International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 25 Issue 5 – APRIL 2018 (SPECIAL ISSUE).

PROTOTYPE OF INTERNET OF THINGS (IOT) BASED AUTONOMOUS CAR

Mrs Nirmala[#1], Mrs Narmadha *[2],Vignesh JM[*3], Vivekanandan S[*4]

[#1]*Assistant Professor,* *[3,4,5] *UG scholar, Dept of Electrical and Electronics Engineering, Prince Shri Venkateshwara Padmavathy Engineering College*

*[2] *Assistant Professor, Dept of Electrical and Electronics Engineering, Jeppiaar Maamallan Engineering College*

***Abstract*: The autonomous vehicle was introduced with the intention of providing safer, faster and energy efficient travel. The main objective of this work is to introduce Internet of Things (IoT) in cars and to make the car either autonomous or semiautonomous. In this connection an attempt is made to integrate obstacle detection, vehicle to vehicle communication and server based control using Arduino microcontroller. A unique and novel approach in the above work is efficient Speed Management and Control System (SMCS) which provides real time data to server to have control over the vehicle.**

***Keywords*: Arduino, Internet of Things (IoT), Wireless Fidelity (Wi-Fi), Server based control, Speed Management and Control System (SMCS)**

I. INTRODUCTION

In the past few years the numbers of fatal road accidents have been in rise. Such accidents are mainly caused due to driver's carelessness, over speeding and traffic violations.

The concept of autonomous cars have been developed to reduce accidents and to travel faster with efficient energy management. IoT implementations in to our day today life are vast and will find an integral part in our daily life in the near future. IoT has lot of advantages. Since everything is controlled over the internet, the human intervention reduces and complex architecture could be designed. Human errors are reduced and increase the efficiency with faster time of accomplishment of a task. In Current scenario the Autonomous cars are programmed to operate on its own by sensing the environment. So the human interaction is devoid. Our day to day life isn't mechanized and complete reliance upon the technology is a matter of concern.

The current autonomous cars are based upon the machine learning concept and physical programming. The real time environment is sensed and the decisions are taken on its own. Inspite of more advanced technology, the practical analysis of physical world could not be done by the cars.

This disadvantage leads to the development of this prototype where the car is controlled or accessed by the server. The servers are completely managed by the technical person; hence in any case with the help of the real time data fed, precise control to the car could be provided.

The car is connected to the server using the internet/intranet through the Wireless fidelity (Wi-Fi). The Internet connectivity through Wi-Fi would be beamed by the Google's project Loon balloons. Every real time feed sensed by the sensors would be sent to the server. The server which is programmed to control the car in any complex situation requests for the control feed when some critical situation has to be analyzed.

Overall operation of the car is completely electric. Electric hybrid cars are going to rule the future. It is completely eco-friendly and the energy loss is less compared to the conventional fuel vehicles. The Tesla developed Powerwall – Lithium ion batteries are used to power the car. It could be charged by plugging into the socket or by using the solar arrays.

The cars built using this concept will have the most critical environment analyzation, offering a huge saving in the energy and decrease in travel time. It also offers great safety and giving tensionless commute.

II. INTERNET OF THINGS (IoT)

The term Internet of Things generally refers to scenarios where network connectivity and computing capability extends to objects, sensors and everyday items not normally considered computers, allowing these devices to generate exchange and consume data with minimal human intervention.

Connectivity Models:

IoT implementations use different technical communications models, each with its own characteristics.

Four common communications models described by the Internet Architecture Board include: Device-to-Device, Device-to-Cloud, Device-to-Gateway, and Back-End Data-Sharing. These models highlight the flexibility in the ways that IoT devices can connect and provide value to the user. The models implemented in this project are

i) Device-to-Cloud Communications.

ii) ii) Back-End Data Sharing.

Device-to-Cloud Communications:

In a device-to-cloud communication model, the IoT device connects directly to an Internet Cloud service like an application service provider to exchange data and control message traffic.

