Kevin Sass

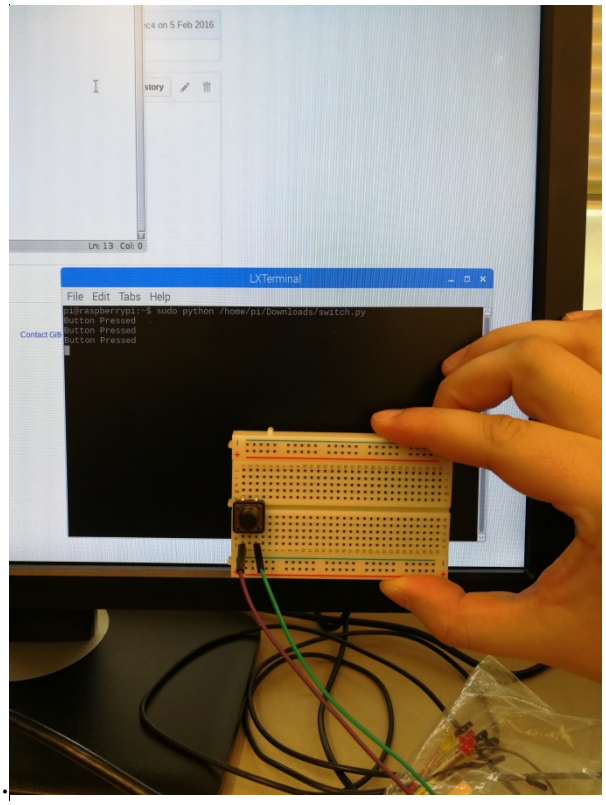
Corey Zheng

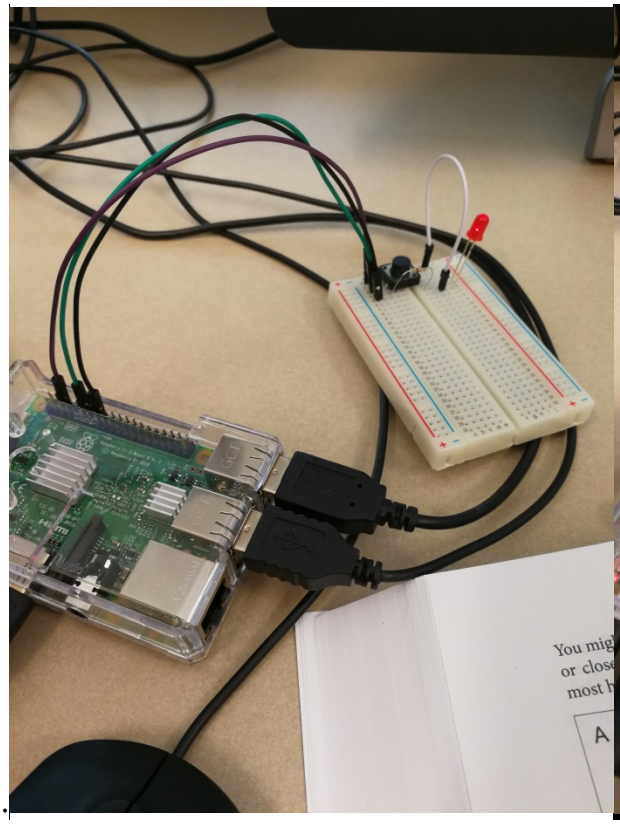
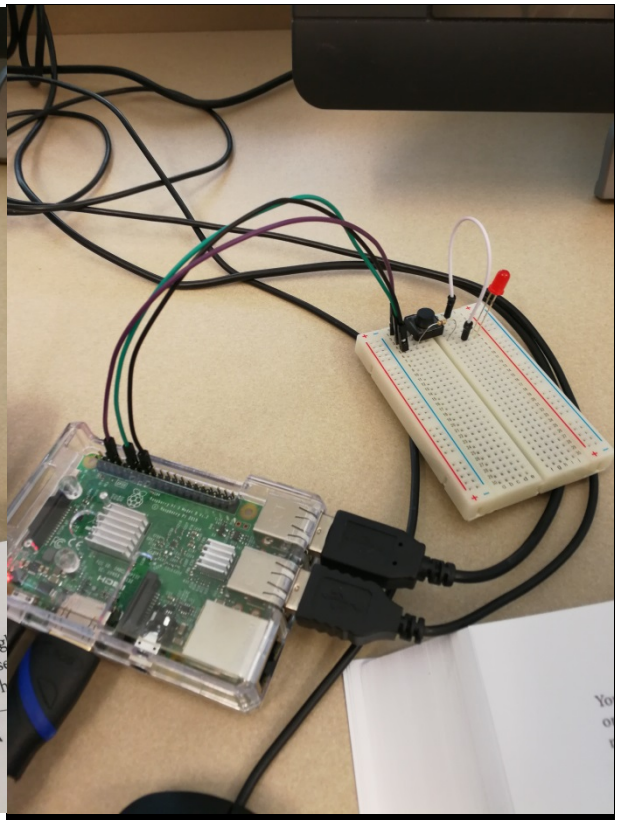
CSC 299 Lab_Week 3 Report

