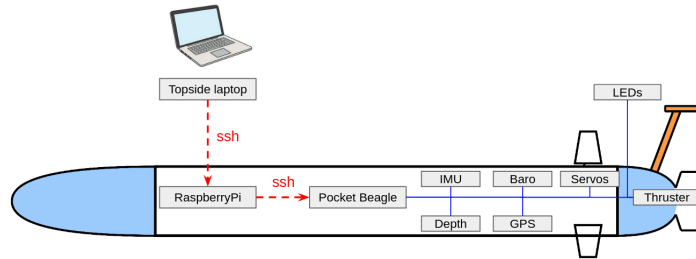


Lab 2b - Unit Testing Sensors and Actuators

2.S01 Introduction to Autonomous Underwater Vehicles



Spring 2026

Supun Randeni, supun@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139

1	Overview and Objectives	2
2	Unit testing sensors	2
2.1	Depth sensor	2
2.2	Barometer sensor	3
2.3	Attitude and heading reference system (AHRS)	4
2.4	Global Positioning System (GPS)	6
2.5	Self checkoff assessment	7
3	Unit testing actuators	7
3.1	Control surfaces	7
3.2	Main propulsion thruster	8
3.3	Light Emitting Diode (LED) indicators	8
3.4	Self checkoff assessment	10
4	Shutting down embedded computers	10
5	Assessment	10

1 Overview and Objectives

- Learning about the core sensors and actuators used in SeaBeaver III AUVs
- Mastering how to login to the vehicle’s embedded computers
- Learning how to unit test individual sensors and actuators to ensure they are in working order
- Gaining an understanding of what each sensor measures

2 Unit testing sensors

As shown in Figure 1, the PocketBeagle serves as a hub that interfaces with vehicle sensors and actuators. Throughout the AUV’s life; that is, starting from initial breadboard design phase to printed circuit board (PCB) testing, pre-vehicle assembly, post-vehicle assembly and during in-water operations; it is necessary to verify that each sensor and actuator functions properly as a standalone unit. To support that, we have developed a set of low-level test programs (i.e. executables) to unit test individual components.

These executables are installed to the directory: `/usr/local/bin`. This directory is in the PocketBeagle’s shell path. Therefore, you do not have to change the directory (`cd`) to this location to run these executables. You can run them from any directory as long as you are logged in to the PocketBeagle.

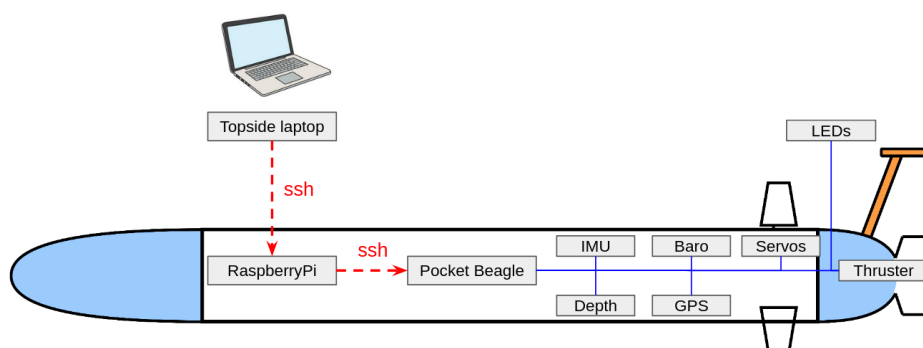


Figure 1: Sensors and actuators of SeaBeaver III AUVs are connected to the PocketBeagle

First, you will need to SSH into the PocketBeagle. If you do not remember how to do that, please refer to lab: *How to login to a SeaBeaver III AUV*.

