openMMC
A modular open source firmware for MMCs

10th meeting of the xTCA interest group

Henrique Aires Silva
Brazillian Synchrotron Light Source (LNLS/Sirius)
Outline

- The MMC in a uTCA system
- Existing Implementations
- Why a new design?
- openMMC Firmware
  - Features
  - Structure
- Porting Example
- Integration Tests
- Future Developments
The MMC in a uTCA system

- MMC
- Hotswap
- Temperature Sensors
- AMC Connector
- IPMB
- RTM
- DCDC Converters
- Current/Voltage Sensors
- Payload
- User Flash
- FRU
- SDR
- JTAG
- SPI
- I2C
- GPIO
- Power
# Existing Implementations

<table>
<thead>
<tr>
<th>DESY MMC</th>
<th>University of Winsconsin</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ATMega128</td>
<td>• AT32UC3A1512</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CERN MMC</th>
<th>Warsaw University of Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ATMega128</td>
<td>• LPC1764</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N.A.T. Europe</th>
<th>COSYLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>• ATMega128</td>
<td>• LPC2136</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>coreIPM</th>
<th>JAMMCI (GSI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• LPC2xxx</td>
<td>• LPC1764</td>
</tr>
<tr>
<td>• Cypress nvPSoC</td>
<td>• MATPEX 1A</td>
</tr>
<tr>
<td></td>
<td>• ATMega128</td>
</tr>
</tbody>
</table>

10th meeting of the xTCA interest group
Why a new design?

- Difficult maintenance with spread forks
- Incompatibility between the MMC and MCHs from different vendors
- **Hard to port the code to different boards/controllers**
  - Microcontroller specific code mixed with application routines
  - IDE-dependent compilation/debug
Features

- Open Source – GPLv3 (All code hosted at GitHub)
- Modular structure
  - Chip specific code separated from application
  - Independent modules
- CMake build system
- All board ports in the main repository
- HPM.1
- Doxygen documentation (in progress)

*Initial collaboration between LNLS (Brazil) and GSI (Germany)
FreeRTOS

- Multitasking
  - Preemptive scheduler
- Inter-task communication
  - Queues
  - Binary and counting semaphores
  - Recursive mutexes
  - Task notifications
- 6K to 12K ROM footprint
- Numerous ports ( > 100 )
  - All maintained and supported by Real Time Engineers!
Code Structure

Application (Tasks)

- RTM
- Sensors
- Payload
- LEDs

HAL

Drivers

- I2C
- SPI
- IPMB

- LPCOpen
- CMSIS
- Atmel ASF
Code Structure

- **Modules**
  - Sensors
  - IPMI
  - LEDs
  - Clock Switch
  - ...

- **Port**
  - uController
    - NXP
    - (ATMEL)
  - Board
    - LPC17xx
    - (ATMEGA128)
    - AFC
    - User FRU
    - Timing AMC
    - Pin Mapping

Custom board modules

10th meeting of the xTCA interest group
Code Structure

Application – Custom task

```c
gpio_set_pin_state( GPIO_DAC_VADJ_RST_PORT, GPIO_DAC_VADJ_RST_PIN, LOW);
```

Port layer – Interface between driver and application

```c
#define gpio_set_pin_state( port, pin, state ) ioprt_set_pin_level( pin, state )
```

Drivers – Custom hardware control

```c
static inline void ioprt_set_pin_level(ioprt_pin_t pin, bool level)
{
    arch_ioprt_set_pin_level(pin, level);
}
```

ATMEL ASF
Code Structure

Application – Custom task

```c
gpio_set_pin_state( GPIO_DAC_VADJ_RST_PORT, GPIO_DAC_VADJ_RST_PIN, LOW);
```

Port layer – Interface between driver and application

```c
#define gpio_set_pin_state( port, pin, state ) Chip_GPIO_SetPinState( LPC_GPIO, port,...
```

Drivers – Custom hardware control

```c
STATIC INLINE void Chip_GPIO_SetPinState(LPC_GPIO_T *pGPIO, uint8_t port, uint8_t pin, bool setting)
{
    if (setting) { /* Set Port */
        pGPIO[port].SET |= 1UL << pin;
    }
    else { /* Clear Port */
        pGPIO[port].CLR |= 1UL << pin;
    }
}
```
Porting Example

Different boards

AFC

```c
#List all modules used by this board
set(AFC_MODULES
    "FRU"
    "PAYLOAD"
    "SDR"
    "WATCHDOG"
    "JTAG_SWITCH"
    "CLOCK_SWITCH"
    "FPGA_SPI"
    "DAC_AD84XX"
    "HOTSWAP_SENSOR"
    "TEMPERATURE_SENSOR"
    "VOLTAGE_SENSOR"
    "HPM"
)
```

TimingAMC

```c
#List all modules used by this board
set(TIMINGAMC_MODULES
    "FRU"
    "PAYLOAD"
    "SDR"
    "WATCHDOG"
    "PLL_CTRL"
    "WHITE_RABBIT"
    "HOTSWAP_SENSOR"
    "TEMPERATURE_SENSOR"
    "VOLTAGE_SENSOR"
    "HPM"
)
```
Porting Example

- Driver functions
- FreeRTOS port for the controller

Driver function aliasing

```c
#define gpio_init() Chip_GPIO_Init( LPC_GPIO )
#define gpio_read_pin( port, pin ) Chip_GPIO_GetPinState( LPC_GPIO, port, pin )
#define gpio_read_port( port ) Chip_GPIO_GetPortValue( LPC_GPIO, port )
#define gpio_set_pin( port, pin ) Chip_GPIO_SetPinOutHigh( LPC_GPIO, port, pin )
#define gpio_set_port( port, mask ) Chip_GPIO_SetPortOutHigh( LPC_GPIO, port, mask )
#define gpio_clr_port( port, mask ) Chip_GPIO_SetPortOutLow( LPC_GPIO, port, mask )
#define gpio_pin_toggle( port, pin ) Chip_GPIO_SetPinOutToggle( LPC_GPIO, port, pin )
#define gpio_set_pin_state( port, pin, state ) Chip_GPIO_SetPinState( LPC_GPIO, port, pin, state )
#define gpio_set_pin_dir( port, pin, dir ) Chip_GPIO_SetPinDIR( LPC_GPIO, port, pin, dir )
```

Module

- Develop new managing / monitoring tasks

Board

- Define which modules will be used

FreeRTOS port

- ARM7_AT91FR40008
- ARM7_AT91SAM7S
- ARM7_LPC2000
- ARM7_LPC23xx
- ARM_CA9
- ARM_CM0
- ARM_CM3
- ARM_CM3_MPU

10th meeting of the xTCA interest group
Integration tests

NAT-MCH-PHYS

UTC002

- Fully functional!
- Mostly functional (Few bugs on FRU shutdown)
- Q3/2016 (Planned)

*All tests with AFC boards*
Future developments

✓ Bootloader upgrade
  • Integrate HPM module

✓ Code documentation
  • 06/2016

✓ Detailed wiki page on GitHub
Try out!

- Already ported to AT Mega and LPC17 families
- Easy to customize
- Large set of tools provided by FreeRTOS to ease development
- Expand the project to other boards and controllers

- Encourage collaboration
  - Open to pull-requests!

openMMC repository
https://github.com/Lnls-dig/openmmcx
Thank you!

Henrique Aires Silva

henrique.silva@lnls.br

openMMC repository
https://github.com/lnls-dig/openmmc/

Critics and suggestions are more than welcome!

Brazillian Synchrotron Light Source (LNLS/Sirius)

10th meeting of the xTCA interest group