CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

DESIGN AND IMPLEMENTATION OF A PARKING GATE CONTROLLER USING ATMEGA3289 MICRO-CONTROLLER

A thesis submitted in partial fulfillment of the requirements

For the degree of Master of Science

Electrical Engineering

By

Pruthviraja Bhupal

December 2016

The thesis of Pruthviraja Bhupal is approved:

Dr. Somnath Chattopadhyay, ECE	Date
Prof. Benjamin Mallard, ECE	Date
Dr. Nagi El Naga, ECE, Chair	Date

California State University, Northridge

ACKNOWLEDGEMENTS

On the occasion of completing the study and project on Advance parking gate controller, I would like to thank my professor Dr. El Naga who helped me by providing his valuable knowledge about the various hardware components, its architecture and also supporting me to analyze the crucial factors regarding this project.

I would also like to thank Dr. Somnath Chattopadhyay and Prof. Benjamin Mallard for providing their inputs and reviewing this project.

I would like to thank Mr. Sandeep Bidarahalli for providing his support and helping me to use various designing tools.

Lastly, I like to express my gratitude and appreciation to my parents for providing the moral support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

DESIGN AND HARDWARE IMPLEMENTATION OF A PARKING GATE CONTROLLER USING ATMEGA328P MICRO-CONTROLLER

By

Pruthviraja Bhupal

Master of Science in Engineering,

Electrical Engineering

The objective of this project is to design a cost effective Parking gate controller for a sliding gate. Atmel based microcontroller is used to control the whole operation. The parking gate controller system has to auto detect the automobile and open the gate if it is moving out of the parking lot or use a RF remote to open the gate. If the automobile is moving inside the parking lot, then RF remote is used to control the gate operation.

An obstacle sensor is used to detect the presence of the obstacles in the way of the gate. When the gate is closing from an open position and if there is any obstacle in its way then the obstacle sensor detects it, the controller opens the gate.

The system is designed using Arduino microcontroller embedded system and the design is done using its Integrated Development Environment and embedded C.

The design presented in this project is flexible for any of the microcontroller family not just restricting it to the Arduino embedded system. Easy installation and working procedure are explained in this project.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Modern Technology has witnessed an epidemic rise in the recent times, it has changed the complicated human life making it easy and has changed the way of living. It has changed the way of communication, method in manufacturing of products, way of transportation, people's thoughts and knowledge.

Increase in production of goods, health care and communication are some of the many basic fields which the modern technology has changed their path. It has also changed the infrastructure we live in. Modern Technology has led to the construction of smart houses making innovations in Security, Safety, advancement in Home automation, Parking garage, Swimming pools, Jacuzzi's and many more.

Engineers have researched and developed many systems in various field and one such system is the Parking Garage's control system. The Smart parking gate system replaces the parking gate system which use the human force to open with electronic circuits and devices to open the gate thus eliminating the human efforts.

The various types of parking gates are listed below:

1. **Sliding gates**

These gates occupy a large amount of horizontal place but are widely used. The gate slides horizontally leaving place for the vehicle to move. The two methods of design are.

a. **Chain driven:** A rail is placed on the floor at the path of the gate. The gate is mounted on this rail. A chain is connected between the drive motor and the gate.
b. **Gear driven:** A horizontal gear strip is mounted at the bottom of the gate just above the wheels of the sliding gate. The motor's pinion and the gear strip are aligned such that both are in synchronization. The rotatory motion of the motor is converted to linear motion of the slide gate thus moving it horizontally on the rail.

These gates are available in light and heavy-duty sizes depending on the requirement of the user.

2. **Swinging gates**

These gates are used in limited place where the gates open inward or outward position. The force required to operate these are high therefore hydraulic or electro-mechanical operators are used.

3. **Barrier and overhead Lifting Arm Gates**

These type of gates are used mainly in commercial complexes and Industrial areas. The mechanism is to lift the barriers thus giving the way to enter or exit the building. These are operated using belt drive, pulley and direct drive mechanisms.

1.2 Objectives

The objective of this project is to design a cost effective Parking gate controller for a sliding gate. Making minor changes in the gear mechanism of the motor control this design can be used for swinging gates and the barrier gates too.

This design makes use of a Radio Frequency technology to open the parking gate when the automobile is trying to enter into the parking lot from outside. If the automobile is exiting from the parking lot then the Parking gate controller system has to detect the automobile, which may be a Motor bike, Compact car, Crossover SUV, Jeep etc and open the gate.

The detection of automobile is irrespective of the automobile make, tenants of an apartment, people working in a commercial building.

This gate controller is designed to open the gate only for the automobiles and not for the people. The gate controller also detects the obstacles when the gate is closing after an open position and stops its operation and moves to open position upon the detection of obstacles.

Some of the features of the system are:

a. Easy to install.
b. Can be limited to a specific size of an automobile or can be generalized based on the requirement of the user.
c. No technicians are required to install this system.
d. Components and devices if damaged over the course of usage are available readily in the market.

1.3 Project Outline

This Project is presented in five chapters.

Chapter 1 describes an introduction to the project, and it presents the objectives and outlines of the project.

Chapter 2 is about the existing technology of the parking gate controllers, its simplified designs, input and outputs of the system and advantages

Chapter 3 gives a brief summary of all the components, comparison between various sensors and the devices required to design the cost effective Parking gate controller.

Chapter 4 is about the design implementation for a real time parking gate controller, its working operation, pin mapping of the microcontroller with the various sensors, and advantages over the existing parking gate controllers.

Chapter 5 is about the software, flowchart and the code used for this project.

Chapter 6 is the conclusion for this project.

CHAPTER 2

PARKING GATE CONTROLLER SYSTEM

2.1 Parking gate controller system

An Electrical controller system enables the interfacing of various peripheral devices to it, thus enabling the user to control all these and make use of these in practical applications based on his requirement.

A Parking gate controller system also has a controller which is used to open, close and stop the movements of the Parking Gate. The modern gate controller system is very flexible since it can access various interfacing devices connected to it, they may be solar panels, loop detectors and many more.

Some of the leading gate controller companies who are pioneering in this field are Aleko which was established in 2005, Mighty mule, LiftMaster, Honeywell etc.

The most commonly used smart parking gate controller system are in apartments, independent houses, villas, industrial areas etc.

The parking gate controller system is one of the fundamental improvements in the modern day electronics but yet has eliminated the complications in operation of the gate making it effortless and uncomplicated to the people using it.

Before implementation of this technology the whole scenario was different. A person who is driving the automobile has to step down from it, walk towards the gate unlock it, open it by using his physical energy to an extent in which his automobile can go through, then go back to his car, start the engine and enter into the building or parking lot. After he parks his automobile he has to step down, pull the gate and lock it again. Any third person entering into the building has to have the keys to open the gate or he has to contact the person who has the keys to open it.

The old key system might have been a secure aspect for many years but with modern tools it is easy to break in and breaking in will not raise any notifications to the owner.

The smart parking system has eliminated all of this complications and simplified the controlling of the gate operation by just pressing one button. High-end alarm systems are also used in this system where breaking in notifies the owner about the issue and a message about the situation is also sent to the nearby Police department.

2.2 Main components of the System

The Parking gate controller system consists of the following components:

1. **Controller**

This is the brain of the system where the user programs it to perform a necessary set of operations to be done upon receiving the signals from the various inputs connected to it, then send the control signals to the output devices to perform actions based on the inputs. The engineers from the manufacturers preprogram the controller.

2. **Input**

These are various sensors which are used to detect the automobiles, people and obstacles. These are connected to the controller and send the signals to it. There are various types of inputs available based on the range, price and company manufacturer. Some of them are summarized in section 2.3.

3. **Output**

The outputs are generally the indicators to indicate the operation of the gate and the AC/DC motor controlling the movement of the gate. Motor specification should meet with the physical specifications of the gate since it has to handle the weight of the gate to move it.

2.3 Input Accessories

1. **RF transmitter:**

RF Transmitter uses a Radio Frequency Channel linked to a RF Receiver, RF Receiver sends a signal to the controller to operate the movement of the gate. Whoever wants to access a parking gate should have the remote (RF transmitter) provided by the gate controller company in order to control it.

Various companies of the controllers use different kinds of remotes. Few companies provide a number of remote controls with the gate controller. Third party remote controllers can also be used based on the company's encoding techniques.

These transmitters should always be with the automobile since pressing of this from a certain limited distance controls the movement of the gate.

The transmitter can be modified to certain frequency and synchronized it with the receiver during the configuration process in order to privatize the operation of the gate control, since the gate controller companies have same frequency remotes and sell them to other customers.

Depending on the company, gate controller remotes can be encoded in certain techniques so that the same company remote from other user doesn't control your controller. Some of the remotes have various switches and enabling them in certain pattern and performing the same pattern in the receiver will secure the parking gate controller. The pattern changing is nothing but enabling a new frequency channel between the receiver and the transmitter.

2. **Keypad:**

Keypad can be used as primary or a secondary type of input to open the gate. By switching the controller to the programming mode the user can configure the desired combination on the keypad.

The Programming mode is explained in the section 3.5.8.

The controller stores this combination into the memory and whenever a user types in the combination the controller compares this to its original combination set by the user. If the correct combination is pressed, the gate opens.

This can be used a secondary input when using other methods of operation such as a transmitter. If the user forgets to keep the transmitter, he can press the correct code from the keypad and open the gate. If a new person is entering the parking lot or into the apartment building for a temporary span of time he can use the keypad combination from any of the authorized personnel and use it.

3. **Telephone entry.**

A landline telephone or a smart phone can also be used as an input to open the gate. These again have to be installed or synched to the controller, calling from these phones can control the operation of the gate.

A GSM (Global System for Mobile communication) module has to connected and programmed with the controller and the phone numbers from the user's phones have to programmed in to the system, so whenever a call is made to this GSM module from the phone a signal is sent to the GSM module connected to the controller, which compares this signal with the inbuilt numbers and an output signal which is an input to the controller is sent to the controller.

If a call from the stored numbers are made, then the gate opens else an error notification appears and the gate remain closed.

4. **Loop Detectors**

These are the sensors which detect the presence of the automobiles based on the principle of induction and magnetic resonance.

These consist of saw-cut wire which is formed into a loop and is installed in the concrete of the floor. They are powered up thus there is a current flow in it. Magnetic field is generated and these resonate at a frequency which is monitored by the controller. Whenever an automobile moves on it there is a change in this frequency due to the metal chassis of the automobile, and this change in frequency is detected by the detector and outputs a signal to the controller.

This controller takes this as an input signal and opens up the gate.

Card readers

A card reader is a device which reads the magnetic encoded strip placed on the card. These readers are configured with the controller. The cards are encoded and these cards are programmed in to the card reader system, whenever a card is inserted into this reader the reader reads it and compares the pattern with the patterns stored in its memory.

If it finds the right pattern, then a command signal is sent to the controller. The controller reads this signal and takes necessary action i.e. if the correct signal is sent from the card reader then it outputs the signal to the motor to open the gate else it just sends an error message to the display.

CHAPTER 3

COMPONENT DESCRIPTION

3.1 Arduino

Arduino is an advanced Micro-controller embedded system and has an open-source software which can be used by anyone and can design various projects using its IDE (Integrated Development Environment). It has its own IDE which is used to compile and run the program on the PC and download it to the Micro-controller embedded board. They have a circuit board which can be programmed in C language.

The Arduino can be programmed using the updated version of C++ and its IDE has a provision to color code the written program which highlights the parts of syntax. The Arduino does not require any new intermediate hardware to download the program from the PC, the Universal Serial Bus (USB) is used to dump the code from PC to the board. It also has a wide range of library functions which are available online from their website making it easy to program various peripheral devices and also to learn more about coding. This property of Arduino is what made it more popular and prevalent in a short span of time.

Arduino embedded system was designed with an initiative to teach coding and electronics to students who have a limited knowledge of this branch. It was invented in Ivrea Interaction Design Institute. Arduino software is compatible for Windows, Linux, and Mac operating software.

Arduino boards are widely known and the recent surveys estimated that in 2005 approximately three hundred thousand boards were produced and in 2013 around seven hundred thousand were available in programmer hands. The invention of Arduino has influenced many people to enhance their creativity, designing and coding skills. Broad range of small and large scale projects have been designed by amateurs and professionals.

3.1.1 Arduino UNO

Arduino UNO is one of the most widely admired embedded system in the whole Arduino class and mainly used by neophytes. The Arduino is basically an embedded system which consists of a physical board which has a circuit of connections to various components. It has a cluster of GPIO pins (General Purpose Input Output pins) for connecting a wide variety of peripheral devices, it has an interfacing system to interact with the interfacing components and devices called Serial communication system. It is one of the cheapest microcontroller available and has become an essential product for the beginners.

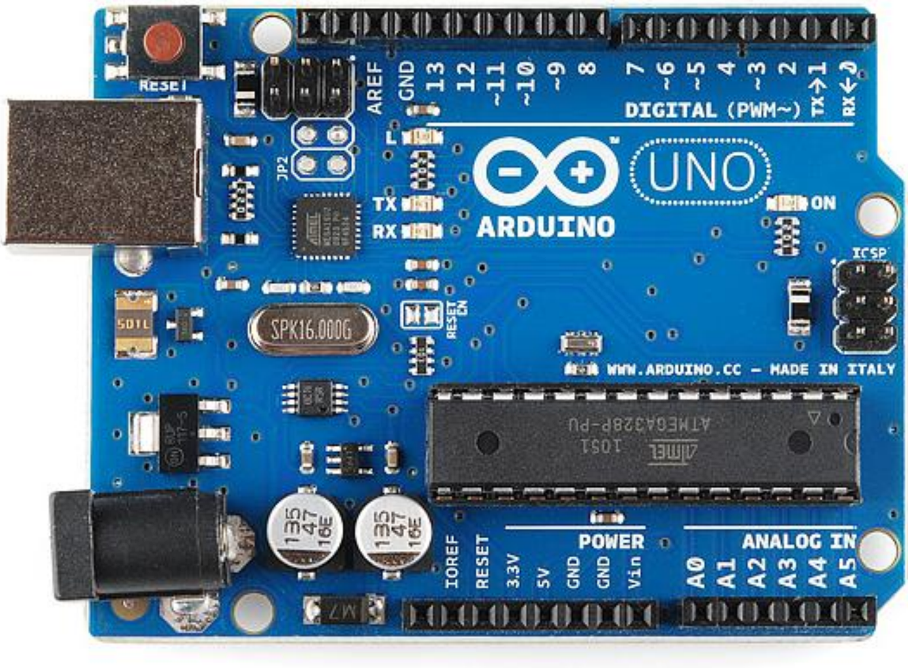


Figure 1 Arduino UNO

Arduino UNO comes in an assembled form and its software can be downloaded online from their website and its schematics for free of cost, data sheets are also available. The Arduino Forum has all the startup guides required for the initial setup and also simple projects to boost up the user's interests.

3.1.2 Hardware

The Arduino is a Micro-Controller based embedded system therefore houses a micro-controller from the ATMEL family. The micro-controller can be 8 bit, 16, bit or 32 bit architecture and is embedded into the board with various supporting components for its stable operation. One of the main attribute of the Arduino is its connecting capability with the supplement modules which can be directly connected to the on board processing unit and called SHEILDS. These can also be connected with the insertion of pins straight to the board.

It has the unique capability to connect various shield on top of each other and access all shields simultaneously. These Shields interact with the Arduino using its Serial Communication System. Some of the Arduino Shields are as follows:

1. **TFT Touch Shield:** A touch screen display interfaced to Arduino.
2. **Ethernet Shield:** Provides Internet connectivity to Arduino.
3. **GPS Shield:** Provides the location and time information which can be made use for various systems.
4. **Motor Shield:** Connecting of AC/DC motor directly to these which are in turn connected to the Arduino.
5. **Flash Shield:** Provides 16 Megabytes of storage.
6. **4-digits Shield:** 4 seven segment displays as an alpha-numerical display.

The Arduino boards require a power supply of 5 volts and crystal oscillator of 16MHz to operate. It has an on board 32 kilobytes flash memory, 1 kilobyte of Erasable programmable read only memory (EPROM). Arduino does not need a chip programmer unlike other microcontrollers which have to be plugged into a programmer, since the Arduino's are equipped with the boot loader thus making it easy to initialize, startup and run the programs. The regular PC itself acts as one of the programmer and the updated version of the boot loader is called OPTI boot loader. The Arduino microcontroller family has different types of Arduino embedded systems which can be used for various applications hence they are equipped with different Atmel family's microcontrollers such as ATmega1280, ATmega328, ATmega168, Atmega2560 and ATmega8.

Level Shifters are used for the conversion between TTL (Transistor-Transistor Logic) and DTE (Data Terminal Equipment) such as RS232 for Serial Data Communication. The updated Arduino UNO boards have an AVR chip used for the communication between the Universal Serial Bus to the Arduino's firmware. This in turn has a technology to program itself using in circuit Serial Programming reader.

The Arduino board has a total of 14 pins which can be accessed directly from the board, out of them 6 of them are PWM (Pulse width modulated), one pin for the Ground (GND). One for the Power supply in (Vcc) and 6 are for analog inputs which in turn can also be used for the digital inputs too.

Arduino is based on the Harvard architecture and has an onboard memory. The code written is stored in a separate memory called Flash program memory and the data is saved in the Data memory. Arduino has 32X8 General purpose registers, Instruction register, Instruction decoder, Analog Comparator, SPI (Serial Peripheral Interface) unit, Program counter, status control, 8-bit Data bus.

3.1.3 On board Circuitry

The Pin map for the Arduino is as follows:

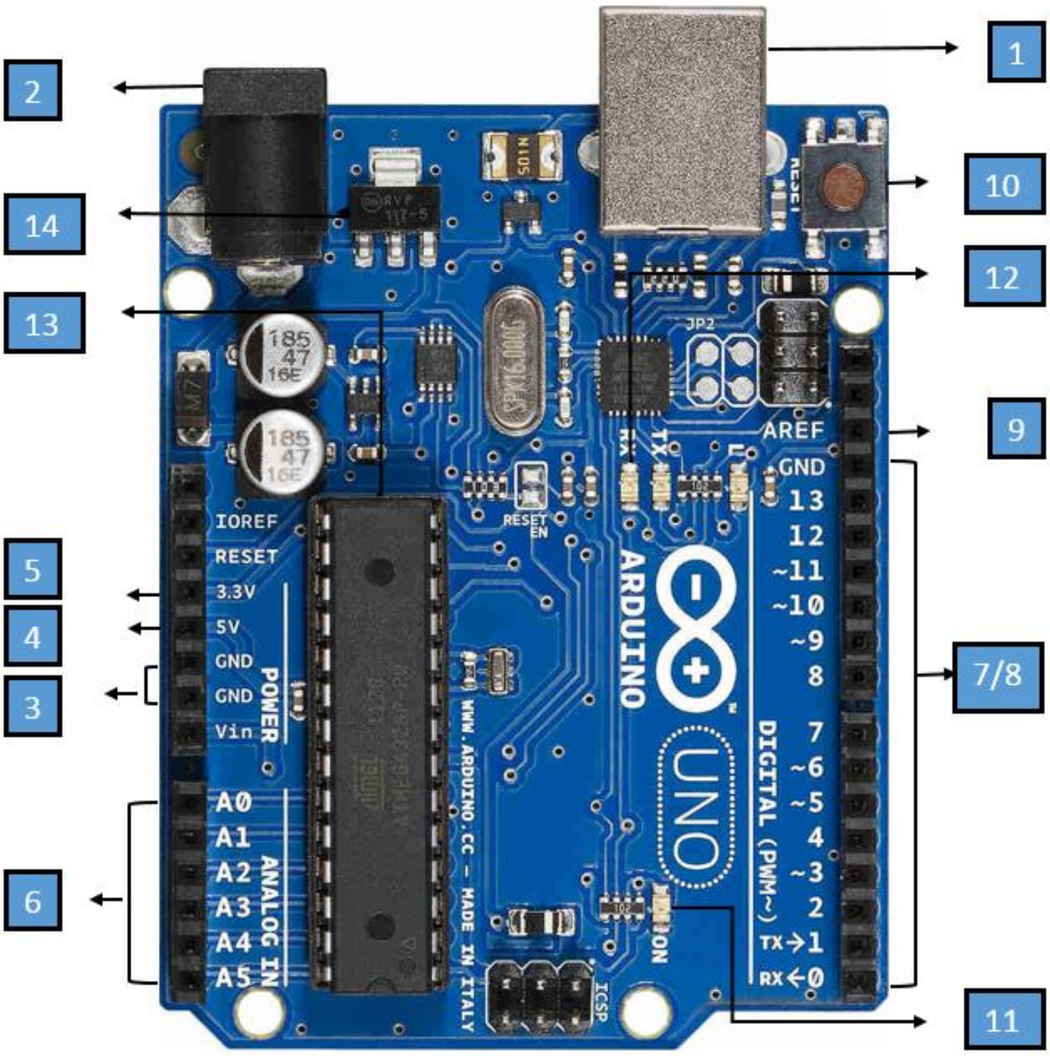


Figure 2 Pin map Arduino

1. **Power Port**

   This is a Universal Serial Bus (USB) which is used to connect to the PC and power up since every embedded system has to be powered up to operate. This port is also used to download the source code from the computer.

2. **Power Barrel Jack Port**

   This port is also used to power up the Arduino board, a 5volt DC power supply which steps down the voltage from 220/110V AC mains to 5V is connected to this port.

   Any power supply above 5VDC should not be used to power up the system.

Pins are the ports which are used to physically connect the GPIO slots to the wires from the developing board. Arduino board usually equipped with a black strip consisting of holes to connect the jumper wires.

3. **GND (Ground)**

   There are three GND pins available on the board to connect the circuit to the ground thus making it complete. These pins play an important role since an extra connection to the ground is not required.

4. **5-volt Pin**

   This supplies the 5volt power to the various components which are interfaced to the Arduino microcontroller thus making it easy as a new power supply is not required to power these peripherals.

5. **3.3-volt Pin**

   This supplies a power of 3.3 volt to the peripheral devices connected to the Arduino microcontroller since some low end components require very less power to operate.

6. **Analog Pins**

   These as shown in the figure are labelled from A0-A4 on the Arduino UNO board which may differ from other Arduino boards, therefore a total of 5 pins which read the analog signals from the components connected to it and output the digital signal.

7. **Digital Pins**

   These as labelled in the above diagram are from 0-13 pins on the right side of the board. These pins take a digital input and deliver a digital output. The 13th pin is equipped with an LED.

8. **PWM**

   The pins such as 3, 5, 6, 9, 10 and 11 are not just regular digital pins but also be used for PWM (Pulse Width Modulation). With these analog output can be obtained using digital platform.

9. AREF

It is used to configure the analog inputs to a high mode and known as Analog Reference. Most of the time it is not used.

10. Reset Button

This is used to reset the whole operation of the Arduino microcontroller. Pressing this button will enable the ground and the pointer resets to the first line of the code. This helps in testing of the projects.

11. Power LED indicator

This is an indicator LED to indicate that the Arduino is in ON or OFF mode. Whenever the Arduino is connected to the power supply through the barrel jack or the USB port this LED turns on. This helps in testing of the board.

12. TX/RX LED's

These pins are for the Serial Communication and they are Transmit, Receive LED's. There are two places on the board where these are visible, one at the 12th marker as shown in the diagram northeast to the microcontroller and one near the digital pins 0 and 1. They turn on when the Arduino is transmitting or receiving the data.

13. Main Integrated Circuit

The 28 pin IC is the brain of the Arduino board. This IC comes in two types i.e. mount type and the other one is the through-hole package and the other is the Surface mount. The IC type must be noted when programming the Arduino.

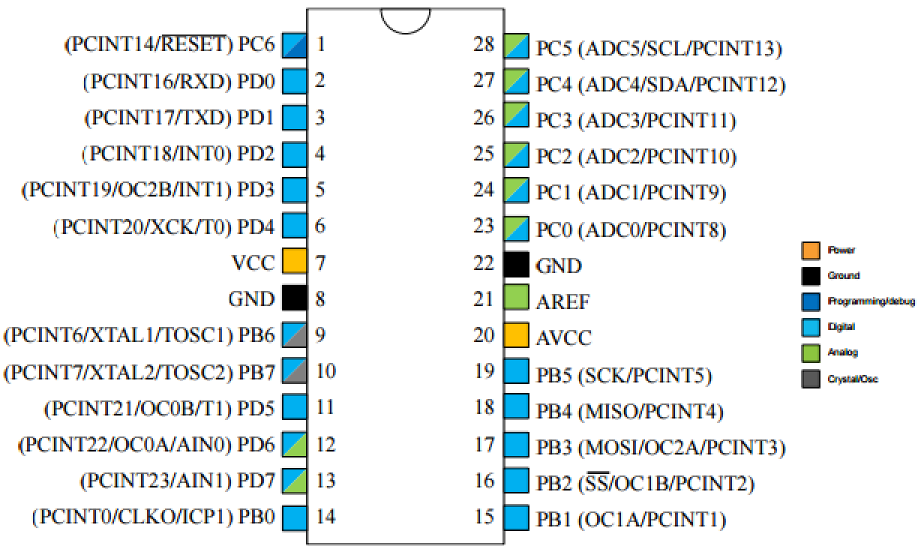


Figure 3

14. Voltage Regulator

This component monitors the voltage that is provided from the power supply and acts like a fuse.

