

Digital Flight Data Recorder

Project Engineering

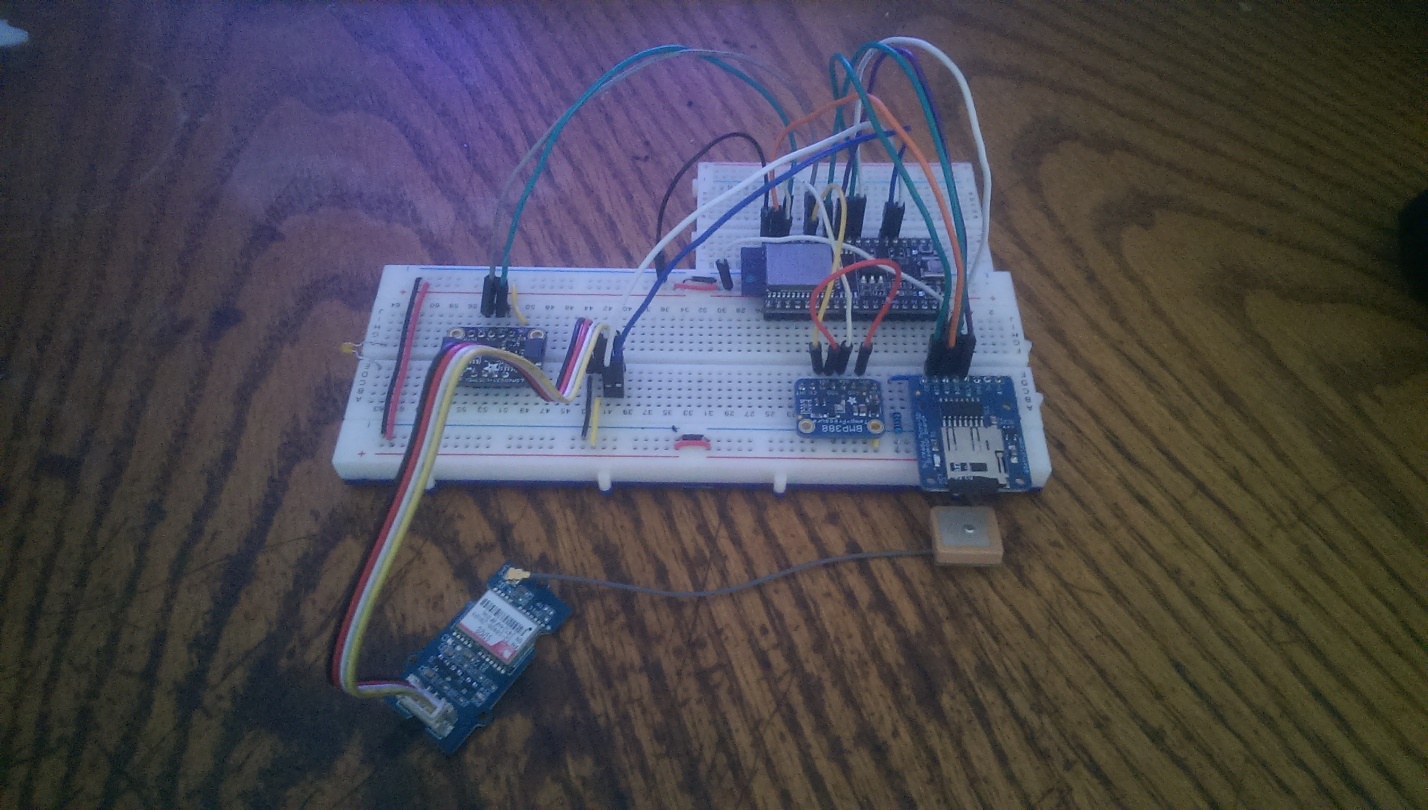
Year 4

Rokas Cesiunas

Bachelor of Engineering (Honours) in Software and Electronic Engineering

Galway-Mayo Institute of Technology

2020/2021



**Project Hardware Components**

Figure 0‑ Diagram of Components

**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.

\_\_\_\_\_\_\_\_\_\_Rokas Cesiunas\_\_\_\_\_\_\_\_\_\_\_\_

**Table of Contents**

[1 Summary 5](#_Toc71982326)

