Automotive Control System

TEAM MEMBERS

Mennatallah Mohamed
Sondos Ghieth
Dina Gamal EL Din
Dalia Abdallah
Nariman Alaa
Verena Emad

FEBRAUARY 2023

Under Supervision Of:
Information Technology Institute (ITI)

Table Of Contents

CHAPTER

1

INTRODUCTION

1. Introduction

Today in an exceedingly fast paced world Speed is one of the most important and basic risk factors in driving, it not only affects the severity of a crash, but also increases risk of being involved in a crash, so we lost our valuable life by making small mistakes while driving. With the increase in number of vehicles, safety concern also increases for both people inside the vehicle and outside the vehicle. Despite many efforts taken by different governmental and non-governmental organizations all around the world by various programs to aware against careless driving, yet accidents are taking place every now and then. World Health Organization (WHO) had conducted various surveys for road fatalities based on number of vehicles, inhabitants and total fatalities globally due to road accidents.

Reducing and mitigating the number of accidents is from the main objectives to traffic authorities, Research groups from the automotive and transportation industry. There are many solutions given to the road authorities and one of such solutions is advanced driver assistance support (ADAS) which are audio systems, visual cues produced by the vehicle itself to warn the driver, The role of ADAS is to prevent deaths and injuries by reducing the number of car accidents and the serious impact of those that cannot be avoided. These systems are fairly commercially available in Today's cars, future trends indicate that increased safety will be achieved through automatic driving controls and an increasing number of sensors in each of the road infrastructure and the car itself.

However, If the accident already accrued many lives could have been saved if the emergency service could get the crash information in time. Many accidents could be prevented only if the emergency services could be provided at the place of accident at the proper time. As such, efficient automatic accident detection with an automatic notification to the emergency service with the accident location is a prime need to save the precious human life.

So, we aim to implement accident prevention, detection and reporting system that helps in reducing the road accidents and preserving driver's life.

1.1. Project Objective

The objective of Automotive Control system is to avoid such kind of accidents by alerting the drivers with voice and visual assistance system and a way to control vehicle's speed according to the road speed limit. In addition to accident detect and alert system to alert rescue team in time when the accident occurs.

1.2. Project Description

Automotive control system is a system used to reduce the number of accidents in highways and to alert the rescue team in time when an accident occur, so this system is consists of 4 features:

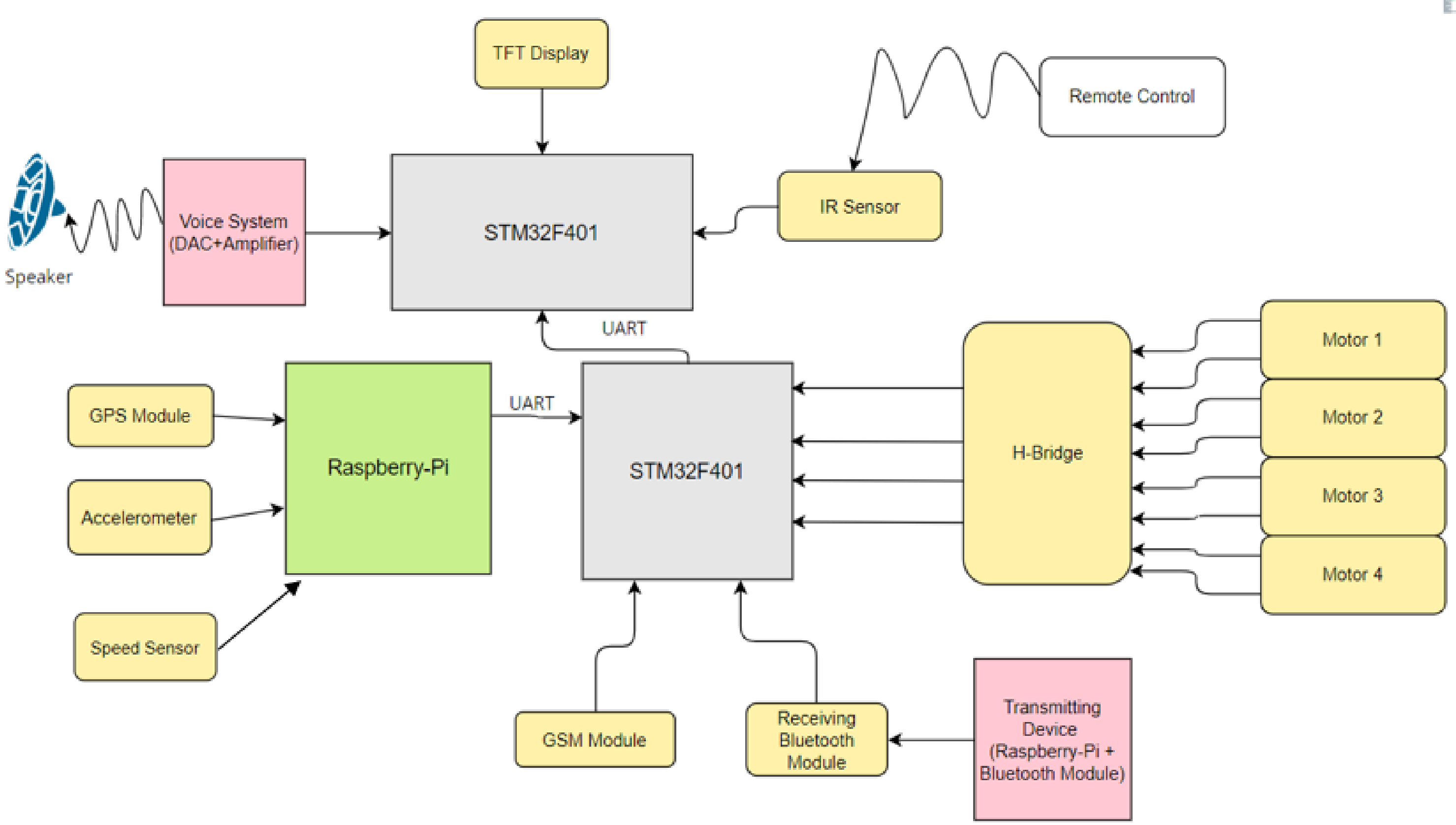
- The first feature is the **Door lock and Start system**, this feature is for the system and the vehicle's security that the system will not be enabled until it detects a specific signal from the user's remote control, this transmitted signal from the remote control will be received by the system from IR sensor, without this signal or if the signal was wrong all system features will be disabled and the vehicle will not be able to run.

- The second feature is **Vehicle's Speed Control**, this feature for reducing the vehicle's speed according to the road's speed limit by Bluetooth module transmitter and receiver, The Transmitter is mounted on the Road Side and it is responsible to transmit the maximum allowable speed limit of that road to receiver, the receiver will be on vehicle. The system will manage the speed of the vehicle's whenever it enters the Bluetooth module range.

- The third feature is **Accident detection and Alert system**, this feature when the accident occurs it uses various components and alerts the Rescue team for help. An efficient automatic accident detection with an automatic notification to the emergency service with the accident location is a prime need to save the precious human life. It detects the occurrence of the accidents by using accelerometer and then reads the exact latitude and longitude of the vehicle involved in the accident by GPS module and sends this information to nearest emergency service provider by GSM module.

- The fourth feature is **Driver voice and visual assistant**, this feature will show that the system received the right signal by the IR sensor by showing a welcome screen on the TFT display and a "welcome" voice will output from the speakers using the voice system, also when the driver drive in high speed and the system detect from the Bluetooth module that he exceeds the speed limit, the system will display a warning message that he exceeds the speed limit on the TFT display and a "warning you exceed the limit" voice will output from the speaker using the voice system.

