Automotive 24GHz radar development kit

Automotive radar kit

About this document

Scope and purpose

This is the Usermanual for the Automotive 24GHz radar development kit.

Table of Contents

1 Hardware und Software Setup Instruction

1.1 The Automotive radar kit contains the following items:

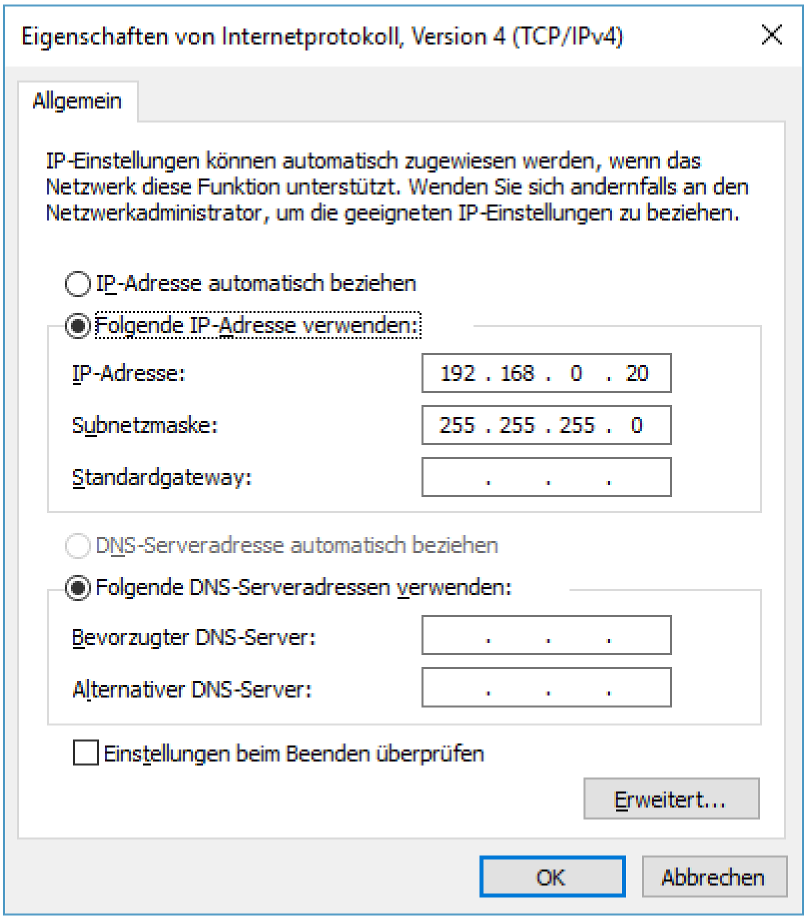
- Automotive radar kitAutomotive radar kit board
- Y supply line with power plug and CAN DSUB9 connector
- Plug-in power supply
- Ethernet straight cable to connect your PC to the Automotive radar kitAutomotive radar kit
- USB Stick with Radar Control Panels Software and documentation

1.2 Installation and Setup:

The procedure is very simple. Please follow the next steps

- Configure the IP address of your PC Ethernet Interface.
- Connect the PC and Automotive radar kit.
- Start “Radar Control Pannel.exe” application on your PC.
-

1.2.1 Network Interface settings



Eigenschaften von Internetprotokoll, Version 4 (TCP/IPv4)

Allgemein

IP-Einstellungen können automatisch zugewiesen werden, wenn das Netzwerk diese Funktion unterstützt. Wenden Sie sich andernfalls an den Netzwerkadministrator, um die geeigneten IP-Einstellungen zu beziehen.

○ IP-Adresse automatisch beziehen
● Folgende IP-Adresse verwenden:

IP-Adresse: 192 . 168 . 0 . 20
Subnetzmaske: 255 . 255 . 255 . 0
Standardgateway: . . .

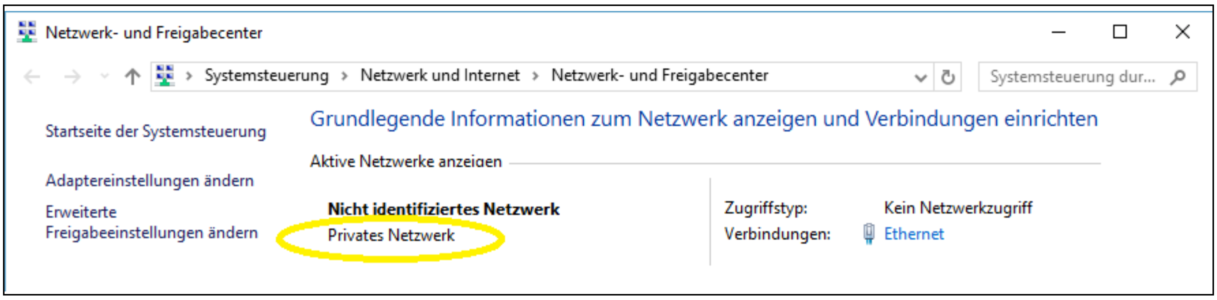
○ DNS-Serveradresse automatisch beziehen
● Folgende DNS-Serveradressen verwenden:

Bevorzugter DNS-Server: . . .
Alternativer DNS-Server: . . .

□ Einstellungen beim Beenden überprüfen

Figure 1 IP Adress settings

The processing board has its IP address pre-configured to: 192.168.0.30. You will have to configure the IP Address of your Network Interface to 192.168.0.20 with Subnet mask 255.255.255.0 in order to communicate with the Automotive radar kit.
Windows will ask for the Network type. It is important to choose Private Network, otherwise, the Firewall will block UDP and/or TCP traffic.


Netzwerk- und Freigabecenter
Systemsteuerung > Netzwerk und Internet > Netzwerk- und Freigabecenter
Systemsteuerung dur...
Startseite der Systemsteuerung
Adaptereinstellungen ändern
Erweiterte Freigabeeinstellungen ändern

Grundlegende Informationen zum Netzwerk anzeigen und Verbindungen einrichten

Aktive Netzwerke anzeigen

Nicht identifiziertes Netzwerk
Privates Netzwerk

Zugriffstyp: Kein Netzwerkzugriff
Verbindungen: Ethernet

Figure 2 Typ of Network

1.2.2 Connect the PC and the Automotive radar kit

Make sure you have the Automotive radar kit, Y supply line, Ethernet straight cable and Plug-in power supply handy.

- Plugin Y supply line to the Automotive radar kit (1), plug in the power supply and attach the low voltage end of the power supply to the Y cable power line (5)
- Connect the board and the PC’s Ethernet Adapter with the straight Ethernet cable
- if power is good, three LED’s should be visible in this area (6)
- After 5 seconds the LED (7) starts flashing
- If the Ethernet link is up, LED (8) goes on

