1 Bonus Challenge

The bonus challenge requires the team to incorporate a 2.4 GHz XBEE Pro radio module (50mW to 63mW, one mile range) to establish uplink/downlink communications prior to and during the flight. The team decided to attempt all three portions of the bonus challenge, and their respective implementation methods are described below.

1.1 Bonus Challenge A: Command Uplink

The team will use a custom Arduino serial interface to setup the rocket. After the vehicle is vertical on the Launchpad, an operator will enter an access code to begin rocket configuration, and then select the flight profile, either flight 1 or flight 2. If flight 1 is selected, the vehicle will enter standby mode, a state where acceleration is used to determine liftoff. The vehicle will then follow the guidelines for flight 1. If flight 2 is selected, the operator will then be prompted to enter roll commands in “angle, direction, time” format, until all commands are entered. The rocket will then enter standby mode in the same manner as flight 1. At any point in the above mentioned process, a unique code may be entered to revert all settings. An image of the sample interface is shown below in Figure 1.

```
ILLINOIS SPACE SOCIETY 2018 SPACE GRANT GROUND CONTROL
------------------------------------------------------
Please enter launch code to access command interface.

Access granted.

Welcome. Please enter flight number (1 or 2).

Flight profile 2 selected. Please enter commands.
( Target Angle, Roll Direction, Hold Time )

0
1
1

Command 1
Target: 0    Direction: 1    Hold: 1

270
-1
1

Command 2
Target: 270    Direction: -1    Hold: 1

90
...
```

Figure 1: XBEE Sample Interface

Data from the 9-DoF, including vehicle orientation, angular velocity, and angular acceleration, will be transmitted via the XBEE communication system to ground in the previously detailed sequence. They will also be recorded on an onboard SD card. Data will be sent as integers at 0.05s intervals, with an empty line separating data between intervals and the security code before each line. Flap controls will also be transmitted and recorded to analyze results after the launch.

1.2 Bonus Challenge C: Crosstalk

An independent portion of code will be activated after liftoff to continuously check for data uplink. If a string with correct dimensions is detected (n-digit security code + 5-digit calculations), the string will be further broken down for analysis and processing. Results would then be sent back in string format.

1.3 Network Setting

Data transmission is achieved by two XBEE-Pro radio modules. One radio module is on the rocket, and the other one is connected to a computer through an Arduino on the ground. The onboard Arduino will collect data from the magnetometer and send it down to the computer on the ground through the radio module. Communication between the two radio modules is setup by manipulating the settings of the radio module. Usually, a network needs a coordinator to work. In this case, however, XBEE Pro can accomplish peer to peer communication in a network without a coordinator. Thus, the two XBEE Pro radio modules are all set to be endpoint rather than a coordinator and a router. In addition, two XBEE pro radio modules should have the same PAN (personal area network) ID to pair up. Basically, the XBEE Pro module will first broadcast its information to find modules in the same network. The team set the PAN ID to 3591 in an effort to prevent having the same PAN ID as other teams. This number avoids many intuitive patterns such as the one in 5555(a repetitive set of numbers), 9909(someone's birth date), etc., that overall makes certain ID's more likely to appear. Moreover, the PAN ID can be reset at the moment of the launch if the same ID as another team through the XCTU software, a GUI used to set up the radio, provided by digit.

1.4 Configuration

Both the XBEE onboard and on the ground are incorporated with an Arduino. The XBEE on the ground is connected to the Arduino through a few wires: a wire that connects Rx on XBEE to Tx on Arduino, a wire that connects Tx on XBEE to Rx on Arduino, a wire that connects Ground on Arduino to Ground on XBEE, and a wire that connects 3.3V power on the Arduino to 3.3V power on the XBEE. The Arduino onboard is connected to the XBEE shield, as to avoid wires falling off during the flight and cluttering the avionics bay. The Arduino on the ground reads and prints the signal sent from the onboard Arduino. The onboard Arduino reads and writes the signal sent from the ground to adjust the maneuver of the rocket. This system grants the ability to reset the maneuver direction of the rocket before launch and receive data collected from the magnetometer during the flight.

1.5 Communication

To maximize data transmission efficiency, the team avoided blocking XBEE signal as much as possible. According to the XBEE handbook, the most efficient way to set up the network is to leave the antenna of the XBEE outside of the rocket body. The only way to do this would be to drill a hole on the side of the rocket, which may cause unwanted turbulence. Since the rocket body is blue tube and not metal, the airframe will have a negligible effect on the transmission of the radios. A picture of the XBEE shield module and antenna is pictured below in Figure 2.

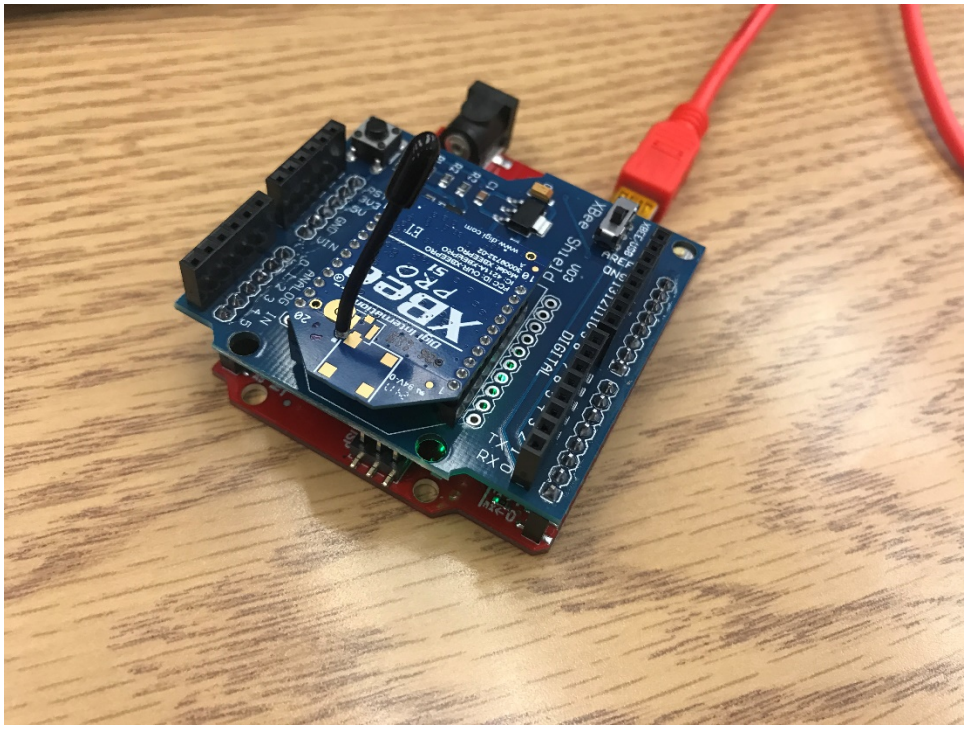

Figure 2: XBEE Shield Module and Antenna