NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY
NAAC ACCREDITED 'A+' GRADE

[SPECIAL ASSIGNMENT REPORT]

PASSWORD CONTROLLED DOOR LOCK SYSTEM USING ARDUINO

SUBJECT: MICROPROCESSOR & MICROCONTROLLER

COURSE CODE: 2EC404

SUBMITTED TO: DR. JAYESH PATEL

SUBMISSION DATE: 06/05/2023

PREPARED BY:

22EC044 - JAIMIN GULALE

❖ INTRODUCTION:

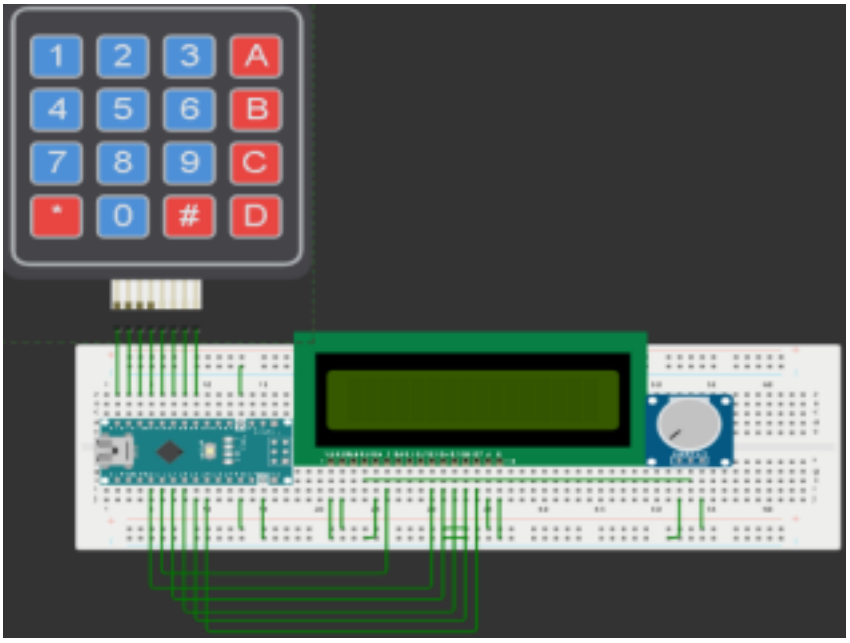
In this Project we have made a Password controlled door lock system by interfacing Arduino Nano with 4×4 Keypad to enter the password. We have also connected a 16*2 LCD to enter user defined password and to display if the input password is right or wrong. With this Project we can build a security system which works with password. In day-to day life everything is becoming simple and advanced, previously to lock something we used to have a padlock, combination locks. But due to increase in thefts and technology, new type of locks such as electronic locks, smart locks were invented and people are using

them widely to protect their assets. From these couple of years Fingerprint based locks, RFID based locks, Smart phone app-based locks, Face recognition locks are trending, Password based door locks are one of them.

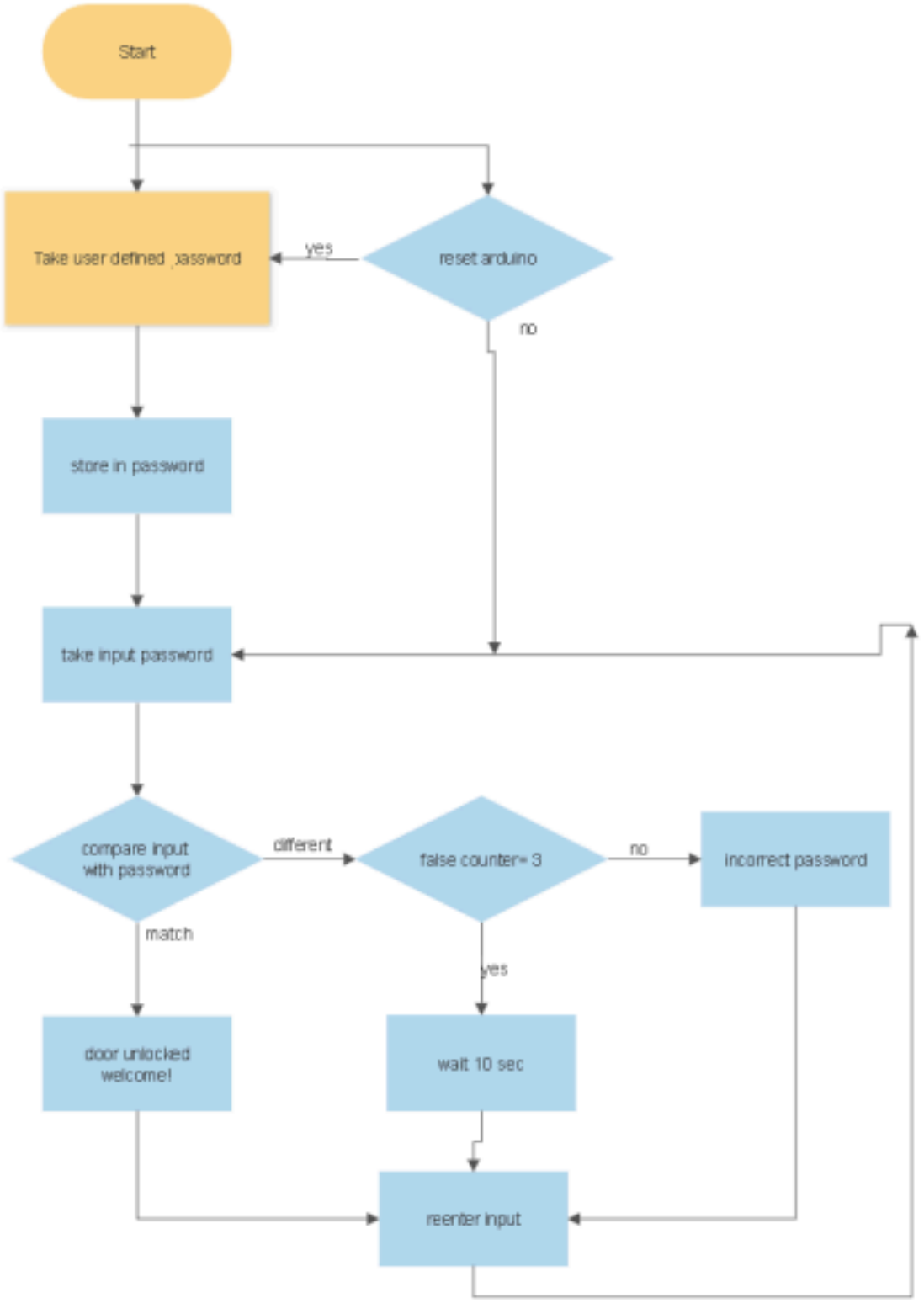
❖ SPECIFICATIONS OF COMPONENETS:

