**PETROL PUMP PROJECT**

**DOCUMENTATION FILE**

# **INTRODUCTION :**

This File consist of the Schematic and code required to install the project at the petrol pump.The code takes the time and date from sensor and write it into a csv file.Then few Command line Interface Commands were used to send that information to database and table.

Language: Python  
Database Software: SQLite  
Raspberry Pi Software: Raspbian Buster 4.19 (<https://www.raspberrypi.org/downloads/raspbian/>)  
Schematic Software: Fritzing

# **DISCUSSION:**

1. After Downloading the Raspbian from above given URL. Flash the SD Card using BalenaEtcher (<https://www.balena.io/etcher/>) .
2. Setup the Raspberry Pi. Upgrade and Update it using following CLI commands

$ sudo apt-get install update

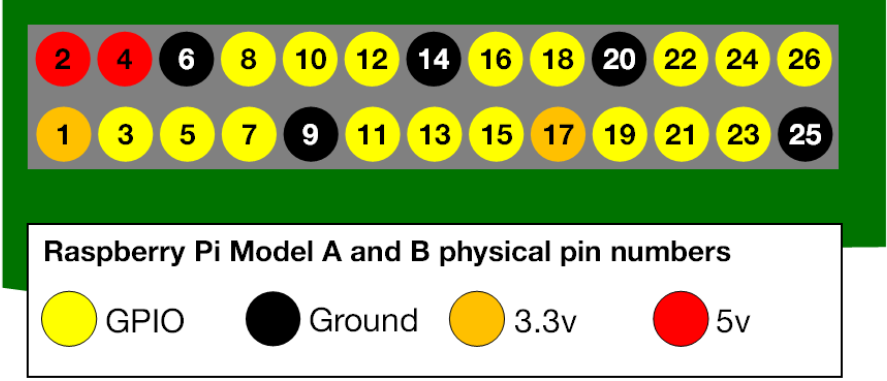
$ sudo apt-get install upgrade

1. Open Python IDE in Thonny and write the Following Code

|  |
| --- |
| import csv  import RPi.GPIO as GPIO # Import Raspberry Pi GPIO library  import datetime  def button\_callback(channel):  if(GPIO.input(channel)==GPIO.HIGH):  data=['sensor1',datetime.datetime.now().isoformat()]  with open ('/home/pi/Desktop/Project/data.csv', 'a') as f:  writer=csv.writer(f)  writer.writerow(data)  GPIO.setwarnings(False) # Ignore warning for now  GPIO.setmode(GPIO.BOARD) # Use physical pin numbering  GPIO.setup(8, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN) # Set pin 8 to be an input pin and set initial value to be pulled low (off)  GPIO.add\_event\_detect(8,GPIO.RISING,callback=button\_callback) # Setup event on pin 8 rising edge  def button\_callback(channel):  if(GPIO.input(channel)==GPIO.HIGH):  data=['sensor2',datetime.datetime.now().isoformat()]  with open ('/home/pi/Desktop/Project/data.csv', 'a') as f:  writer=csv.writer(f)  writer.writerow(data)  GPIO.setwarnings(False) # Ignore warning for now  GPIO.setmode(GPIO.BOARD) # Use physical pin numbering  GPIO.setup(10, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN) # Set pin 10 to be an input pin and set initial value to be pulled low (off)  GPIO.add\_event\_detect(10,GPIO.RISING,callback=button\_callback) # Setup event on pin 10 rising edge  def button\_callback(channel):  if(GPIO.input(channel)==GPIO.HIGH):  data=['sensor3',datetime.datetime.now().isoformat()]  with open ('/home/pi/Desktop/Project/data.csv', 'a') as f:  writer=csv.writer(f)  writer.writerow(data)  GPIO.setwarnings(False) # Ignore warning for now  GPIO.setmode(GPIO.BOARD) # Use physical pin numbering  GPIO.setup(12, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN) # Set pin 12 to be an input pin and set initial value to be pulled low (off)  GPIO.add\_event\_detect(12,GPIO.RISING,callback=button\_callback) # Setup event on pin 12 rising edge  message = input("Press enter to quit\n\n") # Run until someone presses enter  GPIO.cleanup() # Clean up |

1. Now Connect the Sensors as mentioned below

|  |  |
| --- | --- |
| **Sensor\_id** | **Raspberry PI GPIO** |
| 1 | Pin 8 (GPIO) and Pin 1 (3.3V) |
| 2 | Pin 10 (GPIO) and Pin 1 (3.3V) |
| 3 | Pin 12 (GPIO) and Pin 1 (3.3V) |



1. Now Make a DATA.CSV FILE in the same directory as your Petrolpump.py.
2. Run the Code and you will get values in data.csv file as sensor is pressed.
3. Now install SQLite Database on Raspberry PI using following command in CLI

$ sudo apt-get install sqlite3

1. Create a database using the following command

$ sqlite3 <database-name.db>

1. Set Mode in database to csv by using following command

sqlite3>> .mode csv

1. Import data from csv file to the table in database using following command

sqlite3>>.import <csv-file-location><table-name>

1. Check the table using following command

sqlite3>> .schema<table-name>

1. Use Query to Get output values

sqlite3>>SELECT \* FROM <table-name>;

## **Schematic Diagram of Project:**

