Project Title

Front Collision avoidance using Ultrasonic sensor and Raspberry pi 3

Developed by: MSc. Shiva Agrawal

Place: Germany

Date: June 2018

Content

List of figures

Abbreviations

OS	Operating System
RC	Remote Controller
DC	Direct Current
API	Application Program Interface
PC	Personal Computer
LAN	Local Area Network
GPIO	General Purpose Input Output
VNC	Virtual Network Computing

1. Introduction

1.1. Project description

The aim of the project is to design and implement the front collision avoidance system which is generally used in cars. Ultrasonic sensor is used to measure the front-end distance between vehicle and the obstacle or another vehicle. Depending on the measured distance, if the obstacle is too close to the vehicle, then the control signal goes to stop the vehicle to avoid collision.

System uses raspberry pi 3 Model B as main controller and Ubuntu Mate OS for the development environment. For the project, RC toy Car is used because it can replicate the motion dynamics of the actual car. Ultrasonic sensor acts as input to the system and the vehicle motion control acts as output of the system.

In the project, raspberry pi 3 is flashed with Ubuntu mate Operating system which uses 16 GB micro SD card for the memory. DC motors of the vehicle are connected to Motor Shield or Motor driver circuit which is used for high current requirement of the motors and for switching between reverse and forward vehicle motion. Further this motor shield is connected to Raspberry pi to provide the control signal to drive motors. Ultrasonic sensor HC-SR04 is connected to raspberry-pi directly to GPIO pins.

The software (drivers and application layer) of the project is implemented using open source IDE Geany and C++ embedded programming. The API wiringPi.h is used for GPIO control of raspberry pi.

To access remotely the raspberry pi and therefore vehicle from PC, the VNC server is install on the raspberry pi. PC can run the client software with raspberry pi IP address and port number to access it. For such connectivity, both raspberry pi and PC must be in the same network.

1.2. Motivation

This project is one of the basic requirements towards the development of the fully autonomous driving vehicle. This project helps to acquire the hardware and software experience to get acquaint with raspberry pi 3 model B, Ultrasonic sensor, Linux – Ubuntu Mate OS, Motor driver, GPIO programming, server-client connectivity and hardware interfacing.

1.3. Outline

- Chapter 1: Project description, Motivation and outline of the project report
- Chapter 2: Hardware and software interfacing and development.
- Chapter 3: Tests and results.
- Chapter 4: Conclusion
- Chapter 5: References to other websites, technical papers referred, etc.

2. Project development

2.1. Hardware components

Main hardware components used for the project include RC car, Raspberry pi 3 Model B, DC motor driver, power supplies, ultrasonic sensor and connectors (jumpers, wires, headers, etc.)

2.1.1. RC car

The RC car is normal toy car which works on 2.4 Ghz. It is readily available online. For the project, only the car is used. The internal circuit is removed, and motor shield is connected to the wires of the wheel DC motors and steer DC motor. Such RC car from one online website (RC Rock Crawler, 2018) is shown below in Figure 1 - RC Car.

Figure 1 - RC Car

2.1.2. Raspberry pi 3 Model B

(Raspberry pi 3 model B, 2018), It is a compact hardware, but powerful development board comprised of 1.4GHz 64-bit quad-core processor, dual-band wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and Power-over-Ethernet support. It is shown below in Figure 2 - Raspberry pi 3 Model B+

The other specifications are (Raspberry pi 3 model B, 2018):

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE

- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

In this project, raspberry-pi is the core component.

Figure 2 - Raspberry pi 3 Model B+

2.1.3. Motor Shield/driver

This driver component is used for mainly two purpose, to drive the motors with sufficient current and to have direction control using the available L293D ICs. This driver circuit from Adafruit (Adafruit - motor- shield, 2018), can drive up to 4 bi-directional DC motors. The motor shield is shown in Figure 3 - Motor Driver/Shield. This main ICs on the circuit are L293D, which is used for bi-directional motor control and 74HC595N, a shift register (serial to parallel) used to get the control signals to the driver board. The flow of the signals is described in Figure 4 - Signal flow from Raspberry pi to motor.

The details about the working of these ICs is out of scope of this report. But for the details one can refer to following links or online other websites.

