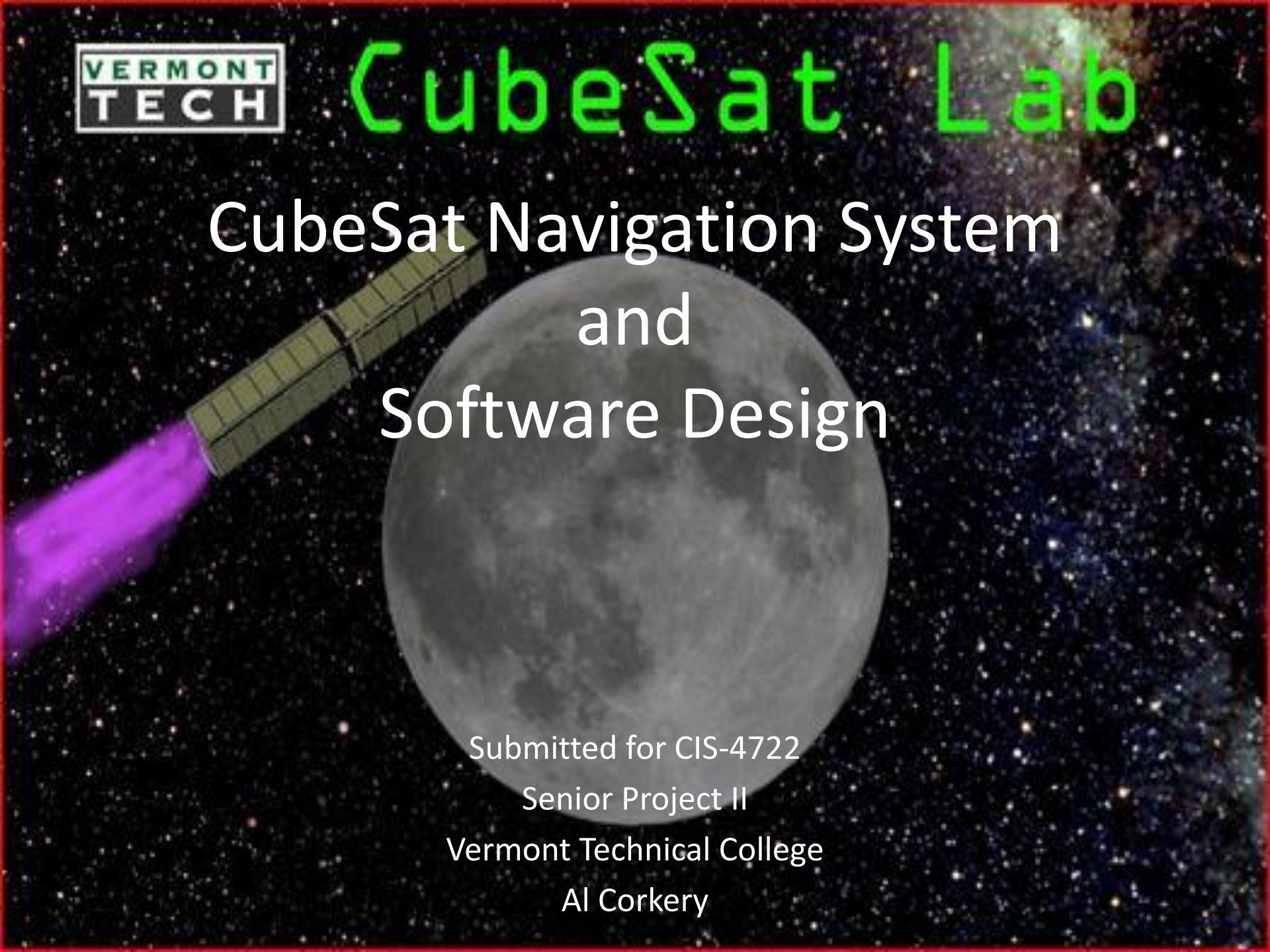


CubeSat Navigation System and Software Design

Submitted for CIS-4722

Senior Project II

Vermont Technical College

Al Corkery

Project Objectives

- Research the technical aspects of integrating the CubeSat's Global Positioning System receiver with the NASA provided navigation software
- Propose a Software/Hardware Architecture solution
- Implement and test that solution
- Report my results to the customer

The CubeSat Lunar Lander Project

- Goal: Develop and deploy a satellite to collect data from the moon and transmit the data back to Vermont.
- Vermont Technical College: structure, thrusters, communication, navigation, electronics
- University of Vermont: low energy transfer, radiation environment modeling, spacecraft coordination and landing

The CubeSat Lunar Lander Project

- Norwich University: optical sensors and robotics
- NASA Goddard Spaceflight Center and NASA Jet Propulsion Laboratory personnel

The CubeSat Lunar Lander Project

- First Mission: Orbital launch
- Test the navigational software.
- VTC proposal approved.
- March 2012 launch secured.

The CubeSat Lunar Lander Project

- ELaNa IV mission
- Falcon 9 (SpaceX) rocket from Cape Canaveral
- 325km (202 mi) orbit

Vermont Tech Student Involvement

- Electro-Mechanical Engineering Technology Bachelors students can take the Spacecraft Systems course and work in the CubeSat Lab, on the Lunar spacecraft which becomes their required Senior Project

- They implement the specific systems for the CubeSat.

Vermont Tech Student Involvement

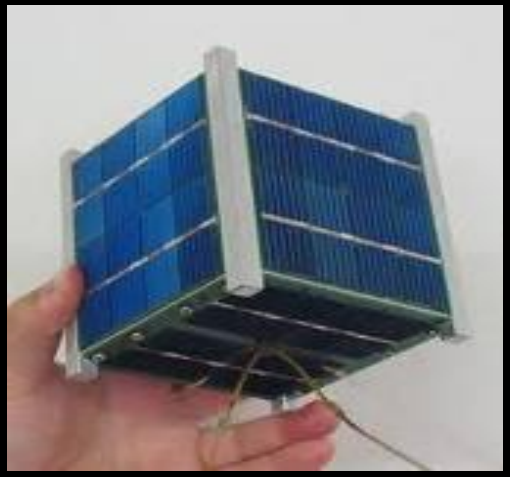
- Software Engineering and Computer Engineering Bachelors students can work on implementing parts of the software systems for their Senior Project and are currently developing a test framework for the navigation system.

- Other Electro-Mechanical students have worked on a steerable dish antenna for the GENSO ground station and a testing system for the Lunar Lander landing gear

What is a CubeSat?

- It’s a miniature satellite used for scientific research built from commercial off-the-shelf electronics.
- A CubeSat is a payload package that is attached to a rocket
- Dimensions 10cm x 10cm x 10cm.
- Has a mass of up to 1.33 kg.
- Relatively low cost.
- Relatively rapid development cycle.

What is a CubeSat?

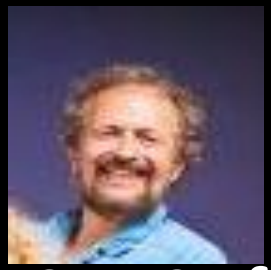


My Assignment

- As a CPE major, work on the low level integration of the GPS receiver within the Navigation Subsystem
- In particular, figure out how our NASA supplied navigation software will interact with the GPS receiver
- Develop/Install the necessary hardware and software components

My Customers

- Dr. Carl Brandon, Head of the CubeSat Lab at VTC and Principal Investigator for the CubeSat Lunar Lander Project.



- Prof. Peter Chapin, CubeSat Software Director at VTC.