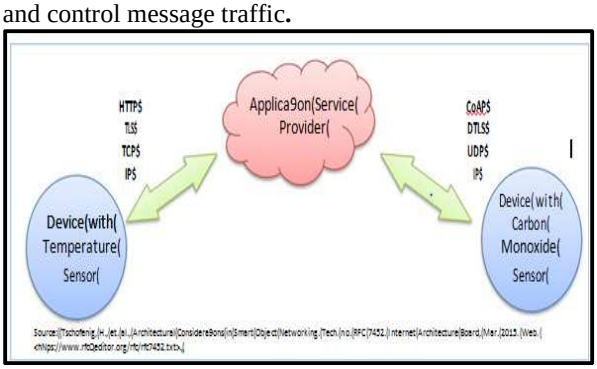


Figure 1: Device-to-Cloud Communications

This approach frequently takes advantage of existing communications mechanisms like traditional wired Ethernet or Wi-Fi connections to establish a connection between the device and the IP network, which ultimately connects to the cloud service

Back-End Data-Sharing Model:

The back-end data-sharing model refers to a communication architecture that enables users to export and analyze smart object data from a cloud service in combination with data

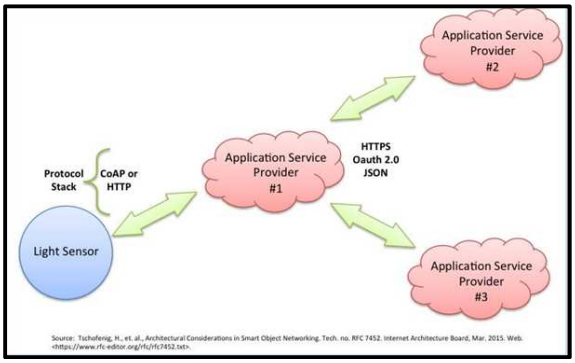


Figure 2: Back-End Data-Sharing Model

from other sources. This architecture supports "the [user's] desire for granting access to the uploaded sensor data to third parties". This approach is an extension of the single device-to-cloud communication model, which can lead to data silos where "IoT devices upload data only to a single application service provider". A back-end sharing architecture allows data collected from single IoT device data streams to be aggregated and analyzed.

Major considerations:

Many Internet of Things devices, such as sensors and consumer items, are designed to be deployed at a massive scale that is orders of magnitude beyond that of traditional Internet-connected devices. As a result, the potential quantity of interconnected links between these devices is unprecedented. Further, many of these devices will be able to establish links and communicate with other devices on their own in an unpredictable and dynamic fashion. Therefore, existing tools, methods, and strategies associated with IoT security may need new consideration.

Many IoT deployments will consist of collections of identical or near identical devices. This homogeneity magnifies the potential impact of any single security vulnerability by the sheer number of devices that all have the same characteristics.

Many Internet of Things devices will be deployed with an anticipated service life many years longer than is typically associated with high-tech equipment. Further, these devices might be deployed in circumstances that make it difficult or impossible to reconfigure or upgrade them; or these devices might outlive the company that created them, leaving orphaned devices with no means of long-term support. These scenarios illustrate that security mechanisms that are adequate at deployment might not be adequate for the full lifespan of the device as security threats evolve. As such, this may create vulnerabilities that could persist for a long time. This is in contrast to the paradigm of traditional computer systems that are normally upgraded with operating system software updates throughout the life of the computer to address security threats. The long-term support and management of IoT devices is a significant security challenge.

Many IoT devices are intentionally designed without any ability to be upgraded, or the upgrade process is cumbersome or impractical.

Many IoT devices operate in a manner where the user has little or no real visibility into the internal workings of the device or the precise data streams they produce. This creates security vulnerability when a user believes an IoT device is performing certain functions, when in reality it might be performing unwanted functions or collecting more data than the user intends. The device's functions also could change without notice when the manufacturer provides an update, leaving the

user vulnerable to whatever changes the manufacturer makes.

Some IoT devices are likely to be deployed in places where physical security is difficult or impossible to achieve. Attackers may have direct physical access to IoT devices. Anti-tamper features and other design innovations will need to be considered to ensure security.

III. SPEED MANAGEMENT AND CONTROL SYSTEM (SMCS)

This concept of Speed Management and Control System works in aid with the internet connectivity. It brings in the new concept of connected cars, where the cars are connected by the internet.

As the term connected cars implies the cars have been connected to sense the speed of drive and make subsequent changes to the speed of a particular car in accordance with speed sensed from other cars.

For instance consider a junction as shown in the figure 3. Car A approaches towards the junction from the north and car B approaches the junction from west. Let car A travel at a speed of 100 km/hr and car B travel at a speed of 80 km/hr. It is more economical to stop a slow moving car than the faster one; hence car B slows down at the junction till car A crosses and then allowed to accelerate at its full speed. This concept of control of speed of a particular car by sensing the speed of other cars is called as Speed Management and Control System (SMCS).

