A
Project Report
on
GPS based Vault with Integrated Fingerprint Scanner and Keypad Lock

Submitted By:

Gunjan K. Khut
Sneha D. Awasthi

B.E. [Electronics & Telecommunication]

Guided By:

Prof. T. H. Jaware

Department of Electronics and Telecommunication

R. C. Patel Institute of Technology,
Shirpur-425405

2013-14

Shirpur Education Society's

R. C. Patel Institute of Technology, Shirpur, Dist-Dhule

CERTIFICATE

This is to certify that project report on project entitled ***"GPS based Vault with Integrated Fingerprint Scanner and Keypad Lock"****, being submitted by* ***Khut Gunjan Kantilal, Awasthi Sneha D.*** *to North Maharashtra University, Jalgaon for the fulfilment of Project, is a record of bonafide work carried out by them under my supervision and guidance during year 2013-2014.*

Date:

Place: Shirpur

Project Guide | **Coordinator**

Head of Department | **Principal**

Abstract

The Vault lock is controlled using GPS, Fingerprint scanner and Keypad lock security. Hence can be called as Three Stage secured Vault.

The GPS module error range is 5 meters so the vault will be only opened if the latitudes of the receiving end are matched with the predefined latitudes on which it is to be opened if it is correct then only the vault is defined to open.
For E.g. If we want to transfer any valuable documents in the vault to any place and to make sure that the vault is not opened in the way to the destination, we can use this vault for this purpose.

Here we will set the latitudes of the destination and then will head for the journey and as it is controlled using GPS and will be checked for fingerprint only if the latitudes match after successful matching of the fingerprint the keypad is enabled to enter the security code for opening the vault so therefore the security for the documents is maintained till it reaches to the concerned receiver. Also receiver has other security to use i.e. keypad or Fingerprint Scanner to open the vault.

We will be using Atmega2560 controller to interface the modules of Fingerprint Scanner, Matrix keypad and GPS to control the lock of the vault.

Acknowledgement

It is a privilege for us to have been associated with **Prof. T. H. Jaware**, our guide, during this project work. We have been greatly benefited by his valuable suggestions and ideas. It is with great pleasure that we express our deep sense of gratitude for his valuable guidance, constant encouragement and patience throughout this work.

We owe a debt of gratitude to **Prof. P. R. Bhole**, Project coordinator for entertaining our queries at all times of the day.

We express our gratitude to **Prof. Dr. P. J. Deore**, Head, Department of Electronics & Telecommunication for his constant encouragement, co-operation, and support.

We take this opportunity to thank all our classmates for their company during the course work and for useful discussion we had with them.

We would be failing in our duties if we do not make a mention of our family members including our parents for providing moral support, without which this work would not have been completed.

Gunjan K. Khut

Sneha D. Awasthi

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

This project is a Three stage secured vault. This concept enables an economic assembly in any existing vault, the lock of the vault will be in control of the sender. Only the sender can decide who shall be the receiver of the documents which are stored in the vault because the vault will not open until the predefined security keys are not provided to the vault. Therefore for the receiver to receive the documents which are locked inside the vault the security key set by the sender is necessary.

1.2 Input Through User - Transmitter

1.2.1 GPS Technology

The full-form of GPS is Global Positioning System. This is a satellite based navigation system. A GPS receiver can record its longitude, latitude, and altitude related information with the help of numerous satellites. These satellites broadcast punctuated time radio signals which can be received by the GPS receivers. By integrating information from various satellites, very precise information can be extracted (latitude, longitude and altitude related). U.S. Department of Defense owns GPS; however, it is free to be used around the world.

The user has to enter the desired combinations of longitude, latitude, and altitude of the location where the vault is to be accessed. The controller will ask for further detail i.e Fingerprint scanner and password only after successful match of the above combinations.

All these processes are advanced by modifying the mechanism in farming which works automatically without man power requirement. The small machine could be assembled from existing mass produced components without the need of specialized design and tooling. Also by studying social aspects which shows that public are ready to use the small intelligent machine in food production by level of interesting observed by media and when having demonstrated. Because of small autonomous machine, its liability and insurance will be lot easier. Also it reduces the required input energy compared with tractors.

By considering the work environment in farming field, like floppy and bumpy road, four wheeler driving system is adopted to robot for improving driving capacity and loading capacity. These four wheels are differentially steered and driven by four current motor directly.

1.2.2 Fingerprint Scanner

Finger print scanner R305 This is a fingure print sensor module with TTL UART interface for direct connections to microcontroller UART or to PC through MAX232 / USB-Serial adapter. The user can store the finger print data in the module and can configure it in 1:1 or 1:N mode for identifying the person. The FP module can directly interface with 3v3 or 5v Microcontroller. A level converter (like MAX232) is required for interfacing with PC serial port.
Optical biometric fingerprint reader with great features and can be embedded into a variety of end products, such as access control, attendance, safety deposit box, car door locks.

1.2.3 Matrix Keypad 4x4

This 16-button keypad provides a useful human interface component for micro-controller projects. Convenient adhesive backing provides a simple way to mount the keypad in a variety of applications. The Keypad 4x4 features a total of 16 buttons in Matrix form. This is a membrane keypad with no moving parts. It has a nice overlay depicting a telephone type keypad with additional four functional buttons. A female 8-pin berg connector is provided for interfacing it with your micro-controller circuits.

1.3 Microcontroller

A microcontroller is the heart of the vault. A microcontroller is a programmable device that can be used to control the vault, to perform any arithmetic and logic operations. The difference between a microcontroller and a microprocessor is the availability of internal memory to store the programme code and it can function as a standalone controller. There are several self sufficient microcontroller boards available of which Arduino is the best suited for the following functions. Arduino is an open-source single-board microcontroller, descendant of the open-source Wiring platform, designed to make the process of using electronics in multidisciplinary projects more accessible. The hardware consists of a simple open hardware design for the Arduino board with an Atmel AVR processor and on-board input/output support. The software consists of a standard programming language compiler and the boot loader that runs on the board. Arduino hardware is programmed using a Wiring-based language (syntax and libraries), similar to C++ with some slight simplifications and modifications, and a Processing-based integrated development environment. This is a simple controller which can be easily incorporated with the GPS module, Fingerprint module and the Keypad to control the electronic lock. This is done by using a simple coding technique similar to the C++ program coding. It is also the cheapest and most widely used controller. As we employ only simple operations this controller is sufficient. The controller takes the input from the GPS module and check the coordinated if matched then will check for the input from fingerprint scanner as soon as the fingerprint is matched, it will take the input from keypad and as the security code matches it will open the electronic lock of the vault. The main reason for employing this controller is its low power consumption and ease in the coding. The supply is drawn from the batteries which are used to run the modules.

