TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Computer Engineering

IAY70LT

Christel Nilsen 131076IASM

LOCALIZATION METHODS FOR ELDERLY PEOPLE ASSISTANCE

Master thesis

Thomas Hollstein
Prof. Dr.-Ing.
Professor

Kalle Tammemäe
Prof. Dr.
Professor

Mairo Leier
M.Sc.
Early stage Researcher

Tallinn 2015

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Christel Nilsen

Date:

Abstract

This master thesis is about localization methods for elderly people assistance. People are getting older and many of today’s technology are used to assist them. Several different sensors and surveillance methods are in use for elderly people assistance. This thesis look at how the Omron thermal IR sensor can be used in pair with Texas Instruments microcontroller. Scenarios are modelled in Google SketchUp. The thermal sensor is mounted in the upper corner of a room and several tests are done. Calibration of the sensor in a specific room scenario has been evaluated and objects detected by the sensor were classified. As a result two algorithms for detection of people using the Omron thermal IR sensor was developed. The algorithm that was defined and the pattern recognition theory used are small enough to be implemented on microcontroller. In the last part of this thesis the conclusion of work and outline for further work are discussed.

This thesis is written in English and is 65 pages long, including 5 chapters, 46 figures and 38 tables.

Annotatsioon

Asukoha tuvastamise meetodid eakate inimeste heaolu järelvalveks

Käesolev magistritöö käsitleb eakate inimeste asukoha tuvastamise meetodit. Inimesed saavad vanemaks ja on mitmeid tänapäevaseid tehnoloogiaid nende abistamiseks. Mitmeid erinevaid sensoreid ja jälgimise meetodeid on kasutusel vanurite abistamiseks. Käesolev magistritöö käsitleb Omron infrapuna termosensori kasutamist koos Texas Instrumentsi mikrokontrolleriga. Stsenaariumid on modelleeritud Google SketchUp abiga. Termosensor on ühendatud ruumi ülemisse nurka ning teostatud on erinevaid teste. Teostatud on sensori kalibreerimine konkreetses ruumis ning leitud objektide klassifitseerimine. Arendatud on inimeste tuvastamise algoritmid Omron infrapuna termosensori abil. Arendatud algoritm ja kasutatud mustrituvastusteooria on piisavalt väikesed kasutamaks neid mikrokontrolleril. Töö viimases osas on kirjeldatud kokkuvõte ning võimalikud arendused.

Antud töö on kirjutatud inglise keeles, sisaldab 65 lehekülge, sealhulgas 5 peatükki, 46 joonist ja 38 tabelit.

Table of abbreviations and terms

ASIC	Application-specific integrated circuit
BD-LLC	Bi-Directional Logic Level Converter
CCS	Code Composer Studio
CRC	Cyclic Redundancy Check
DIP	Dual in-line Package
FOV	Field of view
I2C	Inter-Integrated Circuit (pronounced I-squared-C)
IoT	Internet-of-Things
IPS	Indoor positioning system
LAN	Local Area Network
LPM	Low Power Mode
MEMS	Micro-Electro-Mechanical Systems
MSP	Mixed-signal processor
PCB	Printed Circuit Board
PEC	Packet error check code
QVGA	Quarter Video Graphics Array
RFID	Radio-frequency identification
SCL	Serial Clock Line
SDA	Serial Data Line
SRN	Signal-to-noise ratio
TOF	Time-of-Flight
UART	Universal Asynchronous Receiver/Transmitter
USCI	Universal Serial Communication Interface
VLO	Visible Light Communication
WiMAX	Worldwide Interoperability for Microwave Access

Table of contents

List of figures

List of tables

1. Introduction

This document contains the Master Thesis done in the spring of 2015 at Tallinn University of Technology under Faculty of Information Technology Department of Computer Engineering and the Master programme Computer and System Engineering with specialization Computer System Design.

The subject of this master thesis is Localization Methods for Elderly People Assistance. The thesis start with the tasks to examine state-of-the-art methods for elderly surveillance and state-of-the-art of position detection in living environments. The next is to state the problem and work with the development of one method for localization of elderly people assistance.

The motivation for this choice of the thesis is an interest in programming of microcontrollers and an interest to see things happens. Curiosity and interest of figure out things, is also a motivation, especially what can be done and how? There is no "*it doesn't work*", it is "*how can it be done* ".

This thesis is also topical in a world where there will be increasing amount of elderly people. In Norway in 2013, 13% of the population was over 67 years, and it is expected that in 2050 21 % of the population will be over 67 years. This is commonly called the "*Eldrebølgen*" (*Elderly wave)* in Norway [1], [2]. Many of them will need a lot of assistance. Today relatives and children are working full time and therefore are not able to take care of their elderly parents. The society may not have enough staff or finance resources to give all the assistance the elderly will need. Especially important is the situation for people with dementia and Alzheimer. For example, around 1000 elderly people are reported missing in Norway every year and some of them were never found [3]. Erlend Aarsæther at Norsk Folkehjelp/Norwegian People's Aid says *citation* "*dementia patients can get pretty far away from the place they disappeared, especially in cities where public transport network is widespread"*, and he told he *"participated in a search and rescue that started in Bærum* (in Oslo, Norway, authors remark), *and ended in Stockholm, Sweden*". Patients with Alzheimer will wander and many of them will die if they are not found within 24 hour [4]. Therefore, it is critical to detect unusual behaviours as soon as possible. Another important issue is fall detection. Many elderly that fall will not be able to get up after fall and will need assistance. In many cases it is in the most private rooms where these accidents happen, rooms like bedrooms, bathrooms and toilets. Privacy concerns make it complicated to use traditional camera surveillances. Faster detection and reaction in cases of accident can be crucial.

1.1. Problem statement

The problem will be accessibility and provide necessary assistance when an accident happens without violating privacy. How can we use technology available today to help our elderly generation?

The big question is: is it possible to detect this situation without violating privacy? (*Figure 1*)

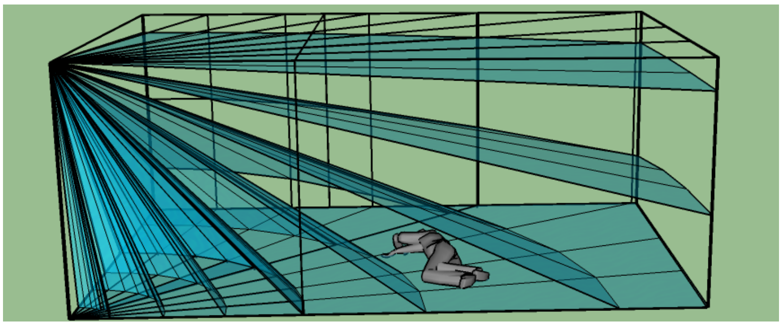


Figure 1 Example of person being lying on floor

This master thesis is a project of one person. Supervisors are Prof. Dr.-Ing. Thomas Hollstein, Prof. Kalle Tammemäe and Early-stage researcher Mairo Leier. The meetings with the supervisors are on demand and frequent basis. OMRON Thermal Sensor, LogicPort logic analyzer, Logic Level Converter, some resistors and some wires were provided by the Tallinn University of Technology, Microcontroller, breadboard, some resistors and LEDs are owned by author. This thesis main task was to explore OMRON MEMS IR Thermal Sensor D6T-44L and develop a method for localization methods for elderly people assistance using this sensor.

The works are divided into these steps

- Gain initial basic knowledge of theories.
- Be familiar with the sensor and hardware.
- Connect D6T-44L sensor to microcontroller
- Transmit temperature data to computer and save the data into file and database.

- Test the sensor for different cases such as lying on the floor, lying on the sofa, sitting on the chair and standing.
- Classification of different patterns detected by the sensor.
- Examine and develop suitable algorithms for usage with this sensor.

1.2. Organizing

The first part of this thesis is to describe state-of-the-art of methods for the surveillance of elderly people and state-of-the-art position detection in the living environment. Chapter 2 will also compare some of the sensors used in position detection systems. The next step is to get an overview of specified equipment and theories used in this master thesis. Chapter 3 will explain the developed method. Chapter 4 will show results of the experiments done. The last chapter will come up with conclusion and outline for improvement and further work.

1.3. Limitations

The main focus was to explore, utilize and test the hardware, which is the OMRON D6T-44L sensor. Since the theory around pattern recognition is so wide and extensive it is only mentioned briefly in this master thesis.

One limitation is that the practical part and software was completed using only one sensor. All discussion of the usage of multi sensors are only theoretically described but not tested.

Prototype connecting microcontroller, sensor and computer had physical limitations making extensive testing complicated. With test setups, position and angle of sensor could not be determined with accuracy.

Because of limited time and resources all tests were carried out with a single test person. There was no test with multiple persons done or completed.

Java Swing application is developed using JDK- and NetBeans 32-bit because Java.comm API is 32-bit. The code will not work in a 64-bit environment.

2. Theory and methods

"Surveillance Technology" is a big topic. There is substantial research and development taking place worldwide. This chapter will describe a few approaches.

2.1. Methods for elderly people surveillance: State-of-the-art

Motions sensors and camera: Many seniors give permission to their caregivers to install camera and motion sensors used for tracking and surveillance so they can live independently at home longer [5]. Motion sensors and cameras are placed around the home. When these sensors detect something unusual they send a text message to the caregivers so they can check their relatives and in case of emergency caregivers can act accordingly.

A tele-operated mobile robot system: Robot communication [6]. The robot is equipped with microphone, speaker and monitor to communicate with the user, and it can also check the patient's condition. The robot navigates in the house searching for an unusual situation or elderly people. If the robot finds a person lying on the floor it inform the operator who is responsible and checks the persons condition. This robot is connected to LAN/internet environment. It has a depth/RGB image sensor, a temperature sensor and pressure sensor. And there is an "Operator" side, where one uses a computer to communicate with the robot. It has two operation modes, autonomous with a regular route and manually mode which is operated by one operator. In this test they had a limitation that all patients were having the same clothes and the target clothing color was blue. The target facilities are hospitals or care houses.

Fall detection: MEMS sensor is composed of an accelerometer, which measures acceleration in x-, y- and z-axis [7]. The wireless sensor needs to be carried on the body. The acquired data was sent wirelessly to ZigBee. Normal activity, static and dynamic were classified. Several "falling" scenario were also classified as "falling when walking", "falling when standing up from a chair" and "slipping over the floor", to name a few. When using wireless signal transmission there is no need for cabling between the sensor units. The MEMS are small (volume of the system is 33 mm x 33 mm x 4mm), using SPI, I2C, PCB antenna and a rechargeable battery. The test result demonstrated the daily activity with high sensitivity and specificity.

Video surveillance: There are different uses of video surveillance at home for elderly people who need extra care because of illness or accident. Omni-directional Vision Sensor together with other sensors track and detect motion and behavior [8]. However to set up video surveillance within a home it must typically be approved by the person to be monitored. In many cases it is not possible perhaps because the person has Alzheimer's or dementia and is not considered capable of giving such permission.

"Granny cams": Other camera surveillance technology is using "*Granny cams*" [9]. Sometimes relatives and family perform hidden camera surveillance of their elderly parents to expose inadequate care, abuse or neglect from caregiving personnel.

GPS tracking: Several GPS-tracking system for tracking elderly people exist on the market today [10]. For location the sender is a small unit such as a wristwatch, brick or even shoe sole [11]. To track the person, smartphone, computer or other devices are used. The system can be configured to have safe zones and give an alert if the person is leaving the predefined area and it also may have a button so the user can call for assistance if needed. Some of the trackers can also be used to track children, teens and pets.

Safety alarms for elderly people: Most municipalities have security alarms offered to elderly people living at home. In case of accidents or if the elderly person feels the need for assistance or help, they have an alarm button which they can use to contact healthcare personnel.

2.2. Position detection in living environments: State-of-the-art

Position detection in an indoor environment is also known as Indoor Positioning System IPS for detecting people and objects inside of buildings. Technology uses RFID, Bluetooth LE, Wi-Fi, Visible light communication and ultrasound, just to name a few.

Locatible is a real time location system, even with a patient fall alert, using Bluetooth indoor location. It is cloud based, real time, it finds people and assets, has 3 feet accuracy, it does not use RDIF [4]. Locatible is a commercial system.

MazeMap is another indoor path and position navigation system [12]. It uses Wi-Fi access points to set the position and the accuracy will vary with the quality of infrastructure, such as placement of the Wi-Fi access point. This is an interactive indoor map used with Smartphones, IPads, and "all" devices. It is a commercial system.

Visible light communication and ultrasound. [13] VLO is uses LED light bulbs with a small microcontroller inside that transmits data. The LED is flickering (switched on and off) to generate digital 1's and 0's. The LED can be used for both sending and receiving. In the test the transmission achieved higher data rate than Wi-Fi, Bluetooth and Wi-Max. The drawback is that when the LED is turned off all communication is terminated. Exact position of the LED lamp is determined using http://mygeoposition.com/ and hardcoded as a global position onto Arduino. High precision ultrasonic sensor is used to improve readings. An ultrasonic sensor sends a sound wave and calculates the distance based on the wave bounced back. LED lamps send their own position and distance to an object from an ultrasonic sensor to a smart phone integrated with a LED receiver. Current prototype is dependent on mobile phone detection.

A new LED lamp using radar for the surveillance/monitoring of elderly people is being developed in Japan (Sep. 2014). The LED lamp has a radar device, developed by Panasonic, and a wireless LAN chip embedded in it [14]. The radar works at 24 GHz with a wavelength of 12 mm and can measure changes down to 3 mm. The radar can detect a person in a distance of up to 8 meters with an angle of 160°. In a normal room it covers the area around 13 m2 and can detect multiple persons. The radar can monitor abnormalities, breathing and sleeping patterns of the person by measuring the distance between the LED lamp and the person and then analyzes the reflected light. The elderly person does not need to wear any kind of devices or sensors. "CQ-S Net", is the company that deals with a remote nursing system, which obtained a patent for development of the system "Laser light" that is used for monitoring and transmission of results over a wireless LAN. The system can notify the caregiver or family if the elderly persons physical condition suddenly change. It seems to be designed as a replacement for an ordinary light bulb and does the installation is easy. It may be powered by the lamp, but if lamp is turned off and no power is supplied, the system may not work [15].

3D depth sensing and RGB-d camera: 3D depth sensing TOF camera uses infrared light to measure the distance from the camera to the object. It resolves the distance based on *speed-of-light* technology. TOF cameras provide low resolution (QVGA) imaging but a high refresh rate [16]. This technology is used to detect distance and for fall detection. Time-of-Flight equipment is currently still expensive.

The benefit of using in-depth camera with infrared LED is that the camera will be independent of illumination of light and can work even in dark rooms, for example in the night. With in-depth camera each pixel value represents the depth value (the distance between the camera and the object) instead of color information. Using an in-depth camera the object size and color are changing according to the distance from the object to the camera and it is also possible to calculate the distance using trigonometry. An in-depth camera can be used in fall detection.

OMRON's D6T MEMS Thermal Sensor [17] is a non-contact infra-red sensor and has a high sensitivity that enables detection of stationary human presence in a room without the need for movement [18]. It reads surface temperatures. It can detect heating, cooling and overheating. It can be used as a controlling device for heating and cooling equipment and even as a fire alarm. It can be used to roughly estimate number of people in a room and detect where the person is in the room and it can also be used to detect if a person is leaving a predefined area.

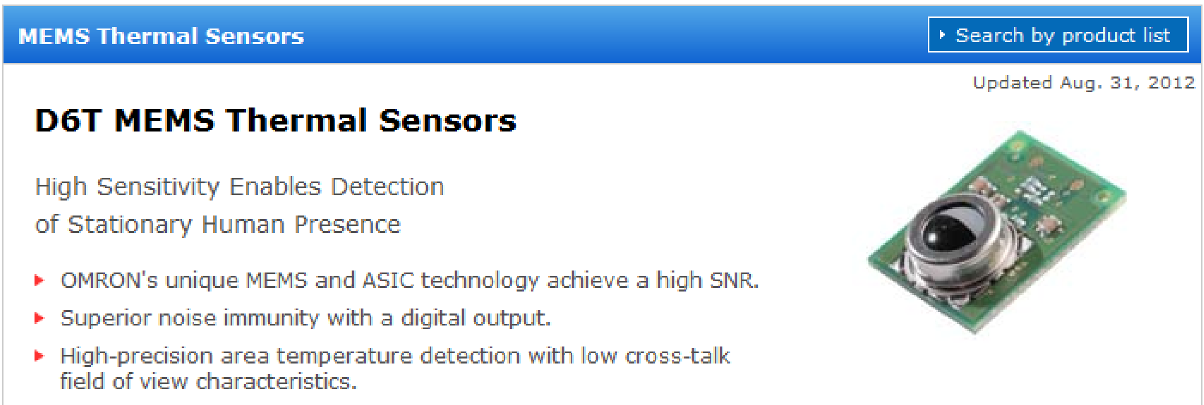


Figure 2 D6T MEMS Thermal Sensors [19]

Kinect sensor. A new project using the Microsoft Kinect sensor for fall detection is ongoing at the hospital in Norway. Hospital Østfold in cooperation with Microsoft Norway is doing a proof of concept for fall detection targeting patients suffering from dementia and epilepsy [20]. To detect the position of the patient, it uses a 3D camera game technology mapping skeleton based on reference points. Images of the patient are not transmitted, only an alarm event is sent, and in that way the privacy of the patient is taken into account.

DUO3D – DUO mini lx [21] [22]**.** This is a mini 3D camera that can be used in various applications. It is very small and lightweight (12.5 g), high speed, and it can work both indoors using integrated IR Led array and outdoors with environmental IR. It has mid-range detection. The framework provides support of C/C++, C#, Python among others.

2.3. Comparing non-contact sensors

Table 1 Comparing non-contact sensors.

	Omron D6T-44L [17]	**Laser Light [14]**	**PIR Motion sensor [23]**	**In-depth camera, SR4000 [24]**	**Ultrasonic Range Finder - LV-MaxSonar-EZ1 [25]**	**HC-SR04 Ultrasonic Range Finder [26]**	**DUO mini lx Compact + Light [22]**
Cost	$ 49,88		$ 9.95	$ 4295	$ 25.95	$ 2.50	$595.00
Voltage	4.5-5.5 V		5-12 V (jumper wire 3.3 V)	12 V	2.5-5.5V	5 V	5 V from USB
Current	5 mA		1.6 mA @3.3 V	1 A	2 mA	2 mA	0.5 A (~2.5 Watt @ +5V DC from USB)
Communication form	I2C			USB 2.0 or Fast Ethernet	RS232 serial output 9600 bps		Hi-speed USB 2.0 (480Mb/s)
Send data	frequently	frequently	when detected	frequently	frequently	frequently	frequently
Detection distance	6 -7 m	8 m	6 m	5 m or 10 m	0-6.45 meters	2 - 500 cm	
Detect stationary human	Yes	Yes	No	Yes	Yes	Yes	Yes
Precision	± 1,5 ° and ± 3,0 °	3 mm					
FOV, Angle in °	44.2 ° x 45.7 °	160 °	110 ° x 70 °	43 °x 34 ° or 69° x 56°		effectual angle: < 15°	170 °
Cover area in m2		13 m2					
Detect temperature	0-50 °C	No	No	Yes	No	No	Yes

	Omron D6T-44L [17]	**Laser Light [14]**	**PIR Motion sensor [23]**	**In-depth camera, SR4000 [24]**	**Ultrasonic Range Finder - LV-MaxSonar-EZ1 [25]**	**HC-SR04 Ultrasonic Range Finder [26]**	**DUO mini lx Compact + Light [22]**
Size	18 x 14 x 3,5 mm		35.4 x 30.6 mm	65 x 65 x76 mm	19,9 x 15,5 x 15,5 mm	2.0 x 4.3 x 1.5 mm	52 x 24 x 11 mm
Other	Design to be integrated to other product	Release sep.15	Design to be integrated to other product	IP40		Design to be integrated to other product	Design to be integrated to other product

Common for all these sensors is that the user does not need to carry any specialized device.

The choice of OMRON Thermal sensor can be justified by the following features:

- Small in size
- Low power consumption, dependent of power management on microcontroller (turn on and off power supply between readings)
- Easily integrated into other products
- Well know and mature suite of protocols
- High level of privacy
- Reasonable and competitive price
- Can detect stationary people
- Detection range is satisfactory for the chosen project
- No additional equipment needed, no devices to carry.

3. IR-based person detection: developed method

This master thesis will look at Omron's D6T MEMS series Thermal IR D6T-44L 4x4 pixel sensor [27]. The microcontroller used is Texas Instruments MSP430g2553 Launchpad [28]. This is a small sized ultra-low power microcontroller. Ultra-fast Wake-Up time makes it possible to go from stand-by to fully operational mode in less than 1 µs. Many different communication extensions makes the microcontroller easily integrated into networks – Internet-of-Things (IoT). The first step is to setup the sensor. The sensor is connected to the microcontroller through SparkFun's Logic Level Converter, in *Figure 3,* sensor uses 5 V and microcontroller uses 3.3 V and therefore a level conversion is needed between the microcontroller and the D6T-44L sensor.

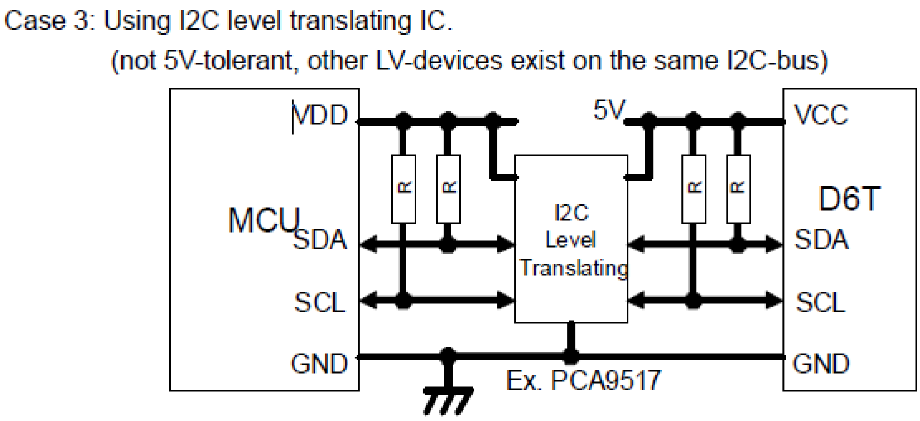


Figure 3 Using I2C logic level converting [29].

Table 2 Components used in this project.

Omron D6T-44L MEMS Thermal sensor	 [27]
Texas Instruments MSP430g2553	 [28]
SparkFun Logic Level Converter - Bi-Directional	 [30]

And important thing to notice in *Figure 4* is that one extra pin is soldered to give the sensor 5V power and also that the jumper on P1.6 has been removed, P1.6 is also used for green LED.

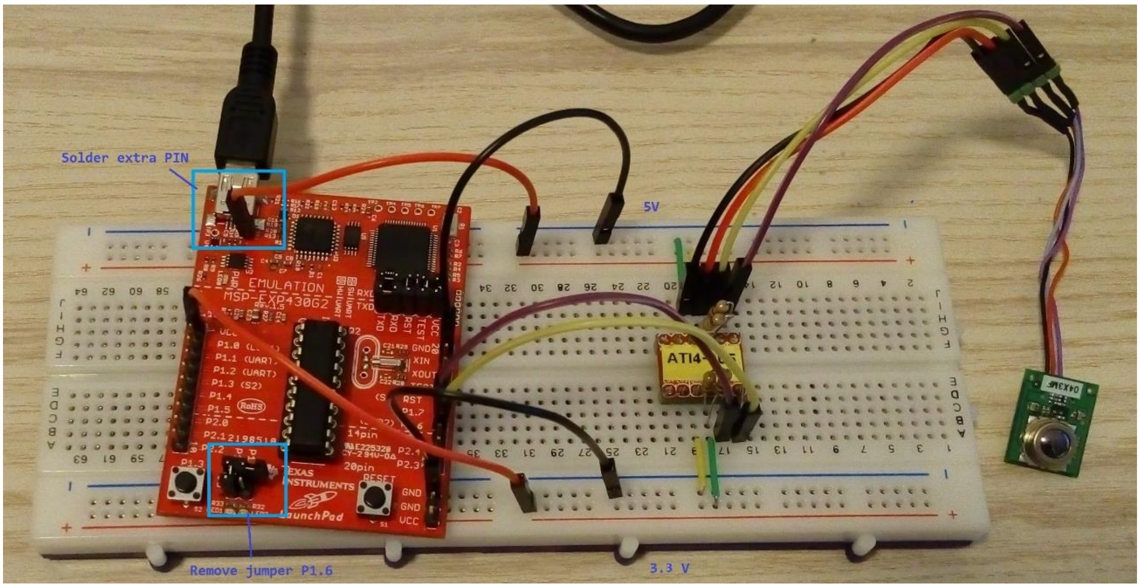


Figure 4 Image of microcontroller connected to sensor.

In project overview, *Figure 5*, the microcontroller reads temperature data from the D6T-44L MEMS sensor, and sends it to the computer. The prototype Java Swing application visualizes the data in real time and saves temperature data to the database and csv file.

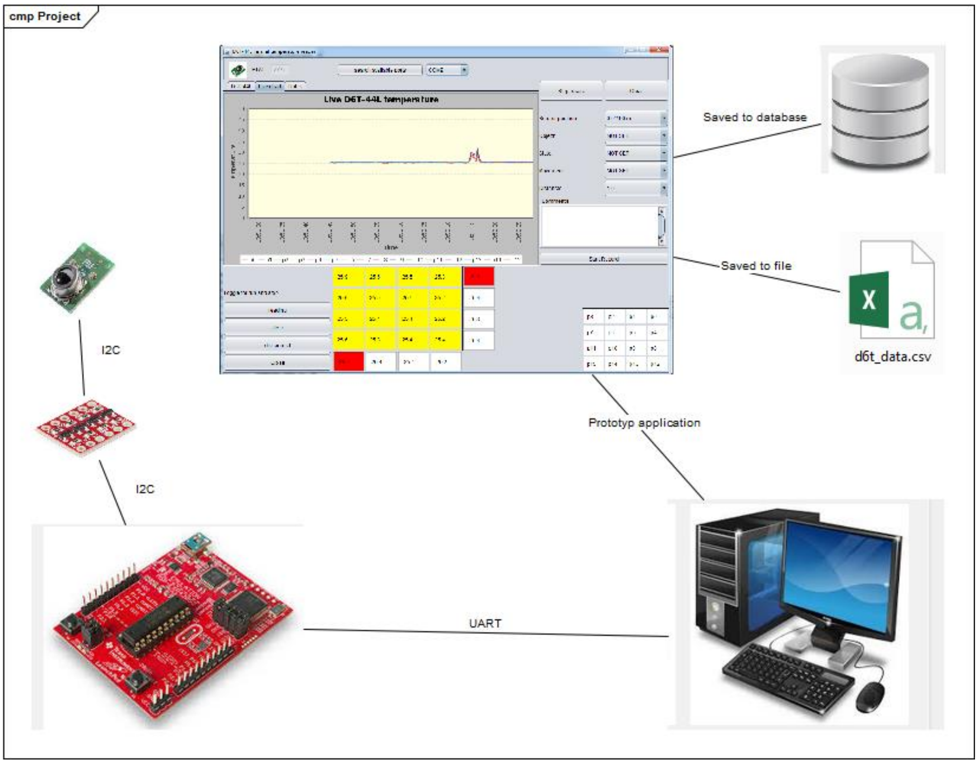


Figure 5 Project overview.

