

RAMSEY

Micro-Tech

™2000

PROFIBUS—DP Slave
Protocol

Micro—Tech™

INDEX

- 1 PROFIBUS overview
 - 1.1. Introduction to the PROFIBUS
 - 1.2. The PROFIBUS-DP
 - 1.3. Data transfer
 - 1.4. Protection mechanisms

- 1. Implementation on the MT2000
 - 2.1. The PROFIBUS interface board.
 - 2.2. MT2000 configuration.
 - 2.3. Data transfer
 - 2.4. Timings
 - 2.5. Communication error
 - 2.5.1 Diagnostics data

- 3. Description of the variables.
 - 3.1. Groups composition
 - 3.1.1. Read groups
 - 3.1.1.1. Read GROUP 0 – Dynamic variables and status
 - 3.1.1.2. Read GROUP 1 – Sets and thresholds
 - 3.1.1.3. Read GROUP 2 – Display
 - 3.1.1.4. Read GROUP 3 – Load out
 - 3.2.1. Write groups
 - 3.2.1.1. Write GROUP 100 – Commands
 - 3.2.1.2. Write GROUP 101 – Sets and thresholds
 - 3.2.1.3. Write GROUP 102 – Load out

TABLES

- Tab 2.1 – Composition of the write telegram
- Tab 2.2 – Composition of the read buffer
- Tab 2.3 – Composition of diagnostics data
- Tab 2.4 – Key codes
- Tab 3.1 – Table of variables
- Tab 3.2 – Read group 0
- Tab 3.3 – Read group 1
- Tab 3.4 – Read group 2
- Tab 3.5 – Read group 3
- Tab 3.6 – Write group 100
- Tab 3.7 – Write group 101
- Tab 3.8 – Write group 102

APPENDIXES

- APPENDIX A – FLOATING POINT NOTATION
- APPENDIX B – INSTRUMENT CONFIGURATION. THE GSD FILE

About this manual

The *PROFIBUS-DP Slave protocol* manual consists of three parts.

Chapter 1, is a brief overview of the PROFIBUS-DP protocol.

Chapter 2, explains how PROFIBUS-DP protocol has been implemented in the Ramsey Micro-Tech instruments. The information contained in this section should be intent as general information, they are valid for all the Micro-Tech models.

Chapter 3, is a detailed description of the data that can be transmitted and received to and from the instrument. This part is specific for the instrument model.

Bibliography

- PROFIBUS DIN 19245 Part 3 Order No. 0.012
- SPC3 Siemens PROFIBUS Controller
User description 10.04.96 Order No. 6ES7 195-0BD00-8BA0

1. PROFIBUS Overview

1.1 Introduction to the PROFIBUS

The PROFIBUS born by the need to have an open, vendor independent communication system that allows to connect all the components used for the automation of technical processes.

The PROFIBUS is a standardized communication system, all its characteristics, rules and technical information are collected in the PROFIBUS DIN 19245 normative.

There are three main variation of PROFIBUS corresponding to the intended application: PROFIBUS-FMS , PROFIBUS-DP and PROFIBUS-PA.

1.2 The PROFIBUS-DP

This is the performance optimized version of PROFIBUS, specially dedicated to time-critical communication between automation system and distributed peripherals.

It is typically used to transfer I/O images between a main PLC and remote devices (normally sensors, actuators, transmitters). In this case it will be used to transfer (read and write) blocks of data also. PROFIBUS is a typical master/slave communication where the main PLC is the master and the remote devices are the slaves. It offers also the possibility to implement complex multi-master structures but this is not relevant for our purpose. The line is a 2 wires twinax cable, connection is EIA RS 485. Baud rate can be from 9,6 kbit/s up to 1,5 Mbit/s, length of the line is max 200 m at 1,5 Mbit/s extendible with repeaters.

1.3 Data Transfer

The interface between the Master and the Slave are structured in parameterization – configuration – and data transfer phase. In the parameterization and configuration phases each Slave compares its real configuration with the configuration data received from the Master. When verifying the configuration (device type, format and the length of information ...) has to be identical. This guarantees a protection against configuration fault. Only if these tests are successful the Slave can enter in the data exchange phase. Max number of data that it is possible to transfer at once in this phase is limited to 246 bytes (123 words).

1.4 Protection mechanisms

Besides the configuration check to avoid erroneous configurations, the Slave uses the Watch Dog control to detect failures on the bus. If a Slave recognizes no successful data transfer with the Master within the Watch Dog control interval, it generates an alarm condition.

2 Implementation on the MicorTech2000

2.1 The PROFIBUS interface board.

Connection to the PROFIBUS will be done through a dedicated board PROFIBUS interface board provided with the SPC3 controller. The SPC3 is an integrated circuit produced by SIEMENS, it handles the interface with the Master up to 12 Mbit/s.

The PROFIBUS boards is provided by Cannon Female 9 pin for RS485 connection. The board can be plugged into the instrument in one of the available expansion slot. The software automatically will recognize it.

