

TG Motion

version 4

Dio type group

operation manual

Revision History

date	version	revision
31 July 2017	1.0	Initial release

Contents

1. Dio group.....	4
1.1 Dio group description.....	4
1.1 Dio group components	4
2. I/O unit control and diagnostics.....	5
2.1 Principles of communication with I/O units.....	5
2.2 Important registers.....	6
2.3 Digital inputs and outputs	6
2.4 Analog inputs and outputs	7
2.5 Strain gauges	7
3. Interface for SDO object communication.....	8
3.1 Description of the structure SDO.....	8
3.2 Important registers.....	8
3.3 SDO registers: description and how to work with them.....	8
3.4 Structure SDO application examples	8
4. Appendix	10
List and description of Dio group registers	10

1. Dio group

1.1 Dio group description

Dio group makes an interface between the input and output unit (I/O unit) and the user program of the virtual PLC. It provides information on the actual status of digital and analog inputs and makes it possible to set digital and analog outputs and take readings of strain gauge output voltages. Furthermore, it can be used to control and diagnose I/O units and allows the user to check general registers.

The Dio group unifies the control interfaces of different I/O units. From the viewpoint of the PLC code user and programmer, all I/O units behave in the same way; the same registers pertain to them, by means of which input and output values can be read and set, strain gauge values can be read or global setting of the I/O unit can be changed. The units can be changed operationally without any need for the PLC code overwriting. The same PLC code may be applied to a multitude of different I/O units.

The Dio interface comprises also a mechanism for communicating with particular I/O units through SDO objects. This communication method allows the user to read or write the parameters of particular I/O units.

1.1 Dio group components

General registers

- I/O unit general settings and parameters
- I/O unit status and operating mode detection and setting
- error messages
- loading the actual status of digital inputs and setting the values of digital outputs
- loading the actual status of analog inputs and setting the values of analog outputs
- loading the actual status of strain gauge output voltages

SDO

- group of registers allowing loading and writing of SDO objects of particular I/O units

Other registers

- other auxiliary registers

2. I/O unit control and diagnostics

2.1 Principles of communication with I/O units

TG Motion can communicate simultaneously with up to 256 stand-alone I/O units through an EtherCAT interface. All registers relating to the I/O units are located in **TGM_Dio** shared memory, the size of which can be found out from the **TGM_System.HEADER.Mem_Size_Dio** register.

Loading of input values, setting of output values and loading of strain gauge voltages of all connected I/O units are carried out within a single cycle of **Cycle_Time**. Similarly, the general register values of I/O units can be changed in any cycle. The size of **Cycle_Time** is defined in **TGMotion4xx.ini** file (250 μ s, 500 μ s, 1 ms, 2 ms, etc.)

The structure **SDO** (Service Data Objects), which uses the SDO object communication, is used for parameterizing the I/O units. Using the service manual of the particular I/O unit, the user finds out the address (index and subindex) of the necessary parameter (object) and its size in bytes. Having done this, he can write the necessary value in the respective register or read the actual parameter value.

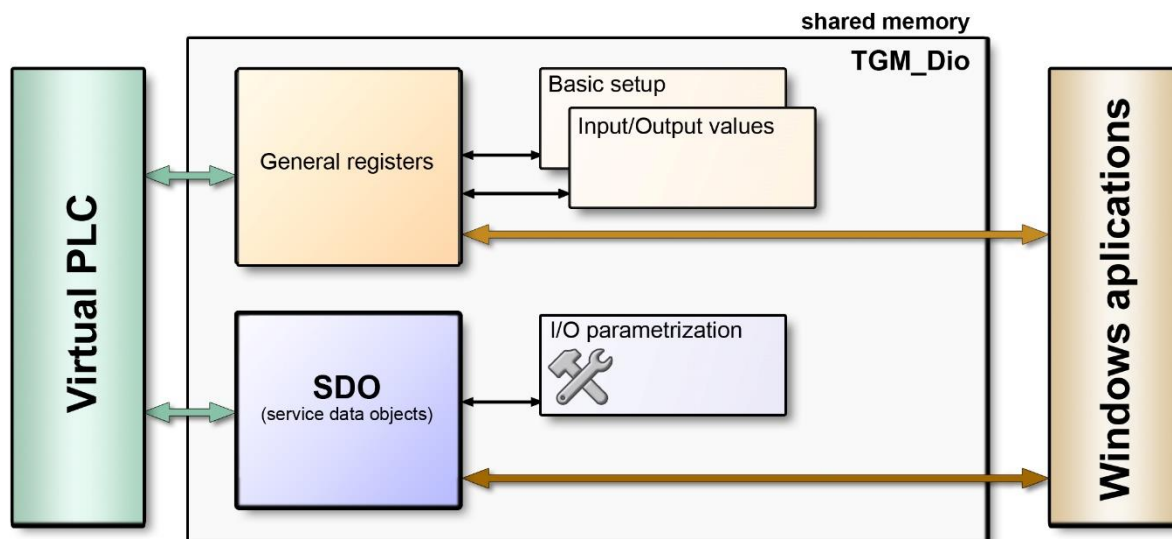


Fig. PLC – I/O – Windows interface

2.2 Important registers

Dio.Number – number of mapped I/O unit.

