Arduino Lesson - Pedestrian Crossing

Allow 1 Hour

Objectives:

- Learn to use a push button, measuring a 0 or 1 (binary) input state
- Learn to use comments, variables, functions, and loops in your code
- Learn how to use for loops

Overview

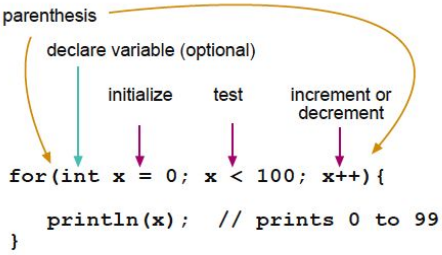
This lesson will involve the building of a pedestrian crossing. We will start by building the ‘blink’ circuit, and writing a simple sketch to control the RGB LED light. Then expand upon this circuit by adding a mini-push button.

Key Terms

- **RGB LED** = Red, Green, Blue Light Emitting Diode. Commonly used in computer screens. This light can use a combination of the colours red, green and blue (from 0 to 255 in intensity) to produce most colours.
- **Conditional Statements**
  - **For Loops** = Do something *for* a set number of times.
  - **if/else** = *If* a condition is met do x, *else* do y.
- **Comment** = information about the program written for humans clarity, it will not affect the program itself.
- **Variable** = information or attributes in your program that change, have a variable value.
- **Functions** = things your program can do, it’s functionality. For example, turning an LED on, or reading the temperature.

For Loops

The for statement is used to repeat a block of statements enclosed in curly braces. An increment counter is usually used to increment and terminate the loop. The for statement is useful for any repetitive operation. There are three parts to the for loop header:



```
for(int x = 0; x < 100; x++){
   println(x);  // prints 0 to 99
}
```

```
for (initialization; condition; increment) {
//statement(s);
}
```

The initialization happens first and exactly once. Each time through the loop, the condition is tested; if it's true, the statement block, and the increment is executed, then the condition is tested again. When the condition becomes false, the loop ends.

Example

```
// Dim an LED using a PWM pin
int PWMpin = 10; // LED in series with 470 ohm resistor on pin 10
void setup(){
  // no setup needed
}

void loop(){
   for (int i=0; i <= 255; i++){
      analogWrite(PWMpin, i);
      delay(10);
   }
}
```

First Exercise

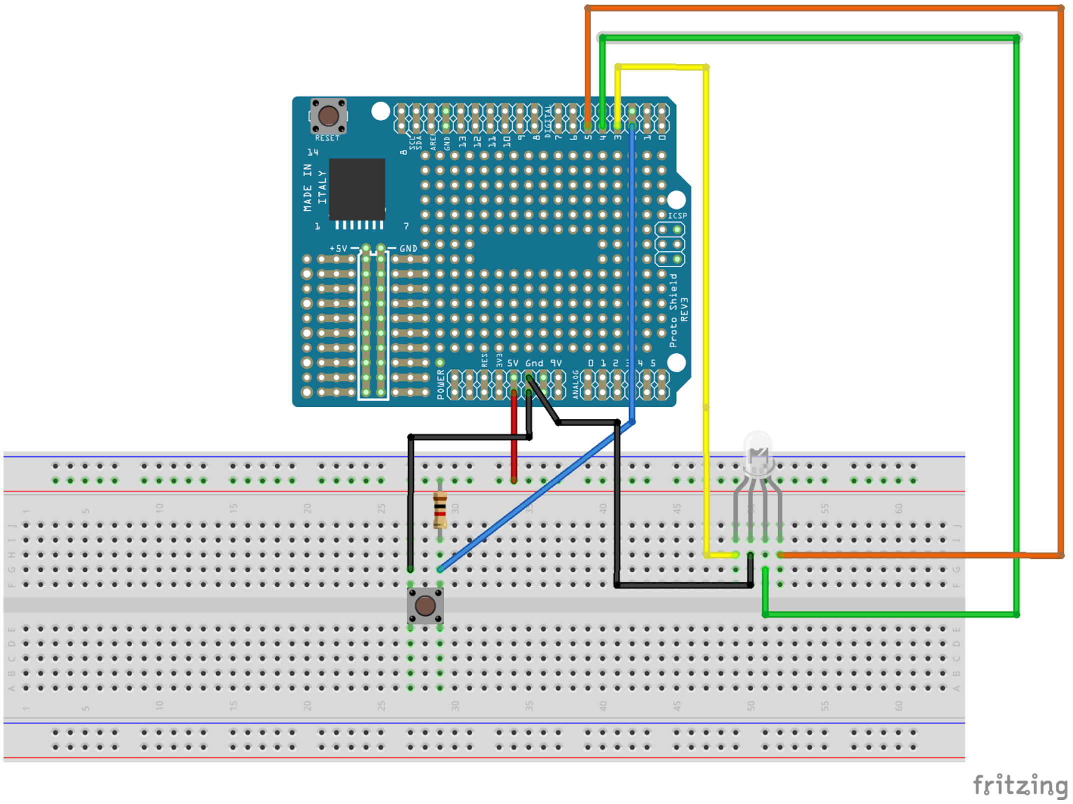
A school crossing has just been installed outside the front of a busy primary school. The crossing consists of both traffic lights for cars, and a pedestrian crossing signal for students. The project manager, Sally, has asked you to develop a pedestrian crossing system.