1.4 Output

1.4.1 LCD 16x4

This is a high quality 16 character by 4 line intelligent display module, with back lighting, works with almost any microcontroller. It displays the latitude and longitude of the destination which are set. Then will display to keep the finger to scan for the fingerprint lock and atlast will display Enter the security code which you want to set for the vault and when the vault will be accessed all these will be asked by displaying on the LCD for it.

1.4.2 Electronic Lock

An electronic lock (or electric lock) is a locking device which operates by means of electric current. Electric locks are sometimes stand-alone with an electronic control assembly mounted directly to the lock. More often electric locks are connected to an access control system. The advantages of an electric lock connected to an access control system include: key control, where keys can be added and removed without re-keying the lock cylinder; fine access control, where time and place are factors; and transaction logging, where activity is recorded.

1.4.3 Relay

A relay is an electrical switch that uses an electromagnet to move the switch from the off to on position instead of a person moving the switch. It takes a relatively small amount of power to turn on a relay but the relay can control something that draws much more power. Ex: A relay is used to control the air conditioner in your home. The AC unit probably runs off of 220V AC at around 30A. That's 6600 Watts! The coil that controls the relay may only need a few watts to pull the contacts together. This is the schematic representation of a relay. The contacts at the top are normally open (i.e. not connected). When current is passed through the coil it creates a magnetic field that pulls the switch closed (i.e. connects the top contacts). Usually a spring will pull the switch open again once the power is removed from the coil.

Chapter 2

Basic Concept and Literature Survey

2.1 Basic Block Diagram

The basic block diagram consist of this project is as follows:

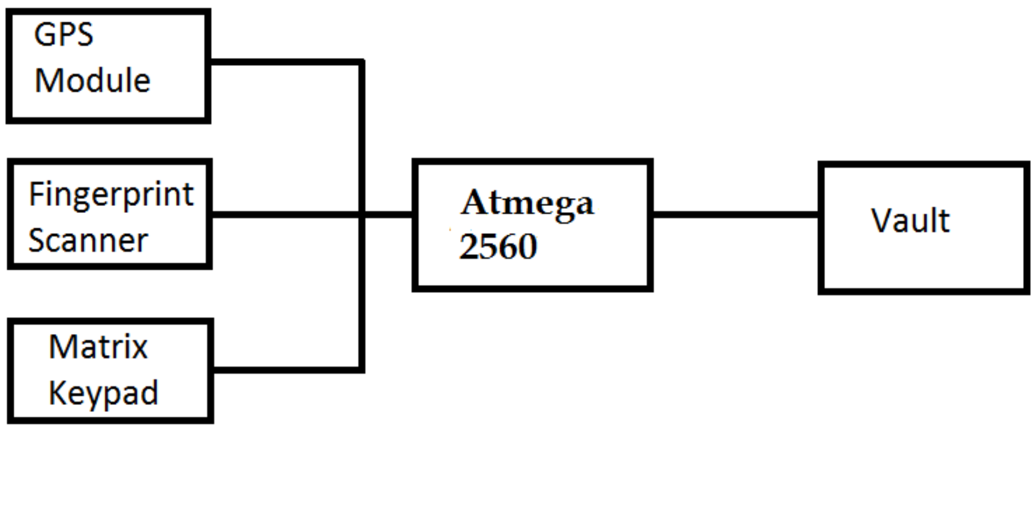

Figure 2.1: Basic Block Diagram

2.1.1 GPS module

Global Positioning System (GPS) satellites broadcast signals from space that GPS receivers, use to provide three-dimensional location (latitude, longitude and altitude) plus precise time. GPS receivers provides reliable positioning, navigation and timing services to worldwide users on a continuous basis in all weather, day and night, anywhere on or near the Earth.

Sunroms ultra-sensitive GPS receiver can acquire GPS signals from 65 channels of satellites and output position data with high accuracy in extremely challenging environments and under poor signal conditions due to its active antenna and high sensitivity. The GPS receivers -160dBm tracking sensitivity allows continuous position coverage in nearly all application environments.

The output is serial data of 9600 baud rate which is standard NMEA 0183 v3.0 protocol offering industry standard data messages and a command set for easy interface to mapping software and embedded devices.

2.1.2 Microcontroller

The microcontroller is the heart of the project. The main function of the microcontroller is to maintain the proper functionality of all the other components. It is an intelligent device which knows what should happen for particular set of inputs. It processing and maintenance is based upon the program loaded in it. In this project the controller takes the input from the GPS module and check the coordinated if matched then will check for the input from fingerprint scanner as soon as the fingerprint is matched, it will take the input from keypad and as the security code matches it will open the electronic lock of the vault.

The micocontroller used in this project is ATmega 2560. It is used with a developer board known as Arduino Mega. Arduino is freeware corporation who develops hardware as well as software.

2.1.3 Fingerprint Scanner

The fingerprint sensor is combination of R305 FP+PIC MCU board that can read different fingerprints and store in its own flash memory. The sensor can perform three functions namely Add (Enroll), Empty Database or Search Database and return the ID of stored fingerprint. Any of three functions can be called simply by making the pin low of the sensor or pressing onboard three switches. The response is either error or ok which is indicated by onboard LED. The response is also returned as single serial data byte. The return byte is a valid ID or error code. The response byte is a single byte at 9600 bps thus making whole sensor very easy to use. We have provided indicating LEDs and function switch already so its ready to use when you receive it. Just give power and start using the sensor using onboard switches. Then you can move on making external application using these functions.

