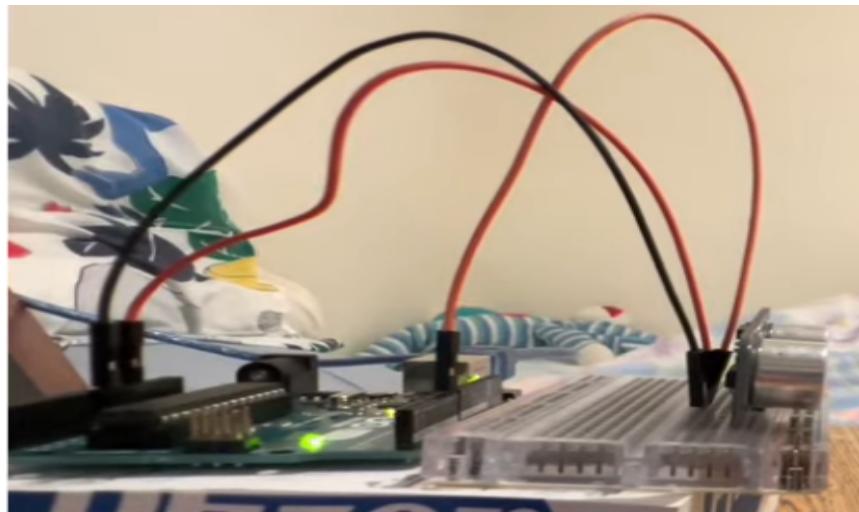


Motion Activated Security System



Prepared By:

Victor Ku

Table of Contents

Product Overview	3
Product Objectives	3
Goal 1: Automatically Open the Camera and Record a Video	4
Subgoal 1: Install Ubuntu	4
Subgoal 2: Gain Camera Access and Ensure Recording is Playable	6
Subgoal 3: Transfer video files from virtual machine to host machine	6
Subgoal 4: Automating everything	7
Goal 2 Sending a Notification to My Phone from my Computer	10
Method 1: Using Google Voice (Fail)	10
Method 2: Sending a Text Message Via Email. (Fail)	11
Method 3: Utilizing Different Messaging APIs. (Fail)	12
Method 4: Command Line Based Texting Apps (Success)	12
Goal 3: Goal 3: Creating a Sensor that can Trigger my Computer to Begin Recording	14
Subgoal 1: Design a Simple Circuit to Test	14
Subgoal 2: Installing the Distance Sensor	17
Final Product	22

Product Overview

As a college student living in a college apartment with other students, oftentimes for the first time, there are moments when I feel uncomfortable leaving my room unlocked. I've heard many horror stories from friends and families about someone losing their valuables because of their roommate.

I could purchase a lock and ensure the most important items are secured, but there is still a risk of losing other items. I could also purchase my own security camera that constantly monitors my room, but that can be costly and is excessive. For this project, I wanted to build a security system that is inexpensive and not over the top that could potentially solve this problem.

I will first focus on the software portion of the project before moving onto the hardware portion. My first instinct is to use Linux to create a bash script to open the camera and begin recording and send me a text message. For the hardware, I've only had experience with Arduinos. Therefore, I'd like to see if I can attach an ultrasonic sensor to one of their boards.

Product Objectives

- Bash script that automatically...
 - Access and record using my computer camera
 - Send a text message to my phone, indicating that recording is in progress
- Use an Ultrasonic sensor to record distance
 - Have some sort of trigger to begin the bash script
- Be able to use Linux in on a Windows computer

Goal 1: Automatically Open the Camera and Record a Video

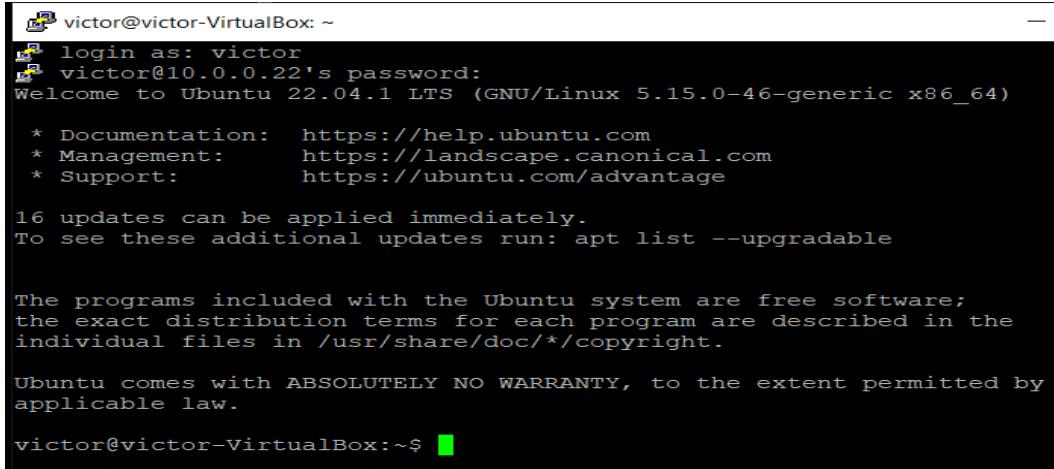
When starting this project, I first need to decide which operating system to work in. My personal computer runs Windows, but I know Linux has a lot more tools. After comparing the two, I ended up choosing Linux for the following reasons: I have more experience using the Linux terminal, I can run bash scripts, and I can use vim. Now I need to decide how to install Linux onto my computer.

Subgoal 1: Install Ubuntu

Linux and an open source operating system with many powerful tools; however, lacks a lot of things that make a complete operating system useful for people. An example is in the Linux terminal, you can type all the commands, but there is no GUI where you can use a mouse to interact with the system. That's where Ubuntu comes in. Ubuntu is equipped with packages that make it a better version of Linux, which is why I chose it.

