Miika Marttinen

Thermal Camera Prototype for Developers

Metropolia University of Applied Sciences

Bachelor of Engineering

Information technology

Thesis

25 October 2017

Author Title Number of Pages Date	Miika Marttinen Thermal Camera Prototype for Developers 43 pages + 3 appendices 25 October 2017
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Professional Major	Smart systems
Instructors	Janne Salonen, Principal Lecturer

The purpose of this thesis was to design and build a thermal camera prototype. The work was carried out for Metropolia University of Applied Sciences in collaboration with a festival organizer and emergency services. From the customer point of view the goal was to find out whether a thermal camera can be used to monitor festival audiences. The customer needed an overall solution that could be used in different applications in the future.

During the design phase different hardware choices were compared to find a cost efficient and effective solution. A cost estimation for one camera was made and also calculations for minimum requirements. A camera program was designed by testing different approaches before the actual implementation.

Thermal data was produced during a field test over the festival weekend. Five copies of the camera were made and tested during the music festival. After that, the test data was processed to be visible for human eye for visual analyzing.

The produced data clearly showed individual humans and groups of people, and estimations regarding amounts of people in groups could be made. This proves that thermal cameras can be used to monitor festival audiences. The camera can also be used in other applications in the future. Based on the results achieved during the field test the data transfer method has to be changed if the camera is used for security purposes. The outcome of the project was successful and the cameras will likely be used in school projects later.

Keywords	embedded systems, thermal camera, machine vision, embedded system design, smart systems

Tekijä Otsikko Sivumäärä Päiväys	Miika Marttinen Lämpökameran prototyyppi kehityskäyttöön 43 sivua + 3 liitettä 25.10.2017
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Smart systems
Ohjaaja	Yliopettaja Janne Salonen

Insinöörityön tarkoitus oli suunnitella ja rakentaa lämpökameran prototyyppi. Työ toteutettiin ammattikorkeakoululle yhteistyössä festivaalijärjestäjän ja pelastuslaitoksen kanssa. Asiakkaan kannalta tavoitteena oli selvittää, voiko lämpökameraa käyttää festivaaliyleisön valvomiseen. Lämpökameran tuli olla yleinen ratkaisu ja käytettävissä erilaisiin tarkoituksiin tulevaisuudessa. Projekti toteutettiin toisen opiskelijan kanssa osittain yhteistyönä.

Suunnitteluvaiheessa erilaisia laitteistovaihtoehtoja vertailtiin edullisen ja toimivan ratkaisun löytämiseksi. Kamerasta tehtiin hinta-arvio, ja lisäksi tehtiin laskelmia kameran vähimmäisvaatimuksille. Ohjelmisto suunniteltiin testaten eri toteutustapoja ennen varsinaista toteutusta.

Työssä tuotettiin infrapunadataa kameroilla ihmismassoista viikonlopun ajan. Kamerasta rakennettiin viisi samanlaista prototyyppiä, ja niitä testattiin musiikkifestivaalin aikana. Testauksen jälkeen data prosessoitiin ihmissilmälle näkyväksi analyysia varten.

Tuotetusta datasta ilmeni, että kamerat soveltuvat yleisön valvomiseen. Yksittäiset ihmiset ja isommat joukot erottuvat selvästi, ja arvioita ihmismääristä voitaisiin tehdä. Kameraa voidaan käyttää jatkossa muihinkin tarkoituksiin. Testin perusteella voidaan todeta, että turvallisuustarkoituksiin käytettäessä kameran datansiirtotapa tulee kuitenkin vaihtaa. Projektin lopputulos oli onnistunut, ja kamera otettaneen käyttöön ammattikorkeakoululle.

Avainsanat	sulautetut järjestelmät, konenäkö, sulautettujen järjestelmien suunnittelu, lämpökamera, älykkäät järjestelmät

Contents

List of Abbreviations

Appendix 1. Lepton 3 Power modes
Appendix 2. Automatic gain control in Lepton 3
Appendix 3. Color lookup tables

List of Abbreviations

CMOS	Complementary Metal-Oxide-Semiconductor is technology for constructing integrated circuits. Commonly used in image sensors.
CCD	Charged Coupled Device is higher quality technology for constructing image sensors.
FOV	Field of View is restriction what is visible for observer.
DOF	Depth of Field is effective focus range, distance between nearest and farthest objects in scene that appear sharp.
LWIR	Long Wave Infrared is infrared radiation in range of 8 – 15 µm.
VOx	Vanadium Oxide is material used as IR absorbing material in thermal cameras.
CCI	Command and Control Interface is Lepton 3 variant of I2C interface.
VoSPI	Video over SPI is name for Lepton 3 SPI video output interface.
NUC	Non Uniformity Correction removes FPN (fixed pattern noise).
FFC	Flat-Field Correction is technique used in imaging to improve quality.
SNR	Signal-to-Noise Ratio is ratio between usable values and noise in data.
SBNUC	Scene-Based Non-Uniformity Correction is variant of NUC.
FPN	Fixed Pattern Noise is repeating noise in otherwise uniform surface in image.
SPI	Serial Peripheral Interface is synchronous serial communication used in electronic devices.

I2C	Inter-Integrated-Circuit is serial communication bus for chip to chip communication inside circuit.
POE	Power Over Ethernet is system that passes electronic power over data cable.
AGC	Automatic Gain Control is system dynamically controlling gain to adjust output.

1 Introduction

The purpose of this thesis was to create a cost efficient and working prototype of a thermal camera. It needed to be a generic solution that can be used for multiple purposes. The customer wanted to know if thermal cameras can be used for monitoring large groups of people in closed areas. To reach the set goals the prototype was designed, built and field tested in a music-festival in August. The prototype was capable of capturing thermal image and sending data to remote storage.

This topic was chosen for strong personal interest and experience working with computer vision and embedded devices. This thesis helps anyone interested in embedded systems. The thesis explains the main concepts of thermal imaging, hardware used and explain how camera module was implemented. At the end there is suggestions for possible usages for this system and alternative ways of implementing thermal camera. Image processing techniques useful for customer purposes are explained and examples of processed image is shown.

The employer for this project was Metropolia UAS in collaboration with emergency services and security staff in Flow-festival. Work was done mainly independently from home. Field test was collaborated with another Metropolia student who was responsible for transferring the data to a receiving server.

2 Thermal camera theory

This section explains several important techniques, technologies and hardware used in this report. The focus is to provide the information needed in thesis. In addition it familiarizes a reader with topics used in the later parts.

2.1 About cameras

Photocells

To detect light and convert the energy from it to a digital value a device called photocell exists. It transforms photons energy to lower a sensor's internal resistance, and by measuring this resistance we get the digital value. If many photocells are put in rows and columns a sensor capable to capture images is created. [1.] The most common of them are called CMOS (complementary metal-oxide-semiconductor) and CCD (charge coupled device) sensors. [2.]

Thermographic imaging

All things radiate many kind of radiation. Extremely hot objects such as the Sun emit a radiation visible to human eyes. Formula 1 shows a simplified formula to calculate the wavelength where the radiation is strongest based on an object's temperature.

$$n = \frac{2898\ \mu m K}{T}$$

Formula 1. Wavelength (µm) corresponding to maximum intensity.

Formula 1 shows how to calculate infrared radiations peak point wavelength (n) based on the object's temperature (T). By using the formula we can calculate the infrared radiation peak humans emit using body temperature 36°C (~309 K) and get the result wavelength of ~9,4 µm.

The visible light and the infrared radiation differ only in wavelengths. Visible light's wavelength is between 390 nm and 690 nm. Wavelengths above that threshold are called a infrared radiation. The main difference between a thermal imaging and normal image capturing is the sensor used. One type of the sensor used in thermal cameras is called a microbolometer. Essentially its functionality is similar to the CMOS or CCD sensor, just in different wavelengths. Unlike many other thermal sensor the microbolometer doesn't need an active cooling system to function a long period of times. It is used in the camera module selected for this project. [3.]

Lenses

A lens affects how a real world is projected to an array of sensors in a regular photography as in a thermographic imaging. Relevant features of lenses for our purposes are a FOV (field of view), a DOF (Depth of field) and an optical distortion.

The FOV tells how wide an angle is in the view the lens projects to an imaging sensor. They may be a different values for a horizontal and a diagonal FOV. The DOF tells the optimal range the lens projects clearly, both the minimum and the maximum values. Optical distortion is usually presented as a percentage. [4.] Figure 1 demonstrates the optical distortion.

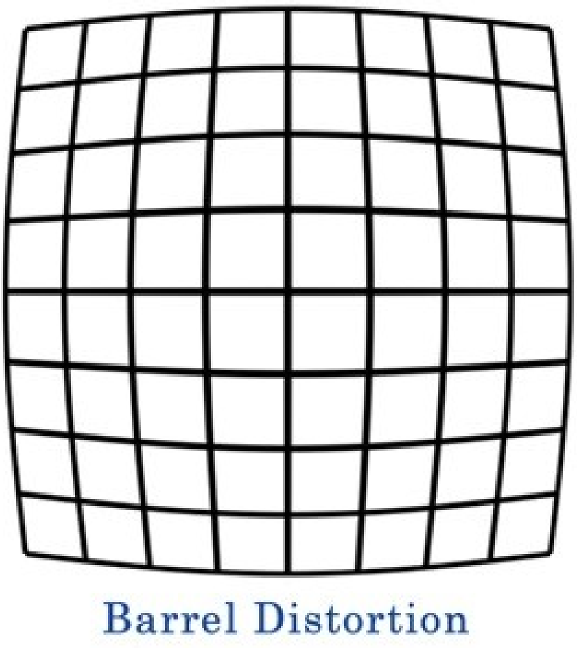


Figure 1. Demonstration of optical distortion. [6.]

Three kind of optical distortions exists. Barrel, illustrated above, pincushion and mustache distortion all occur in a photography. With the Lepton 3 modules lens barrel distortion is the strongest, although not clearly visible in used imaging ranges. [5.]

2.2 Lepton 3 LWIR camera module

Lepton 3 LWIR (long wave infrared) is ready to use factory calibrated camera module. It has been available to buy since march 2017. It is meant for applications such as a mobile phones, a building automation, a thermal vision and a night vision. Figure 2 shows the picture of the Lepton 3 module. [5.]

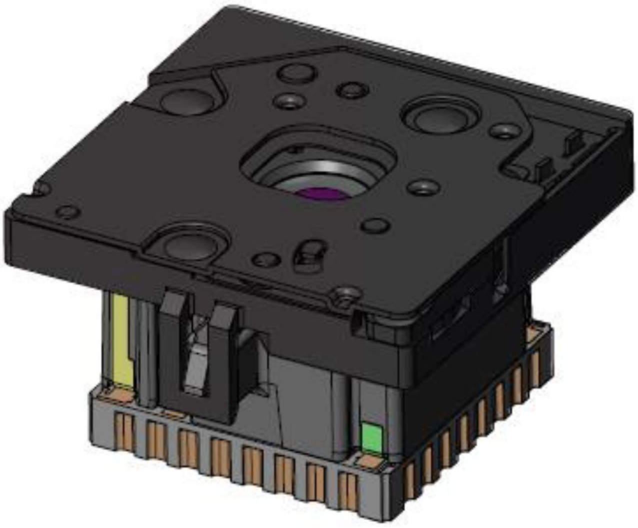


Figure 2. Lepton 3 camera without a socket. [6.]

