# Energia for MSP430 on Beaglebone Black

A while ago I had the great idea of have a fully integrated full featured development system for the MSP430 devices. With Energia, an port from Arduino, a great tool is already available which should be easy to port to other system. Finally I got it setup and running. For those interested I am sharing the required steps below.

Basic system used was the Beaglebone Black with the Angstrom Linux system (I also tried it with Ubuntu but had several issues with the USB port connection to the MSP430 FET, which after some time got tired to further debug). The setup on the Angstrom requires much more steps but the USB driver is working fine.

First make Angstrom ready to run from an SD card as we need some more space to get all the stuff loaded and compiled. I have used a 16GB card (Speed 10 – to have more fun)

1. Go to the www.beaglebone.org page and follow the step to update the image:
   http://beagleboard.org/Getting%20Started#update
   Select the image without the eMMC flasher so that it runs from the SD card
2. As we have bigger card as on the image, we need to repartition it:
   The easiest way, and that what I describe, is with a running Linux on a PC, this can be also a virtual machine. There are plenty of side which describes how to setup on.
   If you have this start gparted (if not available run sudo apt-get -y install gparted):
   a. Run ‘mount’ to identify which connection your SD card has
   b. Run ‘sudo gparted /dev/sdx’ (x should identify your SD card)
   c. Increase the partition of the main block to the max.
   d. Save, apply and exit (apply takes a while)
   e. Now ‘sync’ and remove your SD card and put it into the Beaglebone.
   f. If you now power up the Beaglebone it should boot from the fresh installed SD card.

Next we need some basic installations to make the live easier.
**Note**: I do not take care about a save operating system. If required just google.
I did most of the work with the Beaglebone connected to my PC via the USB cable and therefore wanted to have some more access on this.

1. So I installed putty in my PC and made a connection to 192.168.7.2
   Login as: root
2. I had some problems to access the internet through my PC, with the following three lines everything worked fine:

```
/sbin/ifconfig usb0 192.168.7.2 netmask 255.255.255.252
route add default gw 192.168.7.1
echo "nameserver 8.8.8.8" >> /etc/resolv.conf
```

3. If you now do a ping www.google.com you should get a response
4. Set the system time and date

```
date -set "mon day hh::mm::ss UTC yyyy"
```

5. If you would like you can check for updates of you system (required to run twice to get all the packages):

```
opkg update
opkg upgrade
opkg update
opkg upgrade
opkg install sudo
```

   But this can take several hours and may be repeated some times to get all the packages

6. First I installed a VNC Server -> VNC should start automatically

```
opkg install angstrom-x11vnc-xinit
```

   Reboot the BBB

   From a vnc-client (vnc viewer), access 192:168.7.2:0

7. Then I installed an FTP server, there are some prepared files (vsftpd, vsftpd.conf) required (this can be found in the package …..)

```
## -> this is not available as prebuild so we need to do it manually
## Note: the download from https does not work so download it manually
## and copy it to the BBB
## Note: Windows user can access the boot segment (FAT) and put some
## files there for an easier data exchange. Maybe the BBB needs to be
## restarted to see the files
## access via: /media/BEAGLE_BONE
## wget https://security.appspot.com/downloads/vsftpd-3.0.2.tar.tar
tar -xzvf vsftpd-3.0.2.tar.tar
cd vsftpd-3.0.2
make
useradd nobody
mkdir /usr/share/empty/
mkdir /var/ftp/
useradd -d /var/ftp ftp
chown root.root /var/ftp
chmod og-w /var/ftp
make install
cd ..
rm -r vsftpd-3.0.2
rm vsftpd-3.0.2.tar.tar
cp setup_files/ftp/vsftpd.conf /etc
cd setup_files/ftp/init.d/vsftpd /etc/init.d/vsftpd
chmod +x /etc/init.d/vsftpd
/etc/init.d/vsftpd restart
update-rc.d vsftpd defaults
```

8. So the FTP server should now start automatically on each power up and we can exchange files very easy between the PC and the Beaglebone.

9. Now get the Beaglebone ready for all the software packages to build and install:

```
opkg install git man make gawk bison gcc perl-pod

##opkg install texinfo
## -> this is not available as prebuild so we need to do it manually
wget http://ftp.gnu.org/gnu/texinfo/texinfo-5.2.tar.gz
tar -xzvf texinfo-5.2.tar.gz
cd texinfo-5.2
./configure
make
make install
make clean
cd ..
```

```
rm -r texinfo-5.2
rm texinfo-5.2.tar.gz

wget http://ftp.gnu.org/gnu/texinfo/texinfo-4.13a.tar.gz
tar -xzvf texinfo-4.13a.tar.gz
cd texinfo-4.13
./configure
make
make install
make clean
cd ..
rm -r texinfo-4.13
rm texinfo-4.13a.tar.gz

wget http://cpansearch.perl.org/src/JHI/perl-5.8.1/pod/pod2man.PL
sed 's/local\/bin\/perl/bin\/perl/' pod2man.PL > pod2man
chmod +x pod2man
cp pod2man /usr/bin
```

