

Goal:

Understand how to write programs for the Mudduino

```
#include <stdio.h>

int main(void)
{
    int i = 0;

    i = i + 25;

    while(1)
    {
        printf("%d\n", i);
        i++;
    }

    return 0;
}
```

```
// This is template behind Arduino sketches
int main(void)
{
    setup();
    while(1) loop();
    return 0;
}

//you get to write:
void setup(void);
void loop(void)
```

It would execute this once

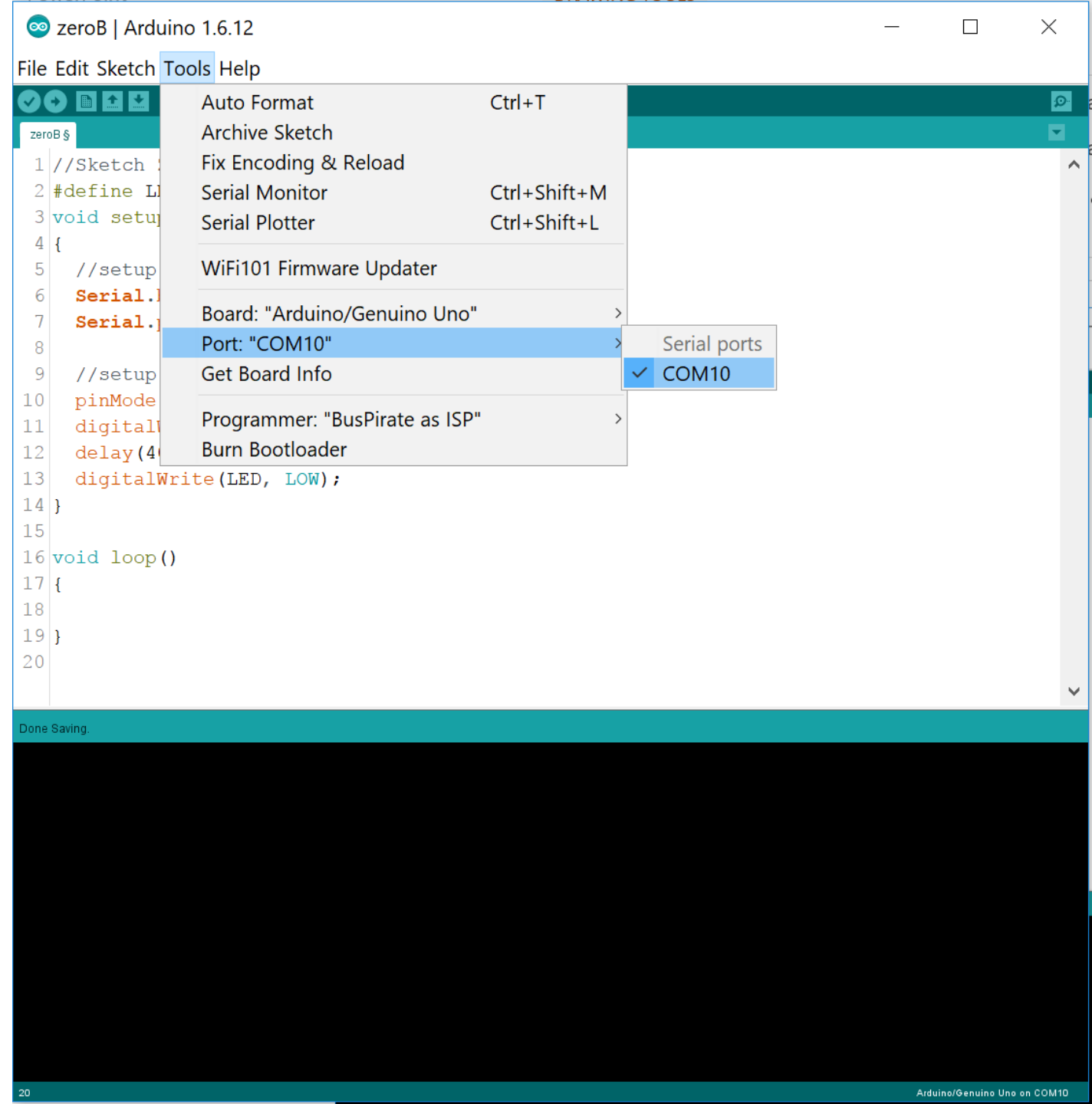
Then it would execute this forever

```
//Sketch Zero
#define LED 13
void setup()
{
  //setup the serial terminal
  Serial.begin(9600);
  Serial.println("I'm alive!");

  //setup LED pin, blink once
  pinMode(LED, OUTPUT);
  digitalWrite(LED, HIGH);
  delay(4000); // in ms
  digitalWrite(LED, LOW);
}

void loop()
{
}
}
```