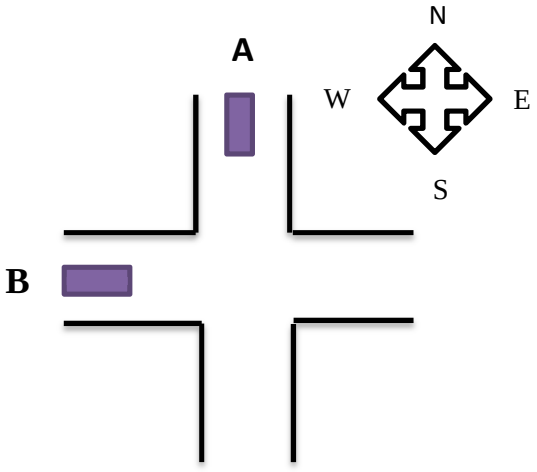


Figure 3: Concept of Speed management

Implementation:

Speed with which every car travels is found from Google maps. As Google gains access to all the Android devices across the globe, the speed of every car can be sensed, by which the concept of SMCS can be implemented.

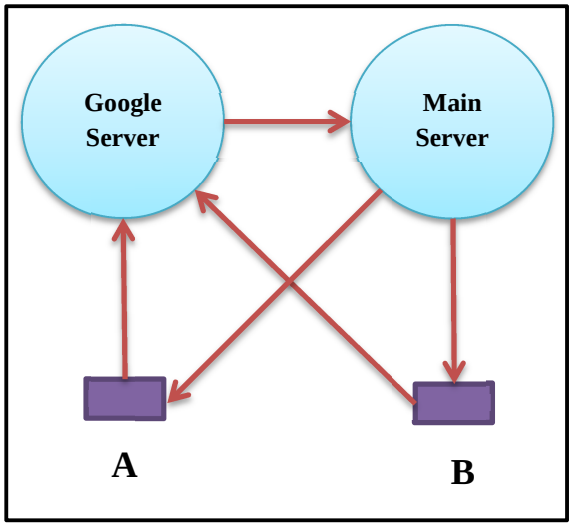


Figure 4: Network diagram of data flow

The speed of the moving car is obtained by mapping the exact GPS (Global positioning System) location of the car and various factors using the Google's algorithms. This data which is stored up in the Google database, which is been accessed by the Main database (main control database for the car). Based upon the data obtained subsequent changes are made to the algorithms and appropriate instructions are sent to the car to control the drive.

The data from the car to the database and vice versa are transferred through Internet Protocol (IP). The Wi-Fi beams are spread throughout the surrounding, which are used to collect data and send instructions. This ambitious project of beaming Wi-Fi in the surrounding was started as project Loon by Alphabet.Inc, which is explained next.

IV. PROJECT LOON: BALOON POWERED INTERNET TO EVRYONE

Project Loon is a network of balloons traveling on the edge of space, designed to connect to people in rural areas. Project Loon balloons float in the stratosphere, twice as high as airplanes and the weather. They are carried around the earth by winds and they can be steered by rising or descending to an altitude with winds moving in desired direction. People connect to the balloon network using a special internet antenna. The signal bounces from balloon to balloon, then to the global internet back to earth.

Figure 5: Project Loon

Balloons:

Project Loon balloons are designed and manufactured at scale to survive the conditions in the stratosphere, where winds can blow over 100 km/hr and the thin atmosphere offers little protection from UV radiation and dramatic temperature swings which can reach as low as -90ºC**.** Made from sheets of polyethylene, each tennis court sized balloon is built to last more than 100 days in the stratosphere before returning to the ground in a controlled descent**.**

Equipment:

Project Loon has taken the most essential components of a cell tower and redesigned them to be light enough and durable enough to be carried by a balloon 20km up in the stratosphere**.** All the equipment is highly energy-efficient and is powered entirely by renewable energy – with solar panels powering daytime operation and charging a battery for use during the night**.**