[2 Poster 6](#_Toc71982327)

[3 Introduction 7](#_Toc71982328)

[4 Flight recorder background 8](#_Toc71982329)

[5 Project Architecture 9](#_Toc71982330)

[6 Project Plan 10](#_Toc71982331)

[7 Sensors 11](#_Toc71982332)

[7.1 Pressure Sensor 11](#_Toc71982333)

[7.2 Magnetometer 13](#_Toc71982334)

[7.3 Accelerometer and Gyroscope 13](#_Toc71982335)

[7.4 GPS receiver 13](#_Toc71982336)

[7.5 Notes on Content 14](#_Toc71982337)

[8 Ethics 16](#_Toc71982338)

[9 Conclusion 17](#_Toc71982339)

[10 Appendix 18](#_Toc71982340)

[10.1 Calibration Coefficient Parsing 18](#_Toc71982341)

[10.2 Convert register values to calibration coefficients 18](#_Toc71982342)

[11 References 19](#_Toc71982343)

# Summary

This project takes inspiration from the black boxes that are legally required on all commercial flights and are also present on most private airplanes. The purpose of this black box is to record various sensors onboard the airplane and to store this data for investigation in the event of an accident or crash.

The goal of this project was to take this idea of a data recorder and reduce its size so that it may be used on remote control aircraft or drones. By having flight data and logs you will be able to look back on past flights to measure performance or to see what went wrong in the case of a malfunction or crash.

To do this, the sensors used were a Pressure sensor to find altitude, GPS for location and speed readings, Accelerometer, Magnetometer, Gyroscope for acceleration, orientation and rotation measurements. The main board used was an ESP-32, it handled the I2C communication and reading of these sensors. The code written in C/C++ and the web page in JavaScript.

At the end of this project, the built in ESP-32 Wi-Fi was used to able to send the data from the sensors to a local webpage. This webpage then redirected the data to a database. When you wanted to view the data, it would read the database and display the data using graphs.