As figure 2 illustrates the Lepton 3 camera module has a shutter mechanism integrated on top. The bottom part connectors attaches the module to a separate socket. [6.]

The main parts in the Lepton 3 module for our use case are a lens assembly, a VOx(vanadium oxide) microbolometer, a thermal image processing engine and communication interfaces. Figure 3 shows a simplified block diagram of the Lepton 3 module. [5.]

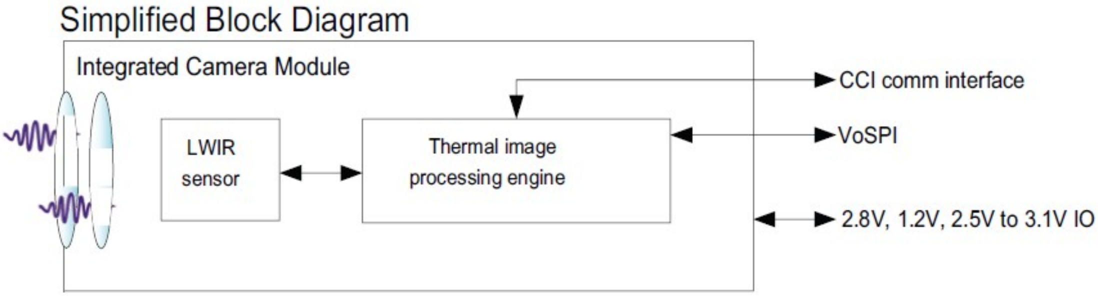


Figure 3. Simplified block diagram of Lepton 3 module. [6.]

As shown in figure 3, an infrared radiation is received through the lens in a LWIR sensor. The data flow goes through the Thermal image processing engine before it is transferred to a platform. Note that the CCI (command and control interface) interface is a control interface and the VoSPI (Video over SPI) is for transferring an image data to the platform. [6.]

The microbolometer functions as a LWIR sensor in Lepton 3. It has a resolution 160 x 120 pixels with a pixel size of 12 µm. Pixel size is the size of a one side of the approximately square shaped sensor, that measures the pixel value. A thermal sensitivity is less than 50 mK (0.05°C) which means it can detect changes in a temperature in that magnitude. It uses a progressive scan. [6.]

The progressive scan is a technique to store, transmit or display an image data. It stores an image one row at a time then moves to the next row until a full image is formed. This is also the way the Lepton 3 module sends the data. In comparison an interlaced scan only stores every other line, and after that stores the rest. [7.]

The Lepton 3 module's lens is made of silicon, has 71° FOV horizontally and 56° diagonally. The optical distortion is around 13% and the DOF is from 28 cm to infinity. The shutter is positioned in front of the lens. It protects the lens but in a power down state the shutter stays open rendering it not so useful in this project. [6.]

The image data is transferred with the VoSPI protocol maximum of 8.8 frames per second. One frame is transferred in 60 packets, 4 segments and totals to 38.4 kB. Each packet contains a packet number and the packet number 20 contains a segment number in its header. The frame rate is lowered by sending the same frame three times to make the camera module export compliant with the US export regulations for thermal cameras. The VoSPI protocol is explained later in the report. Figure 4 shows order of packets and segments in image data sent. [6.]

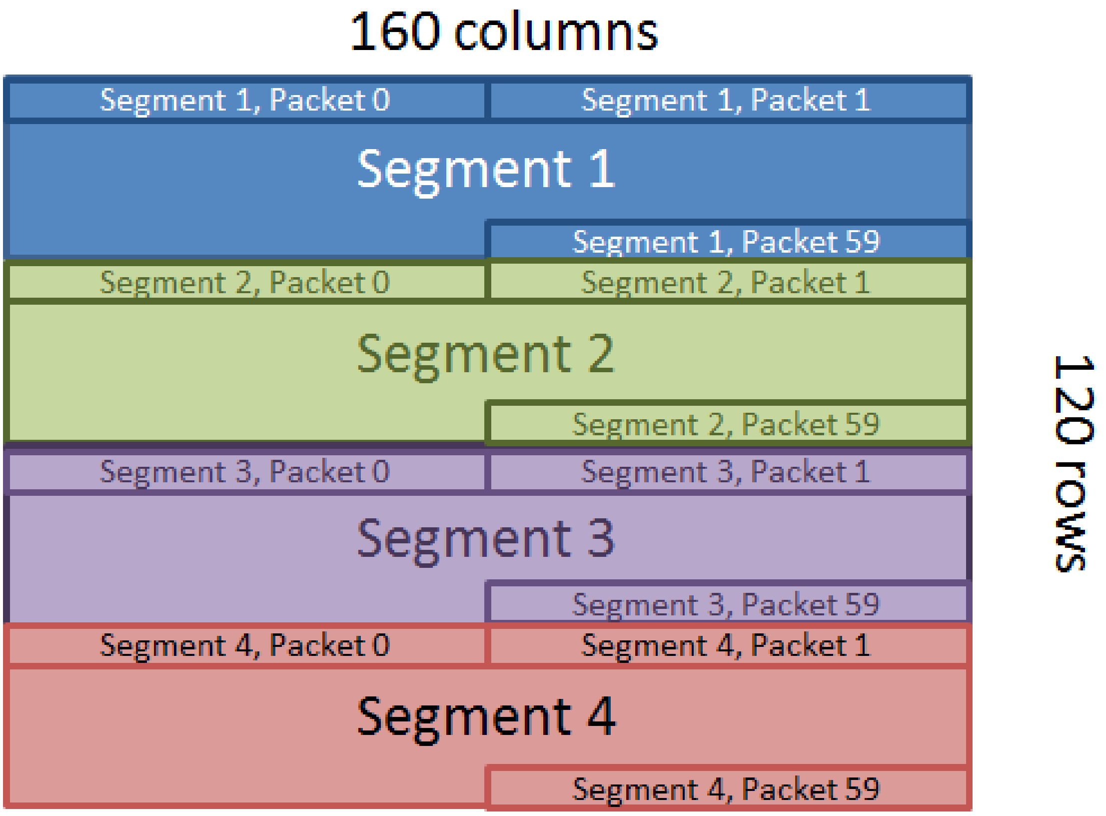


Figure 4. Frame packet order. [6.]

Figure 4 illustrates the order of segments and packets in a full image frame. The top left corner of the image has the first packet of the first segment, and the bottom right corner the fourth packet of the last segments. Inside every segment in the packet number 20, and only in that packet, the segment number where surrounding 59 packets belong to is readable. [6.]

Lepton 3 has a large number of controllable features and some monitoring information available. Controlling camera features is carried out with a CCI interface. The CCI interface is similar to a I2C protocol. Only difference is that Lepton 3's registers are 16 bit wide. Normal I2C protocol uses 8 bit registers. The I2C protocol is explained later in the report. Important controllable features of Lepton 3 for this project are following

- data transmission mode
- Image pipeline options

- internal temperature
- power status
- reboot routine.

Lepton 3 has three different data transmission modes. The first one is to send a raw 14 bit data, the second sends a scaled down 8 bit data and the third is to send a 24 bit color data that is made from the 8 bit data. The first mode was used since it offers most accurate precision. [6.]

The second options controls a image pipeline that performs set of operations to the data. These options and operations are explained later in the report. An internal temperature can be read from Lepton 3's register. It can be used in applications that analyze the data afterwards. A power status tells in which power mode Lepton 3 is currently in[6.]. More on Lepton 3 power modes modes in Appendix 1.

A reboot routine is used to reset the camera module in case of an unsolvable failure. In this project it is used as the last option to restore a synchronization in the VoSPI and during an initialization. [6.]

Sending commands, writing or reading data from Lepton 3 uses four registers. A status register, a command register, a data length register and 16 data registers. Before an action, status register is polled to know if Lepton 3 is ready to receive orders. After the command is written in the command register, the status register can be read to verify when the action is completed. [6]

To read a data from Lepton 3, a length of requested data needs to be written in the data length register. Lepton 3 starts the action after the command is written in the command register. The command specifies a data requested. After the action is completed a data can be read from data registers. To write a data it needs to be written to data registers and the length of data to be written in the length register before giving commands. Figure 5 demonstrates writing a data to Lepton 3. Reading a data and giving plain commands is similar to writing with exceptions mentioned previously. [6.]

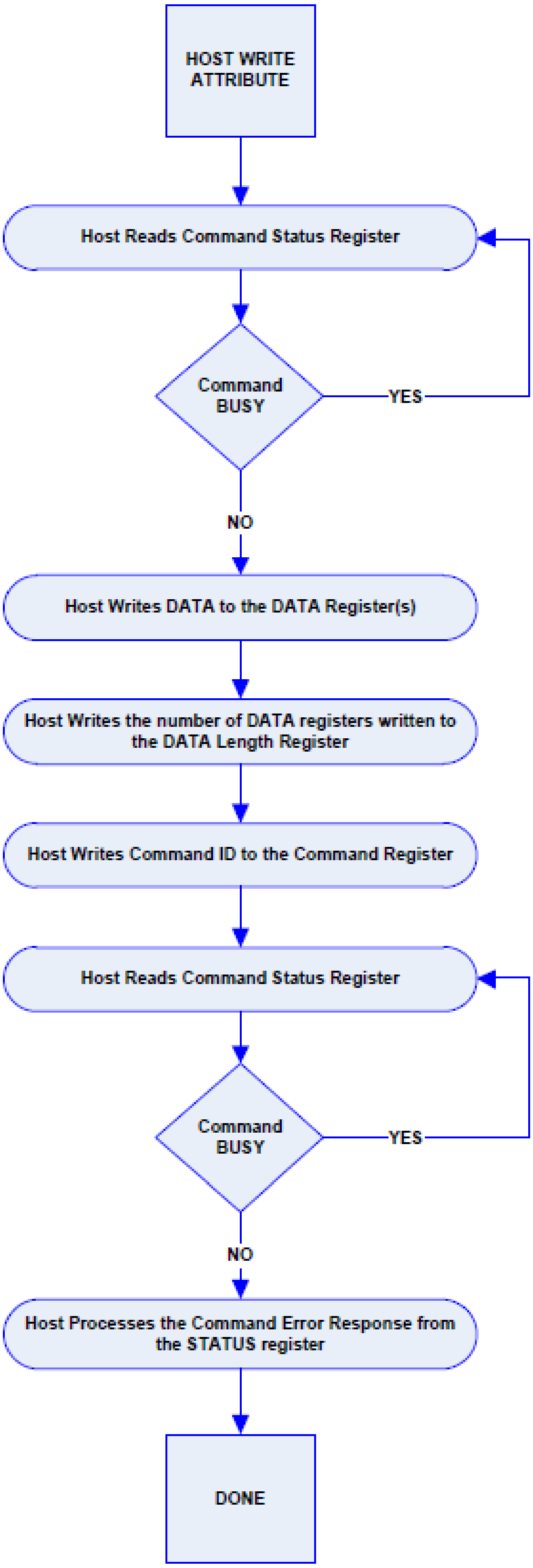


Figure 5. Procedure to write data to Lepton 3. [6.]

Figure 5 illustrates how the platform host communicates with the Lepton 3 camera module. The process begins with host polling the status register. When a status is clear the host writes required data to the camera module and starts polling the status register again. When the register is cleared again the camera module has completed the action. [6.]