1. Plug in your FTDI cable to the computer and board
2. Copy the code into a new sketch
3. Verify the correct Serial port is selected under tools

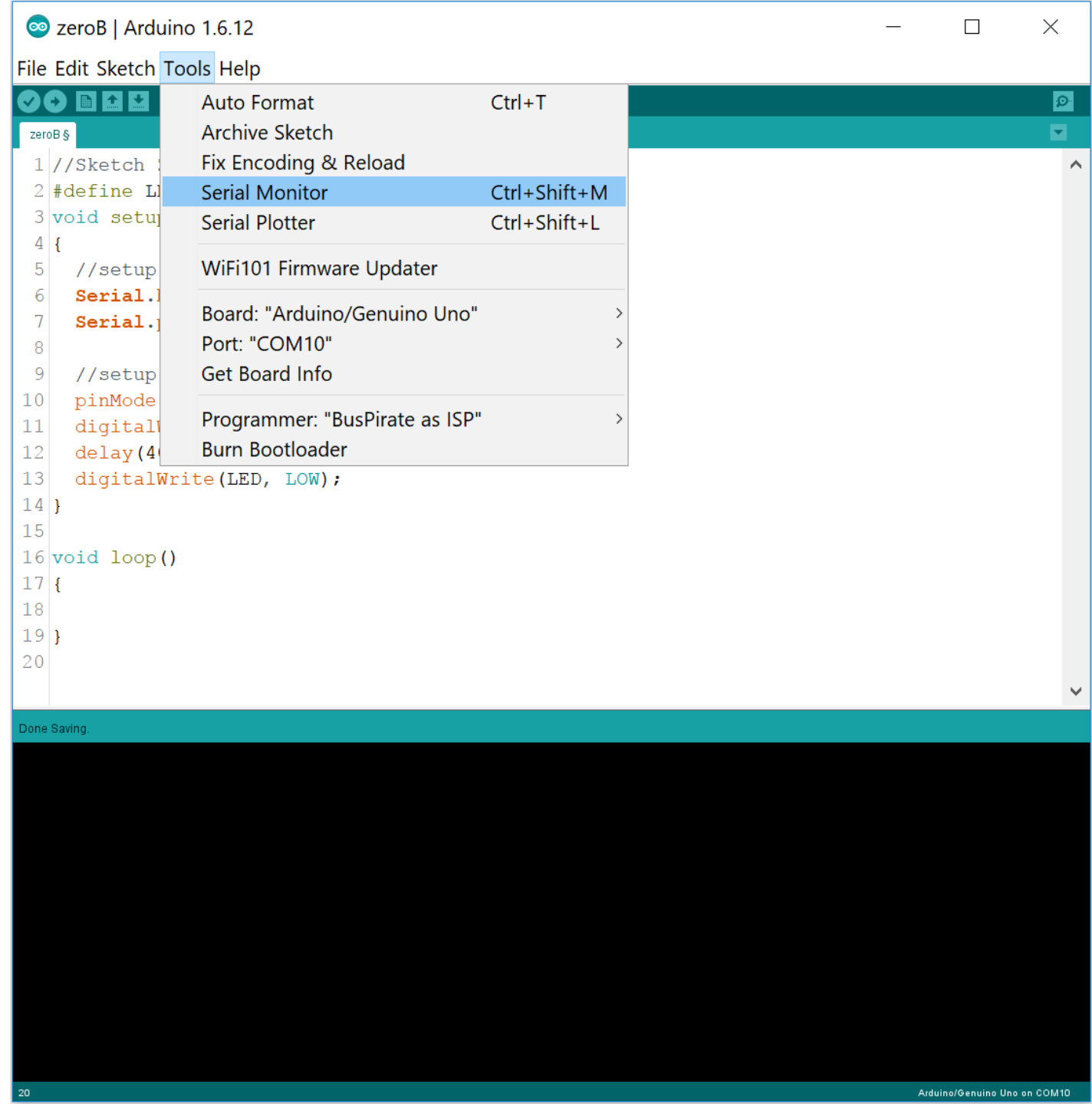


