# UniOP Application Layer for Profibus DP

This document describes the application layer protocol used to support operation of UniOP panels as Profibus DP slaves.

## Contents

1. Introduction ..... 3
2. Application Layer Format ..... 4
  2.1. Request Format ..... 5
  2.2. Response Format ..... 6
3. Job Numbers And Matching Responses To Requests ..... 7
  3.1. The Need For Job Numbers ..... 7
  3.2. Why Use Two Job Numbers? ..... 7
4. Simatic S5 ..... 9
  4.1. PLC Data Type Formats ..... 9
  4.2. PLC Specific Data for Read Request ..... 11
  4.3. PLC Specific Data for Read Response ..... 12
  4.4. PLC Specific Data for Write Request ..... 12
  4.5. PLC Specific Data for Write Response ..... 12
  4.6. PLC Specific Data for Write Bit Request ..... 13
  4.7. PLC Specific Data for Write Bit Response ..... 13
  4.8. Examples ..... 14
    4.8.1. Read Request ..... 14
    4.8.2. Read Request ..... 14
    4.8.3. Write Request ..... 15
    4.8.4. Write Request ..... 15
    4.8.5. Reset Bit ..... 16
    4.8.6. Set Bit ..... 16
5. Simatic S7 ..... 17
  5.1. PLC Data Type Formats ..... 17
  5.2. PLC Specific Data for Read Request ..... 18
  5.3. PLC Specific Data for Read Response ..... 19
  5.4. PLC Specific Data for Write Request ..... 19
  5.5. PLC Specific Data for Write Response ..... 19
  5.6. PLC Specific Data for Write Bit Request ..... 20
  5.7. PLC Specific Data for Write Bit Response ..... 20
  5.8. Examples ..... 20
    5.8.1. Read Request ..... 20
    5.8.2. Read Request ..... 21
    5.8.3. Write Request ..... 21
    5.8.4. Write Request ..... 22
    5.8.5. Reset Bit ..... 22
    5.8.6. Set Bit ..... 22
6. Simatic TI 500 ..... 23
  6.1. PLC Data Type Formats ..... 23
  6.2. PLC Specific Data for Read Request ..... 26
  6.3. PLC Specific Data for Read Response ..... 26

6.4. PLC Specific Data for Write Request ..... 27
6.5. PLC Specific Data for Write Response ..... 27
6.6. PLC Specific Data for Write Bit Request ..... 28
6.7. PLC Specific Data for Write Bit Response ..... 28
6.8. Examples ..... 29
6.8.1. Read Request ..... 29
6.8.2. Read Request ..... 29
6.8.3. Read Request ..... 30
6.8.4. Read Request ..... 30
6.8.5. Write Request ..... 31
6.8.4. Write Request ..... 31
6.8.5. Reset Bit ..... 32
6.8.6. Set Bit ..... 32

# 1. Introduction

UniOP can communicate with a number of different families of PLCs via Profibus DP. Profibus DP is a fairly simple protocol designed to allow an intelligent Master to communicate with multiple Slave Input / Output units. In essence, the Profibus DP protocol consists of a continuous exchange of Output data from the Master to the Slaves and of a continuous exchange of Input data from the Slaves to the Master. For simple I/O units this data is mapped to the actual state of the physical Inputs and Outputs on the units. Profibus DP therefore allows the Master to directly control remote Outputs and to directly monitor the state of remote Inputs.

UniOP is not a simple I/O unit but an operator panel. It therefore needs to itself monitor and control the state of the Master PLC. It does this by imposing an Application Layer on the Input and Output data continuously exchanged via Profibus DP between the UniOP and the PLC. To allow UniOP to function in this way the PLC must execute some proprietary functions which interpret this Application Layer data to satisfy the UniOP requests. These proprietary functions are provided together with the UniOP for the various families of PLCs that UniOP can communicate with via Profibus DP.

The UniOP Application Layer can use either the full 32 bytes of the Profibus DP I/O data or a reduced 16 bytes.

The Application Layer imposed on the I/O data is purely proprietary to UniOP. This document describes its format and how it is used. It should be noted at this point that the UniOP Application Layer imposed on the I/O data varies slightly for the different families of PLCs supported.

## 2. Application Layer Format

Each node on a Profibus DP network needs to have its communication parameters configured by a Profibus DP configuration tool before it can take part in the network. One of the parameters that needs to be configured is the length of the Input/Output data. The Application Layer data used by UniOP for Profibus DP expects that the length of the Input/Output data be either 16 or 32 bytes.

When UniOP is connected to the Profibus DP network it will be informed by the Master of its configured Input/Output length.

UniOP is a slave on the Profibus DP network, it is not the Master. However, UniOP does itself make requests to the Master. It does this by placing data in its Input data. The Master cyclically reads the Input data from the Slaves on the network. Therefore the UniOPs Input data will cyclically be read by the Master. The UniOP formats its Input data such that it contains structured requests. The Master, as long as it understands the structured requests, will then be able to formulate a Response to the Requests and send it back to the UniOP as the UniOPs Output data. The Master formats the Output data for the UniOP such that it contains a structured response that is understood by UniOP.

## 2.1. Request Format

The format used by UniOP for its structured Requests is as follows:-

*for Input/Output data that is 32 bytes long*

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | SC | J1 | OT | R1 | ……PLC Specific Data…… | | |

| byte | 7 | 8 | 9 | ..... | 31 |
|---|---|---|---|---|---|
| | …….....… | | | | J2 |

*for Input/Output data that is 16 bytes long*

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | SC | J1 | OT | R1 | ……PLC Specific Data…… | | |

| byte | 7 | 8 | 9 | ..... | 15 |
|---|---|---|---|---|---|
| | …….....… | | | | J2 |

where:-

| | |
|---|---|
| SC | This must always be set to a value of 01H. |
| J1 | The Job Number for this request. This must match the value of J2. |
| OT | The Operation Type:<br>00H Read Bytes Operation<br>01H Write Bytes Operation<br>11H Reset Bit Operation<br>91H Set Bit Operation |
| R1 | This must always be set to 01H. |
| PLC Specific Data | The data in this section varies from PLC family to PLC family. See later sections for a description of this data. |
| J2 | The Job Number for this request. This must match the value of J1. |

