Lab 3: Building Linux Kernel and Controlling an I²C Device

Due Date: See the course schedule web page.

Objectives

- Understand I²C bus protocol
- Be able to control an I²C device using Linux on a Galileo board
- Be able to capture, store and process camera images on Linux

Description

You should now have a working sensor device interfaced with the Galileo development board via GPIO ports. We would like to add a couple of new devices to the system so that your embedded system has richer functions. The devices are as follows:

(1) A temperature sensor (TMP102). This is an I²C device that measures ambient temperature to a resolution of 0.0625°C. The IC is provided on a breakout board for easy connection. Its details can be found in [3].
(2) A gesture sensor (APDS9960). This is an I²C device that supports gesture detection, proximity detection and many advanced features. The IC is also provided on a breakout board for easy connection. Its details can be found in [4]. {Required only for students in EECE.5520}
(3) A USB webcam to capture images and videos

This lab consists of three objectives:
(1) programming I²C devices from Linux. You will be provided a Linux image and file system, and use them to boot your Galileo with an updated MicroSD card. Once you have Linux running on Galileo, you can then program the I²C devices using Linux I²C libraries and APIs.
Note: The gesture sensor is required only for students in EECE.5520.
(2) programming on Linux to access webcam and capture images. Store the images on the SD card and prepare for further processing (in lab 4).
(3) use temperature sensor or gesture sensor to trigger the capture of images from webcam. You need to define a threshold and check if the sensor data exceed the threshold. If so, capture images and save them to the file system.

Connecting I²C devices to Galileo

Refer to datasheets for the schematic. Your I²C devices should be connected to A4 (SDA) and A5 (SCL) of Galileo's expansion I/O ports. For Galileo Gen 1 "gpio29" must be set to zero to setup the multiplexors properly. For Galileo Gen 2 "gpio60" is the MUX to clear. (Refer to the Galileo Gen1 or Gen2 Linux GPIO pin diagram).

You do not have to wire the pull-up resistors for the I²C bus since the breakout boards already have them.

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`

Programming webcam and connecting Wifi on Linux

Instructions are given in a text file as a part of the github repo:
https://github.com/yanluo-uml/micro2/
You need to use git command to make a copy of the repository:
`git clone https://github.com/yanluo-uml/micro2/`

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] The Linux images for Galileo,
`https://software.intel.com/en-us/iot/hardware/galileo/downloads`
[2] Linux I²C documentation.
`https://www.kernel.org/doc/Documentation/i2c/dev-interface`
[3] Temperature sensor, `https://www.sparkfun.com/products/11931`
[4] Gesture sensor, `https://www.sparkfun.com/products/12787`