MUTUAL RECOGNITION DEVICES VIA INFRARED TRANSCEIVER AND CLOUD COMPUTING

A THESIS SUBMITTED TO
THE FACULTY OF ARCHITECTURE AND ENGINEERING
OF
EPOKA UNIVERSITY

BY

NURI SEKO

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

JUNE, 2022

Approval sheet of the Thesis

This is to certify that we have read this thesis entitled "Mutual Recognition Devices via Infrared Transceiver and Cloud Computing" and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

____

Dr. Arban Uka
Head of Department
Date: June, 17, 2022

Examining Committee Members:

Dr. Arban Uka	(Computer Engineering)	_________________
Assoc. Prof. Dr. Carlo Ciulla	(Computer Engineering)	_________________
Dr. M. Maaruf Ali	(Computer Engineering)	_________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Surname: Nuri Seko

Signature: _______________

ABSTRACT

MUTUAL RECOGNITION DEVICES VIA INFRARED TRANSCEIVER AND CLOUD COMPUTING

Seko, Nuri

M.Sc., Department of Computer Engineering

Supervisor: Dr. M. Maaruf Ali

Through this paper, the possible mechanism of a device that recognizes other devices in a meeting setting is discussed. The purpose of such a device is to simplify the physical human interactions during certain meetings and to help people network with one another. This functionality is achieved through the use of an infrared laser properly power rated for everyday use, internet connectivity of the sensor via microcontroller boards like Arduino Nano 33 IoT or ESP32, and the cloud integration general mechanism. The paper will mainly focus on the hardware of the whole system and the integration capability to the cloud, rather than the cloud structure itself. Reviews of the Infrared and other optical communication methods, and the specific functionality we are trying to achieve show that it should be possible for this device to be functionally implemented in meeting settings of medium to low densities. So in a conclusion, the implementation of a device with the above-mentioned functionality and mechanism is possible, but the accuracy would have to suffer due to user dependency.

Keywords: *Infrared Laser, Microcontroller boards, Arduino Nano 33 IoT, Cloud Integration, User Dependency*

ABSTRAKT

PAJISJE NDERNJOHESE ME ANE TE DRITES INFRA TE KUQE DHE KOMPUTIMIT NE KLAUD

Seko, Nuri

Master Shkencor, Departamenti i Inxhinierisë Kompjuterike

Udhëheqësi: Dr. M. Maaruf Ali

Nëpërmjet këtij punimi, diskutohet mekanizmi i mundshëm i një pajisjeje që njeh pajisjet e tjera në një ambjent ku po kryhet një mbledhje ose event. Qëllimi i një pajisjeje të tillë është të thjeshtojë ndërveprimet fizike njerëzore gjatë takimeve të caktuara dhe të ndihmojë njerëzit të komunikojnë me thjeshtësisht me njëri-tjetrin. Ky funksionalitet arrihet nëpërmjet përdorimit të një lazeri infra të kuq me vale të pështatur për përdoring të përditshem, filtrimit praktik të rezultateve, lidhjes me internet të sensorit nëpërmjet moduleve te microcontrollerave si Arduino Nano 33 IoT ose ESP32 dhe mekanizmit të përgjithshëm të integrimit në Cloud. Punimi do të fokusohet kryesisht në harduerin e të gjithë sistemit dhe aftësinë e integrimit në cloud, në vend të vetë strukturës së resë. Rishikimet e komunikimeve me anë të valëve infa të kuqe dhe metodave të tjera komunikimi optik dhe funksionaliteti specifik që po përpiqemi të arrijmë tregojnë se duhet të jetë e mundur që kjo pajisje të zbatohet funksionalisht në cilësimet e takimeve me densitet të mesëm dhe të ulët. Pra, si përfundim, zbatimi i një pajisjeje me funksionalitetin dhe mekanizmin e lartpërmendur është i mundur, por saktësia do të duhej të vuante për shkak të varësisë nga përdoruesi.

Fjalët kyçe: *Lazer Infra të kuqe, Module Microcontrolleri, Arduino Nano 33 IoT, Integrimi ne Cloud,* Varësisë nga Përdoruesit

TABLE OF CONTENTS

ABSTRACT iii

ABSTRAKT iv

LIST OF TABLES vii

LIST OF FIGURES viii

CHAPTER 1 1

INTRODUCTION 1

1.1 In-Person Networking Difficulties 1

1.2 A device that helps facilitate and simplify the networking 1

1.3 Research Process 2

1.4 Organization of the thesis 2

CHAPTER 2 4

LITERATURE REVIEW 4

2.1 Introduction 4

2.2 Functionality abstraction 6

2.2.1 Remotes and Infrared pulse transmission 7

2.2.2 Infrared Pulse receiving solution 13

CHAPTER 3 15

METHODOLOGY 15

3.2.1 Hardware Integration 15

3.2.2 Connecting the Device to the Cloud 19

CHAPTER 4 26

RESULTS AND DISCUSSIONS 26

CHAPTER 5 45

CONCLUSIONS 45

5.1 Conclusions 45

5.2 Recommendations for future research 45

References 46

LIST OF TABLES

Table 1. Lower Capacitance values 30

Table 2. Higher Capacitance values 30

Table 3. R1 Frequency Response 33

Table 4. R2 Frequency Response 36

Table 5. R1 and R2 Frequency Response 38

Table 6. R1, R2, and C1 Frequency Response 41

Table 7. Prices of electronics components 44

LIST OF FIGURES

Figure 1. *Relay Schematic*. 7

Figure 2. *Remote Controller Schematic*[19] 8

Figure 3. *IR Signal Jammer*[20] 9

Figure 4. *555 Timer IC Schematic* 10

Figure 5. *Sample frequency circuit with a 555 timer* 11

Figure 6. *Laser Schematic* 12

Figure 7. *IR Receiver Module* 14

Figure 8. *Arduino Nano 33 IoT Pin Layout* 15

Figure 9. *Arduino Nano 33 IoT IR Sensor Module Wiring* 17

Figure 10. *Arduino Nano 33 IoT IR LED Wiring to act as a Transmitter* 18

Figure 11. *Arduino Nano 33 IoT IR LED PCB* 26

Figure 12. *Arduino Nano 33 IoT Receiver PCB* 27

Figure 13. *NE555 timer IC Transmitter PCB* 27

Figure 14. *NE555 timer IC circuit for frequency response testing* 28

Figure 15. *NE555 timer IC waveforms in different points of the circuit* 29

Figure 16. *Frequency response as a function of the lower values of capacitance* 32

Figure 17. *Frequency response as a function of the higher values of capacitance* 33

Figure 18. *Frequency response as a function of the lower values of R1* 35

Figure 19. *Frequency response as a function of the higher values of R1* 35

Figure 20. *Frequency response as a function of the lower values of R2* 37

Figure 21. *Frequency response as a function of the higher values of R2* 38

Figure 22. *Frequency response as a function of the sum of R1 and R2* 40

Figure 23. *Frequency response as a function of the ratio of R1 and R2* 41

Figure 24. *Total variability frequency response as a function of the lower values* .. 42

Figure 25. *Total variability frequency response as a function of the higher values* . 43

CHAPTER 1

INTRODUCTION

1.1 In-Person Networking Difficulties

With the technology having advanced in the direction of digitalization the interpersonal communication in an in-person setting has become increasingly anxiety inducing for a big part of the population. This is an issue for people, especially in conference settings where networking is a primary purpose of the activity. And this is why I have chosen for the subject of my research to be a device that can support us in this task.

Networking is what drives the exchange of experiences, thoughts, and ideas. So facilitating, simplifying and by these means amplifying it should be unpredictably beneficial to the society, but beneficial nonetheless. I believe that the right approach to intercepting the problem that digitalization has created in the way people interact with one another live is by getting rid of the anxiety of going out and talking to a person you don't know much about if anything at all. Creating a profile is a common thing nowadays in the digital world, but that is not nearly as present in the everyday interaction, so I believe there needs to be a bridge between these two worlds, so that the behavior of people can be universal once again and the norms of communication to be updated in the live setting too. Considering that the problem was caused by the development of beneficial technology it is natural to seek a solution that maintains those benefits while introducing a simplicity to the interaction.

1.2 A device that helps facilitate and simplify the networking

The way I believe that this problem meets a solution is by means of a device that can recognize other devices in a physical level. What that means is that we create a device that is both a receiver and a transmitter of signals with the capability of targeting and recognizing one another and being connected to the internet so that profiles can be shared from the common database. The approach chosen to be explored through this paper is that of an infrared laser mechanism where you have an infrared laser from 1 device that points to the receiver of another device and sends a signal. The signal sent via the transmitter is a unique sequence belonging to only 1 of the devices, which are since the beginning of the event assigned to a specific individual, thus a specific profile. So once the unique signal has been received then the device now has a target location to send its profile to. This target profile value is sent to the Cloud containing all the profiles via a Wifi module attached to our device and an API key pertaining to the Cloud. Once the connection is achieved and the unique sequence of the receiver of the profile is sent to the Cloud, now the cloud knows the receiver of the profile and which profile to send, thanks to knowing which device made the request to send a profile. An additional feature would be to keep a log of every person that has requested a profile to be sent to them, so that the owner of the profile can be aware of the people interested in them.

1.3 Research Process

The purpose of this research paper is to present a possible hardware solution to the problem of anxiety in live setting networking. The research approach to this paper will be mainly Datasheet driven and hardware implementation-oriented with a bit of coding and interfacing to achieve the above-mentioned functionality. The research resources will be a mix of datasheets of the materials, which are provided by the manufacturers, and online research for the way different protocols of communication and data transmission work until the wifi module and then further on with internet connectivity.

1.4 Organization of the thesis

This thesis is divided in 5 chapters. The organization is done as follows:

In Chapter 1, the problem statement, thesis objective and scope of works is presented. Chapter 2, includes the literature review, which is further divided into an introduction part and a data extraction and transceiving. The latter is further divided into: Remotes and Infrared pulse modulation, Hardware of Receiver and Transmitter, Communication Protocols, Connecting the Device to the Cloud. Chapter 3, consists of the methodology followed in this study. In Chapter 4, the results of the implemented parts are shown. In Chapter 5, conclusions and recommendations for further research are stated.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Networking in person is generally seen as something that is already simple enough, although that has been the case for all other improvements to our everyday life. They are deemed unnecessary until the moment that they become the norm. The way this paper is structured is like a functionality branch that by process of elimination achieves the desired functionality in the most practical and economical way indicated by the research. To go ahead with the branching and elimination process we start by the very first problem, which is determining what the problems with the way things are now are. To do that a few papers concerning human behavior in social settings were analyzed[1]. Analyzed the severe effects a low social performance can have on an individual's psychology[2][3]. While also becoming aware of what the differences between online and onsite networking are, especially in a professional networking setting[4]. The research above pointed to 1 main innate problem that people have with networking, which is that it is highly anxiety inducing. This anxiety increase can be allocated to a few causes, such as low perceived social performance, fear of the unknown, paralysis due to lack of information and the high perceived threshold of taking action[5]. Most of the above-mentioned causes of anxiety can be fixed with the presence of information in the equation, so once information is available to people the threshold to take action gets lowered, you're no longer uninformed about the person you want to network with and you're better informed as to who you want to network with in the first place. So to facilitate that I will be presenting a theoretical discussion on the approach that needs to be taken for a device to be built for such a purpose. In my research on the internet and various papers, I was unable to find a device with the same purpose as this one so the idealization will be fully supported by my problem-solving approach. Taking about the approach I will be going step by step as through the capabilities that the device needs to have for it to be able to achieve the final product.

The list of the functionality would be: means of targeting a person, unique identification of each device and its signal, means of receiving the targeting, hardware requirements and capabilities, internet Connectivity, and display portal.

In choosing the right approach for this paper the considerations start from the basis, which is the ways of communication between two devices can happen, and which one is best for our purpose.

Methods of electronic communication are:

- Wired

Pros: Communication is very simple and identifiable. It has a high reliability.

Cons: Physically restricting and requires a highly regulated and static setting to function.[6]

- Infrared

Pros: Related to the physical work and intuitively maneuverable. Not visible to the eye

Cons: Accuracy suffers and needs a controlled implementation.[7]

- Bluetooth

Pros: Well used. Relatively fast.[8]

Cons: Requires physical proximity and a set of permissions to establish communication.[6]

- Wifi

Pros: Well used. Fast. Practical in purely digitalized solutions.

Cons: Requires physical proximity and a set of permissions to establish communication.[6]

- Light

This is practically the same as infrared, with the added disadvantage of being less practical due to being visible, distracting and sometimes even harmful to people, in epileptic people’s cases.

So based on the above-mentioned pros and cons or each of the methods the conclusion to use infrared communication was decided. Once the infrared solution was decided we also need to know the ratings of the power of the transmitter and the form of it. So the power[9][10] and whether we will use an LED infrared[11], which is efficient or a Laser infrared[12] which is precise. During the consideration for the LED a solution via the use of a Kinect Depth Sensor[13] was also considered.

For the transmission to be functional we need to pulse modulate[14] the infrared signal transmitted so that the signal is uniquely identifiable[15] for each of the devices. The next step would be the receiving side of the uniquely identifiable signal, which is dependent on the transmitting side as the total accuracy and reliability needs to be taken into account. For that the specs of the receiving sensor should be examined, be it power feed and linear sensitivity.

The next step would be to tie this all together in a hardware device schematic that enables the desired functionality[16]. For this, the considerations that need to be taken are the microcontroller, wifi module, and the extremities to enable the transmitting and sensing. The judgment will be made based on price and datasheet specifications from the manufacturers.

The last step would be the internet connectivity configuration and programming of the wifi module and microcontroller[17]. The code would need a slight modification every time the devices change location or a specific extender router with the same settings would need to be used. Future iterations may even see over-the-air, OTA, code updates of the device via the Arduino Cloud[18].

2.2 Functionality abstraction

On this section we will be discussing in greater details the reasons why the device has been implemented the way it has been implemented. To give a general overview of the whole mechanism of the data extraction, transmission and receiving, we can think of it as a relay circuit:

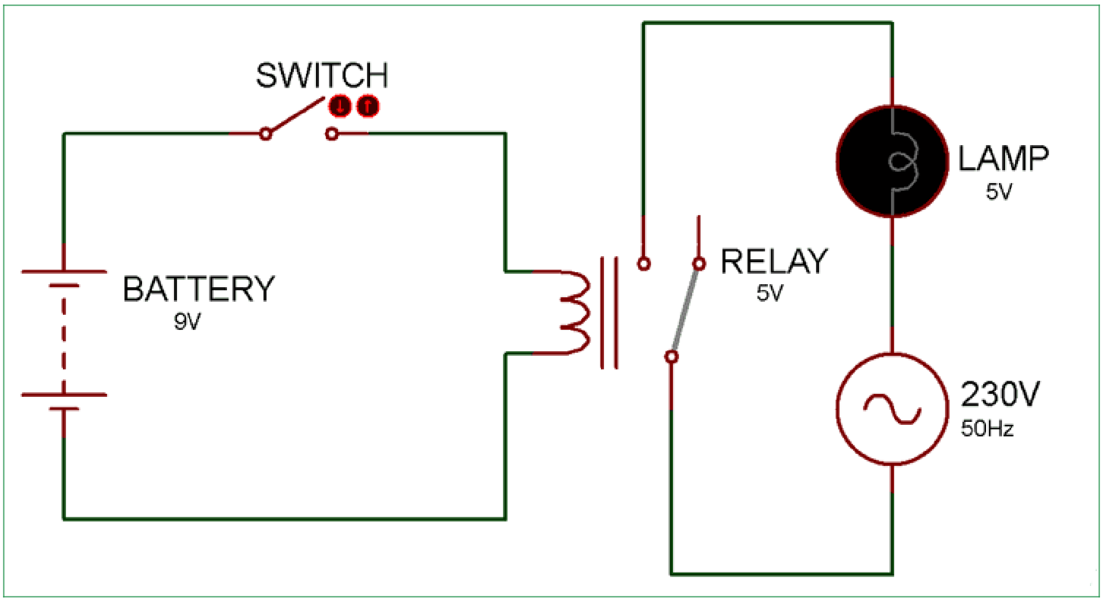

Figure 1. *Relay Schematic.*

Where the "SWITCH" component is the person requesting a profile and the small voltage activating the coil of the relay is the infrared signal which is a signal that does not require a lot of processing power. While on the bigger, main loop we have the higher bandwidth information, which is the loop that is enabled when the infrared signal is sent and is the loop that deals with sending the profile of the requested person to the request maker.