Image pipeline

Before Lepton 3 sends a n image data it will perform a set of operations to it. Figure 6 gives an overview and an order of these operations.

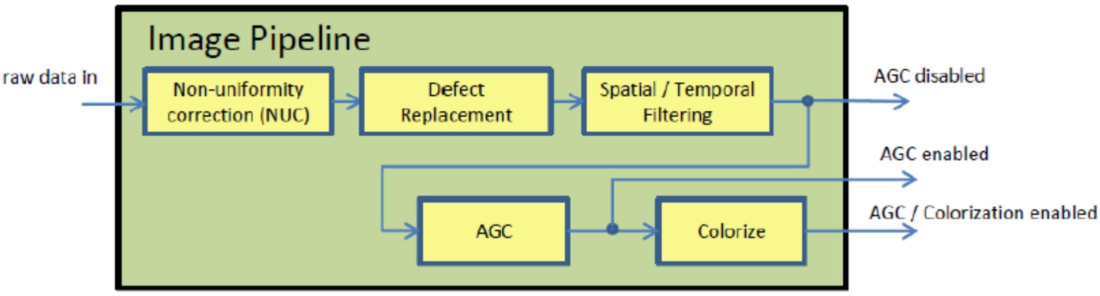


Figure 6. Overview and order of Lepton 3 imaging operations. [6.]

Figure 6 shows a thermal image processing engine flow. A raw data arrives from a imaging sensor. The data goes through a non-uniformity correction, a defect replacement and a spatial / temporal filtering before it can be transferred forward. If an AGC functionality is enabled the data is handled by the AGC before it is outputted. If a colorization is enabled the data goes through the colorization before it is transferred forward. [6.]

The first operation Lepton 3 performs is the NUC (Non uniformity correction). The Lepton 3 datasheet describes NUC with following. “NUC applies correction terms to ensure that the camera produces a uniform output for each pixel when imaging a uniform thermal scene. Factory-calibrated terms are applied to compensate for temperature effects, pixel response variations, and lens-illumination roll-off”. The NUC can also perform an offset transformation of values or a FFC(flat-field-correction), but they are less effective in stationary use cases and are not used in this project. [6.]

The defect replacement is an operation that generates new values for a factory calibration or runtime identified defected pixels. New values are generated using values from adjoined pixels. [6.]

The spatial filtering is described in the Lepton 3 datasheet with following. “The image pipeline includes a number of sophisticated image filters designed to enhance signal-to-noise ratio (SNR) by eliminating temporal noise and residual non-uniformity. The filtering suite includes a scene-based non-uniformity correction (SBNUC) algorithm which relies on motion within the scene to isolate fixed pattern noise (FPN) from image content”. A more detailed information about these operations is not available for the public. [6.]

The AGC (Automatic Gain Correction) transforms a 14 bit image to 8 bit using Lepton 3's own variation of a histogram equalization. While transforming the image visible to human eye it loses 98% of data accuracy due to downgrading values. Since one of the goals was to create generic cameras to be used in all kind of applications, the AGC filter is left a disabled state. [6.] More information about the Lepton 3's AGC in Appendix 2.

The colorization transforms a 8-bit image to a 24-bit color image. No data is added or removed in this stage. It is only to make the image even more visible for human eye. The transformation is based on a LUT (look up table) values. An user can choose from preset look up tables which one to use. [6.] The options are listed in Appendix 3.

SPI- bus

SPI- (serial peripheral interface) bus was created by Motorola in late 1980 and is free to use. It is a synchronous serial communication protocol over short distances used in embedded devices. Today it is found in many embedded components. The SPI- bus uses four wires

- MOSI (Master out Slave in),
- MISO (Master in Slave out),
- clock line
- chip select.

A communication is master driven. The master device generates a clock pulse and if multiple devices are connected to the SPI- bus the master can choose which device to communicate with using a chip select line. Each slave device has it's own chip select line with the master and rest of the lines are shared. [8.]

SPI is a full duplex protocol in which a master and a slave transmit and receive data in each clock cycle. Comparison to a half duplex where both parties can transmit and receive data in turns, not simultaneously. [8.]

Four modes the SPI can use exists. The modes differ in what point of clock cycle a data line is read, either rising- or falling edge. This is called a phase. The second difference, a polarity, is whether a signal is active low or active high. Which mode to choose depends on the slave settings and usually is factory set and cannot be changed. Figure 7 clarifies four different SPI modes. The phase is marked with CKPH and the polarity with CKPL.

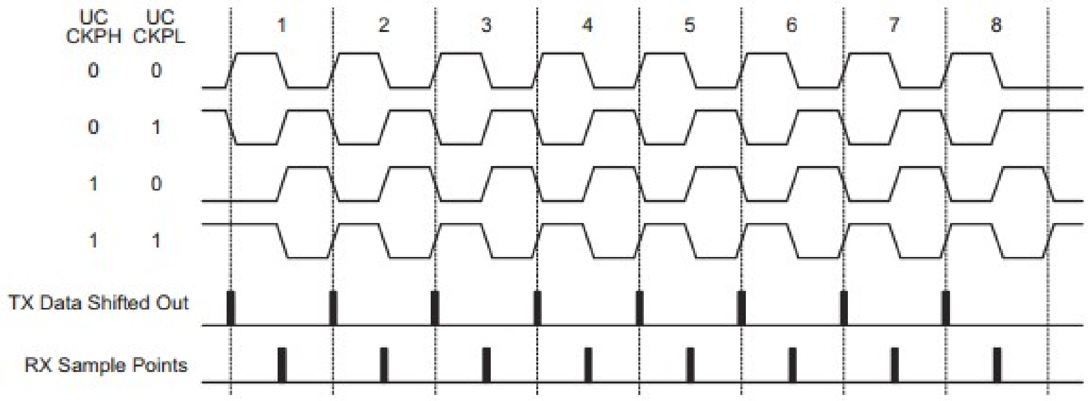


Figure 7. Configuration of the polarity and phase settings.

Figure 7 shows four different ways to handle SPI transactions. Vertical lines show the moment when the data and clock lines change state by a sending device. From the bottom of the figure RX Sample points can be seen as the moments where a receiving device reads the data line.

Lepton 3 module uses a SPI- bus called a VoSPI to transmit image data. It does not utilize the MOSI line, only the single MISO data line. Only Lepton 3 as slave is sending the data and only the master is receiving it. Lepton 3 can operate in 20 MHz SPI clock frequency. Theoretical 8.8 MHz frequency is needed to read all incoming data. Failure to read all incoming image data in time causes the VoSPI going to an out of sync mode. [6.]

I2C protocol

I2C or I^{2}C protocol is a widely used communication protocol in embedded devices or an inter-chip communication. It was created by Philips Semiconductor and has been free to use since 2006. It uses two bidirectional wires, a clock- and data line. Both lines are pulled up with resistors. The I2C bus can have multiple devices same time attached. An address system is used to select the target for a transaction. [9.]

The transaction structure is predefined and differs only if a 10 bit address is used instead of a 7 bit address. A master device initiates the communication by pulling the data line low. The data line is read during a rising edge of the clock line. First the address is sent followed by a R/W (read or write) bit which tells the slave if the data is being send or received. After the address is sent, the data is sent in 8 bit sequences. After the address and every byte of data is transferred an acknowledge bit is send by a receiving device by pulling the data line low. If the R/W bit was set to read, the slave responds with the requested data with similar 8 bit sequences. After the communication is completed the data line is pulled high for stop condition. [9.] Figure 8 demonstrates the I2C transmission protocol.

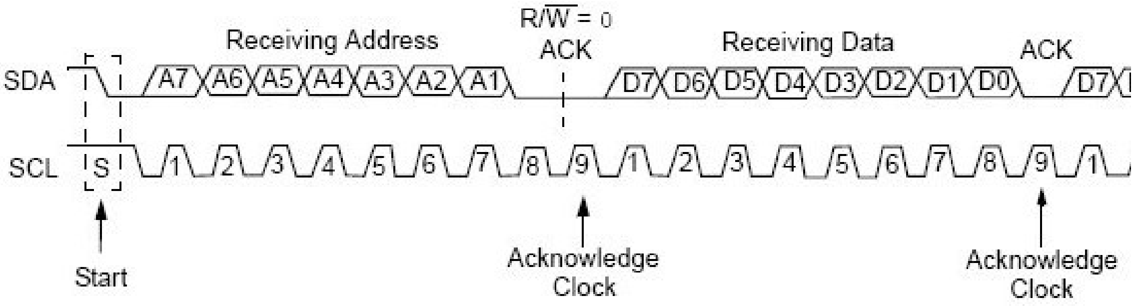


Figure 8. I2C transmission protocol. [9.]

Figure 8 illustrates the i2C transmission protocol with the 7 bit address and a first byte of the data. The sequence can continue with more bytes of data ending with the stop bit.

Udp-protocol

Two widely used network protocols are a TCP (Transmission Control Protocol) and a UDP (User Datagram Protocol). The TCP is reliable, ordered and an error-checked way to transfer data objects, but it increases a latency and amount of data transferred. The UDP protocol is a lighter protocol. A sender defines an ip address and a port and a packet either finds the destination or not. The UDP- protocol is suitable for purposes where an error checking and a correction are either not necessary or are performed in an application. It is preferred option in real-time systems where dropping packets is preferred to waiting for retransmission. [10.]

3 Hardware

This section explains a design and implementation of a hardware part of the thesis. The focus is on a design point of view and implementation methods. In addition it provides more detailed information on the hardware used.

3.1 Goals

When designing an embedded device one of the first things to do is make the goals clear. In case of this project, the goals were partly set by a school and the festival organizers. They set that the system should be able to record a long wave infrared data, also known as a thermographic image data, and send it to a remote location and save it for a later analysis. The system should also be a portable and generic solution for later usages. The goals for the cameras were formed based on those definitions. At the end all the goals were reached but one. The goals were:

- The camera must be able to record and send thermal data.
- The camera must be weatherproof.
- The camera should be able to save data locally.
- The camera should be portable and easy to install.
- The camera should consume little energy for a possible battery usage.

3.2 Design

Camera module

To reach set goals the first thing selected was which camera module to use. Not too many options for buying a thermal camera module exists since it is a fairly new technology and quite expensive. Before selecting the camera module, a few specifications were needed to know. A resolution, a field of view and how high the camera needs to placed. A quick simulation with a Blender 3d modeling software was used to simulate how different angles and resolutions should look in different heights. In Blender it is easy to set an environment and a camera view and alter the parameters to test different settings. Figure 9 shows the simulation view.