2.2 MT2000 configuration and test.

The slave address and the buffer dimensions should be defined on the instrument setup: MAIN MENU 5 – PROFIB scroll.

Define here the address of the instrument in the PROFIBUS net.

```
– PROFIB SCROLL 1 –          Password: SERVICE
Slave number
  1
ENTER
```

Default:	1
Min	1
Max	126

In the next two scroll the user can define independently the read and write buffer dimensions. This possibility is very useful since it allows to reduce the address space to allocate the instrument to the minimum needed. The dimension should be entered in number of words. This number includes also the header of the telegram (see Tab 2.1) that takes 4 words , so the minimum is 5 words.

```
– PROFIB SCROLL 2 –          Password: SERVICE
Write buffer dim
      48 words
ENTER
```

Default:	48
Min	5
Max	48

```
-----  
- PROFIB SCROLL 2 -           Password: SERVICE  
Read buffer dim  
      48 words  
ENTER  
-----  
Default:      48  
Min           5  
Max          48
```

In the TEST SCROLL it is possible to check the actual status of the communication.

```
-----  
- TEST SCROLL 9 -           Password: SERVICE  
Test communication B  
      19.2 kb   Wait Prm  
-----
```

In the third line the baud rate and status are dynamically displayed. The status can be:

```
WAIT PRM      it is waiting for the PARAMETER message  
WAIT CFG      it is waiting for the CONFIGURATION message  
DATA EXC      it is in the DATA EXCHANGE phase , configuration and  
                parametrization have been completed successfully.
```

The baud rate is automatically detected by the instrument, if the displayed value changes continuously it means that there are problems on the line or the CPU is OFF.

2.3 Data transfer

During the communication activity, the MT2000 will always act as Slave, meaning that it will respond to a request from a Master device on the line, but will never attempt to send messages out.

The PROFIBUS interface allows a remote intelligent device to read and write data from and to the instrument. Data is organized in registers, these registers are collected in several groups, some of them are read only others write only groups. Groups are identified by a number, numbers from 0 are reserved for read groups, numbers from 100 are for the write groups.

Hex 0065 write group 101, scale 1

b. The `first register number`

Registers in the groups are numbered from 0 to `n` and the Master has the possibility to write only a portion of the registers composing the group. This parameter specifies the first register to write. 0 means the first register of the group.

c. The `number of registers` to write

This parameter specifies the number of register to write. It can be set to 0 to indicates all the register of the group. In case the dimension of the write buffer is less then the length in words of the group , put 0 here is the same as to put the group dimension minus 4 that is the length of the header.

d. Stamp

In the PROFIBUS protocol, data are transmitted continuously. So the same telegram is sent by the master to the slave more times. To avoid that the slave interprets continuously the same data (it would create problem also, for examples with the commands), the *stamp* is used. The slave interprets received data only if the stamp is different form the stamp received in the previous telegram. The master has only to change the stamp value when it create a new telegram. The stamp of the last interpreted telegram is re-transmitted by the instrument in the read buffer.

The “*Data section*” contains the data to write in the order MSB – LSB.

When the MT2000 processes the `package`, before to write the received data in the registers, it checks that all the parameters are correct (for ex. Group ident. Refers to an existing group) and the data to write do not overlaps specified max and min limit (each writeable register has own max and min limit, see table 3.1.).

If it detects errors the write operation fails, a DIAGNOSTICS message will be activated.

elem. #	register	type	offset (words)	note
1	Group identifier	integer	0	
2	First register	integer	2	
3	Nr of registers	integer	3	
4	Stamp	integer	1	
5	Data #1		4	
...	
...	Data #n		..	

Tab 2.1 – Composition of the write `package`Read operations

In the PROFIBUS protocol the master continuously reads data from the slave. Since the MT2000 has a big number of registers that can be sent to the master, the readable registers are collected in group also. Therefore, the Master should have the possibility to tell to the MT2000 what group and what registers of the group it needs to read.

It can do it by performing a write operation, in the mode explained above, setting as `group identifier` the number of a read group. After the MT2000 receives and interprets this write request, it starts to fill the read buffer with the requested data.

NOTE

In a write operation if the group identifier refers to a read group, only the header of the telegram is interpreted, data are ignored.

The MT2000 will remember which registers have been required the last time and it will continue to update the read buffer with their actual values.

What the Master will receive after a read operation is a buffer of data having the same structure of the write telegram (see tab 2.1). The first 4 words will contain the `group identifier`, `first register`, `number of register` and `stamp` then the `data`.

