**Task 4**

Project Title: **Keypad Security system using SD card interface**

**Member 1 Member 2**

Name: Anshul Surpaithankar Name: Anamay Mahesh Belekar

Div. & Roll no.: Q-14 Div. & Roll no.: Q-3

GR No.: 1710294 GR No.: 1710311

E-ID:[anshul.surpaithankar17@vit.edu](mailto:anshul.surpaithankar17@vit.edu) E-ID:anamay.belekar17@vit.edu

**Member 3**

Name: Aditya Wattamwar

Div. & Roll no.: I-40

GR No.: 1710515

E-mail ID:[aditya.wattamwar17@vit.edu](mailto:aditya.wattamwar17@vit.edu)

* Introduction

The keypad security lock system is basically a door lock system, given every user of the door a password, which on entering correct password opens the door and stores that password in the SD card provided with the system and hence giving the information about who has accessed the door and who has not. The password of the keypad lock may not be a single one, but will have passwords equal to number of operators operating it, if they are permitted to access the door, providing every user a secret and unique password which will be assigned to that user only, while all the passwords are stored in the SD card itself provided with it. According to the password entered, the solenoid connected to output of it will be energised and de-energised and the lock will be actuated accordingly. In worst case, if the SD card is not detected by the micro-controller, the entered password is then compared with the arrays initialised in the memory of micro-controller, if that password is verified, then the user is again asked to enter a unique code on entering correctly on which the solenoid lock gets opened. The password entered in this case is stored back to the memory of the SD card when re-installed successfully.

* Components

1. 4 X4 keypad (or 4X3)
2. SD card and SD card module (Arduino)
3. IC ATMEGA328P (or Arduino Genuino/Uno )
4. 16 X 2 LCD display
5. Buzzer
6. Relay 12V
7. solenoid lock 12V
8. 12V battery
9. Connecting wires, resistances, quartz crystal, capacitors, etc (all required for connection of IC atmega328P)

* **Algorithm**

1. **For keypad**

The execution of this project stands firmly on the process of recognising the keys of 4X4 keypad correctly by the micro-controller. In the total of 16 keys on the 4X4 keypad, we are using only 11 keys for this execution (‘0-9’ for the numbers and the key ‘\*’ for resetting purpose –in case the user fails to enter the password correctly). For this execution and receiving correct data from the keypad, following steps are followed.

In the 4X4 keypad, the pins of micro-controller are given to columns of the keypad and the pins are set as ‘OUTPUT’, while the pins connected to the rows of the keypad are also connected to the micro-controller, but are declared as ‘INPUT’ pins. Since the all the pins of the micro-controller atmega328P are set pulled up internally, through the OUTPUT declared pins (column pins of the keypad), the IC continuously sends low value to each of the column pins and checks at the INPUT pins whether any pin is low or not. Since the pins are internally pulled up, if we don’t press any key, the value that the IC gets at the every input pin is always high (logic ‘1’). When we press a key, the low signal present at the column pins of the keypad will be delivered to the respective input pin of the respective row. In this manner, the conditions are put for all the 10 keys amongst 4 rows and 3 columns, so as to correctly identify the number associated with the particular pin.

3

2

1

6

5

4

9

8

7

0

\*

As mentioned earlier, the IC continuously sends low value to the all the columns one by one to check if any key is pressed. Here, if the key 5 is pressed and after it whenever the IC sends low value to the second column, the input pins of the IC will detect low value at the input pin corresponding to second row and according to the code written for the condition, the digit ‘5’ will be stored accordingly and further execution will be done. The same is done in case of all other keys of the keypad.

1. **For LCD display**

The LCD display requires total of 9 pins for its basic operation, which are four data pins (D4, D5, D6, D7), +VCC, ground, enable and RS and the potentiometer (variable resistance terminal) pin required for changing the contrast of the display. In addition to these pins, to enable the back light of the LCD display pin no. 15 and 16 can be used.

Whenever the system is in use, the LCD prints ‘Password Please’. When we start entering the password, the display prints no. of ‘\*’ equal to the number of digits entered until then. For the correct password, the display prints ‘correct’ and for wrong one, it prints ‘wrong’. In the middle of process of entering the password, if the user wants to re-enter the password, the user can do it using the‘\*’ button on the keypad, which will reset it, so that the user can re-enter the code.