The section will go into further details in regard to the selection process for specific methods, devices, and components for reaching the functionality.

2.2.1 Remotes and Infrared pulse transmission

Speaking of the infrared communication we will be working with, it will be similar to the remotes. As it will have a unique sequence to send when a unique button is pressed. And the interpretation by the TV is done based on the received signal.

The schematic of TV remote is as shown below with the intersections of different wires being activated by the button being pressed and the specific combination leading to a specific sequence sent.

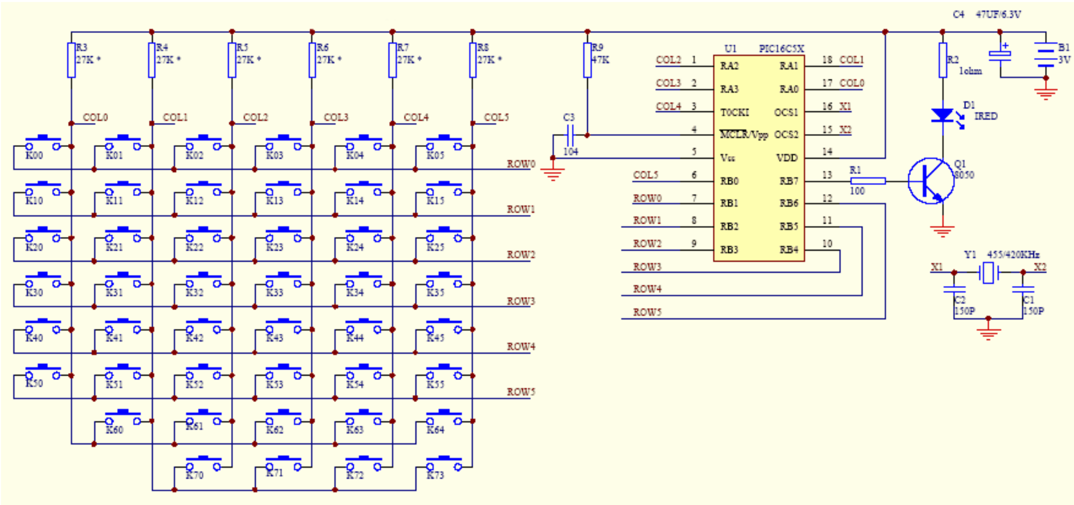

Figure 2. *Remote Controller Schematic*[19]

In the schematic above we can see the 18 pins of the microcontroller being assigned to different purposes, 6 of them are assigned as rows, 6 others are assigned as columns. These two combine to give a set of 6 unique combinations that the microcontroller can recognize. Then we have “X1” and “X2” which are the oscillating inputs connected to the quartz crystal to give the microcontroller a feed of time. We have the MCLR output pin that deals with the clear function and decides the runtimes of the specific actions defined in the program. We have VDD the input power to the microcontroller and the Vgg the ground of the microcontroller. Then we finally have the pin that is the input to the base of the transistor that opens the path to the ground for the current to flow through the infrared LED. In this case, based on the combinations of the columns and rows, we have different resistances applied to the X1, thus a different frequency is inputted at X2, being that in timer ICs we control the frequency with resistance. The higher the resistance the lower the frequency of the output. So keeping this in mind we can opt to try a non-uniform kind of device layout. This means that each device has only 1 resistance and only 1 frequency. The resistance will be unique to the device which means that the frequency will be unique to the device. A similar implementation is done in IR signal jammers with a 555 IC which is a cheap and reliable solution.

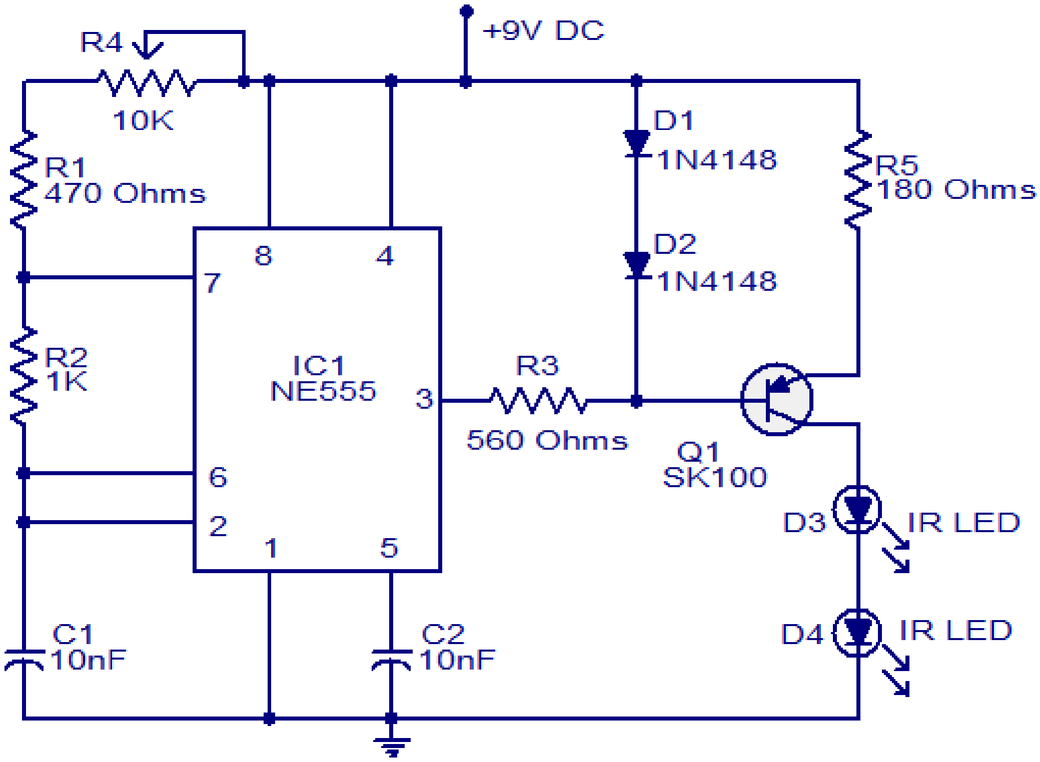

Figure 3. *IR Signal Jammer*[20]

As we can see above we have a few different resistances involved in setting up this 555 timer IC[21] so that it can act as a jammer. The jamming is enabled by the 2 infrared LEDs which interfere with the frequency of the remotes used to control the TV.

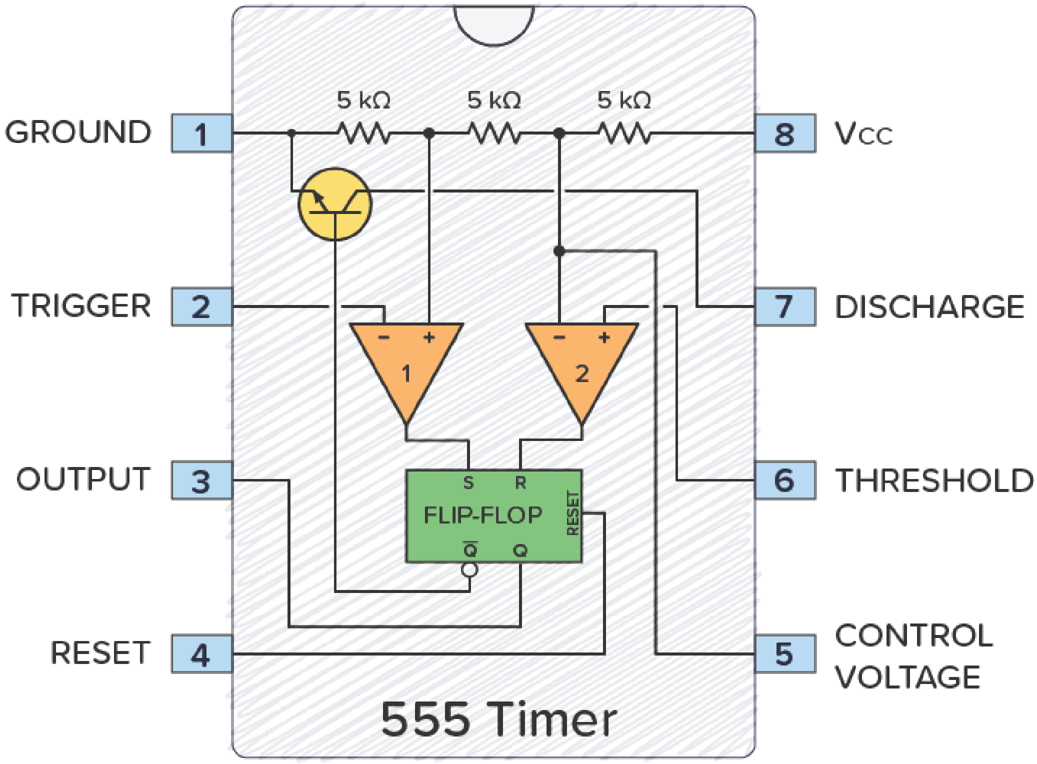

Figure 4. *555 Timer IC Schematic*

Starting with the most basic pins, functionality wise, we have pin 1 which is the ground, and pin 8 which is connected to the positive of the power source. This IC can function in a range of 5-15V and the higher the voltage within the range the more precise, dependable and measurable variability we get in our signal. Then we have pin 2 which is the "Trigger" pin, which gets activated once the voltage drops low, which is considered to be less than 1/3 of the VCC. The "Output" is 1.5V lower than the VCC when high and 0V when low. The output pin can only supply a peak current of 200mA and a steady 100mA flow as shown in the datasheet of NE555[22] model of the timer IC. The "Reset" pin is a normally high pin which when taken to a low state will reset the whole circuit. The "Control Voltage" pin does what is expected and it controls the voltage threshold for the IC time, so that the frequency can be controlled. A capacitor needs to be included so that the frequency is stable. The "Threshold" pin puts the output back to low when the voltage goes high, which is considered to be when the voltage is higher than 2/3 of the VCC. The “Discharge” pin is not connected when the output is high and is connected to ground when the output is low. Now that we have a clear idea of the 555 timer IC we know that we need to make a device with an easily accessible resistance that has a label on it. This is so a when the devices are distributed their frequency can be easily controllable and the device itself easily identifiable and assignable to a person.

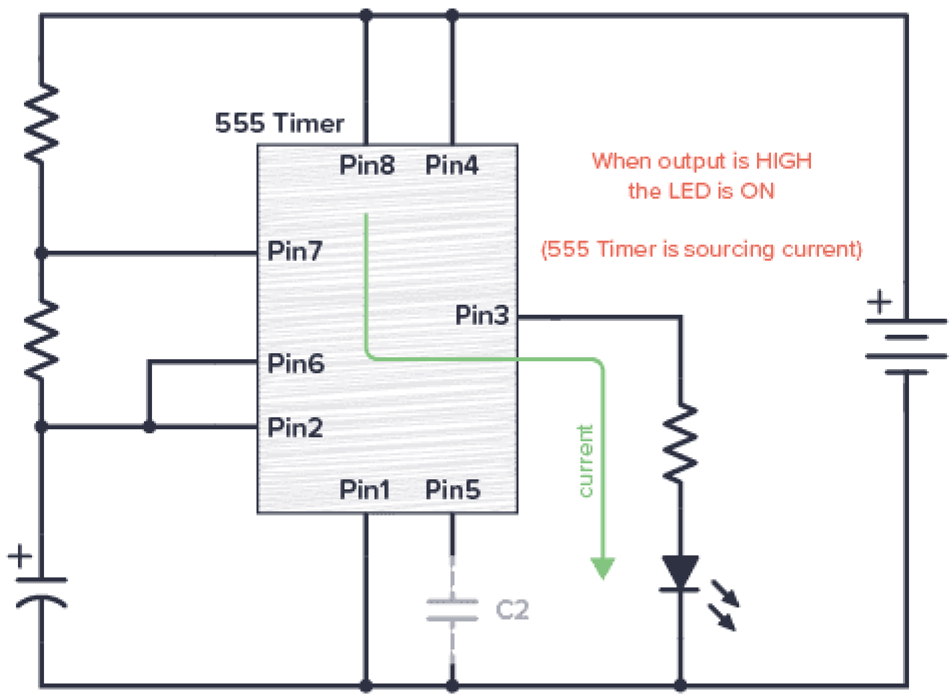

Figure 5. *Sample frequency circuit with a 555 timer*

A possible implementation of the NE555 circuit timer to act as a frequency infrared transmitter is shown above. The price of NE555 ranges from 0.20$[23] to 0.40$[24]. The cost for mass production is 0.128$[25] when the purchase is 1000 or more pcs.

There is the capability of using a microcontroller board for the purpose of simulating a remote single transfer by hooking it up to 1 of the pins and setting up a frequency for the pins. But throughout my experiences with Arduino there is a limit of frequency for the modulation frequency of the signals which is down to 2ms. Which means that with a detection time as long as 1 second we would only have 500 available frequencies which would be hardly detectable.

Now that we have a transmission handled, we can work with the light signal itself, so discussing whether the light needs to be an LED or a laser.

The LED has a wide area of effect but a short range of detectability and a high degree of undesired results in a dense area. This means that the "targeting" of the LED is too indiscriminate for this method to work in a highly dense area of receivers, which is typically how professional networking meetings are. So a more reliable and controllable means of transmission would be a laser application of it. To do that we first need to understand lasers better.

Constructing a laser needs 3 principal parts: an energy source or pump, a gain medium or laser medium, and two or more mirrors that form an optical resonator

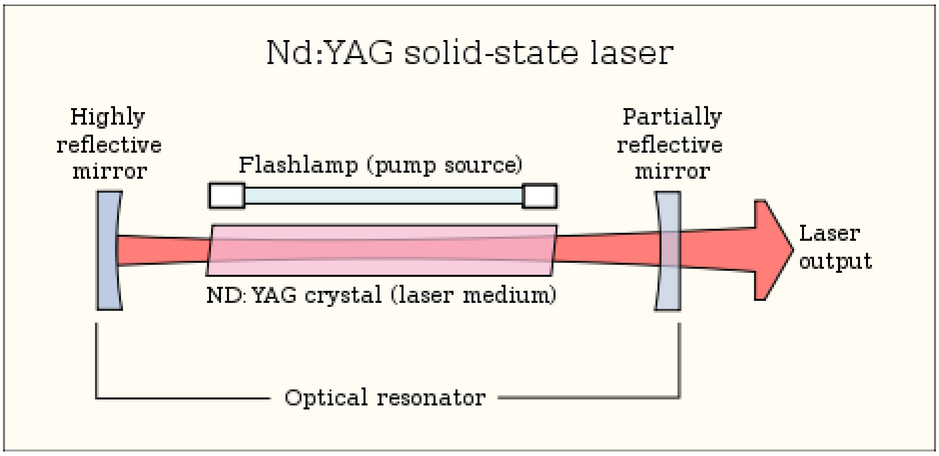

Figure 6. *Laser Schematic*

Lasers have a few parameters that determine their properties, parameters such as[26]: Laser Frequency denoted by v, Cavity Length denoted by L, Cavity width denoted by w, Facet reflectivity denoted by R, Refractive index denoted by n_g, Internal loss denoted by α_i, and Power rating denoted by p.

