REPUBLIC OF TURKEY

GAZİANTEP UNIVERSITY

Determination of Velocity of Object Using Radar Sensors

Graduation Project

in

Engineering of Physics

BY

MUHAMMED UBEYD

GAZIANTEP-FALL-2021-2022

Determination of Velocity of Object Using Radar Sensors

Graduation Project

in

Engineering of Physics

Gaziantep University

Supervisor

Prof. Dr. Ramazan KOÇ

by

Muhammed UBEYD

GAZIANTEP-FALL-2021-2022

ABSTRACT

Determination of Velocity of Object

Using Radar Sensors

UBEYD, Muhammed
Graduation Project in Engineering of Physics
Supervisor: Prof. Dr. Ramazan KOÇ
FALL-2021-2022

The application of radio detection and ranging in different places such as military installation, commercial use is done with the help of RADAR SYSTEM which uses electromagnetic waves for detection of different physical components such as distance, speed, position, range, direction, size etc. which can be either fixed or be in motion. Use of radar system has been developed greatly specially in field of navigation. In this research we study about existing navigation technologies and proposed an Arduino based radar system. It has advantage over other radar system as kit reduces power consumption and connect programmer to wide range or Arduino programmers and open-source code.

ÖZET

Nesnenin Hızının Belirlenmesi

Radar Sensörlerini Kullanma

UBEYD, Muhammed
Bitirme Projesi, Fizik Muh.
Danışman: Prof. Dr. Ramazan KOÇ
GÜZ-2021-2022

Askeri kurulum, ticari kullanım gibi farklı yerlerde radyo algılama ve menzil uygulaması mesafe, hız, konum, menzil, yön, boyut vb. farklı fiziksel bileşenlerin tespiti için elektromanyetik dalgalar kullanan RADAR SİSTEMİ yardımı ile yapılır. sabit veya hareket halinde olabilir. Radar sisteminin kullanımı özellikle navigasyon alanında büyük ölçüde geliştirilmiştir. Bu araştırmada mevcut navigasyon teknolojileri hakkında bir çalışma yaptık ve Arduino tabanlı bir radar sistemi önerdik. Kit, güç tüketimini azalttığı ve programlayıcıyı geniş veya Arduino programcılarına ve açık kaynak koduna bağladığı için diğer radar sistemlerine göre avantajlıdır.

ACKNOWLEDGEMENTS

These last steps in university life would not have been possible without the support of my family, professors, and friends.

I am very pleased and honored to thank my advisor Prof. Dr. RAMAZAN KOÇ who inspired me to never stop learning and improving my skills. He also motivated me to always look for the Professionalism and perfection of what I do.

I would like to convey my gratitude to my supervisor Prof. Dr. HÜSEYİN TOKTAMİŞ, for his help and patient. She taught me that learning is such a journey of fun, because of the atmosphere of friendship and warmness which she usually creates.

With my Love to all teachers at Gaziantep University our esteemed professors who gave us a lot, making great efforts in building the generation of tomorrow to send the nation back.

My family who believed and supported me and taught me how to be ambitious.

My friends, thank you for listening, offering me advice, and supporting me through this entire process.

TABLE OF CONTENTS

Page

ABSTRACT.....iii

ACKNOWLEDGEMENT.....v

TABLE OF CONTENTS..... vi

LIST OF FIGURES.....vii

1. CHAPTER I 1

INTRODUCTION.....1

2. CHAPTER II 2

ARDUINO 2

1. Definition 2
2. Board Description of Arduino 2
3. Arduino Programming 3
4. Arduino Sensor – Types and Applications 4

3. CHAPTER III 6

RADAR SENSORS WITH ARDUINO 6

1. Introduction of Radar System 6
2. Arduino Radar Sensor Overview 7
3. Advantages of the Arduino Radar Sensor.....7
4. Disadvantages and Limitations 8

4. CHAPTER IV 9

HB100 SENSOR 9

1. Definition 9
2. Power Supply 9
3. Transmit Frequency 10
4. Radiation Pattern 10
5. Output Signals 10
6. Sample with Arduino 11
7. Sample Code for Arduino with HB100.....12

5. CHAPTER V 14

RCWL-0516 SENSOR 14

1. Definition 14
2. Module Description 15

3. Schematic ……………………………………………………….……….15
4. Sample with Arduino ……………………………………………..……..18
6. CHAPTER VI ………………….…………………………………………….25
CONCLUSION ………………….……………………………………………25
REFERENCES...26

LIST OF FIGURES

Figure 2.1 Arduino UNO Board

Figure 2.2 Arduino IDE 1.0.6 version

Figure 3.1 Arduino Radar Sensor

Figure 3.2 Block Diagram

Figure 4.1 HB100 module

Figure 4.2 HB100 module Radiation Pattern

Figure 5.1 RCWL-0516 module

Figure 5.2 RCWL-0516 module with Arduino Board

Figure 5.3 RCWL-0516 connected to Arduino with output pin and Pin 12 on U1

Figure 5.4 Arduino output data seems the sensor has been detected a motion

Figure 5.5 Arduino plotter seems the sensor has been detected a motion

Figure 5.6 Moving Human hand on RCWL-0516 Arduino plot

Figure 5.7 Walking toward RCWL-0516 Arduino plot

Figure 5.8 Dropping wet sponge on RCWL-0516 Arduino plot

CHAPTER I

INTRODUCTION

Whether you realize it or not, we use radar sensors every day for different applications. Radar sensors are used for the weekly weather forecast, traffic control, and onboard vehicle systems. In recent years, they have been used for wearable devices, smart buildings, and autonomous cars. But what exactly is a radar sensor? Radar sensors are conversion devices that transform microwave echo signals into electrical signals. They use wireless sensing technology to detect motion by figuring out the object’s position, shape, motion characteristics, and motion trajectory. Unlike other sensors, radar sensors aren’t affected by light and darkness and with the ability to detect obstructions like glass, it can “see” through walls. When compared to other sensor technology, like ultrasound, radar can sense longer distances and is safe for people and animals. One of the biggest advantages radar sensors have over other sensors is its detection of motion and velocity. By detecting an object’s Doppler effect, or change in wave frequency, a radar sensor can compute that object’s speed along with its direction. They can also observe the target’s movement from different perspectives when using multi-channel sensors. Examining the movement from different perspectives in addition to previously collected measurements are used to determine complex movements.

CHAPTER II

Arduino

2.1 Definition

In a nutshell, an Arduino is an open hardware development board that can be used by tinkerers, hobbyists, and makers to design and build devices that interact with the real world. While Arduino refers to a specific type of board design, it can also be used to refer to a company which manufactures a specific implementation of these boards, and is typically also used to describe the community around compatible boards made by other people or companies which function in a similar way.

