Clip Microcontroller / EEPROM
Software Specification

History:

1. Original Version 1.5.0 -- 6/30/2011
2. Version 2.3.0 – 9/06/2011
3. Version 2.5.0: LSI Clock Enable for RTC (deprecated with 2.7 release)
4. Version 2.6.0: LSI Clock Calibration (deprecated with 2.7 release)
5. Version 2.7.0: LSE Clock Enable for RTC
6. Version 2.8.0: Tag Version, Next to last packet Reserved

This document specifies the memory map definition and command interface protocol required by the Clip Microcontroller firmware to operate with an RF Host application. Further, the RF command definition along with commands and interrupt statuses is described for writing an RF Host application.

1. EEPROM Memory Map Definition –refer to Table 1 for Memory Organization. The M24LR16E-R device employed in the Clip design is a dual-interface i2c and RF sides.

   Design Notes:

   1. M24LR16E-R 16 Kbit dual-EEPROM organized as:
      a. 2048 bytes in i2c mode
      b. 512 blocks of 32 bits in RF mode

   2. Write time
      a. i2c: 5 ms (max)
      b. RF: 5.75 ms including the internal verify time

   3. **Current Data** occupies i2c Byte Addresses 0x00-0x17 (shown in grey color in Table 1) and are updated by the microcontroller whenever an interrupt occurs or by the RF side whenever an RF command is supplied and the microcontroller executes it.

   4. **History Data** (shown in green/yellow color) is updated only by the microcontroller. It starts at i2c byte address **0x18** and it is sequentially updated until the bottom of the EEPROM is reached or erased when a Flush History command is received from the RF side. History data grows downward sequentially successively until memory is wrapped or flushed by the RF Host.

   5. History Packet Address is dynamically updated with the next memory PACKET location to write the **next** history values after the current command completes – successfully or not.

   6. First History Packet begins at 0x18, followed by next packet at 0x24, then next at 0x30, etc. until final packet is reached at 2028.

7. The History Packet Number contains the number of history packets written in History. For the M24LR16E-R. A history packet consists of 12 bytes.

8. History can only be flushed "explicitly" via a Flush History command by the RF host but only if a Read History command is first called.

9. rtcDayName represents the name of the week, such as, Monday (0x01, Tuesday (0x02), etc.

10. Calendar Real Time Clock (referred from hereon by RTC) must be set/configured by the RF host whenever the microcontroller is reset, battery is removed (system bring-up), low-battery is detected, or when the microcontroller detects a tamper band condition. Under these conditions, the **Current Data** will be invalid and therefore must be configured by the RF host. The microcontroller will leave, if any, the History Packet Address and History Packet Number. <u>By design, a power-reset will not erase the **History Data**</u>. History Data will only be erased by the Flush History RF command provided the History Data has been read by the RF host first.

11. M24LR16E-R Memory Initial State: The device is delivered with all bits in the user memory array set to 1 (each byte contains 0xFF).

12. RTC must be configured by the RF host upon microcontroller Reset or other conditions stated in item 10 above. RTC Current Data will be erased as well.

13. Battery voltage is updated with Normal (0x00) or Low (0x01) values only.

14. Tag Version is represented in hexadecimal values. For example, the current version of the firmware is 2.8, so the value will be represented as 0x28. Also, when the version transitions from 2.9 to 3.0, the value will be 0x30

Table1: Memory Organization

Sector Number	RF block Address	I2C Byte Address	bit[31:24]	bit[23:16]	bit[15:8]	bit[7:0]
0	0	0x00	reserved	reserved	command status	command
0	1	0x04	history packet number[1]	history packet number[0]	history packet address [1]	history packet address [0]
0	2	0x08	rtcDayName	rtcMonth	rtcDay	rtcYear
0	3	0x0c	reserved	rtcHour	rtcMin	rtcSec
0	4	0x10	tagVersion	reserved	empID[1]	empID[0]
0	5	0x14	battery voltage	interrupt status	reserved	reserved
0	6	0x18	rtcDayName	rtcMonth	rtcDay	rtcYear
0	7	0x1c	command	rtcHour	rtcMin	rtcSec
0	8	0x20	command status	interrupt status	empID1[1]	empID1[0]
0	9	0x24	rtcDayName	rtcMonth	rtcDay	rtcYear
0	10	0x28	command	rtcHour	rtcMin	rtcSec
0	11	0x2c	command status	interrupt status	empID[1]	empID[0]
....	....	....	....	....	....	....
15	167	2028	reserved	reserved	reserved	reserved
15	168	2032	reserved	reserved	reserved	reserved
15	169	2036	reserved	reserved	reserved	reserved
15	170	2040	reserved	reserved	reserved	reserved
15	171	2044	reserved	reserved	reserved	reserved

Note: EEPROM addresses: 2040 – 2047, and 0x0014 – 0x0017 are strictly runtime reserved!

