IrishSat Senior Design Proposal

1 Introduction

IrishSat is a satellite-design team with the mission of designing, building, and testing CubeSats to launch using NASA's CubeSat Launch Initiative (CSLI). The CSLI is a program that provides opportunities for CubeSats built by U.S. universities, high schools and non-profit organizations to fly on upcoming launches. Each year, NASA releases an Announcement of Partnership Opportunity with detailed instructions on how an interested eligible organization may submit a proposal for a CubeSat project. IrishSat plans to complete its first CubeSat proposal by the end of the 2022-2023 academic year for submission to NASA in November 2023.

Since the creation of IrishSat in Fall 2020, the team has been developing the infrastructure to build and operate a CubeSat. For example, IrishSat has built a ground station on campus for satellite communication and an air bearing that can test the satellite's orientation system by rotating it on a frictionless surface. Our team plans on designing a 1U CubeSat, which is a 10x10x10cm satellite that typically weighs less than 2 kg. The CubeSat will be launched into low-earth orbit which encompasses the area of space under 1,200 miles from Earth.

2 Problem Description

The CubeSat will operate over a 3 year cycle, but components and subsystems are expected to start experiencing significant failures in as early as 6 months. Additionally, the satellite will burn up when returning to earth, rendering most data unrecoverable.

For the satellite to be successful over its lifetime, we must ensure that it will operate and transmit data despite any malfunctioning equipment. The data transmitted must be adjusted to reflect the state of the satellite and properly modify or ignore any faulty data, for example, from sensors. A haywire sensor's inaccurate data could disrupt and invalidate the operation of multiple subsystems. If this error could be detected, the sensor could be rebooted (in an attempt to recover it) or otherwise ignored. The CubeSat's actuators would then continue to function based on the best data, which would also be stored for future reference.

A subsystem providing proper power allocation to all other subsystems is also needed. Integrating this feature with the monitoring subsystem could optimize the CubeSat's power consumption dynamically.

Transmission will need to be handled by a digital radio that will interface with the rest of the CubeSat's components using another subsystem. This communication subsystem will be responsible for selecting, compiling, and handling data transmission to the radio, which will then handle the modulation and physical transmission.

3 Proposed Solution

In order to properly transmit and receive our data, we need a robust communication system. We already have a software defined radio (SDR) in our possession that we will be using for our communications. This SDR, which will be programmed by the IrishSat Communications team, will interface to our PCB that contains an ESP32 and a Raspberry Pi microprocessor. This microprocessor will act as the flight computer for our satellite and will be programmed by the IrishSat computing team. The ESP32 will serve as a sort of protection component for the flight computer and the overall board. We will have the ESP32 read in the data from all the sensors on the board (the same signals the flight computer is reading). The ESP32, which will be programmed by our Senior Design team, will be used to detect any anomalies in the sensor signals, shutting sensors and entire systems off if they are acting abnormally. Having this protection will ensure the success of our mission as it will prevent any adverse effects to the rest of the system due to bad sensors. The ESP32 will also be used to read out all the data from the sensors during our testing stages. It will additionally communicate with the flight computer through an I2C connection, relaying SDR transmissions, detecting anomalies in the sensors, and notifying of the shut down of any failed sensors.

In addition to connecting with the SDR, our board will also interface with the ProtoSat PCB with a pinstack. We will not be designing this additional PCB, which will contain the subsystems the flight computer intends to control, such as the reaction wheels. To interface with this board, we will route the Pi's designated output pins to the pinstack to make the signals available to the ProtoSat PCB.

4 Demonstrated Features

The first feature we expect to demonstrate is the ability to take commands from and send notifications to the flight computer. This feature will include established codes to send from the ESP32 to the flight computer in the case of each sensor being shut down as well as to receive messages that the flight computer wants to transmit.

The second feature we expect to demonstrate is the ability to passively receive the I2C communication stream between the flight computer and the sensors it connects to to extract the sensor data using the ESP32, which we will demonstrate with a script that outputs all received sensor data to a serial monitor. ProtoSat has also requested this feature for use in debugging and prototyping.

The third feature is the ability to packetize the incoming sensor data and send it for transmission to the SDR. This feature includes packetizing and sending messages from the flight computer to the SDR for transmission as well.

The fourth feature is monitor software that detects anomalies in input sensor data and decides if a given sensor has to be shut down or can continue inputting data into the flight computer. This software should notify the flight computer over an I2C connection if it decides a sensor should be shut down.

The fifth feature is the ability to start a reboot sequence or cut supply voltage to each sensor on command, which we intend to implement using multiplexors. This feature will execute upon direction from the monitor software.

5 Available Technologies

- Sidekiq Z2 SDR is the primary communications platform for IrishSat Proto-Sat and will receive data packets for transmission from the ESP32 (already acquired)
    - Has a breakout board that interfaces with our PCB
- Raspberry Pi based flight computer (IC only)
- IMU
- Digital thermometer
- Magnetorquers
- Digital barometer
- Various surface mount components (resistors, capacitors, etc.; this will become more clear when we physically design the board)

6 Engineering Content

To bring this system to completion it will be necessary to engineer several different subsystems. First and foremost is the overall layout of our PCB, which will contain all other components and subsystems. This will need to be done using the Eagle design suite, most likely in an iterative development cycle as we prototype and test and revise our design. This PCB will need to be designed as well as machined and tested, resulting in a large amount of work over the complete design cycle.

Second, within this PCB design a bias network will need to be constructed for all active components, including our microcontroller, the flight computer, our various sensors, and our SDR module. This will require research and creative thinking to ensure that all components are biased correctly and are able to operate together.

Third, systems will need to be designed to implement monitoring and control of our different components, ensuring that current and voltage levels are acceptable and enabling us to automatically shut off components that begin to draw too much power. This will require both software and hardware work, including implementation of various digital communications protocols between our sensors and microcontroller and some type of algorithm to determine acceptable levels of power. In addition, we will need to develop an algorithmic solution to manage the biasing of our components in order to enable dynamic power switching based on a control signal.

Fourth, we will need to be able to read sensor data in real time and generate packet frames for digital transmission to our SDR module. This will require implementation of communications between our sensors and our microcontroller as well as development of algorithms to ensure that data is correctly packetized and sent for transmission at appropriate time intervals.

Fifth, it will be necessary to develop some type of communication between our microcontroller and on-chip flight computer, to allow for control signals to be sent back and forth and sensor data to be logged.

Finally, our PCB will need to be able to interface with the PCB being designed by members of the IrishSat Proto-Sat squad which will include several actuators that will need to be able to respond to the data read in, processed, and monitored by our microcontroller. This will require development and management of additional digital communication channels.

7 Conclusions

Overall, our PCB will be used as a sort of processing unit for the IrishSat CubeSat. As mentioned, it will contain the Raspberry Pi microprocessor (which will do the bulk of our computations and control work) as well as an ESP32. It will also interface with both an SDR and another PCB that contains the sensors and motors necessary for our CubeSat. Designing this PCB and getting it to operate as intended is crucial to the future of IrishSat as it will facilitate development of necessary Proto-Sat subsystems, such as power distribution, communications, and computing.