CHAPTER 1

INTRODUCTION

1.1. INTERNET OF THINGS

IoT is primarily marketing, not an engineering term. When we refer to it in the engineering terms, we normally mean an embedded microprocessor controlled system connected directly or indirectly to the web (e.g., web cameras, smart thermostats, health monitors, etc.). On the contrary, a typical distributed embedded system such as a mercial building lighting or heating system or an enterprise control system is never on the public Internet for obvious IT security reasons. As such, IoT refers to a highly secure and well protected embedded system that's fully controlled by microprocessors. That's one of the notions. Further, IoT is a conceptual framework or an architecture that considers how components (e.g. devices) will communicate with each other -enabling semantic interoperability.

Figure 1.1. Internet of Things

1.2. MOTIVATION

The motivation to do this project came from the ever-rising need for self-controlled, self-sufficient systems in today's changing world of technology. IoT has completely revolutionised technology and it has been become necessary more than ever 11 to develop such systems which can operate in a variety of environments for diverse applications. Locomotion is an important aspect in various forms of surveillance or control systems, we decided that if we could design a vehicle which could be controlled through the internet, it could be used in various applications. The prototype we have designed can be extended and expanded upon to suit various industry needs.

1.3. OBJECTIVE

The objective of this project is to build a self-sufficient IoT controlled vehicle which can be operated from a mobile device. The movement of the car's wheels is controlled by the various keys pressed on the GUI developed and accessed through a mobile phone. A prototype car has been constructed in such a way that it no longer needs a remote, but can be manoeuvred through commands given over the internet.

CHAPTER 2

HARDWARE

2.1. RASPBERRY PI 3

2.1.1. Description

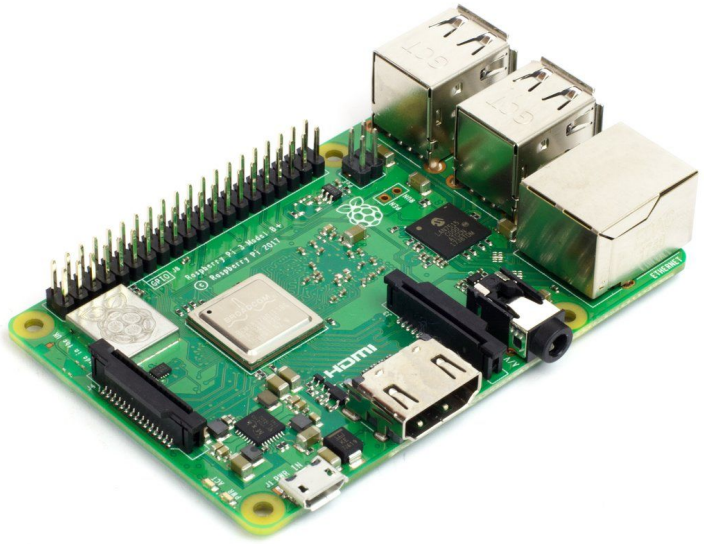

Figure 2.1. Raspberry Pi 3B+

The Raspberry Pi is an economical, “credit-card sized” computer that plugs into a display device (like monitor or Television) and uses a common keyboard and mouse for control. This adept device enables people of all ages to explore computing and to learn how to program in languages like Scratch and Python. It has been technically equipped to do everything a desktop computer can- from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games.

The Raspberry Pi is a range of small single-board computers developed by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools

and in developing countries. The Raspberry Pi is being used by people all over the world to learn to program and understand how computers work.

The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT. The dual-band wireless LAN comes with modular compliance certification, allowing the board to be designed into end products with significantly reduced wireless LAN compliance testing, improving both cost and time to market.

2.1.2. Technical Specifications

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header
- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

2.1.3. Features

- It has 10x Faster - Broadcom BCM2387 ARM Cortex-A53 Quad Core Processor powered Single Board Computer running at 1.2GHz.
- 1GB RAM-can now run bigger and more powerful applications.
- Fully HAT compatible.
- 40pin extended GPIO to enhance “real world” projects.

