Introduction

This document describes the recommended method to configure and connect Western Reserve Controls (WRC) Modbus to DeviceNet gateway for use with the GPD 506/P5.

There currently are three types of gateways available:

- 1782-JDM-1 for RS232,
- 1782-JDM-2 for RS485
- 1782-JDM-3 for isolated RS232.

Although this document focuses on the use of this gateway with the GPD 506/P5 drive, it can be used with any Yaskawa drive that communicates via Modbus.

Configuration & settings.

This procedure assumes that the reader is familiar with the GPD 506/P5 drive, DeviceNet, Modbus communication protocol and PLC programming.

The configuration of the gateway is straightforward and must be done through configuration software such as DeviceNet Manager, RSNetworx for DeviceNet, or equivalent. Figure 1 shows the configuration screen for RSNetworx with the recommended parameter settings.
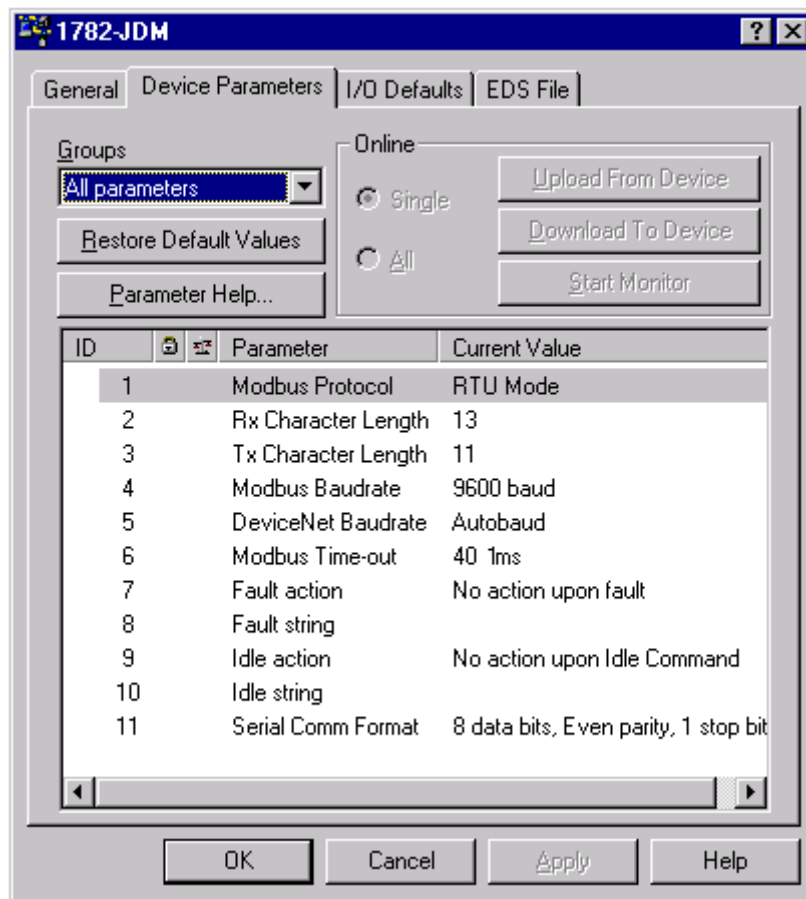

Figure 1. Basic Gateway Settings.

Modbus Protocol:
The GPD 506/P5 supports Modbus RTU protocol only through its serial port. If ASCII protocol is used, the drive will not respond.

Modbus baud rate, Modbus time-out, serial communication format:
These settings are used to configure the serial port, which is fixed on the drive to 8 data bits and 1 stop bit.

Rx & Tx character length:
These parameters refer to the amount of information that will be passed between the GPD 506/P5 and the WRC gateway. This is explained in more detail in the **Mapping the Gateway to the Scanner** paragraph.

The drive's parameters must be properly set for it to communicate. Table 1 indicates what parameters affect the communication and where appropriate, the recommended settings. The settings are based on the gateway configuration as displayed in Figure 1.

Table 1. 506/P5Communication Parameters.