4/12/2018

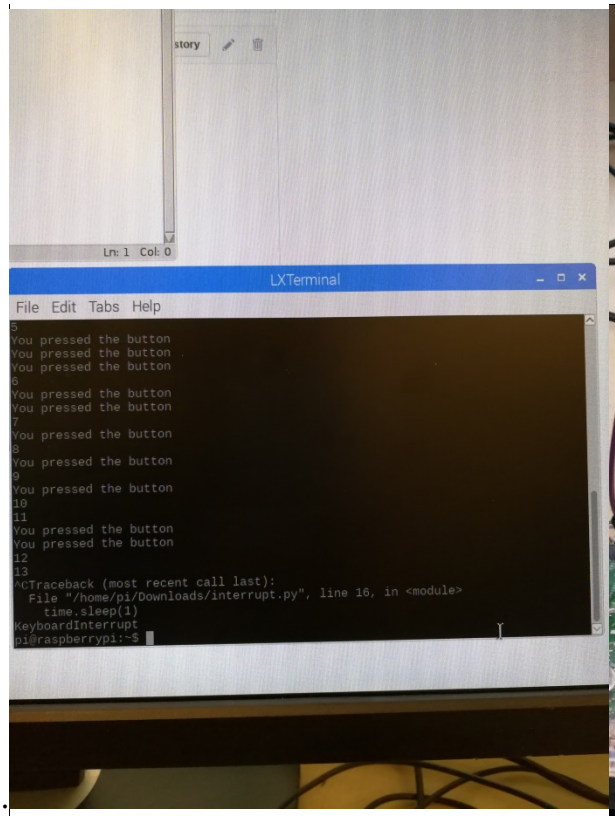
In this lab we learned a lot of different new things that we can do with the Pi. We learned how to install a motion sensor to detect motion, and can even send an email to someone once this happens (good home security/burglary prevention application). We learned how to track mouse movement and display the x and y coordinates of where it is on the screen. We learned how to intercept keys pressed and display their ASCII character to the screen. We learned how to attach a camera to the Pi and take a picture or video, and even send it to an email once motion is detected, an even better improvement on the anti-burglary application. We learned how to properly wire the circuit board to be able to control an LED with a push button switch and output a message to the screen saying it was pressed. We learned how to implement a pull-up resistor design as well as switch it to a pull-down resistor circuit. We programmed an interrupt into the button switch so that there is a 0.2s delay after which the button can be activated again. We implemented a rotary encoder that keeps track of the number the dial is turned to, and outputs it to screen.

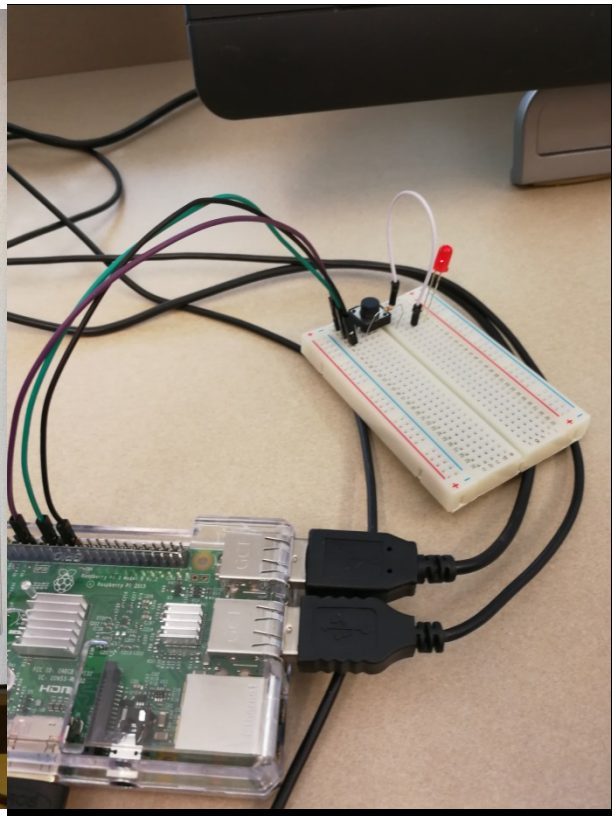
We did have some challenges. When we first set up our motion sensor, it was picking up motion when nothing was moving. We solved it by replacing it with another motion sensor. It turned out the first one was faulty. We were also able to get the motion sensor camera working, but it was only able to take one picture before an error occurred:
“picamera.exc.PiCameraMMALError: Failed to enable connection: Out of resources”.

