



Robotics Board User Guide

Introduction

The Crazy Circuits Robotics Board is a fun way to create robotics and programming projects. The Robotics board features a wide range of handy features for creating projects.

This User Guide will show you how to install the needed Robotics Board Software, upload code, and how to connect it to components. At the end of this guide we'll walk you through a couple of simple projects using the Robotics Board. For more complicated projects please visit our Project Galley or Forums on CrazyCircuits.com.

GitHub Projects Repo

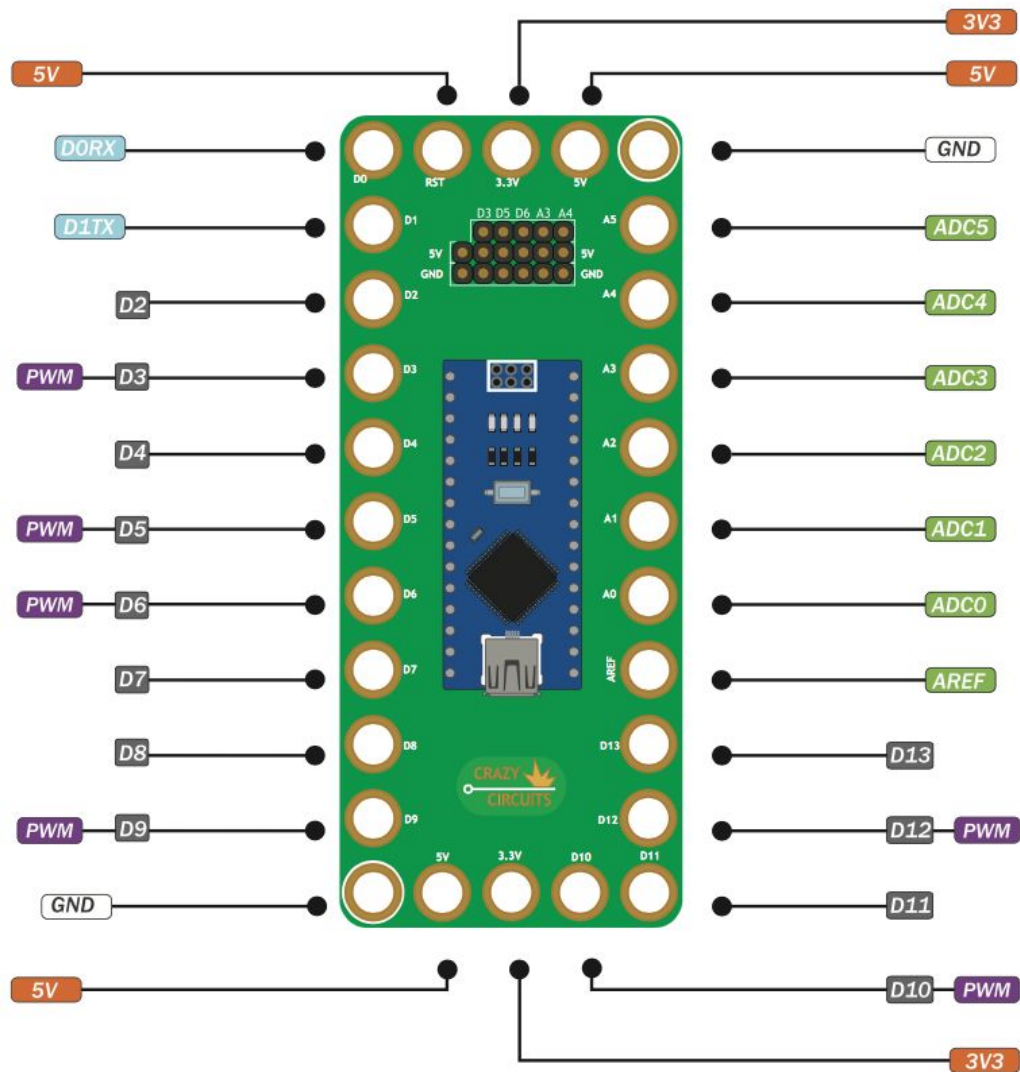
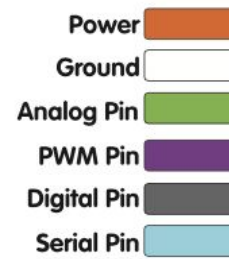
<https://github.com/BrownDogGadgets/CrazyCircuits/>

