ARDUINO-RFID BASED SCHOOL BUS SAFETY AND SECURITY SYSTEM

A PROJECT REPORT

Submitted by

R.K.RAJAVENKADESH	**Register No.:13BEC118**
R.PRABHAKARAN	**Register No.:13BEC204**
K.SAKTHIVEL	**Register No.:13BEC212**
C.MOHANKUMAR	**Register No.:13BEC229**

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION

ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641049

(An Autonomous Institution Affiliated to Anna University, Chennai)

ANNA UNIVERSITY: CHENNAI

APRIL 2017

BONAFIDE CERTIFICATE

Certified that this project report **"ARDUINO-RFID BASED SCHOOL BUS SAFETY AND SECURITY SYSYTEM"** is the bonafide work of **R.K.RAJAVENKADESH [13BEC118], R.PRABHAKARAN [13BEC204], K.SAKTHIVEL [13BEC212] and C.MOHANKUMAR [13BEC229]** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Mr S.Arun Kumar M.Tech.,	Dr K.Malarvizhi M.E., Ph.D,
PROJECT SUPERVISOR	**HEAD OF THE DEPARTMENT**
Department of ECE	Department of ECE
Kumaraguru College of Technology	Kumaraguru College of Technology
Coimbatore-641049	Coimbatore- 641049

The candidates with Register No's: 13BEC118, 13BEC204, 13BEC212 and 13BEC229 are examined by us in the project viva-voce examination held on ___________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our sincere thanks to the Management of Kumaraguru College of Technology and Joint Correspondent **Shri.ShankarVanavarayar** for the kind support and for providing necessary facilities to carry out the project work.

We would like to express our sincere thanks to our beloved Principal **Dr.R.S.Kumar, M.E., Ph.D.,** Kumaraguru College of Technology, who encouraged us in each and every step of the project.

We would like to thank **Dr.K.Malarvizhi, M.E., Ph.D.,** Head of the Department, Electronics and Communication Engineering, for her kind support and for providing necessary facilities to carry out the project work.

We wish to thank with everlasting gratitude to our Project Coordinator **Dr.A.Vasuki, M.E., Ph.D.,** Department of Electronics and Communication Engineering for her consistent support throughout the course of this project work.

We are greatly privileged to express our deep sense of gratitude and heartfelt thanks to our Project Guide **Mr.S.Arun Kumar M.Tech.,** Department of Electronics and Communication Engineering for his expert counselling and guidance to make this project to a great deal of success and also we wish to convey our regards to all teaching and non-teaching staff of ECE Department for their help and cooperation.

Finally, we thank our parents and our family members for giving us the moral support and abundant blessings in all of our activities and our dear friends who helped us to endure our difficult times with their unfailing support and warm wishes.

ABSTRACT

In present time due to increase in number of kidnapping and road accident cases, parents always worry about the safety of their children. This project recommends a SMS based solution that assists parents to keep track of their children location in real time. To track the location, a GPS module is used and to identify the identity of the child. In addition, a RFID card is used in the system which uses Arduino Mega 2560 microcontroller. Whenever a child boards a bus, the RFID tag located in his/her identity card will be detected by the reader present in the bus and the system will identify the child and will send a text message to the parents consisting the current location and time. In this way the parents will be able to keep record of their kid’s whereabouts.

The project also proposes safety systems such as drunk and drive prevention system by using an alcohol sensor (MQ-3), accident alert with location by using piezo vibration sensor, detecting objects in front of vehicle wheels by using ping sensors and detection of human movements on the footboard by using IR proximity sensors. The working prototype model of proposed system exhibits good accuracy with reduced computational time. By using all those above features, we can assure the safety of every school going children.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

RFID	**Radio Frequency Identification**
GSM	**Global System for Mobile**
GPS	**Global Positioning System**
GPRS	**General Packet Radio Service**
IR	**Infra Red**
LED	**Light Emitting Diode**
LCD	**Liquid Crystal Display**
TX	**Transmitter**
RX	**Receiver**
NPN	**Negative Positive Negative**
DC	**Direct Current**
AC	**Alternative Current**
USB	**Universal Serial Bus**
UART	**Universal Asynchronous Receiver Transmitter**
PWM	**Pulse Width Modulation**
NC	**Normally Closed**
NO	**Normally Opened**
IO	**Input/output**
SRAM	**Static Random Access Memory**
EEPROM	**Electrically Erasable Programmable Read Only Memory**
MB	**Mega Bytes**
KB	**Kilo Bytes**

RTC	**Real Time Clock**
TCP/IP	**Transmission Control Protocol/Internet Protocol**
SIM	**Subscriber Identity Module**
TV	**Tele Vision**
DVD	**Digital Video Disk**
OS	**Operating System**
PLM	**Power Line Modem**
ICE	**In Circuit Emulator**
JVM	**Java Virtual Machine**
SPI	**Serial Peripheral Interface**
ICSP	**In Circuit Serial Programming**
TTL	**Transistor Transistor Logic**
I2C	**Inter Integrated Circuit**
ASCII	**American Standard Code for Information Interchange**
GNSS	**Global Navigation Satellite System**
CMOS	**Complementary Metal Oxide Semiconductor**
EGSM	**Extended Global System for Mobile**
PCM	**Pulse Code Modulation**
BAC	**Blood Alcohol Content**
PCB	**Printed Circuit Board**
IC	**Integrated Circuit**
RF	**Radio Frequency**
DPDT	**Double Pole Double Throw**

SPDT	**Single Pole Double Throw**
M2M	**Machine To Machine**
SMT	**Surface Mount Technology**
PC	**Personal Computer**
PDA	**Personal Digital Assistant**

CHAPTER 1

1 INTRODUCTION

School bus plays an essential role in carrying most of children everyday all over the world. While there are several problems that might disturb the parents with respect to the travel of school going kids; the project aspires to look into initiating the safety with respect of school buses through bus tracking and security system that will help the school kids' transportation in a protected and more secure way. The circumstance of forgetting kids on the bus is one of the problems suffered, that has risen considerably in recent years. This has often led to the demise of many students.

An article [4] published in India claims that in every eight minutes a child goes missing as data published by national crime records bureau. Statistical report says that around 50,000 children go missing every year from which 42% children are not found. This system, through entry and exit recordings, intends to create an appropriate environment via following certain set of criteria of security and wellbeing for the school transport that will have a positive impact on the student and their family.

Also drunken driving is a major factor for road accidents. Recently an article [5] was published in The Indian Express newspaper titled "Five children injured after drunk driver rams school bus into railing of bridge ".The driver of the bus was drunk. Police booked him for negligent driving. This shows that even school bus is not safe from the drunken drivers.

An article [6] in business standard news says that "10 year old boy crushed under the school bus wheels". In this case, the school bus driver moved the bus, without noticing a boy who fell under the wheels of the bus. This resulted in loss of the boy's life.

In Coimbatore 2010, a 10-year-old girl and her 7–year-old brother were abducted by a taxi driver while waiting for the van that usually takes them to the school [7].

To alert such road accidents, An Arduino RFID based system is developed with the following features,

RFID MODULE WITH GSM: To identify the identity of the child a RFID card is used which is built within the system. Whenever a child boards a bus, the RFID tag located in his identity card will be detected by the reader present in the bus and the system will identify the child and will send a text message by GSM to the parents consisting the message that "Your child is in/out the bus" with time.

VIBRATION SENSOR WITH GPS: In this feature, vibration sensor is placed in the bus to detect accidents. When an accident occurs, the location details of the bus is collected through the GPS and send to the parents & authority who wants the alerts.

ALCOHOL SENSOR: The system is equipped with alcohol sensor which is integrated on the steering wheel and will sense the percentage of alcohol in the air. If the concentration of alcohol in air is found above some prescribed limit; then the ignition is cut off and the driver will not be able to drive the bus thus keeping the students safe.

ULTRASONIC SENSORS: These sensors are placed under the bus to identify the objects/humans for alert the driver. By this we can avoid the accidents of innocent children and animals.

IR SENSOR: These sensors are placed at the foot board of the school bus to detect movements of children under running condition of bus. By this, driver will know the children is on the foot board and make sense of not to move the bus at stops.