My Customers

VTC Senior Projects resident hero:

Dr. Craig Damon

The Navigation System

- Provides onboard orbit determination
- Computes where you are and where you should be at a given time in the future
- Provides accurate time measurements
- Gives velocities
- Calculates orbital adjustments

The Navigation System Components

- Navigation software
- GPS receiver.
- On Board Computer (OBC).
- The interconnections and communication between the three components

The Navigation Software

- Goddard Enhanced Onboard Navigation System (GEONS)
- Provides onboard orbit determination
- Improves predictive accuracy of GPS point solution fixes
- Written in ANSI standard C language
- Integrated either within GPS receiver, a com receiver, or the OBC

GEONS

Developed by NASA in response to:

1. Increased need for satellite autonomy
2. Support for collaborative scientific missions
3. Support for limited GPS visibility (high-earth orbit)
4. Proposed formation flying missions

GEONS

- Provides OOP functionality through the use of structures.
- Acts as library of methods
- Methods called via a host application
- Requires GPS data retrieved via Application Programming Interface (API) calls
- Size: 189KBytes

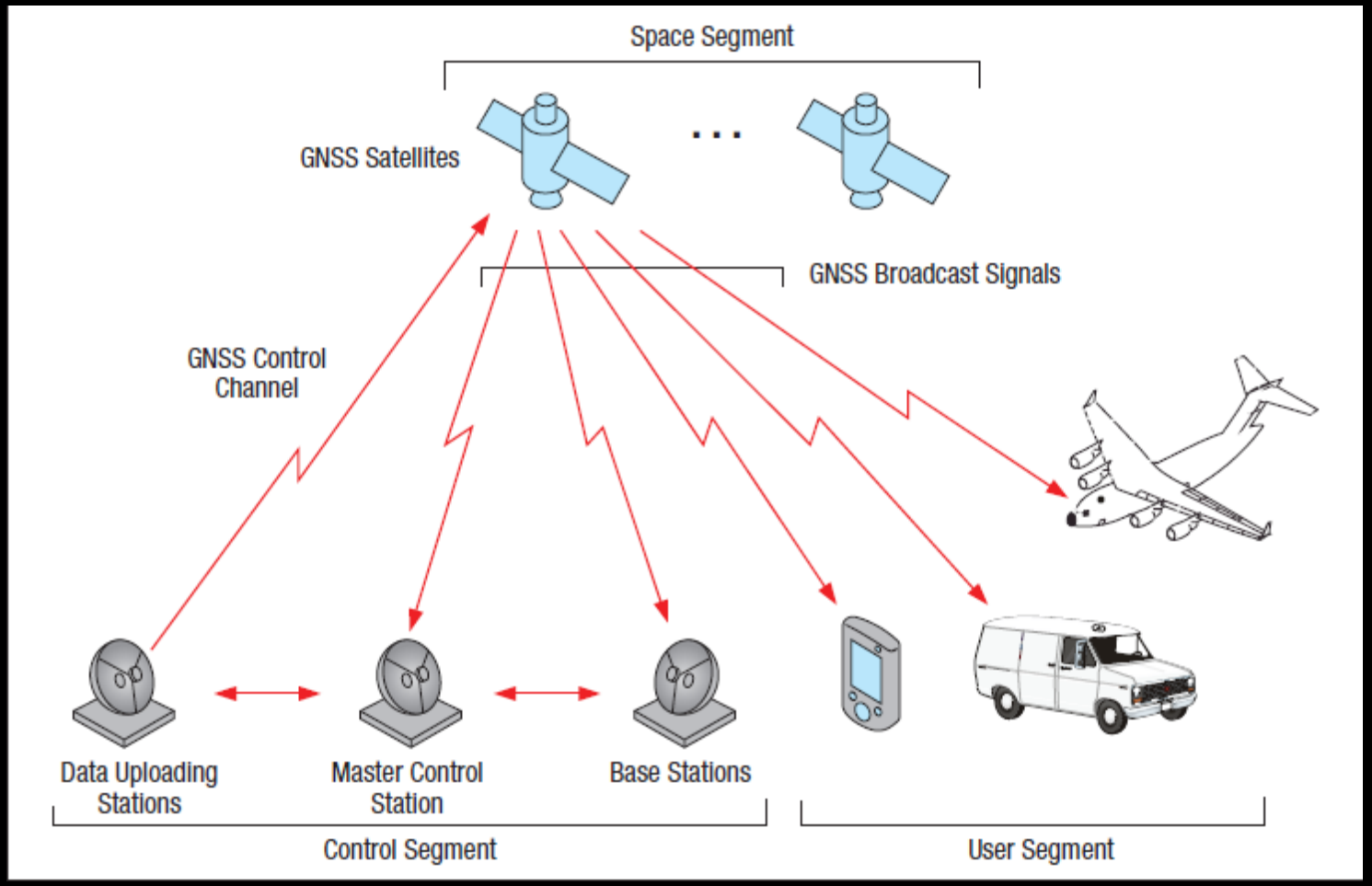
GEONS

- Originally proposed to be loaded and run from the spacecraft's On Board Computer, a Texas Instruments MSP-430
- Maximum available memory for any MSP-430 variant – 256KBytes
- My initial research into our GPS receiver revealed a possible solution
- The OEMV-1 with API option enabled provides 4MBytes
- So, prove it will work

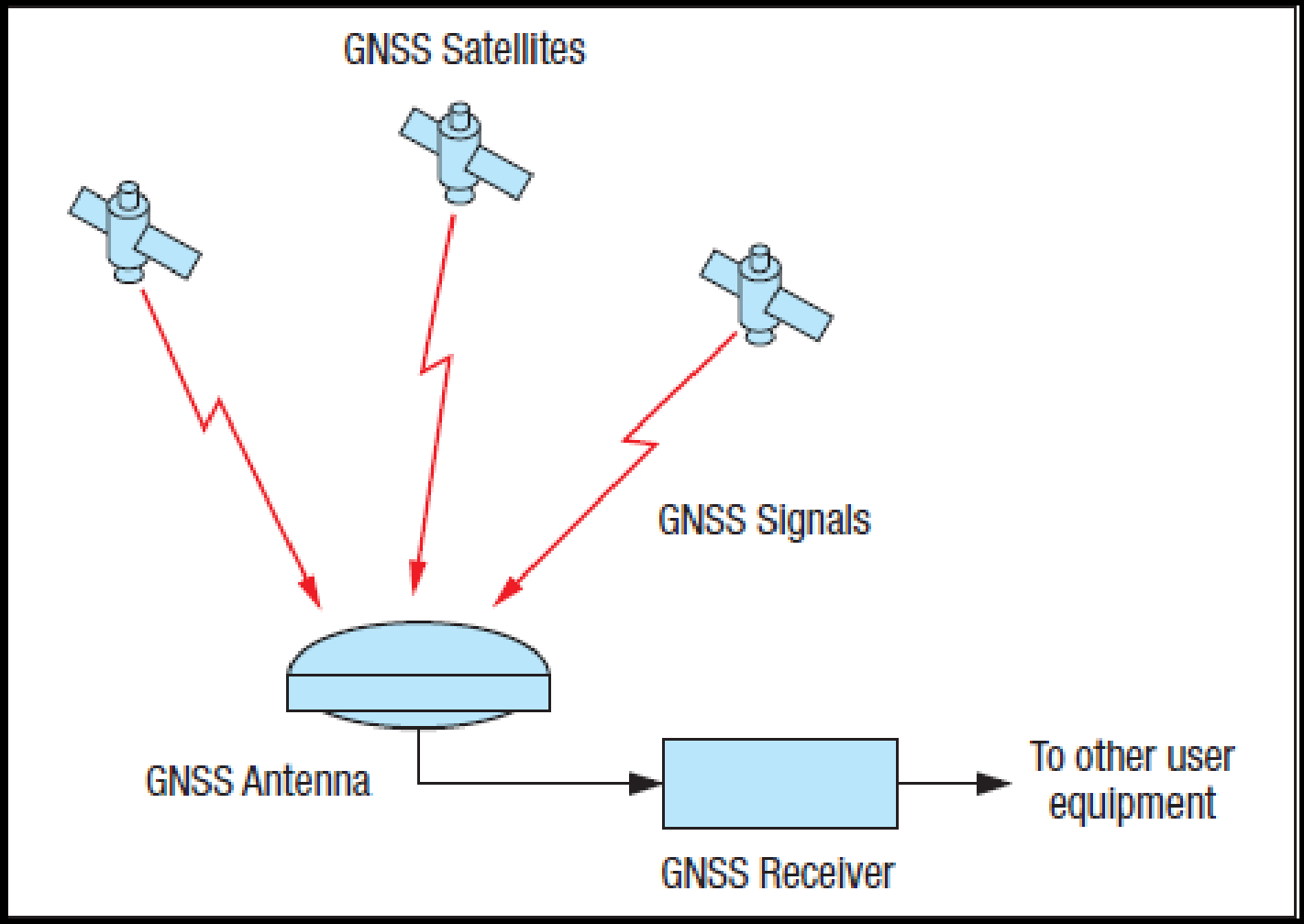
GPS

- Global Positioning System (GPS), GLONASS
- A space-based Global Navigation Satellite System (GNSS)
- Provides reliable location and time data
- Requires Line-of-Sight to 4 or more GPS satellites
- Commercial Receiver Limitations:
  18 kilometers (11 mi) altitude and 515 m/s or 1152 mi/hr velocity