10. Next we need Java:

```
##opkg install default-jdk
## -> this is not available as prebuild so we need to do it manually
## download the file from
## Note: wget does not work on https, so download via web browser, if
## required use the FTP client to transfer it from a host PC to the BBB
## http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-
1880260.html
wget http://download.oracle.com/otn-pub/java/jdk/7u45-b18/jdk-7u45-linux-arm-
vfp-sflt.tar.gz

tar xzf jdk-7u45-linux-arm-vfp-sflt.tar.gz
rm jdk-7u45-linux-arm-vfp-sflt.tar.gz
export PATH=$PATH:/home/root/jdk1.7.0_45/bin
export JAVA_HOME=/home/root/jdk1.7.0_45
# make it persitant
echo "PATH=$PATH:~/jdk1.7.0_45/bin" >> /etc/environment
echo "JAVA_HOME=~/jdk1.7.0_45" >> /etc/environment
opkg install librxtx-java
java -version

# rxtx lib
wget http://rxtx.qbang.org/pub/rxtx/rxtx-2.1-7r2.zip
unzip rxtx-2.1-7r2.zip
rm rxtx-2.1-7r2.zip
cd rxtx-2.1-7r2
chmod u+x *.sh
chmod u+x configure
./configure
make
make install
## Note: you can ignore the error message with RS485
cd ..
```

11. So we are now ready to build Ant:

```
##opkg install ant
## -> this is not available as prebuild so we need to do it manually
wget http://apache.mirror.digionline.de//ant/source/apache-ant-1.9.2-src.zip
unzip apache-ant-1.9.2-src.zip
wget http://search.maven.org/remotecontent?filepath=junit/junit/4.11/junit-
4.11.jar
mv junit-4.11.jar apache-ant-1.9.2/lib/optional/junit.jar
cd apache-ant-1.9.2
chmod u+x *.sh
./build.sh install -Dant.install=/usr/local
cd ..
rm -r apache-ant-1.9.2-src
rm apache-ant-1.9.2-src.zip
```

12. And finally the libraries for USB:

```
# add lib usb and setup usb port for user
##opkg install libusb-dev
## -> this is not available as prebuild so we need to do it manually
git clone http://git.libusb.org/libusb.git
git clone git://git.libusb.org/libusb-compat-0.1.git
cd libusb
./autogen.sh
make
make install
make clean
cd ..
export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig
cd libusb-compat-0.1
./autogen.sh
make
make install
make clean
cd ..
#rm -r libusb

echo 'ATTRS{idVendor}=="0451", ATTRS{idProduct}=="f432", MODE="0660",
GROUP="plugdev"' >> /etc/udev/rules.d/80-usb-msp430.rules
echo 'ATTRS{idVendor}=="0451", ATTRS{idProduct}=="f430", MODE="0660",
GROUP="plugdev"' >> /etc/udev/rules.d/80-usb-msp430.rules
usermod -aG plugdev ubuntu
```

13. Hey we are now already going for Energia -  that was easy:

```
# get Energia and required Packages
git clone --depth 1 git://github.com/energia/Energia.git
# this needs to be copied as it is not available now
#####cp Energia/build/windows/lm4f_tools.zip Energia/build/linux/
```

14. Actually the build.xml file needs some small modifications, while this are not merged into the main branch we just copy it, maybe I can also put it into the branch …xxx…:

```
cp build.xml Energia/build/
cd Energia/build/
ant
cp ../../rxtx-2.1-7r2/armv7l-unknown-linux-gnu/.libs/librxtxSerial.so
linux/work/lib/
```

15. Now Energia is ready and we can use a script in the Energia tree to build the MSP430 compiler tool chain.

```
cd linux/work/hardware/tools
cp ~/setup_files/gmp-4.3.2.tar.bz2 .
#chmod +x build-mspgcc
export BUNDLE_SUPPORT_LIBS=yes
sed 's/--content-disposition //' build-mspgcc > build-mspgcc.1
sed 's|ftp://gmplib.org/pub/gmp-${gmp_version}|http://ftp.gnu.org/gnu/gmp|'
build-mspgcc.1 > build-mspgcc-armchmod +x build-mspgcc-arm
./build-mspgcc-arm
cp -r ~/opt/mspgcc_energia/* /usr/local/
cd ~


wget http://sourceforge.net/projects/mspdebug/files/mspdebug-0.22.tar.gz
tar -xzvf mspdebug-0.22.tar.gz
cd mspdebug-0.22
make WITHOUT_READLINE=1
# Do the install as root (e.g., sudo)
```

```
make install
cd ~
```

16. Believe it or not – we are done. Of course you can now also build a release package and clean up your system to save some space.
    Happy Energia !!!
17.