Forums

TBA

Technical Details

Robotics Board INTEGRATED CIRCUIT



The Crazy Circuits Robotics Board is based off the open source Arduino Nano development board. Advanced users can find all sorts of resources at <https://www.arduino.cc> and their forums. In general, these boards act like a standard Arduino and can run any standard Arduino code.

In addition to connecting components with Conductive Tape or Thread we've added six sets of Row Headers along the back of the board. These allow for a quick and easy way to plug in Servos, Sensor Modules, and Power.

Please note that unlike our Touch Board, our Robotics Board does not have built in Keyboard Emulation or Touch capabilities and will not work with all the projects we have listed on our website. However, it is recommended for most projects that use Servos and third party sensors.

Blank, unsoldered, Robotics Board PCBs are available for sale on CrazyCircuits.com.

Software

In order to upload projects and code to the Robotics Board you'll need to install some software. The software will run on any Mac, PC, or Linux computer with built in USB ports, which is pretty much most all computers these days.

First you'll need to download the free Arduino IDE software from the Arduino Website. This is the computer program that will let you upload code to your Touch Board. Download version 1.8.4 for with Mac or PC. (Version 1.8.5 does not work with our board yet.)

<https://www.arduino.cc/en/Main/OldSoftwareReleases#previous>

Drivers for Mac and PC may be needed from this website.

<https://kig.re/2014/12/31/how-to-use-arduino-nano-mini-pro-with-CH340G-on-mac-osx-yosemite.html>

