TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Haigo Hein 121901IATM

NB-IoT prototype system for smart waste management

Master's thesis

Supervisor: Muhammad Mahtab Alam
Professor

Sven Pärand
PhD

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Haigo Hein 121901IATM

Kitsaribalist asjade internetti kasutava nutika jäätmekäitlussensori prototüüp

Magistritöö

Juhendaja: Muhammad Mahtab Alam
Professor

Sven Pärand
PhD

Tallinn 2020

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: Haigo Hein

02.01.2021

Abstract

The amount of waste generated in the world is only growing along with the population growth among other factors. Various methods to mitigate the waste generation, management and reuse problems are needed to tackle the problem worldwide. Part of the solution is smart waste management, which would make the process of waste collection more effective.

Waste container fill level estimation prototype system is proposed in this thesis. The prototype includes a camera, an ultrasonic distance sensor and NB-IoT and Cat-M1 capable modem. Proposed system would detect container's fill level by estimating the magnitude of image difference of empty and filled status. Prototype's advantages and disadvantages were discussed and analysed: in certain situations the input from camera is rendered futile due to using low complex image difference comparison algorithm and system would rely solely on ultrasonic sensor. Architectural decision of whether to process images on the edge or upload to application server was observed from energy consumption perspective, which is vital in context of low power IoT devices. As a result, for proposed prototype model local processing was preferred over uploading images over NB-IoT. Local processing scenario used about 20% less energy compared to cloud processing where raw data was transported over NB-IoT. In case cloud computing is a requirement, Cat-M1 is suggested for that purpose.

This thesis is written in English and is 43 pages long, including 5 chapters, 33 figures and 8 tables.

Annotatsioon

Kitsaribalist asjade internetti kasutava nutika jäätmekäitlussensori prototüüp

Maailmas tekkiva prügi hulk on aasta-aastalt ainult kasvav. Põhjused selleks on näiteks kasvav rahvaarv, jätkuv linnastumine ja üleüldine tarbimise suurenemine. Selleks, et jäätmete teket vähendada ning haldust ja taaskasutust edendada, on vaja nutikaid viise, mis muudaksid kogu protsessi efektiivsemaks. Üks osa lahendusest on nutikad jäätmekäitlussüsteemid, mis optimeerivad jäätmete kogumisele kuluvaid ressursse.

Selles magistritöös pakutakse välja prügikonteineri täitumise monitooringu IoT seadme prototüüp, mis koosneks OpenMV kaameramoodulist, ultrahelikaugusmõõturist ning NB-IoT ja Cat-M1 tehnoloogiaid toetavast modemist. Loodud süsteem hindaks konteineri täituvust hinnates erinevuse suurust tühjast konteinerist tehtud pildi ja mõõdetava olukorra pildi vahel. Prototüübi potentsiaalsed puudused leitakse ja tuuakse välja, et tingitud väljapakutud kaamera asukohast konteineri siseseinal ja suunast vastasseina poole esineb juhtumeid, kus kaamerast saadav sisend annab vale tulemuse ning süsteem toetub ainult ultraheliandurile. Üritati leida vastust arhitektuursele küsimusele, kas pilte tuleks töödelda IoT seadmes või tuleks need üle raadiovõrgu transportida rakendusserverisse, kus ressursid pole piiratud. Teostatud energiakulu testide tulemusena järeldus, et pakutud prototüübi tööpõhimõttena on soodsam analüüsida pilte IoT seadmes ning saata ainult tulemus üle NB-IoT raadioliidese rakendusserverisse. Lokaalse töötluse ja tulemuse teenusserverile saatmise versioon oli mõõdetud tulemuste järgi umbes 20% energiasäästlikum kui algandmete ehk piltide täielik ülekanne üle NB-IoT. Kui mingil juhul peaks olema vajalik perioodiliselt andmeid suurusjärgus mitu kilobaiti ja rohkem laadida teenusserverisse üle LPWA raadioliidese, siis oleks Cat-M1 eelistatud NB-IoT ees.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 43 leheküljel, 5 peatükki, 33 joonist, 8 tabelit.

List of abbreviations and terms

IoT	Internet of Things
NB-IoT	Narrowband Internet of Things
IR	Infrared
ToF	Time of Flight
LIDAR	Light Detection and Ranging
FOV	Field of View
LPWAN	Low-Power Wide-Area Network
LPWA	Low-Power Wide-Area
3GPP	3rd Generation Partnership Project
ISM	Industrial, Scientific and Medical
LTE	Long Term Evolution
UE	User Equipment
GSM	Global System for Mobile Communications
UMTS	Universal Mobile Telecommunications System
SIM	Subscriber Identification Module
MNO	Mobile Network Operator
VPN	Virtual Private Network
GPRS	General Packet Radio Service
USB	Universal Serial Bus
UART	Universal Asynchronous Receiver-Transmitter
MB	Megabyte
RAM	Random Access Memory
VGA	Video Graphics Array
LED	Light-Emitting Diode
I2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
CAN	Controller Area Network
VCP	Virtual COM Port

RPC	Remote Python/Procedure Call
PC	Personal Computer
IDE	Integrated Development Environment
AT	Hayes command set
PMOD	Peripheral Module
ETSI	European Telecommunications Standards Institute
RDP	Remote Desktop Protocol
MTU	Maximum Transmission Unit
QVGA	Quarter VGA
eMTC	Enhanced Machine Type Communication

Table of contents

List of figures

List of tables

1 Introduction

The world generates billions of tonnes of municipal solid waste annually and the amount is only expected to increase [1]. Major trends which contribute to that are increasing population [2], economic growth in developing countries and urbanization [3]. High income countries generate more than third of the world's waste while having only 16% of the population. It is estimated that one third of that waste generated is not handled in sustainable manner. Sub-Saharan Africa, South Asia, and the Middle East and North Africa, where more than half of waste is openly dumped, are expected to multiply their waste generation by 2050 [1]. It is known, that developing regions collect less waste than developed areas illustrated on Figure 1 and collection rate improves when incomes rise. Overall, global waste is expected to grow more than double of the population growth over the same period by 2050 because of trends mentioned before. Waste collection is still problematic in many developing countries. A good waste management system has positive impact on public health and environmental pollution by eliminating burning and unmanaged dumpsites and additionally enables reuse of materials, often called "circular economy". Recycling helps to conserve scarce resources and reduce the share of waste being dumped, but it heavily depends on keeping the materials separate before collection [3].

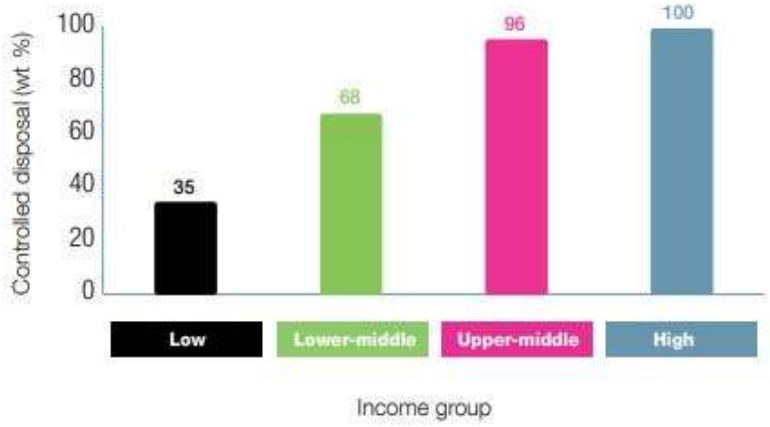

Figure 1. Controlled disposal for selected cities by income level [3]

Waste management is labour intensive and only transportation costs can be in the range of (20-50) $ per tonne, in more developed countries the whole waste management chain has operating costs over 100 $ per tonne generally [1]. Knowing and predicting correctly when trash containers need to be emptied allows to optimise solid waste collection routes which in turn saves time and fuel — kind of systems which base their decisions on data from subjects are called smart systems.

Smart waste management consists of two steps [4]:

1. Reducing the amount of waste
2. Optimizing resource usage for waste management itself

The first step includes smart ways to minimize waste generation overall. It can be more efficient packaging of products, smart management of food and other perishable goods and also previously mentioned circular economy model — meaning that large share of waste materials would be reused for producing new products. The second step implies making waste disposal, collection and sorting optimized to use minimal amount of resources — time, fuel, electricity and any other resources used in process. Implementing IoT (Internet of Things) sensor solution to monitor the fill level of waste containers would lead towards efficiency [5]:

- The IoT solution would provide data to information system guiding waste collection trucks, which would be able to select optimal routes for pickup
- From client perspective — cases of overfilled waste containers would decrease
- Already gathered waste monitoring data would be used to predict the future waste generation
- As a result of this process CO_2 emissions from waste management function would decrease

This thesis proposes a prototype system to support the second cause – waste container fill level monitoring. Three different methods of information gathering are shown on Figure 2. First method is using passive sensors and nearby transfer nodes which carry out the fill level reading and data transfer tasks. The second method implements a waste truck as a transport node where active sensor is using close-range communication technology. The third way, targeted in this work, has active IoT module, which measures and transfers

itself the data to necessary data collection node. Aggregating the data will be useful also for other stakeholders besides waste management company shown as a step 4 on Figure 2, who could use this info for benefit of a society.

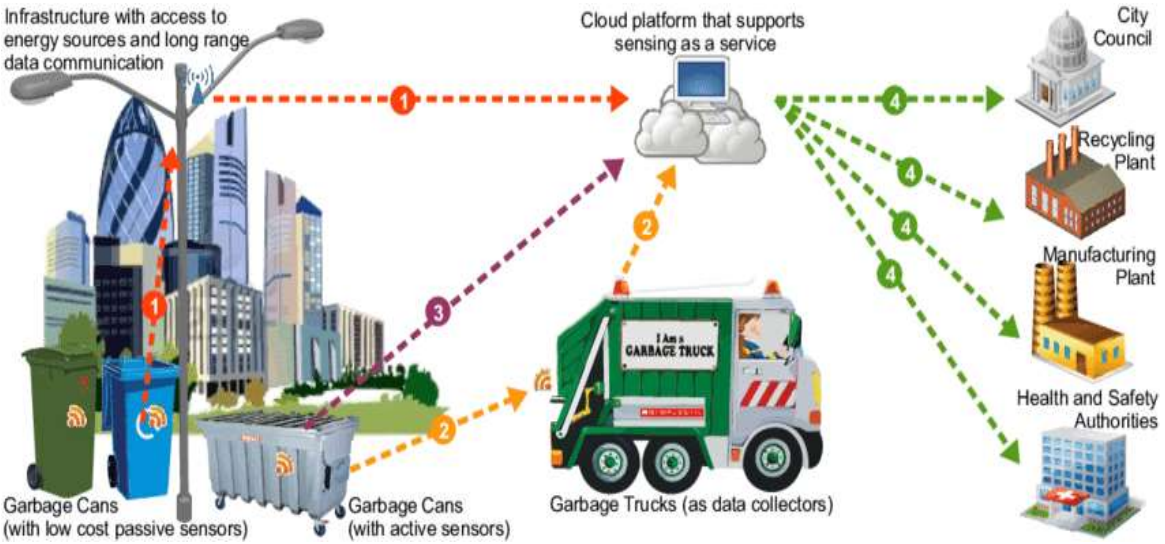

Figure 2. Waste management in smart cities [6]

This thesis consists of five main chapters. Chapter one is an introduction to the smart waste management topic and describes the objectives set for this work. The second chapter gives an overview of technical aspects of previously proposed and commercially marketed devices. The third chapter introduces the hardware used in this thesis and describes the initial image upload experiment. The fourth chapter presents the proposed prototype system and attempts to analyse its feasibility by simulating tests and theoretical analysis. Finally, in fifth chapter, the summary of the thesis highlighting results and conclusions is given.

1.1 Background

This thesis is a continuation of a previous work [7] done on similar subject in Taltech and takes all findings and considerations from there as a starting point. Different low-power communication technologies are discussed and compared and cellular IoT is assessed to be most suitable for this type of solution. Author proposes a fill level estimation prototype system for trash bins used in houses, offices and schools which includes hardware overview and also an image processing algorithm. The solution would be based on optical sensor and consist of camera module and communication module. Several approaches of image transfer from camera to cloud server were considered and problems regarding data

transfer with specific setup detected. The camera module was used to create a set of pictures which were used to simulate the proposed algorithm in Matlab. Simulated algorithm utilized grayscale images taken of a bin, which were later analysed by estimating the non-black area of the image. As prerequisite, black trash bags were used inside bins, so objects placed inside would stand out on a image. As a result, accuracy of proposed method of identifying free space in trash bin was over 90%.

1.2 Objectives

Although better technological solutions cannot solve the entire waste management problem, they might help to make existing procedures, methods and systems work more efficiently which would be another step towards more sustainable society. Need for minimising wasteful resource usage in waste management itself is also present.

