

Lab 3: Building Linux Kernel and Controlling an I²C Device

Due Date: See the course schedule web page.

Objectives

- Know how to build Yocto Linux kernel and file system images
- Be able to control an I²C device using Linux

Description

You should now have a working sensor device interfaced with the Galileo development board via GPIO ports. We would like to give timestamps for the light intensity readings obtained from the PIC sensor device so that we can understand the trend of light intensity changes during a day. Therefore, a real-time clock (RTC) circuitry is to be added to your Galileo board for setting and maintaining the correct system time, even when the Galileo is powered off.

This lab consists of two parts: (1) building Yocto Linux and file system images and (2) programming I²C devices from Linux. Instead of using the Linux image we provide to you in Lab 2, you are required to build your own Linux kernel and file system images from scratch, and use them to boot your Galileo with an updated MicroSD card. Once you have your own Linux running on Galileo, you can then program the RTC1307 chip using Linux I²C libraries and APIs.

Building Yocto Linux

The detailed instructions of building Linux images are given in the lab3 directory on Micro2 repository on github

<https://github.com/yanluo-uml/micro2>

Please closely follow the instructions. You need to have a Ubuntu 12.04 machine, or a virtual machine (e.g. VirtualBox or VMWare Player) running Ubuntu 12.04.

Connecting RTC1307 Circuit to Galileo

You will need to design a real-time clock circuit that consists of RTC1307 chip and peripheral components (e.g. crystal, battery and resistors). Refer to DS1307 Datasheet for the schematic. Your real-time clock should be connected to A4 (SDA) and A5 (SCL) of Galileo's expansion I/O ports. "gpio29" must be set to zero to setup the multiplexors properly. (Refer to the Galileo Linux GPIO pin diagram). Wire the pull-up resistors for the I²C bus.

Programming I²C Devices from Linux

Linux has mature I²C drivers and libraries for programming I²C devices. Please refer to the official documentation on I²C development

<https://www.kernel.org/doc/Documentation/i2c/dev-interface>

There are also other related tutorials, for example

<http://blog.chrysocome.net/2013/03/programming-i2c.html>

Deliverables

A zipped file containing

1. Schematic of the design (in both native and pdf formats)
2. Source code (for Galileo Linux)
3. Reports

References

[1] PIC16F684 datasheet. Available at

<http://ww1.microchip.com/downloads/en/devicedoc/41202f-print.pdf>

[2] Linux I²C documentation.

<https://www.kernel.org/doc/Documentation/i2c/dev-interface>

[3] DS1307 I²C Real-time Clock Datasheet,

<http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>