EMERSON
Industrial Automation

Application Note CTAN #262

The Application Note is specific to the Unidrive SP Family

Accessing 32 bit Registers via Modbus RTU

The Unidrive SP incorporates both 16 bit and 32 bit registers within its parameters.

Example: Menu #20.10 = 16bit, Range = -32,768 to 32,767

Menu #20.21 = 32bit, Range = -2^{31} to 2^{31} -1

Most Modbus drivers available today allow for 32 bit data transfer via 2-16bit registers. Although this type of classical setup will function properly with the Unidrive SP 16-bit parameters, this is not the case with Unidrive SP 32-bit parameters. Each 32-bit Unidrive SP register is a single register with a 32 bit piece of data residing in it. Because of this, a standard 2-word Modbus Read or Write will not access the 32 bits that the user is looking for, rather it will poll the first 16 bits of one 32-bit register and then the first 16-bits of a different 32-bit register. Users familiar with standard Modbus RTU protocol and register structure will undoubtedly be confused by this structure. The Unidrive SP has a backdoor to reading and writing 32 bit parameters via Modbus which makes use of a 6-digit addressing scheme.

Implications:

Applications utilizing SM-EZ motion are now easier to communicate with via standard 2 word (long format) Modbus. (Long words will not have to be dissected before sending them to and from the drive)

PLC registers residing in the SM-Apps and SM-Apps Light modules are now accessible with most HMI products via Modbus drivers.

Limitations:

Accessing 32 bit registers requires 6 digit Modbus addressing. Therefore the user will need a Modbus driver capable of accessing parameters 4xxxxxx.

16 bit word example:

5 digit addressing to read #20.10	=	42010
6 digit addressing to read #20.10	=	402010

Many Modbus drivers contain this capability but some do not. Check your HMI or PLC interface prior to purchasing to verify this functionality.
When accessing 32 bit registers the number of words/registers read/written needs to be an even number. This means that for a single 32 bit register the size parameter needs to be set to "2". That is 2-16 bit registers.

Example:

Read/Write a single 32 bit register (#20.25)

Take the address of the 32 bit parameter and multiply it by 100.

Example: 20.25 * 100 = 2025

Add this number to 16384 (0x4000)

Example: 2025 + 16384 = 18409

Place a “4” in front of this number indicating that it is a holding register.

Example: 18409 => 418409

This register will access the MSW of the 32 bit register 20.25. The LSW may be accessed by adding 1 to this address.

Example:	MSW = 418409
	LSW = 418410

Read/Write multiple 32 bit registers (#20.21 - #20.25)

Take the starting address of the block and multiply it by 100.

Example: 20.21 * 100 = 2021

Add this number to 16384 (0x4000)

Example: 2021 + 16384 = 18405

Place a “4” in front of this number indicating that it is a group of holding registers.

Example 18405 => 418405

This register will access the MSW of the first word of the block (#20.21). The LSW may be accessed by adding 1 to this address. The MSW of the next 32 bit register may be accessed by adding 2 to this address and its LSW by adding 3.

Example:	MSW (#20.21) =	418405
	LSW (#20.21) =	418406
	MSW (#20.22) =	418407
	LSW (#20.22)=	418408
	MSW (#20.23)=	418409
	LSW(#20.23)=	418410
	MSW(#20.24)=	418411
	LSW(#20.24)=	418412
	MSW(#20.25)=	418413
	LSW(#20.25)=	418414

NOTE: Unlike standard Unidrive SP register reads and writes, the 32 bit “calculated registers” as shown above are dynamic and depend on structure of the Modbus request. This means that in the example above, parameter 418414 is not a static Modbus address in the drive, rather the Unidrive SP looks at the Modbus command, grabs the correct data from within the specified registers (in this case #20.25) and returns a response which emulates this data at 418414.

On multiple 32-bit register reads and writes, the starting address of the read or write is going to dictate subsequent addresses. This means that the same parameter could have a different higher level address depending on the starting address of the block read/write. Take care to calculate the address of the Modbus parameter desired prior to programming the Master to reduce potential errors.

From Above:

Single word write #20.25

#20.25 is mapped to	418409 – 418410

Multiple word write #20.21 - #20.25

#20.25 is mapped to	418413 - 418414

Example Mappings:

Parameter	20.21	20.22	20.23	20.24	20.25					32-Bit Parameters
Starting Address	418405	418406	418407	418408	418409	418410	418411	418412	418413	418414

Parameter	20.23	20.24	20.25	20.26	20.27					32-Bit Parameters
Starting Address	418407	418408	418409	418410	418411	418412	418413	418414	418415	418416

Parameter	70.01	70.02	70.03	70.04	70.05					32-Bit Parameters
Starting Address	423385	423386	423387	423388	423389	423390	423391	423392	423393	423394

Questions ?? Ask the Author:

Author: **Tom Schrauth** **e-mail :** Tom.Schrauth@emersonct.com
952-995-8168