She would like a button to be used to control the pedestrian crossing, and estimates that roughly 5 seconds should be allowed for people to cross, with 5 addition seconds for safety.

1 - Draw the schematic for an electrical circuit to solve this problem, including the following components:

- An RGB LED
- A push button

2 - Build your Arduino using the schematic you defined above. The configuration should look something like this:



3 - Now we need to write the program’s algorithm.

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

4 - Next, let’s write a sketch (a small computer program for the Arduino) to perform the instructions (algorithm) in the form of a computer program. Have a go writing the entire program yourself, your final code should look something like this:

```
// Declare the presence of a mini push button in pin 2

int button = 2;

void setup(){
  pinMode(2,INPUT);  //Push button
  pinMode(3,OUTPUT);  // Red leg of RGB LED
  pinMode(4,OUTPUT);  // Green leg of RGB LED
  pinMode(5,OUTPUT);  // Blue leg of RGB LED
  Serial.begin(9600);  //Monitor activity on the button and provide user feedback
}
void loop(){
  button = digitalRead(2);  //button's value is whatever's received from pin 2
  Serial.println(button);

  if(button==0){             //button is pressed
   delay(2000);              //wait 2 seconds
   digitalWrite(4,LOW);      //turn red light off
   digitalWrite(5,HIGH);     //turn green light on
   delay(5000);              //wait 5 seconds
   digitalWrite(5,LOW);      //turn green off

  //for x times, so long as x is less than or equal to 5 run this code
  for(int x=0; x <= 5; x++){
   digitalWrite(4,HIGH); //turn red LED on
   delay(500);
   digitalWrite(4,LOW);  //turn red LED off
   delay(500);
  }
 }
 //if button is not pressed, display red 'wait' pedestrian signal
 else{
   digitalWrite(4,HIGH);
 }
}
```

6 - You’ve submitted your work to Sally and she’s very impressed. Well done. You’ve been discussing possible improvements to your program. You’ve decided to add helpful messages to explain what the user (the pedestrian) should be doing as your program runs. You plan to implement these messages using the Serial monitor, including:

- A message for when the user should push the button.
- A message when the button is pushed, informing the user that cars are stopping.
- A message when it’s safe to cross.
- A message when they should finish crossing.
- A second-by-second countdown of how long they have left to cross.
- A message when the cars start going again.

7 - Your code should look something like this…

```
int button = 2; // Declare the presence of a mini push button in pin 2

void setup(){
  pinMode(2,INPUT);  //Push button
  pinMode(3,OUTPUT);  // Red leg of RGB LED
  pinMode(4,OUTPUT);  // Green leg of RGB LED
  pinMode(5,OUTPUT);  // Blue leg of RGB LED
  Serial.begin(9600);  //Monitor activity on the button and provide user feedback
}
void loop(){
  button = digitalRead(2);  //button's value is whatever's received from pin 2
  Serial.println(button);

  if(button==0){            //button is pressed
   Serial.println("Cars stopping, please wait.");
   delay(2000);             //wait 2 seconds
   Serial.println("It's now safe to cross.");
   digitalWrite(4,LOW);     //turn red light off
   digitalWrite(5,HIGH);    //turn green light on
   delay(5000);             //wait 5 seconds
   digitalWrite(5,LOW);     //turn green off
   Serial.println("Please finish crossing in...");

  //for x times, so long as x is less than or equal to 5 run this code
  for(int x=0; x <= 5; x++){
   digitalWrite(4,HIGH); //turn red LED on
   delay(500);
   digitalWrite(4,LOW);  //turn red LED off
   delay(500);
   Serial.println(5-x);
  }
 }
 //if button is not pressed, display red 'wait' pedestrian signal
 else{
   digitalWrite(4,HIGH);
   Serial.println("Press button to stop cars");
 }
}
```