

I am working on a project that utilizes Arduino Code to control a small LCD screen. When i run it it does not work. Can you please help me find error in code?

I have included below, The code that I have so far, a picture of the board and some feedback that I have received but didn't work when I implemented it. Is there an error in my code? Am I missing something?

The project involves entering a 16 digit pass code using two buttons and seeing various results on a small LCD screen. In its current state the introduction displays on the screen but pressing the buttons does nothing

Code that I have so far:

```
#include <LiquidCrystal.h> //the LC library
LiquidCrystal lcd(13, 12, 11, 10, 9, 8); // pin locations for the LCD
```

```
//set pins for the buttons
int button[] = {5, 7};
```

```
int pressedButton = 2; //a variable to remember which button is
being pressed. 2 is the value if no button is being pressed.
int counter = (0); //tracks which character of the password the
user is on
int buttonSequence[16] = {0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0,
0}; //make an array of the characters that will be the password
int digitsInCode = 16;
```

```
boolean programStart = false; //variable to tell the program
whether or not to play the start sequence
```

//_____FUNCTIONS_____

//START SEQUENCE

void startSequence() {

 lcd.clear(); //clear the display

 lcd.setCursor(0, 0); //set the cursor to the 0,0 position (top left corner)

 lcd.print("SECURITY"); //print SECURITY starting at that position

 lcd.setCursor(0, 1); //set the cursor to the 0,0 position (bottom left corner)

 lcd.print("CHECK"); //print CHECK starting at that position

 delay(3000); //wait for 3 seconds

 lcd.clear(); //clear the display

 lcd.setCursor(0, 0); //set the cursor to the 0,0 position (top left corner)

 int counter = (0);

}

//CHECK WHICH BUTTON IS PRESSED

int buttonCheck() {

 //check if any buttons are being pressed

 if (digitalRead(button[0]) == LOW) {

 return 0;

 } else if (digitalRead(button[1]) == LOW) {

 return 1;

 } else {

 return 2; //this will be the value for no button being pressed

 }

}

//CORRECT PASSWORD SEQUENCE

```
void correctPass() {
```

```
  lcd.setCursor(0, 1);  
  lcd.print("Correct!");
```

```
  delay(2000);  
  lcd.clear();
```

```
  lcd.setCursor(0, 0);  
  lcd.print("CONGRATS");  
  lcd.setCursor(0, 1);  
  lcd.print("YOU GOT IT RIGHT!");  
}
```

```
//WRONG SEQUENCE
```

```
void wrongSequence() {
```

```
  lcd.setCursor(0, 1); //move the cursor to the first space of the  
  bottom row
```

```
  lcd.print("Incorrect"); //print Incorrect at that position
```

```
  delay(3000); //wait for 3 seconds
```

```
  programStart = false; //reset the program so that the start  
  sequence will play again.
```

```
}
```

```
//_____
```

```
void setup() {
```

```
lcd.begin(16, 2); //tell the lcd library that we are using a display
that is 16 characters wide and 2 characters high
lcd.clear(); //clear the display
```

```
//set the button pins as inputs
pinMode(button[0], INPUT_PULLUP);
pinMode(button[1], INPUT_PULLUP);
```

```
int counter = (0); //make a variable to count the correct characters
entered by the user
```

```
}
```

```
void loop() {
```

```
if (programStart == false); { //if the password program hasn't
started yet
startSequence(); //play the start sequence
counter = 0; //reset the counter
delay(1500); //wait a second and a half
programStart = true; //set programStart to true so that this
sequence doesn't start again
}
```

```
lcd.setCursor(0, 0); //set the cursor to the 0,0 position (top left
corner)
lcd.print("Enter Password:"); //print Enter Password: starting at
that position
```

```
for (int i = 0; i <= counter; i++) {
```

```
while (programStart == true); //Loop until the program is false
```

```
pressedButton = buttonCheck(); //every loop check to see which  
button is pressed
```

```
if (pressedButton < 2) { //if a button is pressed... (2 means that no  
button is pressed)
```

```
  lcd.setCursor(0, 1); //move the cursor to the first space of the  
  bottom row
```

```
  lcd.print(pressedButton); //print 0 at that position
```

```
  if (pressedButton == buttonSequence[i]) { //if the button matches  
  the button in the sequence
```

```
    delay(250); //leave the 0 on for a moment
```

```
    lcd.clear();
```

```
    break;
```

```
  }
```

```
  else { //if the wrong key is pressed
```

```
    wrongSequence();
```

```
    break;
```

```
  }
```

```
  }
```

```
  else {
```

```
    lcd.clear();
```

```
  }
```

```
}
```

```
if (programStart == true) {
```

```
  counter = counter + 1; //increase the round number by 1
```

```
  if (counter >= digitsInCode) { //if the user has gotten to the 16th  
  character
```

