Electronics Group Subsystem Plan

Stephanie Slowik
Erik Stockham
Maggie Sullivan
Jennifer Underwood

ElectroMagnetic Formation Flight Of Rotating Clustered Entities

Presentation Overview

Electronics Gp
•Ops Plan
•Communication
•Avionics
•Current Status

- Operations plan
  - System states
  - System layout overview
- Communications
  - Inter-vehicle data transfer
  - Ground station telemetry
- Avionics
  - Subsystem interfaces
  - Computer selection
- Current status
  - Group prototype design and testing

Operations Plan

Electronics Gp
•Ops Plan
•Communication
•Avionics
•Current Status

- System states
  - Normal operational states
    - Start-up
    - Spin-up
    - Position maintenance
    - De-spin
    - Shut down
  - Emergency operational states
  - Software update state
- Fault detection and resolution

Modularity

- Software modules can be designed for each defined system state
- Control Team software will vary greatly depending on the situation
- Procedural instructions for operator

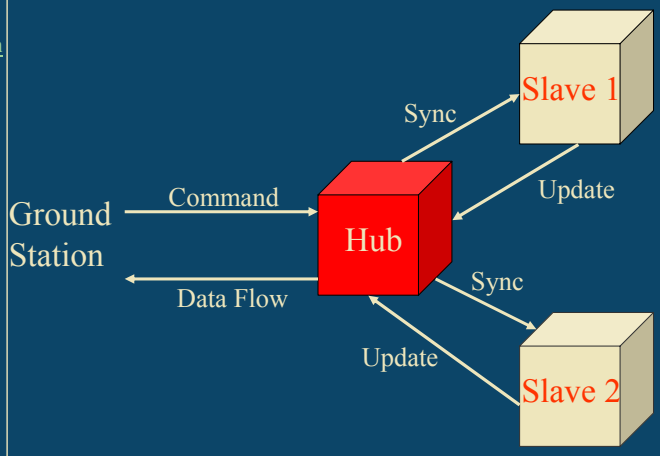
Communication

- Ground station to hub
  - Operator commands
- Cluster interaction
  - Time synchronization
  - System state updates
- Hub to ground station
  - Data transmission
  - System health monitoring

Communication Diagram

MIT Aero/Astro

Electronics Gp
- Ops Plan
- **Communication**
- Avionics
- Current Status



ElectroMagnetic Formation Flight Of Rotating Clustered Entities

Transmission Rates

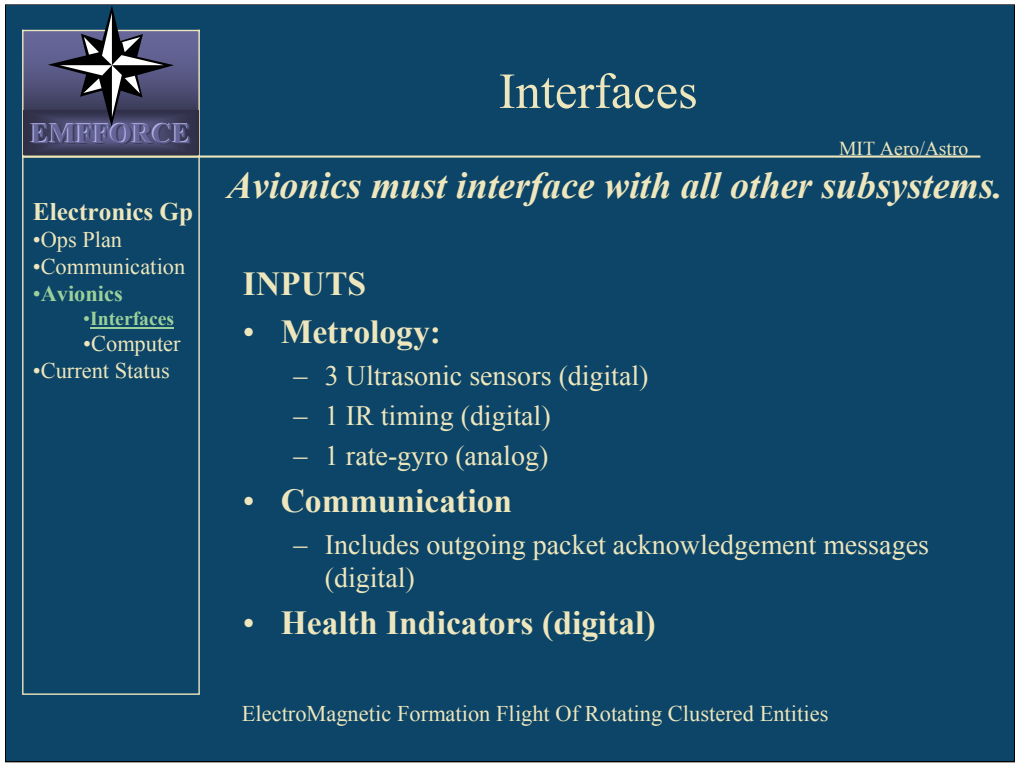
- Cluster Comm Channel
  - 6 variables in state vector (Control)
  - 4 variables to actuators (EM, RWA)
  - About 300 bits per cycle
- Ground Link Comm Channel
  - Undefined requirements
  - On the order of 300 bits per cycle