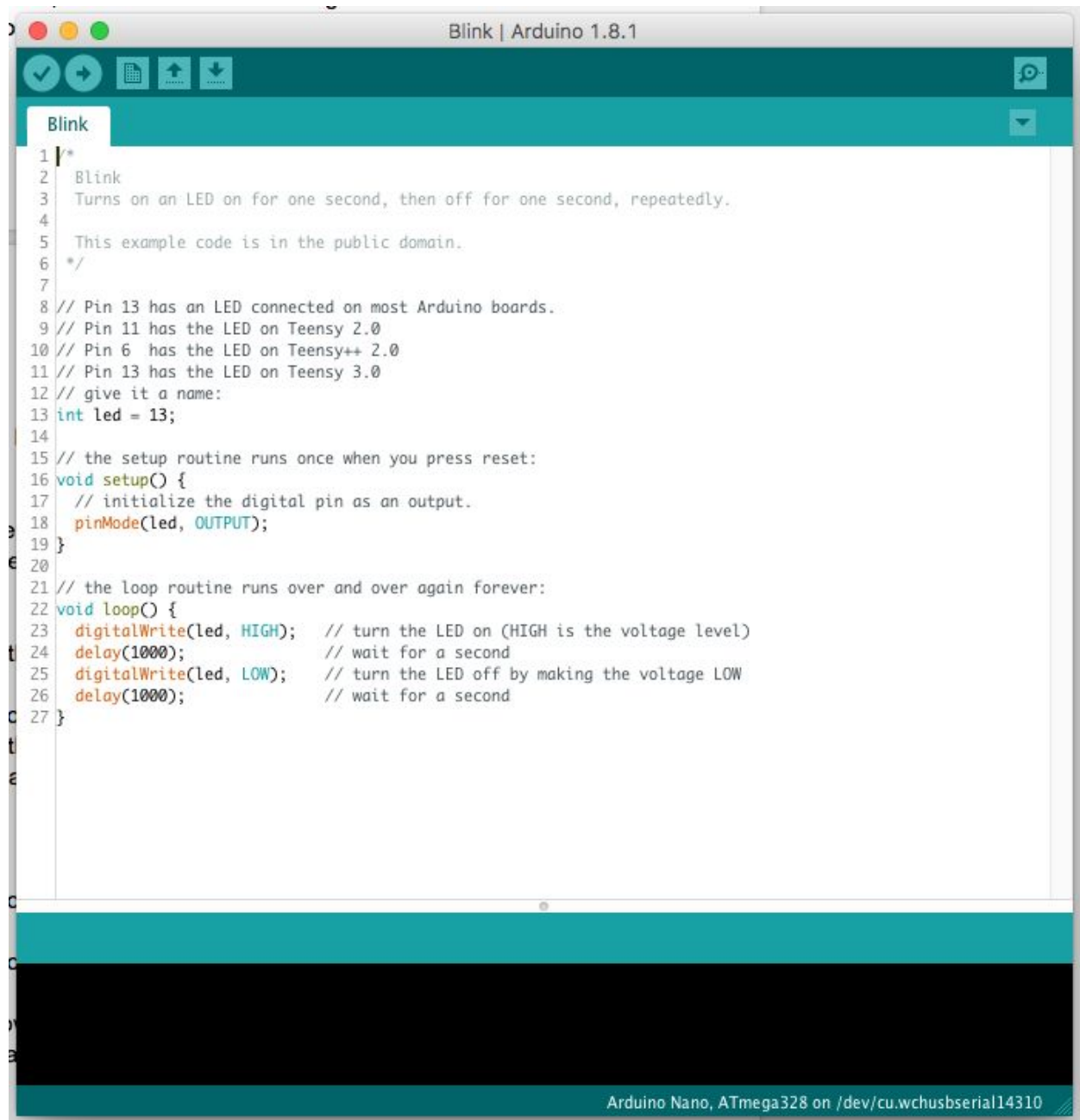
Arduino IDE

Millions of people around the world use the Arduino IDE software to interact with their Arduino Compatible development boards. Our Robotics Board is no different.

Before you start uploading code and projects to your Robotics Board we want to walk through a few basic sections of the Arduino IDE, as it can be a bit confusing for new users. These are tips and directions specific to our Touch Board, which means we won't be explaining everything.

More or less you're going to mostly be dealing with Sketch writing window and the Tools drop down Menu.

The Sketch writing window is where you type in or paste in your code. If you're just following along with our existing Project Guides you'll mainly be doing a Copy and Paste into this window and hitting Upload.



