

1. Vorbereiten das Host, folgende Software Pakete müssen installiert werden:

- gnome-devel
- bison
- flex
- texinfo
- libncurses5-dev
- git
- codeblocks
- putty

2. Buildroot installieren

Buildroot (www.buildroot.org) – erzeugt den Cross-compiler und das „root Filesystem“ für das Target System.

Wichtig: Buildroot sollte unterhalb des „opt“ Verzeichnis installiert werden, wählen Sie kein anderes Verzeichnis! Wenn doch müssen Sie einige Pfadangaben anpassen!

2.1 Download:

```
Host$cd /opt
host$git clone git://git.buildroot.net/buildroot
```

Sollte der „git“ Port über eine Firewall gesperrt sein benutzen Sie

```
host$git clone http://git.buildroot.net/git/buildroot.git
```

2.2 Buildroot konfigurieren

```
host$cd buildroot
host$make gconfig
```

Einstellungen:

Target Architecture → arm

Target Architecture Variant → arm926t

Target ABI → EABI

Build options → Nichts ändern

Toolchain

Kernel Headers → 2.6.34.x

Alle anderen Einstellungen entsprechend Screenshots

Options	Name	N	M	Y	Value
Build options					
Toolchain					
Toolchain type					Buildroot toolchain
Kernel Header Options					
Kernel Headers					Linux 2.6.35.x kernel header
uClibc Options					
uClibc C library Version					uClibc 0.9.31.x
uClibc configuration file to use?	BR2_UCLIBC_CONFIG				toolchain/uClibc/uClibc-0.9.
Thread library debugging	BR2_PTHREAD_DEBUG			Y	Y
Compile and install uClibc tests	BR2_UCLIBC_INSTALL_TEST_SUITE	N			N
Binutils Options					
Binutils Version					binutils 2.21
Additional binutils options	BR2_BINUTILS_EXTRA_CONFIG_OPTIONS				
GCC Options					
GCC compiler Version					gcc 4.3.x
Additional gcc options	BR2_EXTRA_GCC_CONFIG_OPTIONS				
Objective-C cross-compiler support	BR2_GCC_CROSS_OBJC	N			N
Fortran cross-compiler support	BR2_GCC_CROSS_FORTRAN	N			N
Build/install Objective-C compiler and runtime?	BR2_INSTALL_OBJC	N			N
Build/install Fortran compiler and runtime?	BR2_INSTALL_FORTRAN	N			N
Build/install a shared libgcc?	BR2_GCC_SHARED_LIBGCC			Y	Y
Gdb Options					
Build gdb debugger for the Target (NEW)	BR2_PACKAGE_GDB	N			N
Build gdb server for the Target	BR2_PACKAGE_GDB_SERVER			Y	Y
Build gdb for the Host	BR2_PACKAGE_GDB_HOST			Y	Y
GDB debugger Version (NEW)					gdb 6.8
Purge unwanted locales	BR2_ENABLE_LOCALE_PURGE	N			N
Use software floating point by default	BR2_SOFT_FLOAT			Y	Y
Target Optimizations	BR2_TARGET_OPTIMIZATION				-pipe
Toolchain Options					
Enable large file (files > 2 GB) support	BR2_TOOLCHAIN_BUILDROOT_LARGEFILE			Y	Y
Enable IPv6 support	BR2_TOOLCHAIN_BUILDROOT_INET_IPV6	N			N
Enable RPC support	BR2_TOOLCHAIN_BUILDROOT_INET_RPC			Y	Y
Enable WCHAR support	BR2_TOOLCHAIN_BUILDROOT_WCHAR			Y	Y
Enable toolchain locale/i18n support	BR2_TOOLCHAIN_BUILDROOT_LOCALE	N			N
Enable 'program invocation name' support	BR2_TOOLCHAIN_BUILDROOT_PROGRAM_INVOCATION	N			N
Enable C++ support	BR2_TOOLCHAIN_BUILDROOT_CXX			Y	Y
Enable stack protection support	BR2_TOOLCHAIN_BUILDROOT_USE_SSP	N			N
Thread library implementation					linuxthreads (stable/old)
Enable elf2flt support?	BR2_ELF2FLT	N			N
Run mklibs on the built root filesystem	BR2_MKLIBS	N			N
System configuration					
Package Selection for the target					
Target filesystem options					

System configuration

System configuration					
System hostname	BR2_TARGET_GENERIC_HOSTNAME				tmpa900
System banner	BR2_TARGET_GENERIC_ISSUE				Welcome to TMPA900
Generic serial port config	BR2_TARGET_GENERIC_GETTY			Y	Y
Serial port to run a getty on (NEW)	BR2_TARGET_GENERIC_GETTY_PORT				ttyS0
Baudrate to use (NEW)					115200
keep kernel default (NEW)	BR2_TARGET_GENERIC_GETTY_BAUDRATE_KEEP	N			N
9600 (NEW)	BR2_TARGET_GENERIC_GETTY_BAUDRATE_9600	N			N
19200 (NEW)	BR2_TARGET_GENERIC_GETTY_BAUDRATE_19200	N			N
38400 (NEW)	BR2_TARGET_GENERIC_GETTY_BAUDRATE_38400	N			N
57600 (NEW)	BR2_TARGET_GENERIC_GETTY_BAUDRATE_57600	N			N
115200 (NEW)	BR2_TARGET_GENERIC_GETTY_BAUDRATE_115200			Y	Y
Package Selection for the target					
Target filesystem options					

