

Programming Example Kinetix 3 with the Micro 830 and the Connected Component Workbench

Introduction

This sample code example shows three methods to control a Kinetix 3 drive with a Micro830 PLC using Modbus-RTU.

This document will cover the following items:

- Basic drive setup.
- Communication connections for Kinetix 3 and Micro830.
- Brief explanation of the program code.

Choose which axes to use and then add the global variables and the programs associated with those axes. These are the axes that are given as examples:

Axis01: The status word for this axis is monitored in the controller, and the command word for this axis is set in the controller and transmitted over Modbus-RTU to the drive. The drive will ignore any behavior on the physical digital inputs. This axis is configured as an indexer, and moves through a series of indices when running.

Axis02: The status word for this axis is monitored in the controller, and the command word for this axis is set in the controller and transmitted over Modbus-RTU to the drive. The drive will ignore any behavior on the physical digital inputs. This axis is configured for preset velocity, and will cycle through five seconds at each of three different preset speeds.

Axis03: The status word for this axis is monitored in the controller however the command word is ignored. The commands for this drive are issued through the physical digital inputs of the drive. Those inputs could be hooked up to sensors, the controller, or another drive's outputs. This increases wiring and it results in faster execution at the scan rate instead of the at the Modbus transfer rate.

Once the axes have been imported, modify the code or add programs as needed to see drive status, trigger events, and control all the axes.

Basic Drive Setup

Install, connect, and commission the drive per user manual 2071-UM001. The user manual will step through using Ultraware online with the drive, configure the drive for the installed motor/feedback and autotuning.

Note: Ultraware V1.80 or later is required. It is also recommended that the latest version of drive firmware is used.

In addition to the basic setup, the following steps are specific to this sample code. Although the drive configuration is accomplished using Ultraware, parameter numbers are included if the user chooses to use the integral keypad.

Included with the sample code is an Ultraware file of the three drives configured to run with this sample code.

Note: The Ultraware file does not show the drive serial communications as Modbus. This is because the drive is connected to Ultraware when the file was saved.

Set Operations Mode

On the **Drive** tab, set **Operations Mode (Main/Override)** to these modes:

- Axis01: Indexing/None
- Axis02: Preset Velocity/None
- Axis03: Indexing/None

Set Modbus Control

On the **Drive** tab under **Communications**, set **MODBUS Run Function Control** to “Enable” for all three drives and **MODBUS Input Function Control** to these modes:

- Axis01: Enable
- Axis02: Enable
- Axis03: Disable

Index and Velocity Configuration

The following configures the index and velocity values for the drives, but your application code will likely differ.

Note: The motor in the example has an absolute encoder with 131072 counts/rev. Velocity is in RPM. For Axis01, on the **Indexing** tab under **Mode Configuration**, configure **Indices 0-3**:

	Index 0 Setup	Index 1 Setup	Index 2 Setup	Index 3 Setup
Mode	Absolute	Incremental	Incremental	Incremental
Distance	0	131072	262144	-65536
Dwell	0	2000	1500	0
Velocity	750	100	50	200
Accel	62.50	62.50	62.50	62.50
Decel	62.50	62.50	62.50	62.50
Next Index	0	2	3	0
Action When Complete	Stop	Start Next	Start Next	Stop

The following table shows the configuration for the Axis02 sample code.

On the **Preset** tab under **Mode Configuration**, set Preset Speeds:

- Preset Velocity 1: 100 RPM
- Preset Velocity 2: 250 RPM
- Preset Velocity 3: 500 RPM
- Preset Velocity 4: 400 RPM

- Preset Velocity 5: 500 RPM
- Preset Velocity 6: 600 RPM
- Preset Velocity7: 700 RPM

Note: The motor in the example has an absolute encoder with 131072 counts/rev. Velocity is in RPM
For Axis03, on the **Indexing** tab under **Mode Configuration**, configure **Index 0:**

	Index 0 Setup
Mode	Incremental
Distance	0
Dwell	0
Velocity	750
Accel	62.50
Decel	62.50
Next Index	0
Action When Complete	Stop

Note: The motor in the example has an incremental encoder with 2000 lines/revolution and an interpolation factor of 4. Velocity is in RPM.

