Linnæus University
Sweden

Master thesis

New technology for smart traffic lights based on the Arduino microcontroller

Author: Al anbagi Bassam
Supervisor: Sven-Erik Sandström
Examiner: Sven-Erik Sandström
Term: VT 2022
Subject: Electrical Engineering
Level: Master
Course code: 5ED36E

Linnæus University
Sweden

Linnæus University
Sweden

Abstract

Society has changed immensely due to the revolution in information technology. The question is how this technology can be useful to man. My interest lies in how to apply radar to traffic. People in general relate radar to military matters while the focus here is on traffic accidents. Accidents happen and therefore police and ambulance need to be on site as fast as possible.

This master thesis presents a new design with hardware and software combined in one device. The combination is essential in traffic technology. I designed a new and smart traffic light that can be effective in a traffic system. I design, discuss and examine the radar up to the laboratory stage.

It is based on the Doppler effect and has a sound sensor that can identify a police car and also detect vehicle speed. The status of the traffic lights can then be changed to give priority to the police car, based on speed limits and other restrictions. Difficulties as well as advantages with the design are discussed.

Key words

Arduino, Doppler effect, RF receiver

Linnæus University
Sweden

Table of contents

Chapter 1		7
1.1	History of traffic lights	7
1.2	Why smart traffic lights	7
1.3	Methodology	8
Chapter 2		9
2.1	Arduino usage and characteristics	9
2.2	Arduino uno specifications	9
2.3	Features	10
2.4	Arduino uno views	10
Chapter 3		12
3.1	Radar systems	12
3.2	Radar range	13
3.3	Arduino radar	13
3.4	Doppler radar	15
3.5	MH-ET Live HB 100 microwave sensor.	15
3.5.1	1MH-ET Live HB 100 figure	15
3.5.2	MH-ET Live HB 100 microwave parameters	16
3.5.3	MH-ET Live HB 100 microwave sensor characteristics	16
3.6	HB 100 Connection to Arduino uno	17
3.7	Frequency variation	17
Chapter 4		20
4.1	RF receiver / Amplifier	20
4.2	The photocell	20
4.3	The amplifier	21
4.4	TSB 712 OP AMP	21
4.4.1	TSB 712 features	21
4.5	Operational amplifiers	22
Chapter 5		23
5.1	The behavior of sound waves.	24
5.2	Sound wave characteristics	24
5.3	Ranges and units for sound	24
Chapter 6		26
6.1 Prototype design		26
6.2 Prototype usage		27
6.3 Prototype forming		27
6.4 Prototype work & connection		29
6.5 Conclusion		29

References
Appendices

Abbreviation	Explanation
AI	Artificial Intelligence
MC	Microcontroller
RF	Radio Frequency
IC	Integrated circuit
Mic	microphone
Op Amp	Operational Amplifier
USB	Universal Serial Bus
SPL	Sound Pressure Level
SPWL	Sound power level
IL	Intensity Level
GND	Ground
LED	Light Emitting Diode
VCC	Voltage Constant Current
VDC	Voltage Direct Current
V IN	Input Voltage
V OUT	Output Voltage
VS	Source Voltage
C	Light Speed
T	Time
D	Distance
dB	Decibel
KB	Kilo Byte
MHz	Mega Hertz
SRAM	Static Random-Access Memory
EEPROM	Electrical Erasable programable Read Only Memory
FD	Doppler Frequency
mA	Milli Ampere
Ft	Transmitted Frequency
Fr	Reflected Frequency
LPRD	Low Power Radio Device
LDR	Light Dependent Resister
CDS	Cadmium Sulfide
IDE	Integrated Development Environment

.

List of figures

Figure 1	Arduino pin diagram
Figure 2a	Front side Arduino uno MC
Figure 2b	Back side Arduino uno MC
Figure 3	Main parts of radar
Figure 4	Ultrasonic detector
Figure 5a	Light detector by using photocell and Arduino uno MC
Figure 5b	Schematic design for light detector in the form of a photocell and an Arduino uno MC
Figure 6	Front and back figure for MH-ET live HB 100 microwave sensor
Figure 7	Arduino uno with HB 100 sensor
Figure 8	block diagram for HB 100
Figure 9	Antenna beam pattern
Figure 10	Photocell
Figure 11	Block diagram TSB 712
Figure 12	Operational amplifier circuit diagram
Figure 13	Sound waves
Figure 14	Wave compression and diffraction
Figure 15	System flow diagram
Figure 16	System design

Chapter 1

Introduction

1.1 History of traffic lights

The fist traffic light was born in December 1868 which was installed at parliament square in London, that was exploded after just two months. While the first mechanical traffic light using electricity was installed in 1923 in Paris, increasingly stricter regulation of traffic between 1950 -1980 creating a need of traffic lights.[2]

This master`s thesis investigates solutions to one of the main problems at intersections. Accidents occur there every day all over the world. Some of these accidents are caused by police cars or ambulances on duty and often in intense traffic. As an example, the US had more than 6.3 million traffic accidents reported in 2015. According to CDC (Centers for Disease Control and Prevention, a U.S. organization) about 2700 teens aged 16-19 were killed by traffic accidents in the United States. It is important to devise methods and technology that can prevent such accidents and save many lives. That is why I designed a device to accurately control traffic lights. [4],[8],[9],[10].

This design is using Arduino Uno MC, Doppler HB100 and a sound sensor. We could also add a light sensor which can detect the police lights. There is more than one way to implement the idea, some of them will just be mention on this master thesis and can be used in this or more complicated design. Also, will view some of the Arduino applications like Arduino motion sensor just to clarify the idea concept.

The use of a speed camera for detecting speeding vehicles was introduced already in 1975.[13][15], but this thesis discusses a new idea.

1.2 Why smart traffic lights

The aim is to implement a new Arduino design, with components and circuit boards. It has the form of a simple radar receiver to police vehicles, ambulances and fire trucks. This is a method to give right of way at intersections.

An important question is whether Arduino can detect these vehicles with sufficient accuracy. There are also some challenges that can affect this design like:

- Cyclists
- Pedestrians
- Other passengers

1.3 Methodology

The task is then to design a smart traffic light system that can streamline the traffic and prevent accidents. Through theoretical studies and testing in the lab we hope to achieve this.
Three possible alternatives are radar technology, sound equipment and light sensing. Although we can use an RF amplifier which will not be used in this design.
Doppler radar senses the speed of a service car and gives priority. The second method is to use sound sensor, amplifier that identifies the siren signal and can be used to give priority accordingly. The third method is to use light sensing with an optical detector used in a similar way.

The solution is obtained by connecting an Arduino uno microcontroller to a sound detector, Doppler detector, or RF receiver. A small prototype was tested in the laboratory with good results.

Chapter 2
Arduino

In this chapter we are going to explain what Arduino is, its usage and the range of applications.

2.1 Arduino usage and characteristics

Arduino is an open-source microcontroller that could stand alone or be coupled to a computer via a USB cable to run it. Arduino runs with Arduino programs that are $c++$ programs modified for the microcontroller. This is a reason to use this microcontroller. Arduino is easy to use, and the programs are available in a large special library and can be downloaded for free from the Arduino website.

Special examples are also available. There is a large special library for Arduino that deals with these programs and how to connect this microcontroller to other electronic parts and components. Arduino programs are compatible to a lot of operating systems such as Windows, Linux, MacOS.
Arduino can be connected to a wide range of sensors and can be used for servo motors, alarm systems, lights, and cameras.
This makes Arduino one of the most useful microcontrollers for a wide range of uses with low cost and high capacity.[18]

2.2 Arduino uno specifications

The Arduino uno is a single board microcontroller which makes the interactive environments more accessible. Arduino uno truth table:

Microcontroller	ATmega328
Operating voltage	5 V
Input voltage (recommended)	7-12 V
Input Voltage limits (not recommended)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
DC Current for I/O Pin	40 mA
Flash Memory	32 kB (ATmega328) of which 0.5 kB used by boot-loader
SRAM	2 kB (ATmega328)
EEPROM	1kB (ATmega328)
Clock Speed	16 MHz
DC Current for 3.3 V pin	50mA
Analog input pins	6

2.3 Features

The features of the microcontroller are shown in fig. 1.