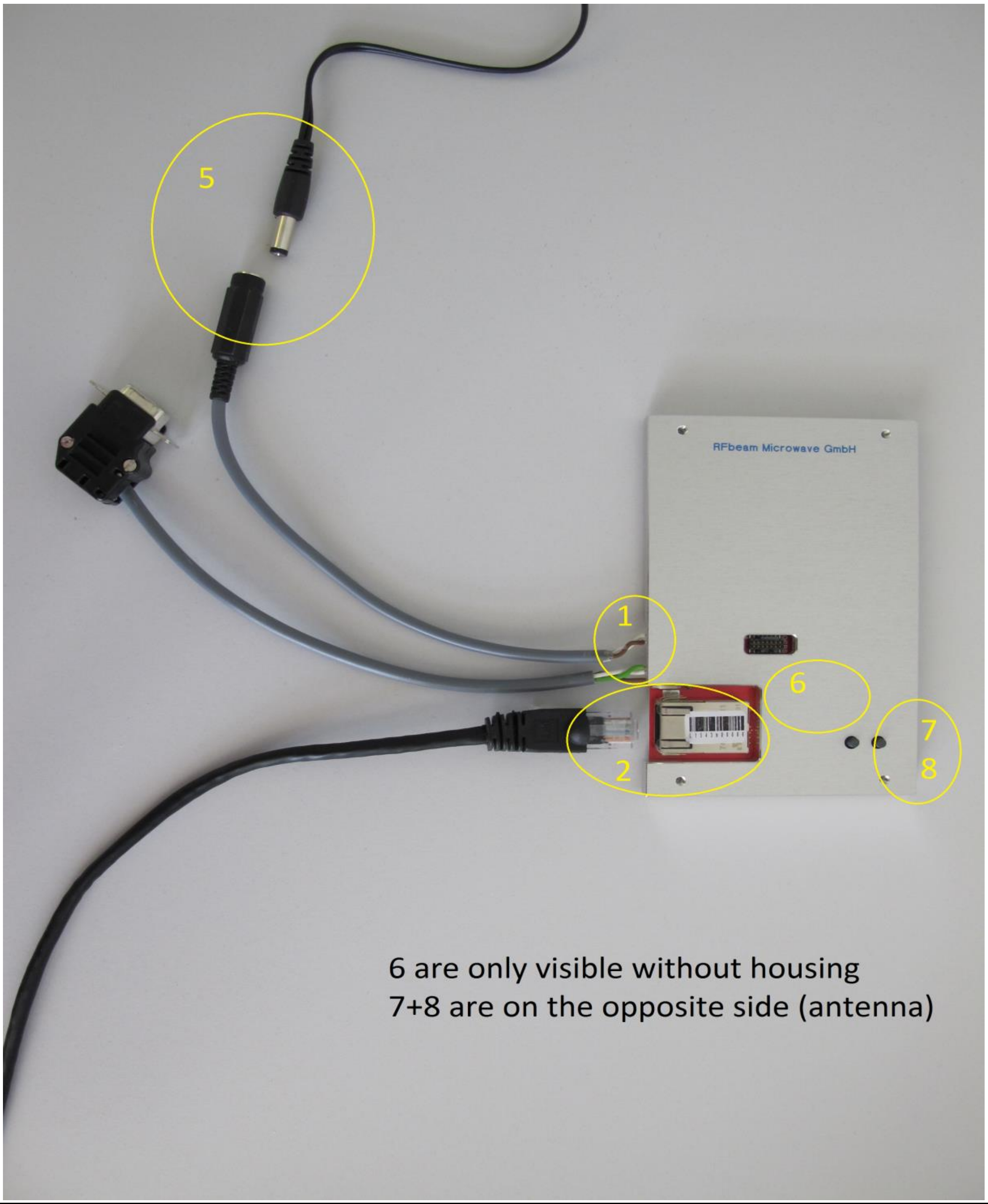

5
RFbeam Microwave GmbH
1
6
2
7
8
6 are only visible without housing
7+8 are on the opposite side (antenna)

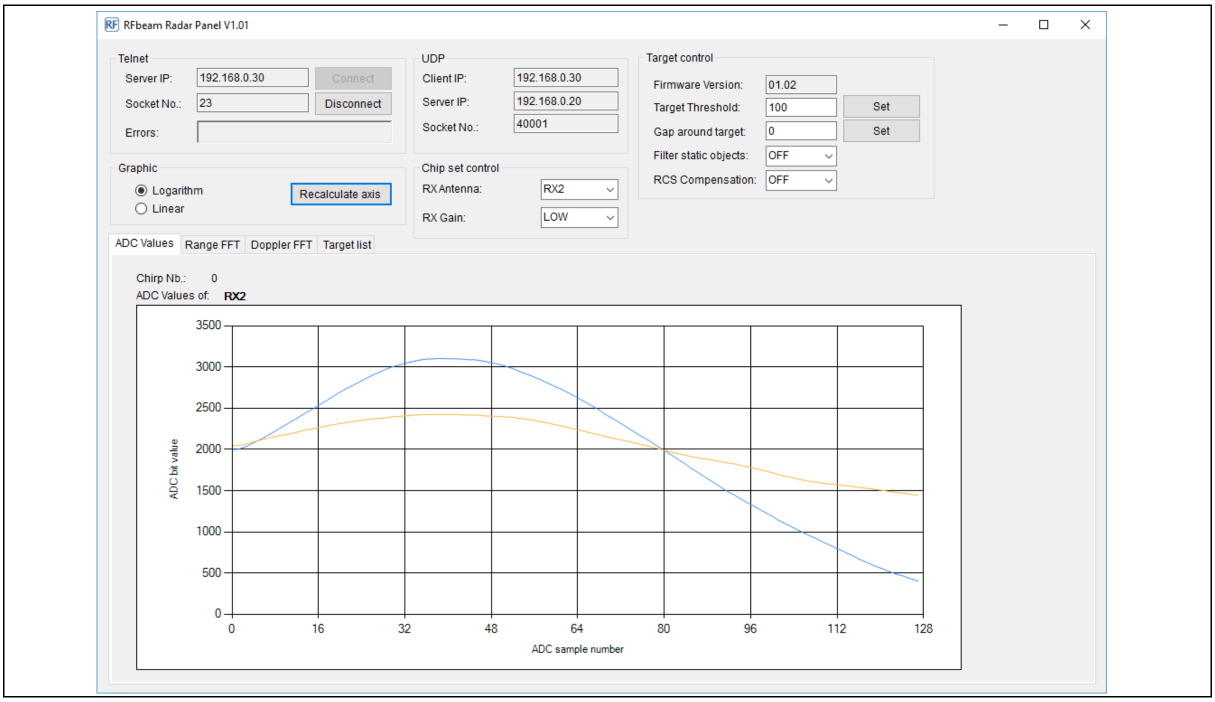
Figure 3 Power supply and connectors

1.2.3 Start PC Software

The installation of the "Radar Control Panel.exe" is very simple. Just extract the .exe from the archive (zip,rar,gz,7-zip) in a directory of your choice.
If the LED marked with 8 is on, start the Radar Control Panel.exe on the PC which is connected to the Automotive radar kit board by double clicking on it in Windows Explorer.
At the first start, windows will ask for access to public or private network for the application. Your network interface should be a private network (as descripted above), so use private.

Hardware und Software Setup Instruction

You will see a window like in the figure below. The curves displayed in the graphic area represent analog input values of the receive antennae (I and Q signals). If you move the radar, the curves will move too.
The fields with white background are changeable. The values have to be sent to the board by the set buttons. A change of the drop boxes is sent to the board immediately. As a simple test, a change of the RX antenna causes a well visible reaction.
If on your PC, UDP is blocked, no curves are visible. If TCP is blocked, the fields with white background will remain empty. If one of these happened, turn off the Windows firewall. Close the Radar Control Panel and restart the Software. If both TCP and UDP are working correctly, the fields are filled up at start up and curves are visible.



RFbeam Radar Panel V1.01

Telnet
Server IP: 192.168.0.30
Socket No.: 23
Errors:

UDP
Client IP: 192.168.0.30
Server IP: 192.168.0.20
Socket No.: 40001

Target control
Firmware Version: 01.02
Target Threshold: 100
Gap around target: 0
Filter static objects: OFF
RCS Compensation: OFF

Graphic
● Logarithm
○ Linear

Chip set control
RX Antenna: RX2
RX Gain: LOW

Chirp Nb.: 0
ADC Values of: RX2

Figure 4 **Radar Control Panel**

Now your hardware is working correctly. The network interface is up and running. TCP packages are sent between Automotive radar kit and PC. The PC receives the UDP packages from the board.
See next chapter for the installation of the development environment: software framework with eclipse and mini-wiggler programming tool.

2 Installing eclipse framework, Demo Project and compiler

2.1 Which compiler shall I use?

The decision for the compiler suite you want to use, should be done first. If you already own a Aurix compiler, use this one. See how to setup the compiler folders later in this document.
Otherwise, 3rd party products are available from HighTec, Tasking, Wind River and other.
You can find free compiler tool chains on the Infineon Homepage.
Search for “Free TriCore Tools - Infineon Technologies” on the web. You will find an actual link.
For the free tool chains, typically a registration on the supplier’s homepage is necessary.
Followed by the delivering of an access key or licence file. Keep in mind this process can take several hours up to days.
Do things in parallel. Your Aurix is a dual core!

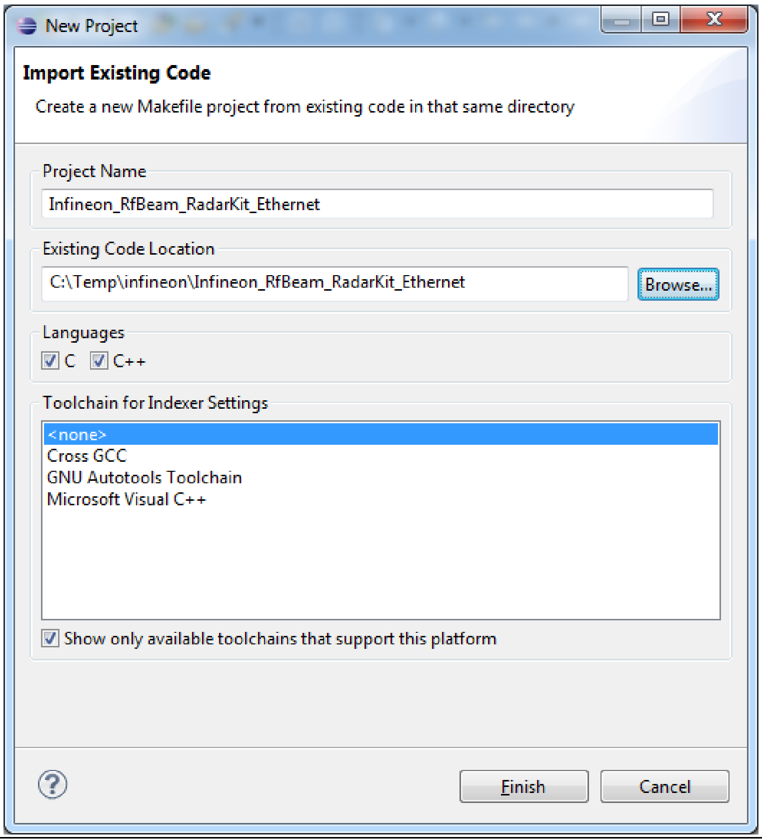
2.2 Installing Eclipse

For further assistance of installation of the eclipse framework, see the Infineon instruction manuals.
Download the “SWFramework_3v1r0.zip” (or newer) from Infineon Homepage. It is available for download in the Aurix download region. An extra registration is necessary. You need username/Password and you have to do a “login -> MyInfineon Login” on the Infineon Homepage to access these web space.
Extract the content to an empty folder and start the install.bat by double click.
The installation batch will first ask for an installation folder. Example is C:\SW_Framework_Tools, but take care, it’s not the default value, It is only an example! You have to retype the folder name! Do not simply press enter! The batch will not report an error and the installation will not succeed! So please type in the folder name C:\SW_Framework_Tools and press enter.
Answer the question for using eclipse by typing Y and enter.
The copy process will start. After a view seconds Windows could ask you to agree the start of BaseFrameworks_3v1r1.exe, depending on your security settings. Agree by pressing Yes.
Go on with Next and accept the licence agreements.
Next Question is the Aurix Workspace folder. Use the default value C:\AurixWorkspace.
Go on with Next and start the copy process with the Copy button.
When finished you can close the installation with the Finish button. The black DOS box in background remains open and wants you to press enter to quit. Click into the DOS box window and press enter, the DOS box will close.
You should find two additional folders in your root directory. The SW_Framework_Tools and the AurixWorkspace. The SW_Framwork_Tools contains a StartFW.bat. It has to be used for starting eclipse! Do not simply start eclipse.exe. The batch sets additional environment variables.
It is recommend to copy a shortcut for StartFW.bat on your desktop.
If you double click the StartFW.bat and a java error is reported, follow up with the installation of java runtime environment. You can find an actual installation package on www.java.com. Take care, remove all check boxes with addition advertisement software during the java installation!
We only want to get java, nothing else.

2.3 Importing the existing Demo Project

Now start eclipse the first time (use StartFW.bat!). Eclipse will ask for the workspace folder. Do not use the default value. Browse to C:\AurixWorkspace and activate checkbox for “use as default and never ask again”. Eclipse will start with a welcome window.

The button will bring you to your workspace. The BaseFramework Projects are already visible and all are open. Close them all. Right click on every BaseFramework… and choose “close Project”.
Next, copy the folder “Infineon_RFbeam_RadarKit_Ethernet” with the Demo project to C:\AurixWorkspace\
If the source is delivered in an archive like zip or rar, extract. Finally, the C:\AurixWorkspace should own a folder “Infineon_RFbeam_RadarKit_Ethernet” which contains the project sub structure. Avoid constructions like
C:\AurixWorkspace\Infineon_RFbeam_RadarKit_Ethernet\Infineon_RFbeam_RadarKit_Ethernet\...
In eclipse, follow the menu items
File -> New -> Makefile Project with Existing Code.
In “Exising Code Location” use Browse to locate the folder where you extracted the Demo Project.
Hopefully C:\AurixWorkspace\Infineon_RFbeam_RadarKit_Ethernet
The Project Name is derived from the last folder name. You can change it if you want.
During import, Eclipse does not move or copy the files into the AurixWorkspace folder. They remain, where they are. Eclipse does not care about their location. But, if you erase or move them outside of eclipse you will run into trouble.


New Project

Import Existing Code

Create a new Makefile project from existing code in that same directory

Project Name: Infineon_RfBeam_RadarKit_Ethernet

Existing Code Location: C:\Temp\infineon\Infineon_RfBeam_RadarKit_Ethernet

Languages: ☑ C ☑ C++

Toolchain for Indexer Settings:
- <none>
- Cross GCC
- GNU Autotools Toolchain
- Microsoft Visual C++

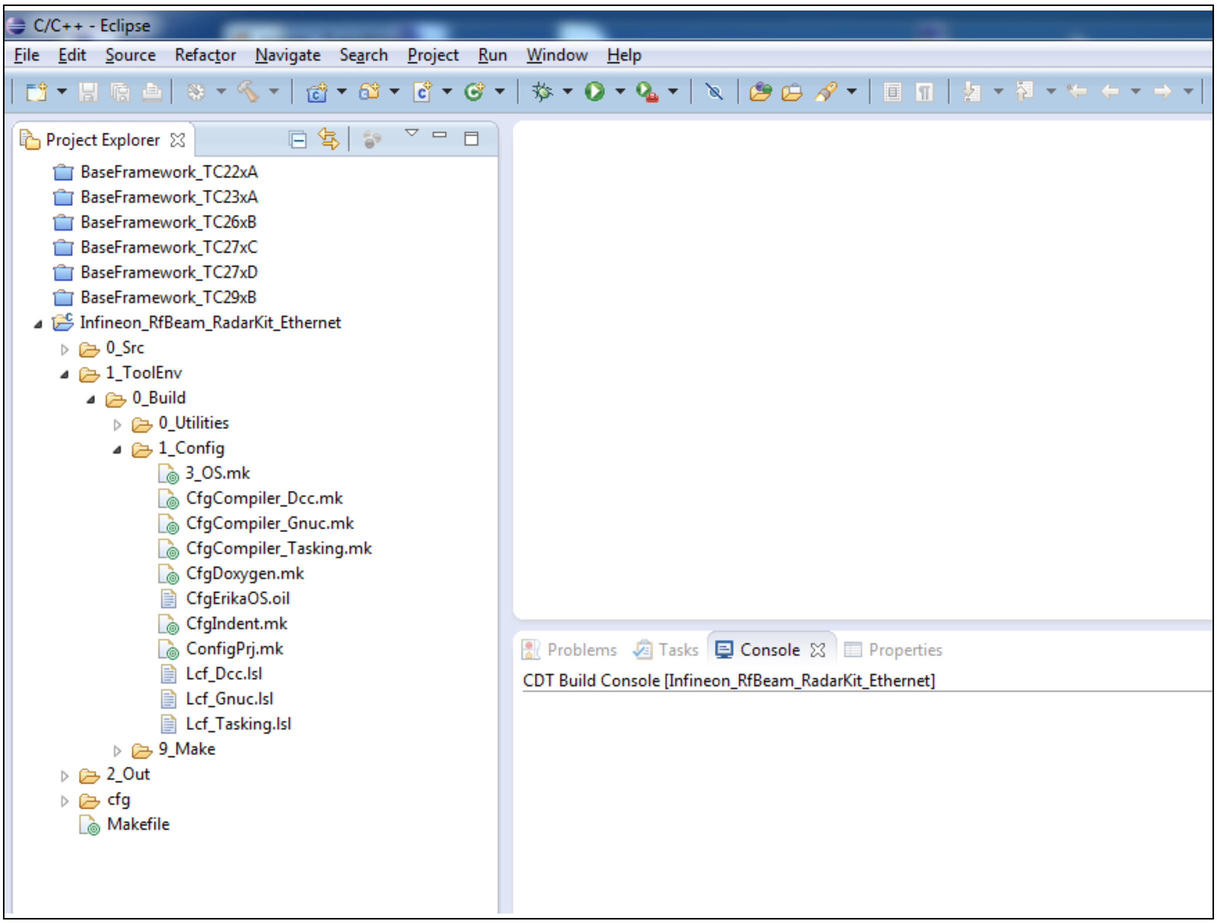
☑ Show only available toolchains that support this platform

Figure 5 Import existing Code

Press Finish to import the Project into your workspace.
Now the Demo Project is visible in eclipse. If no triangle left of the project name is visible, right click on the “Infineon_RFbeam_RadarKit_Ethernet” project entry and choose open Project.

