Support for RESTOR, EMIST, and CHREC Space Processor

Bradley F. Shea
NASA KSC
Major: Electrical Engineering
KSC Intern, Summer Session
Date: 25 06 2014

Support for RESTOR, CHREC Space Processor, and EMIST

Bradley F. Shea[1]
University of Florida, Gainesville, FL, 32607

Abstract

The goal of this project was to provide support for three different projects including RESTOR, CHREC Space Processor, and EMIST. LabVIEW software was written to verify tags in an excel spreadsheet, testing preparation was accomplished for CHREC, and full payload integration was completed for EMIST. All of these projects will contribute to advanced exploration in space and provide valuable experience.

Nomenclature

RESTOR	=	Reusable Earth Synchronous Tele-Operated Refueler
EMIST	=	Exposing Microbes In the Stratosphere
CHREC	=	Center for High-Performance Reconfigurable Computing
CSP	=	CHREC Space Processor
UF	=	University of Florida
PTA	=	Propellant Transfer Assembly
HMA	=	Hose Management Assembly
NTO	=	Nitrogen Tetroxide
ISEM	=	International Space Station Experiment Mini
CeREs	=	Compact Radiation belt Explorer
VAB	=	Vehicle Assembly Building

I. Introduction

YRESTOR is a project to advance the state of the technologies for in-space hypergolic fluid storage and transfer to non-cooperative satellites in geostationary orbit. KSC is supporting NASA Goddard for testing of the PTA and HMA elements. The PTA has been tested using Freon and NTO but, unexpectedly, one of the components was over-pressurized. The root cause was due to an error in the reactive control logic file. The significance of my effort will benefit RESTOR because it will prevent damage to the PTA in the future.

EMIST includes a payload that will carry biological samples that will measure microbial responses to the stratosphere. A monolayer of a known, non-hazardous microorganism will be deposited onto each experimental coupon. The coupons will be contained in a payload enclosure mounted on the balloon gondola and exposed to the upper atmosphere (30 to 35 km) for up to 1-8 hours before the sealed payload returns to the surface on a parachute. The significance of my effort will benefit EMIST because I will be able to help in getting the payload launched on time.

The CSP is currently being designed at the UF and testing/verification is going to be performed using NASA Kennedy's world-class laboratories. The CSP is going to be featured on two upcoming missions including ISEM and CeREs to study charged particle dynamics in geospace. It will demonstrate the use of combining COTS and RadHard (radiation hardened) components rather than just using one or the other. This method will enhance the reliability of space computing. The CSP will provide a combination of performance, reliability, size, weight, and cost for high-speed processing. I will be helping to prepare the CHREC team for testing of the ISEM cubesat.

[1] Flight Instrumentation Intern, Avionics, KSC, University of Florida.

II. Mission Support

A. RESTOR

The first week of arrival was spent familiarizing myself with the software and learning the functionality of each virtual instrument. The code was to extract tags (instructions for control logic) from an excel spreadsheet and verify them against a predefined set of correct tags. In order for LabVIEW to read from a spreadsheet, it had to be converted to a text file. Two files were to be verified including an autosequencer file and an 'R' file. This is a quick overview of how the code works: first, it reads from a text file and searches through it looking for keywords (Type, Condition, Action) and puts their location in an array according to row number. It then finds the difference between each of the positions to determine how many tags need to be verified in each section. The code steps through each tag and verifies that it matches one of the tags in a predefined set. If it does not find a match, an error is reported. The code is able to detect any error, including empty cells, and it can verify extra sections (i.e R04) that might be added.

Figures 1 and 2 below show what was verified and also the results of the verification. In Figure 2, you are able to browse for new file paths and you click on the 'START' button to enable the software.