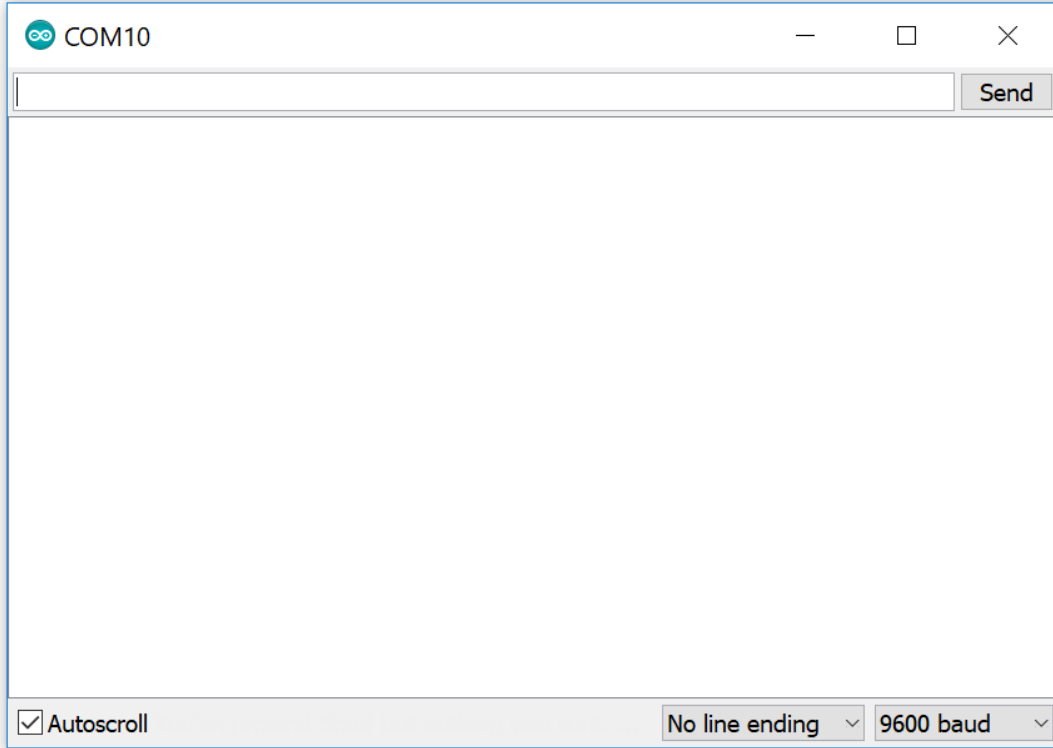
```
//Sketch Zero
#define LED 13
void setup()
{
  //setup the serial terminal
  Serial.begin(9600);
  Serial.println("I'm alive!");

