***Croos\_compiling Bluez***

**INDEX**

Cross\_compiler for target architecture with **enable locale** should enabled in Toolchain options while compiling buildroot(generating cross toolchain)

**1. Prerequisites**

**1.1 GLib**

**1.1.1. zlib**

**1.1.2. libffi**

**1.1.3. gettext**

**1.1.4. GLib**

**1.2 D-Bus**

**1.2.1. The Expat XML Parser**

**1.2.2. D-Bus**

**1.3 libical**

**1.4 Readline**

**1.4.1. ncurses**

**1.4.2. Readline**

**2. Building BlueZ**

**3. Building Obex**

**4. Installing**

**5. Testing**

**Building Pre-requesites:**

**```````````````````````````````**

[*root@linux*](mailto:root@linux)*: mkdir -p /usr/local/{mini2440,mini2440/bluez}*

[root@linux](mailto:root@linux): *mkdir -p /embedd/mini2440/bluez*

[root@linux](mailto:root@linux): cd */embedd/mini2440/bluez*

[*root@linux*](mailto:root@linux)*: export INST\_DIR=/usr/local/mini2440/bluez*

[*root@linux*](mailto:root@linux)*: export PATH=$PATH:/path/2/mini2440/crosscompile/toolchian/ourput/host/usr/bin*

**1.1 Glib**

**1.1.1 zlib:**

[root@linux](mailto:root@linux): *wget -c* [*http://zlib.net/zlib-1.2.8.tar.gz*](http://zlib.net/zlib-1.2.8.tar.gz)

[root@linux](mailto:root@linux): *tar xvf zlib-1.2.8.tar.gz*

[root@linux](mailto:root@linux): *cd zlib-x.x.x*

[root@linux](mailto:root@linux): *prefix=/usr/local/mini2440/bluez CC=arm-linux-gcc CFLAGS="-O4" ./configure*

[root@linux](mailto:root@linux): *make && make install*

**Output:**

[root@linux](mailto:root@linux): build# *tree*

**.**

**|-- include**

**| |--** zconf.h

**| `--** zlib.h

**|-- lib**

**| |--** libz.a

**| |-- libz.so -> libz.so.1.2.8**

**| |-- libz.so.1 -> libz.so.1.2.8**

**| |-- libz.so.1.2.8**

**| `-- pkgconfig**

**| `--** zlib.pc

**`-- share**

**`-- man**

**`-- man3**

**`--** zlib.3

**6 directories, 8 files**

**1.1.2. libffi:**

[root@linux](mailto:root@linux): *wget -c* [*ftp://sourceware.org/pub/libffi/libffi-3.1.tar.gz*](ftp://sourceware.org/pub/libffi/libffi-3.1.tar.gz)

*OR*

[*root@linuc*](mailto:root@linuc)*: wget -c www.mirrorservice.org/sites/sourceware.org/pub/libffi/libffi-3.1.tar.gz*

[root@linux](mailto:root@linux): *tar xvf libffi-3.1.tar.gz*

[root@linux](mailto:root@linux): *cd libffi-3.1*

[root@linux](mailto:root@linux): *./configure --host=arm-linux --prefix=/usr/local/mini2440/bluez*

[root@linux](mailto:root@linux): *make && make install*

**Output:**

[root@linux](mailto:root@linux): **build#** *tree*

**.**

**|-- lib**

**| |-- libffi-3.1**

**| | `-- include**

**| | |-- ffi.h**

**| | `-- ffitarget.h**

**| |-- libffi.a**

**| |-- libffi.la**

**| |-- libffi.so -> libffi.so.6.0.2**

**| |-- libffi.so.6 -> libffi.so.6.0.2**

**| |-- libffi.so.6.0.2**

**| `-- pkgconfig**

**| `-- libffi.pc**

**`-- share**

**|-- info**

**| |-- dir**

**| `-- libffi.info**

**`-- man**

**`-- man3**

**|-- ffi.3**

**|-- ffi\_call.3**

**|-- ffi\_prep\_cif.3**

**`-- ffi\_prep\_cif\_var.3**

**8 directories, 14 files**

If you don't built your cross-toolchain with gettext support, you might be require this.