The parameter of most interest to us is the power rating of 5mW[27], which is the maximal power that a laser can operate at and still not be considered dangerous to human health, especially the eyes. Keeping that in mind we also need to consider the fact that lasers have an additive power effect when pointing to the same place. Considering that a 5mW power-rated pointing laser has got a 1:5 ratio of area compared to the functional area of effect, and about 1:25 ratio of functional distance compared to what we want, we can say we are safe. Taking into account a 20% safety margin we can use a laser of 4mW power with a wider area of effect and a wider cavity for a more spread ray.

Now that the specification is clear the choice for the LED is a T-1 ¾ Infrared model[28].

2.2.2 Infrared Pulse receiving solution

During this section of the paper we will be dealing with the receiver side of the device. On the previous section we had a NE555 timer IC that was doing the modulation and frequency control of the transmission of the infrared unique signal. This time we also need a means of receiving the signal and then decoding the modulated data. For this task a microcontroller is a great solution. For the sake of not overcomplicating the development process I believe choosing an already existing microcontroller board instead of designing a board from scratch and creating an interface for coding into it would be optimal timewise. That being said there is a wide array of microcontroller boards already in the market. For this purpose we will be taking into account some parameters: System requirements, Number of I/O pins, Internet Connectivity, Community Support, and Price.

For the parameters mentioned above, internet connectivity is not negotiable, so we can start off from there. Having that as a limiting factor, we are left with a few ESP32 development boards, a few ESP8266 boards[29], Arduino Nano 33 IoT, Raspberry Pi, and BeagleBone. Then we can remove the ESP8266 boards from the list due to the limited number of I/O pins, despite it being the most economical choice. In regards to community support the best choice would be Arduino Nano 33 IoT and from the remaining boards it is in the mid-range for price and has the most convenient size for a slick device design, which is why we will be going with the Arduino Nano 33 IoT[30]. Now that we have the microcontroller board that will be handling with the interpreting of the infrared sensor signals selected, we will have to select the infrared sensor itself. The options available are a simple infrared sensor[31] connected to an Arduino Nano 33 IoT digital pin to detect the pulse modulations, or an infrared sensor module[32] that has a filtering feature to the design and a 3 pin design, the pins being VCC, Ground and Data.

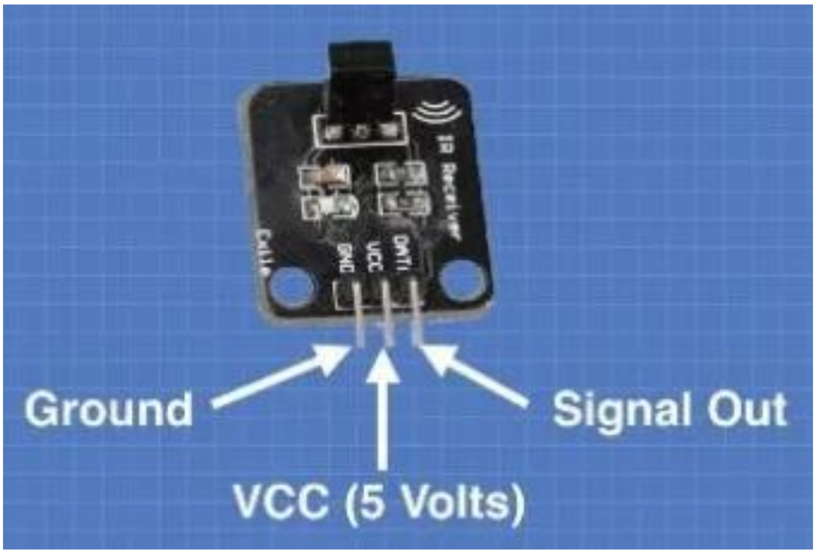

Figure 7. *IR Receiver Module*

The solution chosen for this paper will be the infrared sensor module as it provides a bit more stability to interferences and has a rating of up to 38 kHz.

CHAPTER 3

METHODOLOGY

3.2.1 Hardware Integration

As mentioned in the beginning of the paper the research for the device will be mainly done in the direction of the hardware with a bit of code implementation, this section will be dealing with the wiring up of the transmission and receiving parts of the device, and will also be covering some of the specifications of the devices chosen for the purposes of this paper.

Before continuing with the connections of the transmitting and receiving sides we will have an overview of the Arduino Nano 33 IoT, its pin layout and capabilities in relation to the paper and project at hand.

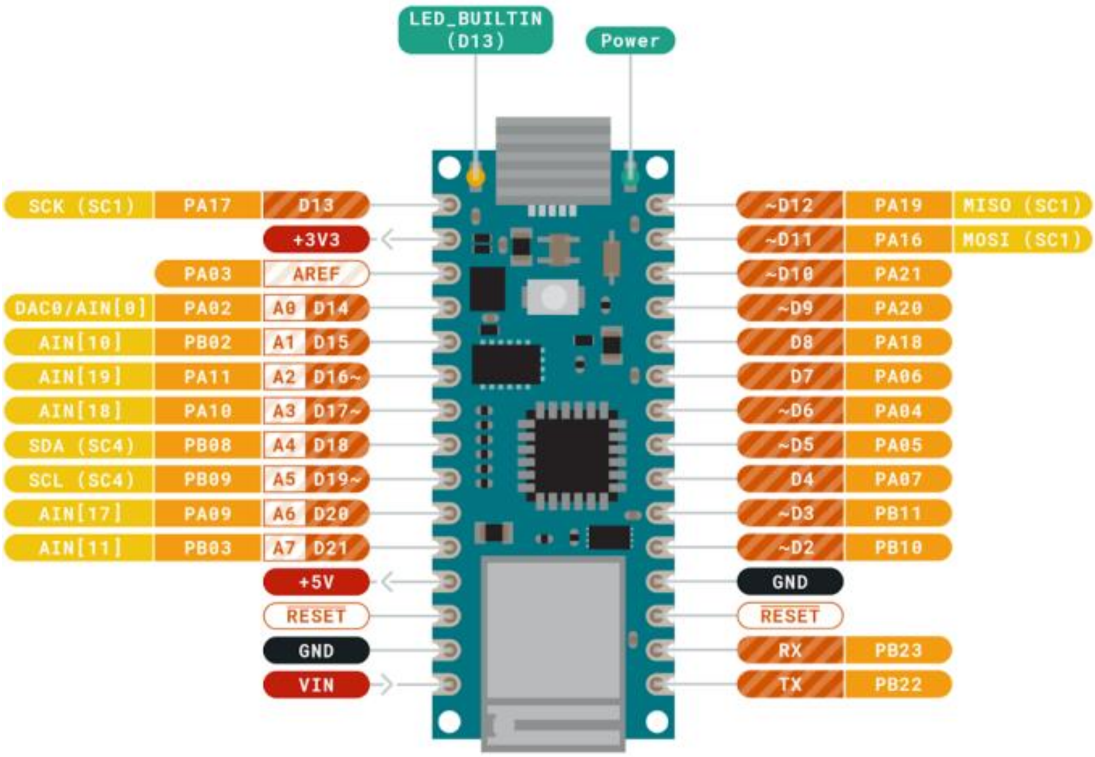

Figure 8. *Arduino Nano 33 IoT Pin Layout*

The Arduino Nano 33 IoT Microcontroller board is designed and built by the Arduino.cc meant for various embedded electronics projects and various commercial applications that can vary from direct sensory inputs and outputs to IoT applications.

The microcontroller on Arduino Nano 33 IoT is SAMD21G18A[33], which has plenty of great features varying from 23 I/O lines to having a high performance, low consumption, highly reliable non-volatile memory segments, SPI Serial Interfaces, Serial USART and many other great features that can be found on the datasheet[34] for further consultation.

The wifi module chip that is found on Arduino Nano 33 IoT is u-blox NINA-W102 Wi[35], which is a great, robust wifi module with normal data transmission speeds of 100kbps and fast transmission speeds of 400kbps.

Another great feature of Arduino Nano 33 Iot that makes it worth purchasing its cheaper competitor boards is the crypto-chip ECC608A[36], which ensures the cyber security of the device via a secure hardware key.

In regards to the pinout of Arduino Nano 33 IoT we have 12 Digital pins, D2 through D13, 8 Analog pins, A0 through A7, 1 3.3V pin, 1 5V pin, 2 ground pins, 2 reset pins and many of the I/O pins are multi-functional. The communication protocols Arduino Nano IoT can support are: Serial Protocol, SPI Protocol and I2C Protocol.

Serial Protocol being done through the D1 and D0 pins which act as RX and TX pins. The RX being Receive Data pin and the TX being Transmit Data pin.

SPI Protocol being done through the use of the pins D10, D11, D12 and D13, which are respectively, SS, MOSI, MISO, and SCK. SS is short for Slave Select and acts as a switch to select 1 of the slave devices. MOSI is short for Master Out Slave In, which is a pin used to send information to the devices while MISO, Master In Slave Out is used to receive information. SCK is the serial clock pin used to coordinate all the data transmission clocks.

I2C Protocol is done through the use of pins A4 and A5, which are respectively SDA and SCL. SDA is the data line while the SCL is the clock line.

Arduino Nano 33 IoT also has PWM, Pulse Width Modulation, pins which can be used for motor control. The PWM pins are pins D3, D5, D6, D9, D10, and D11.

The memory of Arduino Nano 33 IoT is divided into 3 compartments:

- Flash Memory – used for storing code, 32KB

- SRAM Memory – Cache memory, 2KB

- EEPROM – Off state memory, 1KB

Also the input voltage for Arduino Nano 33 IoT is 7V-12V, which is also 1 of the reasons it was chosen over the ESP32, and other ESP microcontroller boards with internet connectivity, it is more reliable and robust.

Finally the price of an Arduino Nano 33 IoT is 26-28.99 not including the shipping costs and for single prototyping purposes. Mass ordering for mass production price is 19.00$[37] over 150 pcs.

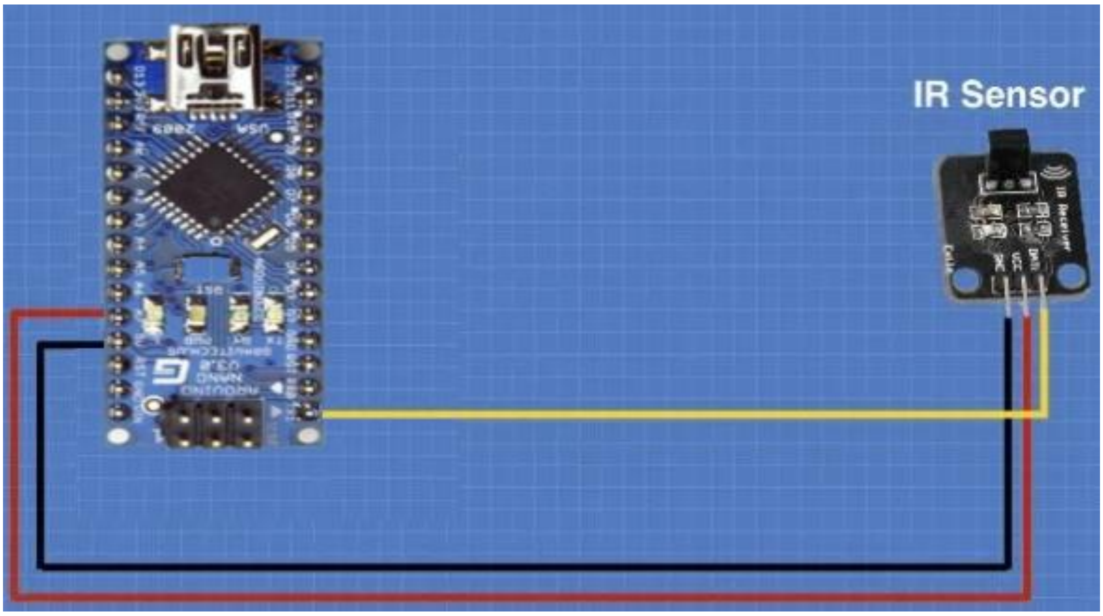

***Figure 9**. Arduino Nano 33 IoT IR Sensor Module Wiring*

Now that we have a better grasp of Arduino Nano 33 IoT has been created we can proceed with wiring the Infrared Sensor. As was displayed in Figure 7. the infrared sensor module has 3 pins, Ground, VCC and Data respectively going from left to right in the above shown picture. This module operates at 5V input and we connect the ground to 1 of the ground pins of the Arduino while connecting the VCC to 1 of the 5V outputs of the Arduino and connecting the Data pin of the infrared sensor to 1 of the 13 Digital pins of Arduino Nano 33 IoT. The pin selected for the Data pin is not crucial as long as it is correctly denoted in the code. The price of the Infrared Sensor Module ranges from 2.10$[38] to 3$[39] not including shipping costs and for a single prototyping purposes. Mass ordering for mass production price is 0.29$[40] over 3000 pcs.

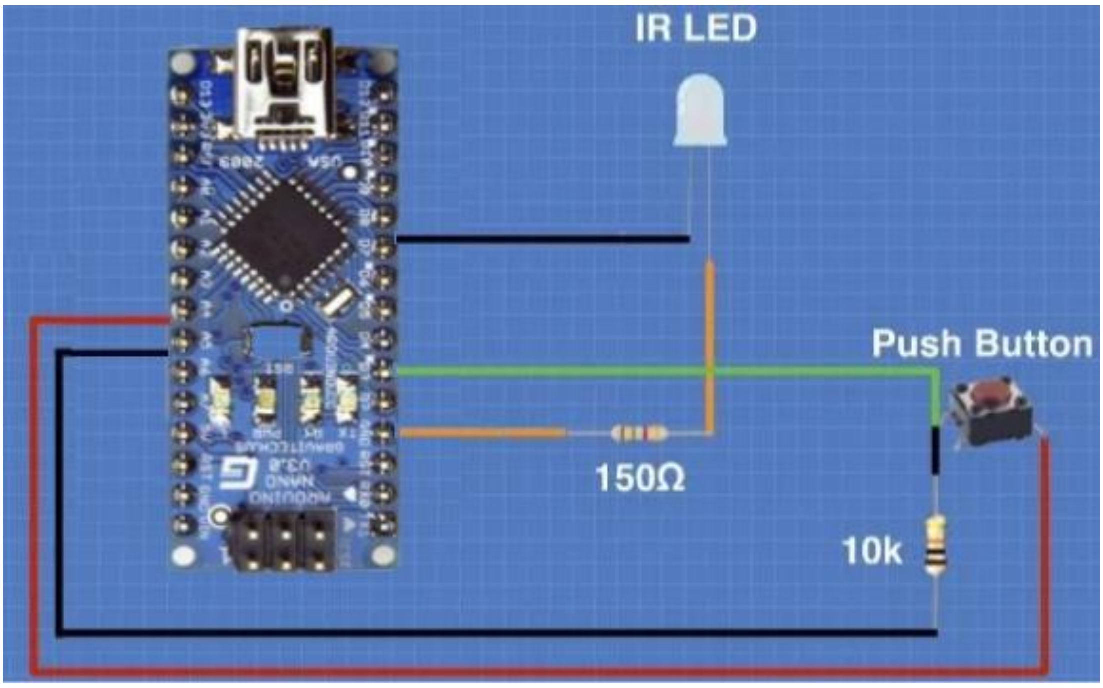

Figure 20. *Arduino Nano 33 IoT IR LED Wiring to act as a Transmitter*

This part is an attempt at making the device dependent on only 1 microcontroller and not to include timer IC. As shown above in the Figure 9. we have connected an infrared LED to the Arduino with a resistance beforehand so that the current going through the LED does is limited and does not burn it. Then we have a Push Button which will be used as the switch for the transmission of the unique signal towards the transmitter. The bonus of having the transmitter connected to the Microcontroller board is that its frequency can be controlled via the code, while the downside is that it will require wires to go from the Arduino to both the transmitter and the receiver part of the device, which is why I also considered the timer IC solution for the transmitter. The price for an IR LED ranges from 0.22$[41] to 0.43$[42] for prototyping purposes, not including shipping. Mass ordering for mass production price is 0.01$[43] with a minimum order of 1000 pcs. The price for a tactile push button ranges from 0.05$[44] to 0.30$[45] for prototyping purposes, not including shipping. Mass ordering for mass production price is 0.0125$[46] with a minimum order of 1000 pcs.