Therefore, firstly this thesis aims to further investigate technical problems related to uploading the images to potential cloud-based system discovered in previous work [7]. Secondly, the goal is to evaluate different design options for waste container monitoring in the context of optical sensor and cellular IoT. Thirdly, selected method should be implemented in a test environment where prototype could be evaluated which will provide information about possible solution’s practicality and adequacy.

2 State of art

This section discusses different proposed or implemented technical solutions for IoT waste management which are able to determine container's fill level and transfer that information to its intended destination, which does not necessarily have to be a cloud server. Therefore, sensor options and wireless data transfer technologies are explored, though this thesis mainly concentrates on NB-IoT (Narrowband Internet of Things) technology. Commonly smart waste management systems consist of sensors, transmission method to data acquisition (and possibly processing) component and connectivity to aggregating information system which could be a cloud service.

2.1 Fill level measurement

As also mentioned in previous work [7], most popular sensors appear to be ultrasonic, IR (Infrared) and weight sensors, each having advantages and deficiencies. Some solutions employ also additional sensors, for example temperature and/or humidity sensors to monitor the environment inside the bin.

Ultrasonic sensor is most common sensor to use for waste container level monitoring shown on Figure 3. It is usually placed under the bin lid where it measures the distance to the bottom of the bin. Ultrasonic wave is an acoustic wave which is above frequencies heard by human ear. The sensor emits an ultrasonic wave and receives an echo back from the target – time till receiving the reflection is proportional to covered distance. When a container is filling, the distance between the sensor and target is nearing zero. Shortcoming of that sensor is, that it has a narrow measuring angle – less than 15° for HC-SR04 [8]. That results in blind spots and is the reason for false results estimated by a system using that sensor.

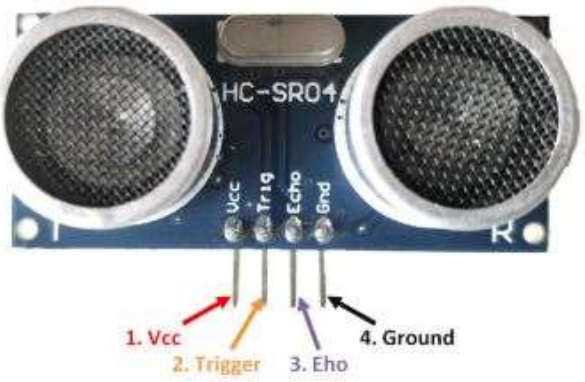

Figure 3. HC-SR04 Ultrasonic Sensor [8]

IR sensors work in principle similarly to ultrasound sensors: they emit infrared light and register the reflection coming back to the sensor show on Figure 4. That way the sensor detects the change in the environment: intensity of the incoming light is compared to preset threshold. IR sensors cost less that ultrasonic, have faster response time and can detect objects better in close range within 0.5m [9].

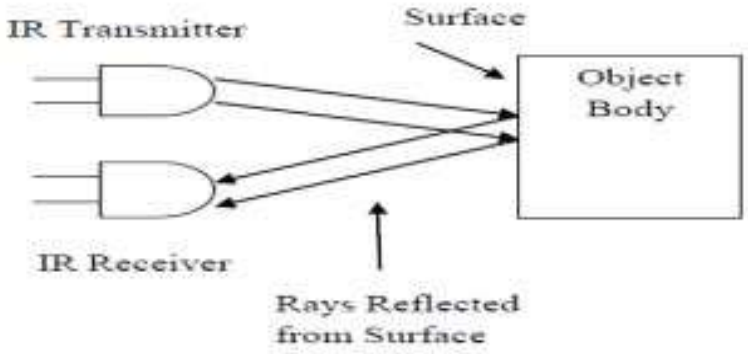

Figure 4. Working principle of IR sensor [10]

Load sensors are used to evaluate the weight of the waste container [11]. For obvious reasons using only this type of sensor cannot provide accurate results, because weight does not give direct insight about the volume of the contents in the container [12].

Cameras as sensors are not preferred in many solutions, which probably is because of the cost of camera and image processing, both from financial and system complexity standpoint. Some proposed solutions have the camera on waste truck and do the level

estimation when the truck reaches to the container [13]. This kind of method would serve a dataset for creating a model for making the prediction for certain locations and containers fill level, but it would not give a real time or near real time information and exclude wasteful drives to half-empty bins. A study [13] refers to a solution that have mounted camera on top of the waste bin, so every time bin lid is opened, the camera takes a picture.

One scientific article proposed an interesting combination of two ultrasonic sensors accompanied with parallel plate capacitor. Schematic is shown on the Figure 5, where it can be seen that one ultrasonic sensor is attached to the lid and is measuring the bin fill level. The second ultrasonic sensor mounted outside of the bin is intended for detecting approaching human and initiating bin lid opening, which was automated in this case. As ultrasonic sensor has blindspots, the capacitor plate design is expected to improve the fill level readings. The idea behind it is simple – if something is placed between the plates, then the capacitance will change. The proposed system is reported to work as initially intended [14]. The problem with that method is that it operated two copper plates within the bin, which are not there by default in currently used containers and could prove to be costly overall.

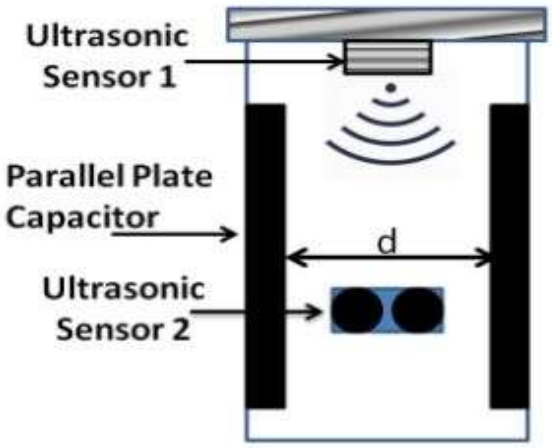

Figure 5. Smart bin with copper plates [14]

In addition to mentioned sensors, some commercially offered smart waste management products utilize ToF (Time of Flight) radar sensors, also called LIDAR (Light Detection and Ranging), some examples are introduced in section 2.3Commercially sold products. The working principle, which is shown also on Figure 6, has similar working principle to

ultrasonic and IR sensor: light is emitted and time until reflection returns to sensor is used to calculate the distance to the object, where the light reflected from. Unlike ultrasonic sensors, ToF cameras are usually outputting not only distance of a closest object in the FOV (Field of View), but they create a depth map of the surroundings which is a basic 3D representation of the objects in front of them. Granularity of the 3D image is determined by number of pixels the sensor is able to record [15].

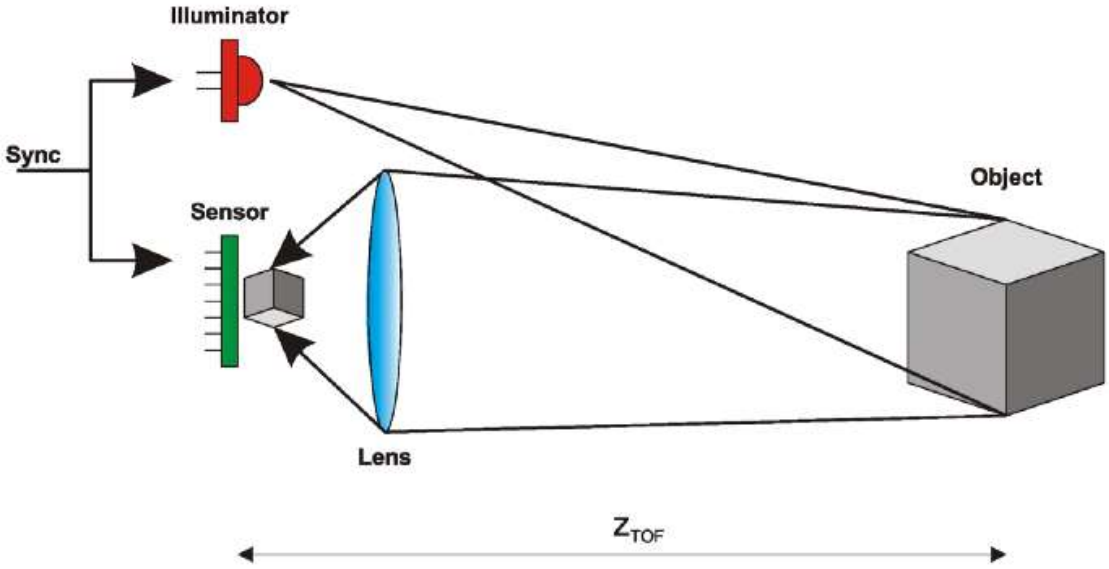

Figure 6. Working principle of ToF laser sensor [16]

2.2 LPWAN Radio technologies

Radio connections technologies intended for battery powered low power devices are named as LPWAN (Low-Power Wide-Area Network). These technologies employ narrow bandwidths and are mostly suitable only transferring small amounts of data at relatively slow speeds. LPWA networks are targeted and used for M2M (machine to machine) and IoT (Internet of Things) communications, those terms are nearly identical. As proprietary LPWA (Low-Power Wide-Area) technologies were earlier on the market than 3GPP (3rd Generation Partnership Project) LPWA technologies, they have had an advantage on market penetration so far, NB-IoT is estimated to be most used cellular or LPWA technology by 2026 [17]. Examples of proprietary LPWA are LoRaWAN and Sigfox. Later on, the cellular based 3GPP standardised technologies like NB-IoT and Cat-M1 were brought to market [18]. Coverage and abilities of different IoT radio access technologies are shown on the Figure 7. NB-IoT and Cat-M1 have a wider area of use cases compared to non-3GPP technologies.

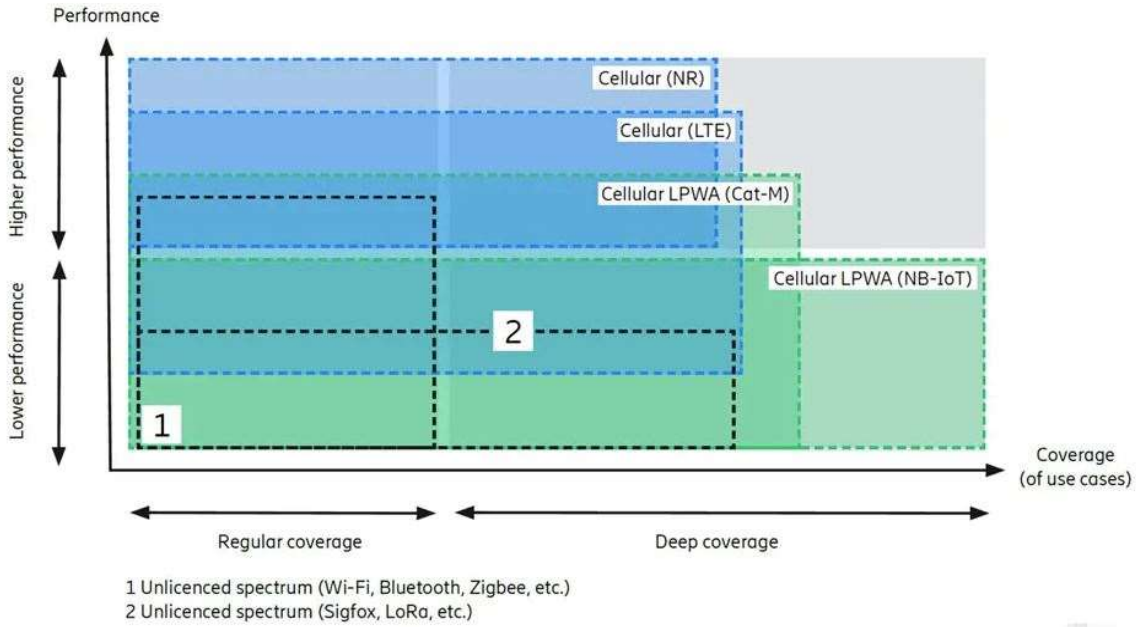

Figure 7. IoT access technologies capabilities comparison [19]

All LPWA technologies have a common purpose to offer connectivity and enhance the battery life of devices which are stationed in worse than average coverage conditions. Compared to cellular technologies (NB-IoT, Cat-M1), Sigfox and LoRa use unlicensed frequency ranges, also named as ISM (Industrial, Scientific and Medical), and that means there is no guarantee of noise levels on those bands. Although mentioned non-3GPP technologies have better coverage than NB-IoT, they have a necessity to build a dedicated radio network. Comparison of selected LPWA radio technologies parameters are displayed in Table 1.

Mutual for Cat-M1 and NB-IoT is that they are intended to be integral part of existing LTE (Long Term Evolution) network infrastructure while offering better coverage for IoT devices. Better coverage is mainly achieved by sending messages over radio interface repetitively. Better battery life is achieved by reducing the radio communication between the UE (User Equipment) and radio node – which leads to worse latency as a tradeoff. Another important part is that devices hardware is simplified compared to regular LTE devices, which is crucial to make it possible to deploy IoT devices on massive scale [20].