1. For L293D and H bridge circuit use for bidirectional motor control - (h-bridge-motor-control-circuit-using-l293d-ic, 2018)
2. For working and programming basics of 74HC595 shift register - (74hc595-shift-register, 2018)

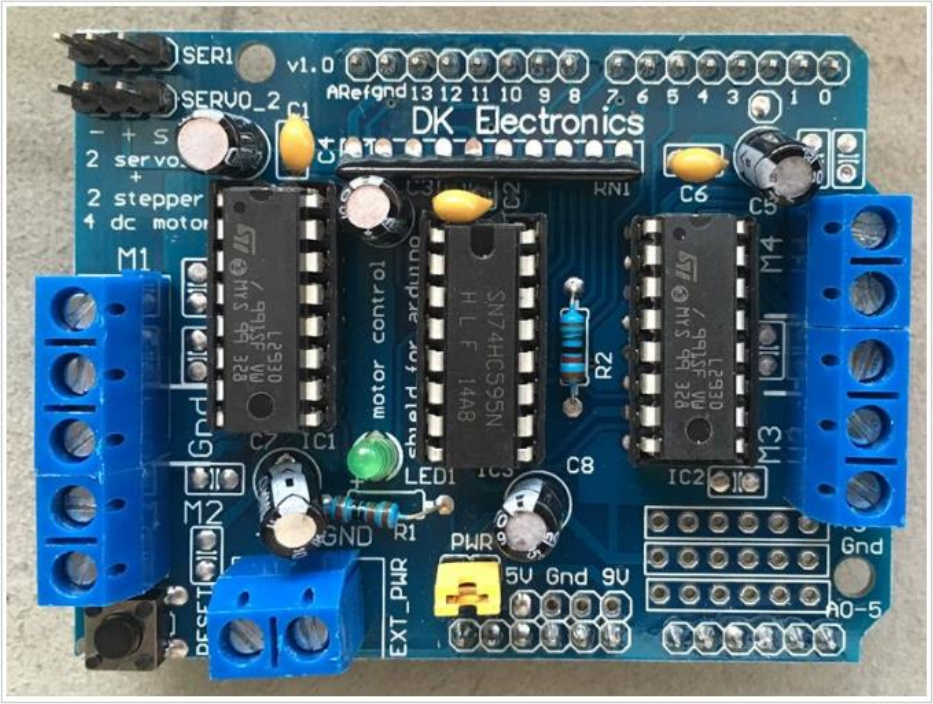


Figure 3 - Motor Driver/Shield

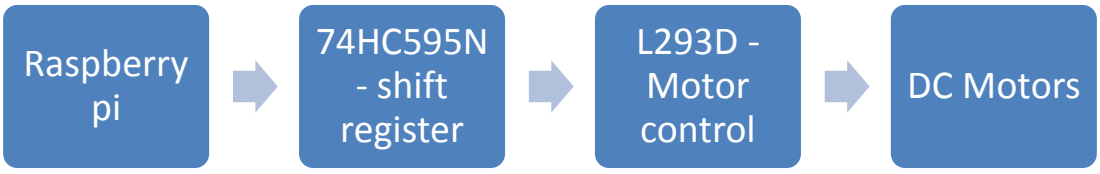


Figure 4 - Signal flow from Raspberry pi to motor

2.1.4. Power supply modules

Two power supply modules are required

1. For the Motor shield circuit and DC motors: This is rechargeable 4.8 V power supply module which came in handy with RC car.
2. For Raspberry – pi: As this board needs 5 V and 700 mAh supply, for the project rechargeable power bank is used which connects directly via USB port of the board (shown in Figure 5 - Power bank and USB cable).

Figure 5 - Power bank and USB cable

2.1.5. Ultrasonic sensor [HC-SR04]

(ultrasonic sensor, 2018) The HC-SR04 Ultrasonic sensor has four pins: ground (GND), Echo Pulse Output (ECHO), Trigger Pulse Input (TRIG), and 5V Supply (Vcc) as shown in Figure 6 - Ultrasonic sensor [HC-SR04] (ultrasonic-sensor-working-pinout, 2018). GND and VCC pins are connected to GND and 5V of raspberry pi. Further raspberry Pi send an input signal to TRIG (10 microseconds), which triggers the sensor to send an ultrasonic pulse. The pulse waves bounce off any nearby objects and some are reflected to the sensor. The sensor detects these return waves and measures the time between the trigger and returned pulse, and then sends a 5V signal on the ECHO pin.

