App Inventor + IoT: Basic breadboard circuit	90 mins

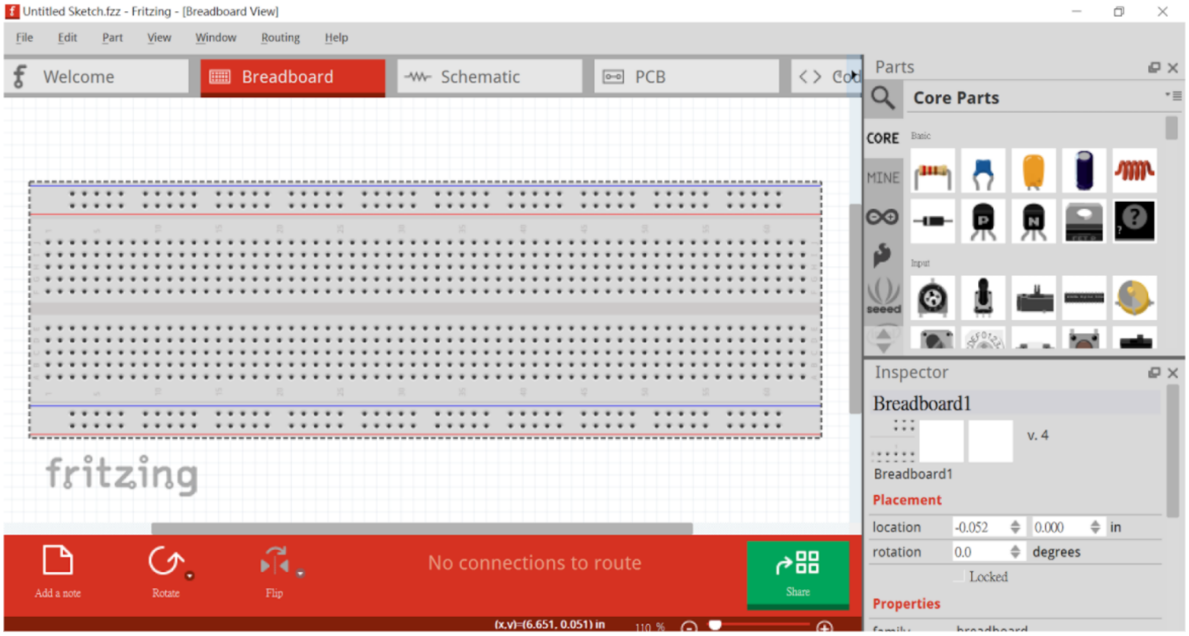
Level: beginner

Introduction

This tutorial shows you how to build a simple circuit for MIT App Inventor’s IoT projects. After reading this tutorial, you will have a basic idea of:

1. the role of MCU boards in IoT
2. common circuit components, like LED, potentiometer, servo, etc.
3. how to use breadboard to connect circuit
4. utilize your project with extension board and starter kits

All circuit in this tutorial is drawn by **Fritzing**, which is a very popular software for drawing circuits and which allows user to import their customized components/products. Please download Fritzing according to your operating system. Fritzing’s main screen is shown below:



location -0.052 0.000 in
rotation 0.0 degrees
□ Locked
family breadboard

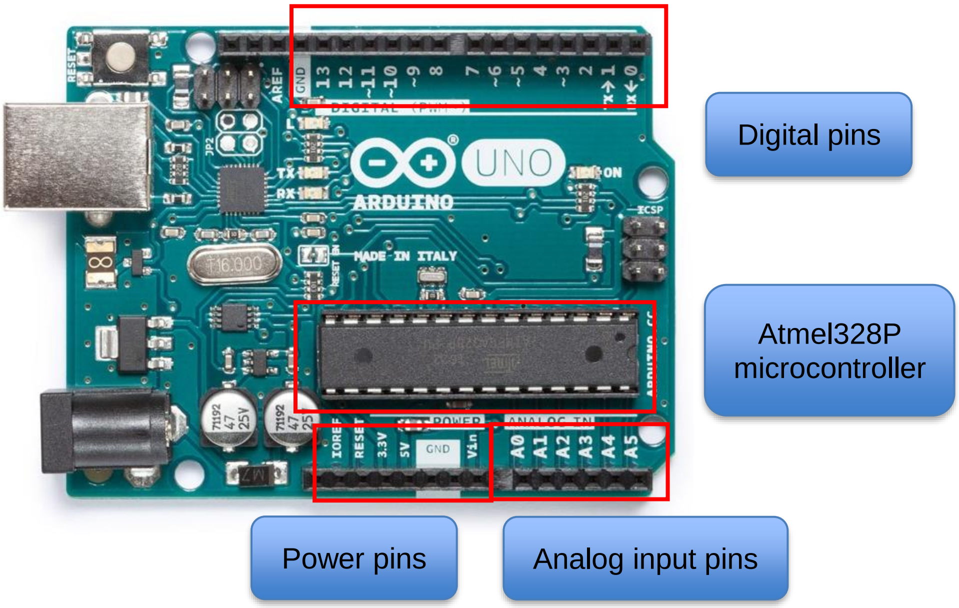
(x,y)=(6.651, 0.051) in 110 %

MCU boards

“MCU” means “microcontroller unit.” An MCU board is a development board with a microcontroller on it. Users can write programs to access its pins to control/read various components like LED, motor, potentiometer, etc. These components are connected with MCU board through breadboard or extension boards.

The most popular MCU boards for zero experience users and makers is Arduino platform. With its open-source spirit, you can easily realize whether an idea is workable and get the prerequisites (software, hardware, etc.).

The figure below is the Arduino Uno board. You can see there is a **Atmel328P** microcontroller on it. Pin0 to P13 are digital input/output pins to connect digital output components like LEDs, servos, and buzzers; and digital input components like buttons.



Arduino Uno R3 board (image from https://store.arduino.cc/usa/arduino-uno-rev3)

In the IoT domain, the MCU boards need communication abilities such as Internet(Ethernet/Wi-Fi) or Bluetooth. For boards which support Ethernet or Wi-Fi (LinkIt 7688 and LinkIt 7697), they can connect to the internet directly, and App Inventor can interact with them through Wi-Fi.

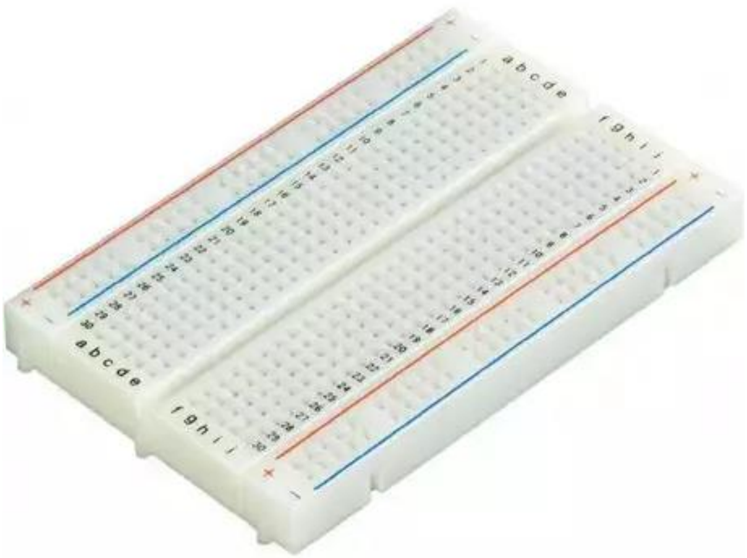
However for boards that support Bluetooth only, (such as Arduino 101, BBC Micro:bit and Lego EV3), they can not connect to the Internet directly. Therefore, they must first connected with App Inventor, then connect to the internet after while, therefore App Inventor in this case is like a middleware.

On the **MIT App Inventor IoT website**, you can see many tutorials demonstrating how to interact with these kind of MCU boards with App Inventor. This means you can control its pin and read its pin status through Bluetooth or Wi-Fi connection.

Breadboard

Breadboard is a handy stuff for circuit prototyping. You can plug various components into it and connect them by wires. You can modify the circuit easily by switching the wires and component, like changing the LED to another pin.

However, the drawback of using breadboard is that the connections are somehow loose and components on it are easy to fall apart. Breadboards come in various rectangular sizes, here is a photo of a wide-used 400-hole breadboard:

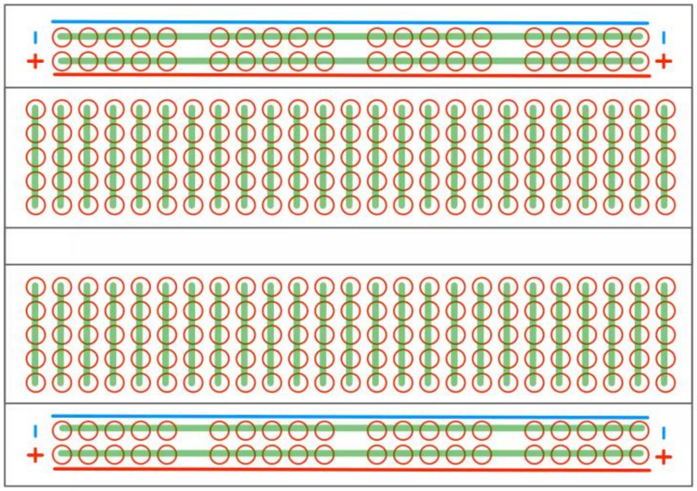


From the figure below, we can see how the hole are interconnected.