Table 1. Radio technologies comparison [21]

Technology/ Feature	**LTE Cat M1**	**LTE Cat NB1 (NB-IoT)**	**LoRaWAN**	**Sigfox**
Bandwidth	1.4 MHz	180 kHz	125kHz	200Hz
Spectrum	Licensed	Licensed	ISM	ISM
Standardization	3GPP Release 13	3GPP Release 13	LoRa Alliance	ETSI
Uplink (peak value)	1 Mbit/s	250 kbit/s, 20 kbit/s	50kbit/s	100bps
Downlink (peak value)	1 Mbit/s	250 kbit/s	50kbit/s	600bps
Latency	10ms–15ms	1.6s–10s	1-10s	1-30s
Duplex Mode	Full or Half Duplex	Half Duplex	Full or Half Duplex	Half Duplex
Batter Life (in years)	10	10+	10+	10+

LTE-based LPWA radio technologies have the same security features as regular LTE. Because LTE is based on standards specified by 3GPP which has learned a lot from previous generation networks like GSM (Global System for Mobile Communications) and UMTS (Universal Mobile Telecommunications System) problems, the built-in security functions of a standard provide typically more secure network compared to non-3GPP versions. Device authentication to network is based on SIM (Subscriber Identification Module) cards which reside in the communication module. A place for choice would be the connection from the MNO (Mobile Network Operator) core network to application server. Options for that visualized on Figure 8. When the application server is located in the MNO infrastructure, then the private encrypted physical connection from the core could be used. If the data needs to traverse a less secure environment like public internet, then VPN (Virtual Private Network) connections could be used. Another way to solve it would be to implement end-to-end security connection between the IoT device and the application server [22].

Figure 8. Security of IoT connection [22]

2.3 Commercially sold products

In this section overview and technological aspects of commercially offered waste management sensor modules are discussed. As it could be expected according to the relevant scientific literature, ultrasonic sensor solutions [23] - [24] are often used in commercial products, but in addition some ideas not found in literature were discovered. SmartBin for example uses besides single ultrasonic sensor additionally dual optical lasers [24]. Laser sensor is another way of measuring the distance, but they cannot be used as a sole sensing mechanism for fill level estimation as they only measure by the narrow laser beam.

Sensoneo is Slovakia-based smart waste management solutions company and its product line includes two models of bin fill level sensors. Main differences between those models are shown in Table 2 and broad difference seems to be that one sensor employs one ultrasonic beam while other just one. Connectivity wise both support major LPWA technologies with added extra GPRS (General Packet Radio Service) for four beam model. Another interesting aspect is usage of accelerometer for detecting situations where the bin lid is not fully closed, as these sensors are intended to be attached under the lid [25].

Table 2. Sensoneo smart sensor comparison [25]

Model / feature	**“Single sensor”**	**“Quatro sensor”**
Measurement	Single ultrasonic beam	Four ultrasonic beams
Measuring range (cm)	3 – 170	15 – 400
Connectivity	LoRa / Sigfox / NB-IoT	GPRS/NB-IoT/LoRa/ Sigfox
Emptying recognition	Accelerometer. Advanced tilt recognition algorithm	Accelerometer

Newer and possibly better method for fill level sensing seems to be optical sensor using ToF technology for distance measuring. One of those companies implementing it in a real product is Danish company Nordsense. Module they are offering is intended to be attached under the bin’s lid and is using 3D ToF laser, which outputs distance sensing results on 256-pixel resolution and is used to produce 3D depth map of the contents inside. Device is shown on Figure 9. Maximum measurable bin depth is reported to be 5 meters [26].

Figure 9. ToF optical sensors where (a) is Nordsense smart container module [26] and (b) is TeraRanger Evo Mini sensor

Terabee is a Swiss company offering wide range of IR ToF technology based distance detection sensors. They do not design the IoT modules but only provide the sensors themselves with outputs for USB (Universal Serial Bus), UART (Universal Asynchronous Receiver-Transmitter) connection. Their lineup also has models suitable for fill level detection for different container sizes, smaller sensors can be modularly combined. For example their TeraRanger Evo Mini is suitable for bins up to 3.3m tall and it has 27° FOV but outputs only maximally 4-pixel result. It is depicted on Figure 9. There also are models with bigger granularity - TeraRanger Evo 64px, which produces 64-pixel output, but has only 15° FOV [27].

3 Prototype system for image processing and transfer

This section will provide an overview of operated hardware and its technical specifications and features. The target was to find if and how image upload over NB-IoT could be implemented. Low complexity image capturing and uploading script is created and demonstrated, also UART communication is tested and ultrasonic sensor working principle is reviewed and prototyped with the camera module.

3.1 Hardware

3.1.1 Camera module

Camera module used in this thesis is OpenMV Cam H7. OpenMV is a project started by Kwabena Agyeman and Ibrahim Abdelkader which could be summarized by the name of the blogpost it was started with:" In Search of a Better Serial Camera Module" [28]. The project took part of Hackaday.io, which is collaborative hardware development community and where the prototype was designed [29]. The production costs to start selling the product were covered by funding from project at kickstarter.com [30].

OpenMV Cam H7 is a user-friendly machine vision camera module that is suitable for beginners because of its ease to use and available features. H7 uses STM 32-Bit Arm Cortex-M7 processor STM32H743VI [31], which is a high-performance ARM type processor intended to use in mobile embedded devices. The processor supports 1 MB (Megabyte) RAM (Random Access Memory) and 2 MB flash memory. M7 is the most powerful of the ARM Cortex-M series processors, it is suitable for a wide variety of application areas including automotive, industrial automation, medical devices, high-end audio, image and voice processing and motor control [32]. OpenMV Cam H7 is equipped with OV7725 image sensor, which is capable of 8-bit grayscale or 16-bit RGB565 VGA (Video Graphics Array) 640x480 pixels size images. Standard lens equipped with the camera has 70.8° vertical FOV and 55.6° horizontal FOV. The board also has 2 IR LEDs (Light-Emitting Diode) which coupled with IR lens should make OpenMV capable of capturing images in the dark [33].

The camera board has the following interfaces I2C (Inter-Integrated Circuit), SPI (Serial Peripheral Interface), CAN (Controller Area Network) bus; UART; USB VCP (Virtual COM Port). Built-in RPC (Remote Python/Procedure Call) library makes it easy to control the camera over USB by PC (Personal Computer) or by some other microcontroller like Arduino or single board computer like RasperryPi, which can use the library over any other interfaces besides USB [33]. On PC the board can be programmed by specific OpenMV IDE (Integrated Development Environment), which has text editor, serial terminal for debugging, frame buffer viewer and image histograms [34].

OpenMV camera can be programmed in MicroPython, which essentially is a subset of Python 3 libraries, which have been rearranged for mobile devices. Python is a popular programming language and Python version 3 is the latest major version of it, having many differences of how things are done compared to the previous major version Python 2. Python 3 was released on December 3rd, 2008 [35]. MicroPython implemented on OpenMV platform has three types of libraries. First ones are standard libraries, which are ported from normal Python and "micro-ified" to fit the circumstances while having the same function names with the added "u" in front of it, where "u" denotes "micro". Secondly, there are MicroPython-specific libraries including common hardware and network functionalities used over different platforms. Thirdly, there are OpenMV-specific libraries which enables the control of all the functionalities of the OpenMV camera board [36].

3.1.2 Modem board

Modem board used in this thesis is Avnet Silica NB-IoT Sensor Shield based on Quectel BG96 module. Modem features Cat-M1 and NB-IoT, but also GSM connectivity. The board can be controlled by AT (Hayes command set) commands over USB using serial port terminal in PC or over PMOD (Peripheral Module). The Avnet board also has a pinout compatible with Arduino ecosystem. AT commands are commands which allow control over a cellular modem and which guidelines for device manufacturers are published by ETSI (European Telecommunications Standards Institute) in technical specifications [37]. Potential use cases for the modem by the producer are listed in the Table 3.

Table 3. Avnet NB-IoT shield potential use cases [38]

Industry and agriculture	Gas detector Soil PG tester Optical sensor Machinery alarm system Irrigation controller Elevator
Consumer market	Asset tracking Electronics Person/pet tracking
Public utilities	Water/gas metering Smart parking system Fire hydrant Smoke alarm Trash can Street lighting

BG96 supports wide range of LTE bands and has a built-in support for multiple protocols to be utilized in communication between the modem and the server in the internet shown in Table 4 [38]. The board has several types of interfaces like USB, UART, I2C.

Table 4. Built-in protocols support for BG96 [38] [39]

Protocol	**Short description**
PPP	Point-to-Point is OSI layer 2 protocol used directly between two hosts.
UDP	User Datagram Protocol is stateless OSI layer 4 transport protocol.
TCP	Transmission Control Protocol is reliable OSI layer 4 transport protocol.
SSL/TLS	Secure Sockets Layer/ Transport Layer Security is OSI layer 5 protocol for secure communications.
FTP(S)	File Transfer Protocol is OSI layer 7 protocol for file transfer. FTPS is secured by TLS.
HTTP(S)	Hypertext Transfer Protocol is OSI layer 7 protocol for web browsing and services. HTTPS is secured by TLS.

PMOD interface is a type of connectors which allows the peripheral modules to connect to the host board and is defined by a company Digilent. There are six-pin and twelve-pin versions of it, meaning the first one has 4 digital I/O pins, 1 power and 1 ground pin. Twelve-pin version features exactly double of those. The peripheral module can be powered by host board through the power pins of this interface, maximum power drawn is assumed to be approximately 100 mA but is not specified by Digilent [40]. Avnet board specification states that PMOD 4 (UART) version is being used. In the newest Digilent specification, the numbering of that type appears to have been changed to type 3 [41] from type 4 [40]. Nevertheless this peculiarity, the Avnet board has PMOD 3A/4A extended UART interface with pins according to Digilent specification listed in the Table 5.

Table 5. PMOD expanded UART pinout [41]

Pin no.	**Signal**	**Direction**	**Alternate signal**	**Direction**
1	CTS	In	GPIO	In/Out
2	TXD	Out		
3	RXD	In		
4	RTS	Out	GPIO	In
5	GND			
6	VCC			
7	GPIO	In/Out	INT	In
8	GPIO	In/Out	RESET	Out
9	GPIO	In/Out		
10	GPIO	In/Out		
11	GND			
12	VCC			

The UART connection signals on PMOD connector are briefly described in the following list [41]:

- CTS - Device will only transmit when this signal is asserted
- RTS - Device is ready to receive data

- RXD - Data from peripheral to host
- TXD - Data from host to peripheral
- INT - Interrupt signal from peripheral to host
- RESET - Reset signal for host to reset peripheral

In this thesis only TXD and RXD pins are actively used for UART connections and CTS and RTS are briefly experimented with.

3.2 Ultrasonic sensor

OpenMV board was selected as the central node for to-be proposed solution and as most of the existing solutions employ ultrasonic sensor, therefore interoperability between the sensor and the camera board was studied and experimented. Because OpenMV board supports only 3.3V output signals on its general I/O (input/output) pins, the special version of ultrasonic sensor HC-SR04 was acquired. HC-SR04P accepts both 3.3V and 5V power source and input signal voltage, for that reason no converter is needed between two devices transforming the power and signal voltages. Running on lower voltage results in the shorter maximum detection distance, but even on 3.3V it is 400cm according to the specification [42]. Micropython code for distance measuring was composed and the task was performed – distance readings from ultrasonic sensor were readable by camera board and had believable values as test results.

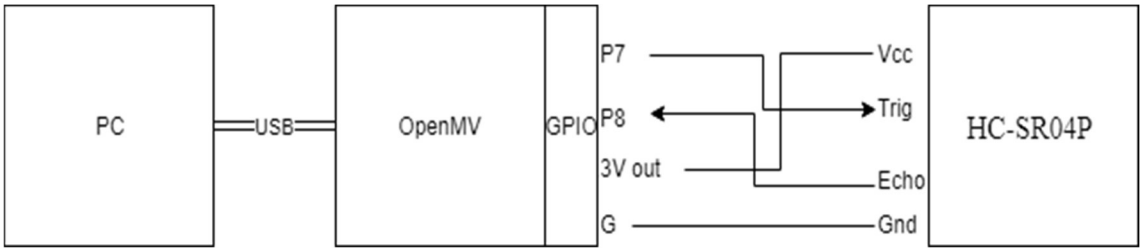

Figure 10. Connection cables between OpenMV and HC-SR04P modules

3.3 Image upload prototype experiment

One of the goals of this thesis was to experiment with image uploading over NB-IoT as explained in the previous work [7], it might be problematic. The OpenMV camera and

Avnet modem board were connected with wires. UART3 pins were used on the camera and PMOD pins on the modem. RX from one side was connected with TX on the other side and vice versa. Also ground and power endpoints were connected so the modem would be powered through the camera board. Connections between boards can be seen on Figure 11.