1.3. Project Block Diagram



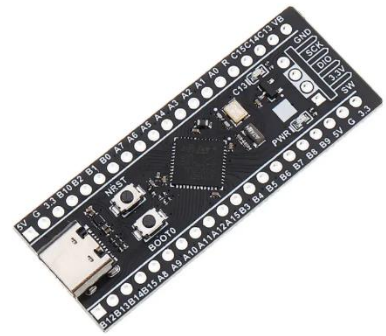
CHAPTER

2

Main Implementation Components

2. Main Implemetion Components

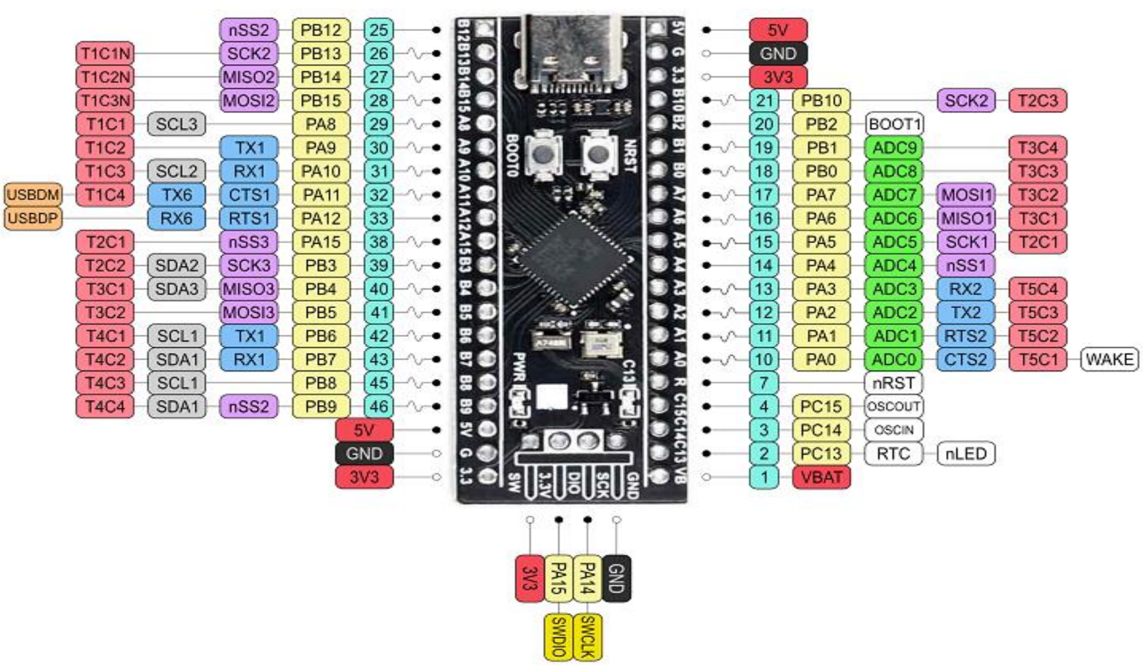
2.1. STM32F401



The STM32F401xB/STM32F401xC devices are based on the high-performance ArmCortex®-M4 32-bit RISC core operating at a frequency of up to 84 MHz. The Cortex®-M4 core features a Floating-point unit (FPU) single precision which supports all Arm single-precision data-processing instructions and data types. It also implements a full set of DSP instructions and a memory protection unit (MPU) which enhances application security.

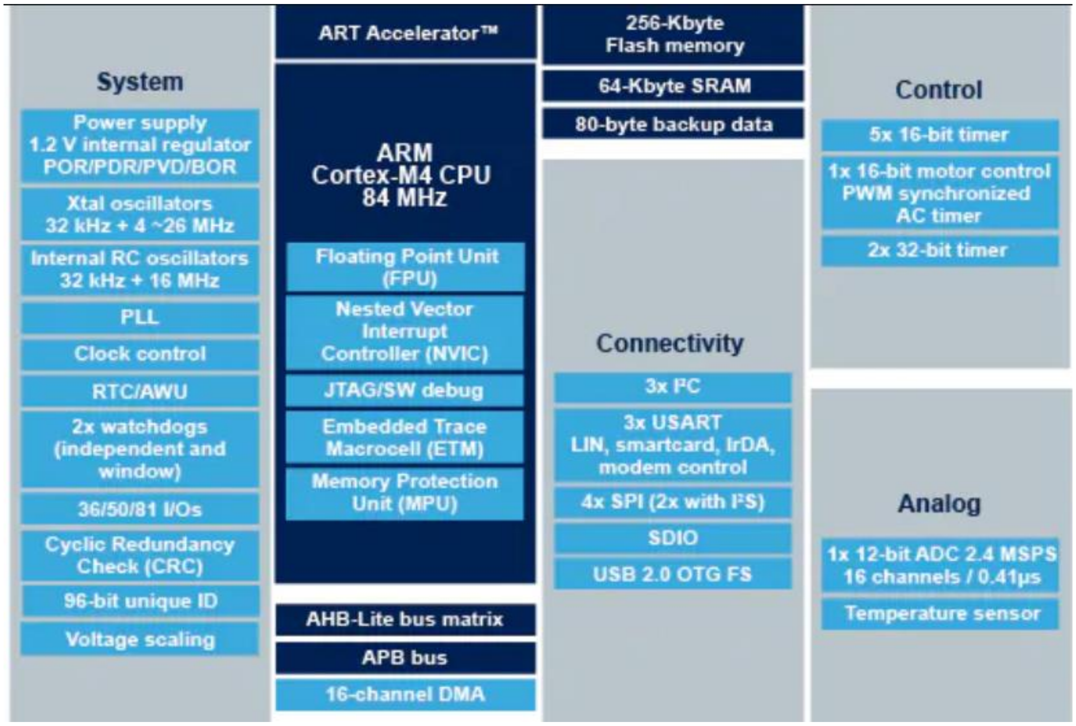
The STM32F401xB/STM32F401xC incorporate high-speed embedded memories (up to 256 Kbytes of Flash memory, up to 64 Kbytes of SRAM), and an extensive range of enhanced I/Os and peripherals connected to two APB buses, two AHB buses and a 32-bit multi-AHB bus matrix.

All devices offer one 12-bit ADC, a low-power RTC, six general-purpose 16-bit timers including one PWM timer for motor control, two general-purpose 32-bit timers. They also feature standard and advanced communication interfaces.



Features:

- Dynamic efficiency line with BAM (batch acquisition mode)
  - 1.7 V to 3.6 V power supply
  - -40 °C to 85/105/125 °C temperature range
- Core: Arm® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait state execution from Flash memory, frequency up to 84 MHz, memory protection unit, 105 DMIPS/1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions.
- Memories
  - Up to 256 Kbytes of Flash memory
  - 512 bytes of OTP memory
  - Up to 64 Kbytes of SRAM
- Clock, reset and supply management
  - 1.7 V to 3.6 V application supply and I/Os
  - POR, PDR, PVD and BOR
  - 4-to-26 MHz crystal oscillator
  - Internal 16 MHz factory-trimmed RC
  - 32 kHz oscillator for RTC with calibration
  - Internal 32 kHz RC with calibration
- 1×12-bit, 2.4 MSPS A/D converter: up to 16 channels
- General-purpose DMA: 16-stream DMA controllers with FIFOs and burst support
- Up to 11 timers: up to six 16-bit, two 32-bit timers up to 84 MHz, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input, two watchdog timers (independent and window) and a SysTick timer
- Up to 81 I/O ports with interrupt capability
  - All IO ports 5 V tolerant
  - Up to 78 fast I/Os up to 42 MHz
- Up to 11 communication interfaces
  - Up to 3 × I^2C interfaces (1Mbit/s, SMBus/PMBus)
  - Up to 3 USARTs (2 x 10.5 Mbit/s, 1 x 5.25 Mbit/s), ISO 7816 interface, LIN, IrDA, modem control)
  - Up to 4 SPIs (up to 42 Mbits/s at f_{CPU}= 84 MHz), SPI2 and SPI3 with muxed full-duplex I^2S to achieve audio class accuracy via internal audio PLL or external clock.



2.2. Raspberry-Pi device

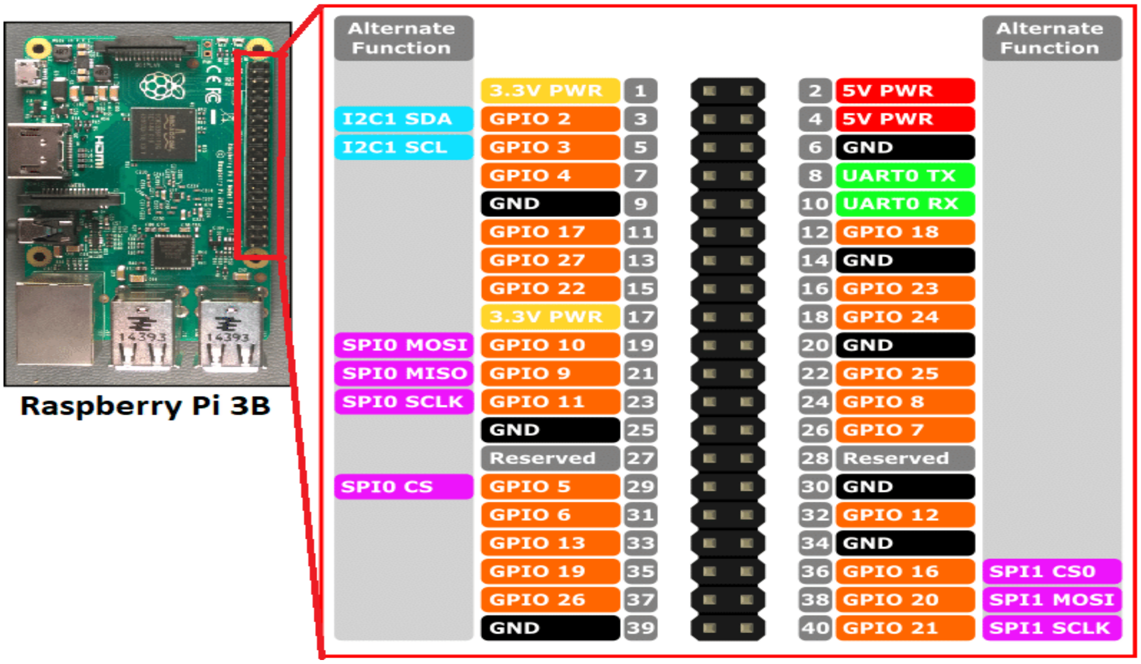
Raspberry-Pi is a Low-cost high performance, credit-card sized computer which has all its components, ports, and features out on display, it plugs into a computer monitor or TV and uses a standard keyboard and mouse. It is a capable little device that makes programming and computing easier. It can do everything you would expect, also, the Raspberry-Pi has the ability to interact with the outside world and has been used in a wide array of digital maker projects.

