Home Car Wash Control System User Manual

Document Version: 1.0

Version	Description	Date
1.0	Release	28th August 2021

Table of Contents

List of Figures

List of Tables

1. Introduction

What is Home Car Wash Control System

The **Home Car Wash Control System** is the automated process of washing a car. It has different components in it which include

- **Graphical User Interface** to send commands to the system related to car wash process as well as to view the status and the progress of the car wash process.
- **Arduino Mega 2560** to control the wash process and send parameters to GUI.
- **Water pump** and **soap pump** to spray water and soap mixture on car.
- **Relays** to control water pump, soap pump and inverter motor.
- **Temperature, water** and **soap level** detection sensor
- **Raspberry pi 4** to display GUI to the user.

How it works?

A brief introduction related to the working of Home Car Wash Control System is given in this heading with details covered in the next part. On the start of the process, the GUI is displayed to the user from which user can select different parameters related to the process which include:

- Car type
- Total Cycles for the wash process
- Total Water Spray Cycles
- Total Soap Spray Cycles
- Delay between water and soap spray
- Delay between soap and next water spray

Then on pressing the start button, the system finds its home position, checks the temperature, water level and soap level and if it satisfies the optimal conditions then the process starts with all the progress displayed on the GUI. Once the war cash process is completed, the user is promoted to the next car settings and once those settings are done,

the process repeats its self for the new car.

2. Setting up Raspbian on Raspberry Pi 4

What is Raspberry pi 4

The Raspberry Pi is a microcontroller that uses python language and supports parallel processing with Linux based Raspbian operating system installed on it. It's a single computer board equipped with a Quad-core Cortex-A72 processor, which requires less processing energy and saves room by maintaining costs low. The Raspberry Pi 4 module is shown in figure below.

Figure 1 Raspberry Pi 4

It has distinct ports in parallel for various operations. There are 4 USB ports (2 USB 3.0 ports; 2 USB 2.0 ports) and one Ethernet port for connecting Raspberry Pi to the port, Raspberry Pi has its own Wireless Fidelity and Bluetooth port (Wi-Fi). Raspberry Pi utilizes the operating system separated known to be Raspbian. It has two micro HDMI ports, audio / video and camera connector to connect the screen. The 5-volt source is linked. A 400MHz video core IV multimedia is provided by Raspberry Pi. The SONAR, light emitting diode (LEDs) and engine driver pins are used to link GPIO pins. Two models A and B are available for the raspberry pi. In short, Raspberry pi 4 is a good processor for extensive and graphical tasks.

Hardware required

- One raspberry pi 4 with 1GB or 2GB ram
- One micro SD-card (16GB or 32 GB)
- One monitor with keyboard and mouse or laptop with HDMI display
- One micro HDMI to HDMI cable for laptop or one micro HDMI to VGA cable for monitor.
- One USB to Type C cable for powering up raspberry pi.
- One 5v/ 3A power supply.

Steps for installing raspbian on raspberry pi

- Open the following link to download the raspbian OS for raspberry pi: https://www.raspberrypi.org/software/operating-systems/
- Different versions of OS are available; choose any of them according to the free space available in the SD-Card of raspberry pi.

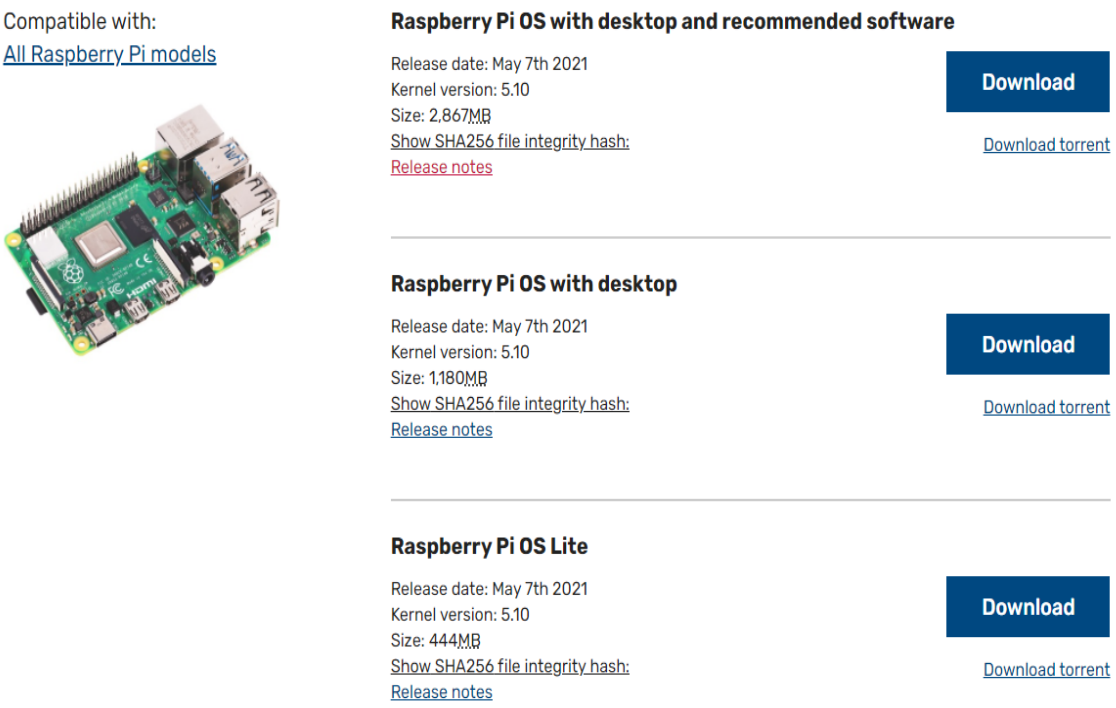


Figure 2 OS available for Raspberry Pi

- Click on the download button to start downloading disk image file of selected OS.

- After downloading the disk image, the next step is to boot the disk image file on SD-card. For this download **"Rufus"** on the laptop using following link https://rufus.ie/en/
- Download, install and open Rufus.
- Insert the micro SD-card in the laptop to boot it.
- Select the inserted micro SD-card in the **device option** in the Rufus.
- Select the disk image file of raspberry pi OS in the boot selection option.

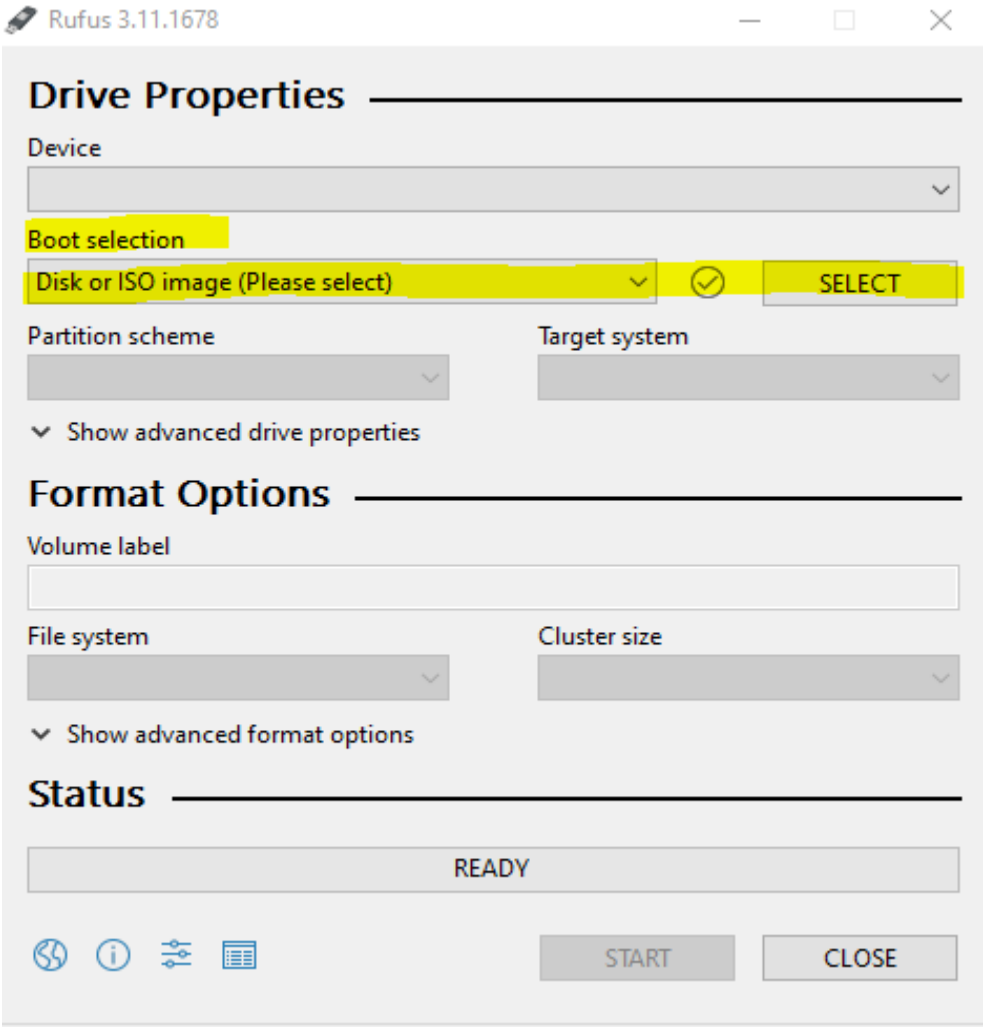


Figure 3 Rufus interface

- Now insert the micro SD-card to the slot available on the raspberry pi.
- Connect the one end of the USB to Type C cable to the Type C slot present in the raspberry pi and the other end to the power supply.

- Connect the “micro HDMI” slot of the micro HDMI to VGA cable to the “HDMI0” port on the raspberry pi and the other end to the VGA of the monitor.
- Power on the supply.
- After some time, the raspberry pi OS desktop environment will be displayed on the monitor screen.

Steps for configuring raspberry pi and connecting to network

- Following OS environment will be displayed on the screen after the OS is successfully installed:

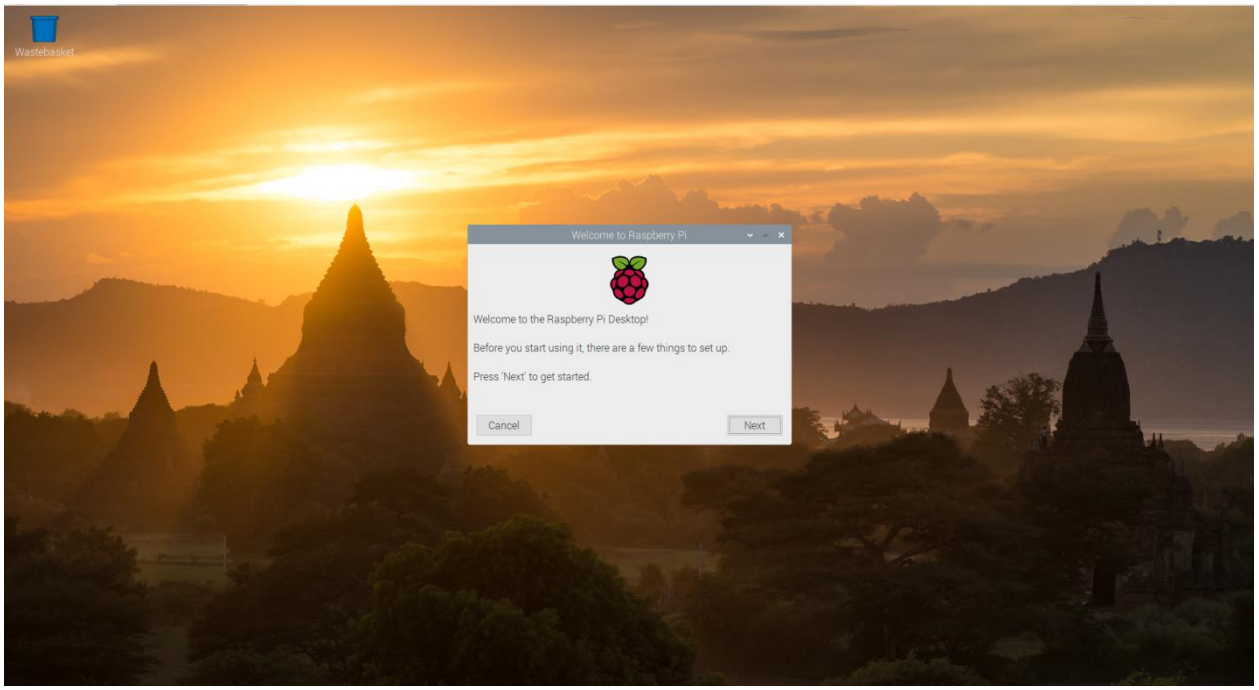


Figure 4 Raspberry Pi Home Screen

- Click on the next to select the country, language and the time zone. Then change the password from the default “raspberry” to the one of your own choice. The username is by default “pi”.
- Then the window for the wifi connection will appear. Select your desired wifi network available and click on next. Give the password for the wifi and your raspberry pi will be connected to the internet.