- Connect a Raspberry Pi camera and touch screen display.
- Stream and watch Hi-definition video output at 1080p.
- Micro SD slot for storing information and loading operating systems.
- 10/100 BaseT Ethernet socket to quickly connect the Raspberry Pi to the Internet

2.1.4. Processor

The Broadcom BCM2835 SoC used in the first generation Raspberry Pi is somewhat equivalent to the chip used in first generation smartphones (its CPU is an older ARMv6 architecture), which includes a 700 MHz ARM1176JZF-S processor, VideoCore IV graphics processing unit (GPU),and RAM. It has a level 1 (L1) cache of 16 KB and a level 2 (L2) cache of 128 KB. The level 2 cache is used primarily by the GPU. The SoC is stacked underneath the RAM chip, so only its edge is visible.
The Raspberry Pi 3 uses a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, with 512 KB shared L2 cache.

2.1.5. Performance

The Raspberry Pi 3, with a quad-core Cortex-A53 processor, is described as 10 times the performance of a Raspberry Pi 1. This was suggested to be highly dependent upon task threading and instruction set use. Benchmarks showed the Raspberry Pi 3 to be approximately 80% faster than the Raspberry Pi 2 in parallelized tasks.
The LINPACK single node compute benchmark results in a mean single precision performance of 0.065 GFLOPS and a mean double precision performance of 0.041 GFLOPS for one Raspberry Pi Model-B board.A cluster of 64 Raspberry Pi Model B computers, labeled "Iridis-pi", achieved a LINPACK HPL suite result of 1.14 GFLOPS (n=10240) at 216 watts.

2.1.6. Overclocking

The CPU chips of the first and second generation Raspberry Pi board did not require cooling, such as a heat sink, unless the chip was overclocked, but the Raspberry Pi 2 SoC may heat more than usual under overclocking.

Most Raspberry Pi chips could be overclocked to 800 MHz, and some to 1000 MHz. There are reports the Raspberry Pi 2 can be similarly overclocked, in extreme cases, even to 1500 MHz (discarding all safety features and over-voltage limitations). In the Raspbian Linux distro the overclocking options on boot can be done by a software command running "sudo raspi-config" without voiding the warranty. In those cases the Pi automatically shuts the overclocking down if the chip reaches 85 °C (185 °F), but it is possible to override automatic over-voltage and overclocking settings (voiding the warranty); an appropriately sized heatsink is needed to protect the chip from serious overheating.

Newer versions of the firmware contain the option to choose between five overclock ("turbo") presets that when used, attempt to maximize the performance of the SoC without impairing the lifetime of the board. This is done by monitoring the core temperature of the chip, the CPU load, and dynamically adjusting clock speeds and the core voltage. When the demand is low on the CPU or it is running too hot the performance is throttled, but if the CPU has much to do and the chip's temperature is acceptable, performance is temporarily increased with clock speeds of up to 1 GHz depending on the individual board and on which of the turbo settings is used.

In the highest (turbo) preset the SDRAM clock was originally 500 MHz, but this was later changed to 600 MHz because 500 MHz sometimes causes SD card corruption. Simultaneously in high mode the core clock speed was lowered from 450 to 250 MHz, and in medium mode from 333 to 250 MHz. The Raspberry Pi Zero runs at 1 GHz.

2.1.7. Networking

The Model A, A+ and Pi Zero have no Ethernet circuitry and are commonly connected to a network using an external user-supplied USB Ethernet or Wi-Fi adapter. On the Model B and B+, the Ethernet port is provided by a built-in USB Ethernet adapter using the SMSC LAN9514 chip. The Raspberry Pi 3 and Pi Zero W (wireless) is equipped with 2.4 GHz WiFi 802.11n (150 Mbit/s) and Bluetooth 4.1 (24 Mbit/s) based on Broadcom BCM43438 FullMAC chip with no official support for Monitor mode but

implemented through unofficial firmware patching and the Pi 3 also has a 10/100 Ethernet port.

2.1.8. Peripherals

The current Model B boards incorporate four USB ports for connecting peripherals. The Raspberry Pi may be operated with any generic USB computer keyboard and mouse.It may also be used with USB storage, USB to MIDI converters, and virtually any other device/component with USB capabilities. Other peripherals can be attached through the various pins and connectors on the surface of the Raspberry Pi.

