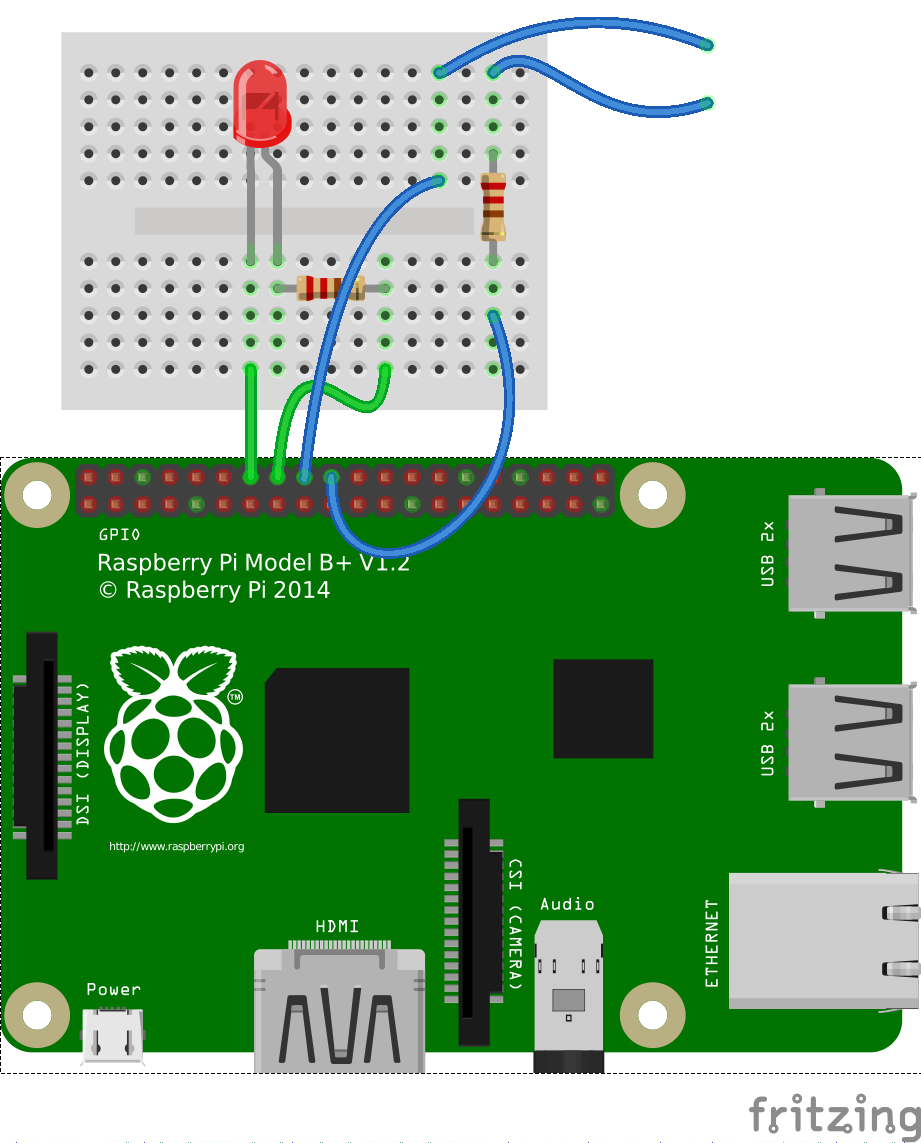
Vypinaní LEDky a podle spinace

BREADBOARD

http://sourceforge.net/p/raspberry-gpio-python/wiki/Examples/

http://raspi.tv/2013/how-to-use-interrupts-with-python-on-the-raspberry-pi-and-rpi-gpio-part-3