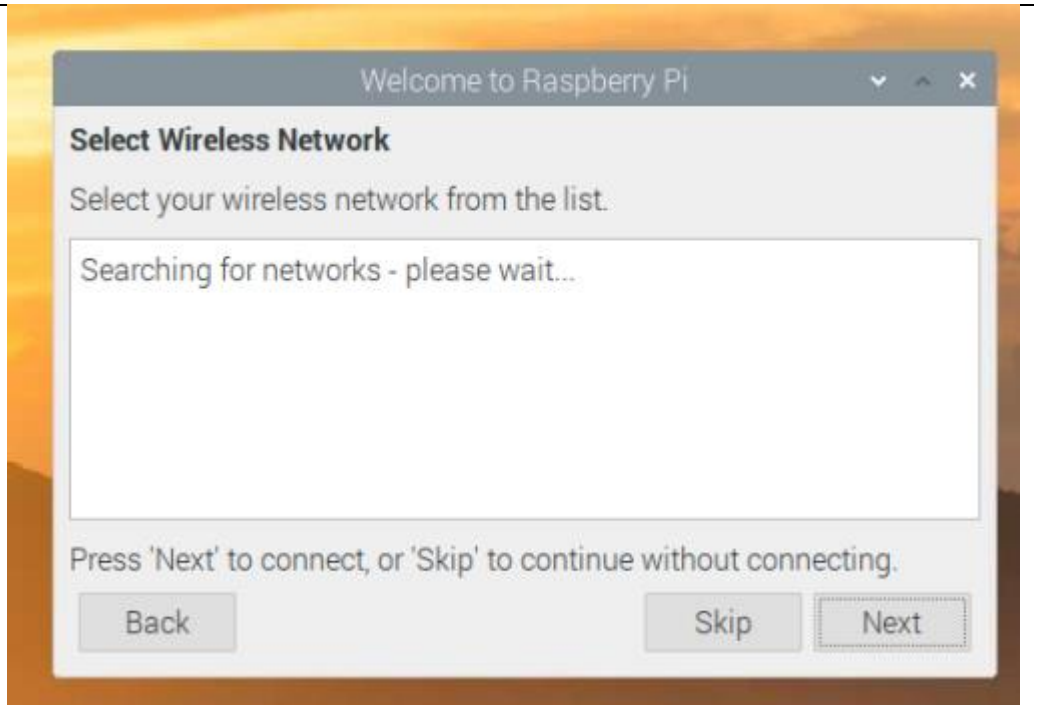


Figure 5 Wifi configure window on Raspberry Pi

- After this the initial configuration will be completed.
- You can also connect other wifi available by clicking on the “wifi option” present on the top right of the raspberry pi window”.

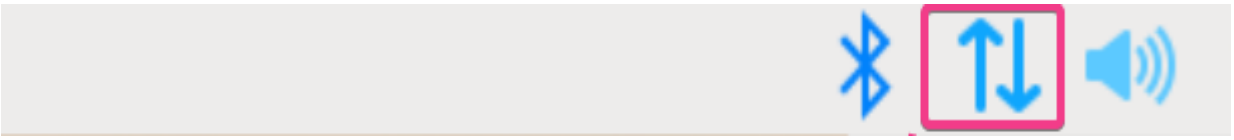


Figure 6 Wifi Icon on Raspberry Pi

- Now open the “terminal” a black icon on the top left of Pi window at fourth place and type following commands to update pi and install remote desktop

1. sudo apt update
2. sudo apt-get install raspberrypi-ui-mods xinit xserver-xorg
3. sudo reboot
4. sudo apt install xrdp

- The last step is to enable the “SSH” on raspberry pi by going to the “raspberry pi configuration” then “interfaces” and then to the “SSH”. Now the Pi is configured to be used.

3. Setting up Arduino IDE

What is Arduino IDE?

Arduino IDE is used to code Arduino Mega 2560 in this project. It is the most common and popular IDE used for the different microcontroller series of Arduino such as Arduino Uno, Arduino Mega 2560 etc. The main features of this IDE include simple and user friendly interface as well fast and efficient debugging and testing tools with the addition of the Serial monitor which makes it very easy for the professionals to use.

Downloading and installing Arduino IDE

- To download the latest version of IDE, go to the following link https://www.arduino.cc/en/software
- Arduino IDE 1.8.15 is the latest IDE available up till now.

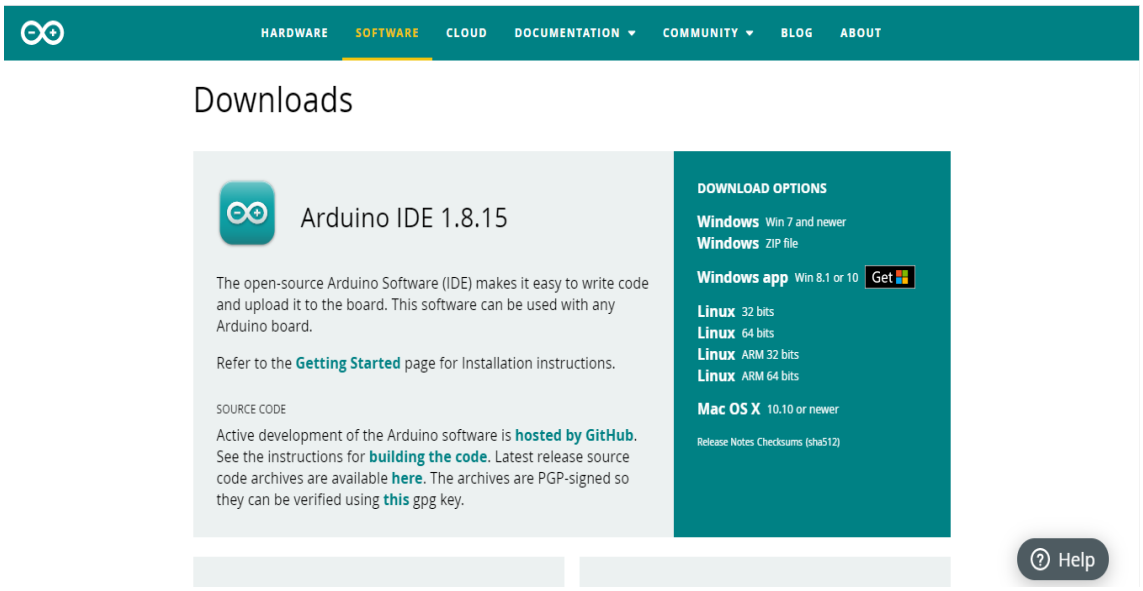


Figure 7 Arduino IDE to download

- Select the windows from the download option as shown in the figure 8 and this will download the Arduino IDE 1.8.15.

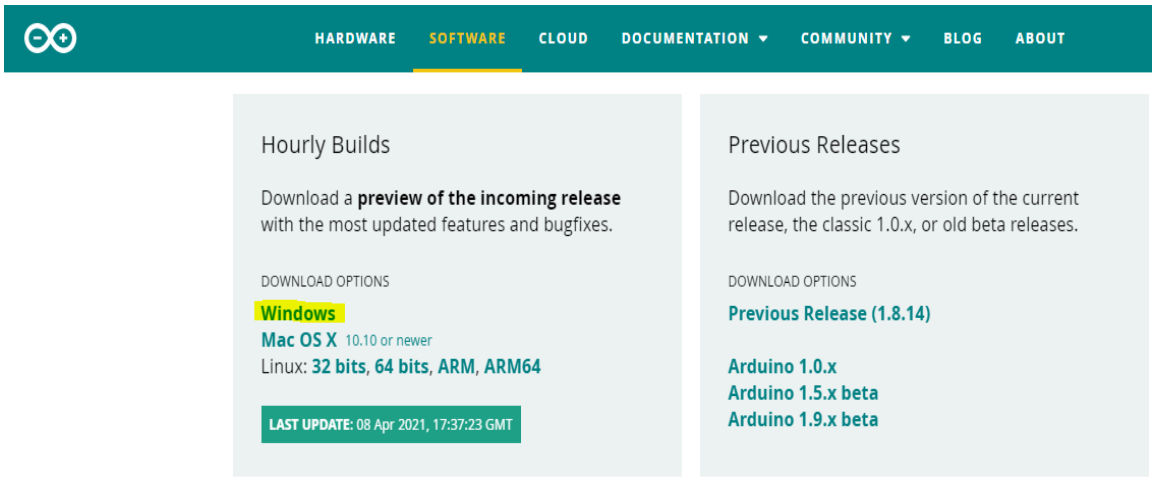


Figure 8 Arduino IDE download options

- Extract the downloaded folder **"arduino-nightly-windows"** to the place where you want to keep the Arduino IDE.
- After successful extraction, a folder named **"arduino-nightly"** will be created in the selected extraction location. This folder contains the Arduino IDE.
- Open the folder and it will be like the one shown below:

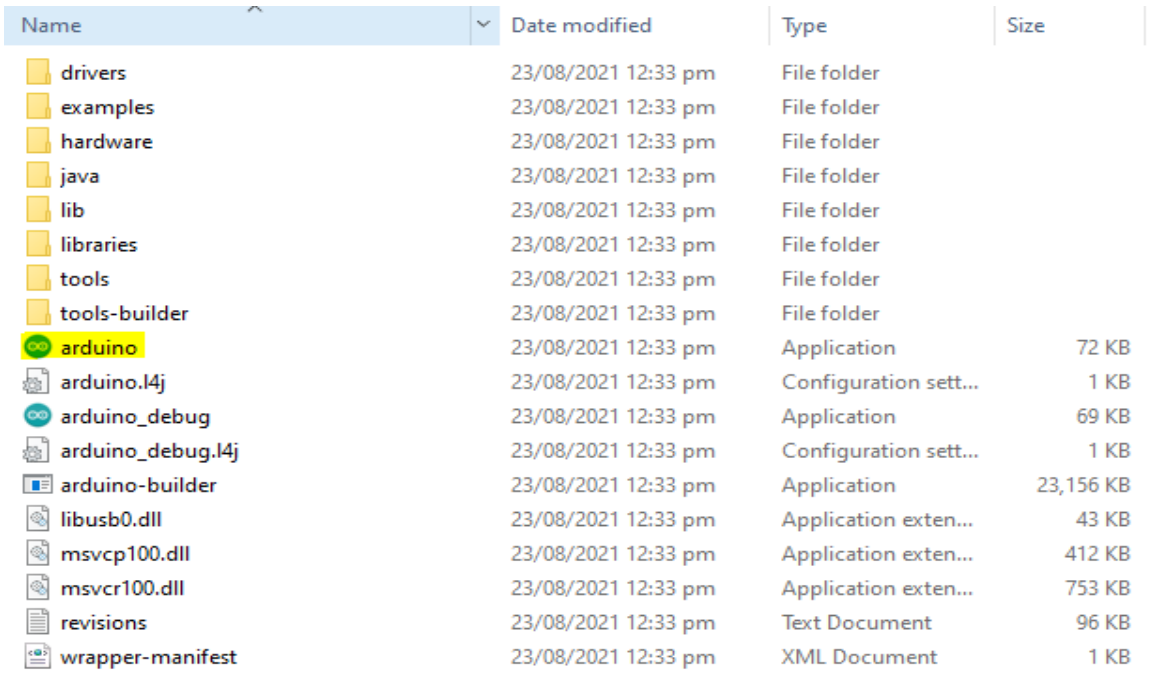


Figure 9 Arduino IDE extracted folder

- Select the highlighted “Arduino” application as shown in the figure 9.
- The Arduino IDE will open and it will look like the one in the figure below:

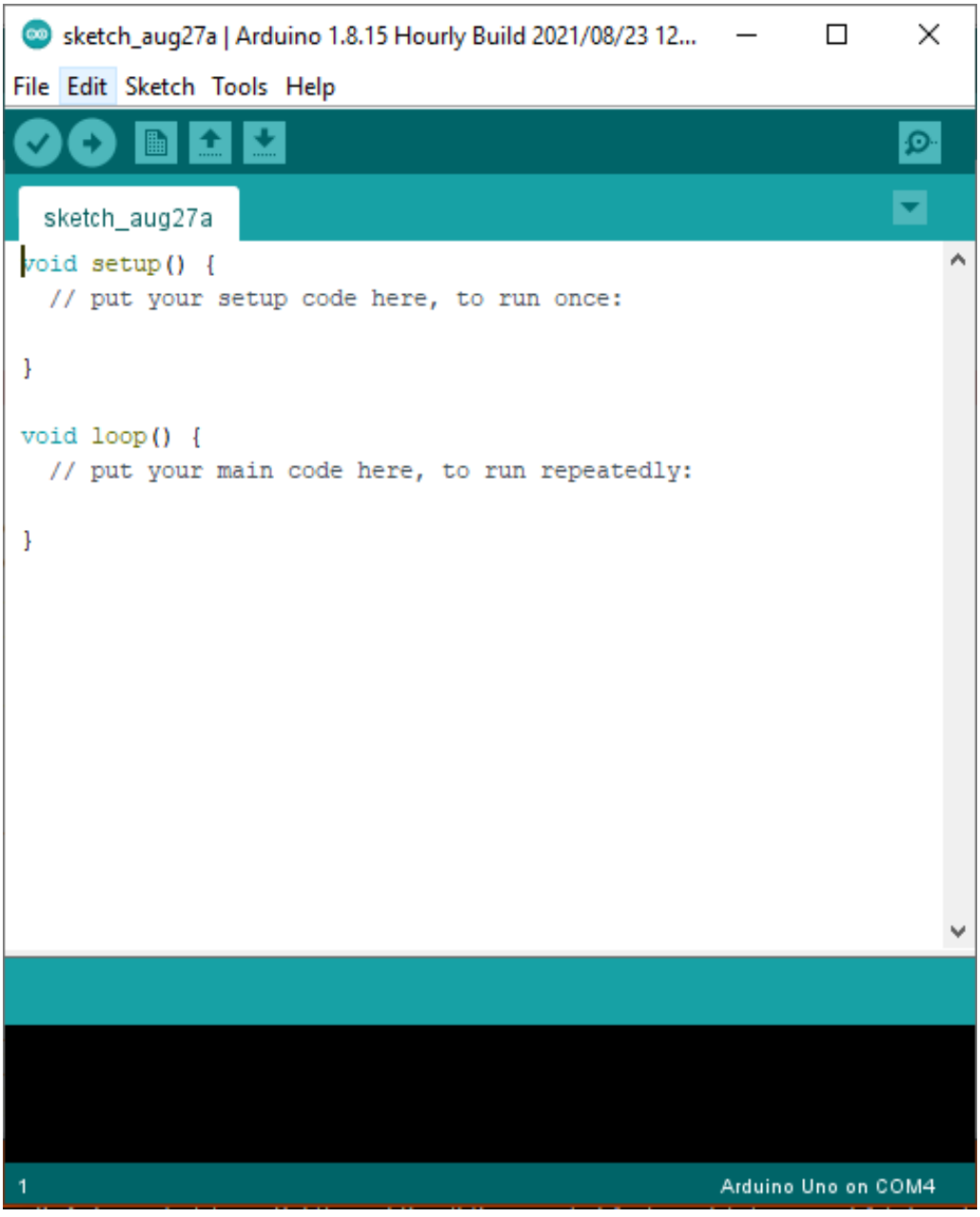


Figure 10 Arduino IDE

- The Arduino IDE is successfully installed and opened.

