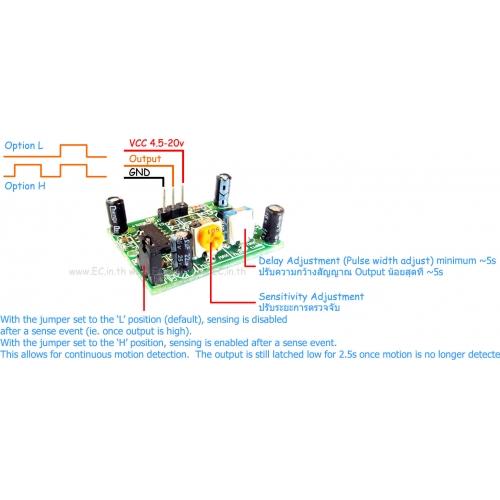
PIR Cidlo



## 

## Delay nastvit na minimum

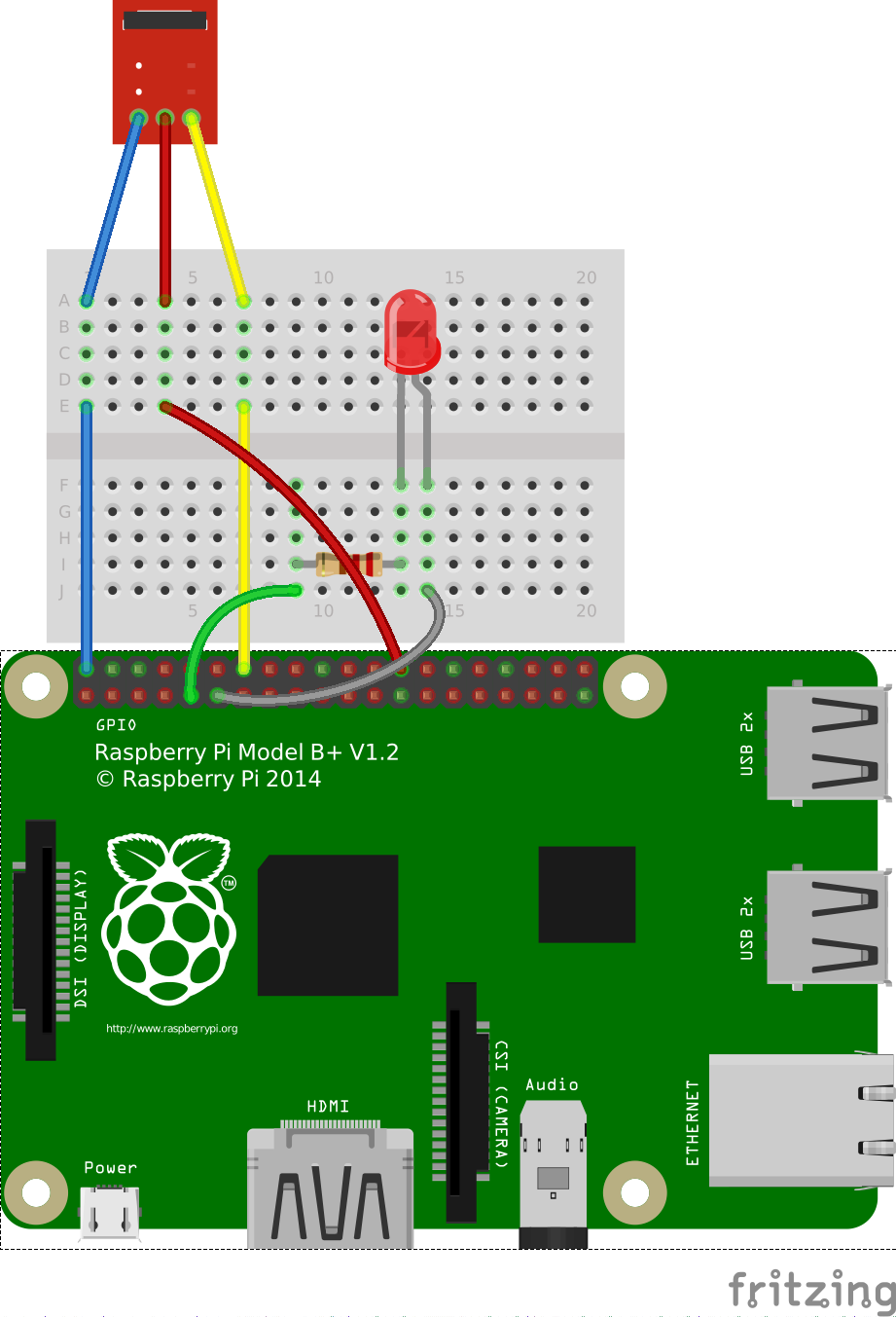
## Citlivost na polovic

http://www.raspberrypi-spy.co.uk/2013/02/cheap-pir-sensors-and-the-raspberry-pi-part-2/

http://www.cmprogrammers.com/post.php?data=2014-08-05+20:38:00

wget http://www.cmprogrammers.com/projects/documents/alarm.py

wget http://www.cmprogrammers.com/projects/documents/uz.sh



**import RPi.GPIO as GPIO  
import time  
   
# Use BCM GPIO references  
# instead of physical pin numbers  
GPIO.setmode(GPIO.BCM)  
# Define GPIO to use on Pi  
GPIO\_PIR = 7  
GPIO\_LED = 17   
   
print "PIR Module Test (CTRL-C to exit)"  
# Set pin as input  
GPIO.setup(GPIO\_PIR,GPIO.IN) # Echo  
GPIO.setup(GPIO\_LED,GPIO.OUT)  
   
Current\_State = 0  
Previous\_State = 0  
   
GPIO.output(GPIO\_LED,True)   
time.sleep(2)  
GPIO.output(GPIO\_LED,False)   
try:  
 print "Waiting for PIR to settle ..."  
 # Loop until PIR output is 0  
 while GPIO.input(GPIO\_PIR)==1:  
 Current\_State = 0  
   
 print " Ready"  
 # Loop until users quits with CTRL-C  
 while True :  
 # Read PIR state  
 Current\_State = GPIO.input(GPIO\_PIR)  
   
 if Current\_State==1 and Previous\_State==0:  
 # PIR is triggered  
 print " Motion detected!"  
 print time.ctime()  
 GPIO.output(GPIO\_LED,True)   
 # Record previous state  
 Previous\_State=1  
 elif Current\_State==0 and Previous\_State==1:  
 # PIR has returned to ready state  
 print " Ready"  
 GPIO.output(GPIO\_LED,False)   
 Previous\_State=0  
   
 # Wait for 20 milliseconds  
 time.sleep(0.02)  
except KeyboardInterrupt:  
 print " Quit. Cleaning up GPIOs"  
 GPIO.cleanup()**