# RUBY-MM-1616A Family

## PC/104 / PC/104-*Plus* I/O Modules with 4, 8 or 16 16-Bit Analog Outputs

User Manual Rev A.3 May 2014

| Revision | Date | Comment |
|---|---|---|
| A.3 | 5/12/2014 | Initial release |

***FOR TECHNICAL SUPPORT***
***PLEASE CONTACT:***
**support@diamondsystems.com**

# TABLE OF CONTENTS

# 1. IMPORTANT SAFE HANDLING INFORMATION

## WARNING!

### ESD-Sensitive Electronic Equipment

Observe ESD-safe handling procedures when working with this product.

Always use this product in a properly grounded work area and wear appropriate ESD-preventive clothing and/or accessories.

Always store this product in ESD-protective packaging when not in use.

### *Safe Handling Precautions*

This board contains several I/O and bus connectors with many connections to sensitive electronic components. This creates many opportunities for accidental damage during handling, installation and connection to other equipment. The list here describes common causes of failure found on boards returned to Diamond Systems for repair. This information is provided as a source of advice to help you prevent damaging your Diamond (or any vendor's) embedded computer boards.

**ESD damage** – This type of damage is usually almost impossible to detect, because there is no visual sign of failure or damage. The symptom is that the board eventually simply stops working, because some component becomes defective. Usually the failure can be identified and the chip can be replaced. To prevent ESD damage, always follow proper ESD-prevention practices when handling computer boards.

**Damage during handling or storage** – On some boards we have noticed physical damage from mishandling. A common observation is that a screwdriver slipped while installing the board, causing a gouge in the PCB surface and cutting signal traces or damaging components.

Another common observation is damaged board corners, indicating the board was dropped. This may or may not cause damage to the circuitry, depending on what is near the corner. Most of our boards are designed with at least 25 mils clearance between the board edge and any component pad, and ground / power planes are at least 20 mils from the edge to avoid possible shorting from this type of damage. However these design rules are not sufficient to prevent damage in all situations.

A third cause of failure is when a metal screwdriver tip slips, or a screw drops onto the board while it is powered on, causing a short between a power pin and a signal pin on a component. This can cause overvoltage / power supply problems described below. To avoid this type of failure, only perform assembly operations when the system is powered off.

Sometimes boards are stored in racks with slots that grip the edge of the board. This is a common practice for board manufacturers. However our boards are generally very dense, and if the board has components very close to the board edge, they can be damaged or even knocked off the board when the board tilts back in the rack. Diamond recommends that all our boards be stored only in individual ESD-safe packaging. If multiple boards are stored together, they should be contained in bins with dividers between boards. Do not pile boards on top of each other or cram too many boards into a small location. This can cause damage to connector pins or fragile components.

**Power supply wired backwards** – Our boards are not designed to withstand a reverse power supply connection. This will destroy each IC that is connected to the power supply (i.e. almost all ICs). This type of damage is not covered under Diamond Systems' warranty. The board will most likely will be unrepairable and must be replaced. A chip destroyed by reverse power or by excessive power will often have a visible hole on the top or show some deformation on the top surface due to vaporization inside the package. **Check twice before applying power!**

**Overvoltage on bus connector** – The PC/104-Plus PCI connector on this product is compatible with 3.3V buses only. If the board is installed on a CPU module with 5V signaling on the PC/104-Plus connector, the FPGA on board that provides the PCI interface may be damaged. The PC/104 ISA bus connector is 5V tolerant and will work without problem in any valid PC/104 system.

**Overvoltage on analog output** – If an analog output is accidentally connected to another output signal or a power supply voltage, the output can be damaged. On most of our boards, a short circuit to ground on an analog output will not cause trouble.

**Overvoltage on digital I/O line** – If a digital I/O signal is connected to a voltage above the maximum specified voltage, the digital circuitry can be damaged. On most of our boards the acceptable range of voltages connected to digital I/O signals is 0-5V, and they can withstand about 0.5V beyond that (-0.5 to 5.5V) before being damaged. However logic signals at 12V and even 24V are common, and if one of these is connected to a 5V logic chip, the chip will be damaged, and the damage could even extend past that chip to others in the circuit.

**Bent connector pins** –This type of problem is often minor and mostly cosmetic, and is easily fixed by bending the pins back to their proper shape one at a time with needle-nose pliers. The most common cause of bent connector pins is when a PC/104 board is pulled off the stack by rocking it back and forth left to right, from one end of the connector to the other. As the board is rocked back and forth it pulls out suddenly, and the pins at the end get bent significantly. The same situation can occur when pulling a ribbon cable off of a pin header. However, if the pins are bent too severely or too often, bending them back can cause them to weaken unacceptably or even break, and the connector must be replaced.

# 2. MODELS AND FEATURES

## 2.1 Description

The Ruby-MM-1616 is a family of PC/104 I/O modules featuring 4, 8 or 16 16-bit analog voltage and current outputs, and 48 digital I/O lines. The module uses the Analog Devices AD5755 16-channel 16-bit DAC chip for the D/A outputs and an FPGA with level-shifting transceivers for the DIO lines. A 50-pin connector provides access to the 16 analog outputs, and another 50-pin connector provides access to the 48 DIO lines. The board operates over the industrial temperature range of -40$^{o}$C to +85$^{o}$C and is supported by Diamond Systems' Universal Driver software. Models are available with either PC/104 or PC/104-*Plus* I/O expansion buses.

Ruby-MM-1616 utilizes individual digital calibration for each analog output channel. The calibration is performed at the factory prior to shipment. All analog components are high accuracy and low drift, enabling the board to avoid the need for recalibration for many applications during its useful lifetime.

## 2.2 Features

**Board Features**

- 4, 8 or 16 analog outputs with 16-bit D/A resolution
- Programmable voltage output ranges: 0-5V, 0-10V, ±5V, ±10V, 0-20mA, 4-20mA, 0-24mA
- Independent output range for each channel
- Waveform generator on up to 16 channels
- Simultaneous update of any combination of channels
- Multi-channel simultaneous waveform output capability with up to 100KHz waveform update rate
- Output current limit +/-5mA per channel in voltage mode
- Autocalibration of D/A circuits using the internal offset and gain registers for each channel
- 40 byte-wide and 8 bit-wide digital I/O lines with programmable direction
- 2 32-bit programmable counter / timers
- 4 24-bit pulse width modulators
- PC/104 ISA 16-bit bus interface or PC/104-*Plus* (ISA + PCI) 3.3V 32-bit bus interface

**Software Support**

Diamond's Universal Driver software with functions including:

- D/A output
- D/A waveform generator
- Calibration
- Digital I/O – bit and byte-wide
- Digital I/O – PWM

**Mechanical, Electrical, and Environmental**

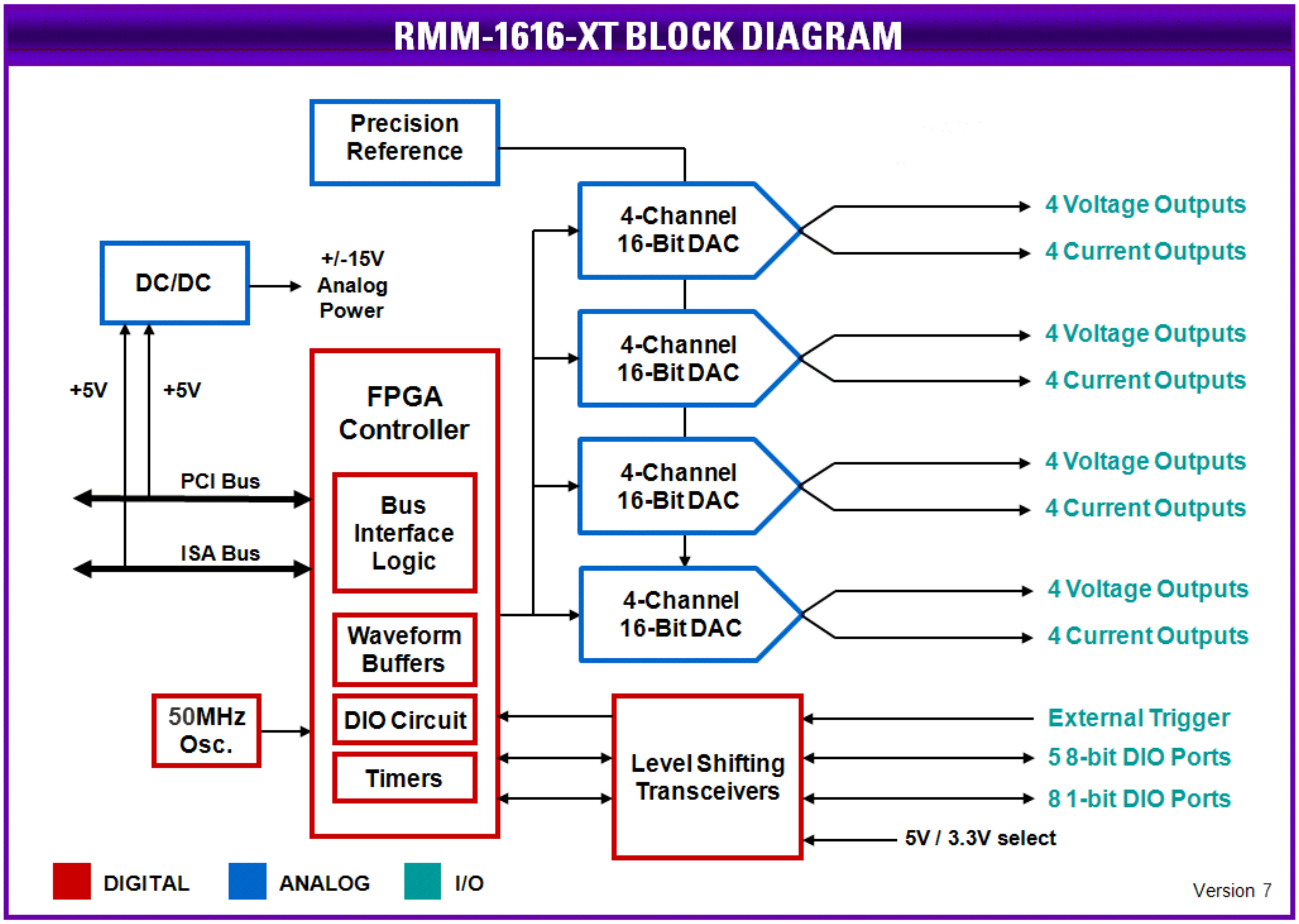
- PC/104 compliant form factor
- Dimensions: 3.550 x 3.775" (90 x 96mm)
- +5VDC input voltage
- -40$^{o}$C to +85$^{o}$C operating temperature
- RoHS compliant