**1.1.3. gettext:**

**To Successfully build gettext the toolchain must have <enable locale support>**

**in buildroot <Toolchain options> otherwise libiconv has to be**

**crosscompiled manually for passthrough this step.**

[root@linux](mailto:root@linux): *wget -c* [*http://ftp.gnu.org/pub/gnu/gettext/gettext-0.19.1.tar.xz*](http://ftp.gnu.org/pub/gnu/gettext/gettext-0.19.1.tar.xz)

[root@linux](mailto:root@linux): *tar xvf gettext-0.19.1.tar.xz*

[root@linux](mailto:root@linux): *cd gettext-0.19.1.tar.xz*

[root@linux](mailto:root@linux): .*/configure --host=arm-linux --prefix=/usr/local/mini2440/bluez*

if it threw-up any error then give it.

[root@linux](mailto:root@linux): *./configure --host=arm-linux –prefix=/usr/local/mini2440/bluez gl\_cv\_func\_memchr\_works=yes ac\_cv\_func\_strnlen\_working=yes*

[root@linux](mailto:root@linux): *make && make install*

**1.1.4. Glib:**

GLib requires zlib, libffi successfully compiled for target architecture.

GLib requires the glib-genmarshal tool installed on your build system otherwise the following ***error*** is likey to result:

*checking for glib-genmarshal... no*

*configure: error: Could not find a glib-genmarshal in your PATH*

An easy way to resolve this is to install libglib2.0-dev:

[root@linux](mailto:root@linux): *sudo apt-get install libglib2.0-dev*

**Building:**

[root@linux](mailto:root@linux): *wget -c* [*http://ftp.gnome.org/pub/gnome/sources/glib/2.40/glib-2.40.0.tar.xz*](http://ftp.gnome.org/pub/gnome/sources/glib/2.40/glib-2.40.0.tar.xz)

[root@linux](mailto:root@linux): *tar -xvf glib-2.40.0.tar.xz*

[root@linux](mailto:root@linux): *cd glib-2.40.0*

[root@linux](mailto:root@linux): *./configure --host=arm-linux --prefix=/usr/local/mini2440/bluez PKG\_CONFIG\_PATH=/usr/local/mini2440/bluez/lib/pkgconfig CC="arm-linux-gcc -L/usr/local/mini2440/bluez/lib -I/usr/local/mini2440/bluez/include" glib\_cv\_stack\_grows=no glib\_cv\_uscore=yes ac\_cv\_func\_posix\_getpwuid\_r=yes ac\_cv\_func\_posix\_getgrgid\_r=yes*

[root@linux](mailto:root@linux): *make && make install*

**1.2 D-Bus**

D-Bus is a message bus system. It requires the ***expat*** library to successfully build otherwise the following ***error*** may result:

checking for XML\_ParserCreate\_MM in -lexpat... no

configure: error: Explicitly requested expat but expat not found

**1.2.1. The Expat XML Parser:**

Expat is an XML parser library written in C and is the only required fullfilling the dependency for the D-Bus daemon.

**Building:**

[*root@linux*](mailto:root@linux)*: wget -c* [*http://sourceforge.net/projects/expat/files/expat/2.1.0/expat-2.1.0.tar.gz*](http://sourceforge.net/projects/expat/files/expat/2.1.0/expat-2.1.0.tar.gz)

[*root@linux*](mailto:root@linux)*: tar -xvzf expat-2.1.0.tar.gz*

[*root@linxu*](mailto:root@linxu)*: cd expat-2.1.0/*

[*root@linux*](mailto:root@linux)*: ./configure --host=arm-linux --prefix=/usr/local/mini2440/bluez*

[*root@linxu*](mailto:root@linxu)*: make && make install*

**Output:**

[*root@linux*](mailto:root@linux)*:* **build***# tree*

.

|-- **bin**

| `-- **xmlwf**

|-- **include**

| |-- expat\_external.h

| `-- expat.h

|-- **lib**

| |-- libexpat.a

| |-- **libexpat.la**

| |-- libexpat.so -> **libexpat.so.1.6.0**

| |-- libexpat.so.1 -> **libexpat.so.1.6.0**

| |-- **libexpat.so.1.6.0**

| `-- **pkgconfig**

| `-- expat.pc

`-- **share**

`-- **man**

`-- **man1**

`-- xmlwf.1

7 directories, 10 files

**1.2.2. D-Bus:**

D-Bus is a message bus system, a simple way for applications to talk to one another. In addition to interprocess communication.

