

Yare 4 project

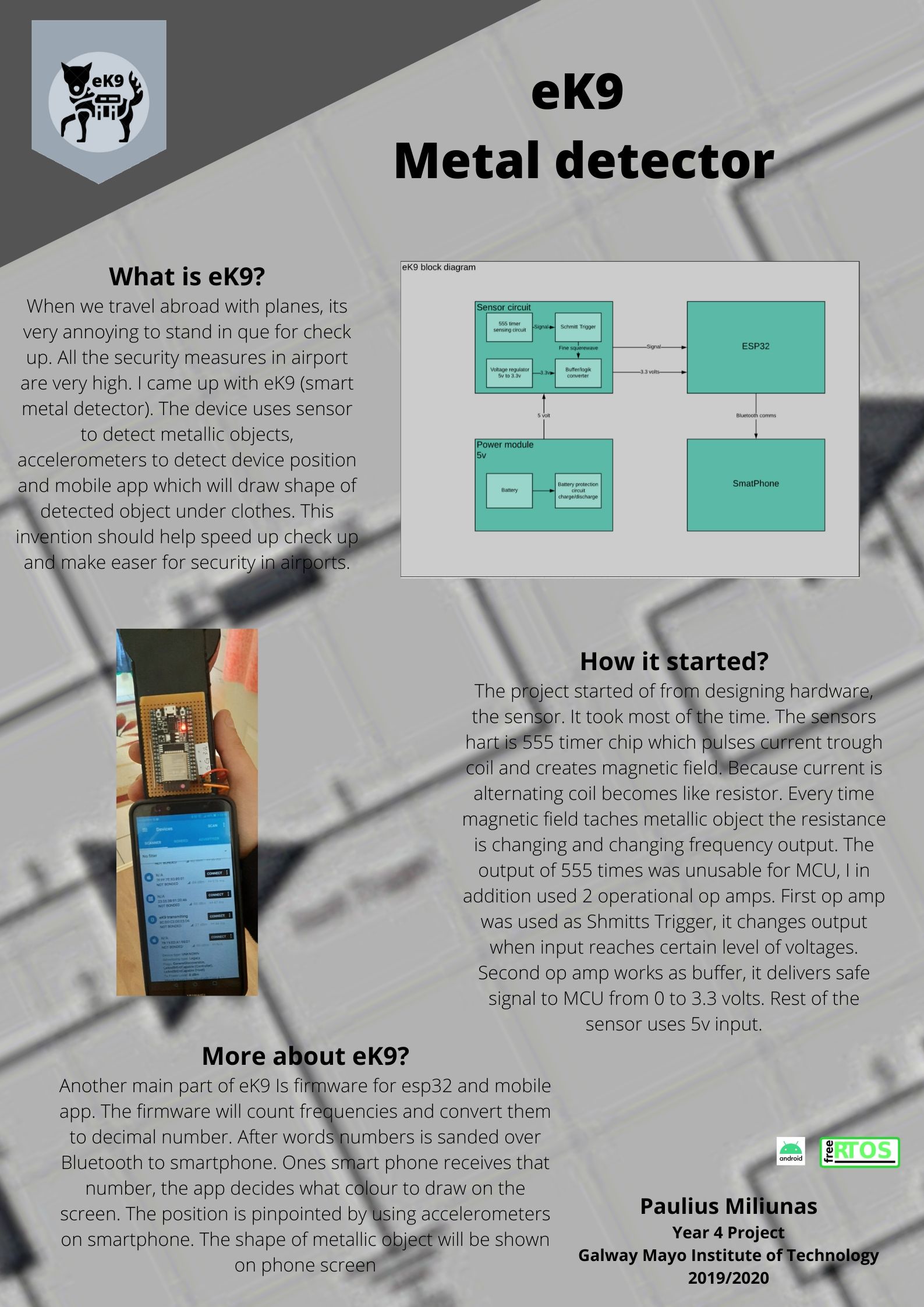
eK9 - Smart Metal Detector

Paulius Miliunas

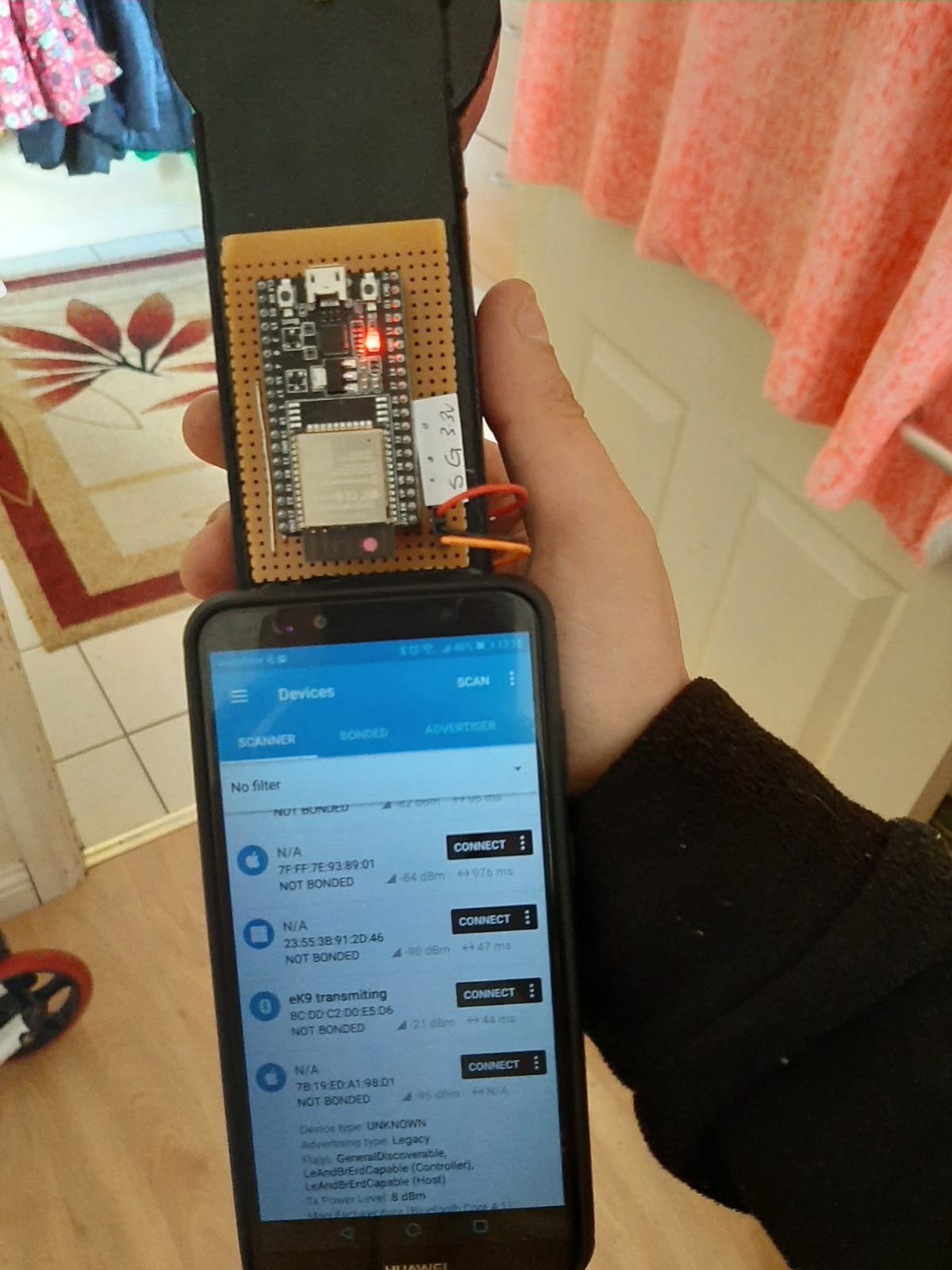
Honours Bachelor of Software & Electronic Engineering

Galway-Mayo Institute of Technology

2019/2020

****

**eK9**



Pictures of eK9. Bottom picture displays connectivity with eK9 app is used from Nordic Semiconductors (it was used for debugging).