## 2.2. Response Format

The format expected by UniOP for the structured Responses to its Requests is as follows:-

*for Input/Output data that is 32 bytes long*

| byte | 0 | 1 | 2 | 3 | 4 ..... 30 | 31 |
|---|---|---|---|---|---|---|
| | ST | J1 | NB | EC | ….....PLC Specific Data……. | J2 |

*for Input/Output data that is 16 bytes long*

| byte | 0 | 1 | 2 | 3 | 4 ..... 14 | 15 |
|---|---|---|---|---|---|---|
| | ST | J1 | NB | EC | ….....PLC Specific Data……. | J2 |

where:-

| | |
|---|---|
| ST | The Status Code. This specifies the current status in the processing of the request. The possible values are:-<br><br>01H. The request has been processed.<br>02H. The request is still being processed.<br><br>After making a request you have to wait until the Status Code has a value of 01H before interpreting the data. |
| J1 | The Job Number for the request being processed. After making a request you have to wait until both J1 and J2 in the Read Response packet matches the Job Number in your Request Packet before interpreting the data. |
| NB | The Number of following response bytes including EC. |
| EC | The Error Code. This specifies whether the request was completed successfully or with errors. The possible values are:-<br><br>01H. The request was completed Without errors.<br>All other values indicate an error. |
| PLC Specific Data | The data in this section varies from PLC family to PLC family. See later sections for a description of this data. |
| J2 | The Job Number for the request being processed. After making a request you have to wait until both J1 and J2 in the Read Response packet matches the Job Number in your Request Packet before interpreting the data. |

# 3. Job Numbers And Matching Responses To Requests

## 3.1. The Need For Job Numbers

As described earlier Profibus DP consists of a continuous exchange of I/O data between the Master and the Slaves. You can think of the communication between the Master and a single Slave as two mailboxes, one inside the Master and the other inside the Slave. The Master continuously fires off to the Slave the data in it's mailbox; without checking to see if the data has changed or if the Slave has requested it. It just transmits it continuously. Similarly, the Slave, when it receives the Input data from the Master, fires back off to the Master the data in it's own mailbox, again without checking to see if the data has changed.

The Application Layer imposed by UniOP is used to make requests on the PLC, for example to write some data in the PLCs memory. As the I/O data is exchanged continuously this means that any single request UniOP makes in the Application Layer will be received multiple times by the Master, one for each time the UniOPs Output data is transmitted.

It only makes sense for an actual request to be processed once by the Master as, for example, we don't want to write continuously into the PLCs memory if we request a single operation to write some data into the PLCs memory. To make sure that this is the case the Application Layer includes fields that specify the Job Number for the request. UniOP increments the Job Number every time it wants to make a new request to the Master. The proprietary program in the PLC itself stores the Job Number for the last request it actually processed for each and every UniOP. It only processes a new request from a UniOP when the Job Number in the Application layer is different to the last Job Number the PLC processed for that UniOP.

This is an essential point. The Master PLC must only process a request from a panel if the Job Number in the request is different to the Job Number that was last used in a request from that panel. This means that the Master PLC program must store, somehow, the last Job Numbers processed for each and every panel that is attached. It therefore needs to store an array of Job Numbers, one for each panel.

The possible values for the Job Numbers as used by UniOP are 01H to 7FH inclusive. At startup UniOP will set it's initial Job Number to 01H. It will then increment it by one for every new request. After a Job Number of 7FH UniOP will jump back to a Job Number of 1.

## 3.2. Why Use Two Job Numbers?

The above section explains the need for using Job Numbers in the UniOP Application Layer for Profibus DP. However, it is not clear yet why two Job Numbers, J1 and J2 are used in the Application Layer. Before we answer this let's just remind ourselves that in the Application Layer both J1 and J2 are set to the same value when we make a request and when we receive a completed response.

So why two Job Numbers? Generally speaking the task that is responsible for physically transmitting and receiving data on the transmission line is seperate from the task that prepares and processes the Application Layer data. For example, for the Siemens S5 family of PLCs the actual transmission and reception of data on the line itself is an intrinsic part of the operation of the PLC and occurs completely independently and concurrently with the PLC program.

The PLC program sends it's Application Layer data by writing the data to a particular memory location of the PLC. Similarly it receives it's Application Layer data by reading the data from another memory location of the PLC. However, the transmission and reception of data from and to these memory locations is handled completely by the inner workings of the PLC and is interrupt driven. This means that the data that is written by the PLC program could be sent immediately on the line. The PLC program therefore has no way of ensuring that it's Application Layer data is only physically transmitted when it has written the complete 32 bytes of data in the PLCs special memory location for transmission. The PLC could, in fact, physically transmit the data when only half of the Application Layer data had been written into this special memory location. A similair argument also applies to the UniOPs transmission.

Therefore a way must be found of checking whether the Application Layer data received is complete or whether only a part of the new Application Layer data was transmitted. The way this is done is by using two Job Numbers, one at the start of the Application Layer data and one at the end. The program responsible for the preperation of the Application layer data for transmission must handle the writing of this data into the special memory area for transmission in a special way. It must be handled in the following manner:

1. Write the value of J1
2. Write the Application layer data
3. Write the value of J2

In this way, the program responsible for the decoding of the received Application Layer data can check whether the full data has been received by first reading the value of J2 and then by reading the value of J1. If they have the same value then the full Application Layer data can be read if they have different values then the program must wait and try again.

Therefore, in conclusion, when UniOP is waiting for a Response it can only interpret the Output data from the PLC as the Response when:-

1. J2 in the PLCs Output data matches the Job Number it used in its request
2. J1 in the PLCs Output data matches J2 in the PLCs Output data
3. ST, the Status Byte, in the PLCs Output data is set to 1

Conversely, the PLC program should only interpret the Input data from a UniOP as a new request when:

1. J2 in the UniOPs Input data is different to the Job Number last processed for this panel by the PLC.
2. J1 in the UniOPs Input data matches J2 in the UniOPs Input data
3. SC in the UniOPs Input data is set to 1

# 4. Simatic S5

UniOP can communicate via Profibus DP with the Simatic S5 family of PLCs. This section describes the PLC specific aspects of the UniOP support for Simatic S5 via Profibus DP.