When the microcontroller reads the sensor, it quality checks the data using CRC [31], [32] method. On the microcontroller a new protocol is made for the transfer of temperature data to the computer. New start bits, a sensor ID and new calculated PEC are added. An application on the computer reads the data and quality checks the transfer between the microcontroller and the computer using the same CRC methods. If everything is correct then the data is saved to the database and the file.

3.1. Assessment on the number of sensors.

To decide the number of sensors one has to consider what one wants to detect or what one wants to monitor. If one wants to detect temperature used for controlling of air-condition, installing one sensor in the ceiling may be enough. If one wants to have light control, one can install a sensor in the ceiling over sitting groups of people. If one wants to count how many people there are or perhaps just detect the direction of movement, mounting just one sensor in the ceiling may be enough. Or if one wants to detect if the person is in bed or leaving the bed, one sensor can be placed in the ceiling directly over the bed. It is also possible to detect the direction of movement. One example is if the sensor is placed on the wall in avertical direction the temperature will increase or decrease when moving away from or towards the sensor.

If one wants to cover the whole room one needs more than one sensor as the sensors FOV is around 45°. In *Figure 6* the sensor is installed in the ceiling and one can see that it will cover only a small part of the room. Putting 4 sensors together in a square 2x2, the sensors reading angle will be 90° in both directions but it will still not cover the whole room. If one want to mount the sensors in the ceiling **and** cover the whole room, one will need 4 sensors side by side, in each direction, to achieve 180° coverage, which will give a total of 16 sensors. Another solution is to place 4 sensors in one corner, as all directions in the corner are 90°, one will need only 4 sensors to cover the whole room. Modelling will be done in Google SketchUp using 4 sensors located in one corner of the ceiling.

Google SketchUp is used for modelling. Tests of sensor are set up after these models.

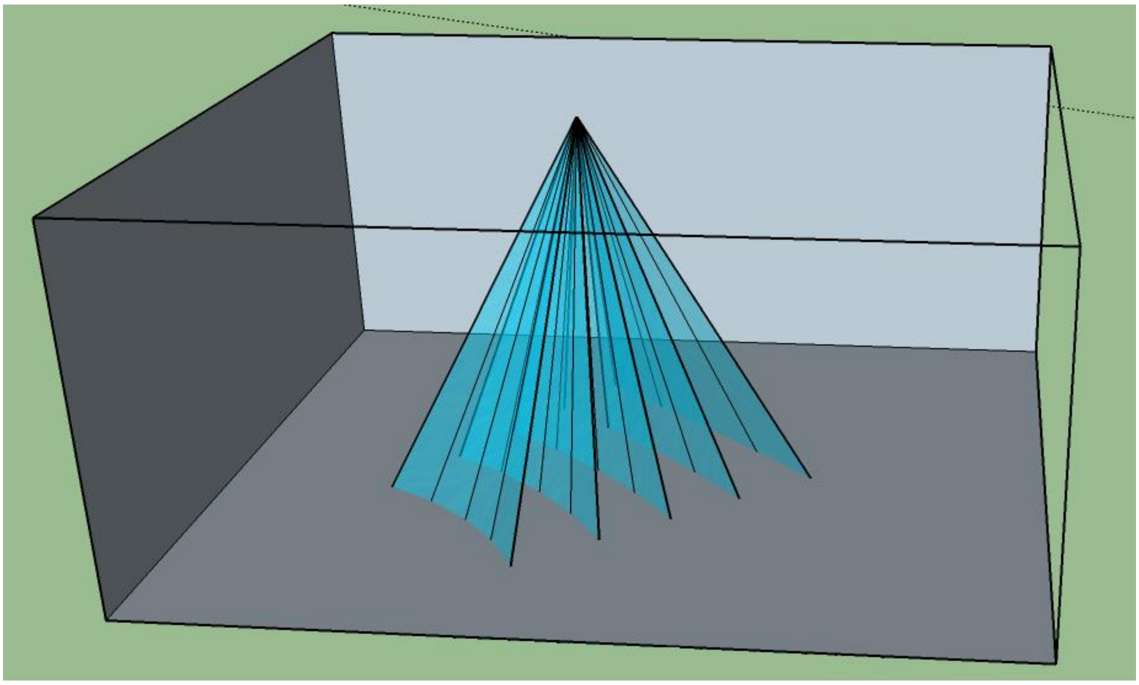


Figure 6 D6T-44L sensor installed in ceiling

3.2. Programming languages and IDE

Programming languages and IDE usage:

- **C** Code on microcontroller is written in C, CCS Code Composer Studio 6.0.1
- **Java** Prototype application written in Java, JDK 32-bit and NetBeans 32-bit
- **SQL** Database, Apache Derby DB SQL (standard ANSI SQL)
- **Modeling** Enterprise Architect and Google SketchUp
- **Verify I^{2}C transmission** LogicPort logic analyzer
- **Verify UART transmission** RealTerm
- **Decision tree** Edraw Max 7.9 trail version

Database and text file

Text file for temperature data, format:

```
Timestamp    [P0, P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14, P15]
```

Example data (***d6t_data.csv***):

```
2015-05-05 21:54:09.619 [255, 257, 257, 262, 254, 257, 257, 258, 253, 256, 257, 256, 254, 255, 254, 256]
2015-05-05 21:54:09.786 [254, 256, 256, 260, 252, 255, 256, 257, 252, 255, 255, 255, 252, 254, 253, 255]
2015-05-05 21:54:09.889 [255, 256, 256, 261, 253, 256, 256, 258, 252, 255, 256, 256, 253, 255, 254, 256]
```

Database:

Database used is Java DB which is Oracle’s supported distribution of the Apache Derby open source database which is also included in JDK [33]. It supports standard ANSI/ISO SQL.

SMALLINT is preferred over INTEGER in tables to reduce the size of database, as only 2 bytes are needed to store the temperature.

3.3. Modelling in Google SketchUp

In *Figure 7* the room is 4 x 6 meters and the ceiling height is 2.4 meters and shown are 2 sensors in the left corner. “*Lower right sensor*” will cover ¼ part of the room and “*Upper right sensor*” will cover the next ¼ part of the room. This will cover only half of the room. *Figure 8* shows example of 4 sensors covering the whole room.

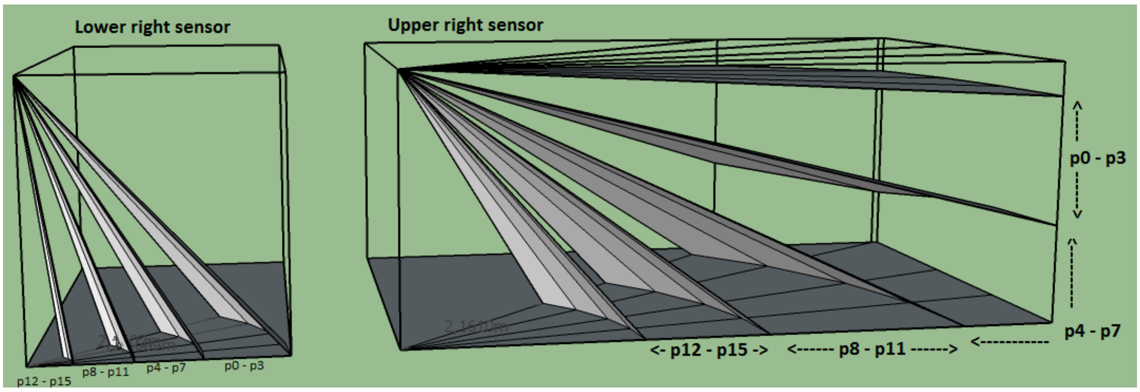


Figure 7 Lower right sensor and upper right sensor.

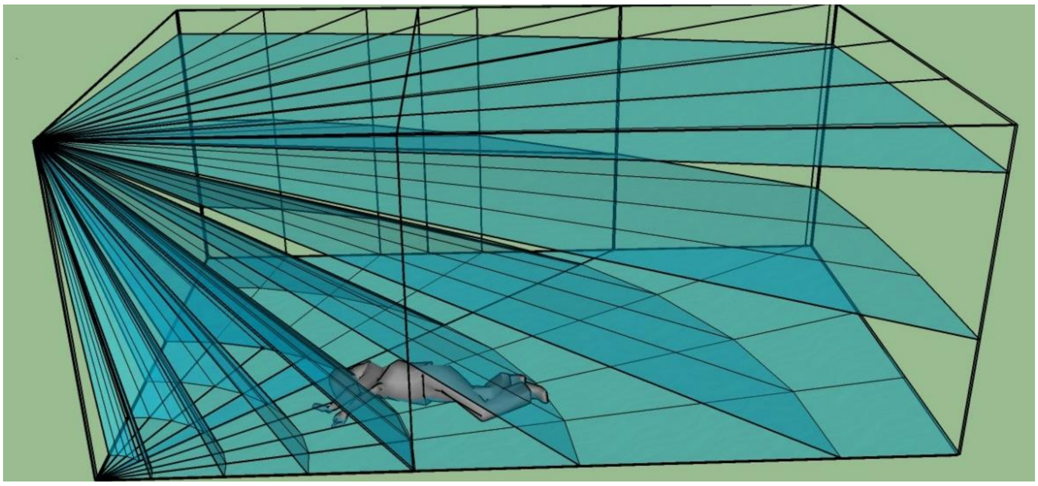


Figure 8 Four sensors covering one room.

If there is no need for surveillance right up to the corner of the room one can place two sensors side by side as in *Figure 9*. This solution will give a 2.16 m “blind spot” on the floor that is not covered by the sensors

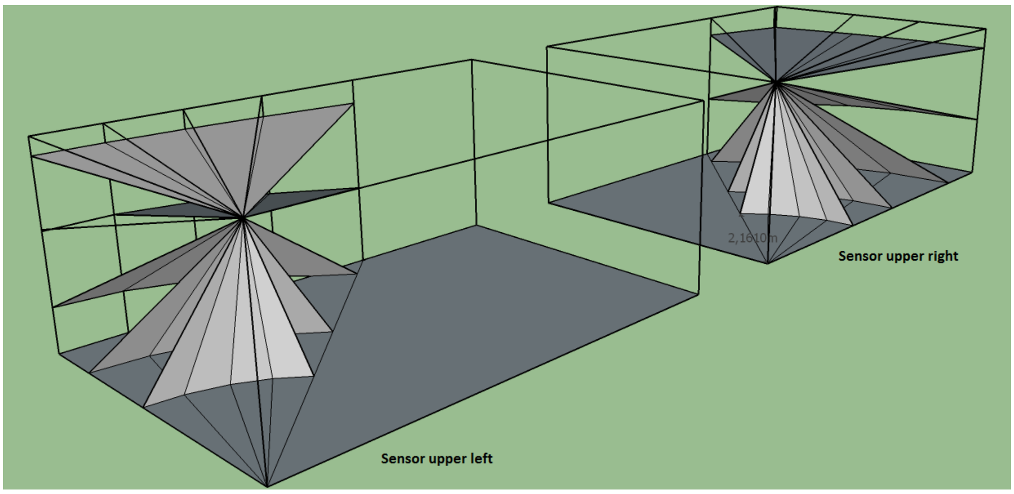


Figure 9 Sensor upper left and sensor upper right.

In these tests, the decision was made to let the upper part of *p0-p3* cover up to a maximum of 2 meters from the floor, as there was no need to detect anything over 2 meters above the floor. The angle between the ceiling and upper border of *p0-p3* is 3.81°, which is the angle that will let the sensor cover up to 2 meters above the floor when the height of the room is 2.4 meter and the length is 6 meter. This angle depends on the size of the room.

From 2D to 3D

One advantage of using modelling it that one can find new characteristics. As each pixel in the sensor can be seen as one detection zone (*Figure 10*), there will be 16 different detection zones each covering its own area. And each detection zone will be projected onto its own area on the floor (*Figure 11*), and now it become 3D as each detection zone will have *x-direction* and *y-direction* on the floor and *z-direction* in the height. They will also have one extra variable which is the temperature. The closer the human is to the sensor the higher the temperature will be.

Mathematically one should be able to calculate the distance from the sensor to a person, but then one needs more tests regarding the human surface temperature. To calculate the distance one can use Polar and Cartesian coordinate systems. As a result the temperature will indicate where in this detection zone the sensor detects the human.

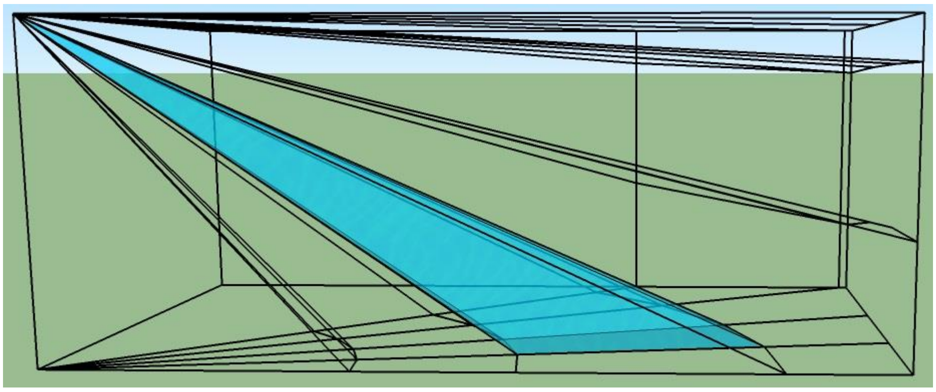


Figure 10 One pixel zone of sensor.

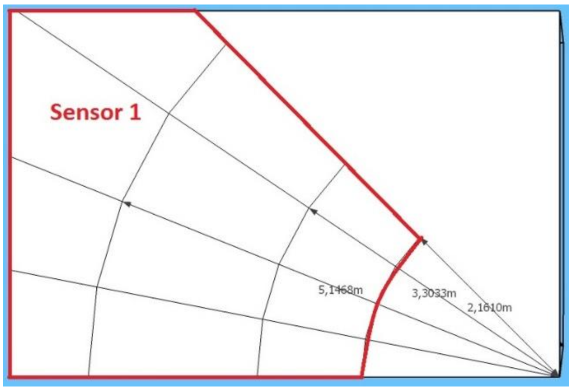


Figure 11 Sensor zones projected to floor.

Rooms with different sizes and persons.

In Microsoft Word rooms were modelled in different sizes, from 2 meter to 7 meter and placed one person standing in it (*Figure 12)*. Then a table (*Table 3*) was made and marked by what pixel the person is detected in (*see full test in Appendix 3 – Model of person in different rooms*). Common for all these rooms is that the person is detected in the same pixel area despite the size of the room.

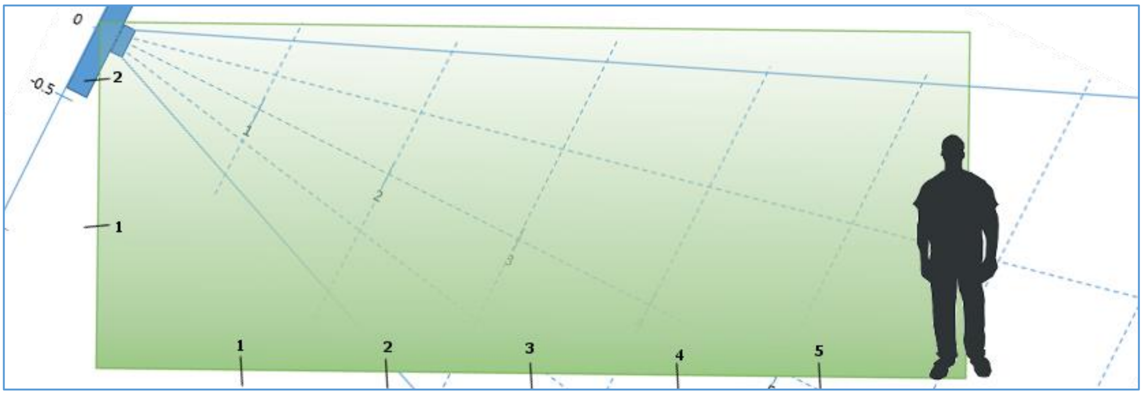


Figure 12 Room at 6 meter and a standing person.

Table 3 Detection of human in the sensors pixel according to distance.

Distance in m	**1**	**2**	**3**	**4**	**5**	**6**
p0-p3				x	x	x
p4-p7		x	x	x	x	x
p8-p11		x	x	x		
p12-p15	x	x				

3.4. Overview of data processing

3.4.1. Calculating Field of View reading area

The D6T-44L FOV is 4 x 4 pixels in x and y direction [27]. *x-direction* has an angle of 44.2° and *y-direction* 45.7°. Each pixel will have *x-direction* of 11.05° and *y-direction* of 11.425° (*Figure 13*). Reading angle and distance from sensor is illustrated in *Figure 14*, and *Table 4* shows the length and width in meters from sensor. For example, if the sensor is placed in the ceiling at 3 meters above the floor facing down, the *x-direction* will be 2.31 meters and *y-direction* will be 2.39 meters which will cover 5.52 m^2 of the floor. If the sensor is placed at 2 meters above the floor the *x-direction* is 1.54 meters and *y-direction* is 1.6 meters and it covers 2.4 m^2 of the floor. One can also read that in a distance at 5 meters from the sensor the detection area for each pixel is 0.96 x 1.0 meter, close to 1 m^2.

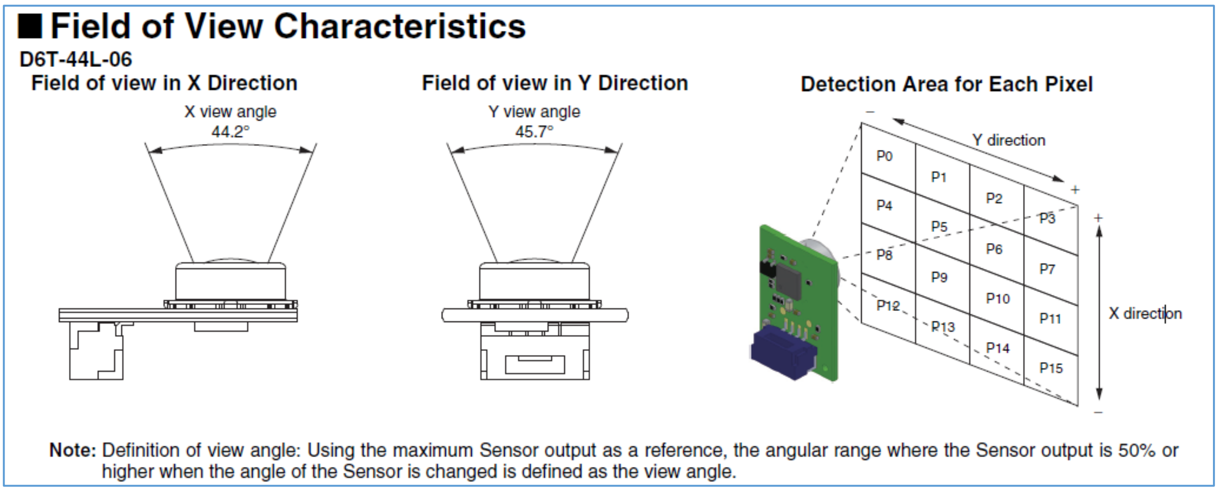


Figure 13 Field of View Characteristics [27].

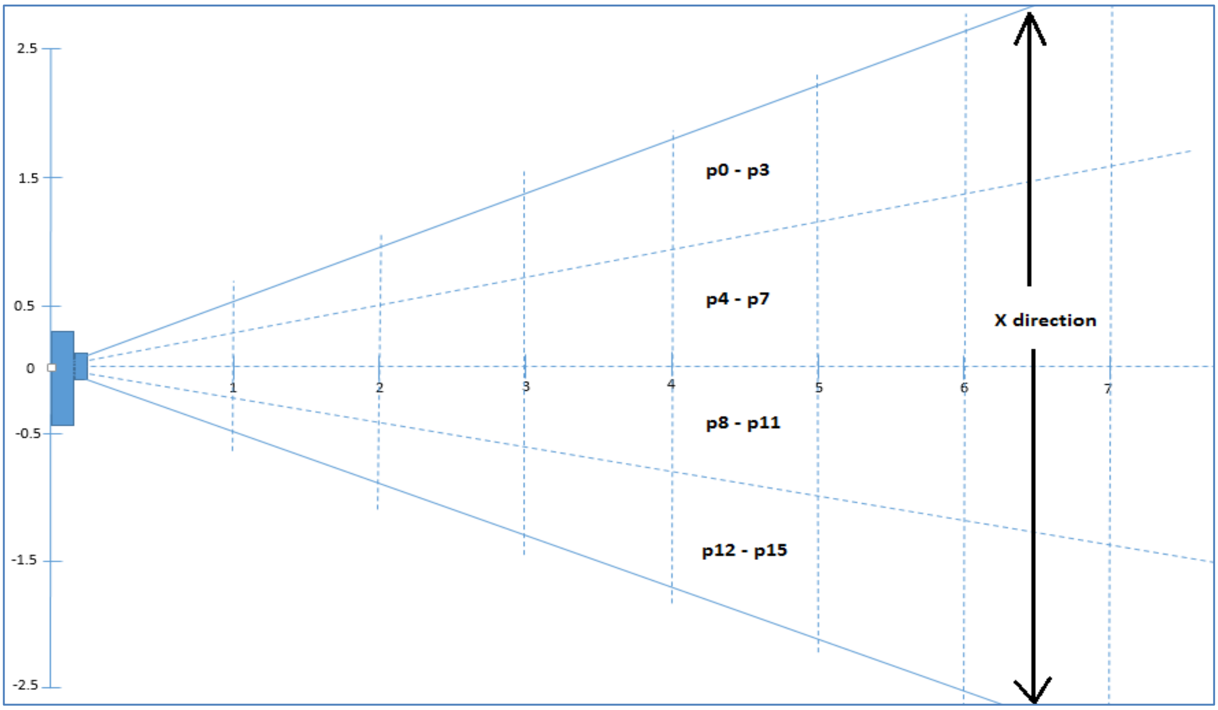


Figure 14 Reading area in distance from sensor.

Table 4 Reading area x- and y- direction, vertical, no angle.

Distance from sensor	**1 m**	**2 m**	**3 m**	**4 m**	**5 m**	**6 m**	**7 m**
X area in m	0,77	1,54	2,31	3,09	3,86	4,63	5,40
Size of each X pixel in m	0,19	0,39	0,58	0,77	0,96	1,16	1,35
Y area in m	0,80	1,60	2,39	3,19	3,99	4,79	5,58
Size of each Y pixel in m	0,20	0,40	0,60	0,80	1,00	1,20	1,40

3.4.2. Variables from the system and distance detection

The variables we get from the sensor are surface temperature and pixel number. The microcontroller is adding the sensor ID, and the software on the computer is adding the timestamp. In summary these are the variables we get from the system:

- Surface temperature
- Pixel number

- Sensor id
- Timestamp

Can the system detect distance? If the sensor is mounted in a corner of a room in the ceiling, the angle of FOV can be used to estimate some distance. If there is detection in pixel nr 12, 13, 14 or 15 this has to be on the floor, and this distance can be calculated using trigonometry. If there is detection in pixel nr 8, 9, 10 or 11 and at the same time no detection in pixel nr 12, 13, 14 or 15, this tell us that this distance is further from the sensor and can also be calculated using trigonometry. Human temperature can also be used to calculate distance. Based on what pixel or the combination of pixels the sensor detects, distance can be estimated as a function of temperature, but it is not precise. For example, if one person is detected in pixels nr 8, 9, 10 and/or 11 and not in pixels nr 12, 13, 14 and/or 15 he/she can still be in distance for pixels nr 12, 13, 14 and/or 15 because he/she can be lying on the sofa, see possibility in *Figure 12*.

3.4.3. Sensor position in these tests

In these tests only one D6T-44L sensor is placed in the upper corner as in *Figure 9*, and it will not cover the whole room.

3.4.4. Dataflow, reading data from D6T-44L and sending over UART

One full temperature data array from D6T-44L contains 35 bytes. Each temperature data is 16-bit width, signed 10 times value of Celsius degree. Example 25.8 °C is 258 and hex value is 0102h and 30.0 °C is 300 and hex value is 012Ch. 2 bytes are PTAT, 32 bytes are temperatures and 1 byte is PEC.

- PTAT - 2 bytes PTAT(Lo) and PTAT(Hi) temperature inside sensor module
- (P0 – P15) - 2 bytes of each temperature P0(Lo) P0(Hi), …, P15(Lo) P15(Hi)
- PEC - Packet error check code.

Before sending over UART from the microcontroller the sequence of temperatures is extended with 2 new start bytes and one byte for Sensor ID, then a new PEC byte is calculated and this byte stream is sent over UART to PC. A sensor ID is needed for the future installations containing several sensors. The choice was to use the alphabet letters as the ID, starting with letter A, because these values will normally be higher than the measured temperature.

To send temperature data from the microcontroller to the server or PC a new protocol had to be made. Because the temperature used 2 byte therefore the decision is to use 2 start bytes to distinguish it from the temperature data. The value of these 2 start bytes are both bigger than temperature data, hex value of alphabet letter are used. One can consider using TT as in TTU but two "T"s together could easily be perceived as an error, and due to the lack of other suggestions, the choice was to use the authors letters CN, but they can easily be changed. If several of these sensors will be used, one will need a *I2C bus switch* [29] because these sensors cannot be programmed with another *slave address* (*Appendix 1 – OMRON FAQ*). It is suggested to use the alphabet and start at letter "A", the next ID is letter "B" and so on.

