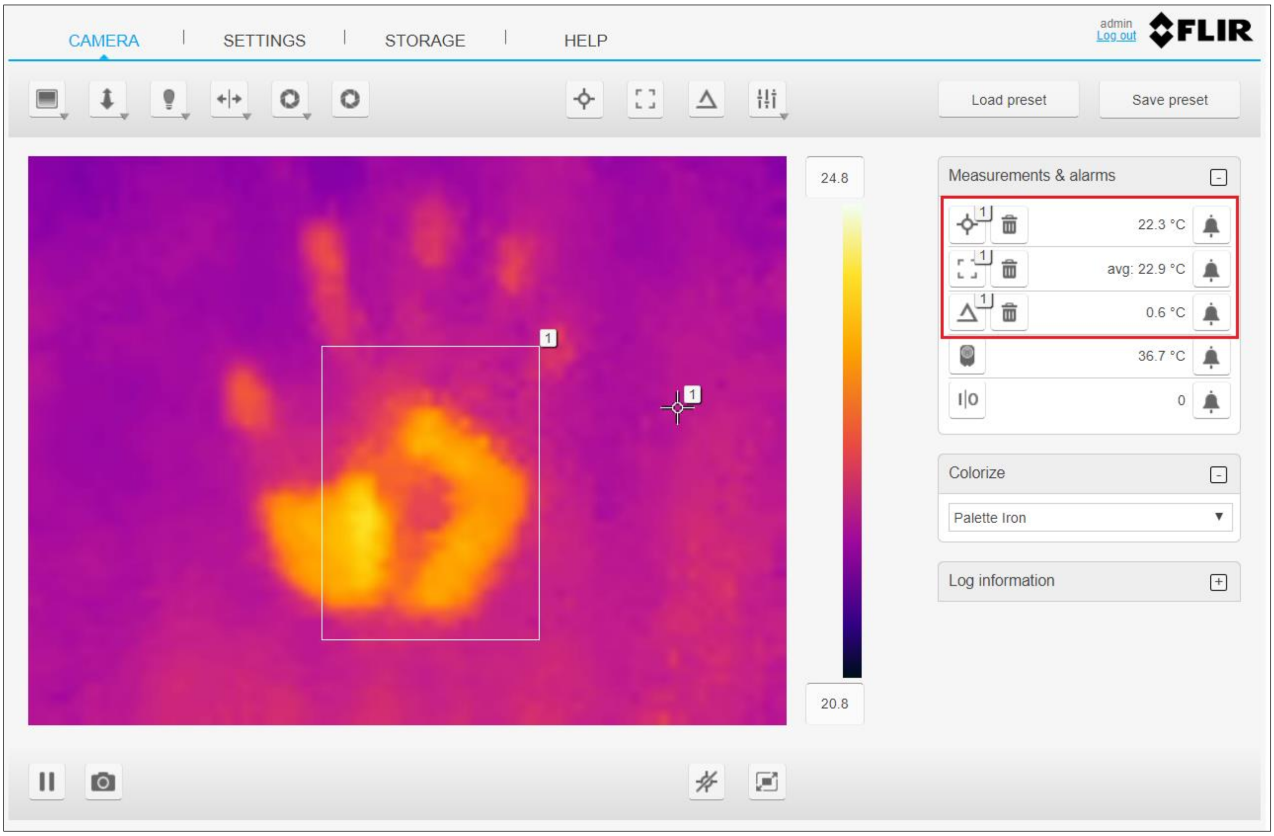
To remove an area – right-click the area in the analysis pane and click `Active`.



Delta Measurement

Another important quantity is the temperature *difference*. We use the Delta measurement for this. The Greek letter Delta (Δ) is often used to denote a difference. It is possible to set up a difference calculation between results from added spots and boxes, as well as a fixed temperature.

In my case, I'm measuring the difference between the average temperature in Box 1 and Spot 1. You can see that the difference is 0.6 °C, because 22.9 °C – 22.3 °C = 0.6 °C.

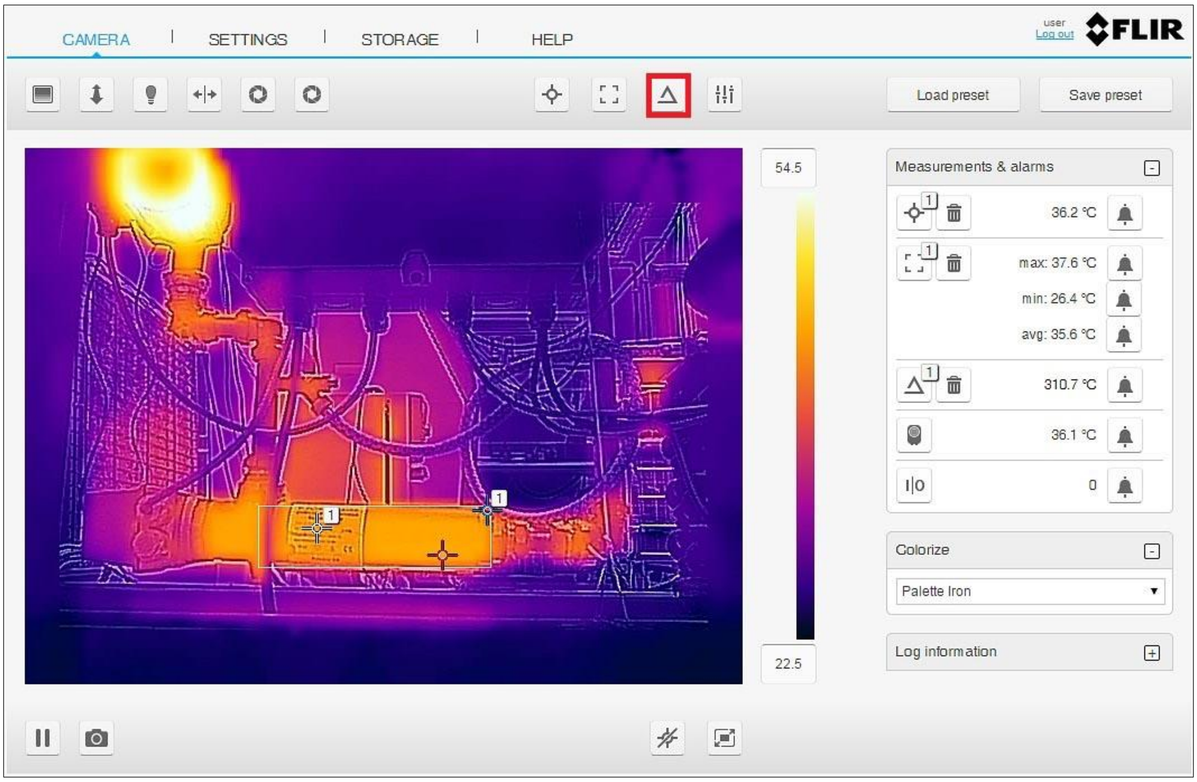


With the FLIR A310 you can set up to four difference calculations.

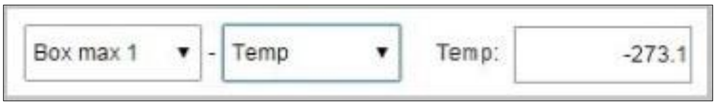
How do I set up a Delta measurement?

FLIR AX8


On the upper toolbar, click the `Delta measurement` icon. This displays the delta tool in the `Measurements & alarms` section.




In the `Measurements & alarms` section, click the `Delta` icon. This displays a dialog box where you can select the measurement tools you want to use in the difference calculation.



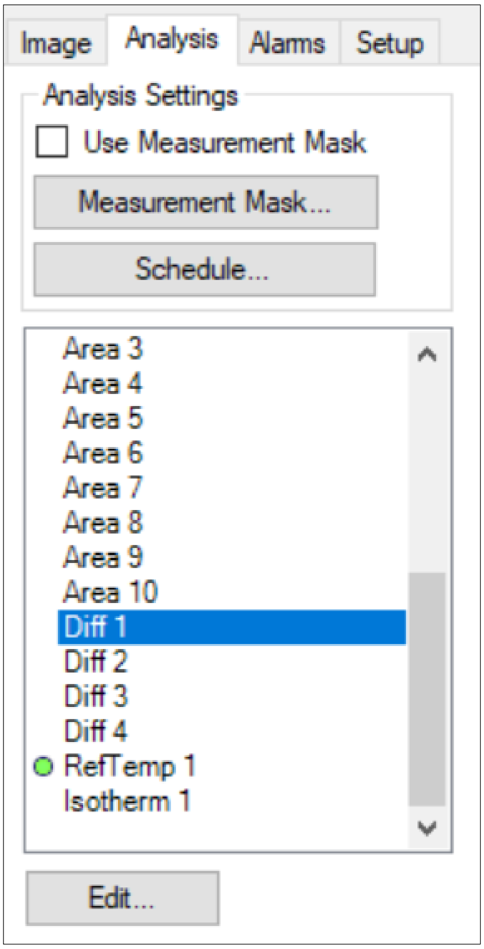
Select the first and second parameter from the list boxes. If you select the parameter `Temp`, also enter the fixed temperature in the `Temp` list box.

When completed, click anywhere outside the dialog box.

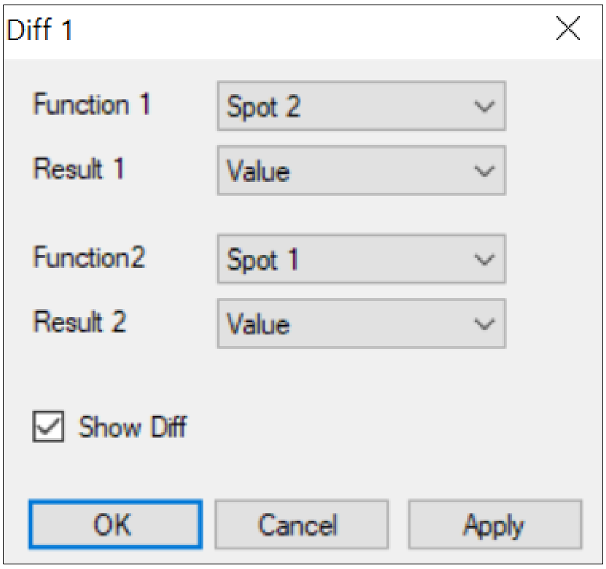
To associate an alarm with the Delta measurement, see section Alarms further down in this chapter.

 To remove the box, click the `Delete` icon next to the tool in the `Measurements & alarms` menu.

FLIR A310

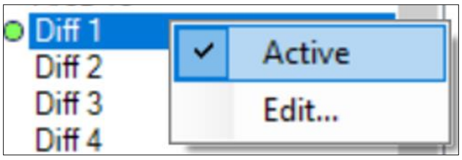


Mark `Diff 1` by clicking on it, then click `Edit…` . This displays a dialog box where you can set the two functions you wish to use.



Tick the `Show Diff` box to see the difference result beside the IR image.

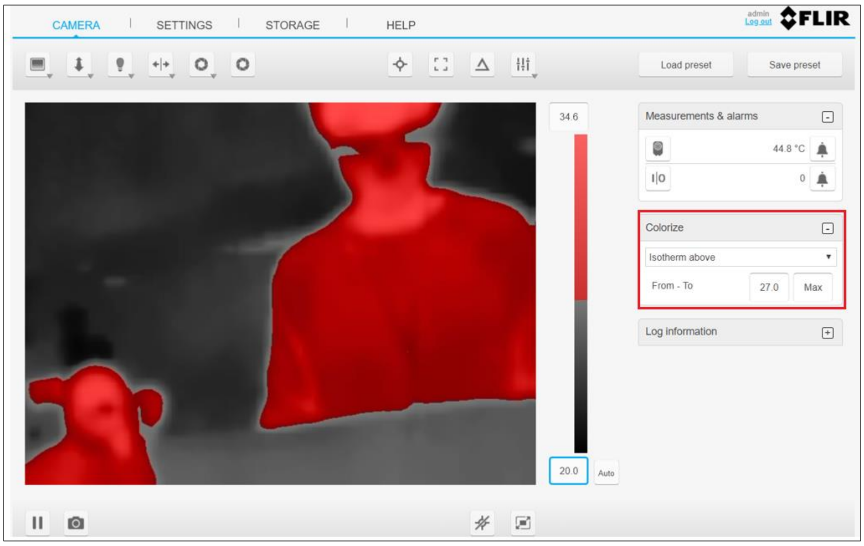
To remove the difference calculation – right-click `Diff 1` and click `Active`.



Isotherms

Sometimes we are interested in a certain temperature range when monitoring with an IR camera. Perhaps we want to find the coolest or hottest parts of the image. Or find all parts of an image that are within a temperature interval. To do this, we use the *Isotherm*, also known as the *Color alarm*. The Isotherm applies a contrasting color to all parts of the image that fulfill a specific temperature condition (above, below, or within temperature levels). Isotherms are great for displaying anomalies in an infrared image.

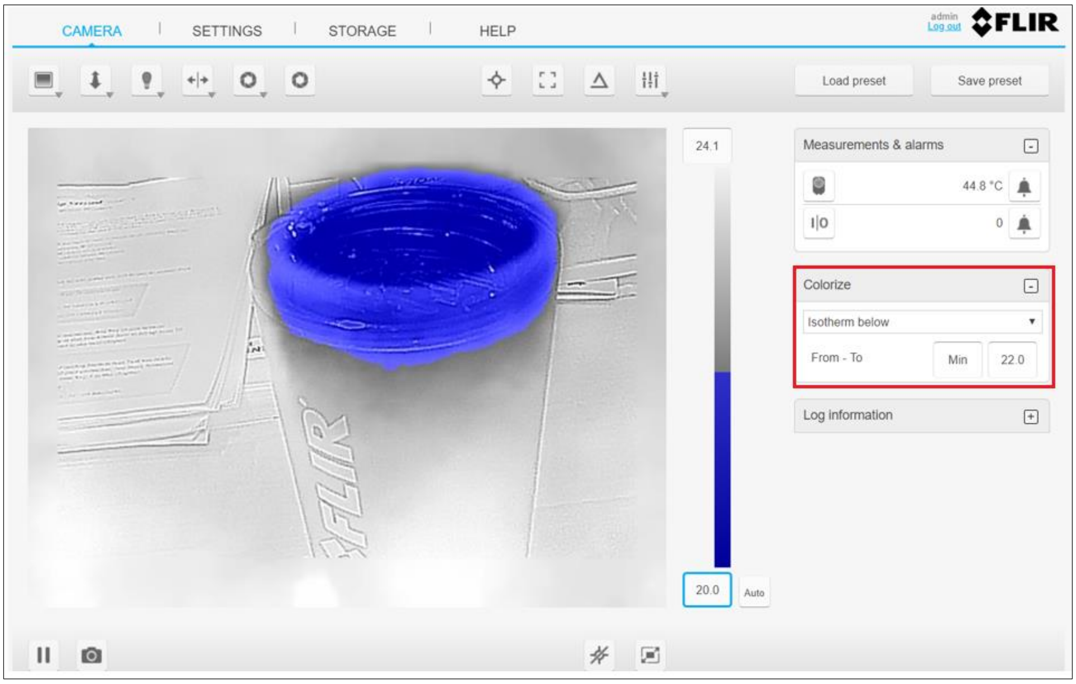
With the FLIR AX8, you find the isotherm function in the Palettes drop-down menu. Here are some examples:



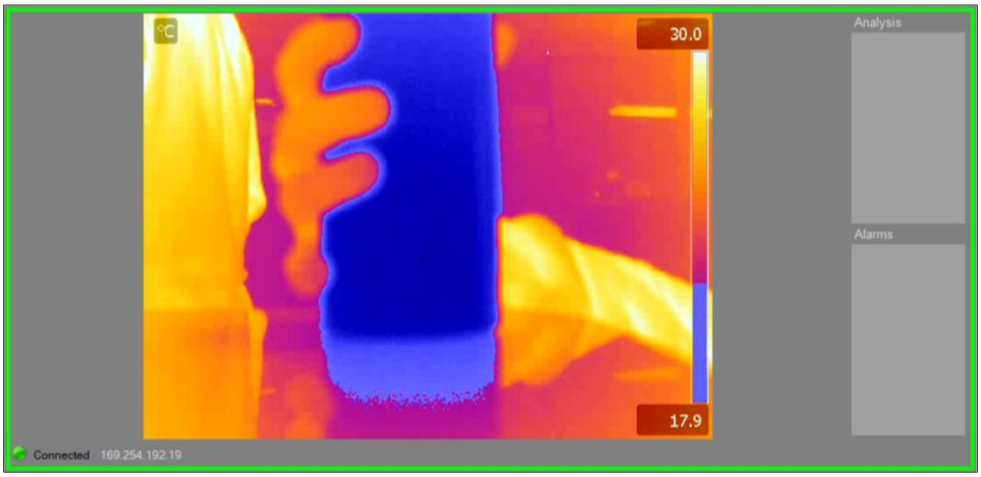
Isotherm above – In the image to the left, I've set the isotherm to show all pixels with apparent temperature *above* 27 °C, these are shown in red.

The same settings with the FLIR A310 – isotherm above 27 °C – yield the image below in IR Monitor.

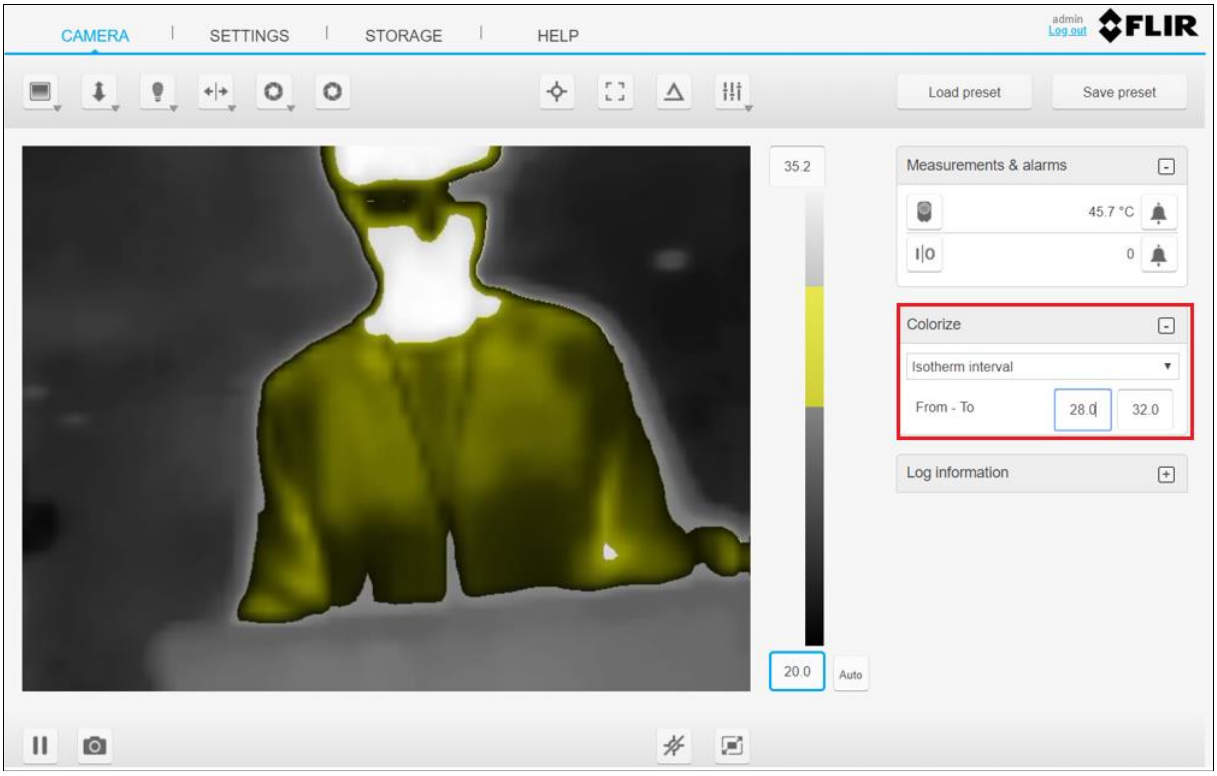
Isotherm below – In the image below, the isotherm shows everything that's colder than 22 °C with a blue color. In this case, it's the top of my cup.



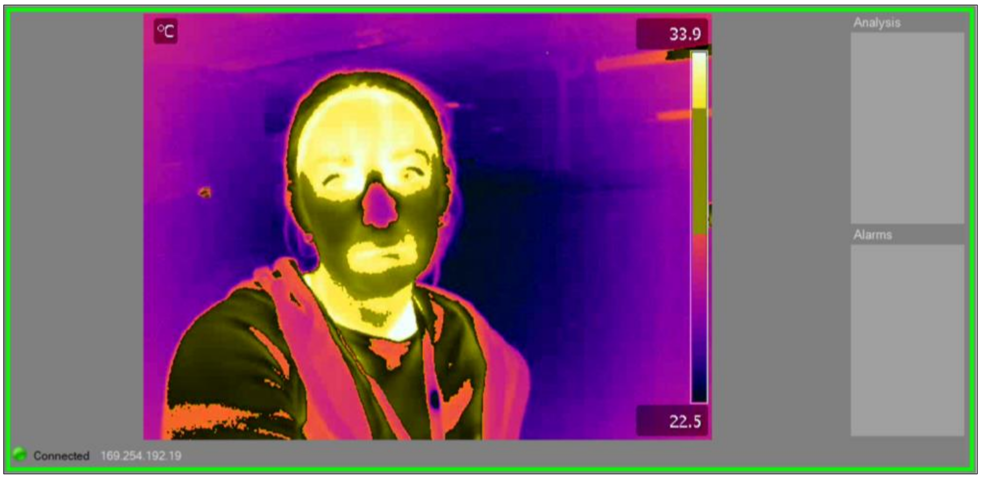
The same settings with the FLIR A310 – isotherm below 22 °C – yield the image below in IR Monitor.



Isotherm interval – In the image below, isotherm displays everything between 28 °C and 32 °C in yellow.



The same settings with the FLIR A310 – isotherm interval between 28 °C and 32 °C – yield the image below in IR Monitor.

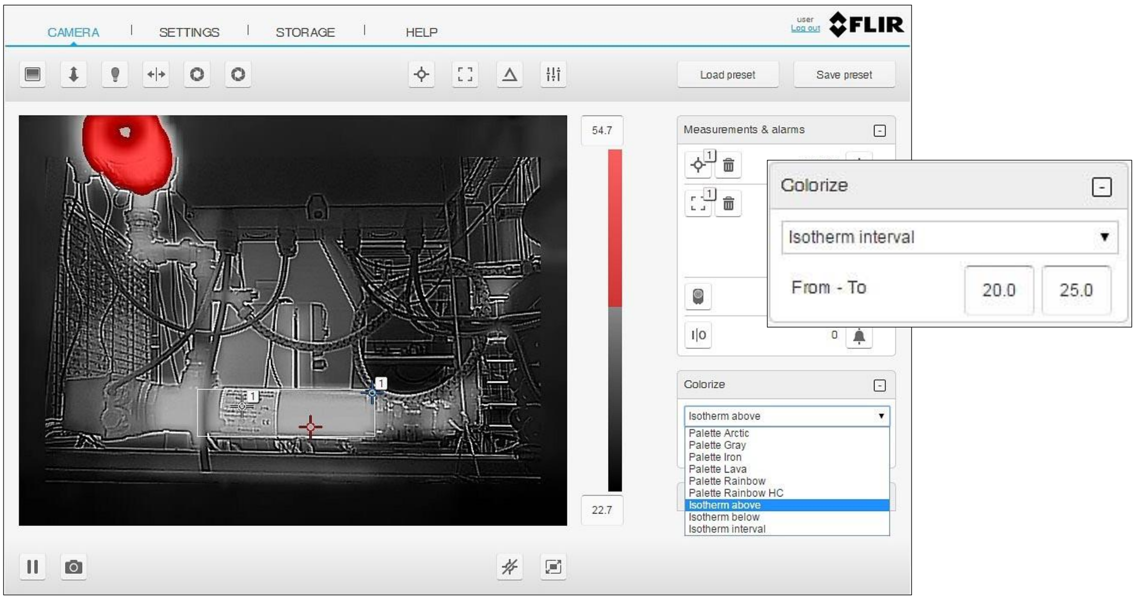


How do I set up an isotherm measurement?

FLIR AX8

In the `Colorize` list box, select one of the isotherms:

- Isotherm above
- Isotherm below
- Isotherm interval



When an isotherm is selected, the threshold temperature(s) are displayed in the `Colorize` section.

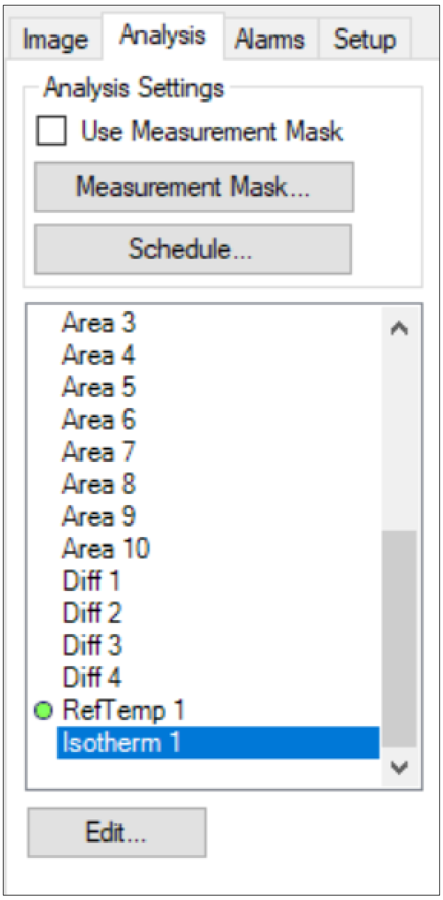
To change the threshold temperature, do the following:

- For the *Isotherm above*, enter the threshold temperature in the `From` text box.

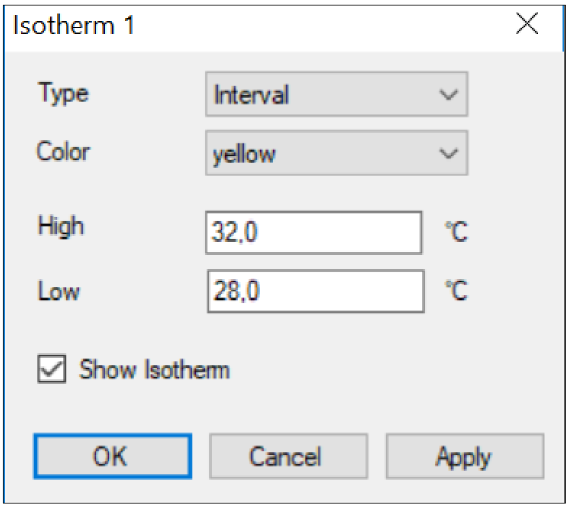
- For the *Isotherm below*, enter the threshold temperature in the `To` text box.

- For the *Isotherm interval*, enter the threshold temperatures in the `From` and `To` text boxes.

FLIR A310



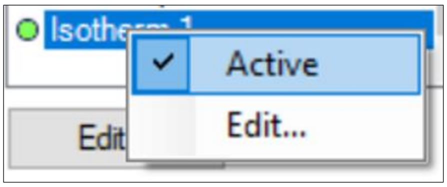
Mark `Isotherm 1` by clicking on it, then click `Edit…` . This displays a dialog box where you can choose which isotherm you want to use: *Above*, *below* or *interval*.



You can then choose the color of your isotherm, the above or below temperature, or the temperature interval.

Tick the `Show Isotherm` box to see the isotherm in the IR image.

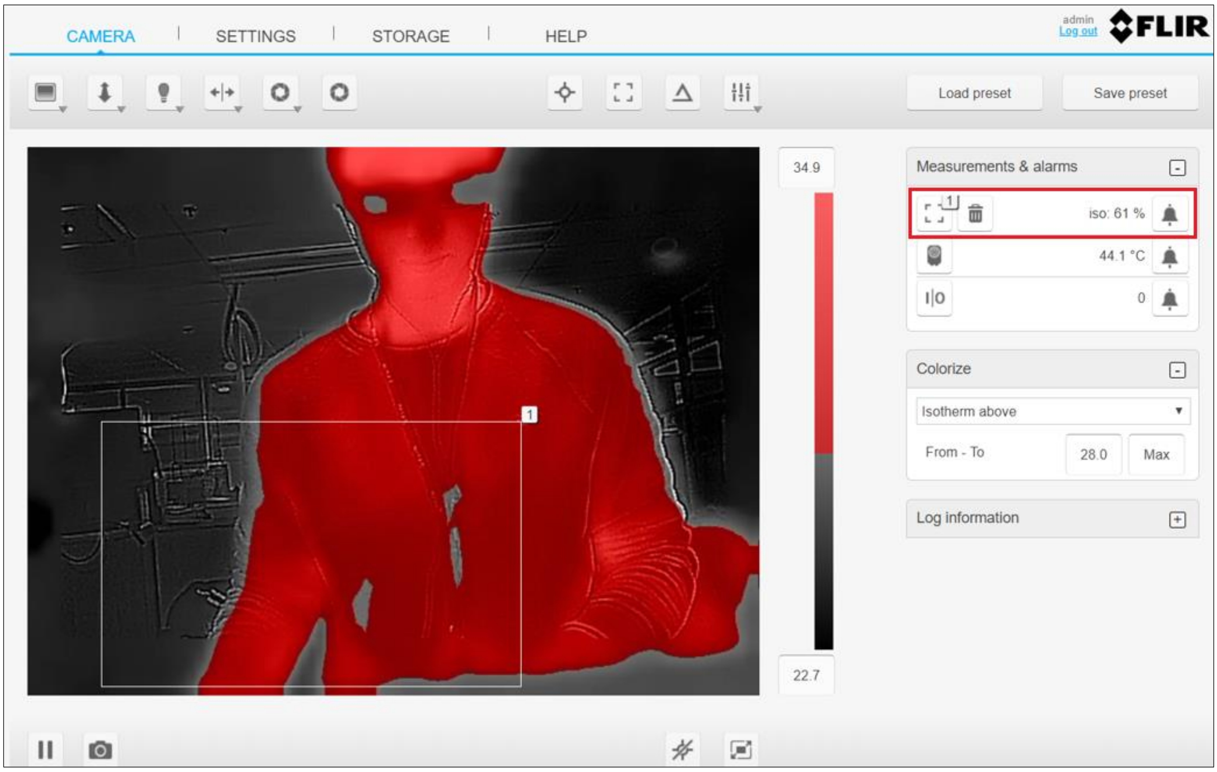
To remove the isotherm – right-click `Isotherm 1` and click `Active`.



Iso-coverage

It might not only be interesting to see what parts of a scene fulfills a certain temperature condition using the isotherm. You might want to quantify this coverage to a *percentage*. To do this, we use the *Isotherm coverage* feature, or *Iso-coverage* for short.

When you set up a *Box measurement* in the FLIR AX8 web interface, you may choose to show the Isotherm coverage (%). This percentage shows *how much of the box* (not the whole picture) fulfills the Isotherm condition. In my case, my Box 1 has an Iso-coverage of 61 %. It means that 61 % of *the area in Box 1* has a temperature higher than 28 °C. If you wish to know the iso-coverage of the whole picture, resize the Box so that it covers the whole monitor display.



How do I set up an iso-coverage measurement?

FLIR AX8

In the `Colorize` list box, select one of the isotherms:

- Isotherm above
- Isotherm below
- Isotherm interval

When an isotherm selected, the threshold temperature(s) are displayed in the `Colorize` section.

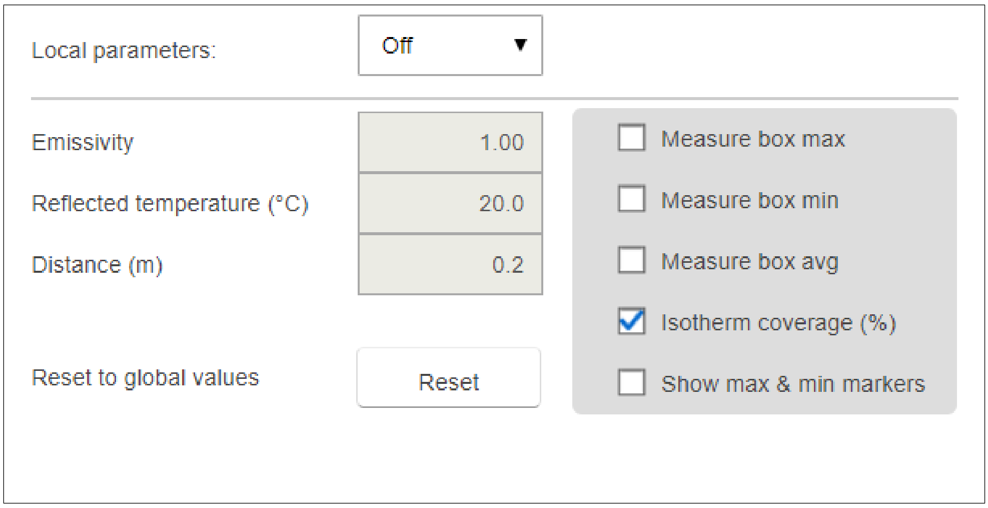
To change the threshold temperature, do the following:

- For the *Isotherm above*, enter the threshold temperature in the `From` text box.
- For the *Isotherm below*, enter the threshold temperature in the `To` text box.
- For the *Isotherm interval*, enter the threshold temperatures in the `From` and `To` text boxes.

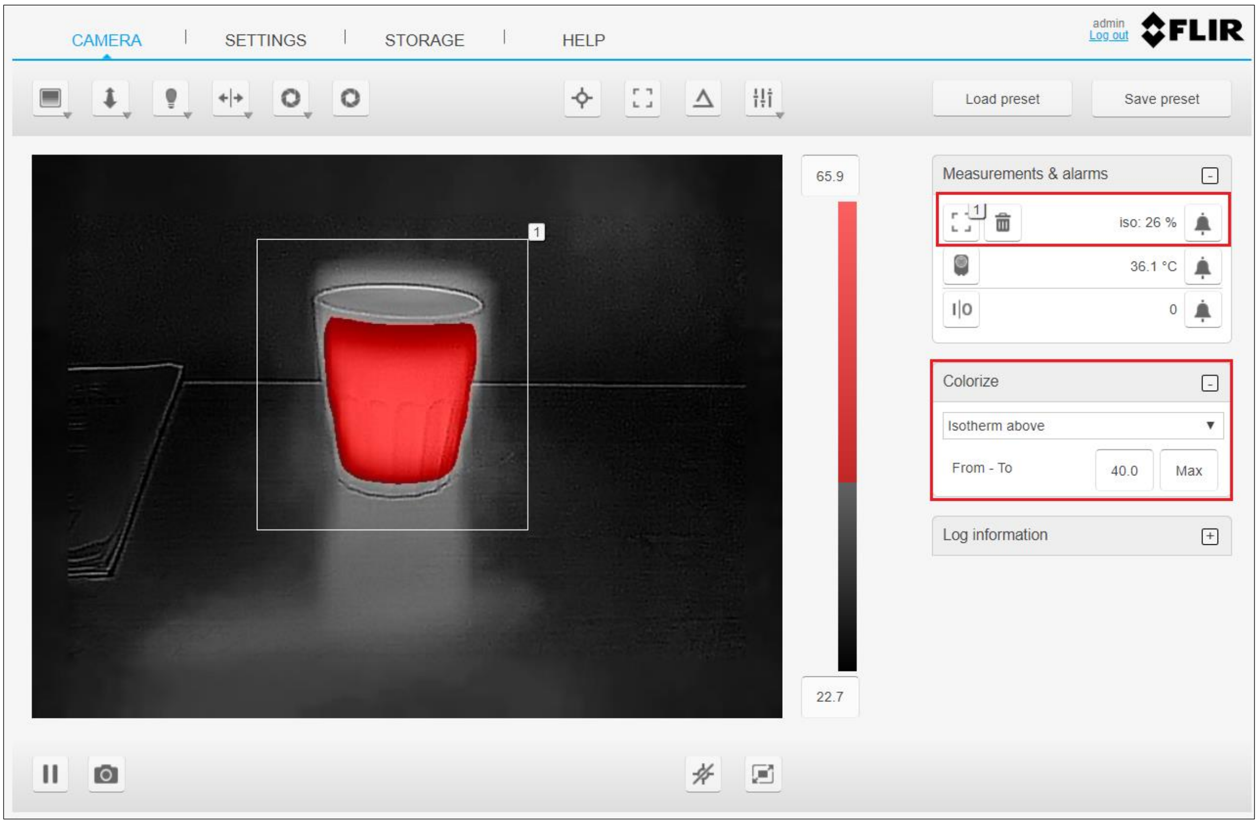
Create a *Box measurement*, by clicking the icon . See the Box measurement section for more information about this measurement.

In the `Measurement & alarms` section, click the `Box` icon . This displays a dialog box where you can configure the settings.

Tick the box `Isotherm coverage (%).`



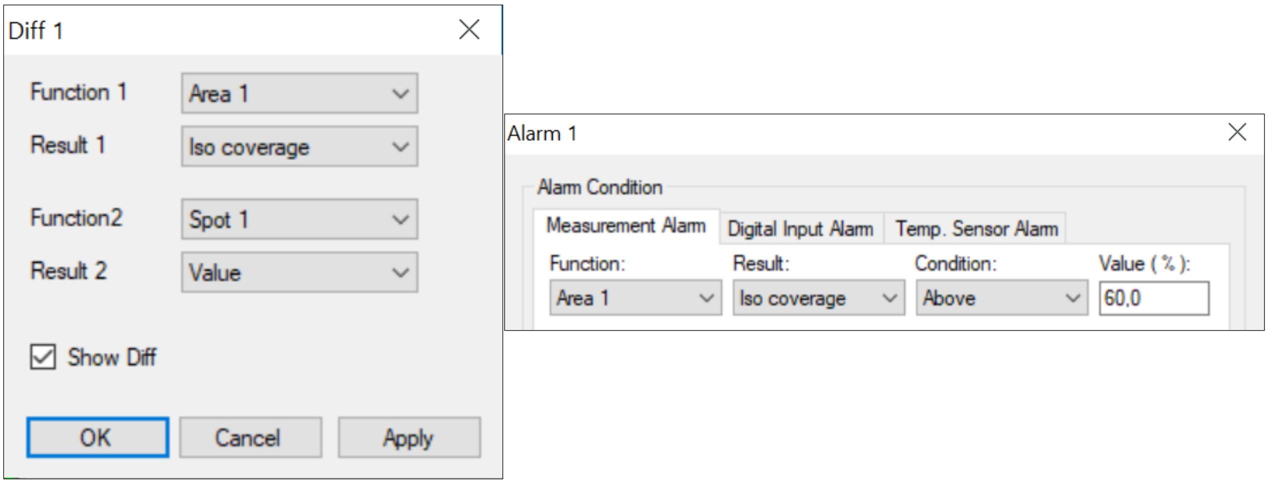
The iso-coverage percentage is now displayed in the `Measurements & alarms` section.



To associate an alarm with the iso-coverage, see section Alarms further down in this chapter.

FLIR A310

With the FLIR A310 you can use the iso-coverage function in a difference equation or as an alarm condition.



Iso-coverage in IR Monitor in a difference equation and as an alarm condition.

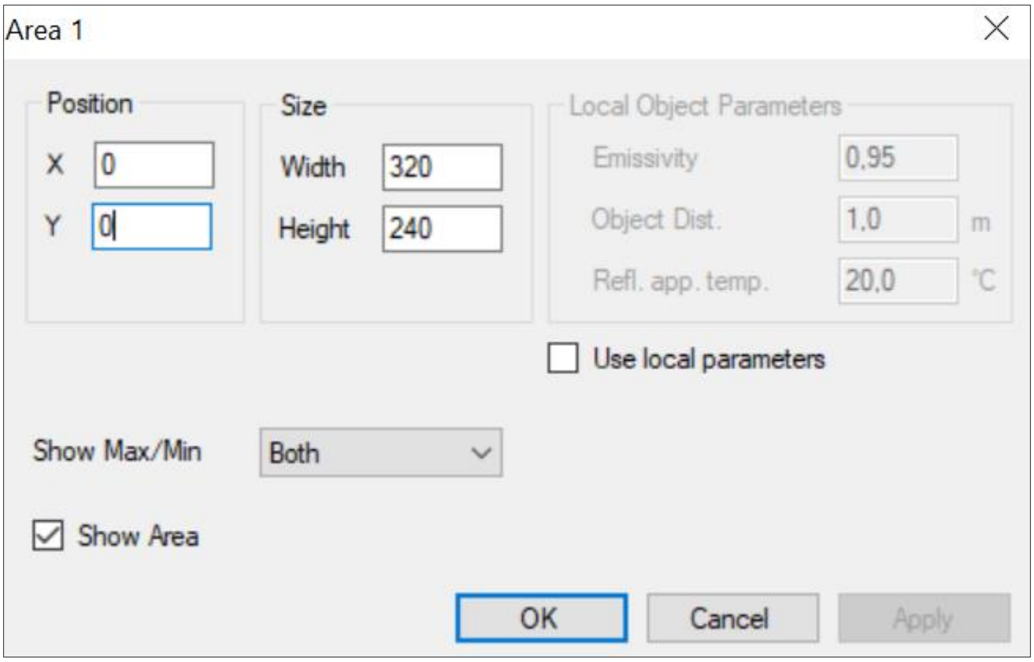
Mask

The measurement mask is analytics feature for the FLIR A310 in IR Monitor. It allows you to create a free-form area, instead of a box or a spot.

How do I set up a mask?

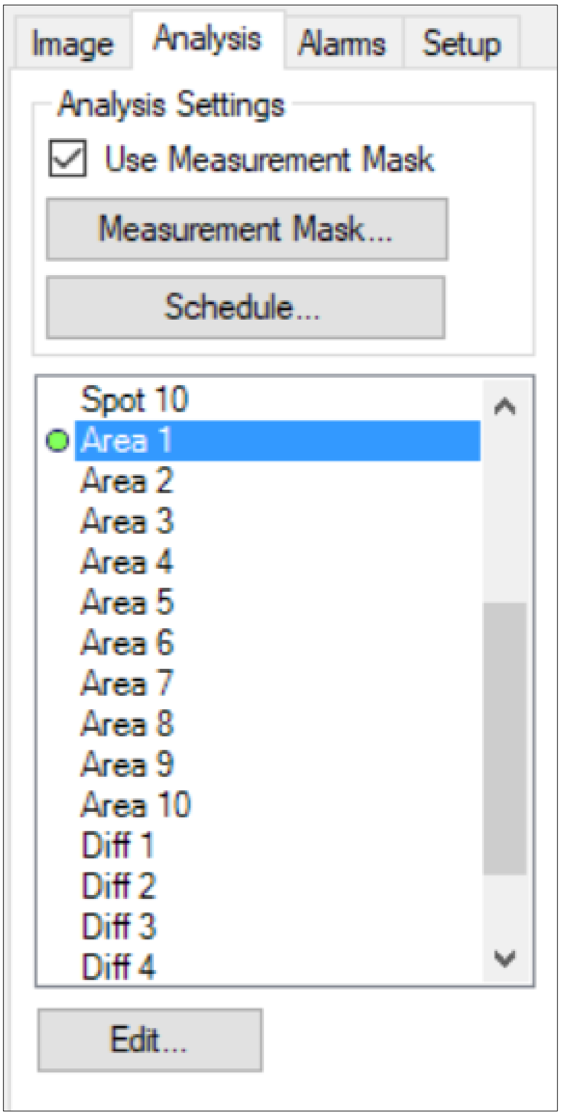
FLIR A310

First, you need to set up an area – or box – inside which we are going to place our mask. When working with the measurement mask, we can cover the whole scene with an area – the measurements will only be in the free-form mask. The width and height of the area needs to be in pixels, so if you don't know the number of pixels you have in your IR camera – you can check the user manual. Look under **Technical data >> IR resolution**. The FLIR A310 has an IR resolution of 320 x 240, so I entered the values displayed in the image below to cover the whole scene.