2.2 Board Description of Arduino

Arduinos contain a number of different parts and interfaces together on a single circuit board. The design has changed through the years, and some variations include other parts as well. But on a basic board, you're likely to find the following pieces:

- A number of pins, which are used to connect with various components you might want to use with the Arduino. These pins come in two varieties:
    - Digital pins, which can read and write a single state, on or off. Most Arduinos have 14 digital I/O pins.
    - Analog pins, which can read a range of values, and are useful for more fine-grained control. Most Arduinos have six of these analog pins.

These pins are arranged in a specific pattern, so that if you buy an add-on board designed to fit into them, typically called a "shield," it should fit into most Arduino-compatible devices easily.

- A power connector, which provides power to both the device itself, and provides a low voltage which can power connected components like LEDs and various sensors, provided their power needs are reasonably low. The power connector can connect to either an AC adapter or a small battery.
- A microcontroller, the primary chip, which allows you to program the Arduino in order for it to be able to execute commands and make decisions based on various input. The exact chip varies depending on what type of Arduino you buy, but they are generally Atmel controllers, usually a ATmega8,

ATmega168, ATmega328, ATmega1280, or ATmega2560. The differences between these chips are subtle, but the biggest difference a beginner will notice is the different amounts of onboard memory.

- A serial connector, which on most newer boards is implemented through a standard USB port. This connector allows you to communicate to the board from your computer, as well as load new programs onto the device. Often times Arduinos can also be powered through the USB port, removing the need for a separate power connection.
- A variety of other small components, like an oscillator and/or a voltage regulator, which provide important capabilities to the board, although you typically don't interact with these directly; just know that they are there.

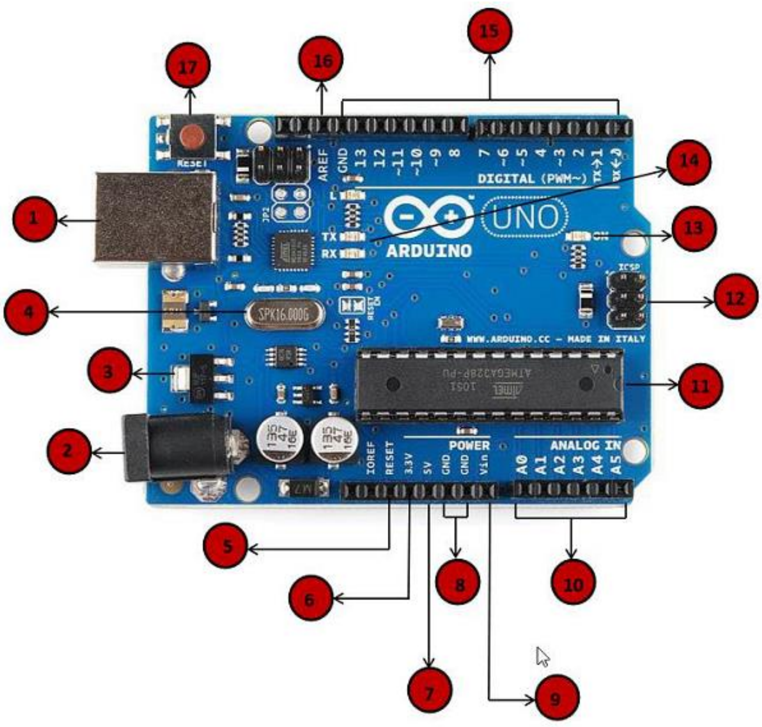

Figure 2.1 Arduino UNO Board

2.3 Arduino Programming

Most Arduino enthusiasts, especially when they are starting out, will choose to use the official integrated development environment (IDE) for the Arduino. The Arduino IDE is open-source software which is written in Java and will work on a variety of platforms: Windows, Mac, and Linux. The IDE enables you to write code in a special environment with syntax highlighting and other features which will make coding easier, and then easily load your code onto the device with a simple click of a button.

The code for Arduino is generally written in Wiring, which is based on the Processing programming language. For more on getting started with Arduino programming, visit the official documentation.

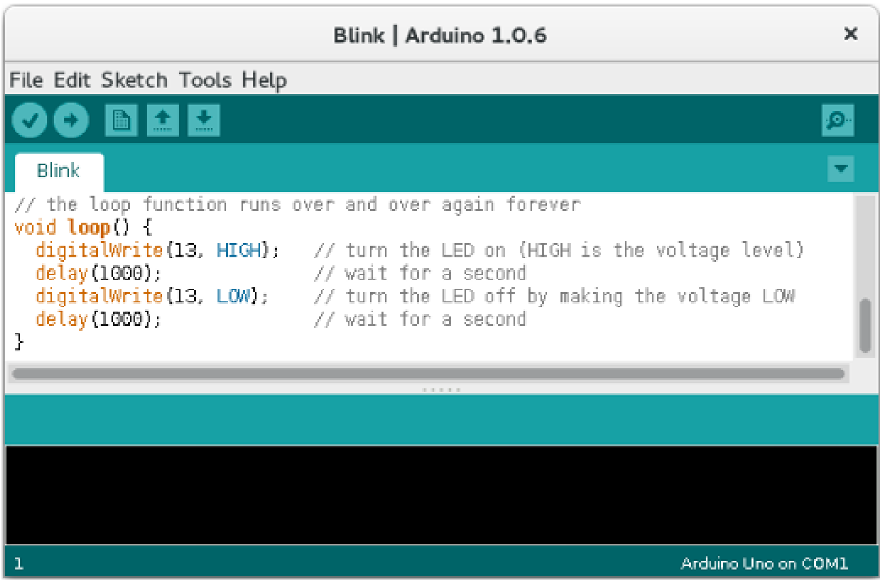

Figure 2.2 Arduino IDE 1.0.6 version

2.4 Arduino Sensor – Types and Applications

There are many projects designed using Arduino sensor for various applications. Arduino is said to be used to make a dream idea into reality. The ultrasonic module is used for non-contact range detection. It makes use of sonar for its working. IR Infrared obstacle avoidance sensor detects objects which are before it and generates a digital signal. It is used in robots. Soil hygrometer is a soil moisture sensor. It generates a digital signal when the moisture in the soil increases above some threshold value. The automatic self-watering plant is designed using this sensor with Arduino. Microscope sensor is used to detect the sound. It generates a signal when the intensity of detected sound increases beyond some threshold value. Digital Barometric pressure sensor is used to measure the absolute pressure of the environment. Hight of the robot or projectile can be measured using this sensor. For light detection, the Photoresistor sensor module is used. Night security light system uses this sensor with Arduino. The temperature sensor is used to detect ambient temperature. To detect the poisonous gases such as LPG, i- Butane, Propane, Alcohol, etc… MQ-2 gas sensor is used. The rain sensor is used for weather monitoring. To detect flame and the ordinary light Flame sensor is used. PIR sensor is used to detect motion from humans and pets.

