25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

LOW COST REAL TIME ROCKET TELEMETRY SYSTEM DESIGN

Gabriel de Almeida Souza
Glênio Simião Ramalho
Larissa Silva Barbosa
Alexandre Zuquete Guarato
Federal University of Uberlandia, Brazil
gabrielsouzaworking@gmail.com, gleniosramalho@hotmail.com, larissa.barbosa@ufu.br, azguarato@ufu.br

Abstract.*This paper is concerned with embedded avionics and telemetry design for a model rocket. The authors approach aiming to build a low cost, reliable and efficient system capable of satisfying EPTA (Propulsion and Aerospace Technology Team) mission requirements. A avionics module was designed to contain the electronic instruments used to document the launch and provide support to the recovery system. Furthermore, to perform safely under diverse constraints, redundant mechanisms were designed. The system has redundant function sensors for controlling the recovery system and two ways for storing data. Aboard the rocket, based on centralized processing, the avionics unit is responsible for managing sensory data, more precisely velocity and rotation rates, GPS position, pressure, temperature, tactile and time coming from many peripherals. The flight management system must process, store and send information to the ground control. Included in the processing there is control of peripherals and recovery system status. The embedded manager actuates the recovery system when one of many thresholds is triggered, including displacement, pressure drop and attitude angle. The in-land control must capture and process in real time the received data, applying data processing techniques, generating reliable tracking. A command may be sent from ground in case the mission supervisor perceives danger, to activate the parachute. After mission termination it is necessary to analyze the model rocket performance, by evaluating flight data. This information can help further develop the systems.*

Keywords: *Avionics, Data Processing, Real Time Systems, Rocket Design*

1. INTRODUCTION

Developing aerospace technology is a challenging and expensive activity. Such fact is due to the abundance of critical parameters that evolves high altitude flights. In these regards, the embedded avionics have important tasks: gathering, transmitting and saving data to actively control flight efficiently and safely (Collinson, 2003). The cost of such systems is usually high, generally about 30% of total expenses (Collinson, 1996). This shows the importance of such mechanisms and justifies the studies put into reducing costs, as this is vital for improving the viability of aerospace systems.

On such regards the objective of this work consists in the development of real time wireless telemetry for model rockets, made out of low cost parts. This system consist in two modules: The embedded one, that is capable of operating in all the different conditions of the rocket's flight (launch, upward motion, apogee transition, fall and landing), transmitting and saving the collected info; the other model being the ground control system (Jovanovic and Starcevic, 2008), that receives the broadcast information, processes it and prints in an intelligible form for the ground staff observing the flight.

More specifically, the model rocket is for a 2 km apogee mission, so all systems and protocols are designed taking this in consideration. Such initiative is for application in the rockets developed by the Propulsion and Aerospace Technology Team (EPTA) that is a team attached to UFU (Federal University of Uberlandia), whose main goal is to develop knowledge and technology in rocketry for its members and community.

The design of a telemetry system takes many parameters into account and many convoluted steps until its completion. First, it is fundamental to establish what state variables are necessary to track and which systems need active actuation. Next, a high-level functional model is built to observe what kind of logic and hardware might be required, given the systems needs. In sequence, it is inspected what hardware is available and if they satisfy specific parameters, such as range and linearity. It is very important to observe external constraints, such as space (as in rocket module available space), monetary and computational costs, availability/exchangeability when developing the system and circuitry. When designing the logical and functional characteristics safety and redundancy are keys in producing a robust system.

Hardware choice is only a part in the design. It also requires peripheral communication protocols and process management supports for basic functioning. Interfacing software, between the transceiver device and computers is very important to correctly send and interpret data. Inside the process manager, decision making takes place in the form of actuator command and activity priority. Data filtering is important to generate more accurate derived attributes. The gathered information must be saved for future analysis in an ordered data bank. The design process is highly convoluted and non-linear, so the engineering teams must communicate their needs and work progress to build a functioning system.

On the next topics it will be discussed the requirements of the mission, the subsystems logical design and the material parts chosen to complete the task. Next, the integration between such items will be shown followed by the CAD (Computer Aided Design) builds for implementation. Afterwards the strategies of data gathering, decision making, transmission and processing are discussed. Then, in the conclusion, the difficulties for implementation are explicit and additional development/future works ambitions are analyzed searching for better techniques and lower cost designs.