## 4.1. PLC Data Type Formats

In the Siemens S5 family of PLC's some PLC data types are Bytes while others are Words. For example, the Flag (also referred to as Merker in the Siemens literature) items are Bytes while Timers are Words. This means that Flag 1 contains a single byte of data while T1 contains a single word or two bytes of data. So to read the data for a Flag item from the PLC UniOP needs to read a single byte, while to read the data for a Timer item UniOP needs to read two bytes.
In the PLC Specific Data part of the Application layer when you want to read or write some data from or to the PLC you specify the type of PLC data that you want to access, i.e. Flag, Timer etc, the data item number from where you want to start the access and the number of data items that you want to access. For example, you might want to read the data for 3 Timer items starting from Timer 3. This would mean that you want to read the data for Timer 3, Timer 4 and Timer 5. As a single Timer contains two bytes of data this would mean that UniOP would need to read 6 bytes of Timer data. On the other hand if you wanted to read the data for 3 Flag items starting from Flag 3 this would mean that you want to read the data for Flag 3, Flag 4 and Flag 5. As a single Flag contains one byte of data this would mean that UniOP would need to read 3 bytes of Flag data.

The format of the data that is actually read from or written to the PLC is dependent on whether the data is for a PLC data type that is a Byte type, such as Flags, or a Word type, such as Timers. The format for Byte types is a simple Byte stream while the format for Word types is an Inverted Word stream. This is because the Siemens S5 PLCs use an inverted byte order within words.
These ideas are represented in the following tables.

| Device Code | Meaning | Item Number | Item Number Units | Maximum Number of Items that can be read in a single Read request *for Input / Output data that is 32 bytes long* | Maximum Number of Items that can be read in a single Read request *for Input / Output data that is 16 bytes long* | Maximum Number of Items that can be written in a single request *for Input / Output data that is 32 bytes long* | Maximum Number of Items that can be written in a single request *for Input / Output data that is 16 bytes long* |
|---|---|---|---|---|---|---|---|
| 0 | Data Block | Data Word Number | Word | 13 | 5 | 10 | 2 |
| 1 | Not Used | - | - | - | - | - | - |
| 2 | Flag | Flag Number | Byte | 26 | 10 | 20 | 4 |
| 3 | Not Used | - | - | - | - | - | - |
| 4 | Input | Input Byte Number | Byte | 26 | 10 | 20 | 4 |
| 5 | Output | Output Byte Number | Byte | 26 | 10 | 20 | 4 |
| 6 | Timer | Timer Number | Word | 13 | 5 | 10 | 2 |
| 7 | Counter | Counter Number | Word | 13 | 5 | 10 | 2 |

Table 1. Device Code Parameters For Supported PLC Data Types

| Device Code | Meaning | Format Of Data |
|---|---|---|
| 0 | Data Block | Inverted Word Stream |
| 1 | Not Used | - |
| 2 | Flag | Byte Stream |
| 3 | Not Used | - |
| 4 | Input | Byte Stream |
| 5 | Output | Byte Stream |
| 6 | Timer | Inverted Word Stream |
| 7 | Counter | Inverted Word Stream |

Table 2. Format Of Data In Frame For Supported PLC Data Types

**Byte Stream**

As an example of the format of the data within Byte Streams, here is the data format for a request for reading 5 Flag Items starting from Flag 0.

| Byte | 0 | 1 | 2 | 3 | 4 | .... |
|---|---|---|---|---|---|---|
| | M0 | M1 | M2 | M3 | M4 | .... |

**Inverted Word Stream**

As an example of the format of the data within Inverted Word Streams, here is the data format for a request for reading 3 Timer Items starting from Timer 0.

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | .... |
|---|---|---|---|---|---|---|---|
| | T0 High Byte | T0 Low Byte | T1 High Byte | T1 Low Byte | T2 High Byte | T2 Low Byte | .... |

## 4.2. PLC Specific Data for Read Request

The Read Request is used to read the value for a number of consecutive data items from the PLC. For example you can use it to read the values for Timers 1, 2, 3, 4 and 5 in one request. The Read Request is prepared by UniOP and is placed in the UniOPs Input data that is sent to the Master PLC.

The PLC Specific Data part of the Application Layer for the Read Request has the following format:-

| Byte | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|
| | DC | DB H | DB L | IN H | IN L | NI |

where:-

| | |
|---|---|
| DC | The Device Code. This specifies the type of data that should be read. See Table 1. for a list of the possible values. |
| DB H | The High Byte of the Data Block Number. This is only used if the selected device code is for Data Block. |
| DB L | The Low Byte of the Data Block Number. This is only used if the selected device code is for Data Block. |
| IN H | The High Byte of the Item Number from where the read operation should start. See Table 1. for a description of the Item Number. |
| IN L | The Low Byte of the Item Number from where the read operation should start. See Table 2. for a description of the Item Number. |
| NI | The Number Of Items, starting from the Item Number, that should be read from the PLC. See Table 1. for the maximum number of items that can be read in one request for the different device codes. |

bytes 10 - 30 are not used and can be set to whatever value.

## 4.3. PLC Specific Data for Read Response

The PLC Specific Data part of the Application Layer for the Read Response has the following format:-

| Byte | 4 ..... 15 or 30 |
|---|---|
| | …….Data…… |

where:-

Data — The requested data from the PLC. See Table 2. for a description of the format of the data for the different device codes.

## 4.4. PLC Specific Data for Write Request

The Write Request is used to write the value for a number of consecutive data items to the PLC. For example you can use it to write the values for Timers 1, 2, 3, 4 and 5 in one request. The Write Request is prepared by UniOP and is placed in the UniOPs Input data that is sent to the Master PLC.