3.1.4 Features of Arduino

1. Arduino is a gigantic family offering wide variety of embedded systems with various number of GPIO pins, boards based on their applications, wide range of shields, sensors and several interfacing devices.
2. Low power consumption and sleep modes are enhanced in the selected AVR chip designs and hence this technology is known as Pico-power technology.
3. This is a very high integration board which has onboard SRAM, Flash memory, Data memory, EPROM, watch dog timer, GPIO pins, and crystal oscillator.
4. Because of the Internet Of Things, the applications of these are limitless and can be used for home automation, medical applications etc. The Internet of Things is possible since this has a processing power to compute various operations.
5. It has a high clock speed of 16MHz, thus making it faster which is one of the most important attribute of the Arduino.

3.1.5 PIN Description

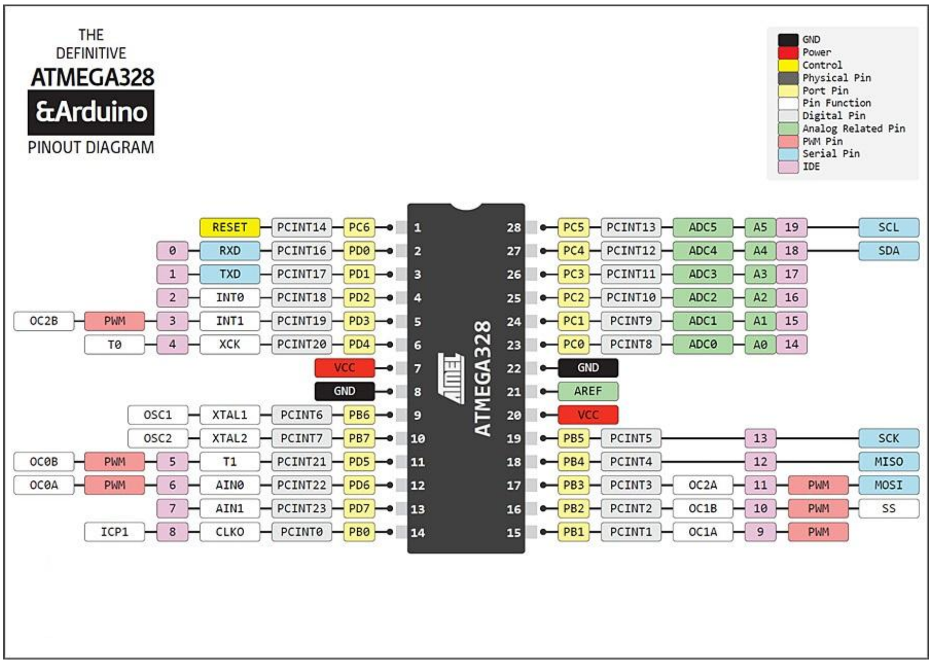


Figure 4 ATmega328 PIN

1. **PIN 7**

   Power supply voltage to the IC referred as Vcc.

2. **PIN 8 and PIN 22**

   Connects to the Ground of the IC referred as GND.

3. **PORT B**

   This Port is an Input-output dual directed port with internal pull up registers. Its sink capacity is high. The Reset becomes enabled when this port goes to tristate mode when the clock is turned off. Port B consists of 8 pins from 14-19 10 and 9 which are referred as PB0, PB1, PB2, PB3, PB4, PB5, PB6, AND PB7 respectively.

   PB7 as in the diagram above has a connection from the Oscillator amplifier hence it acts as an output to it.

   PB6 based on the fuse operation and the selection of the clock can be utilized as an input to the Oscillator amplifier which is in inverting mode.

4. **PORT C**

   It is similar to the PORT B having internal pull up registers and dual detected input-output pins. The Reset condition is enabled if the IC goes in to the tristate mode and does not depend on the clock. Port C extends from pins 23-28 and 1 therefore a total of 7 pins which are referred as PC0, PC1, PC3, PC4, PC5 and PC6 respectively.

   PC6 is connected to the Reset button, thus making it different from the other pins since this is connected the ground. Pressing this button interrupts the ongoing function and starts the whole operation.

   This does not depend on the clock and a pulse length of limited length for longer span of time will generate a Reset operation.

5. **PORT D**

   As the two other ports PORT D is also a bi-directional input-output port which houses internal pull-up resistors. They Reset condition is enabled if the IC goes in to the tristate mode and does not depend on the clock. Port D extends from Pins 2-6 and Pins 11-13 a total of 8 pins referred as PD0, PD1, PD2, PD3, PD4, PD5, PD6 and PD7 respectively.

   AREF is Analog reference pin connecting the Analog-Digital converter.

A Vcc is the power supply voltage pin for the Analog-Digital converter. A low pass filter should be used to connect to the Vcc if Analog-Digital converter is used.

IF Analog-Digital converter is not used then this pin should be connected to an externally to Vcc.

These also serve as 10-bit Analog-Digital converter channels.

3.1.6 Programming

Arduino developers have developed a simple Arduino IDE (Integrated Development Environment) making it easy to the users to download it online for free and use it without using any third party software. It also consists of inbuilt libraries and the sample programs helping the beginners to get a hold on using this software and also to code the microcontroller.

The Atmel IC for the Arduino UNO is ATmega328 which is preprogrammed with new boot loader thus making it uncomplicated to use since this eliminates the usage of a communicators between the Arduino software from the PC to the Arduino system thus eliminating all the external programmers.

In circuit Serial Programming (ISCP) headers can be used to elude the bootloader and directly code the Arduino board.

Atmel has developed the software such as Flip for Windows OS and DFU for the Mac OS to load any newly developed firmware.

Syntax analogWrite() is used to get the 8-bit Pulse Width Modulated output from the pins 3, 5, 6, 9, 10 and 11.

The pin AREF is enabled with the syntax analogReference().

Interrupts play a very vital role in any application, they trigger a signal whenever there is a need of high priority events. Therefore, Arduino UNO has 2 pins which are used as multi-functional pins. Pins 2 and 3 are dedicated for this interrupt function.

Syntax attachInterrupt() is used in the programming to enable the interrupt function. The interrupt function does not depend on the signal edge.

Because of the wide range of the library functions Two Wire Interface (TWI) is possible with the installation of Wire library.

Arduino firmware uses the syntax like pinMode(), digitalWrite(), digitalRead() to program the 14 pins of the Arduino and each of them can used for Input-Output operations. The operating conditions of these pins is that all of them need a power supply of 5volts and current rating of 20milliamps.

Pin 13 has an inbuilt LED which turned on when the pin value is set HIGH and turns off when the pin value is set LOW. Example code to enable the LED

Figure 5 Arduino IDE and Sample Code

The Figure 5 shows the Arduino IDE to code the Arduino embedded system.

The above sample code is used to turn on and off the LED.

1. pinMode(LED, OUTPUT);

   The above code sets the pin 13 referred with the variable LED to output mode
2. digitalWrite(LED, HIGH);

   This code writes the HIGH as to set the LED i.e. pin 13 to turn ON.

3. delay(1000);

   This indicates a delay function, since the LED is turned on there has to be a time specification. 1000 indicates the time in milliseconds therefore turning it on for 1 second.
4. digitalWrite(LED, LOW);
5. This code writes the LOW as to set the LED i.e. pin 13 to turn OFF.

The whole program when executed turns ON the LED connected to the PIN 13 for one second and turns it OFF for one second and the whole operation is done only one time since there is no code written for the multiple executions.

3.1.7 Arduino UNO Specifications

Microcontroller	ATmega328
Length	68.6mm
Width	53.4mm
Weight	25grams
Input voltage	7-12volts
Operating Voltage	5volts
Current rating	20milliamperes
Analog Pins	6
Current rating for 3.3volt Pin	50milliamperes
Pulse Width Pins	6
SRAM	2kilobytes
EEPROM	1kilobytes
Flash Memory	32kilobytes
Built in LED	13
Clock Speed	16Mhz

3.2 Ultrasonic Sensor

Rapid growth of the Information technology, increasing demand for less complicated advanced automation and smart electronics have led to various innovations of small scale eco-friendly devices and components.

Ultrasonic sensors are one of the best examples that fit the above argument. It is a device which emits waves i.e. ultrasonic waves into the open space wherever it is mounted and then receives the reflected waves.

3.2.1 Structure and Working Principles

a. Enclosed Ultrasonic sensor

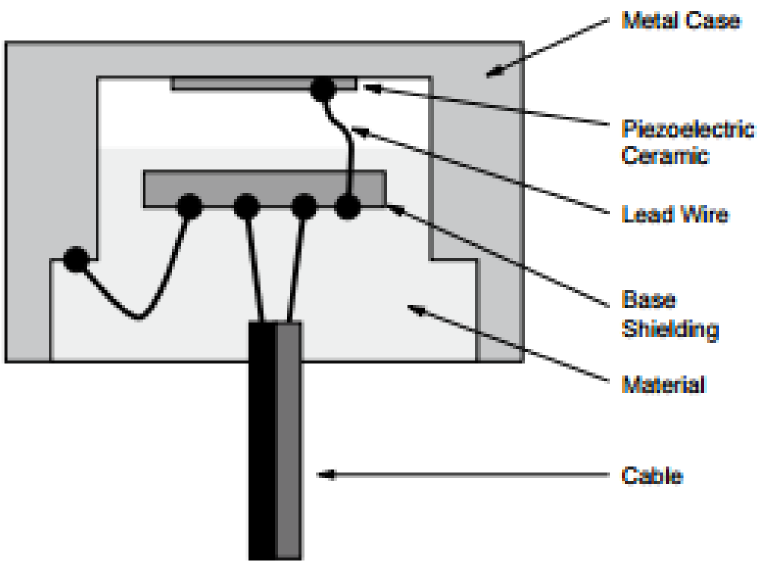


Figure 6 Inner view

Sensors has to be enclosed in order to avoid the exposure of the electrical circuits to rain, heat and any atmospheric constraints depending on the place where it is mounted.

The whole device has a metal case on the outside and a coating of piezoelectric ceramics to its immediate inside layer. A coating of resin is present at the extreme entrance of the sensor.

Figure 6 shows the inner view of the enclosed type sensor and Figure 7 shows the overall appearance of the sensor.

Figure 7 Enclosed Ultrasonic sensor

b. **High Frequency Ultrasonic Sensors**

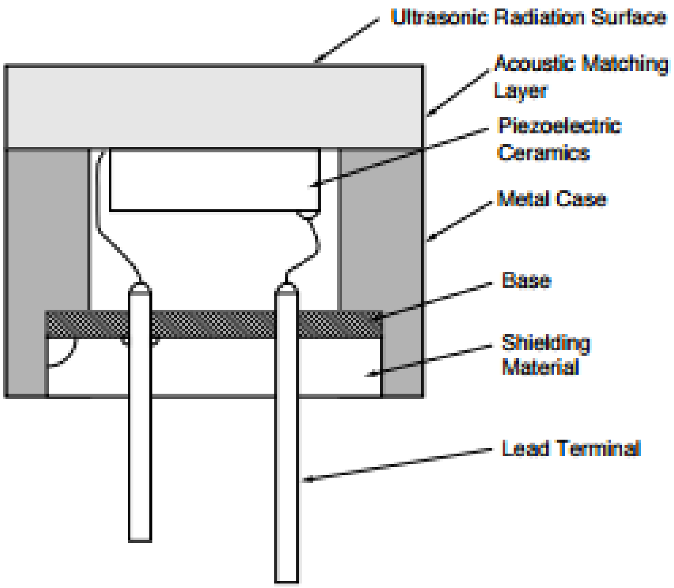


Figure 8 High Frequency Ultrasonic Sensors

These ultrasonic sensors are used in high end applications which require high speed computations. These sensors work at a speed ranging in hundreds of kilo Hertz. The properties of piezoelectric ceramics are what drives these to obtain this speeds. The impedances must be matched between the open space air and the piezoelectric ceramics. A special material is used to match the impedances.

3.2.2 Characteristic Features of Ultrasonic Waves

Ultrasonic waves are the waves produced from the sounds whose frequency ranges above 20kiloHz. The velocity of sound in air is around 344m/s.

1. **Wavelength and Radiation of the sensor**
   Since the speed of these waves range in 344 miles/second the wavelength is shorter and because of this high resolution can be obtained which in turn is the precise computations of the distances.
2. **Reflection**
   When the waves are triggered they hit the obstacles and these obstacles can be any opaque objects such as wood, vehicles, people depending on the place and the application thus the wave reflects from these objects and the ultrasonic sensors are capable of detecting these reflected waves.
3. **Attenuation**
   The purpose of the ultrasonic sensor is to trigger a signal i.e. a wave, these waves get weaker as they travel to the longer distances. The frequency and the distance it travels is inversely proportional i.e. higher the frequency of this sensor shorter is the distance it travels.
4. **Temperature effects**
   The temperature must also be taken to account when measuring the distance using these sensors since wave propagation expression is as follows

   $C=331.5+607T/1000$ T -Temperature in C

   C -Velocity of Sound miles/second

   Temperature must be measured when calculating the distance.

In general, an ultrasonic sensor consists of two sections

1. Transmitter which transmits the ultrasonic waves into open space which are in the range beyond human detection and are in the frequency range of 20 KHz.
2. A receiver to receive the reflected waves which were previously transmitted by the transmitter and are reflected because of the obstacles in the way of the wave propagation.
3. The time to taken to sense the reflected wave depends on the distance of the object from the sensor. If the object is at larger distances, then the time taken is more and vice versa.

4. The receiver cannot detect the wave if the objects are at very long distances since the reflected wave becomes weak and lost.
5. The whole system is built on the principle of Echolocation.

3.2.3 Echolocation

This is a process used by the bats to detect the place of the objects or the prey. This is used by various animals who just act as ultrasonic sensors emitting the calls and then receive the echoes of their calls which are rebounded from various objects. These echoes help them find and spot the objects near them. Depending on their environments they use this process for scavenging and navigation. Few species of whales and dolphins use this technique and mainly bats belonging to the Microchiroptera species.

Active Sonar is also an example for the echolocation. The computation of distance is done by measuring the sound produced from the animal and time taken between its transmission and the echo returned. The angle at which the rebounded waves arrive depends on the information processed by the bat’s ears when they hear the echo. The intensity of the calls differs upon the reflection and the echoes heard by the bat’s two ears are at two different time intervals so, the distance measurement is because of the time and sound intensity parameters.

Echolocation is because of the neural network and the auditory processing capabilities of the brain at the most fundamental level. Bats have the potential to produce sounds which are beyond the hearing abilities of human ears, therefore they call these sounds and hear the rebounded echoes. They deduce the location of these objects based on the estimation of the time. They have the ability to produce the visual and aural information just like humans when they hear the echoes to identify the objects.

3.2.4 Electrical Properties of the Ultrasonic sensors

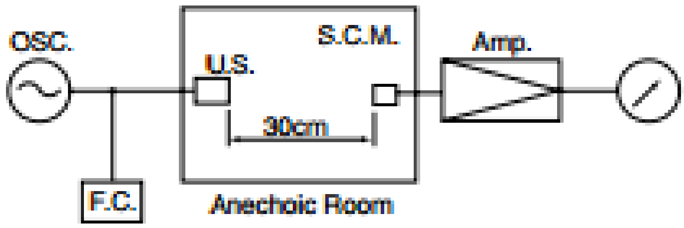
1. **Pressure of Sound Properties**

SPL (Sound Pressure Level) is defined as a unit which indicates the volume of the sound, it is expressed using an expression as follows:

Sound Pressure Level=20 log SP” (dB) SP’

SP” -Sound Pressure

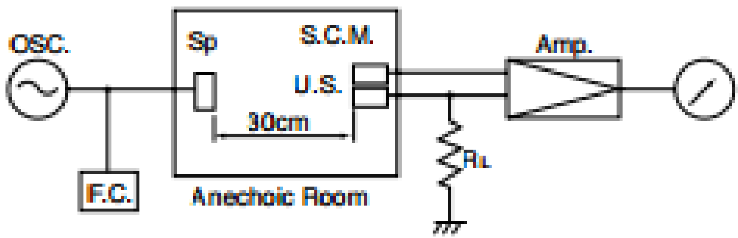
SP’ -Sound Pressure at 20micropascals



U.S.	: Ultrasonic Sensor	
S.C.M.	: Standard Capacitor Microphone	(Brüel & Kjær 4135)
Amp.	: Amplifier	(Brüel & Kjær 2610)
Input Voltage	: 10Vrms, Sine wave	
F.C.	: Frequency Counter	0dB=20μPa

Figure 9 Sound Pressure

2. Sensitivity of Ultrasonic Sensor



RL	: 3.9kΩ	
U.S.	: Ultrasonic Sensor	
S.C.M.	: Standard Capacitor Microphone	(Brüel & Kjær 4135)
Amp.	: Amplifier	(Brüel & Kjær 2610)
OSC.	: Oscillator	
Sp.	: Tweeter	
F.C.	: Frequency Counter	0dB=10V/Pa

Figure 10 Sensitivity

Sensitivity is the ratio of change in output to the change in input and the sensitivity of the Ultrasonic sensor is expressed with the expression as follows

S=20 log So (dB)

Where

S -Sensitivity

So -Sound pressure used for reference (volt/pascal)

In the above figure a 3.9Kohm resistor is used to reduce the impact of noise.

3.2.5 Applications of Ultrasonic Sensor

1. Due to the easy circuit structure these are used in the switches such as approximation and also in counting machines based on the principle of continuous signal and detection of level.
2. The applications depend on the place of usage, the system is configured in such a way so that the automatic door close when a specified number of pulses are detected and these are beneficial in the places of changing environment.
3. The alarm systems making use of the Doppler Effect which generates a distorted signal depending on the object moving closer or farther.
4. Because of the measurement of straight propagation of time they might be used in Densitometers and Flowmeters. The principle is to use the concept of velocity change because of the speed of flow and density properties of gas.
5. Again in flowmeters to measure the number of Karman Vortex which are produced against the speed of flow and the sensor signals are attenuated to a limit when they enter into these vortexes.

3.2.6 Ultrasonic Sensor Module HC-SR04

This is one of the cheapest available Ultrasonic sensor available in the market. It is 2cmX400cm and can detect up to 4meters.

Figure 11 Ultrasonic Sensor HC SR-04

The above figure shows the ultrasonic sensor and its pins are labelled as follows:

PIN 1 -Vcc Power supply

PIN 2 -Trigger a signal

PIN 3 -Echo, Receiver

PIN 4 -Ground

3.2.7 Dimensions of Ultrasonic Sensor HC SR-04

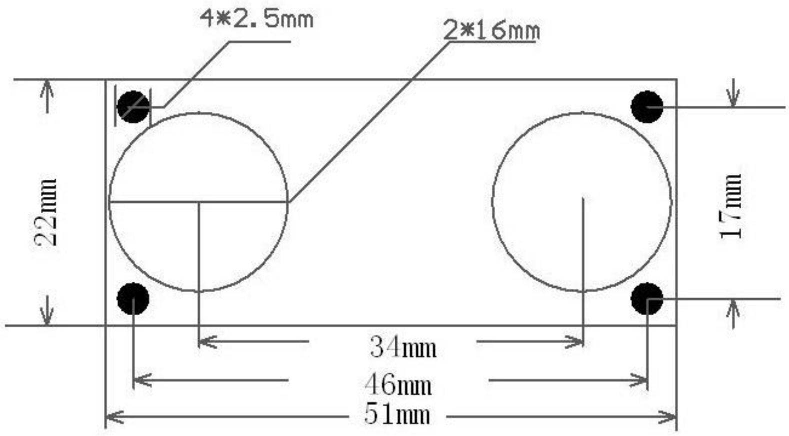


Figure 12 Top view

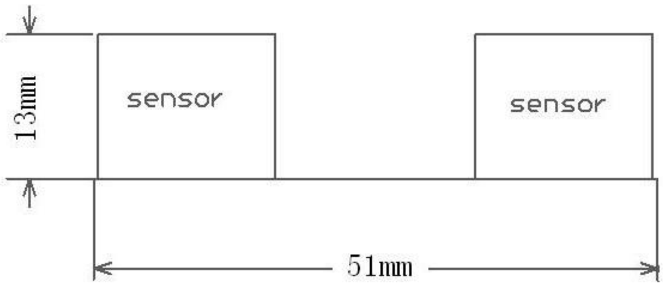


Figure 13 Side View

3.2.8 Schematic diagram when connected with Arduino

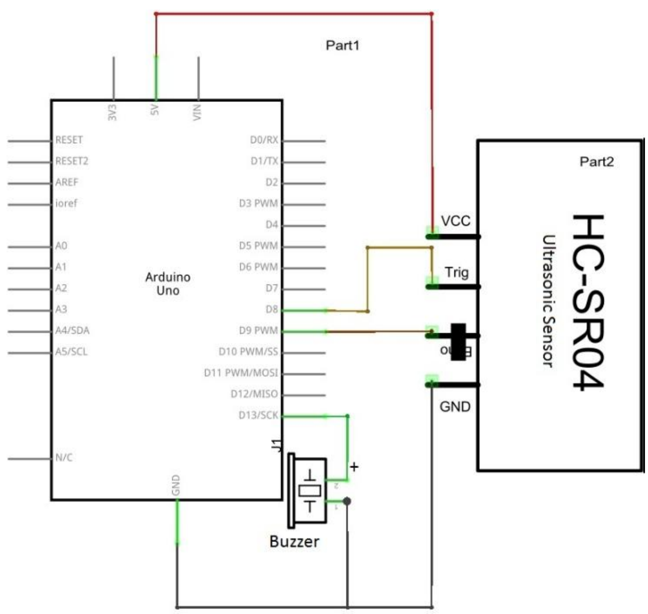


Figure 14 Schematic US with Arduino

3.2.9 Distance Measurement

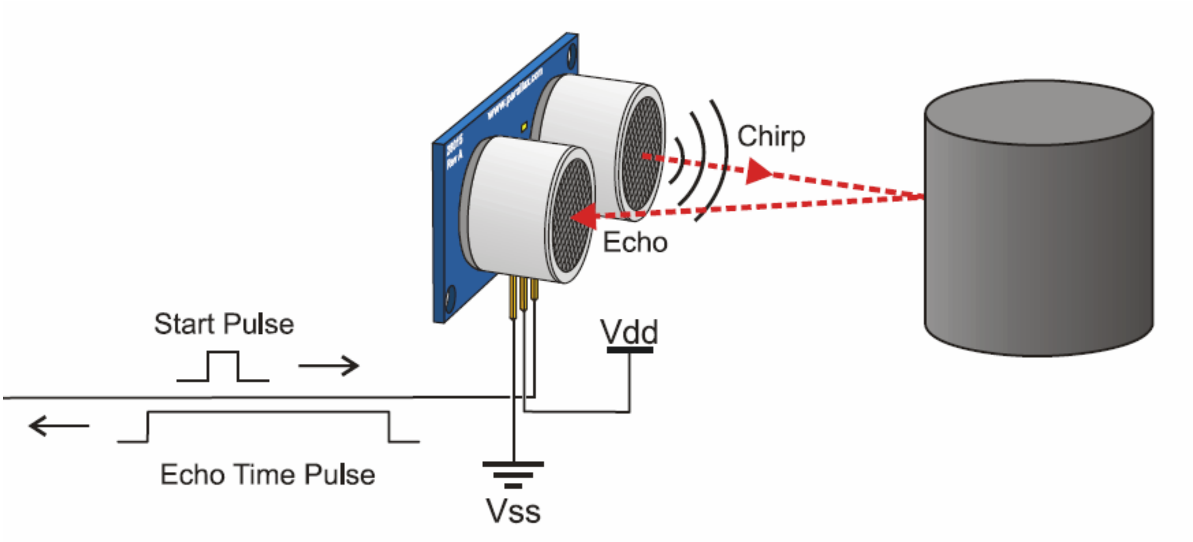


Figure 15 Distance Measurement

The fundamental principle is as to trigger a signal and start timing it and then receive the deflected signal after it encounters with some obstacle. By the principle of time different messurement i.e. S=340 x t / 2 where

S -distance in meters

T -time in seconds

Since the speed of sound is known as 340 meters/second and the time is also noted for the tiggring and receiving of the signal, the above expression can be used to calculate the distance between the sensor and the obstacle. The product must be divided by 2 since the signal has travelled two times i.e. from the ultrasonic sensor and back to it.

3.2.10 Operaton of the Module

1. Module should be connected to the power supply through Vcc port.
2. GND denotes the connection of the ground thus completing the electrical circuit.
3. Both Pins TRIG and ECHO are set low initially and these are set from the microcontroller.
4. A 10 microsecond high level pulse is triggered by making the TRIG pin high.
5. ECHO pin is enabled to seize the reflected triggered signal.
6. when the ECHO pin detects the positive edge of this signal and then again it should detect the negetive edge of the signal then the time is noted by its internal circuitery.

3.2.11 Ultrasonic Sensor HC SR-04 Specification

Operating Voltage	-5 volts DC
Operating Current	-15 milliamperes
Operating Frequency	-40KHz
MaximumDetection Range	-4 meters
Minimum Detection Range	-2 centimeters
Measuring Angle	-15’
Trigger signal	-10 microseconds
Dimensions	44X20X15 millimeters

3.2.12 MaxSonar LV series EZ-1

Figure 16 MaxSonar

3.2.13 MaxSonar Features

1. Objects with Zero distance are also detected.
2. Time to read the output for every cycle is less upto 50 milliseconds.
3. Easy to interface.
4. Measuring the distance, detection of objects in long distances, navigation of autonomous vehicles and many more applications.
5. Operates at 42KHz.
6. Power consumption is low.