2. SYSTEM REQUIREMENTS

In Figure 1 the processes and protocols that take place during mission are presented. The mission consists on assembling and preparing the rocket for launch, so when launched, the avionics systems can capture data, to document the flight and take decisions. The flight manager must transmit data in real-time to the supervising team and activate recovery. When the rocket lands, post-flight protocols must be followed to safely recover the rocket.

To satisfy the system needs, operational logic must be developed in order to choose the adequate instruments. Such instruments requirements are: Need to recover information from the flight (acceleration, temperature, attitude, pressure and more), save it in a non-volatile memory, broadcast it in real time to the mission ground control and activate the recovery system actuator. From the basic requirements, it can be inferred that a set of sensors and data managing devices will be necessary inside the rocket, as well as a wireless transceiver device to communicate with the base. The base needs a receiver and a computer. More details on requirements can be observed on Botero *et al.* (2017).

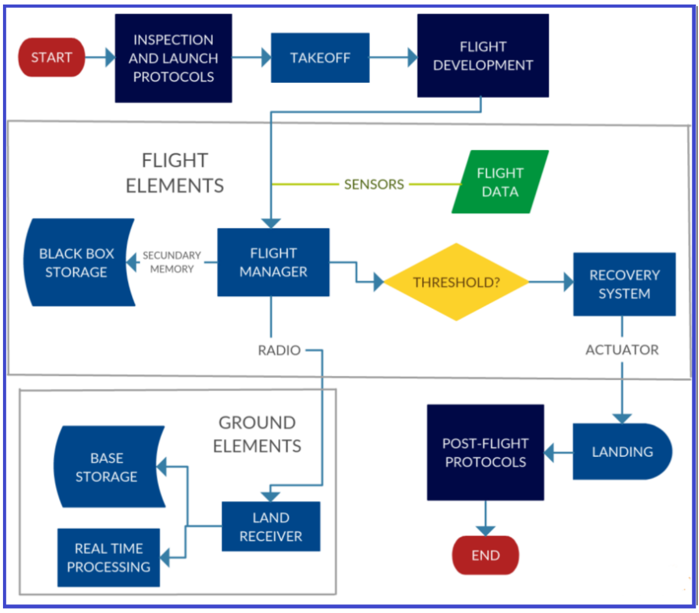


Figure 1: Processes and protocols

The first step is to follow the standard procedures, such as inspecting and calibrating components and preparing to fire the rocket. Then the system must be ready to meet the mission requirements. During flight the management system will receive, send, save and process sensory data. If one of the safety conditions is reached the recovery system is actuated to deploy the parachute. When landing it is necessary to follow post-flight protocols, priming safety and integrity. The detailed internal description of the protocols are currently being developed, following NAR (National Association of Rocketry), NASA (National Aeronautics and Space Administration) and other international and local organizations guides.

There are other very important requirements that the chosen items need to respect, low cost being the guideline one. All sensors, controllers and actuators use are constrained by performance parameters. Some can easily be established, like range and compatibility. Others require deeper studies, as is latency and sampling frequency. Most information can be easily obtained from the manufacturer datasheet and be experimentally tested.

3. MATERIALS

Aiming to achieve a robust and reliable system, the electronics were determined to attend the mission requirements. Most of the hardware used is specified in Figure 2. The components are all easily obtainable and have competitive price. The items were chosen to fulfill instrumental requirements or to give parametric support, like the battery and converter.

An IMU (Inertial Measurement Unit) is useful for spatial tracking, as it possesses in-built accelerometer and gyroscope among other (processing) functions. The Barometer will return absolute pressure and temperature that can be directly related to altitude, using a explicit formula from the datasheet. The transceiver unit is a radio system with sufficient range, noise immunity and transmission rate. The other items functions should be obvious, otherwise they are further discussed in other sections. The manufacturing processes, assembly and transport costs are not discussed.