4. Setting up Arduino libraries

There are different libraries available in the Arduino IDE which contains many built-in functions related to different input/output devices. This heading deals with how to use those libraries in the Arduino sketch.

Libraries used in Home Car Wash Control System

The libraries used in the Arduino sketch of Home Car Wash Control System are shown below:

- **#include<Wire.h>**

This library is used to communicate with the I2C devices connected to the Arduino. In this project, if any I2C device is used then this library must be included.

- **#include <SPI.h>**

This library is used to communicate with the Serial Peripheral Interface (SPI) devices connected to the Arduino. In this project, if any SPI device is used then this library must be included.

- **#include<SoftwareSerial.h>**

This library is used to use the serial as virtual terminal in the Proteus simulation of this project.

- **#include <Adafruit_GFX.h>** and **#include <Adafruit_SSD1306.h>**

These two libraries are used to connect OLED with Arduino. OLED can be used with Arduino for debugging code since the serial monitor of Arduino is used as a mean of communication between Arduino and Raspberry pi.

- **#include <dht.h>**

This library is used to communicate with the DHT11 module connected to the Arduino. The DHT module is used to measure the temperature and humidity in this project.

Adding libraries to Arduino IDE

There are two ways to add libraries to the Arduino IDE:

1. **Manual downloading and adding**
2. **Automatically downloading from Arduino IDE**

The first method of manually downloading and adding is used in this project. Most of the above mentioned libraries are already included in the Arduino IDE and only three

are required to be downloaded from the internet. The required libraries are downloaded from different sources on the internet and then added to the Arduino IDE. The libraries are provided by the vendor in the folder named **"Arduino Libraries"** as shown below.

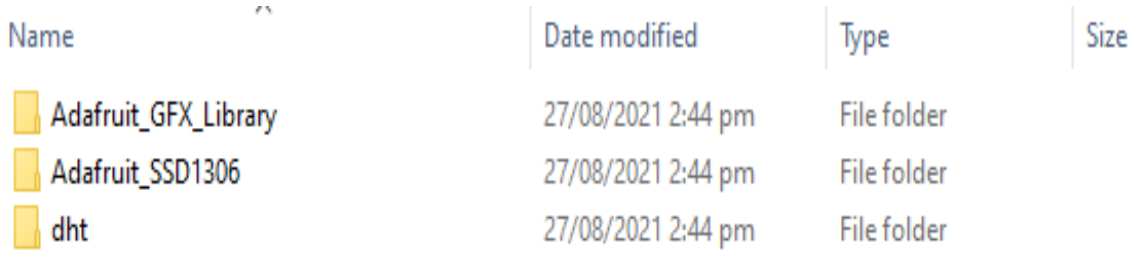


Figure 11 Arduino Libraries downloaded

These libraries should be placed in the "libraries" folder present in the Arduino IDE's extracted folder as shown in the figure below:

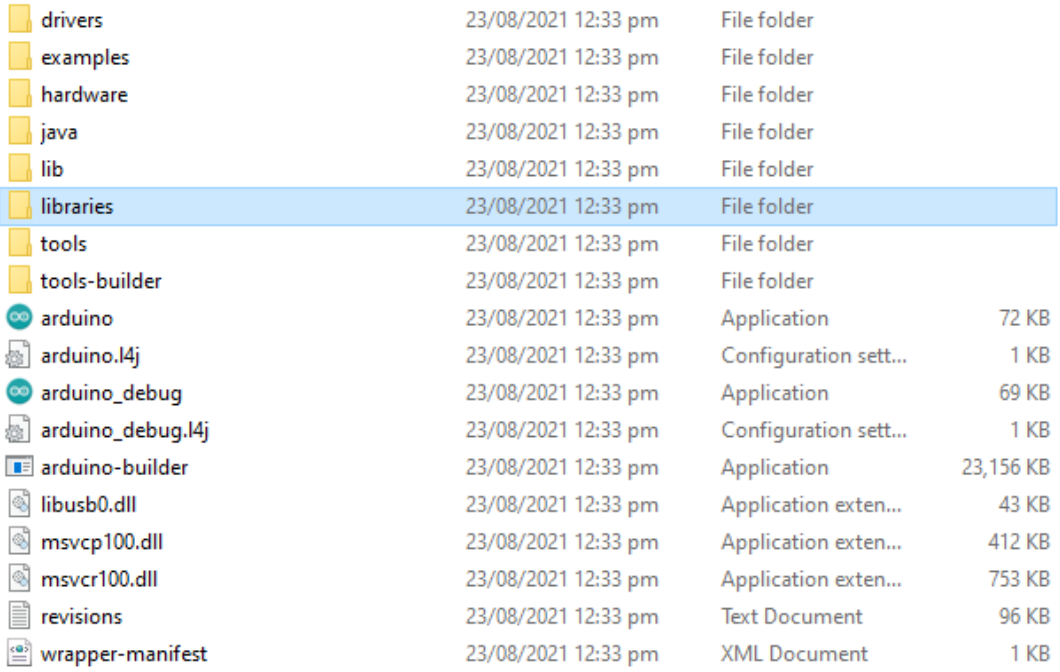


Figure 12 libraries folder for adding a library

Once the libraries are placed in the folder shown above then the Arduino IDE is ready to be used for compiling and uploading program to the Arduino Mega 2560.

5. Components used in project

The different components used in this project are briefly discussed below:

Arduino Mega 2560

Arduino Mega2560 is used in this project for controlling different input/ output devices. It is used as the main control unit. This microcontroller board is based on Atmega 2560.

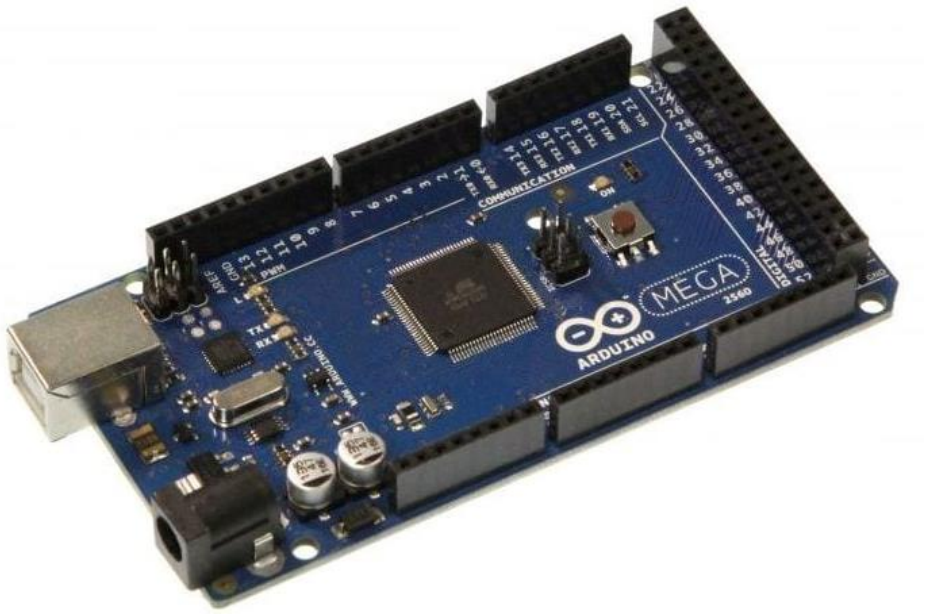


Figure 13 Arduino Mega 2560

The above diagram shows the Arduino Mega 2560. It consists 54 digital input or output pins. Due to large number of pins this microcontroller is commonly used in projects with many input/output devices. Its specifications are shown below in table below:

Table 1 Arduino Mega 2560 Specifications

Input Voltage	7-12V
DC Current per I/O Pin	20 mA
Operating Voltage	5V
Analog Input Pins	16
Input Voltage(limit)	6-20V

Digital I/O Pins	54
Flash Memory	256 KB
DC Current per 3.3V Pin	50 mA
Clock Speed	16 MHz
Weight	37 g
Width	53.3 mm
Length	101.52 mm
EEPROM	4 KB

Raspberry pi 4

Raspberry Pi 4 is used in this project to run GUI which is a computationally extensive task. It is also used to serially communicate with Arduino in bi-directional mode so that it can send the GUI parameters to Arduino and receive the progress and current operations from Arduino. . Its specifications are shown below in table below:

Table 2 Raspberry Pi 4 Specifications

Input Voltage	5v
Input current supply	3A
DC Voltage per I/O Pin	3.3v
Digital I/O Pins	54
Clock Speed	1.5 GHz
Weight	46 g
Dimensions	88 mm x 55mm x 19.5 mm
SD-Card support	32 GB

Two Channel Active Low 5v Relay Modules

The relay modules are used in this project to control the water pump, soap pump and the axis inverter. The two channel active low relays are used as shown in the figure below:

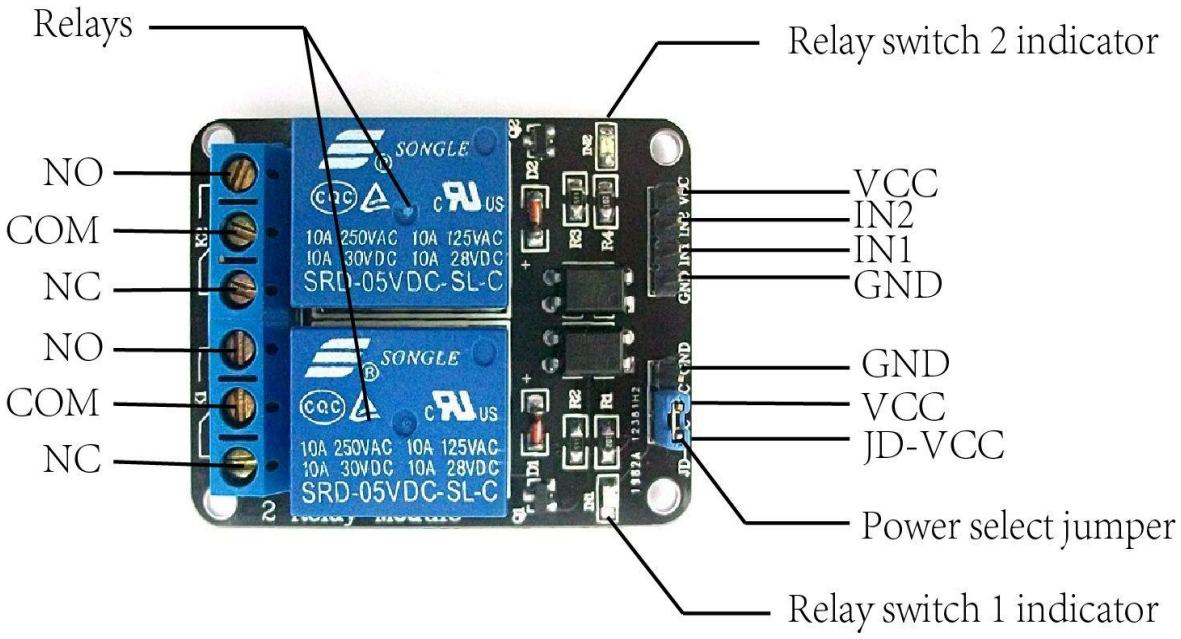


Figure 14 Two channel active low relay module

The term **active low** means that this relay is turned on when a low voltage signal is applied to its trigger pin and the term **two channel** means that this relay module consists of two switches which can be controlled by two trigger pins. The specifications of the relay module are shown in the table below:

Table 3 Relay module specifications

Trigger Voltage	5v
DC Switching Voltage	30v
AC Switching Voltage	250v AC
Maximum current	10 A
Trigger Level	Active Low

Temperature and Humidity Sensor DHT11

Temperature sensor used in this project is DHT11. It is used to detect the temperature and atmospheric moisture. It has long term stability and reliability.

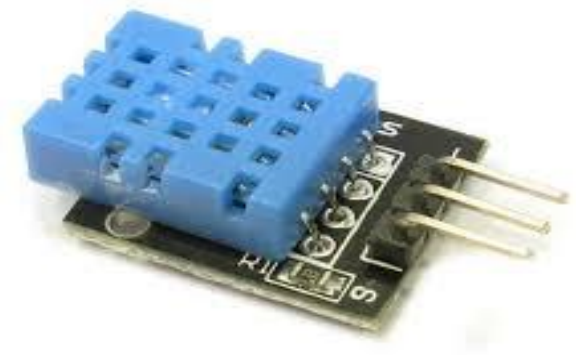


Figure 15 DHT11 sensor

This sensor consists of resistive element and a sense of wet negative temperature coefficient device. It has advantage like fast response steadiness and reliability. It has four pins and its specifications are shown in the table below:

Table 4 DHT11 specifications

Name	DHT-11
Input Voltage	5V
Humidity Range	20 - 90%
Temperature Range	0 – 50 C

Ultrasonic Sensor

The ultrasonic sensor is used in this project to find the water and the soap level. The model used for the ultrasonic sensor is HC-SRO4 ultrasonic module as shown below:

Figure 16 Ultrasonic sensor

The specifications of the ultrasonic sensor are shown in the table below:

Table 5 Ultrasonic sensor specifications

Power supply	5v dc
Current	15ma
Distance Range	2cm – 400cm
Dimensions	45 mm x 20mm x 15 mm

6. Variables used in project

There are different variables declared and used to complete the scope of this project. These variables are discussed in the tables shown below:

Input / Output devices variables

The variable used for the input and output devices with their names and respective I/O devices are shown in the below table:

Table 6 I/O devices variables

Variable name	Pin on Arduino	I/O Device name and function
relay_water_pump	4	Water Pump on/off switch
relay_soap_pump	5	Soap Pump on/off switch
relay_forward_inverter	7	Inverter forward on/off switch
relay_reverse_inverter	6	Inverter reverse on/off switch

home_s	12	Home Sensor input
forward_limit_s	13	Forward Sensor input
reverse_limit_s	11	Reverse Sensor input
DHT_Pin	10	Temperature Sensor
trigger	9	Water Level Sensor
echo	8	Water Level Sensor
trigger2	3	Soap Level Sensor
echo2	2	Soap Level Sensor

GUI Start Process variables sent to Arduino

The essential variables initialized by the user on the GUI which are sent to the Arduino for the wash process are shown in the below table:

Table 7 GUI start process variables

Variable name	Functionality
total_cycle	It decides the number of times whole wash process will run.
water_cycle	It decides the number of times water wash process will run in each cycle.
soap_cycle	It decides the number of times soap wash process will run in each cycle.
w_s_delay	It decided the delay in seconds between the water and soap process.
s_w_delay	It decided the delay in seconds between the soap and next water process.