The two sections of our breadboard are composed of many columns, each has five holes. We have to insert the component pin into these holes to build a complete circuit.

The four rows on the top and bottom of the breadboard are power rails(**+**) and GND rails (**-**). Since Arduino board usually has only one 5V pin, you can connect all components' **5V** (sometimes noted as **Vcc** or **+**) pins to any one of the holes on this power rail then connect Arduino's **5V** pin to this rail. In this way, all components' **5V** pins connected to this rail can get **5V** power supply from Arduino. GND rail is the same idea, all components' **GND** pins connected to this rail are also connected to ground to form a complete circuit.

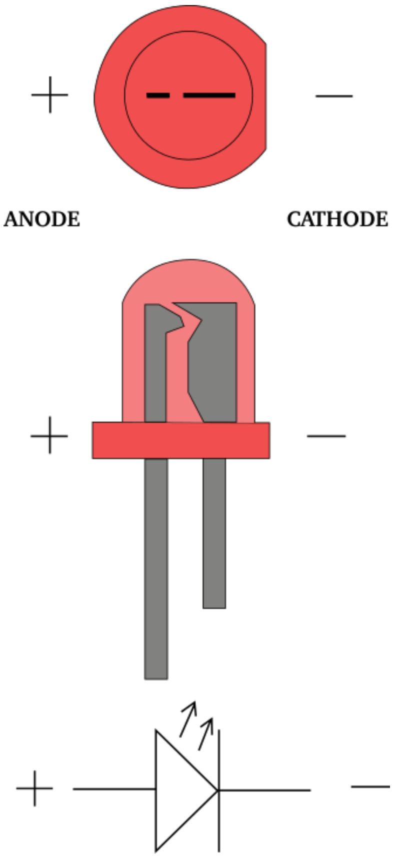
Notice there is a gap in the middle of almost every breadboard, the two sections are not connected with each other. You can use a wire or propriate resistor to link these two sections.



LED (digital output component)

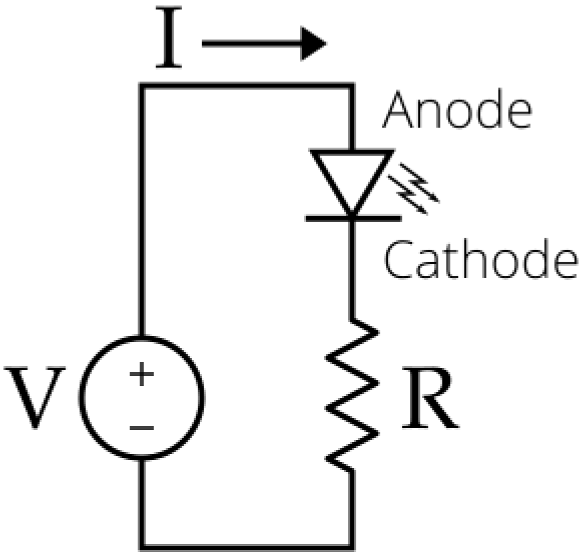
LED (Light Emitted Diode) is the most common and maybe the first component for everyone's circuit prototype. When you connect the LED's longer pin (**anode**) to the battery's positive (**+**) terminal, and connect LED's shorter pin (**cathode**) to the battery's negative (**-)** terminal, the current from the battery will light up the LED right away.

Note: specification of each LED differs from each other, be sure to the check the datasheet of the actual LED you used.



