

PCM Frame Generator for Telemetry System Tests

Hatice CELIK¹, Furkan Mucahid YILDIZ¹, Mahmut Kursad ARPACIOGLU¹

*¹Turkish Aerospace Inc. Fethiye Mah. Havacilik Bulvari No:16 Kahramankazan/Ankara/Turkiye
hatice.celik@tai.com.tr, furkanmucahid.yildiz@tai.com.tr, mkarpacioglu@tai.com.tr*

Abstract:

Flight test telemetry systems involve tightly coupled processes including data generation, encoding, transmission, reception, and decommutation, making end-to-end validation both critical and challenging. Conventional validation approaches rely heavily on onboard instrumentation systems, which restrict test flexibility, complicate fault isolation, and limit comprehensive performance evaluation due to hardware constraints. This paper presents a configurable IIRIG 106 Chapter-4 PCM Frame Generator designed to enable independent and controlled verification of the telemetry chain.

The proposed system supports user-defined PCM frame structures, including synchronization words, subframe configurations, and adjustable bit rates, along with deterministic and repeatable data content modes. By modeling PCM frame definitions from XidML inputs, the generator preserves structural integrity while allowing dynamic modification of data content in real time. The system operates at the hardware level, producing compliant PCM outputs across multiple transmission rates and data patterns.

Through its RS-422 interface, the generator can directly feed both transmitter and receiver units, enabling isolated or integrated testing of RF links and decommutation systems. Known data patterns allow bit-level verification across the entire telemetry chain. The proposed solution provides a scalable and flexible test platform, improving test coverage, repeatability, and fault isolation capability in telemetry system verification and development.

Key words: Flight test telemetry, IIRIG-106 Chapter-4, PCM frame generator, XidML, Digital twin, Hardware-in-the-loop testing, FPGA

Introduction

In certification processes, flight tests are of great importance for the verification of the aircraft. For aircraft whose design has not yet been proven, telemetry systems are used to provide real-time data monitoring during flight tests and to measure and verify the reliability of the systems. With the real-time transfer of telemetry data from the aircraft to the ground station, instantaneous status checks can be performed, and precautions and decisions can be made in the event of an adverse condition.

A telemetry system consists of an onboard transmitter system, RF broadcasting, and a ground-side receiver system. The input of the transmitter system is the onboard instrumentation system, and it provides PCM Frames as output. In preparation for flight tests, system activity checks for both the transmitter and the receiver systems are important, because in the event of a failure in the transmitter or receiver system during the test, data transfer will be interrupted and the test will reach a critical point. In addition, checks are essential for

configurations that are modified and developed on the receiver system. For these reasons, verifying the operability of both the transmitter and the receiver system is critically important. Due to the dynamic test schedule and development activities, always relying on the aircraft as the data source is a constraining factor.

The hardware limitation of the instrumentation system, which is the input to the telemetry system, is an obstacle to using the maximum capacity of the telemetry system. Considering that this limitation may be removed in the future, testing and verifying the system capacity becomes important. To overcome the current capacity limitation and to perform verifications under changing conditions in the future, a software/kits requirement has emerged that will provide the input of the onboard instrumentation system.

Turkish Aerospace Industries (TAI) develops a broad portfolio of advanced aircraft, including the HÜRJET advanced jet trainer, the KAAK fifth-generation fighter, the HÜRKUŞ trainer, and a

range of rotary-wing and unmanned platforms. Each program employs distinct PCM frame structures, parameter sets, and bitrate allocations tailored to specific flight test objectives. Existing ground-based test equipment used across these programs relies on fixed configurations and commercially available PCM generators (KAM500), which limit flexibility in adapting to evolving test scenarios and do not support rapid reconfiguration between different aircraft configurations. Furthermore, these commercial solutions do not provide the capability to optimize frame structures or validate system performance under non-standard conditions. The need for a unified, reconfigurable test infrastructure capable of serving multiple programs with minimal setup time, while also enabling frame optimization and custom data injection, has driven the development of the present system.

The bandwidth used by telemetry systems has reached its maximum due to the increasing number of aircraft and the increasing bandwidth used per aircraft. To use the bandwidth most efficiently, it is assessed that the PCM Frame structure broadcast from the aircraft must be designed in its most optimal form. By analyzing the inputs of the instrumentation system into the PCM Frame and the per-parameter requirements, the goal is to derive the most suitable packet structure that satisfies the requirements. For the resulting packet, an analysis software has also become necessary that supports the most efficient parameter placement and provides an optimal layout meeting the requirements.