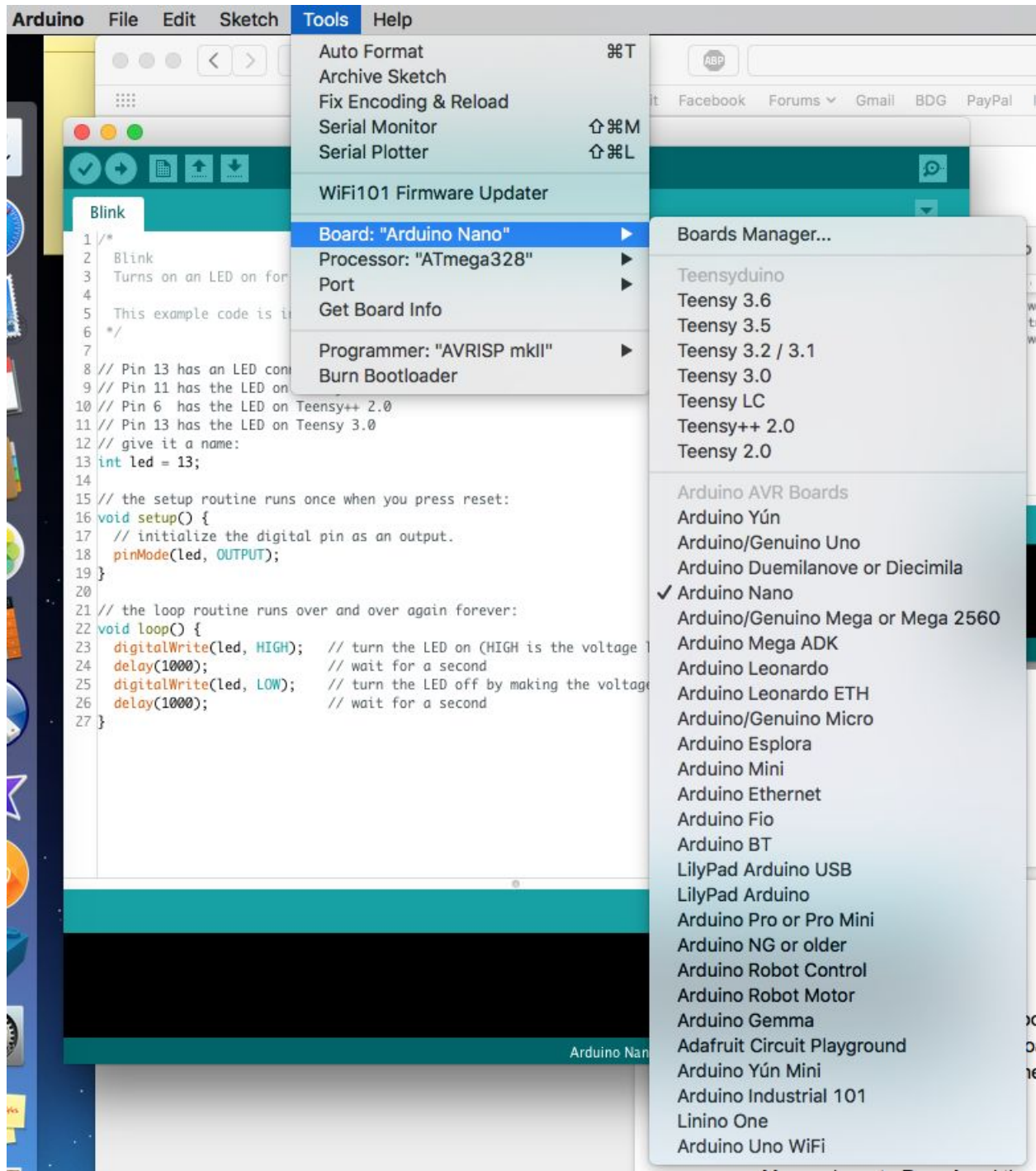
The screenshot shows the Arduino IDE interface with the 'Blink' sketch loaded. The title bar reads 'Blink | Arduino 1.8.1'. The sketch editor contains the following code:

```
1 /*
2  * Blink
3  * Turns on an LED on for one second, then off for one second, repeatedly.
4  *
5  * This example code is in the public domain.
6  */
7
8 // Pin 13 has an LED connected on most Arduino boards.
9 // Pin 11 has the LED on Teensy 2.0
10 // Pin 6 has the LED on Teensy++ 2.0
11 // Pin 13 has the LED on Teensy 3.0
12 // give it a name:
13 int led = 13;
14
15 // the setup routine runs once when you press reset:
16 void setup() {
17   // initialize the digital pin as an output.
18   pinMode(led, OUTPUT);
19 }
20
21 // the loop routine runs over and over again forever:
22 void loop() {
23   digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
24   delay(1000);             // wait for a second
25   digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
26   delay(1000);             // wait for a second
27 }
```

The status bar at the bottom indicates 'Arduino Nano, ATmega328 on /dev/cu.wchusbserial14310'.