Examples of Arduino Sensor

- HC- SR04 Ultrasonic Module
- IR Infrared Obstacle Avoidance sensor
- Soil Hygrometer Detection Module
- Soil Moisture Sensor
- Microphone sensor
- Digital Barometric Pressure Sensor
- Photoresistor sensor
- Digital thermal sensor – Temperature sensor
- Rotary Encoder Module
- MQ-2 Gas sensor
- SW-420 Motion sensor
- Humidity and Rain Detection sensor
- Passive Buzzer Module
- Speed sensor Module

In this project we are discussing radar sensors.

CHAPTER III

RADAR SENSORS WITH ARDUINO

3.1 Introduction of Radar System

Rader is an object detection system. It uses Microwaves to determine the range, altitude, direction, or speed of objects. The radar can transmit radio waves or microwaves which bounce off any object in their path. So, we can easily determine any object in the radar range. Arduino is a single-board microcontroller to make electronics more discipline. The radar system has different performance specifications and also it comes in a verity of size.

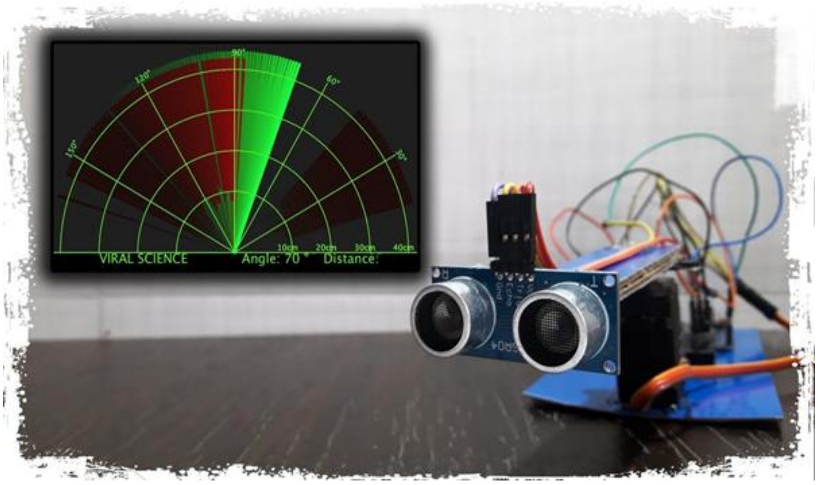

Figure 3.1 Arduino Radar Sensor

During World War II, many countries were made radar secretly. In 1940 US Navy acronym Radar word from Radio Detection and Ranging. The modern radar system is more advanced and the uses of radar are highly diverse. Such as Air traffic control, Air-defense system, Radar Astronomy, Antimissile system, Ocean Surveillance system, outer space surveillance and many more.

3.2 Arduino Radar Sensor Overview

The Arduino is an electronics platform which based on easy-to-use hardware and software. The Open-source Arduino Software (IDE) makes easy to write code and upload into the board. It can run in Windows, Mac, and Linux. The program is written in Java and based on processing software. This software can be used in any Arduino board. The program write on IDE is known as sketches. An Arduino radar project is more than a visual project because of its circuit implementation. There are different hardware use to accomplish the Arduino Radar Sensor. Like as, Arduino UNO. HC-SRo4 Ultrasonic Sensor including a Servo Motor. The main appearance is the visual narration in the Processing Application.

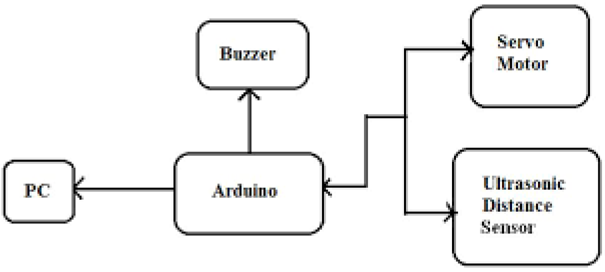

Figure 3.2 Block Diagram

You may ask how the processing application works here. It is very simple; the Ultrasonic Sensor collects the object information with the help of Arduino and passes it to Processing Application. In the processing application, there is a simple Graphics application implemented which mimic a radar screen.

3.3 Advantages of the Arduino Radar Sensor

1. It is not affected by color or transparency. Basically, the Ultrasonic Sensors transmit the sound off of the object, hence the color and transparency have no effect on the radar reading.

2. Any dark environments have no effect on this Arduino radar sensor's detection procedure. So, it can also use at night.

3. Easy to design and low price. The ultrasonic sensors are available at the market with very cheap price.

4. It has high frequency, high sensitivity, therefore, it can easily detect the external or deep objects.

5. This radar sensor is not affected by dust, rain, snow, and many more.

6. It has a self-cleaning system to continue running and less downtime.

7. The Arduino Radar Sensor is easy to use. Also, it is completely safe during the operation to nearby objects, human or equipment.

8. The Ultrasonic sensor can easily interface with any types of the microcontroller.

3.4 Disadvantages and Limitations:

1. The Arduino Radar Sensor conduct sound to continue the work. So, it is not working in a vacuum as there is no air for the sound to travel through.

2. A very soft fabric can absorb more sound waves. Therefore, it is hard to detect objects which are covered with soft fabric.

3. If temperature changes of 5 to 10 degree or more then it is the effect on the sensing accuracy. Although this is true that there have many more temperature compensated sensors available.

4. Another limitation is the detection range. This depends on which Ultrasonic sensor have used to make the Arduino Radar Sensor.

5. While the radar using for inspection purpose, make sure it should be water resistive. Otherwise highly chances of damage.

CHAPTER IV

HB100 SENSOR

4.1 Definition

HB Series of microwave motion sensor module are X-Band Mono-static DRO Doppler transceiver front-end module. These modules are designed for movement detection, like intruder alarms, occupancy modules and other innovative ideas. The module consists of Dielectric Resonator Oscillator (DRO), microwave mixer and patch antenna. This Application Note highlights some important points when designing-in HB100 module. Most of the points are also applicable to other models in this series.

Figure 4.1 HB100 module

4.2 Power Supply

The module operates at +5 Vdc for Continuous wave (CW) operation. The module can be powered by +5V low duty cycle pulsed trains in order to reduce its power consumption. Sample & Hold circuit at the IF output is required for pulse operation.

4.3 Transmit Frequency

The transmit frequency and power of the module is set by factory. There is no user adjustable part in this device. The module is a low power radio device (LPRD) or intended radiator. Local communication authority regulates use of such a device. Though user license may be exempted, type approval of equipment or other regulation compliance may be required.

4.4 Radiation Pattern

The module to be mounted with the antenna patches facing to the desired detection zone. The user may vary the orientation of the module to get the best coverage. The radiation patterns of the antenna and their half power beam width (HPBW) are shown in below diagram.