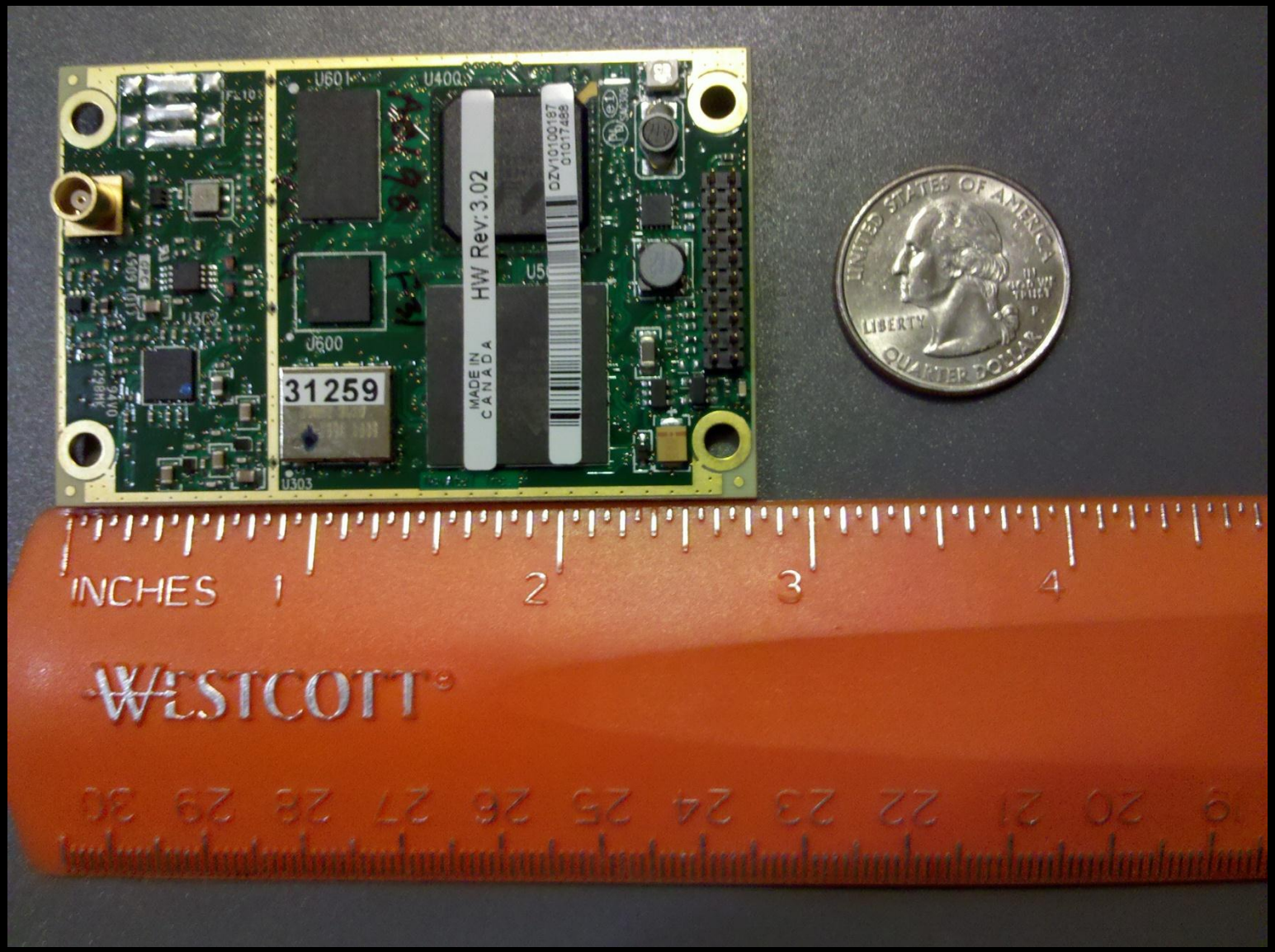
GPS


Space Segment
GNSS Satellites
GNSS Broadcast Signals
GNSS Control Channel
Data Uploading Stations
Master Control Station
Base Stations
Control Segment
User Segment

GPS


GNSS Satellites
GNSS Signals
GNSS Antenna
GNSS Receiver
To other user equipment

NovAtel OEMV-1



NovAtel OEMV-1

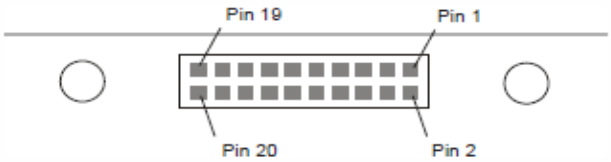
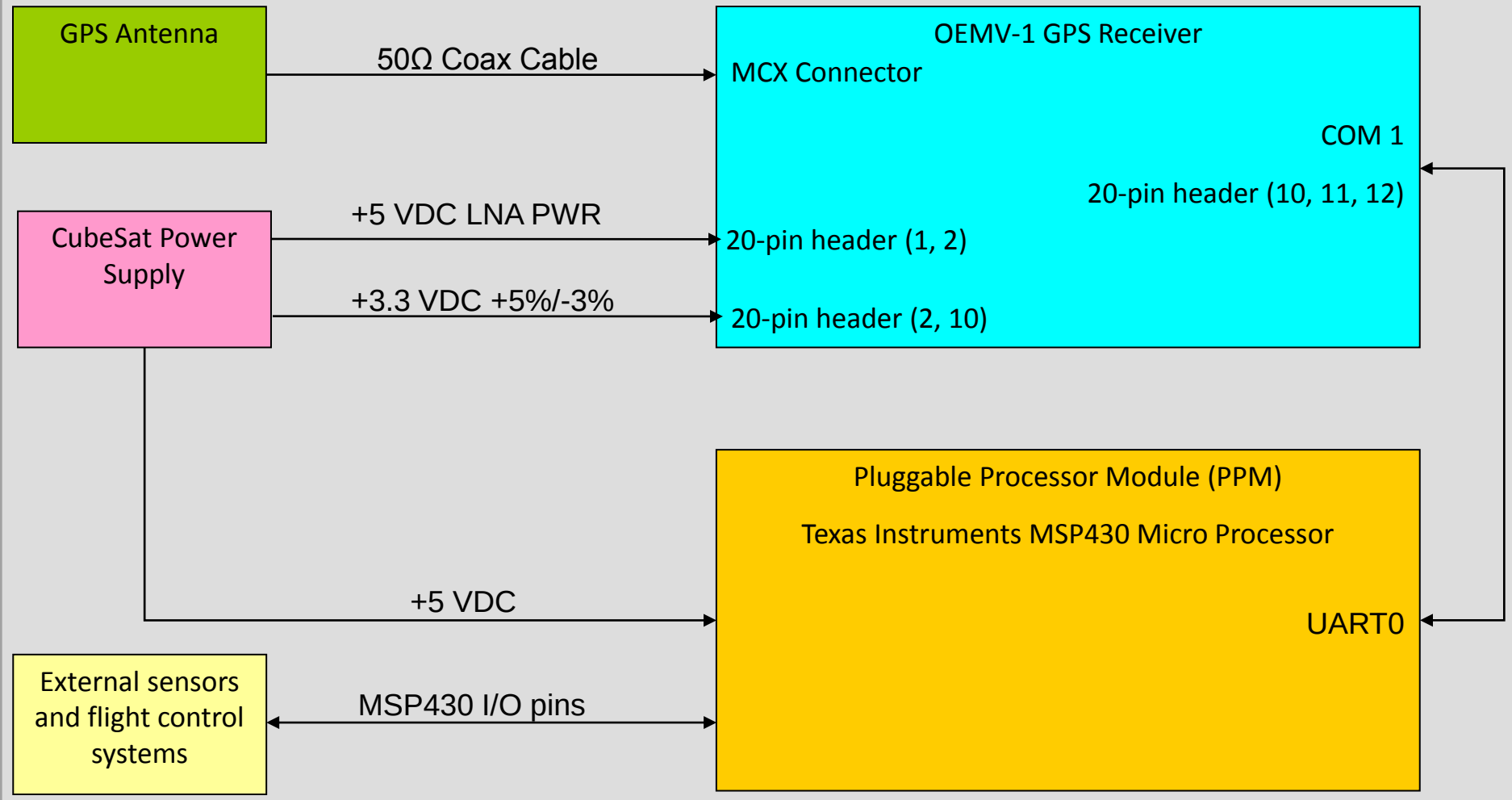