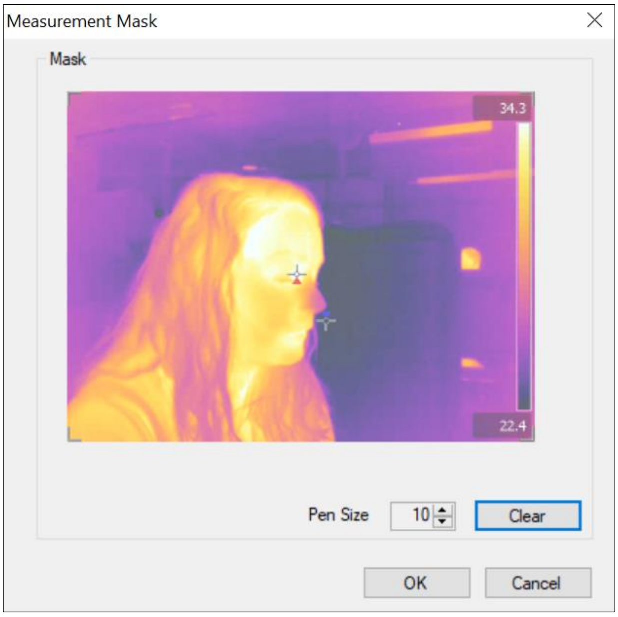


The `X` and `Y` positions are set to `0`, since I want the area to start from the top left corner of the scene. The `width` and `height` size are set to the number of pixels of my FLIR A310.

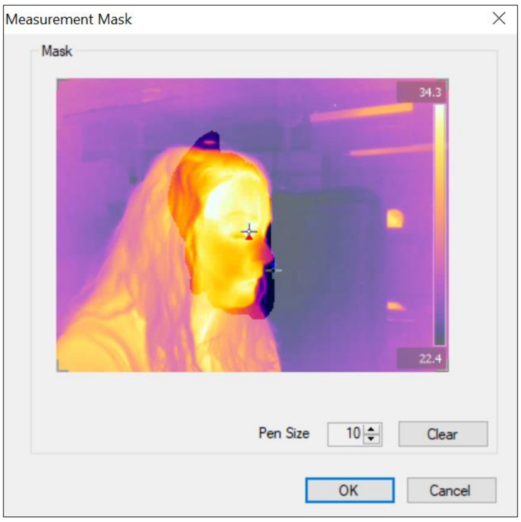
Now to the measurement mask.

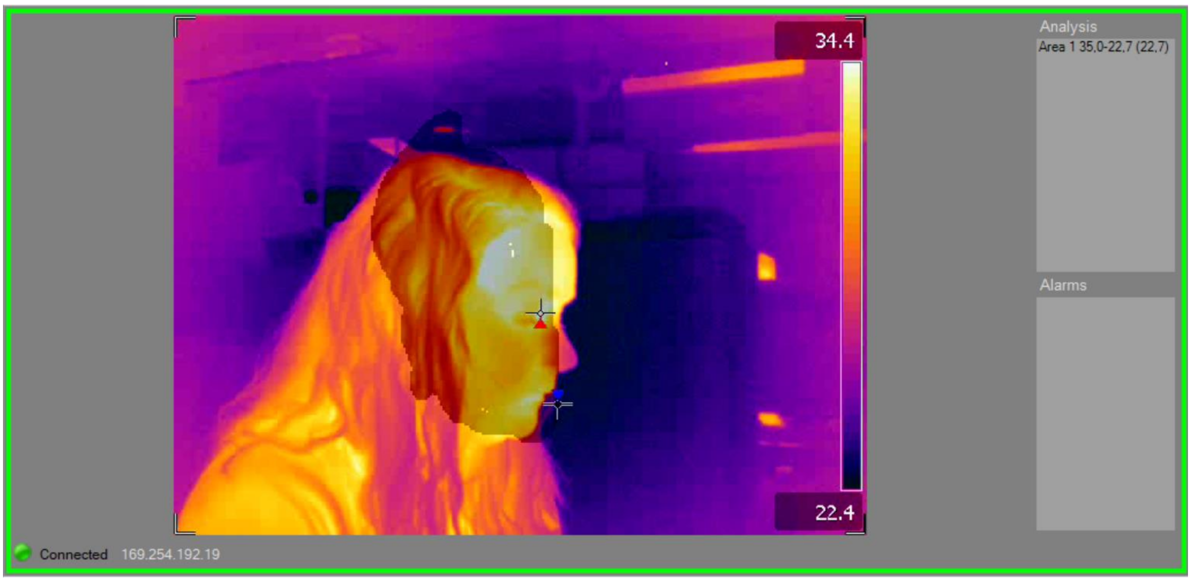


Tick the box `Use Measurement Mask` and click `Measurement Mask...` . This displays a dialog box, where you can choose the pen size. The mask is then drawn by using the mouse marker in the IR image in the dialogue box.

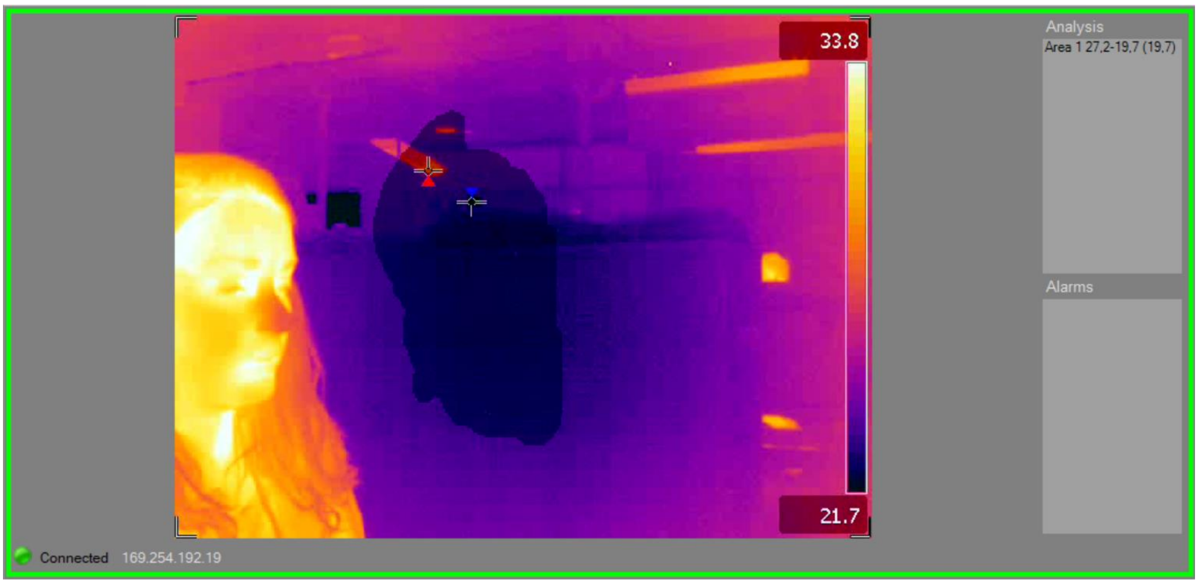


When you are done marking your measurement mask, click `OK`. If you want to remove your measurement mask, click `Clear`.

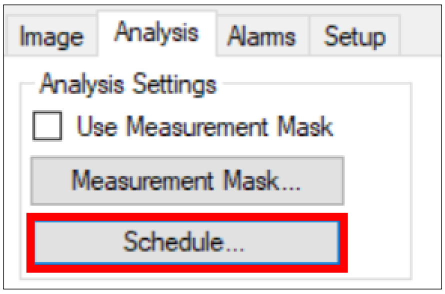




The maximum and minimum temperatures shown under the Analysis pane will now be restricted to the mask I've drawn – and ignore the rest of the area. When I move out of the measurement mask, the maximum and minimum markers will find the maximum and minimum temperatures inside the mask – and ignore me.

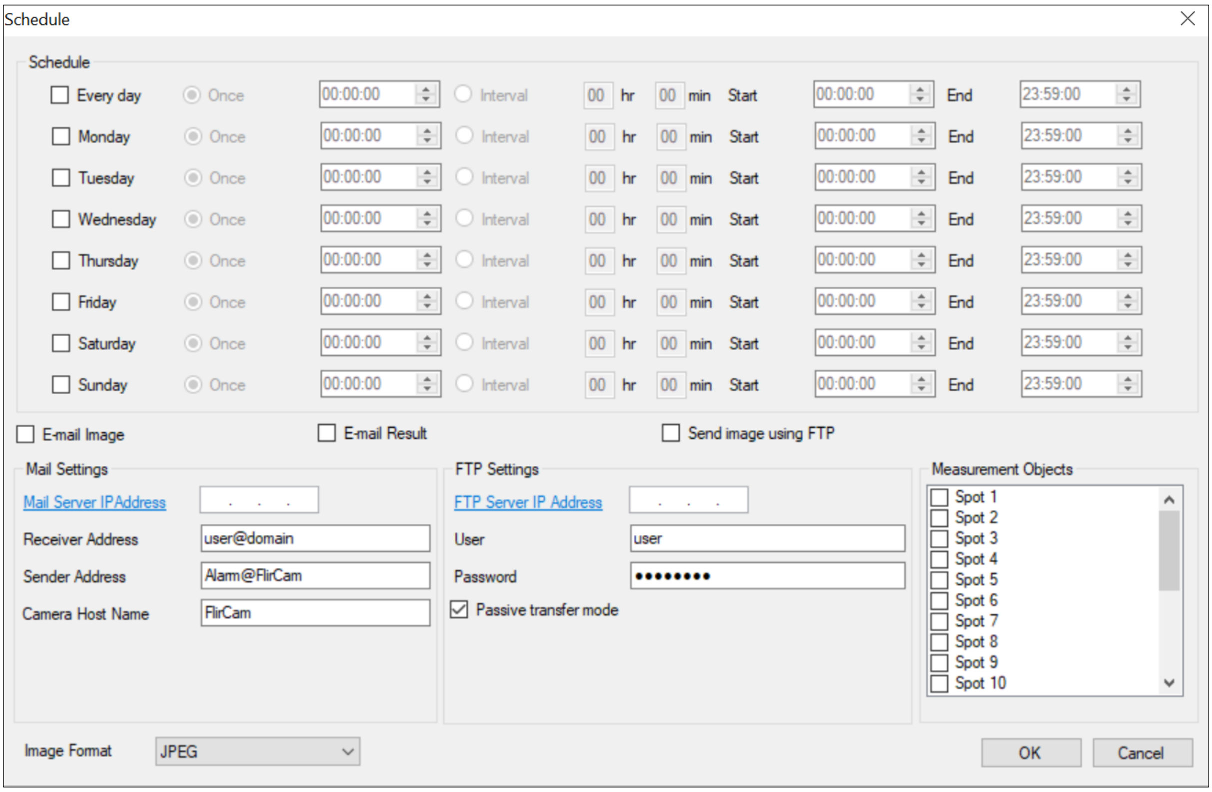


Schedule



IR Monitor allows you to schedule periodically captions to be stored and sent to you. Clicking the `Schedule…` button will display a dialog box, where you can set the day and time you want captions to be sent. You can also set what you wish to be sent to you: an image or an E-mail result.

The recipient can be either an E-mail or an FTP server. Setting up a local E-mail and FTP server are covered in the Protocols chapter on pages 177 and 167 respectively.



For more information about the schedule feature of IR Monitor, refer to the user manual of IR Monitor. It can be downloaded from FLIRs web site, or by typing the following link in your web browser

`https://flir.custhelp.com/app/account/fl_download_manuals`

and enter `IR monitor` in the search field.

SUMMARY

In this chapter we have discussed different analysis functions. I will list the key concepts. Read them through and reflect on what you have read.

> The FLIR AX8 have five analytics features:
>
> Spot: displays the temperature of one spot in the image.
>
> Box: displays the temperature of a set area.
>
> Delta: displays the temperature difference between two measurements.
>
> Isotherm: shows all pixels that satisfies a certain temperature condition.
>
> Iso-coverage: Displays the percentage in a box that the isotherm covers.
>
> The FLIR A310 also have the spot, box, difference and isotherm analytics and these additional two:
>
> Mask: displays the temperature of a free-form area
>
> Schedule: sends captions or videos at set time

Alarms

The analysis features would lose their full potential if it weren't for the possibility to set alarms. It would be very taxing and inefficient if we always had to monitor the situation ourselves. Instead we let the camera and computer do the job. In this section, I will go through how you set up an alarm for condition monitoring purposes.

Objectives

When you have worked through this unit, my aim is that you will be able to answer these questions

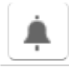
> How do I set up an alarm?
>
> What is Hysteresis?
>
> How do hysteresis and threshold time help prevent false alarms?

For the Alarms section of this chapter, we'll cover the following:

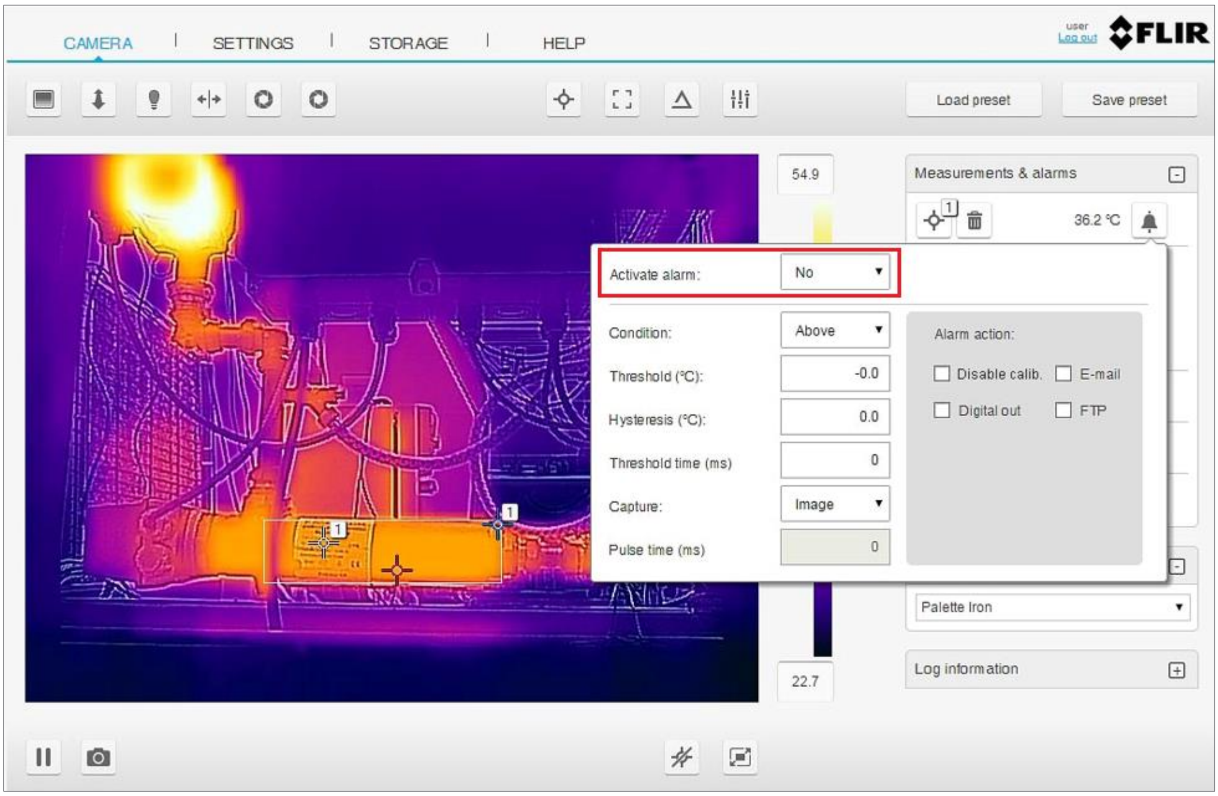
> How do I set up an alarm?
>
> Alarm options
>
> Hands-on Exercise
>
> - Threshold time and Hysteresis

How do I set up an alarm?

FLIR AX8

All features presented in the Analytics section have the option to enable an alarm. To access the alarm tab of a measurement, click on the  icon. In my example, I have accessed the alarm window of a Spot.

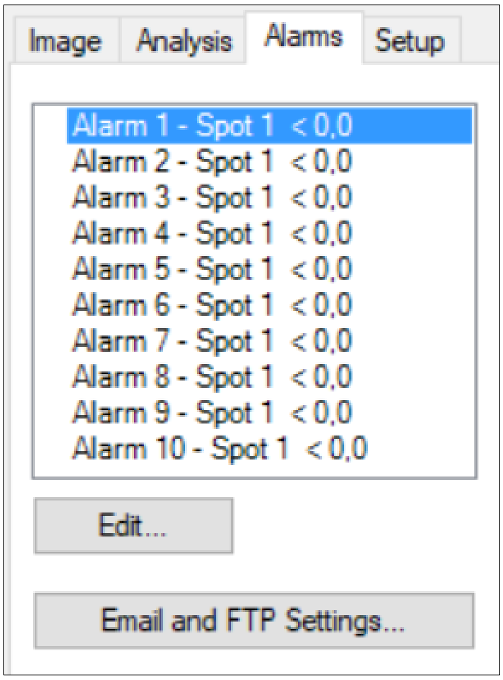
To activate an alarm, select `Yes` in the dropdown menu to the right of `Activate alarm`.



When an alarm is activated, the *Alarm settings icon* is marked with a blue frame: 

When an alarm is triggered, the *Alarm settings icon* is marked with a red frame: 

FLIR A310

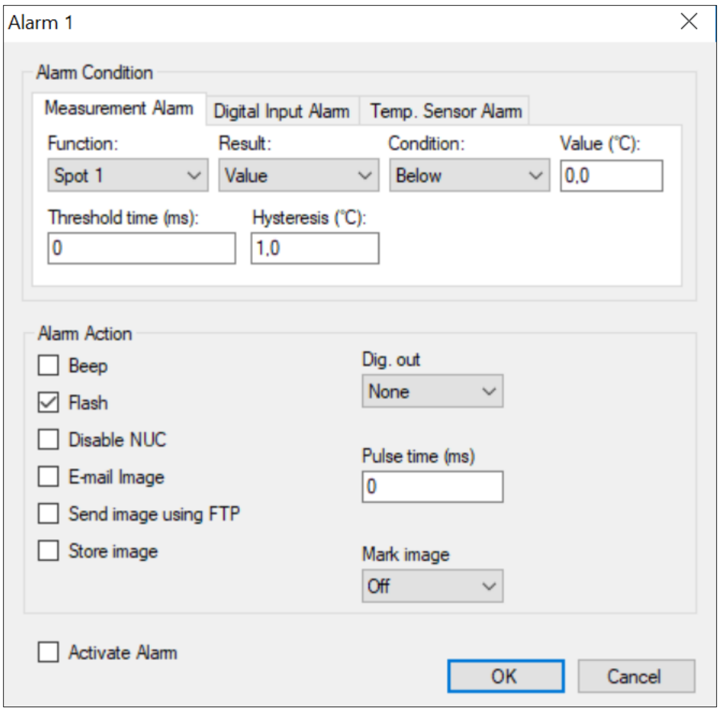


Under the `Alarms` tab you can set up to ten alarms in IR Monitor. Double-click an alarm or mark it and then click `Edit…` to edit the alarm.

This will display a dialog box where you can choose the *Alarm Condition* and the *Alarm Action*.

The options for alarm condition are:

- `Measurement Alarm`: An alarm will trigger when an analysis function reaches a set value. This can be a temperature measurement in a spot, for instance.
- `Digital Input Alarm`: An alarm will trigger when the digital input reaches a set value (for more on digital input, see chapter Input on page 204).
- `Temp. Sensor Alarm`: an alarm will trigger when the temperature of your IR camera reaches a set value.



Now we'll go through the different options to manage when setting up an alarm. These include:

- Condition
- Threshold
- Delay
- Hysteresis
- Threshold time (Delay)
- Capture
- Pulse time
- Alarm action

Alarm options

Condition

In the Condition field, you can choose if your condition should be above or below. You might wish an alarm to trigger when the temperature goes *above* 100 °C or perhaps *below* 5 °C.

Threshold

Here you choose which temperature should be the trigger limit for the alarm.

Note

- The threshold is often but *not always* a temperature, it depends on what type of alarm you wish to set. If you set an iso-coverage alarm, then the threshold will be measured in percent (%).

A big issue when dealing with alarms is the risk of *false alarms*. Two functions used to better manage alarms and minimize the amount of false alarms are *Hysteresis* and *Threshold time*.

Hysteresis

An alarm can either be triggered *on* or *off*. Sometimes this switching between the on and off states may occur very quickly. We then utilize *hysteresis* when we wish to prevent this unwanted rapid switching. To illustrate the concept of Hysteresis, let me start with a non-IR example. It involves the idea of a *control system*. Image you have a floor heating system, and you wish to keep this floor at a temperature of 20 °C. To do this, you have a heating mechanism which can either be turned on or off.

You begin heating the floor.

The temperature reaches 20 °C, and the heating mechanism turns off.

But because the heating mechanism is off, the floor gets colder and drops below 20 °C.

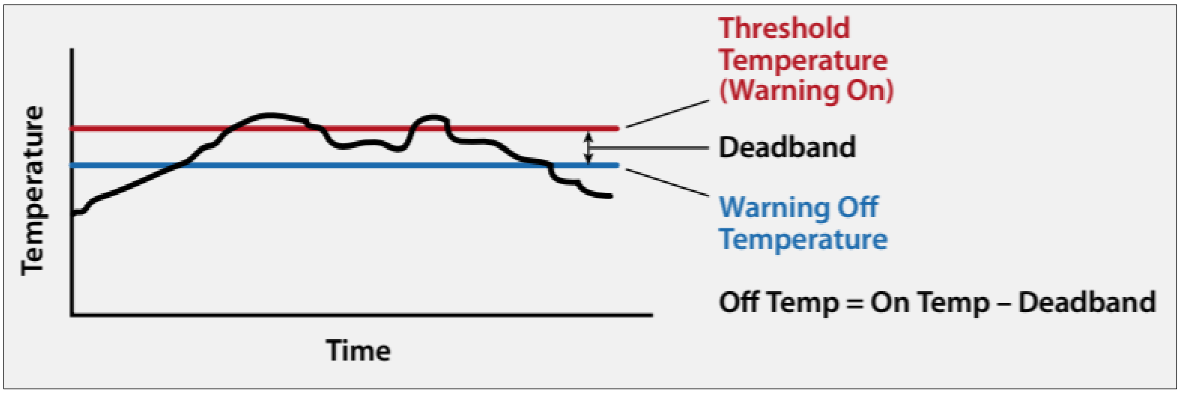
Now the heating system is turned on again and starts – yet again – heating the floor.

The heating continues just to the point where the floor temperature reaches 20 °C. The heating is then turned off.

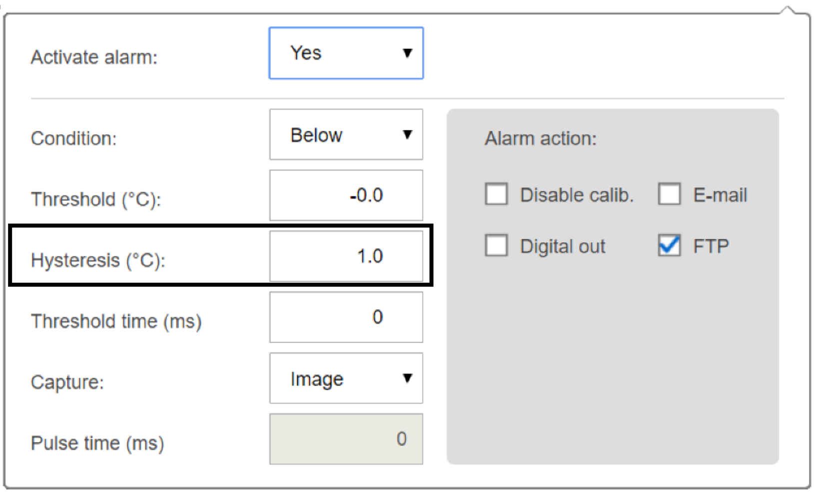
This process will repeat, and the temperature will oscillate around 20 °C. The goal of keeping the floor at a temperature around 20 °C is fulfilled, but the heating mechanism will be rapidly switching on and off, perhaps as much as 50 times per second.

Perhaps you see that this rapid switching on and off becomes a very inefficient method. To prevent this – we add a threshold below 20 °C – such that the temperature must fall below the threshold for the heating to begin again. This threshold is called a *hysteresis*, also known as *deadband*.

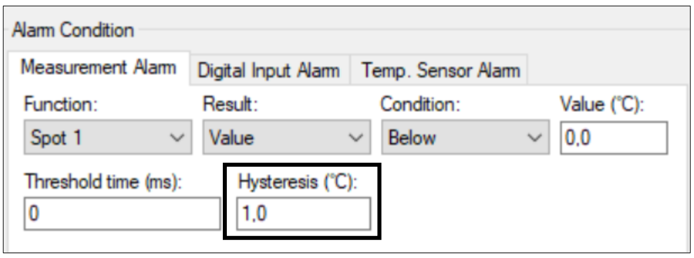
In our example with the heating floor system, let's say that we set a hysteresis of 1 °C. This means that the system will begin heating the floor again only when the floor temperature drops below 19 °C, which is 1 °C below our desired 20 °C.



Now let's return to our IR camera. The same problem of rapid switching may occur if the alarm condition lies near the monitored temperature. The alarm could be triggered on and off many times every second. To prevent this unwanted rapid triggering, we add a hysteresis.



Hysteresis option in the web interface for the FLIR AX8



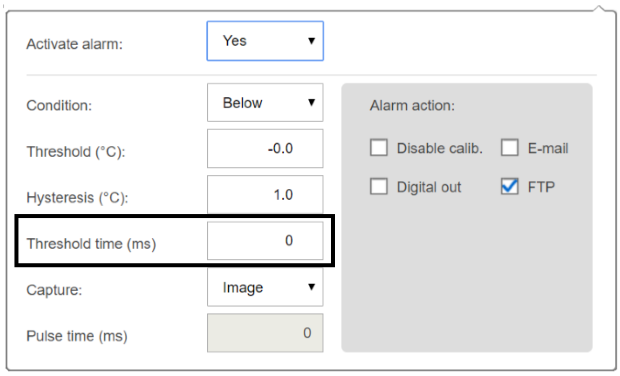
Hysteresis option in IR Monitor for the FLIR A310

Hysteresis is the interval in which the temperature value is allowed to vary without causing a change in the trigger. You can calculate the hysteresis, by using the formula below.

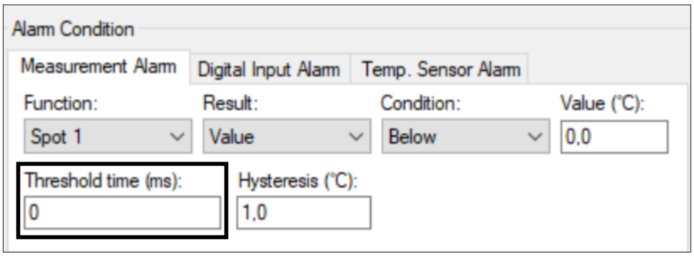
$$Hysteresis = Alarm\ trigger\ temperature - Alarm\ off\ temperature$$

Threshold time (Delay)

The threshold time is the minimum duration that must be matched or exceeded for the alarm to be triggered. This is a powerful tool for avoiding false alarms. The threshold time is set in milliseconds (ms). Remember that 1 second is 1000 milliseconds.

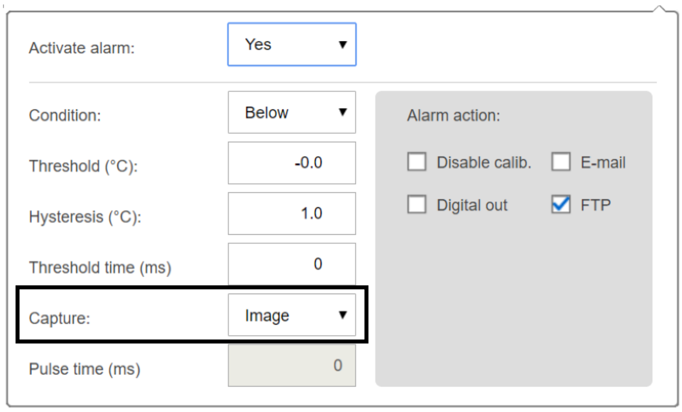


Threshold time option in the web interface for the FLIR AX8



Threshold time option in IR Monitor for the FLIR A310

Capture

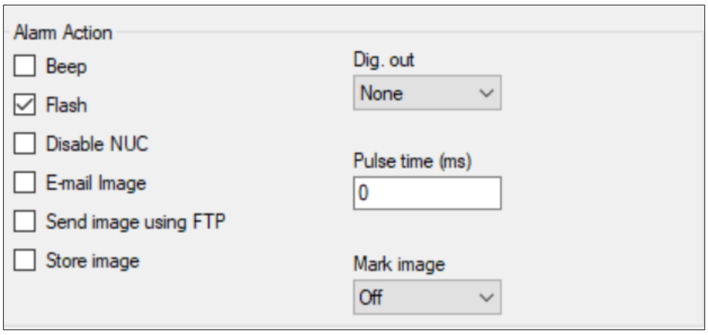


In the `Capture` list box, select if an image or a video sequence will be captured and saved when an alarm is triggered. You can view and manage the image or video under the `Storage` tab.

- Select `Image` to capture the image frame that triggered the alarm.
- Select `Video` to capture a 5 second video sequence when the alarm is triggered.
- Select `None` and no image/video will be captured.

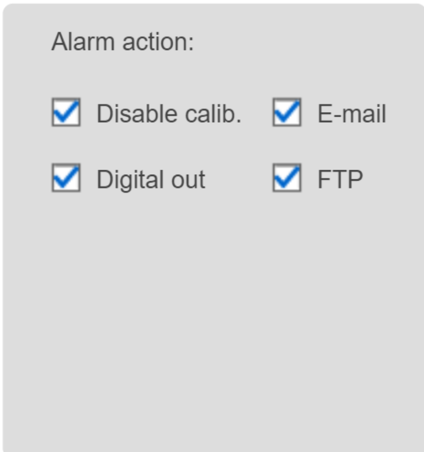
Alarm actions

Under `Alarm action`, use the check boxes to select which actions the camera will perform when an alarm is triggered:



Alarm Action in IR Monitor

- `Disable calib./Disable NUC`: Temporarily disables the periodic calibration (the NUC, see page 123) while the image/video is being captured.
- `E-mail/E-mail Image`: Automatically sends the captured image or video to the recipients defined in `Settings >> Alarm recipients/Alarms >> E-mail and FTP Settings…` .
- `Digital out/Dig. out`: Outputs a digital pulse (see page 198).
- `FTP/Send image using FTP`: Automatically sends the captured image or video to the FTP site defined in `Settings >> Alarm recipients/Alarms >> E-mail and FTP settings`.

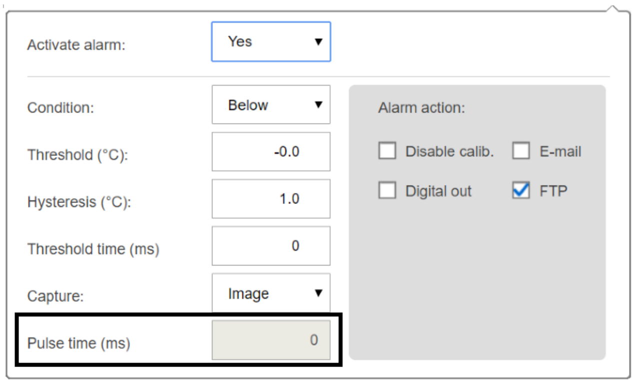


Alarm action in the web interface

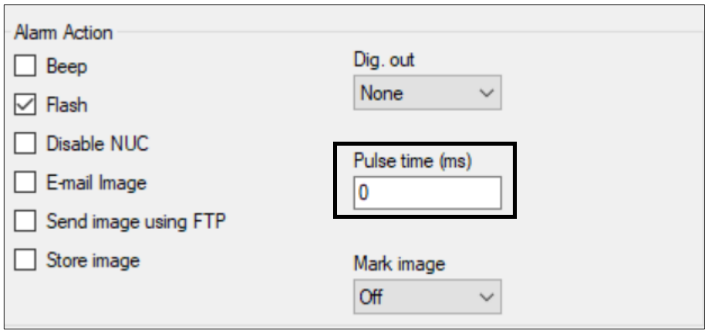
Pulse time

If you have selected the alarm action `Digital out`, enter the pulse length (in milliseconds) in the `Pulse time` text box.

For more information about *Digital inputs and outputs,* see chapter Input and output on page 198.



Pulse time option in the web interface



Pulse time option in IR Monitor

SUMMARY

In this chapter, we have discussed alarms and their settings. The features we have discussed is listed below. As you go through the list, I would advise you to remember and reflect on what you have read.

> Condition: determines when an alarm should be triggered. The condition can be set to above or below a set temperature.
>
> Threshold: determines which temperature or percentage that should be the trigger for the alarm.
>
> Hysteresis: determines the interval in which the temperature value is allowed to vary without causing a change in the trigger.
>
> Threshold time (delay): determines the minimum duration that must be matched or exceeded for the alarm to be triggered.
>
> Alarm actions: determines what action the IR camera will perform when an alarm is triggered.

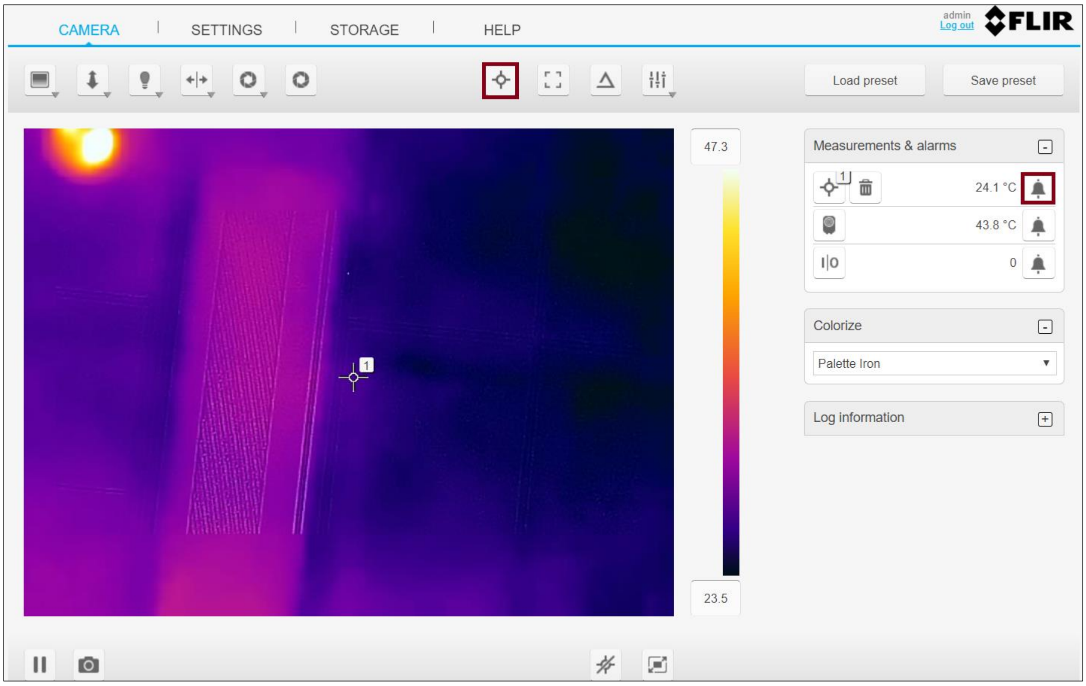
Hands-on Exercises

How to set up an alarm with threshold time and hysteresis

This exercise is meant to show you how to set up an alarm and to display the features of *hysteresis* and *threshold time*.

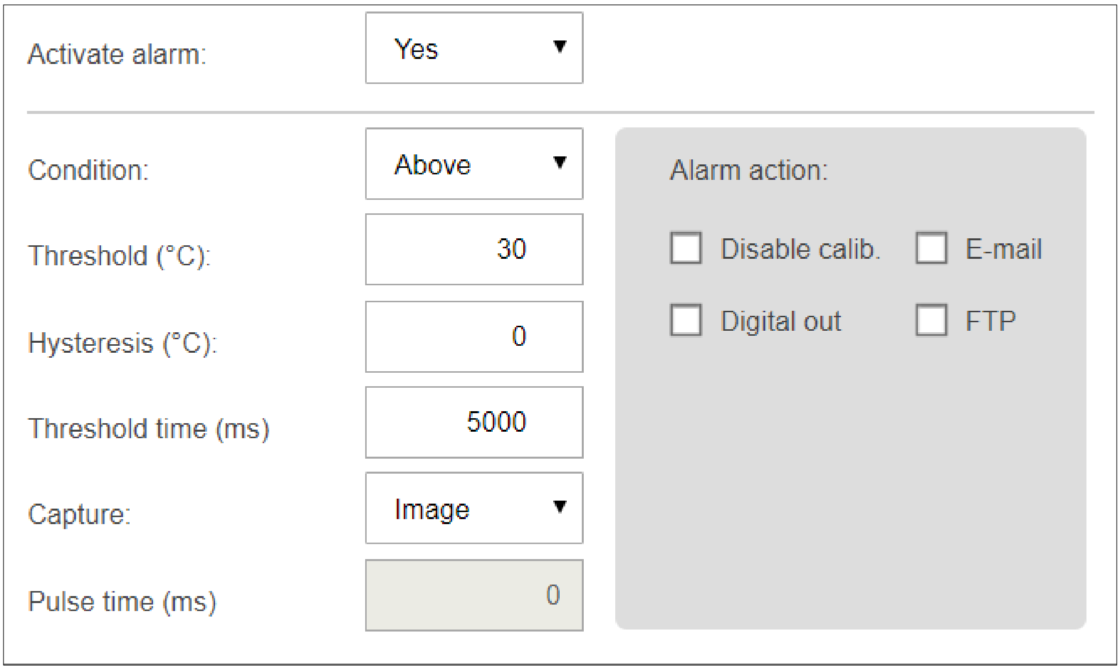
For this exercise I will use the FLIR AX8, but the main principles are the same regardless of IR camera.

Threshold time



1. **Click** the  icon to create a spot on your display.
2. In `Measurements & Alarms` pane, **click** the  icon to access the alarm dialog box (where you can configure alarm parameters and actions)

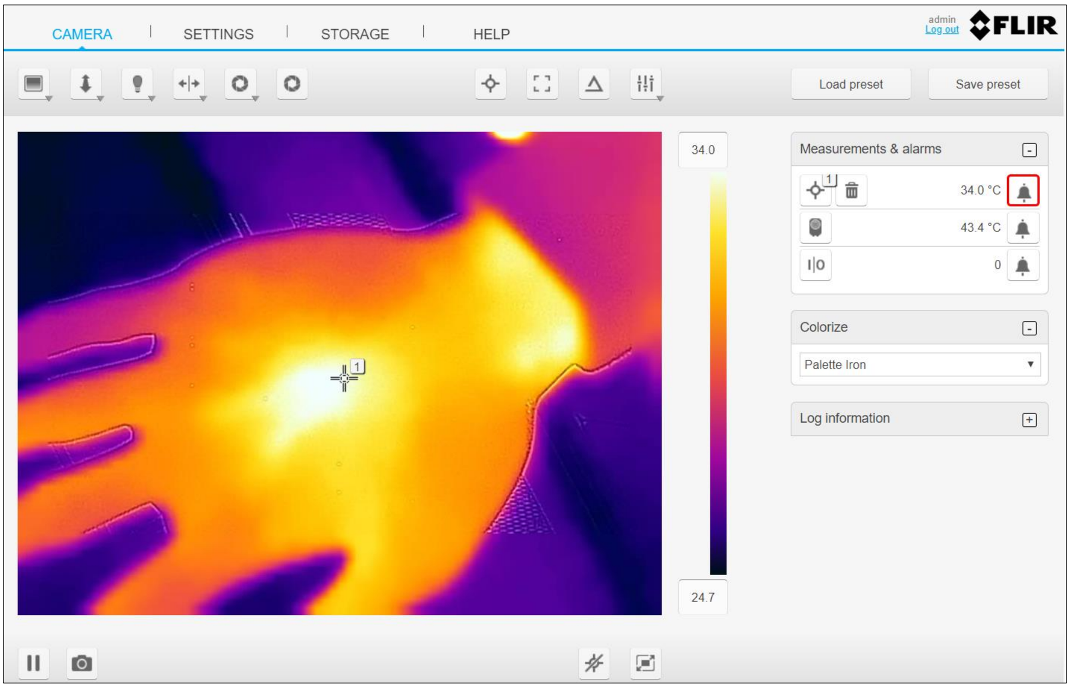
3. **Set** the parameters as shown in the image below.



4. **Click** outside anywhere outside the dialog box to close it.

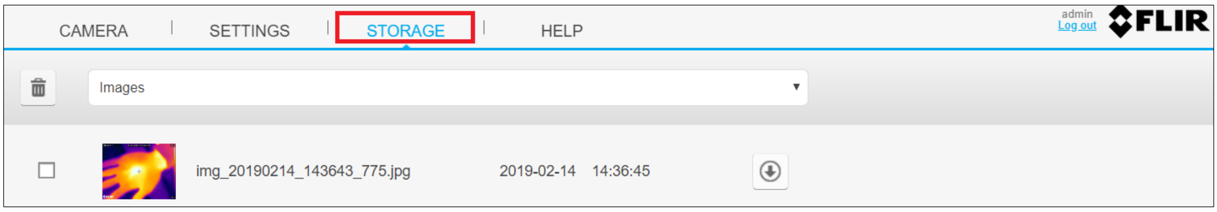
5. Hold your hand in front of the spot. Keep an eye on the icon.

6. After a little while, the icon should turn red: , which means that the alarm is triggered



The fact that the alarm didn't trigger at once is because of the threshold time. The threshold time was set to 5000 ms (5 seconds) – which means that the spot had to measure an apparent temperature above 30 °C for at least 5 seconds before the alarm could go off.

If you go the Storage tab, you should find an image capture of your hand, because you set the Capture to Image in the Alarm dialog box.



Hysteresis

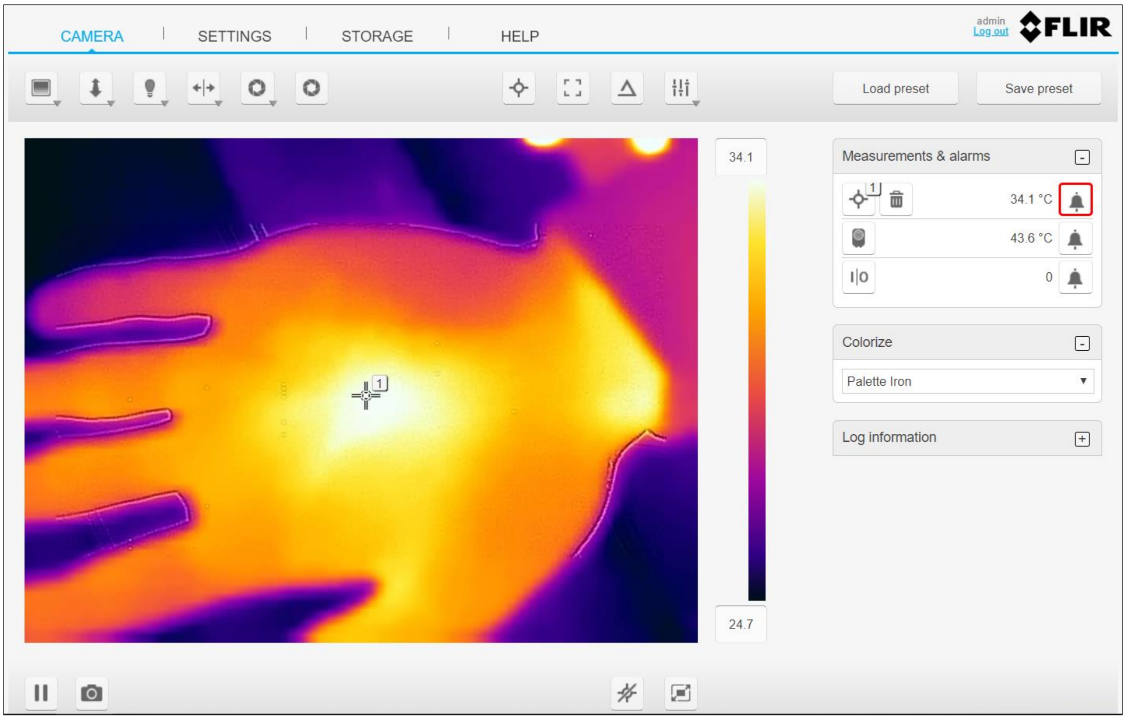
Go back to the Alarm dialog box by clicking the icon.

Set the Hysteresis (°C) to 20.

