# Infrared Tyre Temperature System

**Maxamillian Toby Hutchings**

UP892525

School of Computing

Internet of Things

# Table of Contents

# Chapter 1

# Introduction

## 1.1 Internet Of Things

The term Internet of Things was first noted by Kevin Ashton and was used to describe a system in which objects in the physical world could be connected to the Internet by sensors. Later in 2012, the International Telecommunications Union (ITU), defines the term Internet of Things as "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interpolated information and communication technologies" (International Telecommunication Union, 2012). IoT systems originally focused on the use of Radio–Frequency Identification (RFID) tags, but has developed to be used alongside multiple types of hardware to be used in thousands of situations.

## 1.2 Scenario

The temperature of tyres in track racing is very important in making sure that tyres are within the optimal operating window. The tyre temperatures need to be displayed to both the teams and the drivers, as the drivers will need to manage tyre temperatures to get the most out of the tyres, and teams can use this information to calculate tyre degradation levels and make strategical decisions based on them.

Modern Formula One cars can have over 300 sensors on each car (McLaren, 2018) to measure every aspect of the car to get the most out of the cars performance, including tyre temperatures. Tyres that are overheating will cause blistering, where the inner temperature of the tyre causes the bonds between the rubber layers to fail and sheer of in large chunks, meaning that the tyres will lose grip significantly. Tyres that are too cold gives so little grip that the cars would have significant wheel spin, which can also cause graining. Therefore, it is very important to measure a tyres temperature and keep it within the tyres optimal operating window.

Although the use of IoT for measuring tyre temperatures is already an integral part of Formula One, it is not common to have temperature sensors in lower class auto racing leagues and road vehicles.

Therefore this project aims to produce an IoT system to measure tyre temperatures whilst the vehicle is driving. The system should measure the temperature and present the temperature to the user, either through the exact temperature values, or through LED lights to show if the tyre is too cold, too hot, or in the optimal range. The system should be small enough to fit on a vehicle without applying extra significant weight, whilst also providing accurate temperature readings. The system should also be energy efficient, so that the system should be able to run on a small battery pack whilst also lasting a good length of time. The outputs would be shown to the driver of the car through the use of LEDs in the dashboard or steering wheel, and the analytical data will be sent to the team so that they can work out tyre degradation levels and calculate pit stop strategies.

The system that is to be implemented in this project, is a system that includes the sensor to measure the temperature, along with the output to the user with the exact temperature values and a LED light system to signify the temperature to the user. To simulate the information the team would see, the system will also provide some basic analysis graphs to show the change in temperature over time, which will be outputted on a mobile device.

# Chapter 2

# Details of the Design

## 2.1 Components Used

**ESP8266**

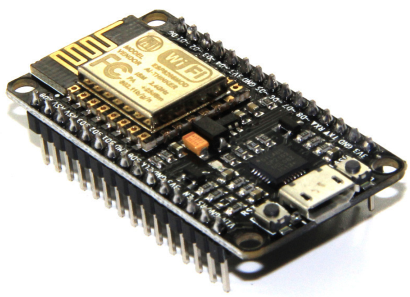

**Figure 2.1:** NodeMCU ESP8266

The ESP8266 is a Micro Controller Unit (MCU), System on a Chip (SoC) with a Wi–Fi transceiver that was developed by Espressif, that runs the open source Lua based NodeMCU firmware. The NodeMCU ESP8266 supports compatibility with multiple development environments, such as Arduino IDE, which gives it access to large amounts of external libraries and support for programs to be written in multiple languages, like C++. With the ESP8266s

Wi–Fi library, the chip can operate as a station or a soft access point, meaning it is perfect for an IoT system where the device cannot connect to a Wi-Fi enabled device directly. The NodeMCU ESP8266 has connectivity through a mirco–USB which can be connected to a battery pack, meaning that the microcontroller can be left in a place without connectivity to a socket or secondary machine giving it the ability to be placed pretty much anywhere. See Figure 2.1, for the image of the microcontroller.

