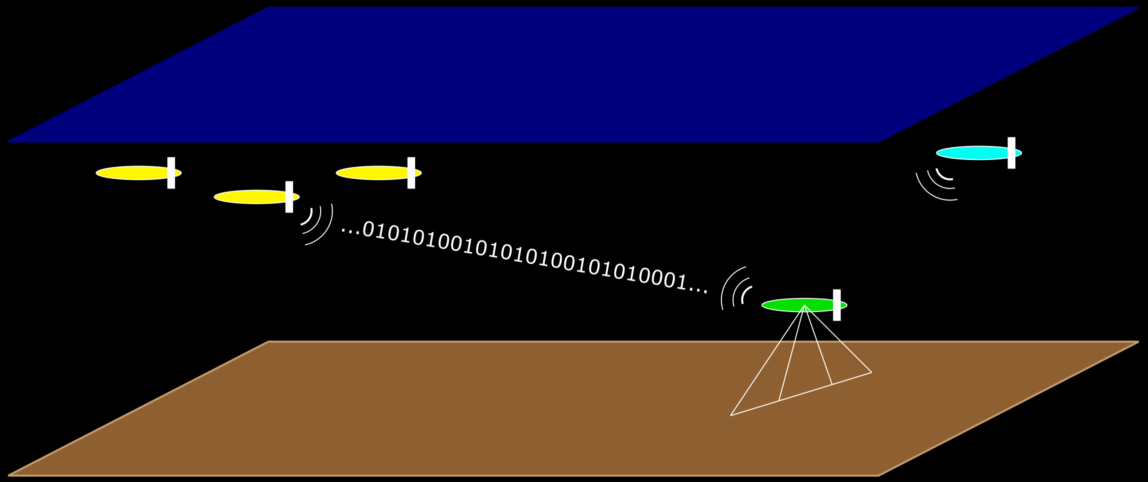
Goby3 Course

Day 4: Sensing



Course Sponsored by:

Mission Systems

BlueOcean SEISMIC SERVICES

Raytheon Technologies

Toby Schneider

GobySoft, LLC

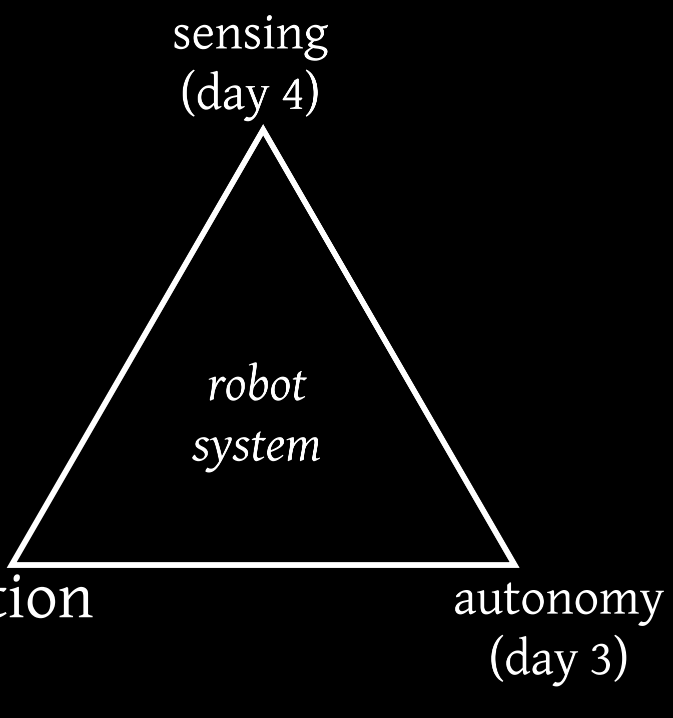
Mashpee, MA, USA

gobysoft.org aquatic software

Robots (revisited!)

In many systems, this triad represents tradeoffs:

- More communications = less need for autonomy (UAVs)
- Better autonomy = better data from cheap sensors (Adaptive sampling)
- Better sensors = less need for outside data (Manned subs)



Sensors in Marine Robotics

Wide range of oceanographic sensors:

- Physical: CTD, Water velocity, ADCP, Magnetometers
- Chemical: pH, CO2, nutrients
- Biological: DNA, cytometers
- Imagery: seafloor cameras, water column imaging (Mesobot)
- Remote sensing (sonars): seafloor mapping, hull inspection, etc.

