A Guide to Raspberry Pi and Microsoft Azure (IoT)

Handbook for Microsoft Azure IoT

Pontus Bergström

Degree Thesis for a Bachelor of Engineering

Degree Programme in Electrical engineering and automation

Vaasa 2023

BACHELOR'S THESIS

Author:	Pontus Bergström
Degree Programme:	Electrical Engineering and Automation
Specialization:	Automation
Supervisor(s):	Jan Berglund, Novia University of Applied Sciences
	Hans Lindén, Novia University of Applied Sciences

Title: A Guide to Raspberry Pi and Microsoft Azure (IoT)

Date: 26.5.2023 Number of pages: 38

Abstract

This bachelor's degree thesis was conducted on behalf of Novia University of Applied Sciences, with the purpose of getting familiar with IoT (Internet of things) which is becoming more and more a key future in today's society and can be found in almost all processes in today's day-to-day activity in some way, more specifically to get an understanding of cloud computing and transmitting data from a device (Raspberry Pi) to the cloud service application Microsoft Azure. The project was inspired by the idea of transmitting temperature data from a summer cottage to a smartphone or computer at home.

The thesis structure gets the reader a basic knowledge and theory needed for the project. The theory section will describe Cloud Computing, Raspberry Pi, Temperature sensor DS18B20, Programming language, Python, Microsoft Azure, and more.

The methodology of gathering needed information for this thesis has been broken down into self-learning, own experiences, and high interest in IoT.

From a theoretical point of view, this was a success, the project is ready to be implemented provided that the user has power and a stable Internet connection. Furthermore, an attached handbook is presented as a beginner's guide to building on this project and adding functions.

Language: English
Key Words: Raspberry Pi, IoT, Microsoft Azure, Automation

EXAMENSARBETE

Författare:	Pontus Bergström
Utbildning och ort:	El- och Automationsingenjör, Vasa
Inriktning:	Automationsteknik
Handledare:	Berglund Jan, Yrkeshögskolan Novia
	Hans Lindén, Yrkeshögskolan Novia

Title: Automation med Raspberry Pi och Microsoft Azure

Datum: 26.5.2023 Sidantal: 38

Abstrakt

Detta examensarbete har genomförts på uppdrag av Yrkeshögskolan Novia, med syftet att bekanta sig med IoT (Internet of things) som blir mer och mer en stor stödpelare i dagens samhälle och finns i nästan alla processer i nutidens dagliga aktiviter på ett eller annat sätt, mer specifikt för att få en förståelse för moln tjänster och överföring av data från en enhet (Raspberry Pi) till molntjänstapplikationen Microsoft Azure. Projektet var inspirerat av idén att kunna överföra temperaturdata från en sommarstuga till din smartphone eller dator hemma.

Strukturen för arbetet ger Läsaren de grundläggande kunskaper och teorier som behövs för projektet. Cloud Computing, Raspberry Pi, Temperatursensor DS18B20, Programmeringsspråk, Python, Microsoft Azure och mer kommer att beskrivas i teoriavsnittet.

Metodiken för att samla in nödvändig information för detta examensarbete har brutits ner i självlärande och egna erfarenheter, och stort intresse för ämnet IoT. Från en teoretisk synvinkel var detta en succé, projektet är redo att genomföras förutsatt att du har ström och en stadig internetuppkoppling. Dessutom presenteras en bifogad handbok som en guide för att bygga vidare på detta projekt och lägga till funktioner.

Språk: Engelska

Nyckelord: Raspberry Pi, DS18B20, Azure, Automation

Table of Contents

List of Figures

1 Introduction

With everything in today's society being increasingly automated, almost everything has to do with automation in some way. "Internet of Things" is becoming popular for simplifying daily activities and tasks. IoT (Internet of Things) is a collective network of connected devices communicating with one another in the cloud. IoT is used in many day-to-day activities such as industry, businesses, and offices but is now also very popular in what is today called a "Smart Home", which simplifies tasks around the house such as thermostats, vacuum cleaners, washers, lights, and much more, which is all connected to a cloud that is accessible from the smartphone, for example. (What is IoT?, 2022)

This project covers IoT on a simplified level to get a better understanding of all its aspects and surrounding technology. This assignment aims to set up a Raspberry Pi with sensors to easily get temperature readings remotely together with a cloud service application. From a practical point of view, this project's goal is to be able to for example set up a Raspberry Pi at the summer cottage and remotely stream temperature reading to a computer at home with many different possibilities, which will be explained in another section.

On behalf of Novia of Applied Sciences, the main goal of this project is however not to just get a better understanding of the Raspberry Pi but also to get a good understanding of the cloud service called Microsoft Azure and all the surrounding aspects that come with it, this thesis can easily use as a startup guide with limitless possibilities to work beyond this project itself. A separate handbook will be available as an attachment to this project as a hands-on guide.

The structure of this thesis is made so that the reader first gets familiar with the hardware and other relevant details to easily follow along with the hands-on guide and also get a better understanding of the components and aspects of the thesis. Firstly, a description of a Raspberry Pi and its different possibilities. Sensor DS18B20 will also be briefly explained. In the hands-on guide, I will use a step-by-step method to eliminate errors from the user.

1.1 Background

1.1.1 Automation and IT Technology

Automation and IT (information technology) are terms that grow by the day in today's society and can be defined as technology applications where human participation is minimized to take off human workload and simplify processes and make them more efficient. Examples of this would be (BPA) business process automation/IT automation, and home automation which simplifies the human's day-to-day tasks around the house and makes for a more comfortable living situation.

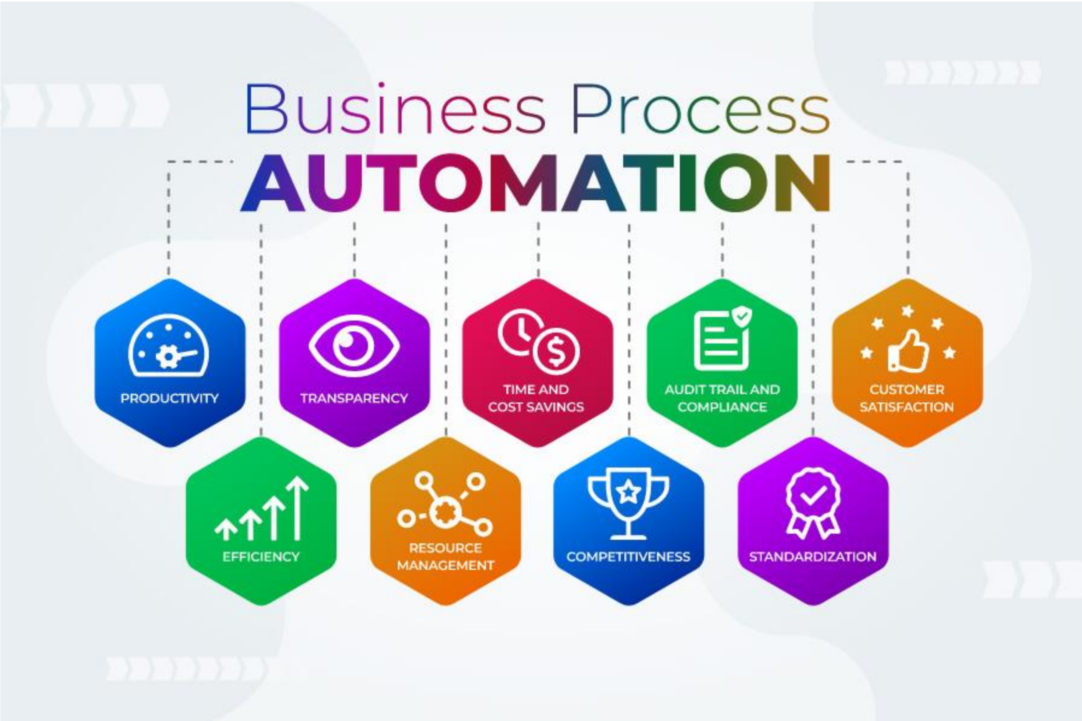

Figure 1. Business Process. (Why should you automate your business processes?, 2022)

Defining **Business Process Automation**, can be looked at it from different perspectives, from an administrative level or used in physical processes to simplify multistep business operations and minimize human errors. For example, on an administrative level, handling and storing of documentation, storage of data, and administrative task that would take excessive amounts of time and that with even a small human error can cause damage to the company's security, efficiency, economy, and safety. When talking on a more physical level, take a product line for example, instead of having multiple humans stationed at different stages of a process where precision and repeatable action is required, and processes where even the smallest of error can cause damage to the end product, a robot can be built and automated to do the same thing as a human would, but more efficiently and with better precision.

Automation software and hardware's main goal is to try to prevent and secure these issues. By developing a sustainable and sufficient business automation strategy, a company can take control of its processes and provide a significant cost-saving, error-free, and efficient business.

IT automation can with the help of software automate and set up repeatable instructions, processes, and policies to save time and make IT departments able to concentrate on other more strategic work.

Another way of implementing these processes is what in today's age is called a "Smart Home". Taking these functions, implementing them into our homes, and making living more enjoyable and efficient. The possibility to manage and control day-to-day tasks straight from the smartphone or by installing voice-activated hubs such as Apple "Siri", Android "Alexa" and google assistant. These technologies, enable remote control and adjustment for example, via connected heating systems change the temperature in a building, start the vacuum cleaner or pre-heat the oven, manage a full-scale security system, and so much more. The imagination is the limit here. (Why should you automate your business processes?, 2022)

1.1.2 IoT

The Internet of Things, or IoT, is influencing our lifestyle today, from the way one can control home air condition from the smartphone to how a cars navigation system finds the fastest route to a desired destination, and a smartphone or smartwatch is tracking a person's daily activities during the day. So, IoT is a variety of devices that are connected to the network, they gather and share data about how they are used and in which environment they are operated. This is all made possible thanks to sensors, almost all physical electric devices are embedded with sensors in some kind of way, and these sensors send data to a network about the device status and implementation.

IoT works as a place where all these devices and sensors can dump all their data, and work with a common language to communicate with each other. IoT platforms combine the collected data from various sources where further analytics can be performed to extract the wanted data for our requirement. Finally, the data result is shared with other devices for better user experience automation and improving efficiencies.

