Project Title:

Smart vehicle with Advanced Driver-Assistance System (ADAS)

Supervised By:

Dr. Mohamed El-Dakroury

Students:

1. **Mohamed Raafat Afifi Afifi**
2. **Abdullah Ezzat Sawey**
3. **Ahmed Hatem Abdelhamid**
4. **Haidy Mahamod Ahmed Ibrahim**
5. **Raghda Hammad Amr**
6. **Salma Sayed Shaban Refaie**

ACKNOWLEDGMENT

We would like to thank **Dr. Mohamed El-Dakroury** our supervisor for encouraging us always and supporting us during the whole time of the project. He was the 7th member of the team.

We would also like to show our deep appreciation to our **professors** who helped us during our educational journey

Table of figure:

Table of Contents

1. Introduction

According to the National Highway Traffic Safety Administration (NHTSA), “They lost 35,092 people in crashes on U.S. roadways during 2015”. An analysis revealed that about 94% of those accidents were caused by human error, and the rest by the environment and mechanical failures. Road traffic accidents, with an estimated 37,133 lives lost on U.S. roads in 2017, are a leading cause of death. So, we are facing big problem that we have to solve using technology and science after god’s will. Currently, the way smart vehicles operate is that they have sensors to sense surrounding objects and information about them all over the car. These sensors, appear to be limited in precision past very short distances. And because cars travel at such rapid speeds and have to continuously identify changing information in their surroundings, sensors often do not have sufficient range to identify the information needed for complicated decisions (crash detection at intersections or highways, avoidance of erratic drivers, traffic decisions, etc.).

2. Problem definition

Basically, the world now depends on vehicles as a means of transportation under human leadership, which led to the occurrence of traffic accidents, a matter that all countries suffer from, the most important of which is the sudden factor, weather conditions and dangerous roads that humans cannot deal with. Therefore, to enhance transportation safety and reduce accidents, some companies have turned to Self-driving vehicles by using sensors to sense and driving safely, despite that, it is limited and suffers from short range and inability to deal with emergency situations that occur suddenly and some problems that require human intervention to solve and it’s still under development until now. So, it’s necessary to develop a system that controls the movement of vehicles which help both self-driving and human manual driving. So, it’s very powerful to improve the safety and save human’s life.

3. Proposed Solution

We will develop an ADAS System that will contain perception, planning and V2X modules,
These modules will give the system a lot of features such as:

- Map based navigation and localization.
- Optimization of computer vision using FPGA
- Pedestrian avoidance
- V2C

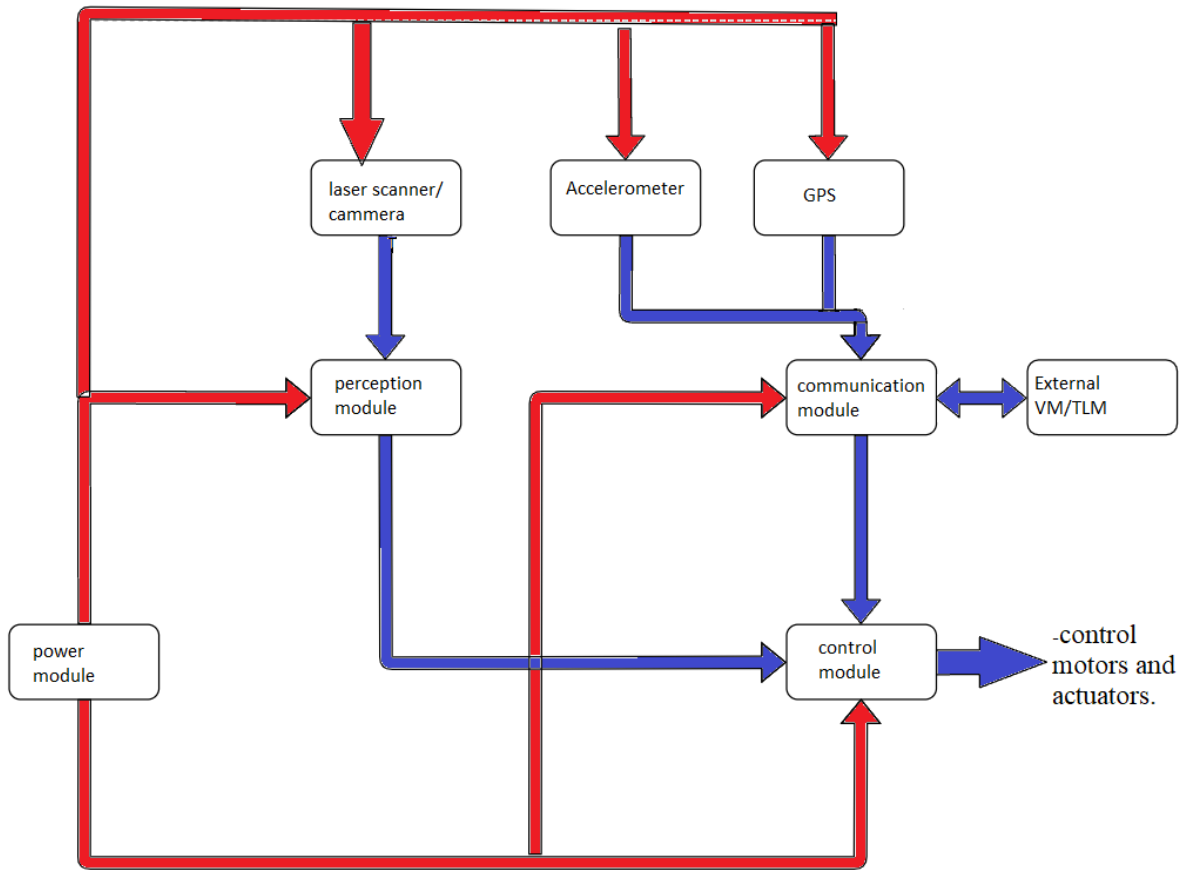


Figure 1Block Diagram of the system

4. System Architecture & modules:

Our developed system consists of four modules:

- Perception
- Communication
- Control
- Vehicle

4.1 Perception Module:

4.1.1Raspberry pi

Figure 2 Raspberry Pi

Raspberry Pi is a small single board computer commonly known as an SBC, is a computer built on only one circuit board which integrates the CPU and GPU in a single integrated circuit, with the RAM, USB ports, and other components soldered onto the board for an all-in-one package. It has an SD card slot you can use to house your operating system and files. The Raspberry Pi is small, doesn't use much power, and is relatively inexpensive. It is also popularly used for real time Image/Video Processing, IoT based applications and Robotics applications.

Raspberry pi operating system

Official operating system Raspbian OS is freely used. For usage with Raspberry Pi, this OS is effectively optimised. The GUI for Raspbian has tools for web browsing, Python programming, office, gaming, etc. To store the operating system, we have to utilise an SD card (minimum 8 GB is advised) (operating System).

Raspberry Pi OS has a desktop environment, PIXEL, based on LXDE, which looks similar to many common desktops, such as macOS and Microsoft Windows. The desktop has a background image. A menu bar is positioned at the top and contains an application menu and shortcuts to a web browser (Chromium), file manager, and terminal. The other end of the menu bar shows a Bluetooth menu, Wi-Fi menu, volume control, and clock. The desktop can also be changed from its default appearance, such as repositioning the menu bar.

Package management

Packages can be installed via APT, the Recommended Software app, and by using the Add/Remove Software tool, a GUI wrapper for APT.

Raspberry pi 3 model B

1. Features

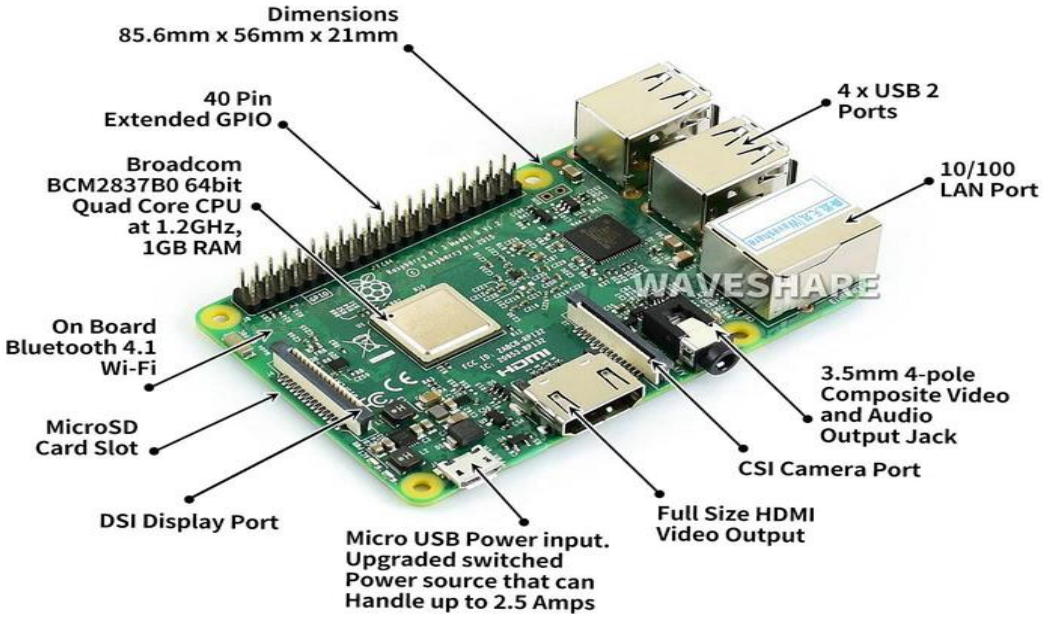


Figure 3 Raspberry Pi PINS

Features	Raspberry Pi 3 Model B
SoC	BCM2837
CPU	Quad Cortex A53
Operating Freq.	1.2 GHz
RAM	1 GB SDRAM
GPU	400 MHz Videocore IV
Storage	micro-SD
Ethernet	Yes
Wireless	No

- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board.
- 100 Base Ethernet.
- 40-pin extended GPIO.
- 4 USB 2 ports.
- 4 Pole stereo output and composite video port.
- Full size HDMI.
- Maximum power 2.5A/ 5v

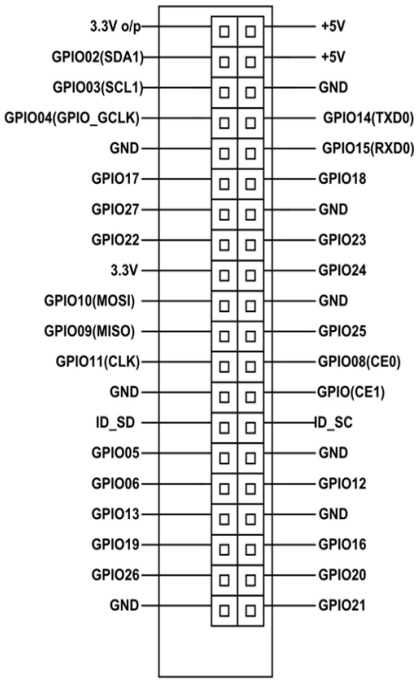
HDMI (High-Definition Multimedia Interface): It is used for transmitting uncompressed video or digital audio data to the Computer Monitor, Digital TV, etc. Generally, this HDMI port helps to connect Raspberry Pi to the Digital television.

CSI Camera Interface: CSI (Camera Serial Interface) interface provides a connection in between Broadcom Processor and Pi camera. This interface provides electrical connections between two devices.

DSI Display Interface: DSI (Display Serial Interface) Display Interface is used for connecting LCD to the Raspberry Pi using 15-pin ribbon cable. DSI provides fast High-resolution display interface specifically used for sending video data directly from GPU to the LCD display.

2. Performance

32kB Level 1 and 512kB Level 2 cache memory on ARM Cortex-A53 processor cores operating at 1.2GHz making the device about 50% faster than the Raspberry Pi 2. A Video Core IV graphics processor runs at 400MHz, and is linked to a 1GB LPDDR2 memory module on the rear of the board.



3. Pinout

- UART Interface
  (RXD, TXD) [(GPIO15,GPIO14)]
- SPI Interface (MOSI, MISO, CLK,CE) x 2 [SPI0-(GPIO10 ,GPIO9, GPIO11 ,GPIO8)] [SPI1--(GPIO20 ,GPIO19, GPIO21 ,GPIO7)]
- TWI Interface (SDA, SCL) x 2 [(GPIO2, GPIO3)] [(ID_SD,ID_SC)]

- Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19

4. Application

After ADRUINO, the RASPBERRY PI platform is the most popular. Additionally, it is an open-source platform where one may get a wealth of relevant information, enabling customization of the system based on demand.

The fact that the system processing is so intensive led to the choice of this microcontroller and development board above others. Since the majority of Arduino boards have clock speeds around 100MHz, they can only accomplish tasks that fall within their scope. High end programs for applications like weather stations, cloud servers, gaming consoles, etc. cannot be processed by them. The RASPBERRY PI can carry out all those sophisticated tasks thanks to its 1.2GHz processor and 1 GB of RAM.

4.1.2 LIDAR sensor

A laser scanner (LIDAR) system determines how long it takes light beams to strike an item or surface and bounce back to the laser scanner. Utilizing the speed of light, the distance is then determined. This type of measurement is called "Time of Flight". It communicates with the raspberry pi using “UART” serial protocol. Send the required information on a message called “Scan “, which raspberry pi uses to serve the required map. The output data of the laser scanner (scan message) could be visualize using Rviz platform which Ros provide.

As modern vehicles have much more powers than previous ones, making them essentially super heroes. For features like adaptive cruise control, parking assistance, automated emergency braking, and blind spot monitoring, the majority of automobiles utilize a mix of cameras, ultrasonic sensors, and radar; they allow the cars to "see," but are constrained in terms of range and depth.

It stands for “Light Detection and Ranging” and describes a sensor technology that can create a map of the environment around it. LiDAR sensors send out infrared light, and measure the time it takes for the light to bounce off an object and back to the sensor, creating a three-dimensional map.

Figure 4 LIDAR sensor

1. Types

- **Time of Flight (ToF)**

It is the most common form of LiDAR on vehicles that map their surroundings by measuring pulses or photons of light that it sends out and bounce back. It has a 360-degree range, allowing for a single device to do the job.

- **frequency-modulated continuous-wave (FMCW)**

It sends out a continuous stream of light rather than pulses of light, to map its surroundings. This form of LiDAR has a limited field of view. Used in vehicles with multiple LIDAR sensors.

2. Comparison between RADAR and LIDAR

LiDAR uses lasers with a much lower wavelength than the radio waves used by RADAR. Thanks to this, LiDAR has better accuracy and precision, which allows it to detect smaller objects that are vital for self-driving features like pedestrian, cyclist, and animal detection, in more detail, and create 3D images based on the high-resolution image it creates.

On the other hand, RADAR is much more robust than LiDAR with a lower starting price. While you don't get as many details, it works in worse operating conditions and has a wider range than LiDAR and it is used to detect objects that surround a car, and can determine how far away they are and how fast they're moving. This is why automakers use radar for parking sensors, blind spot monitors, and adaptive cruise control

Product used: 8 Meter Range Lidar Scanner Sensor LiDAR Positioning Navigation Obstacle Avoidance 360 Deg. 3i Lidar Delta 2A

3. Product parameters

- Range: 0.13m – 8m (white wall)
- Sweep frequency: 6.2Hz(4-10Hz)
- Laser power: 3mW (max)
- Laser wavelength: 780nm
- Angular range: 0-360 degree
- Distance resolution < 0.5mm

Figure 5 LIDAR sensor_2

- Angular resolution ≤ 1 degree
- Sampling rate: 5K/s
- Voltage: 5Vdc
- Power consumption: 1.5 W
- Current: 500 mA
- Size: 107 * 76 * 53 mm

4. Communication interface: RS232 (TTL) 3.3V-TTL

- 8-bit data, 1 bit stop bits, no check.
- Baud rate: 230400
- Output low level: < 0.4
- Output high level: 2.9-3.5

5. Working environment

- Work Environment Temperature: 0 ° C – 45 ° C
- Work Environment humidity: < 90%
- Work Levelness 0 ° – 1 °
- Work Environment illumination<1000Lux

6. Application environment:

- 1. Robot SLAM positioning and obstacle avoidance
- 2. fast and accurate surveying and mapping
- 3. robot ROS development
- 4. Game interaction.

4.2. Communication Module:

4.2.1 NodeMcu

NodeMCU is a development board and open-source Lua-based firmware that is specifically targeted for Internet of Things (IoT) applications. It has hardware based on the ESP-12 module and firmware that runs on Espressif Systems' ESP8266 Wi-Fi SoC. You may use the Arduino IDE or the straightforward and effective LUA programming language to program the ESP8266 WiFi module.

You may transform your ESP8266 into a web server, create input/output pins according to your needs precisely like Arduino, make a WiFi connection, and do much more with only a few lines of code. The nodeMCU Dev board allows immediate flashing from a USB port thanks to its USB-TTL. It combines the advantages of a microcontroller with a WIFI access point. The NodeMCU is an incredibly powerful tool for wireless networking because to these characteristics. It can function as a station or access point, run a web server, or connect to the internet to download or upload data.

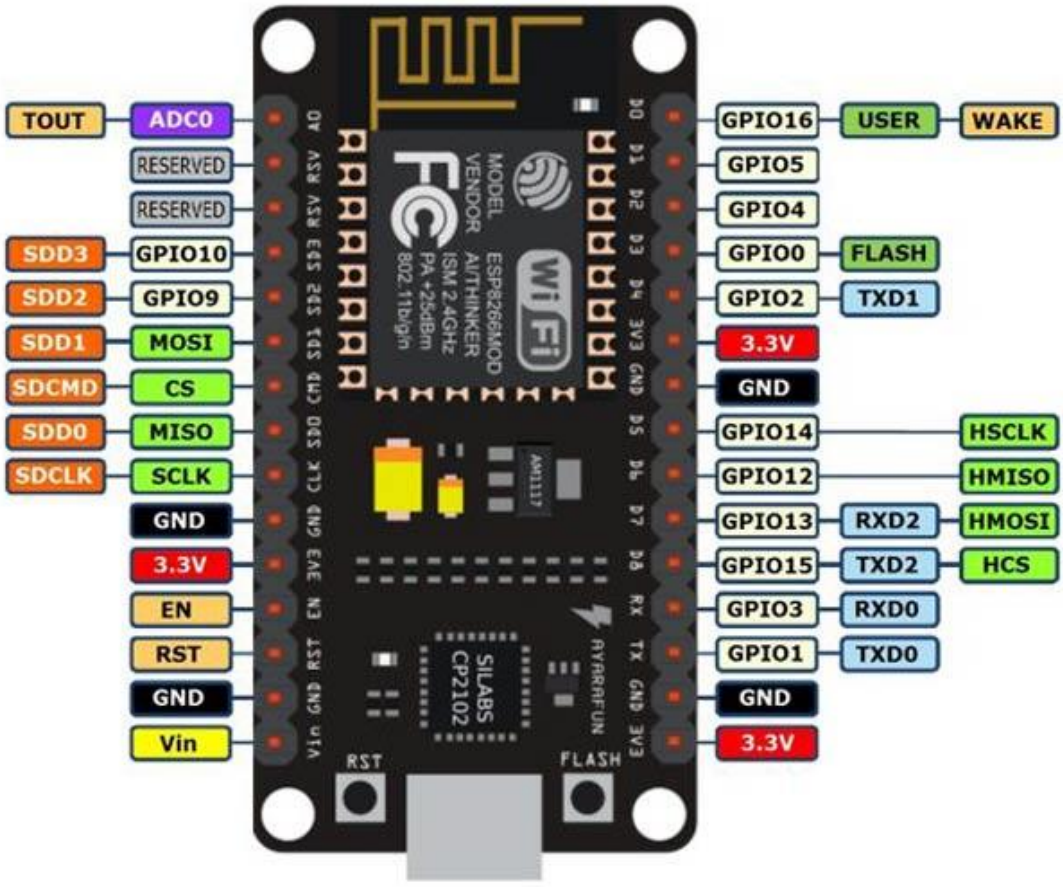


Figure 6 NodeMCU Pins

Feature and specifications

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

The module has a wireless WiFi transceiver operating in an unlicensed frequency range of 2400-2484 MHz in the IEEE 802.11 b/g/n standard, with support for TCP/IP communication protocol stack and WiFi security including WAP3. It can work with firmware to provide Wi-Fi connectivity to external host MCUs like the Arduino or as self-sufficient programmable MCUs with an RTOS-based SDK that can run applications.

Functions

ESP8266 has many applications when it comes to the IoT. Here are just some of the functions the chip is used for:

- Networking: The module's Wi-Fi antenna enables embedded devices to connect to routers and transmit data
- Data Processing: Includes processing basic inputs from analog and digital sensors for far more complex calculations with an RTOS or Non-OS SDK
- P2P Connectivity: Create direct communication between ESPs and other devices using IoT P2P connectivity
- Web Server: Access pages written in HTML or development languages.

4.2.2 Accelerometer

The purpose of an accelerometer is to measure the acceleration forces acting on an item in its instantaneous rest frame in order to locate the object in space and track its movement.