2.1.9. Video:

The early Raspberry Pi 1 Model A, with an HDMI port and a standard RCA composite video port for older displays.

The video controller can emit standard modern TV resolutions, such as HD and Full HD, and higher or lower monitor resolutions and older standard CRT TV resolutions. As shipped (i.e., without custom overclocking) it can emit these: 640×350 EGA; 640×480 VGA; 800×600 SVGA; 1024×768 XGA; 1280×720 720p HDTV; 1280×768 WXGA variant; 1280×800 WXGA variant; 1280×1024 SXGA; 1366×768 WXGA variant; 1400×1050 SXGA+; 1600×1200 UXGA; 1680x1050 WXGA+; 1920×1080 1080p HDTV; 1920x1200 WUXGA.

Higher resolutions, such as, up to 2048×1152, may work or even 3840x2160 at 15 Hz (too low a frame rate for convincing video).Note also that allowing the highest resolutions does not imply that the GPU can decode video formats at those; in fact, the Pis are known to not work reliably for H.265 (at those high resolutions), commonly used for very high resolutions (most formats, commonly used, up to Full HD, do work).

Although the Raspberry Pi 3 does not have H.265 decoding hardware, the CPU is more powerful than its predecessors, potentially fast enough to allow the decoding of H.265-encoded videos in software. The GPU in the Raspberry Pi 3 runs at high clock frequencies of 300 MHz or 400 MHz.

The Raspberry Pis can also generate 576i and 480i composite video signals, as used on old-style (CRT) TV screens and less-expensive monitors through standard connectors – either RCA or 3.5 mm phone connector depending on models. The television signal standards supported are PAL-BGHID, PAL-M, PAL-N, NTSC and NTSC-J.

2.1.10 Real-time clock:

None of the current Raspberry Pi models have a built-in real-time clock, so they are unable to keep track of the time of day independently. As a workaround, a program running on the Pi can retrieve the time from a network time server or from user input at boot time, thus knowing the time while powered on. To provide consistency of time for the file system, the Pi does automatically save the time it has on shutdown, and re-installs that time at boot.

A real-time hardware clock with battery backup, such as the DS1307, which is fully binary coded, may be added (often via the I²C interface).

2.2 H-Bridge MOTOR DRIVER

2.2.1. Description

Figure 2.2. L298 Dual H-Bridge Motor Drive IC

This dual bidirectional motor driver is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently

control two motors of up to 2A each in both directions. It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. The circuit incorporates 4 direction LEDs (2 per motor), a heat sink, screw-terminals, as 27 well as eight Schottky EMF-protection diodes. Two high-power current sense resistors are also incorporated which allow monitoring of the current drawn on each motor through your microcontroller. An on-board user-accessible 5V regulator is also incorporated which can also be used to supply any additional circuits requiring a regulated 5V DC supply of up to about 1A. The circuit also offers a bridged mode of operation allowing bidirectional control of a single motor of up to about 4A.

2.2.2. Features:

- ➔ Motor supply: 6 to 35 VDC
- ➔ Control Logic: Standard TTL Logic Level
- ➔ Output Power: Up to 2 A each
- ➔ Current Sense Outputs
- ➔ Onboard Power Resistors Provided for Current Limit
- ➔ Enable and Direction Control Pins
- ➔ External Diode Bridge Provided for Output
- ➔ Heatsink for IC
- ➔ Power-On LED indicator
- ➔ 4 Direction LED indicators
- ➔ Dimensions**:** 3 x 2.5"
- ➔ H-Bridge's are typically used in controlling motors speed and direction, but can be used for other projects such as driving the brightness of certain lighting projects such as high powered LED arrays.

2.2.3. Working

An H-Bridge is a circuit that can drive current in either polarity and be controlled by Pulse Width Modulation (PWM). Pulse Width Modulation is a means of controlling the duration of an electronic pulse. In motors try to imagine the brush as a water wheel and electrons as the flowing droplets of water. The voltage would be the water flowing over the wheel at a constant rate, the more water flowing the higher the voltage. Motors are rated at certain voltages and can be damaged if the voltage is applied to heavily or if it is dropped quickly to slow the motor down. Thus PWM. Take the water wheel analogy and think of the water hitting it in pulses but at a constant flow. The longer the pulses the faster the wheel will turn, the shorter the pulses, the slower the water wheel will turn. Motors will last much longer and be more reliable if controlled through PWM.

