

# ERSP-U Agenda

## Overview

ERSP-U provides a hands-on introduction to programming with ERSP 3.0.

Five main areas are covered:

- The Task Execution Layer (TEL)
- The Behavior Execution Layer (BEL)
- The Hardware Abstraction Layer (HAL)
- Object Recognition
- Navigation

Attendees will modify and test programs relating to each of these topics, plus a final exercise, which combines several topics.

ERSP 3.0 includes an architecture with Task, Behavior, and Hardware layers; a large library of pre-written tasks, behaviors, and hardware drivers; an object recognition library, navigation, and mapping functions. Typically developers need to use tasks, behaviors, and drivers, that are included with ERSP, as well as develop custom ones. The lessons and exercises in ERSP-U cover using existing modules and writing custom ones.

ERSP-U consists of two days of intensive training.

## Day 1

Day 1 provides an in-depth introduction to all 3 levels of ERSP and Object Recognition, including examples which are tested on robots, and code walk-throughs. The first day will not include any code \*writing\* though, and is designed as a less-technical day.

*Day 1 is appropriate for technical managers and engineers. Prerequisites for the first day are basic computer skills, such as running scripts and compiling programs.*

**Day 1 will cover:**

- **Programming tools and the logging system**
  - "Hello world" program
- **Task Execution Layer**
  - Compile and run a program which drives the robot, speaks, and

- takes a picture
- **Behavior Execution Layer**
  - Run a behavior network, which uses recognizes an object
- **Hardware Abstraction Layer**
  - Run a program, which moves the robot and measures the motion
- **Object recognition**
  - Create an object database
  - Extract features
  - Recognize an object from the database

## Day 2

Day 2 exercises allow participants to write and test custom modules with help from the instructors.

*Day 2 is intended primarily for engineers who will be programming with ERSP. Participants for the second day should be experienced C++ programmers who have some familiarity with robots. Additionally, it will be helpful if participants have already tried running some of the ERSP sample programs to familiarize themselves with the process.*

### **Day 2 will cover:**

- **TEL:**
  - Write a custom navigation task using VSLAM
- **BEL:**
  - Write a custom behavior using Object Recognition
- **HAL:**
  - Write a custom odometry resource, and use it to measure robot motion
- A final exercise constructing a Task, which moves the robot and uses object recognition.

After both days, participants should have an understanding of the important software interfaces in ERSP and be ready to develop custom applications.

Robot and computer hardware will be provided.