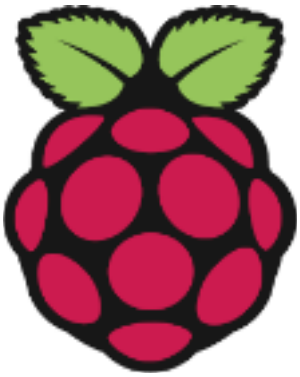
CAR CRASH DETECTION

USING RASPBERRY PI

EL 214 : ANALOG CIRCUITS

Prof. Rutu Parekh

EL Group 23



ACKNOWLEDGEMENTS

We would like to express our special thanks of gratitude to Prof. Rutu Parekh for providing us the the golden opportunity to do this wonderful project on the topic Car Crash Detection, which also helped us in doing a lot of Research and we came to know about many new things.

Next we would like to thank the TAs for providing us with the much required help and assistance whenever we approached them.

NAME	ID Gmail ID Webmail ID
Aashini Soni	201501004 aashini96@gmail.com 201501004@daiict.ac.in
Amarnath Karthi	201501005 97amarnathk@gmail.com 201501005@daiict.ac.in
Soumya Agrawal	201501006 soumya201415@gmail.com 201501006@daiict.ac.in
Neelanshi Varia	201501018 neelanshiv2@gmail.com 201501018@daiict.ac.in
Paramjeet Desai	201501024 desai.paramjeet3@gmail.com 201501024@daiict.ac.in
Nidhi Davawala	201501033 davawalanidhi@gmail.com 201501033@daiict.ac.in
Akshit Soni	201501085 akshitsoni0000@gmail.com 201501085@daiict.ac.in
Nikisha Patel	201501157 nikisha12345@gmail.com 201501157@daiict.ac.in
Kopal Bhat	201501210 kopalbhat97@gmail.com 201501210@daiict.ac.in
Darshan Makwana	201501425 darshanmakwana36@gmail.com 201501425@daiict.ac.in

PROJECT VISION AND MISSION

The frequency of **traffic collisions in India** is amongst the highest in the world. A National Crime Records Bureau (NCRB) report revealed that every year, more than 135,000 traffic collision-related deaths occur in India. Our Electronics project aims to collect pictures of the surrounding environment immediately after a collision has occurred, so as to provide a much needed critical data, from which one can analyse and thus in future prevent similar situations from happening again, analogous to a black-box in an aircraft.

PROJECT INTRODUCTION

1. Identify whether our vehicle has undergone a crash.
2. If it has, take 8 pictures of 2 rotations of the surrounding environment.
3. On taking the picture successfully, store the data in a protected location to be used later.

The assumptions that we have taken are :

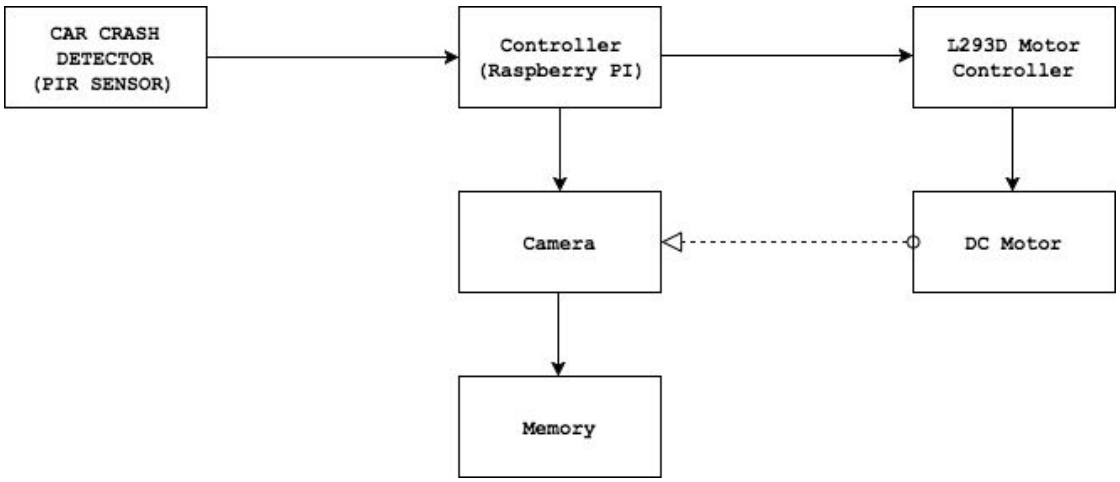
- The vehicle in question is stationary.
- The crash will occur from only one side.
- The hardware we are using does not get damaged during the crash.