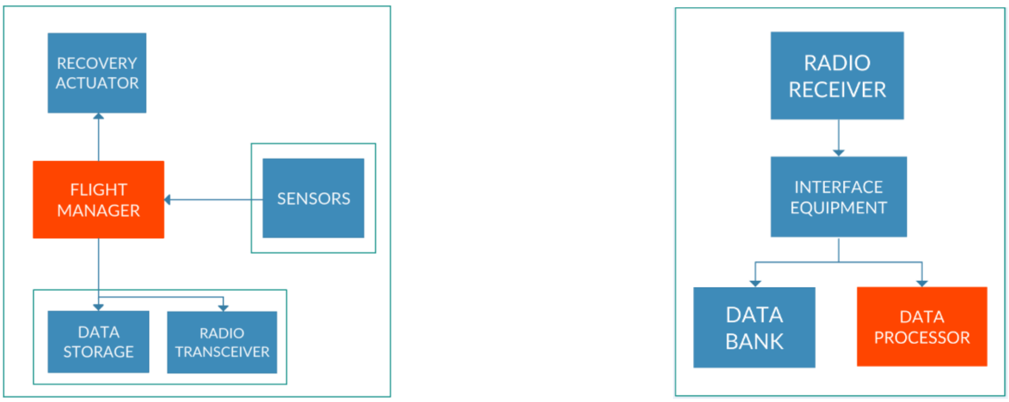
Funtion	Name	Cost/unity(R$)	Quantity
Battery	12V Li Ion	60,00	1
Voltage Converter	Stepdown Converter	9,90	2
Control Unit	Arduino MEGA 2560	50,00	1
Localization	NEO-6MV2 GPS	47,50	1
Communication	RF LoRa 1276	56,50	1 pair
Inertial Measurement Unit	MPU 6050	10,20	1
Barometer	BMP 180	11,25	1
Storage	SD card	4,55	1
Actuator	SG90 servomotor	16,60	1
Circuit Board	Phenolite Board 25x15 [cm]	10,35	1
Ground Interface	Arduino UNO	25,75	1

Figure 2: List of components and prices

4. SUBSYSTEM DESIGN

The telemetry system is composed by two parts that communicate with each other, the embedded one and the ground control. Their integration is through the radio devices that transmit select data from the rocket to base. The proposed paradigm is a wireless real-time telemetry system for documenting and actuating in the rocket model trajectory similar to Wright and Christensen (2006) design.

The components should function consistently throughout the trajectory, but the system must be robust enough to operate in case some peripherals fail. The mission control must be composed of two macro-systems, the embedded components and in-land control as shown in Figure 3. In Figure 3a the necessary avionics subsystems are presented, being: Power and Warning systems for energy and detecting installation problems; Data transmitting by radio and inbuilt storage; Data acquisition through sensors; Recovery actuator; and a central microcontroller that represents the flight control manager and integrate the components. In Figure 3b there is the ground control, composed of a radio receiver, interface apparatus and a processing, storage computer. This way, the base staff can observe the flight evolution in real-time, save data and to intervene in the rocket if deemed necessary.



(a) Embedded Subsystem (b) Ground Control

Figure 3: Avionics and Telemetry macro-systems

4.1 Embedded Avionics and logic

According to Watkins and Walter (2007) there are advantages in working with a modular paradigm. The method used in this avionic system is to centralize the decision making into the microcontroller.This type of build simplifies the implementation of centralized logic and is more practical to realize than distributed tasks on a system of this magnitude. As there are many sensors and peripherals it would be more challenging to implement a federated system, as the dispersed tasks would be harder to serialize and maintain consistent.

The embedded avionics is responsible for the correct functioning of a flight system (Moir and Seabridge, 2008) through real-time processing. Figure 4 shows how information is gathered and managed inside the microcontroller. The devices used were introduced in Figure 2.