## 2.3 Available Models

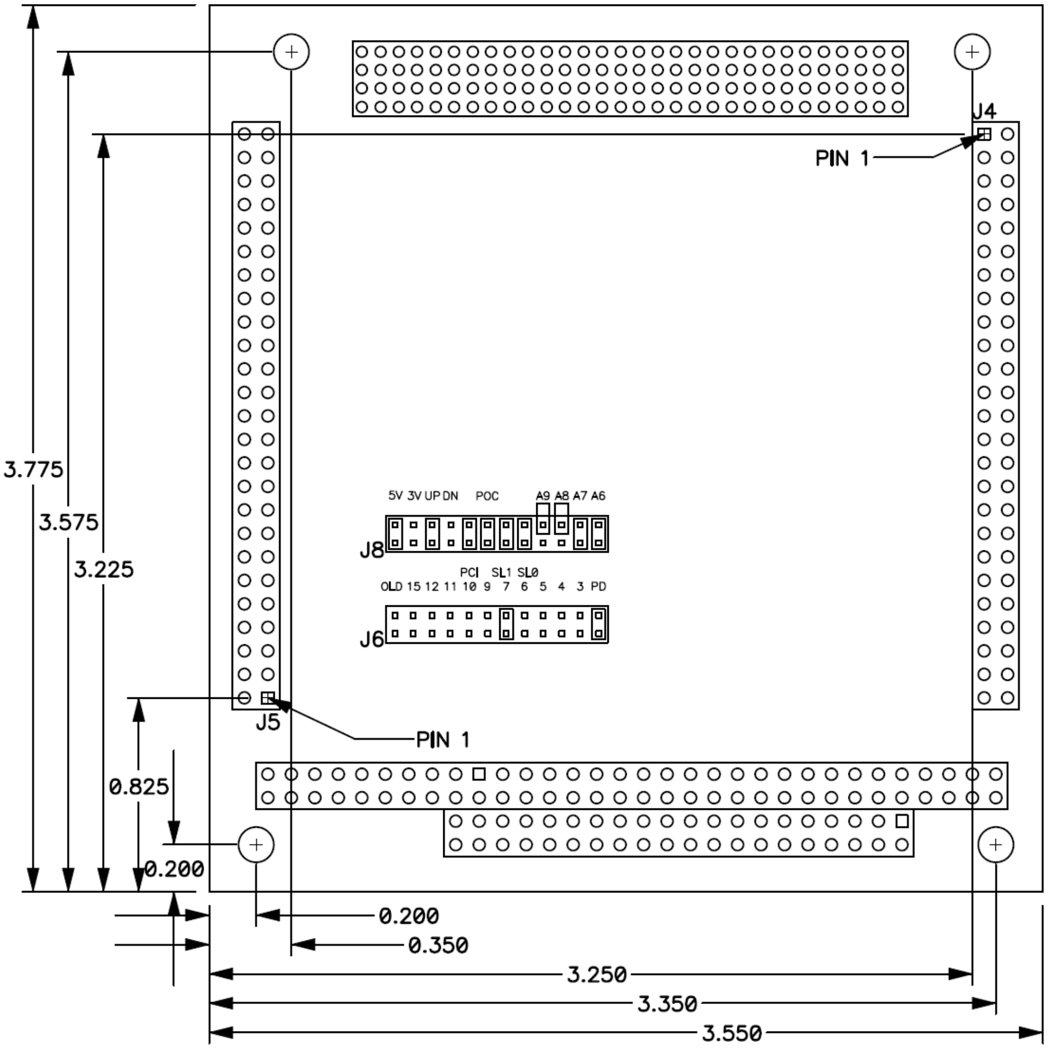
The following models are available, offering a selection of analog outputs for each bus configuration.

| Model Number | Analog Outputs | Digital I/O | Bus Interface | Notes |
|---|---|---|---|---|
| RMM-1616AP-XT | 16 | 48 | PCI + ISA | Standard stocked item |
| RMM-816AP-XT | 8 | 48 | PCI + ISA | Standard stocked item |
| RMM-416AP-XT | 4 | 48 | PCI + ISA | 50 unit minimum order |
| RMM-1616A-XT | 16 | 48 | ISA | Standard stocked item |
| RMM-816A-XT | 8 | 48 | ISA | Standard stocked item |
| RMM-416A-XT | 4 | 48 | ISA | 50 unit minimum order |

# 3. BLOCK DIAGRAM



# 4. BOARD DRAWING



| Connector | Description |
|---|---|
| J1 & J2 | PC/104 connectors |
| J3 | PCI connector |
| J4 | Analog connector |
| J5 | Digital connector |
| J6 | IRQ jumper block |
| J8 | Base Address jumper block |

Note: Connectors that are not intended for customer use are not identified or described in this manual.

# 5. I/O CONNECTORS

## 5.1 PC/104 (J1, J2)

Connectors J1 and J2 provide the standard PC/104 16-bit ISA bus. These connectors are installed on all standard models.

The PC/104-*Plus* has pins for +5V, -5V, 12V, and -12V power. RMM-1616A-XT uses only the +5V pins for power and does not require any other power source. The 5V pins on the PC/104 and PC/104-*Plus* connector, if installed, will both be used by the board, however any one connector's pins are sufficient for powering the board.

For information on the signal definitions of these connectors, please view the PC/104 or PC/104-*Plus* specifications available on www.pc104.org.

**View from Top of Board**

**J1: PC/104 8-bit bus connector**

| | | | |
|---|---|---|---|
| IOCHCHK- | A1 | B1 | Ground |
| SD7 | A2 | B2 | RESET |
| SD6 | A3 | B3 | +5V |
| SD5 | A4 | B4 | IRQ9 |
| SD4 | A5 | B5 | -5V |
| SD3 | A6 | B6 | DRQ2 |
| SD2 | A7 | B7 | -12V |
| SD1 | A8 | B8 | 0WS- |
| SD0 | A9 | B9 | +12V |
| IOCHRDY | A10 | B10 | Key |
| AEN | A11 | B11 | SMEMW- |
| SA19 | A12 | B12 | SMEMR- |
| SA18 | A13 | B13 | IOW- |
| SA17 | A14 | B14 | IOR- |
| SA16 | A15 | B15 | DACK3- |
| SA15 | A16 | B16 | DRQ3 |
| SA14 | A17 | B17 | DACK1- |
| SA13 | A18 | B18 | DRQ1 |
| SA12 | A19 | B19 | Refresh- |
| SA11 | A20 | B20 | SYSCLK |
| SA10 | A21 | B21 | IRQ7 |
| SA9 | A22 | B22 | IRQ6 |
| SA8 | A23 | B23 | IRQ5 |
| SA7 | A24 | B24 | IRQ4 |
| SA6 | A25 | B25 | IRQ3 |
| SA5 | A26 | B26 | DACK2- |
| SA4 | A27 | B27 | TC |
| SA3 | A28 | B28 | BALE |
| SA2 | A29 | B29 | +5V |
| SA1 | A30 | B30 | OSC |
| SA0 | A31 | B31 | Ground |
| Ground | A32 | B32 | Ground |

**J2: PC/104 16-bit bus connector**

| | | | |
|---|---|---|---|
| Ground | D0 | C0 | Ground |
| MEMCS16- | D1 | C1 | SBHE- |
| IOCS16- | D2 | C2 | LA23 |
| IRQ10 | D3 | C3 | LA22 |
| IRQ11 | D4 | C4 | LA21 |
| IRQ12 | D5 | C5 | LA20 |
| IRQ15 | D6 | C6 | LA19 |
| IRQ14 | D7 | C7 | LA18 |
| DACK0- | D8 | C8 | LA17 |
| DRQ0 | D9 | C9 | MEMR- |
| DACK5- | D10 | C10 | MEMW- |
| DRQ5 | D11 | C11 | SD8 |
| DACK6- | D12 | C12 | SD9 |
| DRQ6 | D13 | C13 | SD10 |
| DACK7- | D14 | C14 | SD11 |
| DRQ7 | D15 | C15 | SD12 |
| +5V | D16 | C16 | SD13 |
| MASTER- | D17 | C17 | SD14 |
| Ground | D18 | C18 | SD15 |
| Ground | D19 | C19 | Key |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## 5.2 PC/104-*Plus* PCI Bus Connector (J3)

Connector J3 provides the standard PC/104-*Plus* 32-bit PCI bus interface. This connector is installed on all models with **P** in the model number.

**Warning: The PC/104-*Plus* connector on RMM-1616A-XT is not 5V tolerant. The board must be used in a system with 3.3V signaling on the PC/104-*Plus* connector.**

The PC/104-*Plus* connector has pins for both +3.3V and +5V power. RMM-1616A-XT uses only the +5V pins for power and does not require +3.3V from the host computer. If the PC/104-*Plus* connector is not installed, the board will still obtain +5V from the PC/104 ISA connectors J1 and J2.