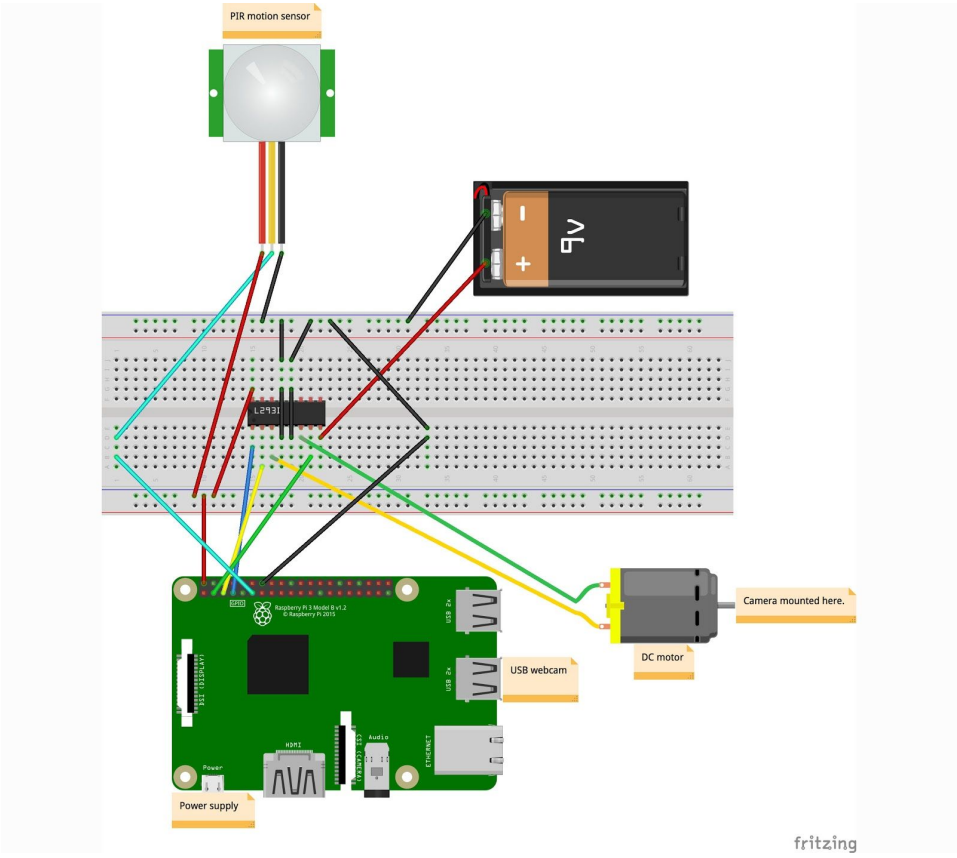
HARDWARE

The components used are:

- Raspberry Pi 3B
- PIR Motion Sensor
- DC motor
- Motor driver IC- L293D
- USB Web Camera
- Breadboard
- Battery
- Connecting wire

BLOCK / CIRCUIT DIAGRAM :





WORKING:

Stage 1 : The stationary car will be equipped with PIR sensor. An object in motion, approaching the car with force will approach(the sensitivity of the sensor is set such that it will detect if a stray object comes in close range).The PIR sensor itself has two slots in it, each slot is made of a special material that is sensitive to IR. When the sensor is idle, both slots detect the same amount of IR,the ambient amount radiated from the room or walls or outdoors. When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a *positive differential* change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected and sent as result to the Raspberry Pi.

Stage 2 : The Raspberry Pi will be programmed such that if an object is detected by the PIR sensor then a signal (high voltage) will be sent to the camera module.

Stage 3 : The camera will then start taking pictures of the crashing object and save them into memory for further use. A signal will also be sent to the motor on which the camera would be mounted. The motor will be programmed in a way that it rotates 90 degree per second. The camera will simultaneously take pictures every second. This procedure will be repeated two times, hence two rotations and we will get sixteen pictures of the car crash field.

COMPONENTS USED :

(1) PIR SENSOR : DETECTING CLOSE RANGE MOTION

PIR stands for Passive InfraRed. This motion sensor consists of a fresnel lens, an infrared detector, and supporting detection circuitry. The lens on the sensor focuses any infrared radiation present around it toward the infrared detector. Our bodies generate infrared heat, and as a result, this heat is picked up by the motion sensor. The sensor outputs a 5V signal for a period of one minute as soon as it detects the presence of a car/person. It offers a tentative range of detection of about 6-7 meters and is highly sensitive. When the PIR motion sensor detects a person, it outputs a 5V signal to the Raspberry Pi through its GPIO and we define what the Raspberry Pi should do as it detects an intruder through the python coding.

(2) RASPBERRY PI

The Raspberry Pi is programmed such that if the change is greater than a particular threshold acceleration (subject to the sensitivity of the accelerometer) then a signal (high voltage) will be sent to the camera module.

