Technical Brief:

Using the RMC Discrete I/O Input to Event Mode

DELTA COMPUTER SYSTEMS, INC.
11719 NE 95th Street, Suite D
Vancouver, WA 98682-2444
Telephone 360-254-8688
Fax 360-254-5435
www.deltacompsys.com

Abstract

The RMC-DI/O is capable of sophisticated motion control using small and inexpensive Programmable Controllers with simple discrete I/O. An RMC with a DI/O communications interface is capable of three discrete I/O interfaces: Command Mode, Input to Event Mode, and Parallel Position Mode. Of these three communication modes, Input to Event mode is the simplest for many repetitive-process applications; many applications may not require a programmable controller.

This technical brief will compare the three discrete I/O interfaces of the RMC100 product-line, describe implementing Input to Event Mode, and finally provide a sample application using Input to Event Mode.

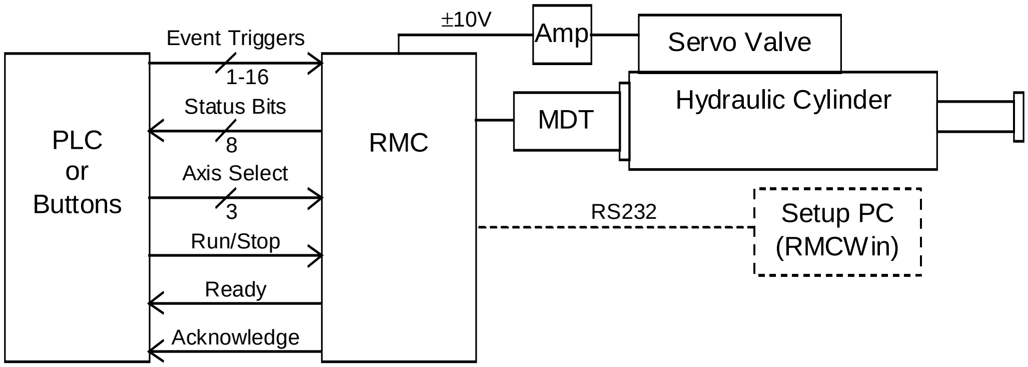
DI/O Communication Mode Comparison

The following chart lists the advantages and disadvantages of each communication mode. Each word or phrase in bold print appears in RMCWin’s online help index.

Interface Mode	Advantages	Disadvantages
Input to Event Mode	• Does not require a PLC • Multiple axes may be given commands from a single input • Allows use of Event Control feature	• Returns only a **Halted Bit**—indicating an error has occurred—and an **In Position Bit** per axis • Issues only **Start Event** commands
Parallel Event Mode	• Up to four axes may be commanded at once • PLC or thumb-wheel switches may be used • Allows use of Event Control feature	• Requires parallel inputs • Returns only a **Halted Bit**—indicating an error has occurred—and an **In Position Bit** per axis • Sequence must be pre-programmed in Event Control
Parallel Position Mode	• Any position can be moved to in a single PLC scan • Any open loop drive can be triggered in a single PLC scan • Multiple profiles can be selected for any move	• Requires a PLC • Returns only a **Halted Bit**—indicating an error has occurred—and an **In Position Bit** per axis • Issues only **Go** and **Open Loop** commands • Gives commands to only one axis per scan
Command Mode	• Any RMC command can be issued • Any status information can be retrieved (including **Actual Position**, **Actual Speed**, **Drive**, **Error Bits**, and other **Status Bits**)	• Requires a PLC • Requires 2 PLC scans per command • Gives commands to only one axis per command cycle

Implementation

The following diagram shows the electrical control connections of a single-axis hydraulic system using RMC-DI/O in Input to Event Mode (the PC and its RS232 cable are needed only during setup):



This document discusses only the connections between the Programmable Controller and the RMC. For details on the transducer and drive wiring, look up **Wiring Notes** in the RMCWin index.

Input to Event mode requires use of the **Event Control** feature of the RMC. You should first familiarize yourself with **Event Control** and the **Event Step** table. The RMCWin online help is a good reference for this information.

Input to Event mode does not require multiple inputs to be switched simultaneously. For this reason, many applications can use the RMC without a controlling PLC. The following points describe the operation of Input to Event mode:

- Parameters and tables used by the RMC are configured using the RMCWin software and stored in the RMC **FLASH memory**.
- All axes are initialized by raising the **Run/Stop** (CPU input 0) line on the RMC, and all axes are immediately stopped by the falling of the **Run/Stop** line. Therefore, the **Run/Stop** input is often connected to the system's emergency-stop button.
- The **Ready** (CPU output 0) line echoes the **Run/Stop** input line to notify the controlling system when the RMC is initialized.
- When the **Run/Stop** line is raised and a **Trigger** (DI/O inputs 0-15) input rises or falls, the **Input to Event table** is used to determine which axes receive an event sequence.

