***TXServiceCode User Guide***

*How to use:* The TXServiceCode is designed to be able to simplify the transmission process to the radio receiver. It takes a single file (txFile.txt), this file contains the information need to transmit, and the information that the user desires to transmit. The file follows the following format.

First line: The transition window in milliseconds. EX 900000 (15 minutes)

All subsequent lines: 000000001 (time stamp, 10 chars) : (divider flag) Data. EX 000000001: 123,123,123

Notes:

1. DO NOT put an extra new line at the end of the file.
2. The divider is not transmitted
3. If the divider happens to be contained in the data it will still be transmitted.
4. There are certain data types that do not transmit the time stamp and the divider.

To call the TXService code it is required that you navigate to the piTXISR folder, the full path is CubeWorks/TXISR/piTXISR. Once in the folder the code is called by calling the they binary executable, these files most commonly end in .out or .run, although this is not strictly required. Usually lunix will highlight this files in a different color. The data type also need to follow the excitable name. The format is as follows ./<executable name> <space> <data type> EX *./TXSerice.run 1*

Notes:

1. There are 5 data types. (0 - 4)
2. If the code is passed data type 3 it does not transmit the time stamp. This is meant to be used for pictures. If multiply data types need to have this behavior this may easy be changed in the TXSerciveCode.c If the number would like to be changed that can also be done easaly by changing the corresponding #define in the first lines of the code. I

*Workings of the TXSeriveCode.c:*  As a forestated the object of this code is to simplify the transmission process. Therefore the process is as follows,

* + - Step one: The serial port is opened, and the files are verified as being good files, if either of these steps fail the code terminates with an abnormal termination flag.
    - Step two, the code waits exactly 5 seconds for the transmission window.
      * Note: This means that the code should be called exactly 5 seconds before the transmission window.
      * Note: The code uses delta t this means that the code does not need to know what the system time is. It also is not important that the system time is correct so long as delta t is still correct.
    - Step three: the radio is put into pip mode
    - Step four the transmission
      * The code takes in one line of thtxFile.txt at a times. It reads the time stamp and then transmit the data.
      * The code then saves the time stamp to the flagsFile.txt.
      * The code waits 120 milliseconds.
      * This process is repeated until all the data is sent or the transmission window expires.

Notes:

1. The code will NOT watch the packet size, if one line of data is more then 128, it will still transmit the whole line.
2. The code saves the time stand after every transmission. If the code should fail the last send transmission will always be saved.

*Compiling the code:* GCC is used to compile the TXServiceCode. There are two files that should be created each time the code is compiled, 1 a.out and 2 TXService.run. To create this files the process is as follows.

* + 1. Navigate to the piTXISR folder.
    2. Enter gcc TXServiceCode.c (This code is for the testUART.py)
    3. Enter gcc TXServiceCode.c -o TXService.run
    4. Enter cd ..
    5. Enter cp piTXISR/TXService.run TXServiceCode

*Camara files:* Camara transmission files have only one difference from normal transmission files, this is that the time stamp is replaced with a line number. EX 1: (camara data). The line number is important because should a transmission fail we will know where to resume transmissions on the next window.

Shawn Jones

GASPACS Software Team Lead