Here are the ways of downloading Ubuntu from my research:

- Download wsl and use it for Ubuntu on the host machine
 - WSL stands for Windows Subsystem for Linux, which allows Windows users to use Linux tools without the need to dual boot.
 - **Instructions:** <https://docs.microsoft.com/en-us/windows/wsl/install>
 - **Problem:**
<https://answers.microsoft.com/en-us/windows/forum/all/installing-wsl-and-rebooting-makes-my-laptop-go/9c3d9c03-6c69-4b65-9583-fb73989524f7>
 - **Attempted Solution:** Downloaded WinDbg in order to read Minidump files; however, no files were created after the BSOD.
 - Updated the BIOS settings, drivers, latest updates. The problem persisted.
- SSHing into Virtual Machine from Host cmd line.
 - This method could be useful in case I cannot interact with the virtual machine itself. This method would be my plan B.
 - Downloaded Putty for remote access to server computers over a network using the SSH protocol. Attempted to connect via my virtual machine's ip address.
 - **Problem:** Connection Refused Error: It looks like my SSH Service is down and is not looking for connections to accept.
 - See reason one on this website:
<https://kinsta.com/knowledgebase/ssh-connection-refused/#why-is-my-ssh-connection-refused-5-reasons-for-connectivity-errors>
 - **Solution:** Simply had to reinstall SSH.
<https://askubuntu.com/questions/1161579/ssh-server-cannot-be-found-even-though-installed>
 - Exact same functionality as Ubuntu in my virtual machine:



```
victor@victor-VirtualBox: ~
login as: victor
victor@10.0.0.22's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

16 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

victor@victor-VirtualBox:~$
```

- Use Linux inside Windows in a virtual machine:
 - It is likely this is the best option and will try this route. **Assume I am using this method unless I state otherwise.**
 - Successfully downloaded [Oracle VM Virtual Box](#) and installed Ubuntu

Subgoal 2: Gain Camera Access and Ensure Recording is Playable

Ubuntu comes with software already installed. Here, we are testing to ensure Cheese (Ubuntu's camera application) is functioning properly.

- **Problem:** Camera cannot be found. The virtual machine blocks all access to hardware, ports, CD drives, etc. unless given permission.
 - **Solution:** <https://www.youtube.com/watch?v=ec4-1gF-cNU>
- **Problem:** Cheese is EXTREMELY laggy and quality is very poor.
 - **Solution:** Downloaded Guvcview, a webcam application, to speed up the fps.
 - Resolution is 680 x 480 and fps is 5/1 for maximum smoothness.
 - https://www.youtube.com/watch?v=PwjFR_4aalY&t=1s
- **Problem:** Can't open recording in Videos (Ubuntu's video playback application)
 - Received an "Unable to create a GL context" error. It turns out my Nvidia 1660-TI graphics card is not supported with OpenGL2.1, an API used to render graphics, and developers do not plan on upgrading it anytime soon.
 - **Solution:** I downloaded another video player, VLC.

Subgoal 3: Transfer video files from virtual machine to host machine

- **Instructions:** <https://www.youtube.com/watch?v=TcrfrVNNGMU>
- **Problem:** Could not mount the media/drive 'C:\Program Files\Oracle\VirtualBox\VBoxGuestAdditions.iso' (VERR_PDM_MEDIA_LOCKED).
 - **Solution:** <https://www.youtube.com/watch?v=PVuoYzhurSw>
 - Simply had to eject the current disk and install the proper one.
- My shared folder in Ubuntu is called **sf_SecurityCamera**
- The shared folder in my host machine is called **SecurityCamera**

Subgoal 4: Automating everything

- The whole purpose of a bash script is to run multiple terminal commands together as a way of automating a task. The goal of my bash script should accomplish the following:
 - In Ubuntu, open the camera and begin recording and exit once finished.
 - Locate the recording (destination is the home folder) and move it into the shared folder that is connected to the Host computer. Ensure there are no overwritten/missing files.
- Algorithm
 - Each file will be named in the following format: recording-#.mkv
 - **If the (home folder > shared folder) in terms of recordings**, we first need to move it to the shared folder, before lowering the # to the next largest recording in the shared folder.
 - Ex. 5 files in home and 2 in shared.
 - When we add a new recording, it will be named recording-6.mkv.
 - We first move recording-6.mkv from home to shared folder
 - Then we rename to recording-3.mkv
 - This is to prevent overwriting recording-3.mkv in the home folder.
 - **If (home folder = shared folder)**, we simply move from home to shared folder. No renaming necessary.
 - **If the (home folder < shared folder) in terms of recordings**, we first need to increase the # to the next largest recording in the shared folder, then move the recording from home to shared folder.
 - Ex. 2 files in home and 8 in shared.
 - When we add a new recording, it will be named recording-3.mkv.
 - We first rename to recording-9.mkv
 - Then we move recording-9.mkv from home to shared folder
 - This prevents overwriting recording-3.mkv in the shared folder.
 - **Note:** It appears that the for loops run at least once, even if the condition is false. That is why we check if file/file1 exists to break out of the loop in case it isn't so we don't increment our counter. See **code on page 8**.
- **Disclaimer:** It is important to note that this bash scripts only works if the recordings in either folder are named sequentially
 - Ex. 3 recordings in shared folder means there are the following: recording-1.mkv, recording-2.mkv, recording-3.mkv
- Bash Script Code

```

#!/bin/bash

#automatically record a video using Guvcview
#Params: the device we are using, resolution, file format, FPS,
#filename, video duration, exiting guvcview after recording
guvcview -d /dev/video0 -x 640x480 -f MJPG -F 5/1 -j recording -y 3 -e

#counts the number of recordings in the shared folder
filesInSharedFolder=0
for filename in /media/sf_SecurityCamera/recording-* .mkv
do
    #checking if FILE exist
    FILE=/media/sf_SecurityCamera/recording-1.mkv
    if test -f "$FILE"; then
        filesInSharedFolder=$((filesInSharedFolder + 1))
    else
        break
    fi
done

#counts the number of recordings in the home folder
filesInHomeFolder=0
for filename in /home/victor/recording-* .mkv
do
    #checking if FILE1 exist
    FILE1=/home/victor/recording-1.mkv
    if test -f "$FILE1"; then
        filesInHomeFolder=$((filesInHomeFolder + 1))
    else
        break
    fi
done

#the value for the new recording
sharedFolderNextIndex=$((filesInSharedFolder + 1))

#when there are more recordings in home folder:
#we move from home to shared folder, then update the name.
if [ $filesInHomeFolder -gt $sharedFolderNextIndex ]
then
    sudo mv /home/victor/recording-${filesInHomeFolder}.mkv /media/sf_SecurityCamera
    sudo mv /media/sf_SecurityCamera/recording-${filesInHomeFolder}.mkv
    /media/sf_SecurityCamera/recording-${sharedFolderNextIndex}.mkv