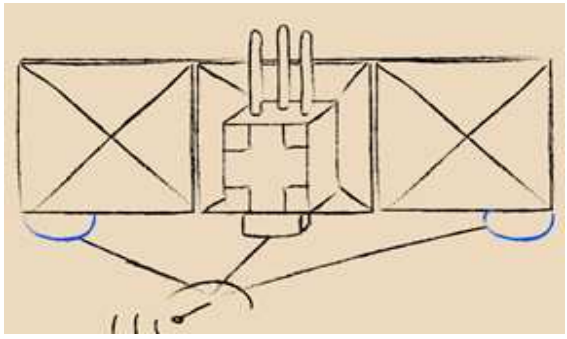


Figure 6: Transceivers, Solar panels, Flight Capsules

Launching:

Custom-built Autolaunchers are designed to launch Loon balloons safely and reliably at scale**.** Huge side panels provide protection from the wind as the balloon is filled and lifted into launch position, and then the crane is pointed downwind to smoothly release the balloon up in the stratosphere**.** Each crane is capable of filling and launching a new balloon into the Loon network every 30 minutes**.**

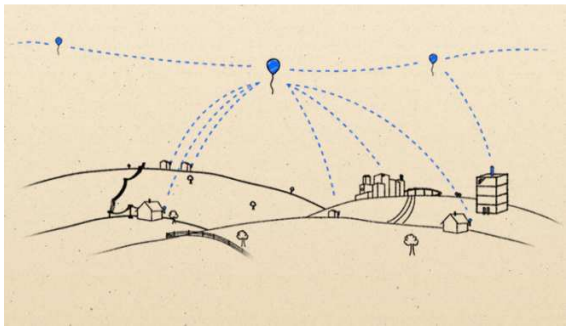


Figure 7: Project Loon- Design Boom

Navigating:

Balloons travel approximately 20km above the earth's surface in the stratosphere**.** In the stratosphere winds are stratified, and each layer of wind varies in speed and direction**.** To get balloons to where they need to go, project Loon uses predictive models of the wind and decision-making algorithms to move each balloon up or down into a layer of wind blowing in the right direction**.** By moving with the wind, the balloons can be arranged to provide coverage where it's needed**.**

V. PROTOTYPE CONFIGURATION

Overall control of the prototype is done by Arduino (Uno) microcontroller**.** It is based on ATmega328 with 14 digital I/O pins and 6 analog pins**.** The Ultrasonic sensor (HC-SR04) with crystal oscillator is used to measure the distance**.** In addition to the above it consists of Esp8266 (Wi-Fi module) to provide internet connectivity and motor driver to control the motors**.** Blynk server is used as control server (Main server) and data feed**.**

VI. OPERATION OF PROTOTYPE

A 12V, 1300 mAh Lithium Ion battery provides power to the prototype. Power is distributed to Arduino microcontroller, Esp8266 Wi-Fi module, HC-SR04 Ultrasonic sensor and to the motor driver. The Ultrasonic sensor detects the distance of the object in the front and sends appropriate signal to the Arduino microcontroller. Based on the data received from the sensor, the distance of the object in the front is calculated. This data is passed to the control server (Blynk Server) through Wi-Fi using Esp8266. The control server analyze the data and check for constraint limit violations and generates the control signal which is sent to the prototype and corresponding action take place.

The Ultrasonic sensor checks for the obstruction continuously every 50ms and sends the data to the microcontroller, which is transferred to the server. The server updates the mobile app as real time feed,

which provides the user with all data related to the prototype. In times of very critical situation the server alerts through notification, where at that point the control is transferred to the manual monitored control. Thus autonomy is attained by the prototype and conveyance is achieved.

VI. ARDUINO MICROCONTROLLER

Arduino Uno is a microcontroller board based on the ATmega328P**.** It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button**.** It contains everything needed to support the microcontroller; by simply connecting it to a computer with a USB cable or powering it with an AC-to-DC adapter or battery it can be started**.**

The Uno board and version 1**.**0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases**.** The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform**.**

Salient Features:

The Arduino Uno is a microcontroller board based on the ATmega328.It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP (In-Circuit Serial Programming) header and a reset button.

Specifications:

- Microcontroller : ATmega328P
- Operating Voltage : 5V (DC)
- Operating Limit : 7V – 12V (DC)
- DC current per I/O pin : 40mA
- DC current for 3.3V pin : 50mA
- Flash Memory : 32KB (0.5KB for Bootloader)
- SRAM : 2KB (ATmega328P)
- EEPROM : 1KB (ATmega328P)
- Clock Speed : 16 MHz

VII. ULTRASONIC SENSOR

HC-SRO4 Ultrasonic sensor is used to measure of an object. It works on the principle of SONAR (Sound Navigation and Ranging) which evaluates the object using radio waves. It generates high frequency sound waves and evaluates the echo which is received back by the sensor.