  //setup LED pin, blink once
  pinMode(LED, OUTPUT);
  digitalWrite(LED, HIGH);
  delay(4000); // in ms
  digitalWrite(LED, LOW);
}

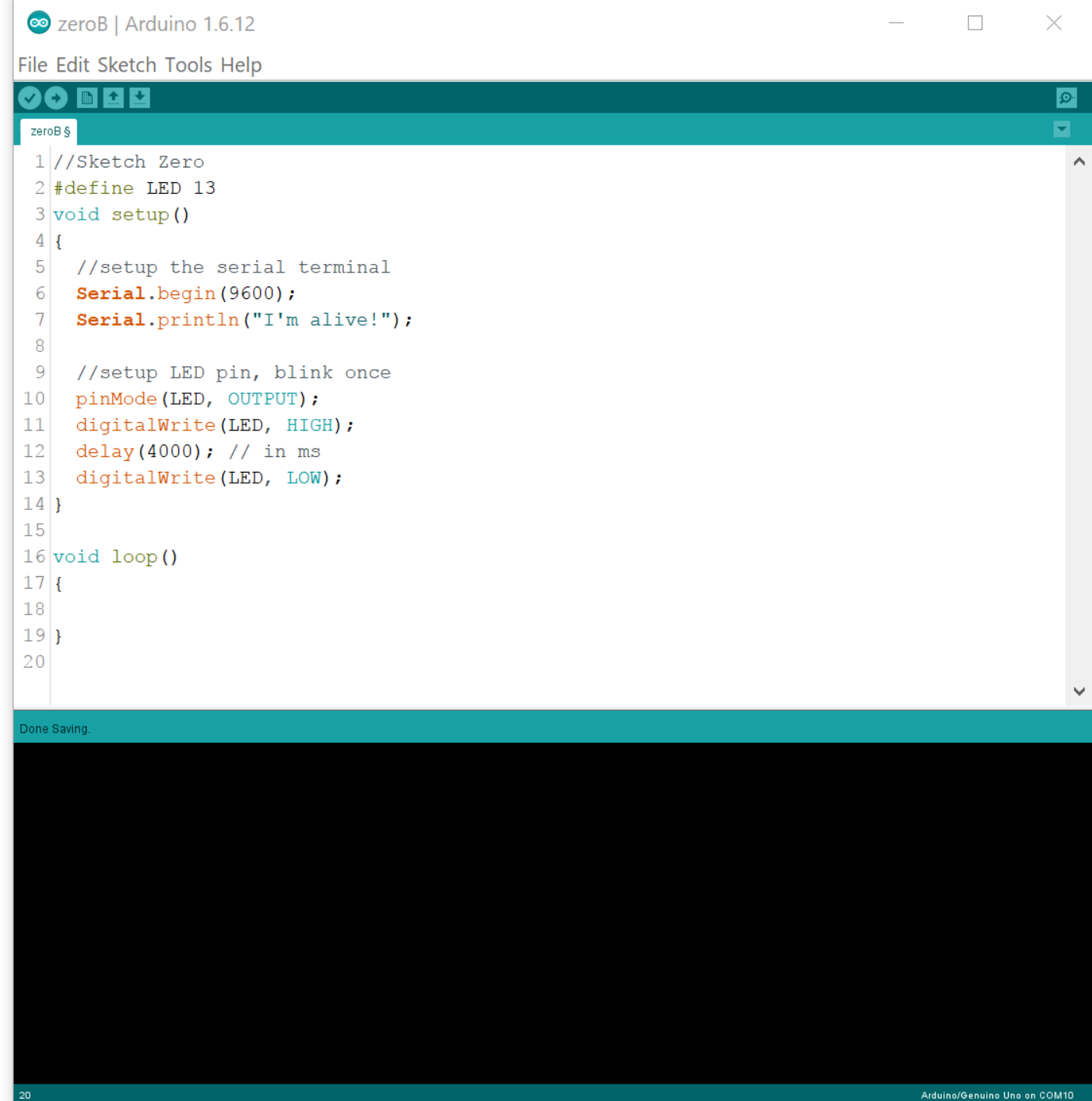
void loop()
{
}
```