Dio.Node – address of the I/O unit, which was set directly on the I/O unit.

Dio.Status – actual status of the I/O unit.

Dio.Control – fault or EtherCAT communication reset:

bit 0 – fault reset.

bit 2 – EtherCAT communication reset.

Dio.Error – error messages (depending on the particular I/O unit).

Dio.Number_In – number of bytes used for digital inputs (1 byte = 8 digital inputs).

Dio.In0–7, Dio.Ext_In0–7 – 16 octets of digital inputs (logical slots 0–7, 8–15).

Dio.Number_Out – number of bytes used for digital outputs (1 byte = 8 digital outputs).

Dio.Out0–7, Dio.Ext_Out0–7 – 16 octets of digital outputs (logical slots 0–7, 8–15).

Dio.Number_Analn – number of connected analog inputs.

Dio.AI0–3, Dio.Ext_AI0–3 – 8 analog inputs.

Dio.Number_AnaOut – number of connected analog outputs.

Dio.AO0–3, Dio.Ext_AO0–3 – 8 analog outputs.

Dio.Number_MeasureAmpl – number of connected strain gauges.

Dio.Ext_MeasureAmpl0–7 – output voltages of strain gauges 0–7.

Dio.EtherCATState – I/O unit communication status from EtherCAT point of view.

For a complete list of all Dio group registers including their description, refer to Appendix.

2.3 Digital inputs and outputs

The interface makes it possible to load as many as 128 digital inputs and set up to 128 digital output of a single I/O unit. The digital inputs are mirrored by octets in 16 registers **Dio.In0–7, Dio.Ext_In0–7** (logical input slots 0–15), the digital outputs, in registers **Dio.Out0–7, Dio.Ext_Out0–7** (logical output slots 0–15). Information on the number of digital input octets connected and digital output octets connected is available in registers **Dio.Number_In** and **Dio.Number_Out**. The interface updates the input and output status within one **Cycle_Time** which has been set in **TGMotion4xx.ini**.

If **Dio.Number_In** register value = 5, **TG Motion** will update in every cycle **Cycle_Time** only the values of registers **Dio.In0–4** (logical input slots 0–4) according to the values of the digital input octets, which are actually connected. The other digital input register values may be random.

Similarly, if, for example, **Dio.Number_Out** = 3, **TG Motion** will set in every cycle **Cycle_Time** only the first three octets of digital outputs, depending on the values of registers **Dio.Out0–2** (logical output slots 0–2). The values of other digital output registers are ignored.

TG Motion sets the values of **Dio.Number_In** and **Dio.Number_Out** registers during startup or EtherCAT initialization. Their zero values mean that no digital inputs or digital outputs are available.



It holds for PLC that for the program to be executed correctly, the user must, at the beginning of his PLC loop, write the digital input status into the internal variables to ensure that all logical operations in the loop body are running with the same digital input status. The same procedure is recommended to apply to the digital output value settings. This procedure is not required to be followed in the case of PLC Program_04.

2.4 Analog inputs and outputs

The interface makes it possible to load as many as 8 analog inputs and set up to 8 digital outputs of a single I/O unit. The analog inputs are mirrored in 16 registers **Dio.AI0–3** and **Dio.Ext_AI0–3**, the analog outputs, in registers **Dio.AO0–3** and **Dio.Ext_AO0–3**. Information on the number of analog inputs connected and analog outputs connected is available in registers **Dio.Number_Analn** and **Dio.Number_AnaOut**. The interface updates the input and output status within one **Cycle_Time** which has been set in **TGMotion4xx.ini**.

If **Dio.Number_Analn** register value = 2, **TG Motion** will update in every cycle **Cycle_Time** only the values of registers **Dio.AI0–1** according to the values of the first two analog inputs, which are actually connected. **TG Motion** does not deal with the values of other analog input register values, which may be random.

Similarly, if, for example, **Dio.Number_AnaOut** = 3, **TG Motion** will set in every cycle **CycleTime** only the first three analog outputs, according to the values of registers **Dio.AO0-2**. The values of other analog output registers are ignored.