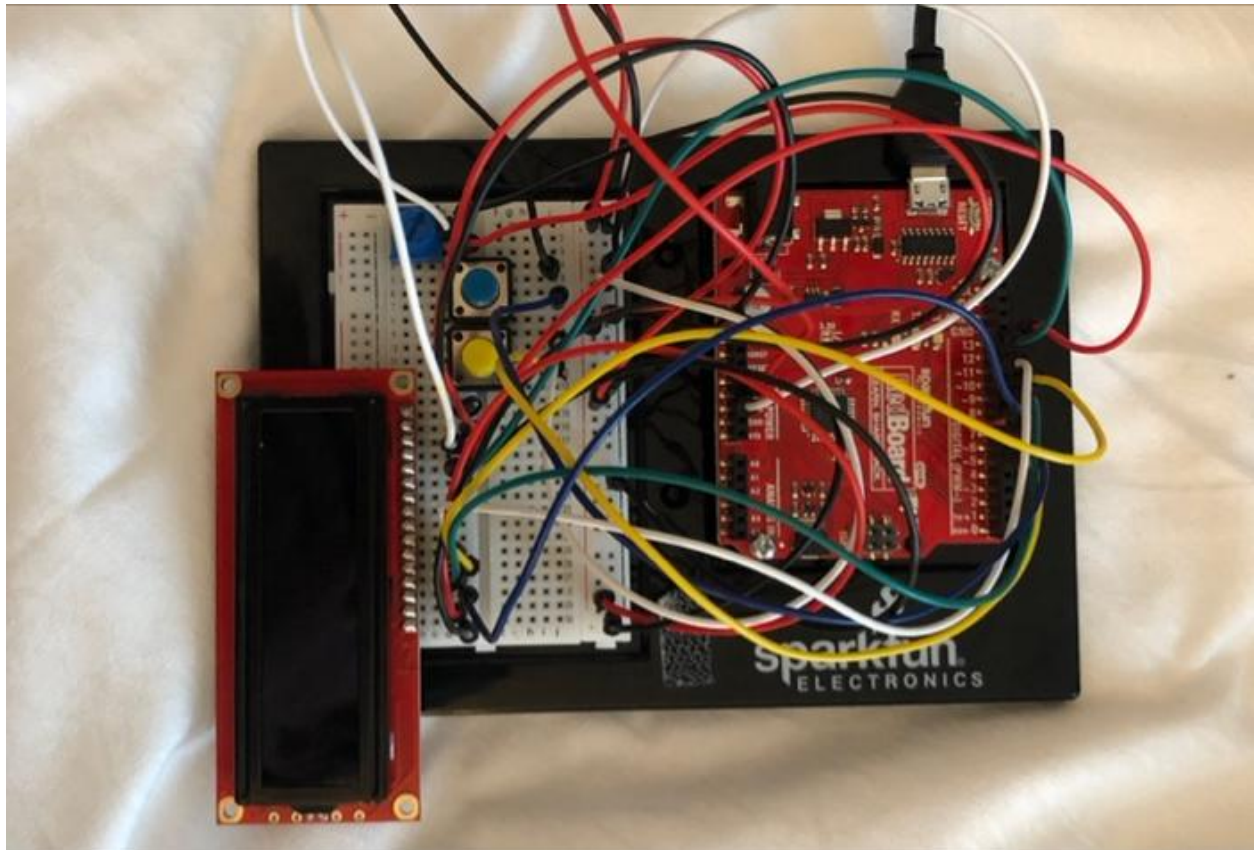
```
    correctPass();
```

```
  }
```

```
delay(500); //wait for half a second  
}  
}
```

```
// The password: 0110000101101100
```

THE BOARD



I received the following feedback already, but it didn't work to fix the problem. Please let me know any other suggestions:

To get the buttons to do something, you need to add some code to the **loop** function that will check if a button is pressed and respond accordingly.

Here is an example of how you could modify the **loop** function to check for button presses and respond to them:

```
void loop() {  
    // Check which button is currently being pressed  
  
    int button = buttonCheck();  
  
    // If a button is being pressed  
  
    if (button != 2) {  
        // If the correct button is being pressed  
  
        if (button == buttonSequence[counter]) {  
            // Increment the counter  
  
            counter++;  
  
            // If the password has been entered correctly  
  
            if (counter == digitsInCode) {  
                // Play the correct password sequence
```

```
correctPass();
```

```
// Reset the counter and program start variable
```

```
counter = 0;
```

```
programStart = false;
```

```
}
```

```
}
```

```
// If the wrong button is being pressed
```

```
else {
```

```
// Play the wrong password sequence
```

```
wrongSequence();
```

```
// Reset the counter and program start variable
```

```
counter = 0;
```

```
programStart = false;
```

```
}
```

```
}
```

```
}
```


- Explanation for step 1

One way to do this is to add a while loop inside the loop function that will continuously check if a button is being pressed, and store the value of the pressed button in the `pressedButton` variable.

Then, you can use an if statement to check if the `pressedButton` value is 0 or 1, and respond accordingly.

This implementation will check which button is currently being pressed, and if it is the correct button in the password sequence, it will increment the **counter** variable. If the password has been entered correctly, it will play the correct password sequence and reset the **counter** and **programStart** variables. If the wrong button is pressed, it will play the wrong password sequence and reset the **counter** and **programStart** variables.