# MODBUS Network Communication

The IP3416 and IP99 use the MODBUS<sup>TM</sup> protocol, originally standardized by Modicon and is used widely in the PLC industry. For instance, many graphic interface packages available in the market support the MODBUS<sup>TM</sup> protocol. The protocol is nonproprietary and detailed specifications are available from the AEG/Modicon Company. The IP3416 supports both the ASCII and RTU modes of the MODBUS<sup>TM</sup> protocol and the IP99 supports the RTU protocol.

The following MODBUS<sup>TM</sup> functions are supported:

      Function 01 :  Read output status
      Function 02 :  Read input status
      Function 03 :  Read output registers
      Function 04 :  Read input registers
      Function 05 :  Force single coil
      Function 06 :  Preset single register
      Function 15 :  Force multiple coils
      Function 16 :  Force single register

The ASCII mode has the following packet format, `:iiffaaaaddddccCRLF`, where,

| | |
|---|---|
| `ii` | slave identification (1...F7 hex or 1...247 decimal ) |
| `ff` | function code |
| `aaaa` | start address of data |
| `dddd` | # of points to retrieve or data to set register to |
| `cc` | packet checksum |
| CR | carriage return |
| LF | line feed |

The colon indicates the start of the packet and a carriage return and line feed pair indicates the end of the packet. All characters are ASCII characters of hexadecimal numbers.

      i.e.    `:010100000001FD`CRLF
             regular text characters

The RTU mode has the following format, `ifaaddcc`, where,

| | |
|---|---|
| i | slave identification (1...F7 hex). |
| f | function code. |
| aa | start address of data minus 1. |
| dd | # of points to retrieve or data to set register to. |
| cc | packet checksum. |

The packet data is in pure binary form.

      i.e.    `010100000001FDCA`
             hexadecimal numerals

## Read Output Status Function

This function allows the user to obtain the ON/OFF status of all the outputs, Y0, ..., Y15, of the IP3416.  The address for the discrete outputs is,

| Registers | Address |
|-----------|---------|
| Y0, ..., Y15 | 00001 |

The function format is as follows,

Master Query:
```
AA 01 0000 0001 CC
```

```
AA    slave identification
CC    checksum
```

Slave Response:
```
AA 01 01 JJJJ CC
```

```
AA    slave identification
JJJJ  status of Y0, ..., Y15
```

The returned status is in the following format,

```
 BIT 15                                            BIT 0
    Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0 Y15 Y14 Y13 Y12 Y11 Y10 Y9 Y8
```

## Read Input Status Function

This function allows the user to obtain the ON/OFF status of all the discrete inputs of the IP3416.  The addresses for the discrete inputs, X0, ..., X47, are,

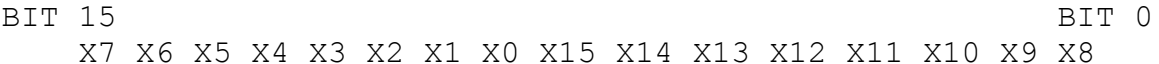| Registers | Addresses |
|-----------|-----------|
| X0, ..., X15 | 10001 |
| X16, ..., X31 | 10016 |
| X32, ..., X47 | 10031 |

The function format is as follows,

Master Query:
```
AA 02 aaaa nnnn CC
```

```
AA    slave identification
aaaa  starting address minus 10001 in hexadecimal format
nnnn  number of data points to retrieve
CC    checksum
```

Slave Response:
```
AA 02 BB JJJJ KKKK LLLL MMMM CC
```

```
AA    slave identification
BB    number of data bytes sent
JJJJ  first 16 inputs
  •
  •
MMMM        last 16 inputs
CC    checksum
```

The returned status is in the following format,

```
BIT 15                                                          BIT 0
    X7 X6 X5 X4 X3 X2 X1 X0 X15 X14 X13 X12 X11 X10 X9 X8
```

## Read Output/Input Status Function

This function allows the user to obtain the contents of the IP3416 internal registers.  The user may obtain a maximum of 8 consecutive registers.  The addresses for the internal registers are,

| Registers | Addresses |
|-----------|-----------|
| T0, ..., T31 | 40001...40032 |
| C0, ..., C31 | 40033...40064 |
| D0, ..., D31 | 40065...40096 |
| R0, ..., R255 | 40097...40112 (in blocks of 16) |
| A24, ..., A31 | 40121...40128 |

