IoT based DC motor control, protection and monitoring

Arohi Srivastav, 2018111018
Sridhar M, 2018111021
Yash Bhansali, 2018101068

November 27, 2019

ABSTRACT

DC motor plays a very important role in different industries. In this project, we are discussing about a system which provides protection to the DC motor as well as helps in control and monitor various parameters. We have used ESP32 and web server also with the help of some transducers we can easily achieve our goal to protect and control the motor as well as to monitor various parameters. We have provided various controls through internet to avoid faults in DC motor

1. DEVELOPERS PROJECT DOCUMENTATION

1.1 Introduction:

1.1.1 Problem Statement

Controlling speed of a 150rpm DC motor in autopilot and user mode

1.1.2 Scope

Our project can be deployed to all those places where there is a moving/static device connected to motor and needs to control it wirelessly and monitor its live situation.

1.1.3 Purpose

- To automate speed control of motors.
- To collect data regarding speed variations of automated devices in chosen situation.
- To wirelessly control speed of motors

1.1.4 Overview of system

When in auto pilot mode, SR04 sensor is used to check the distance of obstacles coming either in front or rear side of the device send that data to ESP32 and control the speed of motor accordingly. Otherwise, in user mode, user wirelessly controls the speed of motor independent of data from SR04 sensor. Data of speed corresponding to distance of obstacle is collected.

1.2 Design document

1.2.1 System requirements

Hardware:

- Wifi connection
- Power supply
- 220V to 5V adapter
- WiFi connectivity

Software:

- Arduino IDE

1.2.2 System Specifications

- Microcontroller with WiFi connectivity: ESP32
- Windows 8 or Linux 16.04 or Mac OS Yosemite and above
- WiFi driver
- Back-end server for storing data e.g OneM2M, Thingspeak, etc. Here, http://onem2m.iiit.ac.in:443/webpage

1.2.3 Stakeholders

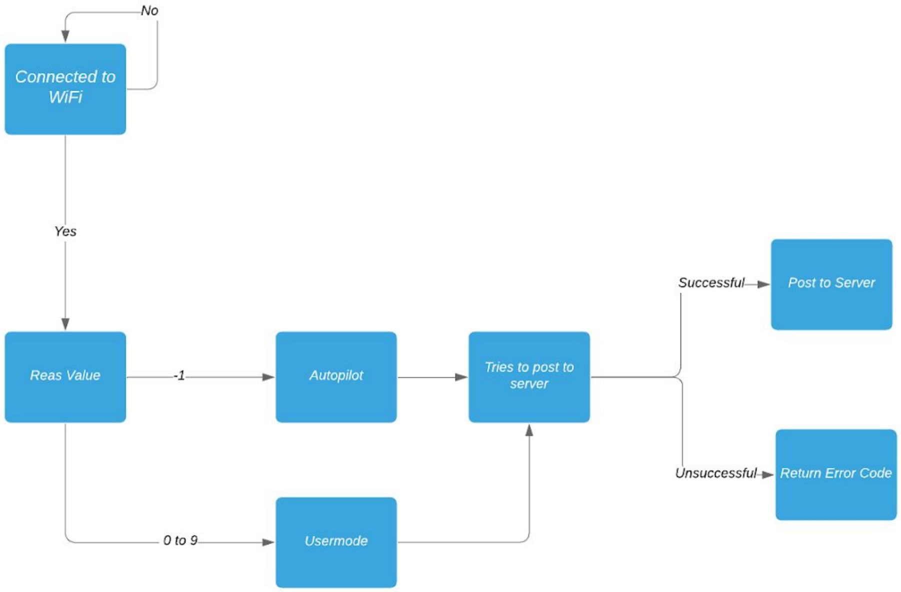
- Our Team members- Arohi Srivastav, Sridhar M and Yash Bhansali.
- Guiding Faculty- Dr. Madhava Krishna.
- Other Faculty involved in the course.
- Assigned Teachings Assistant- Chaitanya Kharyal and Faizan Khan.