2.1 Depth sensor

The depth sensor of an AUV measures the vehicle’s current depth, by measuring the external hydrostatic water pressure and converting it to depth, using the equation:

$$depth = P/(\rho g)$$

where P is the measured hydrostatic pressure, ρ is the density of water, and g is the gravitational acceleration). In SeaBeaver III AUVs, we use a BlueRobotics Bar30 depth sensor: <https://bluerobotics.com/store/sensors-cameras/sensors/bar30-sensor-r1/>

This sensor is connected to the PocketBeagle via the Inter-Integrated Circuit (I2C) interface. When the depth sensor is plugged in to the frontseat board, you can run the `test_br_depth` program to test if the depth sensor is connected to the PocketBeagle, and is in proper working order:

```
$ test_br_depth
```

If the depth sensor is working as expected, you will see an output similar to this:

```
debian@beaglebone:~$ test_br_depth
Acquired PROM contents
crc_data: 28673, c1: 30169, c2: 29708, c3: 18186, c4: 18854, c5: 28659, c6: 26886
Raw temp: 7386608, raw pressure: 4304880
dT: 4716612, ti: 49904
Depth: -0.09 m, Pressure: 1004.10 mbar, Temperature: 21.59 C
Raw temp: 7387120, raw pressure: 4305392
dT: 4716612, ti: 50416
Depth: -0.06 m, Pressure: 1007.10 mbar, Temperature: 21.61 C
Raw temp: 7387632, raw pressure: 4305392
dT: 4716612, ti: 50928
Depth: -0.06 m, Pressure: 1007.10 mbar, Temperature: 21.63 C
Raw temp: 7387120, raw pressure: 4305392
dT: 4716612, ti: 50416
```

Press `ctrl+c` to exit. If the depth sensor is not properly connected or faulty, you will see an output similar to this:

```
debian@beaglebone:~$ test_br_depth
ERROR: in rc_i2c_send_bytes, system write returned -1, expected 1
ERROR: Can not send reset
```

Troubleshooting:

- Make sure that the depth sensor is plugged into the J4 connector of the frontseat board.
- If you have a different depth sensor that you know works (perhaps a depth sensor that has been tested by a classmate), try it. This will help isolate the problem.
- Talk to an instructor.

2.2 Barometer sensor

SeaBeaver III AUVs come with an internal barometer that measures the pressure inside the pressure hull. As you will learn later in this class, observing internal pressure is important to check if the pressure hull has leaks (*That is, once you assembled the vehicle, you will suck the air inside the pressure hull out to create a vacuum. If the vacuum leaks at a rate higher than what is acceptable,*

that means your pressure hull has a leak somewhere. You will then have the opportunity to track it down and fix it).

In SeaBeaver III, we have a MPL3115A2 barometric pressure sensor (https://cdn-shop.adafruit.com/datasheets/1893_datasheet.pdf) connected to the PocketBeagle via I2C interface. You can unit test the internal barometer using the `check_vacuum` program:

```
$ check_vacuum
```

If the barometer functions as expected, you will see an output similar to this:

```
debian@beaglebone:~$ check_vacuum
Looping until I connect to StackUxV gateway. system:125
Connected.
Pressure : 102.248kPa
Temperature : 32.25Celsius
Vacuum leak rate : -2.5e-05kPa/s
Pressure : 102.244kPa
Temperature : 32.25Celsius
Vacuum leak rate : -0.000525kPa/s
Pressure : 102.245kPa
Temperature : 32.25Celsius
Vacuum leak rate : -0.000575kPa/s
```

Press ctrl+c to exit. If you get an error, talk to an instructor to debug.

2.3 Attitude and heading reference system (AHRS)

The attitude and heading reference system (AHRS) is essentially an Inertial Measurement Unit (IMU) with a fusion algorithm that outputs the attitude of the AUV (i.e. the roll, pitch and heading angles of the vehicle). These are crucial measurements for underwater navigation.