2.4 Configuring Eclipse for your compiler

Click on the tringles left of the entries, to open the project tree. Navigate to the compiler configuration files. The project explorer should look like that.


C/C++ - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Problems Tasks Console Properties

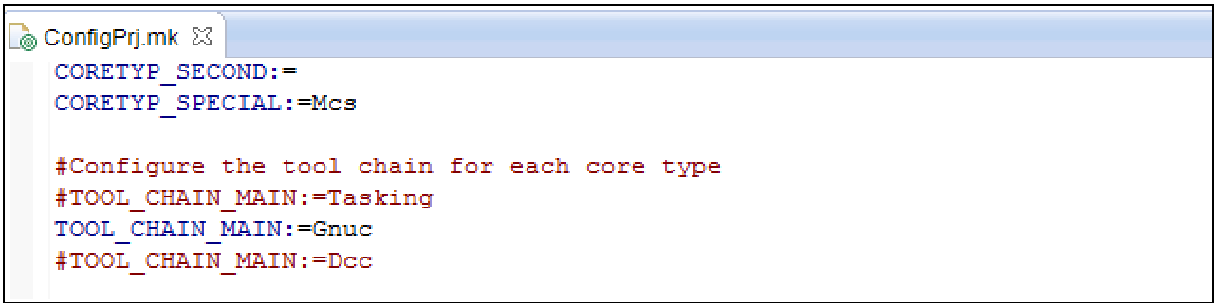
```
BaseFramework_TC22xA
BaseFramework_TC23xA
BaseFramework_TC26xB
BaseFramework_TC27xC
BaseFramework_TC27xD
BaseFramework_TC29xB
Infineon_RfBeam_RadarKit_Ethernet
  0_Src
  1_ToolEnv
    0_Build
      0_Utilities
      1_Config
        3_OS.mk
        CfgCompiler_Dcc.mk
        CfgCompiler_Gnuc.mk
        CfgCompiler_Tasking.mk
        CfgDoxygen.mk
        CfgErikaOS.oil
        CfgIndent.mk
        ConfigPrj.mk
        Lcf_Dcc.lsl
        Lcf_Gnuc.lsl
        Lcf_Tasking.lsl
      9_Make
  2_Out
  cfg
  Makefile
```

CDT Build Console [Infineon_RfBeam_RadarKit_Ethernet]

Figure 6 Project Tree

Depending on the installed compiler suite, you have to modify entries in ConfigPrj.mk and CfgCompiler_<mycompiler>.mk.

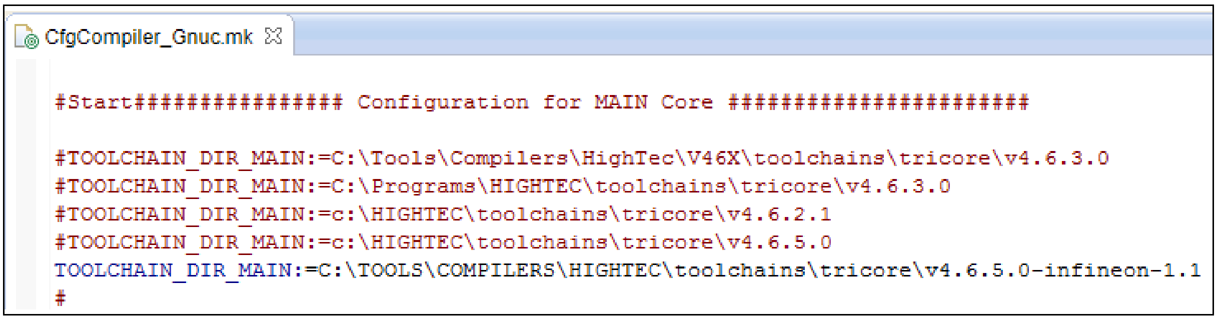
First, double click on the ConfigPrj.mk File. It will open in the right window. Three compilers are predefined, Tasking, Gnuc, Dcc. Remove the hashtag left of the entry to activate it. Only one is allowed. All other have to carry the hashtag. As an example, if you install the free HighTec toolchain, you have to activate Gnuc.



```
CORETYP_SECOND:=
CORETYP_SPECIAL:=Mcs

#Configure the tool chain for each core type
#TOOL_CHAIN_MAIN:=Tasking
TOOL_CHAIN_MAIN:=Gnuc
#TOOL_CHAIN_MAIN:=Dcc
```

Figure 7 Compiler settings

Save and close the modified file. Next, open the according CfgCompiler_< mycompiler >.mk. As an example, we open the CfgCompiler_Gnuc.mk.



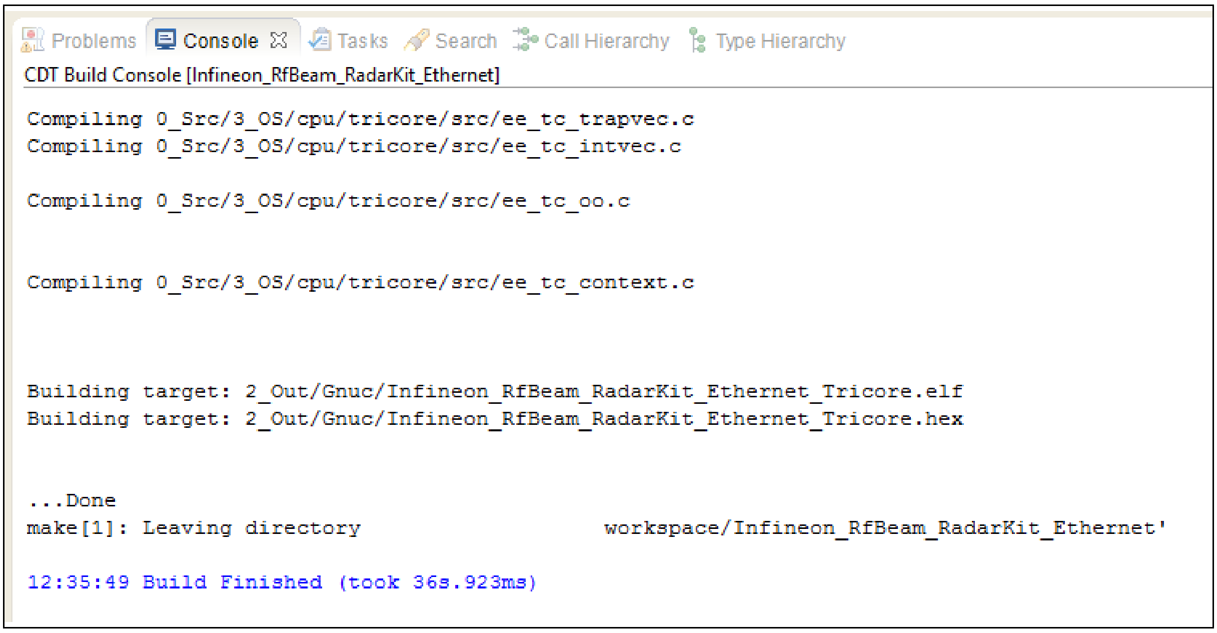
```
#Start############### Configuration for MAIN Core ######################

#TOOLCHAIN_DIR_MAIN:=C:\Tools\Compilers\HighTec\V46X\toolchains\tricore\v4.6.3.0
#TOOLCHAIN_DIR_MAIN:=C:\Programs\HIGHTEC\toolchains\tricore\v4.6.3.0
#TOOLCHAIN_DIR_MAIN:=c:\HIGHTEC\toolchains\tricore\v4.6.2.1
#TOOLCHAIN_DIR_MAIN:=c:\HIGHTEC\toolchains\tricore\v4.6.5.0
TOOLCHAIN_DIR_MAIN:=C:\TOOLS\COMPILERS\HIGHTEC\toolchains\tricore\v4.6.5.0-infineon-1.1
#
```

Figure 8 Toolchain folder path

Depending on the compiler version and install location, you have to modify the TOOLCHAIN_DIR_MAIN entry. So, eclipse is able to find the compiler on your computer. If you do not know or remember your installation location, search for the compiler exe file. As an example, the tricore-gcc.exe was located in the following folder
C:\TOOLS\COMPILERS\HIGHTEC\toolchains\tricore\v4.6.5.0-infineon-1.1\bin
Do not forget to save the CfgCompiler_< mycompiler >.mk file after modification. Unsaved modifications are ignored and eclipse does not save automatically by default.

2.5 First compilation run

Now you should be able to compile your project by pressing Ctrl-B, or by menu item
Projekt -> Build Project
or right click in the project explorer tree root entry “Infineon_RFbeam_RadarKit_Ethernet” and choose Build Project. Watch the reporting of the build process in the console window. If the compilation was successful, a .hex and .elf file is available.

