CRAZY CIRCUITS

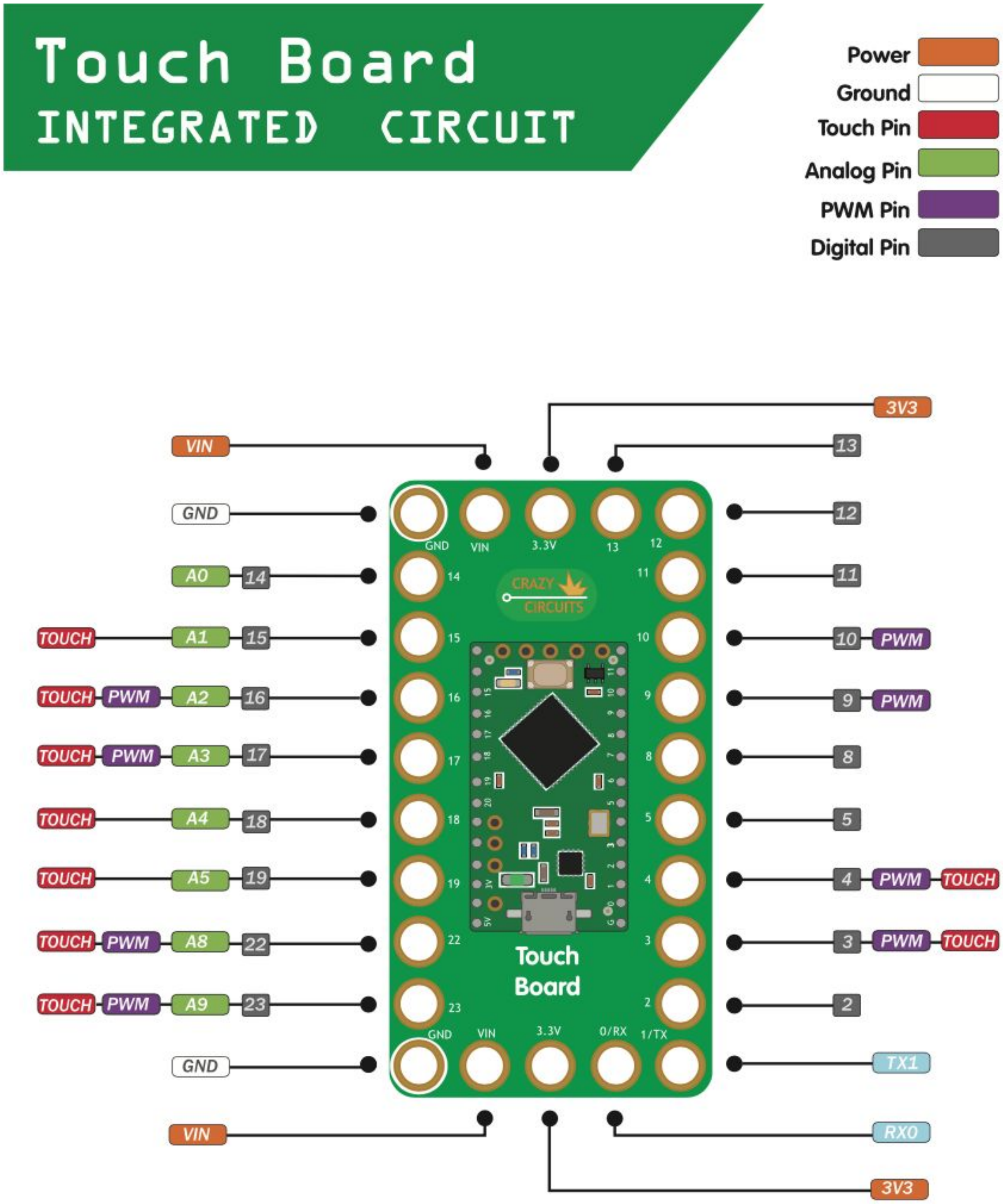
Touch Board User Guide

Introduction

The Crazy Circuits Touch Board is a fun way to create interactive projects. The Touch Board has 11 built in Touch Points for use with projects and also features built in KeyBoard emulation, which allows it to interface with any computer.