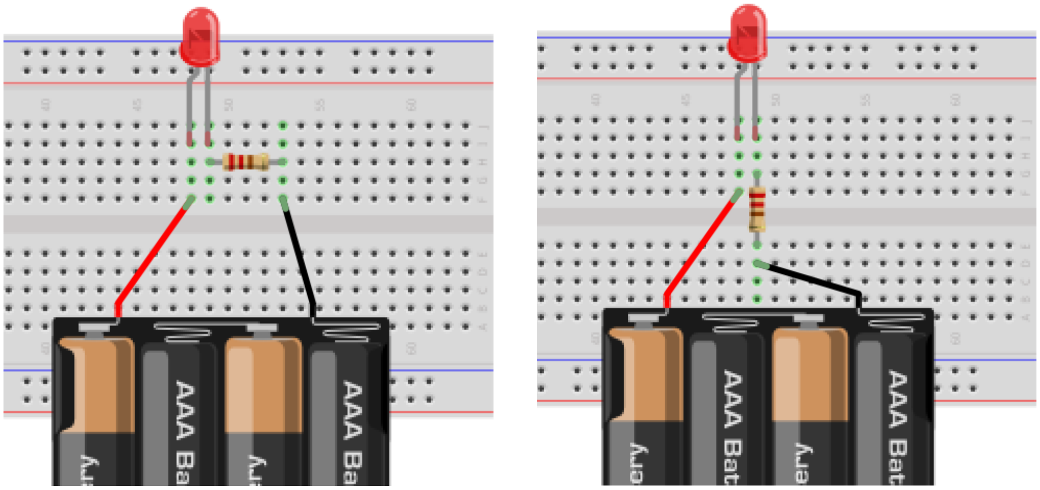
(Image from: https://en.wikipedia.org/wiki/LED_circuit)

The way to light up a LED is illustrated below:



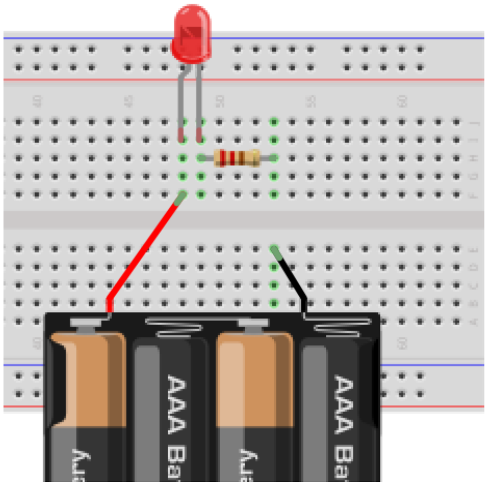
(Image from: https://en.wikipedia.org/wiki/LED_circuit)

This is a simple LED circuit, we usually use a 220-ohm resistor to protect the LED to be burned by a current.



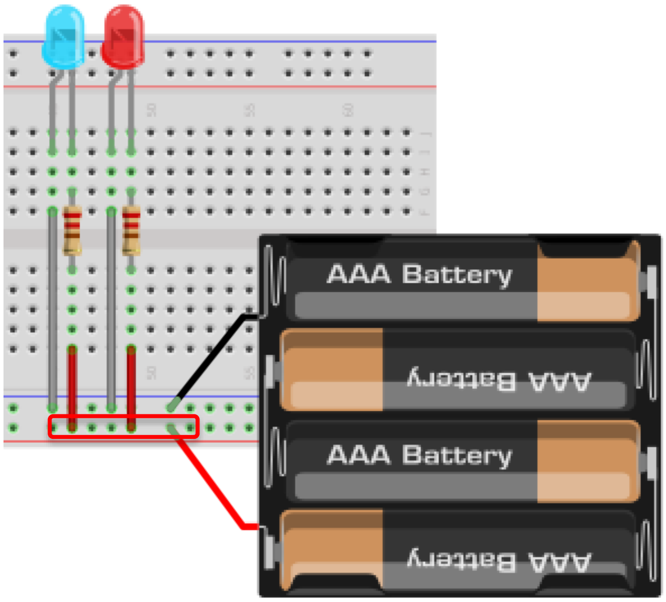
Connected

This circuit is not complete since the upper and lower section of the breadboard are divided.



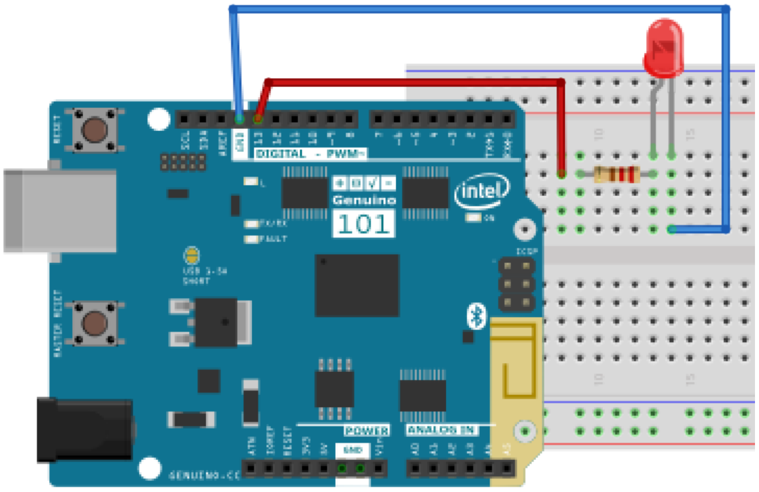
The resistor and the black wire are NOT connected together.

