

# Low-Cost Satellite Communication for CubeSats

- Joseph Dennison (CompE)
- Robert Goes (CS & Math)
- Taegan Williams (CS)
- Cosette King (CS)

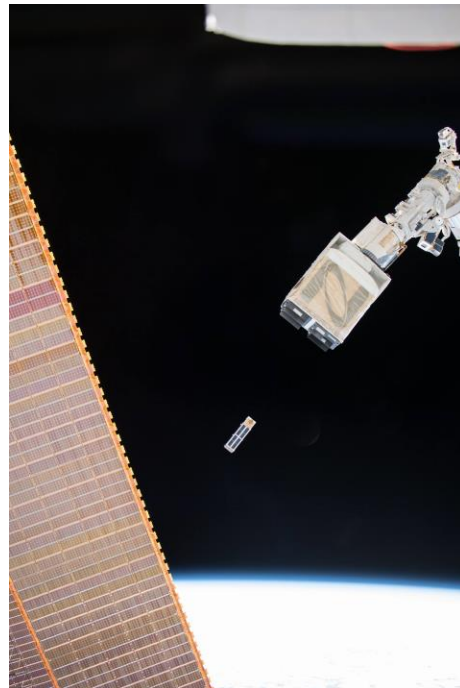


University  
*of* Idaho



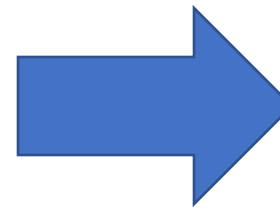
# Background: TechEdSat

- **TechEdSat:** Technology Education Satellite
- **What?:** Investigation to employ small CubeSat spacecraft
- **Why?:** To evaluate, demonstrate, and validate new technologies



# Background: Iridium 9523

- **Iridium 9523:** A consumer voice and data satellite transceiver module (from a satellite phone)
- **What?:** Communicates with the Iridium satellite constellation
- **Why?:** Can establish a full internet connection to a ground network via:
  - SBD: Similar to text messaging
  - RUDICS: Similar to a data plan





- The cost of common satellite radios are normally in the tens of thousands of dollars.
- By switching to an off-the-shelf satellite network, the cost for future NASA satellite missions will be drastically reduced.

# Objective

To develop the software for one of the CubeSat's microcontrollers to establish a live data connection via the Iridium Network for an upcoming TechEdSat mission.

