

MINI PROJECT REPORT ON

**“TWO LEVEL VERIFICATION DOOR LOCK SYSTEM”**

SUBMITTED BY:

**CHANDRA ROHITH– 1NH18EC025**

Under the guidance of

**Ms TESSY TOMY**

**Assistant Professor, Dept. of ECE, NHCE, Bengaluru.**

**NEW HORIZON COLLEGE OF ENGINEERING**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION**

**ENGINEERING**



**CERTIFICATE**

Certified that the mini project work entitled “**Two-Step Verification Door Lock System**” carried out by **CHANDRA ROHITH CHOUDARY (1NH18EC025),** bonafide students of Electronics and Communication Department, New Horizon College of Engineering, Bangalore. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree.

Project Guide HOD ECE

**Ms . TESSY TOMY**  **Dr.SANJEEV SHARMA**

|  |  |
| --- | --- |
| **External Viva** |  |
| Name of Examiner | Signature with  Date |

1.

2.

**ACKNOWLEDGEMENT**

The satisfaction that accompany the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement helped us succeed.

We thank **Dr. Mohan Manghnani**, Chairman of **New Horizon Educational Institution**, for providing necessary infrastructure and creating good environment.

We also record here the constant encouragement and facilities extended to us by **Dr.Manjunatha**, Principal, NHCE and **Dr. Sanjeev Sharma**, head of the department of Electronics and Communication Engineering. We extend sincere gratitude to them.

We sincerely acknowledge the encouragement, timely help and guidance to us by our beloved guide **Ms Tessy Tomy** to complete the project within stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of electronics and communication department for their co-operation extended to us, who helped us directly or indirectly in this successful completion of mini project.

CHANDRA ROHITH -1NH18EC025

**TABLE OF CONTENTS**

ABSTRACT Error! Bookmark not defined.

**CHAPTER 1………………………………………………………………………..…..1**

INTRODUCTION**…………………………………………………………………...…..**Error!

Bookmark not defined.

**CHAPTER 2………………………………………………………………...………….**Error!

Bookmark not defined.

LITERATURE REVIEW…………………………………………………………..……**2**Error!

Bookmark not defined.

**CHAPTER 3**

PROPOSED METHODOLOGY…………………….…………………………………..4

**CHAPTER 4**

PROJECT DESCRIPTION…………………………….…………………………...…..17

# CHAPTER 5

RESULTS AND DISCUSSION………………………………………...…………..….38

# CHAPTER 6

CONCLUSION AND FUTURE SCOPE ………………………………………………39

REFERNCES……………………………………………………………………………42

APPENDIX……………………………………………………………………………...43

**List of Figures:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Figure no** | **Page no** | **Figure description** |
| 1 | 1 | 6 | Circuit diagram |
| 2 | 2 | 7 | Block diagram |
| 3 | 3 | 8 | Flowchart |
| 4 | 4 | 13 | Arduino pins |
| 5 | 5 | 14 | pin specifications of Arduino |
| 6 | 6 | 17 | RFID pins and description |
| 7 | 7 | 21 | GSM module pins/block diagram |
| 8 | 8 | 23 | SIM900 topview |
| 9 | 9 | 27 | SIM900 interfacing with Arduino |
| 10 | 10 | 29 | 4\*4 keypad |
| 11 | 11 | 30 | Schematic outline of keypad |
| 12 | 12 | 31 | SG90 servo motor pins & specifications |
| 13 | 13 | 32 | I2C LCD |

**List of Tables:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Table no** | **Page no** | **Table description** |
| 1 | 1 | 3 | Literature survey |
| 2 | 2 | 8 | Tag reader pin description |
| 3 | 3 | 8 | I2C LCD pin connection with Arduino |
| 4 | 4 | 9 | 4\*4 keypad pin connection with Arduino |
| 5 | 5 | 9 | SIM900 pin connection with others |
| 6 | 6 | 11 | Features of Arduino |
| 7 | 7 | 19 | RFID active v/s passive tags |
| 8 | 8 | 21 | Electrical features of GSM |
| 9 | 9 | 22 | Interfacing pin description of GSM |
| 10 | 10 | 22 | Power status of GPRS |
| 11 | 11 | 23 | Pin description of SIM900 |

**ABSTRACT**

Over the decades, we have seen an immense growth in various, diverse fields of technology so as to upgrade one’s way of living to make it as comfortable as possible. Security systems is one of these fields that have been upgraded a lot from the first ever model. There’s always a more advanced system being produced in the market. However there’s always another side to the coin.

Improving technology can also give rise to thefts, forgeries and frauds. Thus we can never be too safe with our vaults, houses and safes. We always keep looking forward for a more secure, more user-friendly, more advanced model to ensure maximum safety for our valuables.

Most of the lock systems existing these days have single step security which mostly give access through either a key or a tag, both of which are easily duplicable or forgeable. Most of the times the user also doesn’t get to know of any break-in attempts, if any. Thus here we are trying to bring forward a model that will give you multiple level security so as to lessen any chances of theft and also include a user-alert system so as to alert the user at the instant of any unrecognized break-in.

This will enhance the security level of the system thus providing a more secure model which will also give instant user alert.

**CHAPTER 01**

**INTRODUCTION**

Door-Lock systems are primarily security systems that help us secure a house or a container. From a simple latch to a high-end, advanced safes, almost every house is equipped with at least one of these lock systems, which of these are mostly the conventional external lock and key systems. Few of the more developed crowd use internal lock systems. When the right key slides into the lock, the pattern (pointed teeth and notches) on the blade of the key allow the pins in the lock to move up down until the pins align with the shear line and the cylinder can turn to open the lock.

Due to the developments in today’s technology, a lot of forgeries and frauds seem to have arised since it is very easy to break in to or gain access to these security systems, as most of them do not have multiple security levels and the only security level they have, which are mostly keys, tags etc. can easily be forged/duplicated. Thus, anyone can break into places like this, provided there is no manual intervention existing. Most of these times, the owner/user doesn’t get to know of the theft until it is too late to act upon it. Thus, there is no user-alert without any manual intervention, i.e., someone notices and alarms the user of the break-in.

This is where multi-level security door lock systems come into the picture. Here the main aim is to provide different levels of security so as to reduce the risk of break-in. In this project, we will be using two levels of security, card-reading through RFID tag reading system and password entry through 4\*4 keypad and I2C LCD. We also add user alert through GSM module and interface all three to the Arduino UNO such that only after the right inputs for access are given, the door will open. The user will be given information at any instant of input at the system for access gain.

Thus, the user will need to carry both his tag and remember his password for entry gain the door. The user can also perform the basic OPEN/CLOSE commands from his phone through the GSM module such that in case the user might forget to bring his tag or his password or both. He can decide whether to open or close the system from any input for access gain at all.

We can interface the Arduino with the card reading system, password entry system and the gsm module altogether to produce the model.

**CHAPTER 02**

**Literature Review**

The concept of security system is nothing new. People have been adopting their security systems as basic as a boulder, a piece of wood or even a meagre piece of curtain in ancient times. As time progressed everyone started using conventional latch systems to secure their doors which were mostly large pieces of wood with hinges attached. Various domestic animals were also used to guard people’s houses so as to alert in case they smell any danger or intrusion

|  |
| --- |
| Early 1700’s- Tildesley created a door alarm by using a set of chimes  mechanically linked to the door lock. |

|  |
| --- |
| 1778-Lever tumbler lock is invented |

|  |
| --- |
| 1818-Chubb lock was introduced |

|  |
| --- |
| 1853- Augustus Russell Pope invented the  first electromagnetic alarm system in Boston, MA |

|  |
| --- |
| 1966- home video security system a.k.a    video surveillance introduced by |

|  |
| --- |
| 1970’s- Heat and smoke detector alarms were introduced into the market. |

|  |
| --- |
| 1984-Home Automation systems technology spreads to garage doors, environment control, IR systems, fibre optics and more |

|  |
| --- |
| 1997- Wi-Fi entered the consumer market |

|  |
| --- |
| 2000’s- hardware systems became less common and systems primarily relied on cellular monitoring as backup |

Marie Van

It was in the early 1700’s that the first alarm system was invented along with the lock system by creating a door alarm mechanically linking a set of chimes to the door lock. It was also around this time that Ben Franklin created ‘busybody’ mirrors that allow people to discreetly see who has come calling. In the 1800’s, the Chubb lock was invented along with the first electromagnetic alarm system by Augustus R. Pope and four years later, Pope sold the patent rights to his invention to Edwin Holmes, where initially the sales were sluggish but later the company took off. The American telephone and telegraph company purchased the business soon after the turn of the 20th century. This is the origin of the modern monitoring that many homeowners and also businesses rely on today. Although innovations continued throughout the century and alarm systems became to be more affordable and widely-used with time, the next major leap occurred in the 1970’s, when engineers integrated motion- sensing technology in the alarm systems. Throughout the 20th century to the present, sensors and detectors were introduced into the system to enhance the security in different, various aspects.