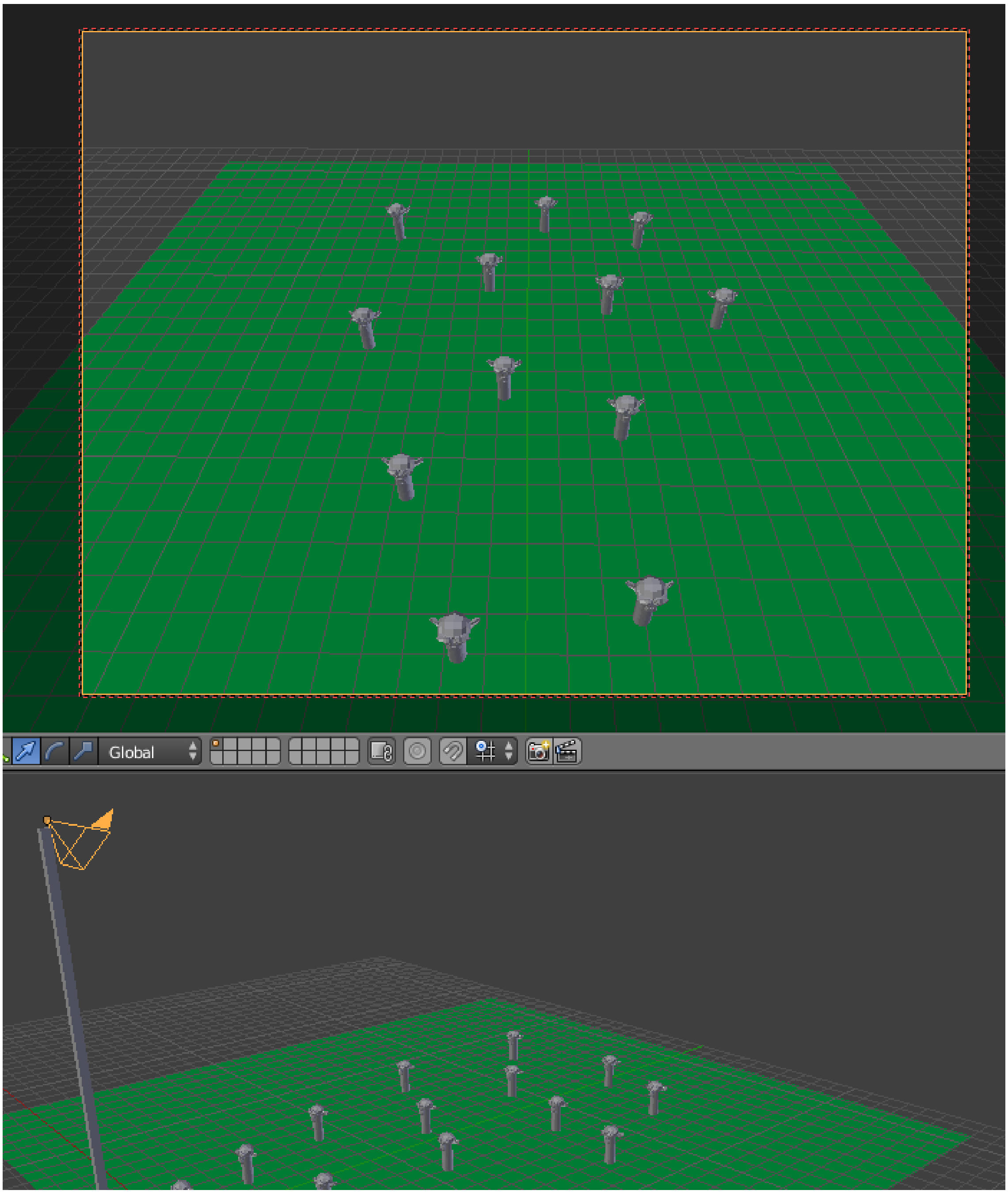


Figure 9. Blender simulation to test different camera parameters

In figure 9 every square is one square meter and the camera is placed in 15 meter height. The lens's angle is 71 degrees horizontally and the aspect ratio is 4:3. Both values are the same as in the camera module selected. Note that monkey headed figures presenting humans are slightly oversized to improve perception.

The simulation proved that suitable specifications for the camera would be combination where a one pixel shows an area of 100 – 400 cm^2 and the camera would be placed at least in 8 meter height above the surface. The best result is achieved in 15-25 meter height, higher than 35 meter would be too high.

The camera module also had to have a suitable DOF and an optical distortion. For the DOF a suitable range was smaller than 1 meter for a minimum value and a maximum value larger than 200 meter. Usually in camera modules the maximum value for the DOF is infinity and a minimum value is usually marked below 10 cm, when a camera module is not meant to capture only close up images. The optical distortion of lenses was not a problem since in all camera module options an optical distortion was in an acceptable range. When capturing images over long distances an optical distortion can be corrected in an application that processes an image data.

Lepton 3 camera module with a resolution of 160x120 pixels and a lens's angle 71°x55° was selected. [6.] Optimal operation height with the module was calculated to be from 10 m to 30 m.

Platform

For selecting a suitable platform to meet the goals and camera module specifications a Esp-8266 12E microcontroller was chosen. It is cheap, has a build-in WiFi, enough digital io ports and a fast enough SPI- bus frequency. It is also fast enough to handle reading a data from the camera and sending it forward. [11.] [12.]

Esp-8266 12E is a low cost microcontroller chip with a build-in WiFi router chip and an antenna. It has a 160 MHz clock running on 160 MHz. A Esp-8266 chip can be set to run only on 80 MHz speed if needed for example to save power. In this project it was left at a full speed since the full speed was needed. A program code is stored in a 4 MB Quad instruction SPI flash memory [13.] that runs on 80 MHz, and can also be set to run half speed. For performance reasons it was left running at full speed. The chip also contains some resistors and capacitors required to use with the Esp-8266 microcontroller. [12.] Figure 10 displays a full schematics of the factory made Esp-8266 12E chip.

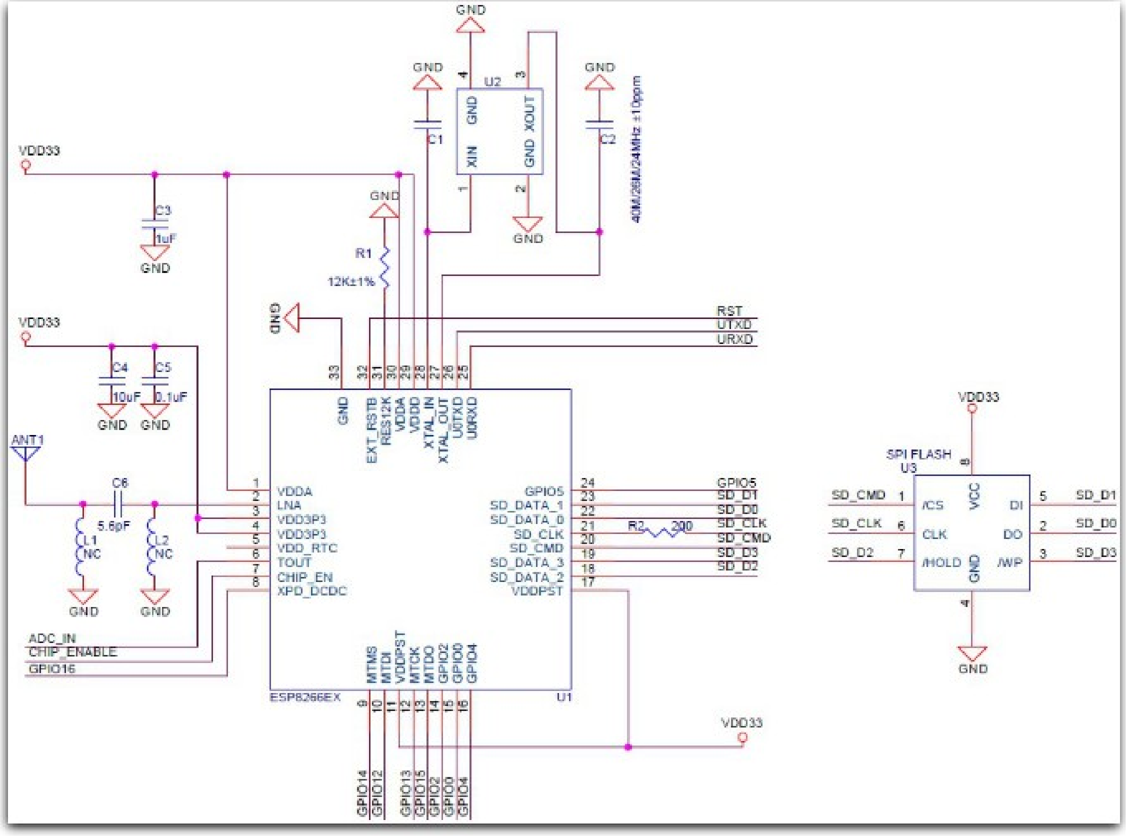


Figure 10. Schematics of Esp-8266 12E. [11.]

Figure 10 schematics shows how Esp-8266 12E is made. The Esp-8266 chip with all ports is shown in the middle. An antenna connection and a power connection are on the left side of the chip. An oscillator chip that gives a clock signal to Esp-8266 is on the top. Separate part on the right is a SPI flash chip that stores a program code. [11.]

Esp-8266 has three boot modes. Booting from a SPI flash, booting from a SD-card or download a new code from a UART. The SPI flash boot and the download new code modes were utilized in this project. To select which mode to run Esp-8266 reads three io pins during powering up. By pulling these pins high or low a developer can choose a boot mode. To switch between the two used modes only one of these pins had to be adjustable. [14.]

Storing the data offline was meant to be implemented by writing a raw data directly to the SD- card using the SPI interface. Unfortunately it became clear during a testing phase with the hardware that the camera would not be capable to save the data in the SD-card during runtime. It could have been possible to get an external flash memory block or a card reader unit to handle the data storing offline. Due to a tight schedule around the project there was no time left to explore these options and the feature was left out. [15.] [16.]

Circuit

Circuit was designed around the camera module and the microcontroller. It was designed to ease a developing and testing the camera.

The circuit handled regulating an input voltage from 5 V to 3,3 V. This was achieved with a LM1117-T voltage regulator and two 10 µF capacitor. Capacitors were added to reduce noise. The circuit has one 470 µF capacitor to prevent rebooting the module during power spikes. It has a power led for emptying capacitors after a power supply is unplugged, some resistors acting as pull-up or pull-down resistors, a button to reset the board and a button to enable easier switching between boot modes. Figure 11 shows the full circuit with parts mentioned above.

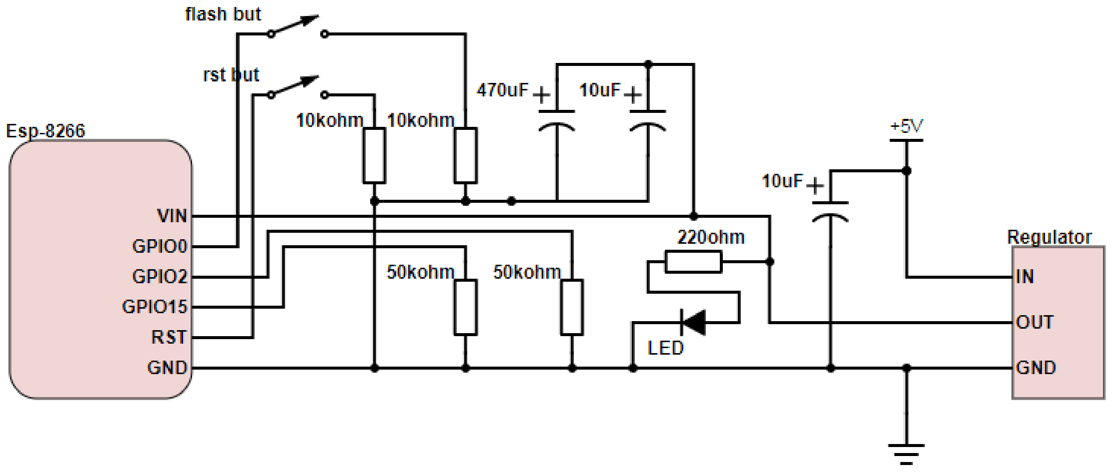


Figure 11. Full circuit of camera

Figure 11 illustrates how components are connected. Capacitor and resistor values were chosen by experience and they proved working after testing the full circuit first time. A power connector and extensions for programming pins were also soldered in the circuit.