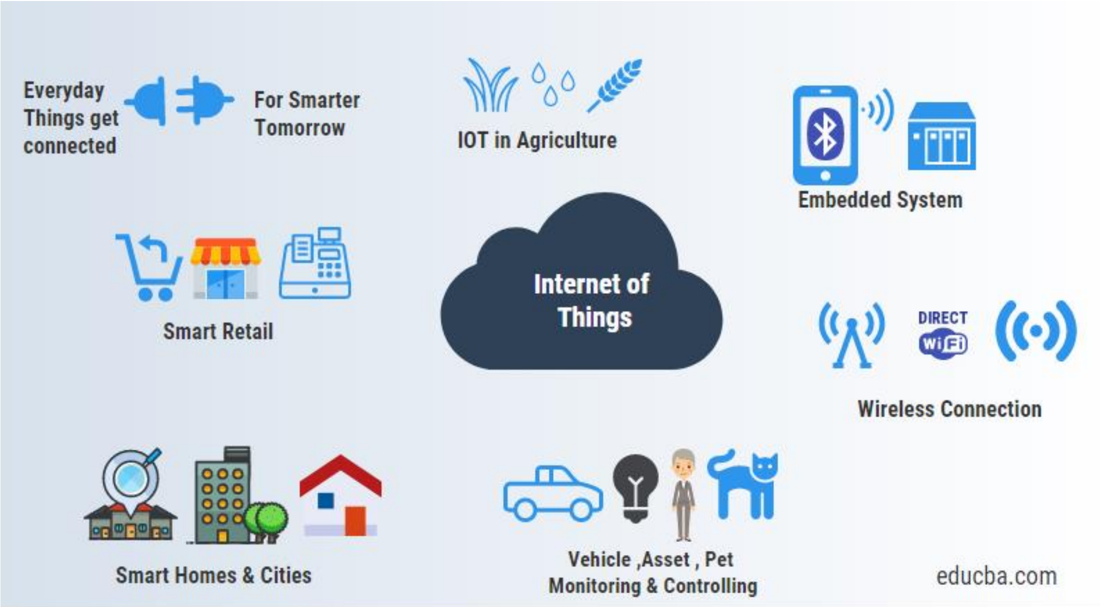

Figure 2. Internet of Things. (What is IoT?, 2022)

A short example of this is an air conditioner compressor with an embedded sensor built in that stream data on the working condition of the device and temperature. This data is then stored and analyzed to be able to provide information if the device is broken and needs to be repaired or if the temperature is dramatically changed.

Why the Internet of Things? IoT allows businesses to automate human interaction of processes and enables the reduction of labor expenses. IoT simply makes the whole business run more smoothly and efficiently, making everything cheaper and faster to deliver goods to customers as well as offering simplicity in customer dealings. (What is IoT?, 2022)

1.2 Purpose

The purpose of this thesis is divided into two parts, firstly to connect and configure a Raspberry Pi with the 1-wire temperature sensor (DS18B20) to extract data and remotely get data pushed to the cloud service Microsoft Azure, and with the application Power Bi get a live feed of data(temperature) shown as a diagram, this will be explained in depth in the theory section.

Secondly, to provide a handbook that includes how to set up Raspberry Pi, and with a programming language, Python configures Microsoft Azure to receive data from the Raspberry Pi.

1.3 Scope

The scope of this thesis will elaborate on the following questions:

- How does one easily set up a small-scale IoT network?
- What is IoT and how does it work?
- How can someone use this thesis as a basic guide to future research?

The scope of this thesis along with the side-by-side handbook is a basic guide on how to set up, manage and maintain an IoT system with Microsoft Azure with key features such as an on-screen live feed. This is the limitation of this thesis and functions as a basic project that has countless possibilities to build upon.

1.4 Delimitation

On behalf of Novia University of Applied Sciences, this thesis will focus on getting familiar with the systems Microsoft Azure and the Basics of Raspberry Pi. However, brief comparisons and alternatives for these systems and why these were chosen. This project will only use one temp sensor to provide needed information.

In parallel with this thesis, there will also be a step-by-step handbook, as this thesis is only on a basic level and there are countless possibilities for expanding on the project. The thesis limits are creating a database in Azure, presenting data with Power Bi, and a basic understanding of the Raspberry Pi. This thesis will not go into depth on other alternatives.

The opportunities to use Microsoft Azure are endless, receiving data remotely from the Raspberry Pi and temperature sensor DS18B20, and Presenting live feed data in Power Bi.

1.5 Disposition

The first chapter of this thesis is an introduction to the project, what the limits are, on whose behalf this thesis is made, and the purposes of the project, and a short look at the background of automation technology and IoT (Internet of Things).

The next chapter is the theory, which will dive into the relevant information to better understand the project. This chapter will also describe all stages of the process and component overview. This chapter also includes comparisons of the chosen components and program and why.

Installation, which is the third chapter, will include the process chosen for this thesis and will have parts from the handbook included to get an understanding of how everything is set up and configurated. This chapter will also display the programming of the Raspberry Pi and the code used.

The fourth chapter is the **result**. This will serve as a result of what was accomplished with the thesis, what worked and what didn’t. The chapter will also process the handbook that goes hand-in-hand with this thesis.

The fifth and final chapter that processes this thesis is the **discussion** of the result, this will provide a conclusion of the thesis and future possibilities to work on and use this project as a base for potential applications.

2 Theory

The following chapter will present the theoretical aspects of this thesis, and facts gathered through studying the relevant information needed to carry out the project and reach a result. All relevant information, datasheets, basic knowledge of things such as programming, cloud computing, Power Bi, and hardware descriptions, this chapter will also determine why certain things have been used compared to other alternative methods. Other methods will also be present but only briefly inform of comparisons, advantages, and disadvantages. Chapter “Installation” will go down deeper into the more practical execution of the project.

2.1 Cloud Computing

To get a better understanding of cloud computing, (Figure 3), the alternatives, On-premises computing systems, and cloud-based computing systems are compared

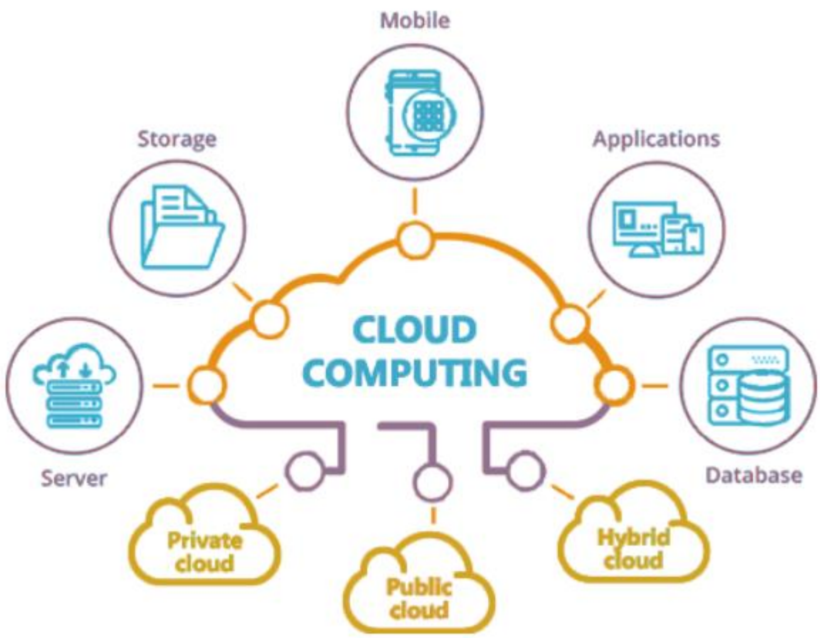

Figure 3.Cloud Computing. (A primer on Cloud Computing, 2019)

2.1.1 Scalability

On-premises scalability, in upscaling, investment with a lot of money and few possibilities, further making downscaling difficult often with extensive economical losses, when on the other hand cloud computing systems go by the method, of "pay as needed" with an easy option for upscaling and downscaling. (Cloud computing in 6 minutes, 2023)

2.1.2 Server Storage

In server storage, there might be a big advantage in cloud computing, as one, pay for what is used, and cloud computing providers manage and maintain server storage. On-premises, server storage often needs a lot of space for server rooms, power, and constant maintenance. (Cloud computing in 6 minutes, 2023)

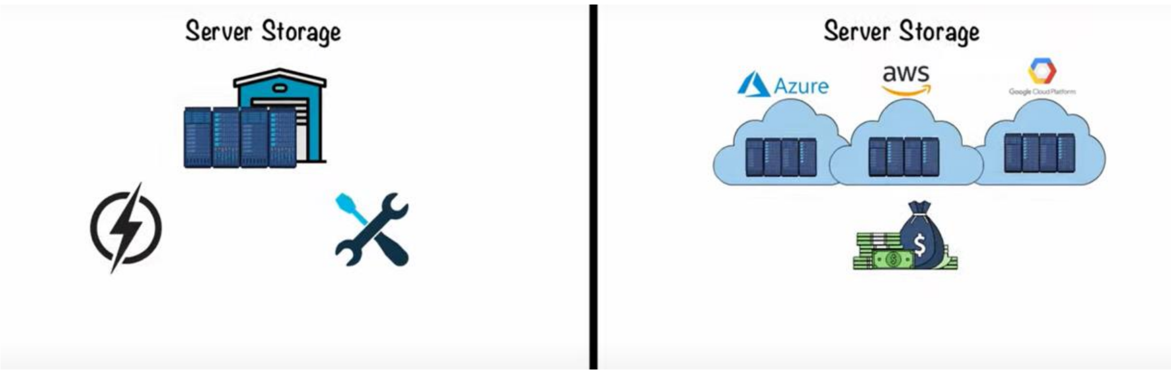

Figure 4.Server Storage. (YouTube cloud computing in 6 minutes)

2.1.3 Data Loss

If data is lost, on-premises systems can be hard to recover due to difficulties of backup systems maintenance and this comes with questions such as storage space and costs. If a cloud storage provider is used, they already have robust systems and procedures in case of disaster, making for much faster and safer data recovery. (Cloud computing in 6 minutes, 2023)

2.1.4 Maintenance

To have teams of people hired to attend to the maintenance of systems and server storage comes with many costs and is time-consuming. On the other hand, using a cloud computing system this is automatically maintained by the provider, saving the company both a lot of money and time. (Cloud computing in 6 minutes, 2023)

2.1.5 Service Model

For setting up a cloud computing system, companies usually follow one of these three service models depending on the demand. These different methods differ from the On-Premises as then the company needs to maintain everything themselves.