**Declaration\*\***

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Software & Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgements**

Fourth year was real very challenging. I received a lot support from everyone in college. Lectures helped a lot. But most of support I received from my kids and my wife. Without them my project would not see day light. I very happy whit what I made and achieved.

During year there was a lot of thing to do and specially to learn. Loads of time was spend o studying other modules, I wanted to understand how all works, that’s why I had to priorities my work. In the end I am very happy with what I achieved. I learned a lot and finally I understand much better how the coding works. After finishing college, I will continue study and advancing trough knowledge.

**Table of Contents**

[1 Summary 7](#_Toc36851109)

[2 Introduction 8](#_Toc36851110)

[3 eK9 Overview 9](#_Toc36851111)

[4 eK9 Architecture 10](#_Toc36851112)

[5 Development Platform and Tools 11](#_Toc36851113)

[6 eK9 Hardware 12](#_Toc36851114)

[6.1 Main Sensor V1 12](#_Toc36851115)

[6.1.1 Power supply 12](#_Toc36851116)

[6.1.2 Oscillator 13](#_Toc36851117)

[6.2 Main Sensor V2 14](#_Toc36851118)

[6.3 The brain of eK9 16](#_Toc36851119)

[7 Software 18](#_Toc36851120)

[7.1 Firmware 18](#_Toc36851121)

[7.2 App 18](#_Toc36851122)

[8 eK9 frame 19](#_Toc36851123)

[9 Problem Solving 21](#_Toc36851124)

[10 Conclusion 22](#_Toc36851125)

[11 References 23](#_Toc36851126)

[12 Code 24](#_Toc36851127)

# Summary\*\*

eK9 is a simple smart metal detector. This device can detect an object at 10-15 cm. The device reacts differently to different materials and size of it. If steel is detected the device will show reading different then on the detection of aluminum or other non-magnetic metals.

I used esp32, the code for it was written in Visual Studio Code from Microsoft. PlatformIO addon was used in IDE. Esp32 have Wi-Fi and Bluetooth transmitters, I was using Bluetooth functionality to communicate with smartphone. On the smartphone app was written with android studio.

The eK9 should be portable, lightweight device. The device should communicate with smartphone and transmit data to smartphone. Smartphone should create picture on 2d plane to display shape od the object. eK9 should use accelerometers from smartphone to detect how much phone vas moved.

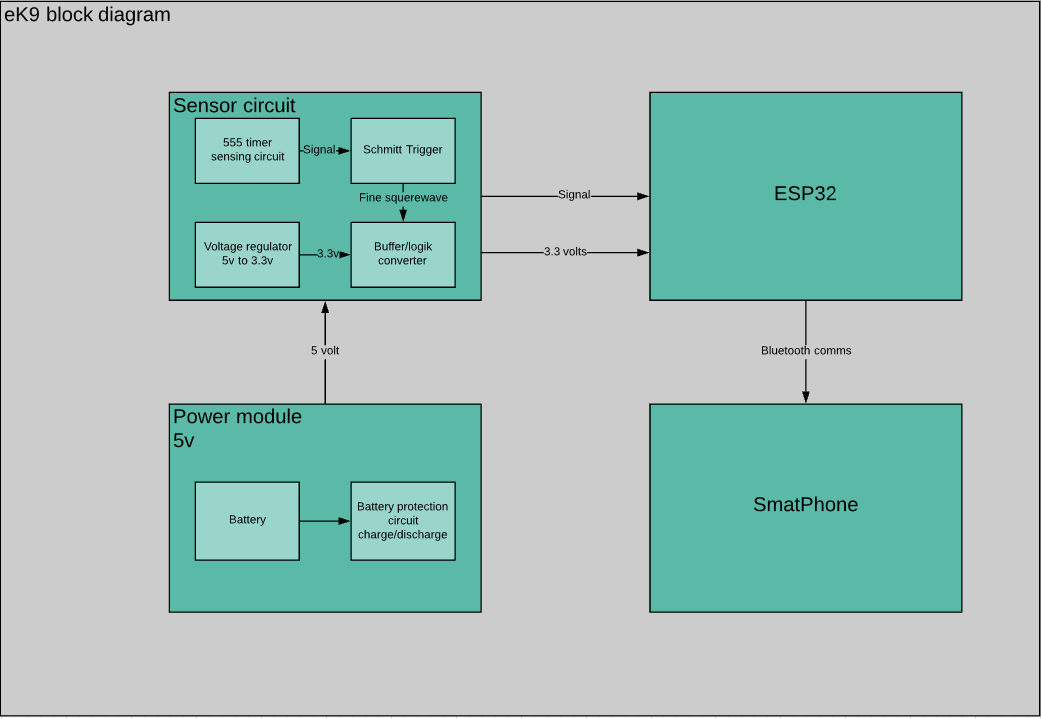
# Introduction

This year I was developing a device that will detect metallic materials and display the shape of the detected object. The device name was chosen eK9. The name was chosen in one of the lectures on project management. Most of the class was suggesting this name will be best and I liked it. Main sensor of device pulse current trough the coil, coil create alternating magnetic field. If field is distorted sensor changes frequency output. Sensor capable seeing difference in ferrite and non-ferrite objects. Ferrite material would be considered as metals which are attracted by magnet, non-ferrite materials are not attracted to magnet. The frequency reading from sensor is feed to microcontroller and send via Bluetooth to smartphone. The smartphone used data to decide when to draw on the screen when don’t. Accelerometers on smartphone will detect distance how much sensor was moved. Combined data from microcontroller and data from accelerometers I should be able to draw shape of detected object.

# eK9 Overview

In the beginning, I was trying to build a powerful metal detector. But it consumed to much time and I didn’t finish the circuit on time. All first semester was consumed on building hardware. Original idea was to build a dual power supply with the virtual ground and positive, and negative + volt outputs. This power supply was supplying power for op-amps which were crating sinusoidal waveform. I connected coil to output and noise was appearing on the signal. I was hoping that I will be able to use this setup. Next thing I was thinking to add op-amp in differential amplifier configuration and feed signal from both coils. On output, I was hoping to get a difference in signal. But this was not happening. So, I decided to go a different way. At the beginning of the project, I wanted to use 555 timers as my project but this setup cant increase power to the coil. The device will be not that powerful. I used a simple circuit where 555 timer uses a coil and when any metallic material amperes near coil output frequency was changing. I modified the circuit that I was able to get square waves from 0 to 3.3 volts.