Package Selection for the target

Options	Name	N	M	Y	Value
▼ Package Selection for the target					
▼ <input checked="" type="checkbox"/> BusyBox	BR2_PACKAGE_BUSYBOX	-		Y	Y
▶ <input type="checkbox"/> BusyBox Version					BusyBox 1.18.x
BusyBox configuration file to use?	BR2_PACKAGE_BUSYBOX_CONFIG				package/busybox/busybox-1.18.x.config
<input type="checkbox"/> Show packages that are also provided by busybox	BR2_PACKAGE_BUSYBOX_SHOW_OTHERS	N		-	N
<input type="checkbox"/> customize	BR2_PACKAGE_CUSTOMIZE	N		-	N
▶ Audio and video libraries and applications					
▶ Compressors and decompressors					
▶ Debugging, profiling and benchmark					
▼ Development tools					
<input type="checkbox"/> autoconf	BR2_PACKAGE_AUTOCONF	N		-	N
<input type="checkbox"/> automake	BR2_PACKAGE_AUTOMAKE	N		-	N
<input type="checkbox"/> libbfd (binutils)	BR2_PACKAGE_BINUTILS	N		-	N
<input type="checkbox"/> bison	BR2_PACKAGE_BISON	N		-	N
<input type="checkbox"/> bsdiff	BR2_PACKAGE_BSDIFF	N		-	N
<input type="checkbox"/> ccache	BR2_PACKAGE_CCACHE	N		-	N
<input type="checkbox"/> cvs	BR2_PACKAGE_CVS	N		-	N
<input type="checkbox"/> distcc	BR2_PACKAGE_DISTCC	N		-	N
<input type="checkbox"/> fakeroot	BR2_PACKAGE_FAKEROOT	N		-	N
<input type="checkbox"/> flex	BR2_PACKAGE_FLEX	N		-	N
native toolchain needs development files in target filesystem					
<input checked="" type="checkbox"/> gettext	BR2_PACKAGE_GETTEXT			Y	Y
<input type="checkbox"/> Use libgettext.a instead of libgettext.so.*	BR2_PACKAGE_GETTEXT_STATIC	N		-	N
<input checked="" type="checkbox"/> libintl	BR2_PACKAGE_LIBINTL			Y	Y
<input type="checkbox"/> gmp	BR2_PACKAGE_GMP	N		-	N
<input type="checkbox"/> gperf	BR2_PACKAGE_GPERF	N		-	N
<input type="checkbox"/> make	BR2_PACKAGE_MAKE	N		-	N
<input type="checkbox"/> mpc	BR2_PACKAGE_MPC	N		-	N
<input type="checkbox"/> mpfr	BR2_PACKAGE_MPFR	N		-	N
<input type="checkbox"/> libtool	BR2_PACKAGE_LIBTOOL	N		-	N
<input type="checkbox"/> m4	BR2_PACKAGE_M4	N		-	N
<input type="checkbox"/> pkg-config	BR2_PACKAGE_PKG_CONFIG	N		-	N
<input type="checkbox"/> sstrip	BR2_PACKAGE_SSTRIP	N		-	N
▶ Games					
▼ Graphic libraries and applications (graphic/text)					
Graphic applications					
<input type="checkbox"/> rrdtool (NEW)	BR2_PACKAGE_RRDTOOL			N	- N
graphic libraries					
▼ <input checked="" type="checkbox"/> directfb	BR2_PACKAGE_DIRECTFB			-	Y Y
<input type="checkbox"/> directfb debugging (NEW)	BR2_PACKAGE_DIRECTFB_DEBUG			N	- N
<input type="checkbox"/> directfb call trace support (NEW)	BR2_PACKAGE_DIRECTFB_TRACE			N	- N
<input type="checkbox"/> compile pxa3xx graphics driver (NEW)	BR2_PACKAGE_DIRECTFB_PXA3XX			N	- N
<input checked="" type="checkbox"/> compile /dev/input/eventX input driver (NEW)	BR2_PACKAGE_DIRECTFB_LINUXINPUT			-	Y Y
<input checked="" type="checkbox"/> compile keyboard input driver (NEW)	BR2_PACKAGE_DIRECTFB_KEYBOARD			-	Y Y
<input type="checkbox"/> compile PS2 mouse input driver	BR2_PACKAGE_DIRECTFB_PS2MOUSE			N	- N
<input type="checkbox"/> compile serial mouse input driver	BR2_PACKAGE_DIRECTFB_SERIALMOUSE			N	- N
<input checked="" type="checkbox"/> enable touchscreen support (NEW)	BR2_PACKAGE_DIRECTFB_TSLIB			-	Y Y
<input checked="" type="checkbox"/> enable GIF support (NEW)	BR2_PACKAGE_DIRECTFB_GIF			-	Y Y
<input checked="" type="checkbox"/> enable JPEG support (NEW)	BR2_PACKAGE_DIRECTFB_JPEG			-	Y Y
<input checked="" type="checkbox"/> enable PNG support (NEW)	BR2_PACKAGE_DIRECTFB_PNG			-	Y Y
<input type="checkbox"/> enable advanced dithering of RGB16 surfaces (NEW)	BR2_PACKAGE_DIRECTFB_DITHER_RGB16			N	- N
<input type="checkbox"/> build directfb tests (NEW)	BR2_PACKAGE_DIRECTFB_TESTS			N	- N
<input type="checkbox"/> directfb examples (NEW)	BR2_PACKAGE_DIRECTFB_EXAMPLES			N	- N
<input type="checkbox"/> directfb virtual input extension (NEW)	BR2_PACKAGE_DIVINE			N	- N
<input type="checkbox"/> fbdump (Framebuffer Capture Tool)	BR2_PACKAGE_FBDUMP			N	- N
<input type="checkbox"/> fbgrab	BR2_PACKAGE_FBGRAB			N	- N
<input type="checkbox"/> fbv	BR2_PACKAGE_FBV			N	- N