Parameter	Description	Comments	Setting
n002	Operation mode selection.	This setting is process dependent.	User
n101	Modbus timeout detection.	The setting is process dependent.	User
n102	Stop method on Modbus communication error.	The setting is process dependent.	User
n103	Modbus frequency reference unit.	This refers to the resolution of the reference number sent by the master device.	User
n104	Modbus slave address.	Addresses must be between 0 and 31. Each drive must have its own unique address.	User
n105	Modbus BPS selection.	This should be set to the max.	2: 9600
n106	Modbus parity selection.	Based on limitations of the gateway, even parity is recommended.	1: Even parity

Gateway Message Format.
The Yaskawa family of drives can handle two of the Modbus commands; the multiple register read (03h) and the multiple register write (10h). These commands can be used to send the start/stop and frequency reference as well as to read the drive status and operating frequency.

Important: Information will not be formatted in a typical ODVA inverter drive assembly.

1. The output data format (information from the scanner to the drive) for a write command is:

Byte	Data
0	Record number.
1	0Bh. Length. Number of bytes of Modbus message.
2	Modbus address (drive's address, parameter n104).
3	10h. Multiple register write.
4	01h. Starting register lower address.
5	00h. Starting register upper address.
6	02h. Quantity of registers to write lower segment.
7	00h. Quantity of registers to write upper segment.
8	04h. Number of bytes that will be sent (2 per register)
9	XXh. First register (0001h) data lower segment.
10	XXh. First register (0001h) data upper segment.
11	YYh. Second register (0002h) data lower segment.
12	YYh. Second register (0002h) data upper segment.

The first register (0001h) controls the drive's run/stop command and the second register (0002h) controls the frequency.

2. The output data format for the read command is:

Byte	Data
0	Record number.
1	06h. Length. Number of bytes of Modbus message.
2	Modbus address (drive's address, parameter n104).
3	03h. Multiple register read.
4	20h. Starting register lower address.
5	00h. Starting register upper address.
6	05h. Quantity of registers to read lower segment.
7	00h. Quantity of registers to read upper segment.

In this case the data being read are registers 20h (drive status), 21h (drive fault contents), 22h (communication data link progress), 23h (frequency reference) and 24h (output frequency).

The record number is used by the gateway to avoid sending the same message more than once; this means that the number must change for the next message to be sent.

After the message has been sent, converted to a true Modbus format and sent on to the drive, the drive will respond. The response is stripped of the CRC and added to it are the record number (same as the message that it corresponds to), communication status (bit 7 is set to 1 while the gateway awaits for a response, and to 0 when it receives it) and the message length in bytes.

3. The input data format (information from the drive to the scanner) for a write command is:

Byte	Data
0	Record number.
1	Status.
2	Length. Number of bytes of Modbus message.
3	Modbus address (drive's address, parameter n104).
4	10h. Multiple register write.
5	00h. Starting register lower address.
6	01h. Starting register upper address.
7	00h. Quantity of registers to write lower segment.
8	02h. Quantity of registers to write upper segment.

The record number will match the one of the sent messages and the status will be 0 unless a communication error has occurred. This response verifies that the registers were written to. The PLC can monitor the status register and will ignore the message unless there is an error.

4. The input data format for a read command is:

Byte	Data
0	Record number.
1	Status.
2	Length. Number of bytes of Modbus message.
3	Modbus address (drive's address, parameter n104).
4	03h. Multiple register read.
5	0Ah. Number of data bytes.
6	UUh. First register data lower segment.
7	UUh. First register data upper segment.
8	VVh. Second register data lower segment.
9	VVh. Second register data upper segment.
10	WWh. Third register data lower segment.
11	WWh. Third register data upper segment.
12	XXh. Fourth register data lower segment.
13	XXh. Fourth register data upper segment.
14	YYh. Fifth register data lower segment.
15	YYh. Fifth register data upper segment.

The read message response will always be longer because of the data coming across. The Modbus response also is stripped of the CRC and the record number. Status and message length are added.

For a comprehensive explanation of the Modbus protocol and the data format, refer to the **GPD 506/P5 Modbus RTU Technical Manual**.

Mapping the Gateway to the Scanner.

Correct mapping of the gateway will eliminate a lot of confusion with regards to the appearance of the information. In the previous section, the data tables indicate the format in which the information is set on the gateway. Since typical mapping is done by placing the individual bytes in word registers, the data may look confusing.

For example, if a write message is sent to a drive with the following information:
Modbus address: 03
Command: 16 (10h)
Run command: a value of 1 on register 0001.
Frequency: a value of 300 (30.0 Hz) on register 0002.

