Space Based ADS-B receiver for ISTsat-1

Luís Pereira
Instituto Superior Técnico
Universidade de Lisboa
Av. Prof. Dr. Cavaco Silva, 2744-016 Porto Salvo
Email: luis.bras.pinto.pereira@tecnico.ulisboa.pt

***Abstract*—A CubeSat is a 10 cm cube low-cost satellite with a mass up to 1.33 kg commonly launched into a Low-Earth orbit (LEO), which extends until an altitude of 2000 Km. The ISTsat-1 is the first nanosatellite developed by the ISTnanosat team. This satellite is intended to test the reliability of an aircraft tracking system namely the Automatic Dependent Surveillance-Broadcast(ADS-B) system, being its scientific payload.**

The major requirement for the payload mission is related to the acquisition of signals while decoding ADS-B messages at the same time. This master thesis focuses on the description of the ADS-B messages that are relevant to the algorithm developed in order to receive and decode them. Alongside, the possible architectures to overcome the mission requirements are described with particular attention to the ADS-B algorithm necessary to detect and decode ADS-B messages and the LPC4370 peripherals that will help to receive and sample the respective messages. Therefore, the several results acquired with the final solution are provided and a performance evaluation is given regarding the CPU usage and the power consumption.

Finally, a real test case is provided in order to understand the feasibility of the system in the earth's surface and to guarantee that the system works as required.

***Keywords*—ISTSat-1, ADS-B, Mode-S message, LPC4370, algorithm.**

I. Introduction

A CubeSat is a 10 cm cube low-cost satellite with a mass up to 1.33 kg used in a Low-Earth Orbit(LEO). A LEO is an Earth-centered orbit until an altitude of 2000 Km. Professors Jordi Puig-Suari from California Polytechnic State University and Bob Twiggs from Stanford University developed the first CubeSat back in 1999, enabling graduate students to design, build, test, and operate limited capabilities of artificial satellites. CubeSats were first deployed by a military organization in 2002 and by a university organization in 2003.

The ISTSat-1 is the first Cubesat developed in Portugal. This project consists of a standard dimension of 10 cm x 10 cm x 10 cm (1U) CubeSat being developed by the ISTnanosat team at Instituto Superior Técnico(IST). This project is developed under the Fly Your Satellite(FYS) program supervised by the European Space Agency(ESA) Education office, allowing to complement the academic education of students and prepare them for future work on the space sector [1]. In the ISTSat-1, the idea was to design and build all subsystems in-house to follow the concept of the program. The mission of the ISTsat-1 will serve to demonstrate in orbit the Automatic Dependent Surveillance - Broadcast(ADS-B) technology. Moreover, the aircraft tracking data collected and transmitted to the ground shall be analyzed and correlated with ground segment datasets to characterize the "Cone of Silence" and measure the performance parameters of the receiver chain, namely Probability of Target Acquisition(PTA), Probability of Detection(POD) and Probability of Identification(POI) [2]. Besides, the payload mission was proposed to test this new technology using a new in-house designed receiver whose reception results shall be compared with the ones already acquired by a commercial one.

This dissertation focus on the design, implementation, testing, and performance regarding the software of the Payload subsystem that will be responsible for capturing and processing all data related to the mission of the satellite. The proposed mission will serve to demonstrate the necessary capabilities when offering a system to capture the ADS-B messages.

A. Motivation and goals

The ADS-B system is the most studied technology that will improve the monitoring of aircraft not only in remote areas and oceanic routes but also on landing and take-off operations. ADS-B is a system where an aircraft periodically broadcasts its position (gathered via Global Navigation Satellite System(GNSS)), its unique identification (International Civil Aviation Organization(ICAO) address) and other information regarding its speed and altitude, enabling the aircraft to be easily tracked.

The ICAO believes that ADS-B will eventually become the preferred surveillance technology worldwide in the next few years, as it will be mandatory the use of this system by all aircraft in 2020 [3]. The system works independently, and neither pilots nor controllers have to do any actions. In the last couple of years, projects regarding the Space-based ADS-B system have been developed in small satellites to verify the operation of this system which can solve all the problems that current tracking technologies are facing. The main goal of this dissertation is the reception of ADS-B messages on the payload system. However, the mission of the ISTsat-1 is to detect, store and downlink the received messages and compare them to ground segment datasets of messages on the Ground Station(GS).

This document describes the research and work developed to accomplish a fully functional ADS-B payload, being organized in six chapters. Chapter 1 introduces this report, with the motivation and objectives of the master thesis. Chapter 2 describes the ADS-B system, its components, space-based

ADS-B systems and ISTsat-1 internal organization. Chapter 3 details the mission requirements, describing the architecture of the solution. Chapter 4 describes the implementation of the architecture giving particular details regarding the algorithm choices, the peripherals usage and the test bench. Chapter 5 describes an evaluation of the performance of the algorithm responsible for detecting, decoding, and filtering the message as well as the power consumption of the subsystem developed. Chapter 6 summarizes the developed work.

II. State of the Art

ADS-B is one of the most recent technologies designed to help controllers and pilots in the aviation industry. Europe and United States(US) share now the same deadline date (January 1, 2020) to have all aircraft equipped with the ADS-B system.

Along with the usual ADS-B system, several studies and investigations have been developed during the last few years regarding the Space-Based ADS-B system implementation. The ISTSat-1 is intended to be part of the studies and investigations regarding this space-based technology.

A. ADS-B system

The ADS-B system was designed to complement the conventional radar, allowing to control aircraft with greater precision and over a substantial percentage of the earth's surface, being also much cheaper to build and implement. Therefore, with the usage of satellites, aircraft transmitters and ground station receivers, the goal of this system is to help both controllers and flight crews and to cover 99% of the earth.

According to [4], ADS-B means:

- Automatic, because position and speed information is automatically transmitted (at least once every second) without flight crew or operator inputs.
- Dependent, because transmission is dependent on proper operation of on-board equipment that determines position and speed.
- Surveillance, because information about aircraft is in continuous observation.
- Broadcast, because the information is broadcasted to any aircraft or ground station equipped with ADS-B receiver.