1.1 BLOCK DIAGRAM

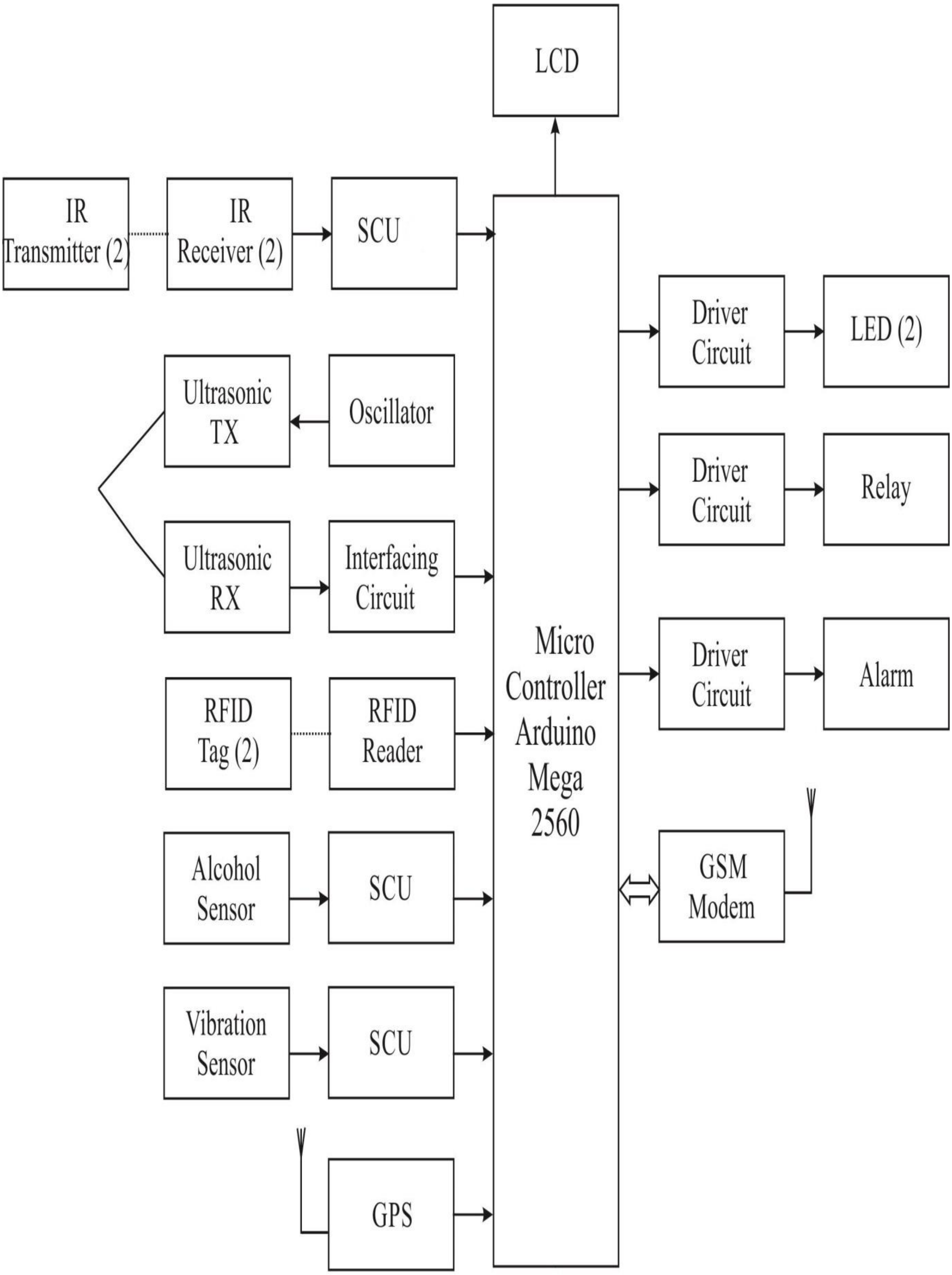


Figure 1

1.3 COMPONENTS LIST AND ITS USES

Arduino Mega 2560	The main microcontroller for this project.
EM-18 RFID Module	To detect and identify the children arrival/departure in the bus.
G.top 019 GPS Module	To obtain the location details of the bus.
SIMcom 800 GSM Module	To transmit and receive the GSM messages.
MQ-3 Alcohol sensor	To detect the alcohol in air inside the bus.
HC SR04 Ultrasonic sensor	To detect objects under the bus.
Piezo vibration sensor	To detect the accident.
AI-3035 Buzzer	To alert the driver when object is detected.
IR sensor	To identify the children movements on foot board.
LCD 16 x 2	To display messages or warnings.
Relay 12v	To turn off the ignition of the bus when alcohol is detected on the bus.
Power supply (12v/5v)	To give the supply voltage for all of the above DC components.

Table 1

1.4 PROJECT DESCRIPTION

The main aim of the project is to make children safe and secure. Humans are prone to being careless and because of that it might result in an irreparable loss. As said by elders, “Prevention is better than cure”, this project suggests a concept that, “Precaution is way better than suffering”. In this fast world where everything around us is happening so swift, the probability of accidents and kidnapping is also increasing to a massive range. Safer transportation of school children has been a critical issue as it is often observed that, kids find themselves locked in the school bus at the bus stop after going to school, they miss the bus, or ride the wrong bus with no way to track and secure them. Hence we have come up with a solution to overcome this huge tension.

The proposed circuit makes use of a Arduino (Mega 2560) that controls the RFID module with GSM, Vibration sensor with GPS, alcohol sensor (MQ-3), IR sensors (Human Movement Detection in the foot board), Ultrasonic sensor(Object detection under the bus) and a buzzer.

CHAPTER 2

2 HARDWARE DESCRIPTION

2.1 ARDUINO MEGA 2560

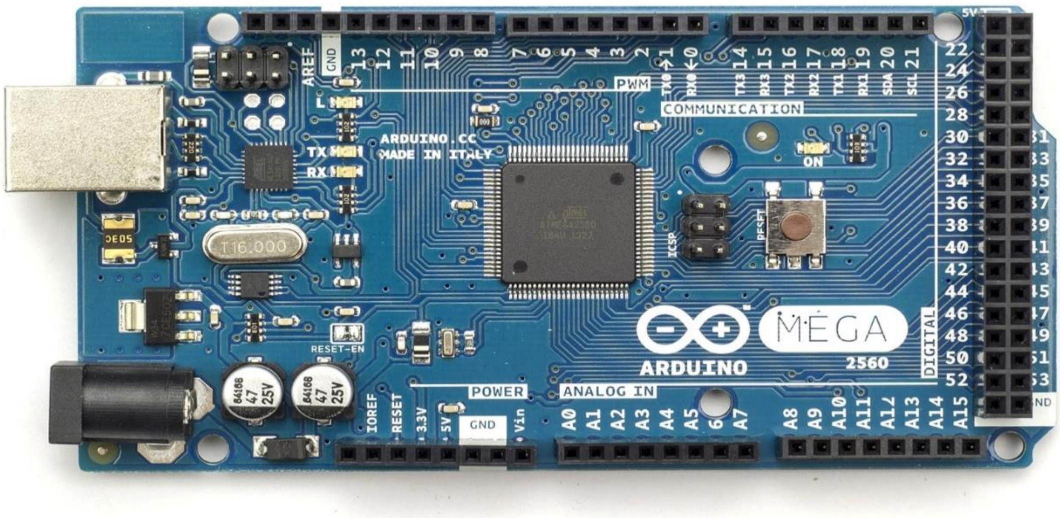


Figure 3: Arduino Mega 2560

Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Technical Specifications

Microcontroller	ATmega2560

Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power Configuration

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery.

The adapter can be connected by plugging a 2.1mm centre-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega328 programmed as a USB-to-serial converter.

The power pins are as follows:

- VIN: The input voltage to the Arduino 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: The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

- 3.3: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

- GND: Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 54 digital pins on the Mega 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 a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 Kohms. In addition, some pins have specialized functions:

• Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

• External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

• PWM: 0 to 13. Provide 8-bit PWM output with the analogWrite() function.

• SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.

• LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

• I2C: 20 (SDA) and 21 (SCL). Support I 2C (TWI) communication using the Wire library. Note that these pins are not in the same location as the I2C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provides 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 analogReference () function.

There are a couple of other pins on the board:

- AREF: Reference voltage for the analog inputs. Used with analogReference().
- Reset: Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically.

The Arduino software 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 ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega's digital pins. The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

Programming

The Arduino Mega2560 can be programmed with the Arduino software. The Atmega2560 on the Arduino Mega comes pre-burned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header.

2.2 EM-18 RFID MODULE

Figure 4: EM-18 RFID module

The EM-18 RFID Reader module operating at 125kHz is an inexpensive solution for RFID based application. The Reader module comes with an on-chip antenna and can be powered up with a 5V power supply. Power-up the module and connect the transmit pin of the module to receive pin of your microcontroller. Show the card within the reading distance and the card number is thrown at the output. Optionally the module can be configured for also a weight and output.

Features

The electronic and mechanical features of em-18 RFID module is given below,

Features	
RF Transmit Frequency	125kHz
Supported Standards	EM4001 64-bit RFID tag compatible
Communications Interface	TTL Serial Interface, Wiegand output
Communications Protocol	Specific ASCII
Communications Parameter	9600 bps, 8, N, 1
Power Supply	4.6V - 5.5VDC ± 10% regulated
Current Consumption	50 mA < 10mA at power down mode.
Reading distance	Up to 100mm, depending on tag
Antenna	Integrated
Size (LxWxH)	32 x 32 x 8mm

Table 2

Working

- Reader gets and executes commands from the back-end system.
- Reader emits radio frequency (RF) waves via its antenna.
- Waves travel through air and “energize” a passive transponder.
- Tag responds and transmits data signal via its antenna.
- Reader captures the tag data signal.
- Reader processes data signal.
- Reader delivers the processed information to the back-end system.

2.3 SIMCOM 800 GSM MODULE

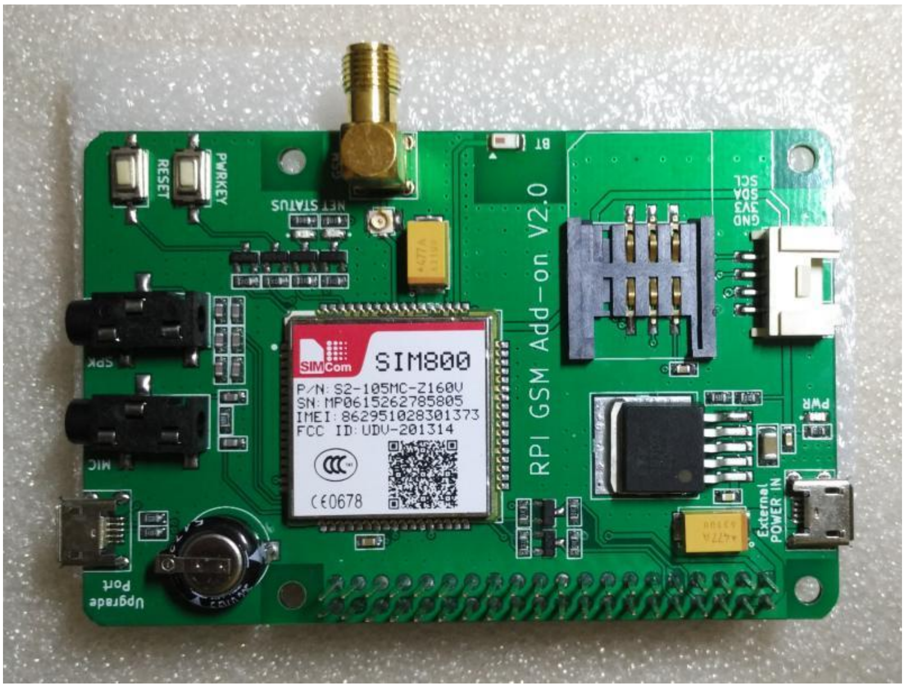


Figure 5: SIMcom 800

SIMcom 800 Overview

Designed for global market, SIM800 is a quad-band GSM/GPRS module that works on frequencies GSM850MHz, EGSM 900MHz, DCS 1800MHz and PCS 1900MHz. SIM800 features GPRS multi-slot class 12/ class10 (optional) and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4.

With a tiny configuration of 24x24x3mm, SIM800 can meet almost all the space requirements in user applications, such as M2M, smart phone, PDA and other mobile devices.

