/\*

State change detection (edge detection)

Often, you don't need to know the state of a digital input all the time,

but you just need to know when the input changes from one state to another.

For example, you want to know when a button goes from OFF to ON. This is called

state change detection, or edge detection.

created 27 Sep 2005

modified 30 Aug 2011

by Tom Igoe

modified 27 Apr 2012

Robert Wessels

This example code is in the public domain.

\*/

const int buttonPin = PUSH2; // the pin that the pushbutton is attached to

const int ledPin = RED\_LED; // the pin that the LED is attached to

// Variables will change:

int buttonPushCounter = 0; // counter for the number of button presses

int buttonState = 0; // current state of the button

int lastButtonState = 0; // previous state of the button

void setup() {

// initialize the button pin as a input:

pinMode(buttonPin, INPUT\_PULLUP);

// initialize the LED as an output:

pinMode(ledPin, OUTPUT);

// initialize serial communication:

Serial.begin(9600);

}

void loop() {

// read the pushbutton input pin:

buttonState = digitalRead(buttonPin);

// compare the buttonState to its previous state

if (buttonState != lastButtonState) {

// if the state has changed, increment the counter

if (buttonState == HIGH) {

// if the current state is HIGH then the button

// wend from off to on:

buttonPushCounter++;

Serial.println("on");

Serial.print("number of button pushes: ");

Serial.println(buttonPushCounter);

}

else {

// if the current state is LOW then the button

// wend from on to off:

Serial.println("off");

}

}

// save the current state as the last state,

//for next time through the loop

lastButtonState = buttonState;

// turns on the LED every four button pushes by

// checking the modulo of the button push counter.

// the modulo function gives you the remainder of

// the division of two numbers:

if (buttonPushCounter % 4 == 0) {

digitalWrite(ledPin, HIGH);

} else {

digitalWrite(ledPin, LOW);

}

}