Equipment Required

- One TT8 computer per vehicle
- Antennae mounted near center of mass on top of vehicle
- Laptop required for ground station

Error Checking

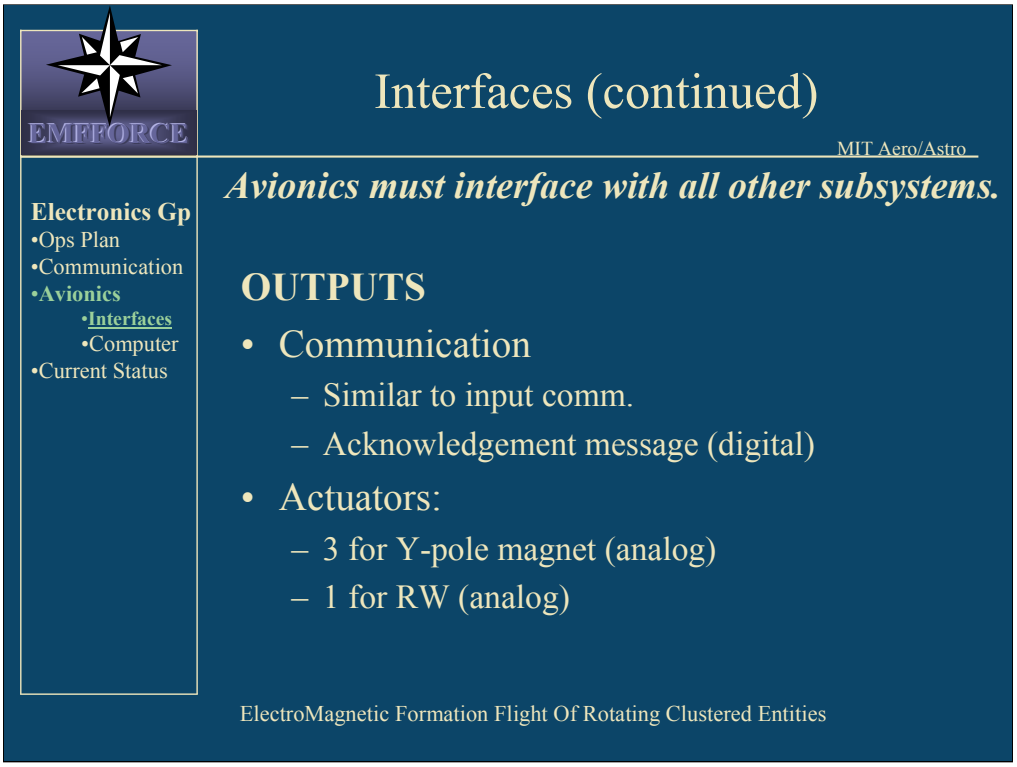
- Need to implement error checking in the TT8 software
  - Parity
  - Cyclic Redundancy Check
- Major focus of communications system design and lab work



The avionics subsystem must interface with all other subsystems. The following are predicted needs for input/output setup for the CDIO3 Team.