2. RF Host / Microcontroller Interface

The microcontroller RF commands are interrupt-driven and are executed by the microcontroller when the RF host is no longer busy writing into the EEPROM, and a valid command-status is available. Under this condition, the microcontroller proceeds to read the command from the EEPROM as follows:

a. Microcontroller ignores RF activity until command-received status is available
b. Microcontroller then reads the RF command
c. Microcontroller updates the Command Status in this order:
    i. Command Running Status
    ii. Command Complete Status. If an error is detected, the Command Status will contain the error code instead of the command-complete status.
    iii. Microcontroller updates the Interrupt Status only should a tamper condition be detected.

3. RF Host Command Protocol. The RF Host initiates an RF command as follows:

<u>RF Host command sequence</u>:

a. RF Host writes the ***command and command-status*** at EEPROM RF block address 0 with the following 32-bit value:

bit[31:24] :	0xff
bit[23:16]:	0xff
bit[15:8]:	***command-status*** (see Table 3: Command Status)
bit[7:0]:	***command*** (see Table 2: RF Host Commands)

b. The Microcontroller ignores any command until the RF host is done initializing any data required by the command. The Microcontroller executes the command only when the RF host has changed the command-status to command-received value. Refer to Table 3: Command Status
c. RF Host then waits for the command to complete by reading the same RF block address 0, and inspecting the ***command status*** bit[15:8] for command-complete status or error code. The microcontroller updates the ***command status*** and ***interrupts status*** bytes accordingly as it runs and completes the ***command*** successfully or with the corresponding error code. Refer to Table 5.

4. RF Host Commands. The microcontroller implements the following RF commands as indicated in Table 2: RF Host Commands.

Table2: RF Host Commands

command (**bit[7:0]**)	Description
0x00	NO-OP. Microcontroller returns gracefully.
0x01	**Read RTC**. RTC data will be valid as long as the RF Host has explicitly called ***Start Timers*** with Calendar RTC data as defined in the memory definition.
0x02	**Start Timers**. This command allows the microcontroller to ***configure*** the Calendar RTC with the data supplied by the RF Host. Without RTC configuration, the Tamper Band interrupt is ignored. Once this command is successfully executed, the Tamper Band interrupt will be engaged thereafter. By the same token, the microcontroller will return a failure if this command is called and the Tamper Band is open. To successfully configure the Calendar RTC, the Tamper Band must be closed.
0x03	**Stop Timers.** This command stops the Calendar RTC and interrupts. This command must be called first before an authorized user can open the Tamper Band. History Data will be first updated by the microcontroller before proceeding to stopping the Calendar RTC and interrupts.
0x04	**Read History**. This command returns an error if a Start Timers has not been issued. Otherwise, History Data will be updated.
0x05	**Flush History**. This command returns an error if a Start Timers or a Read History has not been issued, respectively. Else, History Data will be erased with 0xff.
0x06	**Read Battery Voltage**. A low-level (**0x01**) or normal-level (**0x00**) indication will be returned. If a low-level is detected, the microcontroller will shut down operation until a new battery is installed.
0x07	**Tamper Switch Test.** This is a test command to test the tamper switch. Under this command, there will be no history updates. To run this command, the RF host must first call Stop Timers if already configured, otherwise, opening the band will cause a real tamper.
0x08	**Factory Data Reset.** This command can be called any time, when called, it will simply reset the runtime area of the EEPROM to initial conditions, including real-time information about battery condition and tamper switch band status.
0xaa	**Tamper Interrupt Command.** *Command History* is updated with 0xaa to denote a Tamper Interrupt
0xbb	**Battery Interrupt Command.** *Command History* is updated with 0xbb to denote a Battery Interrupt

5. Command Status

Table3: Command Status

command status (**bit[15:8]**)	Description
0x01	Command Busy
0x02	Command Busy Acknowledge
0x03	Command Received
0x04	Command Running
0x00	Command has completed successfully
(-1) to (-11)	Error Code: See Table 4: Error Codes
0xff	Command status unknown

6. Command Status: Error Codes

Table4: Error Codes

Error Codes (**bit[15:8]**)	Description
0 (0x00)	No error
-1 (0xff)	Flush History error if Read History not done first
-2 (0xfe)	Tamper Band is open while configuring Calendar RTC
-3 (0xfd)	RTC Configuration is required
-4 (0xfc)	Invalid Command
-5 (0xfb)	Invalid Command Status
-6 (0xfa)	I2C Bus Busy
-7 (0xf9)	I2C Master Mode Select Event Failure
-8 (0xf8)	I2C Master Transmitter Mode Event Failure
-9 (0xf7)	I2C Master Byte Transmitting Event Failure
-10 (0xf6)	I2C Master Byte Transmitted Event Failure
-11 (0xf5)	I2C Master Receiver Mode Selected Event Failure

7. Interrupt Status

Table5: Interrupt Status

interrupt status (bit[23:16])	Description
0x00	Tamper Band Closed
0x02	Tamper Band Opened