After power on, the instrument starts to update the read buffer with the register of the read group 0 (scale 1), the Master will get:

Structure				Read Data	
elem. #	register	type	offset	Value	Note
1	Group identifier	integer	0	0	Ident for read group 0
2	First register	integer	1	0	From the first reg
3	Nr of registers	integer	2	nnn	Number of word of the group 0, it depends by the MT model.
4	Stamp	integer	3	0	
5	Data #1		4	xxx	Contents first reg.
...	xxx
nnn+5	Data #nn		..	xxx	Contents last reg

Tab 2.2 – Composition of the read buffer after power on

In all the MT2000s the group 0 is the `dynamic` group, it contains all that variables that changes dynamically as for example the actual weight or speed, the totalizers etc. Other groups contain setup data as for example the weight threshold set. Normally, the Master will ask the group 0 ,only when necessary it will ask other groups.

2.4 Timings

The instrument updates almost all its basic variables (e.g. weight, rate, totals ...) each 100 milliseconds. With the same frequency the instrument updates the read buffer. If the master performs more communications in this period, it will receive more time the same data.

The write request are interpreted by the instrument in polling with a period of 100 milliseconds.

2.5. Communication errors

The alarm condition **PROFIBUS comm error** is activated by the instrument in two conditions:

1. If the SPC3 controller installed on the **PROFIBUS interface board** recognizes no successful data transfer within the watch dog control interval.
2. If the received data contains errors (value overlaps limits, register number does not exist, group number does not exist ...)

The alarm can be defined as SHUT DOWN and the shut down digital output can be used to set the system in safety condition.

In the second case the instrument also activates an EXPANDED DIAGNOSTIC request to the master.

2.5.1. Diagnostics data

The Master system has the possibility to detect a **communication error** condition by checking the diagnostics. In case of error the MT2000 will activate the request for diagnostic acquisition indicating that expanded device related diagnostics data are present.

When the master requires the diagnostics it will get:

Byte	Bit position								Diagnostics data
	7	6	5	4	3	2	1	0	
0									StatStatus1
1									StatStatus2
2									StatStatus3
3									MasterAdd
4									IdentNumberHigh
5									IdentNumberLow
6	0	0	0	0	0	0	1	0	Ext Diag – Header
7	0	0	0	0	GE	RE	LE	CE	Ext Diag – Data

Tab 2.3 – Composition of the diagnostics data

GE	Group Error	Invalid group identifier, the group does not exist
RE	Register number Error	The number of requested registers is wrong
LE	Limits Error	A write operation has been performed but the value to write overlaps the limits.
CE	Coherency Error	The Master has tried to write or read partially a variable composed by more registers (e.g only one word of a floating variable)

Refer to DIN 19245 Part 3 for details about the meaning of the first 5 bytes.

3. Description of the variables

The following table lists the variables accessible by the Master specifically for the **MT2109** models.

It is necessary at this point make a distinction between register and variables. A *register* is the basic unit of the data at which the Master can access. The groups are structured in registers, they are numbered and the identification number is used by the Master to identify the portion of a group to read or write. Their dimension is always one word.

A *variable* is the format of storing of the data in the instrument memory. Its dimension can change depending by the variable type, we can have:

INTEGER variable An integer is a variable of 1 word (1 register). It contains integer values in the ranges 0 to 65535 or +32767 to -32768. The integer variables can be structured in array of integers with various dimensions.

FLOAT variable A float is a variable of two words (2 register). It contains REAL values in the single precision IEEE format (See Appendix A at the end of this document). The single precision format can represent values in the range $3.4 \cdot 10^{+38}$ to $1.18 \cdot 10^{-38}$

NOTE

- Some variables as weight, rate are filtered before to be displayed on the instrument. For what regards the communication, instrument always sends the unfiltered value.
- A lot of real values are displayed on the instrument with a number of decimals that depends by the selected division. For what regards the communication, instrument always sends the value with all the decimals. For example:

Net weight	Displayed value:	10.2
	Sent value:	10.179982

Legend:

- register A conventional name
- type Can be: RO The register can be read but can not be written. It will be inserted only in the read groups.
 RW The register can be read or written. It will be found either in the read or in the write groups.
 WO The register can only be written. It will be inserted only in the read groups.
- low limit The minimum acceptable value for the variable. Lower values are considered as errors.

- high limit The maximum acceptable value for the variable. Higher values are considered as errors.
- refresh t. The time between two updates of the variable in the instrument's memory.
- format Can be: integer or float

register	type	low limit	high limit	refresh time [ms]	format
Commands					
Commands	WO	0	65535	–	integer
Set/Reset out#	WO	0	(*)	–	integer
Sim_Key	WO	(**)	(**)	–	integer
Dynamic values and status					
Display	RO	–	–	200	integer
Status	RO	–	–	–	integer
Alarms	RO	–	–	100	integer
IO_Image	RO	–	–	100	integer
Rate	RO	–	–	200	float
Concentration	RO	–	–	200	float
Velocity	RO	–	–	200	float
Reset_Total	RO	–	–	100	float
Operator_Total	RO	–	–	100	float
Load out (batch)					
Batch_Number	RW	0	999	100	integer
Batch_Dev	RW	0	100.0	–	integer
Batch_Set	RW	0	10000.0	–	float

register	type	low limit	high limit	refresh time [ms]	format
Batch_Preset	RW	0	10000.0	–	float
Batch_Preact	RW	0	10000.0	–	float
Batch_Total	RO	–	–	100	float
Sets and thresholds					
Scale_Cap	RO	–	–	–	float
High_Rate_Set	RW	0	105.0%	–	float
Low_Rate_Set	RW	0	105.0%	–	float
Product					
Product number	RW	0	99	–	integer

Tab 3.1 – Variables and their characteristics

- (*) Max limit is the number of digital inputs, it depends by the hardware configuration of the instrument.
- (**) All the values are accepted but only a specific set of codes are interpreted as keys. See description of the variable.