<input type="checkbox"/> imagemagick	BR2_PACKAGE_IMAGEMAGICK	N	-	N	
<input type="checkbox"/> LiTE (toolbox engine) (NEW)	BR2_PACKAGE_LITE	N	-	N	
<input type="checkbox"/> SawMan (Window Manager) (NEW)	BR2_PACKAGE_SAWMAN	N	-	N	
<input type="checkbox"/> SDL	BR2_PACKAGE_SDL	N	-	N	
other GUIs					
<input type="checkbox"/> Qt (NEW)	BR2_PACKAGE_QT	N	-	N	
<input type="checkbox"/> X.org X Window System, X11R7, release 7.5 (NEW)	BR2_PACKAGE_XORG7	N	-	N	
X libraries and helper libraries					
<input type="checkbox"/> Liberation (Free fonts)	BR2_PACKAGE_LIBERATION	N	-	N	
X Window managers					
X applications					
<input type="checkbox"/> gob2 (NEW)	BR2_PACKAGE_GOB2	N	-	N	
midori requires C++, WCHAR in toolchain and libgtk2					
▸ Hardware handling					
▸ Interpreter languages and scripting					
▾ Libraries					
▾ Compression and decompression					
<input type="checkbox"/> libarchive (NEW)	BR2_PACKAGE_LIBARCHIVE	N	-	N	
<input type="checkbox"/> lzo	BR2_PACKAGE_LZO	N	-	N	
<input checked="" type="checkbox"/> zlib	BR2_PACKAGE_ZLIB			Y Y	
▸ Crypto					
Options	Name	N	M	Y	Value
▾ Package Selection for the target					
▸ <input checked="" type="checkbox"/> BusyBox	BR2_PACKAGE_BUSYBOX	-		Y	Y
<input type="checkbox"/> customize	BR2_PACKAGE_CUSTOMIZE	N	-	N	
▸ Audio and video libraries and applications					
▸ Compressors and decompressors					
▸ Debugging, profiling and benchmark					
▸ Development tools					
▸ Games					
▾ Graphic libraries and applications (graphic/text)					
Graphic applications					
<input type="checkbox"/> rrdtool (NEW)	BR2_PACKAGE_RRDTOOL	N	-	N	
graphic libraries					
▾ <input checked="" type="checkbox"/> directfb	BR2_PACKAGE_DIRECTFB	-		Y	Y
<input type="checkbox"/> directfb debugging (NEW)	BR2_PACKAGE_DIRECTFB_DEBUG	N	-	N	
<input type="checkbox"/> directfb call trace support (NEW)	BR2_PACKAGE_DIRECTFB_TRACE	N	-	N	
<input type="checkbox"/> compile pxa3xx graphics driver (NEW)	BR2_PACKAGE_DIRECTFB_PXA3XX	N	-	N	
<input checked="" type="checkbox"/> compile /dev/input/eventX input driver (NEW)	BR2_PACKAGE_DIRECTFB_LINUXINPUT	-		Y	Y
<input checked="" type="checkbox"/> compile keyboard input driver (NEW)	BR2_PACKAGE_DIRECTFB_KEYBOARD	-		Y	Y
<input checked="" type="checkbox"/> compile PS2 mouse input driver	BR2_PACKAGE_DIRECTFB_PS2MOUSE	N	-	N	
<input type="checkbox"/> compile serial mouse input driver	BR2_PACKAGE_DIRECTFB_SERIALMOUSE	N	-	N	
<input checked="" type="checkbox"/> enable touchscreen support (NEW)	BR2_PACKAGE_DIRECTFB_TSLIB	-		Y	Y
<input checked="" type="checkbox"/> enable GIF support (NEW)	BR2_PACKAGE_DIRECTFB_GIF	-		Y	Y
<input checked="" type="checkbox"/> enable JPEG support (NEW)	BR2_PACKAGE_DIRECTFB_JPEG	-		Y	Y
<input checked="" type="checkbox"/> enable PNG support (NEW)	BR2_PACKAGE_DIRECTFB_PNG	-		Y	Y
<input type="checkbox"/> enable advanced dithering of RGB16 surfaces (NEW)	BR2_PACKAGE_DIRECTFB_DITHER_RGB16	N	-	N	
<input type="checkbox"/> build directfb tests (NEW)	BR2_PACKAGE_DIRECTFB_TESTS	N	-	N	
<input type="checkbox"/> directfb examples (NEW)	BR2_PACKAGE_DIRECTFB_EXAMPLES	N	-	N	
<input type="checkbox"/> directfb virtual input extension (NEW)	BR2_PACKAGE_DIVINE	N	-	N	
<input type="checkbox"/> fbdump (Framebuffer Capture Tool)	BR2_PACKAGE_FBDUMP	N	-	N	
<input type="checkbox"/> fbgrab	BR2_PACKAGE_FBGRAB	N	-	N	
<input type="checkbox"/> fbv	BR2_PACKAGE_FBV	N	-	N	
<input type="checkbox"/> imagemagick	BR2_PACKAGE_IMAGEMAGICK	N	-	N	

