California State Polytechnic University, Pomona

Aerospace Engineering Department

ARO 2311L-1, Fall 2018

Electronic Flight Instrument System (EFIS) Build

EFIS Final Report

By: Eliot Khachi

To: Professor Tarek Elsharhawy

Date: December 6, 2018

Table of Contents

1. **Objective** – Design and assemble an EFIS and implement a potentiometer controlling a DC motor using an Arduino Mega.
2. **Outline of Procedure**
    a. Plan Scope
    b. Implement Block Diagram
    c. Design Schematic
    d. Order Parts
    e. Assemble
    f. Develop and Test Software
3. **Summary and Results**

The primary aim of the EFIS project was to display basic flight information from sensory input data. A microcontroller must be used to manage this data. The load cell and potentiometer were integrated in addition to this primary aim, to be implemented as a part of the EFIS.

a. Arduino IDE source code of all sensors individually compile, run, and read accurate data, and further calibration on the few sensors was successfully implemented.
b. TFT LCD successfully displays Heading, Pressure, Altitude, Speed, Artificial Horizon, Latitude/Longitude, % Power, and Weight.

Appendix A – EFIS Project Scope

The object of this project is to build an Arduino-based EFIS, an Electronic Flight Instrument System. This device is meant to display real-time flight data, in order to provide the pilot with the needed information during flight. This project emphasizes independent learning, as well as the gained knowledge learned in class. The functions, that have been determined by the team to be featured in the device, are listed below, as well as a timeline of when each task is to be completed. This list of functions is subject to be slightly modified, depending on factors such as, cost, complexity, or the availability of parts.

Features:

- ASI (Airspeed Indicator)
- HDG (Heading)
- AH (Artificial Horizon)
- Temperature
- Altitude
- Latitude/Longitude
- DC Motor % Power
- Weight
- Pressure

Semester Outline (Subject to be modified):

1. Week 3 - Scope (Completed)

2. Week 5 - Components board power, sensors display (finalize which ones)
then the **Overall System Design**

3. Week 6 - Draft of schematics

6. Week 10 - Begin Assembly

7. Week 11 - Finish Assembly

8. Week 12 - Testing of the EFIS

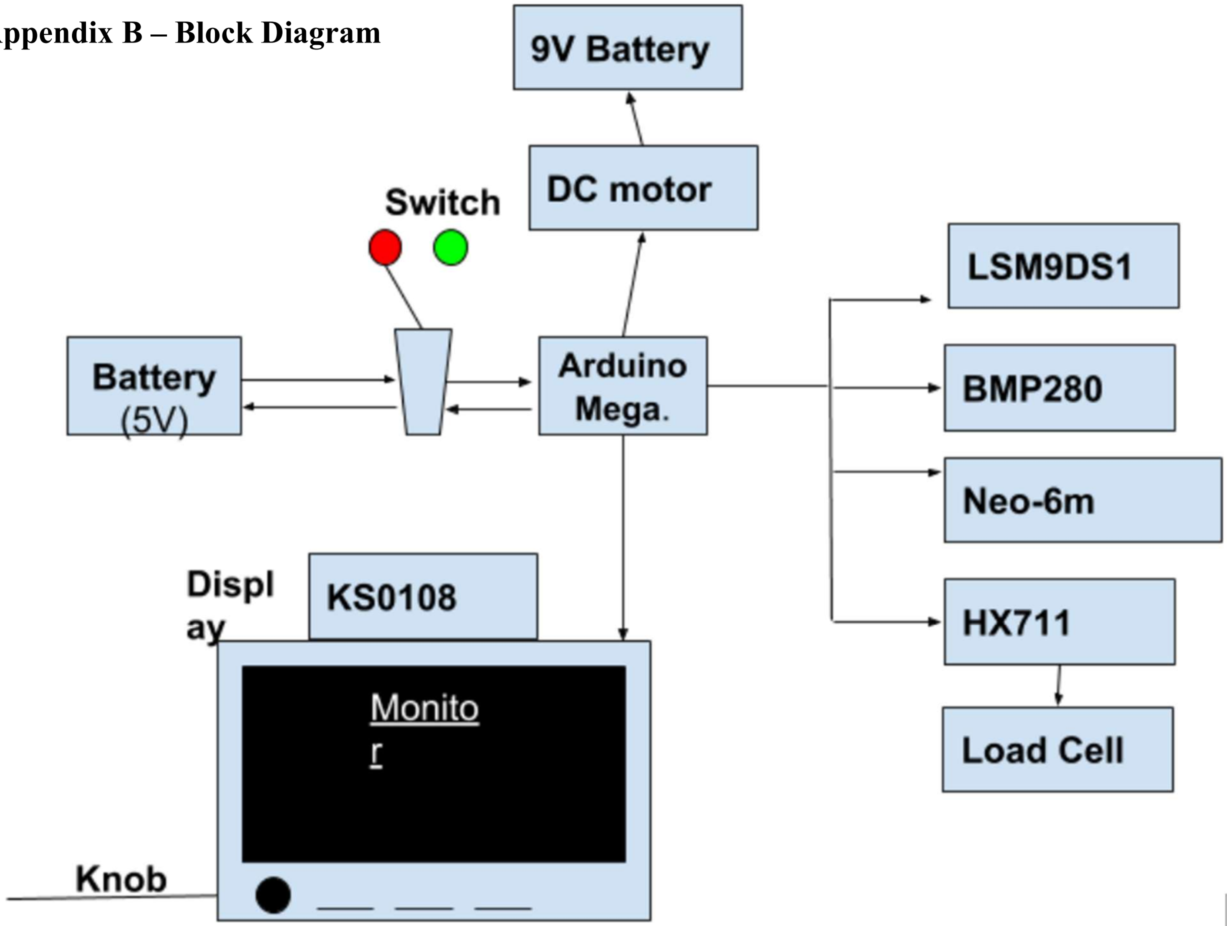
4. Week 7 - **Final Schematics, Hardware, & Software.**