Figure 37: Top-view of 20-Pin Connector on the OEMV-1

Signal	Behavior [a]	Descriptions	Pin
LNA_PWR	Input DC	Power supply for external antenna LNA	1
V_{IN}	Input DC	DC power supply for card	2
USB D (-)	Bi-directional	USB interface data (-)	3
USB D (+) / COM3_Rx	Multiplexed	Multiplexed pin behavior default: USB D (+)	4
RESETIN	See strobes	Card reset	5
VARF / CAN1_Rx	Multiplexed	Multiplexed pin behavior, see strobes default: VARF	6
Event2 / CAN1_Tx	Multiplexed	Multiplexed pin behavior, see strobes default: Event2	7
CAN2_RX	Bi-directional	CAN Bus dedicated port	8
Event1 / COM3_Tx	Multiplexed	Multiplexed pin behavior, see strobes default: Event1	9
GND	Ground	Digital Ground	10
COM1_Tx	Output	Transmitted Data for COM 1 output	11
COM1_Rx	Input	Received Data for COM 1 input	12
GND	Ground	Digital Ground	13
COM2_Tx	Output	Transmitted Data for COM 2 output	14
COM2_Rx	Input	Received Data for COM 2 input	15
GND	Ground	Digital Ground	16
PV	See strobes	Output indicates 'good solution' or valid GPS position when high	17
GND	Ground	Digital Ground	18
PPS	See strobes	Pulse output synchronized to GPS Time	19
CAN2_TX	Bi-directional	CAN Bus dedicated port	20

System Decomposition



NovAtel OEMV-1

- L1 band GPS receiver (1575.42MHz)
- Multiple GPS message formats: RTCA, RTCMV3, RTCM,CMR, CMRPLUS and NMEA
- ARM XScale processor
- API option: develop specialized C/C++ applications to further extend the functionality of the OEMV-1 receiver
- 4MB of available memory

Hardware Tasks

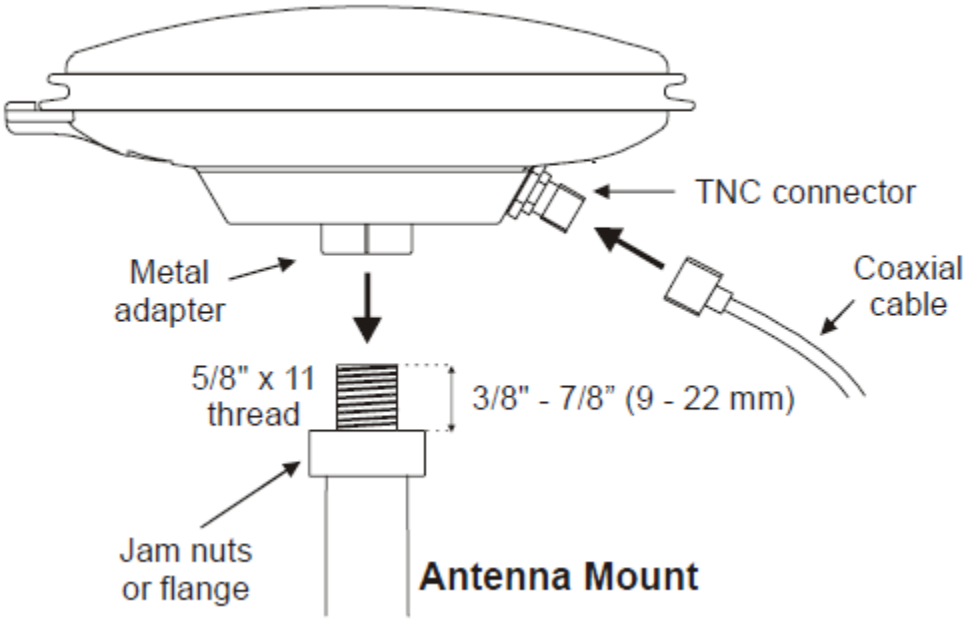
- Perform a firmware upgrade to enable API option and disable Altitude/Velocity restrictions
- Upgrade can only be done through COM1 or COM2
- These ports are LVTTL, PC serial ports are RS-232

Hardware Tasks

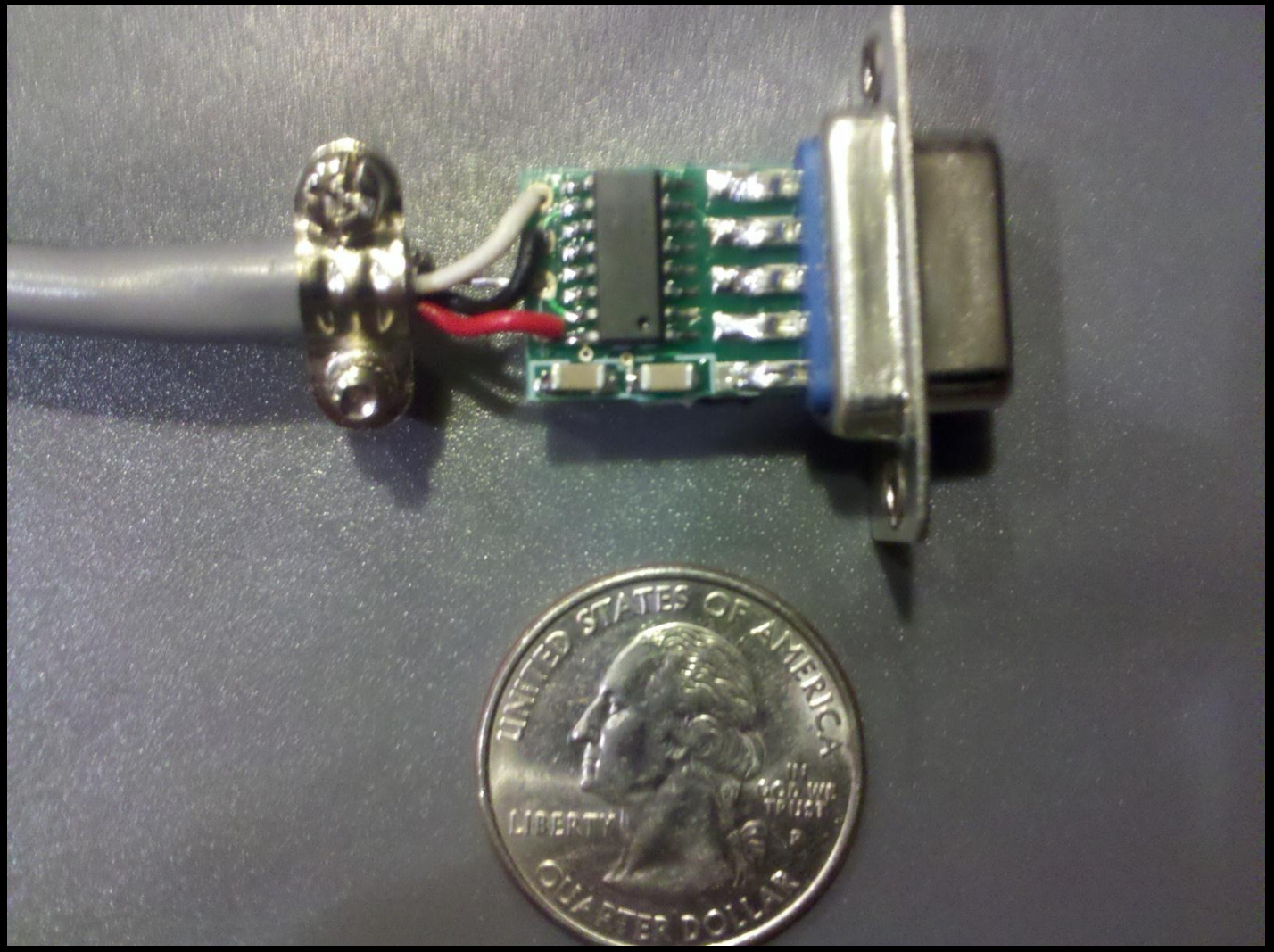
- Specify and mount outdoor GPS antenna and cable
- Connect to OEMV-1: COM1, COM2, USB1 from a PC
- Design and build a signal breakout board
- Include an RS-232 to LVTTL transceiver