1.2.4 Design Entity

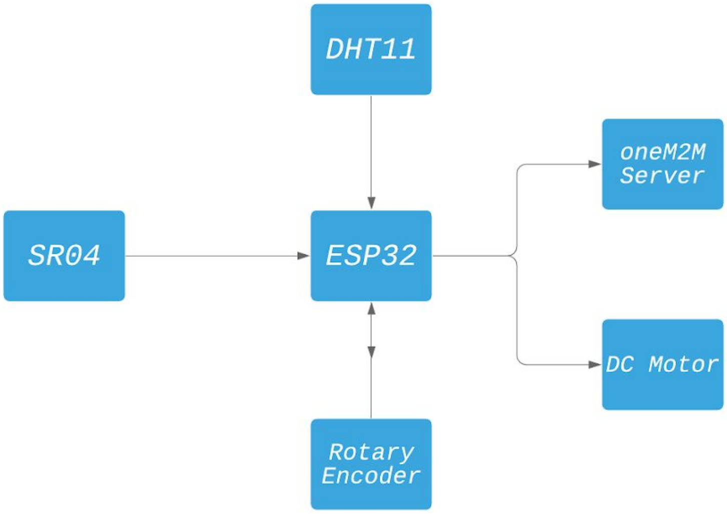
- ESP 32 board
- 150 rpm DC motor
- L293D Motor control chip
- Rotary encoder
- SR04 sensor
- DHT11 sensor

1.2.5 Design details

1.2.5.1 Conceptual flow



1.2.5.2 Entity interaction



1.2.6 Oprational requirements

- System need: Power supply is needed to keep ESP32 functional all the time. Wifi is needed to upload the data on the site so that observations can be made and analysis can be done.

- UI design: Our UI is made using HTML and CSS and can be used to switch between user and auto-pilot mode. It is also used for controlling speed of motor in user mode.

- Analytical System: Data is collected from the sensor at an interval of 5 minutes and sent to the OneM2M server. A python application has been designed by us that scraps the incoming data values at the remote server and stores it in a local database.On running the application, a graph is generated that gets updated in real-time. The values in the database are also separately displayed in the UI.

2. USERS PROJECT DOCUMENTATION

2.1 Introduction

The project aims at controlling speed of DC motor according to the proximity of obstacles. Also, it helps in controlling the speed wirelessly.
Our project can be deployed to all those places where there is a moving/static device connected to motor and needs to control it wirelessly and monitor its live situation.

2.2 System requirements

- Code can be edited, compiled and dumped through Arduino IDE, available across Windows and Linux platforms.
- Python 3 to run the web application to view the data, control mode of operation and manage speed of motor.
- Any browser to view the data analytics on localhost.
- 5V power supply for the ESP-32 board.

2.3 System working model

2.3.1 Base state

In base state, the relay will be connected to the ESP32 board but it won’t receive any signal so relay will maintain its default state which is normally closed and thereby completing the circuit and allowing the street light to be in ON condition.

2.3.2 Deployment

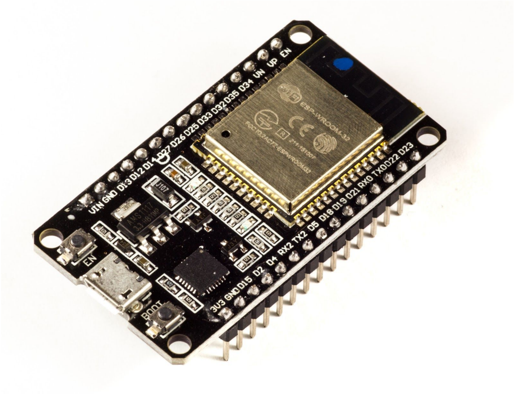
- Preferably avoid contact of sensors with water.
- Connect the ESP-32 board to a 5V power supply using provided adapter.
- Edit WiFi name and password in code before flashing it.
- Ensure that the ESP-32 board is deployed in a waterproof condition since it is easily fails when it comes in direct contact with enough water

2.3.3 Working State

In auto-pilot mode, the SR04 detects the distance value and sends it to the ESP32board. The ESP32 board processes the code by comparing the value received through the SR04 with the threshold values and if an obstacle is at a distance less than 10cm then it switches off the motor, between 10cm and 30 cm speed increases linearly with distance and when distance is greater than 30cm motor rotates at top speed. However, in user mode speed can be controlled via web application provided

3. HARDWARE SPECIFICATIONS

- ESP 32 board:

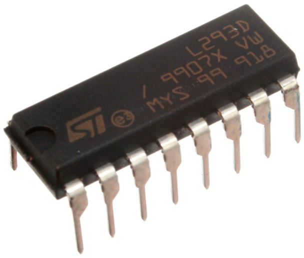


Low-cost, low-power SoC microcontroller with integrated Wi-Fi and dual-mode Bluetooth. Capable of working with operating temperatures ranging from −40°C to +125°C. Highly-integrated with in-built antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. Engineered for mobile devices, wearable electronics and IoT applications, ESP32 achieves ultra-low power consumption with a combination of several types of proprietary software. Can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor.