Homing Configuration

Homing is required before the drive will run an Index. The two drives shown in this example in indexing mode (Axis01 and Axis03) have different homing modes to illustrate the benefit of using the digital inputs for run control instead of Modbus Run Control.

On the **Homing** tab under **Mode Configuration**, configure only the following Homing parameters:

	Axis01	Axis03
Homing Type	Home to Marker	To Home sensor/Fwd to Marker
Auto Start Homing	Inactive	Inactive
Homing Velocity	100 RPM	100 RPM
Home Sensor Polarity	Active-Going Transition	Active-Going Transition

Configure Digital Inputs

Though Axis01 and Axis02 will not be able to read from the digital inputs while the **Modbus Input Function** is *Enabled*, the pins must be set correctly to avoid a warning on the drive display. The inputs for Axis03 must be wired according to the controller outputs as well.

On the **Digital Inputs** tab, configure the following parameters:

	Axis01	Axis02	Axis03
Input 1	Drive Enable	Drive Enable	Drive Enable
Input 2	Fault Reset	Fault Reset	Fault Reset
Input 3	Moving Enable	Moving Enable	Homing Sensor
Input 4	Unassigned	Preset Select 1	Start Homing

Input 5	Unassigned	Preset Select 2	Start Indexing
Input 6	Unassigned	Preset Select 3	Index Select 0
Input 7	Unassigned	Preset Direction	Stop Homing
Input 8	Unassigned	Unassigned	Stop Indexing

Set Serial Port Configuration

The drive serial port configuration can be viewed with Ultraware. However changing one of the parameters will cause a communication loss. After the configuration is complete in Ultraware, it is recommended that the serial port configuration and node address be set through the drive keypad.

Note: Only one drive will be visible in Ultraware at a time.

Pr0.09 Serial Port Configuration

Kinetix 3 Drive serial port configuration for communication with a PC running Ultraware:

- Set to 0x0005. (0x0005 – Default)
- 0x5 – Digit 0 Baudrate - 57600 bps
- 0x0 – Digit 1 8 Data, No Parity, 1 Stop bit
- 0x0 – Digit 2 Protocol ASCII
- 0x0 – Digit 3 RS232 – 0

Kinetix 3 Drive serial port configuration for Modbus communication with Micro830 controller:

- Set to 0x1102. (0x0005 – Default)
- 0x2 – Digit 0 Baudrate - 19200 bps
- 0x0 – Digit 1 8 Data, No Parity, 1 Stop bit
- 0x1 – Digit 2 Protocol MODBUS-RTU
- 0x1 – Digit 3 RS485 - 1

Pr0.07 Drive Address

- Axis01 – 1
- Axis02 – 2
- Axis03 – 3

Kinetix 3 communication Connections

The Kinetix 3 communication port can be configured for RS-232 communications for Ultraware or RS-485 Modbus RTU.

3 cables are available for serial communication with the Kinetix 3.

2090-CCMPCDS-23AAxx

Kinetix 3 Comm0A or Comm0B to PC 9-pin serial port for RS-232 programming with Ultraware.

2090-CCMCNDS-48AAxx

Kinetix 3 Comm0A or Comm0B to Micro830 embedded serial port (channel 1) for RS-485 Modbus RTU. This is not a recommended connection since this serial port is not isolated and is not recommended for connection to variable frequency drives. This cable can be modified to connect to the 2080-SERIALISOL plug-in module.

2090-CCMDSDS-48AAxx

Kinetix 3 Comm0A or Comm0B to Kinetix 3 Comm0A or Comm0B for multi-drive RS-485 Modbus RTU. This cable can be modified to connect to the 2080-SERIALISOL plug-in module.

Note: The configuration of the serial port includes both Comm0A and Comm0B. Changing between RS-232 and RS-485 Modbus requires different cabling and serial port configurations. If the Kinetix 3 is the only device on the network or the last device on the network, the 485 Term switches on the front of the drive need to be set "ON".

Basic Controller Setup

If a different controller is used than the 2080-LC30-24QBB base in the example project, then the controller will need to be configured. Add a 2080-SERIALISOL module to one of the slots. Set the parameters:

- Driver: Modbus RTU
- Baud Rate: 19200
- Parity: None
- Unit Address: 0
- Modbus Role: Modbus RTU Master

Under Advanced Settings for the 2080-SERIALISOL, set:

- Media: RS485

Sample Code

The 2080-LC30-24QBB is used in the example but can be converted to another controller in the Micro830 family.