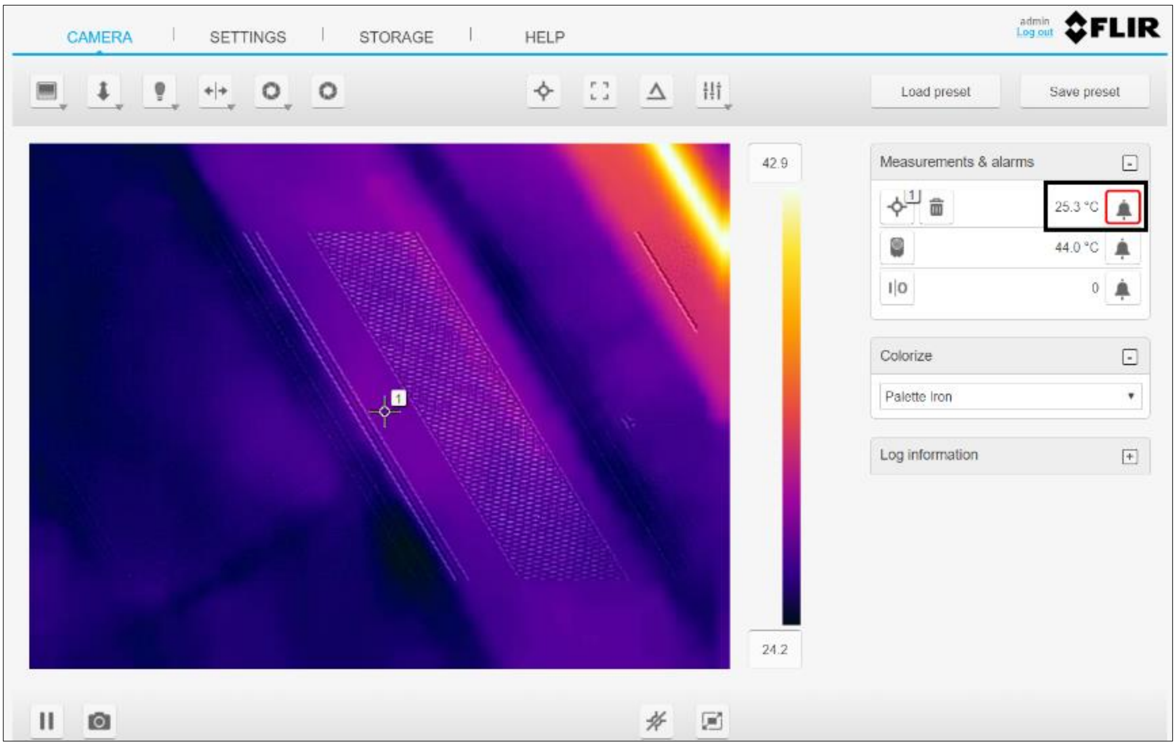
Now, do the same procedure to trigger the alarm – put your hand in front of the spot.

Once the alarm is triggered – take you hand away from the spot.

Do you notice anything different from before?



Despite the hand being removed and the temperature being *below* 30 °C, the alarm is still triggered!



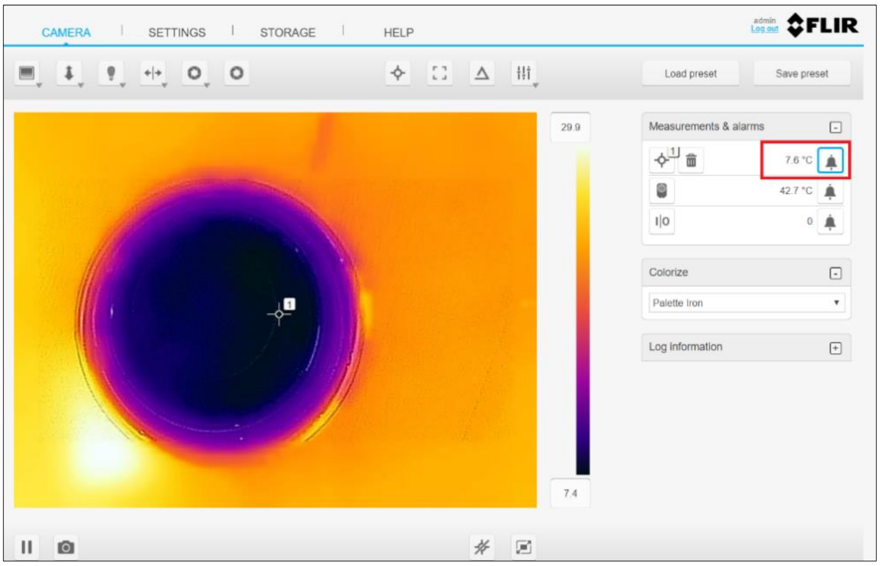
This is because of the hysteresis. The alarm was set to trigger for temperatures *above* 30 °C. The hysteresis was set to 20 °C. This means that *if* the alarm is triggered, it will stay triggered until the temperature in the spot drops 20 °C *below* the alarm temperature.

30 °C – 20 °C = 10 °C

The spot would then have to register a temperature *below* 10 °C for the alarm to be switched off.

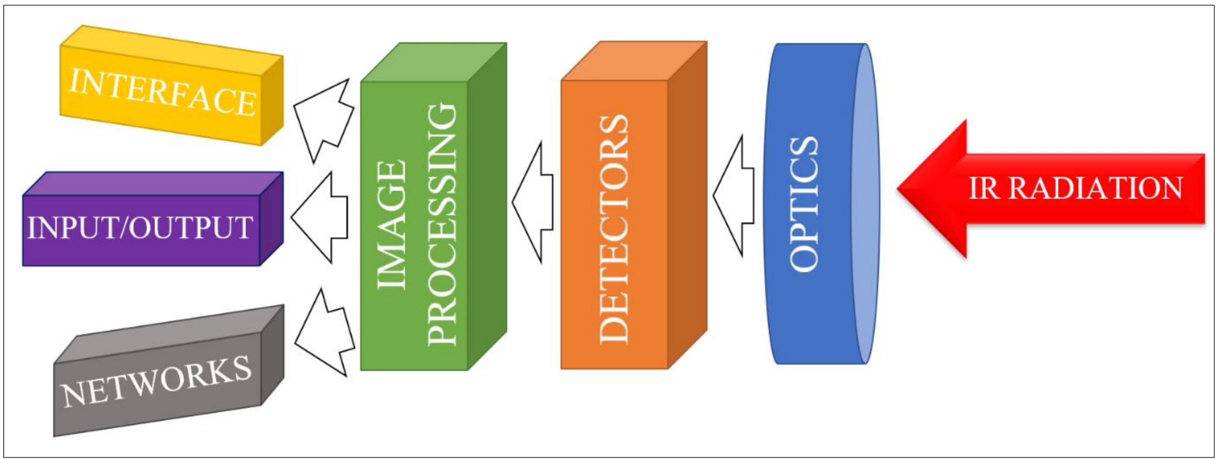
In the image to the right, I have targeted the Spot at my cup of cold water.

Because the apparent temperature is below 10 °C, the alarm is now switched off.



The IR camera system

In earlier chapters, we have discussed what happens before the incoming IR radiation reaches the IR camera. We have also discussed the user interface – what happens when the IR radiation has gone through the IR camera and is displayed with colors and numbers. Now's the time to discuss what happens inside the IR camera.



To do this, we will follow the IR radiation's way through the IR camera to the user interface. I have separated the IR camera system into three parts: Optics, Detectors and Image processing. Bear in mind that this chapter aims at letting you know the overall functions, and not exhausting the subjects fully. My wish is that you obtain a general knowledge of the IR camera system, so that you may utilize the full potential of your IR camera as well as understand its limitations.

Optics

Objectives

When you have worked through this part, my aim is that you will be able to

- Discover the different parts of the optics system of your IR camera
- Describe what the FOV and the IFOV is
- Identify the optical data that affect your IR imaging

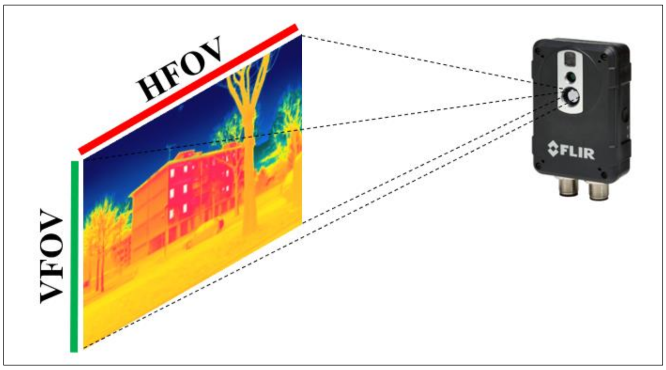
I've separated the Optics chapter into five parts:

> Field of View (FOV)
> Spatial resolution (IFOV)
> Focal Length
> Depth of Field / Minimum Focus Distance
> F-number

Field of view (FOV)

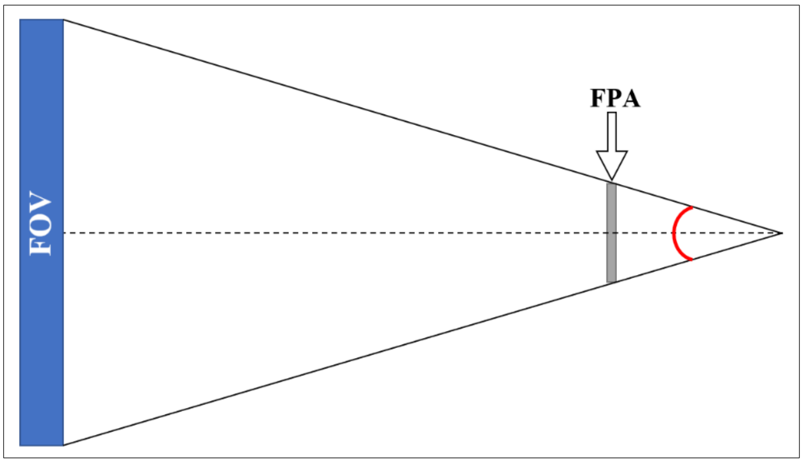
The Field of View – or the FOV, is the part of the scene that the IR camera covers. One can say that the FOV is the part of the view that you see on your IR image.

If you look in the user manual for your IR camera, you will see that the FOV is given in degrees, and not area. The FLIR AX8, for instance, has an FOV of 48° x 37°. These angles correspond to the horizontal length and vertical length of the FOV. I'll show you an image to illustrate.



The red line in the picture – the horizontal field of view (HFOV) – corresponds to the first value given in the user manual. For the FLIR AX8, the red line corresponds to 48°.

The green line in the image – the vertical field of view (VFOV) – corresponds to the second value (37° for the FLIR AX8). Since the FOV is given in angles, we must look at the image from the side. In the next image, we will ignore the camera body and look at the FOV with respect to the FPA – *the Focal Plane Array* – where the detector elements are arranged.



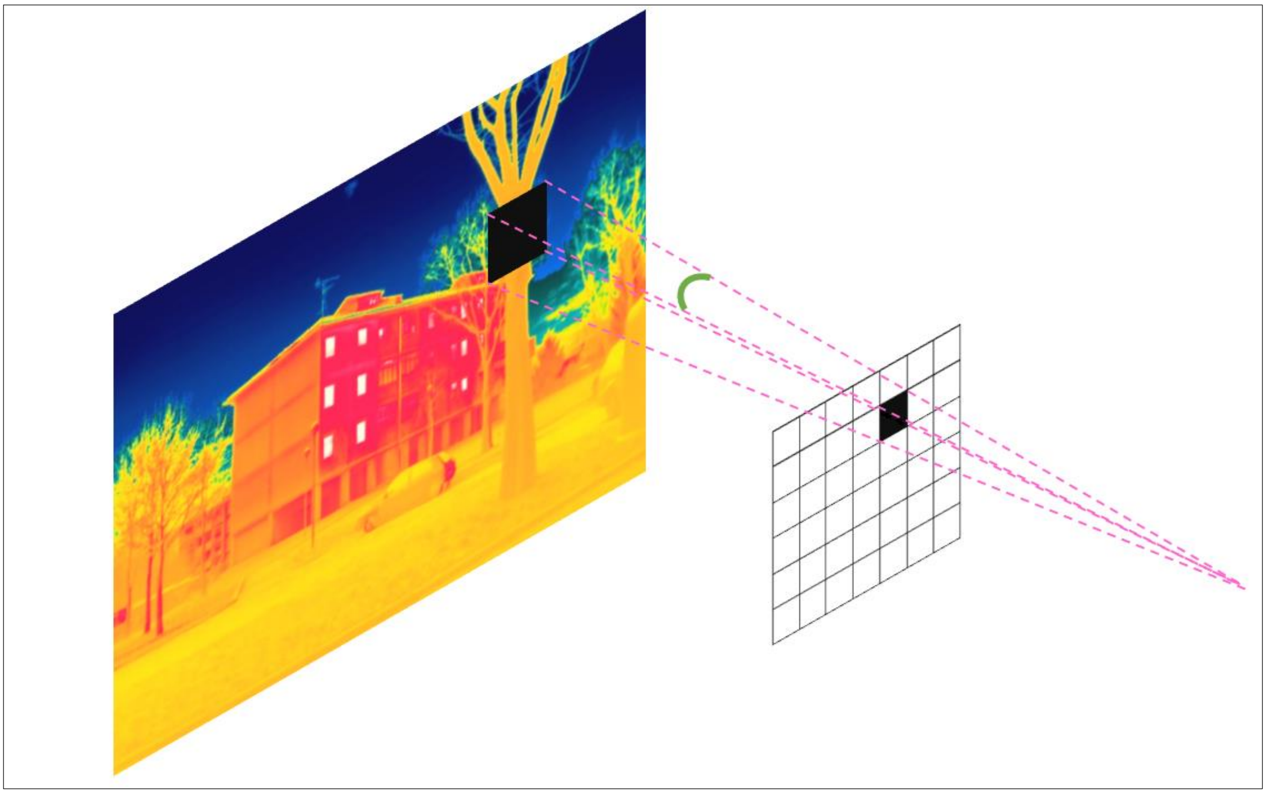
This image is very exaggerated to give you an idea about the FOV. The red angle in the image is the FOV. The FOV angle is given with respect to the FPA.

The angle itself might not be of great help for you when installing your IR camera. You can calculate the corresponding lengths by yourself – it can be difficult if you're not fond of math – but FLIR has an FOV calculator. Search for `FLIR FOV calculator` in your web browser, and let the calculator do the job for you.

Spatial resolution (IFOV)

IFOV is an acronym for Instantaneous Field of View. There are two ways of thinking about the IFOV. The first is that the IFOV is the FOV – what your IR image covers of the scene – divided into smaller squares. The size of these squares is given by the number of pixels – or detector elements – that your IR camera has.

Another way of thinking about it is that the IFOV is the area that is created by projecting a pixel on the IFOV. I'll show you an image to illustrate, since this can be a tricky matter.

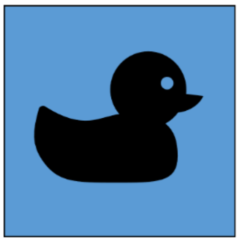


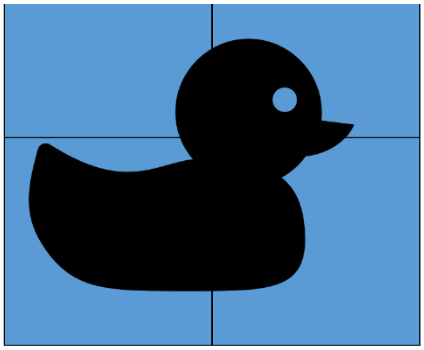
The IFOV – just like the FOV – is usually given as an angle. The IFOV angle is marked with a green arc in the image above. In the image, the grid in the front represents the FPA, and the black square represents one detector element (or pixel). The black square on the IR image is the projection of one detector element – the IFOV. If we were to project all the detector elements, we would cover the whole FOV.

One detector element can only measure one temperature at one time. We'll discuss this more thoroughly as we move along – but for now – this means that the projected detector element, the IFOV, is the smallest possible area you can measure. If the area or object you want to measure is smaller than your IR camera's IFOV, you will not get an accurate measurement.

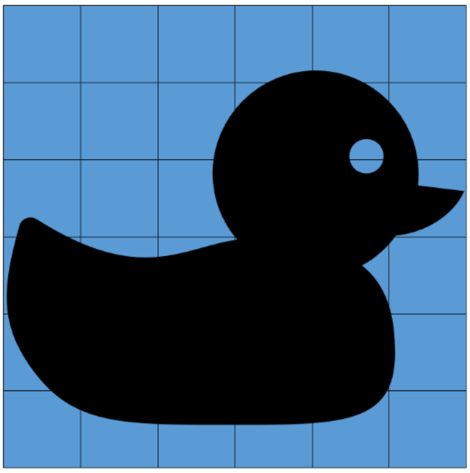
When it comes to the smallest possible area and the IFOV, you can think of it like this: imaging that you have a grid, where you can only use one color in every square. It would not be possible for you to paint something that is smaller than the size of one square. I would suggest that you move closer to your object, so that it covers more squares in the grid.

The principle is similar for the IR camera. If the blue square is the IFOV of your IR camera – the area that one detector element covers – then the duck would not be detected. The detector element gathers all the incoming IR radiation from the IFOV and would in this case probably record a temperature in between the duck and the background.





This duck – to the left – is covered by four detector elements, or four IFOVs. This is better, but still not good enough. As you can see, the majority of the IFOV in at least three out of the four is background, and not duck. This would probably not give an accurate temperature measurement.

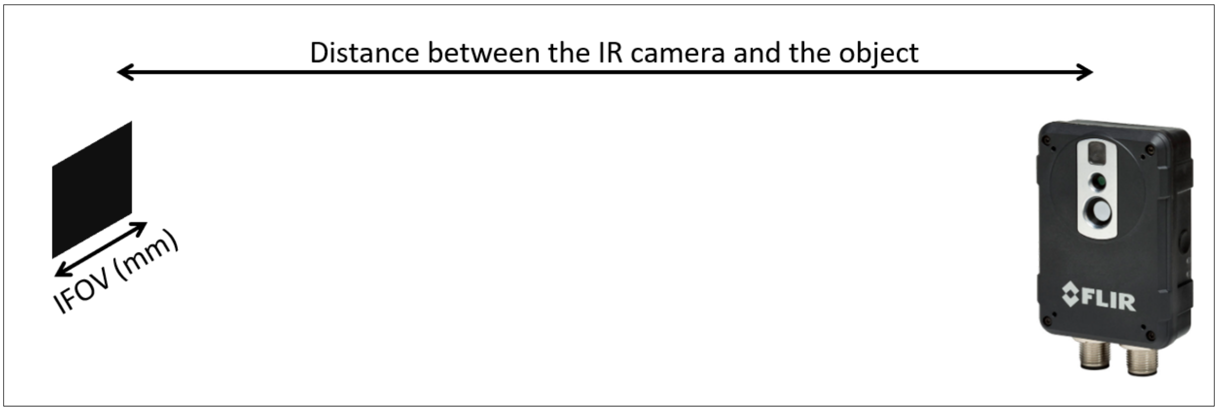


Now, this is more like it. This duck is covered by 6 x 6 detector elements. This you can use as a rule of thumb. *Your target object should be covered by at least 6 x 6 detector elements to get an accurate temperature measurement*. Although the number of pixels needed may vary depending on camera and conditions, the rule of thumb applies in most cases.

The IFOV is usually given as an angle, but in milliradians (mrad) instead of degrees. The FLIR FOV calculator I mentioned earlier also calculates the IFOV.

An easy approximation to calculate the IFOV in millimeters by yourself is to use the following formula.

$$IFOV\ (mm) = Distance\ to\ object\ (m) \cdot\ IFOV(mrad)$$



SAQ

IFOV and accurate temperature measurements

You have a FLIR AX8 and want to measure the temperature of a 50–millimeter in diameter object, from 2 meters afar.

Can you be certain that the temperature measurement is accurate?

You may find it helpful to use the FLIR FOV Calculator.

SAQ

You have a FLIR AX8 and want to measure the temperature of a 50–millimeter in diameter object, from 2 meters afar. Can you be certain that the temperature measurement is accurate?

The answer is **no**, we **cannot** be certain that the measurement is accurate. Now, why is that?

To answer this, I used the FLIR FOV Calculator found at `https://flir.custhelp.com/app/fl_download_datasheets`

There I found the FLIR AX series and clicked on the FOV calc.

Here we see that the IFOV at 2 meters is 22.08 millimeters.

D	0.50	1.00	2.00	5.00	10.00	25.00	50.00	100.00
HFOV	0.44	0.88	1.77	4.42	8.83	22.08	44.16	88.31
VFOV	0.33	0.66	1.32	3.31	6.62	16.56	33.12	66.23
DOF near	0.06	0.07	0.07	0.07	0.07	0.07	0.07	0.07
DOF far	Infinity	Infinity	Infinity	Infinity	Infinity	Infinity	Infinity	Infinity
IFOV	5.52	11.04	22.08	55.19	110.39	275.97	551.95	1103.90

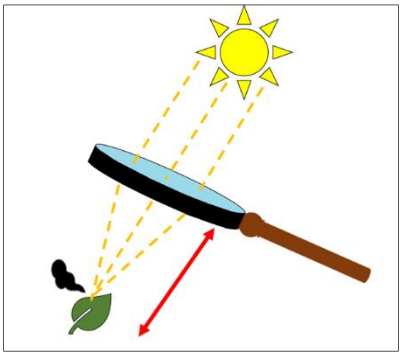
Our object is 50 millimeters in diameter, which is **larger** than the IFOV. That should guarantee an accurate temperature measurement, right?

No. The fact that our object is larger than the IFOV is **not enough**. Earlier I presented a rule of thumb that the object should be covered by at **least 6 detector elements** in diameter. 6 times the IFOV in this case is about 130 mm, and our object is a lot smaller than that. Thus, we cannot guarantee an accurate temperature measurement.

If you got this right, that is great. If not, I am very glad that you took the time to go through this response. My hope is that it made things clearer for you.

Focal length

The focal length of an optics system determines where the incoming radiation will focus. An easy image to illustrate this is to think of a magnifying glass and how it concentrates the sun rays to one spot. In the image to the right, the focal length is marked with a red double-arrow – that is, the length between the lens and the point where the rays converge. The main principle is the same in your IR camera. The lens system of your IR camera focuses the incoming IR radiation on the detectors. Easy put, one can say that the focal length is the distance between the lens and the detectors – or the FPA (the Focal Plane Array).



In general, one can say that optic systems in any kind of photography with a longer focal length leads to a greater magnification and a narrower field of view, than shorter focal lengths.

Your IR camera consists of several lenses, just like a normal camera. But unlike the normal camera, the material of the lenses is not glass. Glass does not usually transmit IR radiation (see page 45 for illustrations), but the element germanium does. Germanium is a very good transmitter of IR radiation, so most lenses in IR cameras are made of it.

Depth of field / Minimum focus distance

The depth of field tells us the maximum depth of a scene that stays in focus. In some IR cameras, the focus is fixed. This means that you cannot focus on a particular plane or object in the scene. The IR camera with a fixed focus will have an even focus in the whole depth of the image.

Three images with different focus. Image 1: Focus on the cords in the foreground. Image 2: Focus on the person in the middle of the image. Image 3: Focus on the person in the background.

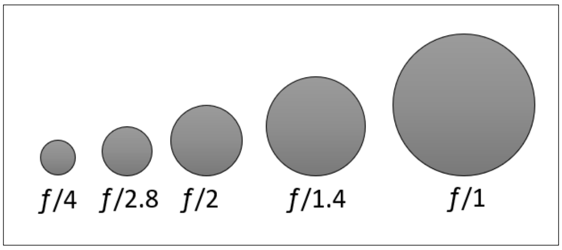
The focus is not fixed for all IR cameras. In many IR cameras, the focus shifts, either automatically or by the user. If you have an IR camera with changeable focus, *make sure that the focus is on your target object!* Otherwise, your temperature measurement might not be accurate.

f-number

The f-number is a measure of the amount of radiation reaching the detectors. It is dependent on the focal length and the aperture of your IR camera. The aperture is the opening through which the incoming radiation passes to get to focus on the detectors. You can think of it like the pupils of your eyes getting wider or narrower to adjust the amount of light entering your eyes.

The f-number is usually given in the form $f/1.4$, where f denotes the focal length and 1.4 is the f-number. The f-number is inversely proportional to the diameter of the aperture. What this means is simply that if you increase the diameter, the f-number will decrease. The formula for the f-number is $f - number = \frac{focal\ length}{aperture\ diameter}$.

Through this, it is easy to deduce that a lens with a low f-number – or a large diameter – allows for more IR radiation to go through it. More IR radiation will then reach the detector, which means that the detector will react more to this incoming IR radiation.



Germanium – the material the lens is made of – is a very expensive material. Having a lens with a large diameter will drastically increase the cost of the IR camera. IR camera manufacturers are therefore using lenses with a small diameter, or high f-number. There is also a practical reason – beside the fact that a smaller lens leads to a smaller and more manageable IR camera. The depth of field is very shallow for big, low f-number lenses. This means that objects close to the IR camera will be in focus, but objects far away will be blurry.

Typical f-numbers are $f/1.4$, $f/2$ and $f/2.8$.

SAQ

True or False?

A large f–number means a large intake of IR radiation through the lens.	True	False
The lenses for IR cameras are made of the same material used in visual cameras.	True	False
The IFOV is the minimal size an object can have to give us an accurate temperature measurement.	True	False

SAQ

All three statements are **False**. More importantly, let's see **why** that is.

A large f–number means a large intake of IR radiation through the lens.	True **False**

- Perhaps you though that the f-number tells us something about how much radiation we let in through the lens. However, a large f-number means a small intake of IR radiation and a small f-number means a large intake of IR radiation.

The lenses for IR cameras are made of the same material used in visual cameras.	True **False**

- In some cases, IR cameras work similarly to visual cameras. However, the lens material is different. IR camera lenses are mostly made of germanium, because it is a material that is good at letting IR radiation through. If we would use glass for IR camera lenses, little IR radiation would reach the detectors that analyze the scene. This would make it very difficult to get accurate IR images. Since IR camera lenses are made of germanium and visual camera lenses are not, the statement is false.

The IFOV is the minimal size an object can have to give us an accurate temperature measurement.	True **False**

- The IFOV tells us the minimum size of object for it to be detected by the IR camera. However, this is not the same as giving us an **accurate** measurement, thus the statement is false. In order to get an accurate temperature measurement, it is recommended for the object size to be a couple of times larger than the IFOV. Depending on the application, the recommended minimal measurement area varies. However, a good **rule of thumb** is for the object to be at least **6 times the IFOV** in diameter.

If you got these statements right, excellent. If not, excellent. The fact that you are reading this means that you have gone through the feedback above and hopefully reflected over these concepts some more.

SUMMARY

In this part, we have discussed the optics of the IR camera system. I will give you a list with the main parts that we've covered. When you go through the list, try to reflect over what you have read.

> The Field of View (FOV) is the part of the scene that your IR camera covers. The value is given in degrees. FLIR has an FOV calculator on their web page.
>
> The Instantaneous Field of View (IFOV) is the part of the FOV that one detector covers, or the projection of one detector onto the FOV.
>
> The focal length is a measure of where the lens focuses the incoming radiation.
>
> The depth of field is a measure of how deep in the scene the IR camera focuses.
>
> The f-number is a measure of how much IR radiation is let in to the detectors.

Detectors

Objectives

When you have worked through this part, my aim is that you will be able to

- Identify the limitations of the detector
- Briefly explain the process of the detectors
- Use your IR camera sensible, in order to get accurate readings

I've separated the detectors chapter into six parts:

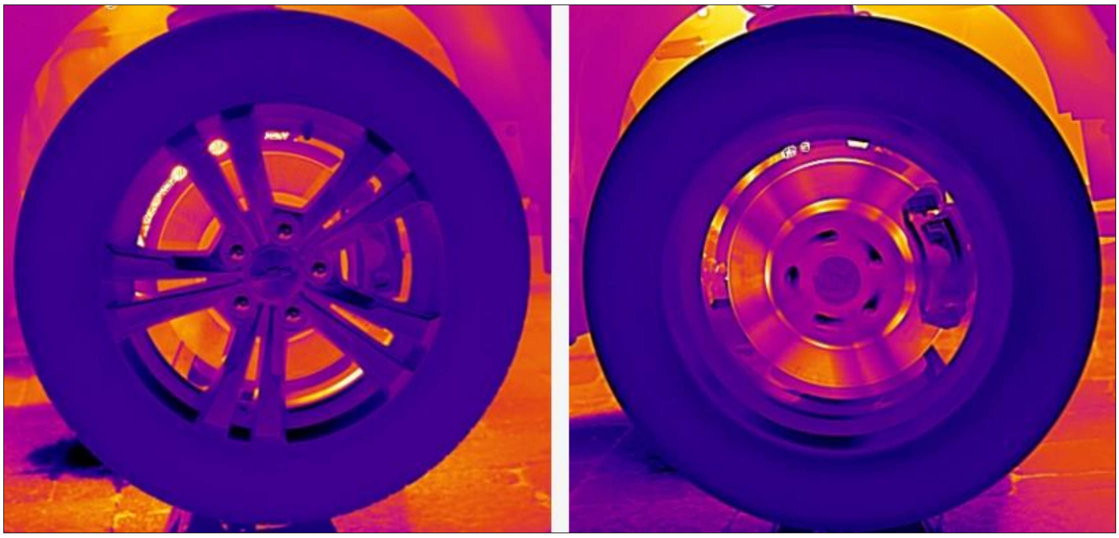
- Detector Type
- IR Resolution
- Thermal Sensitivity (NETD)
- Accuracy
- Detector Time Constant
- Spectral Range

Detector type

The detectors are the heart of the IR camera system, in that they are the elements that actually detect the incoming IR radiation. The other elements function is to focus the radiation on the detectors or translate what the detectors are measuring, but the detectors are the key part.

Detector types can be divided into two categories: Cooled detector and thermal detector (or uncooled detector). The cooled detectors are so called because they need to be cooled down to temperatures as low as -200 °C. The fact that they need to be cooled down makes them very expensive, but they are superior to the uncooled detectors in many aspects.

An example of this is the image below.

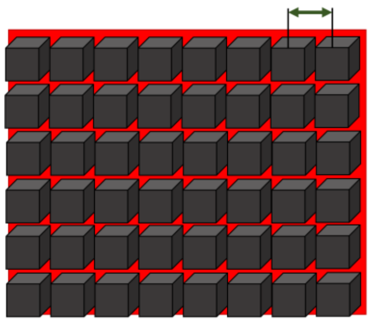


The images are of a tire rotating at 20 mph. The left is taken with a cooled detector and the right with an uncooled. As you can see, the cooled detector captures the tire as if it was still. The uncooled image appears blurry.

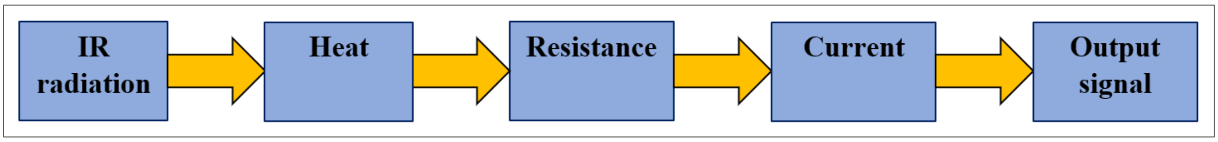
Although the uncooled detectors are superior – the uncooled can operate in room temperature – which is their main advantage. The choice between an uncooled and a cooled detector depends on application. In most applications, there is no need for the performance of a cooled detector. They are mainly used in R&D (research and development) and science labs, so I will not discuss them further here.

The Microbolometer

When it comes to uncooled detectors, the microbolometer is a usual type. In the IR camera, the FPA consists of a matrix of microbolometers, where each and every one of them covers their part of the FOV that makes up your IR image. The distance between the center of one detector and the center of the neighboring detector is called the detector pitch (marked with a double arrow in the image to the right). The magnitude of the detector pitch and the microbolometer is micrometer. That means that one detector element is around 10 µm = 0.00 001 m wide.

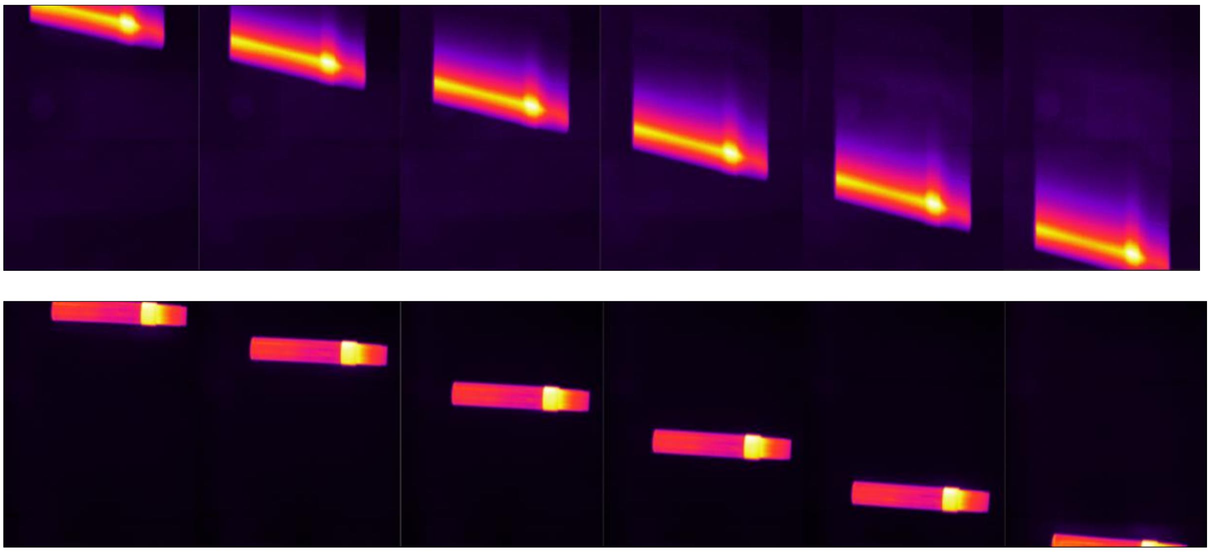


The microbolometer is made up of a material that is sensitive to IR radiation in order to detect it. There is a chain of transformation that the IR radiation goes through before it reaches your eyes, in the form of an IR image.



Incoming IR radiation hits the microbolometer. The microbolometer absorbs the heat, which in turn changes the resistance of it. The change in resistance changes the current through the microbolometer. This change in current is converted into a digital number that is later transformed into a colored pixel in your IR image.

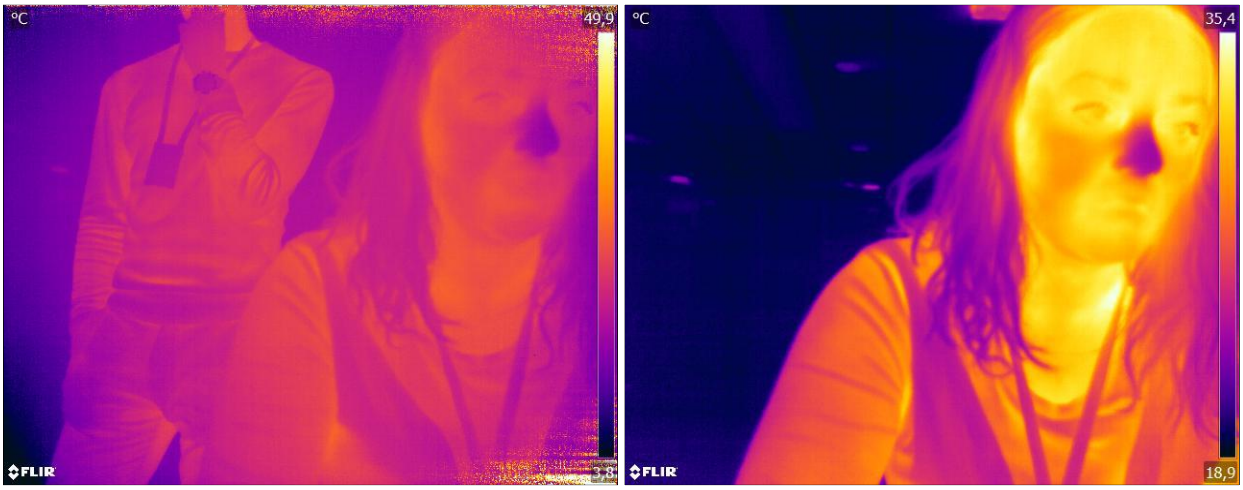
Since detecting the incoming IR radiation requires a physical change in the microbolometer (the absorptance of heat and resistance change), the response time will depend upon it. By this I mean that with a thermal detector, such as the microbolometer, the detector time constant (see Detector time constant below) cannot be improved – it is a physical property. This can show when trying to measure fast events with a microbolometer detector. It may result in blurring – or even missing the fast event – in the resulting image.



Heated pen falling, captured with a microbolometer (top) and a cooled detector (bottom).

Non-Uniformity Correction (NUC)

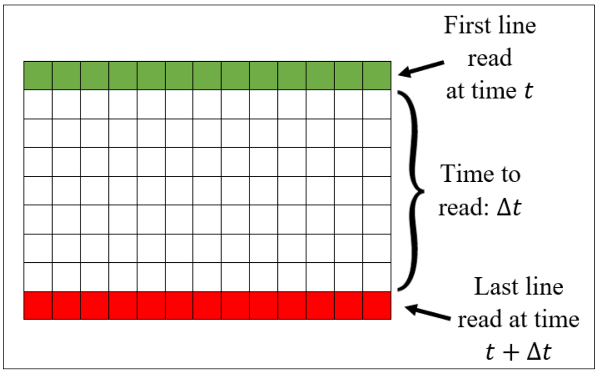
The clicking you might have heard from your IR camera is caused by the Non-Uniformity Correction, or the NUC. All detectors are separated and basically works alone with absorbing heat and giving of an electrical signal. There are no perfect detectors, so they might drift a little in their measurement. This drift may cause an unnatural "jump" in temperature between neighboring detectors, making the IR image "more pixelated". The NUC corrects this by sending down an evenly tempered plate in front of all the detectors. Since the IR camera "knows" the temperature of the plate, all the signals from the detectors can be corrected, so that they send the same electrical signal when presented to the plate.



The IR camera capturing the image to the left hasn't NUCed for over an hour. The image to the right is captured directly after a NUC. As you can see, the image to the left has plenty of "bad" pixels. The temperature span is not corrected according to the temperatures in the scene, making the IR image low in contrast. In the image to the right, when the IR camera has NUCed, these faults are corrected.

Shutter

The shutter is the feature of the IR camera that "reads" the current given from the detectors. It is essentially the messenger that sends the value from the detectors to the electronics of the IR camera. The cooled detectors have what is called a *global shutter*. The global shutter is able to read out all the detectors at the same time. The shutter in an IR camera with thermal detectors cannot do this, since the system would quickly become overheated. IR cameras with a microbolometer instead has a *rolling shutter*. The rolling shutter reads the detector signals line by line.



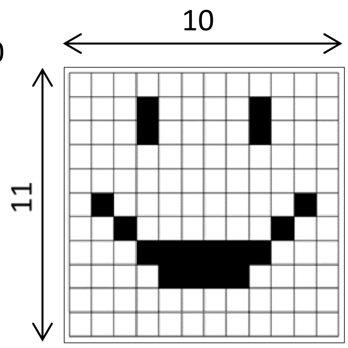
This may result in smearing in the IR image when fast events are recorded.

Image taken with a rolling shutter, causing the fast-moving blades of the fan to look distorted.

IR resolution

The IR resolution of an IR camera tells us how many detector elements it has. They are often called pixels in the user manuals. The amount of detector elements ranges from 80 x 60 to 1280 x 1024. So, what does it mean to have, for instance 80 x 60 pixels?

The image to the right has 10 x 11 pixels. As you can see, it's not the greatest resolution. 10 x 11 pixels means that there are 10 rows and 11 columns of pixels in a matrix.



The number of pixels given in your user manual may not be the same size as the resulting IR image will be. Some IR cameras sample the image, so that the size is more suitable for the viewing program. We'll discuss image sampling more thoroughly in chapter Sampling. Not all IR cameras sample the images when streaming, the FLIR A310 for instance, has an IR resolution of 320 x 240 pixels and streams in the same format.

The main point about pixels is that the pixel is the smallest possible building block of the image – one projected pixel makes up the IFOV, as we discussed earlier. This means that the smiley face in the image above cannot have a smoother smile or rounder eyes. To accomplish this we need more pixels, since we cannot split existing pixels in half. In IR resolution one pixel equals one detector, which means that there is a limit to the resolution of the IR image. One detector can only detect one temperature at one time, and not a variety of temperatures inside its detection area.

Thermal sensitivity (NETD)

NETD is an acronym for *Noise Equivalent Temperature Difference* and this is essentially the lowest temperature difference that the IR camera can measure.

Thermal sensitivity/NETD	< 0.10°C @ +30°C (+86°F) / 100 mK

This is an example of what it might look like in the user manual. It is from the user manual of the FLIR AX8. Translating the text yields that at 30 °C or 86 °F the maximum temperature difference that the FLIR AX8 can detect is less than 0.10 °C or 100 mK. This means that if I view an object with a temperature of 30.10 °C with the FLIR AX8 and the background is 30.00 °C, the IR camera will detect the difference and assign the object and the background with different colors. One can say that the NETD determines what can be distinguished from each other in the IR image and not be blurred together.

The Noise that is referred to NETD is the noise that comes from the IR camera itself. It is not an auditory noise, but a thermal. The detectors and electronics of the IR camera also has a temperature, which means that they emit thermal radiation. This thermal radiation will also be detected by the microbolometers. The noise from the IR camera is what determines the NETD, and subsequently part of the resolution. Since the temperature of the IR camera limits the minimum temperature it can measure, it is easy to see why cooled detectors generally has lower NETD than uncooled ones.

The IR camera is calibrated to operate at a certain temperature, that is, the temperature of the IR camera. One thing to bear in mind is that if the IR camera recently has been turned off, or if the surrounding temperature changes extremely, the IR camera will no longer be at the same temperature. This may cause inaccurate measurements. *It is best to wait ten minutes or so after starting the IR camera, so that it has time to warm up and give accurate measurements*.

Accuracy

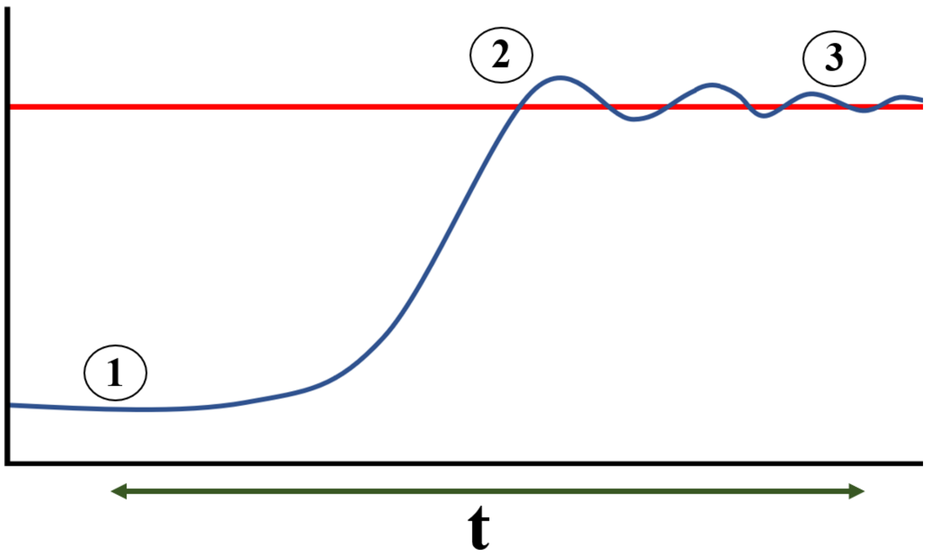
The accuracy of an IR camera is typically ± 2 °C or 2 % of the reading. These numbers are the result of an uncertainty analysis technique called “Root-Sum-of-Squares”, or RSS. The main idea of the technique is to calculate all possible partial errors that may contribute to the error total. Among these partial errors are camera response, calibrator accuracy and an estimate of wrongly set object parameters (this does not mean that you shouldn’t be as meticulous as possible when setting the parameters). All the partial errors are squared and added up. Lastly the square root of the total sum is taken. This is to ensure that two partial errors do not cancel each other out, so that the total accuracy have at most that margin of error.

When the IR cameras are tested in laboratory environment by FLIR lab personnel, the accuracy is as low as ± 1 °C, but this requires lab conditions. The number ± 2 °C is valid for short ranges – less than 20 meters – as the partial error contribution from the atmosphere increases with range.

