New course proposal: Introduction to Embedded Systems

Pedro Nariyoshi

Audience

This course is aimed at students who want to deepen their knowledge of the use of microcontrollers and embedded systems. The material will cover communication protocols and microcontroller programming with complementary laboratory sessions using sensors and actuators.

Pre-requisites

Students are expected to know C language [1], Digital logic and Digital Systems (or an equivalent microprocessor architecture course)

Course Objectives

Introduce students to industry standard communication protocols and how to use and choose between them. Introduce students to different peripherals and how to integrate them in their projects.

Learning Outcomes

At the end of the semester, the student will be able to:

1. Describe common built-in peripherals included in microcontrollers (GPIO, A/D, timers & PWM)
2. Write programs that utilize these built-in peripherals
3. Differentiate communication protocols (UART, I^2C & SPI) and their primary characteristics
4. Use a pre-built software library to interface microcontrollers with sensors and actuators
5. Select the correct communication protocol for different applications
6. Independently read a sensor/actuator datasheet and use it to interface the microcontroller to it.
7. Develop libraries for a given sensor/actuator
8. Build a system that uses multiple sensors and/or actuators to solve a problem

[1] a short review will be provided at the beginning

9. Be able to observe a non-working circuit and provide a reasonable procedure to diagnose and solve the problem

Modules

Introduction

Introduction to Embedded Systems :

- Introduction & Syllabus (0.5 lecture)
- Motivation (0.25 lecture)
- Applications (0.25 lecture)
- Practical limitations (0.5 lecture)
- C programming refresher (0.25 lecture)
- Circuits refresher (0.25 lecture)

Lab 0 (take-home C exercise)

Built-in Peripherals :

- GPIO
  - Definition & important parameters (0.5 lecture)
  - Applications (0.5 lecture)
  - Sample circuits & Programming examples/exercise (1 lecture)
- A/D
  - Definition & important parameters (1 lecture)
  - Sample circuits & Programming examples/exercise (1 lecture)
- Timers & PWM
  - Definition & important parameters (1 lecture)
  - Sample circuits & Programming examples/exercise (1 lecture)

Labs 1 (GPIO), 2 (A/D) & 3 (Timers & PWM)

External Peripherals/Communication Protocols :

- UART
  - Definition, History & Applications (0.5 lecture)
  - Network Topology & Electrical connections (0.5 lecture)
  - Sample circuits & Programming example/exercise (1 lecture)
- I^2C
  - Definition, History & Applications (0.5 lecture)

  - Network Topology & Electrical connections (0.5 lecture)
  - Sample circuits & Programming example/exercise (1 lecture)
- SPI
  - Definition, History & Applications (0.5 lecture)
  - Network Topology & Electrical connections (0.5 lecture)
  - Sample circuits & Programming example/exercise (1 lecture)

Labs 4 (UART) , 5 (I^2C) & 6 (SPI)

Software Libraries :

- Motivation (0.5 lecture)
- Using pre-built libraries (0.5 lecture)
- Datasheet interpretation (2 lectures)
- Writing a new library (1 lecture)

Labs 7 (pre-built) & 8 (self-written)

Advanced topics (if possible):

- Maximum ratings (1 lecture)
- Level Shifters (1 lecture)
- Power Management (1 lecture)
- Bit-banging (1 lecture)
- Data processing (2 lectures)

Final Project :

- Identify a solvable problem
- Select the appropriate sensors/actuators
- Create software to interface with sensors/actuators
- Build circuit integrating microcontroller and sensors/actuators
- Write report
- Demonstrate system (2 lectures, teams present)

Labs 9 - 13 (Final Project)