The function format is as follows,

Master Query:        `AA 03 aaaa nnnn CC`

```
AA      slave identification
aaaa    starting address minus 40001 in hexadecimal format
nnnn    number of data points to retrieve
CC      checksum
```
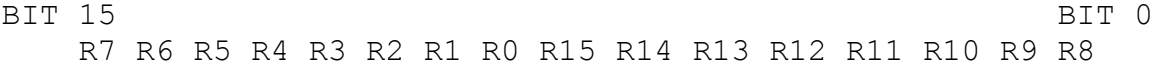
Slave Response:      `AA 03 BB JJJJ KKKK LLLL MMMM CC`

```
AA      slave identification
BB      number of data bytes sent
JJJJ    first register value
  •
  •
MMMM           last register value
CC      checksum
```

The IP3416 also supports function 4 and the only difference between function 3 and 4 are the addresses.  The addresses for function 4 are as follows,

| Registers | Addresses |
|-----------|-----------|
| T0, ..., T31 | 30001...30032 |
| C0, ..., C31 | 30033...30064 |
| D0, ..., D31 | 30065...30096 |
| R0, ..., R255 | 30097...30112 (in blocks of 16) |
| A24, ..., A31 | 30121...30128 |

The returned status for the internal relays is in the following format,

```
BIT 15                                                          BIT 0
    R7 R6 R5 R4 R3 R2 R1 R0 R15 R14 R13 R12 R11 R10 R9 R8
```

and a numerical value from 0 to 32767 for all others.

## Force a Single Output Function

This function allows the user to turn ON or OFF a single discrete output.  The addresses for the outputs are,

| Registers | Addresses |
|---|---|
| Y0, ..., Y15 | 00001...00016 |

The function format is as follows,

Master Query:          AA 05 aaaa nnnn CC

                     AA     slave identification
                     aaaa  starting address minus 00001 in hexadecimal format.
                     nnnn  FF00 forces output ON and 0000 forces output OFF
                     CC     checksum

Slave Response:          slave will respond with the master query packet.


## Write to a Register Function

This function allows the user to write a value to the IP3416 internal registers.  The addresses for the internal registers are,

| Registers | Addresses |
|---|---|
| T0, ..., T31 | 00001...00032 |
| C0, ..., C31 | 00033...00064 |
| D0, ..., D31 | 00065...00096 |
| R0, ..., R255 | 00097...00112 (in blocks of 16) |
| A24, ..., A31 | 00121...00128 |

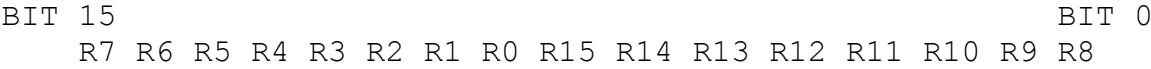The function format is as follows,

Master Query:          AA 06 aaaa nnnn CC

                     AA     slave identification
                     aaaa  starting address minus 00001 in hexadecimal format.
                     nnnn  value to write to register
                     CC     checksum

Slave Response:          slave will respond with the master query packet.

The format for setting the internal relays is,

```
 BIT 15                                                    BIT 0
     R7 R6 R5 R4 R3 R2 R1 R0 R15 R14 R13 R12 R11 R10 R9 R8
```

and a numerical value from 0 to 32767 should be used for all others.

# Force Multiple Outputs Function

This function allows the user to turn ON or OFF multiple discrete outputs.  The address for the outputs is,

| Registers | Address |
|---|---|
| Y0, ..., Y15 | 00001 |

The function format is as follows,

Master Query:
```
AA 0F 0000 0001 02 nnnn CC
```

```
AA     slave identification
nnnn   1 is ON and 0 is OFF,  in the format of  Y7, ..., Y0, Y15, ..., Y8
CC     checksum
```

Slave Response:
```
AA 0F 0000 JJJJ CC
```

```
AA     slave identification
JJJJ   new status of outputs
CC     checksum
```

# Write to Multiple Registers Function

This function allows the user to write data to multiple registers of the IP3416 at the same time.  The addresses for the internal registers are,

| Registers | Addresses |
|---|---|
| T0, ..., T31 | 00001...00032 |
| C0, ..., C31 | 00033...00064 |
| D0, ..., D31 | 00065...00096 |
| R0, ..., R255 | 00097...00112 (in blocks of 16) |
| A24, ..., A31 | 00121...00128 |

The function format is as follows,

Master Query:
```
AA 10 aaaa nnnn dddd gggg CC
```

```
AA     slave identification
aaaa   starting address minus 00001 in hexadecimal format of registers to change
nnnn   quantity of  register to set
dddd   new values to set into registers
```
  •
  •
```
gggg   new values to set into registers
CC     checksum
```

Slave Response:        slave will respond with the master query packet.