All IR camera manufacturers does not calibrate for ambient temperature compensation. That is, the IR camera’s own temperature changing and affecting the measurement. FLIR uses the measurement data for the IR camera at different ambient temperatures and includes it in the calibration equation. This ensures accurate temperature readings through the entire range of operating temperatures. The operating temperature of an IR camera typically ranges from -10 °C to 50 °C. If the operating temperature is not accounted for, the inaccuracy can increase to up to ± 10 °C. Although compensated for, it is wise to let the IR camera warm up before making critical measurements, as I mentioned earlier.

Detector time constant

The detector time constant – as I mentioned earlier – is the time it takes for the detector to absorb enough heat to produce a signal. Now, this is not completely correct, but the way of thinking works for our purposes. The main principle is that the detector needs enough time to heat up and change its resistance to subsequently change the current. This process is not linear, the change in resistance will at first “overshoot”, and then go up and down until it finally settles on the correct value. The time this process takes is the detector time constant. The typical value for microbolometers is 12 ms = 0.012 s.



It may be easier to understand when investigating the steps in the image above. The red line is the value of the output signal, the blue line is the current through the detector, and the green line with a “t” is the detector time constant.

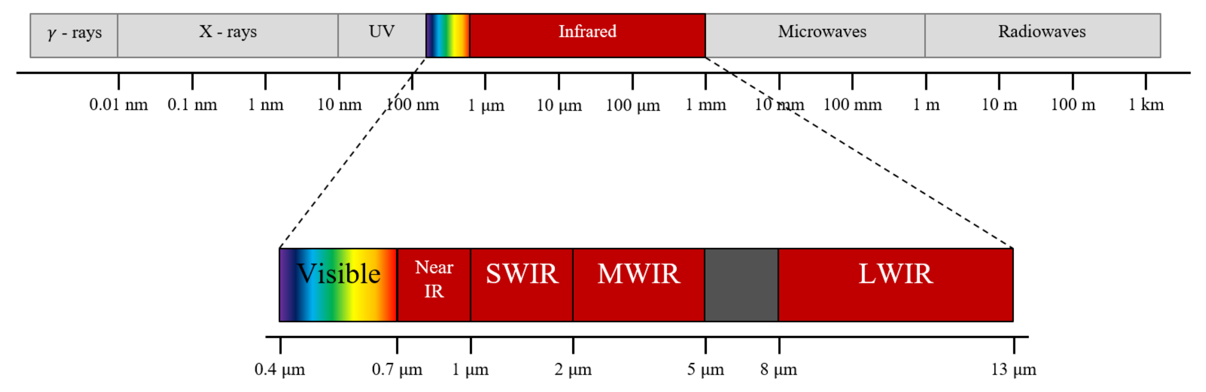
1. The thermal radiation hits the detector and the detector starts absorbing heat and consequently starts to change the current.
2. The current through the detector “overshoots” and oscillates around the output signal value.
3. The current through the detector stabilizes to the output signal value.

The time it takes for this process is the detector time constant.

As I mentioned earlier: *the detector time constant cannot be manipulated by the user, it is a constant*.

Spectral range

The detectors in your IR camera does not detect thermal radiation over the total spectrum, but in a *spectral band*, or *spectral range*. The spectral range for the FLIR AX8 is 7.5-13 µm, which corresponds to the spectral band that is called Long Wavelength InfraRed (LWIR)



As you can see, in the image above, there are two more spectral bands where IR imaging is used. These are Short Wavelength InfraRed (SWIR) and Medium Wavelength InfraRed (MWIR). As you also might have noticed, there is a grey area between the MWIR and the LWIR. This is because the transmissivity of the atmosphere is very low between wavelengths 5 µm and 8 µm. That is, the atmosphere lets through very little radiation at these wavelengths, or the atmosphere is opaque there.

The energy of the thermal radiation is inversely proportional to the wavelength. This means that as the wavelengths go shorter, the energy goes larger. The more energy that hits the detectors, the more heat they can absorb. IR cameras in the SWIR band is therefore more sensitive to differences in radiation, but they cannot be calibrated below 350 °C. It is also in this temperature area where the SWIR cameras are most suitable. If you need to measure along a wide spectral range – for instance from -40 °C up to several hundred degrees – the best choice is the LWIR spectral band. MWIR cameras are most suited for measuring temperature changes starting at room temperature. They can be calibrated from -20 °C with no upper limit.

SAQ

True or False?

The thermal time constant of microbolometers can be made shorter through careful calibration.	True	False
The thermal sensitivity (NETD) of uncooled IR cameras is usually about ±2 °C.	True	False
IR cameras with uncooled detectors use rolling shutter which may cause distortion effects when recording rapid events.	True	False

SAQ

The correct answers are **False**, **False**, **True**. Let's look at why.

The thermal time constant of microbolometers can be made shorter through careful calibration.	True	**<u>False</u>**

- You may have thought that a lot can be improved through careful calibration. That is true in many cases, but unfortunately not for the thermal time constant. As the name states, it is a **constant**. It represents the time it takes for a detector element to **heat up**. This is a physical property and a **limiting factor** for the speed in uncooled IR cameras.

The thermal sensitivity (NETD) of uncooled IR cameras is usually about ± 2 °C.	True	**<u>False</u>**

- We need to separate between the **thermal sensitivity** (NETD) and the **accuracy** of the IR camera. The NETD is usually 0.1 °C or lower, while accuracy is usually ± 2 °C. I understand that these can cause some confusion as it seems that both refer to the same thing; some accuracy within the camera system. Then why is the NETD typically much lower than the accuracy? Here we need to make an important distinction between **relative** and **absolute** temperature measurements. Thermal sensitivity (NETD) refers to **relative temperature measurements**. Let's assume that our IR camera has an NETD of 0.1 °C. Then two objects that differ by 0.1 °C will be determined by the IR camera to have **different** temperatures. However, this does not say anything about the actual temperatures of the objects. On the other hand, the **accuracy** refers to **absolute temperature**. If our IR camera is properly calibrated and we have correctly set the object parameters, then our IR camera should measure the temperature of an object to within ± 2 °C of its **actual temperature**.

IR cameras with uncooled detectors use rolling shutter which may cause distortion effects when recording rapid events.	**<u>True</u>**	False

- This is absolutely true, and an important limitation to know about in uncooled detectors. Uncooled detectors cannot read the whole IR image at once, and instead use rolling shutter. This means that the detector signals are read line by line. When the event we are trying to record matches or exceed the read-out speed of the detector signals, the IR image may get distorted.

If you got these questions right, well done! If not, don't worry about it. It happens to the best of us. The fact that you have gone through my response is fantastic. Hopefully, things have become a little clearer.

SUMMARY

In this part, we have discussed the detectors of the IR camera system. I will give you a list with the main parts that we've covered. When you go through the list, try to reflect over what you have read.

> There are two types of detectors: cooled and uncooled. One type of uncooled detector is the microbolometer.
>
> The IR resolution gives you the number of pixels, or equivalently the number of detector elements in your IR camera.
>
> The NETD gives you the lowest temperature difference that your IR camera is capable to detect.
>
> The accuracy gives you the margin of error of your temperature measurements.
>
> The detector time constant for a microbolometer is around 12 ms and it cannot be changed by the user.
>
> There are three spectral ranges used in IR imaging: LWIR, MWIR and SWIR.

Image processing

Objectives

When you have worked through this part, my aim is that you will be able to

- Distinguish between an image streaming camera and a smart sensor camera
- Describe the difference between radiometric and compressed image streaming
- Recognize the link between histogram equalization and contrast
- Recall that the analysis functions do not update with the same frequency as the image

I've separated the image processing chapter into four parts:

- Bit Depth
- Image streaming cameras
- Smart sensor cameras
- Sampling

Bit depth

Depending on what type of IR camera you have, the information from the detector elements will be processed differently. Some settings can be chosen by the user, and some are predetermined by the manufacturer.

In order for us to discuss the image processing, we are going to have to look at bits. The bit depth of an image or an image stream is a measurement on the amount of information that you IR camera is able to send. If you IR camera streams in 8-bit for instance, then every detector element – or pixel – can assume a value between 0 and 255. These numbers are not arbitrary, the 8 in 8-bit streaming tells us that there is $2^8 = 256$ distinct types of information that your IR camera can send.

In your IR camera, different detectors will receive a different amount of IR radiation. These different amounts of IR radiation need to be quantified in order for the IR camera and the connected computer to being able to compose the IR image. When all the detector values are read, the IR camera compares these to assign a value between 0 and 255 corresponding to the amount of IR radiation (if you are streaming in 8-bit format). The raw data in the IR camera from the scene is thus a matrix of numbers. What happens with the matrix of numbers depends on what IR camera you have: an image streaming camera or a smart sensor camera.

Image streaming cameras

Image streaming cameras – such as the FLIR Ax5, the FLIR A315 and the FLIR A615 – need to be connected to a computer to operate, they cannot work by themselves. They can essentially send two types of information to the computer: the matrix of numbers or the calculated temperature value of the numbers in the matrix. When the image streaming camera calculates the temperature value of each pixel, it often doesn’t come out as the actual temperature, the image streaming camera can work in a mode called *temperature linear*. This simply means that the image streaming camera can calculate a number that is linearly related to the temperature. Let’s look at an example. The FLIR Ax5 has temperature linear mode and when you enable it, you can choose which resolution you would like – high or low. If you choose high resolution, every output value for every pixel is given by the following formula.

$$T = 0.04 \cdot S$$

Where S is the pixel value and the temperature is given in kelvins. For the low resolution, the multiplying factor is 0.4 instead of 0.04. You can find all the information about the temperature linear mode of your image streaming camera in the user manual.

When the image streaming has sent the matrix with numbers to the computer, it is the computers job to assign colors corresponding to the numbers. If you use an 8-bit greyscale, for instance, the pixels with the number 0 are usually assigned the color black. The pixels with the number 255 are usually assigned the color white, and the numbers in between are assigned grey colors in their order of the spectrum.

The main principle is the same whichever palette you use.

8-bit image. $2^8 = 256$ *colors*

7-bit image. $2^7 = 128$ *colors*

6-bit image. $2^6 = 64$ *colors*

5-bit image. $2^5 = 32$ *colors*

4-bit image. $2^4 = 16$ *colors*

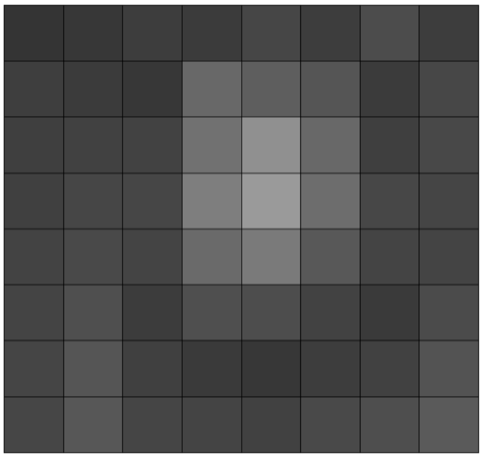
3-bit image. $2^3 = 8$ *colors*

2-bit image. $2^2 = 4$ *colors*

1-bit image. $2^1 = 2$ *colors*

Let's look at an example! Say that your image streaming camera has 8 x 8 pixels – that is, 8 x 8 = 64 detector elements – and that it is streaming in 8-bit format. If the information that your image streaming camera sends to the computer is this matrix to the left below,

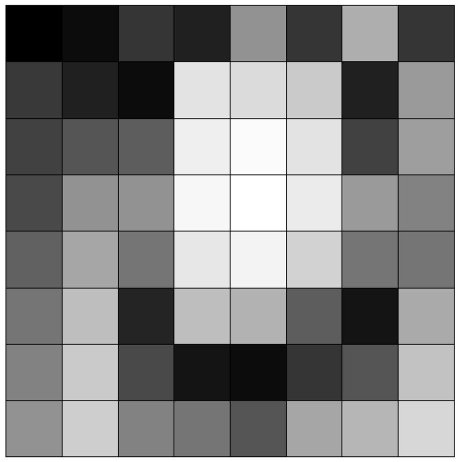
$$\begin{bmatrix} 52 & 55 & 61 & 59 & 70 & 61 & 76 & 61 \\ 62 & 59 & 55 & 104 & 94 & 85 & 59 & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 & 79 & 77 & 66 & 58 & 75 \\ 69 & 85 & 64 & 58 & 55 & 61 & 65 & 83 \\ 70 & 87 & 69 & 68 & 65 & 73 & 78 & 90 \end{bmatrix}$$



the computer will translate it by giving the corresponding color and show the image to the right above.

If you look closely at the numbers in the matrix above, you can see that they are clustered in the middle of the grey scale. The lowest number is 52 and the highest is 154, so the image does not use the full grey scale. This can happen if you're observing a scene with small temperature differences – that is, all the detector elements will receive a similar amount of IR radiation. These images can have low contrast, so it may be difficult to distinguish different features in it. Then you can use the *histogram equalization*. The histogram equalization basically makes the computer use the whole grey scale to enhance the contrast. If we use the histogram equalization on the example above, we would get the following result.

$$\begin{bmatrix} 0 & 12 & 53 & 32 & 146 & 53 & 174 & 53 \\ 57 & 32 & 12 & 227 & 219 & 202 & 32 & 154 \\ 65 & 85 & 93 & 239 & 251 & 227 & 65 & 158 \\ 73 & 146 & 146 & 247 & 255 & 235 & 154 & 130 \\ 97 & 166 & 117 & 231 & 243 & 210 & 117 & 117 \\ 117 & 190 & 36 & 190 & 178 & 93 & 20 & 170 \\ 130 & 202 & 73 & 20 & 12 & 53 & 85 & 194 \\ 146 & 206 & 130 & 117 & 85 & 166 & 182 & 215 \end{bmatrix}$$



After the histogram equalization, the image does now have more contrast and uses the whole grey scale. The lowest number in the matrix is now 0, and the highest 255, corresponding to the maximum and minimum values of the 8-bit grey scale.

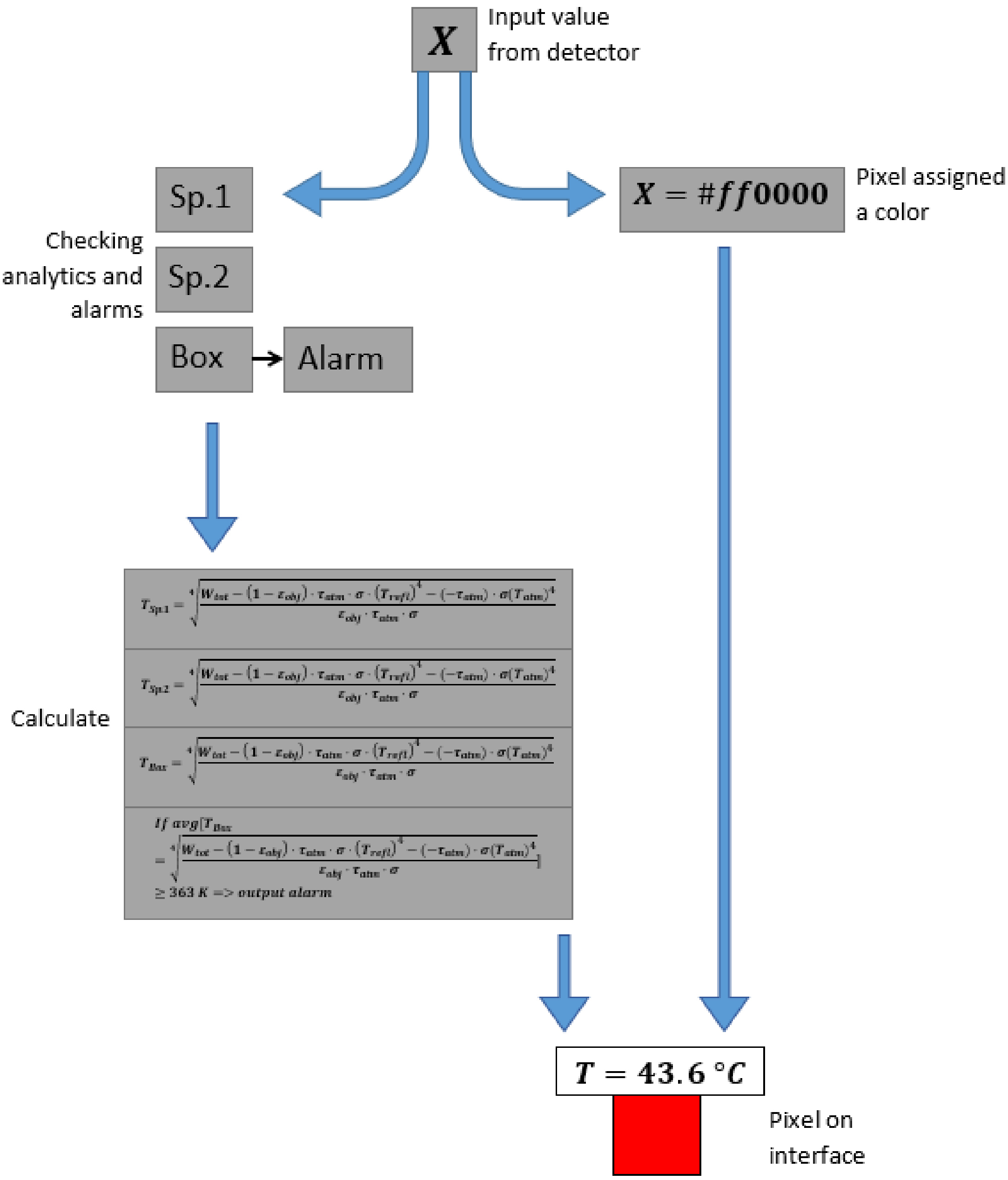
The image streaming camera sends the value of the amount of IR radiation of every pixel – be it temperature linear or signal linear. Each pixel value contains information about the IR radiation of the scene, and this is called *radiometric streaming*. The alternative is *compressed streaming*, where the information in all the pixels is the color, and not temperature value. Image streaming cameras use radiometric streaming, and smart sensor cameras use compressed streaming.

Smart sensor cameras

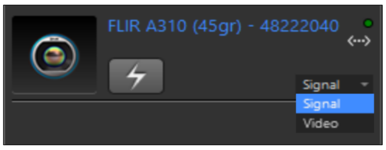
The main difference between a smart sensor camera and an image streaming camera is that the image streaming camera needs a connected computer to analyze the scene, and the smart sensor camera can do this by itself. You can think of it as the smart sensor camera having a computer of its own inside it, whilst the image streaming camera records the scene and lets the connected computer do the analysis. The image streaming camera sends information about every pixel – radiometric data – but lets the computer colorize and analyze it. This means that if you want to set an alarm or monitor a certain area of the scene, you have to program it yourself on a connected computer with the image streaming.

The smart sensor camera has analysis functions itself, which means that you can set alarms and monitor area, and then disconnect the computer. The smart sensor camera needs only be connected to a power source and does all the calculations itself – such as calculating the maximum and minimum temperature of an area – and raise the alarm when the set criteria is met.

Smart sensor cameras stream in a compressed format, but whenever an image is captured – be it by hand or by alarm – the image will be in radiometric format. The smart sensor camera only sends temperature information for those pixels where there are spots and boxes, and color numbers for the rest. Since the analysis functions require the smart sensor camera to perform calculations, these will not stream in the same speed as the color values. The analysis function may stream in 3 Hz – that is, 3 updates per second – while the image streaming may be in 9 Hz, or higher. The analysis streaming also depend upon how many analysis functions you set, since the smart sensor camera has to work harder to do more calculations.



Some IR cameras lets you choose the mode that they will operate in – image streaming mode or smart sensor mode – such as the FLIR A310, for instance. In FLIR Tools (see chapter FLIR Tools on page 210) you can choose to stream in video – compressed format, or in signal – radiometric format, with the FLIR A310.



Signal or Video option for the FLIR A310 in FLIR Tools.

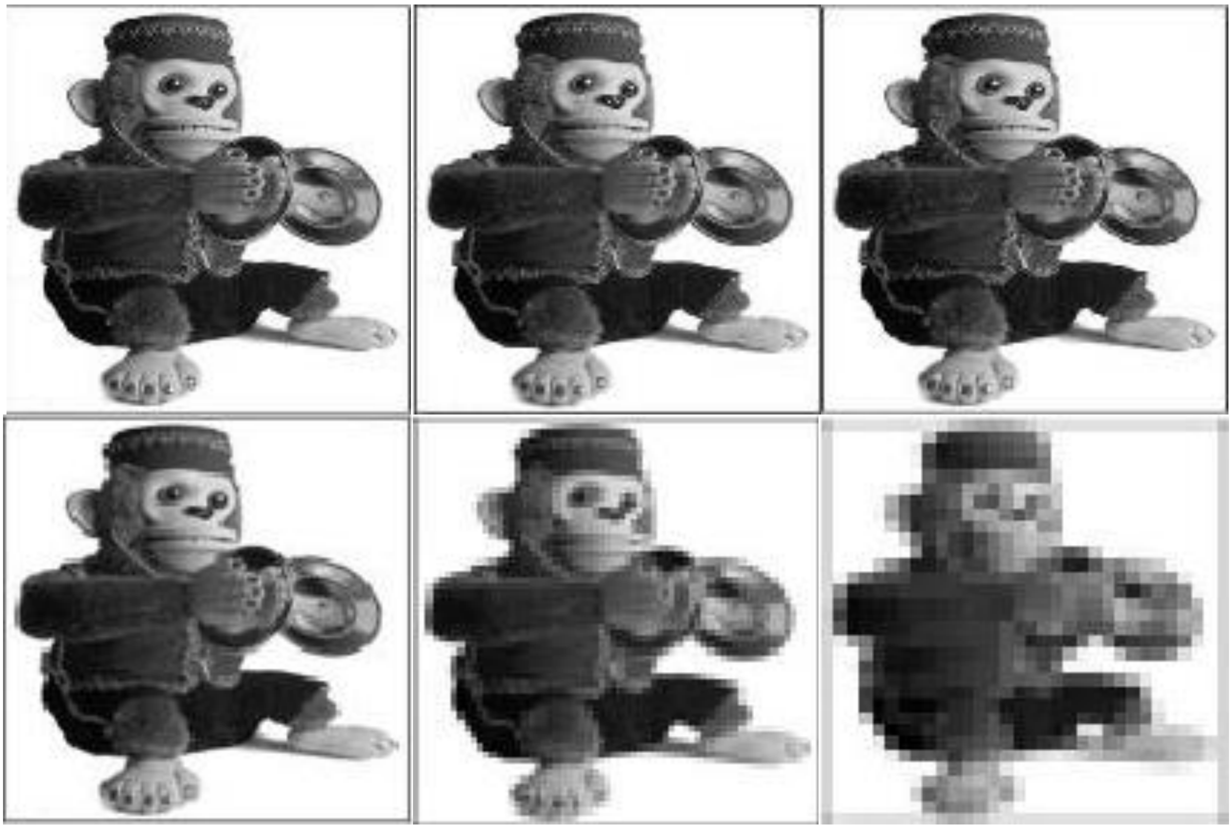
Sampling

We've looked at the IR resolution of your IR camera – that is, the number of pixels that can measure a temperature, and subsequently give us a color in the IR image. If you have an IR resolution of 320 x 240, for instance – the resulting image or video might be in another format – or size. The change of image size is the result of sampling. The sampling up or sampling down may be just so that the image matches the program you're viewing it in.

I will not discuss sampling further or how it is performed, but I will show you an example to illustrate how it may look.

These images of a monkey are of different pixel sizes. Ranging from largest to smallest, the pixel sizes are: 1024 x 1024, 512 x 512, 256 x 256, 128 x 128, 64 x 64, and 32 x 32.

If we were to resize the images, making them the same size – 1024 x 1024 – it would look like this.



The image in the top left corner remains in size, while all the other images have been upsampled from their earlier pixel sizes to 1024 x 1024.

SAQ

We have now reached an SAQ – self-assessment question. Here we take a moment to pause and reflect on what we have covered up until now. In this SAQ, we will cover the differences between image streaming cameras and smart sensor cameras.

Your job is to identify which features belong to which type of IR camera.

Which features describes image streaming cameras and smart sensor cameras? Check the boxes that apply. Once you are done, turn to the next page for my response and feedback.

	Image streaming camera	**Smart sensor camera**
Built–in analysis		
Able to operate without connected computer		
Compressed streaming		
Radiometric streaming		
Temp. linear		
A captured IR image contains radiometric data		

SAQ

The table below contains the correct answers. More importantly, I have added a response to **why** each answer is correct. Let's go through them one by one.

	Image streaming camera	Smart sensor camera	Response
Built–in analysis		X	This is one of the trademarks for smart sensor cameras, they not only capture the scene but is also able to **perform analysis** of the IR image.
Able to operate without a connected computer		X	We just said that smart sensor cameras have built-in analysis. That's because the camera has a **built-in computer** that performs the analysis. As long as smart sensor cameras are supplied with power, they can operate without a connected computer. On the other hand, image streaming cameras only **stream** the data (remember the matrices of numbers representing the raw data from an image streaming camera). This means that they need to be **connected to a computer** in order to perform analysis and for alarms to be set up.
Radiometric streaming	X		Radiometric data is information about the **temperature** recorded for every pixel in the IR camera. This is the information that image streaming cameras send to the connected computer, which then analyzes the IR images.
Compressed streaming		X	Unlike image streaming cameras which streams radiometric data, smart sensor cameras only send **color information** in every pixel. In other words, we see the IR image, but the stream doesn't give us any temperature information.
Temp. linear	X		Temp. linear is another streaming format that can be selected for image streaming cameras. The idea is that the information in every pixel value – a number – can be multiplied by a factor to give the temperature.
A captured IR image contains radiometric data	X	X	We have already said that image streaming cameras stream radiometric data to its connected computer. But why is smart sensor camera also the correct answer? Didn't smart sensor cameras stream in compressed mode? We need to distinguish between a **stream** and a **captured image**. Smart sensor cameras cannot stream radiometric data. However, when we capture an image of the scene the image will also contain radiometric data, which can be analyzed afterwards in a program such as FLIR Tools.

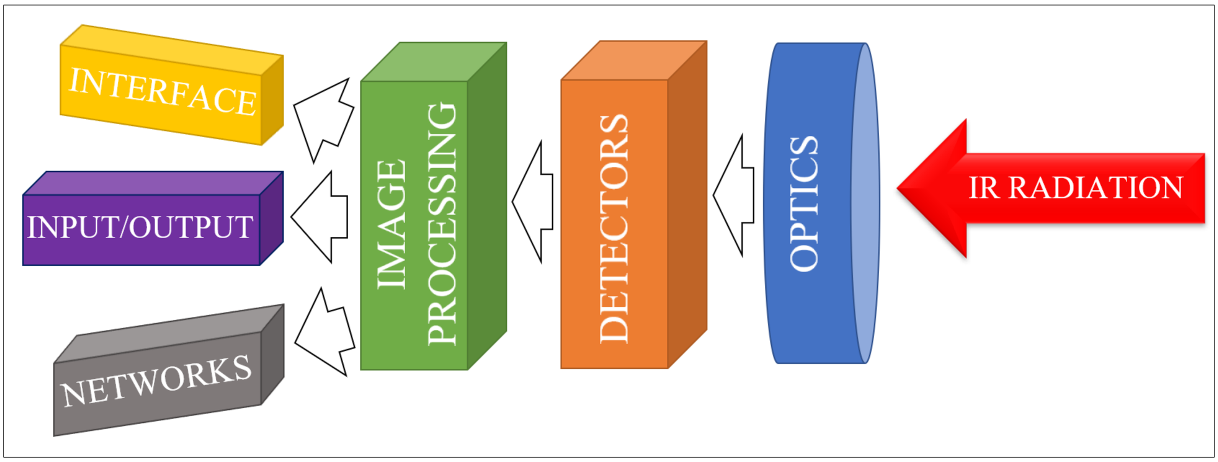
If you got these answers right, excellent! If not, excellent! Being wrong is an important part of learning, and the fact that you have worked through this SAQ is fantastic.

SUMMARY

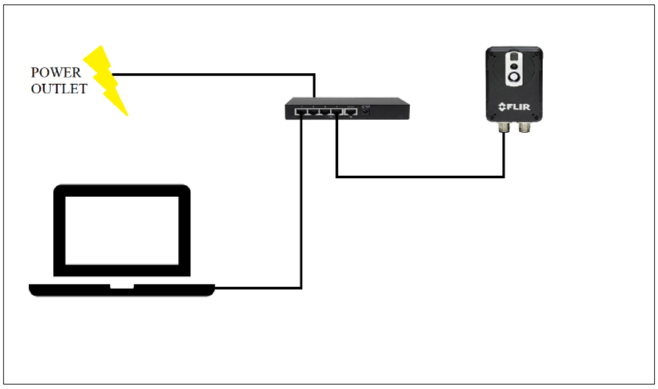
In this part, we have discussed the image processing of an IR camera. I will give you a list with the main parts that we've covered. When you go through the list, try to reflect over what you have read.

> The bit depth tells us how many colors there are in an image, an example being 8-bit image that can have 2^8=256 different colors.
>
> Image streaming cameras does not have a built-in computer and can therefore not lay out analysis functions or alarms by themselves. This is done in a connected computer.
>
> Image streaming cameras stream in radiometric mode, which means that every pixel contains measurement information.
>
> Histogram equalization gives more contrast in the image.
>
> Smart sensor cameras have a built-in computer which allow them to lay out analysis and alarms. When alarms are set, the smart sensor cameras can operate without a computer connected.
>
> Smart sensor cameras stream in compressed format, which means that the pixels are only assigned a color and not measurement information. When an image is captured, the format will however be radiometric.
>
> The analysis functions of a smart sensor camera are slower than the image stream.
>
> Sampling of images may lead to images looking pixelated.

Networks



This chapter is about networks. We've exited the IR camera system and are now were the information from the IR camera is communicated to other devices – like a computer.



Our first question should then be: *What is a network?* Simply put, a network is a group of computers or other devices that communicate with each other. These devices can be connected by wire or wirelessly. A simple but important example of a network is a computer and an IR camera. We can also regard the Internet as a network, with an enormous number of connected devices.
Regardless of the number of connected devices, the basic principles of network communication are the same. How do we make these devices communicate with each other? In this chapter, we will cover some of the important pieces needed for devices to be able to communicate in a network.

Objectives

When you have worked through this part, my aim is that you will be able to answer the following questions.

What is an IP address?

How does the subnet mask determine the length of the Network ID and Host ID?

What is the purpose of a gateway?

How do ports work together with IP addresses?

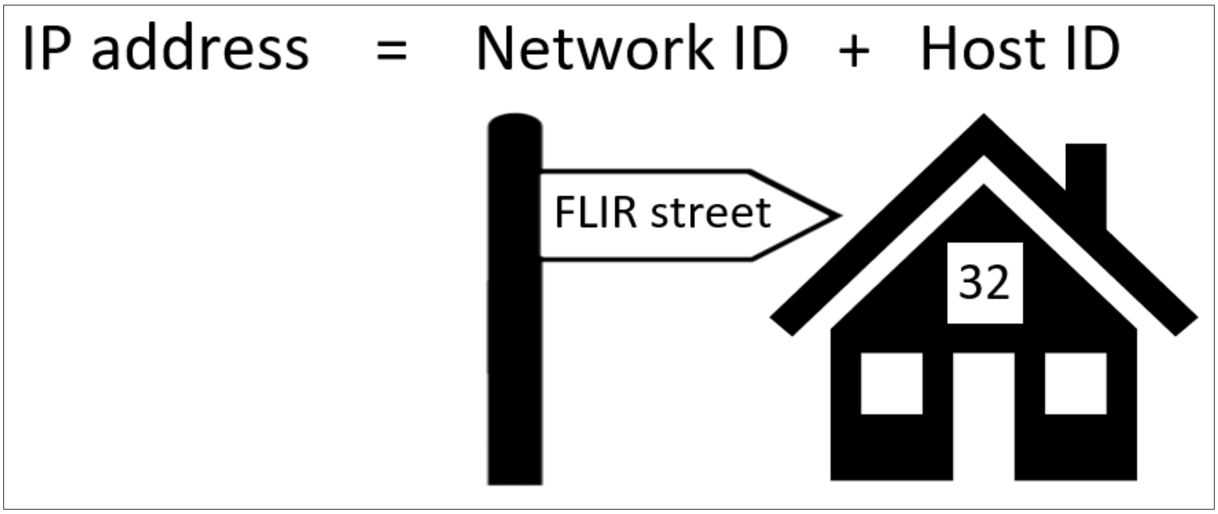
For you to know how to answer these questions, I have divided this chapter into four parts:

IP Address
Subnet Mask
Gateway
Port

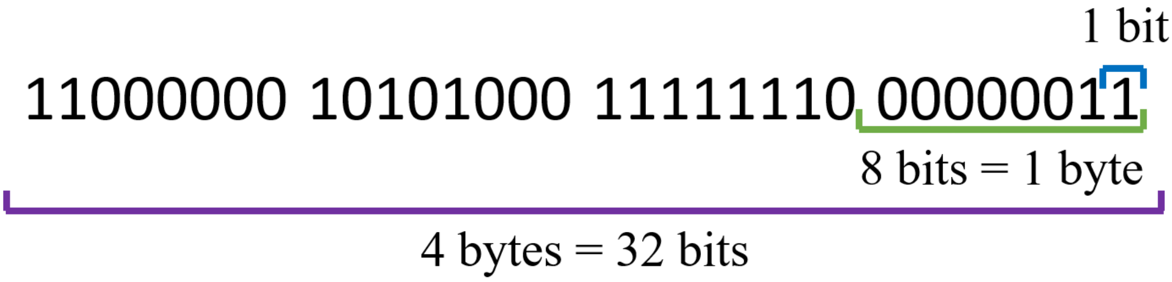
IP address

As I mentioned, a network is a group of devices that communicate with each other. For this communication to work, we need a system of identification in order to distinguish one device from another. This is precisely what IP address are – a labeling system for devices. A similar question we could ask is: why do we have house addresses? Answer: so that people can reach us. It is the same with IP addresses. We make devices reachable in a network by assigning them unique IP addresses.

Just as house addresses, IP addresses consist of two parts: a street name and a house number. In an IP address, the street name is called **network ID** and the house number is called **host ID**. You can think of all devices in a network as neighbors. They all live on the *same street*, but they have *different house numbers*. This means that all devices on the same network must have the *same network ID*, but *different host ID*.



At the very basic level, computers communicate with each other by sending a lot of ones and zeros. They use what we call the **binary** numeral system – or **base-2** system. To a computer, an IP address is simply 32 digits of ones and zeros. A digit which is either 1 or 0 is called a **bit**. Thus, an IP address is comprised of 32 bits. 8 bits is called a **byte**. We can also say that an IP address is comprised of 4 bytes.



Although computers are really good at working with binary numbers, humans are typically bad at it. To solve this problem, we split the IP address into four parts, convert from binary to decimal and put dots in between. This is so we get a nice and readable format, which we call **dotted decimal format.**

I hope you agree with me when I say that 192.168.254.3 is a lot easier to read than 11000000101010001111111000000011. Believe it or not, these two formats represent the *same* IP address.

IP address	11000000101010001111111000000011			
Separate into 4 fields	11000000	10101000	11111110	00000011
Convert into decimal	192	168	254	3
Add dots for readability	192.168.254.3			

As you see in the table above, an IP address in dotted decimal notation consists of four fields. Each field has a number ranging from 0 to 255. This is because the largest number we can create using 8 bits is 11111111 in binary, which is 255 in decimal notation. The smallest number we can create using 8 bits is 00000000, which is 0 in decimal notation.

IPv4

There are different versions of IP addresses and the version we discuss in this chapter is called **IPv4**, which stands for *Internet Protocol version 4*. IPv4 is the most commonly used version and when someone says "IP address", that person will most likely refer to an IPv4 address. A newer version that is becoming more popular is called IPv6 – *Internet Protocol version 6*. It has the benefit of having many more available addresses than IPv4. However, I will only discuss IPv4 addresses in this chapter.

By now, we have said that an IP address is a collection of ones and zeros which uniquely labels a computer and distinguishes it from every other device in the network. We have also said that an IP address consists of a network ID and a host ID. But what part of the IP address is the network ID and what part is the host ID? We will address this question in the next section on subnet masks.

Subnet mask

As I mentioned earlier, the IP address is comprised of two parts: the network ID and the host ID. Devices in the same network can be thought of as neighbors – they live on the same street, but not in the same house. Thus, their IP addresses all share the **same network ID** (street name) but have **different host ID** (house number).

To answer the question posed earlier "What part of the IP address is the network ID and what part is the host ID?", we look at the **subnet mask** of the network. The subnet mask defines what part of the IP address is the network ID, and what part of the IP address is the host ID.

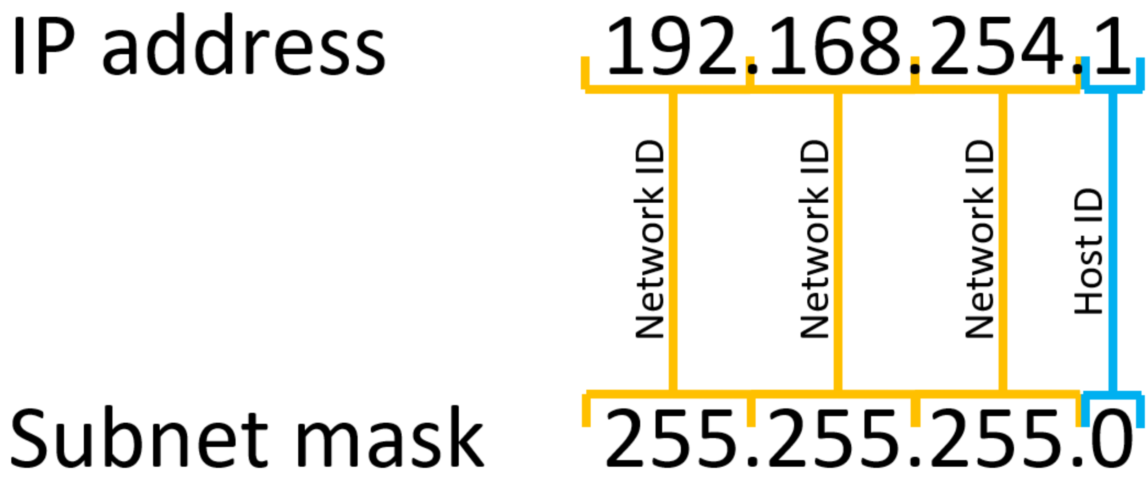
In a small network with just a couple of connected devices, we do not need a lot of IP addresses to make sure every device is assigned a unique IP address. In this case, we can set the subnet mask so that the network ID (street name) is a large part of the IP address and the host ID (house number) is a small part. However, in a larger network we need to make the host ID part of the IP address bigger and the network ID part smaller. This to ensure that there are enough available IP addresses for the devices in the network.

A street in New York might need more house numbers than a street in Ljungbyholm, Sweden.
Images from Google earth, Google/Digital Globe.

The length of the network ID - A simple rule

Similar to the IP address, the subnet mask is also written in **dotted decimal notation** – four numbers separated by dots. And in many cases the fields in the subnet mask will either be 255 or 0.

In that case, there is an easy rule: The **255s** in the subnet mask correspond to the **network ID (street name)**. The **0s** in the subnet mask correspond to the **host ID (house number)**.



The network ID and host ID street names and house numbers are also written in **dotted decimal notation,** with four number fields. This means that in the figure above, 1 is not actually the house number, but 0.0.0.1 is. Similarly, 192.168.254 is not the street name, but 192.168.254.0 is. The table below shows some more examples of IP addresses and how the subnet masks determine the network ID and host ID.

IP address	Subnet mask	Network ID	Host ID
192.168.254.1	255.0.0.0	192.0.0.0	0.168.254.1
169.254.100.2	255.255.0.0	169.254.0.0	0.0.100.2
213.100.31.47	255.255.255.0	213.100.31.0	0.0.0.47

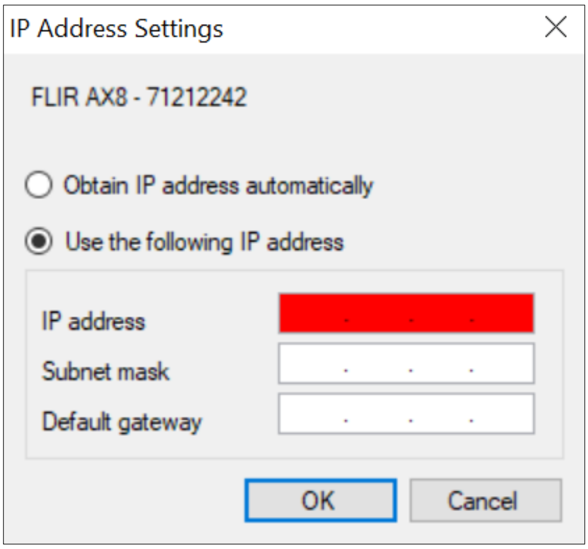
Can the subnet mask consist of numbers other than 255 and 0?

Yes, the subnet mask may be 255.255.252.0 for example. This makes things a little trickier, since the network ID ends somewhere *inside* the third field. To figure this out, we need to use binary numbers instead of decimal numbers. I give the whole picture of subnet masks with the help of binary numbers in the More on binary numbers section below.

Configure a valid IP address for your IR camera.

Let’s put this knowledge to good use. One important and practical use of your knowledge on IP addresses and subnet masks is to make sure that your IR camera is properly configured to communicate with your computer.

To find connected IR cameras in a network and assign them IP addresses, we use the program **FLIR IP Config**. For more detailed information on how to install and use FLIR IP config, see the chapter Step 1: Download and install FLIR IP Config (page 11). There, we also go through how to find the IP address of your own computer.



Let us assume that your computer has the IP address **169.254.200.10** and the subnet mask of the network is **255.255.0.0**. To configure your IR camera in the network, FLIR IP config will ask you for three pieces of information: IP address, subnet mask and default gateway. Below, I will go through what to put in each row.

IP address

Remember that an IP address consists of two parts: network ID (street name) and host ID (house number). A valid IP address for the IR camera has the **same street name** (network ID) as the computer, but **different house number** (host ID).

Since the computer has the IP address is 169.254.200.10 and the subnet mask is 255.255.0.0, the network ID is 169.254.0.0 and the host ID is 0.0.200.10. The host ID of the IR camera must differ in **at least one** of the two last fields. 0.0.200.11 is a valid host ID, as well as 0.0.201.11.

An example of a valid IP address for the IR camera is therefore: **169.254.200.11.** You may change both of the last two fields but changing one of them works just fine.

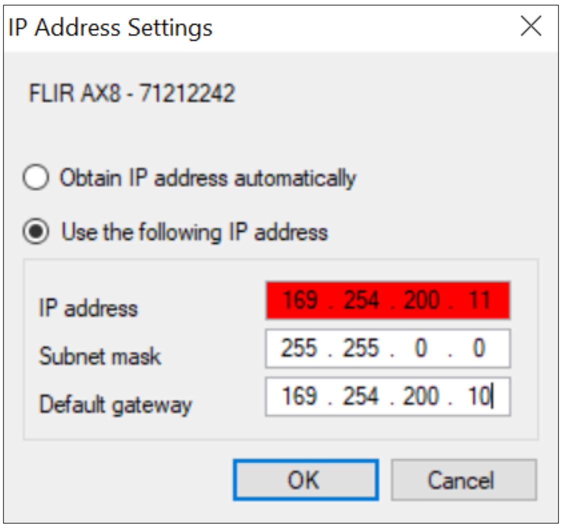
Subnet mask

Since the IR camera is on the same network as the computer, we simply put in the subnet mask of the current networks. In our case, we put in **255.255.0.0.**

Default gateway

In the section below, we will cover what gateways are more in depth. In short, a gateway connects two networks with each other and controls the flow of data. A router is a good example of a gateway. The *default gateway* is the device that data is sent to if no recipient of the data is specified. The default gateway is typically the network's router.

In the simple network of a computer and an IR camera, we technically do not have a default gateway. However, FLIR IP config demands us to put *something* in. Since the IR camera is only sending data directly to the computer, we put the default gateway to be the IP address of the computer. In this case, **169.254.200.10**.



In the end we have the following result (showed in the image to the right).

Box tip – Reserved IP addresses

It is good to know that some IP addresses are often reserved for specific purposes. Then you can avoid the mistake of setting the IP address of your IR camera to any of these reserved IP addresses.

Let's say our network has the subnet mask 255.255.255.0 and the street name is 192.168.254.0. Theoretically, we should have 256 available addresses, ranging from 192.168.254.0 to 192.168.254.255. In reality, the number of available IP addresses is smaller.

The *lowest* IP address in a network – 192.168.254.0 in our case – is called the **network address**. It is used as an *identifier* for the network and is not a valid IP address for any device in the network.

The *highest* IP address in a network – 192.168.254.255 in our case – is called the **broadcast address**. It is used to send information to *all devices* on the network and is not a valid IP address for any device on the network.

