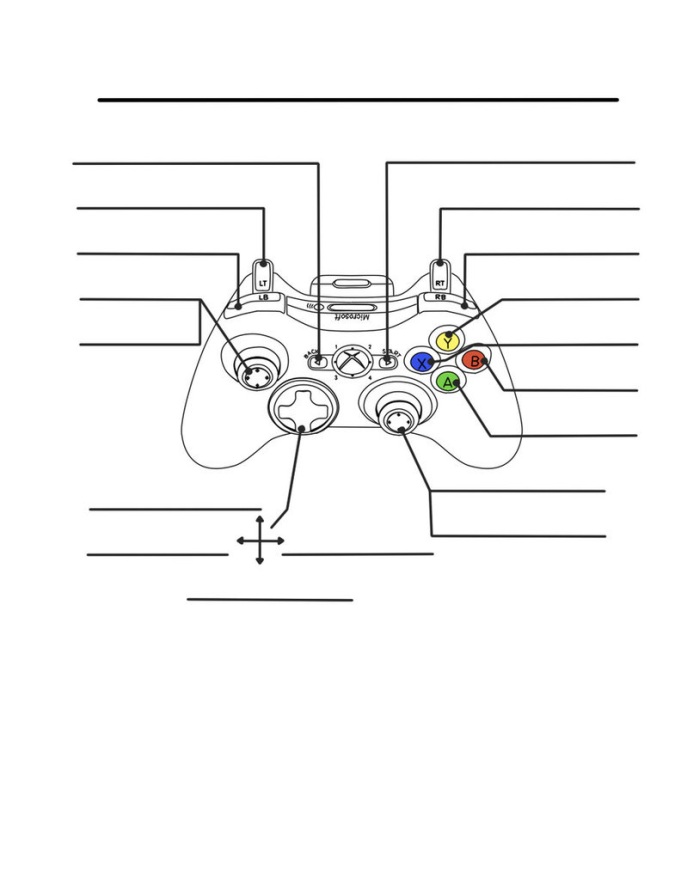
**MSLED Surface Control software v 2.0**

Features:

* Video recording, 30 fps for recorded video, ~15-20 fps while recording (FFDshow compression)
* Screenshots, can be taken with Xbox controller.
* Data logging, logs various values in real time. Click LOG DATA on the Dashboard to enable. (Logs as csv)
* CTD support added, polls data in real time
* Refined GUI, a more visual approach
* Blocking loop to avoid momentary communication breaks
* Reduced number of threads, more efficient
* Thrust inversion feature
* Angle based servo control

Controls:

XBOX controller mapping diagram

None None

Capture frame Turbo

Camera focus Camera focus

Fin control Hide HUD

Fin control Take screenshot

LEDs on

LEDs off

Buoyancy servo Thruster forward

Thruster reverse

Buoyancy servo

Running the code and troubleshooting basic issues:

1. Code generally present on desktop. If needed, go to github.com/msled/EERILControlSystem, and download the code as a zip file. Extract and open the EERILControlSystem.sln file.
2. Press F5 from Visual Studio.
3. No camera shown in the main window? Check communications and check that wireless is not turned on. That can interfere with the transceiver.
4. Program crashing without showing video? Try commenting out offending lines wherever the exception is shown.
5. Missing values on HUD? Serial communication problem, or some sensors might have gone bad.
6. No pitch, roll changes on the HUD? Switch to NO BUOYANCY mode and back to FULL.
7. Fins hitting each other? Pull back the fin range slider in the dashboard a little bit.
8. Servos not centered? Go to VideoDisplayWindow.xaml.cs

Search for the part that looks like this

if (pressed)

{

deviceManager.ActiveDevice.FocusPosition = 90;

}

else

deviceManager.ActiveDevice.FocusPosition = 51;

The not-being-pressed (else) part would be the center position, so edit it as per requirement. Check operations, and repeat if needed: it will be mostly a trial and error thingy, unless the position is known with LABVIEW.

1. CTD values off? Visit CTDConversion.cs, and try to subtract/add some offset in the Sal/ExtTemp/Depth values.

Main modules in the software:

Devices/MSLED.INO : Arduino code for the Atmel processor on the CDH board. Performs main low level operations, direct interfacing with the hardware etc. Controls the various motors, sensors and communicates with the surface control software.

How to modify the existing code on the CDH:

1. Open Arduino 1.0.2 in Program Files (x86), open the exe to enter the IDE, open required files.
2. Make modifications
3. Connect AVR Dragon to the ICSP header in the middle of the CDH board (use female-female jumper cables). Pay attention to the dot (1st pin) and the last pin, all pins should correspond with each other between the Dragon and the CDH.
4. Make sure no peripherals are connected to the CDH and power CDH on.
5. Connect Dragon to the USB port on the computer.
6. Click on UPLOAD in the Arduino IDE. If it doesn’t work, press and hold shift, and then click on upload.
7. The program will be compiled and dumped into the processor.
8. Turn off power for CDH, disconnect the jumper cables and unplug the AVR Dragon.
9. The new program will be run when the CDH is powered on the next time.

Tip: If servo center positions need to be modified, and if that cannot be done with the desktop software, the center positions are contained in the neutralize() function in the Arduino code. The modification should be done in neutralize() and the focus()/buoyancy() functions as needed.

EERILControlSystem:

The heart of the surface control software. Contains various submodules for communications with the CDH board, some code that are the APIs for camera control, image processing, Xbox controller support, etc.

Main files:

v4/Device.cs : Contains code which communicates with the CDH. Careful while modifying this file. It contains all the serial communication commands, and various other little things like thrust inversion, etc.

Camera/\*.\* : Camera APIs for image processing, socket control etc. Need not be modified unless major workarounds need to be done

AviWriter.cs : Video recording module. If any problems occur with the video recording, do not record video. If the program does not start, and crashes to some point in THIS PARTICULAR file, please change the

videoWriter = new AVIWriter("ffds");

line to

videoWriter = new AVIWriter("DIB "); (note the space)

Controller.cs: XNA framework for the Xbox controller. No key bindings are done here, this is just to teach the software about the controller. Nothing important.

CTDConversion.cs : Converts the raw binary values we get from the CTD into readable Temp, Depth and Salinity values. A thousand conversion constants and complex mathematical equations are used here. Please modify if the calibration needs to be changed for some reason.

DashboardWindow.xaml(.cs): .xaml is the window design file, and .xaml.cs is the underlying code. The axis binding for the fin/thrust control is done here.

Deployment.cs : Initializes deployment. Creates all necessary directories and starts off the mission.

VideoDisplayWindow.xaml(.cs): Let’s call this the heart within the heart. Performs all major activities of the ground station software.

* Receives all the serial stuff from Device.cs (sensors, CTD etc) and makes them ready for display
* Takes care of the controller key bindings, for focus ring control, buoyancy etc.
* Logs important data every one second
* Takes screenshots if needed

If program crashes to any specific spot in this file, try commenting out the lines. If it doesn’t work, then an in-detail analysis would probably be needed.

HeadsUpDisplay.cs:

All the visual parts. Reads the IMU data and renders the pitch gauge and the compass, displays all the values in readable 0.00 format, renders the depth meter, battery bar etc. Problems? Comment out the offending lines and see what happens.

**External dependencies:**

FFDShow codecs, downloadable at

<http://www.afterdawn.com/software/audio_video/codecs/ffdshow.cfm>

If on a different computer, the compression settings will be located in FFDS folder in the code folder. Double click to load the settings.