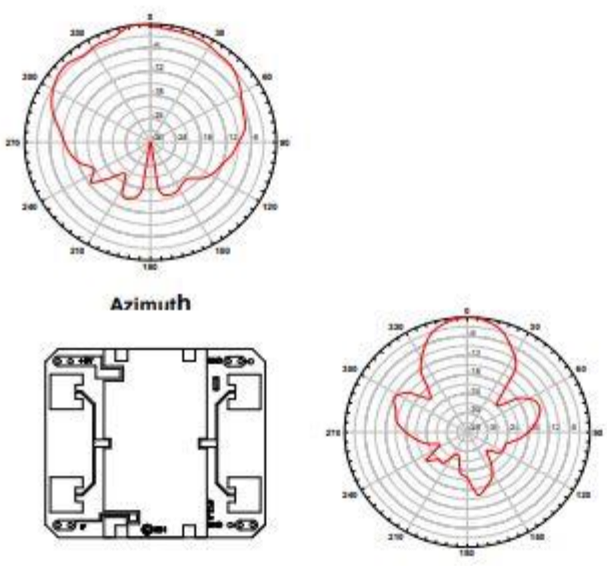

Figure 4.2 HB100 module Radiation Pattern

4.5 Output Signals

Doppler shift - Doppler shift output from IF terminal when movement is detected. The magnitude of the Doppler Shift is proportional to reflection of transmitted energy and is in the range of microvolts (μV). A high gain low frequency amplifier is usually connected to the IF terminal in order to amplify the Doppler shift to a processable level. Frequency of Doppler shift is proportional to velocity of motion. Typical human walking generates Doppler shift below 100 Hz. Doppler frequency can be calculated by Doppler equation. The Received Signal Strength (RSS) is the voltage measured of the Doppler shift at the IF output. The RSS figure specified in the technical data sheet is level of a 25 Hz Doppler shift, generate from the modulated microwave signal received at the received antenna, The received microwave signal is attenuated to 93 dB below the transmit microwave signal from the transmit antenna of the same unit. The 93dB loss is the total losses combining two ways free space loss (82.4 dB for 30 meters at 10.525 GHz), reflection less and absorption loss of the target, as well as other losses. This RSS figure can be view as an approximation of the output signal strength for a human at 15 meters away walking straight to the module at 1.28 km/hour. Reflection of a human body is varied with the size of the body, clothing, apparels and other environmental factors; RSS measured for two human bodies may vary by 50%. Circuit designer must take note the maximum and minimum Received Signal Strength (RSS) specified in technical data sheet, when designing the amplifier. Sensitive deviation between modules has to be considered when setting amplifier gain or alarm threshold. On-production-line gain adjustment may be necessary if a narrow window for triggering threshold is required. Noise - The noise figure specified in the technical data sheet is the noise measured in an Anechoic chamber, that shield the unit-under-test from external interference, as well as reflection from surfaces. Hence, the figure is only presenting the noise generated by the internal circuit itself. Other than noises generate from internal electronic circuit, in actual applications, other noises may be picked up from surrounding, or other part of the electronic circuit. Specially attention has to be given to the interference pick up from fluorescent light, as the 100/120 Hz noise is closed to the Doppler frequency generated by human movement On and off switching of certain devices (relay, LED, motor, etc.) may generated high magnitude of transient noise at the IF terminal. Careful PCB layout and time masking is necessary to prevent false triggering.

DC Level - DC level (0.01 to 0.2 Vdc) exists at the IF terminal and its polarity can be positive and V1.02 negative. Its magnitude may vary over temperature. AC coupling is recommended for IF terminal connection.

4.6 Sample with Arduino

A series of low-cost, specialized sensor modules that detect motion using this principle is the HB100 X-Band Radar sensor. On their own, these sensors are unusable due to the large amounts of noise and very low output voltage (around the order of microvolts).

Prior to processing by a micro-controller, a signal conditioning circuit must be used to both filter the required frequencies and amplify the signal to an appropriate level. After successful processing, you are left with a resultant signal with a frequency that corresponds to the measured Doppler frequency. Thus, the higher the Doppler frequency, the faster and more motion there is taking place in the targeted area.

The HB100 outputs a low-level voltage (few uV) whose frequency represents the speed at which an object is moving towards or away from the sensor. The output can be very noisy, so in addition to amplifying the signal, we need to filter out frequencies that don't match what we expect from ordinary objects.

4.7 Sample Code for Arduino with HB100

```
// Below: pin number for FOUT
#define PIN_NUMBER 4
// Below: number of samples for averaging
#define AVERAGE 4
// Below: define to use serial output with python script
//#define PYTHON_OUTPUT
unsigned int doppler_div = 19;
unsigned int samples[AVERAGE];
unsigned int x;

void setup() {
  Serial.begin(115200);
  pinMode(PIN_NUMBER, INPUT);
}

void loop() {
  noInterrupts();
  pulseIn(PIN_NUMBER, HIGH);
  unsigned int pulse_length = 0;
  for (x = 0; x < AVERAGE; x++)
  {
    pulse_length = pulseIn(PIN_NUMBER, HIGH);
    pulse_length += pulseIn(PIN_NUMBER, LOW);
    samples[x] =  pulse_length;
  }
  interrupts();

  // Check for consistency
  bool samples_ok = true;
  unsigned int nbPulsesTime = samples[0];
  for (x = 1; x < AVERAGE; x++)
  {
    nbPulsesTime += samples[x];
    if ((samples[x] > samples[0] * 2) || (samples[x] < samples[0] / 2))
    {
      samples_ok = false;
    }
  }

  if (samples_ok)
  {
    unsigned int Ttime = nbPulsesTime / AVERAGE;
    unsigned int Freq = 1000000 / Ttime;

    #ifdef PYTHON_OUTPUT
      Serial.write(Freq/doppler_div);
    #else
      //Serial.print(Ttime);
      Serial.print("\r\n");
      Serial.print(Freq);
      Serial.print("Hz : ");
```

```
      Serial.print(Freq/doppler_div);
      Serial.print("km/h\r\n");
    #endif
  }
  else
  {
    #ifndef PYTHON_OUTPUT
      Serial.print(".");
    #endif
  }
}
```

CHAPTER V

RCWL-0516 SENSOR

5.1 Definition

The RCWL-0516 is an active sensor. It emits microwaves with a frequency of ~3.18 GHz and analyzes the reflected radiation. To achieve this, it uses the Doppler effect that most of you certainly know. If not: this is the effect that causes the siren of an approaching ambulance to sound higher than when it is driving away from you. In astronomy, the effect is exploited by determining the direction of motion and speed of stars. If a star moves towards the Earth, its light is shifted to the blue area, i.e., more short-wave. Accordingly, the light is shifted to the red area, i.e., more long-wave as it moves away from the earth.

Light and sound spread as waves, that is at a certain frequency. If you move towards the source, you will be hit by more waves per unit of time. Therefore, the frequency appears increased. If you move away from the source, it is the other way around.