Because the PCM Frame input comes from the instrumentation system and the output data varies, receiver-system checks are performed in a data-independent manner. For bit-level system checks, a user-defined data content is required.

To satisfy the needs listed above, a test setup has been required that is user-configurable for verifying the entire chain from the transmitter system to the end point of the receiver system, and that can provide output with the desired data content. A software and hardware design capable of meeting this need has been carried out.

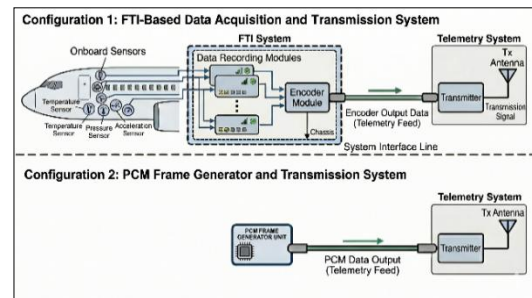


Fig. 1. Telemetry system chain (transmitter, RF, receiver, decommutation) and the position of the PCM Frame Generator in the test flow.

PCM Frame Structure

The software is designed both to create PCM Frames of user-defined dimensions and to replicate the structure of an existing PCM Frame has been designed. Certain rules have been established so that PCM Frames can be produced with standard-compliant dimensions. These rules have been formulated by taking Chapter-4 of the IRIG-106 standard as reference.

The PCM Frame Generator provides data output compliant with the RS-422 standard. The clock signal is critically important for correct data sampling and synchronization. The sampling frequency of the PCM Frame to be generated must lie within the operating range of the component used for clock generation. For this reason, the generator is designed to produce frame structures whose sampling frequencies are compatible with feasible clock generation.

In this work, the Digilent Cmod A7-35T module (Xilinx Artix-7 family) is used, and the output clock computation for the MMCM-based clock generation mechanism is given as in Equation (1).

$$\text{OutputClock} = \text{InputClock}(12\text{MHz}) \times \frac{\text{Multiplier}}{\text{Divider}} \quad (1)$$

Due to constraints inherent to the board, the multiplier and divider values can take values of 0.125 and its multiples.

$$\text{VCO} = \text{Output Clock} \times \text{Divider} \quad (2)$$

The voltage-controlled oscillator (VCO), again as required by the board, must operate between 600 MHz and 1200 MHz to provide low jitter. The VCO frequency is equal to the product of the output clock and the divider, as defined in Equation (2).

PCM Frame Generator Software Design

The study is structured to enable the user, through the software interface, to create a PCM Frame and define the content of the data.

A documentation review has been conducted to determine how the PCM Frame is to be

produced, how its content is to be defined, and how the parameter placement algorithm within the PCM Frame is to be resolved.

After the review, software work was initiated, and processors that would support the software outputs for the hardware design were investigated.

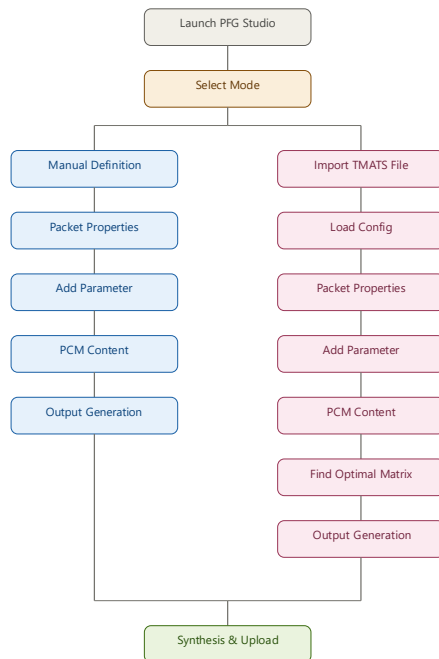


Fig. 2. Software Flowchart

PCM Frame Creation by Manual Defined Parameters

The software provides a configuration interface in which the user can define the PCM Frame structure from scratch. The user can construct a PCM Frame structure suited to their needs by specifying the fundamental parameters related to packet size (number of minor frames, number of words per row, bits per word, and frame rate). The defined structure is not transferred to the hardware until its compliance with the IRIG-106 Chapter-4 standard has been verified by the software.

According to need, the user can create a PCM Frame compliant with the rules specified in the IRIG-106 Chapter-4 document by entering Minor Frame, words per minor frame, rate, and bits per word information.

This makes it possible to place parameters with different data contents at different bit rates into the PCM Frame and apply different test scenarios.