GUI Variables sent to Raspberry Pi by Arduino

These variables contain the signals which are sent from Arduino to the Raspberry pi and then the relative data is displayed on the GUI.

Table 8 GUI variables sent to pi

Variable name	Functionality
water_level_s	This indicates the level of water in the water tank
soap_level_s	This indicates the level of soap in the soap tank

home_sensor_s	This indicates the home position of inverter
forward_sensor_s	This indicates the forward motion of inverter
reverse_sensor_s	This indicates the reverse motion of inverter
water_pump_s	This indicates the on/off status of water pump
soap_pump_s	This indicates the on/off status of soap pump
motor_s	This indicates the on/off status of inverter.
temperature	This indicates the temperature
humidity	This indicates the humidity
progress	This indicates the current operation of the process
percentage	This indicated the completed percentage of the project
finish_s	This indicates the process finish signal

Variables for setting optimum temperature, water and soap level

There are three variables which can be used to set the optimum temperature as well as optimum water and soap level as shown in the table below.

Table 9 variables for setting optimum conditions

Variable name	**Functionality**
optimal_temperature1	This sets the lower range of optimum temperature for wash process.
optimal_temperature2	This sets the upper range of optimum temperature for wash process.
optimal_water_level	This sets the optimum water level for wash process.
optimal_soap_level	This sets the optimum soap level for wash process.
time_for_cycle	This indicated the total time axis inverter takes to reach from reverse to forward position in seconds

7. GUI used in project

Graphical User Interface is an important phase of this project. A customized and responsive GUI has been developed for the user to control the functionality of the system. By using this GUI user can easily access and visualize the whole car wash system. GUI is developed using the most popular and industrial level tool known as PyQt5 along with Qt designer.

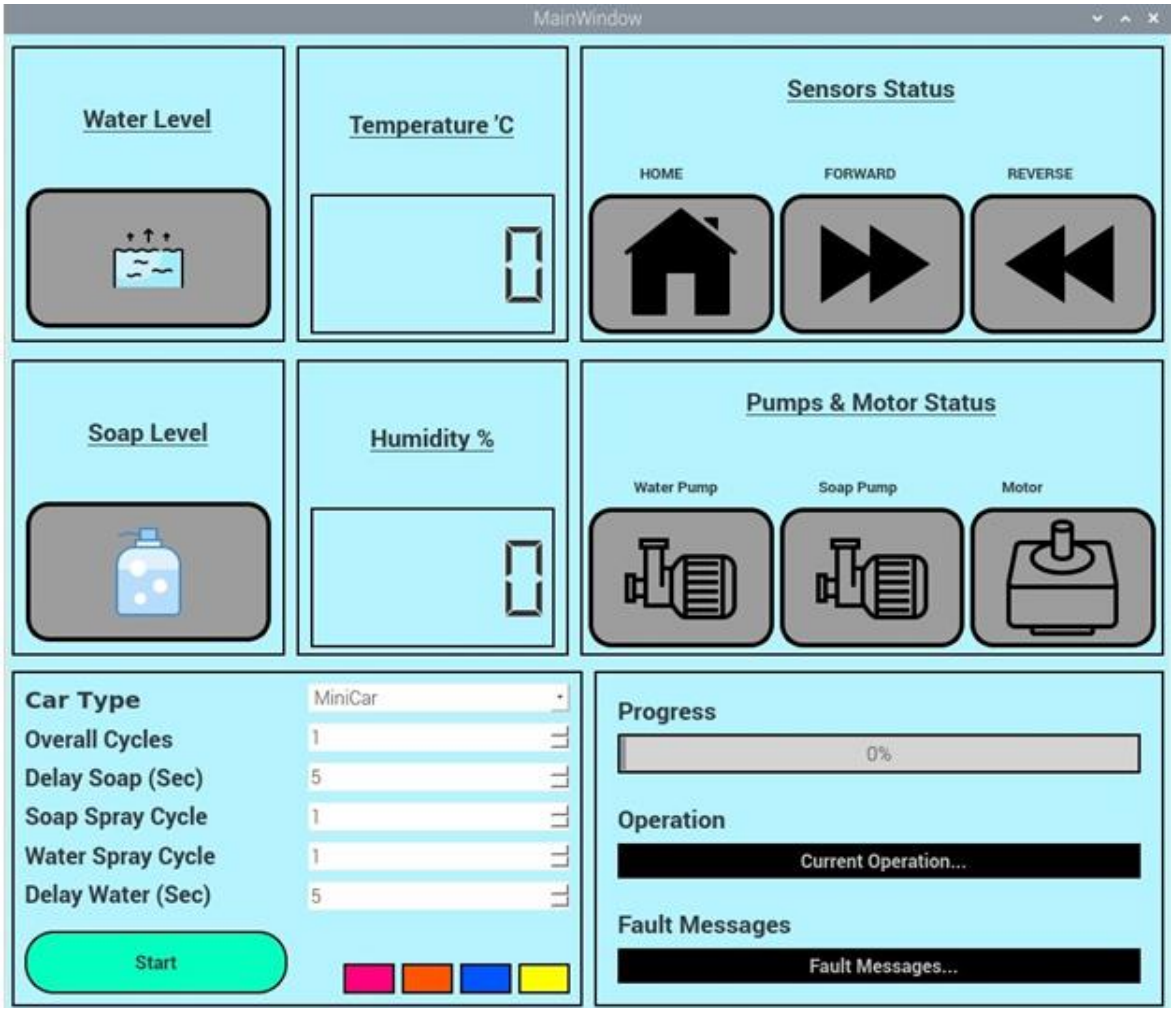


Figure 17 GUI used in project

GUI Features

GUI perform bi-directional communication which means it send data to Arduino and take signals from the Arduino and display it on GUI. GUI has several sections that take inputs which includes Car Type. We have allotted three options for car type i.e. MiniCar, Sedan and MPV. On acknowledgement from the Arduino it will display the sensor, pump and

motor status along with soap and water level status. Furthermore, the current status of the process, progress and the finish message is also displayed on the GUI.

Sections of GUI

GUI is particularly divided into three frames top, middle and bottom. Each frame has its own subdivisions of frame known as left, right and center. Below we will discuss each frame.

Top Frame

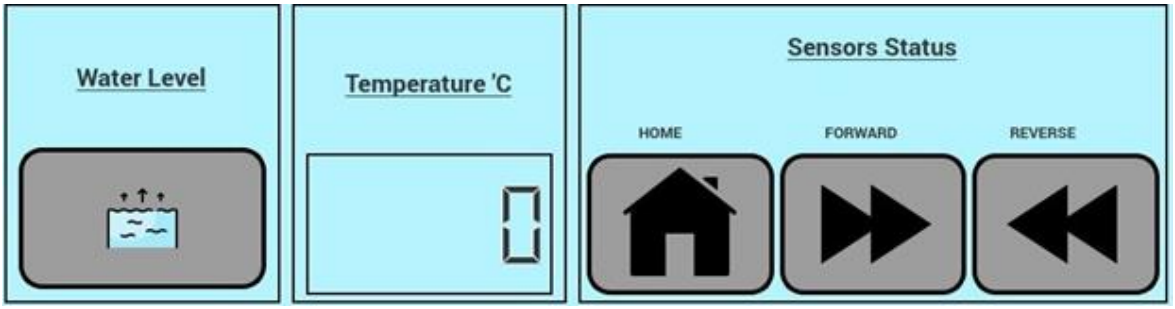


Figure 18 Top Frame of GUI

In top frame we have further divisions of frames into top Right, top Left and top Center which contains sensor status, water level and temperature respectively.

Middle Frame

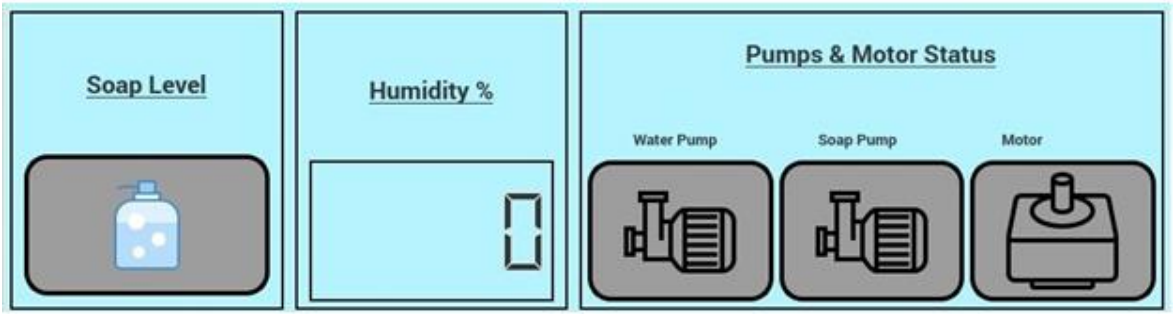


Figure 19 Middle frame of GUI

In the middle frame we have further divisions of frames into middle Right, middle Left and middle Center which contains pump and motor sensor status, soap level and humidity respectively.

Bottom Frame

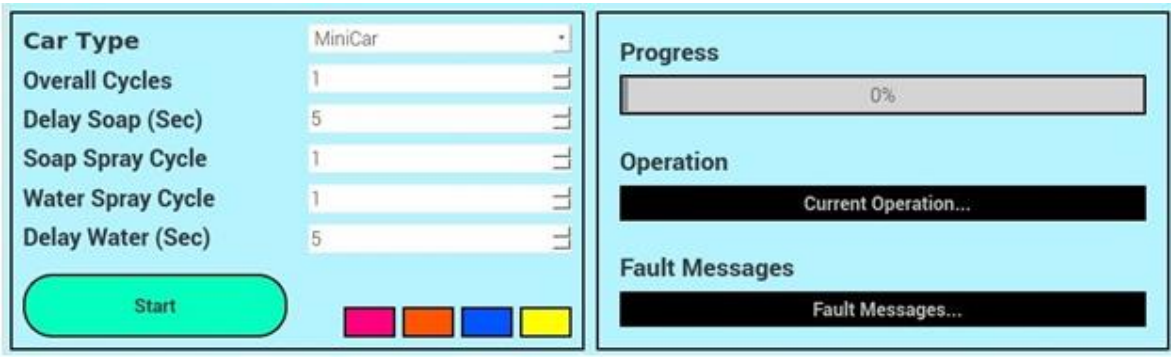


Figure 20 Bottom Frame of GUI

In bottom frame we have further divisions of frames into bottom Right and bottom Left. Frame bottom Right contains progress bar, current operation status and fault messages while Frame bottom Left contains parameters and combo Boxes for each parameter respectively.

8. How to Run Project

The folder named "Smart Car Wash data" is provided by the vendor which contains all codes, Proteus simulations, installation files and libraries. To run the Arduino sketch perform the following steps.

- Open the ino file named as "car_wash_code" present in the following directory **\Smart Car Wash data\Codes\Arduino code\car_wash_code**
- Run it and flash it on the Arduino MEGA. Make sure that the components are connected with Arduino according to the schematics shown in the below headings.
- Place the folder named as **"gui"** in the raspberry pi and open the **"final_gui.py"** file with **"Python 3"** or **"Thonny IDE".** This will load the GUI on the screen.
- The directory for the "gui" file is below:
  Smart Car Wash data\Codes\Raspberry pi code\gui

9. Project Block Diagram

The overall block diagram of the project is shown below:

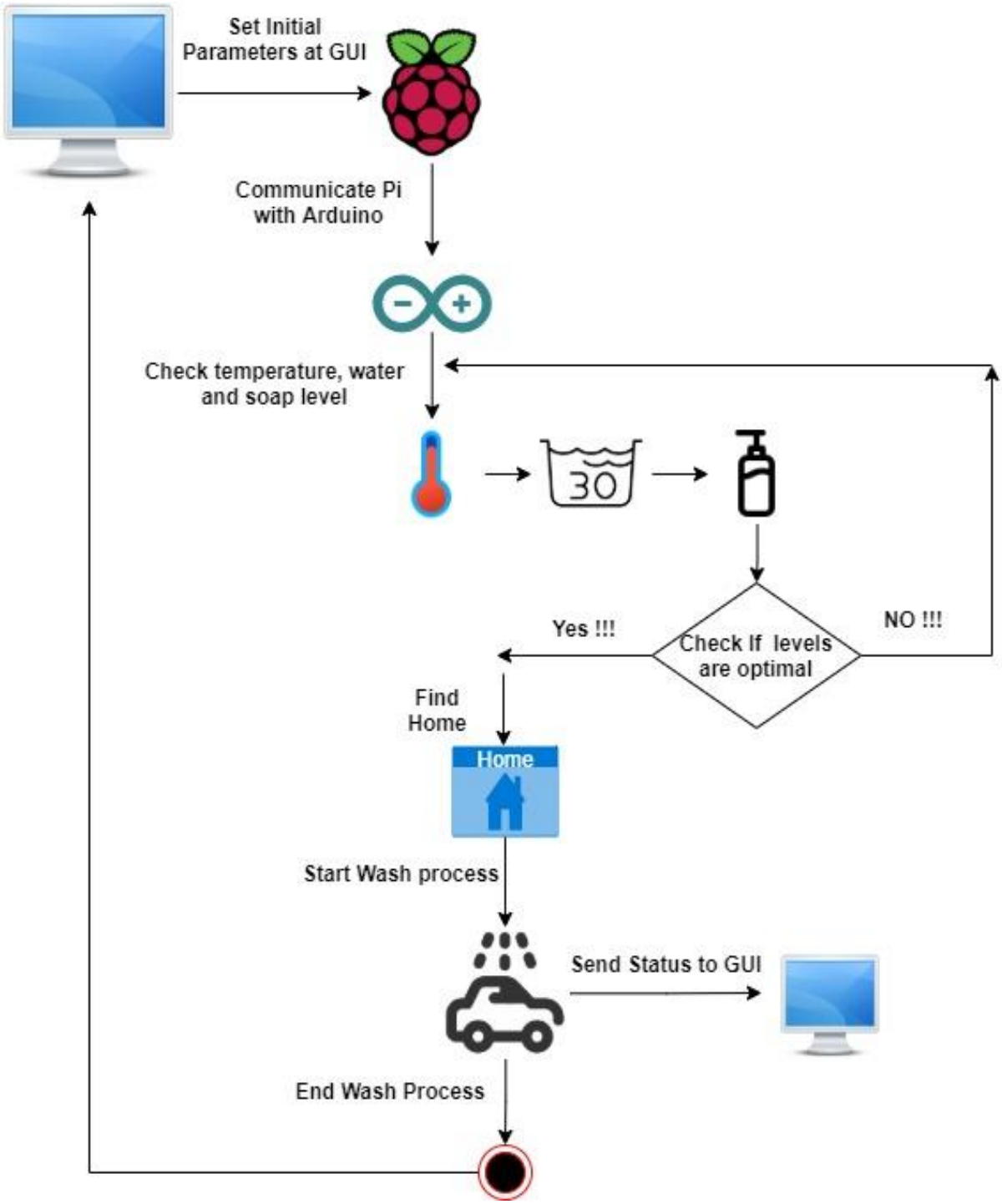


Figure 21 Project Block Diagram

10. Project Execution Sequence

This project is executed in the following sequence shown below:

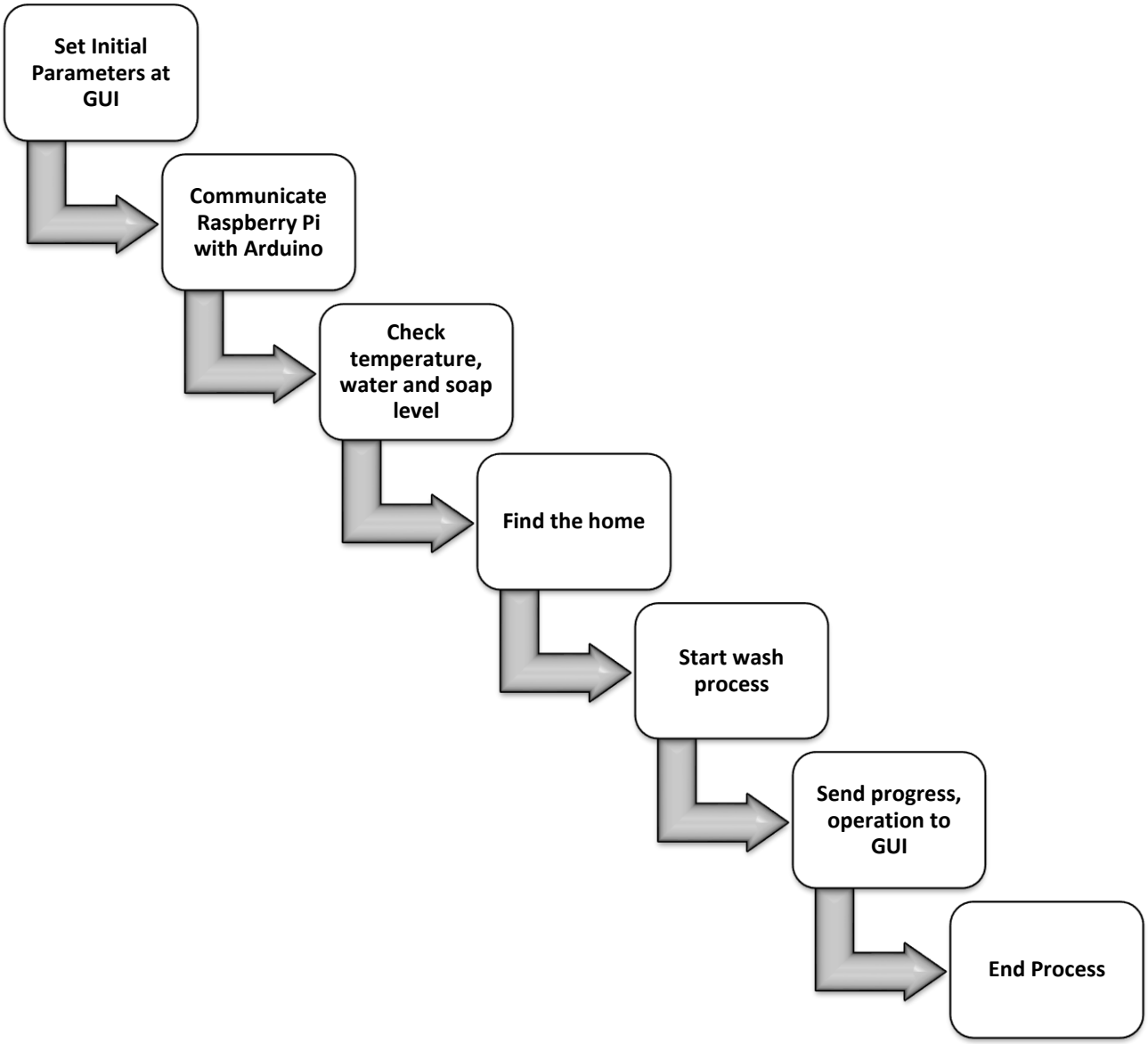


Figure 22 Project execution sequence

11. Project Schematic

The connections of all the components with Arduino are shown in the figure below:

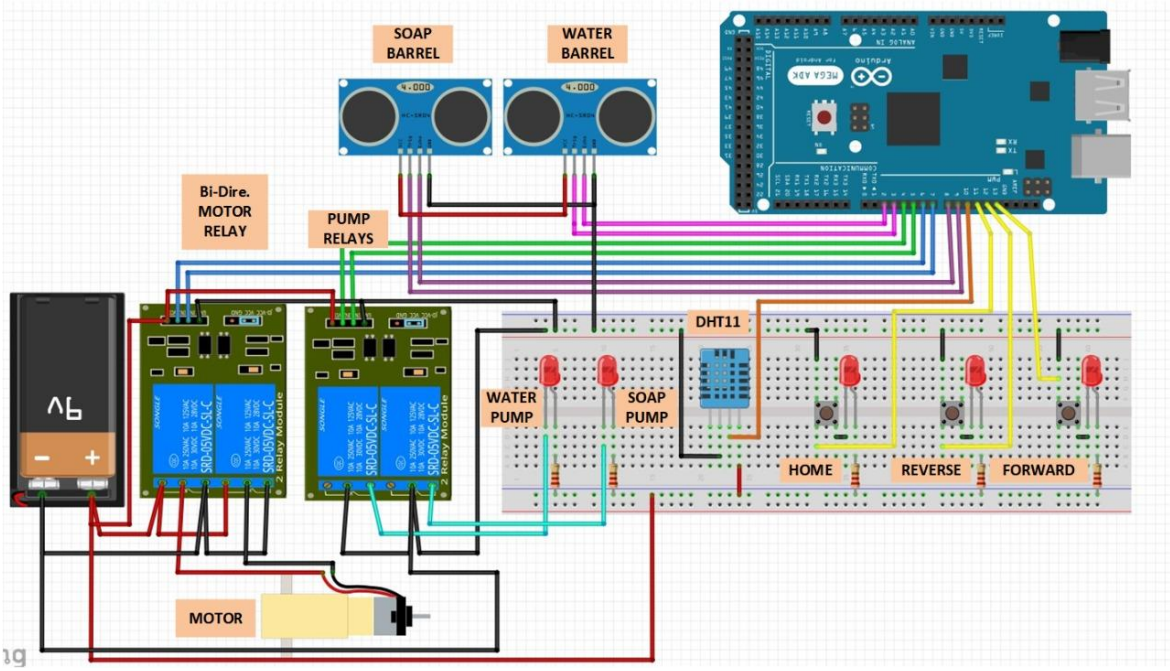


Figure 23 Connections of components with Arduino

Similarly the connection of Arduino with Raspberry Pi is shown below:

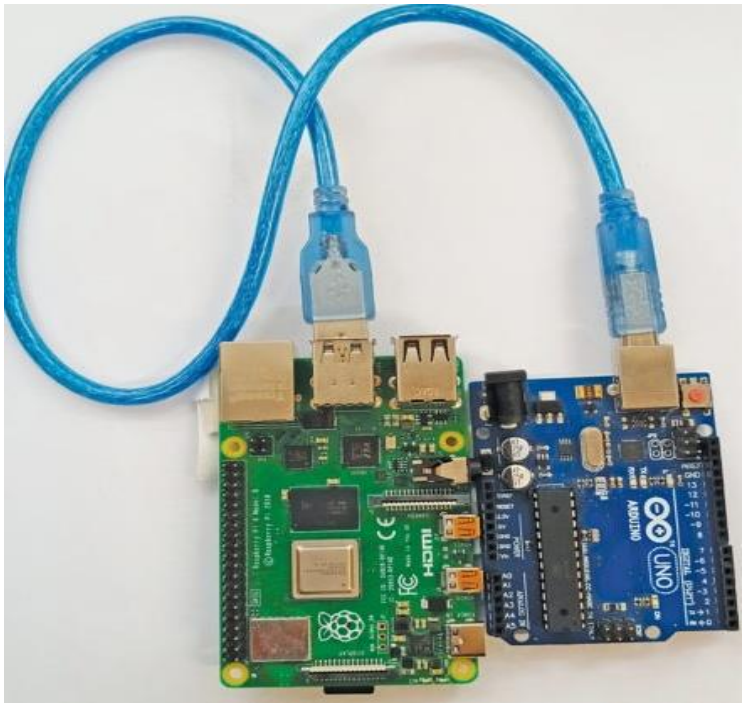


Figure 24 Arduino and Raspberry Pi connection

12. Results on Proteus Simulation

Majority of the parts of this project can be designed, tested and verified in the **"Proteus"** software with only GUI and pi- mega communication not included in it. There are some steps which need to be covered before running the Proteus simulation.

Proteus version required

The schematic for this project is build on "Proteus 8.11 SPO" so for this schematic to run, one must have at least **Proteus 8.11 SPO.** The setup of this Proteus version will be provided by the vendor. All need to be done is to install that setup.

Required Libraries

There are a few Proteus components libraries which must be included in Proteus before running the schematic as shown below:

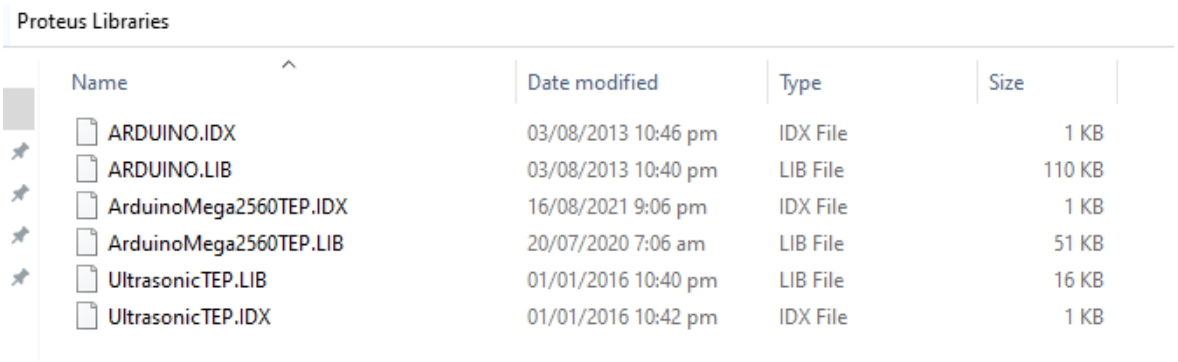


Figure 25 Proteus Libraries to be installed

These libraries are provided by the vendor in a folder named "Proteus Libraries". All need to be done is to copy those libraries to the Proteus directory i.e.
"C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\LIBRARY" as shown below:

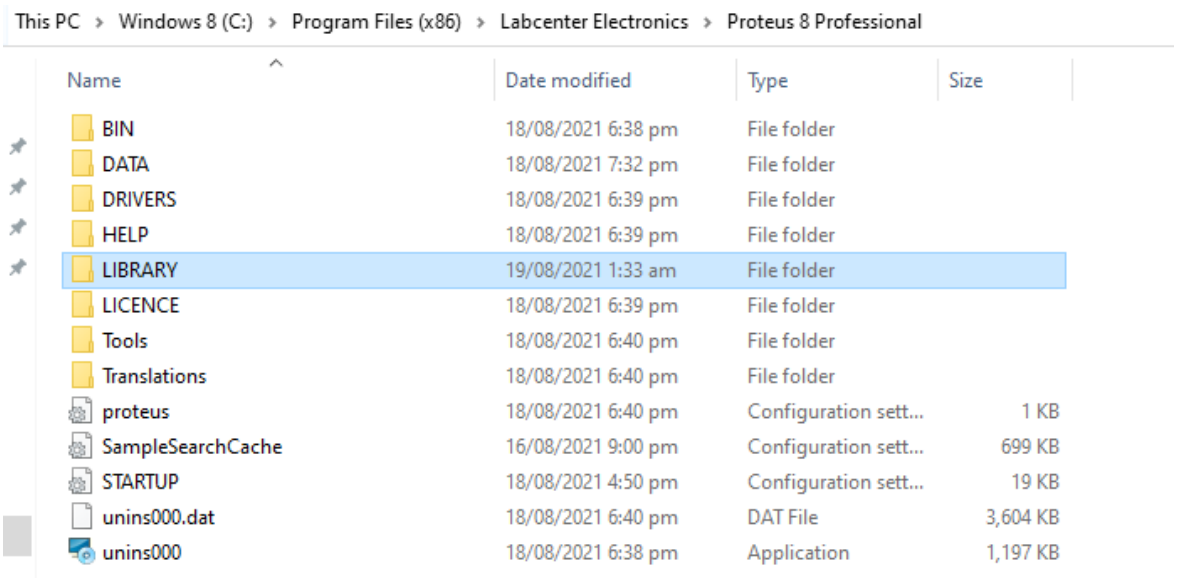


Figure 26 Proteus library location

Proteus Simulation

The Proteus simulation used in this project is shown below:

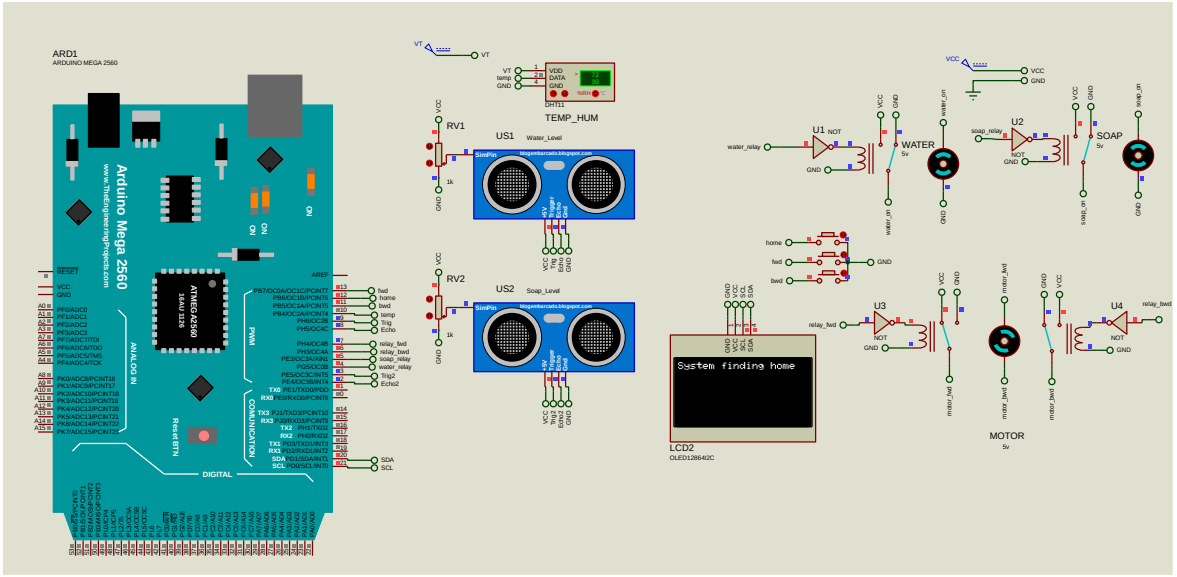


Figure 27 Proteus simulation

For this simulation to run, a hex file is to be placed in the Arduino Mega 2560 as well as the two Ultrasonic sensors named as Water Level and the Soap Level by double clicking on them and then placing the hex file in the locations as shown below:

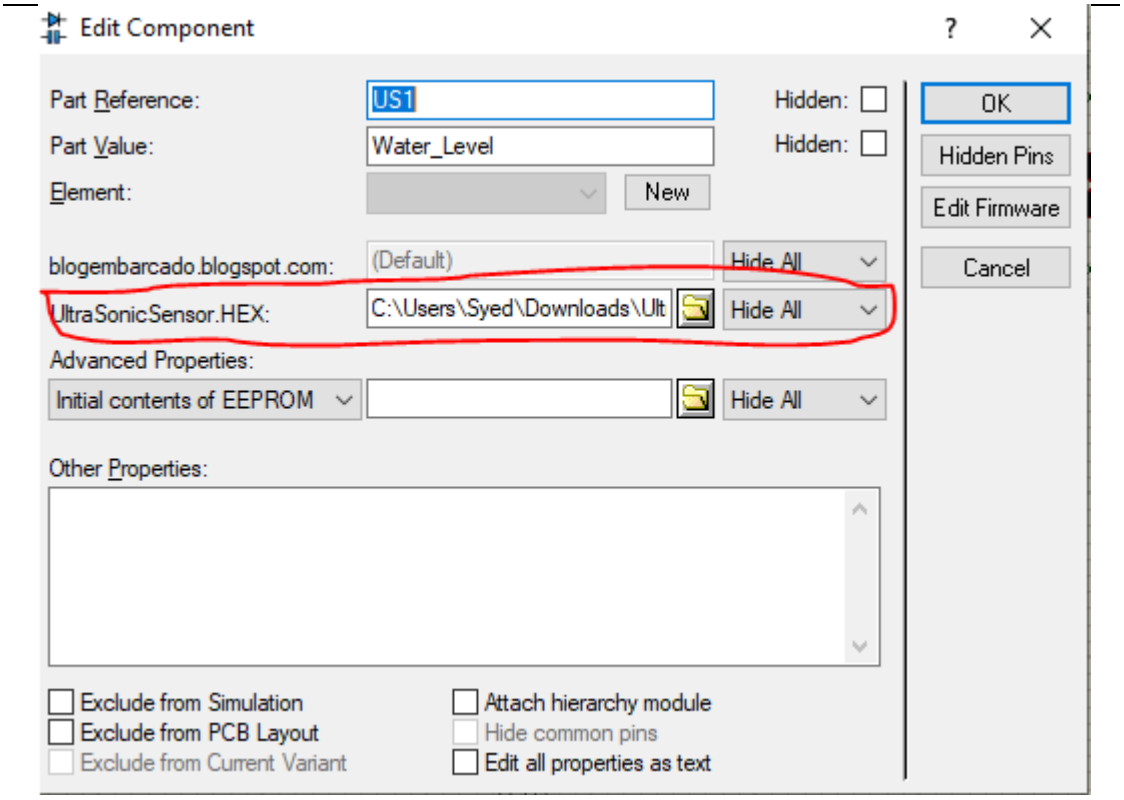


Figure 28 Ultra sonic hex file location

Similarly the hex file location for Arduino Mega 2560 is as below:

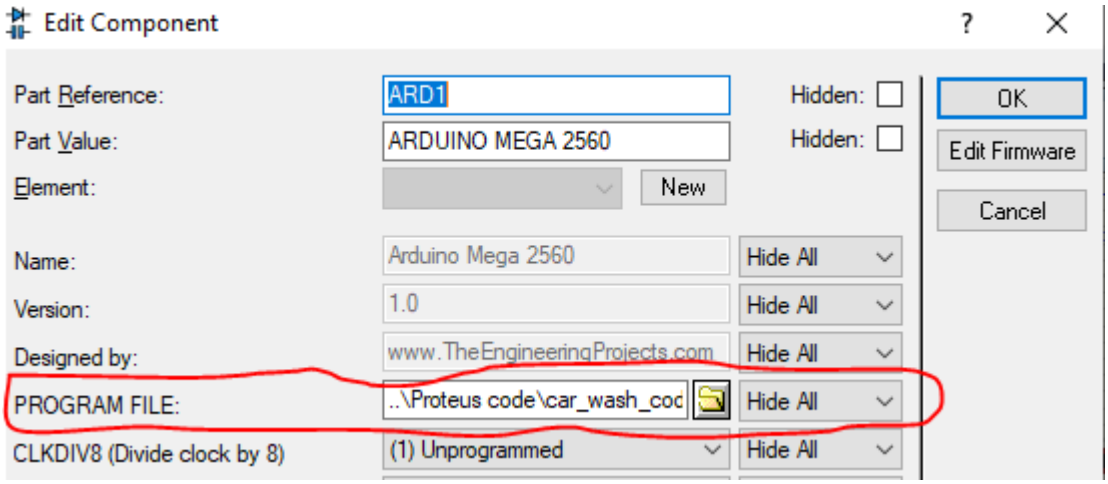


Figure 29 Arduino Mega 2560 hex file location

The hex files are provided by the vendor in the folder named **"HEX Files"**.

13. Accessing Raspberry Pi For Remote GUI

Raspberry Pi is setup remotely in order to run Graphical User Interface (GUI) on remote desktop. For this purpose following configuration needs to be done.

Remote Desktop Settings

First step is to enable the remote desktop connection parameter in the network settings of laptop or where you want to access remote GUI. For this click on windows button at bottom left corner of your screen. Type “Remote Desktop Settings” and select the respective option. Another window will prompt up with remote desktop settings.

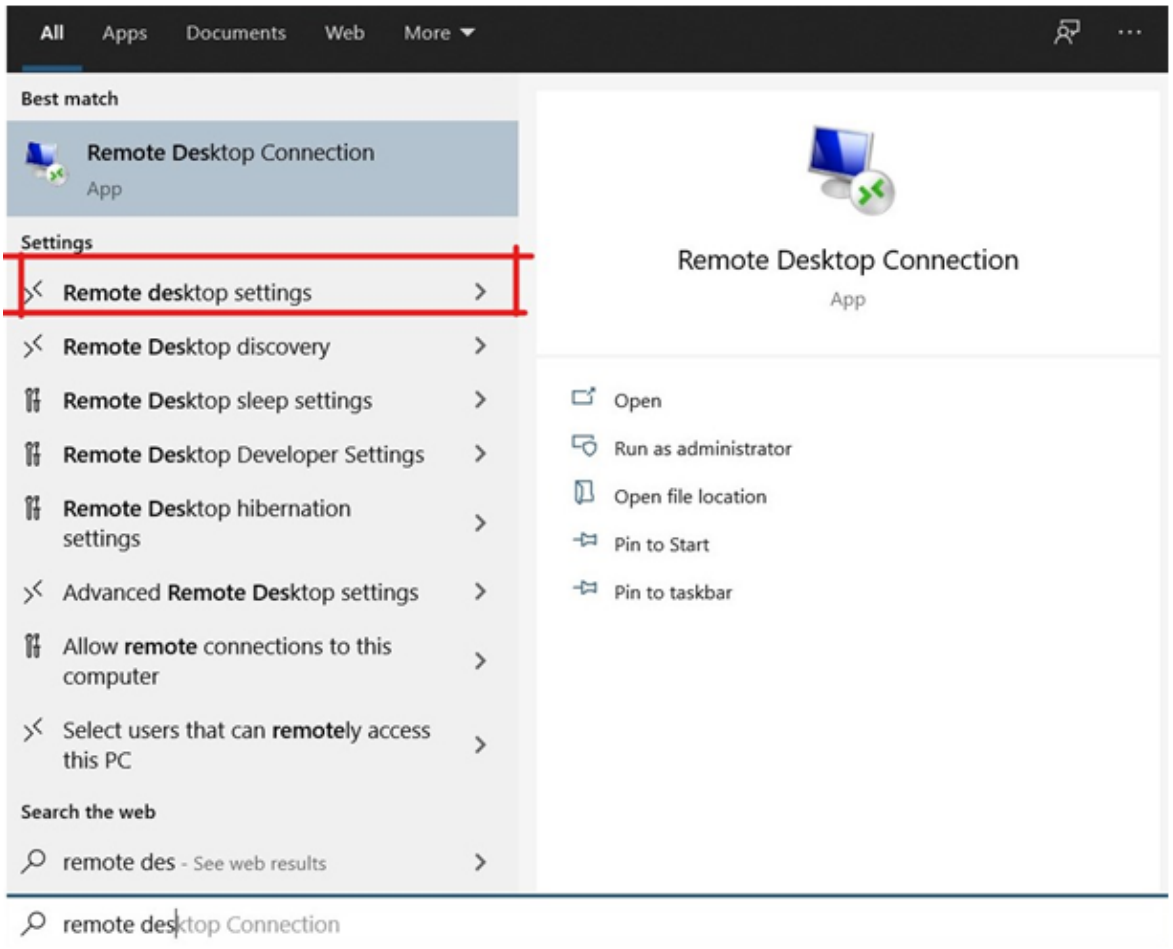


Figure 30 Remote Desktop Settings

Enable Remote Desktop Settings

Settings window will be open that contains an option of “Enable Remote Desktop”. Currently it is in “OFF” state. Turn it “ON” to enable the remote desktop functionality. This functionality enables you connect to and control this PC from a remote device by using a Remote Desktop client. You will be able to work from another device as if you were working directly on this PC.