The N10 register table (if using a 1771-SDN) would contain the following information and would be sent to the WRC gateway via DeviceNet.

Register	Data	Description
N10:1	2816	Combination of record number and message length.
N10:2	4099	Combination of address and Modbus command.
N10:3	1	Combination of starting address upper and lower.
N10:4	2	Combination of records to read upper and lower.
N10:5	260	Combination of number of bytes and first register lower.
N10:6	256	Combination or first register upper and second lower.
N10:7	44	Second register upper.

A response message from the drive, by means of a multiple register read, would also appear incoherent.

Example of Segmented Mapping.

There is a another way to map the WRC Gateway so that the information sent and received is easier to read. This procedure requires mapping different segments of the gateway into different areas of the scanner. There can be up to 4 segments for each device on the scan list.

Drive's Modbus address: 03
Drive will be running: register 0001h must have a value of 1.
Drive will be running at 30.0 Hz: register 0002h must have a value of 300.
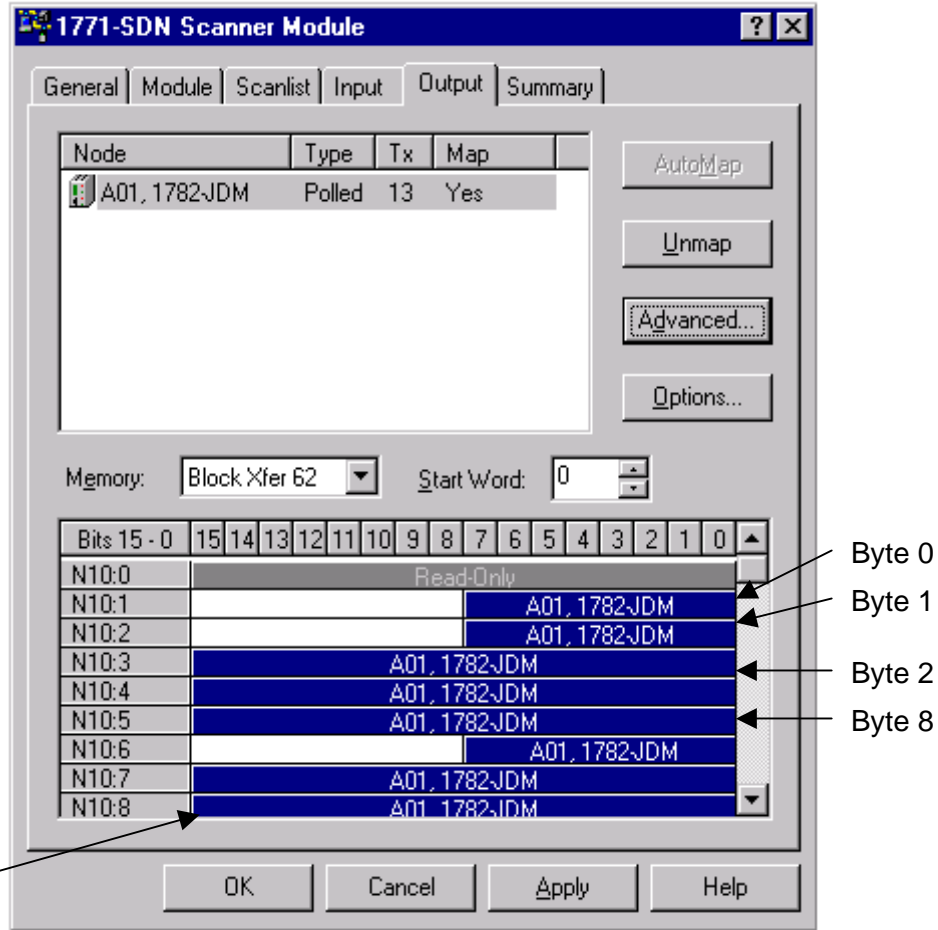
Figure 2. Segmented output mapping.

The N10 register table would contain the following information, which would be sent to the WRC gateway via DeviceNet.

Register	Data	Description
N10:1	1	Record number.
N10:2	11	Length.
N10:3	4099	Combination of Modbus address and write command.
N10:4	1	Starting register.
N10:5	2	Number of registers to write.
N10:6	4	Number of data bytes sent.
N10:7	1	Data of first register.
N10:8	300	Data of second register.