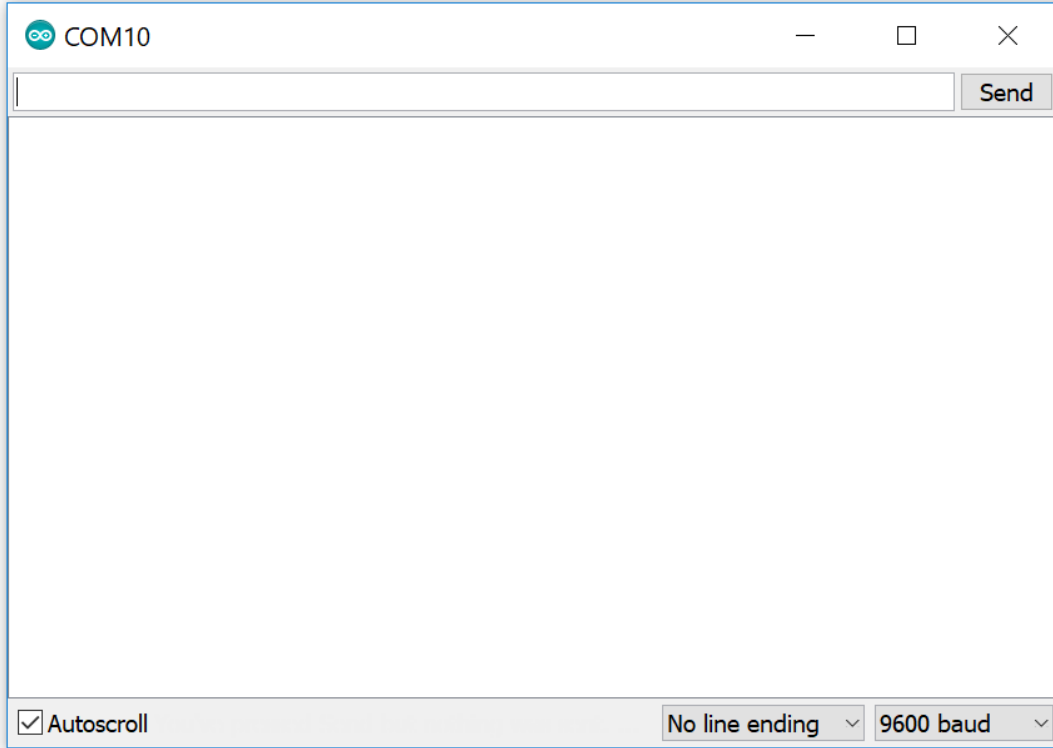
1. Plug in your FTDI cable to the computer and board
2. Copy the code into a new sketch
3. Verify the correct Serial port is selected under tools
4. Open the serial monitor



