# Project Laboratory Documentation

Kristof Kalocsai

March 15, 2015

## 1 Firstofall

firstofall

install TI energia IDE: http://energia.nu/download/

to access serial port and programmer as regular user, had to make udev rules, add user to dialout group:

```
sudo touch  /etc/udev/rules.d/99-ti-launchpad.rules
echo "# EK-TM4C123GXL, EK-TM4C1294XL and EK-LM4F120XL LaunchPad
ATTRS{idVendor}=="1cbe", ATTRS{idProduct}=="00fd", MODE="0660",GROUP="dialout"
" > /etc/udev/rules.d/99-ti-launchpad.rules
```

unzip energia

ooops, have to install Java first

sudo apt-get install default-jdk

profit! haha had to install 32bit compatible libraries :

```
sudo apt-get install libc6:i386 libx11-6:i386 libstdc++6:i386 libexpat1:i386
```

everythings working fine, except when the laptop(from which is share the network to the board) is connected to an IPv6 network, then it fails, needs more debug.

runnning into that energia is not suited for this kinda stuff, could not find example of freeRTOS, lwip working on it, trying to install a more sophisticated environment, using GCC ARM Embedded toolchain, lm4flash, eclipse:

install gcc-arm-none-eabi

```
sudo apt-get install gcc-arm-none-eabi
dpkg --get-selections | grep -v deinstall | grep gcc
gcc install
gcc-4.4 install
gcc-4.4-base:amd64 install
gcc-4.8 install
gcc-4.8-base:amd64 install
gcc-4.8-base:i386 install
gcc-4.9-base:amd64 install
```

```
gcc-4.9-base:i386 install
gcc-arm-none-eabi install
gcc-avr install
libgcc-4.8-dev:amd64 install
libgcc1:amd64 install
libgcc1:i386 install
```

then needed the TI libs and headers, fortunateli khazy supplied me with these

and for the flasher: lm4flash

```
git clone https://github.com/utzig/lm4tools.git
cd lm4tools/lm4flash
make
Package libusb-1.0 was not found in the pkg-config search path.
Perhaps you should add the directory containing `libusb-1.0.pc'
to the PKG_CONFIG_PATH environment variable
No package 'libusb-1.0' found
Package libusb-1.0 was not found in the pkg-config search path.
Perhaps you should add the directory containing `libusb-1.0.pc'
to the PKG_CONFIG_PATH environment variable
No package 'libusb-1.0' found
cc -Wall  -O2 lm4flash.c  -o lm4flash
lm4flash.c:30:20: fatal error: libusb.h: No such file or directory
 #include <libusb.h>
                    ^
compilation terminated.
make: *** [lm4flash] Error 1
```

oops, had to install libusb-1.0.0-dev also

```
sudo apt-get install libusb-1.0-0-dev
```

now it makes succesfully

```
make
cc -Wall -I/usr/include/libusb-1.0   -O2 lm4flash.c -lusb-1.0   -o lm4flash
```

but how to install it? move it to the local/bin, to stay on the PATH

```
sudo mv lm4flash /usr/local/bin
```

now install eclipse `http://kernelhacks.blogspot.ro/2012/11/the-complete-tutorial-for-stellari` `25.html`

```
sudo apt-get install eclipse-cdt
```

settings was a mess, docs later

trying to compile lwip by itself

hat to install check framework for lwIP unittest

```
sudo apt-get install check
```

had to make link for stubs.h

```
cd /usr/include
sudo mkdir gnu
sudo ln -s /usr/include/x86_64-linux-gnu/gnu/stubs.h /usr/include/gnu/stubs.h
```

and friggin 32 bit ones too

```
sudo apt-get install libc6-dev-i386
```

installs tons of stupid packages, maybe i do not need that test equipment

DO I NEED THAT? yup

have to make cc.h for porting

using TI's for now ( from the tivaware libs)

MASSIVE FAIL cannot work by itself do need an OS(or OS functions implemented barebones) an aplication, drivers, etc

now trying RTOS on its own, then trying to put lwIP on top of that

RTOS says TI cortex-m4f IS supported, so is gcc, giving it a try

on a slightly unrelated note, trying to setup openocd found a .cfg file in chibiOS pasted it in ONLAB folder root copied it to

```
/usr/share/openocd/scripts/board/ek-tm4c1294xl.cfg
```

had to install gdb-arm-none-eabi

```
sudo apt-get install gdb-arm-none-eabi
```

friggin fail! now trying to build openocd myself, and apply the patch mentioned in `http://kernelhacks.blogspot.hu/2012/11/the-complete-tutorial-for-stellaris_23.html`

had to install libtool libusb-dev automake

made it from source now it works doesnt even had to apply patches the problem was the maintainer mode, and enabling ICDI

now almost everything works smoothly

however, i have to add a project to debug in the gdb configurations, how to manage that? means for every new project have to overwrite the project name in the gdb label

wow

making progress!

if setting linker to -nostartfiles, can actually call ROM API functions, `http://www.ti.com/lit/ug/spmu363a/spmu363a.pdf`, even compile TivaWare examples like timer!

this way can make timers, etc which are needed for the FreeRTOS also can use UART now

```
cat /dev/ttyACM0
```

tomorrow i'll try the HW independent FreeRTOS example, thisway for starters, i only need to write a few HW specific fns/defines