# eK9 Architecture



Ek9 Block diagram is shown above.

Power Module delivers power to device. Module have battery protective circuit. This power module is made from scrap parts of small power bank. Module charges battery cell with mini USB cable and output 5volts. Module have voltage regulator, charge protection and discharge protection. Charge protection is monitoring voltage of battery and when battery voltage reaches 4.2volts it disconnects charging. Discharging protection protect battery cell from discharging it to dangers low voltages, if lithium-ion battery gets discharged below voltages defined by manufacturer battery cell can be damaged and will not perform good afterword’s. Battery cell used in project cut of voltage is 2.8-3.0 volts, that’s mean protective circuit will allow draw current until 2.8 volts is reached on battery.

Sensor circuit have four parts. Main part is 555 timer circuit. This circuit sends pulses to coil at frequency, that frequency is resonant for coil and capacitor connected to it works like LC circuit. Any time ferrite of non-ferrite metal object passes by the coil, object changes magnetic field capacity and coil changes its resonant frequency. Depending on material frequency increases or decreases from idle frequency value(2897Hz). 555 timer sensing circuit outputs signal in amplitude 6.3 volt. The output signal has spikes on starting and ending edge. The spikes are produced by coil when it is switching voltages of happens. To eliminate spikes, I used op-amp as Schmitt’s trigger. It converts signal to nice square wave, with which I can work further along. Schmitz trigger is stiches voltage on output when its input is at values defined with resistors. Buffer circuit act as logic converter and converts 5 volts signal to 3.3 volts.

Then signal is feed to esp32 where it programmed to take time measurement between falling edge of the signal. In this way I calculated frequency. Bluetooth capability on esp32 was programmed to broadcast frequency. Where smartphone app is picking up data and using to decide when to draw on screen and when not to draw.

# Development Platform and Tools

eK9 was developed on multiple platforms. For firmware programming I was using Visual Studio code IDE with PlatformIO addon installed. This tool is very helpful, with this addon possible to use more than 800 microcontroller development kits. C language was used to active functionality required.

Circuits was designed with Altium. This Software is powerful. Tool is used from design circuit to complete product, and toll capable to generate various documentation on the project. I used this toll for circuit design and PCB design.

Another great toll I was using is LiveWire. This tool has simple simulation of electronics components and circuits. I used it to sketch my circuits for report and used to prototype parts of system before building on breadboard.

Smartphone app was written in java using Android Studio. This IDE is very powerful. It has all required extensions to work with all version of the android. The IDE have Smartphone emulator which make much ease debugging, but not always. In my case working with Bluetooth was causing app to shut down, emulator can’t emulate Bluetooth hardware. I had to use my own phone, and that’s was more amazing than using emulator.

# eK9 Hardware

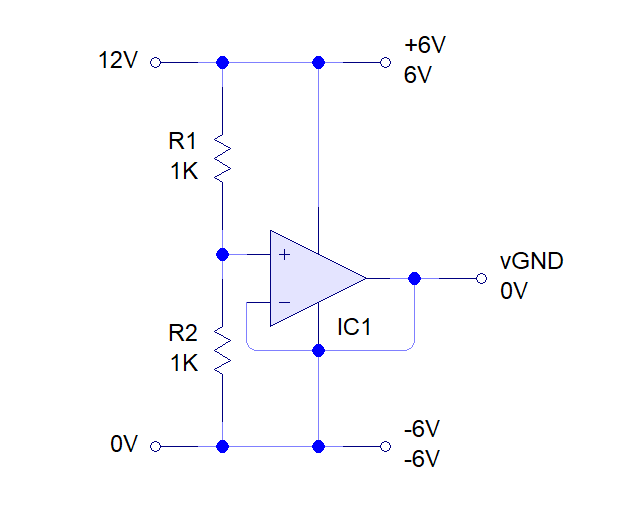
eK9 device hardware is explained in this section. Every subsection corresponds to different modules of eK9. There was 2 versions of the sensor. Both was included in this section.

## Main Sensor V1

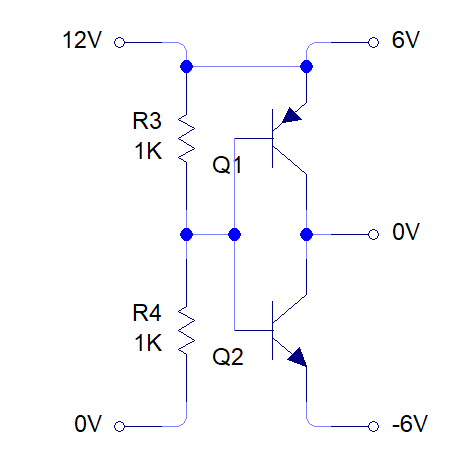
First version of the sensor I was trying to design, was to ambitious. Idea was to use dual power lines (-6, 0 +6) volts. Negative and positive power lines required for op amps operate and both negative and positive sides with sine waves. After I had to crate sine wave to pulse to coil at selected frequencies, like frequency generator.

### Power supply

After locking trough web how to split voltages I found few different setups. First, I tested circuit where voltage division was achieved with op-amp. Op-amp gets on positive input half of maximum voltage in power supply and used like buffer will have same ½ voltage one output (circuit below). This circuit is stable can provide stable virtual ground which is simply divided voltage output is used as ground reference, and real ground is used as negative voltage. Negative side of this circuit it can’t supply much current (LM741 can supply up to 25mA of current).



Second circuit I approached is using transistors. Their inputs are connected to 2 resistor networks as voltage divider. This circuit can deliver current depending on transistor current trough put (circuit below).