```

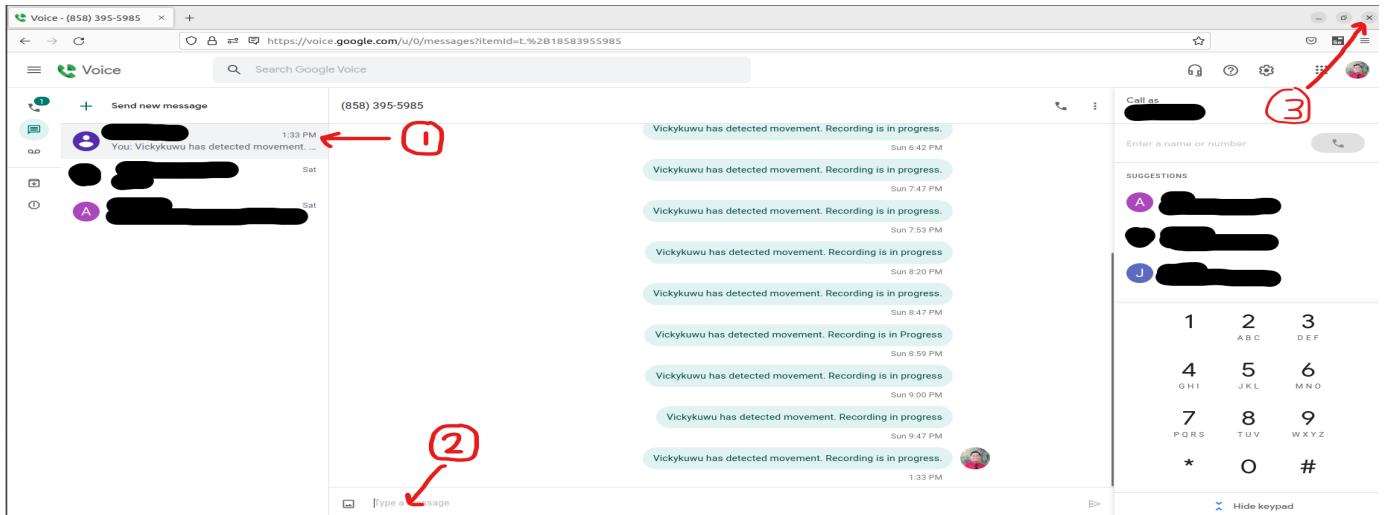
```
#when the recordings between home and shared folder are equivalent:  
#we move from home to shared folder  
elif [ $filesInHomeFolder -eq ${sharedFolderNextIndex} ]  
then  
    sudo mv /home/victor/recording-${filesInHomeFolder}.mkv /media/sf_SecurityCamera  
  
#when there are more recordings in shared folder:  
#we update the name, then move from home to shared folder  
else  
    sudo mv /home/victor/recording-${filesInHomeFolder}.mkv  
/home/victor/recording-${sharedFolderNextIndex}.mkv  
    sudo mv /home/victor/recording-${sharedFolderNextIndex}.mkv /media/sf_SecurtiyCamera  
fi
```

Goal 2: Sending a Notification to My Phone from my Computer

Now that I have my bash script that records the intruder after running “bash record.sh”, I’d like to be notified in real time via some sort of notification. Ideally, the notification would be a text message.

Method 1: Using Google Voice (fail)

- Google Voice gives you a phone number that allows you to text or call from your phone, computer, and the web. I am hoping that in my bash script, I open the Google Voice webpage and text the phone number connected with my phone.



- In order to text, I need to first click on the conversation with my phone. Then click on the chat box and type out my message. Finally, close the entire web page.
- I tried using Selenium IDE in order to record my movements on the web and produce a script to automate them.
- **Problem:** Script is having trouble clicking the chat box.
 - I tried rerecording the scripts a few times, avoiding as many commands as possible, and searched the internet for possible solutions. Unfortunately, I could not find anything.
- I tried using alternatives to Selenium IDE such as Testim and Katalon Recorder. Recording with these software mimicked my movements perfectly; but I had trouble running it from the command line. They required an extensive amount of configuration that ultimately made it too difficult to even try to run the actual script.