ECHO will be “low” (0V) until the sensor is triggered when it receives the echo pulse. Once a return pulse has been located ECHO is set “high” (5V) for the duration of that pulse. Pulse duration is the full time between the sensor outputting an ultrasonic pulse, and the return pulse being detected by the sensor receiver. This duration is measured and then distance to the object is calculated in cm as per below formula.

$$34300 = \frac{Distance}{Time/2}$$

$$17150 = \frac{Distance}{Time}$$

$$17150 \times Time = Distance$$

NOTE: The sensor output signal (ECHO) on the HC-SR04 is rated at 5V. However, the input pin on the Raspberry Pi GPIO is rated at 3.3V. Sending a 5V signal into that unprotected 3.3V input port could damage your GPIO pins. Hence voltage divider circuit is used between the echo pin of the ultrasonic sensor and raspberry pi as shown in Figure 7 - Interface: Ultrasonic sensor with pi.

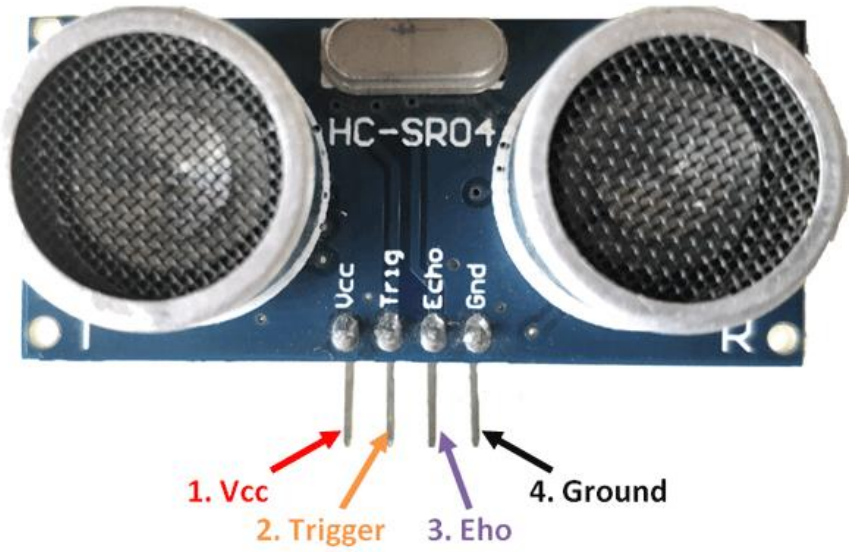


Figure 6 - Ultrasonic sensor [HC-SR04]