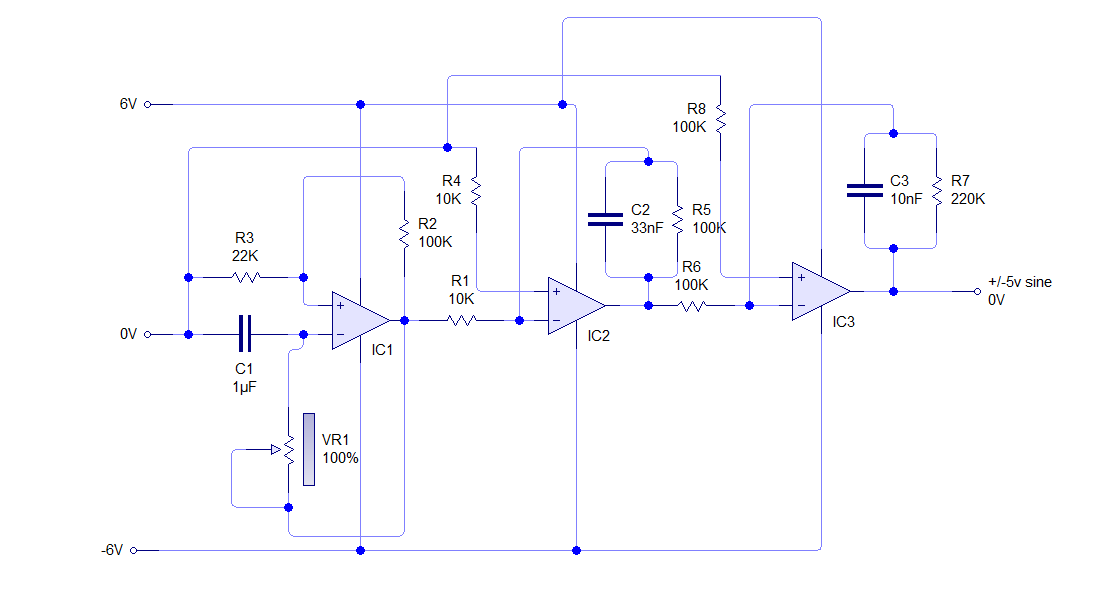
I decided to combine both circuits together to active stable virtual ground and enough current to supply for next circuits. As u can see in circuit bellow simple voltage dedidion insetred to op-amp positive input gives ½ of woltage aplyed on power teminals. Transistors gets op-amp output and regulates curent flov trough transistor. Op-amp feadback is connected betven transitors to keep voltage at same lvl. Capasitor is used to smoth riples on the power lines.

A close up of a map

Description automatically generated

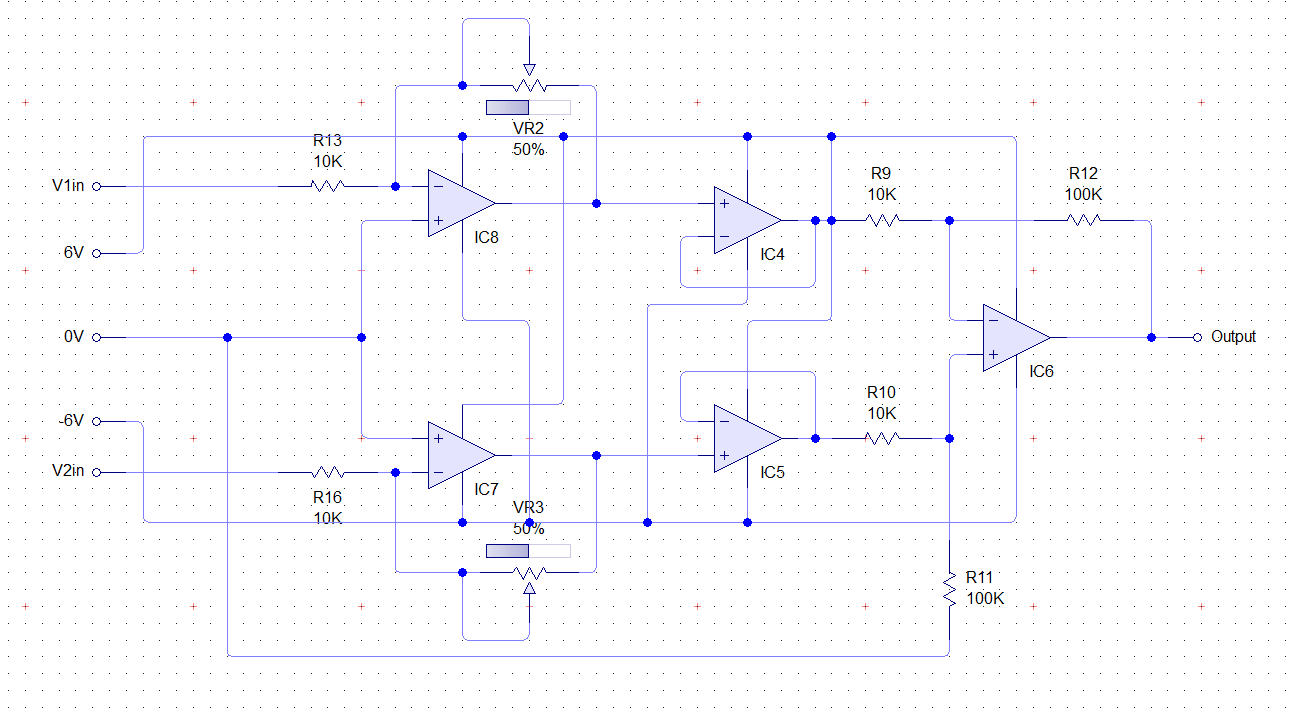
### Oscillator

Oscillator circuit I found in second year notes for analog electronics. I replicated circuit on bread board and used my split power supply I was able to get oscillations. After connecting coil to output I noticed noise in signal, which I was not able to remove. I was thinking to build next part of sensor (differential op amp) and connecting filters between.

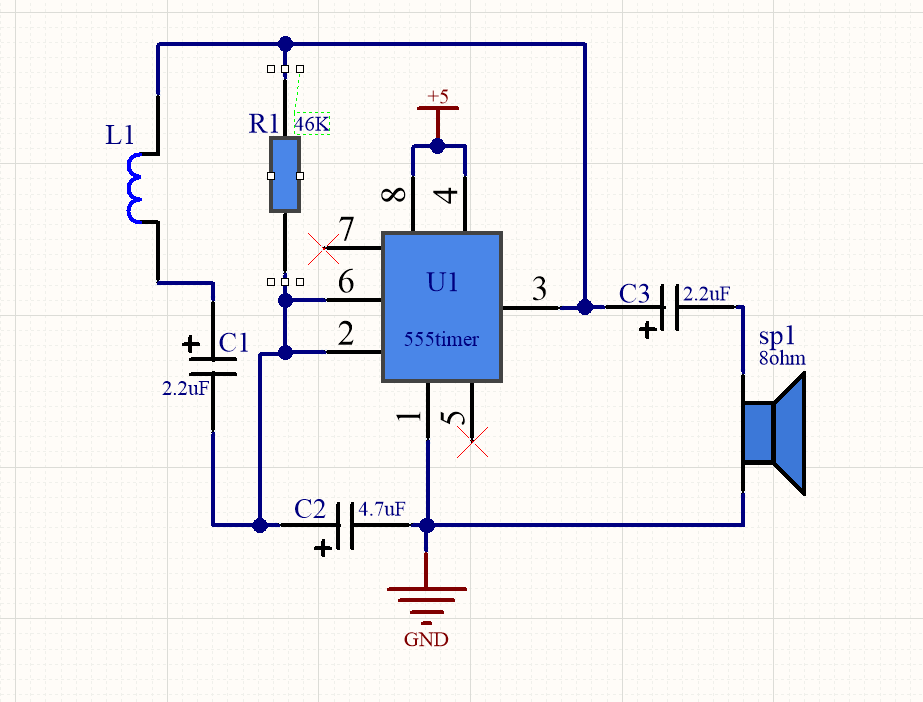


### Differential Amplifier

This circuit was newer build. Due delays on development and decision was made to change sensor which les complex to develop. For this circuit I had decomposition task. I think if I would feed signal from transmitting coil to one input and correctly preamplifier receiving signal, and feed it to second signal input, the output should show only difference in signals. That means if rx signal minus tx signal what left would be signals repelled from metal objects. Then signal can be integrated for simplicity and feede to MCU. This output I believe would be possible to use on any MCU. With addition hardware would be possible to use sensor on any digital system desired.

