AN205129

F²MC-8FX Family MB95410H/470H Series 8-Bit Microcontroller One Phase Power Meter (RN8209) Solution IC Card and ESAM Operation

Associated Part Family: MB95410H/470H Series

This application note describes how to use One Phase Power Meter (RN8209) solution's IC Card and ESAM function.

1 Introduction

This application note describes how to use One Phase Power Meter (RN8209) solution's IC Card and ESAM function.

Section 2 explains the Background.

Section 3 explains the HW Diagram.

Section 4 explains the HW Reference SCH.

Section 5 explains the FW Diagram.

Section 6 explains the FW Function List.

2 Background

Background of Prepayment Function

2.1 Overview

Prepayment function is realized by the communication between IC card and ESAM module.

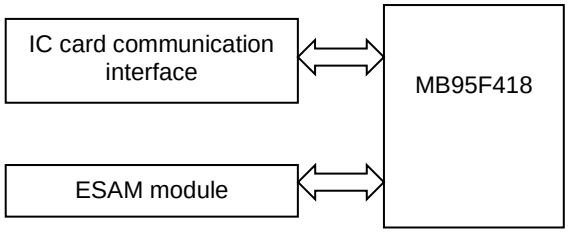
2.2 Communication Protocol

Byte communication protocol refers to ISO7816-3-2006 protocol, and frame structure refers to TimeCOS_PBOC protocol.

3 HW Diagram

3.1 HW Diagram of Prepayment Function

Figure 1. Hardware Diagram



4 HW Reference SCH

Hardware reference SCH of prepayment system

Figure 2. IC Card Interface Circuit

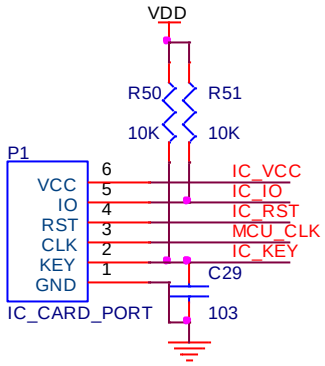
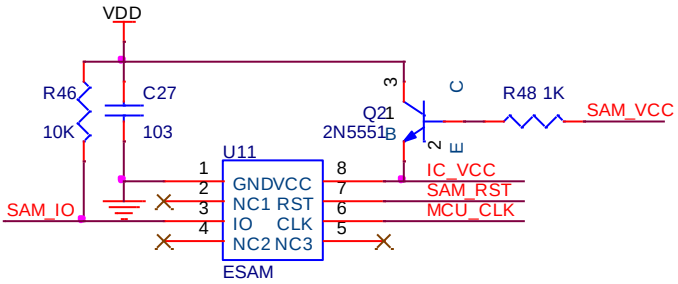


Figure 3. ESAM Module Circuit



5 FW Diagram

Firmware system diagram of AMR system

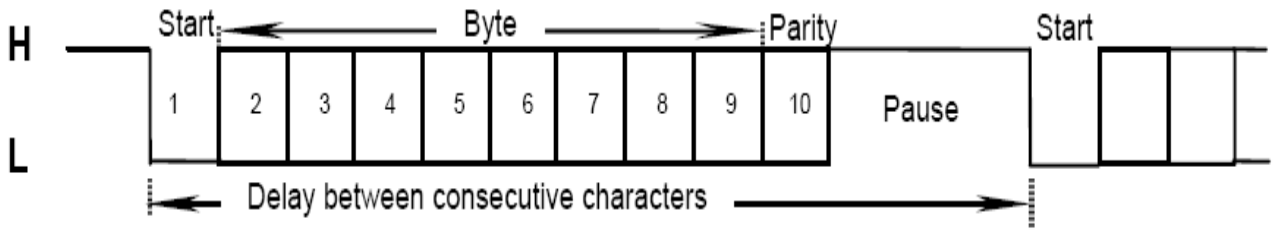
5.1 ISO7816-3-2006 Protocol

Communication protocol's character frame is shown in Figure 4.

According to Figure 4, a character consists of ten consecutive moments numbered 1 to 10. Each moment is either at state H or at state L.

- Before moment 1, the electrical circuit I/O shall be at state H.
- Moment 1 shall be at state L. It is the character start.
- Moments 2 to 9 shall encode a byte according to a coding convention (send the LSB first).
- Moment 10 shall encode the character parity.
- After moment 10, both the card and the interface device shall remain in reception mode (in error-free operation) for a certain time of "pause", so that the electrical circuit I/O remains at state H.