Also, for back-light of the display pin no. 15 and 16 can be used.

1. **SD card and SD card module**

As mentioned earlier, the system can have more than one password (depending upon the number of users using it), so all the passwords valid for the system are stored in a text file (test.txt) in the SD card through the SD card module. Whenever a user enters a 7 digit password, the micro-controller accepts it and compares it with the number of passwords stored in test.txt file in the SD card. All the communication between the micro-controller and the SD card module is done through the SPI (Serial Peripheral Interface) communication. If the entered password is matched with any of the passwords present in the list in test.txt file, the micro-controller sends data to the LCD display to print ‘correct’ to the screen and makes an IC pin high which is connected to the coil of relay which will be actuated and the solenoid lock connected to the output of it will be energised and the door will open. Also, at the same time, that correct entered password is stored in another text file (test1.txt), so that one will come to know which password was entered and hence will come to know who had accessed the system. For the wrong password, the LCD prints ‘wrong’ but the entered password is not stored in the text file, thus not allowing the relay and hence the lock to actuate.

1. **Pin Connections (peripheral pins 🡨🡪 Arduino pins)**
2. (Keypad) R1 ~ R4 🡨🡪 10,9,2,7
3. (Keypad) C1 ~ C3 🡨🡪 3,5,6
4. (LCD) D4 ~ D7🡨🡪 A2,A3,A4,A5
5. (LCD) RS 🡨🡪 A0, EN 🡨🡪 A1, GND, VCC, RW 🡨🡪 GND
6. (SD card module) CS 🡨🡪 4, SCK 🡨🡪13, MOSI🡨🡪11, MISO 🡨🡪 12, VCC 🡨🡪 +5V, GND
7. Solenoid 🡨🡪8
8. All remaining connections for biasing of the IC.

The user enters his/her password through the keypad. The entered 7-digit password is stored in an array. Since we have to compare this array with the contents which is recognised by the micro-controller as characters in the text file in the SD card, the entered password is also stored in form of characters only. For example, if key ‘4’ is pressed then the content stored in the array will be a[0]=’4’ but not a[0]=4. So it will be easier for the micro-controller to compare directly two char values instead of comparing a char and integer value.

Then, the array is compared with the various passwords stored in test.txt file line by line. If the array is matched with any of the password stored in that file, then the LCD displays ‘correct’ and that entered password is stored in another text file test1.txt. At the same time, the pin associated with the solenoid goes high and the relay energises thus actuates the solenoid. Also, by some mechanical mismatch, if micro-controller fails to detect SD card, the entered password is compared internally with the arrays, storing the number of passwords, if that password is matched successfully, then it asks user to enter unique code which is known to every user operating the system. As soon as the SD card detection problem is fixed, those entered passwords are again stored back to the SD card. Hence the fail of SD card detection will not affect the functioning of the system.

1. **Insights-**

* For connections of the keypad with the arduino board, the row and column pins of the keypad should be identified carefully.
* For testing purpose, to work with the keypad in ‘Rudra board’, along with connection of 5V supply, the ground of that board should be made common with the UNO board (or the IC 328P).
* While accepting the input from the user through the keypad, sufficient delay should be given in the code after accepting the input once, otherwise, since the frequency of the IC is very high (8 MHz), the IC will accept the same input multiple times as long as we press that particular key. I.e. make sure when a key pressed once, should be accepted by the IC only once.
* All the connections to the LCD display mentioned above should be made for proper functioning of that display.
* The SD card should be given proper input voltage according to its requirement (5V or 3.3V), if not given the IC will fail to detect the SD card. The SD card module that we used had operating voltage of 5V.
* **Results**

The project was successful in storing the correct passwords entered through the keypad, in the SD card, displaying the result on the LCD display and hence opening the solenoid lock accordingly.

* **References**
* <https://www.arduino.cc/en/Reference/SD>
* <https://www.instructables.com/id/Micro-SD-Card-Tutorial/>
* Arduino examples> SD card