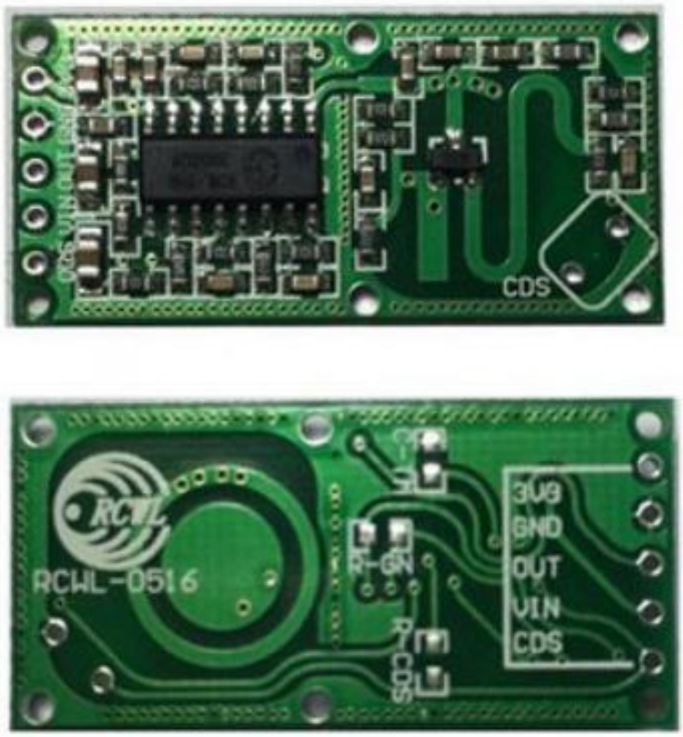

Figure 5.1 RCWL-0516 module

5.2 Module Description

The module is a single circuit board with the following connections:

Be certain that you note that the *3V3* pin is a 3.3-volt output, not a power supply input. The device has an integrated 3.3-volt voltage regulator which can provide up to 100 mA of current for powering external logic circuitry.

The *Output* pin is a 3.3-volt logic level pin. When the device detects a moving object it will send this pin HIGH for two seconds. The device does not measure the distance to the detected object or its velocity, just its presence.

The *VIN* pin is the positive power supply input, accepting any voltage from 4-volts to 28-volts. The RCWL-0516 does not consume very much current and can easily be powered by the 5-volt output from an Arduino or a Raspberry Pi.

The *CDS* pins are where you can attach an optional light sensor, a light dependent resistor (LDR). The LDR can actually be hooked up two ways:

- Using the two *CDS* pads on the top of the sensor printed circuit board.
- By connecting one end to the *CDS* pin on the main terminal section and the other end of the LDR to ground.

The light sensor will disable the device when it detects ambient light. You may also use the *CDS* pin as an Enable control for the module.

The RCWL-0516 comes without ant connecting pins attached. For my experiments I soldered a right-angled Dupont connector to the main pins, allowing the device to be mounted “standing up” on a solderless breadboard.

I also soldered a couple of pins to the CDS pads at the top of the device. If you decide to do this note that these pins don’t observe the standard 0.1-inch spacing common to most electronic components so you will need to use individual pins.

5.3 Schematic

The only schematic I could find is very low resolution and it's hard to make out some of the text. However, I've been reverse engineering it and adding my annotations. There are two parts to this schematic. A microwave frequency transmitter/receiver/mixer and a much lower frequency part based on an IC (marked RCWL-9196) which is very similar to the BISS0001 IC used in PIR motion detectors. First the microwave part: The best explanation of how the microwave part of this works is in patent EP3091605A1. It describes as similar type of module operating at 5.8GHz, At the heart of the RF is a Q1 a MMBR941M high frequency NPN transistor [5] in what is probably a Colpitts oscillator [6] configuration. The schematic above is misleading because it omits a key inductor and capacitors constructed from PCB traces (a microcline inductor and capacitor). The inductor is the S curve trace on the top surface and capacitors are the ring structure on the bottom surface and also the rectangular block to the left of the S curve. Using the formula at reference [12] below I calculate the inductance of the S curve to be (very approximately) 10nH. A critical function of a doppler radar is to be able to 'mix' the reflected signal with the transmitted signal to arrive at a frequency which is the difference between the transmitted and reflected signal. In this board Q1 also cleverly assumes the function of the mixer: [TODO: this really needs to be explained]. The low doppler frequency difference is extracted by a low pass RC filter (C9 1nF, R3 1k, fc $1/2\pi RC \approx 160$kHz) and amplified by the RCWL9196 IC and treated exactly the same as a signal from a PIR sensor. Update 4 Jan 2017: finally found the signal at 3.181GHz with the HackRF One SDR! One interesting observation: waving my hand in front of the sensor causes significant changes in the transmitting frequency, shifting by up to 1MHz. My theory: the low frequency doppler shift causes small changes in the transistor base bias.

Doppler effect calculations

If ft is the transmitted frequency, fr is the reflected frequency (as measured by the common transmit/receive antenna on the sensor), v is the speed of the target relative to the sensor (negative if receding, positive if advancing toward sensor), c is the speed of light and fd (fr-fd) is the doppler shift, then:

$fr = ft (c + v) / (c – v)$

$fd = fr - ft\ 2v\ ft / (c – v)$

If ($c << v$) then $fd \approx 2v\ ft / c$

Assume typical human motion speed of v 1 m/s. ft 3.181GHz, c 2.998E8 m/s, then fd 10Hz.

```
void setup() {
pinMode(PA0,INPUT);
pinMode(PA1,INPUT);
}
void loop() {
Serial.print("0,");
Serial.print(analogRead(PA0));
Serial.print(",");
Serial.print(analogRead(PA1)+100);
Serial.println(",4500");
delay(10);
}
```

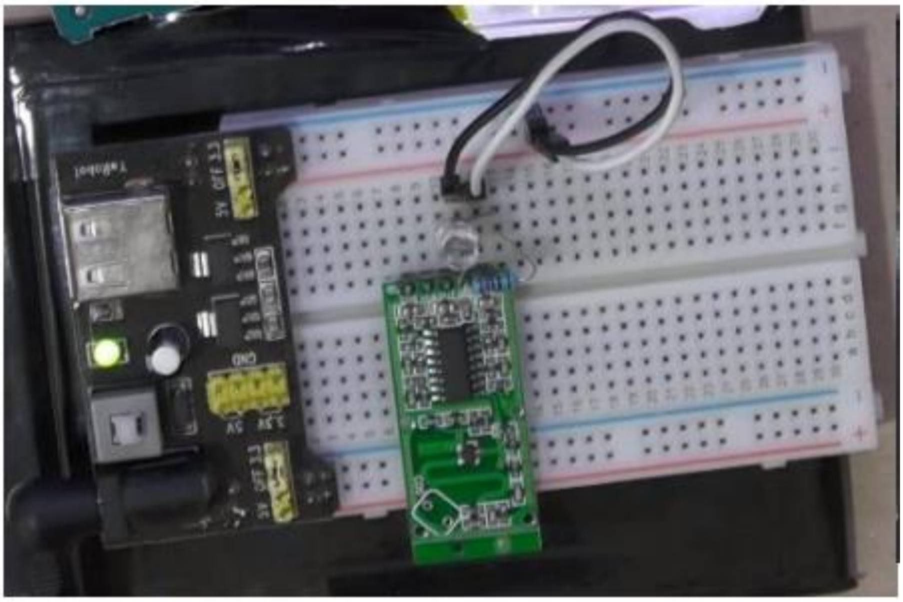

Figure 5.2 RCWL-0516 module with Arduino Board

5.4 Sample with Arduino

The main difference in my approach is that I have connected a wire from the analogue signal output on the only IC on the board, into an analogue input on a STM32F103C8 (aka Blue Pill) board.