[root@linux](mailto:root@linux): *wget -c* [*http://dbus.freedesktop.org/releases/dbus/dbus-1.8.0.tar.gz*](http://dbus.freedesktop.org/releases/dbus/dbus-1.8.0.tar.gz)

[*root@linux*](mailto:root@linux)*: tar -xzvf dbus-1.8.0.tar.gz*

[*root@linux*](mailto:root@linux)*: cd dbus-1.8.0/*

[*root@linux*](mailto:root@linux)*: export PKG\_CONFIG\_LIBDIR=/usr/local/mini2440/bluez/lib/pkgconfig*

[*root@linux*](mailto:root@linux)*: echo ac\_cv\_have\_abstract\_sockets=yes > arm-linux.cache*

[*root@linux*](mailto:root@linux)*: ./configure --host=arm-linux --prefix=/usr/local/mini2440/bluez CC="arm-linux-gcc -L/usr/local/mini2440/bluez/lib -I/usr/local/mini2440/bluez/include -lexpat" --cache-file=arm-linux.cache*

[*root@linux*](mailto:root@linux)*: make && make install*

[*root@linux*](mailto:root@linux)*: make install DESTDIR=/nfs/bluezfs*

**1.3 libical**

Provides a read/write library of classes for object oriented languages (Initial goals of PHP and Python) that implement and enforce the iCal standard (RFC 2445).

libical requires ***cmake*** and ***g++*** cross compilers(while building toolchain make sure to select c++ support) to be installed:

[root@linux](mailto:root@linux): *sudo apt-get install cmake*

**Building:**

[root@linux](mailto:root@linux): *wget -c* [*http://downloads.sourceforge.net/freeassociation/libical-1.0.tar.gz*](http://downloads.sourceforge.net/freeassociation/libical-1.0.tar.gz)

[*root@linux*](mailto:root@linux)*: tar -xzvf libical-1.0.tar.gz*

[*root@linux*](mailto:root@linux)*: cd libical-1.0/*

[*root@linux*](mailto:root@linux)*: export CC=arm-linux-gcc*

[*root@linux*](mailto:root@linux)*: export CXX=arm-linux-g++*

[*root@linux*](mailto:root@linux)*: cmake -DCMAKE\_INSTALL\_PREFIX=/usr/local/mini2440/bluez*

[*root@linux*](mailto:root@linux)*: make && make install*

**1.4 Readline**

Readline references functions in ***ncurses***. If ncurses is not linked, the following ***errors*** are likely to result when building **BlueZ**:

/usr/arm-linux-gnueabi/lib/libreadline.so: undefined reference to `PC'

/usr/arm-linux-gnueabi/lib/libreadline.so: undefined reference to `tgetflag'

/usr/arm-linux-gnueabi/lib/libreadline.so: undefined reference to `tgetent'

/usr/arm-linux-gnueabi/lib/libreadline.so: undefined reference to `UP'

/usr/arm-linux-gnueabi/lib/libreadline.so: undefined reference to `tputs'

/usr/arm-linux-gnueabi/lib/libreadline.so: undefined reference to `tgoto'

/usr/arm-linux-gnueabi/lib/libreadline.so: undefined reference to `tgetnum'

/usr/arm-linux-gnueabi/lib/libreadline.so: undefined reference to `BC'

/usr/arm-linux-gnueabi/lib/libreadline.so: undefined reference to `tgetstr'

First we must build ncurses and then link it to Readline using**SHLIB\_LIBS=-lncurses.**

If that error doesn't occur the **SHLIB\_LIBS=-lncurses** optioncan beignored.

(it should be ignored while compiling for cubie2)

**1.4.1 ncurses**

[root@linux](mailto:root@linux): wget -c <http://ftp.gnu.org/pub/gnu/ncurses/ncurses-5.9.tar.gz>

[root@linux](mailto:root@linux): tar -xvzf ncurses-5.9.tar.gz

[root@linux](mailto:root@linux): cd ncurses-5.9

[root@linux](mailto:root@linux): ./configure --host=arm-linux --prefix=/usr/local/mini2440/bluez CXX="arm-linux-g++"

[root@linux](mailto:root@linux): make && make install

**Output:**

See ncurses\_output file in current directory.

**1.4.2 Readline**

[*root@linux*](mailto:root@linux)*: wget -c ftp://ftp.cwru.edu/pub/bash/readline-6.3.tar.gz*

[*root@linux*](mailto:root@linux)*: tar -xvzf readline-6.3.tar.gz*

[*root@linux*](mailto:root@linux)*: cd readline-6.3/*

[*root@linux*](mailto:root@linux)*: ./configure --host=arm-linux --prefix=/usr/local/mini2440/bluez bash\_cv\_wcwidth\_broken=yes*

[*root@linux*](mailto:root@linux)*: make CC="arm-linux-gcc -L/usr/local/mini2440/bluez/lib -I/usr/local/mini2440/bluez/include -lc" SHLIB\_LIBS=-lncurses*

[*root@linux*](mailto:root@linux)*: make install*

The ***bash\_cv\_wcwidth\_broken=yes*** parameter avoids the following ***error*** when cross-compiling:

checking for wcwidth broken with unicode combining characters...

configure: error: in `/.../readline-6.3':

configure: error: cannot run test program while cross compiling

**Output:**

See the readline\_output file in current working directory.

