**Ground Station Tele-command module for Tafiti Nano-satellite**

**By Gerald Mbuthia**

The ground station will receive and transmit data from the ground to the Nano-satellite. The sent data will include commands to the satellite requesting various information such as an onboard system report or system updates and maintenance files while the data received will include responses from the various commands which may include sensor readings, configuration settings etc.

The ground station should have the capacity to be small and minimal and therefore the design will edge towards building modules to be interfaced to a PC or Laptop which will process the data and provide the application interface to interact with the Nano-satellite.

**Block Diagram of the proposed Ground-station module**

**How it will work**

The Ground station will comprise of two systems, the Payload receiver and the Telemetry, Tracking and Telecommand Transceiver. The payload receiver will serve to receive image data sent from the high data rate S-band Endurosat Transmitter on the Nano-satellite. The sampling rate of the receiver should be greater than the 20Mbps data rate of the transmitter. The Endurosat Transmitter also modulates its data using QPSK, 8-PSK, 16-APSK. A software defined radio that meets the above requirements would suffice as a receiver.

The Telemetry, tracking and Telecommand Transceiver system will allow bidirectional communication between the User and the Nano-satellite. A half-duplex communication would suffice for this particular use case and the data rates determined by system requirements. Input and output interaction will be via the PC. Connection between this module will be via USB. Communication between the PC and this ground module will be via serial communication.

**Serial communication between STM32 microcontrollers and the PC**

1. **Pinout**

STM32 boards have various communication protocols and interface that support this. For serial communication, we can use 2 interfaces, either UART or USB. Using the STM32CubeIDE we can define our peripherals to either of the above interfaces.

1. **Connection Options**

Since we require a USB interface for our computer, the following are ways we could achieve this on our custom built module.

1. Software defined USB virtual Com Port.

Software can be written that implements a virtual COM port over USB. Example projects are available from ST’s site to easen the development processes but it is so far the most time consuming section. A direct connection of the STM32’s USB pins to the USB connector would suffice. Link to the example is: <https://www.st.com/en/embedded-software/stsw-stm32121.html> .

A USB device library by ST is also available here: <https://www.st.com/content/st_com/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32-standard-peripheral-library-expansion/stsw-stm32121.html>

1. USB to UART bridge.

A chip may be purchased to provide conversion from UART to USB for termination via a USB connector. It involves an extra chip but is an easier option to get working. Connection would be the MCU’s UART pins to the RX/TX pins on the chip and then wire the USB pair on the FTDI chip to the USB connector.

1. USB to TTL UART Cable.

**3. Data Input and Output on PC**

To receive and send data and messages via Serial communication, various options exist according to the Operating system used. Terminal emulator applications are a go to with Putty being one option for windows and ckermit an option for linux and mac.

In our case, Putty is a viable optionsince it is open source and under active development. Initial setup needs to be implemented. These includes:

1. Setting up the comp port.
2. Setting up the connection type.
3. Setting the baud rate.

Putty also provides a really handy way of storing session and log data. Simply click on Session > Logging. Set the location and file name to save the log to and save the settings. All log data and files will be saved to the defined file.

To view the resulting log files or process them, accessing the file generated via notepad or any other means capable of reading log data may be used and processing may be done through this file access means.

**References**

<https://electronics.stackexchange.com/questions/436803/how-to-design-an-stm32-board-with-uart-over-a-usb-connection>

<https://www.st.com/en/embedded-software/stsw-stm32121.html>

<https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT230X.pdf>

<https://www.ftdichip.com/Products/Cables/USBTTLSerial.htm>.