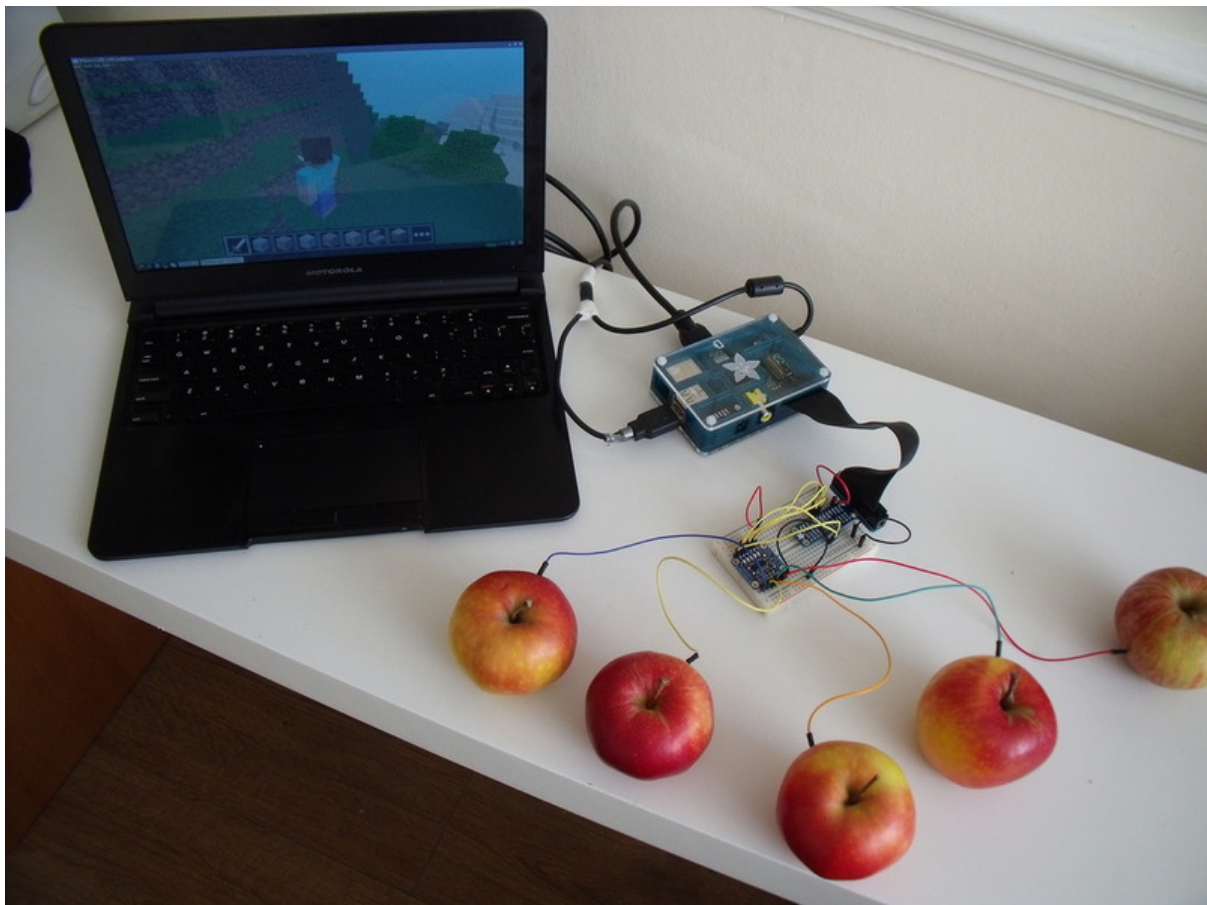




Capacitive Touch Sensors on the Raspberry Pi

Created by Craig Richardson



<https://learn.adafruit.com/capacitive-touch-sensors-on-the-raspberry-pi>

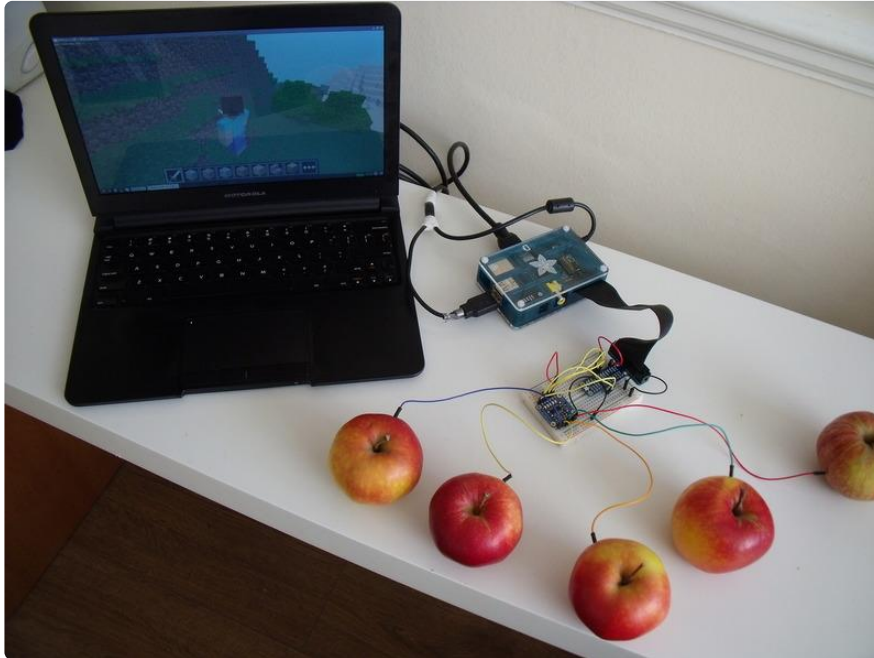
Last updated on 2024-06-03 01:27:16 PM EDT