For information on the signal definitions of this connector, please view the PC/104-*Plus* specification available on www.pc104.org.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A1 | GND/KEY5 | B1 | Reserved | C1 | +5V | D1 | AD00 |
| A2 | VI/O | B2 | AD02 | C2 | AD01 | D2 | +5V |
| A3 | AD05 | B3 | GND | C3 | AD04 | D3 | AD03 |
| A4 | C/BE0* | B4 | AD07 | C4 | GND | D4 | AD06 |
| A5 | GND | B5 | AD09 | C5 | AD08 | D5 | GND |
| A6 | AD11 | B6 | VI/O | C6 | AD10 | D6 | M66EN |
| A7 | AD14 | B7 | AD13 | C7 | GND | D7 | AD12 |
| A8 | +3.3V | B8 | C/BE1* | C8 | AD15 | D8 | +3.3V |
| A9 | SERR* | B9 | GND | C9 | SB0* | D9 | PAR |
| A10 | GND | B10 | PERR* | C10 | +3.3V | D10 | Reserved |
| A11 | STOP* | B11 | +3.3V | C11 | LOCK* | D11 | GND |
| A12 | +3.3V | B12 | TRDY* | C12 | GND | D12 | DEVSEL* |
| A13 | FRAME* | B13 | GND | C13 | IRDY* | D13 | +3.3V |
| A14 | GND | B14 | AD16 | C14 | +3.3V | D14 | C/BE2* |
| A15 | AD18 | B15 | +3.3V | C15 | AD17 | D15 | GND |
| A16 | AD21 | B16 | AD20 | C16 | GND | D16 | AD19 |
| A17 | +3.3V | B17 | AD23 | C17 | AD22 | D17 | +3.3V |
| A18 | IDSEL0 | B18 | GND | C18 | IDSEL1 | D18 | IDSEL2 |
| A19 | AD24 | B19 | C/BE3* | C19 | VI/O | D19 | IDSEL3 |
| A20 | GND | B20 | AD26 | C20 | AD25 | D20 | GND |
| A21 | AD29 | B21 | +5V | C21 | AD28 | D21 | AD27 |
| A22 | +5V | B22 | AD30 | C22 | GND | D22 | AD31 |
| A23 | REQ0* | B23 | GND | C23 | REQ1* | D23 | VI/O |
| A24 | GND | B24 | REQ2* | C24 | +5V | D24 | GNT0* |
| A25 | GNT1* | B25 | VI/O | C25 | GNT2* | D25 | GND |
| A26 | +5V | B26 | CLK0 | C26 | GND | D26 | CLK1 |
| A27 | CLK2 | B27 | +5V | C27 | CLK3 | D27 | GND |
| A28 | GND | B28 | INTD* | C28 | +5V | D28 | RST* |
| A29 | +12V | B29 | INTA* | C29 | INTB* | D29 | INTD* |
| A30 | -12V | B30 | REQ3* | C30 | GNT3* | D30 | GND/KEY3 |

## 5.3 Analog Output (J4)

Connector J4 brings the analog output signals to a pin header.

| | | | |
|---|---|---|---|
| VOUT 0 | 1 | 2 | IOUT 0 |
| AGND | 3 | 4 | VOUT 1 |
| IOUT 1 | 5 | 6 | AGND |
| VOUT 2 | 7 | 8 | IOUT 2 |
| AGND | 9 | 10 | VOUT 3 |
| IOUT 3 | 11 | 12 | AGND |
| VOUT 4 | 13 | 14 | IOUT 4 |
| AGND | 15 | 16 | VOUT 5 |
| IOUT 5 | 17 | 18 | AGND |
| VOUT 6 | 19 | 20 | IOUT 6 |
| AGND | 21 | 22 | VOUT 7 |
| IOUT 7 | 23 | 24 | AGND |
| VOUT 8 | 25 | 26 | IOUT 8 |
| AGND | 27 | 28 | VOUT 9 |
| IOUT 9 | 29 | 30 | AGND |
| VOUT 10 | 31 | 32 | IOUT 10 |
| AGND | 33 | 34 | VOUT 11 |
| IOUT 11 | 35 | 36 | AGND |
| VOUT 12 | 37 | 38 | IOUT 12 |
| AGND | 39 | 40 | VOUT 13 |
| IOUT 13 | 41 | 42 | AGND |
| VOUT 14 | 43 | 44 | IOUT 14 |
| AGND | 45 | 46 | VOUT 15 |
| IOUT 15 | 47 | 48 | AGND |
| EXT TRIG | 49 | 50 | DGND |

Connector type: 0.1” pitch 50-pin (2x25) dual row right-angle pin header with gold flashing.

Mating cable: Diamond Systems C-50-18

Connector signal definitions

Vout 0-15 Analog voltage outputs; quantity depends on model; All outputs start with Vout 0 and go up from there to Vout 3, Vout 7, or Vout 15

Iout 0-15 Analog current outputs; quantity depends on model; Outputs are aligned in the same manner as voltage outputs

Agnd Analog ground; one is provided for each channel, however all analog grounds are tied together so the application can use one or more as desired.

Ext Trig External trigger for D/A waveform generator when using external clocking

Dgnd Digital ground, used as return for Ext Trig signal

## 5.4 Digital I/O (J5)

Connector J5 brings the 48 digital I/O signals to a pin header. These lines have 3.3V logic levels with 5V input tolerance.

| | | | |
|---|---|---|---|
| DIO A0 | 1 | 2 | DIO A1 |
| DIO A2 | 3 | 4 | DIO A3 |
| DIO A4 | 5 | 6 | DIO A5 |
| DIO A6 | 7 | 8 | DIO A7 |
| DIO B0 | 9 | 10 | DIO B1 |
| DIO B2 | 11 | 12 | DIO B3 |
| DIO B4 | 13 | 14 | DIO B5 |
| DIO B6 | 15 | 16 | DIO B7 |
| DIO C0 | 17 | 18 | DIO C1 |
| DIO C2 | 19 | 20 | DIO C4 |
| DIO C4 | 21 | 22 | DIO C6 |
| DIO C6 | 23 | 24 | DIO C7 |
| DIO D0 | 25 | 26 | DIO D1 |
| DIO D2 | 27 | 28 | DIO D3 |
| DIO D4 | 29 | 30 | DIO D5 |
| DIO D6 | 31 | 32 | DIO D7 |
| DIO E0 | 33 | 34 | DIO E1 |
| DIO E2 | 35 | 36 | DIO E3 |
| DIO E4 | 37 | 38 | DIO E5 |
| DIO E6 | 39 | 40 | DIO E7 |
| DIO F0 | 41 | 42 | DIO F1 |
| DIO F2 | 43 | 44 | DIO F4 |
| DIO F4 | 45 | 46 | DIO F6 |
| DIO F6 | 47 | 48 | DIO F7 |
| +5VDC | 49 | 50 | DGND |

Connector type: 0.1” pitch 50-pin (2x25) dual row right-angle pin header with gold flashing.

Mating cable: Diamond Systems C-50-18

Connector signal definitions

DIO A0-A7 Digital I/O port A or 0; directions of all 8 bits are set in unison

DIO B0-B7 Digital I/O port B or 1; directions of all 8 bits are set in unison

DIO C0-C7 Digital I/O port C or 2; directions of all 8 bits are set in unison

DIO D0-D7 Digital I/O port D or 3; directions of all 8 bits are set in unison

DIO E0-E7 Digital I/O port E or 4; directions of all 8 bits are set in unison

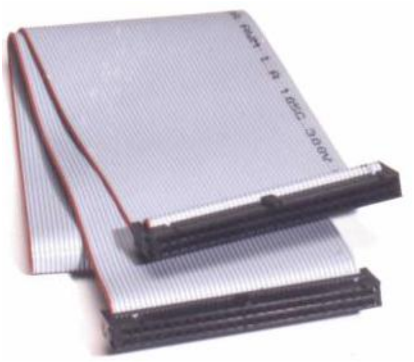
DIO F0-F7 Digital I/O port F or 5; directions of each bit is set independently; these bits also serve as counter and PWM I/O signals

+5VDC +5VDC available for powering external logic; limited to ~100mA

Dgnd Digital ground, used as return for Ext Trig signal

## 5.5 Ribbon Cable

The figure below shows Diamond Systems Cable C-50-18, which mates with both I/O connectors J4 and J5 on the RMM-16161 board. This is a 50-conductor ribbon cable, 18 inches length, with 2x25 .1” pitch IDC female connectors at both ends.



# 6. ARCHITECTURE OVERVIEW

## 6.1 Bus Interface

The RMM-1616A-XT family uses a new FPGA architecture that includes both PCI and ISA bus interfaces. This allows the full functionality of the board to be offered in both PC/104 (ISA connector only) and PC/104-*Plus* (ISA and PCI connector) formats. (A PCI-104 version with only the PCI connector is available by special order.) The PC/104-*Plus* models may use either the PCI or ISA connector with a jumper option. In most cases the user will want to use the default PCI interface.

The board's register map contains a total of 256 addressable 8-bit registers organized as 16 pages of 16 registers each. Registers are grouped into pages according to function: A/D, D/A, digital I/O, etc. This page method allows the board to occupy a small footprint of only 16 bytes in ISA mode. For consistency purposes the identical page system is utilized in PCI mode as well.

In both PCI and ISA modes all I/O is 8 bits wide. Operations that require 16-bit or wider data, such as D/A registers, counter/timer registers, and PWM registers, are managed with multiple 8-bit transactions.

## 6.2 D/A Circuit

The RMM-1616A-XT family utilizes the Analog Devices AD5755 D/A converter for all analog output functions. The AD5755 provides 4 16-bit DACs with high accuracy, low drift, programmable voltage and current output ranges, and digital calibration. Up to 4 of these devices may be installed on the board depending on the model. A precision, low-drift 5V voltage reference circuit provides the basis for the overall accuracy of the analog outputs.

The AD5755 contains an integrated digital calibration circuit consisting of a multiplier and adder. Each time data is written to a DAC, it undergoes a multiplication / addition operation, and the result is then transferred to the DAC channel. This operation takes about 5 microseconds to complete. Thus each write to a DAC channel results in a 5 us delay before the output begins to update to the new value. The total settling time for one channel consists of the settling time for the DAC plus this calibration time.