As illustrated in Figure 1, aircraft obtain their position through the GNSS and altitude and velocity from the on-board systems, which are then broadcasted. This feature allows the reception of signals on a satellite, other aircraft or GS. This is only possible because aircraft have one antenna mounted on the bottom of the fuselage and another on the top of the fuselage.

B. ADS-B components

Currently, the ADS-B system has three components: airborne components for ADS-B OUT, airborne components for ADS-B IN and ground components.

ADS-B OUT is is responsible for broadcasting the speed, altitude, identification and location of the aircraft to other aircraft and ground stations, being its equipment required in all aircraft by the Federal Aviation Administration(FAA) and European Commission(EC). Commonly, a GNSS receiver processes the GNSS satellite signals to produce the aircraft position and velocity that will be broadcasted.

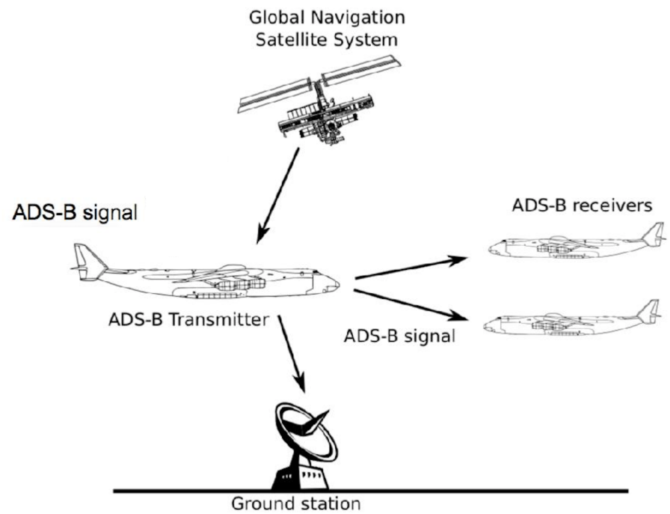

Fig. 1. ADS-B system overview

FAA and EC do not currently require ADS-B IN equipment. However, if this equipment is mounted on an aircraft, it will have the ability to provide pilot-usable flight information such as traffic and weather information.

Ground components include an ADS-B antenna receiver with an unobstructed view towards the horizon responsible to capture the signals emitted by an ADS-B OUT system. This component is responsible for receiving the messages used by the control towers to have more accurate locations of the planes.

C. ADS-B messages

Every aircraft equipped with the ADS-B OUT system can send messages that will be captured by an ADS-B IN receiver mounted on another plane or ground segment, needing a rule-compliant ADS-B OUT solution to transmit those messages and a 1090 MHz extended squitter unit to be compliant with international rules.

This extended squitter is a subsystem of the current Mode-S where a transponder squit only sends the most necessary information about aircraft identification, system status and pressure altitude information. The word squitter refers to a periodic burst or broadcast of aircraft-tracking data that is periodically transmitted without interrogation from the controller's radar. Mode-S is a Secondary Surveillance Radar(SSR) process that allows selective interrogation of aircraft using the 24 bit ICAO address. The extended Mode-S allows sending 49 individual parameters compared to the 7 for the normal Mode-S.

The structure of an ADS-B message always has two segments: Preamble and Data block. The former lasts for 8μ and is used to synchronize the receiver with the transmitter, while the latter is used to transport the specific information of an aircraft like its position, speed, altitude and identification. Therefore, ADS-B has a data rate of 1 Mbit/sec.

Figure 2 gives an overview of the whole preamble pattern.

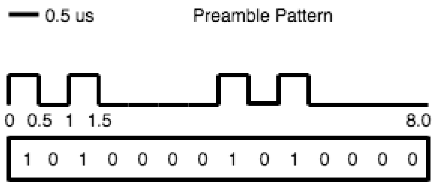

Fig. 2. Preamble pattern of an ADS-B message

Only the data block is encoded using Pulse Position Modulation(PPM) and Manchester Encoding while the Preamble part only has spikes. The receiver uses Manchester decoding to determine the high and low bits inside the data block. Inside this block, in the time allocated for transmitting one bit, the pulse may occupy the earlier (left) or later (right) half of the timeslot encoding either 1 or 0. For a logic "0", a transition from 0 to 1 is needed. For a logic "1", a transition from a 1 to 0 is needed.

Each data block contains an ADS-B message with a defined format of 112 bits, as shown in Figure 3.

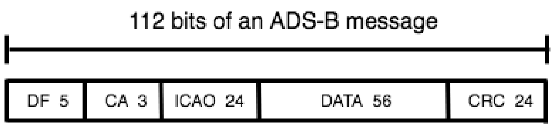

Fig. 3. Main ADS-B message fields

- Downlink Format(DF) (5 bits) - Identifies the type of message. The correspondent decimal value should be equal to 17 or 18 to identify an ADS-B message.
- CA (3 bits) - Additional identifier that is intended to give users more information regarding each message.
- ICAO (24 bits) - Carries the unique address of the aircraft, being also called as Mode-S "hex code", and is part of the Certificate of Registration of a single aircraft. The Certificate of Registration, which is the same as a personal identification card but for aircraft, is maintained by the NAA, a government authority responsible for the approval and regulation of the civil aviation. Some rules should be taken into account for the ICAO addresses where these addresses shall not be changed during a flight, and those with 24 ZEROs and 24 ONEs shall not be assigned to any aircraft as a security measure.
- DATA (56 bits) - Carries information regarding the position, velocity, location and altitude depending on the message sent by the aircraft.
- The InterrogatorID/Parity (24 bits) - Based on a Cyclic Redundancy Check(CRC) which allows validating the correctness of the received message.

Decoding the first 5 bits of the 56 bits of the DATA segment is useful to understand the type of message. For each one of those messages, the remaining bits of the DATA segment carries information to calculate or to directly retrieve the values of velocity, location, altitude, speed or identification according to the particular type of message.

Each ADS-B message is broadcasted with a period that ranges randomly between 0.4 and 0.6 seconds or between 4.8 and 5.2 second depending on the type of message and location of the plane (air/ground). This randomization function is designed to prevent aircraft from having synchronized transmissions on the same frequency, and thus overlapping each other's communications. However, the overlapping in communications is still usual and can lead to errors, as several aircraft can send their data at the same.