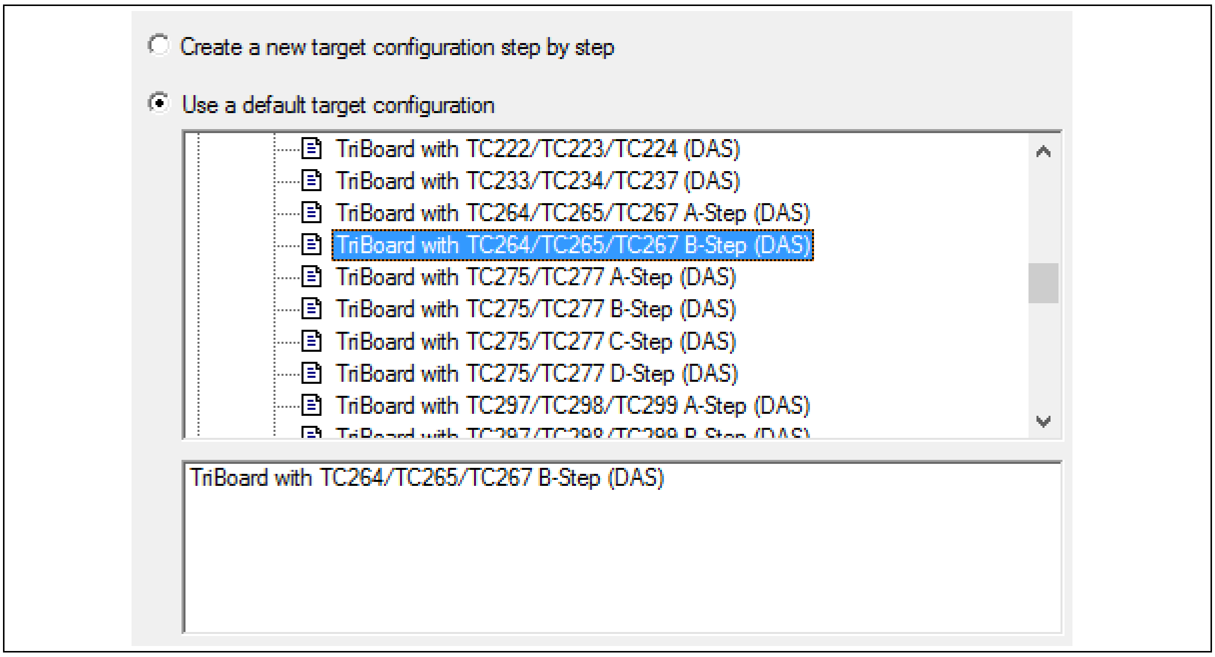


```
Compiling 0_Src/3_OS/cpu/tricore/src/ee_tc_trapvec.c
Compiling 0_Src/3_OS/cpu/tricore/src/ee_tc_intvec.c

Compiling 0_Src/3_OS/cpu/tricore/src/ee_tc_oo.c


Compiling 0_Src/3_OS/cpu/tricore/src/ee_tc_context.c



Building target: 2_Out/Gnuc/Infineon_RfBeam_RadarKit_Ethernet_Tricore.elf
Building target: 2_Out/Gnuc/Infineon_RfBeam_RadarKit_Ethernet_Tricore.hex


...Done
make[1]: Leaving directory                  workspace/Infineon_RfBeam_RadarKit_Ethernet'

12:35:49 Build Finished (took 36s.923ms)
```

Figure 9 First build result

2.6 Programming the Demo Board

Finally, we would like to program the code to the Demo Board. If you are equipped with a high level-debugging tool, see the documentation of your tool how to program elf, hex or binary files. A cost efficient way to bring code onto your Aurix is the Infineon “DAP miniWiggler” together with Infineon’s free of charge programming software MemTool. You can download MemTool from the Infineon homepage. Search for “Software Downloads - Infineon Technologies” on the web to find an actual download link. You can order the “DAP miniWiggler” from the Infineon Homepage.
Search for “DAP miniWiggler - Infineon Technologies” to get an actual link.
After downloading the MemTool installation File, double click the downloaded .exe and follow the installation instructions. The driver software for the DAP-miniWiggler will be installed, together with MemTool.
The mini-Wiggler is equipped with an USB connector. Connect it to your computer. The detection should by reported from windows by an acoustic signal. You can check the presence in the Device Manager of windows. Under USB-Controller you should find the “Infineon DAS JDS COM” and “Infineon DAS JDS DEBUG” or similar entries.
Start MemTool, you will be asked to choose the target configuration. Use the Default button to choose a default configuration.
Switch the radio button to “Use a default target configuration” and follow the tree list by clicking
⊞
TriCore -> StarterKits (DAS) -> TriBoard with TC264/TC265/TC267 B-Step (DAS) -> Finsih Button



○ Create a new target configuration step by step

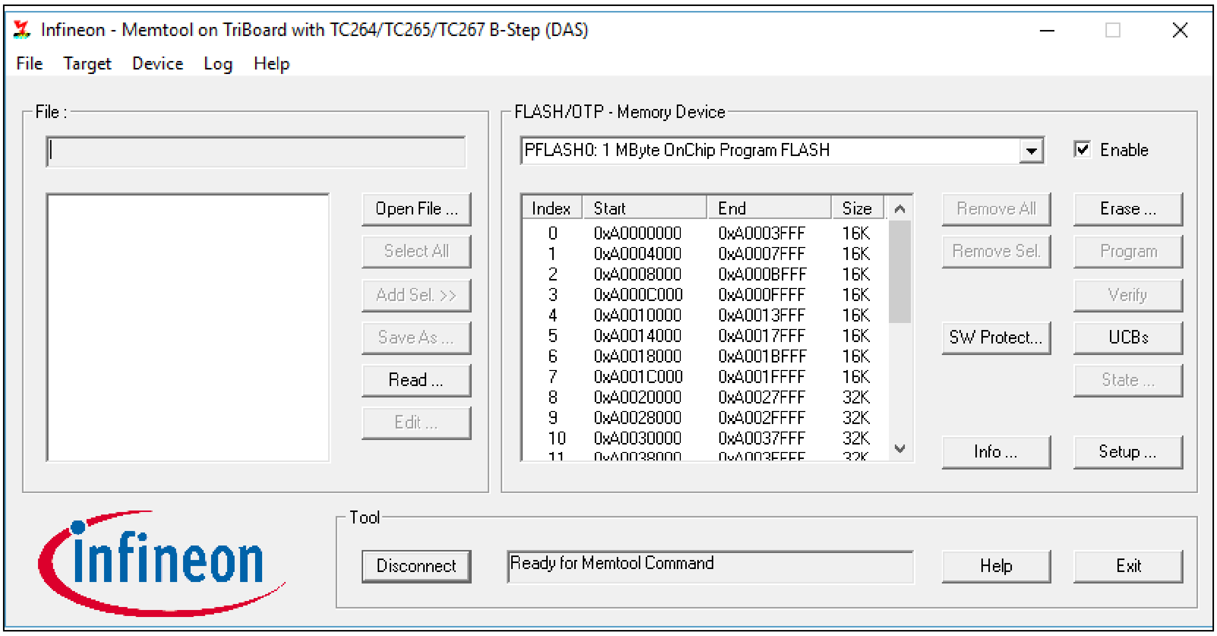
● Use a default target configuration

- TriBoard with TC222/TC223/TC224 (DAS)
- TriBoard with TC233/TC234/TC237 (DAS)
- TriBoard with TC264/TC265/TC267 A-Step (DAS)
- TriBoard with TC264/TC265/TC267 B-Step (DAS)
- TriBoard with TC275/TC277 A-Step (DAS)
- TriBoard with TC275/TC277 B-Step (DAS)
- TriBoard with TC275/TC277 C-Step (DAS)
- TriBoard with TC275/TC277 D-Step (DAS)
- TriBoard with TC297/TC298/TC299 A-Step (DAS)
- TriBoard with TC297/TC298/TC299 B-Step (DAS)

TriBoard with TC264/TC265/TC267 B-Step (DAS)

Figure 10 Memtool target configuration

Save the configuration.
Now MemTool is ready to connect. Plug in the mini-Wiggler on your Demo Board’s DAP2 connector.
Press Connect. It will take up to 10s to get connected. If you are successful, MemTool should report to be “Ready for MemTool Command”.



Infineon - Memtool on TriBoard with TC264/TC265/TC267 B-Step (DAS)

FLASH/OTP - Memory Device: PFLASH0: 1 MByte OnChip Program FLASH

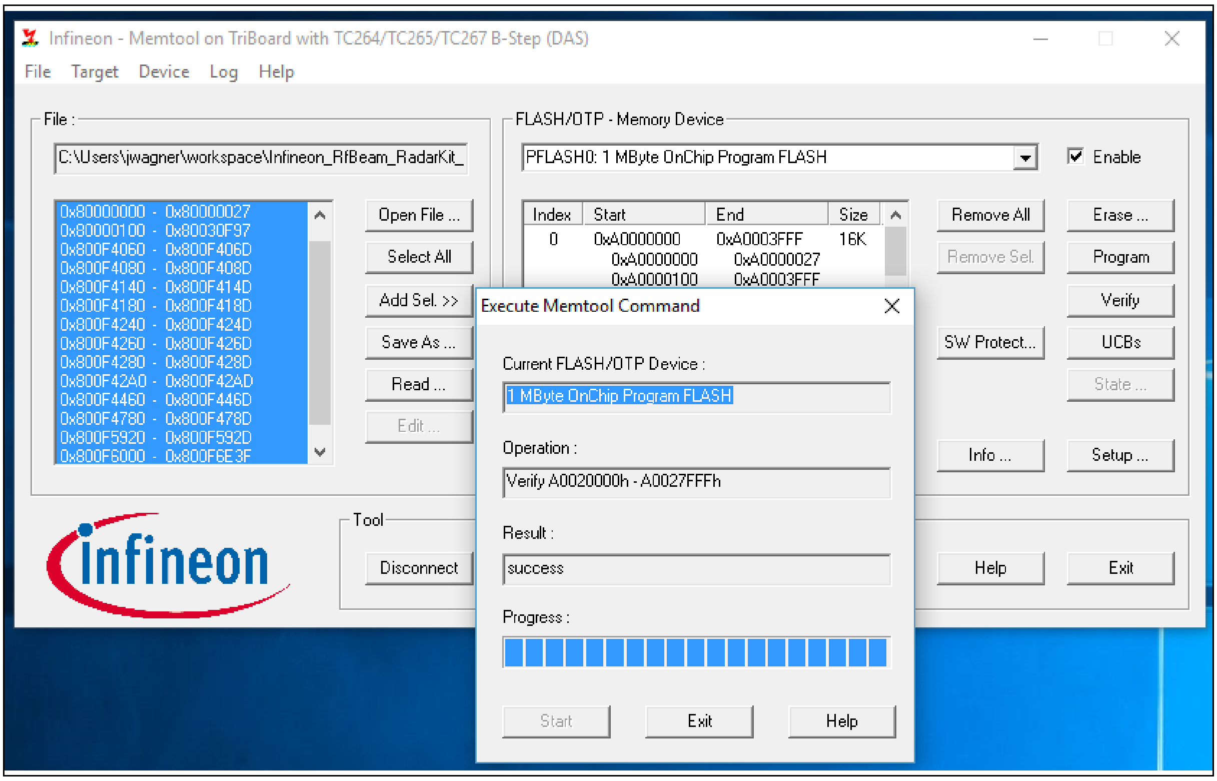
☑ Enable

Index	Start	End	Size
0	0xA0000000	0xA0003FFF	16K
1	0xA0004000	0xA0007FFF	16K
2	0xA0008000	0xA000BFFF	16K
3	0xA000C000	0xA000FFFF	16K
4	0xA0010000	0xA0013FFF	16K
5	0xA0014000	0xA0017FFF	16K
6	0xA0018000	0xA001BFFF	16K
7	0xA001C000	0xA001FFFF	16K
8	0xA0020000	0xA0027FFF	32K
9	0xA0028000	0xA002FFFF	32K
10	0xA0030000	0xA0037FFF	32K
11	0xA0038000	0xA003FFFF	32K

Ready for Memtool Command

Figure 11 Memtool started

By pressing the Open File button, you can load your hex file into Memtool. A list of memory ranges will appear. Press “Select All” and “Add Sel >>” to add all regions of the hex File to the areas of further operations. Press “Program” to download the binary data to the Aurix Flash memory.



Infineon - Memtool on TriBoard with TC264/TC265/TC267 B-Step (DAS)

File : C:\Users\jwagner\workspace\Infineon_RfBeam_RadarKit_

```
0x80000000 - 0x80000027
0x80000100 - 0x80030F97
0x800F4060 - 0x800F406D
0x800F4080 - 0x800F408D
0x800F4140 - 0x800F414D
0x800F4180 - 0x800F418D
0x800F4240 - 0x800F424D
0x800F4260 - 0x800F426D
0x800F4280 - 0x800F428D
0x800F42A0 - 0x800F42AD
0x800F4460 - 0x800F446D
0x800F4780 - 0x800F478D
0x800F5920 - 0x800F592D
0x800F6000 - 0x800F6E3F
```

FLASH/OTP - Memory Device: PFLASH0: 1 MByte OnChip Program FLASH

☑ Enable

Index	Start	End	Size
0	0xA0000000	0xA0003FFF	16K
	0xA0000000	0xA0000027	
	0xA0000100	0xA0003FFF	

Execute Memtool Command

Current FLASH/OTP Device : 1 MByte OnChip Program FLASH

Operation : Verify A0020000h - A0027FFFh

Result : success

Progress :

Figure 12 Memtool programing successfully done

Press Exit to close the programming window and the Reset Button on the Demo board to restart the Aurix. MemTool can remain connected or you can Disconnect and unplug the mini-Wiggler from the Demo Board.
It is recommended to plug and unplug the mini-Wiggler DAP2 connector with non-powered Demo board.
Now feel free to modify the Demo code in eclipse, save it (Ctrl-S), compile it (Ctrl-B), load the new hex to MemTool, select and add it, program it to the Demo board and finally restart the controller.