We have come a long way from Benjamin franklin’s clever invention. These days, homeowners and commercial enterprises have a high variety of home and commercial security options, with technology ranging from motion -sensing infrared cameras to high-definition CCTV and shooter detection systems to fingerless identity solutions to make the world a better, secure and a safer place.

|  |  |  |  |
| --- | --- | --- | --- |
| **Title of the paper** | **Author and year of publication** | **outcome** | **limitation** |
| Automatic door lock system using Arduino | Chinedu Reginald  2017 | Automatic door opening with the right  password for limited  time | Cannot detect any theft or alert user  after wrong attempts |
| RFID based door lock system with arduino | Tarun Agarwal  2015 | Door will open with the right tag | Easily forgeable and easy to break in |
| RFID based security access control system using GSM technology | Peter Adole  2016 | Door will open for the right tag and the user  will be alerted of every  event | Still a chance to forge the tag and enter. |

*Table 1*

**CHAPTER 3**

**PROPOSED METHODOLOGY**

**PRINCIPLE:**

Two-Level Verification Door Lock System is a project that operates mainly upon the interfacing of RFID system, I2C LCD and the GSM module with the Arduino for ensuring the operation of multiple levels of security. The general idea here is to:

>create higher/multiple levels of security

>to recognise any possible chances of break-in/thefts.

>to alert the user of every instant of open/close activity taking place at the door.

Since in the case of any actual theft, the user would not be able to take any action in time or it would be too late. In such cases, we would want to know immediately of such actions. Thus we will be adding GSM module so as to let the user know of all the activity regarding any input for access at the door. This will help us know if the person trying to gain access through the door is genuine or the intended user and also help us to restrict any unwanted actions.

**WORKING:**

This project is mainly based on four components: RFID tag reader,4\*4 keypad, Arduino and the GSM module. These four components will help us in building higher level security for the project.

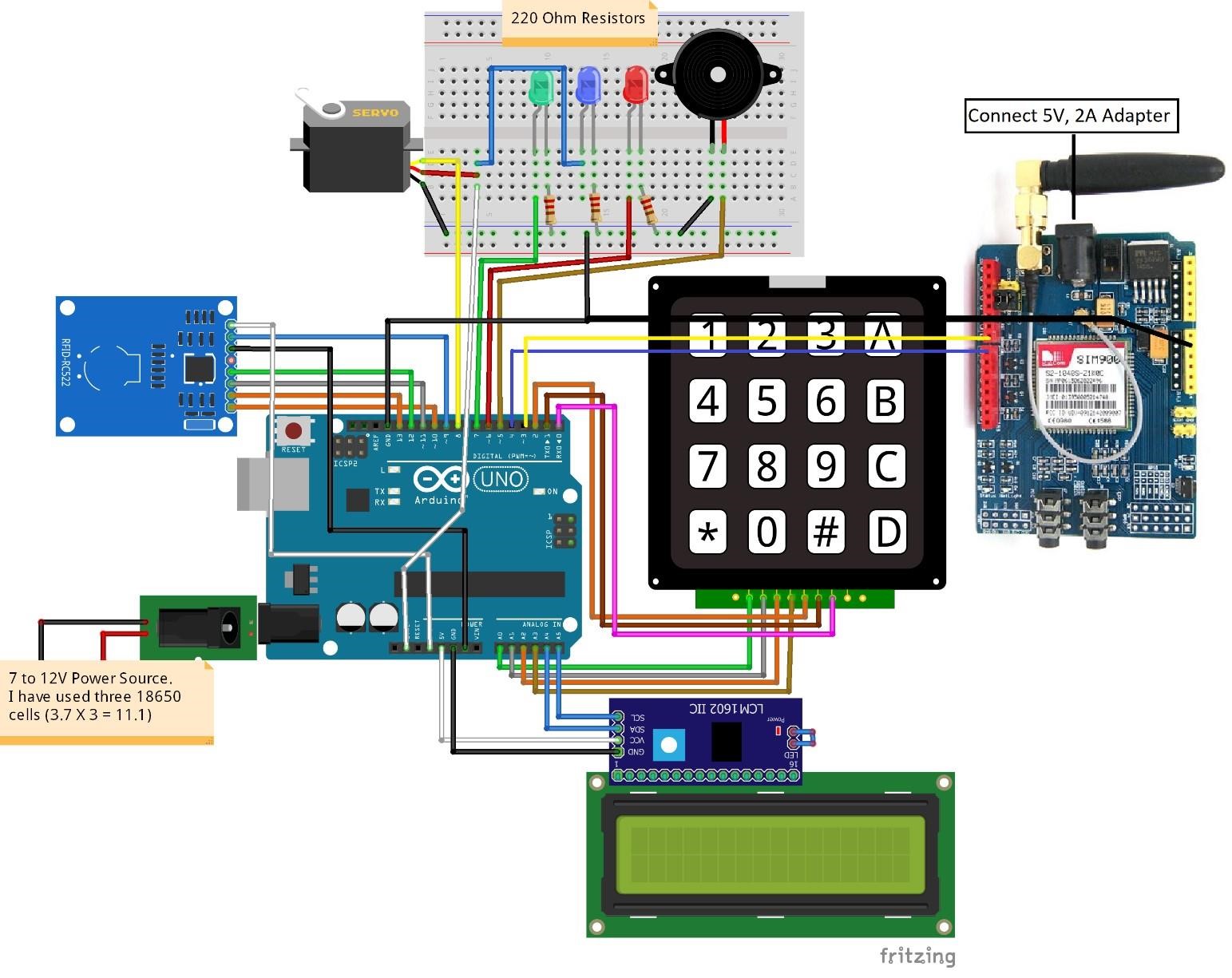
Here our project has mainly two levels of security i.e., RFID tag reading, a password lock and a user alert system for extra safety measures. The user will have to carry a tag with him all the time and keep in mind the password to get through the door, all in the while the sim card linked will be intimated of all the activity taking place at the door.

The RFID tag reader, the 4\*4 keypad, the LCD display and the GSM module are interfaced with the Arduino. The RFID reader communicates with the Arduino through the SPI protocol. The I2C LCD communicates with the Arduino through I2C protocol.