Registers in the table are explained below, they are sorted in alphabetical order.

Alarms

In the alarms register, each bit represents the status of an alarm. If the alarm is active, the relevant bit will contain '1', otherwise it will contain '0'.

Alarms 1

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0

A1.15	CLOCK FAIL	A1.06	HIGH RATE
A1.14	CORRELATOR FAIL	A1.05	HIGH RATE S1
A1.13	CORRELATOR FAIL S1	A1.04	HIGH RATE S2
A1.12	CORRELATOR FAIL S2	A1.03	LOW RATE
A1.11	BAD WEIGHT SIGNAL	A1.02	LOW RATE S1
A1.10	BAD WEIGHT SIGNAL S1	A1.01	LOW RATE S2

A1.09 BAD WEIGHT SIGNAL S2 A1.00 HIGH CONC.
 A1.08 RAM FAIL
 A1.07 RAM FAIL

Alarms 2

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0

A2.15 HIGH CONC. S1
 A2.14 HIGH CONC. S2
 A2.13 LOW CONC.
 A2.12 LOW CONC. S1
 A2.11 LOW CONC. S2
 A2.10 HIGH VEL.
 A2.09 HIGH VEL. S1
 A2.08 HIGH VEL. S2
 A2.07 LOW VEL.
 A2.06 LOW VEL. S1
 A2.05 LOW VEL. S2
 A2.04 WARM START
 A2.03 COLD START
 A2.02 PWD DURING CALIB
 A2.01 CAL TIME ELAPSED
 A2.00 CAL TIME ELAPSED S1

Alarms 3

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0

A3.15 CAL TIME ELAPSED S2
 A3.14 EXTERNAL ALARM 1
 A3.13 EXTERNAL ALARM 2
 A3.12 EXTERNAL ALARM 3
 A3.11 OVERFLOW TOTALIZER
 A3.10 OVERFLOW TOTALIZER S1
 A3.09 OVERFLOW TOTALIZER S2
 A3.08 AZT OVER LIMIT SLOT 1
 A3.07 AZT OVER LIMIT S1
 A3.06 AZT OVER LIMIT S2
 A3.05 BATCH DEVIATION
 A3.04 COMMUNICATION ERROR
 A3.03 BCD OVERFLOW
 A3.02 MATHEMATIC ERROR
 A3.01 PRINTER ERROR
 A3.00 HW CNF CHANGE

Alarms 4

1	1	1	1	1	1	0	N	N	N	N	N	N	N	N	N
5	4	3	2	1	0	9	U	U	U	U	U	U	U	U	U

A4.15	HW CNF CHANGE SLOT 2	A4.07	NU
A4.14	HW CNF CHANGE SLOT 3	A4.06	NU
A4.13	HW CNF CHANGE SLOT 4	A4.05	NU
A4.12	HW CNF CHANGE SLOT 5	A4.04	NU
A4.11	HW CNF CHANGE SLOT 6	A4.03	NU
A4.10	RIO COMM ERROR	A4.02	NU
A4.09	PROFIB COMM ERROR	A4.01	NU
A4.08	NU	A4.00	NU

Alarms 5

N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

- Batch_Dev** The maximum acceptable deviation from the batch set point. Usually entered by the user.
- Batch_Number** The number of the currently running load out or the number of the last finished one. The MT2000 will automatically increment the batch number when a new batch is started.
- Batch_Preact** The set point of the pre act for the current or the next load out. Entered or downloaded by the user (if in MANUAL mode) or calculated by the MT2000. Defines when the load out has to stop to compensate the queue of material from the loading point to the scale.
- Batch_Preset** The pre-set point for the current or the next load out. Defines when the rate will be lowered to increase batch accuracy.
- Batch_Set** The set point for the current or the next load out. Usually entered or downloaded by the user. For reading operations, the Batch_set value is updated only when a batch is started.
- Batch_Total** The current contents of the load out totalizer. Usually read at end of batch to check the result of the load out. This register is automatically cleared when a new batch is started.
- Commands** Each bit of the commands register is specified as follows.

In order to give a command, the Host must set the relevant bit to 1 and write (send) the register to the instrument. The action will be performed if the write message is accepted.