## 6.3 Digital I/O

The FPGA provides 48 digital I/O lines that are brought out to a separate I/O connector. All I/O lines are buffered for protection and have user-configurable pull-up/down resistors as well as user-configurable 3.3V/5V logic levels. These lines are grouped as 5 8-bit ports and 8 1-bit ports. The 8-bit ports utilize 8-bit bi-directional transceivers whose directions are individually controlled via a register. The 1-bit ports utilize 1-bit bi-directional transceivers whose directions are similarly individually controlled via a separate register. These 1-bit ports also serve as handshake signals for the other 5 ports, as well as I/O signals for the counter/timers and PWMs.

Special functions are enabled on Port F to support digital I/O, counter/timer, and PWM operation. These functions are controlled via register bits or commands in these other circuit blocks. The special configuration, when enabled, overrides the current direction and value of the assigned port F bit.

The special functions for digital I/O and counter/timer operation are shown below:

| Port F bit | F7 | F6 | F5 | F4 | F3 | F2 | F1 | F0 |
|---|---|---|---|---|---|---|---|---|
| Function | Ack | Latch | Ctr1Clk | Ctr0Clk | Ctr1Out | Ctr0Out | Ctr1Gate | Ctr0Gate |
| Direction | Out | In | In | In | Out | Out | In | In |

I/O signals F3-0 may also be reassigned as PWM outputs as shown below:

| Port F bit | F7 | F6 | F5 | F4 | F3 | F2 | F1 | F0 |
|---|---|---|---|---|---|---|---|---|
| Function | | | | | PWM3 | PWM2 | PWM1 | PWM0 |
| Direction | | | | | Out | Out | Out | Out |

When a board reset occurs, all DIO lines are released from any counter or PWM assignment and return to normal operation. They are reset to input mode, and their output registers are reset to 0.

## 6.4 Counter/Timers and Clock Sources

The FPGA includes 2 32-bit programmable counter/timers that serve a variety of purposes. The most common uses are a divide-by-n counter to create a programmable rate generator and counting external pulses. Each counter has programmable input, gate (count enable), and output functions. The input may be selected from the on-board 50MHz clock, a 1MHz clock derived from the 50MHz clock, or an external signal via one of the 1-bit digital I/O ports. The gate is optional and also derives from one of the 1-bit ports. The output is also optional. If enabled it is driven onto one of the 1-bit ports as well. Both positive and negative output polarities are supported.

## 6.5 Pulse Width Modulators

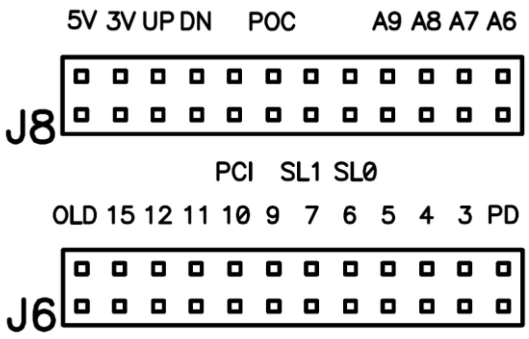
The FPGA includes 4 24-bit pulse-width modulator circuits. Each circuit includes a period register as well as a duty cycle register. Both registers may be updated in real-time without stopping the PWM. Duty cycles from 0-100% inclusive are supported, as well as both positive and negative output polarity. The PWM clock may be selected from the on-board 50MHz clock or a 1MHz clock derived from the 50MHz clock. The PWM outputs may be enabled onto the 1-bit port pins. These are general purpose I/O pins with limited voltage and current capability. The user must determine whether these pins provide the appropriate voltage and current levels for the intended application or whether additional buffering or amplification is required.

## 6.6 Interrupts

Interrupts enable the board to request service independently of program operation, typically in response to a user-defined time interval or an external event. The board supports interrupts from a variety of sources, including both counter/timers and the digital I/O circuit. The application program is responsible for providing the interrupt service routine to respond to the interrupt request. An unserviced interrupt request may cause the computer to crash. Diamond's Universal Driver software includes built-in interrupt handling routines that can link to user-defined code. This feature lets you define the conditions that will generate an interrupt and then define the behavior of the system when an interrupt occurs.

# 7. JUMPER CONFIGURATION

This section describes how to configure the RMM-1616 board using jumper blocks J6 and J8. An illustration of the two jumper blocks is shown below. All jumpers which are in use are installed in a vertical orientation across two pins in the jumper blocks. The board may be supplied with several jumpers installed over only one pin, which is a storage position. The jumper is not needed in the default configuration but is supplied for possible future use. Jumpers that are not needed may be removed.



## 7.1 Base Address (ISA models only)

RMM-1616 occupies a range of 16 bytes in the I/O memory of the PC. The starting address of this 16-byte window is called the base address. On ISA bus models, the base address is configurable with jumper block J8. On PCI models, the base address is configured by the BIOS during system startup.

The jumper locations labeled A9 – A6 are used to select the ISA base address. These lines correspond to ISA bus address lines SA9 – SA4. By definition, the lowest 4 address bits, SA3 – SA0, are all 0 for the base address, since these are used to select a byte within the 16-byte window. The upper address lines, SA15 – SA10, are always decoded as 0 to ensure a unique base address decode.

The table below lists the valid base address settings. When a jumper is installed, the corresponding address line is 0. When a jumper is removed, the corresponding address line is 1. Addresses that are below 100 hex are considered invalid and should not be configured. The default base address is 300 hex / 768 decimal.

| A9 | A8 | A7 | A6 | Base Address (Hex) | Base Address (Decimal) | |
|---|---|---|---|---|---|---|
| In | Out | In | In | 100 | 256 | |
| In | Out | In | Out | 140 | 320 | |
| In | Out | Out | In | 180 | 384 | |
| In | Out | Out | Out | 1C0 | 448 | |
| Out | In | In | In | 200 | 512 | |
| Out | In | In | Out | 240 | 576 | |
| Out | In | Out | In | 280 | 640 | |
| Out | In | Out | Out | 2C0 | 704 | |
| Out | Out | In | In | 300 | 768 | default |
| Out | Out | In | Out | 340 | 832 | |
| Out | Out | Out | In | 380 | 896 | |
| Out | Out | Out | Out | 3C0 | 960 | |

## 7.2 IRQ Selection (ISA models only)

The ISA bus interrupt level is selected with jumper block J6 from levels 3, 4, 5, 6, 7, 9, 10, 11, 12, and 15. Install a jumper in any **ONE** location to select that IRQ number. Only one IRQ may be selected at a time.

The PC/104 bus requires a 1K ohm pull-down resistor on each IRQ line that is in use by a peripheral. If multiple boards in the system are sharing the same IRQ number, only one board should have the pull-down resistor installed. The pull-down resistor is enabled with jumper position PD on jumper block J6. The default is IRQ 7 and pull-down enabled.

In PCI bus systems, the IRQ level is unique to each slot and is selected with the PCI slot selection jumpers.

## 7.3 PCI Slot Number (PCI models only)

PCI devices in a PC/104 system are configurable to occupy a “slot” on the bus. This emulates the physical slots on a PCI backplane, where each PCI slot has its own unique signals for ID select, clock, and IRQ. Each PCI device must be assigned to a unique slot. The PCI slot number for RMM1616 is selected with jumper positions SL1 and SL0 on jumper block J8. The default is both jumpers installed to set PCI slot ID=00. **Note**: On many PC/104-Plus SBCs, slot 0 may already be occupied by a device on the SBC, such as an Ethernet controller or some other on-board peripheral.

| SL1 | SL0 | Slot |
|---|---|---|
| In | In | 0 |
| In | Out | 1 |
| Out | In | 2 |
| Out | Out | 3 |

## 7.4 Bus Select

On models with both the ISA and PCI connectors installed, the board can be set in auto-bus select mode. In this mode, if the board is installed in a system with only the PC/104 connector, it will use the ISA interface, and if it is installed in a system with the PCI connector (with or without the ISA connector), it will use the PCI interface. To use auto-bus select mode, or to use PCI mode, install a jumper in position PCI on jumper block J8. This is the default configuration. The jumper must be installed to enable PCI bus mode. If this jumper is removed, the board will always communicate over the ISA bus, even if the PCI bus is present. The table below lists all the possible configurations:

| Jumper | PCI present | ISA present | Bus Used |
|---|---|---|---|
| Out | No | Yes | ISA |
| Out | Yes | Yes | ISA |
| In | No | Yes | ISA |
| In | Yes | Yes | PCI |

## 7.5 Digital I/O Logic Level

The logic levels for the digital I/O can be set to either +5V or +3.3V logic levels. All DIO lines are configured for the same voltage levels; it is not possible to configure some for 5V and some for 3.3V operation. The voltage level is selected on jumper block J8. Install **ONE** jumper in either the 5V or 3V jumper location. The default is +5V. WARNING: Install only one jumper in either 5V or 3.3V positions. Installing jumpers in both positions will short the two power supplies together and cause damage to the board and/or the entire system.

## 7.6 Digital I/O Pull-Up/Down

The digital I/O lines are tied to individual pull-up/down resistors which may be jumpered to either the DIO logic level voltage selected above or ground using locations UP or DN on jumper block J8, respectively. All DIO lines are pulled up or down together; it is not possible to configure some for pull-up and some for pull-down. The default is pull-up.

## 7.7 D/A Converter Power On Control

The D/A converters can be configured to power up in two modes, selected with jumper position POC on jumper block J8:

1. Jumper installed: Current outputs are tristated and voltage outputs are tristated.
2. Jumper not installed: Current outputs are tristated and voltage outputs are tied to analog ground with a 30K ohm resistance.

The default setting is "jumper installed".

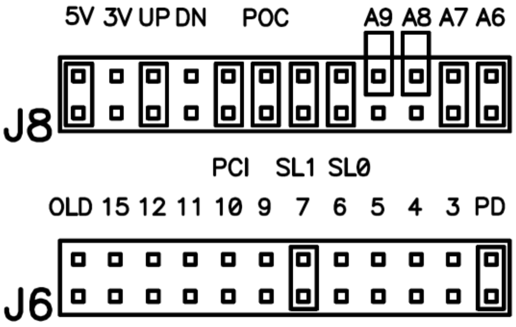
## 7.8 Old Jumper

Jumper block J6 has one position labeled "Old". This jumper is reserved for future use and is currently not enabled.

