Universal Ground Station Receiver

For Hubsan X4

Assembly/Build Guide & Documentation

Embedded Systems Laboratory EN.525.743.91

Kyle Mercer – 2/26/2018

**RESERVED FOR TOC**

# System Description

The Universal Ground Station Receiver is a headless embedded system which will bind with compatible Bluetooth devices, listen to incoming Bluetooth control packets, and re-broadcast control packets in a format that the Hubsan X4 (H107L) Quadcopter can understand. The ultimate goal is for the system to be able to accept virtually any Bluetooth slave device as a transmitter, enabling a broad range of standard compliant devices to potentially become a quadcopter controller with minimal software setup time. Examples of devices which could communicate with the ground station include (but are not limited to):

* A Bluetooth enabled personal computer
* A smartphone
* A Bluetooth handheld classic controller (Corrie’s project)
* A glove controller with Bluetooth adapter (Corrie’s project)

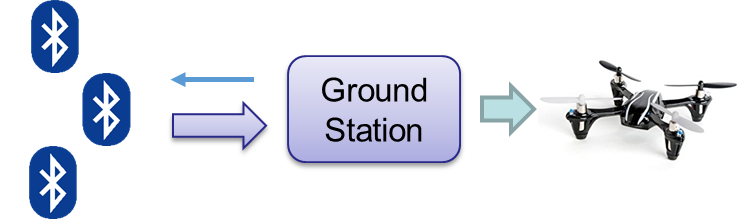


Figure : Top level control flow of system (Hubsan X4 pictured on right)

## Capabilities

The Ground Station Receiver supports the following capabilities:

* Perform command translation and relay communication for Quadcopter controls
* Any Bluetooth enabled device may pair
* Supports a Quadcopter over Bluetooth Universal Protocol (QoBUP) communication scheme
  + QoBUP has an accompanying developer ICD which is included in Appendix A
* Supports controlling the Hubsan Quadcopter’s onboard LEDs via a push button on the ground station

## Limitations

The Ground Station Receiver has the following limitations:

* Only the Bluetooth protocol is supported between transmitter and Ground Station
* Single Quadcopter brand supported (Hubsan X4 H107L)
* Only one quadcopter may be commanded by one transmitter
* Only the A7105 RF chip is supported for the Quadcopter RF communication

# Functional Description

## Overall System

This document describes only one half of the entire functional system. In order to fully realize the capability of the Universal Ground Station Receiver two different types of controllers have been implemented by another member of the class, Corrie Russell. She is the project partner and her half of the system is required for the full system functionality. See the documents pertaining to Corrie’s controllers for specific implementation details on the transmitter side. Each controller designed will adhere to the QoBUP standard. Each transmitter will leverage the QoBUP to send controls to the ground station.