http://makezine.com/projects/tutorial-raspberry-pi-gpio-pins-and-python/



RASPI

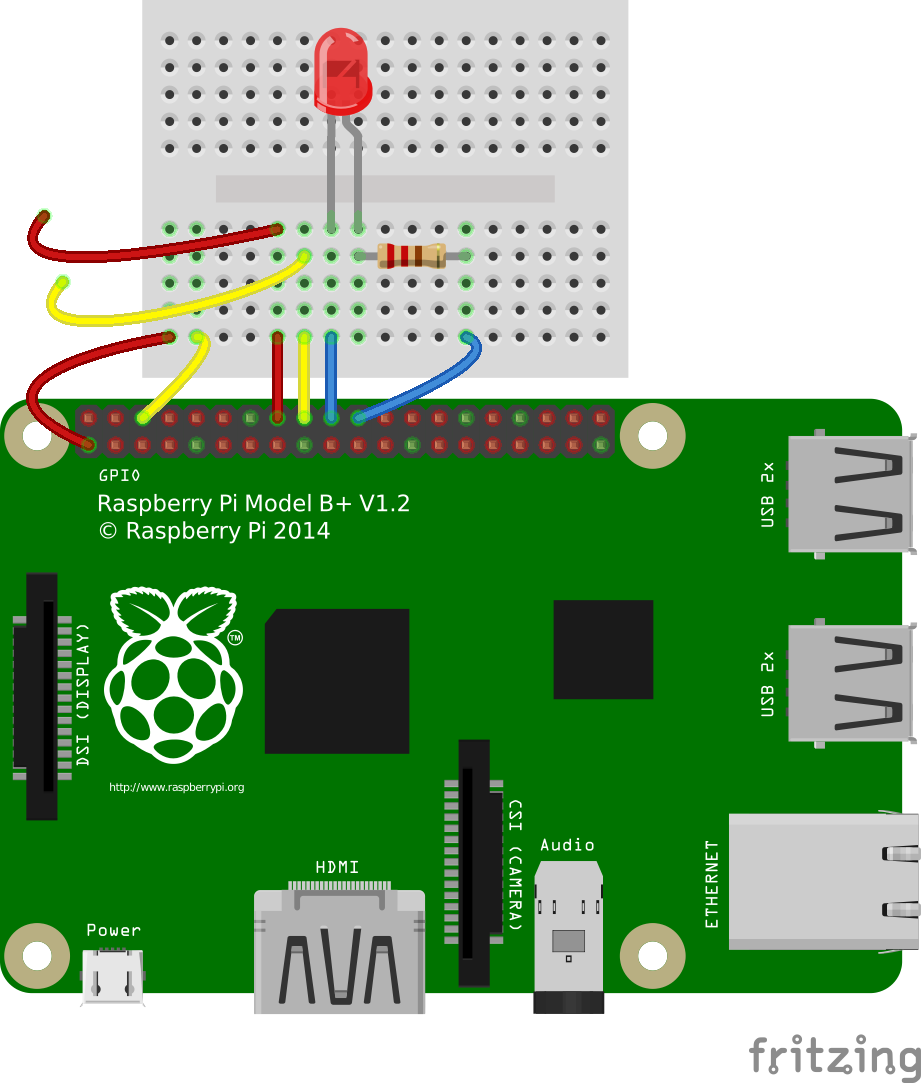
## SPINAC.PY

**import RPi.GPIO as GPIO ## Import GPIO library  
import time  
  
GPIO.setmode(GPIO.BOARD) ## Use board pin numbering  
GPIO.setup(18, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)  
GPIO.setup(16, GPIO.OUT) ## Setup GPIO Pin 16 to OUT  
  
while 1:  
 if GPIO.input(18):  
 print "Nic se nedeje"  
 GPIO.output(16, False)  
 time.sleep(0.2)  
 else:  
 # When the button switch is not pressed, turn off the LED.  
 print "Nekdo prisel, svitime!"  
 GPIO.output( 16, True)  
 time.sleep(0.2)**

Ale co víc spínačů?

(žlutej na žlutej!!!! červenej na červenej!!!!)

**SPINAC2.PY**



**import RPi.GPIO as GPIO ## Import GPIO library  
import time  
  
  
GPIO.setmode(GPIO.BOARD) ## Use board pin numbering  
GPIO.setup(16, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)  
#PUD\_DOWN = na 3.3V  
GPIO.setup(18, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)  
#PU\_UP = na zem  
GPIO.setup(22, GPIO.OUT) ## Setup GPIO Pin 22 to OUT  
  
while 1:  
 if (GPIO.input(16) == 0):  
 print "Nic se nedeje"  
 GPIO.output(22, False)  
 time.sleep(0.2)  
 else:  
 # When the button switch is not pressed, turn off the LED.  
 print "Nekdo prisel, svitime!"  
 GPIO.output(22, True)  
 time.sleep(0.2)  
 if (GPIO.input(18) == 0):  
 print "Osmnactka jede"  
 time.sleep(0.2)**

Ale jak na to, abych hledal změnu stavu? *Ale čekám na to, až se mi - jedna podmínka za druhou splní…. fuj*

**SPINAC3.PY**

**import RPi.GPIO as GPIO ## Import GPIO library  
import time  
  
GPIO.setmode(GPIO.BOARD) ## Use board pin numbering  
GPIO.setup(18, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)  
GPIO.setup(16, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)  
  
while 1:  
 GPIO.wait\_for\_edge(16, GPIO.RISING) # cekam a cekam….   
 print("Button 1 Pressed")  
 GPIO.wait\_for\_edge(16, GPIO.FALLING)  
 print("Button 1 Released")  
 GPIO.wait\_for\_edge(18, GPIO.FALLING)  
 print("Button 2 Pressed")  
 GPIO.wait\_for\_edge(18, GPIO.RISING)  
 print("Button 2 Released")  
  
GPIO.cleanup()**

Ale teď už fakt raději správně:

použijeme callback a bounce time 300ms?

**SPINAC4.PY**

**import RPi.GPIO as GPIO  
  
GPIO.setmode(GPIO.BOARD)  
GPIO.setup(16, GPIO.IN, pull\_up\_down = GPIO.PUD\_DOWN)  
GPIO.setup(18, GPIO.IN, pull\_up\_down = GPIO.PUD\_UP)  
  
def printFunction(channel):  
 print("Button 1 pressed")  
 print("Note how the bouncetime affects the button press")  
  
GPIO.add\_event\_detect(16, GPIO.RISING, callback=printFunction, bouncetime=300)  
  
while True:  
  
 GPIO.wait\_for\_edge(18, GPIO.FALLING)  
 print("Button 2 Pressed")  
 GPIO.wait\_for\_edge(18, GPIO.RISING)  
 print("Button 2 Released")  
  
  
GPIO.cleanup()**

takže přístě:

**SPINAC5.PY**

**import RPi.GPIO as GPIO  
import time  
  
GPIO.setmode(GPIO.BOARD)  
GPIO.setup(16, GPIO.IN, pull\_up\_down = GPIO.PUD\_DOWN)  
GPIO.setup(18, GPIO.IN, pull\_up\_down = GPIO.PUD\_UP)  
  
def sestnactka(channel):  
 print("Sestnactka - Button "), channel  
  
def osmnactka(channel):  
 print("Osmnactka - Button "), channel   
  
GPIO.add\_event\_detect(16, GPIO.RISING, callback=sestnactka, bouncetime=300)  
GPIO.add\_event\_detect(18, GPIO.FALLING, callback=osmnactka, bouncetime=300)  
  
while True:  
  
 print time.ctime()  
 time.sleep(2)  
  
GPIO.cleanup()**