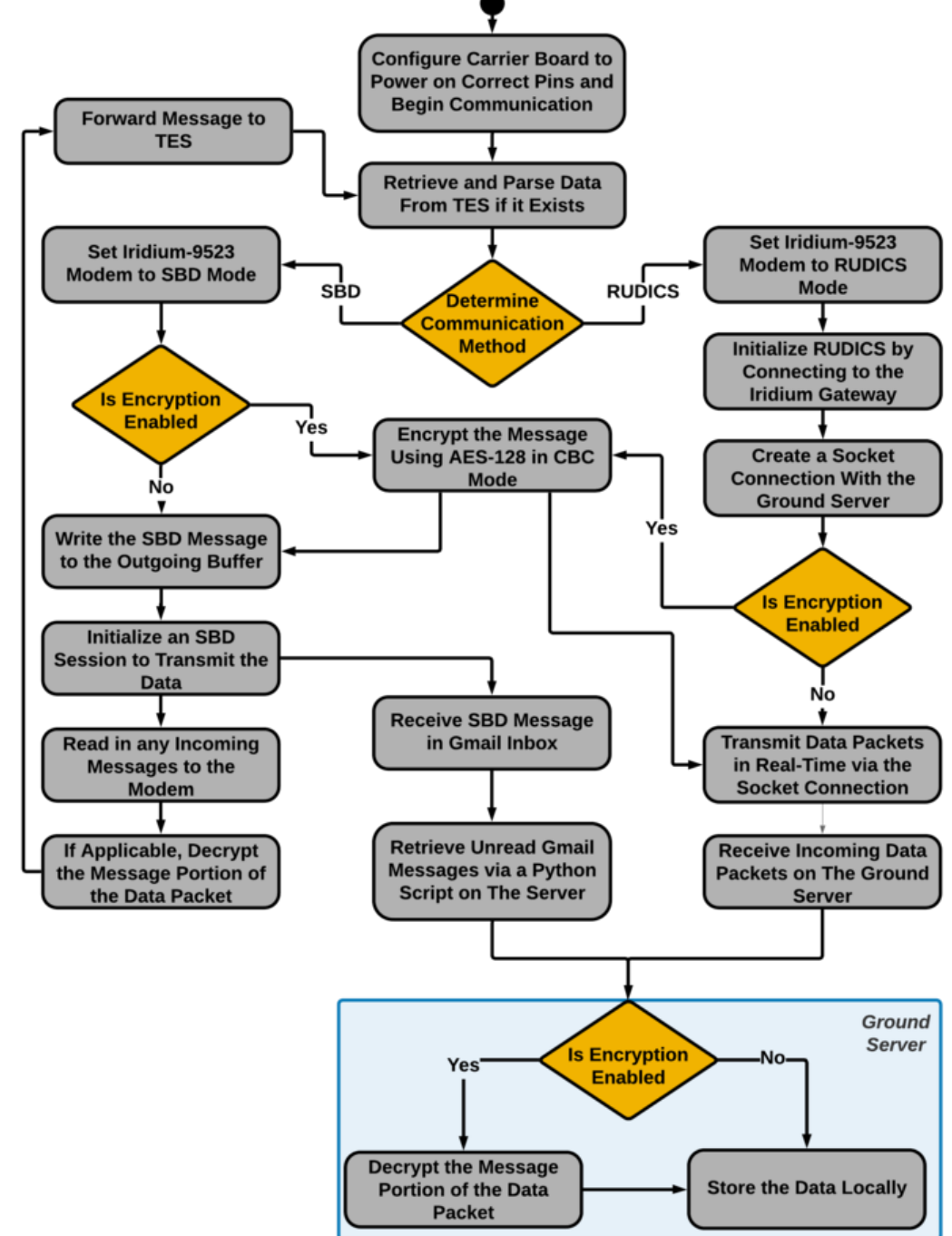
- Components:
  - Firmware Development:
    - Robert Goes and Joe Dennison
  - Ground Server and Encryption:
    - Taegan Williams and Cosette King

# Previous Years

Team SCUBEE (2018-2019): Developed working carrier PCB



FIRE (2019-2020): SBD Functionality with SCUBEE board via Raspberry Pi



# Baseline Project Requirements

- Sample temperature every second and power consumption every 30 seconds.
- Able to turn off at any time if 9523 overheats.
- Send status messages of modem and board.
- Transmit randomly generated data via SBD and RUDICS to a ground server.

# Satellite Firmware

```
include
├── README
├── lib
├── README
├── src
│   ├── main.cpp
│   ├── utility.cpp
│   └── utility.h
├── test
│   ├── README
│   ├── gitignore
│   ├── platformio.ini
│   └── TechEdSat_firmware
├── pio
├── vscode
├── c_cpp_properties.json
├── extensions.json
├── launch.json
├── include
│   └── IridiumSBD
│       ├── README
│       ├── lib
│       ├── README
│       ├── src
│       │   ├── globals.h
│       │   ├── heartbeat.cpp
│       │   ├── heartbeat.h
│       │   ├── iridiumtask.cpp
│       │   ├── iridiumtask.h
│       │   └── main.cpp
```

```
53     TECH_ED_SAT,
54     HEARTBEAT_TASK,