5. Week 8 - Have everything purchased

9. Week 13 - Alter design, fix and test again, if needed.

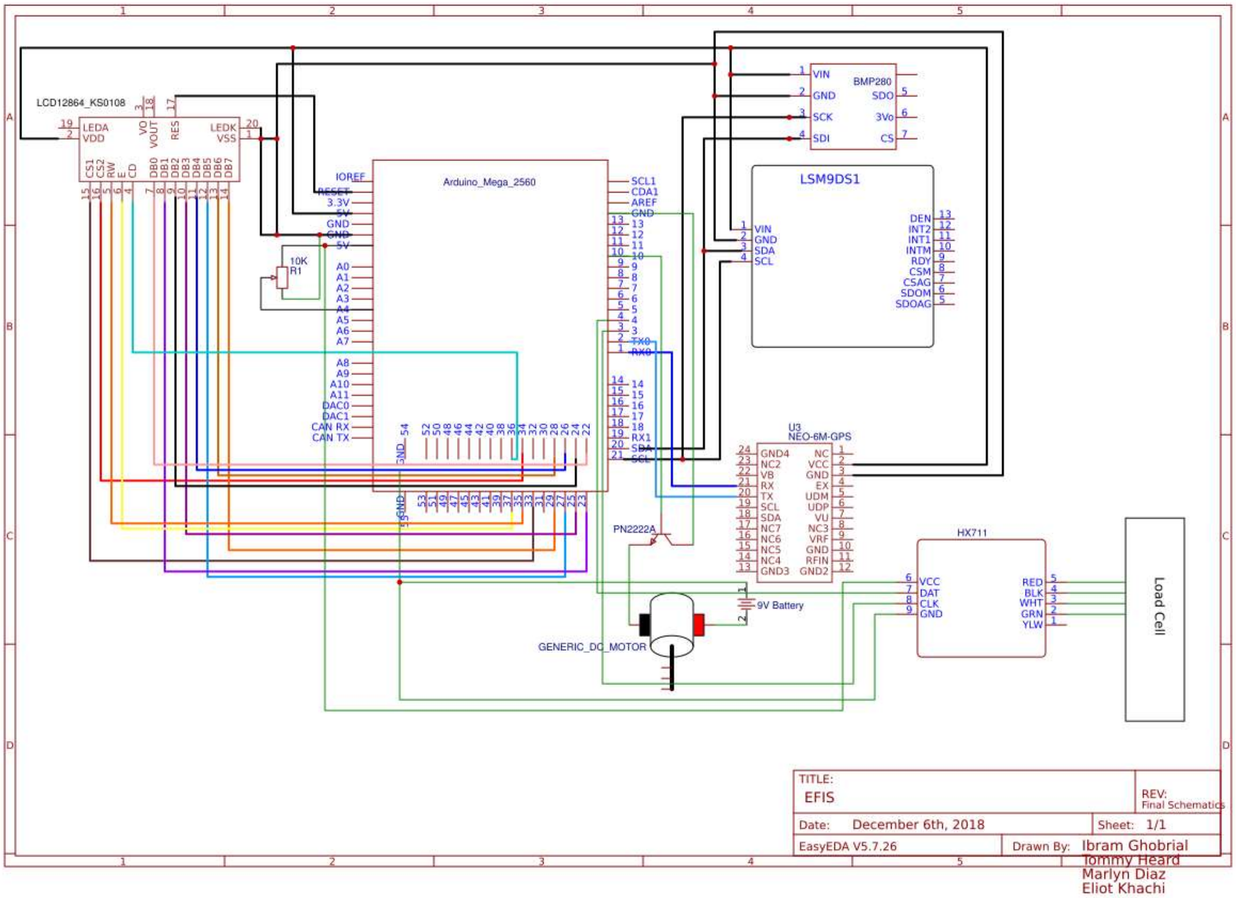
10. Week 15 - **Final Demo**

Appendix B – Block Diagram



The sensors of the EFIS along with the potentiometer and load cell serve as the analog inputs while the display and DC motor serve as the digital outputs. The LSM9DS1 is a 9-DOF IMU whose sense data will be used to obtain the Artificial Horizon. The BMP280 is an environmental sensor that reads and displays ambient temperature and pressure, which will be used to display altitude as well. The NEO-6M GPS is used to display latitude, longitude, and speed. The Potentiometer controls the DC Motor's speed. The HX711 signal amplifier and load cell are used to display weight. The recommended voltage intake by the Arduino Mega2560 is 7-12V, so a 9V battery is used. An additional 5V battery is used incase the DC Motor requires more power.

Appendix C – Schematic



In the schematic the LSM and BMP are connected in parallel utilizing the SDA/SCL communication line to the Arduino, which is standard I2C format. The NEO-6M utilizes the TX/RX pins while the Load Cell/Signal Amplifier and Potentiometer/DC motor each have their own communication line on separate input pins. The KS0108 display is powered by and connected to the digital output pins of the Arduino. The Arduino and DC Motor are powered by a 9V battery.

