Micro Motion® Modbus Interface Tool
Slots and Slot Address Sequences

SA 1 Overview

Slots provide a way to read non-consecutive registers using one read command. When a slot address sequence (a set of consecutive slots) has been configured, the transmitter automatically replicates real-time data from non-consecutive addresses into the consecutive addresses of the slots. Without slots, multiple read commands must be issued to read the non-consecutive addresses; with slots, only one read command is required.

There are 64 slots: 32 slots used for sets of addresses (Slot Type 1) and 32 slots used for sets of process variables (Slot Type 2). Each slot has a pre-assigned configuration register (integer) and a corresponding value register pair. The configuration registers are read-write; the value register pairs may be read-write or read-only.

For Slot Type 1:

- The configuration register stores a Modbus address. The address specifies the first register of any two consecutive registers (Registers X and X+1) in the Modbus memory map. These registers may contain boolean, integer, floating-point, ASCII, or binary data.
- The first register of the value register pair stores the real-time value of Register X, and the second register stores the real-time value of Register X+1. Depending on the particular registers specified, the second register may or may not contain useful data.

For Slot Type 2:

- The configuration register stores a process variable code.
- The value register pair stores the real-time value of the assigned process variable. Depending on the data type of the process variable, the second register may or may not contain useful data.

See Table SA-1 for a list of the 64 slots and their associated configuration registers and value register pairs.

Section SA-2 provides instructions for configuring slot address sequences.

Section SA-3 provides instructions for reading slot address sequences.

Section SA-4 provides instructions for using long integers in slot address sequences.

Section SA-5 provides instructions for using binary (double-precision) values in slot address sequences.

Section SA-6 provides detailed examples of slot address sequence setup and use.

Table SA-1 Slot address registers

Slot	Slot Type 1		Slot Type 2	
	Address of configuration register [1]	Address of value register pair	Address of configuration register [2]	Address of value register pair
0	655	687 688	751	783 784
1	656	689 690	752	785 786
2	657	691 692	753	787 788
3	658	693 694	754	789 790
4	659	695 696	755	791 792
5	660	697 698	756	793 794
6	661	699 700	757	795 796
7	662	701 702	758	797 798
8	663	703 704	759	799 800
9	664	705 706	760	801 802
10	665	707 708	761	803 804
11	666	709 710	762	805 806
12	667	711 712	763	807 808
13	668	713 714	764	809 810
14	669	715 716	765	811 812
15	670	717 718	766	813 814
16	671	719 720	767	815 816
17	672	721 722	768	817 818
18	673	723 724	769	819 820
19	674	725 726	770	821 822
20	675	727 728	771	823 824
21	676	729 730	772	825 826
22	677	731 732	773	827 828
23	678	733 734	774	829 830
24	679	735 736	775	831 832
25	680	737 738	776	833 834
26	681	739 740	777	835 836
27	682	741 742	778	837 838
28	683	743 744	779	839 840
29	684	745 746	780	841 842
30	685	747 748	781	843 844
31	686	749 750	782	845 846

(1) Contains a Modbus address.
(2) Contains a process variable code.

SA 2 Configuring slot address sequences

To configure a slot address sequence of Slot Type 1:

1. Determine the values that the read command will return, and the order in which they will be returned.

   For example, you might want to read volume flow rate and the volume flow unit.

2. Identify the addresses containing these values.

   Volume flow rate is stored in register pair 253–254, and the volume flow unit is stored in register 42.

3. Determine the slots to be used in the slot address sequence. Each value that the read command will return requires a separate slot.

   To read volume flow rate and the volume flow unit, you need two consecutive slots, e.g., Slot 0 and Slot 1.

4. Write the address of the first value, minus 1, to the configuration register of the first slot in the sequence. See Table SA-1.

   Write the value 252 to register 655. The contents of register 253 will be replicated in register 687, and the contents of register 254 will be replicated automatically in register 688.

5. Write the address of the second value, minus 1, to the configuration register of the second slot in the sequence. See Table SA-1.

   Write the value 41 to register 656. The contents of register 42 will be replicated in register 689. Additionally, the contents of register 43 will be replicated automatically in register 690, and register 690 will be read automatically. The host program must ignore the value from register 690.

6. Continue specifying register pairs in consecutive slots until all required values been configured.

To configure a slot address sequence of Slot Type 2:

1. Determine the process variables that the read command will return, and the order in which they will be returned. See the Integer Codes list in the Modbus Interface Tool for information on process variables that can be assigned to slot addresses.

   For example, you might want to read volume flow rate and volume total.

2. Identify the integer codes for the process variables.

   The integer code for volume flow rate is 5; the integer code for volume total is 6.

3. Determine the slots to be used in the slot address sequence. Each value that the read command will return requires a separate slot.

   To read volume flow rate and volume total, you need two consecutive slots, e.g., Slot 2 and Slot 3.

4. Write the integer code of the first process variable to the configuration register of the first slot in the sequence. See Table SA-1.

   Write the value 5 to register 753. A floating-point value representing volume flow rate will be stored in registers 787–788.

5. Write the address of the second value to the configuration register of the second slot in the sequence. See Table SA-1.

   Write the value 6 to register 754. A floating-point value representing volume total will be stored in registers 789–790.

6. Continue specifying integer codes in consecutive slots until all required values been configured.

SA 3 Reading slot address sequences

Refer to Table SA-1 and issue a read command that identifies the first value register in the sequence and the number of registers to be read.