You can see that the two LEDs are all connected to the battery through the power (**+**) and GND (**-**) rail of the breadboard. Please check the red rectangle to emphasize how to connect LEDs' longer pins and Battery's positive terminal to the power rail.

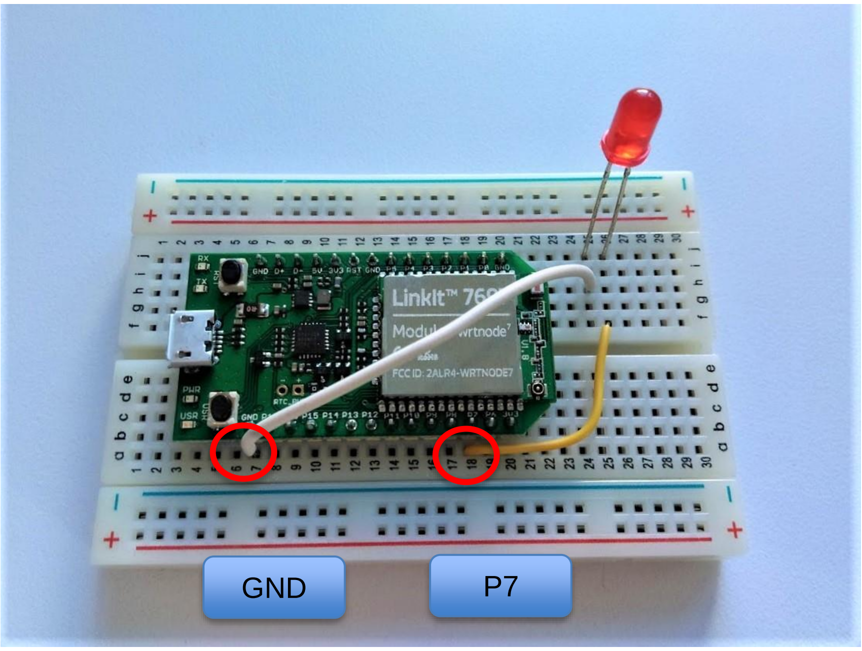


MCU board and breadboard

At this point, we cannot control the circuits above. One approach is to add an MCU board and control the circuit with its pins. Let’s connect a LED on Arduino 101 and LinkIt 7697:



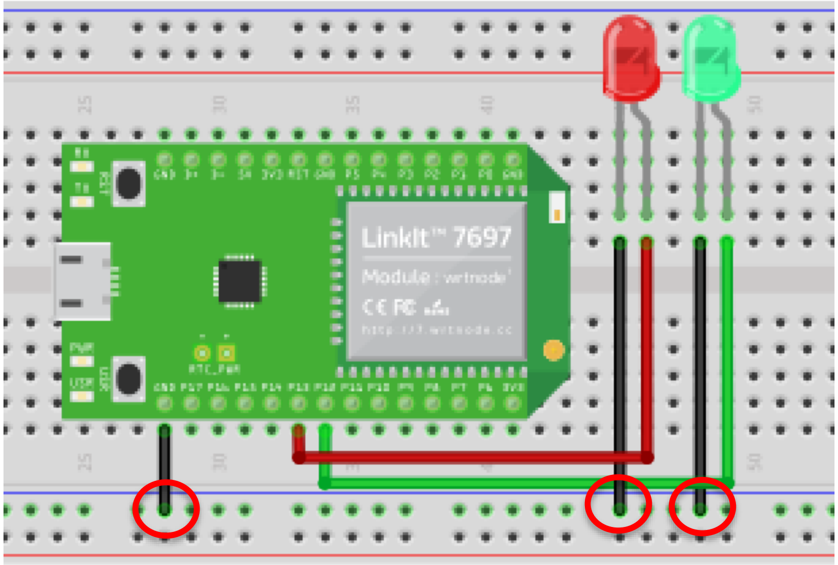
(Connect the LED’s longer pin to Arduino 101’s D13 pin)



(Connect the LED’s longer pin to LinkIt 7697’s P7 pin)

Use power rails to share 5V and GND pins

Another way to use a breadboard is two LEDs are sharing the same GND pin. They are all connected on the breadboard GND (**+**) rail as the red circles below. But since we have control them by MCU board, the longer pins of these LEDs have to be connected to different pins.