Raspberry-Pi OS:

Raspberry-pi is a pocket PC size computer that enable to use an external operating system. The Raspberry Pi does not have a pre-installed operating system in which the user can start programming on, it needs a micro-SD card inserted in it where the operating system is installed on it. The Raspberry-Pi uses a lightweight operating system known as the Raspbian OS, it is a Linux operating system that comes with different applications and integrated development environment (IDE) pre-installed on it for programming which was used to write all the algorithms and hardware programs to control the pins.

Features:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU.
- 1GB RAM.
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board.
- 100 Base Ethernet.
- 40-pin extended GPIO.
- 4 USB 2 ports.
- 4 Pole stereo output and composite video port.
- Full size HDMI.
- CSI camera port for connecting a Raspberry Pi camera.
- DSI display port for connecting a Raspberry Pi touchscreen display.
- Micro SD port for loading your operating system and storing data.
- Upgraded switched Micro USB power source up to 2.5A.



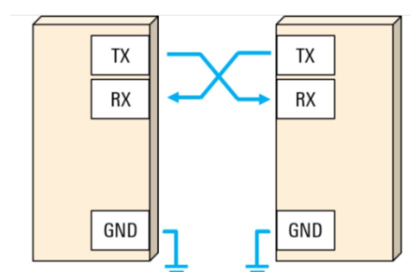
2.3. Main Components Communication

<u>Comunication between Raspberry-Pi device and STM32 Microcontroller:</u>

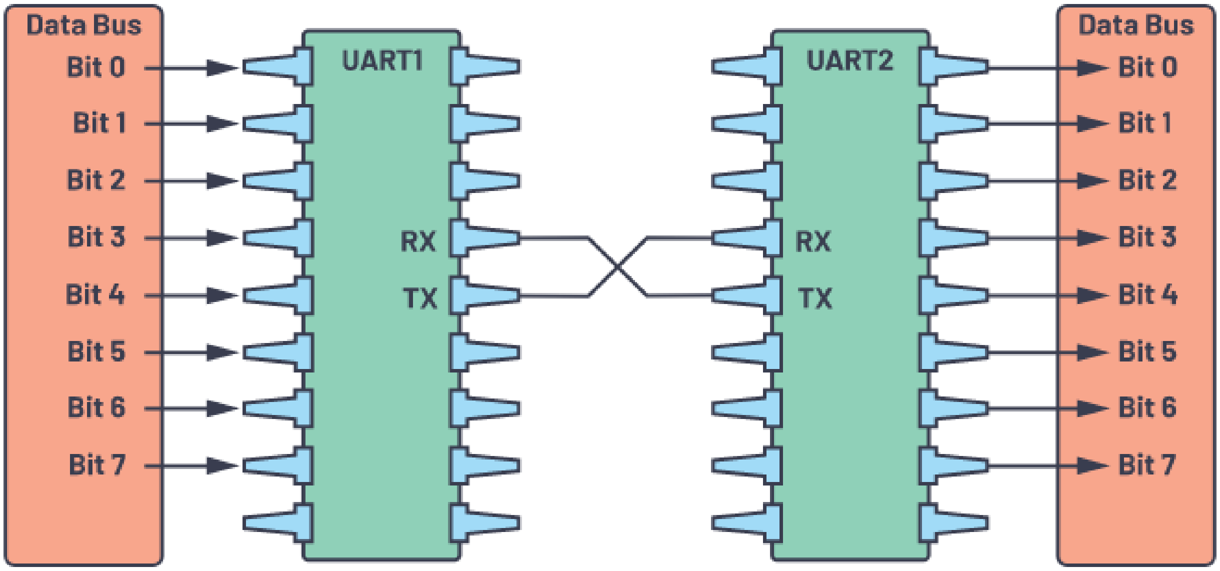
The used communication protocol in our system is UART (Universal Asynchronous Receiver-Transmitter), it is of the most used device-to-device communication protocols.

By definition, UART is a hardware communication protocol that uses asynchronous serial communication with configurable speed. Asynchronous means there is no clock signal to synchronize the output bits from the transmitting device going to the receiving end.

UART is very simple and only uses two wires between transmitter (Tx) and receiver (Rx) to transmit and receive in both directions. Both ends also have a ground connection. The used communication in UART is full-duplex (both sides can transmit simultaneously).



The main purpose of a transmitter and receiver line for each device is to transmit and receive serial data intended for serial communication.
The transmitting UART is connected to a controlling data bus that sends data in a parallel form. From this, the data will now be transmitted on the transmission line (wire) serially, bit by bit, to the receiving UART. This, in turn, will convert the serial data into parallel for the receiving device. The UART lines serve as the communication medium to transmit and receive one data to another.



Timing and synchronization of UART protocols

For and most serial communications, you must ensure that the baud rate needs to be set the same on both the transmitting and receiving device. The baud rate is the rate at which information is transferred to a communication channel. In the serial port context, the set baud rate will serve as the maximum number of bits per second to be transferred.

In UART the transmitter and receiver do not share a common clock signal. Although this greatly simplifies the protocol, it does place certain requirements on the transmitter and receiver. Since they do not share a clock, both ends must transmit at the same, pre-arranged speed in order to have the same bit timing. The most common UART baud rates in use today are 4800, 9600, 19.2K, 57.6K, and 115.2K.

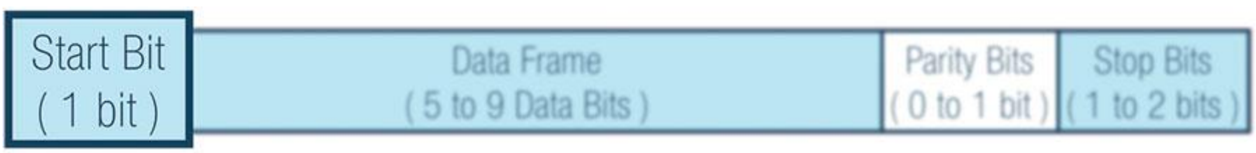
Data Transmission

In addition to having the same baud rate, both sides of a UART connection also must use the same frame structure and parameters.
The piece that connects the transmitter and receiver includes the creation of serial frame and controls those physical hardware lines.
A packet consists of a start bit, data frame, a parity bit, and stop bits.

Full frame

- **Start Bit**

The UART data transmission line is normally held at a high voltage level (idle state) when it's not transmitting data. To start the transfer of data, the transmitting UART pulls the transmission line from high to low for one (1) clock cycle. When the receiving UART detects the high to low voltage transition, it begins reading the bits in the data



frame at the frequency of the baud rate.

- **Data Frame**

The data frame contains the actual data being transferred. It can be five (5) bits up to eight (8) bits long if a parity bit is used. If no parity bit is used, the data frame can be

nine (9) bits long. In most cases, the data is sent with the least significant bit first.

- **Parity Bit**

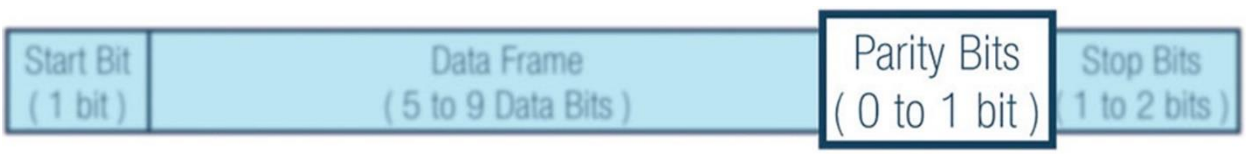
Parity describes the evenness or oddness of a number. The parity bit is a way for the receiving UART to tell if any data has changed during transmission.

After the receiving UART reads the data frame, it counts the number of bits with a value of 1 and checks if the total is an even or odd number.

If the parity bit is a 0 (even parity): the 1 or logic-high bit in the data frame should total to an even number.

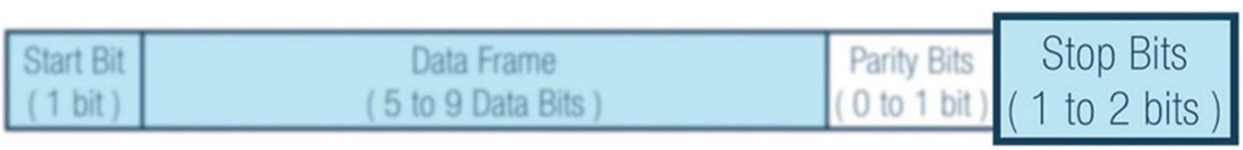
If the parity bit is a 1 (odd parity): the 1 bit or logic highs in the data frame should total to an odd number.