1. 
LXTerminal
File Edit Tabs Help
pi@raspberrypi:~$ sudo python /home/pi/Downloads/switch.py
Button Pressed
Button Pressed
Button Pressed

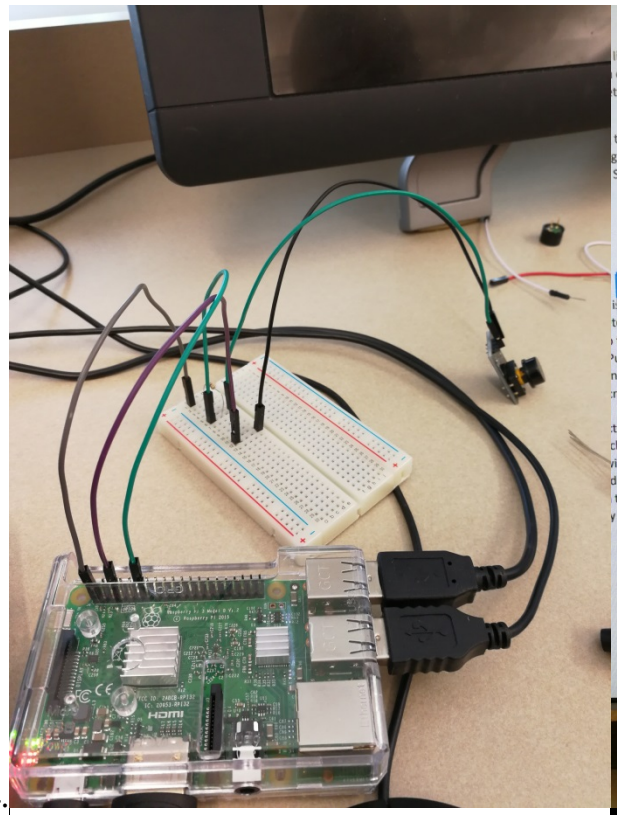
2. 


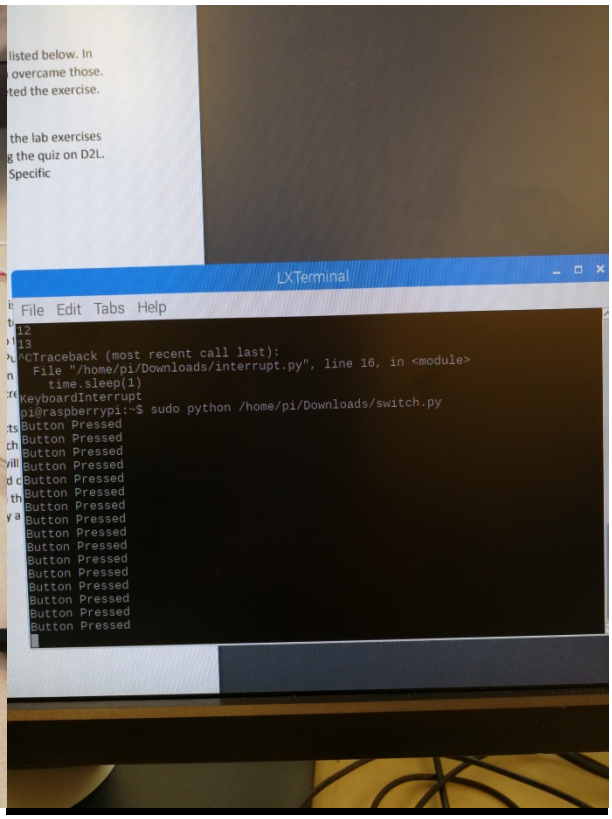
3.

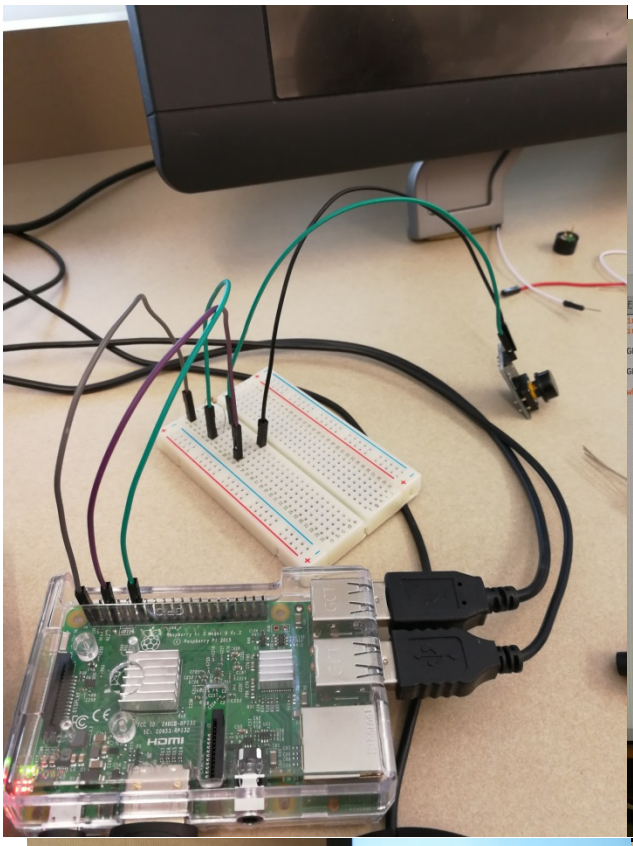

story
Ln: 1 Col: 0
LXTerminal
File Edit Tabs Help
You pressed the button
You pressed the button
You pressed the button
You pressed the button
You pressed the button
You pressed the button
You pressed the button
You pressed the button
You pressed the button
^CTraceback (most recent call last):
File "/home/pi/Downloads/interrupt.py", line 16, in <module>
time.sleep(1)
KeyboardInterrupt
pi@raspberrypi:~$