D. Space-Based ADS-B Systems

Space-Based ADS-B systems are being developed over the last years attempting to fill the gaps of the conventional ADS-B system. One of the most significant achievements of this evolution will allow that even if ground stations do not exist to receive the ADS-B signal, air traffic controllers will have access to the correct location of the aircraft. This system will have great utility in remote areas or oceanic routes where there is no available Ground Segment equipment.

1) DLR: From 2009 to 2013, German Aerospace Center(DLR) and SES Techcom Services developed the first project of ADS-B over Satellite or Space-based ADS-B. The PROBA-V mission was the first in-orbit demonstration of this surveillance system. It was made in the frame of ESA's PROBA-V mission (PROBA Vegetation [5]) on 23 May 2013, where, in just two hours, at an altitude of 820 km it recorded more than 12,000 ADS-B and Mode-S messages. Proba-V was the first experiment of this kind with proved feasibility of a Space-based ADS-B system and is responsible for the basis of future developments in this area.

2) Iridium: Iridium is set to be one of the future companies regarding real-time tracking ADS-B systems [6]. Iridium and NAV CANADA signed a contract for a joint venture to be run under the company Aireon LLC with support from the US FAA. The ADS-B receiver payloads, to be mounted on each Iridium NEXT satellite, will operate independently and perform the air traffic surveillance function separately from the primary mission of the spacecraft. The Iridium NEXT LEO constellation is the world's largest with 66 operational satellites providing global coverage, system availability, redundancy and flexibility [7].

E. ISTSat-1 System Overview

A brief overview of the ISTsat-1 system is done to contextualize the satellite for which the Payload subsystem is developed. The general structure of the ISTsat-1 follows a typical system structure of a CubeSat with four major components: On-Board Computer(OBC), Electrical Power System(EPS), Tracking, Telemetry and Control(TTC) and Communications Processor(COM). The Payload , described in this report, is added and is related to this CubeSat mission. Inside the ISTSat-1 all subsystems are interconnected through a common and standard bus system, the PC/104, which allows the stacking of each subsystem board on top of the other.

In the case of ISTsat-1, the aircraft tracking data collected and transmitted to the ground shall be analyzed and correlated with ground segment datasets to characterize the "Cone of Silence" and measure the performance parameters of the receiver chain, namely PTA, POD and POI [2].

After acquiring the ADS-B messages, the Payload subsystem shall send these to the COM subsystem, responsible for the satellite data storage. In this subsystem, the messages are compressed and sent afterwards to the Ground Segment through the TTC, when a space link is available. Here, one of the most conditioning factors on the ISTSat-1 is related to the limited bandwidth of communications between the satellite and the GS. Three different data rates may be used: 1.2 kbit/s, 9.6 kbit/s and 48 kbit/s. Depending on the space link and signal quality, these communication bandwidths may affect the performance of the ISTsat-1 mission, as the received ADS-B messages may take longer to be downloaded to the GS.

III. Mission

The mission of the ISTsat-1 is to characterise its ADS-B OUT receiver. To accomplish that characterization, it is necessary to collect the messages broadcasted by commercial aircraft. The aircraft data collected in orbit and transmitted to the ground station shall be analysed and correlated with ground-based datasets of messages to demonstrate the ADS-B detection technology. The results of this correlation will confirm if the system is working correctly (reception of signals and software processing of messages).

Aircraft typically use a blade antenna mounted on top of the fuselage and another at the bottom, alternating the transmission between the two antennas, at 1090 MHz. Its linearly polarized radiation pattern generally presents a null along the z-axis which is responsible for a "Cone of Silence" around the aircraft zenith and nadir, which can extend over a range between 60km and 100km at an altitude of 400km. This means that, when a aircraft are between this range, the satellite antenna will not be able to detect messages broadcast by them, which will probably cause interruption on the reception of messages from a specific aircraft during a period. The ISTSat-1 will use an ADS-B antenna with circular polarisation that will be responsible for receiving ADS-B OUT messages sent by the top-mounted antenna of an aircraft. Figure 4 represents a "Cone of Silence" that may occur on the ISTsat-1 mission [8].

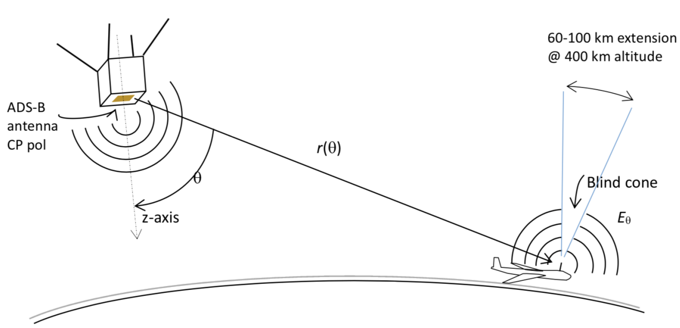

Fig. 4. Cone of silence representation. Source: [8]

The "Cone of Silence" will be studied in our mission. The idea is to verify in the ground station if a breakdown in communications occurred for a particular aircraft, being necessary to confirm if the plane was passing directly above the satellite afterwards.

As described in [9], there are several requirements approved by ESA specialists that need to be met in this mission. It is mandatory to refer to those requirements as they are the basis of the software architecture developed for the Payload system.

- Requirement PLS-FKT-180: The ADS-B receiver shall be able to receive samples and decode messages simultaneously;
- Requirement PLS-FKT-150: The ADS-B receiver shall allow filtering the ADS-B messages sent for downlink using filters;
- Requirement PLS-DCP-030: The ADS-B receiver shall be able to communicate with the COM subsystem through a digital interface;
- Requirement PLS-ITF-110: The generated Message Statistics shall include: number of preambles detected and number of messages decoded;
- Requirement PLS-CTR-360: The ADS-B receiver power consumption shall not exceed 0.8W in Normal mode;
- Requirement MIS-DCP-030: The mission is considering that in a time frame of 10 minutes, it should collect around 40.000 messages from 20 different aircraft.

To verify the ISTsat-1 solution regarding the ADS-B receiver, only three of the four parameters defined in PROBA-V mission will be used: the PTA, the POI and the POD.

