NETMF for STM32

Technical Notes Release 4.2

File:	NETMF for STM32 - Technical Notes Release 4.2.pdf
Date:	07.09.2011
Author:	B. Heeb, Oberon microsystems, Inc.
Distribution:	NETMF Porting Kit

Scope

The port is intended for the high and XL density performance line microprocessors STM32F103xE/xF/xG. It can also be used as a basic port for the connectivity line devices (STM32F105, STM32F107). A separate port is necessary for the second generation controllers (STM32F2x), because their peripherals have additional features.

Cortex-M3 Core Port

The STM32 port includes a generic port to the Cortex-M3 core. The sources are stored in the directory \DeviceCode\Targets\Native\STM32\DeviceCode\CortexM3.

There are two subdirectories:

- GlobalLock: interrupt enable/disable handling
- TinyHal: startup code and the interrupt handler tables

The corresponding files for other cores are found under the directories \Application\common and \DeviceCode\cores.

STM32 Drivers

The following basic drivers for STM32 are available:

- STM32_Bootstrap (clock configuration)
- STM32_IntC (interrupt handling)
- STM32_Power (sleep handling)
- STM32_Time (timer interrupt)

The following peripheral devices are supported:

- STM32_Analog
- STM32_Flash (internal Flash write/erase)
- STM32_GPIO
- STM32_I2C
- STM32_PWM
- STM32_SPI
- STM32_USART
- STM32_USB

Platform Configuration

The platform configuration file (platform_selector.h) allows customizing the port for a specific platform. There are some noteworthy additions to the standard entries:

- I2C Device Configuration:

  I2C1 is used by default. If you like to use I2C2 instead, include the following line:

  ```
  #define STM32_USE_I2C2  1
  ```

- Clock Configuration:

  The following STM32 clocks can be set within the limits allowed by the controller:

name	*STM32 clock*	*proven values*		
SYSTEM_CRYSTAL_CLOCK_HZ	HSE	8,000,000	8,000,000	8,000,000
SYSTEM_CLOCK_HZ	SYSCLK	72,000,000	72,000,000	48,000,000
SYSTEM_CYCLE_CLOCK_HZ	HCLK	72,000,000	72,000,000	12,000,000
SYSTEM _APB1_CLOCK_HZ	PCLK1	36,000,000	9,000,000	12,000,000
SYSTEM _APB2_CLOCK_HZ	PCLK2	72,000,000	9,000,000	12,000,000

- USB Attach Pin:

  If the USB attach pull-up resistor is controlled by a GPIO pin, this can be configured as follows:

  ```
  #define STM32_USB_Attach_Pin_High    <pin>  (active high USB attach pin)
  #define STM32_USB_Attach_Pin_Low     <pin>  (active low USB attach pin)
  #define STM32_USB_Attach_Pin_Direct  <pin>  (USB pull-up resistor directly connected to pin)
  ```

Pins are numbered as follows:

PA0 = 0, PA15 = 15,
PB0 = 16, PB15 = 31,
and so on

The MCBSTM32E Solution

The MCBSTM32E solution is a port to the Keil MCBSTM32E evaluation board. The port uses the external Flash to store the managed code assemblies and the external RAM for the heap.

Solution Drivers

- BlockStorage: Flash configuration (internal & external)
- Init: IO and FSMC initialization
- M25P64: external Flash driver (SPI based)
- USB: USB configuration

Booter

The PortBooter is not implemented.

The TinyBooterDecompressor is not used. The TinyBooter starts directly from Flash. The booter cannot overwrite itself. Therefore, to rewrite the booter, a RAM version of the booter has to be loaded first.

Memory Map

Address	*Type*	*Content*	*Comments*
08000000 - 0800AFFF	Flash	TinyBooter	
0800B000 - 0805FFFF	Flash	Firmware image	CLR and libraries
08060000 - 08061FFF	Flash	Firmware configuration	
08062000 - 08063FFF	Flash	Storage region A	Extended week references
08064000 - 08065FFF	Flash	Storage region B	Extended week references
08066000 - 0807FFFF	Flash	(Deployment)	Currently unused
00000000 - 003FFFFF	Ext. Flash	Deployment	Deployed managed code
20000000 - 2000BFFF	RAM	Variables	
2000C000 - 2000C1FF	RAM	Custom Heap	Interrupt handler table only
2000C200 - 2000FFFF	RAM	Stack	
68000000 - 680FFFFF	Ext. RAM	Heap	

The STM32Stamp Solution

The STM32Stamp solution is a port to the Futurlec ET-STM32-Stamp module. The module just contains an STM32F103RET and a serial connection to COM1. The port uses minimal RAM and ROM space and can be used as a generic 'small system port'.

Solution Drivers

- BlockStorage: Flash configuration
- Init: IO initialization

Booter

Neither the TinyBooter nor the PortBooter are used in this solution. The reset vector directly starts the CLR. The built-in system bootloader of the STM32 is used to reload the firmware if needed. The Flash Loader Demonstrator, a freeware tool from ST, is used on the PC side to download the firmware image to the controller:

- tinyclr.bin\ER_FLASH must be written to the address 0x08000000.
  tinyclr.bin\ER_CONFIG must be written to the address 0x08040000.

Memory Map

Address	*Type*	*Content*	*Comments*
08000000 – 0803DFFF	Flash	Firmware image	CLR and libraries
0803E000 – 0803EFFF	Flash	Storage region A	Extended week references
0803F000 – 0803FFFF	Flash	Storage region B	Extended week references
08040000 – 08041FFF	Flash	Firmware configuration	
08042000 - 0807FFFF	Flash	Deployment	Deployed managed code
20000000 – 20005DFF	RAM	Variables	
20005E00 – 20005FFF	RAM	Custom Heap	Interrupt handler table only
20006000 - 20007FFF	RAM	Stack	
20008000 - 2000FFFF	RAM	Heap	