It is 3-axis accelerometer (x,y,z) with high resolution measurement at up to +/-16 g. Digital output data is accessible through either a SPI (3 or 4 wire) or I2C digital interface. You can set the sensitivity level to either +-2g, +-4g, +-8g or +-16g. The lower range gives more resolution for slow movements, the higher range is good for high-speed tracking, and its high resolution (4 mg/LSB) enables measurement of inclination changes less than 1.0

Features

- 2.0-3.6VDC
- Supply Voltage
- Ultra Low Power: 40uA in measurement mode, 0.1uA in standby 2.5V
- Tap/Double Tap Detection
- Free-Fall Detection

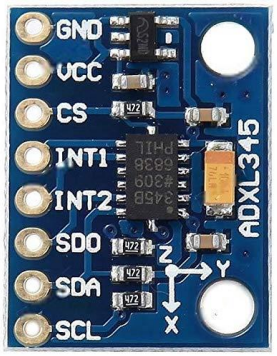


Figure 7 Accelerometer

4.3 Control Module:

4.3.1 Arduino

The open-source Arduino platform is used to create electrical projects. A physical programmable circuit board, or microcontroller, plus software called IDE (Integrated Development Environment), which is the primary text editor used for Arduino programming that runs on your computer and is used to create and upload computer code to the physical board.

The ATmega328P-based Arduino UNO is a microcontroller board. It contains 14 digital I/O pins (six of which are PWM outputs), 6 analogue inputs, a 16 MHz ceramic resonator, a USB connection, a power connector, an ICSP header, and a reset button. It comes with everything you need to support the microcontroller; simply connect it to a computer through USB or power it using an AC-to-DC converter or battery to get started.

Arduino may be used to develop interactive devices that accept input from a number of switches or sensors and operate a range of lights, motors, and other physical outputs. Both standalone and computer-based Arduino projects (such as those using Flash, Processing, or MaxMSP) are possible. The Arduino code is written in C++ and includes specific methods and functions. C++ is a computer language that is easy to understand. When you produce a ‘sketch' (Arduino code file), it is analyzed and compiled to machine language.

Figure 8 Arduino UNO

4.3.1 Why Arduino?

- Ready to Use

  The primary benefit of Arduino is its structure, which is ready to use. The 5V regulator, a burner, an oscillator, a microcontroller, a serial communication interface, an LED, and headers for the connections are all included in the full package that is Arduino. You don't need to consider any additional interfaces or programmer connections when you're programming.

- Code examples

  The library of examples included in the Arduino software is another significant benefit of the platform.

- Simple functions

  While programming an Arduino, you'll notice certain features that make life really simple. The automated unit conversion feature of Arduino is another benefit. You may claim that unit conversions are not a concern when debugging. Just focus all of your effort on the key components of your initiatives. There are no adverse effects to be concerned about.

- Large community

  There are several forums where people discuss the Arduino on the internet. Arduino is used by professionals, enthusiasts, and engineers to create their creations. You can simply obtain assistance with anything. Additionally, every single function of Arduino is explained on the official website.

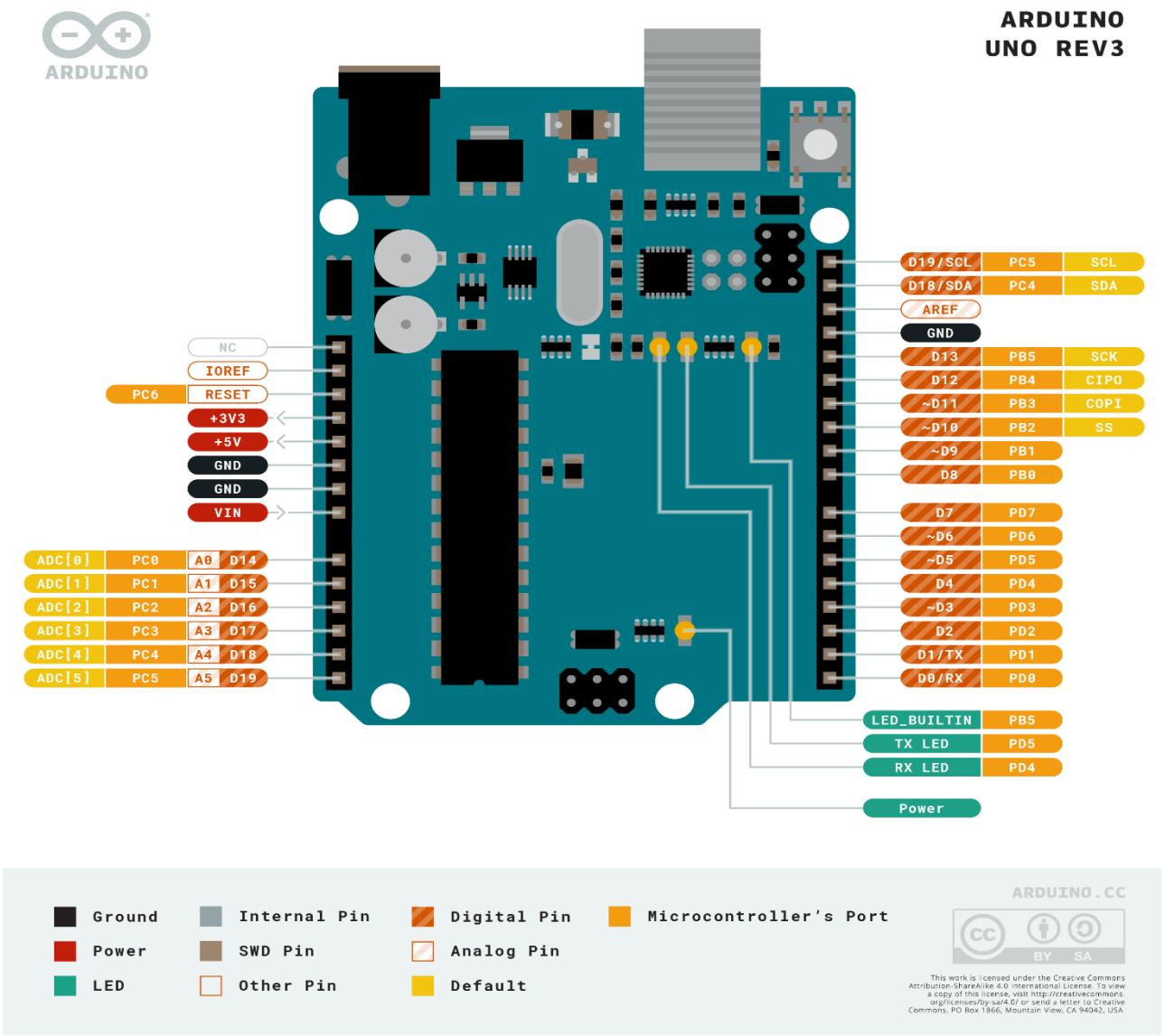


Figure 9 Arduino PINS

Technical specifications

Board	**Name**	Arduino UNO R3
	SKU	A000066
Microcontroller	ATmega328P	
USB connector	USB-B	
Pins	**Built-in LED Pin**	13
	Digital I/O Pins	14
	Analog input pins	6
	PWM pins	6
Communication	**UART**	Yes
	I2C	Yes
	SPI	Yes
Power	**I/O Voltage**	5V
	Input voltage (nominal)	7-12V
	DC Current per I/O Pin	20 mA
	Power Supply Connector	Barrel Plug
Clock speed	**Main Processor**	ATmega328P 16 MHz
	USB-Serial Processor	ATmega16U2 16 MHz
Memory	**ATmega328P**	2KB SRAM, 32KB FLASH, 1KB EEPROM
Dimensions	**Weight**	25 g
	Width	53.4 mm
	Length	68.6 mm

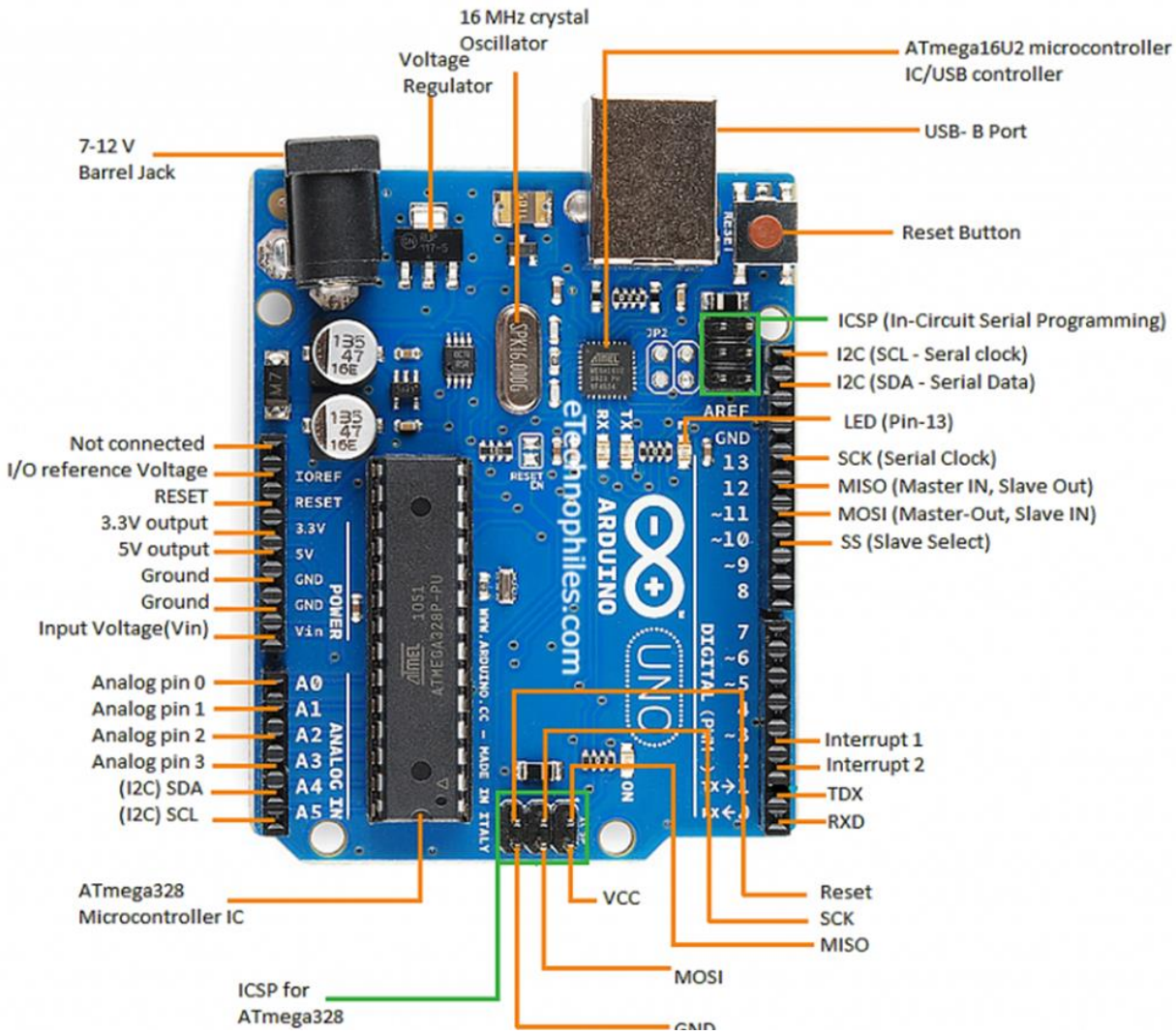


Figure 10 Figure 9 Arduino PINS_2

Arduino pins

✓ Atmega328P Microcontroller

It is a high-performance, single-chip microcontroller made by Atmel. The microcontroller chip is an 8-bit AVR RISC device. There are 23 general-purpose I/O pins, 32 KB of read-write ISP flash memory, 2 KB of static RAM, 1 KB of EEPROM, and 32 KB of SRAM.

✓ Atmega16U2 Microcontroller

In the Arduino UNO, it serves as a USB to serial converter.

✓ Voltage Regulator

It converts the input voltage to 5V. A voltage regulator's main function is to regulate the voltage level in the Arduino board. The regulator's output voltage stays stable and close to 5 volts even if the input supply voltage fluctuates in any way.

✓ Crystal Oscillator

It operates at a frequency of 16 MHz, giving the microcontroller the clock signal it needs to provide basic timing and control for the board.

✓ RESET Button

Every time we flash the code to the board, it is advised that we press this button.

✓ Barrel Jack

The Arduino board is powered by an external power source using the barrel jack, also known as the DC Power Jack. The manufacturer advises maintaining it between 7 and 12 volts even though it is often attached to an adaptor that runs between 5 and 20 volts.

✓ USB B-port

As well as connecting the board to the computer, this connector may be used to power a device using a 5V supply.

✓ ICSP header

It stands for “In-Circuit Serial Programming”. We can use these pins to program the Arduino board’s firmware. These pins are used to code and boot an Arduino from an external source. Arduino uno have two ICSP one for Atmega328P and the other for Atmega16U2

The ICSP header consists of 6 pins which are MISO, MOSI, SCK, V+, Ground, and Reset.

✓ I2C

It uses a two-wire serial connection. Inter-Integrated Circuits is the abbreviation. The I2C transmits and receives data across two lines.

✓ SCL

It represents Serial Clock. The pin or line is where the clock data is sent. It synchronises the data transfer between the two devices (master and slave). The master device creates the serial clock.

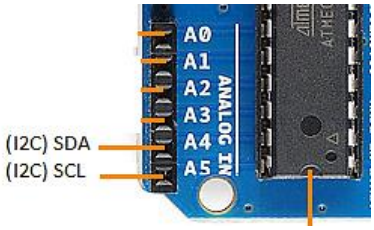
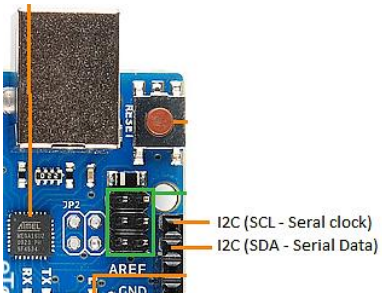


Figure 11 I2C Pins

✓ SDA

Its acronym is Serial Data. The line that the slave and master utilise to transmit and receive data is what is meant by this definition.



✓ SPI

Serial Peripheral Interface is what it stands for. Microcontrollers utilise it to swiftly communicate with one or more peripheral devices.

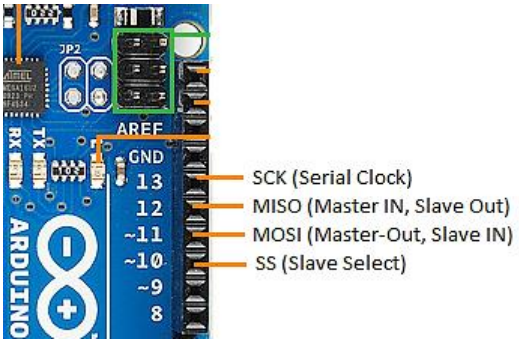


Figure 12 SPI Pins

- SCK-It, also known as a serial clock, These are the pulses produced by the clock that synchronise data transmission.
- Master Input/ Slave Output, or MISO, is its acronym. The data from the Slave is received on this data line in the MISO pin.
- MOSI-It, or Master Output/ Slave Input, is an acronym. Data transmission to the peripherals occurs on this line.
- Slave Select is abbreviated as SS-It. The master uses this line as the enable line. A device is able to interact with the master when the Slave Select pin value is set to LOW. When it has a HIGH value, it disregards the master. This enables us to share the MISO, MOSI, and CLK lines among many SPI peripheral devices.

✓ External Interrupts (2 and 3)

On a low value, a rising or falling edge, or a change in value, these pins can be used to initiate an interrupt.

✓ TXD and RXD

The serial communication that takes place on these pins. The data is transmitted using the TXD, and it is received using the RXD. It also symbolises the smooth flow of information.

4.3.2 H bridge

An H-Bridge circuit contains four switching elements, transistors or MOSFETs, with the motor at the centre forming an H-like configuration. By activating two particular switches at the same time, we can change the direction of the current flow, thus change the rotation direction of the motor. Utilized to regulate speed as well. It may be applied to lighting projects like high-powered LED arrays to regulate their brightness. An H-bridge is a circuit that can drive a current in either polarity and be controlled by pulse width modulation.

A DC motor spins either backward or forward, depending on how you connect the plus and the minus. If you close switch S1 and S4, you have plus connected to the left side of the motor and minus to the other side. And the motor will start spinning in one direction. If you instead close switch S2 and S3, you have plus connected to the right side and minus to the left side. And the motor spins in the opposite direction.

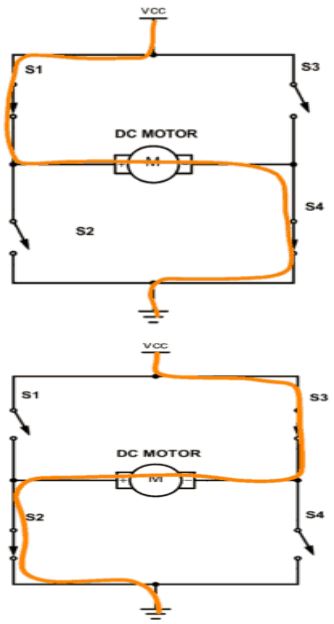


Figure 13 H-bridge circuit

Pinout

ENA & ENB pins are speed control pins for Motor A and Motor B while IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and Motor B.

Specification

- Dual H-bridge drive
- Chip: L298N (ST NEW)
- Logic voltage: 5V
- Drive voltage: 5V-35V
- Logic current: 0mA-36mA
- Drive current: 2A(MAX single bridge)

- Storage temperature: -20 to +135
- Max power: 25W
- Weight: 30g
- Size: 43*43*27mm (approx. 1.75" x 1.75" x 1")
- Compatible with L297/L298 driver

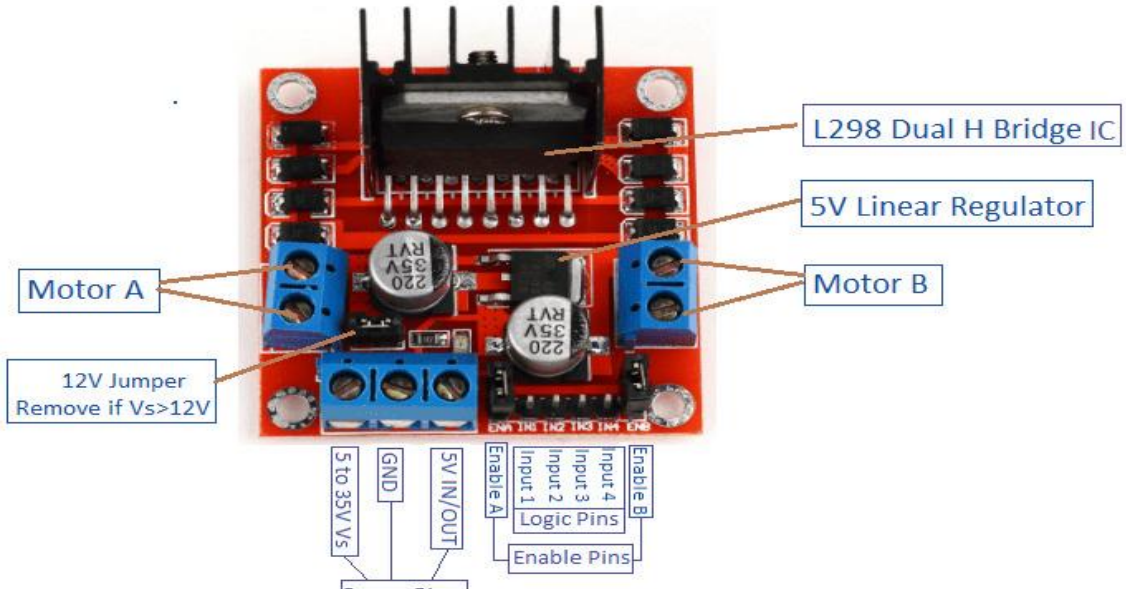
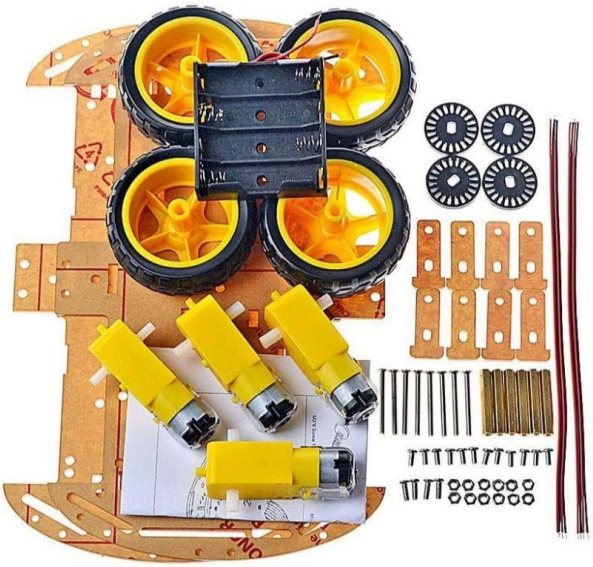


Figure 14 H-bridge

4.3.3 DC Motors

A DC motor or direct current motor is an electrical machine that transforms electrical energy into mechanical energy by creating a magnetic field that is powered by direct current. When a DC motor is powered, a magnetic field is created in its stator. The field attracts and repels magnets on the rotor; this causes the rotor to rotate. To keep the rotor continually rotating, the commutator that is attached to brushes connected to the power source supply current to the motors wire windings.