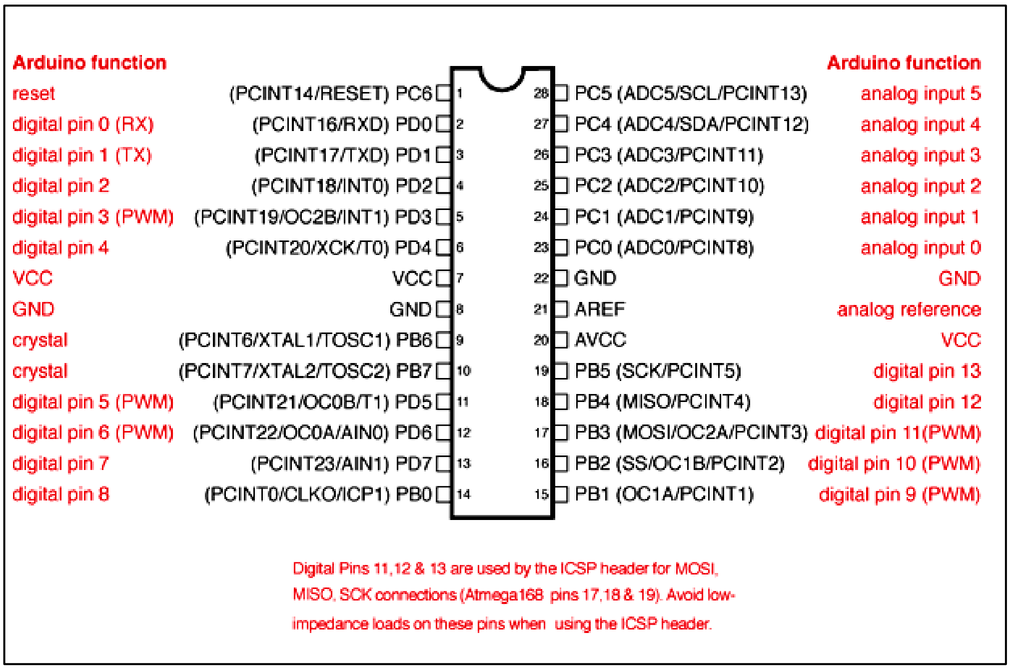


Figure 1: Arduino pin diagram.

2.4 Arduino uno views

Arduino uno is microcontroller board based on AT mega 328p, it has 14 digital input/output pins (6 of them can be use as PWM output pins). 6 analog pins for input, 16 MHz clock, power jack, USB connector, reset button, AC to DC adapter or battery and an ICSP header. We can see the front and back sides of the Arduino uno in figures 2a and 2b bellow.[18].

Figure 2.a: The front side of the Arduino Uno microcontroller.

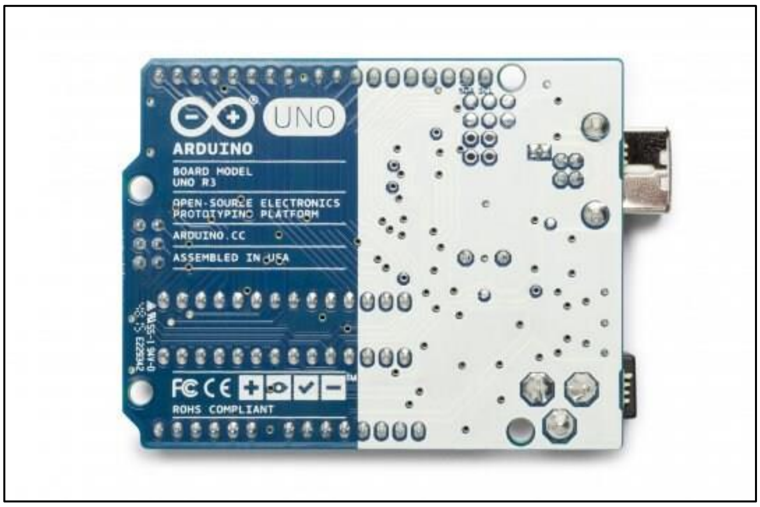


Figure 2.b: The back side of the Arduino Uno microcontroller.

Chapter 3

Radar

3.1 Radar systems

A technique that has been used for a long time is Radar. The word means radio detection and ranging. Radar can detect moving or stationary objects and distance to the objects. Radar has many applications at sea, on land and in space. It is useful for air traffic, ship safety and a host of military applications.
It gradually became necessary not only to detect the objects but also to identify their speed, shape and direction of movement.

Radar consists of a transmitter, a receiver and at least one antenna. The transmitter and receiver typically use the same antenna to transmit and receive the signal. The received signals are a reflection of the transmitted signal that is created when the wave hits the object. .[11][12]. Figure 3 below shows the main parts of the radar

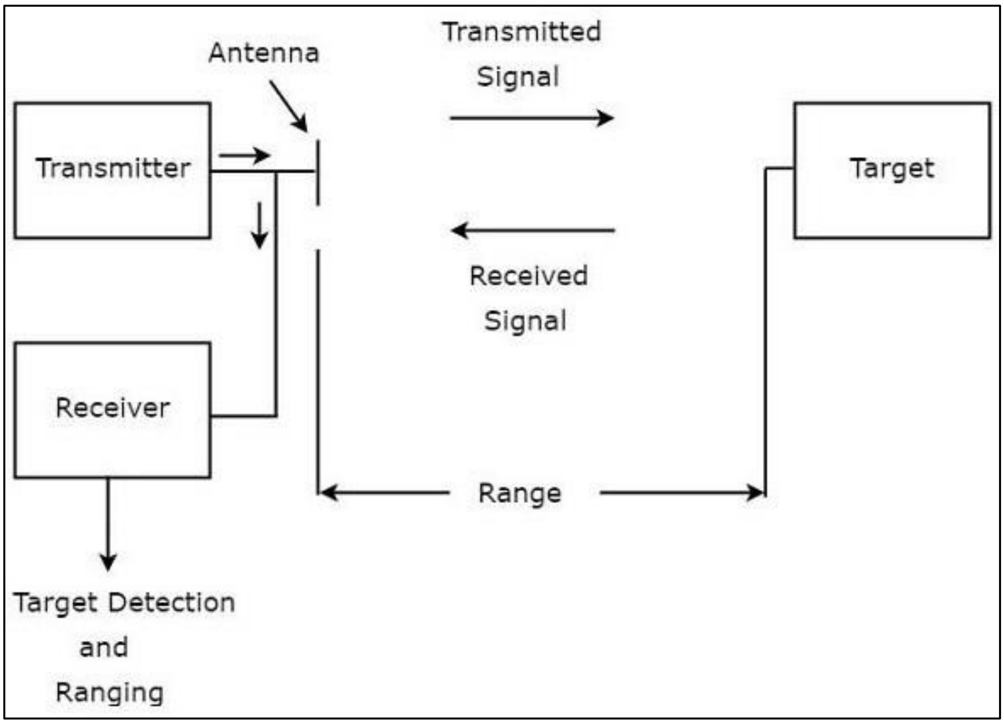


Figure 3: Main parts of the radar.

3.2 Radar range

The radar range is the distance between the object and the radar. The radar transmits a signal and receives an echo signal from the object, with the total distance 2R. Assuming wave propagation with the speed of light we can calculate the range as below:

Then the range as: $R = cT/2.$

The first radar was invented by Christian Huelsmeyer in Duesseldorf, Germany. In 1904 he used an electrical wave from a transceiver to detect objects at a range of 300 m. He obtained patent No.810,150 dated jan.16,1906.[11][15]

3.3 Arduino radar

One important Arduino application is radar or Arduino motion detector. This could be arranged by adding components to an Arduino microcontroller. There could be photocells, laser beam transceiver, RF transceiver, ultrasonic equipment. In this master thesis two types of sensors are used, the sound and the Doppler sensor.

The principle of radar or motion detector is shown in Fig.3 but when the idea is applied to ultrasound one finds that the range is no more than 3 meters which excludes the concept.[3] The Hc–sr 04 is a one of the typical ultrasonic detectors that can be connected to Arduino as shown in figure 4.