The encoder equipment used in FTI systems within TAI supports a data transmission rate of 20 Mbps; meanwhile, by configuring the equipment in its inventory, the telemetry receiver system has the capacity to decode data transmissions at 40 Mbps. To use the telemetry ground systems at full capacity, the PCM Frame Generator software is used to produce PCM Frames at a 40 Mbps data transmission rate. This enables capacity trials and development activities for the telemetry infrastructure.

Creation of a PCM Frame from Imported TMATS File

The second operating mode of the software allows the creation of an exact replica of the existing PCM Frame structure of the onboard instrumentation system. In this mode, the packet structure is not defined by the user from scratch; the TMATS(.xidml) file obtained from the existing system is provided as input to the software, and the PCM Frame is reproduced while preserving its structure (dimensions, parameter placement, sync word, sub-frame ID ordering). This approach enables telemetry chain checks to be performed independently of the aircraft.

The XidML files generated for the onboard instrumentation system can be provided as input to the software. From the XidML file given by the user as input, the PCM Frame structure and the information of all placed parameters can be copied.

The copied PCM Frame structure and parameters can again be configured according to the user's needs.

By preserving the PCM Frame structure while parameters are added or removed and the data content of parameters is defined by the user, the user can test the onboard FTI system and the PCM Frame independently of the aircraft.

It enables telemetry system checks to be tested independently of the aircraft following updates to the instrumentation system.

XidML File Generation

Within TAI, the instrumentation system is built using equipment from Curtiss-Wright and is configured using the DAS Studio software.

In the instrumentation system, XidML-formatted files are produced both for recording and for generating the PCM Frame. To make sense of the data sent by the transmitter on board the aircraft, the data commutated in the transmitter system must be decommutated in the ground systems. XidML-formatted files are used both for generating the PCM Frame and for decommutation operations.

The real-time visualization software at the end point of the receiver system requires data in interpreted form. For this reason, the software produces an XidML-formatted file for the PCM Frame generated to be used in ground tests.

The PCM Frame Generator software emulates the onboard instrumentation system; since Curtiss-Wright is also used in TAI's instrumentation systems, the operating algorithms of these equipment and software have been taken as the basis for both PCM Frame generation and parameter placement design.

To create the XidML file, the format was studied and, under each heading, the information required for PCM Frame generation and parameter placement was identified. As a result of this work, the file required for generating the PCM Frame, commutating the signal to be transmitted by the transmitter system, and decommutation on the receiver system has been produced.

Data Content Definition

Defining the data content is one of the critical features of the software with respect to test needs. To verify the telemetry chain at the bit level, every word in the PCM Frame leaving the transmitter system must be exactly comparable with the value obtained at the receiver system. For this reason, the software allows deterministic data content to be defined on a per-parameter basis.

The software supports three fundamental data content modes: (i) Constant value mode, in which the constant value defined by the user for each parameter is repeated in every frame; (ii) Counter mode, in which sequentially increasing counter values are placed into the parameter at each sampling point; and (iii) Pattern mode, in which a bit or word sequence defined by the user is placed. This makes it possible to perform verification both for constant content and for time-varying content scenarios on a single test setup.

In tests performed with known data content, the Bit Error Rate (BER) can be computed directly by comparing the data received at the receiving component with the data transmitted. As defined in Equation (3), the BER is obtained by taking the ratio of the number of erroneous bits to the total number of transmitted bits:

$$BER = \frac{N_{\text{faulty bits}}}{N_{\text{total bits}}} \quad (3)$$

Construction of the Optimum PCM Frame Structure

A PCM Frame can be modeled as a matrix consisting of M rows and N columns. Each cell

contains a word of 16 bits. Each row (also referred to as a minor frame) contains, at its beginning, 2 synchronization words and one word holding the sub-frame ID value.

Sync Word, is user-configurable. The receiver side recognizes that the transmitter has moved to a new minor frame during data transmission by examining these 2 words. The loss of this word means the loss of the entire minor frame.

Sub-frame ID (SFID), carries the number of the row in which it resides (with the first row being 0). The receiver side, by examining this single-word information that follows the Sync Word, determines the positions of the subsequent data words. It interprets what kind of data the incoming data is. The loss of this word means that the incoming data cannot be placed correctly within the matrix, and consequently again means the loss of the entire minor frame.

Each matrix cell other than these is a data word.

