Developer:
SIE Juergen Sievers <JSievers@NadiSoft.de>

Directories:

+ FreeRTOS/source contains the FreeRTOS real time kernel source code.

+ FreeRTOS/Source/portable/ZDS II/eZ80 contains the EZ80F91 port code.

+ FreeRTOS/demo contains a pre-configured demo project for my Demo Application running on ZiLOG's EZ80F910300ZCOG Developer Kit using ZDS II Acclaim 5.2.1 Developer Environment

+ See http://www.freertos.org/a00017.html for full details of the FreeRTOS directory structure and information on locating the files you require.

  + FreeRTOS-Plus contains additional FreeRTOS components and third party complementary products. THESE ARE LICENSED SEPARATELY FROM FreeRTOS although all contain open source options. See the license files in each respective directory for information.

+ FreeRTOS-Plus/Demo contains pre-configured demo projects for the FreeRTOS-Plus

  components. Most demo projects run in a Windows environment using the FreeRTOS windows simulator. These are documented on the FreeRTOS web site http://www.FreeRTOS.org/plus

Further readme files are contains in sub-directories as appropriate.

The easiest way to use FreeRTOS is to start with one of the pre-configured demo application projects (found in the FreeRTOS/Demo directory). That way you will have the correct FreeRTOS source files included, and the correct include paths configured. Once a demo application is building and executing you can remove the demo application file, and start to add in your own application source files.

See also -
http://www.freertos.org/FreeRTOS-quick-start-guide.html
http://www.freertos.org/FAQHelp.html

Prepare to build and use this port.
You need ZiLOG's EZ80F910300ZCOG Developer Kit or equivalents.
Additional ZiLOG's ZDSII Acclaim 5.2.1 Developer Environment. Free download from www.zilog.com

Now lets go:

This Demo is based on FreeRTOS V9.0.0rc2. Some files were modified, because ZiLOG's Dev-Enwironment isn't able to compile FreeRTOS as it is.

The Project are located at “Z:\ZDSII_eZ80Acclaim!_5.2.1\FreeRTOS_V9.0.0rc2” if you install the project on different location you must change the ZDS II Project file too.

Start the ZDSII eZ80Acclaim! IDE an load the project file at
Z:\ZDSII_eZ80Acclaim!
_5.2.1\FreeRTOS_V9.0.0rc2\FreeRTOS\Demo\ZDSII_eZ80F91\EZ80F91_PortDemo.zdsproj

There are two configuration included Debug-Ram and Debug-Flash. Select Debug-Ram to let the IDE map System-Ram to the start area downloading and running the code on Ram (Debugger only). For stand alone system use Debug-Flash.

Debug-Ram settings

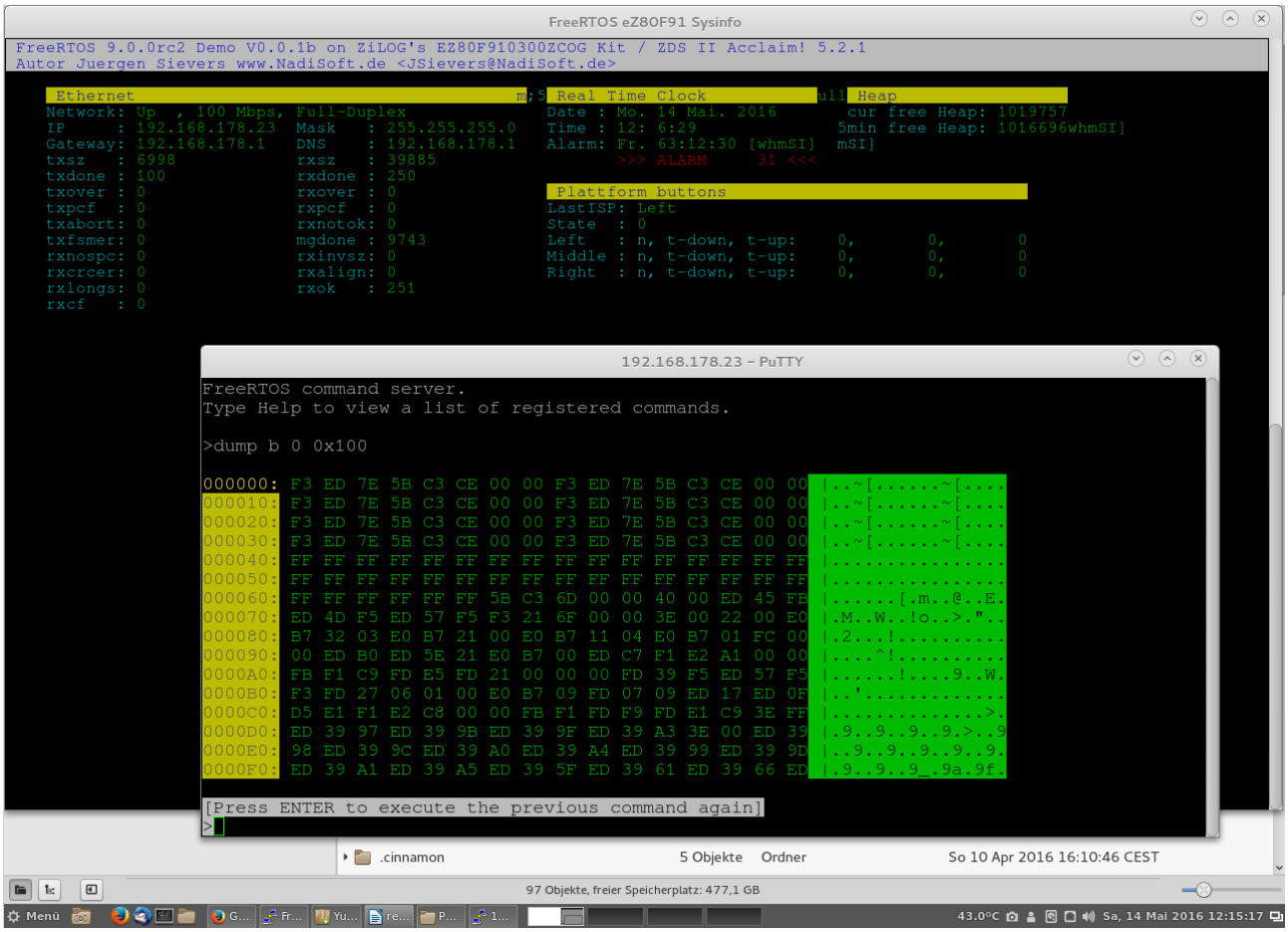
```
RANGE ROM $0 : $07FFFF       (mapped ram)
RANGE RAM $B7E000 : $BFFFFF
RANGE EXTIO $0 : $FFFF
RANGE INTIO $0 : $FF
DEFINE __CS0_LBR_INIT_PARAM = $10
DEFINE __CS0_UBR_INIT_PARAM = $1f
DEFINE __CS0_CTL_INIT_PARAM = $a0
DEFINE __CS0_BMC_INIT_PARAM = $02
DEFINE __CS1_LBR_INIT_PARAM = $00
DEFINE __CS1_UBR_INIT_PARAM = $07
DEFINE __CS1_CTL_INIT_PARAM = $28
DEFINE __CS1_BMC_INIT_PARAM = $02
DEFINE __CS2_LBR_INIT_PARAM = $80
DEFINE __CS2_UBR_INIT_PARAM = $bf
DEFINE __CS2_CTL_INIT_PARAM = $28
DEFINE __CS2_BMC_INIT_PARAM = $02
DEFINE __CS3_LBR_INIT_PARAM = $00
DEFINE __CS3_UBR_INIT_PARAM = $00
DEFINE __CS3_CTL_INIT_PARAM = $00
DEFINE __CS3_BMC_INIT_PARAM = $02
DEFINE __RAM_CTL_INIT_PARAM = $C0
DEFINE __RAM_ADDR_U_INIT_PARAM = $B7
DEFINE __FLASH_CTL_INIT_PARAM = $60
DEFINE __FLASH_ADDR_U_INIT_PARAM = $00
define _SYS_CLK_FREQ = 50000000
```