Programming Raspberry Pi:

Operating System : Raspbian

Language : Python

Libraries required :

- time
- smbus
- fswebcam
- RPI.GPIO

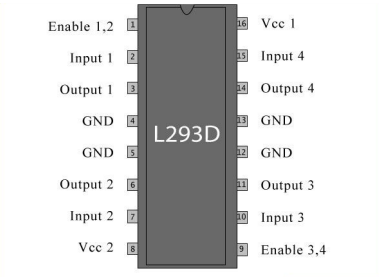
(3) USB-WEBCAM

As soon as the crash is detected, a signal to the camera module will be and the camera will then start taking pictures of the crashing object and save them into memory for further use. A signal will also be sent to the motor on which the camera would be mounted.This procedure will be repeated two times, hence two rotations and we will get sixteen pictures of the car crash field.

(4) MOTOR CONTROLLER L293-D CONTROLLING SPEED AND DIRECTION

A motor takes large amounts of currents for functioning. A Raspberry Pi cannot supply this amount of current. Hence the motor must run from a different power source like a 9V Cell. But the motor still needs to work according to the inputs of the Pi. Therefore we need to use a Motor Controller IC. A Motor Controller IC provides a high power output to a device (the

Motor) on the basis of the input from a low power device (Raspberry Pi). L293D is an IC which can control 2 different motors simultaneously.



Enable 1,2 - This pin is the enable control for motor 1. If input is High, the motor works. If not then the motor receives no current.

Pins 4,5,12,13 - Ground

Vcc1 - Is the voltage which is required to power the IC's logical components. Vcc should be about 5V.

Vcc2 - It is the voltage which is required by the motor. It can range from 0V to 36V.

Input1 - This controls the voltage of the Output 1 pin. If this is high then then Output 1 gets Vcc Volts. If it is low then Output 1 gets 0V.

Input2 - This controls the voltage of the Output 2 pin. If this is high then then Output 2 gets Vcc Volts. If it is low then Output 2 gets 0V.

Output1 and Output2 are the pins where we connect the motor terminals. Therefore the motor follows the following logic -

Enable1	Input1	Input2	Motor Rotation
HIGH	HIGH	HIGH	OFF
HIGH	HIGH	LOW	Clockwise
HIGH	LOW	HIGH	Anticlockwise
HIGH	LOW	LOW	OFF

LOW	X	X	OFF

X=Do Not Care

To control the speed of the DC Motor we use Raspberry PI GPIO's in-built pulse width modulation (pwm) function. This function generates pulses of time period 1/500 seconds. Therefore if we put the enable pin of the motor in a PWM, it essentially will turn the motor on and off 500 times every second. By controlling the duty cycle of these pulses, we can effectively increase the time for which the motor is on every 1/500 seconds, thus changing the speed. Therefore if we put the duty cycle of the PWM as 75%, we change the speed to ¾ of the maximum value.

The following is the python code -

```
pwm=GPIO.PWM(04,100) #configuring Enable pin means GPIO-04 for PWM
pwm.start(50) #starting it with 50% dutycycle
pwm.ChangeDutyCycle(80) #changing the duty cycle to 80%
```

In our case, the motor did not work under 15% duty cycle due to the weight of the camera.

RESULTS AND CONCLUSIONS

Whenever the PIR sensor detects an object coming very close to it, it signals the PI. The PI,on receiving the signal, activates the motor controller to rotate and stop at various angles. At each stop point, the PI takes the photo of the surroundings and then saves it in the memory. These photos can later be accessed for monitoring purposes.

LIMITATIONS

Our working model has the following limitations -

- It is assumed that the collision does not damage the camera or the PIR sensor. Incase of a damage the circuit might not work. Thus our model can safely detect only minor

collisions but not any major collisions.

- As we have used the PIR sensor, it does not detect the collision directly. It assumes that if an object comes very near to the vehicle then it’s a collision. But there are a few exceptions to this logic. For example if some human or an animal comes near to the vehicle, there is no actual collision, but still the sensor detects the collision and acts accordingly.
- As we have used a low power DC Motor for the camera movement, we cannot take photos at precise angles. Thus the angles on which the photos are taken depend on several factors like the weight of the camera component, the strain and stiffness of the camera cable and the weight of the cable.

SCOPE FOR IMPROVEMENT

Due to various constraints like time and budget, we created a relatively simple model for detecting a car crash. The following improvements can be made to the project to make it more utilitarian and of practical importance -

- Several PIR sensors and cameras can be added to the circuit in various different positions so that we can detect and monitor the car crash at different positions.
- A more robust Servo motor can be used instead of the DC Motor, to make the programming and implementation more easier and precise.
- A software feature can be added which uploads the pictures post collision to the cloud so that it can be accessed easily on the internet by police, hospital and other stakeholders.

- Apart from a camera, a GSM Module can be added which on detection of a collision contacts the nearest hospital/ambulance.
- The camera can be programmed to detect the number plate of the colliding vehicle in case of a hit and run case.

-----</THE END>-----