CSE321 Fall 2017 Term Project

Home Security Camera

Matthew Holder
<mjholder@buffalo.edu>
Joshua Emerling
<joshuaem@buffalo.edu>

Problem Statement/Solution

1. Executive Summary

Home Security Camera. The purpose of this project is to help homeowners protect themselves from home invasion without having to search through hours and hours of recordings. It uses a proximity sensor and a camera to only record when there is something a meter in front of the camera opposed to a constant video stream. This is an advantage because a most of the recording from a constantly recording camera has no use, forcing the user to waste time searching the recording. Our project avoids this issue.

Motivation

The problem with many home security cameras is that they are constantly recording. This means that the majority of film is useless, that is it is just recording when nothing interesting/worthwhile is in front of the camera. However our camera will only record when there is something in front of the camera. This will reduce useless footage making it a lot easier for the user to utilize the recording opposed to sifting through hours and hours of useless footage.

Design/Architecture

2. Project objectives

Our Home Security Camera Project will meet the following objectives:

- Provide the user with an easy to use camera that performs mostly on it's own, all the user has to do is turn on the camera then later look at the recordings.

- Record only when there is something useful to record, i.e. when something is in view of the camera.
- The camera will constantly run, starting and stopping recordings when something sets off the proximity sensor.
- Recordings will be in ten second intervals to reduce the chance of recording useless video.

3. Project Approach

We will first work on making the circuit for the HC-SR04 range meter and Raspberry Pi. Then connect the camera to the raspberry pi camera port. Lastly, using Python, we will code the functionality and interaction of both the camera and range meter. This code will run indefinitely, constantly measuring the distance of the nearest object in front of it until something gets within an arbitrary range, 1 meter for the default range, and begin to record for at least 10 seconds.

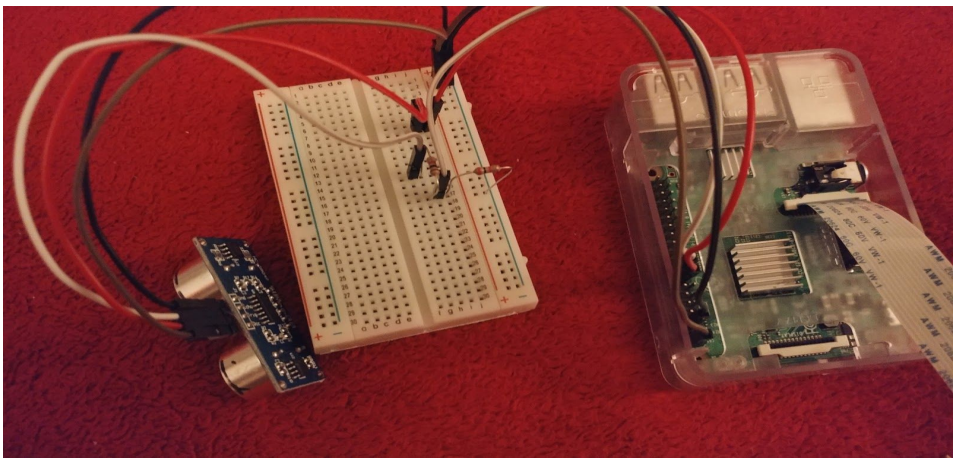
Responsibilities:

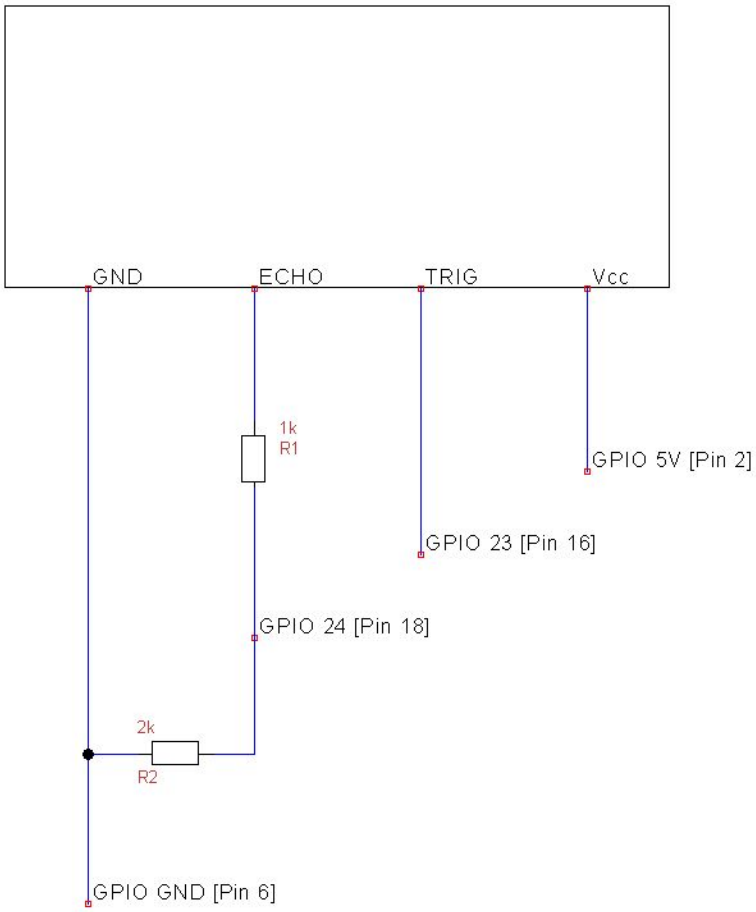
Josh: Software

Matt: Hardware

Technical Description

4. Project Description





The two diagrams above show our implementation and the basic circuitry for the project respectively. The resistors R1 and R2 are required for taking the 5v output of ECHO and reducing it to the 3.3v required for the GPIO input. The python code send a short signal through the TRIG and waits for the input through ECHO. This delay gives us the distance the range meter measured. When we receive ECHO the camera then turns on and appends at least ten seconds of video to the recording. This recording is stored on the camera.

5. Design Details

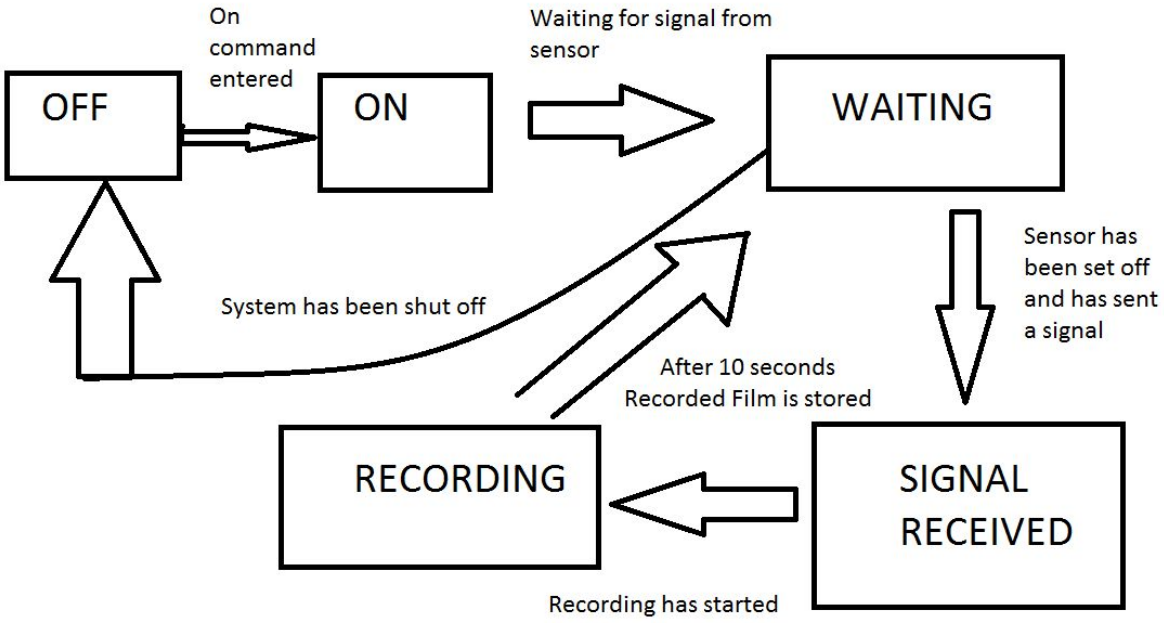
Tools Used:

Raspberry Pi

HC-SR04 Proximity Sensor

Keyestudio 5MP Camera

State Diagram Below:



The system is initially off, upon entering the command “sudo python main.py” the system will start up then move to the waiting state. Here the program is in an infinite loop waiting for the proximity sensor to be set off. Once it is set off it will notify the camera to begin recording. After 10 seconds the recorded film is appended to the already stored film, then the system is back into the waiting state. The loop will continue until the system is shut off.

6. User's Manual

When deployed the camera will have an internet connection and power connection. It will run constantly but can be controlled via SSH. Through SSH you can kill/start the camera and SCP the recording file.

