RMB–Series
Read Me File

for Digital I/O, Odometer, Com4, and Power Management

CONTACT INFORMATION
Front Desk: 1–303–430–1500
Technical Support: 1–303–426–4521
FastHelp@octagonsystems.com
www.octagonsystems.com

Table of Contents

List of Tables

Overview

This document contains programming information for the RMB–C1 digital I/O, Odometer input, COM4, and Power Management functions.

Digital I/O, Odometer input and COM4 are accessed through registers in an on-board FPGA. Power Management is accessed through a Micro Controller, which is hardwired to the internal COM2 port.

FPGA Functions (Digital I/O, Odometer input and COM4)

The FPGA provides three functions. The first function is the digital input and digital output function. The second function is the counter and odometer function. The third function is the COM4 RS–232/485 and termination control.

These functions can be controlled by reading and writing to the FPGA in the ISA bus address space. The ISA memory range that the FPGA occupies is set by modifying the PCS2 address and PCS2 width BIOS Setup configuration fields. The value entered in the PCS2 address field is the base address for the FPGA. These fields are found in the BIOS Custom Configuration Setup Screen. The PCS2 address can be set to any address that allows a PCS2 width of 16 bytes. For example, if the PCS2 address is set to 0x140 and the PCS2 Width is set to 16, the FPGA can be accessed by performing IO reads and writes from 0x140 through 0x14f.

The FPGA internal address space is broken up into three banks. Each bank consists of 16 8-bit bytes. The bank select register controls the bank that is being accessed. This register is set by writing a value of 0 to 2 out to base + offset 0x0f. This register is accessible in all three banks of the FPGA address space.

The information about the various registers is listed in tables 1, 2 and 3. Table 1 shows all the registers that exist in bank zero. Table 2 shows all the registers that exist in bank one. Table three shows all the registers that exist in bank two.

There are two types of registers. The first type of register is the configuration registers. The second type of register is the operation registers. The data in table one, table two and table three shows both the required state of the function select register and the register type.

Table 1 Bank 0 registers

Address Offset	Function	Access	Description
0x00	Digital Output FUNCSLE = 0x03	Write only	OPERATIONAL Four bits of digital output 3:0
0x01	Digital Input FUNCSEL N/A	Read only	OPERATIONAL Seven bits
0x02	IRQ Edge Select	Write only	Defines the edge that generates an interrupt if routed to a counter. Set to 1 is positive edge Set to 0 is negative edge
0x03	Not Used		
0x04	Not Used		
0x05	Com Control FUNCSEL = 0x02	Write Only	CONFIGURATION Bits 1:0 Defines RS-232/ RS422 and RS-422 termination
0x06	Not Used		
0x07	Not Used		
0x08	Not Used		
0x09	Not Used		
0x0a	Counter Zero Input route FUNCSEL = 0x02	Write only	CONFIGURATION Bits 7:0, when set to one will route to the counter0 device
0x0b	Counter One Input route FUNCSEL = 0x02	Write only	CONFIGURATION Bits 7:0, when set to one will route to the counter0 device
0x0c	Counter Two Input route FUNCSEL = 0x02	Write only	CONFIGURATION Bits 7:0, when set to one will route to the counter0 device
0x0d	Counter Three Input route FUNCSEL = 0x02	Write only	CONFIGURAITON Bits 7:0, when set to one will route to the counter0 device
0x0e	Digital Input Mode Select FUNCSEL = 0x02	Write only	CONFIGURATION Bits 7:0, when set input is in IRQ mode and counter is active When clear bit is input mode, counter is disabled
0x0f	Bank Select Register FUNCSEL N/A	Read Write	OPERATIONAL Selects internal register bank number