Appendix D – Order Parts

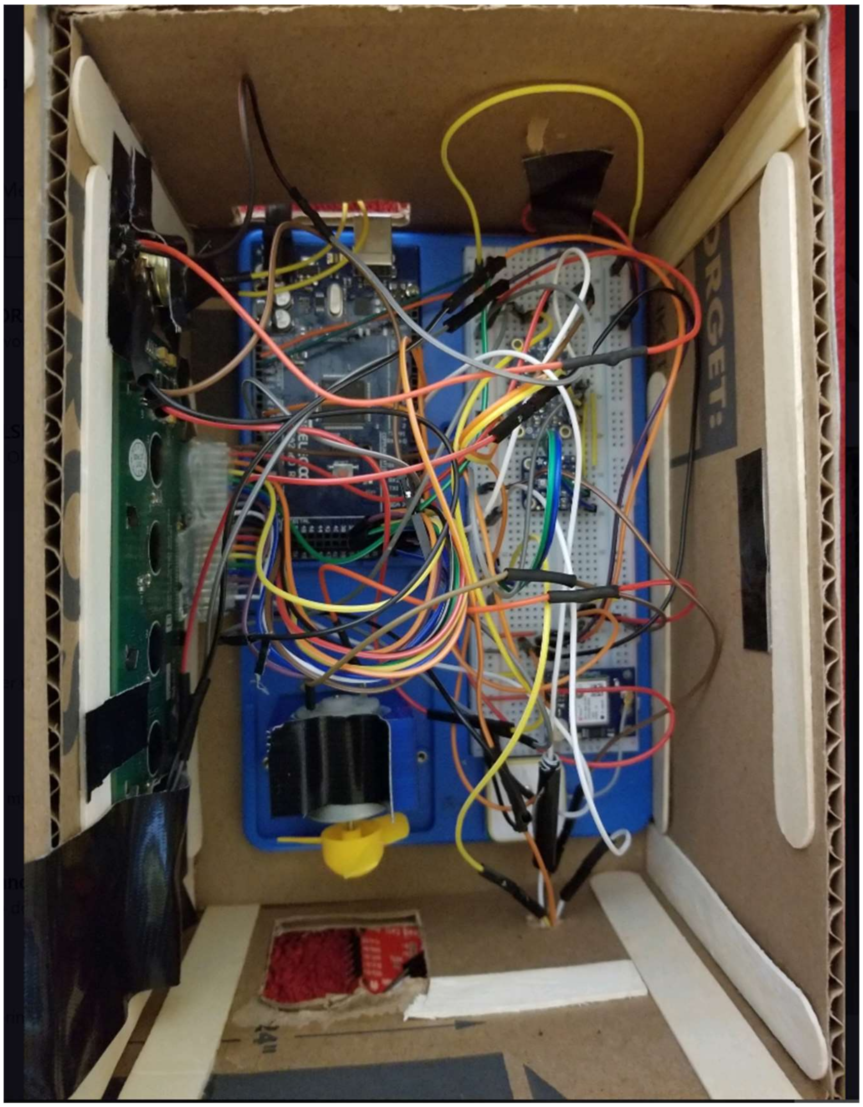
1. Arduino Mega 2560 ($15) – The Arduino Mega 2560 is used instead of the Arduino Uno to ensure that its specs meet the requirements for this EFIS project. It has 54 digital I/O pins, 16 Analog input pins, 256 kB of memory, and a clock speed of 16MHz.
2. LSM9DS1 ($15.50)– 9 DOF (Degree of Freedom) IMU (Inertial Measurement Unit) contains a 3-axis accelerometer, 3-axis magnetometer, and 3-axis gyroscope. The accelerometer acts as a G-meter, measuring forces acted upon the aircraft, and is used to measure airspeed and vertical speed. The magnetometer senses the direction of the strongest magnetic force, usually due to Earth's magnetic field. The gyro measures aircraft's spin/twist forces and gives orientation.
3. BMP280 ($15.50)– Pressure/Temperature sensor gives readings of pressure with $\pm 1\text{hPa}$ accuracy and of temperature with ± 1℃ accuracy. The pressure readings are used to give altitude.
4. NEO-6M ($15)– GPS module gives latitudinal/longitudinal positions along with time. Could also give velocity and altitude if necessary.
5. Potentiometer ($9) – Knob used to control DC Motor speed.
6. DC Motor ($6) – Rotary electrical machine that converts electrical energy into mechanical energy. Its power intake and therefore speed is regulated by the potentiometer. Powers the fan on the EFIS.
7. KS0108 Display ($20) – This TFT display requires raw communication with the Arduino using the TFT libraries on the Arduino IDE. No driver is present. Code is manually written and calibrated.

8. Load Cell and Amplifier ($34) – Measures weight using a Wheatstone Bridge wire configuration and a signal amplifier.