2.1.4 4x4 Matrix Keypad

This 16-button keypad provides a useful human interface component for micro controller projects. The Keypad 4x4 features a total of 16 buttons in Matrix form. This is a membrane keypad with no moving parts. The keypad is used to give the input to the controller. In this project the key pad is necessary as the security code for the vault will be give by the keypad. The keypad has total of 16 buttons out of which 10 (0 to 9) will be used for the input of security code and one key will be for "OK" and another will be for "clear". There are also alphabets which can be used in between the security code to make the code a strong one.

Matrix keypads use a combination of four rows and four columns to provide button states to the host device, typically a micro controller. Underneath each key is a push button, with one end connected to one row, and the other end connected to one column.

In order for the micro controller to determine which button is pressed, it first needs to pull each of the four columns (pins 1-4) either low or high one at a time, and then poll the states of the four rows (pins 5-8). Depending on the states of the columns, the micro controller can tell which button is pressed. For example, say your program pulls all four columns low and then pulls the first row high. It then reads the input states of each column, and reads pin 1 high. This means that a contact has been made between column 4 and row 1, so button A has been pressed. The security code input to the vault is given through the keypad.

2.1.5 Electronic Lock

The most basic type of electronic lock is a magnetic lock (commonly called a mag lock). A large electro-magnet is mounted on the vault frame and a corresponding armature is mounted on the vault. When the magnet is powered and the vault is closed, the armature is held fast to the magnet. Mag locks are simple to install and are very attack resistant. One drawback is that improperly maintained mag locks looses its magnetic power and also that one must unlock the mag lock to both open and close the vault. The most prevalent form of electronic lock is that using a numerical code for authentication; the correct code must be entered in order for the lock to deactivate. Such locks typically provide a keypad, and some feature an audible response to each press. Combination lengths are usually between 4 and 6 digits long.

2.2 Application

The main application of this Vault is to improved the quality of Security for many applications. The security is the basic requirement as military and Bank level. also there are various other applications where this vault can be used for securing various expensive items or products

The applications of this Vault in various fields are:-

- Military applications such as.
  - confidential papers
  - Arms, Weapons and Explosives
  - cash
- Transfer money from Banks to Banks or to ATMs.
- Transfer of jewellery from jewellery shops.
- Transfer of any expensive items.
- Secure the weapons while deals.
- While travelling or sending any important document to any place.

2.3 Inspiration

The main inspiration of this theme are different types of security vaults. Some are controlled by fingerprint scanner and some are controlled by keypad security. but we have planned to make a security vault which both these types of security levels in addition to this we have add GPS to the vault.

In our project, we will set the latitudes of the destination and then will head for the journey and as it is controlled using GPS and will check for fingerprint only if the latitudes match after successful matching of the fingerprint the keypad is enabled to enter the security code for opening the vault so therefore the security for the documents is maintained till it reaches to the concerned receiver. Also receiver has other security to use i.e. keypad or Fingerprint Scanner to open the vault.

Chapter 3

Component Specifications

3.1 Atmega 2560

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 64 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 mega 2560.

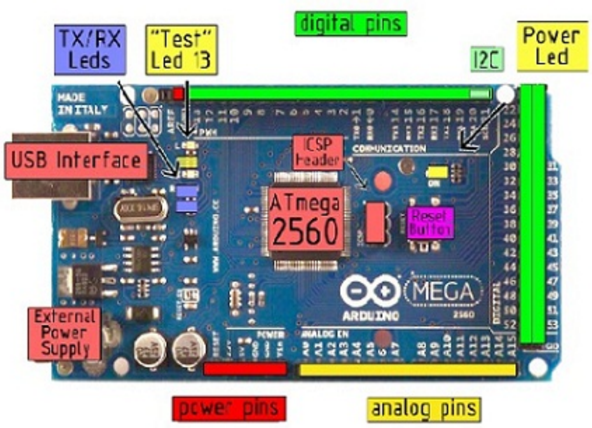

Figure 3.1: Arduino Mega 2560

3.1.1 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

3.1.2 Power

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 center-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 Atmega8U2 programmed as a USB-to-serial converter. The power pins are as follows:

1. 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.
2. 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. 3V3 : A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
4. GND : Ground pins.

3.1.3 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 I2C (TWI) communication using the Wire library (documentation on the Wiring website). 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 provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

1. AREF. Reference voltage for the analog inputs. Used with analogReference().
2. 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.

3.1.4 Physical Characteristics and Shield Compatability

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins. The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. Please note that I2C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).

3.2 GPS module S1315RL

S1315RL ROM-based Low-Power High-Performance Low-Cost 65 Channel SMD GPS Module. The S1315RL is a small form factor GPS module solution intended for a broad range of OEM products, where fast and easy system integration and minimal development risk is required. The user only need to provide DC power of 3.0V to 3.6V and GPS signal; the S1315RL will output navigation solution in standard NMEA-0183 protocol format.

The S1315RL features 65 channel GPS receiver with fast time to first fix and improved -148dBm cold start sensitivity. The superior cold start sensitivity allows it to acquire, track, and get position fix autonomously in difficult weak signal environment. The receivers -165dBm tracking sensitivity allows continuous position coverage in nearly all application environments.

The high performance search engine is capable of testing 8,000,000 time-frequency hypotheses per second, offering industry-leading signal acquisition and TTFF speed. Measuring 13mm x 15mm, the S1315RL contains integrated LNA, SAW filter, 0.5ppm TCXO, 65 channel positioning engine, RTC crystal, and low-leakage backup supply LDO regulator. The RF section has cascaded noise figure of 1.2dB, allowing passive antenna operation without extra external LNA.

The receiver is optimized for applications requiring high performance, low power, and low cost; suitable for a wide range of OEM configurations including mobile phone, PND, asset tracking, and vehicle navigation products. The metal RF shielding provides protection and allows standard surface mount device pick-and-place process in fully automated assembly process; enabling high-volume, very cost-efficient production. The S1315RL is available in tape-and-reel form.

