MATLAB® & Simulink® with Raspberry Pi®
A hands-on workshop on hardware support for project based learning

Seminar Overview

The workshop describes the built-in support for prototyping, testing, and running Simulink models on Raspberry Pi®. This platform aims to address the growing need for hands-on and project-based learning via a low-cost, easy to use hardware and software environment that builds on the widely used MATLAB & Simulink.

The Simulink built-in support for hardware enables students to access the hardware capabilities of the popular credit-card sized, ARM11 based Raspberry Pi® computer, from within Simulink environment, and deploy for embedded implementation. Faculty who attend will have a chance to work through lab modules with examples of video and image processing algorithms, from very simple video in/out handling to a more sophisticated processing such as object recognition and edge detection. They will have an opportunity to gain practical hands-on experience in building such high-level examples themselves, and by extension understand the potential for use in the classroom with undergraduate students.

Agenda

Time	Title
14.00 - 14.15	Introduction to Simulink and HW Support for Project Based Learning • Overview of the Simulink built-in support for target hardware • Advances in low-cost embedded hardware • Building models in Simulink • Simulink for Model-Based Design of embedded or real-time applications • Rapid Prototyping and Automatic Code Generation tools
14.15 - 15.15	Example lab module 1 • Example 1: Explore a pre-built Simulink example (barcode scanning), download and execute on Raspberry Pi® • Example 2: Build and configure a simple video-loop example from scratch following the user's guide • Example 3: Use the video input acquired by a webcam to develop an object detection example with standard library functions/blocks from the user's guide Key Concepts: System configuration and data types
15.15 - 15.45	Break
15.45 - 16.45	Example lab module 2 • Building a more advanced video and image processing algorithm starting from its mathematical formulation • Example 1: Edge Detection • Example 2: reducing algorithm complexity • Example 3: algorithm optimization and verification Key Concepts: algorithm optimization, system verification, creating test benches
16.45 - 17.00	Wrap-up