CHAPTER 1
INTRODUCTION

1.1 GENERAL BACKGROUND

Based on our survey, in 2018 alone the number of lives fallen on roads recorded around 2 lakh. The major reason behind this is the delay in taking the injured ones to the hospital. Our system aims to reduce the delay in taking the victims to the hospital and tries to speed up the rescue operation at a faster rate .Raspberry pi is used here. The location of the vehicle is tracked by using GPS module. When accident take place an alert message is send to the nearby hospital. A video camera is interfaced with raspberry pi to capture the photograph of vehicles which can cause accident. By calculating the depth using ultrasonic sensor the photographs are captured and if accident happens, these photographs are mailed to nearby police station. Or else it is automatically erased. Once the mail is received by the nearby police station, the police authorities can take down this case as soon as possible. Thus the rescue operation can be done at a faster rate. This system act as the accident detection and alert system which will alert the nearby hospital and police station whenever accident occurs. The rescue operation is done at a very faster rate. Thus the lives fallen on the roads can be saved.

1.2 WHAT IS AN ACCIDENT ALERT AND DETECTION SYSTEM?

This system is based on new technology, its main purpose is to detect an accident and send alert to the nearby hospital, so the victim can find some help as soon as possible. It can detect accidents the intensity of the accident without any visual contact from control room. If this system is inserted in every vehicle then it is easy to understand how many vehicles are involved in a particular accident and how intense is it. So that the help from control room will be according to the message received at the control room. The present board designed has both vehicle tracking and accident alert systems, which make it more valuable and useful. This board alerts us from theft and on accident detection also. The special feature of this project is that the photograph of accident caused vehicle is send the very next moment to the accident to the nearby police station.

CHAPTER 2
LITERATURE SURVEY

For ACCIDENT ALERT AND DETECTION SYSTEM we went through following journals, we studied and analyzed all the aspects in making the hospital authority high alert to make the rescue operations as soon as possible right after the occurrence of accident.

Reference [1] proposed an Accident alert system based on microcontroller AT89S52 is used to provide security to the vehicle in very reasonable cost. For this project C programming is used for better accuracy and for tracing vehicles GPS and GSM modules are used. The user segment and control segment of GPS are used to trace the vehicle perfectly. The exact location of the vehicle and the alert message are send using GSM module. The sensor used is accelerometer. When the accelerometer senses the accident an information is sent to the microcontroller .The GPS and GSM module of this system are interfaced to the same microcontroller .The GSM modem is used to send the data to the contacts which are stored in the data base, also with this GSM information is sent to the police station.

In [2], the sensor used is MEMS based accelerometer module. When the sensor senses the acceleration or deceleration above a threshold value a message is send to the stored numbers. In order to get immediate help in situations like heart attack, asthma, allergy etc... , there is switch called “panic switch” which is placed inside the vehicle. In case of minor accidents, when the message has been sent the occupant can call back and acknowledge that he is fine.

In[3] this accident alert system is based on Raspberry pi. Raspberry pi is a microprocessor. The major advantage of raspberry pi over arduino is that it can use any programming language. It is miniaturized computed. This is the most advanced version. Using this the accident alert system is made. The sensor senses the static as well as dynamic shock of the vehicle. When the sensor output is above a certain threshold voltage level a message is send to the nearby hospital via GPS module. The location is identified by the GPS module.

There is a cloud storage to store data and an electronic health record storage system to record the electronic health of the patient. The health record of the patient is stored directly to the cloud in this system RFID technology is being used. In Raspberry pi the RFID reader is embedded. RFID reader provides reading and writing operation to the reader. RFID system is used to access the health record of the patient. GPS module is used to locate the position of the vehicle. The sensor used is ADXL345.This project aims at increasing the chances of survival of the accident victim.

In [4], author suggested about the security of the traveler. The mechanism used is wrong path alert mechanism by using smart phone technology for providing safety and security to the traveler. Whenever the accident occurs this system ends the current location and the speed of vehicle to the owners mobile. This system mainly focuses on the safety of students. The change in pressure or acceleration is measured by the sensor. The sensor used here is piezoelectric sensor. Using GPS module the location of the vehicle is tracked and send to the parents mobile which will help in locating the vehicle as soon as possible.

It[5] this system is also GPS and GSM based system. In the case of network failure of GSM module a redundant technology, VANET is used. Using this the emergency message is delivered. VANET is a wireless communication network and they communicate through the wireless link which is connected on each node of the vehicle. The main characteristics of VANET are speed and quick pattern movement, higher node mobility. The routing of VANET is achieved by selecting the source and destination. The signal strength between the nodes is measured and the average value is considered. If the received signal strength is close to the measured average value, then it is further processed or else it is discarded. This system focuses on rescuing the accident victim even in the areas which has no network.

In[6] combination of both hardware and software is Internet of Things (IOT). This system reduces the human effort. Components like accelerometer, GSM module, GPS module, IR sensor, microcontroller, Wi-Fi module, LCD display, and power supply are used in this system. All these components are connected to central processing unit, the microcontroller. The system enhances the security of the traveler.

In[7] this system enhances the security by sending SMS to the predefined mobile numbers. Both GPS and GSM modules are used. The sensor used is piezoelectric sensor. The microprocessor used is ARM7TDMI-S. It is a 32-bit microprocessor. The ARM architecture is based on RISC and CISC. RISC is reduced instruction set computer and CISC is complex instruction set computer. The accident alert SMS is send to the pre stored numbers. These numbers are stored in EEPROM which will retain the data stored in it even when the power is off. The major advantage of this system is the easiness to operate. The design is simple and reliable. The major disadvantage of this system is that it does not work without internet.

In[8] this system proposes to detect the accident and alert message is send to the rescue team as soon as possible. GPS and GSM modules are used. The sensor used in piezoelectric sensor. The number to which the alert message to be send is stored in EEPROM. There is switch called OFF switch in order to avoid false SMS. The ARDUINO Mega ATmega2560 is used here. Arduino supports the programming languages C and C++. The GSM module used is SIM800A. The major advantage of his system is the easiness to operate. The major disadvantage of this system is that the network is mandatory.

CHAPTER 3
EXISTING SYSTEM

Our existing system is Arduino based vehicle accident alert system using GPS, GSM modules and Accelerometer. Accelerometer senses the sudden acceleration or deceleration of the vehicle and GSM module sends the alert message on the Mobile Phone with the location of the accident. Location of accident is sent in the form of Google Map link, derived from the latitude and longitude from GPS module. The Message also contains the speed of vehicle in knots... This Vehicle Accident alert project can also be used as a Tracking System and much more, by just making few changes in hardware and software. Arduino Uno plays the key role here. The output form the accelerometer i.e., an analog voltage is given as the input to the Arduino. It runs the code and send the message of “HELP”, if necessary, to the receiver at the hospital using GSM module. Otherwise, the message will not be send. The decision of whether the message must be send or not is decided by the condition given.