The information, although the same as in the previous example, is easier to understand, the only register that presents an incoherent number is N10:3. This is because the decimal number shown is the combination of the Modbus address (lower) and the function code (upper).
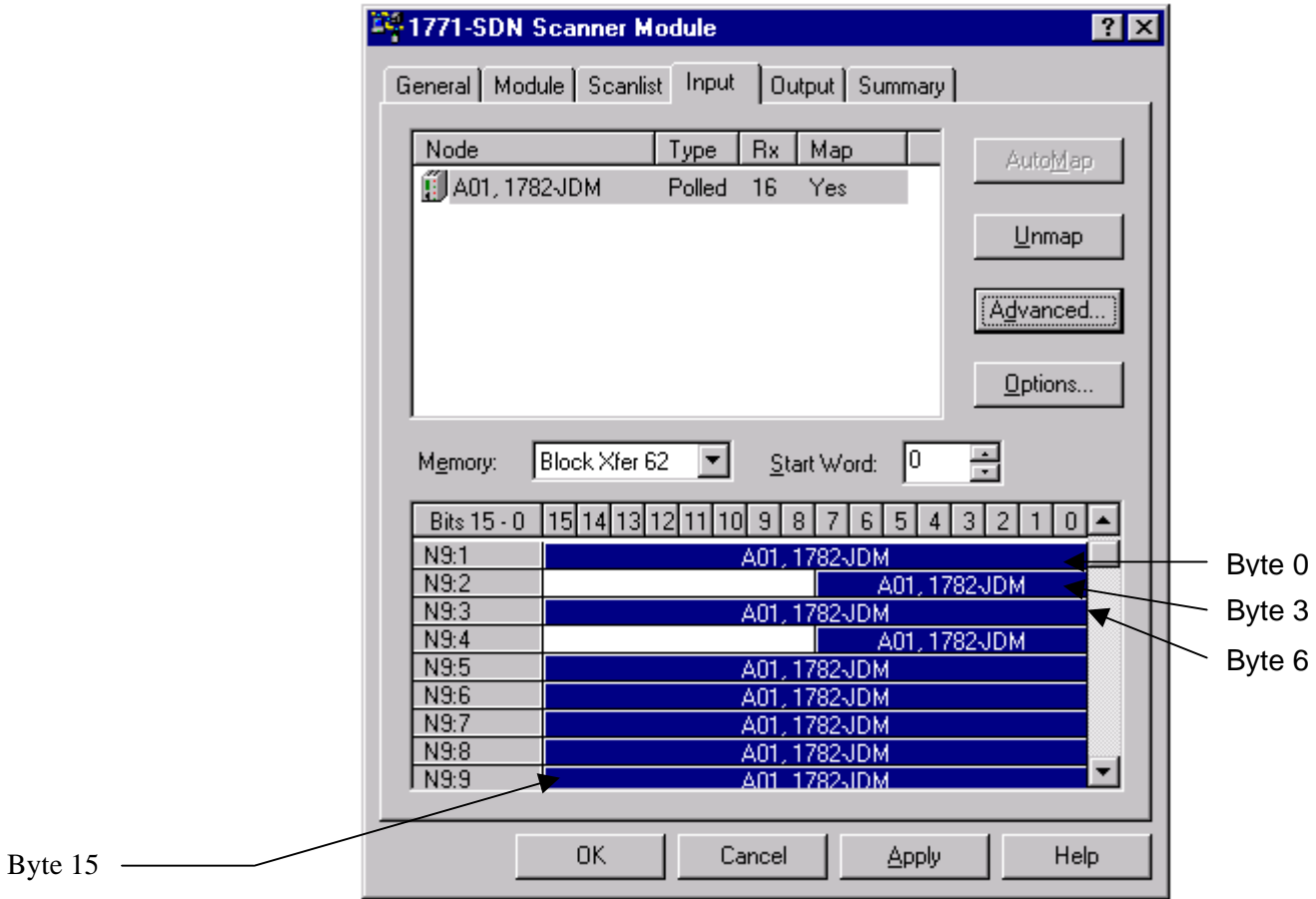
Figure 3. Segmented input mapping.

A response message from the drive, by means of a multiple register read, would also be easier to understand. Again, because of the way the information is formatted on the scanner, figure 3.