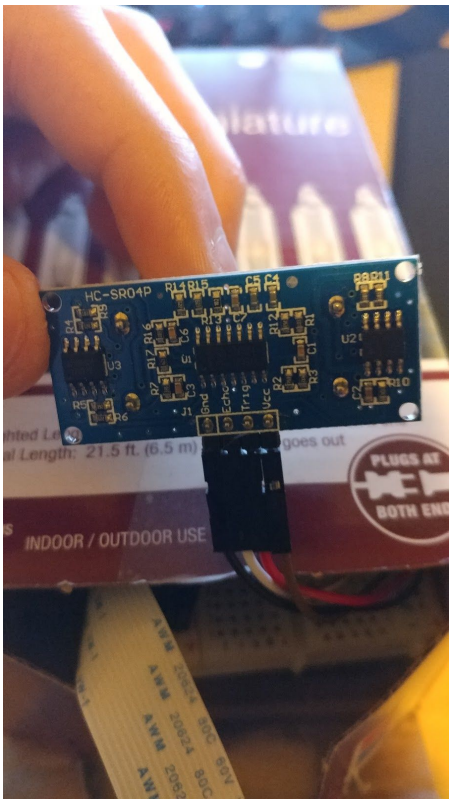
7. Programmer's Manual

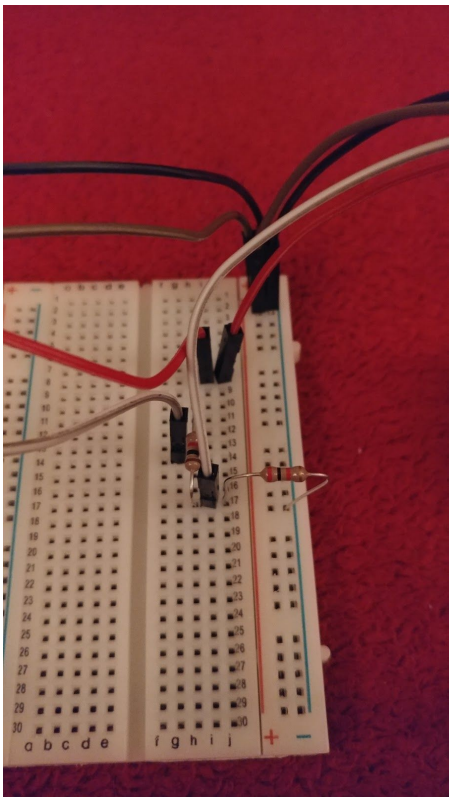
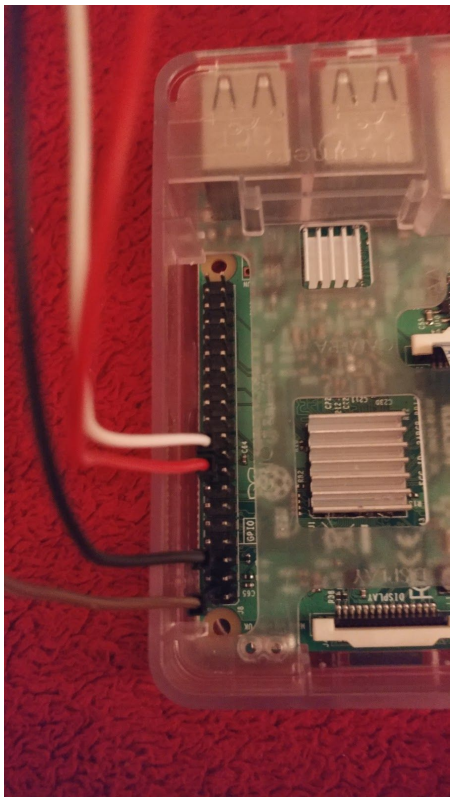
```
import RPi.GPIO as GPIO

import time

import picamera


GPIO.setmode(GPIO.BCM)


camera = picamera.PiCamera()

stream = picamera.PiCameraCircularIO(camera, seconds = 3)

trigger = 23

echo = 24


GPIO.setup(trigger, GPIO.OUT)

GPIO.setup(echo, GPIO.IN)

GPIO.output(trigger, False)


def getDistance():
```

```
  GPIO.output(trigger, True)
  time.sleep(0.00001)
  GPIO.output(trigger, False)

  while GPIO.input(echo) == 0:
    pulse_start = time.time()

  while GPIO.input(echo) == 1:
    pulse_end = time.time()

  pulse_duration = pulse_end - pulse_start
  distance = pulse_duration * 17150
  distance = round(distance, 2)
  return distance

def control():
  # distance in cm
  zone = 100
  curr = 999
  camera.start_recording(stream, format = 'h264')
  moved = 0
  try:
    while(True):
      time.sleep(1)
      curr = getDistance()
```

```
      print curr
      while(curr <= zone):
        if not camera.recording:
          camera.start_recording(stream, format = 'h264')
        print curr
        curr = getDistance()
        camera.wait_recording(5)
        stream.copy_to('motion.h264')
        moved = 1

      if camera.recording and moved:
        camera.stop_recording()

  finally:
    camera.stop_recording()

control()
GPIO.cleanup()
```

To run you only need to run "sudo python TermProgCam.py".



The three pictures are closer views of the wiring which are aligned to show how they connect.

8. References:

HC-SR04:

https://www.modmypi.com/blog/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi

Pi Camera API:

https://picamera.readthedocs.io/en/release-1.13/