Type:

A01	Tag	Argument		Value	Unit	Action	"Transfer Mode 1 - Recirc Startup w/
1	POS01	<	30	Position		Verify	Bellows at (POS01) %
2		>	135	PSIG	Verify	PT-10 verified > 135 PSIG	
3	SV06	=	Closed	0	Verify	SV-06 Verified Closed	
4	SV02	=	Open	0	Verify	SV-02 Verified Open	
5	SV03	0	0	0	Open	SV-03 Commanded Open	
6	SV03	=	Open	0	Verify	SV-03 Verified Open	
7	Rate	=	20	Hz	0	Sample Rate 20Hz	
8	P0(X)	0	M(X) Speed		RPM	On	Pump (X) commanded to (X)RPM
9	Delay	0	5	seconds	Delay	Delaying 5 seconds to Recirculate	
10	SV06	0	0	0	Open	SV-06 Commanded Open	
11	SV06	=	Open	0	Verify	SV-06 Verified Open	
12	SV03	0	0	0	Close	SV-03 Commanded Closed	
13	2	=	Closed	0	Verify	SV-03 Verified Closed	
14	P0(Xn)	0	M(X) Speed		RPM	On	Pump Startup for Pumps (Xn) complete
15	P0(Xn)	=	TF	Flow	PID	Maintaining Constant Flow	
16	Delay	=	10	seconds	Delay	Waiting to change sample rate	
17	Rate	=	1	Hz	0	Sample Rate 1Hz	
18	SST	>=	TM*.99	Lbs	Verify	99% of target mass	
19	Rate	=	20	Hz	0	Sample Rate 20Hz	
20	SST	>=	TM	Lbs	Verify	Target mass achieved	
21	Display	0	0	0	Display	Shutdown initiated	
22	5	0	0	0	Open	SV-03 Commanded Open	
23	SV03	=	Open	0	Verify	SV-03 Verified Open	
24	SV06	0	0	0	Close	SV-06 Commanded Closed	
25	SV06	=	Closed	0	Verify	SV-06 Verified Closed	
26	SV02	0	0	0	Close	SV-02 Commanded Closed	
27	SV02	=	Closed	0	Verify	SV-02 Verified Closed	
28	7	0	0	RPM	Off	All Pumps Commanded Off	
29	SV03	0	0	0	Close	SV-03 Commanded Closed	
30	SV03	=	Closed	0	Verify	SV-03 Verified Closed	
31	Delay	=	5	seconds	Delay	Waiting to change sample rate	
32	Rate	=	1	Hz	0	Sample Rate 1Hz	
33	Display	0	0	0	Display	Pump shutdown complete	

Figure 1. "A" file to read. *The data under the tag section gets verified. (i.e. SV06)*

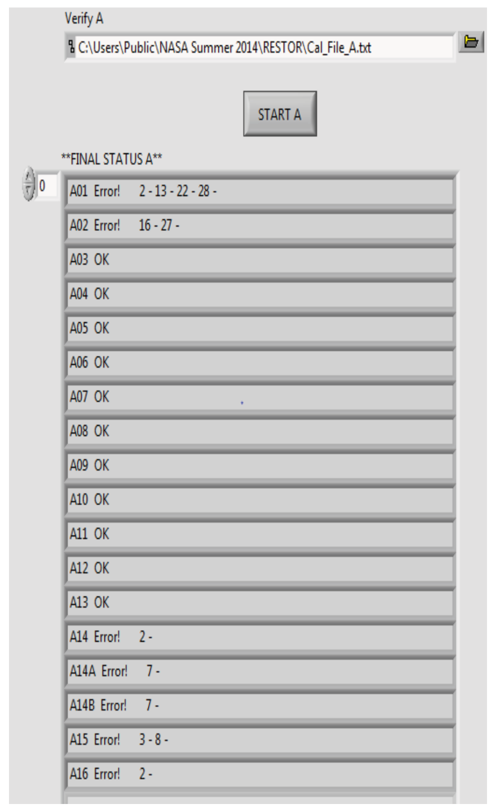
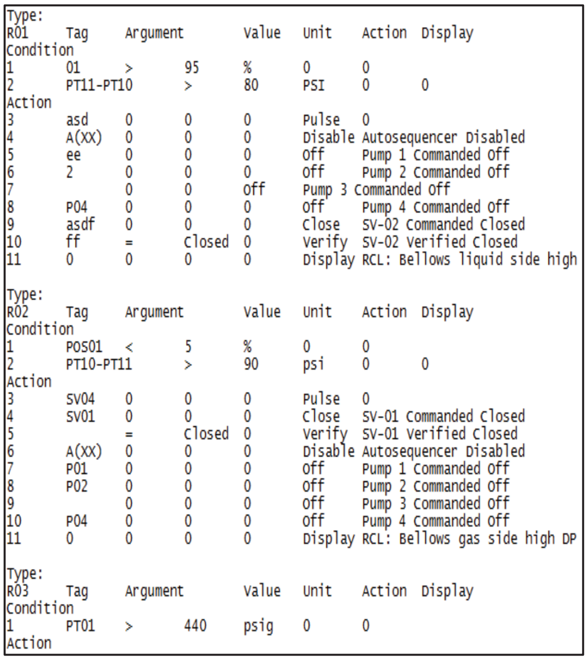