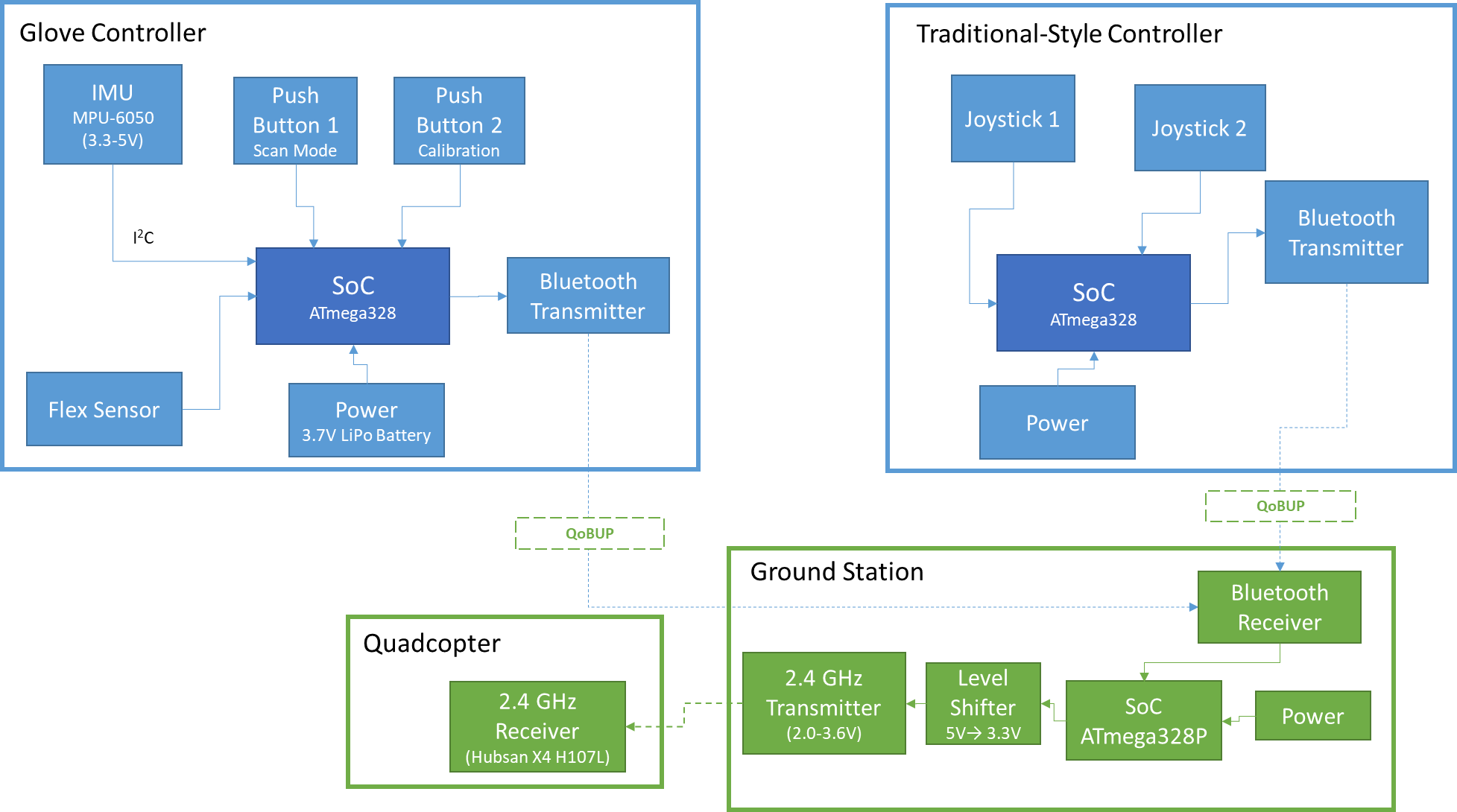


Figure : Overall system block diagram.

## Ground Station System

Upon reception of controls from the Bluetooth enabled transmitter the Microcontroller Unit (MCU) reads the contents of the incoming message over a standard UART interface which connects to the Bluetooth module receiver. As per the QoBUP ICD, a status message is transmitted back to the transmitter for each message received. Refer to the Quad over Bluetooth Universal Protocol (QoBUP) Developer ICD for specific details on about the QoBUP.

