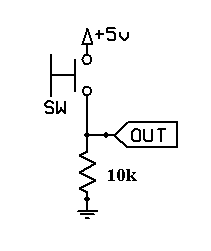
DeBounce Circuit:

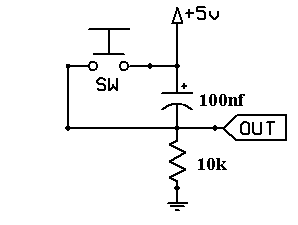
***Software Debouncing:***



CODE:

const int buttonPin = 2;     // the number of the pushbutton pin  
const int ledPin =  13;      // the number of the LED pin  
  
// Variables will change:  
int ledState = HIGH;         // the current state of the output pin  
int buttonState;             // the current reading from the input pin  
int lastButtonState = LOW;   // the previous reading from the input pin  
  
// the following variables are long's because the time, measured in miliseconds,  
// will quickly become a bigger number than can be stored in an int.  
long lastDebounceTime = 0;  // the last time the output pin was toggled  
long debounceDelay = 50;    // the debounce time; increase if the output flickers  
  
void setup() {  
  pinMode(buttonPin, INPUT);  
  pinMode(ledPin, OUTPUT);  
}  
  
void loop() {  
  // read the state of the switch into a local variable:  
  int reading = digitalRead(buttonPin);  
  
  // check to see if you just pressed the button   
  // (i.e. the input went from LOW to HIGH),  and you've waited   
  // long enough since the last press to ignore any noise:    
  
  // If the switch changed, due to noise or pressing:  
  if (reading != lastButtonState) {  
    // reset the debouncing timer  
    lastDebounceTime = millis();  
  }   
    
  if ((millis() - lastDebounceTime) > debounceDelay) {  
    // whatever the reading is at, it's been there for longer  
    // than the debounce delay, so take it as the actual current state:  
    buttonState = reading;  
  }  
    
  // set the LED using the state of the button:  
  digitalWrite(ledPin, buttonState);  
  
  // save the reading.  Next time through the loop,  
  // it'll be the lastButtonState:  
  lastButtonState = reading;  
}

Circuit debouncing:



Code:

#define buttonPin  2    // the number of the pushbutton pin  
#define ledPin  13      // the number of the LED pin  
  
void setup() {  
  pinMode(buttonPin, INPUT);  
  pinMode(ledPin, OUTPUT);  
}  
  
void loop() {  
   digitalWrite(ledPin, digitalRead(buttonPin));  //sets the LED to current state of button  
}