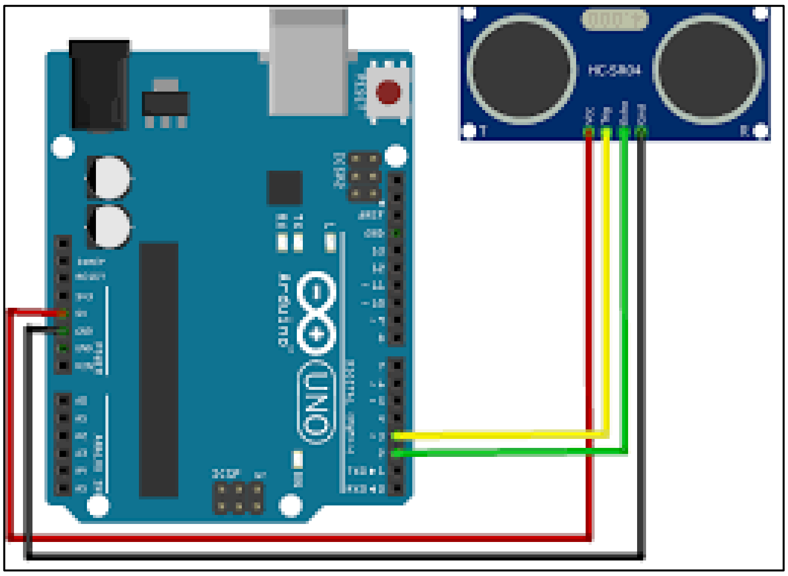


Figure 4 Ultrasonic detector.

For long range detection one may have to use an RF. The ultrasonic detector has two mechanisms, detection, and distance measurement. The use of a photocell to detect the police alarm light for identification requires a system of the type shown in Fig. 5.

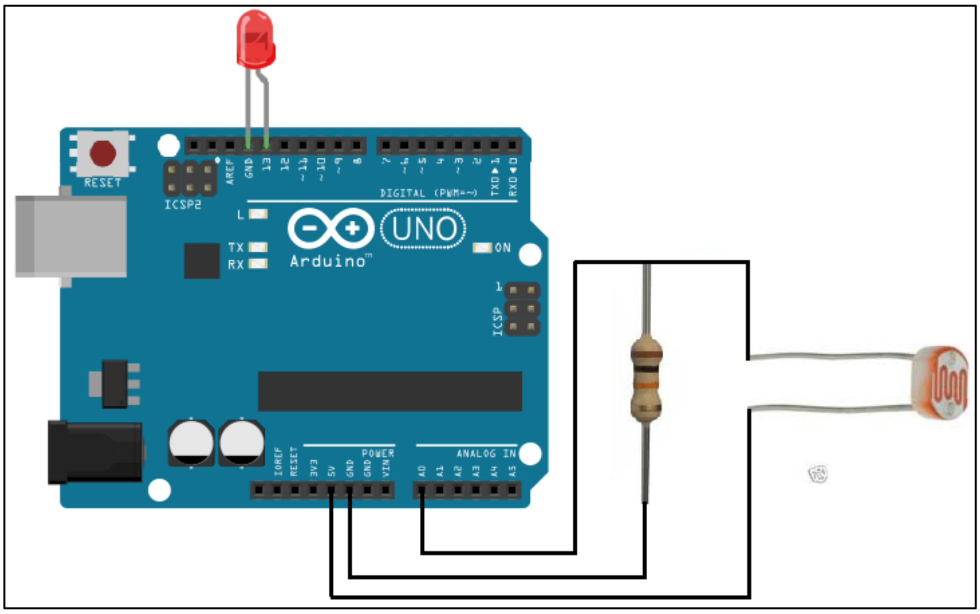


Figure 5a: Light detector by using photocell and Arduino Uno microcontroller.

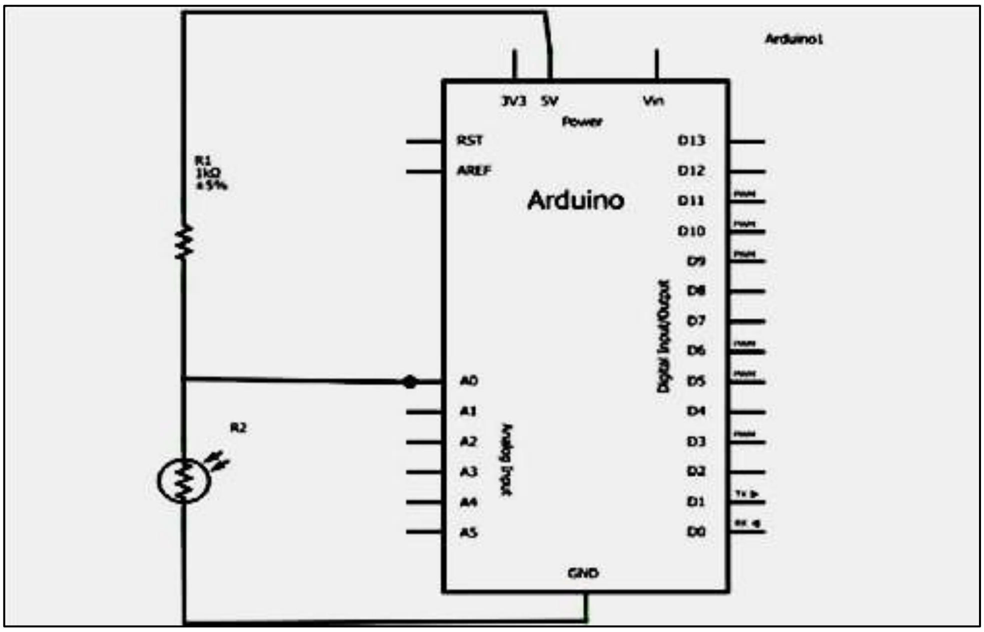


Figure 5b: Schematic design for a light detector in the form of a photocell and an Arduino microcontroller.

3.4 Doppler radar

The term ‘Doppler’ comes from the inventor, the Austrian physicist Christian Doppler. The Doppler shift is the change in frequency that is caused by the relative movement of transmitter and receiver. In other words, when the source of the waves is moving towards the observer, the wave fronts are observed at a higher rate so the frequency increases.

The arriving waves bunch together, i.e., there is less separation between the wave fonts. If the detected objects are moving away from the source the interval between the successive waves will be larger and hence the frequency will be lower.[11]

3.5 MH-ET Live HB 100 microwave sensor.

The HB 100 Microwave sensor is a simple X-Band bi-static Doppler transceiver. This is a microwave moving objects detector and contains a double microstrip patch array and a dielectric resonator oscillator. It is compatible to the Arduino micro controller which makes it perfect for the design. Although it is limited to a range of 16 meters, the low cost makes it one of the best choices for this prototype.

3.5.1 1MH-ET Live HB 100 figure

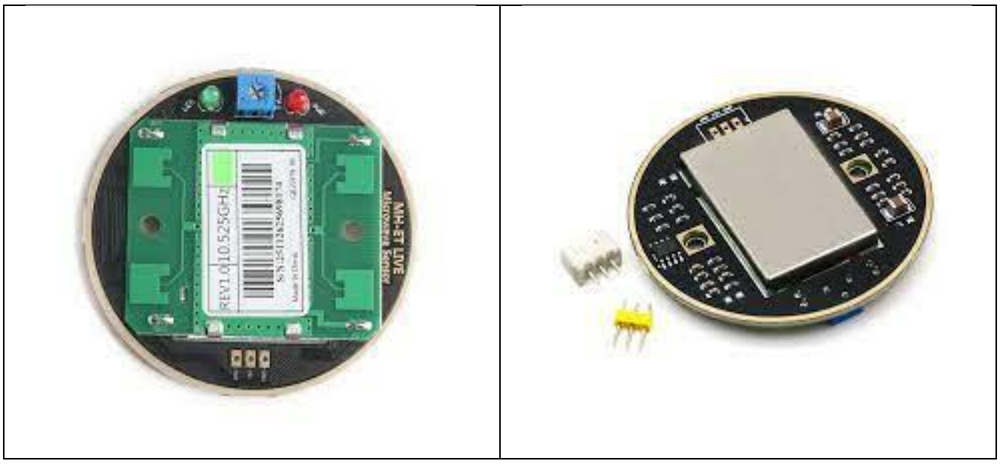


Figure 6 Front and back view for MH-ET live HB 100 microwave sensor.