3.2.2 Connecting the Device to the Cloud

This section will be dealing with the codes that are necessary for the implementation of the project.

The code modules we will need are as following: infrared Transmission Module Code, infrared Receiving Module Code, internet Connection Code, and cloud/Application API.

Infrared Transmission Module Code:

~~~~~~

```
#include <IRremote.h>

// Define switch pin
const int switchPin = 5;

// Define a variable for the button state
int buttonState = 0;

// Create IR Send Object
IRsend irsend;

void setup()
{
  // Set Switch pin as Input
  pinMode(switchPin, INPUT);
}

void loop() {

  // Set button state depending upon switch position
```
~~~~~~

```
  buttonState = digitalRead(switchPin);

  // If button is pressed send power code command
   if (buttonState == HIGH) {
    irsend.sendNEC(0xFEA857, 32); // TV power code
  }

    // Add a small delay before repeating
    delay(200);

}
```

~ ~

The code shown above deals with the transmission of a specific value to the receiver. The way this is done is through the use of a custom made library called " IRremote.h"[47], which enables us to create an "IR send object" which we can further on define. We selected 5 in this case as the input of the switch to the Arduino. The next steps are having a digital read on the input pin and a condition with a message to be sent, which is also delayed a bit so as to not overload the receiver with data.

Infrared Receiving Module Code:

~ ~

```
// Include IR Remote Library by Ken Shirriff
#include <IRremote.h>

// Define sensor pin
const int RECV_PIN = 4;

// Define LED pin constants
const int redPin = 8;
const int yellowPin = 7;

// Define integer to remember toggle state
int togglestate = 0;

// Define IR Receiver and Results Objects
IRrecv irrecv(RECV_PIN);
decode_results results;


void setup(){
  // Enable the IR Receiver
```

```
  irrecv.enableIRIn();
  // Set LED pins as Outputs
  pinMode(redPin, OUTPUT);
  pinMode(yellowPin, OUTPUT);
}


void loop(){
    if (irrecv.decode(&results)){

        switch(results.value){
          case 0xFECA35: //Red Keypad Button
        // Turn on LED for 2 Seconds
        digitalWrite(redPin, HIGH);
        delay(2000);
        digitalWrite(redPin, LOW);
        break;

          case 0xFE8A75: //Yellow Keypad Button
        // Toggle LED On or Off
        if(togglestate==0){
        digitalWrite(yellowPin, HIGH);
        togglestate=1;
        }
        else {
        digitalWrite(yellowPin, LOW);
        togglestate=0;
        }
        break;

    }
    irrecv.resume();
  }

}
```

~ ~

The code shown above is for a scenario where the values transmitted to the receiver are custom values and for each of the two values that we receive we activate 1 of the 2 wired pins to an LED. Using the “IRremote.h” library we can even use premanufactured remoted to achieve the functionality we want. This library enables us to detect the manufacturer of the remote and then utilize that remote’s specific button codes to communicate.

Internet Connection Code[48]**:**

~~~~~~

```
#include <WiFiNINA.h>

//please enter your sensitive data in the Secret tab
char ssid[] = SECRET_SSID;                // your network SSID (name)
char pass[] = SECRET_PASS;                // your network password (use for WPA, or
use as key for WEP)
int status = WL_IDLE_STATUS;              // the Wi-Fi radio's status
int ledState = LOW;                       //ledState used to set the LED
unsigned long previousMillisInfo = 0;     //will store last time Wi-Fi information was
updated
unsigned long previousMillisLED = 0;      // will store the last time LED was updated
const int intervalInfo = 5000;            // interval at which to update the board
information

void setup() {
  //Initialize serial and wait for port to open:
  Serial.begin(9600);
  while (!Serial);

  // set the LED as output
  pinMode(LED_BUILTIN, OUTPUT);

  // attempt to connect to Wi-Fi network:
  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to network: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network:
    status = WiFi.begin(ssid, pass);

    // wait 10 seconds for connection:
    delay(10000);
  }
  // you're connected now, so print out the data:
  Serial.println("You're connected to the network");
  Serial.println("---------------------------------------");
}
void loop() {
  unsigned long currentMillisInfo = millis();

  // check if the time after the last update is bigger the interval
  if (currentMillisInfo - previousMillisInfo >= intervalInfo) {
    previousMillisInfo = currentMillisInfo;

    Serial.println("Board Information:");
    // print your board's IP address:
```
~~~~~~

```
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);

    // print your network's SSID:
    Serial.println();
    Serial.println("Network Information:");
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

    // print the received signal strength:
    long rssi = WiFi.RSSI();
    Serial.print("signal strength (RSSI):");
    Serial.println(rssi);
    Serial.println("---------------------------------------");
  }
    unsigned long currentMillisLED = millis();

  // measure the signal strength and convert it into a time interval
  int intervalLED = WiFi.RSSI() * -10;

  // check if the time after the last blink is bigger the interval
  if (currentMillisLED - previousMillisLED >= intervalLED) {
    previousMillisLED = currentMillisLED;

    // if the LED is off turn it on and vice-versa:
    if (ledState == LOW) {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }

    // set the LED with the ledState of the variable:
    digitalWrite(LED_BUILTIN, ledState);
  }
}
```

~ ~

For the code shown above we first need to set it up by adding the internet connection library for the wifi module that is on Arduino Nano 33 IoT, which is called “WiFiNINA.h”.

Once we have the library we start by adding in the piece of code that will deal with the credentials of the network the device will be connected to.

The next part is to start the connection and to ensure that the code returns warnings and signals in regards to its connectivity state.

The next section is for the code to print in the serial monitor the wifi credentials and the IP address of it.

In regards to the API integration, 2 options were considered, these option being: Cloud API, and Application API.

In regards to the Cloud API integration it was further divided into an already existing Cloud structure which is innate to Arduino boards, the Arduino IoT Cloud[49][50] or a custom made Cloud.

When comparing the two mentioned above we will be deciding based on 3 factors: Speed to market, Initial Cost, and Long term Cost.

The speed to market is definitely faster with Arduino IoT Cloud, as it is an already refined platform and the learning curve is limited to its functionality, which when combined with the extensive Arduino Community[51] can smooth the learning curve and start to make the product profitable sooner.

The initial costs are once again more beneficial for the Arduino IoT Cloud than the custom made Cloud[52] as the Arduino IoT Cloud functions on a subscription basis pricing with subscriptions ranging from 2.99$ to 23.99$[53], with the 23.99$ subscription being used for 100 devices only. Which increases the complexity of the implementation as some of the devices would have to be connected to one another and some to the cloud directly to increase the number of users but is still a very small initial cost compared to the full development of a Cloud architecture[54].

The Long term Cost I believe is higher with the Arduino IoT Cloud as when the code is done there will still be internal maintenance for both, but with the Arduino IoT Cloud we would be paying for the subscription too.

Moving to the other option for the API possibility which is with a custom Application. The option that I found to be relatively intuitive for an application API to Microcontroller board device was the MIT AppInventor[55], which had a blocks and charts approach to programming. Once again relatively smooth learning curve, very affordable, but not practical in the long run when the device will be used for large scale events.

Also another mechanism that I found to be of interest to the topic was Arduino Rest API, which is used to communicate data between the Arduino and external systems. This mechanism enables remote control of the Arduino, be it reading values, writing values or over-the-air sketch updating. The only functionality we are currently interested in are the remote value reading capabilities and the OTA sketch updates.

Sample Arduino Rest API Code[56]**:**

~ ~

```
// Create aREST instance
aREST rest = aREST();
// NeoPixel Init
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB +
NEO_KHZ800);
void setup() {
  Serial.begin(115200);
  // Register RGB function
  rest.function("rgb", setPixelColor);
  Serial.println("Try DHCP...");
  if (Ethernet.begin(macAdd) == 0) {
    Serial.println("DHCP FAIL...Static IP");
    Ethernet.begin(macAdd , ip, myDns, myGateway) ;
  }
  server.begin();
  Serial.print("server IP: ");
  Serial.println(Ethernet.localIP());
  pixels.begin();
  Serial.println("Setup complete.\n");
}
void loop() {
  // listen for incoming clients
  EthernetClient client = server.available();
  rest.handle(client);
  wdt_reset();
}
int setPixelColor(String hexColor) {
  hexColor="0x" + hexColor;
  Serial.println("Hex color " + hexColor);
  long n = strtol( &hexColor[0], NULL, 16);
  Serial.println("N :" + String(n));
  long r = n << 16;
  long g = n << 8 && 0xFF;
  long b = n && 0xFF;
  // set single pixel color
  return 1;
}
```

~ ~

CHAPTER 4

RESULTS AND DISCUSSIONS

So as a result we can say that the device's general hardware functionality has been reached. The infield testing of the device remains to be done in further follow up researches and the software to enable the functionality via the code is present.

Speaking of the hardware functionality and implementation a PCB design was done for each of the functional parts of the device, namely speaking: the transmitter via Arduino, the receiver via Arduino and the transmitter via NE555 timer IC.

The PCB design for the Arduino transmitter is a compact design with a length of 6cm and width of 2.5 cm. The components of it are the microcontroller, tactile pushbutton, resistor and an infra-red LED.

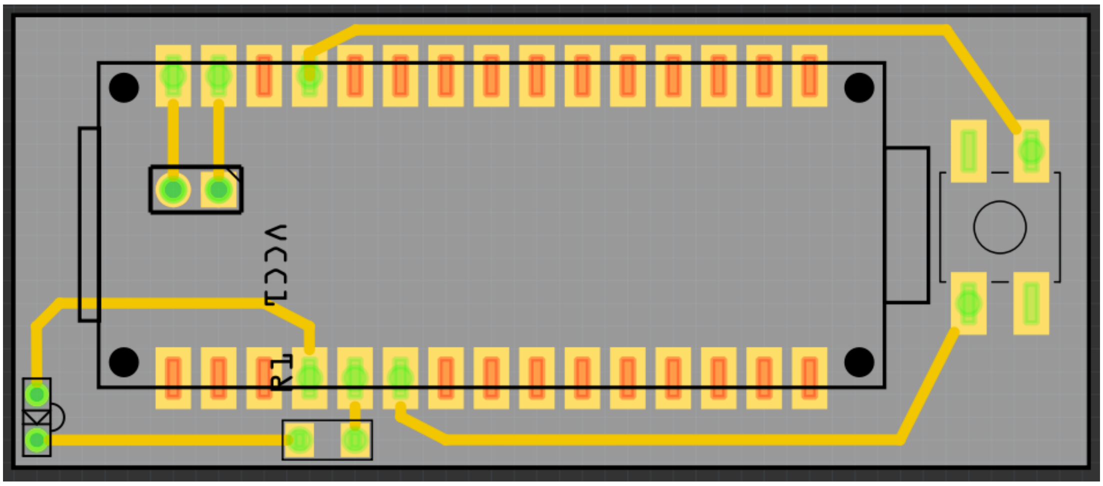

Figure 31. *Arduino Nano 33 IoT IR LED PCB*

The PCB design for the Arduino receiver is a compact design with a length of 7cm and width of 2.5 cm.

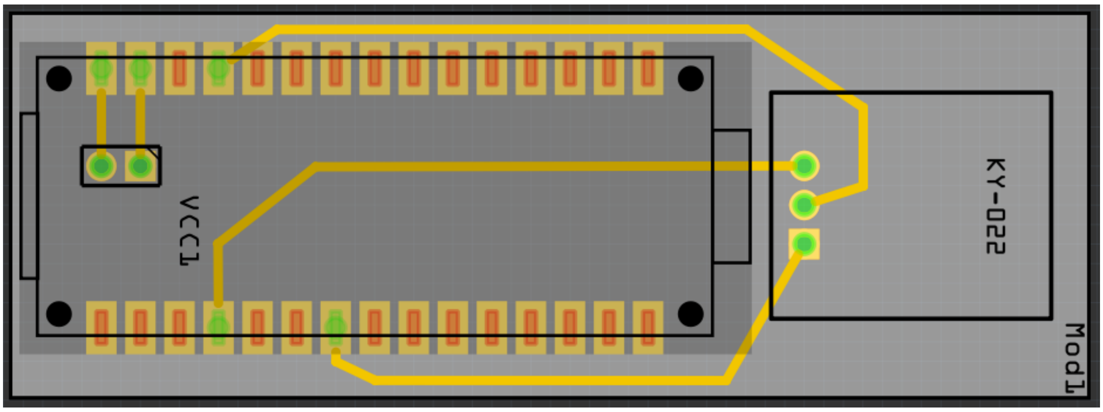

Figure 42. *Arduino Nano 33 IoT Receiver PCB*

The PCB design for the NE555 timer circuit transmitter is a compact design with a length of 6cm and width of 5 cm.