**Building BlueZ**

BlueZ is the official Bluetooth protocol stack for Linux and include tools such as hciattach, hciconfig, hcitool and rfcomm.

BlueZ 5.18 requires GLib >= 2.28, D-Bus >= 1.6 and libudev >= 143. When these prerequisites are meet, you can download, build and install BlueZ using:

[*root@linux*](mailto:root@linux)*: wget -c http://www.kernel.org/pub/linux/bluetooth/bluez-5.20.tar.xz*

[*root@linux*](mailto:root@linux)*: tar -xvf bluez-5.20.tar.xz*

[*root@linux*](mailto:root@linux)*: cd bluez-5.20*

***If dont want to cross-compile obex then give bellow command***

[*root@linux*](mailto:root@linux)*: ./configure --host=arm-linux --prefix=/usr/local/mini2440/bluez LIBS="-lncurses" PKG\_CONFIG\_PATH=/usr/local/mini2440/bluez/lib/pkgconfig --disable-systemd --disable-udev --disable-cups --disable-obex --enable-library CC="arm-linux-gcc -L/usr/local/mini2440/bluez/lib -I/usr/local/mini2440/bluez/include -lc"*

**If you want to cross-compile obex also for target board then give below**

[*root@linux*](mailto:root@linux)*: ./configure --host=arm-linux --prefix=/usr/local/mini2440/bluez LIBS="-lncurses" PKG\_CONFIG\_PATH=/usr/local/mini2440/bluez/lib/pkgconfig --disable-systemd --disable-udev --disable-cups --enable-library CC="arm-linux-gcc -L/usr/local/mini2440/bluez/lib -I/usr/local/mini2440/bluez/include -lc"*

[*root@linux*](mailto:root@linux)*: make*

[*root@linux*](mailto:root@linux)*: make install DESTDIR=/usr/local/mini2440/bluez*

[*root@linux*](mailto:root@linux)*: make install DESTDIR=/nfs/bluez*

Obex:

Obex needs all above packages and bluez cross-compiled.

If you configure bluez as --disable-obex then it obex throughs a following error.

Requires bluez >=1.40

Building:

[root@linux](mailto:root@linux): ./configure --host=arm-linux --prefix=/usr/local/mini2440/bluez PKG\_CONFIG\_PATH=/usr/local/mini2440/bluez/lib/pkgconfig CC="arm-linux-gcc -L/usr/local/mini2440/bluez/lib -I/usr/local/mini2440/bluez/include -lc"

[root@linux](mailto:root@linux): make && make install

**Installing**

Some of the BlueZ tools (e.g. sdptool requires libglib, bluetoothd requires libdbus-1, dbus-daemon requires libexpat, bluetoothctl requires libreadline) will require the following shared libraries installed:

[*root@linux*](mailto:root@linux)*: cp /usr/local/mini2440/bluez/lib/libglib-2.0.so.0.4000.0 /**nfs/bluezfs/lib*

[*root@linux*](mailto:root@linux)*: cp /usr/local/mini2440/bluez/lib/libdbus-1.so.3.8.3 /nfs/bluezfs/lib*

[*root@linux*](mailto:root@linux)*: cp /usr/local/mini2440/bluez/lib/libexpat.so.1.6.0 /nfs/bluezfs/lib*

[*root@linux*](mailto:root@linux)*: cp /usr/local/mini2440/bluez/lib/libreadline.so.6.3 /nfs/bluezfs/lib*

[*root@linux*](mailto:root@linux)*: mkdir -p /nfs/bluezfs/usr/{usr/local,usr/local/mini2440}*

[*root@linux*](mailto:root@linux)*: cp -Rfp /usr/local/mini2440/bluez /nfs/bluezfs/usr/local/mini2440/*

The Bluetooth Daemon requires the D-Bus Daemon to be running, otherwise the following error will result:

D-Bus setup failed: Failed to connect to socket /usr/arm-linux-gnueabi/var/run/dbus/system\_bus\_socket: No such file or directory