The basic operation principle is below; it uses IO port TRIG to trigger ranging. It needs 10 us high level signal at least. Module will send eight 40 kHz square wave automatically and will test if there is any signal returned. If there is signal returned, output will be high level signal via IO port ECHO. The duration of the high level signal is the time from transmitter to receiving with the ultrasonic**. Testing distance = duration of high level x sound velocity (340m/s) / 2**

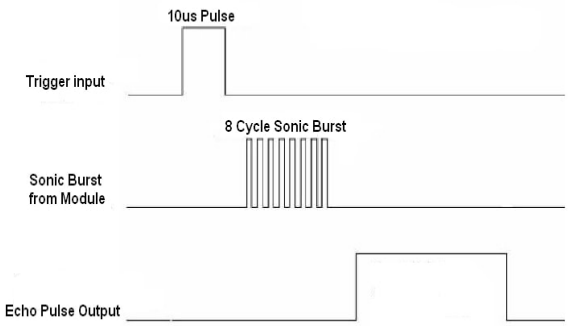


Figure 8: Timing diagram of Ultrasonic sensor

Specifications:

- Working Voltage : 5V (DC)
- Static Current : Less than 2mA
- Output Signal : Electric frequency signal, high level 5V, low level 0V
- Sensor angle : Not more than 15º
- Detection distance : 2cm - 450cm
- High precision : Up to 0.3cm
- Input trigger signal : 10us TTL impulse
- Echo signal : Output TTL PWL signal

VIII. ESP8266 WI-FI MODULE

The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to the Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware. The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces. It contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

Features:

- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- WiFi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4s guard interval
- Deep sleep power < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20 dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C

IX. BLYNK

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, and it can store data, visualize it and do many other cool things.

There are three major components in the platform:

- **Blynk App** - allows to you create amazing interfaces for your projects using various widgets we provide.
- **Blynk Server** - responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your private Blynk server locally. Its open-source could easily handle thousands of devices and can even be launched on a Raspberry Pi.
- **Blynk Libraries** - for all the popular hardware platforms - enable communication with the server and process all the incoming and out coming commands.

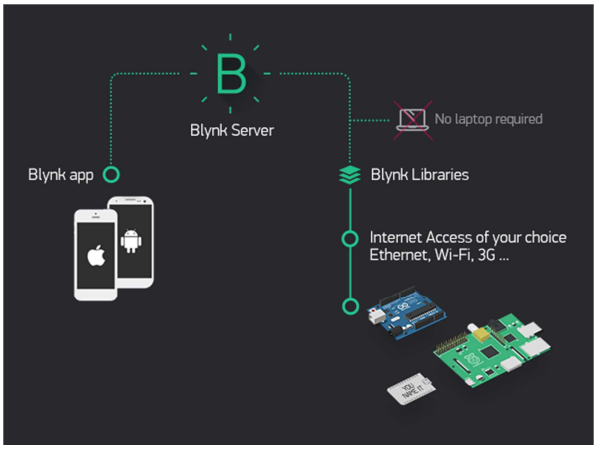


Figure 9: Blynk Server connection

X. SENSOR AND ACTUATOR

Sensor and actuator play a critical role in determining automotive control system performance. Sensor is a device that receives a signal or stimulus and responds to it in a distinctive manner. Sensor acts as a transducer which converts received signals to that form which can be interpreted by the processor.

An actuator is a mechanical device that converts the controller output signal into some form of action. The motor driver acts as an actuator. It is responsible for the directional control of rotation, starting and stopping of motor. A dual H-bridge transistor based motor driver is used to control two motors.