The PLC Specific Data part of the Application Layer for the Write Request has the following format:-

| Byte | 4 | 5 | 6 | 7 | 8 | 9 | 10.....14 or 30 |
|---|---|---|---|---|---|---|---|
| | DC | DB H | DB L | IN H | IN L | NI | …..Data…… |

where:-

DC — The Device Code. This specifies the type of data that should be read. See Table 1. for a list of the possible values.

DB H — The High Byte of the Data Block Number. This is only used if the selected device code is for Data Block.

DB L — The Low Byte of the Data Block Number. This is only used if the selected device code is for Data Block.

IN H — The High Byte of the Item Number from where the write operation should start. See Table 1. for a description of the Item Number.

IN L — The Low Byte of the Item Number from where the write operation should start. See Table 1. for a description of the Item Number.

NI — The Number Of Items, starting from the Item Number, that should be written to the PLC. See Table 1. for the maximum number of items that can be written in one request for the different device codes.

Data — The data to be written to the PLC. See Table 2. for a description of the format of the data for the different device codes.

## 4.5. PLC Specific Data for Write Response

There is no PLC Specific Data part of the Application Layer for the Write Response. The success or failure of the Write request is indicated by the EC element of the Application Layer.

## 4.6. PLC Specific Data for Write Bit Request

The Write Bit Request is used to set or reset an individual bit within a PLC data item. For example you can use it to set or reset an Output. A Set operation sets the value of the bit to 1 while the Reset operation sets the value of the bit to 0. The Write Bit Request is prepared by UniOP and is placed in the UniOPs Input data that is sent to the Master PLC.

The PLC Specific Data part of the Application Layer for the Write Request has the following format:-

| Byte | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|
| | DC | DB H | DB L | IN H | IN L | BN |

where:-

| | |
|---|---|
| DC | The Device Code. This specifies the type of data that should be read. See Table 1. for a list of the possible values. |
| DB L | The Low Byte of the Data Block Number. This is only used if the selected device code is for Data Block. (Not used for the Write Bit operation). |
| IN H | The High Byte of the Item Number for the Write Request. See Table 1. for a description of the Item Number. |
| IN L | The Low Byte of the Item Number for the Write Request. See Table 1. for a description of the Item Number. |
| BN | The Bit Number that should be written. This can take a value between 0 and 7. 0 being the first bit within the MB, EB or AB byte and 7 being the last bit. |

## 4.7. PLC Specific Data for Write Bit Response

There is no PLC Specific Data part of the Application Layer for the Write Bit Response. The success or failure of the Write request is indicated by the EC element of the Application Layer.

## 4.8. Examples

### 4.8.1. Read Request

To read the data for DB10 DW4, DW5 and DW6 from the PLC the format for the Read Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 00H | 01H | 00H | 00H | 0AH |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | 00H | 04H | 03H | ..... | J1 |

A successfull response ssuming that DB10 DW4 has a value of 1223H, that DW5 has a value of 00F5H and that DW6 has a value of 9A76H, would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 07H | 01H | 12H | 23H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | F5H | 9AH | 76H | ..... | J1 |

### 4.8.2. Read Request

To read the data for MB7, MB8 and MB9 from the PLC the format for the Read Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 00H | 01H | 02H | 00H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | 00H | 07H | 03H | ..... | J1 |

A successfull response assuming thatMB7 has a value of 4CH, that MB8 has a value of 09H and that MB9 has a value of 7BH, would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 04H | 01H | 4CH | 09H | 7BH |

| byte | ..... | 15 or 31 |
|---|---|---|
| . | ..... | J1 |

### 4.8.3. Write Request

To write the data 1223H, 00F5H and 9A76H to DB10 DW4, DW5 and DW6 respectively the format for the Write Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | 00H | 00H | 0AH |

| byte | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|
| . | 00H | 04H | 03H | 12H | 23H | 00H | F5H |

| byte | 14 | 15 | ..... | 31 |
|---|---|---|---|---|
| . | 9AH | 76H | ..... | J1 |

A successfull response would be:-

| byte | 0 | 1 | 2 | 3 | ..... | 31 |
|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | ..... | J1 |

### 4.8.4. Write Request

To write the data 4CH, 09H and 7BH for MB7, MB8 and MB9 respectively to the PLC the format for the Write Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | 02H | 00H | 00H |

| byte | 7 | 8 | 9 | 10 | 11 | 12 | ..... |
|---|---|---|---|---|---|---|---|
| . | 00H | 07H | 03H | 4CH | 09H | 7BH | ..... |

| byte | 15 or 31 |
|---|---|
| . | J1 |

A successfull response would be:-

| byte | 0 | 1 | 2 | 3 | ..... | 15 or 31 |
|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | ..... | J1 |

### 4.8.5. Reset Bit

To Reset Bit 2 in MB3 the format for the Write Bit Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 11H | 01H | 02H | 00H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | 00H | 03H | 02H | ..... | J1 |

A successfull response would be:-

| byte | 0 | 1 | 2 | 3 | ..... | 15 or 31 |
|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | ..... | J1 |

### 4.8.6. Set Bit

To Set Bit 2 in MB3 the format for the Write Bit Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 91H | 01H | 02H | 00H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | 00H | 03H | 02H | ..... | J1 |

A successfull response would be:-

| byte | 0 | 1 | 2 | 3 | ..... | 15 or 31 |
|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | ..... | J1 |

# 5. Simatic S7

UniOP can communicate via Profibus DP with the Simatic S7 family of PLCs. This section describes the PLC specific aspects of the UniOP support for Simatic S7 via Profibus DP.

## 5.1. PLC Data Type Formats

In the Siemens S7 family of PLC's all PLC data types are Bytes. So to read the data for any PLC item from the PLC UniOP needs to read a single byte.

In the PLC Specific Data part of the Application layer when you want to read or write some data from or to the PLC you specify the type of PLC data that you want to access, i.e. Flag, Input etc, the data item number from where you want to start the access and the number of data items that you want to access. For example, you might want to read the data for 3 Flag Byte items starting from MB 3. This would mean that you want to read the data for MB 3, MB 4 and MB 5. As a single Flag Byte contains 1 byte of data this would mean that UniOP would need to read 3 bytes of Flag data.