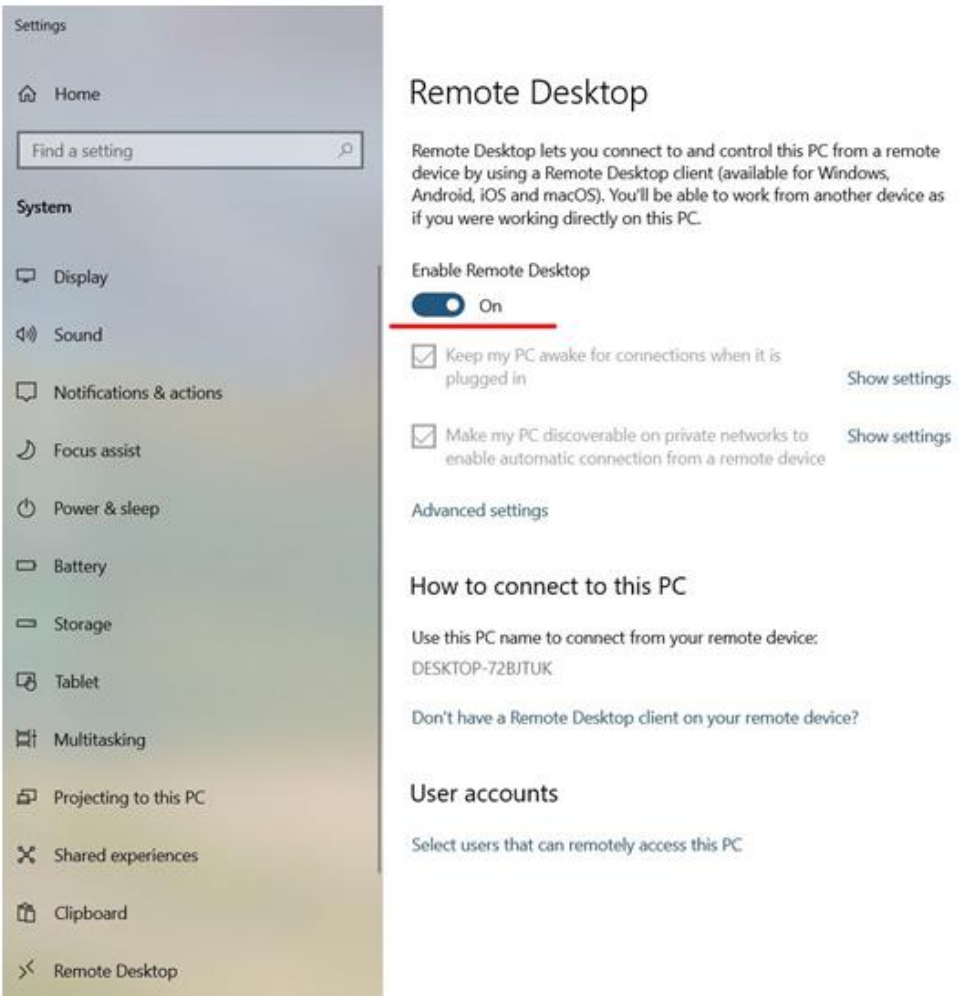


Figure 31 Enable Remote Desktop

Configure Wifi Network

For remote connection with raspberry pi you have to make sure you have a stable internet connection through which your remote pc and raspberry pi both will be connected. In my case I enable hotspot through my cell phone that has a username as “Umer#House”.

Connect your pi with 5v adapter or connect it via remote desktop/pc via Type C cable. It will power up your raspberry pi. Now find the IP address of the raspberry pi connected to the selected network using “IP scanner” or any other tool. In my case, on my cell phone 2 devices are connected via hotspot one is remote desktop and second is raspberry pi as shown below:

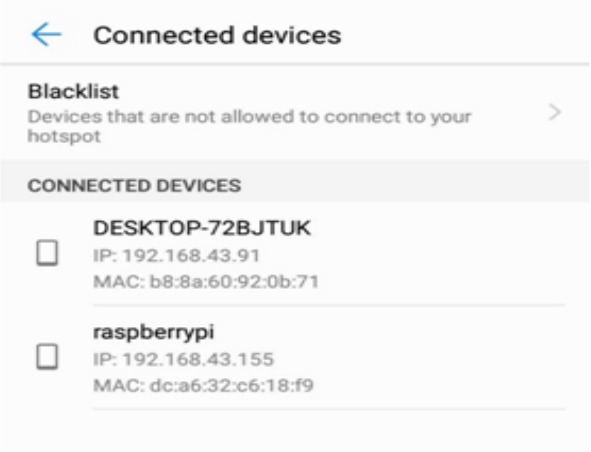


Figure 32 IP address Info

Open Remote Desktop Connection

Now after finding the IP address of Pi, click on the windows button in bottom left corner of the screen and type “Remote Desktop Connections”.

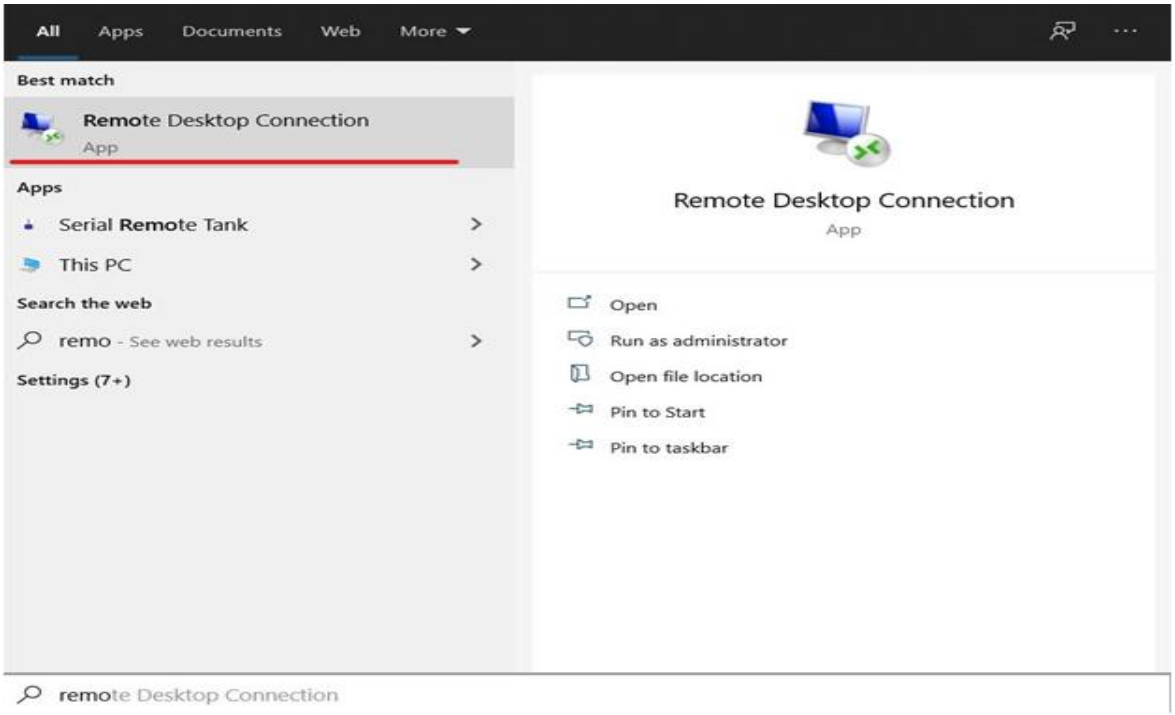


Figure 33 : Remote Desktop Option

Configure IP Address for Pi

After clicking on the Remote Desktop Connection option, another window will pop up. Enter the IP address of Raspberry Pi which was allocated to Pi.

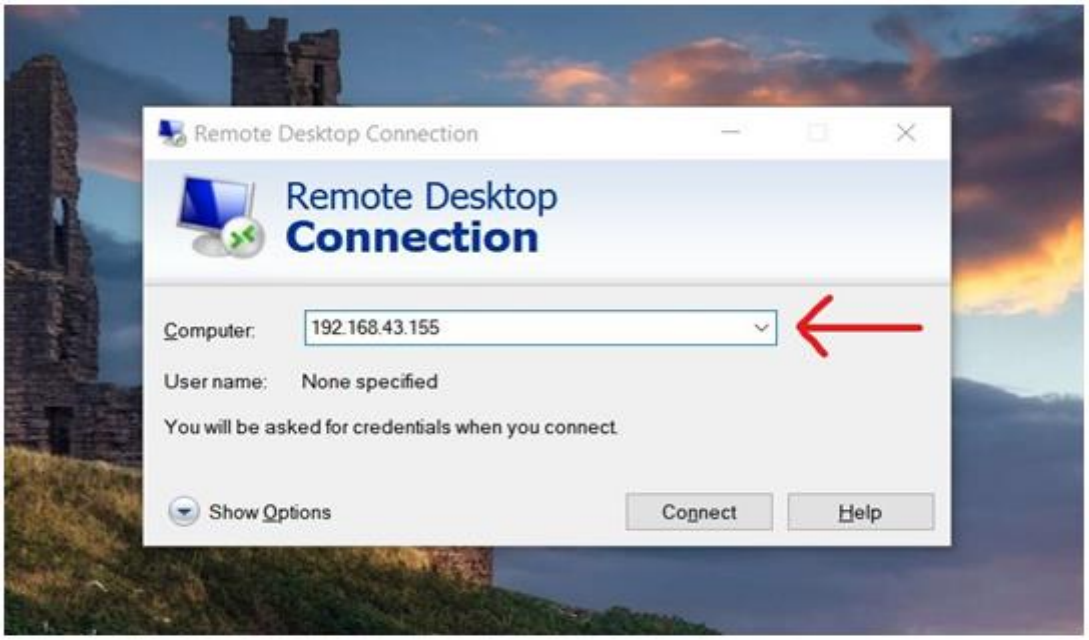


Figure 34 Remote Desktop connection window

Press connect, it will pop up a warning window, press ok to continue.

Figure 35 Warning message

Login to Raspberry Pi

Raspberry Pi login window will pop up running on the IP address provided by the cell phone hotspot. Enter the username which is by default “pi” and the password set for pi .

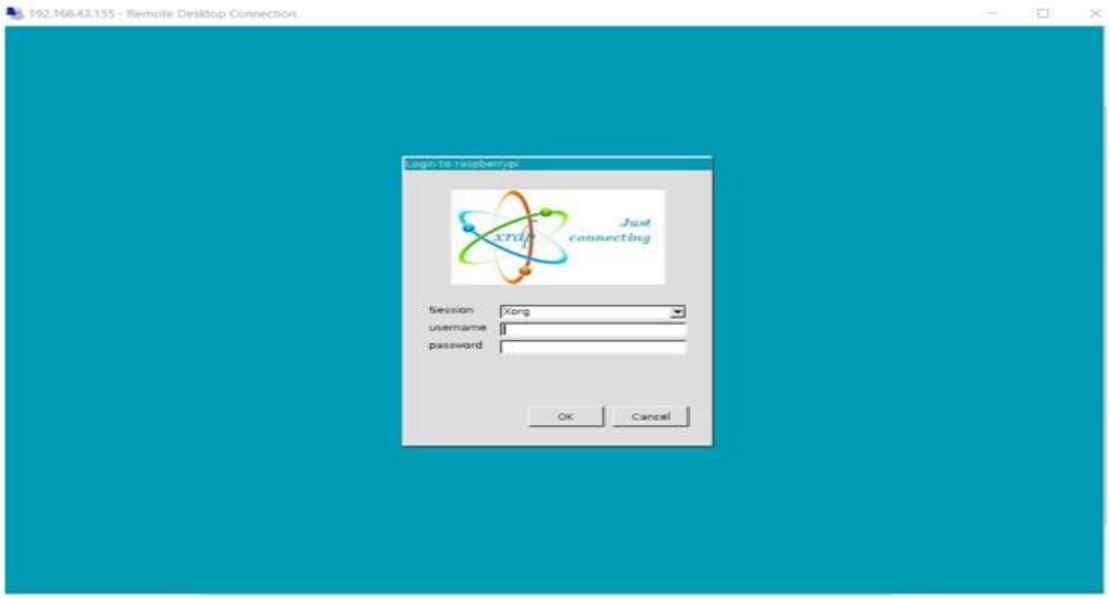


Figure 36 Raspberry Pi log in page

Raspberry Pi Activate on Remote Desktop

After providing the credentials now you have your raspberry pi active on remote desktop.

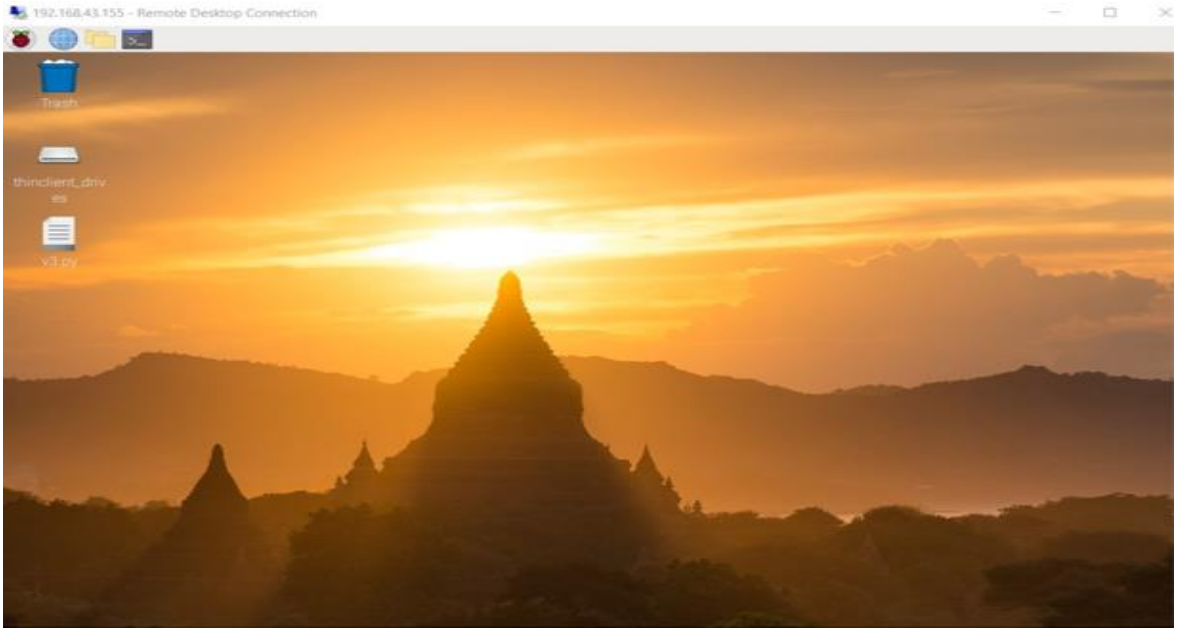
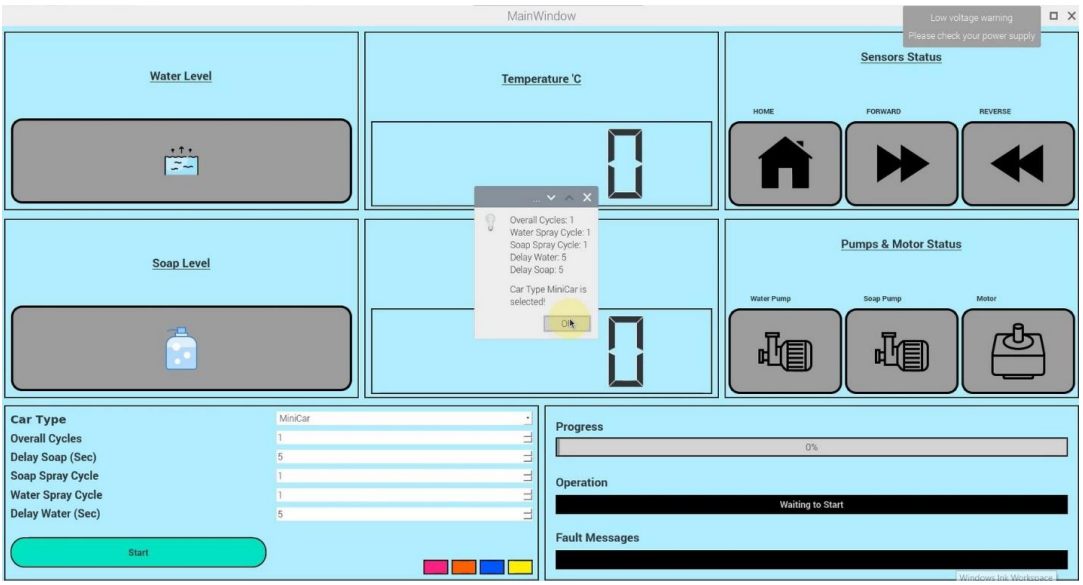