## 7.9 Default Configuration

The default configuration (configuration in which the board is shipped from stock) is shown and illustrated below. Note that both ISA and PCI configurations are provided for all boards, however only the relevant configuration will matter depending on which bus connector is installed / in use.

- ISA base address: 300 hex / 768 decimal
- ISA IRQ level: IRQ 7, pull-down resistor enabled
- PCI slot number: Slot 0
- Bus select: Auto-bus select / PCI bus if available
- Digital I/O logic levels: 5V
- Digital I/O pull-up/down: Pull-up
- D/A power-on control: Current and voltage outputs tristated



# 8. ANALOG OUTPUT TECHNOLOGY

## 8.1 Output Channel Organization

The RMM1616A/AP-XT modules use the Analog Devices AD5755 quad 16-bit D/A converter chips to provide the analog outputs. One to four chips are installed on the board, depending on the model. The channels are organized as follows:

| Board Channel | Chip Number | Channel Number on Chip |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 0 | 2 |
| 3 | 0 | 3 |
| 4 | 1 | 0 |
| 5 | 1 | 1 |
| 6 | 1 | 2 |
| 7 | 1 | 3 |
| 8 | 2 | 0 |
| 9 | 2 | 1 |
| 10 | 2 | 2 |
| 11 | 2 | 3 |
| 12 | 3 | 0 |
| 12 | 3 | 1 |
| 12 | 3 | 2 |
| 12 | 3 | 3 |

## 8.2 Output Ranges and Resolution

These chips provide both voltage and current outputs in multiple output ranges. Each channel on each chip can be set to a different output range, including voltage and current ranges on the same chip. Each channel has a voltage output pin, a current output pin and a ground return pin. The application wiring must connect to the voltage output pin or the current output pin, as needed.

A D/A converter converts a number, or output code, into an output voltage or current that is proportional to the number. The output range is the range of possible output values, from the smallest (lowest) value up to the highest (largest) value. The difference between the highest and lowest output value is called the span. For a +/-5V output range, the span is 10V, and for a 4-20mA output range, the span is 16mA.

The smallest change in output value, or resolution, is equal to 1/2n x the span, in which n = the number of bits (in this case 16). For a +/-5V output range, the resolution is 10V / 65535 = 153uV. This smallest change is commonly referred to as 1 LSB or the Least Significant Bit.

RMM-1616AP-XT uses straight binary coding for all output values; the range of output codes is 0-65535. The theoretical top value, 65536, requires 17 bits to be represented in binary form, which is unachievable in a 16-bit value. Therefore the top value of each output range is unavailable, and instead the maximum output value is 1 LSB less than the top value. Because the lowest output code is always 0, which is represented in binary form, the bottom value of each range is always equal to the exact nominal value of the range (within tolerance of the accuracy).

The table below summarizes all this information for all output ranges on RMM-1616AP-XT.

| **Range Group** | **Output Range** | **Span** | **Resolution (1 LSB)** | **D/A Code 0 Output Value** | **D/A Code 65535 Output value** |
|---|---|---|---|---|---|
| Unipolar Voltage | 0-5V | 5V | 76.3uV | 0.0000V | 4.9999V |
| Unipolar Voltage | 0-10V | 10V | 153uV | 0.0000V | 9.9998V |
| Bipolar Voltage | +/-5V | 10V | 153uV | -5.0000V | 4.9998V |
| Bipolar Voltage | +/10V | 20V | 305uV | -10.0000V | 9.9997V |
| Current | 0-20mA | 20mA | 0.305uA | 0.0000mA | 19.9997mA |
| Current | 0-24mA | 24mA | 0.366uA | 0.0000mA | 23.9996mA |
| Current | 4-20mA | 16mA | 0.244uA | 4.0000mA | 19.9998mA |

The conversion formula for analog outputs is the same for all ranges and is shown below (minimum output value = output value for a code of 0 shown in the table above):

**Output V/I = (D/A code / 65536) x Span + Minimum output value**

On power-up or when the board is reset, all voltage and current output pins are set to a tristate mode. All outputs are reset to the 0-5V output range with the output code of 0 programmed into the output registers, so that when outputs are enabled the voltage pins will have 0V on them (and the current pins will remain tristated). The D/A initialization sequence described later in this manual also sets all outputs to 0-5V range with 0V output. This sequence can be modified to use a different output range as desired.

## 8.3 Output Modes

Each output channel on RMM-1616AP-XT can be updated individually, or any number of channels can be updated simultaneously, including different channels on different chips. Simultaneous update is useful when the application requires precise timing to avoid distortion or errors in the behavior of the controlled device, for example when controlling the movement of an X-Y table or a laser beam. The **Error! Reference source not found.** chapter in this manual describes both single-channel and multi-channel output modes.

## 8.4 Calibration

**Note:** The RMM-1616 is factory calibrated. All calibration settings are stored in an on-board EEPROM for instant automatic recall each time the board powers up. All analog outputs power up to 0V for safety. If recalibration or calibration for nonstandard D/A ranges are needed, please contact Diamond Systems for technical support.
All analog components contain inherent errors in offset and gain which affect the accuracy of the signals they generate. These errors are very small on RMM-1616AP-XT; however they are still present and could present a problem for some high-precision applications. Calibration is used to correct these errors so that the actual output of the D/A channels is as close as possible to the theoretical output.

The AD5755 D/A converter uses a digital calibration method to correct for offset and gain errors. Each output channel has a 16-bit Offset register, called the C register, and a 16-bit Gain register, called the M register. This enables each channel to be calibrated independently for maximum overall accuracy. Each time an output code is written to a channel, the chip will automatically apply the offset and gain correction to the code, resulting in a corrected digital value. This corrected value is then converted to the output voltage or current according to the output range. The calibration process takes about 5us and is unavoidable. This 5us delay is included in the specified settling time for the analog outputs.

For improved accuracy, the bipolar voltage, unipolar voltage, and current output range groups each have their own calibration settings. Within any group, for example between the 0-5V and 0-10V ranges, the differences in errors are very small, so the same calibration values are used for the entire group. However between range groups the errors are noticeable, so separate calibration values are used for each group.

The calibration values for the unipolar and bipolar voltage range groups are stored in an EEPROM on the board. On power-up or reset, the unipolar voltage range calibration values are read from the EEPROM and loaded into the AD5755 chips. If needed, the calibration values for a different range can be read from the EEPROM and stored. This is explained in the How-to section later in this manual.

**Note**: There are no calculated calibration values for current output ranges. Instead, default values are used. When using a current output range, the application software can load the default calibration values into the DACs. See the Calibration EEPROM information below for the default values.

The conversion formula from the written output code and the calibrated code is as follows:

Corrected code = Written code x (M register / 65535 (0xFFFF)) + (C register – 32768 (0x8000))

The minimum value is always 0, and the maximum value is always 65535 / 0xFFFF. Any result which exceeds these limits will be automatically set to the limit.

The corrected code is then converted to the output voltage or current according to the formula above.

# 9. CALIBRATION EEPROM

The board contains an EEPROM that stores the factory calibration settings for the board. For safety, a protected area of the EEPROM stores a backup copy of the factory calibration settings. In case the calibration settings become accidentally corrupted, contact Diamond Systems Technical Support for information on how to restore the calibration settings from the backup copy.

Each channel has its own set of calibration values. Each calibration value is 16 bits, occupying two bytes in the EEPROM. Each output range group has two calibration values, one for offset (called the C value) and one for gain (called the M value). There are two output range groups with calibration values, unipolar voltage (0-5V and 0-10V) and bipolar voltage (+/-5V and +/-10V). Thus there are a total of 8 calibration bytes for each channel, or 128 bytes total for the board:

| | |
|---|---|
| Unipolar offset (C) | 2 bytes |
| Unipolar gain (M) | 2 bytes |
| Bipolar offset (C) | 2 bytes |
| Bipolar gain (M) | 2 bytes |

Current output ranges do not have calibration values. The calibration values for all current ranges on all channels are fixed as follows:

| | |
|---|---|
| Offset / C value | 0x8000 |
| Gain / M value | 0xFE00 |

The calibration data is stored in the EEPROM as shown below:

| **Address** | | **Definition** |
|---|---|---|
| **Hex** | **Decimal** | |
| 0 | 0 | Channel 0 M (gain) value LSB for unipolar ranges |
| 1 | 1 | Channel 0 M (gain) value MSB for unipolar ranges |
| 2 | 2 | Channel 0 C (offset) value LSB for unipolar ranges |
| 3 | 3 | Channel 0 C (offset) value MSB for unipolar ranges |
| 4 | 4 | Channel 1 M (gain) value LSB for unipolar ranges |
| 5 | 5 | Channel 1 M (gain) value MSB for unipolar ranges |
| 6 | 6 | Channel 1 C (offset) value LSB for unipolar ranges |
| 7 | 7 | Channel 1 C (offset) value MSB for unipolar ranges |
| … | … | … |
| 3C | 60 | Channel 15 M (gain) value LSB for unipolar ranges |
| 3D | 61 | Channel 15 M (gain) value MSB for unipolar ranges |
| 3E | 62 | Channel 15 C (offset) value LSB for unipolar ranges |
| 3F | 63 | Channel 15 C (offset) value MSB for unipolar ranges |

In general you should never have to write new data to the EEPROM, since the analog components on RMM-1616A-XT feature high accuracy and very low drift. However reading from the EEPROM is necessary to recall calibration settings when you change analog output ranges in order to ensure best accuracy. It is advisable to recall the calibration values from the EEPROM whenever the range is changed from one range group to another. When changing ranges within a range group, no calibration recall is necessary.

| | |
|---|---|
| Unipolar voltage range group: | 0-5V, 0-10V |
| Bipolar voltage range group: | +/-5V, +/-10V |
| Current range group: | 0-20mA, 0-24mA, 4-20mA |

# 10. PROGRAMMING INSTRUCTIONS

## 10.1 Overview

The analog outputs on RMM-1616 are provided by 1–4 Analog Devices AD5755 quad 16-bit D/A converters. This chip has a high degree of functionality and a correspondingly complex programming procedure. When possible, it is recommended to use the Diamond Systems Universal Driver software to control the board instead of programming it directly. For applications using operating systems that are not supported by Universal Driver, the information provided in this chapter may be used as a guide for programming the most common functions.