There are three programs in the sample code and each program contains most of its variables locally, with a few variables that are stored globally. This code is designed to work together, but also so that each program can be imported independently of the others. Therefore, each program references some global variables that may need to be changed, depending on the application. Make sure that any references to global variables in the program correspond to the correct variable.

Within each of the programs, some of the local variables may need configuring. For any variable of the type MODBUSLOCPARA, the Channel must be configured to the location of 2080-SERIALISOL (with the

first slot as 5, the second slot as 6, etc). For any local variable of the type MODBUSTARPARA, the Node must be changed to match Pr0.07 in the drive.

Here are the global variables common to all of the programs:

Name	Description	Data Type	Dimension
AxisXX_Comm_Fault	Indicates a fault in any of the message instructions sent to that drive	BOOL	
AxisXX_Cmd	Represents the command word of that Kinetix 3 drive	BOOL	[0..47]
AxisXX_Sts	Represents the status word of that Kinetix 3 drive	BOOL	[0..31]
HMI_Fault_Reset	Global command to reset alarms on the drives from a user interface	BOOL	
HMI_Run	Global command to start the run sequence for the drives from a user interface	BOOL	
HMI_Start_Homing	Global command to start homing from a user interface	BOOL	
HMI_Stop	Global command to stop all motion from a user interface	BOOL	
Global_MSG_IP	Prevents flooding the buffer by only allowing one message instruction at a time.	BOOL	

The following variables are global, but are only used by Axis03. They would be used to represent a registration sensor and distance.

Name	Description	Data Type	Dimension
Inp_Trigger	Represents a digital input that would be wired to a registration sensor	BOOL	
Distance	Represents a calculated registration distance that the drive will move.	DINT	

Axis01 Program

- **Fault Reset** – This rung holds the alarm reset bit high for 2 seconds, approximately double the time for a full message cycle at worst-case conditions [2 * (Write Timeout + Write Delay + Read Timeout + Read Delay)].
- **Homing Sequence** – This sequence will check the ready status of the drive, then enable the drive and start homing. When homing is complete, the drive will be disabled again.
- **Running Sequence** – This sequence will check the home status and ready status of the drive, then enable the drive and set the bits for Index 1. After a 2 second delay, Index 1 is started. When Index 3 is complete, the drive will wait two seconds and then start Index 0. When Index 0 is complete, Index 1 is initiated again after a 2 second delay, and the process repeats.

- **Stop Sequence** – This sequence stops the motor and disables the drive after motion has come to a stop.
- **Fault Stop and Warning Detection** – If a fault is detected in the drive, end all of the sequences and reset the control bits.
- **Communications Fault Stop** – If a communications fault occurs end all of the sequences and reset the control bits. This may not actually stop the drive since a communications fault could be persistent (such as a broken cable). Additional branches can be added along with wiring to the physical ESTOP input of the drive to stop motion if communications fails.
- **Communications Sequence** – Copies the command word to the message array and then sends the command message. After the programmed delay, reads the status message and copies the message array to the status word. Repeats the cycle after another programmed delay. To accommodate other traffic on the Modbus network the spacing between messages is 250 ms.
- **Error Monitoring and Fault Reading** – If an alarm or warning is present in the drive, reads the code. Also monitors for communications faults due to timeouts or errors.
- **Status and Command Bits** – Each bit is copied to the command word or read from the status word.

Axis02 Program

- **Fault Reset** – This rung holds the alarm reset bit high for 2 seconds, approximately double the time for a full message cycle at worst-case conditions [$2 * (\text{Write Timeout} + \text{Write Delay} + \text{Read Timeout} + \text{Read Delay})$].
- **Running Sequence** – This sequence will check the ready status of the drive, then enable the drive and set the bits Preset Velocity 1. After 5 seconds within the speed window, Preset Velocity 2 is started. After 5 seconds within the speed window, Preset Velocity 3 is started. After 5 seconds in the velocity window, it loops back to Preset Velocity 2.
- **Stop Sequence** – This sequence stops the motor and disables the drive after motion has come to a stop.
- **Fault Stop and Warning Detection** – If a fault is detected in the drive, end all of the sequences and reset the control bits.
- **Communications Fault Stop** – If a communications fault occurs end all of the sequences and reset the control bits. This may not actually stop the drive since a communications fault could be persistent (such as a broken cable). Additional branches can be added along with wiring to the physical ESTOP input of the drive to stop motion if communications fails.
- **Communications Sequence** – Copies the command word to the message array and then sends the command message. After the programmed delay, reads the status message and copies the message array to the status word. Repeats the cycle after another programmed delay. To accommodate other traffic on the Modbus network the spacing between messages is 250 ms.
- **Error Monitoring and Fault Reading** – If an alarm or warning is present in the drive, reads the code. Also monitors for communications faults due to timeouts or errors.
- **Status and Command Bits** – Each bit is copied to the command word or read from the status word.

