EL213 : Analog Circuits

Surveillance using Raspberry Pi Zero W
Group-15

Report

Aniket Kaduskar (201701005)
Dhruv Charan (201701023)
Donika Juthani (201701037)
Dushyant Pathak (201701062)
Bhargey Mehta (201701074)
Daivik Sakaria (201701115)
Kartikeya Gokhale (201701175)
Axil Vaghela (201701220)
Rakshit Vasava (201701236)
Yash Kataria (201701404)

Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar

Contents

1 Abstract and Objective

Using Raspberry Pi, save the picture in a memory from a rotating camera when motion is detected. Capture 8 images for 2 rotations of a camera and save them on a remote server.

The main objective was to integrate the componenets namely the accelerometer, the Pi zero W, the motor driver and the motor to make a system which is able to sense a change in motion (acceleration) and then capture images for surveillance purposes. These images also had to be uploaded to a remote server which can be accessed by a computer with internet access.

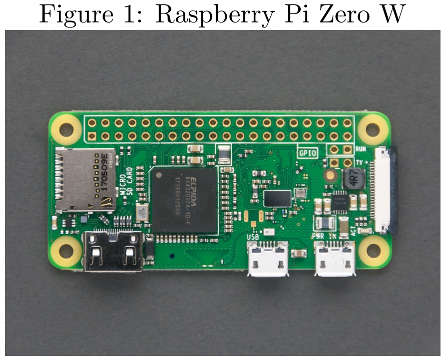
2 Components

- Raspberry Pi Zero W along with a case
- Stepper Motor With Driver (ULN 2003)
- Accelerometer (ADXL-345)
- Raspberry Pi Camera Module
- Power Sources
- Male To Female Wires
- Micro SD Card

2.1 Raspberry Pi Zero W

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. We have used Raspberry Pi zero w model with inbuilt wifi and bluetooth. It is half of the size of other raspberry pi models. It has 1Ghz single core CPU, 512 MB RAM and 40 GPIO pins.It is a slightly more bare-bones version of the original raspberry pi with a couple of features missing and a lower end processor with different versions of the proprietary ports like HDMI and USB.

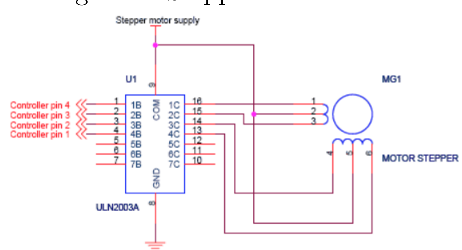
Figure 1: Raspberry Pi Zero W



2.2 Stepper Motor With Driver (ULN 2003)

To rotate the camera we are using Stepper motor as we can rotate it as we want. We are using stepper motor model 28BYJ-48 and as a driver we are using ULN2003 driver. The stepper motor has 2 coils with 2 teeth-rings each, that is, a total of 4 teeth-rings. Each has 8 teeth in it. Through various combinations of the teeth, we can rotate stepper motor as much according to requirement. This allows for us to have it be controlled by an appropriate motor driver so as to meet our requirements.The pi is programmed to rotate at a certain speed once the sensor output changes to HIGH.The motor driver delivers power to the stepper motor at the pre-programmed time intervals and hence the mounted camera on the apparatus rotates along with it and the images are captured as per the predetermined time count.

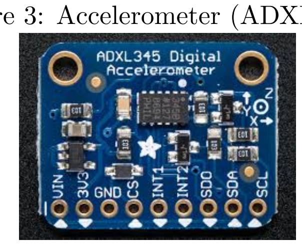
Figure 2: Stepper Motor Driver



2.3 Accelerometer (ADXL-345)

The ADXL345 is a low-power, 3-axis MEMS accelerometer modules with both I2C and SPI interfaces. The boards for these modules feature on-board 3.3v voltage regulation. The ADXL345 features 4 sensitivity ranges from +/- 2G to +/- 16G. It has been used in our project so as to provide the impulse required to start the camera module and to detect the intrusion or accident depending on how and where the device is being used.