When the parity bit matches the data, the UART knows that the transmission was free of errors. But if the parity bit is a 0, and the total is odd, or the parity bit is a 1, and the total is even, the UART knows that bits in the data frame have changed**.**



- **Stop Bits**

To signal the end of the data packet, the sending UART drives the data transmission line from a low voltage to a high voltage for one (1) to two (2) bit(s) duration.



Steps of UART Transmission

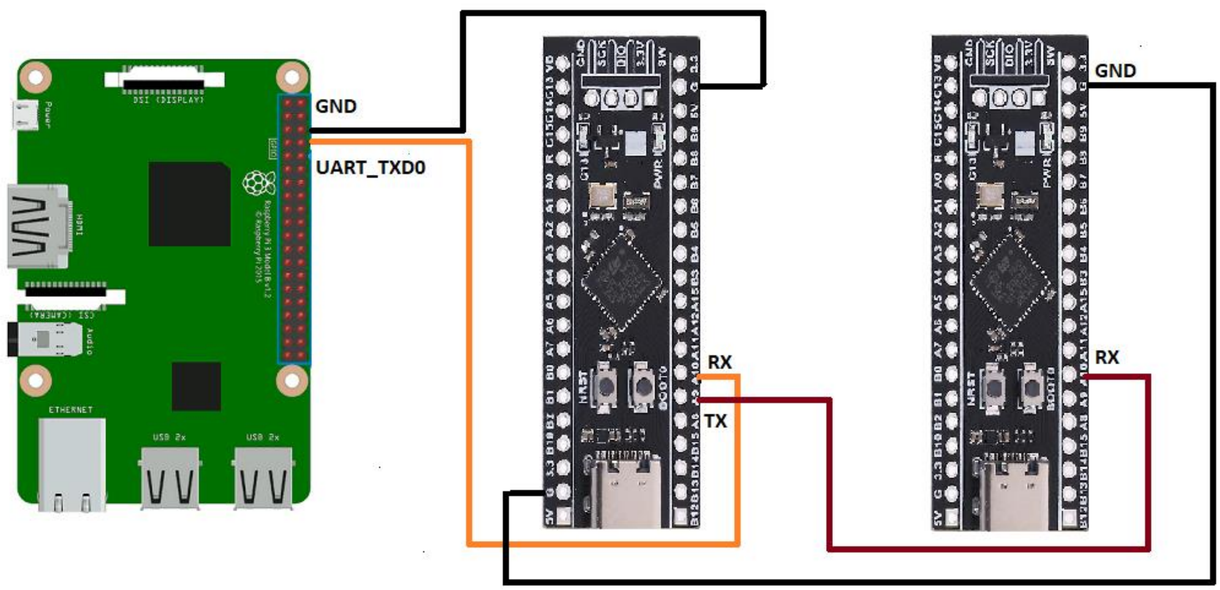
First: The transmitting UART receives data in parallel from the data bus.

Second: The transmitting UART adds the start bit, parity bit, and the stop bit(s) to the data frame.

Third: The entire packet is sent serially starting from start bit to stop bit from the transmitting UART to the receiving UART. The receiving UART samples the data line at the preconfigured baud rate.

Fourth: The receiving UART discards the start bit, parity bit, and stop bit from the data frame.

System connections using UART



Firstly, we connected the Raspberry pi device to the stm32 microcontroller to send the data and status of the sensors interfaced with Raspberry-Pi, so depend on this data the microcontroller will take specific actions.
Secondly, we connect the first stm32 microcontroller to the second stm32 microcontroller to make it know what to display in its voice and visual system.

Implementation

- Connect UART_TXD0 of Raspberry pi to Rx pin on stm32 microcontroller
- Connect the Tx pin of first stm32 microcontroller to Rx pin of the second
- Enable UART on raspberry-pi by using command: enable_UART=1
- Intialize and configure the frame in all devices as:
    a. Start bit
    b. 8-bits of data
    c. No parity bit used
    d. One Stop bit

CHAPTER

3

HARDWARE COMPONENTS

3. Hardware components

3.1. Global Positioning System (GPS)

GPS is a satellite-based radio navigation system owned by the United States government and operated by the United States Space Force. It is one of the global navigation satellite systems (GNSS) that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.

There are several different methods for obtaining a position using GPS. The method used depends on the accuracy required by the user and the type of GPS receiver available. The techniques can be broken down into three basic classes:

Autonomous Navigation:

Using a single stand-alone receiver. Position Accuracy is better than 100m for civilian users and about 20m for military users.

Differential Phase position:

Gives an accuracy of 0.5-20mm. Used for many surveying tasks, machine control etc.

Differentially corrected positioning:

More commonly known as DGPS, this gives an accuracy of between 0.5-5m. Used for inshore marine navigation, GIS data acquisition, precision farming.

Autonomous Navigation…

This is the simplest technique employed by GPS receivers to instantaneously give a position and height and/ or accurate time to a user. The accuracy obtained is better than 100m (usually around the 30-50m mark) for civilian users and 5-15m for military users. Receivers used for this type of operation are typically small, highly portable handheld units with a low cost.

The GPS concept is based on time and the known position of GPS specialized satellites. The satellites carry very stable atomic clocks that are synchronized with one another and with the ground clocks. Any drift from true time maintained on the ground is corrected daily. In the same manner, the satellite locations are known with great precision. GPS receivers have clocks as well, but they are less stable and less precise.

Each GPS satellite continuously transmits a radio signal containing the current time and data about its position. Since the speed of radio waves is constant and independent of the satellite speed, the time delay between when the satellite transmits a signal and the receiver receives it is proportional to the distance from the satellite to the receiver. As presented in one of Isaac Newton's laws of motion is: Distance = Velocity x Time

A GPS receiver monitors multiple satellites and solves equations to determine the precise position of the receiver and its deviation from true time. At a minimum, four satellites must be in view of the receiver for it to compute four unknown quantities (three position coordinates (x, y, z) and clock deviation from satellite time).

The receiver location is expressed in a specific coordinate system, such as latitude and longitude using the WGS 84(World Geodetic System) geodetic datum or a country specific system. By getting the value of latitude, longitude and altitude we can define any point on the earth with these coordinates.

NEO-6 u-blox 6 GPS Modules...

The NEO-6 module series is a family of stand-alone GPS receivers featuring the high-performance u-blox 6 positioning engines. These flexible and cost-effective receivers offer numerous connectivity options in a miniature 16 x 12.2 x 2.4 mm package. It can track up to 22 satellites over 50 channels and achieve the industry's highest level of tracking sensitivity it can perform 5 location updates in a second with 2.5m horizontal position accuracy. The U-blox 6 positioning engine also has a Time-To First-Fix (TTFF) of less than 1 second.

- **GPS performance**

This table illustrates performance measures for all The NEO-6 module series but will focus on NEO-6 M which will be used in our project.

Parameter		**Specification**	
Time-To-First-Fix1	NEO-6G/Q/T	NEO-6M/V	NEO-6P
Cold Start2	26 s	27 s	32 s
Warm Start2	26 s	27 s	32 s
Hot Start2	1 s	1 s	1 s
Sensitivity	NEO-6G/Q/T	NEO-6M/V	NEO-6P
Tracking & Navigation	-162 dBm	-161 dBm	-160 dBm
Maximum Navigation update rate			
NEO-6G/Q/M/T		NEO-6P/V	
5Hz		1 Hz	
Horizontal position accuracy			
GPS		2.5 m	
Velocity accuracy		0.1m/s	
Operational Limits			
Dynamics		4g	
Altitude		50,000 m	
Velocity		500 m/s	

- **Power Management**

u-blox receivers support different power modes. These modes represent strategies of how to control the acquisition and tracking engines in order to achieve either the best possible performance or good performance with reduced power consumption**.**

- **Maximum Performance Mode**

During a Cold start, a receiver in Maximum Performance Mode continuously deploys the acquisition engine to search for all satellites. Once the receiver has a position fix (or if pre-positioning information is available), the acquisition engine continues to be used to search for all visible satellites that are not being tracked.

- **Eco Mode**

During a Cold start, a receiver in Eco Mode works exactly as in Maximum Performance Mode. Once a position can be calculated and a sufficient number of satellites are being tracked, the acquisition engine is powered off resulting in significant power savings. The tracking engine continuously tracks acquired satellites and acquires other available or emerging satellites.

- **Power Save Mode**

Power Save Mode (PSM) allows a reduction in system power consumption by selectively switching parts of the receiver on and off.