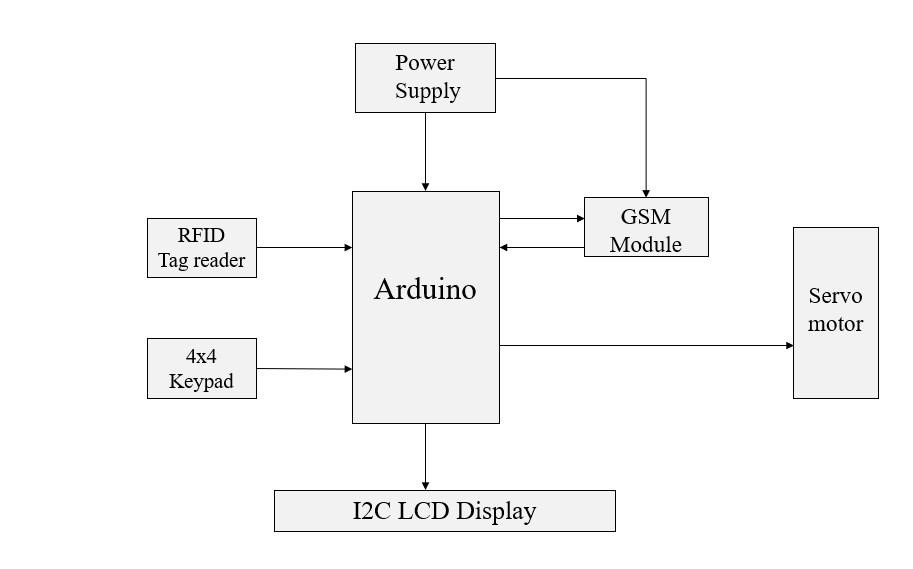
Input will be taken from the tag reader and the keypad and will be given to the Arduino UNO and the LCD display connected to the Arduino will communicate with us whether the given input i.e., the tag and the password is the correct one to gain access through the door or not.

If yes, the door will be opened and a message will be displayed on the LCD display about it and the same will be sent to the phone. If not the right input, the same will be displayed on the LCD and be sent to the phone. The password, the tag UID, the response timeout can be changed within the code according to the user preferences.

**CIRCUIT DIAGRAM:-**

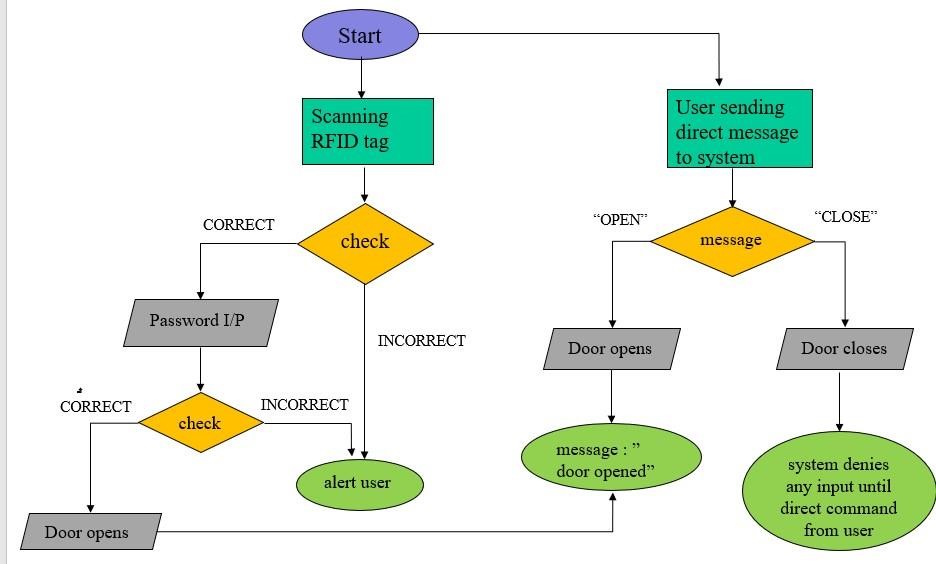


*Figure 1*

 *Figure 2*

* Firstly the user will have to scan the tag at the MFRC522 RFID tag reader, which will be interfaced with the Arduino UNO. The output from the tag reader will be then given to the Arduino.
* Upon scanning the right tag, the system/model will then ask for a password which will be pre-set in the code. This password will be taken input into and verified by the 4\*4 keypad and the I2C LCD which will be interfaced with the Arduino.
* On entering the right password, the I2C LCD will show a display that the door has been opened and the user will be sent an sms to the phone containing the sim/linked to the sim through the GSM module SIM 900 that the door has been opened.
* The GSM module interfaced with the Arduino helps us to communicate with the system by basic commands.
* The door will remain open for the programmed amount of time and then shuts close automatically. This response timeout can be programmed accordingly in the code.
* On scanning the wrong tag or password, the door will not open and the user will be sent an alert message about the status/activity
* At any point that the user feels suspicious about any activity at the door, he can send a message through his phone to close the door and the whole system will deny any access and the door will remain unlocked until any further user command.
* The user can also directly open the door through the phone command message if in case he forgets to bring the tag or the password.

**FLOWCHART:**

 *Figure 3*

The above flowchart precisely shows the flow of information across the project including what happens when the given input is accurate until the very end and what happens when there is a wrong input at any stage of the process, the user will be sent an alert about the wrong input and he can decide whether to take action through basic commands from the phone.

The RFID reader communicates with the Arduino board through SPI protocol and different Arduino boards have different SPI pins. Here are the pin connections of the RFID tag reader with the Arduino UNO:-

|  |  |
| --- | --- |
| **MFRC522**  **RFID Reader** | **Arduino UNO** |
| RST/Reset | 9 |
| SDA | 10 |
| MOSI | 11/ICSP-4 |
| MISO | 12/ICSP-1 |
| SCK | 13/ICSP-3 |
| GND | GND |
| 3.3 V | 3.3 V |

*Table 2*

The I2C LCD communicates with the Arduino through the I2C protocol. Different Arduino boards have different pins and the pins for I2C on Arduino Uno are A4 and A5. The pin connections of the I2C LCD with the Arduino are:-

|  |  |
| --- | --- |
| **I2C LCD** | **Arduino UNO** |
| SCL | A5/SCL |
| SDA | A4/SDA |
| GND | GND |
| VCC | 5V |

*Table 3*

The 4\*4 keypad has 8 connections but we will not be requiring the last column of the keypad, we will only be requiring the numbers for the password. So we don’t use the last pin of the keypad which is for the fourth column. The pin connections of the keypad with the keypad are given below:-

|  |  |
| --- | --- |
| **4\*4 Keypad** | **Arduino UNO** |
| 1st pin | A0 |
| 2nd pin | A1 |
| 3rd pin | A2 |
| 4th pin | A3 |
| 5th pin | D2 |
| 6th pin | D1 |
| 7th pin | D0 |

*Table 4*

The below table describes the connections of Arduino with SIM900 and the other components:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Arduino** | **SIM900** | **LED’s** | **Servo** | **Buzzer** |
| D3 | D7 |  |  |  |
| D4 | D8 |  |  |  |
| 5V |  | Blue LED | Positive wire (red) |  |
| D8 |  |  | Signal wire (yellow) |  |
| D7 |  | Green LED |  |  |
| D6 |  | Red LED |  |  |
| D5 |  |  |  | + |
| GND | GND | Negative leads  through 220 ohm  resistor | GND  Wire(brown) | - |

*Table 5*

Once the SIM900 module has been powered up, the power light will light up and on pressing the power key, the status light should light up and the netlight should start blinking. We can verify if the module is working properly by making a call from the module to our mobile.

0000

**CHAPTER 4**

# PROJECT DESCRIPTION

**COMPONENTS:**

1.Arduino UNO

2.I2C LCD

3.MFRC522 RFID reader

4.Tags

5.SIM 900 GSM module

6.LEDs (blue, green, red)

7.Resistor (221 ohm)

8.5V,2A Adapter

9.power supply (6 to 12V)

10.SG90 micro-servo motor

11.4\*4 keypad

12.Buzzer

**Arduino UNO**

The Arduino UNO is a microcontroller board based on the ATMEGA328. It has 14 digital input/output pins (out of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, 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 UNO differs from all preceding boards in that it does not use the FTDI USB to serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB to serial converter. “Uno” means one in Italian and is named to mark the upcoming release of Arduino 1.0.

the

Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in the series of USB Arduino boards, and the reference model for the Arduino platforms: for any comparision with previous versions.