SIM800 has 68 SMT (Surface Mount Technology) pads, and provides all hardware interfaces between the module and customers' boards.

- Support up to 5x5x2 Keypads.
- One full function UART port, can be configured to two independent serial ports.

- One USB port can be used as debugging and firmware upgrading.
- Audio channels which include a microphone input and a receiver output.
- Programmable general purpose input and output.
- One SIM card interface.
- Support Bluetooth function.
- Support one PWM.
- PCM/SPI/SD card interface, only one function can be accessed synchronously. (default is PCM)

SIM800 is designed with power saving technique so that the current consumption is as low as 1.2mA in sleepmode.SIM800 integrates TCP/IP protocol and extended TCP/IP AT(Attention) commands which are very useful for data transfer applications.

A GSM module is used to send SMS, make and receive calls, and do other GSM operations by controlling it through simple AT commands from micro controllers and computers. It uses the highly popular SIM300 module for all its operations. It comes with a standard RS232 interface which can be used to easily interface the modem to micro controllers and computers. Fig 2.8 shows the GSM module.

The modem consists of all the required external circuitry required to start experimenting with the SIM300 module like the power regulation, external antenna, SIM Holder, etc. GSM300 usually operates at a baudrate of 9600, with 1 stop bits, No parity, No Hardware control and 8 data bits.

AT Commands

The AT commands that used to configure a GSM module is given below,

Command	Description
AT	It is the Prefix of every command sent to the modem. It is also used to test the condition of the modem.
AT+CSMINS?	Command to check if the Modem has a SIM inserted in it. It checks if the SIM is inserted.
AT+CMGF=1	Command to set the communication to textmode. By default the communication is in the PDU mode.
AT+CMGR=1	Command to read an SMS at the index one. Generally the index depends upon the how many number of SMS that a SIM can store. SIM Memory is the only memory available when GSM Modem is used and hence the number of SMS's stored depends on the SIM. It is usually 20. Any message received is arranged in the order of arrival at specific indices.

Table 3

FEATURES

- Uses the extremely popular SIM300 GSM module
- Provides the industry standard serial RS232 interface for easy connection to computers and other devices
- Provides serial TTL interface for easy and direct interface to microcontrollers
- Power, RING and Network LEDs for easy debugging
- On board 3V Lithium Battery holder with appropriate circuitry for providing backup for the module’s internal RTC
- Can be used for GSM based Voice communications, Data/Fax, SMS, GPRS and TCP/IP stack

- Can be controlled through standard AT commands
- Comes with an on board wire antenna for better reception.
- Board provides an option for adding an external antenna through an SMA connector
- The SIM300 allows an adjustable serial baud rate from 1200 to 115200 bps
- (9600 default)
- Modem a low power consumption of 0.25 A during normal operations and around 1 A during transmission
- Operating Voltage: 7 – 15V AC or DC (board has onboard rectifier.

RF Functionalities

The RF part of this module converts RF signals to baseband for receiver chain and translates base band signals into RF frequency spectrum.

The operating frequencies are:

Rx (EGSM 850): 869 to 894MHz

Tx (EGSM 850): 824 to 849MHz

Rx (EGSM 900): 925 to 960MHz

Tx (EGSM 900): 880 to 915MHz

Rx (DCS 1800): 1805 to 1880MHz

Tx (DCS 1800): 1710 to 1785MHz

Rx (PCS 1900): 1930 to 1990MHz

Tx (PCS 1900): 1850 to 1910MHz

Baseband Functionalities

The baseband part of SM5210 is composed of a SPREADTRUM‟s SC6600D chip. This chipset uses 0.18μm mixed signal CMOS technology which allows massive integration as well as low power consumption. SC6600D provides single chip solution to wireless Quad-band telephone handsets and data modems confirming to the EGSM 900, GSM 850, DCS 1800 and PCS 1900

Memory Functionalities

The memory used in this module is a combination of 32 Megabit (2M x 16Bit) CMOS 3.0 Volt-only, simultaneous operation flash memory and 4 Megabit (256K x 16-Bit) static RAM.

2.4 Gtop.019 GPS MODULE

Figure 6: Gtop 019 GPS module

The Global Positioning System (GPS) is a satellite-based navigation system made up of a network of 24 satellites placed into orbit by the U.S. Department of Defence. GPS was originally intended for military applications, but in the 1980s, the government made the system available for civilian use. GPS works in any weather conditions, anywhere in the world, 24 hours a day. There are no subscription fees or setup charges to use GPS.

Working

GPS satellites circle the earth twice a day in a very precise orbit and transmit signal information to earth. GPS receivers take this information and use triangulation to calculate the user's exact location. Essentially, the GPS receiver compares the time a signal was transmitted by a satellite with the time it was received. The time difference tells the GPS receiver how far away the satellite is. Now, with distance measurements from a few more satellites, the receiver can determine the user's position and display it on the unit's electronic map.

A GPS receiver must be locked on to the signal of at least three satellites to calculate a 2D position (latitude and longitude) and track movement. With four or more satellites in view, the receiver can determine the user's 3D position (latitude, longitude and altitude). Once the user's position has been determined, the GPS unit can calculate other information, such as speed, bearing, track, trip distance, distance to destination, sunrise and sunset time and more.

Overview

The GlobalTop PA6E-CAM module, which utilizes the MediaTek new generation GNSS ChipsetMT3333, has the highest level of sensitivity (-165dBm) and Time-to-First Fix (TTFF) by precise GNSS signal processing. It supports both GPS and GLONASS (**Global Navigation Satellite System)** simultaneously positioning system. Benefiting by the GLONASS, It gives the most possible to work in a more precise positioning and better TTFF even under poor GPS signals.

It support up to 210 PRN channels with 99 search channels and 33 simultaneous tracking channels. Up to 12 multi-tone active interference canceller (ISSCC2011 award), customer can have more flexibility in system design.

The PA6E-CAM has a built-in GLONASS and GPS band patch antenna, which simplifies customer's integration of GNSS positioning system.

Pin Configuration

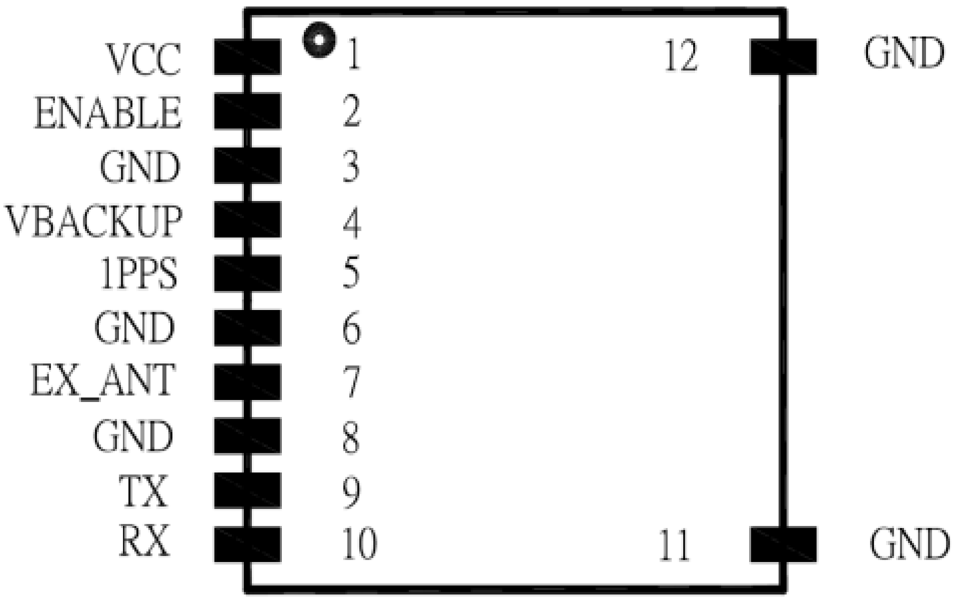


Figure 7

Pin Description

Pin	Name	I/O	Description & Note
1	VCC	PI	Main DC power input
2	ENABLE	I	High active, or keep floating for normal working
3	GND	P	Ground
4	VBACKUP	PI	Backup power input
5	1PPS	O	1PPS Time Mark Output 2.8V CMOS Level
6	GND	P	Ground
7	EX_ANT	I/O	External Antenna 3.0V input and output for external
8	GND	P	Ground
9	TX	O	Serial data output of NMEA
10	RX	I	Serial data input for firmware update
11	GND	P	Ground
12	GND	P	Ground

Table 4

2.5 MQ-3 ALCOHOL SENSOR

Sensor **Placement**

Figure 8

Figure 9

Description

Blood alcohol content (BAC), also called blood alcohol concentration, blood ethanol concentration, or blood alcohol level is most commonly used as a metric of alcohol intoxication for legal or medical purposes.

Blood alcohol content is usually expressed as a percentage of alcohol (generally in the sense of ethanol) in the blood. For instance, a BAC of 0.10 means that 0.10% (one tenth of one percent) of a person's blood, by volume (usually, but in some countries by mass), is alcohol.

This alcohol sensor is suitable for detecting alcohol concentration on your breath, just like your common breath analyser. It has a high sensitivity and fast response time. Sensor provides an analog resistive output based on alcohol concentration.