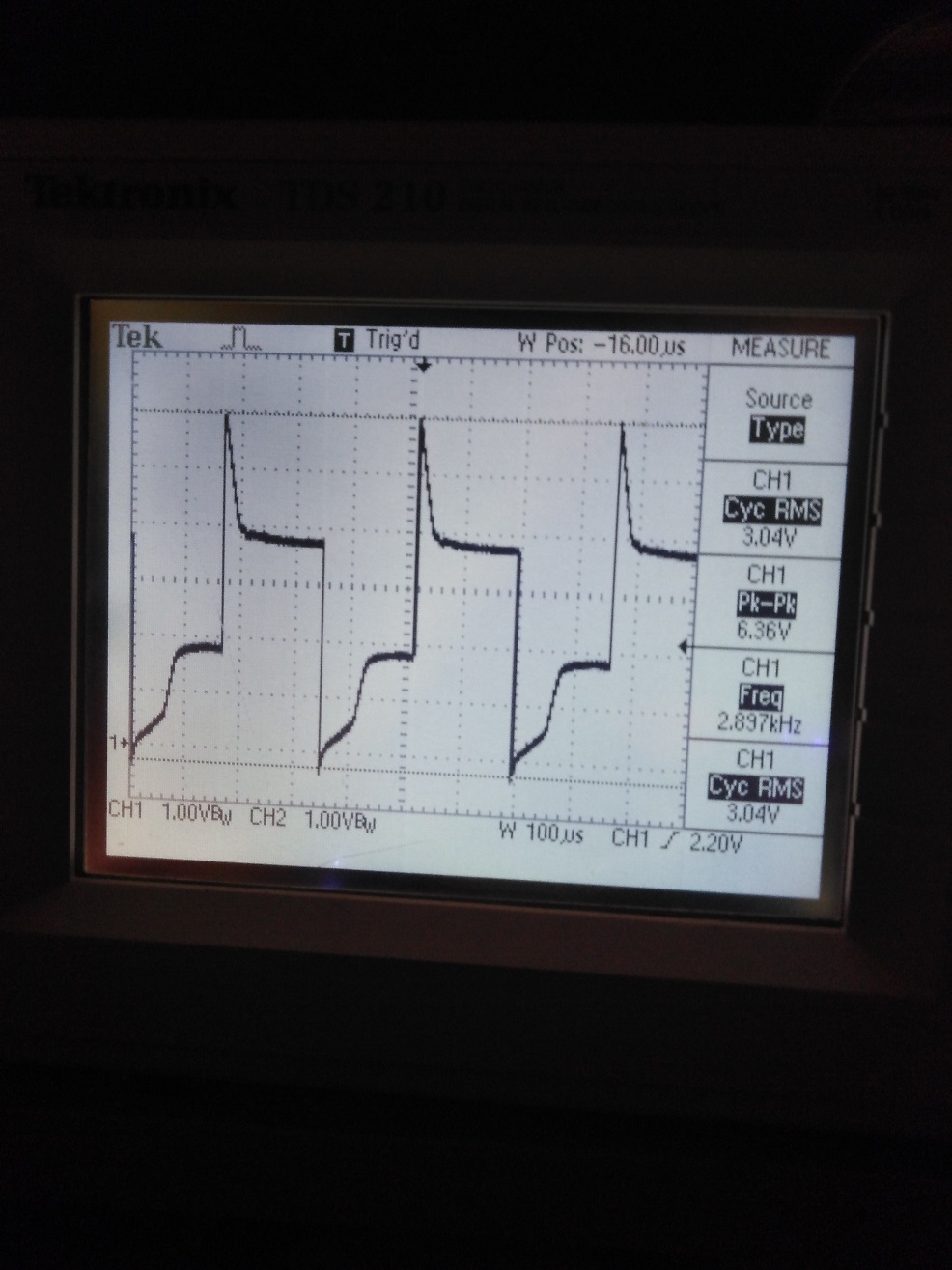


## Main Sensor V2

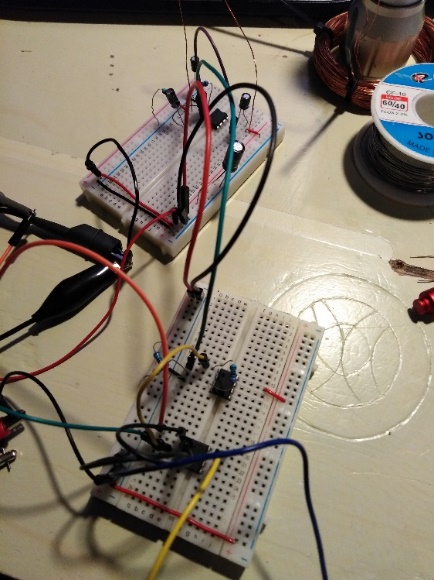


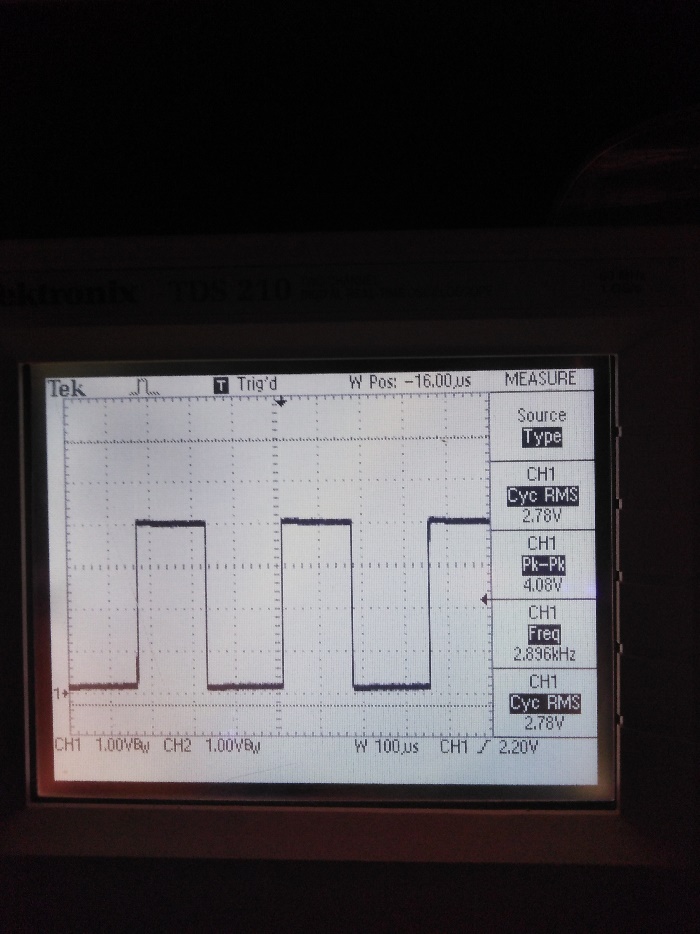
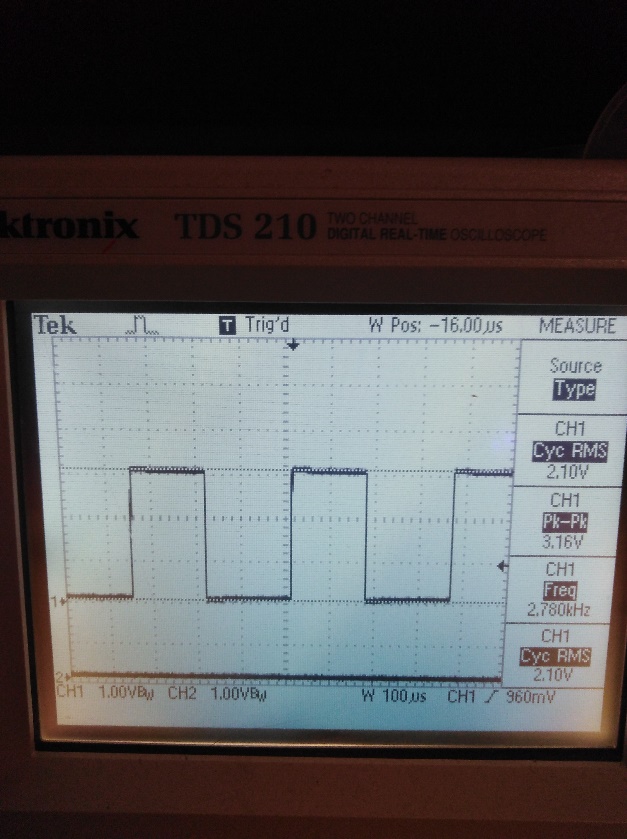
*555 timer circuit using the coil to vary the output.*

Before I have seen the circuit, very simple which using 555 timers, coil, and speaker to notify the user. YouTube is full of that circuit. So, I decided to use that circuit and add some modifications. I removed the speaker and instead added a resistor. Measured output with an oscilloscope.



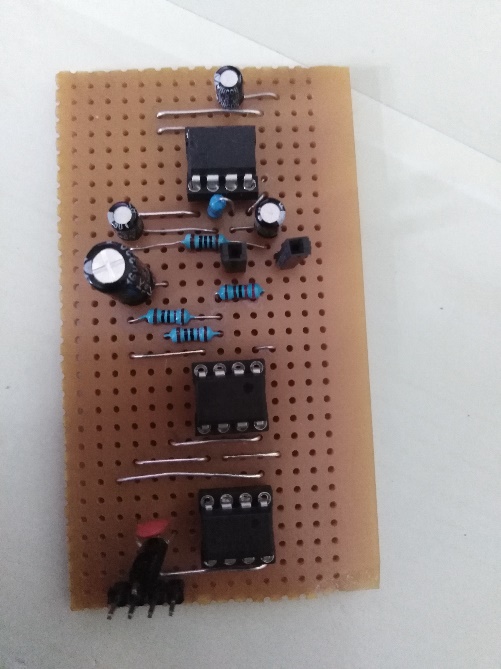
The frequency is steady 2897HZ. I tested different materials like steel aluminum copper, gold. I noticed that if I use material which are attracted by magnet frequency was decreasing and if aluminum coper or good frequency was increasing, and change was depending on size of tested object. In the future, I will use this to approximate materials