When comparing these service models to on-premises models, everything is managed by themselves, like application, data, server, storage, and network. (Cloud computing in 6 minutes, 2023)

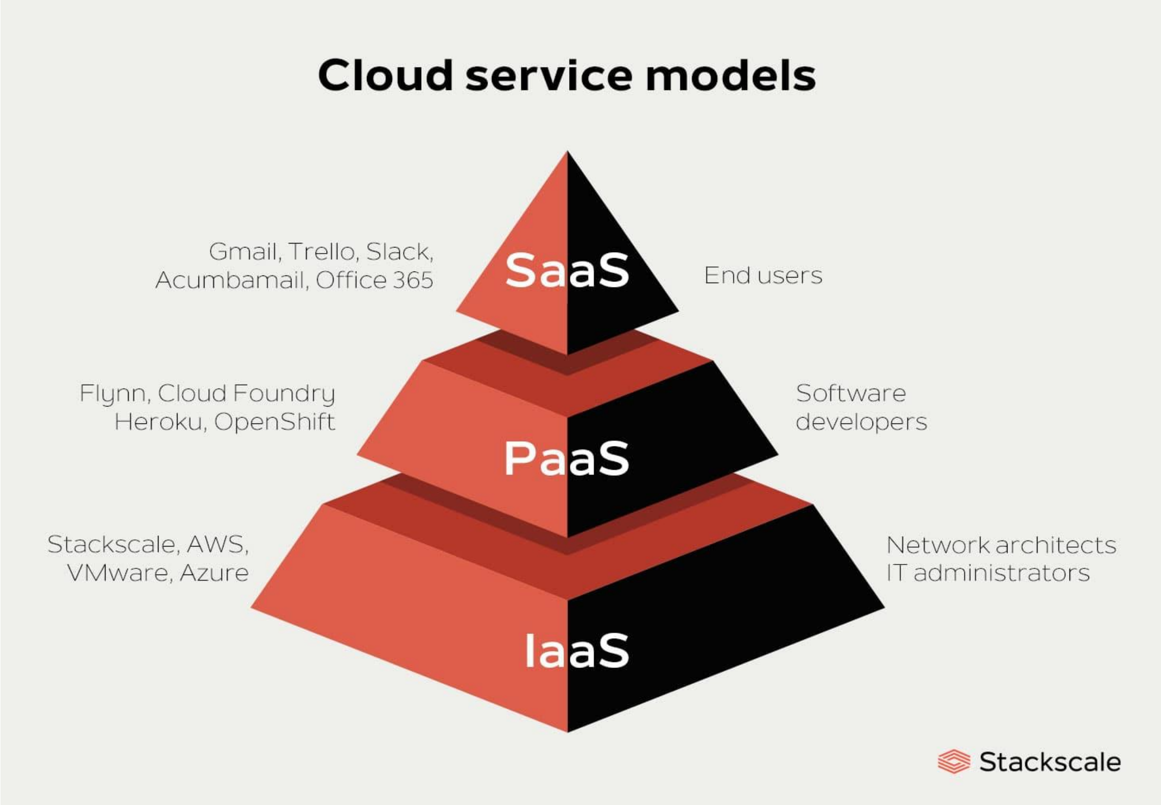

Figure 5.Cloud service models. (Main cloud service models: IaaS, PaaS and SaaS, 2022)

- IaaS (infrastructure as a service) allows users to basic computing infrastructure, usually handled by IT executives, this model is suitable for companies with On-Premises server storage and uses of virtual machines. The only thing the user will handle on their own here is the application, runtime, data, middleware, and operating system. Some examples are AWS, Microsoft Azure, and GCE (Google Compute Engine) (Main cloud service models: IaaS, PaaS and SaaS, 2022)

- PaaS (Platform as a Service) if the organization needs a platform for only creating applications and does not what the hassle of managing and maintaining the architecture. Often used by software developers. The things the user has to handle here are application and data. Windows Azure, AWS Elastic Beanstalk, and Google App Engine are examples of these. (Main cloud service models: IaaS, PaaS and SaaS, 2022)

- SaaS (Software as a Service) is probably something everyone uses daily. Used as cloud services for hosting and handling software applications. Software and hardware requirements are met by the suppliers. Examples of this are google workspace, dropbox, salesforce, Power BI, and WebEx. (Main cloud service models: IaaS, PaaS and SaaS, 2022)

2.2 Raspberry Pi

Raspberry Pi is a series of small single-board computers introduced in February 2012. It was developed in the United Kingdom by the Raspberry Pi Foundation and was originally aimed to provide an easy way to teach basic computing and programming in schools for an affordable price as it only needs an SD card to install an operating system. (Wikipedia, 2022)

Raspberry Pi gets power from a simple micro-USB cable or newer models with a USB-C cable. As this computer only comes in sizes between a deck of cards and as small as a stick of gum it is the perfect option for smaller projects and little space. The operating system Raspberry Pi uses is called Raspbian and is an optimized variant of Linux Debian, but other versions of Linux are also able to run on the Raspberry Pi. (Wikipedia, 2022)

Since February 2012 when the first RPI (Raspberry PI) was released, there have been as many as 15 versions released. In 2020 the newest version was released in from of the Raspberry Pi 400, change up the form of the device it now comes with a built-in keyboard in one compact device, along with the RPI compute module 4 that had the practical handheld form. (The Different Versions of the Raspberry Pi, 2022)

Figure 6.Raspberry 400(2020). (The Different Versions of the Raspberry Pi, 2022)

2.2.1 Raspberry Pi Hardware

Along with the newer Raspberry Pi's before the 1 Model B+ comes the powerful 40-pin GPIO (general-purpose input/output) header compared to the older versions only provides 26-pin headers. Without going into depth detail, GPIO performs digital input/output functions and by design does not have a predefined purpose. (Wikipedia, 2022)

2.2.2 Raspberry Pi 4B

The Raspberry Pi used in this project is the model 4B which offers an increase in processor speed compared to the previous Pi 3 Model B+, other upgrades to the previous versions are things such as software performance, memory, and connectivity while still providing backward compatibility with the older version.

With a high-performance Broadcom BCM2711, quad-core Cortex-A72, 64-bit at 1.5GHz processor compared to the Raspberry 3 model B+'s Broadcom BCM2837B0, Cortex-A53 64-bit at 1.4GHz the new model 4B also provide dual-display support with 4K resolution maximum connected via two micro-HDMI ports, maximum of 8GB of RAM, Gigabit Ethernet, wireless LAN 2.4/5.0 GHz, and Bluetooth 5.0. For power supply; 5V/2.5A DC power input.

(Raspberry Pi 4 Tech Specs, 2023)

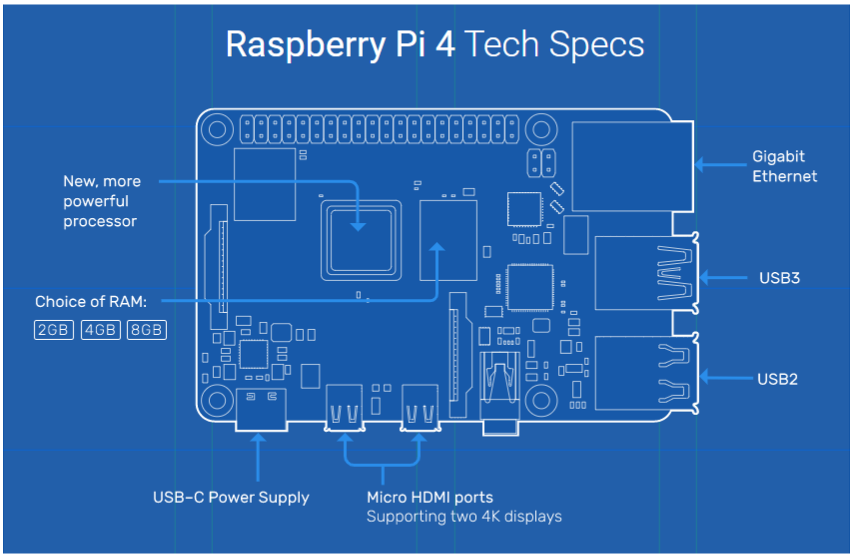

Figure 7.Raspberry Pi 4. (Raspberry Pi 4 Tech Specs, 2023)

2.3 Temperature sensor DS18B20

The DS18B20 digital thermometer offers 9-bit to 12-bit Celsius temperature measurements. Because DS18B20 uses a 1-Wire interface for communication, a bus that by definition requires only one data line (and ground) for communication with a central microprocessor. DS18B20 uses a unique 64-bit serial code that makes it possible to set up multiple identical sensors to operate on the same 1-Wire bus. This makes this sensor a good choice when operating the sensor with a single-board microprocessor even with multiple sensors in use. This sensor has a wide variety of applications, from industrial systems to thermally sensitive systems and consumer products. Range of temperature reading between -55 and +125 degrees Celsius with an accuracy of 0.5 from -10 to +85 degrees Celsius. (Programmable Resolution 1-Wire Digital Thermometer, 2022)

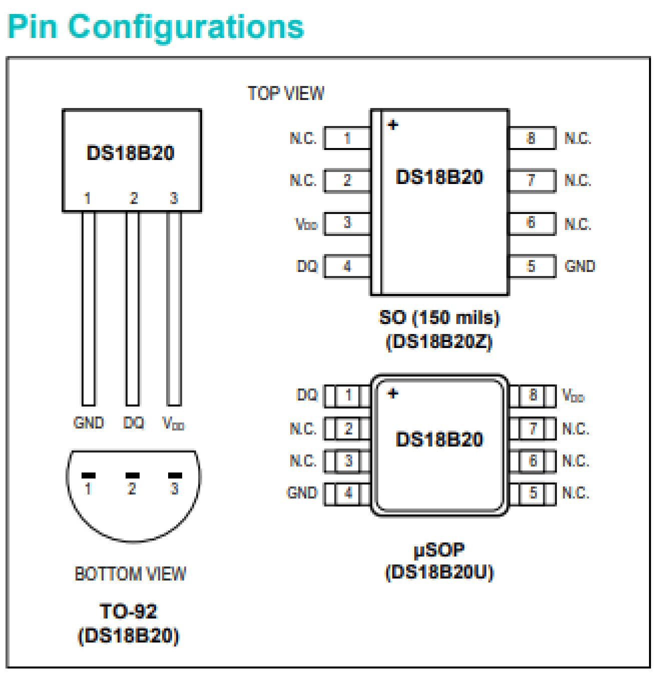

Figure 8. Pin Configuration. (Programmable Resolution 1-Wire Digital Thermometer, 2022)