### MCU Software Operation

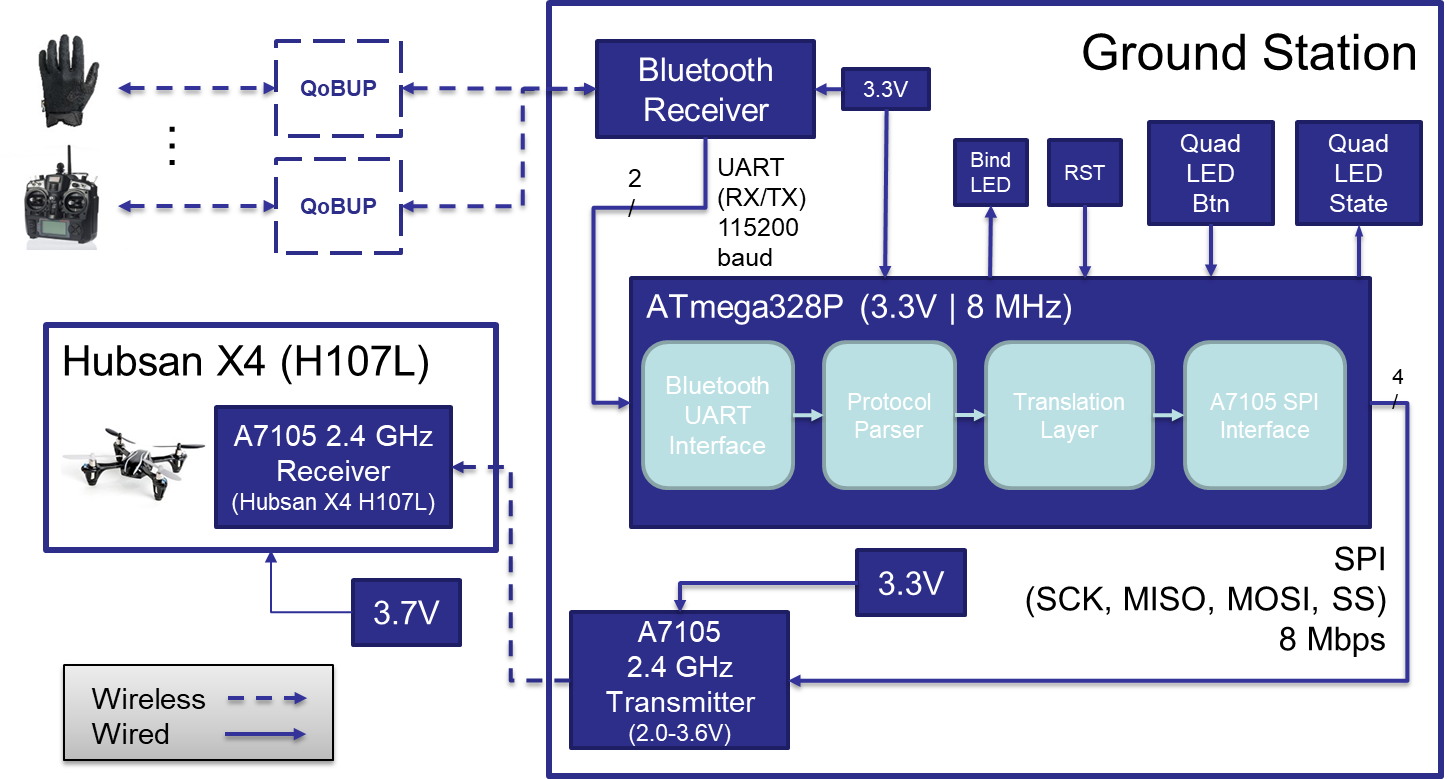
The MCU reads in control messages through the Bluetooth UART Interface into a buffer where it will get picked up by the Protocol Parsing Interface. At this stage, validation is done on the message to ensure the message adheres to the QoBUP ICD. Validation is complete once all sub-blocks are parsed out into common software structures and command processing is initiated via the Translation Layer. During this stage of software processing, the allocated structures are picked up by a translation stack which is written specifically for the target quadcopter, in this case, the Hubsan X4. The structures are then translated into a set of primitive register read/write commands which are then queued up into the A7105 SPI Interface. Controls are then dispatched to the A7105 over the SPI bus at a controlled rate dictated by the Hubsan X4 communication spec. The flight control update rate for the Hubsan X4 is 10ms (Hung, 2015). All modules including the peripheral complements and MCU operate at a common 3.3V. No level shifting is required. The overall ground station hardware/software block diagram can be seen in Figure 3.

Figure : Overall ground station hardware/software block diagram.

Full Doxygen has been generated for the project’s codebase and is included in the appendix section. The Doxygen iscommensurate with the above software flow description. It also provides an API framework if a developer wishes to implement an interface for another quadcopter.

### MCU Software Startup

During system initialization, the ground station will attempt to automatically bind to the Hubsan X4. The handshaking involved to complete the binding process is quite involved; however, the entire procedure has been decoded into a step-by-step process outlined on the internet (Hung, Reverse Engineering a Hubsan X4 Quadcopter, 2015).

An additional aspect of system startup will include configuration and initialization of the Bluetooth transceiver module. The Bluetooth module connected to the ground station will act as a Bluetooth master whereas all transmitters that wish to pair will act as the slave device. All communication (both configuration and controls) to and from the Bluetooth module will occur over UART connection set to an 115200 baud rate.

# Materials & Resources

## Hardware

The list of necessary hardware for the completed ground station is scoped out as follows:

* Arduino Pro Mini ATmega328P AVR MCU – 3.3V @ 8MHz
* FTDI Basic Breakout – 3.3V Programmer
* COTS Hubsan X4 (H107L) Quadcopter
* 3.7V LiPo Batteries
* 9V Alkaline Batteries
* 9V Battery Clip Connectors
* Resistors: ~39Ω x1 and ~68Ω x1
* LEDs: Green x1 and Yellow x1
* Prototyping/breadboard
* Insulated prototyping wires
* XL7105-D03B (A7105 Wireless RF 2.4GHz Transceiver Module with Integrated Power Amplifier)
* BlueSmirf Silver Bluetooth Modem (RN-42)
* Slide Switch
* 2mm to 2.54mm pitch header adapter (optional)

The ground station will operate standalone with a 9V battery powering the entire system, including the MCU and peripherals.