Based on the example settings the drive would return the following data:

Drive register 0020h	04h	Drive status
Drive register 0021h	00h	Fault indicators
Drive register 0022h	00h	Communication status
Drive register 0023h	12Ch	Drive's frequency reference
Drive register 0024h	12Ch	Drive's output frequency

The way the data would be transferred into the N9 registers would look like:

Register	Data	Description
N9:1	1	Combination of record number and status.
N9:2	13	Length.
N9:3	771	Combination of Modbus address and read command.
N9:4	10	Number data bytes received.
N9:5	5	Drive status (0020h).
N9:6	0	Fault indicators (0021h).
N9:7	0	Communication status (0022h).
N9:8	300	Frequency reference (0023h).
N9:9	300	Output frequency (0024h).

As with the output registers, the only register with an incoherent number is N9:3 and the reason is the same. This register is the combination of the Modbus address (lower) and the Modbus read command (upper).

The first register (N9:1) will indicate the record number (lower) and if there is communication problem, the communication error (upper). If there are no communication errors, the upper segment of the 16-bit register will always be 0.

For a procedure on how to segment information on a scanner module, consult the help files or the user's manual of the configuration software.

Ladder logic example.
The following example was written for a SLC using a 1747-SDN.

```
      Enables the scanner module.
                                                              O:1.0
  0  ├──────────────────────────────────────────────────────( )──
                                                                0
      Copies the modbus message located on N7:1 through N7:9 and
      the record number (N7:0).

         B3                                    ┌─COP─────────────┐
  1  ├───┤ / ├───────────────────────────────────┤ Copy File      │
          0                                     │ Source    #N7:0 │
                                                │ Dest    #M0:1.0 │
                                                │ Length       10 │
                                                └─────────────────┘

                                                      B3
                                                    ─(L)──
                                                      0
```
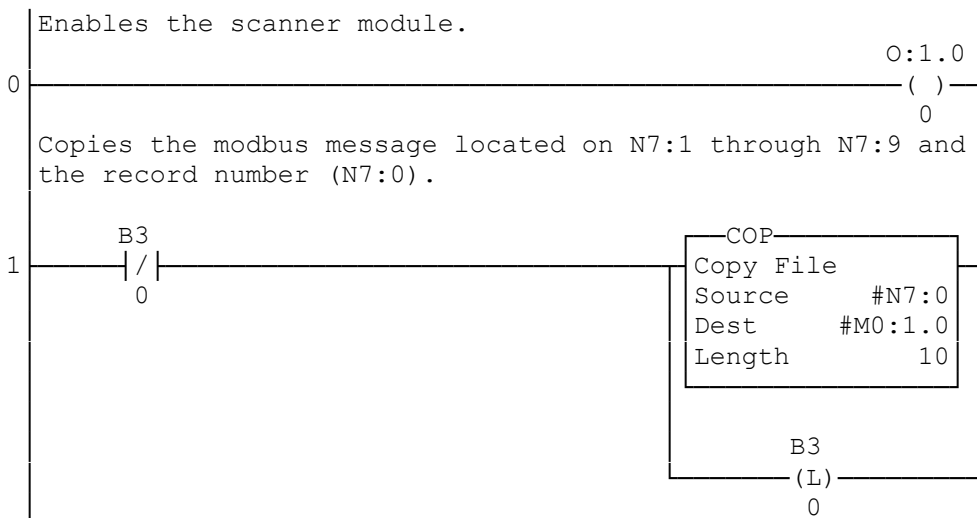
This rung is looking for the multiple register read
(command 03h). This command is in register 1 of the
response message and is combined with the address of the
drive (03h), the result of this combination is 771. If the
data in the M file corresponds to a response to the read
command then place the data on the N7:20 registers for
further use.

```
       M1:1.0            ┌─EQU──────────────┐  ┌─COP──────────────┐
 2 ─────┤ / ├────────────┤Equal             ├──┤Copy File         ├──
         15              │Source A   M1:1.1 │  │Source     #M1:1.0│
                         │                 *│  │Dest       #N7:20 │
                         │Source B      771 │  │Length         15 │
                         └──────────────────┘  └──────────────────┘

                                                          B3
                                                         (U)──
                                                          0
```
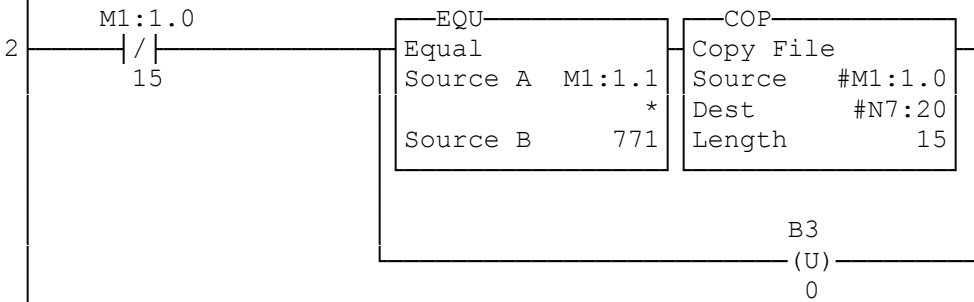
The rest of the program handles the record number and the
flip flop of the data read and write.