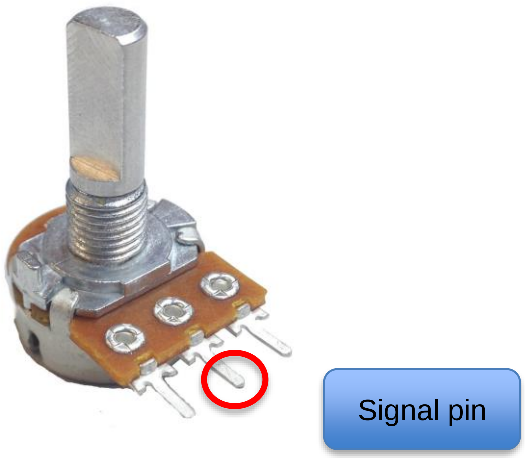
To learn how to control an LED with your Arduino only, please visit: https://www.arduino.cc/en/Tutorial/Blink

Potentiometer (analog input component)

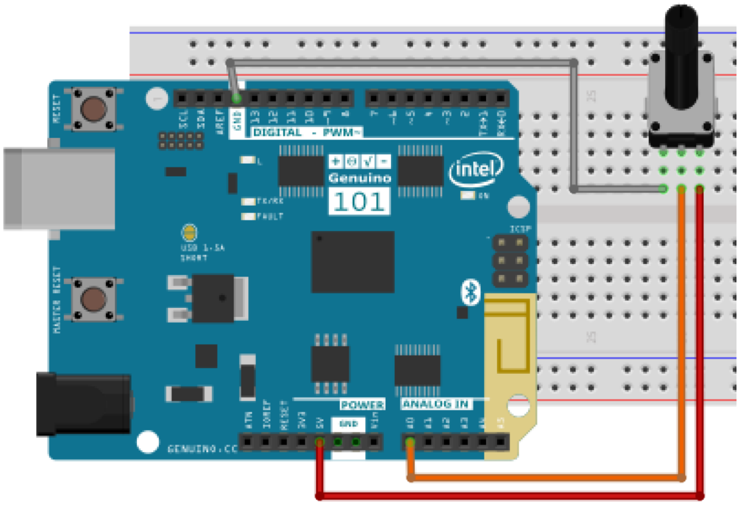
A potentiometer is a resistor. The resistance varies when its shaft is rotated. It's sometimes used as a speaker volume knob.

It is a common component used to represent the analog pin status of the MCU boards. You have to connect the middle pin of potentiometer to one of MCU board's analog pin. And the other two pins should connect to MCU board's 5V and GND pin. When the shaft is rotated, its status can be detected by MCU board's analog pin.

Note: potentiometer is a non-polarized component, which means the difference to connect potentiometer ***left*** *pin to MCU board's 5V pin (another pin to GND pin) is that when you rotate potentiometer* *shaft clockwise, the pin value will change in opposite way than to connect its **right** pin to MCU board's 5V pin.*



Check figure below to connect a potentiometer to Arduino 101. The middle pin (**orange** wire) connects to Arduino 101's A0 pin, the left pin (**grey** wire) connects to Arduino 101's GND pin and the right pin (**red** wire) connects to Arduino 101's 5V pin.



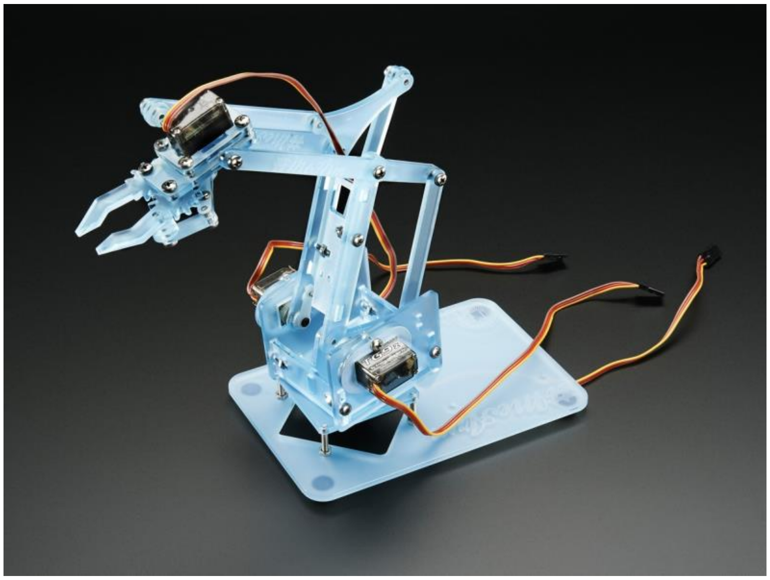
To learn how to control the potentiometer with your Arduino only, please review:

https://www.arduino.cc/en/Tutorial/AnalogReadSerial

Servo (digital output component)