The format of the data that is actually read from or written to the PLC is a simple Byte stream.

These ideas are represented in the following tables.

| **Device Code** | **Meaning** | **Item Number** | **Item Number Units** | **Maximum Number of Items that can be read in a single Read request *for Input / Output data that is 32 bytes long*** | **Maximum Number of Items that can be read in a single Read request *for Input / Output data that is 16 bytes long*** | **Maximum Number of Items that can be written in a single request *for Input / Output data that is 32 bytes long*** | **Maximum Number of Items that can be written in a single request *for Input / Output data that is 16 bytes long*** |
|---|---|---|---|---|---|---|---|
| 0 | Data Block | Data Byte Number | Byte | 26 | 10 | 20 | 4 |
| 1 | Not Used | - | - | - | - | - | - |
| 2 | Flag | Flag Byte Number | Byte | 26 | 10 | 20 | 4 |
| 3 | Not Used | - | - | - | - | - | - |
| 4 | Input | Input Byte Number | Byte | 26 | 10 | 20 | 4 |
| 5 | Output | Output Byte Number | Byte | 26 | 10 | 20 | 4 |

Table 3. Device Code Parameters For Supported PLC Data Types

| Device Code | Meaning | Format Of Data |
|---|---|---|
| 0 | Data Block | Byte Stream |
| 1 | Not Used | - |
| 2 | Flag | Byte Stream |
| 3 | Not Used | - |
| 4 | Input | Byte Stream |
| 5 | Output | Byte Stream |

Table 4. Format Of Data In Frame For Supported PLC Data Types

**Byte Stream**

As an example of the format of the data within Byte Streams, here is the data format for a request for reading 5 Flag Items starting from Flag 0.

| Byte | 0 | 1 | 2 | 3 | 4 | .... |
|---|---|---|---|---|---|---|
| | MB0 | MB1 | MB2 | MB3 | MB4 | .... |

## 5.2. PLC Specific Data for Read Request

The Read Request is used to read the value for a number of consecutive data items from the PLC. For example you can use it to read the values for Input Byte 1, 2, 3, 4 and 5 in one request. The Read Request is prepared by UniOP and is placed in the UniOPs Input data that is sent to the Master PLC.

The PLC Specific Data part of the Application Layer for the Read Request has the following format:-

| Byte | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|
| | DC | DB H | DB L | IN H | IN L | NI |

where:-

| | |
|---|---|
| DC | The Device Code. This specifies the type of data that should be read. See Table 3. for a list of the possible values. |
| DB H | The High Byte of the Data Block Number. This is only used if the selected device code is for Data Block. |
| DB L | The Low Byte of the Data Block Number. This is only used if the selected device code is for Data Block. |
| IN H | The High Byte of the Item Number from where the read operation should start. See Table 3. for a description of the Item Number. |
| IN L | The Low Byte of the Item Number from where the read operation should start. See Table 4. for a description of the Item Number. |
| NI | The Number Of Items, starting from the Item Number, that should be read from the PLC. See Table 3. for the maximum number of items that can be read in one request for the different device codes. |

bytes 10 - 30 are not used and can be set to whatever value.

## 5.3. PLC Specific Data for Read Response

The PLC Specific Data part of the Application Layer for the Read Response has the following format:-

| Byte | 4 ..... 15 or 30 |
|---|---|
| | …….Data…… |

where:-

Data — The requested data from the PLC. See Table 4. for a description of the format of the data for the different device codes.

## 5.4. PLC Specific Data for Write Request

The Write Request is used to write the value for a number of consecutive data items to the PLC. For example you can use it to write the values for Output Bytes 1, 2, 3, 4 and 5 in one request. The Write Request is prepared by UniOP and is placed in the UniOPs Input data that is sent to the Master PLC.

The PLC Specific Data part of the Application Layer for the Write Request has the following format:-

| Byte | 4 | 5 | 6 | 7 | 8 | 9 | 10.....14 or 30 |
|---|---|---|---|---|---|---|---|
| | DC | DB H | DB L | IN H | IN L | NI | …..Data…… |

where:-

DC — The Device Code. This specifies the type of data that should be read. See Table 3. for a list of the possible values.

DB H — The High Byte of the Data Block Number. This is only used if the selected device code is for Data Block.

DB L — The Low Byte of the Data Block Number. This is only used if the selected device code is for Data Block.

IN H — The High Byte of the Item Number from where the write operation should start. See Table 3. for a description of the Item Number.

IN L — The Low Byte of the Item Number from where the write operation should start. See Table 3. for a description of the Item Number.

NI — The Number Of Items, starting from the Item Number, that should be written to the PLC. See Table 3. for the maximum number of items that can be written in one request for the different device codes.

Data — The data to be written to the PLC. See Table 4. for a description of the format of the data for the different device codes.

## 5.5. PLC Specific Data for Write Response

There is no PLC Specific Data part of the Application Layer for the Write Response. The success or failure of the Write request is indicated by the EC element of the Application Layer.

## 5.6. PLC Specific Data for Write Bit Request

The Write Bit Request is used to set or reset an individual bit within a PLC data item. For example you can use it to set or reset an Output bit. A Set operation sets the value of the bit to 1 while the Reset operation sets the value of the bit to 0. The Write Bit Request is prepared by UniOP and is placed in the UniOPs Input data that is sent to the Master PLC.

The PLC Specific Data part of the Application Layer for the Write Request has the following format:-

| Byte | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|
| | DC | DB H | DB L | IN H | IN L | BN |

where:-

| | |
|---|---|
| DC | The Device Code. This specifies the type of data that should be read. See Table 3. for a list of the possible values. |
| DB L | The Low Byte of the Data Block Number. This is only used if the selected device code is for Data Block. |
| IN H | The High Byte of the Item Number for the Write Request. See Table 3. for a description of the Item Number. |
| IN L | The Low Byte of the Item Number for the Write Request. See Table 3. for a description of the Item Number. |
| BN | The Bit Number that should be written. This can take a value between 0 and 7. 0 being the first bit within the MB, IB or QB byte and 7 being the last bit. |

## 5.7. PLC Specific Data for Write Bit Response

There is no PLC Specific Data part of the Application Layer for the Write Bit Response. The success or failure of the Write request is indicated by the EC element of the Application Layer.