3.3 Building phase

Initially one camera module was built to verify it was correctly planned and in case of later changes they would be easier to add.

At first the Esp-chip was attached to a circuit board with solder along with a camera module socket. Actual camera module was disconnected during an assembly to avoid mishaps. Big elements such as a regulator and a capacitor were next followed by smaller components. Figure 12 illustrates these connections.

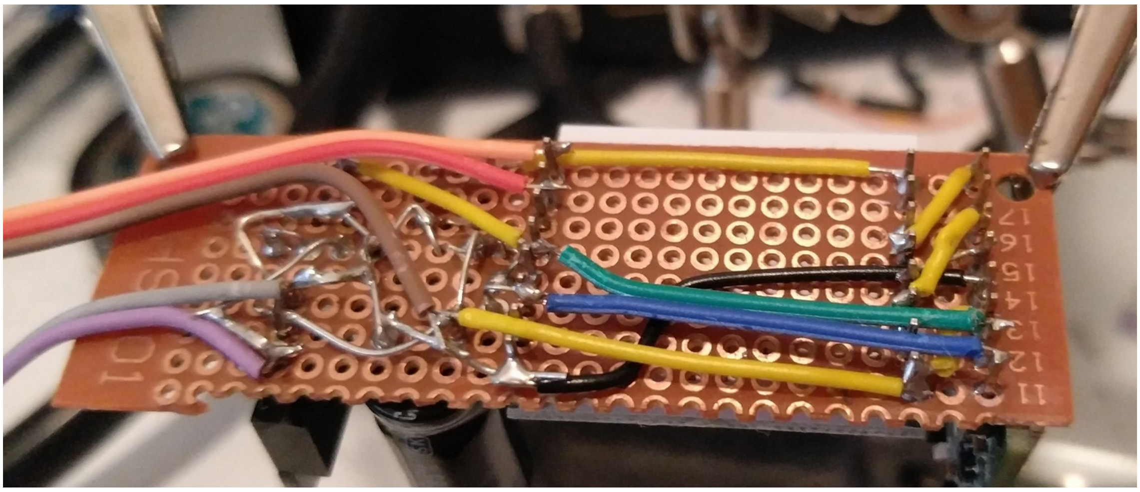


Figure 12. Finished backside of the camera board

Figure 12 shows how connections were made in the camera circuit. Note that component legs were bent to form connections along with single core copper wires. The image is from a finished circuit.

About two weeks before a field test when the hardware had proved to be functioning properly and the system was ready to be tested with more than one camera rest of the cameras were assembled by hand. They were built with a conveyor belt style in small phases and all cameras simultaneously. First big components were attached continuing to smaller ones till all parts were attached to the board, then connections between components were made. This was to ensure the same outcome for all boards and minimize a change for an error in a construction. Figure 13 shows finished electronics for the camera.

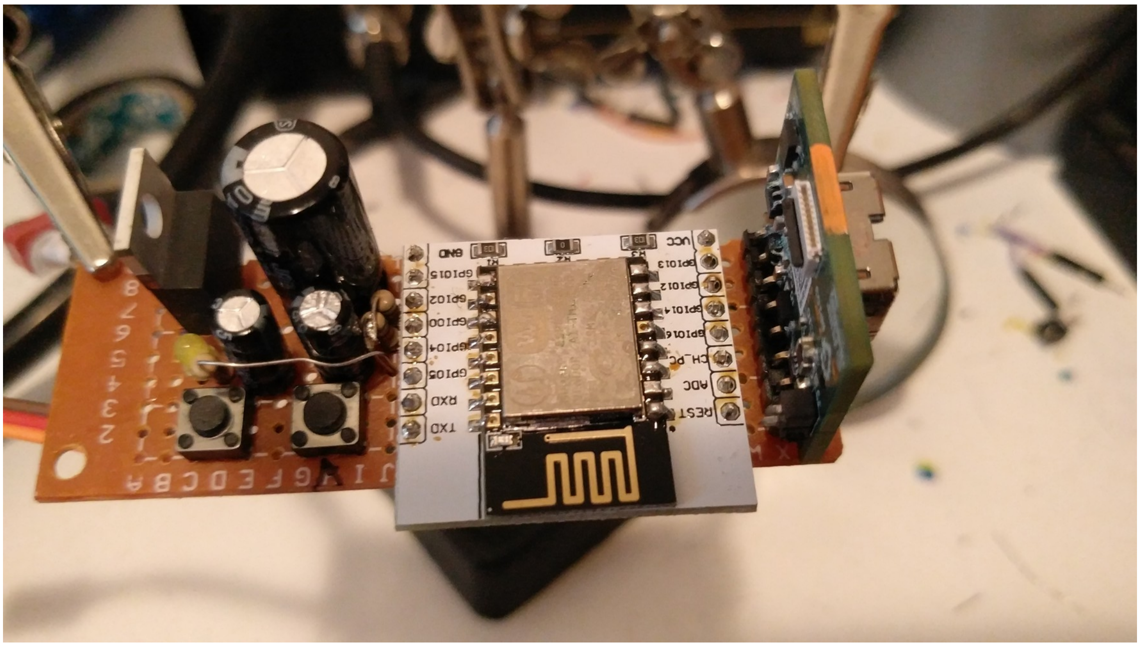


Figure 13. Image of finished camera electronics

In figure 13 the camera module socket is on the right, in the center Esp-8266 12E on a soldering platform and the left side there is the external circuit shown in figure 11. Note two buttons that make uploading a new program on Esp easier.

Casing

Finished electronics needed a casing around them to protect from a weather and a physical damage. The casing needed to be an easy to install on many kind of surfaces and the cameras had to be able to be aimed to a monitored direction.

A camera leg was made from aluminum bars to keep it light weight. Two simple joints were made from bolts and nuts. With a wrench the angle of these joints can be adjusted during an installation. Figure 14 shows finished joints.

Figure 14. Close up image of camera leg.

Figure 14 illustrates two joint structures. The joints have 2 nuts and a bolt to allow adjusting and locking the camera in a position.

A hull for the casing is made from a Raspberry pi casing. At the time they were easily available and they are cost effective. The weather protection has two layers. An inner layer is made from contact paper to keep water away. On top of the contact paper duct tape was used to keep all parts together. Few holes are drilled on the leg to allow zip ties be used when attaching the cameras. A small hole for the camera lens to capture an infrared radiation is in front of the hull. Finished camera weighs about 200 grams. A dimensions for the taped body are 96 mm, 65 mm and 43 mm. The leg is 130 mm long. Figure 15 shows the finished camera body.

Figure 15. Finished casing weather protected

Figure 15 shows weather protected camera casing. Note power cable coming from left side and holes for zip ties. Power cable and camera lens are only parts without weather protection layer.

4 Software

This section explains a design and an implementation of a software part related to the thesis. The focus is on explaining a design process. In addition it provides more information to reproduce the software for the camera.

4.1 Design

Designing a software is an important part that many students ignore when desire to create a code is overwhelming. If a software to create is large or otherwise complex there should be some kind of architectural decisions made before actual coding can start. This way a code structure will be easier to follow and modify later and is usually more efficient. A decent architecture also eases a debugging phase.

To get a robust and lightweight camera software several techniques were tested before actually designing the program structure. These included measures of an actual SPI-handling speed of the system, a fault recovery of the camera module, a time required to record the data to the SD-card and a time required to handle a UDP transaction. Timings were took by performing a specific task multiple times as fast as possible measuring the time used with a Esp-8266 chips internal clock.

SPI handling test

The SPI maximum frequency between the camera module and Esp-8266 was tested by reading the VoSPI stream from the camera module. According to the datasheet 20 MHz is the maximum speed, but the VoSPI could be read with 40 MHz without errors. At first a generic library was tested to read the VoSPI. It proved to be too slow option. A faster speed was achieved using directly Esp-8266 registers that control the SPI- bus. The Esp-8266 register handling and transferring the data to a RAM during reading each packet added ~30% more time to an actual data transfer speed with a 20 MHz SPI- bus frequency.

VoSPI fault recovery test

Fault recovery capabilities were tested by reading the SPI- bus too slow causing the VoSPI going an out of sync state and testing different ways to restore a synchronization. The most reliable method to restore the synchronization is by hard resetting the camera module by disconnecting a power line. The next best option is to perform a software reboot. The quickest way is following Lepton 3 datasheet instructions for restoring the synchronization, but in some cases they did not always restore the synchronization.

Writing data to SD–card

An ability to save the data directly on the camera was meant to be carried out using a SD-card. With the current configuration free pins to use with the SD-card limited writing to be implemented with the SPI- protocol. This could have caused complications since the VoSPI interface with the camera module uses the SPI- bus as well, but there would have been solutions to use both simultaneously, since the camera module does not receive any data, and a response data from the SD-card can be ignored. However by testing the SD-card write speed with the SPI- bus it proved to take at least twice the time needed to read three packet from the camera module than writing them to the SD-card. At this point the offline storing option was dropped.

UDP sending speed

To test an UDP sending speed a local network was used and a Raspberry pi as receiving end. Esp-8266 was put to send data as fast as possible with different sized packets. Sizes used were 164 bytes, 484 bytes and 3204 bytes. The fastest speed was achieved with the biggest packet size, the second largest packet having a transmission speed slightly slower. However a reliability with bigger packets tends to be lower in a public network and it is better to queue more smaller packets in a sending device to ensure fewer data losses in case of WiFi delays. This test was only to compare an error rate and a transmission speed with different packets sizes. It couldn't tell how fast the transmission will be in a real environment. This was left to be tested in the field test.

Decisions

Based on previously mentioned tests the program architecture was divided to three parts to ease testing and debugging. An initialization tasks, handling the VoSPI- bus and handling data sending with WiFi. Program parts were built and tested separately. After merging parts they were tested again together. All code was written in C and compiled with an arduino compiler. A debugging was carried out with serial messages to a computer terminal.

4.2 Implementation

Initialization

An Initialization tasks are explained below in order they are executed each one time after the camera boots up. A soft reset on Esp-8266 also triggers the initialization sequence.

At first two FIFO (First In First Out) buffers are allocated in a RAM. A buffer to store packets from the VoSPI and a buffer to store packets that can't fit in an UDP transmission queue. Timing variables, packet counters and phasing variables for reading the SPI- bus in phases are allocated and set to initial values.

Connecting a WiFi module to an access point is carried out by first disconnecting possibly existing previous connections, then setting the WiFi module mode to a stationary mode and giving a connection command. These are library calls to an arduino WiFi library. If the connection is not made within 30 seconds the soft reboot is performed on Esp-8266.

UDP packet buffering is initialized with a library call giving a target IP and a port to send packets to. A sign up message is sent to a server to notify the camera being online. No receiving details is configured since the camera does not receive any data from the server.