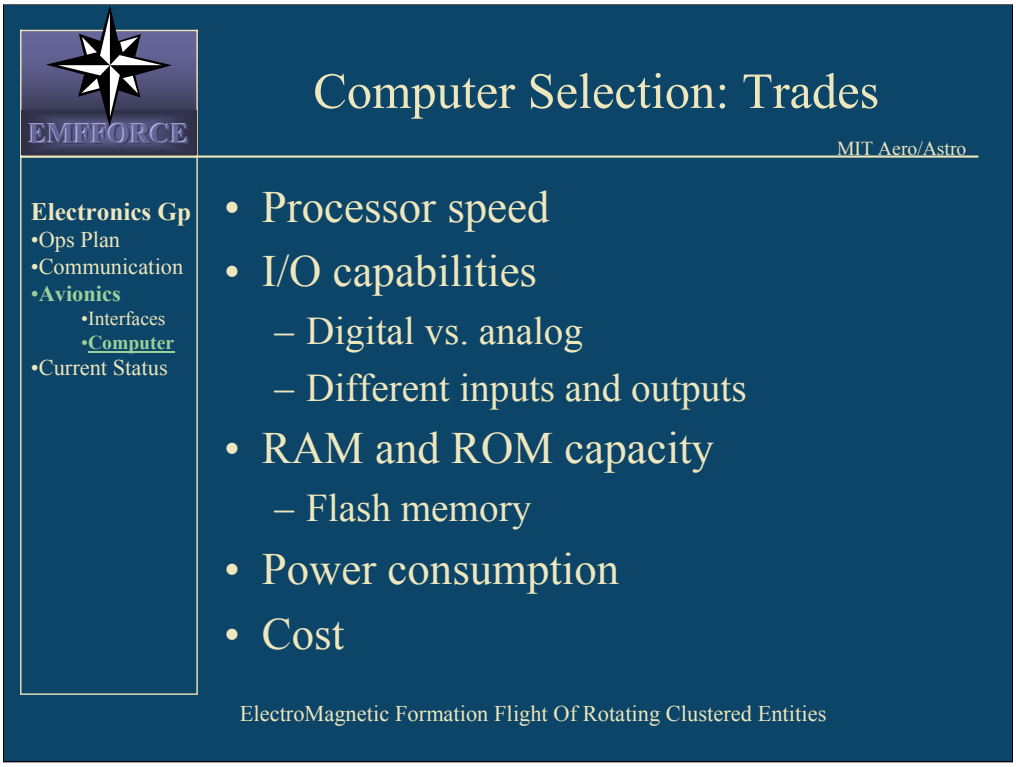
•I/O capabilities: require both analog and digital receiving and transmitting. Some information can be represented as binary signals; some must indicate levels and therefore requires analog transmission.

- •In: metrology (3x ultrasound, digital; 1x IR-timing, digital; 1x gyro-rate, analog), communication (1x incoming stream, digital - includes acknowledgment packets), health indicators such as low battery
- •Out: communication (1x outgoing stream, digital), actuators (3x magnet, analog; 1x RWA, analog)



•I/O capabilities: (as mentioned earlier)require both analog and digital receiving and transmitting. Some information can be represented as binary signals; some must indicate levels and therefore requires analog transmission.

•Out: communication (1x outgoing stream, digital), actuators (3x magnet, analog; 1x RWA, analog)



There are several characteristics to address when choosing a computer for this project.

•Processor speed

- •MIPS vs. FLOPS (fixed vs. floating-point calculations – reason for different metrics in table)
- •Estimating instructions required based on state and control matrix multiplications

•I/O capabilities: require both analog and digital receiving and transmitting. Some information can be represented as binary signals; some must indicate levels and therefore requires analog transmission. Refer to subsystems interfaces for more info.

