Expertise
and insight
for the future

Eetu Jääskeläinen

Easy to Install Smart Parking Solution

Metropolia University of Applied Sciences

Bachelor of Engineering

Electrical and Automation Engineering

Thesis

29 April 2021

Metropolia
University of Applied Sciences

Author Title Number of Pages Date	Eetu Jääskeläinen Easy to Install Smart Parking Solution 30 pages 29 April 2021
Degree	Bachelor of Engineering
Degree Programme	Electrical and Automation Engineering
Professional Major	Electronics
Instructor(s)	Heikki Valmu, Principal Lecturer

This study was conducted to a small startup company called LeViteZer Oy located at Nokia startup space and it was commissioned by Nokia. The challenge, that this thesis work tried to solve, was to design new sensors to detect cars entering and exiting a parking lot. These sensors should be easy to install and require no construction work in order to be cost effective. The solution was tested by upgrading two existing non-intelligent parking areas where the usage level (i.e. number of parked cars) was calculated and reported to an external server.

The new sensor devices that were created in this study, were modular and easy to install. They used Power over Ethernet (PoE) for power and communication. The sensors required some manual calibration to function, but all that could be done remotely when sensors were connected to Ethernet LAN network. The devices used mostly LiDAR technology to detect vehicles, but in two places thermal cameras were used instead. The enclosure for the sensors was done using 3D printers.

This study was a first step towards a more extensive solution that tries to predict a parking lot availability for a car before it arrives to the area. The sensors would provide data about the car lot usage and intelligent software in a back-end server would do predictions based on historical usage.

This study is limited to the design and implementation of the sensor devices. The electronics consisted of sensor modules and microcontrollers. The enclosure was 3D printed and the microcontroller was programed. The server side of the project is not part of this study. The original aim was to use wireless connections to the sensors, but due to time constraints a wired solution was used.

Live testing of the setup gave promising results in vehicle detection, but it also revealed multiple new problems. The problems the new solution had were that it might miss motorcycles, count humans as cars and count wrongly when a vehicle uses wrong lanes and uses exits as entrance or vice versa. The thermal camera solution had also promising results, but it should be tested in winter conditions to see whether it could distinguish for example a car with snow on top of it.

Keywords	Smart Parking, Internet of Things, LiDAR, Thermal Camera

Tekijä Otsikko Sivumäärä Aika	Eetu Jääskeläinen Helposti asennettava älykäs pysäköintiratkaisu 30 sivua 29.4.2021
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	sähkö- ja automaatiotekniikka
Ammatillinen pääaine	elektroniikka
Ohjaajat	yliopettaja Heikki Valmu

Insinöörityö toteutettiin startup yrityksessä LeViteZer Oy, joka sijaitsee Nokian aloitteleville yrityksille tarkoitetussa tilassa. Työn tavoitteena oli suunnitella uusi laite, joka tunnistaa autojen saapuvan ja lähtevän parkkihallista, laskee niiden määrän ja ilmoittaa saadun luvun palvelimelle. Jotta laitteen hinta voisi olla alhainen, sen asennus ei saa vaatia suurempia rakennustöitä tai työmaata. Ratkaisua kokeiltiin päivittämällä kaksi ei-älykästä parkkihallia, jotta niiden käyttöasteet (esim. parkissa olevien autojen määrä) voitaisiin laskea.

Työssä suunniteltiin ja toteutettiin käyttökohteiden tarpeisiin mukautettavia ja yksinkertaisesti asennettavia anturilaitteita. Ne käyttävät Power over Ethernet (PoE) -tekniikkaa kommunikoidakseen ja virran saamiseen. Laitteet vaativat manuaalista kalibrointia toimiakseen, mikä voidaan suorittaa Ethernet-lähiverkon avulla. Havaitakseen autoja ne käyttivät enimmäkseen LiDAR-teknologiaan perustuvia antureita, mutta käyttökohteiden takia muutamassa laitteessa käytettiin lämpökameroita. Kotelot anturilaitteille valmistettiin 3D-tulostamalla.

Insinöörityö on ensiaskelmia isommalle ratkaisulle, jossa pyritään ennakoimaan tyhjien parkkipaikkojen määrää, ennen kuin käyttäjä saapuu paikalle. Anturi-laitteet välittäisivät tietoa pysäköintihallien käytöstä ja älykäs ohjelmisto tekisi päätelmiä historiatietoa käyttäen.

Insinöörityö on rajattu anturilaitteiden suunnitteluun ja toteutukseen. Laitteiden elektroniikka koostui anturimoduuleista ja mikrokontrollerista. Työssä laitteiden kotelot 3D-tulostettiin, mikrokontrollerit ohjelmoitiin, laitteet asennettiin ja testattiin. Älykkään ennakoivan ohjelman toteutus ei kuulu työhön. Alkuperäinen suunnitelma oli luoda laitteista langattomia, mutta aikarajoitusten takia käytettiin langallista toteutusta.

Toteutuksessa päästiin lupaaviin tuloksiin ajoneuvojen havainnoimisessa, mutta myös huomattiin uusia ongelmia. Moottoripyörät saattoivat ohittaa täysin anturin, ajokaistalla kävelevät ihmiset saatettiin laskea autoiksi ja väärää kaistaa käyttävät autot saatettiin laskea lähteviksi, vaikka olisivat juuri tulossa ja päinvastoin. Lämpökameralliset laitteet antoivat myös lupaavia tuloksia, mutta tarvitsevat lisää aikaa ääriolosuhteissa testaamiseen.

Avainsanat	älykäs pysäköinti, asioiden internet, LiDAR, lämpökamera

Contents

List of Abbreviations

3D	Three Dimensional
AI	Artificial Intelligence
CoAP	Constrained Application Protocol
EDA	Electronic Design Automation
EV	Electric Vehicle
GPIO	General Purpose Input/Output
I^2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IoT	Internet Of Things
LAN	Local Area Network
LiDAR	Light Detection and Ranging
LVTTL	Low Voltage Transistor to Transistor Logic
LWIR	Long Wavelength Infrared
MIPI	Mobile Industry Processor Interface
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
OTA	Over-the-Air programming
PCB	Printed Circuit Board
PoE	Power over Ethernet
RAM	Random-access memory
SPI	Serial Peripheral Interface
SRAM	Static random-access memory
ToF	Time of Flight
UART	Universal Asynchronous Receiver-Transmitter

1 Introduction

Internet of Things (IoT) is a generic description of networks that include different kind of sensors and actuators connected to application servers through Internet. Sensors measure various physical parameters such as temperature, wind speed, pressure, and status of machines (vibration etc.), whether a door is open or closed and so on. Actuators are devices that can act based on remote commands. These can be used to open valves, turn on and off lights and basically anything that can be operated remotely.

Application servers include dedicated software to collect the data from sensors, process and analyze the data and finally visualize it to the user. This application software may include Artificial Intelligence (AI) or Machine Learning (ML) based algorithms to provide more advanced processing capabilities to the collected data. Based on the processed data the software may send automatically commands back to sensors and actuators.

Sensors and actuators can be connected to the Internet through wired or wireless communication technologies, though wireless solutions have gained popularity due to simpler installation process. Cyber security is one of the main concerns in IoT systems and building a good enough security to small sensors and actuators is a challenge. For this reason, many times data is collected locally to so called edge device which may preprocess the data and provide the needed security features before the data is send to application servers through Internet.

This thesis work concerns of a new IoT application and its development. The target of this application is to calculate the number of cars in a parking garage. The same application could potentially also be used to calculate how many people are present in different physical places such as meeting rooms in an office environment. These calculations were done based on the information collected from Light Detection and Ranging (LiDAR) sensors and a few infrared cameras. Because the collected information does not directly provide the needed output, i.e., number of cars or people in a certain place, the data needs to be processed with a dedicated software.

This study was conducted to a small startup company Levitezer. The company provides solutions and services for film industry, virtual reality applications and monitoring solutions on how people are moving inside buildings etc. As a small company, it is also investigating new solutions that utilize cameras as described in this study. LeViteZer is located at Nokia Startup Space, 30 minutes from Helsinki airport.

At the time this thesis study was started, the smart parking solution was not yet completely operational. The solution had been initially tested and one surveillance camera had been installed close to the parking building. The target of this setup was to calculate the number of parked vehicles in the building.

In order to improve the solution, following things needed to be improved:

- The system monitored only one entrance and exit point of the building though there were two entrance and exit points.
- The new system should be modular, and installation of the used sensors should not require any laborious construction work.
- The existing camera had been placed in the neighboring building located in a different property. This caused problems with maintenance because one needed a permission for all actions done to the camera.
- The used camera was a regular camera capturing high quality images, and this caused potential privacy issues as there was a possibility to identify persons from the images.
- The load of the server responsible for analyzing the data was too high because of the vast amount of data generated by the camera. The analysis was based on image recognition, i.e., finding a car from the captured video stream.
- The server was not originally designed to be operational 24/7 and maintenance breaks occurred every now and then causing also breaks in the analysis of the parking place usage.
- There were also some challenges in the accuracy of the solution as it missed occasionally some cars entering or leaving the building.