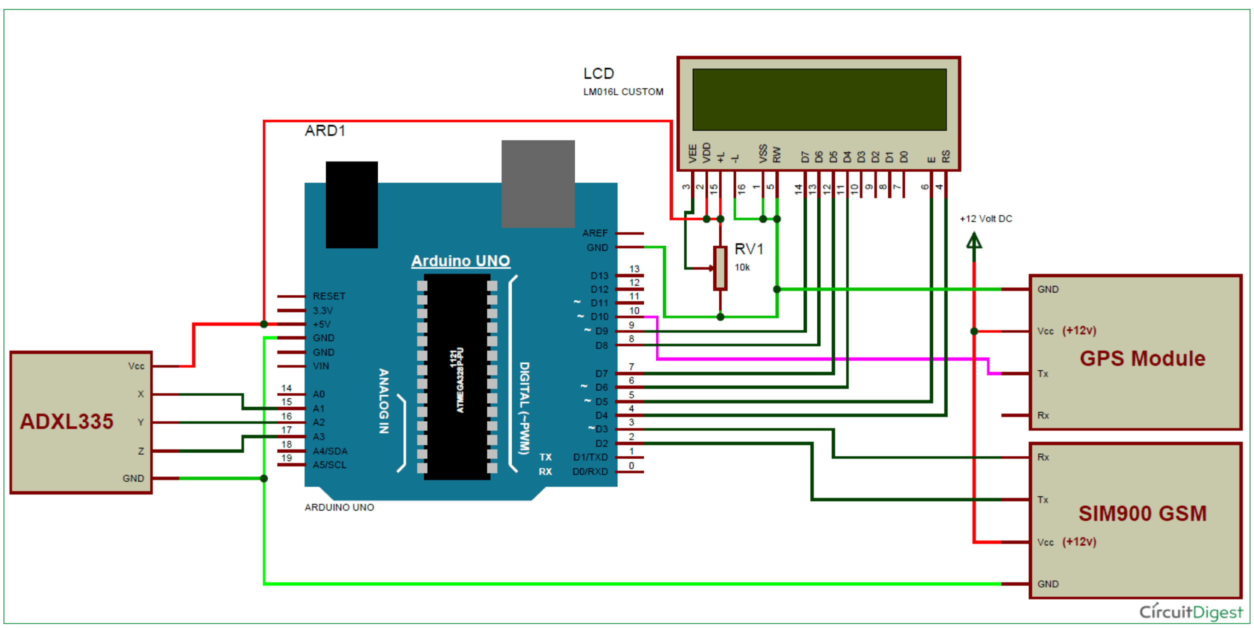


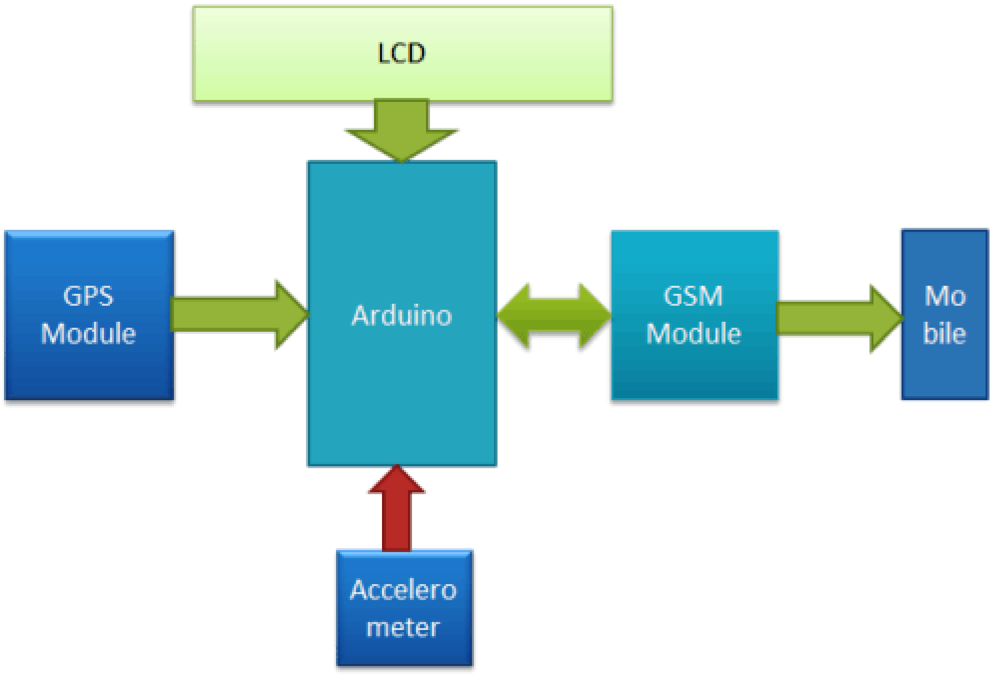
Figure 1: Working diagram of existing system

Circuit Connections of this project is drawn above. Here Tx pin of GPS module is directly connected to digital pin number 10 of Arduino. By using Software Serial Library here, we have allowed serial communication on pin 10 and 11, and made them Rx and Tx respectively and left the Rx pin of GPS Module open. By default Pin 0 and 1 of Arduino are used for serial communication but by using the Software Serial library, we can allow serial communication on other digital pins of the Arduino. 12 Volt supply is used to power the GPS Module. GSM module's Tx and Rx pins of are directly connected to pin D2 and D3 of Arduino. For GSM interfacing, here we have also used software serial library. GSM module is also powered by 12v supply. An optional LCD's data pins D4, D5, D6, and D7 are connected to pin number 6, 7, 8, and 9 of Arduino. Command pin RS and EN of LCD are connected with pin number 4 and 5 of Arduino and RW pin is directly connected with ground. A Potentiometer is also used for setting contrast or brightness of LCD. An accelerometer is added in this system for detecting an accident and its x, y, and z-axis ADC output pin are directly connected to Arduino ADC pin A1, A2, and A3.

3.1 WORKING

In this project, Arduino is used for controlling whole the process with a GPS receiver and GSM module. GPS Receiver is used for detecting coordinates of the vehicle, GSM module is used for sending the alert SMS with the coordinates and the link to Google Map. Accelerometer namely ADXL335 is used for detecting accident or sudden change in any axis. And an optional 16x2 LCD is also used for displaying status messages or coordinates. We have used GPS Module SIM28ML and GSM Module SIM900A.

3.1 BLOCK DIAGRAM



When we are ready with our hardware after programming, we can install it in our vehicle and power it up. Now whenever there is an accident, the car gets tilt and accelerometer changes his axis values. These values read by Arduino and checks if any change occurs in any axis. If any change occurs then Arduino reads coordinates by extracting $GPGGA String from GPS module data (GPS working explained above) and send SMS to the predefined number to the police or ambulance or family member with the location coordinates of accident place. The message also contains a Google Map link to the accident location, so that location can be easily tracked. When we receive the message then we only need to click the link and we will redirect to the Google map and then we can see the exact location of the vehicle. Speed of Vehicle, in knots (1.852 KPH), is also sent in the SMS and displayed on the LCD panel.

CHAPTER 4
PROPOSED PROJECT

The main components used in this project are Raspberry Pi (Model B+), mobile phone, Power Supply, Video camera, Ultrasonic sensor. HC-SR04 is the sensor used here. The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module. And the distance is calculated. If the calculated distance is less than the threshold value, then the picture is captured by the video camera. This property can be used to capture the photographs of the hitting vehicle at the time of accident. Ultrasonic sensor is interfaced with Arduino Uno. When the ultrasonic sensor senses the distance, Arduino Uno give command to the Raspberry pi and the Raspberry pi initializes the camera module. The camera module takes the photograph of the accident caused vehicle. The mobile application used is the Share GPS. This application is used to share the accident location to the nearby hospital. Thus the rescue operation is done at a faster rate.

4.1COMPONENTS REQUIRED

4.1.1 RASPBERRY PI

It is a credit card sized computer powered by the Broadcom BCM2835 system on a chip (SoC). This SoC includes a 32-bit ARMV6 processor, clocked at 700MHZ and a video core IV GPU. It also has 256MB of RAM in a POP package above the SoC. The Raspberry pi is powered by a 5V micro USB AC charger. Processor speed ranges from 700MHz to 4GHz. SD cards are used to store the operating system and program memoryPi .Raspbian is the official operating system of Raspberry. The Raspberry Pi 3 Model B+ is the latest production Raspberry Pi 3 featuring a 64-bit quad core processor running at 1.4 GHz.