detected. Due coil one end connected to output other to capacitor we can see spikes on change edges. I had to figure out how to get a usable signal for MCU. After listing second-year analog devices papers, I have remained about smith’s trigger. After building it on the separate breadboard (picture below).



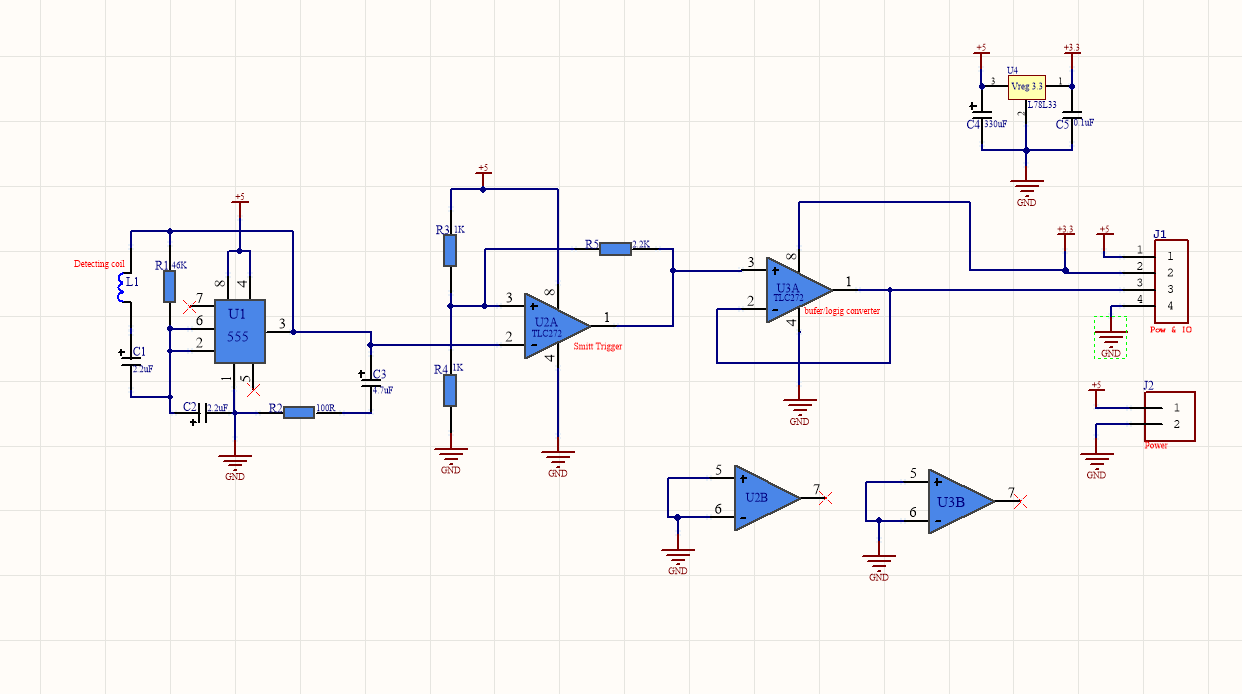
 

(Signal output capture smiths trigger) (Signal output of second op amp as buffer)

I measured outputs. Signal was looking much better. Finally, was my chance to test op-amp as a buffer (logic level sifter from 5v to 3.3v). Waveforms you can observe above. That signal was feed to esp32 to calculate frequency. After testing the circuit with esp32 I soldered everything on stripboard as you can see below.



Prototype on stripboard

Parts U2B and U3B are not used in this case. Their input is terminated to ground to minimize interference and energy consumption. 