Commands

N	N	N	N	1	1	0	0	0	0	N	N	N	N	N	N
U	U	U	U	1	0	9	8	7	6	U	U	U	U	U	U

C1.15	NOT USED	C1.07	CLEAR RESET TOTAL
C1.14	NOT USED	C1.06	CLEAR OPERATOR TOTAL
C1.13	NOT USED	C1.05	NOT USED
C1.12	NOT USED	C1.04	NOT USED
C1.11	STD BY BATCH	C1.03	NOT USED
C1.10	ABORT BATCH	C1.02	NOT USED
C1.09	START BATCH	C1.01	NOT USED
C1.08	RESET ALARM	C1.00	NOT USED

Reset Commands

When the instrument receives a **Clear Reset total** or a **Reset alarm** command, it executes the command for all the scales.

Clear Operator total Clear the operator total register.

Clear Reset total Clear the reset total register.

Reset Alarms Reset any pending alarm

In order to give a command, the Host must set the relevant bit to 1 and write (send) the register to the instrument. The action will be performed if the write message is accepted.

Display Is the ASCII image of the instrument display.

IO_Image The instrument has physical inputs and outputs to which logical input and output functions are associated. As far as communication is concerned, only the status of physical inputs and outputs are transferred.

IO_image 1 – mother board inputs

N	N	N	N	1	1	0	0	N	N	N	N	N	N	N	N
U	U	U	U	1	0	9	8	U	U	U	U	U	U	U	U

IO1.11 in 4 – mother board
 IO1.10 in 3 – mother board
 IO1.09 in 2 – mother board
 IO1.08 in 1 – mother board

IO_image 2 – 4in/16out boards inputs

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0

IO2.15 in 4 board 4i/16o #1 IO2.07 in 4 board 4i/16o #3
 IO2.14 in 3 board 4i/16o #1 IO2.06 in 3 board 4i/16o #3
 IO2.13 in 2 board 4i/16o #1 IO2.05 in 2 board 4i/16o #3
 IO2.12 in 1 board 4i/16o #1 IO2.04 in 1 board 4i/16o #3
 IO2.11 in 4 board 4i/16o #2 IO2.03 in 4 board 4i/16o #4
 IO2.10 in 3 board 4i/16o #2 IO2.02 in 3 board 4i/16o #4
 IO2.09 in 2 board 4i/16o #2 IO2.01 in 2 board 4i/16o #4
 IO2.08 in 1 board 4i/16o #2 IO2.00 in 1 board 4i/16o #4

IO_image 3 – 16in/4out boards #1 inputs

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0

IO_image 4 – 16in/4out boards #2 inputs

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0

IO_image 5 – 16in/4out boards #3 inputs

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0

IO_image 6 – 16in/4out boards #4 inputs

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0

IOx.15	in 8 board 16i/4o #x	IOx.07	in 16 board 16i/4o #x
IOx.14	in 7 board 16i/4o #x	IOx.06	in 15 board 16i/4o #x
IOx.13	in 6 board 16i/4o #x	IOx.05	in 14 board 16i/4o #x
IOx.12	in 5 board 16i/4o #x	IOx.04	in 13 board 16i/4o #x
IOx.11	in 4 board 16i/4o #x	IOx.03	in 12 board 16i/4o #x
IOx.10	in 3 board 16i/4o #x	IOx.02	in 11 board 16i/4o #x
IOx.09	in 2 board 16i/4o #x	IOx.01	in 10 board 16i/4o #x
IOx.08	in 1 board 16i/4o #x	IOx.00	in 09 board 16i/4o #x

IO_image 7 – mother board outputs

N	N	N	N	N	1	0	0	N	N	N	N	N	N	N	N
U	U	U	U	U	0	9	8	U	U	U	U	U	U	U	U

IO7.10 out 3 – mother board
 IO7.09 out 2 – mother board
 IO7.08 out 1 – mother board

IO_image 8 – 4out/16in boards outputs

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0

IO8.15 out 4 board 4o/16i #1 IO8.07 out 4 board 4o/16i #3

IOx.15 out 8 board 16o/4i #x	IOx.07 out 16 board 16o/4i #x
IOx.14 out 7 board 16o/4i #x	IOx.06 out 15 board 16o/4i #x
IOx.13 out 6 board 16o/4i #x	IOx.05 out 14 board 16o/4i #x
IOx.12 out 5 board 16o/4i #x	IOx.04 out 13 board 16o/4i #x
IOx.11 out 4 board 16o/4i #x	IOx.03 out 12 board 16o/4i #x
IOx.10 out 3 board 16o/4i #x	IOx.02 out 11 board 16o/4i #x
IOx.09 out 2 board 16o/4i #x	IOx.01 out 10 board 16o/4i #x
IOx.08 out 1 board 16o/4i #x	IOx.00 out 09 board 16o/4i #x

High_Rate_Set The set point for the alarm of high rate in per cent of the Scale_Cap value.

Low_Rate_Set The set point for the alarm of low rate in per cent of the Scale_Cap value.

Master_Total The current value of the master totalizer.