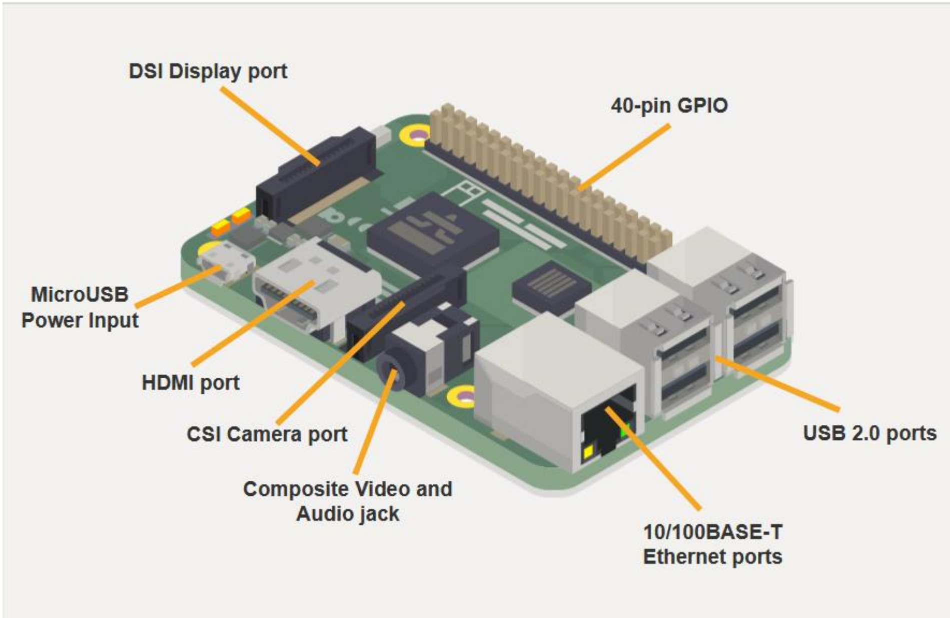


Figure 2: Raspberry pi 3 model B plus

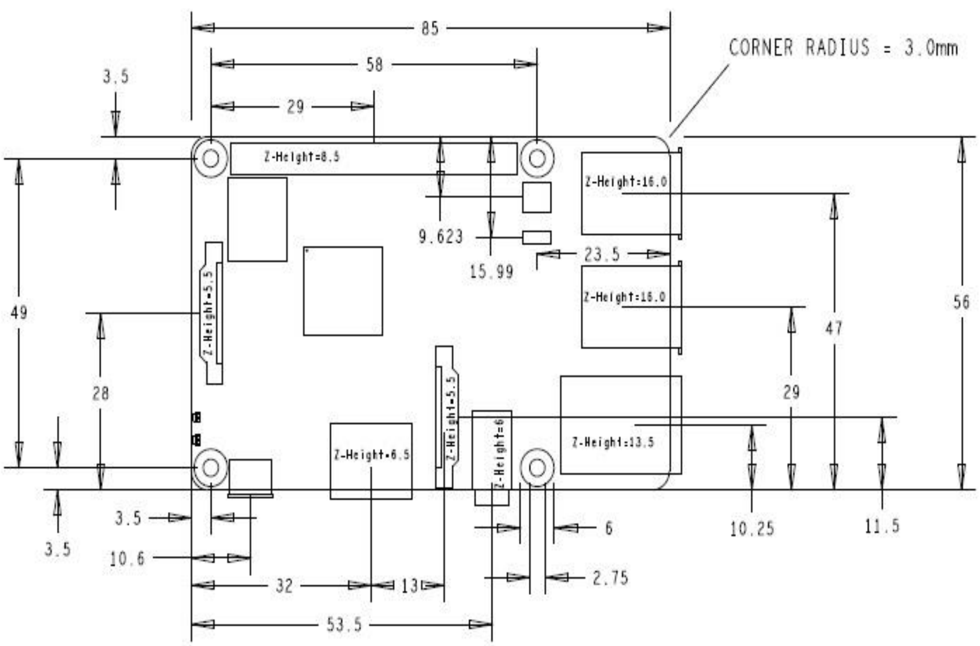


Figure 3: Mechanical drawing of Raspberry pi

SPECIFICATIONS

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input

HARDWARE

The Raspberry Pi hardware has evolved through several versions that feature variations in memory capacity and peripheral-device support.

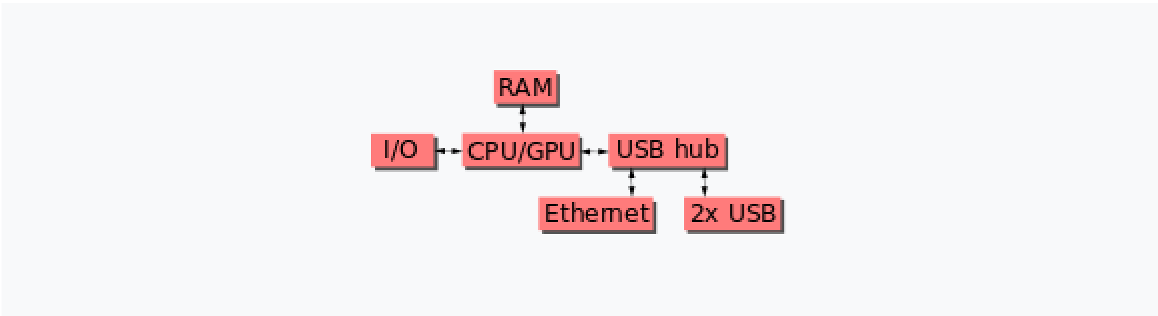


Figure 4: Hardware of Raspberry pi

This block diagram describes Model B and B+; Model A, A+, and the Pi Zero are similar, but lack the Ethernet and USB hub components. The Ethernet adapter is internally connected to an additional USB port. In Model A, A+, and the Pi Zero, the USB port is connected directly to the system on chip (SoC). On the Pi 1 Model B+ and later models the USB/Ethernet chip contains a five-port USB hub, of which four ports are available, while the Pi 1 Model B only provides two. On the Pi Zero, the USB port is also connected directly to the SoC, but it uses a micro USB (OTG) port.

PROCESSOR

The Broadcom BCM2835 SoC used in the first generation Raspberry Pi includes a 700 MHz ARM11 76JZF-S processor, VideoCoreIV graphics processing unit (GPU), and RAM. It has a level 1 (L1) cacheof 16 KB and a level 2 (L2) cache of 128 KB. The level 2 cache is used primarily by the GPU. The SoC is stacked underneath the RAM chip, so only its edge is visible. The 1176JZ (F)-S is the same CPU used in the original iPhone, although at a higher clock rate, and mated with a much faster GPU.The earlier V1.1 model of the Raspberry Pi 2 used a Broadcom BCM2836 SoC with a 900 MHz 32-bit, quad-core ARM Cortex-A7processor, with 256 KB shared L2 cache. The Raspberry Pi 2 V1.2 was upgraded to a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, the same SoC which is used on the Raspberry Pi 3, but under clocked (by default) to the same 900 MHz CPU clock speed as the V1. The Raspberry Pi 3+ uses a Broadcom BCM2837B0 SoC with a 1.4 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache. The Raspberry Pi Zero and ZeroW use the same Broadcom BCM2835 SoC as the first generation Raspberry Pi, although now running at 1GHz CPU clock speed.

PERFORMANCE

While operating at 700 MHz by default, the first generation Raspberry Pi provided a real-world performance roughly equivalent to 0.041 GFLOPS. On the CPU level the performance is similar to a 300 MHz Pentium II of 1997–99. The GPU provides 1 Gpixel/s or 1.5 Gtexel/s of graphics processing or 24 GFLOPS of general purpose computing performance. The graphical capabilities of the Raspberry Pi are roughly equivalent to the performance of the Xbox of 2001. Raspberry Pi 2 V1.1 included a quad-core Cortex-A7 CPU running at 900 MHz and 1 GB RAM. It was described as 4–6 times more powerful than its predecessor. The GPU was identical to the original. In parallelised benchmarks, the Raspberry Pi 2 V1.1 could be up to 14 times faster than a Raspberry Pi 1 Model B+. The Raspberry Pi 3, with a quad-core ARM Cortex-A53 processor, is described as having ten times the performance of a Raspberry Pi 1. This was suggested to be highly dependent upon task threading and instruction set use. Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelised tasks.

RAM

On the older beta Model B boards, 128 MB was allocated by default to the GPU, leaving 128 MB for the CPU. On the first 256 MB release Model B (and Model A), three different splits were possible. The default split was 192 MB (RAM for CPU), which should be sufficient for standalone 1080p video decoding, or for simple 3D, but probably not for both together. 224 MB was for Linux only, with only a 1080p frame buffer, and was likely to fail for any video or 3D.

PIN DIAGRAM

A powerful feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the top edge of the board. A 40-pin GPIO header is found on all current Raspberry Pi boards (unpopulated on Pi Zero and Pi Zero W). Prior to the Pi 1 Model B+ (2014), boards comprised a shorter 26-pin header. Any of the GPIO pins can be designated (in software) as an input or output pin and used for a wide range of purposes.

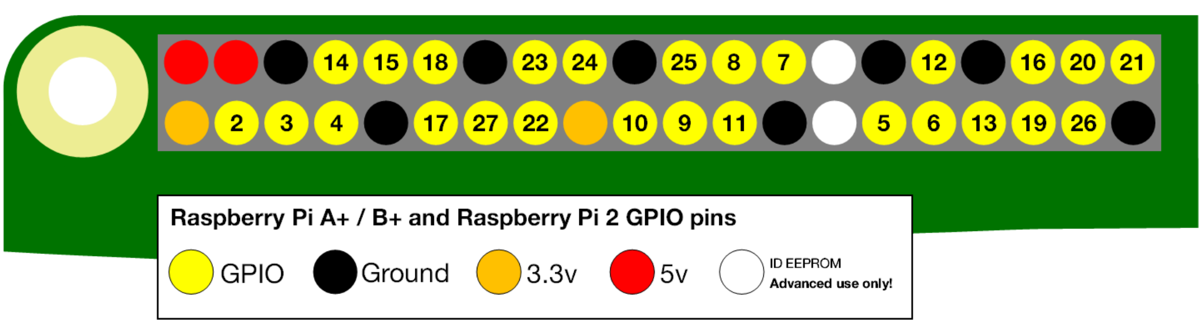


Fig 5: Pin diagram of Raspberry pi

Two 5V pins and two 3V3 pins are present on the board, as well as a number of ground pins (0V), which are un configurable. The remaining pins are all general purpose 3V3 pins, meaning outputs are set to 3V3 and inputs are 3V3-tolerant. A GPIO pin designated as an output pin can be set to high (3V3) or low (0V). A GPIO pin designated as an input pin can be read as high (3V3) or low (0V). This is made easier with the use of internal pull-up or pull-down resistors. Pins GPIO2 and GPIO3 have fixed pull-up resistors, but for other pins this can be configured in software. As well as simple input and output devices, the GPIO pins can be used with a variety of alternative functions, some are available on all pins, others on specific pins.

- PWM (pulse-width modulation)
    - Software PWM available on all pins
    - Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19
- SPI
    - SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
    - SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)
- I2C
    - Data: (GPIO2); Clock (GPIO3)
    - EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)
- Serial
    - TX (GPIO14); RX (GPIO15)

4.1.2 RASPBERRY PI CAMERA MODULE V2 1080P

This 8mp camera module is capable of 1080p video and still images and connects directly to your Raspberry Pi. Connect the included ribbon cable to the CSI (Camera Serial Interface) port on your Raspberry Pi, boot up the latest version of Raspbian. The board itself is tiny, at around 25mm x 20mm x 9mm and weighing in at just over 3g, making it perfect for mobile or other applications where size and weight are important. The sensor has a native resolution of 5 megapixel and has a fixed focus lens on-board. In terms of still images, the camera is capable of 2592 x 1944 pixel static images and also supports 1080p30, 720p60 and 640x480p60/90 video. This module is only capable of taking pictures and video, not sound.