The following expressions define three of the four parameters defined in PROBA-V mission that will also be used for the ISTsat-1 ADS-B receiver:

$$PTA = \frac{\text{Actual number of targets detected}}{\text{Expected number of targets to be detected}} * 100$$

Targets detected means the reception of a single message with the correspondent ICAO address (unique ID of aircraft).

$$POD = \frac{\text{Actual position messages received}}{\text{Expected positions messages to be received}} * 100$$

Position messages mean the reception of ADS-B messages with a TC decimal value between 9 and 18 or between 20 and 22.

$$POI = \frac{\text{Actual identification messages received}}{\text{Expected identification messages to be received}} * 100$$

Identification messages mean the reception of ADS-B messages with a TC decimal value between 1 and 4.

A. Architecture

Figure 5 describes in detail the proposed solution and a description of each block inside this architecture is detailed further.

The left part of this architecture is called the receiver chain, which is intended to be a straight-forward process. The process always starts by detecting signals from a particular aircraft through the hardware ADS-B Radio Frequency(RF) frontend.

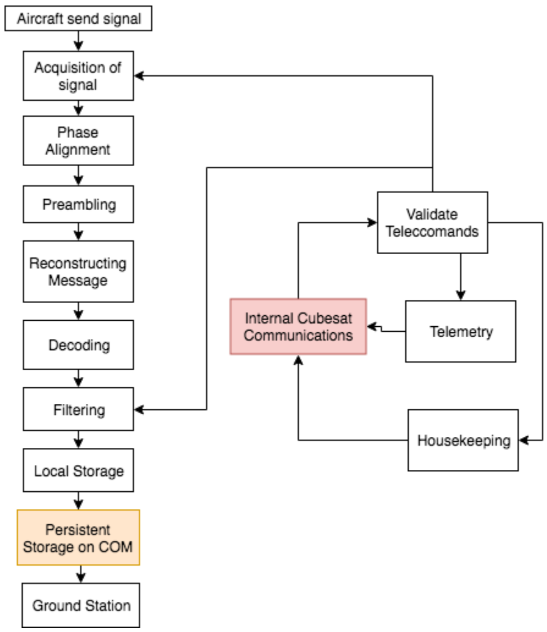

Fig. 5. Functions representation to accomplish the mission

After that, all the functions of the receiver chain are software-based and will allow the system to detect, decode and filter the correct messages. The right side of the figure represents other small functions necessary for the Housekeeping, Telemetry and command validation. In the middle, the coloured block demonstrates the internal communications done inside the ISTSat-1. The following sections describe in detail each block, giving particular attention to what are their roles inside this architecture.

1) Acquisition of Signal: The signal acquisition block is responsible for the reception of ADS-B signals and converts them into samples. The antenna mounted on the ISTSat-1 is responsible for capturing the ADS-B OUT signals sent from the aircraft. The receiver then uses an 12-bit High Speed Analog-to-Digital Converter(ADCHS) that can receive input voltages and then convert them into samples, which digital values corresponding to a specific voltage in a defined bit range. After this conversion, made only recurring to hardware, the software will start to be fully used to process each one of the samples. As described earlier, ADS-B messages are encoded using Manchester code with a bit rate of 1 Mbit/s. This must be taken into account when choosing the sampling rate necessary as Manchester code needs an oversampling for the decoding process.

B. Preambling and phase alignment

The phase alignment is intended to align the start of the preamble, being responsible of identifying the first sample of the preamble sequence so that no propagation errors occur. The preamble is a distinctive pattern or sequence of pulses that can be easily detected, allowing the receiver to unambiguously determine the position of the high and low bits within a message (synchronization method).

The number of samples to use in this step is related to the sampling rate. For a better understanding, a sampling rate of 12 MSamples/s is assumed, where each preamble step occupies half the time bit, meaning that only six samples are used to determine each step. In addition, the preamble does not use the Manchester code, which means that for a transition, it is necessary to compare each step value of six sample values. The objective of the preambling phase is thus to verify the several spikes represented as ”1” in the preamble pattern, as this verification will correspond only on the difference between consecutive samples.

Along with this, a variable is used to maintain the number of preambles detected that will correspond to all Mode-S messages received.

C. Reconstructing Message

The following stage starts the reconstruction of the ADS-B message, as the transmitter and receiver are correctly aligned due to the preamble pattern. The samples acquired in the acquisition phase are again used to reconstruct each bit of the message, which are encoded in Manchester code. The Manchester code for each bit creation always displays a transition on the bit time in one of two possible ways: from high to low or from low to high. The transition occurs in the middle of the bit time. Through this method, the number of samples to evaluate is always equal to the Sampling rate divided by the message bit rate. For example, at 12 MSamples/s, it is necessary to analyse 12 samples to construct one bit. It is possible to reconstruct each bit in two different cases. The first one is to have the first six samples at a high level and the next six at a low level, creating a bit ”0”, and the other case is to have the first six samples at a low level and the next six at a high level, creating a bit ”1”. This process is repeatedly used until the 112 bits of the message are reconstructed.

D. Decoding Message

The message decoding is mainly based on the dump1090 code, which is an OpenSource code available to all people interested in the reception of ADS-B messages and other Mode-S messages [10]. For the purpose of this work, only the specific decoder for the ADS-B messages is considered, as other Mode-S messages are not relevant for this mission. This code allows any user to use a simple Software Defined Radio(SDR) connected to their computer to instantly start seeing live messages sent by the aircraft on a terminal or web page.

A couple of statistical measures regarding the number of messages decoded with good CRC and the total number of messages decoded shall be maintained during this process.

E. Filtering

Filtering will play a vital role to restrict the number of messages stored on the COM subsystem. They will also be essential to calculate the different measures of the mission accurately. A couple of filters have been defined for this purpose on the payload, namely:

- Filter by Location, used to create a virtual rectangular surface area where it is supposed to accept only the messages coming from aircraft inside that area.
- Filter by Time, used to accept messages between a defined start and end time.
- Filter by Altitude, used to accept messages above or under an altitude defined in the filter.
- Filter by Velocity, used to accept messages above or under a certain velocity defined in the filter.
- Filter by ICAO, used to accept messages from a particular ICAO.
- Filter by Type Code Number, used to accept messages with a specific Type Code.
- Filter by Aircraft Identification, used most of the times with the Type Code Number filter, being another way to retrieve only messages that bring the identification of the aircraft.

