IETF Hackathon: Software Updates for IoT (SUIT)

IETF 102
3-4 Nov, 2018
Bangkok
The Group

Team:
- Hannes Tschofenig
- Jaime Jiménez
- Tadahiko Ito
- Yohei Kaieda
- Yuichi Takita
- Dan Petrie
- Emmanuel Baccelli
- Henk Birkholz
- Chris Inacio
- Laurence Lundblade

1<sup>st</sup> time IETF
1<sup>st</sup> time SUIT Hackathon
Lots of Hardware

1. STM32F4-DISCOVERY
2. NUCLEO-F207ZG
3. SAMR21 xpro
4. Renesas Starter Kit for RX231
5. MCBSTM32F400
6. Custom SUIT Hackathon Board
Create Firmware & Manifest

Transport Firmware & Manifest

Dockerized development environments (Mbed OS & RIOT)

Laptop

Renesas

USB

JLINK

STM

CoAP

SAMR21 xpro
MCUs...

- Configuring MCUs is often necessary to have only those features enabled that are needed.
- This helps to keep the bootloader as small as possible.
- Example shows a screenshot of configuring the pin layout of the STM32F407 MCU via the STM32 CubeMX utility.
- Reading through the reference manual is obviously useful when configuring low level features. Example can be found here.
### Table 5. Flash module organization (STM32F40x and STM32F41x)

<table>
<thead>
<tr>
<th>Block</th>
<th>Name</th>
<th>Block base addresses</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main memory</td>
<td>Sector 0</td>
<td>0x0800 0000 - 0x0800 3FFF</td>
<td>16 Kbytes</td>
</tr>
<tr>
<td></td>
<td>Sector 1</td>
<td>0x0800 4000 - 0x0800 7FFF</td>
<td>16 Kbytes</td>
</tr>
<tr>
<td></td>
<td>Sector 2</td>
<td>0x0800 8000 - 0x0800 BFFF</td>
<td>16 Kbytes</td>
</tr>
<tr>
<td></td>
<td>Sector 3</td>
<td>0x0800 C000 - 0x0800 0FFF</td>
<td>16 Kbytes</td>
</tr>
<tr>
<td></td>
<td>Sector 4</td>
<td>0x0801 0000 - 0x0801 1FFF</td>
<td>64 Kbytes</td>
</tr>
<tr>
<td></td>
<td>Sector 5</td>
<td>0x0802 0000 - 0x0803 0FFF</td>
<td>128 Kbytes</td>
</tr>
<tr>
<td></td>
<td>Sector 6</td>
<td>0x0804 0000 - 0x0805 0FFF</td>
<td>128 Kbytes</td>
</tr>
<tr>
<td></td>
<td>Sector 7</td>
<td>0x0805 0000 - 0x0805 0FFF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sector 8</td>
<td>0x0805 0000 - 0x0805 0FFF</td>
<td></td>
</tr>
<tr>
<td>System memory</td>
<td></td>
<td>0x1FFF 0000 - 0x1FFF 77FF</td>
<td>30 Kbytes</td>
</tr>
<tr>
<td>OTP area</td>
<td></td>
<td>0x1FFF 7800 - 0x1FFF 7A0F</td>
<td>528 bytes</td>
</tr>
<tr>
<td>Option bytes</td>
<td></td>
<td>0x1FFF C000 - 0x1FFF C00F</td>
<td>16 bytes</td>
</tr>
<tr>
<td>Position</td>
<td>Priority</td>
<td>Type of priority</td>
<td>Acronym</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Reserved</td>
</tr>
<tr>
<td>-3</td>
<td>fixed</td>
<td>Reset</td>
<td>Reset</td>
</tr>
<tr>
<td>-2</td>
<td>fixed</td>
<td>NMI</td>
<td>Non maskable interrupt. The RCC Clock Security System (CSS) is linked to the NMI vector.</td>
</tr>
<tr>
<td>-1</td>
<td>fixed</td>
<td>HardFault</td>
<td>All class of fault</td>
</tr>
<tr>
<td>0</td>
<td>settable</td>
<td>MemManage</td>
<td>Memory management</td>
</tr>
<tr>
<td>1</td>
<td>settable</td>
<td>BusFault</td>
<td>Pre-fetch fault, memory access fault</td>
</tr>
<tr>
<td>2</td>
<td>settable</td>
<td>UsageFault</td>
<td>Undefined instruction or illegal state</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Reserved</td>
</tr>
<tr>
<td>3</td>
<td>settable</td>
<td>SVCall</td>
<td>System service call via SWI instruction</td>
</tr>
<tr>
<td>4</td>
<td>settable</td>
<td>Debug Monitor</td>
<td>Debug Monitor</td>
</tr>
</tbody>
</table>
// Declare function pointer
void (*firmware_reset_handler)(void);

// Configure MSP
uint32_t msp_value = *(volatile uint32_t *)FLASH_BASE_ADDRESS;
__set_MSP(msp_value);

// Configure reset handler
uint32_t resethandler_address = *(volatile uint32_t *) (FLASH_BASE_ADDRESS + 4);

firmware_reset_handler = (void*) resethandler_address;

// Jump to reset handler of application firmware
firmware_reset_handler();
What we learned

• First day of a Hackathon is always pain. We always get something wrong and forget stuff at home...
  – Power adapter killed → Shop visit
  – Missing Serial-to-USB cable → Shop visit
• Help is nearby
  – CDDL is hell → Ask Carsten
  – COSE is hell too → Ask Jim
• SUIT manifest draft needs an update (CDDL and examples)

+1
No development boards have been killed this time!
Yayy!! Software Updates for IoT (SUIT) was the winner at the #IETFHackathon #IETF103
More Info

IETF SUIT Working Group: https://datatracker.ietf.org/wg/suit/about/

Manifest draft used: https://tools.ietf.org/html/draft-moran-suit-manifest-03

Poster: http://jaimejim.github.io/docs/suit_poster.pdf

Detailed write-up available at https://etherpad.tools.ietf.org/p/FUIETF103