Table 2 Bank 1 registers

Address Offset	Function	Access	Description
0x00	Not Used		
0x01	Prescalar Low Byte FUNCSEL = 0x02	Write only	CONFIGURATION Seven bits, defines counter or odometer prescale value
0x02	Prescalar High Byte FUNCSEL = 0x02	Write Only	CONFIGURATION Defines counter or odometer prescale value
0x02	IRQ Edge Select FUNCSEL = 0x02	Read	CONFIGURATION Returns Irq edge select register value
0x03	Irq Source Register FUNCSEL N/A	Read Write	OPERATIONAL Returns source of interrupt for counter and for odometer Bits 3:0 are counter 3:0 Bit 4 is odometer A write clears the IRQ and the counter and odometer counts
0x04	Function Select Register FUNCSEL N/A	Read Write	OPERATIONAL Defines the operating mode of device Bits 3:0 0010 config mode 0011 run mode 1111 reset device
0x05	Load counter 0 count FUNCSEL N/A	Write	OPERATIONAL When Funcsel = 0x05 a write will load counter0 count value into the readcount mux register
0x06	Load counter 1 count FUNCSEL N/A	Write	OPERATIONAL When Funcsel = 0x05 a write will load counter1 count value into the readcount mux register
0x07	Load counter 2 count FUNCSEL N/A	Write	OPERATIONAL When Funcsel = 0x05 a write will load counter2 count value into the readcount register
0x07	Device capabilities register FUNCSEL = 0x02	Read	CONFIGURATION Device capability register Bits 2:0 number of counters supported Bit 3 odometer Set : supported Clear: no odometer Bits 6:4 FPGA revision level
0x08	Load counter 3 count	Write	OPERATIONAL When Funcsel = 0x05 a write will load counter3 count value into the readcount mux register

0x09	Load odometer count FUNCSEL N/A	Write	OPERATIONAL When Funcsel = 0x05 a write will load odometer count value into the readcount mux register
0x0a	Counter Zero Input route FUNCSEL = 0x02	Write only	CONFIGURATION Bits 7:0, when set to one will route to the counter0 device
0x0b	Counter One Input route FUNCSEL = 0x02	Write only	CONFIGURATION Bits 7:0, when set to one will route to the counter0 device
0x0c	Counter Two Input route FUNCSEL = 0x02	Write only	CONFIGURATION Bits 7:0, when set to one will route to the counter0 device
0x0d	Counter Three Input route FUNCSEL = 0x02	Write only	CONFIGURATION Bits 7:0, when set to one will route to the counter0 device
0x0d	Counter count value Low byte FUNCSEL N/A	Read only	CONFIGURATION Reads the counter count value loaded into the readcount mux register
0x0e	Counter count value High byte FUNCSEL N/A	Read only	CONFIGURATION Reads the counter count value loaded into the readcount mux register
0x0f	Bank Select Register FUNCSEL N/A	Read Write	OPERATIONAL Selects internal register bank number

Table 3 Bank 2 registers

Address Offset	**Function**	**Access**	**Description**
0x00	Not Used		
0x01	Not Used		
0x02	Counter 0 input route FUNCSEL N/A	Read Only	OPERATIONAL Returns the input route for counter 0
0x03	Counter 1input route FUNCSEL N/A	Read Only	OPERATIONAL Returns the input route for counter 1
0x04	Not Used		
0x05	Counter 2 input route FUNCSEL N/A	Read Only	OPERATIONAL Returns the input route for counter 2
0x06	Counter3 input route FUNCSEL N/A	Read Only	OPERATIONAL Returns the input route for counter 3
0x06	Write prescale to count0 FUNCSEL N/A	Write only	CONFIGURATION Writes the prescale to counter 0
0x08	Write prescale to count1 FUNCSEL N/A	Write only	CONFIGURATION Writes the prescale to counter 1
0x09	Write prescale to count2	Write only	CONFIGURATION Writes the prescale to counter 2

	FUNCSEL N/A		
0x07	Write prescale to count3 FUNCSEL N/A	Write only	CONFIGURATION Writes the prescale to counter 3
0x0a	Write prescale to odm FUNCSEL N/A	Write only	CONFIGURATION Writes the prescale to odm
0x0b	Not Used		
0x0c	Not Used		
0x0d	Not Used		
0x0e	Not Used		
0x0f	Bank Select Register FUNCSEL N/A	Read Write	CONFIGURATION Selects internal register bank number

Digital Input and Output

The digital input and output functions use registers that are in both bank zero and in bank one. There are two data access registers used to read the digital inputs and to write the digital outputs. There are two configuration registers used to set up the digital inputs. These configuration registers are the *input mode select* register and the *function select* register.