Hardware Tasks

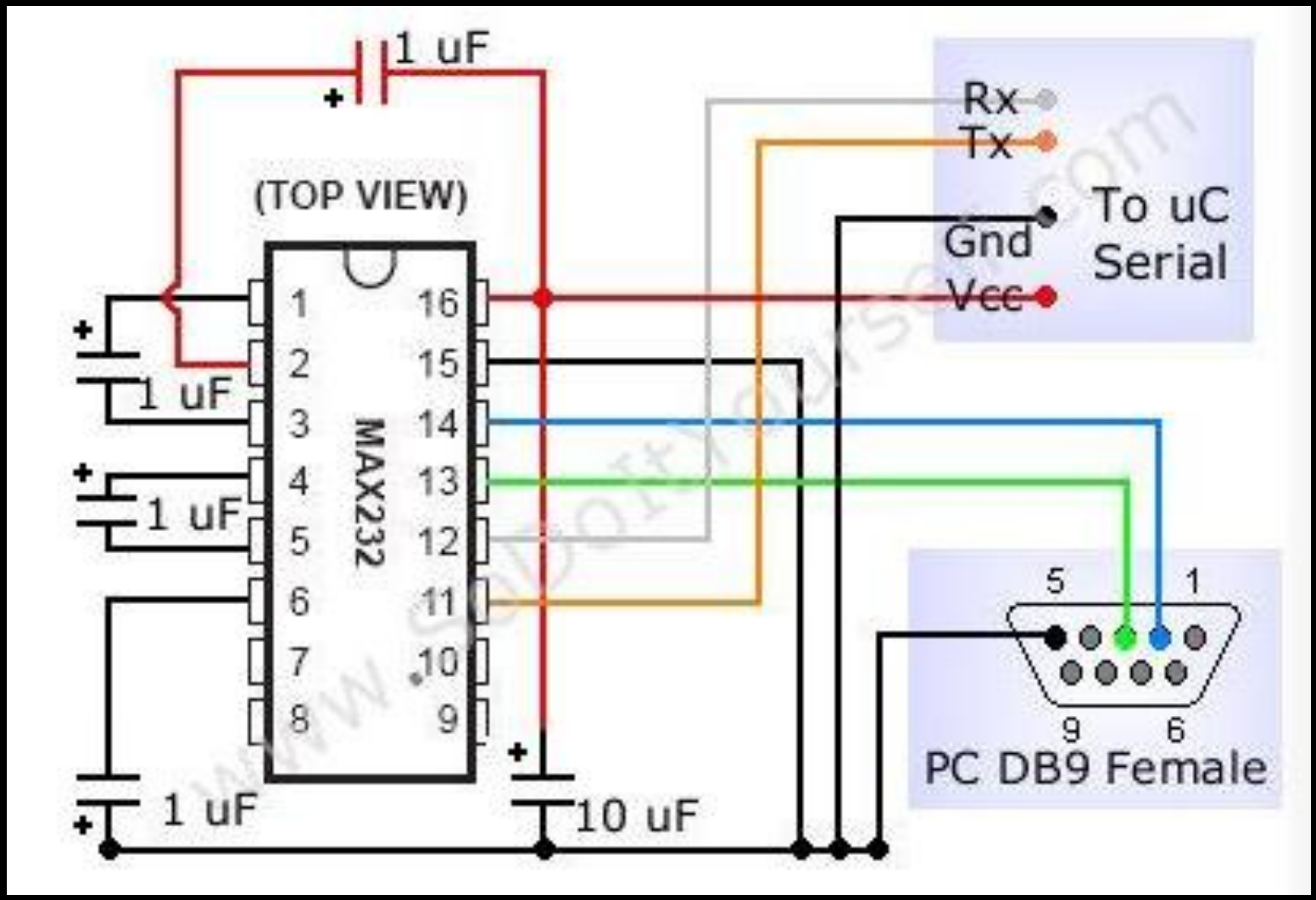
- **GPS-701-GGL Antenna**
- **15 meters RG-56 50 ohm cable**
- **MCX to TNC adaptor cable**