The pin I connected to is Pin 12 on U1, (RCWL 9196 and is the output of the 2nd OpAmp in that chip).

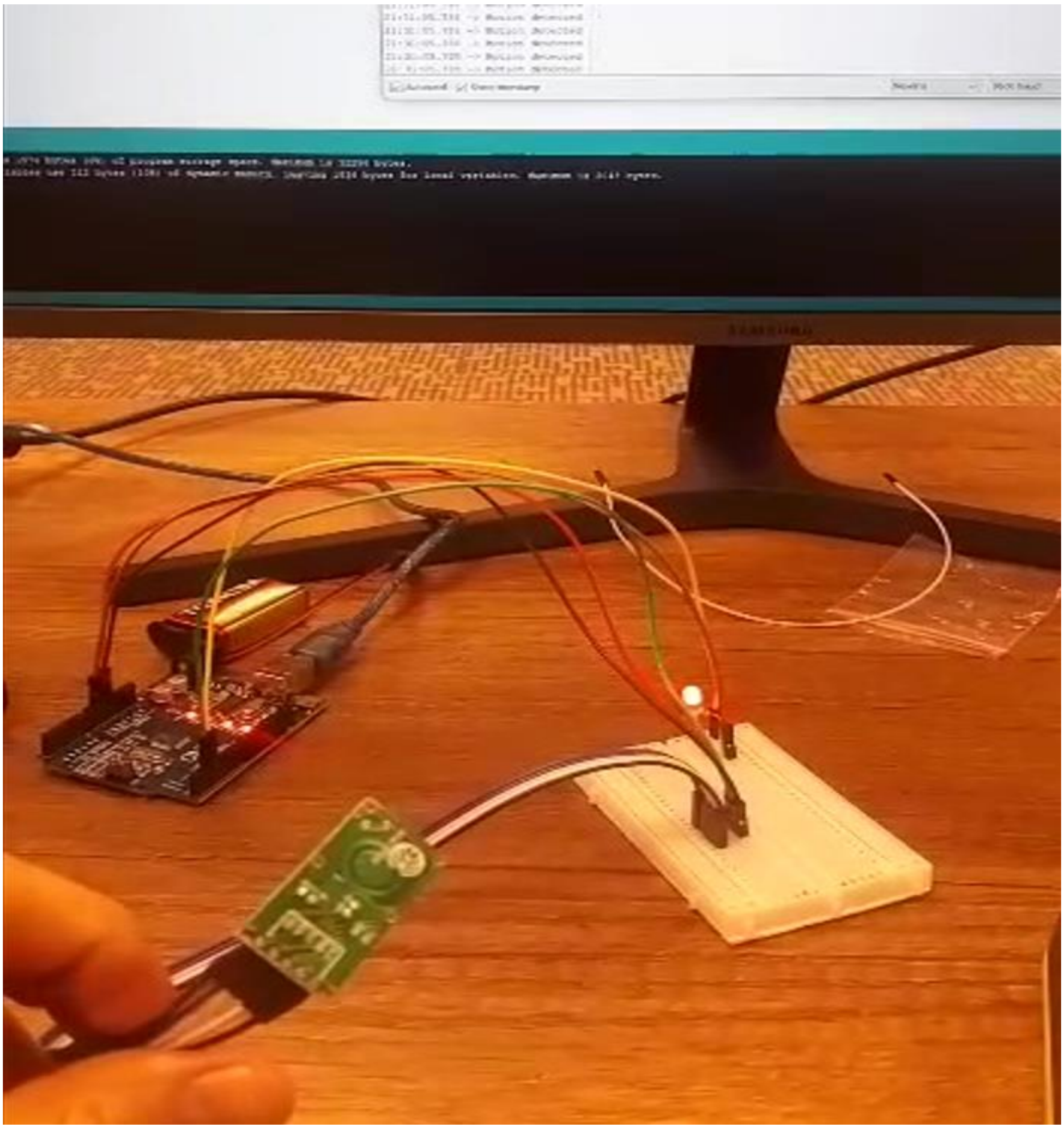

Figure 5.3 RCWL-0516 connected to Arduino with output pin and Pin 12 on U1

The code for Arduino with LED

```
#define Sensor 2
#define LED 3


int micro_read = 2;
volatile int iter = 0;


void setup() {

 pinMode(micro_read,INPUT);

pinMode(Sensor,INPUT);
pinMode(LED,OUTPUT);
Serial.begin(9600);ar



}

int rv = -1;

void loop() {

  // // start section for frequency 1

  int v = digitalRead(Sensor);
  if ( v != rv) {
    rv = v;

    char msg[20];

    sprintf(msg, "R %lu %d", millis() / 1000, v );

    Serial.println(msg);


  }

  delay(100);




  bool Detection = digitalRead(Sensor);

  if(Detection == HIGH){
```

```
    Serial.println("Motion detected !!");
    digitalWrite(LED,HIGH);
  }

  if(Detection == LOW) {
    Serial.println("Clear");
    digitalWrite(LED,LOW);
  }
}
```

The Code for Arduino with analog input and connected pin 12 on the U1

```
#define PA0 A0 // pin 12 on the transistor
#define PA1 A1 // output pin on the sensor

double sum = 0;
int count = 0;

float fd;   // Frequenz
float v;   // Geschwindigkeit
float rVal;   // ReadingValue


void setup() {
  Serial.begin(115200);
pinMode(PA0,INPUT);
pinMode(PA1,INPUT);
}

void loop() {
  Serial.print("0,");
  Serial.print(analogRead(PA0)); // to close to the zero
  Serial.print(",");
  Serial.print(analogRead(PA1));
  Serial.println(",4800");
  delay(100);

}
```

