

Dynamic Car Information

Project Engineering

Year 4

Sean O’Shaughnessy

Bachelor of Engineering (Honours) in Software and Electronic Engineering

Galway-Mayo Institute of Technology

2020/2021

Project Graphic (Optional)

**Declaration**

This project is presented in partial fulfilment of the requirements for the degree of Bachelor of Engineering (Honours) in Software and Electronic Engineering at Galway-Mayo Institute of Technology.

This project is my own work, except where otherwise accredited. Where the work of others has been used or incorporated during this project, this is acknowledged and referenced.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgements**

I would like to thank Niall O’Keefe, Michelle Lynch, Paul Lennon, and Brian O’Shea for their help with my project.

They answered any questions I had in detail and helped me with different types of challenges I faced throughout the year completing my project.

**Table of Contents**

[1 Summary 6](#_Toc69472124)

[2 Poster 7](#_Toc69472125)

[3 Introduction 8](#_Toc69472126)

[3.1 Project Goals: 8](#_Toc69472127)

[3.2 Project Motivation: 8](#_Toc69472128)

[3.3 Overview 8](#_Toc69472129)

[4 Background 9](#_Toc69472130)

[5 Project Architecture 10](#_Toc69472131)

[6 Project Plan 11](#_Toc69472132)

[7 Can Bus 12](#_Toc69472133)

[7.1 Subheading 12](#_Toc69472134)

[8 Amazon Web Services (AWS) IOT Core 14](#_Toc69472135)

[9 On-Board-Diagnostics (OBD) 15](#_Toc69472136)

[10 Free RTOS 16](#_Toc69472137)

[11 Conclusion 17](#_Toc69472138)

[12 References 18](#_Toc69472139)

[13 Appendix 19](#_Toc69472140)

# Summary

**Ever wanted to monitor important information from your car without being in it?**

Important information will be gathered from a car’s Electronic Control Unit (ECU) to allow the user to monitor their car from the comfort of the driver’s seat or their home. This is more aimed to performance cars that have turbo charged engines as the tolerance for error is little to none. It would also be a very useful tool for fleet owners such as, taxis or trucks as the owner can easily monitor each vehicle without being present and they can see if there is any foul play with their vehicles. Oil Pressure, oil temperature and water temperature are some of the important aspects that would need to be monitored in a car.

The OBD (On-Board Diagnostic) port would be used to access the ECU and using the correct protocol and Can bus to communicate with the ECU, the data from the vehicle can be monitored in real-time. I would then use the cloud to store the data that is retrieved from the ECU of the car as this would allow the user to access it from anywhere that you have internet connection. An android application will allow the user to view the live data that is recorded.

This complete package it very easy to install into a vehicle and the company can use a portable router that the device can connect to so it can be always online and sending the valuable data to the owner. Foul play to vehicles can cost fleet owners thousands, as well it gives peace of mind to owners of performance cars as they can easily monitor their vehicle.

# Poster

# Introduction

## Project Goals:

* To create a portable, device that can make owning your car/cars safer by monitoring valuable data.
* To help car enthusiasts monitor their vehicle, allowing the oil to properly warm up whilst keep an eye on boost levels and water temperatures.
* Help fleet managers to monitor their vehicles and prevent or catch any foul play to the vehicles by members of staff.
* Ease of use so it can be a plug and play device with minimal setup needed and once internet connection is available it keeps posting data to the cloud.

## Project Motivation:

My motivation for my project is that I have a great interest in the automotive industry and this project can help both car enthusiasts fleet owners monitor their vehicles. In Europe there are more than 35 million commercial vehicles and buses whilst there are over 245 million cars in the European union alone.

As I am a car owner and a car enthusiast, I believe that a project like this can help many people and vehicle owners as well it can highlight areas where preventative maintenance is needed. It can highlight damages that are caused to vehicles in fleet by foul play by staff by over revving the engine or not staying in the optimal fuel zone to save fuel and costing the company more.

These problems got me thinking into how I could create a simple way to monitor the vehicle without being near the vehicle.

## Overview

My final year project report contains all the work that I have completed throughout the year and it explains the main areas of the project in detail which are Can Bus, Amazon Web Services, On-Bored Diagnostics and Free RTOS.

# Background

Automotive Can Bus

Cloud (AWS & SQL)

You should change the title of this section to suit your own project subject. The aim of this section is to introduce to the reader any relevant background information that is required for your project.

You may have multiple ‘background’ sections. Think of any of the questions you had to answer during the research phases of your project – these likely should be addressed in a section like this.

# Project Architecture

Figure ‑ Architecture Diagram

For my final year project, I am using a NXP Freedom K64-F development board and a Ozan Electronic On-Board Diagnostic (OBD) Simulator that is used to simulate a vehicles Electronic Control Unit. Using Can Bus as the form of communication, I need to use a transceiver to drive and detect data on the bus, for this task I am using the Mikroe ATA6563 transceiver. To enable the Freedom K64-F to connect to the internet I am using the Mikroe WIFI 10 Click board which allows me to use Amazon Web Services (AWS).

# Project Plan

# Can Bus

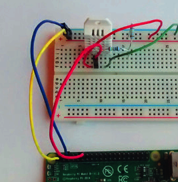
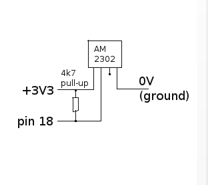
This is an example heading for a section in a project. You choose your sections to suit your project.

## Subheading

This is a subheading, use subheadings to break up a large topic into smaller sections.

Notes on content:

Photographs are not technical diagrams and are not a good substitute for professional technical diagrams. Use photographs to enhance a report, but not as a replacement for diagrams.



V

Figure ‑ A photograph is not a replacement for a circuit diagram

In describing software, you need diagrams and/or summaries of software design & layout. It is not sufficient to just paste some code. You should describe what your code is designed to do, in English. If you decided to put your code in functions or libraries or objects, describe this architecture. One good layout is to include a snippet(s) of code alongside an explanation. You do not have to explain every part of your code, pick the important parts.

Write out any mathematical equations or calculations that are important in your project and explain them.

Include details of any major problems or challenges you encountered in an area, and how you solved them.

IEEE referencing style is recommended a the default style to choose, however if you are familiar with a different referencing style then you can use that.

When you need to reference add a citation in the relevant sentence, usually at the end, before the full stop [1]. Then have this numbered citation referenced in the list of references at the end of the document.

# Amazon Web Services (AWS) IOT Core

# On-Board-Diagnostics (OBD)

# Free RTOS

# Conclusion

Write a short conclusion. What is the outcome of the project? Perhaps you have a product prototype, or some results, or a demonstratable system.

Do not use your conclusion to tell the reader what you might have done if you had more time, but keep it focussed on what you actually have done. You can mention future opportunities for further development of the work, but keep this part short.

# References

[1] Digilent. *Basys 3* Reference [Online]. Available: <https://reference.digilentinc.com/basys3/refmanual>

[2] P. J. Ashenden. *Digital Design (Verilog): An Embedded Systems Approach Using Verilog*. Burlington: Morgan Kaufmann, 2007.

[3] M. Lynch, “Combinational and Sequential Logic with Verilog & Xilinx Vivado” [Online], 2020. Available: <https://learnonline.gmit.ie>.

[4] M. Lynch, “FPGA Stopwatch Project Source Code” [Online], 2020. Available: <https://learnonline.gmit.ie>.

# Appendix