Table of Contents

Overview	3
<ul style="list-style-type: none">• Momentary• Toggle• 5-Pad	
Assembly	5
<ul style="list-style-type: none">• Headers• Place the Board• Solder• Admire	
Wiring	6
<ul style="list-style-type: none">• Momentary Board Wiring• 40-Pin (A, B, B+ and Zero) T-Cobbler Plus Schematic• 20-Pin (Raspberry Pi Rev 1 and Rev 2) Cobbler Schematic• Toggle Board Wiring• 40-Pin (A, B, B+ and Zero) T-Cobbler Plus Schematic• 20-Pin (Raspberry Pi Rev 1 and Rev 2) Cobbler Schematic• 5-Pad Sensor Wiring• 40-Pin (A, B, B+ and Zero) T-Cobbler Plus Schematic• 20-Pin (Raspberry Pi Rev 1 and Rev 2) Cobbler Schematic• LEDs• Connecting to Objects	
Programming	11
<ul style="list-style-type: none">• CircuitPython Library Setup• Continuous Output• Single Output• Running the Code	
Programming the 5-Pad Sensor	14
<ul style="list-style-type: none">• Running the Code	
Controlling Minecraft: Pi Edition	15
<ul style="list-style-type: none">• Install Minecraft: Pi Edition• Copy the API• Super Jump• Melons• Control Panel (5-Pad Board)• Download the Code• Running the Code	

Overview

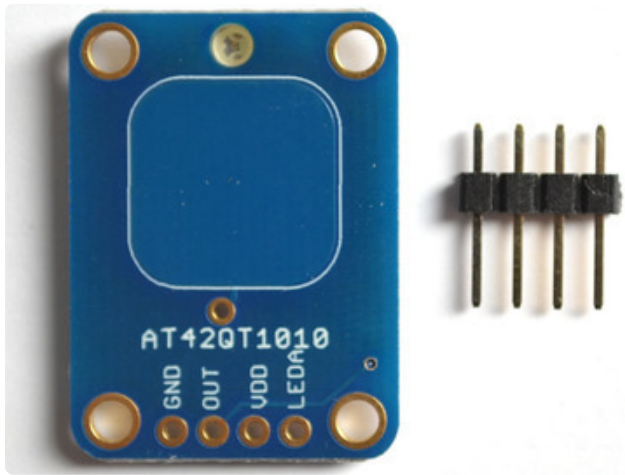
Capacitive touch breakout boards are an excellent way to use household objects as inputs on your Raspberry Pi. Any conductive object can act as a switch when connected to sensor boards, including potatoes, apples, spoons and pencil graphite.



The capacitive touch sensors detect when you touch the board's pad or an object connected to the board. Humans carry a small electrical charge. When you touch a capacitive touch sensor it will detect this charge.

With some simple wiring to the Raspberry Pi and a few lines of Python, you've got a fun and easy way to control your projects and games. For example at the end of this guide we'll show you how to control Minecraft: Pi Edition using a apples.

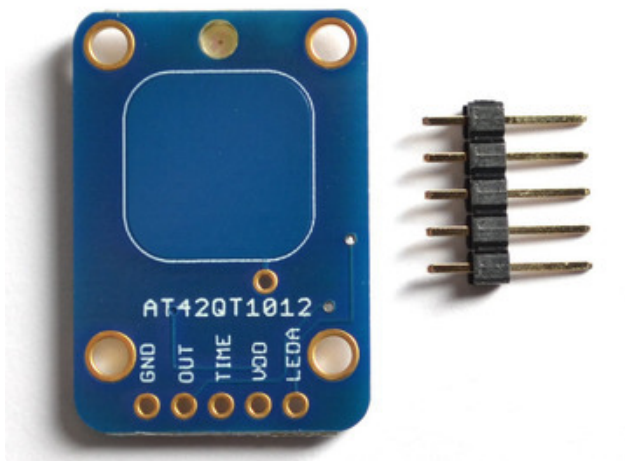
There are three breakout boards that this guide will show you how to use with your Raspberry Pi.



Momentary

The [momentary capacitive touch sensor](http://adafru.it/1374) (<http://adafru.it/1374>) will be active as long as something is touching it. The LED shows whether a touch has been detected.

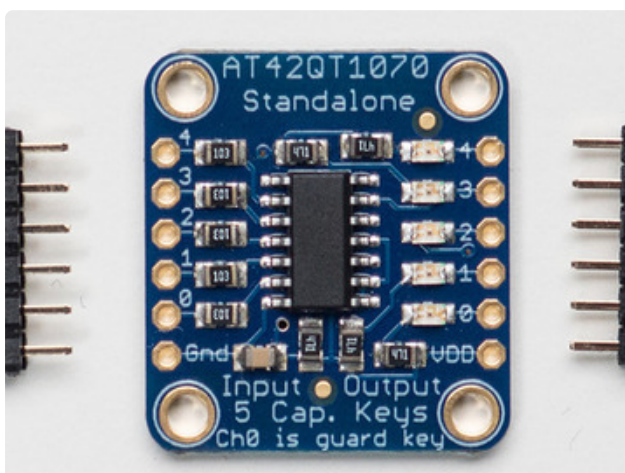
The board has a large touch-pad. The small copper hole near the touch-pad can be used to connect the board to capacitive items such as a drawing made with pencil graphite.