Features

- 5V DC or 5V AC

- Requires heater voltage
- Operation Temperature: -10 to 70 degrees C
- Heater consumption: less than 750mW

Dimensions

- 16.8mm diameter
- 9.3 mm height without the pins

2.6 ULTRASONIC SENSOR

Sensor **Placement**

Figure 10

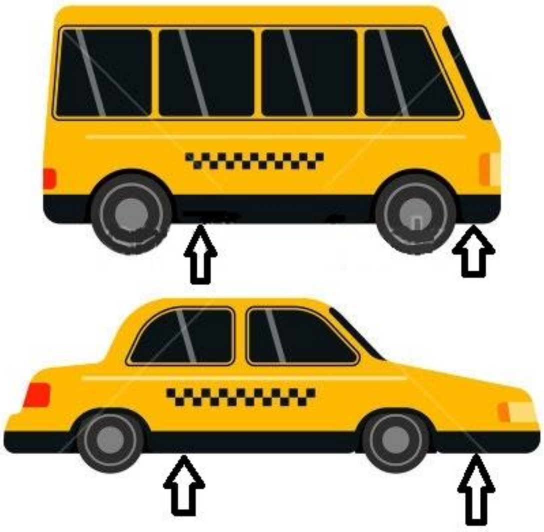


Figure 11

Description

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

- Using IO trigger for at least 10us high level signal,
- The module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- If the signal back, through high level, time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time × velocity of sound (340m/s) / 2

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

Electronic Features

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Table 5

2.7 PIEZO VIBRATION SENSOR

Sensor **Placement**

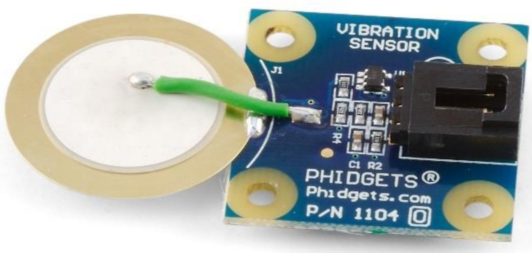


Figure 12

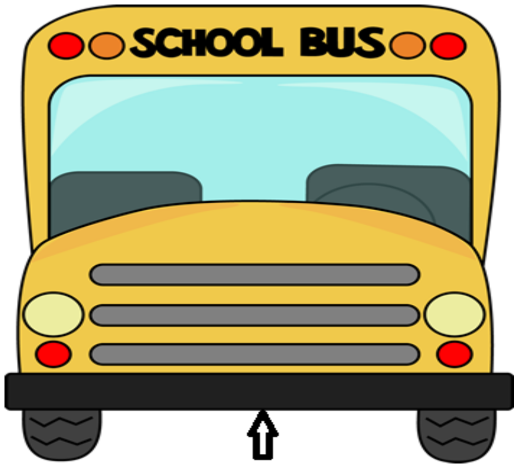


Figure 13

Description

Vibration sensor is a piezo electric transducer. It works on the principle of piezo electric effect. Piezoelectric transducers are based on the property of accumulating charges if stressed (direct effect) and to strain in case of an electric signal is applied across their electrodes (inverse effect).

Mechanical compression or tension on a poled piezoelectric ceramic element changes the dipole moment, creating a voltage. Compression along the direction of polarization, or tension perpendicular to the direction of polarization, generates voltage of the same polarity as the poling voltage.

Working

- Piezo electric plate converts the mechanical vibration to electrical signal.
- The electrical signal voltage is given to amplifier unit.
- The amplified output is in the form of AC signal the diode is used to rectify the negative signal.

VIBRATION SENSOR

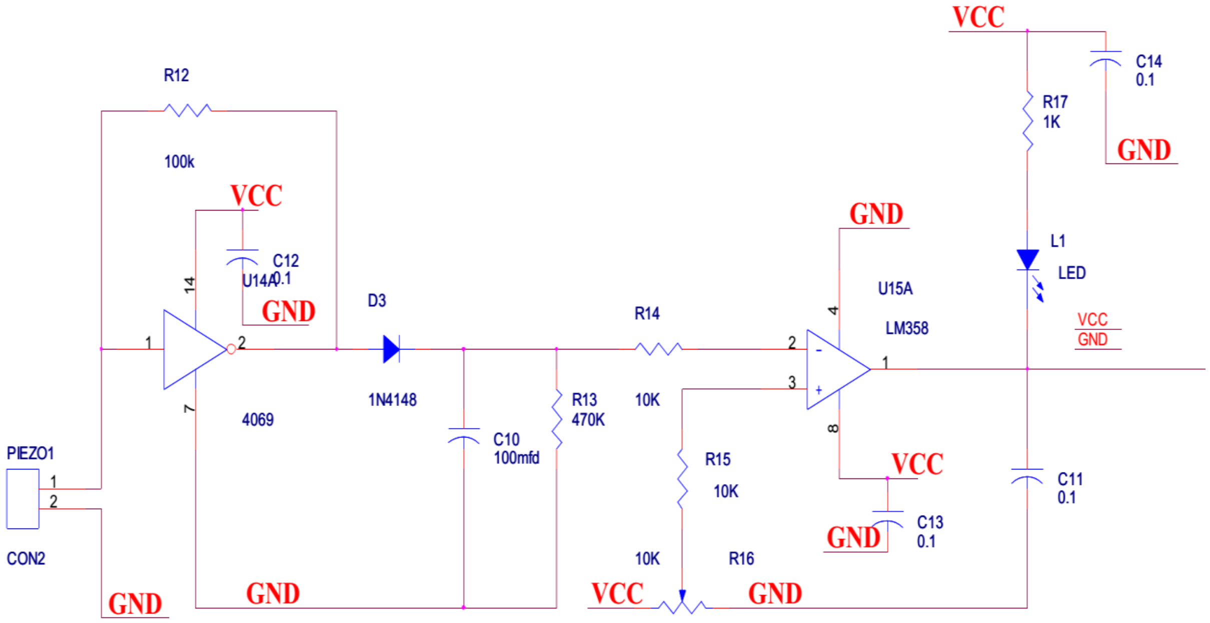


Figure 14

- Then the electrical signal voltage is given to comparator unit through 0.1uf capacitor in order to filter the noise signal.
- The rectified signal is given to comparator.
- The comparator circuit is constructed with LM358 operational amplifier in which the signal is given to inverting input terminal.
- The reference voltage is given to noninverting input terminal.
- It converts the input signal to +5V to 0V square pulse.
- The square pulse is given to microcontroller.
- Comparator output is always in digital form so we can connect directly to the microcontroller.
- At the time of vibration the output will be in zero state.
- The reference voltage is given to no inverting input terminal.

- So the LED already gets the positive 5 volt power supply in anode side.
- Now LED receive ground potential and it will be in a forward bias so the will glow.
- The output pin is monitoring by using any microcontroller or processor.

2.8 IR SENSOR

Sensor **Placement**

Figure 15 **Figure 16**

Circuit with Description

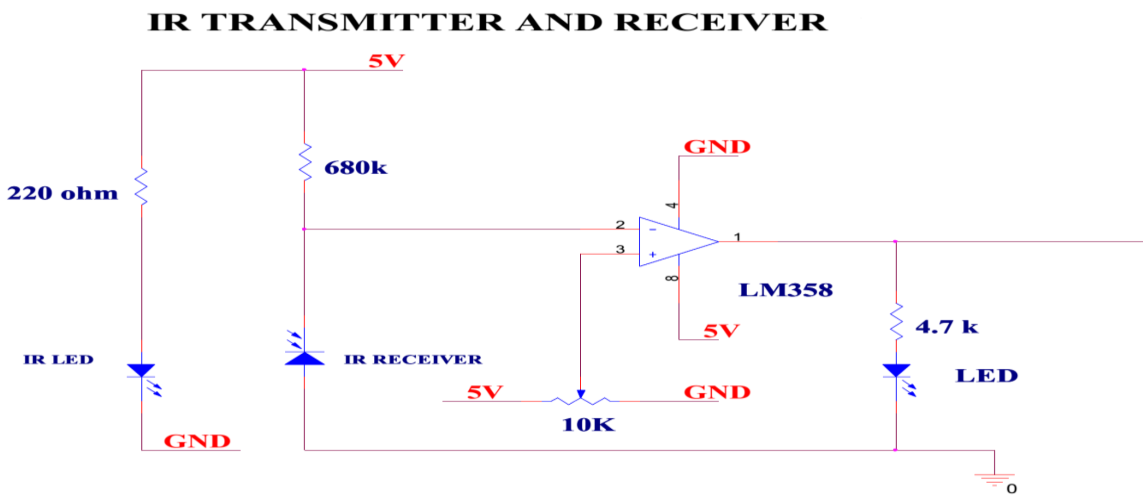


Figure 17

Infrared transmitter is one type of LED which emits infrared rays generally called as IR Transmitter. Similarly IR Receiver is used to receive the IR rays transmitted by the IR transmitter. One important point is both IR transmitter and receiver should be placed in line of sight.

The transmitted signal is given to IR transmitter whenever the signal is high, the IR transmitter LED is conducting it passes the IR rays to the receiver. The IR receiver is connected with comparator. The comparator is constructed with LM 358 operational amplifier. In the comparator circuit the reference voltage is given to inverting input terminal. The non-inverting input terminal is connected IR receiver. When interrupt the IR rays between the IR transmitter and receiver, the IR receiver is not conducting. So the comparator non inverting input terminal voltage is higher than inverting input. Now the comparator output is in the range of +5V. This voltage is given to microcontroller or PC. So led will glow.

When IR transmitter passes the rays to receiver, the IR receiver is conducting due to that non inverting input voltage is lower than inverting input. Now the comparator output is GND so the output is given to microcontroller or PC. This circuit is mainly used to for counting application, intruder detector etc.

2.9 RELAY MODULE

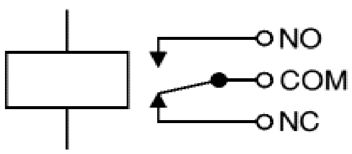
Figure 18

Description

A relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and they are double throw (changeover) switches. Relays allow one circuit to switch a second circuit which can be completely separate from the first. For example a low voltage battery circuit can use a relay to switch a 230V AC mains circuit. There is no electrical connection inside the relay between the two circuits; the link is magnetic and mechanical.

The coil of a relay passes a relatively large current, typically 30mA for a 12V relay, but it can be as much as 100mA for relays designed to operate from lower voltages. Most ICs (chips) cannot provide this current and a transistor is usually used to amplify the small IC current to the larger value required for the relay coil. The maximum output current for the popular 555 timer IC is 200mA so these devices can supply relay coils directly without amplification.

Relays are usually SPDT or DPDT but they can have many more sets of switch contacts, for example relays with 4 sets of changeover contacts are readily available. Most relays are designed for PCB mounting but it consist soldered wires directly to the pins to avoid melting the plastic case of the relay. The animated picture shows a working relay with its coil and switch contacts. We can see a lever on the left being attracted by magnetism when the coil is switched on. This lever moves the switch contacts. There is one set of contacts (SPDT) in the foreground and another behind them, making the relay DPDT.



The relay's switch connections are usually labelled COM, NC and NO:

- **COM** = Common, always connect to this, it is the moving part of the switch.
- **NC** = Normally Closed, COM is connected to this when the relay coil is **off**.
- **NO** = Normally Open, COM is connected to this when the relay coil is **on**.