2.2.4. Pins:

- Out 1: Motor A lead out
- Out 2: Motor A lead out
- Out 3: Motor B lead out
- Out 4: Mo (Can actually be from 5v-35v, just marked as 12v)
- GND: Ground
- 5v: 5v input (unnecessary if your power source is 7v-35v, if the power source is 7v-35v then it can act as a 5v out)
- EnA: Enables PWM signal for Motor A (Please see the "Arduino Sketch Considerations" section)
- In1: Enable Motor A
- In2: Enable Motor A
- In3: Enable Motor B
- In4: Enable Motor B
- EnB: Enables PWM signal for Motor B

2.2.5 Specifications:

- Double H bridge Drive Chip: L298H
- Logical voltage: 5V Drive voltage: 5V-35V
- Logical current: 0-36mA Drive current: 2A (MAX single bridge)
- Max power: 25W
- Dimensions: 43 x 43 x 26mm
- Weight: 26g

2.3 POWER BANK

Power Banks are comprised of a special battery in a special case with a special circuit to control power flow. They allow you to store electrical energy (deposit it in the bank) and then later use it to charge up a mobile device (withdraw it from the bank). Power Banks have become increasingly popular as the battery life of our beloved phones, tablets and portable media players is outstripped by the amount of time we spend using them each day. By keeping a battery backup close by, you can top-up your device(s) while far from a wall outlet. Power Banks are good for almost any USB-charged devices. Cameras, GoPros, Portable speakers, GPS systems, MP3 players, smartphones and even some tablets can be charged from a Power Bank.

Most commonly, a Power Bank will have a dedicated input socket for receiving power. This power can come from a USB socket on your computer but may charge faster when using a wall socket adapter. We most often see Power Banks use a Mini or Micro-USB socket for charging, and full-sized USB sockets for discharging. On very rare occasions, Power Banks can use the same socket for input and output, but this is rare and should not be assumed of any Power Bank, as trying to force power into an output can damage the battery. Always check the manual for specific instructions if you're not able to find a clearly marked input socket.

Depending on the capacity of the Power Bank and its current charge level, it can take quite a while to fill up. For example, a 1500mAh rated Power Bank should take

about the same time as your typical smartphone to charge. For larger banks, this time can be doubled, tripled or quadrupled. Most Power Banks have both an LED indicator to show when they are at capacity, and a safety cut-off to prevent overcharging and overheating. Whenever possible, remove the Power Bank from the charge when it is full, or at least avoid leaving it connected long-term after its full. Ambient temperature and power flow will also affect charge times, so it's best to keep it topped off regularly. There are two important life expectancies to consider: The number of charge/discharge cycles a Power Bank can reliably perform in its lifetime. A good Power Bank can hold the charge for 3 to 6 months with minimal loss. Lower quality Power Banks may struggle to retain a useful charge more than 4 to 6 weeks. In this regard, you get what you pay for, and if you need a long-term emergency power supply consider increasing your budget to ensure you're not going to be caught short. Most Power Banks will slowly lose charge over time, to a degree influenced by the environment and their treatment. For example, leaving a Power Bank in the car where the temperature can fluctuate greatly over time can shorten its lifespan.