4.3.4 RC CAR kit



5. System Features:

5.1-Map based navigation and localization

5.1.1 Installing ROS

Robot Operation System (ROS) is a framework that enables the use of numerous different packages to control a robot. These packages cover a wide range of topics, including motion control, path planning, mapping, localization, SLAM, perception, transformations, communication, and more. With regard to interacting with those packages, ROS offers the option to build unique packages.

5.1.2 What is ROS?

The Robot Operating System (ROS) is a set of software libraries and tools that help you build robot applications.
ROS contains everything you need for your upcoming robotics project, from drivers to cutting-edge algorithms and with robust development tools.

The future of robots in business, industry, and for developers is powered by ROS.

Due to its flexible and user-friendliness, Ubuntu has been the main platform for ROS since the beginning.

Similar to how Canonical supports Ubuntu, Open Robotics is the organisation that guides ROS; nevertheless, the community is what propels the project forward.

Companies and developers keep looking to Open Robotics and Canonical to help them realise their vision as the robotics sector expands.

5.1.3 ROS versions

Distribution	Release date	Poster	EOL date	Support duration
Noetic Najimy's (last ROS 1 release)	23 May 2020		**May 2025**	5 years
Melodic Morenia	23 May 2018		2023-05-30	5 years
Lunar Loggerhead	23 May 2017		2019-05-30	2 years
Kinetic Kame	23 May 2016		2021-05-30	5 years
Jade Turtle	23 May 2015		2017-05-30	2 years
Indigo Igloo	22 July 2014		2019-04-30	5 years
Hydro Medusa	4 September 2013		2014-05-31	0.5 years

Groovy Galapagos	31 December 2012		2014-07-31	2 years
Fuerte Turtle	23 April 2012		--	
Electric Emys	30 August 2011		--	
Diamondback	2 March 2011		--	
C Turtle	2 August 2010		--	
Box Turtle	2 March 2010		--	
(Initial Release)	2007	n/a	--	--

We are using the last version which is Ros Noetic.

5.1.4 Why ROS?

The open-source ROS (Robot Operating System) software development kit is used for robotics applications.
Developers from a variety of sectors have access to ROS, which provides a standard software platform that will support them from research and development all the way to deployment and production.
So, we use ROS due to this resons:

1. Global Community

The ROS project has fostered a global community of millions of developers and consumers who contribute to and enhance that software for more than ten years, creating a massive ecosystem of software for robotics.
That community, who will serve as its guardians in the future, helped construct ROS.

2. Proven in Use

Throughout the robotics sector, ROS is used.It is customary while teaching robotics.
It serves as the foundation for the majority of robotics research, including small student initiatives, institutional partnerships, and massive contests. And it may be found within the manufacturing robots that are currently in use all over the world.
ROS has contributed to the creation of value worth billions of dollars in just the autonomous mobile robot (AMR).

3. Shorten Time to Market

By giving you the resources, libraries, and tools you need to create robotics applications, ROS enables you to focus more of your time on tasks that are crucial to your company's success.
You have the freedom to choose where and how to use ROS, as well as the ability to modify it to suit your needs, because it is open-source.
Furthermore, ROS isn't exclusive; you don't have to decide between using ROS or another software stack; ROS interfaces seamlessly with other software to bring its tools to bear on your issue.

4. Multi-domain

From indoor to outdoor, automotive to undersea, consumer to industrial, ROS is ready for usage in a wide range of robotics applications.

5. Multi-platform

With Linux, Windows, macOS, and micro-ROS support and testing, along with a number of embedded systems, ROS 2 enables the rapid creation and deployment of on-robot autonomy, back-end administration, and user interfaces.

The tiered support strategy enables the introduction and promotion of ports to new platforms as interest in and investment in them grow, such as real-time and embedded operating systems.

Since ROS is and always will be open-source, our whole worldwide community will always have free and unrestricted access to a top-notch, top-of-the-line, fully functional robotics SDK.

We use open standards (like OMG's DDS) and develop ROS on top of other open-source initiatives whenever we can.

6. Commercial Friendly

We never insist that our user community contribute open-source code to ROS, even if we sincerely welcome it.

We make ROS available under permissive open-source licences, with Apache 2.0 serving as our standard.

You are free to alter ROS, combine it with your own and other closed-source software, and then distribute the resulting private product without even informing us.

7. Industry Support

The ROS 2 Technical Steering Committee's membership demonstrates the industry's strong support for the software. In addition to creating products on top of ROS 2, businesses big and small from all over the world are devoting their resources to open-source contributions to the platform.

5.1.5. How is it work:

In general, ROS is made up of tools and code that support your project's code in executing and doing its tasks, as well as the infrastructure needed to support its operation, such as messages travelling between processes.

Every node in the loosely linked system known as ROS is a process, and each node is expected to do a single job.

Nodes exchange messages with one another over logical channels called topics. The publish/subscribe architecture allows each node to communicate with the other node and transmit or receive data. Later, we'll witness that in action.

In order to provide a built-in package system, ROS' main objective is to promote code reuse in robotics research and development. Remember once more that ROS is either an operating system, a library, or an RTOS. It's a framework that makes use of the idea of an OS.

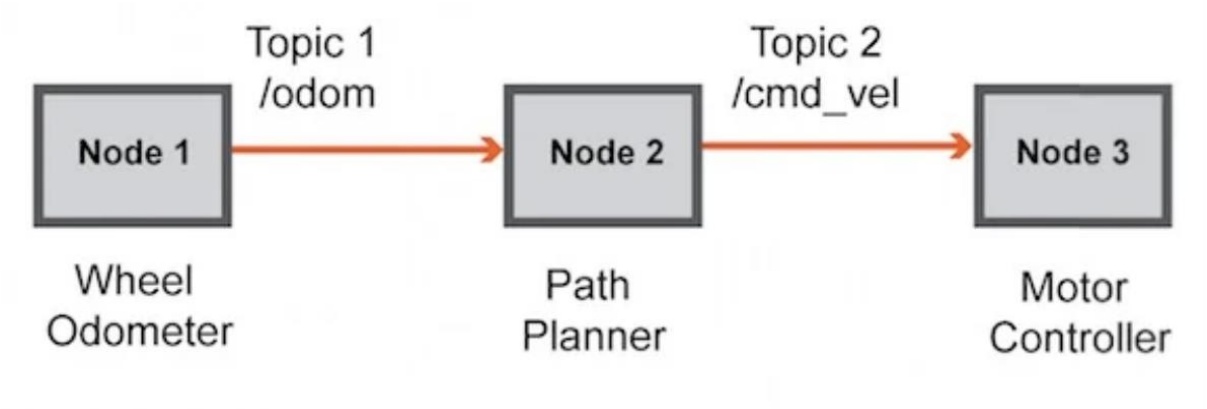


Figure 15 ROS node example

In ROS, there is a Master who is in charge of name registration and lookup for the rest of the system in order to manage this loosely linked environment. Nodes wouldn't be able to communicate or find one another without the Master.

You need type "roscore" or "roslaunch" to start the Master.

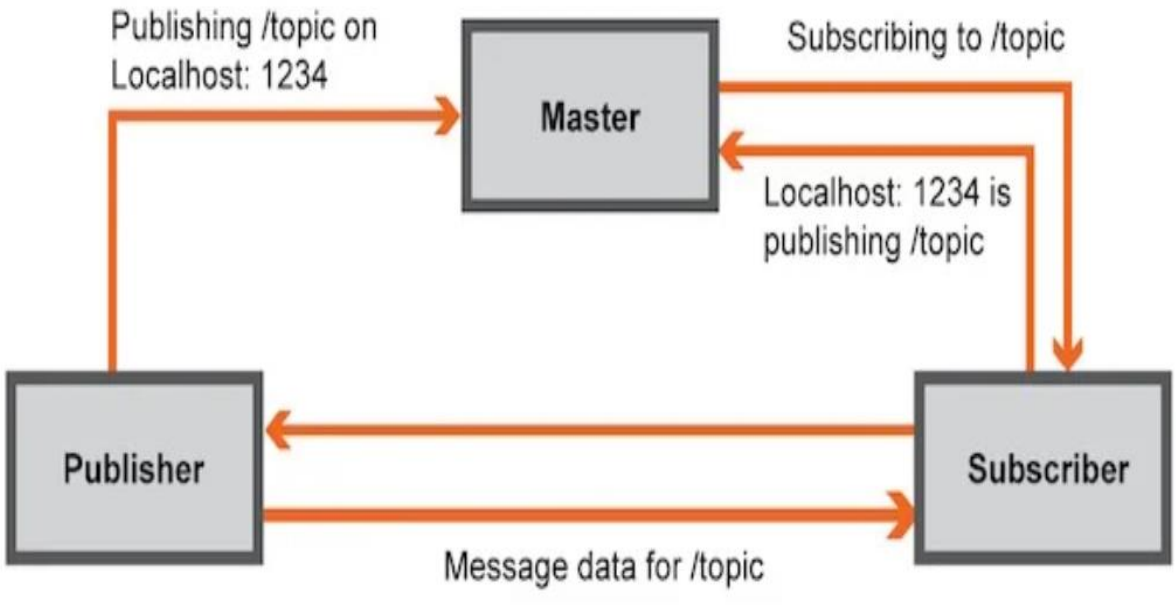


Figure 16 Publisher and Subscriber relation

Return to the messages, Nodes fill the structs of data that make up messages with bits of data. Nodes exchange them over logical connection channels known as topics, after which nodes post topics or subscribe to them.

Specifically, using the "rostopic" command-line tool, you may list topics.

5.1.6 Main ROS component:

Nodes

One process running the ROS graph is represented by a node.

Before it can do anything else, each node must register its name with the ROS master.

If a node is designated as anonymous, it will produce a second identifier at random to go along with its supplied name. Multiple nodes with various names can exist under several namespaces.

The majority of ROS client code is written in the form of a ROS node, which performs actions depending on information it gets from other nodes, provides information to other nodes, or sends and receives requests for actions from other nodes.

Topics

Buses are topics that are used by nodes to transmit and receive messages.

Additionally, topic names must be distinctive inside their namespace.

To send messages to a topic, a node must publish to said topic, while to receive messages it must subscribe. The publish/subscribe approach is anonymous; each node only understands that it is transmitting or receiving on a particular topic.

There are many different and user-definable message kinds that may be passed on a topic.

These messages may contain sensor data, motor control instructions, condition details, actuator instructions, or anything else.

Services

A node may also advertise services. A service is an action a node may do that will result in a single outcome.

Services are so frequently employed for tasks that have a specified beginning and conclusion, such as taking a single frame of video, as opposed to processing velocity directives to a wheel motor or odometer data from a wheel encoder. Nodes advertise services and call services from one another.

5.1.7 Tools of ROS:

There are different tools used in ROS:

1) Rviz is a three-dimensional visualization tool that emphasizes the visualization of existing data;

2) Gazebo is a 3D physical simulation platform that emphasizes the creation of a virtual simulation environment.

The difference between rviz and gazebo in ROS:

Rviz requires existing data.

Rviz provides a lot of plug-ins, these plug-ins can display images, models, paths and other information, but the premise is that these data have been published in the form of topics, parameters, what **rviz** does is to subscribe to these data, and complete the visual rendering, so that development It is easier for the person to understand the meaning of the data.

Gazebo is not a display tool. It emphasizes simulation. It does not need data, but creates data.

We can create a robot world for free in Gazebo, which can not only simulate the motion function of the robot, but also simulate the sensor data of the robot. And these data can be displayed in **rviz**, so when using gazebo, it is often used in conjunction with **rviz**. When we don't have robot hardware or the experimental environment is difficult to build, simulation is often a very useful weapon.

In summary, if you already have a robot hardware platform on hand, and you can complete the required functions on it, using **rviz** should be able to meet the development needs.

If you don't have any robot hardware, or you want to test some algorithms and applications in a simulation environment, **gazebo + rviz** should be what you need.

In addition, **rviz** can cooperate with other function packages to establish a simple simulation environment, such as **rviz + ArbotiX**. You can refer to "ROS by Example", but the essence of **rviz** is to handle the display part.

5.1.8 Network configuration:

In this stage we determine the Master and observer machines. In our case the robot is the Master machine and the Workstation (laptop) is the observer. We want to use our laptop to access the communications sent and received by ROS. Additionally, it will enable us to conveniently visualise things (with rviz). In order to achieve this, we have to follow the following steps:

1- Run a Linux machine with ROS noetic, or older versions. Either a virtual machine or a real machine. In our case we are using Oracle VM with Ubuntu 20.

2- Look for ROS IP and ROSMASTER URI on the Master. Both machines' communication will require these two pieces of information. By using the ifconfig command, you may find the ROS IP (the IP address of the master computer). In this example, we set the robot's export ROS IP=192.168.1.22 export ROS MASTER URI=http://192.168.1.22:11311 (your network's IP addresses would likely be different).

3-Export ROS IP=192.168.4.12 was configured on the observer laptop. Exporting ROS MASTER URI to ubiquityrobot.local:11311 Different from other master URIs It should be the same as the.local, thus I think setting ubiquityrobot.local to 192.168.1.22 would work, but I haven't tested it. I would either write a script to execute these lines simultaneously or put them to the Observer machine's.bashrc file. It’s not required, but it will simplify the process.

4- Activate Rescore (which is the main Ros node) on the master (robot).

5.1.9 Testing the lidar:

I am using the Delta 2A lidar for this build. The first step is to install the necessary drivers. The driver is a ROS package. We can write your Own configured drivers or you can use the component package. Select the correct serial which the LiDAR is connected to it.

The LiDAR uses to main nodes:

1-publish_node:which sets the frequency, the serial port and the maximum range.

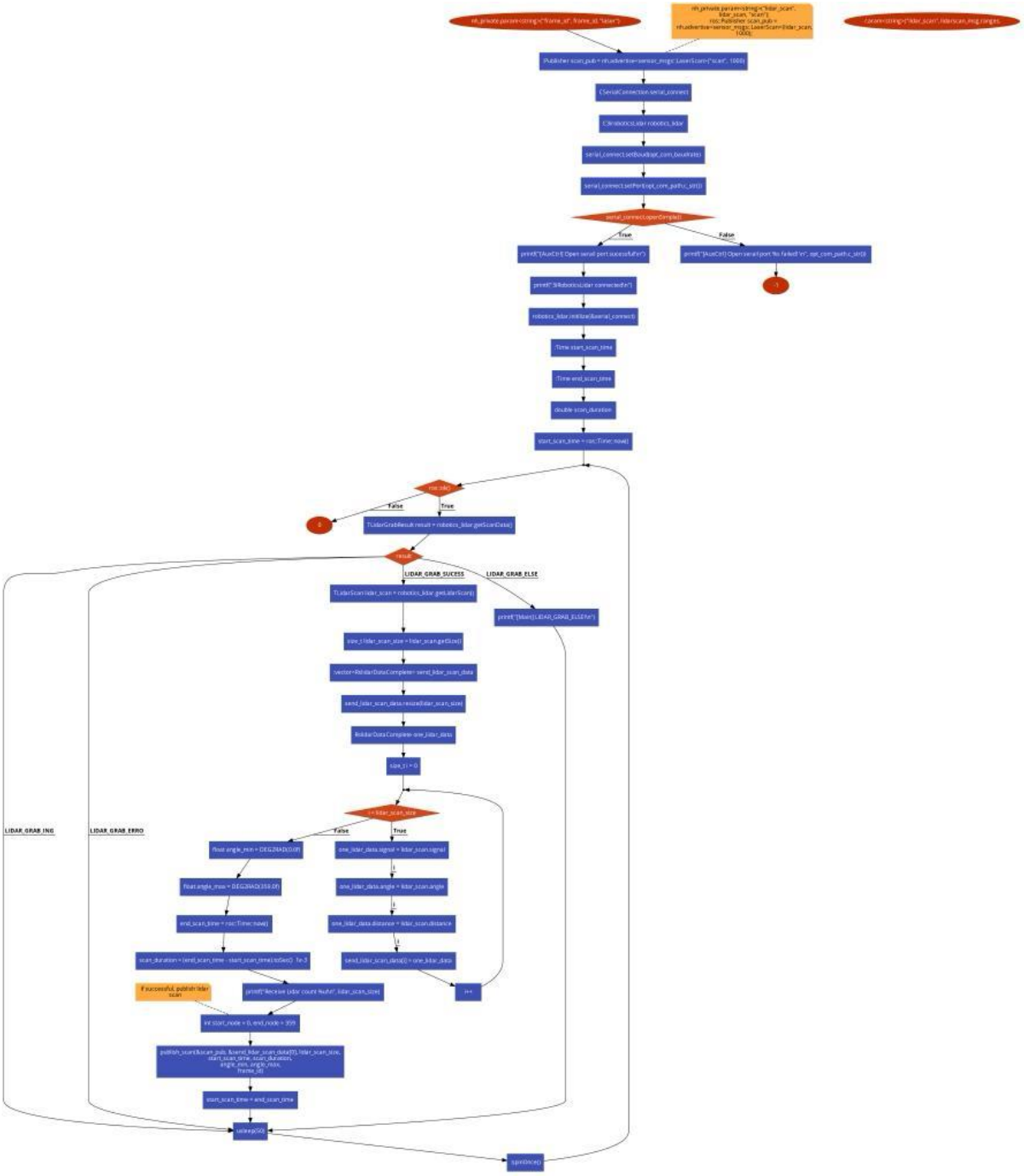


Figure 17 publishes_node flowchart

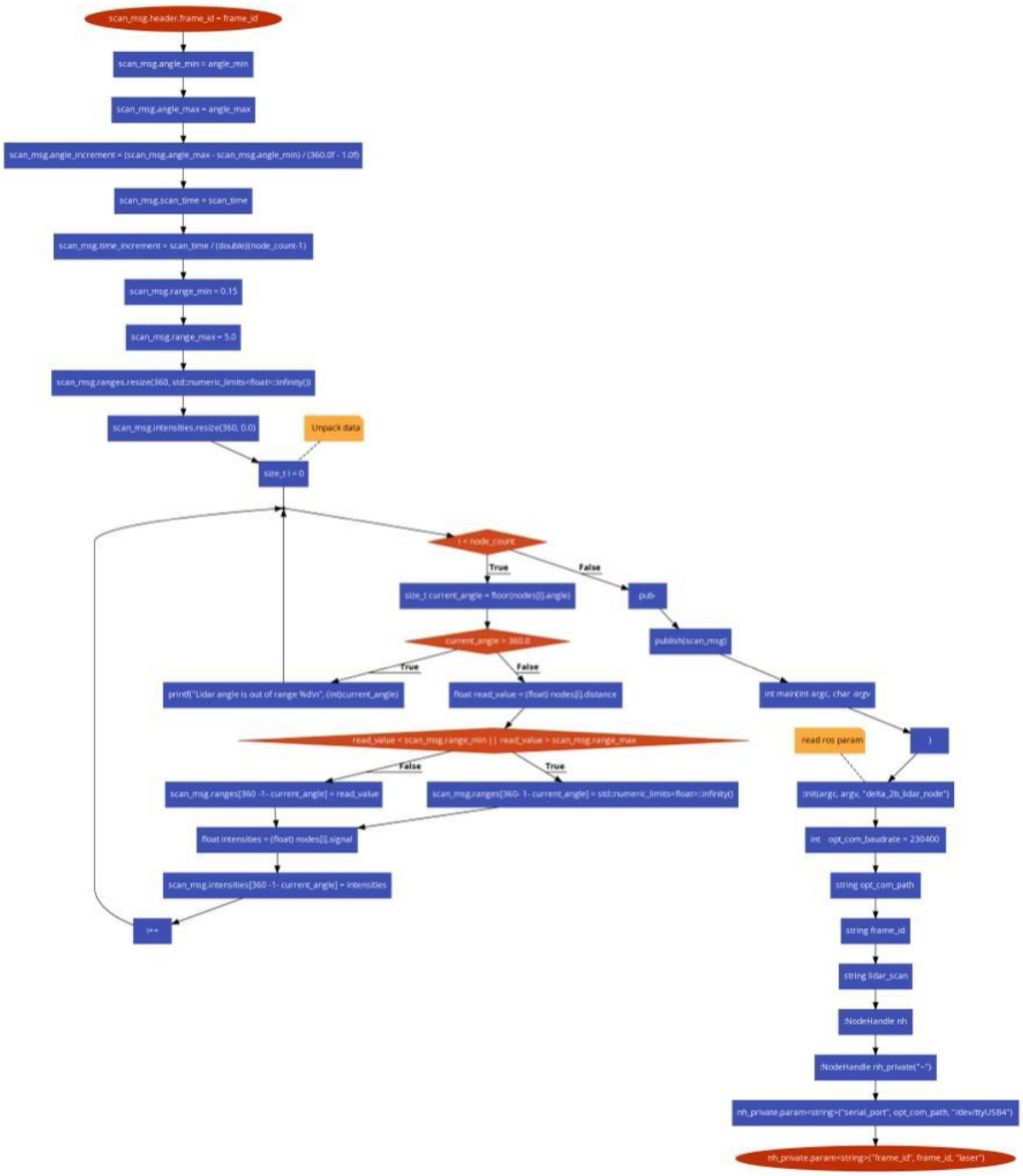


Figure 18 publish_node flowchart

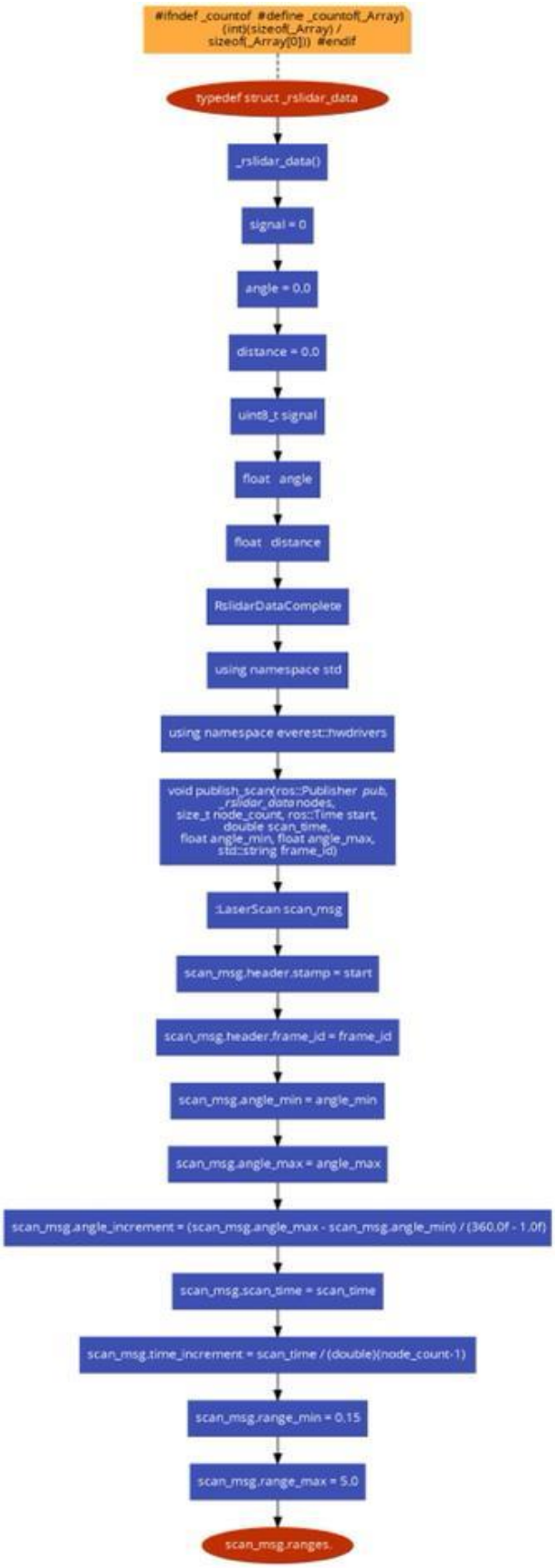


Figure 19 publish_node flowchar

2-subscribe_node: collect the scanned data in order to visualize it later, and performs the required calculations.