2.10 BUZZER

Buzzer **Circuit**

Figure 20

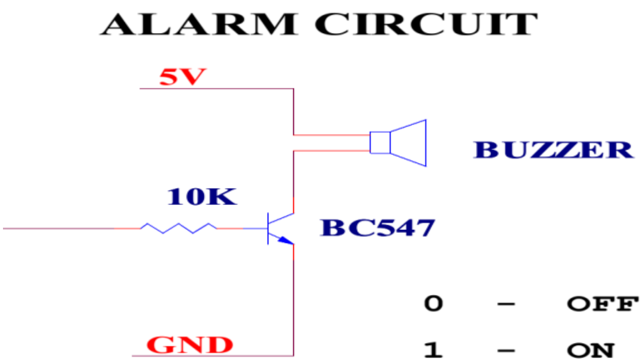


Figure 19

Circuit Description

The circuit is designed to control the buzzer. The buzzer ON and OFF is controlled by the NPN transistor (BC 547). The buzzer is connected in the transistor collector terminal.

When high pulse signal is given to base of the transistors it will be turned on and now alarm get ground so it will be on.

If low pulse is given to the NPN transistor base means it will be off and also alarm goes to the off state.

Voltage signal from Arduino	Transistor	Buzzer
1	**ON**	**ON**
0	**OFF**	**OFF**

Table 6

2.11 LCD 16X2

Figure 21

Description

The module, interfaced to the system, can be treated as RAM input/output, expanded or parallel I/O. Since there is no conventional chip select signal, developing a strobe signal for the enable signal (E) and applying appropriate signals to the register select (RS) and read/write (R/W) signals are important. The module is selected by gating a decoded module – address with the host – processor's read/write strobe. The resultant signal, applied to the LCDs enable (E) input, clocks in the data. The 'E' signal must be a positive going digital strobe, which is active while data and control information are stable and true. The falling edge of the enable signal enables the data / instruction register of the controller. All module timings are referenced to specific edges of the 'E' signal. The 'E' signal is applied only when a specific module transaction is desired. The read and write strobes of the host, which provides the 'E' signals, should not be linked to the module's R/W line. An address bit which sets up earlier in the host's machine cycle can be used as R/W. When the host processor is so fast that the strobes are too narrow to serve as the 'E' pulse

a. Prolong these pulses by using the hosts 'Ready' input
b. Prolong the host by adding wait states
c. Decrease the Hosts Crystal frequency

In spite of doing the above mentioned, if the problem continues, latch both the data and control information and then activate the 'E' signal When the controller is performing an internal operation he busy flag (BF) will set and will not accept any instruction. The user should check the busy flag or should provide a delay of approximately 2ms after each instruction. The module presents no difficulties while interfacing slower MPUs. The liquid crystal display module can be interfaced, either to 4-bit or 8-bit MPUs. For 4-bit data interface, the bus lines DB4 to DB7 are used for data transfer, while DB0 to DB3 lines are disabled. The data transfer is complete when the 4-bit data has been transferred twice. The busy flag must be checked after the 4-bit data has been transferred twice. Two more 4-bit operations then transfer the busy flag and address counter data. For 8-bit data interface, all eight-bus lines (DB0 to DB7) are used.

2.12 POWER SUPPLY

Block Diagram

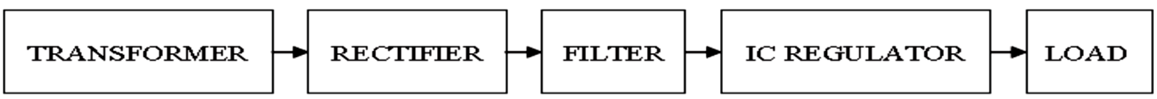


Figure 22

Circuit Description

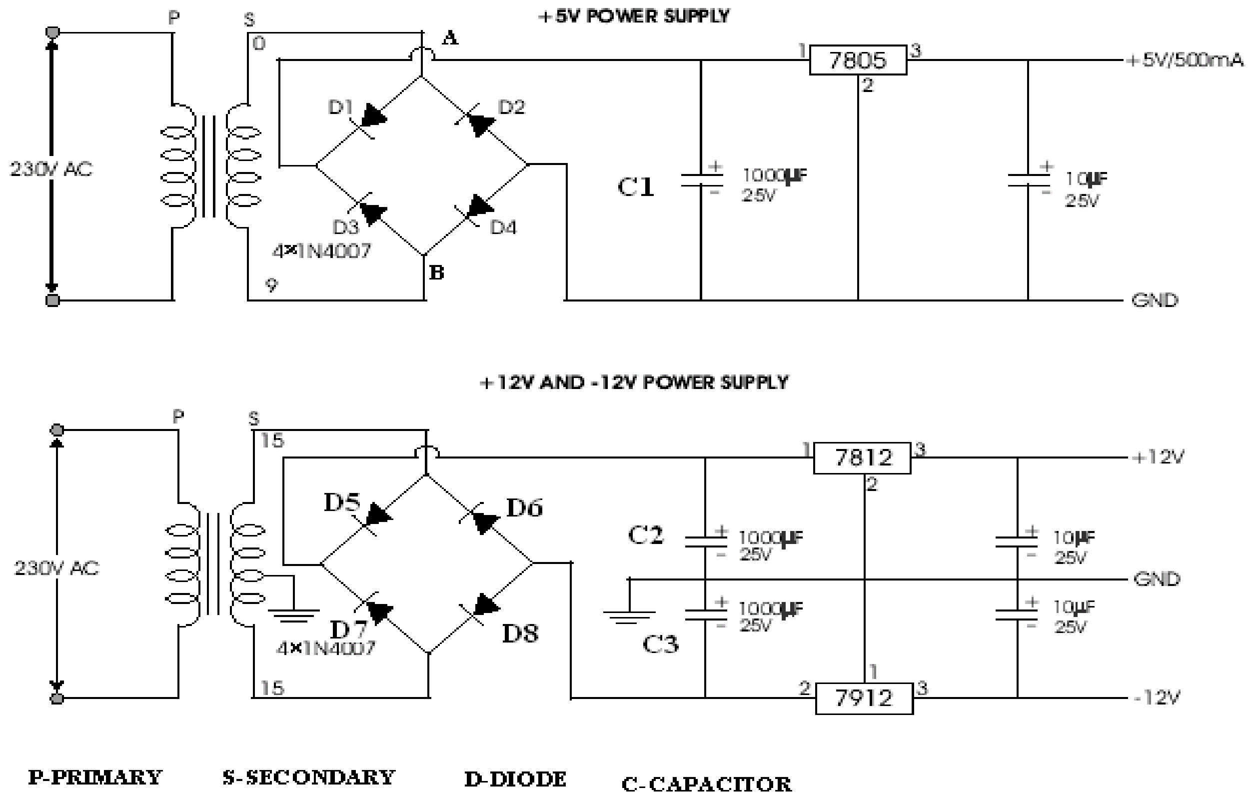


Figure 23

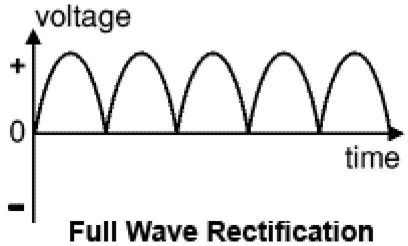
Working principle

Transformer

The potential transformer will step down the power supply voltage (0-230V) to (0-9Vand 15-0-15) level. If the secondary has less turns in the coil then the primary, the secondary coil's voltage will decrease and the current or AMPS will increase or decreased depend upon the wire gauge. This is called a STEP-DOWN transformer. Then the secondary of the potential transformer will be connected to the rectifier.

Bridge rectifier

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners.



Full Wave Rectification
(Varying DC)

Let us assume that the transformer is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4. The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow.

The path for current flow is from point B through D1, up through Load, through D3, through the secondary of the transformer back to point B. One-half cycle later the polarity across the secondary of the transformer reverse, forward biasing D2 and D4 and reverse biasing D1 and D3. Current flow will now be from point A through D4, up through Load, through D2, through the secondary of transformer, and back to point A. Across D2 and D4. The current flow through Load is always in the same direction.

In flowing through Load this current develops a voltage corresponding to that. Since current flows through the load during both half cycles of the applied voltage, this bridge rectifier is a full-wave rectifier. One advantage of a bridge rectifier over a conventional full-wave rectifier is that with a given transformer the bridge rectifier produces a voltage output that is nearly twice that of the conventional half-wave circuit.

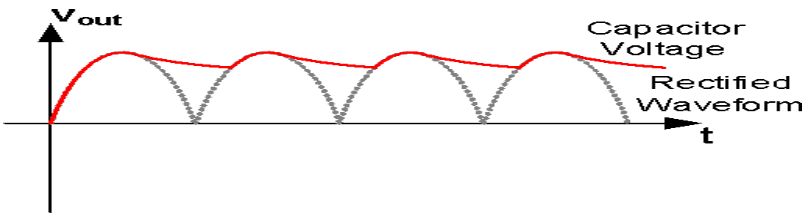
This bridge rectifier always drops 1.4volt of the input voltage because of the diode. We are using 1N4007 PN junction diode, its cut off region is 0.7Volt. So any two diodes are always conducting, total drop voltage is 1.4 volt.

Positive 12volt and Negative 12 volt circuit

The unregulated AC/DC power supply part of the circuit consists of a transformer that steps down 230VAC to 15 volts across a center tapped secondary winding 15V AC individually across the two halves of the secondary winding with opposite polarities, diodes (D5) to (D8) that rectify the AC appearing across the secondary with (D5) and (D7) providing ‘full wave rectification to produce a positive output, (D6) and (D8), providing full wave rectification to produce a negative output, capacitors (C2) and (C3) providing the filtering action. 7812 is a fixed output positive three terminal regulator whereas 7912 is a fixed output negative three terminal regulator.

Filter

If a Capacitor is added in parallel with the load resistor of a Rectifier to form a simple Filter Circuit, the output of the Rectifier will be transformed into a more stable DC Voltage. At first, the capacitor is charged to the peak value of the rectified Waveform.



Beyond the peak, the capacitor is discharged through the load resistor until the time at which the rectified voltage exceeds the capacitor voltage. Then the capacitor is charged again and the process repeats itself.

IC voltage regulators

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustable set voltage. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from milli watts to tens of watts.