The target of this study was to design, implement and test following matters to improve the existing solution:

- Design and implement an alternative device using LiDAR and later thermal sensors to replace the existing camera. This included the local processing of the raw data, so that the device would send to the network only the information about the number of cars that had passed the device.

- Test the solution in live environment to see how well it was able to detect cars entering and leaving the car park.
- Have the design blend in with the surroundings.

The study was conducted in the following way. First the requirements for the new solution were collected and detailed specifications were drawn. Then the layout of the solution was planned and required components were acquired. After that the solution was built and tested to see how well it met the requirements.

This thesis has been divided into 5 sections. The first section introduces the problem that needed to be solved and how the problem was solved. The second section describes how objects can be detected in smart parking, what kind of sensors are used and what kind of sensors were used in this study. The third section includes the project specifications, and the fourth section describes how the solution was implemented and tested. The fifth section includes conclusions of this study.

2 Smart Parking and Object Detection

This section describes first the basic network structure of a typical Internet of Things (IoT) solution and the main application areas where it is used. After that it then explains in more detail a solution called Smart Parking and how different sensors can be used in such environment. Finally, it describes the sensors used in this study.

2.1 Internet of Things

Internet of Things is generic term to describe networks that include various types of sensors and actuator connected to application servers through Internet. There are various applications and use cases and some of them are shown in Figure 1.

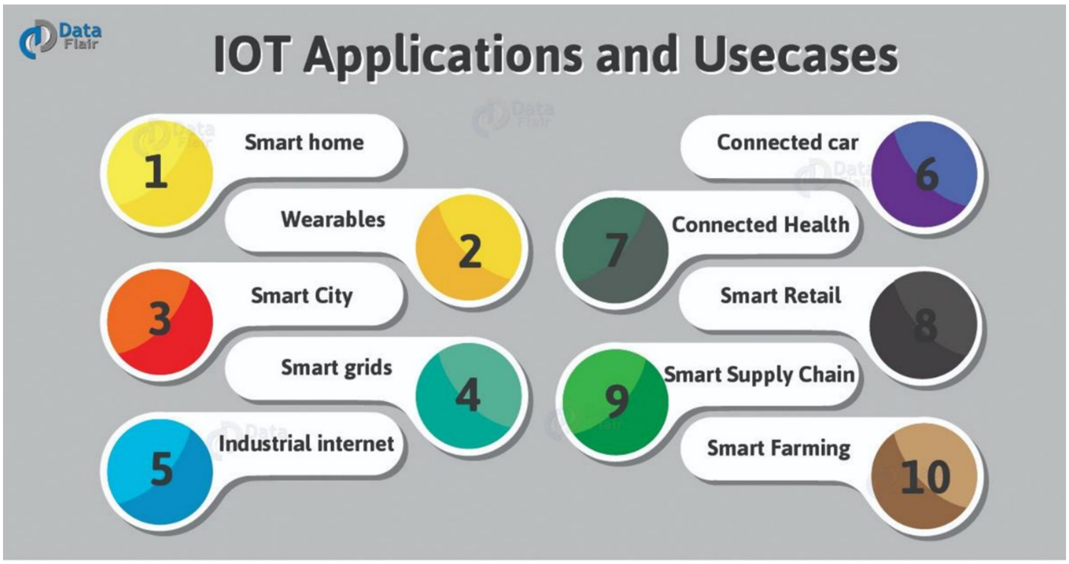

Figure 1. IoT applications and use cases [1].

Smart home and wearables are the two use cases where the number of connected devices and users are increasing with highest speed [1]. In addition to these consumer related use cases, several more challenging professional types of solutions are spreading in various areas. These include smart cities, smart grids, and industrial internet to name a few.

A typical IoT network architecture is shown in Figure 2. The lowest layer includes sensors and actuators that measures the needed physical parameters and activate physical devices based on the commands send from the network. In many cases these devices may also do some raw data processing to reduce the amount of data to be sent to the network and they may also have inbuilt communication interfaces. In those cases, these devices are often called smart devices.

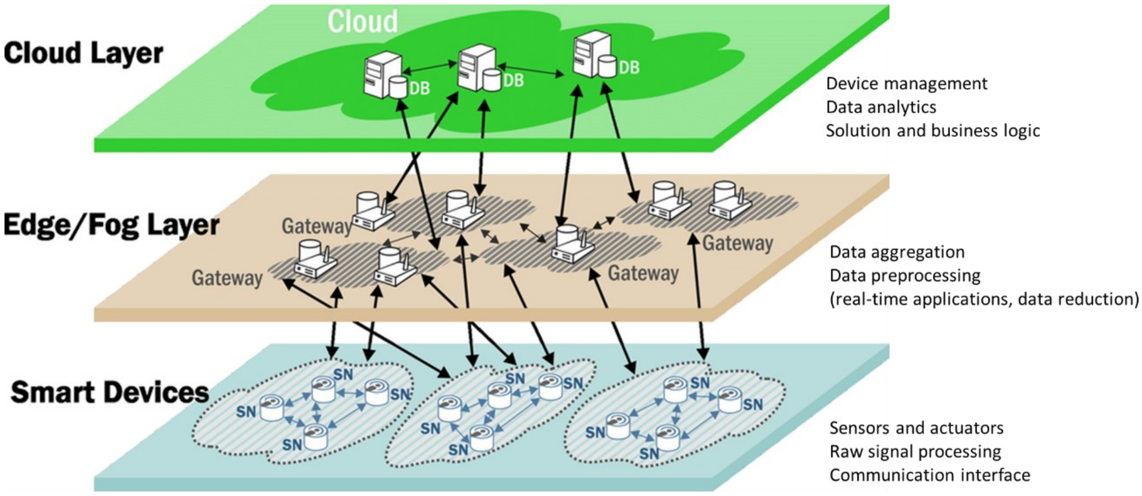

Figure 2. A typical IoT network architecture [2].

The second layer edge or fog layer, provides needed data aggregation and preprocessing, which is especially needed for time critical applications. Typically, devices in this layer also provide the needed security features before the data is sent to Internet. The topmost layer includes those application servers providing data analytics and solution and business-related logic. Also, device management, which is responsible for device provisioning of new devices, is included in this layer.

Figure 3 below shows how the collected data is analyzed through the technology stack and how the intelligence of the data increases when it moves from bottom to top of the stack. At the same time, the volume of the data decreases.

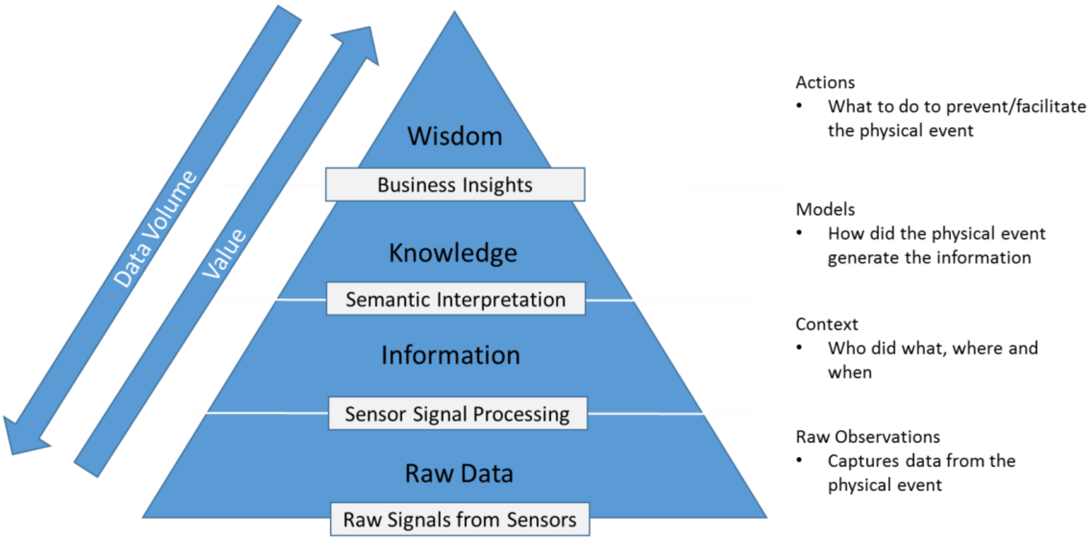

Figure 3. Technology Stack for IoT Analytics [3].

The sensors and actuators (or smart devices) have often very limited power to use, especially if they are operating with batteries. This sets strict requirements for the communication protocols, which should be very bandwidth efficient and at the same time meet the high security requirements. Figure 4 shows a comparison between typical IoT communication protocol stack and Web application protocol stack. As seen in the figure 4, there are often differences in all layers, though IPv6 and UDP are used in both applications.