55     IRIDIUM_TASK,
56     TESBUSS_TASK
57 };
58
59 //Types of task commands
60 enum task_cmd_type
61 {
62     //---Commands to the IRIDIUM Task---
63     SBD_MSG_DATA, //a-xxxx Sends a SBD msg of length xxxx
64     RUDICS_MSG_DATA, //b-xxxx Sends a RUDICS msg of length xx
65     SBD_MSG_RDM, //c-xxxx sends a random sequential sbd r
66     RUDICS_MSG_RDM, //d-xxxx sends a random sequential sbd r
67     STATUS, //e-xxxx
68     //---Commands to the HEARTBEAT Task---
69     SHUTOFF, //f- shuts off the device
70     GET_HEARTBEAT, //g- gets modem and board status
71     //---Commands to the TECH_ED_SAT
72     SEND_HEARTBEAT, //h-xxxx sends a heartbeat data structure
73     ACK, //A- acknowledges a transmission
74     //---Commands notifying error---
75     UNKNOWN_ERROR,
76     DATA_PACKETS_LOST
77     //TODO: add more error types
78 };
79
80 //Structure that is placed in a queue to tesbuss task or receive
81 typedef struct task_cmd
82 {
83     task_cmd_dest dest; //Desition of task command IE: TECH_ED_S
84     task_cmd_dest source; //Source of task command IE: HEARBEAT_TAS
85     task_cmd_type type; //Type of task command IE: HEARTBEAT_DAT
86     int id; //Callback of id of task IE: 0, or a num
```