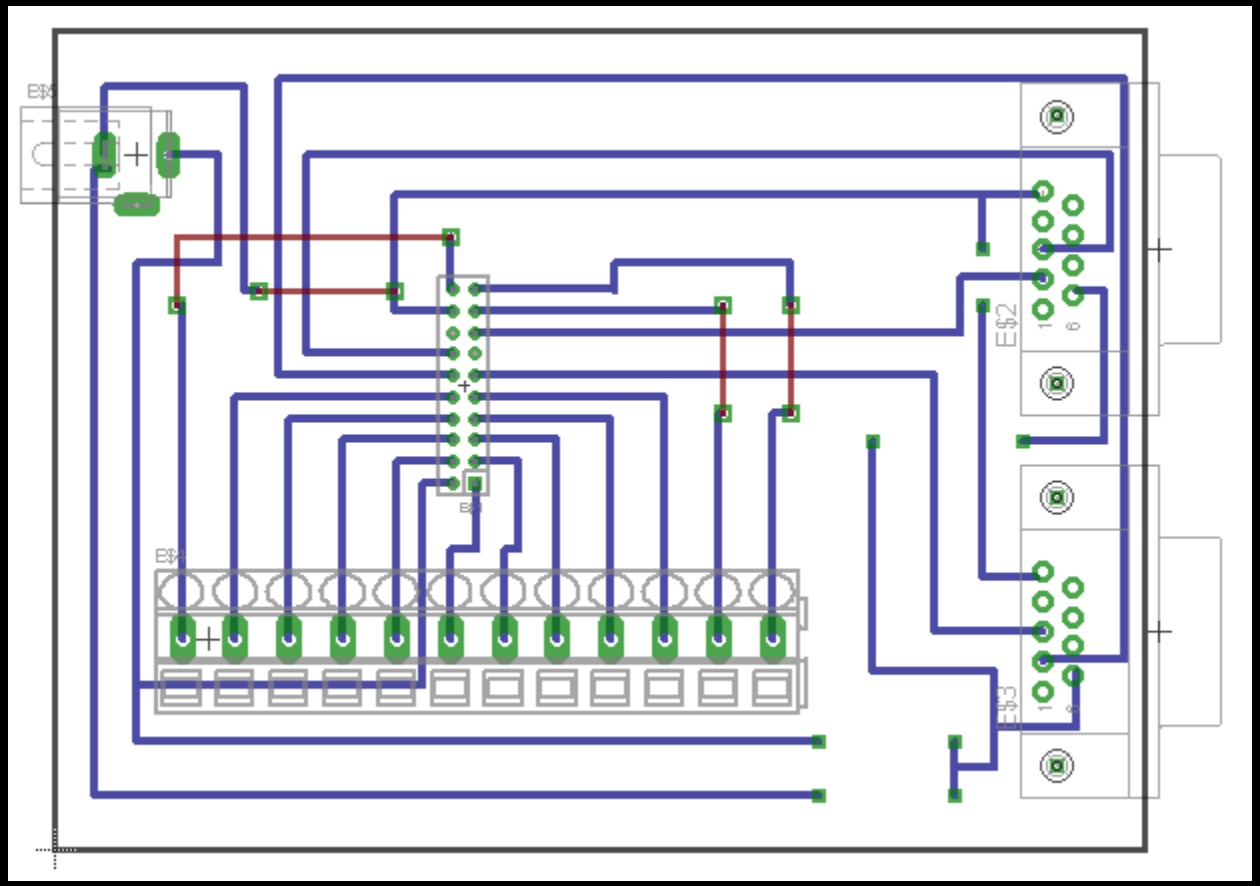
Hardware Tasks



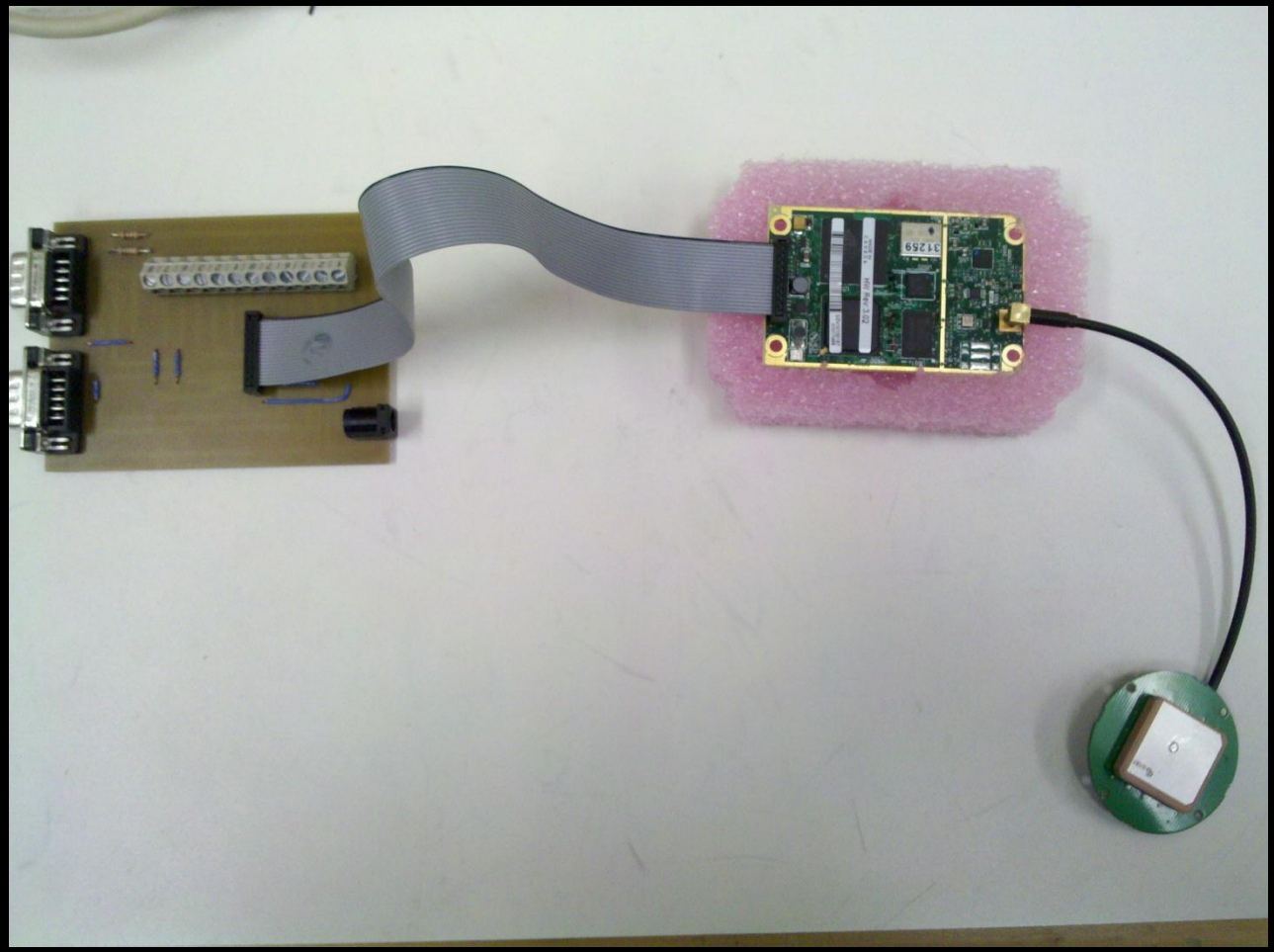
Hardware Tasks



Signal Breakout Board



The Hardware



High Level Software Architecture

3 baseline processing functions

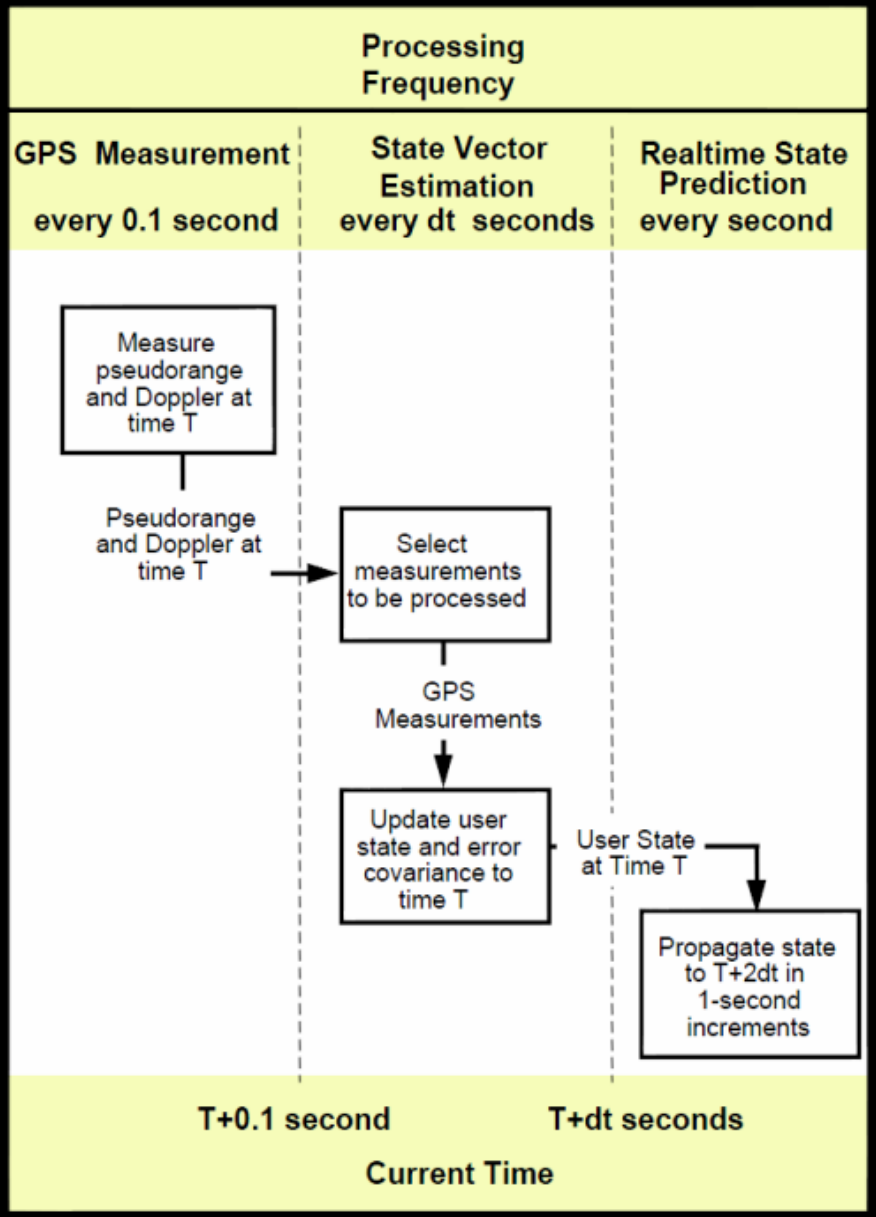
- Interface: Executed as needed to process receiver data, GEONS commands, and GEONS telemetry requests
- State Vector Estimation: executed every 30 seconds to propagate the spacecraft's current state estimates to the time and correct current state using GPS data