Figure 4. Character Format Data



5.2 TimeCOS_PBOC Protocol

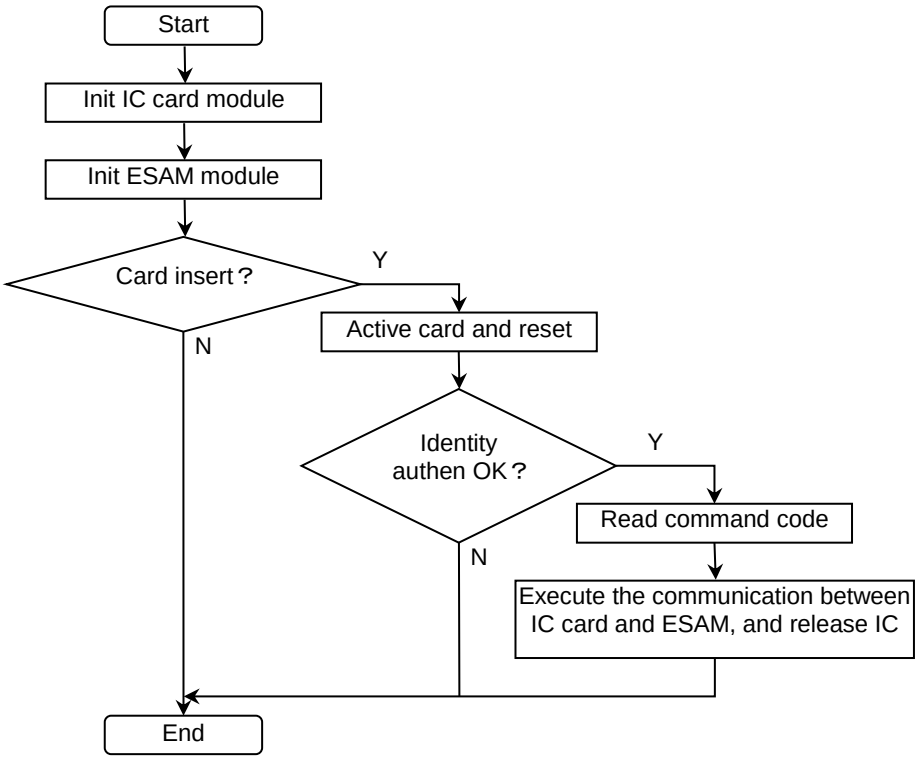
Figure 5. Command and Response Frame Format

CLA	INS	P1	P2	Lc	DATA	Le

DATA in Le byte	SW1	SW2

5.3 Firmware System Diagram

Figure 6. Firmware System Diagram



6 FW Function List

6.1 API

Table 1. FW API List

Function Prototype	Description
void IC_Card_Init(void)	Initialization for IC card connection
INT8U active_card(INT8U* receivebuffer)	Active IC card
void release_card(void)	Release IC card
INT8U user_card_process(void)	Process user card
INT8U user_card_01_process(void)	Process build-user card
INT8U user_card_02_03_process(void)	Process purchase card and complementary card
INT8U key_card_process(INT8U cardtype)	Process key card
INT8U cur_para_set_card_process(void)	Process current parameter set card
INT8U esam_data_copy_card_process(void)	Process ESAM data copy card
INT8U para_preset_card_process(void)	Process parameter preset card
INT8U meter_num_set_card_process(void)	Process meter number set card
INT8U increase_charge_card_process(void)	Process increase charge card
INT8U relay_test_card_process(void)	Process relay test card
INT8U sys_identity_authen(void)	Identity authen between IC card and ESAM
INT8U card_process(void)	Process card
void Esam_Init(void)	Initialization for ESAM connection
void ActiveEsam(void)	Active ESAM module
void ReleaseEsam(void)	Release ESAM module
void GetUserNum(INT8U* buffer)	Get user number from ESAM
void GetMeterNum(INT8U* buffer)	Get meter number from ESAM
void UpdateEsamBalance(INT8U decValue)	Update ESAM balance
void GetEsamBalance(void)	Get ESAM balance

6.2 HAL

Table 2. FW HAL List

Function Prototype	Description
void CardPortResetRx(void)	Reset IC card RX procedure to RX standby mode
void CardPortResetTx(void)	Reset IC card TX procedure to TX standby mode
INT8U IsCardPortDataReady(void)	Check if any data received from IC card port
INT8U CardPortDataRead(void)	Retrieve a data byte from FIFO of IC card port
INT8U IsCardPortTxBusy(void)	Check if card port is busy in the middle of TX
INT8U IC_SAM_send_data(INT8U mode, INT8U length, INT8U* buffer)	Send data to IC card or ESAM
INT8U IC_SAM_receive_data(INT8U mode, INT8U length, INT8U* buffer)	Receive data from IC card or ESAM
void IC_SAM_send_byte(INT8U mode, INT8U data)	send a byte data to IC card or ESAM
void EsamPortResetRx(void)	Reset Esam procedure to RX standby mode
void EsamPortResetTx(void)	Reset Esam procedure to TX standby mode
INT8U IsEsamPortDataReady(void)	Check if any data received from ESAM port
INT8U EsamPortDataRead(void)	Retrieve a data byte from FIFO of ESAM port
INT8U IsEsamPortTxBusy(void)	Check if ESAM port is busy in the middle of TX

7 Additional Information

For more information on Cypress products, visit our website:

http://www.cypress.com/cypress-microcontrollers

Document History

Document Title: AN205129 – F²MC-8FX Family MB95410H/470H Series 8-Bit Microcontroller One Phase Power Meter (RN8209) Solution IC Card and ESAM Operation

Document Number: 002-05129

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	HUAL	06/01/2011	Initial release
*A	5261724	HUAL	06/15/2016	Migrated Spansion Application Note MCU-AN-500112-E-10 to Cypress format.
*B	5848917	MALI	08/09/2017	Updated logo and copyright

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2011-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.