2.4 LITHIUM ION BATTERY

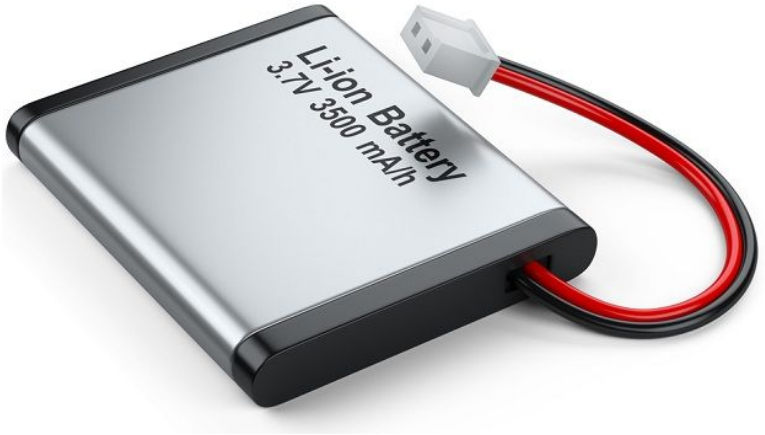

Figure 2.3. Lithium-ion battery

The term lithium-ion battery refers to a rechargeable battery where the negative electrode (anode) and positive electrode (cathode) materials serve as a host for the lithium ion (Li+). Lithium ions move from the anode to the cathode during discharge and are intercalated into (inserted into voids in the crystallographic structure of) the cathode. The ions reverse direction during charging. Since lithium ions are intercalated into host materials during charge or discharge, there is no free lithium metal within a lithium-ion cell. In a lithium-ion cell, alternating layers of anode and cathode are separated by a porous film (separator). An electrolyte composed of an organic solvent and dissolved lithium salt provides the media for lithium ion transport. For most commercial cells, the voltage range is approximately 3.0 V (discharged, or 0 % state-of-charge, SOC) to 4.2 V (fully charged, or 100% SOC).

CHAPTER 3

VEHICLE BUILDING

3.1 PROTOTYPE BLOCK DIAGRAM

All the components are mounted on top of the car base. The Power bank provides power supply to the Raspberry Pi, which is connected to the L298H Motor Driver. The Motor driver has connections to the motors as well as a switch which is connected to a battery to drive the motors.

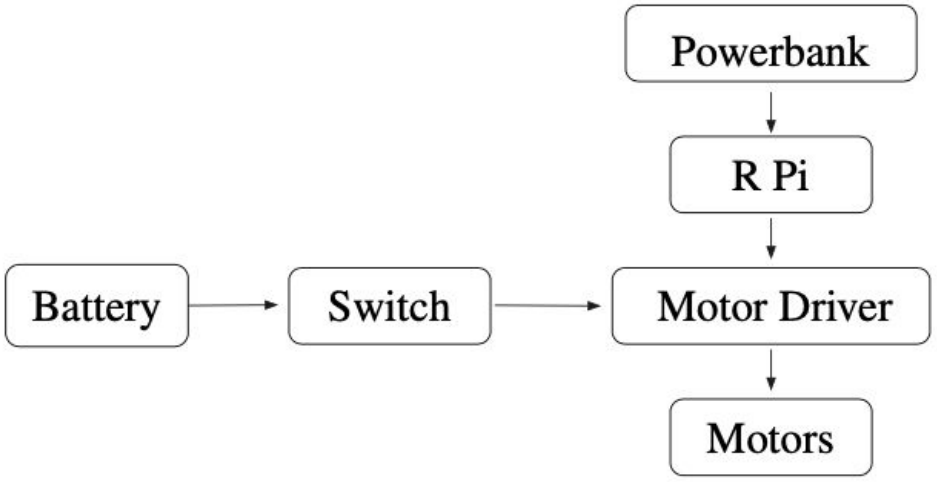

Figure 3.1. Prototype Block Diagram

3.2 CONNECTIONS

In order to correctly wire the prototype, we need to understand the Raspberry Pi GPIO pins which are used to connect various components to the Pi. These pins send the signals to the components connected, which could be turning on a light to driving a motor or reading data from a temperature or proximity sensor. The ones highlighted in green are the 17 basic GPIO pins which are what we are going to use in our project. These pins can be configured in either input or output mode. I connected the Anode (+ve) of 2 sets of LEDs for left and right turn signals with one

330Ω resistor each. Resistors help keep the amount of current passing through the LED's at a correct level, otherwise you could burn out the LED very quickly.