This User Guide will show you how to install the needed Touch 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 Touch Board.

Technical Details



The Crazy Circuits Touch Board is based off the Teensy LC development board. Advanced users can find all sorts of resources at pjrc.com and their forums. In general, these boards act like a standard Arduino and can run any standard Arduino code.

Two features we really enjoy about the Teensy LC is its built in Capacitive Touch capability and KeyBoard Emulation. These features are heavily relied on for many of our Projects. For a full list of board features and abilities, click the link below.

https://www.pjrc.com/teensy/teensyLC.html

The Touch Board PCB also accepts the Teensy 3.2 development board if more advanced features are needed.

Software

In order to upload projects and code to the Touch 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

Next, you'll need to install the Teensyduino software add-on. This installs the necessary drivers into the Arduino software to run the Teensy.

https://www.pjrc.com/teensy/td_download.html

Follow the install directions for your type of computer

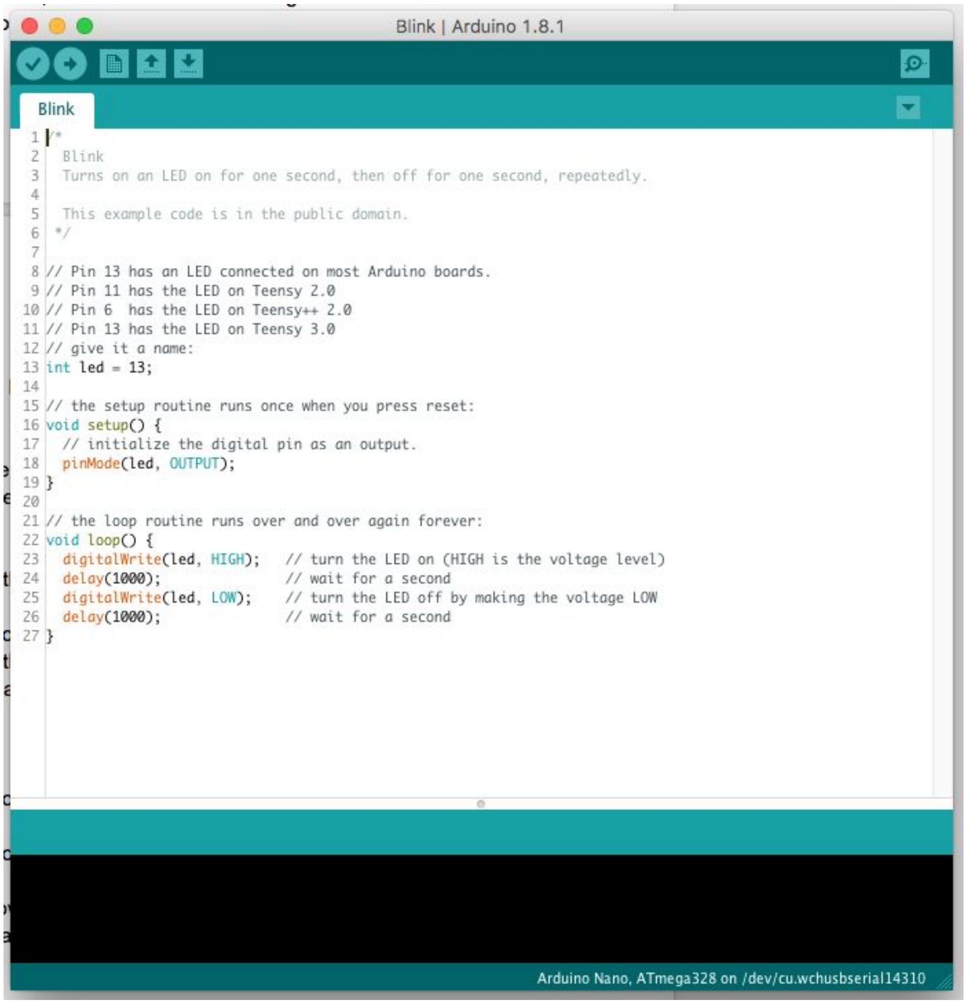
Arduino IDE

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

Before you start uploading code and projects to your Touch 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 allow with our existing Project Guides you'll mainly be doing a Copy and Paste into this window and hitting Upload.

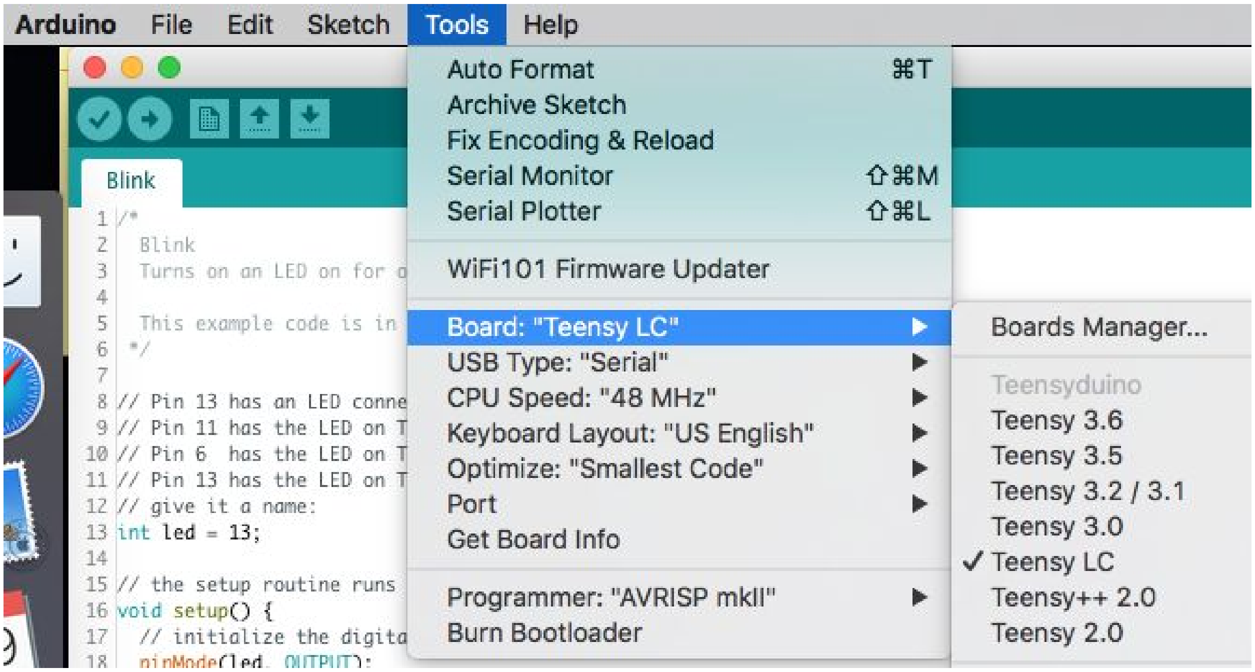


```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  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
}
```

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.