- **Protocols and interfaces**

Protocol	Type
NMEA	Input/output, ASCII, 0183, 2.3 (compatible to 3.0)
UBX	Input/output, binary, u-blox proprietary

- NEO 6 M chip supports the listed protocols and interfaces with UART, USB and SPI but in our project, we will depend on commercialized NEO-6 M module shown in the figure which supports an interface with UART only, NEO-6M modules include one configurable UART interface for serial communication.

NEO-6M specifications:

- Receiver Type= 50 channels, GPS L1(1575.42Mhz)
- Horizontal Position Accuracy= **2.5m**
- Navigation Update Rate= 1HZ (5Hz maximum)
- Capture Time Cool start= 27sHot start: 1s
- Navigation Sensitivity = -161dBm
- Communication Protocol = NMEA, UBX Binary, RTCM
- Serial Baud Rate = 4800-230400 (default 9600)
- Operating Temperature = -40°C ~ 85°C
- Operating Voltage = 2.7V ~ 3.6V
- Operating Current = 45mA
- TXD/RXD Impedance= 510Ω

➢ **Default settings**

Interface	**Settings**
Serial Port 1 Output	9600 Baud, 8 bits, no parity bit, 1 stop bit Configured to transmit both NMEA and UBX protocols, but only following NMEA and no UBX messages have been activated at start-up: **GGA, GLL, GSA, GSV, RMC, VTG, TXT** (In addition to the 6 standard NMEA messages the NEO-6T includes **ZDA**).
Serial Port 1 Input	9600 Baud, 8 bits, no parity bit, 1 stop bit Automatically accepts following protocols without need of explicit configuration: **UBX, NMEA** The GPS receiver supports interleaved UBX and NMEA messages.
Power Mode	Maximum Performance mode

• Once you a buy a module you can find it in of default configurations. The module was on 9600 Baud and supports UART framing of 8 bits, no parity bit, 1 stop bit. Also, it was Configured to transmit both NMEA and UBX protocols.

• Design options:
Depend on NMEA protocol frames.
Depend on UBX protocol frames.

• Our design option was to depend on NMEA protocol frames for the following reasons:
- NMEA frames are sent continuously with no need to request data from the receiver each time you need GPS frame.

- UBX protocol is bidirectional communication protocols, hence we need to send a request for the module and wait for it to respond with the needed frame which consumes double the time taken in NMEA frames reading.

• The GPS module actually receives 6 types of NMEA frames which is GGA, GLL, GSA, GSV, RMC, VTG.

```
$GPGSA,A,3,07,05,30,20,09,08,17,,,,,,2.68,1.45,2.25*0F
$GPGSV,2,1,08,02,00,247,,05,31,305,27,07,45,038,37,08,13,064,31*76
$GPGSV,2,2,08,09,29,111,33,17,05,163,25,20,46,266,26,30,62,352,23*76
$GPGLL,3000.33660,N,03112.36919,E,222436.00,A,A*6C
$GPRMC,222437.00,A,3000.33671,N,03112.36938,E,0.222,,220622,,,A*73
$GPVTG,,T,,M,0.222,N,0.412,K,A*26
$GPGGA,222437.00,3000.33671,N,03112.36938,E,1,07,1.45,63.7,M,15.3,M,,*69
$GPGSA,A,3,07,05,30,20,09,08,17,,,,,,2.68,1.45,2.25*0F
$GPGSV,2,1,08,02,00,247,,05,31,305,27,07,45,038,37,08,13,064,31*76
$GPGSV,2,2,08,09,29,111,32,17,05,163,25,20,46,266,27,30,62,352,23*76
$GPGLL,3000.33671,N,03112.36938,E,222437.00,A,A*6E
```

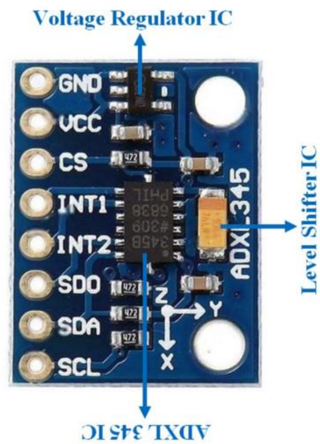
RMC (Recommended Minimum) is an NMEA frame which consists of 70 to 75 characters contains the needed data and other data which is unneeded**.**

```
$GPRMC,222437.00,A,3000.33671,N,03112.36938,E,0.222,,220622,,,A*73
```

Latitude: 3000.33671 Longitude: 03112.36938

3.2. Accelerometer

The ADXL345 is a small, low power, complete 3-axis MEMS accelerometer modules with both I2C and SPI interfaces. The ADXL345 board feature on-board 3.3V voltage regulator and level shifter, which makes it simple to interface with 5V microcontrollers, it also contains resistors, and capacitors in an integrated circuit.



Pin Configuration:

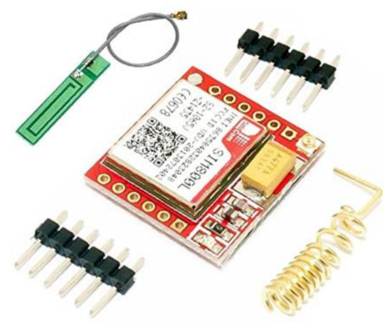
Pin Name	Description
GND	Ground Pin
VCC	Power Supply pin (3V to 6V)
CS	Chip Select Pin
INT1	Interrupt 1 Output
INT2	Interrupt 2 Output
SDO	Serial Data Output
SDA	Serial Data Input & Output
SCL	Serial Communication Clock

Features & Specifications:

- 3V-6V DC Supply Voltage
- On-board LDO Voltage regulator
- Built-in Voltage level convertor (MOSFET based)
- It can be interface with 3.3V or 5V Microcontroller.
- Ultra-Low Power: 40uA in measurement mode, 0.1uA in standby@ 2.5V
- Tap/Double Tap Detection
- Free-Fall Detection
- SPI and I2C interfaces
- Measuring Range: ±16g
- Measuring Values (-16g to +16g):
- X: -235 to +270
- Y: -240 to +260
- Z: -240 to +270

3.3. GSM

The SIM800L is a miniature GSM modem that can be used in a variety of IoT projects. You can use this module to do almost anything a normal cell phone can do, such as sending SMS messages, making phone calls, connecting to the Internet via GPRS. It interfaces with the microcontroller by UART and with the mobile by AT commands.

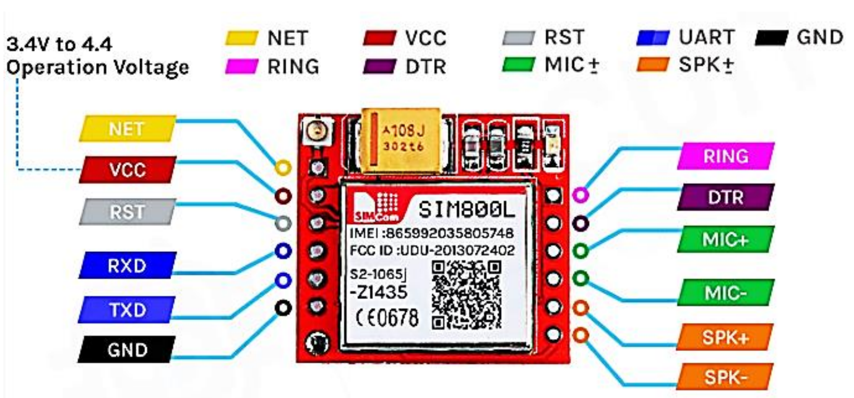


Features

- Quad-band 850/900/1800/1900MHz
- Send and receive SMS messages
- AT command interface with "auto baud" detection
- Input Voltage: DC 3.7-4.2V.

Pin Configuration

Pin Name	Description
NET	External antenna attachment pin
VCC	Power supply pin, 3.4V to 4.4V input
RST	Reset pin, pull low for 100ms to perform hard reset
RXD	Serial data input
TXD	Serial data output
GND	Module ground reference
SPK	Speaker differential output
MIC	Microphone differential input
DTR	Serial data terminal ready pin, pull high to enable sleep mode
RING	Interrupt output, active low



3.4. TFT Display

TFT stands for thin-film transistor, which means that each pixel in the device has a thin-film transistor attached to it. Transistors are activated by electrical currents that make contact with the pixels to produce impeccable image quality on the screen. Here are some important features of TFT displays.