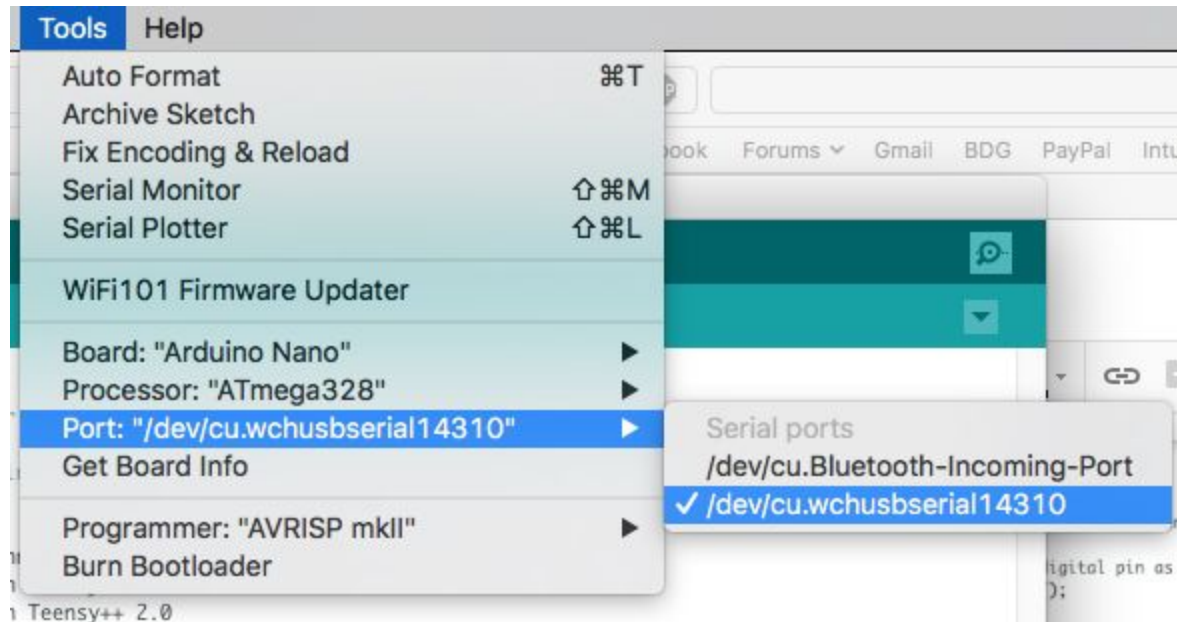
Before you can upload your code you'll need to select a few settings. (If you're only ever going to be using our Touch Board these settings will rarely need to be changed in the future) Click on the drop down **Tool** menu at the top of your screen.

Mouse down to **Board** and then choose Arduino Nano.



Next, plug in your Robotics Board to your computer.

Again, go to the Tools Menu and down to Port. Choose the USB Serial Option. Depending on if you're using a Mac or PC, you may get different options.



If you use different types of Arduino Boards, such as our Touch Board or a standard Arduino Uno, you'll need to change Board Types in the **Tools** Menu.

Uploading Code and Projects

Lets walk through a simple code upload to make sure that everything is installed and working properly.

We're going to start by uploading the Arduino Blink Sketch to our board. This uses the built in LED (Pin 13) on the Robotics Board Board and makes its blink. If you want to you can hook up a Crazy Circuits LED to Pin 13 on the board via Conductive Tape.

- 1) Plug in your Robotics Board
- 2) Open the Arduino Software
- 3) Double check to make sure that you've selected the Arduino Nano from the Tools menu.
- 4) Make sure you've selected the Arduino Nano in Tools -> Port.
- 5) Copy and Paste the below code into the Arduino Sketch Window. (You can first delete everything else if you'd like.)
- 6) Otherwise, click the Upload button.

7) Check to see if your LED is blinking.

If you want to play around with the blink speed, change the delay(1000) code to a higher or lower number. A number of 1000 equals 1 second of time. Once you've changed the number click the Upload button to send the new code to the Touch Board.