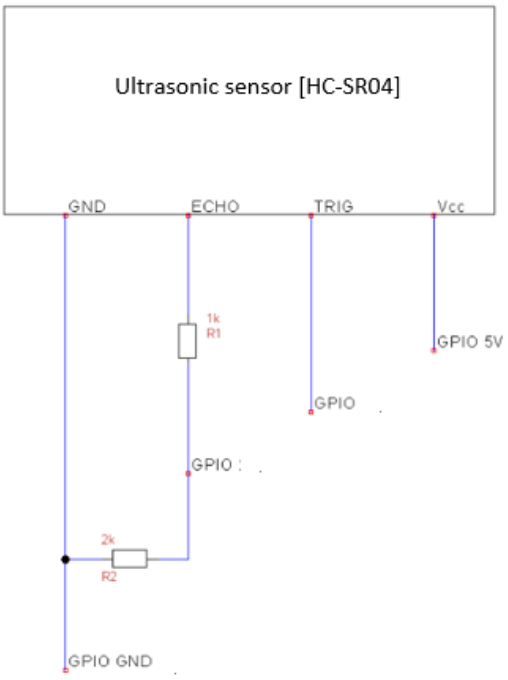


Figure 7 - Interface: Ultrasonic sensor with pi

2.1.6. RC car mounted with all the components

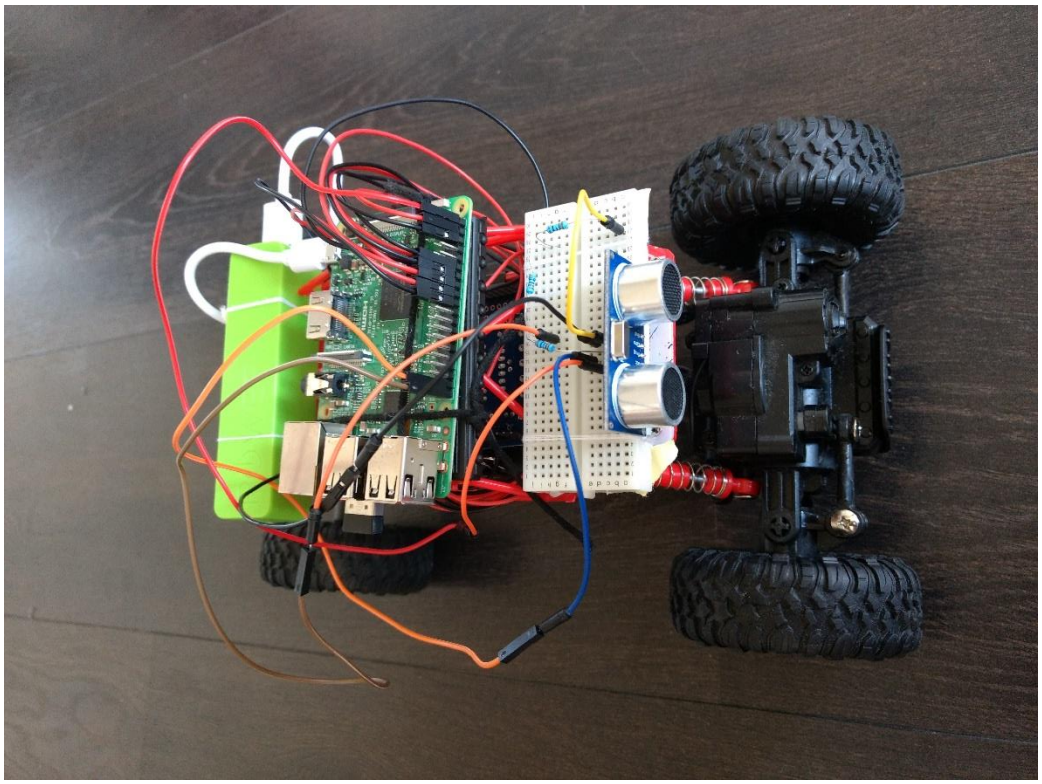


Figure 8 - RC car with mounted hardware

2.2. Software components/environment

The project is developed in Ubuntu Mate OS which is installed on Raspberry pi 3 Model B. The IDE used for the programming is Geany and C/C++ API wiring.h is used for GPIO control of pi board. Also VNC server client software are used for accessing raspberry pi remote using PC for remote control of the vehicle.

2.2.1. Installation: Ubuntu Mate 16.04 on raspberry-pi

Raspberry pi is compatible with many different operating systems. Raspbian is the official operating system for all models of the raspberry pi developed by the foundation community. But other OS like Ubuntu mate, snappy ubuntu core, windows 10 IoT core, etc. can be installed.

To install the raspberry pi, take 16 GB or higher size micro SD card, insert it into Laptop. Download the image file for the Ubuntu mate for Raspberry pi (ARMv7) from official website (Ubuntu Mate download, 2018). Use the flashing software like etcher (etcher, 2018) and create the bootable microSD card. For detail steps please refer online material.

Once the microSD is flashed, remove it from laptop and insert it into raspberry pi. Plug in other accessories like mouse, keyboard, display, etc. and then insert the power supply.

In some seconds, the desktop version of the Ubuntu is ready to use on raspberry pi. For direct use, it is advisable to connect display with board.

Note that the above steps are same even if other OS like Raspbian, to be installed on raspberry pi.

2.2.2. Installation: Geany IDE

Geany (Geany IDE Home page, 2018) is a text editor using the GTK+ toolkit with basic features of an integrated development environment. It was developed to provide a small and fast IDE, which has only a few dependencies from other packages. It supports many filetypes like C, C++, Python, Java, etc.

The complete software (driver and application) is developed in Geany IDE for this project. The IDE is installed on raspberry pi.

For installation, open the terminal in ubuntu mate with admin rights and enter the below command

```
sudo apt-get install geany
```

at prompt, type in the sudo password of the system and Geany IDE get install in few minutes.