TFT is:
- Excellent Color Display(16BIT RGB 65K color)
- Extended Half-Life.
- TFT displays can have either resistive or capacitive touch panels.
- TFT displays offer exceptional aspect ratio control.
- TFT displays are incredibly versatile**.**

<u>Features</u>

- Size:1.8 inch
- Resolution: 128*160
- Driver IC: ST7735
- Integrated voltage regulator IC, support 5V or 3.3V power supply.
- On-board level conversion scheme, truly compatible with 5V/3.3V IO level, supporting various microcontroller IO connections.
- Integrated SD card expansion circuit.
- Reserve SPI FLASH font circuit to facilitate extended application.
- Backlight: 2 White Led
- Working temperature: -20 ℃ ~ 60 ℃
- Storage temperature: -30 ℃ ~ 70 ℃

Pin Configuration:

Pin Name	Description
VCC	LCD screen power supply (3.3V ~ 5V)
GND	LCD screen power ground
GND	LCD screen power ground
NC	No definition, reserved
NC	No definition, reserved
NC	No definition, reserved
CLK	LCD screen SPI bus clock signal
SDA	LCD screen SPI bus write data signal
RS	LCD screen register / data selection signal, low level: register, high level: data
RST	LCD reset signal, low level reset
CS	LCD chip select signal, low level enable

3.5. IR Sensor

An infrared sensor is an electronic device, that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measure only infrared radiation, rather than emitting it. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. Infrared band of the electromagnet corresponds from 300GHz to 430THz and a wavelength of 980nm. The propagation of light waves in this band can be used for a communication system (for transmission and reception) of data. This communication can be between two portable devices or between a portable device and a fixed device.

IR Sensor pros

- Cannot be detected by human eyes.
- Infrared radiation is not harmful to human beings. Hence infrared communication can be used at any place
- High Speed data Communication: The data rate of Infrared communication is about 1Gbps and can be used for sending information like video signal.
- Long distance = 10 m'.
- Common, inexpensive, and easy to use wireless communication technology.
- IR light is very similar to visible light, except that it has a slightly longer wavelength.
- IR is undetectable to the human eye - perfect for wireless communication.

Main Parts of IR communication systems:

Transmitter (remote-control):

Is an electronic device that radiates or locates infrared radiation to sense some part of its surroundings. They are undetectable to human eyes.

Features

- Input voltage: 5VDC.
- Sensing Range: 5cm.
- Output signal: analog voltage.
- Emitting element: Infrared LED.

Receiver (photodiode):

The part that receives the transmitted signal and indicate if it is a right signal or not.

Features

- Supply Voltage is –0.3-6.0 V.
- Supply Current is 5 mA.
- Output Voltage is –0.3-6.0 V.
- Output Current is 5 mA.

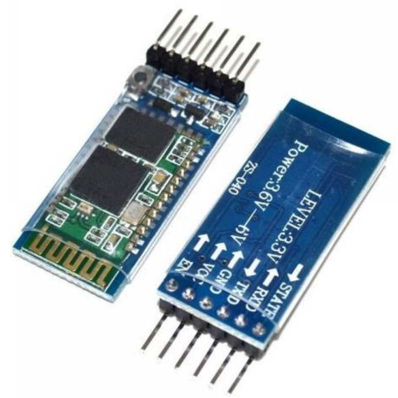
NEC Infrared Transmission Protocol

The NEC IR transmission protocol uses pulse distance encoding of the message bits.

- Logical '0' – a 562.5µs pulse burst followed by a 562.5µs space, with a total transmit time of 1.125ms.

- Logical '1' – a 562.5µs pulse burst followed by a 1.6875ms space, with a total transmit time of 2.25ms.

3.6. Bluetooth Module



HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.
Bluetooth serial modules allow all serial enabled devices to communicate with each other using Bluetooth.

- The HC-05 is very easy to use Bluetooth to serial converter.

- HC-05 connects microcontrollers to other Bluetooth enabled devices. This allows the devices tocommunicate wirelessly with each other.

- HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.

- It has range up to <100m which depends upon transmitter and receiver.

- It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART).

Pin configuration

Pin Name	**Description**
Key/EN	It is used to bring Bluetooth module in AT commands mode. If Key/EN pin is set to high, then this module will work in command mode. Otherwise by default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode.
VCC	Connect 5 V or 3.3 V to this Pin.
GND	Ground Pin of module.
TXD	Transmit Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)
RXD	Receive data serially (received data will be transmitted wirelessly by Bluetooth module).
State	It tells whether module is connected or not.

HC-05 module Information:

- HC-05 has red LED which indicates connection status, whether the Bluetooth is connected or not. Before connecting to HC-05 module this red LED blinks continuously in a periodic manner. When it gets connected to any other Bluetooth device, its blinking slows down to two seconds.

- This module works on 3.3 V. We can connect 5V supply voltage as well since the module has on board 5 to 3.3 V regulator.

- As HC-05 Bluetooth module has 3.3 V level for RX/TX and microcontroller can detect 3.3 V level, so, no need to shift transmit level of HC-05 module. But we need to shift the transmit voltage level from microcontroller to RX of HC-05 module.

Modes of Operation

The HC-05 Bluetooth Module can be used in two modes of operation: Command Mode and Data Mode.

- **Command Mode**

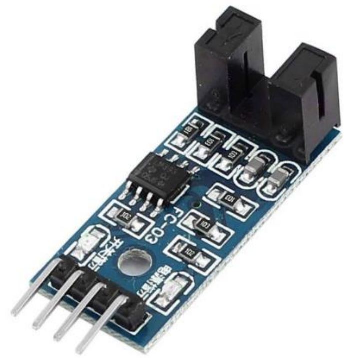
In Command Mode, you can communicate with the Bluetooth module through AT Commands for configuring various settings and parameters of the Module like get the firmware information, changing Baud Rate, changing module name, it can be used to set it as master or slave.

A point about HC-05 Module is that it can be configured as Master or Slave in a communication pair. In order to select either of the modes, you need to activate the Command Mode and sent appropriate AT Commands.

- **Data Mode**

Coming to the Data Mode, in this mode, the module is used for communicating with other Bluetooth device i.e. data transfer happens in this mode.

3.7. Speed Sensor



We used LM 393 speed sensor shown in the figure:
It consists of two parts: the sensor part and the control part.

- Sensor Part:
an Infrared LED and an NPN Photo Transistor. These two components are placed facing each other is a special housing made of black thermoplastic. This special housing ensures that the Photo Transistor receives light only from the Infrared LED and all the external source of light is eliminated.

- Control Part: LM393 Voltage Comparator and a few passive electronic components. The signal from the photo transistor is given to the LM393 and based on the presence or absence of an object between the Infrared LED and the Photo Transistor, the Output of the LM393 IC will either be HIGH or LOW. So in a nutshell it's the IR LED always on and passes an IR ray if it meets the motor's wheel slit so IR passes to the NPN photo transistor so the voltage comparator results HIGH, and if the IR ray meets blank areas between motor's wheel slits it blocks the ray so it doesn't reach the photo transistor so the voltage comparator results LOW.
Designed with slotted optical switch sensor, indicator light. Working with official MCU board. Widely used in motor speed detection, pulse counter.

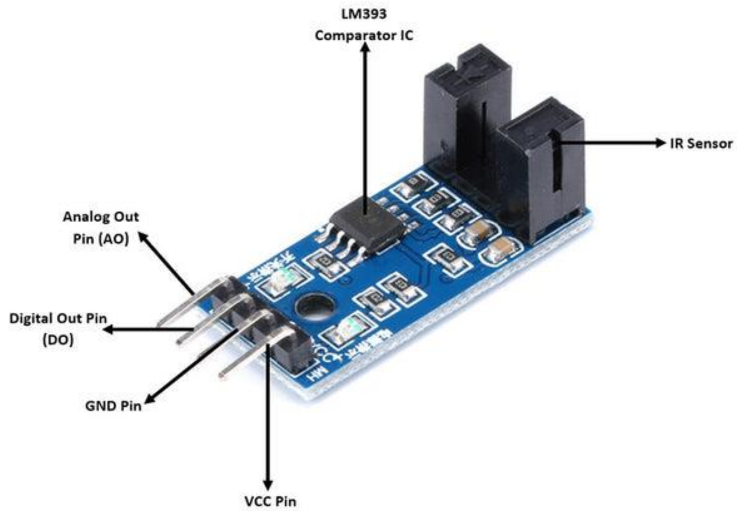
Features:

- Using imported groove optical coupling sensor, width 5mm.
- It has the output state light, if output high level, the lights are off. if output low level, it on.
- If it covered, it will output high level; otherwise, it output low level.
- Good signal and waveform, with strong driving ability for more than 15mA.
- The working voltage of 3.3V to 5V.
- Output: digital switch output (0 and 1).
- Equipped with a fixed bolt hole, easily install.
- Small board PCB size: 3.2 cm x 1.4 cm.
- Use the LM393 wide voltage comparer.

Pin Configuration:

Pin Name	Description
VCC	Connect positive 3.3 or 5 v power supply
GND	Ground Pin of module.
DO	Digital out pin
AO	Analog out pin

- The DO output interface can be directly connected to a micro-controller IO port, if there is a block detection sensor, such as the speed of the motor encoder can detect.