**FEATURES OF AN ARDUINO :-**

|  |  |
| --- | --- |
| Microcontroller | Atmega328 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage(limits) | 6-20V |
| Digital I/O Pins | 14(of which 6 provide PWM output) |
| Analog input pins | 6 |
| DC Current per I/O pin | 40 Ma |
| DC current for 3.3V pin | 50 Ma |
| Flash memory | 32KB (Atmega328) of which 0.5KB used by bootloader) |
| SRAM | 2KB(Atmega328) |
| EEPROM | 1KB(Atmega328) |
| Clock speed | 16 MHz |

*Table 6*

**PINS DESCRIPTION:**

**Power**

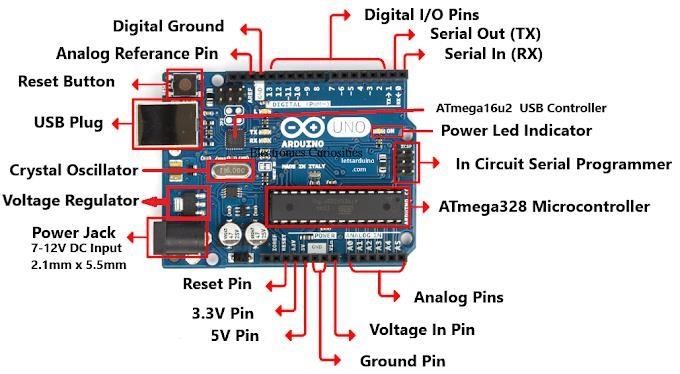
The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

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 2.1 mm centre-positive plug into the board’s power jack. Leads from 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.

**Power pins are :-**

* **Vin**: The input voltage to the Arduino boar when its using an external power source(as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
* **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack(7-12V), the USB connector(5V) or the Vin pin of the board(7-12V) supplying voltage via the 5V or 3.3V pins bypasses the regulator and damage your board.
* **3V:** A 3.3volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
* **GND**: ground pins.
* **IOREF**: This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.



*Figure 4*

**Memory**

The ATmega328 has 32KB (with 0.5 KB used for the bootloader). It also has 2KB of SRAM and 1 KB of EEPROM.

**Input and Output**

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite() and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50kOhms. In addition, some pins have specialized functions.

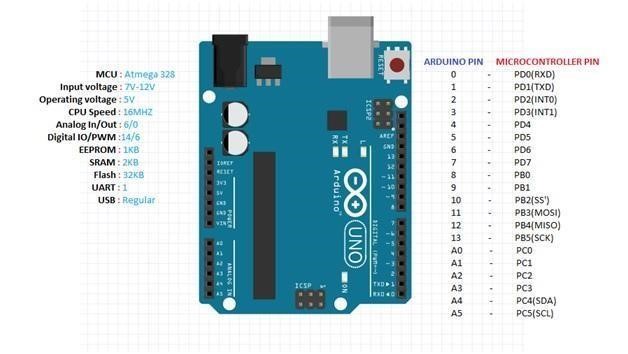
* **Serial: 0(RX) and 1(TX):** used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL serial chip.
* **External Interrupts: 2 and 3:** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge or a change in value.
* **PWM: 3,5,6,9,10,11:** provide 8-bit PWM output with the analogWrite() function.
* **SPI: 10(SS), 11(MOSI), 12(MISO), 13(SCK),** these pins support SPI communication using the SPI library.
* **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, its off.

The Uno has 6 analog inputs, labelled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionally:

* **TWI: A4 or SDA pin and A5 or SCL pin:** support TWI communication using the Wire library.

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.



*Figure 5*

**Communication:**

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

A software serial library allows for serial communication on any of the Uno’s digital pins.

The ATmega328 also does support I2C(TWI) and SPI communication. The Arduino software includes a wire library to simplify use of the I2C bus. For SPI communication, use the SPI library.

**RFID Tag Reader:-**

RFID stands for radio frequency identification. It uses radio waves to automatically identify people or objects. RFID is an automated data-capture technology that can not only be used to electronically identify and track but also store information contained on a tag. A radio frequency reader scans the tag for data and sends the information gained to a database, which stores the data contained on the tag.

Identification processes that rely on AIDC (automatic identification and data capture) technologies are significantly more reliable and less expensive than those that aren’t automated. The most common AIDC technology is bar code technology, which uses optical scanners to read the labels. Most people have direct experience with bar codes because they have seen cashiers scan items at supermarkets and retail stores. Bar codes are an enormous improvement over ordinary text labels because personnel are no longer required to read numbers or letters on each label or manually enter data into an IT system, they would just have to scan the label present. The invention of bar codes has greatly improved the speed and accuracy of the identification process and facilitated better management of inventory and pricing when coupled with the information systems.

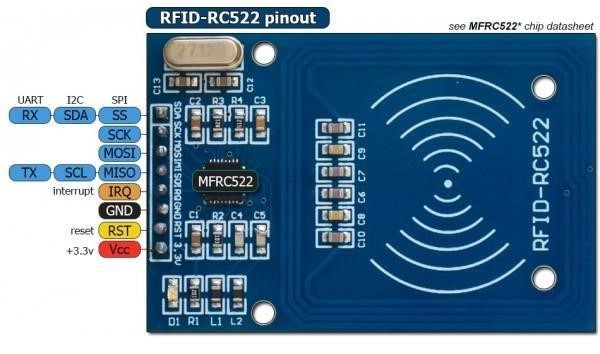
RFID represents a technological advancement in AIDC because it offers the advantages that are not available in the AIDC systems such as bar codes. RFID can offer these advantages because it relies on radio frequencies to transmit information rather than light, which is required for the optical AIDC technologies. The use of radio frequencies means that RFID communication can occur:

* Without any optical line of sight, since radio waves can penetrate many materials.
* At greater speeds, because many of the tags can be read quickly, whereas optical technology often requires time to manually reposition the objects to make their bar codes visible.
* Over greater distances, because many radio technologies can transmit and receive signals more effectively than optical technology under most operating conditions.

the ability of RFID technology to communicate without the optical line of sight and over greater distances than other AIDC technology further reduces the speed for human involvement in the identification process. For example, several retail firms have pilot RFID programs to determine contents of a shopping cart without removing each item and placing it near the scanner to scan as is seen at most of the stores today. In this case the ability to scan a cart without removing the contents will speed up the checkout process, thereby decreasing transaction costs for the retailers. This application of RFID also has the potential to significantly decrease checkout time for the consumers.

RFID products often support other feature that bar codes and other AIDC technologies do not have, such as rewritable memory, security features and also environmental sensors that enable RFID technology to record a history of such events. That types of events that can be recorded include temperature changes, sudden shocks or high humidity levels. Today people typically perceive the label identifying a particular object of interest as static, but RFID technology can make this label dynamic or even “smart” by enabling the label to acquire or gain data about the particular object even when people are not present to handle it.

RFID tags and readers must be tuned into the same frequencies to enable communications. RFID systems can use a wide variety of frequencies to communicate. But because radio waves work and act differently at different frequencies, a frequency for a given specific RFID system is often dependent on its application. High frequency RFID systems (850 MHz to 950 MHz and 2.4 GHz) offer transmission ranges of more than 90 feet, although wavelengths in the 2.4 GHz range are absorbed by water, which includes the human body and therefore has its limitations.



*Figure 6*

An **RFID Tag** or transponder consists of a chip and an antenna. A chip can store a unique serial number or other information based on the type of memory of the tag, which can be readonly, read-write or write-once read-many. The antenna which is attached to the microchip transmits information from the chip to reader. Typically, a larger antenna indicates a longer read range. The tag is usually attached to or embedded in and object to be identified, such as a product, case or pallet and can be scanned by a mobile or stationery readers using the radio wave.

CLASSIFICATION OF TAGS:-

Tags are classified into different types based on battery and memory. They are-

-passive tags

-active tags

-semi passive tags -read only tags

-read write tags

-write once read many times tags

PASSIVE TAGS

The simplest version of a tag is the passive tag where the tags here do not contain their own power source, such as battery, nor will they be able to initiation with a reader. Instead the tag will respond to the reader’s radio frequency emissions and derives its power from the energy waves that will be transmitted from the reader.