## Software

The following list of development and operational software will be leveraged for the project:

* Linux (development environment)
* Arduino CLI toolchain for Linux
  + <https://www.arduino.cc/en/Main/Software>
* Git
* Ground Station Source Code
  + <https://github.com/thesocrdu/emb_sys_lab>

# Implementation Guide

The following sections breakdown the steps necessary to rebuild and deploy software for the Universal Ground Station Controller. The prototype initially designed is provided as a reference below.

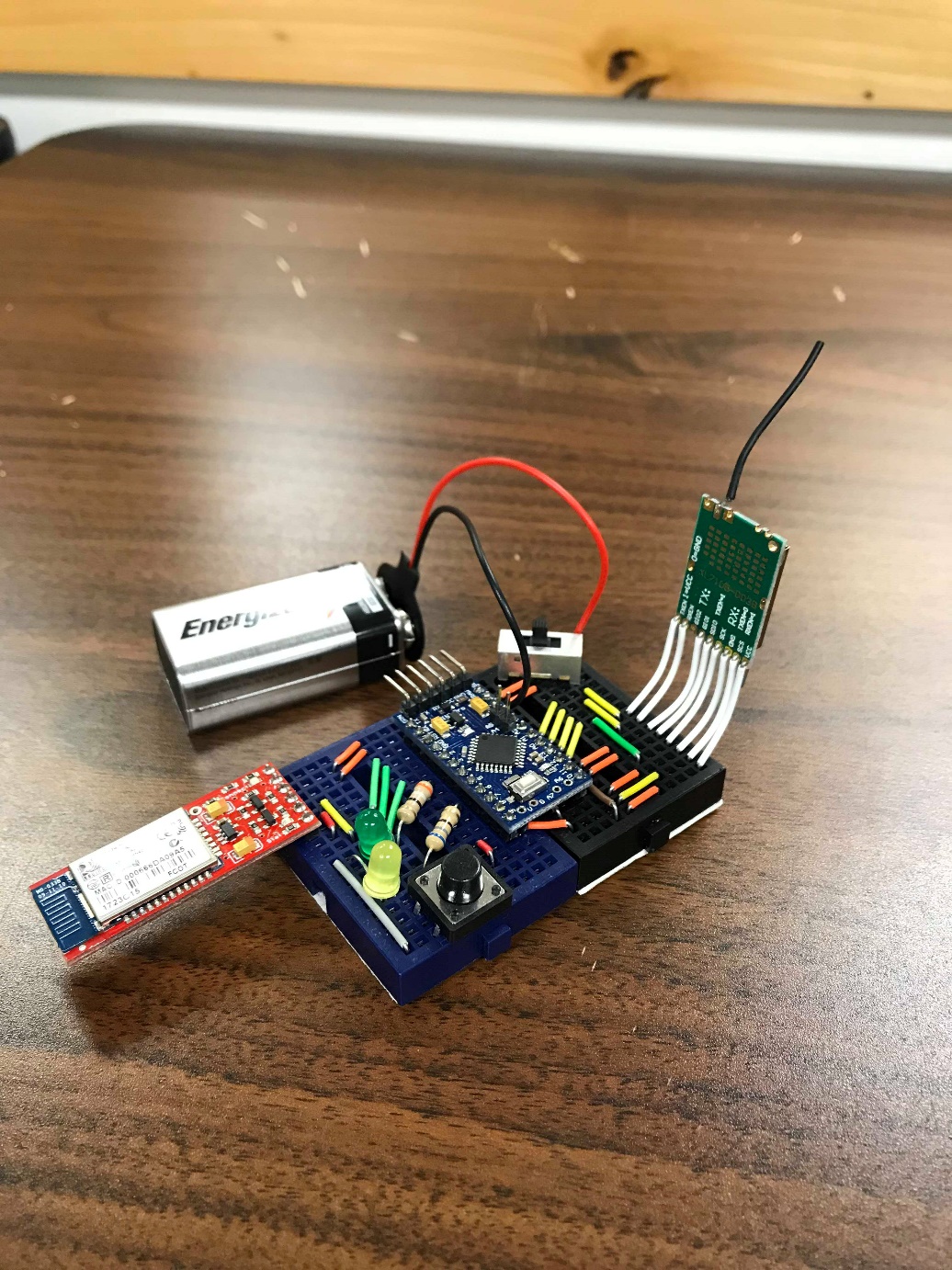


Figure : Prototype of Universal Ground Station Controller - fully assembled

## Build Environment Setup

### Obtaining the Project Codebase

Open a terminal window and type the following to clone the project repository from git hub

$ git clone <https://github.com/thesocrdu/emb_sys_lab.git>

Then run the following commands

$ cd emb\_sys\_lab/projects/ground\_station

$ pwd

Take note of the directory path output, you will need it in the next subsection.