Fig 6: Raspberry pi camera

FEATURES

- Fixed focus lens on-board
- 8 megapixel native resolution sensor-capable of 3280 x 2464 pixel static images
- Supports 1080p30, 720p60 and 640x480p90 video
- Size 25mm x 23mm x 9mm
- Weight just over 3g
- Connects to the Raspberry Pi board via a short ribbon cable (supplied)
- Camera v2 is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system

4.1.3 ARDUINO UNO

The Arduino UNO is an open source microcontroller microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is also similar to the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

The word "UNO" means "one" in Italian and was chosen to mark the initial release of the Arduino Software. The Uno board is the first in a series of USB-based Arduino boards, and it and version 1.0 of the Arduino IDE were the reference versions of Arduino, now evolved to newer releases. The ATmega328 on the board comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer.

Fig 7: Arduino Uno

SPECIFICATIONS

- Operating Voltage: 5 Volts
- Input Voltage: 7 to 20 Volts
- Microcontroller: Microchip ATmega328P
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA

- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by boot loader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz

GENERAL PIN DIAGRAM

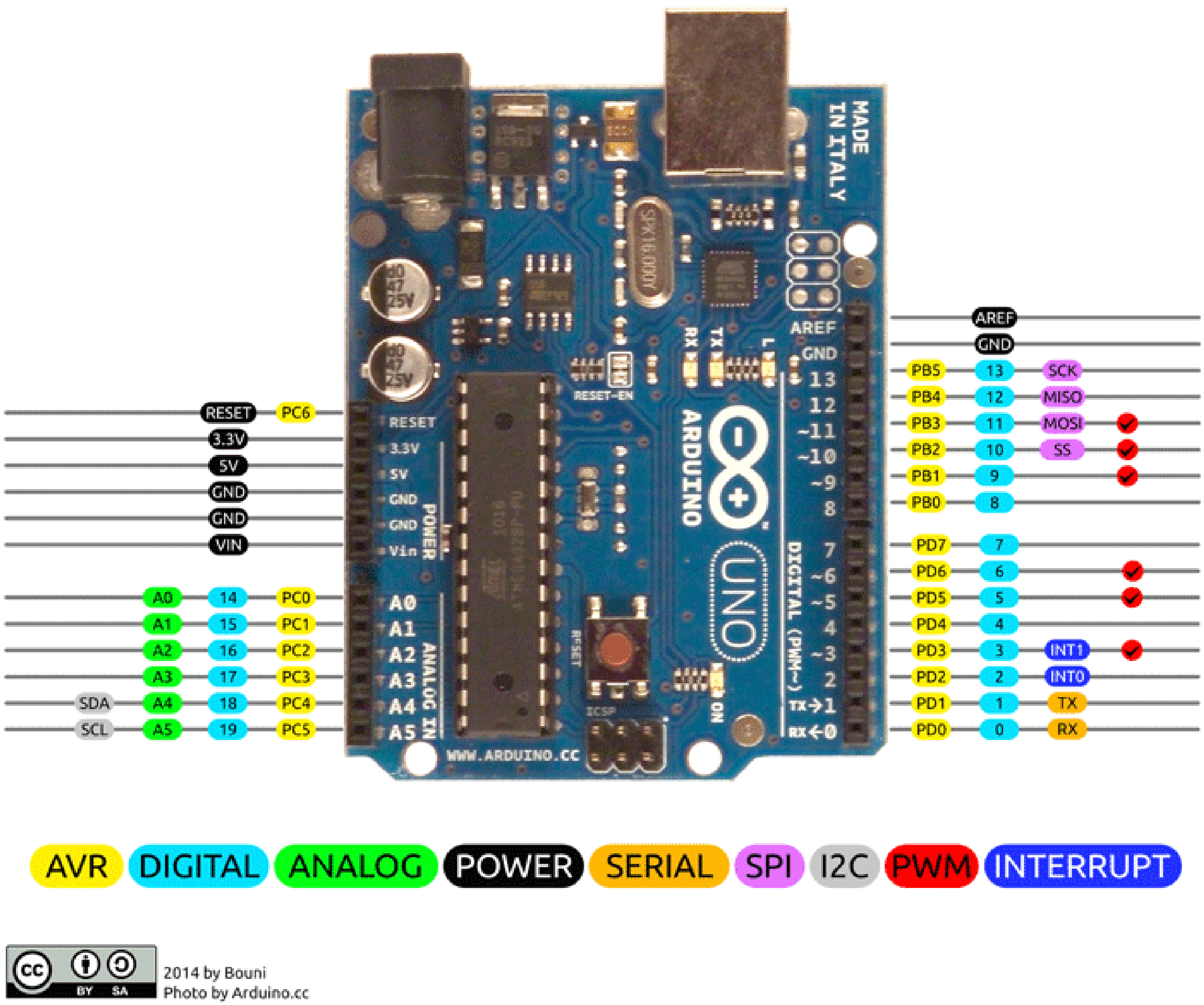


Fig 8: Pin diagram of Arduino Uno

Pin Category	Pin Name	Details
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output Pins	Digital Pins 0 - 13	Can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.

Arduino Uno Technical Specifications

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

GENERAL PIN FUNCTIONS

- **LED**: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it's off.
- **VIN**: The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V),

- or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
- **3V3**: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND**: Ground pins.
- **IOREF**: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
- **RESET**: Typically used to add a reset button to shields which block the one on the board.

SPECIAL PIN FUNCTION

Each of the 14 digital pins and 6 analog pins on the Uno can be used as an input or output, using pinMode (), digitalWrite (), and digitalRead () functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference () function.

In addition, some pins have specialized functions:

- **Serial** / UART: pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.
- **External interrupts**: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM** (pulse-width modulation): 3, 5, 6, 9, 10, and 11. Can provide 8-bit PWM output with the analogWrite () function.

- **SPI** (Serial Peripheral Interface): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- **TWI** (two-wire interface) / I²C: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
- **AREF** (analog reference): Reference voltage for the analog inputs.

COMMUNICATION

The Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows serial communication on any of the Uno's digital pins.

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 Nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. This setup has other implications. When the Uno is connected to a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the boot loader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened.

APPLICATIONS

- Arduino Uno is used in prototyping.
- In developing projects based on code-based control
- Development of Automation System
- Designing of basic circuit designs.

4.1.4 ULTRASONIC SENSOR

Ultrasonic sensors measure distance by using ultrasonic waves. The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.

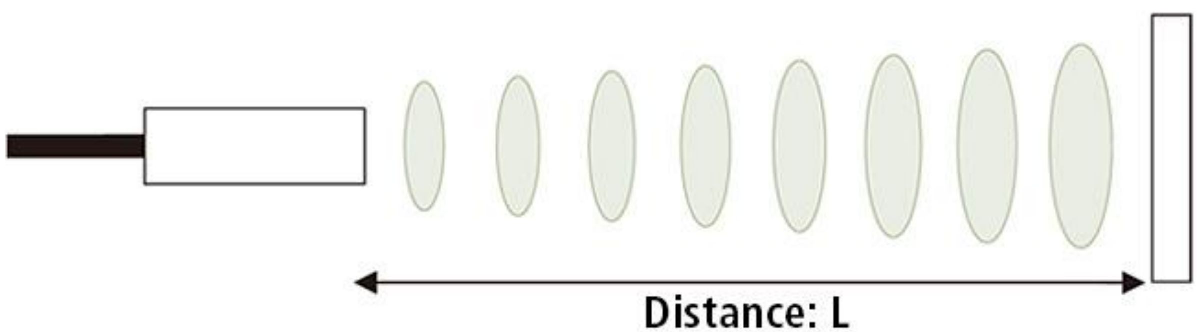


Fig 9: Ultrasonic sensor distance measurement

An optical sensor has a transmitter and receiver, whereas an ultrasonic sensor uses a single ultrasonic element for both emission and reception. In a reflective model ultrasonic sensor, a single oscillator emits and receives ultrasonic waves alternately. This enables miniaturization of the sensor head.

The distance can be calculated with the following formula:

Distance $L = 1/2 \times T \times C$

Where L is the distance, T is the time between the emission and reception, and C is the sonic speed. (The value is multiplied by 1/2 because T is the time for go-and-return distance)

HC-SR04

It has stable performance and high ranging accuracy make it a popular module in electronic market. Compared to the Shap IR ranging module, HC-SR04 is more inexpensive than it.