### DHT11

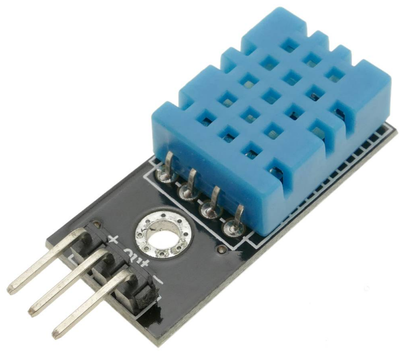

**Figure 2.2:** DHT11 Temperature and Humidity Sensor

The DHT11 is a basic digital temperature and humidity sensor that can interface with the NodeMCU ESP8266. The DHT11 uses a resistive–type humidity sensor and a NTC thermistor to accurately measure the humidity and temperature of the air in the sensors area. The temperature sensor works by using a Negative Temperature Coefficient (NTC) thermistor, which causes a decrease in its resistance value with increase in temperature ("DHT11 Sensor and its Working", n.d). The humidity sensor uses a capacitive humidity sensing element which consists of two electrodes with a moisture holding substrate as a dielectric between them ("DHT11 Sensor and its Working", n.d), these changed resistance values are measured and converted to digital form. The DHT11 is noted to operate in the temperature range of 0°C to

50°C with a 2°C degree accuracy, and a humidity range of 20% to 80% with a 5% accuracy ("DHT11 Sensor and its Working", n.d). See Figure 2.2, for image of the temperature and humidity sensor. The DHT11 is a good choice due to its low cost, low power consumption and precise accuracy.

**MLX90614**


**Figure 2.3:** MLX90614

The MLX90614 is a non–contact infrared thermometer that has high accuracy with a large operating range. The sensor has a temperature range of -20°C to 120°C, with an accuracy resolution of 0.02°C. The MLX90614 has both an IR sensitive thermophile detector chip and a signal conditioning ASSP integrated within the TO–39 can. An optical filter is integrated to provide ambient and sunlight immunity. The MLX90614 is a good choice due to its large operating range and accuracy in measuring in that range, as well as the low cost of the sensor. This temperature sensor is more suitable for this situation, as the DHT11 would require direct contact with the wheel in order to accurately measure the temperature, which would not be useful in measuring the tyre whilst the vehicle is moving. Whereas the MLX90614 can be pointed at the tyre to gather the temperature data, making it much more suitable for live feeds to the tyre whilst moving. See Figure 2.3, for an image of the sensor.

## 2.2 Electrical Design

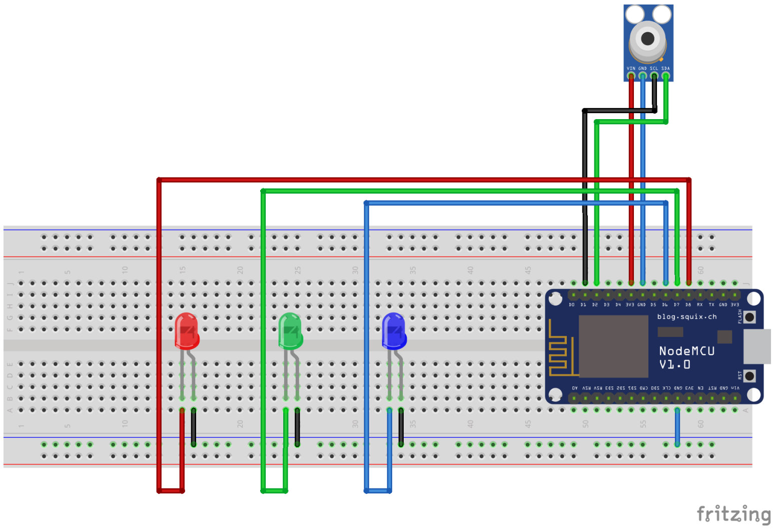


**Figure 2.4:** Electrical Design

## 2.3 Software

The software that this project uses will be the Arduino IDE, with the server and user interface being hosted through Blynk.