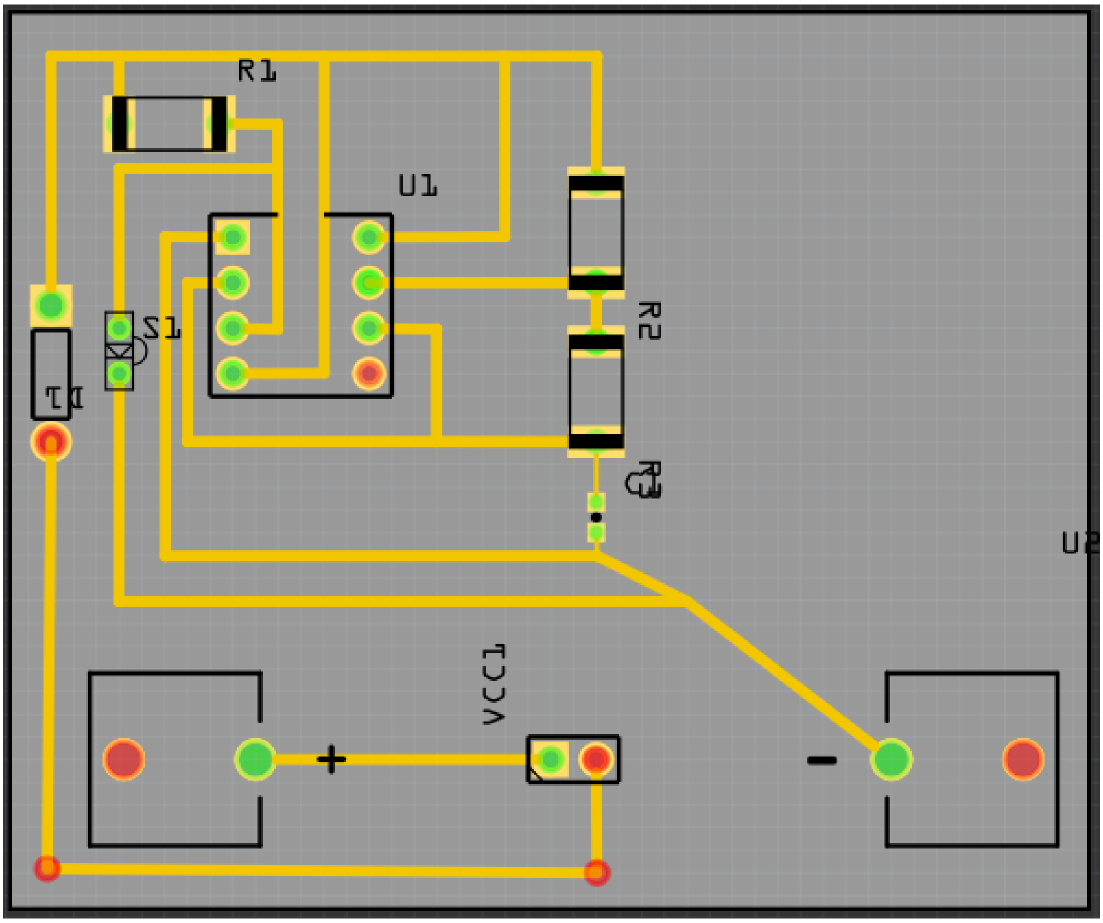

Figure 53. *NE555 timer IC Transmitter PCB*

The next section of the results is dedicated to the testing done to the 555 timer circuit for different values of resistance and capacitance in relation to the response frequency. The schematic of the circuit can be seen below:

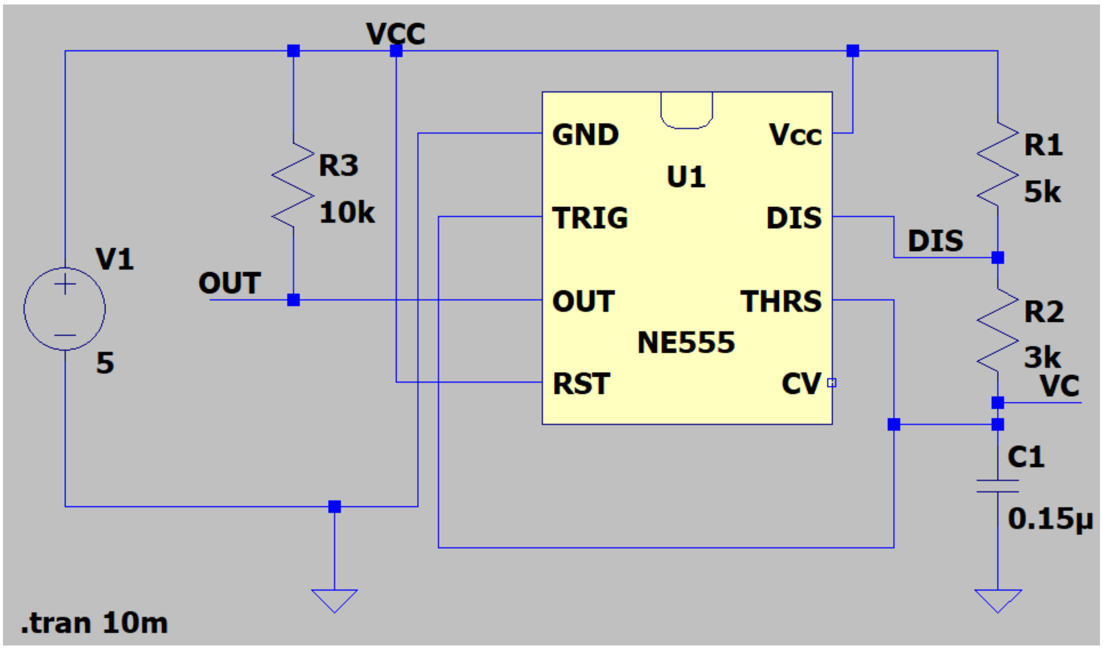

Figure 64. *NE555 timer IC circuit for frequency response testing*

From the components shown above the values of R3 and V1 do not have an effect on the frequency response, so we will be investigating the frequency response based on the changes we make to the values of R1, R2 and C1. Before we explain how the investigation was set up we can see below the waveforms in the different parts of the circuit, respectively,

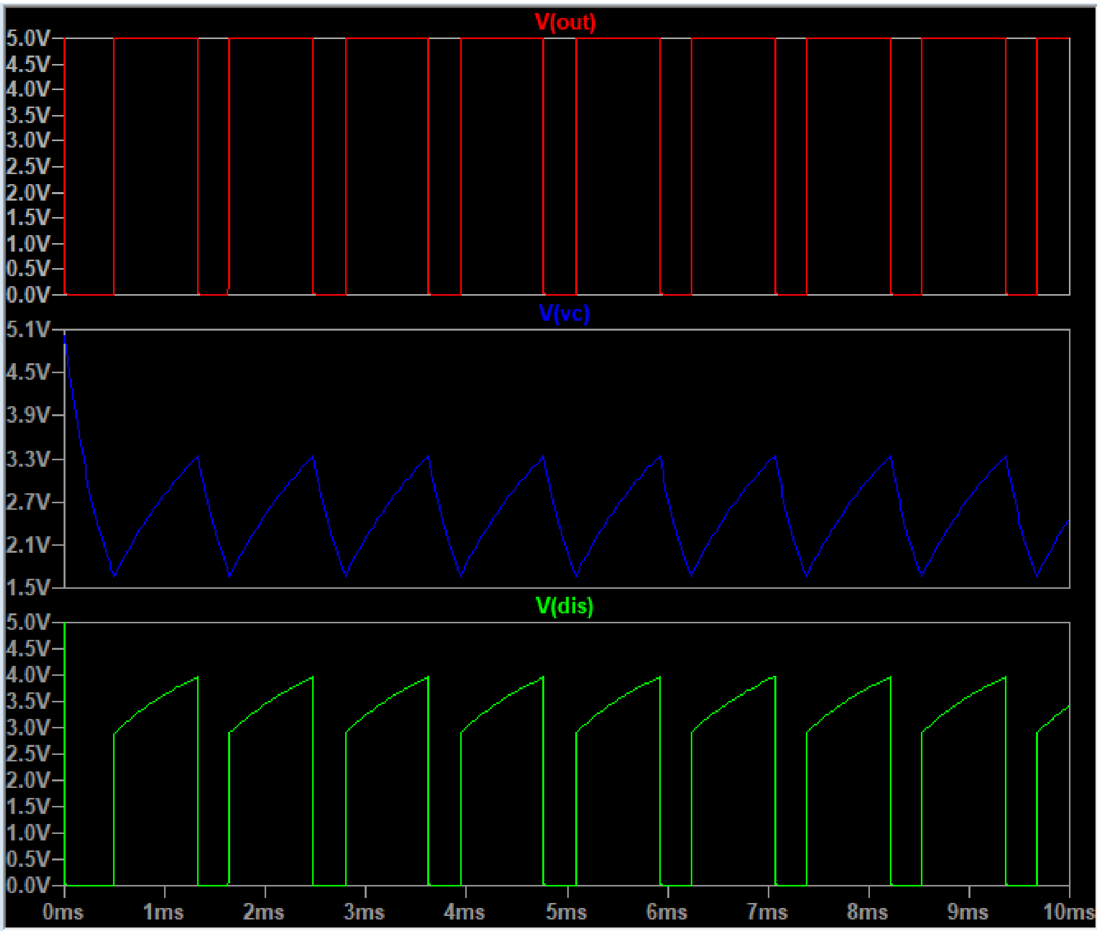

Figure 75. *NE555 timer IC waveforms in different points of the circuit*

Next, we explain how the experimenting was set up. We have a total of 5 tests done for the frequency response of the 555 timer circuit, name speaking: Testing the response when only C1 is changing, testing the response when only R1 is changing, testing the response when only R2 is changing, testing the response when R1 & R2 are changing, and finally testing the response when R1, R2 & C1 are changing.

In the first test where only the values of C1 were changed, there was a proportional change in the waveform of the response, so the ratio of time between a 0 and 1 in a full wave was maintained. Another observation achieved is that the response of the frequency is inversely proportional to the value of the capacitor, so the higher the value of the capacitor the lower the frequency. This is to be expected as the higher the value of the capacitor the longer it will take for it to get charged to the "threshold level".

Table 1. Lower Capacitance values

Capacitance (µF)	Frequency response (Hz)
0.0025	49474
0.003	39801
0.0035	34165
0.004	29949
0.0045	26677
0.005	24033
0.0055	21867
0.006	20034
0.0065	18513
0.007	17203
0.0075	16051
0.008	15074
0.0085	14166
0.009	13408
0.0095	12692
0.01	11223
0.05	2418

In the first set of values for the capacitance which are the values 0.05 µF and lower we can see that the frequency response is quiet, going as high as 49474 Hz, and this is achieved while keeping the resistance values all the same 5k for R1 and 3k for R2.

Table 2. Higher Capacitance values

Capacitance (µF)	Frequency response (Hz)
0.1	1285.714
0.15	879.765
0.2	636.943
0.25	515.198
0.3	427.35
0.35	365.344
0.4	318.471
0.45	281.69
0.5	253.55
0.55	234.542
0.6	214.638
0.65	197.759
0.7	183.276

0.75	170.977
0.8	160.385
0.85	150.505
0.9	142.103
0.95	136.026
1	129.129
1.05	122.963
1.1	117.196
1.15	112.085
1.2	107.342
1.25	103.04
1.3	98.912
1.35	95.178
1.4	91.781
1.45	88.456
1.5	85.516
1.55	82.764
1.6	80.184
1.65	78.477
1.7	76.123
1.75	73.906
1.8	71.843
1.85	69.829
1.9	68.013
1.95	66.289
2	64.565
2.1	61.482
2.2	58.598
2.3	56.045
2.4	53.671
2.5	51.522
2.6	49.459
2.7	47.589
2.8	45.893
2.9	44.226
3	42.758
3.1	41.382
3.2	40.094
3.3	39.238
3.4	38.062
3.5	36.952
3.6	35.921
3.7	34.914

3.8	34.006
3.9	33.144
4	32.282

In the second set of values for the capacitance which are the values above 0.05 μF we can see that the frequency response is starts dropping off rather exponentially even though the values of capacitance are increasing linearly, which is why I decided to split the values so they can be presented better in a chart.

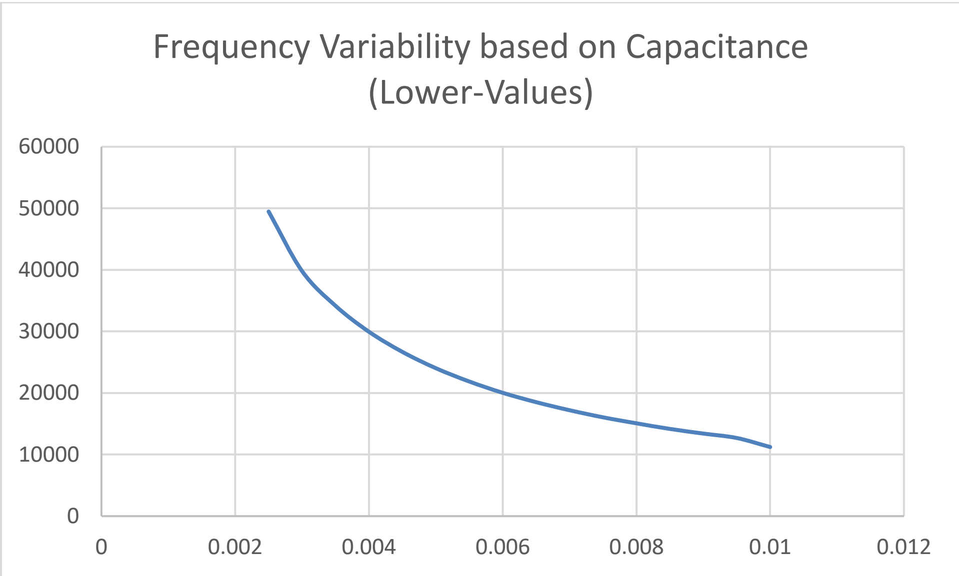

Figure 86. *Frequency response as a function of the lower values of capacitance*

As we can see, the values take a parabolic line, with the values that are smaller producing an increasingly higher frequency response.

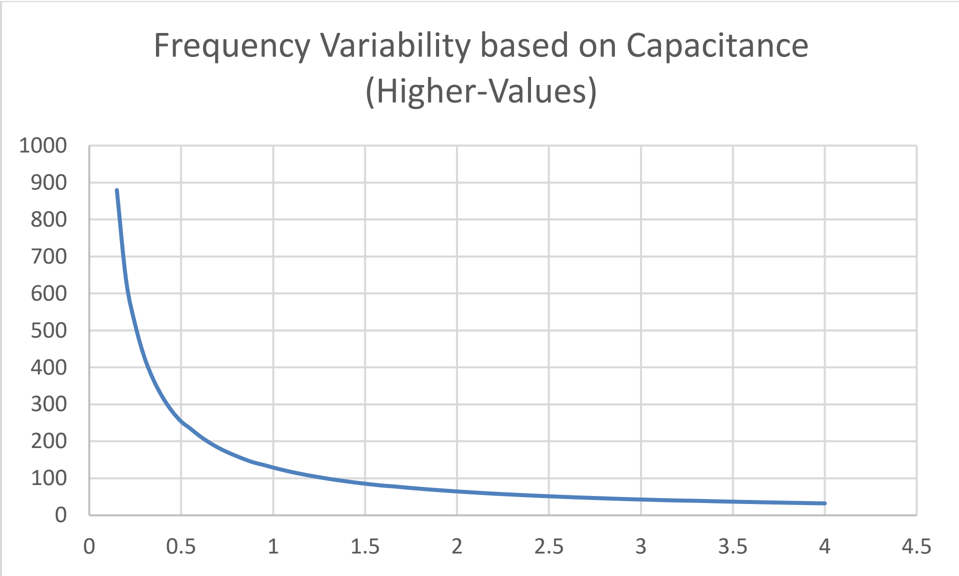

Figure 97. *Frequency response as a function of the higher values of capacitance*

As we can see, the values take a parabolic line, with the values that are smaller producing an increasingly higher frequency response just like the smaller values of capacitance, shown above.

Our next test is seeing the frequency response when only the value of R1 is changed. Which based on the observations done affects the width of the time when the signal is "high" in a proportional manner to the increase of the resistance. So the higher R1 the wider the "high" time of each wave. The "low" part of the signal is not effected, so based on these knowledges we can deduce that the frequency is inversely proportional to R1 values. The higher the R1 values become, the wider the "high" state of the signal and the lower the frequency due to that.