Options	Name	N	M	Y	Value
▶ Graphics					
▼ Hardware handling					
<input type="checkbox"/> libaio	BR2_PACKAGE_LIBAIO	N			N
<input type="checkbox"/> libraw1394	BR2_PACKAGE_LIBRAW1394	N			N
<input checked="" type="checkbox"/> libts - The Touchscreen tslib Library	BR2_PACKAGE_TSLIB			Y	Y
<input type="checkbox"/> libusb	BR2_PACKAGE_LIBUSB	N			N
▶ Networking					
▼ Other					
<input type="checkbox"/> argp-standalone	BR2_PACKAGE_ARGP_STANDALONE	N			N
<input type="checkbox"/> libatomic_ops	BR2_PACKAGE_LIBATOMIC_OPS	N			N
<input type="checkbox"/> libcap	BR2_PACKAGE_LIBCAP	N			N
<input type="checkbox"/> libdaemon	BR2_PACKAGE_LIBDAEMON	N			N
<input type="checkbox"/> libelf	BR2_PACKAGE_LIBELF	N			N
<input type="checkbox"/> libevent	BR2_PACKAGE_LIBEVENT	N			N
<input checked="" type="checkbox"/> libglib2 (NEW)	BR2_PACKAGE_LIBGLIB2			Y	Y
<input type="checkbox"/> liboil	BR2_PACKAGE_LIBOIL	N			N
▼ Text and terminal handling					
<input type="checkbox"/> enchant (NEW)	BR2_PACKAGE_ENCHANT	N			N
<input type="checkbox"/> icu (NEW)	BR2_PACKAGE_ICU	N			N
<input checked="" type="checkbox"/> libiconv	BR2_PACKAGE_LIBICONV			Y	Y
<input type="checkbox"/> ncurses	BR2_PACKAGE_NCURSES	N			N
<input type="checkbox"/> newt	BR2_PACKAGE_NEWT	N			N
<input type="checkbox"/> pcre	BR2_PACKAGE_PCRE	N			N
<input type="checkbox"/> popt	BR2_PACKAGE_POPT	N			N
<input type="checkbox"/> readline	BR2_PACKAGE_READLINE	N			N
<input type="checkbox"/> slang	BR2_PACKAGE_SLANG	N			N
▼ XML					
<input checked="" type="checkbox"/> expat	BR2_PACKAGE_EXPAT			Y	Y
<input type="checkbox"/> ezxml	BR2_PACKAGE_EZXML	N			N
<input checked="" type="checkbox"/> libxml2	BR2_PACKAGE_LIBXML2			Y	Y
<input type="checkbox"/> libxslt	BR2_PACKAGE_LIBXSLT	N			N
<input type="checkbox"/> xerces-c++ (NEW)	BR2_PACKAGE_XERCES	N			N
▶ Miscellaneous					

Target filesystem options

▼ Target filesystem options					
Custom script to run before packing files	BR2_ROOTFS_POST_BUILD_SCRIPT				
Path to the device table	BR2_ROOTFS_DEVICE_TABLE				target/generic/device_
▼ <input type="checkbox"/> Root FS skeleton					default target skeleton
<input checked="" type="radio"/> default target skeleton	BR2_ROOTFS_SKELETON_DEFAULT			Y	Y
<input type="radio"/> custom target skeleton	BR2_ROOTFS_SKELETON_CUSTOM	N			N
<input type="checkbox"/> cramfs root filesystem	BR2_TARGET_ROOTFS_CRAMFS	N			N
<input type="checkbox"/> cloop root filesystem for the target device	BR2_TARGET_ROOTFS_CLOOP	N			N
<input type="checkbox"/> ext2 root filesystem	BR2_TARGET_ROOTFS_EXT2	N			N
<input type="checkbox"/> jffs2 root filesystem	BR2_TARGET_ROOTFS_JFFS2	N			N
<input type="checkbox"/> ubifs root filesystem	BR2_TARGET_ROOTFS_UBIFS	N			N
<input type="checkbox"/> squashfs root filesystem	BR2_TARGET_ROOTFS_SQUASHFS	N			N
▼ <input checked="" type="checkbox"/> tar the root filesystem	BR2_TARGET_ROOTFS_TAR			Y	Y
▼ <input type="checkbox"/> Compression method (NEW)					no compression
<input checked="" type="radio"/> no compression (NEW)	BR2_TARGET_ROOTFS_TAR_NONE			Y	Y
<input type="radio"/> gzip (NEW)	BR2_TARGET_ROOTFS_TAR_GZIP	N			N
<input type="radio"/> bzip2 (NEW)	BR2_TARGET_ROOTFS_TAR_BZIP2	N			N
<input type="radio"/> lzma (NEW)	BR2_TARGET_ROOTFS_TAR_LZMA	N			N
other random options to pass to tar (NEW)	BR2_TARGET_ROOTFS_TAR_OPTIONS				
<input type="checkbox"/> cpio the root filesystem	BR2_TARGET_ROOTFS_CPIO	N			N
initramfs requires a Linux kernel to be built					
<input type="checkbox"/> romfs root filesystem	BR2_TARGET_ROOTFS_ROMFS	N			N