CHAPTER

4

SYSTEM FEATURES

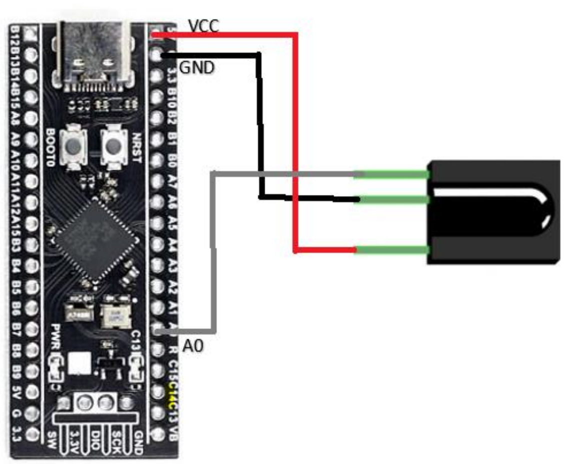
4. System Features

4.1. Door lock and Start system

4.1.1. Description

This feature is for the system and the vehicle's security that the system will not be enabled until it detects a specific signal from the user's remote control, this transmitted signal from the remote control will be received by the system from IR sensor, without this signal or if the signal was wrong all system features will be disabled and the vehicle will not be able to run.

4.1.2. Implementation



We connected the IR Sensor with the stm32 microcontroller to be the receiver that will receive the transmitted signal from the remote control.

Connections

IR sensor	Stm32 Microcontroller
VCC	5V
GND	GND
OUT	A0 (interrupt pin)

- When a key is pressed on the remote controller it transmits a signal this signal is received by IR senor which is connected to stm32 microcontroller, the output of the IR sensor will input the signal to the microcontroller at its interrupt pin so when the interrupt happen the systick peripheral will be enabled the start counting the time ticks from the first raising edge and to the second raising edge the calculate the time between them and put it into an array and so on for all the signal then check on the frame and its data to check for the right signal.

NEC Infrared Transmission Protocol

The NEC IR transmission protocol uses pulse distance encoding of the message bits.

- Logical '0' – a 562.5µs pulse burst followed by a 562.5µs space, with a total transmit time of 1.125ms.

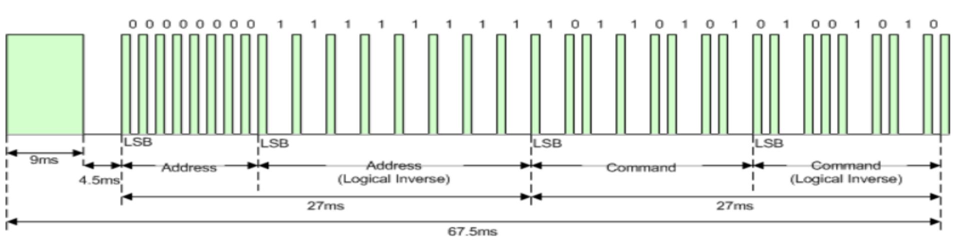
- Logical '1' – a 562.5µs pulse burst followed by a 1.6875ms space, with a total transmit time of 2.25ms.

The message transmitted consists of the following, in order:

- a 9ms leading pulse burst (16 times the pulse burst length used for a logical data bit).
- a 4.5ms space.
- the 8-bit address for the receiving device.
- the 8-bit logical inverse of the address.
- the 8-bit command.
- the 8-bit logical inverse of the command.
- a final 562.5µs pulse burst to signify the end of message transmission.

The four bytes of data bits are each sent least significant bit first.

As example:



Required software for this implementation

- RCC Driver.
- GPIO Driver.
- NVIC Driver.
- AXT/ANT Driver.
- SYSTIK Driver.

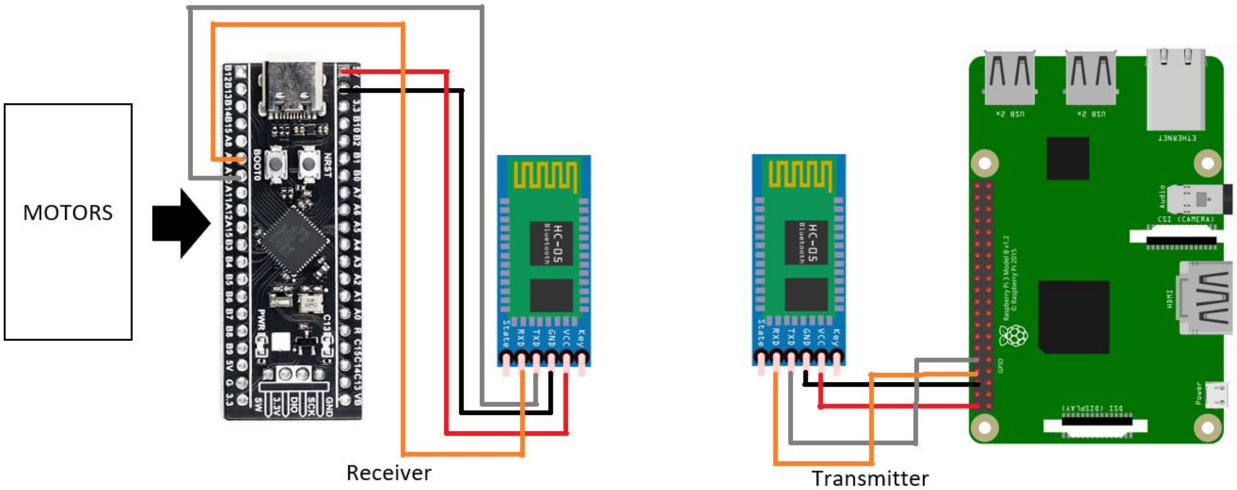
4.2. Vehicle’s Speed Control

4.2.1. Description

This feature for reducing the vehicle’s speed according to the road’s speed limit by Bluetooth module transmitter and receiver, The Transmitter is mounted on the Road Side and it is responsible to transmit the maximum allowable speed limit of that road to receiver, the receiver will be on vehicle. The system will manage the speed of the vehicle’s whenever it enters the Bluetooth module range. The transmitter will be transmitting the data continuously regardless presence of the vehicle with a range of up to 10m.

The transmitter system is a Bluetooth module configured as a master and Raspberry pi device and The Receiver system is embedded with the vehicle system, it comprises of Bluetooth module configured as slave, stm32 microcontroller.

4.2.2. Implementation



we connected the master bluetooth module with raspberry pi device (Transmitter) this part is the part that will be on the side of the road and then connected the slave bluetooth module with stm32 microcontroller the transmitter will tarnsmit the signal from it the stm32 microcontroller will know the speed limit of the road the control and control the speed of the motors to limit it to the needed speed limit.

Connections

Bluetooth module	Raspberry-Pi
VCC	5V
GND	GND
Tx	GPIO15
Rx	GPIO14

Bluetooth module	Stm32 microcontroller
VCC	5V
GND	GND
Tx	A10
Rx	A9

- First part of our implementation is to configure the transmitter bluetooth module as master and the reciever bluetooth module as slave, then pairing the two bluetooth modules together:

As we mentioned before at the hardware components chapter, the Bluetooth module has two modes of operation Data mode and command mode, so to make the configuration we will put it at the command mode and this happens by pressing on the enbale button before connecting the bluetooth VCC.

- **Slave Bluetooth module Configuration**

The Bluetooth module ay default is configured as a slave, so all we need to do is to get the address of the module by entering the AT command mode and writing these commands:

```
AT+NAME=SLAVE        #NAME OF SALVE
AT+ROLE=0            #SLAVE MODE
AT+CMODE=1           #CONNECT TO ANY ADDRESS
AT+ADDR              #ADDRESS OF SLAVE
```

- **Master Bluetooth module Configuration**

Here we want to configure the module as Master and then initialize the address of the slave on it to be paired together and to be able to send data from master to slave, so we did that by entering the AT command mode the writing the following commands:

```
AT+NAME=MASTER       #NAME OF MASTER
AT+ROLE=1            #MASTER MODE
AT+CMODE=0           #OPEN TO ANY COMMUNICATION WITH
                       ANY DEVICE
AT+ADDR              #ADDRESS OF SLAVE
```

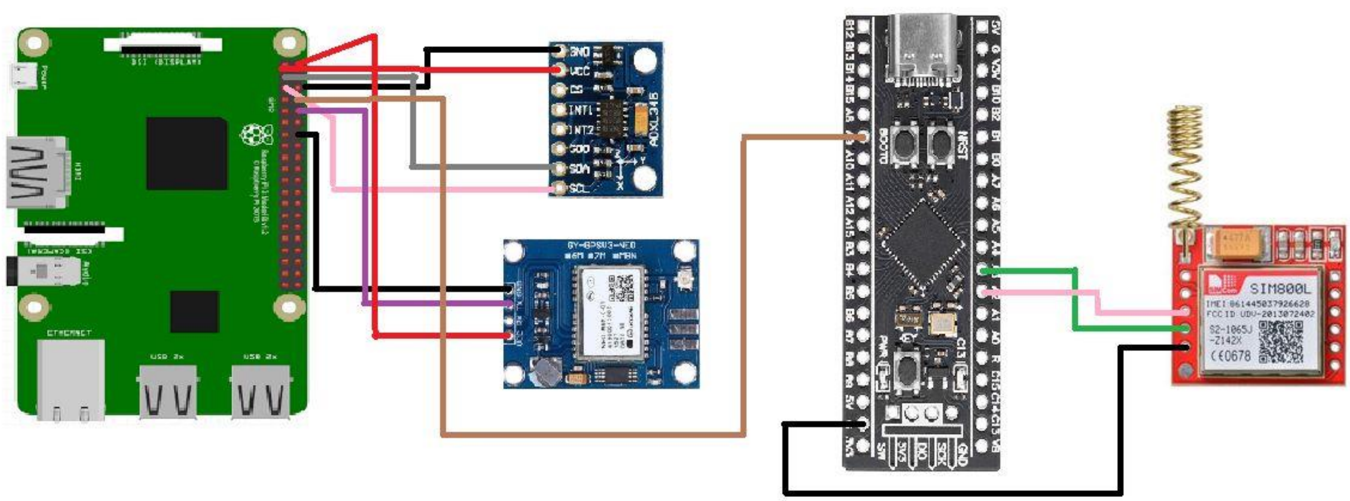
- Second part of the implementation is to program each module on its target and send a specific data from master to slave when the slave receives this data the microcontroller will check on it and compare the motors speed that is detected by the speed sensor by the received speed limit and control the speed of the motors by PWM signal to satisfy the required speed limit.