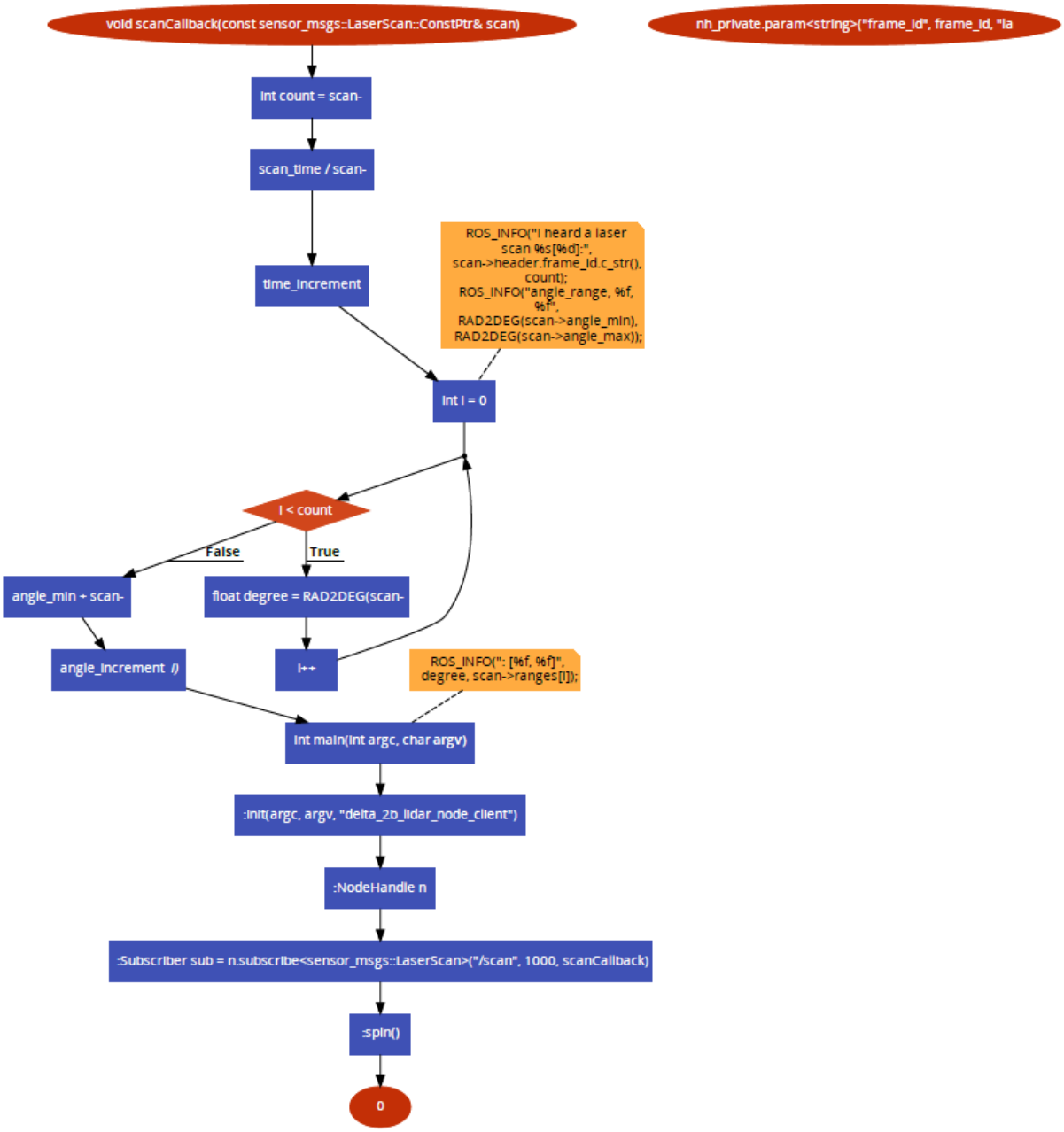


Figure 20 subscribe_node flowchart

You can visualize the scans in Rviz, by adding the topic /scan.

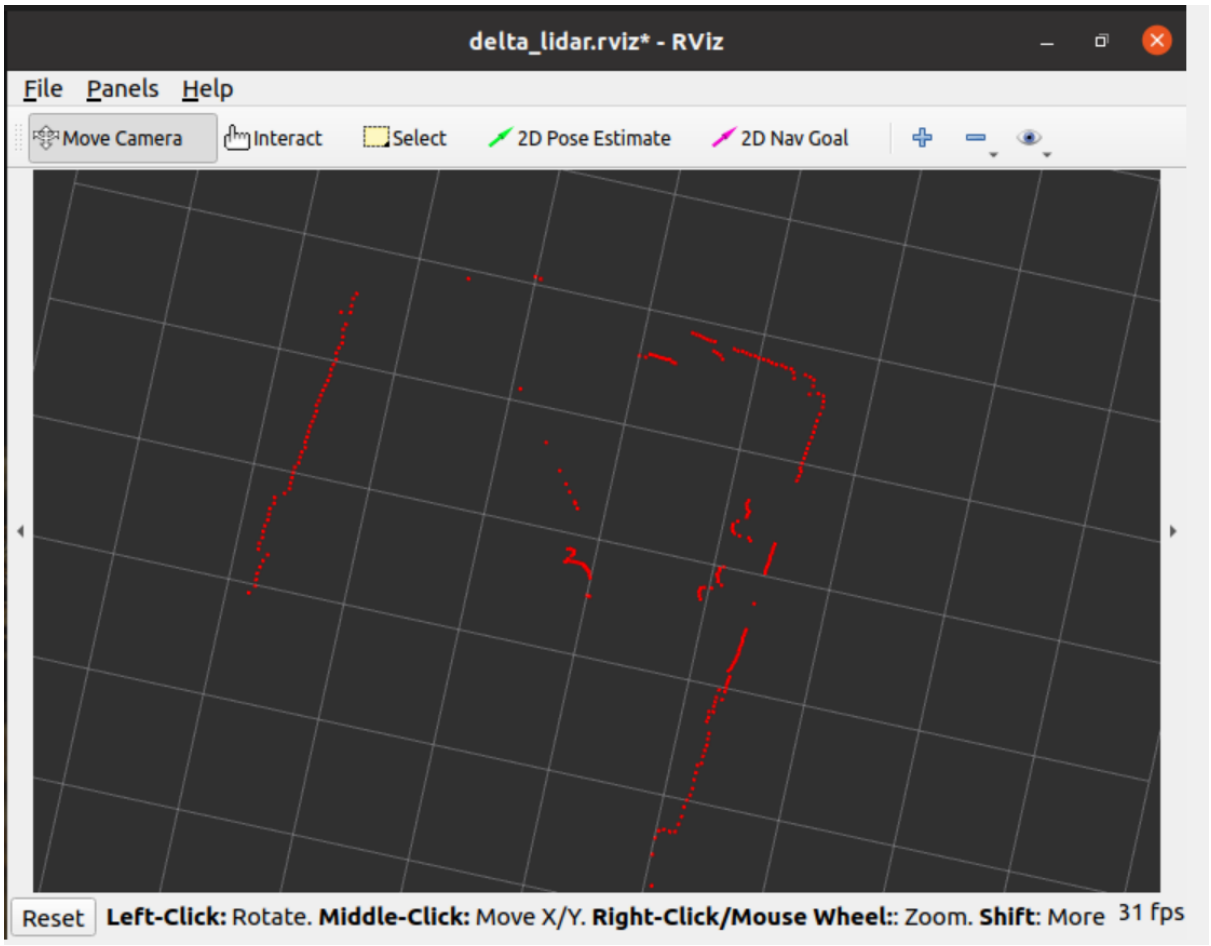


Figure 21 Realtime scans

The LiDAR uses UART serial protocol in order to communicate with the robot machine.

5.1.10. Connecting ROS with Arduino.

Our objective is to send commands from the Raspberry Pi to the Arduino so that the Arduino can direct the motors. To do that, we'll install rosserial, a ROS module that makes it possible for Arduino and ROS to communicate, on both the Raspberry Pi and the Arduino. As per the instructions on the ROS website, we begin by installing the Rosserial package, which links the Arduino platform with ROS.

5.1.11. Using Hector-SLAM

We will now provide our robot the mapping and (in a little while) localization functionality. We used the Hector-SLAM package. It enables us to produce the maps (with a Lidar alone, without the need for an IMU) that we will afterwards utilise for localization and navigation.

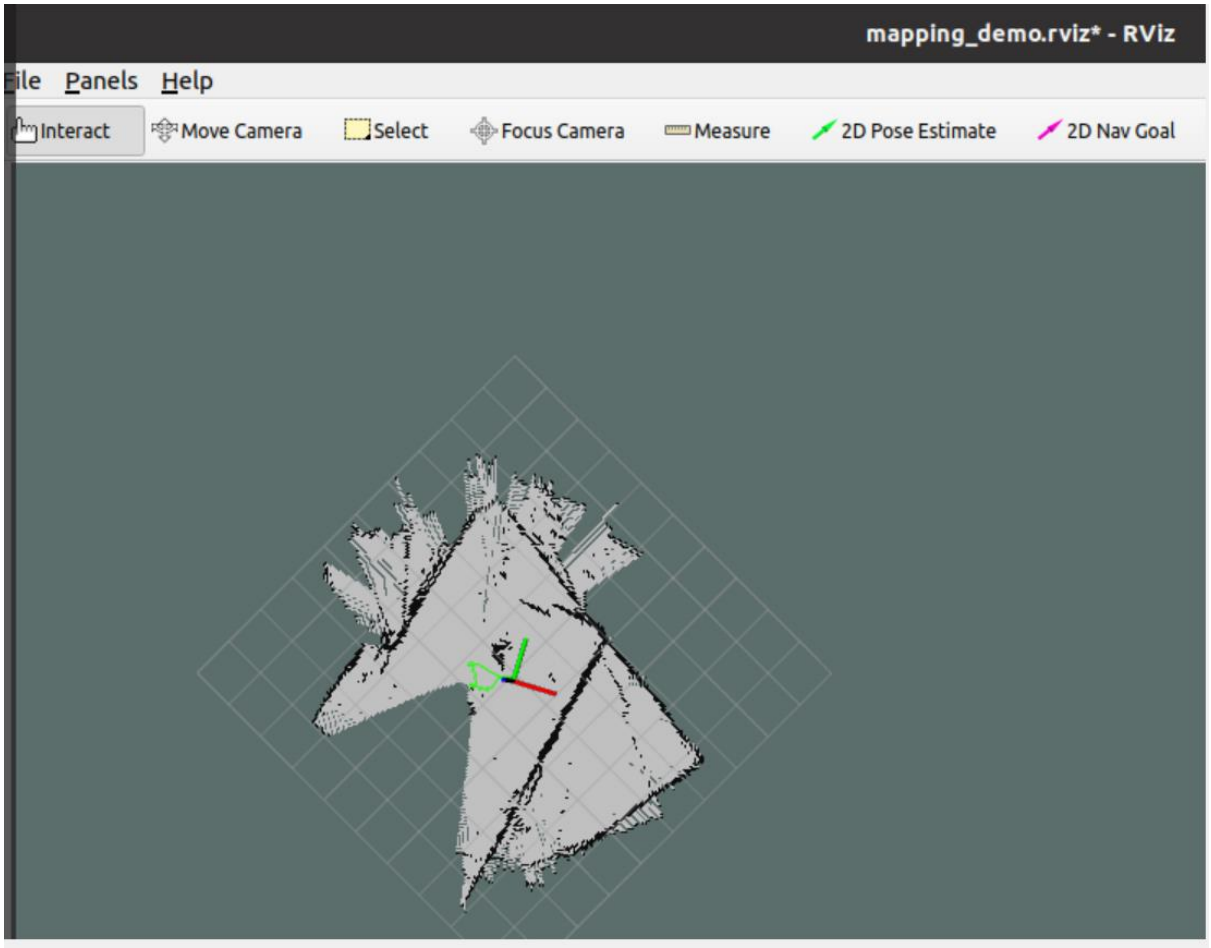


Figure 22 Realtime mapping

5.1.12. Low level control

In order to move the robot across the world, we now want to build a ROS package that would support ROS communication. Making a subscriber node that would run on the Arduino and listen to the topic /cmd vel is what we want to do. The first thing we'd want to do is start giving the robot commands via the keyboard.

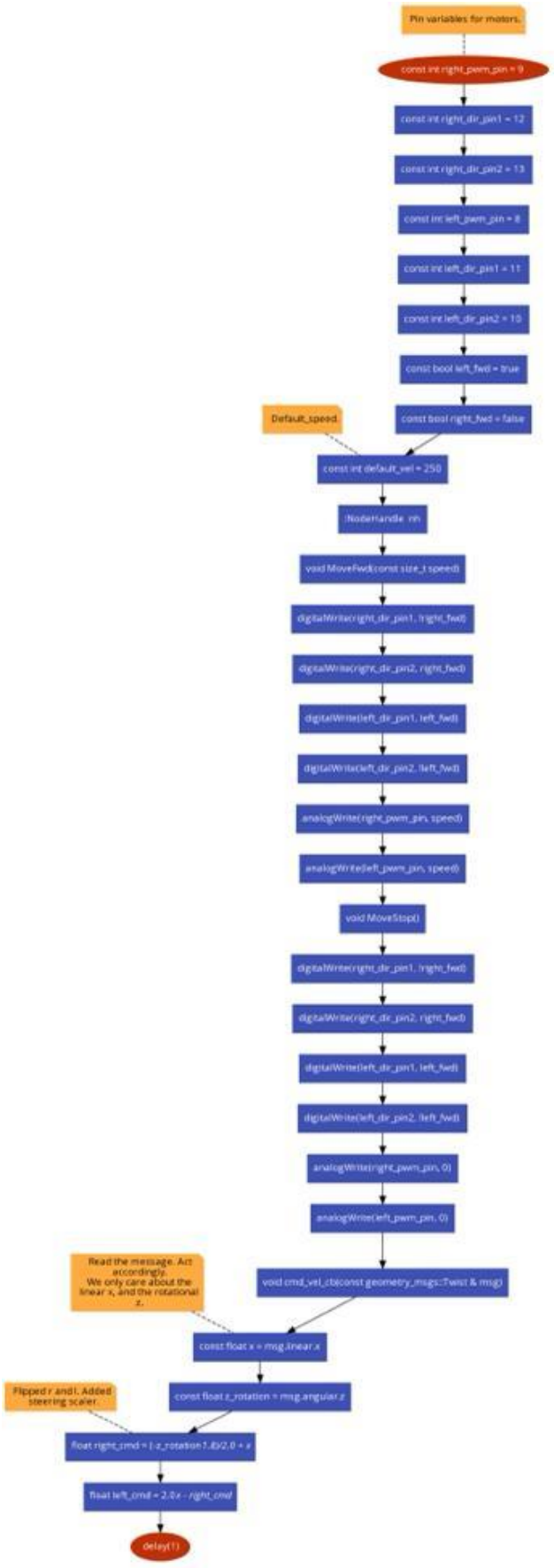


Figure 23 Low level control flowchart

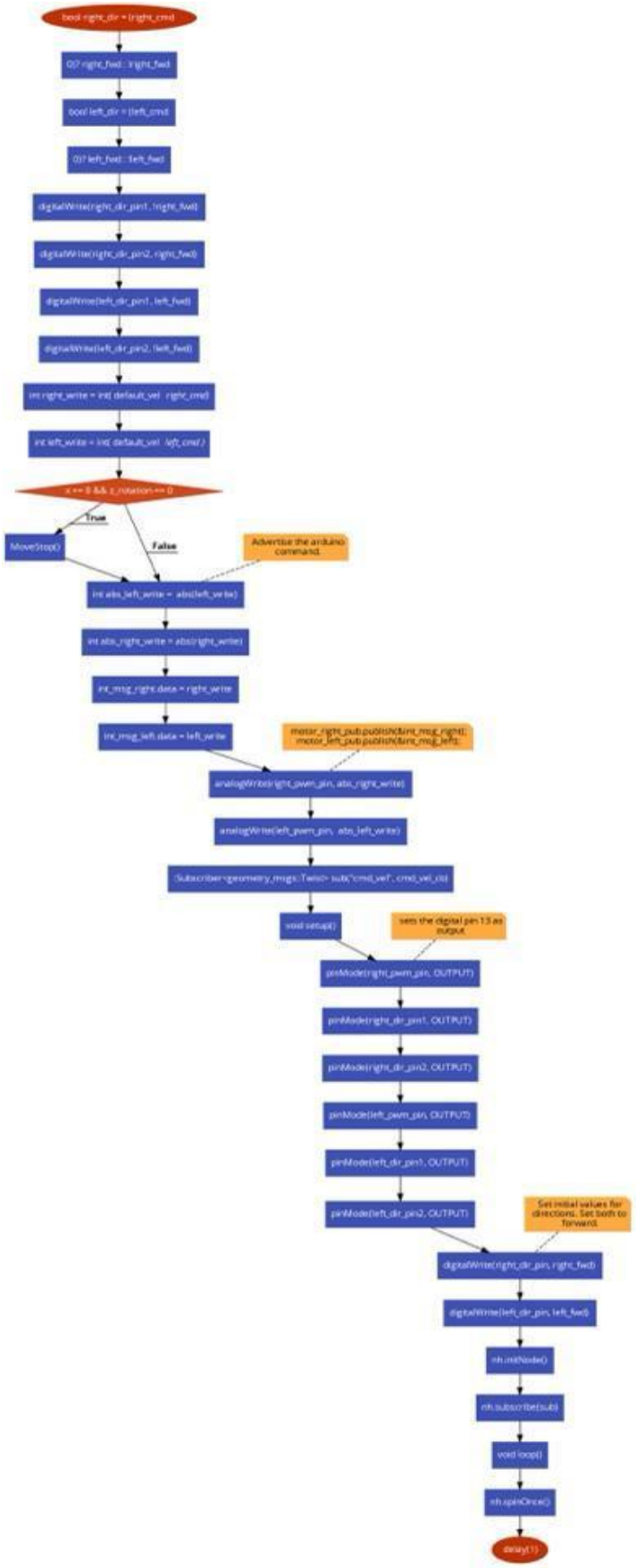


Figure 24Low level control flowchart

5.1.13. Save a map

Downloading the map server will allow us to get started with this topic. To record a map, we should spin up the robot by executing the LiDAR node as well as the hector slam node with we have explained before.

```
rosrun map_server
map_saver -f my_map
```

my_ map.yaml and my_ map.pgm files will be saved by this command. These two files detail the map's occupancy information. By substituting any name for the map argument, you can modify the name of this map. You may view the produced map by using the.pgm file.