- DC motor:

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor.

- L293D Motor control chip:

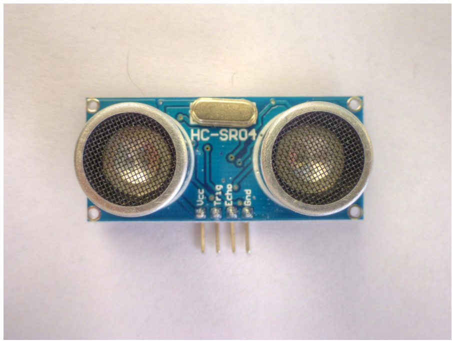


The L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. It works on the concept of H-bridge. H-bridge is a circuit which allows the voltage to be flown in either direction. Voltage needs to change its direction for being able to rotate the motor in clockwise or anticlockwise direction, Hence H-bridge IC are ideal for driving a DC motor.

- Rotary encoder:

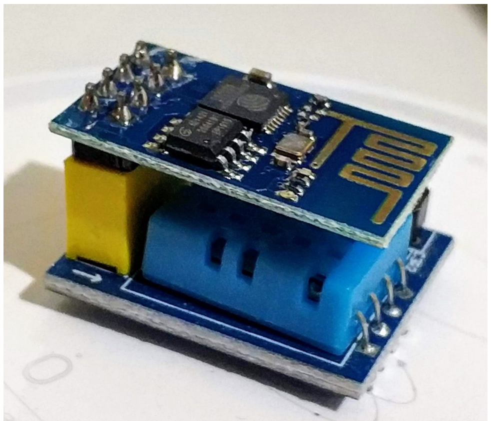
A motor encoder is a rotary encoder mounted to an electric motor that provides closed loop feedback signals by tracking the speed and/or position of a motor shaft. The Photoelectric Speed Sensor Encoder Coded Disc code wheel is a Slotted Opto isolator module, with an IR transmitter & a photodiode mounted on it. Performs Non-Contact Object Sensing.

- SR04 sensor:



The HC-SR04-Ultrasonic Range Finder used to measure distance and detects the objects. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The HC-SR04 ultrasonic sensor uses sonar to determine the distance to an object. Its operation is not affected by sunlight or black material.

- DHT11 sensor:



DHT11 temperature range is from 0 to 50 degrees Celsius with +-2 degrees accuracy. For measuring temperature these sensors use a NTC temperature sensor or a thermistor. A thermistor is a variable resistor that changes its resistance with change in temperature. These sensors are made by sintering of semiconductive materials such as ceramics or polymers in order to provide larger changes in the resistance with just small changes in temperature. The term “NTC” means “Negative Temperature Coefficient”, which means that the resistance decreases with increase of the temperature.

4. COMMUNICATION

ESP32 is WiFi and bluetooth enabled microprocessor. In this project we have used Wi-Fi for all the data transfer purposes.

5. SOFTWARE SPECIFICATIONS

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board.

6. DATA HANDLING MODEL

We have used oneM2M protocols for data handling and management. Initially we used *thinkspeak.com*. The purpose and goal of oneM2M is to develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with M2M application servers worldwide. A critical objective of oneM2M is to attract and actively involve organizations from M2M-related business domains such as: telematics and intelligent transportation, healthcare, utilities, industrial automation, smart homes, etc.

7. DATA VISUALISATION / ANALYSIS

Data obtained using our project can be used to derive multiple results. When deployed in static environment it can be used to calculate traffic variation for those surroundings. Otherwise, when deployed in dynamic situations on a large scale it can tell the speed of vehicles corresponding to the obstacle proximity. With this data we can tell which city has safer driving practices and also help in traffic management on the basis of speed of vehicles on different paths.