The *lowest available* IP address – 192.168.254.1 in our case – is *often* the IP address of the **router**. This is not always true. To check the IP address of *your* router, **type** `cmd` in your computer search menu and **press** `Enter`. In the Command Prompt, type `ipconfig` and **press** `Enter`. **Find** the *Default Gateway IP address*.

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : FLIR-Guest
   Link-local IPv6 Address . . . . . : fe80::10d7:3976:78f0:2e59%14
   IPv4 Address. . . . . . . . . . . : 10.64.21.131
   Subnet Mask . . . . . . . . . . . : 255.255.252.0
   Default Gateway . . . . . . . . . : 10.64.20.1
```

SAQ

Valid IP addresses for your IR camera

Your computer has the following network information:

IP address (IPv4):	169.254.100.10
Subnet Mask:	255.255.0.0

You want to correctly configure your camera.

Which (one or more) of following configures are correct?

a) IP address: 169.254.100.10
   Subnet mask: 255.255.0.0
   Default network: 169.254.100.10

b) IP address: 169.254.101.11
   Subnet mask: 255.255.255.0
   Default network: 169.254.100.10

c) IP address: 169.254.110.99
   Subnet mask: 255.255.0.0
   Default network: 169.254.100.10

d) IP address: 169.254.100.11
   Subnet mask: 255.255.0.0
   Default network: 169.254.100.10

SAQ

The correct answer configures are **c)** and **d)**, but why?

The computer had the following network information:

IP address (IPv4):	169.254.100.10
Subnet Mask:	255.255.0.0

Let's look at the alternatives one at a time.

a)	**IP address:**	**169.254.100.10**
	Subnet mask:	255.255.0.0
	Default network:	169.254.100.10

Notice that the IP address of the camera is exactly the same as that of the computer. This will not properly configure the camera. Remember, the rule is: **same network ID** (street name), **different host ID** (house number).

b)	IP address:	169.254.101.11
	Subnet mask:	**255.255.255.0**
	Default network:	169.254.100.10

The subnet mask does not match the computer's, these have to match in order for the camera to be properly configured. Otherwise the network ID length of the camera would be different than the computer's, and that is bad.

c)	**IP address:**	**169.254.110.99**
	Subnet mask:	**255.255.0.0**
	Default network:	169.254.100.10

The Network ID of the camera (169.254.0.0) is the same as the computer's and the host ID (0.0.110.99) is different. The subnet mask also matches the computer. This combined means that the camera is properly configured.

d) **IP address:** **169.254.100.11**
Subnet mask: 255.255.0.0
Default network: 169.254.100.10

This is also a correct configure! The rule is: **same network ID** (street name), **different host ID** (house number). This means that **at least one** of the two last fields need to be different. This combined with a matching network ID and subnet mask means that the configuration is good.

If you chose c) and d), great job! If not, remember that being wrong is a key part in learning. The fact that you have worked through this SAQ and given it time is great!

New notation – CIDR

We have now covered that IP addresses are used to uniquely label devices in a network. The IP address consists of a network ID and a host ID. The subnet mask tells us what portion of the IP address constitutes the network ID and host ID respectively. The notation of using a subnet mask (255.255.255.0, for instance) is actually an old notation, although it is still used frequently. It is good to know about the new notation. It is called **CIDR** notation, which stand for Classless Inter-Domain Routing. What the acronym stands for is not really that important for us, more important is the how notation works.

The CIDR notation uses a slash (/) after the IP address to denote the length of the network ID. For example, **192.168.254.2/24**. The /24 refers to the size of the network ID in **number of bits**.

Recall that an IP address is made up of 32 bits – a row of 32 zeros and ones. These 32 bits are divided into four fields of 8 bits each and separated by dots. In the example above, we had **192.168.254.2/24**. Since we have a /24, it means that the first 24 bits in the IP address is the network ID. This means that the 8 remaining bits in the IP address is the host ID.

The first 24 bits corresponds precisely to the first three fields in the IP address – 192.168.254. From that, we know that the network ID is 192.168.254.0 and the host ID is 0.0.0.2. Therefore, /24 in the new notation is equivalent to 255.255.255.0 in the old notation.

The table below shows a few examples that connects the old notation (subnet mask) to the new notation (CIDR).

Old notation (Subnet mask)		New notation (CIDR)
IP address	Subnet mask	
192.168.254.1	255.0.0.0	192.168.254.1/8
169.254.100.2	255.255.0.0	169.254.100.2/16
213.100.31.47	255.255.255.0	213.100.31.47/24

Can we have values other than /8, /16 or /24?

Yes, we might have a 22-bit network ID, denoted by /22, or any other whole number between 0 and 32. This is the same problem as before, when the subnet mask contained numbers other than 255 or 0. As I mentioned before, things get a little trickier and we need to involve binary numbers. If you want the whole picture, see the More on binary numbers section below. If you feel satisfied with what we have covered so far, feel free to skip that section.

More on binary numbers

In this section, I want to go a little deeper into how subnet masks and CIDR notation work. I will do this by making a stronger connection to **binary numbers**. In the **decimal system** we use the digits 0, 1, 2, … ,9 and the position of the digits are based on **powers of 10**. For example, we think of the number 169 as

$$169 = 100 + 60 + 9 = 1 \cdot 10^2 + 6 \cdot 10^1 + 9 \cdot 10^0.$$

In the binary system, we only use the digits 0 and 1 and the position of the digits are based on **powers of 2**. For example, we interpret the binary number 10101001 as:

$$10101001 \ = \mathbf{1} \cdot 2^7 + \mathbf{0} \cdot 2^6 + \mathbf{1} \cdot 2^5 + \mathbf{0} \cdot 2^4 + \mathbf{1} \cdot 2^3 + \mathbf{0} \cdot 2^2 + \mathbf{0} \cdot 2^1 + \mathbf{1} \cdot 2^0 =$$

$$= 2^7 + 2^5 + 2^3 + 2^0 =$$

$$= 128 + 32 + 8 + 1 = 169.$$

You see that the decimal number 169 and the binary number 10101001 represent the *same quantity*, but with *different notation*. In binary, you can think of the 1s as indicating *on* – these are the powers of 2 we want in our number. Similarly, the 0s indicate *off* – these are the powers of 2 we don't want in our number.

We have said that an IP address consists of 32 bits, and a bit is a digit which is either 0 or 1. In other words, an IP address is simply a **32-digit binary number**. To make it more readable, the IP address is separated into four parts – each part with 8 bits is called an **octet**. Because of this separation, knowing the first eight powers of 2 makes everything a lot easier.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Subnet mask

Now that we have introduced binary numbers more thoroughly, let me be more precise about *how* the length of the network ID is determined.

The **rule** is: The subnet mask is a **32-digit binary number** with a row of **only 1s** followed by a row of **only 0s**. The 1s correspond to the network ID and the 0s correspond to the host ID.

Example: Find out the network ID and host ID from an IP address

Your computer has the IP address **169.254.103.2** and the network has the subnet mask **255.255.252.0**. What are the network ID and host ID of the IP address?

First let's convert the IP address and subnet mask into binary. Let's make use of our chart of powers of two.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

$169 = 128 + 32 + 8 + 1$. For the binary number, we want a 1 in these positions and 0 in the rest. Thus, 169 is **10101001** in binary.

$254 = 128 + 64 + 32 + 16 + 8 + 4 + 2$. We want a 1 in all positions except the last. Thus, 254 is **11111110** in binary.

$103 = 64 + 32 + 4 + 2 + 1$. Thus, 100 is **01100111** in binary. Note that we *do not* get rid of the leading zero, because the IP address is a 32-bit number.

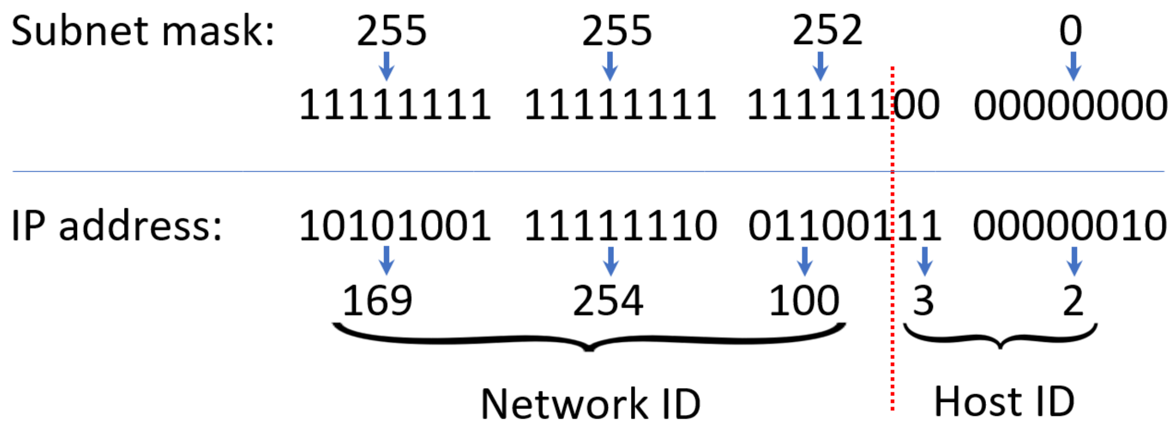
2 only has a 1 in the 2-position. Thus, 2 is **00000010** in binary. Again, we don't get rid of any leading zeros, because the IP address is a 32-bit number.

Now to the subnet mask. 255 is the biggest 8-bit number we can make, so it consists of all 1s. Thus, 255 is **11111111** in binary.

$252 = 128 + 64 + 32 + 16 + 8 + 4$. Thus, 252 is **11111100** in binary.

0 is the smallest number we can make. Thus, 0 is **00000000** in binary.

We now have the following IP address and subnet mask in binary:



As our rule above states, the leading ones in the subnet mask indicate the Network ID in the IP address. We see that the third octet is split into a network ID part and a host ID part.

Now we know that 100 out of the 103 in the third octet belongs to the network ID and 3 belongs to the host ID. We now have our answer. The IP address 169.254.103.2 with subnet mask 255.255.252.0 has the **network ID 169.254.100.0** and the **host ID 0.0.3.2**.

CIDR notation

As I mentioned earlier, the CIDR notation (/16) is a new alternative to subnet masks (255.255.0.0). It uses a slash (/) to denote the length of the network ID. More precisely, the number after the slash (/) denotes the length of the network ID in **number of bits**. Now that we have worked some with binary numbers, let me show you how to convert between subnet mask and CIDR notation.

Example: Convert from subnet mask to CIDR notation

In the example above, our network had the subnet mask 255.255.252.0. What is this in CIDR notation?

We've already converted the subnet mask to binary and obtained that 255.255.252.0 is 11111111.11111111.11111100.00000000 in binary. We count the number of leading 1s, which is 22. Thus, the **CIDR notation is /22**.

Example: Convert from CIDR notation to subnet mask

Let's assume your network is denoted by /20, what is the subnet mask?

The /20 means that the subnet mask in binary has 20 leading 1s, followed by 0s.

This is 11111111.11111111.11110000.00000000.

Now, we have to convert this into decimal. As we have seen before, 11111111 in binary is 255 in decimal and 00000000 in binary is 0 in decimal. The binary number 11110000 is 240 in decimal, because

$$11110000_2 = 128 + 64 + 32 + 16 = 240.$$

Thus, /20 in CIDR notation is equivalent to the **subnet mask 255.255.240.0**.

SAQ

True or False?

The subnet mask 255.255.255.128 is the same as /25 in CIDR notation.	True	False
255.255.0.1 is a valid subnet mask.	True	False
/18 corresponds to the subnet mask 255.255.224.0.	True	False
/21 means that the host ID of the IP address is 21 bits long.	True	False

SAQ

The first statement is true, and the rest are false, but why? Let's go through the statements one by one.

The subnet mask 255.255.255.128 is the same as /25 in CIDR notation.	**True**	False

- Let's convert the subnet mask into binary. We then have 11111111.11111111.11111111.10000000. Now we count the number of leading ones, which is 25. This means that /25 corresponds to the subnet mask 255.255.255.128 – the statement is true.

255.255.0.1 is a valid subnet mask.	True	**False**

- Remember that the subnet begins with **only ones** and ends with **only zeros** when we express it in binary. This is so we know the length of the network ID and host ID of the IP address. The number of ones is the length of the network ID and the number of zeros if the length of the host ID. However, 255.255.0.1 in binary is 11111111.11111111.00000000.00000001. The one at the end is the problem. Thus, 255.255.0.1 cannot be a valid subnet mask.

/18 corresponds to the subnet mask 255.255.224.0.	True	**False**

- /18 tells us that the subnet mask has 18 leading ones followed by only zeros. This becomes 11111111.11111111.11000000.00000000 in binary. Now let's convert this binary number to decimal. 11111111 in binary is 255 in decimal, 11000000 in binary is 128+64=192 in binary and 00000000 in binary is simply 0 in decimal. We now get the subnet mask 255.255.192.0. Thus, the statement is false.

/21 means that the host ID of the IP address is 21 bits long.	True	**False**

- Here we need keep track of the terms. The **first part** of the IP address is the **network ID** and the **last part** is the **host ID**. /21 refers to the length of the first part – the network ID. **Not** the host ID, which is why the statement is false.

If you got these statements correct, fantastic. If not, don't worry about. Being wrong is an important part of learning and the fact that you have gone through this whole SAQ is fantastic. I hope that my responses to each statement have assisted you in your thought process.

Gateway

Up until this point, we have discussed networks mostly from a local perspective. For example, a computer and an IR camera connected to each other in a local network. However, we also want to connect to other networks, such as the Internet. This is what **gateways** are used for. As the name suggests, you can think of a gateway as a **gate** between two networks. In order for information to pass from one network to another, it must pass through the gateway. The gateway controls the flow of information coming in and out of the network. One important example of a gateway is a **router**, and we will talk more about the role of the router in a concrete example below.

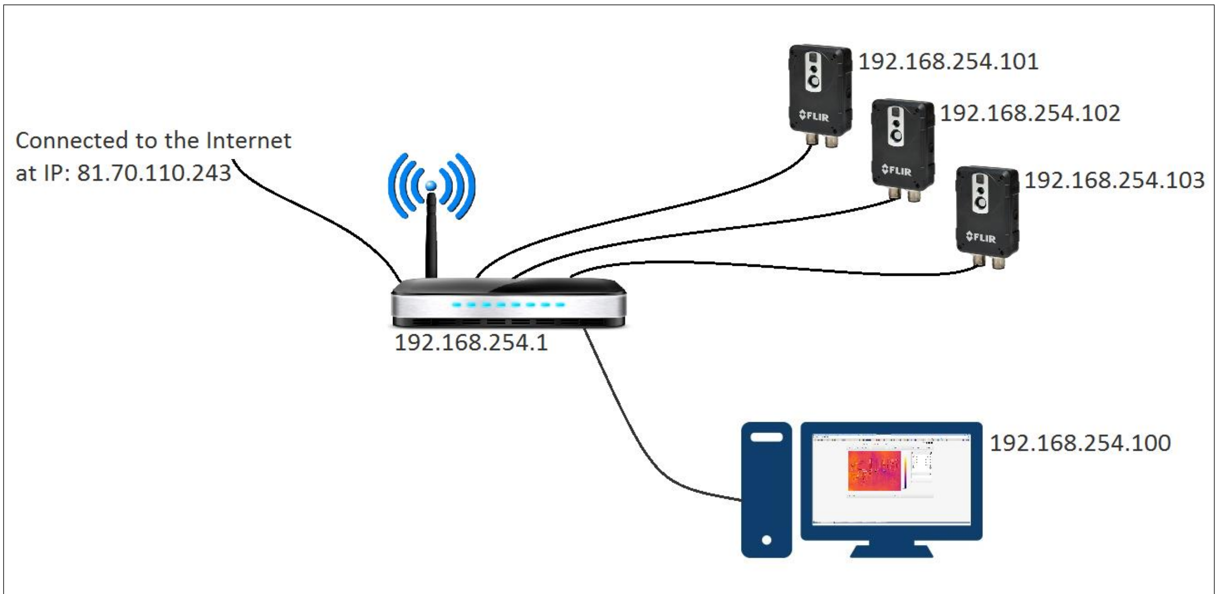
A concrete example

Let us look at a concrete example to hopefully make things clearer. The question below was posted and answered on the FLIR support site. I will use this question to illustrate how IP addresses, subnet masks, gateways and ports can be used in practice.

> *How do I set up my network so that I can access my FLIR AX8 from outside my local network, for example when I'm connected to the Internet at a distant location from the camera?*

If you wish to read the answer in its entirety, **go to** `https://flir.custhelp.com`. **Type** *`AX8 internet access`* in the search field and **select** the page named *`Configuring FLIR AX8 access from the Internet to a home/local network.`*

To understand and answer the question, we will analyze and go through the network situation given in the figure below.



We have a local network composed of a computer, a router and three FLIR AX8, each with its uniquely assigned IP address. These five devices represent our **local network**. We want to know how to access the web interface of a FLIR AX8 from *outside* the local network, via the Internet.

Note that these five devices all have the same first three fields in their IP addresses – 192.168.254. Then we can deduce that these numbers represent the network ID of the IP addresses. This also means that the host ID is given by the forth field in the IP addresses. Another way of saying this is that the **subnet mask** of the network is 255.255.255.0 – or /24 in **CIDR** notation.

But what is the IP address in the top left of the figure – 81.70.110.243? This is where the **gateway** comes in. In this case, the gateway is a **router**. The router is the *link* to the Internet. It allows the devices in our network to access devices and applications outside the local network. In other words, it allows us to send and receive data via the Internet.

On the Internet however, the whole network is represented by **only one** IP address – 81.70.110.243. This IP address is called a **public address**, as it is the only IP address that other devices connected to the Internet can see. Let's say you send an email over the Internet from the computer 192.168.254.100. Then the receiver will register the mail coming from the public address 81.70.110.243, *not* 192.168.254.100. The IP addresses beginning with 192.169.254 in our local network are called **local addresses**, because they are only visible to the devices inside the local network.

But hang on, we want to access a FLIR AX8 remotely via the Internet. How do we access a *specific camera* if the whole network and all its devices are represented by *only one* IP address? The answer is with the use of **ports**, which we will cover next.

Port

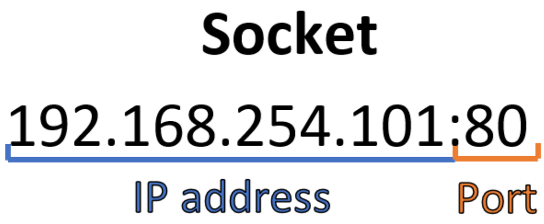
In computer networking, the term **port** has several meanings. It can refer to a physical endpoint between devices. For example, you might have an **USB port** on your computer to which you can connect a mouse or another device.

However, a port also refers to a way of **identifying a specific network process** – not a physical port. It is this meaning of the word *port* that we will focus on. We have said that a network consists of devices that communicate with each other. In order to communicate, the devices need **rules** and these rules are called **protocols**. I dedicate a whole chapter to protocols (see chapter Protocols). For now, just remember that protocols are the rules which specify how devices should communicate with each other.

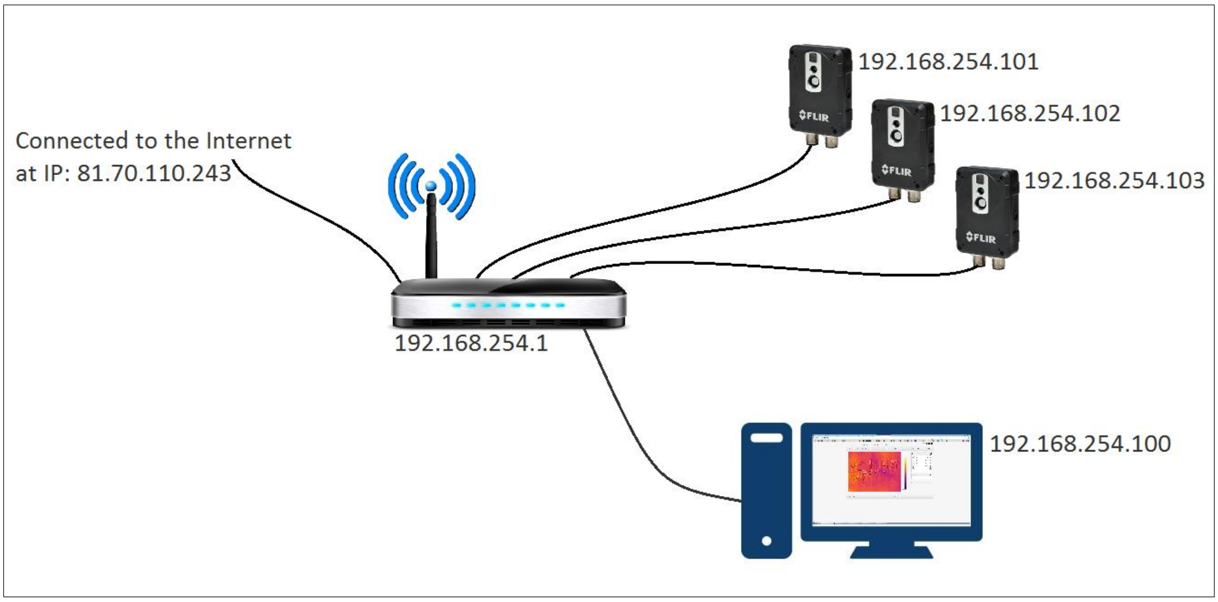
Let us assume that you have a computer connected to the Internet. Then you may have many processes running at the same time – browsing the web, sending email or watching videos online. All these processes require access to the Internet, which means that a lot of data is constantly sent through the network. To structure and manage this constant flow of information, each process is assigned its own **port** and **port number**. Processes or applications that are widely used have been given specific port numbers. For example, email programs always use port 25 and web browsers use port 80.

Let us return to the analogy of a network as a street and the devices in the network as houses on that street with their individual house numbers. Now, each house has several rooms. In our analogy, every room is assigned a service of some sort – the kitchen is for cooking, the dining room is for dining, and so on. Say that you are away from home and want to go to sleep. You then need to locate your street, your house and then your bedroom to accomplish your task.

We've already looked at the network ID as the street name, and the host ID as the house number. To continue the analogy, the port number would then be the *room number*. In other words, the **IP address** specifies a certain *device* in a network and the **port** specifies a certain *process* in that device. An IP address together with a port is called a **socket**, and it is written like this: <IP address:port>.



Now that we have said a bit about ports in general, let us return to the example of how to access the FLIR AX8 web interface remotely via the Internet.



The problem is that all devices outside our local network only see the public IP address 81.70.110.243. The public address represents the *whole* local network, how do we then access the web interface of a *specific* FLIR AX8? We use **port forwarding**.

Port forwarding

When you access the web interface of a FLIR AX8, you normally type in the local IP address of the camera in your web browser. In order for this to work, your web browser is using a protocol – set of rules – called HTTP. HTTP uses the port 80, and this a **local port**. Local ports are similar to local IP addresses, they are accessible *inside* the local network. Similar to public IP addresses which are visible outside the local network, we also have ports that are visible outside the local network. These ports are called **public ports**.

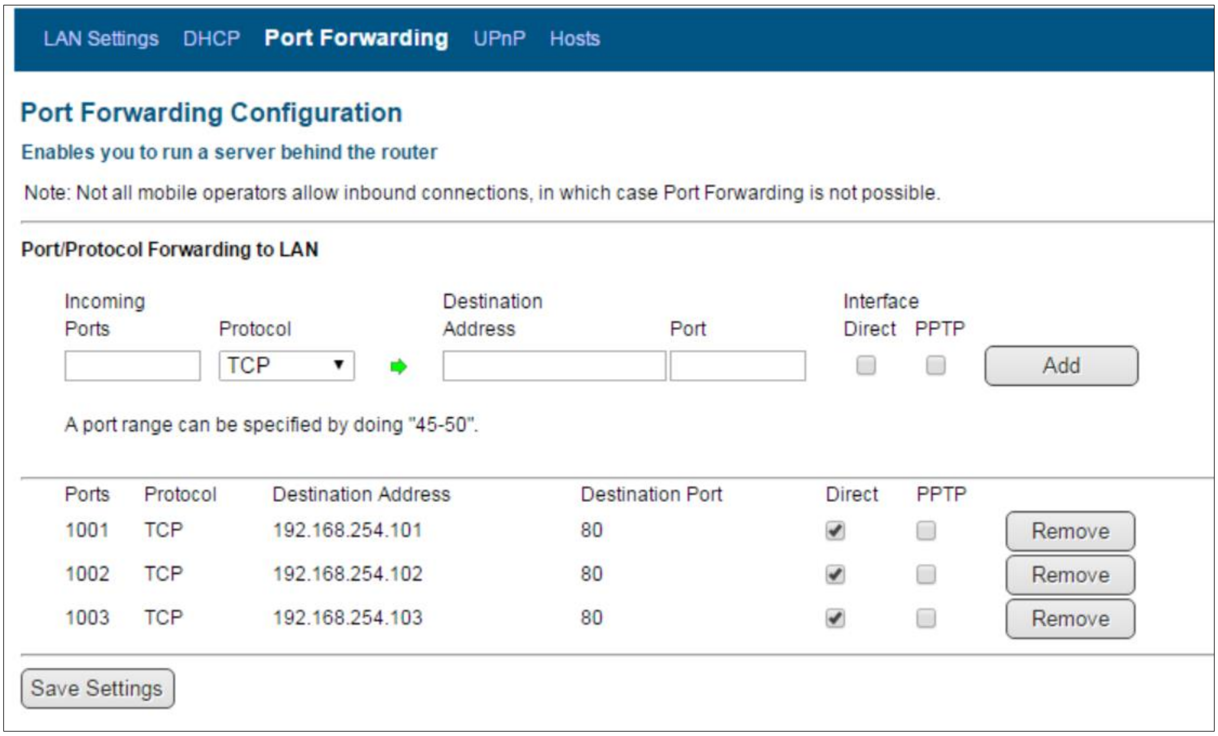
To access the IR cameras from outside the local network, we need to **link** the public IP address and public ports to the local IP addresses and local ports. This is called **port forwarding**, as shown in the table below. The port forwarding is configured in the **router**, since this is the gateway that regulates the flow of data going in and out of the network. To configure the router, we need to be connected to the **local network**. Once everything is set up, we can access the cameras from *outside* the local network.

Public IP address	Public port	Local IP address	Local port
81.70.110.243	1001	192.168.254.101	80
81.70.110.243	1002	192.168.254.102	80
81.70.110.243	1003	192.168.254.103	80

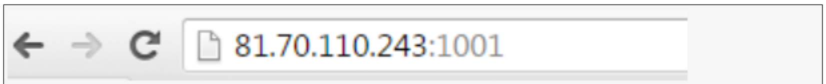
Note that in the table above, the local IP addresses correspond to the three IR cameras. Since the web browser uses the protocol HTTP when displaying the FLIR AX8 web interface – and HTTP uses port 80 – we define our local port to be 80. The public IP address is the same for all cameras since this is the only visible IP address to outsiders. However, note that the public ports are *different* – this is the key. The different public ports make the router forward us to different cameras.

The local port 80 must be 80 because this is the pre-defined port associated with the protocol HTTP. However, the public ports 1001, 1002 and 1003 are port numbers chosen by us. These can be any numbers you wish, as long as they are not the same as any pre-defined port, such as port 80 (HTTP) or port 25 (email). To avoid any collision with established ports, a good rule of thumb is to assign ports *above* 1000.

The image below shows how an interface may look when you configure the *port forwarding* in the **router**, but keep in mind that all router interfaces are different. We access the router by entering its local IP address – in this case 192.168.254.1 – in a web browser and log in with our credentials. We go to the **port forwarding tab** and define our public ports, local IP addresses and local ports.



Once the port forwarding is set up, we may access the IR cameras from outside the local network. The table below shows how to enter the different **sockets** (IP address + port) in the web browser. The router (gateway) will acknowledge our request and forward us to the web interface of one of the IR cameras.



URL	Camera
81.70.110.243:1001	*AX8* with local IP address 192.168.254.101
81.70.110.243:1002	*AX8* with local IP address 192.168.254.102
81.70.110.243:1003	*AX8* with local IP address 192.168.254.103

SUMMARY

In this chapter, we have discussed the concept of networks. I here give you a list of the main ideas that we've covered. When you go through the list, reflect over key points and connect it to what you have read.

> A network is a group of devices that communicate with each other.
>
> An IP address uniquely labels each device in the network.
>
> The IP address consist of a network ID (street name) and host ID (house number).
>
> The subnet mask defines the lengths of the network ID and host ID.
>
> A gateway controls the flow of data coming in and out of the network.
>
> A port describes a certain process taking place in the device.

Protocols

In this chapter, I will discuss **protocols**. You can think of protocols as *rules* that specify *how* and *when* computers or other devices should talk to each other. When people talk to each other, we often know how to converse by following social and cultural codes. Computers, however, don't have social codes. Therefore, they have to be programmed to follow specific rules on how to communicate with each other. The rules they follow are the protocols.

There are *a lot* of network protocols out there, and it can feel like a vast sea of abbreviations and acronyms. Below I will go through some common protocols and give a brief explanation of their use and connect to some applications when these protocols are used. I will also use some schematic figures to illustrate the basic functions of each protocol. My aim is that you will feel more confident with the basics of some protocols and don't feel like you are lost in a sea of acronyms. In the second half of this chapter I will focus on some important protocols used in the industrial arena, such as Modbus, EtherNet/IP, and GigE Vision.

Objectives

When you have worked through this part, I hope that you will feel more comfortable to