Figure 25Example of a map

5.1.14. Localization and navigation

In the following phase, our objective is to localise our robot on the recorded known map and direct it to a specified pose.

In order to achieve this, we need to use a lot of packages.

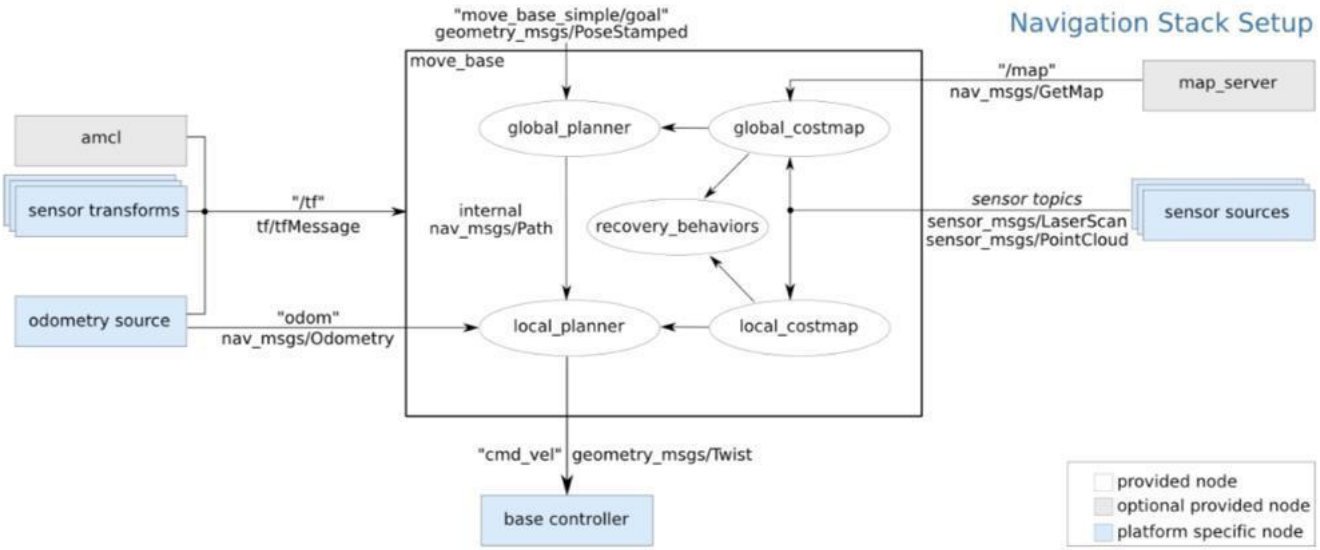


Figure 26navigation stack block diagram

First the navigation stack assumes that the robot is configured in particular manner in order to run. The diagram shown provides an overview of this configuration. White components are required components, they already implemented in ROS. These components are:

1. Global planner: responsible for planning the static path from source to Location, which uses the A* algorithm.
2. Local planner: responsible for executing the mission, following the path and avoiding obstacles.

The recovery behaviours: represents backup plans when the robot gets stuck.

Global and local cost maps: maps used by global planner and local planner respectively.

The gray components are optional components that are already implemented, it includes the map server that is ROS API which is responsible for providing the map to the navigation stack. Amcl which implements the localization of the Robot and provide the amcl pose. The blue components depend on the robot platform it includes the transformation of each sensor attached to the robot, the source of odometry messages, the base controller and sensor sources.

5.1.15. Recovery behaviour

In some cases, the vehicle will stuck, in order to achieve the goal it follows the recovery behaviour. At first global planner sends to local planner the optimal path that should be followed and local planner guides RC car by sending to it velocity and direction of motors required as cmd_vail() message and while RC car moving if it got stuck it informs local planner and the last responds to by sending

clear_rotating() command to round in all directions around the obstacle and last local planner will continue sending guide message until goal is achieved.

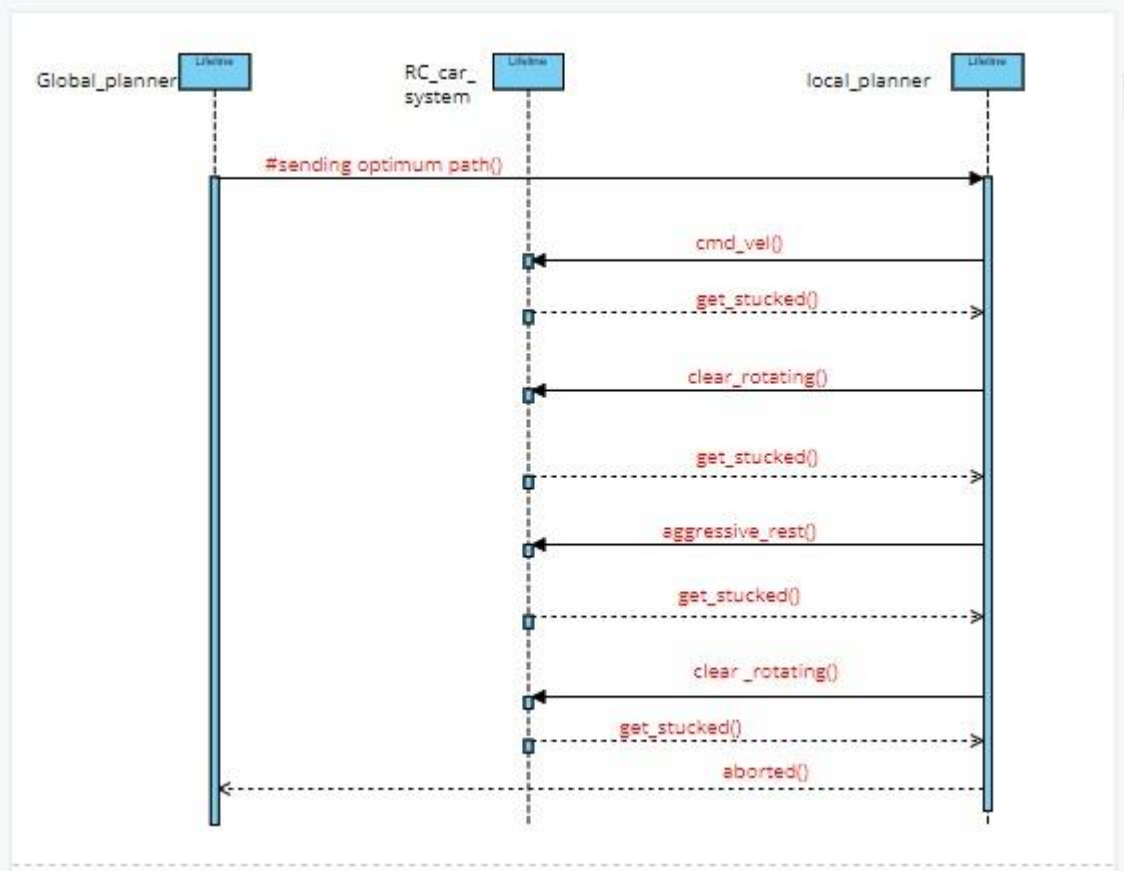


Figure 27Recovery behaviour sequence diagram

5.2 Design Rationale

There were different architectures and components that we could use, for example we could replace the laser scanner with PI camera, and ultrasonic sensor. In this case we will not have a 360 degree of vision, and the vehicle will not explore Obstacles 2 meters away. It also will require a machine learning model which will increase the processing time and therefore it will limit the vehicle maximum speed, also because that you don't have a 360 degree of vision, you are not able to create a map and calculate the optimum path which the vehicle must follow. On the other hand, it will recognize the traffic signs which will guide the vehicle. According to all these parameters we preferred the laser scanner.

6. Optimization of computer vision using FPGA

6.1. Computer vision:

Computer vision is a branch of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital photos, videos, and other visual inputs and then perform actions or make recommendations based on that information. If AI allows computers to think, computer vision allows them to see, watch, and understand. Computer vision tasks encompass ways for acquiring, processing, analysing, and comprehending digital images, therefore we can divide computer vision work into three main steps:

• Image acquisition:

There are different sources for image acquisition:
1-Camera (monochrome or colour), line scan camera,
2-Using Sensors such as:
-High Precision, Single Sensor.
-Using Sensor Array.

• Image Processing:

By using filters or deep learning models, which automate many steps in this process. However, the models are frequently trained by first being fed a large number of photos that have already been tagged or identified.

• Understanding the image:

The final stage is the interpretative step, in which an object is identified or categorised.

6.2. Computer vision Techniques :

6.2.1. Image segmentation :

Partitioning an image into its component sections or objects is one of the most challenging undertakings. For instance, extracting words and letters, characters and lips, etc.

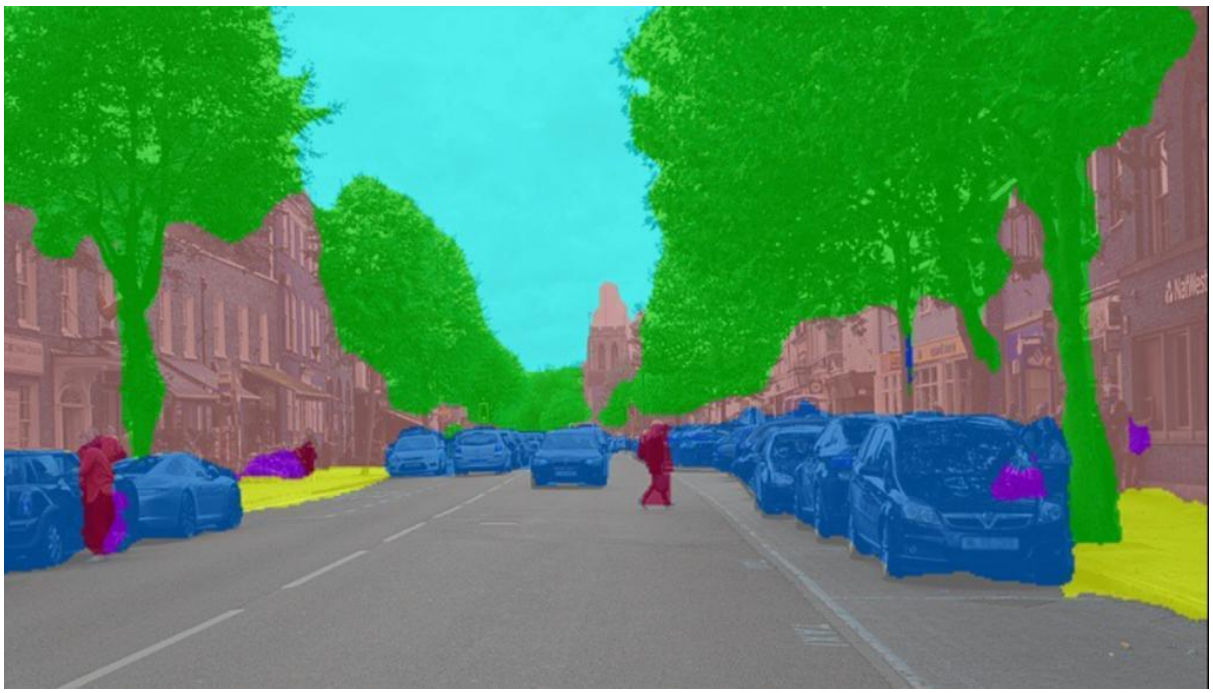


Figure 28Image segmentation

Green represents trees, blue represents vehicles and yellow represents the sidewalk.

6.2.2. Edge detection:

Is a way for better identifying what is in a picture by locating the edge of an object or landscape.

Figure 29Edge detection

6.2.3. Object detection:

Identifies a certain object in a picture the advanced object detection can recognise many items in a single image, such as an offensive player, a defensive player, a ball, and so on. These models build a bounding box and identify everything inside it using an X,Y coordinate.

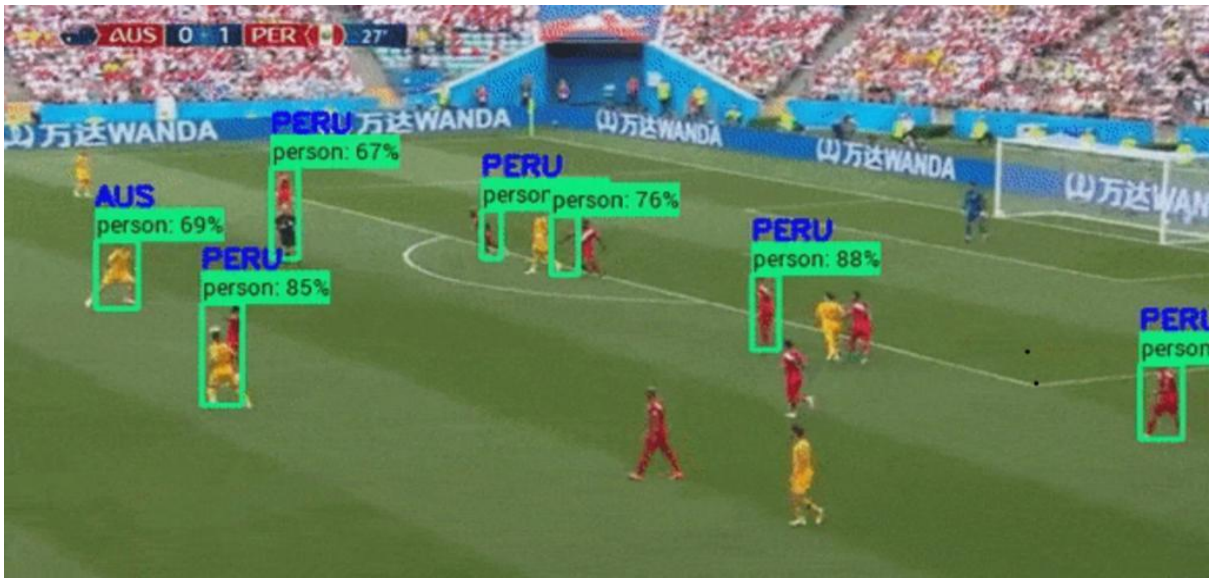


Figure 30Object detection

6.2.4. Facial recognition:

Which is an advanced form of object detection that can identify a particular person in addition to recognising a human face in an image.

6.2.5. Pattern detection:

Recognizing repeated patterns in images, such as fingerprints, can be done through the method of pattern detection.

6.2.6. Image classification:

Divides images into many categories, such as classifying cats or dogs, etc.

6.3-Computer vison Applications:

Computer vision can be used in many applications for example:

- ✓ Automatic character recognition
- ✓ Industrial machine vision for assembly and
- ✓ Inspection
- ✓ Military recognizance
- ✓ Automatic recognition of fingerprints
- ✓ Also, it is used in different fields such as:
- ✓ Autonomous vehicles
- ✓ Medicine
- ✓ Military
- ✓ Sports

Each field of can use computer vision in many things and for a lot of applications.

6.4-ADAS with computer vision:

The term "advanced driver assistance system" (ADAS) refers to a technological tool that helps drivers with basic driving and parking tasks.

Through a secure human-machine interface, ADAS improves vehicle and road safety, reduces traffic accidents, and may ease congestion. In order to identify surrounding impediments or driving errors and take appropriate action, (ADAS) use automated technology, such as sensors and cameras.

The most advanced field now is ADAS, which relies on battery-powered functioning.

ADAS with computer vision applications:
Lane change assist (LCA): aids drivers in changing lanes safely,Majorly based on camera and LIDAR sensors.

Lane departure warning (LDW): To prevent accidents caused by drifting or leaving your lane, lane departure warning is designed to assist you. Majorly based on camera.

Blind spot detection (BSD): mostly using camera and LiDAR sensors.
Traffic signs Recognition: With the use of traffic-sign recognition technology, a vehicle can detect road signs like "speed limit," "children," and "turn ahead", which we will focus in this project.

6.5. Traffic signs Recognition:

The traffic sign recognition technology "sees" road signs and displays them to the vehicle's driver. They are usually shown on a screen in the instrument cluster. This technology typically employs a forward-facing camera mounted behind the windshield to "search" for traffic signs. Some vehicles employ a specialised forward-facing camera for this system, while others employ the same ADAS camera that is used for lane departure warning and other systems.

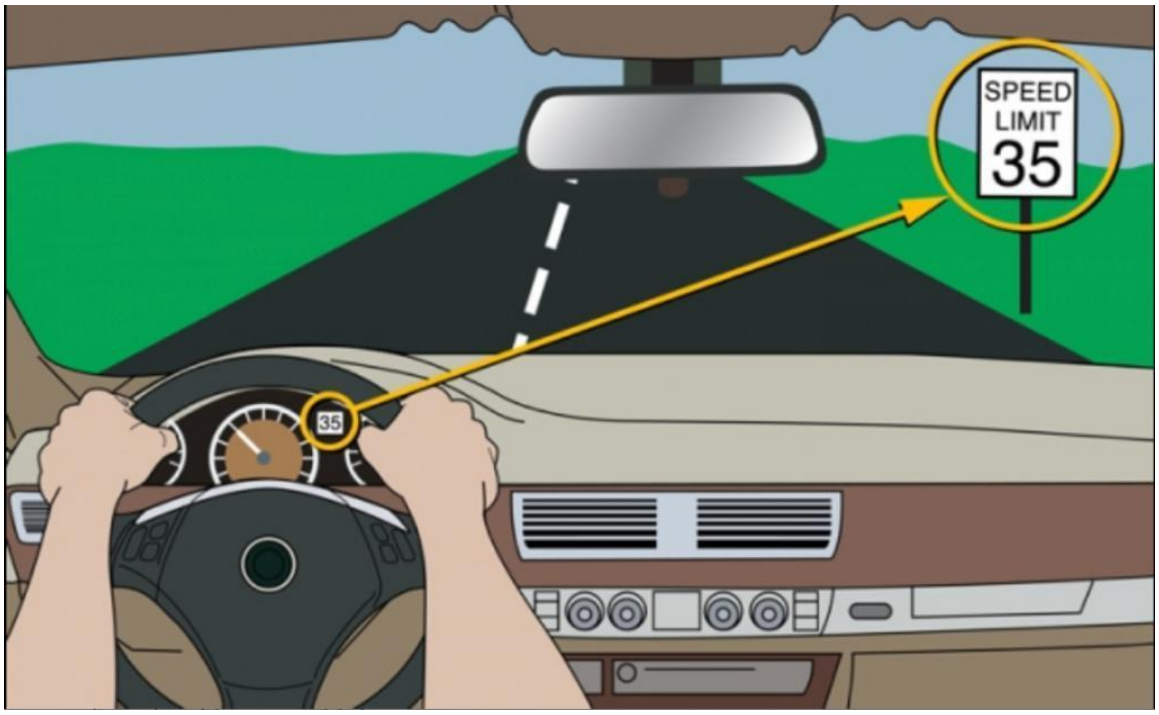


Figure 31Traffic signs Recognition

Some vehicles feature haptic (vibration) or audio warning systems that activate when the driver, for example, exceeds the posted speed limit or enters a "do not enter" road. After displaying a recognised sign, the system may save it to "memory" so that it can be recognised more readily the next time.

6.6. Project idea:

One of the most important technologies in use today is computer vision. The need for increased computational capacity arises from some of these applications, though, which demand real-time processing for videos and images to complete the intended work.
The advanced driver assistance systems (ADAS), one of the most significant computer vision applications that demands real-time processing, where time is a crucial component that needs to be taken into consideration. However, in a system like this where power is important resource, the high computer power required for real-time processing is not always possible.
As a result, there is a growing need for an alternative to the conventional implementation of ADAS computer vision tasks on general-purpose CPUs.
Our approach is to detect different traffic signs using FPGA, which will decrease the recognition time, thus, it can serve the vehicle at high speeds, due to the small recognition time.

6.6.1 What is FPGA?

An integrated circuit that may be modified is called a field programmable gate array (FPGA). The best platform for driving assistance systems is one that has a parallel computing architecture and easy access to local memories. Real-time video processing can be done by an FPGA, allowing for prompt delivery of the necessary warnings to the drivers. In addition, current FPGAs are relatively inexpensive and power efficient when compared to CPU and GPU.
Gate array: Interconnect is done once with a specified function.

FPGAs: Programmable gates and programmable interconnects between gates.