•RAM: Random Access Memory for immediate use. Disappears with computer reset or power-out.

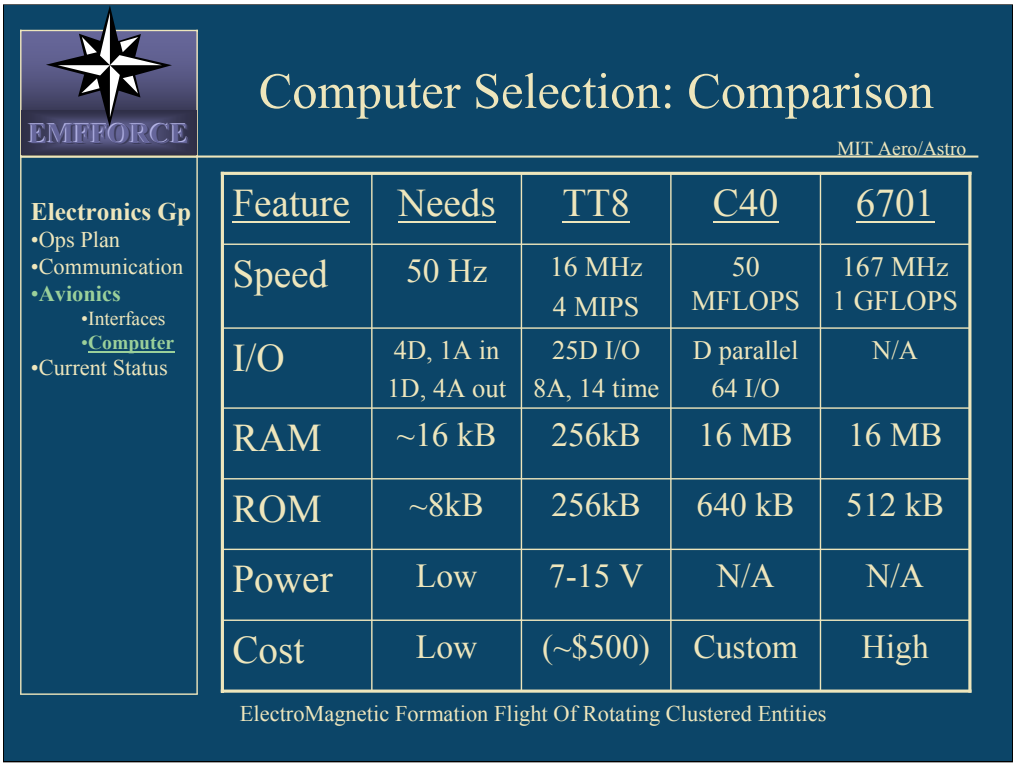
- •Estimate approximately 2kB for communication data, 1kB for control matrices. Round up to *16 kB* "to be safe"

•ROM: Read-Only Memory for storing programs and data. For our purposes we will use FlashMemory: re-writeable, ROM-like memory. Must be erased using a hard reset.

- •Estimate approximately 1000 lines for *one* test program – about 4kB to store. Double to *8kB* "to be safe"

•Power consumption: minimize to save battery life, increase available power for actuators

•Cost: minimize based on class budget



•The mission of our project is to demonstrate formation flight. To be efficient, we need a computer that will support the mission with as little extra cost, weight, power drain, and size as possible.