Toggle

The [toggle capacitive touch sensor](http://adafru.it/1375) (<http://adafru.it/1375>) will activate when you press it and deactivate when you press it a second time. The LED will light up to indicate whether your touch has been detected.

Like the momentary sensor, this sensor has a large touch-pad. The small copper hole near the touch-pad allows you to use a wire to connect your board to everyday objects, like a spoon, which can then be used as an input.



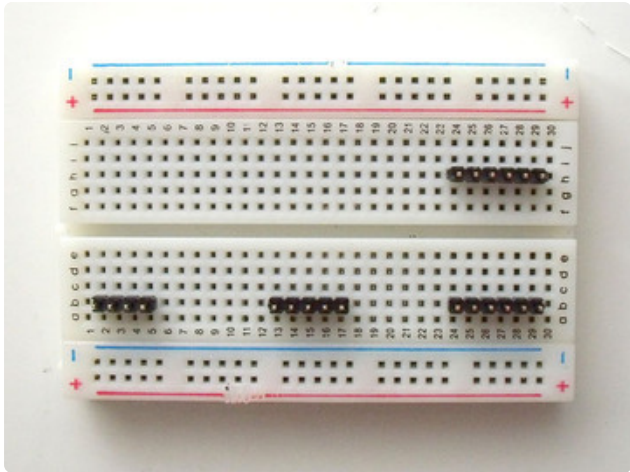
5-Pad

The [5-Pad capacitive touch sensor](http://adafru.it/1362) (<http://adafru.it/1362>) offers a total of 5 inputs on one board. It is very similar to the momentary board, but includes more inputs. Unlike the other boards it does not have any pads, instead the pins on the right of the board are used to detect touch. These pins can be connected to wires, which can be attached to everyday objects, like bananas.

Assembly

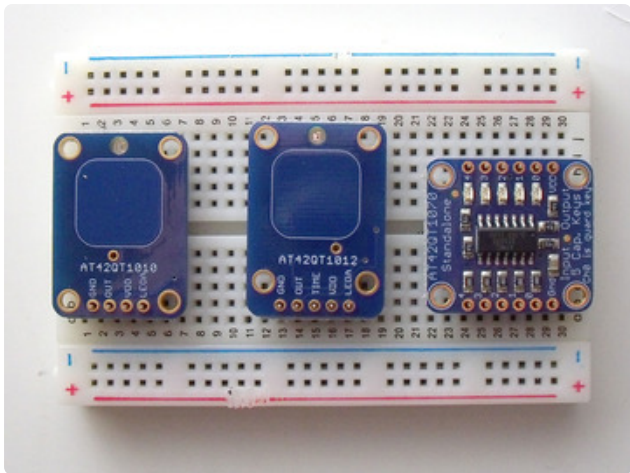
Assembling the capacitive touch sensors is very straightforward and can be achieved with minimal soldering.

If you're new to soldering, check out the [Adafruit Guide to Excellent Soldering \(https://adafru.it/dgp\)](https://adafru.it/dgp).



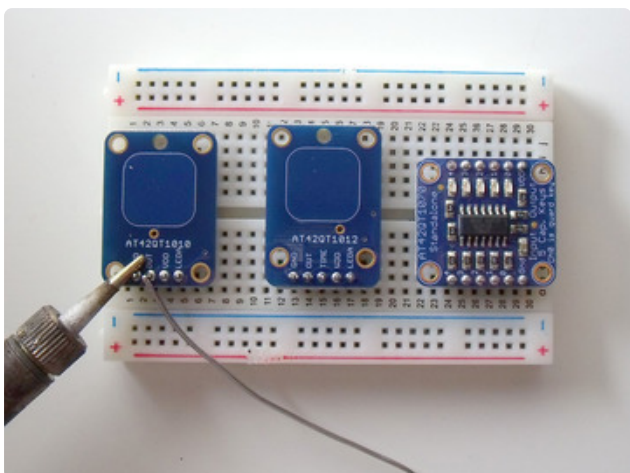
Headers

Cut the headers and place them onto a solderless breadboard.



Place the Board

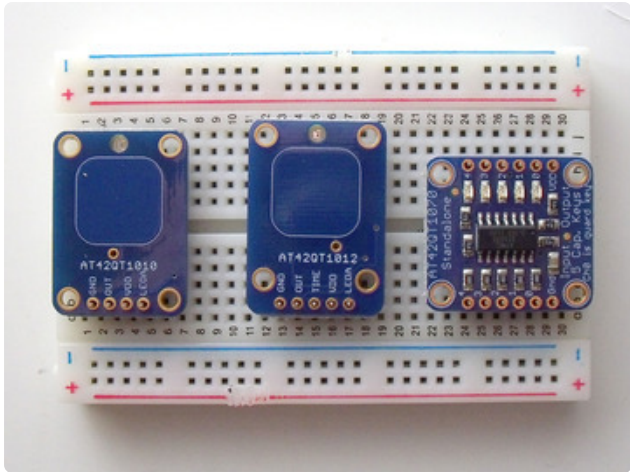
Place the boards onto the headers and check that they are spaced correctly.