2.3.1 The 1-Wire Family

Developed by the company Dallas Semiconductor today known as Maxim Integrated since 2002, 1-Wire is a communication protocol that allows different devices to communicate with one another by only using one data/power wire and ground connection. This technology is most often used when connecting low-speed peripheral devices to microcontrollers and integrated circuits. By using a serial communication method that transmits data in a so-called division multiplexing manner. All devices connected to the 1-Wire bus can be master or slave devices, which means the master device takes the active role by initiating the communication, while the slave devices respond and provide the requested data or perform the requested actions. (1-Wire, 2023)

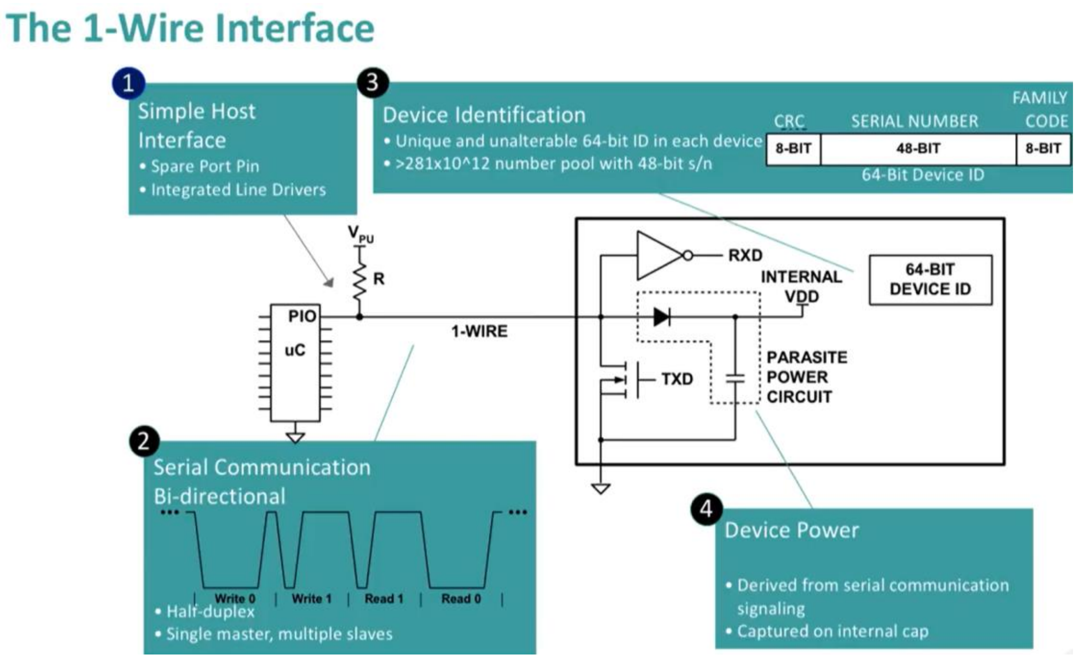

Figure 9. The 1-Wire Interface. (Overview of 1-Wire Technology and Its Use, 2023)

2.3.2 Features

One-Wires are mostly used as temperature sensors, humidity sensors, real-time clocks, and more, most often designed to be small, low-power, and able to withstand harsh environments. (Overview of 1-Wire Technology and Its Use, 2023)

1. Single-wire connection: Only one wire (and ground wire) is required for communication and power. This makes it suitable for applications where simplicity and low pin count are important.

2. Power-over-data line: The One-Wire bus can provide power to connected devices. Power is transmitted over the same wire used for communication.

3. Unique device addressing: Each device on the bus has a unique 64-bit address, allowing multiple devices to be connected without conflicts. This addressing scheme enables easy identification and selection of specific devices on the bus.

4. Data integrity: The protocol includes mechanisms for error detection and correction, ensuring reliable data transmission.

5. Low-cost implementation: One-Wire devices are relatively simple and inexpensive to manufacture, making them widely available for various applications.

Overall, the One-Wire protocol provides a convenient and efficient way to connect and communicate with peripheral devices using a single wire, making it popular in many applications where simplicity, low pin count, and cost-effectiveness are important considerations. (Overview of 1-Wire Technology and Its Use, 2023)

2.4 Programming

Programming, also known as coding, is code executed in a computer program to perform the desired task, depending on what type of code is written. One can define programming as a set of instructions given to the computer that tells how a program should behave.

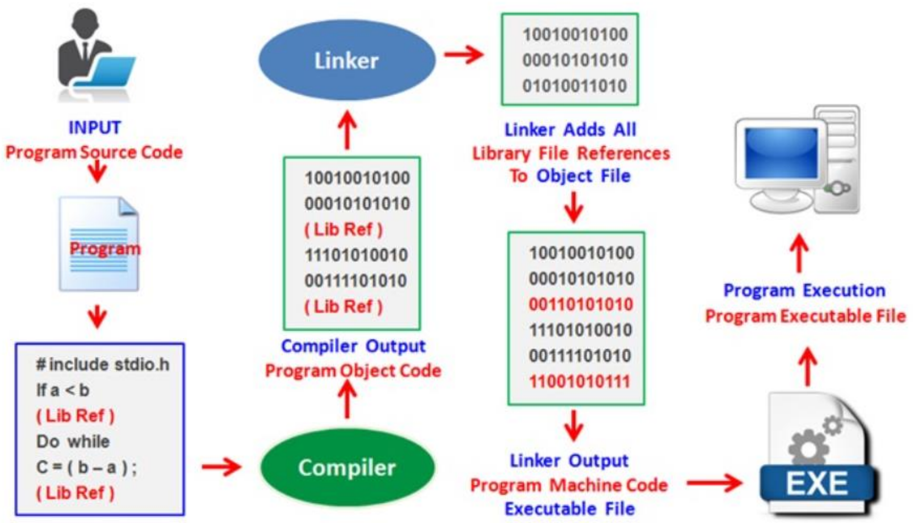

Figure 10. Programming Process. (What is computer program?, 2022)

To write this kind of code, a computer program is used, or in other words a software application, these programs use a so-called "human readable" high-level programming language that uses a human-friendly interface, that is then later compiled into a low-level machine code that is written in binary so that the computer can execute the program performed user-specified operations. A program like this is used for example data processing, different calculations, data storage, or whatever the user has specified the program to do. (What is computer program?, 2022)

2.4.1 Python

One of today's fastest-growing programming languages is Python, which is a very easy and accessible programming language for both beginner and advanced programmers. Python has a huge demand among Software Engineers, Mathematicians, data analysts, accountants, and many more, and also provides kids with fun and instructive projects to start learning programming at a young age. Python is easily applicable to people who are not software developers. Python is a perfect programming language when it comes to automating, if working with writing Excel spreadsheets, pdfs, and so on, python is probably capable to automate it. (Python Tutorial for Beginners, 2023)

<u>Why Python?</u>

- Able to solve complex problems with minimal time and lines of code.
- Has a huge community ready to help if the user is having problems with the code.
- Work as well on a beginner level as on an advanced level of programming.
- Cross-platform applicable.
- Huge library of functions, whatever is needed, there is a big chance Python can do it.
- Can do what other programming languages can but simplified.

2.4.2 Programming Categories

Countless different programming languages exceed in different areas, some are very general and can be used for almost any situation, while others are better for different purposes. These different categories of programming languages are Procedural programming languages, functional programming languages, object-oriented programming languages, scripting languages, and logic programming languages.

Procedural programming languages: this type of language creates a sequence of statements and commands to reach its output/goal. A program written in this type of language will have a series of procedures to be executed inside other procedures. Languages that use this type of language commonly are for example Python, C/C++, Java, and Pascal, which were designed to teach programming languages and are today mostly replaced by C/C++ and Java.

Functional programming languages: Instead of executing different kinds of statements and procedures within each other, functional programming aims more towards mathematical evaluations and focuses on performing a specific task and returning for a result where each function is reusable, and the result varies on the user input to the function. Scala, Haskell, Elixir, and F# are common examples of this.

Object-oriented programming: Pretty self-explanatory this type of programming uses the code inside as "groups of objects" that are composed of data and different program elements, these are referred to as attributes and methods, the purpose of this is that the objects can be reused in the program itself or saved to be used in different programs with makes it easy to keep track in more complex programs because it is easy to scale and reuse. Object-oriented programming is used in but not limited to Java, Python, C++, and Ruby.

Scripting languages: simply put, without any need of compiling, a series of commands that are being executed which is used by an interpreter that translates it directly from the source code. This language is used to automate and simplify repetitive tasks, web content for example weather sites uses this to show a dynamic weather report on the screen. Programming languages that can perform this are programs such as PHP, Ruby, Python, etc.

Logic programming languages: compared to the other type of languages, with the logic programming language, the user doesn't have to "tell" the computer what to do but use a series of statements and rules to initiate how the computer makes decisions in the program. Prolog, Datalog, and Absys are typical logic programming languages.

(Coursera, 2023)

2.5 Microsoft Azure

Microsoft Azure is one the biggest cloud computing services on the market and provides all kinds of different services and possibilities, it was launched on February 1st, 2010. Azure is an online portal that the user can access as long as there is an Internet connection, with access to all of their services. Azure has a pay-per-use type of model that is free to start, and where the user can test it out for free with all functions before deciding to buy. Another thing that makes Azure so powerful is that around 80% of the world's top 500 companies use Azure in one way or another and provide support for most programming languages out there such as Python, C#, JavaScript, and so on. With 18 categories and over 200 different services, Azure can provide the user with almost whatever is needed, and if it does not exist, one can be almost certain that it is a work in progress.