Fig 10:HC-SR04

SPECIFICATIONS

- Power supply: 5V DC
- Quiescent current: <2mA
- Effectual angle: <15°
- Ranging distance: 2cm – 500 cm
- Resolution: 1 cm

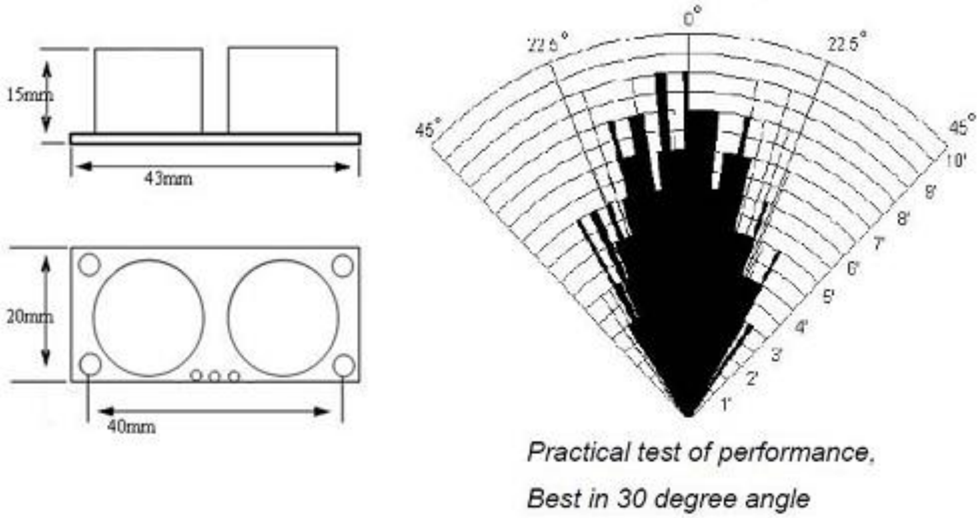
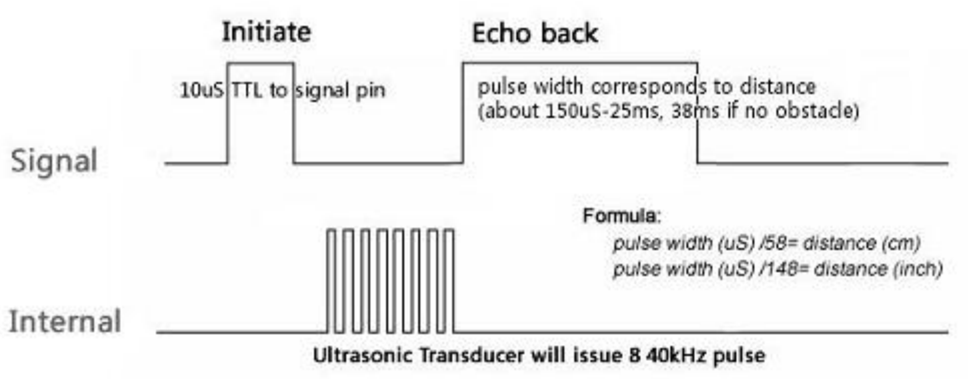


Fig 11: Schematic Diagram of HC-SR04

SEQUENCE CHART



A short ultrasonic pulse is transmitted at the time 0, reflected by an object. The senor receives this signal and converts it to an electric signal. The next pulse can be transmitted when the echo is faded away. This time period is called cycle period. The recommend cycle period should be no less than 50ms. If a 10μs width trigger pulse is sent to the signal pin, the Ultrasonic module will output eight 40 kHz ultrasonic signal and detect the echo back. The measured distance is proportional to the echo pulse width and can be calculated by the formula above. If no obstacle is detected, the output pin will give a 38ms high level signal.

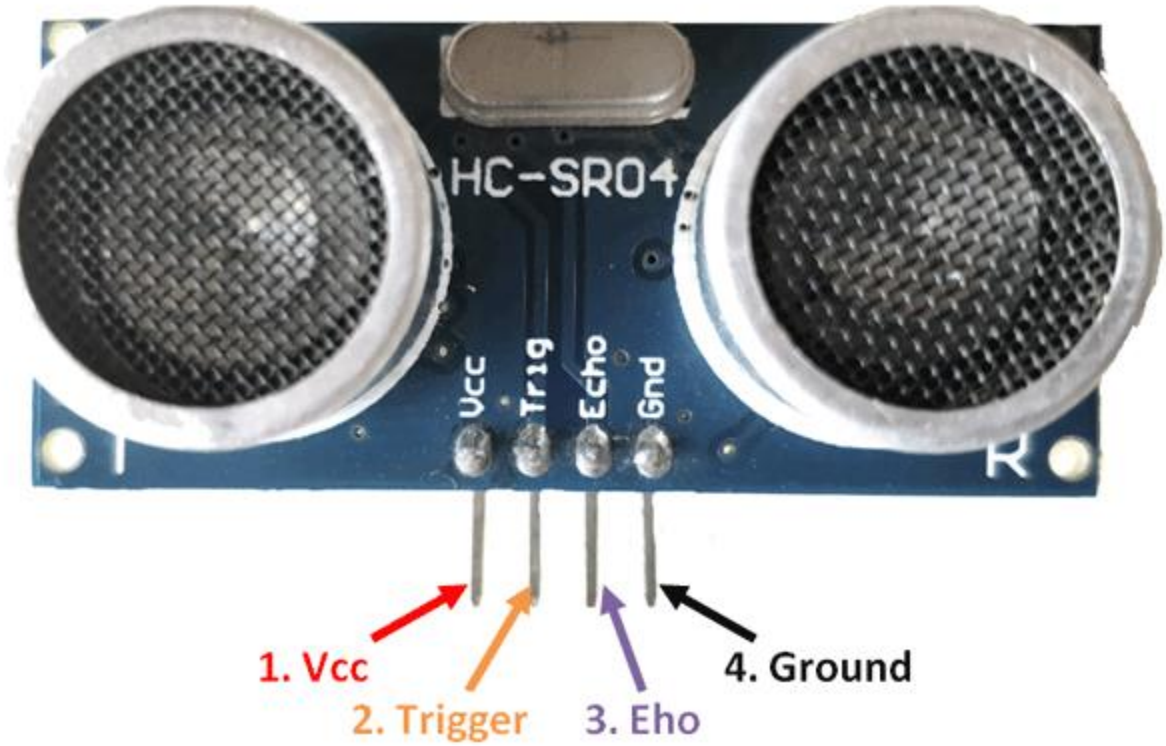


Fig 12: Pin Diagram of HC-SR04

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

As shown above the HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

$$\text{Distance} = \text{Speed} \times \text{Time}$$

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

Fig 13: Working of ultrasonic sensor

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor. HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used. Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both

I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will

trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor. The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading. For example, if the object is 10 cm away from the sensor, and the speed of the sound is 340 m/s or 0.034 cm/µs the sound wave will need to travel about 294 u seconds. But what you will get from the Echo pin will be double that number because the sound wave needs to travel forward and bounce backward. So in order to get the distance in cm we need to multiply the received travel time value from the echo pin by 0.034 and divide it by 2.

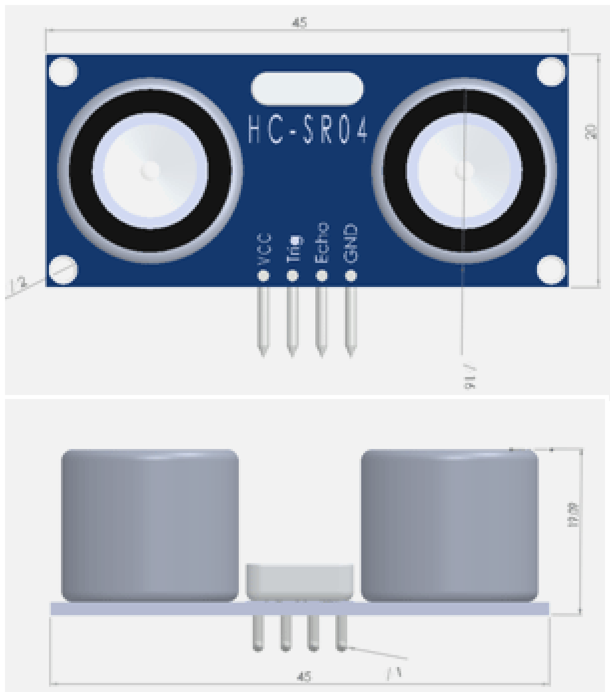
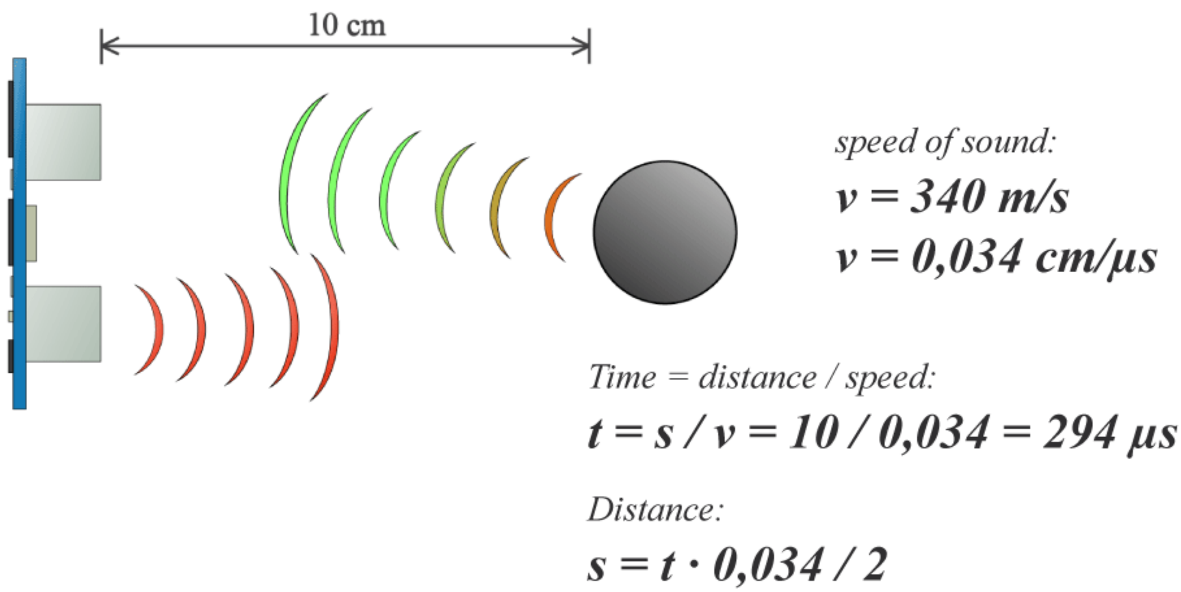


