Towards Automated Launch Vehicle Telemetry Format Verification

Priya Prabhakar
QDAS,SR
Vikram Sarabhai Space Centre (ISRO)
Thiruvananthapuram, Kerala
priya8695@gmail.com

***Abstract*—The telemetry system is provided in launch vehicles to ensure the reliable acquisition of the data regarding vehicle performance and other onboard payloads [5]. Telemetry controller collect the data from Remote Units (RUs) and generate the Pulse Code Modulation (PCM) output which will be transmitted through an S-Band Transmitter. An 8bit channel data is a time slot in the PCM-frame. The resolution and sampling rate of each of the channel data depends upon the mission requirement due to limited bandwidth. 24-bit microcode residing in EEPROM of the telemetry controller controls its operation. 24-bit microcode contains the information of each slot. This 24-bit microcode is generated by taking some text files as input which contains information like bandwidth, format size, slot allocation of each channel data, etc. Telemetry controller also provides the provision to store and delay some mission-critical parameters during non-visibility period. This information is also provided in these text files. These files ensure to achieve the required PCM output format of the telemetry controller. Any error in these files can give misleading information. In the present configuration of our launch vehicles, telemetry controller can be programmed with a maximum of 4 formats with 2K size each. Manual verification of 4096 slots (max.) in each of the 4 formats is very human-intensive and error-prone method. This verification generally will take 1 week depending upon the proficiency of the person doing it. So, We have designed an automated system to verify these text files required for telemetry controller programming [2]. Our software minimizes any human-induced errors along with time consumed when done manually [4]. With the proposed system, we now able to complete the verification within 10 minutes.**

***Index Terms*—Pulse Code Modulation (PCM), Telemetry controller**

I. Introduction

The telemetry (meaning metering/measuring from a distance) system is used for monitoring various parameters in the vehicle like pressure, temperature, acceleration, converters, etc. The parameters are converted to digital form and transmitted to the ground in a serial mode by modulating an RF carrier in S-Band (2-4 GHz). A typical launch telemetry system as shown in Fig.1 consists of a sensor system that is then fed to signal conditioners for amplification, filtering, isolation, windowing, level shifting, analog to digital conversion, etc [6]. The telemetry controller communicates with RUs (max 16 RUs possible) and collects the data to generate the PCM output. RF system consisting of an S-Band transmitter accepts the processed telemetered data and transmits to the ground station after suitable modulation. There is no limit on the number of 16-bit data words that can be received from the RUs. A 24-bit microprogram stored in the EEPROM controls the operation of the telemetry controller. In each word interval, the corresponding 3 bytes (24 bits) are read from the EEPROM. This consists of 8-bit control field and 16-bit command/address filed. The control field decides whether the data is internal or external, whether the data is to be stored/delayed, and also gives the link ID to receive the next word in that group. In case of external data (from RUs), the command/address field becomes the command that will be sent to the RU and in case of internal data, this field contains required address/data. This 24-bit microprogram is stored depending upon the 3 text files as discussed in the next section.

The telemetry controller also consists of on-board memory to store data during the non-visibility period and re-transmit it during the visibility region. This memory can be used for delay mode data re-transmission at other times. In this mode, the parameters will be transmitted with a delay of “n” major frames.Telemetry controller also has a delay memory capacity, where the data to be continuously delayed are stored. There is also storage facility for the LSB of 16-bit data from the RUs.

After lower stages separation, monitoring those parameters is futile so, there is a provision of format switch over. After format switch over a new set of PCM format data is telemetered. In the present configuration of our launch vehicles, the telemetry controller can be programmed with a maximum of 4 formats with 2K size each. These formats reside in an EEPROM. Data set that is delayed or stored is programmable along with the duration by which parameters are to be stored or delayed and these data set can vary with format switch-over.

II. Input files required for verification

There are four files: .ppm, .gul, .txt and .pcm which are required for 24bit microcode generation residing in EEPROM which controls the operation of telemetry controller. These files are generated as per the mission requirement. Details of each of the files are given below:

1) . ppm file - The software-programmable parameters like no of groups, words per major frame, subframe ratio, delay duration, storage duration, format switch over type, etc are

