AMI 1.2: EMCM Firmware Overview

March 7, 2016

Outline

- Architectural Overview of AMI 1.2 Firmware
- Boot Sequence
- Main Run Mode
- Communication with RPMA node
- Communication with HES
- Communication with the Meter
- Meter Reading and Reporting Model
- Over-The-Air Requests

Outline

- Data Stored in External Flash
- Meter Program Update / C12.19 Batch Commands
- Time Sync and Localization of Time
- Configuration and Control Data Stored in Internal Flash
- Power Fail / Last Gasp
- Error and Failure Handling
- Hardware Abstraction and Drivers

Outline

- Code Download and Serial Image Installation
- Serial and Debug Tools
- OTA Communication and Debugging Tools
- Provisioning for Developers
- EMCM Manufacturing / ATE / Meter Assembly
- RMA Processing
- Firmware Build Environment
- Off-Target Test Suite

Architectural Overview of AMI 1.2 Firmware

- Freescale Kinetis K20, ARMv7 Cortex-M4
- 256K memory mapped program flash, executable
- 32K internal data flash, 2-8MB external SPI flash
- 64K SRAM (2 x 32K banks), plus 2K Freescale FlexRAM
- Single-threaded event-loop architecture
- Fork of RACM (Reference Application Communication Module)
- ARM NVIC interrupt model
- Critical Section locking model
- Callback model
- Minimal execution from RAM
- Static memory allocation model, single stack, no heap

Boot Sequence

- SRAM test
- RAM code load (limited to code that must execute from RAM)
- Clock Init (~21MHz internal to FLL 32KHz TCXO, 48MHz bus clock)
- Config Init / Journal Recovery
- GPIO Init
- Random seed init (uptime, RTC time, build stamp, EMCM ID)
- DOM (Data Object Manager) Init
- RPMA Node Startup, Join, TimeSync
- Ustream Sync
- Meter Configuration / Serial Number / Data Orders Check
- Main Processing Loop (MPL)

Main Processing Loop

- Handle one RACM event
- Do UART task processing
- Let host_cmn manage the node
- Decide if there's more work to do
- Loop
  - Or sleep
  - In PowerFail, sleep deep

Handle One RACM Event

- HOST_EVENT_EMCM_INIT
  - EMCM system initialization
- HOST_EVENT_EMCM
  - EMCM_Run()
- HOST_EVENT_RTC
  - Handle RTC second timer wake-up events
- HOST_EVENT_TIME_PAL
  - Handle sub-second timer events
- HOST_EVENT_NVM_PAL
  - Handle external SPI flash operations

EMCM_Run()

- flags = TASK_PAL_GetEventFlags()
- MCM_APP_Run()
  - HandleNowEvent()
    - HandleLaterEvent()
      - IsTxInfoAvailable()
      - MCM_APP_HandleRxTxEvent()
- METER_Run()
  - C12_18_APP_Run()
- MCM_AHP_Run()
- MCM_HCQ_Run()

EMCM Application States

- INIT
- WAIT_FOR_JOIN
- WAIT_FOR_TIME_SYNC
- WAIT_FOR_METER_INFO
- WAIT_FOR_AUTHORIZATION
- RUN
- MANUF_CAL
- BAD_CONFIG
- TEST_MODE
- LAST_GASP
- WAIT_FOR_USTREAM_SYNC
- POST_FAIL

EMCM Application Modes

- NORMAL
- MANUF_CAL
- BAD_CONFIG
- TEST_MODE
- POST_FAIL

Communication with RPMA node

- Custom SPI interface
  - Master Request, Slave Request, Slave Ready
- host_cmn RPMA node communication library
- RF Connection to RPMA Access Point
- Secure Connection to RPMA Gateway
- Ustream
- EMCM Code Download broker
- AppType / Update Interval / Listen Interval (UI/LI)

Communication with HES

- Demux Uplink Service Data Units (SDUs) based upon AppType
- Ustream (AMI 1.1 / AMI 1.2 AppType 30)
- AMI OTA Protocol Header
  - OTA Protocol version
  - Async Indications / Events
  - Request / Response
- AMI OTA Protocol Payload
  - Google Protobuf
    - “Optional” fields
  - Implemented on EMCM using nanopb
- Stream priorities
- Uplink staging buffer reservations