Figure 2. LabVIEW A final output. *Shows which sections are OK and which have errors.*



```
Type:
R01     Tag     Argument        Value   Unit    Action  Display
Condition
1       01      >       95      %       0       0
2       PT11-PT10       >       80      PSI     0       0
Action
3       asd     0       0       0       Pulse   0
4       A(XX)   0       0       0       Disable Autosequencer Disabled
5       ee      0       0       0       Off     Pump 1 Commanded Off
6       2       0       0       0       Off     Pump 2 Commanded Off
7               0       0       Off     Pump 3 Commanded Off
8       P04     0       0       0       Off     Pump 4 Commanded Off
9       asdf    0       0       0       Close   SV-02 Commanded Closed
10      ff      =       Closed  0       Verify  SV-02 Verified Closed
11      0       0       0       0       Display RCL: Bellows liquid side high

Type:
R02     Tag     Argument        Value   Unit    Action  Display
Condition
1       POS01   <       5       %       0       0
2       PT10-PT11       >       90      psi     0       0
Action
3       SV04    0       0       0       Pulse   0
4       SV01    0       0       0       Close   SV-01 Commanded Closed
5               =       Closed  0       Verify  SV-01 Verified Closed
6       A(XX)   0       0       0       Disable Autosequencer Disabled
7       P01     0       0       0       Off     Pump 1 Commanded Off
8       P02     0       0       0       Off     Pump 2 Commanded Off
9               0       0       0       Off     Pump 3 Commanded Off
10      P04     0       0       0       Off     Pump 4 Commanded Off
11      0       0       0       0       Display RCL: Bellows gas side high DP

Type:
R03     Tag     Argument        Value   Unit    Action  Display
Condition
1       PT01    >       440     psig    0       0
Action
```

Figure 3. 'R' text file to read. *LabVIEW converts an excel spreadsheet into a form that it can read which is text format. The data under the 'Tag' section gets verified.*

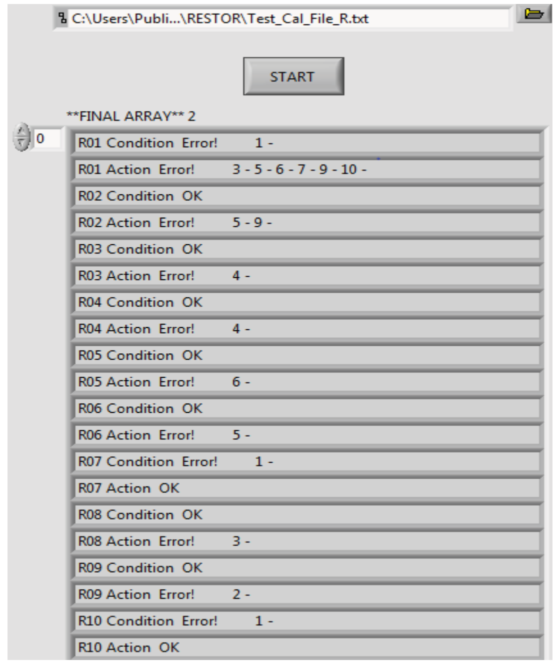


Figure 4. LabVIEW R verification final output. *This shows whether each tag section in the code contains errors or not. For example, section R01 Action has 6 errors. The numbers represent the location of each individual error.*

B. CHREC Space Processor

After my undergraduate research experience in the CHREC lab in Spring 2014, I was assigned to going back to Kennedy Space Center to test the ISEM cubesat. I was able to meet with the vibrational test lab lead to acquire a checklist of things that would be needed for testing. Some of the requirements included vibration profiles, duration, attitudes, and test limits. There were also video, data capture, and sample rate concerns that had to be taken care of.

Many more questions had to be answered for thermal vacuum chamber testing and EMI testing. They were very similar to the questions for vibrational testing. Although the ISEM cubesat never got shipped down to KSC during Summer 2014, I will be back to oversee the testing in Fall 2014. I was still able to help in the preparation aspect and take care of all the requirements that are needed to even begin testing.