Speichern und beenden Sie die Konfiguration

2.3 Quellen herunterladen

```
host$make source
```

2.4 Quellen übersetzen

```
host$make
```

3. Das „Root File System“ auf dem NFS Server kopieren
 1. Auf dem NFS Server das Verzeichnis „rootfs“ anlegen
 2. Root File System auf dem Server entpacken

```
host$cd /opt/buildroot/output/images  
host$sudo tar -xvf rootfs.tar -C /.../rootfs
```

.... Verzeichnis anpassen

Hinweis: Damit man als „root“ auf einen NFS Server zugreifen kann muss die Option „no_root_squash“ in der „exports“ Datei ergänzt werden.

Beispiel:

```
/rootfs (rw,sync,no_root_squash)
```

Sie haben es geschafft!

3.1 Testen das „Root File Systems“

1. Starten Sie PuTTY
2. Wählen Sie Connection type: Serial, Serial line: <Das Device Ihrer Schnittstelle am PC>, Speed: 115200
3. Verbinden Sie das „Null Modem Kabel“ mit der PC Schnittstelle
4. Schließen Sie das Ethernet-Kabel an
5. Schließen Sie die Spannungsversorgung am EVA Board an
6. Nach einer kurzen Boot Zeit können Sie sich über die Serielle Console als root einloggen.

4. Die Entwicklungsumgebung CODE::BLOCKS

Damit das CODE::BLOCKS oder auch selbst erstellte „Make-Files“ die CrossCompiler Umgebung nutzen können müssen einige Umgebungsvariablen richtig gesetzt werden, was von dem Shell Script „environment-setup“ erledigt wird. Damit das Script nicht als eigene Shell ausgeführt wird muss es mit dem „Punkt“ Kommando gestartet werden.