# Poster

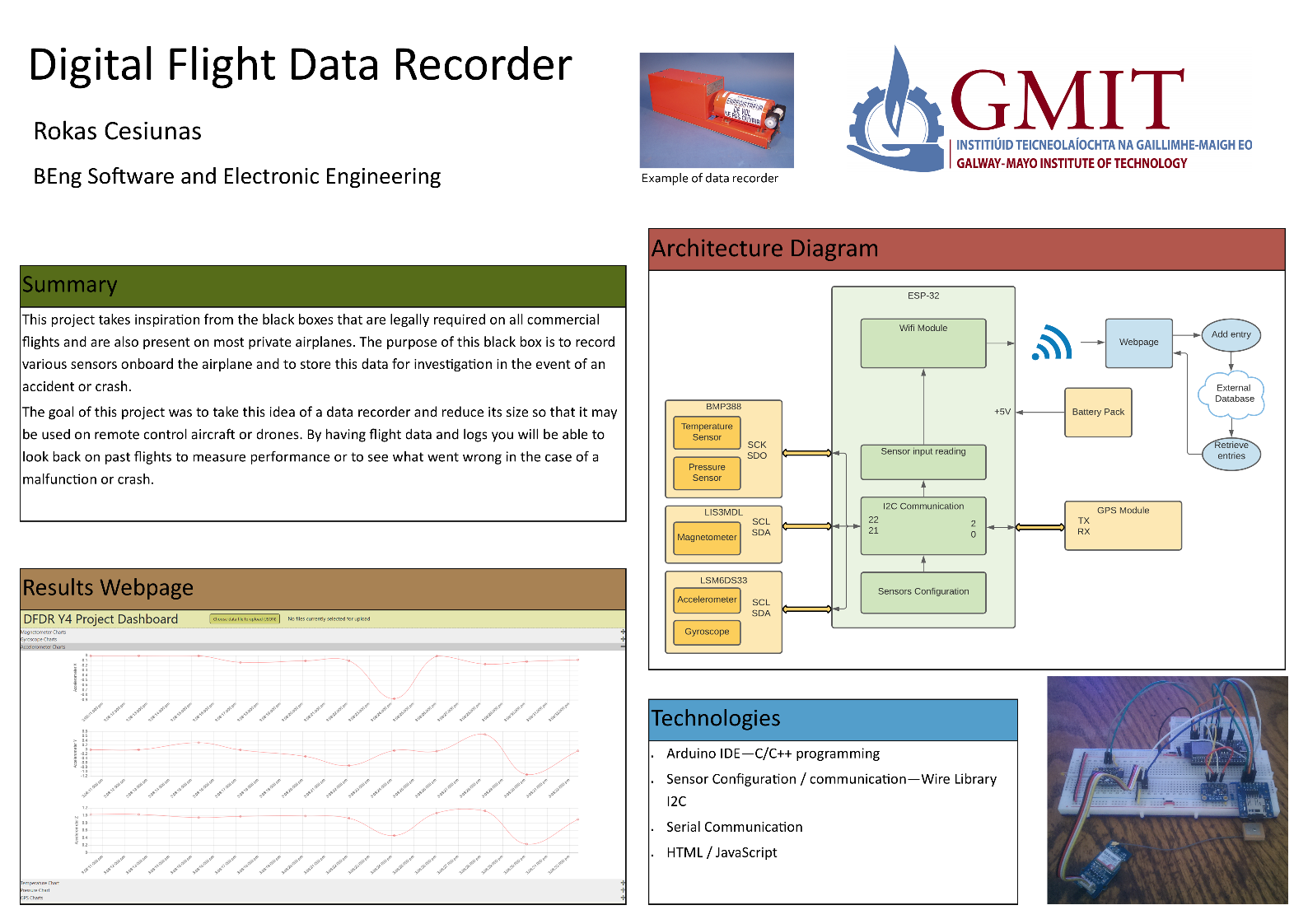


Figure 2‑ Project Poster

# Introduction

This project was intended to be used on small remote-control aircraft or drones as a means of recording flight data which could be used to find the cause of a crash if it were to occur. The data recorded, such as location, speed, orientation, altitude, and rotation would give the operator information about what is happening onboard the flight. This data would allow you to go back, inspect all the parameters and performance of the aircraft. This would be useful for many reasons such as, building prototype drones to ensure that they meet specifications. Debugging flight systems or finding out caused a malfunction.

It is all built upon a main programmable board (ESP-32) with the sensors connecting to it. The sensors used were an accelerometer and gyroscope (LSM6DS33) which record the acceleration and rotation respectively, magnetometer (LIS3MDL) which can tell us magnetic direction, pressure sensor (BMP388) for altitude measurement, and GPS sensor (SIM28). The data is sent to a webpage via the built in Wi-Fi on the ESP-32 board.

# Flight recorder background

A flight data recorder is a device that is used to record significant flight parameters, a cockpit voice recorder that records all sounds inside the cockpit would also be integrated to make up a flight recorder. These recorders are built into a secure box to ensure that the recordings can survive strong impacts 3400g and extreme temperatures over 1000°C as required by EUROCAE ED-112.

These devices have been made mandatory since 1965 on all commercial airplanes, multiengine airplanes, airplanes having 10 or more passenger seats as well as helicopters. This device has to record 88 different parameters during flight, such as roll attitude, heading, and time of each radio transmission.

The reason for making the recorders mandatory is to help investigators in the case of an accident. The data recorded can provide invaluable information about what happened or what went wrong. This in turn can help to improve safety by learning from mistakes, improving existing aircraft and developing new safety features.