The *input mode select* register is used to either route the digital input pin directly to the ISA data bus or route the digital input pin to one of four counters that exist in the FPGA. The *input mode select* register is at base + offset 0x0e. To write this register the *bank select* register must be set to zero and the *function select* register must be set to 0x02.

The *function select* register is used to control the operating mode of the FPGA. The *function select* register is located in bank 1 at base address + offset 0x04. This is a read and a write register and is detailed in table 2.

The *digital input* register is accessed by reading base address + offset 0x01. This register can be accessed regardless of what bank is selected in the bank select register. The *digital input* register can also be read regardless of the value written to the *function select* register.

The *digital output* register can only be accessed at base address + offset zero when the *bank select* register has been set to zero. The *function select* register must also be set to run mode (0x03) to write data to the digital output pins.

Digital Input/Output registers

DIGITAL OUTPUT: BASE + 0x00 : FUNCSEL = 0x03 : BANKSEL = 0x00

DIGITAL INPUT : BASE + 0x01 : FUNCSEL = NA : BANKSEL = 0,1,2

DIMODESEL : BASE + 0x0e : FUNCSEL = 0x02 : BANKSEL = 0x00

Counter and Odometer

The FPGA contains four counters and one odometer. The counters can use as their input source any of the digital input bits. The odometer uses as its' input source the odometer input and does not map to any of the digital inputs.

The counters and odometer can count events on either a rising edge or a falling edge of their input signals. The counter outputs are combined to produce one interrupt to the ISA bus. This interrupt is at IRQ7. If none of the counters are enabled this IRQ line will be High Z making IRQ7 available for other devices that might be on the ISA bus.

The odometer output will generate IRQ5. If the odometer is disabled this IRQ line will be High Z making IRQ5 available for other devices that might be on the ISA bus.

The counters and the odometer each support a 16-bit prescale value. This value will divide the input count by 2**(n) where n represents a bit set in the prescale register for the device being configured. The divisors for the counters and odometer are shown in table 4.

Table 4 *Counter/Odometer Prescale Values*

PRESCALE VALUE (HEX)	COUNT (DECMAL)	PRESCALE VALUE (HEX)	COUNT (DECMAL)
0x8000	32,768	0x0040	64
0x4000	16384	0x0020	32
0x2000	8192	0x0010	16
0x1000	4096	0x0008	8
0x0800	2048	0x0004	4
0x0400	1024	0x0002	2
0x0200	512	0x0001	1
0x0100	255	0x0000	20
0x0080	128		
NOTE: NO OTHER VALUES ARE SUPPORTED			

The counters will count input events and generate an IRQ7 event when the prescale count is reached. The odometer will multiply the odometer input by the (prescale count x 20) and generate an IRQ5 event when that multiple is reached. For example, if the odometer prescale is set to one when the odometer sees 20 input transitions it will generate an IRQ5 on the ISA bus, or if the prescale is set to two when the odometer sees (20 x 2) input transitions an IRQ5 event will be generated on the ISA bus.

Counter and Odometer setup

The counters and odometers use registers that are located in bank0, bank1, and bank2. These registers are described in tables 1, 2 and 3. The counters and odometer prescale values are loaded using the same process. The counters have two additional steps that route a digital input bit to a counter and set the edge detect to rising or falling.

To configure a counter or an odometer the sequence below must be followed. An example listing the registers follows the steps.

1. Make sure that the *digital input mode select* register bit value is set to zero for the counter being configured.

2. Load the prescale low byte and high byte for the counter to be used.

3. Load the 16-bit prescale value to the counter being configured.

4. Set the input route byte for the counter being configured to route an input to the counter. (COUNTER ONLY)

5. Set the edge detect to rising or falling in the edge select register. (COUNTER ONLY)

6. Enable the counter by setting the appropriate bit in the digital input mode select bit to one.

COUNTER/ ODOMETER PRESCALE REGISTERS
Refer to the steps above for procedure.