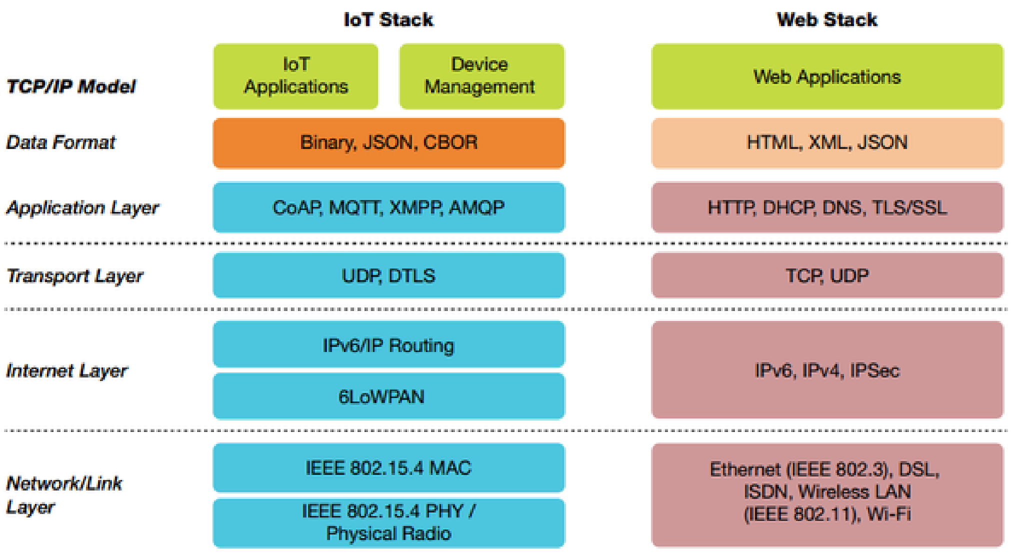

Figure 4. A typical communication protocol stack in IoT [4].

The biggest differences are in the messaging protocols in the application layer, where many IoT applications are using Constrained Application Protocol (CoAP), Message Queuing Telemetry Transport (MQTT) or similar dedicated protocols suitable for low-speed communication channels.

2.2 Smart Parking

A smart city is a concept that aims to optimize city functions by using sensors and applications. These sensors collect data to help manage assets and services of the city efficiently. With the collected data a wide variation of action can be done, for example traffic lights can be automatically controlled to reduce congestion or warnings can be issued for severe weather or other natural incidents etc. [5.]

Smart parking is a part of smart city and provides a solution to prevent unnecessary driving, when for example a driver is searching for non-occupied parking space. A lot of time can be saved when they are guided directly to an empty spot. Smart direction indicators or signs telling how many empty spaces there are left are becoming more and more common to modern parking systems. Also, more intelligent systems have been built and tested as shown in Figure 5.

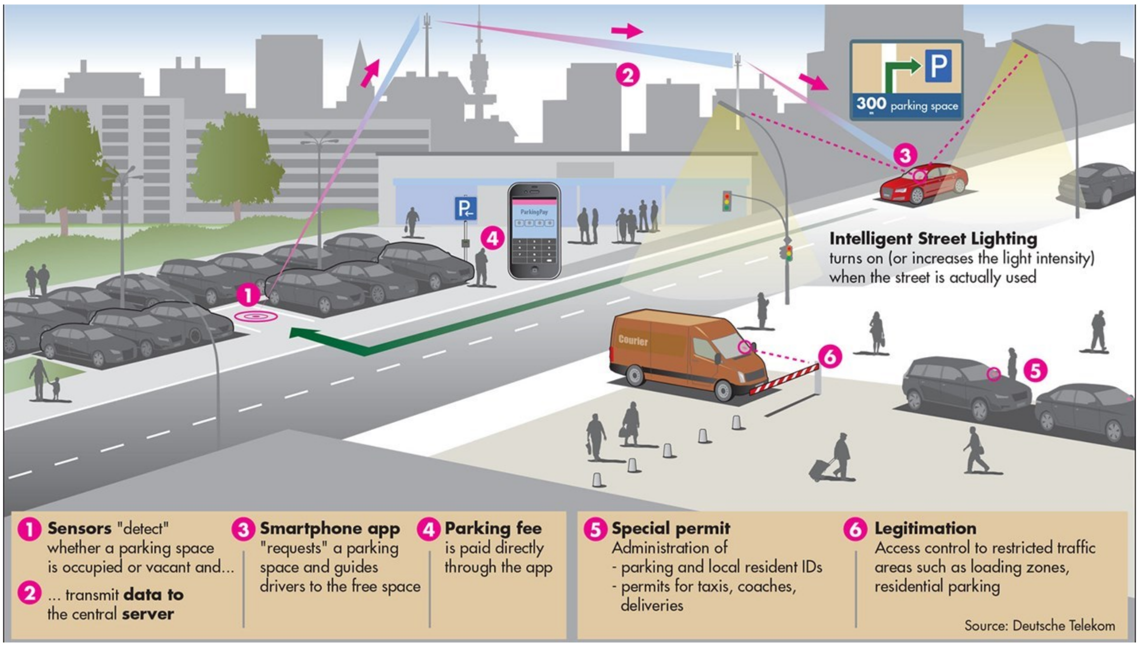

Figure 5. Smart parking solution [6].

The smart parking system illustrated in Figure 5 includes sensor to detect free parking spaces and this data is sent to a central application server. Users can have access to this information through their smartphone application, which will guide them to right place. The solution includes also additional features such as paying parking fees with mobile app, special permits for residents etc. and access control for restricted areas. [6.]

There are several technical solutions to detect whether a parking place is occupied or not and the most used are listed below:

- Inductive loop detectors
- Infrared sensors (passive and active)
- Radar/LiDAR
- Ultrasonic sensors
- Camera

Inductive loops measure the change in magnetic field, and they can detect this change when a car moves near them. These detectors work well and are reliable for most cars. However, SUV or others that have high ground clearance and electric vehicles (EV) made with less ferrous materials are harder to detect. [7.] Installing these types of sensors usually require modifications to the pavement and that can be expensive. Figure 6 shows a typical inductive loop installation, these can also be used with traffic lights to automatically change lights or with two loops a speed camera can measure the speed of a vehicle.

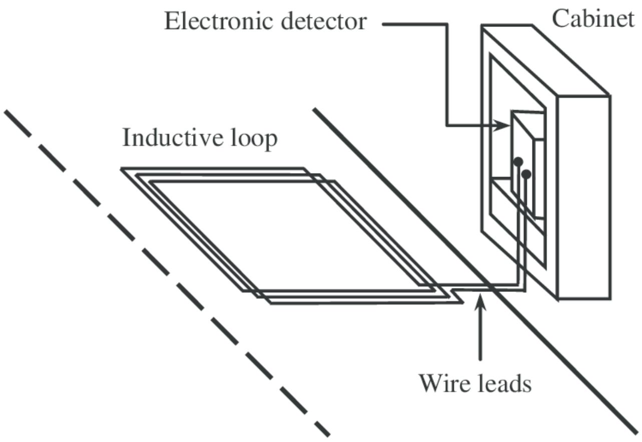

Figure 6. Inductive loop system [8].

Infrared sensors can be either passive or active. Passive sensors measure infrared light radiation and can detect a difference in empty and occupied parking spot but are affected by sunlight and other infrared light sources. Active sensors have their own emitter that emits infrared light and a sensor that measure reflected light levels. This approach has better immunity to ambient noise sources. The reflected infrared light changes depending on whether there is a car or not in the sensors field of view. There exist also active sensors that can measure a distance to an object, by calculating the time it takes for the infrared light to bounce back and multiplying it by speed of light. [7.] Typically, all these sensors work in a reliable manner, but each parking space requires its own sensor. Figure 7 shows the difference of passive and active IR sensors. The passive sensor requires an infrared light source to work and the active sensor emits its own infrared light.

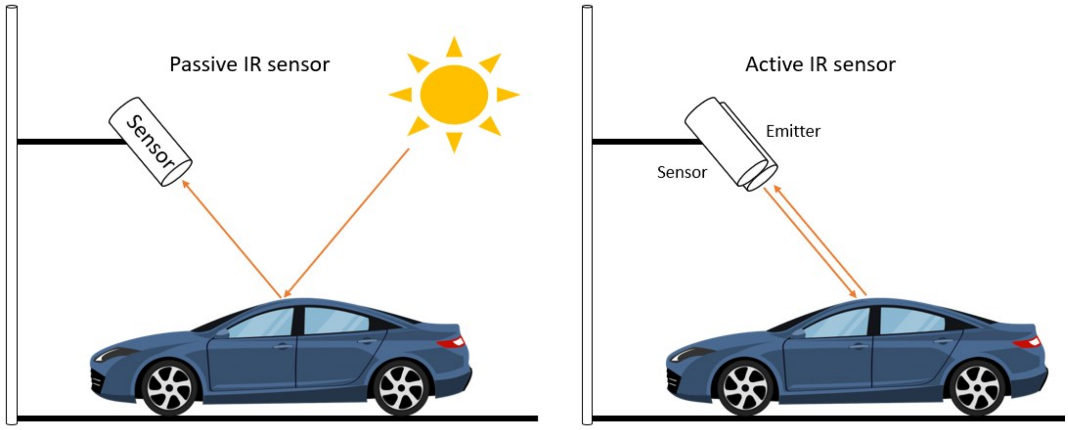

Figure 7. Difference of active and passive IR sensors.

Radars and LiDARs operate with the same principle as active infrared sensors, but they use different frequency range for the transmission. These sensors can use modulation wave as the emitted signal and measure a phase difference of the received signal to accurately calculate distance. Overhead Radar and LiDAR sensors can reliably cover 4 to 5 parking spots, but then decrease in accuracy as the reflections of the emitted signals become more scattered. Overhead radar solution is shown on figure 8.