Fig 14: 2D diagram of Ultrasonic sensor

APPLICATIONS

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it

- Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water



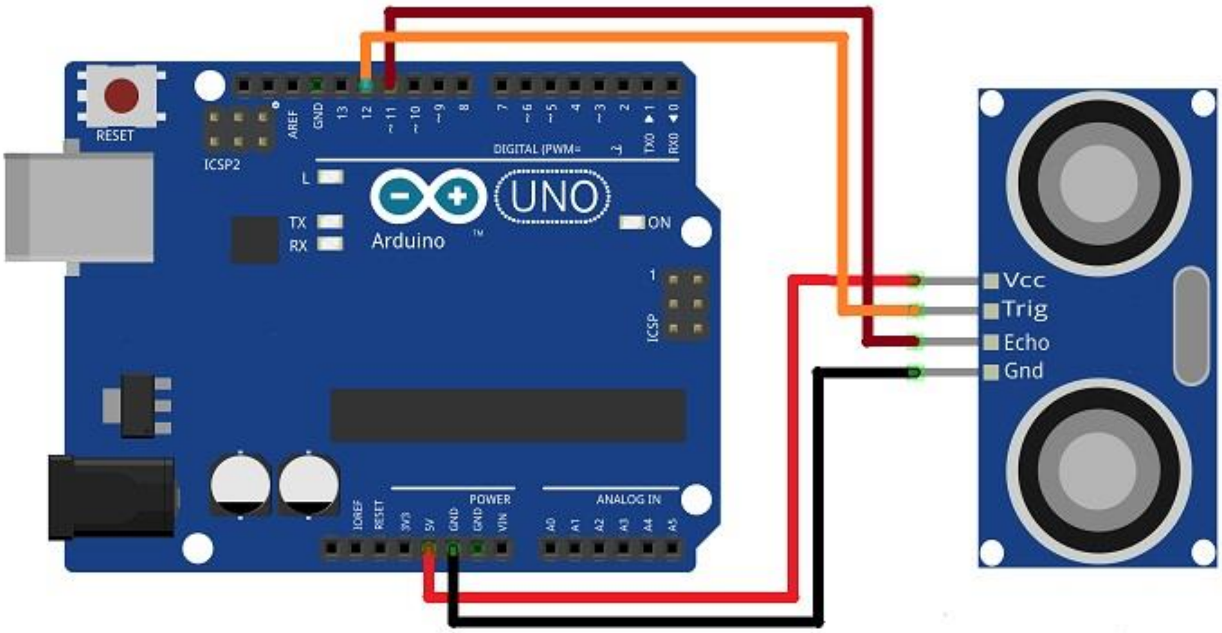


Fig 15: Connection diagram of Arduino and HC-SR04

4.1.5 SHARE GPS

It is android application for sharing GPS data with mapping applications.

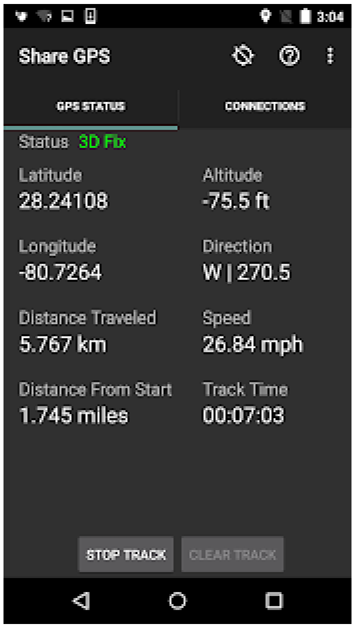


Fig 17: Share GPS

Share GPS supports the following major activities

- Use your mobile as an external GPS for laptops/tablets that don't have GPS.
- Share and update location real-time to a remote person using Google Earth or other KML compatible program
- Save locations visited and share later with images and links with KML
- Save tracks during activities such as driving, biking, and walking and share via KML files

Share NMEA GPS sentences or act as a GPSD compatible server via the following methods:

- Bluetooth
- TCP/IP (wifi/mobile data)
- USB

Share KML files via the following methods:

- Dropbox
- Google Drive
- SSH/SCP
- Local phone storage
- Provider application

Share GPS is an application for real-time location data sharing via a variety of methods. It also supports saving tracks via KML/KMZ, allowing the user to record footprints while the app is running in the background. Share GPS supports sending standard NMEA data over Bluetooth, USB, and TCP/IP. For local connections with a program like Google Earth, you could turn your laptop into a big screen navigation device, basically using your mobile as an external GPS. If you have a VPN setup, you could also do remote sharing of location data via TCP/IP over your 3G/4G or Wi-Fi connections. If it turns out your mobile does not support NMEA, Share GPS has a setting for creating the NMEA strings for you.

Share GPS also supports sending location data via the KML file format, compatible with Google Earth. KMZ files can be shared via SCP, Google Drive, Dropbox, Local File, or using another provider on the phone (email as an example). It supports periodic sending and location updating of the file, which would support many possibilities. Using a cloud solution with a shared folder, such as Drive or Dropbox, several users could send KMZ files periodically. Then a PC user with Google Earth could set it up to periodically read the files for real-time updating. Mobile users could also access the KMZ files as well with Google Earth or other applications. The status screen indicates current data, such as the GPS coordinates, altitude, speed, and direction. In addition, related to track collection, the status screen show distance traveled, distance from start (as the crow flies, useful for range measurements), and total track time. Any of the status screen items can be copied to the clipboard by pressing the item. Units can be changed by long pressing the item and choosing from the list of options. Share GPS contains other features as well. The app keeps a constant notification up, which can

be customized to include any of the status items, allowing for viewing on the lock screen for some mobiles. The notification is also used to indicate Share GPS is running, even when in the background. In addition, the copy location to clipboard is customizable, allowing the user to have a custom string that includes the location, for example, to create a URL for a website.

4.2 BLOCK DIAGRAM

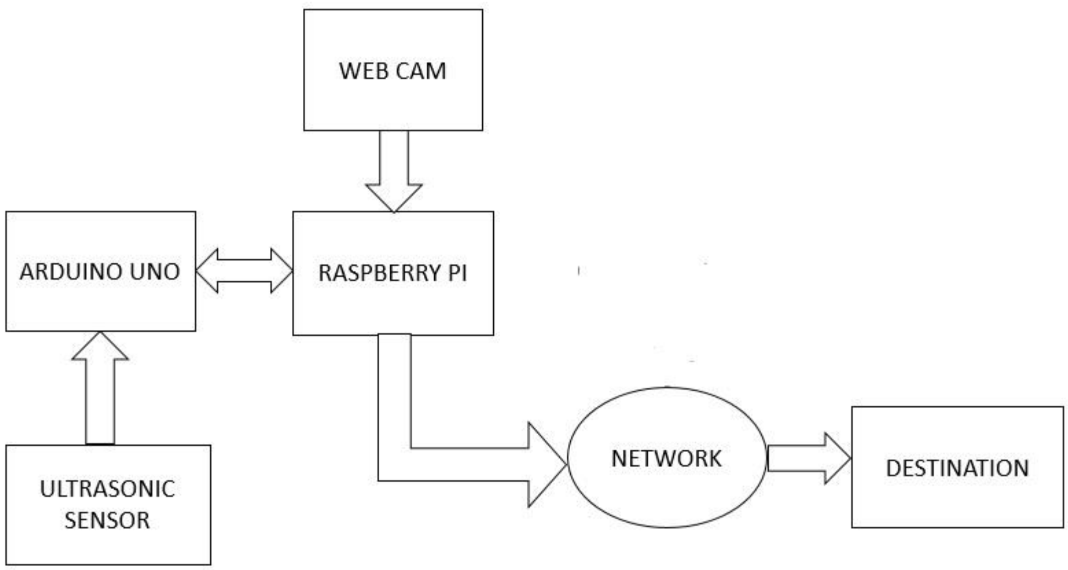


Fig 18: Block Diagram of Accident Alert and Detection System

Ultrasonic sensor is interfaced with Arduino Uno. If the sensor sensed value goes below the threshold value, then the Arduino give command to the Raspberry pi. The Raspberry pi initializes the camera module. The camera module takes the photograph of the accident caused vehicle. This photograph is mailed to the destination. The mobile application used is the Share GPS. This application shares the accident location to the nearby hospital.