A passive tag contains, at the least, a unique identifier for the individual item attached to the tag. Depending on the storage capacity of a given tag, additional data can be added. Under perfect conditions, the tags can be read from a range of about around 10 to 20 feet. The costs of these tags can vary based on the radio frequency used, the amount of memory, the design of the antenna and the packaging around the transponder, among the other tags requirements.

Passive tags can operate at a high, ultrahigh, low or microwave frequency. Examples of massive tag applications include mass transmit passes, building of access badges and the consumer products in the supply chain. The development of these inexpensive tags has created a revolution/turning point in the RFID adoption and made wide scale use of them a real possibility for government and industry organizations.

ACTIVE TAGS

Active tags can contain a power source and a transmitter, in addition to the antenna and chip and sends a continuous signal. These tags typically have read/write abilities-tag data can also be rewritten and/or modifies. Active tags can initiate communication and communicate over longer distances like up to 740 feet which depends on the battery power. The relative expense of these tags can make them an option for use only where their high cost can be justified. Active tags are more expensive than the passive tags.

Examples of active tag applications are toll passes, such as “E-Z pass” and the in transit visibility applications on the major items and consolidated cargo moved by the DOD (Defense of development).

TECHNICAL CHARACTERISTICS

Active RFID and passive RFID are fundamentally different technologies. While both of them use radio frequency energy to communicate between any given tag and a reader, the method of powering the tags is different. Active RFID uses an internal power source such as a battery within the tag to continuously power the tag and its RF communication circuitry, whereas passive RFID primarily relies on RF energy transferred from the reader to the tag so as to power the tag.

Passive RFID either 1) reflects energy from the reader or 2) absorbs the energy and temporarily stores a very small amount of energy from the reader’s signal to generate its own quick response. In either case, passive RFID operations here require very strong signals from the reader and the signal strength returned from the tag is constrained to very low levels by limited energy, On the other hand, active RFID allows very low-level signals to be received by the tag since the reader does not need to power the tag and the tag can produce high-level signals back to the reader driven from its internal power source. Also, additionally the active RFID tag is continuously powered whether it is in the reader field or not.

|  |  |  |
| --- | --- | --- |
|  | ACTIVE TAGS | PASSIVE TAG |
| Power source | Internal (tsg) | Energy transferred via reader |
| Battery | Yes | No |
| Availability of tag power | Continuous | Only within reader field |
| Required signal strength from reader to tag | Low | High |
| Available signal strength from the tag to reader | High | low |
| Communication range | Long range (>100m) | Short/very short(<3m) |
| Multi tag collection | Collects more tags in long range | Collects few tags in shorter range |
| Sensor capability | Continuous monitoring and recording sensor input | Can only read and transfer the sensor values when the tag is powered by the reader. |
| Data storage | Large read-write data storage | Small read/write storage |
| Circuit | Complex | Simple |
| Area | Occupies larger area | Occupies smaller area |
| Speed | Fast | Slow |
| Cost | More than passive | less |
| Lifetime of tag | Lesser when compared to passive. | More when compared to active. |

|  |  |  |
| --- | --- | --- |
| application | Long distance communication like check post automation, animal identification, vehicle identification. | Short distance communication like supply chain management such as in super markets. |

*Table 7*

RADIO FREQUENCIES FOR RFID SYSTEMS

Choice of the radio frequency is a key operating characteristic of the RFID systems. The frequency largely determines the speed of communication and the distance from which the tag can be read. Generally, higher frequencies indicate a longer range of read. Certain applications are more useful for one type of frequency than other types, because radio waves behave differently at each of these frequencies. For instance, low frequency waves can penetrate the walls better than the higher frequencies, but the higher frequencies have a faster data-rates.

RFID system use an unlicensed frequency range, classified as industrial scientific medical or short-range devices, which is authorized by the FCC

**GSM Module SIM 900:-**

SIM900 GSM/GPS shield is a GSM modem, which can be integrated into a great number of IoT projects. You can use this shields to accomplish almost upto everything anything and everything a normal cell phone can do or accomplish such as SMS text messages, making or receiving phone calls, connecting to internet through GRPS, TCP/IP and many more such.

SIM900A is a double band GSM/GPRS motor that chips away at frequencies EGSM 900MHz and DCS 1800MHz. SIM900A highlights GPRS multi-opening class 10/class 8

(discretionary) and underpins the GPRS coding plans CS-1, CS-2, CS-3 and CS-4

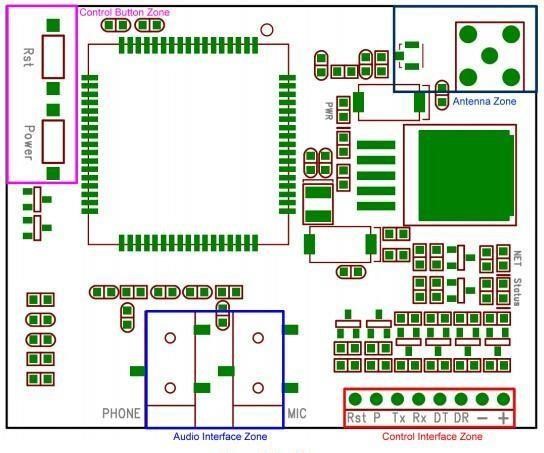
FEATURES

* Quad-Band 850/900/1800/1900 MHz
* Double Band 900/1900 MHz
* GPRS multi-opening class 10/8GPRS versatile station class B
* Consistent to GSM stage 2/2+Class 4 (2 W @850/900 MHz)
* Class (1 W @ 1800/1900MHz)
* Control by means of AT orders (GSM 07.07 ,07.05 and SIMCOM improved AT Commands)
* Low force utilization: 1.5mA(sleep mode)
* Activity temperature: - 40°C to +85 °C

ELECTRICAL FEATURES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Min. | Typical | Max. | unit |
| Power voltage (vsupply) | 4.5 |  | 5.5 | VDC |
| Input voltage  VH | 0.7VCC |  | 5.5 | V |
| Input voltage VL | -0.3 | 0 | 0.3VCC | V |
| Current consumption (pulse) | - |  | 2000 | mA |
| Current consumption (continuous) |  |  | 500 | mA |
| Band rate |  | 115200 |  | bps |

*Table 8*



*Figure 7*

|  |  |  |
| --- | --- | --- |
| Interface | Pin | Description |
| Rst | 1 | Reset the SIM900 module |
| P | 2 | Power switch pin of SIM900 module |
| Tx | 3 | UART data output |
| Rx | 4 | UART data in |
| Dt | 5 | Debug UART data output |
| Dr | 6 | Debug UART data input |
| - | 7 | GND |
| + | 8 | VCC |

*Table 9*

**Power on GPRS module**

Client can control on the GPRS module by pulling down the PWR button or the P pin of control interface for in any event 1 second and delivery. This pin is as of now pulled up to 3V in the module inner, so outer draw up isn’t important. At the point when power on strategy is finished, GPRS module will send following URC to demonstrate that the module is prepared to work at fixed baud rate.

