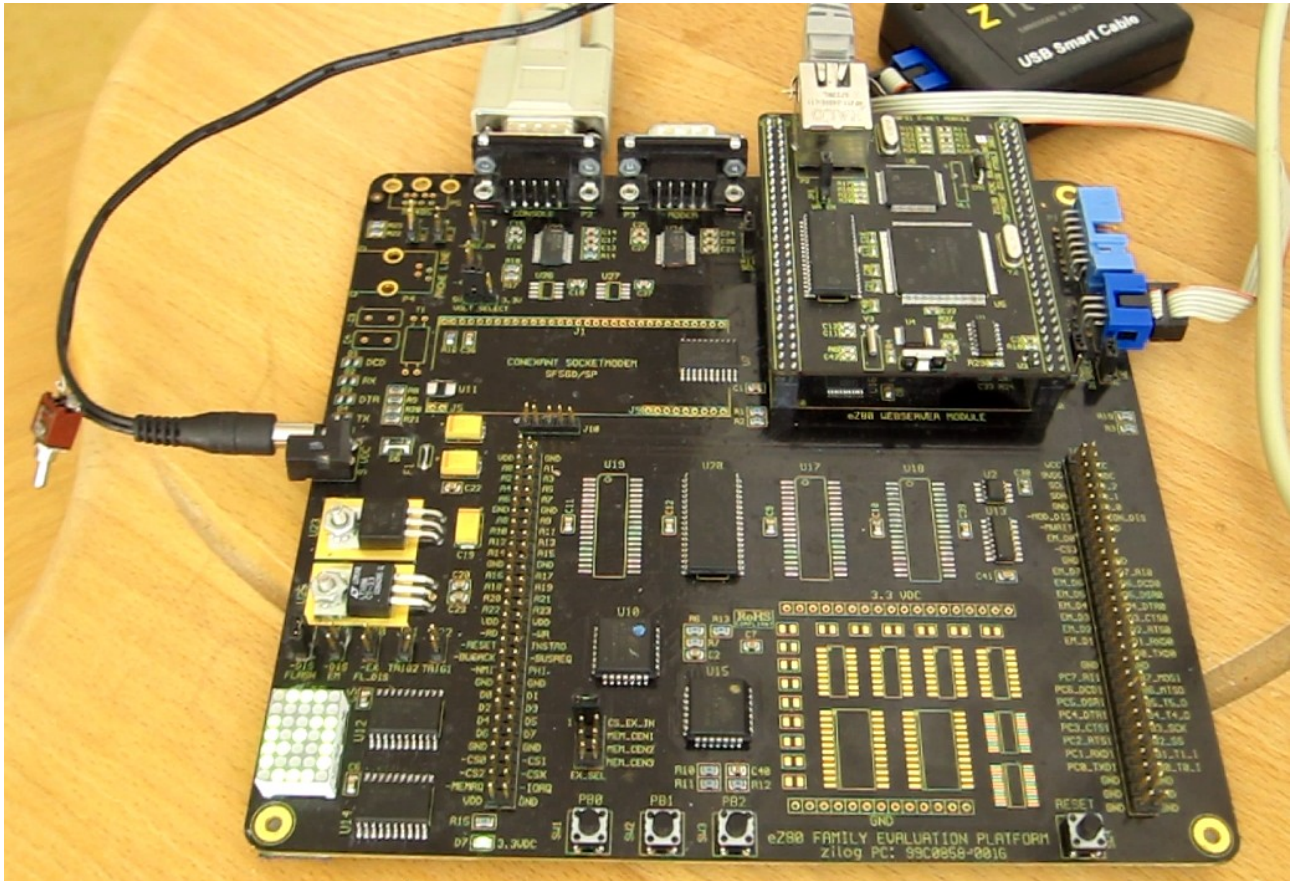


FreeRTOS eZ80F91 Acclaim! Port - Copyright (C) 2016 by NadiSoft All rights reserved

This file is part of the FreeRTOS port for ZiLOG's EZ80F91 Module.
Copyright (C) 2016 by Juergen Sievers <JSievers@NadiSoft.de>
The Port was made and rudimentary tested on ZiLOG's
EZ80F910300ZCOG Developer Kit using ZDSII Acclaim 5.2.1 Developer
Environmen and comes WITHOUT ANY WARRANTY to you!

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-Plus/Source/FreeRTOS-Plus-TCP/portable/Compiler/ZDSII_eZ80Acclaim!_5.2.1 contains ZDS II compiler specifics.
- + FreeRTOS-Plus/Source/FreeRTOS-Plus-TCP/portable/NetworkInterface/eZ80F91 contains EMAC Device Driver
- + 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 V9.0.0. Some files were modified, because ZiLOG's Dev-Environment isn't able to compile FreeRTOS as it is.