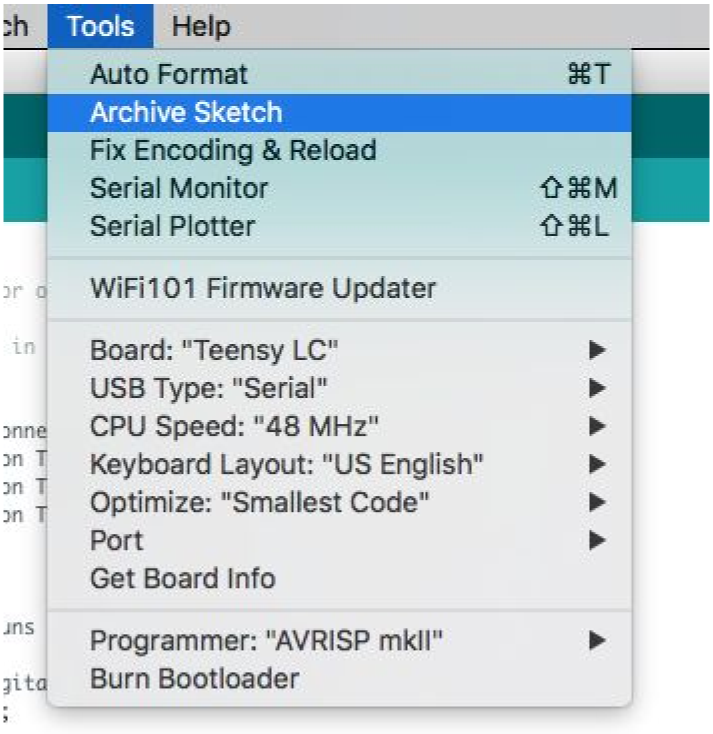


```
/*
  Blink
  Turns on an LED on for o

  This example code is in
 */

// Pin 13 has an LED conne
// Pin 11 has the LED on T
// Pin 6  has the LED on T
// Pin 13 has the LED on T
// give it a name:
int led = 13;

// the setup routine runs
void setup() {
  // initialize the digita
  pinMode(led, OUTPUT);
```

Board: "Teensy LC"
USB Type: "Serial"
CPU Speed: "48 MHz"
Keyboard Layout: "US English"
Optimize: "Smallest Code"
Programmer: "AVRISP mkII"

Mouse down to **Board** and then choose Teensy LC. Then, plug in your Touch Board to your computer.

In general your settings should always look like this:



Board: "Teensy LC"
USB Type: "Serial"
CPU Speed: "48 MHz"
Keyboard Layout: "US English"
Optimize: "Smallest Code"
Programmer: "AVRISP mkII"

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

Note: For nearly all Arduino variations a user usually has to select which "port" their board is on. With the Teensy (which is used on the Touch Board) you don't. This is because the Arduino software throws the compiled code over to the Teensy loader program. Thats why it pops up whenever you're programing.

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 Touch 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. Check the diagram below to see how that would look.

1) Plug in your Touch Board
2) Open the Arduino Software
3) Double check to make sure that you’ve selected the Teensy LC from the Tools menu.
4) Copy and Paste the below code into the Arduino Sketch Window. (You can first delete everything else if you’d like.)
5) If this is the first time uploading a project to your Touch Board, you’ll need to click the Verify Checkmark, and then press the little button on the Touch Board.
6) Otherwise, click the Upload button.
7) Check to see if your LED is blinking. (Next to the button on the middle of your Board.)

If you want to play around with the blink speed, chance 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.

```
/*
  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
}
```

Using the Touch Features

The Touch Board features 11 Capacitive Touch Points which allow you to turn anything conductive into a button. Typically we use Conductive Tape to create a path to a conductive item (such as a ball of foil or spot of conductive paint) but Alligator Clips work as well.

Below is an example demo of how to use the Capacitive Touch features to turn on one of two LEDs. One Touch Point turns on LED A, and the second Touch Point turns on LED B. Only one LED is on at a time.