A SPI- bus setup that includes setting SPI control register values to control a bus frequency and an used mode. Different SPI modes are explained in the theory part. The SPI mode Lepton 3 uses, mode 3, is set to a register. A speed of the bus is controlled by setting clock divider. Esp-8266 runs with a 160 MHz clock so proper divider is 4 to achieve the fastest reading speed 40 MHz.

I2C setup consisted setting an used frequency and pins on the board with arduino library calls. A connection to the camera module is then verified by reading a temperature data from the camera module. If the connection is not correctly formed the soft reset is performed on Esp-8266.

Rebooting the camera module is carried out with a reboot command send to the camera module. Rebooting should not take more than 6 seconds. After 7 seconds a temperature from the camera module is read again to verify a successful restart.

The last part in the initialization process is a synchronization for the VoSPI. It is carried out by pulling a chip select pin to a high state, waiting 300 ms and pulling the chip select pin back low state as the Lepton 3 datasheet instructed. During the synchronization software buffers for reading image data are also reseted.

Loop structure

Continuously running loop consist of reading one image packet in phases, buffering the previously read packet during actual VoSPI reading and checking whether the packet is to be discarded or sent next. Every time possible the loop yields to give an execution time to an internal uncontrollable process that handles a communication to the router part of the esp-8266 chip.

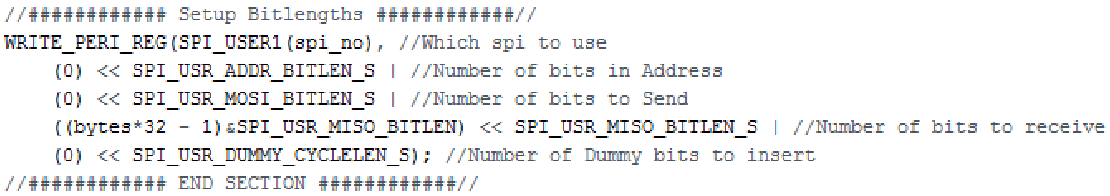
During the loop multiple numbers of packets is monitored. Keeping a count on a packet number read from Lepton 3 verifies that the VoSPI stays synchronized. If the synchronization is lost a re-synchronization is tried same way as in the the last part of initialization phase. A number of failed UDP packets is monitored to detect a too low success rate. A time between each successfully read frame is also monitored. If too many sent UDP packets fails, the re-synchronization fails or the time between frames is too long the soft reboot is performed.

Reading VoSPI

Using the SPI- bus in a microcontroller is usually carried out with library calls to hide details and a complexity of protocol. In some cases, like in this project, library calls don't provide a required speed and a RAM usage needed. Alternative options depend on a chip used.

Esp-8266 provides SPI- bus handling with registers. One register controls a length of transaction, one command and execution and maximum of 16 32-bit registers can be used to writing or reading data. A lower level data transaction is handled by the chip itself. By polling the command register a program can know when a transaction is completed. To read a complete image packet reading is implemented in 3 parts. Register values are copied to a temporary buffer after each part is done. Reading the VoSPI is carried out in phases to minimize idle time and ensuring the best possible WiFi connection.

Figure 16 shows macro used to setting and getting SPI control register values. All but “bytes” parameter in the following macro are predefined compiler set values to increase a speed during runtime.



```
//############# Setup Bitlengths ############//
WRITE_PERI_REG(SPI_USER1(spi_no), //Which spi to use
    (0) << SPI_USR_ADDR_BITLEN_S | //Number of bits in Address
    (0) << SPI_USR_MOSI_BITLEN_S | //Number of bits to Send
    ((bytes*32 - 1)&SPI_USR_MISO_BITLEN) << SPI_USR_MISO_BITLEN_S | //Number of bits to receive
    (0) << SPI_USR_DUMMY_CYCLELEN_S); //Number of Dummy bits to insert
//############# END SECTION ############//
```

Figure 16. Example macro used to read or write data to registers

Figure 16 illustrates macros used to control Esp-8266 registers. A compiler optimizes this macro to contain just variables spi_no, bytes, multiplier and a possible constant multiplier and addition. A resulting value is written to the specified register at runtime.

Sending UDP-packets

Actual WiFi transaction part inside Esp-8266 12E is hidden to developers. It is handled by its own module that reads a buffer from a RAM to send the data. The data is added to that library controlled buffer by using library calls. Size of that buffer is an unpublished information but the behavior of the chip indicates it to be around 8000 bytes. UDP packets are sent to a developer defined IP-address and a port with a library call.

The image packet read in the previous program loop part is verified to be a valid image packet it is selected to be sent forward and copied to a small buffer. When three packets are selected, packet headers are removed and a new header is added before payloads are combined. This 484 bytes long UDP packet is then copied to an UDP library buffer. If that buffer is full the packet is queued and tried to add to the library buffer in a next loop cycle. This queue is as long as there is a safely usable free RAM left in the chip, which equals to the size of 40 of these packets. If this overflow queue fills up and too many packets are discarded the soft reset is performed. A data integrity check is not needed when streaming UDP packets.

The header attached to each payload contains an information such as a packet type, a camera id number, a packet id number and a part number. The camera id is unique for each camera, the packet number is an increasing number that is used to identify a sending order and the part number is to identify where these 3 packets belong in a frame. The packet type is to separate a sign in message and an image data. The new header is 32 bits long.

The root cause why there is a big failure percentage in sending packets with WiFi is when an access point cannot respond to the camera when needed. The camera can only buffer 60 packets total to be sent. If the camera has to wait longer than filling up buffers takes, which is the minimum of 27 ms, following packets are discarded. Higher quality WiFi routers respond faster to WiFi queries and that significantly reduces wait periods.

5 Field test

This section explains the course of field test. Section explains the results and provide proof that thermal cameras can be used to monitor festival audience.

5.1 Preparations and setting

Thermal cameras were tested in a Suvilahti festival area during Flow festival. They were originally meant to be tested twice in Suvilahti, during Tuska- and Flow festival, but the organizers of Tuska could not handle any more moving parts. The Flow festival was arranged between Friday 11.8.2017 and Sunday 13.8.2017. That weekend had poor weather conditions.

Preparations for the field test started about the same time as designing the camera. Initially there was no knowledge of budget or place for the cameras. After a price for the module was estimated the organizers decided to reserve money to build 5 cameras. At this time the plan was to use at least four cameras in two locations. During the building phase the number of cameras to be installed was reduced to three when the offline storing was left out, but a quick alternative plan to set up a fourth camera was made by the request of the Flow festival organizers.

Multiple meetings took place over the course of planning the field test. Participants for two bigger meetings included flow organizers, school office representatives, teachers and students. In these meetings the project scope was defined and budget set. The camera installation plan and areas to monitor with the cameras were set and timetables agreed. Some smaller meetings including just teachers and students to monitor project progress and handle practical arrangements were arranged.

Camera positions were planned with the assistance of Metropolia teachers and Flow festival organizers. The only requirement for the cameras were to get power and be located high enough, more than 10 meter and have visibility to the area where people move. A couple of possible positions were planned before the installation from 3D model of the area and visiting Suvilahti one time before construction of the area and once after it had started. The possible positions were a big, old, round building that had lights on it, a very tall stationary light tower, an audio editing booth with a custom pole increasing its height and a light tower. The final installation spot was verified 30 minutes before the installation and it was around 20 meter high light tower.

The cameras were built with a 'plug and play' mindset. No configurations had to be made during the installation. Only a power cable was needed to be connected and the cameras start searching for an access point. Three cameras were attached on top of the light tower with cable ties and duct tape. To make sure they won't fall down during the predicted storm they were secured with metal cables. Power to the cameras was provided by the festival organizers.

The fourth camera was installed on top of a light post near the main gate to monitor people flowing through a street. The camera had its own access point and a server receiver in a nearby tent, since it was overly complicated to connect that camera to the same network than other cameras with resources available. The whole secondary setup consisted of the camera, a basic home WiFi router and a Raspberry pi.

5.2 During test

Initially the access point was set on a ground below the light tower. This proved quickly to be a bad choice when a WiFi signal was next to nothing. During the first day no useful data went through. This mistake was corrected on Saturday morning by attaching same router closer to the cameras on top of the tower. After the improvement an average signal was above 70%. The signal rate dropped during an evening when people filled the festival area.

Weather conditions were not optimal during the weekend. During Saturday night around ten a power went down in Leppävaara and the receiving server was shut down. During the Saturday night after the power had been restored the server was restarted. Tests were made to ensure the cameras were unharmed by the storm and testing could have continued without problems. After the damage control at the morning, our network cable on Suvilahti was cut and the cameras could not send any data.

There was an emergency plan already made in case the network fails to deliver the image data. We set a battery powered WiFi 4G module bottom of the tower in attempt to get some data delivered during the Sunday evening. Unfortunately a WiFi network in the festival area was too crowded for the less powerful 4G module, and practically no data went through.

5.3 After test

Monday morning the cameras were retrieved. Casings were opened and expensive camera modules were removed from circuits and cleaned. No water or any kind of damage was found inside the camera hull. The data stored in the Raspberry pi server was retrieved and discovered useless. A total amount data gathered for the weekend was checked and an useful material found totaled around 8 hours from 2 cameras. A third camera on the light tower had too poor signal for the data to be used.

5.4 Results

The festival organizers wanted to know if thermal cameras can be used to monitor festival audience. Easiest way to prove that was visual analyzing of images. To make the data visible to human eye some imaging operations needed to be performed. A simple Opencv program was created to transform the raw data to a video and png images.

The first image operator used was scaling to a 8 bit image using static manual threshold values. A dynamic thresholding based on neighboring pixel averages was performed after that. Finally rotation to correct horizontal line. Alternative to the manual thresholding could be a histogram analysis. Figure 17 presents a processed image, using around 6% range in pixel values compared to the original data, compressed in a 8 bit gray-scale image.