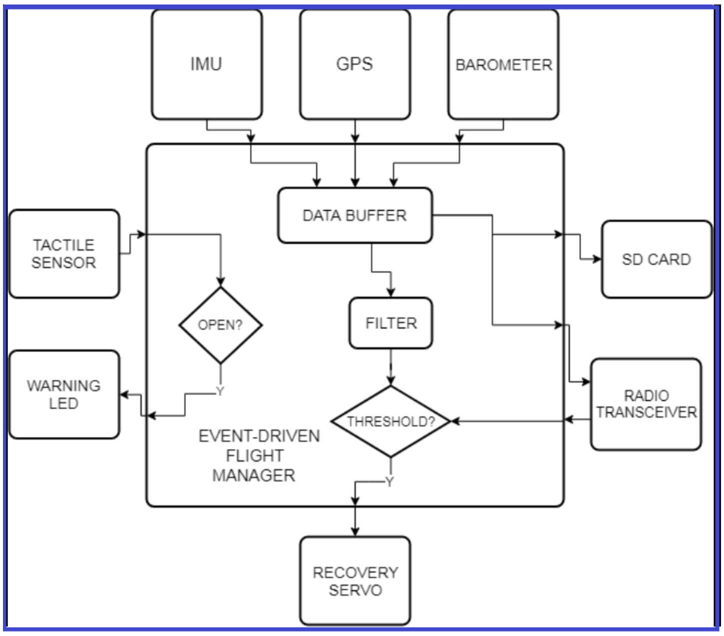


Figure 4: Flight Manager Architecture

The flight management system deals with different sampling frequencies, as the sensors outputs a diversity of information throughout different methods. To function with asynchronous tasks and eventual data loss the system is programmed in a time-driven archetype, where every task is processed in a specified period inside a cycle, unless a flag highlights priority, calling the next activity even if the input data was not received. A prioritized task is recovery system activation, which should be triggered right after the apogee.

This asynchronous operation is modeled as a chrono-triggered finite-state machine (Ericsson *et al.*, 2004), where in each node an action is performed, avoiding collision. The time between each activity is not uniform, but is explicitly programed, depending on the current and next tasks. Interrupts might change the next task, as activating the control servomotor or saving a decision in the SD card.

About the threshold for actuating, it is confronted sensory data to a bounded reference or to previous inputs. The most reliable methods observed in literature (Botero *et al.*, 2017) are related to attitude, altitude drop or time. Attitude can be directly calculated from IMU data, integrating accelerometer data with gyroscope rate of rotation, and if the rocket attitude surpasses a specified number, the recovery system is deployed. Altitude data is derived from pressure and temperature in the barometer, and can be compared to previous inputs to infer about falling. Time can be used in an absolute manner (very unreliable) or in a relative to launch manner, where an acceleration peak occurs, then the deployment time can be calculated through theoretical models and reliable software. The input data is processed using a moving average filter.

The system is also robust to instantaneous power/manager failures and component malfunctions. There are three sensors for gathering data, two ways of storing info and a WatchDog timer is used to in-flight reboot the manager if a task freeze happens. The relative timer functions and total displacement could also be used as redundancy for the threshold and the radio transceiver can receive data from the ground control to deploy recovery.

4.2 Ground Mission Control

The ground control system is responsible for receiving raw data and storing it in a database, so data can be further explored in posterior studies. It is also designed to decode, filter and present real time information in an intuitive interface (Nardie and Zarmer, 1990). The supervisor may observe the navigation development to incur about tracking problems and can decide to send a command to actuate on the rocket recovery system, as the system supports half-duplex communication. This manager must possess heuristic knowledge and also be supplied with highly comprehensible data in order to avoid risks (Jovanovic and Starcevic, 2008).

To communicate with the embedded rocket system, the base is composed by a LoRa transceiver, an Arduino UNO and a notebook. The LoRa transceptor is used as a receiver to get flight data, and can be used as a transmitter to activate recovery, similar to Wright and Christensen (2006) approach. The techniques used for identifying data are described in section 6. The Arduino is an interface to access the radio, as it is simpler to connect the radio transceiver to it, rather than directly to the final computer. The notebook is the final element, where data is stored, processed and displayed in real-time.

5. AVIONICS HARDWARE INTEGRATION AND FIXATION

This section discusses about how the hardware is connected, integrated and fixed into the rocket. To properly connect the components it is necessary to observe about each individual item needs and constraints, then it is necessary to assemble in a circuit board to be fixed in the rocket, through a module. Optimizing about space and weight is also important for the rockets overall performance.

5.1 Parameters and Constraints

Every component has voltage and power requirements for regular operation. In Figure 5, the components are listed with their electrical parameters and the total demand on the microcontroller and battery in flight operation. The values for the actuator are peak (the standby is very inferior). The devices powered by the microcontroller are the inertial sensor, the barometer and the SD module.