# Project Architecture

Diagram

Description automatically generated

Figure 5‑1 Architecture Diagram

# Project Plan

Graphical user interface

Description automatically generated

Figure 6‑ Project Plan

# Sensors

The sensors I used on the project are pressure sensor, magnetometer, accelerometer and gyroscope, and a GPS receiver. They are used to locate the device in 3d space.

SIM28

## Pressure Sensor



The BMP388 is a digital pressure sensor, it senses the ambient atmospheric and can give a reading back in Pascals. The BMP388 also has its own temperature sensor that is used to compensate the pressure readings and for later conversion to altitude in meters by finding the local mean sea level pressure.

To get readings correct readings from the BMP388, you need to read the calibration coefficients stored on the chip. These are programmed in the factory during calibration of the chip.

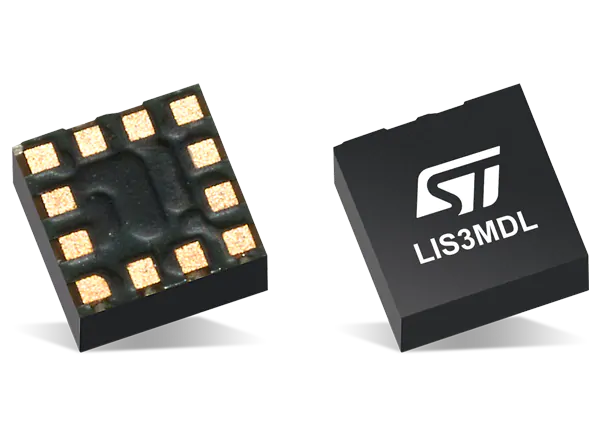


Some of these register values are 16 bits so it is necessary to add them together, and also save some as signed integers. Datasheet tells us which are 16 bit and or signed. After reading the coefficients into an array, the datasheet also gives us formula to convert these numbers into the calibration coefficients. (appendix 10.1 and 10.2 for parsing and calculation of coefficients)



To get the correct pressure out of the sensor, we read the register that stores this data, which is 3 bytes. Add those together and then send it to be compensated. The compensate pressure function takes the uncompensated pressure value and multiplies with the calibration coefficients and the compensated temperature value. The exact formulas are stated in the datasheet.

## Magnetometer



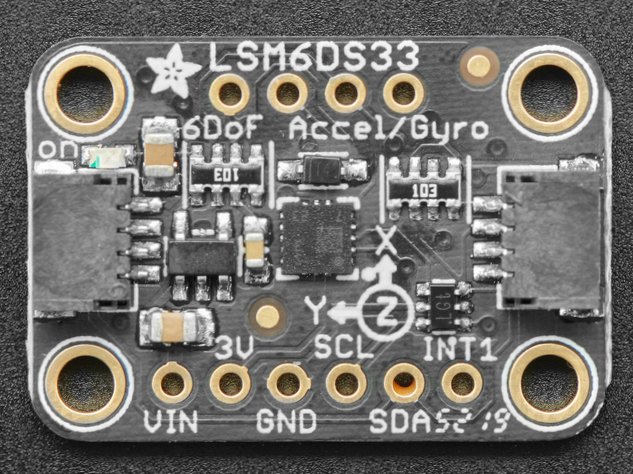
The LIS3MDL is a three-axis magnetic sensor, it measure the magnitude of magnetism or geomagnetism. This sensor returns each axis as 2 bytes values in gauss. The use of this could be as a magnetometer or compass, it can tell us which direction the drone is facing. Using this sensor is very easy, 

First, we need to send all the configuration parameters that we want to the chip, the last configuration also enables the chip to start taking continuous measurements.



First, we need to check our status register if its clear, and then read our mag data into the array. To get the correct reading, the data needs to be divided by the sensitivity chosen when configuring the sensor.