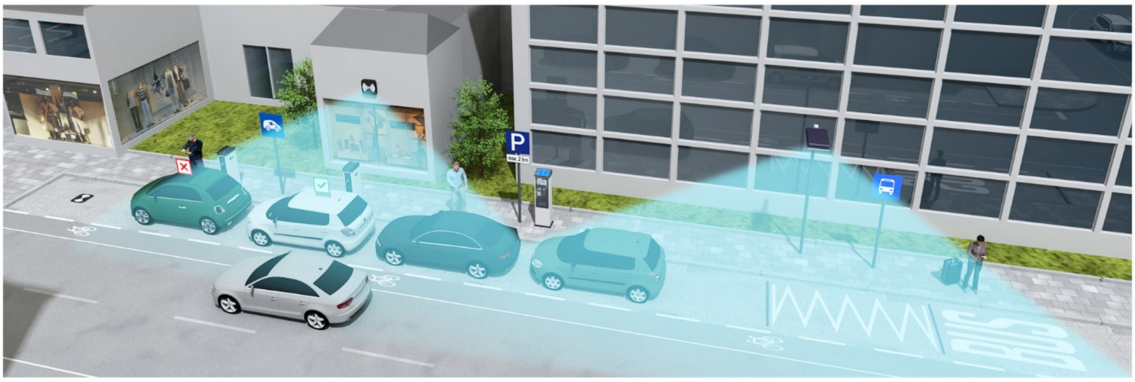

Figure 8. Overhead radar solution [9].

Ultrasonic sensors also measure the bounce back time of emitted signal, but they operate with ultrasound waves instead of light waves. Parking assistance systems in vehicles, which helps drivers to detect if the car is too close to an object like a wall, also use ultrasonic sensors. This solution also requires one sensor per parking space. Figure 9 shows how ultrasonic sensors work. First a sound wave is produced and then reflected echo is measured. A distance can be calculated from their time difference.

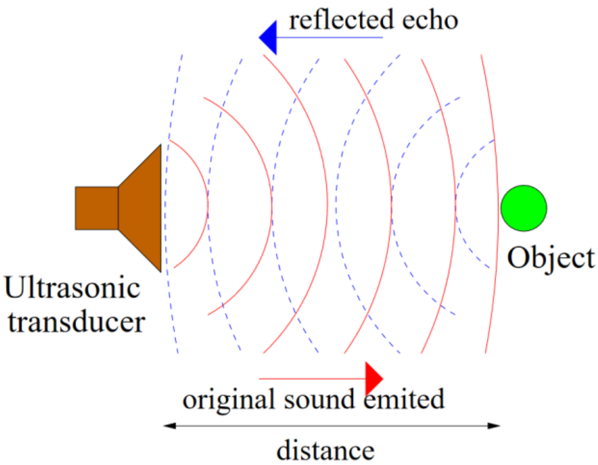

Figure 9. Illustration of the operation of ultrasonic sensor [10].

Cameras, operating either at visual light or infrared frequencies, can be used to detect the presence of a car in a parking place with the help of image recognition. The main difference compared to those other methods mentioned is that cameras can use wide angle optics and detect several parking spots at the same time. A drawback of this solution is that weather conditions may affect the performance. [11.] Figure 10 illustrates how artificial intelligence can be used with a camera to detect vehicles in a parking lot. Red rectangles indicate occupied spots and green rectangles are empty. Purple outlines show where the software has detected a vehicle.

Figure 10. CCTV camera with artificial intelligence software detecting empty spots in a parking lot. [12]

With smart parking system, drivers can be guided to empty spots and parking fees billed automatically, avoiding user from searching a parking meter. Security can also be improved with artificial intelligence in these systems. Detection software for malicious activity or automatic gate opening for restricted parking lots can be achieved with proper software and sensors. [13]

2.3 Sensors Used in This Study

In this study active LiDAR sensors, that are measures distance to a target, were used. Also, relatively affordable thermal cameras were installed to a few spots to diversify the solution. These sensors were chosen for their price and for their capabilities.

The used LiDAR sensor was TFMini Plus. This sensor is a time of flight (ToF) LiDAR sensor capable of measuring the distance to an object from 10 centimeters to 12 meters [14]. Figure 11 shows the used sensor.

Figure 11. LiDAR sensor TFMini Plus [15].

The LiDAR sensor had following main features:

- Operating Range: 0.1m~12m
- Accuracy: ±5cm@(0.1-6m), ±1%@(6m-12m)
- Distance resolution: 5mm
- Frame rate: 1-1000Hz(adjustable)
- Ambient light immunity: 70klux
- Operating temperature: -20℃~60℃
- Enclosure rating: IP65
- Light source: LED
- Central wavelength: 850nm
- FOV: 3.6°
- Supply voltage: 5V±0.5V
- Power consumption: 550mW
- Communication level: Low Voltage Transistor to Transistor Logic (LVTTL) (3.3V)
- Communication type: Universal Asynchronous Receiver-Transmitter (UART)

TFMini Plus is improved version of TFMini series of LiDAR sensors. This version has IP65 rated enclosure and can work in down to -20°C cold environments. As these sensors are placed inside an enclosure with other electronics that generate small amount of heat and to a ceiling of a parking garage they were expected to survive through Finnish winter. The sensor module uses ToF principle and emits modulation wave of near-infrared light. The modulation wave is then reflected from a surface back to the sensor and time of flight is obtained by the phase difference of measured signal, as shown in figure 12. These LiDAR sensors use UART communication interface and send distance measurements as serial data packets. This allows the use of devices that have minimal processing power as the time critical calculations are done in the sensor module.

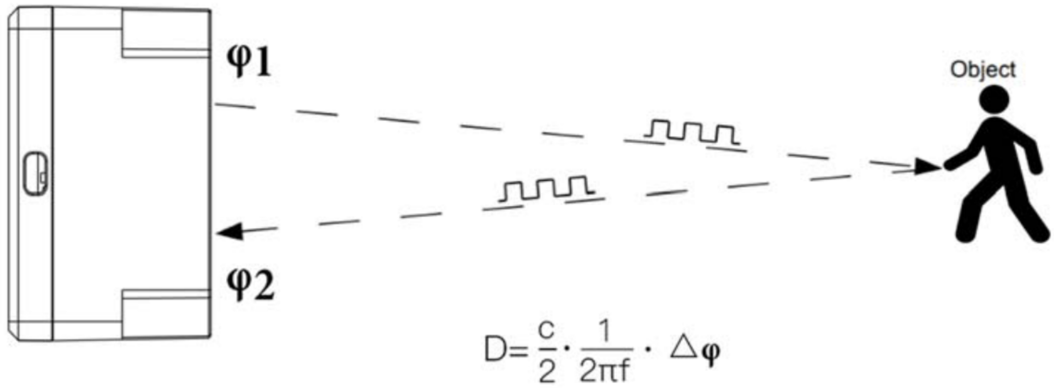

Figure 12. TFMini ToF measurement technique. [14]

The thermal camera module used in this study was FLIR Dev kit which has Lepton Long Wavelength Infrared (LWIR) sensor and breakout board for the sensor. The module acts as a sort of camera and has a resolution of 80×60 pixels. The camera body is smaller than a dime and captures infrared radiation input in its nominal response wavelength band and outputs a uniform thermal image. [16]. Figure 13 shows a picture of the used thermal camera module.

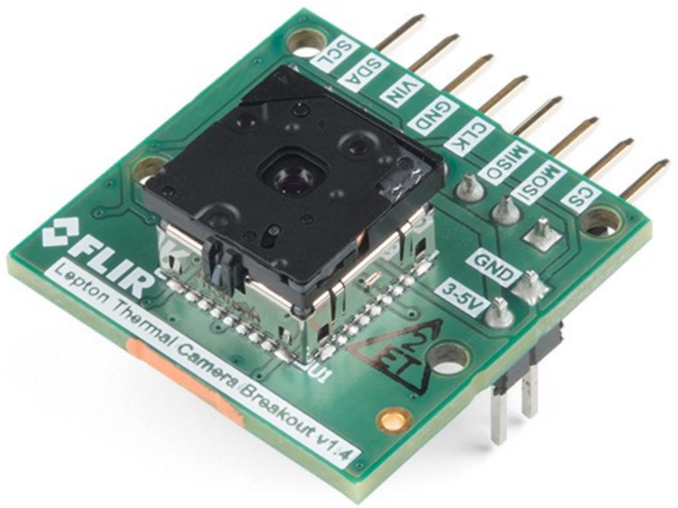

Figure 13. FLIR Lepton Thermal camera sensor module. [17]

The thermal camera module has following main features:

- LWIR sensor, wavelength 8 to 14μm
- 51° Horizontal FOV, 63.5° diagonal FOV
- 80×60 active pixels
- Thermal sensitivity <50mK
- Mobile Industry Processor Interface (MIPI) and Serial Peripheral Interface (SPI) video interfaces
- Two-wire Inter-Integrated Circuit (I^2C) -like serial-control interface
- Fast time to image (< 0.5 sec)
- Power consumption: 150mW

This thermal camera captures infrared radiation with nominal wavelength of 8 to 14microns and outputs thermal images. Some modules can provide radiometry data to calculate accurate temperatures. The module contains a shutter and does automatic flat field correction which calibrates the sensor and reduces noise from the image. Thermal cameras with over 9Hz framerate are export-controlled by the U.S. government causing the sensors to have 8.7Hz framerate [18]. The camera stream is 26Hz video, but frames are repeated 3 times. The sensor also provides telemetry data to compensate the thermal image, but is not repeated like the image frames, and must be read with each frame for accurate temperature measurement. This causes a lot of communication between the sensor and a connected device.