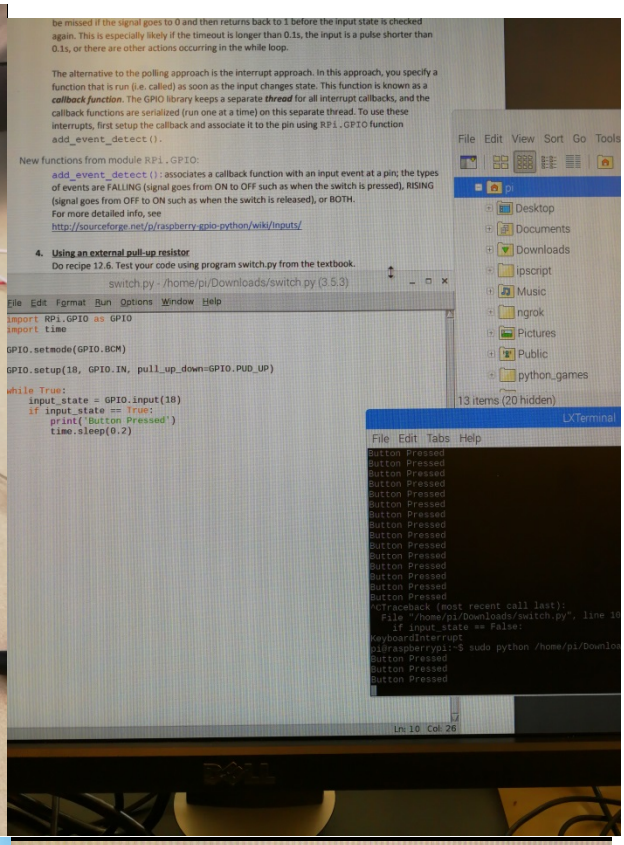


4.