3.2.14 Pin description

1. Pin_1: It should be left open for serial output.

2. Pin_2: This is a pulse width port and this pin is used to generate the timing signal.
3. Pin_3: Analog pin connected to the GPIO pin of the embedded board, this is used to produce the analog values.
4. Pin_4: If the signal is Low then the sensor stops operating, if high it continually output the values.
5. Pin_5: Connected to the 5V power supply.
6. Pin_6: Connected to the GND.

3.2.15 I2CXL Max Sonar MB-1202

This is the similar version of the Max Sonar MB-1202 but it consists of an I2C bus thus making it more reliable and less complicated to interface it to other sensors.

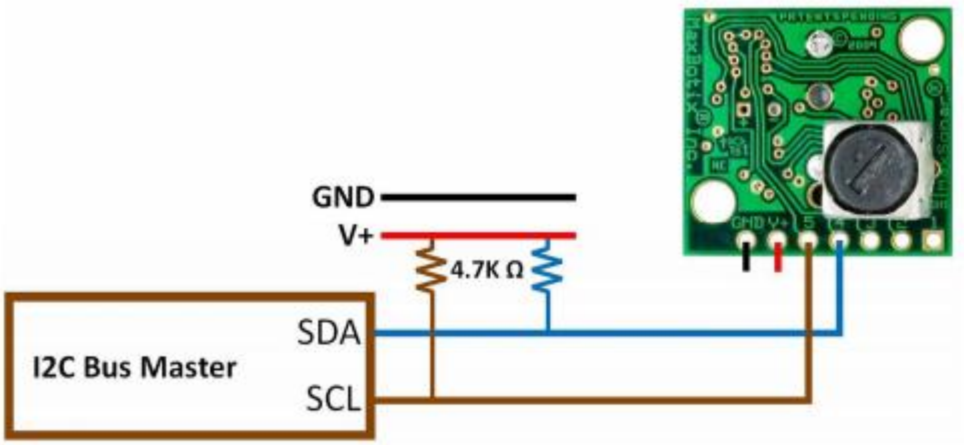


Figure 17 I2C bus connection

3.2.16 Features

1. The bus I2C is useful for the controll and limited pin connection of multiple sensors.
2. Gain variability is continous.
3. Timing to read the signal is around 25 milli seconds.
4. Less expensive.
5. Low power consumption.
6. Acoustic power is relatively high compared to the previous ultrasonic sensors.
7. Measuring the distance, detection of objects in long distances, navigation of autonomous vehicles and many more applications.

3.2.17 Pin description

1. Pin_1: This acts as a reset button, upon pressing this button the the GND is enabled thus clearing the register and resetting the whole system.
2. Pin_2: This provides the status indication for the present state of the ultrasonic sensor.
3. Pin_3: This pin is not used.
4. Pin_4: I2C bus data connection connected to the SDA equivalent of the embedded microcontroller.
5. Pin_5: I2C communication line for controllin the sensor operation.
6. Pin_6: Connected t the GND.

3.2.18 Comparison

Features	US SR-04	Max Sonar MB1000	I2C Max Sonar MB 1202
Zero Distance Measurement	No	Yes	Yes
Timing Cycle	100 milliseconds	50 milli seconds	25 milli seconds
Power consumption	Low	Low	Low
Operating frequency	40 KHZ	42 KHz	42 KHz
Max distance measurement	4 meters	6.45 meters	7.5 meters
Price Range	5$	10-15$	15-25$

By comparing the different types of ultrasonic sensors, I2C Max Sonar would be the correct choice to use it for this project since its specifications meet the design's requirement and it is more reliable than the others discussed above.

3.3 Radio Frequency Communication

Radio Frequency communication refers to wireless communication that uses an antenna to capture the signals which were previously transmitted from a radio transmitter. A radio tuner is a device which tunes to a particular desired frequency. Radio communication has changed the medieval period of technology to a modern era. There are small and large scale applications based on this technology.

Radio communication is possible because of the generation of electromagnetic waves which are in the wavelength range of 3Hz to 300GHz. These waves travel at a speed equivalent to that of light and does not need any physical or any sort of medium to travel. Some of the industries who use radio communication are broadcasting of television, Radar systems, Cellular phone networks, various Remote controls, Computer networks etc.

3.3.1 RF Transmitter IC-PT2262

This is transmitter chipset which is connected to the PT2272 chipset, operates using CMOS technology. This provides a large number of address codes thus avoiding the code collusion since this has 12-bit tristate address pins.

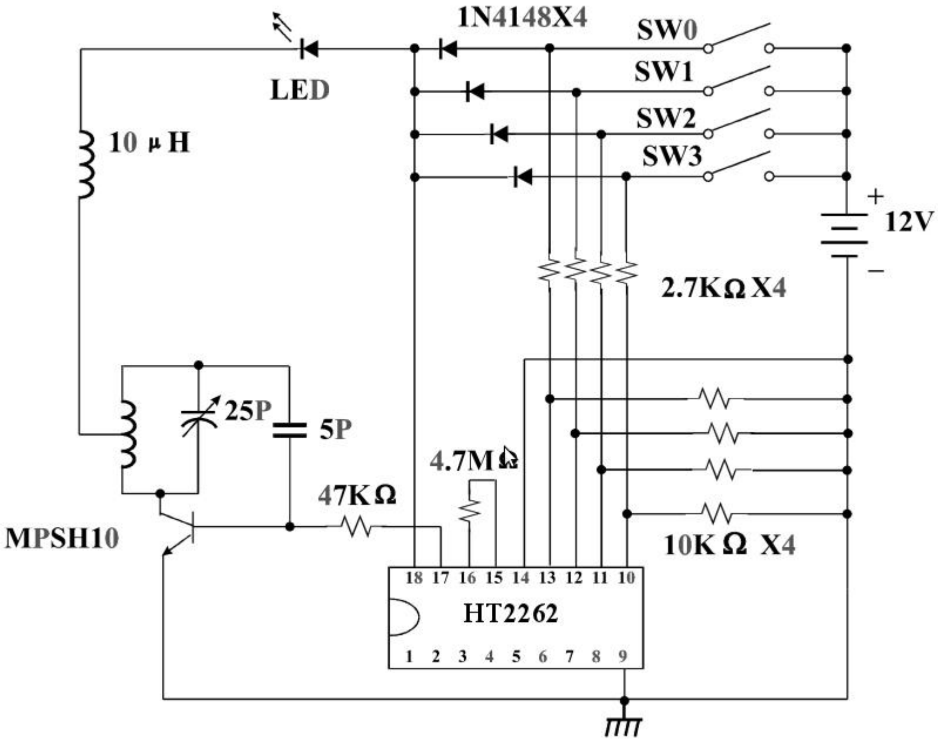


Figure 18 PT2262 Transmitter

Figure 19 Transmitter

3.3.2 RF Receiver IC PT2272

These receivers are easy to use since there is no programming required to activate this. They just have to be connected to the power supply and they are enabled.

The chipset PT2272 pairs up with the PT2262 transmitter chipset and both of them are Radio Frequency Receiver and Transmitter which are tuned at 315MHz frequency.

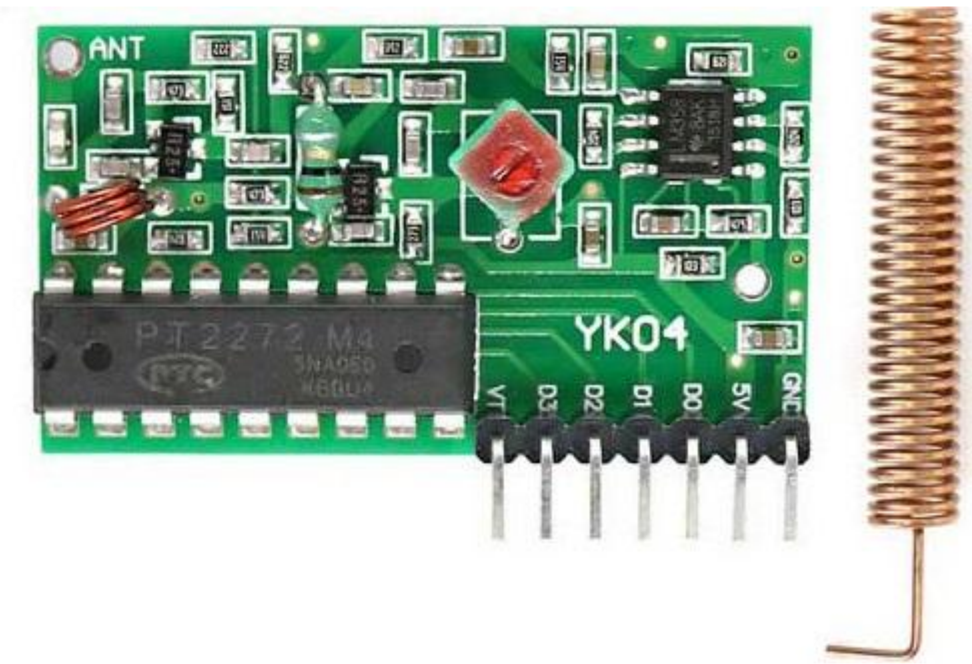


Figure 20 PT2272 Receiver

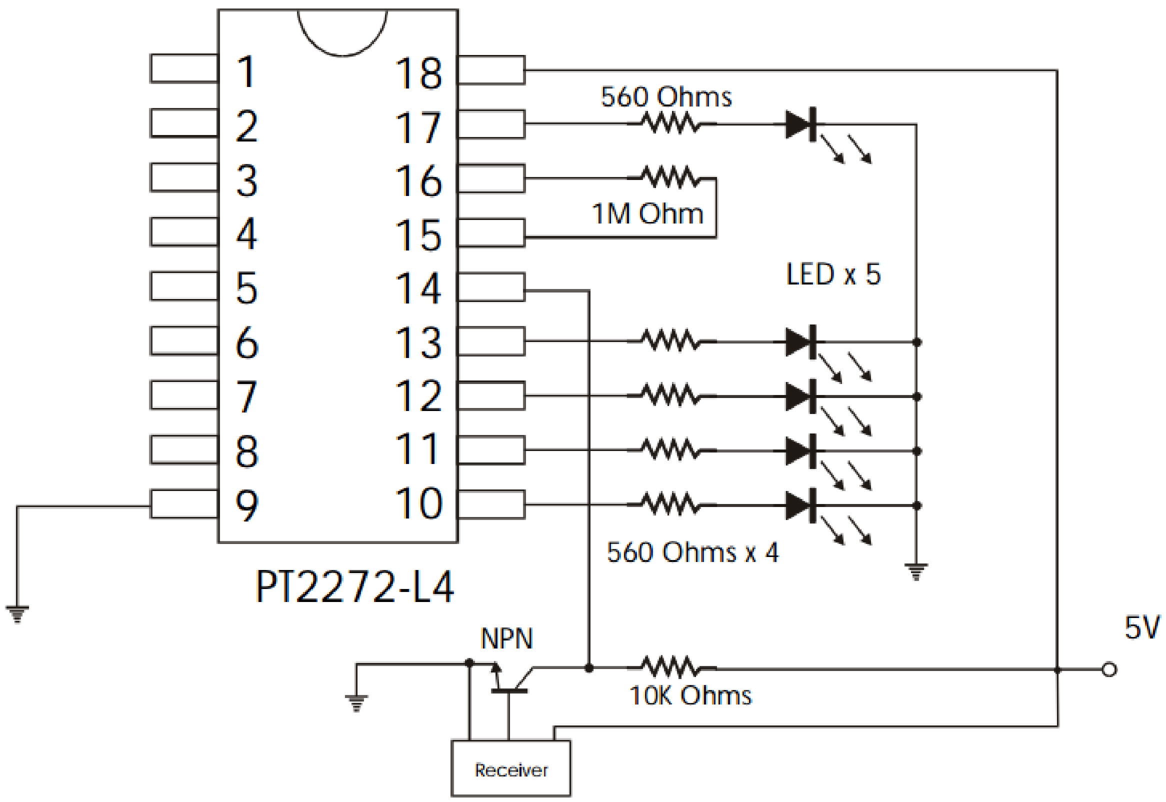


Figure 21 PT2272 IR application circuit

Figure 18 and Figure 19 shows the physical PT2272 acting as a receiver and the IR application circuit respectively.

3.3.3 Operation of Transmitter and Receiver

1. The receiver works with an operating voltage of 5 volts, therefore it has to be connected to 5 volt DC power supply and the GND pin should be connected to the ground thus completing the flow of current.
2. Four pins of the receiver correspond to the four buttons of the transmitter. Enabling the transmitter buttons enables the corresponding pin on the receiver.
3. These receivers can be connected to the Arduino board and each of the pins from receiver can be used to control four different operations.
4. Maximum range of signal detection is limited to 25 feet thus making them ideal for parking gate control system.
5. The pins are in toggle mode i.e. when a particular button is pressed the corresponding pin goes high and when that button is pressed again the mode of the pin goes low.

3.3.4 Specifications

Technology	-CMOS
Power consumption	-Low
Noise Immunity	-High
Tristate Address pins	12
Operating Voltage	-4-15 volts
Data pins	-up to 6

3.3.5 Applications

1. Security systems in car
2. Parking garage gate controller
3. Small scale remote controlled fan
4. Toys operated with remote controllers
5. Industrial remote controls
6. Educational accessory for beginners

3.4 Photoelectric sensor

As the name indicates this sensor reacts to the changes in the light intensity. These sensors have many applications depending on the type of requirement, they serve as the judge to decide the winning cars in the racing, to stop the closing of the parking gate and even the elevator when it encounters an obstacle, a wave of hand can turn on or off the lights, they serve for the automatic door opening in various industries, homes, convenient stores, offices etc. Photoelectric sensors have replaced the conventional inductive proximity sensors, to discover the existence or the absence of any foreign object these sensors are ideal since they use a ray of light or multiple beam of lights depending on the application.

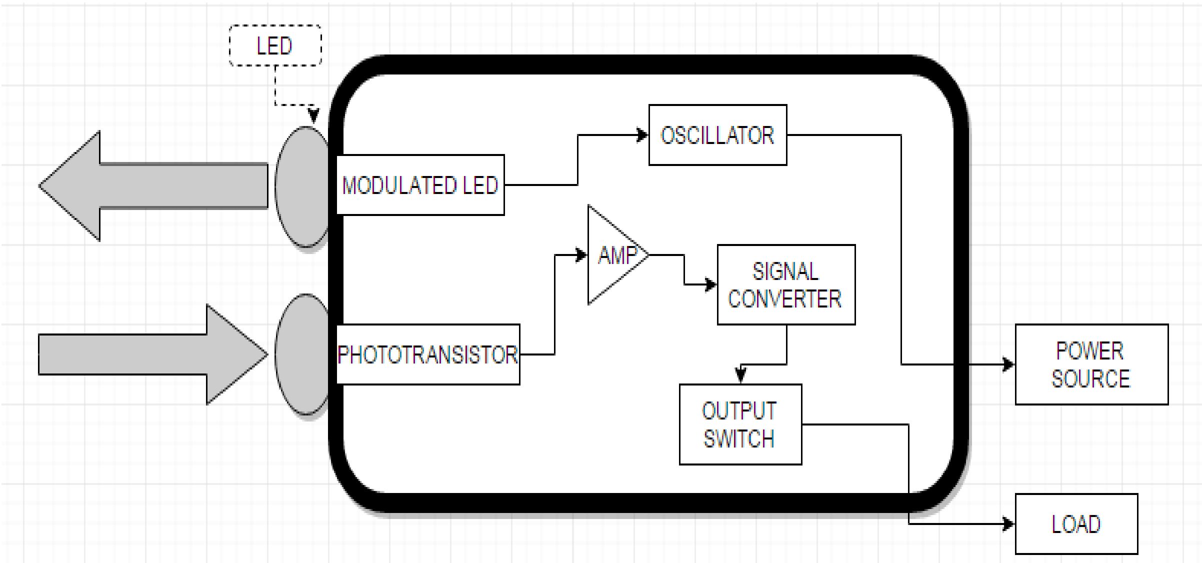


Figure 22 Photoelectric sensor

The above figure 20 shows the block diagram of the photoelectric sensor. It consists of the following components:

1. A power source to power the photoelectric sensor.
2. The oscillator is used to generate signals to monitor the LED.
3. A modulated LED acting as a transmitter which transmits a light beam.
4. The phototransistor senses the change in intensity and then the signal is sent to the amplifier, the amplifier amplifies this weak signal.
5. The signal converter sends the signal to the output switch after converting this signal equivalent to the controller.
6. Load is the controller which takes this signal as input and performs the required operation.

3.4.1 Through Beam sensors

A through beam sensor is one of the most widely used photoelectric sensor and has various applications based on the used requirement, they serve as the alternatives for the Magnetic sensor, Inductive sensor, capacitive sensor etc. The through beam sensors are used in small and large scale applications with the variations in the intensity of the beam and also they are available in single beam mode and high density multi beam mode.

The beams from the Through beam sensor can penetrate into dust, smoke etc. They don't have any restrictions to detect the object of any type of color, transparency levels or any sort of reflections.

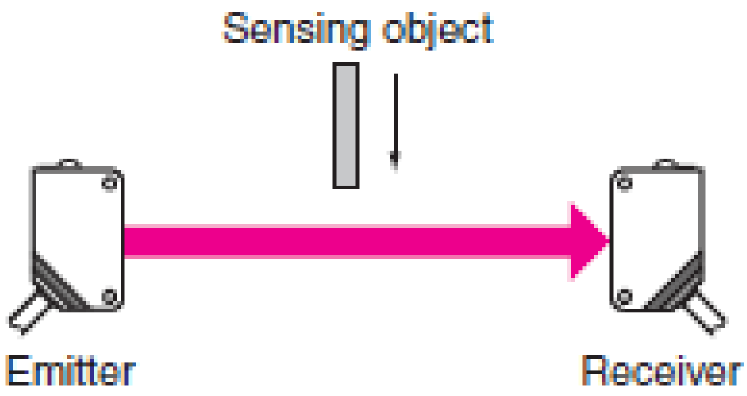


Figure 23 Through beam sensors

The above figure shows the Through beam sensor which consists of two parts i.e.

1. Emitter
2. Receiver

3.4.2 Working of the Through beam sensor

1. Emitter which emits high light beam in a straight direction with high intensity and the wavelength may range in the Infrared, Visible light or the Laser.
2. Receiver is mounted exact opposite to the emitter facing towards the emitter and both of them are calibrated to be in the straight direction.
3. Whenever an object passes through this beam there is change in the intensity level of the beam that hits the receiver or the receiver emitter connection is lost for a short moment of time, thus a signal is generated from the receiver which acts as a controlling input to the controller.

3.4.3 Features of Through beam sensor

1. Any changes in the sensing region of this sensor does not affect the sensing operations.
2. It is very stable and output generated instantaneously.
3. Shape, size, transparency of the obstacle does not affect the sensor's performance.

3.4.4 Applications

1. Used to open the doors automatically and also the turnstiles.
2. Open the Industrial doors automatically by detecting the movement of people.
3. Door opening in public transportation.
4. Escalators, Elevators use multiple beam through beam sensor.

3.4.5 M30 Through beam photoelectric sensors

Figure 24 M30 Through beam sensor

Through beam sensor Specifications:

Operating Voltage	-10-30 volts
Pins	-4
Distance Ranging	-10 meters
Type	-Through beam Photoelectric sensor
Current Rating	-200mA
Frequency Response	-100Hz
Beam type	-Single Beam

3.4.6 Enforcer Photobeam Sensor E-932 D33TBQ

This is a Through Beam photoelectric sensor consisting of a transmitter and receiver. It comes with a decent housing and this is mainly used for low level applications.

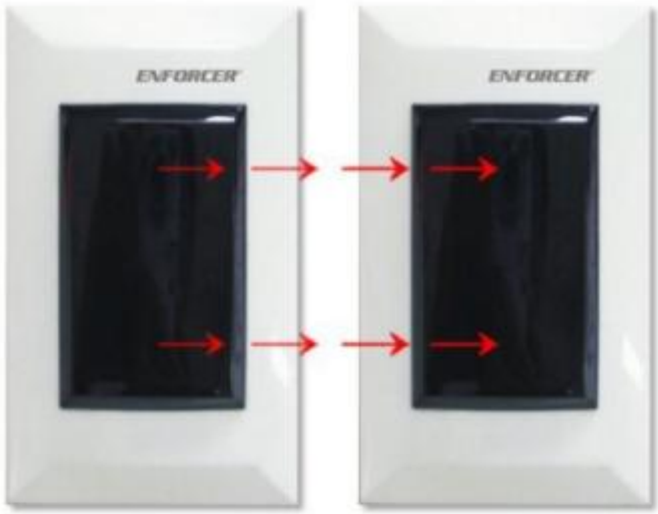


Figure 25 Enforcer E-932

3.4.7 Features of Enforcer

1. Easy to install and use.
2. Perfectly suited for office rooms, homes, parking lots etc.
3. 15-degree angle adjustment is possible when mounted vertically or horizontally.

3.4.8 Specifications

Sensor type	Dual Through Beam sensor
Range of sensing	10 meters
Operating Voltage	12-18V DC
Operating current	50 mill amperes
Time to response	10 milli second
Alignment angle	15-degree
Operating Temperature	-10-50-degree Celsius
Switching capacity	1ampere at 120VAC

3.4.9 Carlo Gavazzi P-30 Photoelectric sensor

This photoelectric sensor is a high quality sensor built in order to sustain a severe environment. It is encased in a hard plastic material thus making it strong and highly resistant to application of gases or any cleaning agents.

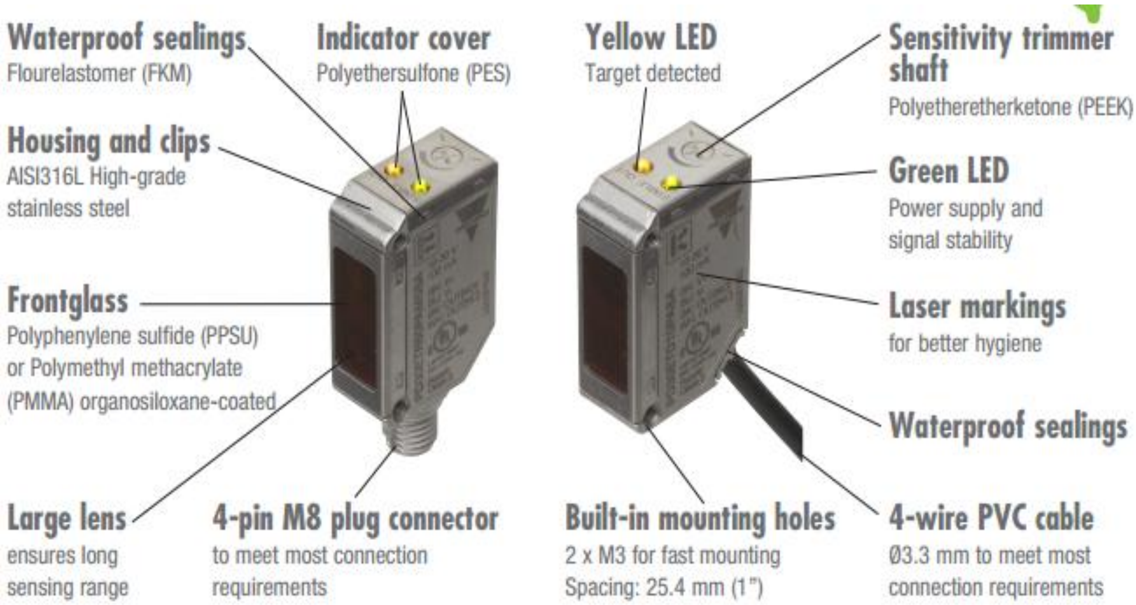


Figure 26 P-30 Carlo Gavazzi sensor

3.4.10 Features

It is available in both cable and plug versions.

The light beam is a modulated infrared of 850 Nano meter.

3.4.11 Specifications

Operating voltage	10-30VDC
Sensing Range	6 meters
Operating Temperature	-10-50-degree Celsius
Material	Polyphenylene sulphide
Pin	4-pin
Humidity	35-95%RH
Hysteresis	Less than 10%
Dimensions	11*31.5*21 millimeter
Weight	100 grams

3.4.12 Comparison

Features	M-30	Enforcer	Carlo Gavazzi
Price	5$	30-35	70-80$
Reliability	No Warranty	1yr-Warrenty	1yr-Warrenty
Parking gate usage	Yes	Ideal	Expensive
Operating Voltage	10-30V	12-18V	10-30V

Through beam sensor M-30 can be used for experimental purposes but Enforcer is the right choice to be used for this project.

3.5 Magnetic Proximity Sensor

A sensor which detects the presence of the obstacles or the objects near its field of operation is a proximity sensor. These proximity sensors have revolutionized the modern industry thus utilizing them in various applications. Wireless proximity sensors are those who are not connected to any physical medium but still detect presence of foreign obstacles and send the signal to the controller, here there is no contact made by the obstacle or the object to the sensor.

Magnetic proximity sensors belong to the wireless proximity sensors family, these are basically used to detect the magnetic entities present near them. The area of detection of these magnetic entities depends on the intensity of the magnetic sensor.