The digital I/O, counter/timer, and PWM functions are simpler. Typically, these functions can be easily programmed using direct register read/write operations. Instructions to do so are provided in this chapter.

## 10.2 RMM-1616 Register Map

RMM-1616 is controlled by an FPGA with a number of registers organized into pages. Each page has 16 8-bit registers. The 16 pages overlap each other in a physical 16-byte window in the PC's I/O memory space. The same register structure and memory footprint are used in both ISA and PCI interface modes. The lowest address of this 16-byte window is called the base address. In ISA mode, the base address is selected with jumpers on the board. In PCI mode, the base address is programmed into the board by the PC's BIOS during startup and is accessible in the board's PCI registers.

To select the page, the desired page number is written to the lowest 4 bits of register 15. These bits have the same definition in all pages, so the page number can be both written and read at any time.

| Offset | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 15 | | | | | PAGE3 | PAGE2 | PAGE1 | PAGE0 |

## 10.3 Reset Board

Resetting the board causes the FPGA to rest all its registers and functionality, so that everything is in a default and inactive state. The board is automatically reset whenever a power-on reset or system reset occurs. A reset causes the following to occur:

- All digital I/O lines revert to input mode, so that they are not actively driving anything external to the board. This will also cause all digital I/O lines to attain the logic level set by the DIO pull-up/down resistor jumper.
- All counter/timers, PWMs, and the D/A waveform generator are stopped.
- All D/A outputs revert to 0V output (current outputs are tristated).

The reset procedure is as follows:

1. Write a 1 to the BRDRST bit: Page 7, register 15, bit 7.
2. Monitor EEBUSY bit and wait until it is 0: Page 5, register 7, bit 7.

## 10.4 Initialize Board

Before any analog output functionality can be used, the board must be initialized. Board initialization consists of two tasks:

1. Reset the board to its initial state as described above.
2. Initialize the D/A converter chips.

### 10.4.1 D/A initialization

Initializing the D/A chips is a complex procedure that requires programming several configuration registers for each DAC on each chip. The RMM-1616 may have 1 to 4 DAC chips, and each chip has 4 DACs. Each chip, as well as each DAC on each chip, needs to be configured during the initialization process.

All data transmission to the DACs is in a 24-bit/3 byte format that includes an 8-bit command / register ID plus 16 bits of data. The 3 bytes are written to the DAC register group, page 1, registers 0-2. These registers are shown below. Register 2 shows the specific definitions of the command bits.

**Page 0, registers 0-2:**

| Offset | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 (W) | DA7-0 | | | | | | | |
| 1 (W) | DA15-8 | | | | | | | |
| 2 | RD/~WR | DA_D1 | DA_D0 | DA_R2 | DA_R1 | DA_R0 | DA_A1 | DA_A0 |

When a write occurs to register 2, the FPGA on RMM1616 transmits the 24 bits to the specified DAC. During certain transmissions, the status bit DABUSY for the target DAC will be 1 indicating the DAC is busy processing the command. The software must therefore wait for a chip's DABUSY bit to be 0 before attempting a new data transmission to that chip. Writing to a different chip does not require waiting for the first chip's DABUSY signal to return to 0.

**Page 0, register 7:**

| Offset | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 7 (R) | DABUSYD | DABUSYC | DABUSYB | DABUSYA | FAULTD | FAULTC | FAULTB | FAULTA |

The instruction sequences below assume a 16-channel board with 4 DAC chips, each with 4 channels. For programming purposes the chips and channels are both numbered 0-3. Chip 0 has channels 0-3, chip 1 has channels 4-7, and so on. The commands for all 4 chips are shown in sequence starting with chip 0. If a board has fewer than 4 chips, the additional commands may be omitted, or they may be executed with benign results.

1. Select the D/A register page 1 by writing 0x01 to register 15 (0x0F) in any page.
2. Perform chip reset on all DAC chips. One command is required per chip. The commands are:

   1C8555, 3C8555, 5C8555, 7C8555
3. Program main control registers on all DAC chips. One command is required per chip. The recommended commands are:

   1C2000, 3C2000, 5C2000, 7C2000
4. Program clear code registers on all channels on all DAC chips. The clear code is the code that will be loaded into the DAC when the Clear command is executed (this is different than a reset command). Four commands are required for each chip (one per channel):

   Chip 1: 180000, 190000, 1A0000, 1B0000

   Chip 2: 380000, 390000, 3A0000, 3B0000

   Chip 3: 580000, 590000, 5A0000, 5B0000

   Chip 4: 780000, 790000, 7A0000, 7B0000

5. Program DC/DC control register on each chip. One command is required per chip. The recommended commands are:

   1C6077, 3C6077, 5C6077, 7C6077

6. Write to all DAC control registers on all DAC chips to set 0-5V range with outputs disabled. Four commands are required for each chip (one per channel):

   Chip 1: 1C4110, 1D4110, 1E4110, 1F4110

   Chip 2: 3C4110, 3D4110, 3E4110, 3F4110

   Chip 3: 5C4110, 5D4110, 5E4110, 5F4110

   Chip 4: 7C4110, 7D4110, 7E4110, 7F4110

7. Write 0x0000 to all DAC registers on all DAC chips (16 commands). Four commands are required for each chip (one per channel):

   Chip 1: 000000, 010000, 020000, 030000

   Chip 2: 200000, 210000, 220000, 230000

   Chip 3: 400000, 410000, 420000, 430000

   Chip 4: 600000, 610000, 620000, 630000

8. Wait 200us

9. Write to all DAC control registers on all DAC chips to set 0-5V range with outputs enabled. Four commands are required for each chip (one per channel):

   Chip 1: 1C4150, 1D4150, 1E4150, 1F4150

   Chip 2: 3C4150, 3D4150, 3E4150, 3F4150

   Chip 3: 5C4150, 5D4150, 5E4150, 5F4150

   Chip 4: 7C4150, 7D4150, 7E4150, 7F4150

10. Wait for all 4 DABUSY status bits = 0: Page 1, register 7.

## 10.5 Configure D/A Output Range

The output range for each channel on the AD5755 is set independently. It is possible to configure each chip for any combination of output ranges, including both voltage and current outputs. The method for changing the output range on the AD5755 DAC is as follows:

1. Select the D/A register page 1 by writing 0x01 to register 15 (0x0F) in any page.
2. Set output to 0V / 0mA code.

   Before setting the output, ensure the current output code is stored in the DAC; this code will be needed later in this process.

   a. If the currently configured range is 2 (+/-5V) or 3 (+/-10V), the required output code is 0x8000, otherwise it is 0x0000. Write the 16-bit code to registers 0-1.

   b. Command = 0x20 x chip number + channel number. Write the  command to register 2.

   c. Wait for DABUSYn = 0 in register 7.
3. Write to the DAC control register with the new output range. The output is temporarily disabled.

   a. Data = 0x4110 + range number. Range numbers are listed below. Write Data to registers 0-1.

   b. Command = 0x1C + 0x20 x chip number + channel number. Write the  command to register 2.

| Number | Range |
|---|---|
| 0 | 0-5V |
| 1 | 0-10V |
| 2 | +/-5V |
| 3 | +/-10V |
| 4 | 4-20mA |
| 5 | 0-20mA |
| 6 | 0-24mA |
| 7 | Invalid setting |

4. Recall calibration settings if desired or if a current output range is selected (4, 5, or 6).

   When changing between unipolar voltage ranges, unipolar ranges, and current ranges, it is recommended to recall the stored calibration values for best accuracy. When changing within unipolar, bipolar, or current ranges, no calibration recall is necessary or available. For example: When changing from 0-5V to +/-5V, recall the bipolar calibration settings for the channel. When changing from 0-5V to 0-10V, no calibration recall is required.

   The calibration values are stored in the EEPROM. See the EEPROM section for information on reading data from the EEPROM and writing the calibration values to the DAC.
5. Write the existing output value to the DAC data register.

   Write the 16-bit D/A code to registers 0–1.

   Command = 0x20 x chip number + channel number. Write the  command to register 2.

   Wait for DABUSYn = 0 in register 7.
6. Write to DAC control register again to enable the output.

   Data = 0x4150 + range number. Range numbers are listed in step 2 above.

   Command = 0x1C + 0x20 x chip number + channel number. Write the command to register 2.

## 10.6 Generate D/A Conversion, Single Channel

This sequence assumes the output range has been previously set to the desired range and that the SIM bit = 0, meaning simultaneous update mode is not enabled.

1. Select the D/A register page 1 by writing 0x01 to register 15 (0x0F) in any page.
2. Write the desired 16-bit output code to registers 0-1.
3. Command = 0x20 x chip number + channel number. Write the  command to register 2.
4. Wait for DABUSYn = 0 in register 7.
5. Write 1 to command bit DALDn in register 6 to update DAC (n = chip number).

## 10.7 Generate D/A Conversion, Multi-Channel with/without Simultaneous Update

1. Select the D/A register page 1 by writing 0x01 to register 15 (0x0F) in any page.
2. To use simultaneous update mode, the SIM bit must be set before any updates are made. This is done by writing 1 to register 5 bit 0.
3. Any number of channels can now be written to with new data as described in the single channel procedure above, steps 1–3. However, do not write 1 to the DALDn bit after writing to the channel.
    - Wait for DABUSYn = 0 for any chip before writing to that chip.
4. After all channels are loaded with new data, wait for all DABUSYn = 0 to ensure all channels are ready.
5. Update all channels simultaneously: write 1 to register 6 bit 7.