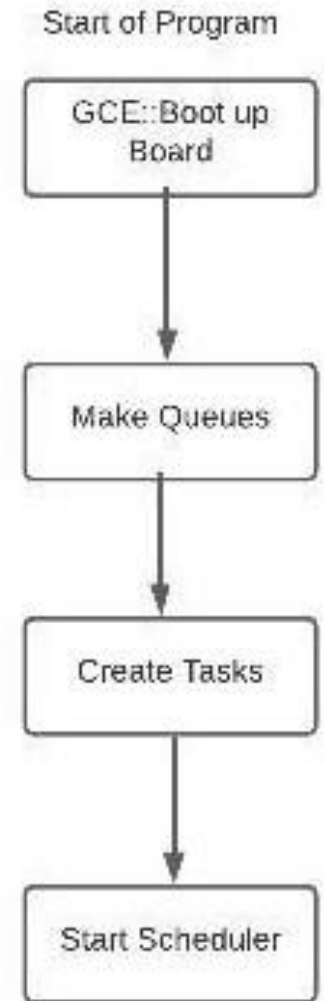


# Software Libraries

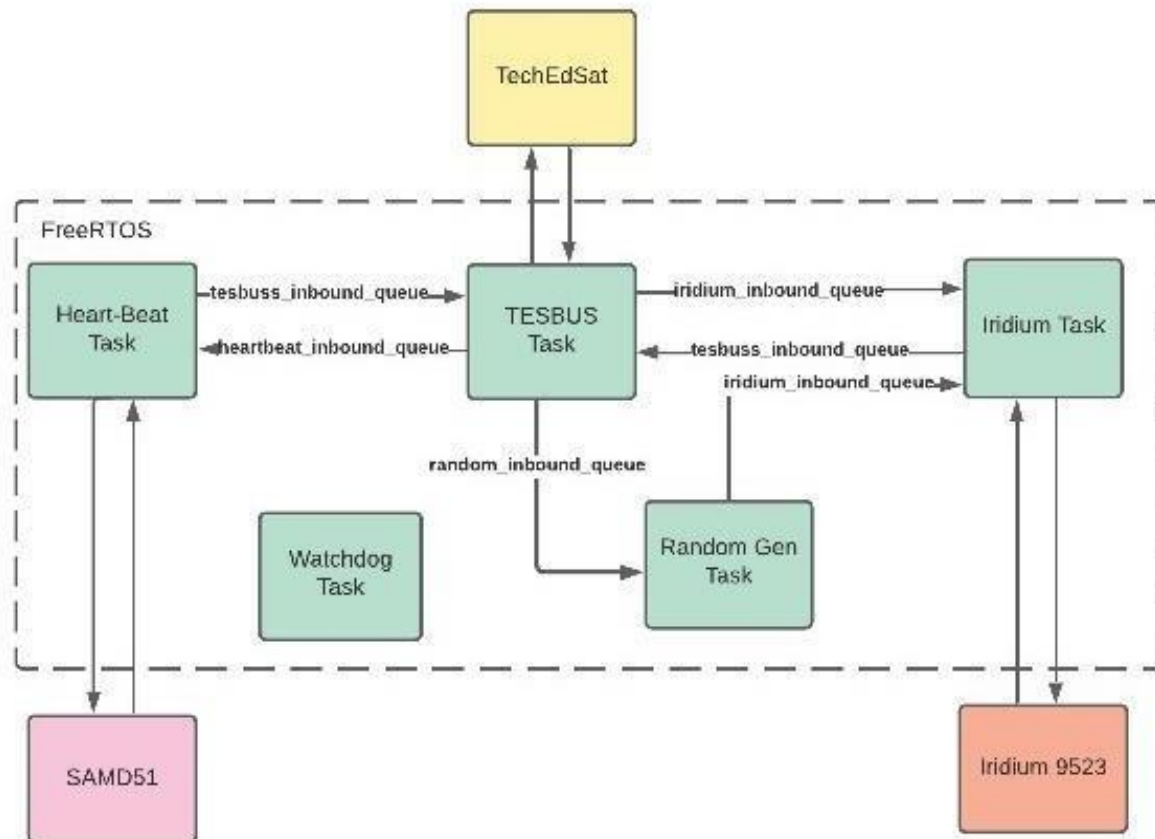
- **GrandCentral:** Board functions such as sensors
- **SerialTransfer:** Serial communication for Arduino
- **IridiumSBD:** Modified to add RUDICS support
- **FreeRTOS\_SAMD51:** Free RTOS
- **SleepyDog:** Hardware watchdog timer for errors
  