### The Arduino CLI

To install the Arduino software perform the following steps:

1. Go to <https://www.arduino.cc/en/Main/Software>
2. Select your platform and download the Arduino IDE
   * Note: this project targets Linux IDE version 1.8.5
   * Your mileage may vary if you use a different version or platform
3. Unpack the Arduino files to your home directory
   * Note: the build steps below assume you extract the folder to your home directory
4. Run the *“arduino”*executable inside of the folder
   * The Arduino IDE window should come up
5. Go to File 🡪 Preferences
6. In the “Sketchbook location:” box enter the path from the $pwd command you ran in the previous subsection.
7. Click “OK” and close the Arduino IDE

### Building the Project Codebase

Back in your terminal window, run the ls command. You will see a “*build.sh*” file. This is a build script that may be used for building and flashing the Arduino Pro Mini’s firmware. You can test compile the main controller software by issuing the following command:

$ ./build.sh –v –b arduino:avr:pro:cpu=8MHzatmega328 gs\_async\_main/gs\_async\_main.cpp

You may see some warnings display on the screen. This is normal.

## Hardware Assembly

### System Schematic

### Assembly Considerations

Both the Bluetooth module and A7105 RF module emit RF output at 2.4GHz. As such, it was determined that keeping the modules as separated as possible resulted in the most reliable communication between the Bluetooth transmitter and the A7105’s RF communication with the quadcopter.

## Flashing Software to Ground Station

Flashing the Arduino Pro Mini requires the following steps once the hardware is assembled:

1. Connect the FTDI programmer to the programming breakout pins of the Arduino Pro Mini
2. Plug the other end of the programmer into the PC’s USB port.
   * The Arduino’s LEDs should turn on. If they don’t you may have the programmer attached in the wrong orientation.
3. Run the following command from the terminal window inside of the *ground\_station* directory:

$ ./build.sh –p /dev/ttyUSB0 –b arduino:avr:pro:cpu=8MHzatmega328 \ gs\_async\_main/gs\_async\_main.cpp

Once the process is complete your Arduino should now be programmed.

## Testing the Ground Station

If the Ground Station is currently being powered (either through the USB or through a 9V battery) then remove the power so the device turns off. Perform the following steps:

1. Use one of the 3.7V LiPo batteries to apply power to the Hubsan Quadcopter
   * The blue LEDs on the Hubsan should blink back and forth while waiting to an initial bind request
2. Re-apply power to the Arduino Pro Mini and it should begin executing the binding process
   * You have successfully bound with the Hubsan if the Hubsan lights stop blinking and the green LED on the Ground Station is now lit.
3. The Ground Station is now waiting for an initial Bluetooth pair request to come through
4. Take your preferred Bluetooth transmitter (eg. the glove controller) and initiate the pairing mode from that device (see Corrie’s setup guide for how to do this).
   * Pairing is complete once the BlueSmirf Bluetooth module shows a solid green LED
   * You may now use the transmitter to control the Hubsan Quadcopter

# References

The following references are critical to the success of ground station receiver.

* **Hubsan X4 reversed protocol specification:** This text document gives a breakdown of the necessary setup and binding steps required to connect to the Hubsan X4 with an A7105 RF chip. This will be used as a guide when writing the A7105 SPI interface software.
  + <http://www.jimhung.co.uk/wp-content/uploads/2014/11/HubsanX4_ProtocolSpec_v1.txt>
  + <http://www.jimhung.co.uk/?p=1349>
* **Deviation10 firmware (targets 32-bit ARM):** The Deviation10 framework is an open source, aftermarket firmware written specifically for the DEVO brand controllers (ARM-based). The controller must be hardware modded and loaded with the firmware prior to use. If loaded, the firmware essentially provides controls for a wide variety of quadcopters and small UAV planes. Since it provides support for the Hubsan quadcopters, this could be a helpful resource for reference on establishing communication with the Hubsan X4, despite the difference in target MCU architectures.
  + <https://bitbucket.org/PhracturedBlue/>
* **Bluetooth modem RN-42 AT command set:** This source provides documentation for interfacing with the Bluetooth module.
  + <https://cdn.sparkfun.com/datasheets/Wireless/Bluetooth/bluetooth_cr_UG-v1.0r.pdf>
* **Atmel’s AVR documentation:** This reference provides the necessary documentation about the MCU being using for the ground station.
  + <http://www.microchip.com/wwwproducts/en/ATmega328p>