Communication with the Meter

- Meter serial
  - PSEM / C12.18 / C12.21
  - Security Service
  - Optimistic session duration / termination
    - Toggle state timing
- C12.19
  - Table reads and writes, procedures
  - Standard and Manufacturer Tables
- Physical meter interface signals
  - meter sense, meter force, mux ctrl, meter busy

Meter Reading and Reporting Model

- Data Orders
  - orderId
  - UOMs, scaling factors
  - Selection filters
  - Read frequency
  - Reporting frequency

Meter Reading and Reporting Model

- Periodic Indications
  - Billing data, second billing data
  - Load profile data
  - Power quality reports
- Asynchronous Indications
  - Meter flags, relay state
  - Power quality events
  - Application events, MeterCommErrors, Error Indications

Billing Data: I210+C, kV2c, SGM

- Read from meter according to read schedule
  - Summations, Demands, Coincidents
- Encoded into OTA Protobuf message
- Time-shifted in circular buffer in internal flash
- At reporting time, enqueue to Ustream RAM and erase from flash
- One reading per report attempt

Billing Data: I210+

- Summations only (and temperature, voltage, . . .)
- Read from meter according to read schedule
- Encoded into GeI210PlusReading Protobuf sub-message
- Time-shifted in circular buffer in internal flash
- At reporting time, concatenate up to 16 sub-messages into BillingDataInd
- Enqueue to Ustream buffer, and erase from flash

Load Profile Data: I210+C, kV2c, SGM

- Copied from meter according to reporting schedule
- Use snapshot for data coherency in meter flash
- At reporting time, concatenate up to 16 latest-dated LP readings from meter
  - Since last LoadProfileInd, or ~one hour before boot
- Enqueue to Ustream buffer
  - Remember time as most recent LoadProfileInd
- SGM has up to four independent sets
  - Each has its own schedule
  - Each has its own indication, etc.

Over-The-Air Requests

- Get Meter Configuration and EMCM Info
- Set Data Orders
- Get / Set Relay State and other Control Points
- Get instantaneous billing data
  - Optional Billing Data Order filter
- Get historical load profile data
  - startTime, endTime, lpSet (SGM only)
- Get power quality report
  - selectFlags

Over-The-Air Requests

- Get / Clear Meter Flags
  - Standard Flags (edMode, edStatus, …)
  - Manufacturer Specific Flags
  - Extra Mfg Flags
- Perform / Schedule Demand Reset
  - Immediate, TTL (time-to-live), delay until, or valid range
- Enable / Disable / Schedule RTP
  - Immediate, or start/end range
- Perform / Schedule Season Change
  - Immediate, delay until, TTL, or valid range
- Get / Set EMCM Config Values

Over-The-Air Requests

- Meter Read / Write table request
  - Enqueue
  - Execute
  - Stop Programming
  - Flush
- Get Uptime
- EMCM Reset / Hard Reset
  - eraseDataOrders
  - flushOustandingMeterData
  - eraseCountsBlock
  - reformatDom
- Get / Clear EMCM panic blocks / exception buffers

Data Stored in External Flash

- External SPI flash (2MB or 8MB)
- Data Object Manager (DOM)
  - Static allocation of object size
  - Wear-leveling with multiple slots for a given object
  - Initialization and reformatting
- Ustream “resumable” stream persistent state
- MPU/Resumable Requests and Responses
- Code Download / Serial Installation image staging area

Meter Program Update / C12.19 Batch Commands

- HES converts GE MeterMate XML to AMI 1.2 Protobuf
- Table reads / writes / reads
  - Compatibility Check
  - Program Operation / Local Patches
  - Audit Check
- Ustream "resumable" streams
- Multicast User Data (MUD)
- External SPI flash / DOM
- Optional start time / retries

Time Sync and Localization of Time

- Representations of Time
  - POSIX time (seconds since January 1st, 1970)
  - Meter time (C12.19 LTIME_DATE)
  - RPMA Node time (YYYY/MM/DD Seconds.FractionalSeconds)
- Time Sync from RPMA node
- Time Sync to Meter
- DST/TZ config from meter
  - Limited RDATE support, ST54
  - DST/TZ overrides via Data Orders