Note: When reading a slot address sequence, be sure to specify the value register pairs, not the configuration registers, and be sure to process or ignore the value in the second register of the value register pair, as appropriate.

SA 4 Long integers in slot address sequences

To include a long integer value in a slot address sequence, configure all registers that contain the long integer value. Be sure to add them to the slot address sequence in the order you want to read them.

When reading the slot address sequence, the host program must be able to interpret and concatenate the returned values as required.

SA 5 Binary (double-precision) values in slot address sequences

Several process variable codes are used in pairs to return double-precision values. These pairs include:

- 34 and 35 – used to return mass total
- 36 and 37 – used to return volume total
- 38 and 39 – used to return temperature-corrected volume total
- 40 and 41 – used to return standard volume total
- 42 and 43 – used to return net mass total
- 44 and 45 – used to return net volume total

To include a double-precision value in a slot address sequence, two slots are required. Configure the first slot for the first process variable code, and configure the second slot for the second process variable code. For example, to return mass total as a double-precision value, write 34 to the configuration register of the first slot, and write 35 to the configuration register of the second slot.

When you read the first value register pair, the first register will contain Word 0 and the second register will contain Word 1. When you read the second value register pair, the first register will contain Word 2 and the second register will contain Word 3. The returned values are in a special, non-standard, 8-byte floating point format.

Note: For MVD Direct Connect or Series 1000, Series 2000, or Series 3000 transmitters, byte order in floating-point registers is configurable. Binary totals are not affected by byte order.

To convert the binary code to the total value, follow the steps below.

1. Calculate M:

$$M = [((\text{Word1} \times 65536) + \text{Word2}) \times 65536] + \text{Word3}$$

2. Set P = Word 0.
3. Calculate as follows:

$$\text{TotalFlow} = M \times 2^{(P-47)}$$

4. If you are reading mass total (codes 34 and 35), this value represents mass total in grams. If you are reading volume total (codes 36 and 37), this value represents volume total in cubic centimeters.

Note: Both M and P are twos complement notation. If you are working with negative values (i.e., reverse flow), adjust this method as required.

Example

Reading binary totals

Reading the slot addresses returns the following values:

- Word 0 = 001B (decimal value: 27)
- Word 1 = 75BC (decimal value: 30140)
- Word 2 = D152 (decimal value: 53586)
- Word 3 = 0000 (decimal value: 0)

Calculate M:

$$M = [((30140 \times 65536) + 53586) \times 65536] + 0$$

$$M = 1.2945383 \times 10^{14}$$

P = 27.

Calculate total flow:

$$\text{TotalFlow} = 1.2945383 \times 10^{14} \times 2^{(27-47)}$$

$$\text{TotalFlow} = 1.2945383 \times 10^{14} \times 2^{(-20)}$$

$$\text{TotalFlow} = 1.2945383 \times 10^{14} \times 9.5367432 \times 10^{-7}$$

$$\text{TotalFlow} = 123,456,789.125$$

SA 6 Slot address examples

Example 1

Slot address sequence – Slot Type 1 (addresses)

Several times per day, the operator needs to read the following:

- Floating-point volume flow rate
- Volume flow unit
- Floating-point process density
- Density unit

The volume flow unit is gal/min (integer code 16). The density unit is g/cm^3 (integer code 91).

Configure a slot address sequence so the operator can read the required values by issuing a single read command. Slots 0 through 3 will be used.

1. Write the integer value 252 to register 655. The value represents register pair 253–254, which stores the volume flow rate.
2. Write the integer value 41 to register 656. The value represents register pair 42–43. Register 42 stores the volume flow unit. You must ignore the value stored in register 43.
3. Write the integer value 248 to register 657. The value represents register pair 249–250, which stores the process density.
4. Write the integer value 39 to register 658. The value represents register pair 40–41. Register 40 stores the density unit. You must ignore the value stored in register 41.

If the operator reads register pairs 687–688 through 693–694, the transmitter returns a series of values similar to the following:

23.038	16 6	1.091	91 33
(Volume flow rate)	(Volume flow unit plus next register)	(Process density)	(Density unit plus next register)

Example 2

Slot address sequence – Slot Type 2 (process variables)

Several times per day, the operator needs to read the following:

- Temperature in degrees Fahrenheit
- Volume flow rate in gallons/minute
- Drive gain in milliamps
- Left pickoff voltage in millivolts
- Right pickoff voltage in millivolts

Configure a slot address sequence so the operator can read the required values by issuing a single read command. Slots 4 through 8 will be used.

1. Write the integer value 1 to register 755. The value represents the integer code for temperature.
2. Write the integer value 5 to register 756. The value represents the integer code for volume flow.
3. Write the integer value 47 to register 757. The value represents the integer code for drive gain.
4. Write the integer 49 to register 758. The value represents the integer codes for left pickoff voltage.
5. Write the integer code 50 to register 759. The value represents the integer code for the right pickoff voltage.

If the operator reads register pairs 791–792 to 799–800, the transmitter returns a series of values similar to the following:

60.09	23.038	7.009	17087.05	17087.02
(Temperature)	(Volume flow rate)	(Drive gain)	(Left pickoff)	(Right pickoff)

Each of these values is returned using the unit that has been configured for the process variable. This example assumes that the required units (as listed above) have already been configured.

Micro Motion®

EMERSON™
Process Management