Solder

Solder the pins with a soldering iron.

To get the momentary and toggle boards to lie flat on the headers, solder a single pin. Reheat the solder on the pin and adjust the board until you are happy with its positioning.



Admire

Take a moment to sit back and admire your work.

Wiring

In this section we'll learn how to wire up the sensor boards to a Raspberry Pi. For this you will need:

- A capacitive touch sensor ([momentary \(http://adafru.it/1374\)](http://adafru.it/1374), [toggle \(http://adafru.it/1375\)](http://adafru.it/1375) or [5-pad \(http://adafru.it/1362\)](http://adafru.it/1362))
- [A solderless breadboard \(http://adafru.it/64\)](http://adafru.it/64)
- [Breadboarding/jumper wire \(http://adafru.it/153\)](http://adafru.it/153)
- Pi Cobbler (pick one):
 - 40-pin (Pi - A, B, B+, Zero) - [Assembled Pi Cobbler Plus \(http://adafru.it/2029\)](http://adafru.it/2029)
 - 40-pin (Pi - A, B, B+, Zero)- [Assembled Pi T-Cobbler Plus \(http://adafru.it/2028\)](http://adafru.it/2028)
 - 20-pin (Pi - Rev 1 and 2 only) - [Assembled Pi T-Cobbler \(http://adafru.it/1754\)](http://adafru.it/1754)
- A 10k resistor (no resistors are required for the 5-pad board)
- Apples, potatoes, bananas or another fun object that you can connect the board to
- [A Raspberry Pi \(http://adafru.it/3775\)](http://adafru.it/3775) - All Models except the compute node are supported.
- Power supply, mouse, monitor, monitor cable, keyboard and SD card for the Raspberry Pi

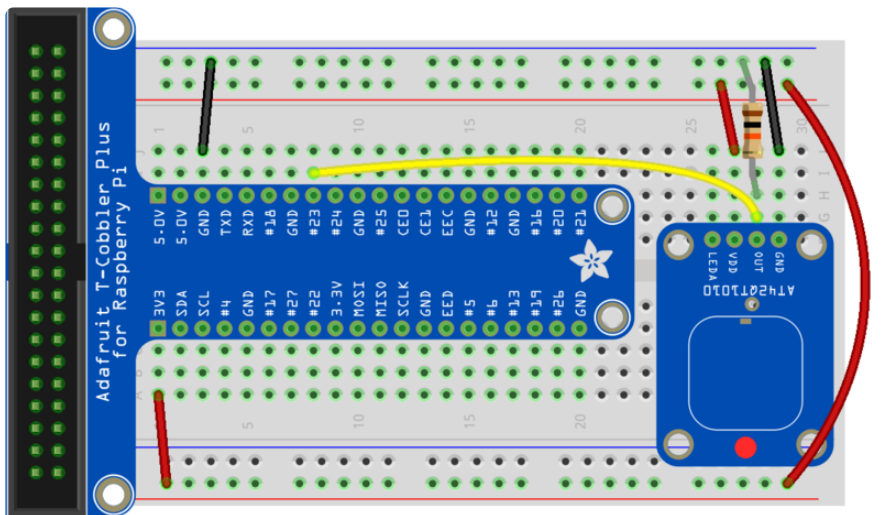
Momentary Board Wiring

When wiring my boards I prefer to use Adafruit's Pi Cobbler to connect my Raspberry

Pi to the breadboard, but you can also use jumper wire to connect to the Raspberry Pi's GPIO instead.

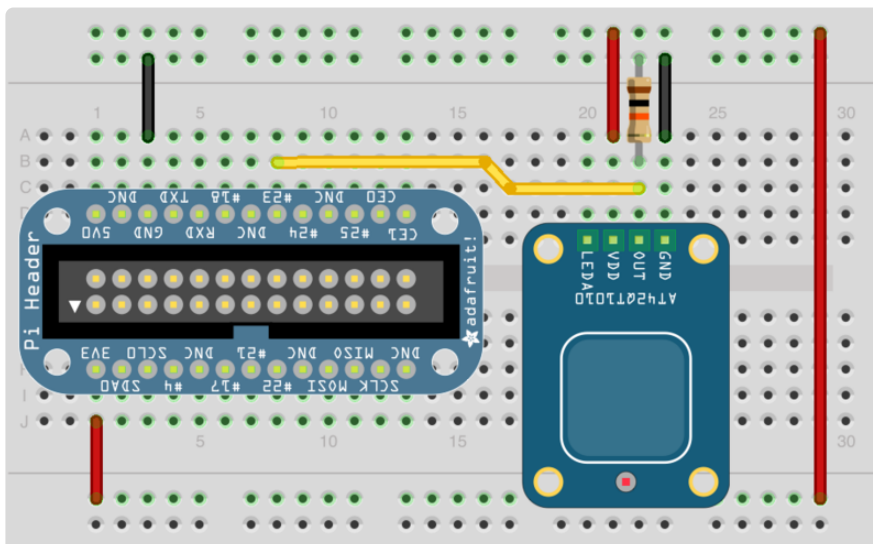
Below there are both 40-pin and 20-pin schematics for modern and older Raspberry Pi models.