In SeaBeaver III, we use an XSens MTi-3 Micro-Electro-Mechanical systems (MEMS) AHRS

<https://www.xsens.com/hubfs/Downloads/Leaflets/MTi-3.pdf>

The IMU is connected to the PocketBeagle using a universal asynchronous receiver/transmitter (UART) interface. You can unit test this sensor using the `test_xsens_mti3` program.

```
$ test_xsens_mti3
```

If the sensor is connected and is in working order, you will be asked to check if the latitude and longitude currently written in the sensor's non-volatile memory is correct (as you will learn later in this class, the MTi-3 AHRS corrects for the magnetic declination; therefore, it is important that it knows our approximate operational area):

```
debian@beaglebone:~$ test_xsens_mti3
Testing the Xsens MTi-3 IMU
First scanning for USB devices...
No USB device found. Scanning: /dev/tty04 at baud rate: 115200
Found a device with ID: 038898CE @ port: /dev/tty04, baudrate: 115200
Opening port...
Device: MTi-3-8A7G6, with ID: 038898CE opened.
Putting device into configuration mode...
Configuring the device...

Current Latitude: 52.2405 | Longitude: 6.83822 | Altitude: 1.29442e-312
Do you want to change the position for magnetic declination?
Enter Y to change Lat/Long. Any other input to keep them.
```

In this example, latitude and longitude does not represent Cambridge, and this will most likely be the case for the AHRS in your brand new training-kit. Press **Y** and hit return:

```
Current Latitude: 52.2405 | Longitude: 6.83822 | Altitude: 1.29442e-312
Do you want to change the position for magnetic declination?
Enter Y to change Lat/Long. Any other input to keep them.
y
Enter the new Latitude..
```

Then you will be asked to enter your latitude. The latitude of the MIT sailing pavilion is 42.3584, let us enter that.

```
Current Latitude: 52.2405 | Longitude: 6.83822 | Altitude: 1.29442e-312
Do you want to change the position for magnetic declination?
Enter Y to change Lat/Long. Any other input to keep them.
y
Enter the new Latitude..
42.3584
Enter the new Longitude..
```

Then you will be asked to enter your longitude. The longitude of the MIT sailing pavilion is -71.08745, let us enter that. Then the sensor will start outputting raw data – you will see an output similar to this:

```
New Latitude: 42.3584 | Longitude: -71.0875 | Altitude: 9.54898e-313
Putting device into measurement mode...

Reading data...
Acc X:1.06, Acc Y:-0.15, Acc Z:9.70 |Gyr X:0.01, Gyr Y:0.01, Gyr Z:-0.00
|Mag X:0.09, Mag Y:0.36, Mag Z:-0.56 |Roll:-0.91, Pitch:-6.29, Yaw:37.35, Hdg:232.65
```

Turn the training-kit around and observe how the raw acceleration, gyroscope and magnetometer readings; as well as fused roll, pitch and heading angles vary. Press **ctrl+c** to exit.

2.4 Global Positioning System (GPS)

While GPS does not work underwater, AUVs use GPS to initialize the position while on the water surface. SeaBeaver III vehicles use an Adafruit Ultimate GPS Breakout V3 as the GPS module (<https://cdn-learn.adafruit.com/downloads/pdf/adafruit-ultimate-gps.pdf>). This module is connected to the PocketBeagle via the UART interface.

You can unit test the GPS by using the linux minicom tool (<https://linux.die.net/man/1/minicom>). Minicom will display the NMEA strings that the GPS outputs.

```
$ minicom -D /dev/tty00 -b 9600 -w
```