Time Sync and Localization of Time

- Time challenges on GE meters
  - Line Frequency vs. Crystal
  - Demand Event Times
  - Load Profile block end times
- RTC clock stores HW uptime
  - SW uptime marked relative to HW RTC at boot
  - Fractional seconds synced to RPMA Node Time (~1 ms accuracy wrt GPS)
- debugTimeOffset
- Timers: second and subsecond

Configuration and Control Data Stored in Internal Flash

- Copy / Erase / Modify / Write
  - No wear-leveling
- Manufacturing Block
- Security Block / Application security keys
- Config Block / Auto-upgrade
- Counts Block / Auto-upgrade
  - Future-scheduled events (demand reset, season change, rtp)
  - Journaled due to writes at sensitive times
- Provisioning Params Block
  - Antenna selection (Ptero only)
- Data Orders
  - Flash struct versioning

Power Fail / Last Gasp

- Initial meter power-up / power failure detection
  - PotentialPowerFail (PPF) message to RPMA node
- EMCM reduced power mode
  - Disable LEDs
  - Disable meter comm lines
  - Monitor power fail, zero-crossing, power supply voltage signals
- Momentary vs. Sustained Outage Criteria
  - MaxMomentaryOutageDurationSec (~120 seconds)
  - MaxMomentaryInterruptionDurationSec (~20-60 seconds)
  - MinPowerOnDurationMsec (~3 seconds)
- Record Time of Potential Power Failure

Power Fail / Last Gasp

- If power recovers in time, record Momentary Outage
  - Record:
    - cumNumMomentaryOutages
    - numMomentaryOutages
    - firstEventTimestamp
    - nbrEventInterruptions
  - Send PowerRestore message to RPMA node
- Or commit to Last Gasp / Sustained Outage
  - Send PowerFail (PF) message to RPMA node
  - Reboot when complete AND power is restored

Error and Failure Handling

- Error Indications
  - Timestamp, Result, CorrelationId (for requests), ErrData
  - Lowest Priority Stream/Reservation, may be discarded
- Loss of Ustream Sync
- Loss of RPMA Network Join Status
- Loss of Data Orders Authorization
  - Invalid Configuration Hash
  - Change of Serial Number (module swap)
  - Invalidation by HES (empty message / Hard Reset)
- Unknown Association
  - Periodic retry of BootInd
- 48-hour timeout on ability to enqueue new BillingDataInd messages

Error and Failure Handling

- Debugging Breadcrumbs
  - Uptime
  - Last Network Time
  - Application Boot Reason
  - Abnormal Boot Status
- ARM Exceptions / K20 Watchdog
  - Exception Number
  - Program Counter / Link Register
  - Fault Stats Register / Fault Status Aux
- Panics / Assertions
  - File name / Line number
  - Assert message

Error and Failure Handling

- EMCM manages RPMA Node target state
  - Enable/Disable RF
  - Enable/Disable Node Power
  - Enter/Exit PowerFail / LastGasp mode
- host_cmn manages RPMA Node Interface behavior
  - 24-hour host_cmn inactivity timeout on RPMA node liveness criteria
  - Node SPI failures
  - Node Interface ACK timeouts
- Node Errors not reported in-band
  - Exception: nodeExceptionHash

Hardware Abstraction and Drivers

- K20 MCU reset filtering
- Node SPI driver + 3 signals
- Meter Interface lines / Meter Serial
- Comparator for power supply
  - Smooth vs. Efficient modes
- Super Capacitor charging / balancing HW/SW
  - ADC
- RTC
- LEDs
- SPI Flash
- Debug Serial
- JTAG

Code Download and Serial Image Installation

- Notifications from RPMA node of image availability
  - First image chunk header evaluation, optional reject
  - OTA per-node permission check / cutover
- Image chunk staging (from node or serial)
  - Staged in the DOM (AMI 1.2 only)
- Image container validation
  - DeviceType check
  - Version check
  - CRC check
- Installation
  - Execution from RAM to manipulate Program Flash
  - Point of no return

Serial and Debug Tools

- Python scripts
  - tools/emcm/emcm*.py
- Serial Commands
  - emcm_msg.py message library
- Application Host Protocol (AHP)
- Reliable Host Transport wrappers
  - “reliable”
  - “rht_v2”
  - “opt_psem”
- USB high-voltage isolator in series with USB serial adapter, 115200 baud
- Optical Port
  - Wrapper opt_psem, 9600 baud
  - Ptero / kV2c with Enhanced Power Supply only