  The Input to Event table has a row for each edge (rising and falling) of each **Trigger** input (32 rows total). Each of these rows has a field for each axis; each field may either be left blank or hold an **Event Step** number (0 to 255).

  When an edge occurs on a **Trigger** input, each axis's field in the row corresponding to the input edge that took place is checked. For each field containing a valid value, the axis starts executing the event sequence beginning with the **Event Step** number given by the **Input to Event table** entry.

- Each time the **Run/Stop** or a **Trigger** input transitions, the **Acknowledge** (CPU output 1) line toggles. Stand-alone applications generally ignore this, but PLC-controlled applications may wish to use this signal.
- The eight **Status** (DI/O outputs 0-7) outputs default to being used for the following:

DI/O Output #	2-4 Axis RMC's	5-8 Axis RMC's
0	Axis 0 In Position	Axis 0 In Position
1	Axis 1 In Position	Axis 1 In Position
2	Axis 2 In Position	Axis 2 In Position
3	Axis 3 In Position	Axis 3 In Position
4	Axis 0 Stop on Error	Axis 4 In Position
5	Axis 1 Stop on Error	Axis 5 In Position
6	Axis 2 Stop on Error	Axis 6 In Position
7	Axis 3 Stop on Error	Axis 7 In Position

Using the RMCWin software, these outputs may be marked to be user-controlled instead of being used for the above default assignments. User-controlled outputs are set and cleared from the **Event Step table**.

The following features of Input to Event mode add complexity to the mode, and are necessary only for a small number of applications:

- Using RMCWin, any of the **Trigger** inputs can be marked as **Single-axis** inputs. When an edge occurs on a **Single-axis** input, only one field in the **Input to Event table** is used. This field is given by the row of the input edge that took place and the axis selected by the three **Axis Select** inputs (CPU input 1, DI/O inputs 16-17).

NOTE: This feature is not required in most cases where only a single axis is to be affected by an input. Most often this can be done by leaving all axes' event step numbers blank in the **Input to Event table** except the one axis that is to be affected.

The following table shows how to select an axis with these inputs:

Desired Axis	DI/O Input 17	DI/O Input 16	CPU Input 1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

- Using RMCWin, any of the **Trigger** inputs can be marked as **Non-linked** inputs. When an edge occurs on one of these inputs, the **Input to Event table** is still referenced to find the axes that have **Event Step** numbers. However, instead of starting an event sequence on the axis, *only the command* from the event step is given to the axis and *any pre-existing event sequence continues execution*.

The following general steps must be taken to set up a system using Input to Event mode:

1. **Design the System**

   Designing the system begins with selecting the appropriate method of communication. First, decide whether one of the RMC's field bus solutions fits your application, and if you decide to use digital I/O, then decide which of the communication modes fits your application.

   If you decide to use digital I/O using Input to Event mode, then you must design the wiring of the system, the event step table, and the input to event table.

2. **Program the Event Step Table**

   Programming the Event Step table is described in the RMCWin online help. The event step table will hold the majority of the controlling logic. Be sure to save your Event Step table both in the RMC FLASH memory, and also on disk from RMCWin.

3. **Program the Input to Event Table**

   Programming the Input to Event table is described in the RMCWin online help. This table serves the purpose of mapping edges of inputs to event sequences in the Event Step table. Be sure to save your Input to Event table both in the RMC FLASH memory, and on disk from RMCWin.

4. **Configure the RMC Communication**

   The following steps are required to configure the Communication DI/O from RMCWin:

   - Connect the serial port to the module being configured.
   - On the **Tools** menu, click **Module Configuration**.
   - In the **Slots** list, select the **Communication DI/O** item, and then click **Slot options**.
   - In the **DI/O mode** list, select **Input to Event** mode.
   - If you need to invert any inputs or outputs to match your hardware, select the appropriate check boxes in the **Invert inputs** and **Invert outputs** areas.
   - If you wish to use the Communication DI/O counter feature, click either **Edge** or **Quadrature** under **Counter Type**. See **Using Counters** in the RMCWin online help for more details.
   - Click **Mode options**.
   - If you want to define one or more inputs as either single axis or non-linked, then use the check boxes in the **Single-axis inputs** and **Non-linked inputs** areas.
   - If desired, you may select to control any of the outputs by checking the appropriate **User-controlled outputs** check boxes.