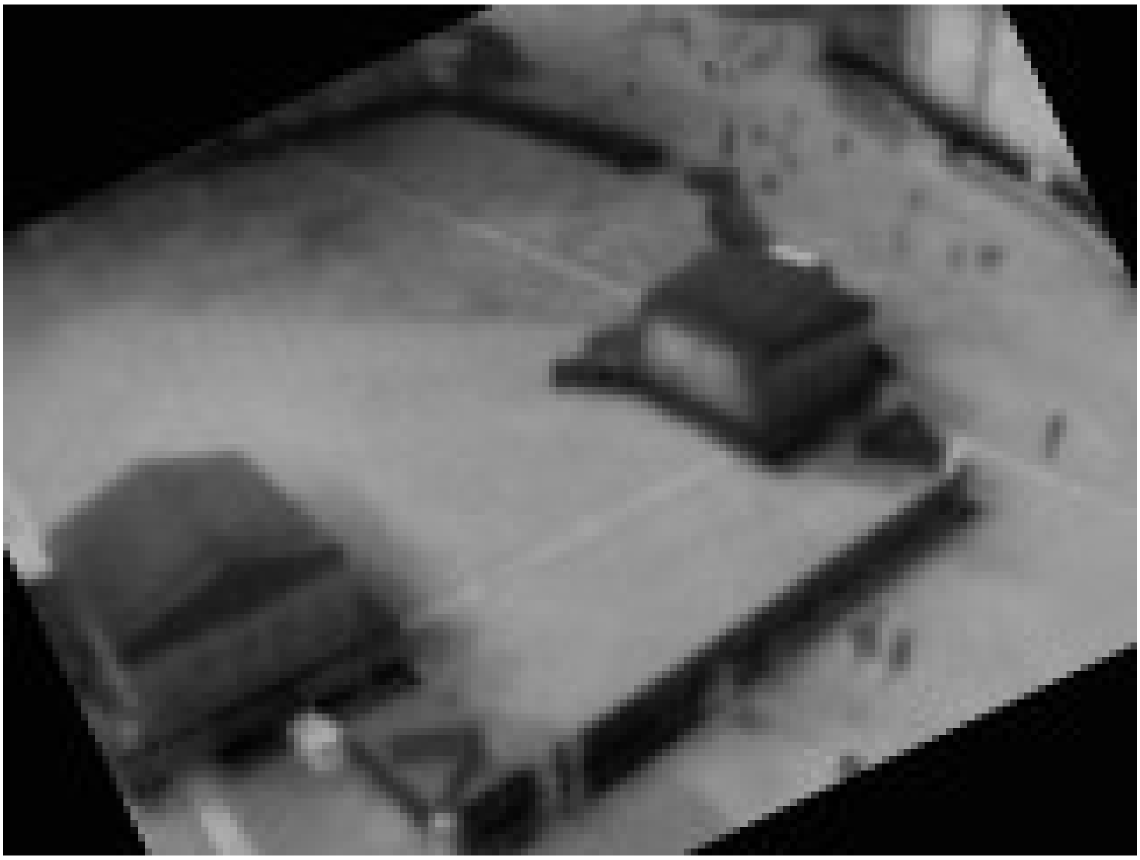


Figure 17. Processed thermal image from Suvilahti festival area.

Figure 17 shows a thermal image from Suvilahti festival area. The image is rotated to match horizontal line. Tents in the image are a mixer booth and a light control center. Black spots on the right side and the bottom side are people. Darker pixels are warmer than brighter.

The data shows that individual humans and groups are clearly visible from the background even with just a quick image processing. More thorough processing would reveal even more details in the data, details possibly visible only to a computer. Some of these operators are explained in the following chapter. Figure 18 presents an another image that shows a larger group of people against a fence. Figure 18 also demonstrates black stripes caused by poor WiFi signal.

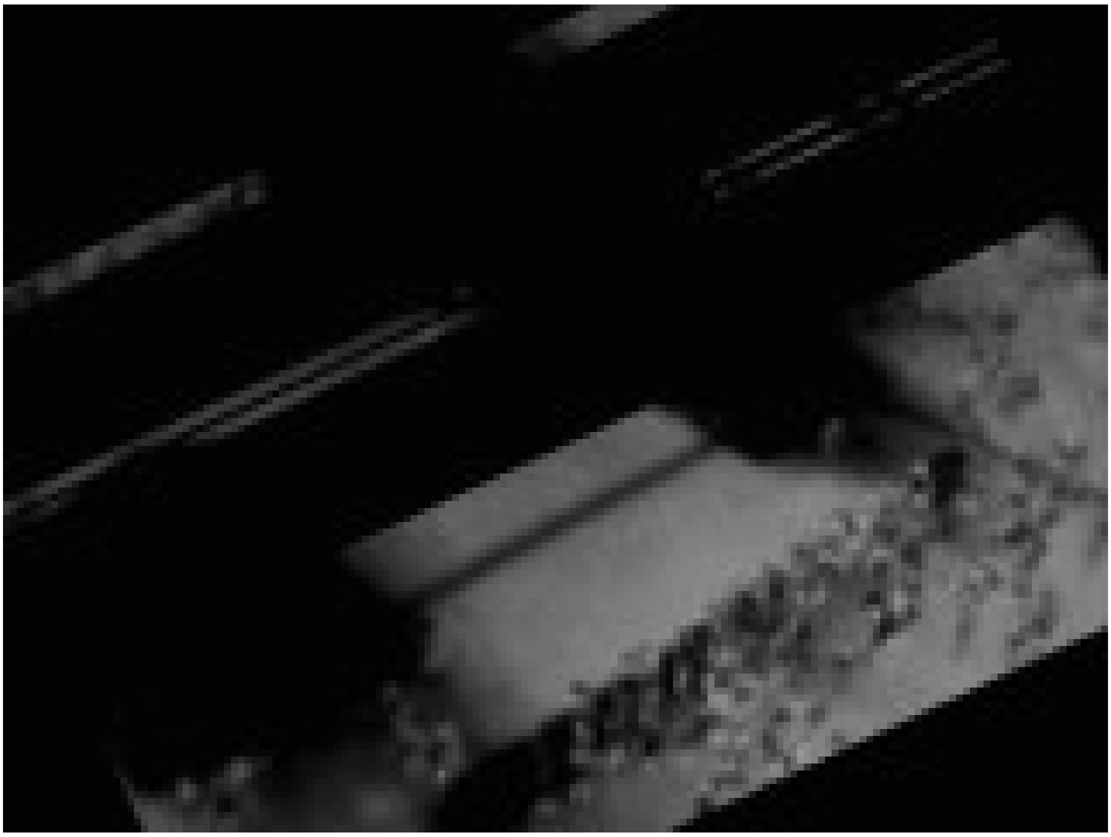


Figure 18. Processed thermal image of larger crowd of people.

Figure 18 is taken from the same area as figure 17. Black stripes are from transmission errors. Bottom part of the image there is a visible larger crowd of people. An estimation by eye says around 50 to 100 people near the fence. With video individual humans can be detected from a crowd and accurate estimations about density and head count can be made.

6 For future

This section explains several techniques that can be used to create a machine vision program for data analysis. The focus is for continuum of following projects, and to provide improvement suggestions.

6.1 Image processing

This project is meant to be the first phase in a longer goal to generate advanced systems that can use thermal cameras as sensors. In following chapters few image processing techniques useful with an one channel data such as a thermal camera data are explained.

Scaling down

Thermal data from these cameras is 14 bit wide. It contains a number of useless data and to make a processing easier a thresholding should be one of the first operations to perform. The thresholding is an operation where low and high limits are set and every pixel value outside that range is discarded. A histogram analysis is a histogram made from all pixel values in an image and determining threshold values based on peak values in the histogram. After thresholding pixels should be scaled to 8 or 10 bit ranges depending on an accuracy needed.

Contrast enchanting

Contrast can be improved by linearly increasing higher pixel values and lowering lower pixel values, or same operation around histogram peaks. It can be made in a full image level or for each pixel individually comparing to its neighboring pixels. Latter is usually better but takes more processing cycles to complete.

Smoothing

To merge multiple heat sources together and separating cold areas from hot ones smoothing is recommended method. In this case a Gaussian or median filter would probably be the best option. The Gaussian filter uses Gaussian values as weights to make pixels similar values compared to neighboring pixels. The median filter takes nearby pixel values and chooses a median value for each pixel. With the median filter it is usually a good idea to run it a couple of times. Both filters can be used in each frame.

Thresholding

To clearly say which pixels are people and which are background, a thresholding is needed. A dynamic thresholding based on a histogram of smaller areas is recommended unless a background in full image is absolutely uniform in this stage. This time thresholding should make background pixels zeros and everything else keeps its current value.

6.2 Improvements

Data transfer

A bottleneck for this project was a data transfer method. WiFi was used since it is easy to setup and installation part offers more freedom compared to wired data transfer. However WiFi generally is not reliable enough for a security purposes or any purpose in crowded area. Replacement for WiFi could be wired data transfer to a centralized access point. Offline data storing could be carried out there and sending to public network handled in more powerful device than camera.

The best solution for wired data transfer would be using Ethernet cable. POE (Power over Ethernet) cable could be used to power camera to keep wiring minimal. Hardware changes would require Ethernet adapter and possibly new platform. Esp-8266 should be able to handle communicating with Ethernet adapter when WiFi would not be used but build in WiFi capabilities would be wasted.

If wireless transmission is required feature good alternative to WiFi would be 4G modules inside camera. If used with priority channels 4G network is much less crowded than WiFi in festivals. These priority channels require priority sim-cards that may be difficult to get. Modems would also remove the need for external system delivering data to public network thus making the cameras even easier to install and maintain.

Another way to upgrade wireless data transfer would have been usage of 5.8 GHz WiFi band instead of 2.4 GHz band that is embedded in Esp-8266 12E module. This would require changing platform to another. WiFi bands are divided into channels. Around 13 channels that partially overlap are available in 2.4 GHz standard. This makes total number of individual channel to 3. At least 23 non overlapping channels are in 5 GHz band that makes it better option.

One of the main purposes for this project was to prove thermal imaging is useful monitoring festival audience. Data gathered in field test is enough to make conclusions. This was the reason why the cameras were not modified again after test and left as they were.

Platform

To achieve offline data storing in camera Esp-8266 needs to be replaced with microcontroller with more computing power or IO-ports. Storing data through SD-card using SPI mode is too slow to save the stream coming from camera. Native way to store data to SD-card is needed. This requires 6 IO pins to handle data transferring and enough time in processor to handle timings. Esp-8266 could handle offline storing and reading camera module without sending data outside of camera if there would be more IO-ports.

More testing

To satisfy customer and teachers additional tests are to be made in future if suitable testing conditions are found. They are not included in this thesis.

6.3 Possible usages

Purpose of this project was to create generic camera which can be later used for different purposes. Camera system is fully independent system and data can be accesses from server. Following part explains different possible usages for the camera and the data.

Security in festivals

One of the original ideas was that an image data can be used in festival areas to detect possible dangerous situations. In example too many people in tight area or a knife fight in an audience. By analyzing people's movements in a crowd an automated computer system could detect people suddenly moving away from certain point or direction. There could be visible hotspots in the image where people are too tight. From image data applications could calculate how many people are inside the image.

If the cameras are not placed too high, somewhere above 30 meter, individual people can be detected from the image easily. This would allow monitoring restricted areas to prevent unauthorized personnel moving freely there. It could be problematic to separate a staff member from an audience based only on a pixel value. Applications that monitor people could raise an alert when specific barriers are crossed and ignore staff members using allowed paths.

By detecting individual humans system could also detect when specific people haven’t been moved for long time, and healthcare personnel could check if those people are too intoxicated, having sudden attack or even dead.

Marketing

In festivals there could be an application that automatically calculates and predicts where people are and will go. With on site screens people could be directed to empty bars or restaurants to increase selling and keep track that popular sites have enough products to sell. Food in restaurants could be prepared when people are moving towards restaurant areas.

Heat source monitoring

Any monitoring system can easily be made to gather statistics. With thermal cameras statistics can be gathered from heat sources. Possible usages are to improve a thermal insulation of buildings or minimize energy losses in closed thermodynamic environments by detecting spots where energy escapes monitored area as heat. If installed in drones monitoring a ground surface and combined with terrain knowledge most optimized spots for solar panels and geothermal heating could be calculated.

Hospitals

Thermal camera can be used to detect patients presence in a bed or in a hospital room. I can also measure patient's body temperature. This usage has been tested and proved working in Metropolia school with lower resolution, 4 x 16 pixel thermal camera.

Fire safety

Burning generates a great deal of heat. These cameras could monitor forest areas to detect forest fires. Dry savannah areas could be similar target in case of grass or bush fires.

City surveillance

One good aspect with thermal image is that individual people cannot be identified. It allows surveillance operate in areas where using normal cameras would need permit to film people and notify them they are being monitored. Thermal cameras could be used to monitor entire cities to help authorities in emergency situations and in everyday situations.