Command	Target	Value
1 ✓ open	/u/0/messages	
2 ✓ set window size	1846x933	
3 ✓ click	css=div:nth-child(1) > .gvC onversationList-item .md-in k-ripple	
4 ✗ run script	window.scrollTo(0,0)	
5 click	id=input_0	

Method 2: Sending a Text Message Via Email. (fail)

- Many wireless carriers have the capability to take in an email and deliver it as a text message to a mobile phone. I am thinking of utilizing this feature by sending an email directly from the terminal using the following command: `echo 'Sent from my terminal!' | mail -s 'Linux is fun' '1111111111@vtext.com'`
- Instructions:** <http://www.usrsb.in/blog/blog/2011/03/24/sending-emails-and-texts-from-the-command-line/>
- Problem:** command doesn't do anything
 - After writing out the command and press enter, nothing happens. Not even "sent from my terminal!" appears and I cannot find the reason for this.
- I wanted to simplify the problem to see if the idea worked in the first place and tried sending an email from my personal Gmail account. Here are the instructions straight from verizon's website: <https://www.verizon.com/about/news/vzw/2013/06/computer-to-phone-text-messaging>

Option 2: Use @vtext.com to send text through email

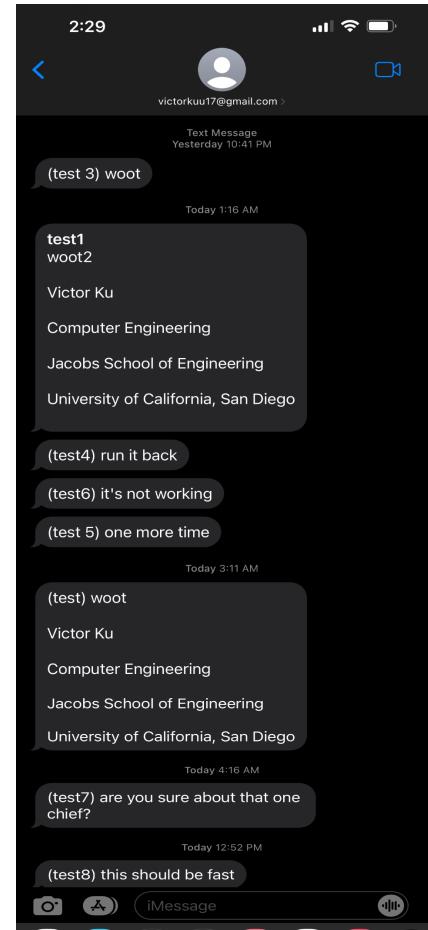
Here's another way to send a text message from a computer to a fellow Verizon Wireless customer:

- Compose a new email and use the recipient's mobile phone number as the email address, with the addition of "@vtext.com" at the end.
- For example, if the phone number is 555-123-4567, type "5551234567@vtext.com."
- Make sure you keep your message under 140 characters. (Remove your email signature before sending, if you use one.)
- Once you're finished, hit send.

- Problem:** messages are not instantaneously received
- The point of sending a notification to my phone is to alert me in real time that an intruder gained access to my room, therefore it is critical to receive the message immediately. I ran 10 test messages and here are the results:

□ ★ To: 8583955985	test 10 - with caller	11:36 AM
□ ★ To: 8583955985	test 9 - this one should be slow	11:15 AM
□ ★ To: 8583955985	test8 - this should be fast	11:14 AM
□ ★ To: 8583955985	test7 - are you sure about that one chief?	1:33 AM
□ ★ To: 8583955985	test6 - it's not working	1:32 AM
□ ★ To: 8583955985	test 5 - one more time	1:28 AM
□ ★ To: 8583955985	test4 - run it back	1:27 AM
□ ★ To: 8583955985	test 3 - woot	10:12 PM
□ ★ To: 8583955985	test1 - woot2 Victor Ku Computer Engineering	10:07 PM
□ ★ To: 8583955985	test - woot Victor Ku Computer Engineering	10:02 PM

<u>Results</u>		
Test~ 5 hrs	Test 1 ~3 hrs	Test 3~ 30 mins
Test 4 ~1 min	Test 5 ~7 mins	Test 6~ 0 mins
Test 7 ~3 hrs	Test 8 ~2 hrs	Test 9/10~ missing



- As you can see, the results are inconsistent with only 2 near instantaneous messages, 5 messages delayed at least by 30 minutes, and 2 flat out missing.
- I tried calling Verizon to see if I could expedite the process. The representative sent me a screenshot of the database handbook that explains the VText feature. Here's what it says:

VText.com

Usage Policy Server Migration Changes Troubleshooting

Vtext.com is for consumer use only.

The service is not intended to be used for large volumes of commercial messages or emergency messages. Due to the inherent limitations of Internet email, Verizon provides no assurance regarding the timeliness or receipt of messages sent using this service. Businesses and organizations looking to send a high volume of messages should subscribe to Verizon Enterprise Messaging.

For more information, see [Enterprise Messaging](#).

- Without an “assurance regarding the timeliness or receipt of messages,” I unfortunately must find another method.
- **Solution:**
 - Rather than sending an email via Vtext, we send an email using Google Voice! Texts are delayed no longer than 2 minutes!
 - <https://stackoverflow.com/questions/11388640/is-it-possible-to-use-google-apps-script-to-send-a-text-message-when-a-missed-call>
- The thing is, I still cannot send an email from the command line. Tried troubleshooting the problem but the problem still persisted.

Method 3: Utilizing Different Messaging APIs. (Fail)

- I thought I'd use another API, like Telegram, Messenger, Discord, to send my messages. This failed for multiple reasons: There's no notification if I send messages to myself and I still cannot integrate these messaging apps to my command line.
- I then tried using apps that are designed to send notifications from a computer to my phone. I tested Push bullet, Airmore, KDE Connect, Nfty, and Qpush. They all had a problem with either the application cost money or the app was designed specifically for Android devices.