3 Software Stucture

The Automotive radar kit should demonstrate two basic functionalities of the Aurix TC264 microcontroller.

Part one is the use of FFT module together with VADC as signal source. Four on chip VADC Converters have to sample the prepared antenna signals simultaneously and deliver their data to the FFT unit.

Part two is the use of the on chip Ethernet MAC for data interchange between a host PC and the demo board.

3.1 Signal flow

To fullfill these requirments, the TC264 is supported by a 24 Ghz Transceiver chip which generates the HF signals for the transmission antenna. The Transceiver chip also prepares the two receive antenna signals.

On the output side, an Ethernet Transceiver PHY is used to generate the electrical signals for the Ethernet link (physical Layer).

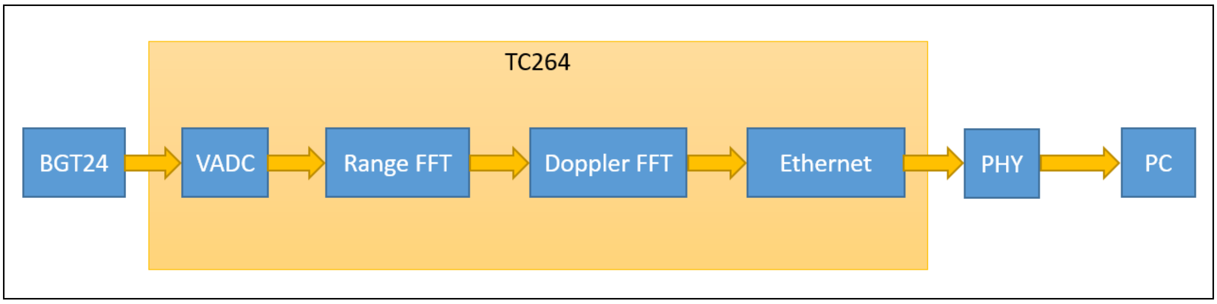


Figure 13 Signal flow through the Demo board

3.2 Operation System

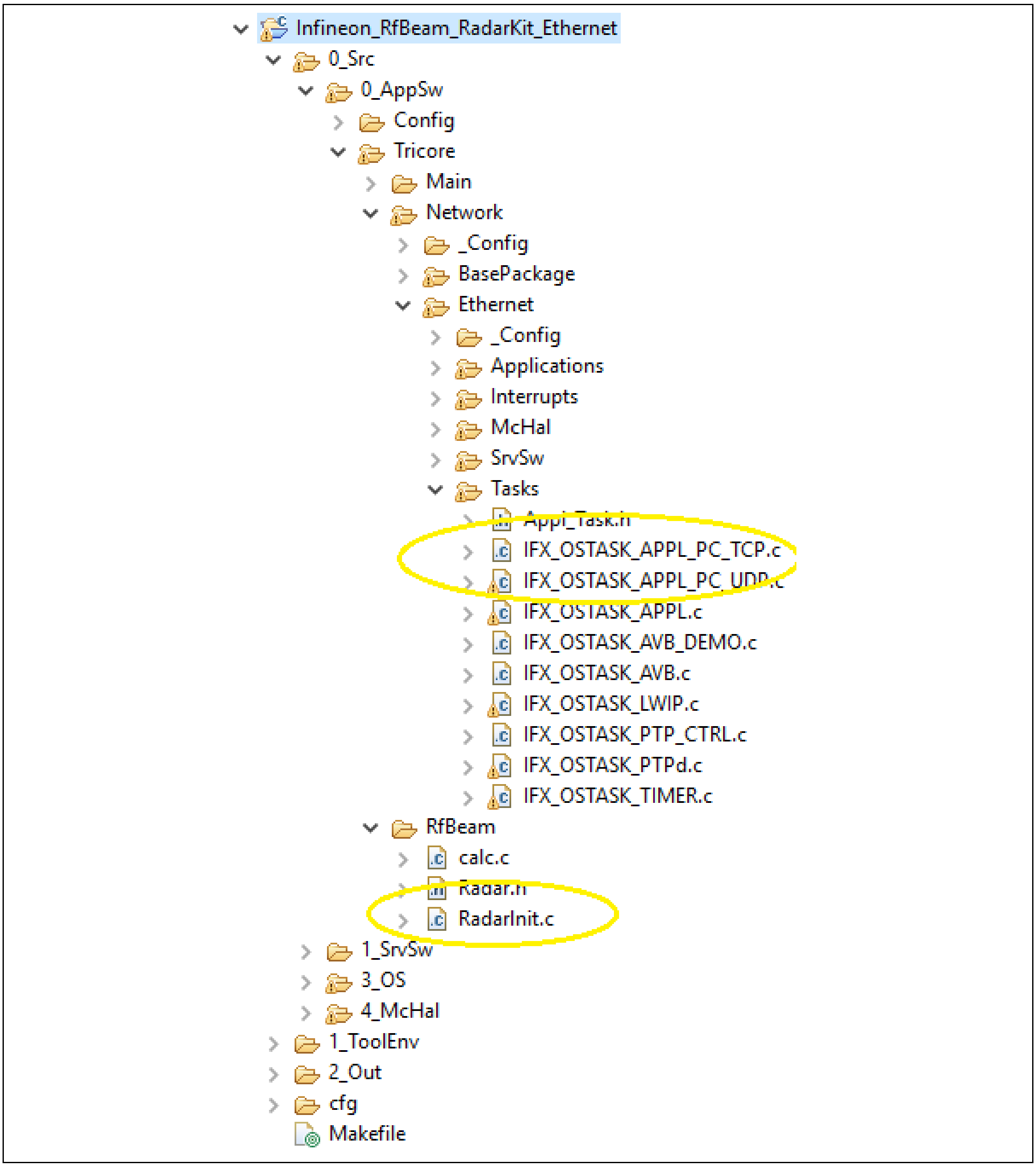
The Software of the Automotive radar kit is based on “Erika OS” Operating System.

See detailed information about Erika OS on the web. http://erika.tuxfamily.org

To offer TCP and UDP protocol on the Ethernet Interface, the free LWIP stack is used.

3.3 Project structure and core routines

To understand the core functionality of the Automotive radar kit, first take a look at the most important “routines”.



```
Infineon_RfBeam_RadarKit_Ethernet
  0_Src
    0_AppSw
      Config
      Tricore
        Main
        Network
          _Config
          BasePackage
          Ethernet
            _Config
            Applications
            Interrupts
            McHal
            SrvSw
            Tasks
              Appl_Task.h
              IFX_OSTASK_APPL_PC_TCP.c
              IFX_OSTASK_APPL_PC_UDP.c
              IFX_OSTASK_APPL.c
              IFX_OSTASK_AVB_DEMO.c
              IFX_OSTASK_AVB.c
              IFX_OSTASK_LWIP.c
              IFX_OSTASK_PTP_CTRL.c
              IFX_OSTASK_PTPd.c
              IFX_OSTASK_TIMER.c
        RfBeam
          calc.c
          Radar.h
          RadarInit.c
    1_SrvSw
    3_OS
    4_McHal
  1_ToolEnv
  2_Out
  cfg
  Makefile
```

Figure 14 Eclipse project tree

IFX_OSTASK_APPL_PC_UDP.c
Here you can find the Task, which prepares the data for the unidirectional traffic from the Automotive radar kit to the PC. Whenever a full set of data is available, UDP Task is triggered by an event.

Radarinit.c
Here you can find all the initialisation and service functions, which are radar specific.

RFbeam_initInterfaces();
is the summarisation of all the initialisation routines for the hardware units which are used be the Radar specific software components. CAN, SPI, DMA, VADC, CC60, FFT, IRQ, GTM.

Software Stucture

RFbeamDoppler();
Is called at the end of last sample, the range FFT's are already done in parallel to the sampling. Sampling is stopped and the Doppler FFT's will be done.
To perform the most important functions in parallel to the code execution, the DMA controller is used. The DMA channels are configured in groups and concatenated by daisy chaining.
See how the daisy chaining works, with the channel 40-36 group as an example.
General purpose Timer 12 is responsible for the sample clock. The Timer triggers DMA channel 40 with every overflow. DMA Channel 40 is responsible for the transfer of the next DAC value to the SPI modul. The DAC value controls the frequency of the BGT24. At the end of the DMA transfer, DMA channel 40 triggers the next lower DMA channel 39. Channel 39 is responsible for transfer ADC result of antenna 1 signal q to the ADC buffer. At the end, DMA channel 39 triggers DMA channel 38 and so on. Down to channel 36, which is the last channel in this daisy chain. Channel 36 transfers the ADC result of antenna 2 signal I to the ADC buffer. With channel 36, the activities end.

These are all the daisy chained DMA channels and their functionality.

DMA 45 – 42:	handle the end of a sweep
DMA 40 – 36:	set DAC output for next frequency and transfer last ADC samples into buffer
DMA 30 – 22:	transfer buffered ADC values to FFT and result back to range FFT result buffer
DMA 20 – 11:	transfer range FFT result data to FFT and result back to Doppler FFT result buffer