The Project are located at "Z:\ZDSII_eZ80Acclaim!_5.2.1\FreeRTOS" 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\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 : \$BFFFFFF

RANGE EXTIO \$0 : \$FFFF

RANGE INTIO \$0 : \$FF

CHANGE STRSECT is ROM

ORDER .RESET, .IVECTS, .STARTUP, CODE, DATA

COPY DATA ROM

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 = \$b8

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 = 5000000000
```

Debug-Flash settings

```
RANGE ROM $0 : $03FFFF, $100000 : $1FFFFFF
RANGE RAM $B7E000 : $C7ffff
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 = $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 = $B8
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 = 500000000
```

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.

```

FreeRTOS eZ80F91 Sysinfo
FreeRTOS 9.0.0rc2 Demo V0.0.1b on ZiLOG's EZ80F910300ZCOG Kit / ZDS II Acclaim! 5.2.1
Autor Juergen Sievers www.NadiSoft.de <JSievers@NadiSoft.de>

Ethernet: Up, 100 Mbps, Full-Duplex
IP: 192.168.178.23 Mask: 255.255.255.0
Gateway: 192.168.178.1 DNS: 192.168.178.1
txsz: 6998 rxsz: 39885
txdone: 100 rxdone: 250
txover: 0 rxover: 0
txpcif: 0 rxpcif: 0
txabort: 0 rxnotok: 0
txfsmr: 0 mgdone: 9743
rxnospc: 0 rxinvsz: 0
rxrcrcr: 0 rxalign: 0
rxlongs: 0 rxok: 251
rxcf: 0

5 Real Time Clock
Date: Mo. 14 Mai. 2016
Time: 12: 6:29
Alarm: Fr. 63:12:30 [whmSI] mSI]
>>> ALARM 31 <<<

Platform buttons
LastISP: Left
State: 0
Left: n, t-down, t-up: 0, 0, 0
Middle: n, t-down, t-up: 0, 0, 0
Right: n, t-down, t-up: 0, 0, 0

192.168.178.23 - PuTTY
FreeRTOS command server.
Type Help to view a list of registered commands.

>dump b 0 0x100

000000: F3 ED 7E 5B C3 CE 00 00 F3 ED 7E 5B C3 CE 00 00 [...]~[.....~[....
000010: F3 ED 7E 5B C3 CE 00 00 F3 ED 7E 5B C3 CE 00 00 [...]~[.....~[....
000020: F3 ED 7E 5B C3 CE 00 00 F3 ED 7E 5B C3 CE 00 00 [...]~[.....~[....
000030: F3 ED 7E 5B C3 CE 00 00 F3 ED 7E 5B C3 CE 00 00 [...]~[.....~[....
000040: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF [...]~[.....~[....
000050: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF [...]~[.....~[....
000060: FF FF FF FF FF FF 5B C3 6D 00 00 40 00 ED 45 FB [...]~[.....~[....
000070: ED 4D E5 ED 57 F5 F3 21 6F 00 00 3E 00 22 00 E0 [...]~[.....~[....
000080: B7 32 03 E0 B7 21 00 E0 B7 11 04 E0 B7 01 FC 00 [...]~[.....~[....
000090: 00 ED B0 ED 5E 21 E0 B7 00 ED C7 F1 E2 A1 00 00 [...]~[.....~[....
0000A0: FB F1 C9 FD E5 FD 21 00 00 FD 39 F5 ED 57 F5 [...]~[.....~[....
0000B0: F3 FD 27 06 01 00 E0 B7 09 FD 07 09 ED 17 ED 0F [...]~[.....~[....
0000C0: D5 E1 F1 E2 C8 00 00 FB F1 FD F9 FD E1 C9 3E FF [...]~[.....~[....
0000D0: ED 39 97 ED 39 9B ED 39 9F ED 39 A3 3E 00 ED 39 [...]~[.....~[....
0000E0: 98 ED 39 9C ED 39 A0 ED 39 A4 ED 39 99 ED 39 9D [...]~[.....~[....
0000F0: ED 39 A1 ED 39 A5 ED 39 5F ED 39 61 ED 39 66 ED [...]~[.....~[....

[Press ENTER to execute the previous command again]

```

SIE 160526 add FreeRTOS trace

To use trace facility set configINCLUDE_TRACE_FACILITY to 1 on FreeRTOSConfig.h

To download the trace-data start the CLI console (telnet target-ip 5010) and enter "ihex trace"

The CLI command dumps the trace-data in intel-hex format to the telnet-client.