### Blynk

Blynk is an application designed for iOS and Android which is made to interact with IoT devices. Blynk can "control hardware remotely, it can display sensor data, it can store data, visualize it" and many more things ("Blynk Documentation", n.d). The mobile application handles the creation of the user interface through the use of various drag–and–drop widgets, which can display the sensor information in multiple ways like gauges, value displays, charts, push notifications, etc. In terms of the communication between the IoT device and the mobile application, Blynk provides a cloud server which is responsible for this. Blynk also has extensive libraries that support multiple hardware platforms.

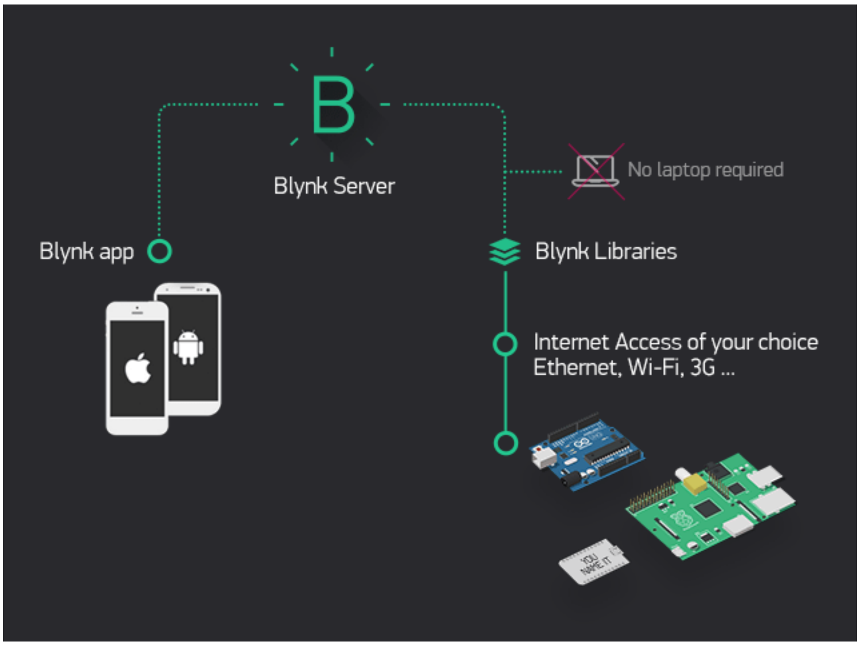


**Figure 2.5:** Blynk Architecture ("Blynk Documentation", n.d)

## 2.4 Flowchart

The system will be started by the battery being plugged into the NodeMCU, which will begin gathering the temperature data from the sensor. The MLX90614 provides two output values, the object temperature and an ambient temperature value. The object temperature data returned from the sensor will be put through a series of checks that determine whether the tyre is too hot, too cold, or in the correct operating window.

The NodeMCU will send results to the Blynk server, the data includes a text string stating what state the tyre is in, as well as the exact temperature value. Both the object and ambient temperature values will also be sent to the Blynk server, which will be displayed in the form of two gauges on the Blynk app.

The NodeMCU will also provide local outputs, in the form of LED lights that correspond to the tyres state, where a blue LED is cold, red LED is hot, and