It was tested during this study that is not possible to identify a person from the thermal image when the person is more than a few dozen centimeters away from the sensor, as it has too few pixels.

The usage of these sensors in this study is described in section four.

3 Project Setup

This section describes first the status of the smart parking system when this study was started. Then it shows the new solution architecture which was designed and implemented during the study. Finally, it describes the requirements set for the new LiDAR device used to monitor cars entering and leaving the car park.

3.1 Preliminary Setup for Car Park Usage Monitoring

At the beginning of this study, there was already a preliminary solution for observing the cars entering and leaving a car park. There was only one regular video camera monitoring one entrance point of the parking garage. The analysis was based on image recognition, i.e., finding a car leaving or entering the building from the video stream captured by the camera.

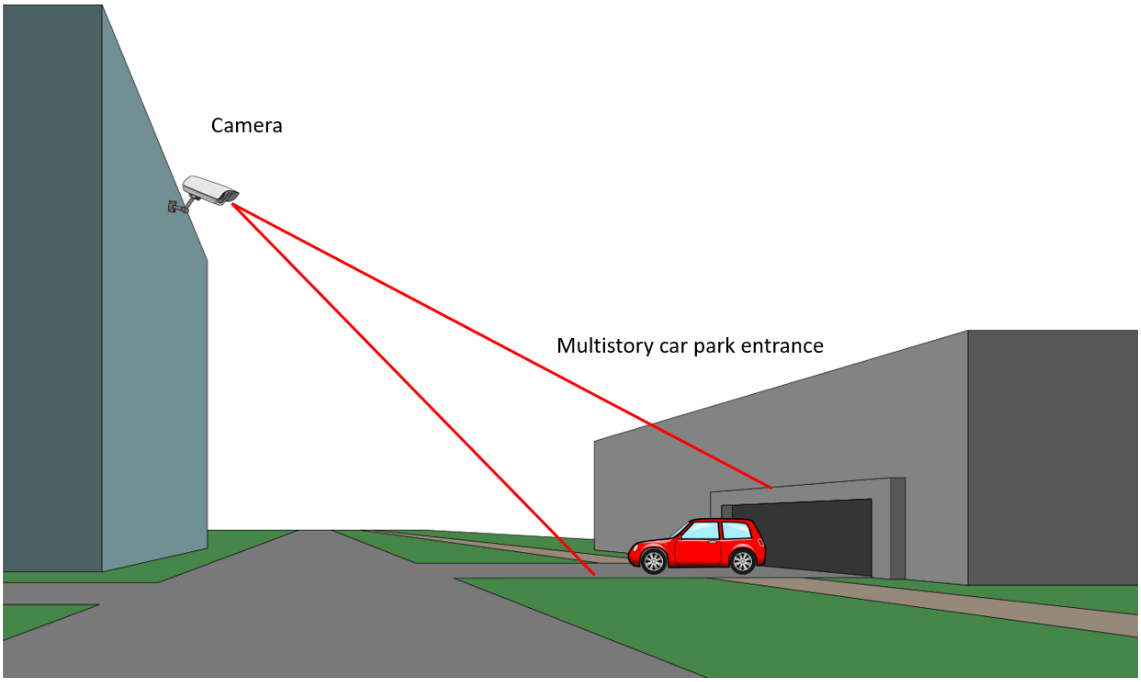

Figure 14. Initial solution to car park usage monitoring.

Figure 14 shows this preliminary solution. The video stream was sent from the camera with a cable connection directly to a local server. This server included the image recognition software and an algorithm to calculate the number of cars in the garage.

The solution worked reasonably well as it was able to detect most cars from the video stream. However, the solution had several shortages as described in the Introduction section. Some of the major shortages were related to the vast amount of data generated by the video camera and privacy concerns with image recognition software. For these reasons, a different kind of sensors was needed and in this study LiDAR sensors were mainly used for that purpose.

3.2 New Solution Architecture

This subsection describes the new solution architecture on a general level and what elements were used to build it. The new solution is shown in Figure 15.

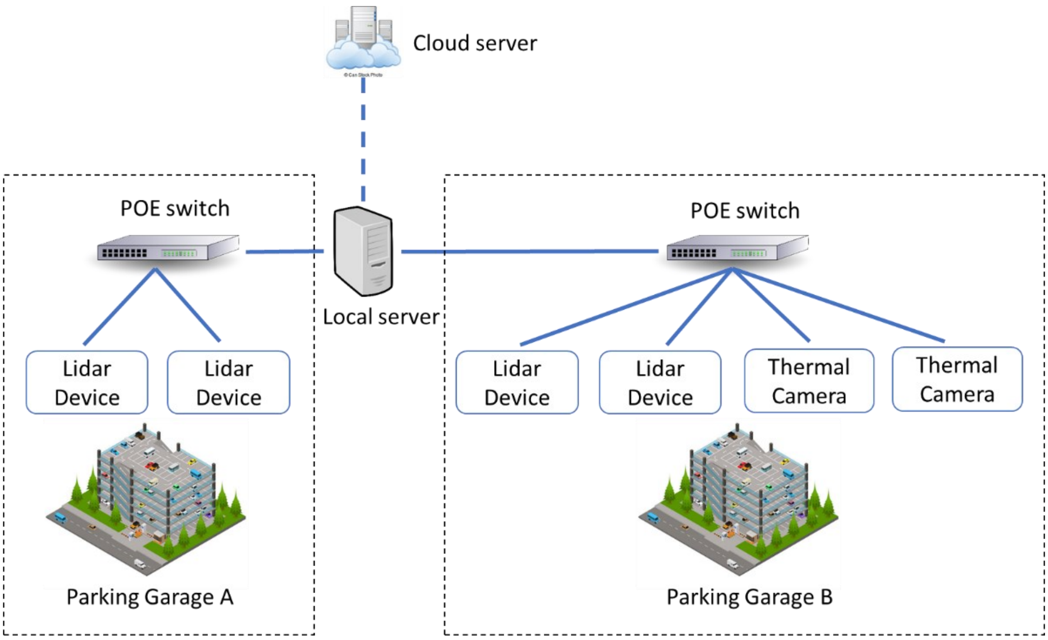

Figure 15. New solution architecture.

In the new solution one additional parking garage was included. Two LiDAR devices were installed to both parking garages and each of these devices included two LiDAR sensors. In addition, two thermal camera devices were installed to the new parking garage. One to monitor side entrances and one to monitor the entrance to the roof, where electric vehicle charging stations were.

Both garages had their own Power Over Ethernet (PoE) switch, that provided power and Ethernet connection to the devices. From the switches Ethernet cables were used to connect local area network (LAN) to the local server. This solution was selected due to its simplicity and because the main target was to test those LiDAR and thermal camera devices. However, in a commercial solution, some type of wireless communication method would make more sense, because wired Ethernet network might not exist in parking garages.

One of the main advantages of this new solution was that the raw data collected from the LiDAR's was processed locally. The device only needed to send information that a car has passed the sensor to a server. This reduced the communication needed between the device and the server to minimum. The design and implementation of these LiDAR devices is described in next section.

The LiDAR was also able to capture a rough profile of the passing car as it was measuring the distance from the sensor to an object multiple times per second. The Figure 16 illustrates how this profile could have looked for a typical car.

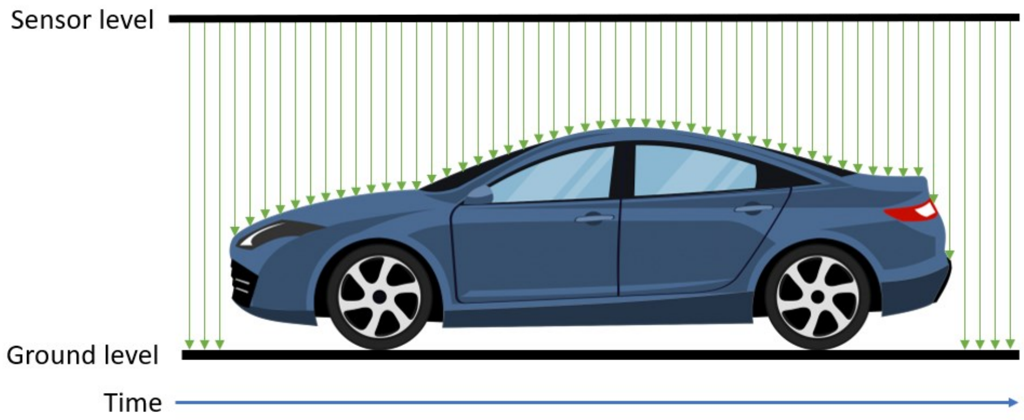

Figure 16. A profile of a car measured by LiDAR device. Green arrows indicate measured distances by the sensor as the car passes below it.