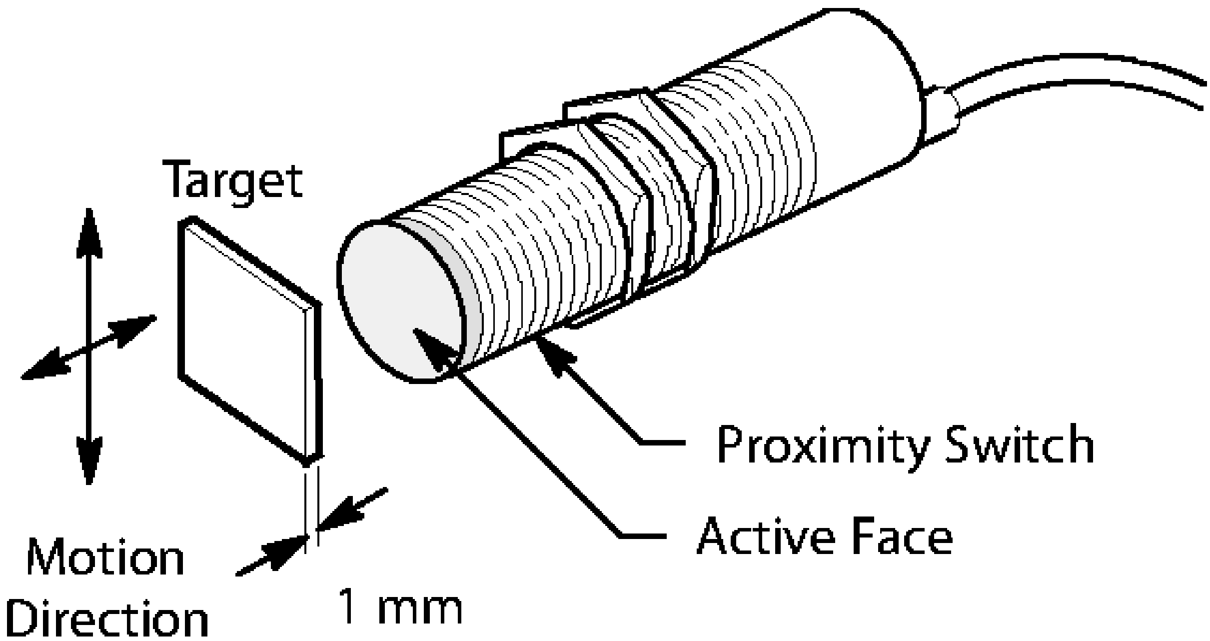


Figure 27 Magnetic proximity sensor

The above figure 23 shows the magnetic proximity sensor consisting of proximity switch and a target.

3.5.1 Working of Magnetic proximity sensor

1. Magnetic sensor is connected to a power source where the positive is connected to the Vcc, GND is connected to the ground and the third pin is connected to the controller.

2. Whenever the sensor encounters a magnetic field there is a signal generated which in turn is connected to the microcontroller thus the microcontroller detects this and performs the user defined operation.

3.5.2 59140 Miniature Magnetic sensor

Figure 28 59140 Magnetic Sensor

3.5.3 Features

1. Acts as a position senor based on the application.
2. Sensitivity variation can be done depending on the user's requirement.
3. Independent of the presence of non-ferrous materials such as glass, wood, paper etc.
4. Ideal for Microcontroller design

3.5.4 Applications

1. Sensing of the limitation in automated gates.
2. Sensing of the position.
3. Switches in doors.
4. Actuators with linear motion.

3.5.5 Monnit wireless magnetic detection sensors

Figure 29 Monnit wireless magnetic switch

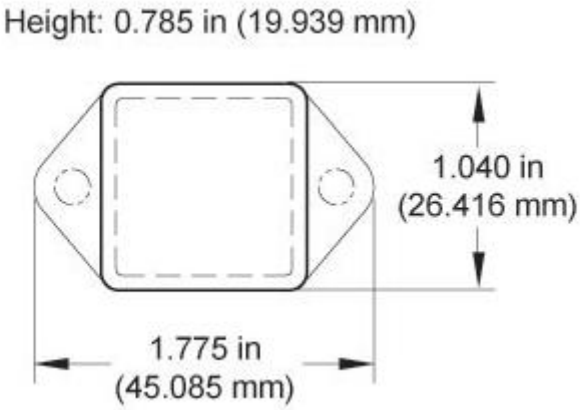


Figure 30 Dimensions

3.5.6 Specifications

Type of the Battery	Replaceable cell
Battery Voltage	3-5VDC
Operating Temperature	-20-60-degree Celsius
Operating Frequency	900MHz
Dimension	15in*1in*0.7in

3.5.7 Pin Connections

1. Magnetic Sensor has inbuilt connections for GND and Power supply.
2. The external wire extending from the sensor's housing is connected to the Arduino Microcontroller embedded system.

3.5.8 Manual Programming

The main objective of the manual programming is to define the open and close position of the gate and to teach the controller to register these positions in its memory.

Procedure to perform the Manual Programming:

1. Three push buttons are connected to the Arduino Micro controller i.e. PB_0(Push Button 1), PB_1(Push Button 2) and PB_2(Push Button 3)
2. PB_1 is used to configure the Open position and PB_2 is used to configure the Close position of the gate.
3. Pressing the PB_0 the manual programming of the Microcontroller is enabled.
4. The gate is pushed open to the Open position and the PB_1 is pressed, now the controller registers this as is its new Open position.
5. The gate is pulled to Close position and the PB_2 is pressed, now the controller again registers as its new closing position.
6. The PB_0 is again pressed for five seconds, this indicates that the manual programming is disabled.

3.5.9 Limit Switch

GTO R4421 Limit switch

Figure 31 GTO R4421 Limit Switch

These are the electrical limit switches and are commonly used in parking gates. Figure 31 shows the Alecko Limit switches.

Pin Connections:

1. GND and 5V are connected to the pins protruding out of the limit switches.
2. The small plate is connected to the Microcontroller system.

3.5.10 Working Procedure

The Limit switches are powered up and are placed in their positions as shown in the figure 32. The extreme right movement is Close position and the extreme left is the Open position.

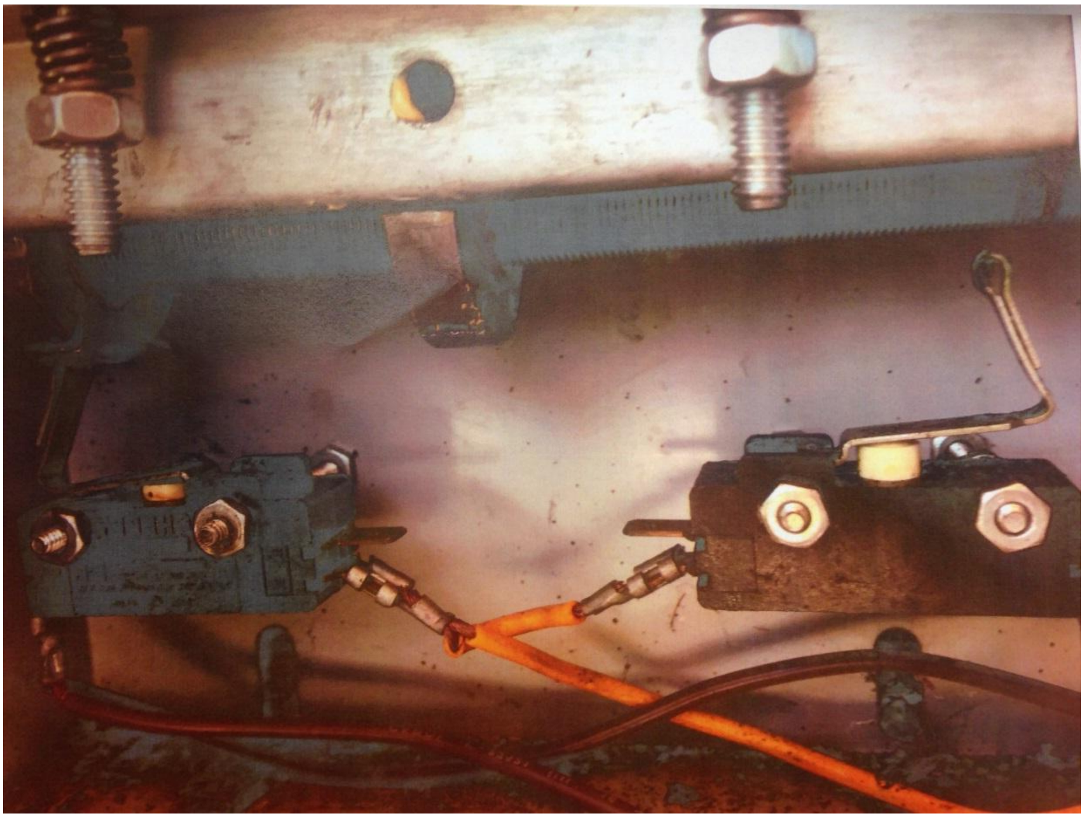


Figure 32 Working of Limit Switches

The movement of the gate makes the cylinder to rotate and the gear mechanism is scaled down i.e. the movement of the gate from close to open position is shifted down in this arrangement to few inches.

A metal knob is mounted down to the cylinder thus the rotatory movement of this cylinder is converted to the linear movement of the knob.

When the gate is in closed position, the knob is in contact with the extreme right limit switch and the knob is in contact with the extreme left limit switch if the gate is in open position.

When any of the limit switches is in contact with the knob, a signal is sent to the Micro controller system. The microcontroller system generates a stop signal and outputs to the motor thus the motor stops to rotate.

3.5.11 Comparison

Features	59140 Proximity	Monnit Magnetic	GTO R4421
Operating Voltage	5VDC	5VDC	5VDC
Operating Field	Magnetism	Magnetism	Electricity
Installation	Easy	Easy	Difficult
Manual Programming	Not Required	Not Required	Required
Reliability	Reliable	Reliable	Reliable
Price Range	15-80$	20-25$	20-25$

Depending on the user skill and requirement any of the above Limit switches can be used.

3.6 Motor Driver

An electrically operated switch is called a relay. Electrical isolation is provided for the two circuits since they are mechanically operated by an electromagnet. A motor driver is a circuit enabling the control over an AC motor which operates with the power from the AC mains from a microcontroller which is generally operated with 5 volts.

The motor driver is an amplifier too which steps up the low current into high current thus increasing the gain to run a motor.

3.6.1 2-channel Relay Motor Driver

Figure 33 Saint smart Motor driver

3.6.2 Pin Map

Vcc	-Connected to the 5V on the Arduino UNO system.
GND	-Common ground connection.
IN1, IN2	-Controlling pins connected to the digital pins in Arduino.
NC (Normallyconnected)	-Both are connected to the positive of the AC mains.
NO (Normally open)	-Both are connected to the ground from the AC mains.
COM (Common)	-Connected to the motor.

IN1, IN2 are the motor driver pins which are connected to the Arduino or any microcontroller to control the motor operation in terms of the forward, backward, stop and speed control of the motor.

3.6.3 Motor driver Specifications

Load Rating	-240 volts / 7 amps
	-125 volts / 10 amps
	-28 VDC / 10 amps
Relays	-2
Signal Control	-Transistor-Transistor logic
Time delay	-5-10milliseconds
Microcontrollers	-supports Arduino. ARM, Raspberry pi, 8051, AVR
Indicator	-Onboard LED

3.6.4 L298 Motor driver

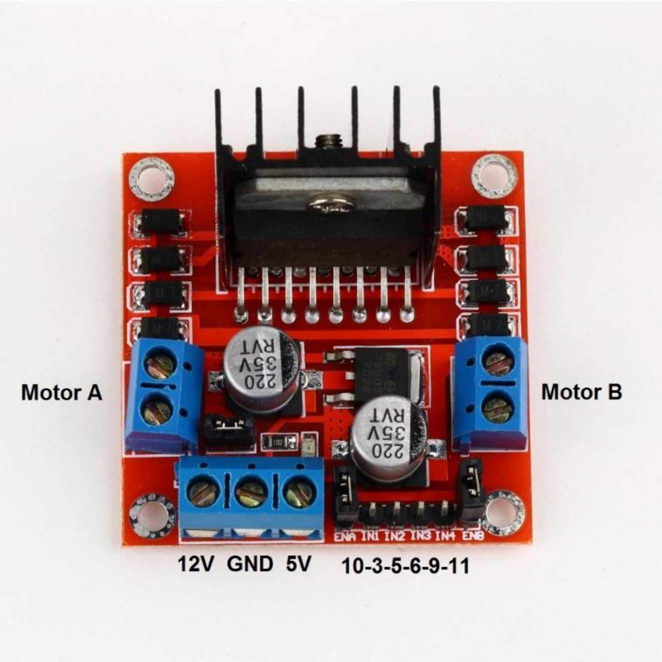


Figure 34 L298 Motor Driver

3.6.4 Pin Map

Pin connection between the motor driver, motor and the embedded board.

1. GND is connected to the common ground.
2. Two motors can be connected and controlled i.e. Motor and Motor B.
3. IN1 and IN2 are connected to the Embedded board which controls the rotation of the motor.

4. Two connections from the DC motor are connected to the two pins of the motor driver which is named as Motor A.

3.6.5 Specifications

Operating Voltage	5V
Operating current	0-36milliamperes
Maximum Power	25W
Weight	26 grams
Dimensions	43*43*26 millimeter

3.6.6 L293D Motor Driver

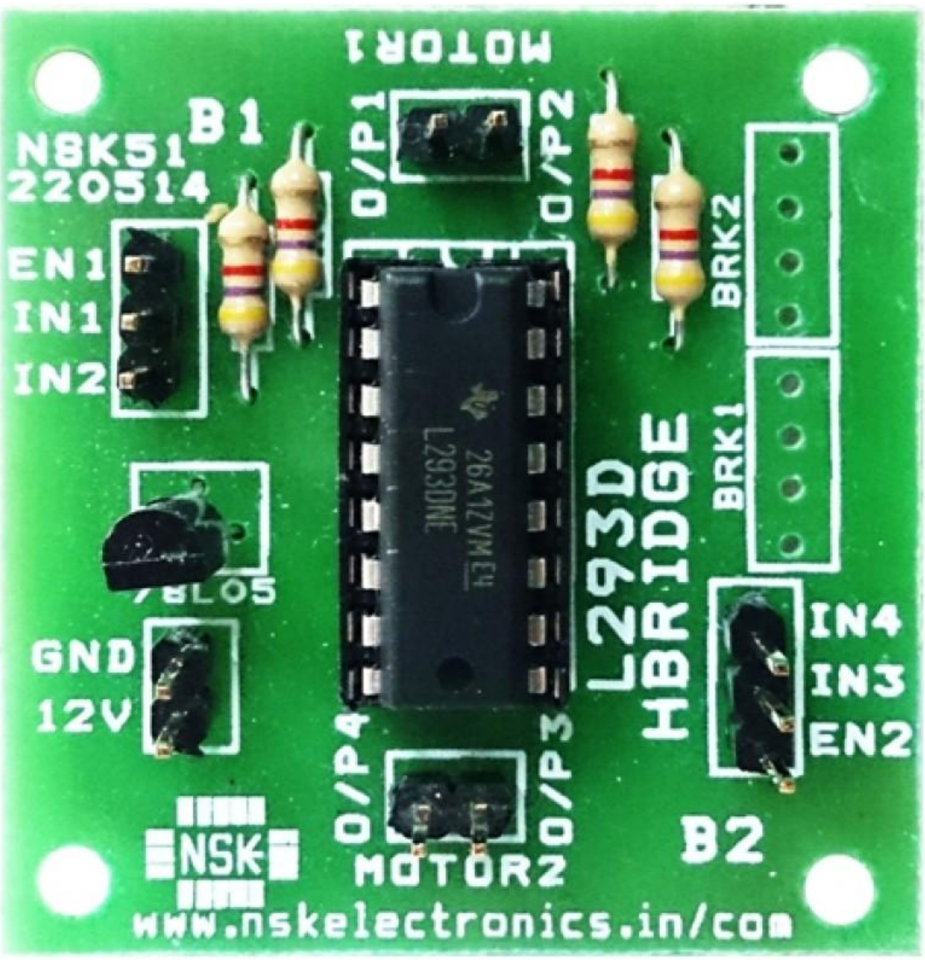


Figure 35 L293D Motor Driver

3.6.7 Specifications

Operating Voltage	5-30VDC
Peak output current	2 amperes
Junction Temperature	150-degree Celsius
Weight	25 grams
Maximus Power	25W

3.6.8 Comparison

Features	**Saint smart-2CH**	**L298D**	**L293D**
Operating Voltage	5-12V	5V	5-30VDC
Reliability	More reliable	Reliable	More reliable
Parking gate control	Ideal	Can be used for experimental purposes	Can be used for experimental purposes
Price range	10$	5-10$	5-10$
Motor Operation	Both AC and DC	DC motor only	DC motor only

By considering the above factors Saint-smart 2-channel relay motor driver is used for this project both for experimental and real time applications.

CHAPTER 4

DESIGN OF GATE CONTROLLER

4.1 A New Design

The main objective of this project is to design the parking gate controller to be is cost effective, uncomplicated, easy to install and most importantly to have an automatic gate opening which is limited only to automobiles.

In order to design this, the smart devices defined in the chapter 3 are used. Two ultrasonic sensors are used for the detection of car and this can be modified by using more ultrasonic sensors as well.

4.2 Placing the devices for new design

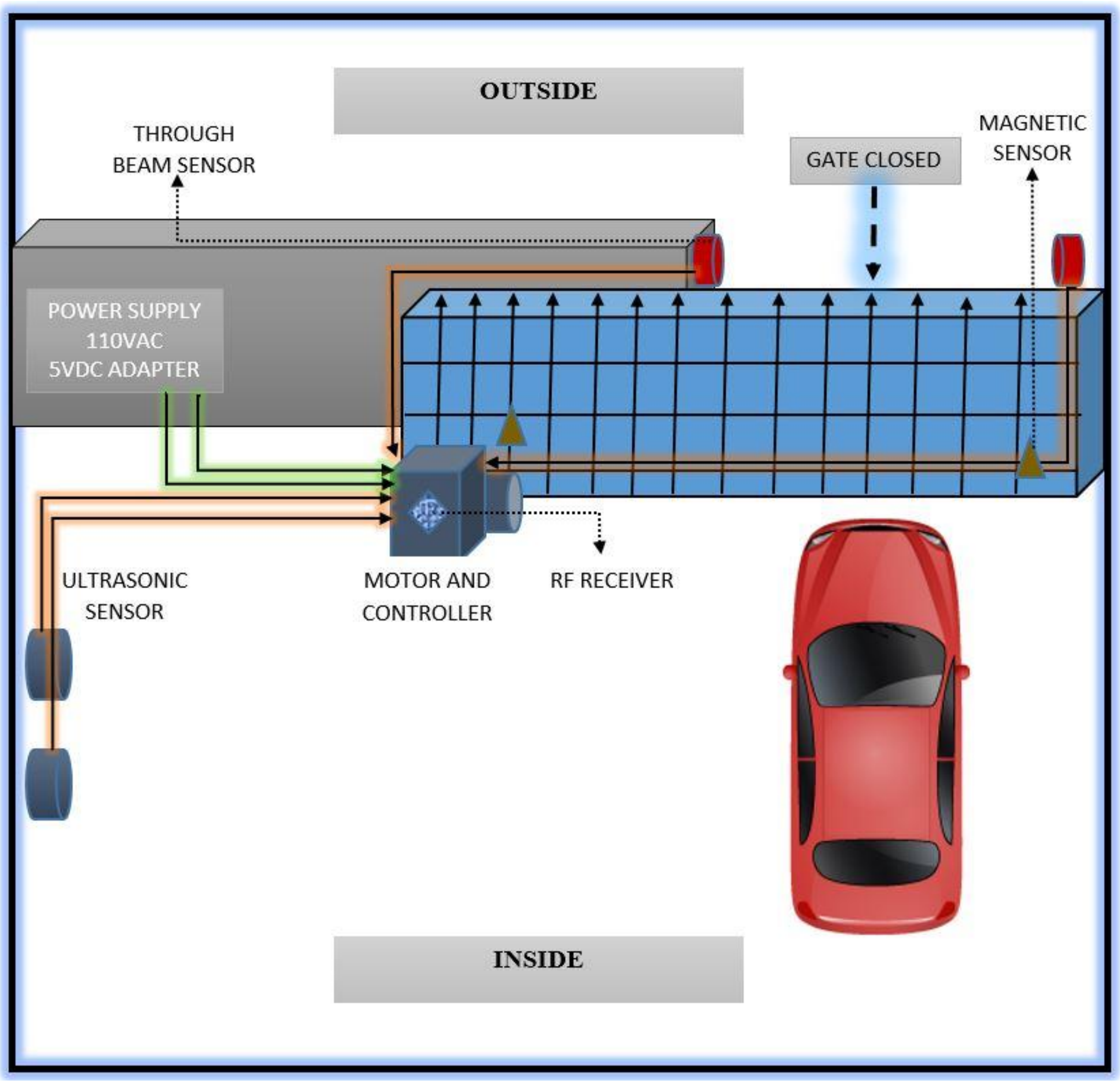


Figure 36 Design outline

The above Figure shows the scenario of the Controller and various other devices that are connected to and from the controller.

a. The controller and the motor are placed in a single housing and then the whole setup is placed in front of the gate from inside.
b. The gate is mounted on the wheels and the length of the gate is generally half of the total open horizontal length.
c. A gear rack is mounted at the bottom of the gate and this gear rack is almost the length of the gate.
d. The controller-motor setup is placed in the middle of the space in such a way that the pinion on the motor is aligned with the gear strip.
e. Ultrasonic sensors are placed in the user desired position but these are placed in such a way that they are covering the length of the automobile.
f. RF receiver is also mounted with the controller but can also be mounted in the user desired position but has to make sure that it is not reachable and not exposed to the atmosphere.
g. The Through beam photoelectric sensor is placed at the edge to edge of the gate so that it is able to scan along the length of the gate way.
h. Magnetic sensors are placed at the bottom.
i. The First magnetic sensor is placed at the left most side and its receiver is again mounted to the controller housing facing opposite to the gate. Both of them should be calibrated at first. This remains to be the closed position of the gate.
j. The second Magnetic sensor is placed at the rightmost edge of the gate, this sensor should be mounted just above the line of the first sensor.
k. The receiver for the second magnetic sensor should be placed above the first receiver.
l. RF transmitter remains with the user at all times.

4.3 Pin mapping of the Hardware

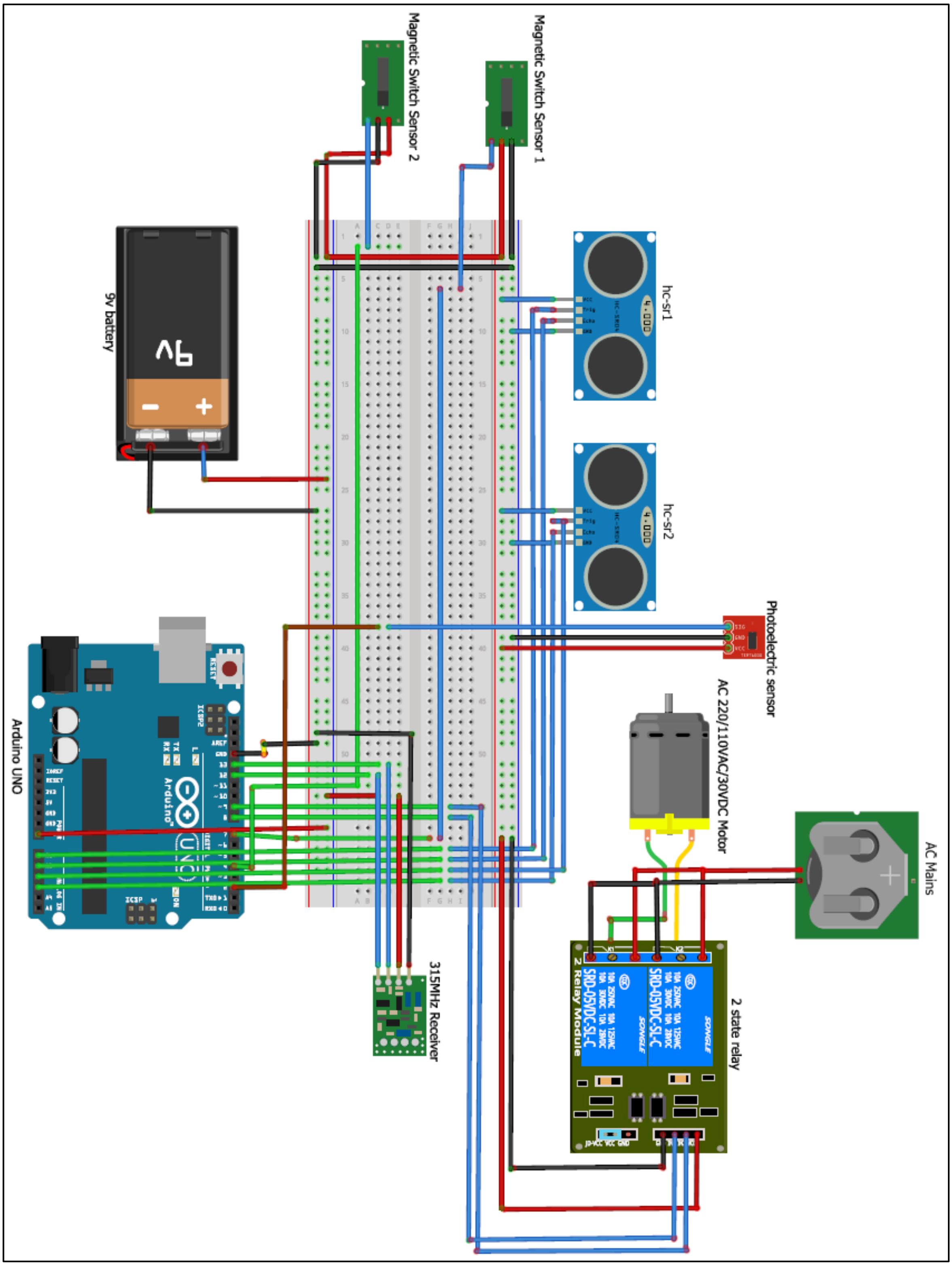


Figure 37 PIN MAP

The above Figure 37 shows the pin mapping of the hardware setup.