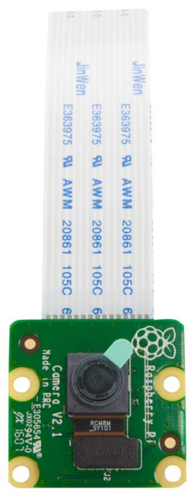
Figure 3: Accelerometer (ADXL-345)



2.4 Raspberry Pi Camera Module

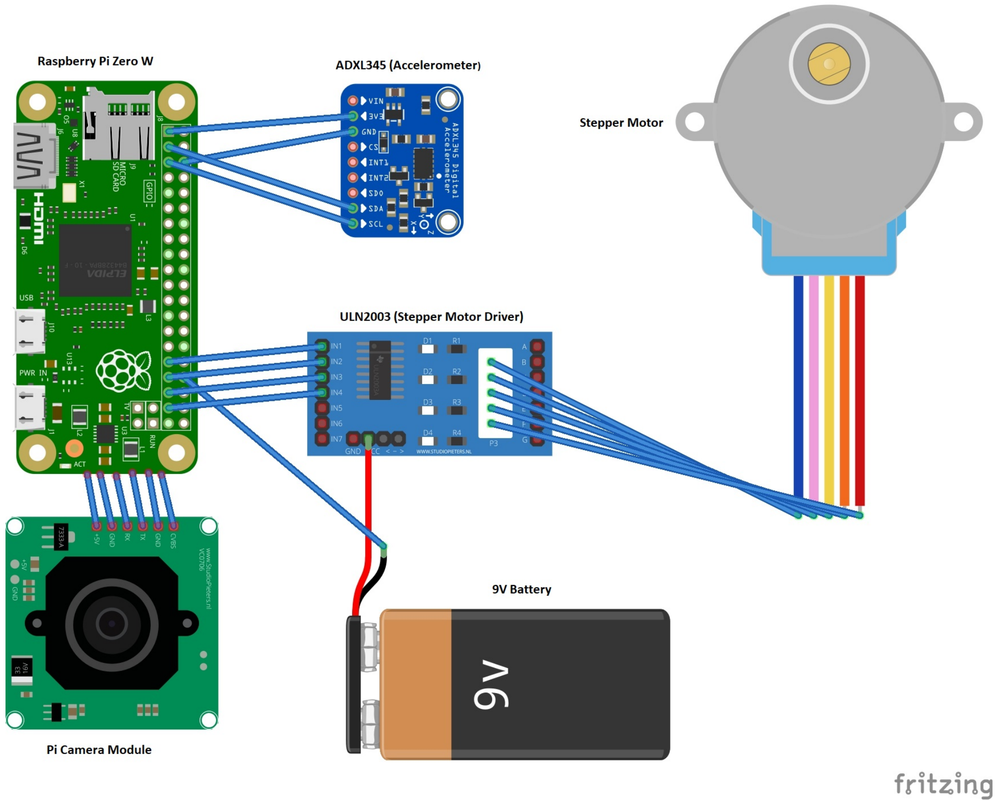
TThe Raspberry Pi Camera Module is a custom designed add-on for Raspberry Pi. It attaches to Raspberry Pi by way of one of the small sockets on the board upper surface. This interface uses the dedicated CSi interface, designed especially for interfacing to cameras. The board is tiny, at around 25mm x 20mm x 9mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short ribbon cable. The sensor itself has a native resolution of 5 megapixel, and has a fixed focus lens on-board. In terms of still images, the camera is capable of 2592 x 1944 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video.

Figure 4: Raspberry Pi Camera Module



3 Block Diagram

Figure 5: Block Diagram



4 Working

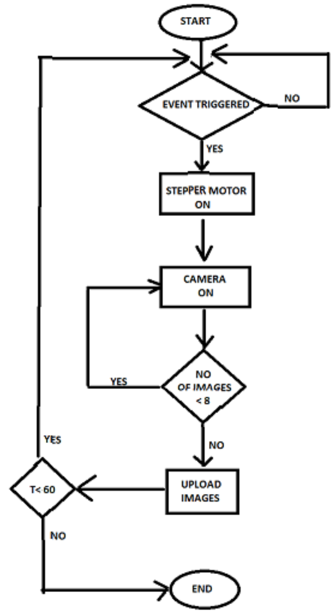
The Raspberry Pi camera module has numerous applications. It can be utilized for security reason anywhere: home, office, schools etc. It might likewise be helpful for wildlife photographer who wish to capture the rare snapshots of the untamed life living space.