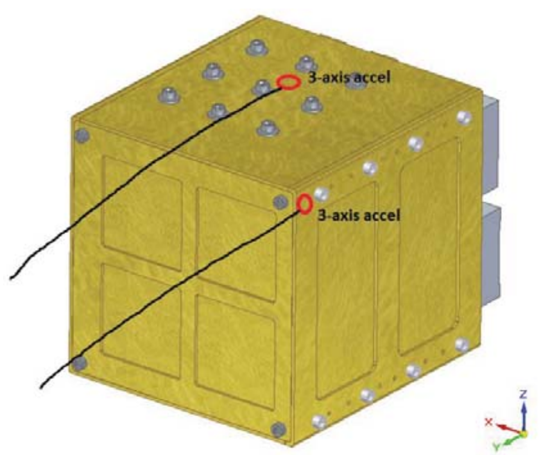


Figure 5. Accelerometer locations for ISEM. *This is part of the questions that had to be answered for the testing checklist for ISEM.*

C. EMIST

During my involvement with EMIST, I was able to gain so much practical and technical knowledge. I helped with integration from bread-board development to full payload integration. The first task that I completed involved helping the avionics lead prepare for a design review with chief engineers. We worked on a power budget which is an analysis of how much power you can draw without draining the battery prematurely during the mission.

Component Description	Voltage (V)		Current (A)		Power (W)		Duration (Hr)	Sleep (Hr)	Power Consumption for whole Mission (W-Hr)
	Min	Max	Min	Max	Min	Max			
Photocell Light Sensor	5	5	0.1	0.1	0.5	0.5	8	0	4
Internal Humidity/Temperature Sensor	3.3	6	0.001	0.0015	0.0033	0.009	8	0	0.072
12V Gear Motor w/ Encoder	12	12	0.07	1.773	0.84	21.276	0.022222222	7.978	7.17432
Arduino MicroController	5	5	0.058	0.1	0.29	0.5	8	0	4
4 Channel DC Motor Controller	12	12	0.1	3	1.2	36	0.022222222	7.978	10.3736
5V Heating Pad 5x10cm	3.3	5	0.7	0.7	2.31	3.5	4	4	14
Altimeter	3	3.6	9E-07	1.4E-07	2.7E-06	5.04E-07	8	0	0.000004032
microSD Module	5	5	0.2	0.2	1	1	8	0	8
GPS Receiver	3.3	3.3	0.075	0.075	0.2475	0.2475	8	0	1.98
									49.59992403
Battery Life (W-Hr) = 372.96									

Figure 6. Power budget. *This analysis shows how much power each component of the payload is going to require from the battery. The payload needs a total of about 50 Watt-hours for an 8 hour mission.*

We also worked on a flow diagram which specifies the main outline of the software and used it as a reference for logic testing. The general flow of the code is power on, initialize all sensors, check if GPS is available, poll all sensors, and check to see if the trigger altitude is reached (70000 feet) which will open the skewers. After the design review, we ran into many challenges with designing a circuit for 3 heating pads. We initially had very low watt heaters that you couldn't connect directly to the battery. This was a problem because we had to try to come up with a circuit without burning up other inline components. The heaters draw a significantly higher current than what most resistors and voltage regulators can handle. We