3.5.2 MH-ET Live HB 100 microwave parameters

Working voltage	5V +/- 0.25 V
Operating current (CW)	50 mA max, 30 mA typical
Size	R=30.6 mm

Emission parameters:

Frequency setting accuracy	3 MHz
Output power (minimum)	13 dBm EIRP
Detection distance	2- 16 meter continuously adjustable
Harmonic emission	< -10 dBm
Average current (5% DC)	2 mA type
Pulse width	5mv
Duty cycle	1 %

Receive parameters:

Bandwidth	10.3 Hz – 80 Hz Bandwidth Clutter 10 mv
Antenna gain	8 dB
Sensitivity	10 dB S/N ratio for 3 Hz – 80 Hz Bandwidth
Vertical 3 dB beam width	36 degrees

3.5.3 MH-ET Live HB 100 microwave sensor characteristics

The detection method in HB 100 has advantages:

- Non-contact detection.
- Detection distance.
- Anti- radio frequency interference ability.
- The output power is small and causes no harm to the body structure.
- Not affected by temperature, air flow, humidity, noise, dust, light, etc.
- Supports the detection of inanimate objects.
- The directivity of the antenna is very good.

3.6 HB 100 Connection to Arduino uno

One of the main advantages of using the HB 100 sensor is the simple connection to the Arduino uno which is the main part of this project. This sensor is one of a few sensors that is based on the Doppler effect for a high-speed target. The range of the HB 100 in relation to the low price makes it fair enough to use it. Other sensors like XB 100 have longer range but are much more expensive.
In figure 7 we can see the how to connect the HB 100 to the Arduino Uno MC. [18]

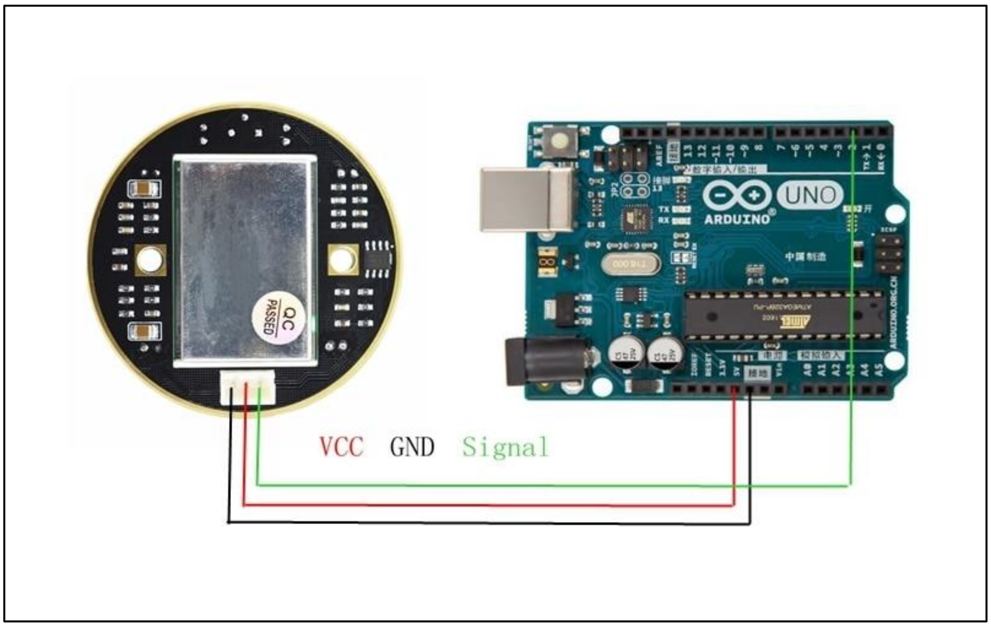


Figure 7: Arduino Uno with HB 100 sensor.

3.7 Frequency variation

The Doppler frequency can be measured by calculating the difference between the transmitted and reflected wave frequency. Assume that the source and the moving target are aligned and that v_{source} and v be the relative velocity and the speed of the source, respectively.
To obtain f-observed, let v be the velocity of the target and v_{source} the speed of source.
Then for a receding source:

$$f_{observed} = \left[\frac{v}{v + v_{source}}\right] f_{source}$$

And for an approaching source:

$$f_{observed} = \left[\frac{v}{v - v_{source}}\right] f_{source}$$

When the reflected wave arrives at an angle to the transmitted wave the doppler frequency is given by the cosine of the angle between the directions.

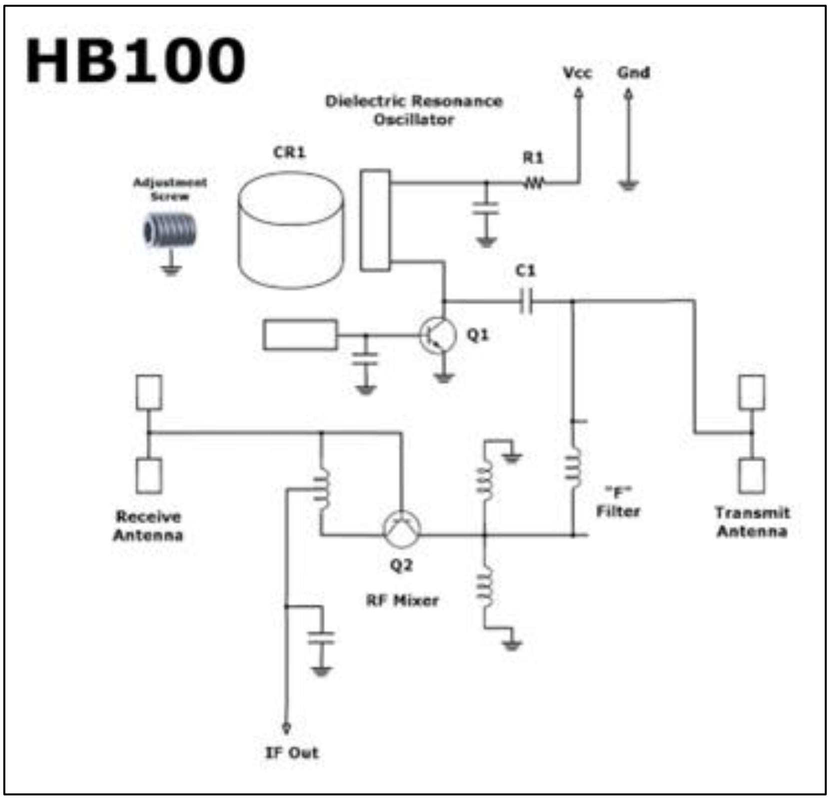


Figure 8**.** Block diagram for HB 100.

We can see from figure 8 that the HB 100 has a transmitting antenna connected to a dielectric resonance oscillator. The antenna radiates towards the object of interest. A receiving antenna senses the reflected wave, and the two waves are mixed in the RF mixer to produce the Doppler frequency. The doppler frequency is the difference between the transmitted and the received frequency.

The HB100 is a type of the Doppler sensor that has a low power radio device (LPRD) that is non-adjustable with an oscillator that produces a sinusoidal wave with a frequency up to 10.525 GHz. The antenna should face the direction of detection or can be oriented to get the best coverage.

The magnitude of the Doppler signal is proportional to the transmitted and received signal strength which is just a few millivolts. This received signal strength (RSS) is the voltage that we get on the IF terminal of the sensor. To process this signal with Arduino we often need to use an amplifier to get the level suitable for Arduino. Suitable amplifiers are discussed in the next chapter.

Radiation patterns.

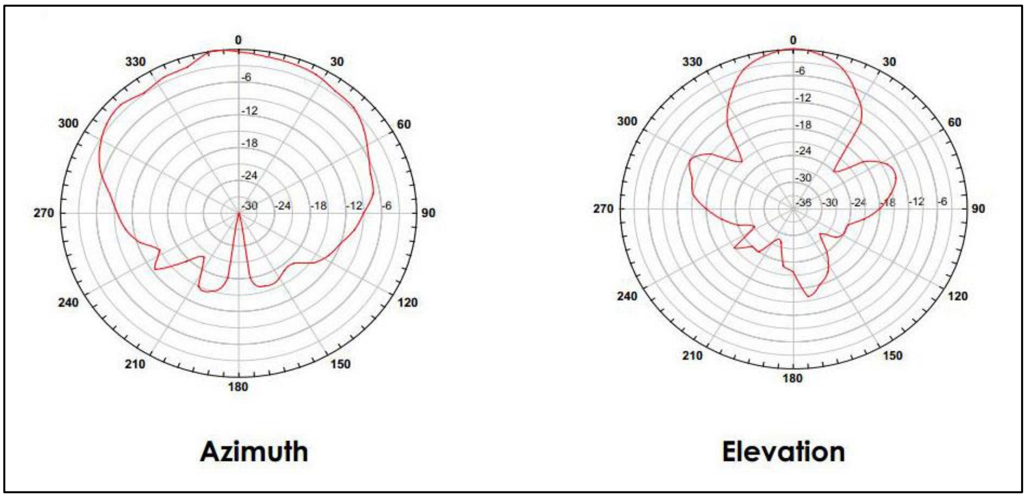


Figure 9, Azimuth and elevation diagrams.