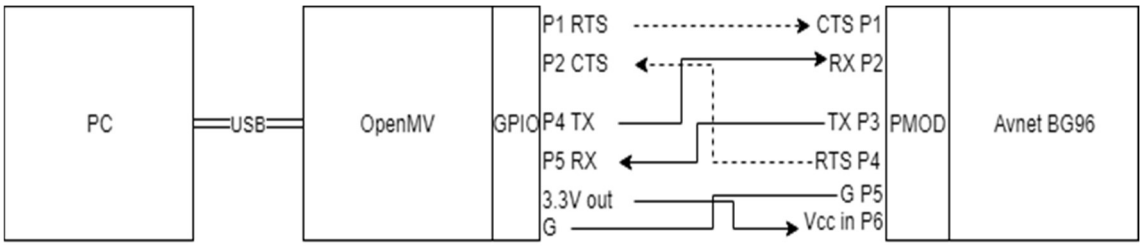

Figure 11. Connection cables between OpenMV and Avnet boards during tests

Inspired by previous work's [7] problems with image uploading, idea of implementing UART hardware flow control was tested. Flow control function's main purpose is to prevent overfilling a buffer on modem side, especially when uploading images. For that purpose CTS pin of one device was connected with RTS pin on other and vice versa. After the hardware flow control was enabled on modem, the UART connection initiation was initiated from camera module, but it was unsuccessful. Therefore, empirically found well-functioning round UART timeout values were used: 1 second for NB-IoT and 100 ms for Cat-M1. This timeout parameter can be defined as time to wait between UART characters while writing or reading. That means that each response reading from UART takes at least 1 second in case of NB-IoT, not matter how fast the command was executed and responded to by modem.

To set up a receiving end, a free cloud server was registered in Google Cloud Platform. As RDP (Remote Desktop Protocol) port 3389 was open by default in the virtual firewall, that was used in testing. A simple python script was composed in that server to listen on port 3389 and write the received data to a file — it is shown Figure 12.

```
import socket, binascii
TCP_IP = '0.0.0.0'
TCP_PORT = 3389
BUFFER_SIZE = 8192
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((TCP_IP, TCP_PORT))
s.listen(1)
conn, addr = s.accept()
print ('Connection address:', addr)
f=open("pilt.jpg", "wb")
while(True):
    data = conn.recv(BUFFER_SIZE)
    if not data: break
    print ("received data:", data)
    f.write(bytearray(binascii.unhexlify((data))))
print("saved to pilt.jpg")
conn.close()
f.close
```

Figure 12. Receiver script on server side

At first some general AT commands were tested and modem settings reviewed directly by using the program Realterm, which is a serial terminal for PC shown on Figure 13. Modem was connected to a PC by a regular USB cable. After initial experimenting, a MicroPython script for camera module was compiled. Script executes the following operations:

1. Initializing the camera sensor and taking a snapshot
2. Compressing the image to JPG format and converting it to a string from binary form
3. Initializing UART connection with the modem and sending some inquiring commands to validate the state of the modem and its connectivity status
4. Instructing the modem to initiate the TCP connection to given destination
5. Sending the data towards the server by breaking the image to suitable packet size parts and executing the sending AT command cycle until the everything is sent to the modem
6. Modem is being interrogated to report received and sent bytes statistics
7. TCP connection and subsequently UART connection stopping

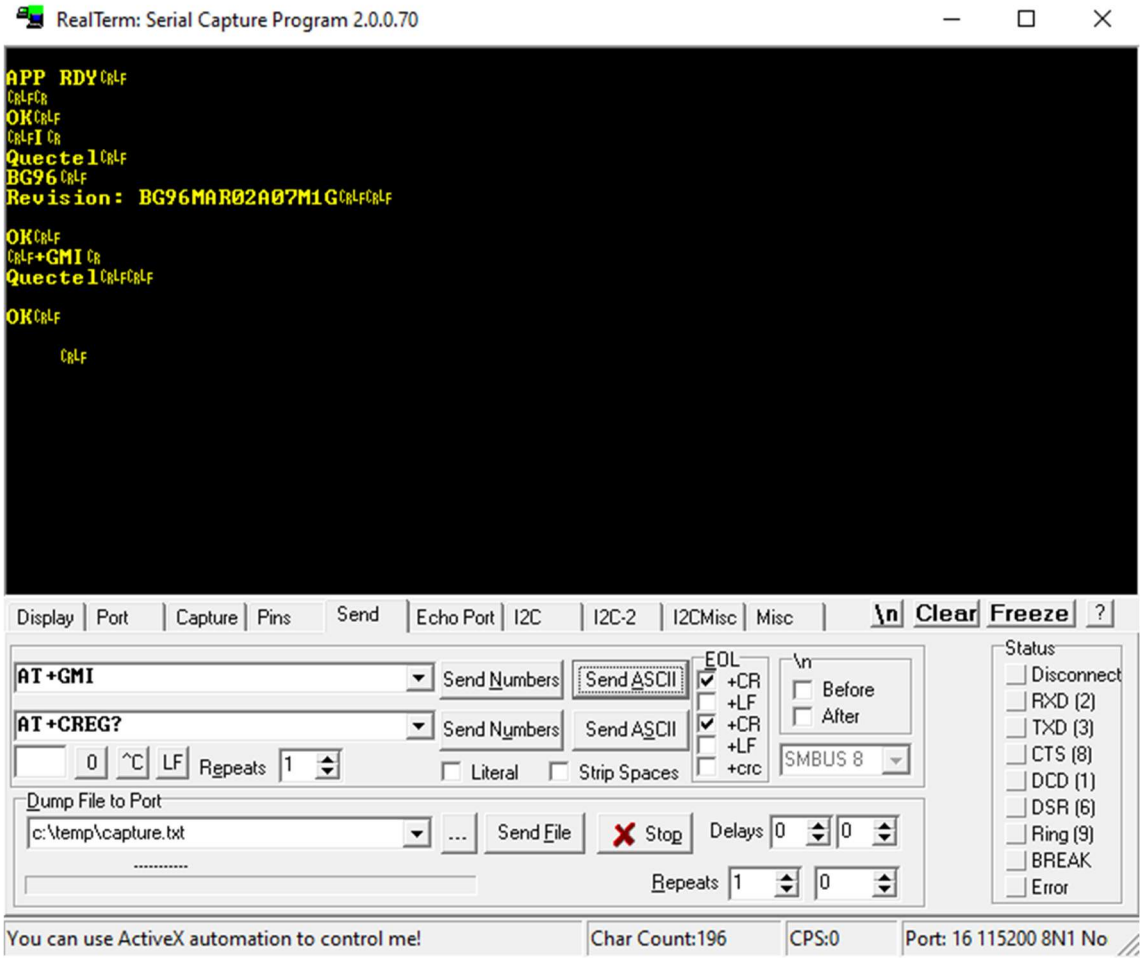

Figure 13. Realterm serial terminal

OpenMV camera is storing the taken image in memory in binary format. In order to send the image by AT commands to the modem, it has to be converted to string format. There are several binary-to-text encoding methods available, one of the most known is the hexadecimal format, also known as Base16. Instead of only using numbers 0-9, this method uses also letters A-F to denote the possible values of 4 bits. E*very byte of data is converted into the corresponding 2-digit hex representation. The returned bytes object is therefore twice as long as the length of data* [43]. As by definition, Base16 uses 2 bytes for every 1 byte of useful data, which in turn means double the volume to be sent over the radio interface (compared to the useful data volume) and that could be considered wasteful. Another common encoding method used for these purposes is Base64 [44], where each Base64 digit represents 6 bits of input binary shown on Figure 14. Due to dividing bytes up unequally between two Base64 symbols, this encoding is more complex to encode and decode than Base16, where always each byte is converted to 2 Base16 symbols. To overcome this problem, Base64 uses padding symbol "=" which is used to

indicate how many symbols the last group of four Base64 digits represent, because it is not warranted that input data length is multiple of 6 bits. Advantage of Base64 is smaller overhead compared to Base16 – on average there are 4 Base64 digits per 3 input symbols. Base16 was used in all the following tests.

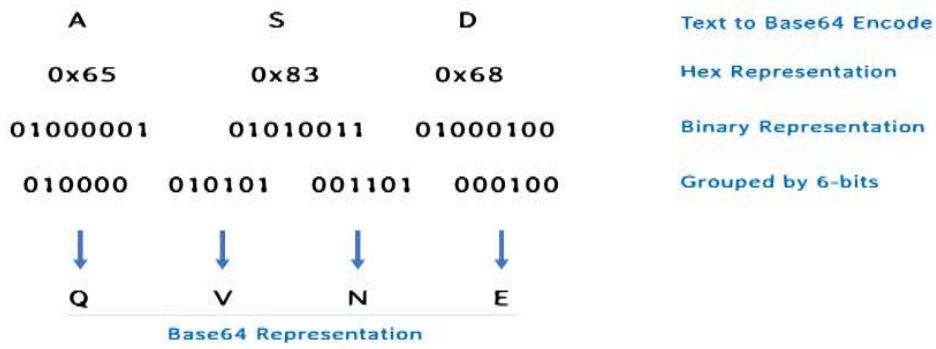

Figure 14. Binary vs Hex [45]

There are two possible BG96 AT commands available in the modem documentation for the fifth step – sending the data to modem [46]. The first one is "AT+QISEND", which is a general TCP data sending command with maximum packet size of 1460 bytes according to Quectel documentation. Testing different send string sizes indicated, that it would indeed be safe to send up to 1460 bytes of useful data to get the data transferred over the radio interface by BG96. Example of sending a string to be transferred to some IP network destination over NB-IOT network from OpenMV board can be seen on Figure 15. The number 0 denotes the connection identificator and 5 the number of characters to be sent.

```
uart.write("AT+QISEND=0,5\r\n")
Response:    b'AT+QISEND=0,5\r\r\n> '
uart.write("abcde")
Response:    b'abcde\r\nSEND OK\r\n'
```

Figure 15. Command and response of sending data over NB-IoT by BG96 modem

The alternative command is "AT+QISENDEX" for sending the data in hex string format with the max data length 512 bytes according to the manual, but testing confirmed packet sizes up to 500 bytes to be certainly working and with possible problems with packet size larger than that. This command is more convenient to use as the data to be sent is part of that same AT command as shown on Figure 16, which is not the case for the previously mentioned method.

```
uart.write('AT+QISENDEX=0,"4C6F"\r\n')
Response: b'AT+QISENDEX=0,"4C6F"\r\r\nSEND OK\r\n\r\n+QIURC:
"closed",0\r\n'
```

Figure 16. Command and response of sending data over NB-IoT by BG96 modem 2

Packet size of a TCP packet is limited and whereas the target is to upload images, then unless in case of very small grayscale image and/or very high compression on it - it is not possible to upload images in a single packet. Therefore, it is needed to divide the string, which is converted from binary image, to parts which fit into maximum allowed packet size limit. The string will be sent part by part and reattached again on the server side, where it can be converted back to binary form and saved to image file.

Packet trace printout is shown on Figure 17, which from cloud side allows to conclude that even though the modem accepts 1460 bytes large input, it is being delivered in 2 separate packets — 1400 and 60 bytes — which could mean that the maximum packet size is 1400 bytes BG96 on TCP protocol or some router between the modem and the google cloud server had a MTU (Maximum Transmission Unit) of 1400 bytes.

```
22:37:45.858371 IP 46.131.59.255.16089 > 10.166.0.2.3389: Flags [P.], seq
1:1401, ack 1, win 16384, length 1400
22:37:45.858491 IP 10.166.0.2.3389 > 46.131.59.255.16089: Flags [.], ack
1401, win 245, length 0
22:37:45.976464 IP 46.131.59.255.16089 > 10.166.0.2.3389: Flags [P.], seq
1401:1461, ack 1, win 16384, length 60
22:37:45.976545 IP 10.166.0.2.3389 > 46.131.59.255.16089: Flags [.], ack
1461, win 245, length 0
```

Figure 17. Packet capture from server side

Five main tools used in process are shown on the Figure 18. MicroPython was written in OpenMV IDE, which enables to run it in USB-connected camera module and shows the printable output from running the code in IDE terminal (b). The captured picture (c) is transferred from the camera board's frame buffer to IDE to assist troubleshooting. Besides those tools, also the printout of server-side script from Figure 12 outputs the relevant info about the connection status. And finally the tcpdump packet capture was ran in the cloud server to monitor any packet traffic between the server and the NB-IoT modem. Uploading the images over NB-IoT and using different AT commands for TCP connection succeeded.