Table 3. R1 Frequency Response

Cap (µF)	R3 (kΩ)	R1/R2	R1 (Ω)	R2 (Ω)	R1+R2 (Ω)	Freq (Hz)
0.15	10	0.013	13	1000	1013	1750
0.15	10	0.0169	17	1000	1016.9	2678
0.15	10	0.022815	23	1000	1022.815	3258
0.15	10	0.03080025	31	1000	1030.80025	3551
0.15	10	0.04158034	42	1000	1041.58034	3796
0.15	10	0.05613346	56	1000	1056.13346	3957
0.15	10	0.07578017	76	1000	1075.78017	4080
0.15	10	0.10230322	102	1000	1102.30322	4128

0.15	10	0.13810935	138	1000	1138.10935	4134
0.15	10	0.18644762	186	1000	1186.44762	4100
0.15	10	0.25170429	252	1000	1251.70429	4020
0.15	10	0.33980079	340	1000	1339.80079	3885
0.15	10	0.45873107	459	1000	1458.73107	3717
0.15	10	0.61928695	619	1000	1619.28695	3512
0.15	10	0.83603738	836	1000	1836.03738	3249
0.15	10	1.12865046	1129	1000	2128.65046	2939
0.15	10	1.52367812	1524	1000	2523.67812	2623
0.15	10	2.05696547	2057	1000	3056.96547	2278
0.15	10	2.77690338	2777	1000	3776.90338	1926
0.15	10	3.74881956	3749	1000	4748.81956	1613
0.15	10	5.06090641	5061	1000	6060.90641	1303
0.15	10	6.83222365	6832	1000	7832.22365	1052
0.15	10	9.22350193	9224	1000	10223.5019	844
0.15	10	12.4517276	12452	1000	13451.7276	655
0.15	10	16.8098323	16810	1000	17809.8323	502
0.15	10	22.6932736	22693	1000	23693.2736	388
0.15	10	30.6359193	30636	1000	31635.9193	293
0.15	10	41.3584911	41358	1000	42358.4911	221
0.15	10	55.8339629	55834	1000	56833.9629	166
0.15	10	75.37585	75376	1000	76375.85	124
0.15	10	101.757397	101757	1000	102757.397	92
0.15	10	137.372487	137372	1000	138372.487	69
0.15	10	185.452857	185453	1000	186452.857	60
0.15	10	250.361357	250361	1000	251361.357	38
0.15	10	337.987832	337988	1000	338987.832	28
0.15	10	456.283573	456284	1000	457283.573	21
0.15	10	615.982823	615983	1000	616982.823	15.5
0.15	10	831.576811	831577	1000	832576.811	11.5
0.15	10	1122.6287	1122629	1000	1123628.7	8.5
0.15	10	1515.54874	1515549	1000	1516548.74	6.3
0.15	10	2045.9908	2045991	1000	2046990.8	4.6
0.15	10	2762.08758	2762088	1000	2763087.58	3.4

Shown above are the values of the frequency response based on the variability of R1. Below they are represented into 2 graphs split into lower values and higher values of R1.

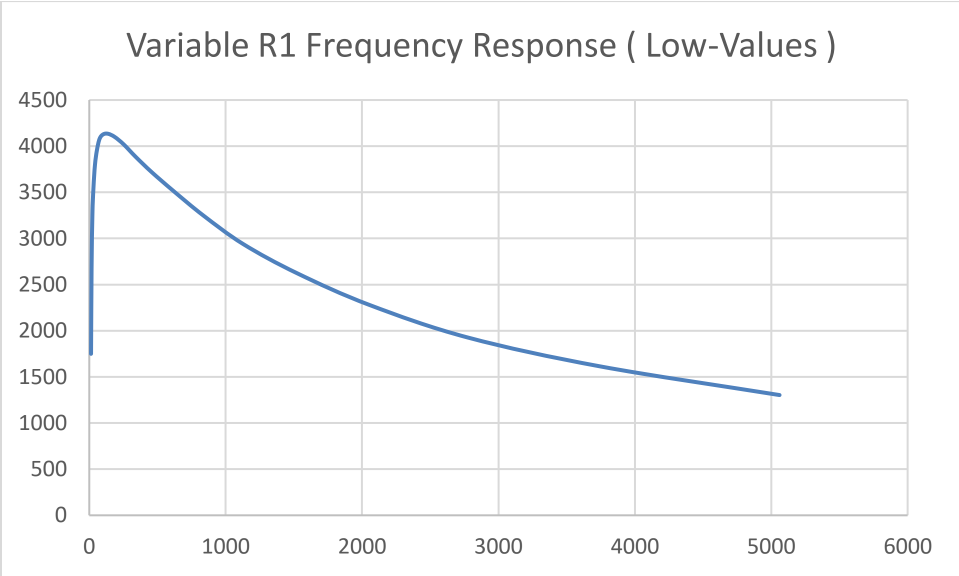

Figure 108. *Frequency response as a function of the lower values of R1*

As we can see above there is an absolute maximum at the frequency response of 4100 and resistance values 1100, meaning that the values of resistance below 1100 produce a lower frequency response than values at 1100, which is explained by the fact that the "low" state is not changed by the change of R1, so lowering the value of R1 has diminishing returns.

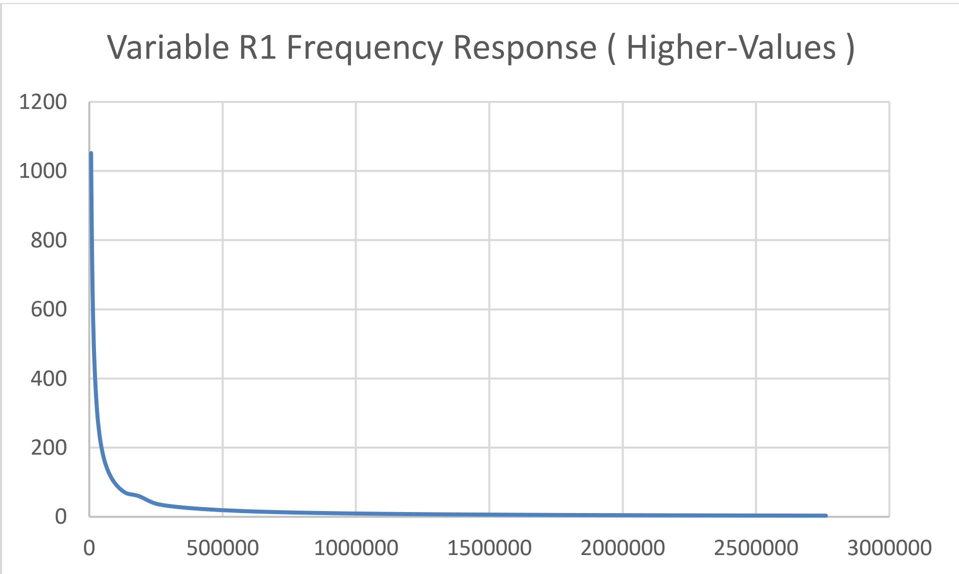

Figure 119. *Frequency response as a function of the higher values of R1*

As we can see above on the graph for the higher values of R1 we have an exponential growth of frequency response the lower the R1 values get. The values of R1+R2 and R1/R2 will be used later in a common chart of all the responses.

Our next test is seeing the frequency response when R1 is constant at 1000 ohms and the value of R2 varies. Based on the observations done, this causes the "low" state of the signal in a single wave to become proportionally bigger and causes the "high" state of the signal in a single wave to become proportionally bigger based on the voltage divider rule between R1 and R2.

Table 4. R2 Frequency Response

Cap (μF)	R3 (kΩ)	R1/R2	R1 (Ω)	R2 (Ω)	R1+R2 (Ω)	Freq (Hz)
0.15	10	1000	1000	1	1001	9184
0.15	10	666.666667	1000	1.5	1001.5	9184
0.15	10	444.444444	1000	2	1002.25	9184
0.15	10	296.296296	1000	3	1003.375	9184
0.15	10	197.530864	1000	5	1005.0625	9184
0.15	10	131.687243	1000	8	1007.59375	9184
0.15	10	87.7914952	1000	11	1011.390625	9184
0.15	10	58.5276635	1000	17	1017.085938	9100
0.15	10	39.0184423	1000	26	1025.628906	8959
0.15	10	26.0122949	1000	38	1038.443359	8772
0.15	10	17.3415299	1000	58	1057.665039	8466
0.15	10	11.5610199	1000	86	1086.497559	8056
0.15	10	7.70734663	1000	130	1129.746338	7487
0.15	10	5.50524759	1000	182	1181.644873	6889
0.15	10	3.93231971	1000	254	1254.302822	6211
0.15	10	2.80879979	1000	356	1356.023951	5429
0.15	10	2.00628557	1000	498	1498.433532	4603
0.15	10	1.43306112	1000	698	1697.806944	3759
0.15	10	1.02361508	1000	977	1976.929722	3030
0.15	10	0.73115363	1000	1368	2367.701611	2301
0.15	10	0.52225259	1000	1915	2914.782255	1833
0.15	10	0.37303757	1000	2681	3680.695157	1335
0.15	10	0.26645541	1000	3753	4752.97322	1066
0.15	10	0.19032529	1000	5254	6254.162508	760
0.15	10	0.13594664	1000	7356	8355.827511	535
0.15	10	0.09710474	1000	10298	11298.15852	345
0.15	10	0.06936053	1000	14417	15417.42192	249
0.15	10	0.04954323	1000	20184	21184.39069	215
0.15	10	0.03538802	1000	28258	29258.14697	151

0.15	10	0.02527716	1000	39561	40561.40575	104
0.15	10	0.01805511	1000	55386	56385.96806	66
0.15	10	0.01289651	1000	77540	78540.35528	53
0.15	10	0.00921179	1000	108556	109556.4974	41
0.15	10	0.00657985	1000	151979	152979.0963	27
0.15	10	0.00469989	1000	212771	213770.7349	17
0.15	10	0.00335707	1000	297879	298879.0288	12.4
0.15	10	0.00239791	1000	417031	418030.6404	10.8
0.15	10	0.00171279	1000	583843	584842.8965	7.4
0.15	10	0.00122342	1000	817380	818380.0551	5.1
0.15	10	0.00087387	1000	1144332	1145332.077	3.2
0.15	10	0.00062419	1000	1602065	1603064.908	2.3
0.15	10	0.00044585	1000	2242891	2243890.871	1.65

As we can see above, due to the values of R2 changing both the “high” and the “low” part of the wave signal the range of frequency response for R2 variable values is much bigger than the variable R1. Not only is the range of frequency bigger but the frequency values extend to below and above R1 too, with the lowest frequency being 1.65 Hz and the highest being 9184 Hz, as compared to R1’s lowest of 3.4 Hz and highest of 4134 Hz. This is attributed to the effect R2 has on both the “low” and “high” part of the wave.

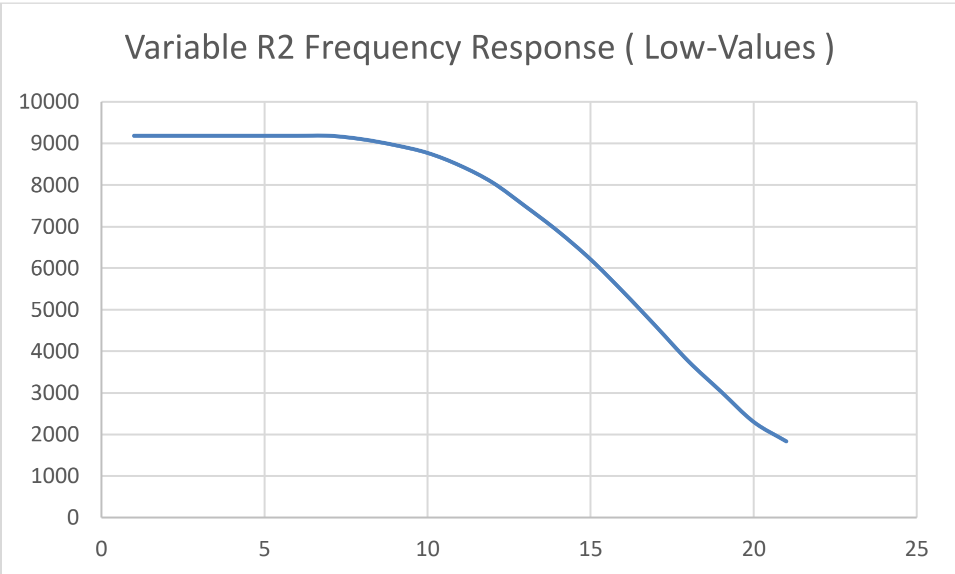

Figure 20. *Frequency response as a function of the lower values of R2*

As we can see above in the graph of the lower values of R2 we have a flat response of 9184 Hz for R2 values that are 11 ohm or lower and then we have an exponential decrease of the frequency.

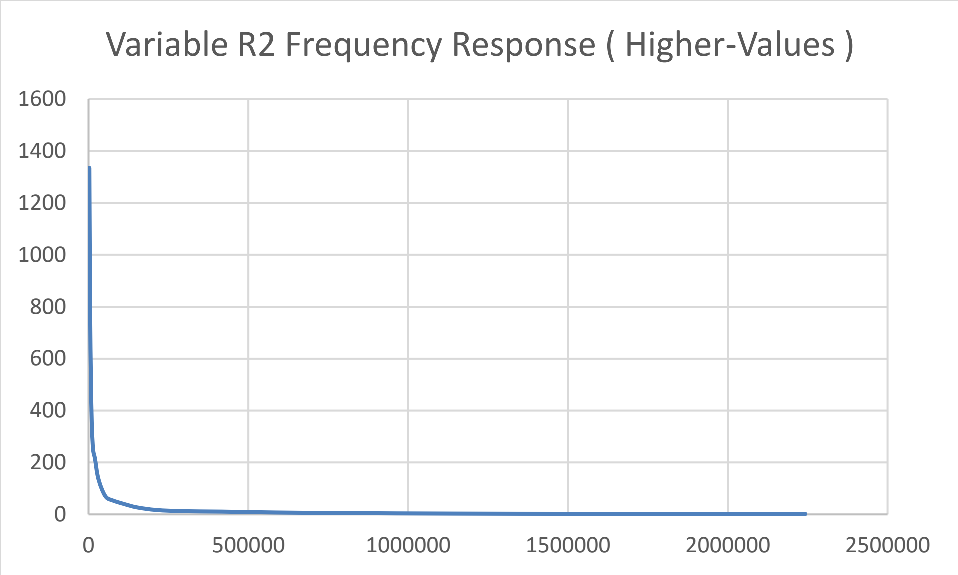

Figure 212. *Frequency response as a function of the higher values of R2*

As we can see on the higher values of R2 there is a steep fall on the frequency response all the way to 66 Hz, and the other higher values of R2 have diminishing effects on the frequency response the higher the values go from 70000 Ω.

The next frequency response test is done with the changing values of both R1 and R2 simultaneously while keeping the values of C1 constant. During this test it was observed that the frequency response is much wider, ranging from 246862 Hz to 1.21 Hz, which is way wider of a frequency range than either R1 or R2's frequency responses.