## 5.8. Examples

### 5.8.1. Read Request

To read the data for DB10 DB4, DB5 and DB6 from the PLC the format for the Read Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 00H | 01H | 00H | 00H | 0AH |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | 00H | 04H | 03H | ..... | J1 |

A successfull response ssuming that DB10 DB4 has a value of 12H, that DB5 has a value of F5H and that DB6 has a value of 9AH, would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 04H | 01H | 12H | F5H | 9AH |

| Byte | ..... | 15 or 31 |
|---|---|---|
| . | ..... | J1 |

### 5.8.2. Read Request

To read the data for MB7, MB8 and MB9 from the PLC the format for the Read Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 00H | 01H | 02H | 00H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | 00H | 07H | 03H | ..... | J1 |

A successfull response assuming thatMB7 has a value of 4CH, that MB8 has a value of 09H and that MB9 has a value of 7BH, would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 04H | 01H | 4CH | 09H | 7BH |

| byte | ..... | 15 or 31 |
|---|---|---|
| . | ..... | J1 |

### 5.8.3. Write Request

To write the data 12H, F5H and 9AH to DB10 DB4, DB5 and DB6 respectively the format for the Write Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | 00H | 00H | 0AH |

| byte | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|
| . | 00H | 04H | 03H | 12H | F5H | 9AH |

| byte | ..... | 31 |
|---|---|---|
| . | ..... | J1 |

A successfull response would be:-

| Byte | 0 | 1 | 2 | 3 | ..... | 31 |
|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | ..... | J1 |

### 5.8.4. Write Request

To write the data 4CH, 09H and 7BH for MB7, MB8 and MB9 respectively to the PLC the format for the Write Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | 02H | 00H | 00H |

| byte | 7 | 8 | 9 | 10 | 11 | 12 | ..... |
|---|---|---|---|---|---|---|---|
| . | 00H | 07H | 03H | 4CH | 09H | 7BH | ..... |

| byte | 15 or 31 |
|---|---|
| . | J1 |

A successfull response would be:-

| byte | 0 | 1 | 2 | 3 | ..... | 15 or 31 |
|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | ..... | J1 |

### 5.8.5. Reset Bit

To Reset Bit 2 in MB3 the format for the Write Bit Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 11H | 01H | 02H | 00H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | 00H | 03H | 02H | ..... | J1 |

A successfull response would be:-

| byte | 0 | 1 | 2 | 3 | ..... | 15 or 31 |
|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | ..... | J1 |

### 5.8.6. Set Bit

To Set Bit 2 in MB3 the format for the Write Bit Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 91H | 01H | 02H | 00H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | 00H | 03H | 02H | ..... | J1 |

A successfull response would be:-

| byte | 0 | 1 | 2 | 3 | ..... | 15 or 31 |
|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | ..... | J1 |

# 6. Simatic TI 500

UniOP can communicate via Profibus DP with the Simatic TI 500 family of PLCs. This section describes the PLC specific aspects of the UniOP support for Simatic TI 500 via Profibus DP.

## 6.1. PLC Data Type Formats

In the Simatic TI 500 family of PLC's some PLC data types are Bits while others are Words. For example, the Image Registers are individual bits while the Word Inputs are Words. This means that X1 contains a single bit of data while WX1 contains a single word or two bytes of data. So to read the data for an Image Register item from the PLC UniOP needs to read a single bit, while to read the data for a Word Input item UniOP needs to read two bytes.

As an optimisation, UniOP can read the data for multiple Bit items in a single request. It does this by packing the individual bits into bytes. For example, UniOP could read the data for X1..X8 in a single request. The data for X1..X8 would be packed into a single byte, the $0^{th}$ bit for X1, the $1^{st}$ bit for X2 and the $8^{th}$ bit for X8.

To facilitate this bit reading optimisation UniOP does not actually use the element address for Bit Sized elements directly in the request. Instead it actually accesses all Bit Sized elements as if they were components of a Byte Sized element. I.e. UniOP would never directly try to read the value of X3, instead it would read the byte that contains X1..X8 i.e. Byte 0; similarly UniOP would not try to read the value of X10, instead it would read the byte that contains X9..16 i.e. Byte 1. In the requests UniOP does NOT use the element addresses such as X3 or X10 but the Byte numbers to which they correspond. Thus the bit numbers are normalised into byte item numbers.

It should also be noted that the addresses for word sized items such as Word Input are 1 based in the Simatic TI500 programming packages, i.e. WX1, WY1 etc. When UniOP formulates the requests it normalises these addresses by making them 0 based; i.e. the address for WX1 is actually normalised to 0 in the requests, WX10 is normalised to 9.

In the PLC Specific Data part of the Application layer when you want to read or write some data from or to the PLC you specify the type of PLC data that you want to access, i.e. Image Register, Word Input etc, the normalised data item number from where you want to start the access and the number of normalised data items that you want to access. For example, you might want to read the data for 3 Word Inputs starting from Word Input 3. This would mean that you want to read the data for Word Input 3, Word Input 4 and Word Input 5. As a single Word Input contains two bytes of data this would mean that UniOP would need to read 6 bytes of Word Input data.

The format of the data that is actually read from or written to the PLC is dependent on whether the data is for a PLC data type that is a Bit type, such as Image Registers, or a Word type, such as Word Inputs. The format for Bit types is a simple Byte packed Bit stream while the format for Word types is an Inverted Word stream. This is because the Simatic TI500 S5 PLCs use an inverted byte order within words.