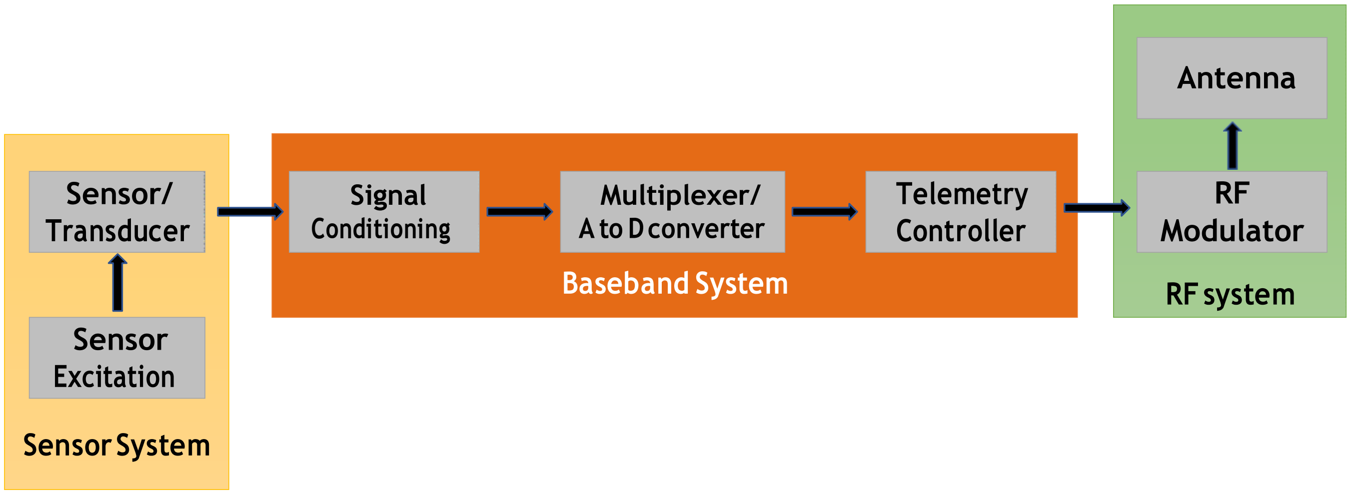

Fig. 1. Launch vehicle telemetry system

specified in. ppm file as per the mission requirement. The grouping of RUs is programmable and this configuration is selected from EEPROM. This can be changed along with the format change over.

2) .gul file - Group information, unit code and link code for each RUs is provided in this file.

3) .txt file - This contains the address mapping of each of the signal conditioners.

4) . pcm file -All the words/slots are addressed in this file. The total number of words to be stored per format is decided to depend upon the bandwidth provided and the duration of one major frame. Data to be transmitted may be either from the various sources such as external RUs, and various internal data such as Frame sync code, FID, timer, checksum, delay1 data, LSB of 16-bit data. A 24-bit timer, incriminating every one millisecond or major frame is provided by allocating 3 slots in the first minor frame followed by one slot in each of the remaining minor frames. As per the sampling rate and resolution require- ment, words for each of the parameters from external RUs are allocated in this file. Parameters which are to be delayed also specified by using the suffix '-W1' for real-time slots and '-D1' for transmitting delay data. Suffix '-S' is used for specifying the parameters which are to be stored. Depending upon the re-transmission ratio, corresponding stor- age slots are defined to transmit the stored data. Some slots are provided for the video imaging system and for monitoring bus data as well.

III. Method of Implementation

A. Present mode of verification

Depending upon the no of telemetry controller and format switch-overs decided for the mission, these text files are generated for each format. Currently, all these files are verified manually using excel [1]. Manual verification is error-prone and a very time-consuming method as it depends upon the efficiency of the assigned person. Suppose in a mission two telemetry controller are decided (for redundancy) with one format switch over and 2Mbps bandwidth assigned for each format, then total 4096 slots will be present in each format (8192 slots considering two format switch-over). The assigned person has to verify a total of 16384 slots (8192 slots for each telemetry controller). This type of verification will generally take around 1 week.

B. Proposed automated system of verification

We have created software to verify the above files required for 24bit microcode generation residing in EEPROM which controls the operation of the telemetry controller [3], [4]. The software is programmed in python 2.0 and GUI is created using PyQt5 [7]. The software is very user-friendly and a self-explanatory user manual is also available for a first time user.

The following aspects of verification are covered by the software (as shown in Fig.2):

1) Allocation of all parameters in the format as per the project requirement
2) Slot allocation of parameters as per the sampling rate
3) Addressing of both lower and upper bytes for 12Bit parameters
4) Address mapping of parameters of signal conditioners to external RUs
5) Slots allocated for Overheads (Frame Sync Code, Checksum etc)
6) Delay and storage parameters verification as per the project requirement
7) Slots verification of bus monitor data
8) Video imaging system's slots verification A separate func- tion is written to verify all the above aspects which makes code transparent and easy for debugging. The software verifies only one format at a time and provides a log file respectively which enlist all the errors present in any of the text files of that format. For the proper functioning of the software following inputs has to be given by the user:

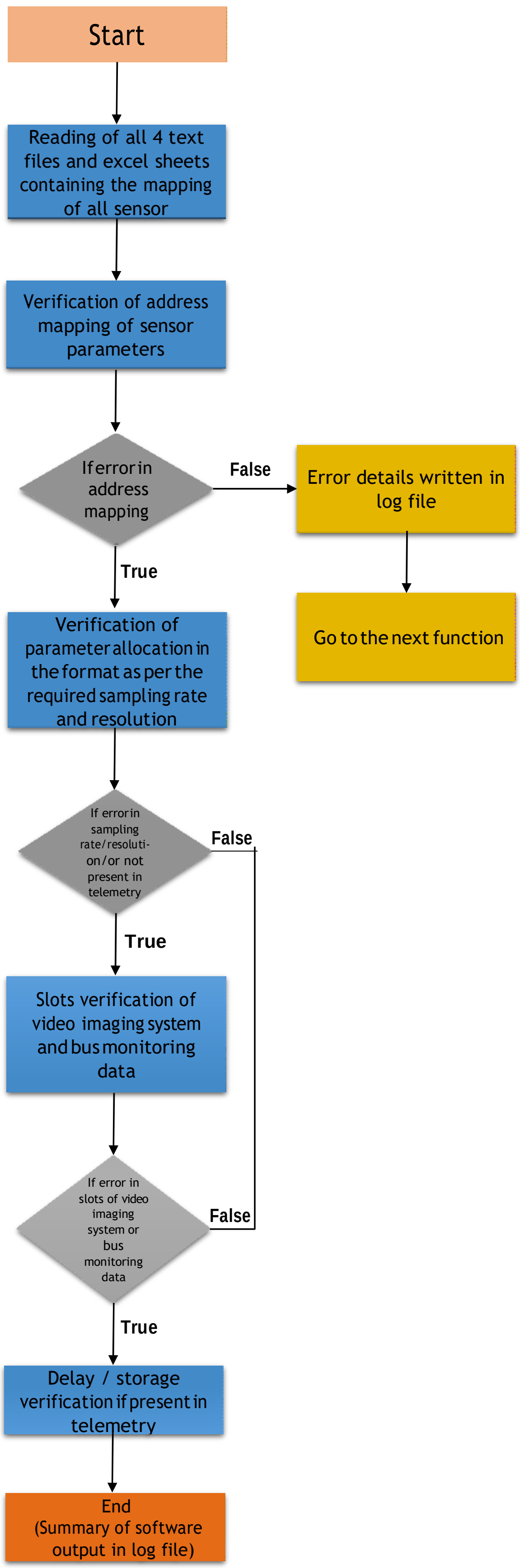

Fig. 2. Flow chart of the proposed software functioning

1) Location of the format file
2) Size of the format (e.g 128X16)
3) Location of files containing the information of parameters of each RUs.
With all the inputs available, the software completes the verification within 10 minutes

IV. Result

The software is successfully tested with various 50 test cases. The software also clearly gives a message box if there is an error in input files and guides the user to correct those input files for the proper functioning of the software. The log-file created by the software is in the form of a text file which helps the user to correct any error (if any) present in the format in a very convenient manner. The software has reduced the format verification time of one week when done manually to 10 minutes with maximum efficiency.

V. Conclusion and Future Scope

Our proposed automation method has made a very human-intensive job to 10 minutes of work. Similarly, there is much time-consuming work that can be easily be automated. The best use of automation will be in the area of data analysis. An artificial neural network (ANN) based software which can analyze the data with its intelligence and can detect a small error which may miss when done manually. The analysis of data of each parameter obtained through telemetry by ANN can help to detect even a small threat which can have a very serious impact in the future.

References

[1] Portnoy, M., An Implementation of a Complete XML System for Telemetry System Configuration, International Foundation for Telemetering, 2005.
[2] Ortiz J, Castillo J, Automating engineering verification in ALMA subsystems, InObservatory Operations: Strategies, Processes, and Systems V 2014 Aug 6 (Vol. 9149, p. 914929). International Society for Optics and Photonics.
[3] Sanner, Michel F, "Python: a programming language for software integration and development," J Mol Graph Model 17, no. 1 (1999): 57-61.
[4] Perkel, Jeffrey M., "Programming: pick up Python." Nature News 518, no. 7537 (2015): 125.
[5] Chenguang, Z., Ming, L. and Gang, W., "Research on launch vehicle telemetry data processing [J]", Missiles and Space Vehicles, 2, pp.9-11, 2005.
[6] Juan, Lin, and He Zhengguang, "Telemetry System of Manned Launch Vehicle [J]." Missiles and Space Vehicles 4 (2006).
[7] Siahaan, Vivian, and Rismon Hasiholan Sianipar, "LEARNING PyQt5: A Step by Step Tutorial to Develop MySQL-Based Applications", SPARTA PUBLISHING, 2019.