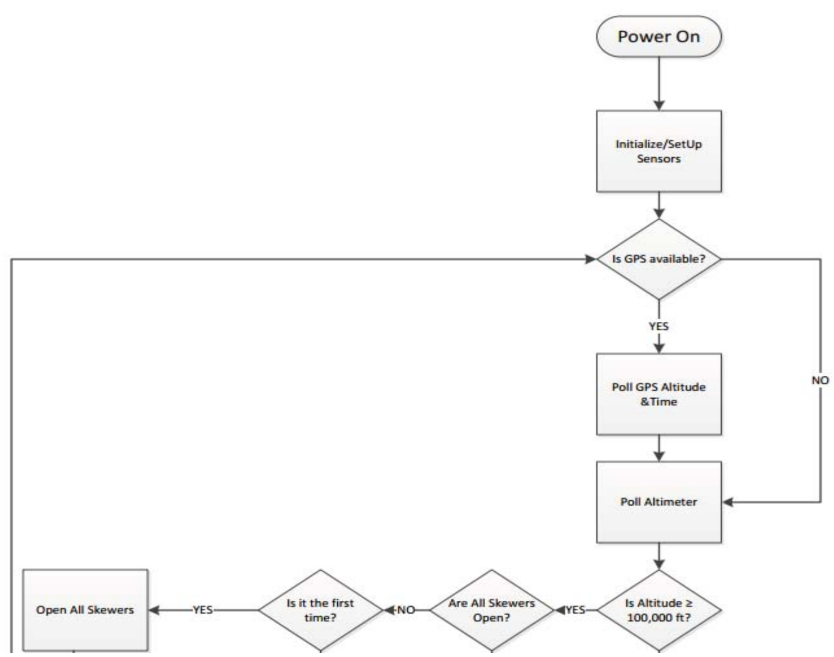


Figure 7. Software flow chart. *This diagram shows a portion of the software flow chart. It gives a general idea of how the software was written.*

tried to design a voltage regulator circuit that would drop the voltage from 16 V down to the 5 V required for the heaters. This design created a huge amount of heat through the voltage regulator and it had internal protection on it that limited the current flow to less than needed. Heat sinks could have been mounted to the regulators but they are very bulky, add a significant amount of weight, and are not very efficient. We ended up buying 28 V heaters that could be connected directly to the battery and this eliminated the need for extra circuitry.

We then started working on transferring all the components (GPS, SD card, altimeter, temperature/humidity sensor) to a proto-shield that can be mounted on top of an Arduino microcontroller. Each component was mounted and tested individually to make sure all solder connections were okay. After all components were verified, each individual piece of code for each sensor was integrated into a final version. Wiring harnesses were then fabricated for the status LEDs, motors, SD card, and GPS.

I was also able to write some code to turn the motors counter-clockwise and clockwise based on values from the motor encoder. The skewers on the payload needed to turn clockwise 180 degrees and then counter-clockwise 180

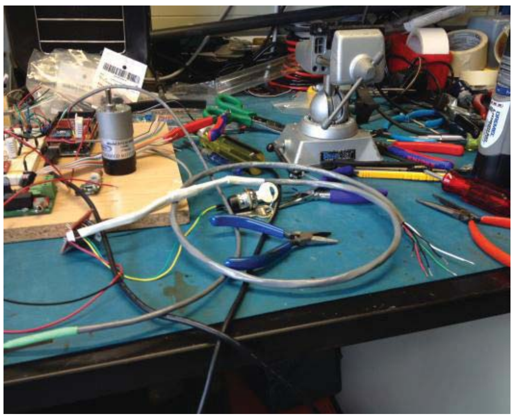


Figure 8. EMIST wiring harness. *This harness was used to connect a motor to the microcontroller.*

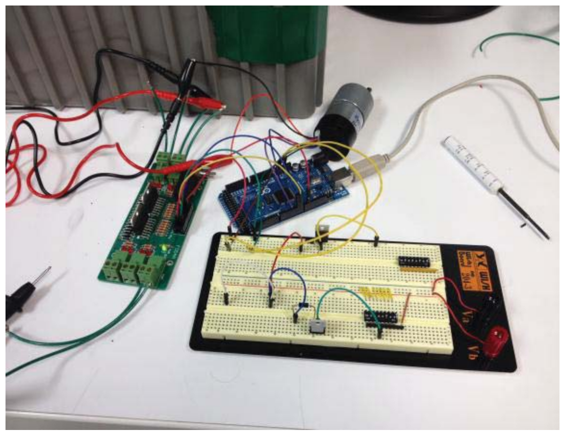


Figure 9. Testing motor control. *Shows testing with motor controller, motor and Arduino.*

degrees. Each motor needs to have its own set of calibration values because they don't turn the exact same distance every time.