Chapter 4

Receiver and amplifier

4.1 RF receiver / Amplifier

The receiver has to be selective so that the desired signal is extracted from the spectrum. The receiver circuit has several stages of filtering and amplification to reduce noise and interference.
The RF receiver is tuned to a certain band in the electromagnetic spectrum. The operating frequency lies in the band from 3 kHz to 300 GHz. The TRF, or tuned radio frequency receiver, is a multistage amplifier that is tuned to the transmitted frequency.
The first RF receiver was invented and designed by Edwin H Armstrong. He was the inventor of the FM station and was born in 1890 New York City where he also died in 1954 [16]. FM radio could be used to produce a voice channel for example.

4.2 The photocell

A photocell can be defined as a resistor that changes its resistivity depending on the received light on its surface, a light dependent resistor (LDR). It is a sensor that detects light. Referring to the battery, made of cadmium sulphide, it is often called a CDS [20]. Figure 10 shows the parts of photocell.

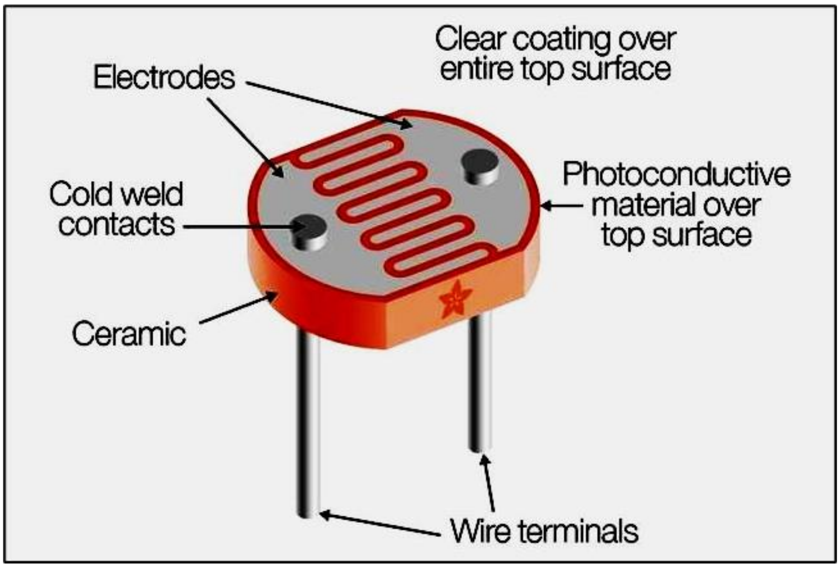


Figure 10, Photocell components.

The cells use low power, are cheap and easy to use, although there are some disadvantages. The accuracy is limited, especially in open spaces, and the cell is difficult to use outside the laboratory.

4.3 The amplifier

In general, the term amplifier refers to a circuit that produces a larger version of the input signal. The classification of amplifiers depends on the mode and configuration of operation. Amplifiers could be operational amplifiers for small signals or power amplifiers for large signals. One of the most suitable amplifiers for this system is the op amp TSB 712.

4.4 TSB 712 OP AMP

This operational amplifier has a high gain, low noise, a wide input range and a temperature range from -40 to 125 degrees centigrade [21].

4.4.1 TSB 712 features

- Rail-to-rail input and output terminals.
- Low offset voltage: 300 µv maximum.
- Wide supply voltage range: 2.7 V to 36 V.
- Dual supply ± 1.35 V to ± 18 V.
- Gain bandwidth product: 6 MHz.
- Slew rate: 3 V/µs.
- Low noise: 12 nV/Hz.
- Integrated EMI filter.
- 2 kV HBM ESD tolerance.
- Extended temperature range: -40 to +125 degrees centigrade.
- Automotive-grade available.
-

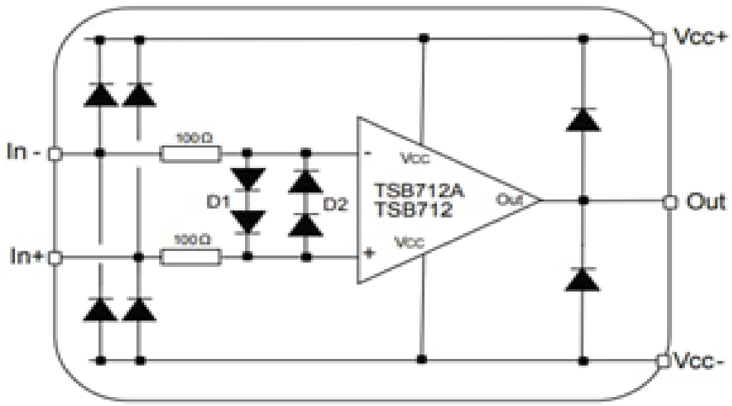


Figure 11, Block diagram for the TSB 712.

4.5 Operational amplifiers

The OP AMP or operational amplifier is a high gain linear integrated circuit with some inputs and output terminals, normally one end output and different inputs. The op amp is fitted with external feedback in the form of capacitors and resistors between its input and output terminals. Can be voltage amplifier.
The input terminal(s) with a – signs are called inverting inputs and those with + signs are called non-inverting input. The v+ and v- are the DC supply that are connected to the positive and negative terminals of the power supply respectively.[21]

There are various OP AMP amplifiers such as:
Differential, isolation, negative feedback, instrumentation and power amplifier. They typically amplify signals from antennas and microphones.

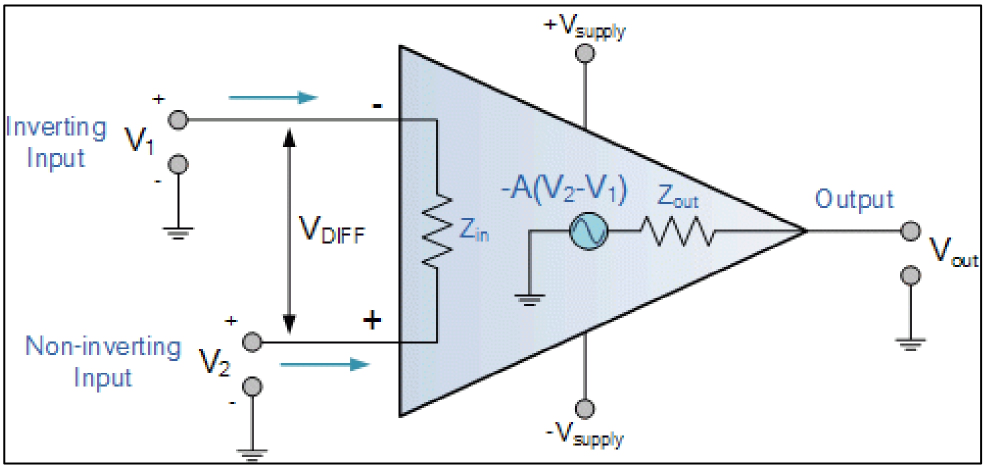


Figure 12: Operational amplifier circuit diagram.

The voltage gain of an operational amplifier is given by

$$voltage\ gain(A_V) = \frac{V_{out}}{V_{in}}$$

And it can be given in decibels (dB) as:

$$20logA_v \quad \text{or} \quad 20\log\frac{V_{out}}{V_{in}}$$

Chapter 5

Sound Waves

The transport of energy through media like gases, liquids or solid matter causes a disturbance. The pattern of that a disturbance is often in the form of sound waves. A vibrating object can disturb particles in the surrounding medium and the particles push the nearest neighbor and this repeats itself. With decaying intensity, the waves move from the source. See figure 13.