Method 4: Command Line Based Texting Apps (Success)

- This time, I specifically looked for applications that can easily be integrated into the command line. I found this GitHub repository designed to connect Google Voice with the command line. While it may no longer be supported, it did recommend APIs that are fully functional.
- After setting up my account, I simply had to install Twilio-CLI with these instructions: <https://www.twilio.com/docs/twilio-cli/getting-started/install>
- Finally, I followed these instructions: <https://www.youtube.com/watch?v=rrx4ux-hChw>,
 - I registered with a phone number they supplied and copied and pasted the code Twilio generated into my terminal and I received a message! Code is on page 13

- **Disclaimer:** I am currently using the free trial, which gives me \$15.50 for free. Each text message sent is \$0.0075. The text must begin with “Sent from your Twilio trial account -”.

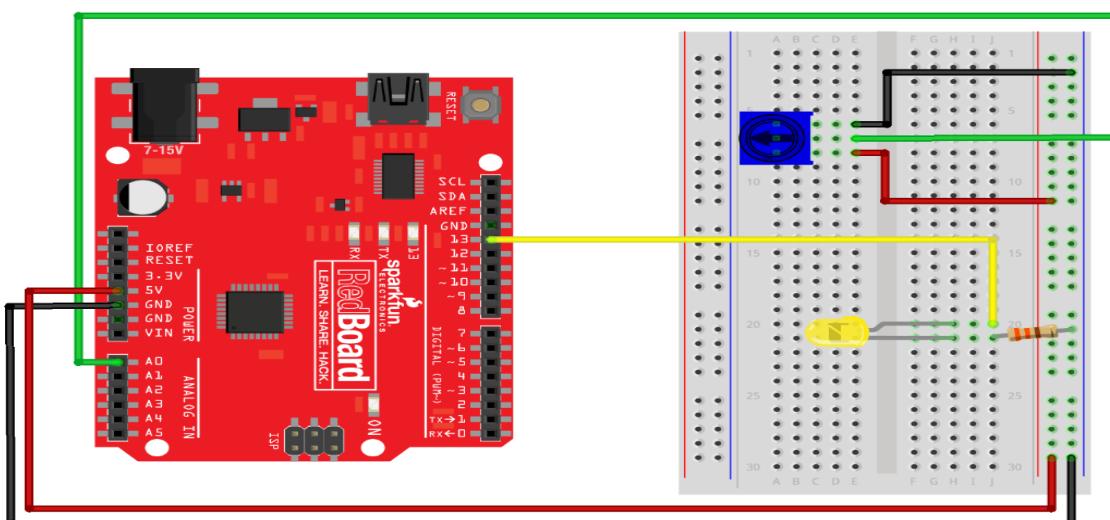
```
$url = "https://api.twilio.com/2010-04-01/Accounts/AC69b35c67ff6813562e59dc0d58e456b5/Messages.json"
$params = @{
    To = "+18583955985";
    From = "+19705361698";
    Body = "Hello from Twilio"
}
$secret = "b60aedf78b125b3e759f294aed0ec7a4" | ConvertTo-SecureString -asPlainText -Force
$credential = New-Object
System.Management.Automation.PSCredential(AC69b35c67ff6813562e59dc0d58e456b5, $secret)
Invoke-WebRequest $url -Method Post -Credential $credential -Body $params -UseBasicParsing |
ConvertFrom-Json | Select sid, body
```

Goal 3: Creating a Sensor that can Trigger my Computer to Begin Recording

When beginning the hardware portion, I knew it must be able to do three things: run a bash script from the terminal, send data to my computer ready to be processed, and have the capability to install a distance sensor. I wanted to begin my research with Arduino because I have experience with it. My first step is to create a simple schematic to test and see if the three features are possible.

Subgoal 1: Design a Simple Circuit to Test

- I started with building a LED circuit and had my computer read the values from a potentiometer using Python. The circuit itself makes the LED turn on if the potentiometer reading is above 500 and off if under 500.
- **Specs for the sensor:**
<https://www.mouser.com/datasheet/2/321/28015-PING-Sensor-Product-Guide-v2.0-461050.pdf>
- **Instructions:**
 - **Instructions for installing Arduino IDE in Ubuntu:**
 - <https://ubuntu.com/tutorials/install-the-arduino-ide#2-installing-via-a-tarball>
 - **Instructions for allowing my virtual machine to access a usb port:**
 - <https://help.ubuntu.com/community/VirtualBox/USB#:~:text=For%20temporary%20device%20connection%20to%20VM&text=Start%20or%20go%20to%20VM.as%20normal%20for%20the%20OS.>
 - I specifically had to import the PySerial package in order to connect with my Arduino circuit from the terminal.
 - **Instructions:**
 - https://www.youtube.com/watch?v=lk8SZHUG8_s
 - **Problem:** Permission denied on the port connecting to the Arduino /dev/ttyACM0
 - **Solution:** sudo chmod a+rwx /dev/ttyACM0
- Schematic
 - Here is the Arduino schematic taken from [Reading a Sensor with Python](#) that prints the potentiometer values.



- Arduino Code

- Here is the Arduino code taken from [Reading a Sensor with Python](#) that turns the LED on if the potentiometer reading is above 500 and off if under 500.

```
// potentiometer_read.ino
// reads a potentiometer and sends value over serial
int sensorPin = A0; // The potentiometer on pin 0
int ledPin = 13; // The LED is connected on pin 13
int sensorValue; // variable to stores data

void setup() // runs once when the sketch starts
{
    // make the LED pin (pin 13) an output pin
    pinMode(ledPin, OUTPUT);

    // initialize serial communication
    Serial.begin(9600);
}

void loop() // runs repeatedly after setup() finishes
{
    sensorValue = analogRead(sensorPin); // read pin A0
    Serial.println(sensorValue); // send data to serial

    if (sensorValue < 500) { // less than 500?
        digitalWrite(ledPin, LOW); } // Turn the LED off

    else { // greater than 500?
        digitalWrite(ledPin, HIGH); } // Keep the LED on

    delay(100); // Pause 100 milliseconds
}
```