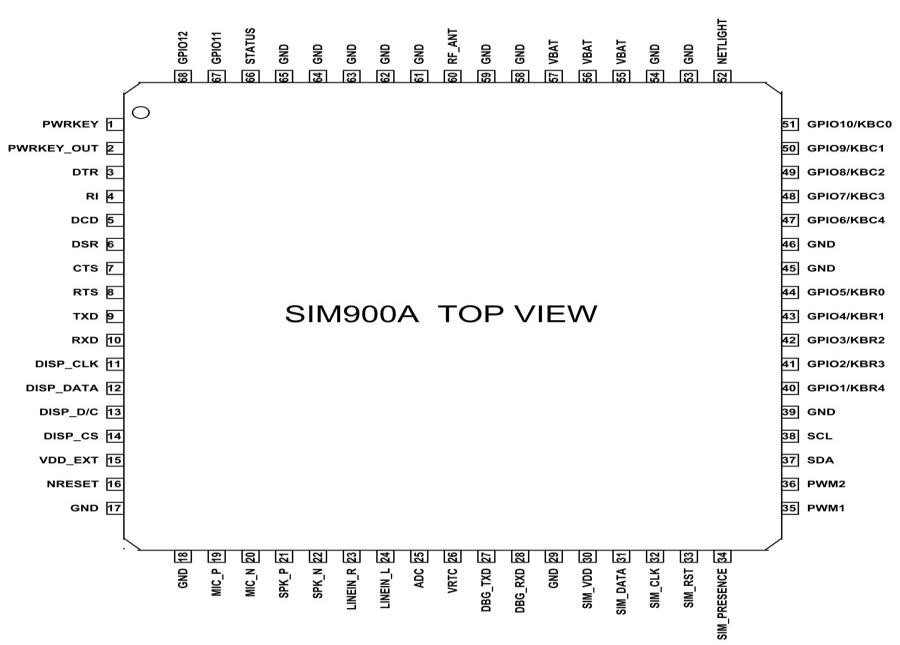
|  |  |
| --- | --- |
| **Status** | **Description** |
| Off | SIM900 is not running 64ms On/800ms |
| Off | SIM900 not registered the network |
| 64ms On/3000ms Off | SIM900 registered to the network |
| 64ms On/3000ms Off | GPRS communication is established |

*Table 10* **STATUS**: Power status of SIM900.

**PWR**: Power status of GPRS module.

**PWR:** After the GPRS module power on, you need to press the POWER button for a second to control on the SIM900 module.

**RESET:** Reset the SIM900 module.

 *Figure 8*

**Pin description:-**

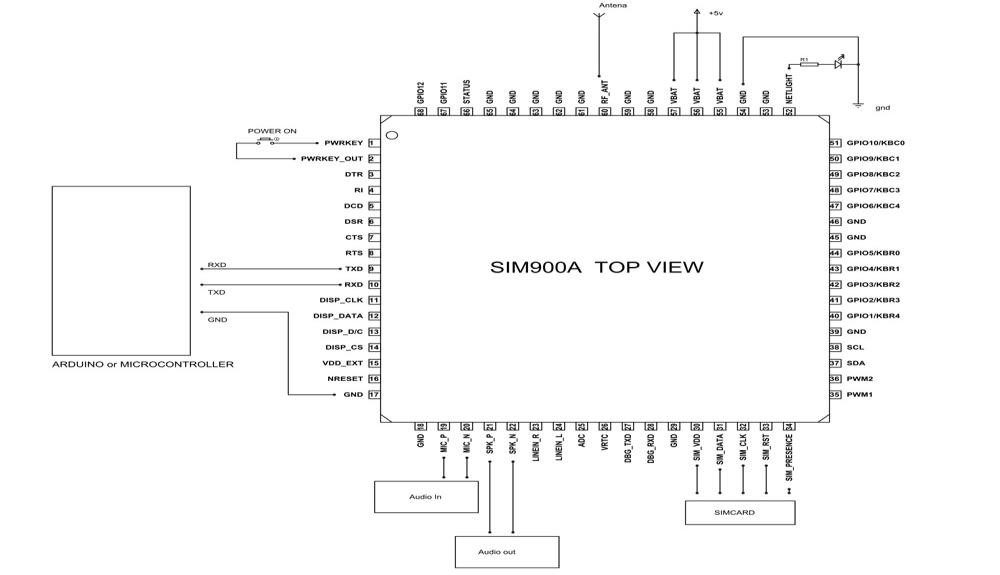
|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name**  PWRKEY  PWRKEY\_OUT  DTR  RI | **Description** |
| 1 | Voltage input for PWRKEY. PWRKEY should be pulled low to power on or power off the system.  The user should keep pressing the key for a short time when power on or power off the system because the system need margin time in order to assert the software. |
| 2 | Connecting PWRKEY and PWRKEY\_OUT for a short time then release also can power on or power off the module. |
| 3 | Data terminal Ready [Serial port ] |
| 4 | Ring indicator [Serial port ] |
| 5 | DCD  DSR  CTS  RTS  TXD  RXD  DISP \_CLK  DISP\_DATA  DISP \_D/C | Data carry detect [Serial port ] |
| 6 | Data Set Ready [Serial port ] |
| 7 | Clear to send [Serial port ] |
| 8 | Request to send [Serial port ] |
| 9 | Transmit data [Serial port ] |
| 10 | Receive data [Serial port ] |
| 11 | Clock for display [Display interface] |
| 12 | Display data output [Display interface] |
| 13 | Display data or command select [Display interface] |

|  |  |  |
| --- | --- | --- |
| 14 | DISP \_CS  VDD\_EXT  NRESET  GND  MIC\_P  MIC\_N  SPK\_P  SPK\_N | Display Enable [Display interface] |
| 15 | 2.8V output power supply |
| 16 | External reset input |
| 17,18,29,39,45,  46,53,54,58,59,  61,62,63,64,65 | Ground |
| 19 | Microphone Positive |
| 20 | Microphone Negative |
| 21 | Speaker Positive |
| 22 | Speaker Negative |
| 23 | LINEIN\_R  LINEIN\_L    ADC  VRTC  DBG\_TXD  DBG\_RXD | Right Channel input [External line inputs are available to directly mix or multiplex externally generated analog signals such as polyphonic tones from an external melody IC or music generated by an FM tuner IC or module.] |
| 24 | Left Channel Input |
| 25 | General purpose analog to digital converter. |
| 26 | Current input for RTC when the battery is not supplied for the system.  Current output for backup battery when the main battery is present and the backup battery is in low voltage state. |
| 27 | Transmit pin [Serial interface for debugging and firmware upgrade  ] |
| 28 | Receive pin [Serial interface for debugging and firmware upgrade ] |

|  |  |  |
| --- | --- | --- |
| 30 | SIM\_VDD  SIM\_DATA SIM\_CLK  SIM\_RST  SIM\_PRESENCE  PWM1  PWM2  SDA  SCL  KBR0 to KBR4 &  KBC4 to KBC0 | Voltage supply for SIM card |
| 31 | SIM data output |
| 32 | SIM clock |
| 33 | SIM reset |
| 34 | SIM detect |
| 35 | PWM Output |
| 36 | PWM Output |
| 37 | Serial Data [I2C] |
| 38 | Serial Clock [I2C] |
| 40,41,42,43,44 &  47,48,49,50,51 | Keypad interface [ROWS & COLUMNS] |
| 52 | NETLIGHT  VBAT  RF\_ANT  STATUS  GPIO 11  GPIO 12 | Indicate net status |
| 55,56,57 | Three VBAT pins are dedicated to connect the supply voltage. The power supply of SIM900A has to be a single voltage source of VBAT= 3.4V to 4.5V. It must be able to provide sufficient current in a transmit burst which typically rises to 2A. |
| 60 | Antenna connection |
| 66 | Indicate working status |
| 67 | General Purpose Input/output |
| 68 | General Purpose Input/output |

*Table 11*

**How to use SIM900A**



*Figure 9*

As Shown over, the correspondence with this module is done through UART or RS232 Interface. The information is shipped off the module or got from the module however UART interface.

The module is commonly associated with +4.0V standard force supply. It can deal with +4.5V directed force and any higher voltage may harm the module. Furthermore, the force source should have the option to convey a pinnacle current of 2A. The UART interface is set up as appeared in figure. You should simply interface RXD of module to TXD of Arduino and TXD is associated with RXD of ARDUINO. The ground of regulator and module should be associated for voltage reference. Here AUDIO IN is associated with MIC and AUDIO OUT is associated with a speaker or headset. What's more, finally we need to interface a working GSM SIM card to the module. On fueling the module the NETLIGHT LED will flicker occasionally to state fruitful association.