Figure 37 Raspberry Pi Desktop on Remote PC

14. Results of GUI and Hardware

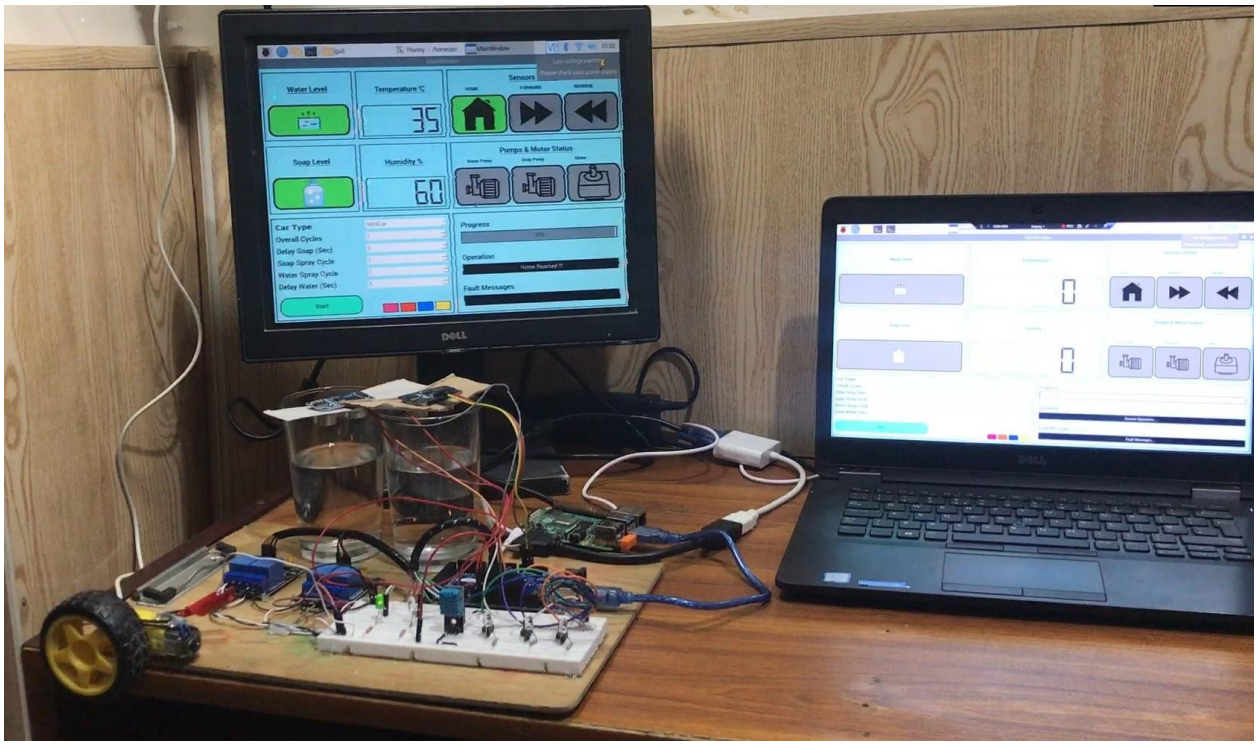
System with Hardware Prototype as follows

Set parameters for the car and then press ok to start the system. It will pop up the parameters you have entered as shown in the figure below.



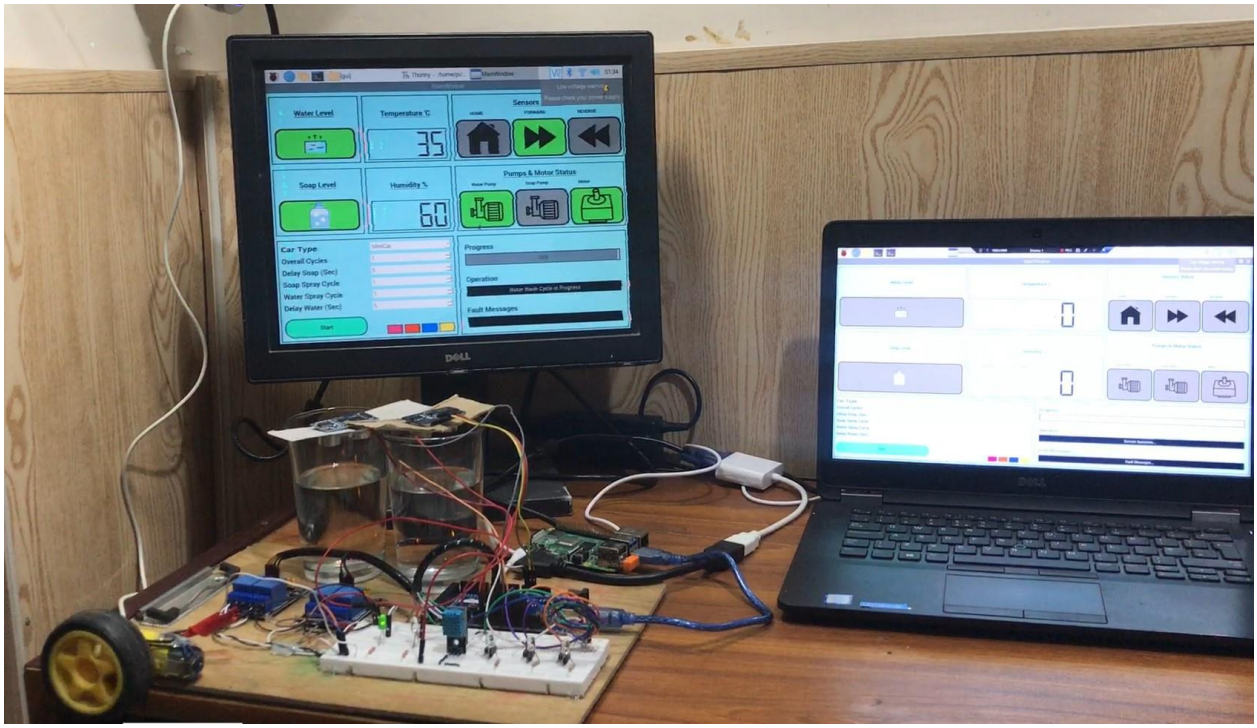
System Finding Home

Initially when system start, motor will find its home position before begin its spray cycles.



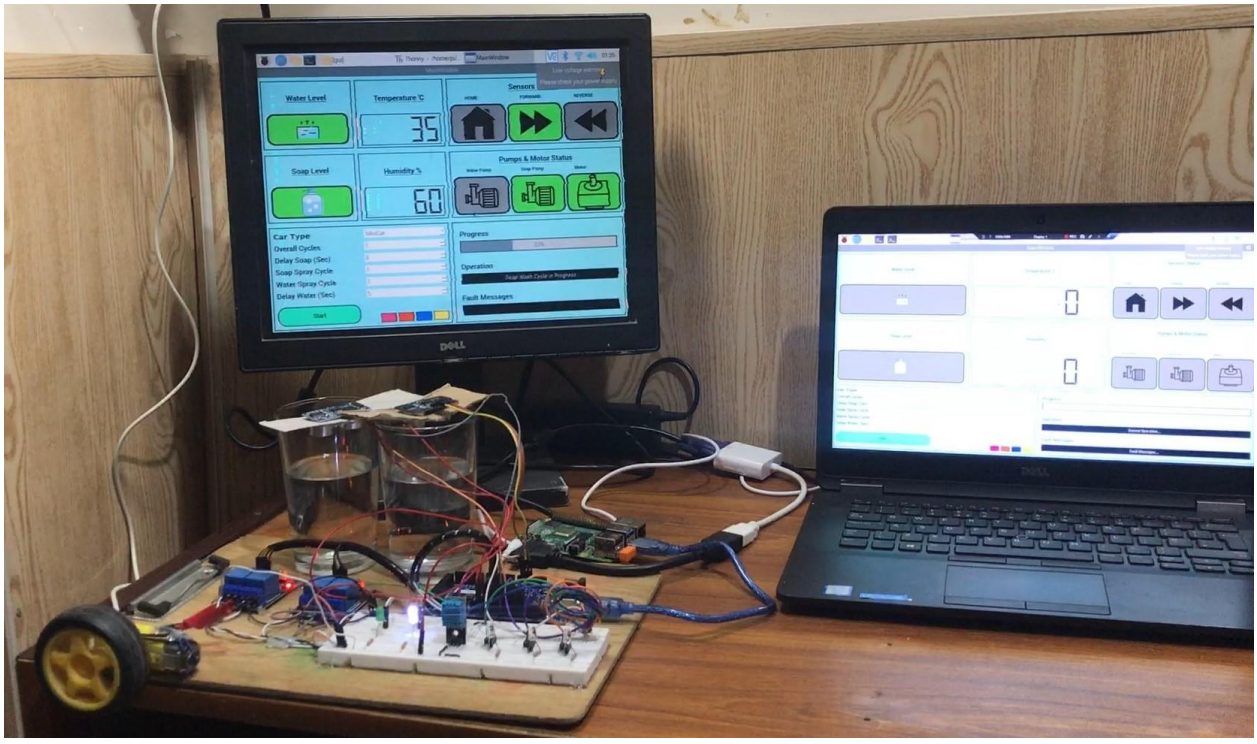
First Water Cycle Started

As system find its home position now it will start water pump and sprinkle it over the whole car. In hardware it will light up the Green Led on hardware First it will go up till forward limit sensor then comes in reverse direction until reach reverse limit sensor and as it reach there it comes to home position.



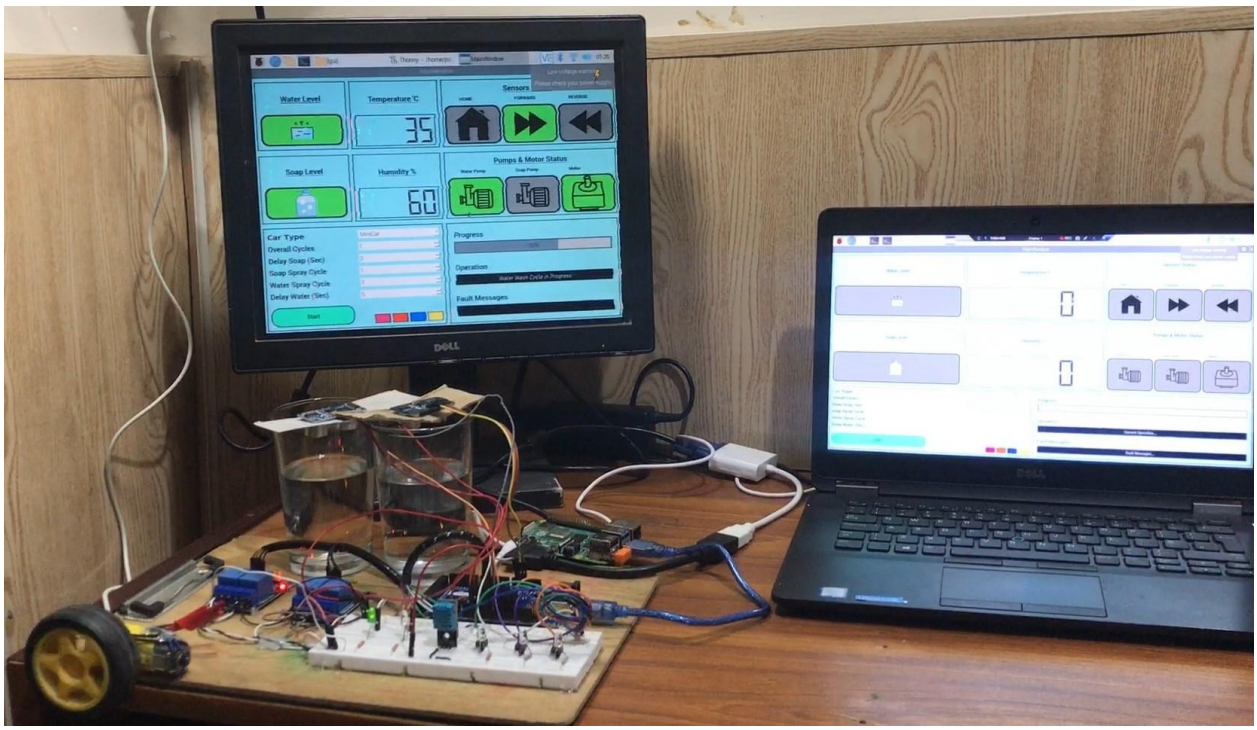
Soap Cycle Started

After completing the first water cycle system will start soap pump and sprinkle it over the whole car. In this process system adopt the same strategy as followed in water cycle. In hardware White Led will light up for Soap Pump.



Second Water Cycle Started

In the final phase, water sprinkle is again repeated to completely wash the car and complete its 100% progress.



Car Washed

Hence, your car is washed!

FINISHED!