## 10.8 D/A Waveform Generator

Programming the D/A waveform generator is a complex operation which is beyond the scope of this manual. When possible, the Diamond Systems Universal Driver software should be used for this feature. If assistance is needed to use the waveform generator without using Universal Driver, please contact Diamond Systems technical support.

## 10.9 Program Counter/Timer for Down Counting

The most common uses of the counter/timers are as a rate generator for controlling time functions, such as D/A waveform generator or timed interrupts, and as a counter of external events. To program a counter/timer for down counting, the procedure below may be used.

1. Calculate the divisor needed to generate the desired pulse rate.
   - This requires selecting the clock source. There are two possible internal clock sources, 50MHz and 1MHz. For the best time resolution, the 50MHz clock should be used when possible. A good rule of thumb is that if the desired rate is 5Hz or higher use the 50MHz clock, otherwise use the 1MHz clock.
   - The divisor formula is: **Divisor = clock source / desired rate**
   - Round the calculated divisor value up or down as needed for best accuracy.
2. Select the counter/timer register page 3 by writing 0x03 to register 15 (0x0F) of any page.
3. Write the divisor as a 4 byte unsigned value to registers 0-3, with 0 being the LSB and 3 being the MSB.
4. Select either counter 0 or counter 1. Write the counter number to register 4.
5. For all remaining operations below, register 4 must remain loaded with the selected counter number.
6. Write command 0x10 to register 5 to load the data into the selected counter.
7. Write command 0x20 to register 5 to select down counting.
8. For 50MHz clock, write 0x62 to register 5. For 1MHz clock, write 0x63 to register 5.
9. Write command 0x30 to register 5 to disable external gating of the counter. This allows the counter to run freely.
10. Option: To see the counter output on a DIO pin, do one of the following:
    - For positive output pulses, write 0x82 to register 5.
    - For negative output pulses, write 0x83 to register 5.
    - Counter 0 output is available on DIO pin F2. Counter 1 output is available on DIO pin F3.
    - When a counter/timer is configured for output, the normal DIO behavior of that pin is superseded with the counter output signal, and the pin is set to output mode. When the board is reset, the DIO pin returns to normal operation and reverts back to input mode.
11. To avoid having the counter output appear on a DIO pin, write 0x80 to register 5.
12. Write command 0x71 to register 5 to enable auto-reload. This allows the counter to run continuously.
13. Write command 0x41 to register 5 to start the counter.

## 10.10 Program Counter/Timer for Up Counting

This procedure enables you to program a counter for up counting using an external signal on a DIO pin.

1. Select the counter/timer register page 3: write 0x03 to register 15 (0x0F) of any page.
2. Write the value 0x00 to registers 0-3 to clear the counter register.
3. Select either counter 0 or counter 1. Write the counter number to register 4.

**Note**: For the remaining operations below, register 4 must remain loaded with the selected counter number.

4. Write command 0x10 to register 5 to load the zero data into the selected counter.
5. Write command 0x21 to register 5 to select up counting.
6. Write command 0x60 to register 5 to select external DIO pin for the clock source.
   - For counter 0, the source signal is DIO pin F4.
   - For counter 1, the source signal is DIO pin F5.
   - A falling edge on the DIO pin will increment the counter.
7. Write command 0x30 to register 5 to disable external gating of the counter. This allows the counter to run freely.
8. For up counting, the counter output signal is irrelevant; write command 0x80 to register 5 to disable the counter output signal.
9. Write command 0x71 to register 5 to enable auto-reload. This allows the counter to run continuously: If the counter reaches its terminal count of 2^32-1, it will roll over to 0 and start counting up again.
10. Write command 0x41 to register 5 to start the counter.

## 10.11 Generate PWM Output

The board contains four PWMs that are numbered 0, 1, 2 and 3. Each PWM includes two counters. The rate counter controls the pulse rate, while the duty cycle counter controls the duty cycle.

The duty cycle can vary between 0% and 100%. The output can be programmed for either high pulses (positive polarity) or low pulses (negative polarity), with the pulse being defined as the active portion of the duty cycle. For example, a pulse with 10% duty cycle and positive output polarity is high 10% of the period and low 90% of the period.

PWM outputs are available on Digital I/O pins F0-F3 for PWMs 0-3, respectively. When a PWM is configured for output, the normal DIO behavior of that pin is superseded with the PWM signal, and the pin is set to output mode. When the board is reset, the DIO pin returns to normal operation and reverts back to input mode.

1. Select the counter/timer register page 3 by writing 0x03 to register 15 (0x0F) of any page. (This page also controls the PWMs.)
2. Write command byte 0x08 + PWM number 0-3 to register 11 to stop the PWM.
3. Calculate the divisor needed to generate the desired pulse rate.
    a. This requires selecting the clock source. There are two possible internal clock sources, 50MHz and 1MHz. For best time resolution, the 50MHz clock should be used when possible. A good rule of thumb is if the desired rate is 5Hz or higher use the 50MHz clock; otherwise use the 1MHz clock.
    b. The divisor formula: **Divisor = clock source / desired rate.**
    c. Round the calculated divisor value up or down as needed for best accuracy.
    d. The divisor is programmed as a 24-bit number. Break the divisor into three 8-bit bytes.
4. Write the divisor to registers 8, 9, and 10. Register 8 is the LSB, register 10 is the MSB.
5. Write 0x10 + PWM number to register 11 to load the pulse rate divisor.
6. Calculate the duty cycle value as divisor * duty cycle percent.
7. Write the duty cycle value to registers 8, 9, and 10. Register 8 is the LSB, register 10 is the MSB.
8. Write 0x18 + PWM number to register 11 to load the duty cycle value.
9. For positive output polarity, write 0x20 + PWM number to register 11. For negative output polarity, write 0x28 + PWM number to register 11.
10. Write 0x38 + PWM number to register 11, then write 0x58 + PWM to register 11. These two commands enable the pulse to appear on the associated port F pin.
11. Write 0x70 + PWM number to register 11 to start running the PWM.
12. Write 0x08 + PWM number to register 11 to stop running the PWM.

## 10.12 Configure DIO Port Direction

The digital I/O ports on RMM-1616A-XT are organized as 6 8-bit ports called A-F or 0-5 respectively. On ports A-E, all 8 bits for each port are set for the same direction with a single bit for each port. On port F, each of the 8 bits can have its direction set independently with an 8-bit configuration register.

All digital I/O lines on RMM-1616A-XT power up in input mode. All I/O lines have pull-up/down resistors that can be configured for up or down with a jumper on the board. When a bit is read, it will always return the value on the physical I/O pin. If the bit is in input mode, the pin is being driven either by an external signal or by the pull-up/down resistor. When the bit is in output mode, the pin is being driven by the output register on the board's FPGA, essentially creating a readback operation.

To configure a port's direction, use the following procedure. This procedure assumes non-latched mode. For latched mode instructions, see below.

1. Set page = 2 to select the digital I/O register page by writing 0x02 to register 15 (0x0F) in any page.
2. For ports A-E / 0-4:
   a. Command byte = port number x 4 + direction, where direction = 0 for input or 1 for output
   b. Write command byte to register 6.
3. For port F / 5:
   a. Command byte is an 8-bit value where each bit is 0 or 1 to set the corresponding I/O for input or output respectively. For example, a value of 0xF0 sets bits 7-4 as output and bits 3-0 as input.
   b. Write command byte to register 7.

## 10.13 DIO Input Byte

Byte mode can be used for any of the 6 ports. If port F is being read in byte mode, all 8 bits are returned in a one-byte format, and the individual bit values can be extracted from it. Any bit which is in output mode will return the value in its output register, which is the same as the value appearing on the physical I/O pin.

1. Set page = 2 to select the digital I/O register page by writing 0x02 to register 15 (0x0F) in any page.
2. Read the byte data from register n; n = 0–5 for ports A–F / 0–5.

## 10.14 DIO Output Byte

Byte mode can be used for any of the 6 ports. If a port is in input mode, the written data will be stored in that port's output register but will not affect the pin's physical state. However, if the port is later changed to output mode, the value of the output register will appear on the port's I/O pins. This allows the register to be preloaded with a value before being configured for output.

If port F is being written in byte mode, only the bits that are in output mode will be affected. The same output register preloading condition applies to port F bits.

1. Set page = 2 to select the digital I/O register page by writing 0x02 to register 15 (0x0F) in any page.
2. Write the byte data to register n, where n = 0–5 for ports A–F / 0–5.

## 10.15 Digital Input Bit

Bit mode can be used for any of the 6 ports. Any bit which is in output mode will return the value in its output register, which is the same as the value appearing on the physical I/O pin.

1. Set page = 2 to select the digital I/O register page by writing 0x02 to register 15 (0x0F) in any page.
2. Read the byte data from register n, where n = 0-5 for ports A-F / 0-5.
3. Mask off the desired bit with a bitwise AND function:
   - ♦ Value = register value & 2^(bit number 0-7)
4. Shift the bit into the LSB position to achieve a 1/0 value:
   a. Value = value >> n (0-7)
   b. Alternatively you can simply evaluate the value as zero or nonzero.

## 10.16 Digital Output Bit

Bit mode can be used for any of the 6 ports. If a port is in input mode, the written data will be stored in that port's output register but will not affect the pin's physical state. However, if the port is later changed to output mode, the value of the output register will appear on the port's I/O pins. This allows the register to be preloaded with a value before being configured for output.

1. Set page = 2 to select the digital I/O register page by writing 0x02 to register 15 (0x0F) in any page.
2. Read the byte data from register n, where n = 0-5 for ports A-F / 0-5.
3. Set or clear the desired bit to 1 or 0:
   - ♦ To set a bit, use the operation value = value | 2^n (n = 0-7).
   - ♦ To clear a bit, use the operation value = value & (0xFF – 2^n) (n = 0-7).
4. Write the new byte data to register n, where n = 0-5 for ports A-F / 0-5.