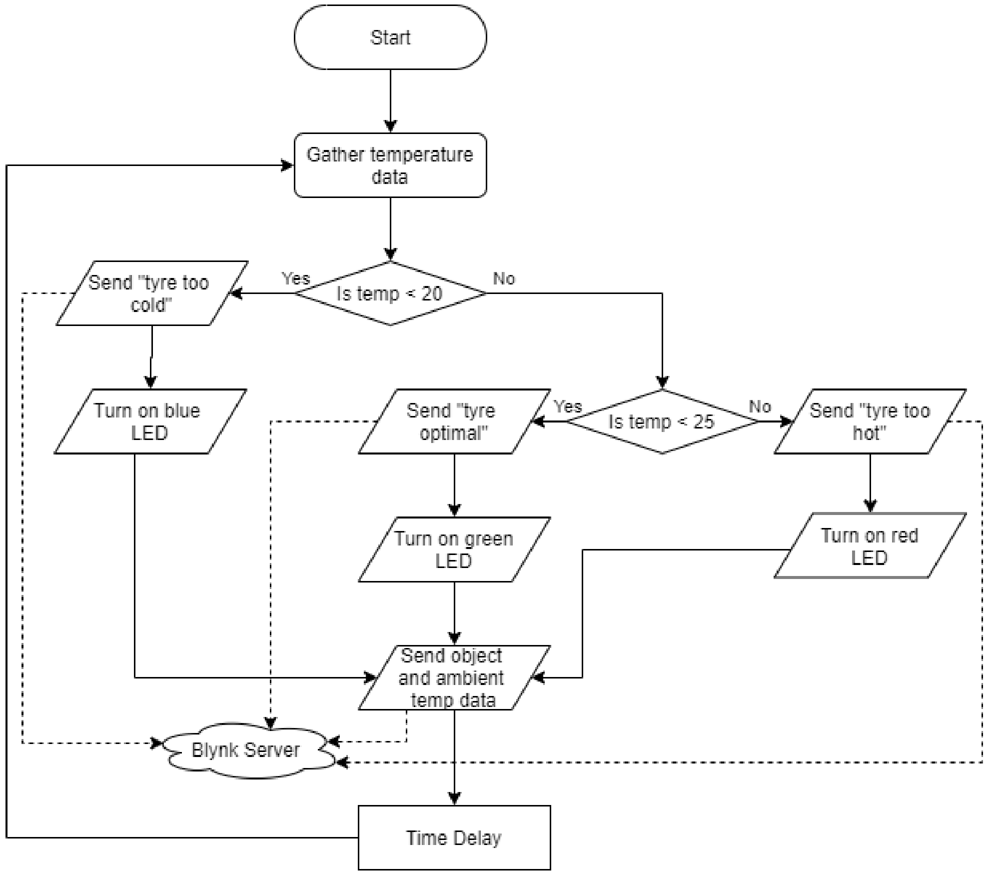


**Figure 2.6:** Flowchart Design

green is optimal. Once the outputs have been sent, then there will be a 2 second delay, as to not spam the mobile phone with updates. See Figure 2.6 for flow chart.

## 2.5 Cost Estimation

| **Component** | **Price** |
|---|---|
| NodeMCU ESP8266 | £5.00 |
| MLX90614 | £10.60 |
| 3x 5mm Coloured LEDs | £1 |
| 10x Jumper Wire | £2.37 |
| Total | £18.97 |

**Table 2.1:** Requirement Set 2 Table

# Chapter 3

# Implementation

The implemented electric circuit is the same as what was mentioned in the design. The only variance is that the ground wires connected to the LED lights are connected directly to the NodeMCU, rather than along the power rails as originally designed.

The change was due to the power rails not working when connecting the LED ground cable to the ground power rail. Another variance is the colour of the LEDs, as I did not have access to any green or blue LEDs when creating the system.

The layout however is the same, where the red LED indicates an overheating tyre, the middle yellow LED is an optimal tyre, and the right yellow LED is a cold tyre. See Figure 3.1 and Figure 3.2 for images of the implemented electrical design.