This rung increases the offset register (N7:10) by 10. This
will allow the transfer of the read command into the N7:1
area so that it can be sent to the drive.

```
        B3                                    ┌─ADD──────────────┐
 3 ─────┤ / ├───────────────────────────────┤Add               ├──
         0                                   │Source A       10 │
                                             │Source B    N7:10 │
                                             │                0 │
                                             │Dest        N7:10 │
                                             │                0 │
                                             └──────────────────┘
```
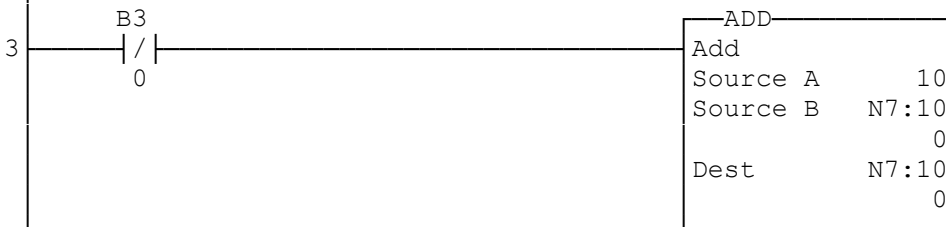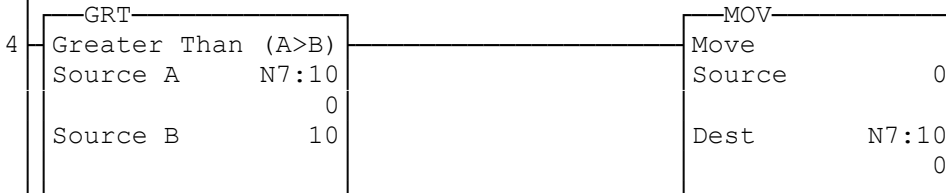
If the offset register is greater than 10, the read
command, the reset it to 0, the write command.

```
    ┌─GRT────────────────┐                   ┌─MOV──────────────┐
 4 ─┤Greater Than (A>B)  ├───────────────────┤Move              ├──
    │Source A     N7:10  │                   │Source          0 │
    │                0   │                   │                  │
    │Source B        10  │                   │Dest        N7:10 │
    └────────────────────┘                   │                0 │
                                             └──────────────────┘
```

Copy the command from the N10 file to the N7 file. N7:10 is
used as the offset for the area to be copied.

```
        B3                                    ┌─COP──────────────┐
 5 ─────┤ / ├───────────────────────────────┤Copy File         ├──
         0                                   │Source #N10:[N7:10]│
                                             │Dest         #N7:1 │
                                             │Length           9 │
                                             └──────────────────┘
```

```
      The N7:0 register is increased by 1, this is so that the
      next message will have a different record number.
      Each consecutive message must have a different record
      number otherwise it will not be sent.

          B3                                          ┌───ADD─────────────┐
  6 ──────┤ / ├───────────────────────────────────────┤Add                │───
           0                                          │Source A         1 │
                                                      │Source B      N7:0 │
                                                      │               460 │
                                                      │Dest          N7:0 │
                                                      │               460 │
                                                      └───────────────────┘

      This is just acting as a limiter for the record number.
      ┌───EQU──────────────┐                          ┌───MOV─────────────┐
  7 ──┤Equal               ├──────────────────────────┤Move               │───
      │Source A      N7:0  │                          │Source           0 │
      │               460  │                          │                   │
      │Source B      1000  │                          │Dest          N7:0 │
      │                    │                          │               460 │
      └────────────────────┘                          └───────────────────┘

  8 ───────────────────────────────────────────────────────────[END]───────
```

The N10 data table has the following information:

N10:0	11	4099	1	2	4	1	300	0	0	0
N10:10	6	771	32	5	0	0	0	0	0	0

N10:0 through N10:6 have the write command data to be written, N10:10 through N10:13 have the read command.