Function	Name	Vin [V]	Ic [mA]	P [mW]
Controller	Mega2560	7.0 - 12.0	26	200
Localization	NEO-6MV2	3.3	100	330
Radio	LoRa1276	3.3	120	400
IMU	MPU 6050	3.3	4	13
Barometer	BMP180	3.3	1	4
Secondary Memory	SD module	5.0	25	30
Actuator	SG92R	5.0	53	262
Buck Converter	TSR12433	4.5 - 23.0	-	-
Total Controller	Mega2560	5	-	60
Total Battery	9V Battery	-	210	1400

Figure 5: Voltage, Current and Potency parameters

This information was estimated from Datasheets and Botero *et al.* (2017) guide, requiring experimental tests for confirmation. The values provided by this table serve as a reference to know if the system will work correctly, as well as select the appropriate converters and battery. The chosen converter efficiency is informed to be of 92%.

The avionic system is boarded on the rocket, because of that, space is limited. The rocket has other systems, such as propulsion (motor and propellant) and recovery (parachutes) that occupy most of the space, leaving a reduced space for the complete avionic module. In a rocket with 2 m of height and 12 cm of external diameter, the avionics module has an approximate space of 30 cm in height and 10 cm in diameter. Therewith, the PCB (Printed circuit board) with the components has an approximate measurement of 9x23 cm.

Because the system is boarded and of limited size, there is only one battery that will power the entire circuit. But the components differ in their voltage value, therefore, the supply has a common origin and step downs are used on specific components that need a different value. The components also differ in their communication protocols and pin requirements with the controller. The GPS (NEO-6MV2) use serial communication, the radio (LORA1276) and the SD module use SPI communication, the inertial sensor and the barometer use I2C communication. This information can be found in the Datasheets of the components. No heat sink is required.

5.2 Circuit Board Layout

The circuit must respect the characteristics of each component, because there are items that need to be connected to certain ports, as they use a specific protocol for communication. The pinning choice is done to satisfy the electrical

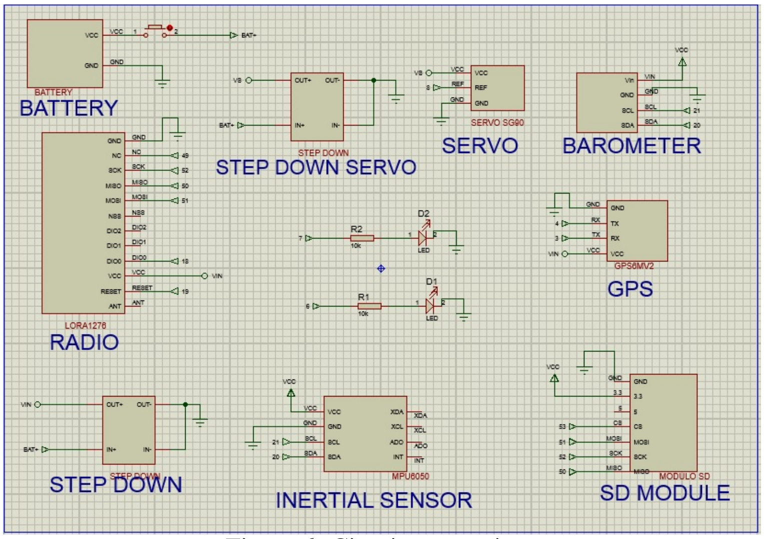


Figure 6: Circuit connections

parameters as well, so some components require feed directly from the power supply. Figure 6 shows the connections between each peripheral and the Arduino Controller, using Proteus software.

The PCB (Figure 7) is the medium that holds the components connect. Most peripherals are directly fixed in its structure, so that the board itself can be set on the avionics module. The positioning and track routing was designed to occupy the least space, and the final board dimensions where defined as 9x23 cm.

Most components are connected through the PCB. Some peripherals where left outside of the board because they require to be put in other specified locations to exert their function, but still all peripherals answers to the controller, so external wiring is used in those cases to connect the pins. The servo and its buck converter are external to the avionics module, being placed in the recovery module to trigger the parachute deployment system. The battery is fixed directly above the module, so that it can be placed on a centralized position, to avoid imbalance. Buttons are used for contact reboot, by the personnel, and as a reference for module fixation.

Because of the constrained size of the board, the PCB became rectangular and the components were fixed close together, so some jumpers were needed to complete the connections. Another note is that the Arduino MEGA will not be soldered on the board, so in Figure 7 it is represented in pink and the other components in blue, because they are on opposite sides.

