

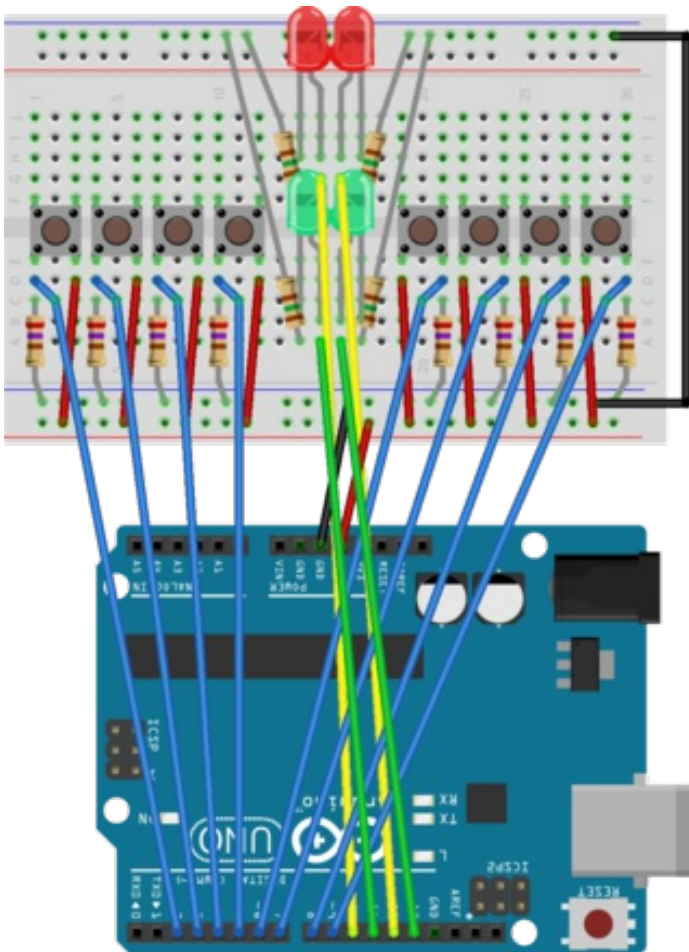
# Trivia Project

This project will demonstrate a way to use the serial connection between the Arduino and the computer to create a trivia game.

## Items needed:

- 8 270  $\Omega$  resistor
- 4 150  $\Omega$  resistors
- 2 Red LEDs
- 2 Green LEDs
- 8 buttons

## Hardware:



## Processing:

The Arduino can be used with software on the computer to allow the computer to communicate with the Arduino. One of the more common languages used to do this is called Processing.

## Obtaining the Processing sketch

1. Download a zip file containing the Processing IDE at <http://download.processing.org/processing-2.2.1-windows32.zip> (<http://download.processing.org/processing-2.2.1-windows32.zip>) .
2. Extract the zip file.
3. Navigate to the extracted folder and open processing.exe.
4. Download a zip file containing the Processing sketch at [https://www.dropbox.com/s/25e7i51345qlxfv/trivia\\_proc.zip?dl=1](https://www.dropbox.com/s/25e7i51345qlxfv/trivia_proc.zip?dl=1) ([https://www.dropbox.com/s/25e7i51345qlxfv/trivia\\_proc.zip?dl=1](https://www.dropbox.com/s/25e7i51345qlxfv/trivia_proc.zip?dl=1)) .
5. Extract the zip file.
6. In Processing, click File->Open then navigate to the folder where the sketch was extracted and open trivia\_proc.pde.
7. After the Arduino hardware and code are set up and running, start the Processing sketch.

## Code:

```
□ const int player1[] = {9, 8, 7, 6}; // A, B, C, D for player 1
const int player2[] = {5, 4, 3, 2}; // A, B, C, D for player 2

const int player1Output[] = {12, 13}; // red, green LEDs for player 1
const int player2Output[] = {10, 11}; // red, green LEDs for player 2

int player1Guess = -1; // -1 = no guess
int player2Guess = -1;

long finishTime = -1; // -1 if a question is in progress, otherwise represents the time that the current question finished

/*
 * Serial communication:
 * 0-3 = player 1 guess
 * 4-7 = player 2 guess
 * 8 = finished with question; show correct answer
 * 9 = finished with question; show next question
 */

void setup() {
  for(int i = 0; i < 4; i++) { // Set all the pin modes for the input buttons
    pinMode(player1[i], INPUT);
    pinMode(player2[i], INPUT);
  }

  for(int i = 0; i < 2; i++) { // Set all the pin modes for the LEDs
    pinMode(player1Output[i], OUTPUT);
    pinMode(player2Output[i], OUTPUT);
  }

  Serial.begin(9600); // Open a serial connection
```

```
}
```

```
void loop() {
```

```
    if(finishTime >= 0) { // Check if the current question is done and the program is waiting to go to the next question
```

```
        if(millis() - finishTime >= 2000) { // After two seconds, request the next question
```

```
            player1Guess = -1; // Reset the player guesses
```

```
            player2Guess = -1;
```

```
            digitalWrite(player1Output[0], LOW); // Reset the LEDs
```

```
            digitalWrite(player1Output[1], LOW);
```

```
            digitalWrite(player2Output[0], LOW);
```

```
            digitalWrite(player2Output[1], LOW);
```

```
            Serial.write(9); // Request the next question from the computer program over the serial connection
```

```
            finishTime = -1; // Reset finish time
```

```
        }
```

```
    return; // Don't continue with the rest of the program logic if we are waiting for the next question
```

```
}
```

```
for(int i = 0; i < 4; i++) { // Check all the player inputs for each player
```

```
    if(digitalRead(player1[i]) == HIGH) { // If player 1 is pressing a button
```

```
        if(player1Guess < 0) { // If player 1 hasn't already made a guess
```

```
            player1Guess = i; // Set player 1's guess
```

```
            Serial.write(player1Guess); // Send player 1's guess over the serial connection (player 1 guesses: 0-3)
```

```
            boolean correct = waitForResponse() == 1; // Wait for a response from the computer to see if the player's guess is correct
```

```
            if(correct) { // If the player is correct
```

```
                digitalWrite(player1Output[1], HIGH); // Turn on the green LED
```

```
                finishQuestion(); // Finish the question
```

```
            }
```

```
            else { // If the player is incorrect
```

```
                digitalWrite(player1Output[0], HIGH); // Turn on the red LED
```

```
                if(player2Guess >= 0) { // If both player's have guessed incorrectly, finish the question.  
                    finishQuestion();
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
if(digitalRead(player2[i]) == HIGH) { // Repeat the same behavior for player 2
```

```
    if(player2Guess < 0) {
```

```
        player2Guess = i;
```

```
Serial.write(player2Guess + 4); // Send player 1's guess over the serial connection (player 2 guess: 4-7)
```

```
boolean correct = waitForResponse() == 1;
```

```
if(correct) {  
    digitalWrite(player2Output[1], HIGH);
```

```
    finishQuestion();  
}
```

```
else {  
    digitalWrite(player2Output[0], HIGH);
```

```
    if(player1Guess >= 0) {  
        finishQuestion();
```

```
    }
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
void finishQuestion() {
```

```
    finishTime = millis(); // Set the finish time to the current time
```

```
    Serial.write(8); // Send 8 over the serial connection to indicate to the computer program to show the correct answer  
}
```

```
int waitForResponse() { // Pauses the program until there is a response from the computer  
    while(Serial.available() < 1) {}
```

```
    return Serial.read();  
}
```