The new protocol has 39 bytes:

```
start bytes (2)    sensor id (1)    length of data in bytes (1)   data (34)       crc checksum(1)
```

3.4.5. MSP430g2553 Timer, I2C, UART - Interrupt flow

The code on the microcontroller has several forms of interrupts that have to be handled: Timer interrupt, I^{2}C and UART. These interrupts are nested interrupts and can be handled by turning on and off the interrupt or using flags [34]. This choice fell on a combination of turn on/off and flags that gave no error when validated by the flow with the Logic Analyzer. The flow is shown in *Figure 15*. It starts in the main while-loop that disable interrupt. The next is to call the methods that read from sensor and end with call the method to send over UART. The flag is set when ready to start UART. When finished sending UART the flag is cleared, exit to while-loop and while-loop ends by going to LPM3. The timer is interrupted 4 times each second and if UART-flag is cleared the timer exits LPM3 and goes to main.

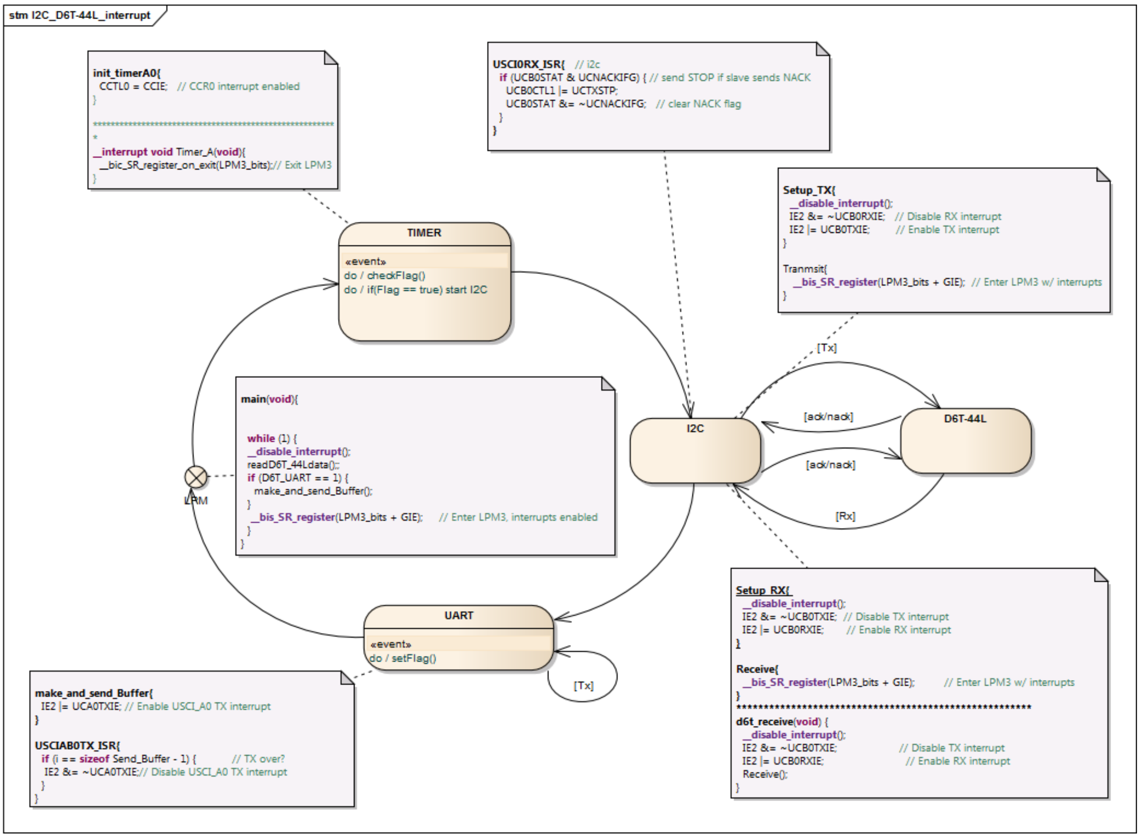


Figure 15 MSP430g2553 Interrupts, Timer, I2C, UART flow.

3.4.6. I2C calculations for robust communications

For reliable robust communications through I2C bus the pull-up resistors must be correctly sized up [35]. Selecting the correct resistor helps to avoid wasting power, erroneous bus conditions or transmission errors caused by noise, temperature or operating voltages. A strong pull-up will prevent the line to be sufficiently low.

Factors to calculate the pull-up resistor value R_P:

- V_{DD} - Supply voltage 3.3 V and 5 V
- C_{BUS} - Total bus capacitance 100 pF
- I_{IH} - Total high level input current

Clock frequency of 100 kHz

- V_{IH} 0.7 % VDD - Logical high and will read as 1.
- V_{IL} 0.3 % VDD - Logical low and will read as 0.
- R_P pull-up resistor

- R_S serial resistor
- I_{OL} low-level output current I2C specification = 3
- V_{OL} low-level output voltage I2C specification = 0.4
- T_R Clock/Data Rise Time I2C specification = 1000 ns for Clock speed 100 kHz

Minimum pull-up resistance:

$$Rp\ (3.3\ V)\ \geq\ \frac{Vdd - V_{OL}}{I_{OL}} = \frac{3.3V - 0.1\ V}{3} \geq\ 1.067\ K\Omega \qquad (1)$$

$$Rp\ (5\ V)\ \geq\ \frac{Vdd - V_{OL}}{I_{OL}} = \frac{5\ V - 0.1\ V}{3} \geq\ 1.633\ K\Omega \qquad (2)$$

The minimum pull-up resistance is 1.067 kΩ for 3.3V and 1.633 kΩ for 5V.

It remains to calculate the maximum pull-up resistance as it was problem to find T1 and T2.

3.5. Prototype of Java Swing application

From autumn 2013 the course “Advanced Course in Programming” teaches one how to transfer and read bytes over socket between client computer and server using both Java and C/C++. This course used NetBeans for visual client application. The author also has some experience in programming Java Swing applications, using database and JFreeChart. Therefore the choice of Java and NetBeans are used as tools to make the prototype application for testing.

3.5.1. Java setup for communicate with MSP340g2553 using Java Serial port

Java Communication 3.0 API is not available for Windows 64-bit yet, so Java Communication for 32-bit must be used, which means that NetBeans 32-bit and JDK 32-bit must be used. *Table 5* points out the configuration of java and on the microcontroller just to control that it has the same settings, baud rate 9600, 8 bit, 1 stop bit and no parity bit. The same settings as used in RealTerm terminal when reading UART data from the microcontroller.

Table 5 Port configuration on microcontroller and in java.

MSP430g2553 - UART

```
  DCOCTL = 0;              // Select lowest DCOx and MODx settings
  BCSCTL1 = CALBC1_1MHZ;   // Set DCO
  DCOCTL = CALDCO_1MHZ;
  P1SEL = BIT1 + BIT2 ;           // P1.1 = RXD, P1.2=TXD
  P1SEL2 = BIT1 + BIT2 ;   // P1.1 = RXD, P1.2=TXD
  UCA0CTL1 |= UCSSEL_2;           // SMCLK
  UCA0BR0 = 104;                  // 1MHz 9600
  UCA0BR1 = 0;                    // 1MHz 9600
  UCA0MCTL = UCBRS0;              // Modulation UCBRSx = 1
  UCA0CTL1 &= ~UCSWRST;           // **Initialize USCI state machine**
  IE2 |= UCA0RXIE;         // Enable USCI_A0 RX interrupt
  P1SEL |= BIT6 + BIT7;           // Assign I2C pins to USCI_B0
  P1SEL2|= BIT6 + BIT7;           // Assign I2C pins to USCI_B0
```

Java Serial Port

```
serialPort = (SerialPort) portId.open("D6T_reader", 2000);
// D6T_reader will be the owner of the port
// 2000 is timeout in milliseconds
serialPort.setSerialPortParams(9600,
                                 SerialPort.DATABITS_8,
                                 serialPort.STOPBITS_1,
                                 serialPort.PARITY_NONE)
DEFAULT: 9600 baud, 8 data bits, 1 stop bit, no parity
```

3.5.2. JFreeChart

JFreeChart is a free Java library for displaying professional quality charts in applications. It supports 2D and 3D charts [36]. To easily set up JFreeChart see "*Tutorial Points*" homepage [37]. In this Java Swing application there is the creation of dynamic chart with Time Series showing temperature in real time for the last 60 seconds, a chart for drawing the room according to where the sensor is placed, and a graph to show temperature fluctuations.

3.5.3. Swing Application

Swing application is developed using agile methods. All functionality was implemented as the needs arose. For example, the first need was an application which could read serial port. Hence the first part of this application was a simple connection and printed a log of the read data into a text area. Then next need was to visualize the temperature data live, then JFreeChart charts were added, and so on. Finally the test for this sensor began and one discovered that much manual work was needed that require a lot of time. Therefore the application was changed to do this work, and now this application takes screenshots, toggles temperatures and pixel numbers, visualizes the room, and makes graphs (see *Appendix 2 – Functionality of D6T-44L Thermal Temperature Reader).*

3.6. Human in sensor zones

To detect distance in this sensors FOV every pixel zone is divided according to distance (*Figure 16*). They are named C1, C2, C3, C4, D2, D3, D4, E3 and E4. Thus reading "**in C4, D4 and E4**" one knows that this is the *p0-p3* Omron's y-direction FOV, and "C3, D3 and E3" are in *p4-p7* y-direction FOV, and so on.

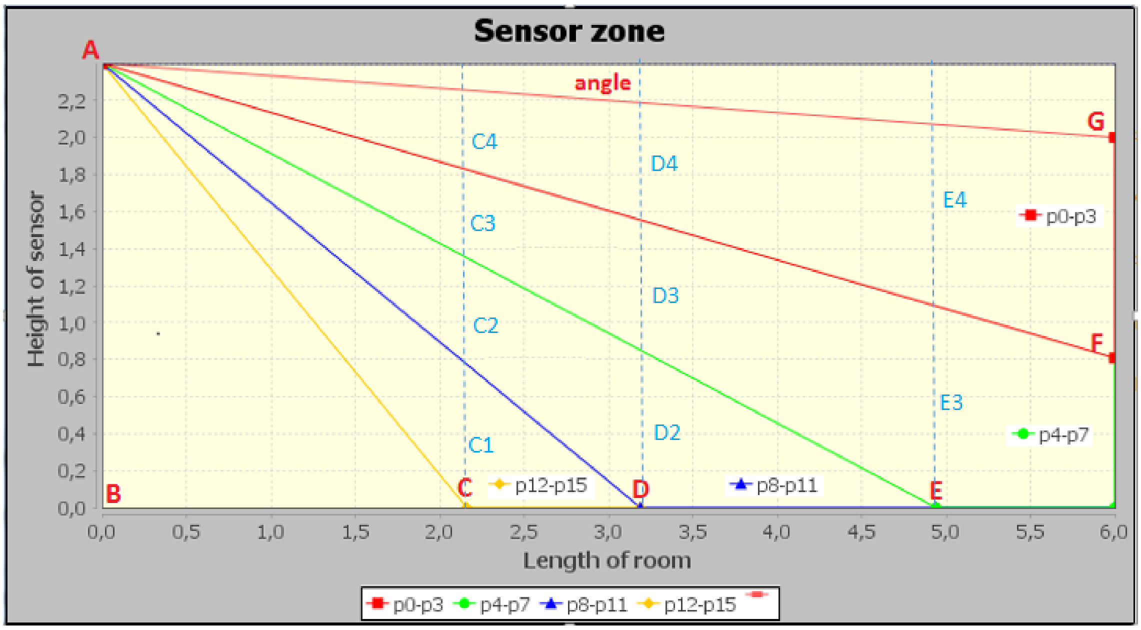


Figure 16 Sensor zones named C, D and E.

3.6.1. Single cell based detection

One clarification: it is impossible to detect 1 human in 1 pixel with a coverage area which is less than the human body area. Table 4 is used to calculate circa square meters for each pixel. (*To calculate more specific distance and area of FOV one needs to use Polar and Cartesian coordinate systems*.) Human location can be described by the distance from the sensor over the floor and the height from floor. The sensor's position is projected down to the floor. This is a simplified calculation based on the size of each pixel. Table 6 will analyze if one can detect a person in 1 pixel in C4 without detecting the same person in

C1-3 and so on. (Calculations will be uploaded to git: filename: calculationOfPixelArea.xlsx)

Table 6 Single cell based human detection.

Estimated distance from sensor in meter on the floor		**Height from floor**	**Circa square**	**Possible situation**	**Detected**
Person in C4	2.16-3.2	> 1.8 meter	0.18-0.41	P ≈ 0, Impossible	p0-p3
Person in C3	2.16-3.2	> 1.0 meter	0.20-0.45	P ≈ 0, Impossible	p4-p7
Person in C2	2.16-3.2	> 0.8 meter	0.25-0.55	P ≈ 0, Impossible	p8-p11
Person in C1	2.16-3.2	> floor	0.33-0.0	P ≈ 0, Impossible	p12-p15
Person in D4	3.2-5.0	> 1.2 meter	0.41-1.00	Possible, but unlikely	p0-p3
Person in D3	3.2-5.0	0.5 meter	0.45-1.11	Possible, Maybe sitting or lying on a sofa	p4-p7
Person in D2	3.2-5.0	floor	0.55-0.0	Possible, but unlikely	p8-p11
Person in E4	5.0-6.0	> 0.9 meter	1.00-1.44	Possible, standing on table	p0-p3
Person in E3	5.0-6.0	floor	1.11-1.01	Possible lying/sitting	P4-p7

3.6.2. Multi-cell based detection

In *Table 7* the question posed is: Is it possible to be in C1 (or C2, or C3 and so on) and in vertical direction have detection in 1 pixel, 2 pixels, 3 pixels or 4 pixels at the same time?

Table 7 Multi-cell based human detection.

Can one detect	**1 pixel**	**2 pixel**	**3 pixel**	**4 pixel**
person in C4	Hard to believe	Np[1]	Np	Np
person in C3	Yes if lying on a table	Yes if sitting on a table	Np	Np

[1] Np – not possible

Can one detect	1 pixel	2 pixel	3 pixel	4 pixel
person in C 2	Yes if lying on a table	Yes if sitting on a table	Yes if standing on a table	Np
person in C 1	Yes	Yes	Yes	Possible, but rarely
person in D4	Hard to believe	Np	Np	Np
person in D 3	Yes if sitting on a chair/sofa	Yes if sitting/standing on a chair/sofa	Np	Np
person in D 2	Yes	Yes	Yes	Np
person in E4	Yes if standing on a table	Np	Np	Np
person in E 3	Yes	Yes	Np	Np

3.7. Calibration

Static calibration is defined as temperature that is constant or slowly varying. Dynamic calibration is when the temperature varies significantly over a certain time or range. I.e., a human enters the room and the temperature may be rising, or dropping if the human has been outside.

3.7.1. Maximum and minimum temperature

DigInfo TV [38] has a video where the sensor was used to check for human presence and hot food. It may be perceived that it is possible to detect temperature out of range as informed in the "*object temperature detection range*" from datasheet [27] which is 0 to 50 °C.

In *Table 8* one can read that maximum and minimum human temperature the sensor read in a distance 1 – 6 meters varying from 25-30 °C.

Table 8 Reading of human body temperatures on different distance.

Distance	Temperature	Comments
10 cm	35 °C	very close to sensor
1 meter	not over 30 °C	in direct distance
1-2 meter	28 - 29 °C	average room temperature ca 25 °C
5-6 meter	around 25 °C	average room temperature ca 25 °C

3.7.2. Room

To calibrate a room, (a static calibration) one must find out what equipment can have different temperatures, either higher or lower. Examples of equipment that can have a higher temperature is a TV, ceiling lamp, wall lamp, windows with direct sunlight, heaters, a robotic cleaner among others not mentioned here. Examples of equipment with temperature that can be lower can be a refrigerator and freezer, as well as open windows and open doors.

To get one static calibration of a room with a TV, one needs one snapshot of the room with the TV on and one snapshot of the room with the TV off, that is a total of 2 snapshots. To calibrate the room with a ceiling lamp, one needs 2 more snapshots, one with the ceiling lamp on and one with the ceiling lamp off. To calibrate the room with a TV and a ceiling lamp one needs 4 different vectors. And if one have 1 more piece of equipment with an on/off one need 2 more snapshots. To get all the possible combinations of these 3 equipments one need 2^3 which is 8 different vectors. For each piece of extra equipment with 2 different states, you count it in power of 2. For example, if you have 1 TV, 1 door, 1 ceiling lamp, 2 wall lamps, 1 heater you will have $2^6 = 64$ different vectors just to cover one room with 6 different pieces of equipment. And at the end, if the person is refurnishing the room the calibration would have to be done all over again.

To calibrate a dynamic situation, you need to use the time and you also need to have a history of the previous readings. One needs a time counter and a timer limit that should give an alarm if things happen at times when normally there is no activity, for example in the middle of night. But of course, one person may wake up in the middle of the night and needs to go to toilet, and that should not produce an alarm. To calibrate a dynamic situation to detect what is normal activity should be handled jointly by a server and the user of the system.

To be able to test the sensor, calibration of the room was done when the room was empty. One snapshot taken at a moment in time was used as the default value.

The room was without any human and the temperature fluctuated with plus/minus 0.2 to 0.4 Celsius degrees. The temperature under the ceiling was a little bit higher than on the floor.

Under testing one saw that in small room (corridor) the temperature rose and everything around the human became the same temperature. It was impossible to distinguish between a human, the walls and the floor, and all pixels had the same value.

Also more than 1 human present in a room will increase the temperature in the room. The floor in the room kept the temperature of the person for some minutes before the temperature leveled out.

3.8. Classifying

Based on temperatures measured, experimental results and modelling (*Figure 17*), the following classifications are:

- **Object – static:** Object permanently installed in the room
- **Human – dynamic:** Object not permanently installed in the room, can have both static and dynamic behaviours.
- **Time:** The reading frequency of the sensor, maximum reading frequency is up to 4 times each second.

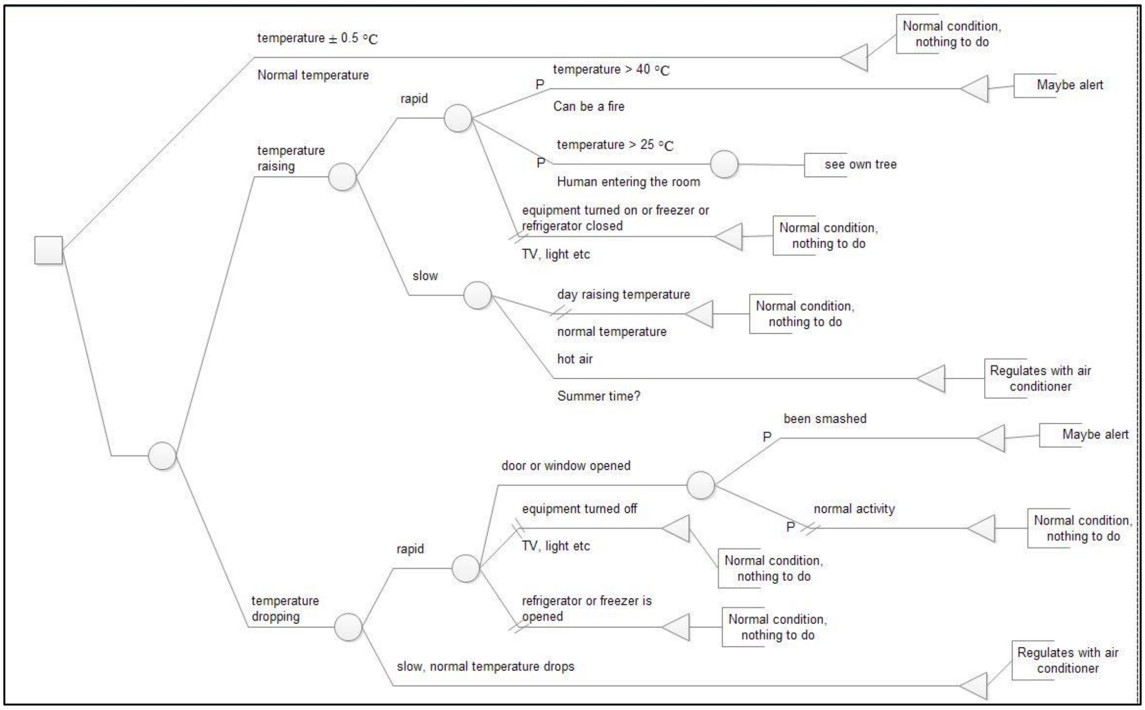


Figure 17 Decision tree for temperature changes.

Class Object (*Figure 18*):

Object has permanent position, and is not moving.

- **Ambient** Temperature fluctuate ± 0.5°C
- **Colder** A sudden drop of temperature can be caused by open or smashed window/door.
- **Warmer** A sudden raise of temperature can be caused by turning on a TV, a light, a heater or by a fire in the room or fireplace.

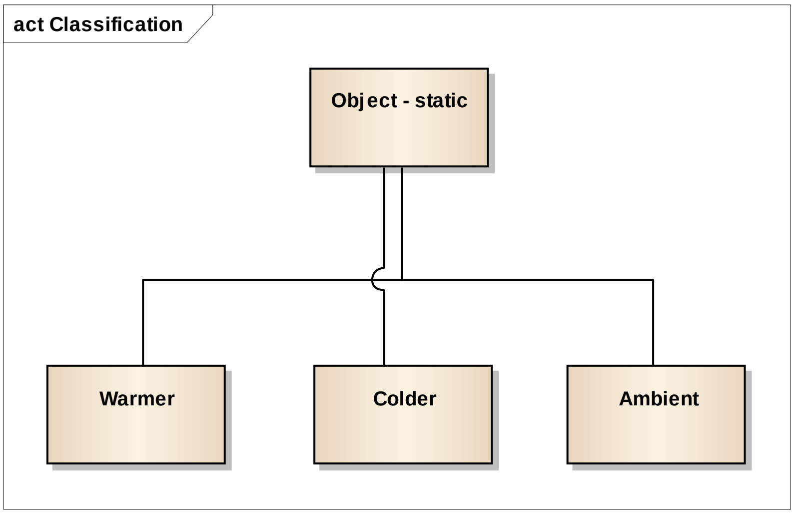


Figure 18 Classifying – Object.

Class Human (*Figure 19*):

Measured temperature for a human in an indoor situation with a normal temperature between 25 and 29 °C for distances between 2.2 and 6 meter (*4.2.4*). The clothing worn by the person will have an influence on the surface body temperature. A human can also have a temperature that is less than measured here since if the person has been outside in minus or lower temperature he or she will probably have low surface body temperature for a while.

- **Moving** – dynamic behavior. Common for movement is detection in 2 or 3 pixels in a row at the same time and changes in temperature between sensor readings.

- **Moving to right** – detection is in rows, 2 or 3 rows at the same time, never in only 1 pixel (of course if there are a person in wheelchair he/she can be detected in only one pixel). The detection will change to the adjacent pixel in the next sensor reading, in a right direction.
- **Moving to left** – the same holds as in the moving to the right, but this time in a left direction.
- **Moves from sensor** – detection in 2 or 3 pixels and the temperature will drop in the next sensor reading, or in a few sensor readings, it depends on the reading frequency of the sensor.
- **Moves towards the sensor** – the same as moving from the sensor, but this time the temperature will rise.
- **Falling** – sudden changes from a vertical pattern to a horizontal pattern, or from lying on the sofa to lying on the floor.

- **Not moving** – static behavior.
- **Standing** – vertical detection in 2 or 3 pixels in a row at the same time. The middle of the pixels has the highest temperature.
- **Sitting** – vertical detection in 1 or 2 pixels in a row at the same time. Temperature in the upper most pixel will normally have a higher temperature than a pixel in the lower most part (as most of the body area will be in the upper most pixel).
- **Lying** – horizontal or cross detection in 1, 2, 3 or 4 pixels at the same time. One must analyze the temperature for distinguishing between lying on the sofa or lying on the floor.
    - **Lying on the sofa** – to detect a person lying on the sofa there are 2 possibilities, there can be horizontal detection in a row or there can be detection crosswise.
    - **Lying on the floor** – same as in lying on the sofa plus if the detection is <u>only</u> in *p12-p15* this has to be lying on the floor (it is not possible to have detection in *p12-p15* and be sitting at the same time).

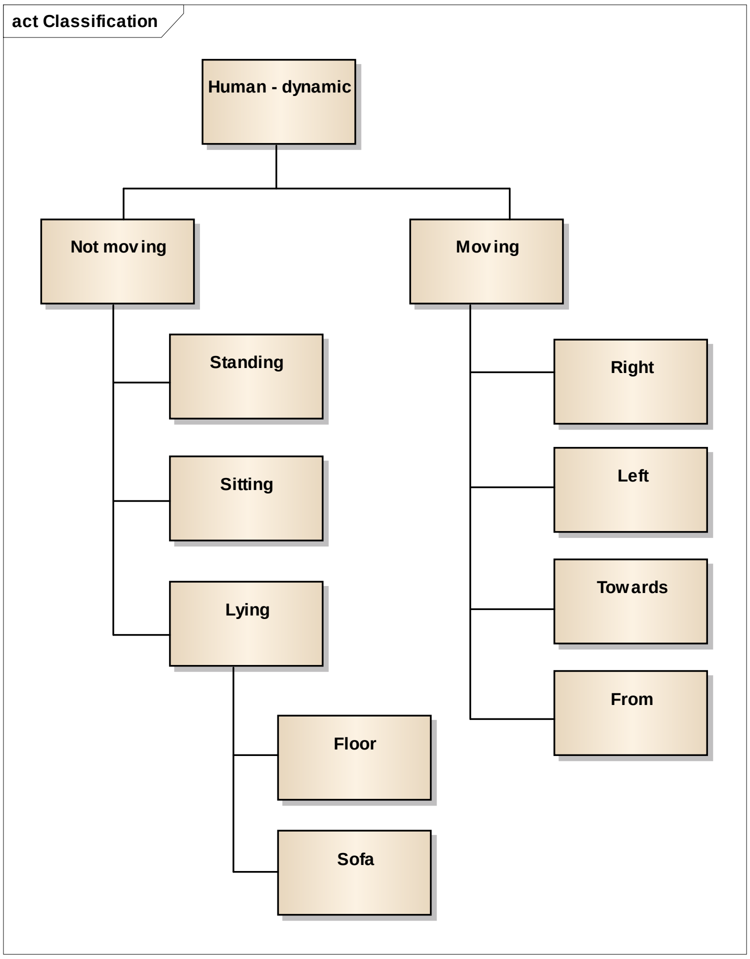


Figure 19 Classifying – Human.

