/\*

AnalogReadSerial

Reads an analog input on pin 0, prints the result to the serial monitor.

Graphical representation is available using serial plotter (Tools > Serial Plotter menu)

Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.

This example code is in the public domain.

\*/

// the setup routine runs once when you press reset:

**void setup() {**

// initialize serial communication at 9600 bits per second:

**Serial.begin(9600);**

**}**

// the loop routine runs over and over again forever:

**void loop() {**

// read the input on analog pin 0:

**int sensorValue = analogRead(A0);**

// print out the value you read:

**Serial.println(sensorValue);**

**delay(1); // delay in between reads for stability**

**}**

/\*

DigitalReadSerial

Reads a digital input on pin 2, prints the result to the serial monitor

This example code is in the public domain.

\*/

// digital pin 2 has a pushbutton attached to it. Give it a name:

**int pushButton = 2;**

// the setup routine runs once when you press reset:

**void setup() {**

// initialize serial communication at 9600 bits per second:

**Serial.begin(9600);**

// make the pushbutton's pin an input:

**pinMode(pushButton, INPUT);**

**}**

// the loop routine runs over and over again forever:

**void loop() {**

// read the input pin:

**int buttonState = digitalRead(pushButton);**

**// print out the state of the button:**

**Serial.println(buttonState);**

**delay(1); // delay in between reads for stability**

**}**

/\*

Blink

Turns on an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO

it is attached to digital pin 13, on MKR1000 on pin 6. LED\_BUILTIN is set to

the correct LED pin independent of which board is used.

If you want to know what pin the on-board LED is connected to on your Arduino model, check

the Technical Specs of your board at https://www.arduino.cc/en/Main/Products

This example code is in the public domain.

modified 8 May 2014

by Scott Fitzgerald

modified 2 Sep 2016

by Arturo Guadalupi

modified 8 Sep 2016

by Colby Newman

\*/

// the setup function runs once when you press reset or power the board

**void setup() {**

// initialize digital pin LED\_BUILTIN as an output.

**pinMode(LED\_BUILTIN, OUTPUT);**

**}**

// the loop function runs over and over again forever

**void loop() {**

**digitalWrite(LED\_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)**

**delay(1000); // wait for a second**

**digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltage LOW**

**delay(1000); // wait for a second**

**}**

/\*

Button

Turns on and off a light emitting diode(LED) connected to digital

pin 13, when pressing a pushbutton attached to pin 2.

The circuit:

\* LED attached from pin 13 to ground

\* pushbutton attached to pin 2 from +5V

\* 10K resistor attached to pin 2 from ground

\* Note: on most Arduinos there is already an LED on the board

attached to pin 13.

created 2005

by DojoDave <http://www.0j0.org>

modified 30 Aug 2011

by Tom Igoe

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Button

\*/

// constants won't change. They're used here to

// set pin numbers:

**const int buttonPin = 2; // the number of the pushbutton pin**

**const int ledPin = 13; // the number of the LED pin**

// variables will change:

**int buttonState = 0; // variable for reading the pushbutton status**

**void setup() {**

// initialize the LED pin as an output:

**pinMode(ledPin, OUTPUT);**

// initialize the pushbutton pin as an input:

**pinMode(buttonPin, INPUT);**

**}**

**void loop() {**

// read the state of the pushbutton value:

**buttonState = digitalRead(buttonPin);**

// check if the pushbutton is pressed.

// if it is, the buttonState is HIGH:

**if (buttonState == HIGH) {**

// turn LED on:

**digitalWrite(ledPin, HIGH);**

**} else {**

// turn LED off:

**digitalWrite(ledPin, LOW);**

**}**

**}**