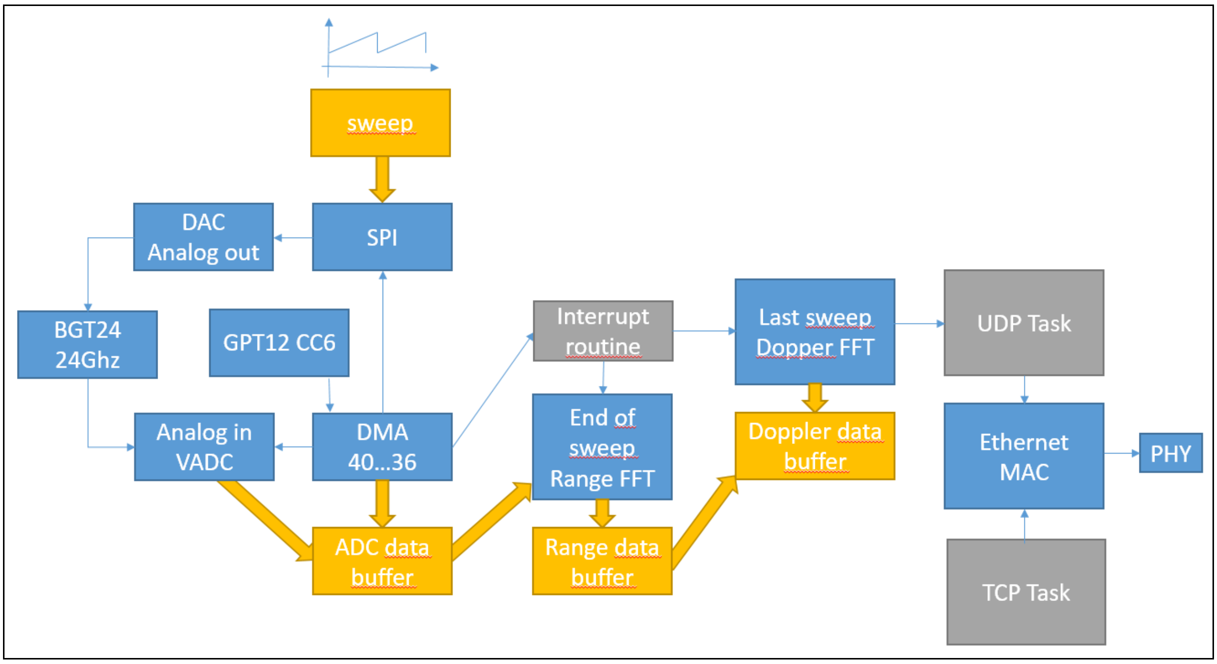
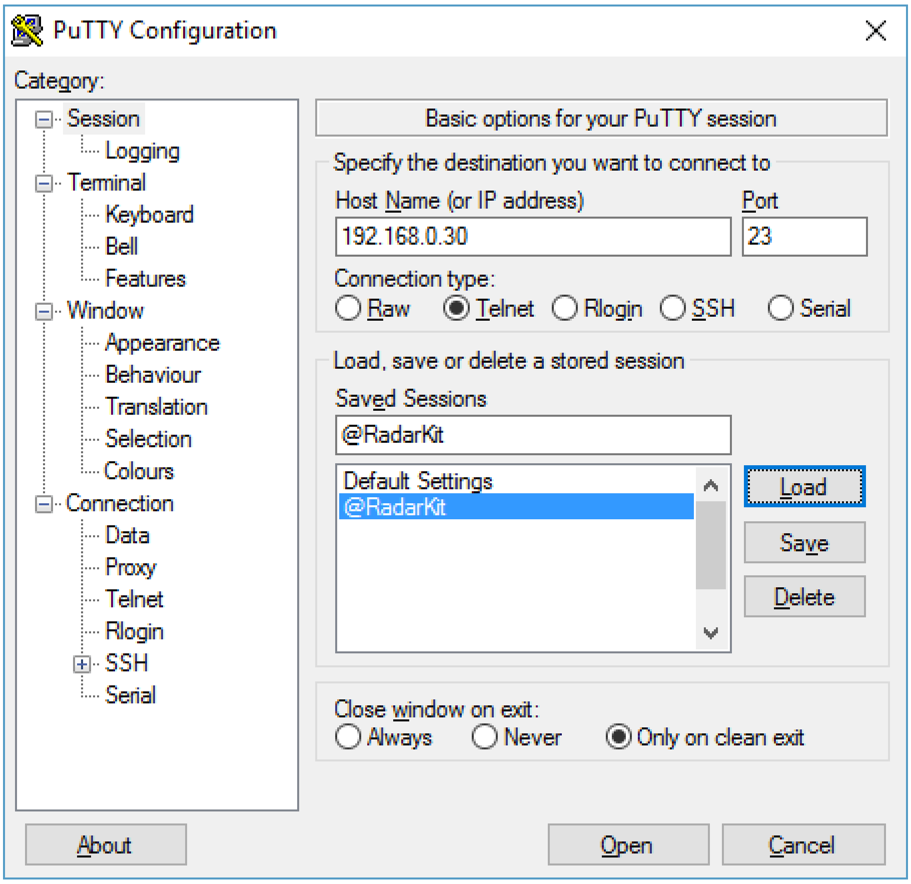


Figure 15 Radar relevant software components.

3.4 Network client Protocol

The TCP Task receives simple text commands via Ethernet interface. You can use a telnet client like putty for direct communication without the Radar panel software. The following values should be used when using putty as telnet client.



Specify the destination you want to connect to

Host Name (or IP address): 192.168.0.30

Port: 23

Connection type: ○ Raw ● Telnet ○ Rlogin ○ SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: @RadarKit

Default Settings
@RadarKit

Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 16 Telnet configruation

The following text Commands are supported:

Command	Command	Remark	read	write
01	UDP client IP	IP Address of the UDP client	Yes	No
02	UDP Port		Yes	No
03	UDP data output mode	valid out modes are 0 = Target List 1 = Range FFT 2 = doppler FFT 3 = ADC raw data	Yes	Yes
04	RCS Target threshold	Threshold of RCS value, Targets with bigger RCS than threshold will be detected	Yes	Yes
05	Rx channel	select channel number for CAN transmission	Yes	Yes
06	Target Gap	Gap around a detected target where all lower targets are deleted	Yes	Yes
11	Firmware Version	Version = Major*100 + Minor	Yes	No
12	Receive antenna	antenna 0 or 1	Yes	Yes
13	Filter static opjects	0=disabled, 1=enabled	Yes	Yes
16	RCS compensation	0=disabled, 1=enabled	Yes	Yes

Command 03 controls the type of output data on the UDP channel.
The UDP data is sent to the PC in the following format:

Software Stucture

Target List packet:

Byte Address	Description
0	Identifier for Target list packet 0x1973
2	32 bit sequence number
6	Number of detected targets
8	target 0 distance
10	target 0 speed
12	target 0 angle
14	target 0 RCS
	…
256	target 31 distance
258	target 31speed
260	target 31 angle
262	target 31 RCS

Range FFT packet:

Byte Address	Description
0	Identifer for Range FFT packet 0x1984
2	chirp number
3	Sample 0: real componente of FFT
5	Sample 0: imaginare componente of FFT
	…
511	Sample 127: real componente of FFT
513	Sample 127 imaginare componente of FFT

Doppler FFT packet:

Byte Address	Description
0	Identifer for Range FFT packet 0x1995
2	chirp number
3	chirp 0: real componente of FFT
5	chirp 0: imaginare componente of FFT
	…
511	chirp 127: real componente of FFT
513	chirp 127 imaginare componente of FFT

ADC raw data packet:

Byte Address	Description
0	Identifer for ADC raw packet 0x2014
2	not used, 0
3	sample 0: Q channel
5	sample 0: I channel
	…
511	sample 127: Q channel
513	sample 127: I channel

3.5 Software Flowchart

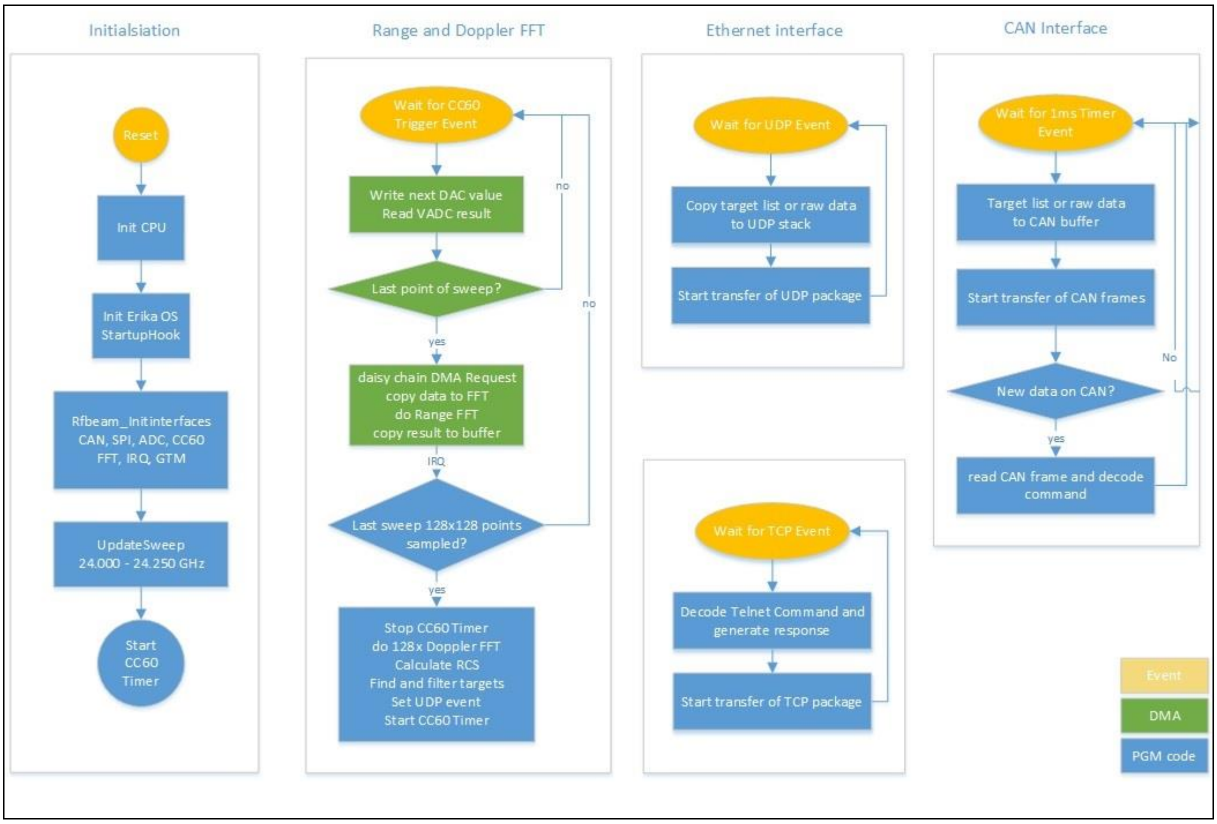

Initialsiation
Range and Doppler FFT
Ethernet interface
CAN Interface
Reset
Init CPU
Init Erika OS StartupHook
Rfbeam_Initinterfaces CAN, SPI, ADC, CC60 FFT, IRQ, GTM
UpdateSweep 24.000 - 24.250 GHz
Start CC60 Timer
Wait for CC60 Trigger Event
Write next DAC value Read VADC result
Last point of sweep?
no
yes
daisy chain DMA Request copy data to FFT do Range FFT copy result to buffer
IRQ
Last sweep 128x128 points sampled?
no
yes
Stop CC60 Timer do 128x Doppler FFT Calculate RCS Find and filter targets Set UDP event Start CC60 Timer
Wait for UDP Event
Copy target list or raw data to UDP stack
Start transfer of UDP package
Wait for TCP Event
Decode Telnet Command and generate response
Start transfer of TCP package
Wait for 1ms Timer Event
Target list or raw data to CAN buffer
Start transfer of CAN frames
No
New data on CAN?
yes
read CAN frame and decode command
Event
DMA
PGM code

Figure 17 Visio Chart

3.6 Routines in detail

3.6.1 Initialization Functions:

3.6.1.1 Ifx_OSTask_initFFT

Initialize the FFT unit to perform the 128 dot range FFT, sweep by sweep in parallel to the VADC sampling of the following sweep.
Ifx_init_LED_Monitor_Port
Initialize the IO ports to operate the LED's (Ethernet link up and UDP packet transmit).
User LED 1 is connected to P00.2 and indicates a UDP packet transmission (in code LED2_TOGGLE !)
User LED 2 is connected to P00.1 and indicates an Ethernet Link up (in code LED1_ON !)
P23.1 is the measurement input for the BGT24 frequency

3.6.1.2 Ifx_OSTask_initCAN

Initialize the CAN Controller for sending the target list or VADC samples via CAN bus in parallel to the data on the Ethernet Interface.

3.6.1.3 Ifx_OSTask_initDMA

Initialize the DMA Controller and configure the channels. Several channels are concatenated to groups by the daisy chain mechanism to realize complex sequences.

3.6.1.4 Ifx_OSTask_initADC

Initialize the VADC for synchronized conversion of the 4 antenna signals and deliver the values to the result registers.

3.6.1.5 Ifx_OSTask_initIRQ

Initialize the IRQ arbiter to service requests for VADC results, DMA Channels, Timer 13 period and FFT result available.

3.6.1.6 Ifc_OSTask_initCC60

Initialize the Timer 13 and couple CAPCOM60 with the VADC to generate the sample clock.

3.6.1.7 FFT_Window

Updates the FFT Window Memory area with new values. Hanning, Blackman and Harris are available.

3.6.1.8 AdjustDAC

Adjusts the DAC output to realise the expected radiation frequency. If channel 1 is requested, channel 2 is adjusted first to 50% of the DAC output. Only channel 1 tuned to reach the requested value.
If channel 2 is requested, channel 1 remains on it's last setting. Only channel 2 is tuned.

3.6.1.9 UpdateSweep

Determines the DAC output for the lowest and highest output frequency. Then the 128 sweep values are linear interpolated.

3.6.2 CAN Interface related functions:

3.6.2.1 RFbeam_CANservice

Fetches the data for the next CAN Frame and triggers the transmission. The CAN receive channel is polled and available data is processed.

3.6.2.2 CANsend

Send one CAN Frame with fixed length of 8 byte. The function is non-blocking. If the transmission queue remains full, the routine returns after the timeout expired.