As shown in (Figure 10), the services used in this project are networking, IoT, and analytics. More in-depth will be explained in the Installation chapter.

(Simplilearn, 2023)

Figure 11. Microsoft Azure Applications. (Simplilearn, 2023)

2.5.1 Azure Vs AWS Vs GCP

Microsoft Azure, Amazon Web Services, and Google Cloud Platforms are three of the biggest cloud computing platforms out there today, the following will be a brief comparison of the platforms, the project itself is only using Azure as this was requested for the handbook of this thesis.

As of today, Amazon Web Services has about 32% of the total worldwide public cloud share, Microsoft Azure owns about 16% of the total public market share, and following Google Cloud Platform with around 9%. These three major cloud computing services are similar to some degree but vary somewhat:

2.5.2 AWS

can be used when the user wants to create and deploy applications within the cloud and provides services over the Internet. AWS is the oldest of the three and was launched in 2006.

Database Service: AWS relational database service is cost-effective and automates processing tasks with the design to simplify the operation, setup, and scaling of a database. AWS uses Dynamo DB as a NoSQL database service for low latency and scalability.

Advantages/disadvantages: Provides beginner-friendly services with instant access to all recourses within the service for the specific need on a trial period, surpasses speed and agility, and is secure and reliable.

These services are not perfect, and one drawback of AWS is that they don't have free immediate technical support, the user may choose from package prices ranging between 29$ and up to 15,000$ depending on the size of the business. Aside from that AWS has some network connectivity issues, and general issues when uploading to the cloud for example downtime, limited control, and backup protection. These are temporary issues with solutions coming. (AWS Vs Azure Vs GCP, 2023)

2.5.3 Azure

As already stated in the introduction Azure was launched on February 1st, 2010, and provides a large range of services for the user to build, manage and deploy different applications on the network with the help of different tools and frameworks, software frameworks refer to pre-built support programs, compilers, and built-in code libraries and toolsets.

Database Service: Azure uses SQL Server a database is a software that is used as a service platform within Azure. It has pre-built-in intelligence that learns data patterns and adapts them to maximize performance, reliability, and data protection. This makes it easy to transfer SQL server databases without changing the user's application. Azure Cosmos DB natively supports NoSQL crates for low latency and scalable applications.

Advantages/disadvantages: Microsoft Azure will provide the user with superior development operations with a strong and secure environment as Azure follows the security model of detecting, reviewing, analyzing, and stabilizing. Azure is also a very cost-effective alternative as are all of these three services as they all use the pay-as-you-use model. Azures Cloud allows the user to launch both internal and customer applications in the cloud which saves time and cost on IT infrastructure, and the Internet provides a lot of beginner guides.

Looking at some of the flaws in Azure, it uses a different codebase depending on whether working online in the cloud or offline on-premises level, A codebase is the complete body of the source code for a given program or application and these differ from each other in the cloud and offline. Another disadvantage of Azure is the usage of PaaS instead of IaaS which is more efficient. The interface and management tools are not user-friendly, and an unfamiliar user is forced to do a lot of searches and clicks to perform simple tasks. (AWS Vs Azure Vs GCP, 2023)

2.5.4 GCP

The smallest of the three, Google Cloud Platform was launched on April 7th, 2008, and offers user application development and integration services for the end-user, GCP also provides cloud management as well as machine learning, IoT, and Big Data. Big Data refers to data sets that are too big to be handled by traditional processing software.

Database Service: GCP uses could SQL and is a fully managed database service that is easy to use and maintains relational PostgreSQL, MySQL, and SQL Server databases in the cloud hosted by GCP cloud SQL. Cloud SQL also gives a database infrastructure for applications running everywhere. GCP Cloud Datastore provides high performance and auto-scaling of application development.

Advantages/disadvantages: Google Cloud platform outperforms the pricing department as they use the bills-in-minute model, which means the user only pays for the computing time. GCP also provides good discounts for long-running workloads and applications, if required, a heavy user of GCP will be rewarded for it with discounts. GCP also provides a live migration between VMs (Virtual Machines) and that means that moving a running VM from one server to another out disruption or availability for the end user and this future GCP excels for other cloud service providers.

Google Cloud Platform same as the other two, is not perfect. GCP uses an expensive support service with fees of around 150$ monthly for the most basic service package, and GCP also uses quite difficult to understand pricing schematic which makes it expensive to for example download data from the cloud storage. Most of the GCP flaws go to pricing as it can get expensive very fast.

All these three services have a pay-as-you-use approach. This means the user pays for the services they need, for as long as needed without committing to some long-term agreement or complicated contract arrangements. (AWS Vs Azure Vs GCP, 2023)

Instance type (virtual servers)	AWS	Pricing	Azure	Pricing	GCP	Pricing
General purpose	m5.xlarge	$0.123	B4MS	$0.097	n1-standard-4	$0.128
Compute optimized	c5.xlarge	$0.107	F4s v2	$0.099	n1-highcpu-4	$0.095
Memory optimized	r5.xlarge	$0.159	E4 v3	$0.156	n1-highmem-4	$0.159
GPU instances	g3s.4xlarge	$0.551	NC 6	$0.572	NVIDIA@Tesla @P4	$0.864

Figure 12. Cloud Services Prices. (Simplilearn, 2023)

2.6 Power BI

In this project, Power BI is used for a live presentation of the temperature with a live graph feed, integrated with Azure to extract data from the cloud. Power BI is a software service that can process user data and turn it into visualization in the form of live feeds and diagrams or graphs. For example, excel spreadsheets or data the user want to visualize in the cloud to extract important data, Power BI allows customizing this and extracting important data to share with anyone. Power BI comes in three different versions, *Power BI Desktop*, Online software as a service (SaaS) called *Power BI Service,* and a mobile version for Windows, IOS, and Android devices called *Power BI Mobile.* More details of the service will be presented in the installation chapter. (Microsoft Power BI, 2023)

3 Installation

This chapter handles the installation and the procedure of my thesis. The chapter will go hand in hand with the handbook as an attachment that goes more in-depth about how to project is set up. By studying online different setups and configurations the following installation has been chosen. No other alternative setups will be mentioned with the same set of components. Hardware components used in this project are a Raspberry Pi 4 model B, a breakout board with GPIO ribbon cable, male-to-male jumper cables, one 4.7k resistor, and one DS18B20 temperature sensor. Before setting up the configuration, the user needs to set up the Raspberry Pi configuration, next steps will show how to set it up.

3.1 Configuration of Raspberry Pi

There are several steps to getting the Raspberry Pi set up and ready for use. Configuration involves setting up the operating system, and configuring the network settings, updating the system to the latest software, and additionally, if some more features are required, such as mirroring or access to the system remotely SSH or VNC for example. If a MicroSD card larger than 32GB is used, formatting it to FAT 32 might be needed.

Hardware required:

- MicroSD card with at least 16GB of space is recommended If a MicroSD card larger than 32GB is used, formatting it to FAT 32 might be needed.
- Power supply that comes with the Raspberry Pi.
- HDMI Cable to be able to connect to a monitor.

- USB keyboard and mouse (when the Raspberry Pi is set up, the user can also use Bluetooth connections).

- There are also other components that can be used such as cases, additional fans, or other accessories.

Preparation of the microSD card:

- Plug the microSD card into a computer to then download and install Raspberry Pi software called Imager from the official Raspberry Pi website, previously this was called NOOBS, but has now been upgraded to Raspberry Pi Imager.

- Start the application and select an operating system (the most commonly used is Raspberry Pi OS).

- Choose the storage location (the MicroSD card).

- Lastly, press “Write” to Image/install the operating system to the selected storage location.

Before startup:

- When the operating system has been written to the microSD card, insert it into the Raspberry Pi.

- Connect the HDMI cable to a monitor.

- Lastly, connect the power supply cable to the Raspberry Pi (this will automatically boot it up).

Configure the operating system:

- On the first startup, follow the instruction on the screen such as preferred language, time, and date, and set up a secure password.

- After this the Raspberry Pi, might take some time to set up, when completed it will go into the Raspberry Pi desktop home screen.

Configuration of the network:

- With a wired Ethernet connection, it should be automatically connected.
- If not, the first thing that is needed is to choose a network from the Wi-Fi list, enter the password, and connect.

Get the latest software update:

- To avoid further configuration resetting, it is recommended to update to the latest software.
- Open a new terminal window and run the following commands:
  - Sudo apt update
  - Sudo apt upgrade

Alternative configurations:

- Depending on requirements, here are some useful configurations:
  - Enable SSH: in a new terminal window run the command **sudo raspi-config**, this will access the Raspberry Pi configuration tool, select “interfacing options” and enable the SSH function.
  - Enable VNC: First, download the RealVNC viewer software and follow the instruction. To connect the RPI just type in **vnc://*Ip address***, to check the IP address, if needed, use the terminal with the command **dig *hostname*** from this command it’s possible to find the IP address. After that just follow the instructions prompted on the screen. The be noted it might be required to change some security settings to get everything up and running.