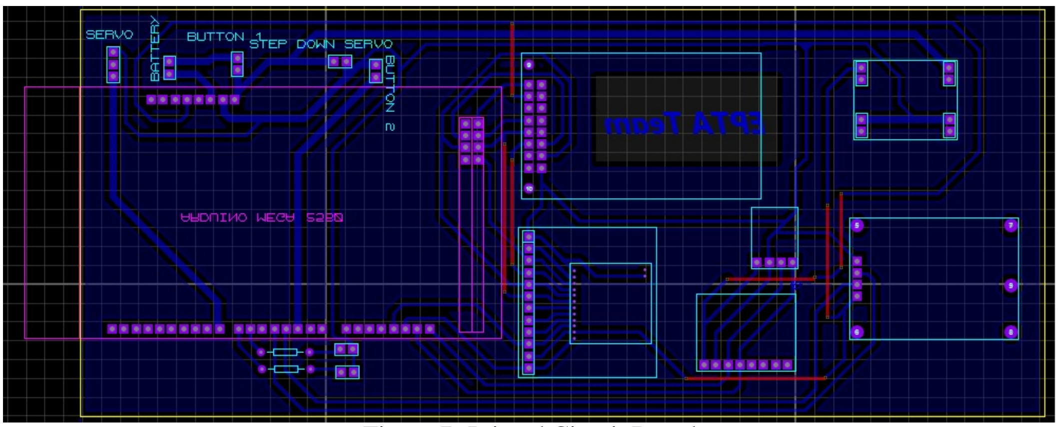


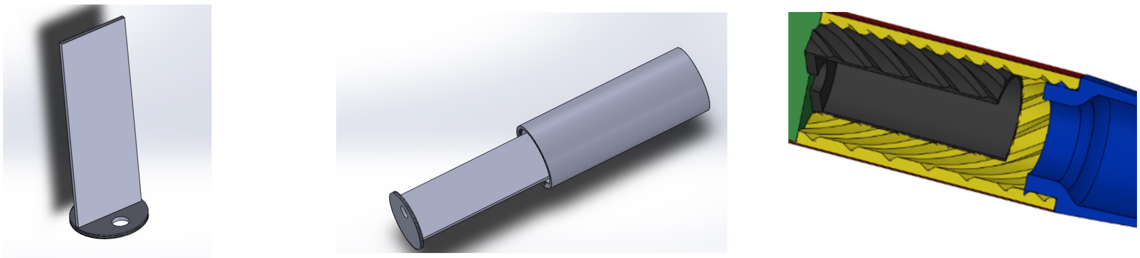
Figure 7: Printed Circuit Board

The total mass in the components is a very important information to be fed for the other project divisions. The battery has a mass of 200g, the step down has 4g each one (two were used), the controller (Arduino MEGA) has 35g, the GPS has 18g, the radio has 15g, the inertial sensor has 2 g, the barometer has 2 g, the SD module has 8g, the servo motor has 9g and the board has 15g. With this, the total mass expectation for electronic components is 312 grams, excluding wiring and fixation. In practice the module itself has mass and the module inertia need to be balanced regarding its center of gravity, so these problems are addressed in the module design subtopic.

5.3 Module Design

SolidWorks is the design software of choice for projecting and calculating total mass of the module, whole main design characteristics are represented in Figure 8. The coupling structure of the avionic system on the rocket consists of an enclosed cylinder (Figure 8a) with the system embedded in it. The board will stand vertically, fixed to the internal rectangular part of the structure (Figure 8b), which is then closed with the upper part, screwed into the lower part, forming a cylinder which has some holes for the output of wires for external connections and components.

This structure was chosen to facilitate the construction of the board; a rectangular plate is simpler and lessens the risk of errors and failures because it is a single plate, comporting most items. In addition, the plate will be fixed and closed inside the module, becoming isolated from the other systems and closer to the vertical inertial axis, reducing imbalance and vibration. The closed module will be docked into the rocket body, using a wall interfacing screw, similar to Figure 8c, where the darker cylinder is the module hull. This fixation technique was chosen because it is very simple to realize and reduces the amount of nuts and screws used in the rocket.



(a) Rectangular plate for the PCB (b) Module Assembly (c) Internal structure wall screw

Figure 8: Module design and fixation

The total mass of the module is calculated as 941 g. The total estimated launch mass for the rocket is 20 kg. So the mass ratio is 4.7%, which appears to be a small fraction of the rocket launch inertia. However, rocket stability is a design priority (Botero *et al.*, 2017), which explains why so much care is put into reducing imbalance.