40-Pin (A, B, B+ and Zero) T-Cobbler Plus Schematic



fritzing

20-Pin (Raspberry Pi Rev 1 and Rev 2) Cobbler Schematic



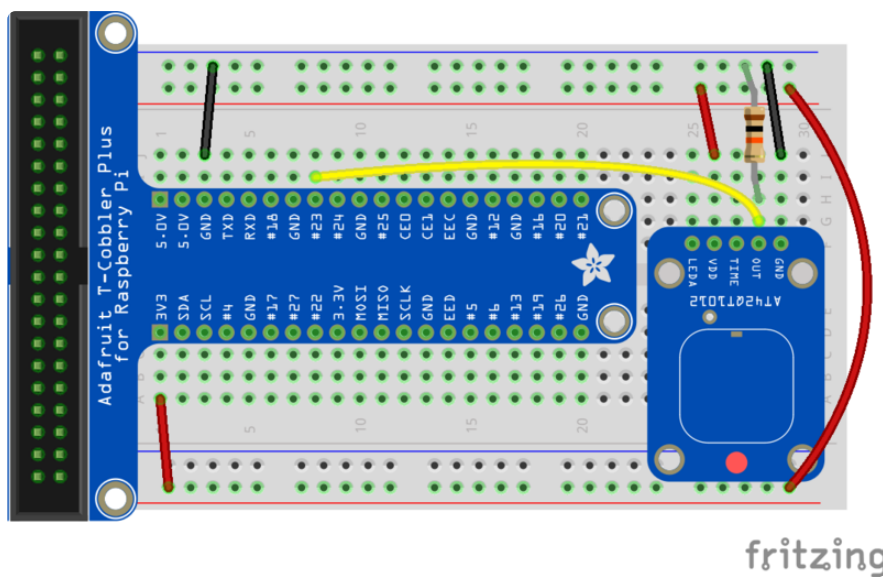
Made with  Fritzing.org

Toggle Board Wiring

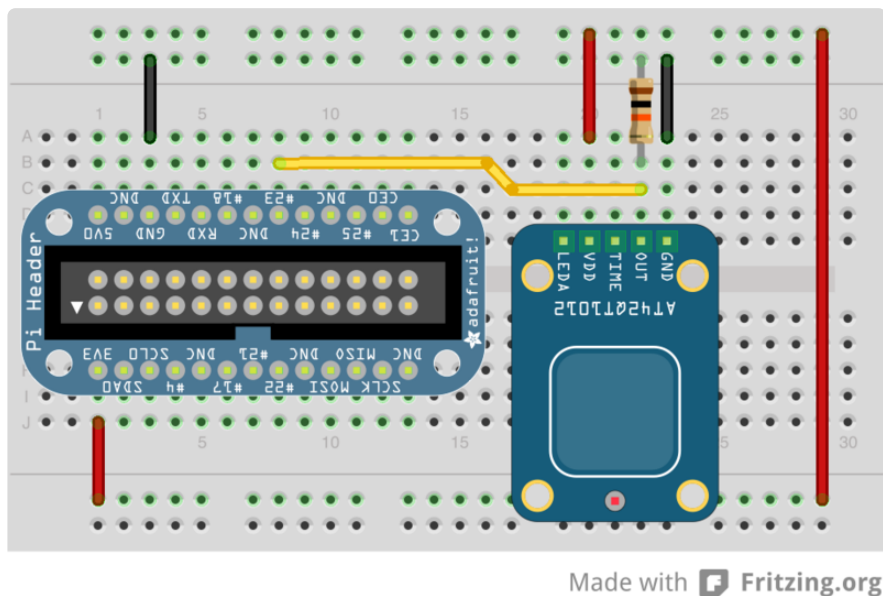
The wiring for the momentary and toggle breakouts is not identical. The momentary board uses 4 header pins whereas the toggle board uses 5 header pins. The VCC is in a different position on the two boards.

Below there are both 40-pin and 20-pin schematics for modern and older Raspberry Pi models.

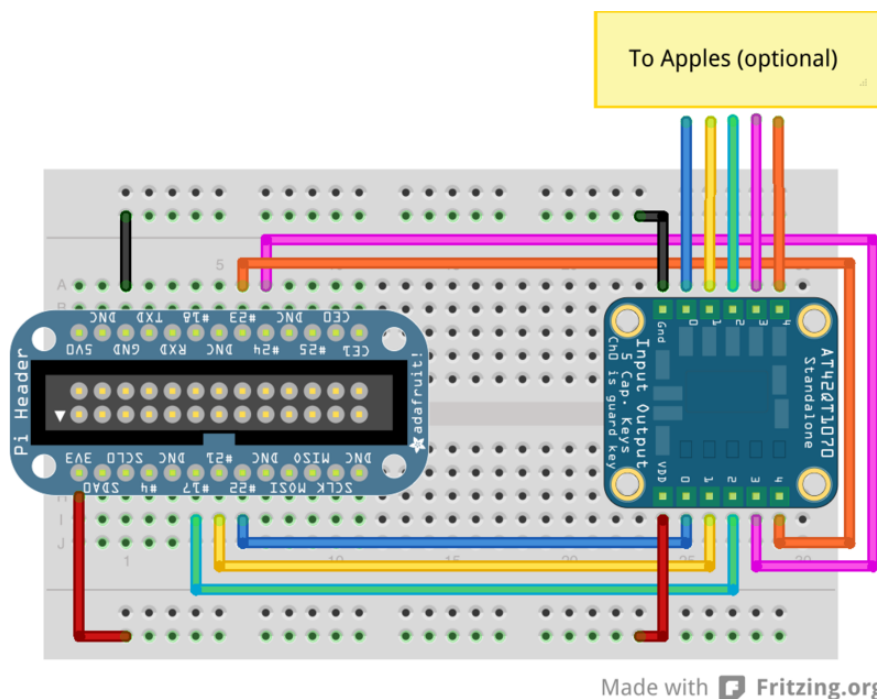
40-Pin (A, B, B+ and Zero) T-Cobbler Plus Schematic



20-Pin (Raspberry Pi Rev 1 and Rev 2) Cobbler Schematic



20-Pin (Raspberry Pi Rev 1 and Rev 2) Cobbler Schematic



When using the 5-pad capacitive touch sensor make sure you don't cross the input wires. When the wires touch they will both detect the same touch when either is pressed, which can cause unexpected results.