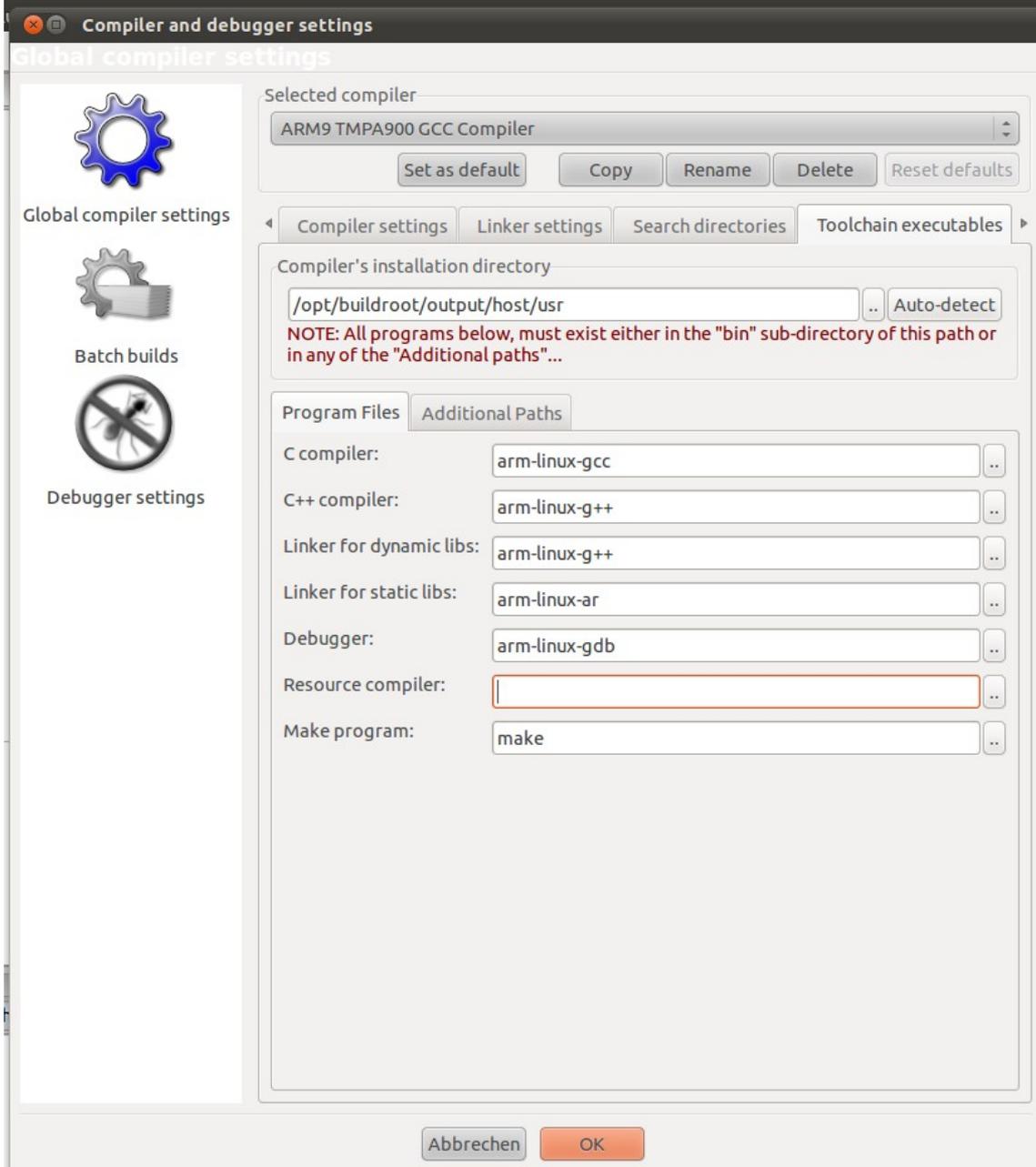
```
Host$. environment-setup
```

Vergessen Sie nicht das Leerzeichen zwischen „.“ und „environment-setup“.
Starten Sie nun CODE::BLOCKS von der shell aus.

```
host$codeblocks
```

4.1 Compiler in CODE::BLOCKS einrichten

1. Menü „Settings“ → „Compiler and debugger“ öffnen
2. „Selected compiler“: „GNU ARM GCC Compiler“ auswählen
3. „Copy“ klicken
4. Als Namen für den neuen Compiler „ARM9 TMPA900 GCC Compiler“ eingeben
5. Toolchain executables entsprechend Screenshot eingeben



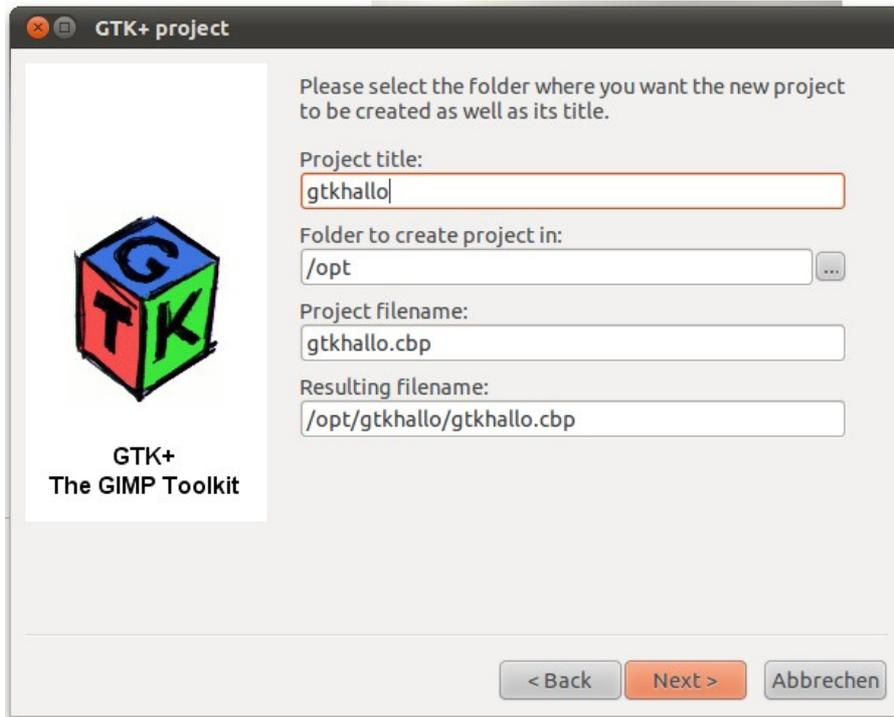
6. Den Dialog mit Ok schließen

5. Hello World Programm erstellen

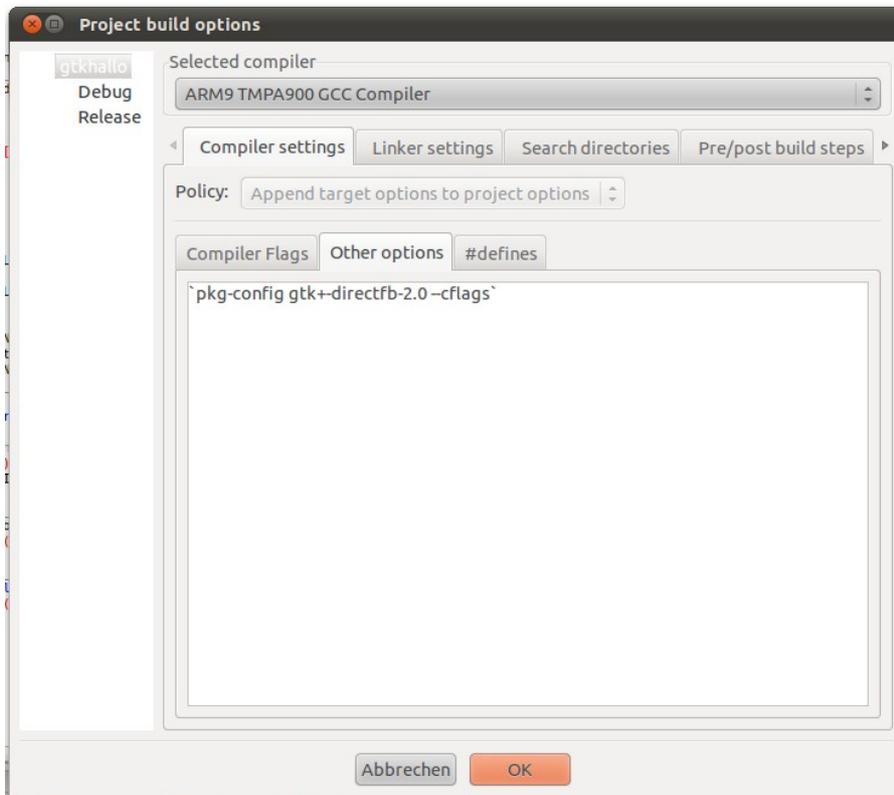
Das Projekt „Hello Word“ ist bereits Bestandteil dieses Tutorials, die beschreibung dient nur als Anleitung für eigene Projekte, der Automatisch erzeugte Quellcode ist auch für unsere Embedded Anwendung nicht zu gebrauchen.