- Describe some common protocols, such as SMTP, FTP and RTSP
- Briefly explain the main ideas behind industry protocols, such as Modbus, EtherNet/IP and GigE Vision
- Set up notifications via email and FTP when an alarm is triggered

Some common protocols

Before we get into some common protocol, let me say something about *servers*, as there will be many references to servers. What is a server? The name itself gives a hint; a **server** is something that **provides a service.** The ones that are being served are called **clients**. Together these make up the **client-server model**.

One example of a computer application that uses the client-server model is *Email*. The email that you send and receive are stored on a **mail server**. When you wish to view your email, your **email client** – such as MS Outlook – retrieves your email from the mail server to your computer. Some other example of servers are file servers, media streaming servers and web servers.

The protocols we will discuss in this part are:

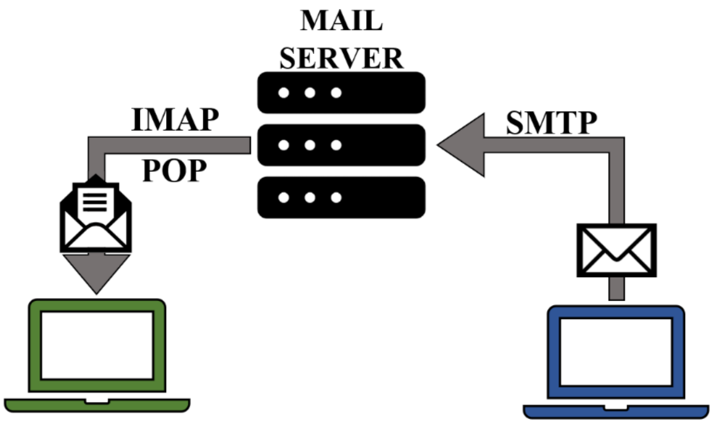
> Email Protocols – SMTP and POP/IMAP
> File Transfer Protocol (FTP)
> Real Time Streaming Protocol (RTSP)
> Network Time Protocol (NTP)

Email protocols – SMTP and POP/IMAP

Simple Mail Transfer Protocol (SMTP)

SMTP is a protocol used for **sending emails**. When you send an email to your friend, the email travels from your computer to your mail server. The email is then transferred over to your friend's mail server. This is all done through computers and mail servers using SMTP in order to communicate with each other.

You can think of **SMTP** as a **mailman**, who will find the correct route for your mail and deliver it to your mailbox.



Post Office Protocol (POP) and Internet Message Access Protocol (IMAP)

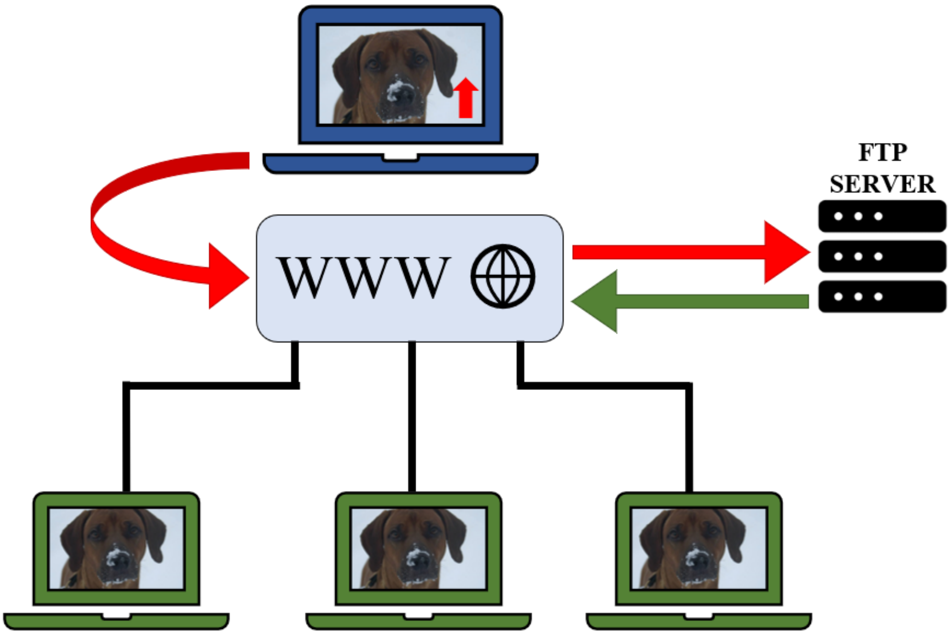
We now know that SMTP is a protocol used for *sending* email. However, in order for your friend to see that what you have sent, the email must be retrieved from your friend's mail server. This is where protocols such as **POP** and **IMAP** are used to **retrieve email** from the mail server. An email client – such as MS Outlook – is connected to the mail server and delivers the email to the computer. POP stands for *Post Office Protocol* (POP) and IMAP stands for *Internet Message Access Protocol*. Both protocols retrieve email from a mail server, but they do it in slightly different ways.

If you think of SMTP as your mailman – who delivers mail to your mailbox – then think of **POP or IMAP** as your **butler**, who retrieves mail from your mailbox and hands it over to you.

With the FLIR AX8, you can send an email notification when an alarm is triggered. These features use the email protocols SMTP and POP/IMAP. At the end of this chapter there will be an exercise showing you how to set up your local mail server and receive email when an alarm is triggered.

File Transfer Protocol (FTP)

FTP is a protocol used to **transfer files** between devices. Say that you wish to share some files to other people over the internet. One way for you to do this is to upload your files to an FTP server. There, people may log on to your FTP server and access the files. It is also possible for your own computer to act as an FTP server, for example by using Windows Internet Information service (IIS). The owner of the FTP server sets up authentication and chooses if access to the FTP server requires a login and password.



When an alarm is triggered by the FLIR AX8, you may choose to store a captured image or video on an FTP server. At the end of this chapter I will go through how to set up an FTP server and receive images or videos from your IR camera when an alarm is triggered.

Real Time Streaming Protocol (RTSP)

RTSP stands for *Real Time Streaming Protocol*. It is a protocol used to **control media streaming**, highly used in entertainment and communications systems. Example of applications using RTSP are YouTube, VLC and Spotify.

Box tip – view your FLIR AX8 stream in VLC

You can access the camera stream with a media player – such as VLC.

Open VLC and in the top left corner, **click** on *Media*.

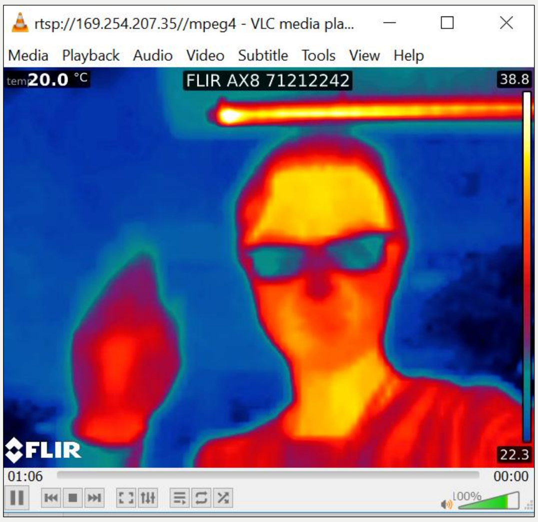
Select *Open network Stream (Ctrl + N)*.

In the field *Pleases enter a network URL:*

type *rtsp://<your-FLIR-AX8-ip-address>/mpeg4*.

In my case I type in rtsp://169.254.207.35/mpeg4.

Now you should see a live stream of your FLIR AX8.



Network Time Protocol (NTP)

NTP is a protocol for **clock synchronization**. In many applications it is important that the devices in a network have synchronized clocks. This is not completely trivial due to *latency*, which means that it takes different amounts of time for different devices to communicate in a network. NTP take these latencies into consideration and sync all participating devices within a few milliseconds of Coordinated Universal Time (UTC).

Hands-on exercises

Set up a local FTP server and send an image or video to your FTP site

As I mentioned earlier, FTP is a protocol used to transfer files over a network. You might also remember that your own computer may serve as an FTP server. In this exercise, I will go through one way to set up an FTP server on your computer and access it. I will also show how you can set up the FLIR AX8 so that when an alarm is triggered, a captured image will be sent to your FTP server.

Note

- The FLIR AX8 camera and the computer running the FTP server must be on the same local network, otherwise the camera will not be able to connect to the server.

Requirements

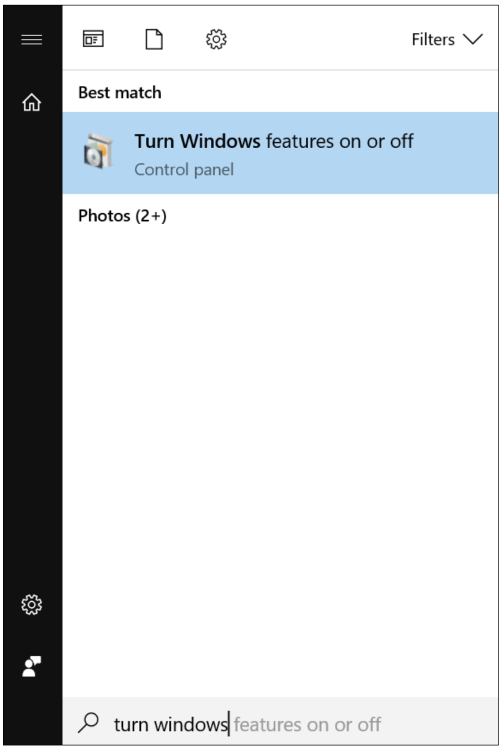
- PC running Microsoft Windows 10 (user with admin rights).
- FLIR AX8, with power and Ethernet cables.

Set up Windows Internet Information Services (IIS)

First, you need to enable Internet Information Services (IIS) on Windows. This is what we're going to use to set up the FTP server.

Click the Windows Start button and then **type** `turn windows` in the search field.

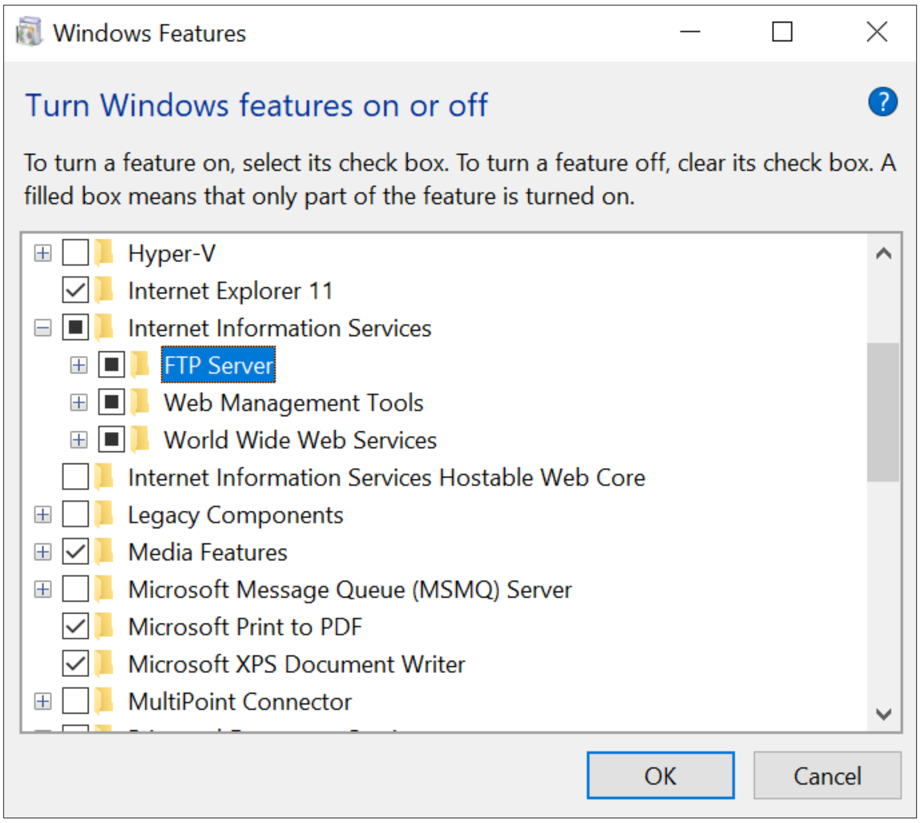
Click `Turn Windows features on or off.`



Expand `Internet Information Services`, and **check** the following options:

`FTP Server`, `Web Management Tools` and `World Wide Web Services`.

Then **click** `OK`.

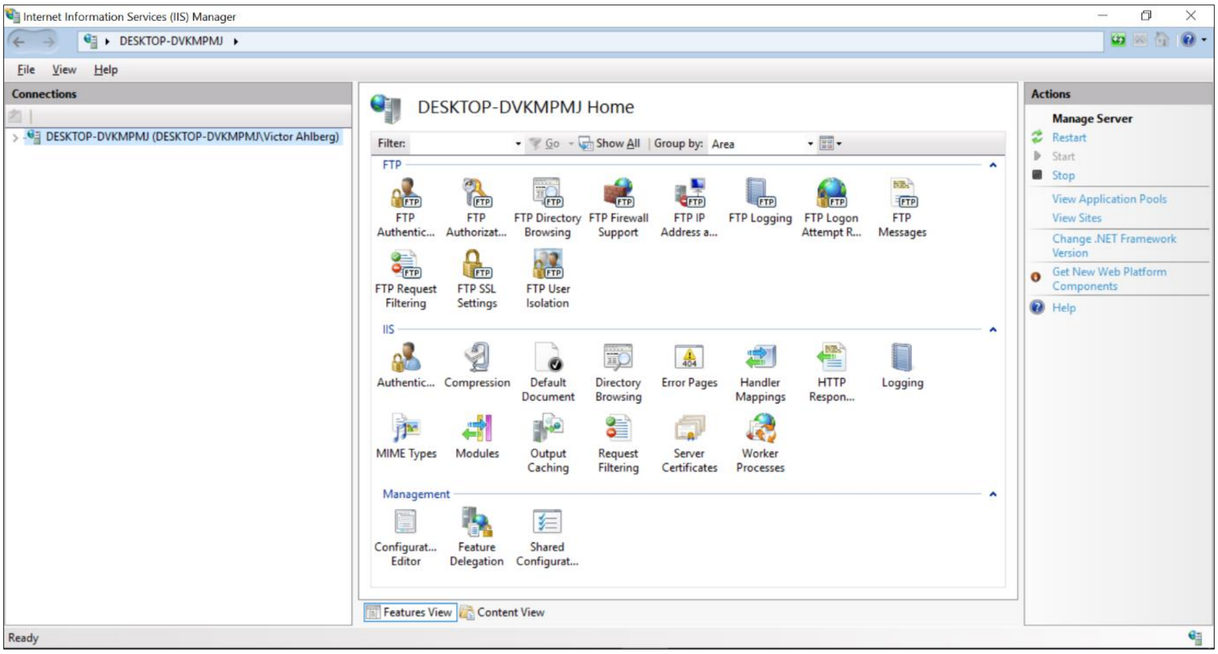


Configure your FTP Server

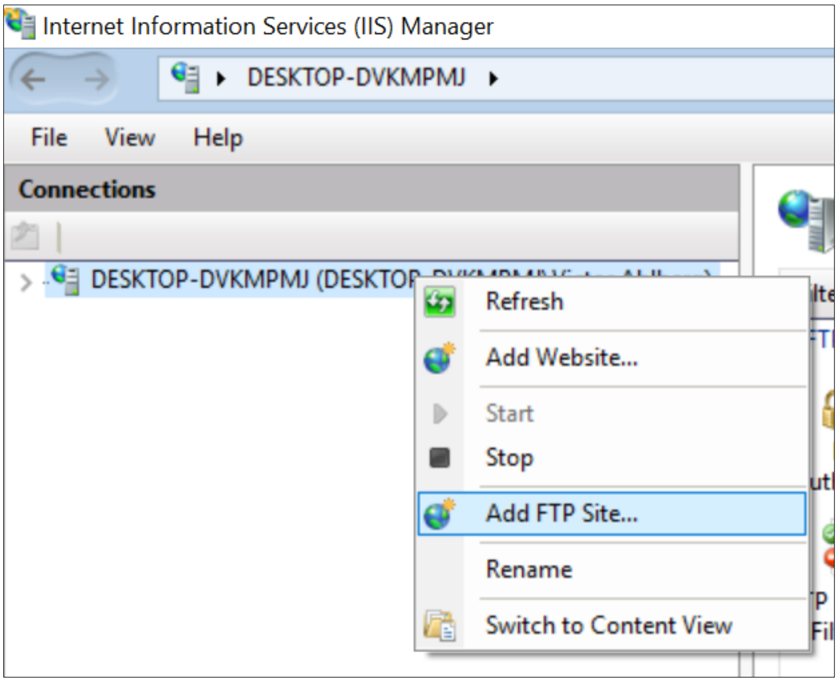
First, make sure your FLIR AX8 is **turned on and connected** to your computer, this is important when we set up the FTP server.

Click the Windows Start button and **search** for `IIS`. **Click** `Internet Information Services (IIS) Manager.`

You should see a window pop up like this one:

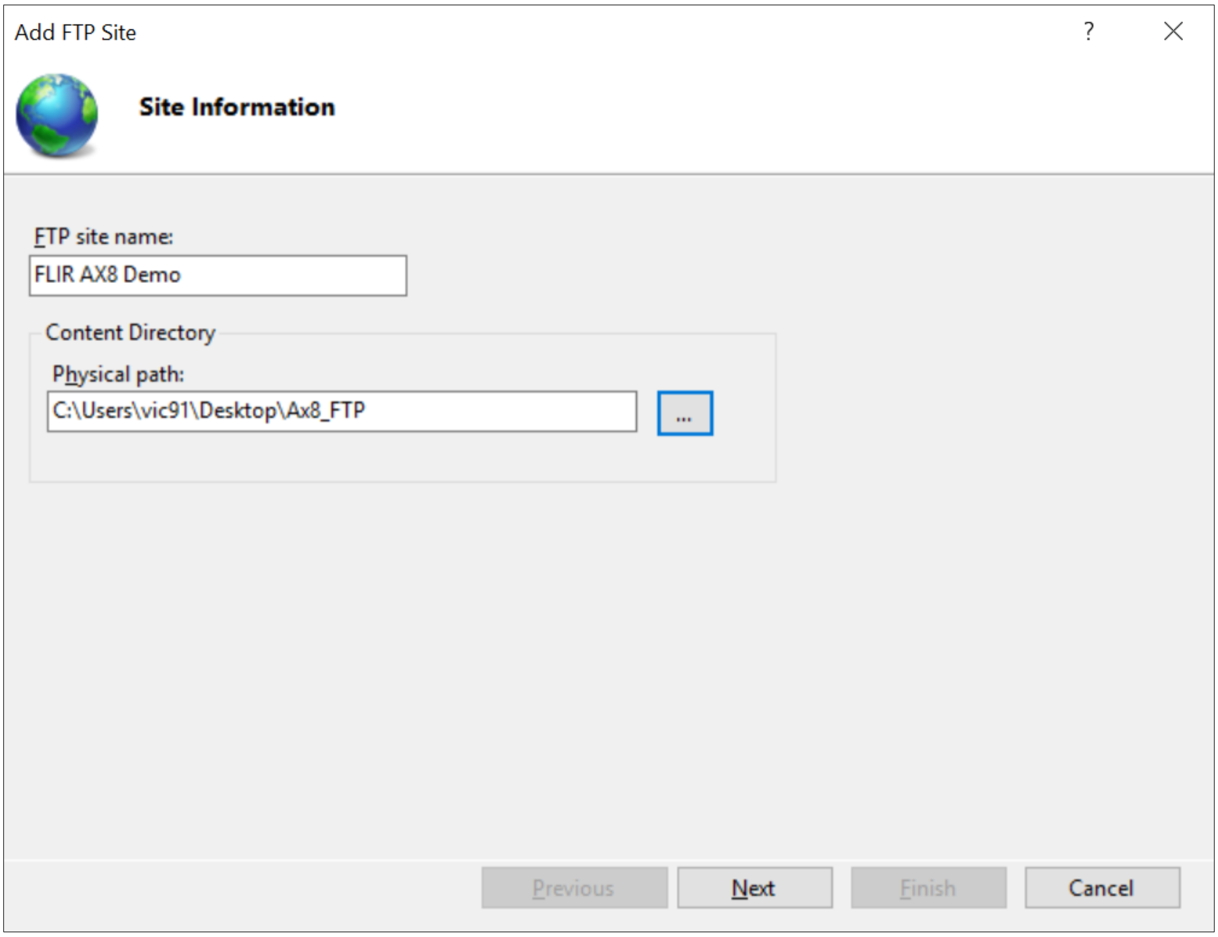


Right-click the left pane and **click** `Add FTP Site`.



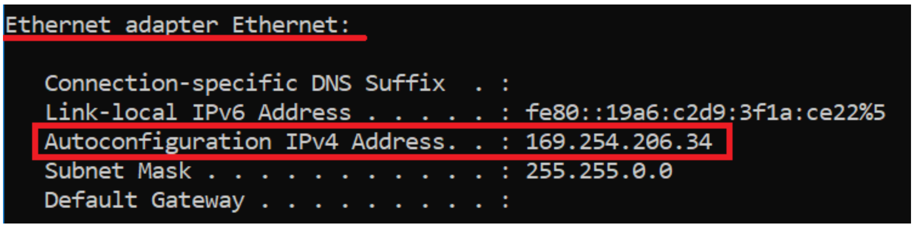
Give your FTP site a name. **Create** and **select** the folder where you want the images and videos to be stored. Then **click** Next.

In my case, I have named my FTP site *FLIR AX8 Demo*. I have created a folder on my desktop called *Ax8_FTP*. This is the folder where the images and videos will be stored.



In the IP Address drop-down menu, **Select** your computer's *IP address*. However, there may be *several* IP addresses to choose from. Your computer is assigned an IP address for *each* network it is connected to. This means that if your computer is also connected to a wireless network, you should find at least *two* IP addresses in the drop-down menu. So, which one IP address is the correct one?

Since our computer and IR camera are communicating via an *Ethernet* cable, it is this corresponding IP address we want. To find this IP address, **type** `cmd` in your computer search menu and **press** `Enter`. In the Command Prompt, type `ipconfig` and **press** `Enter`. **Find** `Ethernet adapter Ethernet` and the `Autoconfiguration IPv4 Address` below. In my case, the correct IP address is 169.254.206.34.

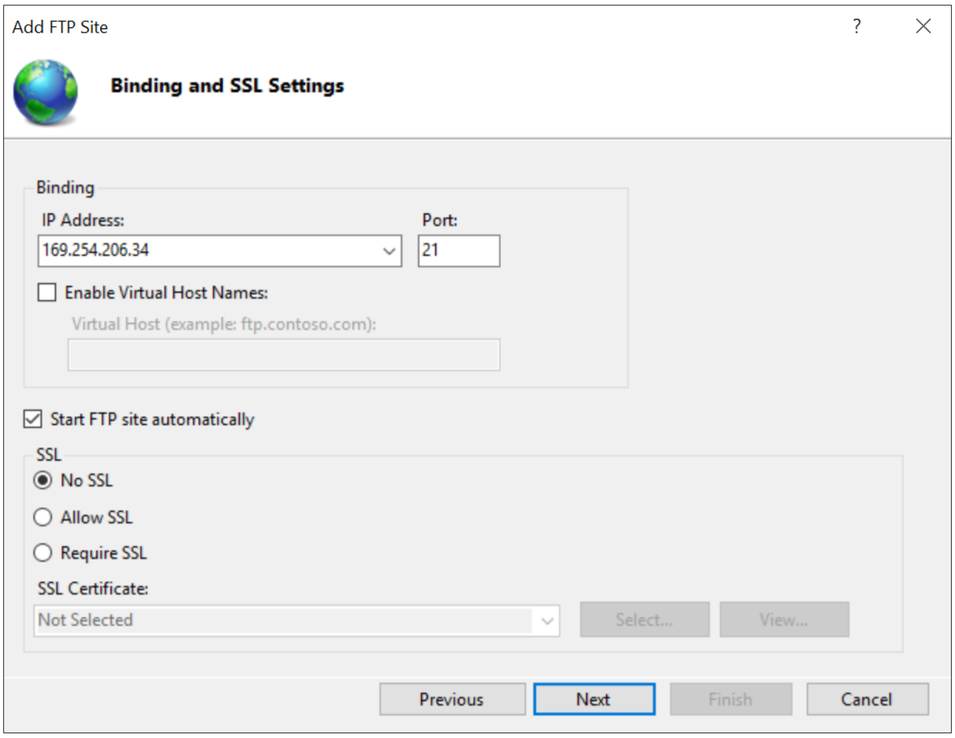


The *port* should be 21, this is the normal port that FTP uses for communication (see page 157 for more on ports).

Note

- Make sure your FLIR AX8 is **turned on** and **connected** to your computer. Otherwise, you may not find your computer's IP address in the IP Address drop-down menu. If you do not find the correct IP address, you may type it in manually.

Check the box `Start FTP site automatically`, **select** `No SSL`, then **click** `Next`.

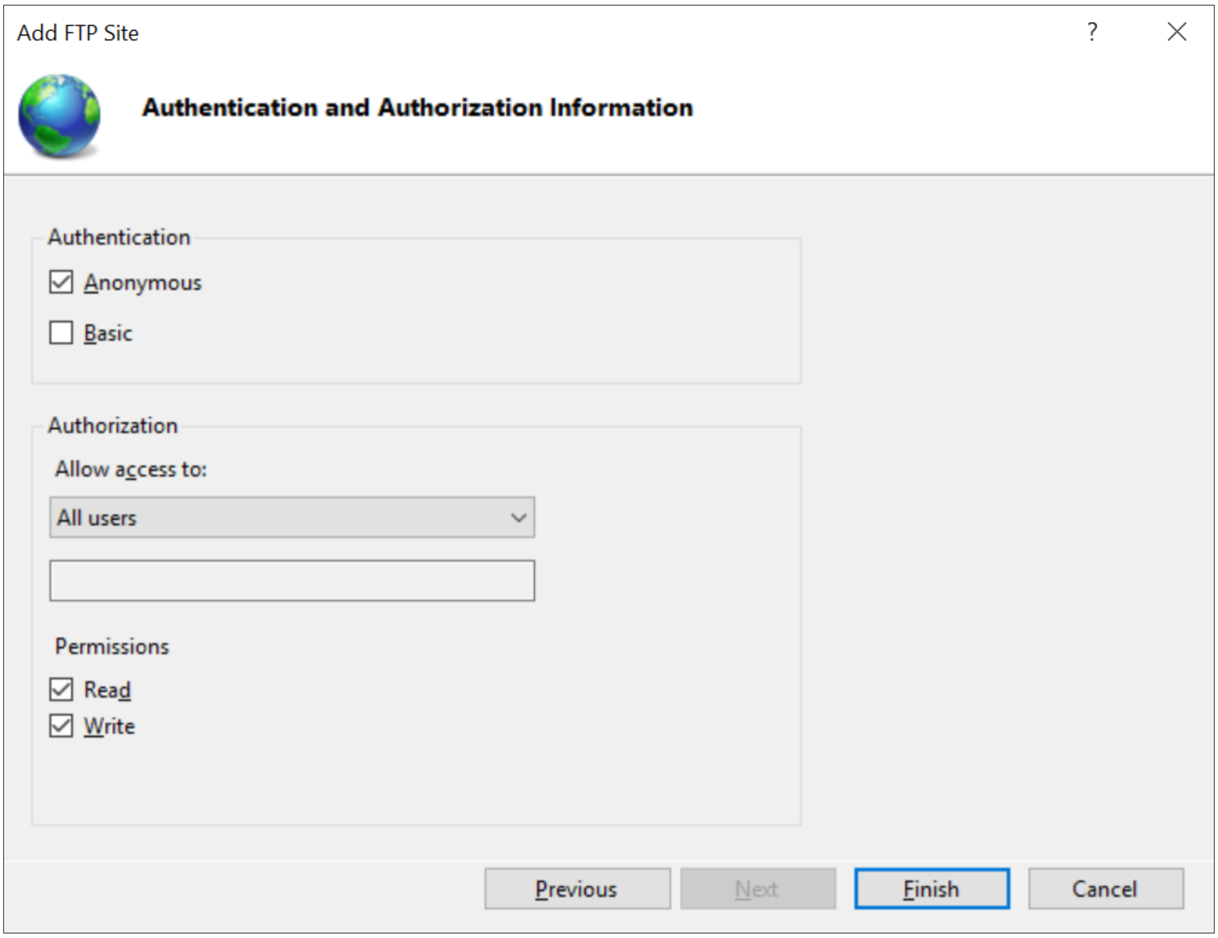


Check `Anonymous` under Authentication.

Select `All users` under Authorization.

Also, **check** `Read and Write`, then **click** `Finish`.

Here we're setting up our FTP with minimal authentication requirements.

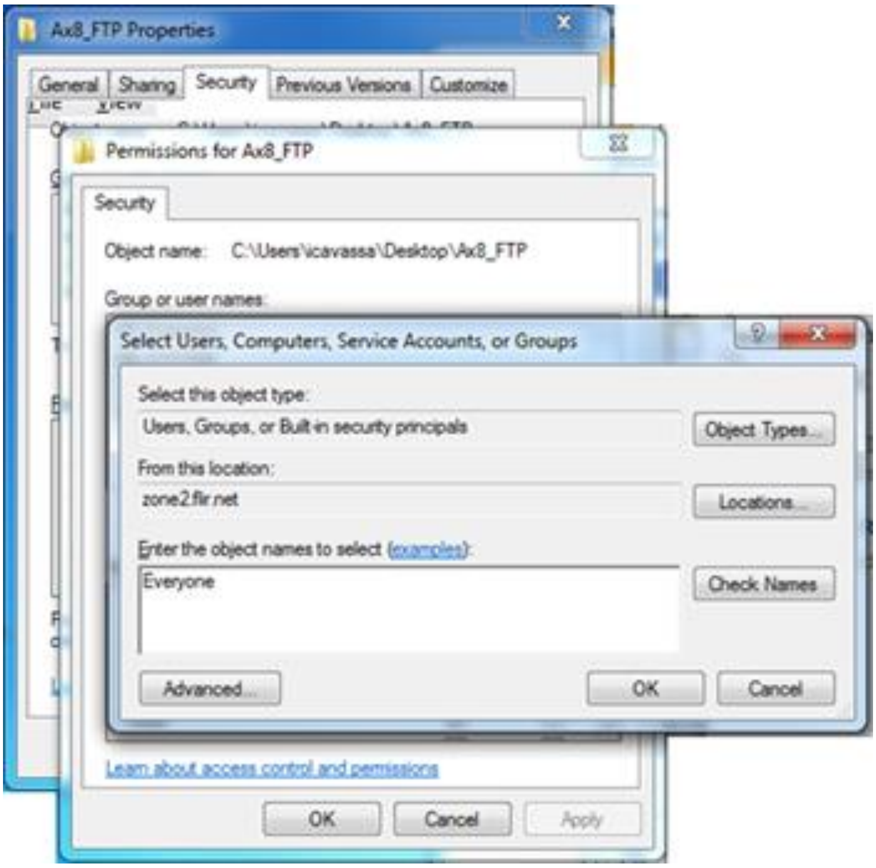


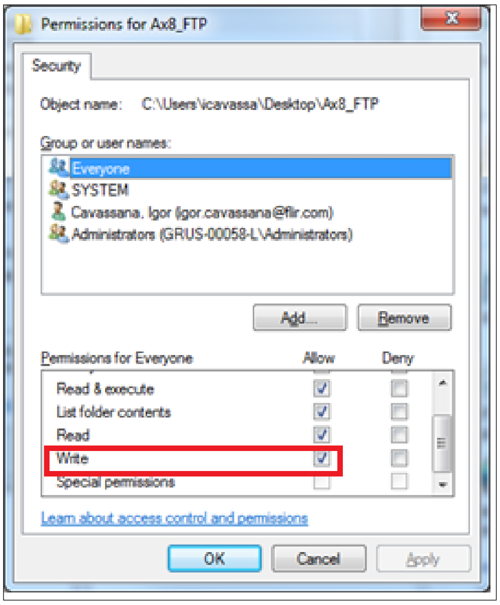
Enable your FLIR AX8 to save images/videos on your FTP server

In order for your FLIR AX8 to save files in the folder we selected, we must ensure that a regular Windows user is allowed to *read* and *write* in that folder*.* In my case, the name of that folder was *Ax8_FTP.*

To do this, **right-click** your folder that should receive the files and **click** `Properties`. **Select** `Security tab`, **click** `Edit…`, **click** `Add…` and **type** in `Everyone`.

Click `OK`.





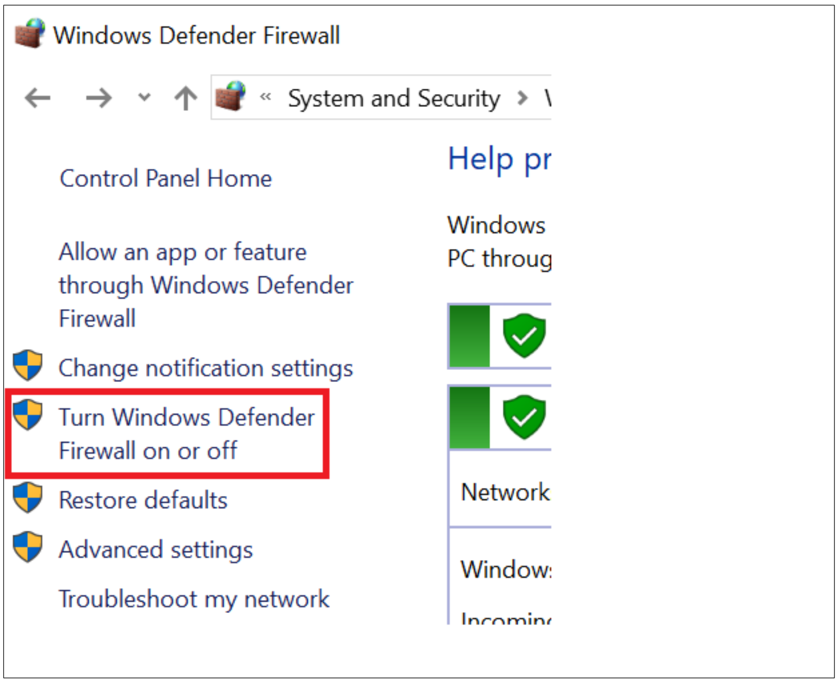
Select the user `Everyone`, **check** the `Write` option, and **click** `OK`. This is important because otherwise the FLIR AX8 will not be able to add any pictures or videos to the folder.

When you've come this far, well done! Now all that is left is to allow FTP through the Windows Firewall and to tell your FLIR AX8 the name of your FTP server.

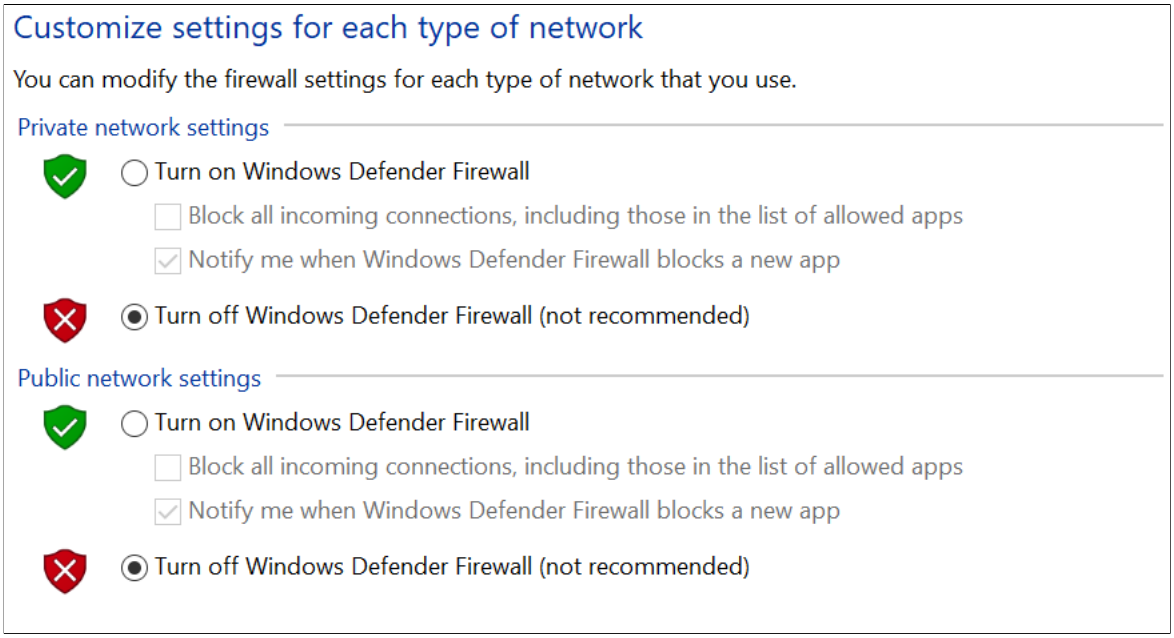
Allow FTP through the Windows Firewall

The easiest way to allow FTP through the firewall is to simply turn it off. Since this exercise is for practice purposes only, we will turn it off. However, in real applications, it is important to ensure your network is secure.

Click the `Windows Start Button`, **search** for `firewall` and **click** `Windows Defender Firewall`.



Select `Turn off Windows defender Firewall (not recommended)` for both `Private` and `Public` network settings.



Set up the FTP server on the FLIR AX8

Log in to your FLIR AX8 web interface and **go to** the `Settings` tab.

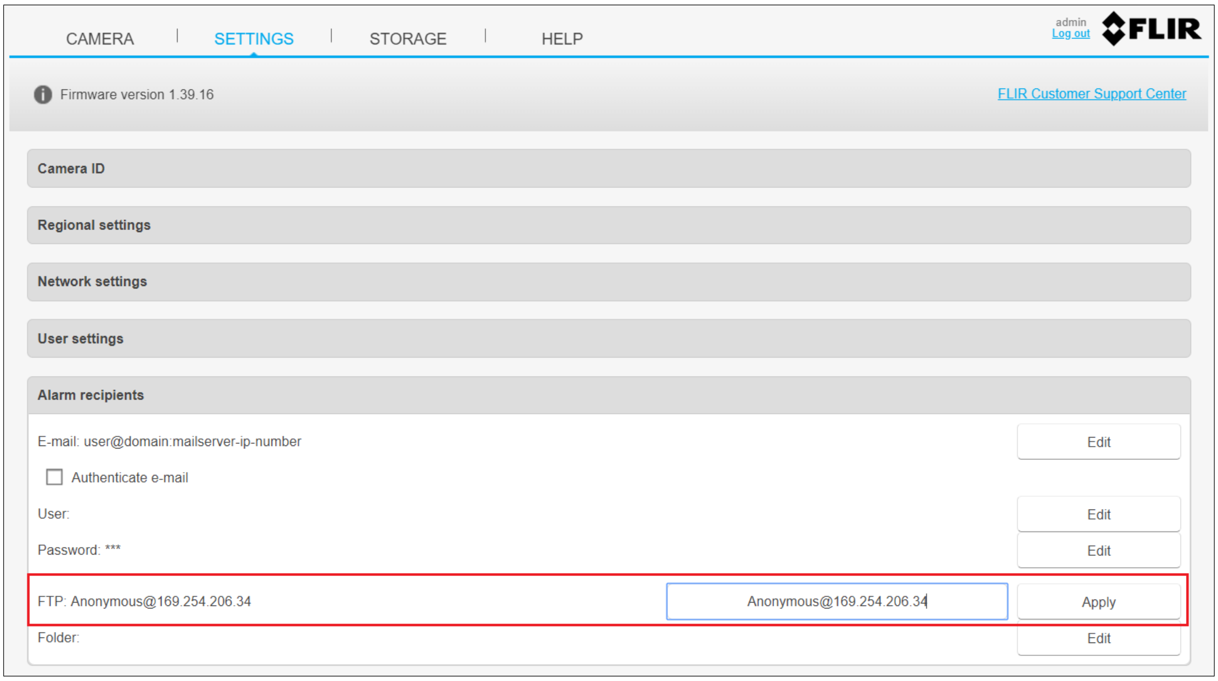
Under the `Alarm Recipients` menu, **edit** the `FTP server` option to

`Anonymous:@<your-computer-IP-address>`

In my case, I put in Anonymous:@169.254.206.34.

Click `Apply` to save changes.

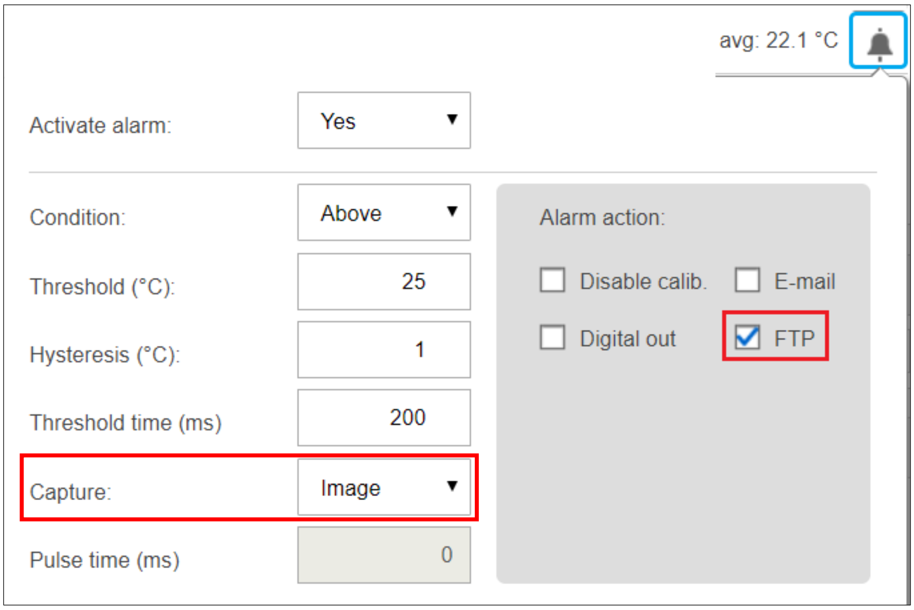
The format is actually *user:password@<your-computer-IP-address>* – but since we set up our FTP server *not* to require a password – we tell the camera to access the FTP Server as an anonymous user with no password.



You can also set the camera to save images or videos to a sub-folder by **clicking** `Edit`. The camera will not, however, create the folder itself. If you do not create the sub-folder, the camera will save the files to the FTP root.

This whole exercise was meant to show you how to send an image or video to your FTP server when an alarm is triggered. We must not forget the final and perhaps most important part: to set up the alarm!

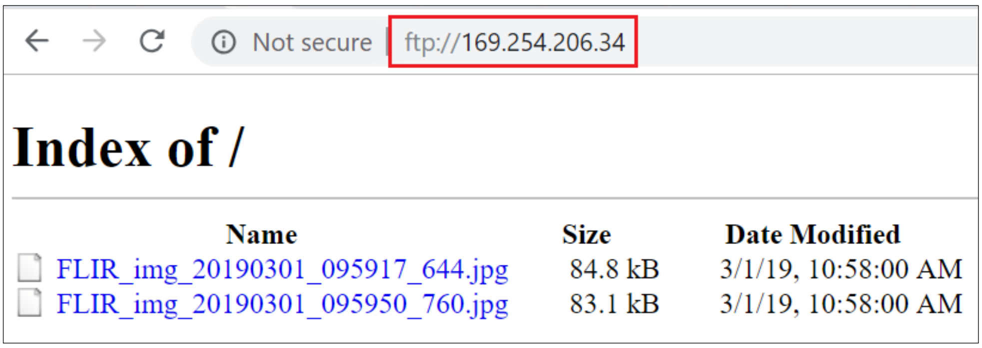
When you set up your alarm, remember to **check the box** `FTP` under `Alarm action` and **select** either `Image` or `Video` to the right of `Capture`.



Everything is now set up**, well done**!

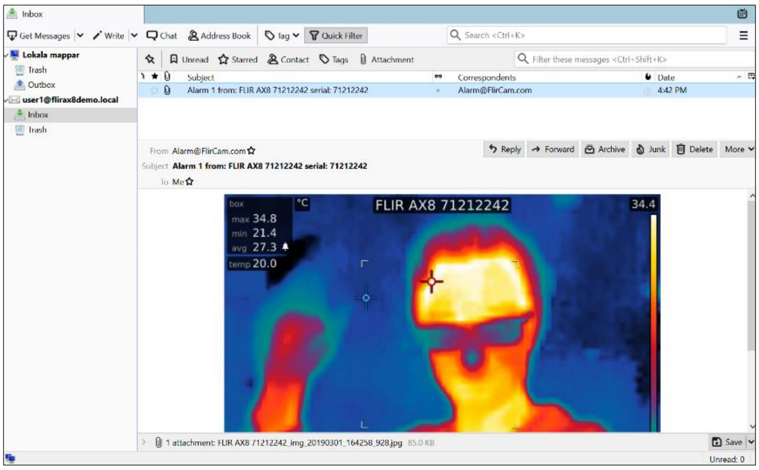
When an alarm is triggered you should now get an image or video capture in your FTP folder.

You can also access your FTP via a web browser, such as Google Chrome. **Type** `ftp://<your-computer-IP-address>` in the search bar. Notice that the address begins with `ftp` and not `http` or `https`, which are the usual protocols used to access web pages. This is precisely because we wish to access an FTP server, and thus we specify that the browser should use the FTP protocol.



Set up a local mail server and receive email when an alarm is triggered

In this exercise will show you how to set up a local mail server and email client in order to receive an email notification when an alarm is triggered.



Note

- The FLIR AX8 camera and the computer running the mail server must be on the same local network, otherwise the email will not reach the mail server.

Requirements

- PC running Microsoft Windows 10 (user with admin rights).
- FLIR AX8, with power and Ethernet cables.

Set up your local mail server

For this demonstration I will use the free software *hMailServer*. The hMailServer program allows your computer to act as a mail server.

To **download** the software, visit https://www.hmailserver.com/download or search for `hMailServer download` in your web browser's search field.

When installing, I have three important advices for you:

1. **Select** `full installation`, with both server and administrative tools.

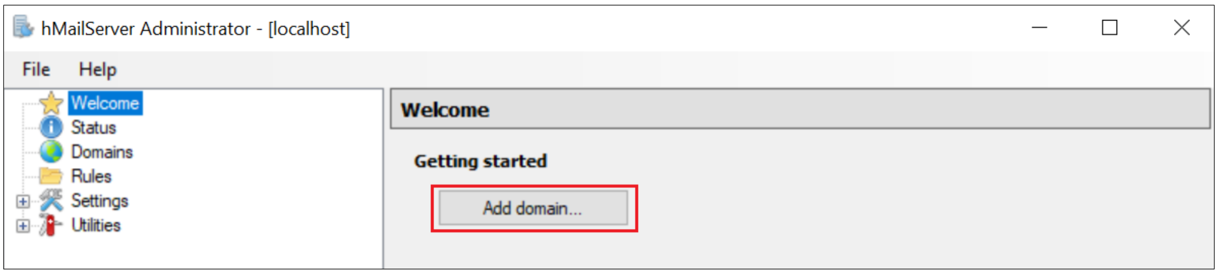
2. **Select** `Use built-in database engine` (Microsoft SQL Compact)

3. You will be asked to choose a password, **remember it**! We will need it when we set up the local mail server.

Once installed, **open** `hMailServer Administrator`.

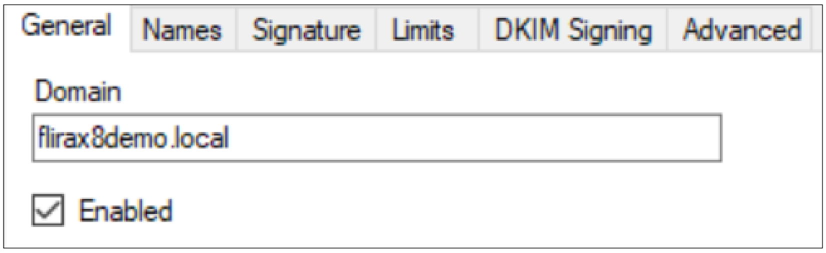
Click `connect` and **enter** the password you chose during the installation.

You should arrive at a window like this:



Click `Add domain`. Now **choose** a domain name. This is what will follow the @-sign. I picked flirax8demo.local

Make sure to **check** the box `Enabled`. **Click** `Save`.

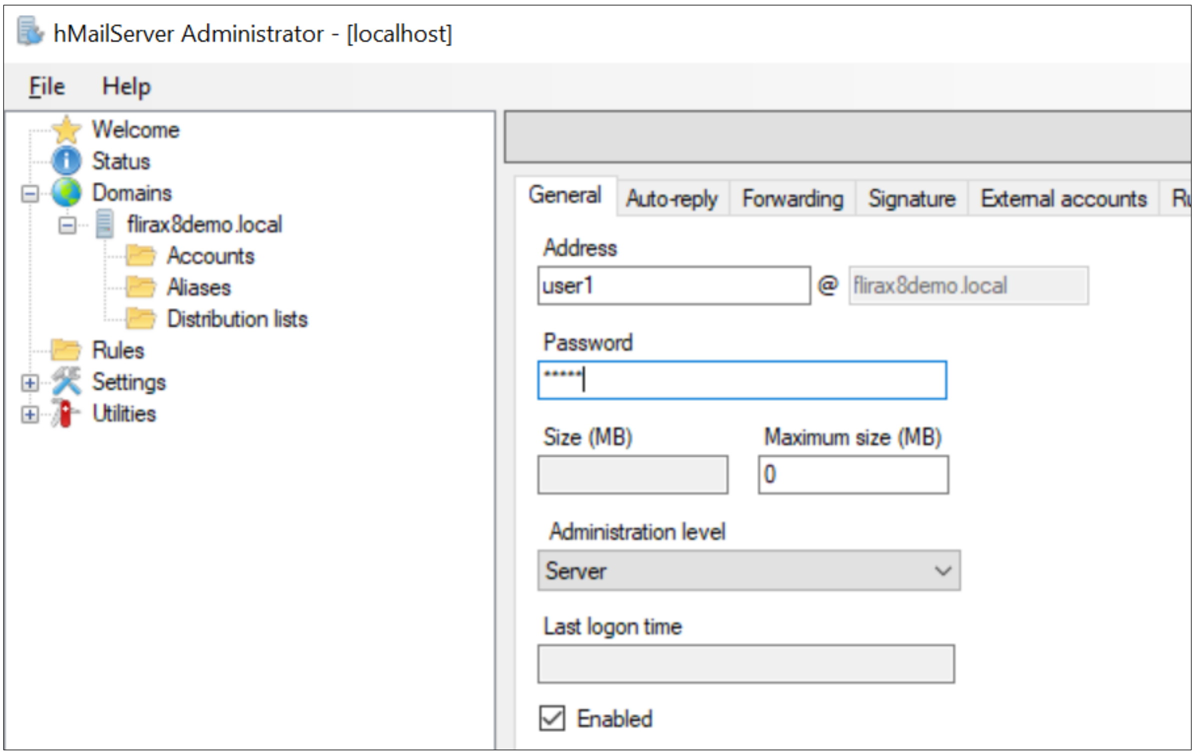


We will now add a user so that the camera has someone to send the capture to via email.

Select the folder `Accounts` and **click** `Add`.

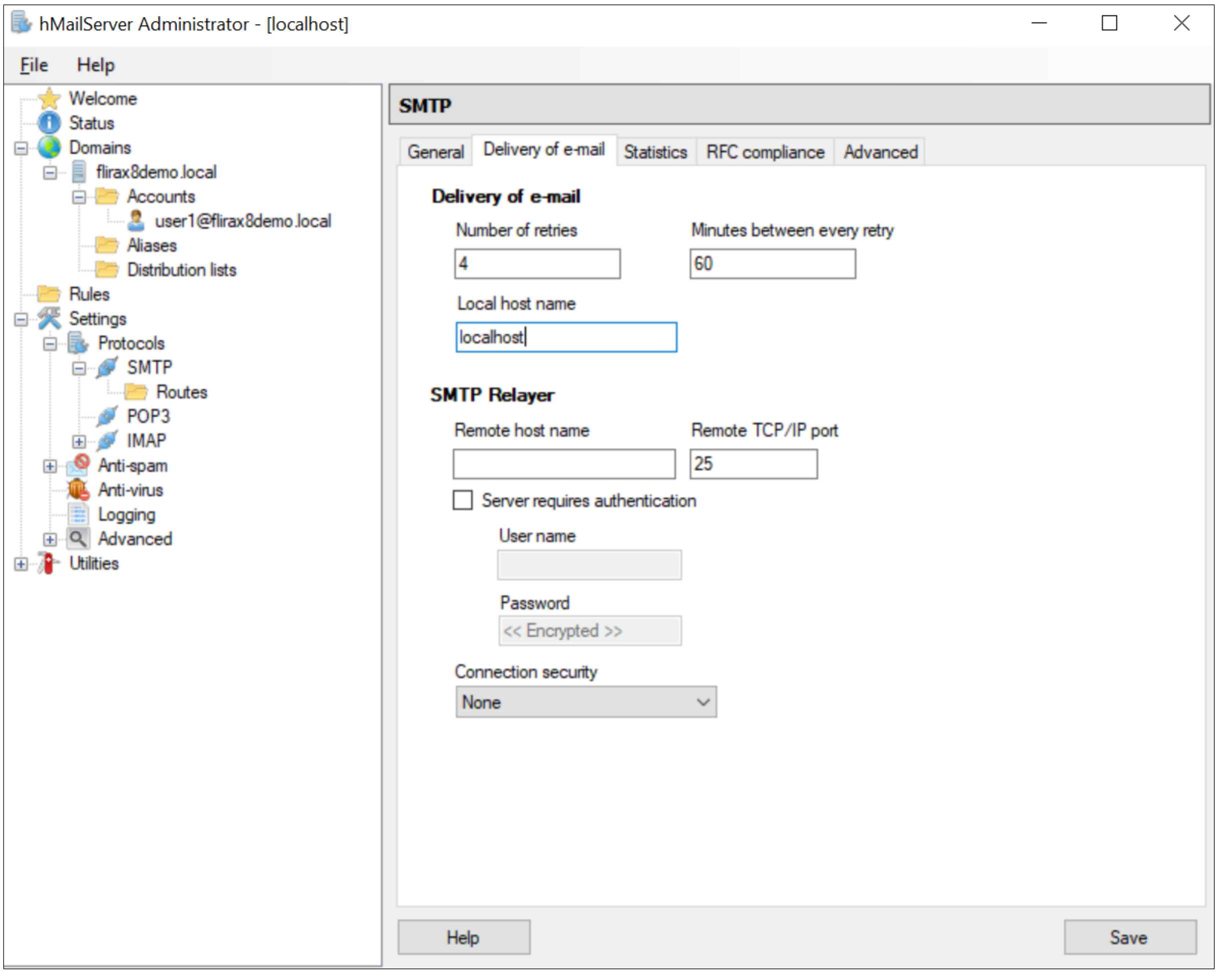
In the `Address` field, **choose** a username and a password. I picked user1.

Under `Administration level`, **select** `Server`. **Make sure** to tick the `Enabled` box. Then **click** `Save`.


hMailServer Administrator - [localhost]
File
Help
Welcome
Status
Domains
flirax8demo.local
Accounts
Aliases
Distribution lists
Rules
Settings
Utilities
General
Auto-reply
Forwarding
Signature
External accounts
Address
user1
@
flirax8demo.local
Password
*****
Size (MB)
Maximum size (MB)
0
Administration level
Server
Last logon time
Enabled

Now **expand** `Settings >> Protocols` and **click** on `SMTP`.

In the `Delivery of e-mail` tab, put `localhost` in the Local host name field. Then **click** `save`.



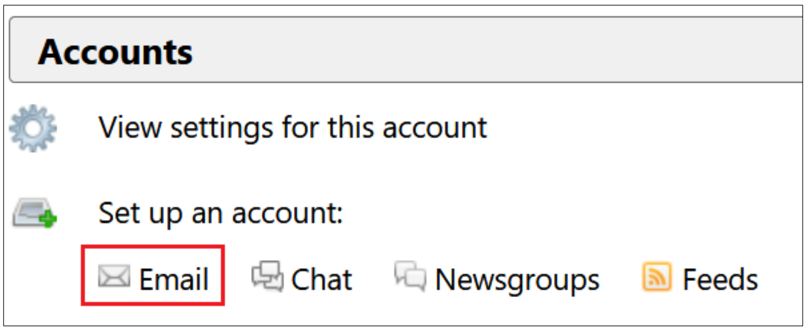
Now, we have set up our mail server. All email sent from the FLIR AX8 will be stored on this mail server. We now need a way to *retrieve* the email from the mail server to our computer. Remember that this is exactly what we use an **email client** for. Email client will access the mail server and retrieve the email to the computer using the protocols POP or IMAP.

Install and set up an Email client

There are many email clients out there and you are free to use whichever you prefer. For this demonstration I will use **Mozilla Thunderbird**, which you can **download** at `https://www.thunderbird.net` or by **searching** for `Mozilla thunderbird download` in your web browser.

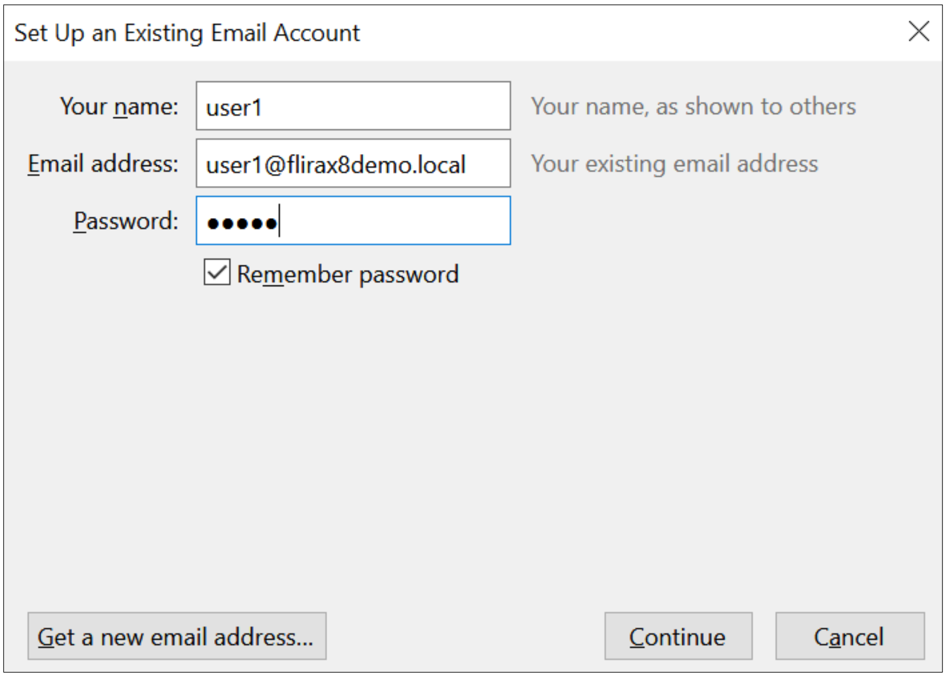
Download and **install** `Mozilla Thunderbird`, then **open** the program.

Under `Accounts` >> `Set up an account`, **click on** `Email`.



Choose a name for the account. The Email address and Password *must be the same* as the account you added in hMailServer.

In my case, the email address was *user1@flirax8demo.local*.



Then **click** `Continue`.

Thunderbird will say it failed to find the settings for your email account. This is okay!

For *Incoming:*

Select `POP3` *Server hostname:* `localhost`

Port: `110` *SSL:* `None`

Authentication: `Autodetect`

What we're setting up here in *Incoming* is how our email client should *retrieve* email from the mail server. Remember that there are two mainly used protocols used for retrieving email. These are *POP3* and *IMAP*. When the FLIR AX8 sends an email notification to the mail server, our email client needs to know how to get it to the computer.

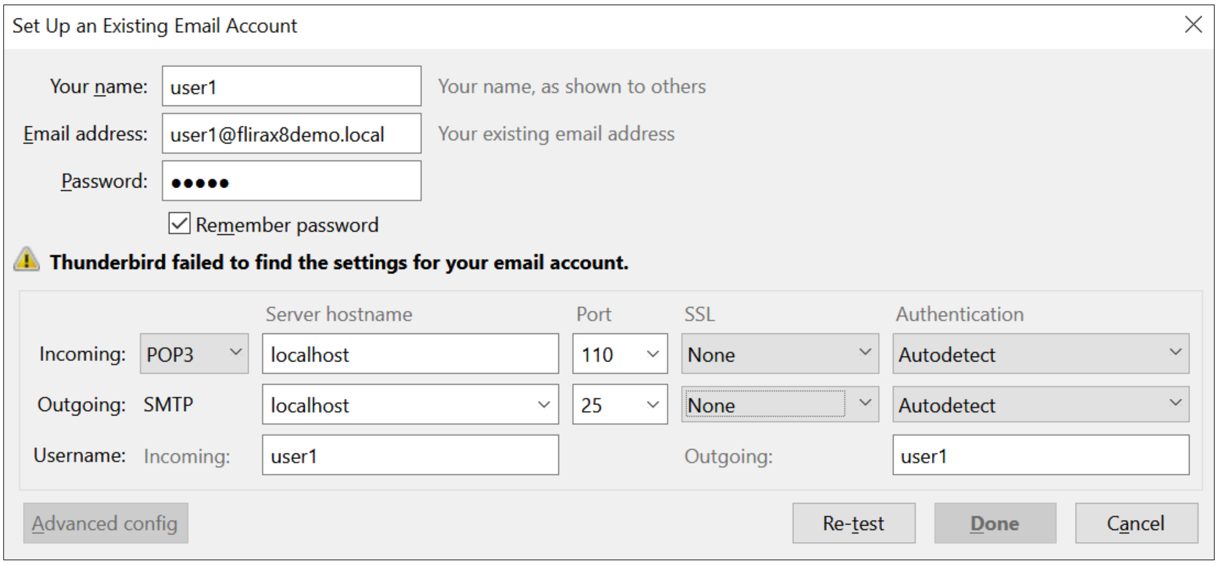
For *outgoing:*

`SMTP` *Server hostname:* `localhost`

Port: `25` *SSL:* `None`

Authentication: `Autodetect`

In *Outgoing,* we're setting up how our email client should *send* email. Remember that the protocol used for this is SMTP, which is why we see it in the Outgoing field.



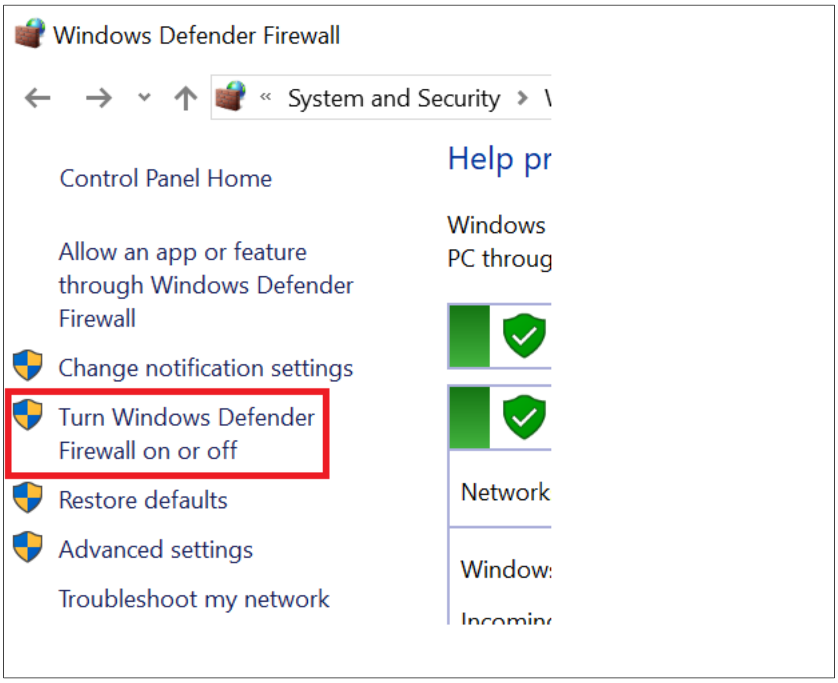
Click `re-test` and then `Done`. **Check** the box that you understand the risks of not using encryption and **click** `Done`.

With the last click, you have successfully set up your email client to access your mail server. Now, we have to make sure that the firewall won't stop our email activity. After that, all that is left is to make the FLIR AX8 find you via email.

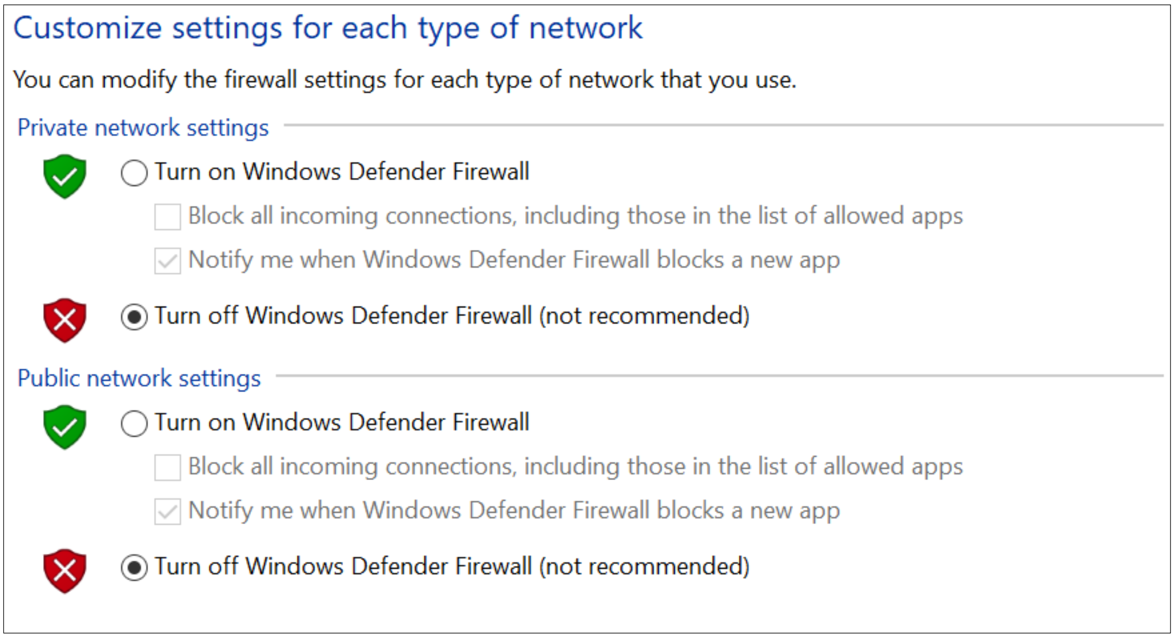
Turn off the Firewall

The easiest way to allow emails through the firewall is to simply turn it off. Since this exercise is for practice purposes only, we will turn it off. However, in real applications, it is important to ensure your network is secure.

Click the `Windows Start Button`, **search** for `firewall` and **click** `Windows Defender Firewall`.



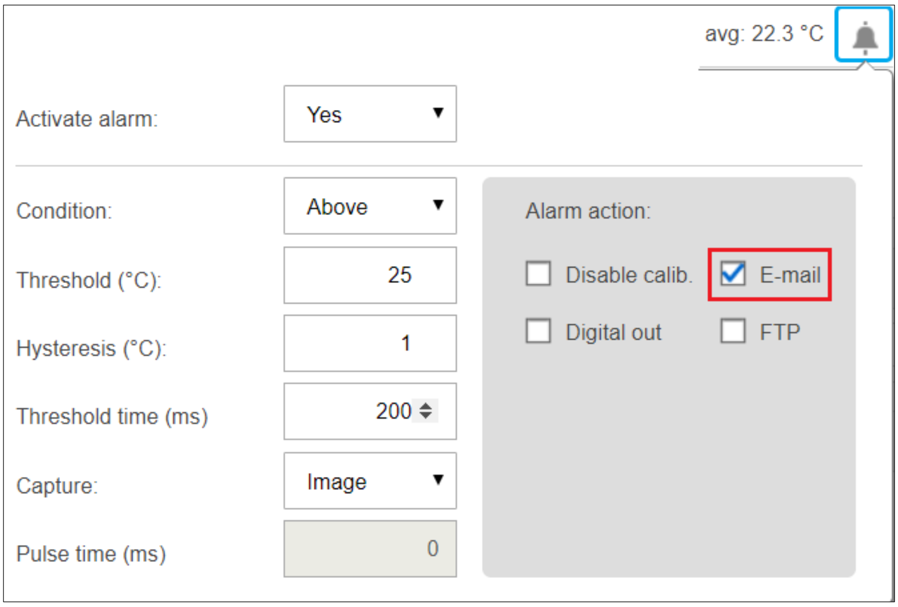
Select `Turn off Windows defender Firewall (not recommended)` for both `Private` and `Public` network settings.



Almost there! All that's left is to activate an alarm with the FLIR AX8 and specify the email recipient.

Set up an alarm in FLIR AX8

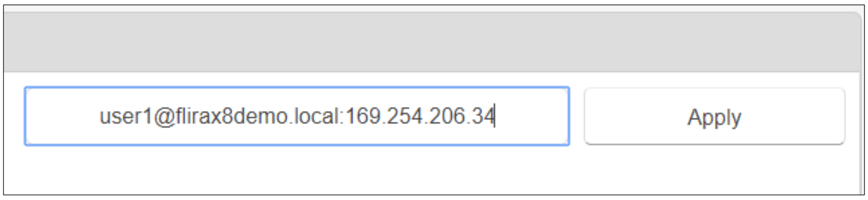
Log on to your FLIR AX8 web interface and **set up an alarm**. When you set up your alarm, remember to **tick the box** `E-mail` under `Alarm action`.



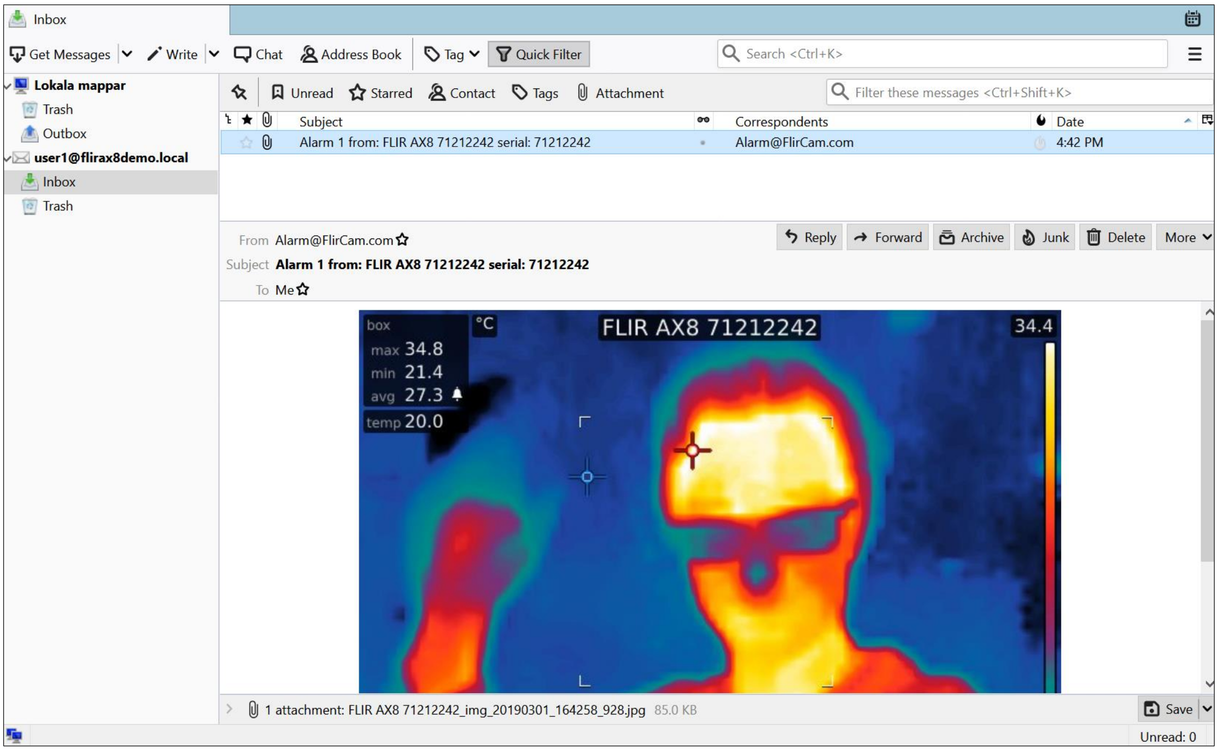
Under the **Settings tab** go to `Alarm recipients`. **Enter** your email address and IP address of the computer hosting the mail server, separated by a colon.

In my case, I entered *user1@flirax8demo.local:169.254.206.34*.

Click `apply`.



Well done! Now you should receive an email when the alarm is triggered, and it should look something like this:



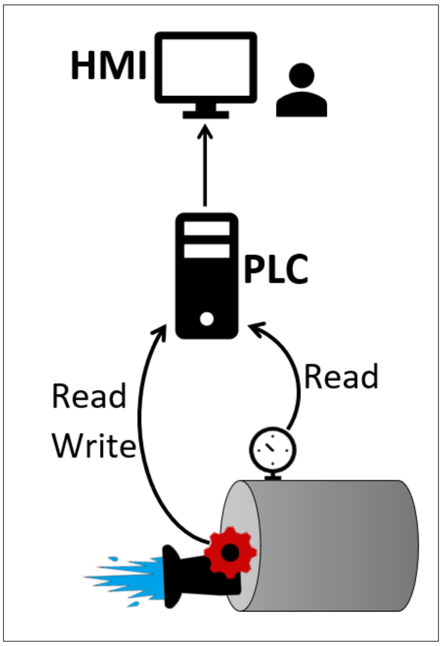
Fieldbus

In this section, we will look at protocols and systems used in the industrial arena. **Fieldbus** is the name of a family of industrial computer network protocols used for real-time distributed control. An automated industrial system is often complex, think for example of a manufacturing assembly line. For the whole system to function we need an organized hierarchy of control systems with computers and instruments connected to a single network that allows for real-time control and monitoring. When talking about these types of systems we often use the term **SCADA**, which stands for *Supervisory Control and Data Acquisition*.

SCADA simplified – three levels

Let's look at a very simplified model of a SCADA system, consisting of three levels.

Say that we wish to regulate the flow of water in a pipe. To do this, we have two devices directly connected to the pipe: a valve that regulates the flow and a flow meter that measures the water flow. These devices belong to the bottom level in the SCADA system.



However, the valve does not know how much water is flowing and the measuring device does not know how much water that *should* flow through the pipe. That's why at the middle level, we have a computer connected to the bottom level devices that controls the system. The computer receives information from the measuring device about how much water is flowing. Based on that information, it tells the valve how much it should restrict the water flow. This computer is called a *Programmable Logic Control* – or **PLC**.

Even though the PLC knows what's going on in the system, we – the operators – also want to monitor the system. At the top level, the information sent from the PLC is translated and presented in a format that is readable to humans. This is called a *Human Machine Interface* – or **HMI** – and it displays information about the ongoing process to the operator.

In reality – these systems are of course much more complex, with more devices and levels – but I hope you see the main idea.

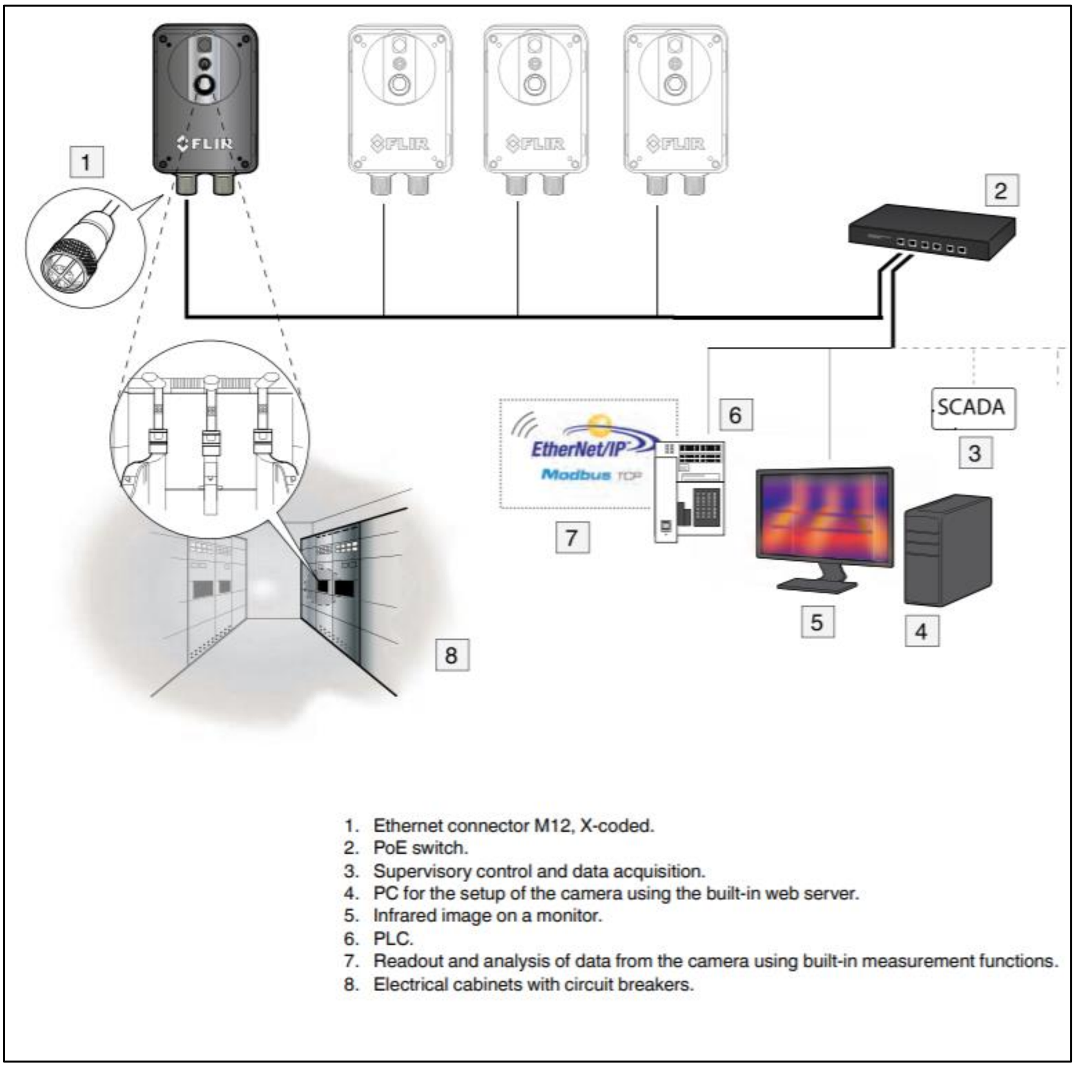
- At the bottom level, we have the devices actually doing the work. In the example used earlier, these would be the valve regulating the water flow and the measuring device.

- At the middle level, we have PLCs that control and manage the system, based on the data received from devices at the bottom level. In the example used earlier, this would be the computer reading the measured water flow and telling the valve how to regulate it.

- At the top level, the HMIs translates the data displays it to us – the humans – so that we can monitor the process.

Typical system overview – FLIR AX8 manual

Now let’s take a closer look at a typical system overview, presented in the FLIR AX8 user manual.



You see that the FLIR AX8 is monitoring an electrical cabinet (1). This is to alert if the circuits are getting too hot *before* a problematic power failure occurs.

The camera is connected to a Power over Ethernet switch (2). Out from the switch, several connections can be made.

At (3) you see a reference to the control system architecture SCADA, which we covered in the section above.

Perhaps what you’re most familiar with is to connect the camera to a computer (4) and set it up using the FLIR AX8 web interface (5). However, the FLIR AX8 can also be connected a

Programmable Logic Controller – PLC (6). This refers to the same sort of PLC that we covered in the introduction to SCADA.

The FLIR AX8 supports two protocols in the Fieldbus family: **Modbus TCP** and **EtherNet/IP** (7). In this chapter, we will dedicate some time and effort to cover these protocols. First, let's make an important distinction between two types of IR cameras: smart sensor cameras and image streaming cameras.

Two different types of IR Cameras

We can distinguish between two types of IR cameras: **smart sensor cameras** and **image streaming cameras**. The FLIR AX8 and FLIR A310 are considered **smart sensor cameras**, because they have built-in measurement functions for analysis. They not only register infrared images, but the cameras themselves also analyze the images (see chapter Smart sensor cameras on page 133). The FLIR AX8 and FLIR A310 support the protocols **Modbus TCP** and **EtherNet/IP**.

All other FLIR automated infrared cameras – such as the FLIR Ax5 and FLIR A615 – are what we call **image streaming cameras**, these cameras only capture the infrared images (see chapter Image stream on page 131). The analysis is then made on an external computer. Some applications require rapid analysis, and the analysis done in the camera may be too slow. Think of a conveyor belt with rapidly moving products. Image streaming cameras don't support Modbus TCP or EtherNet/IP, but instead support another protocol called **GigE Vision**. We will cover this protocol later in this chapter.

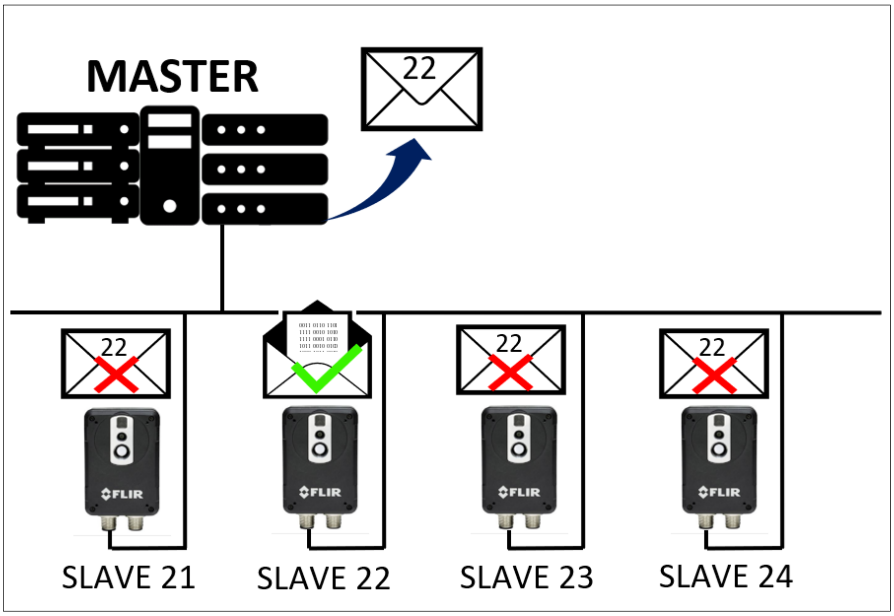
Modbus

Modbus is one of the most common automation communication protocols used in the industrial arena. You can think of Modbus as a protocol that provides a *common language* for devices and equipment to communicate with each other. There are many protocol variants of Modbus and some of these are Modbus RTU, Modbus ASCII and Modbus TCP/IP and Modbus Plus. If you think of Modbus as the *common language* used by devices to talk to each other, you may think of the different protocol variants as *different accents* of the same language.

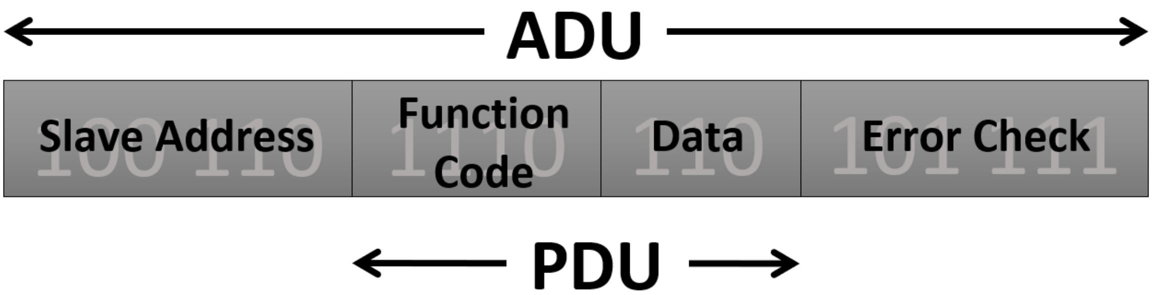
Modbus communication

Modbus is based on a master/slave architecture. Think of it as a model of communication where the master device has control over one or several slave devices. A slave can be any measuring device which processes information. The IR camera is one kind of slave device. For an automated system to work, we need ways to access the data that the measuring device – in this case the IR camera – has registered. This data is stored in the slave's **memory register.**

