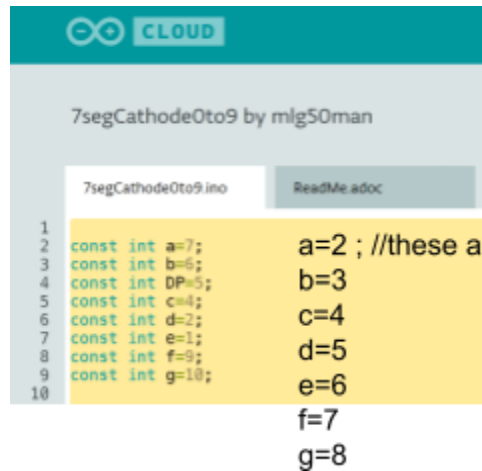


W9B: DE Arduino Variables Tutorial

Introduction

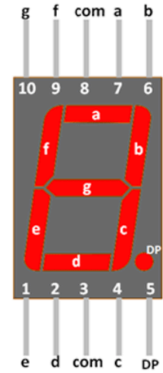
Programming languages have their own grammar called “syntax”. Programs written with the Arduino software are called Sketches. A **Sketch** (program written with **Arduino**) will contain: a title, **constants**, **variables**, `setup()` functions, and `loop()` functions.

If the syntax of a language is not followed, the program will not compile correctly. This means that no executable code will be produced. Fortunately, the **Arduino** integrated development environment (IDE) will provide error messages that will help you fix your “bad grammar”... called “syntax errors”. One of the most common syntax errors that students make is forgetting that lines of code need to end with a semicolon.



```

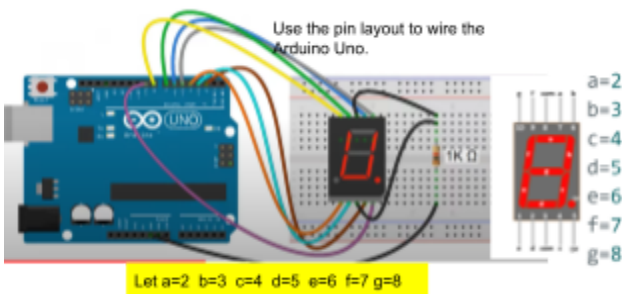
1
2 const int a=7;
3 const int b=6;
4 const int DP=5;
5 const int c=4;
6 const int d=2;
7 const int e=1;
8 const int f=9;
9 const int g=10;
10
a=2 ; //these are the correct pin connections
b=3
c=4
d=5
e=6
f=7
g=8
    
```



Equipment

Computer with Arduino Software
Recall the Common Cathode Seven Segment Display:

Procedure



Introduction: Create a “New Sketch” and enter the code:

Guzman_Count.ino

```

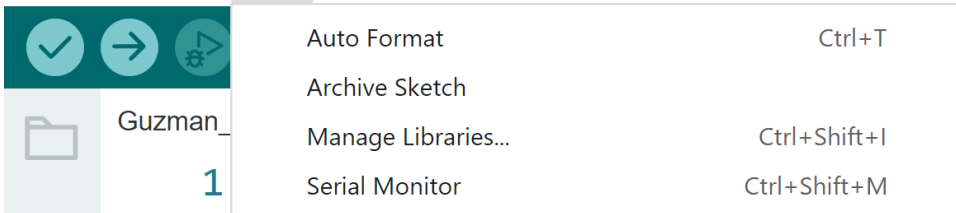
1  int countUp=0;
2  void setup()
3  {
4      // put your setup code here, to run once:
5      Serial.begin(9600);
6  }
7
8  void loop()
9  {
10     // put your main code here, to run repeatedly:
11     countUp++;
12     Serial.print(countUp);
13     delay(1000);
14 }

```

View the code using Serial Monitor located under the Tools tab:

Guzman_Count | Arduino IDE 2.1.1

File Edit Sketch Tools Help



1. Use the video provided to understand the use of variables using C++ Code. Use the Arduino to program the code and the reference guide:

<https://www.arduino.cc/reference/en/>

Use this code: <https://create.arduino.cc/editor/mlg50man/7b36c803-f0f2-41fe-a540-1dba1553a8e6/preview>

What does the C++ code do? Did you define the variables on your Arduino?

This C++ code cycles through 10 digits (0-9) on an Arduino.