XI. SIMULATED PERFORMANCE CHARACTERISTICS

i) Supply voltage vs. Current Characteristic of ATmega328

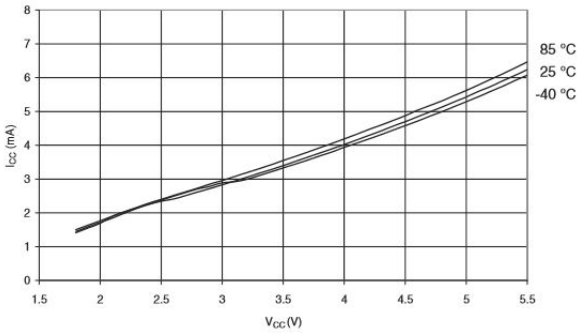
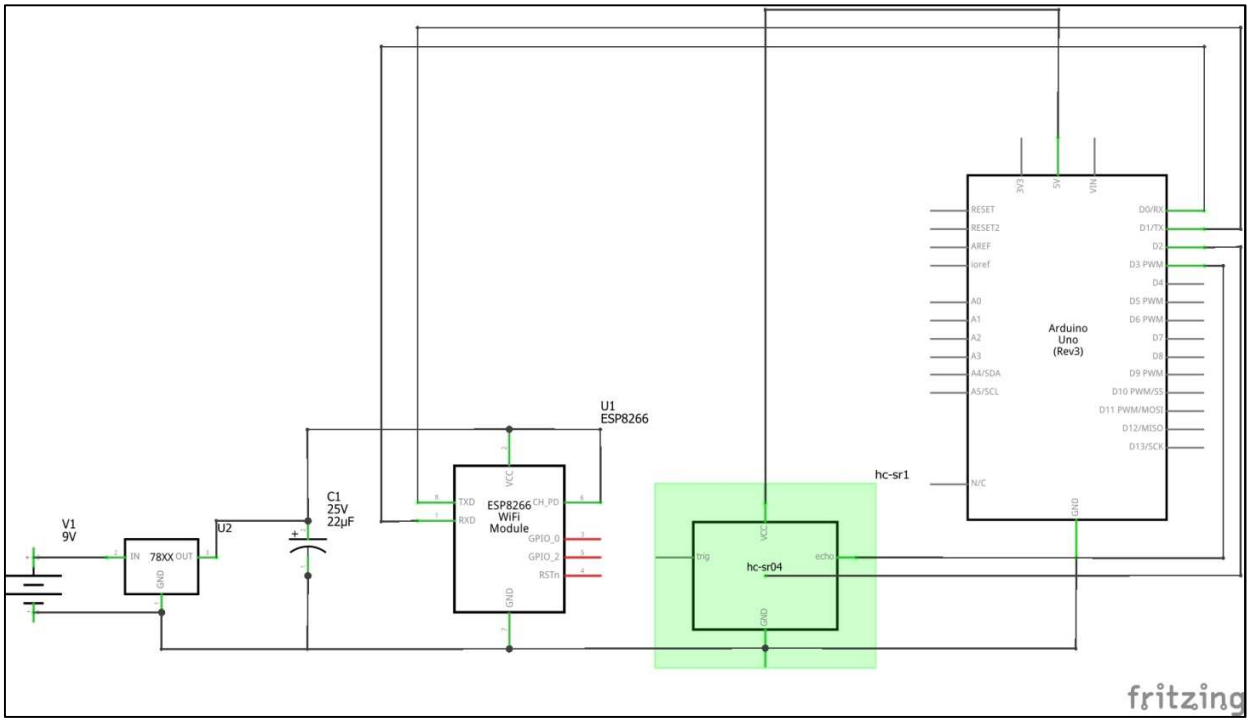


Figure 10: Supply current vs. Vcc



ii) I/O pin output voltage vs. source current

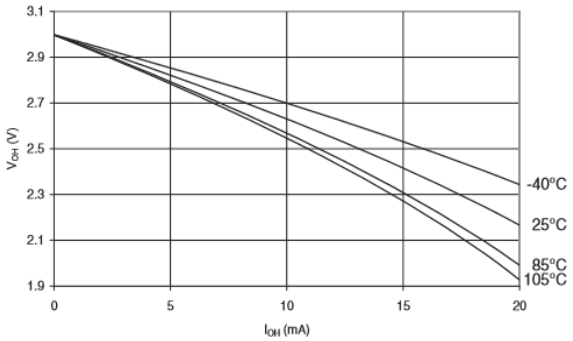


Figure 11: I/O pin output voltage vs. Source current

Fig. 10 shows the characteristic of supply voltage and current. The supply voltage vs. current have almost linear characteristic (i.e. current increases with rise in voltage).For the same voltage, the value of current increases with increase in temperature.

Fig. 11 show the characteristic of I/O pin output voltage vs. source current. It is observed that voltage level decreases with increase in I/O current. As the temperature increases, a dip in voltage is observed for same current

XII. CONCLUSION

IoT based Autonomous Car has been proposed and its suitability to the dynamic environment is instilled through various concepts governing the control and management of the prototype system. The prototype is designed and modeled suitable to the current technological growth which has its future scope of around 25 years. The prototype attains its place in the most energy efficient model compared to the current fuel driven cars.