Payload software must handle with one or multiple active filters at the same time. Furthermore, some filters are not strictly connected to the mission measures but will provide some extra testings on the reception of the different messages.

F. Internal Cubesat Communications

For the communications between two subsystems, the IST-nanosat Control Protocol(INCP) shall be used to correctly format the messages exchanged by the subsystems [11].

In the send function, an INCP encoding and an I2C frame formatting to send the information from the payload to COM or OBC shall be done. The send function will be executed whenever the payload needs to respond to a command, report an error or transmit the mission data.

The inverse process shall be done in the receiver function while receiving new messages from those two subsystems. The I2C handler can check the integrity and reliability of the message.

In both functions, statistics regarding I2C messages transmitted and I2C messages received will be done during the entire time payload is active.

G. Telecomands

Internally, some telecomands have been defined for the payload subsystem. This function will be responsible of dealing with all the cases defined and trigger the respective action. A list of the different telecomands that shall be available is given: setup each one of the filters, reset filters, activate or deactivate a filter, get the filters list, reset statistics and enable or disable the Housekeeping and Telemetry.

This will be essential to receive orders from other subsystems and trigger one of the desired actions without any mistake.

H. Telemetry

Telemetry function is going to collect private information regarding the system and the mission. This information shall then be sent to the OBC subsystem after being requested. A list of the telemetries that shall be collected inside the payload is given: number of active filters, total number of filters, number of decoded messages, number of detected preambles, last ADS-B message received, number of I2C messages transmitted and number of I2C messages received.

I. Housekeeping

The housekeeping block must be triggered in a defined interval of time to understand the operation of the different functions. This functions will be responsible for testing each small part of the software of the payload providing information about every process running on the Payload system. Receiver chain shall be verified by inserting a good preamble followed by a message and understand if it can process the preamble, reconstruct the message, decode and filter. This information shall be stored on a report that will be sent downwards to the Ground Station.

IV. Implementation

A. Design of the Solution

LPC4370 is an ARM Cortex-M4 based microcontroller for embedded applications which include an ARM Cortex-M0 co-processor and an ARM Cortex-M0 subsystem. In this microcontroller, all cores can manage peripherals such as ADCHS, General Purpose Direct Memory Access(GPDMA) and Serial General Purpose I/O(SGPIO). Additionally, 64kB of ROM can contain boot code and on-chip software drivers.

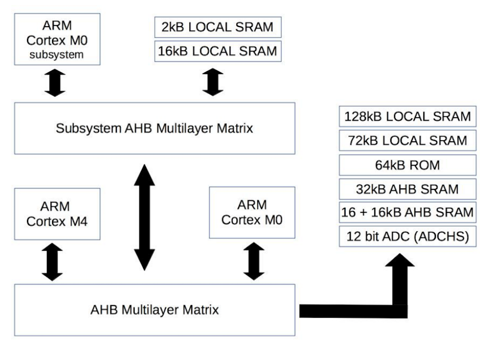

Fig. 6. LPC4370 memory and processors

The final architecture is shown in figure 7 and only has the ARM Cortex-M4 active. After the implementation of the receiver chain, it was possible to understand that this core can handle with all functions of the system. Moreover, special attention will be given to the receiver chain that contains the signal acquisition, phase alignment, preambling, message reconstruction, decoding, filtering and local storage. This is a straight-forward process where each function receives the output of the last one. Phase alignment will receive the samples retrieved in the signal acquisition. Preambling will receive the starting point of the preamble gathered on the phase alignment. Message reconstruction will take place as soon as preamble conditions are met. Decoding can only start after the correct reconstruction of the 112 bits of the message. Filtering can only be used after the correct decoding of the message. Local storage will only occur if the message has passed the active filters at the moment. If all these conditions are met, the message may be sent through the I2C formatted with the INCP to the COM subsystem for persistent storage. Telemetry, Housekeeping and Validate telecomands functions will only occur sporadically or on demand from another subsystem.

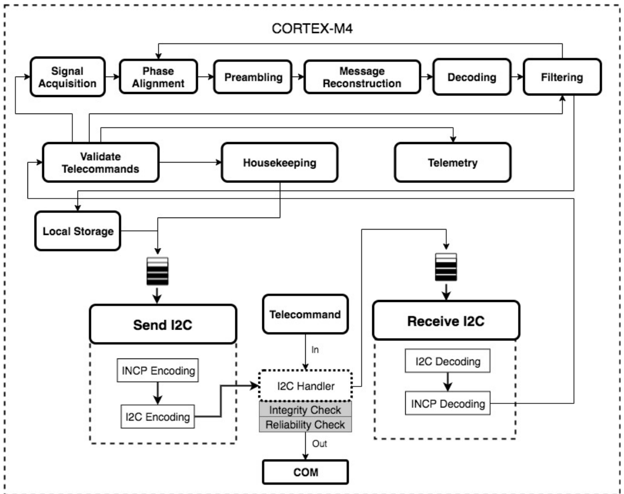

Fig. 7. Final architecture design of the LPC4370

V. High Speed ADC and GPDMA

The LPC4370 was chosen mainly because of the ADCHS peripheral and because of the triple-core architecture that allows detecting and decoding messages at the same time. This peripheral is responsible for receiving input voltages from the RF frontend and converting them into samples at a specific sampling rate, which are transferred to a particular location on the Static Random Access Memory(SRAM) memory using the GPDMA. The usage of the GPDMA peripheral is intended to save processing power from the M4-Core as the converting is done by hardware and not software-based. Therefore, the samples required for the Payload algorithm are provided through a combination of the ADCHS and GPDMA peripherals, as shown in Figure 8.