3.2.1 Block Diagram

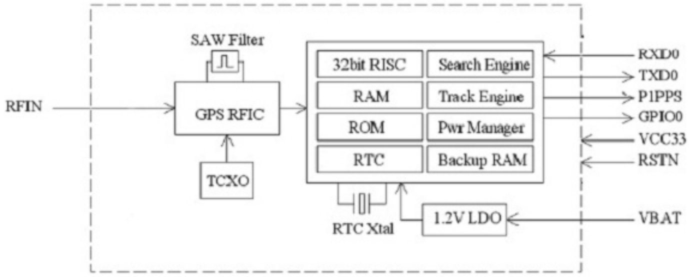

Figure 3.2: GPS Module Block Schematic

3.2.2 Features

1. Perform 8 million time-frequency hypothesis testing per second
2. Open sky hot start 1 sec
3. Open sky cold start 29 sec
4. Cold start sensitivity -148dBm
5. Tracking sensitivity -165dBm
6. Multipath detection and suppression
7. Jamming detection and mitigation
8. Accuracy 2.5m CEP
9. Maximum update rate 20Hz
10. Tracking current 23mA
11. Supports active and passive antenna
12. Operating temperature -40^{o} to +85^{o}C
13. RoHS compliant

3.2.3 Applications

1. PND
2. MID / Netbook
3. Smart-Phone

4. Geo-Tagging
5. Automatic Vehicle Location
6. Personal Tracking

3.2.4 Technical Specifications

1. Receiver Type: L1 C/A code, 65-channel Venus 6 engine
2. Accuracy Position : 2.5m CEP and Velocity 0.1m/sec and Time 60ns
3. Startup Time : 1 second hot start under open sky $<$ 29 second warm start under open sky (average) 29 second cold start under open sky (average)
4. Reacquisition : 1s
5. Sensitivity : -148dBm cold start -165dBm tracking
6. Multi-path Mitigation : Advanced multi-path detection and suppression
7. A-GPS : Support PromptFix AGPS
8. Update Rate : Supports 1 / 2 / 4 / 5 / 8 / 10 / 20 Hz update rate (1Hz default)
9. Dynamics : 4G (39.2m/sec2)
10. Operational Limits : Altitude $<$ 18,000m or velocity $<$ 515m/s
11. Serial Interface : 3.3V LVTTL level
12. Protocol : NMEA-0183 V3.01 GPGGA, GPGLL, GPGSA, GPGSV, GPRMC, GPVTG*1 9600 baud, 8, N, 1
13. Datum Default : WGS-84 User definable
14. Input Voltage : 3.3V DC +/-10%
15. Input Current : 23mA tracking
16. Dimension : 15mm L x 13mm W Weight: 1.5g
17. Operating Temperature : -40^{o}C to +85^{o}C
18. Storage Temperature : -55^{o}C to +100^{o}C
19. Humidity : 5 to 95

3.3 Fingerprint Scanner R305

This is a fingure print sensor module with TTL UART interface for direct connections to microcontroller UART or to PC through MAX232 / USB-Serial adapter. The user can store the finger print data in the module and can configure it in 1:1 or 1: N mode for identifying the person.The FP module can directly interface with 3v3 or 5v Microcontroller. A level converter (like MAX232) is required for interfacing with PC serial port.

Optical biometric fingerprint reader with great features and can be embedded into a variety of end products, such as: access control, attendance, safety deposit box, car door locks.

3.3.1 Features

- Integrated image collecting and algorithm chip together, ALL-in-One.
- Fingerprint reader can conduct secondary development, can be embedded into a variety of end products.
- Low power consumption, low cost, small size, excellent performance.
- Professional optical technology, precise module manufacturing techniques.
- Good image processing capabilities, can successfully capture image up to resolu tion 500 dpi.

Figure 3.3: Fingerprint Scanner

3.3.2 Types of function

There are namely three functions you can call for the fingerprint sensor:

- **Add (Enroll) Function:** Adds a fingerprint to database and return a byte of newly added ID. Return values are from 0x00 to 0xFE. In case of error like no finger placed, return code is 0xFF. Here 0xFF means error executing function
- **Search Function:** When a finger is put and search function is called, it returns a matching ID if found in its existing memory. Return values are from 0x00 to 0xFE. In case of error like no finger placed, return code is 0xFF. Here 0xFF means error executing function.
- **Empty Function:** When you wish to empty all fingerprint data stored on sensor you can use this function. After executing this function, you will get 0xCC as OK or 0xFF in case of error.

3.4 Matrix Keypad

This 16-button keypad provides a useful human interface component for micro controller projects. Convenient adhesive backing provides a simple way to mount the keypad in a variety of applications.

3.4.1 Key Specifications

- Maximum Rating: 24 VDC, 30 mA
- Life Expectancy: 1 million closures; Bounce time: 5 ms
- Insulation Resistance: 100M Omh, @ 100V
- Dielectric Withstand: 250VRms (@ 60Hz, 1min)
- Interface: 8-pin access to 4x4 matrix
- Operating temperature: 32 to 122 ^{o}F (0 to 50^{o}C).
- Dimensions: Keypad: 2.7 x 3.0 in (6.9 x 7.6 cm)

  Cable: 0.78 x 3.5 in (2.0 x 8.5 cm)