6. COMMUNICATION AND ARCHIVING

This work real time processing approach is derived from Mhatre *et al.* (2015). To satisfy the objective of collecting and processing data in the ground base control it was developed a Graphical User Interface (GUI) on JavaSwing, a java toolkit that can be used to create applications with powerful components (Eckstein *et al.*, 1998), connected to Arduino IDE that works receiving data strings from the serial port, saving it automatically on a database, built on MySQL. The GUI exposes data in a specified vector-sized window, so that only the most recent information is observed in real-time.

In Figure 9 the ERD (Entity Relationship Diagram) is presented. The received data and the associated time, as well as the instructions given to the other peripherals are necessary because the knowledge of deployment time and data input is critical. The non-sensory peripherals were modeled as attributes as they do not provide active data during the mission. This model relates the logical process, the peripherals and the sensors through periodic instructions, so all info related to a command should be grouped for easy query. Relevant derived information can be stored in the same structure.

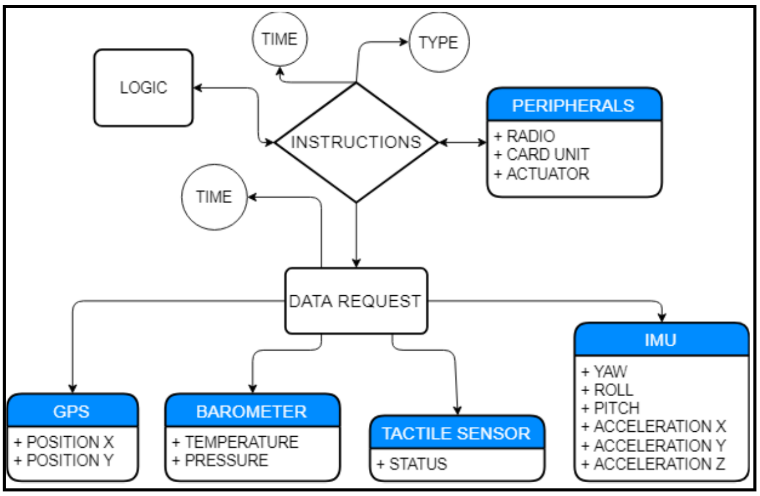


Figure 9: Entity Relationship Diagram for raw data as attributes

In relation to the source code of the GUI it is used a series of java libraries that allows serial communication with the IDE, the creation of the graphics and the user interface, the last one made with JavaSwing, a library for graphic interface. To save the received data it was chosen the database software MySQL, an open source robust database application which has a better organization and reading/writing rate compared to other competitive tools (Reese *et al.*, 2002).

7. PROCESSING AND FILTERING

According to B Liang (2010) and Menegatti and Molin (2004) algorithms for data filtering are a very important element in data processing. Considering that there is a significant amount of noise among the collected raw data, predicted in the sensors and external factors, sensor fusion is to be implemented to reduce bias. The necessity for treating data to obtain reliable results is factual.

In the telemetry, a basic moving average filter is used in the barometer and IMU data to increase reliability for actuating the recovery system, as the sensors information is prone to corruption. This method reduces the chance of occurring a catastrophic failure, that is the premature recovery activation.

To correctly capture and identify attributes a method is applied. Characters are used as prefixes for radio transmission, so a 'p' can be used to identify pressure data, a 't' for temperature. It means that in case a packet is not fully received, the captured attribute can still be used properly, and it gives freedom for the order and data to be sent in each package. The values are also bounded, so if an obviously absurd value is received, it is inferred about its corruption and is rejected.

More complex sensor filtering, integration and fusion are not being currently employed, bu EPTA plans on applying methods as described in Kok *et al.* (2017) to explore the IMU potential. In Park *et al.* (2009), inertial sensors are fused with image landmark tracking to produce better results. In our current rocket design, the IMU information could be complemented by the GPS, barometer and dynamic models, for future works.

8. CONCLUSIONS AND FUTURE WORKS

There were difficulties associated with the development of both the embedded avionics and ground control systems. In the avionics, because of the constrained space, the circuit board design was challenging, as to correctly place the electronics and the trails, as well as the external components in the module and other parts of the rocket. As for software, for implementing the manager logic, the code had to be organized into several functions and intensively tested. In the ground control, the input data has to be displayed in an intuitive manner, and designing a good GUI is an iterative, time demanding task.