After all associations are done, we require to compose a program for the microcontroller to trade information with module. Since information trade grouping among regulator and module is truly unpredictable we will utilize libraries prewritten for the module. You can download libraries for regulator or module through their sites. Utilizing these libraries makes the correspondence simple. You should simply download these libraries and call them in projects. When the header record is incorporated, you can utilize straightforward orders in the program to advise the regulator to send or get information. The regulator sends the information to the module through UART Interface dependent on convention arrangement in libraries. The module sends this information to another GSM client utilizing cell organization. In the event that the module gets any information from the cell organization (or another GSM client) it will send it to regulator through UART sequential correspondence.

**Applications**

* Cellular Communication
* Robotics
* Mobile Phone Accessories
* Servers
* Computer Peripherals
* USB Dongles
* Automobile

**4\*4 keypad :**

The 4\*4 framework keypad ordinarily is utilized as contribution to a venture. It has 16 keys altogether, which implies similar info esteems.

The SunFounder 4\*4 Matrix Keypad Module is a grid non-encoded keypad comprising of 16 keys in equal. The keys of each line and section are associated through the pins outside – pin Y1-Y4 as named alongside control the lines, when X1-X4, the segments.



*Figure 10*

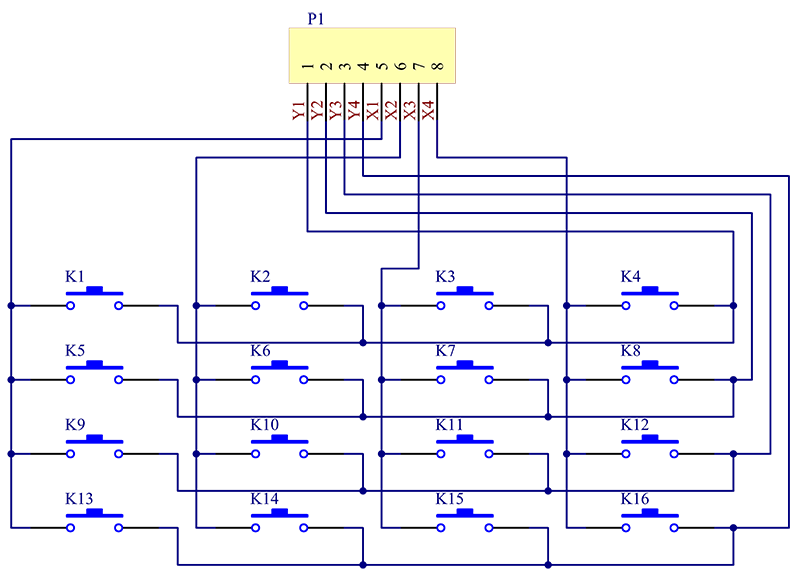
**How does it work:**

First test whether any key is pushed down. Interface capacity to lines, so they are High level. At that point set all the lines Y1-Y4 as Low and afterward recognize the status of the sections. Any section of Low shows there is key squeezing and that the key is among the 4 keys of the segment. On the off chance that all segments are High, it implies no key is pushed down.

Next, find the key. Since the section wherein the squeezed key untruths is recognized, realizing the line would finish the testing. Consequently, set the columns as Low in turns until any is revealed likewise – different lines will even now be High.

Presently the column can be distinguished. Distinguish the status of every section reciprocally. The section tried Low is the one converging with the line – their cross point is only the key squeezed.

The schematic outline:

 *Figure 11*

**SG90 micro-servo motor:**

It is small and lightweight with high yield power. This servo can turn around 180 degrees (90 toward every path), and works much the same as the standard sorts yet more modest. You can utilize any servo code, equipment or library to control these servos. It accompanies a 3 horns (arms) and equipment.

**specification:**

Working voltage: 4.8 V (~5V)

Working rate: 0.1 s/60 degree

Stall torque: 1.8 kgf·cm

Dead band width: 10 µs

Temperature range: 0 ºC – 55 ºC

Equipment and Software Required

SG90 Micro Servo engine

Arduino Uno

Arduino IDE (1.0.6V)

**Hardware connections:**

The SG90 miniature servo engine has 3 wire interface in which the associations should made as follows:

Red wire-5V

Earthy colored wire-Ground

Yellow wire-computerized pin 9

 *Figure 12*

**I2C LCD display:**

As we as a whole know, however LCD and some different shows significantly enhance the man-machine association, they share a typical shortcoming. At the point when they are associated with a regulator, numerous IOs will be involved of the regulator which has no so numerous external ports. Likewise it confines different elements of the regulator. Along these lines, LCD1602 with an I2C transport is created to tackle the issue.



*Figure 13*

I2C transport is a sort of sequential transport created by PHLIPS. It is an elite sequential transport which has transport administering and high or low speed gadget synchronization work needed by various host framework. I2C transport has just two bidirectional sign lines, Serial Data Line (SDA) and Serial Clock Line (SCL). The blue potentiometer on the I2C LCD1602 is utilized to change backdrop illumination to make it simpler to show on the I2C LCD1602.

As we all know, though LCD and some other displays greatly enrich the man-machine interactions they share a common weakness when they are connected to a controller, multiple IOs will be occupied of the controller which has not so many outer ports, also it restricts other functions of the controller. Therefore, LCD1602 with a I2C bus is developed to solve the problem.

It is a high-performance serial bus which has bus ruling and high or low speed device synchronization function required by multiple host system.

The SCL line is the clock signal which synchronizes the data transfer between the devices used on the I2c bus and its generated by the master device. The other line is the SPL line which carries the data

**Resistor:**

A resistor is a passive element. Resistors is an electrical component that reduces the electrical current flowing through a circuit line. The resistor’s ability to reduce the current is known as resistance and is measured in ohms.

• Here we have used three220 ohm resistors in the project.

**Power supply:**

A power supply is a device that supplies electric power to electric load. The term is the most commonly referred electric power that converts one form of electrical energy to other, though it may also refer to that convert another form of energy (mechanical, chemical, solar) to electrical energy. The regulated power supply is that controls the output voltage or current to a specific value.

The power supply we have used is a 6-12 V power source. The recommended power for powering the SIM900 module is to use 5V,2A.

**CHAPTER 5**

**RESULTS AND DISCUSSION**

The expected result here will be-

* User input at the RFID tag reader
* Upon correct tag- the system will ask for the password
* Upon wrong tag input- the system will deny access
* Upon the right password- the door lock will open, remain open for a programmed amount of time and close automatically
* Upon the wrong password- the door will deny access

User alert :-

* The user must be alerted of every activity at the door, including any input for access, wrong and right inputs and opening/closing of the door lock.
* the user must be able to close or open the door through simple commands at any point of time if he feels the need to do so.

**CHAPTER 6**

**CONCLUSION AND FUTURESCOPE**

**CONCLUSION:**

As the technology rises, so do thefts and forgeries. Here, we have tried to approach the drawback of a single step security system having higher chances of break-in or thefts and make a model that is both user friendly and provides higher security system by incorporating multiple levels into the circuit.

Through the use of a tag reader and a password lock system we have set up two levels of security so as to tighten the access and lessen the chances of break-in. we approached the limitation of no immediate user alert with the help of a gsm module. By interfacing SIM900 GSM module with the circuit, we overcame the problem of delayed user alert in case of a possible theft. Not only can we open or close the system with a simple command, we can also be alerted of any input being given at the door through the system. Thus the system provides a far more secure, safer model than the existing single step security systems.

**FUTURE SCOPE:**

* Model can be further developed by adding extra security levels such as face recognition, Iris recognition, Fingerprint sensor to ensure the right individual gets access through the system.
* We can also update the model by adding features through GSM module where emergency services will be contacted automatically if there is any attempt to break-in recognized.