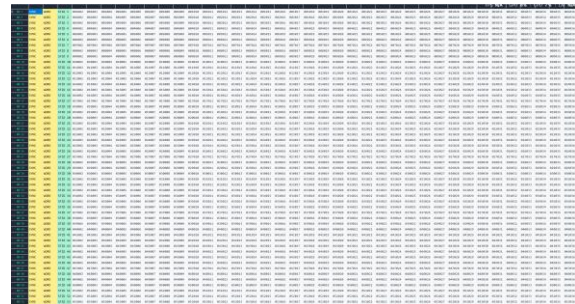


Fig. 3. PCM Frame matrix structure according to the IRIG-106 Chapter-4 standard: placement of the Sync Word, Sub-frame ID, and data words within a minor frame.

If, in the constructed PCM Frame structure, an $M \times N$ matrix is assumed, with words of width W and F denoting how many times the frame repeats per second, the bitrate calculation should be as given in Equation (4).

$$\text{Bitrate} = M \times N \times W \times F (\text{bits/second}) \quad (4)$$

Based on this information, the aspect ratio of a frame must be treated as an optimization problem. The optimum PCM Frame structure is, beyond placing the desired data into the correct matrix at the correct rates, the process of finding the most robust structure in a scenario in which data losses must also be tolerated. In a frame structure consisting of a small number of rows and many columns (e.g., 3 rows $\times$ 300 columns), missing a sync word corresponds to the loss of a significant portion of the packet (one third of the packet in the given example). Conversely, in a structure with many rows and very few columns, the bitrate increases considerably due to the additional sync word and SFID values that are added per row, and the bandwidth becomes problematic. Therefore, the optimal scenario

depends on the number of rows and columns in the matrix being close to each other.

How many times a data item is to be transmitted per second is determined by the occurrence. Its calculation is given in Equation (5).

$$\text{Occurrence} = \text{DataWordPerFrame (DPF)} \times \text{FrameRate} \quad (5)$$

This value must under no circumstances be skipped during data placement and is the most important consideration in placing the data. This value must be placed in correlation with the number of rows, and the frame structure must be constructed according to these components. There are two different scenarios for placement, depending on whether the DPF value is greater than or less than the number of rows M .

If the DPF value is exactly divisible by M , the data is placed by filling one or more columns. The required number of columns is determined according to the calculation given in Equation (6).

$$\text{Req. Columns} = \text{Occurrence} / M \quad (6)$$

After the number of columns has been determined, placement is performed by computing the spacing between columns. The column-spacing calculation is given in Equation (7).

$$\text{ColumnSpacing} = N / (\text{Occurrence} / M) \quad (7)$$

If the DPF value is an exact divisor of M , the data should be placed within a single column with equal row spacing. Placement here is performed by computing the row spacing. The row-spacing calculation is given in Equation (8).

$$\text{Row Spacing} = M / \text{Occurrence} \quad (8)$$

Tab. 1: Optimum Placement Rules

Occurrence	<M (divisor)	>M (multiple)
Placement Type	Single Column	Multiple Columns
Spacing Rule	$M / \text{occurrence}$ rows	$N / (\text{occ}/M)$ cols
Example ($M=10, N=64$)	DPF =2 → rows=0,5	DPF = 40 → cols=3,19,35,51

The effort to construct an optimum PCM Frame is carried out by taking into account all of the matrix structure, the optimum row-column counts, and the data placement properties discussed above.

Fig. 4. Parameter placement algorithm: example matrix placements for single-column ($DPF < M$) and multi-column ($DPF > M$) cases.

Finding Optimum PCM Frame from Imported TMATS File

The optimal PCM frame generation process described in the previous section can be applied to an FTI-prepared TMATS file by following the software workflow: Load Config → Find Optimal Matrix. The TMATS file used during this study defines a frame repeated 32 times per second with a 64×256 matrix structure. After excluding SFID and SYNCWORD fields, a total of 16,192 available cells remain. Among these cells, 2,737 distinct sensor parameters occupy 11,012 data words, while 5,180 data words are entirely empty and filled with a fill pattern. The payload utilization ratio for this configuration is calculated as 68%.

$$\text{Occupancy rate} = \frac{\text{DataWord}_{\text{Filled}}}{\text{DataWord}_{\text{Available}}}$$

In the efficiency calculation, the effectiveness of sensor parameter placement within the matrix is evaluated. During this assessment, syncword and SFID fields increase the bitrate independently of sensor data. However, in the event of packet loss, receiver systems rely on these syncword and SFID values to identify incoming data streams. Consequently, these overhead elements must be minimized to avoid excessive bitrate consumption, which necessitates a larger column count in the matrix. Conversely, to ensure recoverability in the event of data loss, a higher row count is also required. Additionally, the matrix should be populated with sensor data to the greatest extent possible, minimizing the number of empty data words. After processing the values mentioned in the first paragraph of this section through the software algorithm, the following results were obtained.