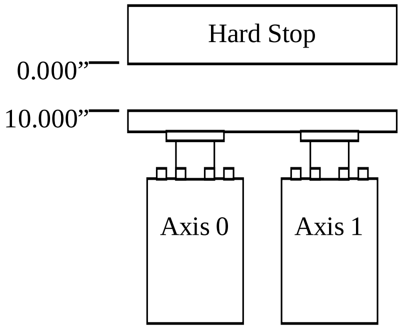
- Click **OK**.
- Click **Update RMC**.
- The **Update Module Configuration** dialog box will be displayed to indicate the progress. If the module could not be reset manually, you may be prompted to reset the module yourself.

5. **Wire, Test, and Tune the System**

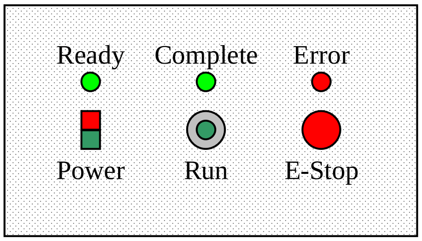
   Wiring and testing should follow your design. Tuning the system is described in the RMCWin online help. Be sure to save your tuning parameters both in the RMC FLASH memory, and also on disk from RMCWin.

Sample Application

In this application a user wants to use a motion controller to precisely control a pair of hydraulic cylinders to the 1/1000th of an inch resolution. The cylinders are part of a press and are tied together at the surface of the press. The user wishes to have the cylinders begin at 10.000", extend to 0.250", and retract back to 10.000", as shown in the following diagram:



The following simple control panel will be used for this application:



The buttons and indicators are to operate as follows:

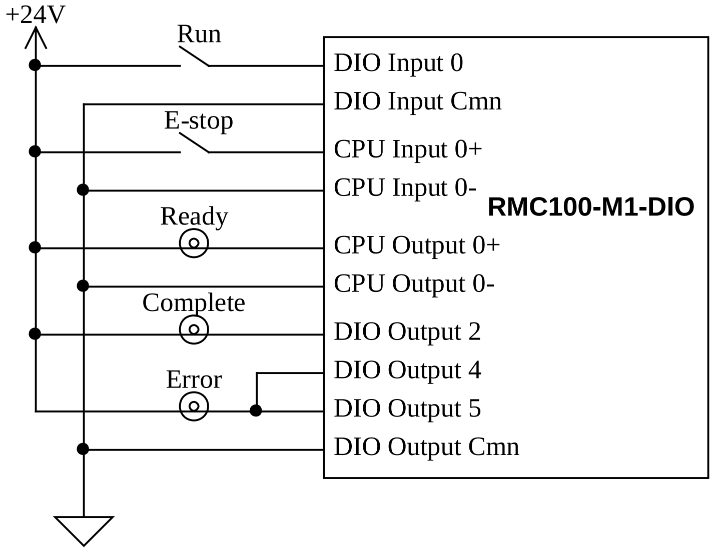
- The **E-stop** button—closed when pulled, and open when pushed—must be pulled in order for the system to run. When pressed, the system will halt immediately.
- The normally-open **Run** button starts the extension and retraction cycle of the cylinders.
- The green **Ready** light indicates when power is supplied and the E-stop is not pushed.
- The green **Complete** light indicates when the cycle is completed.
- The red **Error** light indicates that the cylinders stopped because of an error (for example, perhaps the cylinders lagged behind the target positions by too much).

1. Design the System

This application requires two axes, and we will assume that the position feedback is given by magnetostrictive displacement transducers (MDT's). Therefore, the appropriate module to use is the **RMC100-M1-DI/O**.

The RMC controls based on the transducer or encoder counts received. However, counts rarely correspond to meaningful engineering units. Therefore, the RMC allows the user to convert counts into meaningful position units by using **Scale** and **Offset** parameters. Refer to RMCWin's online documentation for details. In this application, the position units used will be thousandths of an inch. Therefore, speeds will be given in thousandths of an inch per second.