Class time:

One has to use the time to decide if this is a dynamic or a static detection.

- **Slow changes** – small changes over time can be seen as normal.
- **Sudden changes** – sudden changes can be a human entering a room, it can also be a smashed window, fire or someone falling on the floor.

3.9. Proposed theory – algorithm

The goal is to examine whether it is possible to:

- Detect a person lying, standing or sitting.

- Avoid using data from the database or text file.
- Possible to run the algorithm on microcontroller
- Keep data in memory, and keep data small

The question is: Is it possible to use one temperature reading to figure out if there has been a human present?

Is it possible to use the average value of read temperatures to see where the person is? And how is it compared to the graph?

- **tP**[] – array of temperature readings, same as OMRON's naming convention
- **avgMea -** average of all measured temperatures
- **mea_avg**[] - *read value minus average of measured temperatures*

$$mea\_avg[i] = tP[i] - avgMea \qquad (3)$$

Let's take a look at one example reading: 25.9, 26.6, 26.6, 26.4, 25.7, 26.9, 26.7, 26.0, 26.3, **29.5**, 26.4, 26.1, 26.8, **28.0**, 26.6 and 26.7. The test person is detected in 2 pixels with values 29.5 and 28.0 (*pixels p9 and p13*). Provided temperature-set has an average temperature of 26.7 °C. While using *D6T-44L Thermal Temperature Reader*[2] one saw that the temperature in the room normally fluctuated with up to ± 0.5 °C. The theory is that if one subtracts the average temperature value from every temperature in the data packet from D6T-44L and in addition subtracts the fluctuation temperature at 0.5 °C then all the positive values will correspond to the human detected. A human surface temperature will normally be higher than the average temperature, so we only need to check the positive temperature values after subtraction.

Testing theory: *Figure 20* shows the 4x4 pixels square calibrated values (the little square), and the 4x4 pixels square real time values (the biggest square). The graph drawn shows the "*measured real time values* minus *calibrated values*" as series nr 5 (the tallest curve). *Figure 21*, a drawing graph shows using 2 different average values, the average value of calibrated snapshots and the average value of real time read data. **"Mea"** is real time

[2] Own developed prototype test application

measure values, **"Cal"** is calibrated snapshot values, **"Mea - Cal"** is then **"***real time measure values*" minus "*calibrated snapshot values*", **"avg-mea"** is "**Mea – avg(Mea)"** and **"avg-cal"** is "**Cal – avg(Cal)", column 6** without name is **"Measured – Calibrated".**

Comparing these graphs it seems that there are small differences between **"avg-mea"** and "**Mea - Cal"**. Comparing **"avg-cal"** and **"avg-mea"** there are opposite differences in the pixels with detections, pixel *p8* and *p12*.

Figure 20 Snapshot measured value

1	2	3	4	5	6
Pixel	**Calibrated**	**avg-cal**	**Measured**	**avg-mea**	
0	25,6	-0,2	25,6	-0,4	0
1	26,2	0,4	26,1	0,1	-0,1
2	26,3	0,5	26,2	0,2	-0,1
3	26,5	0,7	26,4	0,4	-0,1
4	25,4	-0,4	25,6	-0,4	0,2
5	26,1	0,3	25,9	-0,1	-0,2
6	26,1	0,3	26	0,0	-0,1
7	26,0	0,2	26	0,0	0
8	25,3	-0,5	26,8	0,8	1,5
9	25,6	-0,2	25,6	-0,4	0
10	25,8	0,0	25,7	-0,3	-0,1
11	25,8	0,0	25,7	-0,3	-0,1
12	25,6	-0,2	27,1	1,1	1,5
13	25,5	-0,3	25,6	-0,4	0,1
14	25,4	-0,4	25,4	-0,6	0
15	25,8	0,0	25,7	-0,3	-0,1
Sum	**413**		**415,4**		
Avg	**25,8**		**26,0**		

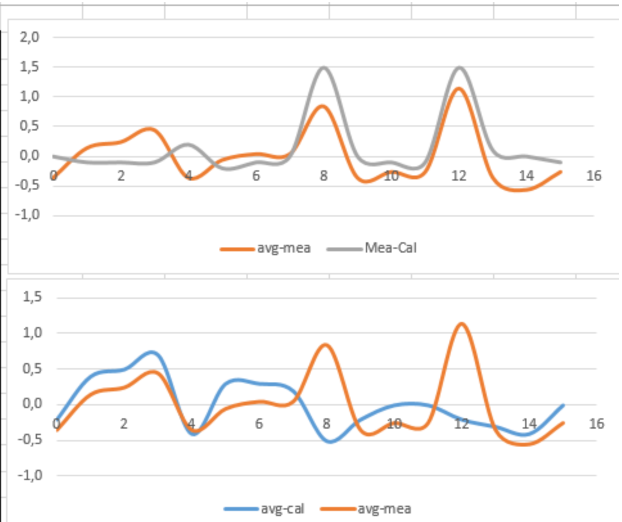


Figure 21 Snapshot measured value model.

The author suggests 2 algorithms restrictions: There is only one person and there is only a normal indoor temperature. Human temperature will never be lower than indoor temperature.

Algorithm 1, use 16 values (the calibrated array values):

```
Loop through tP[i]
For each value do
      temp = tP[i] temperature - calibrated[i]
      if (temp > 0.5)
            detection[i][j] = temp;
            j++;
end loop;
```

Algorithm 2, using 1 value (the measured average value):

```
Loop through tP[i]
For each value do
      temp = tP[i] temperature - avg(tP[i])
      if (temp > 0.5)
            detection[i][j] = temp;
            j++;
end loop;
```

Both algorithms will return the array with detection (`detection[i][j]`), pixel nr and temperature value.

Next step is to figure out if the person is standing, sitting or lying using a decision tree (*Figure 22*) and excel calculation (*calculationOfPixelArea.xlsx*). Probability is not calculated as a number. These results will also be used in the pattern recognition chapter (3.10).

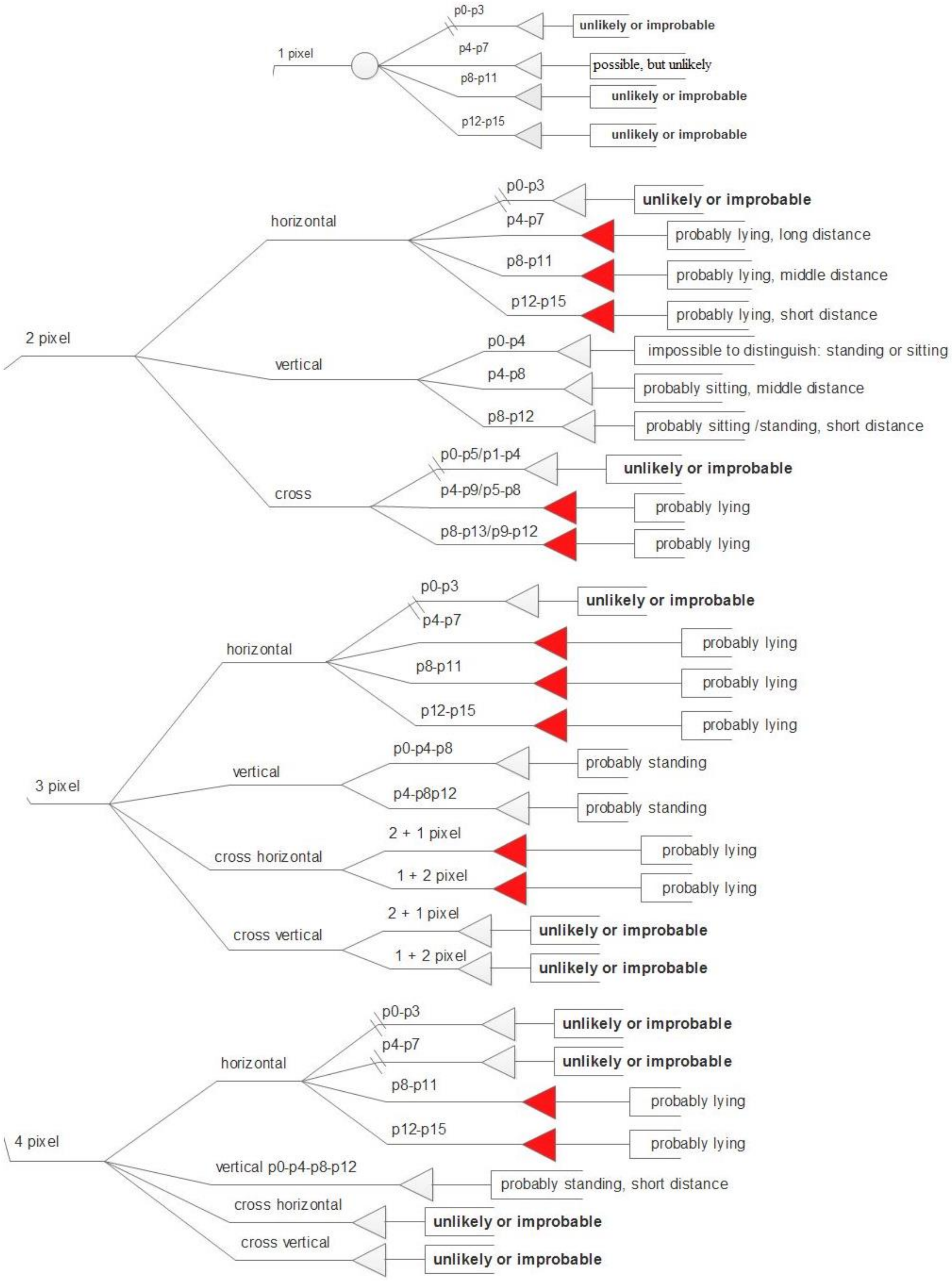


Figure 22 Decision tree 1 to 4 pixels

3.10. Proposed pattern recognition for standing, sitting or lying person

3.10.1. Static detection using document-term matrix

Many of the pattern recognition theories are mentioned in *Appendix 5 - Literature study.* Pattern recognition literature study is a huge area, several universities have their own courses just about pattern recognition (*one just has to google "Pattern recognition course."*) Many of these theories are about large amounts of data and in this master thesis the work is about a small 4 x 4 matrix and this is very little according to those theories. Another point is that this might have to run on microcontrollers and then it has to be "small" Further examination shows that one can use matrix multiplication as in [39] and [40] chapter 2.3 "Anvendelse: Binære matriser og søkevektorer" (*2.3 The use: Binary matrices and vectors search*) to solve pattern recognition.

Binary matrices and vectors search use general matrix multiplication [41]: Ax=b, normal matrix multiplication where A is the matrix with all the patterns, x is the search vector and b is the result vector. When using binary matrix and matrix multiplication the b vector will contain "*the number of*" '1's from the A matrix's rows. We look for the same "*number of '1's*" in b as there are '1's in x. If they match then we have one solution, however can be more than one solution in the b vector. The index in b will match with the index in A and this will give *one* solution. For example if row 15 in b has number 3 and there are 3 '1's in x, then we look up index 15 in A and this will be the solution.

Test of "*document-term matrix*" using binary matrices and vector search are chosen because here there are only two different situations, detection or not, 0 or 1 and therefore binary matrix will be perfect to use on microcontrollers. (Using binary matrix for detection is also smaller than using calibration array for rooms (3.7.2)). Patterns for a person in different situations was analyzed and gave a total of 48 different patterns. For simplification in this explanations only the pattern for a standing person is used, full pattern will be attached in excel file and it will be possible to add an extra pattern if necessary. Here are the 18 different patterns of a standing person:

Sn is situation –> rows in matrices, Tn is Term n –> columns in matrices.

Table 9 Matrix of standing patterns

S1	Person standing on floor in pixel 12, 8, 4,
S2	Person standing on floor in pixel 13, 9, 5
S3	Person standing on floor in pixel 14, 10, 6
S4	Person standing on floor in pixel 15, 11,7
S5	Person standing on floor in pixel 8, 4, 0
S6	Person standing on floor in pixel 9, 5, 1
S7	Person standing on floor in pixel 10,6, 2
S8	Person standing on floor in pixel 11, 7, 3
S9	Person standing on floor in pixel 4, 0
S10	Person standing on floor in pixel 5, 1
S11	Person standing on floor in pixel 6, 2
S12	Person standing on floor in pixel 7, 3
S13	Person standing on floor in pixel 12, 8,4,13, 9,5
S14	Person standing on floor in pixel 13, 9, 5, 14,10,6
S15	Person standing on floor in pixel 14, 10, 6, 15,11,7
S16	Person standing on floor in pixel 8, 4, 0,9,5,1
S17	Person standing on floor in pixel 9,10, 5,6, 1,2
S18	Person standing on floor in pixel 10,11, 6,7, 2,3

Table 10 The terms

T0	pixel 0
T1	pixel 1
T2	pixel 2
T3	pixel 3
T4	pixel 4
T5	pixel 5
T6	pixel 6
T7	pixel 7
T8	pixel 8
T9	pixel 9
T10	pixel 10
T11	pixel 11
T12	pixel 12
T13	pixel 13
T14	pixel 14
T15	pixel 15

This will give following matrix with patterns (*Table 11*) (to facilitate the reading 0 is omitted and blanks are basically 0):

Table 11 Binary matrices array for pattern recognition.

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
S1					1				1				1			
S2						1				1				1		
S3							1				1				1	
S4								1				1				1
S5	1				1				1							
S6		1				1				1						
S7			1				1				1					
S8				1				1				1				
S9	1				1											
S10		1				1										
S11			1				1									
S12				1				1								
S13					1	1			1	1			1	1		
S14						1	1			1	1			1	1	
S15							1	1			1	1			1	1
S16	1	1			1	1			1	1						
S17		1	1			1	1			1	1					
S18			1	1			1	1			1	1				

To find the pattern we also need a "search vector," this will be the pattern from the sensor/microcontroller. The algorithm from the previous chapter will give us one array with the pixel nr that provides detections (`detection[i][j]`). These pixels nr are the '1's in the search vector, the rest are '0's. To find the pattern the following example explains this using matrix multiplication: (S*n* is the different patterns (*in this example all patterns for standing person*), T*n* is pixel nr, x is column vector with detection from sensor (*this is the search vector*), b is the answer column.)

Table 12 Matrix multiplication with search vector example.

	A																	x		b
	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15				
S1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0		**1**		2
S2	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0		0		0
S3	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0		0		0
S4	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1		0		0
S5	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0		**1**		**3**
S6	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0		0		0
S7	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0		0		0
S8	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	*	0		0
S9	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0		**1**	=	2
S10	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0		0		0
S11	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0		0		0
S12	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0		0		0
S13	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0		0		2
S14	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0		0		0
S15	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1		0		0
S16	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0		0		**3**
S17	0	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0				0
S18	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	0				0

In the search vector (x) there are 3 bits with '1's. We then look in the 'b' answer column for the number of 3. This column tells us that we have 2 solutions with this patterns. And if we then look at both of these solutions we find out that only one of them is identical to our search vector and we have a solution. If we don't find the number of '3' in the 'b' column then there is no solution. It is important that the 'b' vector has the same number as "number of '1's" in the search vector. Here we have S5 as a unique solution and when we look up the "S" array we see that S5 tells us that this is "*Person standing on floor in pixel 8, 4, 0*".

To use matrix multiplication for binary matrices in coding one just AND the search vector 'x' with every row in pattern matrix (A) and count number of '1's. If the number of '1's (in 'b') correspond with the number of '1''s in 'x' then there will be a solution. We are looking for patterns that match the search pattern and if there is more than one, we need to do an addition comparison to find the unique one. There can be 2 or more patterns that correspond to our search vector 'x'. That can indicate that it is not possible to uniquely identify the situation. For example, with some detection is impossible to decide if it is a standing or sitting person, or lying or sitting person. To try to identify these situations one needs to do more tests and use the test results to identify if there is a sitting or a lying person based on the temperatures.

3.10.2. Dynamic detection using document-term matrix.

Analyses and modelling of states:

In *Figure 23,* sitting on floor and lying on floor have 2 common possible next stats: standing on the floor or ALERT. To distinguish between these 2 states, time needs to be a variable here, if after a certain time the state is unchanged then one has to give an ALERT.

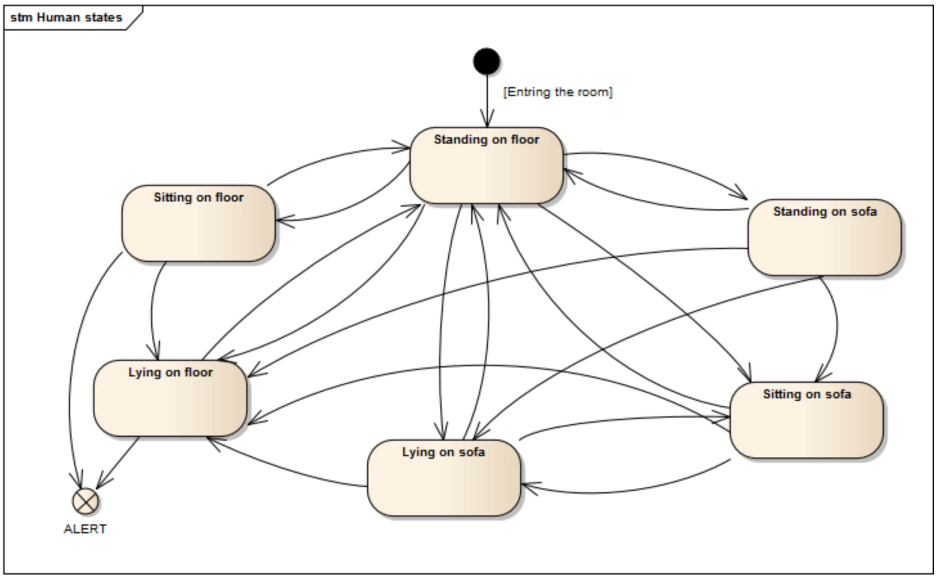


Figure 23 Analyses and modelling of human situations.

For analyzing/detection of dynamic states one can use the same "*document-term matrix*" with binary matrix theory as in previous chapter (3.10.1).

Table 13 Human positions.

D1	Standing on floor
D2	Standing on sofa
D3	Sitting on floor
D4	Sitting in sofa
D5	Lying on floor
D6	Lying in sofa

Table 14 Terms of states.

T1	Standing
T2	Sitting
T3	Lying
T4	Floor
T5	Sofa

Table 15 Human states matrix.

	T1	T2	T3	T4	T5
D1	1	0	0	1	0
D2	1	0	0	0	1
D3	0	1	0	1	0
D4	0	1	0	0	1
D5	0	0	1	1	0
D6	0	0	1	0	1

This time one need only one 8-bit variable and input states can be decoded from static detection and be coded using switch/case:

```
switch (search_vector) {
  case 0b00010010: // standing on floor;
  case 0b00010001: // Standing in sofa;
  case 0b00001010: // Sitting on floor, ALERT;
  case 0b00001001: // Sitting in sofa;
  case 0b00000110: // Lying on floor. ALERT;
  case 0b00000101: // Lying in sofa;
  default: throw new IllegalArgumentException();
}
```

3.11. Low power mode and cost

For managing power consumption on D6T-44L one should turn the power for the sensor on/off programmatically between each reading. The microcontroller MSP430g2553 can run on LPM3 as the timer runs on ACKL and power consumption can be reduced. In this example, the use of 2 CR2032 coin batteries in a series to give 5 V (in series they give 6V), D6T-44L needs 5V. Formulas used in this exercise are from lecture in IAY0330 Embedded Systems.

Table 16 Conversion and battery data.

1	s	=	1 000 000	us	Battery capacity mAh (CR2032)		Battery capacity µAh
1000	µA	=	1	mA	3 V	220	220000
					6 V	440	440000

Formulas (*es_05_PowerOptimization.pdf*):

$$\boldsymbol{Average\ current\ during\ event} = \frac{\begin{bmatrix} (State\ 1\ time)*(State\ 1\ current) + \\ (State\ 2\ time)*(State\ 2\ current) + \ldots \end{bmatrix}}{(Total\ awake\ time)} \quad (5)$$

$$\boldsymbol{Average\ current} = \frac{\begin{bmatrix} (\ Interval - Total\ awake\ time)*(Average\ sleep\ current)+ \\ (Total\ awake\ time)*(Average\ current\ during\ event) \end{bmatrix}}{(Interval)} \quad (6)$$

$$\boldsymbol{Expected\ battery\ life\ running\ continuously} = \frac{(Battery\ capacity)}{(Average\ current)} \quad (7)$$

Current consumption on D6T-44L: 5mA [27]

If using 2 CR2032 batteries in series D6T-44L will without any power management run for 88 hours: 5 mA load on batteries of 440 mAh and will run for 440 / 5 = 88 hours.

Using power management programmatically *(Appendix 1 – OMRON FAQ)* to turn power supply on and off between each reading, will change the power consumption, see *Table 17.*

Table 17 Power consumption for D6T-44L.

D6T-44L Power consumption			
D6T-44L	**time µs**	**current µA**	**Average current µA**
Every 2 sec	4,4736	5000	0,0224
Every sec	8,8173	5000	0,0441
Twice a sec	17,9192	5000	0,0896
4 times each sec	35,7484	5000	0,1787

Estimated power consumption on microcontroller:

MSP430G2553 Ultra-Low Power Consumption, active mode 230 µA at 1 MHz, 2.2 V, standby mode 0.5 µA, interval 1 second.

This is a very simplified calculation done the same way as in the subject IAY0330 Embedded Systems. The test is done using LogicPort and LEDs. One LED is turned on and off for reading I2C and another turned on and off for “make and send data” over UART, in total 2 LEDs are used. Waves in LogicPort are used to read the time each operation took. The test is done for comparing current usage based on how many times the temperature is read from a D6T-44L sensor (*once every two second, every second,* *twice a second and 4 times per second*). As *Table 18* and *Table 19* shows there are only small differences in current consumption when reading from the sensor, total awake time is small even if the microcontroller is reading 1 time or 4 times per second. Furthermore there are big differences whether using or not using low power mode.

Table 18 Microcontroller MSP430g2553 power consumption, CPU. LPM0, LPM3 - every 2 sec.

Every 2 sec	**state 1 - I2C**		**state 2 Read and send buffer - UART**		**state 3 Sleep**		**Total awake time**	**Average current**
	time µs	**current µA**	**time µs**	**current µA**	**time µs**	**current µA**	**time µs**	**µA**
LPM OFF	4,4736	230	22,1623	230	999973,36	230	26,6358	230,00
LPM0	4,4736	230	22,1623	230	999973,36	56	26,6358	56,00
LPM3	4,4736	230	22,1623	230	999973,36	0,5	26,6358	0,51

Table 19 Microcontroller MSP430g2553 power consumption, CPU. LPM0, LPM3 - every 1 sec.

Every 1 sec	**state 1 - I2C**		**state 2 Read and send buffer - UART**		**state 3 Sleep**		**Total awake time**	**Average current**
	time µs	**current µA**	**time µs**	**current µA**	**time µs**	**current µA**	**time µs**	**µA**
LPM OFF	8,8173	230	43,8045	230	999947,38	230	52,6218	230,00
LPM0	8,8173	230	43,8045	230	999947,38	56	52,6218	56,01
LPM3	8,8173	230	43,8045	230	999947,38	0,5	52,6218	0,51

Table 20 Microcontroller MSP430g2553 power consumption, CPU. LPM0, LPM3 - twice a sec.

Twice a sec	**state 1 - I2C**		**state 2 Read and send buffer - UART**		**state 3 Sleep**		**Total awake time**	**Average current**
	time µs	**current µA**	**time µs**	**current µA**	**time µs**	**current µA**	**time µs**	**µA**
LPM OFF	17,9192	230	87,5852	230	999894,5	230	105,5044	230,00
LPM0	17,9192	230	87,5852	230	999894,5	56	105,5044	56,02
LPM3	17,9192	230	87,5852	230	999894,5	0,5	105,5044	0,52

Table 21 Microcontroller MSP430g2553 power consumption, CPU. LPM0, LPM3 - 4 times per sec.

4 times pr sec	**state 1 - I2C**		**state 2 Read and send buffer - UART**		**state 3 Sleep**		**Total awake time**	**Average current**
	time µs	**current µA**	**time µs**	**current µA**	**time µs**	**current µA**	**time µs**	**µA**
LPM OFF	35,7484	230	175,1144	230	999789,14	230	210,8628	230,00
LPM0	35,7484	230	175,1144	230	999789,14	56	210,8628	56,04
LPM3	35,7484	230	175,1144	230	999789,14	0,5	210,8628	0,55

MSP430g2553; battery life is calculated based on *Table 16* 6V and mAh 440.

Table 22 Expected battery life running continuously - every 2 sec.

Every 2 sec	**Average current µA**	**Expected battery life running continuously, h**	**Days**	**Months**	**Years**
LPM OFF	230,00	1 913,04	79,71	2,62	0,22
LPM0	56,00	7 856,48	327,35	10,77	0,90
LPM3	0,51	869 174,15	36215,59	1191,30	99,28

Table 23 Expected battery life running continuously - every 1 sec.

Every 1 sec	**Average current µA**	**Expected battery life running continuously, h**	**Days**	**Months**	**Years**
LPM OFF	230,00	1 913,04	79,71	2,62	0,22
LPM0	56,01	7 855,83	327,33	10,77	0,90
LPM3	0,51	858 861,36	35785,89	1177,17	98,10

Table 24 Expected battery life running continuously - twice a sec.

Twice a sec	**Average current µA**	**Expected battery life running continuously, h**	**Days**	**Months**	**Years**
LPM OFF	230,00	1 913,04	79,71	2,62	0,22
LPM0	56,02	7 854,52	327,27	10,77	0,90
LPM3	0,52	838 618,76	34942,45	1149,42	95,79

Table 25 Expected battery life running continuously - 4 times each sec.

Four times pr sec	**Average current µA**	**Expected battery life running continuously, h**	**Days**	**Months**	**Years**
LPM OFF	230,00	1 913,04	79,71	2,62	0,22
LPM0	56,04	7 851,90	327,16	10,76	0,90
LPM3	0,55	801 003,42	33375,14	1097,87	91,49

Power consumption evaluation in *Table 20* is for 1 D6T-44L sensor on 1 microcontroller. Even connecting 4 sensors to the same microcontroller while managing power supply for these sensors will have little impact on the total power consumption for 4 sensors and 1 microcontroller.