High Level Software Architecture

3 baseline processing functions

- Real-Time Prediction: Executed to propagate the current state estimate ahead in time to the requested output time, typically the next whole second

GPS/GEONS Processes

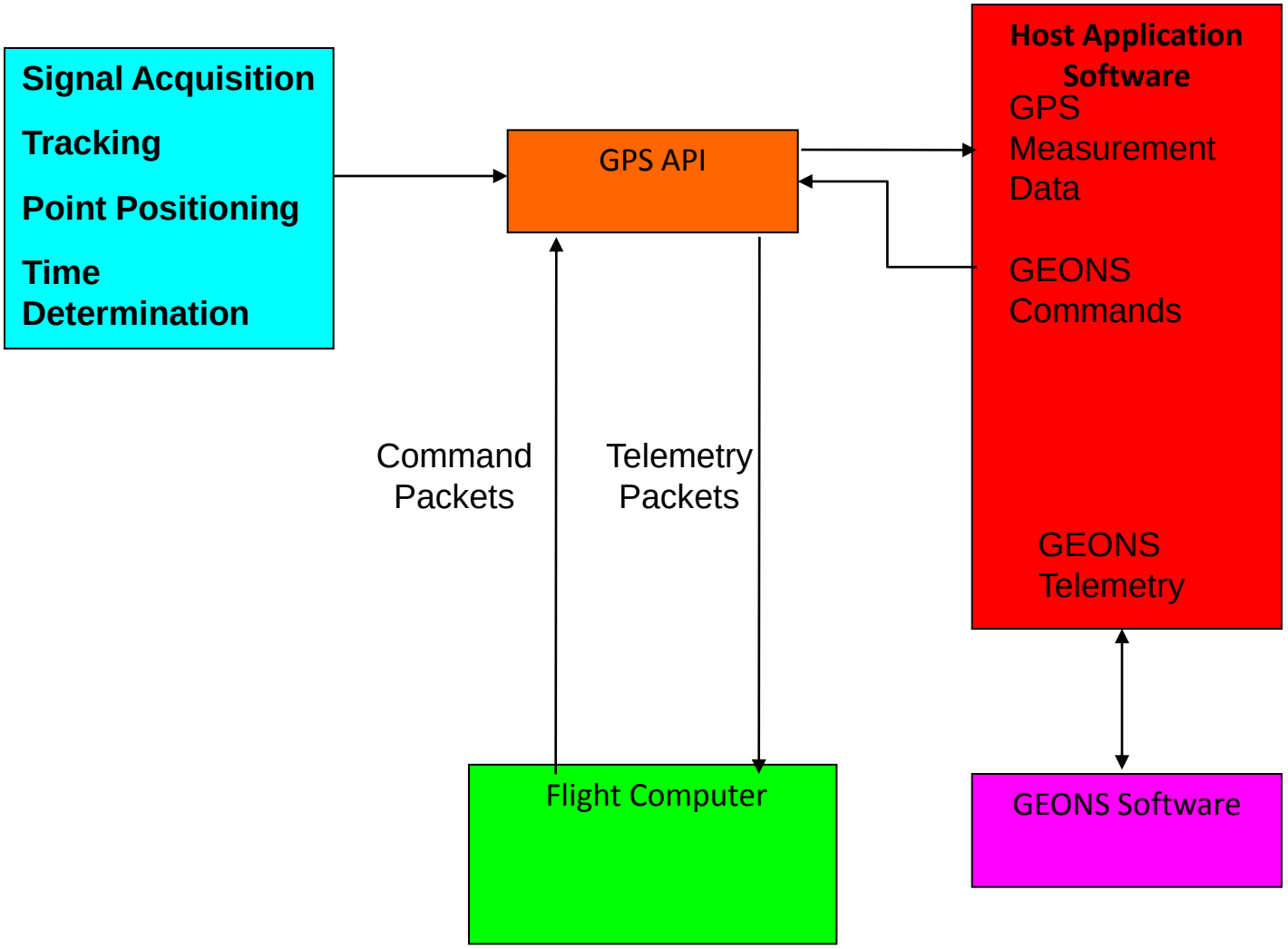


High Level Software Architecture

Components and Interrelationships

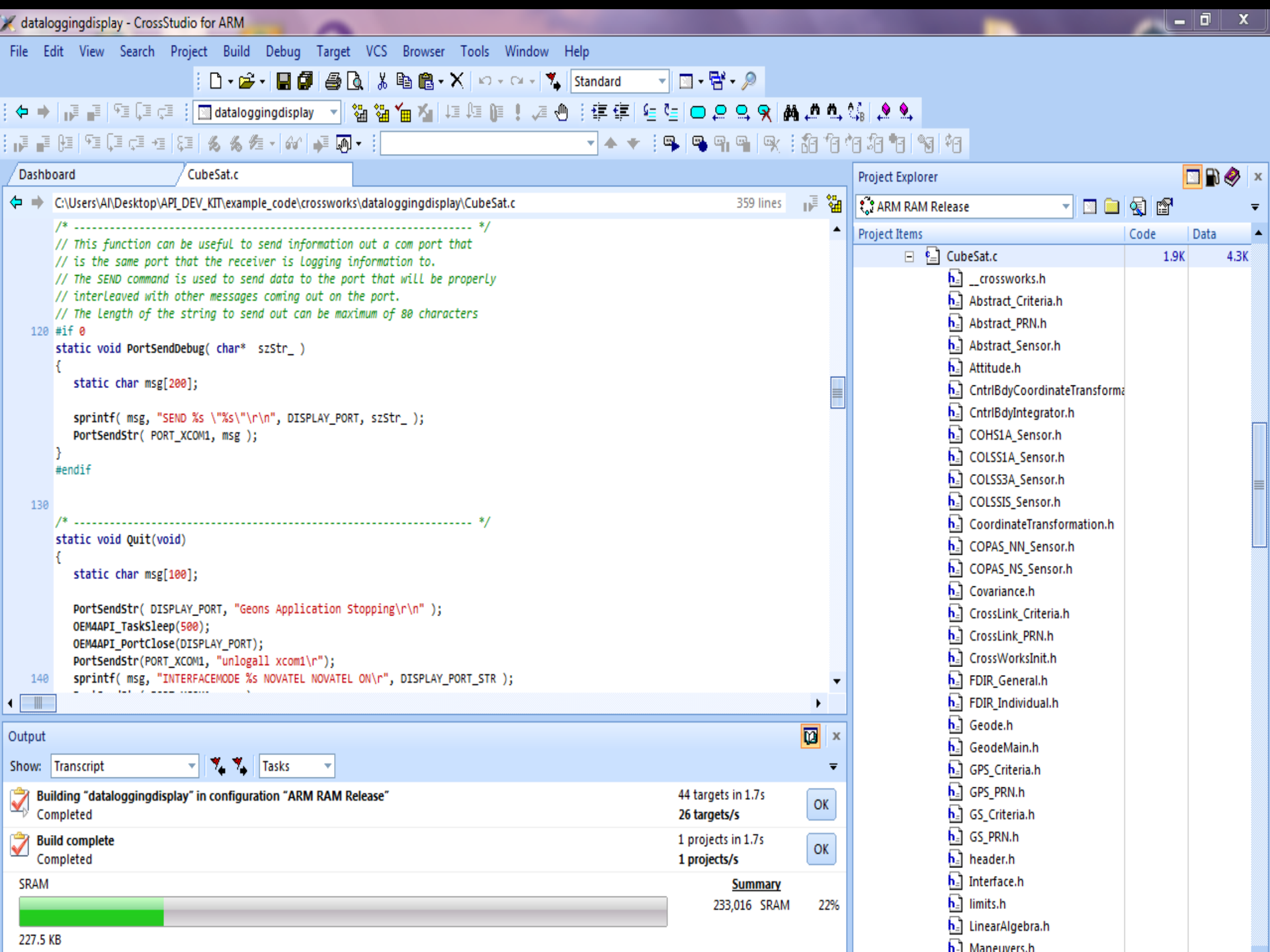
- GEONS, host application and API on the GPS receiver
- Flight control processes on the OBC
- The host application acts as an interface to the GEONS
- Host application communicates with and gets data from the GPS via the GPS API
- Flight control processes initiates its data transfers via the host application