The N7 data table has the following information:

N7:0	200	11	4099	1	2	4	1	300	0	0
N7:10	0	0	0	0	0	0	0	0	0	0
N7:20	199	13	771	10	4	0	0	300	300	0

The information read from the drive is placed on registers N7:20 through N7:29, of which N7:24 through N7:28 are actual drive data.

Scanner Configuration Notes.

The number of bytes used by the polled message (devicenet) is not the same as the number of bytes used for a modbus message. The JDM configuration screen, Figure 1, is used to set the number of bytes for the modbus message, the number of bytes sent to and received from the drive.

The polled sizes are larger because of the extra information that is sent between the scanner and the WRC gateway;
- 16 bytes for the Rx. This adds the space for the record number, the communication status and the length
- 13 bytes for the Tx. This adds the space for the record number and the length.

For more information regarding the WRC gateway and it's configuration refer to the 1782-JDM DeviceNet-ModBus Gateway User's Manual.

Please consult with WRC with regards to current gateway firmware and documentation.

Wiring.

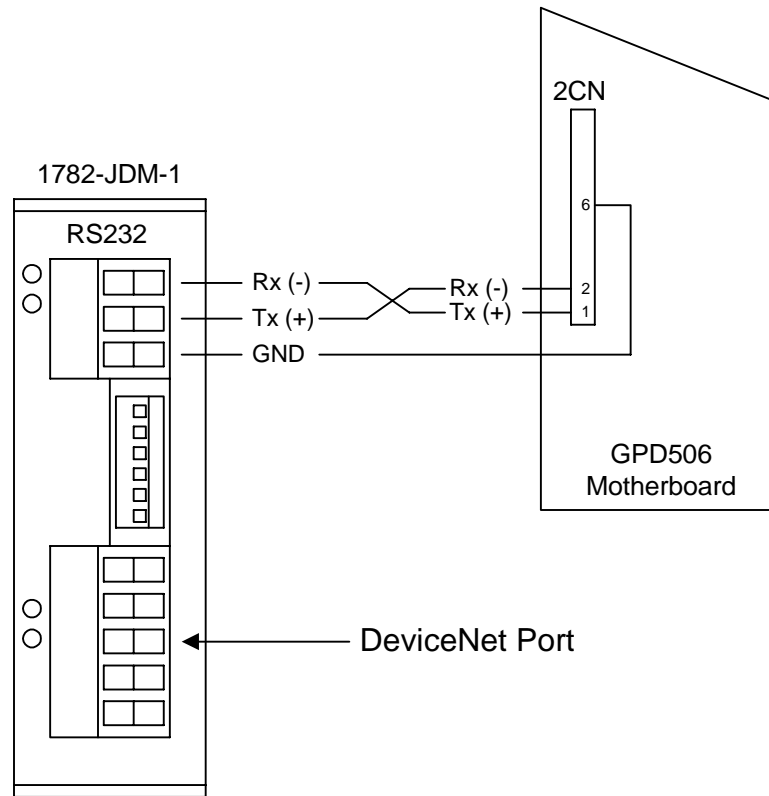Figure 4 shows the proper wiring for RS232 between the WRC gateway and the GPD 506/P5.

Figure 4. RS-232 wiring scheme.

The GPD 506/P5 connection side requires a special connector, the Yaskawa part number for this connection kit is **CM088**.

Please refer to the **GPD 506/P5 Modbus RTU Technical Manual for the RS485 wiring scheme and required parts.**

Reference material.
The following documents can be used as a reference on how to use the WRC gateway for a different drive or with a Modbus 485 network:

GPD 506/P5	GPD 506/P5 / Modbus RTU Technical Manual
1782-JDM-X	User Manual for the Gateway.
SLC Scanner Module	1747-SDN Installation Instructions, May 1997 or later
PLC Scanner Module	1771-SDN Installation Instructions, May 1997 or later