1. Neues GTK Projekt anlegen, Menü „File“ → „New“ → „Project“ → „GTK+ project“ auswählen, „Ok“, „Next“

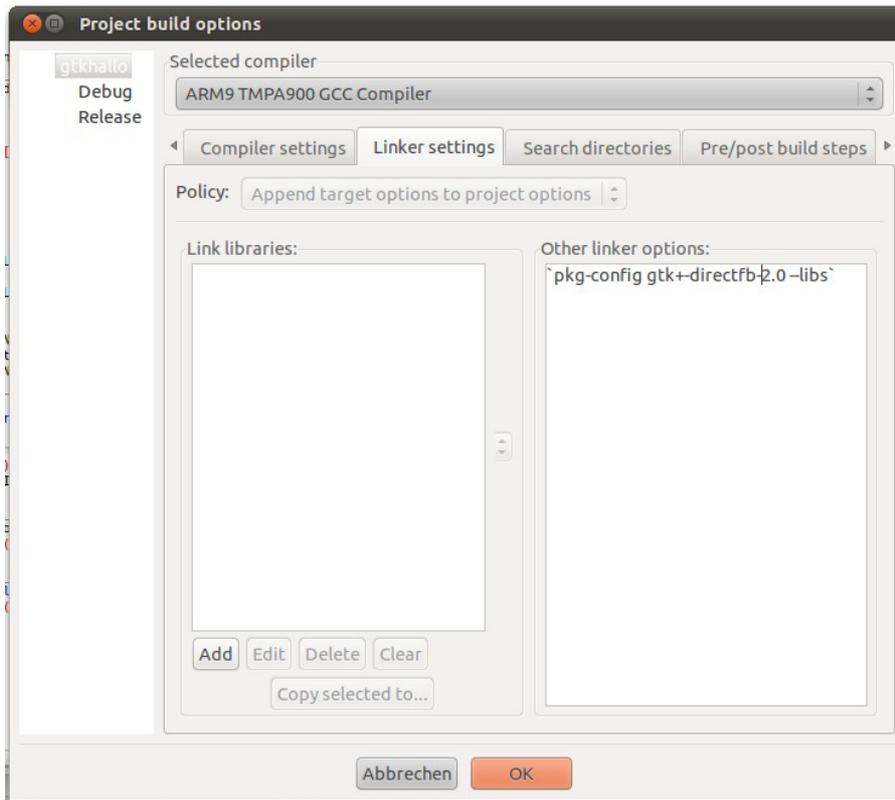
2. Project Namen, Arbeitsverzeichnis eingeben



3. Weiter mit Next und Dialog mit Finish schließen
4. Compiler umstellen, Menü „Project“ → „Build options...“
5. Am linken Rand „gtkhallo“ auswählen, damit die Einstellungen für „Debug“ und „Release“ gültig sind
6. Selected Compiler auf „ARM9 TMPA900 GCC Compiler“ umstellen
7. Compiler Flags