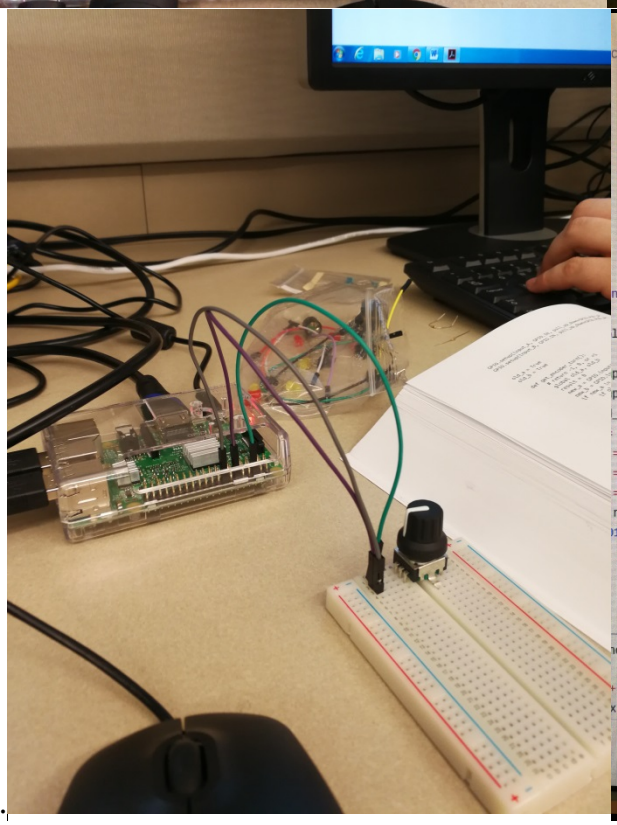


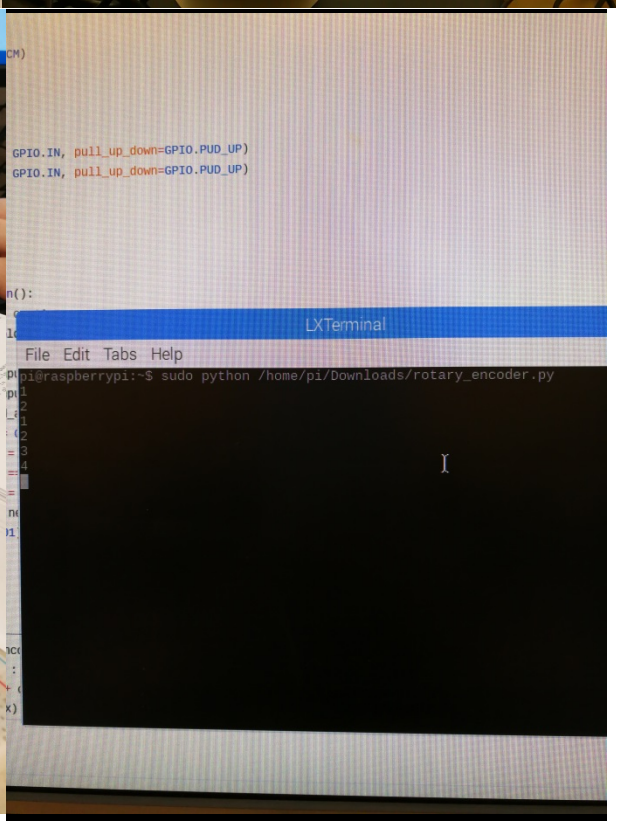

LXTerminal
File Edit Tabs Help
12
13
^CTraceback (most recent call last):
File "/home/pi/Downloads/interrupt.py", line 16, in <module>
time.sleep(1)
KeyboardInterrupt
pi@raspberrypi:~$ sudo python /home/pi/Downloads/switch.py
Button Pressed
Button Pressed
Button Pressed
Button Pressed
Button Pressed
Button Pressed
Button Pressed
Button Pressed
Button Pressed
Button Pressed
Button Pressed
Button Pressed
Button Pressed
Button Pressed
Button Pressed
Button Pressed



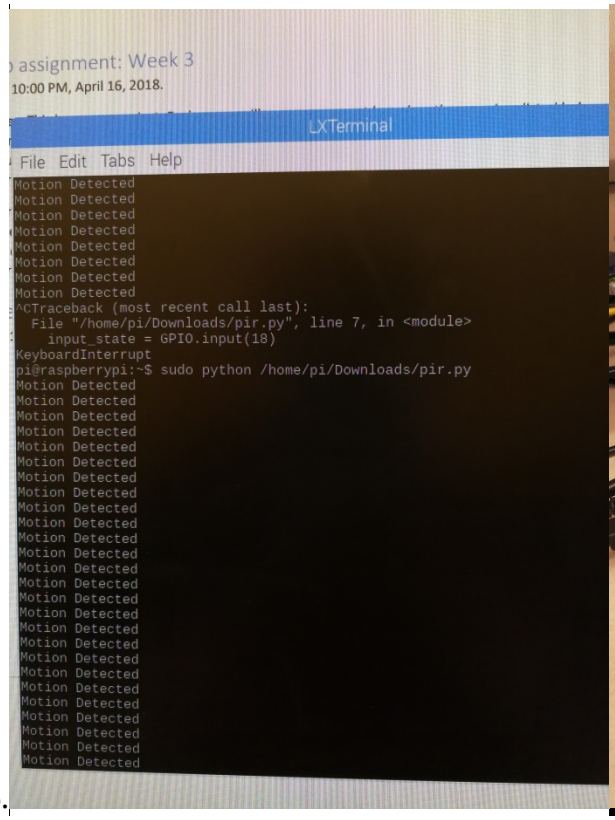

be missed if the signal goes to 0 and then returns back to 1 before the input state is checked again. This is especially likely if the timeout is longer than 0.1s, the input is a pulse shorter than 0.1s, or there are other actions occurring in the while loop.
The alternative to the polling approach is the interrupt approach. In this approach, you specify a function that is run (i.e. called) as soon as the input changes state. This function is known as a callback function. The GPIO library keeps a separate thread for all interrupt callbacks, and the callback functions are serialized (run one at a time) on this separate thread. To use these interrupts, first setup the callback and associate it to the pin using RPi.GPIO function add_event_detect().
New functions from module RPi.GPIO:
add_event_detect(): associates a callback function with an input event at a pin; the types of events are FALLING (signal goes from ON to OFF such as when the switch is pressed), RISING (signal goes from OFF to ON such as when the switch is released), or BOTH.
For more detailed info, see
http://sourceforge.net/p/raspberry-gpio-python/wiki/Inputs/
4. Using an external pull-up resistor
Do recipe 12.6. Test your code using program switch.py from the textbook.
switch.py - /home/pi/Downloads/switch.py (3.5.3)
File Edit Format Run Options Window Help
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.IN, pull_up_down=GPIO.PUD_UP)
while True:
input_state = GPIO.input(18)
if input_state == True:
print('Button Pressed')
time.sleep(0.2)
File Edit View Sort Go Tools
pi
Desktop
Documents
Downloads
ipscript
Music
ngrok
Pictures
Public
python_games
13 items (20 hidden)
LXTerminal
File Edit Tabs Help
Button Pressed
Traceback (most recent call last):
File "/home/pi/Downloads/switch.py", line 10
if input_state == False:
KeyboardInterrupt
pi@raspberrypi:~$ sudo python /home/pi/Downloa
Ln: 10 Col: 26