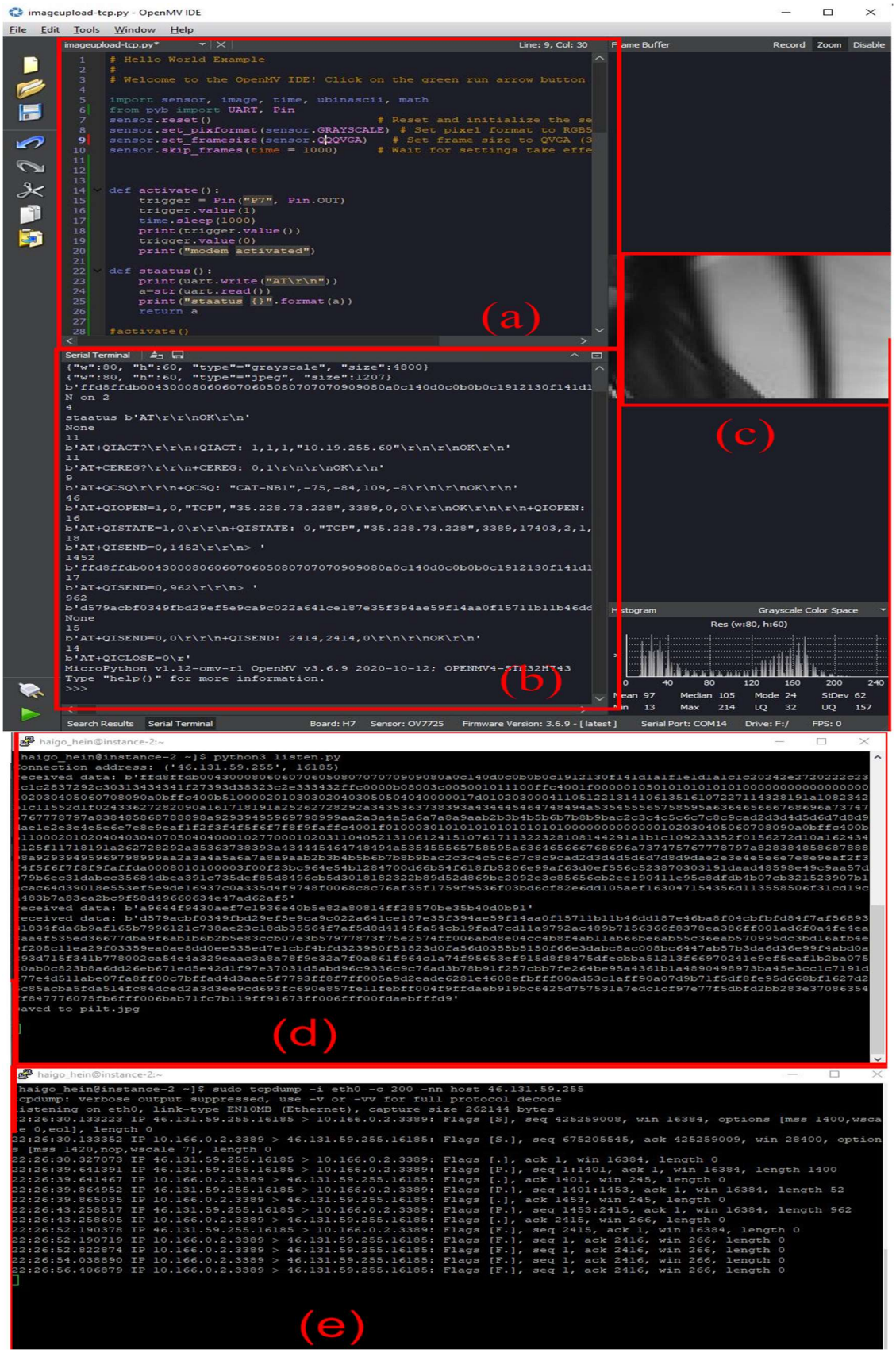

Figure 18. Experiment setup where is (a) program code, (b) serial terminal, (c) picture from frame buffer, (d) printout from script from (Figure 12) and (e) tcpdump printout

4 Proposed system

In this section a waste container fill level monitoring prototype is proposed. Examining the relevant research papers and different methods presented there concluded in realization that camera-based solutions are often not considered as camera would need illuminated environment, extensive image processing and/or machine learning algorithm and possibly relatively large upload bandwidth and volume, which are all countering the purpose of low-power mobile sensor actually being a low-power device, which could work for years without power source replacement. To overcome these obstacles, an approach to use the camera and its input in more efficient way is presented. The solution was planned while having in mind the standard 660L container shown on Figure 19 and used by apartment buildings, offices and larger households. Proposed system would be practicable also in different-sized containers, but it's potential usefulness is evaluated using shape and measurements of this make.

Figure 19. 660L waste container [47]

4.1 Proposed sensor placement and system architecture

The general idea for the prototype is to use the camera as a sensor stationed on the wall of the container and looking towards the other wall. It would be positioned at the top edge of the container. The ultrasonic and IR sensors are used in most of the reviewed solutions, but they have drawbacks. IR sensor is able to only detect change on one certain level –

are the contents of the container over certain height or not. Ultrasonic sensors on the other hand are commonly attached to the cover of the container and looking towards ground – measuring the distance to waste. Shortcoming in here is the narrow beam and large uncovered area around it, which means that in larger containers most of the cubage would not be under monitoring. Using a camera facing the other side of the container would effectively act like a group of IR sensors instead of one or two commonly used. Number of virtual IR sensors detecting the fill level would be equal by the number of vertical pixels taken by the camera. The area covered on the opposite side wall by the camera would be determined by the field of view of the camera lens. On the Figure 20 is depicted the vertical field of view of OpenMV H7 regular lens (55.6°), where D_1 is the location of the camera, A_1B_1 is the bottom of the container and D_1A_1 is the side where camera is mounted. The height and the width of the container are equal in this example and the whole camera lens' view is used, it can be seen that most of the area inside are monitored.

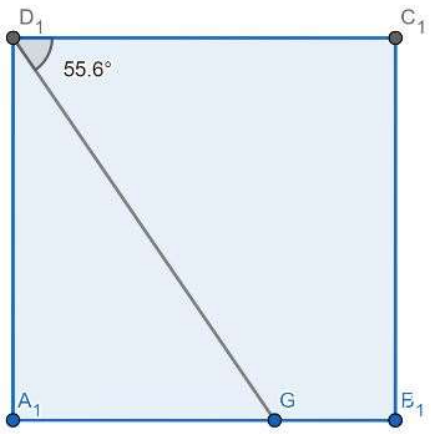

Figure 20. Field of view of OpenMV regular lens

To indentify if the whole opposite side of the container could be observed by the camera in specific settings, the common trigonometric function tangent (1) can be used to calculate the theoretical maximum viewable wall height. If the found maximal viewable height x is bigger than the height of the container wall C_1B_1, then it is true – the whole opposite wall is observable. If we consider the information about our example container from Figure 19, which has internal measures 1088x1257x776 mm [48], then the maximum viewable height of the other side is 1133mm, which is more than the containers height 1088mm.

$$\tan(\alpha) = \frac{x}{\text{width of the container}} \quad (1)$$

The vertical field of view would also be dependant on how far is the camera from the other side – the width of the container. By knowing the camera's vertical viewable angle and inserting into the equation (1), it is possible to estimate the viewable height x dependency from the width of the container (2), where tan(55.6°) is aproximately 1.46.

$$x = 1.46 \times width \quad (2)$$

That means maximum viewable side wall height is around 1.4 times of the width of container.

Similarly, taking under consideration that OpenMV camera has 70.8° vertical FOV, it is possible to estimate if the camera would be able to observe the whole length of the waste container. View from the top is depicted on Figure 21, where C is the location of the camera, AE is the width and AB length of the container.

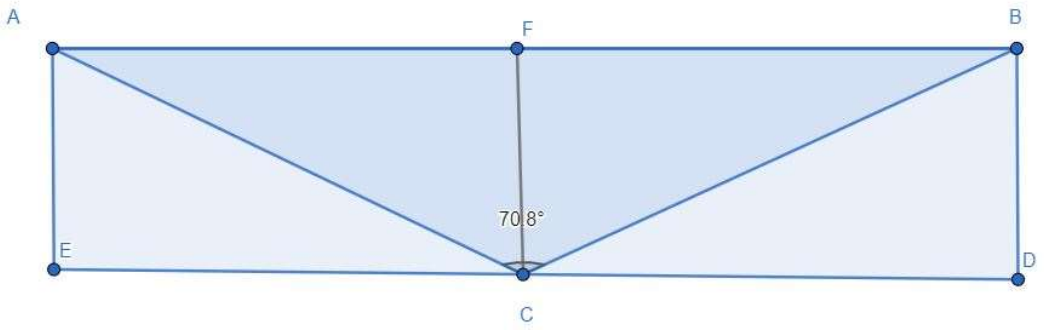

Figure 21. Top view perspective field of view

Furthermore, we can calculate the viewable area dependency from the width of the container. It can be done by dividing the triangle ABC to two triangles and using the tangent function as shown in (3)

$$\tan(\alpha) = \frac{\frac{x}{2}}{width} \quad (3),$$

where α is 35.4°. Therefore, formula for viewable length would be:

$$x = 2 \times 0.71 \times width = 1.42 \times width \quad (4),$$

where 0.71 is tan(70.8°/2). Inserting the width 776mm of example container [48] into that equation, it can be deduced that viewable length of that container would be approximately

1101mm, which is slightly smaller than the length of the container 1257mm, but would be marginal in that case. If the container shape would be even narrower, then it could be sensible to use lenses with wider viewing angles, to be able to use this type of solution. OpenMV is also providing an Ultra Wide Angle Lens [49] accessory, which has 99.0° horizontal and 81.9° vertical FOV.

One of the basic approaches on how to detect the container fill level by using images is measuring the change magnitude between two images captured of empty and not empty container. This kind of system would be fairly simple in principle and would not need machine learning and image processing options available on OpenMV H7 board. On the other hand, undemanding algorithm could possibly be more energy efficient and reliable.

There are two major downsides of this setup. The first one is that proposed solution would also have a blindspot – right under the device itself. The second one is related to the problem that there would be no sense of depth understanding by the camera. So, if a thin and tall object would be placed in the view of the sensor, the algorithm would report vastly incorrect fill level. To overcome these problems, the ultrasonic sensor will be used coupled with the camera module and the problems would be largely mitigated by that sensor. The comprising FOV would be more complete and it would also assist on detecting the false positive readings from the image analysis, which are caused by the second problem. The aggregate FOV is depicted on Figure 22, where D is the location of the sensors, DA and CB are the walls of the container and AB the bottom. Area observable by ultrasonic sensor would be DEF and by camera DGBC.

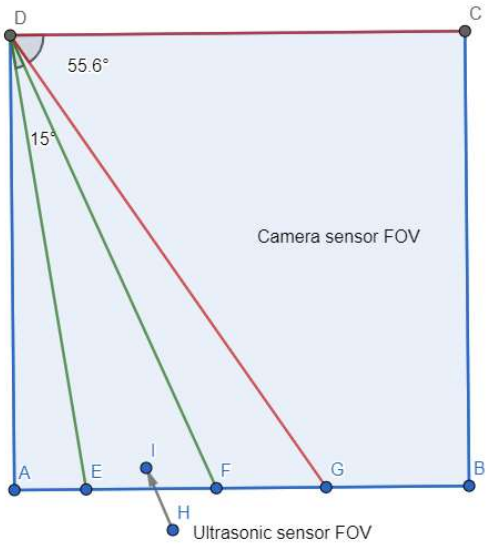

Figure 22. Field of "view" with ultrasonic sensor

On general architectural view there is also a principial choice of where to do the data processing and decision making. One option is to deliver all the data to the cloud and process it there. The second option would be to do all the processing and decisions in the edge device and just send the results to cloud. Thirdly, it is also possible to make a compomise between those two approaches. Transferring all the data could be too resource intensive regarding both data bandwith and energy use. When considering the first option in this use case, then the images would be uploaded and it could be expected, that it is more costly to transfer them than process them locally. Although processing locally would mean less demanding data transfer, it could have more power usage due to running the algorithm(s) locally. Both the first and second arhitectural option are trialed in section 4.3, but it is expected that uploading the images over the NB-IoT could be too costly compared to running the algorithm locally.

4.2 Proposed prototype algorithm

This section will mainly discuss about two processes: general prototype working cycle and the procedure of determing the fill level based on image differences. The general cycle is illustrated on Figure 23. It is expected that taking the reference image is not part of the cycle, it is separate procedure which should be automated and be resultant from the emptying of the waste container, but it is not analyzed from that aspect in this thesis.