Copy the default bluetooth.conf over to dbus-1/systemd.d on the target:

*cp bluez-5.18/src/bluetooth.conf /home/export/rootfs/etc/dbus-1/system.d/*

The D-Bus Daemon wants to spawn from the user/group messagebus/messagebus. Failure to create a user and group results with:

Failed to start message bus: Could not get UID and GID for username "messagebus"

Create the appropriate group and user account:

[root@linux](mailto:root@linux): *grep root /etc/group > /nfs/bluezfs/etc/group*

[root@linux](mailto:root@linux): *grep root /etc/passwd > /nfs/bluezfs/etc/passwd*

[*root@mini2440*](mailto:root@mini2440)*# mkdir /home*

[*root@mini2440*](mailto:root@mini2440)*# mkdir /home/messagebus*

[*root@mini2440*](mailto:root@mini2440)*# addgroup -S messagebus*

[*root@mini2440*](mailto:root@mini2440)*# adduser -S messagebus -G messagebus*

Now start the D-Bus Daemon:

[root@mini2440](mailto:root@mini2440)# *dbus-daemon --system*

It will throw up an error like:

Unknown group "lp" in message bus configuration file

Failed to start message bus: The pid file "/usr/local/mini2440/bluez/var/run/dbus/pid" exists, if the message bus is e... blah blah blah

If pid error occurs go to that path and remove that file.

[root@mini2440](mailto:root@mini2440): *rm /usr/local/mini2440/bluez/var/run/dbus/pid*

[root@mini2440](mailto:root@mini2440)# *dbus-daemon –system*

Neglect this:

Unknown group "lp" in message bus configuration file

Finally, start the Bluetooth Daemon:

[root@mini2440](mailto:root@mini2440):# */libexec/bluetooth/bluetoothd &*

Testing

The hciconfig (BlueTooth Host Controller Interface Configuration utility) will show any BlueTooth host controllers present:

[root@mini2440](mailto:root@mini2440):# *hciconfig*

hci0: Type: BR/EDR Bus: USB

BD Address: 00:19:0E:15:5A:EF ACL MTU: 1021:8 SCO MTU: 64:1

DOWN

RX bytes:0 acl:0 sco:0 events:0 errors:0

TX bytes:0 acl:0 sco:0 commands:0 errors:0

You can bring up the interface by issuing:

[root@mini2440](mailto:root@mini2440):# *hciconfig hci0 up*

Now you can scan for BlueTooth devices:

[root@mini2440](mailto:root@mini2440):# *hcitool -i hci0 scan*

Scanning ...

7C:6D:62:9A:86:9A n/a

C8:19:F7:EF:D7:03 GT-P5100

B4:62:93:CF:82:F5 GT-I9300

Ping a BlueTooth Device:

[root@mini2440](mailto:root@mini2440):# *l2ping C8:19:F7:EF:D7:03*

Ping: C8:19:F7:EF:D7:03 from 00:19:0E:15:5A:EF (data size 44) ...

44 bytes from C8:19:F7:EF:D7:03 id 0 time 6.00ms

44 bytes from C8:19:F7:EF:D7:03 id 1 time 5.86ms

44 bytes from C8:19:F7:EF:D7:03 id 2 time 4.82ms

44 bytes from C8:19:F7:EF:D7:03 id 3 time 34.82ms

44 bytes from C8:19:F7:EF:D7:03 id 4 time 6.04ms

44 bytes from C8:19:F7:EF:D7:03 id 5 time 7.35ms

44 bytes from C8:19:F7:EF:D7:03 id 6 time 8.54ms

44 bytes from C8:19:F7:EF:D7:03 id 7 time 6.06ms

44 bytes from C8:19:F7:EF:D7:03 id 8 time 23.45ms

Interrogate SDP (Service Discovery Protocol) Descriptors:

[root@mini2440](mailto:root@mini2440):# *sdptool -i hci0 browse B4:62:93:CF:82:F5*

Starting Bluetoothctl program

[root@mini2440](mailto:root@mini2440):# *bluetoothctl*

[bluetooth]# *help*

[bluetooth]# *power on*

[bluetooth]# *scan on*

[bluetooth]# *trust blue\_addr*

[bluetooth]# *pair blue\_addr*

[bluetooth]# *discoverable on*

[bluetooth]# *pairable on*

[bluetooth]# *agent on*

[bluetooth]# *default-agent*

[bluetooth]# *exit*

[*root@mini2440*](mailto:root@mini2440)*:*