LEDs

When you press the pad on the board the LED should light up. You must have your Raspberry Pi switched on and connected to the board in order for this to happen. Even if you're not running any programs that use the GPIOs, the LEDs will still light up as the boards are connected to the power and ground pins on the Pi.

If it works, well done, you're ready to move onto creating a Python program that uses the board.

The 5-pad board has 5 LEDs, one for each input.

Connecting to Objects

Each of the boards can be connected to everyday objects that are conductive. This includes fruit, vegetables, plants, metals, animals, conductive fabrics, pencil graphite and conductive paint to name a few.

With some objects you may find that the sensor will be activated when your hand is near the object, but not touching it. This is normal and very common with fruits and vegetables that contain a lot of water.

Adafruit also stocks a number of materials which are excellent for connecting your sensors to:

- [Bare Conductive paint \(http://adafru.it/1305\)](http://adafru.it/1305) (also [available as a pen \(http://adafru.it/1306\)](http://adafru.it/1306))
- [Copper tape \(http://adafru.it/1128\)](http://adafru.it/1128)
- [Conductive fabric \(http://adafru.it/1167\)](http://adafru.it/1167)
- [Conductive thread \(http://adafru.it/641\)](http://adafru.it/641)
- [ITO \(Indium Tin Oxide\) Coated PET Plastic \(http://adafru.it/1309\)](http://adafru.it/1309)

Materials that are not conductive will not work. Plastics, wood, fabric and glass are all not conductors.

A tip: Although your novel idea of using raspberries with your capacitive touch sensor and your Raspberry Pi may seem like a stroke of genius, the raspberries are very soft and will quickly turn to mush. Try using firmer fruit instead, like an apple or a melon.

Programming

These example programs show you how to use the capacitive touch sensors with our CircuitPython Libraries that are part of adafruit-blinka. It is relatively straightforward to adapt the programs to do different things. Just change the lines with print "pressed".

Two versions of the code are provided. One will continuously print output to the terminal while the touch sensor is pressed. The other will only print output once, irrelevant of how long the pad is pressed. Both pieces of code will work with the momentary and toggle boards.

Code for 5-pad board can be found on the next page of the guide.

The software is exactly the same on all 40-pin and 20-pin Raspberry Pi models.

Make sure you have connected the boards using the wiring instructions in the previous section before continuing.

CircuitPython Library Setup

See [CircuitPython Libraries on Raspberry Pi \(https://adafru.it/Deo\)](https://adafru.it/Deo) to get a fresh Raspberry Pi setup.

If you have a running Raspberry Pi with an up to date copy of Raspbian you can simply run the following command to install adafruit-blinka.

```
$ sudo pip3 install adafruit-blinka
```

Continuous Output

This version of the code will continuously print output while the pad is pressed.

```
# SPDX-FileCopyrightText: 2019 Mikey Sklar for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
from digitalio import DigitalInOut, Direction

pad_pin = board.D23

pad = DigitalInOut(pad_pin)
pad.direction = Direction.INPUT

while True:
    if pad.value:
        print("pressed")

    time.sleep(0.1)
```

Single Output

This version of the Python code will only print an output once each time the sensor detects a touch.

```
# SPDX-FileCopyrightText: 2019 Mikey Sklar for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
from digitalio import DigitalInOut, Direction

pad_pin = board.D23

pad = DigitalInOut(pad_pin)
pad.direction = Direction.INPUT

already_pressed = False
```

```
while True:
    if pad.value and not already_pressed:
        print("pressed")

    already_pressed = pad.value
    time.sleep(0.1)
```

Running the Code

Open a terminal on your Pi.

Pull Down the two scripts directly onto your Raspberry Pi using wget.

```
$ wget https://raw.githubusercontent.com/adafruit/Adafruit_Learning_System_Guides/master/Capacitive_Touch_Sensors_on_the_Raspberry_Pi/Continuous_Output.py
$ wget https://raw.githubusercontent.com/adafruit/Adafruit_Learning_System_Guides/master/Capacitive_Touch_Sensors_on_the_Raspberry_Pi/Single_Output.py
```

Run the code using the following commands. Try tapping vs holding your finger on the keypad. You should see "pressed" output in continuous mode or per tap depending on the script.

```
$ sudo python3 ./Continuous_Output.py
```

```
$ sudo python3 ./Continuous_Output.py
pressed
pressed
pressed
pressed
```

```
$ sudo python3 ./Single_Output.py
```

```
$ sudo python3 ./Single_Output.py
pressed
pressed
pressed
pressed
```