9. Breadboard ($5) – Houses all components and wirings.
10. Jumper Cables and Switches ($6) – Cables serve as wiring between the Arduino, the components, and the breadboard, and the switches are used to monitor potentiometer signals to the DC motor.
11. Light ($10) – LED lights denote starboard and port side.

Total Cost: $148

Appendix E – Assembly

The final assembly of the EFIS consists of the Arduino Mega2560, KS0108 TFT display, LSM9DS1 IMU, BMP280, NEO-6M GPS, Load Cell, HX711 Signal Amplifier, potentiometer and knob, a 9V and 5V battery, and of course, the breadboard and jumper wires to connect them together. Header pins for each sensor were soldered, and after connecting them to the breadboard, jumper wires connect all sensors in parallel to the Arduino in I2C connection. The display is connected using the Arduino's Digital output pins. Everything is mounted on a 3D-printed plastic insulated board, and the full assembly is held inside a cardboard container for the sake of weather-proofing and presentation.



Appendix F – Develop and Test Software (Technical Contribution)

All implemented source code used on the final Arduino IDE sketch were obtained online; minimal modifications were manually written onto the sketch. First, the example codes were found for each sensor individually, with each sensor having its own Arduino IDE sketch being test ran one at a time. After test running all components to print their raw data outputs, source code was found online to convert that raw data into displayable data using the TFT library on the Arduino IDE. Table F shows what function each component's raw data serves in the EFIS.