These ideas are represented in the following tables.

| Device Code | Meaning | Item Number | Item Number Units | Maximum Number of Items that can be read in a single Read request *for Input / Output data that is 32 bytes long* | Maximum Number of Items that can be read in a single Read request *for Input / Output data that is 16 bytes long* | Maximum Number of Items that can be written in a single request *for Input / Output data that is 32 bytes long* | Maximum Number of Items that can be written in a single request *for Input / Output data that is 16 bytes long* |
|---|---|---|---|---|---|---|---|
| 0 | Var. Mem | Var. Mem Number - 1 | Word | 13 | 5 | 10 | 2 |
| 1 | Word Input | Word Input Number - 1 | Word | 13 | 5 | 10 | 2 |
| 2 | Word Output | Word Output Number - 1 | Word | 13 | 5 | 10 | 2 |
| 3 | Cntrl. Relay | Cntrl. Relay Byte Block 0 based | Byte | 26 | 10 | 20 | 4 |
| 4 | X Image Register | X Image Register Byte Block 0 based | Byte | 26 | 10 | 20 | 4 |
| 5 | Y Image Register | Y Image Byte Block 0 based | Byte | 26 | 10 | 20 | 32 |
| 6 | T/C Preset | Timer Number | Word | 13 | 5 | 10 | 2 |
| 7 | T/C Curr. | Timer Number | Word | 13 | 5 | 10 | 2 |
| 8 | Drum S. Pre. | Counter Number | Word | 13 | 5 | 10 | 2 |
| 9 | Drum S. Curr. | Counter Number | Word | 13 | 5 | 10 | 2 |
| 10 | Edrum Cnt. Pre. | Counter Number | Word | 13 | 5 | 10 | 2 |
| 11 | Drum Cnt. Curr. | Counter Number | Word | 13 | 5 | 10 | 2 |
| 12 | Drum Time Base | Counter Number | Word | 13 | 5 | 10 | 2 |

Table 5. Device Code Parameters For Supported PLC Data Types

| Device Code | Meaning | Format Of Data |
|---|---|---|
| 0 | Var. Mem | Inverted Word Stream |
| 1 | Word Input | Inverted Word Stream |
| 2 | Word Output | Inverted Word Stream |
| 3 | Cntrl. Relay | Byte packed Bit Stream |
| 4 | X Image Register | Byte packed Bit Stream |
| 5 | Y Image Register | Byte packed Bit Stream |
| 6 | T/C Preset | Inverted Word Stream |
| 7 | T/C Curr. | Inverted Word Stream |
| 8 | Drum S. Pre. | Inverted Word Stream |
| 9 | Drum S. Curr. | Inverted Word Stream |
| 10 | Edrum Cnt. Pre. | Inverted Word Stream |
| 11 | Drum Cnt. Curr. | Inverted Word Stream |
| 12 | Drum Time Base | Inverted Word Stream |

Table 6. Format Of Data In Frame For Supported PLC Data Types

**Byte packed Bit Stream**

As an example of the format of the data within Byte packed Bit Streams, here is the data format for a request for reading X1, X2, X3, X4, X5, X6, X7 and X8.

| Byte | 0 - Bit 7 | 0 - Bit 6 | 0 - Bit 5 | 0 - Bit 4 | 0 - Bit 3 | 0 - Bit 2 | 0 - Bit 1 | 0 - Bit 0 |
|---|---|---|---|---|---|---|---|---|
| | X8 | X7 | X6 | X5 | X4 | X3 | X2 | X1 |

**Inverted Word Stream**

As an example of the format of the data within Inverted Word Streams, here is the data format for a request for reading 3 Word Input Items starting from Word Input 1.

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | .... |
|---|---|---|---|---|---|---|---|
| | WX1 High Byte | WX1 Low Byte | WX2 High Byte | WX2 Low Byte | WX3 High Byte | WX3 Low Byte | .... |

## 6.2. PLC Specific Data for Read Request

The Read Request is used to read the value for a number of consecutive data items from the PLC. For example you can use it to read the values for Word Input 1, 2, 3, 4 and 5 in one request. The Read Request is prepared by UniOP and is placed in the UniOPs Input data that is sent to the Master PLC.

The PLC Specific Data part of the Application Layer for the Read Request has the following format:-

| Byte | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|
| | DC | DR H | DR L | IN H | IN L | NI |

where:-

| | |
|---|---|
| DC | The Device Code. This specifies the type of data that should be read. See Table 5. for a list of the possible values. |
| DR H | The High Byte of the Drum Number. This is only used if the selected device code is for Edrum Cnt. Pre. |
| DR L | The Low Byte of the Drum Number. This is only used if the selected device code is for Edrum Cnt. Pre. |
| IN H | The High Byte of the Item Number from where the read operation should start. See Table 5. for a description of the Item Number. |
| IN L | The Low Byte of the Item Number from where the read operation should start. See Table 5. for a description of the Item Number. |
| NI | The Number Of Items, starting from the Item Number, that should be read from the PLC. See Table 5. for the maximum number of items that can be read in one request for the different device codes. |

bytes 10 - 30 are not used and can be set to whatever value.

## 6.3. PLC Specific Data for Read Response

The PLC Specific Data part of the Application Layer for the Read Response has the following format:-

| Byte | 4 | ..... | 15 or 30 |
|---|---|---|---|
| | …….Data…… | | |

where:-

| | |
|---|---|
| Data | The requested data from the PLC. See Table 6. for a description of the format of the data for the different device codes. |

## 6.4. PLC Specific Data for Write Request

The Write Request is used to write the value for a number of consecutive data items to the PLC. For example you can use it to write the values for Timers 1, 2, 3, 4 and 5 in one request. The Write Request is prepared by UniOP and is placed in the UniOPs Input data that is sent to the Master PLC.

The PLC Specific Data part of the Application Layer for the Write Request has the following format:-

| byte | 4 | 5 | 6 | 7 | 8 | 9 | 10.....15 or 30 |
|---|---|---|---|---|---|---|---|
| | DC | DR H | DR L | IN H | IN L | NI | ……Data…… |

where:-

| | |
|---|---|
| DC | The Device Code. This specifies the type of data that should be read. See Table 5. for a list of the possible values. |
| DR H | The High Byte of the Drum Number. This is only used if the selected device code is for Edrum Cnt. Pre. |
| DR L | The Low Byte of the Drum Number. This is only used if the selected device code is for Edrum Cnt. Pre. |
| IN H | The High Byte of the Item Number from where the write operation should start. See Table 5. for a description of the Item Number. |
| IN L | The Low Byte of the Item Number from where the write operation should start. See Table 5. for a description of the Item Number. |
| NI | The Number Of Items, starting from the Item Number, that should be written to the PLC. See Table 5. for the maximum number of items that can be written in one request for the different device codes. |
| Data | The data to be written to the PLC. See Table 6. for a description of the format of the data for the different device codes. |

## 6.5. PLC Specific Data for Write Response

There is no PLC Specific Data part of the Application Layer for the Write Response. The success or failure of the Write request is indicated by the EC element of the Application Layer.