7 Conclusions

Data gathered during the field test proves that thermal cameras are useful in monitoring large amount of people. They don’t suffer from too much light or darkness such as normal cameras do. The data gathered shows individual people as well larger crowds. An estimation about amounts, behavior and density of people could be achieved with an advanced computer vision application.

WiFi is not an optimal method for sending a live image feed in busy areas, but it works in quiet environments with a good access point. The cameras produced can be used again without modifications if the transmission quality allows and no perfect result is required, but then it is recommended changing the data transfer method. Otherwise the hardware can be the same. The camera module itself performs well and is easy to implement with other platforms.

The total outcome for this project fulfills the requirements set at the beginning of the project despite multiple setbacks. But as one of the teachers said, engineering is about trial and error. It is a way to know for certain what does and what does not work.

References

1 Photo resistor. Online material. <http://www.resistorguide.com/photoresistor/> Accessed 5.10.2017.

2 CCD vs CMOS. Online material. Teledyne dalsa inc. <https://www.teledynedalsa.com/imaging/knowledge-center/appnotes/ccd-vs-cmos/>. Accessed 5.10.2017.

3 The Electromagnetic Spectrum. Online material. <https://imagine.gsfc.nasa.gov/science/toolbox/emspectrum1.html> . Accessed 5.10.2017.

4 Basic lens specifications. Online material. Kenrockwell.com. 2008. <http://www.kenrockwell.com/tech/lens-specifications.htm>. Accessed 5.10.2017.

5 Distortion. Online material. <https://www.edmundoptics.com/resources/applicationnotes/imaging/distortion/> Accessed 5.10.2017.

6 Flir Commercial Systems. 2014. FLIR LEPTON 3 Long Wave Infrared (LWIR) Datasheet. Revision 100.

7 Interlaced vs. progressive scanning methods. Briene, Danny; Hurley, Pat. Online material. <http://www.dummies.com/consumer-electronics/home-theater/interlaced-vs-progressive-scanning-methods/>. Accessed 5.10.2017.

8 Serial peripherial interface (SPI). Online material. Sparkfun electronics. <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>. Accessed 5.10.2017.

9 I2C. Online material. Sparkfun electronics. <https://learn.sparkfun.com/tutorials/i2c>. Accessed 5.10.2017.

10 TCP vs. UDP. Online material. <http://www.diffen.com/difference/TCP_vs_UDP>. Accessed 5.10.2017.

11 Espressif Inc . 2017. Esp Technical reference. Version 1.3.

12 Esp modules. Online material. <http://www.esp8266.com/wiki/doku.phpid=esp8266-module-family>. Accessed 5.10.2017.

13 Quad SPI. Online material. <http://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.nrf52840.ps%2Fqspi.html>. Accessed 5.10.2017.

14 Esp Boot process. Online material. <https://github.com/esp8266/esp8266-wiki/wiki/Boot-Process#esp-boot-modes>. Accessed 5.10.2017.

15 Technical committee SD card association. 2007. SD Specification Part 1E SDIO Simplified Specification. Version 2.0.

16 How to use MMC/SDC. Online material. <http://elm-chan.org/docs/mmc/mmc_e.html>. Accessed 5.10.2017.

17 ESP8266 Thing hookup guide. Online material. <https://learn.sparkfun.com/tutorials/esp8266-thing-hookup-guide/using-the-arduino-addon>. Accessed 5.10.2017.

18 Wire Library. Online material. <https://www.arduino.cc/en/Reference/Wire> Accessed 5.10.2017.

19 Image thresholding. Online material. <http://docs.opencv.org/3.2.0/d7/d4d/tutorial_py_thresholding.html>. Accessed 5.10.2017.

20 Adaptive thresholding. Online material. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm> Accessed 5.10.2017.

21 Histogram equalization. Online material. <http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_equalization/histogram_equalization.html> Accessed 5.10.2017.

22 Smoothing images. Online material. <http://docs.opencv.org/2.4/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html> Accessed 5.10.2017.

23 Mesenbrink, John. 2001. Online article. <https://www.securitymagazine.com/articles/77639-protecting-borders-with-thermal-imaging-1> Accessed 5.10.2017.

Appendix 1. Lepton 3 Power modes

Figure 1 shows Lepton 3 power modes. Shutdown mode is during initialization. Reading Lepton 3 power register tells which mode the module is currently. During an off state the register cannot be read. [6.]

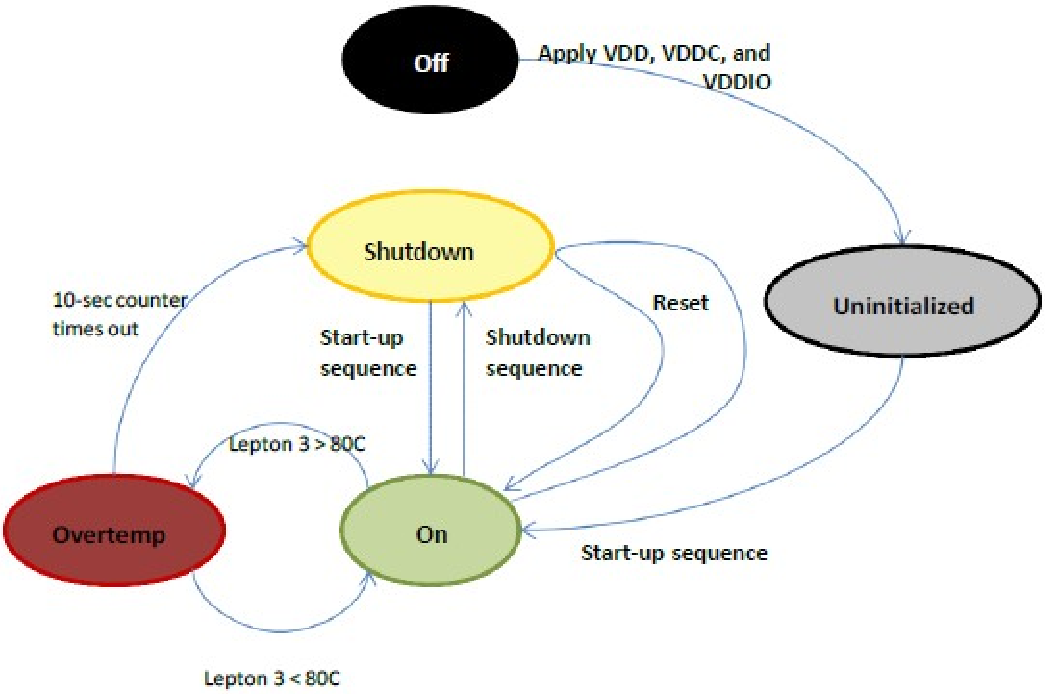
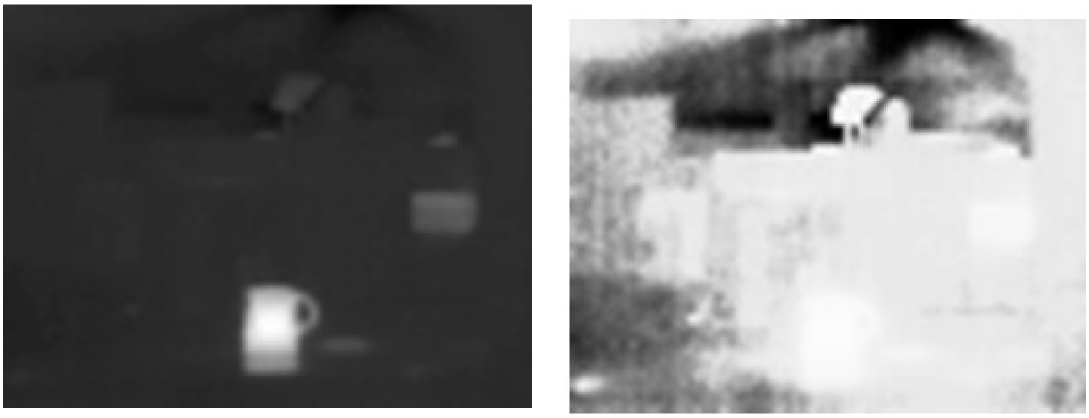


Figure 1. Lepton 3 power modes. [6.]

Figure 1 illustrates Lepton 3 power modes. When the camera module powers on, the off state changes towards the on state. The camera stays at on state if module doesn't overheat or is commanded to perform reset or shutdown. [6.]

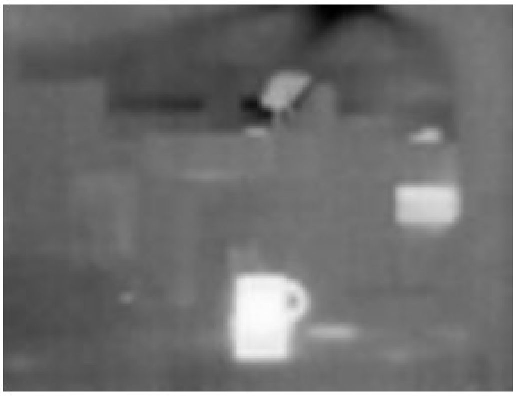
Appendix 2. Automatic gain control in Lepton 3

Lepton 3 has option to use 14 bit to 8 bit conversion. AGC (Automatic Gain Control) is a process where this conversion is carried out to set a range more appropriate for human eye. Simplest way to transform 14 bit data to 8 bit is to perform linear mapping. Lepton 3 AGC algorithm uses s modified histogram equalization from full image to determine the best range to do the scaling. It has some additional features compared to a classic histogram equalization. Figure 1 shows differences between these three techniques. [6.]



(a) Linear AGC

(b) Classic Histogram Equalization



(c) Lepton's Variant of Histogram Equalization

Figure 1. Comparison of linear AGC, Classic Histogram Equalization and Lepton 3 variant of Histogram Equalization. [6.]

In figure 1 can be seen three images from the same scene. One bright heat source is in the middle of the scene, that appears to be a cup, and few other objects and a non-uniform background. Different images gives an idea how Lepton 3's AGC algorithm performs compared to more traditional methods. [6.]

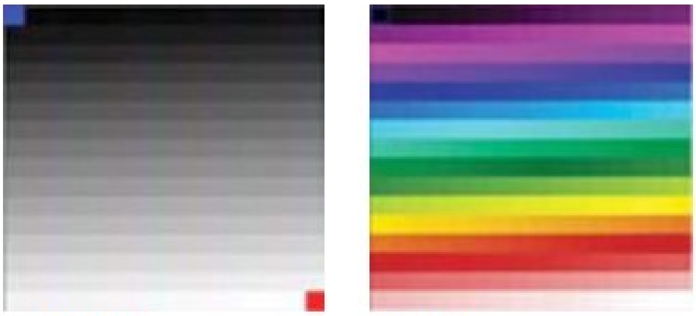
Appendix 3. Color lookup tables

Figure 1 shows different options for 8-bit gray to color conversion. The upper left corner represents the color associated with an input value of 0. The lower right corner represents the color associated with an input value of 255. [6.]

Figure

(a) Wheel 6 (b) Fusion (default) (c) Rainbow

(d) Globow (e) Sepia (f) Color



(g) Ice Fire (h) Rain

Figure 1. Lepton 3 color conversion LUT choices. [6.]