2.2.3. C/C++ API: wiringPi.h

This is the API for C++ used for GPIO control of raspberry-pi. Further information can be found at (wiringpi, 2018).

Details about how to use with raspberry-pi and Geany IDE are available at (raspberrypi gpio, 2018). The pin convention of raspberry pi 3 model B used for wiringPi.h are described in Figure 9- Raspberry pi 3 model B pins.

Wedge Silk	Python (BCM)	WiringPi GPIO	Name	P1 Pin Number		Name	WiringPi GPIO	Python (BCM)	Wedge Silk
			3.3v DC Power	1	2	5v DC Power			
SDA		8	GPIO02 (SDA1, I2C)	3	4	5v DC Power			
SCL		9	GPIO03 (SCL1, I2C)	5	6	Ground			
G4	4	7	GPIO04 (GPIO_GCLK)	7	8	GPIO14 (TXD0)	15		TXO
			Ground	9	10	GPIO15 (RXD0)	16		RXI
G17	17	0	GPIO17 (GPIO_GEN0)	11	12	GPIO18 (GPIO_GEN1)	1	18	G18
G27	27	2	GPIO27 (GPIO_GEN2)	13	14	Ground			
G22	22	3	GPIO22 (GPIO_GEN3)	15	16	GPIO23 (GPIO_GEN4)	4	23	G23
			3.3v DC Power	17	18	GPIO24 (GPIO_GEN5)	5	24	G24
MOSI		12	GPIO10 (SPI_MOSI)	19	20	Ground			
MISO		13	GPIO09 (SPI_MISO)	21	22	GPIO25 (GPIO_GEN6)	6	25	G25
		(no worky 14)	GPIO11 (SPI_CLK)	23	24	GPIO08 (SPI_CE0_N)	10		CD0
			Ground	25	26	GPIO07 (SPI_CE1_N)	11		CE1
IDSD		30	ID_SD (I2C ID EEPROM)	27	28	ID_SC (I2C ID EEPROM)	31		IDSC
G05	5	21	GPIO05	29	30	Ground			
G6	6	22	GPIO06	31	32	GPIO12	26	12	G12
G13	13	23	GPIO13	33	34	Ground			
G19	19	24	GPIO19	35	36	GPIO16	27	16	G16
G26	26	25	GPIO26	37	38	GPIO20	28	20	G20
			Ground	39	40	GPIO21	29	21	G21

Figure 9- Raspberry pi 3 model B pins

2.2.4. VNC (Virtual Network computing): Server and client

2.2.4.1. VNC server setup in raspberry pi

The installation and setup steps are available at (configure vnc server for ubuntu Mate, 2018). Once the setup is ready, the VNC server can be configured to start by default when raspberry pi is powered on. For the project the port number for the server is set to 1.

2.2.4.2. VNC client at PC

The client application is required to run in order to access the raspberry pi OS remotely from any PC which is in the same network as pi. The application can be installed from official webpage of realvnc viewer (realvnc viewer, 2018). The client application is shown in Figure 10 - VNC viewer (Client application). The IP address of the raspberry pi and the port number of the server is required to get the remote control of raspberry pi board.

If the pi is powered and the server is started correctly, then the remote access gets activated and looks like as shown in Figure 11 - Remote view of Raspberry pi OS at PC (client side).

Now there is no need of any display, mouse or keyboard to connect to pi board. Everything can be done from the remote PC only.