Operator_Total The current value of the operator totalizer.

Rate The instantaneous rate in engineering units as currently displayed on the RUN screen.

Product number The current value of the product number as currently displayed on the RUN screen.

Reset_Total The current value of the reset totalizer of the MT2000. Reset total can be zeroed by setting a bit in the commands register.

Scale_Cap The maximum rate of the flow meter. Entered by the user in the instrument setup. It is the reference value for the high and low rate set expressed in per cent. It can be only read.

Set/Reset_Out# It is a variable that allow to the master to directly handles the spare digital outputs of the MT2000. There are 5 of this variable in the write group so that the Master has the possibility to set or reset max 5 outputs simultaneously.

To set or reset an output it is enough transmit the number of the physical out with the most significant bit set to 0 to reset and 1 to set. If the contents is 0 the command is ignored. The MT2000 does not controls if the output the Master is setting/resetting is spare or not. If the Master tries to change the status of a not-spare output, a conflict could happen.

Sim_Key

The master has the possibility to send a key code to the instrument. Interpreted key codes are:

Key	Key Code	Key	Key Code
UP_ARROW	0048 H	EIGHT	0038 H
DOWN_ARROW	0050 H	NINE	0039 H
DEC. POINT	002E H	MENU	004D H
CLEAR	0053 H	RUN	0052 H
ZERO	0030 H	F1	003F H
ONE	0031 H	F2	0040 H
TWO	0032 H	F3	0041 H
THREE	0033 H	TOTAL	003C H
FOUR	0034 H	PRINT	003B H
FIVE	0035 H	START	003D H
SIX	0036 H	STOP	003E H
SEVEN	0037 H		

Tab 2.4 – key codes

Status

The actual status of the instrument is resumed in four words, each bit has an own meanings, when the bit is 1 the associated status is true.

Status 1 – Generals

1	1	1	N	N	N	N	0	0	0	0	0	0	0	0	0
5	4	3	U	U	U	U	8	7	6	5	4	3	2	1	0

S1.15 Cumulative shut down
 S1.14 Cumulative Alarms
 S1.13 Calibration running
 S1.08 Ready

S1.07 High rate
 S1.06 Low rate
 S1.05 High Conc.
 S1.04 Low Conc.
 S1.03 High Vel.
 S1.00 Low Vel.

Status 2 – Control deviations & Batch

N	N	N	N	N	N	N	N	N	N	N	N	0	0	0	0
U	U	U	U	U	U	U	U	U	U	U	U	3	2	1	0

Status 3 – For further use

N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

Status 4 – For further use

N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

3.1. Groups composition

3.1.1. Read groups

3.1.1.1. Read GROUP 0 – Dynamic variables and status

It is the read group of default, the group the MT2106 send to the Master for a read request if the Master does not change request.

It contains dynamic values (rate, status ...) and information about the status of the instrument.

elem. #	register	type	offset (words)	Note
1	Status (1)	integer	0	
2	Status (2)	integer	1	
3	Status (3)	integer	2	
4	Status (4)	integer	3	
5	Alarms (1)	integer	4	
6	Alarms (2)	integer	5	
7	Alarms (3)	integer	6	
8	Alarms (4)	integer	7	
9	Alarms (5)	integer	8	
10	IO_Image(1)	integer	9	
11	IO_Image(2)	integer	10	
12	IO_Image(3)	integer	11	
13	IO_Image(4)	integer	12	
14	IO_Image(5)	integer	13	
15	IO_Image(6)	integer	14	
16	IO_Image(7)	integer	15	
17	IO_Image(8)	integer	16	

elem. #	register	type	offset (words)	Note
18	IO_image(9)	integer	17	
19	IO_image(10)	integer	18	
20	IO_image(11)	integer	19	
21	IO_image(12)	integer	20	
22	Product number	integer	21	
23	Rate	float	22	
24	Conc.	float	24	
25	Velocity	float	26	
26	Master_Total	float	28	
27	Reset_Total	float	30	
28	Operator_Total	float	32	

Tab 3.2 – Read group 0

3.1.1.3. Read GROUP 1 – Sets and thresholds

It contains the definition of the thresholds and the set of the reference values.

elem. #	register	type	offset (words)	Note
1	Scale_Cap	float	0	
2	High_Rate_Set	float	2	
3	Low_Rate_Set	float	4	
4	High Conc.	float	6	
5	Low Conc.	float	8	
6	High Vel.	float	10	
7	Low Vel.	float	12	

Tab 3.4 – Read group 2

3.1.1.4. Read GROUP 2 – Display

It contains the image of the instrument display.