Table 5. R1 and R2 Frequency Response

Cap (µF)	R3 (kΩ)	R1 (Ω)	R2 (Ω)	R1+R2 (Ω)	R1/R2	Freq (Hz)
0.15	10	13	1	14	13	246862
0.15	10	17	2	18.4	11.2666667	253218
0.15	10	23	2	25.065	10.14	236001
0.15	10	31	3	34.17525	9.126	192390
0.15	10	42	5	46.6428375	8.2134	147710

0.15	10	56	8	63.7272056	7.39206	111657
0.15	10	76	11	87.1707901	6.652854	85179
0.15	10	102	17	119.38916	5.9875686	62722
0.15	10	138	26	163.738257	5.38881174	45188
0.15	10	186	38	224.890983	4.84993057	34786
0.15	10	252	58	309.369331	4.36493751	25032
0.15	10	340	86	426.298353	3.92844376	18093
0.15	10	459	130	588.47741	3.53559938	12885
0.15	10	619	182	800.93182	3.40932798	9368
0.15	10	836	254	1090.3402	3.28756626	6820
0.15	10	1129	356	1484.67441	3.17015318	4901
0.15	10	1524	498	2022.11165	3.05693343	3731
0.15	10	2057	698	2754.77241	2.94775723	2706
0.15	10	2777	977	3753.8331	2.84248019	1944
0.15	10	3749	1368	5116.52117	2.74096304	1393
0.15	10	5061	1915	6975.68866	2.6430715	1015
0.15	10	6832	2681	9512.91881	2.54867609	772
0.15	10	9224	3753	12976.4752	2.45765194	559
0.15	10	12452	5254	17705.8901	2.36987866	404
0.15	10	16810	7356	24165.6598	2.28524014	302
0.15	10	22693	10298	32991.4321	2.20362442	219
0.15	10	30636	14417	45053.3412	2.12492355	158
0.15	10	41358	20184	61542.8818	2.04903342	114
0.15	10	55834	28258	84092.1099	1.97585365	82
0.15	10	75376	39561	114937.256	1.90528745	61
0.15	10	101757	55386	157143.366	1.83724147	44
0.15	10	137372	77540	214912.842	1.7716257	32
0.15	10	185453	108556	294009.354	1.70835336	23
0.15	10	250361	151979	402340.453	1.64734074	16.4
0.15	10	337988	212771	550758.567	1.58850714	12.2
0.15	10	456284	297879	754162.602	1.53177474	8.8
0.15	10	615983	417031	1033013.46	1.4770685	6.3
0.15	10	831577	583843	1415419.71	1.42431606	4.7
0.15	10	1122629	817380	1940008.75	1.37344762	3.37
0.15	10	1515549	1144332	2659880.82	1.32439592	2.41
0.15	10	2045991	1602065	3648055.71	1.27709607	1.72
0.15	10	2762088	2242891	5004978.45	1.2314855	1.21

As we can see above the frequency response to both R1 and R2 being variable isn't an additive result of the separate R1 and R2 frequency responses, nor is it a multiplicative result. The resultant max frequency response is approximately 60 times higher than R1's max frequency response and 30 times higher than R2's max frequency response. This can be attributed to R1 and R2 being mutually dependent in the frequency response they create when the resistance values change and to the capacitor limiting off the higher range.

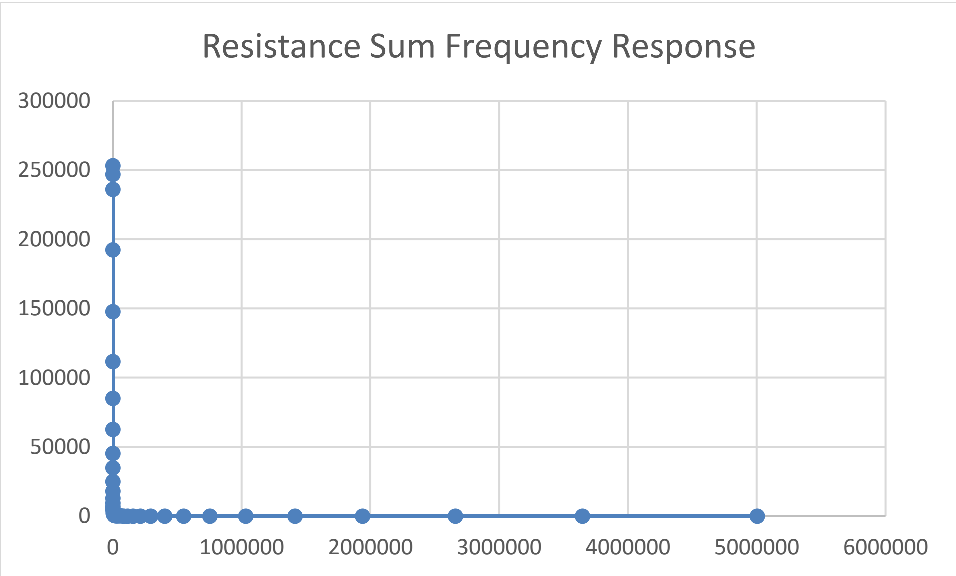

Figure 22. *Frequency response as a function of the sum of R1 and R2*

As we can see above the frequency response became even more exponential to the lower values even though the resistance value range has almost double, since we added R1 and R2.

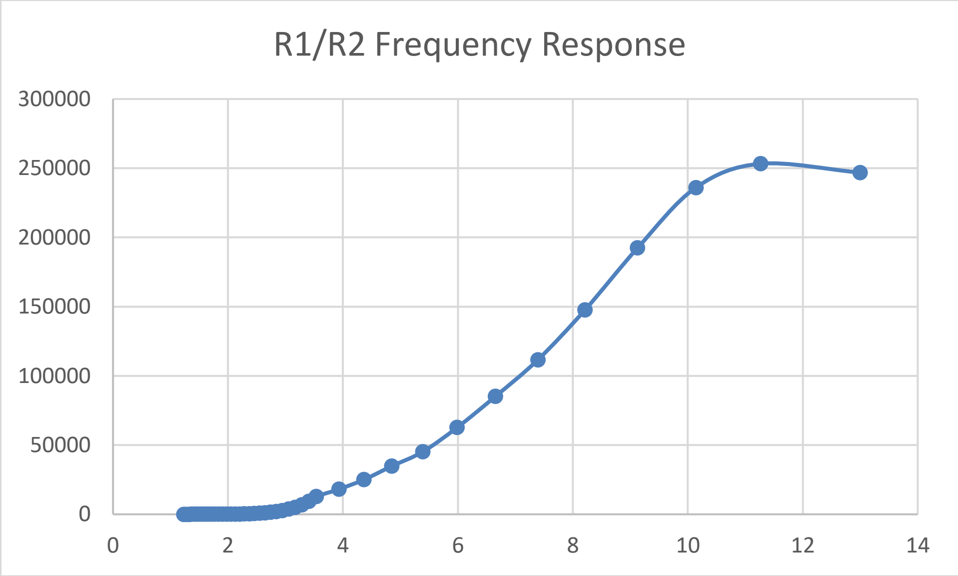

Figure 23. *Frequency response as a function of the ratio of R1 and R2*

From the data set above we can see that there is little correlation between the ratio of R1 to R2 due to the initial dataset being skewed towards R1 with the initial value of 13 and R2 with an initial value of 1. There is no correlation between the ration R1 and R2 to the frequency response.

Our next and final frequency response test is having R1, R2 and C1 all be variable from the lower values to the highest values. This test produced the highest frequency response range and value so far.

Table 6. R1, R2, and C1 Frequency Response

R1 (Ω)	R2 (Ω)	Cap (µF)	Freq (Hz)
13	1	0.0025	4243281
17	2	0.003	3441992
23	2	0.004	2932551
31	3	0.005	2464066
42	5	0.006	1930916
56	8	0.007	1471516
76	11	0.008	1097695
102	17	0.009	814249
138	26	0.01	572023
186	38	0.05	101952

252	58	0.1	37481
340	86	0.15	18093
459	130	0.2	9784
619	182	0.3	4769
836	254	0.4	2636
1129	356	0.5	1532
1524	498	0.6	938
2057	698	0.7	588
3749	1368	0.8	275
6832	2681	0.9	130
12452	5254	1	62
22693	10298	1.5	21.9
55834	28258	2	6.33
185453	108556	2.6	1.342
456284	297879	3	0.438
1122629	817380	3.6	0.141
2762088	2242891	4	0.047

The results above are the highest frequency we have achieved so far in this circuit's test, but the values of frequency shown in red in the table are not available to use as we would be going out of the product's specifications of an absolute overclocked frequency of 2 MHz. So the results in red are only theoretical. Now that all the resistance variability is available we can do a total chart.

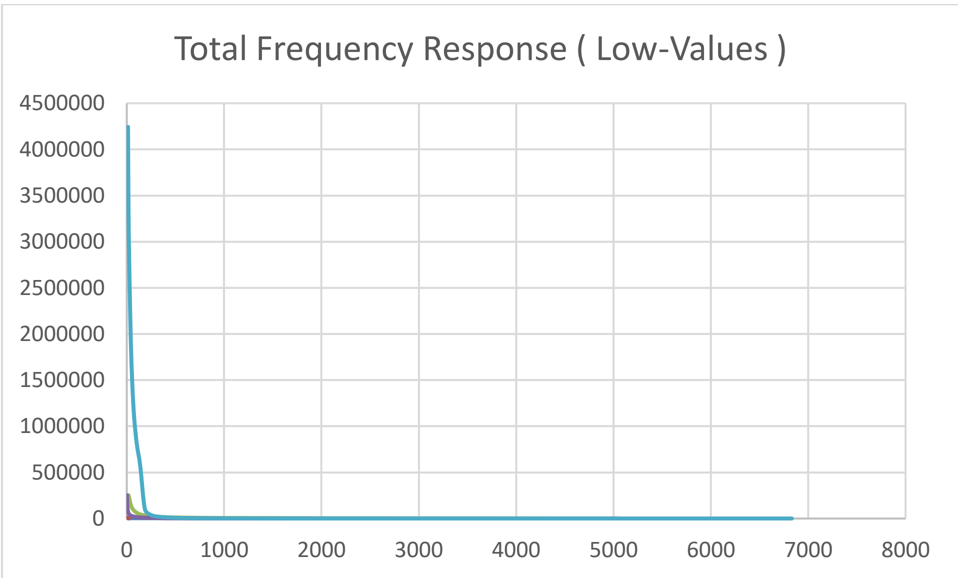

Figure 24. *Total variability frequency response as a function of the lower values*

As we can see above now that we have all the values at hand the chart constructed above contains all 5 resultants, which now clearly indicates just how big of a difference of frequency response we have for the R1, R2 and C1 changing as compared to the individual variables.

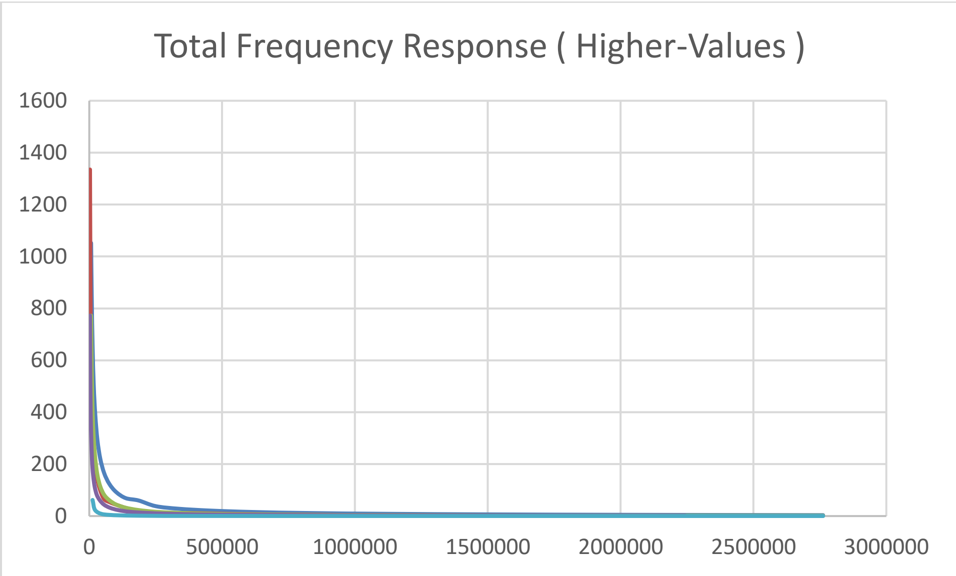

***Figure 25**. Total variability frequency response as a function of the higher values*

Here we have the total comparison in the higher values, where we can see that the change is still present but not as exponentially different as with the lower values.

So to sum the results up, the lower values of R1, R2 and C1 can have exponential effects on the circuit' frequency response, while the higher values are not as decisive. C1 affects the circuit's frequency the most, then R2 and finally R1. C1 is limited on how low it can go based on the values of R1 and R2, which is due to the charging.

The PCBs for each of the sections can be quite compact and functional too coming in small mass-producible format.

Speaking of the price of the device's components we can see in the table below:

Table 7. Prices of electronics components

Components	Lower-end Prototype Cost ($)	Upper-end Prototype Cost ($)	Mass Production Cost ($)
Arduino Nano 33 IoT	26	28.99	19
Infrared receiver module	2.10	3	0.29
Infrared LED	0.22	0.43	0.01
Tactile Push Button	0.05	0.30	0.0125
NE555 timer IC	0.20	0.40	0.128

Now we have a clear idea as to what the product's cost can be in regards to the electronics, which is at the lower end of prototyping 28.57$ + 2$ for the PCB service , so 30.57$. And at the upper end of prototyping 33.12$ + 2$ = 35.12$ and at mass production level 19.4405$ + 2$ = 21.4405$.

The price of the mechanical parts was not calculated.

CHAPTER 5

CONCLUSIONS

5.1 Conclusions

In this paper the general structure of a device to simplify and facilitate networking in alive setting is discussed. The functionality has been achieved in this theoretical cover of the device. We have the hardware, the method of communication and acknowledgement between the devices and the internet connectivity too. So the basis of the hardware level has been laid with the background still needing to be laid, which includes the Cloud structure and the needed system requirements for the Cloud, also the display functionality too.

5.2 Recommendations for future research

To further explore this device there are a few fine tunings that need to be done throughout the device idealization as it is to make it match the in-field challenges. During the paper I tried to make the modeling as considerate of physical issues that might come up during use, but other difficulties in use are expected to rise. Some of the expected difficulties are the aiming of the infrared transmitter, since it is not visible, and the fine tuning of the receiver size to reduce accidental profile triggering. Another thing that needs to be done is further exploration of the Cloud, the structure and the correct API interfacing properly designed to handle the traffic, which will need to be measured. And the final step would be the display of the profile received by the initial request maker. I believe a good solution would be an app to connect to your profile to the Cloud so you can have easy familiar access and interfacing through your mobile smartphone.

References

[1] M. E. Coles, C. L. Turk, R. G. Heimberg, and D. M. Fresco, "Effects of varying levels of anxiety within social situations: relationship to memory perspective and attributions in social phobia," *Behav. Res. Ther.*, vol. 39, no. 6, pp. 651–665, Jun. 2001, doi: 10.1016/S0005-7967(00)00035-8.

[2] E. M. Seabrook, M. L. Kern, and N. S. Rickard, "Social Networking Sites, Depression, and Anxiety: A Systematic Review," *JMIR Ment Heal. 2016;3(4)e50 https//mental.jmir.org/2016/4/e50*, vol. 3, no. 4, p. e5842, Nov. 2016, doi: 10.2196/MENTAL.5842.