3.2 The Code

The Python script reads temperature data from sensor DS18B20 connected to the Raspberry Pi and then sends the data to Azure Iot Hub. The code will be explained in detail below.

```
from base64 import b64encode, b64decode
from hashlib import sha256
```

```
from urllib.parse import quote_plus, urlencode
from hmac import HMAC
import requests
import json
import os
import time
# Temperature Sensor
BASE_DIR = '/sys/bus/w1/devices/'
SENSOR_DEVICE_ID = '28-00000a54bc7e'
DEVICE_FILE = BASE_DIR + SENSOR_DEVICE_ID + '/w1_slave'
# Azure IoT Hub
URI = 'Pontus-iot-hub.azure-devices.net'
KEY = 'elmb3mp2abphZiSV1zHsUdKISqg6ReSBiowyH7LWTQE='
IOT_DEVICE_ID = 'raspberrypi'
POLICY = 'iothubowner'
def generate_sas_token():
    expiry=3600
    ttl = time.time() + expiry
    sign_key = "%s\n%d" % ((quote_plus(URI)), int(ttl))
    sign_key_bytes = sign_key.encode("utf-8")
    signature = b64encode(HMAC(b64decode(KEY), sign_key_bytes, sha256).digest())
    rawtoken = {
        'sr' :  URI,
        'sig': signature,
        'se' : str(int(ttl))
    }
    rawtoken['skn'] = POLICY

    return 'SharedAccessSignature ' + urlencode(rawtoken)

def read_temp_raw():
    f = open(DEVICE_FILE, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        return temp_c
def send_message(token, message):
    url = 'https://{0}/devices/{1}/messages/events?api-version=2016-11-14'.format(URI, IOT_DEVICE_ID)
    headers = {
        "Content-Type": "application/json",
        "Authorization": token  }
    data = json.dumps(message)
    print (data)
    response = requests.post(url, data=data, headers=headers)
if __name__ == '__main__':
    # 1. Enable Temperature Sensor
    os.system('modprobe w1-gpio')
    os.system('modprobe w1-therm')
    # 2. Generate SAS Token
```

```
token = generate_sas_token()

 # 3. Send Temperature to IoT Hub

 while True:

   temp = read_temp()

   message = { "temp": str(temp) }

   send_message(token, message)

   time.sleep(1)
```

1. The script imports necessary modules: **base64** for encoding and decoding data, **hashlib** for cryptographic hash functions, and **urllib.parse** for URL encoding, **hmac** for generating message authentication codes, **requests** for making HTTP requests, **JSON** for working with JSON data, **os** for executing operating system commands, and **time** for working with time-related functions.

2. The script defines some constants related to the temperature sensor and Azure IoT Hub.

   - **BASE_DIR** represents the base directory path where the temperature sensor data is stored.
   - **SENSOR_DEVICE_ID** is the unique identifier of the temperature sensor.
   - **DEVICE_FILE** is the file path to access the temperature sensor data.
   - **URI** is the URI of the Azure IoT Hub.
   - **KEY** is the shared access key used for authentication.
   - **IOT_DEVICE_ID** is the identifier of the IoT device (Raspberry Pi in this case) sending the temperature data.
   - **POLICY** is the name of the policy used for authentication.

3. The function **generate_sas_token()** is defined to generate a Shared Access Signature (SAS) token used for authentication with the Azure IoT Hub. It calculates the signature by creating a string composed of the URI and the token's expiry time, then encodes and signs it using HMAC-SHA256. The token includes the URI, signature, and expiry time.

4. The function **read_temp_raw()** reads the raw temperature data from the device file.

5. The function **read_temp()** reads and returns the temperature in Celsius. It calls **read_temp_raw()** and waits until the sensor is ready. It then extracts the temperature value from the data and converts it to Celsius.

6. The function **send_message(token, message)** sends a message containing the temperature data to the Azure IoT Hub. It constructs the URL for sending the message using the URI and device ID. The token is included in the request headers

for authentication. The temperature data is serialized as JSON and sent as the request payload.

7. The script's main execution starts with an **if __name__ == '__main__':** block.
8. Inside the block, the script enables the temperature sensor by running the **modprobe** commands using **os. system()**.
9. It generates a SAS token by calling **generate_sas_token()**.
10. It enters a loop that continuously reads the temperature from the sensor using **read_temp()**, constructs a message with the temperature data, and sends it to the Azure IoT Hub using **send_message()**. The loop runs indefinitely with a delay of 1 second between each iteration.

The codes' function summarizes that it reads temperature data from a sensor and sends it to Azure Iot Hub for further processing and analysis, and further on, with the help of Power Bi, presents a live feed of the data value.

3.3 Azure

After configuring the RPI, hardware, and implementing the code Azure is used to collect data. Furthermore, a live presentation that shows the current temperature and live line chart on temperature development, the live visualization is done in co-operation with Power Bi. The goal is to push the sensor data from the RPI to Azure using the resource "IoT Hub" and then build on it with the resource "Stream Analytics" to get the presentation into Power Bi for real-time visualization. Power Bi in this case is used as a SAAS (Software as a service). This project is free-to-use in Azure as long as the user only uses Azure subscription 1, which is free when the user has entered the payment method. It allows up to 8000 events per day.

Starting the configuration of the IoT Hub, it can be found when the user creates a new resource under the "Internet of Things" tab. After that, by following the instructions on the screen but keep in mind, if this is the first IoT hub as a resource, the user chooses the free version which allows up to 8000 messages per day. When this is created, the user needs to create a "Stream Analytics job" that can be found using the search bar. After these two are created we need to go into the IoT Hub to register and connect the RPI by choosing the "IoT Devices" tab on the left-hand side and adding a device. In a real-life scenario, thousands of devices could be connected, but for this project, only an RPI is in use, so the device ID can be kept simple, keep “Auto Generate Keys” ticked and enable the IoT Hub. When that is done, the user needs to create a consumer group for the IoT Hub that the Stream Analytics Job will retrieve its data. Under the “messaging” tab, the user needs to create endpoints, click the events tab, create a consumer group name, and hit save. That is IoT Hub done for now.

Next, head into Stream Analytics Job and add our inputs and outputs. Add stream input, choose IoT Hub and name the input, change the “Consumer group” to the one that was created earlier, and done. Next up, configure the outputs. This is an output for the Power Bi dataset, so it will connect to Power Bi, name the output, dataset, and table name, authorize, and save. Keep in mind that later when writing the query, one cannot use underscores or special signs, so keep the name in lowercase letters (string type).

Now when that is done, start configuring the RPI to start sending data, the code used in this project will be shown and explained in the “The Code” section. When the Python script is up and running and, in the background, sending data using the IoT hub REST API which acts as the gateway to send events to the cloud.

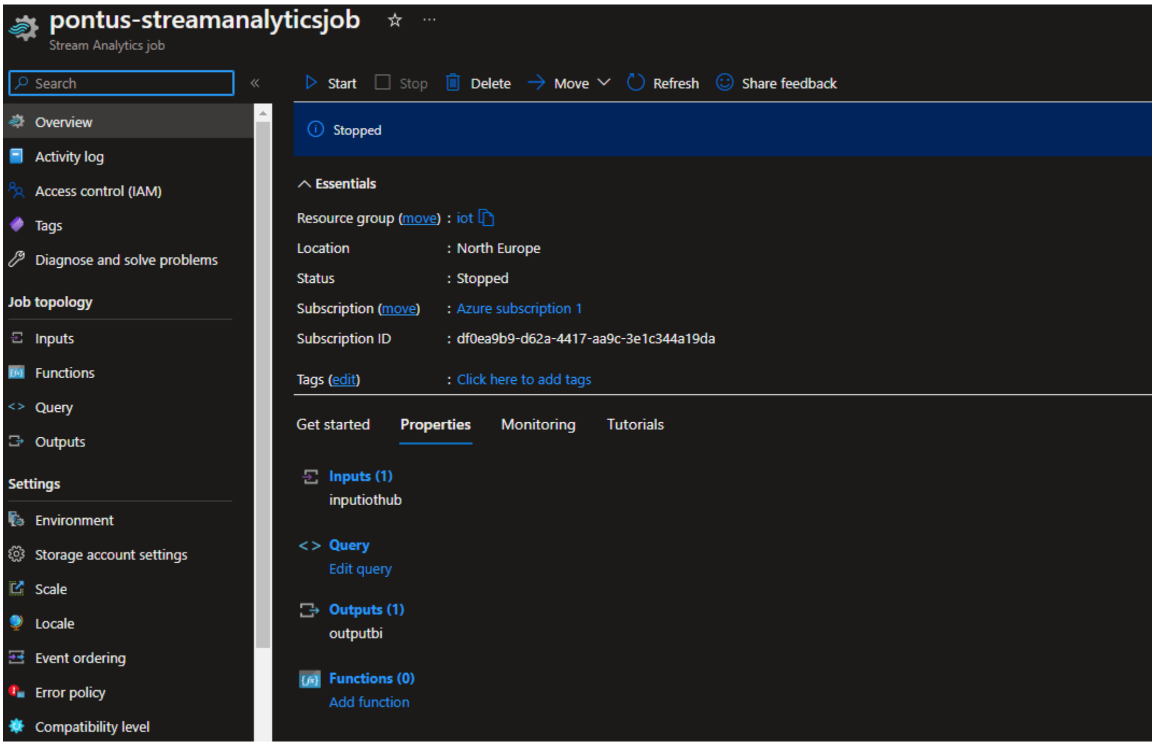

Figure 13. Stream Analytics Job, Overview page. (Microsoft Azure, 2023)

3.3.1 Visualization

To get a live feed it is needed to set up the output for Power Bi, by navigating to the output within “stream analytics job” adding new output, and choosing Power Bi. Choose a name for the output and a “Dataset Name” (this will be present in Power Bi), after this, a prompt to log in or sign up to Power Bi will pop up. Check the Query to see that everything is set up correctly. (Figure 16)

Test query   Save query   Discard changes

```
SELECT
    CAST(iothub.EnqueuedTime AS datetime) AS event_date,
    CAST(temp AS float) AS temp
INTO
    outputbi
FROM
    inputiothub
```

Figure 14. Test Query, (Microsoft Azure, 2023)

Make sure that the following is correct in the Python script, or it will not work properly.

- Sensor_Device_ID: on the RPI navigate to sys/bus/w1/devices, the ID should something like this (28-011620f167ee).
-
- URI = “YOUR_IOT_HUB_NAME.azure-devices.net”.
- KEY = In IoT Hub, navigate to “Shared Access Policies”, iothubowner, and copy the “Primary key”.
- IOT_Device_ID = the registered IoT device, in this project “pontus-iot-hub”.

To be noted if running Raspbian, it is most probably need to install pip (Package Installer for Python) which can easily be done from the terminal by entering “*sudo apt-get install python-pip*). The code used in this project has dependencies on the request library. Messages and events that are being sent to IoT Hub are via “REST API”.

Before the user can present the project live in Power Bi, a test is needed to configure, by going to the query in Stream Analytics, selecting “query”, and selecting test (the program/code on the RPI needs to be sending events). Then by selecting the “three dots” in the input and choosing “sample data from input” it will then present a live feed from the sensor as shown in (figure “Web version”). Only presenting relevant information showing in Power Bi with the code:

“CAST(iothub.EnqueuedTime AS datetime) AS event_date”, “CAST(temp AS float) AS temp”. When this is running, go to Stream Analytics Job and click Start.