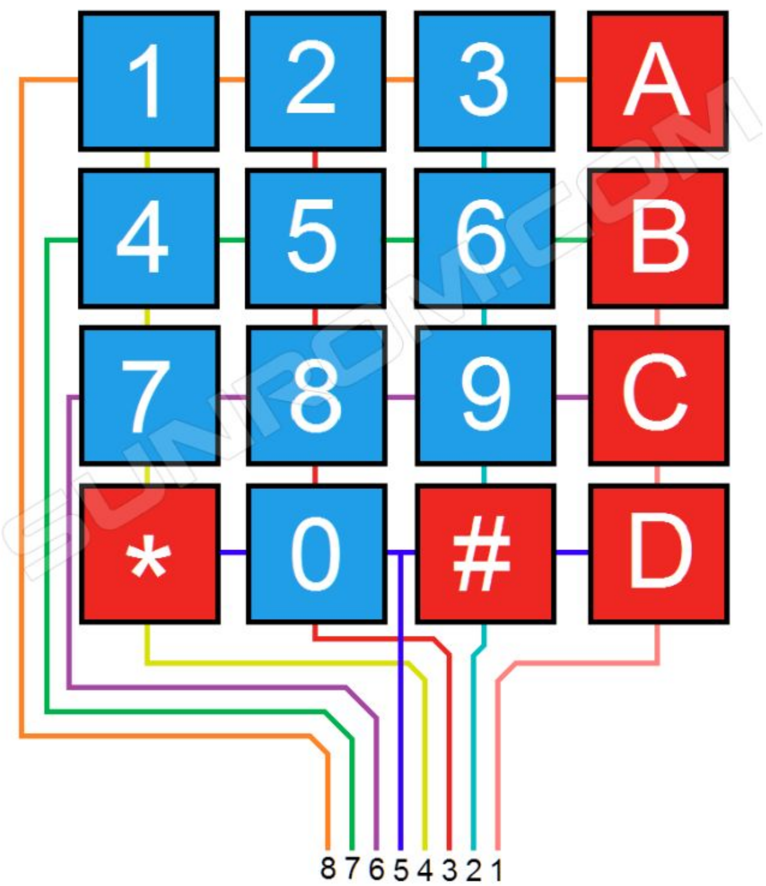

Figure 3.4: Matrix keypad

3.5 Relay

A relay is an electrical switch that uses an electromagnet to move the switch from the off to on position instead of a person moving the switch. It takes a relatively small amount of power to turn on a relay but the relay can control something that draws much more power. A relay is used to control the air conditioner in your home. The AC unit probably runs off of 220VAC at around 30A. That's 6600 Watts! The coil that controls the relay may only need a few watts to pull the contacts together. This is the schematic representation of a relay. The contacts at the top are normally open (i.e. not connected). When current is passed through the coil it creates a magnetic field that pulls the switch closed (i.e. connects the top contacts). Usually a spring will pull the switch open again once the power is removed from the coil.

A relay coil is not only an electromagnet but it's also an inductor. When power is

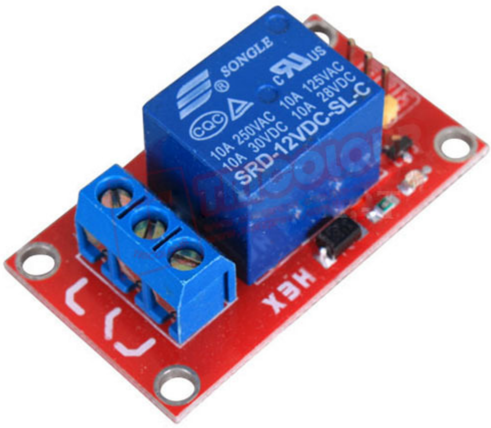

Figure 3.5: Relay

applied to the coil the current in the coil builds up and levels off at its rated current (depends on the DC resistance of the coil, I = V/R). Some energy is now stored in the coil's magnetic field (E = 05LI2). When the current in the coil is turned off this stored energy has to go somewhere. The voltage across the coil quickly increase trying to keep the current in the coil flowing in the same direction (V = Ldi/dt). This voltage spike can reach hundreds or thousands of volts and can damage electronic parts. By adding a flyback diode the current has a path to continue flowing through coil until the stored energy is used up. The diode also clamps the voltage across the coil to about 0.7V protecting the electronics. The stored energy dissipates quickly in the diode (E = V*I*t). The current stops flowing and the relay turns off. The diode should be able to handle the coil current for a short time and switch relatively fast. Note: A resistor or zener diode can be placed in series with the diode to use up the stored energy quicker. This increases the amplitude of the voltage spike above 0.7V but the energy is used up quicker (i.e. the voltage spike won't last as long). Usually it doesn't matter if the relay takes 1ms or 100ms to turn off.

Chapter 4

Interconnections

In this chapter we will see the interconnection as well as interfacing of all the components used with each other.

4.1 Interfacing of GPS Module with Arduino ATmega2560 and LCD

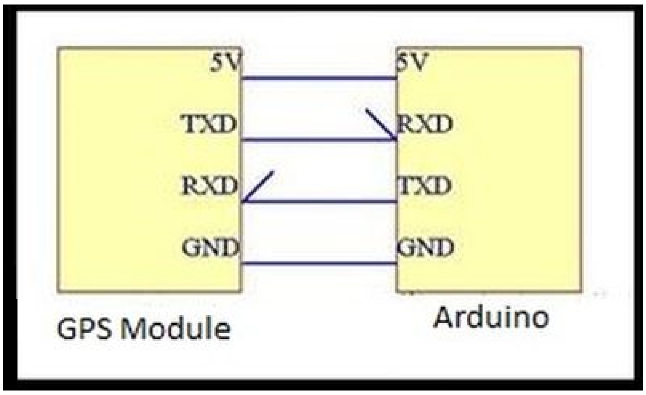

Figure 4.1: Interfacing of GPS module with Arduino

As shown in the figure the Vcc pin of the GPS module must connect to the Vcc pin of the Arduino board. Then the Gnd pin of the GPS module must connect to the Gnd pin of the Arduino board. Then the Tx pin of the GPS module must connect to the Rx pin of the Arduino board. Then the Rx pin of the GPS module must connect to the Tx pin of the Arduino board.

4.1.1 LCD Pin Configuration

Pin 1: Vss : Function : Ground

Pin 2: Vdd : Function : +3V to +5V.

Pin 3: Vo : Function : Contrast/Brightness adjustment.

Pin 4: RS : Function : H/L Register select signal.

Pin 5: R/W : Function : Read/Write Signal.

Pin 6: EN : Function : H->L Enable Signal.

Pin 7: DB0 : Function : H/L Data Bus Line.

Pin 8: DB1 : Function : H/L Data Bus Line.

Pin 9: DB2 : Function : H/L Data Bus Line.

Pin 10: DB3 : Function : H/L Data Bus Line.

Pin 11: DB4 : Function : H/L Data Bus Line.

Pin 12: DB5 : Function : H/L Data Bus Line.

Pin 13: DB6 : Function : H/L Data Bus Line.

Pin 14: DB7 : Function : H/L Data Bus Line.

Pin 15: A/Vee : Function : 4.2V.

Pin 16: K : Function : Power Supply for B/L.

4.1.2 GPS Pin Configuration

Pin 1: Vss : Function : Ground

Pin 2: Vdd : Function : +5V.

Pin 3: Tx : Function : Transmitter

Pin 4: Rx : Function : Receiver.