## 10.17 Digital IO Latched Mode

Digital I/O ports A, B, C, and D may operate in latched mode. This mode uses handshaking signals to control the transfer of data between the board and an external device. Latch mode is enabled by setting MODE = 1 when setting the port's direction. To do this, follow the instructions for configuring the digital I/O port direction as described above (10.16 Digital Output Bit), except add 2 to the value written to register 6. All ports with MODE = 1 operate in the same manner. To avoid undefined or undesired behavior, all ports operating in latch mode should have the same direction setting.

When any MODE bit is 1, the port F pins are reassigned as follows:

- Pin F6 is forced to input mode and operates as a latch signal LATCH-, and register bit F6 reads as 0.
- Pin F7 is forced to output mode and operates as an acknowledge signal ACK-, and register bit F7 reads as 0.

### 10.17.1 Latched Mode Input

1. An external device supplies data to the digital I/O pins.
2. The external device then drives LATCH- pin low (DIO pin F6). The falling edge latches data on all ports where MODEn = 1. This sets LATCHn = 1 for all ports where MODEn = 1.
3. Software monitors LATCHn status bits to determine when new data is available. To do this, read bits 3-0 in register 6 of DIO page 2. Bit 3 = port D, bit 2 = Port C, bit 1 = port B, bit 0 = port A.
4. After software reads data, it pulses ACK- low (DIO pin F7) by writing a 1 to DIO bit F7 (refer to 10.16 Digital Output Bit instructions). This clears all LATCHn status bits to 0.

   The cycle is repeated as needed by the application.

### 10.17.2 Latched Mode Output

1. Software writes data to the digital I/O port output register.

1. After software writes data, it pulses ACK- low (DIO pin F7) by writing a 1 to DIO bit F7 (see digital I/O bit output instructions). This clears all LATCHn status bits to 0.

2. External device pulses LATCH- low (DIO pin F6) to acknowledge receipt. This sets LATCHn = 1 for all ports where MODEn = 1.

3. Software waits for LATCHn bit(s) to be 1 before proceeding with new data. This is done by reading bits 3-0 in register 6 of DIO page 2. Bit 3 = port D, bit 2 = Port C, bit 1 = port B, bit 0 = port A.

   The cycle is repeated as needed by the application.

## 10.18 Reading from EEPROM

Reading from the EEPROM can be done very easily. All addresses can be read without any unlock code requirements.

1. Set page = 5 for the EEPROM access page by writing 0x05 to register 15 (0x0F) in any page.
2. Write desired address to register 0. For addresses above 255 write a 1 to register 1 (8th address bit). See the EEPROM information in this manual for the register map showing how calibration information is stored in the EEPROM.
3. Write 0xC0 to register 7 to initiate the read operation.
4. Read register 7 bit 7 (EEBUSY) and wait for it to be 0.
5. Read data from register 4.

## 10.19 Writing to EEPROM

Writing to the EEPROM requires writing an unlock code beforehand. This is done in order to prevent accidental writes that could destroy calibration data stored in the EEPROM. The EEPROM has 512 bytes. Addresses 0-255 are accessible to the user. Addresses 256-511 are reserved for the factory backup data and are not intended for user access.

1. Set page = 5 for the EEPROM access page by writing 0x05 to register 15 (0x0F) in any page.
2. Before any write operation is allowed, the unlock sequence must be performed.
3. Write 0xA5 to register 8.
4. Write 0x24 to register 8.
5. The EEPROM may now be written to. Only addresses 0-255 are accessible. Write the desired address to register 0. Register 1 must be 0 to select an address in the range 0-255.
6. Write the data to register 4.
7. Write 0x80 to register 7 to initiate the write operation.
8. Read register 7 bit 7 (EEBUSY) and wait for it to be 0.

The write operation is complete.

Additional writes can be done without rewriting the unlock code. Write access is enabled until the board is reset or reinitialized.

## 10.20 Recall Calibration Values

When changing from one analog output range group to another, it is recommended to recall the calibration settings for the new range for best accuracy. The procedure is shown below.

1. Read C and M values from EEPROM: See the Reading from EEPROM section above. Read the 4 calibration bytes from the EEPROM for the channel and output range you need (C register LSB, C register MSB, M register LSB, M register MSB).
2. Write the C value to the selected channel:
3. Set page = 5 by writing 0x05 to the register 15 (0x0F).
   a. Write the LSB to register 0, then write the MSB to register 1.
   b. Write the C register write command to register 2. See information below.
4. Write the M value to the selected channel:
   a. Write the LSB to register 0, then write the MSB to register 1.
   b. Write the M register write command to register 2. See information below.

New settings do not take effect immediately. The next channel update command will implement the new settings.

C register write command = 0x10 + chip number * 0x20 + channel number.

Binary format: 0 <chip no. 0-3> 1 0 0 <DAC channel 0-3>

- Example: for channel 1, write 0x11 (0x10 + 0 * 0x20 + 1).
- Example: for channel 15, write 0x73 (0x10 + 3 * 0x20 + 3)

M register write command = 0x08 + chip number * 0x20 + channel number.

Binary format: 0 <chip no. 0-3> 0 1 0 <DAC channel 0-3>

- Example: for channel 1, write 0x09 (0x08 + 0 * 0x20 + 1).
- Example: for channel 15, write 0x6B (0x08 + 3 * 0x20 + 3).

# 11. SOFTWARE DRIVER OVERVIEW

## 11.1 Universal Driver

Diamond Systems' free Universal Driver software provides a powerful and efficient method of accessing all of the features and capabilities of RMM-1616A-XT. While many I/O boards are very simple in operation and can be controlled easily with simple I/O commands, the D/A, counter/timers, PWM, and interrupt features of RMM-1616A-XT are sophisticated and require complex setup and management to utilize their full capabilities. The how-to section of this manual provides an overview of the most common uses of the board which can suffice for many applications if you are using an operating system not supported by Universal Driver. However if you are using an OS supported by Universal Driver it is far more efficient and effective to use Universal Driver. The Universal Driver software and documentation are available online at www.diamondsystems.com/products/dscud. The driver package consists of several downloadable modules: The main driver software package, the board-specific demo programs, and the documentation.

Universal Driver provides a collection of high level software functions in a library compatible with popular C language compilers. The functions include board initialization, D/A operations, D/A waveform generator, digital I/O, counter/timer operations, PWM functions, and interrupts. Universal driver lets you utilize all the features of RMM-1616 quickly and efficiently without having to deal with register-level commands or complex sequences of operations. A collection of demo programs provide illustration of the use of the driver functions and also provide a convenient starting point for your own application development.

## 11.2 GUI demo

Included with Universal Driver is a Windows GUI program (graphical user interface) that runs in Windows XP and 7. The GUI demo provides a quick way to access all the features of the board in a simple and intuitive manner. The GUI is highly recommended as a starting point when using the RMM-1616. It can be used to verify that the board is installed and operating correctly as well as to become familiar with the various capabilities of the board. The GUI demo is included with the main Universal Driver package.

# 12. SPECIFICATIONS

| | |
|---|---|
| **Host Interface** | |
| Interface bus | PC/104 8-bit ISA bus or PC/104-*Plus* PCI bus |
| Bus selection | Autodetection with PCI priority; manual ISA override |
| **Analog Outputs** | |
| Number of outputs | 4, 8 or 16 |
| Resolution | 16 bits |
| Output ranges | 0-5V, 0-10V, ±5V, ±10V, 0-20mA, 0-24mA, 4-20mA |
| Settling time | 10us maximum to ±.003% |
| Linearity error | ±2 LSB maximum |
| Differential nonlinearity | ±2 LSB maximum |
| Monotonicity | 15 bits minimum |
| Output current limit (voltage outputs) | 8mA or 16mA, software selectable per channel |
| Open-circuit voltage limit (current outputs) | 23V, 24.5V, 27V or 29.5V, software programmable per channel |
| Reset | All DACs reset to tristate on voltage and current output pins<br>Programmable clear code, independent value per channel |
| **Waveform Generator** | |
| Number of channels | 1 to 16 |
| Buffer size per channel | 1 to 2048 samples, software programmable<br>2048 total sample limit |
| Trigger source | Software, counter/timer or external signal, software selectable |
| **Digital I/O** | |
| Number of lines | 40 byte-wide, 8 bit-wide, programmable direction<br>TTL/CMOS compatible |
| Input voltage | Logic 0: -0.5V min, 0.8V max<br>Logic 1: 2.0V min, 5.5V max |
| Output voltage | Logic 0: 0.0V min, 0.4V max<br>Logic 1: 3.0V min, 4.6V max |
| Output current | ±2.5mA maximum per line |
| I/O data transfer modes | Normal or latched, software selectable |
| **Counter/Timers** | |
| Number of counters | 2 32-bit, software programmable |
| Clock sources | 50MHz, 5MHz or external signal, software selectable |
| Count direction | bidirectional, software programmable; optional automatic reload at 0 for rate generator use |
| Clock source | Internal or external, software selectable |
| Gate | Programmable counter gate for each counter |

| **PWMs** | |
|---|---|
| Number of PWM channels | 4 24-bit, software programmable period and on-time |
| Duty cycle | 0 – 100%, programmable |
| Output polarity | Positive or negative, programmable |
| Clock sources | 50MHz or5MHz, software selectable |
| **Interrupts** | |
| Sources | Counter/timer, digital I/O or D/A fault, software selectable |
| Software support | Universal Driver supports execution of user-supplied code on interrupt |
| **Mechanical** | |
| Dimensions | 3.550” x 3.775” / 90.2 x 95.9mm; PC/104 compliant |
| Component height | Top side: 0.435” / 11.2mm max<br>Bottom side: 0.1” / 2.5mm max (not including PC/104 connector pins and PC/104-*Plus* shroud) |
| PC/104 stacking | Compatible with both top and bottom stacking |
| I/O connectors | Dual 50-pin 2x25.1” pitch right angle pin headers; compatible with industry standard 50-pin ribbon cables |
| Weight | 3.0oz (85g) |
| **General** | |
| Operating temperature | -40°C to +85°C |
| Input power | +5VDC +/-5% |
| MTBF | 100,000 hours (calculated) |
| RoHS | Compliant |