## 6.6. PLC Specific Data for Write Bit Request

The Write Bit Request is used to set or reset an individual bit within a PLC data item. For example you can use it to set or reset an Output. A Set operation sets the value of the bit to 1 while the Reset operation sets the value of the bit to 0. The Write Bit Request is prepared by UniOP and is placed in the UniOPs Input data that is sent to the Master PLC.

The PLC Specific Data part of the Application Layer for the Write Request has the following format:-

| byte | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|
| | DC | DB H | DB L | IN H | IN L | BN |

where:-

| | |
|---|---|
| DC | The Device Code. This specifies the type of data that should be read. See Table 5. for a list of the possible values. |
| DR H | The High Byte of the Drum Number. This is only used if the selected device code is for Edrum Cnt. Pre. |
| DR L | The Low Byte of the Drum Number. This is only used if the selected device code is for Edrum Cnt. Pre. |
| IN H | The High Byte of the Item Number for the Write Request. See Table 5. for a description of the Item Number. |
| IN L | The Low Byte of the Item Number for the Write Request. See Table 5. for a description of the Item Number. |
| BN | The Bit Number that should be written. This can take a value between 0 and 7. 0 being the first bit within the item byte and 7 being the last bit. |

## 6.7. PLC Specific Data for Write Bit Response

There is no PLC Specific Data part of the Application Layer for the Write Bit Response. The success or failure of the Write request is indicated by the EC element of the Application Layer.

## 6.8. Examples

### 6.8.1. Read Request

To read the data for V3, V4 and V5 from the PLC the format for the Read Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 00H | 01H | 00H | 00H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| | 00H | 02H | 03H | ..... | J1 |

A successfull response ssuming that V3 has a value of 1223H, that V4 has a value of 00F5H and that V5 has a value of 9A76H, would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 07H | 01H | 12H | 23H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | F5H | 9AH | 76H | ..... | J1 |

### 6.8.2. Read Request

To read the data for DCP Drum 3 Step 2 , DCP Drum 3 Step 3 and DCP Drum 3 Step 3 from the PLC the format for the Read Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 00H | 01H | 0AH | 00H | 03H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| | 00H | 01H | 03H | ..... | J1 |

A successfull response assuming that DCP Drum 3 Step 2 has a value of 4CH, that DCP Drum 3 Step 3 has a value of 09H and that DCP Drum 3 Step 2 has a value of 7BH, would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 07H | 01H | 00H | 4CH | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| | 09H | 00H | 7BH | ..... | J1 |

### 6.8.3. Read Request

To read the data for X1..X32 from the PLC the format for the Read Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 00H | 01H | 04H | 00H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | 00H | 00H | 04H | ..... | J1 |

A successfull response assuming that the byte for X1..X8 has a value of A4H, that the byte for X9..X16 has a value of 4CH, that the byte for X17..X24 has a value of 12H and that the byte for X25..X32 has a value of DEH would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 05H | 01H | A4H | 4CH | 12H |

| byte | 7 | ..... | 15 or 31 |
|---|---|---|---|
| | DEH | ..... | J1 |

### 6.8.4. Read Request

To read the data for X17..X24 from the PLC the format for the Read Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 00H | 01H | 04H | 00H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | 00H | 02H | 01H | ..... | J1 |

A successfull response assuming that the byte for X16..X23 has a value of 6DH would be:-

| byte | 0 | 1 | 2 | 3 | 4 | | 15 or 31 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 02H | 01H | 6DH | …….. | J1 |

### 6.8.5. Write Request

To write the data 1223H, 00F5H and 9A76H to WY4, WY5 and WY66 respectively the format for the Write Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | 02H | 00H | 00H |

| byte | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|
| . | 00H | 03H | 03H | 12H | 23H | 00H | F5H |

| byte | 14 | 15 | ..... | 31 |
|---|---|---|---|---|
| . | 9AH | 76H | ..... | J1 |

A successfull response would be:-

| byte | 0 | 1 | 2 | 3 | ..... | 31 |
|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | ..... | J1 |

### 6.8.4. Write Request

To write the data 4CH, 09H and 7BH for CR1..8, CR9..16 and CR17..24 respectively to the PLC the format for the Write Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | 03H | 00H | 00H |

| byte | 7 | 8 | 9 | 10 | 11 | 12 | ..... |
|---|---|---|---|---|---|---|---|
| . | 00H | 00H | 03H | 4CH | 09H | 7BH | ..... |

| byte | 15 or 31 |
|---|---|
| . | J1 |

A successfull response would be:-

| byte | 0 | 1 | 2 | 3 | ..... | 15 or 31 |
|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | ..... | J1 |

### 6.8.5. Reset Bit

To Reset Y10 the format for the Write Bit Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 11H | 01H | 05H | 00H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | 00H | 01H | 01H | ..... | J1 |

A successfull response would be:-

| byte | 0 | 1 | 2 | 3 | ..... | 15 or 31 |
|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | ..... | J1 |

### 6.8.6. Set Bit

To Set Y30 the format for the Write Bit Request would be:-

| byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| | 01H | J1 | 91H | 01H | 05H | 00H | 00H |

| byte | 7 | 8 | 9 | ..... | 15 or 31 |
|---|---|---|---|---|---|
| . | 00H | 03H | 05H | ..... | J1 |

A successfull response would be:-

| byte | 0 | 1 | 2 | 3 | ..... | 15 or 31 |
|---|---|---|---|---|---|---|
| | 01H | J1 | 01H | 01H | ..... | J1 |