4.2 Arduino Mega2560 & Fingerprint Scanner

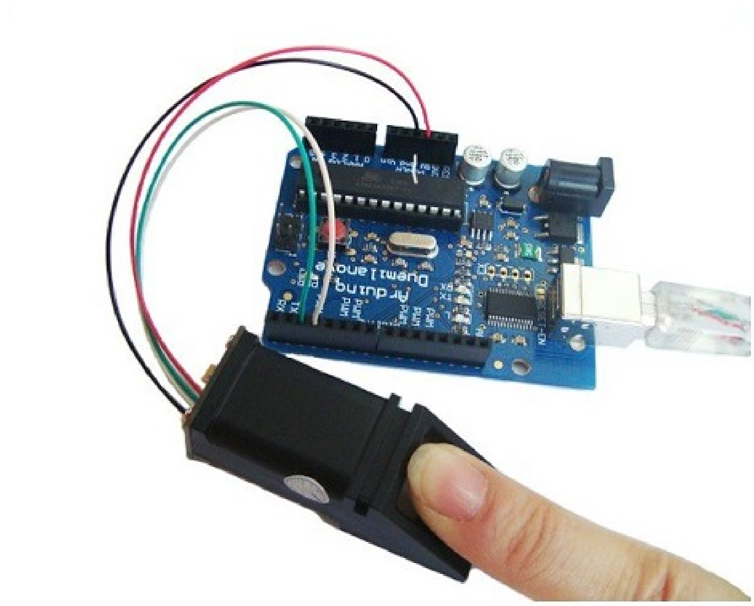

Figure 4.2: Interfacing of Arduino with Fingerprint Scanner

As shown in the figure the Vcc pin of Arduino is connected to the Vcc pin of Fingerprint scanner. and Gnd pin of Arduino is connected to the Gnd pin of Fingerprint scanner. RXD of Arduino is connected to TX-out of Fingerprint scanner. Add of Arduino is Connected to $\overline{Add}$ of Fingerprint scanner. Similarly Search of Arduino is Connected to $\overline{Search}$ of Fingerprint scanner. Similarly Empty of Arduino is Connected to $\overline{Empty}$ of Fingerprint scanner and VCC of Arduino is connect to VCC of fingerprint scanner.

4.2.1 Pin Configuration

Pin 1(GND) : GND

Pin 2(TXD) : Rx

Pin 3(RXD) : Tx

Pin 4(Vcc 5V) :5V

4.3 Matrix Keypad with Arduino Mega2560

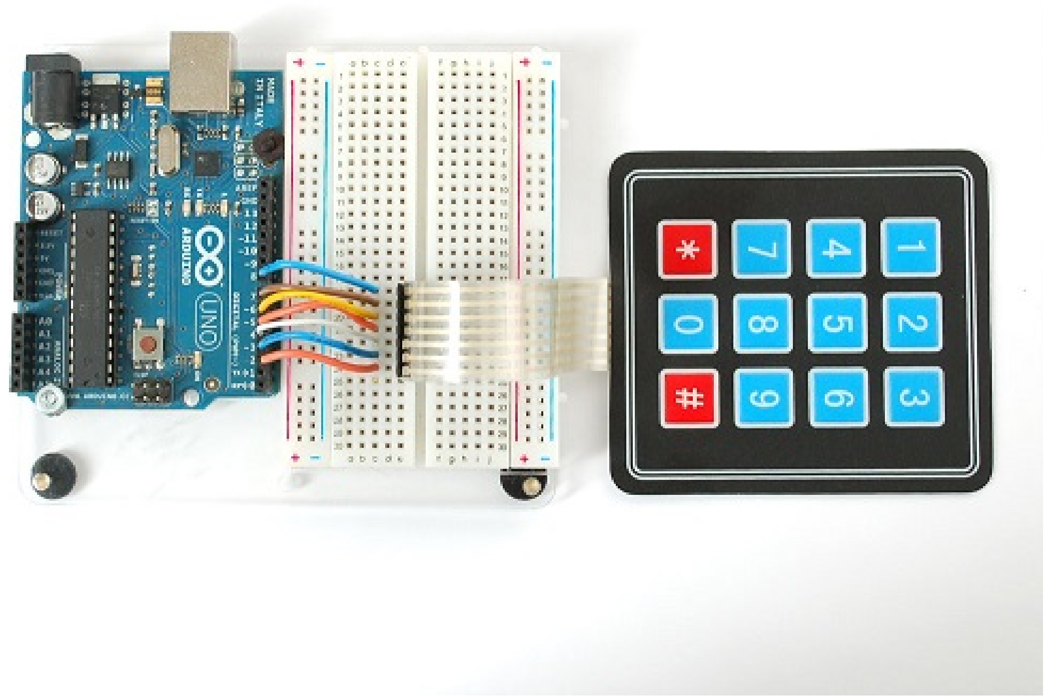

Figure 4.3: Interfacing of Matrix Keypad with Arduino Mega 2560

As Shown in the figure

∗Pin 1: Row 1.

∗Pin 2: Row 2.

∗Pin 3: Row 3.

∗Pin 4: Row 4.

∗Pin 5: Column 1.

∗Pin 6: Column 2.

∗Pin 7: Column 3.

∗Pin 8: Column 4.

Chapter 5

Arduino Software Environment

5.1 Introduction

The Arduino integrated development environment (IDE) is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a “sketch”.

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called “Wiring” from the original Wiring project, which makes many common input/output operations much easier. Users only need define two functions to make a runnable cyclic executive program:

- setup(): a function run once at the start of a program that can initialize settings.
- loop(): a function called repeatedly until the board powers off.

5.2 First Program "LED Blinking"

A typical first program for a micro controller simply blinks an LED on and off. In the Arduino environment, the user might write a program like this:

```
#define LED_PIN 13
void setup(){
pinMode (LED_PIN, OUTPUT);
}
void loop()
{
digitalWrite(LED_PIN,LOW);
delay(1000);
}
```

It is a feature of most Arduino boards that they have an LED and load resistor connected between pin 13 and ground, a convenient feature for many simple tests. The previous code would not be seen by a standard C++ compiler as a valid program, so when the user clicks the "Upload to I/O board" button in the IDE, a copy of the code is written to a temporary file with an extra include header at the top and a very simple main() function at the bottom, to make it a valid C++ program.