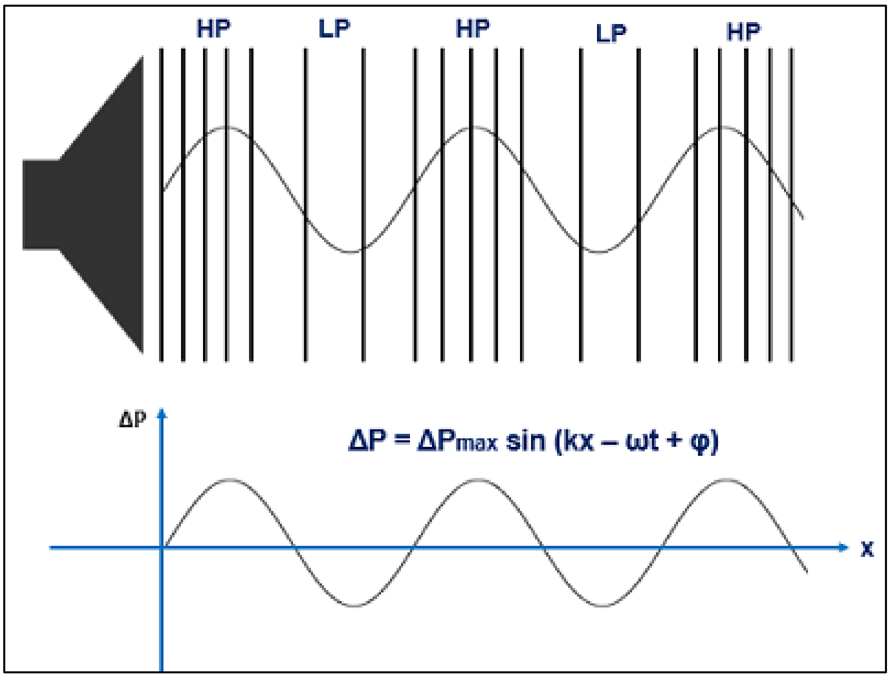


Figure 13, sound waves as longitudinal waves.

5.1 The behavior of sound waves.

In principle we can classify sound waves as:

- Pressure waves
- Longitudinal waves
- Mechanical waves

Sound is a mechanical wave caused by the vibration of particles. It can be affected by the temperature of the medium in which sound waves are propagating. The motion of the particles is longitudinal. i.e., moving in the direction of propagation.

Hence the sound wave has a periodic pattern of high and low-pressure regions. The fluctuations in pressure can be detected by the human ear or some other instrument. The fluctuation is periodic and corresponds to an arrival of high and low pressure.

Sound waves have a frequency that relates to the period T as f=1/T and that is measured in Hertz (Hz).

The wave length relates to the frequency as lambda $\lambda = v_p/\text{f}$ where v_p is the phase velocity [22].

5.2 Sound wave characteristics

A sound wave is often a periodic wave with characteristics (frequency, amplitude, phase and wave shape) that depend on the medium and the source that produces it. For example, the sound wave speed in the air at 20℃ is about 344 m/s and in water is about 1500 m/s. Sound could also be non-periodic with pitches that are hard to detect for the human ear [7][22].

5.3 Ranges and units for sound

The range of frequency for the human ear is typically: 20 Hz < f < 20 kHz. Although the maximum sensitivity for the human ear is at 3 kHz. Sound levels are often measured logarithmically since this corresponds to the sensitivity of the ear.

The reference levels are $W_0 = 10^{-12}$w, and p_o=20 µpa

SPL=$SWL + 10 log Q/(4\pi d^2)$

Q is the directivity factor and d is the distance from the observer to the center of source.

Hence p: is the mean square of the acoustic pressure fluctuations

P_0=$2x10^{-5}$ pa

in the air,

pa = Pasqual.

Now the sound intensity I per surface area with respect to sound wave propagation. The intensity level will be:

$$IL_{dB} = 10 \log (I_{mean} 10^{-12} \frac{w}{m^2})$$

The mean intensity level is the measured sound level intensity.

This value relates to threshold of typical human hearing at 1 kHz at the reference pressure level in air of $p_{oref} = 2x10^{-5} pa$. For progressive plane wave intensity is:

I = $p^2{}_{rms}/\rho 0 c 0 \quad {}^{w}/_{m^2}$

Where p_{rms} is the root mean square of the acoustic pressure fluctuation [7][9].

$p_{ref} = 10^{-6} pa\ in\ other\ media.$

Under the atmospheric condition $\rho 0 c 0 = 4.10^2 kg/m^2 s$.

The human hear can detect sound intensity between -20 dB to 140 dB as shown in the table below:

W/m^2	**dB-SPL**	**example sound**
10^{-2}	140 dB	pain to human ears
10^{-4}	120 dB	
10^{-6}	100 dB	
10^{-8}	80 dB	
10^{-10}	60 dB	normal conversion
10^{-12}	40 dB	
10^{-14}	20 dB	
10^{-16}	0 dB	
10^{-18}	-20 dB	weakest audible

Hence, for f = 1 kHz we have in air:
SPL = 140 dB, p = $2x10^2$ pa , u =$5x10^{-1}$ m/s.
SPL = 0 dB, p = $2x10^{-5}$ pa , u = $5x10^{-8}$ m/s.
Where u is the particle velocity (m/s).

Chapter 6

Product design & implementation

6.1 Prototype design

In order to test the concept, a prototype was designed and built in a simple way. This prototype was tested in the laboratory and can also be tested in the field.

The main part of this design is the doppler radar HB100 and the sound sensor. This combination can give suitable signals to the Arduino MC and drive the whole system.

The HB 100 doppler detects the speed of a moving object and produces a sinusoidal signal as an output. The problem is that the signal is weak so a high gain multi- stage amplifier is needed to feed the Arduino MC.

In addition, the sound sensor will pick up the signal of a possible siren to recognize the vehicle type. The output signal of the doppler radar and the sound sensor feed the Arduino MC. With this combination of signals the Arduino will indicate the type and speed of the incoming vehicle.

The microcontroller reads the Doppler frequency and calculates the speed of the observed vehicle. If the speed of that vehicle exceeds the limit, then the MC will give a signal to the traffic light to change the status as illustrated in figure 14.

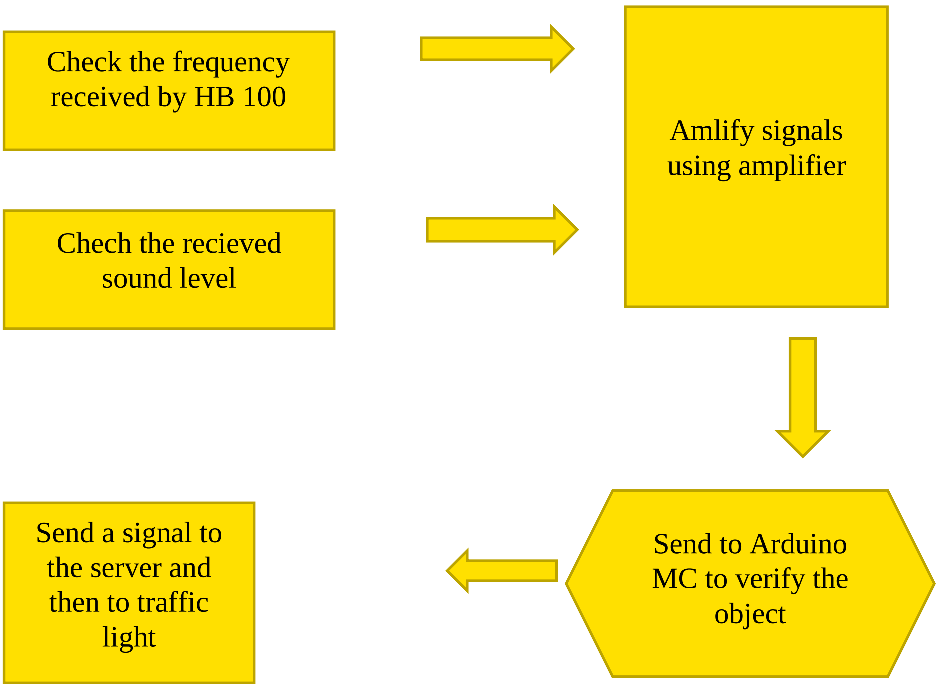


Figure 14, System flow diagram.

6.2 Prototype usage

This concept is suited to traffic lights at the intersection of high-speed routes at the outskirts of cities rather than intersections inside cities. However, the performance of the prototype depends on speed limits and road geometry. It can be useful in intersections that have more than five or six driving fields. Due to the limited sensitivity of this design, it may be useless in crossings that are surrounded by many buildings because of reflection, interference and distortion of waves. The concept can be applied everywhere but with more complicated designs but then with more additional parts and devices.

6.3 Prototype forming

Electronic devices and parts are combined in this prototype built on Arduino MC as shown in figure 15.