A fixed three-terminal voltage regulator has an unregulated dc input voltage , applied to one input terminal, a regulated dc output voltage, from a second terminal, with the third terminal connected to ground. The series 78 regulators provide fixed positive regulated voltages from 5 to 24 volts. Similarly, the series 79 regulators provide fixed negative regulated voltages from 5 to 24 volts.

This is a regulated power supply circuit using the 78xx IC series. These regulators can deliver current around 1A to 1.5A at a fix voltage levels. The common regulated voltages are 5V, 6V, 8V, 9V, 10V, 12V, 15V, 18V, and 24V. It is important to add capacitors across the input and output of the regulator IC to improve the regulation. In this power supply circuit we get 5, 12 and -12Volt output.

CHAPTER 3

3 SOFTWARE DESCRIPTION

3.1 EMBEDDED C

Most of the electronic devices such as digital cameras mobile phones washing machines etc. has some kind of processors in it and are called the embedded systems. The hardware components form the body of the embedded system whereas the embedded software coded in the processor forms the brain of the system. It is the embedded software which primarily governs the functioning of embedded systems. Initially assembly level languages are used to code the processors. As assembly language programs are specific to a processor, assembly language did not offer portability across systems. To overcome this disadvantage, several high level languages, including C, came up. Some other languages like PLM, Modula-2, Pascal, etc. also came but couldn’t find wide acceptance. Amongst those, C got wide acceptance for not only embedded systems, but also for desktop applications. Due to the wide acceptance of C in the embedded systems, various kinds of support tools like compilers & cross-compilers, ICE, etc. came up and all this facilitated development of embedded systems using C.

3.2 DIFFERENCE BETWEEN C AND EMBEDDED C

- Normal C does not provide any feature to address a bit value. Minimum of 1 byte can be addressed. Using embedded c it is possible even to access a single bit value.
- C is a widely used general purpose high level programming language mainly intended for system programming. Embedded C is an extension to C programming language that provides support for developing efficient programs for embedded devices.

- C is for desktop computers, embedded C usually is for microcontroller based applications.
- C use the resources of desktop computers (memory, OS, etc.)
- Embedded C use only limited resources available in chip
- C has a free-format program source code, in a desktop computer while embedded C has different format based on embedded processor (microcontrollers/microprocessors).
- C has normal optimization, in programming while embedded C high level optimization in programming.
- Embedded C includes extra features over C, such as fixed point types, multiple memory areas, and I/O register mapping.
- C programming must have required operating system while embedded C may or may not be required operating system.
- C language uses the desktop OS memory while embedded C uses the controllers inbuilt or any externally attached memory.
- Compilers for C typically generate OS dependent executable. i.e. it allows the users to run program from OS terminal directly while, embedded C requires compilers to create files, and downloaded to the processor, (microcontrollers/microprocessors) where it needs to run.
- C programming run in console, i.e. the output can be seen in the OS (desktop) while, embedded C run in real time constraints.
- In C programming it is easy to input program data, when running while, embedded C have pre-defined data that have been given while programming.
- Embedded c code generates a. hex file while a convention c code generates a compatible .exe file.

Example of C program is, OS based software, simple logic program, etc. Example of embedded C is TV, DVD, washing machine, etc.

3.3 ADVANTAGES

- Embedded compilers give access to all resources which is not provided in compilers for desktop computer applications.
- C is easier to use for making more complex programs.
- C syntax is a lot easier to learn than Assembler syntax.
- Compared to other high level languages, C offers more flexibility because C is relatively small, structured language; it supports low-level bit-wise data manipulation.
- It is small and simpler to learn, understand, program and debug.
- Compared to assembly language, C code written is more reliable and scalable, more portable between different platforms.
- C compilers are available for almost all embedded devices in use today.
- Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/microcontroller or any system. This makes it convenient for a user to develop programs that can run on most of the systems.
- As C combines functionality of assembly language and features of high level languages, C is treated as a „middle-level computer language‟‟ or „high level assembly language‟‟.
- It is fairly efficient.
- It supports access to I/O and provides ease of management of large embedded projects.

- Java is also used in many embedded systems but Java programs require the Java Virtual Machine (JVM), which consumes a lot of resources. Hence it is not used for smaller embedded devices.

3.4 PROGRAM

```
#include <LiquidCrystal.h>

#include <EEPROM.h>

#include <string.h>


LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);/*Define arduino pins for lcd*/


/*define inputs*/

#define alcohol 3

#define ir1 5

#define ir2 6

#define echo 10


/*define outputs*/

#define relay 4

#define led1 7

#define led2 8

#define trig 9

#define alarm 11
```

```
/*variable declaration*/
char rfid[13],rec[2];
unsigned char no[]="979047945797904794579790479457";
int i,k,count1,count2,j,jj,vib,cc;
char card1[]="4800358218E7";
char card2[]="4B00404B9CDC";
unsigned long echoVal = 0;
unsigned long ultrasoundValue = 0;
unsigned long dist;
char a[70],gps[70];
void gsm_init();
void gsm_num(unsigned char *);
void gsm_send();
unsigned long ping1();
long microsecondsToCentimeters(long);
void Lcd_Decimal2(int, int, int);
void Lcd_Decimal3(int, int, int);
void Lcd_Decimal4(int, int, int);

/*Obtaining serial data in buffers*/
void buffer1_clear()
```

```
{
while(Serial1.available())
Serial1.read();
}

void buffer2_clear()
{
while(Serial2.available())
Serial2.read();
}

void buffer3_clear()
{
while(Serial3.available())
Serial3.read();
}

/*get rfid data*/
void serialEvent1()
{
if(Serial1.available())
{
```

```
rfid[i++] = Serial1.read();
}
}
void call()
{
vib=1;
}

/*Main function*/
void setup()
{
  pinMode(alcohol,INPUT);
  pinMode(ir1,INPUT);
  pinMode(ir2,INPUT);
  pinMode(echo,INPUT);