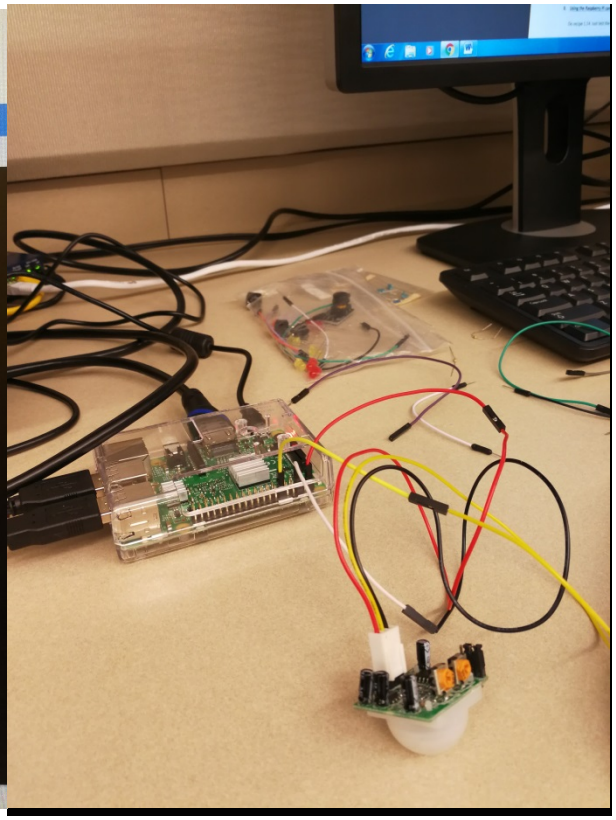
5.



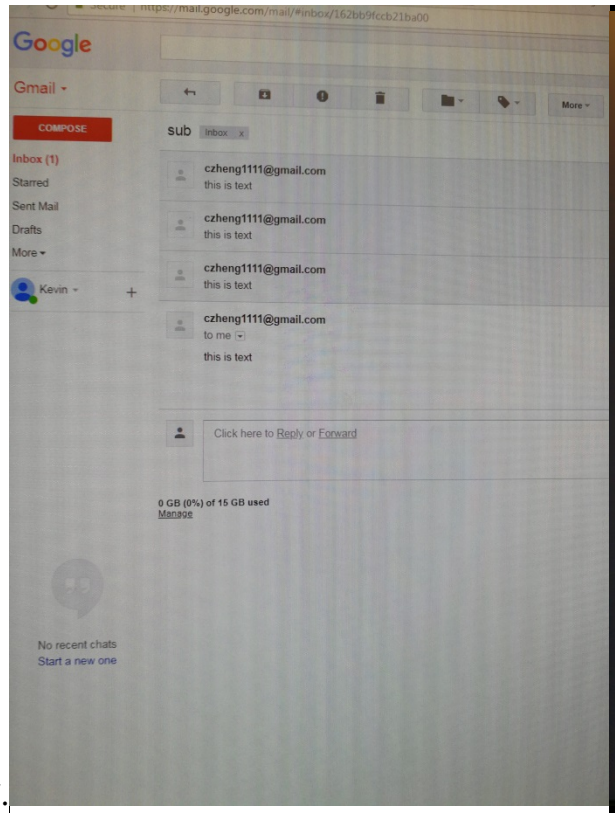

GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.IN, pull_up_down=GPIO.PUD_UP)
LXTerminal
File Edit Tabs Help
pi@raspberrypi:~$ sudo python /home/pi/Downloads/rotary_encoder.py

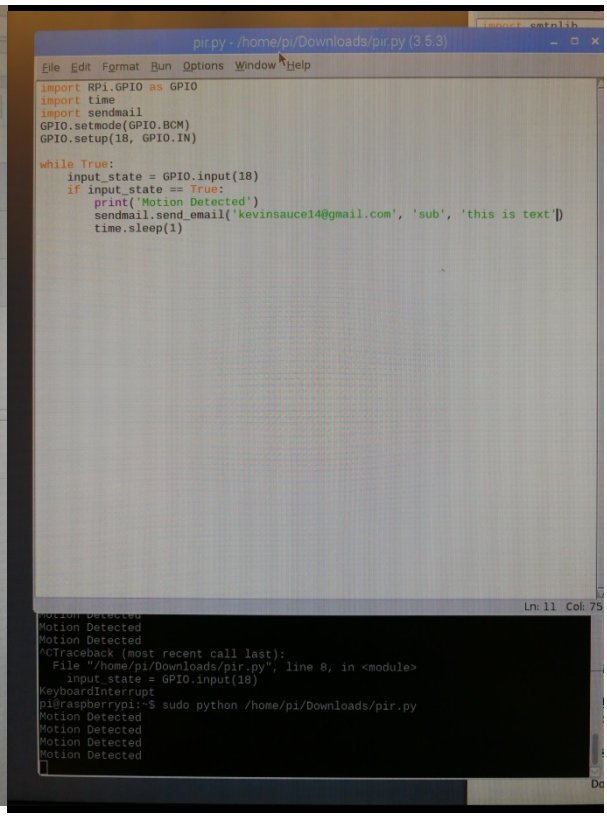
6.


assignment: Week 3
10:00 PM, April 16, 2018.
LXTerminal
File Edit Tabs Help
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
^CTraceback (most recent call last):
File "/home/pi/Downloads/pir.py", line 7, in <module>
input_state = GPIO.input(18)
KeyboardInterrupt
pi@raspberrypi:~$ sudo python /home/pi/Downloads/pir.py
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected
Motion Detected