Debug-Flash settings

```
ANGE ROM $0 : $03FFFF, $100000 : $1FFFFF
RANGE RAM $B7E000 : $C7ffff
RANGE EXTIO $0 : $FFFF
RANGE INTIO $0 : $FF
DEFINE __low_romdata = copy base of DATA
DEFINE __low_data = base of DATA
DEFINE __len_data = length of DATA
DEFINE __low_bss = base of BSS
DEFINE __len_bss = length of BSS
DEFINE __stack = highaddr of RAM + 1
DEFINE __heaptop = highaddr of RAM
DEFINE __heapbot = top of RAM + 1
DEFINE __low_romcode = copy base of CODE
DEFINE __low_code = base of CODE
DEFINE __len_code = length of CODE
DEFINE __copy_code_to_ram = 0
DEFINE __crtl = 1
DEFINE __CS0_LBR_INIT_PARAM = $10
DEFINE __CS0_UBR_INIT_PARAM = $1f
DEFINE __CS0_CTL_INIT_PARAM = $a8
DEFINE __CS0_BMC_INIT_PARAM = $02
DEFINE __CS1_LBR_INIT_PARAM = $c0
DEFINE __CS1_UBR_INIT_PARAM = $c7
DEFINE __CS1_CTL_INIT_PARAM = $28
DEFINE __CS1_BMC_INIT_PARAM = $02
DEFINE __CS2_LBR_INIT_PARAM = $80
DEFINE __CS2_UBR_INIT_PARAM = $bf
```

```
DEFINE __CS2_CTL_INIT_PARAM = $28
DEFINE __CS2_BMC_INIT_PARAM = $02
DEFINE __CS3_LBR_INIT_PARAM = $00
DEFINE __CS3_UBR_INIT_PARAM = $00
DEFINE __CS3_CTL_INIT_PARAM = $00
DEFINE __CS3_BMC_INIT_PARAM = $02
DEFINE __RAM_CTL_INIT_PARAM = $C0
DEFINE __RAM_ADDR_U_INIT_PARAM = $B7
DEFINE __FLASH_CTL_INIT_PARAM = $68
DEFINE __FLASH_ADDR_U_INIT_PARAM = $00
define _SYS_CLK_FREQ = 50000000
```

Compile and download the Project. I use the USBSmartCable. If you use a different debugger don't forget to change the configuration.
To see the System-Monitor output connect a terminal (PuTTY) to UART 0 (CONSOLE) Set the terminal to 115200,8,n,1 (RTS/CTS). The System-Monitor uses ANSI-Codes for decorations. So set your terminal-emulation to understand ANSI on 132 columns 42 rows. You may also connect by raw-telnet on port 5010 to one small CLI thread.



Have fun. Hints are welcome, join the project - if you like.

Regrads
Jürgen