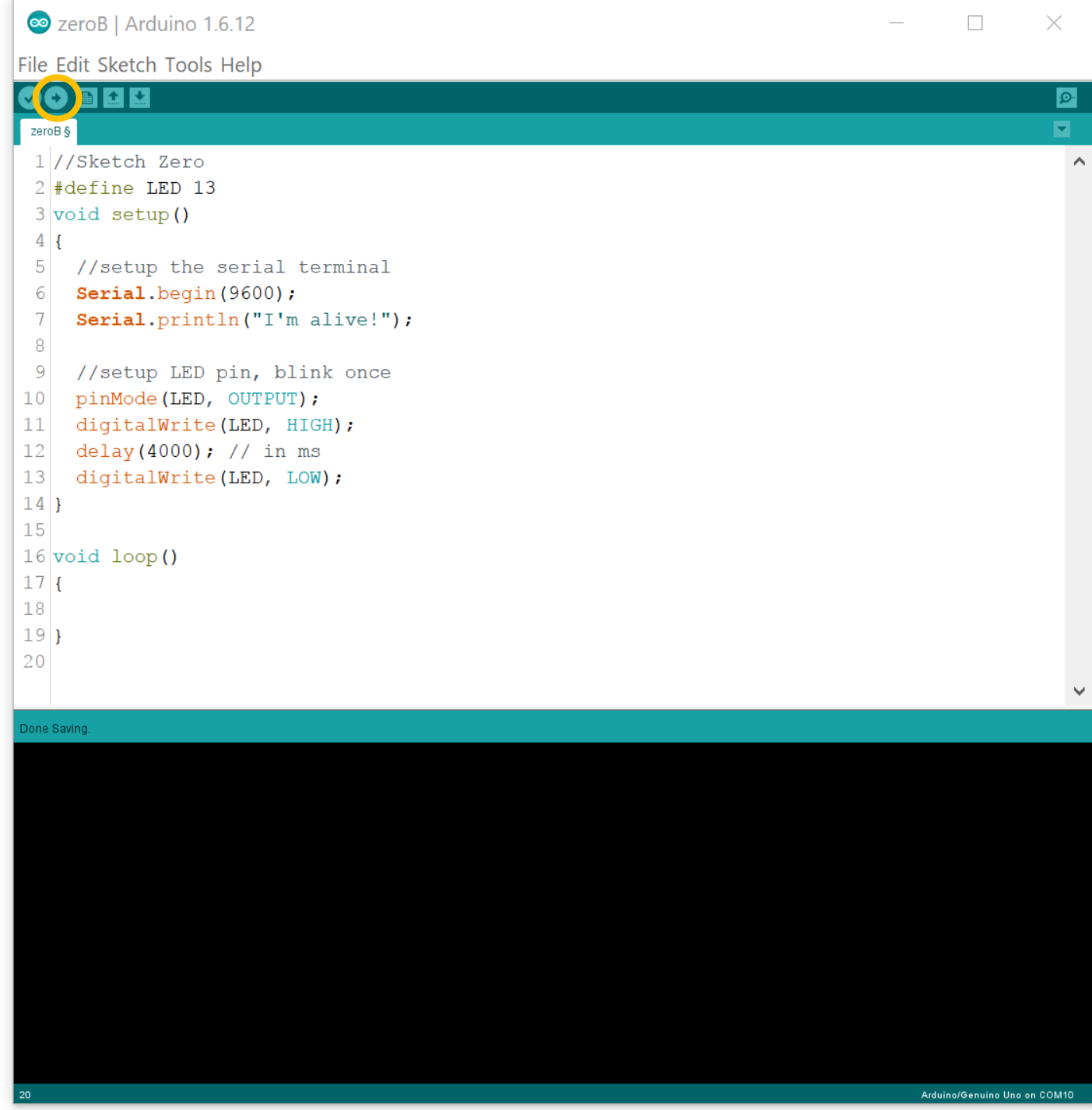


1. Plug in your FTDI cable to the computer and board
2. Copy the code into a new sketch
3. Verify the correct Serial port is selected under tools
4. Open the serial monitor





1. Plug in your FTDI cable to the computer and board
2. Copy the code into a new sketch
3. Verify the correct Serial port is selected under tools
4. Open the serial monitor
5. Click upload



```
//Sketch Zero
#define LED 13
void setup()
{
    //setup the serial terminal
    Serial.begin(9600);
    Serial.println("I'm alive!");

    //setup LED pin, blink once
    pinMode(LED, OUTPUT);
    digitalWrite(LED, HIGH);
    delay(4000); // in ms
    digitalWrite(LED, LOW);
}

void loop()
{
}
```

The green LED is on PIN 5

Modify this sketch to blink the Green LED after the Red LED. Have the Green LED on for 2 seconds

When both LEDs are done flashing
print: “I’m done” to the serial terminal

```
//Sketch Zero Flicker
#define LED 13
void setup()
{
  //setup the serial terminal
  Serial.begin(9600);
  Serial.println("I'm alive!");

  //setup LED pin, blink once
  pinMode(LED, OUTPUT);
}

void loop()
{
  digitalWrite(LED, HIGH);
  delay(random(10,50)*10); // in ms
  digitalWrite(LED, LOW);
  delay(random(10,50)*10); // in ms
}
```

Let's have some fun and make our LED seem to flicker. To do this we need to create random numbers.

Thankfully the Arduino library has a function for that: random(min, max)

random(min, max) creates a pseudo random integer from min to max-1

So our code will create on/off times from 100 ms to 499 ms

Flash the code and see


```
//Sketch Zero Read
#define LED 13
void setup()
{
  //setup the serial terminal, and outputs
  Serial.begin(9600);
  Serial.println("I'm alive!");
  pinMode(LED, OUTPUT);
}

void loop()
{
  unsigned char keystroke;
  //check if the serial port is available
  if (Serial.available())
  {
    keystroke = Serial.read(); //get a character
    Serial.println(keystroke+1); //print next higher ASCII value
    //blink the Red LED
    digitalWrite(LED, HIGH);
    delay(500); // in ms
    digitalWrite(LED, LOW);
    delay(500); // in ms
  }
}
```

The serial connection can be used not only for printing information it can be used for getting input