The Arduino IDE uses the GNU toolchain and AVR Libc to compile programs, and uses avrdude to upload programs to the board.

As the Arduino platform uses Atmega microcontrollers, Atmega's development environment, AVR Studio or the newer Atmega Studio, may also be used to develop software for the Arduino.

Chapter 6

Development

The core Arduino developer team is composed of Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis and Nicholas Zambetti. Massimo Banzi was interviewed on the March 21st, 2009 episode of FLOSS Weekly on the TWiT.tv network, in which he discussed the history and goals of the Arduino project. He also gave a talk at TEDGlobal 2012 Conference, where he outlined various uses of Arduino boards around the world.

Arduino is open source hardware: the Arduino hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino Web site. Layout and production files for some versions of the Arduino hardware are also available. The source code for the IDE is available and released under the GNU General Public License, version 2.

Although the hardware and software designs are freely available under copyleft licenses, the developers have requested that the name "Arduino" be exclusive to the official product and not be used for derivative works without permission. The official policy document on the use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the"Arduino" name by using "duino" name variants.

6.1 GPS Programing

The program for tracking up the latitudes and longitudes and storing it, is as given below in Appendix:

```
# include <LiquidCrystal.h>
# include <SoftwareSerial.h>
# include <TinyGPS.h>
float latitude, longitude;
SoftwareSerial gpsSerial(51, 50);
TinyGPS gps;
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
/∗ LCD RS pin to 8
∗ LCD Enable pin to 9
∗ LCD D4 pin to 10
∗ LCD D5 pin to 11
∗ LCD D6 pin to 12
∗ LCD D7 pin to 13
∗ LCD R/W pin to ground∗/
void setup()
{
Serial.begin(9600);
gpsSerial.begin(9600);
//set up the LCD’s number of columns and rows
lcd.begin(16,4);
lcd.clear();
lcd.home();
lcd.print(“Arduino”);
lcd.setCursor(0,1);
lcd.print(“GPS Check”);
```

```
delay(3000);
lcd.clear();
lcd.print("NO GPS DATA");
delay(5000);
}
void loop()
{
while(gpsSerial.available())//make sure data is at the serial port
{
if(gps.encode(gpsSerial.read()))
{
gpsData(gps);//write data to LCD
texttt gps.f_get_position(&latitude, &longitude);
Serial.println("Position : ");
Serial.print("LATITUDE : ");
Serial.print(latitude,6);
Serial.print(" ");
Serial.print("LONGITUDE : ");
Serial.print(longitude,6);
Serial.println();
if(latitude > 19.0000 && latitude < 19.9999 && longitude > 72.0000 && longitude
< 72.9999)
{
delay(5000);
lcd.setCursor(16,0);
Serial.println("Location Found");
lcd.print("Location Found");
digitalWrite(13,LOW);
} else
{
```

```
delay(5000);
lcd.setCursor(16,0);
Serial.println("Wrong Location");
lcd.print("Wrong Location");
digitalWrite(13,HIGH);
}
}
}
}
void gpsData(TinyGPS&gps)
{
float latitude,longitude;
texttgps.f_get_position(&latitude, &longitude);
lcd.home();
lcd.print("Lati:");
lcd.print(latitude,6);
lcd.setCursor(0,1);
lcd.print("Longi:");
lcd.print(longitude,6);
}
```

6.2 Finger Print Sensor

Detecting the valid user and giving access to it is thus shown in the program in Appendix:

```
# include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
# include <Adafruit_Fingerprint.h>
# if ARDUINO >= 100
# include <SoftwareSerial.h>
# else
# include <NewSoftSerial.h>
# endif
textttuint8_t getFingerprintEnroll(textttuint8_t id);
int k;
# if ARDUINO >= 100
SoftwareSerial mySerial(51, 50);
# else
NewSoftSerial mySerial(51, 50);
# endif
Adafruit_Fingerprint finger = Adafruit_Fingerprint(& mySerial);
uint8_t id = 0;
void setup()
{
lcd.begin(16, 4);
lcd.print(“FingerPrintSensor”);
Serial.begin(9600);
Serial.println(“fingertest”);
finger.begin(57600);
if (finger.verifyPassword()) {
Serial.println(“Found fingerprint sensor!”);
```

```
}
Else
{
Serial.println("Did not find fingerprint sensor :(");
while (1);
}
lcd.clear();
}
void loop() // run over and over again
{
Serial.println("Type in the ID # you want to save this finger as...");
lcd.print("ID=" );
id++;
Serial.print("Enrolling ID # ");
Serial.println(id);
lcd.print(id);
while (! getFingerprintEnroll(id) );
}
uint8_t getFingerprintEnroll(uint8_t id) {
uint8_t p = -1;
Serial.println("Waiting for valid finger to enroll");
lcd.print("Put the Finger");
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
Serial.println(".");
```

```
break;
case FINGERPRINT_PACKETRECIEVEER:
Serial.println("Communication error");
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
break;
default:
Serial.println("Unknown error");
break;
}
}
// OK success!
p = finger.image2Tz(1);
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image converted");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECIEVEER:
Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
return p;
```

```
default:
Serial.println("Unknown error");
return p;
}
Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
p = finger.getImage();
}
p = -1;
Serial.println("Place same finger again");
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK: {
Serial.println("Image taken");
break;
}
case FINGERPRINT_NOFINGER:
Serial.print(".");
break;
case FINGERPRINT_PACKETRECIEVER:
Serial.println("Communication error");
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
break;
default:
Serial.println("Unknown error");
```

```
break;
}
}
// OK success!
p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image converted");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECIEVER:
Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
return p;
default:
Serial.println("Unknown error");
return p;
}
// OK converted!
p = finger.createModel();
if (p == FINGERPRINT_OK) {
Serial.println("Prints matched!");
}
```

```
else if (p == FINGERPRINT_PACKETRECIEVER) {
Serial.println("Communication error");
return p;
}
else if (p == FINGERPRINT_ENROLLMISMATCH) {
Serial.println("Fingerprints did not match");
return p;
} else {
Serial.println("Unknown error");
return p;
}
p = finger.storeModel(id);
if (p == FINGERPRINT_ OK) {
Serial.println("Stored!");
} else if (p == FINGERPRINT_PACKETRECIEVER) {
Serial.println("Communication error");
return p;
} else if (p == FINGERPRINT_BADLOCATION) {
Serial.println("Could not store in that location");
return p;
} else if (p == FINGERPRINT_FLASHER) {
Serial.println("Error writing to flash");
return p;
} else {
Serial.println("Unknown error");
return p;
}
}
```