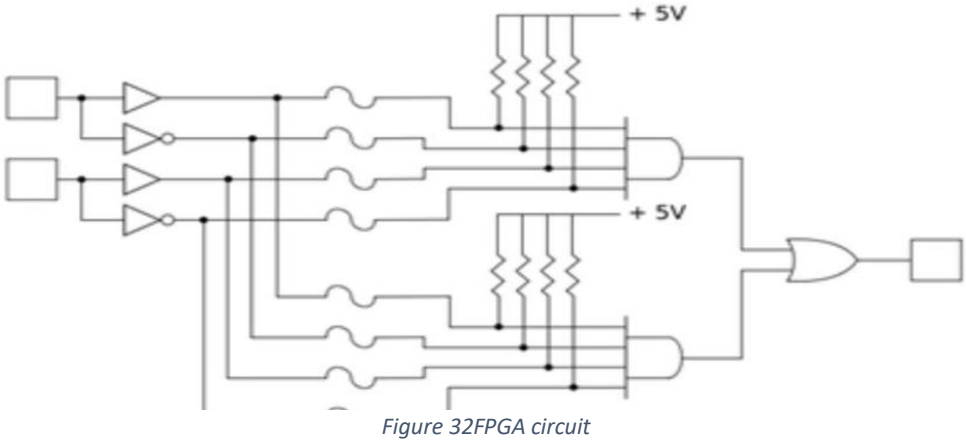
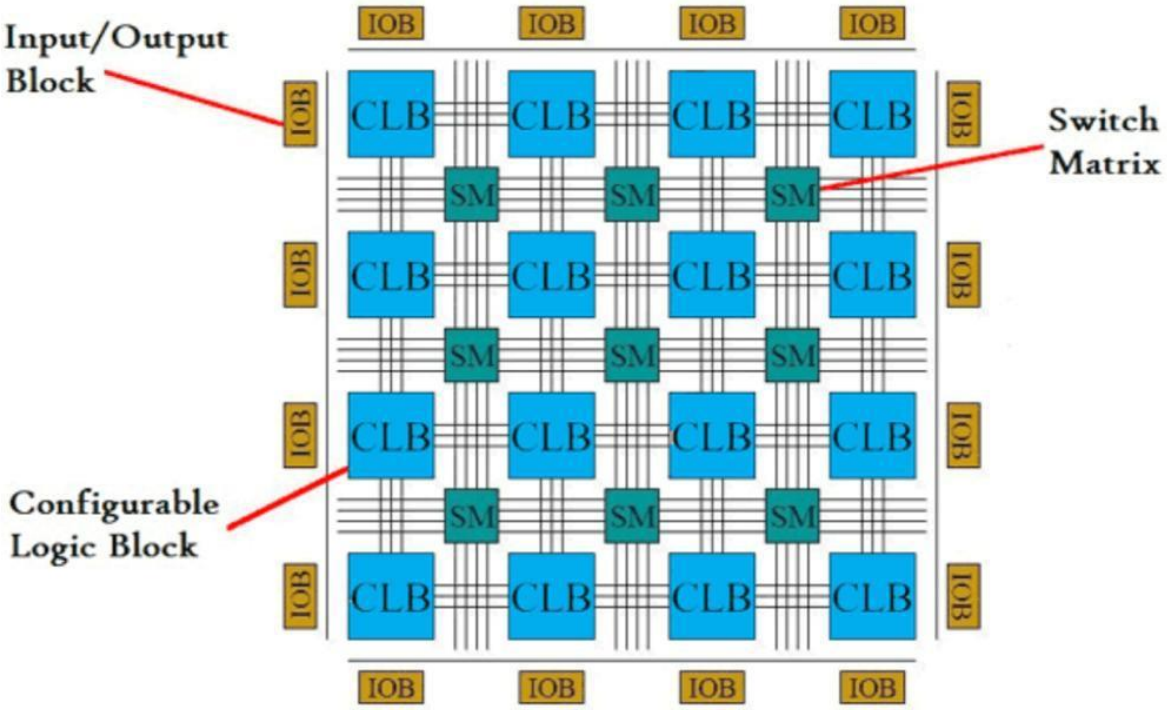


Figure 32FPGA circuit



FPGAs are programmable, off-the-shelf devices that offer a configurable platform for quickly and affordably incorporating unique functionality. They are primarily made up of a collection of programmable logic cells known as configurable logic blocks (CLBs), a programmable interconnection network, and a collection of programmable input and output cells positioned everywhere around the device. They also have a wide range of embedded components, such as high-speed I/O links, clock management units, flip-flops, block RAMs (BRAMs), look-up tables (LUTs), and digital signal processing (DSP) blocks, which are used to perform arithmetic-intensive operations like multiply-and-accumulate.

6.6.2 Why FPGA?

In the literature, various methods for integrating vision algorithms in embedded systems have been discussed. Due to their programmable functionalities, microcontrollers and microprocessors were the only devices utilised for many years. The implementation of vision systems using a microcontroller or microprocessor is widely used in the literature.

Digital signal processors are a similar approach (DSPs). In order to increase processing speed, DSPs have certain parallel processing capabilities as well as the ability to perform one cycle multiply and accumulate operations. When using microcontrollers alone was insufficient for image and audio signal processing, DSPs have traditionally been utilised. It is challenging to attain real-time performance with simply microprocessors or DSPs due to the new cameras' increased resolution and frame rate.

An option is hardware implementation, which can achieve substantially higher computing performance. For a case study, the below table shows how the FPGA implementation of the YOLO algorithm has increased the method's efficiency by reducing the time of execution while keeping a low power consumption rate. The CPU approach takes longer to execute than GPUs but uses less power overall.

The GPU implementation executes the algorithm the fastest but uses the most power. In comparison to the GPU and CPU, the FPGA implementation offers very good execution times with lower power consumption.

	ARM CPU	FPGA	GPU
Platform	ARMv7-A	ZC706	Titan X
Technology	28 nm	24 nm	16 nm
Clock Freq.	Up to 1 GHz	200 MHz	1531 MHz
Num. of Cores	2 cores	—	—
YOLO	430.6 s	0.744 s	0.010 s
Faster R-CNN	Failed	0.875 s	0.062 s
YOLO	1.6 W	1.167 W	230 W
Faster R-CNN	Failed	1.167 W	81 W
YOLO	688.96 J	0.868 J	2.30 J
Faster R-CNN	Failed	1.02 J	5.02 J

Figure 33FPGA comparison

Over ASICs and ASSPs, FPGAs have the following advantages:

- The ability to be reprogrammed can be a great benefit during the design and prototype phases as well as for field upgrades.

- For large-scale tasks, FPGAs have the benefit of optimising performance per Watt of power usage. This makes them a fantastic option as battery-powered device accelerators, which is ideal for our purpose.

- FPGA tools became more appealing to employ for quick development in short timeline applications with the adoption of software-level programming models like the open computing language (OpenCL) standard or MATLAB.

- FPGAs provide a distinct benefit in that they can design customised hardware circuits that are thoroughly pipelined and multithreaded, allowing for real-time processing.

- Numerous configurations - one board, same FPGA, multiple bitstreams

- Products that meet several standards in multiple countries

- Customers can have their features and add-ons customised.

- Simply utilise a new bitstream instead of holding numerous board versions or larger devices to handle all standard/feature variations.

6.7. Project implementation

In order to test the idea, we used MATLAB and ModelSim, which is a multi-language environment developed by Siemens (previously by Mentor Graphics) for the simulation of hardware description languages like VHDL, Verilog, and SystemC. It also has a built-in C debugger. ModelSim can be used separately or in conjunction with Intel Quartus Prime, PSIM, Xilinx ISE, or Xilinx Vivado. The graphical user interface (GUI) or automatically running scripts are used to carry out simulation.

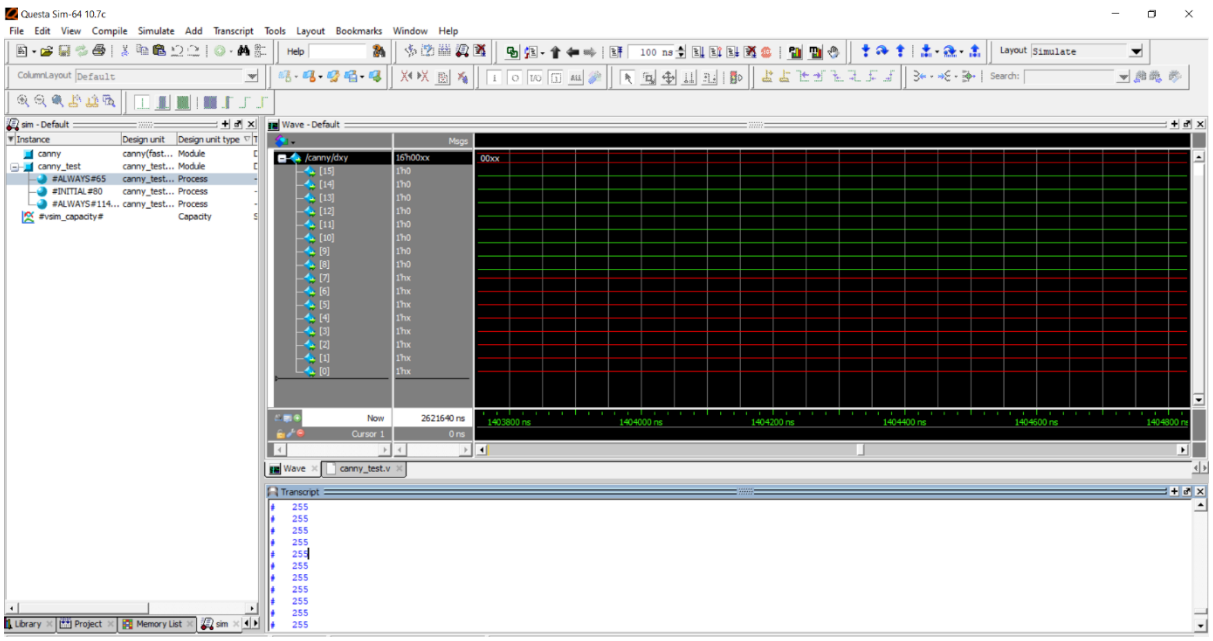


Figure 34ModelSim waveform

The functional block diagram below shows the implementation steps required to achieve the required image.

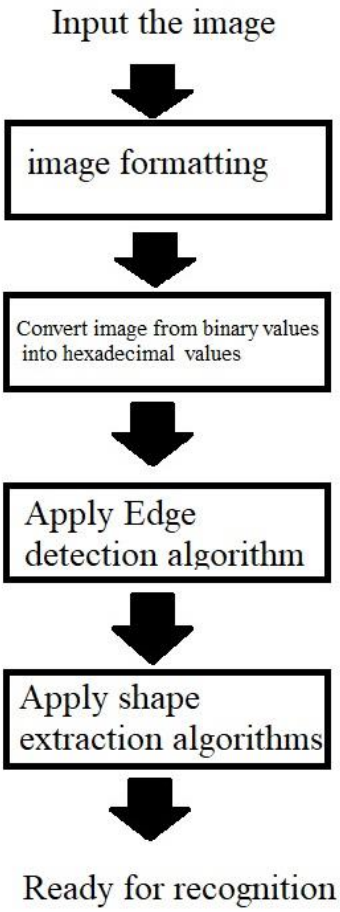


Figure 35) functional block diagram of FPGA circuit

6.7.1. Image Formatting:

Contains the following steps:

- Convert image from RGB into gray scale image:to simplify the process
- Apply Zero padding: Adding zeros is equal to interpolating samples of your spectrum with sinc function. Therefore you will find it looking more smoothly, and it will not affecting frequency resolution in any way.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	34	239	255	255	254	251	212	203	203	203	202	234	255	255	255
0	34	239	255	255	254	251	212	203	203	203	202	234	255	255	255
0	34	239	255	254	254	251	212	203	204	203	202	234	255	255	255
0	34	240	255	254	255	252	212	201	204	202	203	234	255	255	255
0	34	236	252	251	251	247	215	207	208	206	206	233	252	251	251
0	28	200	216	214	213	216	238	244	245	244	243	225	213	214	213
0	27	190	205	203	202	206	246	255	255	255	255	225	201	203	203
0	27	191	205	203	203	206	245	255	255	255	255	225	202	203	205
0	27	191	207	205	203	206	245	255	255	255	255	225	204	205	205
0	28	190	205	203	203	206	246	255	255	254	254	223	201	203	202
0	28	195	210	208	208	210	242	250	249	248	249	225	207	208	208
0	33	231	248	245	246	243	218	211	212	211	211	233	246	245	246
0	34	240	255	255	255	251	211	202	203	203	202	233	255	255	255
0	34	239	255	255	254	250	212	202	204	204	203	235	255	255	255
0	34	239	255	255	254	251	213	202	203	203	203	234	255	255	255
0	34	240	255	255	255	251	212	201	202	202	203	233	255	255	255
0	32	231	248	247	247	244	219	210	212	211	210	233	248	247	247
0	28	193	209	207	207	210	243	252	251	250	249	226	206	207	207
0	27	190	205	203	202	205	245	255	255	253	254	223	201	203	203
0	27	191	206	204	203	206	246	255	255	254	255	223	202	204	204
0	27	191	206	204	203	206	246	255	255	254	254	224	203	204	203
0	27	190	204	202	202	206	247	255	255	255	255	223	201	203	203
0	28	200	215	213	214	217	239	245	244	243	244	228	213	213	213

Figure 36Image matrix after applying zero padding

- Resize the image(60*60)

6.7.2. Convert image from binary values into hexadecimal values:

The FPGA deals only with hexadecimal values.

```
0
0
0
0
22
ef
0
22
ef
0
0
0
22
ef
ff
22
ef
ff
0
0
0
ef
ff
ff
ef
ff
ff
0
0
0
ff
ff
```

Figure 37Image pixels after Converting it from binary values into hexadecimal

6.7.3. Apply Edge detection algorithm:

We used the Sobel operator which evaluates a 2-D spatial gradient on an image and highlights high-frequency regions that correspond to edges. It is usually used to determine the approximate absolute gradient magnitude at each location in a grayscale image input.

Here, we still choose a window size of 3 by 3, then apply two kernels on the sliding window separately and independently. We also need a window to perform the scanning work. The equation below represents the x &y directed kernel:

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

The resulting gradient approximations can be combined to give the gradient magnitude, using:

$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

The following image is applied to the simulution, which apply the previous calculations on it.