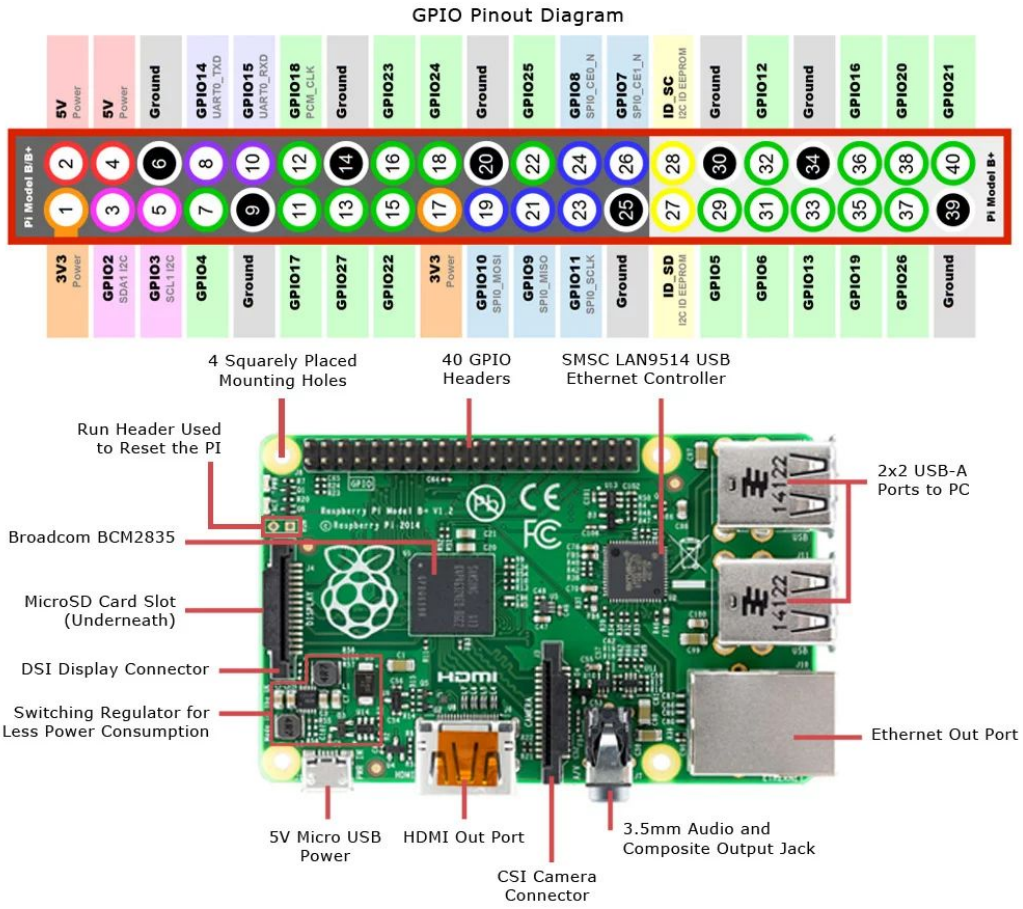

Figure 3.2. Raspberry Pi 3 Pinout

The reason we rely on a motor drive controller is that it can handle two motors up to 35V. Raspberry Pi only sends a maximum of 3.3V as its high signal. It not only provides enough power but controls direction and speed. L298N Motor Drive Controller Dual H-Bridge is the key component of my Raspberry Pi WiFi RC Car which powers all the motors and sends the signal to control the direction and speed of the motors. This controller can drive 2 motors with PWM (Pulse Width Modulation) signal. PWM in itself is a vast topic so we won't cover it here. Think of this as a technique to control the amount of power going through pretty much anything you want.

Setup the Pi Software*:*

The Pi uses node.js to run a web server; a wi-fi dongle on the PI uses your phone as a wireless hotspot to enable wifi communications. Once you enter the web address of the Pi a dialog box appears prompting you to begin racing; at that point, you can control your car by tilting your phone. An emergency stop is built into the app so if it loses comms to your phone the vehicle stops accelerating and steers forwards. The Pi-Blaster program allows pins 17 and 18 of the Pi to act as PWM outputs and control steering and throttle. The app runs on Raspbian using a Raspberry Pi although you could use the A model. To set up your Pi you need to download some software packages then setup your Pi to run the node.js program on boot.

Setup Your Phone: Configure PI to use Smartphone Wi-Fi

We now need to set your Pi up to use your phone as Wi-Fi.