According to the flowchart, using the ultrasonic sensor to measure the distance towards the bottom of the container is always done first. Based on that reading it is decided, if any further action should be taken. Decision is based on preset constant D, which is representing the level on distance after which the image analysing process would be performed. Following that, the result will be uploaded to cloud server over NB-IoT modem. Constant D could be for example one third or half of the height of the container it is a compromise between battery saving and monitoring accuracy. For container on Figure 19 it could be 70 cm for example — it would mean that the bin is not empty and has some content in it before starting capturing images. Eliminating the need for always taking picture, analyzing it and uploading the result will spare the power source greatly. There could also be other criteria to decide if the result(s) should be uploaded. For example there could be a condition that the sensor needs to send at least one reading to the server per selected numer of hours, which would then function as a heartbeat signal indicating the device is still working to an application service.

Part of that process is fill level estimation procedure, which consists of four general steps:

1. Taking a picture
2. Calculating the difference image (pixel-by-pixel subtraction) compared to reference image
3. Finding the first row with significant difference starting from the top row and moving downwards
4. Calculating the fill level based on the row position

This procedure is more extensively explained in section 4.3.1.

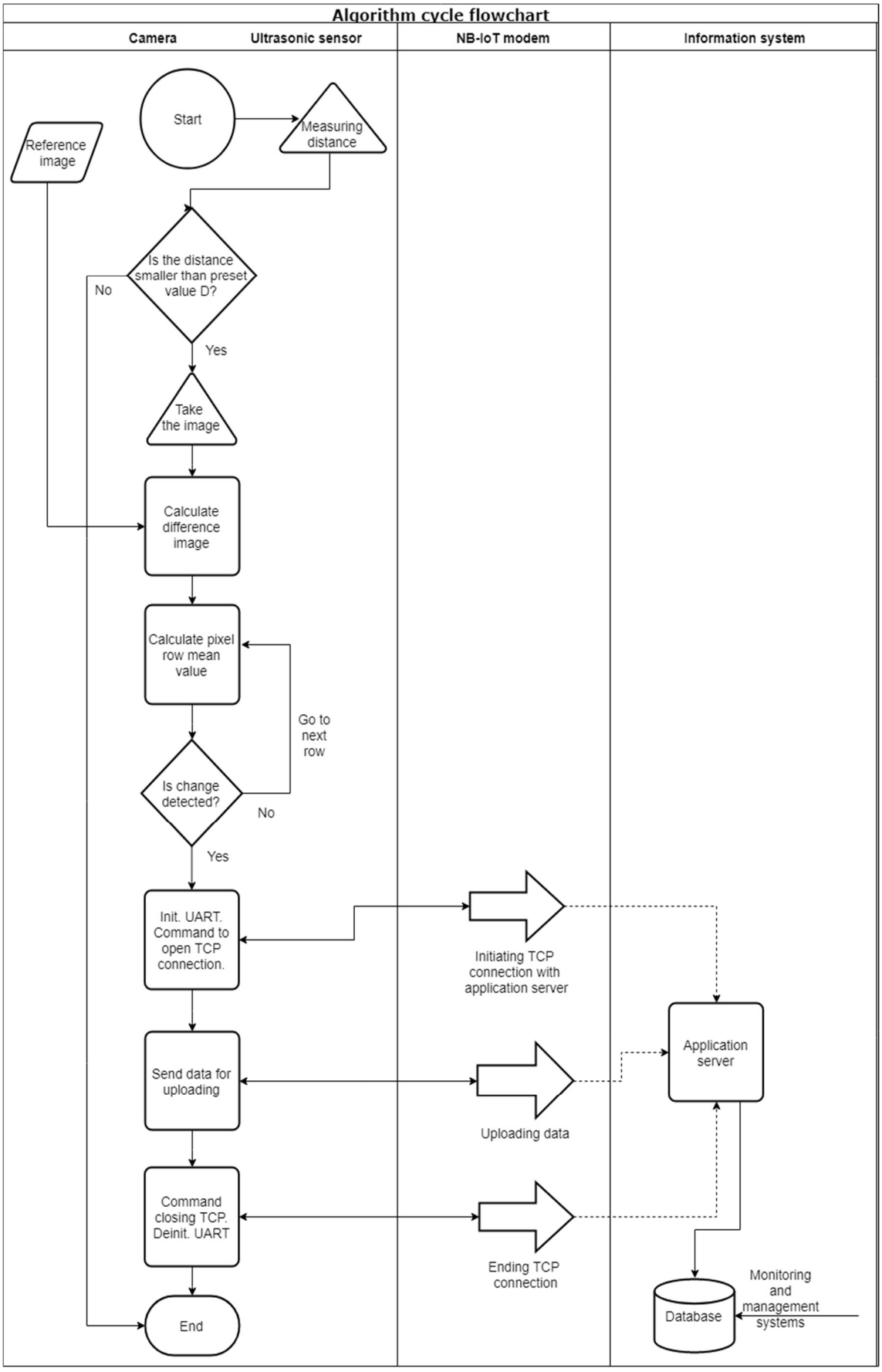

Figure 23. Flowchart of the prototype algorithm

Important considerations in fill level estimation process are:

- Are grayscale images suitable for it compared to color images?
- What image size is sufficient and small enough at the same time to be procecced by a processor having limited working memory capacity?
- Is fill level estimation based on difference calculation possible in intended conditions?

Grayscale images should be prefered for two reasons. One is that grayscale images take less storage space, which is a valid concern on mobile devices both from local processing and transferring standpoint. Second is that in real application the pictures should be taken in darkness and doing that with the help of IR light does not have a reason to save that image down in color format, as it will naturally be rather black and white. Regarding image size, it was learned that proposed algorithm has enought memory to work with up to QVGA (320x240 pixels) grayscale images while the reference image is saved and read from a memory card. The algorithm could probably work with smaller number of pixels also, but following the intuitive logic that more the input data, better the evaluation result. And finally, the operation of fill level estimation process was experimented and is further discussed in the next section.

4.3 Evaluation of proposed system

The idea of using the camera for monitoring the fill level was assessed with the simple level estimation algorithm described in previous section. In order to simplify conducting the simulation – the camera was placed horizontally and therefore only half of the field of view was useful Figure 24, which results in a narrow area of observation. On the Figure 24 D is the position of camera, CB the opposing wall, AB the floor and CE the observable area.

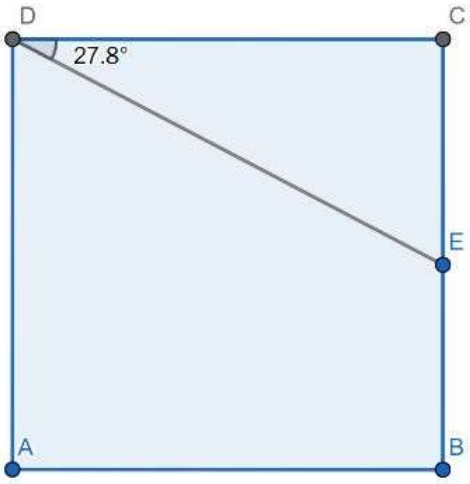

Figure 24. Test setup field of view

The device was attached at the height of 62cm from the floor and stationed around one meter from the opposing wall as shown on Figure 25. If we use the equation (1) to apprehend how large part of the opposing side is viewable:

$$x = \tan(27{,}8^{\circ}) \times 100 \approx 53 \quad (5),$$

which means that viewable height is theoretically 53 cm and less than the height of „virtual“ container. That means that observable area on the opposite wall does not reach to the floor and when coming towards the camera FOV narrows even further. In the middle of the „virtual“ container the observable height would be approximately only 26 cm.

Figure 25. Virtual container

4.3.1 Evaluation of the algorithm

The idea of the process is to take an initial picture of an empty container and start comparing the images taken in new conditions with it. The grayscale color format was used, despite the initial idea to use 1-bit bitmap images, which would have reduced the image size drastically both for the local processing and also uploading it to the server and most probably would have given advantage to cloud computing. 1-bit bitmap image means that pixel value can be only 1 or 0, white or black. The problem encountered was that saving and processing bitmap images was not sufficiently implemented in OpenMV MicroPython libraries.

The algorithm takes the reference image and finds the difference with the new grayscale image – it is essentially pixel by pixel subtraction and the result of it will be called „difference image“ in this thesis. Example of reference image is shown on Figure 26. The grayscale image pixel can have value from 0 to 255, where value 0 means black and 255 white. After subtraction the cycle go through „difference image“ pixel rows starting from the top. If there is no change in the camera FOV, then the calculated „difference image“ should be solidly black meaning pixel values should be 0 or close to it.

Figure 26. Reference image from (a) OpenMV and (b) setup illustration from a regular camera

A low complexity method to measure if the row has significantly changed compared to the reference, is to look at the mean (or median) pixel value from the difference image – if the mean is notably greater than 0, then the change between the images is detected. Starting the cycle from the top of the image it is possible to evaluate the entirely unchanged area by stopping the cycle when change on a row is larger than predetermined difference amount. The algorithm was trialed by adding objects to imaginary container in

the FOV of camera. Difference images and real situation by the estimated fill level are reported in Figure 27.

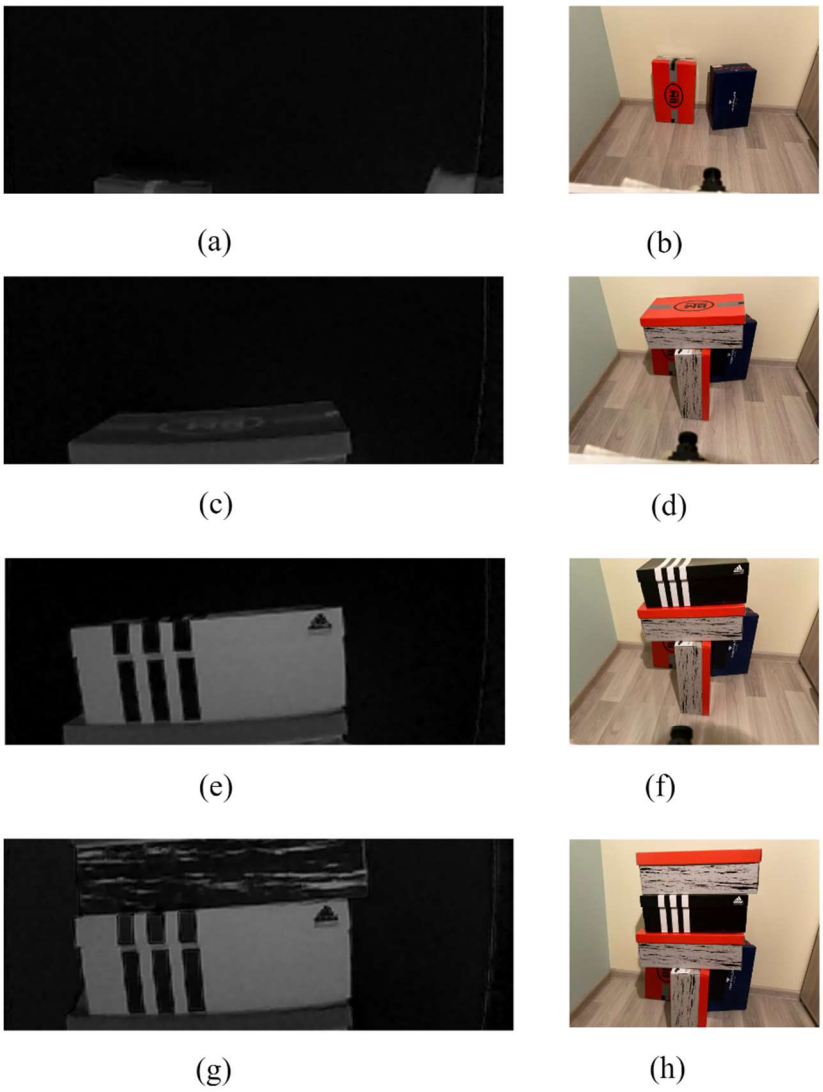

Figure 27. Pairs of "difference images" and images describing objects positions on different reported fill levels: (a) and (b) 10%; (c) and (d) 30%; (e) and (f) 73%; (a) and (b) 100%

Results show that the algorithm does what it was intended to, but also hightlights the deficiencies expected while designing this type of solution. The largest downsides are large blind spot under the camera – especially on this setup where camera is placed horizontally and the whole lens view is not used – and inability to recognize the depth of view. Approximate field of view is shown on Figure 28. It could be argued that proposed

method could work relatively well even without ultrasonic sensor in containers where household trash is deposited. If some tall thin object would be placed in the view of the camera, then the level estimation would obviously be erroneous, but household waste is predominantly disposed in form of trashbags containing small-scale objects, so generally it could be expected to be trustworthy. On the other hand, this kind of system uses two sensors, camera and ultrasonic, but in some cases still rely only on latter, then the usefulness of adding a camera might be questionable.