If the GPS is connected to the PocketBeagle properly, and is it working order, you will see an output similar to this:

```
Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on May  6 2018, 10:36:56.
Port /dev/tty00, 22:10:29

Press CTRL-A Z for help on special keys

$GPGGA,222359.000,4221.5879,N,07106.2830,W,1,06,1.24,3.5,M,-33.7,M,,*51
$GPGSA,A,3,06,12,05,11,25,19,,,,,,,,,1.56,1.24,0.95*04
$GPRMC,222359.000,A,4221.5879,N,07106.2830,W,0.40,139.56,300324,,,A*74
$GPGGA,222400.000,4221.5880,N,07106.2831,W,1,06,1.24,3.5,M,-33.7,M,,*5D
$GPGSA,A,3,06,12,05,11,25,19,,,,,,,,,1.56,1.24,0.95*04
$GPRMC,222400.000,A,4221.5880,N,07106.2831,W,0.41,139.56,300324,,,A*79
$GPGGA,222401.000,4221.5881,N,07106.2831,W,1,06,1.24,3.5,M,-33.7,M,,*5D
$GPGSA,A,3,06,12,05,11,25,19,,,,,,,,,1.56,1.24,0.95*04
$GPRMC,222401.000,A,4221.5881,N,07106.2831,W,0.45,139.56,300324,,,A*7D
```

Try to figure out what these data strings mean (hint: these are NMEA sentences). To exit minicom, press ctrl+A, and then X.

If you are indoors, you may not have a GPS signal. In that case, you will get a response similar to this:

```
Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on May  6 2018, 10:36:56.
Port /dev/tty00

Press CTRL-A Z for help on special keys

$GPGGA,233552.089,,,,,0,0,,M,,M,,*49
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPRMC,233552.089,V,,,,,0.00,0.00,290325,,N*43
$GPGGA,233553.089,,,,,0,0,,M,,M,,*48
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPRMC,233553.089,V,,,,,0.00,0.00,290325,,N*42
$GPGGA,233554.089,,,,,0,0,,M,,M,,*4F
$GPGSA,A,1,,,,,,,,,,,,,*1E
```

2.5 Self checkoff assessment

Check:

- Have you confirmed that you can read depth data (i.e. if a depth sensor is plugged in)?
- Have you confirmed that you can read internal pressure data?
- Have you confirmed that you can read AHRS data?
- Have you confirmed that you can read GPS data?

3 Unit testing actuators

Now let's focus on unit testing the actuators.

3.1 Control surfaces

The servo motors that actuate the control surfaces of the SeaBeaver III AUVs are driven by Pulse-Width Modulation (PWM) commands provided by the PocketBeagle. The signal lines of the servo motors are connected to certain Programmable Real-time Unit (PRU) output pins of the PocketBeagle that are configured to generate the necessary PWM signals. Within the range, each PWM command corresponds to a certain position/angle of the servo motor.

You can test the servo motors that run the control surfaces by running `test_control_surfaces` program:

```
$ test_control_surfaces
```

If the servos are properly connected to the J6 pins of the frontseat board, you will observe the servos moving, one at a time. Press ctrl+c to exit.

3.2 Main propulsion thruster

WARNING: Always consider that the main thruster is live. Do NOT place fingers near the propeller while the thruster is connected!!!

SeaBeaver III AUVs use a Blue Robotics T200 thruster (<https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/>) as the main propulsion thruster. Similar to servo motors, the main thruster is driven by PWM signals. However, in this case, the PWM signals correspond to a thruster speed value. You can test the main thruster by running the `test_br_thruster` program:

```
$ test_br_thruster
```

Now you will be asked to provide a thruster percentage value. If you are testing the actual Blue Robotics T200 thruster, do not provide a value beyond 50%. 20% is a good value for testing the Blue Robotics T200 thrusters.

```
debian@beaglebone:~$ test_br_thruster
Enter thruster percentage between 100 and -100. ctrl-c to exit: 20
```

If everything is in proper order, you will now see the propeller spinning. **Note: Please do not run the Blue Robotics T200 thruster in air for more than 10 seconds as it will overheat.** You may run the mock-up main motor provided in your training-kit as long as you like.

3.3 Light Emitting Diode (LED) indicators

Although LED indicators are not crucial for SeaBeaver III AUVs, they can make vehicle operators' lives easier by indicating the current status of the vehicle, increasing the mission success rate. They are connected to the PocketBeagle using the general-purpose input/output (GPIO) pins.