## Accelerometer and Gyroscope



The LSM6DS33 is a dual sensor chip, it has an accelerometer and a gyroscope. The accelerometer is used for measure acceleration of the drone, but the measurement is converted so that it is in relation to gravity (g)



## GPS receiver

A picture containing text, electronics

Description automatically generated

The SIM28

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

IEEE referencing style is recommended the default style to choose for citations and referencing, 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. Then have this numbered citation referenced in the list of references at the end of the document.

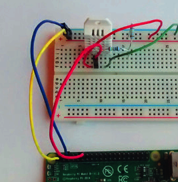
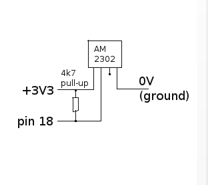
Here I might write something about something, e.g. image processing, and I need to cite my sources, like this [1]. Here I have used MS Word’s ‘Insert Citation’ feature, with IEEE style selected, to create that number inside brackets. Here’s another citation [2]. Word increments the number automatically. I can fill in the details about my reference now or later. I can then go the end of the document and create a page of references automatically. See the demonstration in class on this (also recorded via Teams). Here I am adding another citation [3]. And another [4].

You then need to insert a References section at the end of the document. In Word, choose References->Bibliography->References. This will pull all your citations into a reference page, as shown at the end of this document. The References section in this document also includes examples of further references that have not yet been cited in the text – to demonstrate IEEE style for different types of resources, i.e. books/websites/lectures/source code/etc.

You could also manually add all your citations & references, without using MS Word’s citation & referencing wizards.

## 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 6‑2 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.

# Ethics

Include a short section on ethical considerations in your project or in the field of study of your project.

# 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.

# Appendix

## Calibration Coefficient Parsing



We need to store some of the register values in the array as 16 bits, so we shift the upper 8 bits and or the lower 8 to them.

## Convert register values to calibration coefficients



Now we need to convert the calibration coefficients into floating point numbers

# References

|  |  |
| --- | --- |
| [1] | H. Kinsley, "Reinforcement Learning," PythonProgramming, [Online]. Available: https://pythonprogramming.net/q-learning-reinforcement-learning-python-tutorial/. [Accessed 02 02 2021]. |
| [2] | [Online]. |
| [3] | MakeSigns, "Scientic Posters Tutorial," [Online]. Available: https://www.makesigns.com/tutorials/scientific-poster-parts.aspx. [Accessed 09 02 2021]. |
| [4] | Arduino. [Online]. Available: https://www.arduino.cc/. [Accessed 09 02 2021]. |
|  |  |
|  |  |

[x] J. C. Russ and F. Brent Neal. *The Image Processing Handbook*. 7th ed. Boca Raton FL: CRC Press, 2017.

[x] S. Lippman, J. Lajoie and B.E. Moo. “Classes” in *C++ Primer*. 5th ed. Massachusetts: Addison Wesley, 2013.

[x] IEEE Signal Processing Society. “Signal Processing for 5G”, *YouTube*, 2019. [Online]. Available: <https://www.youtube.com/watch?v=uca6X4Ykcmg&list=PLcZOnmyqlalacL9YqkhyufLQGIW_C78Os&index=6>. Accessed: Feb 2, 2021.

[x] Digilent. “Basys 3 Reference Manual”, *Digilent Reference*. [Online]. Available: <https://reference.digilentinc.com/basys3/refmanual>. Accessed: Feb 2, 2021.

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

[x] M. Lynch. “Discrete Fourier Transform (DFT)”, Lecture, Digital Signal Processing, Galway-Mayo Institute of Technology, Galway, 2020.

[x] MRC Centre for Global infectious Disease Analysis. (2021). *Covid-Sim*. [Source Code]. Available: <https://github.com/mrc-ide/covid-sim>. Accessed: Feb 2, 2021.

[x] OpenCV. “Face Detection Using Haar Cascades”, *OpenCV-Python Tutorials*. [Online]. Available: <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html#face-detection>. Accessed: Feb 2, 2021.