Software Decomposition



Software Development Tool Chain

- Rowley and Associates CrossStudio for ARM 2.0. Software Development IDE Executable and Linking Format .elf
- TOSREC.exe - Converts the .elf to industry standard ASCII strings for ease of data transfer
- DATABLK.exe – Target platform (XScale), version data, data block address
- WinLoad.exe – writes DATABLK modified srecs to the receiver's flash memory

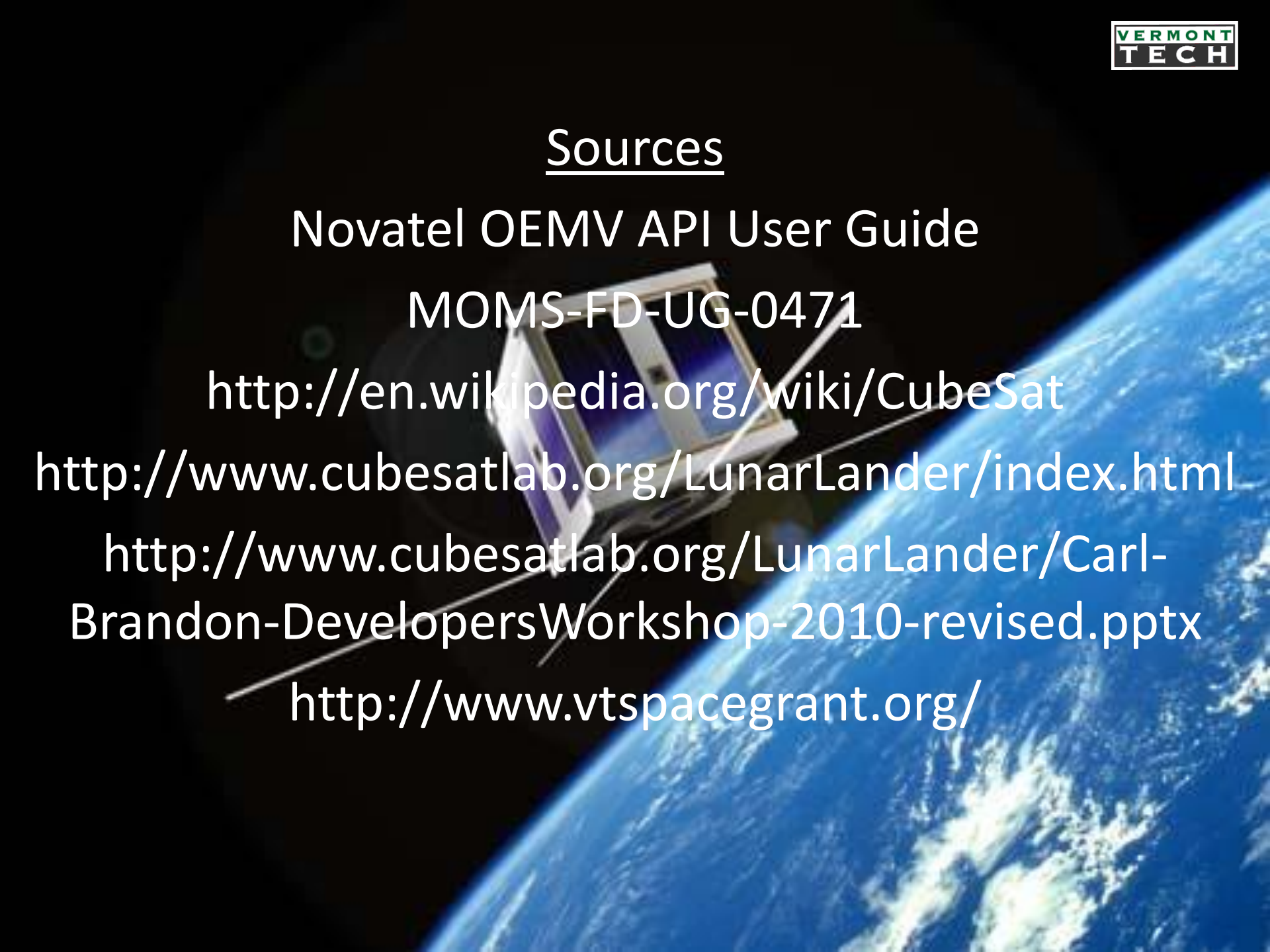

dataloggingdisplay - CrossStudio for ARM
File Edit View Search Project Build Debug Target VCS Browser Tools Window Help
Standard
dataloggingdisplay
Dashboard
CubeSat.c
C:\Users\Al\Desktop\API_DEV_KIT\example_code\crossworks\dataloggingdisplay\CubeSat.c
359 lines
/* ---------- */
// This function can be useful to send information out a com port that
// is the same port that the receiver is logging information to.
// The SEND command is used to send data to the port that will be properly
// interleaved with other messages coming out on the port.
// The length of the string to send out can be maximum of 80 characters
120 #if 0
static void PortSendDebug(char* szStr_)
{
static char msg[200];
sprintf(msg, "SEND %s \"%s\"\r\n", DISPLAY_PORT, szStr_);
PortSendStr(PORT_XCOM1, msg);
}
#endif
130
/* ---------- */
static void Quit(void)
{
static char msg[100];
PortSendStr(DISPLAY_PORT, "Geons Application Stopping\r\n");
OEM4API_TaskSleep(500);
OEM4API_PortClose(DISPLAY_PORT);
PortSendStr(PORT_XCOM1, "unlogall xcom1\r");
140 sprintf(msg, "INTERFACEMODE %s NOVATEL NOVATEL ON\r", DISPLAY_PORT_STR);
Output
Show: Transcript Tasks
Building "dataloggingdisplay" in configuration "ARM RAM Release"
Completed
44 targets in 1.7s
26 targets/s
OK
Build complete
Completed
1 projects in 1.7s
1 projects/s
OK
SRAM
227.5 KB
Summary
233,016 SRAM 22%
Project Explorer
ARM RAM Release
Project Items
Code
Data
CubeSat.c 1.9K 4.3K
__crossworks.h
Abstract_Criteria.h
Abstract_PRN.h
Abstract_Sensor.h
Attitude.h
CntrlBdyCoordinateTransforma
CntrlBdyIntegrator.h
COHS1A_Sensor.h
COLSS1A_Sensor.h
COLSS3A_Sensor.h
COLSSIS_Sensor.h
CoordinateTransformation.h
COPAS_NN_Sensor.h
COPAS_NS_Sensor.h
Covariance.h
CrossLink_Criteria.h
CrossLink_PRN.h
CrossWorksInit.h
FDIR_General.h
FDIR_Individual.h
Geode.h
GeodeMain.h
GPS_Criteria.h
GPS_PRN.h
GS_Criteria.h
GS_PRN.h
header.h
Interface.h
limits.h
LinearAlgebra.h
Maneuvers.h

Some Code

```
void geonsloop(unsigned int ignore)
{
  int i = 0;
  while(1)
  {
      get_gps_data();
      i++;
      if(i%READING_PER_VECTOR_EST == 0)
      est_vect();
      if(i%READING_PER_PROJ_EST == 0)
      project();
      OEM4API_TaskSleep(100);
  }
```

In Conclusion

- Hardware in place for future GPS/GEONS Software and Hardware development
- The entire GEONS Library is happily living on the GPS receiver
- More than enough memory remaining for expansions and additions
- Students interested in getting involved in the CubeSat Project: Dr. Brandon or Prof. Chapin
- Become a Partner: http://www.vtspacegrant.org/



Sources

Novatel OEMV API User Guide

MOMS-FD-UG-0471

http://en.wikipedia.org/wiki/CubeSat

http://www.cubesatlab.org/LunarLander/index.html

http://www.cubesatlab.org/LunarLander/Carl-Brandon-DevelopersWorkshop-2010-revised.pptx

http://www.vtspacegrant.org/

Questions?