4.3. Accident Detection and Alert system

4.3.1. Description

This feature when the accident occurs it uses various components and alerts the Rescue team for help. An efficient automatic accident detection with an automatic notification to the emergency service with the accident location is a prime need to save the precious human life. It detects the occurrence of the accidents by using accelerometer and then reads the exact latitude and longitude of the vehicle involved in the accident by GPS module and sends this information to nearest emergency service provider by GSM module.

4.3.2. Implementation



We connected the Accelerometer and the GPS module to the Raspberry pi device and the GSM module to the stm32 microcontroller, and as we mentioned before the raspberry pi device and the stm32 microcontroller is communicate together by UART communication protocol to send the status of the accelerometer and the location of the vehicle that is detected by the GPS from raspberry pi to stm32 microcontroller

Connections:

Accelerometer	Raspberry-Pi
GND	GND
VCC	3.3V
SDA	SDA PIN3
SCL	SCL PIN5

GPS Module	Raspberry-Pi
GND	GND
VCC	5V
TX	UART0 RX

GSM Module	STM32 Microcontroller
RXD	A2
TXD	A3
GND	GND

Raspberry-Pi	STM32 Microcontroller
UART0 - TX	A10

- First part in the implementation was on the raspberry pi device we configured the GPS module and accelerometer on it and download their libraries that are used in their programming, at first, we programmed each module individually to take its readings.

  - GPS MODULE

As we mentioned before at the hardware components chapter, the GPS outputs its data on the form of NMEA format to after receiving this dah we parse it by using pynmea library to get the needed data only from it which is the latitude and the longitude.

GPS Output:

```
$GPGSA,A,3,07,05,30,20,09,08,17,,,,,,2.68,1.45,2.25*0F
$GPGSV,2,1,08,02,00,247,,05,31,305,27,07,45,038,37,08,13,064,31*76
$GPGSV,2,2,08,09,29,111,33,17,05,163,25,20,46,266,26,30,62,352,23*76
$GPGLL,3000.33660,N,03112.36919,E,222436.00,A,A*6C
$GPRMC,222437.00,A,3000.33671,N,03112.36938,E,0.222,,220622,,,A*73
$GPVTG,,T,,M,0.222,N,0.412,K,A*26
$GPGGA,222437.00,3000.33671,N,03112.36938,E,1,07,1.45,63.7,M,15.3,M,,*69
$GPGSA,A,3,07,05,30,20,09,08,17,,,,,,2.68,1.45,2.25*0F
$GPGSV,2,1,08,02,00,247,,05,31,305,27,07,45,038,37,08,13,064,31*76
$GPGSV,2,2,08,09,29,111,32,17,05,163,25,20,46,266,27,30,62,352,23*76
$GPGLL,3000.33671,N,03112.36938,E,222437.00,A,A*6E
```

RMC (Recommended Minimum) is an NMEA frame which consists of 70 to 75 characters contains the needed data and other data which is unneeded**.**

```
$GPRMC,222437.00,A,3000.33671,N,03112.36938,E,0.222,,220622,,,A*73
```

Latitude: 3000.33671 Longitude: 03112.36938

We developed a code that decodes the RMC frame and extract data of it and started the testing of module alone. It contains the needed data which is latitude and longitude. Based on this investigation we found that RMC frame supports all needed information and it would be enough for our application.

- ACCELEROMETER

We got the output data of the accelerometer which is the three dimension axis (x,y,z) of the car and determined the normal behavior and the unnormal behavior of the accelerometer readings and then by this reading we can check if the vehicle did an accident or not.

After that we integrated the accelerometer and the GPS module together so whenever the accelerometer detects unexpected behavior, the raspberry pi takes the data from the GPS about the vehicle's location and send it to the stm32 microcontroller, in addition to the accelerometer status that will make the microcontroller knows that there is an accident happened using UART communication protocol.

\- Second part of the implementation is the interfacing of the GSM module with the stm32 microcontroller in which when ever the stm32 microcontroller received a data that indicates that there is an accident happened it will take this data and send it to the rescue team number as an SMS.

- GSM MODULE

The GSM configured at stm32 microcontroller by using AT commands to send SMS,

AT comments configurations:

AT	#STATUS
AT+CREG?	#NETWORK REGISTERATION
AT+CIMI	#REQUEST INTERNATIONAL MOBILE SUBSCRIBER INDENTITY
AT+PBF	#FIND PHONEBOOK ENTRIES
AT+CMGF	#SELECT SMS MESSAGE FORMAT
AT+CMGS	#SEND SMS MESSAGE

LED Status Indicators:

Blink every 1s: The chip is running but hasn’t made a connection to the cellular network yet.

Blink every 3s: The module has made contact with the cellular network and can send/receive voice and SMS.

Power Consumption:

One of the most important parts of getting the SIM800L module working is supplying it with enough power. The SIM800L, depending on its state, can be a relatively power-hungry device. The module’s maximum current draw is around 2A. So, we used power supply(12v-1A) and Buck converter (LM2596).

4.4. Voice and Visual Assistant

4.4.1. Description

this feature will provide the driver with the needed information that is detected by the the other features of the system, So it will show that the system received the right signal by the IR sensor by showing a welcome screen on the TFT display and a “welcome” voice will output from the speakers using the voice system, also when the driver drive in high speed and the system detect from the Bluetooth module that he exceeds the speed limit, the system will display a warning message that he exceeds the speed limit on the TFT display and a “warning you exceed the limit” voice will output from the speaker using the voice system.

4.4.2. Implementation

Voice Implementation

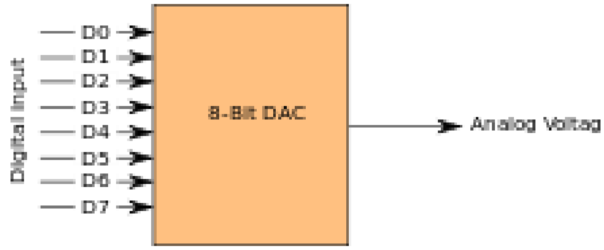
Voice is an analog signal like most of the signal around us in the world which we deal with on a daily basis, but microcontrollers can only handle the Digital signal so we should convert the signal from analog to digital to store it and then reverse it to display. The audio signal processing in Figure 1, ADC converts the analog signal collected by audio input equipment, such as a microphone, into a digital signal that can be processed by computer. The computer may add sound effect such as echo and adjust the tempo and pitch of the music. DAC converts the processed digital signal back into the analog signal that is used by audio output equipment such as a speaker.

Audacity Tool

By this tool we record the audio message and covert it to hex file (digital form) so we can store it in the memory of the microcontroller.

DAC (Digital to Analog Converter)

DAC is used to convert digital data streams into analog audio signal, so we can display it. Here is a simplified functional diagram of an 8-bit DAC.



Amplifier (LM386)

- LM386 amplifier provides a clear and loud sound.

- The UTC LM386 is a power amplifier, designed for use in low voltage consumer applications.

- The gain is internally set to 20 to keep external part count low, but the addition of an external resistor and capacitor between pin 1 and pin 8 will increase the gain to any value up from 20 to 200.

- The inputs are ground referenced while the output automatically biases to one-half the supply voltage.

- The quiescent power drain is only 24 mwatt when operating from a 6 voltage supply, making the LM386 ideal for battery operation.

Features of LM386 Amplifier

*Battery Operation
*Minimum External Parts
*Wide Supply Voltage Range: 4V~12V
*Low Quiescent Current Drain:4mA
*Voltage Gains:20~200
*Ground Referenced Input
*Self-Centering Output Quiescent Voltage
*Low Distortion: 0.2%
(Av =20, Vs=6V, RL=8Ω, Po=125mW, f=1kHz)

Speaker

The output from the LM386 Amplifier will be the input of the Speaker so we can hear the voice.