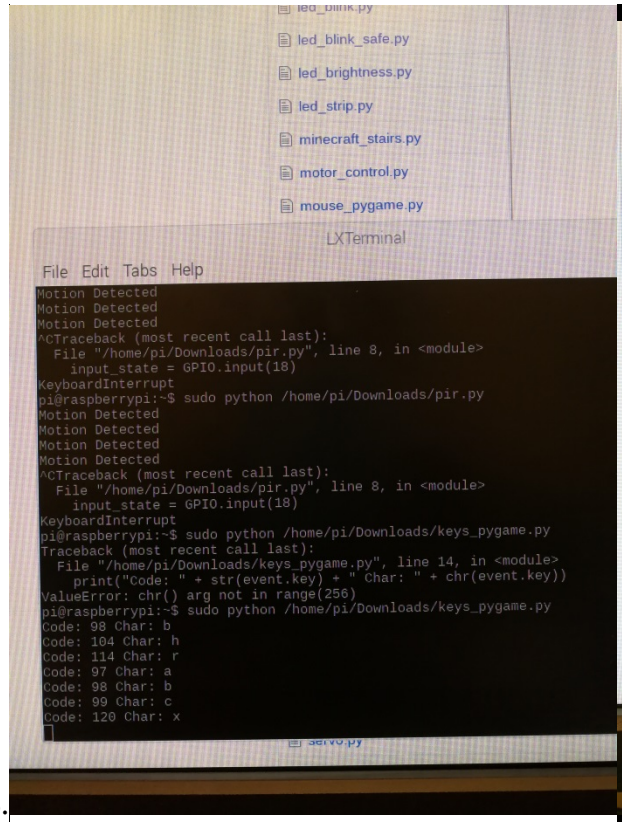


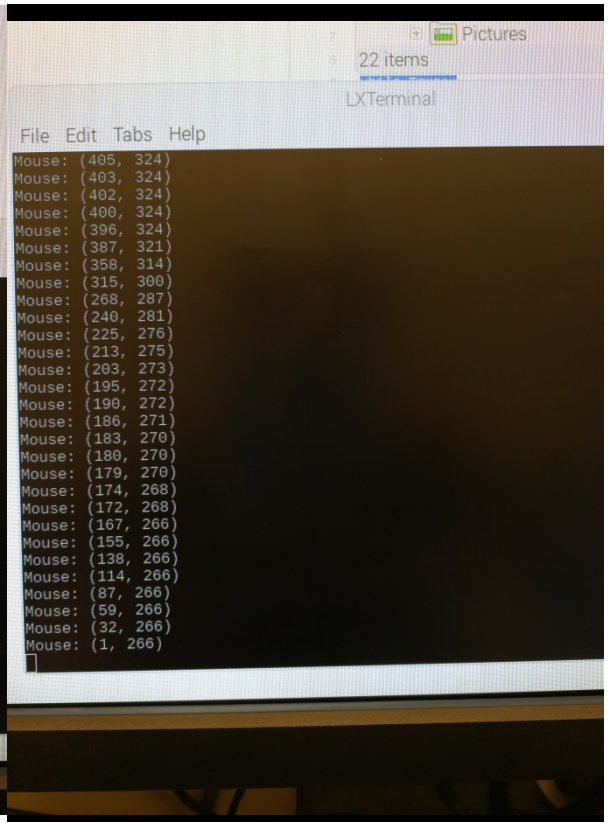
7.


Google
Gmail
COMPOSE
Inbox (1)
Starred
Sent Mail
Drafts
More
Kevin
sub
Inbox x
czheng1111@gmail.com
this is text
czheng1111@gmail.com
this is text
czheng1111@gmail.com
this is text
czheng1111@gmail.com
to me
this is text
Click here to Reply or Forward
0 GB (0%) of 15 GB used
Manage
No recent chats
Start a new one