```

===== PuTTY log 2016.05.26 21:34:46 =====
FreeRTOS command server.
Type Help to view a list of registered commands.

>ihex trace

# Tracebuffer B7FB8E-B8139E
:0200000400B743
:10FB8E000102030471727374F1F2F3F4A11A04000A
:10FB9E000101800009B070100E8030000FB010000A5
:10FBAE0001000000F401000098010000A25F8B41EB
:10FBBE000100000000000000000000000000000036
...
:10137000C7060703410402085E170608B1035005BB
:10138000EA035005E203A800030088017061000031
:0E1390000000A0B0C0D71727374F1F2F3F48D
:00000001FF

[Press ENTER to execute the previous command again]
>

```

Save the captured data to an file file.ihex.

On the linux shell enter

```
srec_cat -Output <file>.bin -Binary <file>.ihex -Intel -offset -<offset>
```

(replace <file> by the file name the dump has been saved and <offset> by the first value shown on the first dumped line B7FB8E. See above
Tracebuffer B7FB8E-B8139E)

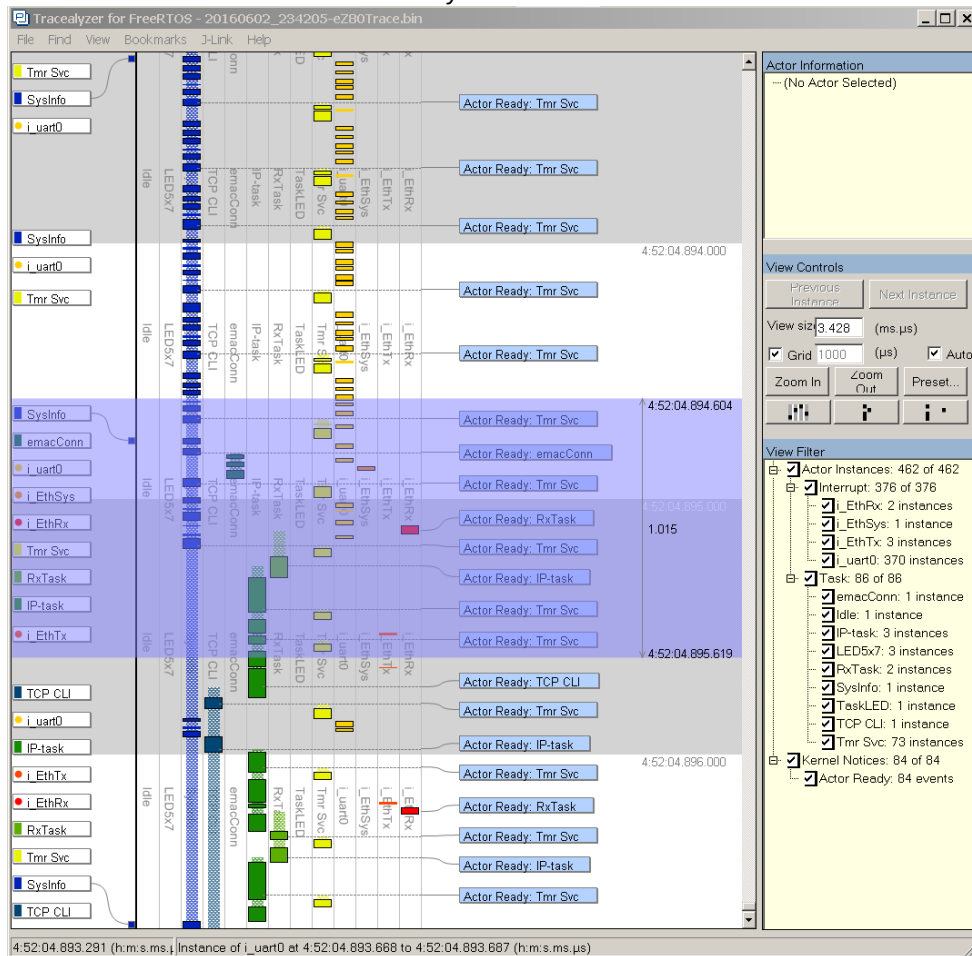
After that conversion you can load the <file>.bin on the trace viewer.
See Tracealyzer Trace Recorder Library

Percepio AB
www.percepio.com
for more information about this viewer.

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

Regrads
Jürgen

History:
06/02/16 SIE: update to FreeRTOS 9.0.0
Fix CLI ihex address input.
04/06/16 SIE: Label Demo Version 1.0.0
Added trace facility.



06/06/16 SIE: Add Mixed-Mode to run FreeRTOS on mixed mode. Also gets able to run Z80 tasks each on its own secure 64K-Page.