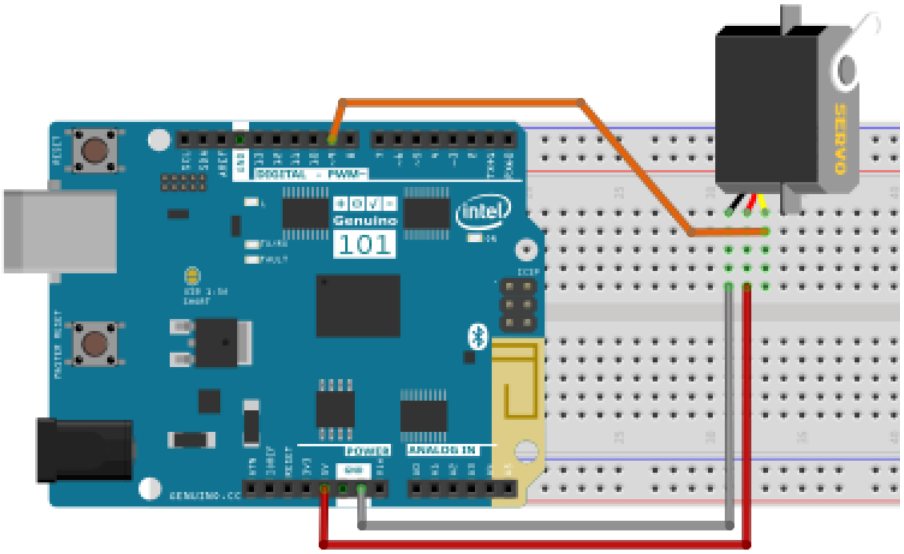
Servo is used to control light-weight mechanism like leverage or sensor mount.

For example, the open source desktop **MeArm** is composed of four servos as a joint.



(Image from https://www.adafruit.com/product/2012)

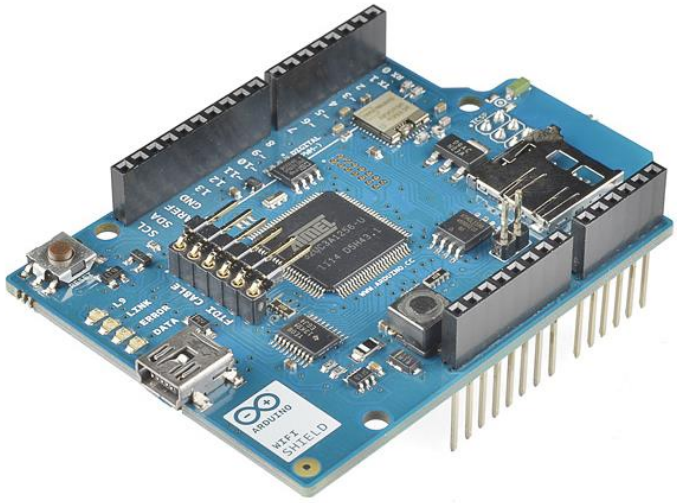
Check the figure below to learn how to connect a servo to your Arduino 101, with its signal pin (**orange** wire) connected to the Arduino 101's D9 pin, its GND pin (**grey** wire) to the Arduino 101's GND pin and its POWER pin (**red** wire) to the Arduino 101's 5V pin.



To control the servo with your Arduino only, please review: https://www.arduino.cc/en/Tutorial/Sweep

Extension boards

Extension boards (or shields) are add-on boards for MCU boards to get additional function, such as Ethernet, Wi-Fi, motor control, GPS, LCD, etc.