Table 26 Power consumption evaluation.

Power consumption evaluation					
Reading frequencies		**2 sec**	**1 sec**	**0,5 sec**	**0,25 sec**
MSP430g2553 LPM off	Average current µA	230,00	230,00	230,00	230,00
D6T-44L constant on	Average current µA	5000,00	5000,00	5000,00	5000,00
MSP430g2553 LPM3	Average current µA	0,51	0,51	0,52	0,55
D6T-44L power management	Average current µA	0,02	0,04	0,09	0,1787
Total awake time LPM3 + D6T-44L power management		31,6094	62,4391	125,4236	250,6112
Total current consumption µA LPM3 + D6T-44L power management		0,53	0,56	0,61	0,73

Cost:

- MSP430g2553 Price $9.99
- D6T-44L Price $49.88

Total cost for covering one room using 1 microcontroller and 4 D6T-44L sensors is $209.51

4. Experimental results

4.1. Transmission of data

The test of transmission of data was completed using LogicLevel analyzer, LED's and RealTerm, LogicLevel analyzer for correctly verifying I^2C transmission waves, LEDs on microcontroller to see if the error check succeeded or failed, and RealTerm terminal on computer was used to verify if there was correct transmission between microcontroller and computer.

The LogicPort logic analyzer was used to verify the correct I^2C transmission (*Figure 24*). The LED on microcontroller would blink if an error was detected, and it ran without blinking. RealTerm was used to verify the transmission of data between the microcontroller and computer. Data was saved to the file from RealTerm and no error was detected.

The Java Swing application was used to read data from microcontroller and to save temperature data to the file and database. Transmitted data was checked using CRC and at some intervals there were errors in reading. These errors were saved to the text file with an additional timestamp. Because of these errors RealTerm was used to verify the transmission from the microcontroller. There was no error from the microcontroller, so the error is in the Java application. Troubleshooting and debugging of the swing application was done and it seems that the error lies in the method for serial port and input stream [42]. Example of error:

```
crc-8: PEC-error (Packet error check code) failed: 2015-06-05 16:24:15.609
      [67, 78, 101, 35, 29, 1, -2, 0, 3, 1, 3, 1, 1, 1, -4, 0, 0, 1, 2, 1, -
2, 0, -1, 0, 1, 1, 1, 1, 2, 1, 8, 1, 6, 1, 2, 1, 6, 1, 84]
```

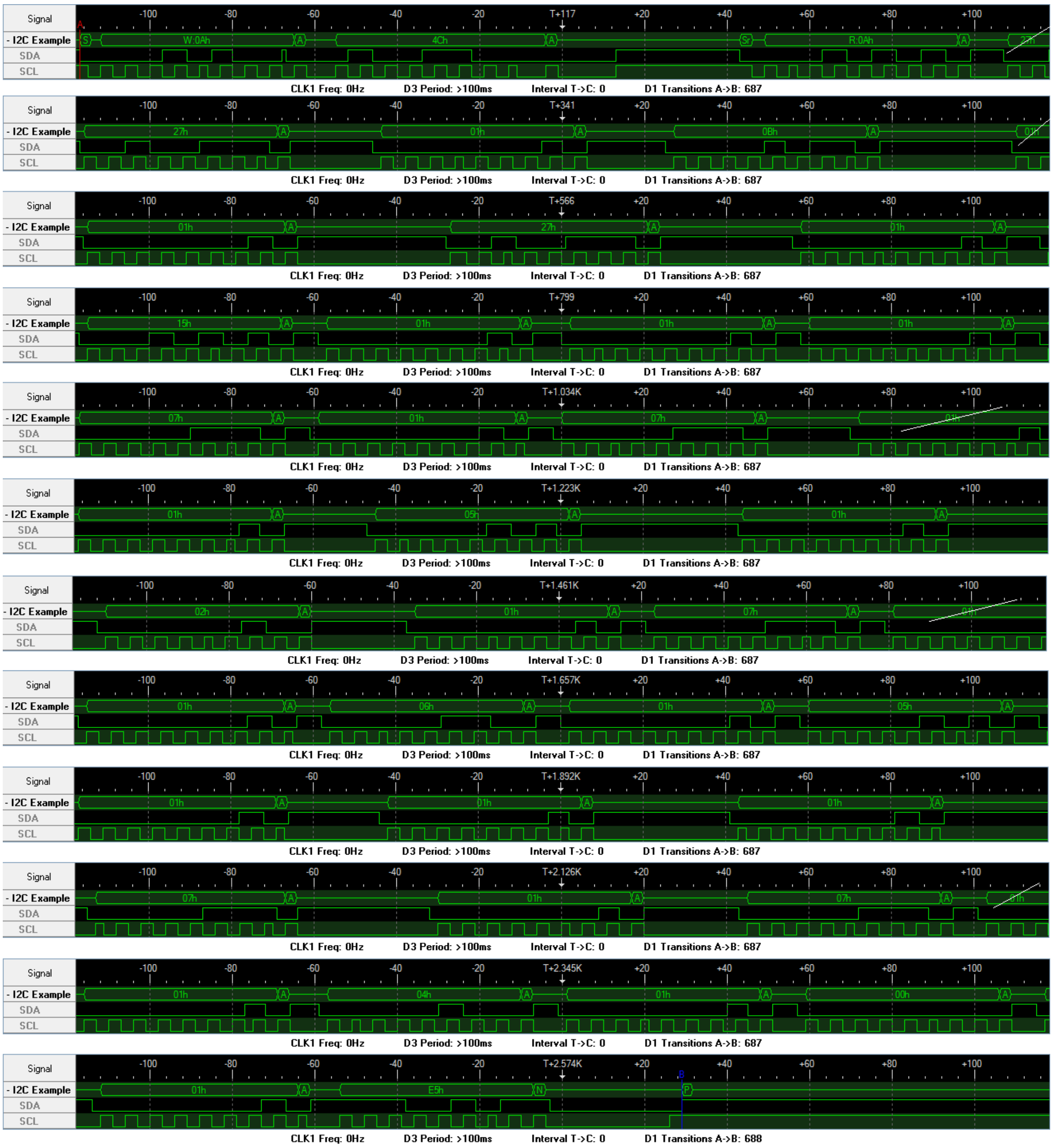


Figure 24 D6T-44L I2C transmission waves.

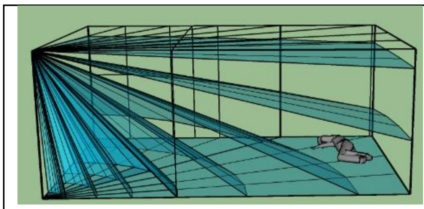
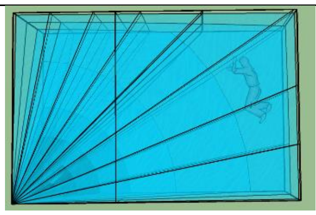
4.2. Measurement

Testing pictures are not correct according to the situation. Pictures have the sensor at 2.4 meter from the floor and at an angle of 25.91°. In the real situation the sensor is ca 2.25 meter from the floor and the angle is around 20-30 °. Distance measured in the room is also not accurate. More tests need to be done before one can make a conclusion. The human at 5 – 6 meters from the sensor can be hard to detect. It also depends on the room temperature and the clothing on the person and this holds true whether the person is lying, sitting or standing.

4.2.1. Measured temperatures: Test 1, lying person.

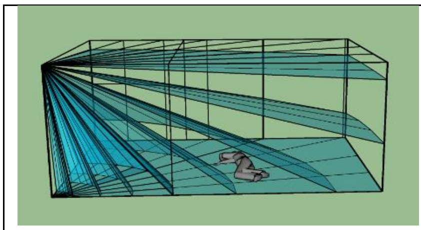
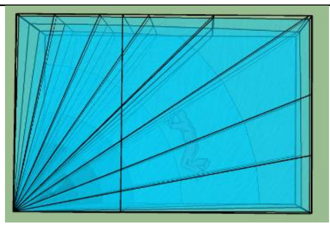
Tests presented here are from 2 different days, marked with Test day 1 and Test day 2. Test on “Test day 2” can be compared with “Calibration day 2”.

Table 27 Test 1, lying person, distance 5.0-6.0 meters.

<table>
<tr><td></td><td></td><td>1

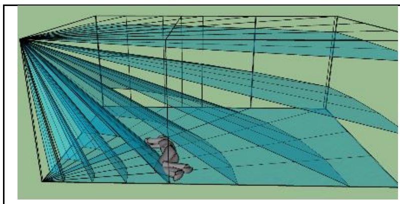
This person should be detected in p4-p7, second row from top.</td></tr>
<tr><td>Test day 1</td><td>Test day 2</td><td>Calibration day 2</td></tr>
<tr><td>
<table>
<tr><td>23.8</td><td>24.1</td><td>24.5</td><td>24.7</td></tr>
<tr><td>24.9</td><td>25.7</td><td>25.2</td><td>24.1</td></tr>
<tr><td>24.4</td><td>25.1</td><td>25.1</td><td>24.7</td></tr>
<tr><td>23.8</td><td>24.8</td><td>25.3</td><td>24.7</td></tr>
</table>
</td><td>
<table>
<tr><td>23.7</td><td>24.5</td><td>24.8</td><td>24.7</td></tr>
<tr><td>23.7</td><td>26.5</td><td>25.4</td><td>24.8</td></tr>
<tr><td>25.3</td><td>25.9</td><td>25.9</td><td>25.5</td></tr>
<tr><td>25.4</td><td>26.0</td><td>26.1</td><td>26.1</td></tr>
</table>
</td><td>Calibrated:
<table>
<tr><td>23.7</td><td>24.5</td><td>24.7</td><td>24.7</td></tr>
<tr><td>23.3</td><td>25.1</td><td>25.3</td><td>24.7</td></tr>
<tr><td>25.2</td><td>25.7</td><td>25.7</td><td>25.4</td></tr>
<tr><td>25.4</td><td>25.9</td><td>26.1</td><td>26.0</td></tr>
</table>
</td></tr>
<tr><td colspan="3">Comparing these result one can see that in “Test day 2“ p4-p7 have a higher temperature compared to calibrated data. Test day 2 calculated difference is 1.4.</td></tr>
</table>

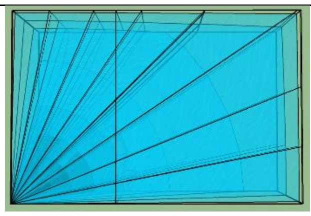
Table 28 Test 1, lying person, distance 3.2 – 5.0 meters.

<table>
<tr><td></td><td></td><td>2

This person should be detected in p8-p11, third row from top.</td></tr>
<tr><td>Test day 1</td><td>Test day 2</td><td>Calibration day 2</td></tr>
<tr><td>
<table>
<tr><td>23.7</td><td>24.2</td><td>24.6</td><td>24.6</td></tr>
<tr><td>24.3</td><td>25.6</td><td>25.4</td><td>24.2</td></tr>
<tr><td>25.0</td><td>26.1</td><td>25.4</td><td>24.9</td></tr>
<tr><td>23.8</td><td>24.9</td><td>25.4</td><td>24.9</td></tr>
</table>
</td><td>
<table>
<tr><td>23.8</td><td>24.5</td><td>24.8</td><td>24.8</td></tr>
<tr><td>23.5</td><td>25.5</td><td>25.5</td><td>25.0</td></tr>
<tr><td>25.8</td><td>28.3</td><td>26.7</td><td>25.6</td></tr>
<tr><td>25.6</td><td>26.1</td><td>26.3</td><td>26.2</td></tr>
</table>
</td><td>Calibrated:
<table>
<tr><td>23.7</td><td>24.5</td><td>24.7</td><td>24.7</td></tr>
<tr><td>23.3</td><td>25.1</td><td>25.3</td><td>24.7</td></tr>
<tr><td>25.2</td><td>25.7</td><td>25.7</td><td>25.4</td></tr>
<tr><td>25.4</td><td>25.9</td><td>26.1</td><td>26.0</td></tr>
</table>
</td></tr>
<tr><td colspan="3">Comparing these results one sees that the temperature difference in day 2 is significant, this is also expected as the human is closer to sensor.</td></tr>
</table>

Table 29 Test 1, lying person, distance 2.16- 3.0 meters.





3

This person should be detected in **p12-p15, fourth row from top.**

Test day 1

23.7	24.2	24.5	24.7
24.2	25.2	25.4	24.3
24.7	25.8	26.1	25.1
24.0	26.1	26.8	25.2

Test day 2

23.8	24.6	24.8	24.8
23.6	25.4	25.4	25.0
25.5	26.4	26.3	25.7
25.7	28.5	27.9	26.4

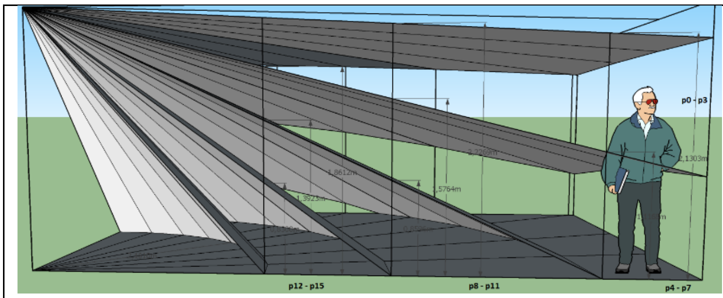
Calibration day 2

Calibrated:

23.7	24.5	24.7	24.7
23.3	25.1	25.3	24.7
25.2	25.7	25.7	25.4
25.4	25.9	26.1	26.0

In these tests the differences in temparetures are also significant, it also gives warmth to cells over the detected person, the person is heating up the air and the sensor is detecting also these temperatures.

Table 30 Test 2, standing person, distance 5.0-6.0 meters



6

This person should be detected in **p4-p7 and p0-p3, first and second row from top.**

Test day 1

24.9	24.9	24.7	24.9
24.8	25.6	25.7	24.5
25.1	25.8	25.7	25.2
24.2	25.4	25.8	25.2

Test day 2

23.9	25.0	25.7	24.8
23.6	25.7	26.2	25.0
25.5	26.0	26.2	25.7
25.6	26.2	26.3	26.3

Calibration day 2

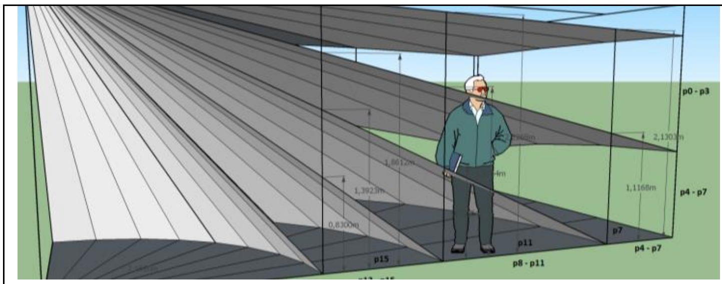
Calibrated:

23.7	24.5	24.7	24.7
23.3	25.1	25.3	24.7
25.2	25.7	25.7	25.4
25.4	25.9	26.1	26.0

In this test, detection is in the two upper most cells, and it still detects the warmth from previous position. It seems that the heat from the test person "is still in the room" because it is detected in the lower cells. Calculating the differences from test day 2 will give detection in 3 cells. Such errors can also cause inaccurate position of the sensor.

4.2.2. Measured temperatures: Test 2, standing person

Table 31 Test 2, standing person, distance 3.0-5.0 meters.



5

This person should be detected in **p8-p11, p4-p7 and p0-p3, first, second, and third row from top.**

Test day 1

24.5	25.0	24.7	24.8
25.1	26.2	25.6	24.4
25.0	25.9	25.6	25.1
24.2	25.4	25.8	25.1

Test day 2

23.9	24.8	25.8	24.9
23.7	26.0	27.6	25.1
25.5	26.2	26.8	25.8
25.6	26.2	26.5	26.3

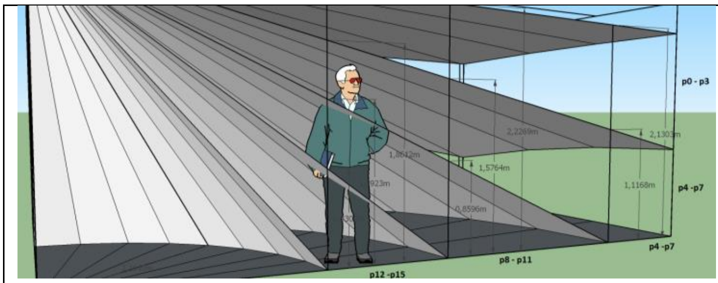
Calibration day 2

Calibrated:

23.7	24.5	24.7	24.7
23.3	25.1	25.3	24.7
25.2	25.7	25.7	25.4
25.4	25.9	26.1	26.0

Still significant differences in temperature and person is detected in 2-3 cells, but the warmth from previous test is also detected.

Table 32 Test 2, standing person, distance 2.16-3.0 meters.



4

This person should be detected in **p12-p15, p8-p11 and p4-p7, second, third and fourth row from top.**

Test day 1

23.9	24.7	24.6	24.7
24.7	28.1	25.7	24.3
25.1	28.0	25.8	25.0
24.1	26.0	25.8	25.0

Test day 2

23.9	24.7	25.5	25.0
23.7	25.9	28.7	25.3
25.5	26.5	28.5	26.1
25.6	26.4	27.4	26.4

Calibration day 2

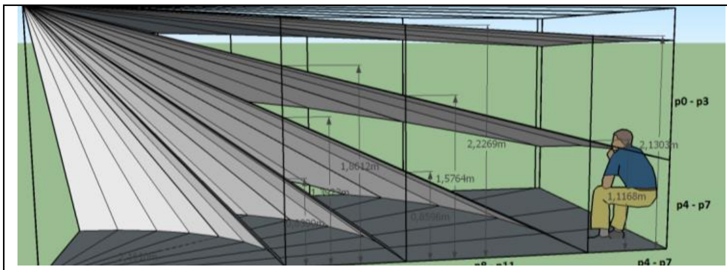
Calibrated:

23.7	24.5	24.7	24.7
23.3	25.1	25.3	24.7
25.2	25.7	25.7	25.4
25.4	25.9	26.1	26.0

Significant differences in temperature. Person is standing and as expected detection is in 3 cells/pixels in a row.

4.2.3. Measured temperatures: Test 3, sitting person.

Table 33 Test 3, sitting person, distance 5.0-6.0 meters.



7

This person should be detected in **p4-p7 and p0-p3, first and second row from top.**

Test day 1

24.3	24.5	24.7	24.9
25.0	25.7	25.6	24.5
25.0	25.7	25.7	25.2
24.2	25.4	25.7	25.2

Test day 2

23.9	24.7	25.1	24.8
23.7	25.9	26.4	25.0
25.5	26.0	26.2	25.7
25.6	26.1	26.4	26.2

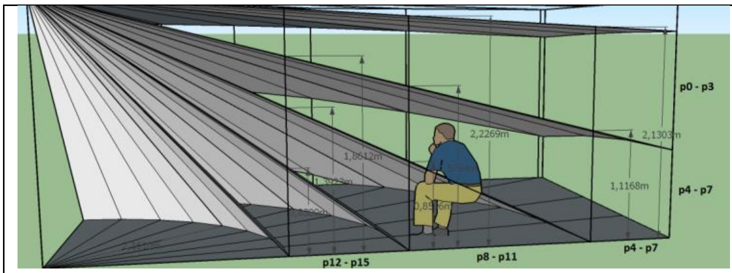
Calibration day 2

Calibrated:

23.7	24.5	24.7	24.7
23.3	25.1	25.3	24.7
25.2	25.7	25.7	25.4
25.4	25.9	26.1	26.0

Here we continue testing and it seems that the heat from the test person is “still in the air” as there is a detection of higher temperature in the cells without a person. It should not be possible to detect a person in a cell where he is not, but where he was some minutes ago. The reason may be that the test was done in a closed room and the test person was producing heat and heating the floor.

Table 34 Test 3, sitting person, distance 2.16-3.0 meters.



8

This person should be detected in **p8-p11 and p4-p7 second, and third row from top.**

Test day 1

24.0	24.5	24.7	24.8
25.0	26.7	25.7	24.5
25.1	25.9	25.7	25.2
24.3	25.4	25.8	25.2

Test day 2

23.9	24.6	24.9	24.9
23.7	26.1	27.1	25.1
25.5	26.4	26.7	25.8
25.7	26.2	26.4	26.3

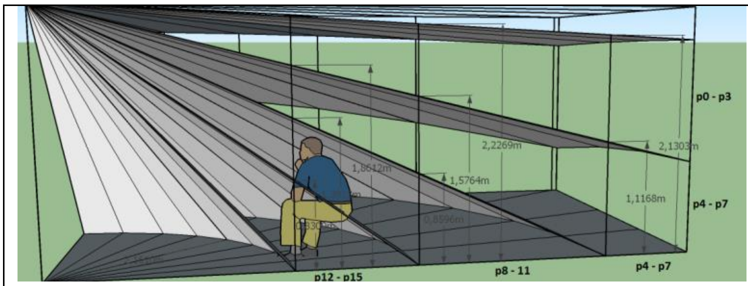
Calibration day 2

Calibrated:

23.7	24.5	24.7	24.7
23.3	25.1	25.3	24.7
25.2	25.7	25.7	25.4
25.4	25.9	26.1	26.0

In test day 2 the person is detected in 4 cells, *p5*, *p6*, *p9* and *p10*, this can happens because he can be detected in 2 adjecent cells at once.

Table 35 Test 3, sitting person, distance 2.16-3.0 meters.

<table>
<tr><td colspan="2">
</td><td>9

This person should be detected in p12-p15 and p8-p11, third and fourth row from top.</td></tr>
<tr><td>Test day 1</td><td>Test day 2</td><td>Calibration day 2</td></tr>
<tr>
<td><table>
<tr><td>23.9</td><td>24.3</td><td>24.7</td><td>24.8</td></tr>
<tr><td>24.5</td><td>26.4</td><td>25.8</td><td>24.5</td></tr>
<tr><td>25.1</td><td>28.2</td><td>26.3</td><td>25.2</td></tr>
<tr><td>24.3</td><td>26.1</td><td>26.1</td><td>25.3</td></tr>
</table></td>
<td><table>
<tr><td>23.9</td><td>24.6</td><td>24.9</td><td>24.9</td></tr>
<tr><td>23.6</td><td>25.6</td><td>26.3</td><td>25.1</td></tr>
<tr><td>25.6</td><td>27.0</td><td>28.7</td><td>25.9</td></tr>
<tr><td>25.7</td><td>26.9</td><td>27.5</td><td>26.4</td></tr>
</table></td>
<td>Calibrated:<table>
<tr><td>23.7</td><td>24.5</td><td>24.7</td><td>24.7</td></tr>
<tr><td>23.3</td><td>25.1</td><td>25.3</td><td>24.7</td></tr>
<tr><td>25.2</td><td>25.7</td><td>25.7</td><td>25.4</td></tr>
<tr><td>25.4</td><td>25.9</td><td>26.1</td><td>26.0</td></tr>
</table></td>
</tr>
<tr><td colspan="3">Here is a significant temperature detection. Test day 2 compared to Calibraiton day 2 shows that there are detection in 3 cells in row, again it can be inaccureate position of the sensor or the upper cell can be heated from the person. He also been detected in 2 adjecent cells.</td></tr>
</table>

4.2.4. Measured temperatures.

Table 36 Measured temperature of different objects.

Objects	**Temperature in pixel cell**	**Comments**
TV on	29-30 °C	
Wall light	29 °C	
Human close to sensor	28.3 °C	
Human long distance to sensor	25 °C	
Indoor temperature	23-27 °C	
Open terrace door	14 °C	Outside temperature °C
Ceiling LED lamp	29.0 °C	
Open window	14.8 °C	

Objects	Temperature in pixel cell	Comments
Empty room (living room)	22.3-23.5 °C	
Robot vacuum cleaner, returned to charge.	26.0 °C	Alerted on Robot vacuum cleaner.
Robot vacuum cleaner stated	26.0 °C	Will be detected in only one cell at time
Mobil on charging		
Human face, 5 cm	33.0 °C	
Human face, 1 meter	29.5 °C	Partial coverage of FOV
Human body, T-shirt, 1 meter	29.4 °C	Better coverage of FOV

4.2.5. Human patterns

The temperature patterns of a person shows that head and legs positions have a lower temperature than the body. This is also logical because the area of head is smaller than the area of body.

Example of position patterns where human can be detected (*Figure 25*).

Figure 25 Example of position of human being lying, standing or sitting

The variables from the system are surface temperature, pixel nr, sensor ID and timestamp (0). When using only one sensor the variables available for detection of human pattern are surface temperature and pixel nr (*pixel position*). Timestamp is usable for dynamic detection.

Graphical presentation of sensor data:

- Line graph/Bezier curve (*Figure 26*) – to display line chart one needs two attributes, x-axis and y-axis.
- Bar graph (*Figure 27*) – to display bar chart one needs two attributes, x-axis and y-axis.
- Bubble chart (*Figure 27*) – to display bubble chart one need three attribute, x-axis, y-axis and a size-by values.
- Combo chart (*Figure 29*) – it is combined by two different scales (y-axis) and accompanied with different data sets and chart types.

The need to compare several temperature readings together to detect common characteristics arose. Using Line graph or Bezier curve it is easy to compare several temperature readings, each line represents one reading. Using a Bar graph it limits the number of readings to compare, one can easily compare 2 or 3 but more than this will be harder. A Bubble chart is best if one wants to look at only one reading at time. Several readings in Bubble chart will mix colors together and be harder to analyze. A Combo chart is used when one has multiple and different information/datasets sharing the same dimension. And when one also has to code using a 16-bit temperature array it may be is easier to “read and think” thus data as one long line instead of a bar chart or a bubble chart. The goal is to come up with a solution with as little as possible to code on a microcontroller.

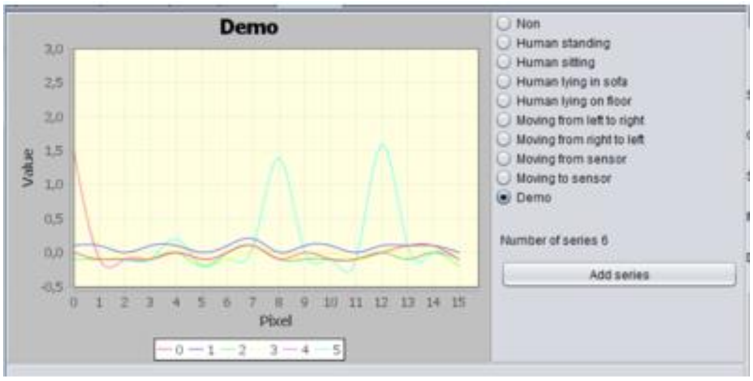


Figure 26 Comparing different sensor reading with each other.

