P09201
Generation II Software for Underwater ROV
The Team

Colby Gregory  Project Manager, CAD Design, Hardware Interfacing
Bryan Lavrich  Firmware Development, Microcontroller Interfacing
Florian Letourneux  PCB Design, CAD Design, Communication interfacing
William Stanton  Digital Electronic Design, PCB Design, Hardware Interfacing
Dan Brandt  Software Interfacing
Jon Savage  GUI Development
What we’ll talk about

- Our Mission \ Scope
- Our design specifications \ Customer needs
- Our concept
  - User
  - Layers of abstraction \ device communication
  - Hardware interface
- Our risks
- State of design
- Our schedule \ Timeline
- Milestones for 20082
What we’re about

Our project will create a control system that allows the user to interface with an underwater ROV in real time. An ancillary requirement for providing this system is that it must also enable and streamline the communications between the current and future individual components of the ROV.
Project Scope

- Provide a user-friendly graphical interface for controlling the lighting and thrusters of the ROV.
- Define software communication protocols to be used by all current and future UUV modules.
- Define hardware communication interface to be used by all current and subsequent UUV modules.
- Design a bridge for easy networking of modules -- including other bridges.
# Our Customer Needs

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Software</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Customer Needs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All components must fit in standard housing already developed</td>
<td>Open Source Software Implementation</td>
<td>Interfaces with other projects (to some extent)</td>
</tr>
<tr>
<td>Minimize power efficiency and heat dissipation</td>
<td>Must accept user input through a GUI</td>
<td>Design documentation is a must</td>
</tr>
<tr>
<td>Develop testing apparatus to show operation of software and interface</td>
<td>Software should control light intensity, spectrum, thrust direction and speed</td>
<td>Both HW and SW should be able to control a slave bridge</td>
</tr>
<tr>
<td>Compatible with RP projects</td>
<td>Control both surrogate lights and thrusters simultaneously</td>
<td>Anticipate the migration to land vehicles, not necessarily plug and play compatible</td>
</tr>
<tr>
<td>Must be scalable, must be able to add additional bridges to the interface</td>
<td>Compatible with RP projects</td>
<td>Leave the door open for compatibility, establish commonalities between projects</td>
</tr>
<tr>
<td>Design should be modular, allow for adding of components easily</td>
<td>GUI incrementally adds modules as devices are attached</td>
<td>HW/SW Command prioritization</td>
</tr>
<tr>
<td>RS232 interface to surface</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend**

- **High Priority**
- **Medium Priority**
- **Low Priority**
Our Concept

- Starts at a GUI and a user input
- Users command is then wrapped and sent down the communication and interfacing layers
- Command ends up at interface board, routes to device
- This is a two way path
Our Concept - User

1. Functional
2. Intuitive
3. Aesthetically pleasing
Our Concept - data

- Joystick input
- Java - cloud - more abstract cloud - even more abstract cloud - confusion - assembly - VOLTAGE!
  - Basically, this means we’re using a ‘layered’ architecture
- Communication protocol
  - RS232
  - RS485
Speaking of communication

- Protocol, protocol, protocol
  - Get data encoded and decoded reliably
- Multi-layered protocols are in common use throughout digital communications
- Low level protocol that all devices MUST strictly adhere to. This enables a basic level of communication
- High level protocol(s) that are device and software specific. These enable meaningful data to be exchanged
Our Concept - The Bridge

- Our hardware interface between software and the device
- 1 Input - 10 Outputs, can be made up to 16
- Completely expandable to other bridges
- RS232 to surface
- RS485 to devices and other bridges
- Uses a time sharing algorithm to allocate a fair amount of time to each module
- Programmed through JTAG
Our Risks

- **SW Risks**
  - Communication problems
  - Coding standard violations
  - SW/HW integration

- **HW Risks**
  - Prototyping
  - Noise
  - PCB layout
  - Drive capacitance
Where we stand

- GUI is implemented
- Joystick Interface:
  - Code works with a multi-axis controller on Mac OS X
  - May need minor tweaks to work with a joystick
- Communications:
  - ROV Communications Protocol fully implemented
  - Bridge Communications Protocol partially implemented
- Bridge Programming
  - UART code fully implemented (will need tweaking to make it more configurable)
  - SPI interface code (for off-chip UART) fully implemented
  - Control code for external chip not yet implemented
  - Timer code partially implemented (for time-sharing).
  - Packet queuing partially implemented.
- Bridge HW
  - Parts arrived this week, prototyping has begun
Where we’re going

- By the beginning of MSD II
  - Software:
    - Interfaces have begun testing communication
    - GUI code talks to other layers of abstraction
    - Fully operational SW/HW interface between user and port in Mac OS X, other systems have begun
  - Hardware
    - Prototyping complete, PCB design begins
### Where we’re going

<table>
<thead>
<tr>
<th>Date</th>
<th>HARDWARE</th>
<th>Microcontroller</th>
<th>SOFTWARE</th>
<th>SW/HW Interfacing</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/5/08</td>
<td>Prototyping complete, PCB Design commences</td>
<td>Continue testing assembly code</td>
<td>Implementation of optional components has begun.</td>
<td>One OS is complete and functional</td>
</tr>
<tr>
<td>12/19/08</td>
<td>PCB design complete + Ordered</td>
<td>Integration with PCB has begun</td>
<td>All optional components have been implemented and talk to each other correctly.</td>
<td>Other OS’s have funcionality</td>
</tr>
<tr>
<td>1/16/09</td>
<td>PCB Implemented</td>
<td></td>
<td>Proof of modularity concept has been integrated</td>
<td>DONE: Debug</td>
</tr>
<tr>
<td>1/30/09</td>
<td></td>
<td></td>
<td>SYSTEM LEVEL DESIGN PRESENTED TO CUSTOMER FOR REVIEW</td>
<td></td>
</tr>
<tr>
<td>2/13/09</td>
<td></td>
<td></td>
<td>DEBUG \ ADD FUNCTIONALITY</td>
<td></td>
</tr>
<tr>
<td>2/27/09</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Questions and Answers

Ask me the easy questions
Ask them the hard ones