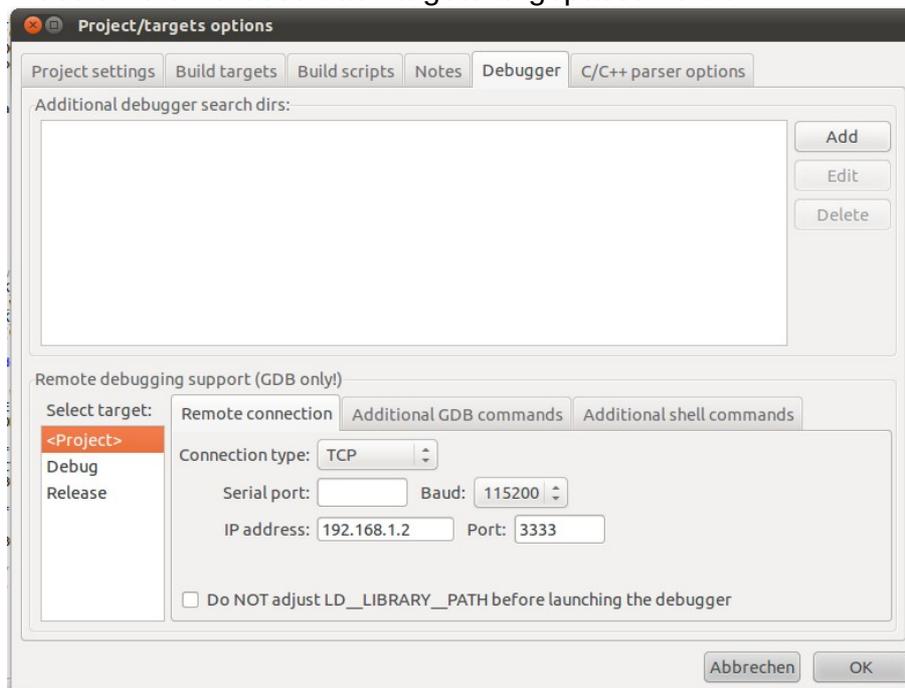


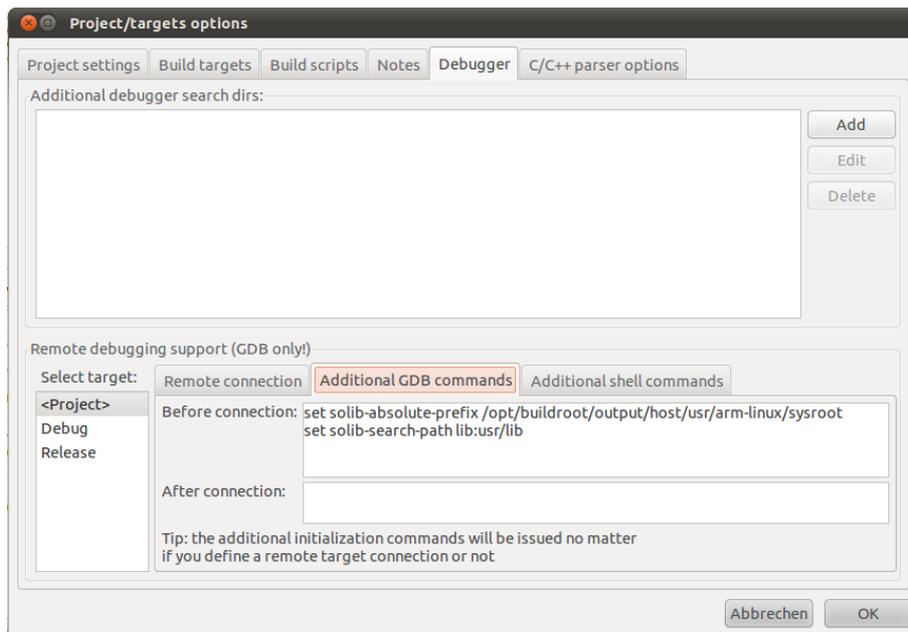
8. Einstellungen für den Linker



9. Dialog schließen

10. Debugger konfigurieren, Menü „Project“ → „Properties“, die IP adresse muss an die Adresse das Targets angepasst werden





6. Tiny-CAN

6.1 Tiny-CAN Treiber kompilieren

```
host$cd /opt/LinuxGo
host$. environment-setup
host$cd treiber/mhstcan/tmpa900
host$make
```

Die Datei „libmhstcan.so“ ins „opt“ Verzeichnis des Targets kopieren

6.2 Embedded CAN-View kompilieren

```
host$cd /opt/LinuxGo/target
host$make
host$cd bin/Debug
```

Die Dateien „COPYING“, „emb_can_view“ und „tiny-can.png“ ins „opt“ Verzeichnis des Targets kopieren

„emb_can_view“ kann auch mit Code::Blocks kompiliert werden, laden Sie das Projekt „emb_can_view.cbp“

7. Touch

Umgebungsvariable für den Touch setzen

```
target$export TSLIB_TSDEVICE=/dev/input/event0
```

Touch kalibrieren

```
target$ts_calibrate
```

Touch testen

```
target$ts_test
```

Dateien/Verzeichnisse:

environment-setup	Script zum setzen der Umgebungsvariablen für Cross-Compiling
can_api	Tiny-CAN API Files
emb_can_view	Embedded Tiny-CAN View
trieber	Quellen Tiny-CAN API Treiber
gtkhallo	GTK „Hello World“ Demo Programm
sample1	Test Programm für die Console

Probleme:

Wenn ich eines der GTK Programme beende hängt sich das ganze auf.
Das debuggen in den GTK Programmen funktioniert noch nicht
Das „sample1“ kann man mit Code::Blocks debuggen.

Auf dem Target mit gdbserver starten:

```
target$gdbserver :3333 sample1
```

Die Bootparameter, der Kernel befindet sich auf dem Target das „root“ Filesystem auf dem NFS Server.

```
bootargs_base=setenv bootargs console=ttyS0,115200n8 ${mtdparts} root=/dev/nfs  
nfsroot=192.168.1.99:/data/rootfs ip=192.168.1.2 4{videoparams} ethaddr=${ethaddr}
```

```
bootcmd=run bootargs_base; nboot kernel; bootm
```

Anbei auch noch der Kernel und Uboot den ich verwende