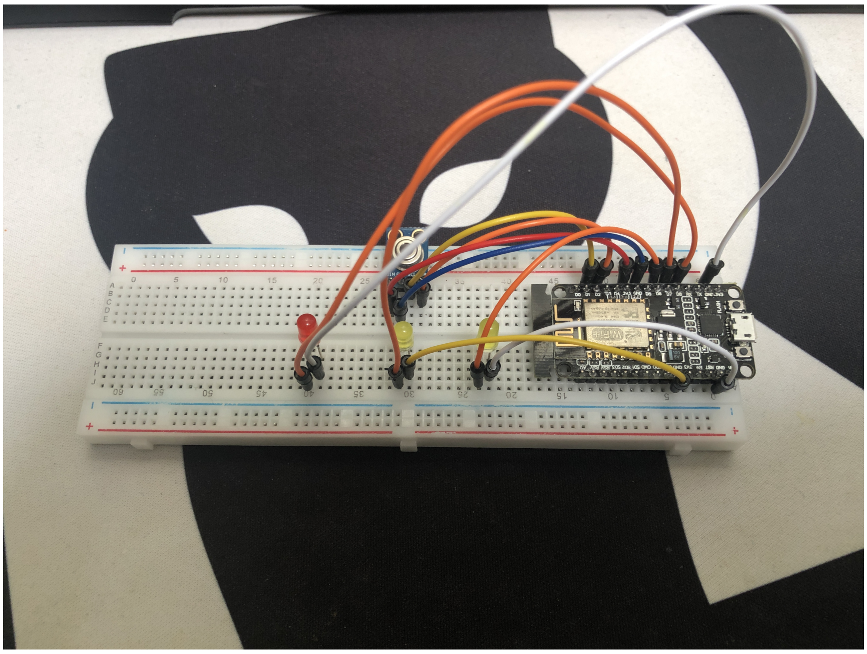

**Figure 3.1:** View of the implemented design

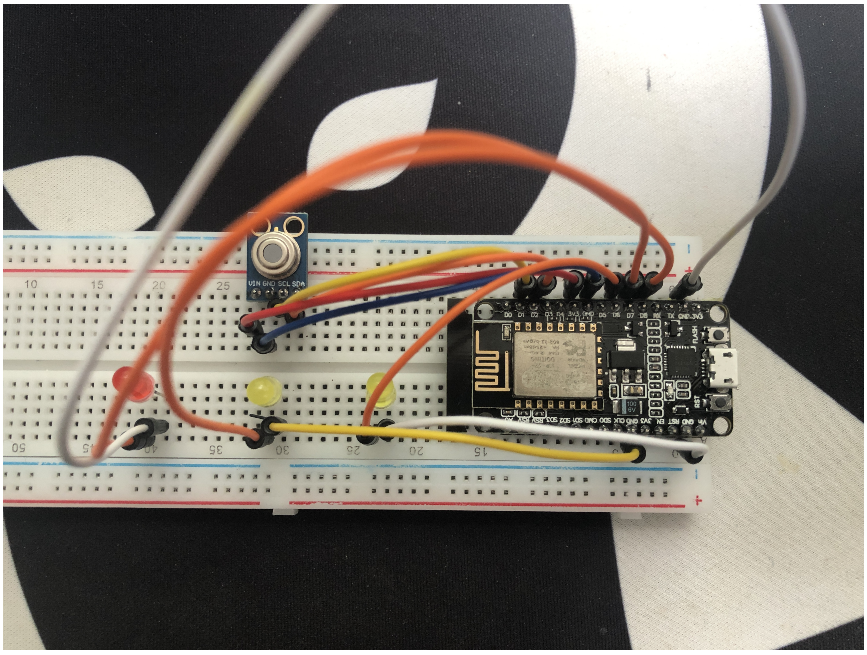

**Figure 3.2:** Another view of the design

### User Interface

The user interface is made in the Blynk mobile application and includes all the relevant information that the user would want to see. This includes two gauges which show the temperature the sensor is detecting directly in–front of it, as well as the ambient temperature that it is detecting. This also includes a graph that plots the direct temperature data onto a line graph with every update that is sent, this can be used to view the variance in temperatures over the last 15 minutes, 1 hour, 6 hours and 1 day. The object temperature gauge and ambient temperature gauge are updated once every second, whereas the LED lights and terminal widgets will be updated once every 4 seconds.