This study did not utilize the profile data, but the idea was to be able to collect these profiles for potential further usage. One could potentially differentiate different kinds of vehicles (passenger car, van etc.) if it would be relevant for other applications. During testing phase, it was noticed that the shape of a real car and the profile captured by the sensor did not match. The vehicles hood and roof gave good results, but the sensor had trouble reading correct distances from windows.

3.3 Requirements for LiDAR and Thermal Camera Devices

This subsection briefly describes the requirements that the sensor device needed to meet. The more detailed implementation of the device is described in next section.

The device requirements were following:

- The device needs to be operative outside throughout the year (IP54).
- The device should work independently even when a connection to the server is broken.
- It should be able to process locally the collected data from the connected sensors.
- It should include Power over Ethernet interface to provide the data communication capability and to provide needed power.
- Normal TCP/IP communication stack should be used so that the device could be connected to existing LAN network.
- The device should blend in with the environment.

During testing phase, it was noticed that once the device is installed to a location, it would be hard to reprogram it. For updates, manual calibrations of the sensors and to test server functionality implementing a basic Over the Air (OTA) programming capability was necessary. The OTA functionality allowed reprogramming of the sensor device over the Ethernet connection and eliminated the need to tear down installation of a sensor devices to update them.

4 Results and Analysis

This section describes the layout of the LiDAR device, its functionality and physical implementation. It also shows how these devices were installed in the live testing environment and the results of the initial tests.

4.1 New Intelligent LiDAR Device

To simplify component list, the same microcontroller was used for both the LiDAR devices and for the thermal camera devices. Also, to get first prototype quickly, a requirement was to use a processor that is easy to program and has existing examples. To simplify the designing process and to minimize potential failures, it was decided to use microcontroller development boards. These boards already contain all necessary parts for running applications and have easy access to General Purpose Input/Output (GPIO) pins. For example, development boards usually contain power circuitry, microprocessor, Random Access Memory (RAM), Flash memory for the application and crystal oscillator for the processor clock. Also, development boards usually contain connectors and pin headers for connecting external parts and USB connector to program and debug the microcontroller.

One major candidate was Arduino family of development boards. They are widely used, easy to work with and have a lot of support available but were not powerful and compact enough for this application. It was decided to use ESP32 microcontroller family, the specific module and the development board are confidential information and remain undisclosed by the company the study was performed to. ESP32 microcontrollers have 32-bit dual-core microprocessors that can operate at 240 MHz and have 520 KiB of static random-access memory (SRAM) [19]. The used development board has 4 MiB program memory, built in Ethernet MAC interface and can be powered with PoE. Also, these microcontrollers already have built-in support for I^2C, SPI and UART protocols that the sensors required. The development board can power external devices with either 12V or 5V with 1 amperage of current. ESP32 supports Arduino Integrated Development Environment (IDE), which allowed quick first tests for the ESP32 microcontroller and the sensors. However, it lacks a lot of common features that many different IDEs provide, and the needed code was written using Visual Studio Code and Platform IO extension.

With the dual-core processor, one core was programmed to read attached sensors and one core handled communication and data analysis. Communication protocols usually have wide tolerances for response timings and doing data analysis in parallel does not interfere it, but sensors must be read constantly to not lose any measurements. Also, in case of communication problems, for example not receiving acknowledgement bits, it was important to not just wait and stop all other tasks the processing had and possibly miss car profile data or more importantly fail to detect passing cars.

The development board had connector for Ethernet cable and GPIO headers to solder wires to, but to ease assembly and possible maintenance tasks and to make the system more modular, using connectors for the sensors was necessary. This required manufacturing a breakout board that would contain a matching header to the development board GPIO and all necessary sockets for the LiDAR and thermal camera sensors.

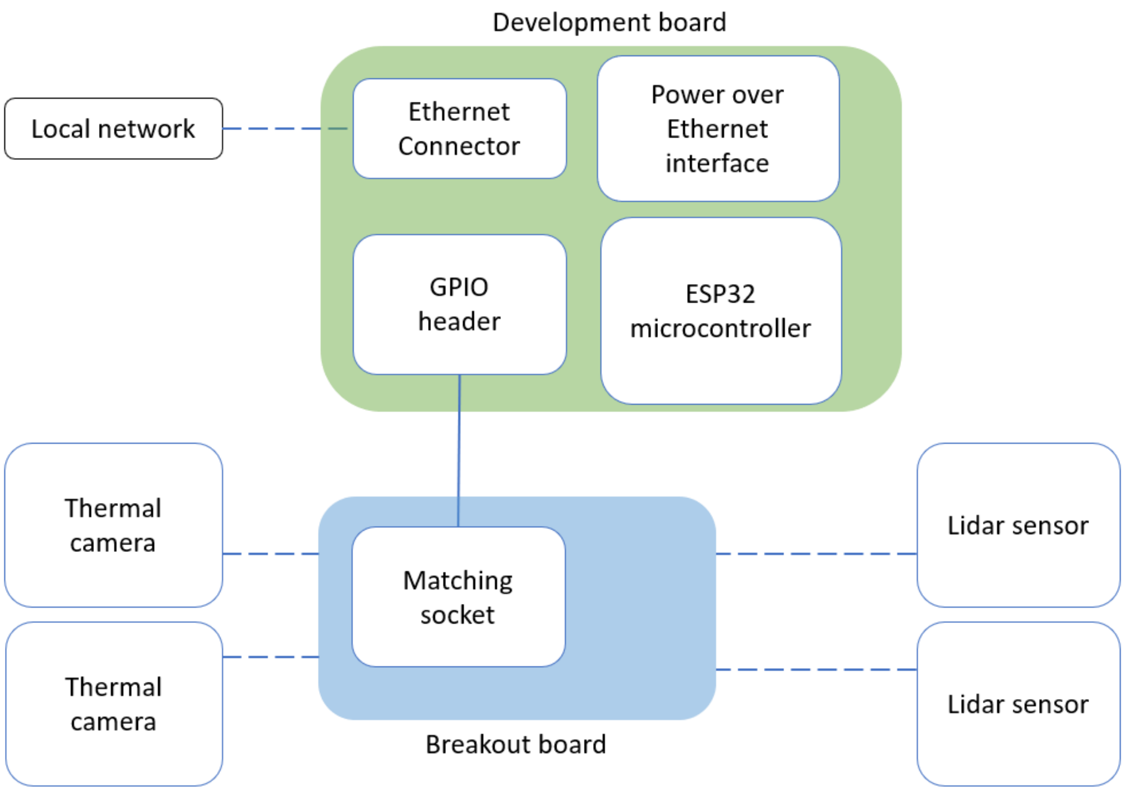

Figure 17. A layout of the new finished deign.

On Figure 17 the green rectangle represents the development board and the blue one represents the breakout board. Dashed lines mean possible connections with connectors and solid line between boards indicate a solid connection. More specific electronic schematics will remain unpublished. All Printed Circuit Board (PCB) designs were made using Electronic Design Automation (EDA) software tool called KiCAD. This is a free open-source cross-platform tool that contains electronic schematic editor, PCB layout editor and supports Gerber file generation and visualization. [20]. Gerber files contain each PCB layer information in accurate vector format and are used widely for PCB manufacturing. To minimize different PCB designs, the breakout board was designed to contain all necessary connectors for both thermal camera and LiDAR sensors. This allowed the electronics to be identical for both devices, excluding the sensors, and it only needed a different three dimensional (3D) printed enclosure. Some additional components, such as pull-up resistors for communication lines and decoupling capacitors for each connected device, were inserted to the breakout board to finish the design. The development board had a standard RJ-45 connector for LAN network was PoE capable and the breakout board had two 4-pin connectors for the LiDAR sensors and two 8-pin connectors to allow 2 thermal cameras.

None of the sensor devices were originally planned to have two thermal cameras and the design choice was more for the future and for next versions of the device. That turned out to be a good decision, because during live testing it was decided to include another thermal camera sensor to one of the thermal camera devices. The breakout board had smaller dimensions than the development board and it was connected using male pin headers. This approach only added some height to the electronics, but because the sensors were so big it did not add any height to the enclosure. All designs of the breakout boards used 2 copper layers as they are affordable to produce.

The first prototypes of the breakout boards were manufactured in a local workshop for startups and students by using milling machine. The results were adequate for testing the devices inside office space but were not good enough for outside usage. These prototype boards had bare copper top and bottom layers and had no protection to air or moisture. This could lead to broken traces and signal loss to the sensors when the copper would start to oxidize, or water could cause electrical short that could damage the sensors and the microcontroller. The next iteration of the breakout board was made by a Finnish PCB manufacturer that provided much better finished PCBs. These new boards had solder mask to protect the copper from moisture, silkscreen to indicate component placement, and had higher overall quality.

Locally manufactured PCBs have much steeper cost compared to Asian PCB manufacturers and when they are ordered in low volumes can significantly increase a device cost, Local manufacturers usually have faster turnaround time and allows for faster development of product which can offset the cost. When the sensor devices design is finished, mass production utilizing foreign manufacturers should be considered, especially when an assembly services are required, to get price per circuit board down.