Figure 38Input image

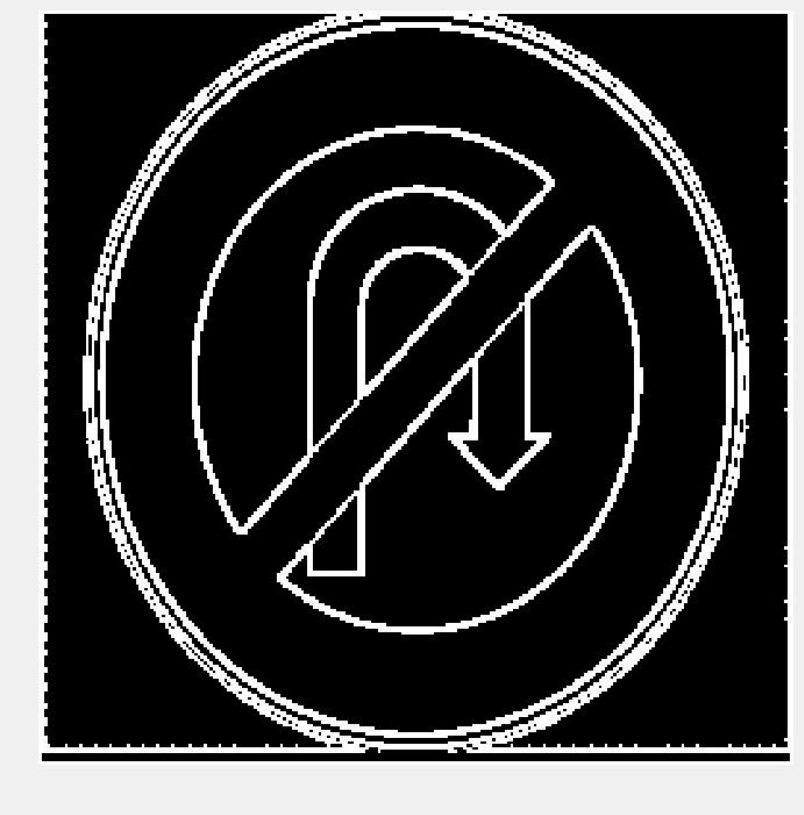


Figure 39 the output image of the simulition

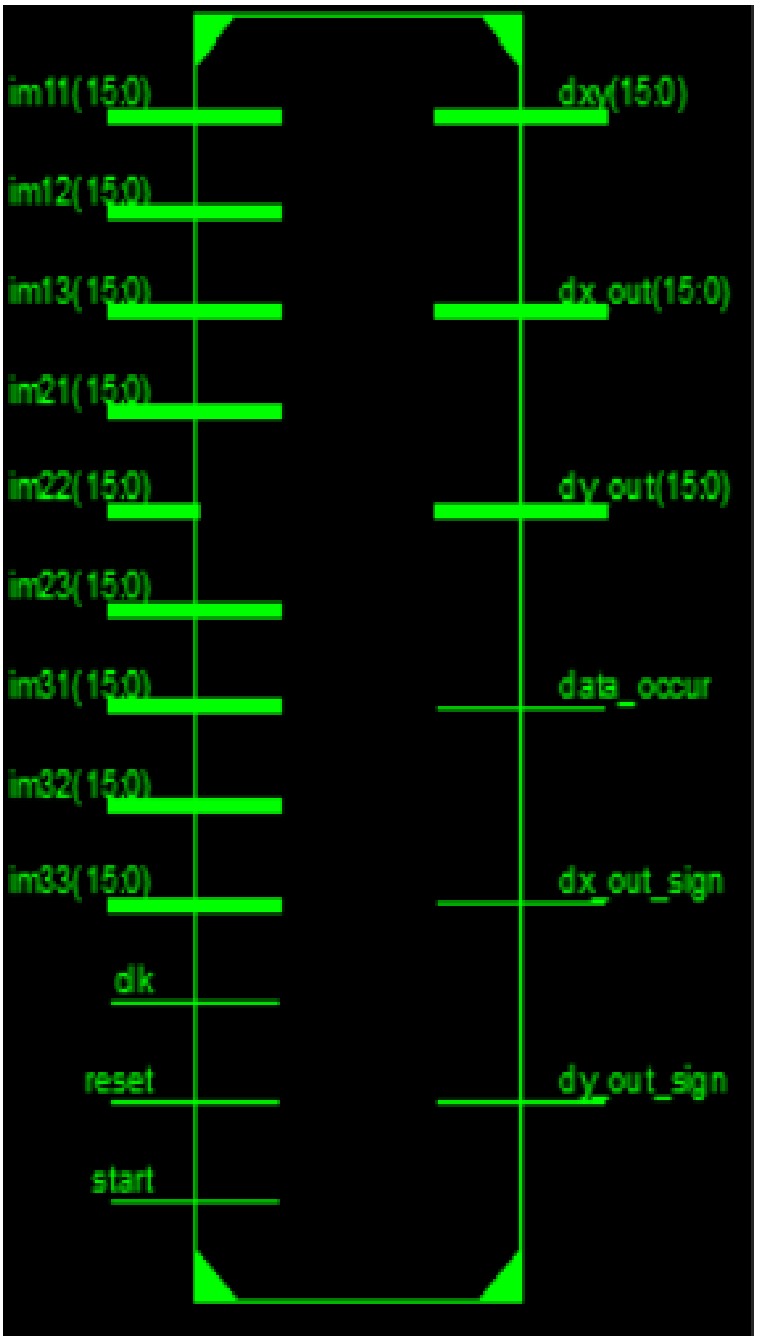


Figure 40Overall RTL schematic

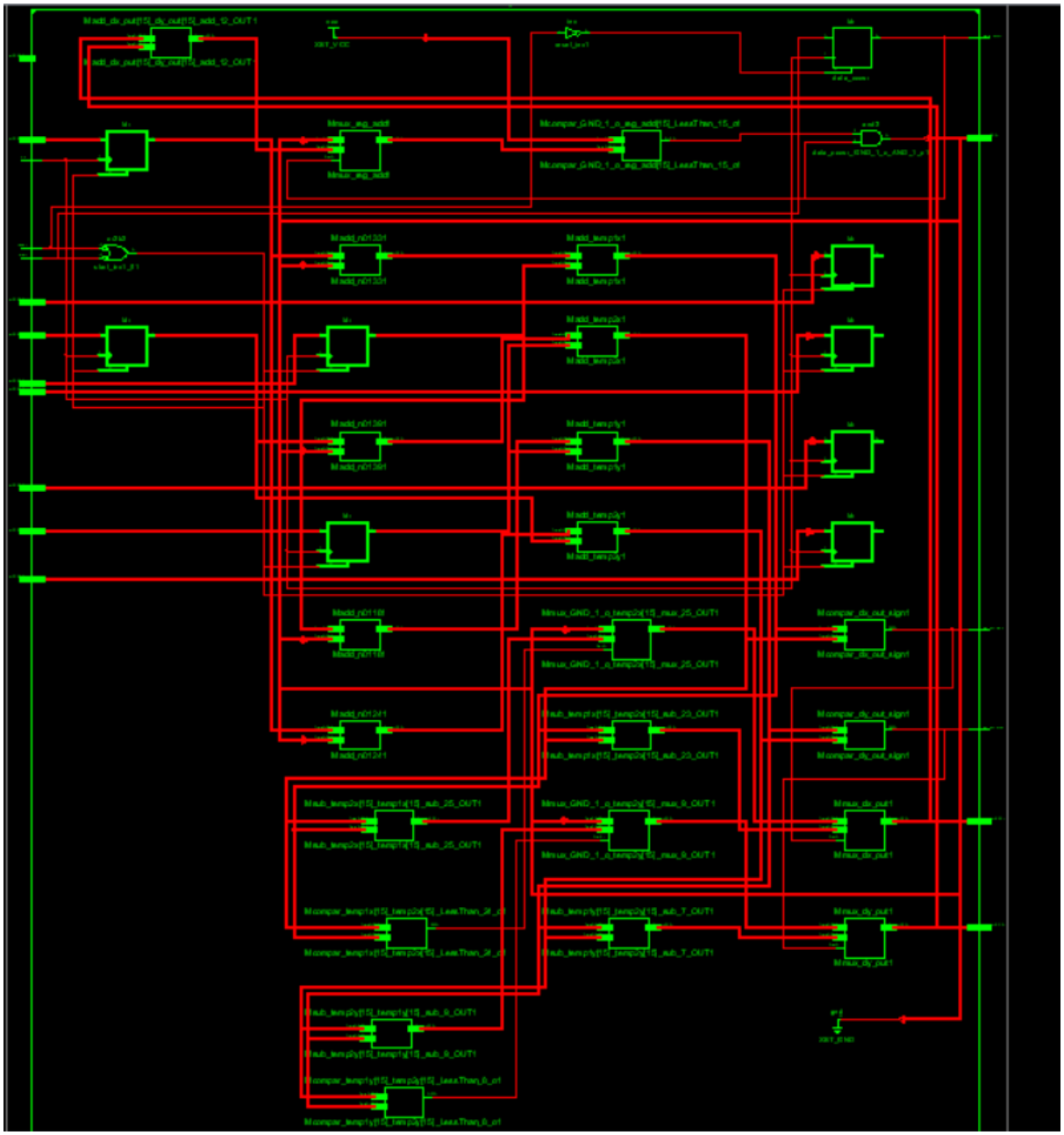


Figure 41RTL schematic

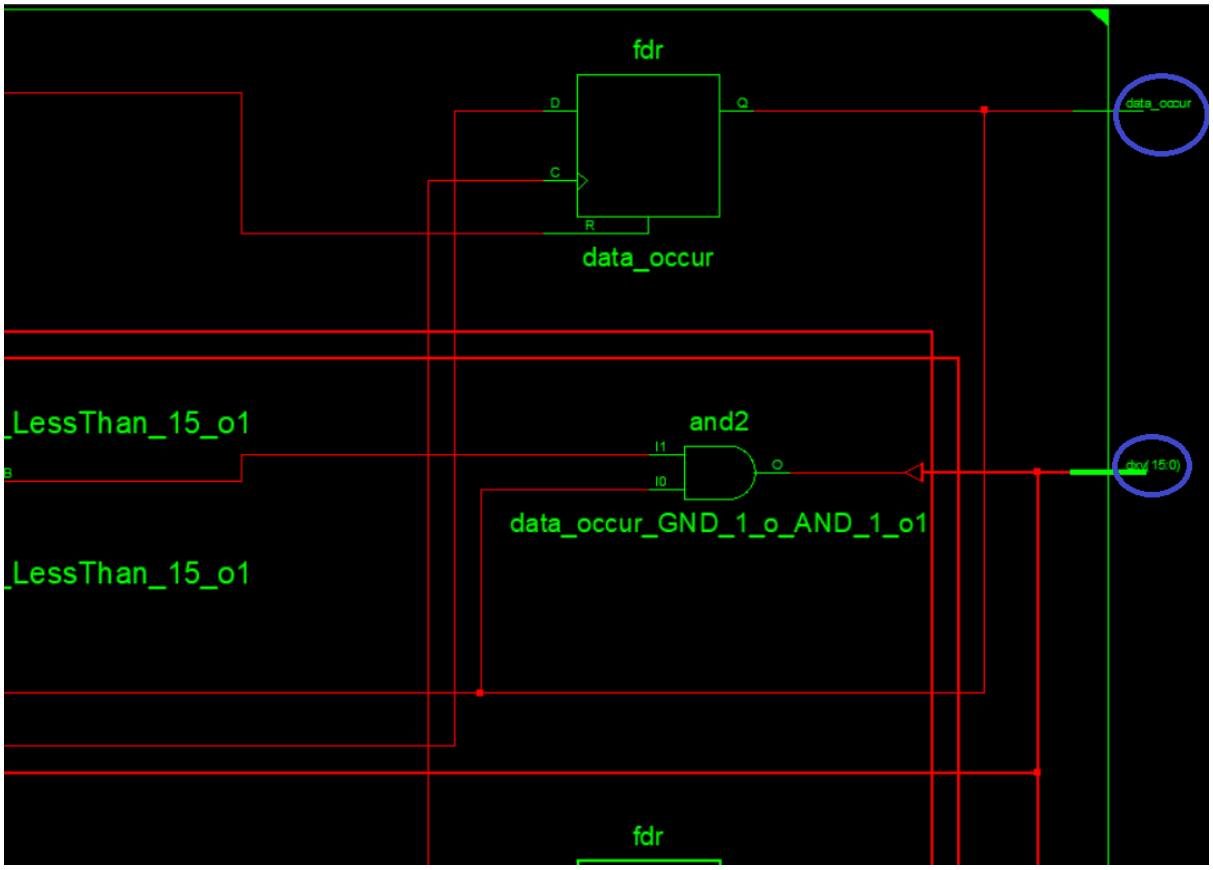


Figure 42RTL schematic

D xy (15:0) component: represents gradient magnitude (G).

Data Occur: it becomes one if the data inserted, to start calculations.

6.7.4. Apply shape extraction algorithms

Start with the circle:

CHT is applied for circle detection. After calculating the edge image for each potential radius, CHT constructs an accumulator space that represents the centre votes for each pixel in the image. The radius and the circle's centre coordinates are thus the elements of the parameter space. Alternatively said, the parameter space is three dimensional (xc,yc,r). The parametric equation for a circle with centre (xc,yc) and radius r is provided in the following equation ;

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

Circular traffic signs, however, could seem elliptical due to perspective distortion. The ROI and centre of the traffic sign cannot be accurately detected with CHT in these elliptical objects. Figure(25) shows an elliptical version of a circular traffic sign. Following the use of CHT, the sign's centre and the resulting detected circle are highlighted in red.

Figure 43Detected Circle After Applying CHT (example)

Ellipses cannot be detected by CHT. As a result, it has to be extended in order to recognise ellipses. The equation below gives the ellipse's parametric equation:

$$(x - x_c)^2 + k.(y - y_c)^2 = r^2$$

The "k" parameter is the only distinction between the parametric equations of a circle and an ellipse. Ellipse turns into a circle when k is equal to one. If k is less than 1, the ellipse's y diameter is greater than its x diameter; if k is bigger than 1, the ellipse's x diameter is greater than its y diameter. Figure (26) illustrates how the "k" parameter has an impact.

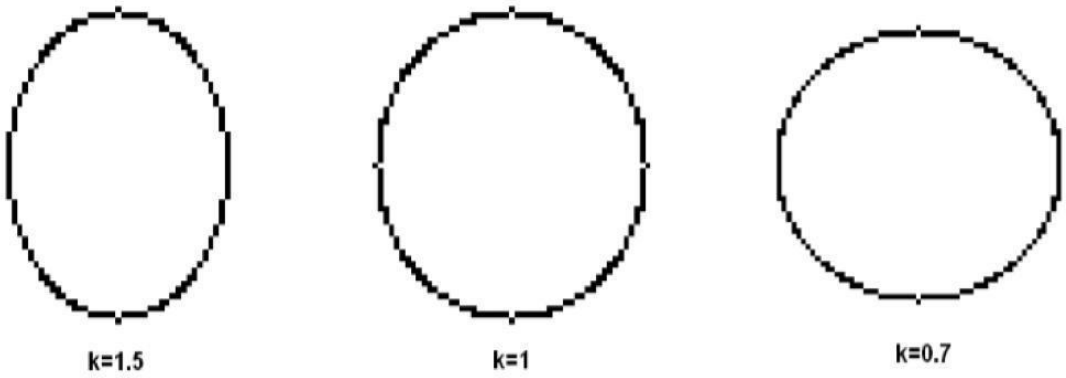


Figure 44 effect of the "k" parameter

It is simple to expand the CHT to identify ellipses because the parametric equation for the ellipse is known. The quantity of parameters is the biggest drawback of ellipse detection in comparison to circle detection. Circle detection uses a three-dimensional (xc,yc,r) parameter space, whereas ellipse detection uses a four-dimensional parameter space (xc,yc,k,r). In the Hough Transform, adding more parameters immediately raises the level of complexity. Line detection has an O(n) of 2, circle detection has an O(n) of 3, and ellipse detection has an O(n) of 4.

Some assumptions are made in order to speed up the ellipse detection procedure. which are

In the 60x60 pixel image patch, the traffic sign's radius must be less than 10 pixels.

The ellipse's long radius to short radius ratio cannot be more than 2.

These factors are plausible. Since a traffic sign can only be reliably recognised if it is as close to the car as possible, if the sign is spotted when its radius is less than 10 pixels, it will likely be detected again when its radius is higher than 10 pixels.

In addition, for the second assumption, perspective distortion makes it practically difficult to see a circle as an ellipse with a long-to-short-radius ratio greater than two when moving through traffic.

The discovered ROI and centre of the sign are displayed in Figure (25) after ellipse detection has been applied to the image. The ROI and centre of the sign are detected more accurately with the "k" parameter in ellipse detection than they are with CHT.

RECTANGULAR SIGN DETECTION:

Four lines should be identified in total when detecting rectangular signs. The Hough Transform's line identification problem can be reduced to a peak histogram finding problem utilising x and y projections because the angles of the lines are between 0 and 90 degrees and around. Due to the margins of the rectangular traffic sign's border, two peaks should be seen in the x and y projections. Figure (27) shows one as an illustration. The edges of the traffic sign are represented by two peaks in this figure in both picture projections. After determining the rectangle's borders, the centre coordinates can be determined by averaging the boundary coordinates. Figure 28 displays the identified border lines.

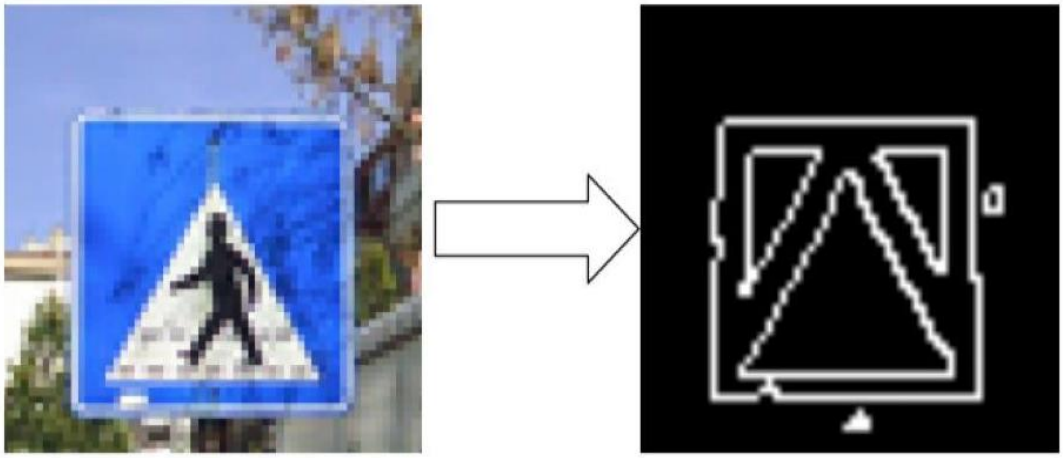


Figure 45 Rectangular Sign and Edge Image

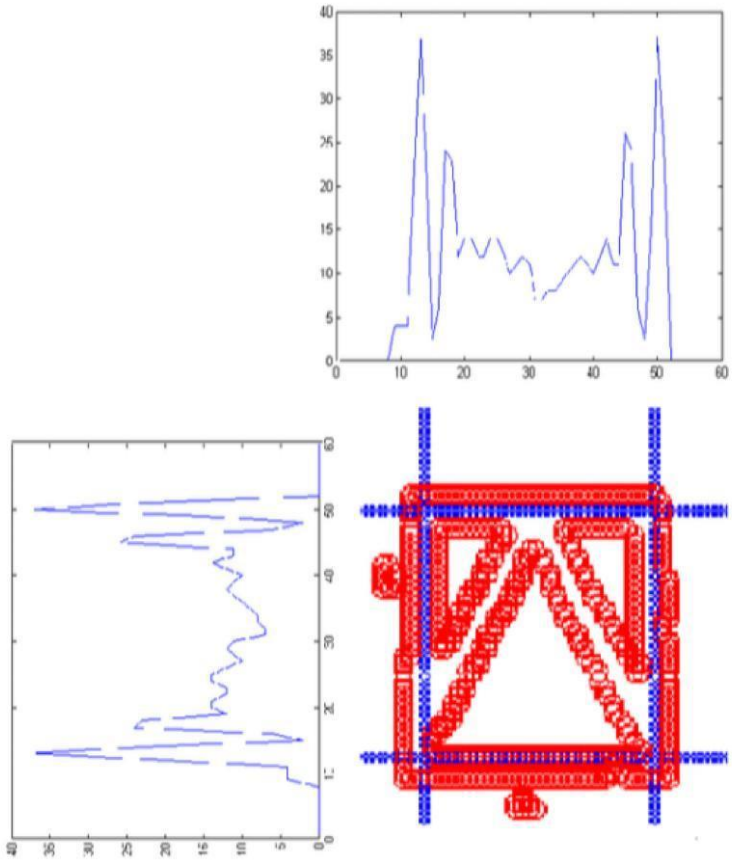


Figure 46 X &Y projections of the Edge Image

7. V2X (vehicle to everything) and MQTT cloud

7.1 What is V2X

V2X is one of the latest technologies to be used in automotive safety. V2X stands for “vehicle-to-everything” communications, and it means that the vehicle can communicate with other vehicles, pedestrians, and roadside infrastructure.

This communication happens via Wi-Fi technology, but because it uses a dedicated short-range (DSRC) frequency, V2x has a range of 800 meters compared with 50-100 meters for radar or cameras. As a result of this extended reach, V2X offers key safety benefits and is being used in tests to develop autonomous driving systems. The cost of installing V2X is expected to decrease as it becomes more popular in the automotive industry.

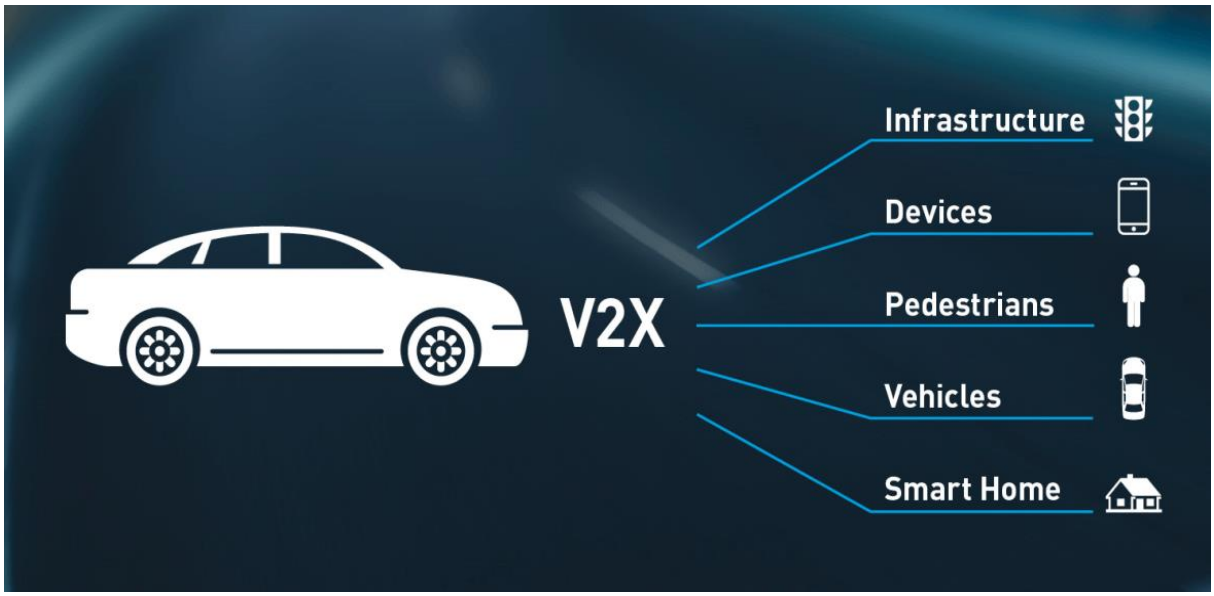


Figure 47 V2X Communication

V2X is the umbrella term for the communication between vehicles and their surroundings, including roadside infrastructure, other vehicles, and pedestrians.

V2X is the umbrella term for the communication between vehicles and their surroundings, including roadside infrastructure, other vehicles, and pedestrians. V2X technology can be used to exchange information about traffic conditions in real-time.

It is estimated that 80% of accidents could be prevented or mitigated by V2X.

V2X communications can help drivers take control of their vehicles if necessary

The range of V2X is far greater than radar or cameras, which typically have ranges of up to 50-100 meters. In Europe, there are already some tests taking place in urban areas On the road to autonomous driving, V2X communications play a key role. V2X is a key component of autonomous driving applications. It refers to the communication between vehicles and their surroundings, including other vehicles, pedestrians and road infrastructure.V2X stands for the vehicle to everything (or vehicle to anything). The term V2X covers all different types of communications between vehicles and their surroundings. This includes connections between cars (V2V), between cars and road infrastructures (V2I), between cars and pedestrians (V2P), etcetera.

V2X communications can help drivers take control of their vehicles if necessary. V2X communications can help drivers take control of their vehicle if necessary. In the event of an accident or other emergency, for example, V2X will allow vehicles to receive information from surrounding vehicles and infrastructure in order to warn drivers about hazards or get them on the road again after a crash. The technology is also highly useful for responding to hazards before they become problems,

For example, if a driver sees an obstacle on the road ahead that might not be visible by traditional means (like fog), he or she can simply tap into V2X data from surrounding cars and make adjustments accordingly.

V2X has a range of up to 800 meters, compared with only 50-100 meters for radar or cameras.

V2X is a radio communication system between vehicles and their surroundings. It provides vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-anything (V2X) communications. V2X broadcasts information about the road such as traffic congestion, accidents, bad weather conditions, and more. This means that when you're driving your car behind another one with a V2X sensor, you'll be able to see what's going on ahead of you without having to rely on the driver in front of you or any other external sources like radar or cameras. That's good news for safety because it allows for faster reaction times in emergency situations like accidents or bad weather conditions.

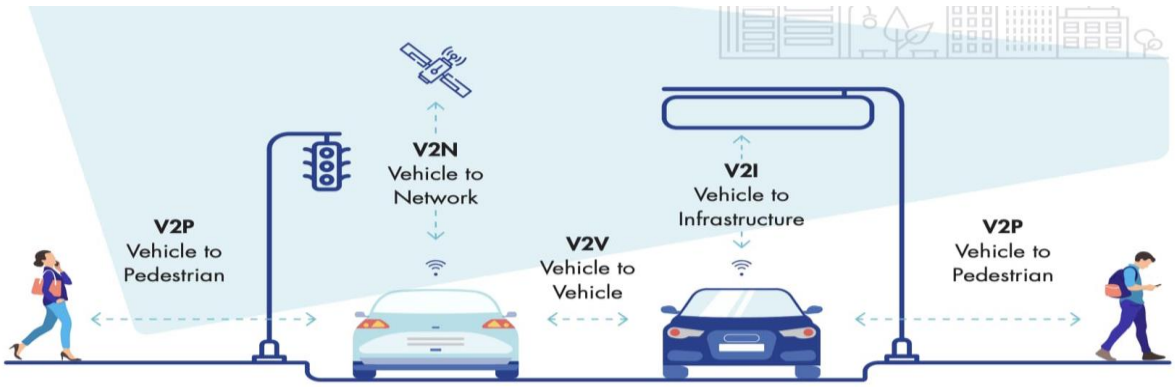
7.2 How Vehicle to Everything (V2X) Works

In a V2X communication system, data from the vehicle's sensors and other sources is transmitted across high-bandwidth, high-reliability lines, enabling communication with other vehicles, infrastructure like parking spots and traffic signals, and smartphone-tossing pedestrians.

The technology increases driver awareness of potential hazards and helps lower the severity of injuries, road accident fatalities, and collision with other cars by sharing information, such as speed, with other entities around the vehicle.

The use of alternate routes suggested by the technology, traffic warnings, and the identification of parking spots all help to improve traffic efficiency.

7.3 Components of V2X Technology



1. V2I, or "Vehicle-to-Infrastructure," is the transmission of data from a vehicle to equipment that is installed next to a road and is typically referred to as a "roadside unit" (RSU). V2I is frequently utilised to inform drivers about traffic conditions and emergencies.
2. "Vehicle-to-Network," also known as "vehicle-to-cloud," is the term for when a vehicle connects to a network to access cloud-based services.
3. Data transfer between vehicles is referred to as V2V, or "Vehicle-to-Vehicle." Information conveyed by V2V technology can come from automobiles a few hundred metres ahead or even cars hidden behind trucks or buildings, which is more information than what sensors can give the car.
4. "Vehicle-to-Pedestrian," or V2P, refers to the data transfer between a car and pedestrians.