elem. #	register	type	offset (words)	Note
1	Display(1)	integer	0	
2	Display(2)	integer	1	
3	Display(3)	integer	2	
4	Display(4)	integer	3	
5	Display(5)	integer	4	
6	Display(6)	integer	5	
7	Display(7)	integer	6	
8	Display(8)	integer	7	
9	Display(9)	integer	8	
10	Display(10)	integer	9	
11	Display(11)	integer	10	
12	Display(12)	integer	11	
13	Display(13)	integer	12	
14	Display(14)	integer	13	
15	Display(15)	integer	14	
16	Display(16)	integer	15	
17	Display(17)	integer	16	

elem. #	register	type	offset (words)	Note
18	Display(18)	integer	17	
19	Display(19)	integer	18	
20	Display(20)	integer	19	
21	Display(21)	integer	20	
22	Display(22)	integer	21	
23	Display(23)	integer	22	
24	Display(24)	integer	23	
25	Display(25)	integer	24	
26	Display(26)	integer	25	
27	Display(27)	integer	26	
28	Display(28)	integer	27	
29	Display(29)	integer	28	
30	Display(30)	integer	29	
31	Display(31)	integer	30	
32	Display(32)	integer	31	
33	Display(33)	integer	32	
34	Display(34)	integer	33	
35	Display(35)	integer	34	
36	Display(36)	integer	35	
37	Display(37)	integer	36	
38	Display(38)	integer	37	
39	Display(39)	integer	38	
40	Display(40)	integer	39	

Tab 3.5 – Read group 3

3.1.1.2. Read GROUP 3 – Load out

It contains data regarding the load out function.

elem. #	register	type	offset (words)	Note
1	Batch_Number	integer	0	
2	Batch_Dev	integer	1	
3	Batch_Total	float	2	
4	Batch_Set	float	4	
5	Batch_Preset	float	6	
6	Batch_Preact	float	8	

Tab 3.3 – Read group 1

3.1.2. Write groups

3.1.2.1. Write GROUP 100 – Commands

It gives the possibility to send commands and set/reset outputs.

elem. #	register	type	offset (words)	note
1	Commands	integer	0	
2	Product Number	integer	1	
3	Set/Res_Out#	integer	2	
4	Set/Res_Out#	integer	3	
5	Set/Res_Out#	integer	4	
6	Set/Res_Out#	integer	5	
7	Set/Res_Out#	integer	6	
8	Sim_Key	integer	7	

Tab 3.6 – Write group 100

3.1.2.3. Write GROUP 101 – Set and thresholds

It gives the possibility to write the thresholds and sets

elem. #	register	type	offset (words)	note
1	High_Rate_Set	float	0	
2	Low_Rate_Set	float	2	
3	High_Conc_Set	float	4	
4	Low_Conc_Set	float	6	
5	High_Vel_Set	float	8	
6	Low_Vel_Set	float	10	

elem. #	register	type	offset (words)	note
7	Set	float	12	
8	Preset	float	14	
9	Queue	float	16	

Tab 3.8 – Write group 102

[VAT7]

3.1.2.2. Write GROUP 102 – Load out

It gives the possibility to define the load out parameters

elem. #	register	type	offset (words)	note
1	Batch_Number	integer	0	
2	Batch_Dev	integer	1	
3	Batch_Set	float	2	
4	Batch_Preset	float	4	
5	Batch_Preact	float	6	

Tab 3.7 – Write group 101

[VAT6]

APPENDIX A
FLOATING POINT NOTATION

MT2000 store s floating point data types using the IEEE single precision format. The format contains a sign bit, an exponent field and a fraction field or mantissa.

- The represented value

the value of the number being represented is equal to the exponent multiplied by the fractional part with the sign specified by the bit sign field:

$$(-1)^{\text{sign}} * (1.0 + \text{fraction}) * s^{(\text{exp} - \text{bias})}$$

For detailed information, refer to the *ANSI IEEE Standard for Binary Floating Point Arithmetic*.

- Sign Bit

The sign of the number being represented is stored in the sign bit. If the number is positive, the sign bit contains the value 0. If it is negative, it contains the value 1. The sign bit is stored in the most significant bit of a floating point value.

- Exponent Field

Using an exponent increases the range of representable numbers. The exponent field of the number contains a 'biased' form of the exponent. A bias is subtracted from the exponent field, letting the actual exponent represent both positive and negative exponents. The value of this bias is hexadecimal 7F therefore the effective exponent (Rexp) of the number can be obtained as:

$$\text{Rexp} = \text{Exp} - 7\text{F}$$

Note

If both the exponent field and the fraction field are equal to zero, the number being represented is zero.

- Fraction field (Mantissa)

IEEE floating point format stores the fractional part of a number in a "normalized" form. It assumes that all nonzero numbers are of the following form:

$$1.\text{xxxxxxxx (binary)}$$

The character x represent either 0 or 1 (binary).

APPENDIX B
INSTRUMENT CONFIGURATION
THE GSD FILE

Profibus devices have different performances characteristics. Features differ in regards to available functionality or possible bus parameters such as baud rate and time monitoring. These parameters vary individually for each device type and vendor. For this reason the PROFIBUS Organization has standardized a database file, it contains all the characteristics of the PROFIBUS device and must be delivered by the vendor together with the device. This file has a name that depends by the vendor and an extension that is normally GSD but can be different depending by the country of origin of the device. The third letter of the extension in fact defines the country:

D	–	Default, is valid for each country
I	–	Italian
G	–	German
E	–	English
F	–	French
P	–	Portuguese
S	–	Spanish

Refers to the manual of the device you are using as Master of your PROFIBUS network to get information about how to use the GSD file.