Figure 10. Plate drilling. *Here we are drilling holes in the plate so we can mount the avionics.*

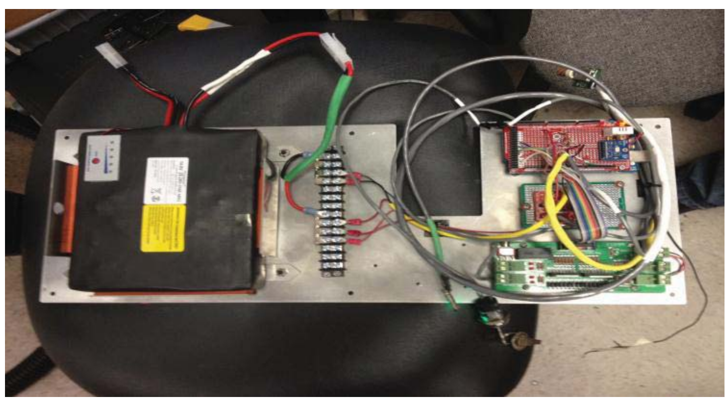


Figure 11. Avionics plate. *Most of the hardware has been added to the plate.*

All of the avionics were then mounted to a plate so we could perform vibrational testing. We mounted a terminal block, motor controller, a board that contains a 5 V power supply and circuits for 3 temperature sensors, and microcontroller. After this was complete, we began vibrational testing to ensure that our avionics were still going to perform (write data to SD card) under a stressed environment. The test would be conducted in a X, Y, and Z axes configuration. During setup, we had a mishap on our plate. The GPS got grounded to the plate and shorted out our 5 V power supply. This was a quick fix because we had an extra one on hand and we wrapped tape around the GPS to prevent future shorts. We didn't have any problems throughout the rest of the test and data was successfully written to the SD card.

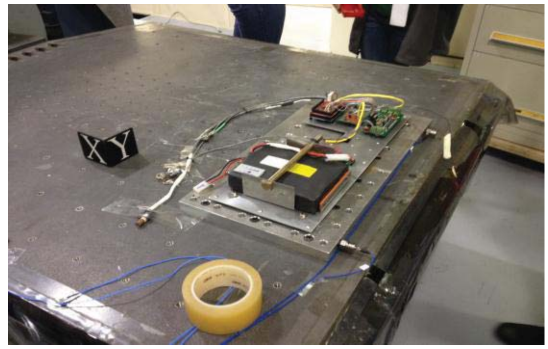


Figure 12. Vibrational testing. *This is a test conducted along the Y-axis.*

After testing we performed some validation testing of our altimeter by going up to the VAB roof. The altimeter wasn't giving us very good data at sea level and it was only reading 365 feet on top of the VAB (525 feet). We added some calibration code to the software to zero it out at sea level.

Figure 13. Altimeter test. *We were verifying the data from the altimeter on top of the VAB.*

The logic testing of the software involved writing in "dummy" data to the software to make it think that it has reached trigger points to open/close the skewers. For example, the motors are set to turn when the altimeter reads 70000 feet. If you are at ground level (2 feet), you can add 72000 feet to that and we verified that our software was correct. The motors did spin. The software also has low battery voltage detection which closes the skewers to save the experiment. This was also verified in logic testing.

The final steps for EMIST was the integration of the avionics into the enclosure and thermal testing. We were then able to perform final testing with a load on the motors and made sure all the cables were properly secured. I also helped with putting the pieces of the final build together. For thermal testing, we ramped the temperature from 80 °F down to -25 °F for what is called a cold soak. We wanted to verify that the skewers turned when the chamber reached the correct pressure (that correlates to 70000 feet) and that all of the components still worked after being exposed to this intense temperature. It turned out that we had a bug in our software because one of the header files for the altimeter didn't contain enough information for data above 32000 feet. We made some edits to the code and will perform another test to verify that it works as expected.

Figure 14. Thermal testing. *This is EMIST inside of the thermal vacuum chamber.*

III. Conclusion

Many engineering challenges came up during my time at KSC but we were able to figure them out and work through them as a team. I will leave KSC with an immense amount of knowledge and will always remember what I accomplished during my internship. The skills I will take away from this experience include LabVIEW software, Arduino platform, fabrication of cables, and soldering.

Acknowledgments

I would like to thank my mentor, Cara Evers, for providing me with the opportunity to work at KSC on an awesome project. She provided me with the support and guidance I needed to complete my project. I would also like to thank Anthony Bharrat for allowing me to help him on the EMIST payload and for teaching me some skills in C programming. It will be a great feeling to have a payload flown to near-space that I was involved in. I thank United Space Research Association for financial support as well.

References

Evers, Cara. Avionics Engineer.
Bharrat, Anthony. Computer Engineer.
Otermat, Nicole. Avionics Engineer.