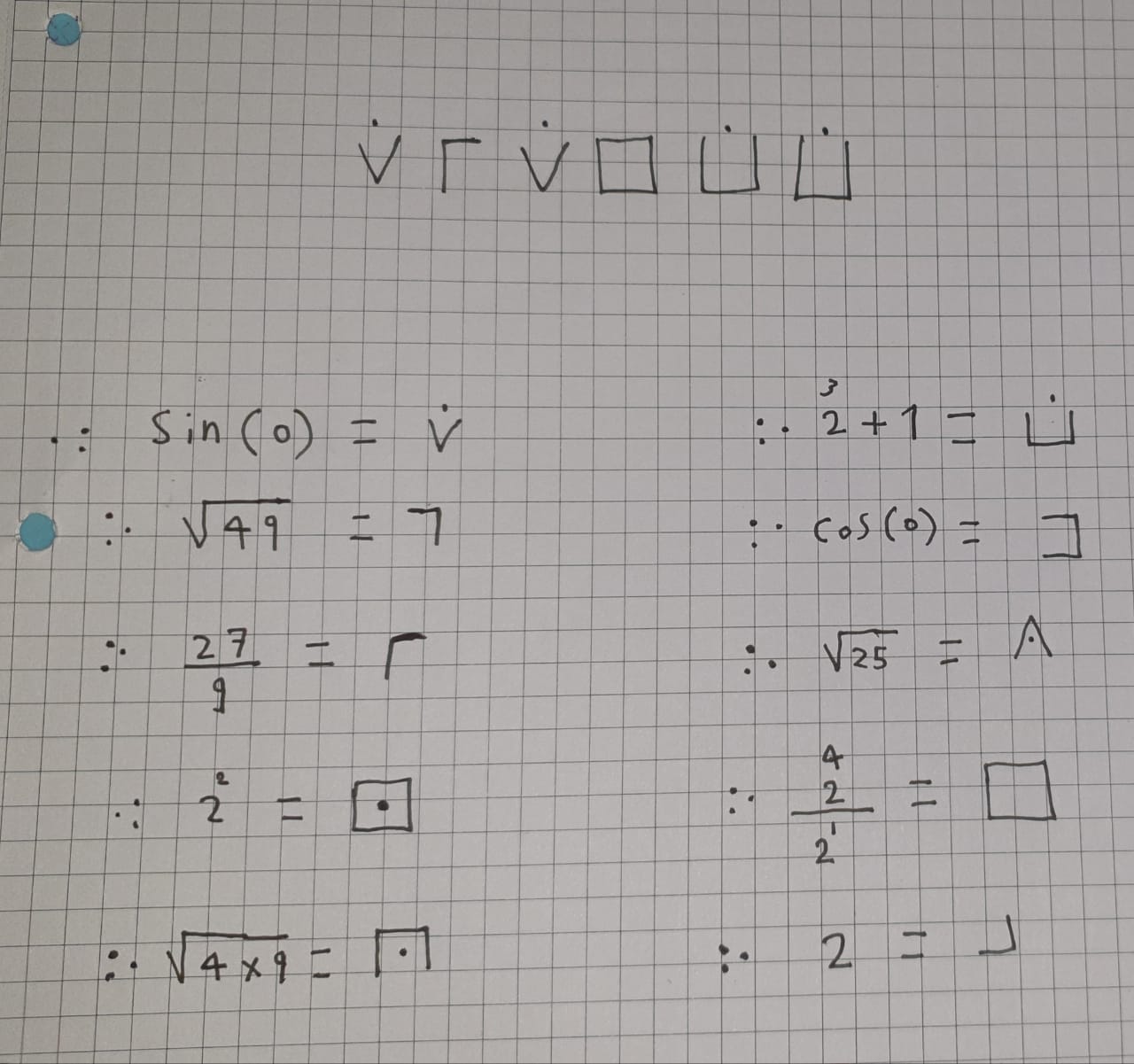
The main idea of this task is decipherment, the player is asked to solve a problem to get the password that he has to enter in the application for doing this task , in the lab the player will find a paper that include a symbols and he is asked to decode the symbols into a number (instruction will be provided in the paper as it shown) , after decoding and getting the number the player have to enter the value into the Arduino , a keypad has been placed for this purpose , if the player entered the correct code a green led will be turned on and he will get the password to be entered in the application , but if the value was wrong a red led will be turned on and he must wait 5 minutes before he gain access to enter another code , also keep in mind that the imposter can sabotage this task by entering a special code for the imposter , a yellow led will be turned on and it indicate that an imposter has sabotaged this task and the player have to wait till it goes off ( it will take 30 seconds) .



#include <Keypad.h>

#include <LiquidCrystal\_PCF8574.h>

#include <Wire.h>

LiquidCrystal\_PCF8574 lcd(0x3F);

const byte ROWS = 4;

const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {

{'1', '2', '3', 'A'},

{'4', '5', '6', 'B'},

{'7', '8', '9', 'C'},

{'\*', '0', '#', 'D'}

};

byte rowPins[ROWS] = {9, 8, 7, 6};

byte colPins[COLS] = {5, 4, 3, 2};

Keypad keypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS );

int green = 13;

int red = 11;

int yellow = 12;

const String password = "030899"; // change your password here

const String imposterPassword = "666";

String input\_password;

void setup() {

Wire.begin();

Wire.beginTransmission(0x3F);

int error = Wire.endTransmission();

Serial.begin(9600);

pinMode(green, OUTPUT);

pinMode(red, OUTPUT);

pinMode(yellow, OUTPUT);

if (error == 0) {

Serial.println(": LCD found.");

lcd.begin(16, 2); // initialize the lcd

lcd.setBacklight(255);