REFERENCES:

1. **“**Access control system using RFID and Arduino” [The Journal of Scientific and Engineering Research](https://www.researchgate.net/journal/2394-2630_The_Journal_of_Scientific_and_Engineering_Research) 05(04):333-340 – by Zeluwa Orji- April 2018
2. “Electronic Door Lock using RFID and Password Based on Arduino” by Ni Ni San

Hlaing -April 2019

1. [https://electronicshobbyists.com/rfid-and-keypad-based-door-lock-and-alert-systemusing-arduino/](https://electronicshobbyists.com/rfid-and-keypad-based-door-lock-and-alert-system-using-arduino/)
2. https://create.arduino.cc/projecthub/muhammad-aqib/rfid-and-keypad-door-lock-andalert-system-using-arduino-60f050

**APPENDIX:**

CODE:

The tag’s UID must be given at String tagUID =”29 B9 ED 23” Number must be added at the given line of the code:

SIM900.println(“AT+CMGS=\”+XXXXXXXXXXXX\’’);

The initial password will be ‘1234’ The code to be used is as follows:-

1. “// Include required libraries”
2. “#include <MFRC522.h>”
3. “#include <LiquidCrystal\_I2C.h>”
4. “#include <Keypad.h>”
5. “#include <SoftwareSerial.h>”
6. “#include <Servo.h>”
7. “#include <SPI.h>” 8.
8. // Create instances00
9. SoftwareSerial SIM900(3, 4); // SoftwareSerial SIM900(Rx, Tx)00
10. MFRC522 mfrc522(10, 9); // MFRC522 mfrc522(SS\_PIN, RST\_PIN)000
11. LiquidCrystal\_I2C lcd(0x27, 16, 2);00
12. Servo sg90;00 14.
13. // Initialize Pins for led's, servo and buzzer000
14. // Blue LED is connected to 5V00
15. constexpr uint8\_t greenLed = 7;00
16. constexpr uint8\_t redLed = 6;00
17. constexpr uint8\_t servoPin = 8;00 20. constexpr uint8\_t buzzerPin = 5;00 21.
18. **char** initial\_password[4] = {'1', '2', '3', '4'}; // Variable to store initial password000
19. String tagUID = "29 B9 ED 23"; // String to store UID of tag. Change it with your tag's UID00
20. **char** password[4]; // Variable to store users password0000
21. boolean RFIDMode = **true**; // boolean to change modes000
22. boolean NormalMode = **true**; // boolean to change modes000
23. **char** key\_pressed = 0; // Variable to store incoming keys000 28. uint8\_t i = 0; // Variable used for counter000

29.

30. // defining how many rows and columns our keypad have00 31. **const** byte rows = 4;00

32. **const** byte columns = 4;0000 33.

1. // Keypad pin map0000
2. **char** hexaKeys[rows][columns] = {0000
3. {'1', '2', '3', 'A'},0000
4. {'4', '5', '6', 'B'},0000
5. {'7', '8', '9', 'C'},0000
6. {'\*', '0', '#', 'D'}0000
7. };0000 41.
8. // Initializing pins for keypad0000
9. byte row\_pins[rows] = {A0, A1, A2, A3};0000
10. byte column\_pins[columns] = {2, 1, 0};0000 45.
11. // Create instance for keypad0000
12. Keypad keypad\_key = Keypad( makeKeymap(hexaKeys), row\_pins, 0000 column\_pins, rows, columns); 0000
13. 0000
14. **void** setup() {0000
15. // Arduino Pin configuration0000
16. pinMode(buzzerPin, OUTPUT); 0000
17. pinMode(redLed, OUTPUT); 0000 53. pinMode(greenLed, OUTPUT); 0000

54.

1. sg90.attach(servoPin); //Declare pin 8 for servo0000
2. sg90.write(0); // Set initial position at 0 degrees0000
3. 0000
4. lcd.begin(); // LCD screen0000
5. lcd.backlight();0000
6. SPI.begin(); // Init SPI bus0000
7. mfrc522.PCD\_Init(); // Init MFRC5220000

62.

1. // Arduino communicates with SIM900 GSM shield at a baud rate of 192000000
2. // Make sure that corresponds to the baud rate of your module0000
3. SIM900.begin(19200); 0000

66.

1. // AT command to set SIM900 to SMS mode0000
2. SIM900.print("AT+CMGF=1\r");0000
3. delay(100); 0000
4. // Set module to send SMS data to serial out upon receipt0000
5. SIM900.print("AT+CNMI=2,2,0,0,0\r");0000
6. delay(100); 0000
7. 0000
8. lcd.clear(); // Clear LCD screen0000
9. }0000 76. 0000



121.digitalWrite(greenLed, LOW); 0000

122.0000

123.lcd.clear();0000

124.lcd.print("Enter Password:"); 0000

125.lcd.setCursor(0, 1); 0000

126.RFIDMode = **false**; // Make RFID mode false0000

127.}0000

128.0000

129.**else**0000

130.{0000

131.// If UID of tag is not matched. 0000

132.lcd.clear();0000

133.lcd.setCursor(0, 0); 0000

134.lcd.print("Wrong Tag Shown");0000

135.lcd.setCursor(0, 1); 0000

136.lcd.print("Access Denied");0000

137.digitalWrite(buzzerPin, HIGH); 0000

138.digitalWrite(redLed, HIGH); 0000

139.send\_message("Someone Tried with the wrong tag \nType 'close' to halt the system."); 0000 140.delay(3000); 0000

141.digitalWrite(buzzerPin, LOW); 0000

142.digitalWrite(redLed, LOW); 0000

143.lcd.clear();0000

144.}0000

145.}0000

146.0000

147.// If RFID mode is false, it will look for keys from keypad0000

148.**if** (RFIDMode == **false**) {0000

149.key\_pressed = keypad\_key.getKey(); // Storing keys0000

150.**if** (key\_pressed) 0000

151.{0000

152.password[i++] = key\_pressed; // Storing in password variable0000 153.lcd.print("\*");0000

154.}0000

155.**if** (i == 4) // If 4 keys are completed0000

156.{0000

157.delay(200); 0000

158.**if** (!(strncmp(password, initial\_password, 4))) // If password is matched0000

159.{0000

160.lcd.clear();0000

161.lcd.print("Pass Accepted");0000

162.sg90.write(90); // Door Opened0000

163.digitalWrite(greenLed, HIGH); 0000

164.send\_message("Door Opened \nIf it was't you, type 'close' to halt the system."); 0000 165.delay(3000); 0000

166.digitalWrite(greenLed, LOW); 0000

167.sg90.write(0); // Door Closed0000

168.lcd.clear();0000

169.i = 0; 0000

170.RFIDMode = **true**; // Make RFID mode true0000

171.}0000

172.**else** // If password is not matched0000

173.{0000

174.lcd.clear();0000

175.lcd.print("Wrong Password");0000

176.digitalWrite(buzzerPin, HIGH); 0000

177.digitalWrite(redLed, HIGH); 0000

178.send\_message("Someone Tried with the wrong Password \nType 'close' to halt the system."); 0000

179.delay(3000); 0000

180.digitalWrite(buzzerPin, LOW); 0000

181.digitalWrite(redLed, LOW); 0000

182.lcd.clear();0000

183.i = 0; 0000

184.RFIDMode = **true**; // Make RFID mode true0000

185.}0000 186.}0000

187.}0000

188.}0000

189.}0000

190.0000

191.// Receiving the message0000

192.**void** receive\_message()0000

193.{0000

194.**char** incoming\_char = 0; //Variable to save incoming SMS characters0000

195.String incomingData; // for storing incoming serial data0000

196.0000

197.**if** (SIM900.available() > 0) 0000

198.{0000

199.incomingData = SIM900.readString(); // Get the incoming data. 0000 200.delay(10); 0000

201.}0000

202.0000

203.// if received command is to open the door0000

204.**if** (incomingData.indexOf("open") >= 0) 0000

205.{0000

206.sg90.write(90); 0000

207.NormalMode = **true**; 0000

208.send\_message("Opened");0000