## The brain of eK9

Esp32 will deliver enough processing power. This little beast is counting the frequency of sensor and transmitting data via Bluetooth (BLE). Development kit from Expresif the ESP32 WROM 32 has imbedded Wi-Fi and Bluetooth which can work as normal Bluetooth and as low power Bluetooth. The system is very interesting, the device work as a server and transmits small packets of data over small distances. Then another device like a smartphone set up as a client can pick up those packets and read its content. This technology will fit my needs to transmit my frequency values over a small distance to my phone for further analysis.

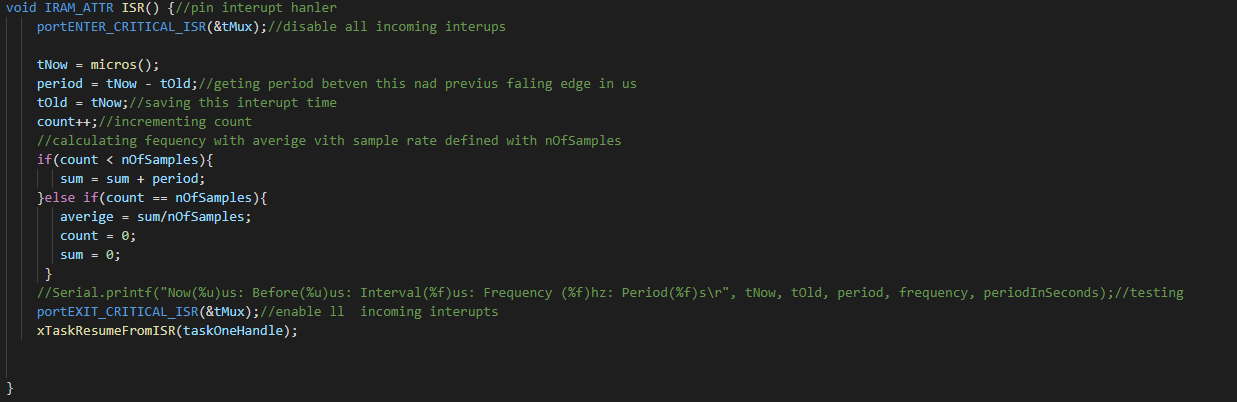
The ssmartphone will receive the data and read it. Bu value app will determine color which will be used to draw. Another type of sensors will be used is an accelerometer. The accelerometer will read the position of the phone and determine how many phones were moved and the app will know how much distance the phone traveled. Two different data will complete all the pictures. The accelerometer will determine distance how much phone is moved and the frequency value will show the intensity of the material.

# Software

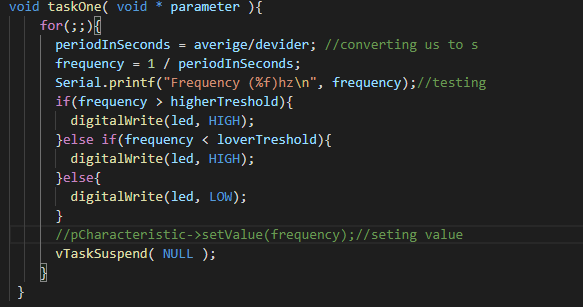
I found very handy IDE Visual Studio Code. Which I was using to program esp32. The IDE is very flexible with plugins. One of plugin I was using “PlatformIO”. Plug in includes more than 800 different board compatability and projects are self-contained. Firmware is written in Visual Studio Code. The Smart Phone App s developed in Android Studio environment.

## Firmware

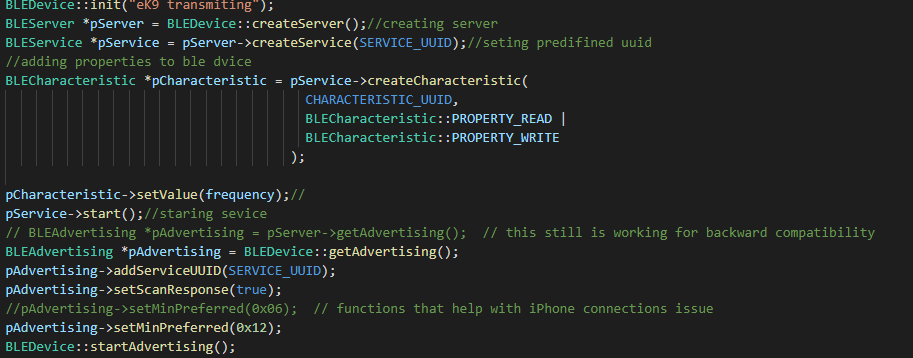
Code task is to measure frequency output of the sensor and transmit it over Bluetooth comms. Firmware running on real time OS with Arduino libraries implemented. BLE libraries makes available Bluetooth to radio data to Smartphone. Code employs interrupt on falling edge. At each interrupt time measurement are taken and compared with previous interrupt time event. The difference is used to calculate period and frequency. Time is measured with internal timer



Interrupt setup to happen when falling edge is detected on pin (pin 4 was used on esp32). After detection interrupt handler runs function shown in sniped. It takes measurement of time in microseconds and save value as previous time value(tOld) when next interrupt happens function measures time again and calculates deferments between interrupt. I applied average filter to calculate average sample value. The value of number of samples ca be set in top of the code(noOfSamples). After algorithm calculates average value its stored in to (average) value.



Every time task runs ones and blocks itself until interrupt happens. Function uses global variable average and calculate frequency. This frequency should be sent vie blue tooth to smartphone. In this case we using LED connected to esp32 which indicates when device is detecting metallic objects. LED control is achieved by if else statements where lover and higher thresholds are predefined.



In this sniped I am showing code which creates Bluetooth server with selected UUID services. I am setting up characteristic to read and write that I can send and receive data via Bluetooth. Server broadcast that it is available, and every time client connects it should print value of the frequency on the phone.

## App

Development of app I found hardest part of all project. I am using Android studio to write code for smartphone. I used number of different examples from internet but no of them I was available decompose and understand in the way that I would able to write my own program. I have some examples done, but I was not able to connect to my devise. I was using Nordic RF App which help me to test esp32 functionality. I was able to connect to device and read data or see different info available about device.

At the beginning of development eK9 connect app I started from Android developer website. I followed tutorial but unfortunately, I failed to understand how the app must developed. From Android development web site, I was able to turn on and turn off blue tooth. Then I speeded number of weeks just locking for examples of android studio code. I came across tutorial on YouTube <https://www.youtube.com/watch?v=bLpwXjk1TG8>. In beginning I used his code just for testing purpose. After following this tutorial, I was able to learn how to write code which will discover other devices and display them in interactive list. But attempts of connecting device with app was keep failing on me. After paling around with esp32 I remembered about one app. Its from Nordic called nRF Connect. This app is brilliant whet it come to RFCOMS like Bluetooth. After I used mainly this app to debug my esp32 code. After failing connect esp32 with my app I used nRF Connect to test connections. And nRF Connect was able to find and connect to my device, I was able to read specifications which I entered in esp32 firmware. After several attempts which failed, I contacted one of the lectures where I was point to one of classmates. He was working on Bluetooth common his own project; I was point to very good tutorial on YouTube.<https://www.youtube.com/watch?v=y8R2C86BIUc&list=PLgCYzUzKIBE8KHMzpp6JITZ2JxTgWqDH2> .I followed this tutorial and I thought I understand how it work. I Type out both app he is providing two times. But still I was not able to get connected to my device. And the worst part of the worst is that issue is very small and stupid like always and I can’t realize what and where.