Now all is ready to configure our Power Bi, log in to Power Bi and select Create a new Dashboard (My Workspace > Create > Dashboard). This project uses two “custom streaming data” tiles using “TempDataSet”, one as a card and the other as a line chart, as the program is already up and running, it should now already be seeing the line chart moving, to test this put the probe in the hand or if waterproof, in a glass of water, and watch the temperature change.

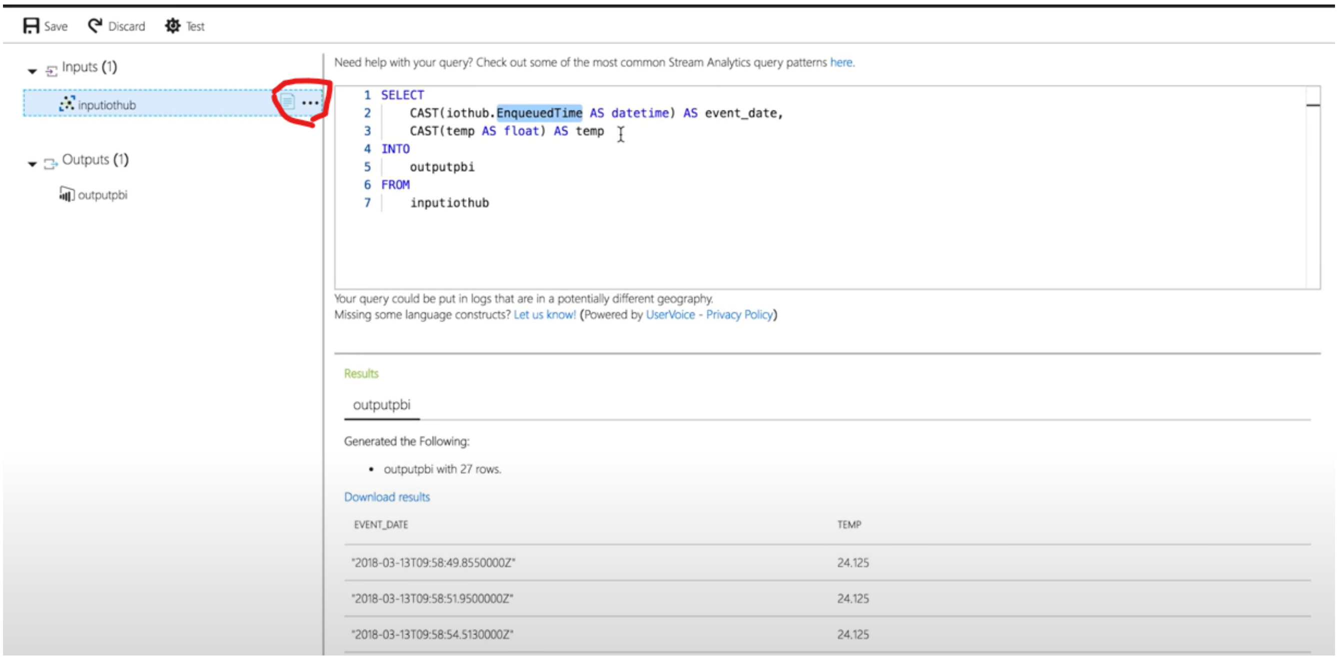

Figure 15. Query Test. (Microsoft Azure, 2023)

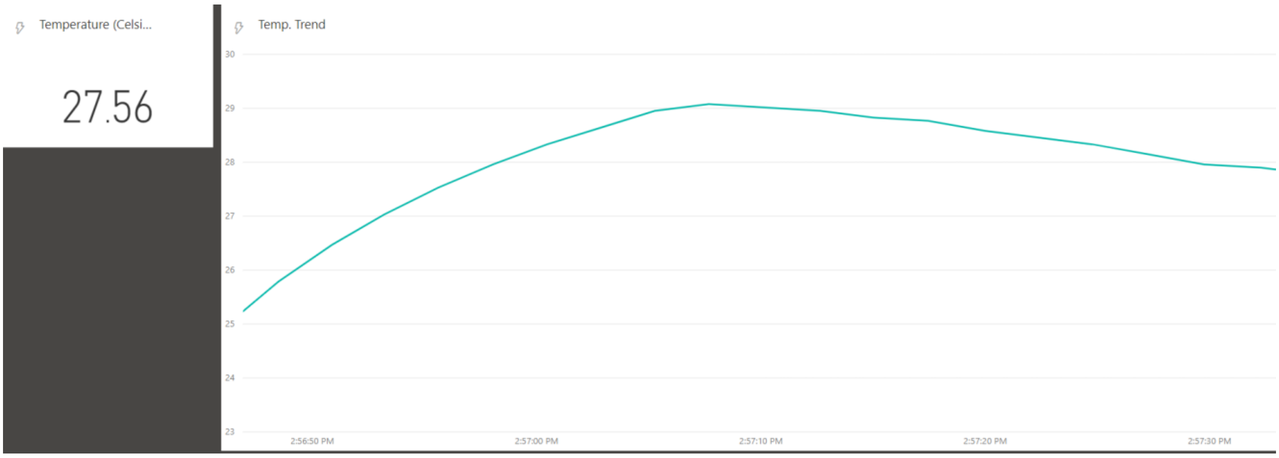

Figure 16. Web version. (Power Bi interface, 2023)

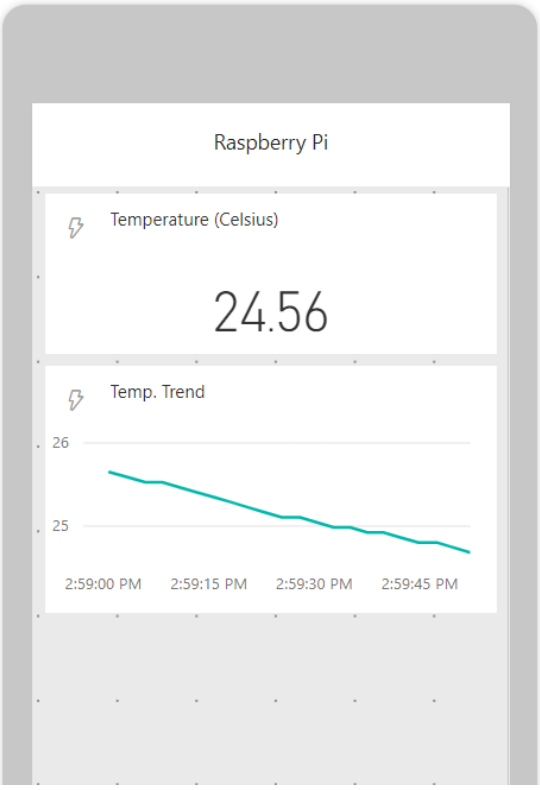

Figure 17. Mobile version. (Power Bi interface, 2023)

3.4 Wiring

The connection is done by using simple jump wires, breadboard GPIO ribbon cable, temp sensor DS18B20 (water resistant version), and one 4,7K ohm resistor, then, of course, a Raspberry Pi. (Connection diagram below)

First up, the sensor has three wires, black=ground/white=Data(GPIO)/Red=3.3V, this goes anywhere on the breadboard but make sure it is convenient when wring the rest of the circuit. Start with connecting the 3.3V according to the below drawing then plug the data jump cable to GPIO4 (this may differ on the ribbon cable), and then connect the black jumper wire to the GND (ground lastly the resistor connects the 3.3 Volts to the signal wire(data).

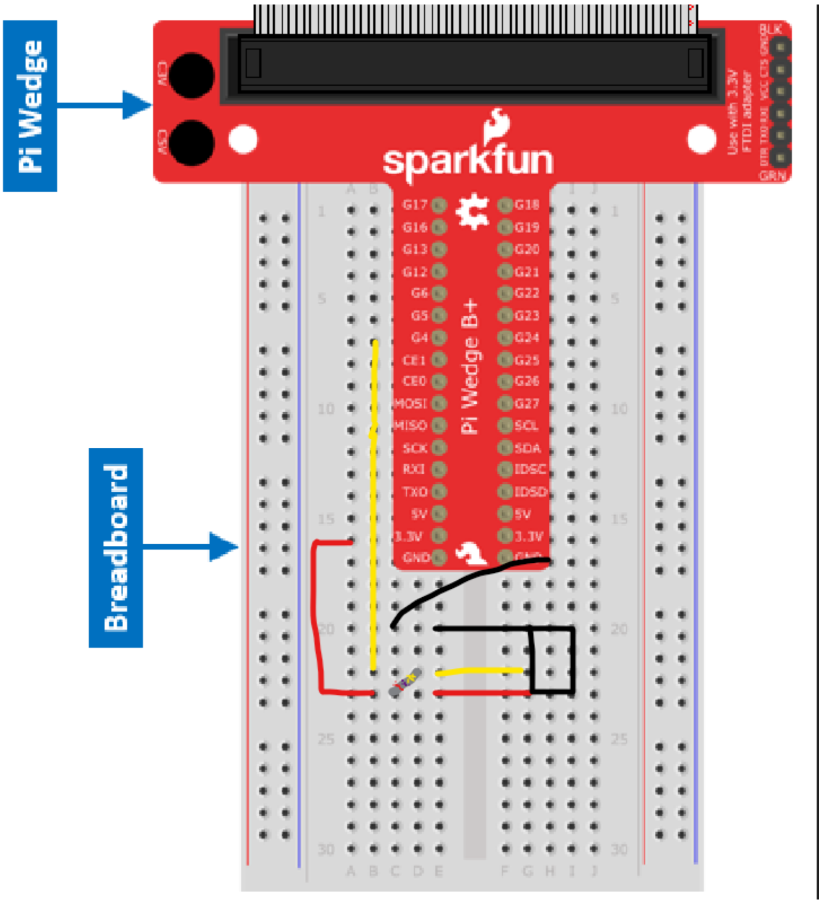

Figure 18. Connection diagram. (SCIENCE BUDDIES , 2023)