Programming the 5-Pad Sensor

The code for the 5-pad capacitive touch sensor can be found below. It cannot be swapped with either of the single pad boards as they are wired differently.

```
# SPDX-FileCopyrightText: 2019 Mikey Sklar for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
from digitalio import DigitalInOut, Direction

# set the GPIO input pins
pad0_pin = board.D22
pad1_pin = board.D21
pad2_pin = board.D17
pad3_pin = board.D24
pad4_pin = board.D23

pad0 = DigitalInOut(pad0_pin)
pad1 = DigitalInOut(pad1_pin)
pad2 = DigitalInOut(pad2_pin)
pad3 = DigitalInOut(pad3_pin)
pad4 = DigitalInOut(pad4_pin)

pad0.direction = Direction.INPUT
pad1.direction = Direction.INPUT
pad2.direction = Direction.INPUT
pad3.direction = Direction.INPUT
pad4.direction = Direction.INPUT

pad0_already_pressed = True
pad1_already_pressed = True
pad2_already_pressed = True
pad3_already_pressed = True
pad4_already_pressed = True

while True:

    if pad0.value and not pad0_already_pressed:
        print("Pad 0 pressed")
        pad0_already_pressed = pad0.value

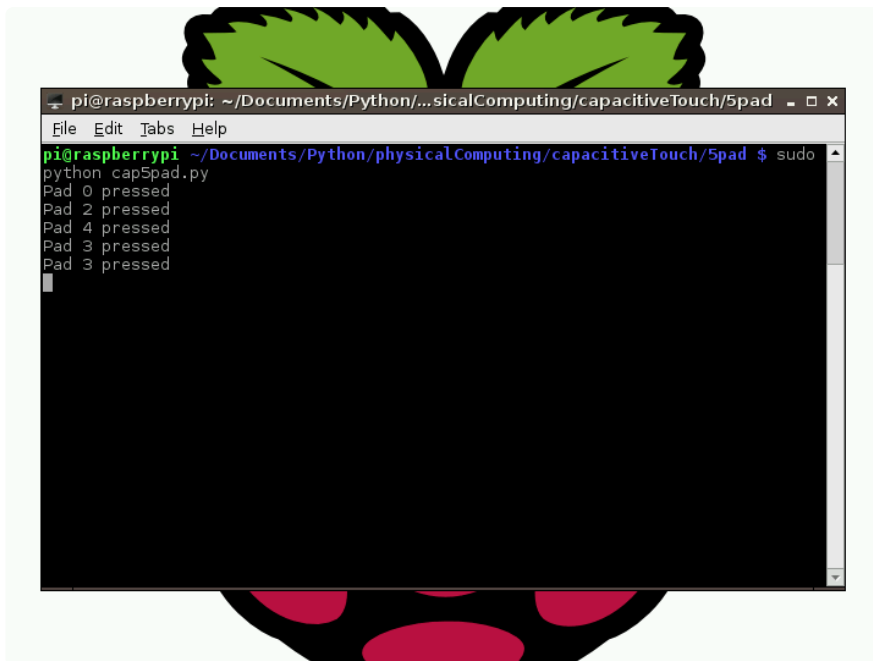
    if pad1.value and not pad1_already_pressed:
        print("Pad 1 pressed")
        pad1_already_pressed = pad1.value

    if pad2.value and not pad2_already_pressed:
        print("Pad 2 pressed")
        pad2_already_pressed = pad2.value

    if pad3.value and not pad3_already_pressed:
        print("Pad 3 pressed")
        pad3_already_pressed = pad3.value

    if pad4.value and not pad4_already_pressed:
        print("Pad 4 pressed")
        pad4_already_pressed = pad4.value

    time.sleep(0.1)
```



Running the Code

Open a terminal on your Pi.

Pull Down the script directly onto your Raspberry Pi using wget.

```
$ wget https://raw.githubusercontent.com/adafruit/Adafruit_Learning_System_Guides/master/Capacitive_Touch_Sensors_on_the_Raspberry_Pi/5pad.py
```

```
$ sudo python3 ./5pad.py
```

The program works in a very similar way to the programs for the momentary and toggle sensor boards. If you're having trouble understanding how the code works for the 5-pad, have a look at the code on the previous section of this guide first.

Controlling Minecraft: Pi Edition

The GPIO pins on the Raspberry Pi are one of the reasons for its popularity. In this guide we've used the CircuitPython libraries to connect to capacitive touch sensors. This has a range of applications and allows you to create some really creative projects.

The Raspberry Pi has a special version of Minecraft that can be programmed with Python. Combining Minecraft on the Raspberry Pi with our capacitive touch sensors allows us to create programs with limitless potential for creativity and fun.

Install Minecraft: Pi Edition

Before we begin you'll need to have Minecraft: Pi Edition installed on your Raspberry Pi. To install Minecraft Pi you can use [this guide on the Adafruit Learning System](https://adafru.it/dgq) (<https://adafru.it/dgq>).

Copy the API

In order to connect to a Minecraft game you will need to copy the API.

Open a terminal and create a new directory named Minecraft within the Documents directory with the following command:

```
mkdir ~/Documents/Minecraft
```

Next, we will copy the API into this folder with the following command:

```
cp -r ~/mcpi/api/python/ ~/Documents/Minecraft
```

Super Jump

This code uses the capacitive touch pad with Minecraft: Pi Edition to make the player jump 10 blocks into the air when the pad is pressed.