High technical advancements made to the car outperform many autonomous cars which are currently in the development phase. The probability of technical glitch encountered is less in case of the prototype thus provides a safer transit with minimum energy usage.

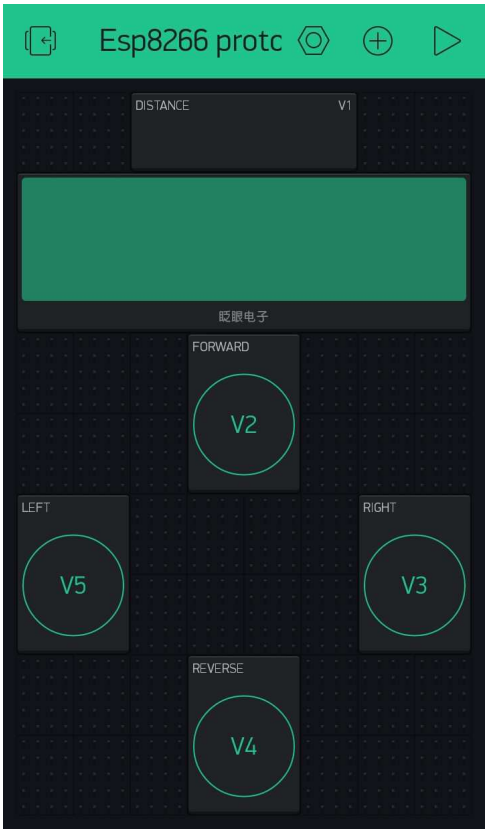


Figure 13: Blynk app

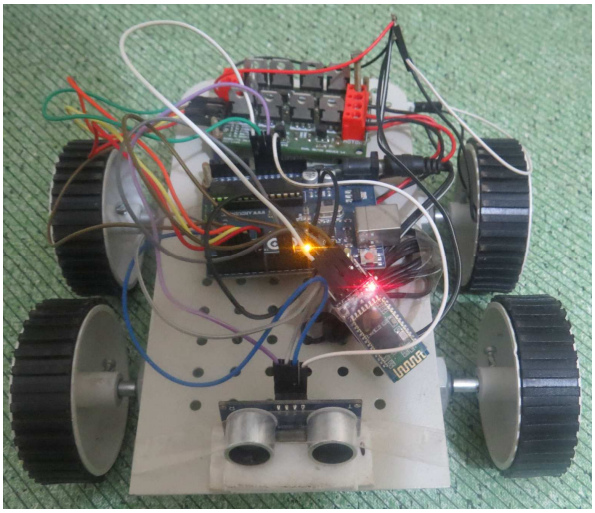


Figure 14: Prototype

REFERENCES:

[1] {ref: The Internet of Things: An Overview; www.internetsociety.org/sites/default/files/ISOC-IoT-Overview-20151014_0.pdf}
[2] {ref: The Autonomous Cars; https://www.lloyds.com/~/media/lloyds/reports/.../autonomous-vehicles-final.pdf}
[3] {ref: A Design Model for Automatic Vehicle Speed Controller; http://research.ijcaonline.org/volume35/number9/pxc3976166.pdf }
[4] {ref: The Tesla Powerwall: Does it bring something new? A Market Analysis; http://www.cee.uma.pt/morgado/Down/Tesla2_3.pdf }
[5] {ref: Tesla – Powerwall https://www.tesla.com/powerwall }
[6] {ref: How Google Maps figures out destination times; http://www.businessinsider.in/Ex-Google-Engineer-Reveals-How-Google-Maps-Figures-Out-Destination-Times/articleshow/28077362.cms }
[7] {ref: A Review paper on Project "LOONS"; http://www.ijarcce.com/upload/2016/march-16/IJARCCE%2035.pdf }
[8] {ref: ATMEL 8-Bit microcontroller 4/8/16/32KB in-system programmable flash- datasheet; http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf }
[9] {ref: Features of Arduino Uno R3; https://forums.oneplus.net/threads/top-5-features-of-arduino-uno-r3-my-first-article-here.330321 }
[10] {ref: Blynk; http://docs.blynk.cc/#blynk-server }
[11] {ref: Ultrasonic Sensor; https://www.sparkfun.com/products/13959 }