STEP ONE: Set digital input mode select bit to zero
DIMODESEL BYTE : BASE + 0x0e : FUNCSEL = 0x02 : BANKSEL = 0x00
DIMODESEL REG : BITS 7:5 unused, always set to zero
BIT 4 : SET : ODOMETER ENABLED
CLEAR : ODOMETER DISABLED.
COUNT IS CLEARED
IRQ HIZ
BITS 3:0 map to counter 3:0
WHEN COUNTER BIT IS SET COUNTER IS ENABLED
WHEN COUNTER BIT IS CLEAR COUNTER IS DISABLED
COUNT IS CLEARED
IRQ FOR COUNTER IS CLEARED

STEP TWO: Write prescale values (see *Table 5*)
PRESCALE LOW BYTE : BASE + 0x00 : FUNCSEL = 0x02 : BANKSEL = 0x01
PRESCALE LOW BYTE : BASE + 0x01 : FUNCSEL = 0x02 : BANKSEL = 0x01

STEP THREE: Load count to counter
COUNTER LOAD BYTE : BASE + 0x06 to 0x0a: FUNCSEL = N/A BANKSEL = 0x02

1. Change to bank2
2. Write any value to counter load register

Table 5 *Counter Prescale Load*

Register Description	Register offset	Register Bank	Register Description
Counter zero load	Base + 0x06	BANK2	Counter zero load
Counter one load	Base + 0x07	BANK2	Counter one load
Counter two load	Base + 0x08	BANK2	Counter two load
Counter three load	Base + 0x09	BANK2	Counter three load
Odometer load	Base + 0x0a	BANK2	Odometer load

STEP FOUR: Set input route byte for counter

- The Function select register must be set to 0x02
- The bank select register must be set to 0x00
- These registers are write only

Table 6 *Counter Input Route*

Register Description	Register offset	Bit description
Counter Zero input map	Base + 0x0a	Bits 7:0 = input 7:0 Bit set, route input to counter Bit clear, input not routed to counter
Counter One input map	Base + 0x0a	Bits 7:0 = input 7:0 Bit set, route input to counter Bit clear, input not routed to counter
Counter Two input map	Base + 0x0a	Bits 7:0 = input 7:0 Bit set, route input to counter Bit clear, input not routed to counter
Counter Three input map	Base + 0x0a	Bits 7:0 = input 7:0 Bit set, route input to counter Bit clear, input not routed to counter

STEP FIVE: Set the edge detect register

- Register is write only

EDGEDETECT BYTE : BASE + 0x02 : FUNCSEL = 0x02 : BANKSEL = 0x00

Table 7 *Edge Detect Register*

Bit number	Description
0	Set to zero falling edge Set to one rising edge
1	Set to zero falling edge Set to one rising edge
2	Set to zero falling edge Set to one rising edge
3	Set to zero falling edge Set to one rising edge
4	Not used, set to zero
5	Not used, set to zero
6	Not used, set to zero
7	Not used, set to zero

STEP SIX: Set the input mode select register

- Register is read/write

DIMODESEL BYTE : BASE + 0x0e : FUNCSEL = 0x02 : BANKSEL = 0x00

Table 8 *Input Mode Register*

Bit number	Description
0	Set to zero disable counter zero Set to one enable counter zero
1	Set to zero disable counter one Set to one enable counter one
2	Set to zero disable counter two Set to one enable counter two
3	Set to zero disable counter three Set to one enable counter three
4	Set to zero disable odometer Set to one enable odometer
5	Not used, set to zero
6	Not used, set to zero
7	Not used, set to zero

Com Port Termination and Control

There are two COM port control functions in the FPGA. These are the *com port control* function and the *com port termination* function. The com port control function can be used to configure the COM4 port in the system to be either RS-232 (default) or RS-485. The com port termination function can be used to terminate COM4 when it is configured as a RS-485 port

COM CONTROL BYTE : BASE + 0x05 : FUNCSEL = 0x02 : BANKSEL = 0x00
BIT 0 : SET; RS-422 Termination enabled
BIT 0 : CLEAR; RS-422 Termination disabled
BIT 1 : SET COM 4 RS-422 enabled
BIT 1 : CLEAR; COM 4 RS-422 disabled

Micro Controller User Interface (Power Management)

The Micro Controller on the interface card can be accessed by using the RMB –C1 motherboard COM 2 serial port. The Micro Controller serial port is configured with the settings of 9600 baud, 8 bit data, no parity, and 1 stop bit. The Micro Controller is the slave device and never initiates communication. All communications are initiated by the host computer master device.