You can test them by running the `test_leds` program:

```
$ test_leds
```

If everything is working as expected, the LEDs should now start blinking. Press `ctrl+c` to exit.

When you switch on the training-kit or vehicle, the indicator LEDs on the will blink in different patterns, indicating various states of the vehicle. It is useful to know what these patterns mean and what the vehicle is trying to tell you. If your vehicle or training-kit does not flash LEDs at all; please talk to an instructor.

1. Main pressure hull has no vacuum:
 - **Red:** solid with a quick 100 millisecond blink, once every 3 seconds
 - **Green:** off
 - **White:** off
2. Main pressure hull vacuum is leaking:

- **Red:** 1-second blink; i.e. 1 second on, 1 second off
 - **Green:** off
 - **White:** off
3. Main pressure hull vacuum is good. A mission is not launched. Vehicle is idle:
- **Red:** 200-millisecond blink; i.e. 200 millisecond on, 200 millisecond off
 - **Green:** 200-millisecond blink; i.e. 200 millisecond on, 200 millisecond off
 - **White:** 200-millisecond blink; i.e. 200 millisecond on, 200 millisecond off
4. Mission has been launched, time to put the vehicle in the water:
- **Red:** off
 - **Green:** off
 - **White:** 1-second blink; i.e. 1 second on, 1 second off
5. Mission has been launched, time to put the vehicle in the water. It is 10 seconds before the propeller starts spinning:
- **Red:** off
 - **Green:** off
 - **White:** starts to blink faster
6. Mission is active:
- **Red:** 1-second blink; i.e. 1 second on, 1 second off
 - **Green:** 1-second blink; i.e. 1 second on, 1 second off
 - **White:** solid
7. Mission is paused. Do not approach the AUV as it may resume the mission (or approach with caution):
- **Red:** solid
 - **Green:** 1-second blink; i.e. 1 second on, 1 second off
 - **White:** 1-second blink; i.e. 1 second on, 1 second off
8. Mission is complete. You can recover the vehicle:
- **Red:** 1-second blink; i.e. 1 second on, 1 second off
 - **Green:** solid
 - **White:** 1-second blink; i.e. 1 second on, 1 second off

3.4 Self checkoff assessment

Check:

- Have you confirmed that you can actuate the control surfaces
- Have you confirmed that you can actuate the main thruster
- Have you confirmed that you can light the LED indicators

4 Shutting down embedded computers

In Lab 2a, we learned how to safely shut down the PocketBeagle and Raspberry Pi. This process involves several steps. To simplify it for you, we have created a `turn_off_seabeaver` command. When run on the Raspberry Pi, this command will automatically shut down the PocketBeagle first, followed by the Raspberry Pi. During the process, you will be prompted to enter the passwords for both the PocketBeagle and the Raspberry Pi.

```
$ turn_off_seabeaver

Remotely triggering shutdown sequence on the SeaBeaver PocketBeagle..
Debian GNU/Linux 10

BeagleBoard.org Debian Buster IoT Image 2020-04-06

Support: http://elinux.org/Beagleboard:BeagleBoneBlack\_Debian

[sudo] password for debian:
Connection to 192.168.7.2 closed.
Initiating the shutdown sequence on the SeaBeaver RasPi..
[sudo] password for seabeaver-raspi:
Connection to 10.42.0.5 closed by remote host.
Connection to 10.42.0.5 closed.
```

5 Assessment

Lab assessments are structured to ensure that you are on track to complete your AUV by the end of the semester for in-water trials. You are expected to demonstrate to one of the instructors that you can complete the following tasks. More importantly, please do not hesitate to communicate with the instructors if you encounter any issues.

1. Pick one of your favorite sensors and demonstrate to an instructor that you can read data from it by running a unit testing program.
2. Pick one of your favorite actuators and demonstrate to an instructor that you actuate it by running a unit testing program.