About the flight manager, its current mode of function is less powerful than parallel processing (threading). The problem for its implementation is that true parallel processing can only be performed in multi-core hardware, which is not the case for most microcontrollers, including the chosen Arduino. Operational system support for threads is also convenient and because of the critical nature of the processes a real-time processor is important for reducing latency. A future work of interest is to fully construct an improved architecture in a more powerful platform that can extract the full potential of each peripheral, using RTOS (Real Time Operational System) and parallel processing.

Aiming for a better algorithm for filtering, studies about the dynamic models and sensors specific features are the next fundamental step. The system model helps by filtering data in a predictive fashion and sensory features, such as variance and frequency are important in building the filter. This data is to be confronted with the theoretical model calculated with software that simulates the trajectory, engine temperature, pressure and acceleration like OpenRocket and Ansys to validate and observe the system performance given the external conditions, thus helping to infer about problems.

The system is to be constructed and assembled for launching in November. Until the mission day, all subsystems are to be further tested, and minor design adaptations might be necessary, to guarantee success and improve performance. The progress in software and data processing is to be continuous.

The final device design shows robustness and cost-effectiveness, as initially proposed for the mission. Such properties qualify it for being used on the rockets developed by the Propulsion and Aerospace Technology Team (EPTA). The development team hopes to further improve the rocket systems by applying better build-ups and more sophisticated techniques. Then, the system is to be mission tested, so that the weak points in the design may be exposed, exhibiting the new work focus and required improvements.

9. ACKNOWLEDGEMENTS

This paper is dedicated to UFU engineering community, our friends and family and our colleagues at EPTA, in special Felipe Ribeiro, for being a very committed president. We thank MAPL, CEPAERO, Prof. Jean Tavares, Prof. Pedro Assis and Prof. Mauricio Escarpinati for their support.

10. REFERENCES

B Liang, Q Yongyuan, Y.G.L.P., 2010. "On integrated navigation for vehicle dead reckoning".

Botero, A.Y., Rodriguez, J.S., Serna, J.G., Gomez, A. and Garcia, M.J., 2017. "Design, construction and testing of a data transmission system for a mid-power rocket model".

Collinson, R.P.G., 1996. *Introduction to Avionics*. Rochester.

Collinson, R.P.G., 2003. *Introduction to Avionics*. Boston.

Eckstein, R., Loy, M. and Wood, D., 1998. *Java Swing*. Denver, USA, 1st edition.

Ericsson, C., Wall, A. and Yi, W., 2004. "Timed automata as task models for event-driven systems".

Jovanovic, M. and Starcevic, D., 2008. "Software architecture for ground control station for unmanned aerial vehicle".

Kok, M., Hol, J.D. and Schon, T.B., 2017. "Using inertial sensors for position and orientation estimation". doi:http://dx.doi.org/10.1561/2000000094.

Menegatti, L.A.A. and Molin, J.P., 2004. "Removal of errors in yield maps through raw data filtering".

Mhatre, V., Chavan, S., Samuel, A., Patil, A., Chittimilla, A. and Kumar, N., 2015. "Embedded video processing and data acquisition for unmanned aerial vehicle".

Moir, I. and Seabridge, A., 2008. "Aircraft systems, mechanical, electrical, and avionics subsystem integration". John Wiley & Sons, Chichester, England.

Nardie, B.A. and Zarmer, C.L., 1990. "Beyond models and metaphors: Visual formalisms in user interface design".

Park, R.S., Bhaskaran, S., Bordi, J.J., Cheng, Y., Jhonson, A.J., Kruizinga, G.L., Lisano, M.E., Omen, W.M. and Wolf, A.A., 2009. "Trajectory reconstruction of the st-9 sounding rocket experiment using imu and landmark data".

Reese, G., Yarger, R.J. and King, T., 2002. *Managing and using MySQL*. Cambridge, England, 2nd edition.

Watkins, C.B. and Walter, R., 2007. "Transitioning from federated avionics architectures to integrated modular avionics". In *Institute of Electrical and Electronics Engineers IEEE 2007*. GE Aviation, Grand Rapids, Michigan USA.

Wright, H.K. and Christensen, B.R., 2006. "Engineering a hybrid rocket avionics system".

11. RESPONSABILITY NOTICE

The authors are the only responsible for the printed material included in this paper.