The information is sent through the system via interactions between masters and slaves These interactions are called **queries.** A master can address individual slaves or initiate a broadcast message to all slaves. A conversation is always started by a master. The slave then interprets the message and responds to it. Slaves cannot initiate conversations; they only respond to messages sent from the master.



All messages in Modbus communication have the same basic structure. This way, we can access the memory register of a slave to retrieve or change information in the same manner regardless of the type of slave. A message consists of four parts: **Slave address**, **Function code**, **Data** and **Error check**. The whole message is called an **Application Data Unit** – or ADU. A function code together with data is called a **Protocol Data Unit** – or PDU. Let's go through the different parts of an ADU and their purposes.



Slave address

Each slave has its own ID in the system. That way, only the slave with the matching **Slave address** specified in the message will respond. All other slaves will ignore the message. You can see in the picture above that all slaves connected to the master receives the message, but all ignore it except for slave 21 which the message is intended for.

PDU = Function code + Data field

The **Function code** field specifies what action the slave should perform. The **Data field** specifies how much information in the memory register the slave is requested to handle. These actions revolve around either **read** or **write**. Perhaps we wish to know about (read) a measurement done by the IR camera or change the value (write) in the camera's configuration settings. Depending on what sort of information we wish to access, the type of data will be different. To deal with different types of data, Modbus has four basic types of registers:

Type of register	Action	Size	Possible values	Information
Discrete inputs	Readable	1-bit	0, 1	On/Off
Coils	Readable/Writable	1-bit	0, 1	On/Off
Input Registers	Readable	16-bit	0 – 65 535	Measurements, Statuses
Holding Registers	Readable/Writable	16-bit	0 – 65 535	Configuration values

The size column gives us the size of the information in the unit *bit*. It is related to binary numbers (see More on binary numbers on page 151). One bit simply means that one piece of information can assume one of two values: 0 or 1. This is not arbitrary – the number that succeeds the unit tells us about the possible values for that piece of information. It's connection to binary numbers is that when talking about bits, we use 2 as a base, and the number succeeding the unit as the exponent. 1-bit will then yield $2^1 = 2$ possible values. For the 16-bit registers, the principle is the same – only the numbers are larger. $2^{16} = 65\,536$ possible values.

A **discrete input** is a single-bit **physical input**, it is also called a *contact*. You can think of it as a fuse, which status you only can see and not change – since it is only *readable*, and not *writable*. A fuse is either on or off, and this goes for the discrete inputs as well.

A **coil** is a single-bit **physical output**. You can think of the coil as a light switch (without dimmer). You can turn the light on or off (writable), and you can also see if the light switch is up or down (readable).

However, not all information we handle can come in the format On or Off. For example, if we wish to know the minimum temperature in a Box measurement, the data size in the register has to be big enough to contain that information. That's why data in input registers and holding registers are bigger than 1 bit. They have been chosen to be 16-bit in size. As a rule of thumb, think of **input registers** as storing *measurements and statuses*. These we want to know about (read), but not change. Think of **holding registers** as storing *configuration values*. These we want both to know about (read) and be able to change (write).

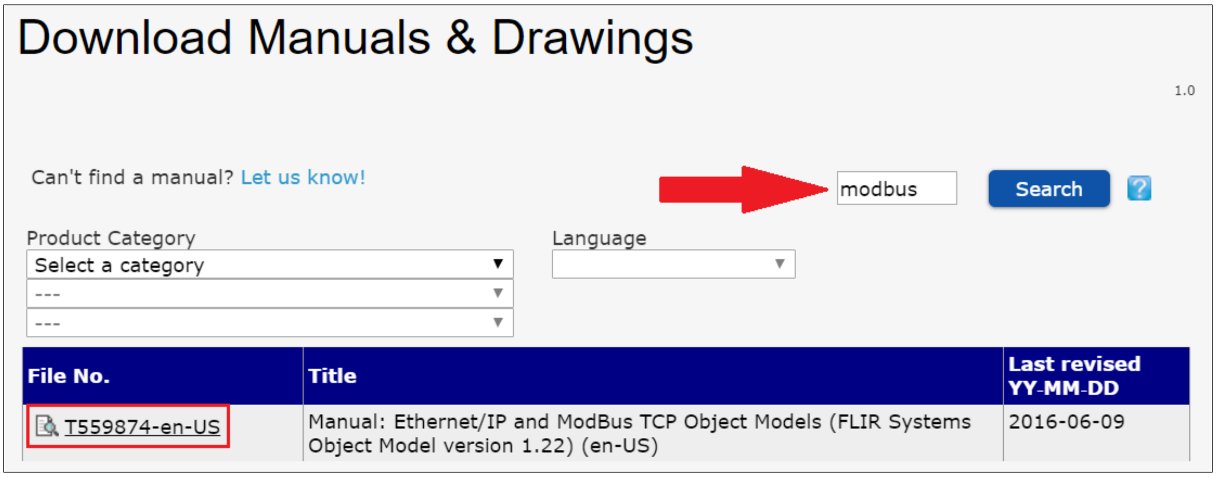
Error check

When information is sent in a network, a lot can go wrong. It then becomes important to manage these errors. The first thing we want to know is *if* an error has occurred. A lot can be said about how these error checking functions work, but I will not dive into that subject. What you need to know right now is that the **error check field** is a way for the master to confirm that the contents of the slave's response message are valid.

Find the Modbus and EtherNet/IP manual for your FLIR AX8 or FLIR A310

The details of how the messaging in Modbus is structured is beyond this course. However, the details of the configuration are specified in the slave's **register map**. It is good to know where to find this document for further reference. The register map is the same for the FLIR AX8 and the FLIR A310. It is also the same document used for details if using the *EtherNet/IP* protocol.

Go to `flir.custhelp.com`. **Click on** `download user manuals and drawings`. In the search field, **type** `modbus` and **click** `Search`.



Click on the **file number** belonging to the title `Manual: EtherNet/IP and ModBus TCP Object Models` (In my case: *T559874-en-US*). This will expand a window.

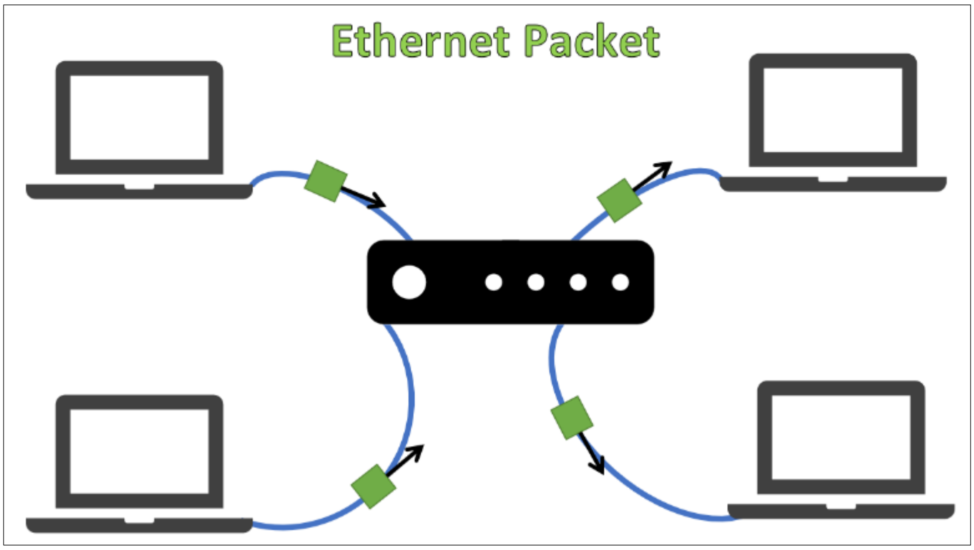
At the bottom of the window, **Click** `Right-click to download as a *.pdf file`.

EtherNet/IP

EtherNet/IP – often referred to as EIP – is another widely used protocol within the industrial arena. I will not cover the details of the protocol, that's beyond both me and the scope of this material. I will however try to give an overview of the EtherNet/IP protocol and point you in the right direction to where you may find additional information. To make this a little less confusing, let's start by separating Ethernet and IP from each other and focus on one at a time.

Ethernet

When someone says Ethernet, most of us think of a physical link between devices. Your computer might be connected to a router with an Ethernet cable. This is an example of a physical connection. However, Ethernet is not actually a physical link between devices, but instead a protocol used to send information between physically connected devices.



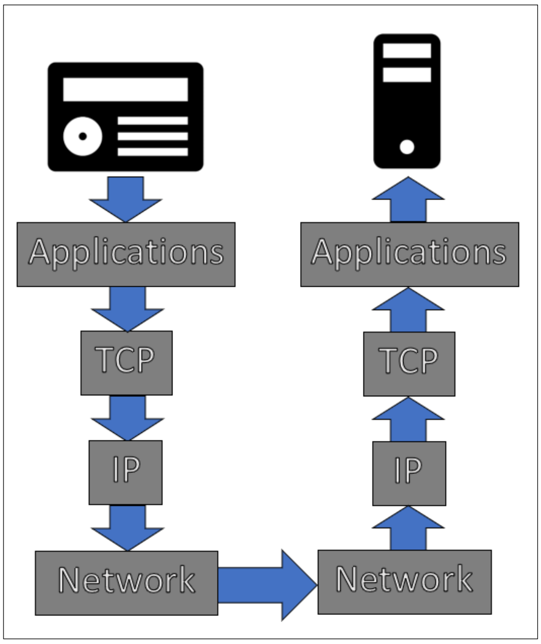
When computers and other devices communicate with each other, they send information in chunks called **packets**. Since many devices may send and receive data at the same time in a network, we need rules for how the packets should be sent and received. One common protocol used for this is the **TCP/IP** protocol. TCP stands for *Transmission Control Protocol* and IP stands for *Internet Protocol*. This is the same IP as in *IP address*. To show how the TCP/IP protocol works, let's look at the TCP/IP model.

The TCP/IP model

The TCP/IP model has 4 layers: Applications, TCP, IP and Network.

Let's assume that a device wants to send a message to another device in a network. The message will begin at the top layer – the Applications layer – and then move down the different layers. At each layer, the message will get additional information attached so that it can be properly sent and received. At the lowest layer – the Network layer – the message is packaged into an **Ethernet packet**. The message can then be sent through an Ethernet cable without collisions or other problems. Once the message is received it runs up the layers network, IP, TCP to finally arrive at the applications layer.

You can think of this kind of message as a physical letter. You write your message on a piece of paper to your pen pal. You fold the paper – *the applications layer*. You put the folded paper in an envelope and write the address – *the IP layer*. You carry the letter to the post office – *The TCP layer*. The post office accepts your letter and sort it by the address – *the network layer*.



We do want your letter to arrive at the right place, so the process has to continue by going up the layers again. But first, the postman has to deliver the letter to the mail box. In this analogy, the postman is the *Ethernet cable*. Your pen pal carries the letter from the mail box – *the network layer*. He or she then opens the envelope (*TCP* layer), unfolds the letter (*IP* layer), and reads what you have written (*applications* layer).

IP – Industrial Protocol

Now, things will get a little more confusing. IP as in EtherNet/IP stands for **Industrial Protocol**. Before when we spoke of IP in the TCP/IP model, IP stood for *Internet Protocol*. These IP abbreviations do *not* refer to the same thing, I do apologize for the potential confusion.

The *Industrial Protocol* part of EtherNet/IP combines the layers of TCP/IP with layers of the Common Industrial Protocol (CIP) to provide the necessary function for an Ethernet infrastructure to work in an industrial environment.

To summarize, we can say that EtherNet/IP is **Ethernet packets** used with the layers of **CIP** and **TCP/IP** to support **data exchange** in a control system.

EtherNet/IP manual for FLIR AX8 or FLIR A310

For further support about the EtherNet/IP structure in the FLIR AX8 and FLIR A310 it is a good idea to refer to the EtherNet/IP manual. This document is the same as the register map for Modbus. To see how to access this document, please go to the Find the Modbus and EtherNet/IP manual for your FLIR AX8 or FLIR A310 on page 191.

Machine Vision

Now I want to make a point about different types of cameras. I want you think about how an IR camera differs from your camera in your cell phone. The fact that they operate in different electromagnetic spectrums – infrared and visible light – is maybe the clearest difference. However, think about how we use an ordinary camera. When we wish to capture an image, we press a button and the camera captures the image. We can also capture images with an IR camera, but more importantly the IR camera *constantly* monitors the scene. We refer to the camera's ability to "see" as **machine vision**. It is the ***automatic*** *extraction of information from images*.

Machine vision is important in many applications within inspection and process control, as we want cameras to provide us with *automatic* inspection and *analysis* of the scene. I should also say that machine vision does not only refer to IR cameras, there are plenty of visual machine vision cameras as well. Machine vision is an umbrella term that refers to the many technologies, methods, software and hardware products in field needed to extract information from images.

Now let's take a look at two standard protocols for machine vision cameras: GigE Vision and GenIcam.

GigE Vision

As I mentioned earlier in this chapter, **image streaming cameras** are IR cameras that don't perform any analysis by itself. Instead, an external computer performs analysis of the infrared images. As we saw earlier in this chapter, the FLIR AX8 and FLIR A310 support the industry protocols **Modbus TCP** and **EtherNet/IP**. However, FLIR image streaming cameras don't support Modbus TCP or EtherNet/IP. Instead, they support the protocol **GigE Vision**, which is short for *Gigabit Ethernet for Machine Vision*. GigE Vision is an industry standard for **high-performance** industrial cameras. The protocol specifies rules for how high-speed video should be transmitted and controlled over Gigabit Ethernet networks.

GenICam

The GigE Vision protocol concerns how data is sent from the camera to the computer through physical media such as

Ethernet cables. GenICam on the other hand, is a protocol that concerns how to *access* **camera controls** and **image streams**.

Let's say that you have two cameras, a visual and an infrared. Both cameras are GenICam compatible. Then you will be able to *change the settings* in one camera just as easily as in the other, despite the two being very different cameras. This is because GenICam provides the same **application programming interface** – or API – in both cameras. When changing the setting in a camera that is GenICam compatible, you access what's called the **XML file** of the camera. Think of the XML file as the camera's DNA. Everything that *defines* the camera – its attributes – is written in the XML file.

Access the XML file of your image streaming camera

You can think of the XML file as the link between hardware and software. I will not say much more on this topic except to show you how to access the XML file of your image streaming camera.

First, you need the program **eBUS Player**. eBUS Player is included in all software installations from FLIR that supports GigE Vision cameras, such as FLIR Tools. In other words, if you have installed FLIR Tools on your computer, then eBUS Player should already be installed.

Otherwise, **download** `FLIR Tools` at `https://www.flir.com/products/flir-tools/`.

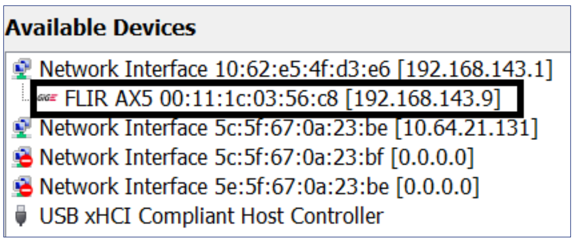
Alternatively, you may **download** `eBUS Player` directly from the Pleora Support Center (Pleora is company behind eBUS Player). **Go to** `https://supportcenter.pleora.com` and search for `eBUS Player 6.0 Toolkit`.

Connect your image streaming camera to your computer and configure the IP address of your camera.

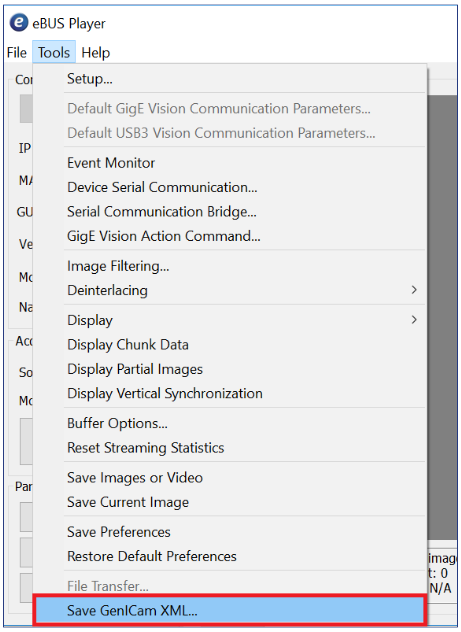
Start `eBUS Player`. **Click** the 

button.

Below `Available Devices`, **select** your IR camera and **click** OK.



Go to `Tools` in the top left and **choose** `Save GenICam XML…`.



Select the save location of the XML file. It will be saved as a zip file.

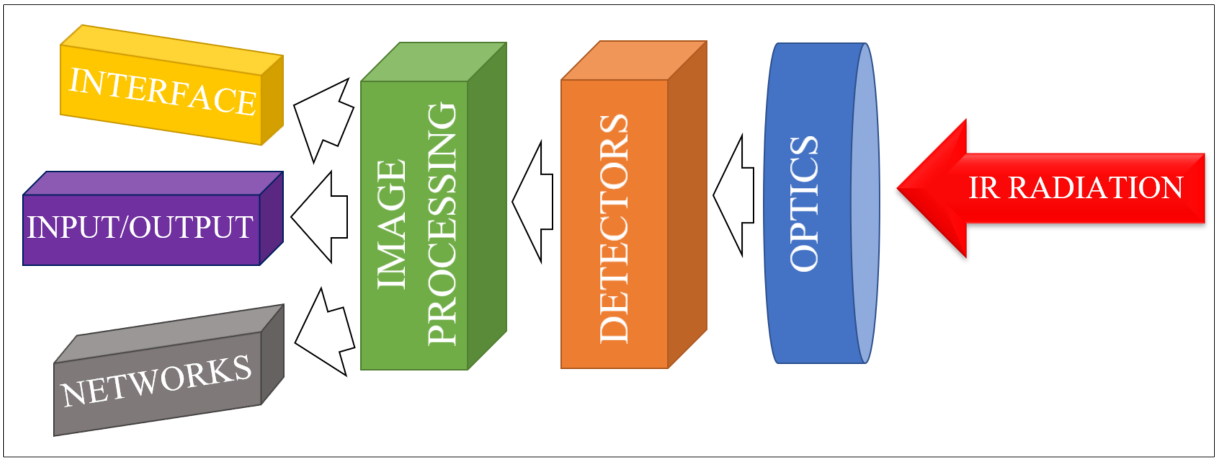
Extract the zip file. You have now successfully accessed the XML file of your IR camera.

SUMMARY

In this chapter, we have discussed digital signals and how to use the IR camera for input and output purposes. Below I have listed some of the main ideas from this chapter. Take some time to reflect and connect my key ideas to what you have read.

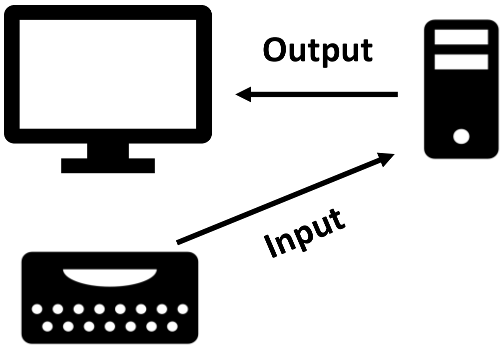
> A server is a device that provides a service to a client, one example is a mail server which receives and stores your email.
>
> SMTP is used for sending emails.
>
> POP and IMAP are examples of protocols used for retrieving email from the mail server.
>
> FTP is used for the transfer of files between computers.
>
> Fieldbus is the name of a family of industrial computer network protocols used for real-time distributed control.
>
> The term used for a system containing an organized hierarchy of control systems in a single network is SCADA.
>
> In a simplified SCADA system, there are three levels:
>
> bottom level – devices that do the work
>
> middle level – PLCs that control and manage the system
>
> top level – HMIs translates the data to humans
>
> Modbus is an automation communication protocol that provide a common language for devices.
>
> EtherNet/IP is a protocol used in the industry.
>
> Machine vision provides automatic inspection and analysis of images.
>
> GigE vision is a protocol that specify rules for how high-speed video should be transmitted and controlled.
>
> GenICam is a protocol that handles how you access the camera controls and image streams.

Input and output



In this chapter, we will cover the basics of input and output – that is, the actions triggered by the information received by the IR camera.

What do we mean by input and output? It has to do with the information that devices send and receive between each other. Here we distinguish between *input* devices and *output* devices. **Input** devices only **send** information to a computer for processing – they *put in* information into the system. **Output** devices **receive** the processed information – they do not produce any new information in the system.



Your computer keyboard is a good example of an input device. Each keystroke causes electrical signals to be *sent* as *input* to your computer. Your computer then analyses that information and sends it to your monitor which displays the letters you have typed. Your computer monitor is an example of an output device, it displays the information that your computer has processed.

Some devices are able to *both* send and receive information. We call these input/output devices – we often refer to them as **I/O devices** (pronounced “eye-oh”). In fact, the IR camera is an I/O device. In this chapter, we will cover the basics of how the IR camera can be used for input and output purposes. I use the FLIR AX8 to provide you with concrete examples, but the basic principles of I/O are the same for all automation cameras.

Objectives

When you have worked through this part, my aim is that you will be able to answer the following questions.

> What is the difference between analog and digital signals?
>
> How do I power the FLIR AX8 with an external power supply?
>
> How does the FLIR AX8 determine a digital input signal as high or low?
>
> How do I make a lamp indicate that an alarm has been triggered?

For you to know how to answer these questions, I have divided this chapter into four parts:

> Analog signals vs. digital signals
>
> I/O cable connections
>
> Input
>
> Output

For you to get the most out of this chapter, it is good if you have accesses to

- A M12 to pigtail cable – T128391ACC (FLIR AX series)
- A 12/24 V DC power supply (to supply power to the FLIR AX8 digital output)

This will allow you to set up everything yourself and not only read about it.

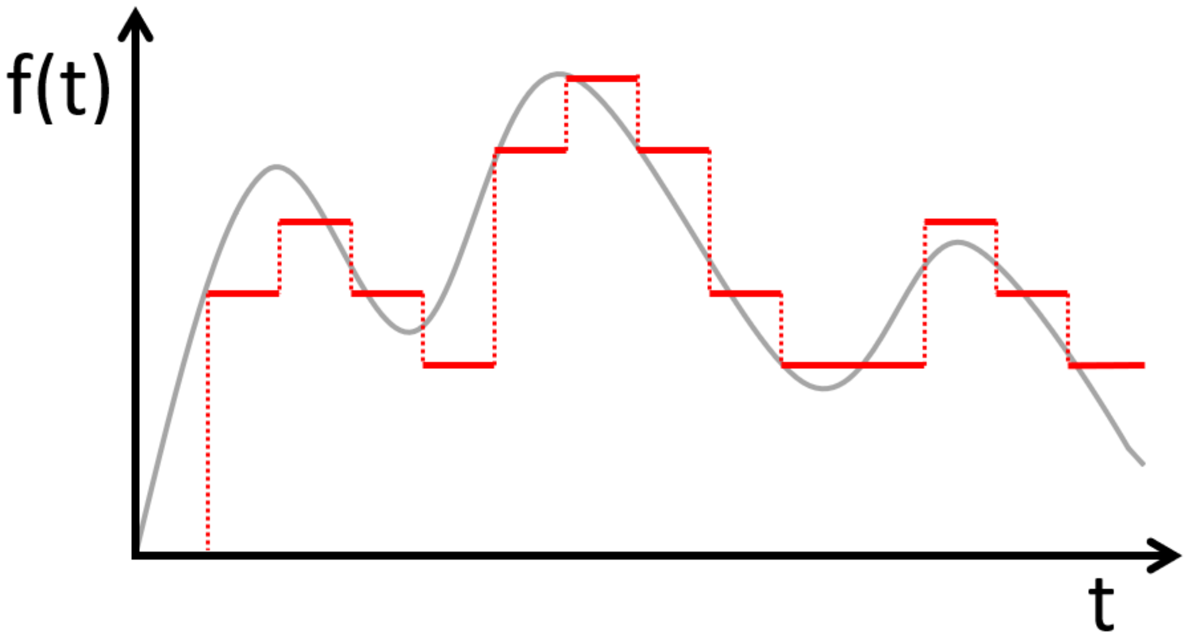
Analog signals vs. digital signals

In the beginning of this chapter, I said that input and output have to do with information being sent and received among devices. More specifically, **signals** are used to send the information. We make the distinction between two types of signals: **analog** and **digital**. We say that **analog** signals are **continuous**, which means that they can take *any value* in a certain range. On the other hand, **digital** signals are **discrete**. It means that discrete signals only take certain values.

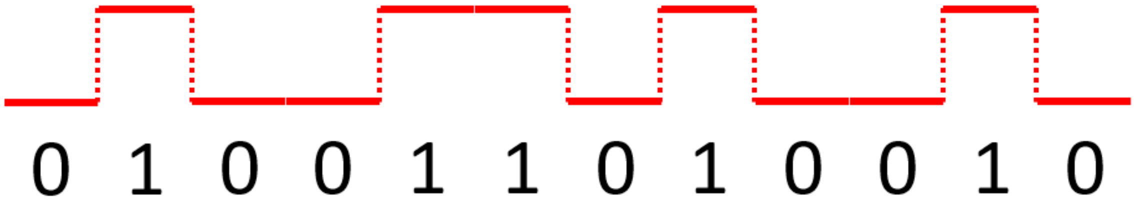
You can think of it like this: **Discrete** things we can **count**, **continuous** things we can **measure**. For example, the number of people in a room is *discrete*, because we can count them; one, two, three, and so on. The height of those people is *continuous* data, because the height can be *any* value within the range of human heights.

Another way of thinking about it is this: we can count discrete things on our fingers, one, two, three, and so on. We cannot count continuous things on our fingers. An example would be: How many points are there in a line from zero to one? You'll have a point at one half. Another at half of the half, and another at half of the half of the half, and so on. There are infinitely many, because the line is continuous.

Take a look at the graph below. The grey curve shows an analog signal and the red curve shows a digital signal. Notice that the red signal only take certain values while the grey signal take all values in between.

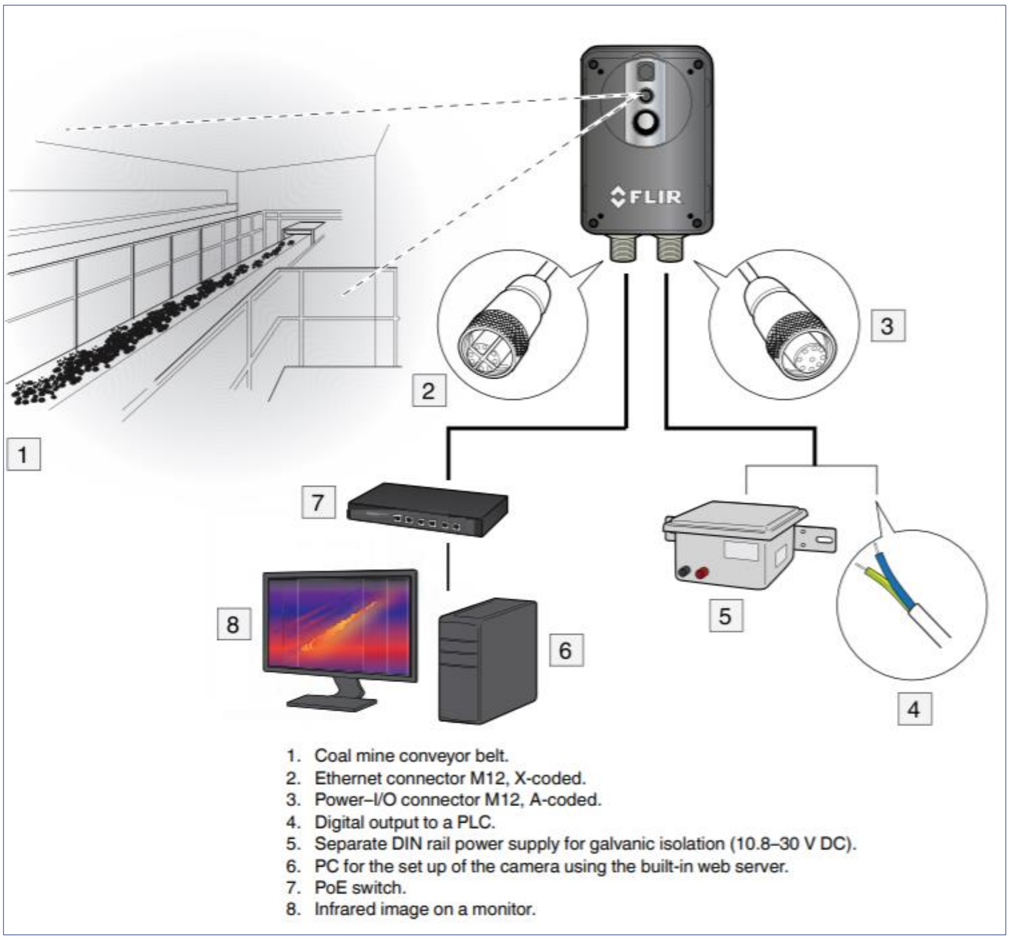


In this chapter we will focus on **digital signals** – discrete signals. More specifically, we will focus on **binary digital signals**. As I've mentioned in earlier chapters, binary numbers only contain ones and zeros. This means that a binary digit can only be in *two different states* – 1 or 0. The same is true for binary digital signals. They can only be in two different states – high or low – on or off – true of false – 1 or 0. There are many analogies and I hope you see the principle of having only two possible levels, as shown in the figure below.



Next, we will take look at how we can set up our IR camera to work with digital signals. Let’s begin by looking at a typical system overview.

Typical system overview



This is a typical system overview which I have taken from the FLIR AX8 manual. The figure above shows a coal mine conveyor belt being monitored by a FLIR AX8 (1). The camera is connected with an Ethernet connector M12, X-coded (2) to a PoE switch, monitor (8) and PC (6).

The cable on the right side on the FLIR AX8 is the Power-I/O connector M12, A coded (3). In (5) we see that the cable can be connected to a power supply to power the FLIR AX8. We may also send digital signals with the FLIR AX8 (4). Although the picture only says digital *output* to a Programmable Logic Controller (PLC), the FLIR AX8 can handle *both* digital *input* and *output* (see chapter SCADA simplified – three levels on page 186 for more on PLC).

Box tip – Digital output signals need power

Digital signals can only be sent if they are supplied with power. We know that the FLIR AX8 can be supplied with power via a *Power over Ethernet* (PoE) switch.

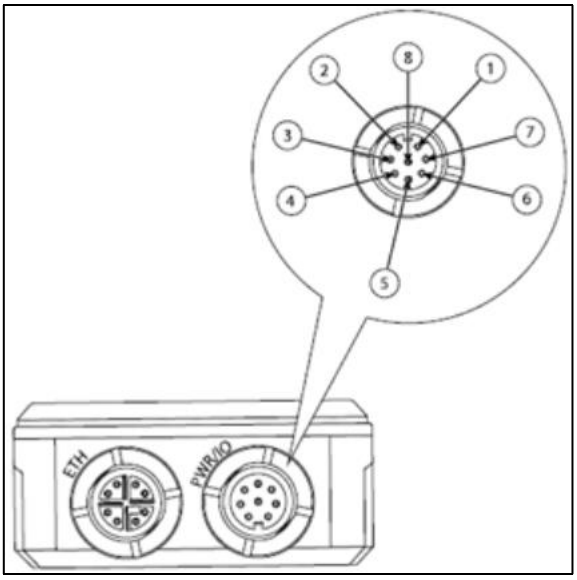
Then we wonder: can the PoE switch also power digital output signals?

The answer is **no**, the power supplied by a PoE switch is not enough to power digital outputs. We need an external power supply if we wish to enable digital output signals.

I/O cable connections

In this chapter, we will focus on the cable on right side of the FLIR AX8, called the **Power-I/O connector**. The name tells us what we can use the cable for. "Power" refers to the fact that you can use this cable to power the FLIR AX8, instead of using a Power over Ethernet (PoE) switch. "I/O" refers to the fact that we use this cable for input and output purposes. Now let's take a closer look on the different features of this cable.

In the table below, I will give you the configuration and a short description of each pin.



Pin	Configuration	Cable color on cable P/N T128391ACC	Description
1	EXT_POWER	Orange/white	Used to power the camera 12/24 V DC (+)
2	DIGIN	Orange	Digital input signal (+)
3	DIG_PWR	Green/white	Digital output power (+)
4	DIG_RTN	Green	Digital input/output power and signal (–)
5	DIGOUT	Blue	Digital output signal (+)
6	Not connected	Blue/white	Not connected
7	Not connected	Brown/white	Not connected
8	GND	Brown	Used to power the camera 12/24 V DC (–)

Power the FLIR AX8 with an external power supply

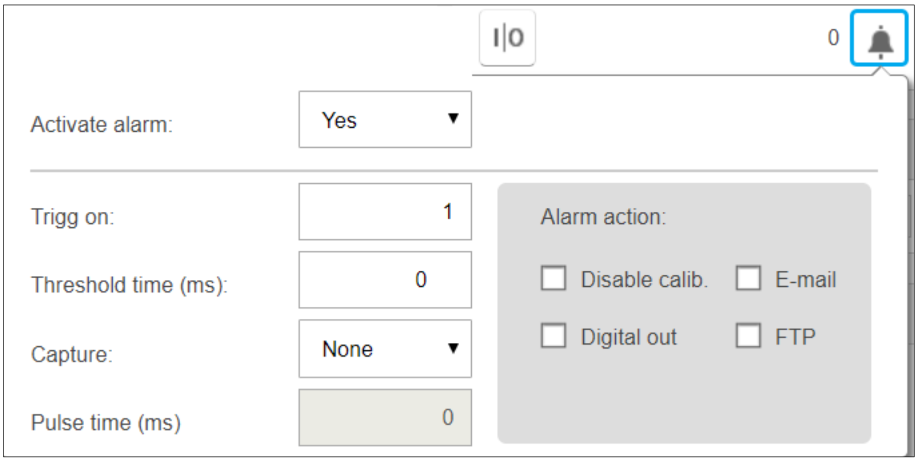
Instead of using PoE, you may power the FLIR AX8 by connecting pins 1 and 8 to an external 12/24 V DC power supply. Pin 1 is called EXT_POWER which stands for *external power*. Pin 8 is call GND which stands for *ground*.

In the table above you see that pins 1 and 8 have the same description except for different signs at the end, + and -. This means that the *positive pole* of the power supply should be connected to pin 1 (orange/white cable) and pin 8 (brown cable) should be connected to the *negative pole* of the power supply.

Input

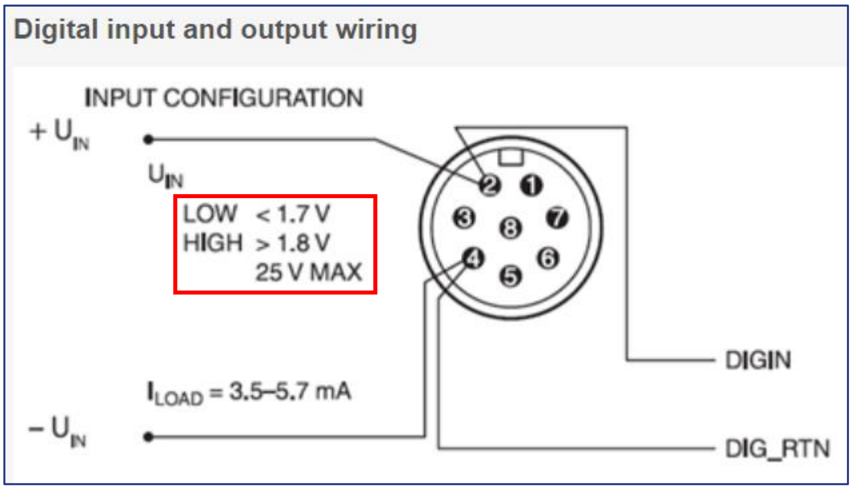
In earlies chapters we've talked about different useful features and functions found in most IR cameras. In the FLIR AX8 interface however, there is one icon found in the `Measurements & alarm` pane that we've not said much about – this one.

This alarm works differently than most other alarms. Instead of being triggered depending on a temperature condition, the digital input alarm is triggered when a digital input signal sent to the IR camera. Below you can see the options for this alarm.



You may recognize most of the options for this alarm, such as the *threshold time* and the different *alarm actions* (see chapter Alarms on page 97). However, **Trigg on** is new to us. As I said earlier, you can think of digital signals as only being in two different states: high or low. *Trigg on* determines whether the alarm should *trigger* on a high or low signal. Trigg on can only be set to 1 or 0. 1 refers to a *high signal* and 0 refers to a *low signal.*

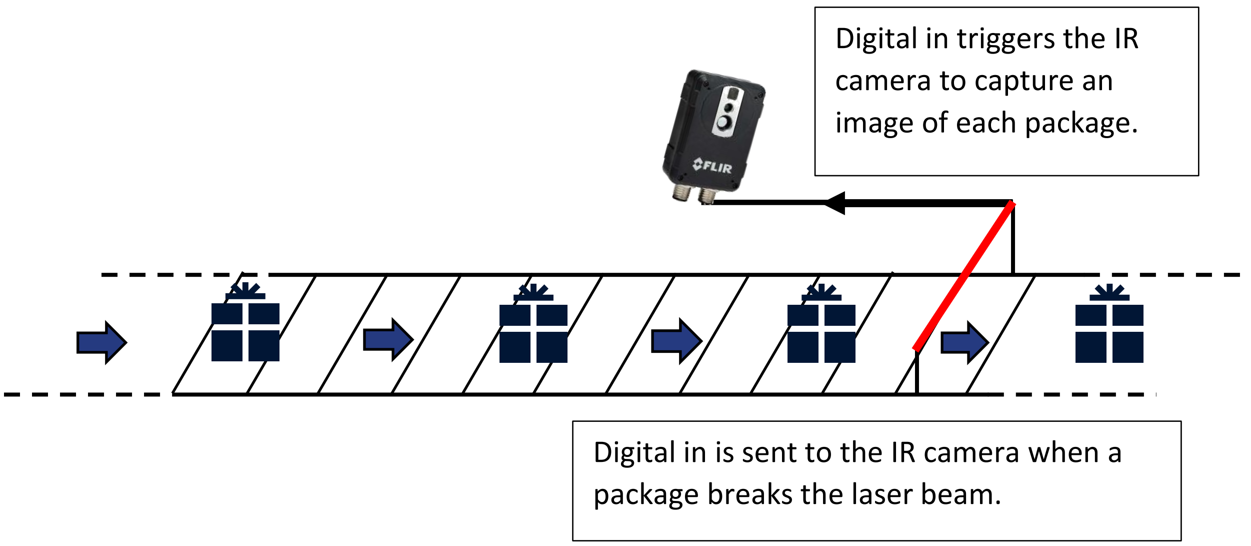
How does the camera then determine a signal to be high or low? To answer this, take a look at the wiring for the input configuration in the figure below.