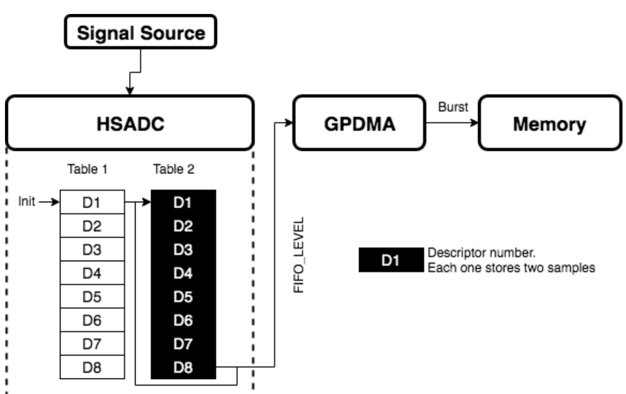

Fig. 8. Process to put ADCHS samples directly to memory using GPDMA

A. High Speed ADC Configuration

The first thing considered for the ADCHS peripheral is the configuration of its sampling rate. Each conversion takes only one clock cycle of the clock of this peripheral, whose maximum value is 80 MHz. However, to provide a maximum sampling rate of 12 MSamples/s, a clock of 12 MHz is selected, as this lower clock peripheral saves power.

Recalling Figure 8, the ADCHS is configured with the use of two descriptor tables, which may be considered as two internal state machines with 8 descriptors each. The ADCHS clock feeds an internal peripheral timer, and this state machine is verified at each increment, meaning that each descriptor may be configured to be called and start a conversion based on a match with a certain value of this timer, providing different acquisition rates.

The first table is only used to provide an initial startup delay, required for the ADCHS peripheral to start its acquisitions correctly. For that, its first descriptor is set with a startup delay required to be higher than 12 μs, according to the manual of the processor. After that, the state machine jumps to the next descriptor table and stays there for the rest of the acquisition time.

Since a stable sampling rate of 12 MSamples/s is used, every descriptor of the second table is configured to reset the aforementioned timer and all match values are set to 0, meaning that two samples are acquired at every successive descriptor call (two samples due to the aforementioned packed feature). A round robin configuration of the second table is done so that the ADCHS is continuously running, meaning that the state machine returns to D1 from D8.

B. GPDMA

At this point, it is known how the ADCHS is configured, running continuously and providing two packed 12-bit samples in a 8 position First In First Out(FIFO) buffer that must be constantly emptied. Therefore, when this buffer is filled with the resultant 16 consecutive samples, a DMA request is issued to transfer these samples directly to another memory region without using the processor cores for that.

The DMA is configured so that these transfers of 32 bytes (2 bytes per sample) are done simultaneously, being incrementally repeated to two consecutive DMA buffers with a fixed size of (4088 * 4) = 16352 bytes. Theoretically, with this size, the chosen DMA buffer takes approximately 681 μs to fill up at a sampling frequency of 12 MSamples/s. Furthermore, the two buffers are configured as two linked lists, meaning that the next buffer is filled up without the interruption of the ADCHS acquisitions and following DMA transfers after the aforementioned DMA interrupt is called.

C. ADS-B Algorithm

The developed ADS-B algorithm contains a couple more significant functions compared to the Dump1090 implementation, namely, Find First Low, Preambling phase, Reconstruct phase, Decoding and Filtering. The Find First Low function has been inspired by the Detect Message function of the Dump1090. However, Dump1090 uses a different way of checking the samples, as the algorithm firstly fills a buffer to process it afterwards, losing new samples while this processing is done. In this new algorithm, it is required to detect and decode messages at the same time. DMA is used to retrieve the samples as fast as possible, which allows saving processing power of the ARM Cortex-M4that will handle the algorithm processing. Figure 9 represents the state machine that is used inside the DecodeBuffer function that is responsible to control the flow of the receiver chain.

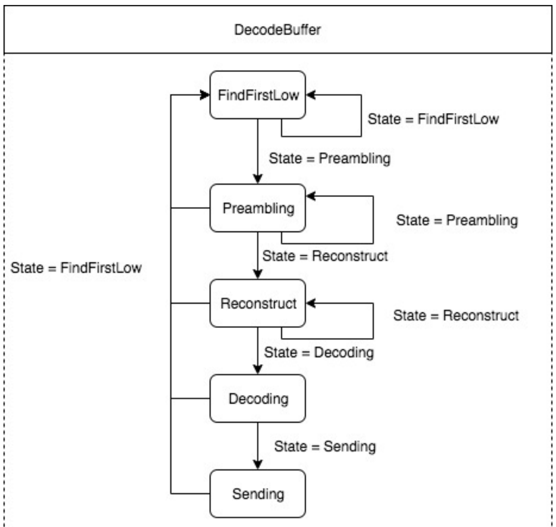

Fig. 9. Machine States of DecodeBuffer Function

1) Dynamic Threshold Calculation: Before entering the DecodeBuffer state machine, the system will be responsible to calculate the value of the dynamic threshold. The calculation is based on processing several buffers and checking the sample values to find the highest and lowest values. This is a crucial step in order to correctly define the dynamic threshold that shall be used to reconstruct the ADS-B messages.

2) FindFirstLow Phase: In this phase, the idea is to process the samples and find the first spike of the preamble pattern using the capabilities of ADCHS and DMA. There are two processes that can be used here. The first one processes the 96 samples correspondent to a preamble each time. if the preamble is not matched, the first sample shall be dropped and the next one on the buffer shall be used. For the construction of each "bit" of the Preamble, six samples shall be used. The second one is intended to save processing time. It always uses 12 samples and the idea is to detect a transition higher than the Fixed Threshold. The dynamic threshold shall be updated when the Payload system is powered on but also from time to time because this threshold is not constant due to different noise levels.

After the possible first spike is found using this process, the idea is to unambiguously detect the first sample under the dynamic threshold of the dynamic range. The idea is to iterate one by one all the 12 samples where the spike has been found and clearly identify the first one smaller than the medium threshold value. When the first sample lower than the threshold value is found, the synchronisation is done and the algorithm moves to the Preambling phase.

3) Preambling Phase: Regarding the Preambling phase, the new algorithm uses the same logic as the Dump1090, using a specific pattern that needs to be matched. Here, the Manchester code is not used, leading to process six samples each time instead of the 12 used with Manchester to reconstruct one bit of the message. The sum of each six samples is stored in a buffer with the size of 16 positions which is the size of the Preamble pattern. The comparison is then made using the values inside that buffer.