} else {

Serial.println(": LCD not found.");

}

lcd.begin(16, 2);

lcd.print("Your Password is : ");

input\_password.reserve(32); // maximum input characters is 33, change if needed

}

void loop() {

char key = keypad.getKey();

digitalWrite(yellow, LOW);

if (key) {

Serial.println(key);

if (key == '\*') {

input\_password = ""; // clear input password

lcd.clear();

lcd.print("password has");

lcd.setCursor(0, 1);

lcd.print("been cleared");

Serial.println("password has been cleared: ");

digitalWrite(green, LOW);

}

else if (key == '#') {

if (password == input\_password) {

Serial.println("maradona");

lcd.clear();

lcd.print("Correct! Pass:");

lcd.setCursor(0, 1);

lcd.print("maradona");

digitalWrite(green, HIGH);

}

else if (imposterPassword == input\_password) {

lcd.clear();

Serial.println("the KeyPad has been Sabotaged , wait 30 seconds");

lcd.print("task Sabotaged");

lcd.setCursor(0, 1);

lcd.print("Wait 30s");

digitalWrite(yellow, HIGH);

delay(10000);

delay(10000);

delay(10000);

lcd.clear();

digitalWrite(yellow, LOW);

}

else {

lcd.clear();

Serial.print("Wrong password, Wait for 5 seconds.");

lcd.print("Wrong password.");

lcd.setCursor(0, 1);

lcd.print("Wait for 5s");

digitalWrite(red, HIGH);

delay(5000);

digitalWrite(red, LOW);

lcd.clear();

}

input\_password = ""; // clear input password

}

else {

lcd.clear();

lcd.print("Your Password is : ");

input\_password += key; // append new character to input password string

lcd.setCursor(0, 1);

lcd.print(input\_password);

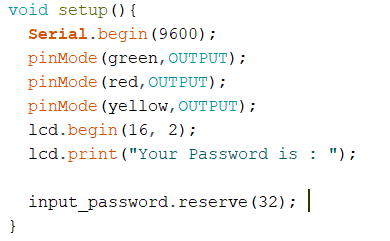
}

}

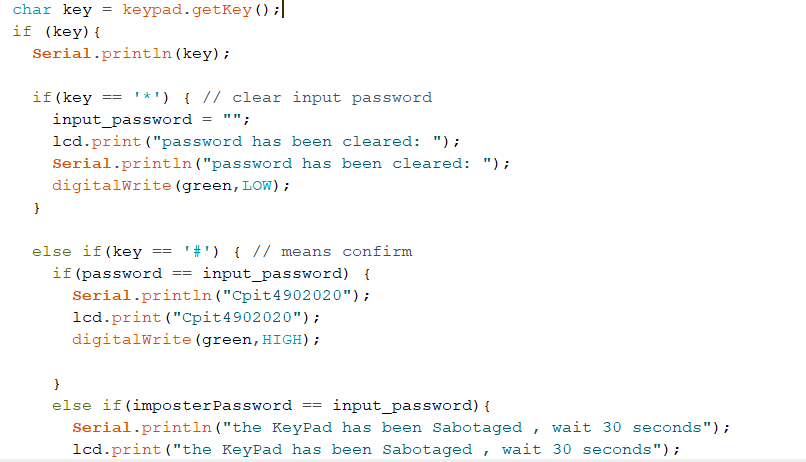
}

* Graphical user interface, text, application

  Description automatically generatedbefore the setup() we are declaring variables and assigning some of them toward a particular pin in the Arduino , hexaKeys is a 4\*4 two dimensional array which include all the values of the keypad , the password is set to be “ 030899 “ while the imposter is set to be “666” , input\_Password is a String variable that will store the values that the player enters ( it takes it as a char then add it to this string )

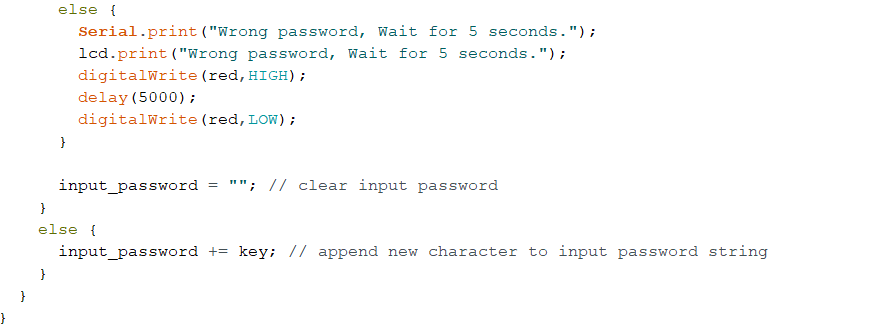


* In the setup() , all the led are assigned as an output , the lcd will be assigned to particular format and it will print to the player “ your password is “ , the input\_password (string that store the keys the user press ) will reserve a maximum of 33 character it can’t take more than that



The variable key will read the button that the user press , the button “ \* “ is for clearing the input\_password string , the point of this button is to take into consideration the factor of human error , the player can delete what he entered and enter the password from the beginning .

As for the “ # “ it means confirm the code , if it was right the green led will go high and it will give the player the password to be entered in the application , if it was wrong red goes high and the player must wait 5 minutes before he can access the Arduino again , and the final probability is that the entered value was the imposter code , the yellow led will go high and he wouldn’t gain access to use the Arduino for 30 seconds , after that the input\_password will be cleared.



If the user didn’t press \* or # the entered key (char) will be stored in the string input\_password