The devices and the components used for this design are as follows:

a. Arduino Microcontroller Embedded system
b. 315Hz Receiver and Transmitter module
c. Magnetic Proximity sensor
d. Through beam Photoelectric sensor
e. Ultrasonic sensor
f. 2-channel relay Motor driver
g. AC/DC Motor

All these devices have to be connected, programmed and placed in a certain specific procedure to make this design work. The circuit connections are done as follows:

1. Arduino

a. Arduino Microcontroller is connected to the 5-9V adapter.
b. The adapter is connected to the AC mains.

2. Ultrasonic sensor #1

a. Sensor's GND pin is connected to the GND i.e. ground of the microcontroller.
b. 5V of the microcontroller is connected to the sensor's 5V pin. Sensor is powered up, thus the circuit's current flow is completed.
c. TRIG pin of the sensor is connected to the A0 pin of the Arduino board, this is digital enabled.
d. ECHO pin of the sensor is connected to the A1 pin of the Arduino board, this is digital enabled.

3. Ultrasonic sensor #2

e. Sensor's GND pin is connected to the GND i.e. ground of the microcontroller.
f. 5V of the microcontroller is connected to the sensor's 5V pin. Sensor is powered up, thus the circuit's current flow is completed.
g. TRIG pin of the sensor is connected to the A2 pin of the Arduino board, this is digital enabled.
h. ECHO pin of the sensor is connected to the A3 pin of the Arduino board, this is digital enabled.

4. RF 315MHz Receiver

a. GND pin of the receiver is connected to the common GND i.e. ground of the Arduino board.
b. Vcc of the sensor is connected to the 5V supply of the board, sensor is powered up using this port.

c. D0 and D1 are the operation pins and are connected to the Arduino board to 12 and 13 pin on the board. They are enabled whenever the corresponding buttons are pressed from the transmitter.

5. Through beam sensor

a. Black wire pin is connected to the GND (ground) of Arduino.

b. Red wire pin is connected to the 5V of the Arduino board.

c. Orange wire is connected to the Pin 2 on the board providing an interrupt function providing the top level priority.

d. High end through beam sensors have to be connected to the 24V adapter depending on the sensor's specifications.

e. The control wire would be connected to the pin 2 on the Arduino board.

6. Motor Driver Relay

a. The Motor driver is powered up by connecting the Vcc and GND on board to the Arduino controller's 5V and GND pins.

b. IN1 and IN2 are the controlling pins to Start, Stop and control the Speed of the motor and are connected to the Pin 8 and Pin 9 of the Arduino board.

c. Middle pins of the relays are connected to the 2 pins of the Motor and the left and right pins of these relays are connected to the AC mains or the DC adapter based on the type of the Motor.

4.4 Working of the Design

As discussed in the section 4.2 all the components and the devices are placed in their positions and powered up. This Gate controller should work irrespective of the car's entrance either from inside or outside therefore the design discusses for all the scenarios.

#1 Scenario

Car's Entrance -from OUTSIDE

Gate Position -Closed

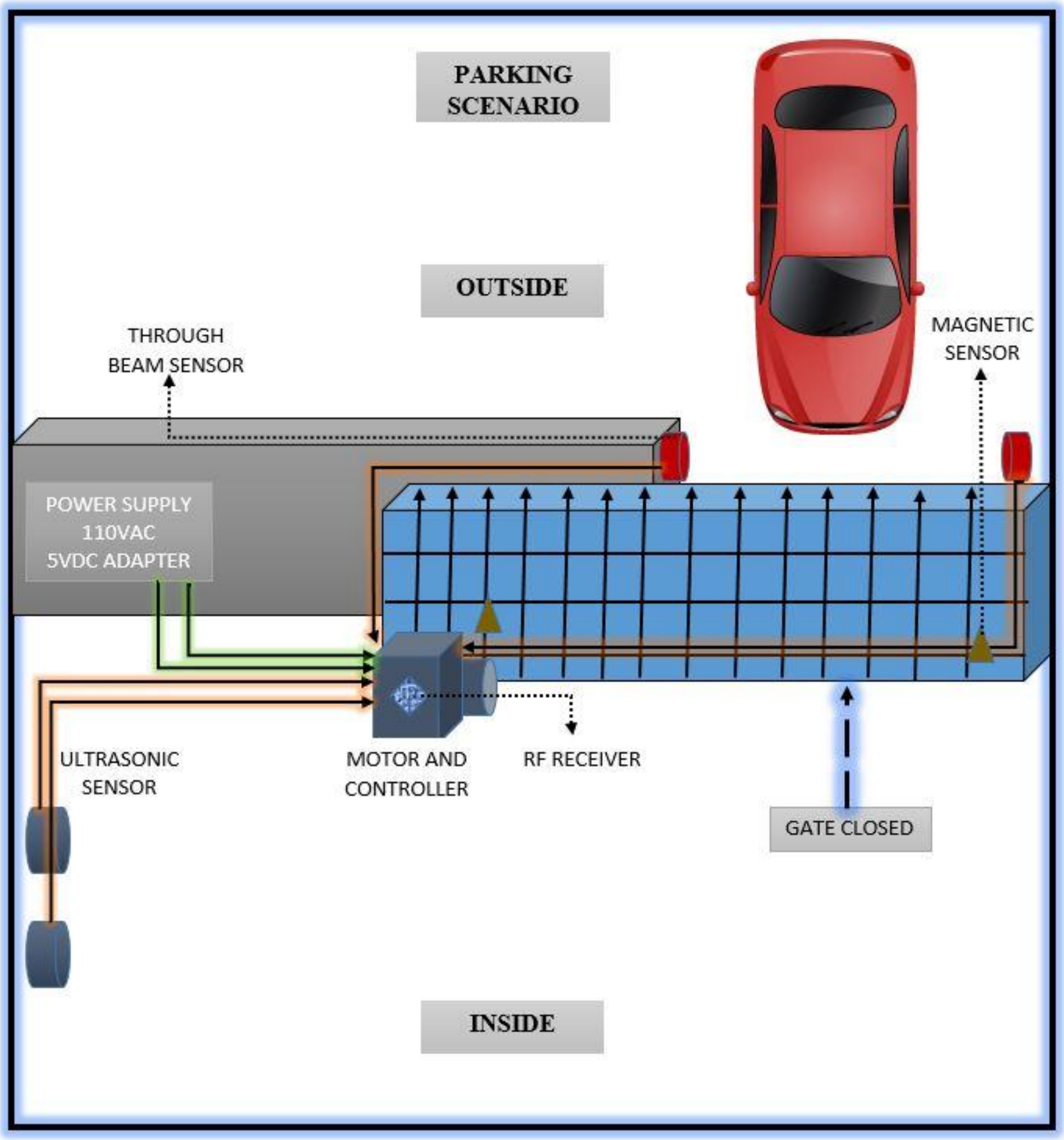


Figure 38 Car entering from Outside #1

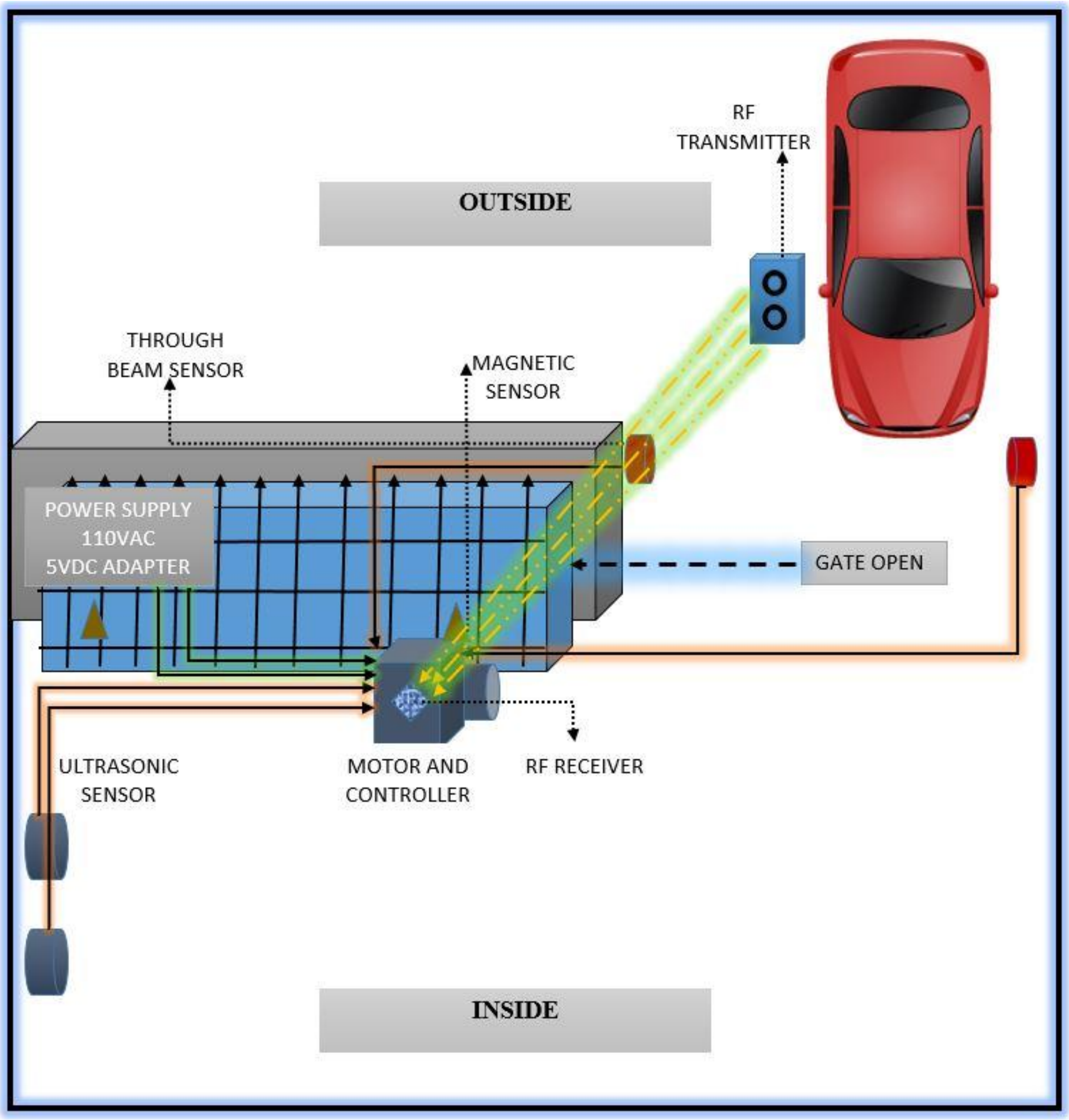


Figure 39 Car entering from outside #2

Figure 38 illustrates the position of all devices and their placement.

The gate remains closed even if the car or any automobile approaches near the gate and the Figure 39 shows the opening of the gate.

When the RF transmitter's button is pressed a radio signal is transmitted into the space which is received at the RF receiver and then an open signal is sent to the Arduino controller which in turn outputs an open signal to the Motor driver.

The Motor connected to the motor driver rotates on its axis and the pinion connected to the shaft rotates on the rack thus opening the door.

The gate opens to that extent until the moment when the magnetic sensor receiver senses the magnet mounted on the left positon of the gate.

The receiver sends a signal to the controller, the controller sends a Stop signal to the motor through Motor driver, thus stopping the movement of the motor.

The controller then starts the timer for about 25-45 seconds and then a Close signal is sent to the motor through motor driver thus the motor starts rotating anti clockwise direction depending on the position of the gate.

To stop the sliding of the gate the second magnetic sensor is used. When this magnetic receiver senses the approach of the second magnetic which is mounted to the right position on the gate a Stop signal is generated by the controller and the Motor stops its rotation.

#2 scenario

Car’s Entrance	-from INSIDE
Gate Position	-Closed

The Figure 40 illustrates that the car is trying to exit from inside.

To open the gate the driver from the automobile can press the transmitter independent of the car’s position with respect to the ultrasonic sensors.

Even if the car is not in the field of ultrasonic sensors the gate opens up because of the usage of RF transmitter.

When the transmitter button is pressed the receiver detects the encoded signal from the transmitter and compares this with the signal pattern which is stored in its memory. If the signal matches then a Logic High signal is sent to the controller. The controller is programmed in such a way that whenever it receives a Logic High signal from the receiver it generates a logic high signal and outputs to the motor through the motor driver.

The Motor opens the gate through its gear and pinion mechanism. The Magnetic proximity sensors are used to monitor the open position of the gate.

The car or an automobile can exit out since the gate remains in open position for about 35-45 seconds. After the time limit the controller sends a Logic Low signal to the Motor to close the gate. The gate can be opened again with the application of the transmitter button. This seizes the operation of the gate from closing and starts to open again because the transmitter and through beam sensor have the top priority levels. An interrupt signal is generated in the controller upon the application of transmitter button and this signal reverses the function of the motor thus sliding the gate to the open position.

The controller again starts its timer at this point and waits for 35-45 seconds and then generates a Logic Low signal which is sent to the motor driver to rotate the motor to clockwise direction to close the gate using its rack and pinion arrangement. Magnetic sensors are used to monitor the position of the gate.

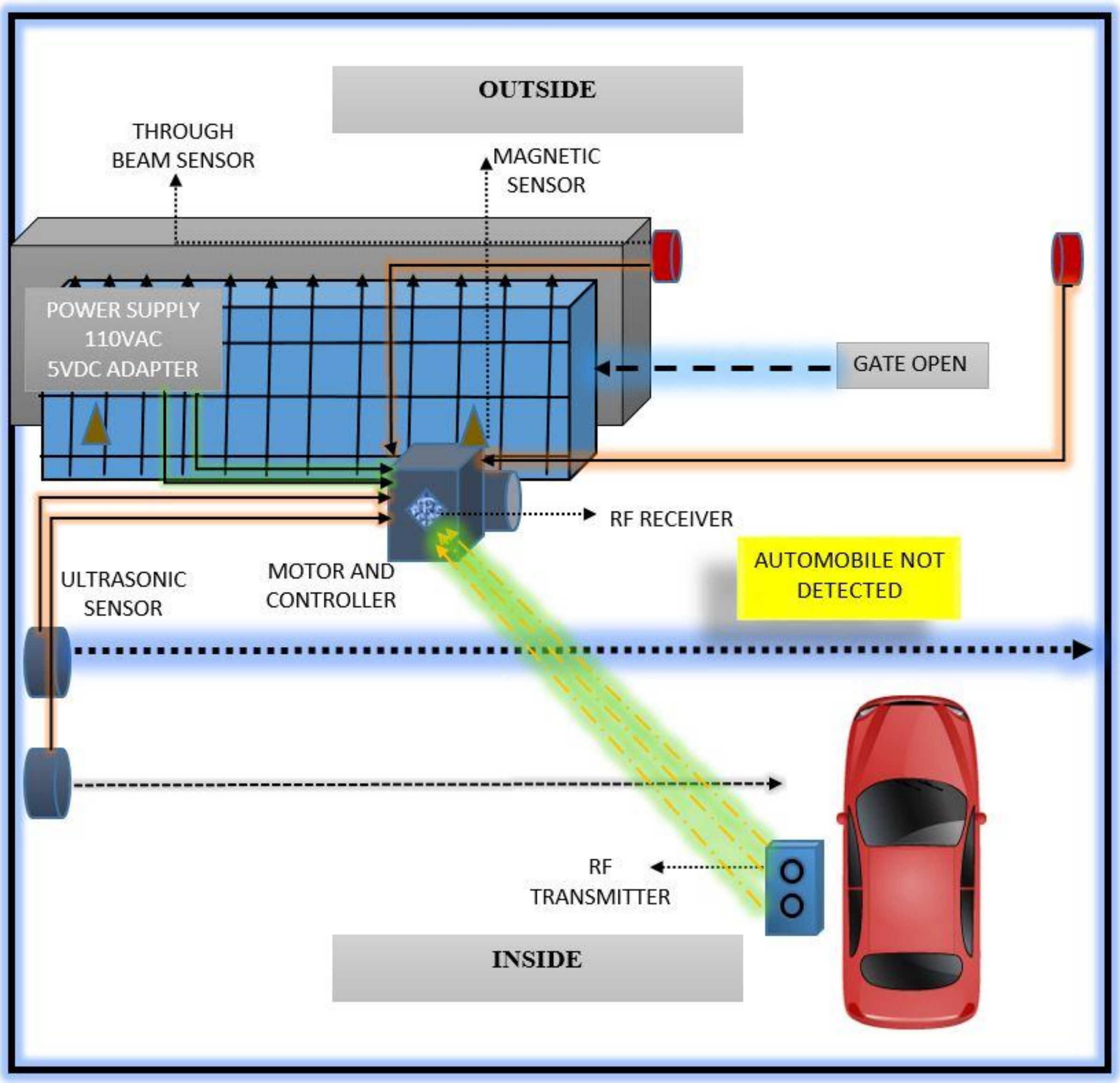


Figure 40 Gate open from inside RF transmitter

In this scenario the gate opens because of the application of the transmitter and this is irrespective of the function of the ultrasonic sensors whether they detected the car or not.

#3 scenario

Car's Entrance	-from INSIDE
Gate Position	-Closed

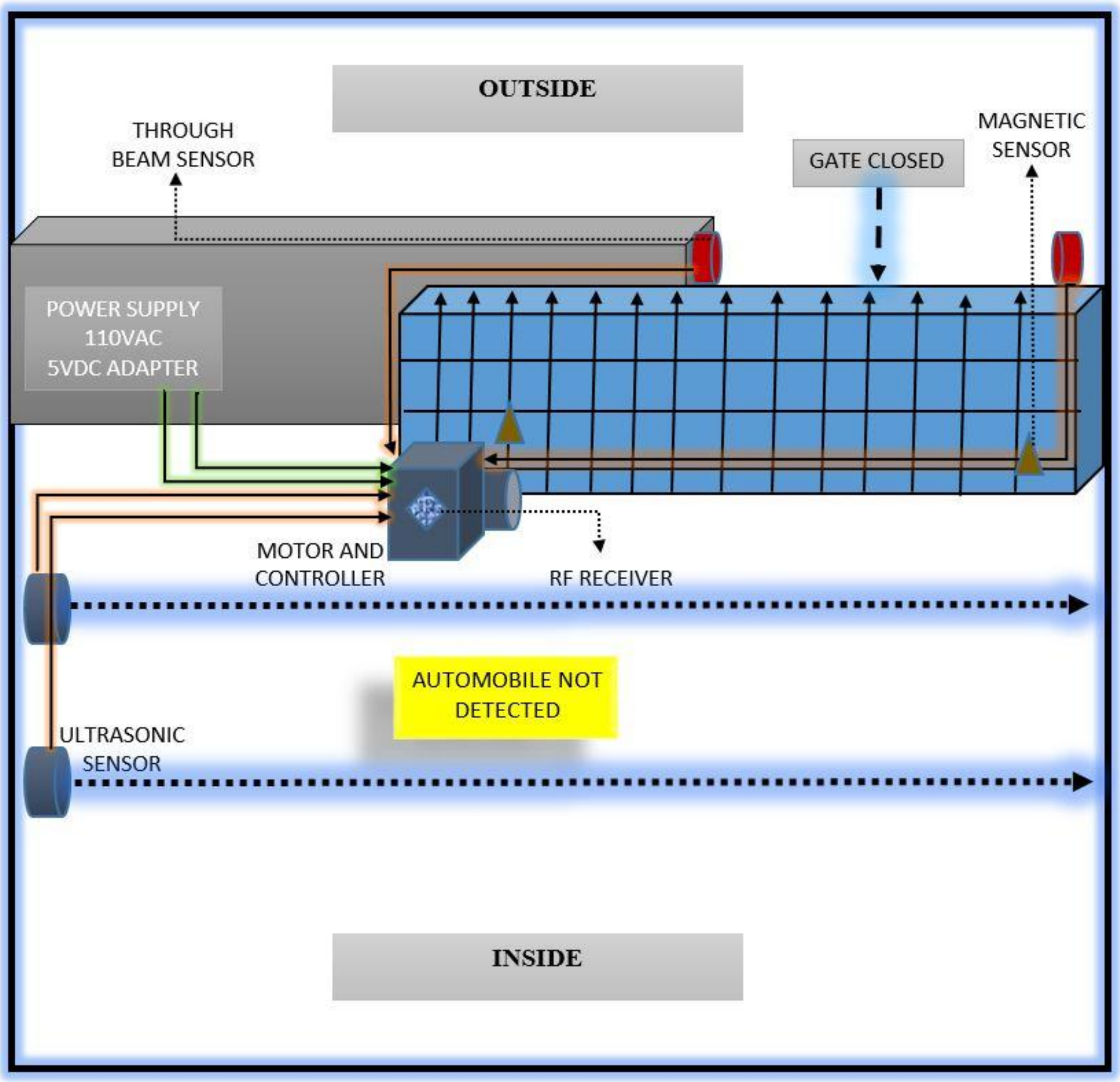


Figure 41 IDLE mode

In order to open the gate from inside the parking lot, RF transmitter can be used, but that contradicts the whole design, since the design objective of this project was to detect the car by the system itself.

The ultrasonic sensors are mounted in such a way that both of them are measuring the distance to the car. When there is no automobile or when the whole system is in idle mode the ultrasonic sensors output the distance from their mounted positions to the wall or the open space, this distance is the threshold value calibrated into the controller thus making it to generate a Logic low signal to keep the gate closed. Figure 41 shows the idle mode of the system where the ultrasonic sensors either measure a distance to the walls or just output an error signal since they have a limitation of measuring distance.

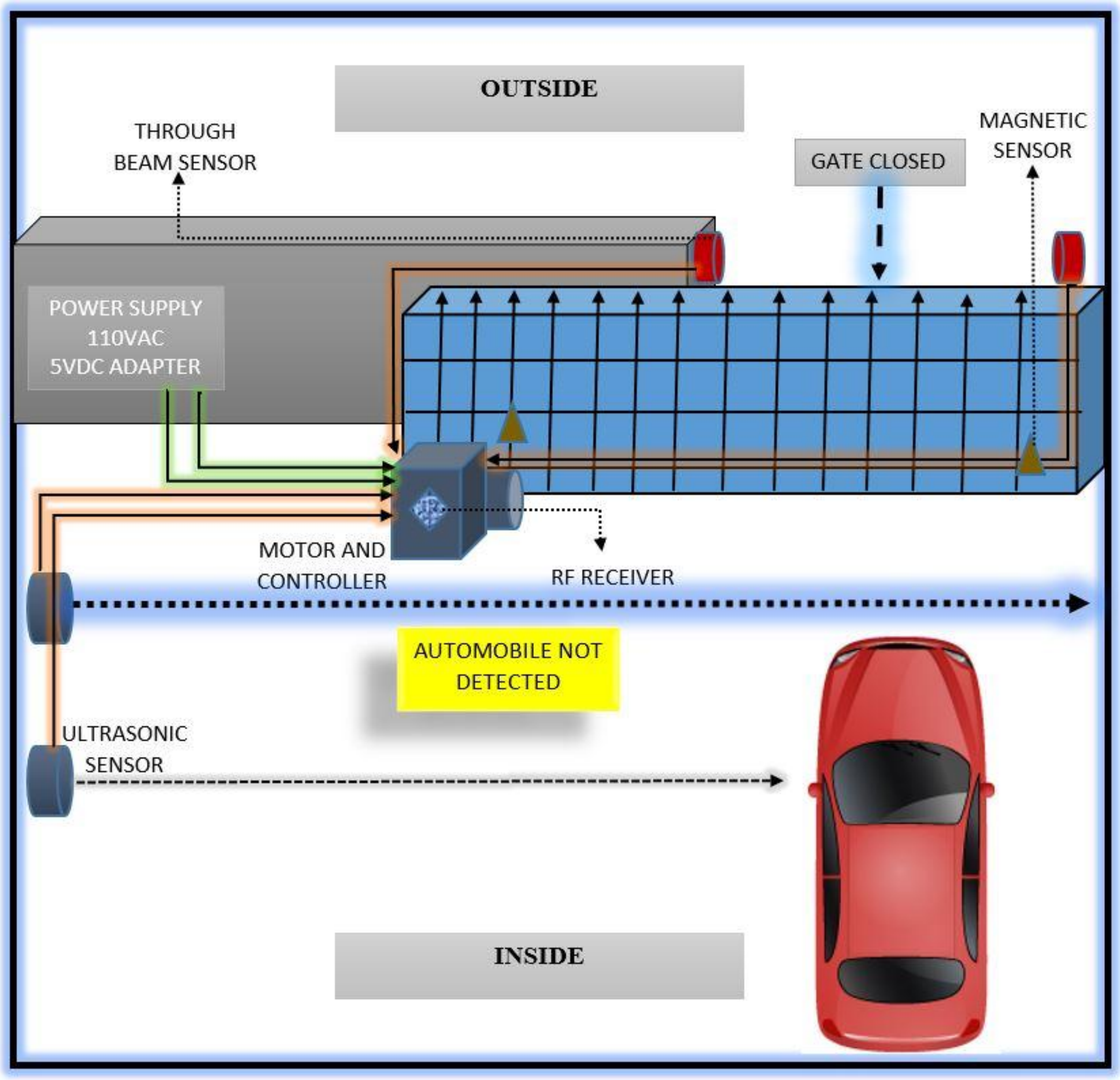


Figure 42 Car from Inside ultrasonic sensor not detected

According to this design, when the car enters into this scenario the gate has to open up itself but as shown in the Figure 42 the gate remains closed. This is due to fact that the car entered is in the range of just one ultrasonic sensor but in order to open the gate the car has to be in the range of both the ultrasonic sensors.