These are the main PROFIBUS characteristics of the Ramsey MT2109 instruments.

Vendor	Ramsey
Model	MT2109
Ident number	0x2109
Protocol	Standard DP
Baud rate supported	from 9.6 kb to 12 Mb
Freeze Mode supp	NO
Sync Mode supp	NO
Auto Baud supp	YES
Set Slave Add_supp	NO
Station Type	Compact
Prm Data Length	7
Diag Data Length	8
Config Data Length	9
Input data length (write buffer)	5–48 word, coherency over word
Output data length (read buffer)	5–48 word, coherency over word

To change I/O data length

The instrument gives the possibility to define the write and read buffer dimensions. Default values are 48 words for both, if the user changes them he should also modify the GSD file so that the configuration in the instrument setup and in the file corresponds.

To make it is easy. The GSD file should be edited with a normal text editor and the value at the directive `Module =` at the bottom of the file should be changed. It normally appears as follows:

```
Module = "MT2000" 0xC0,0x6F,0x6F
```

The first value (0xC0) should be left unchanged, the second value specifies the dimension of the write buffer, the third of the read buffer.

The value to set should be computed as follows:

```
Read_Buffer_Dim   = 63 + RBD
Write_Buffer_Dim  = 63 + WBD
```

where:

RBD is the read buffer dimension defined in the instrument setup.
WBD is the write buffer dimension defined in the instrument setup.

The value should be then converted and written in hexadecimal notation.

The meaning of the default values 0x6F is:

```
6F hex      =    111 dec    =    63 + 48
```

48 is the default dimension of the buffers.

Instructions are also contained in the GSD file.

Follows the contents of the Ramsey MT2109 GSD file.

R MT2109.GSD :

```
#Profibus_DP
GSD_Revision          = 1
;
Vendor_Name           = "Ramsey"
Model_Name            = "MT2109"
Revision              = "Rev.A"
Ident_Number          = 0x2109
Protocol_Ident        = 0
Station_Type          = 0
FMS_supp              = 0
Hardware_Release      = "Rev.A"
Software_Release      = "33.GSD.01"
;
9.6_supp              = 1
19.2_supp              = 1
93.75_supp            = 1
187.5_supp            = 1
500_supp              = 1
1.5M_supp             = 1
3M_supp               = 1
6M_supp               = 1
12M_supp              = 1
;
MaxTsdr_9.6           = 60
MaxTsdr_19.2          = 60
MaxTsdr_93.75         = 60
MaxTsdr_187.5         = 60
MaxTsdr_500           = 100
MaxTsdr_1.5M          = 150
MaxTsdr_3M            = 250
MaxTsdr_6M            = 450
```

```
MaxTsdr_12M           = 800
;
Redundancy             = 0
Repeater_Ctrl_Sig     = 0
24V_Pins              = 0
Implementation_Type   = "ifak"
Bitmap_Device         = "dev.dib"
Bitmap_Diag           = "diag.dib"
Bitmap_SF             = "sf.dib"
;
Freeze_Mode_supp      = 0
Sync_Mode_supp        = 0
Auto_Baud_supp        = 1
Set_Slave_Add_supp    = 0
Min_Slave_Intervall   = 500
;
Modular_Station       = 0
Modul_Offset          = 1
Max_User_Prm_Data_Len = 7
;
Fail_Safe             = 0
Slave_Family          = 5
Max_Diag_Data_Len     = 8
;
Unit_Diag_Bit(0)      = "Coherency Error"
Unit_Diag_Bit(1)      = "Limits Error"
Unit_Diag_Bit(2)      = "Register Number Error"
Unit_Diag_Bit(3)      = "Group ID Error"
;
;
;-----
;
; Buffers dimension
;
; Standard buffer dimension is :   48 words read buffer (0x6F)
;                                     48 words
;                                     write buffer (0x6F)
```

```
;
; Dimensions can be changed by modifying the last two value in the
; 'Module' definition below.
; The value should be obtained as follows:
;
;          63 + WB_WN          or          63 + RB_WN
;
; where WB_WN is the wished words number for the write buffer,
; RB_WN is; the wished words number for the read buffer.
;
; The value should be converted in hexadecimal format. Default
; value; hex 6F corresponds to:
;
;          63 + 48      ->    111 dec    ->    6F hex
;
; WB_WN and RB_WN values must be entered also in the
; instrument setup; in decimal format.
;
;
;Write buffer  ----+      +---- Read buffer
;                |      |
;                v      v
;
;Module = "MT2000" 0xC0,0x6F,0x6F
1
EndModule
;
```