Example Blink Code (Same as built into the Arduino Software)

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// Pin 11 has the LED on Teensy 2.0
// Pin 6 has the LED on Teensy++ 2.0
// Pin 13 has the LED on Teensy 3.0
// give it a name:
int led = 13;

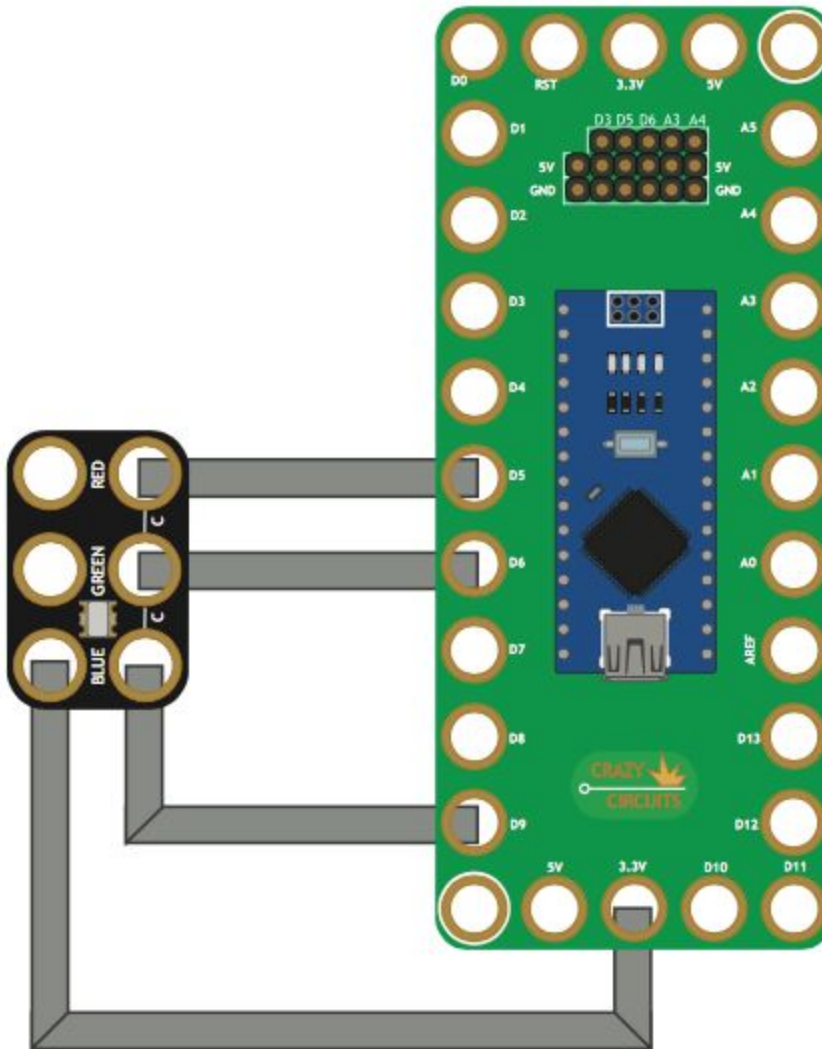
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

RGB LED

You can control the color of an RGB LED using some simple code.