Table F – Component Functions

Sensor/Component	**Raw Data**	**Function/Displayed Data**
BMP280	Pressure	Temperature, Pressure, Altitude (*C°, atm*, *m)*
LSM9DS1	Acceleration, Gyroscope, Magnetometer	Heading, Artificial Horizon (*degrees, graphic visual)*
NEO - 6M	Latitude, Longitude, Time	Latitude, Longitude, Speed (*degrees, mph)*
Potentiometer	Voltage Control	Controls DC Motor Power Intake (*%)*
Load Cell	Voltage Difference	Weight (*lb)*

After each component’s code was individually tested to be correctly read and outputted by the Arduino, all code was integrated under the final IDE sketch. This was done by calling each component’s setup functions and loop functions in the void setup() and void loop() of the final sketch. I was assigned to obtain functional example code for the BMP280, and integrate that code along with all the other components, whose example code was found by the rest of my team. Although I did not produce the final version of the sketch that was used in the project, my initial draft was used and modified by my team.

My main technical contribution was the assessment of the Nextion TFT display we were originally intending to use to display our data. I worked on implementing the code for communications between the Arduino and Nextion. After getting so far with displaying the % power intake by the DC Motor, Latitude/Longitude, speed, pressure, and altitude, I ran into a complication regarding display of the Artificial Horizon, and concluded that the TFT LCD display would be a better approach to solving that problem. Luckily most of the code for communications with the Nextion was easily transferable to the LCD display. The complications of that transfer, as well as further calibration of the code, were primarily handled by my team members.

Appendix G – Summary and Results

The primary aim of the EFIS project was to display basic flight information from sensory input data. More specfically, the displayed data is: Temperature (*C*°), Pressure (*atm*), Altitude (*m)*, Heading (*degrees*), Artificial Horizon (*visual)*, Latitude/Longitude (*degrees)*, Speed (*mph)*, % Power *(%)*, and Weight *(lb)*. The Arduino Mega2560 micrcontroller was used to manage this data. The load cell and DC Motor were integrated in addition to this primary aim, to be implemented as a part of the EFIS, as such topics were covered over the course of this semester. The HX711 Signal Amplifier makes readable the data from the load cell, and the Potentiometer controls the voltage and therefore power running through the DC Motor. Everything is displayed on the KS0108 TFT display as intended, although there remains a small delay when it comes to displaying the speed.

a. Arduino IDE source code of all sensors individually compile, run, and read accurate data, and further calibration on the few sensors was successfully implemented.
b. TFT LCD successfully displays Temperature, Pressure, Altitude, Heading, Artificial Horizon, Latitude/Longitude, Speed, % Power, and Weight.

References

1.0 - "A BASIC INTRODUCTION TO NEXTION ARDUINO LIBRARY." *Nextion*, 29 Oct. 2017, nextion.itead.cc/2017/09/15/basic-introduction-nextion-arduino-library/.

2.0 - Adafruit Industries. "Adafruit BMP280 I2C or SPI Barometric Pressure & Altitude Sensor." *Adafruit Industries Blog RSS*, www.adafruit.com/product/2651.

3.0 - Adafruit Industries. "Adafruit 9-DOF Accel/Mag/Gyro+Temp Breakout Board – LSM9DS1." *Adafruit Industries Blog RSS*, www.adafruit.com/product/3387.

4.0 - "NEO-6 Series." *u-Blox*, 12 July 2018, www.u-blox.com/en/product/neo-6-series.

5.0 - "Nextion Display with Arduino - Getting Started." *Random Nerd Tutorials*, randomnerdtutorials.com/nextion-display-with-arduino-getting-started/.

6.0 - "How to Interface GPS Module (NEO-6m) with Arduino." *Arduino Project Hub*, create.arduino.cc/projecthub/ruchir1674/how-to-interface-gps-module-neo-6m-with arduino-8f90ad.

7.0 - Krishnakumar, Surya, and Instructables. "Arduino DC Motor Speed Control Potentiometer." *Instructables.com*, Instructables, 8 Oct. 2017, www.instructables.com/id/Arduino-DC-motor-speed-control-potentiometer/.

8.0 – *Voltage Dividers*, learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup guide/all.