



Developing Linux kernel driver

Rafal Kapela

*During this labaratory you will develop and compile Linux IIO
(Industrial I/O) kernel dirver*



Note, that in order to follow instructions in this lab session you need to have prepared the following configuration:

- PC computer with Linux Kubuntu installed on it in the lab.
- Galileo2 development board.
- Arduino mezanine board.
- I2C display extension.
- Photoresistor extension.
- Potentiometer extension.
- An SD card reader for PC computer.
- Power supply.
- USB UART cable.
- An SD card prepared on the last laboratory session.

The setup is shown in the following picture.



1 Preparing galileo-linux-stable kernel

1. In the directory where you have Linux kernel that you compiled during the last lab session dive into drivers/iio directory.
2. In the Kconfig file you need to add the config for the caracter IIO device:

```
menuconfig IIO
  tristate "Industrial I/O support"
  depends on GENERIC_HARDIRQS
  help
    The industrial I/O subsystem provides a unified framework for
    drivers for many different types of embedded sensors using a
    number of different physical interfaces (i2c, spi, etc).
    number of different physical interfaces (i2c, spi, etc).
```

3. Now you need to change the `Makefile` to take the `Kconfig` changes into account. Based on the other lines figure out the correct new line that will be responsible for compiling the IIO character device driver.
4. Go to the main kernel directory and check if the new driver is selectable by the menu.
5. Select it as a module and check if `makeARCH=i386-j4` compiles all the modules without errors.
6. If it compiles copy the new modules to the Galileo2 board and try to run the potentiometer readout. Does it work?
7. Fix the `industrialio-event.c` module. Function that is broken: `iio_event_chrdev_read`. In order to do this use the following kernel API:
 - `kfifo_to_user`
 - `spin_lock`
 - `spin_unlock`
8. Change the `ad7298` driver so that it displays the reference voltage.