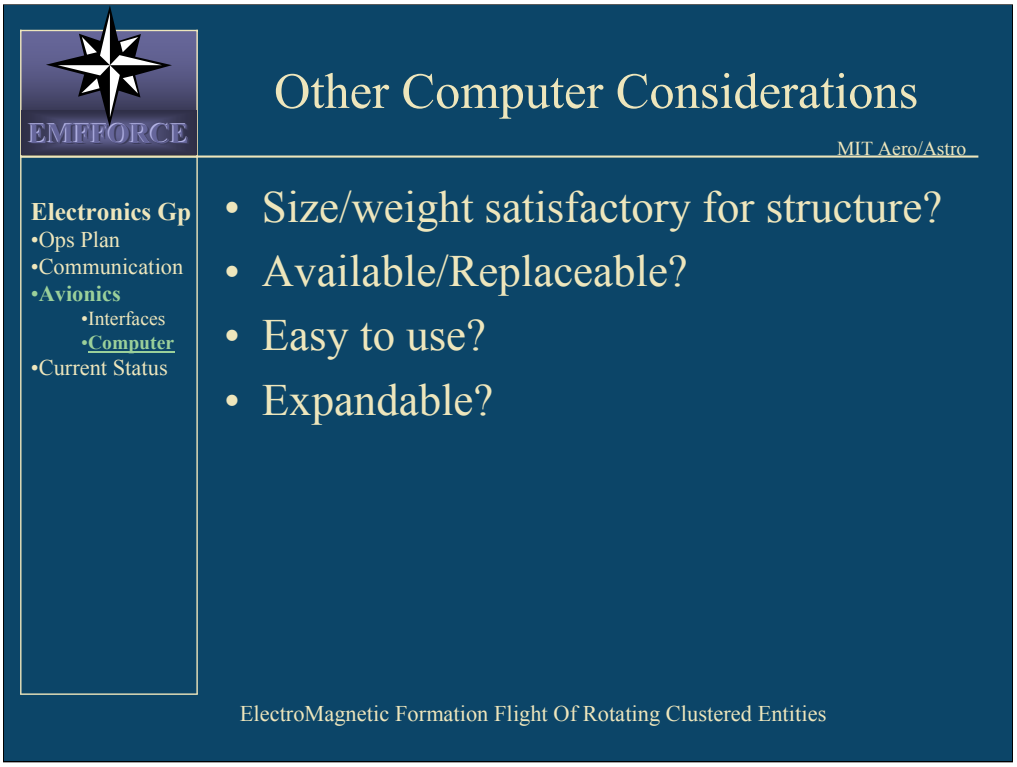
•To keep costs low, we first asked SSL about products available for our use. They have three processors on had: a 6701, C40, and the Tattletale Model 8. The table is the result of our research.

•The 6701 processor, on a Sundance SMT 375 board, was eliminated immediately. The processor speed and board capacity are well over anything we could possibly need for this project, and much of the board would be excess volume, dead weight and power draw.

•The remaining choice was between a C40 processor and the Tattletale Model 8.

- •Both the TT8 and the C40 are more than adequate for our estimated processing power
- •The TT8 comes with an A/D converter; we would have to build one for the C40
- •Both computers have more RAM and ROM [FlashMemory] than our projected needs
- •The TT8 draws less power than the C40
- •We have two TT8 computers immediately available for early prototyping, and 2-4 more which we may be able to use in the vehicles. We also have materials available to code the TT8, and several resources including both manuals and experienced users [Edmund].
- •There are fewer C40 board available to us, if any. The C40 is larger, heavier, and more power-hungry than the TT8.

•Our final decision was to use the TT8, because it is excellent for our needs as well as convenient for prototyping, which needs to be done *soon*.



There are several other items to consider in a computer beyond its processing power and functional capabilities.