6.3 Keypad Lock Program

The Programm for typing password and detecting the right one is thus given below:

```
# include <Password.h>
# include <LiquidCrystal.h>
# include <Keypad.h>
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
Password password = Password( “4321” );
const byte ROWS = 4; // Four rows
const byte COLS = 4; // Three columns
// Define the Keymap
char keys[ROWS][COLS] = {
{‘1’,‘2’,‘3’,‘A’},
{‘4’,‘5’,‘6’,‘B’},
{‘7’,‘8’,‘9’,‘C’},
{‘*’,‘0’,‘ ’,‘D’}
}; // Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.
byte rowPins[ROWS] = 28, 29, 30, 31; //connect to the row pinouts of the keypad
byte colPins[COLS] = 34, 35, 36, 37; //connect to the column pinouts of the keypad
const int buttonPin = 5;
int buttonState = 0;
// Create the Keypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
# define ledPin 13
void setup(){
pinMode(buttonPin, INPUT);
lcd.begin(16, 4);
lcd.clear();
lcd.home();
lcd.print(“Enter Password??”);
```

```
lcd.setCursor(16,0);
digitalWrite(ledPin, LOW); // sets the LED on
Serial.begin(9600);
keypad.addEventListener(keypadEvent); //add an event listener for this keypad
keypad.setDebounceTime(250);
}
void loop(){
keypad.getKey();
buttonState = digitalRead(buttonPin);
if (buttonState == HIGH) {
lcd.clear();
}
}
//take care of some special events
void keypadEvent(KeypadEvent eKey){
switch (keypad.getState()){
case PRESSED:
lcd.print(eKey);
switch (eKey){
case ‘ ’: guessPassword(); break;
case ‘C’: setup(); break;
default:
password.append(eKey);
}
} }
void guessPassword(){
if (password.evaluate()){
digitalWrite(ledPin,HIGH); //activates garaged door relay
delay(500);
digitalWrite(ledPin,LOW); //turns off door relay after .5 sec
```

```
lcd.clear();
lcd.home();
lcd.print("Accesed"); //
password.reset(); //resets password after correct entry
delay(800);
lcd.setCursor(16,0);
lcd.clear();
delay(800);
lcd.print("Welcome");
delay(2000);
lcd.clear();
}
else{
digitalWrite(ledPin,LOW);
lcd.print("INVALID PASSWORD ");
password.reset(); //resets password after INCORRECT entry
delay(800);
lcd.clear();
setup();
}
}
```

Chapter 7

Conclusion & Future Scope

7.1 Conclusion

The GPS module error range is 5 meters so the vault will be only opened if the latitudes of the receiving end are matched with the predefined latitudes on which it is to be opened if it is correct then only the vault is defined to open.

Here we will set the latitudes of the destination and then will head for the journey and as it is controlled using GPS and will be checked for fingerprint only if the latitudes match after successful matching of the fingerprint the keypad is enabled to enter the security code for opening the vault so therefore the security for the documents is maintained till it reaches to the concerned receiver. Also receiver has other security to use i.e. keypad or Fingerprint Scanner to open the vault.

7.2 Advantages

- Used to transfer any confidential data or equipment with high security.
- Only authorized persons can access data.
- Less expensive and is useful in implementing at various locker system.
- If any failure in between security check occurs process will start from beginning.
- GPS is extremely easy to navigate as it tells you to the direction for each turns you take or you have to take to reach to your destination.

- GPS works in all weather so you need not to worry of the climate as in other navigating devices.
- The GPS costs you very low in comparison other navigation systems.
- The most attractive feature of this system is its 100% coverage on the planet.
- It also helps you to search the nearby restaurants, hotels and gas stations and is very useful for a new place.
- Due to its low cost, it is very easy to integrate into other technologies like cell phone.
- The system is updated regularly by the US government and hence is very advance.
- This is the best navigating system in water as in larger water bodies we are often misled due to lack of proper directions.

7.3 Disadvantages

- Requires detailed knowledge of latitude longitude and altitudes.
- Battery is required, without any battery supply vault cannot be opened.
- If any sensor fails in its functioning, vault cannot be opened.
- Sometimes the GPS may fail due to certain reasons and in that case you need to carry a backup map and directions.
- If you are using GPS on a battery operated device, there may be a battery failure and you may need a external power supply which is not always possible.
- Sometimes the GPS signals are not accurate due to some obstacles to the signals such as buildings, trees and sometimes by extreme atmospheric conditions such as geomagnetic storms.

7.4 Future Scope

- Solar Energy can be used instead of Power Supply.
- Face Recognition can be used for more security levels.
- If the vault is stolen the location can be tracked by a Smartphone.

References

[1] Daniel Cooper, "An Arduino-compatible development board by Intel," in 3rd International Conference, on Oct 3rd, 2013.

[2] Vigneshwar Santhanam and Vignesh Viswanathan, "Biomedical Engineering and its Applications," in 3rd International Conference, (ICEBEA'2013), Jan. 26-27, 2013.

[3] J. B. Hayfron-Acquah, M. S. Nixon, and J. N. Carter, "Fingerprint recog nition by symmetry analysis, Pattern Recognit. Lett., vol. 24, no. 13, pp. 21752183, Sep. 2003.

[4] A. Kale, A. Roychowdhury, and R. Chellappa, GPS Tracking , in Proc. IEEE ICASSP, May 1721, 2004, vol. 5, pp. V-901V-904.

[5] Hands-on with TMS320C5515 Finger print scanner module, Texas Instru ments, Jun 12 2013.

[6] C. Y. Yam, M. S. Nixon, and J. N. Carter, "Automated person recognition while walking and running via GPS approaches, GPS Tracking, vol. 37, no. 5, pp. 10571072, May 2004.