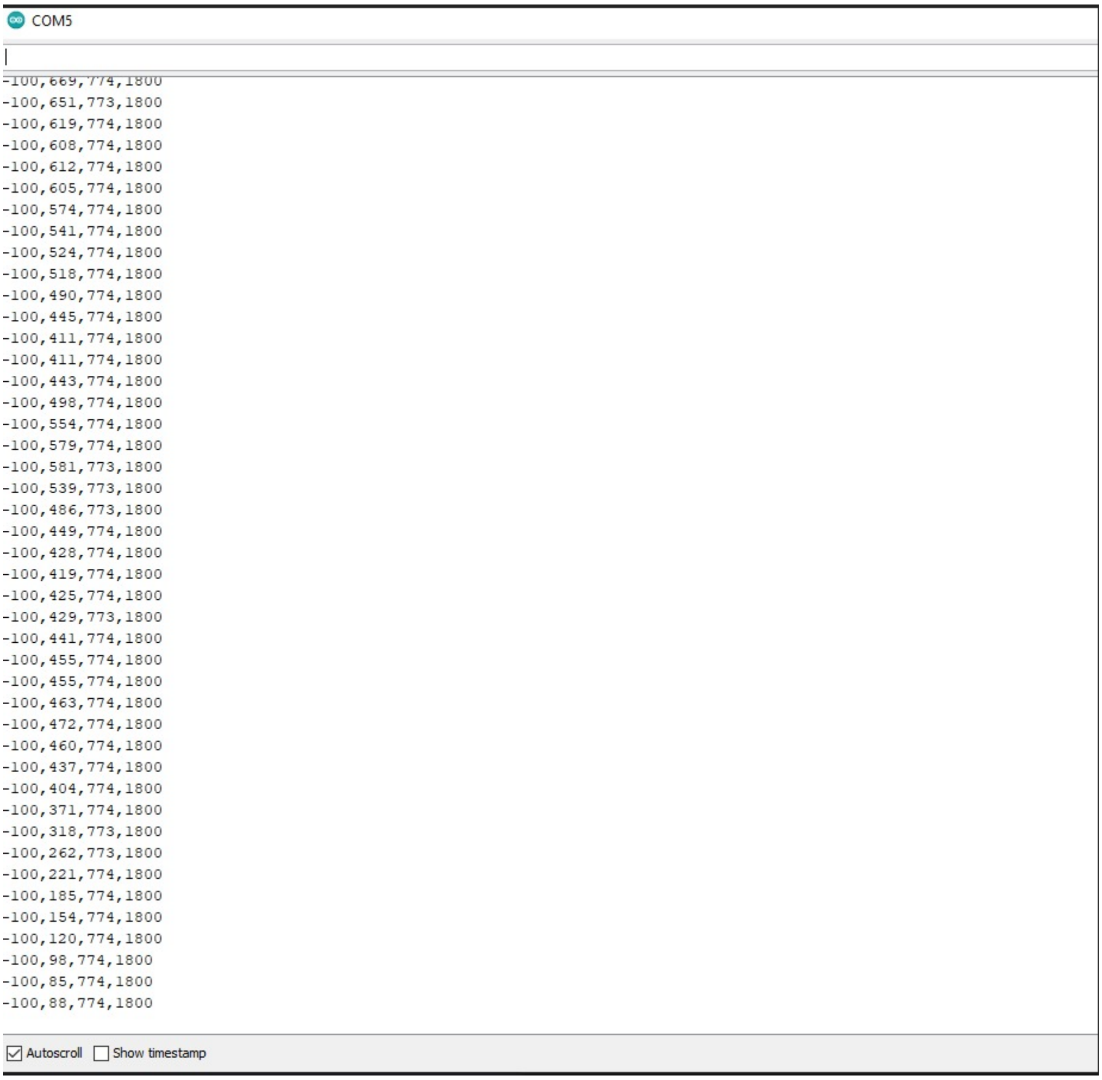

Figure 5.4 Arduino output data seems the sensor has been detected a motion

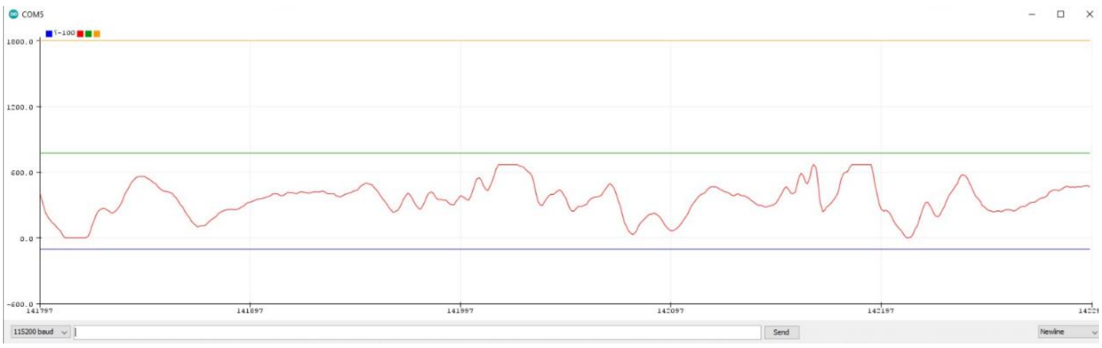

Figure 5.5 Arduino plotter seems the sensor has been detected a motion

If someone start to wave hand, starting with a small movement and then increasing, we see this plot.

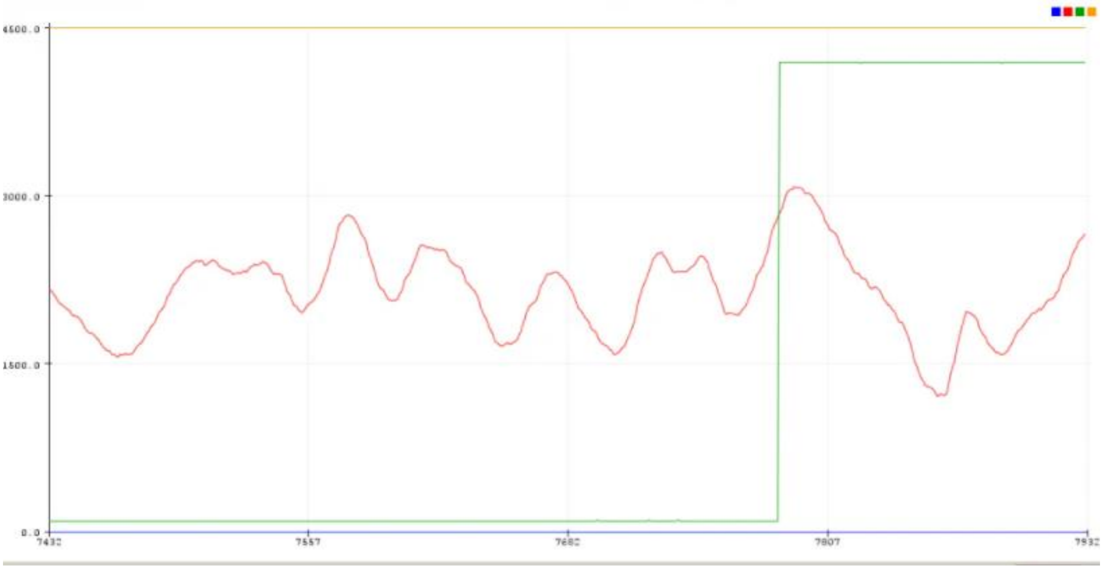

Figure 5.6 Moving Human hand on RCWL-0516 Arduino plot

As we can see the hand movement causes an immediate effect on the output of the second OpAmp, but the value has to go above (or below) a threshold before the digital output triggers.