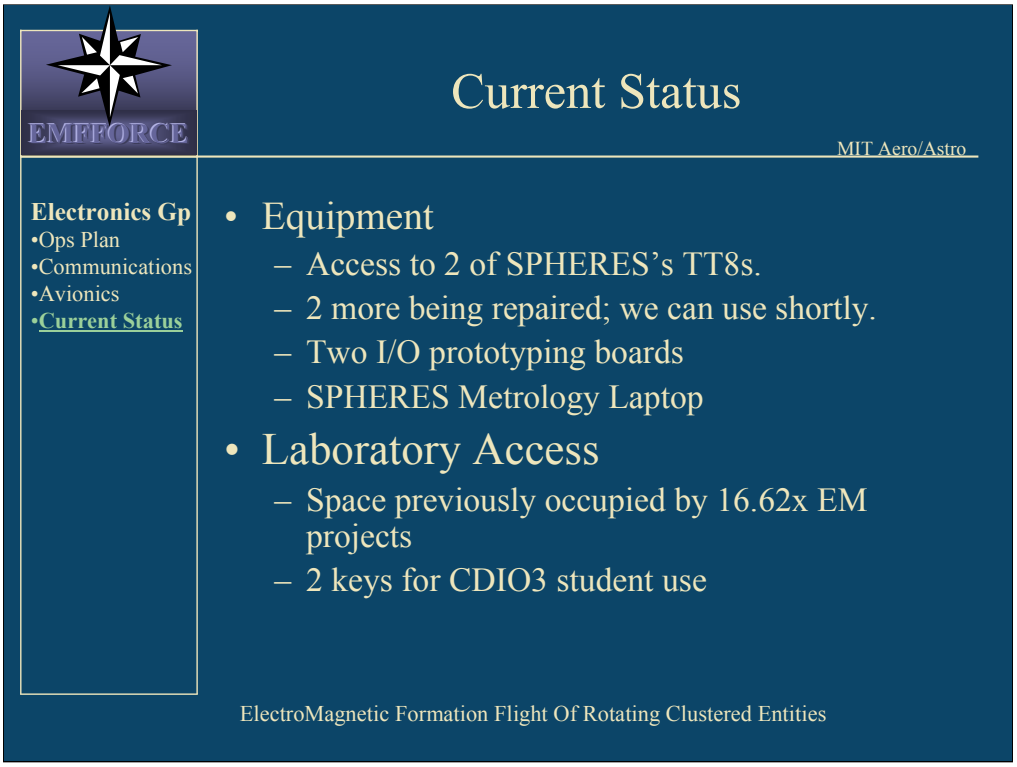
•Size and weight: the TT8 is 2" x 3" x 0.5" and weighs 1 oz. – smallest computer we considered

•Availability: there are several TT8 computers available for our use, including two for immediate prototyping. Also available in the SSL are all hardware and software needed to write code for and interface to the boards, plus technical manuals.

•Repair and replacement: TT8 models are repairable for far lower cost than that of purchase (~$60?); if necessary, a replacement Tattletale is $500.

•Ease of use: the TT8 can be coded in C. Most team members have some experience in C; they can help others who may need it. In addition, there are easy-to-use desktop coding environments (CodeWarrior and Motocross) that will enable fast and simple code compilation and loading of programs onto the board.

•Room for expansion: The TT8 would not be suitable for extensive expansion of the avionics system. To accommodate possible future expansion of the vehicle functions (and resulting expansion of computing needs), we plan to use existing standards for hardware support (e.g., cables and ports) as well as operating voltages and other interfacing requirements.



Where we are:

Equipment:

•We have obtained 2 of SPHERES's TT8 computers. Two more are being sent back for repair and will also be available for our use shortly.

•We have also obtained use of the SPHERES metrology laptop.

•We have usage of 2 of SPHERES's old smaller I/o prototyping boards.

Space Access:

•The portion of the SPHERES/SSL laboratory (37-372) that was formally occupied by the 16.62x EM groups has now been allocated for CDIO3 use.

•Laila has obtained 2 keys for team use as well. These keys will be given to the Systems Group on 18 Apr 2002, and they will decide how the keys will be distributed.

Current Status (continued)

- Tutorial
  - Taught by Edmund: 18Apr2002, 1300h.
  - Attended by Electronics Group.
  - Tattletale8
    - Loading of programs
    - CodeWarrior and MotorCross.
- Prototyping Test and Design
  - Tests include:
    - Blink an LED
    - Signal Reproduction
    - Clock interrupt
  - Testing begins this week

Current Status (continued)

- Communications system prototyping
  - Develop plan for time-sharing on the two channels
  - Transmit data through the TT8
  - Record data from the TT8
  - Study noise level in data
    - How much error checking is sufficient?