SR N0.	COMPONENTS	QUANTITY	COST
1.	BREADBOARD	1	100₹
2.	ARDUINO NANO	1	300₹
3.	16x2 LCD DISPLAY	1	140₹
4.	4x4 KEYPAD	1	120₹
5.	10K POTENTIOMETER	1	5₹
6.	CONNECTING WIRES	20	20₹
7.	BUCK-STRIP	1	10₹

❖ CIRCUIT DIAGRAM:



❖ FLOWCHART:



❖ **ARDUINO CODE & WORKING:**

```
// Include the libraries for the LCD and keypad
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <EEPROM.h>

// Define the pins for the LCD and keypad
const byte ROWS = 4, COLS = 4; //4*4 keypad
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte count = 0;//represents no of elements entered for displaying each as * on LCD
byte false_counter=0;//represents no of false attempts to punish for !0 seconds byte
rowPins[ROWS] = {12,11,10,9}; //Digital pins
byte colPins[COLS] = {8,7,6,5}; //Digital pins

// Define the password and variables
String password = ""; // the user defined password
String input = ""; // the user's input
bool unlocked = false; // whether the door is unlocked or not

// Define the address in EEPROM to store the password
int address = 0;
// Initialize the LCD and keypad
LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);
Keypad keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup() {
  // Set up the LCD
  lcd.begin(16, 2);
  lcd.print("Enter new password:");
  // Set up the keypad
  keypad.setDebounceTime(50);
}

void loop() {
  // Read the keypad
  char key = keypad.getKey();

 // If a key is pressed then only add it to the input
  if (key) {
    input += key;
    lcd.setCursor(count, 1); // move cursor to show each new char as secret '*'
    lcd.print("*");
    count++; // increment data array by 1 to store new char, also keep track of the number of
chars entered
  }
```

```
 // Check if the input is complete
  if (input.length() == 4) {

// If the password has not been set yet, store the input as the new password
    if (password == "") {
      password = input;
      storePassword(password);
      lcd.clear();
      lcd.print("Password set");
      delay(1000);
      lcd.clear();
      lcd.print("Enter password:");
      input = "";
    }

// Otherwise, directly check if the input matches the
    password else {
      // If the input matches, unlock the door
      if (input == password) {
        unlocked = true;
        false_counter=0;
        lcd.clear();
        lcd.print("Door unlocked");
        delay(2000);
        lcd.clear();
        lcd.print("Enter password:");
        input = ""; //make input null string again to retake input
       }

// Otherwise, display an error message and reset the input
      else {
        false_counter++;
        if(false_counter==3){
          lcd.print("wait 10 sec");
          delay(10000); //to punish the user 10 seconds for 3 wrong attempts
          false_counter=0;
         }
        lcd.clear();
        lcd.print("Incorrect");
        delay(1000);
        lcd.clear();
        lcd.print("Enter password:");
        input = ""; //make input null string again to retake input
       }
      count=0;
    }
  }
  // If the door is unlocked, display a message
```

```
  if (unlocked) {
    lcd.clear();
    lcd.print("Welcome!");
    delay(2000);
    lcd.clear();
    lcd.print("Enter password:");
    unlocked = false;
  }
}

// Function to store the password in EEPROM
void storePassword(String password) {
  for (int i = 0; i < 4; i++) {
    EEPROM.write(address + i, password[i]);
  }
}
```

OUTPUT SIMULATION VIDEO:
https://drive.google.com/drive/folders/1qUdRhm30vmKSho6PWFcWElCEnDjC60DW

❖ APPLICATIONS:

- Residential: A password-controlled door lock system can be installed at the entrance of a house, apartment or condominium. This can ensure that only authorized individuals can enter the premises, thus enhancing the security of the occupants.
- Commercial: In commercial buildings, a password-controlled door lock system can be installed to restrict access to certain areas. For example, only employees with the correct password can enter the server room or management office. This can help prevent theft, vandalism, or unauthorized access to sensitive information.
- Healthcare: Password-controlled door lock systems can be installed in hospitals, clinics, and other healthcare facilities to ensure that only authorized personnel have access to patient records and medication storage areas.
- Educational: Schools, colleges, and universities can use password-controlled door lock systems to restrict access to classrooms, laboratories, and other sensitive areas. This can prevent unauthorized entry and protect students and staff from harm.
- Government: Government facilities such as embassies, courthouses, and military bases can benefit from password-controlled door lock systems. These systems can restrict access to sensitive areas, prevent unauthorized entry, and enhance overall security.

❖ SUMMARY:

Overall, Arduino based door lock system is very easy to use and to configure because of low cost, compatibility and scalability of the Arduino platform.