  pinMode(relay,OUTPUT);digitalWrite(relay,HIGH);
  pinMode(led1,OUTPUT);digitalWrite(led1,HIGH);
  pinMode(led2,OUTPUT);digitalWrite(led2,HIGH);
  pinMode(trig,OUTPUT);digitalWrite(trig,LOW);
  pinMode(alarm,OUTPUT);digitalWrite(alarm,LOW);
  lcd.begin(16, 2);
  lcd.setCursor(0,0);
```

```
  lcd.print("SCHOOL BUS TRACK");
  lcd.setCursor(0,1);
  lcd.print("& SECURE SYSTEM ");
  delay(2000); // 2 SECONDS
  Serial.begin(9600);
  Serial1.begin(9600); // for RFID
  Serial2.begin(9600); // for GSM
  Serial3.begin(9600); // for GPS
  lcd.clear();
  pinMode(2, INPUT);
  digitalWrite(2,HIGH);
  attachInterrupt(0, call, RISING);
  gsm_init();
}

/*For sequence execution*/
void loop()
{
dist = ping1();
Lcd_Decimal3(13,0,dist);

if(dist<20)
```

```
{
digitalWrite(alarm,HIGH);
}
else
{
 digitalWrite(alarm,LOW);
}

if(vib==1)
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("ACCIDENT OCCURED");
lcd.setCursor(0,1);
lcd.print("SENDING MESSAGE ");
buffer2_clear();
gsm_num(no+20);
Serial2.println("ACCIDENT OCCURED IN");
Serial2.print("LAT:");
for(i=20;i<=30;i++)
{
Serial2.write(gps[i]); // latitude value
```

```
delay(10);
}
enter();
delay(500);
Serial2.print("LON:");
for(i=32;i<=43;i++)
{
Serial2.write(gps[i]); // longitude value
delay(10);
}
delay(100);
gsm_send();
Serial2.flush();
lcd.setCursor(0,1);
lcd.print("  MESSAGE SENT  ");
delay(1000);
lcd.clear();
vib=0;
}

while(Serial3.available())
{
```

```
a[j] = Serial3.read();

if(a[0]!='$'){j=0;}

else if((a[0]=='$') && (a[1]!='G')){j=1;}

else if((a[0]=='$') && (a[1]=='G') && (a[2]!='P')){j=2;}

else if((a[0]=='$') && (a[1]=='G') && (a[2]=='P') && (a[3]!='R')) {j=3;}

else if((a[0]=='$') && (a[1]=='G') && (a[2]=='P') && (a[3]=='R') &&
(a[4]!='M')){j=4;}

else if((a[0]=='$') && (a[1]=='G') && (a[2]=='P') && (a[3]=='R') &&
(a[4]=='M') &&(a[5]!='C')){j=5;}

else if((a[0]=='$') && (a[1]=='G') && (a[2]=='P') && (a[3]=='R') &&
(a[4]=='M') &&(a[5]=='C')){j++;if(j>60){jj=1;}}

}


if(jj==1)

{

lcd.clear();

lcd.setCursor(0,0);

for(i=20;i<=30;i++)

{

gps[i]=a[i]; // received gps latitude value stored in variable (gps[])

lcd.write(gps[i]);

}
```

```
lcd.setCursor(0,1);

for(i=32;i<=43;i++)

{

gps[i]=a[i]; // received gps longitude value stored in variable (gps[])

lcd.write(gps[i]);

}

delay(1000);

lcd.clear();

buffer3_clear();

jj=0;j=0;

}

while(Serial2.available())

{

rec[k] = Serial2.read();

if(rec[0]=='*')

{

k++;

}

}


if(k>1)

{
```

```
}
lcd.clear();
lcd.setCursor(0,0);
lcd.print("REQUEST RECEIVED");
lcd.setCursor(0,1);
lcd.print("SENDING MESSAGE ");
buffer2_clear();
gsm_num(no+20);
Serial2.print("LAT:");
for(i=20;i<=30;i++)
{
Serial2.write(gps[i]); // latitude value
delay(10);
}
enter();
delay(500);
Serial2.print("LON:");
for(i=32;i<=43;i++)
{
Serial2.write(gps[i]); // longitude value
delay(10);
}
delay(100);
```

```
gsm_send();
Serial2.flush();
lcd.setCursor(0,1);
lcd.print("  MESSAGE SENT  ");
delay(1000);
lcd.clear();
for(cc=0;cc<2;cc++)
{
rec[cc]=0;
delay(10);
}
k=0;
}
if(i>11)
{
i=0;
lcd.clear();
lcd.setCursor(0,1);
lcd.print(rfid);
buffer2_clear();
if(strncmp(rfid,card1,12)==0)
{
```

```
count1++;

if(count1==1)

{

lcd.setCursor(0,0);

lcd.print("CHILDREN 1   IN ");

gsm_num(no);

Serial2.print("CHILDREN 1 INSIDE BUS");

gsm_send();

}

else if(count1==2)

{

lcd.setCursor(0,0);

lcd.print("CHILDREN 1   OUT");

gsm_num(no);

Serial2.print("CHILDREN 1 OUTSIDE BUS");

gsm_send();

count1=0;

}

}

else if(strncmp(rfid,card2,12)==0)

{

count2++;
```

```
if(count2==1)

{

lcd.setCursor(0,0);

lcd.print("CHILDREN 2   IN ");

gsm_num(no+10);

Serial2.print("CHILDREN 2 INSIDE BUS");

gsm_send();

}

else if(count2==2)

{

lcd.setCursor(0,0);

lcd.print("CHILDREN 2   OUT");

gsm_num(no+10);

Serial2.print("CHILDREN 2 OUTSIDE BUS");

gsm_send();

count2=0;

}

}

for(cc=0;cc<13;cc++)

{

rfid[cc]=0;

delay(10);
```

```
}
i=0;
lcd.clear();
}

if(digitalRead(ir1)==HIGH){digitalWrite(led1,LOW);}
else{digitalWrite(led1,HIGH);}

if(digitalRead(ir2)==HIGH){digitalWrite(led2,LOW);}
else{digitalWrite(led2,HIGH);}

if(digitalRead(alcohol)==LOW){lcd.setCursor(13,1);
lcd.print("ALC");digitalWrite(relay,HIGH);}
else{lcd.setCursor(13,1);
lcd.print("   ");digitalWrite(relay,LOW);}
}

unsigned long ping1()
{
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
```

```
  delayMicroseconds(20);
  digitalWrite(trig, LOW);
  digitalWrite(echo, HIGH);
  echoVal = pulseIn(echo, HIGH);
  ultrasoundValue = microsecondsToCentimeters(echoVal);
  return ultrasoundValue;
}

long microsecondsToCentimeters(long microseconds)
{
  return microseconds / 29 / 2;
}

void gsm_init()
{
  lcd.setCursor(0,0);
  lcd.print("GSM INITIALISE..");
  Serial2.println("AT"); // ATTENTION COMMAND
  Serial2.flush();
  delay(1000);
  Serial2.println("AT+CMGF=1"); // ATTENTION WITH COMMAND
MESSAGE FORMAT ( 1 = TEXT MODE)
```

```
  Serial2.flush();
  delay(1000);
  Serial2.println("AT+CNMI=2,2,0,0,0"); // ATTENTION WITH COMMAND NEW MESSAGE INDICATION
  Serial2.flush();
  delay(1000);
  lcd.setCursor(0,0);
  lcd.print(" GSM INIT OVER  ");
  delay(1000);
  lcd.clear();
}

void gsm_num(unsigned char *p)
{
  int j;
  Serial2.print("AT+CMGS=");
  Serial2.print("\"");
  for(j=0;j<10;j++)
  {
    Serial2.write(p[j]); // mobile number
    delay(50);
  }
```

```
  Serial2.println("");
  delay(500);
}

void gsm_send()
{
  delay(1000);
  Serial2.write(0x1A); // message send
  delay(1000);delay(1000);
  Serial2.flush();
}

void enter()
{
  Serial2.write(0x0D);
  Serial2.write(0x0A);
}



void Lcd_Decimal2(int x, int y, int val)
{
  int Lcd_hr,Lcd_t,Lcd_o;
```

```
  Lcd_hr=val%100;

  Lcd_t=Lcd_hr/10;

  Lcd_o=Lcd_hr%10;


  lcd.setCursor(x,y);

  lcd.write(Lcd_t+0x30);

  lcd.setCursor(x+1,y);

  lcd.write(Lcd_o+0x30);

}


void Lcd_Decimal3(int x, int y, int val)

{

  int Lcd_h,Lcd_hr,Lcd_t,Lcd_o;


  Lcd_h=val/100;

  Lcd_hr=val%100;

  Lcd_t=Lcd_hr/10;

  Lcd_o=Lcd_hr%10;


  lcd.setCursor(x,y);

  lcd.write(Lcd_h+0x30);
```

```
  lcd.setCursor(x+1,y);
  lcd.write(Lcd_t+0x30);
  lcd.setCursor(x+2,y);
  lcd.write(Lcd_o+0x30);
}

void Lcd_Decimal4(int x, int y, int val)
{
  int Lcd_th,Lcd_thr,Lcd_h,Lcd_hr,Lcd_t,Lcd_o;

  val = val%10000;
  Lcd_th=val/1000;
  Lcd_thr=val%1000;
  Lcd_h=Lcd_thr/100;
  Lcd_hr=Lcd_thr%100;
  Lcd_t=Lcd_hr/10;
  Lcd_o=Lcd_hr%10;

  lcd.setCursor(x,y);
  lcd.write(Lcd_th+0x30);
  lcd.setCursor(x+1,y);
  lcd.write(Lcd_h+0x30);
```

```
  lcd.setCursor(x+2,y);
  lcd.write(Lcd_t+0x30);
  lcd.setCursor(x+3,y);
  lcd.write(Lcd_o+0x30);
}

void Serial_Decimal3(int val)
{
  int Lcd_h,Lcd_hr,Lcd_t,Lcd_o;

  Lcd_h=val/100;
  Lcd_hr=val%100;
  Lcd_t=Lcd_hr/10;
  Lcd_o=Lcd_hr%10;

  Serial.write(Lcd_h+0x30);
  Serial.write(Lcd_t+0x30);
  Serial.write(Lcd_o+0x30);
}
```