- On your phone enable the wifi hotspot option.
- On your Pi disable the existing WiFi option and connect to your phone.
- You may need to enter a network key to do this; this will be in your phone somewhere.
- Check it all works by accessing the internet from your Pi.
- Reboot your Pi and check it still all works.
- You are now going to set your Pi IP address to static.
- Enter [ifconfig] write down your IP address for wlan0.
- Edit your network setup file [sudo nano /etc/network/interfaces] file and set to the current wlan0 IP address and from DHCP to static using the 'interfaces' file in this project as an example.
- Reboot the Pi and check all still works; your IP address will differ.

● Using Excel or a calculator work out the constants you need to use to convert your phone tilt values into the set PWM API demands required to control your car.

● Comment out the min-max demands if construct and enter your specific min max constants at the top.

● Now edit app.js.

● In the emergency stop function enter the values for steering and throttle that stop your Pi. Now run up the app a few times and convince yourself it's all working.

Configuring WebIOPi and Weaved:

The basic configuration required is to tell where our custom python script will reside which can be done by editing the config file under HTTP section using the following command.

```
$ sudo nano /etc/webiopi/config
[HTTP] enabled = true
port = 8000
doc-root = /home/pi/picar
welcome-file = index.html
```

To start webiopi service with verbose output and the default config file. This is recommended when developing and debugging your scripts.

```
$ sudo webiopi -d -c /etc/webiopi/config
```

You can also start/stop the background service, the configuration will be loaded from /etc/webiopi/config.

```
$ sudo /etc/init.d/webiopi start
$ sudo /etc/init.d/webiopi stop
//To check if the service is running or not
$ sudo /etc/init.d/webiopi status
```

Once you are done building your project you should put this service to auto start when the Pi boots. To manage service at boot here are the commands.

```
$ sudo update-rc.d webiopi defaults
$ sudo update-rc.d webiopi remove
```

Building User Interface:

To access the pi over local network open a browser and navigate to `http://ipAddressOfPi:8000/` from any device in your network. Make sure to type the IP address of the Pi in the URL. Default user "`webiopi`" and password is "`raspberry`". UI is pretty much a basic HTML page with images mapped for directions and stop commands that are wired to the on mouse down event. Those events are then mapped to the python macros which are exposed by the webiopi framework.

CHAPTER 4

SOFTWARE IMPLEMENTATION

4.1 WebIOPi

WebIOPi can control, debug, and use your Pi's GPIO, sensors and converters from a web browser or an app. It is the perfect Swiss-knife to make connected things.

- Written in Python, with facilities to load and execute a custom script, using a comprehensive structure with setup and loop functions
- Unified Serial/SPI/I2C support with a complete and consistent set of functions to control more than 30 devices, including most used analog converters, I/O expanders and sensors
- Javascript/HTML client library to make Web UI
- Python/Java clients, to make Pi-to-Pi systems or Android applications
- CoAP support brings the best Internet of Things protocol on the Pi, as future proof of Pi possibilities
- Includes simple web apps, to debug GPIO, devices and Serial interface

4.2 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as and introduce content into

the page directly. Others such as surround and provide information about document text and may `use them to interpret the content of the page.

Elements:

HTML documents imply a structure of nested HTML elements. These are indicated in the document by HTML tags, enclosed in angle brackets thus:

In the simple, general case, the extent of an element is indicated by a pair of tags: a "start tag" and "end tag". The text content of the element, if any, is placed between these tags. Tags may also enclose further tag markup between the start and end, including a mixture of tags and text. This indicates further (nested) elements, as children of the parent element. The start tag may also include attributes within the tag. These indicate other information, such as identifiers for sections within the document, identifiers used to bind style information to the presentation of the document, and for some tags such as the used to embed images, the reference to the image resource. Some elements, such as the line break, do not permit any content, either text or further tags. These require only a single empty tag (akin to a start tag) and do not use an end tag.

CHAPTER 5

EXECUTION

- After getting the car ready, and the required software installed, the car can be operated through a mobile device with mobile data.
- For running the code and controlling the car through a mobile handset, an App called JuiceSSH is downloaded.
- After the setup, the code is run on the mobile phone.
- The IP address of the registered mobile Raspberry Pi is entered on a web browser on the phone to give rise to the interface.
- Once a given button is selected on the smartphone, the car moves in that particular direction.