- Python Code

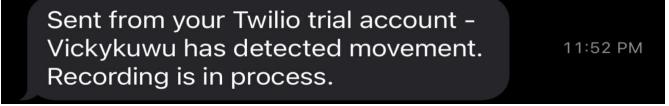
- Here is the Python code taken from [Reading a Sensor with Python](#) that prints the potentiometer values.

```
import serial
import time
import subprocess

# set up the serial line
ser = serial.Serial('/dev/ttyACM0', 9600)
time.sleep(2)

# Read and record the data
data = [] # empty list to store the data
for i in range (50):
    b = ser.readline() # read a byte string
    string_n = b.decode() # decode byte string into Unicode
    string = string_n.rstrip() # remove \n and \r
    print(string)
    data.append(string) # add to the end of data list
    time.sleep(0.1) # wait (sleep) 0.1 seconds
ser.close()

# show the data
for line in data:
    print (line)
```

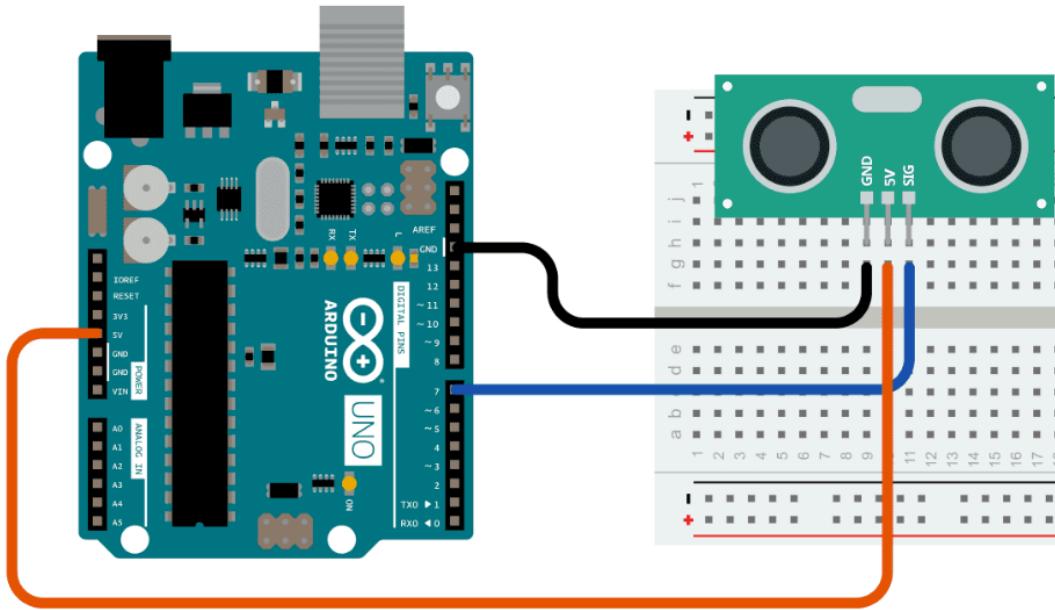
- Finally, I added the following line in order for my python script to run my bash script that automatically records a video from my computer camera.
`subprocess.call('/home/victor/Desktop/camera automation/record.sh')`
- **Problem:** [errno 13] permission denied–python script cannot run the bash script.
 - **Solution:** When running `ls -la`, we see `record.sh` is owned by the root user: `-rw-rw-r-- 1 victor victor 2332 Sep 2 01:13 record.sh`
 - We don't run python from the root user, so we must change the readability of `record.sh` with the following: `chmod 755 record.sh`
- Recording of how the security system currently work: <https://youtu.be/7jp7mgTlj2U>
- Recording of the footage captured by my script, accessed by my host computer: https://youtu.be/wp_SgoZz_E8
- Message received on my phone:
 

Sent from your Twilio trial account -
Vickykuwu has detected movement.
Recording is in process.
- It looks like Arduino is able to take care of all the requirements necessary for my project. We will be keeping this method unless any further issues arise.

Subgoal 2: Installing the Distance Sensor

I happened to find an old Parallax 28015 Ultrasonic Distance Sensor while rummaging through my Arduino kit, so I will be using that as my sensor. It can measure anywhere from 2 centimeters to 3 meters.

- **Schematic** of Ultrasonic Distance Sensor found on [Arduino website](#):



- The Ultrasonic Distance Sensor has three pins. The ground pin ensures that when there is a short circuit, the electrical current can travel along this route to prevent damage to the rest of the circuit. The 5V represents the battery and the amount of voltage the circuit will take. SIG is the pin number that will be used to connect the sensor with the Arduino IDE.

- Arduino Code found on [Arduino website](#)
 - After a two second delay, the Arduino will record values from the Ultrasonic Distance Sensor every 0.1 seconds. With simple math, we can convert those values into centimeters and inches. Finally, those values will be printed. [Output](#)

```
/*
Ping))) Sensor

This sketch reads a PING))) ultrasonic rangefinder and returns the distance
to the closest object in range. To do this, it sends a pulse to the sensor to
initiate a reading, then listens for a pulse to return. The length of the
returning pulse is proportional to the distance of the object from the sensor.

The circuit:
- +V connection of the PING))) attached to +5V
- GND connection of the PING))) attached to ground
- SIG connection of the PING))) attached to digital pin 7

created 3 Nov 2008
by David A. Mellis
modified 30 Aug 2011
by Tom Igoe
```

10cm, 4in
 5cm, 2in
 6cm, 2in
 10cm, 3in
 17cm, 6in
 27cm, 10in
 32cm, 12in
 36cm, 14in
 32cm, 12in
 27cm, 10in
 17cm, 6in
 11cm, 4in
 8cm, 3in
 8cm, 3in
 13cm, 5in