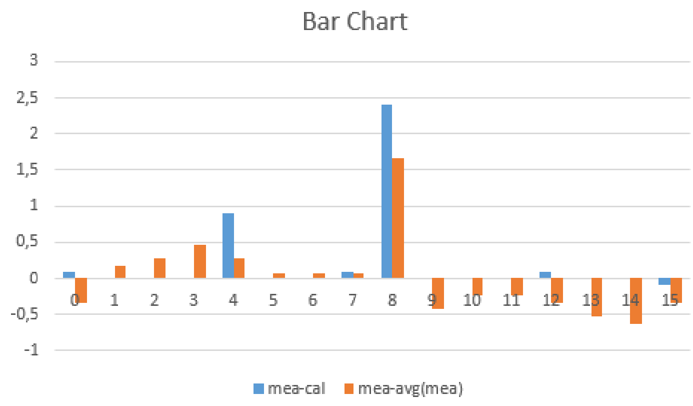


Figure 27 Bar chart comparing values for the two algorithms.

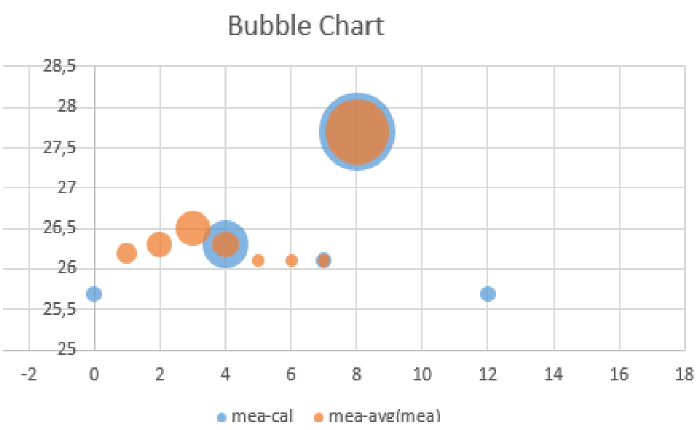


Figure 28 Bubble chart with temperatures and bubble in size of differences.

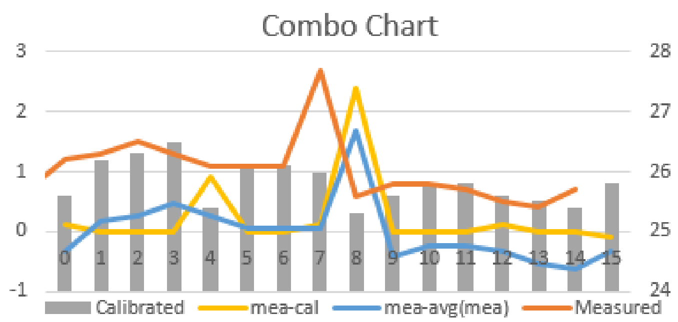


Figure 29 Combined chart of temperature and differences.

Human pattern

In these tests, measurement of 2 people is used. Before reading temperature values, the sensor is calibrated without a person and the calibration is used to calculate and draw a new series in the graph. As the room temperature is raising during the test, all tests are recalibrated before a new test is begun. This means new calibration before every new graph. The graph shows the difference between calibration value and the detected value.

X-value is pixel number 0 to 15, y-value is the temperature differences between the calibrated temperature and the measured temperature, and each serie/curve is one reading. The choice of using Bezier curve over Line graph is pure convenience or habit.

Pattern of empty room:

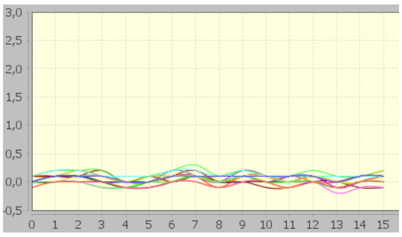


Figure 30 Curves of empty room

In a room without a human the temperature fluctuate with ± 0.4° C, *Figure 30.*

Pattern of person standing, 5 m, 3.5 m, 2.5 m:

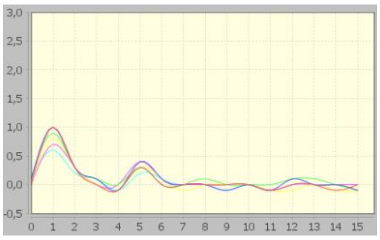


Figure 31 Curves show a person standing at 5 m

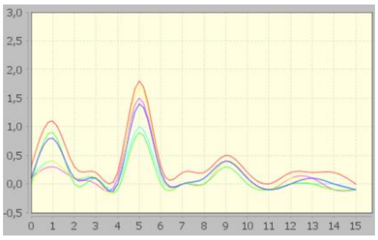


Figure 32 Curves show a person standing at 3.5 m

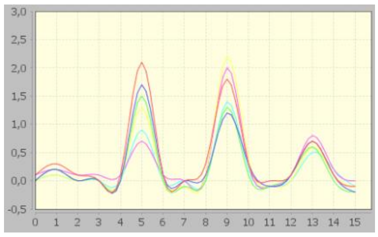


Figure 33 Curves show a person standing at 2.5 m

In *Figure 31* the person is standing, pixel 1 and 5 has a peak that means detection in *p1* and *p5* and 2 pixels in rows. In *Figure 32* the peak is at pixel 1, 5 and 9, which is detection is in pixel *p1*, *p5* and *p9* and 3 pixels in rows. In *Figure 33* the peak is in *p5*, *p9* and *p13* and 3 pixels in rows.

Pattern of person sitting, 5 m, 3.5 m, 2.5 m:

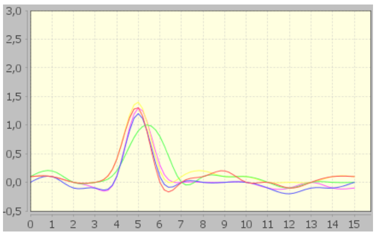


Figure 34 Curves show a person sitting at 5 m

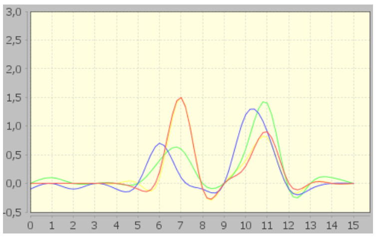


Figure 35 Curves show a person sitting at 3.5 m

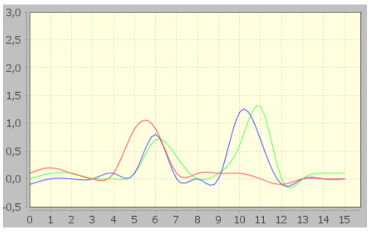


Figure 36 Curves show a person sitting at 2.5 m

In *Figure 34* the peak is in *p5*, only 1 detection. In *Figure 35* peak is in *p6*, *p7*, *p10* and *p11*, this can be differences in measurement of 2 persons or it can be 1 person detected in 4 cells. In *Figure 36* the peak is in *p5*, *p6* and *p10*, *p11*, can be the same as in *Figure 35*.

Pattern of a person lying on the sofa, 5 m, 3.5 m:

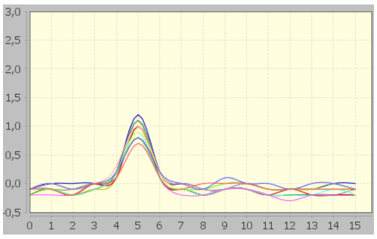


Figure 37 Curves show a person lying on the sofa at 5 m

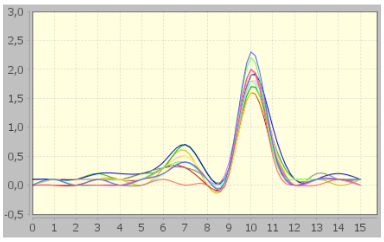


Figure 38 Curves show a person lying on the sofa at 3.5 m

In *Figure 37* peak is in 5 (*p5*), and in *Figure 38* the peak is in 7 and 10 (*p7* and *p10*), which is cross detection.

Pattern of a person lying on the floor, 5 m, 3.5 m, 2.5 m:

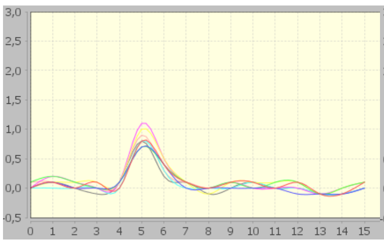


Figure 39 Curves show a person lying on the floor at 5 m

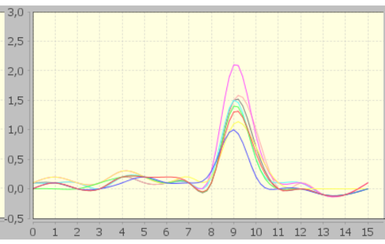


Figure 40 Curves show a person lying on the floor at 3.5 m

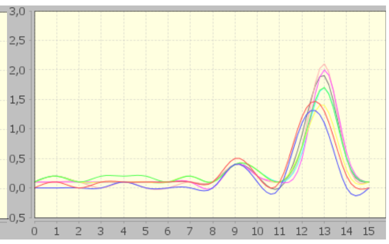


Figure 41 Curves show a person lying on the floor at 2.5 m

In *Figure 39* detection in 5 (p5). In *Figure 40* detection in 9 (p9). In *Figure 41* detection in 9, 12 and 13 (p9, p12 and p13) a sort of cross detection, and this is so close to the sensor that it can be difficult to distinguish between pixels. The pixel area is so small that 1 lying person will easily cover several pixels.

Pattern of moving right to left and left to right:

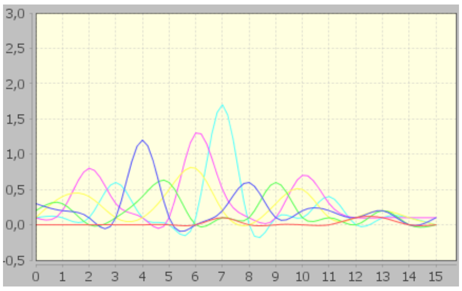


Figure 42 Curves show a person moving from left to right

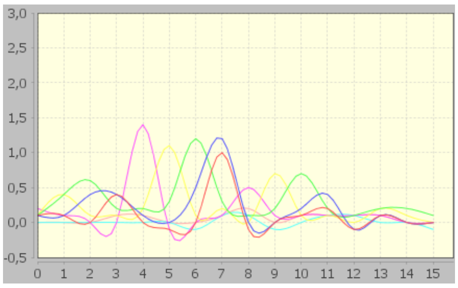


Figure 43 Curves show a person moving from right to left

Person moving from left to right in *Figure 42*. In *Figure 43* person is moving from right to left. Each series is 1 step, to the right or left.

Pattern moving from and against sensor:

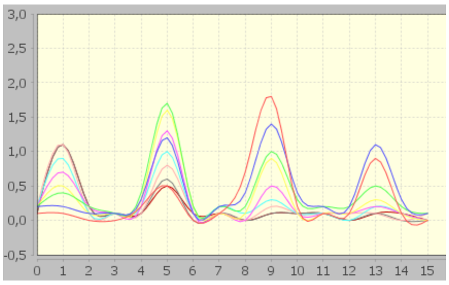


Figure 44 Curves show a person moving from sensor

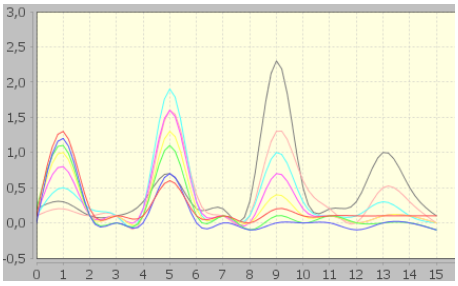


Figure 45 Curves show a person moving towards sensor

Figure 44 and *Figure 45* shows person moving away from and towards sensor. These curves will have the same patterns as a standing person, but in this test the curves are in the same pixels in each step, and the temperature rises and drops according to the distance from the sensor.

When one summarizes temperatures from the graph above one can read following:

- One person will normally not be detected covering all 4 pixels in a row/column at the same time, only very tall persons or a person very close to the sensor. At the most a person can be detected in 3 pixels in a row or column at the same time.

- If one person is standing he or she will be detected in 2 or 3 pixels in a column.
- If one person is sitting or lying he or she will be detected in 1 or 2 pixels in a row.
- Temperature fluctuations are greater the closer the person is to the sensor.
- When using the D6T-44L 4x4 sensor one can also estimate the surrounding temperature (room temperature) from one reading. Programmatically one can calculate the surrounding temperature from one reading. Removing all pixels with detection (*using the proposed theory/algorithm*) and the rest of pixels can be used to calculate the average room temperature.

Table 37 confirms the theory from the previous chapter (3.6) and human patterns from above. Table cells for zones with detection are marked with a light grey background, no measured is marked with hyphen '-'.

Table 37 Temperature fluctuation.

		2.5 m	**3.5 m**	**5-6 m**
Empty room	**p0-p3**	-	-	± 0.3° C
	p4-p7	-	-	± 0.3° C
	p8-p11	-	-	± 0.3° C
	p12-p15	-	-	± 0.3° C
Standing	**p0-p3**	± 0.3° C	**+ 0.3-1.1**	**+ 0.6-1.0**
	p4-p7	**+ 0.7-2.1**	**0.9-1.8**	**0.2-0.4**
	p8-p11	**1.2-2.1**	**0.3-0.5**	± 0.3° C
	p12-p15	**0.5-0.8**	± 0.3° C	± 0.3° C
Sitting	**p0-p3**	± 0.3° C	± 0.3° C	± 0.3° C
	p4-p7	**+ 0.7-1.0**	**+ 0.6-1.5**	**+ 1.0-1.4**
	p8-p11	**+ 0.0-1.3**	**+ 0.8-1.4**	± 0.3° C
	p12-p15	± 0.3° C	± 0.3° C	± 0.3° C
Lying on the sofa	**p0-p3**	-	± 0.3° C	± 0.3° C
	p4-p7	-	**+ 0.0-0.7**	**+ 0.7-1.2**
	p8-p11	-	**+ 1.6-2.3**	± 0.3° C
	p12-p15	-	± 0.3° C	± 0.3° C
Lying on the floor	**p0-p3**	± 0.3° C	± 0.3° C	± 0.3° C
	p4-p7	± 0.3° C	± 0.3° C	**+ 0.7-1.1**
	p8-p11	**+ 0.4-0.5**	**+ 1.3-2.1**	± 0.3° C
	p12-p15	**+ 1.1-2.1**	± 0.3° C	± 0.3° C

4.2.6. Data for historical use:

Each row are 19 x 2 bytes plus bytes for Timestamp = 36 bytes + 10 bytes = 46 bytes. Measuring temperature 4 times each second will give 240 rows each minute, 14 400 rows each hour, 345 600 row each day and 126 144 000 row each year. Total is around 5 GB with temperature data each year.

Table 38 Size of database storing temperature values.

	Each minute	**Each hour**	**Each day**	**Each year**	**Bytes each year**
4 times per sec	240	14 400	345 600	126 144 000	5 802 624 000
2 times per sec	120	7 200	172 800	63 072 000	2 901 312 000
1 times per sec	60	3 600	86 400	31 536 000	1 450 656 000
Each 2 second	30	1 800	43 200	15 768 000	725 328 000

4.3. Test of algorithms

Both of the proposed algorithms are implemented in the D6T-44L Thermal Temperature Reader application and run in real time. Therefore it is possible to compare them live.

Figure 46 compares the 2 algorithms. Algorithm 1 shows the difference between the real time value and the calibrated values (*minus a threshold of 0.5°C*). Algorithm 2 is using the average value of real time samples, which is a fixed value, and shows difference between the real time value and this fixed value (*minus a threshold of 0.5°C*). The value that is subtracted from Algorithm 2 is greater than the value that is subtracted from Algorithm 1. This average temperature will normally be higher than the temperature value from the calibrated room. For example in the pixel with value 29.5, in **Algorithm 1** this will be **29.5-26.4** (*p5 values*) **= 3.1** and in **Algorithm 2** it will be **29.5-26.8** (*average of real time values*) **= 2.7**. One can maybe understand it as Algorithm 2 and therefore also remove some "noise".

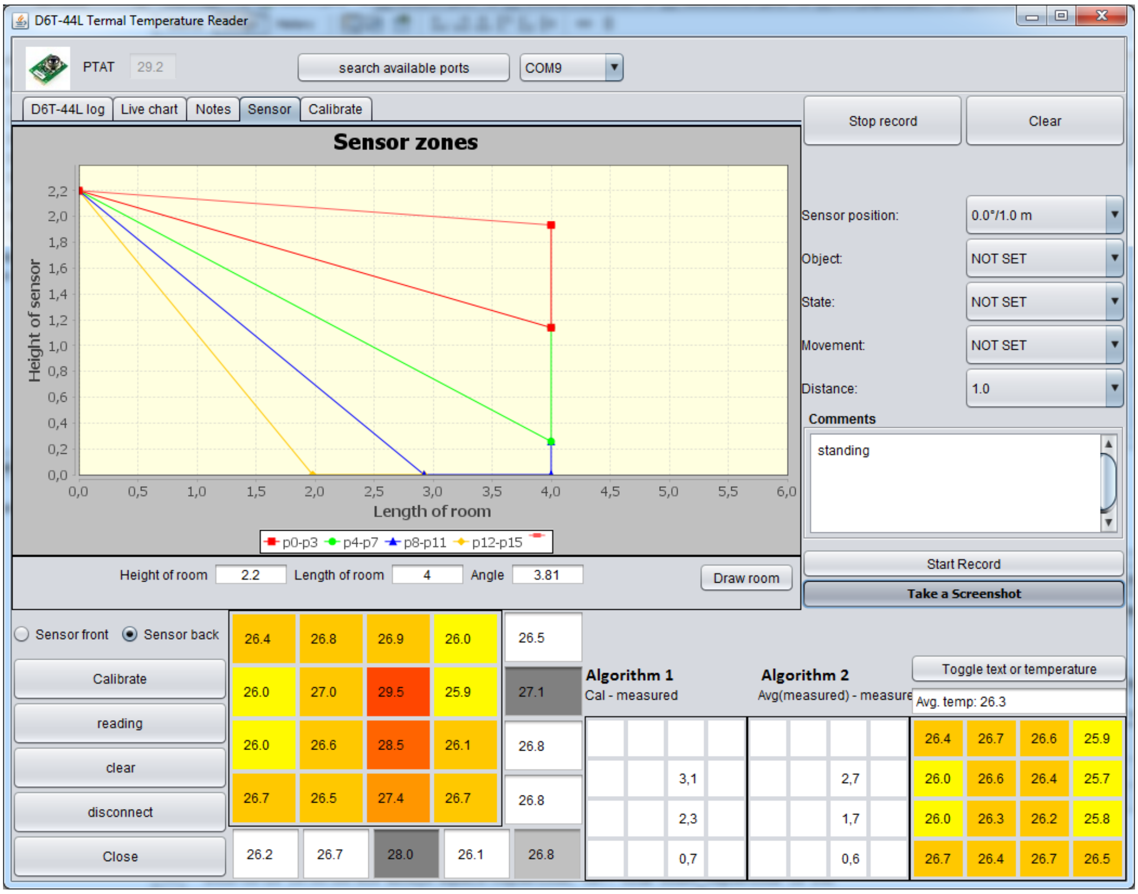


Figure 46 Comparing algorithms of a standing person

5. Conclusion and Outlook

The first part of this master thesis was doing research which is "state-of-the-art" and comparing different non-contact sensors. Some of the sensors have promising properties, and these comparison are based on information and specifications provided by the companies and not tested in this master thesis.

Testing of multiple sensors might depart at the expense of time spent on getting a reliable and secure reading. Too much time has been used on I^2C transmission, and it needed some detailed investigation to get the correct transmission wave and to reread the sensor. The source of the problem of difficult I^2C based sensor addressing has been a combination of the need of correct readout procedure finalization and a well-reflected setting of nested interrupts.

The next part was to classify and calibrate. The difficult thing was to calibrate an elderly person. One could use: the clothing being worn, the size, the gender, handicap or various diseases, but the sensor used the surface area and nothing else and therefore it was omitted. In the end of this master thesis it was discovered that also the temperature can or must be classified.

Proposed algorithm and pattern recognition. The solution with the *"document-term matrix"* and binary matrix was preferred. The binary matrix is so small and suitable for running on a microcontroller and it seems that it is a use of the Markov chain and the Kalman filter.

Further work:

- Implementation and test of the proposed theory and pattern recognition on the microcontroller and test the server. Both of the proposed theories are implemented on the D6T-44L Thermal Temperature Reader java application but not the pattern recognition theory.
- Do more tests on persons lying/sitting detected in the same pixels and do calculation/comparing of the temperatures. For example, question some may be: Are there any similarities or characteristics for the patterns? Is it possible to distinguish between sitting or lying person using temperature data? What is the ratio between temperature in sitting and lying situations? Can the ratio be used?

- Implementation and test of multiple sensor configurations. With usage of multiple sensors there will be detection in more than one sensor at the same time. With a larger detection matrix the pattern detection and decision tree will be more complicated.
- Test of the theory proposed in this thesis on a novel 16x16 thermal sensor with enhanced pattern recognition.
- Detections of pets or intelligent moving robotic home appliances were not considered. It remains to compare these to the human patterns.
- Classify the temperature. Various temperature can give alarm or need an attention.

In one sample reading the D6T-44L sensor used in this master thesis gives information about room temperatures, where the detection is in the FOV and the temperature of the detected pixel. One can subtract the pixels with the detections and use the remaining pixels to calculate the average temperature of the room. One can use a function of temperatures to calculate a certain but not accuracy distance from the sensor to the person. In more than one sample reading the sensor gives movements, directions and changes.

This master thesis has shown how rapidly the spiral of new technology is evolving. When working with one technology one day, the next day this technology is old technology from yesterday. To keep up with the development one needs the ability to quickly adapt. This study used the Omron D6T-44L 4x4 pixels sensor, which had sales start in 2013, while only a short time after this master thesis ends, there will be already a new Omron 16x16 pixel MEMS non-contact thermal sensor with a higher precision and a wider angle.

References

[1] N. as, "Helse-Norge mot 2030 Slik virker eldrebølgen," [Online]. Available: http://www.telenor.no/Images/Rapport1_nyforside_tcm52-208065.pdf. [Accessed May 2015].

[2] H. Lindblad, "Innovasjon står klar til å ta imot eldrebølgen," 26 Jan 2015. [Online]. Available: http://www.aftenbladet.no/meninger/Innovasjon-star-klar-til-a-ta-imot-eldrebolgen-3618145.html. [Accessed May 2015].

[3] E. Bergskaug, "Forsvant sporløst fra sykehus - ble aldri funnet (Disappeared without a trace from hospital - were never found)," 11 11 2014. [Online]. Available: http://www.abcnyheter.no/livet/2014/11/11/211558/forsvant-sporlost-fra-sykehus-ble-aldri-funnet. [Accessed May 2015].

[4] "locatible," Locatible, [Online]. Available: http://locatible.com/index.html. [Accessed May 2015].

[5] V. Stunt, "Use of surveillance tech to monitor seniors at home on rise," 09 Mar 2014. [Online]. Available: http://www.cbc.ca/news/technology/use-of-surveillance-tech-to-monitor-seniors-at-home-on-rise-1.2535677. [Accessed May 2015].

[6] Shin-ichiro Kaneko, Genci Capi, "Human-Robot Communication for Surveillance of Elderly People in Remote Distance," *2014 International Conference on Future Information Engineering,* p. 6, 2014.

[7] "A home monitoring system for elderly people based on MEMS sensors and wireless networks," *SENSORS, 2013 IEEE,* pp. 1-4, 3-6 Nov 2013.

[8] Yiping Tang, Baoqing Ma, Hangchen Yan, "Intelligent Video Surveillance System for Elderly People Living Alone Based on ODVS," *Published Online June 2013 (http://www.scirp.org/journal/ait),* p. 9, 28 Apr 2013.

[9] S. C. Staff, "Are Granny Cams Ethical?," 17 Feb 2012. [Online]. Available: http://www.seniorcare.net/are-granny-cams-ethical/. [Accessed May 2015].

[10] A. Sauer, "10 Lifesaving Location Devices for Dementia Patients," 14 Aug 2014. [Online]. Available: http://www.alzheimers.net/8-8-14-location-devices-dementia.

[11] "GPS Smart Sole," GTX Corp, [Online]. Available: http://www.gpssmartsole.com/. [Accessed May 2015].

[12] "MazeMap," [Online]. Available: http://www.mazemap.com/. [Accessed May 2015].

[13] Lih Chieh Png; Liangquan Chen; Song Liu; Wei Kuan Peh,, "An Arduino-based indoor positioning system (IPS) using visible light communication and ultrasound," *Consumer Electronics - Taiwan (ICCE-TW), 2014 IEEE International Conference on,* pp. 217-218, 26-28 May 2014.

[14] N. D. H. Jyunichi Oshita, "LED Lamp Comes With Radar Device to Monitor Elderly People," 11 11 2014. [Online]. Available: http://techon.nikkeibp.co.jp/english/NEWS_EN/20141111/388150/?P=1. [Accessed 18 05 2015].

[15] G. Povey, "Visible Light Communications Top 10 FAQs," 31 Aug 2011. [Online]. Available: http://visiblelightcomm.com/visible-light-communications-top-10-faqs/. [Accessed May 2015].

[16] T. Instruments, "3D Time of Flight Imaging Solutions," [Online]. Available: http://www.ti.com/ww/en/analog/3dtof/. [Accessed May 2015].

[17] "OMRON Releases Non-Contact D6T MEMS Thermal Sensor," 27 June 2012. [Online]. Available: http://www.omron.com/media/press/2012/06/e0627.html. [Accessed Apr 2015].

[18] O. E. C.-. Americas, "OMRON MEMS Sensors: D6F-PH Differential Pressure and D6T Thermal IR Sensors," 28 Oct 2013. [Online]. Available: https://www.youtube.com/watch?v=-w6rj4L3cg0. [Accessed May 2015].

[19] "D6T MEMS Thermal Sensors," OMRON Corporation, 31 Aug 2012. [Online]. Available: http://www.omron.com/ecb/products/sensor/11/d6t.html. [Accessed 2015].