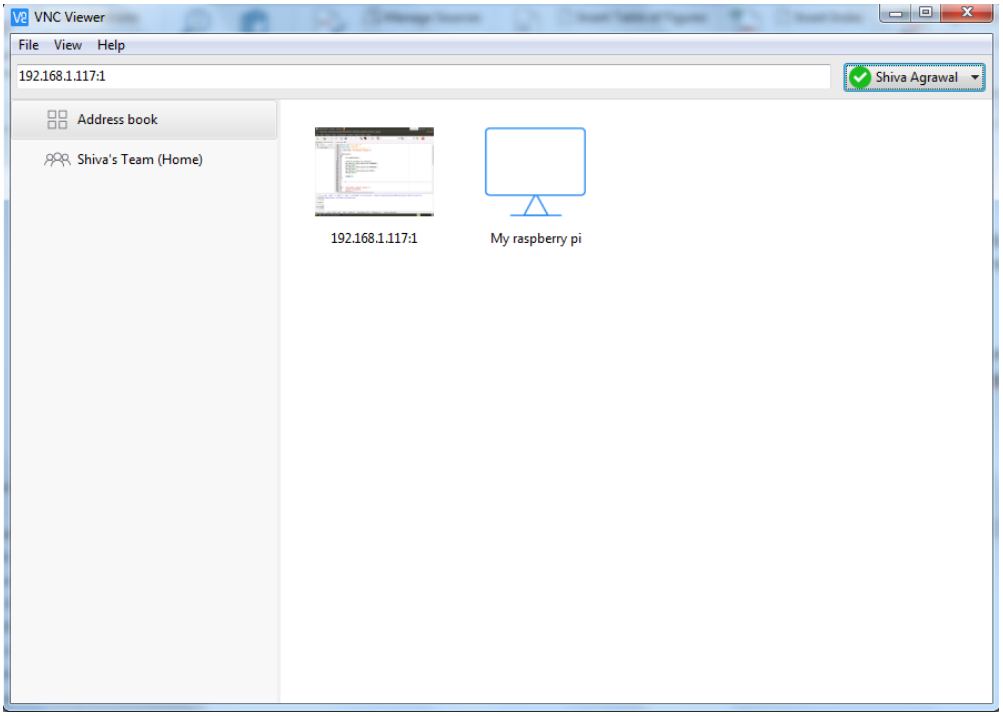


Figure 10 - VNC viewer (Client application)