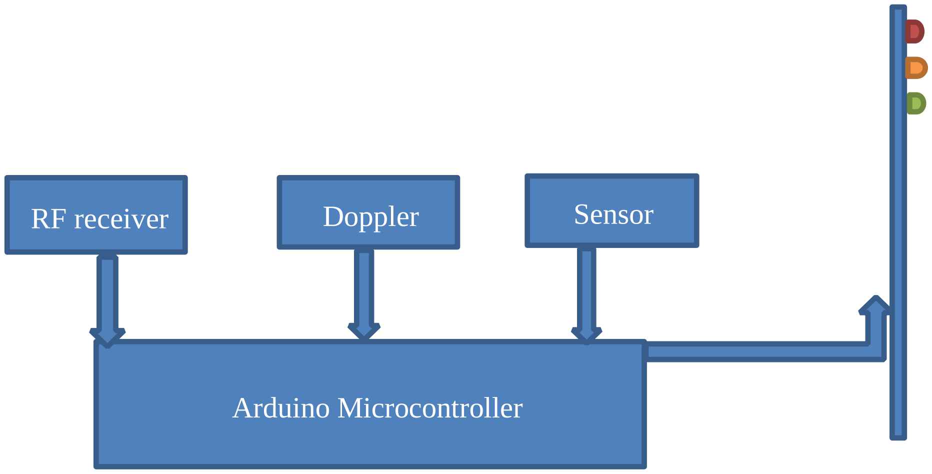


Figure 15, system design.

	State1	State2	State3	State4
Siren signal	0	0	1	1
Doppler signal in	0	1	0	1
Arduino signal out	0	0	0	1
Traffic lights signal	stay	stay	stay	consistency

Figure 16 Truth table and block diagram.

Figure (16) shows the main parts of the device and the way they are connected. These parts are connected with a data buss for sending and receiving the information.

Parts of the device are:

- Arduino Uno MC.
- Doppler radar HB 100.
- Sound sensor.
- Data bus wires.
- Power provided by separate power supply 5 V DC, AC/DC conversion of the traffic light source, or by sun cells.

The doppler radar is a most useful and accurate device to detect the speed of the vehicle. The HB 100 has a clock speed of 10.525 GHz and a built-in mixer and microstrip antenna. There is a phase shift between the radiated and the reflected signal. The output signal of the doppler HB100 is proportional to the speed of the detected object.

The relation between the shift in frequency and the object speed can calculated by the doppler shift equation:

f_d=2cosϴ×f×v/c.

where f_d shift of frequency in Hz.

v is the velocity of the moving object.

c is the electromagnetic wave speed in vacuum.

f is the transmitted wave frequency of the doppler (10.525) for HB100.

A multistage amplifier is also added between doppler and Arduino MC for matching due to the weak output signal of the doppler. The gain of this amplifier depends on the type of amplifier (inverting or non-inverting amplifier).

The sound sensor can sense the siren sound level and feed a suitable level to the Arduino input pin.

The combination of the doppler and the sound information is input to the Arduino MC. The Arduino MC processes these signals and gives a resulting signal to the server that can drive the traffic light.

A light sensor could also be included if necessary.

6.4 Prototype work & connection

After a long and hard work on an electronic circuit designed to control the traffic light, the aim of the thesis was achieved. The first stage is to manufacture a controller prototype and test it according to the standard measurements. A sound sensor is connected to the Arduino microcontroller to obtain and treat the sound signal. By sensing a siren signal that comes from a service car if can be forwarded by the data bus to the Arduino MC. The microcontroller will not act until it receives the second signal from the Doppler sensor.
The Doppler itself will be ready to read the car’s speed. If the object speed exceeds the speed limit the Doppler lemits a signal to the Arduino MC. The microcontroller then will combine the other signal of the sound sensor as an input signal to the Arduino MC, the microcontroller will process these signals and send a signal to the server to change the traffic light status and give priority to the service vehicles. All those processes are implemented with a program installed in Arduino Uno MC. This procedure can take some milliseconds depending on the Arduino microcontroller clock.
In this design a sufficiently good Arduino Uno microcontroller is used. In the future, better microcontrollers could be used. This design has some limitations such as the HB 100 not having a range longer than 11 meters. The angel of detection can be affected by other vehicles, and the interference of sound waves especially inside cities.

6.5 Conclusion

I made a small prototype that is similar to the real device to implement my product with the Arduino microcontroller and some other electronic circuits. This prototype contains a traffic light that is connected to the Arduino microcontroller and sound sensor to get the surround sound. That combination of circuits was tested in a laboratory environment. After a lot of hard work and many tests it works properly and effectively under some conditions. Another model was tested in the laboratory achieved just by connecting a sound sensor to the Arduino microcontroller which worked as a radar system, and has been tested in the laboratory with a good result.

Reflection and interference of sound waves in the laboratory was neglected. Enhancement of this product can be achieved by adding electronic components and software. With more accurate information on the traffic the performance could be improved. My model works with about 80 – 90 percent efficiency. In the future I intend to improve the design and find better and more accurate solutions.

References

1. Smith, Steven W. The scientist and engineer's guide to digital signal processing. California Technical Pub., 1997.
2. *Wagner, L. (2019). https://www.inclusivecitymaker.com/1868-2019-a-brief-history-of-traffic-lights/*
3. Mortenson, Brett. "Characterization of Programmable Arduino Sensors." (2018): 1.
4. Albert, Michael, and Linda F. McCaig. Emergency department visits for motor vehicle traffic injuries: United States, 2010-2011. No. 2015. US Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Health Statistics, 2015.
5. Zhang, Zhenxuan. "Digital Microphone Array Implementation for Hearing Aids Performance Improvement in a Noisy Environment." PhD diss., 2013.
6. McPartland, Randall. Understanding Waves and Wave Motion. Cavendish Square Publishing, LLC, 2014.
7. Rienstra, Sjoerd W., and Avraham Hirschberg. "An introduction to acoustics." Eindhoven University of Technology 18 (2004): 19.
8. Mattson, Margaret E., Rong Cai, and Albert Woodward. "Emergency department visits vs. fatalities among substance-impaired underage youths involved in motor vehicle crashes." Journal of safety research 53 (2015): 45-51.
9. Razlogova, Elena. "The past and future of music listening: between freeform DJs and recommendation algorithms." Radio’s new wave: Global sound in the digital era (2013): 62-76.
10. World Health Organization. Global status report on road safety 2013: supporting a decade of action: summary. No. WHO. NMH. VIP 13.01. World Health Organization, 2013.
11. Gallagher, Kyle A., Gregory J. Mazzaro, Anthony F. Martone, Kelly D. Sherbondy, and Ram M. Narayanan. "Derivation and validation of the nonlinear radar range equation." In Radar Sensor Technology XX, vol. 9829, pp. 188-200. SPIE, 2016.
12. Koshal, Rajindar K. "Deaths from road accidents in the United States: an econometric analysis." Journal of transport economics and policy (1976): 219-226.
13. Colella, R., and L. Catarinucci. "10.525 GHz Backscattering RFID System Based on Doppler Radar Technology for 5G Applications and Telemedicine." In 2019 PhotonIcs & Electromagnetics Research Symposium-Spring (PIERS-Spring), pp. 3689-3695. IEEE, 2019.
14. Nichols, Daniel R., and John O. Wedel. "amplifier." U.S. Patent 3,605,027, issued September 14, 1971.
15. İnanç, Ahmet, and Asaf Behzat Şahin. "Low-Cost Perimeter Intrusion Radar System Development." In 2020 7th International Conference on Electrical and Electronics Engineering (ICEEE), pp. 206-210. IEEE, 2020.
16. ST Microelctronics. (2022). https://www.st.com/en/amplifiers-and-comparators/tsb712.htm.
17. ARDUINO. (10 may 2016). https://www.arduino.cc/.

18. Theory circuit 2013-2022. (20 may 2022) ‘theory circuit-do it yourself electronics projects. https://theorycircuit.com/hb100-microwave-motion-sensor-interfacing-arduino/.
19. Edwin H. Armstrong, December 18 1890- 1 February 1954. (15 may 2016). https://www.britannica.com/biography/Edwin-H-Armstrong.
20. Elsevier B.V, 2022 science Direct. 15 august 2015. https://www.sciencedirect.com/topics/engineering/photocell.
21. Aspin Core 2022, Electronics tutorials. (12 dec 2020). https://www.electronics-tutorials.ws/opamp/opamp_1.html.
22. Eberhard Sengpiel, sengpiel audio. (10 June 2015) http://www.sengpielaudio.com/TableOfSoundPressureLevels.htm.

