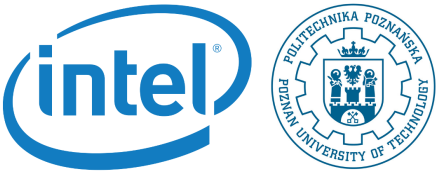

SD card preparation with Linux operating system

Rafal Kapela

During this labaroatory you will install the Linux operating system for the Galileo2 development board.

Note, that in order to validate the installation 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.

The setup is shown in the following picture.

1. Prepare the sd card image as follows:
   (a) Create the directory where you'll store all your files.
   (b) Download the raw SD card image to your machine with the scp command:

```
scp student@150.254.21.85:<path-to-destination> <your-directory>
```

where destination path is:
`/home/student/Downloads/Galileo2/sd_image/iot-devkit-1.5-i586-galileo`
or download it from:
`https://software.intel.com/sites/landingpage/iotdk/board-boot-image.html`

(c) Put SD card in the reader. DON'T MOUNT IT!

(d) use dmesg command to read the volume ID (sdX)

(e) use dd command to copy the image to the Sd card:

```
dd if=<path-to-the-SD-file> of=/dev/sd<X> bs=1
```

(f) After the process is done sync the IO with sync command

(g) Now you can remove the SD card and check if it's working.

2. Connect the mazanine board, power cable, extension baords, USB-UART cable and insert prepared SD card. You don't need to power on the board, it'll start booting once the power cable is attached.

3. Connect to with the board via the UART cable:

```
picocom -b 115200 /dev/ttyUSB0
```

You'll see the Linux transcript followed by the Linux login prompt:

```
[   21.228329] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[   21.233877] Bluetooth: BNEP filters: protocol multicast
[   21.258590] Bluetooth: BNEP socket layer initialized
[  OK  ] Started Redis Server.
[  OK  ] Started Bluetooth service.
[  OK  ] Started Update UTMP about System Runlevel Changes.
         Starting Hostname Service...
         Starting WPA supplicant...
[  OK  ] Started Hostname Service.
[  OK  ] Started WPA supplicant.

galileo login:
root@galileo:~#
```

use `root` user to login, there's no password. You'll see the Linux shell prompt if the login is successful.

4. In order to check the correctness of the installation do the following:

(a) check the voltage on the photoresistor:

```
cat /sys/bus/iio/devices/iio\:device0/in_voltage<X>_raw
```

Occlude the sensor and check it again, the readout should change.

(b) check the voltage on the potentiometer:

`cat /sys/bus/iio/devices/iio\:device0/in_voltage<X>_raw`

Change the position of the knob and check the voltage again, the readout should change.

(c) write simple I2C application that controls the I2C display:

```
#include "jhd1313m1.h"
#include <climits>
#include <iostream>
#include <sstream>
#include <unistd.h>

int
main(int argc, char **argv)
{
  upm::Jhd1313m1 *lcd = new upm::Jhd1313m1(0, 0x3E, 0
      x62);
  float fade = 0.6; // fade value [0.0 .. 1.0]
  uint8_t r, g, b; // resulting LCD backlight color
      components [0 .. 255]
  std::stringstream row_1, row_2; // LCD rows
  row_1 << "Hello. I'm";
  row_2 << "<my name>";
  lcd->setCursor(0,0);
  lcd->write(row_1.str());
  lcd->setCursor(1,0);
  lcd->write(row_2.str());

  // fade the color components separately
  r = (int)(255 * fade);
  g = (int)(64 * fade);
  b = (int)(255 * (1 - fade));
  lcd->setColor(r, g, b);

  usleep(500000);
  delete lcd;

  return 0;
}
```

Listing 1: C++ code for I2C LCD initial test

Compile the application natively with the following command:

`g++ -I/usr/include/upm/ lcd_test.cpp -lupm-i2clcd`

5. Change the I2C display application so that the background color changes from blue to red within a fixed ammount of time.
6. Change the I2C display application so that the background color changes from blue to red based on the position of the potentiometer knob.
7. Change the I2C display application so that the background color changes from blue to red based on the position of the ammount of captures light by the photoresistor.