Increasing miniaturization and reduced power usage increases realistic sensor choices for AUVs.

Sensors from a software view

Some common themes:

- Many are serial based, with a wide range of ad-hoc protocols.
- Little to no standardization
- Quirky state machines
- Often expensive, so having extras just for software dev is challenging.

Goby and Sensors

A few things that Goby offers to make sensor integration easier:

- Suite of I/O threads that can be extended for new protocols:
  - Serial, UDP, TCP, CANBUS, PTY
  - (Regex) line-based ASCII delimiters, MAVLink, easy to add new wire protocols
- goby_gps application for GPSD
- Straightfoward integration with boost::statechart for lifecycle management of sensor states.
- Sensor simulation

Hands-on

(Switch over to VSCode: I/O threads, goby_gps)

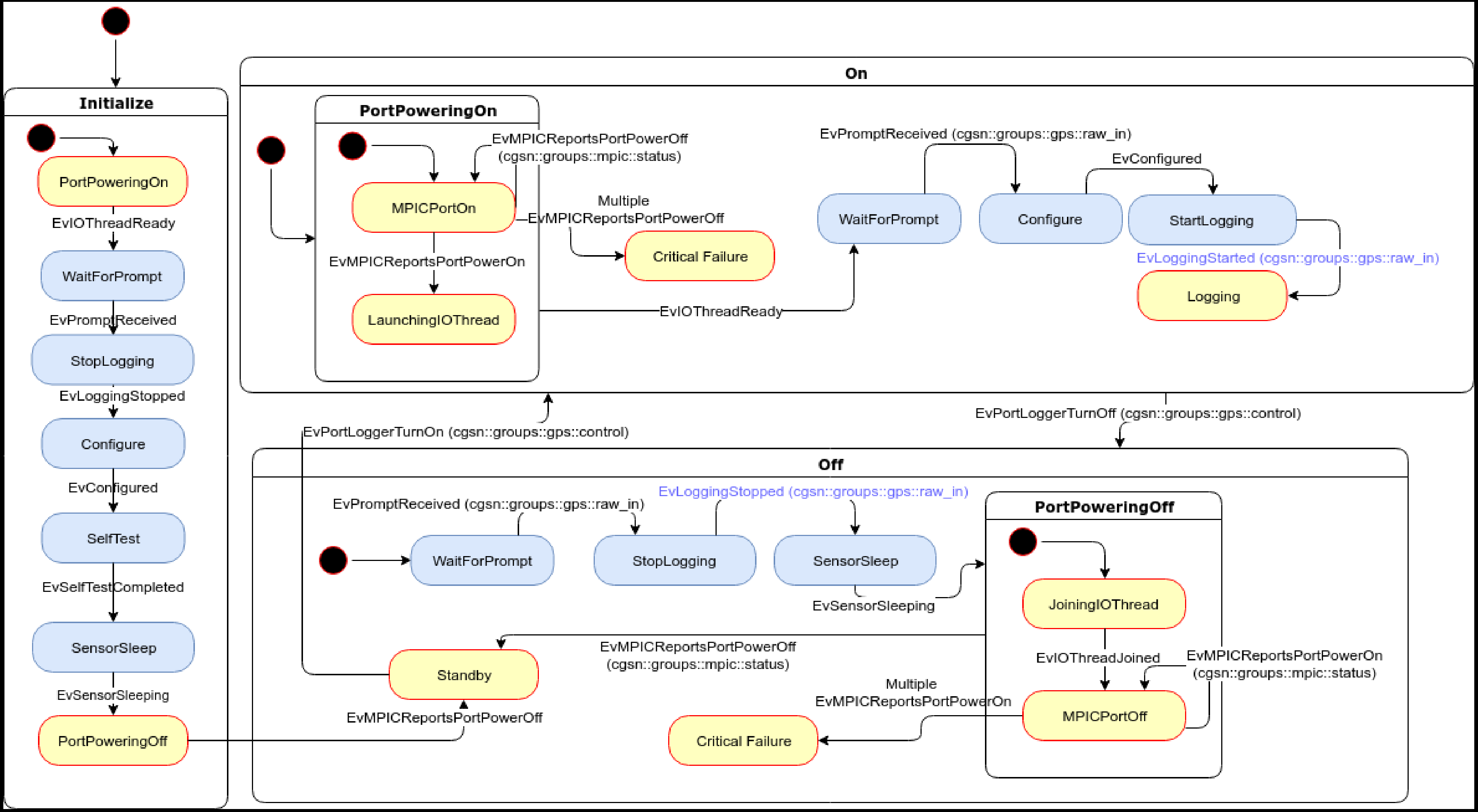
Sensor State Machines

For longer term deployments (e.g. moorings), sensor life-cycle management becomes especially important:

- Sensor powered on / off
- Sensor logging / not logging
- Sensor sleeping (low power) / full power
- (Raw/parsed) data are valid / invalid

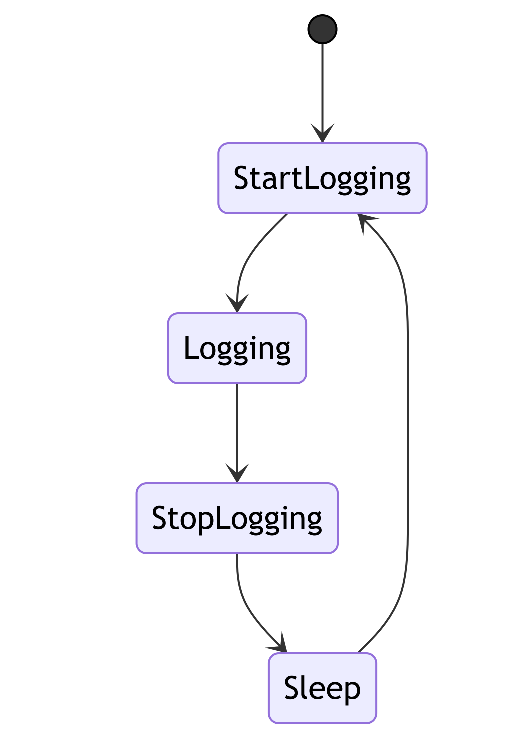
Given the often poor implementation of the sensor interfaces, correctly and accurately tracking these states and their transitions becomes a major role of the payload software.

Real example



Simplified Example

We'll look at a simplified version of this:



We'll start filling this out now, you'll finish during the homework.

Hands-on