Serial and Debug Tools

- Addressable End-points:
  - EMCM Application
  - host_cmn library
  - RPMA node
- ctrl.py
  - RPMA node commands
  - host_cmn commands
  - emcm_msg commands
  - serial logging
- Serial logging
  - packed values / log string dictionary
  - build/*/*.logdict
  - logdict stamp must match

Serial and Debug Tools

- JTAG
  - Segger J-Link
  - JlinkGDBServer
  - USB high-voltage isolator
- GNU gdb for ARM (Sourcery CodeBench Lite 2011.09-69)
  - build/.gdbinit file
- Install FW as .elf file using JTAG
  - Cannot install locked / release builds via JTAG
  - No support for K20 full-chip erase?
- K20 Errata for debugging during sleep modes (e3964)
  - Drain all the electrons

Serial and Debug Tools

- C12.18 parser
- Modes
  - Test Mode / Manufacturing Mode / TxTest Mode
- LED pattern
  - Special debug-only features
- debugTimeOffset
- Super Capacitors
  - Discharge to cold power-on reset (POR) can take hours

OTA Communication and Debugging Tools

- tools/hes_mtp/emcm_hes_logger.py
  - Attach to HES instance to parse and display AMI OTA messages
  - Parse and display HES bus logs after the fact
  - Parse and display individual AMI OTA INF Protobuf messages
  - Shows stream and header information, and arrival/departure time at HES
- tools/hes_mtp/ami_dl_msg_sender.py
  - Sends Downlink Messages OTA through HES MTPDiag port
  - Accepts messages formatted as Protobuf structured text (output of emcm_hes_logger.py)
  - Can be used to encode message to hexadecimal string without sending

OTA Communication: BootInd

- bootDetails:
  - bootReason: 0x0400
  - lastUptime: 2 days, 23:21:49 (256909)
  - lastNetworkTime: 2016/02/23 00:18:24 UTC (1456186704)
  - buildVersion: 2.8.6
  - bootType: BOOT_TYPE_APP
  - appResetReason: INTERNAL_RESET_POWERED_AFTER_LAST_GASP_COMPLETE
  - rtcUptime: 266 days, 20:55:06 (23057706)
- needsHashesAuthorized: false
- endDeviceInfoHash: 0x1918b4b1
- pwrFailTimestamp: 2016/02/23 00:15:46 UTC (1456186546)
- pwrRestoreTimestamp: 2016/02/23 00:18:21 UTC (1456186701)
- timestamp: 2016/02/23 00:23:36 UTC (1456187016)
- bootTime: 2016/02/23 00:18:24 UTC (1456186704)
- bootResult: BOOT_SUCCESS

OTA Communication: BootInd (w/ Exception)

- bootDetails
  - bootReason: 0x0020
  - numWatchdogResets: 1
  - programCounter: 0x00005cee
  - linkRegister: 0x00005ced
  - exceptionNum: 0x0026
  - faultStatus: 0x00000000
  - faultStatusAux: 0x0000dead
  - lastUptime: 14 days, 14:39:10 (1262350)
  - lastNetworkTime: 2015/12/01 09:08:32 UTC (1448960912)
  - buildVersion: 2.8.6
  - bootType: BOOT_TYPE_EXCEPTION
  - rtcUptime: 28 days, 20:55:02 (2494502)
  - numValidPanicBlks: 8
  - panicBlockSeqNum: 12

OTA Communication: Examining Boot Sequence

- BootInd
  - meterDataReportHash
  - RPMA nodeExceptionHash
  - failedCfgBlkBitmask
  - retransmitNum
  - meterSerialNum
- MeterConfigReportReq → MeterConfigReportRsp
  - meterDataReportHash
- EndDeviceInfoReq → EndDeviceInfoRsp
  - meterSerialNum
  - endDeviceInfoHash
- SetDataOrdersReq
  - meterDataReportHash
  - orderId
- HandleErrorReq
  - OTA_UNKNOWN_ASSOCIATION