Appendix 1 Programs

1. Sound sensor

```
sound sensor int
soundSensorPin=A0;      int
soundReading=0;         int
soundThreshold=500;     int
intensity[3]={0,0,0};   int
LEDPins[3]  =  {6,7,5};  int
numberOfPins=3;         int
currentPin=0;           int
fadeCounter=0;          int
fadeDelay=50;       boolean
switcher = true; void setup()
{  pinMode(soundSensorPin, INPUT);  for(int
i=0; i<numberOfPins;i++)

{

 pinMode(LEDPins[i],OUTPUT);

}
} void loop(){
soundReading=analogRead(soundSensorPin);
if(soundReading>soundThreshold) { if(switcher){ aboveThreshold(currentPin);
switcher=true;

      }  } else
    { { if(switcher){ belowThreshold();  switcher=true;
    }
    }
    }   void   aboveThreshold(int   cPin){
    switcher=false;
```

```
if(intensity[cPin]<10)                    {
intensity[cPin]=255;        delay(50);
currentPin=currentPin+1;
} if(currentPin==numberOfPins)



{ currentPin=0;
} } void belowThreshold(){ switcher=false;
fadeCounter++; if(fadeCounter==fadeDelay) {
fadeCounter=0; for(int i=0;  i<numberOfPins;i++)
{ analogWrite(LEDPins[i],intensity[i]);
} for(int i=0; i<numberOfPins;i++){ intensity[i]--; if(intensity[i]<0){
intensity[i]=0;
}
}
}
}
```

2. Light sensor

```
void setup() {           pinMode (5,OUTPUT);
///greenled pin as output           pinMode
(6,OUTPUT);   ///orangeled pin as output           pinMode
(7,OUTPUT);   ///redled pin as output
```

```
   Serial.begin(9600);
  }

  void loop() {

     car1();      // do nothing just go to car1 loop
   }

   void car1(){          int sensorValue
= analogRead(A0); // pinMode (a2,INPUT); unsigned long startTime;
    //variables for 4 bytes memory number     storage
unsigned long elapsedTime;           int led1 =5;  //green  led
int led2 =6;  //orange led                     int led3 =7;  //red    led


   startTime = millis();

  while(analogRead (sensorValue) > 200){    // light brithness 0 to 1024

   digitalWrite(led3, HIGH); //red on   digitalWrite(led2,
LOW);   //orange off digitalWrite(led1, LOW);   //green off

         }
         elapsedTime= millis() - startTime;
          Serial.print(elapsedTime);      // print the value to serial port

          Serial.println(" MS    ");
```

if (elapsedTime > 350){ // time from the blinking light if higer than 350 ms then do ...

digitalWrite(led1, LOW); // green led
digitalWrite(led2, HIGH); //orange led digitalWrite(led3, LOW);
//red led delay
(1000); // standby orange
digitalWrite(led2, LOW); //orange led off

digitalWrite(led1,HIGH); // green led on (wait for the car to drive away) delay(4000);

}

// digitalWrite(led2, LOW); //green }

3.Volume sensing

int out=13; void setup()

{

Serial.begin(9600);//open serialportand setthe budrate 9600bps

}

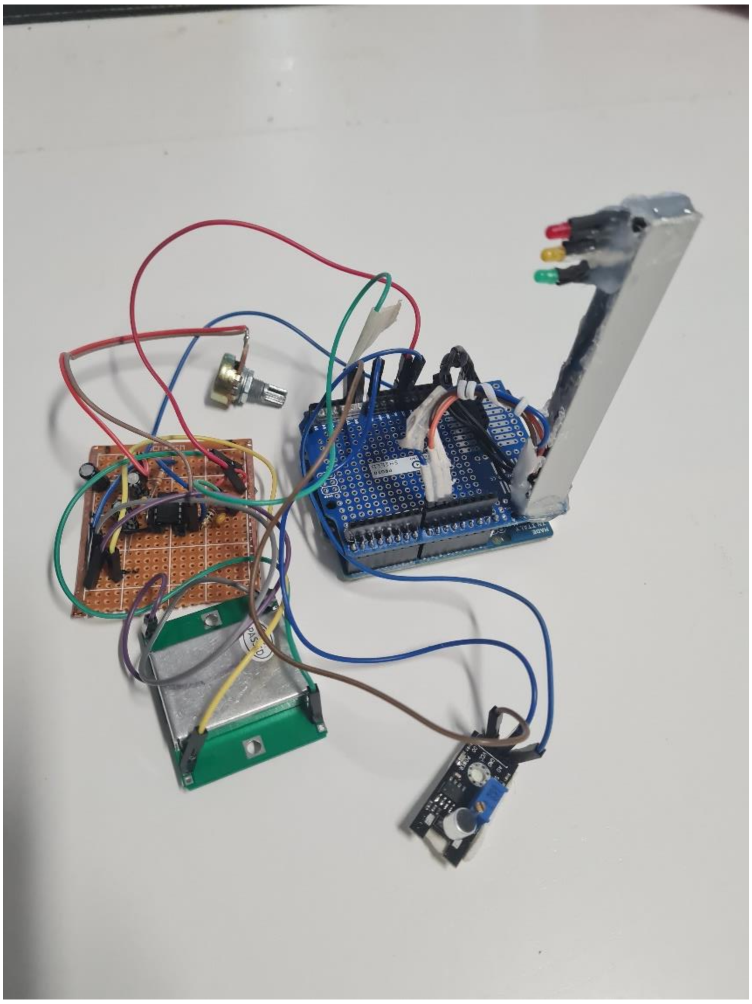
(

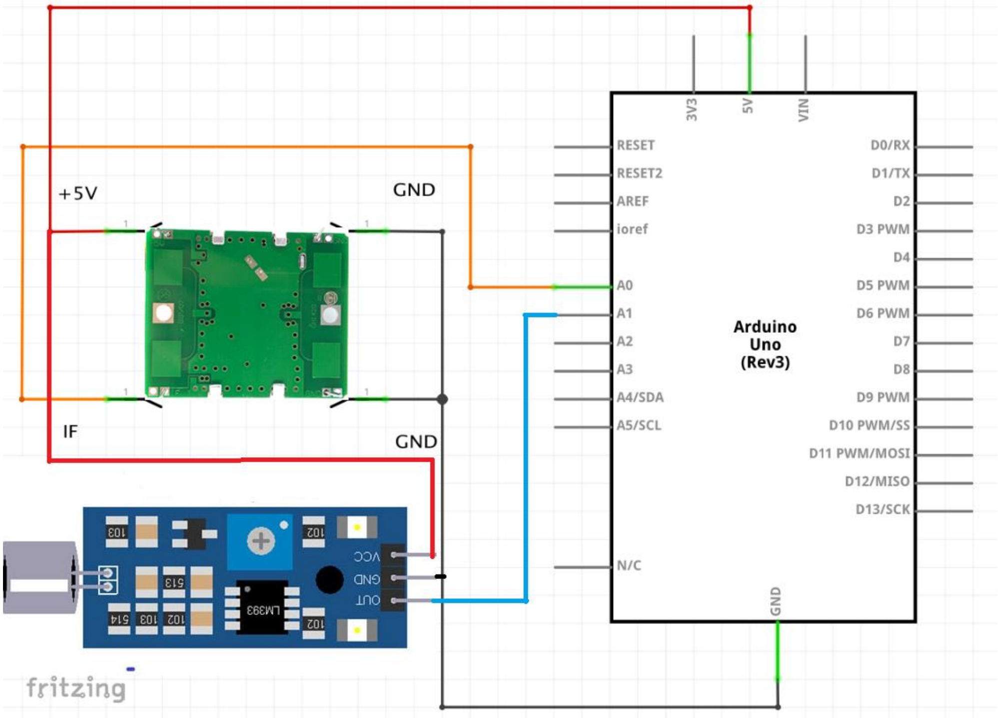
)

{ int val; val=analogRead(A0);//connect the mic to the Sanalog pin A0

Serial.println(val,DEC);//print the sound volume to serial monitor if (val>=300){ digitalWrite (out,HIGH); delay (1000); digitalWrite (out,LOW);

}}




+5V
GND
IF
GND
3V3
5V
VIN
RESET
RESET2
AREF
ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
GND
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
Arduino
Uno
(Rev3)
VCC
GND
OUT
LM393
fritzing