Below is a simple diagram and code. Try building this and uploading the project.




```
int redPin= 5;
int greenPin = 6;
int bluePin = 9;

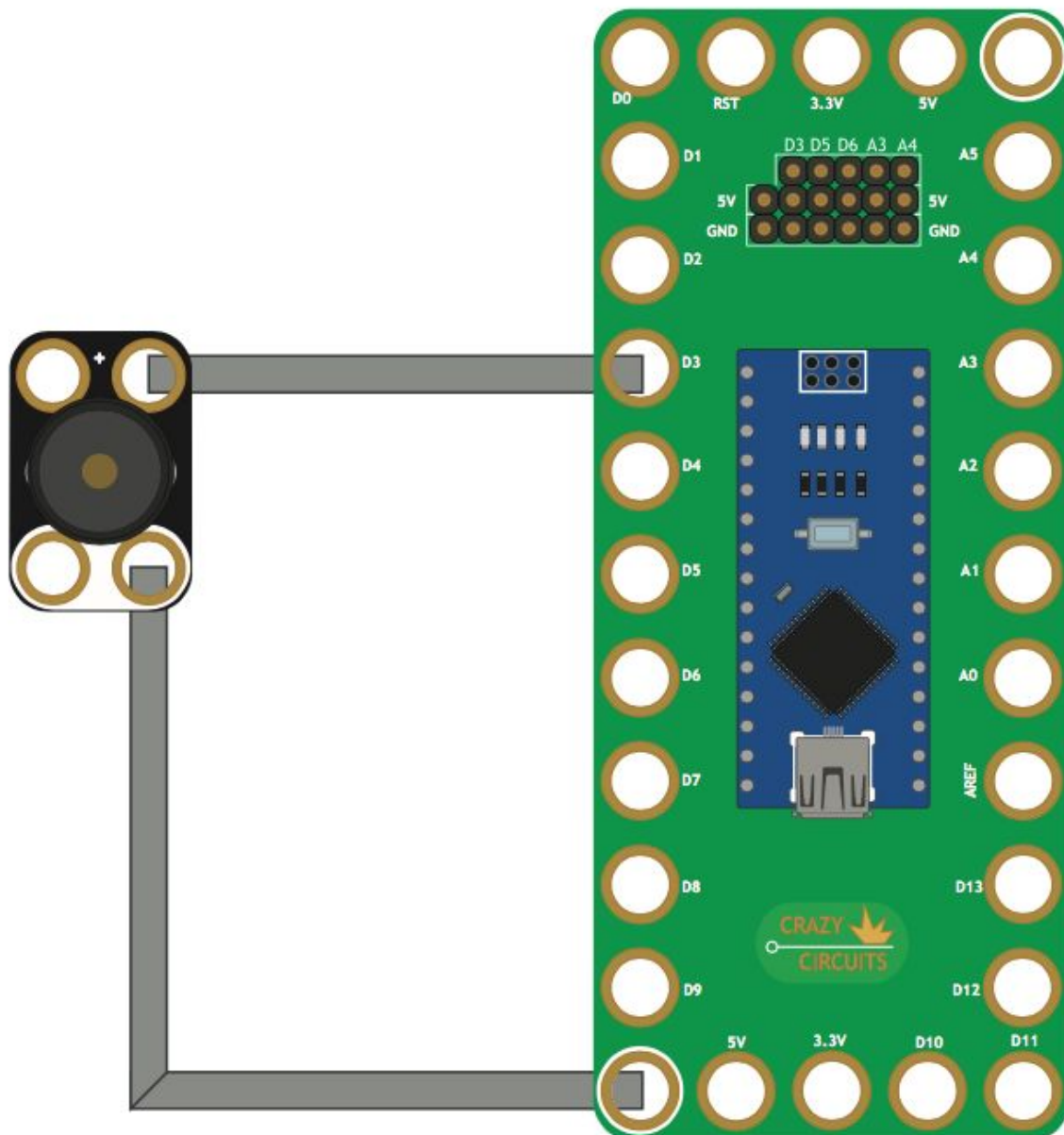
void setup() {
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop() {
  setColor(255, 0, 0); // Red Color
  delay(1000);
  setColor(0, 255, 0); // Green Color
  delay(1000);
  setColor(0, 0, 255); // Blue Color
  delay(1000);
  setColor(255, 255, 255); // White Color
  delay(1000);
  setColor(170, 0, 255); // Purple Color
  delay(1000);
}

void setColor(int redValue, int greenValue, int blueValue) {
  analogWrite(redPin, redValue);
  analogWrite(greenPin, greenValue);
  analogWrite(bluePin, blueValue);
}
```

Simple Tone Generation

Use a Piezo Speaker to generate some sound.



```
const int buzzer = 3; //buzzer to arduino pin 3

void setup(){
  pinMode(buzzer, OUTPUT); // Set buzzer - pin 3 as an output
}

void loop(){
  tone(buzzer, 1000); // Send 1KHz sound signal...
  delay(1000);      // ...for 1 sec
  noTone(buzzer);   // Stop sound...
  delay(1000);      // ...for 1sec
}
```