[20] Heidi Gomnæs, Rune Fredriksen, "Faller du ut av senga? Da går alarmen (Fall detection project in Norwegian hospital)," 21 May 2015. [Online]. Available: http://www.nrk.no/ostfold/faller-du-ut-av-senga_-da-gar-alarmen-1.12371614. [Accessed 25 May 2015].

[21] "Depth Sensor Shootout: Kinect, Leap, Intel and Duo," 30 June 2014. [Online]. Available: http://stimulant.com/depth-sensor-shootout/. [Accessed May 2015].

[22] "DUO3D," DUO3D™ Division Code Laboratories, Inc., [Online]. Available: https://duo3d.com/. [Accessed May 2015].

[23] "PIR Motion Sensor (JST)," SparkFun Electronics, [Online]. Available: https://www.sparkfun.com/products/13285. [Accessed 2015].

[24] "Mesa Imaging SR4000," Mesa Imaging, [Online]. Available: http://acroname.com/products/MESA-IMAGING-SR4000. [Accessed 2015].

[25] "Ultrasonic Range Finder - LV-MaxSonar-EZ1," SparkFun Electronics, [Online]. Available: https://www.sparkfun.com/products/639. [Accessed 2015].

[26] "HC-SR04 Ultrasonic Range Finder," iTead Studio, [Online]. Available: http://www.robotshop.com/en/hc-sr04-ultrasonic-range-finder.html. [Accessed 2015].

[27] "Datasheet for D6T MEMS Thermal Sensors," OMRON Corporation, 31 August 2012. [Online]. Available: http://www.omron.com/ecb/products/pdf/en-d6t.pdf. [Accessed March 2015].

[28] "TI LaunchPad Develop. Make. Innovate.," Texas Instruments, [Online]. Available: http://www.ti.com/ww/en/launchpad/launchpads-msp430-msp-exp430g2.html. [Accessed 2015].

[29] "Application Note 01 Usage of D6T-44L / D6T-8L Thermal sensor," OMRON Corporation, 2012-2013. [Online]. Available: http://www.omron.com/ecb/products/sensor/special/mems/pdf/AN-D6T-01EN_r2.pdf. [Accessed March 2015].

[30] "SparkFun Logic Level Converter - Bi-Directional," SparkFun Electronics, [Online]. Available: https://www.sparkfun.com/products/12009. [Accessed 2015].

[31] D.-I. K. Gorontzi, "Online CRC Calculation," GHS Infotronic, 05 May 2012. [Online]. Available: https://ghsi.de/CRC/. [Accessed April 2015].

[32] M. Vega, "Calculating CRC With TI Battery Management Products," Texas Instruments Incorporated, October 2005. [Online]. Available: http://www.ti.com/lit/an/slua363/slua363.pdf. [Accessed 01 April 2015].

[33] "Java DB," Oracle, [Online]. Available: http://www.oracle.com/technetwork/java/javadb/overview/index.html. [Accessed March 2015].

[34] G. Litovsky, "Tutorials (MSP430 Microcontroller Tutorial)," [Online]. Available: http://glitovsky.com/blog/?page_id=21. [Accessed Mar 2015].

[35] Chris Parris, Jonathan Dillon, "Design calculations for robust I2C communications," 23 04 2012. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1279579. [Accessed April 2015].

[36] D. Gilbert, "JFreeChart," [Online]. Available: http://www.jfree.org/jfreechart/. [Accessed April 2015].

[37] "JFreeChart Installation," Tutorials Point, [Online]. Available: http://www.tutorialspoint.com/jfreechart/jfreechart_installation.htm. [Accessed April 2015].

[38] "Omron MEMS thermal area sensor checks for human presence and hot food," DigInfo TV, 13 July 2012. [Online]. Available: http://www.diginfo.tv/v/12-0135-r-en.php. [Accessed May 2015].

[39] J. Anjer, "Matriser," [Online]. Available: www.jbi.hio.no/bibin/KoG21/matriser.ppt. [Accessed June 2015].

[40] I. C. Borge, "Lineære likningssystemer, vektorer og matriser," 2008. [Online]. Available: http://www.uio.no/studier/emner/matnat/math/MAT1001/h08/Kompendium1/Kapittel2.pdf. [Accessed June 2015].

[41] "Matrix Multiplication," WolframMathWorld, 29 May 2015. [Online]. Available: http://mathworld.wolfram.com/MatrixMultiplication.html. [Accessed 08 June 2015].

[42] C. Taskiran, "problem reading data from serial port," 18 Sep 2008. [Online]. Available: http://www.coderanch.com/t/416504/java-io/java/reading-data-serial-port. [Accessed 2015].

[43] R. V. White, "System Management Bus (SMBus) Specification Version 3.0," System Management Interface Forum, 20 Dec 2014. [Online]. Available: http://smbus.org/specs/SMBus_3_0_20141220.pdf. [Accessed 2015].

[44] "What is MEMS Technology?," MEMS and Nanotechnology Exchange, [Online]. Available: https://www.mems-exchange.org/MEMS/what-is.html. [Accessed 29 April 2015].

[45] D. Askew, "MEMS: A Brief Overview," Mouser Electronics, [Online]. Available: http://www.mouser.ee/applications/mems-overview/. [Accessed April 2015].

[46] "OMRON Develops the World's First* 16x16 Element MEMS Non-Contact Thermal Sensor for Use in Human Presence Sensors Utilizing Wafer-Level Vacuum Packaging Technology," OMRON, 29 May 2013. [Online]. Available: http://www.omron.com/media/press/2013/05/e0529.html. [Accessed March 2015].

[47] A. Ng, "https://www.coursera.org/learn/machine-learning/," Coursera, [Online]. Available: https://www.coursera.org/learn/machine-learning/. [Accessed May 2015].

[48] "Kalman filter," [Online]. Available: http://en.wikipedia.org/wiki/Kalman_filter.

[49] B. Davidsen, "Markov-kjeder," [Online]. Available: http://ansatte.uit.no/bjorn.davidsen/MatteNotater/DifferensLiknSystem/Markov.pdf. [Accessed May 2015].

[50] P. E. I. Addison-Wesley product, "Markov chains," 2003. [Online]. Available: http://www.aw-bc.com/greenwell/markov.pdf.

[51] "Markov chain," [Online]. Available: http://en.wikipedia.org/wiki/Markov_chain. [Accessed May 2015].

[52] "SVM - Support Vector Machines," [Online]. Available: http://www.support-vector-machines.org/. [Accessed May 2015].

[53] "Support vector machine," [Online]. Available: http://en.wikipedia.org/wiki/Support_vector_machine. [Accessed May 2015].

[54] Murata, S.; Yara, C.; Kaneta, K.; Ioroi, S.; Tanaka, H., "Accurate Indoor Positioning System Using Near-Ultrasonic Sound from a Smartphone," *Next Generation Mobile Apps, Services and Technologies (NGMAST), 2014 Eighth International Conference on*, pp. 13,18, 10-12 09 2014.

[55] Sathiya.T, Prof.E.Divya, Prof.S.Raja, "Visible Light Communication for Wireless Data Transmission," *INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL NGINEERING*, p. 5, Feb 2014.

Appendix 1 – OMRON FAQ

OMRON

[D6T-44L/D6T-8L] Application Note No.MDMK-12-0493

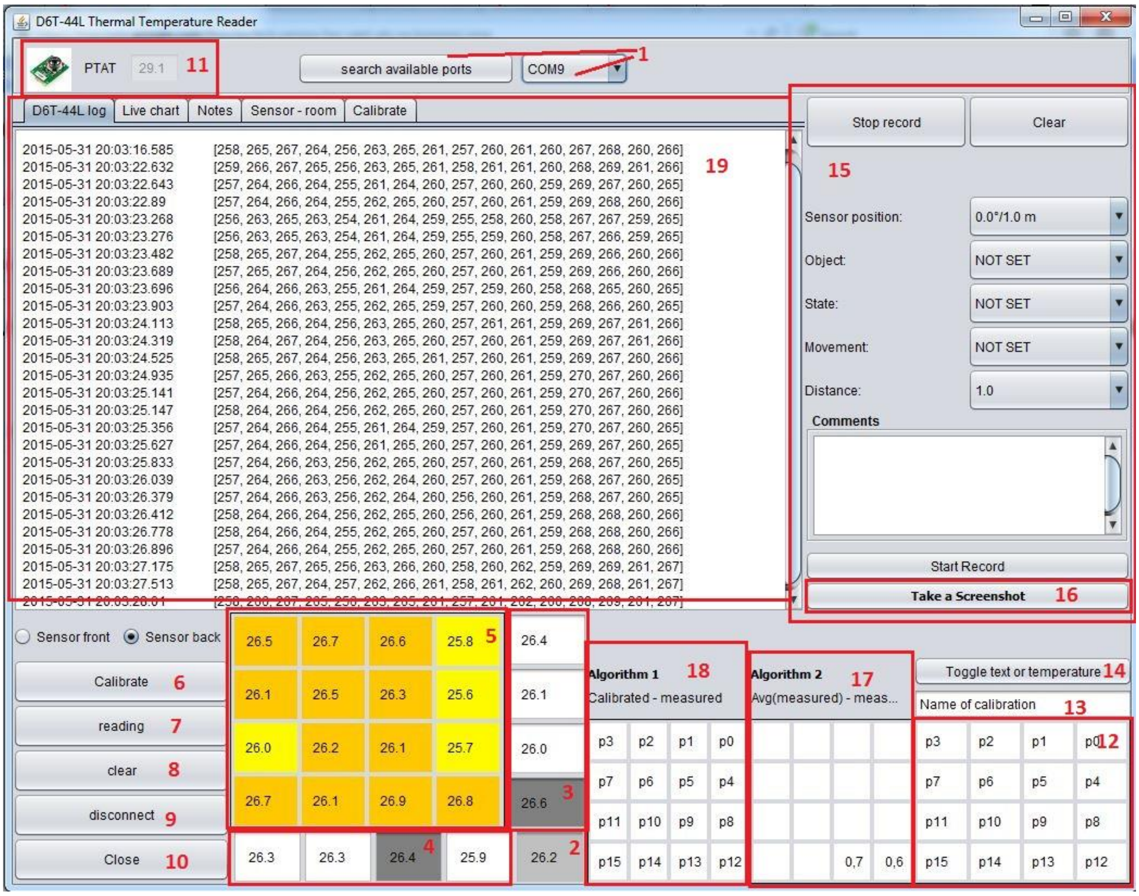
7 FAQ

Question	Can the field of view (FOV) angle be increased?
Answer	No. OMRON set the FOV in consideration of the constraints imposed by the thickness and refractive index of the silicon lens. Measurement distance is reduced as the FOV of one element increases. Therefore, we can not simply widen the viewing angle. A good way to measure a wide range, is to install multiple sensors, or mount the senor on a movable/rotating base.
Question	Are there any effects on an infrared remote controller?
Answer	No. The silicon lens we are using will not pass through most near-infrared and visible light below 1.2 [μm] wavelength. Therefore, it does not affect the infrared signal of the remote controller. The far infrared rays that are emitted as radiant heat are about 4 to 14 [μm].
Question	Is it possible to distinguish between humans, animals, and appliances?
Answer	No. In the non-contact temperature module, you can only acquire surface temperature measurement data. Different objects of the same temperature will read the same. Further discrimination must be based on the behavior of the measured data to distinguish the object by software on the user side. By developing software designed with your specific application in mind, the determination accuracy may possibly be improved.
Question	What is the distance range that can detect the presence of people?
Answer	This is greatly affected by the decision performance and software installation conditions. It is also affected by the size of the object to be measured and the area of the FOV per element. A rough guideline distance is about 5 to 6 meters.
Question	Can the power consumption be reduced?
Answer	No. The D6T thermal sensor does not have a power saving mode. Therefore, in order to reduce power consumption it is necessary to shut off the power.
Question	Is there a sensor that can operate on a supply voltage of 3[V]? Is there an I2C slave address that I can change?
Answer	No. The D6T thermal sensor does not support them.

1FAQ [29]

Appendix 2 – Functionality of D6T-44L Thermal Temperature Reader

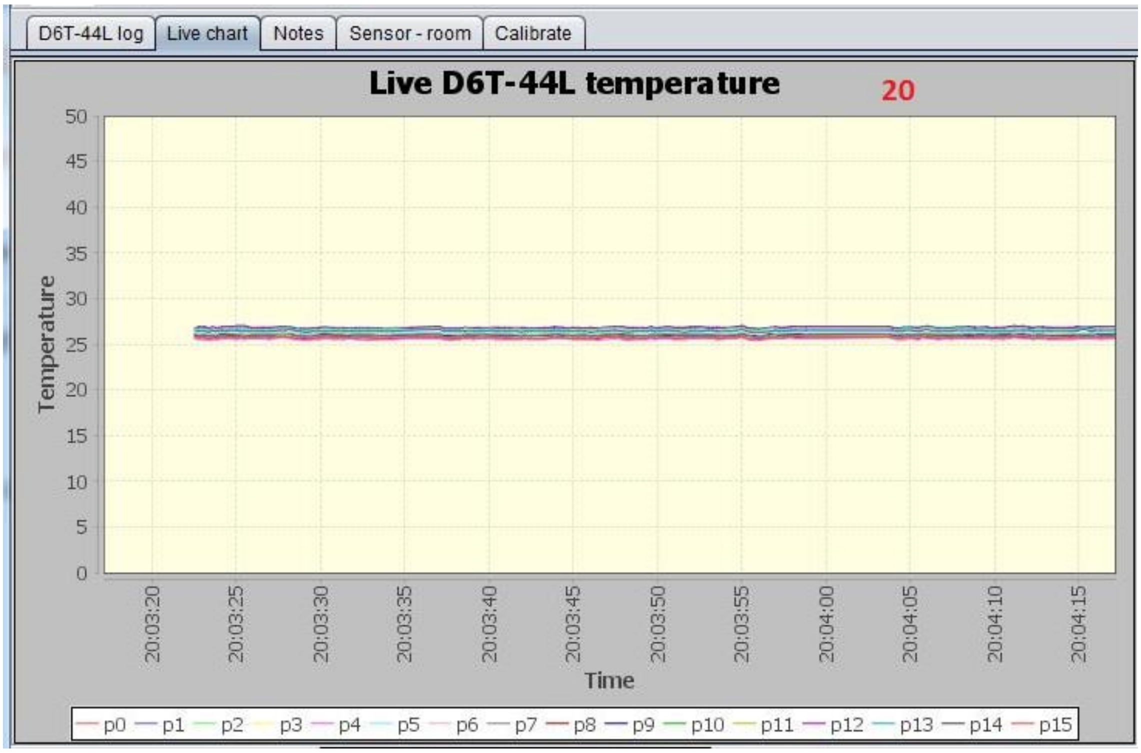
All data are live data and change continually.



2 Main page D6T-44l Thermal Temperature Reader.

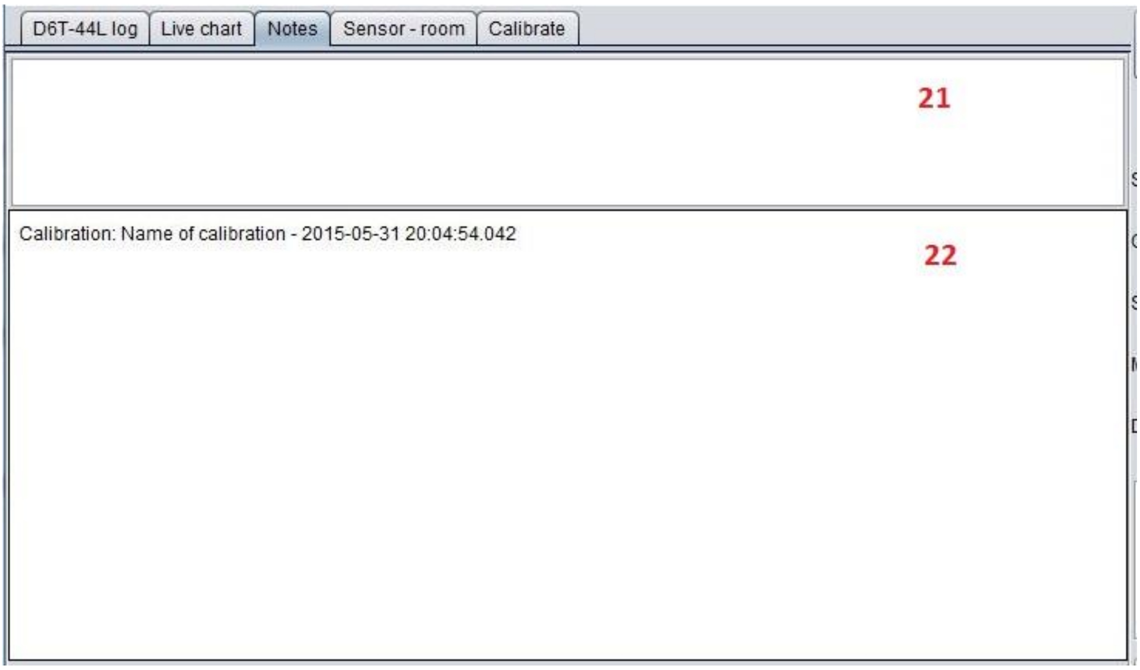
- 1 – connection to serial port
- 2 – average temperature for data in 5
- 3 and 4 – average temperature for each row/column, the dark grey is the highest average temperature
- 5 – real time temperature data from sensor
- 6 – calibrate button, calibration values are saved in 12, and save 12 and 13 (calibrated temperature values and calibrated name) to database
- 7 – is toggle to pause output readings from sensor 19
- 8 – clear log text area 19
- 9 – disconnect from serial port
- 10 – close the application

- 11 – temperature inside sensor PTAT
- 12 – snapshot of calibration data
- 13 – for naming calibration snapshot, and average value of calibrated value.
- 14 – toggle between calibrated values and pixel name
- 15 – creating test-set, saving temperature readings to database, creating test-set.
- 16 – button for taking screenshot of the application
- 17 – calculation results of Algorithm 2
- 18 – calculation results of Algorithm 1
- 19 – area for log data, timestamp and temperature data in short format, this data is saved to file



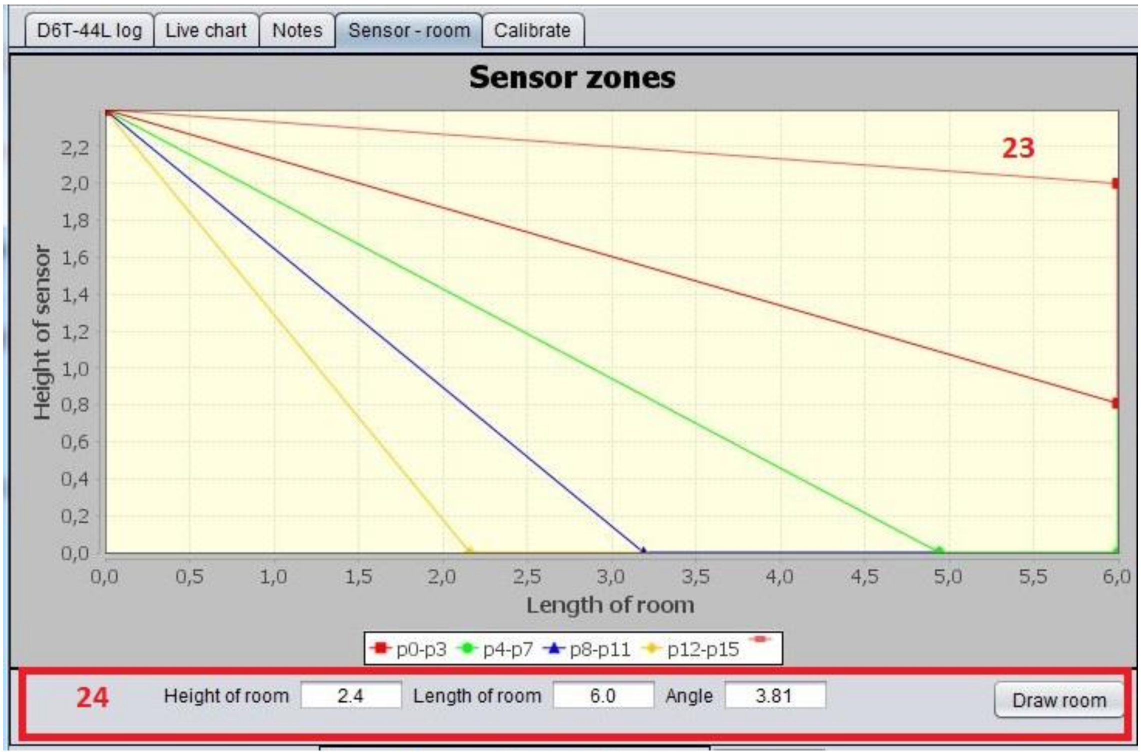
3 Live chart - D6T-44l Thermal Temperature Reader.

- 20 – live temperature data for the 60 last second, it is able to zoom in/out temperature data



4 Notes - page D6T-44l Thermal Temperature Reader .

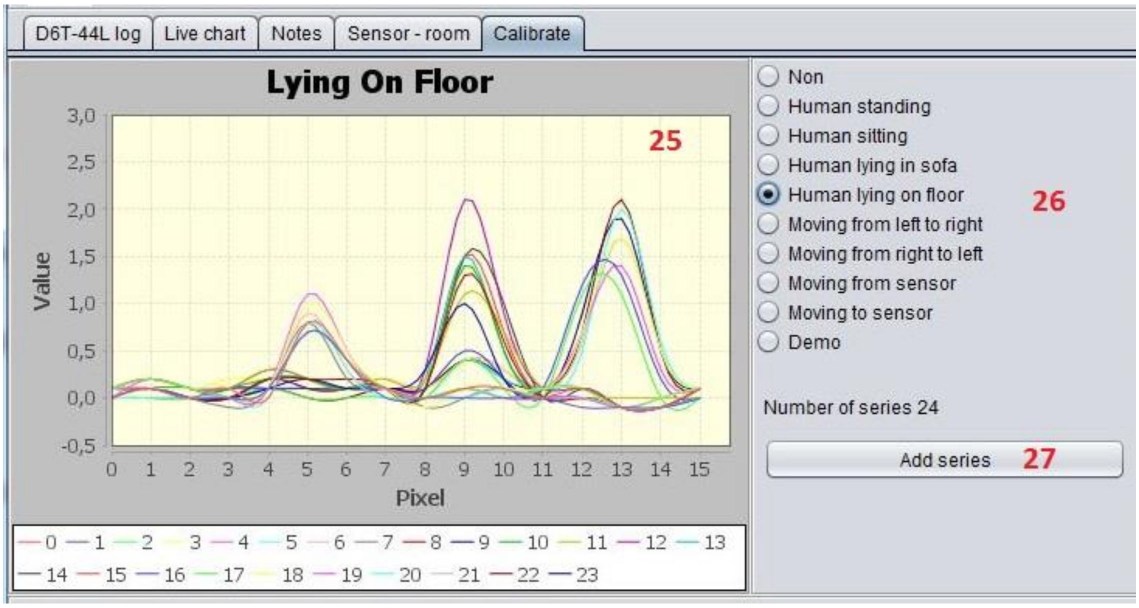
- 21 – text area to write comments, will be stored in database
- 22 – log area for many events in the application



5 Sesor/room - D6T-44l Thermal Temperature Reader.

- 23 – schematic illustration of sensor in a room 6.0 meter long and 2.4 height

- 24 – configuration of height and length of the room and angle of the sensor. Angles outside 0-90° range are not taken into account.



6 Calibarate - D6T-44l Thermal Temperature Reader.

- 25 – graph of live measured temperature data minus calibrated temperature data
- 26 – radio buttons for different predefined situations.
- 27 – adding temperature snapshot as new series for chosen situation.

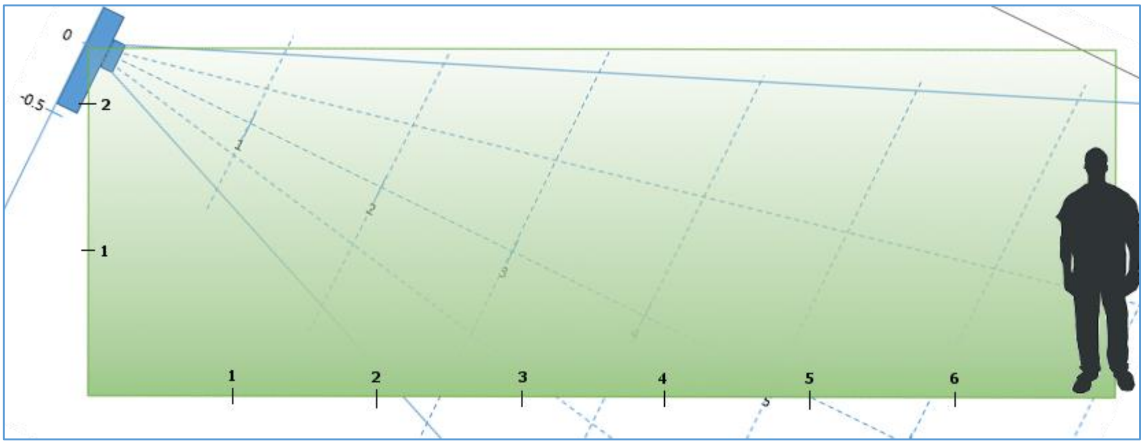
The application is missing a button to clear data in the graph. All data in the graph are saved to text files with the same name as radio buttons. Data can be cleared manually in the text file. It would also be nice to have *Calibrate* chart and *Sensor-room* chart side by side, especially when comparing algorithms.