CHAPTER 5
SYSTEM OPERATION

The programing language used in Raspberry pi 3 is python. The sensor used here is HC-SR04. The first step is sensor interfacing. The sensor is interfaced with Arduino Uno. HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver. And the distance is calculated. If the sensor sensed value goes below the threshold value, then the Arduino give command to the Raspberry pi. The Raspberry pi initializes the camera module. The camera module takes the photograph of the accident caused vehicle. This photograph is mailed to the destination. The mobile application used is the Share GPS. This application shares the accident location to the nearby hospital. Thus the rescue operation is done at a faster rate.

5.1 ALGORITHM

1.Start

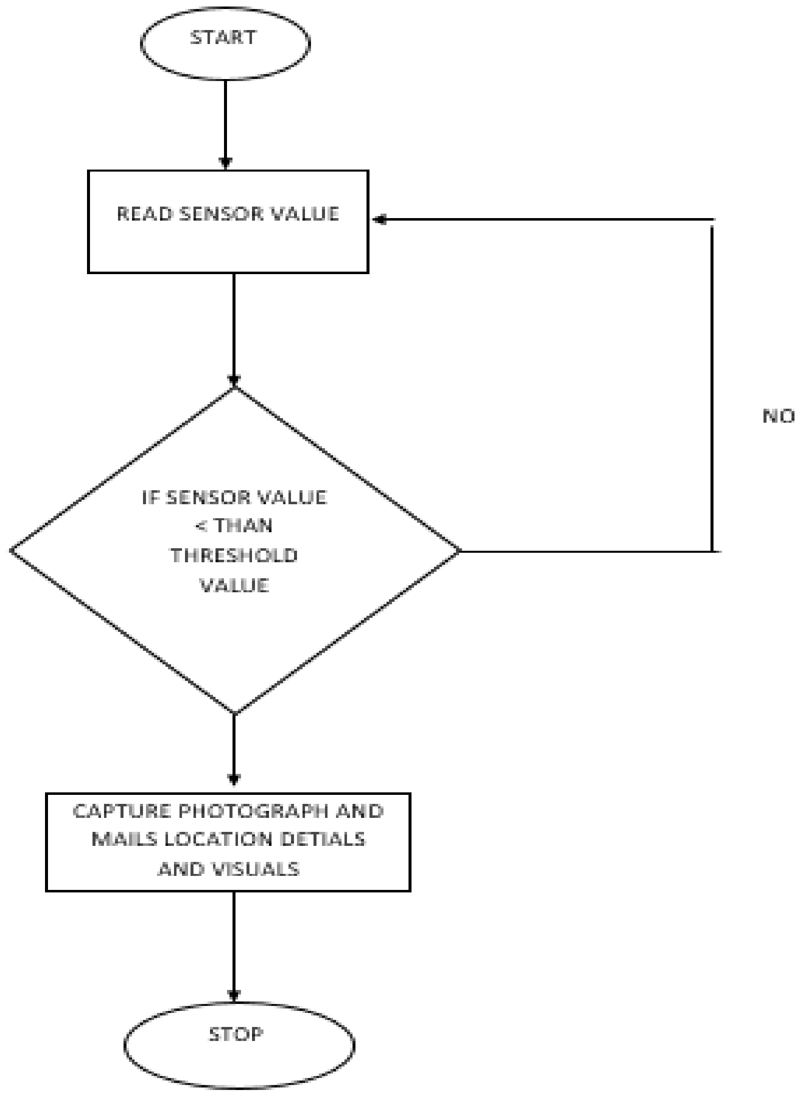
2.Checks HC-SR04 value

3.If the Ultrasonic Sensor value changes from the threshold value, then the picture of accident area is send to the desired mail address.

4.The location is send to the destination.

5. Repeat step 2

5.2 FLOWCHART



CHAPTER 6

WORK DONE

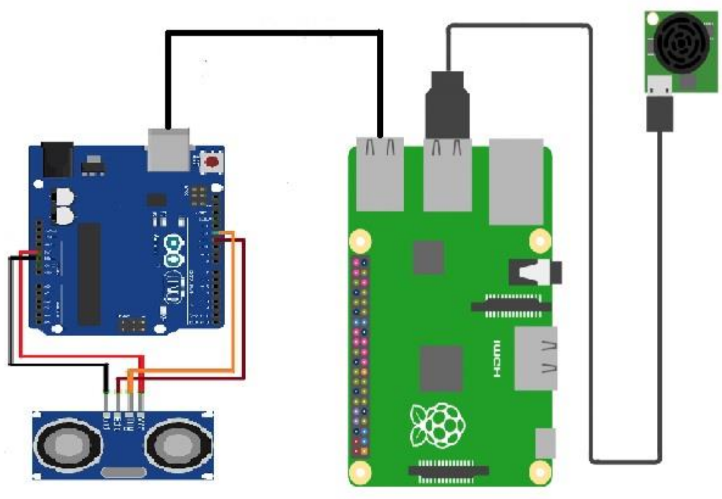


Fig 19: Connection diagram of accident alert and detection system

CHAPTER 7

RESULT AND DISCUSSION

An accident alert and detection system using Raspberry pi was designed and implemented. Ultrasonic Sensor is used to calculate the distance between vehicles.Ultrasonic sensor calculated the distance between the vehicles and captured the photograph of accident caused vehicle. This photograph is send to the destination. The location of the accident spot is send to the nearby hospital through the mobile application, share GPS. The outputs of our project is given below:

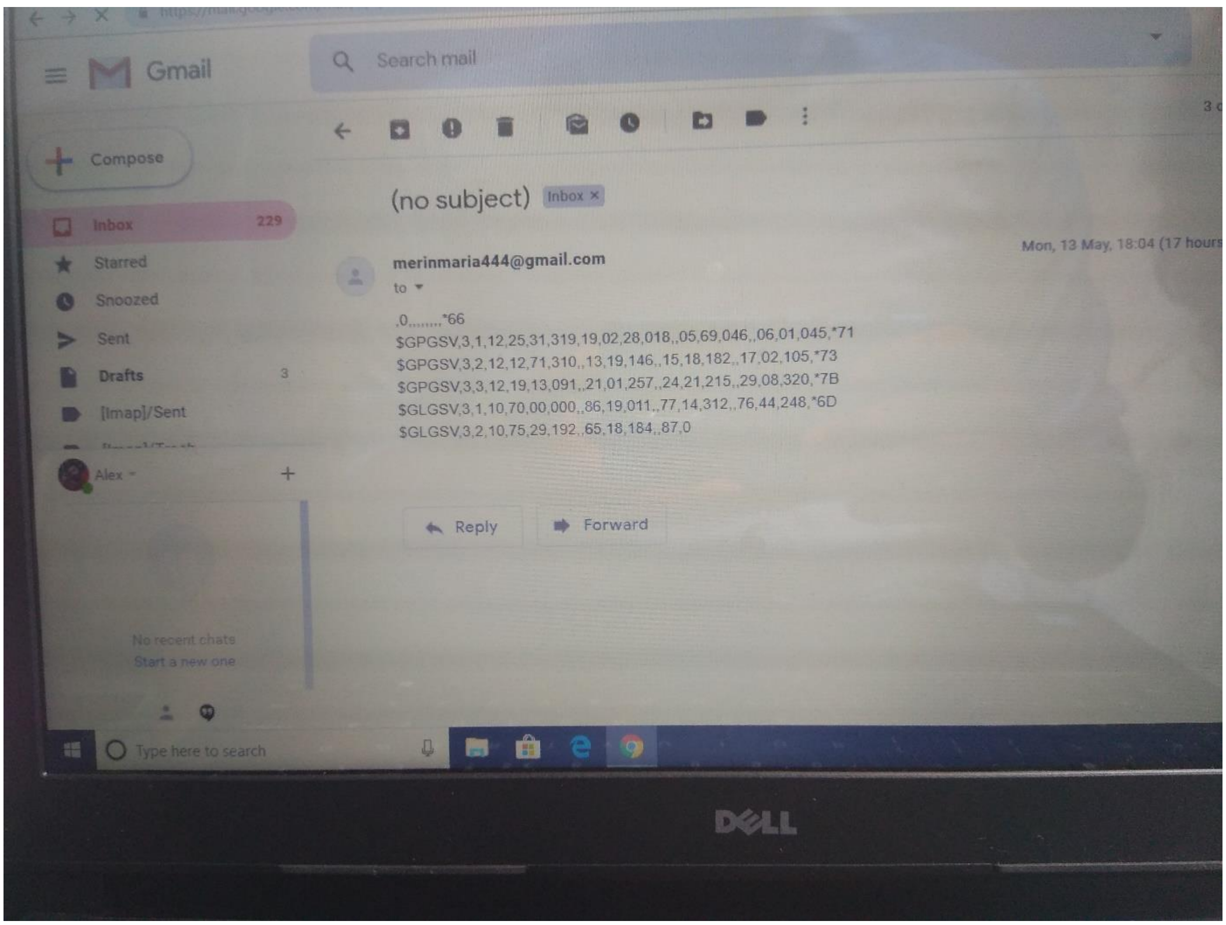


Fig 20: Output 1

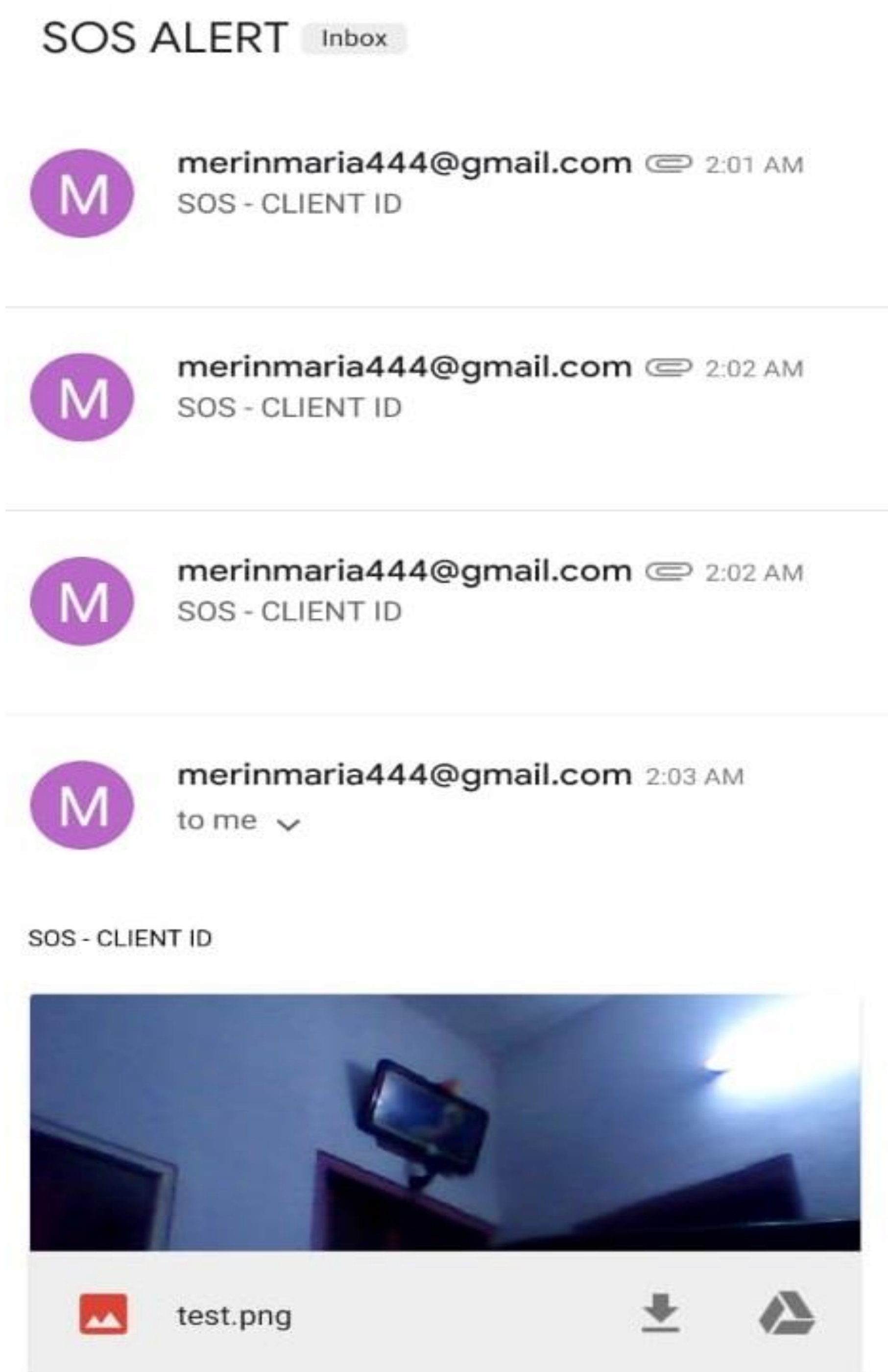


Fig 20: Output 2

ADVANTAGES

- This system is an immediate aid system.
- Alert messages are sent to the nearby hospitals and police stations.
- It is an affordable system.
- Can be used in any kind of vehicle.
- The alert message regarding the accident is automatically sent,
- This system can be used for a social cause.
- It does not need any operation manually.

CHAPTER 8

CONCLUSION

The main motto of our system is to reduce the time delay in rescue operation. Whenever accident occurs the nearby hospital authorities are alerted as soon as possible. Our system is much more useful to the rural places. The vehicle tracking and accident alert system plays a vital role in this century. By overcoming various issues and by the proper implementation of our system around 2 lakhs of lives can save per year. This project aims at an automatic system that can be implemented in any kind of vehicles which helps the victims of an accident to get them to the hospital without any time delay. This system comprises of Raspberry Pi module, ultrasonic sensor, GPS, Raspbian camera etc. within the system makes it an accurate product that functions well at the peak time. This comes really relevant mainly at night and desolate places where people may not be available for help. As it is an automated system, there is no need of any action from the passengers of the vehicle to work.

This prototype would help to accelerate the response of the hospital authorities thereby reducing the number of deaths due to the road accidents. The mobile application Share GPS gives the location details continuously. The ultrasonic sensor calculates the distance between the vehicles. It is interfaced with Arduino Uno. If the calculated distance is less than the threshold value and if accident happens, then the Arduino gives command to the Raspberry pi. Raspberry pi initializes the camera module. the camera takes the photograph of the accident caused vehicle. The installation of camera in the vehicle makes it more advanced that it provides the current pictures from the spot after the accident has happened. This helps the authorities to know about situation very early even before they reach at the spot. They can send this pictures and location details to the nearest hospital on the way to the spot. Cost effectiveness, high speed response and accuracy etc. are its main advantages. It can be placed anywhere in the vehicle, but it is preferred to keep the same under the steering since most of the crashes are happened on the front side. This area is very wide and a lot of future expansion by the addition latest sensors can be done. So, the project can considered as an all-time need.

REFERENCES

[1] Kajal Nandaniya, Viraj Choksi, Ashish Patel, M.B Potdar, "Automatic Accident Alert and Safety System using Embedded GSM Interface" in International Journal of Computer Application volume 85, January 2014.

[2] Anju M. Vasdewani, "Microcontroller 8051 Based Accident Alert System Using MEMS Accelerometer, GPS and GSM Technology" in International Journal of Engineering Trends and Technology, Volume 18 Number 8, December 2014.

[3] Badi Alekhya, S.Aarthi, V.Glory, "Accident Detection with Location and Victim Information Transmission using Raspberry Pi" in International Journal of Advanced Research in Basic Engineering Sciences and Technology, volume 3, special issue 34, March 2017.

[4] D.Santhoshi Rani, K.Radhika Reddy, "Raspberry Pi Based Vehicle Tracking and Security System for Real Time Applications" in International Journal of Computer Science and Mobile Computing, volume 5, Issue 27, July 2016.

[5] Saad Masood Butt, "Accident Detection System Using Intelligent Algorithm for VANET" in Journal of Information Engineering and Applications, Volume 6, number 5, 2016.

[6] Suryakala N, Roopa G , Swaroopa K , Yogitha M P, "IOT Based Vehicle Tracking and Accident Detection System" in International Journal of Innovative Research in Science, Engineering and Technology, Volume 7, Issue 6, May 2018.

[7] C. Prabha, R. Sunitha, R. Anitha, "Automatic Vehicle Accident Detection and Messaging System Using GSM and GPS modem" in International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Volume 3, Issue 7, July 2014.

APPENDICES

```
import socket
import sys
import random
# Create a TCP/IP socket
import cv2
import time
import smtplib
import os
import serial

import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(25, GPIO.IN)
GPIO.setup(24, GPIO.OUT)
ser = serial.Serial('/dev/ttyACM0')
#ser = serial.Serial('/dev/ttyACM0')
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

server_address = ('192.168.225.41', 8081)
camera_capture = cv2.VideoCapture(0)

print('starting up on {} port {}'.format(*server_address))
sock.bind(server_address)
sock.listen(1)
count = 91
```

```
while True:
    # Wait for a connection
    print('waiting for a connection')
    connection, client_address = sock.accept()
    try:
        print('connection from', client_address)

        while True:
            count = count+1
            line = ser.readline()
            line = line.decode('ascii')
            print (line)
            if '0' in line:
                data = connection.recv(300)
                print(data)
                GPIO.output(24, 1)
                print('received {!r}'.format(data))
                data = data.decode('utf-8')
                data = str (data)
                print (data)
                server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
                server.login("merinmaria444@gmail.com", "alexsaji111")
                server.sendmail(
                    "merinmaria444@gmail.com",
                    "alexsaji111@gmail.com",
                    data)
                success, im = camera_capture.read()
```

```
cv2.imwrite('test.png',im)
                              os.system('python3 sosemail1.py')
                              os.system ('python3 sossms.py')
                              GPIO.output(24, 0)



                              time.sleep(5)


finally:
          print('closing socket')
          sock.close()
```