When the breakout board design was finished and its dimensions were known, 3D models for both thermal camera and LiDAR sensor devices were made. These models included both boards connected with male pin headers, the corresponding sensors, and an enclosure. When the enclosure models were finished, they were made using 3D printer. Both thermal camera and LiDAR sensor enclosures were similar in size and look and had only few differences. The thermal camera device had one hole for the camera and was pointing directly downwards, and the LiDAR device had two holes for two sensors in an angle. The angle is illustrated in Figure 18. The first thermal camera device did not need any angles as it had high enough field of view (FOV) to see the whole entrance, but later when a sensor device required dual thermal cameras a similar angle to the LiDAR devices was used.

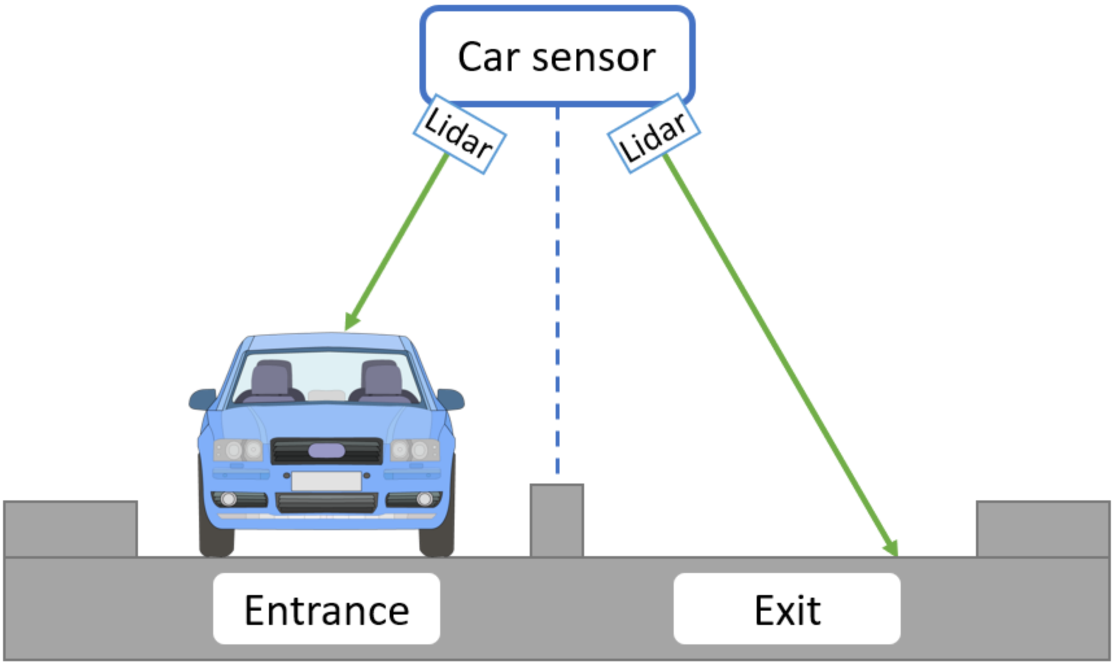

Figure 18. An illustration of the device placement and LiDAR sensors angle.

The angle at which the LiDAR sensor is pointing downwards was found to not be too critical for as long as it was pointing at approximately to the center of the lane. It mostly affected profile of the car the sensor measures. One of the major drawbacks of only measuring one point is a possibility of not detecting motorcycles or other narrow vehicles. Perhaps with a next iteration of the device some sort of a rotatable mechanical structure for the sensor could be used that allows for measuring the whole width of the road or a sensor that measures multiple points from the road such as a scanning LiDAR or flash LiDAR [21].

When the initial tests of the sensors and microcontrollers were done using Arduino IDE, software programming was moved to a more appropriate IDE: Visual Studio Code (VS Code). VS Code has auto-complete feature, syntax highlight, and it shows errors in the code. This makes developing any code much faster than the Arduino alternative. It also features debugging mode that made solving problems with the code more manageable. When switching the program environment, it was decided to use Platform IO extension to the VS Code and C as the programming language, as opposed to C++ that was in use with the Arduino IDE. Platform IO is a cross-platform, cross-architecture, and multiple framework tool for developing embedded systems [22]. The extension handles ESP32 specific libraries and building code for the microprocessor. With these tools it is possible to upgrade the processor to something different with minimal code changes.

The software that was written to the controller was similar to both the LiDAR and the thermal camera versions of the devices. Both devices used one processing core for reading sensors and one core to analyze generated data and to communicate with the connected server. In order to calibrate the sensors, each sensor device creates a local webserver and -page that is only accessible within the same LAN network. These webpages contained changeable threshold values for when to detect a passing car and current readings of the sensors for comparison. The device also contained the number of cars it had detected and various debugging information. The LiDAR device webpage also showed the latest profile of a detected vehicle. The thermal version of the device showed a real time image feed of the sensor and when a car was passing by it showed where it detected the vehicle. The webpages that were created did not utilize any passwords and were vulnerable to malicious activity, but as the sensor devices were connected to a separate private LAN network inside a private parking lot, it was not a major concern during this development phase. Security is in general a big concern in IoT applications and as the project develops, passwords and other security features became necessity.

To protect the circuitry from moisture and dust, the sensor devices were installed inside the car parks and to the ceiling. Care was taken in designing the enclosures to not allow water to seep through any holes and the Ethernet cable was inserted with a waterproof pass through. The finished design is shown on Figure 19.

Figure 19. A picture of the new car sensor devices.

The topmost device without the “ears” was an older case design and later the ears were added for mounting purposes. The initial testing showed promising results of watertightness when the devices were mounted to their place. A gray color was chosen to be discrete when next to a concrete ceiling.

4.2 Testing Sensor Devices

The company this study was conducted for did not own the parking lots where the sensors were going to be installed so the necessary hardware for the sensors were set up by the lot owners. No construction work was necessary for the buildings, so they did the installation of PoE network switches and Ethernet cables from the switches to the sensor devices and to building where the local server was located. The mounting of the sensors was done by us.

At first only the parking garage, that had the preliminary camera solution, had new LiDAR sensor devices installed, because it located closer to the office where the designing was done. As the parking garage did not yet had any mounting holes for the devices zip ties were used in installation. At this point, the old system and the new could be compared and the results showed similar results with the old camera system and the new LiDAR devices. It was quickly noticed that the sensor software was not developed enough and had memory leaks that caused periodic shutdowns, which were unnoticed during the development in the office. A memory leak is when a system is allocating memory to a variable or a function but is not releasing it after it is no longer needed. This is extremely critical in embedded applications where memory is a limited resource. If the memory is fully allocated new function calls or variable changes can overwrite critical parts in the memory and cause unexpected behavior or system reboots. In contrast, a modern non-embedded operating system can detect a memory leak and halt execution of a program. The memory leaks were so small during testing in office that they were only discovered when the sensor devices rebooted after they were left on for more than a day in the parking lot. In order to notice similar problems sooner a more detailed debug information was programmed to the software that also reported information about memory usage. A need for remote programming of the devices became quite prevalent after a few trips to disassemble sensor installations, going back to office to reprogram the devices and then reassemble the installations. As the ESP32 microcontroller supports over the air programming, it was developed to the sensor software. There were no sensor devices installed on the second parking garage as it was located a bit further away, but with the new OTA capable software new LiDAR devices were installed to the second garage's main entrances.

As the development for the sensor continued and the project grew, it was decided that a dual thermal camera sensor was needed for the second building. This dual thermal camera sensor would monitor the roof access where electric vehicle charging stations were located. The roof access had two lanes: entrance and exit but had no dividers between the lanes. It was previously noticed that those lanes were mostly ignored, and vehicles would typically go near the center of the two and this would make LiDAR sensors unable to determine the direction of the vehicle. This required that the used sensor should be capable of measuring the direction where vehicle is going, in order to determine the roof occupancy, and a single thermal camera did not have wide enough field of view. It was a concern that the microcontroller would not be powerful enough to analyze both thermal camera images at the same time, but it turned out that it was capable and did not require any upgrading.

Figure 20. A close-up shot of the thermal camera device installed at the ceiling of the parking garage using zip ties.

Before mounting holes were drilled to the concrete of the parking garages, cable ties were used to mount the sensor devices to metal bars that held lightning, as illustrated in Figure 20. Later the sensors were moved to the ceilings, zip ties were replaced with proper mounting screws, and the casings were replaced with one that had mounting holes. Even with improper mounting solution, the device was secure and was able to detect vehicles. The gray color for the enclosure was chosen for the devices to blend well with the surrounding concrete as dictated by one of the goals of the project. A wider picture of the thermal camera device installed to the roof access and its surroundings is shown in Figure 21. With the two thermal cameras, the sensor can detect vehicles in whole width of the road.

Figure 21. A Dual Thermal Device installed at the ceiling of the garage (see red circle).