(Switch over to VSCode: Sensor state machines)

Future directions for Goby

Next steps (not necessarily in order):

- Official release 3.0.0 (soon)
- Additional intervehicle data flow policies for multiple links:
  - Flood
  - Highest priority
  - Others?
- Add to driver "catalog" for modems & vehicles.
- goby_ros_gateway
- Goby-IvP
- New InterProcessPortal implementations (boost::interprocess?)
- New marshalling schemes (Cap'n Proto, msgpack)

Resources

Summary of Goby3 resources (besides this course):

- Developer manual: https://goby.software/3.0
- Source code: https://github.com/GobySoft/goby3
- Debian/Ubuntu packages: packages.gobysoft.org (see instructions in Developer manual).
  - Ubuntu LTS (now: 16.04, 18.04, 20.04)
  - Debian stable (buster) / oldstable (stretch)
- Wiki: https://github.com/GobySoft/goby3/wiki
- Examples: https://github.com/GobySoft/goby3-examples

Ways you can contribute

Many ways to contribute:

- Modem drivers (use Pull Requests on Github)
  - Write a new one
  - Adopt an existing one and keep it up to date
- Frontseat interface drivers: New or adopt existing
- Write a new example for goby3-examples when you find something that you feel is missing.
- Contribute new & updated documentation pages (Markdown).
- Suggest a contribution of your useful new piece of code.
- Sponsor the project financially:
  https://github.com/sponsors/GobySoft

Thank you!

Thanks for attending!

gobysoft.org
aquatic software

Course Sponsored by:

Mission Systems

BlueOcean
SEISMIC SERVICES

Raytheon Technologies