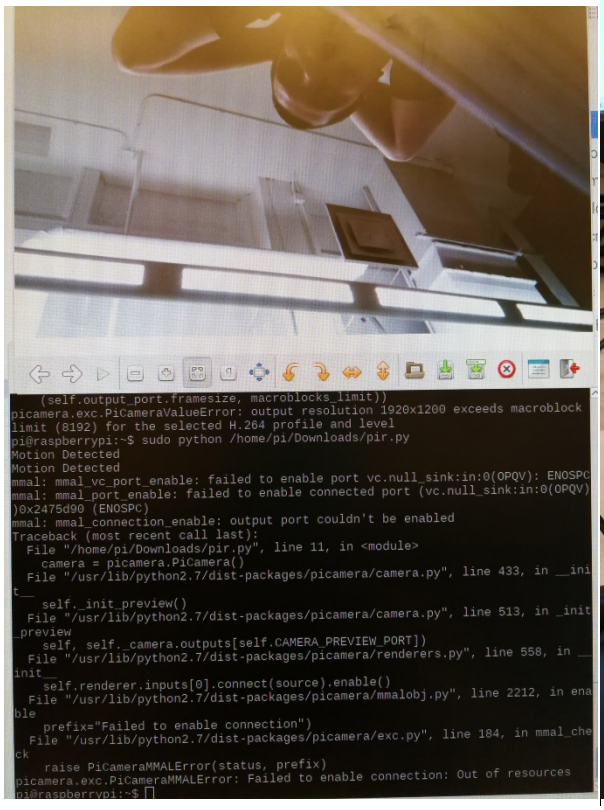

pir.py - /home/pi/Downloads/pir.py (3.5.3)
File Edit Format Run Options Window Help
import RPi.GPIO as GPIO
import time
import sendmail
GPIO.setmode(GPIO.BCM)
GPIO.setup(18, GPIO.IN)
while True:
input_state = GPIO.input(18)
if input_state == True:
print('Motion Detected')
sendmail.send_email('kevinsauce14@gmail.com', 'sub', 'this is text')
time.sleep(1)
Ln: 11 Col: 75
Motion Detected
Motion Detected
^CTraceback (most recent call last):
File "/home/pi/Downloads/pir.py", line 8, in <module>
input_state = GPIO.input(18)
KeyboardInterrupt
pi@raspberrypi:~$ sudo python /home/pi/Downloads/pir.py
Motion Detected
Motion Detected
Motion Detected
Motion Detected

8.


led_blink_safe.py
led_brightness.py
led_strip.py
minecraft_stairs.py
motor_control.py
mouse_pygame.py
LXTerminal
File Edit Tabs Help
Motion Detected
Motion Detected
Motion Detected
^CTraceback (most recent call last):
File "/home/pi/Downloads/pir.py", line 8, in <module>
input_state = GPIO.input(18)
KeyboardInterrupt
pi@raspberrypi:~$ sudo python /home/pi/Downloads/pir.py
Motion Detected
Motion Detected
Motion Detected
Motion Detected
^CTraceback (most recent call last):
File "/home/pi/Downloads/pir.py", line 8, in <module>
input_state = GPIO.input(18)
KeyboardInterrupt
pi@raspberrypi:~$ sudo python /home/pi/Downloads/keys_pygame.py
Traceback (most recent call last):
File "/home/pi/Downloads/keys_pygame.py", line 14, in <module>
print("Code: " + str(event.key) + " Char: " + chr(event.key))
ValueError: chr() arg not in range(256)
pi@raspberrypi:~$ sudo python /home/pi/Downloads/keys_pygame.py
Code: 98 Char: b
Code: 104 Char: h
Code: 114 Char: r
Code: 97 Char: a
Code: 98 Char: b
Code: 99 Char: c
Code: 120 Char: x


Pictures
22 items
LXTerminal
File Edit Tabs Help
Mouse: (405, 324)
Mouse: (403, 324)
Mouse: (402, 324)
Mouse: (400, 324)
Mouse: (396, 324)
Mouse: (387, 321)
Mouse: (358, 314)
Mouse: (315, 300)
Mouse: (268, 287)
Mouse: (240, 281)
Mouse: (225, 276)
Mouse: (213, 275)
Mouse: (203, 273)
Mouse: (195, 272)
Mouse: (190, 272)
Mouse: (186, 271)
Mouse: (183, 270)
Mouse: (180, 270)
Mouse: (179, 270)
Mouse: (174, 268)
Mouse: (172, 268)
Mouse: (167, 266)
Mouse: (155, 266)
Mouse: (138, 266)
Mouse: (114, 266)
Mouse: (87, 266)
Mouse: (59, 266)
Mouse: (32, 266)
Mouse: (1, 266)

9.


(self.output_port.framesize, macroblocks_limit))
picamera.exc.PiCameraValueError: output resolution 1920x1200 exceeds macroblock limit (8192) for the selected H.264 profile and level
pi@raspberrypi:~$ sudo python /home/pi/Downloads/pir.py
Motion Detected
Motion Detected
mmal: mmal_vc_port_enable: failed to enable port vc.null_sink:in:0(OPQV): ENOSPC
mmal: mmal_port_enable: failed to enable connected port (vc.null_sink:in:0(OPQV)
)0x2475d90 (ENOSPC)
mmal: mmal_connection_enable: output port couldn't be enabled
Traceback (most recent call last):
File "/home/pi/Downloads/pir.py", line 11, in <module>
camera = picamera.PiCamera()
File "/usr/lib/python2.7/dist-packages/picamera/camera.py", line 433, in __init__
self._init_preview()
File "/usr/lib/python2.7/dist-packages/picamera/camera.py", line 513, in _init_preview
self, self._camera.outputs[self.CAMERA_PREVIEW_PORT])
File "/usr/lib/python2.7/dist-packages/picamera/renderers.py", line 558, in __init__
self.renderer.inputs[0].connect(source).enable()
File "/usr/lib/python2.7/dist-packages/picamera/mmalobj.py", line 2212, in enable
prefix="Failed to enable connection")
File "/usr/lib/python2.7/dist-packages/picamera/exc.py", line 184, in mmal_check
raise PiCameraMMALError(status, prefix)
picamera.exc.PiCameraMMALError: Failed to enable connection: Out of resources
pi@raspberrypi:~$