During live testing phase of the sensor devices, it was noticed that several drivers used exit and entrance lanes wrongly causing the single point measuring LiDAR devices unreliable. For next iteration of the devices a different sensor, multiple same LiDAR sensors per lane or calculation of car movement direction based on car profile could be a better alternative. Because of the thermal camera sensors cost, they could not be used for every entrance. It was noticed that the generated heat signature of vehicles can drastically change depending on multiple of different factors. For example, an electric car in heavy rain could have almost the same measured temperature than the surroundings and be almost invisible to the thermal camera sensor.

Most of the developing time for the sensor devices were used in programming the software, fixing bugs in the code, and adding new features to the devices as the project grew. For example, the algorithms for detecting vehicles from sensor data were all developed in house. The lack of good live testing solution made calibration and developing the algorithms difficult, as driving around the sensors all day was not an option and relying on the parking garages random usage was inefficient. One possible way to test the LiDAR devices algorithm, that was not implemented, is to record several samples of vehicle profiles and then instead of using a sensor as an input another microcontroller would be used. The other microcontroller would act like a sensor module and feeds those recorded samples to the original sensor device. This way the detection algorithm could be efficiently developed, but it has possibility to cause the algorithm to be too specific to only detect cases that are similar to the recorded samples. Similar approach could be used for the thermal camera devices.

As there were more LiDAR sensors detecting vehicles, fine tuning them was faster than for the thermal camera alternatives. LiDAR sensors detection algorithm was based on the fact that the distance to ground was known and if the measured distance would become too small, an object would be underneath the sensor, the sensor device would start recording distances. Once the measured distance would go back to same level as ground the recording would stop and the device would analyze whether it was a vehicle or a person, mostly by measuring how long it took for the object to pass the sensor. Only vehicles were counted, and this information was sent to the server.

LiDAR sensors had trouble with measuring distances to vehicle windows and would sometimes have similar readings than to ground and would count the same car twice when the window passes. If these small gaps of readings would be ignored then when multiple vehicles entered the garage, with only small distances in-between them, the sensor would only count them as one. Problems with the used thermal camera sensors were that it had some noise in the thermal picture, which was causing false positive detections for objects. These were mostly blurred out by averaging multiple images together and when there was a significant difference to the blurred image a detection algorithm would start. Also, as people would walk near the thermal camera, it was necessary to ignore heat signatures that had small area, but vehicles that were just started produced similar small heat areas, especially during rainy weather, and caused the sensor to not detect them.

When developing the software for the sensors there were some minor problems that are still a bit mystery. Sometimes a LiDAR sensor would just report same readings even when a hand was placed in front of it and required power to be cycled to work properly. This was hard to notice when the sensor device was installed to location as there were no line of sight with the parking garages from the office. These scenarios were rare and mostly happened in the office and not when the device was installed to the parking garage. Also, there were problems with programming the microcontrollers. Sometimes new program code would not install, even with multiple tries, without any warnings or errors. This could happen because of a bad connection with the device and programmer, other user error, or it could be a problem with the programming software.

5 Conclusions

The purpose of this study was to make two non-intelligent multistory parking lots into such that can report their own usage level (i.e., how many cars are parked inside the lot) to a local server. The solution should be affordable and require no major modifications to the existing structure. There was a preliminary solution for one of the parking lots, but that solution contained a lot of problems mentioned on the introduction. With new sensor devices counting vehicles an application could be made, that compares historical and current usage levels, to make predictions whether it would have room for more vehicles even before users would leave their current locations. This application was not part of the study but was one end goals for the solution and was kept in mind during whole development. This study was conducted using LiDAR sensor modules and thermal cameras to detect vehicles entering and exiting the two parking lots. ESP32 based development board was used for time saving purposes.

The developed sensor devices had promising results at detecting the vehicles. Some drawbacks were noticed during development, but due to time constraints it was not possible to solve all of them. One noticed problem was that drivers would use exits and entrance lanes wrongly, making the sensor device count parking lot capacity wrong. Also, the thermal camera versions of the devices would require more testing than there was time for. Despite the problems there were a lot of benefits as well. For example, the new devices were modular and could be installed to many kinds of parking lots, they did not need constant connection to the local server to function, they blended well with the environment and the server load was significantly reduced compared to the old system. These devices are only needed for each entrance and exits, and so only a few sensors are needed to cover whole parking lot, possibly making the system cost more affordable compared to other methods of smart parking. Also, this setup does not need any major construction work and each sensor can be easily installed using only a few screws to concrete, making this setup a desirable solution to improve existing old parking lots.

As the devices were the first prototypes, they were a success. Improvements for next iterations of the design would be using batteries as the power source and low power Wi-Fi as the communication route and transform the device into a proper IoT device. IoT has its own obstacles to overcome, mainly with security and battery management, and were not implemented to the devices yet. Also, the used LiDAR sensors were not able to detect vehicle’s moving direction and needed a new solution to solve a problem with drivers using wrong lanes.

References

1 Dataflair team 2018. IoT Applications | Top 10 Uses of Internet of Things. https://data-flair.training/blogs/iot-applications/ (24.10.2020)

2 Rahmani, Amir M.; Gia, Tuan Nguyen; Negash, Behailu; Anzanpour, Arman; Azimi, Iman; Jiang, Mingzhe and Liljeberg, Pasi 2017. Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. Elsevier.

3 Arpan Pal and Balamuralidhar Perushothaman 2016. IoT Technical Challenges and Solutions. Artech House.

4 A. Gopil and M. Kameswara Rao 2018. An Enhanced Study on Internet of Things. International Journal of Pure and Applied Mathematics. https://acad-publ.eu/hub/2018-120-6/1/15.pdf (25.03.2021)

5 Secure, sustainable smart cities and the IoT. Thales. https://www.thales-group.com/en/markets/digital-identity-and-security/iot/inspired/smart-cities (02.03.2021)

6 Daniels, Guy 2014. Deutsche Telekom uses big data to make parking easier in Pisa. Telecom TV. https://www.telecomtv.com/content/automotive/deutsche-telekom-uses-big-data-to-make-parking-easier-in-pisa-11475/ (01.04.2021)

7 Birchenko, Yury. Smart Parking Solutions - IoT sensors space race. Farnell. https://fi.farnell.com/smart-parking-solutions-the-iot-sensors-space-race (13.03.2021)

8 Lamas, Jose; Castro-Castro, Paula-Maria; Dapena, Adriana and Vazquez-Araujo, Francisco 2016. SiDIVS: Simple Detection of Inductive Vehicle Signatures with a Multiplex Resonant Sensor. MDPI.

9 Intel IoT 2015. Intelligent Parking Management Paves the Way to Better City Living. Intel. https://www.intel.com/content/dam/www/public/us/en/documents/solution-briefs/iot-siemens-smart-parking-brief.pdf (27.04.2021)

10 Romero, Leonardo and Concha, Antonio 2014. Building a Ring of Ultrasonic and Infrared Sensors For a Mobile Robot. Universidad Michoacana de San Nicolas de Hidalgo. http://dep.fie.umich.mx/~lromero/my_papers/sensors_lars2007.pdf (26.04.2021)

11 Baroffio, Luca; Bondi, Luca; Cesana, Matteo; Redondi, Alessandro and Tagliasacchi, Marco 2015. A visual sensor network for parking lot occupancy detection in Smart Cities. IEEE 2nd World Forum on Internet of Things, WF-IoT 2015, Milan, Italy

12 Vezelis, Paulis 2018. PIXEVIA: Artificial Intelligence-based Smart Parking Needs Only a Security Camera. Parking Network. https://www.parking-net.com/parking-news/pixevia/artificial-intelligence-smart-parking (27.04.2021)

13 KYOSIS, 2018. Smart Parking and Practicing Car Theft Preventation this July. https://www.kyosis.com/2018/07/05/smart-parking-and-practicing-car-theft-prevention-this-july/?v=f0aa03aaca95 (26.04.2021)

14 Benewake Co. Ltd. Product Manual of TFmini Plus. https://www.manualslib.com/manual/1544417/Benewake-Tfmini-Plus.html (01.06.2019)

15 Benewake Co. Ltd. TFmini Plus 12m IP65 distance sensor. http://en.benewake.com/product/detail/5c345cd0e5b3a844c472329b.html (10.06.2019)

16 Flir 2018. FLIR LEPTON® Engineering Datasheet. https://www.flir.com/globalassets/imported-assets/document/flir-lepton-engineering-datasheet.pdf (24.06.2019)

17 Sparkfun, FLIR Radiometric Lepton Dev Kit product page. https://www.sparkfun.com/products/retired/14654 (03.04.2021)

18 FLIR. As far as 30 fps vs. 9 fps video rates are concerned, why use one over the other? https://www.flir.com/support-center/oem/as-far-as-30-fps-vs.-9-fps-video-rates-are-concerned-why-use-one-over-the-other/ (03.04.2021)

19 Espressif. ESP32 Series Datasheet. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (10.06.2019)

20 KiCAD. About KiCad. https://kicad.org/about/kicad/ (01.0.6.2021)

21 Khader, Motaz and Cherian, Samir 2020. An Introduction to Automotive LIDAR. Texas Instruments. https://www.ti.com/lit/wp/slyy150a/slyy150a.pdf (01.04.2021)

22 PlatformIO. What is PlatformIO? https://docs.platformio.org/en/latest/what-is-platformio.html (10.06.2019)