

# Thrusters

## Legacy

Previous iterations of the sub have used Seabotix thrusters, which are DC motors. We've controlled them using Sabertooth 2×12 motor drivers made by Dimension Engineering. These controllers communicated via a serial interface, and thus were directly controlled by the computer.

## Current

The new submarine is using AC T200 thrusters by Blue Robotics, which are controlled via the BlueESC, which is integrated directly within the thruster. These controllers communicate via an I<sup>2</sup>C interface, therefore an intermediate microcontroller is used to translate serial commands from the computer to I<sup>2</sup>C.

## T200 + BlueESC Information

Official documentation can be found on the [BlueRobotics website](#). The most pertinent information is copied here for quick reference. There are two cables coming off of them. The thicker cable is power, and the smaller cable is signalling.

### Wiring information:

Power Cable:

- Red - positive power
- Black - ground

Signalling:

- Yellow - PWM control (tie to ground if using I<sup>2</sup>C)
- Green - I<sup>2</sup>C clock
- White - I<sup>2</sup>C data
- Black - ground

### I2C Interface

The thrusters are controlled by sending a throttle command between -32767 and +32767, with 0 is stopped. There is no deadband on these thrusters, however the relationship between throttle and thrust is believed to be quadratic (need to verify). The controllers have a built-in safety timeout, if no throttle command is received for 1-2 seconds (need to verify), the motors will automatically stop. If

further commands are sent, they will be ignored. To activate the thrusters again, a throttle value of 0 must be sent.

The BlueESC also can provide us with information concerning motor RPM, voltage, current, and temperature.

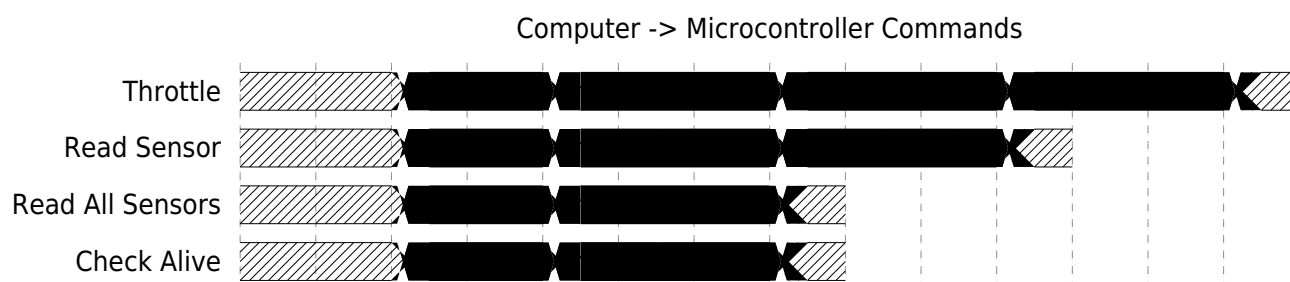
When shipped from the factory, the thrusters all use the I<sup>2</sup>C address of 0x29. To change the address, the firmware must be re-flashed. We have programmed our thrusters with the addresses 0x30-0x37.

## Computer-Thruster Interface

The microcontroller code receives incoming commands from the computer and dispatches them. Some commands result in the microcontroller sending a response back to the computer. Commands and responses are wrapped up via our Packetizer class, so all diagrams are simply describing what should be in the “data” section of a packet.

### Computer to Microcontroller

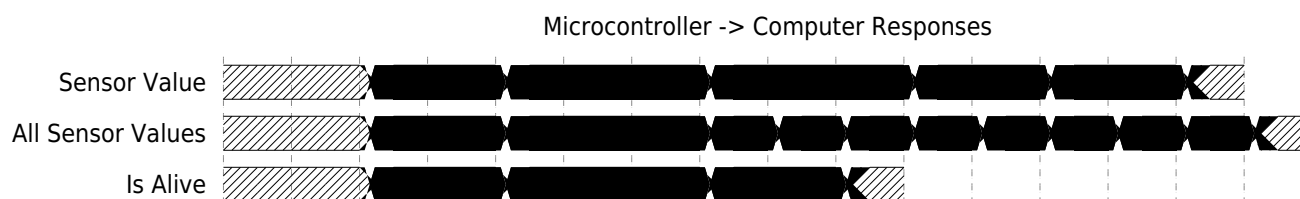
The following types of commands should be accepted by the microcontroller:



The first two commands are the basic interface for controlling the thrusters and getting information. There is the third macro command for reading all the registers of a thruster. There is no macro command for talking to all thrusters because this would require the microcontroller to know what thrusters exist/are active on the sub, which would complicate the code. The final command is a method for checking if a thruster is alive and responding, it will return a bool true/false value.

### Microcontroller to Computer

The computer should expect to receive the following packets from the microcontroller:



The IDs for responses correspond to the IDs for the commands as described in the previous section. The “Read all Sensors” response contains 8 bytes of data (B0-B7), which are described below:

- Byte 0: pulse\_count\_h
- Byte 1: pulse\_count\_l
- Byte 2: voltage\_h
- Byte 3: voltage\_l
- Byte 4: temperature\_h
- Byte 5: temperature\_l
- Byte 6: current\_h
- Byte 7: current\_l

From:

<https://robosub.eecs.wsu.edu/wiki/> - **Palouse RoboSub Technical Documentation**

Permanent link:

<https://robosub.eecs.wsu.edu/wiki/legacy/2016/ee/thrusters?rev=1473810163>



Last update: **2016/09/13 16:42**