

Programming the Arduino



Instructor: Aimee Eddins

2011 March 6

5-7 PM

Today we'll be covering:

- .Syntax

- .Branches

- .Loops

- .Functions

- .Classes

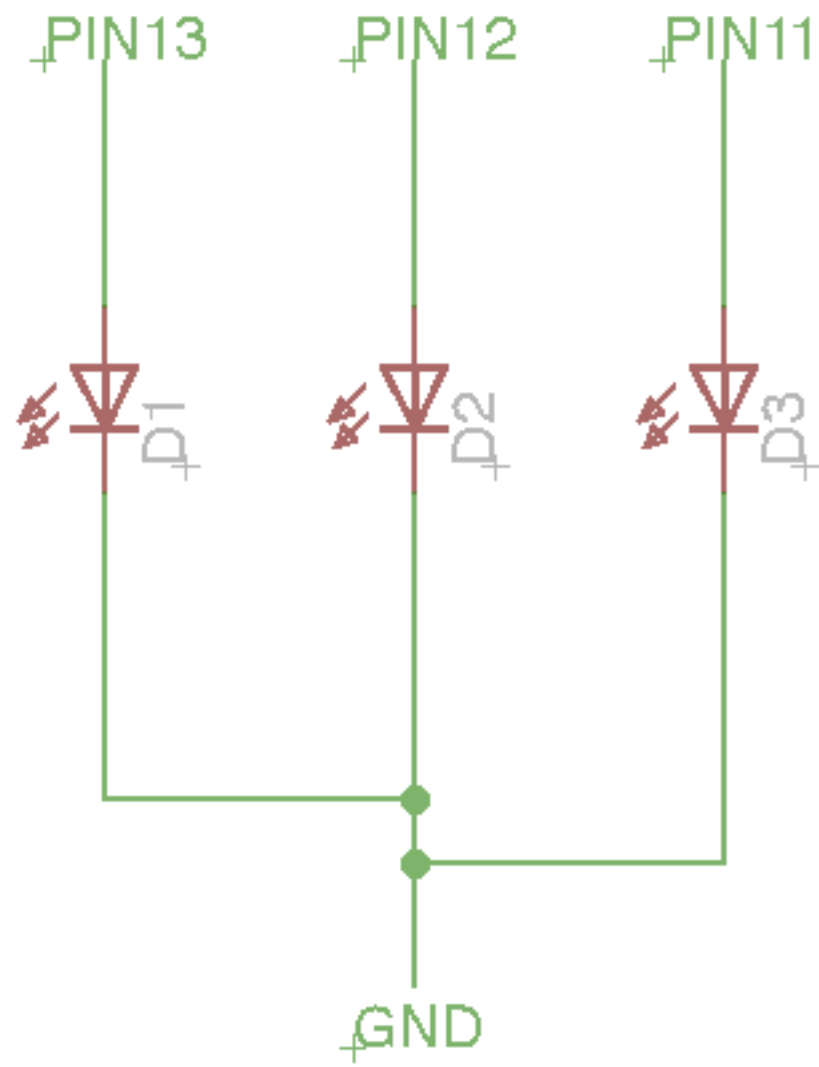
Structure and Syntax

Magic Words

Storing information

That's enough talking.
Let's make stuff

Controlling LEDs with “if”



Turn LED on from Serial Command Line

```
//defines what pin the LEDs are on
int LEDGreen = 11;
int LEDYellow = 12;
int LEDRed = 13;

void setup(){
  pinMode(LEDRed, OUTPUT);
  Serial.begin(9600);
}

void loop(){
  if (Serial.available){
    char value = Serial.read();
    if value == 'r'){
      digitalWrite(LEDRed, HIGH);
    }
  }
}
```

Put functionality in your functions

Toggling LEDs

Code: Part I/III

```
//defines what pin the LEDs are on
int LEDGreen = 11;
int LEDYellow = 12;
int LEDRed = 13;

//these lines define the variable where
//we keep track of what state the LEDs
//are in
int LEDState_R = 0;
int LEDState_Y = 0;
int LEDState_G = 0;

void setup(){
  //the pinMode commands define the state
  //of the digital I/O pins being used by
  //the 3 LEDs
  pinMode(LEDRed, OUTPUT);
  pinMode(LEDYellow, OUTPUT);
  pinMode(LEDGreen, OUTPUT);
  //sets up serial communication
  Serial.begin(9600);
}
```

Code: Part II/III

```
void loop(){
  //if there is serial information available
  //then do what's in my branch (i.e. if statement)
  if (Serial.available){
    //save the byte in the serial buffer to a
    //variable so I can make multiple comparisons
    //if need be
    char value = Serial.read();
    //if I have sent an 'r', turn on the Red LED
    if (value == 'r'){
      LEDState_R = toggleLED(LEDRed,LEDState_R);
    }
    //if I have sent a 'g', turn on the green LED
    if (value == 'g'){
      LEDState_G = toggleLED(LEDGreen,LEDState_G);
    }
    //if I have sent a 'y', turn on the yellow LED
    if (value == 'y'){
      LEDState_Y = toggleLED(LEDYellow,LEDState_Y);
    }
  }
}
```

Toggling LEDs

Code: Part III/III

```
//this new function makes it easier to turn LEDs
//on and off: every time I need to do either thing
//I call this function (toggleLED) and it will take
//the appropriate action
//it returns (gives me back) the value inside LEDState
//which is why it has an 'int' return type listed
//at the beginning of the function definition
int toggleLED(int LEDpin, int LEDState){
    //if the LED is off (it's state is 0), turn it on
    //otherwise, turn it on
    if (LEDState == 0){
        //turn on LED
        digitalWrite(LEDpin,HIGH);
        //make sure LEDState reflects the fact that
        //now the LED is on
        LEDState = 1;
    } else {
        //turn off LED
        digitalWrite(LEDpin,LOW);
        //make sure LEDState reflects the fact that
        //now the LED is off
        LEDState = 0;
    }
    //make sure I know the state of the LED in
    //the rest of my code
    return LEDState;
}
```