$P[0] < P[1]$	$P[8] < P[9]$
$P[1] > P[2]$	$P[9] < P[6]$
$P[2] < P[3]$	$P[10] > P[6]$
$P[3] > P[0]$	$P[11] > P[6]$
$P[4] > P[0]$	$P[12] > P[6]$
$P[5] < P[0]$	$P[13] > P[6]$
$P[6] > P[0]$	$P[14] > P[6]$
$P[7] > P[8]$	$P[15] > P[6]$

TABLE I
PREAMBLE COMPARISON PATTERN

In Table I, each P[number] corresponds to the sum of 6 consecutive samples that shall be stored inside the small buffer with 16 positions (P[16]). Those comparisons are always made step by step, and if a condition does not match, the program will directly go to the FindFirstLow state, which means that no valid preamble has been detected. Furthermore, it will be directed to the DecodeBuffer state machine, restarting the whole process of checking for a preamble starting from the first sample of the last six samples that did not pass the condition.

4) Reconstruct Phase: This phase is responsible for converting samples into bits using Manchester Code. The first thing to determine is the level of the samples. The algorithm grabs the second and third samples and does an average value of both to determine if the level is high or low. High and low samples are determined comparing the average value with the dynamic threshold that has been calculated. After the decision, it is necessary to determine where the transition occurs. Ideally, if all samples come perfectly aligned, twelve samples are needed to construct one bit. However, some of the edge samples can have energy from the previous one and thus leading to one more sample on a certain level. If the first part of the bit construction is being checked, it is not necessary to count until 6, being only necessary to detect the transition from low to high or high to low between consecutive samples. However, if the second part of the bit construction is being checked, it is necessary to have a counter which should count 6 samples and then stop if the transition does not occur before the six samples. If the transition occurs, the bit creation should stop on the last sample of the same level.

5) Decoding Phase: Finally, after the reconstruction of the message, a reduced decoder of the Dump1090 is used. This decoder will only contain information regarding the ADS-B message. Besides, the first thing to check is the CRC of the message. If the CRC is not correct, the message should be automatically dropped so that the processor can save resources for other functions like Housekeeping, Communications or Telecommands. The rest of the decoding phase is based on the

Dump1090 and will allow to gather the necessary information for mission purposes. This decoder can identify and decode each one of the fields that come inside each ADS-B message described in the mode-s website [12].

6) Filter's Planning: For the correct planning of the filters, the analysis of the different messages and different metrics is essential. The idea is to understand how to interconnect the different filters to get the several performance parameters of the mission.

Firstly, it shall be clear that a filter by time and a filter by location will always need to be present. Those filters will allow to reduce the number of received messages, making it easier to identify the ”Cone of Silence” of the antenna. Also, for the POD measure, a database of ICAO addresses shall be used. This will allow storing the new ICAO addresses coming inside the ADS-B messages to understand how many different aircraft have sent an ADS-B message.

Those filters will allow acquiring consistent results that can be correlated with the messages received in the same time frame and over the same area by ground stations. The only thing already defined is the area of interest, which will be around the Azores, Portugal. In this area of interest, it is expected that in a 10-minute mission, a medium of 20 aircraft will be present at the same time inside the respective area of interest. This will represent a number of around 40.000 ADS-B messages sent by plane over that period.

The filters shall not be erased when a shut down to the system occurs, and thus a non-volatile memory shall be used. In the case of the payload, the Flash memory shall always be used to retain the active filters so that on a power down, the data is stored.

The payload system needs to have a couple of functions that allow users in the ground to manage the different filters. For this purpose, it is crucial to have a defined API so that users on the GS can set up the new filters without any confusion. From the ground station, it shall be possible to send telecommands to get the list of filters, to add a new filter or to reset the filters.

7) Testing bench: ADALM-PLUTO is a SDR active learning module that is capable of simulating the transmission of an ADS-B message being sent from an aircraft. GNU-Radio is a free software development toolkit that provides signal processing blocks to implement software-defined radios and signal-processing systems that will be used to generate the respective ADS-B message. Furthermore, with a well formatted ADS-B message, whose output is already known, it should be possible to validate all the receiver chain process.

When the first test is fully functional, a second test shall be done with the antenna and the payload board, which includes the digital and receiver modules shall be used. The idea is to correctly set up the antenna and receiver and check the results of receiving real messages from several aircraft.

In the earth surface, the ”Cone of Silence” generated by the antenna is not going to be the same as the one expected in the LEO. Here, the simulation of the real test in the earth surface means that the satellite is not moving and only the velocity of the plane will be the constraint for determining the ”Cone of Silence”.

VI. Performance Evaluation

A performance evaluation is used to determine how a system performs in terms of responsiveness and stability under a particular workload. It also validates, measures, investigates and verifies other quality attributes of the system, such as scalability, reliability and resource usage.

A. ADALM-PLUTO testings

A couple of tests using the ADALM-PLUTO have been thought as they are done under a controlled environment. These tests serve to verify the correct functioning of the receiver chain. To correctly set up this test, it has been necessary to use an RTL-SDR to receive a couple of real ADS-B messages that serve as test messages to be sent from the ADALM-PLUTO to the payload system.

After the correct reception of those messages, an Octave algorithm has been used to transform the Hexadecimal code of the message (Ex: 8D49528899045E9CE84E0F3460B3) into a .dat file that will be read by the ADALM-PLUTO.

To correctly send ADS-B messages and the respective noise between two consecutive messages, a new file only with noise values has been created. A python algorithm has been used to read the ADS-B and noise files, being the ADALM-PLUTO responsible for reading these files and sending the correspondent signals, as they are the source for the ADS-B message and noise. However, the ADALM-PLUTO system needs a brief period to start sending the values that will correspond to the ADS-B message. Because of that, extra noise must be sent first. The python system has a defined number of messages that will be sent, and the time frame of noise samples can be adjusted.

The first test consisted of sending the example message 8D49528899045E9CE84E0F3460B3 and check if the Payload can detect and decode the message correctly. The produced result has revealed that the received message was equal to the test message sent by the ADALM-PLUTO. Moreover, this was the first time that the receiver chain has revealed to be correctly working.

The TX output of the ADALM-PLUTO is directly connected to the receiver of the payload. Between these, a 40 dB signal attenuator is used to reduce the signal power, as the RF receiver frontend is designed for signals with less power, as the received power signal at 400 km of orbit distance is not equal to the ones received at 10/20 km correspondent to the Earth surface.

