## Design4 Supporting Software

Please use the following link to download the supporting software.

LINK

## Teensy Program

TeensyComSD is the program that creates a UOM\_Atmospheric\_Data Mavlink packet , sends it to the ground control station and log it on an SD card onboard the aircraft. The program works as follows:

1. Continuously read the CO sensor values simultaneously
2. Read Mavlink packets sent from the Autopilot to the GCS
3. Send the read packet to the GCS through the telemetry
4. Check what the type of the message is and extract the data required from it
5. Every certain interval of time combine all the latest readings of the required data for atmospheric sensing with the current reading of the CO sensor and fuse them into one Mavlink packet called UOM\_Atmospheric\_Data.
6. Write the data that went into the UOM\_Atmospheric\_Data to the SD card.
7. Read Mavlink packets sent from the GCS and pass it on to the Autopilot

The program keeps repeating these steps as long as the board is powered.

In order to modify, compile and upload the Teensy program provided you need to install Arduino, Teensyduino and the modified mavlink libraries.

Install Arduino from:

<https://www.arduino.cc/en/Main/Software>

Install Teensyduino from: (Make sure to run the installer as administrator)

<https://www.pjrc.com/teensy/td_download.html>

Copy *mavlink* folder provided to Arduino libraries in “C:\Users\SomeStudent\Documents\Arduino”

## New Mission Planner

Since we added a new Mavlink message to the library, the original GCS cannot recognize the new message. *UoM\_Mission\_Planner* contains a modified version of the GCS that can receive the new message and log it in the Tlog. If you already have the old GCS you can directly run the modified version by executing *UoM\_Mission\_Planner\MissionPlanner.exe (Application)*. Otherwise run the driver installations *UoM\_Mission\_Planner\Drivers\ DPInstx64.exe (Application)*  in case you are using a 64-bit machine. Run *UoM\_Mission\_Planner\Drivers\ DPInstx86.exe (Application)*  in case you are using a 64-bit machine. This should install all the required drivers for running the GCS properly.

## Development using MAVProxy

To process the received sensor data in real-time on the ground you can use MAVProxy. It allows you to broadcast the incoming Mavlink stream from the serial port of the telemetry module over the network. You can use it to generate two streams one going to the Mission Planner over UDP and another UDP stream going into your real-time processing application.

Install MAVProxy development environment using the steps in the link below:

Before compiling MAVProxy copy the content of the folder *ForMAVProxy\v10* provided into “C:\Python27\Lib\site-packages\pymavlink\dialects\v10” this contains the new modified and compiled Mavlink library.

<http://dronecode.github.io/MAVProxy/html/development/mavdevenv.html>

## Hardware installation

The below diagram show the connections between all the components of the system.