The controller using both the sensors computes the distance as seen in the Fig 43 and from the figure it is clear that the distance from the first ultrasonic sensor is definitely less than the distance it used to sense when it was in idle mode and the distance sensed from the second ultrasonic sensor is the same in idle mode.

Taking these two distance parameters the controller generates a Logic low signal and outputs to the Motor to keep it in close position or to get it back to close position if it is in open position, thus the car is not detected. The main advantage of using two sensors is that there are two points in space to detect the presence of an automobile than just one point, with just one point the system cannot differentiate between the automobile or any arbitrary obstacle which may be a person or any object.

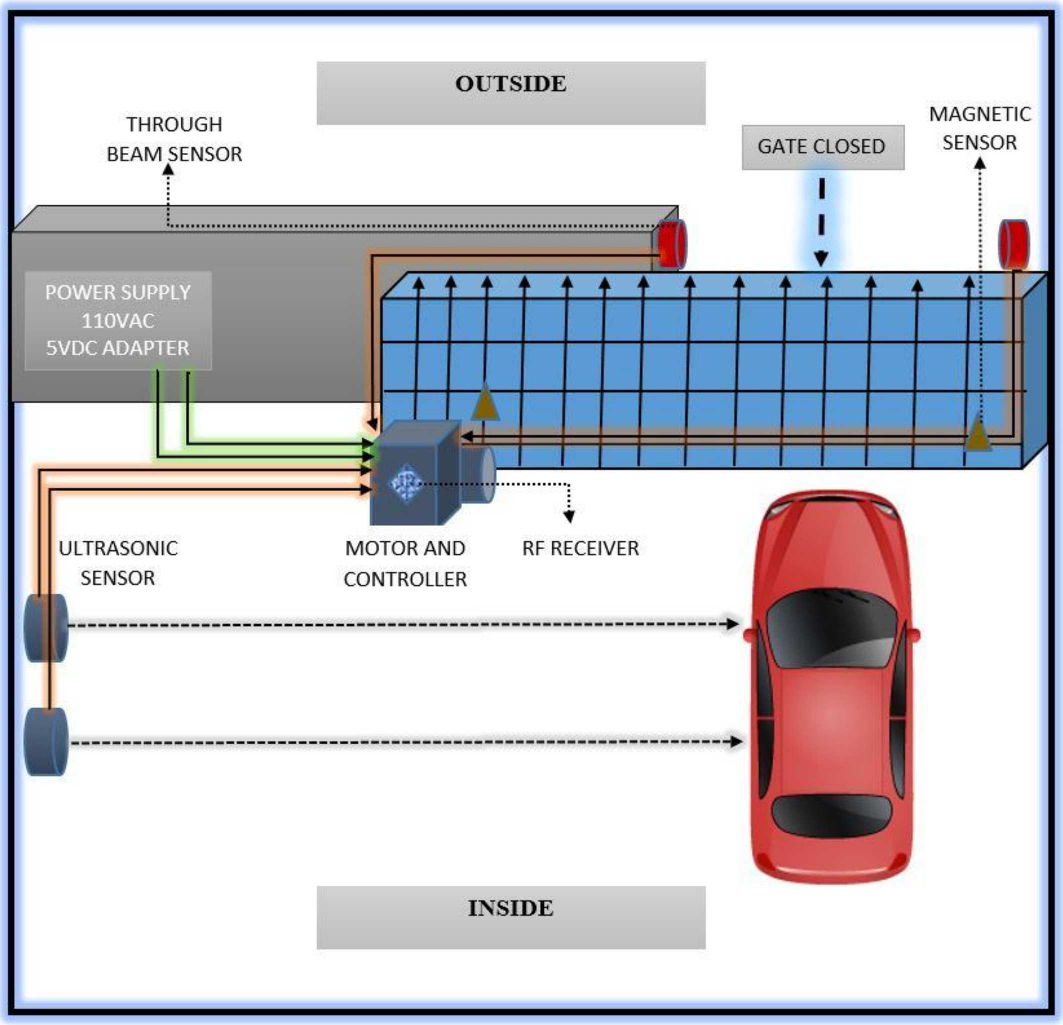


Figure 43 Car detected by Ultrasonic sensors

The above Figure 43 shows the perfect position of the car's placement in order to detect the presence of an automobile by the controller.

The automobile has to be in the range of the ultrasonic sensors. The range does not mean the distance the ultrasonic sensors sense but it is the side face of the automobile which has to cover the distance between both ultrasonic sensors. It does not depend on the specific points on the automobile, all it depends is that both the sensors has to detect a value less than the distance they sense when in idle mode.

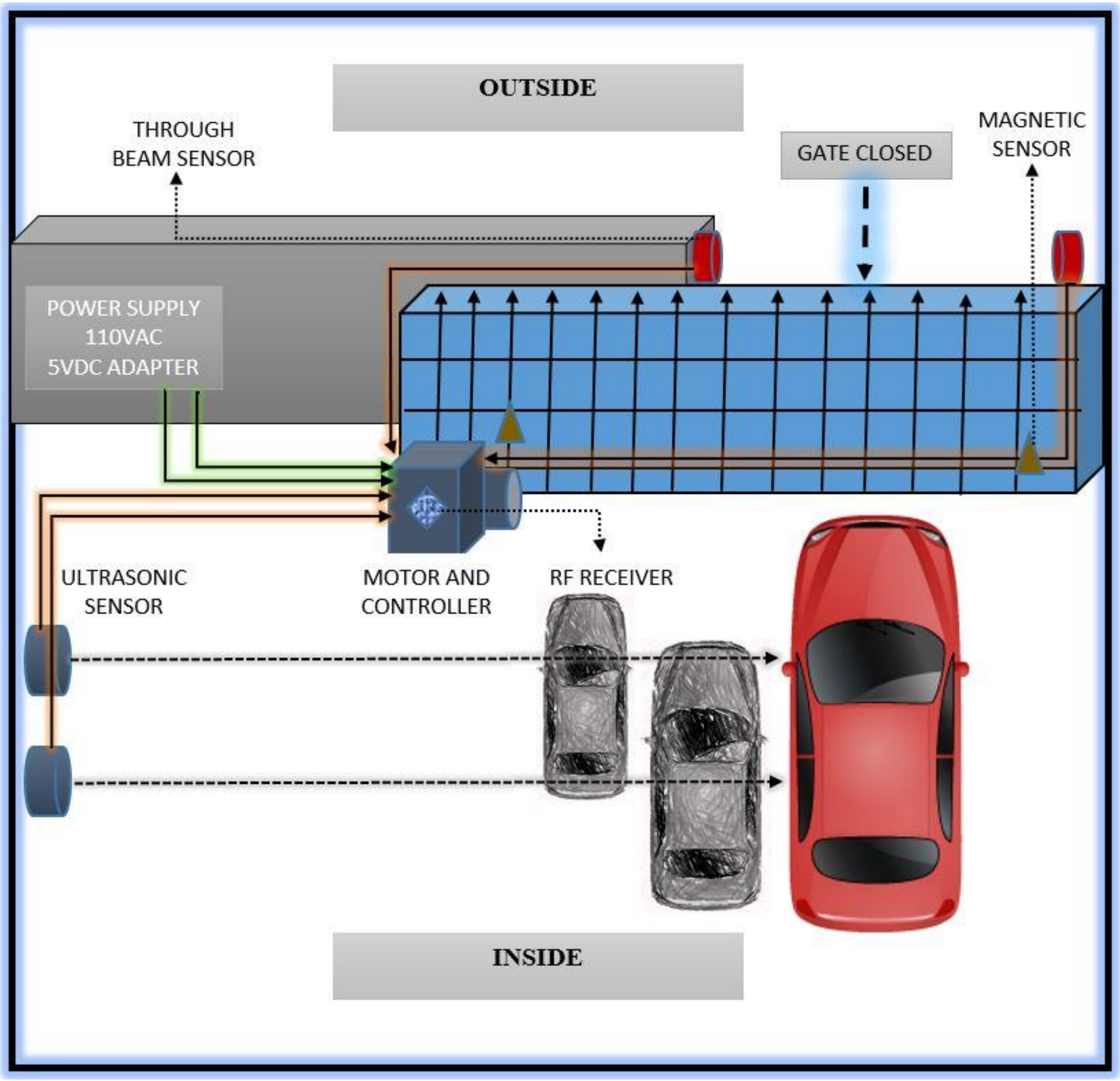


Figure 44 Range of detection

The above Figure 44 shows the detection of the automobiles by the ultrasonic sensors and also the position of the automobiles where the design works.

The Figure 45 shows the opening of the gate after the detection of the automobile.

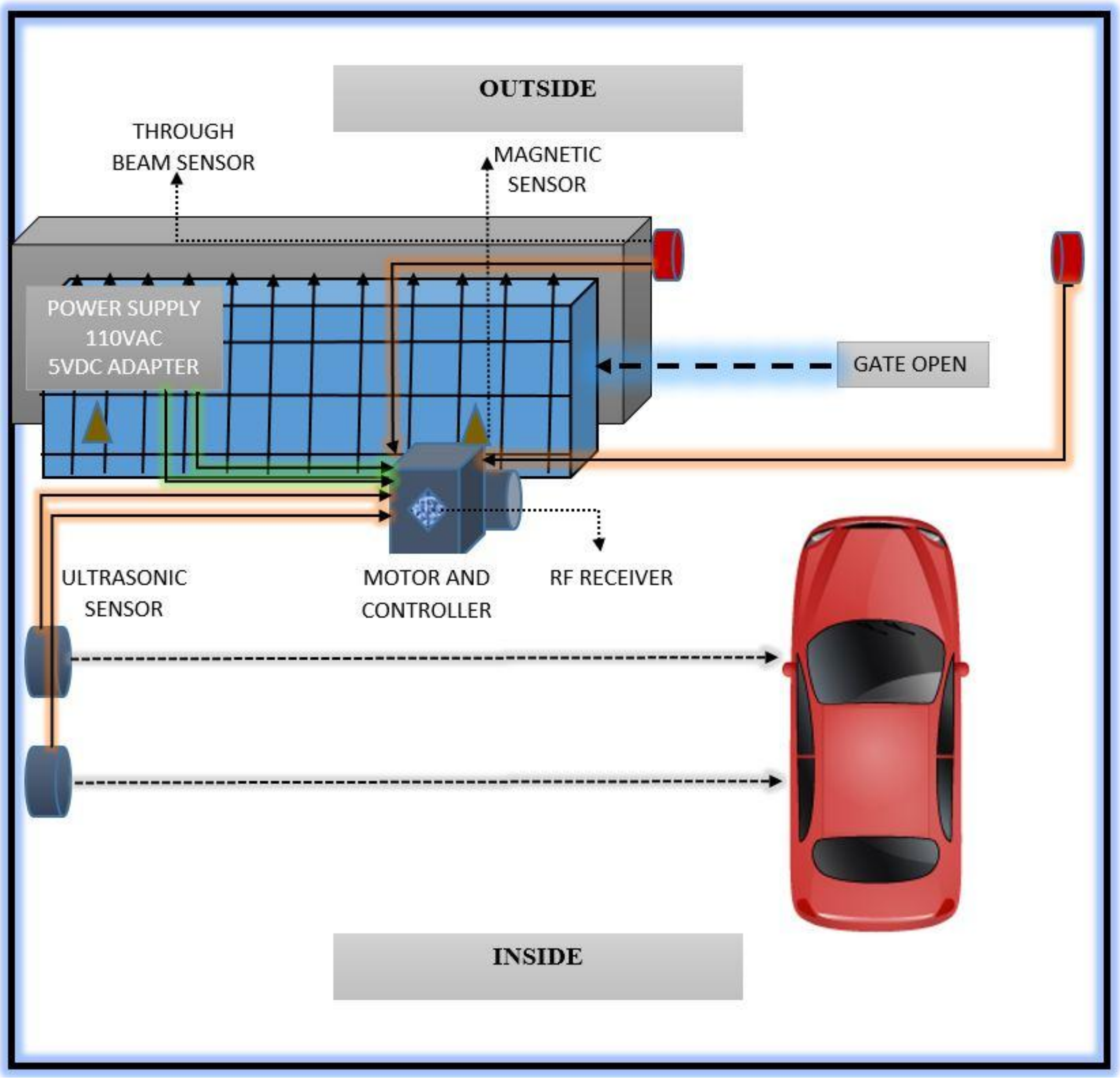


Figure 45 opening of the gate due to the Ultrasonic sensors

When the automobile is in the range of both ultrasonic sensors i.e. the distance computed by the controller is less than the threshold value, the controller generates a Logic High signal and outputs to the Motor through the Motor driver.

The Motor rotates in anti-clockwise direction thus opening the gate. It starts the timer once the gate is in complete open position which is monitored by the magnetic sensor receiver which detects the opening magnet mounted on the right side of the gate.

The gate remains opened for about 35-45 seconds and this can also be modified, then the controller closes the gate by generating a Logic low signal thus conveying the motor to perform its action. The gate controller sends a Logic High signal if it computes distance sensed by two ultrasonic sensors less than the threshold value thus opening the gate in the middle of its operation of closing gate, this means that another automobile is detected.

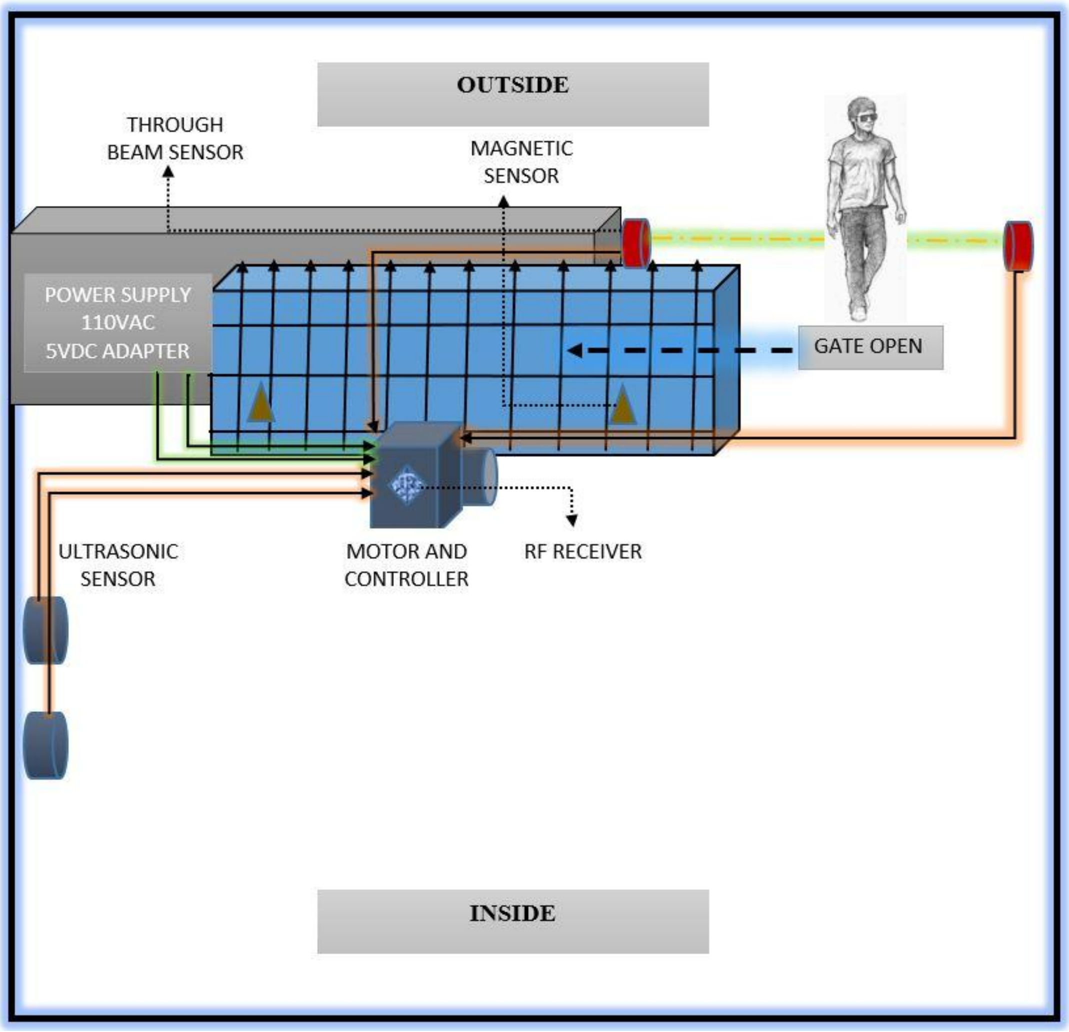


Figure 46 Through beam sensor detection

Then above Figure 46 shows the operation of through beam photoelectric sensor.

When performing a Close operation in any of the scenarios if the through beam sensor detects any presence of obstacles coming in its way then it sends a Logic high signal to the controller.

The controller generates a Logic low signal and outputs it to the motor seizing it rotation and changing its rotation to clockwise direction thus performing an Open gate operation.

The gate remains opened for 35-45 seconds until the controller performs a close operation. The through beam sensor goes to High only when the gate is in making a Close operation.

4.5 Advantages

1. The main advantage of this design is that it is cost effective, the complete controller hardware can be built under $200.
2. The hardware components used are readily available in the market and damage to any parts can be replaced with the similar components based on the availability.
3. In order to make it more sensible for the detection of the automobiles more than two ultrasonic sensors can also be connected to the system.
4. Components used are non-hazardous
5. Serves as one of the most advanced system since it eliminates the usage of any physical devices to open the gate when moving from inside.
6. This design holds flexible for any cheap micro controller such as Raspberry pi, PSoc, 8051 and many more.
7. This design can be used in any type of parking system irrespective of indoor or outdoor
8. All the parameters can be modified depending on the requirement of the user.
9. The position of the ultrasonic sensors can be changed and can be designed in such a way that the gate remains opened by the time it reaches the gate thus saving the 30-40 seconds of waiting period.
10. Installation process is easy and technicians are not required to configure the controller.

CHAPTER 5

SOFTWARE DEVOLOPMENT

Arduino IDE and C language are used to program the Arduino Microcontroller.

5.1 FLOWCHART

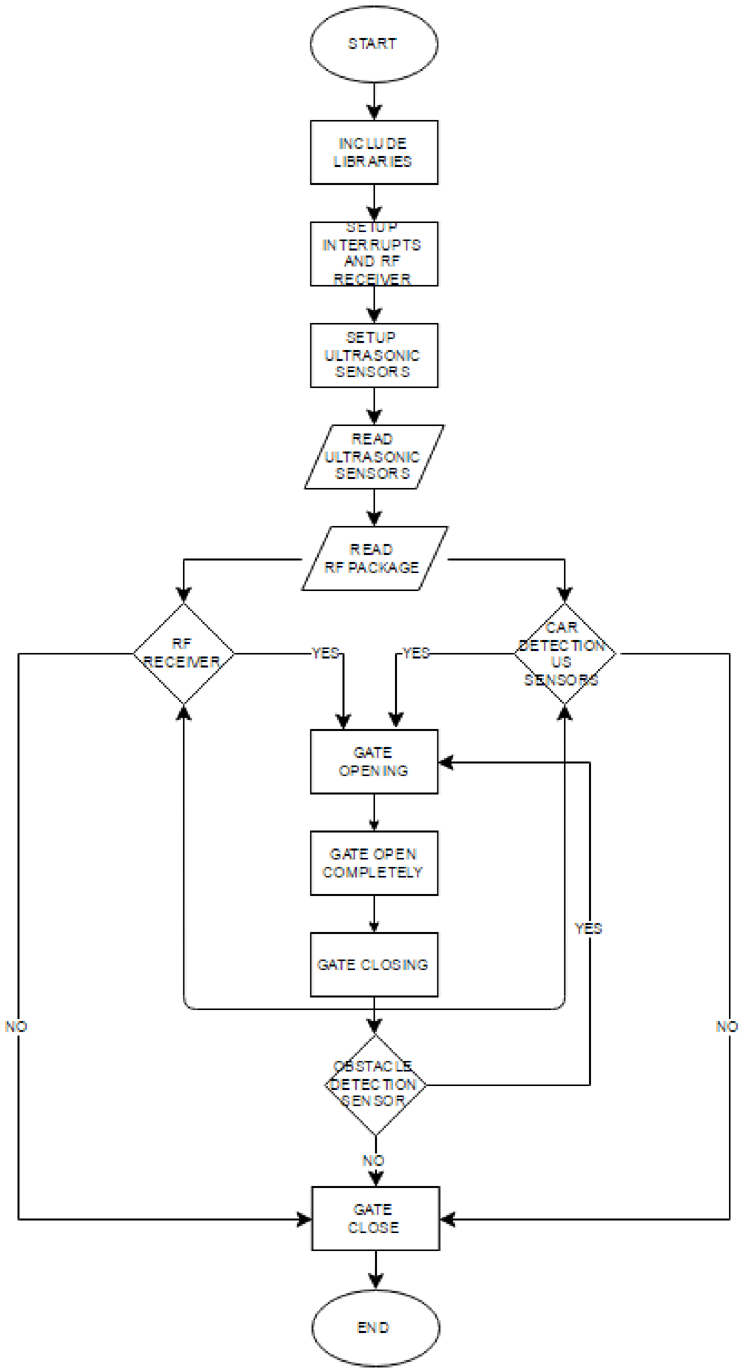


Figure 47 Flowchart of the Design

The figure 47 shows the flowchart of the design used in this project.

5.2 Modules of the Flowchart

5.2.1 Include Libraries

Four libraries are included in the design, which are as follows:

1. SPI.H: This stands for Serial Peripheral Interface, which enables the interfacing of various peripheral devices and components to the Arduino embedded system.
2. WIRE.H: The SDA (data line) and SCL (clock) which are configured at the ports A4 and A5 on the board are used to enable I2C and TWI devices.
3. ADAFRUIT_SSD1306.H: This library holds all the commands required to interface an OLED display to the embedded system.
4. ADAFRUIT_GFX.H: This is secondary library used with the OLED library, which provides various graphics for the display.

5.2.2 Setup Interrupts and RF Receiver

Arduino UNO board has two interrupts, which can be used to perform interrupt function.

Digital Pin 3 is used to setup an Interrupt function. GateOpen_remainOpen_0 is the interrupt call.

Analog Pin A0 is used to setup RF Receiver.

5.2.3 Setup Ultrasonic Sensors and OLED display

.Three Ultrasonic sensors are used in this design and hence they are setup.

Ultrasonic sensor_0 is setup using digital pin 13 and analog pin A1.

Ultrasonic sensor_1 is setup using analog pins A2 and A3.

Ultrasonic sensor_2 is setup using digital pins 8 and 9.

OLED display is also setup.

```
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
```

The above code enables the OLED display with the use of Adafruit library.

5.2.4 Read Ultrasonic sensors and RF Receiver

The system is powered on thus the Arduino board reads the analog values from both the Ultrasonic sensors and the RF receiver.

5.2.5 RF Receiver

The Arduino board continuously reads the values from the RF Receiver. The values read from the receiver are in between 0-250 if the transmitter is not enabled. The values increases to more than 350 if the transmitter is enabled.

This change in values are used to call the interrupt function gateOpen_remainOpen_0.

```
void interrupt_function_1()
{
  if(analogRead(receiver_pin_1)>350)
  {
    digitalWrite(interrupt_pin_1, HIGH);
  }
  else
  {
    digitalWrite(interrupt_pin_1, LOW);
  }
}
```

Above code is used to set the interrupt high if the condition is satisfied.

5.2.6 Car detection Ultrasonic sensors

If the values from the Ultrasonic sensors 0 and 1 meet the condition then the interrupt function gateOpen_remainOpen_0 is called by making interrupt pin high.

```
void interrupt_function_0()
{
  if(distance_0 < 15 && distance_1 < 15)
  {
    OLED_vehicle_detected();
    display.clearDisplay();
    digitalWrite(interrupt_pin_0, HIGH);
  }
  else
  {
    digitalWrite(interrupt_pin_0, LOW);
  }
}
```

The above code sets the interrupt pin high thus calling the interrupt function. Here for the experimental purposes the value used is 15 indicating the distance between the ultrasonic sensors and the opposite wall.

5.2.7 Gate Open, Gate Open Completely and Gate Closing

If any of the RF receiver or the Car detection, ultrasonic sensors become high then all these three functions are called.

5.2.8 Obstacle detection sensor

The ultrasonic sensor_2 is enabled when the gate closing function is enabled. It becomes high if it reads a value less than 18(used here for experimental purposes).

```
if(distance_2 < 18)
{
 for(m=k;m>=0;m--)
{
 interrupts();
digitalWrite(led_pin, HIGH);
delay(900);
digitalWrite(led_pin, LOW);
delay(100);
Serial.print("GATE IS OPENING--->");
```

If distance is less than 18 then Gate Opening function is called.

5.2.9 Gate Close

If RF Receiver, Car detection Ultrasonic sensors and Obstacle detection sensors read a LOW signal then Gate Close function is performed.