The following wiring diagram would allow the above control panel to work with the RMC in Input to Event mode:

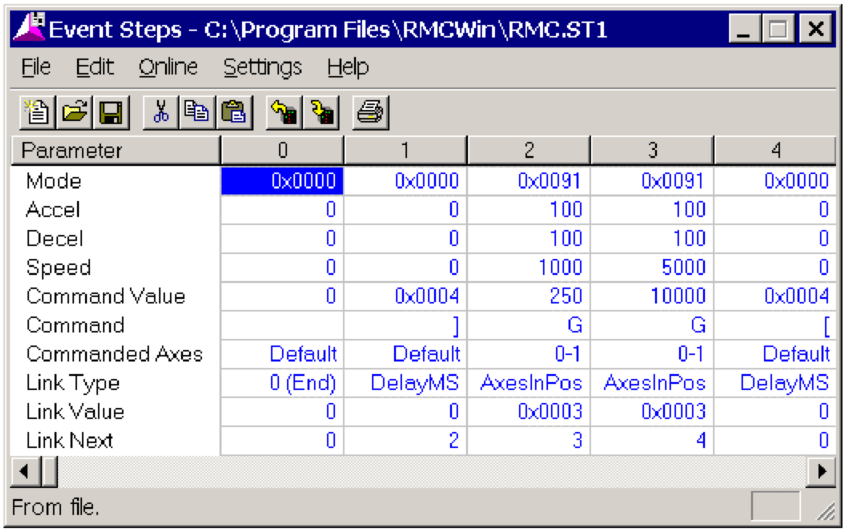


Simply by wiring the system, many of the controls are already handled:

- The **E-stop** button enables or halts the axes as wired into the **Run/Stop** (CPU input 0) input of the RMC.
- The **Ready** light works as specified because the **Ready** (CPU output 0) output on the RMC is on when the system is not halted by the **Run/Stop**.
- The **Error** light works as specified because it combines the Stop on Error status outputs (DI/O outputs 4 and 5) of the two axes. Therefore, when either axis has an error, the light will illuminate.

2. Program the Event Step Table

We now need to write an event sequence that will generate the desired sequence on the two axes and turn on DI/O output 2 to illuminate the **Complete** light at the end. A complete listing of available commands can be obtained from RMCWin's online help. The following event sequence will be used to control both axes:



Parameter	0	1	2	3	4
Mode	0x0000	0x0000	0x0091	0x0091	0x0000
Accel	0	0	100	100	0
Decel	0	0	100	100	0
Speed	0	0	1000	5000	0
Command Value	0	0x0004	250	10000	0x0004
Command		]	G	G	[
Commanded Axes	Default	Default	0-1	0-1	Default
Link Type	0 (End)	DelayMS	AxesInPos	AxesInPos	DelayMS
Link Value	0	0	0x0003	0x0003	0
Link Next	0	2	3	4	0

From file.

Each step is described below, beginning by the first step that will be executed:

Step 1: A **Reset Outputs** (]) command is issued. Its hexadecimal Command Value indicates which bits to reset. Therefore, bit 2 (hexadecimal 00004) represents DI/O output 2. The **DelayMS (D)** link type and its 0 link value indicate to delay 0 milliseconds and then go to step 2.

Step 2: Both axes are commanded to **Go** (G) to 250 position units. The **AxesInPos (A)** link type waits until axes 0 and 1 are both in position, and then goes to step 3.

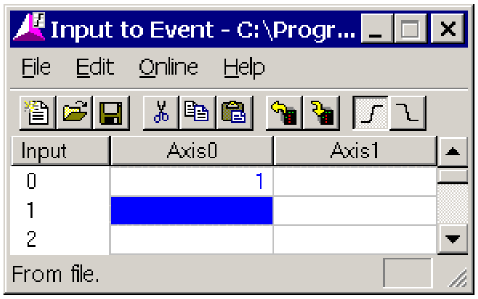
Step 3: Both axes are commanded to **Go** (G) to 10,000 position units. The **AxesInPos (A)** link type waits until axes 0 and 1 are both in position, and then goes to step 4.

Step 4: A **Set Outputs** ([) command is issued, which turns on DI/O output 2. The event sequence then ends by going back to event step 0.

Step 0: This step ends the event step sequence because it has no link type.

3. Program the Input to Event Table

The final step is to cause DI/O input 0 to trigger the event sequence shown. This is done with the following simple Input to Event table:



Input	Axis0	Axis1
0	1	
1		
2		

This single table entry causes axis 0 to start the event sequence beginning with step 1 whenever DI/O input 0 has a rising edge.

4. Configure the RMC Communication

The steps required for this procedure are described in the **Implementation** section of this Technical Brief. For this application, you should not need to use (1) inverted inputs, (2) inverted outputs, (3) counters, (4) single-axis inputs, (5) non-linked inputs, or (6) user-controlled outputs.

5. Wire, Test, and Tune the System

The system should be wired as described in the design above. Test the functionality of the final system, and finally tune the system as described in the RMCWin online help.

Reference

Throughout this technical note, references are made to RMCWin online help index entries. To obtain the RMCWin software package, contact Delta Computer System's web site (www.deltacompsys.com).