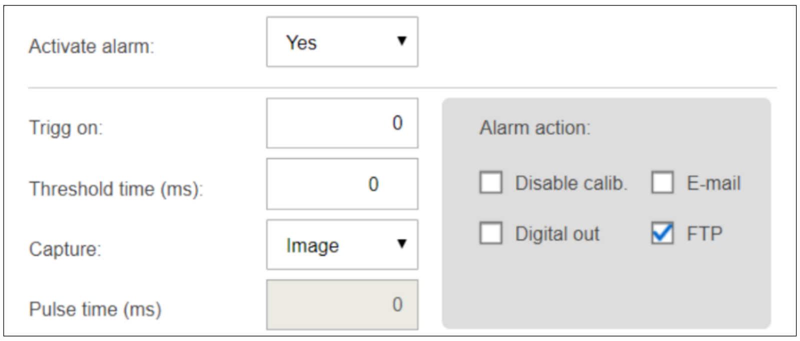
In order to recive digital input signals, the IR camera needs to be connected to an external power supply. You see that we use pins 2 (DIGIN – digital in) and 4 (DIG_RTN – digital return) to crete a closed circuit. The **power supply voltage** determines if the signal is *high* or *low*. The FLIR AX8 will register a digital input as *high* when the voltage on the external power supply is *above 1.8 V*. It will register the digital input signal as *low* when the voltage is *below 1.7 V*.

An example using Digital in

Let's take a look at an example where we can use the IR cameras ability to register digital input signals. The figure below shows a conveyor belt with packages. We want to capture an image of each package and store it on an FTP server (see chapter Set up a local FTP server and send an image or video to your FTP site on page 167).



The setup in the image is as follows: Packages are travelling on the conveyor belt, and when they break the laser beam, the signal to the FLIR AX8 goes low. An alarm is set in the FLIR AX8 to capture an image and send it to an FTP server whenever the digital in is low. That is, we have set Trigg on: 0.



Limitations of uncooled detectors

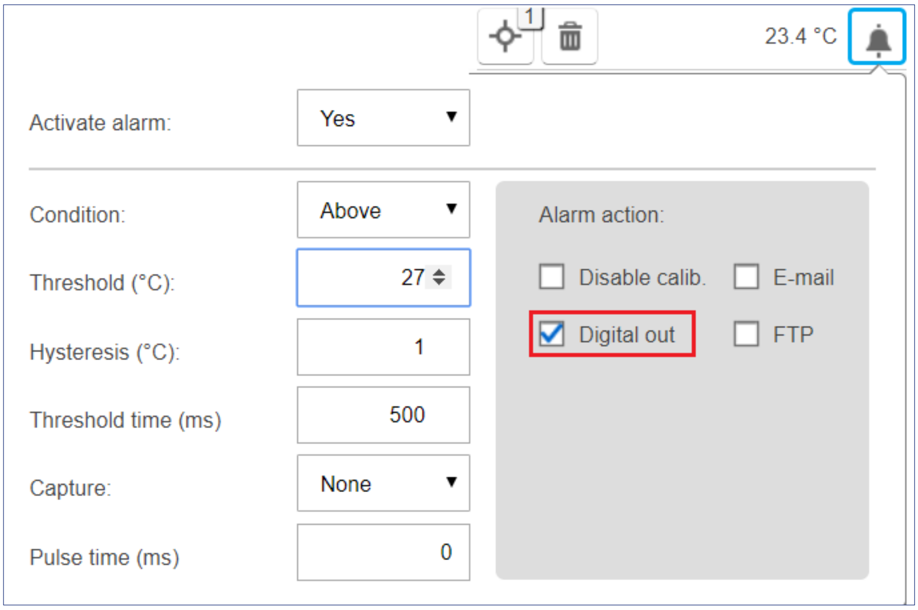
It is good to know that the captured images will *not* be snapshots. What I mean by this is that the FLIR AX8 and other uncooled IR cameras will be able to save an image instantly when it receives the digital in. Instead, there will be some delay in the image processing cycle. This depends on where in the process the current image that is being processed is. Please refer to chapter Image processing on page 130, where we cover this more in detail. I just want to remind you of these uncertainties and limitations. In this example, it might not be a big deal. However, depending on the application, these limitations can make some IR cameras unsuitable for the job.

Output

One of the simplest examples which demonstrates the use of digital output with an IR camera is to make a lamp light up when an alarm is triggered. I will now walk you through how to do this. It consists of two parts: set up an alarm and connect the IR camera in a circuit.

Set up an alarm in the FLIR AX8 web interface

If you are unsure about how to set up an alarm in general, please see the Analytics and alarms chapter on page 72. You are free to choose whichever alarm you want. Since we are discussing digital output, just remember to **check** the box `Digital out`. In my case, I have set the alarm to trigger when the temperature in Spot 1 is above 27 °C.

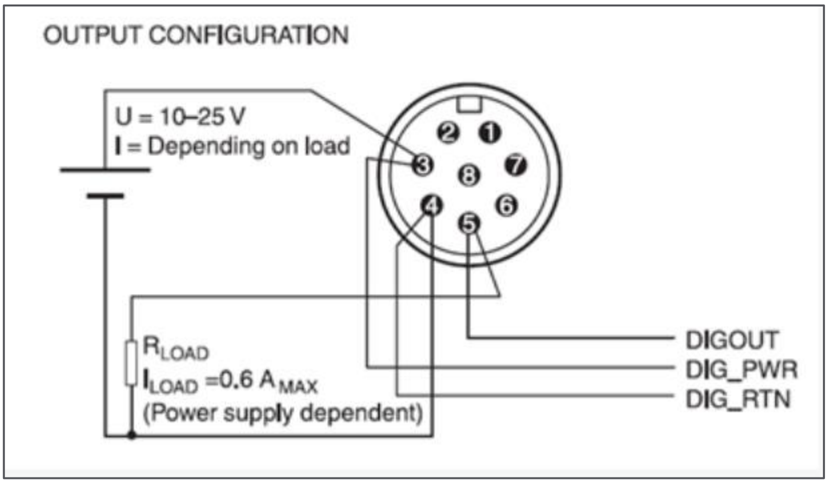


The `Pulse time` determines for how long the digital output signal will be high. For example, if we set the Pulse time to 2000 milliseconds, the FLIR AX8 will send digital out signals for 2 seconds. This will cause our lamp will light up for 2 seconds. I have set the Pulse time to 0, this will make the FLIR AX8 to send digital out signals as long as the alarm is triggered.

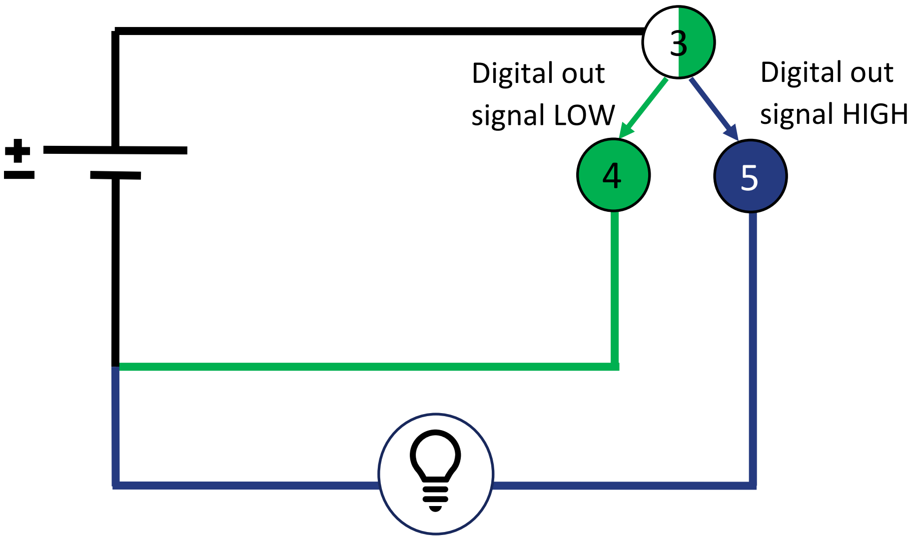
When the alarm is triggered, the IR camera will send out a digital output signal. But only if we have correctly connected the camera to an external power supply. Let's cover this next.

Enable digital outputs with a power supply

To do this we have to refer to the output configuration presented below.



We see that in order to make our lamp light up when an alarm is triggered, we use cables 3 (DIG_PWR – digital power), 4 (DIG_RTN – digital return) and 5 (DIGOUT – digital out). If you feel that the diagram above is a little confusing, let me try to simplify. Take a look at the diagram below.



The current will always flow into the camera through cable 3 (green/white – DIG_PWR). Depending on if the alarm has been triggered or not, the current will take different paths. If the alarm is not triggered, then the digital output signal is *low*, and the current will flow through the camera and out through cable 4 (green – DIG_RTN).

If the alarm is triggered, the digital output signal will be *high*. This causes the current to flow through cable 5 (blue - DIGOUT). This will cause our light bulb to shine and indicate that the alarm has been triggered.

Using Reversed Logic to detect camera or cable failure

Up until now, we have covered the very basics of input and output. Now I want to say something about *straight* and *reversed logic*. In the section above, I discussed how to trigger an digital output signal to make a light bulb shine. In that example we used *straight logic*. We wanted the alarm to trigger once the temperature went *above* 27 °C and this is exactly what we set up in the alarm options. However, what if there is a camera or cable failure? Then the lamp won't turn on despite the temeperature being above the threshold of 27 °C. Is there a way to detect camera and cable failure?

Yes, one solution to this is to use **reversed logic**. We set the alarm to trigger when the temperature is *below* 27 °C. This will make the camera to send a digital out when everything is normal. In other words, all is well as long as the lamp shines – the temperature is not above 27 °C and we don't have any camera or cable failure. If we would have a camera or cable failure, no current will be able to flow through the circuit. This will cause the lamp to be turned off. Similarily, the lamp will turn off when the temperature goes *above* 27 °C.

Now when the lamp is turned off, we will know that *something* is wrong. We won't immidiately know if the alarm is triggered or if there is something wrong with the cable/camera. However, reversed logic gives us a way to *discover* camera and cable failure, which straight logic will not.

SUMMARY

In this chapter, we have discussed digital signals and how to use the IR camera for input and output purposes. Below I have listed some of the main ideas from this chapter. Take some time to reflect and connect my key ideas to what you have read.

> Analog signals are continuous and digital signals are discrete.
>
> The FLIR AX8 can be powered by an external power supply.
>
> Digital in can be used to trigger an alarm in the FLIR AX8.
>
> Power over Ethernet (PoE) cannot supply enough power for digital output signals. An external power supply is needed for this.
>
> Reversed logic can be used to detect camera and cable failure.

Software

FLIR has developed several software packages to help you analyze your IR images and videos. They have also developed different Software Development Kits (SDKs) to help you program everything you need from your IR camera. I will not go through how to use the different software packages – but I will tell you a little bit about the features of FLIR Tools and FLIR Atlas SDK, where you can find them, and where you can read more about how to use them.

FLIR Tools

FLIR Tools allows you to import, edit, and analyze IR images, and turn them into professional PDF inspection reports. It also allows you to view live radiometric video.

To download FLIR Tools, enter the following address in your web browser and follow the instructions displayed there.

`https://www.flir.com/products/flir-tools/`

There are several instructional video clips on YouTube, which you can find by typing the following address in your web browser.

`https://www.youtube.com/playlist?list=PL0554AC28BBF87009`

You can also search for `FLIR Tools and Tools+ Infrared Training Center` on YouTube.

There are also text manuals for FLIR tools found through the following addresses.

ITCs user manual for FLIR Tools:

`http://support.flir.com/answers/A1568/FLIR%20Tools%20User%20Guide%20v2.1.1.pdf`

FLIRs user manual for FLIR Tools:

`https://assets.tequipment.net/assets/1/26/FLIR_Tools_Plus_Manual.pdf`

FLIR Atlas SDK

FLIR has an SDK called Atlas SDK that should always be the main entry point for developing applications interacting with FLIR cameras or managing IR images from FLIR cameras.

The FLIR Atlas SDK is a software development kit that enables developers to create applications. Supported by help files and sample code, developers can add functionality or collaborate with other FLIR products to get the result they want in their application.

Key features (depending on camera model)

- Supports communication and streaming using FireWire, Gigabit, RTSP, and USB interfaces.
- Gives the user full control of the camera.
- Supports recording of images using FireWire, Gigabit, RTSP, and USB interfaces.
- Converts 16-bits signal pixels into temperature data, for maximum user flexibility.
- Allows 16-bit temperature linear, histogram, and signal outputs.

To download and install Atlas SDK, enter the following address in your web browser and follow the instructions displayed there.

`https://flir.custhelp.com/app/answers/detail/a_id/1275`

FLIR Products

In this part, I will present a few of FLIRs IR cameras and their features. FLIR has lots more products than presented here, so feel free to visit the FLIR website and check them out!

FLIRs website: `https://www.flir.com/`

	FLIR AX8	FLIR A310	FLIR A315	FLIR A615	FLIR Ax5
Smart sensor camera	X	X			
Image streaming camera		X	X	X	X
Support for EtherNet/IP fieldbus protocol	X	X			
Support for Modbus TCP field bus protocol	X	X			
Built in analysis functionality	X	X			
MPEG-4 streaming	X	X			
PoE	X	X			X
Built-in web server	X	X			
GigE compliant			X	X	X
GenICam compliant			X	X	X
GPIO[1]	X	X	X	X	X
Signal, temp. linear streaming			X	X	X
Radiometric streaming		X	X	X	X

[1] General-Purpose Input/Output

Glossary

absorption (absorption factor)	The amount of radiation absorbed by an object relative to the received radiation. A number between 0 and 1.
ADU	Application Data Unit. Refers to a message in Modbus communication consisting of slave address, function code, data and error check.
aperture	The opening in the IR camera through which the incoming radiation passes to get to focus on the detectors.
API	Application Programing Interface. Set of communication protocols and tools for building software.
atmosphere	The gases between the object being measured and the camera, normally air.
autoadjust	A function making a camera perform an internal image correction.
autopalette	The IR image is shown with an uneven spread of colors, displaying cold objects as well as hot ones at the same time.
blackbody	Totally non-reflective object. All its radiation is due to its own temperature.
blackbody radiator	An IR radiating equipment with blackbody properties used to calibrate IR cameras.
bit depth	Refers to a measurement of the number of colors available in an image.
calculated atmospheric transmission	A transmission value computed from the temperature, the relative humidity of air and the distance of the object.
cavity radiator	A bottle shaped radiator with an absorbing inside, viewed through the bottleneck.
color temperature	The temperature for which the color of a blackbody matches a specific color.
compressed streaming	Refers to the image streaming where the information in each pixel is a color value.
conduction	The process that makes heat diffuse into a material.

continuous adjust	A function that adjusts the image. The function works all the time, continuously adjusting brightness and contrast according to the image content.
convection	Convection is a heat transfer mode where a fluid is brought into motion, either by gravity or another force, thereby transferring heat from one place to another.
data field	Part of message in Modbus communication. Specifies how much in the memory register the slave is requested to handle.
default gateway	The device that data is sent to if no recipient of the data is specified.
depth of field	The maximum depth of a scene that stays in focus.
detector time constant	The time it takes for a detector to produce a signal.
dual isotherm	An isotherm with two color bands, instead of one.
emissivity (emissivity factor)	The amount of radiation coming from an object, compared to that of a blackbody. A number between 0 and 1.
emittance	Amount of energy emitted from an object per unit of time and area (W/m^2).
estimated atmospheric transmission	A transmission value, supplied by a user, replacing a calculated one.
ethernet/IP	EIP. A protocol used within the industrial arena.
external optics	Extra lenses, filters, heat shields etc. that can be put between the camera and the object being measured.
filter	A material transparent only to some of the infrared wavelengths.
fieldbus	Refers to the collective name of industrial computer network protocols used for real-time distributed control.
f-number	A measure of the amount of radiation reaching the detectors.
focal length	Determines where the incoming radiation will focus in an optic system.

FOV	Field of view: The horizontal angle that can be viewed through an IR lens.
FPA	Focal plane array: A type of IR detector.
FTP	File Transfer Protocol.
function code	Part of a message in Modbus communication. Specifies what action the slave should perform.
gateway	Controls the flow of information coming in and out of a network.
graybody	An object that emits a fixed fraction of the amount of energy of a blackbody for each wavelength.
histogram equalization	Equalizes color distribution in the image, creating more contrast.
HMI	Human Machine Interface. Top level in SCADA systems.
host ID	Part of IP address. Identifies a specific device in a network.
hysteresis	The interval in which the temperature value can vary without causing a change in the trigger.
IFOV	Instantaneous field of view: A measure of the geometrical resolution of an IR camera.
image correction (internal or external)	A way of compensating for sensitivity differences in various parts of live images and also of stabilizing the camera.
image streaming camera	A type of IR camera who streams in radiometric format and cannot perform analysis functions by itself.
IMAP	Internet Message Access Protocol. Protocol for retrieving email from the mail server.
infrared	Non-visible radiation, having a wavelength from about 2-13 µm.
IP address	Identification of network and a specific device within the network.
IR	Infrared

isotherm	A function highlighting those parts of an image that fall above, below or between one or more temperature intervals.
isothermal cavity	A bottle-shaped radiator with a uniform temperature viewed through the bottleneck.
Laser LocatIR	An electrically powered light source on the camera that emits laser radiation in a thin, concentrated beam to point at certain parts of the object in front of the camera.
laser pointer	An electrically powered light source on the camera that emits laser radiation in a thin, concentrated beam to a point at certain parts of the object in front of the camera.
level	The center value of the temperature scale, usually expressed as a signal value.
LWIR	Long Wavelength InfraRed. The radiation with wavelengths within 8 μm – 13 μm.
machine vision	An umbrella term for technologies that automatic extract information from digital images.
manual adjust	A way to adjust the image by manually changing certain parameters.
microbolometer	A type of thermal detector who detects thermal radiation through a change in resistance.
modbus	An automation communication protocol used in the industrial arena.
MWIR	Medium Wavelength InfraRed. The radiation with wavelengths within 2 μm – 5 μm.
NETD	Noise equivalent temperature difference. A measure of the image noise level of an IR camera.
network	A group of devices that communicate with each other.
network ID	Determines the network to which a device belongs.
noise	Undesired small disturbance in the infrared image.
NTP	Network Time Protocol. Protocol for clock synchronization.

NUC	Non-Uniformity Correction. Synchronizes and corrects signal from detectors.
object parameters	A set of values describing the circumstances under which the measurement of an object was made, and the object itself (such as emissivity, reflected apparent temperature, distance, etc.)
object signal	A non-calibrated value related to the amount of radiation received by the camera from the object.
palette	The set of colors used to display an IR image.
PDU	Part of message in Modbus communication. Consists of function code and data field.
pixel	Stands for *picture element*. One single spot in an image.
PLC	Programmable Logic Control. A level in a SCADA system that control and manage the system.
PoE	Power over Ethernet. Allows both data connection and power supply to devices.
POP	Post Office Protocol. Protocol for retrieving email from the mail server.
power-I/O connector	Outlet for power and input/output.
port	Identifies a specific network process.
pseudo color	Colors in an image that is not the same as the visual image of the same scene.
query	Interaction between master and slave in Modbus communication.
radiance	Amount of energy emitted from an object per unit of time, area and angle ($W/m^2/sr$).
radiant power	Amount of energy emitted from an object per unit of time (W).
radiation	The process y which electromagnetic energy, is emitted by an object or a gas.
radiator	A piece of IR radiating equipment.

radiometric streaming	Refers to the image streaming where information other than a color value is given in each pixel.
range	The current overall temperature measurement limitation of an IR camera. Cameras can have several ranges. Expressed as two blackbody temperatures that limit the current calibration.
reference temperature	A temperature which the ordinary measured values can be compared with.
reflection	The amount of radiation reflected by an object relative to the received radiation. A number between 0 and 1.
relative humidity	Relative humidity represents the ratio between the current water vapor mass in the air and the maximum it may contain in saturation conditions.
reversed logic	A way to connect an IR camera that will detect camera or cable failure.
RTSP	Real Time Streaming Protocol. Protocol used to control media streaming.
sampling	Converting continuous colors into discrete color pixels.
saturation color	The areas that contain temperatures outside the present level/span settings are colored with the saturation colors. The saturation colors contain an ‘overflow’ color and an ‘underflow’ color. There is also a third red saturation color that marks everything saturated by the detector indicating that the range should probably be changed.
SCADA	Supervisory Control and Data Acquisition. An organized hierarchy of control systems connected to a single network that allows for real-time control and monitoring.
SDK	Software Development Kit. A set of software development tools for software development and programming.
smart sensor camera	A type of IR camera who streams in compressed format, has an internal computer capable of performing analysis.
server	Computer program or device that provides a functionality to other devices.

SMTP	Simple Mail Transfer Protocol. Protocol used for sending emails.
socket	IP address and port.
span	The interval of the temperature scale, usually expressed as a signal value.
spectral (radiant) emittance	Amount of energy emitted from an object per unit of time, area and wavelength ($W/m^2/\mu m$).
spectral range	Defines an interval of wavelengths.
subnet mask	Defines what part of the IP address is the network ID, and what part is the host ID within a network.
SWIR	Short Wavelength InfraRed. The radiation with wavelengths within 1 µm – 2 µm.
TCP/IP model	Protocol that specifies rules for how information should be sent and received.
temperature difference, or difference of temperature	A value which is the result of a subtraction between two temperature values.
temperature linear	Function in image streaming cameras which sends a signal that is linearly dependent on temperature.
temperature range	The current overall temperature measurement limitation of an IR camera. Cameras can have several ranges. Expressed as two blackbody temperatures that limit the current calibration.
temperature scale	The way in which an IR image currently is displayed. Expressed as two temperature values limiting the colors.
thermogram	infrared image.
transmission (or transmittance) factor	Gases and materials can be more or less transparent. Transmission is the amount of IR radiation passing through them. A number between 0 and 1.
transparent isotherm	An isotherm showing a linear spread of colors, instead of covering the highlighted parts of the image.

visual	Refers to the video mode of an IR camera, as opposed to the normal, thermographic mode. When a camera is in video mode it captures ordinary video images, while thermographic images are captured when the camera is in IR mode.
XML file	Extensible Markup Language file. File that defines the rules for encoding documents.

Index

A

alarm 41, 97, 162
alarm action 41, 103
alarm condition 100
alarms........ 97
analog signals 199
analysis function 40, 134
analytics 72
aperture 117
API 195
apparent temperature 46
atmospheric temperature 63

B

base-2........ 142
binary 142
binary numbers 151
bit 142
bit depth........ 130
blackbody 48
box measurement 76
box tips........ 7
byte 142

C

capture 103
cavity 54
CIDR........ 150
client-server model 162
clock synchronization 166
color alarm 83
compressed streaming 133
configure the IP address 14
cooled detector 120

D

deadband 101
decimal system........ 151
default gateway 17, 18, 145
delta measurement........ 80
depth of field........ 116
detector pitch........ 121
detector time constant 122, 126
determine the emissivity 56
difference calculation........ 81
digital in........ 205
digital signals........ 199
distance 63
dotted decimal format 142

E

EIP 192
emissivity 44, 46, 47, 49, 56
ethernet 192
external IR window 66

F

field of view (FOV)........ 111
fieldbus 186
FLIR Atlas SDK........ 210
FLIR FOV calculator 112
FLIR GEV Demo 26, 28
FLIR IP Config........ 11, 145
FLIR Tools 210
f-number 117
focal length 116
focal plane array 111
focus........ 116
FTP server........ 164

G

gateway........ 156
GenICam........ 194
geometry........ 54
germanium........ 116, 117
GigE Vision 194
global shutter 123
greyscale 131

H

histogram equalization 132
HMI 186
horizontal field of view (HFOV) 111
host ID 141
hysteresis 100, 108

I

image streaming camera 131, 188, 194, 212
input........ 204
IP address........11, 14, 18, 141
IPv4 143
IR Monitor 21, 23, 33, 35
iso-coverage 88
isotherm coverage 79
isotherms 83

L

local address 157
local network 156, 159
local object parameters 36
local port 158
LWIR 126

M

mail server 162, 163
mask 92
master/slave architecture 188
material 49
microbolometer 121
minimum focus distance 116
modbus 188
modbus communication 188
MPEG-4 streaming 212
MWIR 127

N

NETD 124
network 140
network ID 141
noise 125
non-uniformity correction (NUC) 41, 123

O

object parameters 35, 43
opaque 45
output 206
overlay graphics 37

P

palette 36
pixel 124
Planck's law of radiation 48
PLC 186
port 157
port forwarding 158
port number 157
power-I/O connector 202
protocols 162
pseudo colors 32
public address 157
public ports 159
pulse time 104

Q

queries 188

R

radiometric streaming 133
reflected temperature 63
register map 191
registers 190
relative humidity 63
response time 122
retrieve email 163
reversed Logic 208
rolling shutter 123
router 156, 159

S

sampling 135
SAQ 7
SCADA 186
scale .... *See span*
schedule 95
sending emails 163
server 162
set up a local FTP server 167
set up a local mail server 177
set up an alarm 106
smart sensor camera 9, 133, 188, 212
socket 158
span 36
spatial resolution (IFOV) 112
spectral band 126
spectral range 126
spot measurement 73
subnet mask 17, 143
surface structure 49
SWIR 127

T

TCP/IP 192
temperature linear 131
the law of reflection 52
thermal detector 120
thermal MSX 38
threshold 100
threshold time (delay) 102
Trigg on 204

V,W

web interface 18, 33, 35
vertical field of view (VFOV 111
viewing angle 52
window transmissivity 66, 69

X

XML file 195