5.3 Code

```
#include <SPI.h>                          //Library Functions
#include <Wire.h>                         //Library Functions
#include <Adafruit_GFX.h>                 //Library Functions
#include <Adafruit_SSD1306.h>             //Library Functions


#define OLED_RESET 4                      //OLED is reset
Adafruit_SSD1306 display(OLED_RESET);

int i, j, k, m, n, p, value;

int trig_pin_0 = 13;                      //ULTRASONIC_SENSOR_0
int echo_pin_0 = A1;

int trig_pin_1 = A2;                      //ULTRASONIC_SENSOR_1
int echo_pin_1 = A3;

int trig_pin_2 = 8;                       //ULTRASONIC_SENSOR_2
int echo_pin_2 = 9;

int duration_0, duration_1, duration_2;
int distance_0, distance_1, distance_2;

//---------------------------------------------------


int interrupt_pin_0 = 3;                  //Interrupt setup
int interrupt_pin_1 = 3;
int receiver_pin_0 = A5;                  //RF Receiver setup
int receiver_pin_1 = A0;
int led_pin = 5;                          //RF receiver
                                          //indicatior

//--------------------------------------------------
```

```
void setup()
{
Serial.begin(9600);                         //Serial monitor
                                            //enabled
pinMode(trig_pin_0, OUTPUT);                //ULTRASONIC_SENSORS
pinMode(echo_pin_0, INPUT);                 //setup

pinMode(trig_pin_1, OUTPUT);
pinMode(echo_pin_1, INPUT);

pinMode(trig_pin_2, OUTPUT);
pinMode(echo_pin_2, INPUT);
pinMode(led_pin, OUTPUT);

pinMode(interrupt_pin_0, OUTPUT);           //RF RECEIVER
pinMode(interrupt_pin_1, OUTPUT);
//Interrupt setup
attachInterrupt(digitalPinToInterrupt(interrupt_pin_0),gateOpen_remainOpen_0,
RISING);
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
}
void ultrasonic_sensor_0()
{
digitalWrite(trig_pin_0, HIGH);             //Read Ultrasonic sensor_0
delayMicroseconds(10);
digitalWrite(trig_pin_0, LOW);
duration_0 = pulseIn(echo_pin_0, HIGH);
distance_0 = (duration_0/2)/29.1;
Serial.print("1st_Sensor: ");
Serial.print(distance_0);
Serial.print("cm    ");
}
```

```
void ultrasonic_sensor_1()
{
digitalWrite(trig_pin_1, HIGH);                    //Read Ultrasonic sensor_1
delayMicroseconds(10);
digitalWrite(trig_pin_1, LOW);
duration_1 = pulseIn(echo_pin_1, HIGH);
distance_1 = (duration_1/2)/29.1;
Serial.print("2nd_Sensor: ");
Serial.print(distance_1);
Serial.print("cm    ");
}

void ultrasonic_sensor_2()
{
digitalWrite(trig_pin_2, HIGH);                    //Read Ultrasonic senor_2
delayMicroseconds(10);
digitalWrite(trig_pin_2, LOW);
duration_2 = pulseIn(echo_pin_2, HIGH);
distance_2 = (duration_2/2)/29.1;
Serial.print("3nd_Sensor: ");
Serial.print(distance_2);
Serial.print("cm    ");
}

void RF_receiver()
{
//Serial.print("Analog Value 1 = ");               //Read RF Receiver
//Serial.print(analogRead(receiver_pin_0));
//Serial.print("--");
Serial.print("Analog value 1 = ");
Serial.println(analogRead(receiver_pin_1));
}
```

```
void interrupt_function_0()
{                                              //Interrupt function condition
if(distance_0 < 15 && distance_1 < 15)         //Car detection Ultrasonic
{                                              sensors
OLED_vehicle_detected();
display.clearDisplay();
digitalWrite(interrupt_pin_0, HIGH);
}
else
{
digitalWrite(interrupt_pin_0, LOW);
}
}

void interrupt_function_1()
{                                              //Interrupt function condition
if(analogRead(receiver_pin_1)>350)             //RF Receiver
{
digitalWrite(interrupt_pin_1, HIGH);
}
else
{
digitalWrite(interrupt_pin_1, LOW);
}
}

void OLED_welcome()
{
display.clearDisplay();                        //OLED display initial
display.setTextColor(WHITE);                   message setup
display.setTextSize(1);
display.setCursor(13,5);
```

```
display.println("  PARKING GATE");
display.setCursor(8,16);
display.print("CONTROLLER SYSTEM");
display.display();
}

void OLED_vehicle_detected()
{
display.clearDisplay();                    //OLED display message for
display.setTextColor(WHITE);               vehicle detection
display.setTextSize(1);
display.setCursor(15,10);
display.print("VEHICLE DETECTED");
display.display();
}

void OLED_gate_opening()
{                                          //OLED display message for
display.clearDisplay();                    gate opening
display.setTextColor(WHITE);
display.setTextSize(1);
display.setCursor(0,10);
display.print("GATE IS OPENING-->");
display.println(i);
display.display();
}

void OLED_gate_opened_completely()
{
display.clearDisplay();                    //OLED display message for
display.setTextColor(WHITE);               gate opened position
display.setTextSize(1);
display.setCursor(0,5);
```

```
display.println("GATE IN");
display.print("OPEN POSITION-->");
display.println(j);
display.display();
}

void OLED_gate_closing()
{
display.clearDisplay();                    //OLED display message for
display.setTextColor(WHITE);               gate closing
display.setTextSize(1);
display.setCursor(0,5);
display.print("GATE IS CLOSING-->");
display.println(k);
display.display();
}

void OLED_obstacle_detected()
{                                          //OLED display message for
display.clearDisplay();                    obstacle detection
display.setTextColor(WHITE);
display.setTextSize(1);
display.setCursor(0,5);
display.print("OBSTACLE DETECTED-->");
display.println(m);
display.display();
}

void OLED_remote_pressed()
{                                          //OLED display message if
display.clearDisplay();                    RF remote is pressed
display.setTextColor(WHITE);
display.setTextSize(1);
```

```
display.setCursor(0,5);
display.print("REMOTE PRESSED-->");
display.println(p);
display.display();
}

void loop()                                        //OLED display
{                                                  //clear display
OLED_welcome();
display.clearDisplay();
ultrasonic_sensor_0();
ultrasonic_sensor_1();
RF_receiver();
interrupt_function_0();
interrupt_function_1();
delay(1000);
}

void gate_opening()
{
for(i=0;i<=10;i++)
{
interrupts();                                      //Gate opening function
digitalWrite(led_pin, HIGH);                       //declarion
delay(900);                                        //10 seconds
digitalWrite(led_pin, LOW);
delay(100);
Serial.print("GATE IS OPENING-->");
Serial.println(i);
OLED_gate_opening();
}
display.clearDisplay();
}
```

```
void gate_opened_completely()
{
for(j=0;j<=10;j++)
{                                          //Gate opening position
digitalWrite(led_pin, HIGH);               //Declarion
delay(900);                                //waits in the open position for
digitalWrite(led_pin, LOW);                //10seconds
delay(10);
Serial.print("GATE IN OPEN POSITION--->");
Serial.println(j);
OLED_gate_opened_completely();
}
display.clearDisplay();
}

void gateOpen_remainOpen_0()
{                                          //interrupt call function
gate_opening();                            //performs 3 operations
gate_opened_completely();                  //if interrupt is called
gate_closing();                            //gate opening
}                                          //gate open completely
void gateOpen_remainOpen_1()               //gate closing
{
gate_opening();
gate_opened_completely();
gate_closing();
}

void gate_closing()
{                                          //gate closing function
for(k=0; k<=10; k++)
{
interrupts();
```

```
digitalWrite(led_pin, HIGH);
delay(100);
digitalWrite(led_pin, LOW);
delay(900);
Serial.print("GATE IS CLOSING--->");
Serial.println(k);
ultrasonic_sensor_0();
ultrasonic_sensor_1();
ultrasonic_sensor_2();
RF_receiver();
OLED_gate_closing();
if(distance_2 < 18)                          //ultrasonic sensor detects
{                                            //obstacles when closing
for(m=k;m>=0;m--)                            //performs 2 operations
{
interrupts();
digitalWrite(led_pin, HIGH);
delay(900);
digitalWrite(led_pin, LOW);
delay(100);
Serial.print("GATE IS OPENING--->");         //gate opening
Serial.println(m);
OLED_obstacle_detected();
delay(250);
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(1);
display.setCursor(0,10);
display.print("GATE IS OPENING-->");         //gate opening completely
display.println(m);
display.display();
}
display.clearDisplay();
```

```
gate_opened_completely();
k=0;
}
else if (distance_0 < 15 && distance_1 < 15)          //if car detection sensors
{                                                      //sense a car
for(n=k;n>=0;n--)                                      //performs 2 operations
{
interrupts();
digitalWrite(led_pin, HIGH);
delay(900);
digitalWrite(led_pin, LOW);
delay(100);
Serial.print("GATE IS OPENING--->");                   //Gate oepnning
Serial.println(n);
OLED_vehicle_detected();
delay(250);
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(1);
display.setCursor(0,10);
display.print("GATE IS OPENING-->");                   //Gate opened completely
display.println(n);
display.display();
}
display.clearDisplay();
gate_opened_completely();
k=0;
}
else if(analogRead(receiver_pin_1)>350)                //RF receiver
{                                                      //if value more than 350
for(p=k;p>=0;p--)                                      Performs 2 operations
{
interrupts();
```

```
digitalWrite(led_pin, HIGH);
delay(900);
digitalWrite(led_pin, LOW);
delay(100);
Serial.print("GATE IS OPENING--->");                    //gate opening
Serial.println(p);
OLED_remote_pressed();
delay(250);
display.clearDisplay();
display.setTextColor(WHITE);
display.setTextSize(1);
display.setCursor(0,10);
display.print("GATE IS OPENING-->");
display.println(p);
display.display();
}
display.clearDisplay();                                 //OLED display is cleared
gate_opened_completely();
k=0;
}
else
{

}
}
}
```

CHAPTER 6

CONCLUSION

With the advancement in technology many new methods to open, designs, procedures, smart devices and many more are used in Parking gate controller system.

This project starts with surveying various parking gate controllers that are available in the market.

This design is uncomplicated and simple to use compared to other technologies. Some of the components used in this design are ultrasonic sensors which are readily available in the market. There are various kinds of ultrasonic sensors available based on the price, range detection and any of them can be used to detect the automobile.

Comparison is made between various sensors which are used in this project, a brief description is provided with their pin diagrams, connections, price range, availability in market and specifications.

The design can be modified according to the user's application and itworks for any type of parking gate controller, which could be a barrier lift arm, swing gates or even open house doors.

The controller used is Arduino because of its robust design and its low cost compared to other microcontrollers. It's operation is stable for high end applications and serves the best for all small scale designs.

Photoelectric sensors provide a platform for the detection of any obstacles if they are in its area of detection, they serve a very good proximity detection without even knowing their existence.

Thus the whole project is cost effective, easy to install, hardware and software implementation are also presented.

REFERENCES

1. Alan S Morris, *Measurement and Instrumentation principles*, March 9, 2001.
2. Jeremy Blum, Exploring *Arduino: Tools and Techniques for Engineering*, July 16, 2013
3. Krzysztof Iniewski, *Smart Sensors and Industrial Applications*, April 19, 2016
4. Dale Ensminger; Leonard J. Bond, *Ultrasonic Fundamentals, Technologies and Applications*, 2012.
5. Michael J Pont, *Embedded C*, published by Addison-Wesley, May 4, 2002
6. Gary Rockis; Glen A. Mazur, *Electrical Motor controls for integrated Systems Workbook*, published by American Technical publishers, 2013

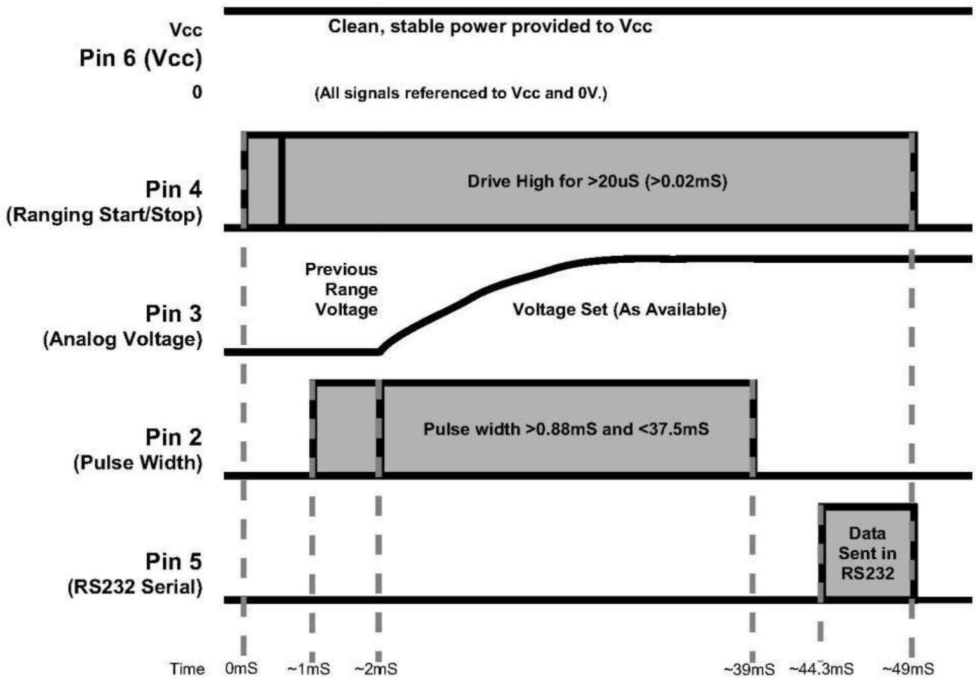
APPENDIX A

General Power-Up Instruction

Each time the LV-MaxSonar-EZ is powered up, it will calibrate during its first read cycle. The sensor uses this stored information to range a close object. It is important that objects not be close to the sensor during this calibration cycle. The best sensitivity is obtained when the detection area is clear for fourteen inches, but good results are common when clear for at least seven inches. If an object is too close during the calibration cycle, the sensor may ignore objects at that distance.

The LV-MaxSonar-EZ does not use the calibration data to temperature compensate for range, but instead to compensate for the sensor ringdown pattern. If the temperature, humidity, or applied voltage changes during operation, the sensor may require recalibration to reacquire the ringdown pattern. Unless recalibrated, if the temperature increases, the sensor is more likely to have false close readings.

Timing Diagram



Timing Description

250mS after power-up, the LV-MaxSonar-EZ is ready to accept the RX command. If the RX pin is left open or held high, the sensor will first run a calibration cycle (49mS), and then it will take a range reading (49mS). After the power up delay, the first reading will take an additional ~100mS. Subsequent readings will take 49mS. The LV-MaxSonar-EZ checks the RX pin at the end of every cycle. Range data can be acquired once every 49mS.

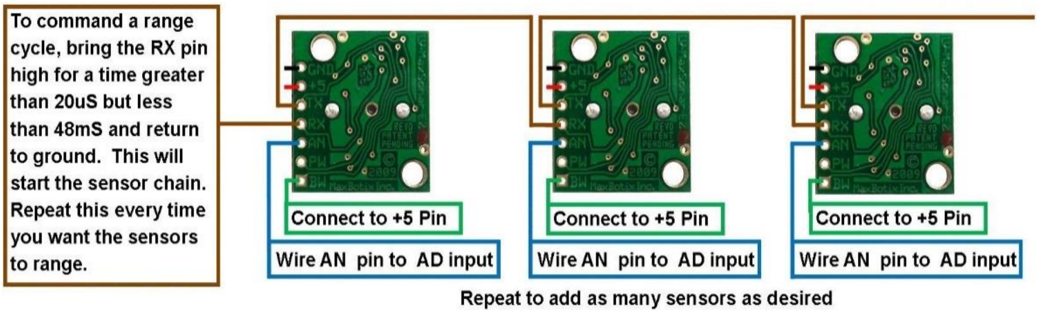
Each 49mS period starts by the RX being high or open, after which the LV-MaxSonar-EZ sends the transmit burst, after which the pulse width pin (PW) is set high. When a target is detected the PW pin is pulled low. The PW pin is high for up to 37.5mS if no target is detected. The remainder of the 49mS time (less 4.7mS) is spent adjusting the analog voltage to the correct level. When a long distance is measured immediately after a short distance reading, the analog voltage may not reach the exact level within one read cycle. During the last 4.7mS, the serial data is sent.

The LV-MaxSonar-EZ timing is factory calibrated to one percent at five volts, and in use is better than two percent. In addition, operation at 3.3V typically causes the objects range, to be reported, one to two percent further than actual.

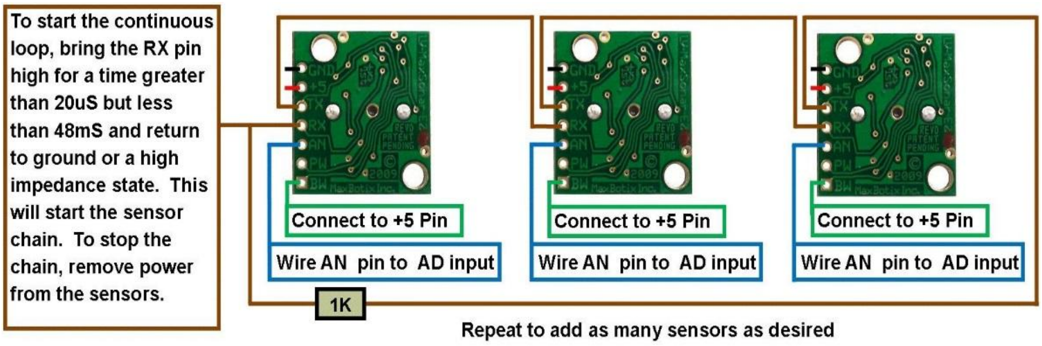
Using Multiple Sensors in a single system

When using multiple ultrasonic sensors in a single system, there can be interference (cross-talk) from the other sensors. MaxBotix Inc., has engineered and supplied a solution to this problem for the LV-MaxSonar-EZ sensors. The solution is referred to as chaining. We have 3 methods of chaining that work well to avoid the issue of cross-talk.

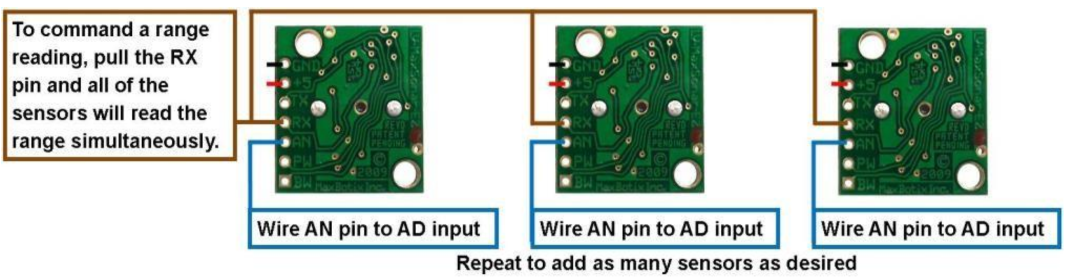
The first method is AN Output Commanded Loop. The first sensor will range, then trigger the next sensor to range and so on for all the sensor in the array. Once the last sensor has ranged, the array stops until the first sensor is triggered to range again. Below is a diagram on how to set this up.



The next method is AN Output Constantly Looping. The first sensor will range, then trigger the next sensor to range and so on for all the sensor in the array. Once the last sensor has ranged, it will trigger the first sensor in the array to range again and will continue this loop indefinitely. Below is a diagram on how to set this up.



The final method is AN Output Simultaneous Operation. This method does not work in all applications and is sensitive to how the other sensors in the array are positioned in comparison to each other. Testing is recommend to verify this method will work for your application. All the sensors RX pins are conned together and triggered at the same time causing all the sensor to take a range reading at the same time. Once the range reading is complete, the sensors stop ranging until triggered next time. Below is a diagram on how to set this up.



Independent Sensor Operation

The LV-MaxSonar-EZ sensors have the capability to operate independently when the user desires. When using the

LV-MaxSonar-EZ sensors in single or independent sensor operation, it is easiest to allow the sensor to free-run. Free-run is the default mode of operation for all of the MaxBotix Inc., sensors. The LV-MaxSonar-EZ sensors have three separate outputs that update the range data simultaneously: Analog Voltage, Pulse Width, and RS232 Serial. Below are diagrams on how to connect the sensor for each of the three outputs when operating in a single or independent sensor operating environment.

Selecting an LV-MaxSonar-EZ

Different applications require different sensors. The LV-MaxSonar-EZ product line offers varied sensitivity to allow you to select the best sensor to meet your needs.

Each LV-MaxSonar-EZ sensor has a calibrated beam pattern. Each sensor is matched to provide the approximate detection pattern shown in this datasheet. This allows end users to select the part number that matches their given sensing application. Each part number has a consistent field of detection so additional units of the same part number will have similar beam patterns. The beam plots are provided to help identify an estimated detection zone for an application based on the acoustic properties of a target versus the plotted beam patterns.

Each beam pattern is a 2D representation of the detection area of the sensor. The beam pattern is actually shaped like a 3D cone (having the same detection pattern both vertically and horizontally). Detection patterns for dowels are used to show the beam pattern of each sensor. Dowels are long cylindered targets of a given diameter. The dowels provide consistent target detection characteristics for a given size target which allows easy comparison of one MaxSonar sensor to another MaxSonar sensor.

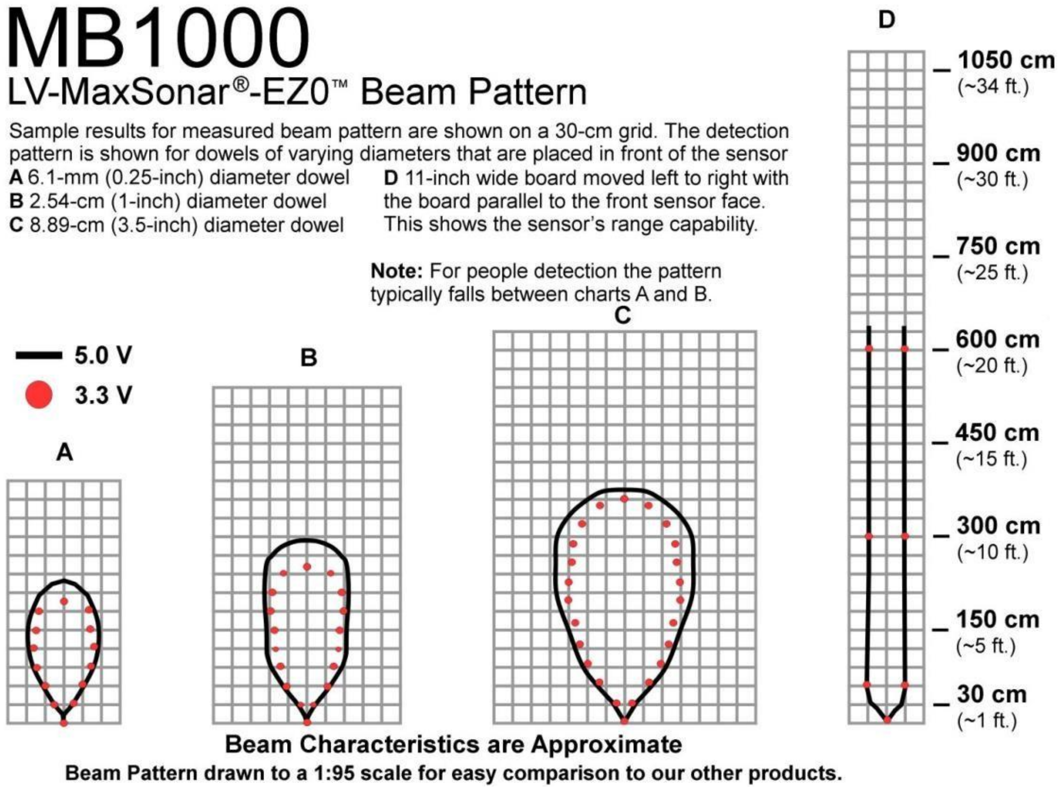
MB1000 LV-MaxSonar-EZ0

The wide beam makes this sensor ideal for a variety of applications including people detection, autonomous navigation, and wide beam applications.

MB1000 Features and

Benefits

1. Widest and most sensitive beam pattern in LV-MaxSonar-EZ line
2. Low power consumption
3. Easy to use interface
4. Will pick up the most noise clutter when compared to other sensors in the LV-MaxSonar-EZ line
5. Detects smaller objects
6. Best sensor to detect soft object.
7. LV-MaxSonar-EZ line.
8. Requires use of less sensors to cover a given area.
9. Can be powered by many different types of power sources.
10. Can detect people up to approximately 10 feet.



I2CXL-MaxSonar® - EZ™ Series

The I2CXL-MaxSonar-EZ series is the first MaxSonar ultrasonic sensor to feature the I2C interface. The sensors have high acoustic power output along with real-time auto calibration for changing conditions (voltage and acoustic or electrical noise) that ensure users receive the most reliable (in air) ranging data for every reading taken. The I2CXL-MaxSonar-EZ low power 3V – 5.5V operation provides very short to long-range detection and ranging, in a tiny and compact form factor. The I2CXL-MaxSonar-EZ detect objects from 0-cm* to 765-cm (25.1 feet) and provide sonar range information from 20-cm out to 765-cm with 1- cm resolution.

Features

1. I2C bus communication allows rapid control of multiple sensors with only two wires.
2. High acoustic power output.
3. Real-time auto calibration and noise rejection for every ranging cycle Calibrated beam patterns
4. Continuously variable gain
5. Object detection as close as 1-mm from the sensor
6. 3V to 5.5V supply with very low average current draw
7. Readings can occur up to every
8. 25mS (40Hz rate)[3] for up-close objects. 15Hz rate for full range. Triggered operation provides a new range reading as desired.
9. Ultrasonic signal frequency of 42KHz.
10. Status pin available to determine sensor state.
11. Power-up address reset pin available.
12. Physical dimensions match other.