This serial interface can be used to enable the power management functions, disable the power management functions, read the voltage level of analog inputs connected to the Micro Controller, read a value from a device connected to the external I2C bus, and write a value to a device connected to the external I2C bus.

A disable command will stop all power management control functions in the Micro Controller except the battery low function. An enable command will start the power management control functions in the Micro Controller. A read analog input register will return the POWER_FIL, the POWER_EXT, or the 5V_UC value from the analog inputs of the Micro Controller.

The POWER_FIL input is the battery voltage. The POWER_EXT is the external system power voltage. The 5V_UC is the 5V main power from the RMB–C1 motherboard. The I^2C read and write commands support 7-bit addresses only at this time and can be used to read or write to external I^2C devices.

Serial Interface Applications Program Interface (API)

The serial interface API consists of five commands. These commands are listed below. The values are shown on the following page.

- Command 0 : Disable Power Management
- Command 1 : Enable Power Management
- Command 2 : Read analog input register
- Command 3 : Not used at this time
- Command 4 : Read I^2C device
- Command 5 : Write I^2C device

The serial API commands take the general format:

From HOST

BYTE 0	: START COMMAND (0x01)
BYTE 1	: COMMA SEPERATOR (0x44)
BYTE 2	: COMMAND (0,1,2,4,5)
BYTE 3	: COMMA SEPERATOR
BYTE 4 to (n-1)	: COMMAND DEPENDANT
BYTE n	: END OF TEXT (0x04)

Each command will result in a response string from the Micro Controller. The response string is dependant on the command sent.

Status returns are:

- 0x00 : Command successful
- 0x01 : Unknown Command
- 0x02 : Unable to enable power management
- 0x03 : Unable to disable power management
- 0x04 : Read register failed
- 0x05 : Not used
- 0x06 : I2C Read Fail
- 0x07 : I2C Write Fail
- 0x08 : I2C ACK failed
- 0x09 : Micro Input Buffer Overflow

COMMAND 0: DISABLE POWER MANAGEMENT

- COMMAND STRING
- START,0x00,EOT

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4
0x01	0x44	0x00	0x44	0x04

- MICRO CONTROLLER RESPONSE
- ACK,STATUS,EOT

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4
0x06	0x44	STATUS	0x44	0x04

COMMAND 1: ENABLE POWER MANAGEMENT

- COMMAND STRING
- START,0x01,EOT

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4
0x01	0x44	0x01	0x44	0x04

- MICRO CONTROLLER RESPONSE
- ACK,STATUS,EOT

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4
0x06	0x44	STATUS	0x44	0x04

COMMAND 2: READ REGISTER

- COMMAND STRING
- START,0x02,Reg Addr,EOT

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE5	BYTE 6
0x01	0x44	0x02	0x44	address	0x44	0x04

- MICRO CONTROLLER RESPONSE
- ACK,STATUS,data,EOT

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
0x06	0x44	STATUS	0x44	High Byte	Low Byte	0x44	0x04

COMMAND 4: READ I²C device register (seven bit address support at this time)

- COMMAND STRING
- START,0x04,DevAddr,Reg Addr,EOT

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	BYTE 8	BYTE 9
0x01	0x44	0x04	0x44	High Byte Slave Addr	Low Byte Slave Addr	0x44	Data Addr	0x44	0x04

- MICRO CONTROLLER RESPONSE
- ACK,STATUS,DATA,EOT

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 7	BYTE 8	BYTE 9
0x01	0x44	0x04	0x44	DATA	0x44	0x04

COMMAND 5: WRITE I²C device register (seven bit address support at this time)

- COMMAND STRING
- START,0x05,DevAddr,Reg Addr,Data,EOT

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	BYTE 8	BYTE 9	BYTE 10	BYTE 11
0x01	0x44	0x05	0x44	High Byte Slave Addr	Low Byte Slave Addr	0x44	Data Addr	0x44	DATA	0x44	0x04

- MICRO CONTROLLER RESPONSE
- ACK,STATUS, EOT

BYTE 0	BYTE 1	BYTE 2	BYTE 8	BYTE 9
0x01	0x44	0x04	0x44	0x04