- *All libraries are open source*

# FreeRTOS

- Real Time Operating System.
- Like windows, but for microprocessors.
- Allows us to create multiple tasks that run at the same time.



# Tasks



- Heartbeat Task
  - Reports temperature, signal quality, etc
- TESBUS Task
  - Communicates with primary satellite computer
- Iridium Task
  - Controls 9523 modem
- RandomGen Task
- Watchdog Task

# TESBUS demo:

- Custom built testing software
- Sent commands and received responses from microprocessor
- Sent via a USB to serial adapter directly to the pins on the microchip

```
Anaconda Powershell Prompt
(base) PS C:\Users\Robert\Desktop\NASA> python test2.py
----Send Command----
1. a-Send n random bytes via sbd
2. b-Send n random bytes via rudics
3. c-Send n bytes of data via sbd
4. d-Send n bytes of data via sbd
5. e-Command text
6. f-Command text
7. g-Command text
8. h-Command text
9. i-Command text
10. j-Command text
Selection:4
----Send Command d----
Messege ID:123
Data length:4
Data:abab
----Waiting for response----
Command ID:d
MsgID:123
```

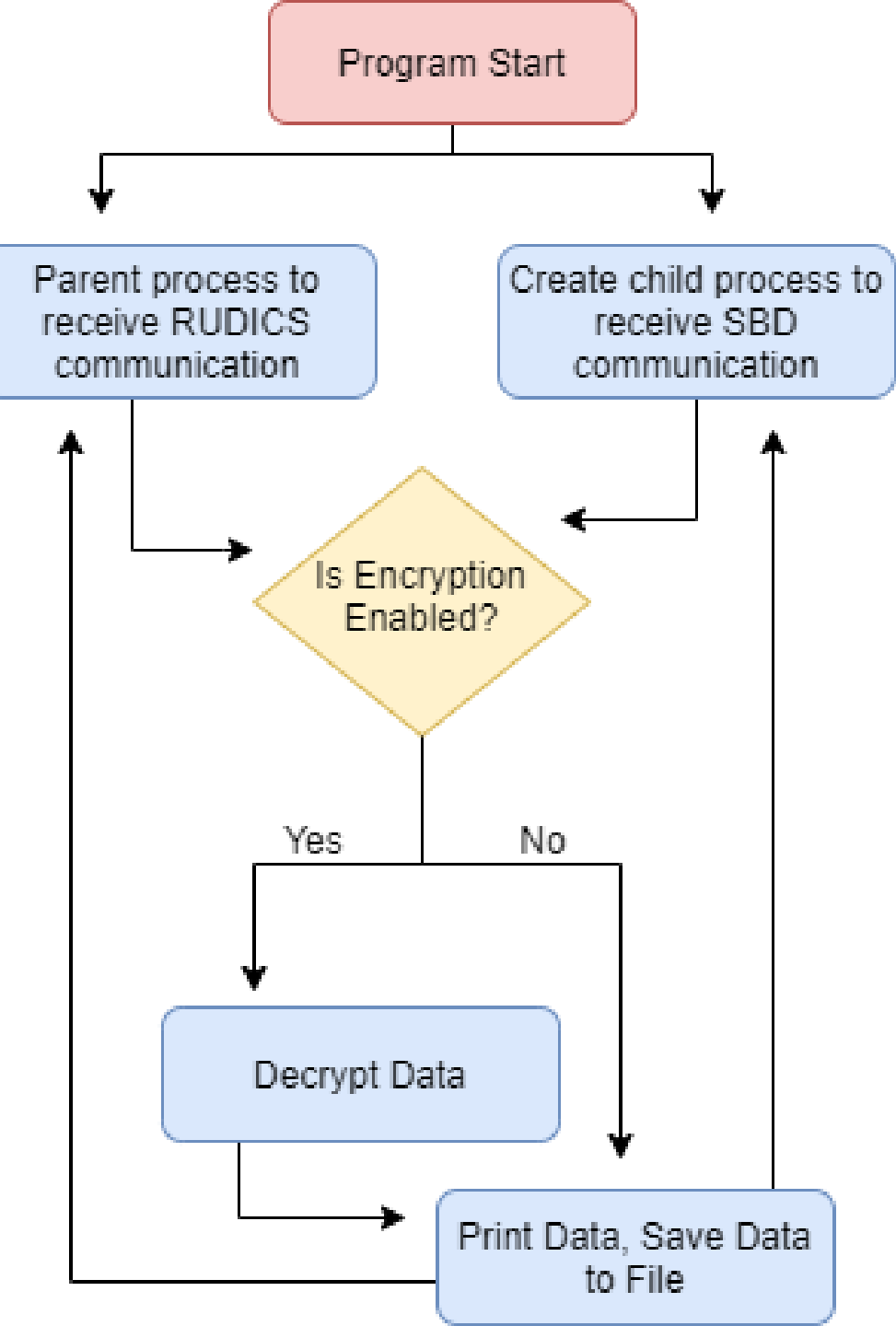
# Ground Server Application

- Post-process information
- Receive Both SBD or RUDICS communication
- Longevity

# Ground Server

- Ubuntu Digital Ocean Server
  - 1 GB Memory, 25 GB Disk
  - Static Public IP address
  - Security
    - Firewall
    - Antivirus Software

```
root@iridiumDropletGS:~/groundServer# cd
root@iridiumDropletGS:~# hostnamectl
  Static hostname: iridiumDropletGS
        Icon name: computer-vm
        Chassis: vm
        Machine ID: 5dc90434ef0f4f84b55f6828c745ff87
        Boot ID: 0c33ab7cb3414918878dca573d36e3e6
  Virtualization: kvm
  Operating System: Ubuntu 20.04.2 LTS
        Kernel: Linux 5.4.0-65-generic
        Architecture: x86-64
root@iridiumDropletGS:~#
```



# Python Code Breakdown

---

- Create child process for SBD communication
  - Binds designated port to host IP address
  - LOOP
    - Listen to receive communication
    - Print received communication to the screen and append to log file
- Parent process used for RUDICS communication