13. XL-MaxSonar-EZ products.
14. -40°C to +65°C operation (+85°C limited operation)

Applications

1. Educational and hobby robotics Environments with acoustic and electrical noise
2. Distance measuring
3. Long range object detection
4. Security systems Motion detection Landing flying objects Collision avoidance Bin level measurement
5. This product is not recommended as a device for personal safety

I2CXL-MaxSonar®-EZ™ Pin Out

Pin 1- Add-Reset: This pin is internally pulled high by the sensor. On power up, the state of this pin is checked; if left high or disconnected, the sensor will use the address stored in EEPROM for I2C communications. If pulled low (to ground), the sensor will use its default shipped address for I2C communications for the current power cycle only (to permanently change the address, one must use the I2C Change Address command).

Pin 2- Status- This pin provides a status indicator for the current state of the sensor.
On power up, it will provide a pulse width representation of the address that will be loaded from EEPROM and used, with a length of 100 microseconds × the 8-bit read/write version of address (~22.4 milliseconds for the default address). During real-time operation, the line will be driven low while I2C bus communications are active and the sensor is listening for a command. Pin 2 will transition within 50 microseconds of receiving a ranging command to high, and will remain high until after the sensor is ready to communicate via I2C again after a range cycle. This is provided so users may poll the line to determine if the sensor has finished its ranging cycle, or set up an interrupt to be generated when the sensor is ready to report the latest range reading.

Pin 3-Not Used- This pin is not used for the I2CXL-MaxSonar-EZ series.

Pin 4-SDA (I2C Data)– This is the data line for I2C communications. The I2CXL-MaxSonar-EZ series operate as I2C slave devices, supporting 7-bit addressing, with a default shipped address of hexadecimal 70 (decimal 112). This corresponds to an equivalent 8-bit address of E0 (decimal 224).

Pin 5-SCL (I2C Clock)- This is the clock line for I2C communications. The I2CXL-MaxSonar-EZ series supports I2C clock frequencies up to 100kHz. Clock stretching must be supported by the master device for read requests with a clock speed above 25kHz.
V+- Vcc – Operates on 3V - 5.5V. The average current draw for 3.3V operation is 2.7mA (50mA peak) and for 5V operation is 4.4mA (100mA peak) respectively. Peak current is used during sonar pulse transmit.

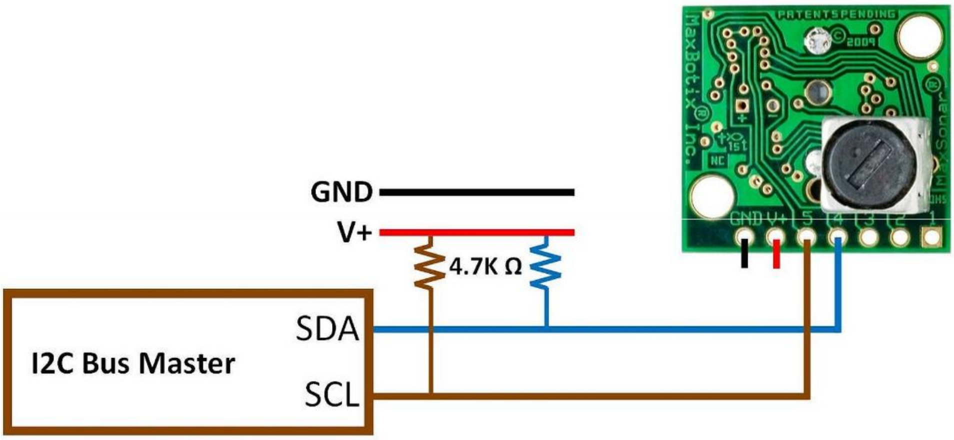
Single Sensor Reference Wiring Diagram

The I2C bus is a bi-directional, two wire interface that consists of a clock line and a data line.

The clock line and the data line both require a pull-up resistor attached to positive supply voltage. The recommended resistance is 4.7 kΩ; however, the value used is dependent upon the user configuration, with more sensors possibly requiring a lower resistance and users that use only one to two sensors per bus could use a larger resistance for the pull-ups. The I2CXL-MaxSonar-EZ series is capable of sinking more current through thepull-up resistors than the I2C specification allows (15mA versus 3mA), so that users may use a much lower resistance if it is required for their application.

Only one pull-up resistor is required each for the SCL and SDA lines per bus– not per sensor.

The pull-up voltage applied to the I2C clock line and data line should not be more than approximately 0.7 volts above what is applied to V+.



I2CXL-MaxSonar®-EZ™ General Operation

After going through the power-up cycle, the I2CXL-MaxSonar-EZ sensor(s) will remain in an idle state, listening to I2C bus communications. The sensor will always operate as an I2C slave; it is never a bus master. The sensor will wait for a read or write message from a bus master which contains an address that matches the sensors current address (determined during power-up).

By default, the sensor has a shipped 7-bit address of 0x70 (decimal 112). This translates to an equivalent 8-bit write address of hexadecimal E0 (decimal 224), and 8-bit read address of hexadecimal E1 (decimal 225). An explanation is given in the figure below. The address may be changed to any other 7-bit value, except 0x00 (decimal 0). Therefore, a single I2C bus could support up to 127 simultaneous I2CXL-MaxSonar-EZ sensors operating on it, as long as the user configuration allows I2C communications work properly.

The sensor may only be reliably addressed at clock speeds up to 100kHz. However, the bus itself may run faster than this as long as the I2CXL-MaxSonar-EZ sensor is not being addressed (tested up to 400kHz).

Sampling faster than 10Hz in certain environments that reflect acoustic noise well could cause the sensor to pick up signals from previous range readings and report false data. However, in certain environments it is possible to take range readings at a significantly faster rate (up to 40Hz depending on the application, see the “Take Range Reading” command page for more details).

The diagram above is an I2C addressing explanation: Each address sent over the I2C bus consists of 7 bits. Because each message over the bus is 8 bits long, the final bit when addressing indicates whether the master wishes to write to or read from the addressed slave. As such, the 7-bit version of the address is effectively the 8-bit version divided by 2 and includes only the first seven bits, whereas the 8-bit read and 8-bit write addresses include the final read/write bit in the address.

Reading from the sensor

After an address match with a read request, the sensor will return the high byte, followed by the low byte (to be interpreted as a 16-bit word), of the latest range reading in centimeters, followed by hexadecimal FF (decimal 255) for each subsequent byte requested. If the sensor receives a range request after power up without taking a range reading (i.e. receiving the “Take Range Reading” command), the sensor will respond with two bytes containing information used for in-factory manufacturing purposes. Therefore, a bus master simply needs to request two bytes back from the sensor after a range cycle to determine the range to a target. The diagram below is an I2C command diagram for reading back two bytes from the sensor after taking a range reading. The stop is optional and signifies only that the bus master does not wish to communicate further.

Start	ADDRESS	R	ACK	High byte	ACK	Low byte	NACK	Stop

Bus Master	Sensor	Bus Master	Sensor	Bus Master

The diagram below shows the SCL and SDA signals for the I2C read request operation. Notice that the sensor will clock stretch after every ACK, while preparing to respond with the latest byte of data. The sensor will release the clock line as soon as the sensor is ready to respond with the latest byte. This clock stretching must be supported if operating over approximately 25kHz clock rate (the clock stretching becomes trivial compared to the time between rising edges).

After an address match with a write request, all data written to the sensor over the bus is compared to the command set that the I2CXL-MaxSonar-EZ series supports; a summary is included above. A match of the received data by the sensor with a command will cause the specific action to be performed.

After receiving a byte from a bus master with a value of the ranging command and responding with the ACK, the sensor will stop responding to I2C communications sent to it. Within 50 microseconds of the ACK bit being generated, the sensor will drive pin 2 high. The sensor will transmit the ultrasonic pulse that will be used to measure distance, and proceed to measure the time of flight to a detectable target. . After detecting a target and determining the range (from approximately 15 milliseconds up to approximately 60 milliseconds after raising pin 2 high), the sensor will resume I2C communications and return to an idle state while listening for incoming messages. Pin 2 will then be brought low to indicate the sensor is ready to communicate via I2C again.

If the sensor is addressed while in the middle of a range reading, all requests for communication will be responded with a NACK. If the sensor receives a read request while ranging, it will respond with NACK for the initial request, and all data bytes returned will be equal to hexadecimal FF.

The figure below shows the I2C command diagram of how to command the sensor to take a range reading. After addressing the sensor with a write request, the “Take Range Reading” command value (0x51, decimal 81) must be written to the sensor. After acknowledging receipt of the command, the sensor will take a range reading.

Through power cycles, the I2CXL-MaxSonar-EZ sensor stores a permanent address in internal EEPROM, which will be used on power up for I2C communications if Pin 1 (Address Reset) is left disconnected or held high. Users may change the power-up address to suit their application by performing the address change sequence. It is recommended to avoid changing the address often, as it could cause premature EEPROM failure due to repeated erase/write

cycles. The sensor includes a safeguard of requiring two separate values to be sent in sequence before changing the stored address.
In order to change the address, the two address change commands, 0xAA (decimal 170) and 0xA5 (decimal 165) must be sent to the sensor sequentially in a single message, with no other bytes in between the commands. The next byte received in the same message will become the new address. If there are no bytes sent within the same message on the bus to the sensor after receiving the two sequential change address commands, the commands will need to be resent on another message if the sensor needs to have the address changed. The sensor will not accept a new address of 0x00 (decimal 0); if the new address of 00 is sent after receiving the two commands to change the address, the sensor will ignore the new address and continue waiting for a valid new address to be received, unless the bus master stops the transaction with a stop or restart condition. The address must be written in an 8-bit read/write format to the sensor (i.e. one would write 0xE0 to the sensor to set the EEPROM address back to the shipped default).
After receiving a valid byte to change the I2C address to, the sensor will write the new address to internal EEPROM, taking approximately 50 milliseconds.

Real-time Noise Rejection

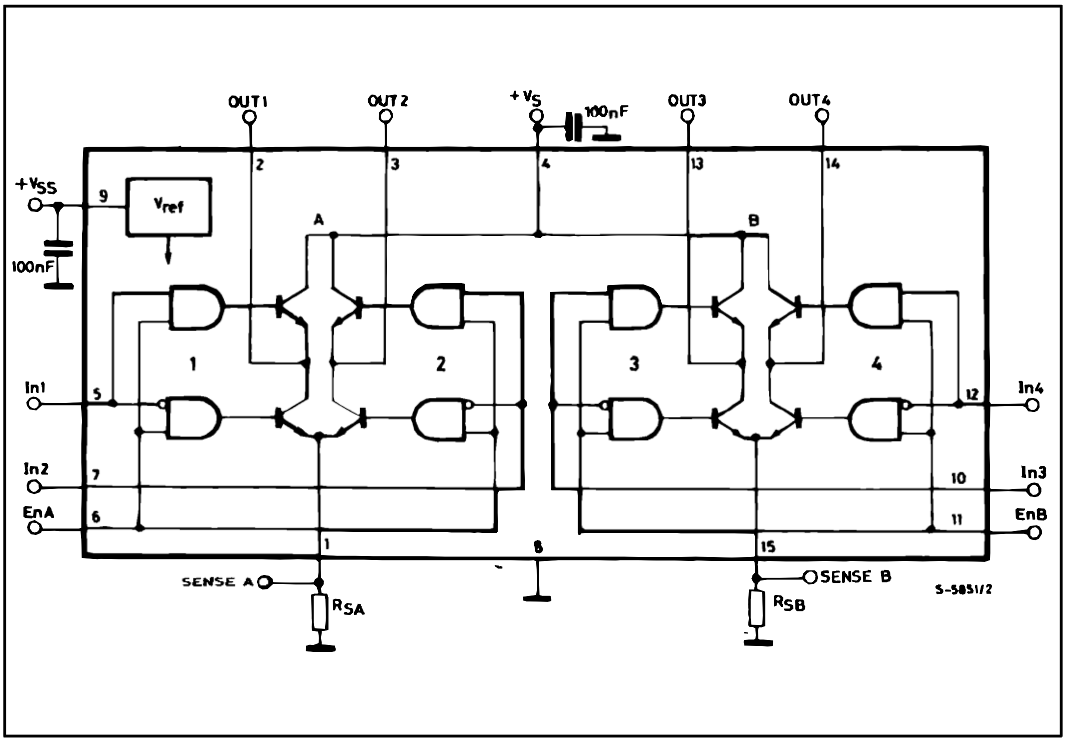
While the I2CXL-MaxSonar®-EZ is designed to operate in the presence of noise, best operation is obtained when noise strength is low and desired signal strength is high. The user is encouraged to mount the sensor in such a way that minimizes outside acoustic noise pickup. In addition, keep the DC power to the sensor free of noise. This will let the sensor deal with noise issues outside of the users direct control (in general, the sensor will still function well even if these things are ignored). Users are encouraged to test the sensor in their application to verify usability.
For every ranging cycle, individual filtering for that specific cycle is applied. In general, noise from regularly occurring periodic noise sources such as motors, fans, vibration, etc., will not falsely be detected as an object. This holds true even if the periodic noise increases or decreases (such as might occur in engine throttling or an increase/decrease of wind movement over the sensor). Even so, it is possible for sharp non-periodic noise sources to cause false target detection.
In addition, *(because of dynamic range and signal to noise physics,) as the noise level increases, at first only small targets might be missed, but if noise increases to very high levels, it is likely that even large targets will be missed.

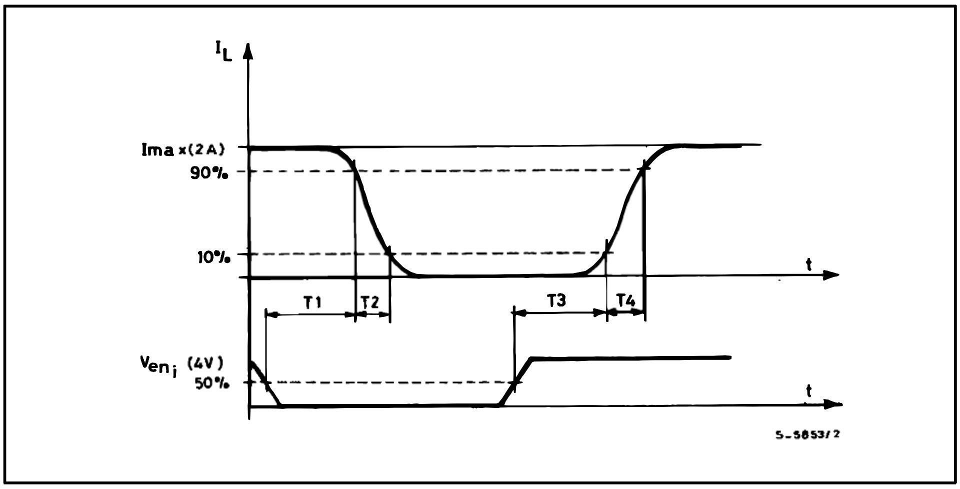
The I2CXL-MaxSonar-EZ series has additional resistance against periodic noise and small target rejection capabilities over the standard XL-MaxSonar-EZ/AE series.

APPENDIX B

L298

The L298 is an integrated monolithic circuit in a 15- lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver Two enable inputs are provided to enable or disable the device independently of the in- put signals. The emitters of the lower transistors of each bridge are connected together and the corre- sponding external terminal can be used for the connection.





POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in com- mon or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (RSA ; RSB.) allows to detect the in- tensity of this current.

INPUT STAGE

Each bridge is driven by means of four gates the in- put of which are In1 ; In2 ; EnA and In3 ; In4 ; EnB. The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

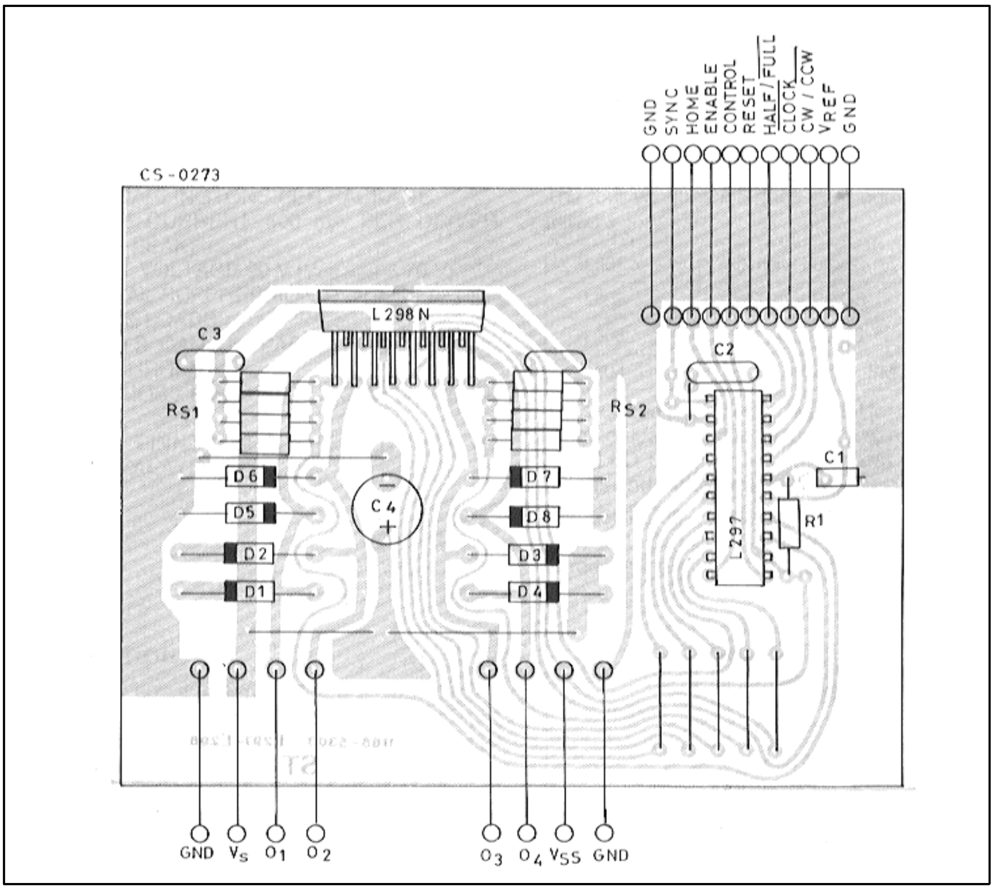
SUGGESTIONS

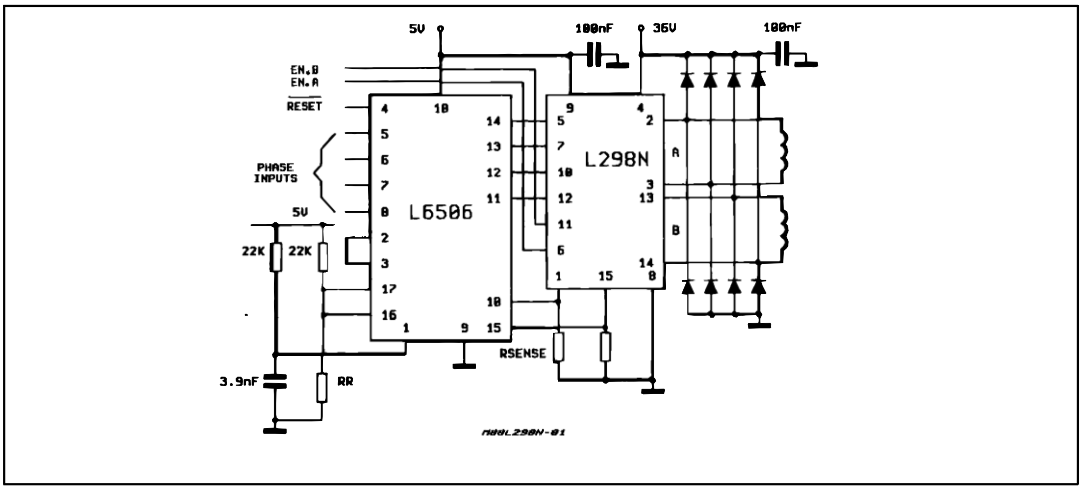
A non inductive capacitor, usually of 100 nF, must be foreseen between both Vs and Vss, to ground, as near as possible to GND pin. When the large ca- pacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

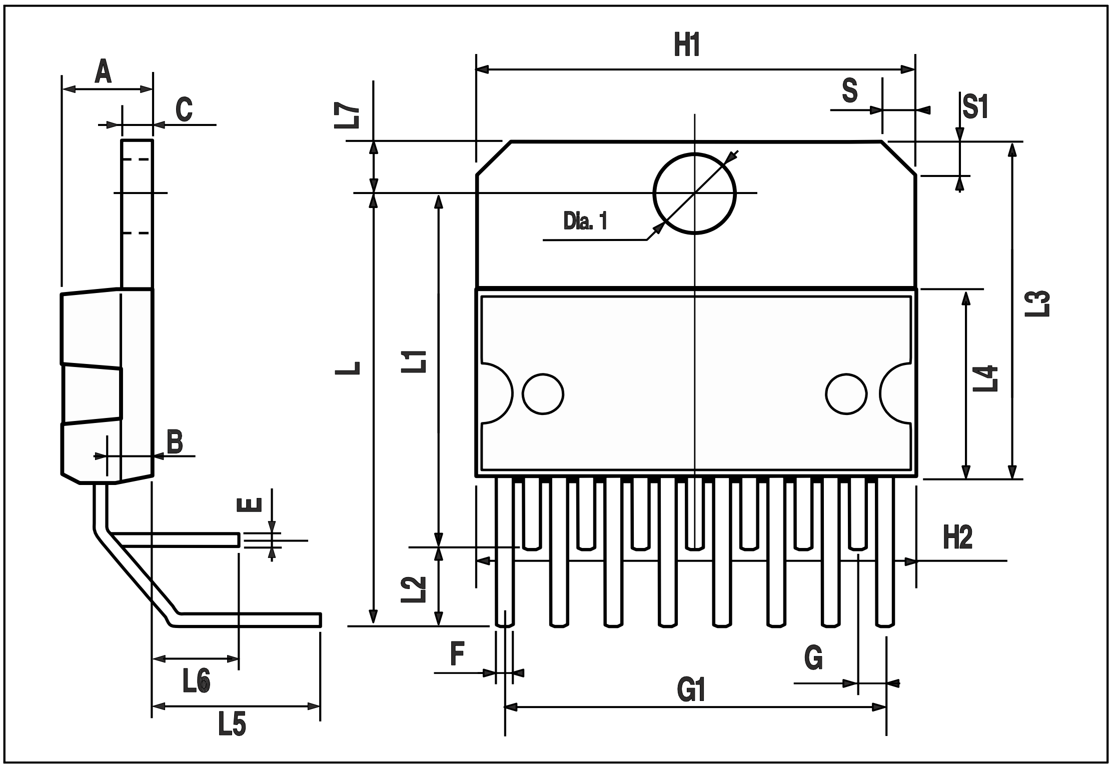
APPLICATIONS

Fig 6 shows a bidirectional DC motor control Sche- matic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements (trr 200 nsec) that must be chosen of a VF as low as possible at the worst case of the load current.
The sense output voltage can be used to control the current amplitude by chopping the inputs, or to pro- vide overcurrent protection by switching low the en- able input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.When the repetitive peak current needed from the load is higher than 2
An external bridge of diodes are required when in- ductive loads are driven and when the inputs of the IC are chopped ; Shottky diodes would be preferred




5V
100nF
36V
100nF
EN.B
EN.A
RESET
PHASE INPUTS
L6506
L298N
5V
22K
22K
3.9nF
RR
RSENSE
A
B


A
C
B
E
L6
L5
H1
S
S1
L7
Dia. 1
L
L1
L2
L3
L4
H2
F
G
G1

APPENDIX C

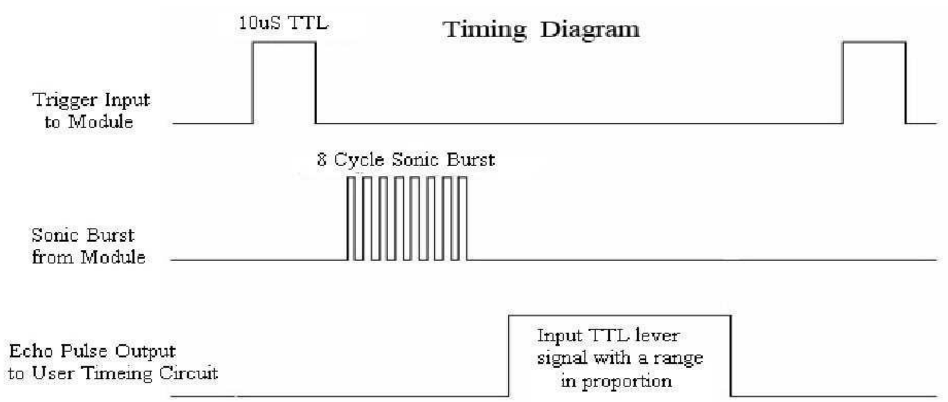
Ultrasonic Ranging Module HC - SR04

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

(1) Using IO trigger for at least 10us high level signal,
(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
(3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.
Test distance = (high level time×velocity of sound (340M/S) / 2,

Timing diagram

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can
calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Attention:
The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise ,it will affect the results of measuring.