	Imported TMATS	Generated Optimum TMATS
M (rows)	64	64
N (cols)	256	176
Frame Rate	32	32
Bitrate	8388608 bps	5767168 bps
Total Sensor Parameter Count	2737	2737
Total DataWord	16192 (sync and sfid words not counted)	11264(sync and sfid words not counted)
Used DataWord	11012	11204
Empty DataWord	5180	60
Occupancy Rate	%68	%99.5

Parameter placement was performed in accordance with the principles stated earlier in this paper and can be opened with Curtiss-Wright DAS Studio.

The modification ratios of the TMATS file are given in the table below.

Modification Ratios	
Bitrate	%31.25 decrease
Empty Data Word Count	%98.8 decrease

User Interface Design

On launch, the software offers the user two fundamental operating modes. The Manual Definition mode allows the user to construct the structure by entering the PCM Frame parameters from scratch; the TMATS Import mode, in turn, allows an existing configuration file to be loaded so that a replica of the packet is produced. Both modes present the steps of packet sizing, parameter placement, data content, and hardware programming to the user in the same sequential flow.

Every definition made by the user is validated in real time. For non-standard dimensions, bitrate values that cannot be produced by the hardware, or inconsistent parameter placements, the interface issues a warning and the hardware programming step is automatically disabled.

This approach guarantees that all generated bitstream outputs are compliant with the IRIG-106 Chapter-4 standard and with the hardware constraints.

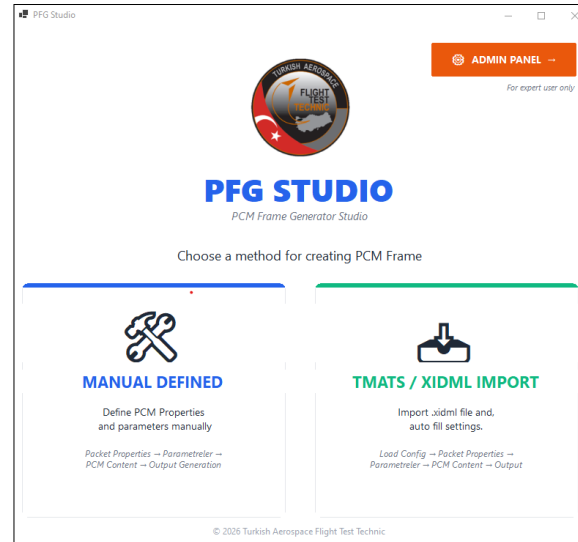


Fig. 5. Main screen view and wizard flow of the PCM Frame Generator software.

PCM Frame Generator Hardware Design

In terms of content, the PCM Frame Generator is designed to operate in two modes. The first is an internal PCM Frame structure stored within the FPGA's own ROM; the second is operation in which the desired PCM Frame output is produced by programming the hardware using the software called PCM Frame Generator Studio. This way, the hardware is also enabled to operate independently of the software.

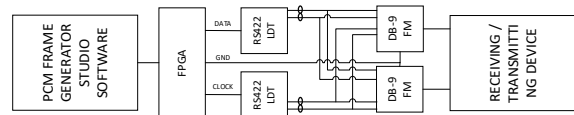


Fig. 6. PCM Frame Generator Hardware

The FPGA in the hardware design shown in the figure above acts as the brain of the hardware, performing internally the clock generation, data transmission, and PCM frame ordering required for data transmission at the desired high rate.

Data and clock transmission is converted to the RS422 format by two RS422 drivers.

Two DSUB-9 FM connectors are placed as intermediate elements so that the data/clock outputs over which data is transmitted can be converted into a cable suitable for the desired device (e.g., a DSUB-9-to-TX-connector cable for the transmitter, or a DSUB-9-to-decommutator-connector cable for the decommutation system).

Output Verification and Tests

For data verification, the CRC-32 bit-integrity check is based on the IEEE 802.3 standard. The checksum for each minor frame is computed by considering all words except the last 2 words of the minor frame. The last 2 words of the minor frame are reserved as the CRC words. In this way, the receiver side can determine whether the data received in each minor frame has arrived correctly.