# Testing limits on the ground server

```
root@iridiumDropletGS:~/groundServer# head -n5 SBD_Communication_Data.txt
03/28/2021, 22:32:19,Hello, world
03/28/2021, 22:35:01,Hello, world
03/28/2021, 22:43:26,Hello, world
03/28/2021, 22:44:32,Hello, world
03/28/2021, 23:06:00,The Project Gutenberg eBook of Frankenstein, by Mary Wollst
onecraft (Godwin) Shelley This eBook is for the use of anyone anywhere in the Un
ited States and most other parts of the world at no cost and with almost no rest
rictions whatsoever. You may copy it, give it away or re-use it under the terms
of the Project Gutenberg License included with this eBook or online at www.guten
berg.org. If you are not located in the United States, you will have to check th
e laws of the country where you are located before using this eBook. Title: Fran
kenstein or, The Modern Prometheus Author: Mary Wollstonecraft (Godwin) Shelley
Release Date: 31, 1993 [eBook #84] [Most recently updated: November 13, 2020] La
nguage: English Character set encoding: UTF-8 Produced by: Judith Boss, Christy
Phillips, Lynn Hanninen, and David Meltzer. HTML version by Al Haines. Further c
orrections by Menno de Leeuw. *** START OF THE PROJECT GUTENBERG EBOOK FRANKENST
EIN *** Frankenstein; or, the Modern Prometheus by Mary Wollstonecraft (Godwin)
Shell
root@iridiumDropletGS:~/groundServer# ls -l
total 136
-rw-r--r-- 1 root root    204 Mar 28 22:44 RUDICS_Communication_Data.txt
-rw-r--r-- 1 root root 130676 Mar 28 23:06 SBD_Communication_Data.txt
-rw-r--r-- 1 root root   2143 Mar 28 23:07 server.py
root@iridiumDropletGS:~/groundServer# _
```



# Post-Processing: Encryption/Decryption

- AES 256 encryption standard
- Coded in python using wolfcrypt

## Satellite Communication

Unencrypted Data Sent --->

Encrypted Data Sent --->

```
(base) taeganwilliams@Taegans-MacBook-Pro groundServer % python3 clientEnc.py
This is encrypted when sent
b"{'cipher_text': b'qj2vGQ40p5pVG8PzWgnpFraIJzxCJ526+ibmoztvdoo=', 'salt': b'uqyK/8vx+STk77pheH44JQ==',
  'iv': b'7r+1iIUN3nyxAwP4Qm3rdg=='}"
(base) taeganwilliams@Taegans-MacBook-Pro groundServer % python3 clientEnc.py
This is encrypted RUDICS communication
b"{'cipher_text': b'KNTYuMq8BX8F80JF0prvXs0CZhZpmBiuNGMIiyQjeZV3LaYIaos0xmiSNvERY3HR', 'salt': b'2zPNKR
mmqL7qniuv9Jts6Q==', 'iv': b'71xraC41EL/0w/q5HZK79w=='}"
```

## Ground Server Communication

SBD comm Received --->

RUDICS comm Received --->

```
root@iridiumDropletGS:~/groundServer# python3 serverENC.py
Connected SBD by ('129.101.212.120', 1325)
Connected SBD by ('129.101.212.120', 1185)
Connected RUDICS by ('129.101.212.120', 1349)
^Z
[1]+  Stopped                  python3 serverENC.py
root@iridiumDropletGS:~/groundServer# cat SBD_Communication_Data.txt
03/29/2021, 01:55:48,This is encrypted when sent
root@iridiumDropletGS:~/groundServer# cat RUDICS_Communication_Data.txt
03/29/2021, 01:56:38,This is encrypted RUDICS communication
root@iridiumDropletGS:~/groundServer#
```

# What's next for the Ground Server



Analyze the received packet to determine communication type



Clean up the collection of data



Rewrite the program to work on a single port

# Validation from NASA

<b>NASA requirement:</b>	<b>Achieved by:</b>
Communication with primary satellite computer	Verifying TESBUS task is functioning
Verification of expected signal quality based on position	Sending current time in the message
Integrity of data transferred	Error detection byte within each message
Volume of data transferred	Sequential message IDs
Changing between SBD and RUDICS mode	Verifying Iridium task is functioning

Questions?