3.6.2.3 CANreceive

Wait for CAN data until timeout expires. Return data if a CAN frame was received.

3.6.3 Network related Functions:

Two Tasks the IFX_OSTASK_APPL_PC_TCP and IFX_OSTASK_APPL_PC_UDP are responsible for the Ethernet Traffic. The TCP Task services receive and transmission of TCP packages. The UDP Task is only responsible for data transmission to the host.
An incoming TCP package is passed over to the RFbeamTelnetDecode function. Function call is done directly in the Task IFX_OSTASK_APPL_PC_TCP.

3.6.4 FFT related Functions:

The TC264 is equipped with a hardware FFT module. The module is able to calculate real and complex FFT's up to 1024 dots in parallel to the normal CPU operation. To offer maximum performance the FFT unit has an own physical data RAM range. Write and read operations are done to buffers (no direct access to the FFT data and result buffers). The input data buffer can be fed with new data in parallel to a running FFT transformation. At the end of the running transformation the next transformation starts immediately (pipelined FFT operation).
Figure 18 shows the activities of both DMA channels which service the FFT module. DMA32 is used to transfer the data to the FFT module. DMA34 is responsible to return the FFT result data to RAM. The Figure shows the 100% busy DMA32, feeding the FFT unit with new data.

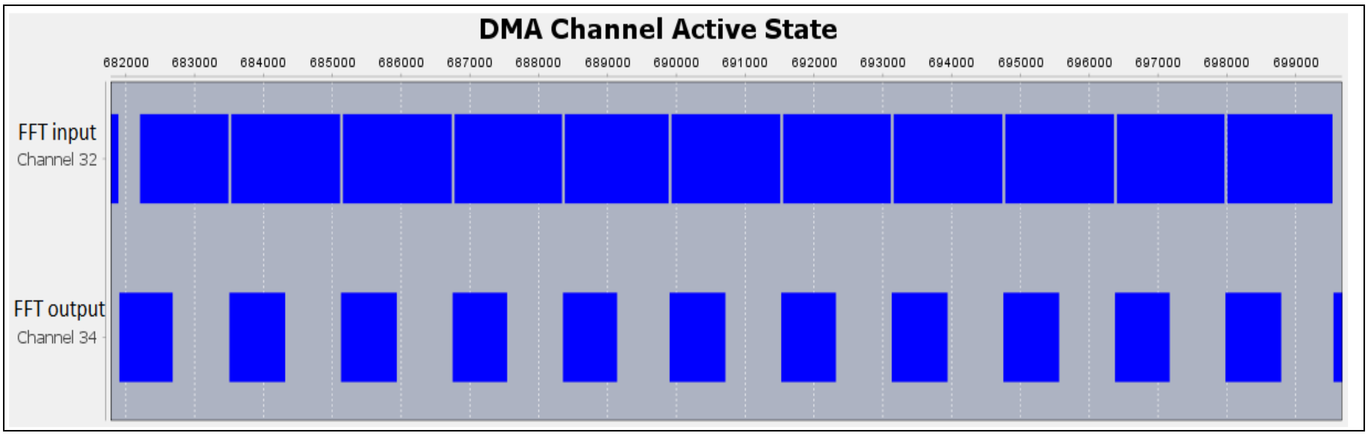


Figure 18 DMA activities during pipelined FFT operation

Only the Doppler FFT is done pipelined. The range FFT is done in parallel to the sampling of the VADC values.
The following functions are used to trigger the Doppler FFT and to calculate the target list.RFbeamDoppler
Prepare the range FFT buffer and force the calculation of the 128 Doppler FFT's.

3.6.4.1 RFbeam_GeoSum

The result of the dopper FFT is a complex number, duple of two 16 bit signed values (re, im). The routine calculats the square (re)+square(im) without root. The whole range FFT buffer is operated (128 FFT's with 128 complex dots per FFT). The value represents the echo amplitude of a target.

3.6.4.2 calc_angle

the angle value is calculated as the difference between the phase of antenna 1 and antenna 2.

3.6.4.3 calc_find_peaks

The function searches the range FFT for echo amplitudes higher than a threshold. In an configurable area around a valid target, only the highest target is taken. All other targets are removed to avoid double targets.

3.6.4.4 calc_find_targets

This routine only picks out all the remaining entry's after calc_find_peaks has cleaned up the area. The remaining peaks are collected into a target list. Angle, speed and distance are calculated. If configured, static objects are ignored. The distance is proportional to the location of the peak in range FFT. Speed is proportional to the peak location in the Doppler FFT. The angle is the difference In phase between antenna 1 and antenna 2.

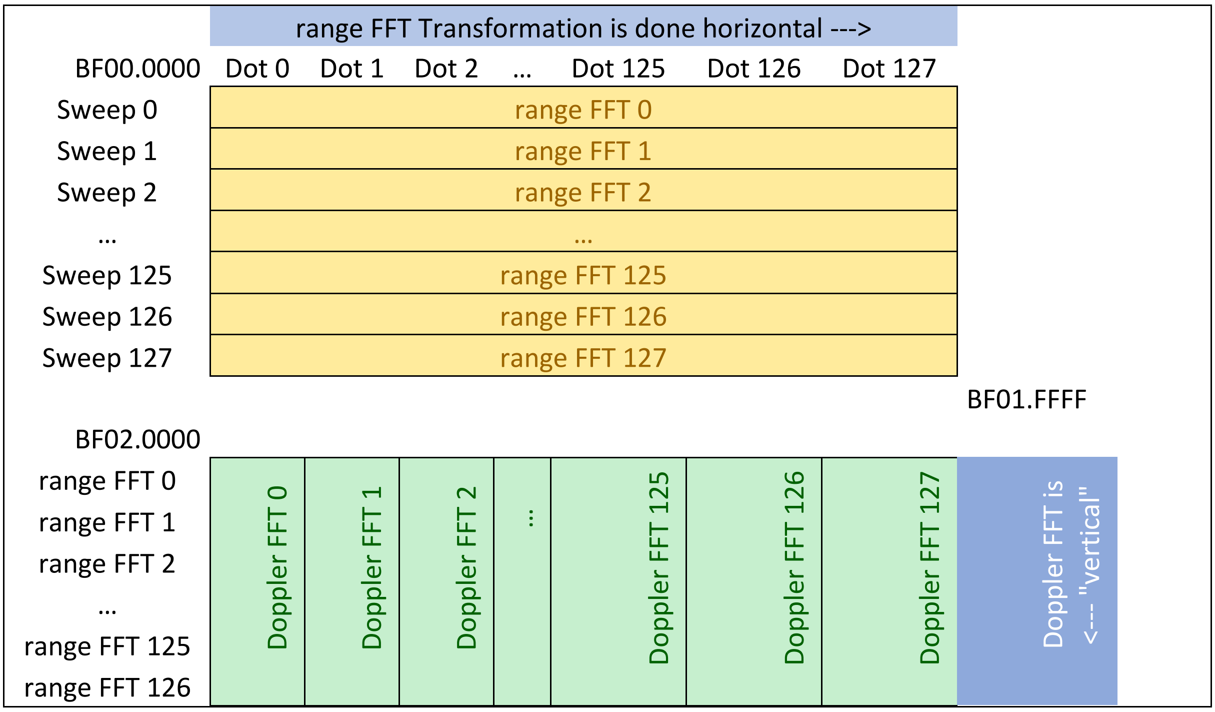
3.6.5 Memory Map and Emulation Memory EMEM

The TC264 Devices are equipped with a 512k Emulation Memory. The RAM is located at the address range 0xBF00.0000 up to 0xBF07.FFFF. The EMEM is not available after reset. It has to be enabled by an unlock sequence in function Ifx_OSTask_initFFT.
All data, raw VADC sample values, range FFT and Doppler FFT results are available in EMEM RAM at any time in parallel. No result values are overwritten. Every sampled dot covers a 32 bit value and consists of two 16bit signed values. Representing I and Q channel of one antenna for the FFT input or real/complex FFT output result values..

Table 1

Information	Location in EMMEM	Length	Schema
VADC raw data ant 1 / I	BF00.0000 ...4...8...C...10	256 byte * 128 dots = 32k	2 byte signed
VADC raw data ant 1 / Q	BF00.0002 ...6...A...E...12	256 byte * 128 dots = 32k	2 byte signed
VADC raw data ant 2 / I	BF01.0000 ...4...8...C...10	256 byte * 128 dots = 32k	2 byte signed
VADC raw data ant 2 / Q	BF01.0002 ...6...A...E...12	256 byte * 128 dots = 32k	2 byte signed
Range FFT real	BF02.0000 ...4...8...C...10	256 byte * 128 dots = 32k	2 byte signed
Range FFT complex	BF02.0002 ...6...A...E...12	256 byte * 128 dots = 32k	2 byte signed
Doppler FFT real	BF04.0000 ...4...8...C...10	256 byte * 128 dots = 32k	2 byte signed
Doppler FFT complex	BF04.0002 ...6...A...E...12	256 byte * 128 dots = 32k	2 byte signed
target amplitude	BF06.0000...B006.FFFF	uint32	Uint32

See the mapping in memory as lines and columns. The range FFT is done "horizontal" trough the memory. To perform the Doppler FFT the data in memory has to be flipped (done by DMA during transfer into FFT buffer). It seems like the Doppler FFT is done "vertical" trought the range FFT data in memory.



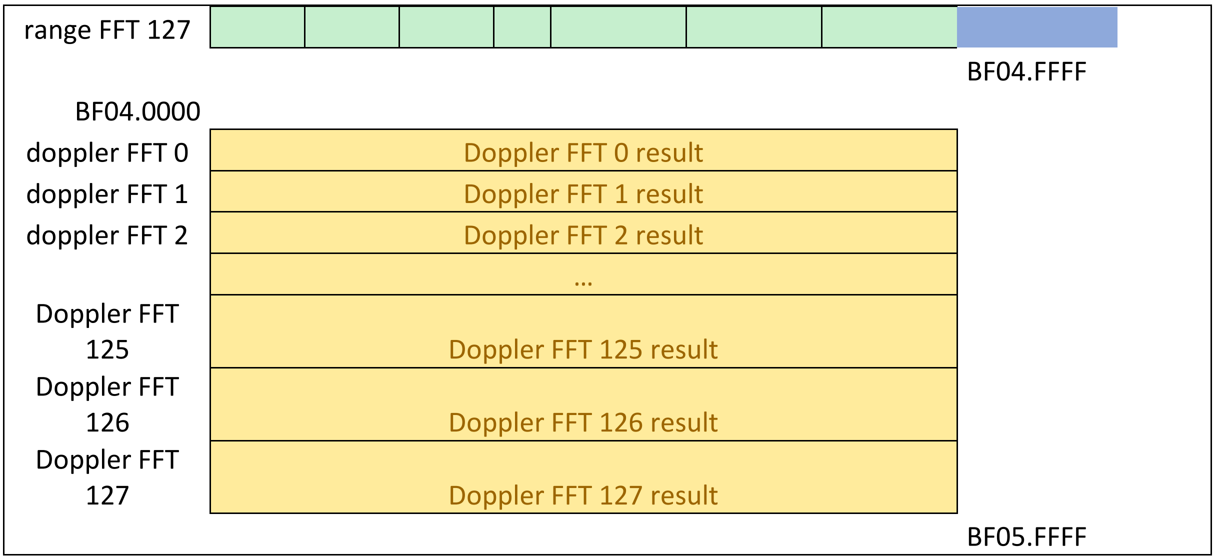


Figure 19 Range and Doppler FFT “horizontal” and “vertical”

3.6.6 DMA Channels

DMA channels do most of the background work for the FFT. The following table should give an overview about the channel usage.