# eK9 frame

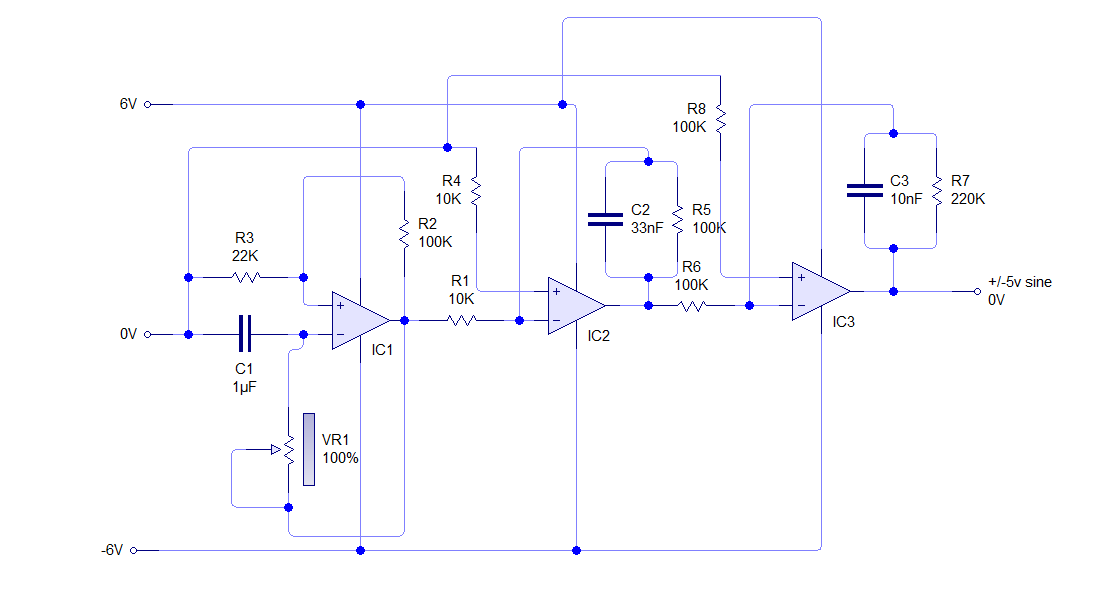
eK9 enclosure I was designing to be handheld, small and portable. Material I used for encloser was carboard paper materiel this give me light device. I simply used carboard box cut it to required peace’s and glued with glue gun. Used some black spray paint. The device had to be prolonged little bit I was making sure that magnetic field won’t cause any issues for micro controller. That’s why the antenna of esp32 was turned away from coil. I know that magnetic field will interfere with comms. After I fit all modules inside enclosure and glued them wit the same hot glue which was very handy, I was very happy with how my device was locking.

# Problem Solving

In project development there was every day problem solving issue. Very beginning of development big dilemma was what I going to build and what I will be able to finish and present. I didn’t have much time and on my head was thought about metal detector so I wanted do make my metal detector and learn on the way.

Firs version of metal detector I starter was to complex but fancy metal detector. I made plan to how I going to build this. Circuit designed in Altium was not very complex task. I prepared all for my first prototype build. First, I made dual power supply after start connecting first op amps and I noticed that when op amps start operating voltage on power lines. After some time of measuring few nodes available on my circuit I realized that my power line splitter is not capable to deliver enough current. So, I decided to go and lock on web is there anything I can find. Then I noticed circuit which can spit power lines, that circuit was using transistors not op amp. I found more information about difference in thus two circuits, and I decided to use both in my circuit. Op amp circuit vas very stable and keeping virtual ground steady and transistor circuit vas capable delivering enough power. I made few sketches and build on breadboard. Rails was tested and I was able to proceed further.

Next difficult task was to get beautiful sinewave where I able control frequency. I tried few different circuits. I found them in second year Analog devices module notes. Wien bridge was my first attempt. But I couldn’t make it work normally I tried for few days and moved on different circuit. Which was using 3 op amps. The circuit work from noise picked up by first op amp resistor help to set frequency which we going to pass to output. Op amp saturates every time, that’s why we see secure wave on output at frequency set by resistor and capacitor. Otter two op amps is integrators and they integrate square wave to triangle wave and to sine wave.



After building circuit and connecting coil to output I was getting noises I think they was caused by coil. I was not able to figure out why noise is appearing. After demonstration on in December I decided to go for something simple.

\*\*\*

Development of second sensor was less stressful. I used simple circuit with 555 timer. The 555-timer pulsating voltage and LC resonant frequency, every time metallic object passes coil resonance changed and frequency is changed. Problem occurred with output. Coil was causing sharp spike few days I was thinking how to change this to something I can red with esp32. After listing same notes from second year, shmits trigger was perfect solution. Build circuit on different breadboard and connecting it, waveform was transformed to square wave with more then 4volts in amplitude. The third thing was to reduce voltage to 3.3v. For this I connected 3.3vort regulator and powered second op amp from it. Signal was reduced and it was safe to connect to esp32.

\*\*\*

Firmware development first part was good. I successfully implemented task to calculate frequency. There was no problem getting interrupt running. Reading timer values. But the Bluetooth brought for me trouble.

# Conclusion

By developing my project there was a lot of hard corners. I was expecting to manage time and finish my project but unfortunately I was not able. The biggest issue was time management. The beginning of the year was good but down the road everything just went wrong.

The idea for my project was to make simple and reliable metal detector which would be able to detect and draw shape of the detected object. The sensor and physical part were completed but programming part was very terrible. I speeded a lot of time trying to understand how to make app functional but every time I was coming to the dead end and had to start from fresh. So far, I tried numbers of tutorials, they all very complicated and I was getting confused. My style of learning programming is to take somebodies example, rewrite code in simple manner then I can learn. Where with Bluetooth I was not able to manage.

After finishing this last year of college, I will continue working on RFCOMS, I would love to learn this and use it in my projects in the future.

# References

Some example references are given here. For example, the section in your report on your servo motors should have a sentence that includes [1] in it, referring to reference [1] here.

[1] Arduino, “Servo library”, arduino.cc, 2018.

[2] Bill Earl, “Memories of an Arduino”, adafruit.com, 2018.

[3] Texas Instruments, “High speed CMOS logic analog multiplexers”, 74HC4051 datasheet, 1997.

[4] Stack Overflow, “Is there a way to have more than 14 output pins on Arduino?”, stackoverflow.com, 2014.

[5] Nick Gammon, “Interrupts”, gammon.com.au, 2012.

# Code

Include your top level Arduino .ino file, and any .cpp/.h files you created. Do not include third party code.