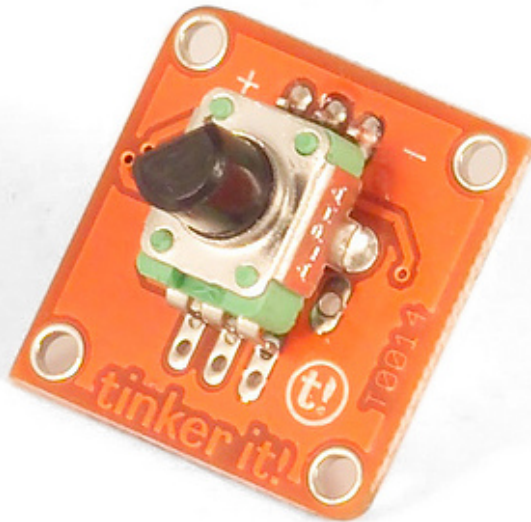




TinkerKit!

## TinkerKit Rotary Potentiometer



### Overview

A **Potentiometer** is a commonly used variable resistor. Turning the knob, you vary the output voltage between 0 and 5V. This value is sent through the middle pin of the pot.

**Output:** This module outputs 5v when turned in one direction, and 0v when turned in the opposite way. When connected to an input on the Arduino using the TinkerKit Shield, you can expect to read values between 0 and 1023.

**Module Description:** This module features a 4k7 Ohm linear potentiometer, a signal amplifier, the standard TinkerKit 3pin connector, a green LED that signals that the module is correctly powered and a yellow LED whose brightness changes according to the position of the potentiometer.

This module is a **SENSOR**. The connector is an **OUTPUT** which must be connected to one of the **INPUT** connectors on the **TinkerKit Shield**.

## Code Example

```
/*
Analog input, analog output, serial output

Reads an analog input pin, and T000140 Rotary Potentiometer Analog Sensor
connected to I0, maps the result to a range from 0 to 255
and uses the result to set the pulsewidth modulation (PWM) on a T010111
LED Module connected on O0.
Also prints the results to the serial monitor.

created 29 Dec. 2008
Modified 4 Sep 2010
by Tom Igoe
modified 7 dec 2010
by Davide Gomba

This example code is in the public domain.

*/

#define O0 11
#define O1 10
#define O2 9
#define O3 6
#define O4 5
#define O5 3
#define I0 A0
#define I1 A1
#define I2 A2
#define I3 A3
#define I4 A4
#define I5 A5

// These constants won't change. They're used to give names
// to the pins used:
const int analogInPin = I0; // Analog input pin that the Pot is attached
to
const int analogOutPin= O0; // Analog output pin that the LED is attached
to

int sensorValue = 0; // value read from the pot
int outputValue = 0; // value output to the PWM (analog out)

void setup() {
// initialize serial communications at 9600 bps:
Serial.begin(9600);
}

void loop() {
// read the analog in value:
sensorValue = analogRead(analogInPin);
```

```
// map it to the range of the analog out:
outputValue = map(sensorValue, 0, 1023, 0, 255);
// change the analog out value:
analogWrite(analogOutPin, outputValue);

// print the results to the serial monitor:
Serial.print("sensor = " );
Serial.print(sensorValue);
Serial.print("\t output = ");
Serial.println(outputValue);

// wait 10 milliseconds before the next loop
// for the analog-to-digital converter to settle
// after the last reading:
delay(10);
}
```