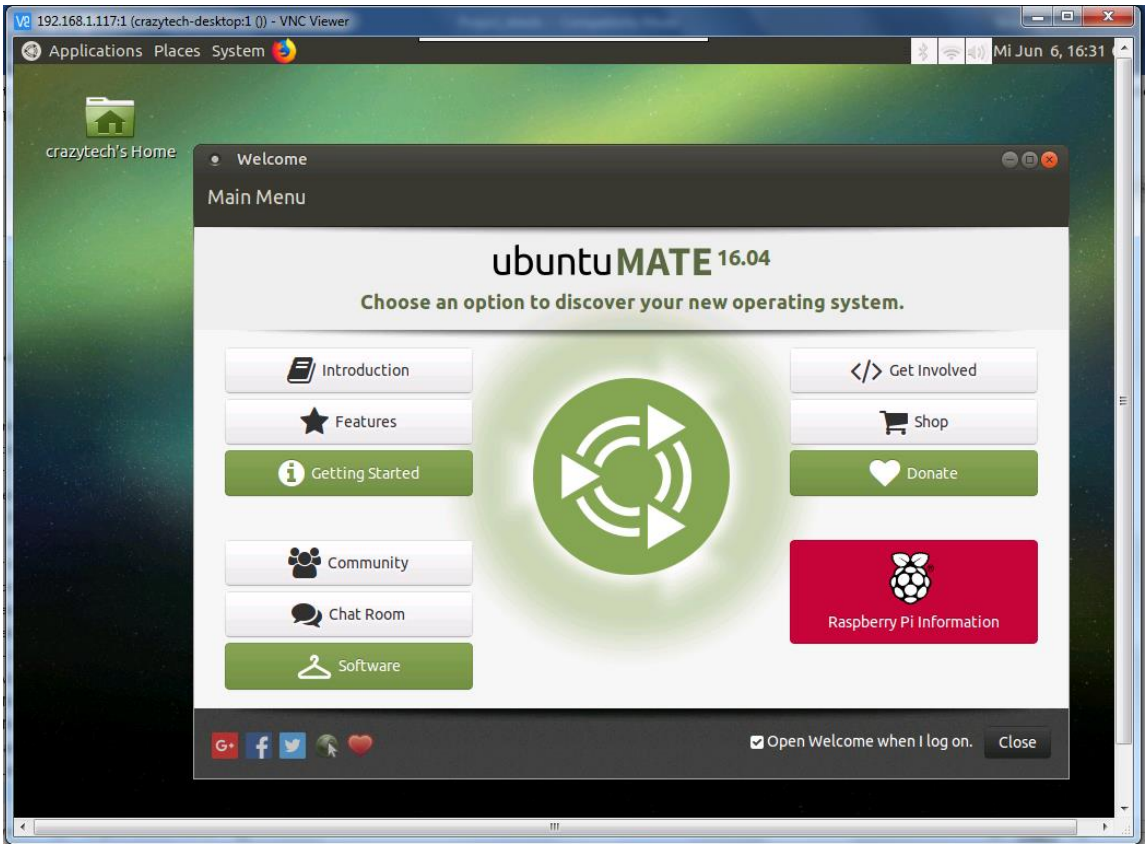
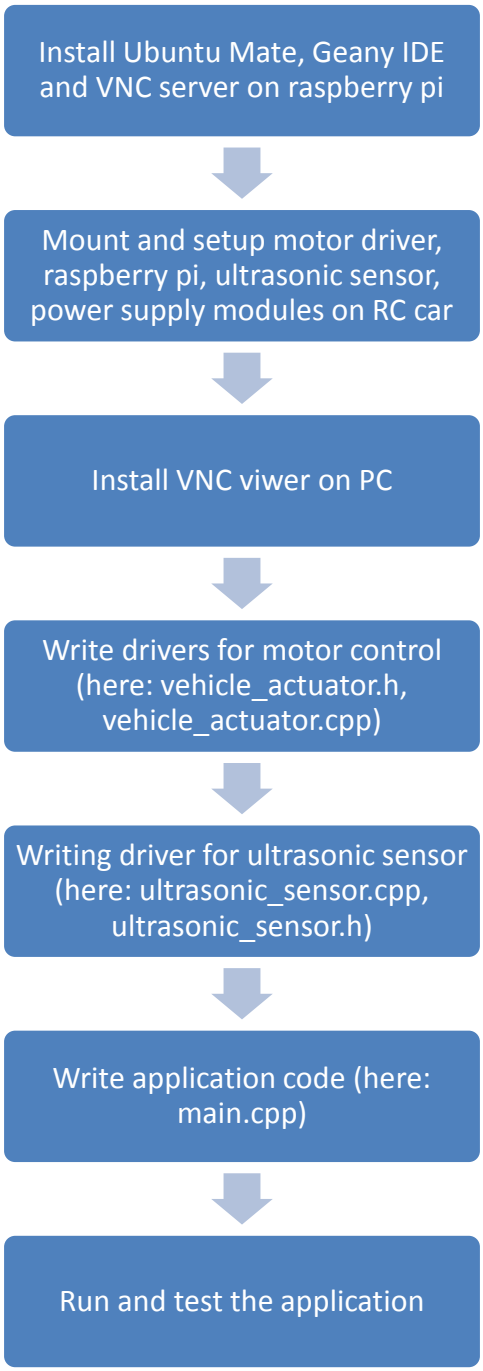