Figure 28. Approximate field of view

4.3.2 Image compression aspect

If the monitoring architecture would foresee uploading the images to the cloud and processing there, then is the size of images even more crucial than in the local processing variant. Every byte would be uploaded over the NB-IoT radio interface, which is not intended for large-scale data communication. For example, empirically found energy consumption of uploading 10 bytes of data over NB-IoT using BG96 based shields would consume in range of 0.110 to 0.816 mWh [50]. Therefore, compression options and effects on OpenMV H7 were explored. OpenMV machine vision library offers JPEG compression format, which was used for evaluation [51]. This function has input parameter „quality“, which can be from level 0 to 100 while 100 is for the best image quality and less compression and 0 is for the worst image quality and best compression. Image size dependency on compression quality parameter is presented in Table 6, where it can be seen that even compressing with good quality level (90) the image consumes

approximately ten times less space in bytes. In case of the lowest quality value tested, the image size is still larger than the maximum allowed packet size 1460 bytes specified in section 3.3.

Table 6. Image compression effectiveness

Compression quality parameter	**Image size before (grayscale binary image), bytes**	**Image size after (JPEG), bytes**
90	76800	7931
70	76800	4156
50	76800	3238
30	76800	2556
10	76800	1802

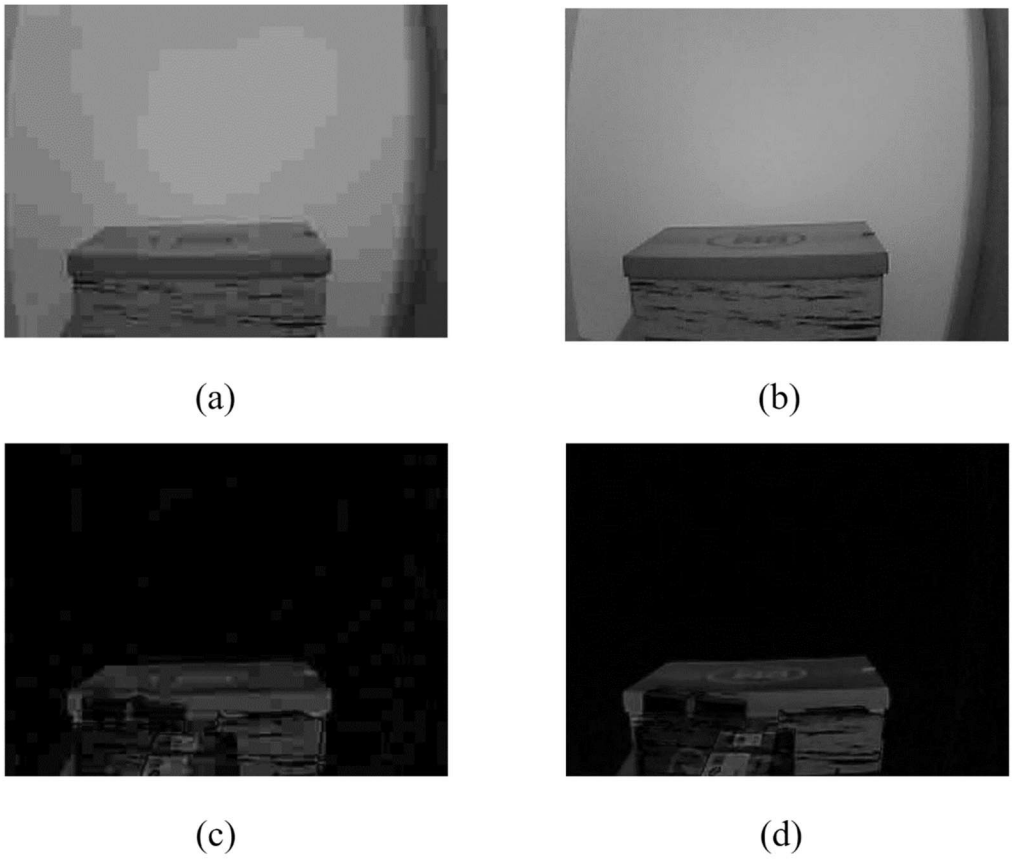

Figure 29. Compressed images with compression level (a) 10 and (b) 90 from which "Difference images" are calculated: (c) from quality level 10 and (d) from quality level 90 images

Simulating the cloud server calculation of „difference image“ from JPEG compressed images resulted in conclusion, that higher compression rate is not a problem with current simple approach. Results of calculation of „difference image“ where an item is added to the FOV after capturing reference photo is shown on Figure 29. Compression distorts image significantly, but it does not seem to cause a problem for the proposed algorithm at compression quality levels neither being 10 or 90, the resulting image is roughly the same.

4.3.3 Energy consumption test scenarios

In order to evaluate whether to do the processing part in the edge, where resources are limited, or in the cloud, where resources are easily available, the energy consumption test was performed. The measuring was done by adding an „USB meter“ device shown on Figure 30 between the power source (personal computer) and the OpenMV board.

Figure 30. USB meter

The device’s voltage and current measuring parameters are presented in Table 7.

Table 7. USB meter data sheet

Voltage measurement range	4-24.000V
Voltage measurement resolution	0.01V
Current measurement range	0-5.0000A
Current measurement resolution	0.001A

The ground for the experiment was already previously used setup which includes both the Avnet board getting the power from the OpenMV and in turn it powered by USB cable as shown on Figure 31.

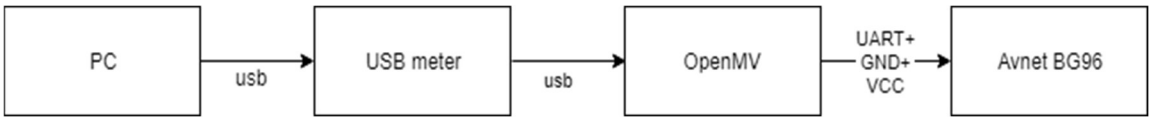

Figure 31. Energy consumption test setup

Test scripts were composed for administering the power usage comparison based on scripts already used in image upload in section 3.3 and algoritm section in 4.3.1. All together three scenarios were compared, which are described in following paragraphs.

„QVGA upload“ scenario is based on image upload script from 3.3 and consists of followig steps:

- QVGA grayscale image is taken
- Image is compressed using lossy JPEG format with quality setting 75
- Compressed image is converted to hexadecimal string
- Hex string is uploaded by 1460 byte chunks
- Cloud server printout was monitored to validate if the image upload succeeded on the receiving end.

„QVGA cropped upload“ scenario is almost identical to previous scenario except image cropping done as second step:

- QVGA grayscale image is taken
- Image is cropped so only lower half of image is further processed
- Continues as „QVGA upload“

Amount of bytes uploaded during the first two scenarios and size of uncompressed image was following:

- Image size before compressing: 76800 bytes
- Image size after compressing in „QVGA upload“: 4452 bytes

- Image size after compressing in „QVGA cropped upload“: 2314 bytes

Local processing scenario is based on the fill level detection script from section 4.3.1 and consists of following steps:

- QVGA grayscale image is taken, cropped to lower half and saved onto the memory card – only done once as this will be used as reference image
- QVGA grayscale image is taken
- Image is cropped so only lower half of image is further processed
- Difference image is calculated
- Starting from the top row of the difference image – each row’s mean value is compared to a reference value
- When the row which falls into the criteria is selected (or no change is found in the whole image), the result string is formulated
- Result string is sent to the application server

The difference between the first and the second scenario is describing the effect of the upload data size on power consumption. The contrast between the second and third option can be basis to choose the prototype service setup architecture: cloud or edge computing.

4.3.4 Energy consumption test results

The results are presented in Table 8. The results presented were achieved by running the test scenarios from 4.3.3 five times each and averaging the results of recorded power usage indicated by the USB meter.

Table 8. Energy consumption

Scenario	**QVGA upload**	**QVGA cropped upload**	**Local processing**		**QVGA cropped upload (Cat-M1)[3]**
			Partial[1]	**Full[2]**	
Average consumption, mWh	11	8.4	<1	5.8	2.43

[1] Data is not uploaded, only local processing part is done.

[2] Full procedure of processing and uploading the result is done.

[3] Not directly comparable with other scenarios

As expected, measurement results show that with the current setup the most efficient would be to do the image processing in the edge device and transmit only the result to the central information system. It is not an evaluation of the energy usage of Avnet or OpenMV board specifically, but compares energy consumption between specified competing scenarios. Difference between "QVGA upload" and „QVGA cropped upload" scenarios is not in the range of 50% as it could be, because approximately half as less bytes were uploaded during the second scenario. It could be explained by data also found in empirical studies, where energy consumption per transferred byte was smaller when packet size was larger and the number of packets sent in a batch bigger. It could also explain why there was recorded energy usage of 0.11 mWh uploading 10 bytes, as referred in section 4.3.2. NB-IoT data transfer per byte energy usage is illustrated on Figure 32, where it can be seen that it is relatively costly to upload small data amounts. Maximal packet size for network in the referred study was 512 bytes, that is why the tests were done in a range of 12 to 500 bytes packet size. From the results it can be deduced that for small packet sizes the energy consumed for the transfer of useful bits is marginal compared to the radio connection procedures [52].

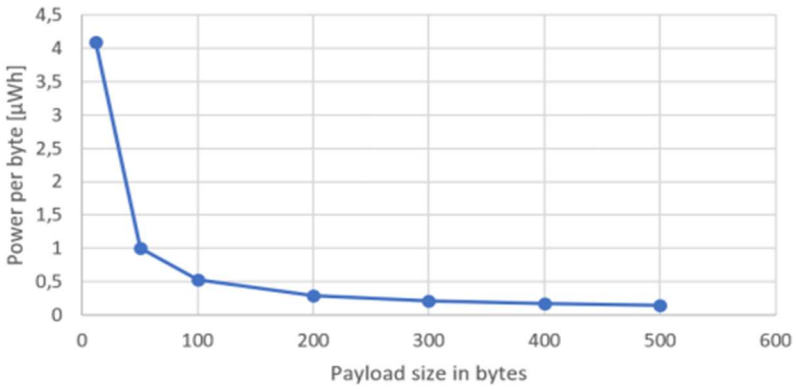

Figure 32. Power per byte for a single packet transmission [52]

Although Cat-M1 was more than two times efficient in the same "QVGA cropped upload" scenario, NB-IoT and Cat-M1 are not directly comparable here, because UART timeout used was 10 times smaller in Cat-M1 case as mentioned in section 3.3, which means that procedure involving the Avnet modem consumed significantly less time — equals less energy used. It is not possible to run with a same timeout on Cat-M1 as was done on NB-IoT, because the 1 second timeout means that commands take too long and LTE radio

resources are released before the script is finished. Besides that, by design Cat-M1 provides significantly better transfer speeds than NB-IoT. Furthermore, it has been found in simulations, that Cat-M1 can be more energy efficient in certain conditions as shown on Figure 33. eMTC (Enhanced Machine Type Communication) a.k.a. Cat-M1 has better energy efficiency in good coverage conditions and especially when larger amount of bytes (10^3 bytes) are being periodically transferred [53].

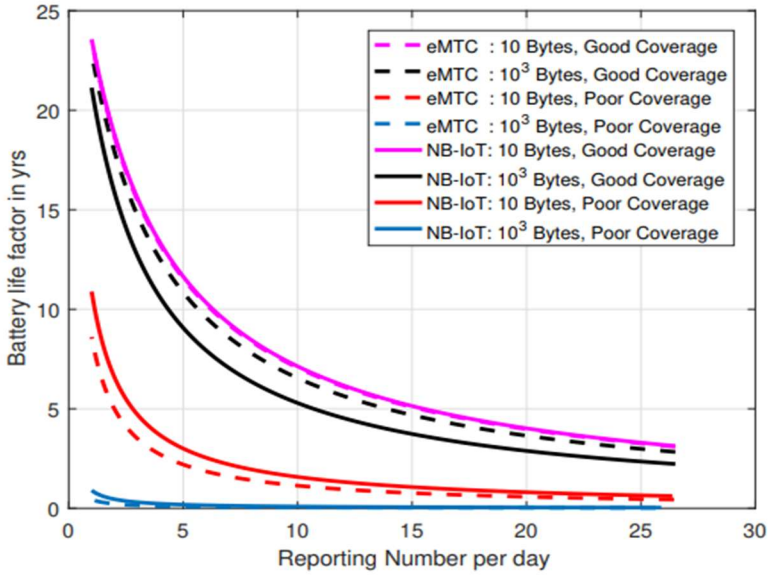

Figure 33. Battery life time under different reporting number, data lengths and coverage scenarios for eMTC and NB-IoT [53]

In conclusion, uploading images over NB-IoT is possible, but not advantageous compared to Cat-M1. When transferrable data amount is small (especially if it fits into one packet) and high latency is not a factor, then NB-IoT is preferred due to better coverage and energy efficiency — especially in poor coverage conditions. In framework of proposed prototype system, local processing would be preferred over uploading the whole images to application server over NB-IoT, but generally Cat-M1 should be preferred for uploading images as it is more suitable for it by design.