Flash your Mudduino board with this sketch and explore what it does.

```
//First sketch
#define LED 13
void setup()
{
    //setup the serial terminal
    Serial.begin(9600);
    Serial.println("I'm alive!");

    //setup LED pin
    pinMode(LED, OUTPUT);
}

void loop()
{
    unsigned char command;
    command = Serial.read();

    if(command == 'o')
    {
        Serial.println("LED On!");
        digitalWrite(LED, HIGH);
    }
    else if(command == 'f')
    {
        Serial.println("LED Off!");
        digitalWrite(LED, LOW);
    }
    else if(command == 'b')
    {
        Serial.println("Blink!");
        digitalWrite(LED, HIGH);
        delay(500);
        digitalWrite(LED, LOW);
        delay(500);
    }
}
```

What does this sketch do?

Read the code first, think about it, then flash your board and type o, f, b into the serial monitor

Write a sketch that once you enter 'g' for go it turns the LED on and off for the amount of time in an array. Have the program terminate when it reaches a negative value (as time is always positive).

Example:

```
short timeArray[] = {500, 400, 500, 400, 100, 200, 100, 200, -1};
```

This would have the LED on for 500 ms, off for 400 ms, on for 500 ms, off for 400 ms, on for 100 ms, off 200 for ms...

```

#define LED 13

short timeArray[] = {500, 400, 500, 400, 100, 200,
100, 200, -1};

void setup()
{
    //setup the serial terminal
    Serial.begin(9600);
    Serial.println("Ready...");

    //setup LED pin
    pinMode(LED, OUTPUT);
}

void loop()
{
    unsigned char command;
    if (Serial.available() > 0)
    {
        command = Serial.read();
        int i = 0;
        int highLow = 1;

        if(command == 'g')
        {
            while(timeArray[i] >= 0)
            {
                digitalWrite(LED, highLow);
                delay(timeArray[i]);
                highLow = !highLow;
                i++;
            }
        }
    }
}

```

Here is one solution to the problem

First we setup the serial port, print that the program is running, then we setup the LED pin as an output

In the main loop we are more careful with the serial port and make sure there is something to read before trying to read it, then if it is 'g' we increment though the array until we reach a negative value flipping the value of the LED by negating highLow.

Write a sketch that allows you to blink both the red and the green LED for different amounts of time simultaneously.

The green LED is on pin 5.

Example:

```
short timeArrayR[] = {500, 400, 500, 400, 100, 200, 100, 200, -1};  
short timeArrayG[] = {100, 200, 100, 300, 50, 50, 50, 50, 50, 50, -1};
```

```

#define RedLED 13
#define GreenLED 5
#define True 1
#define False 0

short timeArrayR[] = {500, 400, 500, 400, 100, 200,
100, 200, -1};
short timeArrayG[] = {100, 200, 100, 300, 50, 50,
50, 50, 50, 50, -1};

```

```

void setup()
{
    //setup the serial terminal
    Serial.begin(9600);
    Serial.println("Ready...");

    //setup LED pins
    pinMode(RedLED, OUTPUT);
    pinMode(GreenLED, OUTPUT);
}

```

```

void loop()
{
    unsigned char command;
    if (Serial.available() > 0)
    {
        command = Serial.read();
        int i = 0;
        int j = 0;
        int notDone = True;
        int notDoneRed = True;
        int notDoneGreen = True;
        long currentTime = 0;

```

```

        if(command == 'g')
        {
            while(notDone)
            {

                notDoneRed = setLED(currentTime, timeArrayR, RedLED);
                notDoneGreen = setLED(currentTime, timeArrayG, GreenLED);
                notDone = notDoneGreen || notDoneRed;
                delay(1);
                currentTime++;
            }
        }
    }
}

```

```

int setLED(long currentTime, int edges[], int pin)
{
    int i = 0;
    long totalTime = 0;
    while(edges[i] > 0)
    {
        totalTime += edges[i];
        if(currentTime < totalTime)
        {
            if(i%2 == 0)digitalWrite(pin, HIGH);
            else digitalWrite(pin, LOW);
            return True;
        }
        i++;
    }

    return false;
}

```