CHAPTER 4

4 RESULTS AND DISCUSSIONS

4.1 WORKING MODEL

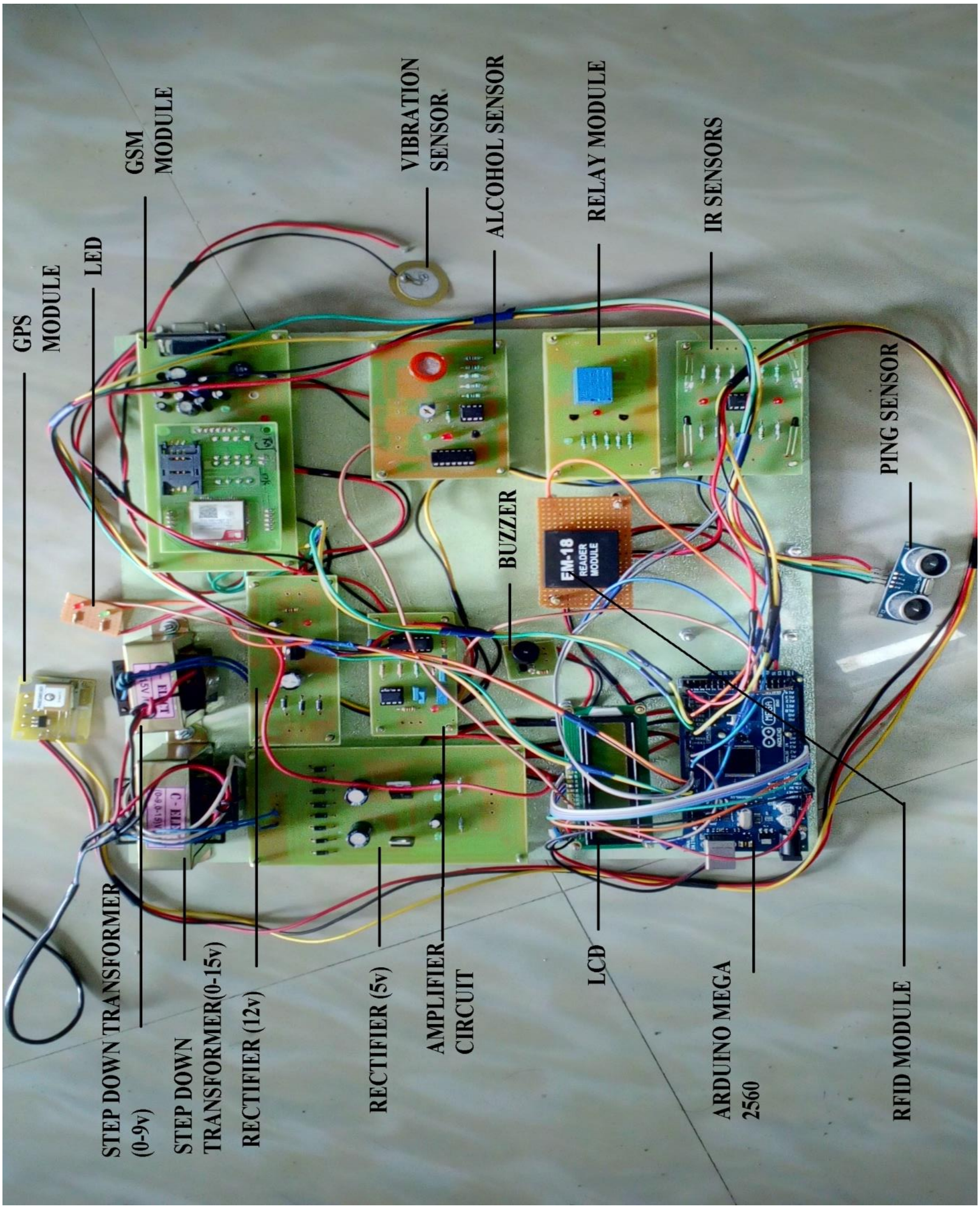


Figure 24: Working model

4.2 WORKING FLOW CHART

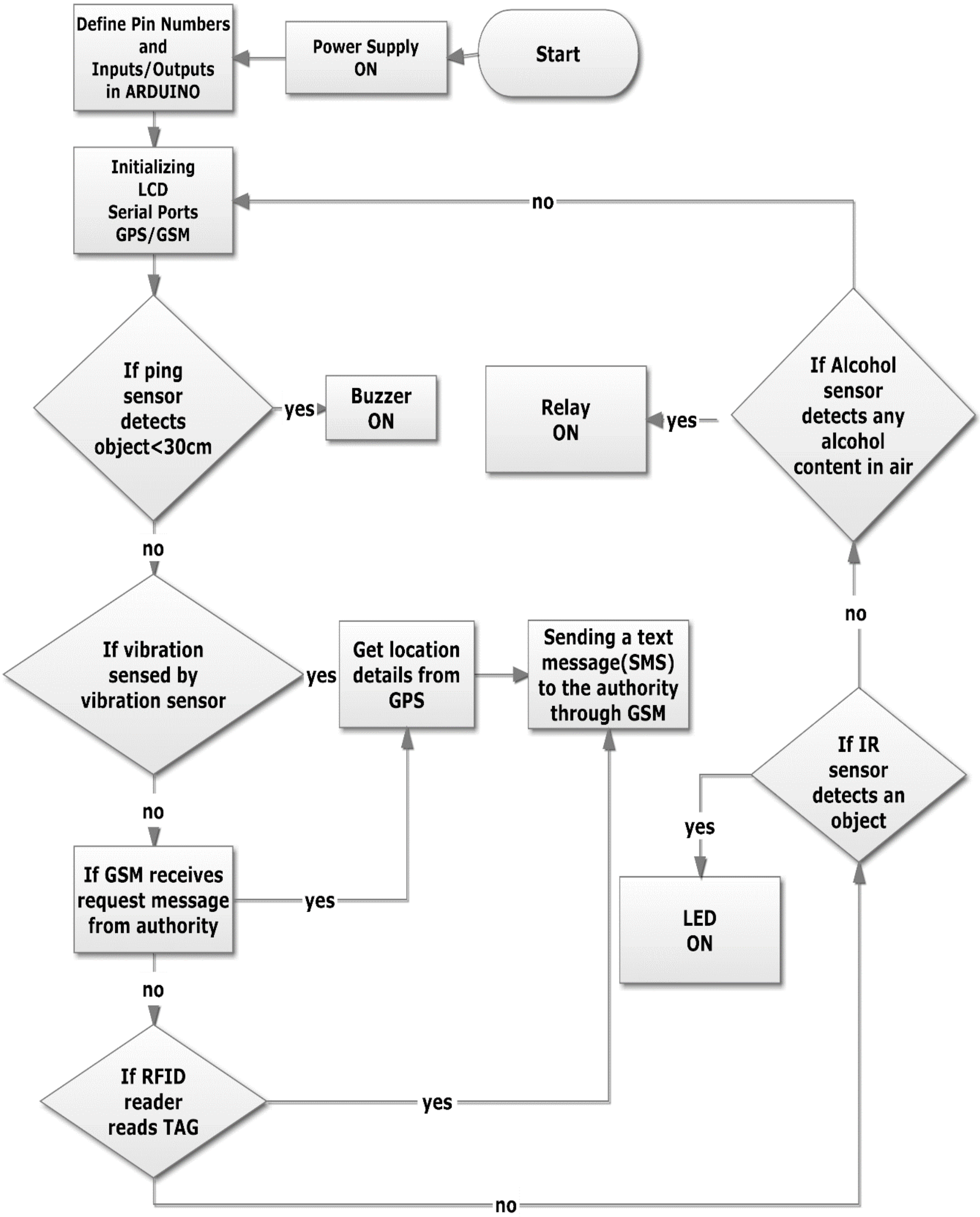


Figure 25: Flow chart

4.3 WORKING ALGORITHM

STEP 1: Start.

STEP 2: Switch ON the 12volt and 5volt power supplies.

STEP 3: Define pin numbers and input/output signal flow directions of all components in Arduino.

STEP 4: Initialize LCD.

STEP 5: Initialize serial ports of Arduino to enable serial buffers.

STEP 6: Enable & Initialize the GPS and GSM modules.

STEP 7: If ultrasonic sensor detects objects < 30cm to the wheels, turn ON the alarm to alert the bus driver.

STEP 8: If any vibrations detected by vibration sensor, get the location values from GPS and send it to the authority via GSM module.

STEP 9: If GSM module receives any location request message from authority, location details are obtained by GPS module and the location values are sent to the authority via GSM module.

STEP 10: If RFID reader reads any child's tag, the arrival message is sent to the corresponding parent.

STEP 11: If IR sensor detects any object on the foot board, LED is turned ON for alert the bus driver.

STEP 12: If alcohol sensor detects any alcohol concentration in air which is nearer to the bus driver seat, relay will turned ON. So the bus ignition system is turned OFF.

STEP 13: Loop of the above process is moved to step 3.

STEP 14: Stop.

4.4 TESTED OUTPUTS

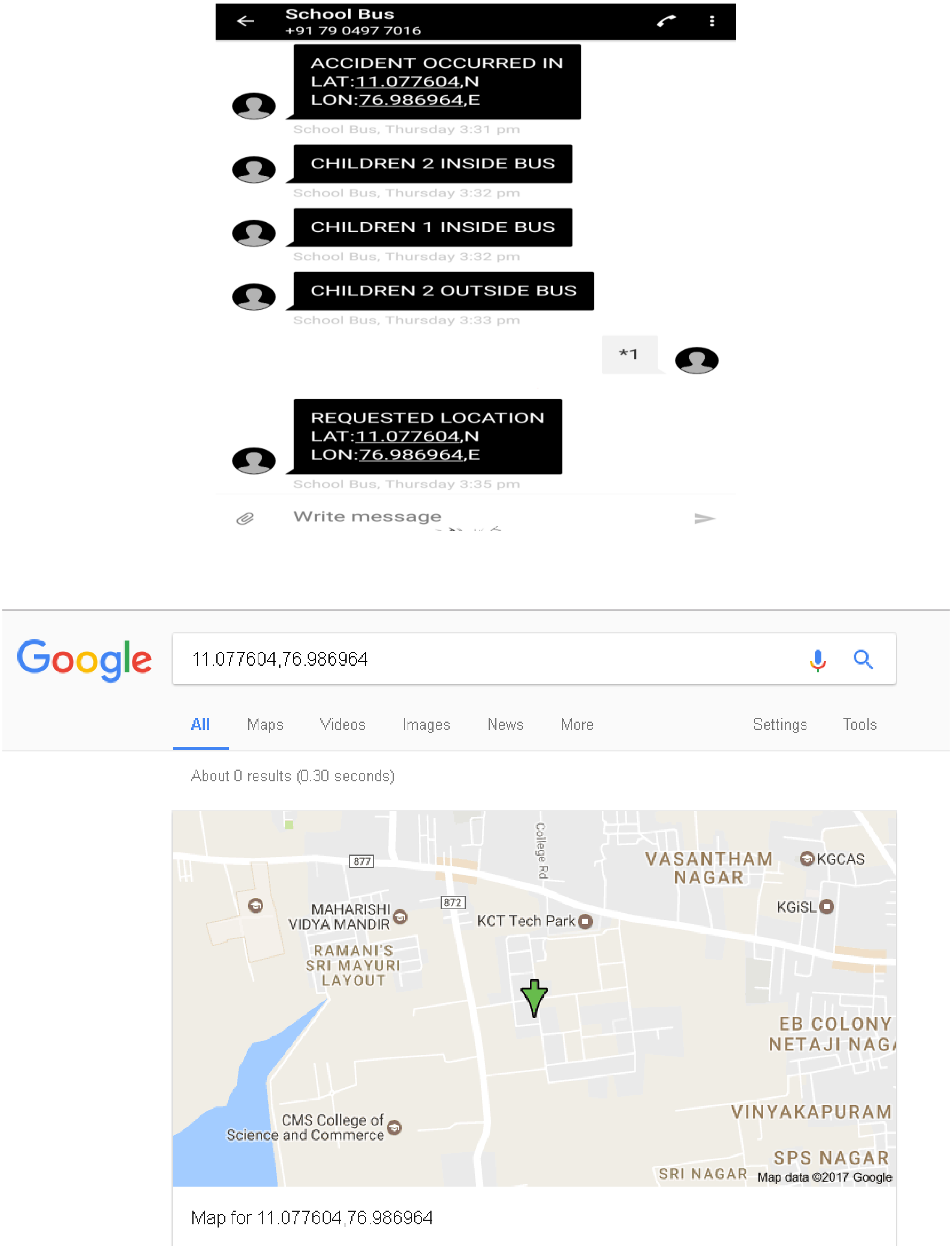


Figure 26

4.5 SIGNIFICANCE OF THE PROJECT

The aim of this project is to develop a Bus Safety System which provides the details of entry and exit of the student from buses using RFID and GSM technologies. The proposed system provides a facility to track the exact location of the bus using RFID and GPS in a cost effective way and also provides some safety and security features. So this could be implemented even in rural schools. Such systems must be installed in order to reduce the number of abduction taking place.

4.6 ENHANCEMENT OF THE PROJECT

Our Future Scope includes the real time implementation of the proposed system with the additional features like route analysis, wireless audio/video transmission from the bus to the school, parking management system for the school buses and to validate the proposed system in real time on a school bus in an effective manner.

4.7 CONCLUSION

Combining RFID, GPS and GSM advances for safety and security reason is incredibly vital. Presently, as a result of increase in mishaps like kids getting out at wrong stations or children getting missed out at the bus may question the safety of school going children. This proposed system shows that RFID based school bus tracking technology is a feasible alternative for supervising and tracing the pupils during their drive to and from school. Additionally, the expense associated with tagging of material is moderately low. Also the drunk and drive prevention system and the accident alert system play a major role to help the children commute safely. In this manner the system is capable of notifying parents/guardians via text message once the child enters/leaves the varsity, enabling parents/guardians to trace the bus, helping smooth and safer rides to the various destinations.

REFERENCES

[1] Shah, Shraddha, and Bharti Singh. "RFID based school bus tracking and security system." *Communication and Signal Processing (ICCSP), 2016 International Conference on*. IEEE, 2016.

[2] Ramanath, T. Shyam, A. Sudharsan, and U. Felix Udhayaraj. "Drunken driving and rash driving prevention system." *Mechanical and Electrical Technology (ICMET), 2010 2nd International Conference on*. IEEE, 2010.

[3] Lee, SeokJu, GirmaTewolde, and JaerockKwon. "Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application." *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014.

[4] India's missing children, by the number. Link available at: http: //blogs.wsj.com

[5] Children injured after drunk driver rams school bus into railing of bridge.See more at: http:// indianexpress.com

[6] 10 year old boy crushed under the school bus wheels. More at: http://www.business-standard.com

[7] Driver kidnaps children who waiting for the school bus. More at: http://timesofindia.indiatimes.com