Provisioning for Developers

- EMCM Provisioning
  - Manufacturing Block (requires magic value for safety)
  - Config Block
  - Security Block
  - Provisioning Params Block (Ptero only today)
- Serial AHP lock
  - When locked, can only check: state, uptime, mcm info, meter info, mfg block, config info, lock status, EMCM exception buffer
  - When locked, FW installation (wipes security block), soft reset
  - Set Serial AHP lock status (with key)
  - Optical port lockout (Ptero only today)
    - No EMCM exception buffer, soft reset, or FW installation

Provisioning for Developers

- Flash lock / JTAG lock
  - Feature of Freescale K20
  - Controlled by bits set in executable flash
  - Changes only take effect on boot
  - Can only be changed via AHP command while EMCM Serial AHP is unlocked
- LED pattern
  - Debug
  - Deployment
  - Disabled
- RPMA node config, RF parameters, security
- RPMA node exception buffer, nodeExceptionHash

EMCM Manufacturing / ATE / Meter Assembly:

- emcm_msg.py serial message library
- EMCM Manufacturing uses AMI 1.0 master image
- Meter factory test mode:
  - TEST_MODE
- EMCM manufacturing / calibration test mode
  - MODE_MANUF_CAL
- Node Tx test mode:
  - TEST_MODE, RF Certification testing, interference testing
- Persistent vs. non-persistent test modes
- RPMA node config: auto-run off
- Kinetis EZPort: full-chip erase

Firmware RMA Processing

- tools/emcm_do_rma.sh
- Meter Factory vs. Fielded Returns
- Unlocking of fielded returns (aqcuire/decrypt key from customer)
- Collection and post-processing of recoverable data:
  - EMCM version and build stamp
  - EMCM State on Boot (Trying to join, POST_FAIL, test mode, bad config, etc....)
  - EMCM configured mode (testMode 0, 1, 2, ...)
  - EMCM ID (does it match the EMCM ID / Node ID on the sticker)
  - EMCM HW options (has it had the SRAM test performed by ORW / Flextronics)

Firmware RMA Processing (Contd.)

- Collection and post-processing of recoverable data:
  - EMCM serialNumber (does it match the sticker)
  - EMCM testDate code
  - RPMA nodeId (if we can talk to the node, does it match sticker)
  - RPMA node version / node build stamp
  - RPMA node config version
  - Is RPMA node auto-run enabled (should be no)
  - Raw SRAM test result identifies bank(s) with stuck bits
  - Any exceptions in the EMCM or RPMA node
  - Check internal image against known good image
  - Check image staging area against known good image (AMI 1.0 and 1.1 only)

Firmware Build Environment

- Linux build environment
  - Scientific Linux 6.x, x86_64
- gcc version 4.6.1 for ARM
  - Sourcery CodeBench Lite 2011.09-69
- Makefile
  - GNU Make 3.81
- Compiled per meter family / Ingenu board pair
- Buildstamp / build version info
- “Fake” and “fraudulent” builds for test purposes

Firmware Build Outputs

- Raw binary (for installation with EZPort / JTAG)
  - emcm_raptor_i210n_rel.bin
- Elf binary (for debugging)
- Disassembly and map file (for convenience)
- Log dictionary (for serial log formatting and display)
- Installer-wrapped binaries (for serial and OTA install)
  - emcm_raptor_i210n_rel.bin.img_v01
  - emcm_raptor_i210n_rel.bin.ota_v01
  - emcm_raptor_i210n_rel.bin.ota_v02
- Tools, scripts, etc.

Firmware: Debug vs. Release builds

- Behavior of ASSERT() calls:
  - Debug:
    - Disable Watchdog and trap
  - Release:
    - Soft Reset
- Flash / JTAG Runtime Locked:
  - Watchdog enabled permanently at boot
    - Enabled in DEBUG halt

Firmware: Debug vs. Release builds

- Flash / JTAG Runtime Unlocked:
  - Watchdog can be disabled after boot
    - Disabled in DEBUG halt
  - If Debug LED Pattern:
    - host_cmn Operating Mode: MONITOR
    - Automatic hostLoggingEnabled
    - Side-effect is logging during power-fail
    - Significant change in sleep behavior
    - MUST have serial logger (ctrl.py or similar) attached

Firmware Off-Target Test Suite

- Python / py4j
- Synthetic RPMA Node/GW/HES/Ustream end-point
- Cached meter table data
- Synthetic meter table data

BACK UP