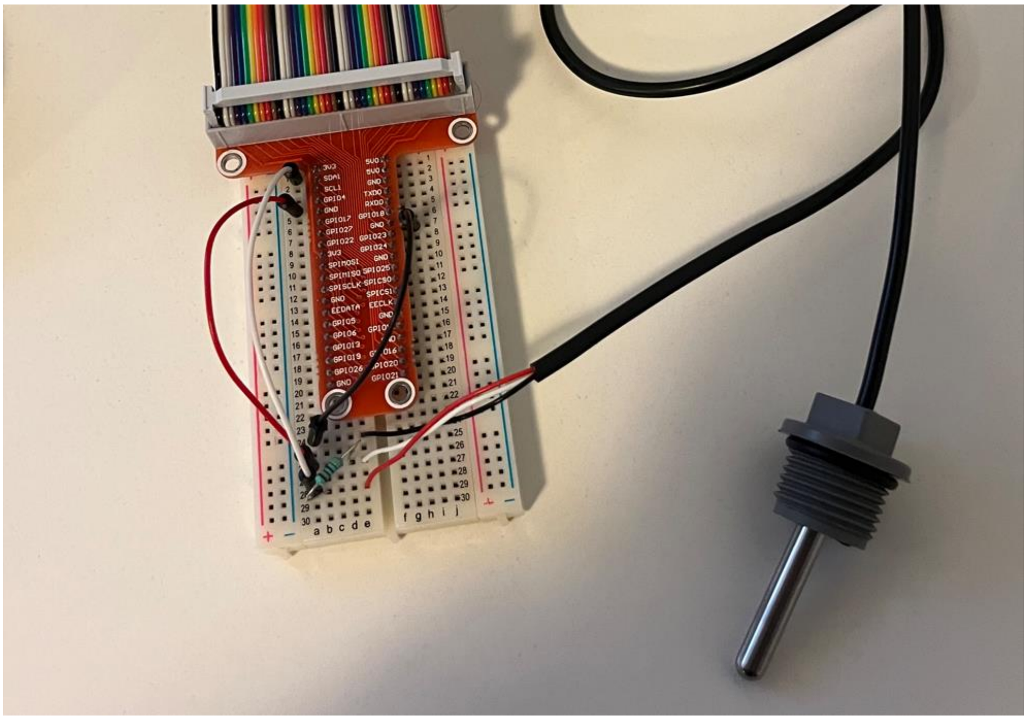

Figure 19. Connection with a waterproof sensor.

4 Results

The goal of this thesis was to get familiar with and get and deeper understanding of IoT and Microsoft Azures IoT functionality. To represent a live project that could be implemented in the real world. I wanted to create a project that could have a beneficial impact in a real-life scenario, the project was to enable remote access to data from your smartphone or computer, in this case, keep track of the temperature in your summer cottage. With the ability to expand the project to more complex data and analysis, for example, humidity, etc.

From a theoretical point of view, this was a success, the project is ready to be implemented provided that you have power and a Stable Internet connection. During the research for this thesis, I have accomplished and learned a lot, and really gotten a broad and deep understanding of IoT, which is getting more and more important in today's society, and this was also one of the goals of this thesis.

The conclusion of this thesis was overall achieved, setting up a Raspberry Pi with the configuration of remote access and implementation of connecting a temp sensor, building a code to steam temperature data for future analysis. Setting up Microsoft Azure IoT hub to push the data from the RPI to the cloud, further visualize with a live feed by setting up a Power Bi configuration to present the data. The good thing about this thesis is that it has been done in a simplified and user-friendly, further, follow the handbook to get started.

5 Discussion

There are many things you could bring forward, implement, and further expand on this project and this thesis can be a start-up guide to that. You could easily with some small modifications to the Python script and Azure configuration set up multiple sensors to analyze different kinds of data.

5.1 Own Thoughts

I have felt really satisfied working on this project, I have gotten an actual interest in implementing IoT in my day-to-day activities, and already thinking of getting started on me on project based on this thesis. A smart home solution is also knocking on the door and plan to implement it in my flat. I would have wanted to get a working database to store data values in, but due to time limitations and cost issues, this was left out and can be considered as the downside of this thesis.

5.2 Challenges

At the beginning of the process, there were a few challenges, grasping the goal and what the result should be. Where to limit the thesis and how to frame it. During the actual work there were only a few hiccups but after reaching the issues, finding a solution was no major complication. The biggest issue was finding solutions to errors in the Python script, with outdated libraries, etc., and how to overcome them, testing until successful was the solution.

5.3 The Database

The database was in the planning stage in the scope of the thesis, but after reaching, the conclusion to leave it out due to time and not cost efficient to implement in Azure due to high startup and maintenance costs.

6 References

1-Wire. (2023). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/1-Wire

A primer on Cloud Computing. (2019). Retrieved from Medium: https://medium.com/@colinbaird_51123/a-primer-on-cloud-computing-9a34e90303c8

AWS Vs Azure Vs GCP. (2023). Retrieved from Youtube: https://www.youtube.com/watch?v=nrqmYvjHHJg&list=PLKbVK16BrZjka1nD_gA8GoyQyBJu0nWAd&index=8&ab_channel=Simplilearn

Cloud computing in 6 minutes. (2023). Retrieved from Youtube: https://www.youtube.com/watch?v=M988_fsOSWo&ab_channel=Simplilearn

Coursera. (2023). Retrieved from Top 5 of Programming Languages: https://www.coursera.org/articles/types-programming-language

Main cloud service models: IaaS, PaaS and SaaS. (2022). Retrieved from Stackscale: https://www.stackscale.com/blog/cloud-service-models/

Microsoft Azure. (2023). Retrieved from Stream Analytics Interface: https://portal.azure.com/#@yhnovia.onmicrosoft.com/resource/subscriptions/df0ea9b9-d62a-4417-aa9c-3e1c344a19da/resourceGroups/iot/providers/Microsoft.StreamAnalytics/streamingjobs/pontus-streamanalyticsjob/queryV2

Microsoft Power BI. (2023). Retrieved from TechTarget: https://www.techtarget.com/searchcontentmanagement/definition/Microsoft-Power-BI

Overview of 1-Wire Technology and Its Use. (2023). Retrieved from Analog Devices, Maxim Integrated: https://www.analog.com/en/technical-articles/guide-to-1wire-communication.html

Power Bi interface. (2023). Retrieved from Power Bi: https://app.powerbi.com/groups/me/dashboards/69e6fe6e-ea30-4882-8f5e-a76385a9cb52?language=en-US

Programmable Resolution 1-Wire Digital Thermometer. (2022). Retrieved from Maxim integrated: https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf

Python Tutorial for Beginners. (2023). Retrieved from Simplilearn: https://www.simplilearn.com/tutorials/python-tutorial#why_learn_python

Raspberry Pi 4 Tech Specs. (2023). Retrieved from Raspberry Pi: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/

Raspberry Pi Alternatives: 14 Best Single Board Computers in 2023. (2023). Retrieved from seeedstudio: https://www.seeedstudio.com/blog/2020/10/20/raspberry-pi-alternatives-17-best-single-board-computers-in-2020/

SCIENCE BUDDIES . (2023). Retrieved from getting started with your Raspberry Pi Projects Kit: https://www.sciencebuddies.org/science-fair-projects/project-ideas/raspberry-pi

Simplilearn. (2023). *What Is Azure? | Microsoft Azure Tutorial For Beginners*. Retrieved from Simplilearn: https://www.simplilearn.com/tutorials/azure-tutorial/what-is-azure

The Different Versions of the Raspberry Pi. (2022). Retrieved from pimylifeup: https://pimylifeup.com/raspberry-pi-versions/

What is computer program? (2022). Retrieved from Learn Computer Science: https://www.learncomputerscienceonline.com/what-is-computer-program/

What is IoT? (n.d.). Retrieved from educba: https://www.educba.com/what-is-iot/

What is IoT? (2022). Retrieved from AWS: https://aws.amazon.com/what-is/iot/

Why should you automate your business processes? (2022). Retrieved from Vault: https://blog.vault-erp.com/post/2021/07/06/why-should-you-automate-your-business-processes

Wikipedia. (2022). Retrieved from https://en.wikipedia.org/wiki/Raspberry_Pi

7 Annex 1 The code

```
from base64 import b64encode, b64decode

from hashlib import sha256

from urllib.parse import quote_plus, urlencode

from hmac import HMAC
```

```
import requests

import json

import os

import time

# Temperature Sensor

BASE_DIR = '/sys/bus/w1/devices/'

SENSOR_DEVICE_ID = '28-00000a54bc7e'

DEVICE_FILE = BASE_DIR + SENSOR_DEVICE_ID + '/w1_slave'

# Azure IoT Hub

URI = 'Pontus-iot-hub.azure-devices.net'

KEY = 'elmb3mp2abphZiSV1zHsUdKISqg6ReSBiowyH7LWTQE='

IOT_DEVICE_ID = 'raspberrypi'

POLICY = 'iothubowner'

def generate_sas_token():

  expiry=3600

  ttl = time.time() + expiry

  sign_key = "%s\n%d" % ((quote_plus(URI)), int(ttl))

  sign_key_bytes = sign_key.encode("utf-8")

  signature = b64encode(HMAC(b64decode(KEY), sign_key_bytes, sha256).digest())

  rawtoken = {

    'sr' :  URI,

    'sig': signature,

    'se' : str(int(ttl))

  }

  rawtoken['skn'] = POLICY


  return 'SharedAccessSignature ' + urlencode(rawtoken)


def read_temp_raw():

  f = open(DEVICE_FILE, 'r')

  lines = f.readlines()

  f.close()

  return lines
```

```
def read_temp():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        return temp_c


def send_message(token, message):
    url = 'https://{0}/devices/{1}/messages/events?api-version=2016-11-14'.format(URI, IOT_DEVICE_ID)
    headers = {
        "Content-Type": "application/json",
        "Authorization": token
    }
    data = json.dumps(message)
    print (data)
    response = requests.post(url, data=data, headers=headers)


if __name__ == '__main__':
    # 1. Enable Temperature Sensor
    os.system('modprobe w1-gpio')
    os.system('modprobe w1-therm')

    # 2. Generate SAS Token
    token = generate_sas_token()

    # 3. Send Temperature to IoT Hub
    while True:
        temp = read_temp()
```

```
message = { "temp": str(temp) }

send_message(token, message)

time.sleep(1)
```