Figure 11 - Remote view of Raspberry pi OS at PC (client side)

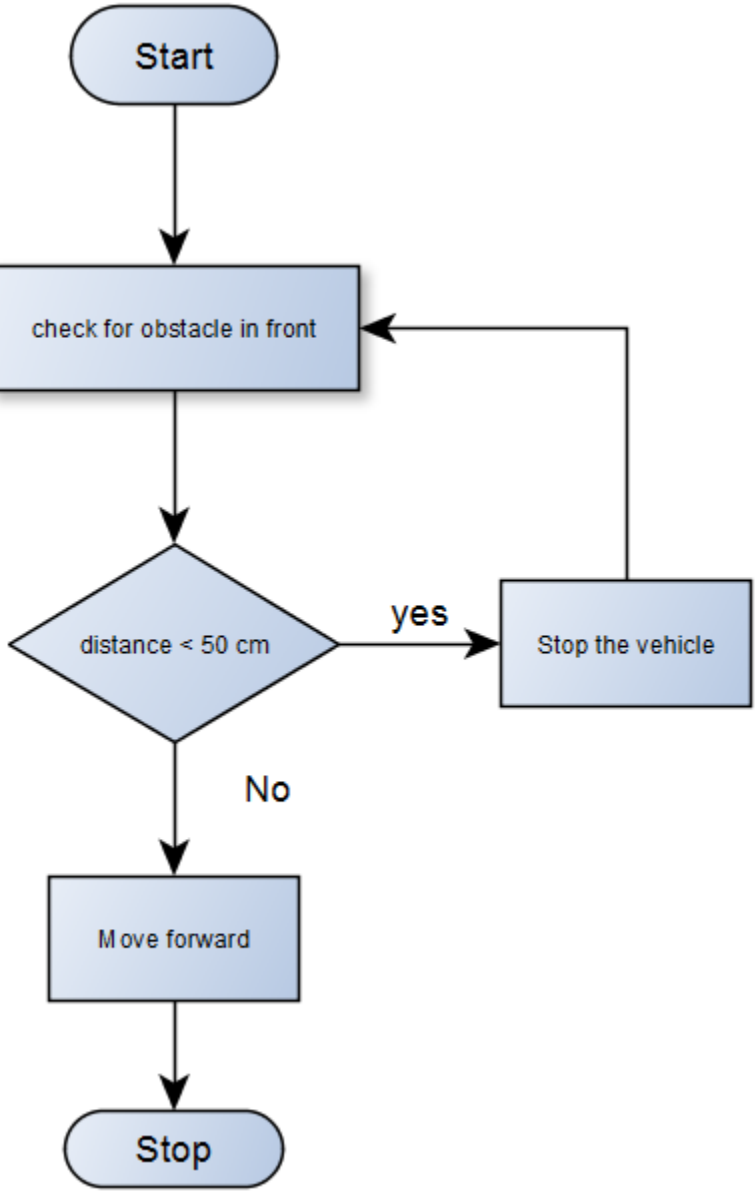
2.3. Implementation

2.3.1. Work flow

In this section, the steps for the development of the project are described in form of flow chart.



2.3.2. Function algorithm



2.3.3. Source code

The source code is not included in this report. It is available separately in the repository of the same project under the folder: **src**

2.3.4. Other relevant documents

The other project related documents like, datasheets, schematics, instructions, etc. are available under the same project in the separate folder: **doc**

3. Test and results

The test file is included under the same project in the separate folder: **test**

4. Conclusion

Front Collision avoidance for the vehicle using ultrasonic sensor is implemented with raspberry pi and ubuntu Mate OS. This project will act as base to develop further projects in the direction of autonomous driving.

5. References

1. *74hc595-shift-register*. (2018, 06 06). Retrieved from prostock.com: https://protostack.com.au/2010/05/introduction-to-74hc595-shift-register-controlling-16-leds/

2. *Adafruit - motor- shield*. (2018, 06 06). Retrieved from Adafruit: https://learn.adafruit.com/adafruit-motor-shield

3. *configure vnc server for ubuntu Mate*. (2018, 06 06). Retrieved from linuxeveryday.com: http://www.linuxeveryday.com/2017/09/install-configure-vnc-server-ubuntu-mate

4. *etcher*. (2018, 06 06). Retrieved from etcher.io: https://etcher.io/

5. *Geany IDE Home page*. (2018, 06 06). Retrieved from geany.org: https://www.geany.org/Main/HomePage

6. *h-bridge-motor-control-circuit-using-l293d-ic*. (2018, 06 06). Retrieved from www.elprocus.com: https://www.elprocus.com/h-bridge-motor-control-circuit-using-l293d-ic/

7. *Raspberry pi 3 model B*. (2018, 6 6). Retrieved from Raspberry pi: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

8. *raspberrypi gpio*. (2018, 06 06). Retrieved from sparkfun: https://learn.sparkfun.com/tutorials/raspberry-gpio

9. *RC Rock Crawler*. (2018, 6 6). Retrieved from Amazon: https://www.amazon.de/Rock-Crawler-CrossRace-Ferngesteuertes-Auto/dp/B075R29MM4/ref=sr_1_2?ie=UTF8&qid=1528283421&sr=8-2&keywords=RC+car+crawler

10. *realvnc viewer*. (2018, 06 06). Retrieved from realvnc.com: https://www.realvnc.com/en/connect/download/viewer/

11. *Ubuntu Mate download*. (2018, 06 06). Retrieved from Ubuntu Mate: https://ubuntu-mate.org/download/.

12. *ultrasonic sensor*. (2018, 06 06). Retrieved from modmpi.com: https://www.modmypi.com/blog/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi

13. *ultrasonic-sensor-working-pinout*. (2018, 06 06). Retrieved from components101.com: https://components101.com/ultrasonic-sensor-working-pinout-datasheet

14. *wiringpi*. (2018, 06 06). Retrieved from wiringpi.com: http://wiringpi.com/