The first test performed was based on sending equal messages with a defined time interval and check how many the Payload was capable of receiving and decoding, confirming if the time frame period between messages is relevant to the test purpose. The results of this test are shown in Table II. These results serve as a basis comparison for the evolution of the algorithm. Table III shows the results of using filters to detect ADS-B messages using the improved algorithm. Table IV shows the results for the mission measures with the received messages.

	Test 1	Test 2	Test 3	Test 4
# Messages Sent	45	45	45	45
# Correct Messages Sent	43	43	43	43
# Reconstructed Messages	32	31	29	31
# Received Messages	24	25	21	26
Percentage of Messages Decoded	56%	58%	49%	60%

TABLE II
FIRST TEST RESULTS WITH ADALM-PLUTO SENDING ADS-B MESSAGES

	Test 1	Test 2	Test 3	Test 4
# Total Messages Sent	450	450	450	450
# Total Velocity Messages Sent	213	213	213	213
# Messages Decoded	287	284	299	287
# Filtered Messages by Location	96	99	101	92
# Filtered Messages by TC	96	99	101	92

TABLE III
FILTERED MESSAGES BY LOCATION AND TC(VELOCITY)

	Test 1	Test 2
# PTA	100%	100%
# POD	67%	67%
# PTI	82%	69%

TABLE IV
MISSION MEASURES FOR THE SAME AIRCRAFT

B. Algorithm Performance

The test for the Double buffer with improvements in functions has been done for all the different available optimization levels. Figure 10 shows the average time to process a buffer, the number of decoded messages and the number of received messages for each one of the optimization levels.

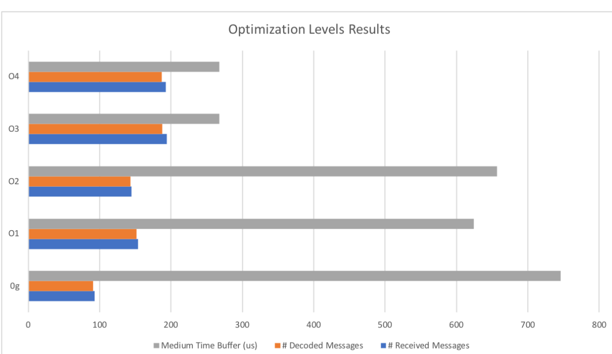

Fig. 10. Several Optimization levels for the ADS-B receiver

The result of the tests revealed that -Og and -02 optimization could not be used to meet the buffer requirements, as there is no available time using this optimization levels to process an entire buffer as they need more than 681 μs. For the -01 optimization, there is available time to process the buffer. However, with this optimization, there is no time to do any other tasks or functions. The optimization levels -03 and -04 produced the same results, and this optimization levels allows to vectorize loops and uses function inlining, which increases the speed of the executable but also increases its size.

C. Energy Consumption

In terms of energy consumption, the power budget document declares that the average consumption of energy while working on about 15% of the orbit time shall not exceed 150mW. On the peak, it should not be above the 1000mW. The orbit of the ISTSat-1 will take around 92.7 minutes with 36 minutes in eclipse. In a standard orbit, the payload can be working on a maximum of 57 minutes. Furthermore, the payload mission will only take a maximum of 10 minutes which allows an understanding that it will only work in maximum consumption for about 11% of the working time. To reduce power consumption, it should be possible to power off the HSADC and the DMA, which will allow reducing significantly the power consumption. Also, the frequency of the payload cores shall be decreased as it is not necessary to be in the 204Mhz frequency. The decrease in the frequency will allow saving energy consumption on the payload system.

Table V shows the consumption for each one of the tests.

	Consumption
ARM Cortex-M4 running	627mW
ARM Cortex-M4 running + DMA	673mW
ARM Cortex-M4 running + DMA + ADCHS	685mW
ARM Cortex-M4 + ARM Cortex-M0_APP running	674mW
Triple-Core running	687mW

TABLE V
ENERGY CONSUMPTION FOR THE SEVERAL CORES AND PERIPHERALS

VII. CONCLUSION

The presented work proposes a design solution in order to overcome the problem of receiving and decoding samples at the same time in the Payload subsystem of the ISTsat-1. The proposed solution presents a couple of tests that proves the feasibility of the code and thus leading to understand that the chosen ARM Cortex-M4 shall have the ability to perform the decoding and acquisition of samples and still have time for other processing. There are good perspectives of receiving and decoding the several messages in the LEO orbit which are the expectations.

REFERENCES

[1] ESA, 'Fly Your Satellite Program, October 2018. [Online]. Available:https://www.esa.int/Education/CubeSats_-_Fly_Your_Satellite/Fly_Your_Satellite!_programme#description.

[2] 'ISTsat-1 Mission Description Document', José Freitas and Rui Rocha and Pedro Gameiro and Alexandre Silva, October 2017. Internal Project Documentation.

[3] Flight Safety Foundation, 'Benefits Analyses of space-based ADS-B', June 2016.

[4] Boeing,'New Air Traffic Surveillance Technology', 2010. [Online]. Available: http://www.boeing.com/commercial/aeromagazine/articles/qtr_02_10/2/

[5] T. Delovski and K. Werner and J. Bredemeyer,'ADS-B over Satellite - Global Air Traffic Surveillance from Space', 2014.

[6] Iridium, 'Iridium to Revolutionize Global Air Traffic Surveillance With the Launch of Aireon(SM)', June 2012. [Online]. Available: http://investor.iridium.com/releasedetail.cfm?ReleaseID=684218.

[7] Iridium, 'Hosted payloads of other customers on Iridium NEXT'. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/i/iridium-next

[8] Tomás Almeida, 'ADS-B Antenna Design Definition', Revision A, September, 2017. Internal Project Documentation.

[9] José Freitas and Tomás Almeida and Rui Rocha, 'ADS-B Payload Design Definition', Revision A, September 2017. Internal Project Documentation.

[10] Salvatore Sanfilippo, 'Dump1090'. [Online]. Available: https://github.com/antirez/dump1090.

[11] Alexandre Silva, 'ISTsat-one Control Protocol', Revision 6, September 2017. Internal Project Documentation.

[12] Junzi Sun, 'The 1090MHz Riddle'. [Online]. Available: https://mode-s.org/decode/.