Figure 5.1. Web User Interface

CHAPTER 6

CONCLUSION & FUTURE SCOPE

6.1 CONCLUSION

We have developed a prototype for an IoT based vehicle which can be used in a variety of arenas practically. The model car is driven and manoeuvred solely by mobile with a proper internet connection. Thus a simple, cost-effective model has been successfully designed and tested. The results of the execution were as predicted and satisfactory

6.2 APPLICATIONS

The IoT based vehicle can be used in a wide range of applications. A few of these are, but not limited to:

1. Connected Cars

The concept of Connected Cars has been a great area of research and investment interest to companies such as BMW, Google and Apple. A connected car is a vehicle which is able to optimize its own operation, maintenance as well as the comfort of passengers using onboard sensors and internet connectivity.

2. Mobile Image Processing Applications

Since this car occupies a minimum amount of space, a camera can be mounted on top to access hard to reach destinations and the images may be streamed to an output device for analysis. These include the detection of various types of flora and fauna in forests and jungles, as well as hazardous components or pipeline failures in factories.

6.3 LIMITATIONS

Considering the fact that this just a prototype, extension to a larger arena will require greater mechanical and electronic support. Also, extra protection to make sure that the polarities of the battery do not meet is a crucial and vital criterion to be met. As the car is mobile, there is a great chance that the polarities may meet and cause catastrophic consequences.

REFERENCES

[1] P. Matzakos, J. Härri, B. Villeforceix, C. Bonnet, "**An IPv6 architecture for cloud-to-vehicle smart mobility services over heterogeneous vehicular networks**", Connected Vehicles and Expo (ICCVE) 2014 International Conference on, pp. 767-772, 2014.

[2] Artem Burkov, Nikolay Matveev, Andrey Turlikov, Alexey Bulanov, Olga Gahnina, Sergey Andreev, **"Upper bound and approximation of random access throughput over chase combining HARQ**", *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT) 2017 9th International Congress on*, pp. 143-147, 2017.

[3] Martin Stusek, Jiri Pokorny, Pavel Masek, Jan Hajny, Jiri Hosek, "**A non-invasive electricity measurement within the smart grid landscape: Arduino-based visualization platform for IoT**", *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT) 2017 9th International Congress on*, pp. 423-429, 2017.

[4] Xiaowei Yang, Xiaoxiao Wang, Yuan Wu, Li Ping Qian, Weidang Lu, Haibo Zhou, "**Small-Cell Assisted Secure Traffic Offloading for Narrowband Internet of Thing (NB-IoT) Systems**", *Internet of Things Journal IEEE*, vol. 5, no. 3, pp. 1516-1526, 2018.

[5] Zhaobin Deng, Yipeng Zhou, Di Wu, Guoqiao Ye, Min Chen, Liang Xiao, "**Utility Maximization of Cloud-Based In-Car Video Recording Over Vehicular Access Networks**", *Internet of Things Journal IEEE*, vol. 5, no. 6, pp. 5213-5226, 2018.

[6] E.-K. Lee, M. Gerla, G. Pau, U. Lee, J. H. Lim, "**Internet of Vehicles: From intelligent grid to autonomous cars and vehicular fogs**", *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 9, pp. 1-14, Sep. 2016.

[7] V. B. Iversen et al., "**Teletraffic engineering handbook**", *ITU-D SG*, vol. 2, pp. 16, 2005.

[8] G. Cardone, A. Corradi, L. Foschini, R. Ianniello, "**ParticipAct: A large-scale crowdsensing platform**", *IEEE Trans. Emerg. Topics Comput.*, vol. 4, no. 1, pp. 21-32, Jan./Mar. 2016.

[9] J. Gozalvez, M. Sepulcre, R. Bauza, "**IEEE 802.11p vehicle to infrastructure communications in urban environments**", *IEEE Commun. Mag.*, vol. 50, no. 5, pp. 176-183, May 2012.

[10] Y. Han et al., "**A service-based approach to traffic sensor data integration and analysis to support community-wide green commute in China**", *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2648-2657, Sep. 2016.