In this next plot. I was walking towards the sensor, and then stopped.

Figure 5.7 Walking toward RCWL-0516 Arduino plot

As we can see the digital output holds for around 2 seconds after the end of the input fluctuations.

The more interesting thing about this plot are the peaks and troughs at the start where someone was walking towards the sensor.

We thought this could be because it's sensing the movement of my legs, so we devised a better test where someone stood on a chair and dropped an object from the ceiling, and observed the results.

We tried a variety of objects, and the best performing object that mimicked the properties of a person moving, turned out to be a damp sponge.

We tried a dry sponge, but it had no effect on the sensor whatsoever, but as soon as we got it wet, it was immediately detected.

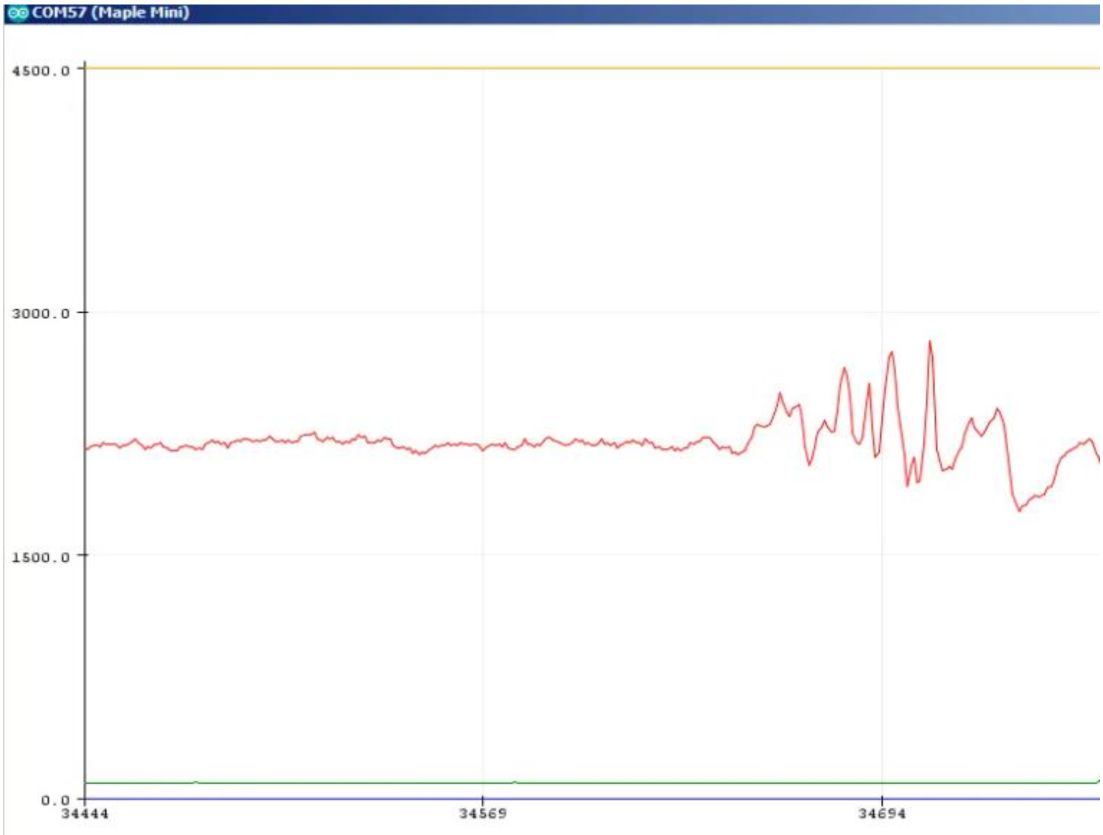

Figure 5.8 Dropping wet sponge on RCWL-0516 Arduino plot

In this plot the sponge is falling about 1m horizontally from the sensor, which is resting on a table about 1m from the ground.

As we can see, there are still multiple peaks and troughs in the output signal.

We know that there was some speculation that these devices work by using the Doppler effect, but this does not appear to be validated the tests I have done.

So, our theory is that the peaks and troughs are caused by reflected signals from objects, interfering with the oscillator / transmitter.
When we say interfering, we mean of wave propagation, where the reflected wave can be in phase with the oscillator or out of phase, resulting in the oscillator drawing more or less current.

From Joe Desbonnet's github repo, he observed that the oscillation frequency is around 3.1GHz with his module. So, assuming my module is similar, then the wavelength of 3.1GHz is around 10cm.

However, in my tests, I am not sure the peaks and troughs match exactly with that frequency, and the effect I see if more consistent with perhaps twice that frequency.

CHAPTER VI

CONCLUSION

Whether you realize it or not, we use radar sensors every day for different applications. Radar sensors are used for the weekly weather forecast, traffic control, and onboard vehicle systems. In recent years, they have been used for wearable devices, smart buildings, and autonomous cars. But what exactly is a radar sensor? Radar sensors are conversion devices that transform microwave echo signals into electrical signals. They use wireless sensing technology to detect motion by figuring out the object's position, shape, motion characteristics, and motion trajectory. Unlike other sensors, radar sensors aren't affected by light and darkness and with the ability to detect obstructions like glass, it can "see" through walls. When compared to other sensor technology, like ultrasound, but seems the sensors that uses doppler effect technology is the best way to measure the distance and the velocity.

By achieving this project, this system essentially helps the manager assistants who spending hours on this duty, even saving teachers of humanin mistakes.

Futur developments may be expended as counting students joining a school trip or any other activity.

REFERENCES

1. https://www.arduino.cc/
2. https://www.filipeflop.com/img/files/download/Datasheet_rcwl-0516.pdf
3. https://www.rogerclark.net/investigating-a-rcwl-9196-rcwl-0516-radar-motion-detector-modules/
4. https://www.youtube.com/watch?v=IJoPkKlxFXA&ab_channel=DroneBotWorkshop
5. https://www.limpkin.fr/public/HB100/HB100_Microwave_Sensor_Module_Datasheet.pdf
6. https://www.electronicdesign.com/industrial-automation/article/21140744/electronic-design-radar-as-a-sensor-who-would-have-thought
7. https://dronebotworkshop.com/rcwl-0516-experiments/
8. https://medium.com/@shambadityamukherjee/arduino-radar-sensor-working-principle-advantages-disadvantages-894ee9060221
9. https://www.ijnrd.org/papers/IJNRD1804002.pdf