# Specific Design Details: Controls System

The controls system is one of the most crucial subsystems in the Robosub UAV. This will be a single PCB with a microcontroller, voltage regulator, UART/USB communication traces, slot connector for attaching to the backplane, and a PWM signal output to each of the 8 Electronic Speed Controllers (ESC) on the sub. Additionally, this board will require a JTAG programming interface to load code onto it. As such, the board will contain:

* STM32F767 ARM Cortex M7 microcontroller
  + PID controls algorithms, first modeled and tuned in Matlab and Simulink
  + UART transmission interface and received data parser
  + Drivers for PWM control
  + Primarily interrupt-based
* Connector Pads for Molex ExtremePower Edge Connector
* UART-to-USB via FT232 IC
* JTAG header (for use with the ST-Link/V2 programmer)
* LEDs for PWM status, power status, communication status (debugging purposes)
* TI LM3940 Linear Regulator (5V to 3.3V)

## 1. Theory of Operation

Given sensor input from an IMU and DVL, with measurements of heading, pitch, roll, depth, and forward tangential velocity, and a desired reference value for the same, the microcontroller will execute the designed mathematical algorithms and output a pulse width modulated signal to each of eight electronic speed controllers in order to ensure desired operation of the vehicle in an aquatic environment.

Instructions on the development of the controller/compensator can be found in the accompanying file “system\_constants\_and\_controller\_design.m”

## 2. Proof of Concept Results

* **Code:** For PoC, we implemented UART and PWM drivers on the STM32 F7 we’ll be putting on our board, and showed that we can use UART to control our motors very practically. This code will simply be ported to the controls board once fabricated.

## 3. Design Notes

* The boards have a few flaws in them - easily fixed, but it is best to allow Daniel to assist in populating them. Additionally, we have found it is easiest to use a Voltera to print solderpaste onto the controls boards (there are several backup boards that we have purchased that will need populated).
* Ensure that the PC can find the FT232 serial port and that you know the naming conventions for your particular PC. If no serial port shows up, power cycle the controls board.
* Tuning of the controls algorithms can be a tedious process, and detailed notes on how to perform this task are included in the code comments in the “Controller Development” directory.

## 4. Bill of Materials

Bill of Materials for the controls system (Detailed bill of materials included in Appendix 1):

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Price ($) | Quantity | Subtotal ($) |
| Microcontroller | 15.98 | 3 | 47.94 |
| Development Board (STM32) | 50 | 1 | 50 |
| 2-layer PCB | 43 | 1 (x3 revisions) | 129 |
| FT232 | 4.50 | 3 | 13.50 |
| ST-LINK | 22.61 | 1 | 22.61 |
| Misc. Parts & Connectors | - | - | 60 |
| Shipping | 40 | - | 40 |
|  |  | **Total ($)** | 363.05 |