2. Using the code above. Create a new code that counts down.

Attach your C++ text code below. To earn full credit make sure your code has comments. Describe what the code does.

```
// Segment patterns for digits 0-9
const byte digitPatterns[10] = {
  // Patterns for digits 0-9 (A,B,C,D,E,F,G)
  0b1111110, // 0
  0b0110000, // 1
  0b1101101, // 2
  0b1111001, // 3
  0b0110011, // 4
  0b1011011, // 5
  0b1011111, // 6
  0b1110000, // 7
  0b1111111, // 8
  0b1111011 // 9
};

// Current digit to display
int currentDigit = 0;

void setup() {
  // Set pins 2-8 as outputs for segments A-G
  for (int pin = 2; pin <= 8; pin++) {
    pinMode(pin, OUTPUT);
  }

  // Set pin 10 & 11 as input for the buttons
  pinMode(10, INPUT_PULLUP); // plus button
  pinMode(11, INPUT_PULLUP); // minus button

  // Initially display 0
  displayDigit(currentDigit);
}

void displayDigit(int digit) {
  // Validate digit is between 0-9
  if (digit < 0 || digit > 9) return;

  // Get the bit pattern for the digit
  byte pattern = digitPatterns[digit];

  // Display the digit by setting each segment
  for (int seg = 0; seg < 7; seg++) {
    // Segments are on pins 2-8
    // Check each bit in the pattern
    digitalWrite(seg + 2, (pattern & (1 << (6 - seg))) ? HIGH : LOW);
  }
}
```

```
}  
}  
  
void loop() {  
  // Read the button state  
  currentDigit = 10;  
  for (int i = 0; i < 10; i++) {  
    currentDigit = currentDigit - 1;  
  
    // Display the new digit  
    displayDigit(currentDigit);  
    delay(1000);  
  }  
}
```

3. Use the YouTube tutorial, Arduino, and your Breadboard to count any number from 0-9 using two buttons to forward to the next number. In the video Yellow=Yellow. Use the myDAQ Breadboard to use the buttons and the 7-Segment Display.

https://www.youtube.com/watch?v=V_3V2R9mm2Y&ab_channel=MouayadAldada

Insert the C++ Code below and the comments to each of the parts of the code. Naming conventions must be followed. For Example "cat" is the same as blink_LED.

```
// Seven Segment Display Control with Button Increment and Reverse  
// Pins 2-8 connected to segments A-G respectively  
// Pin 10 & 11 connected to buttons  
// Common cathode seven segment display  
  
// Segment patterns for digits 0-9  
const byte digitPatterns[10] = {  
  // Patterns for digits 0-9 (A,B,C,D,E,F,G)  
  0b1111110, // 0  
  0b0110000, // 1  
  0b1101101, // 2  
  0b1111001, // 3  
  0b0110011, // 4  
  0b1011011, // 5  
  0b1011111, // 6  
  0b1110000, // 7  
  0b1111111, // 8  
  0b1111011 // 9
```

```
};

// Current digit to display
int currentDigit = 0;

void setup() {
  // Set pins 2-8 as outputs for segments A-G
  for (int pin = 2; pin <= 8; pin++) {
    pinMode(pin, OUTPUT);
  }

  // Set pin 10 & 11 as input for the buttons
  pinMode(10, INPUT_PULLUP); // plus button
  pinMode(11, INPUT_PULLUP); // minus button

  // Initially display 0
  displayDigit(currentDigit);
}

void displayDigit(int digit) {
  // Validate digit is between 0-9
  if (digit < 0 || digit > 9) return;

  // Get the bit pattern for the digit
  byte pattern = digitPatterns[digit];

  // Display the digit by setting each segment
  for (int seg = 0; seg < 7; seg++) {
    // Segments are on pins 2-8
    // Check each bit in the pattern
    digitalWrite(seg + 2, (pattern & (1 << (6 - seg))) ? HIGH : LOW);
  }
}

void loop() {
  // Read the button state
  displayDigit(currentDigit);
  if (digitalRead(10) == LOW) { // Button pressed (LOW because of INPUT_PULLUP)
    // Increment digit
    Serial.print("Increase");
    currentDigit = (currentDigit + 1) % 10;

    // Display the new digit
  }
}
```

```
displayDigit(currentDigit);

// Wait until button is released to prevent multiple increments
while (digitalRead(10) == LOW) {
    delay(10); // Small delay to debounce
}
}
if (digitalRead(11) == LOW) { // Button pressed (LOW because of INPUT_PULLUP)
    // de-Increment digit
    Serial.print("Decrease");
    currentDigit = currentDigit - 1;
    if (currentDigit < 0){
        currentDigit = 9;
    }

    // Display the new digit
    displayDigit(currentDigit);

    // Wait until button is released to prevent multiple increments
    while (digitalRead(11) == LOW) {
        delay(10); // Small delay to debounce
    }
}
}
```

E-Portfolio video with updated code.

E-Portfolio Published link with video file. Upload the file to your Google Drive to upload on your Portfolio. YouTube Videos preferred

<https://sites.google.com/riversideunified.org/matthewjeide/notes/w9b-de-counting-up-and-down-arduino-variables-tutorial>

Conclusion

Answer in complete sentences each of the questions below.

1. How does step 3 compare to step 1 and 2. What changes did you make and why?
Step 3 is creating entirely new code whilst step 1 and step 2 were meant to be modifications or summaries of code. The original code given did not incorporate any button elements at all, step 3 did.