Copy the code found at this link:

https://github.com/BrownDogGadgets/CrazyCircuits/blob/master/Projects/Status%20Light/Status_Light.ino

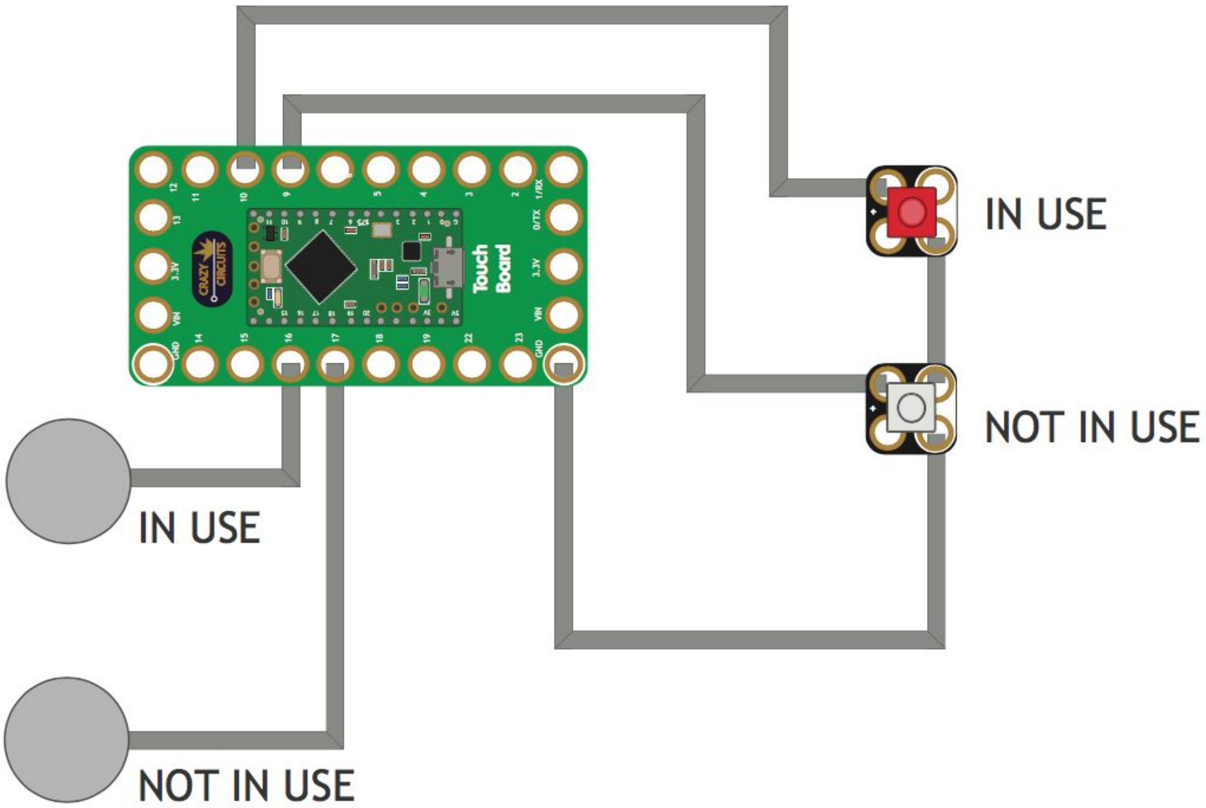
Paste it into the Arduino Software and upload it to your Touch Board.

Important: Touch Board Calibration

Typically all our Touch based code has a built in “Calibration Mode.” When the code starts up the Touch Board calibrates to whatever the activated Touch Points are hooked up to. In short, don’t Touch your Touch Points for 3-5 seconds when you initially turn on/ upload your code.

Once the Touch Board is calibrated it’ll stay Calibrated until it’s unplugged from power or you press the Reset Button on the board.

Use Conductive Tape to create this circuit.



As a KeyBoard Input

The Touch Board can emulate a standard USB Keyboard. This means you can make your own input devices.

Our favorite example of this is creating our own old school NES Controller using a Touch Board, some LEGOs, and six pushbuttons.

Download the code at this link and upload it to your Touch Board:

https://github.com/BrownDogGadgets/CrazyCircuits/blob/master/Projects/NES%20Controller/Teensy_NES_Controller.ino

By default the code emulates the four arrow keys as well as keys A and B. If you want to change those to different keyboard commands you just have to change a couple of letters in the code.