Rarely database connection error: Java DB was preferred because it was included in NetBeans. Within the test there are some random error from the database, randomly, that says there were some error with the connection. A search on the internet showed that this was an error and the helpdesk at Derby DB suggested to install a newer version. But to struggle with this database would be time-consuming and using this function was avoided in this Java Swing application. Error:

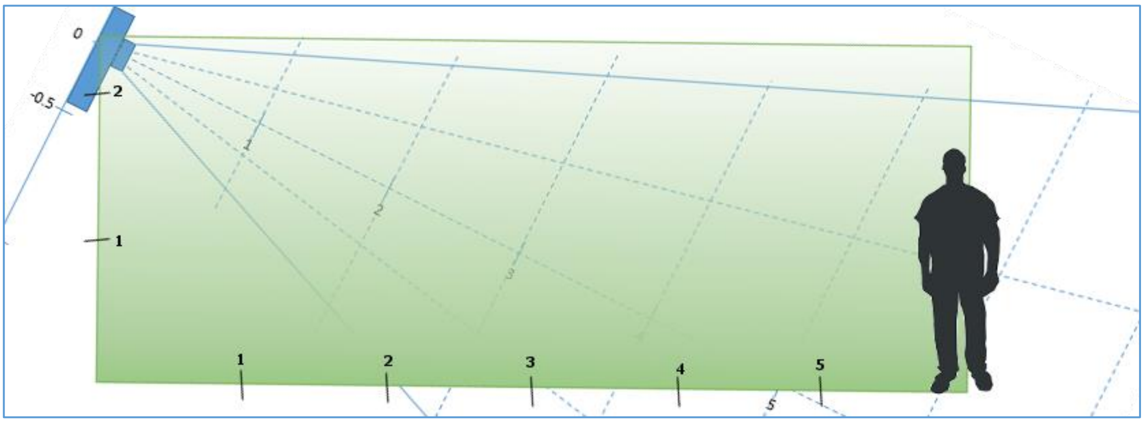
```
SEVERE: null
java.sql.SQLNonTransientConnectionException: No current connection.
```

Appendix 3 – Model of person in different rooms

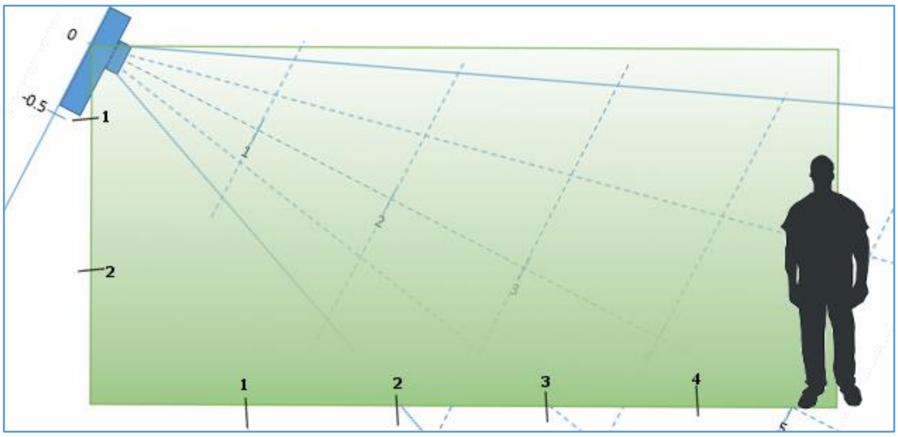
Room 2.4 meter x 7 meter and a person at 7 meter (sensor angle 26.2°)



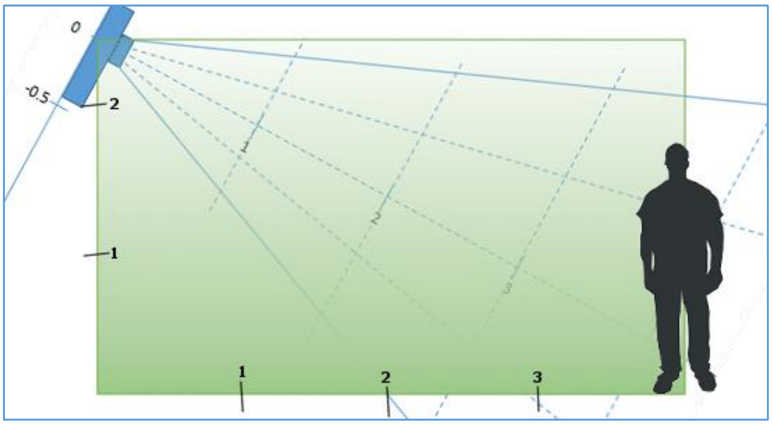
Room 2.4 meter x 6 meter and a person at 7 meter (sensor angle 26.2°)



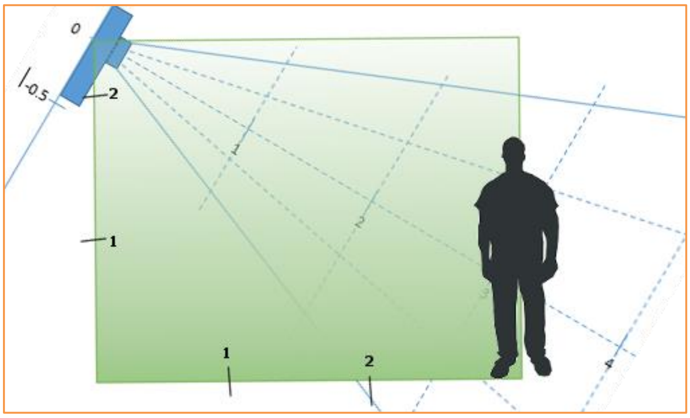
Room 2.4 meter x 5 meter and a person at 7 meter (sensor angle 26.2°)



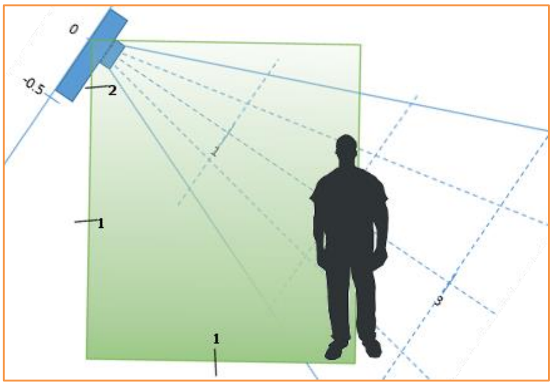
Room 2.4 meter x 4 meter and a person at 7 meter (sensor angle 26.2°)



Room 2.4 meter x 3 meter and a person at 7 meter (sensor angle 26.2°)



Room 2.4 meter x 2 meter and a person at 7 meter (sensor angle 26.2°)



Detection of human

Distance in m	**1**	**2**	**3**	**4**	**5**	**6**	**7**
p0-p3				x	x	x	x
p4-p7		x	x	x	x	x	x
p8-p11		x	x	x			
p12-p15	x	x					

Distance in m	**1**	**2**	**3**	**4**	**5**	**6**
p0-p3				x	x	x
p4-p7		x	x	x	x	x
p8-p11		x	x	x		
p12-p15	x	x				

Distance in m	**1**	**2**	**3**	**4**	**5**
p0-p3				x	x
p4-p7		x	x	x	x
p8-p11		x	x	x	
p12-p15	x	x			

Distance in m	**1**	**2**	**3**	**4**
p0-p3				x
p4-p7		x	x	x
p8-p11		x	x	
p12-p15	x	x		

Distance in m	**1**	**2**	**3**
p0-p3			x
p4-p7		x	x
p8-p11		x	x
p12-p15	x	x	

Distance in m	**1**	**2**
p0-p3		
p4-p7		x
p8-p11	x	x
p12-p15	x	x

Appendix 4 – Clearing and setting bits

Screenshots from

http://www.micahcarrick.com/tutorials/avr-microcontroller-tutorial/avr-c-programming.html

The following C code shows 3 ways in which a variable might be initialized to the decimal value of 15.

```
uint8_t a = 15;          /* decimal */
uint8_t b = 0x0F;        /* hexidecimal */
uint8_t c = 0b00001111;  /* binary */
```

To **set a bit**, OR the value with the bit value mask.

```
uint8_t a = 0x08;  /* 00001000 */
                   /* set bit 2 */
a |= (1<<2);       /* 00001100 */
a |= 0x04;         /* 00001100 */
```

To **clear a bit**, NOT the bit value mask so that the bit of interest is the only bit cleared, and then AND that with the value.

```
uint8_t a = 0x0F;  /* 00001111 */
                   /* clear bit 2 */
a &= ~(1<<2);      /* 00001011 */
a &= ~ 0x04;       /* 00001011 */
```

To **toggle a bit**, XOR the value with the bit value mask.

```
uint8_t a = 0x0F;  /* 00001111 */
                   /* toggle bit 2 */
a ^= (1<<2);       /* 00001011 */
a ^= (1<<2);       /* 00001111 */
a ^= 0x04;         /* 00001011 */
a ^= 0x04;         /* 00001111 */
```

Additional information about working with bit: Bit Twiddling Hacks see https://graphics.stanford.edu/~seander/bithacks.html

Appendix 5 - Literature study

CRC-8, cyclic redundancy check

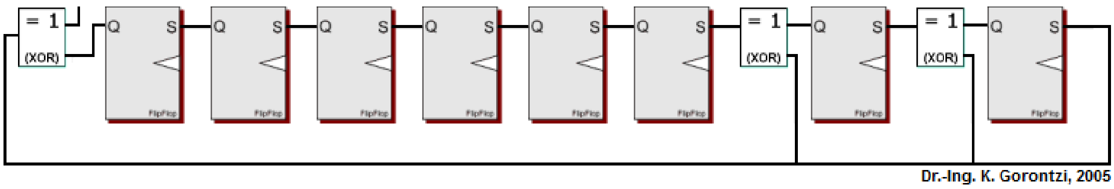
To assure the quality of transmission between the sensor and the microcontroller D6T-44L used CRC-8. CRC is an error check code/algorithm used to calculate a check sum based on input data and a polynomial. This is used for error detection during data transmission [32]. The same CRC algorithm is used to assure the quality of transmission between the microcontroller and client software on PC. PEC (Packet error check) used for D6T-44L is based on “SM bus” specification [29] [43] [32]. In method “*calc_crc*” from [29] the error checking are using CRC-8 and XOR with hex value 0x07.

This algorithm will loop through the byte bit for bit and works as follows: first it assigns the byte (unsigned char data) value to crc, then the byte value is left shift data by one. It then checks if the previous value (before left shift) of the byte is 1, and if it is true it inverts the left shifted byte value. This is done until looped through all 8 bit in the byte and the new crc is returned.

```
unsigned char calc_crc(unsigned char data) {// in - byte
    int index;                   // index of byte
    unsigned char crc;           // temp variable holding data

    for (index = 0; index < 8; index++) {// 8 is byte length
        crc = data;              // assign data to crc
        data <<= 1;              // left shift data by 1
        if (crc & 0x80) {        // if data(crc) is 1
            data ^= 0x07;        // invert data
        }
    }
    return data;  // return the new calculated crc value
}
```

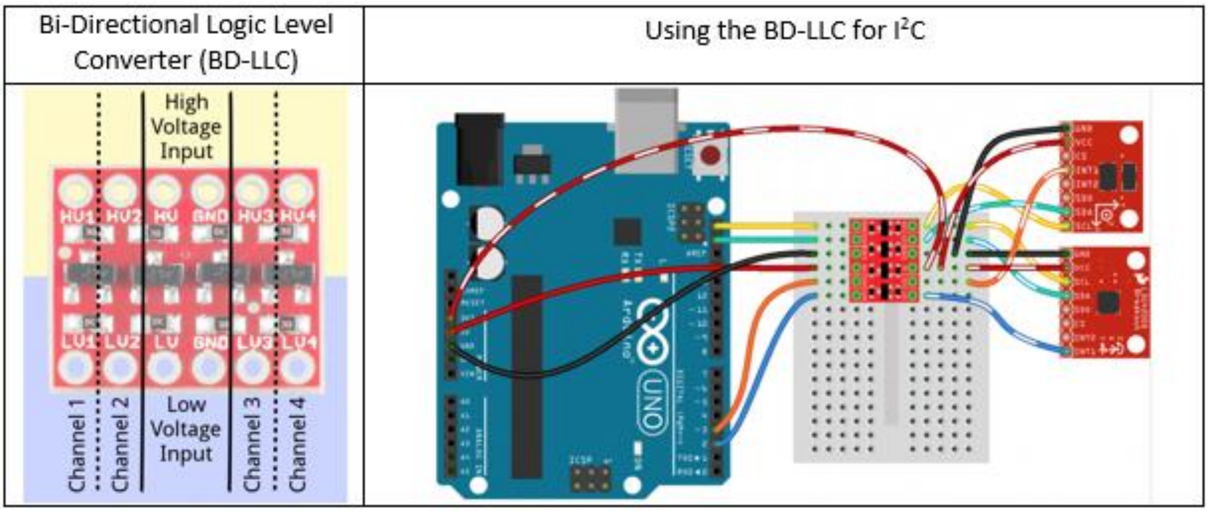
From one online CRC calculation (*Dr.-Ing. K. Gorontzi*) [31] we see that the hex value 0x07 gives us the polynomial **P(x) = x8+ x2+ x1+ x0** which corresponds to following LFSR (Linear Feedback Shift Register):



7 LFSP [31]

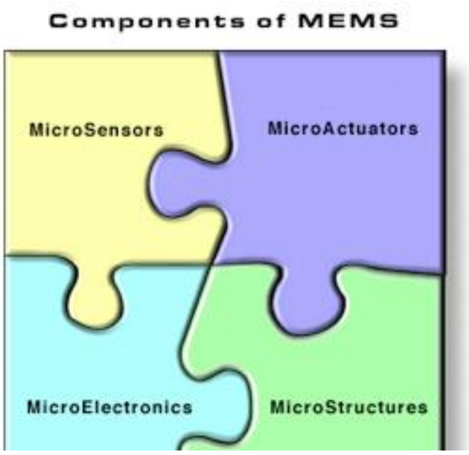
Logic Level Converter - Bi-Directional

A logic level converter, this converts higher voltage to lower voltage and lower voltage to higher voltage. SparkFuns Logic Level Converter [30] step down 5V to 3.3V and step up 3.3V to 5V. Each level converts 4 pins on the high side to 4 pins on the low side with 2 inputs and 2 outputs. The board needs to be powered from the two voltages sources used in the system. So with I^{2}C one can connect 2 I^{2}C pair to each Logic Level Converter, see "*8 SparkFun Logic Level Converter [26]*"



8 SparkFun Logic Level Converter [26]

Introduction of MEMS?



9 Components of MEMS [39]

Micro-Electro-Mechanical Systems, MEMS, is a technology which can be defined as having miniaturized mechanical and electro-mechanical elements, and can be very small in size. In Europe they are known as *microsystems technology* and in Japan they are known as *micro-machines* [44], [45]. They can contain Micro Sensors, Micro Actuators, Micro Electronics and Micro Structures. Its functionality is that it converts measured mechanical signals into electrical signals. Micro Sensors and Micro Actuators act as "transducers[3]" and convert temperature signals into electrical signals. In D6T-44L it converts surface temperature into a digital signals and sends it over I^2C to the receiver which can be a microcontroller.

Thermopile sensor vs Pyroelectric sensor

Both of these infra-red based sensors are detecting temperature and movement. The thermopile sensor continue to detect far-infrared rays while the pyroelectric models do not. Pyroelectric sensors detect the change of a signal (temperature variation) and cannot detect a stationary person while thermopile sensors can detect stationary persons by continually detecting infrared rays coming from an object. A pyroelectric sensor is also known as a PIR sensor.

Omron D6T MEMS Thermal Sensors

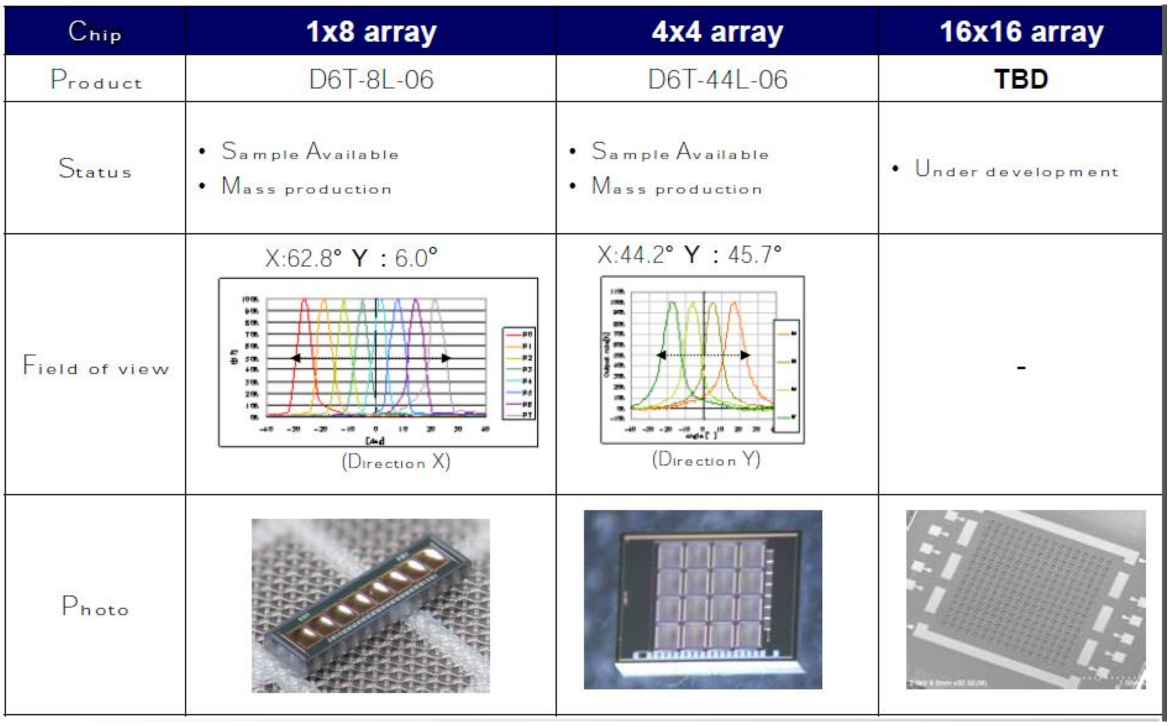
"*OMRON's unique MEMS and ASIC technology achieve a high SNR*."

Omron has 2 D6T thermal sensors, one 1 x 8 pixel and one 4 x 4. Heat radiated from the object goes through the silicon lens and is collected by a thermopile sensor. Thermopile sensors produce an electromotive force. Based on this electromotive force value and the internal temperature on the module, the analog circuit calculates the temperature. The value is outputted through I^2C. The sensor acquires only surface temperature and does not distinguish between humans, animals or appliances. Any object with same temperature will read as the same. Software is needed to detect behavior.

There is also a new chip 16x16 that is under development and testing (TBD). It will be for sale and released in fiscal 2015 [46] . One thing to notice is that Omron's x-direction, is what is commonly referred to as y-direction in the Cartesian coordinate system. So Omron's x-direction is actually the height of FOV and not the width. D6T-44L units have the default address 0x14 and 0x15 and it is not possible to change the slave address [29].

[3] a transducer convert energy from one form to another [44]

The table “Comparing Omron's D6T sensors.” is put together by combining Omron’s datasheet [27] and Omron’s article about the new 16 x16 sensor [46]. Reading speed is up to 4 frames per second which makes it possible to detect people moving at 1m/s. 1 m/s is 3.6 km/h and is considered normal walking.



Chip	1x8 array	4x4 array	16x16 array
Product	D6T-8L-06	D6T-44L-06	TBD
Status	• Sample Available • Mass production	• Sample Available • Mass production	• Under development
Field of view	X:62.8° Y : 6.0° (Direction X)	X:44.2° Y : 45.7° (Direction Y)	-
Photo			

Comparing Omron's D6T sensors.

Chip	1x8	4x4	16x16
Product	D6T-8L-6	D6T-44L-6	
Detectable target temperature range	5 to 50°C	5 to 50°C	5 to 50°C
Power supply voltage	4.5 to 5.5 VDC	4.5 to 5.5 VDC	2.7 to 5.5 VDC
Field of view	x 62.8° y 6.0°	x 44.2° y 45.7°	90 degrees minimum
Data transmission	I2C	I2C	SPI
Release date			Oct 2013 (test sample)
Sales date	2013	2013	2015
Temperature resolution (NETD)	0.14°C	0.14°C	0.15°C
External dimensions	W 14mm x L 18 mm x H 10.71 mm	W 14mm x L 18 mm x H 9.15 mm	W 20mm x L 37mm x H 10.7mm

Frames per second (data measurements and updates within 250ms)	4	4	4
			

Tabell 1 Comparing Omron's D6T sensors.

Sequence of sending

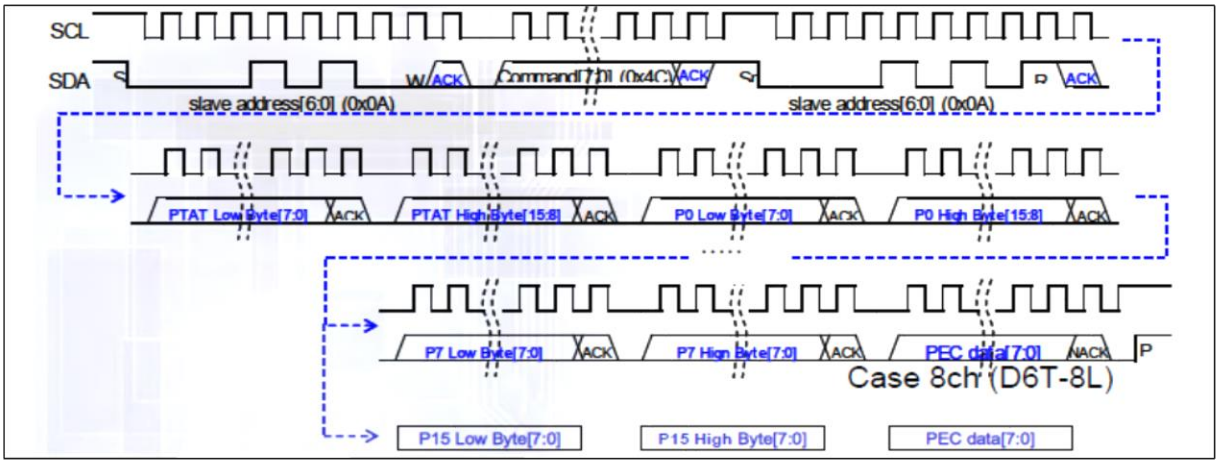
The sequence of sending consists of 2 different sequences. The first one tells the receiver which register it wants to read, in this case it is register `4C`. The command is: `S – 0x14 – 0x4C – Sr – 0x15` – then the reading will start and microcontroller will start receiving the temperatures from P0 to P15. After successful reading the first 16 temperatures the next command can start with `0x15`, and there is no need for the starting sequence `S – 0x14 – 0x4C`. The full command:

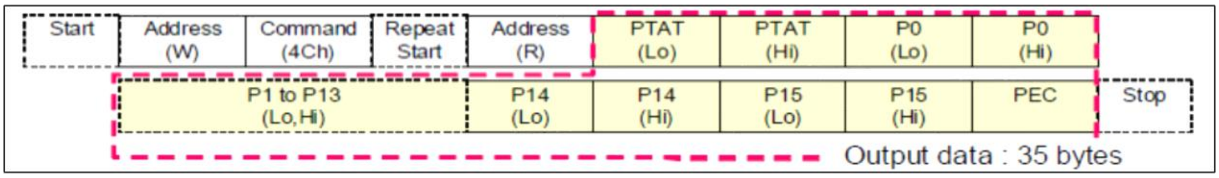
- `S – 0x14 – 0x4C – Sr – 0x15 –`
- `0x15 –`

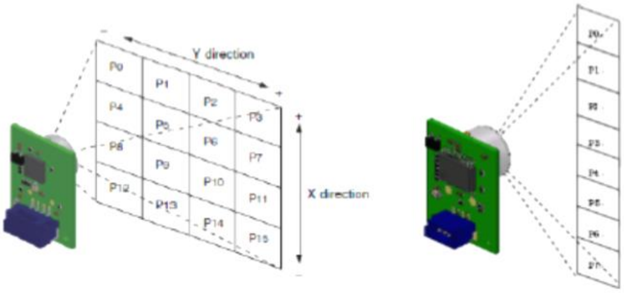
S – start condition, Sr – repeated start condition, P – stop condition, W – write, R – read, ACK – acknowledge replay, NACK – no-acknowledge replay.

When using the **MSP430g2553** microcontroller the **slave address is 0x0A** (`UCB0I2CSA = 0x0A`). The microcontroller uses `UCTR` and add write and read bit when sending the start or stop bit (`UCB0CTL1 |= UCTR` and `UCB0CTL1 &= ~UCTR`).

The following pictures are from the usage of the D6T-44L / D6T-8L thermal sensor [29]:





PTAT	The value of the reference temperature, inside the sensor module. Temperature data (PTAT&Pn) is 16bit-width, singed, 10 times value of degC. Example : 12.7 °C = 007Fh(127) , 25.8 °C = 0102Fh(258)
P0 to P15 (D6T-44L) P0 to P7 (D6T-8L)	Measured value. Pixel order is below. 
PEC	Packet error check code. Based on the “SM bus” specification.

Pattern recognition literature study

Definition of Machine learning?

Cited Coursera *“Machine learning is the science of getting computers to act without being explicitly programmed”* [47]. And cited Wikipedia “*Machine learning is a subfield of computer science [1] that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. [1] Machine learning explores the construction and study of algorithms that can learn from and make predictions on* *data.[2] Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions,[3]:2 rather than following strictly static program instructions.*"

Definition of Kalman filter.

The Kalman filter algorithm was developed by Rudolf Kalman in 1960. Kalman filter is an optimal algorithm estimator. It can use a series of measures over time. It uses a stream of noisy input data to output an optimal estimate. The filter is recursive and uses data in real-time and data is processed as it arrives. The filter is widely used in a time series analysis. One use Kalman filter to make the most secure decision based on uncertain information, such as measurements. It is a state estimate for a time-varying linear model system. When using the Kalman filter one wants to estimate the state of variable in a given process [48].

Definition of Markov chain

The Markov chain is named after the Russian mathematician Andrey Markov. It is a mathematical model that can be considered as a state machine where the next state only depend on the previous state, actually x+1 depend on x. The transition going from state to state as a chains based on the previous state. To solve a Markov chain one can use a transition matrix [49] [50]. It can also be used in a dynamic system, as time series and temperatures. "*Transition of probability*" is also an important factor and is defined [51] as "*The changes of state of the system are called transitions. The probabilities associated with various state changes are called transition probabilities. The process is characterized by a state space, a transition matrix describing the probabilities of particular transitions, and an initial state (or initial distribution) across the state space. By convention, we assume all possible states and transitions have been included in the definition of the process, so there is always a next state, and the process does not terminate.*"

Definition of Support Vector Machine (SVM) classifier and Support Vector Data Description (SVDD)

Definitions that help to define the meaning of SVM:

"*Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression*" [52]

"In machine learning, support vector machines (SVMs, also support vector networks[1]) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis " [53]