This code works best with the momentary board, though can be used with the toggle board and can be adapted for the 5-pad board.

```
# SPDX-FileCopyrightText: 2019 Mikey Sklar for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
import mcpi.minecraft as minecraft
from digitalio import DigitalInOut, Direction

mc = minecraft.Minecraft.create()

pad_pin = board.D23
pad = DigitalInOut(pad_pin)
pad.direction = Direction.INPUT

already_pressed = False

while True:
    if pad.value and not already_pressed:
        pos = mc.player.getPos()
        x = pos.x
        y = pos.y + 10
```

```
z = pos.z
mc.player.setPos(x, y, z)
alreadyPressed = pad.value
time.sleep(0.1)
```

Melons

This code will place a melon block at the current position of the player. If you hold down the touch pad of the capacitive touch sensor or use toggle sensor, a trail of melons will be created wherever the player walks. This example works particularly well with the toggle sensor board.

```
# SPDX-FileCopyrightText: 2019 Mikey Sklar for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
import mcpi.minecraft as minecraft
from digitalio import DigitalInOut, Direction

mc = minecraft.Minecraft.create()

pad_pin = board.D23

pad = DigitalInOut(pad_pin)
pad.direction = Direction.INPUT

# melon block
block_type = 103

while True:

    if pad.value:
        pos = mc.player.getPos()
        x = pos.x
        y = pos.y
        z = pos.z
        mc.setBlock(x, y, z, block_type)

    time.sleep(0.1)
```

Control Panel (5-Pad Board)

In the last example we use the 5-pad capacitive touch sensor board to do 5 different things in Minecraft Pi.

This is what each pad does:

- 0 - Teleports the player to co-ordinates (0, 0, 0)
- 1 - Places a flower at the player's position
- 2 - Places explosive TNT at the player's position
- 3 - Checks if the player is in water
- 4 - Toggles whether or not blocks can be smashed

```

# SPDX-FileCopyrightText: 2019 Mikey Sklar for Adafruit Industries
#
# SPDX-License-Identifier: MIT

import time
import board
import mcpi.minecraft as minecraft
from digitalio import DigitalInOut, Direction
mc = minecraft.Minecraft.create()

# set the GPIO input pins
pad0_pin = board.D22
pad1_pin = board.D21
pad2_pin = board.D17
pad3_pin = board.D24
pad4_pin = board.D23

pad0 = DigitalInOut(pad0_pin)
pad1 = DigitalInOut(pad1_pin)
pad2 = DigitalInOut(pad2_pin)
pad3 = DigitalInOut(pad3_pin)
pad4 = DigitalInOut(pad4_pin)

pad0.direction = Direction.INPUT
pad1.direction = Direction.INPUT
pad2.direction = Direction.INPUT
pad3.direction = Direction.INPUT
pad4.direction = Direction.INPUT

pad0_already_pressed = False
pad3_already_pressed = False
pad4_already_pressed = False

immutable = False

tnt = 46
water = 9
flowers = 38

while True:
    pad0_pressed = not pad0.value
    pad1_pressed = not pad1.value
    pad2_pressed = not pad2.value
    pad3_pressed = not pad3.value
    pad4_pressed = not pad4.value

    if pad0_pressed and not pad0_already_pressed:
        #teleport
        x = 0
        y = 0
        z = 0
        mc.player.setPos(x, y, z)
        pad0_already_pressed = pad0_pressed

    if pad1_pressed:
        #Flowers
        pos = mc.player.getPos()
        x = pos.x
        y = pos.y
        z = pos.z
        mc.setBlock(x, y, z, flowers)

    if pad2_pressed:
        #TNT
        pos = mc.player.getPos()
        x = pos.x
        y = pos.y
        z = pos.z

```

```
mc.setBlock(x, y, z, tnt, 1)

if pad3_pressed and not pad3_already_pressed:
    #Chat message: Are in water?
    pos = mc.player.getPos()
    block = mc.getBlock(pos.x, pos.y, pos.z)
    inWater = block == water
    mc.postToChat("In water: " + str(inWater))
    pad3_already_pressed = pad3_pressed

if pad4_pressed and not pad4_already_pressed:
    #Immutable
    immutable = not immutable
    mc.setting("world_immutable", immutable)
    mc.postToChat("Immutable: " + str(immutable))
    pad4_already_pressed = pad4_pressed

time.sleep(0.1)
```

Download the Code

Open a terminal on your Pi.

Pull Down the above three scripts directly onto your Raspberry Pi using wget.

```
$ wget https://raw.githubusercontent.com/adafruit/Adafruit_Learning_System_Guides/master/Capacitive_Touch_Sensors_on_the_Raspberry_Pi/MC-Super_Jump.py
$ wget https://raw.githubusercontent.com/adafruit/Adafruit_Learning_System_Guides/master/Capacitive_Touch_Sensors_on_the_Raspberry_Pi/MC-Melons.py
$ wget https://raw.githubusercontent.com/adafruit/Adafruit_Learning_System_Guides/master/Capacitive_Touch_Sensors_on_the_Raspberry_Pi/MC-5pad.py
```

Running the Code

To run any of the examples on this page you will need to be in Minecraft game world on your Raspberry Pi. You will get an error if you don't have Minecraft open or if you're on the title screen.

All of these examples use the same wiring as the rest of this guide.

The code runs like any other Python program that uses the GPIO. You can run it from the terminal on your Raspberry Pi with the following commands:

```
$ sudo python3 ./MC-Super_Jump.py
```

```
$ sudo python3 ./MC-Melons.py
```

```
$ sudo python3 ./MC-5pad.py
```