Look at line number 111 (use the above link again). Right now it says:

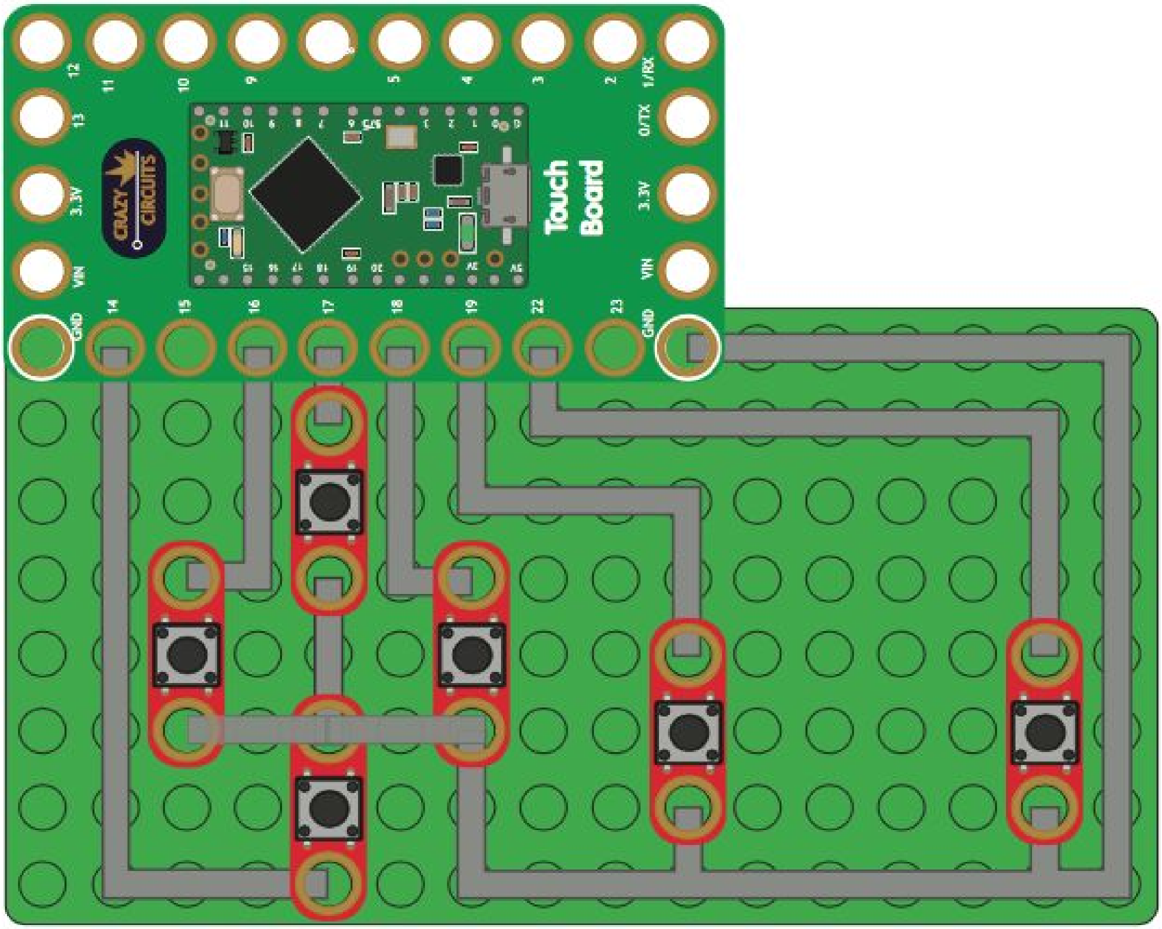
```
Keyboard.set_key6(KEY_A);
```

Change that to:

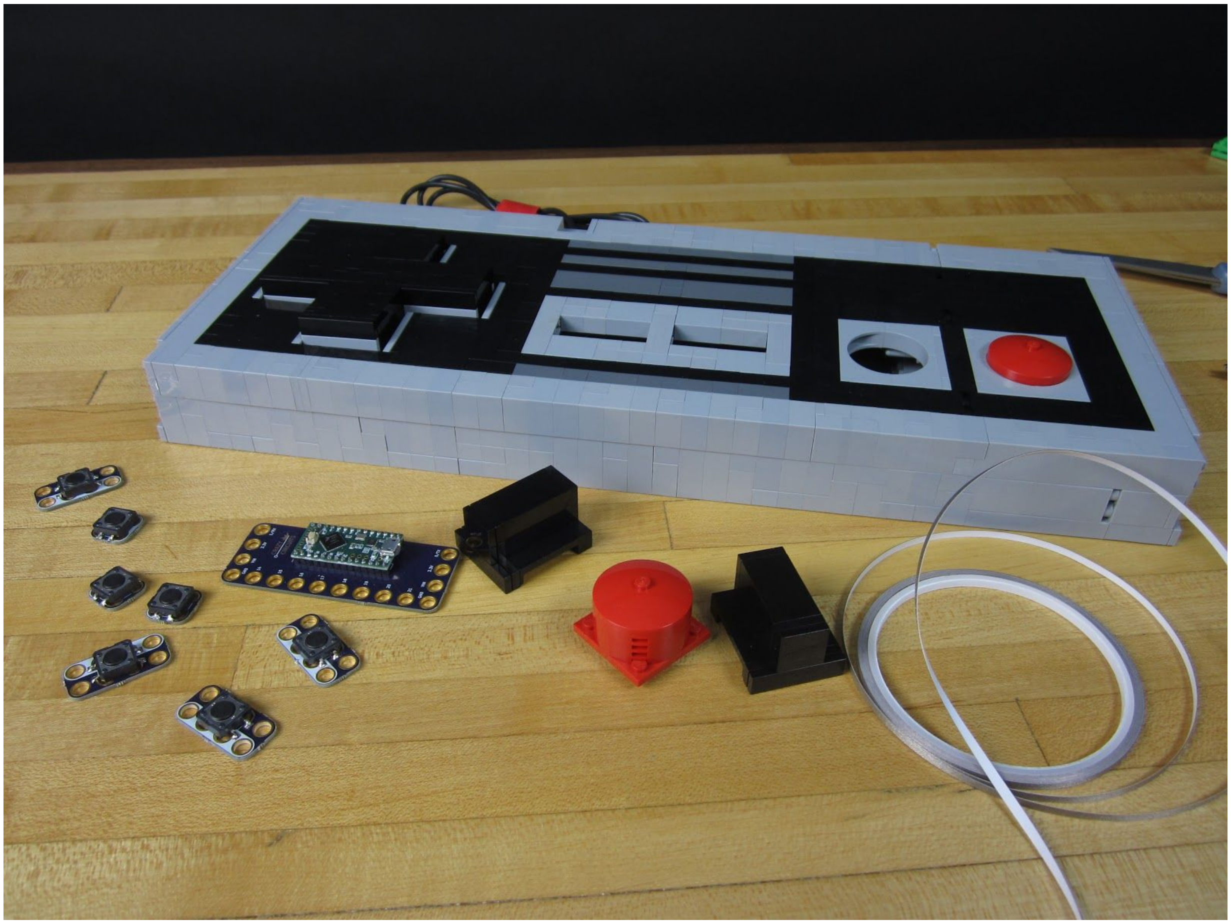
```
Keyboard.set_key6(KEY_M);
```

Now when you press the Pushbutton connected to Pin 20 your computer will type the letter “M”.

Follow the diagram below to create a hand sized NES controller!



We used a similar design to create a GIANT NES Controller made entirely out of LEGOs.



What Crazy things will you make?