5 Summary

The main purpose in this thesis was to investigate the image upload feasibility over NB-IoT in context of a smart waste management sensor device, more specifically fill level detection sensor. At first, the background of the waste management and problems revolving around it were shortly presented. Also reference to and short overview of previous similar thesis done in Taltech is given. In the state of art chapter, smart waste management sensors are more closely inspected by introducing sensor systems discussed in relevant scientific papers. Also some commercially offered smart waste management fill level detection devices, which had some public technical information about offered solution were presented.

A prototype system for waste container fill level measuring system was proposed. It included two sensors: ultrasonic distance sensor and camera. Suggested placement of sensors was upper edge of the waste container side, which is less utilized. Widespread solution is to attach some type of distance sensor under the lid of the container. Proposed algorithm for this type of sensor would use ultrasonic sensor's distance measuring only to save energy and override the deficiencies of a camera sensor. Camera sensor would be evaluating the container's fill level by measuring difference of empty and filled container insides. This kind of method has shortcomings comparable to single ultrasonic sensor under the lid — but the cost of the device would be bigger with camera and ultrasonic sensor both employed. The best option currently seems to be multipixel ToF laser sensor attached under the container's lid, which would perform better than single ultrasonic module.

Prototype's architectural decision between cloud or edge computing was reviewed from energy consumption perspective. Testing given scenarios of uploading images or processing result data over NB-IoT suggested that in these specific settings local processing would be advantageous, but for image uploading overall, Cat-M1 would be most efficient.

As a result of this thesis some questions emerged during writing it which were unsolved, not tried or unanswered and will remain for future work:

- UART hardware flow control implementation in communication between the modem and camera board (or any other controller) could lead to faster request-response flow, especially while modem is in NB-IoT network, which would make the prototype's scripts run faster and save energy by that.

- Another question to be solved is how to send data to modem by AT command with minimal overhead. Currently Base16 binary-to-text conversion was used to transform image object to processable format, but how could it be technically implemented without less or any overhead?

- ToF optical sensor-based fill level measuring method appears to be the most beneficial on present day. Prototype employing this kind of sensor paired with NB-IoT transmission module while aiming for minimal cost in all aspects – components included, could be designed.

References

[1] World Bank, “Trends in Solid Waste Management,” [Online]. Available: https://datatopics.worldbank.org/what-a-waste/trends_in_solid_waste_management.html. [Accessed 7 11 2020].

[2] United Nations, “World Population Prospects 2019,” [Online]. Available: https://population.un.org/wpp/Graphs/Probabilistic/POP/TOT/900. [Accessed 7 11 2020].

[3] United Nations Environment Programme, “Global Waste Management Outlook,” 2015.

[4] LINK LABS, “What Is Smart Waste Management?,” 11 09 2015. [Online]. Available: https://www.link-labs.com/blog/smart-waste-management. [Accessed 31 12 2020].

[5] S. Verma, “How Smart Waste Management is Making Waste Collection Efficient,” RTInsights, 16 09 2019. [Online]. Available: https://www.rtinsights.com/iot-makes-smart-waste-management-efficient/. [Accessed 31 12 2020].

[6] Y. Q. S. R.-M. A. V. Charith Perera, “Fog Computing for Sustainable Smart Cities: A Survey,” *ACM Computing Surveys,* vol. 30, no. 3, 03 2017.

[7] C. Pehlivan, “NARROWBAND INTERNET OF THINGS BASED IMAGE SENSOR PLATFORM FOR SMART WASTE MANAGEMENT,” TALLINN UNIVERSITY OF TECHNOLOGY, Tallinn, 2020.

[8] Components101, “HC-SR04 Ultrasonic Sensor,” [Online]. Available: https://components101.com/ultrasonic-sensor-working-pinout-datasheet. [Accessed 8 11 2020].

[9] P. A. R. A. Abhimanyu Singh, “IoT based Waste Collection System using Infrared sensors,” Jaypee University of Information Technology, Solan, 2016.

[10] M. S. D. S. M. K. G. B. P. A. P. N. P. V. S. G. Miss. Priya A. Jadhao, “SYSTEM, SMART GARBAGE MONITORING AND COLLECTION,” *International Journal of Advance Engineering and Research,* vol. Volume 5, no. Special Issue 06, 2018.

[11] A. K. A. S. M. S. ,. S. I. S.A. Mahajan, “Smart Waste Management System using IoT,” Department of Information Technology, PVG’s COET, Pune, 2017.

[12] C. H. P. P. D. L. L. M. B. N. N. M. T. N. T. H. P. N. V. D. N. T.-Y. H. N. N. D. N. M. D. Tran Anh Khoa, “Waste Management System Using IoT-Based Machine Learning in University,” Hindawi, 2020.

[13] A. E. M. A. H. Maher Arebey, “CONTENT-BASED IMAGE RETRIEVAL SYSTEM FOR SOLID WASTE BIN LEVEL CLASSIFICATION AND RECOGNITION,” *Journal of Humanities and Applied Science ,* no. Issue No. (27), 2015.

[14] “IoT Enabled Smart Waste Bin with Real Time Monitoring for efficient waste management in Metropolitan Cities,” *International Journal of Advanced Science and Convergence,* vol. 1, no. 3, pp. 13-19, 2019.

[15] L. Li, “Time-of-Flight Camera – An Introduction,” Texas Instruments, 2014.

[16] DeCode Staff, “The Time of Flight Camera is Shaping the Future!,” 17 02 2020. [Online]. Available: https://medium.com/decodein/the-time-of-flight-camera-is-shaping-the-future-f47806aba135. [Accessed 29 12 2020].

[17] S. Tabbane, *Traffic engineering and advanced wireless network,* Suva: ITU, 2018.

[18] Paessler AG, “LPWA,” [Online]. Available: https://www.paessler.com/it-explained/lpwa. [Accessed 25 12 2020].

[19] Telefonaktiebolaget LM Ericsson, “Cellular networks for Massive IoT,” [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/cellular-networks-for-massive-iot--enabling-low-power-wide-area-applications. [Accessed 25 12 2020].

[20] Telenor Connexion, “LTE-M vs NB-IoT – a guide exploring the differences between LTE-M and NB-IoT,” [Online]. Available: https://www.telenorconnexion.com/iot-insights/lte-m-vs-nb-iot-guide-differences/. [Accessed 27 12 2020].

[21] S. Chalapati, “Comparison Of LPWA Technologies And Realizable,” Tata Consultancy Services, 2018.

[22] GSMA, “Security Features of LTE-M and NB-IoT Networks,” 2019.

[23] Ecube Labs , “CleanFLEX, the ultrasonic fill-level sensor,” [Online]. Available: https://www.ecubelabs.com/ultrasonic-fill-level-sensor/. [Accessed 27 12 2020].

[24] Smartsensor Technologies Pty Ltd, “SmartBin,” [Online]. Available: https://www.smartsensor.com.au/sensors/smartbin/. [Accessed 27 12 2020].

[25] SENSONEO j. s. a. , “Waste Monitoring with Smart Sensors,” [Online]. Available: https://sensoneo.com/product/smart-sensors/. [Accessed 27 12 2020].

[26] Nordsense, “Smart Containers,” [Online]. Available: https://nordsense.com/ns-navigator/ns-pod/.

[27] Terabee, “Level Sensing for Smart Waste Management,” [Online]. Available: https://www.terabee.com/level-sensing-for-smart-waste-management/. [Accessed 28 12 2020].

[28] I. Abdelkader, “low tech,” [Online]. Available: https://sigalrm.blogspot.com/2013/07/in-search-of-better-serial-camera-module.html. [Accessed 15 11 2020].

[29] I. A. Kwabena W. Agyeman, “OpenMV,” [Online]. Available: https://hackaday.io/project/1313-openmv/. [Accessed 15 11 2020].

[30] M. S. Ibrahim Abdelkader, “Machine Vision With Python,” [Online]. Available: https://www.kickstarter.com/projects/botthoughts/openmv-cam-embedded-machine-vision/. [Accessed 15 11 2020].

[31] STMicroelectronics, “STM32H743VI High-performance and DSP with DP-FPU, Arm Cortex-M7 MCU with 2MBytes of Flash memory, 1MB RAM, 480 MHz CPU, Art Accelerator, L1 cache, external memory interface, large set of peripherals,” [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32h743vi.html. [Accessed 15 11 2020].

[32] Arm Limited, “Cortex-M7,” [Online]. Available: https://developer.arm.com/ip-products/processors/cortex-m/cortex-m7. [Accessed 15 11 2020].

[33] OPENMV, “OpenMV Cam H7,” [Online]. Available: https://openmv.io/products/openmv-cam-h7. [Accessed 15 11 2020].

[34] OPENMV, “Download,” [Online]. Available: https://openmv.io/pages/download. [Accessed 15 11 2020].

[35] Python Software Foundation, “Python 3.0 Release,” [Online]. Available: https://www.python.org/download/releases/3.0/. [Accessed 15 11 2020].

[36] OpenMV LLC, “MicroPython libraries,” [Online]. Available: https://docs.openmv.io/library/index.html. [Accessed 15 11 2020].

[37] ETSI, “Specification #: 27.007,” [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1515. [Accessed 20 12 2020].

[38] Avnet, Inc, “Avnet Silica NB-IoT Sensor Shield,” [Online]. Available: https://www.avnet.com/wps/portal/silica/products/new-products/npi/2018/avnet-nb-iot-shield-sensor/. [Accessed 16 11 2020].

[39] ComputerNetworkingNotes, “Types of Network Protocols Explained with Functions,” [Online]. Available: https://www.computernetworkingnotes.com/networking-tutorials/types-of-network-protocols-explained-with-functions.html. [Accessed 20 12 2020].

[40] Digilent, Inc, “Digilent Pmod™ Interface Specification,” 20 11 2011. [Online]. Available: http://digilentinc.com/Pmods/Digilent-Pmod_%20Interface_Specification.pdf. [Accessed 23 11 2020].

[41] Digilent, Inc, “Digilent Pmod™ Interface Specification,” 29 09 2020. [Online]. Available: https://reference.digilentinc.com/_media/reference/pmod/pmod-interface-specification-1_3_0.pdf. [Accessed 23 11 2020].

[42] Shenzhen Okystar Technology Co., Ltd, “OKY3261-2,” [Online]. Available: https://okystar.com/wp-content/uploads/2017/08/OKY3261-2.pdf. [Accessed 20 12 2020].

[43] The Python Software Foundation, “binascii — Convert between binary and ASCII,” [Online]. Available: https://docs.python.org/3/library/binascii.html. [Accessed 6 12 2020].

[44] IETF Network Working Group , “ The Base16, Base32, and Base64 Data Encodings,” IETF, 2006. [Online]. Available: https://tools.ietf.org/html/rfc4648. [Accessed 6 12 2020].

[45] G. Şengün, “Base64 Encoding nedir ve nerelerde kullanılır?,” 18 3 2019. [Online]. Available: https://medium.com/@gokhansengun/base64-encoding-nedir-ve-nerelerde-kullanılır-d82f5307ea6d. [Accessed 6 12 2020].

[46] Quectel Wireless Solutions Co., Ltd., “BG96_TCP/IP_AT_Commands_Manual_V1.1,” 31 1 2019. [Online]. Available: https://www.quectel.com/support/downloads/ATCommandsManual.htm. [Accessed 4 10 2020].

[47] Elbest Kaubandus OÜ , “Prügikonteiner 660 l, roheline,” [Online]. Available: https://www.elbest.ee/?id=301. [Accessed 8 12 2020].

[48] LAOEKSPERT OÜ, “Prügikonteiner 660L roheline,” [Online]. Available: https://www.laoekspert.ee/e-pood/prugikonteinerid-prugikonteiner-660l-roheline-t2094. [Accessed 8 12 2020].

[49] OpenMV project, “Ultra Wide Angle Lens,” [Online]. Available: https://openmv.io/collections/lenses/products/ultra-wide-angle-lens. [Accessed 8 12 2020].

[50] M. M. A. Y. L. M. A. K. S. P. C. V. Sikandar Zulqarnain Khan, “An Empirical Modelling for the Baseline Energy Consumption of an NB-IoT Radio Transceiver,” TechRxiv, 2020.

[51] OpenMV LLC, “image — machine vision; MicroPython 1.13 documentation,” [Online]. Available: https://docs.openmv.io/library/omv.image.html. [Accessed 16 12 2020].

[52] J. Haukland, “Modelling the Energy Consumption of NB-IoT Transmissions,” Norwegian University of Science and Technology, 2019.

[53] P. Z. F. P. G. D. Mohieddine El Soussi, “Evaluating the Performance of eMTC and NB-IoT for Smart City Applications,” in *2018 IEEE International Conference on Communications (ICC)*, Kansas City, 2018.

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Haigo Hein

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis NB-IOT PROTOTYPE SYSTEM FOR SMART WASTE MANAGEMENT, supervised by Muhammad Mahtab Alam.
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

02.01.2021

1 The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.