Axis03 Program

- **Fault Reset** – This rung holds the alarm reset bit high for 50 ms. Since the fault reset signal is sent through the digital I/O, it can respond much more quickly.
- **Homing Sequence** – This sequence will check the ready status of the drive, then enable the drive and start homing. The home sensor in this case is triggered by the controller. When homing is complete, the drive will be disabled again.
- **Running Sequence** – This sequence will check the home status and ready status of the drive, then enable the drive and set the bits for Index 0. When the input trigger becomes active, the new distance is copied to the drive with a message instruction, and upon message completion, the updated index is executed, before cycling back to wait for the next index.
- **Stop Sequence** – This sequence stops the motor and disables the drive after motion has come to a stop.
- **Fault Stop and Warning Detection** – If a fault is detected in the drive, end all of the sequences and reset the control bits.
- **Communications Fault Stop** – If a communications fault occurs end all of the sequences and reset the control bits. Additional branches can be added along with wiring to the physical ESTOP input of the drive to stop motion if communications fails. The delay is set to 100 ms. This axis is assumed to need more accurate data, so it cycles faster than the others.
- **Communications Sequence** – Reads the status message and copies the message array to the status word. Repeats the cycle after another programmed delay.
- **Error Monitoring and Fault Reading** – If an alarm or warning is present in the drive, reads the code. Also monitors for communications faults due to timeouts or errors.
- **Status and Command Bits** – Each bit is copied to the embedded I/O or read from the status word.

Disclaimer

This sample code delivered by Rockwell Automation is just an example of how to control a KINETIX 3 drive.

Disclaimer Of Warranty

THE MATERIALS DELIVERED WITH THIS EXAMPLE ARE PROVIDED "AS IS" WITHOUT WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OR OTHER VIOLATION OF RIGHTS. ROCKWELL AUTOMATION DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING THE USE, VALIDITY, ACCURACY, OR RELIABILITY OF, OR THE RESULTS OF THE USE OF, OR OTHERWISE, RESPECTING THE MATERIALS MAINTAINED ON OR ACCESSED BY WAY OF THIS SITE OR ANY WEB SITES LINKED TO THIS SITE. FURTHERMORE, ALL WARRANTIES, CONDITIONS, REPRESENTATIONS, INDEMNITIES AND GUARANTEES WITH RESPECT TO THE ACCURACY, OPERATION, CAPACITY, SPEED, FUNCTIONALITY, QUALIFICATIONS, OR CAPABILITIES OF THE SOFTWARE, SYSTEMS AND SERVICES COMPRISING OR UTILIZED IN THE OPERATION OF THIS CODE, WHETHER EXPRESS OR IMPLIED, ARISING BY LAW, CUSTOM, PRIOR ORAL OR WRITTEN STATEMENTS BY ROCKWELL AUTOMATION, OR OTHERWISE (INCLUDING, BUT NOT LIMITED TO ANY WARRANTY OF SATISFACTORY QUALITY, MERCHANTABILITY, FITNESS FOR PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT) ARE HEREBY EXPRESSLY EXCLUDED AND DISCLAIMED.

Disclaimer Of Liability

UNDER NO CIRCUMSTANCES (INCLUDING NEGLIGENCE) WILL ROCKWELL AUTOMATION BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION, BUSINESS INTERRUPTION, DELAYS, LOSS OF DATA OR PROFIT) ARISING OUT OF THE POSTING (OR THE INABILITY TO POST) OR THE USE (OR THE INABILITY TO USE) THE DELIVERED CODE EVEN IF ROCKWELL AUTOMATION HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.