This example code is in the public domain.

```
https://www.arduino.cc/en/Tutorial/BuiltInExamples/Ping
*/
// this constant won't change. It's the pin number of the sensor's output:
const int pingPin = 7;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}

void loop() {
  // establish variables for duration of the ping, and the distance result
  // in inches and centimeters:
  long duration, inches, cm;

  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
```

```
delayMicroseconds(5);
digitalWrite(pingPin, LOW);

// The same pin is used to read the signal from the PING)): a HIGH pulse
// whose duration is the time (in microseconds) from the sending of the ping
// to the reception of its echo off of an object.
pinMode(pingPin, INPUT);
duration = pulseIn(pingPin, HIGH);

// convert the time into a distance
inches = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);

Serial.print(inches);
Serial.print("in, ");
Serial.print(cm);
Serial.print("cm");
Serial.println();

delay(100);
}

long microsecondsToInches(long microseconds) {
    // According to Parallax's datasheet for the PING)), there are 73.746
    // microseconds per inch (i.e. sound travels at 1130 feet per second).
    // This gives the distance traveled by the ping, outbound and return,
    // so we divide by 2 to get the distance of the obstacle.
    // See:
    https://www.parallax.com/package/ping-ultrasonic-distance-sensor-downloads/
    return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
    // The speed of sound is 340 m/s or 29 microseconds per centimeter.
    // The ping travels out and back, so to find the distance of the object we
    // take half of the distance traveled.
    return microseconds / 29 / 2;
}
```

- Python Code that has been modified from [Reading a Sensor with Python](#)

```

import serial
import time
import subprocess
import re

# set up the serial line
ser = serial.Serial('/dev/ttyACM0', 9600)
time.sleep(2)

# Read and record the data
data = []                                     # empty list to store the data (string)
data_int = []                                    # empty list to store the data (integers)
for i in range (100):
    b = ser.readline()                         # read a byte string
    string_n = b.decode()                      # decode byte string into Unicode
    string = string_n.rstrip()                 # remove \n and \r
    print(string)

    number_array = re.findall(r'\d+', string)   # Extracting only numbers from the string

    if len(number_array) > 1:                  # ensure both cm and inch values appear in the array
        centimeter = int(number_array[1])        # turning the value in cm into a int
        data_int.append(centimeter)               # add the integer to the end of data_int list

    if len(data_int) > 2:                      # ensure there are at least two measurements
        first = data_int[len(data_int) - 6]       # finds the first measurement
        second = data_int[len(data_int) - 1]        # finds the second measurement
        if abs(second - first) > 3:              # checks if the difference is greater than 3cm
            print ("oooooooo")
    data.append (string)
    time.sleep(0.1)
ser.close()

```

- The difference between this and the previous code is that in this code we want to track movement. This means we need to have two different points and compare if the values are different.
 - We first need to extract the values from a string given by the Arduino code.
 - To do this, we use the package **re**:
<https://stackoverflow.com/questions/4289331/how-to-extract-numbers-from-a-string-in-python>
 - Occasionally, we are only given the cm reading, which is all we need, but in case this happens, we ensure it doesn't stop the rest of the code from running.
 - Once the value is extracted, we store it in an array, which will then be used to compare with other readings.

- The moment we get two readings, we compare the current value with the value 0.5 seconds ago and if we see a difference greater than 3cm, we know motion is being detected and we print “wooooo”.
- Now that we know this works, we simply replace “ooooooooo” with the command to run our recording bash script and break out of the loop so that we only run it once.
- **Video of the sensor in action:** <https://youtu.be/H5Fa92FtB2M>

Final Product:

Video of the final product: <https://youtu.be/Cov01Y6bvrM>

Video of the security footage captured by my camera: <https://youtu.be/93eusGNqJbg>

Python Code:

```

import serial
import time
import subprocess
import re

# set up the serial line
ser = serial.Serial('/dev/ttyACM0', 9600)
time.sleep(2)

# Read and record the data
data = []                                     # empty list to store the data (string)
data_int = []                                    # empty list to store the data (integers)
while(True):
    b = ser.readline()                         # read a byte string
    string_n = b.decode()                      # decode byte string into Unicode
    string = string_n.rstrip()                 # remove \n and \r
    print(string)

    number_array = re.findall(r'\d+', string)   # Extracting only numbers from the string

    if len(number_array) > 1:                  # ensure both cm and inch values appear in the array
        centimeter = int(number_array[1])       # turning the value in cm into a int
        data_int.append(centimeter)              # add the integer to the end of data_int list

    if len(data_int) > 2:                      # ensure there are at least two measurements
        first = data_int[len(data_int) - 6]      # finds the first measurement
        second = data_int[len(data_int) - 1]      # finds the second measurement
        if abs(second - first) > 3:             # checks if the difference is greater than 3cm
            print("Motion has been detected")
            subprocess.call('/home/victor/Desktop/camera_automation/record.sh')
            break;
    data.append(string)
    time.sleep(0.1)
ser.close()

```

Arduino Code:

```
/*
Ping))) Sensor

This sketch reads a PING))) ultrasonic rangefinder and returns the distance
to the closest object in range. To do this, it sends a pulse to the sensor to
initiate a reading, then listens for a pulse to return. The length of the
returning pulse is proportional to the distance of the object from the sensor.
```

The circuit:

- +V connection of the PING))) attached to +5V
- GND connection of the PING))) attached to ground
- SIG connection of the PING))) attached to digital pin 7

```
created 3 Nov 2008
by David A. Mellis
modified 30 Aug 2011
by Tom Igoe
```

This example code is in the public domain.