Table 2

channel	daisy	from	to	from	to	remark
47						
46						
45	yes	cmd_entr	VADC_G0QMR0	&cmd_entr	F0020504	disable ADC conversion by CCU60
44	yes	cmd_ccu60_1	SRC_CCU60SR3	&cmd_ccu60_1	F003842F	clear request
43	no	cmd_ccu60_2	SRC_CCU60SR3	&cmd_ccu60_2	F003842D	enable IRQ service routine by CCU60
42	no	cmd_fft_h	DMA_CHCSR30	&cmd_fft_h	F00123DF	start horizontal fft by DMA30
41						
40	no	ADR_SWEEP	QSPI1_TXBUF	sweep signal	qspi1.TX	load DAC with new sweep amplituede
39	yes	ADCCH_3	RAW_CH3	analog result 3	BF01.0002	transfer VADC CH3 to buffer
38	yes	ADCCH_2	RAW_CH2	analog result 2	BF01.0000	transfer VADC CH2 to buffer
37	yes	ADCCH_1	RAW_CH1	analog result 1	BF00.0002	transfer VADC CH1 to buffer
36	yes	ADCCH_0	RAW_CH0	analog result 0	BF00.0000	transfer VADC CH0 to buffer
35						
34	no	FFT data	FFT_DOPPLER_0	BE00.0000	BF04.0000	transfer FFT result real/complex to result buffer
33	yes	cmd_fft	FFT_NEXT	0x80	F003.8FC0	prepare FFT for next transformation
32	no	FFT_DOPPLER_0	FFT data	BF04.0000	BE00.0000	transfer data to FFT
31	no	FFT_RANGE_0	FFT_DOPPLER_0	BF02.0000	BF04.0000	flip data from vertical to horizontal temp buffer
30	no	RAW_CH0	FFT data	BF00.0000	BE00.0000	transfer VADC CH0/1 to FFT
29	yes	FFT data	FFT_RANGE_0	BE00.0000	BF02.0000	transfer FFT result real/complex to buffer
28	yes	cmd_dma2	FFT_DONE	24	F003.8FC0	reconfigure FFT IRQ to DMA CH24
27	yes	cmd_fft	FFT_NEXT	0x80	F870.0C40	prepare FFT for next transformation
26	no	RAW_CH2	FFT data	BF01.0000	BE00.0000	transfer ADC CH2/3 to FFT
25	yes	FFT data	FFT_RANGE_1	BE00.0000	BF03.0000	transfer FFT result real/complex to buffer

24	yes	cmd_dma1	FFT_DONE	29	F003.8FC0	reconfigure FFT IRQ to DMA CH29
23	no	cmd_fft	FFT_NEXT	0x80	F870.0C40	prepare FFT for next transformation
22						
21						
20	yes	FFT_RANGE_0	temp RANGE	BF02.0000	6010.0000	flip data from vertical to horizontal temp buffer
19	no	temp RANGE	FFT data	6010.0000	BE00.0000	transfer data from temp to FFT
18	yes	FFT data	FFT_DOPPLER_0	BE00.0000	BF04.0000	transfer FFT result real/complex to buffer
17	yes	cmd_dma4	FFT_DONE	13	F003.8FC0	reconfigure FFT IRQ to DMA CH13
16	yes	cmd_fft	FFT_NEXT	0x80	F870.0C40	prepare FFT for next transformation
15	yes	FFT_RANGE_1	temp RANGE	BF03.0000	6010.0000	flip data from vertical to horizontal temp buffer
14	no	temp RANGE	FFT data	6010.0000	BE00.0000	transfer data from temp to FFT
13	yes	FFT data	FFT_DOPPLER_1	BE00.0000	BF05.0000	transfer FFT result real/complex to buffer
12	yes	cmd_dma3	FFT_DONE	18	F003.8FC0	reconfigure FFT IRQ to DMA CH18
11	no	cmd_fft	FFT_NEXT	0x80	F870.0C40	prepare FFT for next transformation
10						
9						
8						
7	no					Ethernet Controller, LWIP Stack
6						
5						
4						
3						
2						
1						
0	no	A008.0000	BE10.0000		FFT window data	transfer window values to FFT window buffer

3.6.7 Sigma-Delta DAC on Revision-B Board

The dedicated Hardware DAC of the Revision-A Boards (also available on the Revision-B Boards) can be replaced by a simple OR gate hardware and a clever usage of the GTM Module. The procedure is known from AD conversion as the sigma-delta-conversion. Major part of these ADC converts is a Delta Sigma Modulator to produces a bitstream which is low pass filtered. The output of the low pass is back coupled to the input signal.

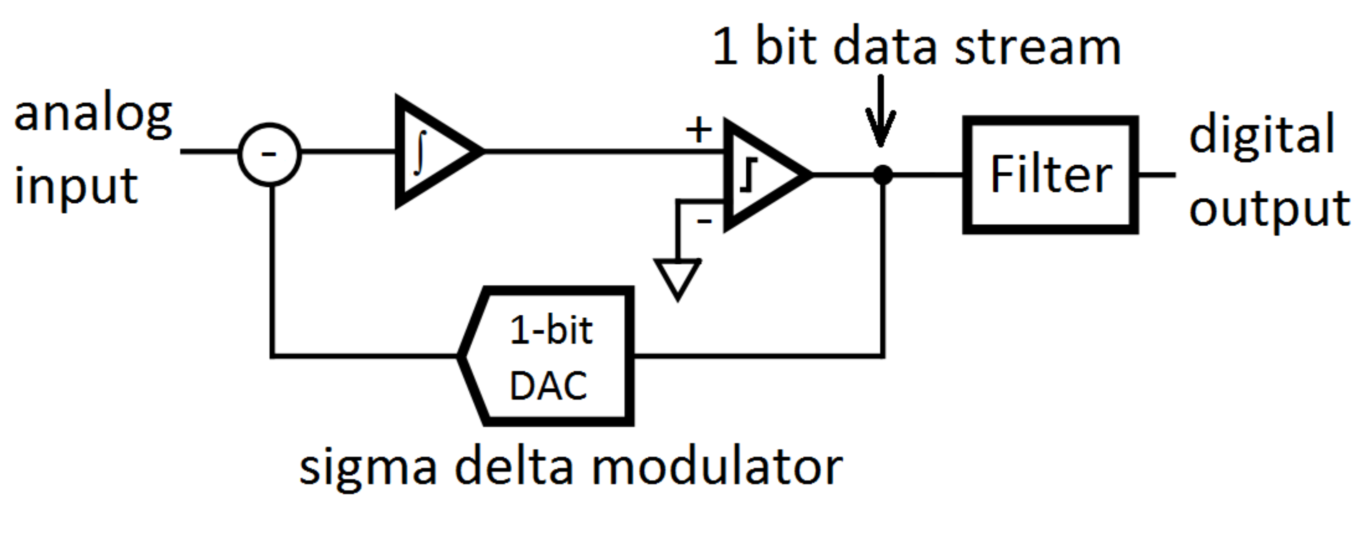


Figure 20 Sigma-Delta ADC

The sigma delta modulator has to generate a analog signal out from an 1 bit digital data stream. The goal is to generator such a single bit data stream. To meet the requirements of high analog resolution at the output, the 1 bit data stream has to be generated with a appropriate high time resultion.

The Multi Channel Sequencers (MCS) of the Generic Timer Module (GTM) are used to generate several bitstreams. These bitstreams are concentrated to a single bitstream by a logic OR gate.

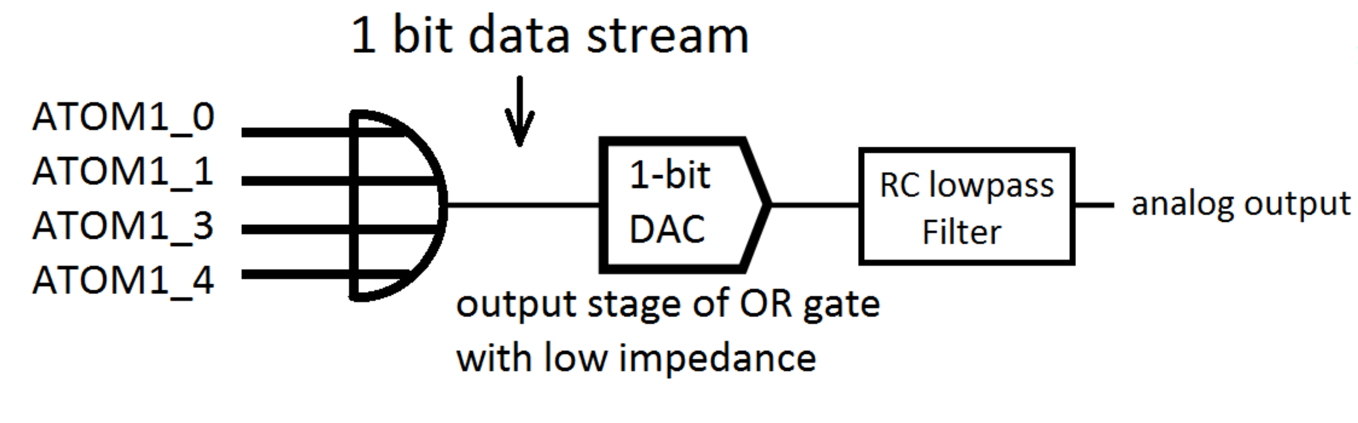


Figure 21 sigmal delta Modulator analog output

On the Demo Board the 1-bit DAC is simply the low impedance output stage of the OR gate. The time resultion of the mixed OR output signal is higher than the time resultion of the single 1-bit datastreams of ATOM3_CHx outputs. This higher time resulation causes a higher analog output signal resultion. This is necessary to supply the frequency tune inputs of the BGT24.

4 Range and speed limits

To calculate range and speed data the following outline may help.
The Output Frequency of the HF Unit is modulated in the range of 24.000 GHz up to 24.250 GHz.
The swing of 250 MHz causes a resolution of 60 cm per dot. With 128 dots per sweep the maxium measureable distance is 0.6m x 128 = 76.4 m.
For the maximum speed calculation the sample frequency of 100MHz / 273 = 366.3 kHz is relevant.
Every sweep consists of 128 dots + 2 dots settling time from sweep to sweep to tune down the oscillator from 24.25 GHz to 24.00 GHz. Also the DAC needs a settling time to stabilize the analog output value.

For a first calculation the following values can be used:
for distance: 150 Mhz = 1 m
for speed: 1 km/h = 44.67 Hz (@ 24.125 GHz)
150 Mhz / 250 Mhz = 0.6 m
366,3 kHz / 130 dots = 2817,7 kHz
2817.7 kHz / 44.67 Hz = 63 km/h full speed range (+/- 31.5 km/h)
The dots in Doppler FFT carry the sign of speed (128 dots = +/-64 dots)

Table 3

Nummber of Dots	128
Frequency swing	250 MHz
Sample Frequency	366.3 kHz
Max Distance	76.4 m
Speed range	+/- 31.5 km/h
angle range	+/- 32°

angle

Revision History

Major changes since the last revision

Page or Reference	Description of change

Trademarks of Infineon Technologies AG
AURIX™, C166™, CanPAK™, CIPOS™, CoolGaN™, CoolMOS™, CoolSET™, CoolSiC™, CORECONTROL™, CROSSAVE™, DAVE™, DI-POL™, DrBlade™, EasyPIM™, EconoBRIDGE™, EconoDUAL™, EconoPACK™, EconoPIM™, EiceDRIVER™, eupec™, FCOS™, HITFET™, HybridPACK™, Infineon™, ISOFACE™, IsoPACK™,
i-Wafer™, MIPAQ™, ModSTACK™, my-d™, NovalithIC™, OmniTune™, OPTIGA™, OptiMOS™, ORIGA™, POWERCODE™, PRIMARION™, PrimePACK™, PrimeSTACK™, PROFET™, PRO-SIL™, RASIC™, REAL3™, ReverSave™, SatRIC™, SIEGET™, SIPMOS™, SmartLEWIS™, SOLID FLASH™, SPOC™, TEMPFET™, thinQ!™, TRENCHSTOP™, TriCore™.

Trademarks updated August 2015

Other Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition <2016-03-08>
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2016 Infineon Technologies AG.
All Rights Reserved.

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
AppNote Number

IMPORTANT NOTICE
The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.