7.4 What is MQTT?

The MQTT (MQ Telemetry Transport) lightweight open messaging protocol offers network users with limited resources an easy way to share telemetry data stored in low-bandwidth contexts.

Machine-to-machine (M2M) communication is enabled through the protocol, which uses a publish/subscribe information exchange structure.

MQTT was developed to operate in an embedded environment where it could offer a dependable, efficient path for communication. It was created as a low-overhead protocol to adjust bandwidth and CPU limitations. MQTT is an excellent option for wireless networks that often face latency variations owing to bandwidth restrictions or poor connections because it connects devices with a small code footprint. The protocol can be used in a variety of sectors, including the automotive, energy, and telecommunications industries.

MQTT, the most widely used open-source protocol for establishing connections between internet of things (IoT) and industrial IoT (IoT) devices, was initially a patented protocol used to communicate with SCADA systems in the oil and gas industry. However, it has gained popularity in the smart device industry.

MQ refers to an item named IBM MQ, while TT in MQTT stands for Telemetry Transport. Despite sometimes being referred to as The Message Queue Telemetry Transport, MQTT communication does not use message queuing.

7.5 How does MQTT work?

The publish/subscribe (pub/sub) communication mechanism offered by MQTT is a replacement for the conventional client-server architecture that communicates with endpoints directly and aims to increase the available bandwidth. In the pub/sub paradigm, however, the client that sends a message (the publisher) is separated from the client or clients that receive the messages (or the subscribers). This is due to the fact that brokers manage the links between the publishers and subscribers instead of the parties themselves, who are neither publishers nor subscribers.

Depending on whether the client is publishing messages or has subscribed to receive messages, MQTT clients can be either publishers or subscribers. You can use the same MQTT client to implement these two features.

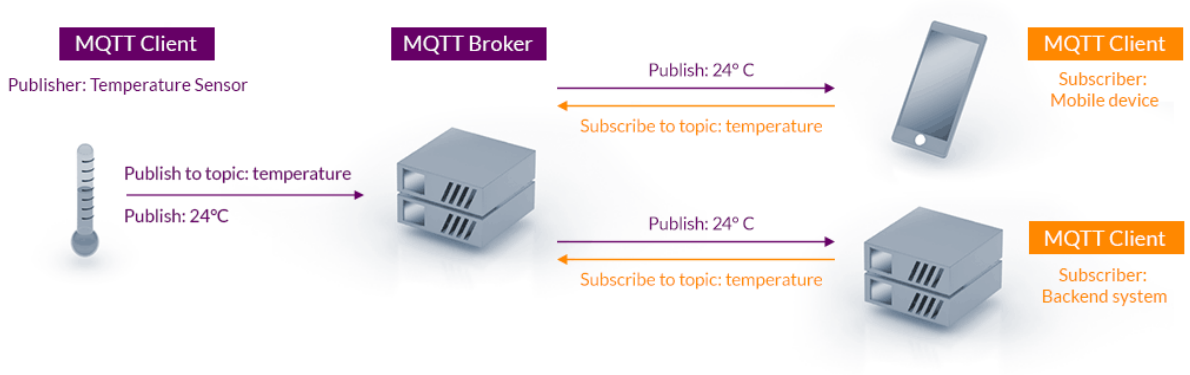
When a device (or client) wants to send data to a server (or broker) it is called a publish. When the operation is reversed, it is called a condone. Under the pub/sub model, multiple clients can connect to a broker and subscribe to topics in which they are involved.

If the connection from a subscribing client to a broker is broken, then the broker will buffer messages and push them out to the subscriber when it is back online. If the link from the publishing client to the broker is detached without notice, then the broker can close the connection and send subscribers a cached message with instructions from the publisher.

An IBM writing describes the pub/sub model: "Publishers send the messages, subscribers receive the messages they are interested in, and brokers pass the messages from the publishers to the subscribers. Publishers and subscribers are MQTT clients, which only communicate with an MQTT broker.

MQTT clients can be any device or implementation (from microcontrollers like the Arduino to a full application server hosted in the Cloud) that runs an MQTT library."

MQTT Publish / Subscribe Architecture



7.6 What we did?

We did a v2c (vehicle to cloud module) consist of:

The familiar nodeMCU esp8266 WIFI module combined with ADXL345 (accelerometer). It communicates with the nodeMCU throughout I2C to provide x,y, and z coordinates that came from ADXL345 (accelerometer).

This inexpensive combination provides a powerful stable communication module. nodeMCU esp8266 WIFI module acts as the brain for this whole module. The module transfers data between the ADXL345 which is connected to the body of the car and the communication ECU to extract car speed and Acceleration.

the WIFI module is connected to the internet and the MQTT broker and subscribes to topics and also publishes messages that contain the acceleration coordinates of the car, this message is sent to the cloud by the (Publisher Client) and then forwarded by use of the broker to the (Subscriber Client) which is can be mobile phone application or desktop application.

7.6 .1 Component of V2C

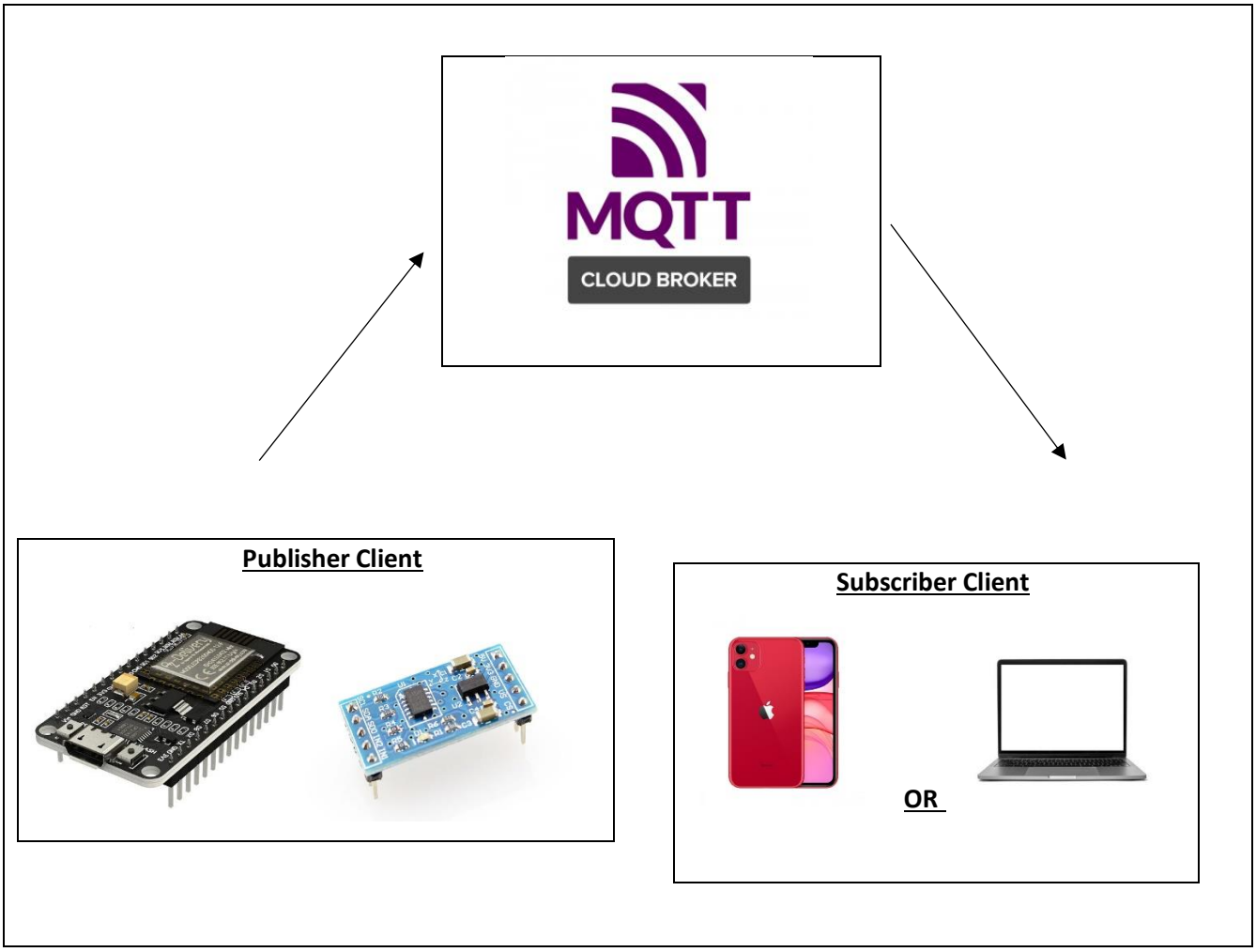
1. ESP8266

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V

- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- Clock Speed: 80 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Small Sized module to fit smartly inside your IoT projects

2. ADXL345

- 2.0-3.6VDC
- Supply Voltage
- Ultra-Low Power: 40uA in measurement mode, 0.1uA in standby 2.5V
- Tap/Double Tap Detection
- Free-Fall Detection

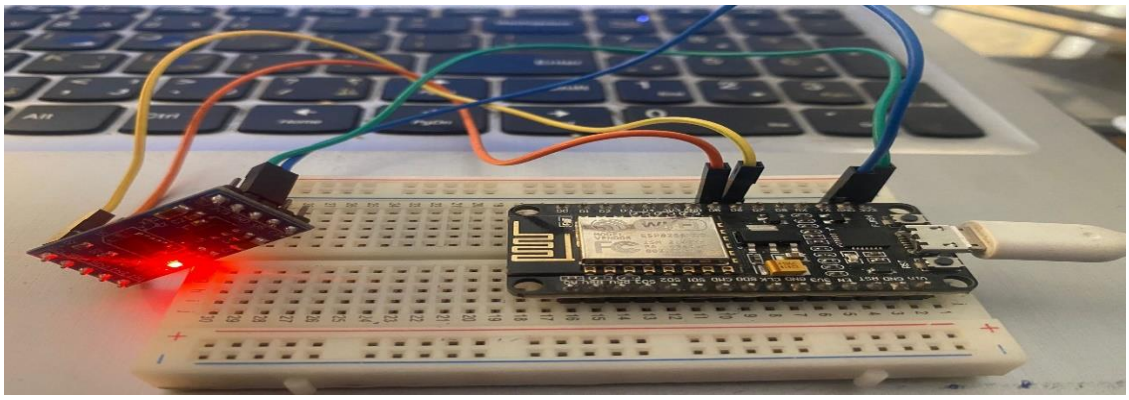

MQTT
CLOUD BROKER
Publisher Client
Subscriber Client
OR

7.6 .2 Steps of Implementation:

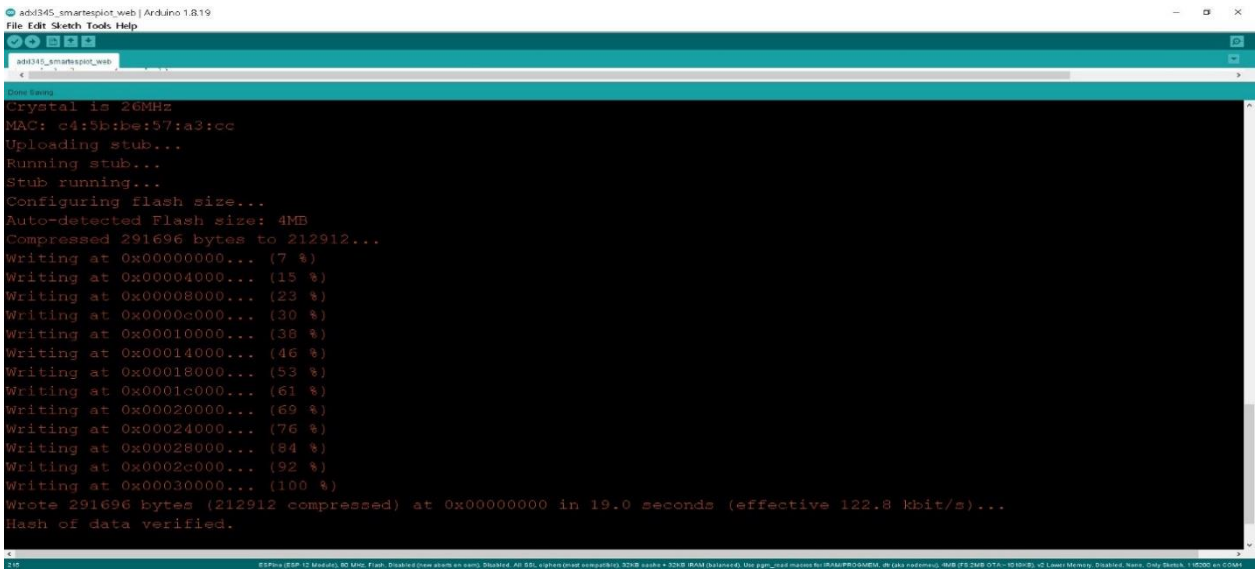
- Step1: Wire connections

ADXL345	NodeMCU
SCL	D6
SDA	D5
GND	GND
VCC	3.3V

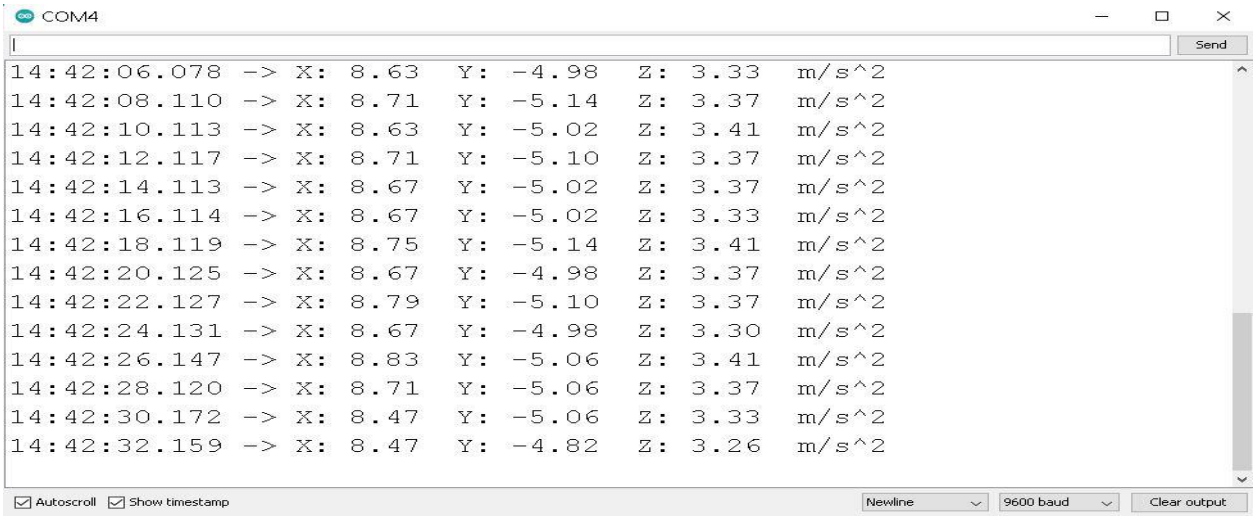
As shown in the figure below



- Step2: Upload the code to the NodeMCU (we used Arduino IDE)

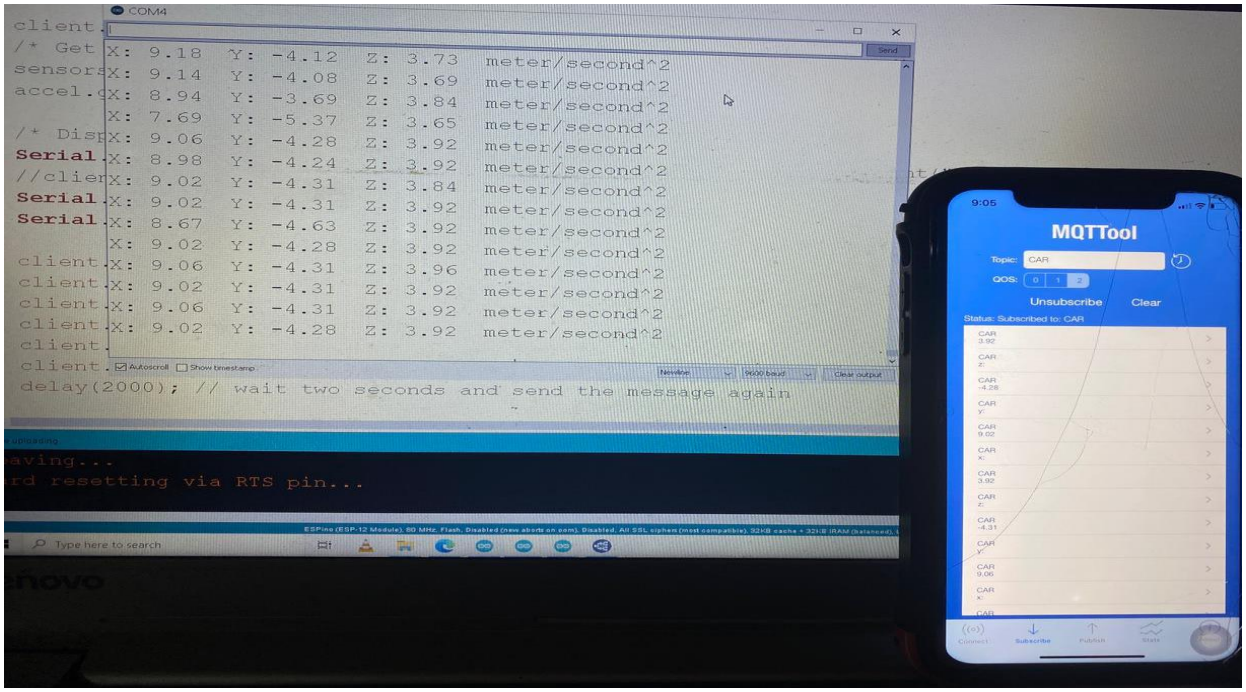


- Step3: Open the serial Monitor in the Arduino IDE to see the sensor's reading



- Step4: Open the mobile phone application Mqttool on an IOS device to see the uploaded readings from the cloud

- Step 5: comparing the reading between the serial monitor and the mobile application



8. References

1. Velez, G.,. A reconfigurable embedded vision system for advanced driver assistance. 2014.
2. Bilal, Muhammad. Resource-efficient FPGA implementation of perspective transformation for bird's eye view generation . 2019.
3. ALINX AX309] XILINX Spartan-6 XC6SLX9 FPGA Development Board LX9 - Spartan6/7 - ALINX.
4. Singh, R. and Ranasinghe,. Accelerating Computer Vision on mobile embedded platforms. 2016.
5. Kortli, Y., Marzougui, M. and Atri, M. . Kortli, Y., Marzougui, M. and Atri, M., 2016. Efficient implementation of a real-time lane departure warning system. 2016 International Image Processing,.
6. Zhao, J.,. Video/Image Processing on FPGA. 2015.
7. MATLAB and Simulink. MathWorks. (n.d.). https://www.mathworks.com/.
8. HASAN IRMAK,"REAL TIME TRAFFIC SIGN RECOGNITION SYSTEM ON FPGA";THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF
9. MIDDLE EAST TECHNICAL UNIVERSITY.
8. Jin Zhao,A .Thesis,"Video/Image Processing on FPGA";WORCESTER POLYTECHNIC INSTITUTE
9. Ros website,http://wiki.ros.org/rosserial
10. Ros website, http://wiki.ros.org/hector_slam
11. Ros website , http://wiki.ros.org/navigation/Tutorials
12. Roswebsite ,http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment
13. Ros website ,http://wiki.ros.org/ROS/Installation
14. "Vehicle-To-Vehicle Communication Technology For Light Vehicles" (PDF). www.google.com. p. e10. Retrieved 2019-12-02.
15. ^ "IEEE 802.11p-2010 - IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments". www.google.com. Retrieved 2021-01-08.

16. ^ Jump up to:a b c d e An assessment of LTE-V2X (PC5) and 802.11p direct communications technologies for improved road safety in the EU.(http://5gaa.org/wp-content/uploads/2017/12/5GAA-Road-safety-FINAL2017-12-05.pdf)
17. ^ Jump up to:a b c White Paper on ITS spectrum utilization in the Asia Pacific Region (http://5gaa.org/wp-content/uploads/2018/07/5GAA_WhitePaper_ITS-spectrum-utilization-in-the-Asia-Pacific-Region_FINAL_160718docx.pdf)
18. ^ C-ITS: Three observations on LTE-V2X and ETSI ITS-G5—A comparison (https://www.nxp.com/docs/en/white-paper/CITSCOMPWP.pdf)
19. ^ Heterogeneous Vehicular Networking: A Survey on Architecture, Challenges, and Solutions(https://doi.org/10.1109/COMST.2015.2440103)
20. ^ EN 302 663 Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band (http://www.etsi.org/deliver/etsi_en/302600_302699/302663/01.02.00_20/en_302663v010200a.pdf)
21. ^ The Case for Cellular V2X for Safety and Cooperative Driving (http://5gaa.org/wp-content/uploads/2017/10/5GAA-whitepaper-23-Nov-2016.pdf)