[3] Y. Chen, R. Li, P. Zhang, and X. Liu, "The Moderating Role of State Attachment Anxiety and Avoidance Between Social Anxiety and Social Networking Sites Addiction," *Psychol. Rep.*, vol. 123, no. 3, pp. 633–647, Jun. 2020, doi: 10.1177/0033294118823178.

[4] L. Baumann and S. Utz, "Professional networking: Exploring differences between offline and online networking," *Cyberpsychology J. Psychosoc. Res. Cybersp.*, vol. 15, no. 1, p. 1, Feb. 2021, doi: 10.5817/CP2021-1-2.

[5] "Anxiety: Causes and Management | The Journal of Behavioral Science." https://so06.tci-thaijo.org/index.php/IJBS/article/view/2205 (accessed Jun. 15, 2022).

[6] V. A. Orfanos, S. D. Kaminaris, D. Piromalis, and P. Papageorgas, "Smart home automation in the IoT era: A communication technologies review," *AIP Conf. Proc.*, vol. 2307, no. 1, p. 020054, Dec. 2020, doi: 10.1063/5.0032939.

[7] J. B. Carruthers, "WILEY ENCYCLOPEDIA OF TELECOMMUNICATIONS, 2002 (PREPRINT) Wireless Infrared Communications."

[8] M. Verma, S. Singh, and B. Kaur, "An Overview of Bluetooth Technology and its Communication Applications Nanomaterials and Polymer Solar Cells View

project International Journal of Current Engineering and Technology An Overview of Bluetooth Technology and its Communication Applications," *1588| Int. J. Curr. Eng. Technol.*, vol. 5, no. 3, 2015, Accessed: Jun. 15, 2022. [Online]. Available: http://inpressco.com/category/ijcetGeneralArticle.

[9] C. Wilson *et al.*, "Laser damage testing of windows with anti-reflection structured surfaces for high power continuous-wave near-infrared laser applications," *https://doi.org/10.1117/12.2519194*, vol. 10985, no. 17, pp. 67–72, May 2019, doi: 10.1117/12.2519194.

[10] F. Angelini and F. Colao, "Optimization of laser wavelength, power and pulse duration for eye-safe Raman spectroscopy," *J. Eur. Opt. Soc.*, vol. 15, no. 1, pp. 1–8, Dec. 2019, doi: 10.1186/S41476-019-0099-Y/TABLES/4.

[11] E. Bernardes, S. Viollet, and T. Raharijaona, "A Three-Photo-Detector Optical Sensor Accurately Localizes a Mobile Robot Indoors by Using Two Infrared Light-Emitting Diodes," *IEEE Access*, vol. 8, pp. 87490–87503, 2020, doi: 10.1109/ACCESS.2020.2992996.

[12] J. Liu *et al.*, "Mid and long-wave infrared free-space optical communication," *https://doi.org/10.1117/12.2530969*, vol. 11133, no. 6, p. 1113302, Sep. 2019, doi: 10.1117/12.2530969.

[13] B. Ganguly, P. Vishwakarma, S. Biswas, and Rahul, "Kinect sensor based single person hand gesture recognition for man–machine interaction," *Lect. Notes Electr. Eng.*, vol. 575, pp. 139–144, 2020, doi: 10.1007/978-981-13-8687-9_13/COVER/.

[14] N. K. Kulachenkov *et al.*, "Photochromic Free MOF-Based Near-Infrared Optical Switch," *Angew. Chemie Int. Ed.*, vol. 59, no. 36, pp. 15522–15526, Sep. 2020, doi: 10.1002/ANIE.202004293.

[15] F. Ahmed, M. Jawaid, F. A. Dahri1, S. Ali, and M. M. Jawaid, "A Review of Modulation Schemes for Visible Light Communication Design of Wearable Patch Antenna for Wireless Body Area Networks View project A Review of

Modulation Schemes for Visible Light Communication," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 18, no. 2, p. 117, 2018, Accessed: Jun. 16, 2022. [Online]. Available: https://www.researchgate.net/publication/324330270.

[16] "BlocklyDuino." https://blocklyduino.github.io/BlocklyDuino/blockly/apps/blocklyduino/ (accessed Jun. 16, 2022).

[17] "ESP32 Useful Wi-Fi Library Functions (Arduino IDE) | Random Nerd Tutorials." https://randomnerdtutorials.com/esp32-useful-wi-fi-functions-arduino/ (accessed Jun. 16, 2022).

[18] "Uploading sketches Over-The-Air (OTA) | Arduino Documentation | Arduino Documentation." https://docs.arduino.cc/cloud/iot-cloud/tutorials/ota-getting-started (accessed Jun. 16, 2022).

[19] "TV Remote control circuit diagram - Remote_Control_Circuit - Circuit Diagram - SeekIC.com." http://www.seekic.com/circuit_diagram/Remote_Control_Circuit/TV_Remote_control_circuit_diagram.html (accessed Jun. 16, 2022).

[20] U. Afruz, "DEVELOPMENT OF TV REMOTE JAMMER CIRCUIT USING 555 TIMER IC AND INFRARED LIGHT EMITTING DIODE," *Int. J. Eng. Appl. Sci. Technol.*, vol. 4, no. 11, 2020, Accessed: Jun. 16, 2022. [Online]. Available: http://www.ijeast.com.

[21] "555 Timer Tutorial and Circuits - Build Electronic Circuits." https://www.build-electronic-circuits.com/555-timer/ (accessed Jun. 16, 2022).

[22] "NE555 pdf, NE555 Description, NE555 Datasheet, NE555 view ::: ALLDATASHEET :::" https://pdf1.alldatasheet.com/datasheet-pdf/view/17972/PHILIPS/NE555.html (accessed Jun. 16, 2022).

[23] "Amazon.com: 555 Timer Chip IC ne555 Pulse Generator DIP-8 Single Precision Timer (pack of 30pcs) : Industrial & Scientific."

https://www.amazon.com/PoiLee-NE555P-Generator-Single-Precision/dp/B07FSDHX5K/ref=sr_1_5?keywords=ne555&qid=1655456149&sr=8-5 (accessed Jun. 17, 2022).

[24] "Amazon.com: Bridgold 40pcs(20pcs*NE555P+20pcs*8P IC Block) NE555 Precision Timing Circuits,8-Pin. : Industrial & Scientific." https://www.amazon.com/Bridgold-20pcsNE555P-20pcs8P-Precision-Circuits/dp/B07ZHHCY33/ref=sr_1_3?keywords=ne555&qid=1655456149&sr=8-3 (accessed Jun. 17, 2022).

[25] "Wholesale SYHY 100PCS NE555 555 DIP-8 IC Timers NEW GOOD QUALITY PRECISION TIMERS NE555P From m.alibaba.com." https://m.alibaba.com/product/62474894462/SYHY-100PCS-NE555-555-DIP-8-IC.html?__sceneInfo=%7B%22cacheTime%22%3A%221800000%22%2C%22type%22%3A%22appDetailShare%22%7D&from=Android (accessed Jun. 17, 2022).

[26] S. Pan, Q. Li, Z. Xian, N. Su, and F. Zeng, "The Effects of Laser Parameters and the Ablation Mechanism in Laser Ablation of C/SiC Composite," *Mater. 2019, Vol. 12, Page 3076*, vol. 12, no. 19, p. 3076, Sep. 2019, doi: 10.3390/MA12193076.

[27] Y. Chen, A. Cunningham, and A. Kotagiri, "The Danger of Laser Pointer–Induced Retinal Damage in Children: A Large United Kingdom Case Series and Survey of Public Awareness," *J. Pediatr. Ophthalmol. Strabismus*, pp. 1–8, May 2022, doi: 10.3928/01913913-20220425-01.

[28] "Technical Data Sheet 5mm Infrared LED , T-1 3/4 IR333-A," Accessed: Jun. 16, 2022. [Online]. Available: www.everlight.com.

[29] "Development Boards | Espressif Systems." https://www.espressif.com/en/products/devkits (accessed Jun. 16, 2022).

[30] "Arduino® Nano 33 IoT Target areas: Maker, enhancements, basic IoT

application scenarios.”

[31] “IR Receiver Modules for Remote Control Systems TSOP382.., TSOP384.. Vishay Semiconductors MECHANICAL DATA,” Accessed: Jun. 16, 2022. [Online]. Available: www.vishay.com.

[32] “IR Sensor 38KHz.” https://www.robotop.lv/en/light/592-ir-sensor-38khz.html (accessed Jun. 16, 2022).

[33] “SAM D21/DA1 Family Low-Power, 32-bit Cortex-M0+ MCU with Advanced Analog and PWM Features,” 2021.

[34] “Arduino Nano (V2.3) User Manual Released under the Creative Commons Attribution Share-Alike 2.5 License,” Accessed: Jun. 16, 2022. [Online]. Available: http://creativecommons.org/licenses/by-sa/2.5/.

[35] “NINA-W10 series Stand-alone multiradio modules Data sheet Title NINA-W10 series Subtitle Stand-alone multiradio modules Document type Data sheet Document number UBX-17065507 This document applies to the following products: Product name Type number Hardware version PCN reference Product status,” Accessed: Jun. 17, 2022. [Online]. Available: www.u-blox.com.

[36] “ATECC608A pdf, ATECC608A Description, ATECC608A Datasheet, ATECC608A view ::: ALLDATASHEET :::” https://pdf1.alldatasheet.com/datasheet-pdf/view/1179500/MICROCHIP/ATECC608A.html (accessed Jun. 17, 2022).

[37] “Source Official Arduino Nano 33 IoT with headers [ABX00032] original from Italy on m.alibaba.com.” https://m.alibaba.com/product/1600398356592/Official-Arduino-Nano-33-IoT-with.html?__sceneInfo=%7B%22cacheTime%22%3A%221800000%22%2C%22type%22%3A%22appDetailShare%22%7D&from=Android (accessed Jun. 17, 2022).

[38] "IR Sensor 38KHz." https://www.robotop.lv/en/light/592-ir-sensor-38khz.html (accessed Jun. 17, 2022).

[39] "Amazon.com : KOOBOOK 2Pcs Digital 38KHz Infrared IR Receiver Sensor Module for Arduino Compatible : Electronics." https://www.amazon.com/KOOBOOK-Digital-Infrared-Receiver-Compatible/dp/B07S66T3WT/ref=sr_1_18?crid=EGBKHMD2QPF9&keywords=infrared+sensor+module&qid=1655446090&sprefix=infrared+sensor+modul%2Caps%2C895&sr=8-18 (accessed Jun. 17, 2022).

[40] "Wholesale Okystar OEM/ODM Digital 38KHz IR Infrared Remote Control Receiver Sensors Module For Robot Electronic Blocks From m.alibaba.com." https://m.alibaba.com/product/62091403504/Okystar-OEM/ODM-Digital-38KHz-IR-Infrared.html?__sceneInfo=%7B%22cacheTime%22%3A%221800000%22%2C%22type%22%3A%22appDetailShare%22%7D&from=Android (accessed Jun. 17, 2022).

[41] "Amazon.com : XINGYHENG 100Pcs 50 Pairs 5mm Infrared Diode LED 940nm IR Emission and Receiver (Three Legs) : Electronics." https://www.amazon.com/XLX-Infrared-Emission-Receiver-Arduino/dp/B01MCYI5AC/ref=sr_1_8?crid=1T7OXJ8X5HUVW&keywords=IR+led&qid=1655447056&sprefix=ir+le%2Caps%2C308&sr=8-8 (accessed Jun. 17, 2022).

[42] "Amazon.com: Adafruit Super-bright 5mm IR LED (25 pack) [ADA388] : Electronics." https://www.amazon.com/Adafruit-Super-bright-5mm-pack-ADA388/dp/B00ULB0U44/ref=sr_1_3?crid=1T7OXJ8X5HUVW&keywords=IR+led&qid=1655446923&sprefix=ir+le%2Caps%2C308&sr=8-3 (accessed Jun. 17, 2022).

[43] "Source IR LED 940nm 5mm high brightness led diode made in china on m.alibaba.com." https://m.alibaba.com/product/1600280858860/IR-LED-940nm--5mm-

high.html?__sceneInfo=%7B%22cacheTime%22%3A%221800000%22%2C%22type%22%3A%22appDetailShare%22%7D&from=Android (accessed Jun. 17, 2022).

[44] "120 Pcs QTEATAK 6 x 6mm x 5mm Momentary Tactile Tact Push Button Switch for PCB 4 Pin DIP Black: Amazon.com: Industrial & Scientific." https://www.amazon.com/QTEATAK-Momentary-Tactile-Button-Switch/dp/B07VSNN9S2/ref=sr_1_8?keywords=push+button&qid=1655456640&s=industrial&sr=1-8 (accessed Jun. 17, 2022).

[45] "20 Pcs 6mm 2 Pin Momentary Tactile Tact Push Button Switch Through Hole Breadboard Friendly for Panel PCB: Amazon.com: Industrial & Scientific." https://www.amazon.com/Momentary-Tactile-Through-Breadboard-Friendly/dp/B07WF76VHT/ref=sr_1_16?keywords=push+button&qid=1655456685&s=industrial&sr=1-16 (accessed Jun. 17, 2022).

[46] "Source TS-C005 brass button 5*5 SMD/SMT momentary tact switch 4 pin mounted M terminal metal push button switch on m.alibaba.com." https://m.alibaba.com/product/60700609059/TS-C005-brass-button-5*5-SMD/SMT-momentary.html?__sceneInfo=%7B%22cacheTime%22%3A%221800000%22%2C%22type%22%3A%22appDetailShare%22%7D&from=Android (accessed Jun. 17, 2022).

[47] "IRremote - Arduino Reference." https://www.arduino.cc/reference/en/libraries/irremote/ (accessed Jun. 17, 2022).

[48] "Connecting the Nano 33 IoT to a Wi-Fi Network | Arduino Documentation | Arduino Documentation." https://docs.arduino.cc/tutorials/nano-33-iot/wifi-connection (accessed Jun. 17, 2022).

[49] "Arduino IoT Cloud API." https://www.arduino.cc/reference/en/iot/api/ (accessed Jun. 17, 2022).

[50] "Arduino Cloud IoT | Arduino Documentation | Arduino Documentation." https://docs.arduino.cc/cloud/iot-cloud (accessed Jun. 17, 2022).

[51] "Arduino Forum." https://forum.arduino.cc/ (accessed Jun. 17, 2022).

[52] A. Bhawiyuga, D. P. Kartikasari, K. Amron, O. B. Pratama, and M. W. Habibi, "Architectural design of IoT-cloud computing integration platform," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 17, no. 3, pp. 1399–1408, Jun. 2019, doi: 10.12928/TELKOMNIKA.V17I3.11786.

[53] "Say hello to Arduino Cloud, more things and two new plans | Arduino Blog." https://blog.arduino.cc/2021/05/03/say-hello-to-arduino-cloud-more-things-and-two-new-plans/ (accessed Jun. 17, 2022).

[54] B. K. Muslmani, S. Kazakzeh, E. Ayoubi, and S. Aljawarneh, "Reducing integration complexity of cloud-based ERP systems," *ACM Int. Conf. Proceeding Ser.*, Oct. 2018, doi: 10.1145/3279996.3280033.

[55] "MIT App Inventor." https://appinventor.mit.edu/ (accessed Jun. 17, 2022).

[56] "Arduino Rest API: How to use and create API in IoT." https://www.survivingwithandroid.com/arduino-rest-api-iot/ (accessed Jun. 17, 2022).