Our system captures images at regular intervals and sends it to a remote server whenever the accelerometer detects the shock. To detect the motion, we are using ADXL345 Accelerometer which continuously measures acceleration in X, Y and Z directions. Whenever it exceeds certain threshold, Raspberry Pi Camera module starts rotating and capturing images simultaneously. The Camera module is connected to the Raspberry Pi which is mounted on the stepper motor, controlled by the driver(ULN2003). Here, the driver is rotated at desired intervals using a python script. The captured images are sent to AWS S3 Bucket with the help of python script. Our project has numerous applications like shock detector, surveillance system, intruder detection etc. This allows for surveillance of the perimeter in case of any disturbance in the form of an intruder or an accident or unexpected jolts. Also this system can be mounted on a vehicle in case of accidents. The system will detect sudden change in motion and may report the server with the images.

The project mainly hinges on two components, the detection of disturbance by the accelerometer and the capture of images using the camera. There is some degree of flexibility in setting the threshold of the accelerometer which makes this particular setup useful in many different use cases depending on the severity of the shock that is to be detected. Therefore this can be used to monitor disturbances ranging from minor ones like a slight movement by an animal to being able to detect when an accident has occurred which would require the device to only activate when a much higher value of acceleration is sensed.

The Pi is powered using a portable power-bank which can be recharged at regular intervals. The portability of the module is bottle-necked by both the power-bank size and the container on which the camera is mounted. Reducing the sizes of both can make the system even smaller and hence may lead to ease of use.

Figure 6: Flowchart of the System



The main flow is shown here in the flow-chart. First the accelerometer (ADXL345) reports the acceleration readings and if the readings cross a certain threshold value which is 0.5g in our case then the motion is registered as an event trigger.

As soon as the image is triggered, the motor (28BYJ-48) is rotated 45 degrees CW and takes the first image and subsequent images are captured by rotating the module (RKI-1598) 90 degrees CW. When the module is at 315 degrees, the module further rotates 45 degrees CW and arrives at 0 degrees position and then takes the fifth image. The remaining images are captured at 270, 180 and 90 degree positions by rotating CW- restoring the module's initial position as well as capturing 8 images in the process.

The images captured are stored first in the local Pi. These images are uploaded at one go to the remote server using the boto3 library. The required client verification details have already been configured in the device. Once these images are uploaded, these images are available at the website hosted using AWS. This website is kind of a repository for the images uploaded.

5 Test Results

The initial problem that we faced was in the connection of the pi camera module with the Pi. The number of pins in the camera module were not the same as that on the Pi since the camera module was meant to be compatible with a the Standard Raspberry Pi, not the Pi Zero W version we had. So we bought a 15 to 22 pin cable for our module which allowed us to finally be able to connect the camera module to our Pi Zero .

When the apparatus was all assembled and tested, we observed that the images were not being uploaded on the AWS Server on random occasions and the execution would be halted. We found out that the library functions used in the code were not efficient and causing the issue. Hence we looked for and found different functions that would be able to do the same task more dependably and and the issue was solved. So on the final testing after this issue also being resolved, the images were captured successfully according to the time count as per programmed and were also uploaded successfully.

6 Conclusions

After completing the testing, we found that the system worked as expected. The PI was getting hot after 30 minutes of constant use but after discussion with other groups, we found that it was a common issue. The internet speed was one of the main factors which made the system slow. Another bottle-neck was the 9v battery used to power the motor. If it drains out, we have to replace it and there is no system to detect its draining except to see whether the motor works or not. The power-supply used to supply power to the PI is also a bottle-neck since it determines how long the system can run autonomously.

By observing the opportunity for fine tuning in the threshold of the accelerometer we can modify our project to suit many different use cases depending on the requirement of the end user. The higher threshold can be set for detection of more severe shocks like accidents whereas the smaller threshold could be useful for detecting more sensitive motions like an intrusion or capturing images of smaller wildlife forms.

7 References

- https://github.com/pimoroni/adxl345-python/blob/master/adxl345.py
- https://42bots.com/tutorials/28byj-48-stepper-motor-with-uln2003-driver-a d-arduino-uno/
- https://www.youtube.com/watch?v=4fHL6BpJrC4
- https://tutorial.cytron.io/2017/08/16/raspberry-pi-zero-w-pi-camera-application/
- https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3.html