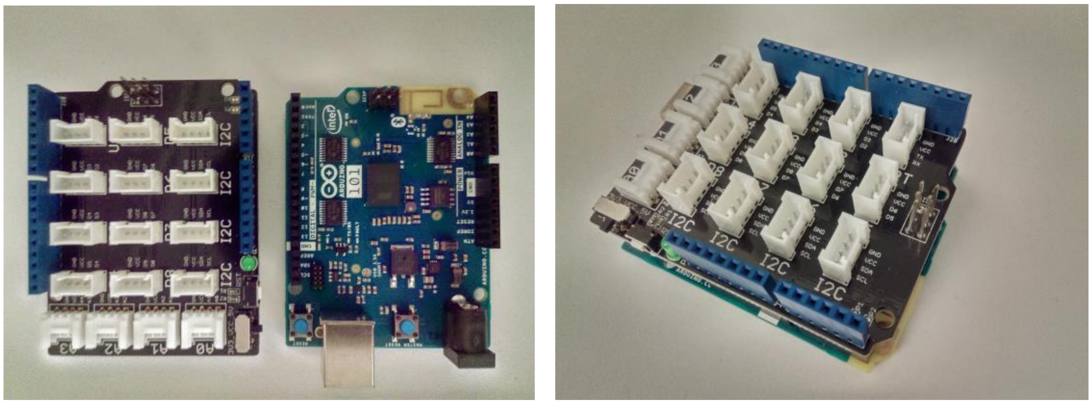


Arduino Wi-Fi shield (Image from https://www.arduino.cc/en/Guide/ArduinoWiFiShield)

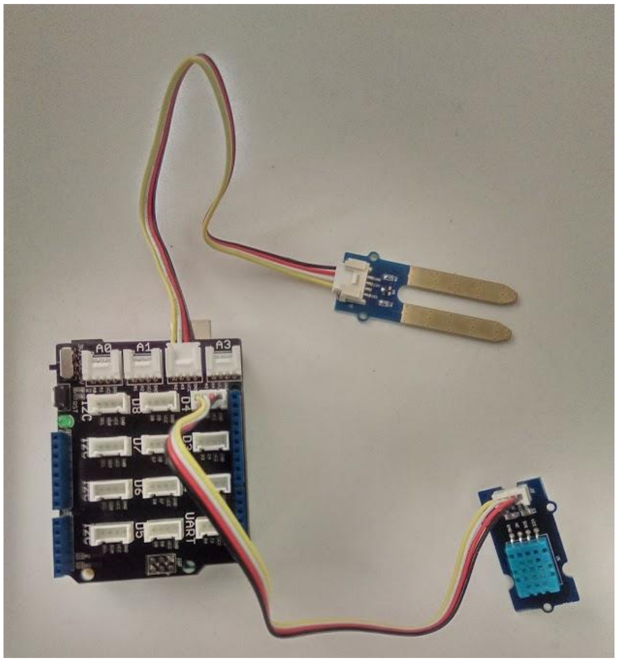
The benefit of using **Extension board** is to save wiring work. These boards also usually have mechanisms to protect your MCU board from accidental damage like a surge.

- Adafruit 2.8" TFT Touch Shield for Arduino with Resistive Touch Screen
- Adafruit 16-Channel 12-bit PWM/Servo Shield

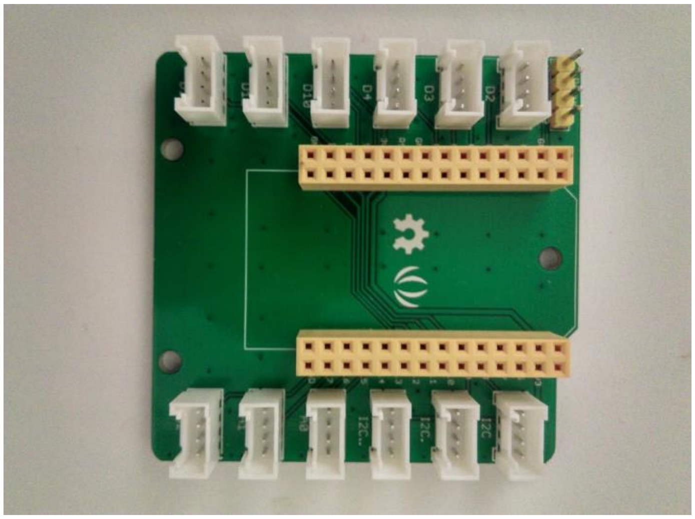
Arduino 101 and Seeed Studio Grove extension board



Connect Grove sensor to Arduino 101 with extension board.



LinkIt 7697’s Grove extension board



Starter kit

A starter kit is a set of peripherals for a specific MCS board, including extension board, sensors and actuators. The benefit of using **Starter kit** is to save work wiring. Also, all the components in the kit are guaranteed by the manufacturer to work normally. Although the starter kit is a bit expensive than separate components, it is very suitable for beginner and educational workshops.

- DFRobot Gravity: Starter Kit for Arduino 101
- Seeed Studio Grove Starter Kit for LinkIt 7697

Resources

- Arduino.cc
- Fritzing
- Circuits on Tinkercad