<https://www.arduino.cc/en/Tutorial/BuiltInExamples/Ping>

```
/*
// this constant won't change. It's the pin number of the sensor's output:
const int pingPin = 7;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
}

void loop() {
  // establish variables for duration of the ping, and the distance result
  // in inches and centimeters:
  long duration, inches, cm;

  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  // The same pin is used to read the signal from the PING))): a HIGH pulse
```

```
// whose duration is the time (in microseconds) from the sending of the ping
// to the reception of its echo off of an object.
pinMode(pingPin, INPUT);
duration = pulseIn(pingPin, HIGH);

// convert the time into a distance
inches = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);

Serial.print(inches);
Serial.print("in, ");
Serial.print(cm);
Serial.print("cm");
Serial.println();

delay(100);
}

long microsecondsToInches(long microseconds) {
    // According to Parallax's datasheet for the PING()), there are 73.746
    // microseconds per inch (i.e. sound travels at 1130 feet per second).
    // This gives the distance traveled by the ping, outbound and return,
    // so we divide by 2 to get the distance of the obstacle.
    // See:
    https://www.parallax.com/package/ping-ultrasonic-distance-sensor-downloads/
    return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
    // The speed of sound is 340 m/s or 29 microseconds per centimeter.
    // The ping travels out and back, so to find the distance of the object we
    // take half of the distance traveled.
    return microseconds / 29 / 2;
}
```

Bash Script Code:

<see next page>

```

#!/bin/bash

#automatically record a video using Guvcview
#Params: the device we are using, resolution, file format, FPS,
#filename, video duration, exiting guvcview after recording
guvcview -d /dev/video0 -x 640x480 -f MJPG -F 5/1 -j recording -y 60 -e

#counts the number of recordings in the shared folder
filesInSharedFolder=0
for filename in /media/sf_SecurityCamera/recording-* .mkv
do
    #checking if FILE exist
    FILE=/media/sf_SecurityCamera/recording-1.mkv
    if test -f "$FILE"; then
        filesInSharedFolder=$((filesInSharedFolder + 1))
    else
        break
    fi
done

#counts the number of recordings in the home folder
filesInHomeFolder=0
for filename in /home/victor/recording-* .mkv
do
    #checking if FILE1 exist
    FILE1=/home/victor/recording-1.mkv
    if test -f "$FILE1"; then
        filesInHomeFolder=$((filesInHomeFolder + 1))
    else
        break
    fi
done

#the value for the new recording
sharedFolderNextIndex=$((filesInSharedFolder + 1))

#when there are more recordings in home folder:
#we move from home to shared folder, then update the name.
if [ $filesInHomeFolder -gt $sharedFolderNextIndex ]
then
    sudo mv /home/victor/recording-${filesInHomeFolder}.mkv /media/sf_SecurityCamera
    sudo mv /media/sf_SecurityCamera/recording-${filesInHomeFolder}.mkv
    /media/sf_SecurityCamera/recording-${sharedFolderNextIndex}.mkv

```

```
#when the recordings between home and shared folder are equivalent:  
#we move from home to shared folder  
elif [ $filesInHomeFolder -eq ${sharedFolderNextIndex} ]  
then  
    sudo mv /home/victor/recording-${filesInHomeFolder}.mkv /media/sf_SecurityCamera  
  
#when there are more recordings in shared folder:  
#we update the name, then move from home to shared folder  
else  
    sudo mv /home/victor/recording-${filesInHomeFolder}.mkv  
/home/victor/recording-${sharedFolderNextIndex}.mkv  
    sudo mv /home/victor/recording-${sharedFolderNextIndex}.mkv /media/sf_SecurtiyCamera  
fi
```

```

#!/bin/bash

#automatically record a video using Guvcview
#Params: the device we are using, resolution, file format, FPS,
#filename, video duration, exiting guvcview after recording
guvcview -d /dev/video0 -x 640x480 -f MJPG -F 5/1 -j recording -y 3 -e

#counts the number of recordings in the shared folder
filesInSharedFolder=0
for filename in /media/sf_SecurityCamera/recording-* .mkv
do
    #checking if FILE exist
    FILE=/media/sf_SecurityCamera/recording-1.mkv
    if test -f "$FILE"; then
        filesInSharedFolder=$((filesInSharedFolder + 1))
    else
        break
    fi
done

#counts the number of recordings in the home folder
filesInHomeFolder=0
for filename in /home/victor/recording-* .mkv
do
    #checking if FILE1 exist
    FILE1=/home/victor/recording-1.mkv
    if test -f "$FILE1"; then
        filesInHomeFolder=$((filesInHomeFolder + 1))
    else
        break
    fi
done

#the value for the new recording
sharedFolderNextIndex=$((filesInSharedFolder + 1))

#when there are more recordings in home folder:
#we move from home to shared folder, then update the name.
if [ $filesInHomeFolder -gt $sharedFolderNextIndex ]
then
    sudo mv /home/victor/recording-${filesInHomeFolder}.mkv /media/sf_SecurityCamera
    sudo mv /media/sf_SecurityCamera/recording-${filesInHomeFolder}.mkv
    /media/sf_SecurityCamera/recording-${sharedFolderNextIndex}.mkv

```

```
#when the recordings between home and shared folder are equivalent:  
#we move from home to shared folder  
elif [ $filesInHomeFolder -eq ${sharedFolderNextIndex} ]  
then  
    sudo mv /home/victor/recording-${filesInHomeFolder}.mkv /media/sf_SecurityCamera  
  
#when there are more recordings in shared folder:  
#we update the name, then move from home to shared folder  
else  
    sudo mv /home/victor/recording-${filesInHomeFolder}.mkv  
/home/victor/recording-${sharedFolderNextIndex}.mkv  
    sudo mv /home/victor/recording-${sharedFolderNextIndex}.mkv /media/sf_SecurtiyCamera  
fi
```