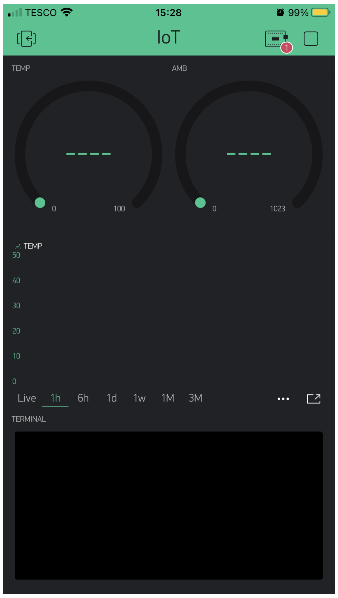


**Figure 3.3:** Design of the User Interface

# Chapter 4

# Discussion

## 4.1 Summary

The system functions by connecting the NodeMCU ESP8266 to a users mobile device through the service Blynk. The NodeMCU and the mobile device connect to the same Wi–Fi network in order to contact each other. The Blynk app connects to the NodeMCU with the authentication code which was generated when creating the Blynk project.

Once the NodeMCU is connected to a power supply, the NodeMCU will gather measurements from the MLX90614 and run them through checks to determine the tyres temperature and what state the tyre is in. The results will be sent to the mobile device which will display them in multiple ways. The precise temperature value will be displayed on a gauge, the tyre state will be displayed through a terminal, and a graph will show the deviation in tyre temperatures over time.

The NodeMCU also connects to 3 LEDs that correlate to the tyres state, where too cold activates the blue LED etc. The lights are added in this project, as the user will not be able to look at their phone whilst driving, so the LEDs can provide simple to understand output. The mobile phone app can gather the information and present it to the user afterwards, or be used to view live values by someone that is not driving.

## 4.2 Benefits

Benefits of this system:

- The application includes graphs that show the temperature data over a long period of time, allowing the driver to analyse their tyres temperature throughout a session.
- The small size of the infrared sensor means that it can be placed easily within view of the tyre whilst the vehicle is moving. Such as in the wheel arches or with a small mounting point.
- The accuracy of the temperature sensor provides good sensor readings back to the user

## 4.3 Limitations

The limitations and downfalls of this system:

- This system would need a Wi–Fi connection in order to see the precise tyre temperature data, meaning that on a remote country road without Wi–Fi, the system cannot be used.
- This implemented system only has one temperature sensor, meaning that only one wheels temperature will be measured at a time.

## 4.4 Solutions and Suggestions

Solutions to the limitations:

- A LCD screen could easily be attached directly to the NodeMCU, allowing the user to see the exact temperature data of the tyre without a

Wi–Fi connection. Although this could have been implemented, I did not buy an LCD to include in this system.

- Another solution to the lack of a Wi–Fi connection, could be the use of Bluetooth by connecting the NodeMCU to an external Bluetooth chip as the NodeMCU does not have the built in functionality of Bluetooth.
- Multiple MLX90614 sensors can be connected to the NodeMCU which would allow the system to measure the temperature of multiple tyres. This again is simple to implement, but I did not buy two of the sensors.

## 4.5 Future Work

The future work of the project would build upon the limitations and add extra functionality to give the user better analysis of their tyre conditions. This would include the addition of an LCD screen which the user would be able to view on their dashboard, and would include the live precise temperature data as well as the temperature window which was already implemented.

The output provided by the LED could also be mounted in a way that is easy for the driver to view, in the same way Formula One steering wheels have to show flags in their track sector. In terms of more functionality, this could provide the user with the rate of tyre degradation, allowing multiple people to connect to the NodeMCU as Blynk supports this etc.

# Bibliography


*Blynk documentation*. (n.d). http://docs.blynk.cc/#intro-how-blynk-works

*Dht11 sensor and its working*. (n.d). https://www.elprocus.com/a-brief-on-dht11-sensor/

International Telecommunication Union. (2012). *Overview of internet of things*. http://handle.itu.int/11.1002/1000/11559

McLaren. (2018). *The brain of an f1 car*. https://www.mclaren.com/applied/blog/brain-of-an-f1-car-mclaren-ecu/#:~:text=With%20help%20from%20over%20300,over%20750%20million%20data%20points.