In addition to the CRC-32 verification, system-level tests have been performed on the produced PCM output. In the test setup, the PCM Frame output produced by the software was connected, in sequence, to the telemetry transmitter, the RF chain, and the telemetry receiver/decommutation system. Because the generated data content is known, bit-level verification across the entire chain was performed by comparing the parameter values obtained after decommutation with the transmitted values.

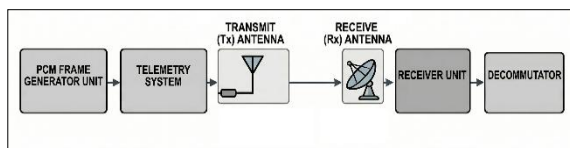


Fig.7. System-level test setup: PCM Frame Generator → Transmitter → RF chain → Receiver → Decommutation.

In tests performed at different bitrate values, the system produced IRIG-106 compliant outputs. At 40 Mbps output, the decommutation system completed synchronization and data decoding without problems; this enabled maximum-capacity tests of the telemetry ground systems to be carried out independently of the aircraft. Clock-signal jitter measurements have confirmed that keeping the PLL/MMCM configuration on the FPGA within its low-jitter operating range (VCO: 600–1200 MHz) is critical.

Separate tests were performed for the manually defined PCM Frame structures and for those replicated via XidML, and it was observed that, in both operating modes, the produced XidML and PCM outputs were correctly interpreted by the receiver side. For the data produced in the counter and pattern modes, an exact bit-level match was obtained; the CRC-32 verification mechanism was verified to successfully capture the single-bit errors that may occur within the chain.

Conclusion and Discussion

In this study, a configurable IRIG-106 Chapter-4 PCM Frame Generator system is presented that enables end-to-end verification of the flight-test

telemetry chain independently of the aircraft. The developed system allows the user either to define the PCM Frame structure from scratch or to construct an exact replica from an existing XidML configuration, and with its deterministic data content support it provides a test setup that is verifiable at the bit level.

The main contributions of the proposed approach can be summarized as follows: removal of the dependency of telemetry system tests on the instrumentation hardware; high-bitrate support (40 Mbps) that enables bandwidth usage to be tested at maximum capacity; the ability to perform end-to-end bit verification across the decommutation chain thanks to the deterministic data content defined by the user; and the fact that the hardware is also designed to operate independently of the software. The matrix-size optimization feature of the software supports minimizing the sync word and sub-frame ID overhead within the structures permitted by the IRIG-106 standard, while keeping potential synchronization losses within acceptable limits in return.

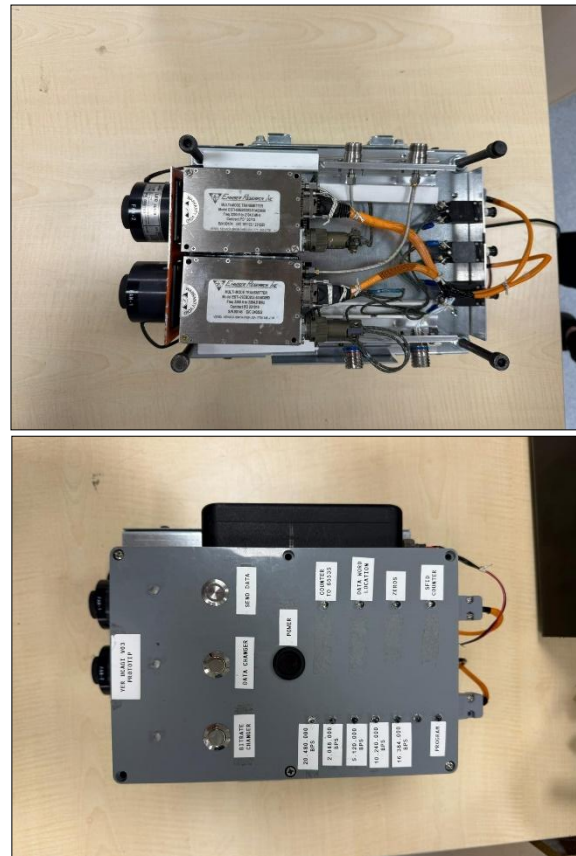


Fig.8. PCM Frame Generator

Future Works

Future work is planned to include extending the system to support Chapter-7 and Chapter-10 packet structures, testing the behavior of the receiver system under boundary conditions via

error-injection mechanisms embedded on the FPGA, and developing automatic optimum packet suggestion algorithms that work in integration with the existing packet optimization software.