



## Developing Linux kernel driver

*Rafal Kapela*

*During this labarootory you will develop and compile Linux I2C  
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/i2c directory.
2. In the Kconfig file you need to add the config for the character I2C device:

```
config I2C_CHARDEV
    tristate "I2C device interface"
    help
        Say Y here to use i2c-* device files, usually found in the /dev
        directory on your system. They make it possible to have user-space
        programs use the I2C bus. Information on how to do this is
        contained in the file <file:Documentation/i2c/dev-interface>.

        This support is also available as a module. If so, the module
        will be called i2c-dev.
```

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 I2C 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 LCD application. Does it work?
7. Fix the `i2c-dev.c` module. Functions that are broken are: `i2cdev_write` and `i2cdev_read`. In order to do this use the following kernel API:
  - `memdup_user`
  - `i2c_master_send`
  - `i2c_master_recv`