TG Motion sets the values of **Dio.Number_Analn** and **Dio.Number_AnaOut** registers during startup or EtherCAT initialization. Their zero values mean that no analog inputs or analog outputs are available.



It holds for PLC that for the program to be executed correctly, the user must, at the beginning of his PLC loop, write the analog input status into the internal variables to ensure that all logical operations in the loop body are running with the same analog input status. The same procedure is recommended to apply to the analog output value settings. This procedure is not required to be followed in the case of PLC Program_04.

2.5 Strain gauges

The interface allows loading of up to 8 strain gauge values. The output voltage values are mirrored in **Dio.Ext_MeasureAmpl0–7** registers. Information on the number of the strain gauges connected is comprised in **Dio.Number_MeasureAmpl** register. The interface updates the strain gauge registers within a single **Cycle_Time**, which has been set in **TGMotion4xx.ini**.

If **Dio.Number_MeasureAmpl** register value = 4, **TG Motion** will update in every cycle **Cycle_Time** only the values of **Dio.Ext_MeasureAmpl0–3** registers according to the values of the first four strain gauges, which are actually connected. **TG Motion** does not deal with the values of other strain gauge register values, which may be random.

The **Dio.Number_MeasureAmpl** register is set by **TG Motion** at the startup or during EtherCAT initialization. The value of **Dio.Number_MeasureAmpl** = 0 means that no strain gauges are actually connected.



It holds for PLC that for the program to be executed correctly, the user must, at the beginning of his PLC loop, write the strain gauge status into the internal variables to ensure that all logical operations in the loop body are running with the same strain gauge status. This procedure is not required to be followed in the case of PLC Program_04.

3. Interface for SDO object communication

3.1 Description of the structure SDO

The structure, which uses SDO objects (Service Data Objects), is employed to parameterize particular I/O units. Using the service manual of the particular unit, the user finds out the address (index and subindex) of the necessary parameter (object) and its size in bytes. Having done this, he can write the necessary value in the respective register or read the actual parameter value.

2.2 Important registers

SDO.Control – SDO object write and read control.

SDO.Status – determines the SDO object write, read or error status.

SDO.Index – index of an SDO object.

SDO.SubIndex – subindex of an SDO object.

SDO.Data – data to write or loaded data; in the case of error (SDO.Status = 2), error code.

For a complete list of all Dio group registers, including their description, refer to Appendix.

3.3 SDO registers: description and how to work with them

The interface is used write and read any parameters of a particular I/O unit through the communication of SDO objects. The communication control is enabled by means of **SDO.Control** and **SDO.Status** registers. The parameters are addressed by means of **SDO.Index** and **SDO.SubIndex** registers. The value of these variables is I/O unit specific. The parameter assignment table is shown in the respective unit manual.

3.4 Structure SDO application examples

a) Parameter writing

Writing start

SDO.NumberByte = 4	<i>number of bytes in given parameter (1–4)</i>
SDO.Index = 24672	<i>index of given parameter</i>
SDO.SubIndex = 1	<i>subindex of given parameter</i>
SDO.Data = 1234	<i>data to be written</i>
SDO.Control = 1	<i>write request</i>

<i>Waiting for SDO.Control = 0</i>	<i>waiting for writing end</i>
------------------------------------	--------------------------------

Writing correctness test

<i>When SDO.Status = 0</i>	<i>write process is accomplished</i>
<i>When SDO.Status = 2</i>	<i>writing error, error number is in SDO.Data register</i>

b) Parameter reading**Reading start**

SDO.NumberByte = 2

number of bytes in given parameter (1–4)

SDO.Index = 24672

index of given parameter

SDO.SubIndex = 0

subindex of given parameter

SDO.Control = 2

read request

Waiting for SDO Control = 0

*waiting for reading end***Reading correctness test**

When SDO.Status = 0

reading finished, actual data in SDO.Data

Then Data = SDO.Data

When SDO.Status = 2

reading error

Then SDO.Data = error code

SDO.Data contains the error code

4. Appendix

List and description of Dio group registers

General register

name	access	offset	description
Number	R	0	number of the mapped I/O unit, interface logical number (unit 0 – Number = 0, unit 1 – Number = 1, etc.)
Node	R	4	address of the physical I/O unit, which is set on the unit directly
Type	R	8	I/O unit type – is not indicated in TGMotion4xx.ini, TG Motion finds out automatically the connected I/O units and updates the Dio.Type register: 0x14 DIO module TG 48/40/4/4 0xF4 DIO module TG 40/24/0/0 0x8 DIO strain gauge old design 0x808 DIO strain gauge new design 0x400 FESTO FB38 0x200 B&R 0x100 BECKHOFF BK1120 0x101 BECKHOFF EK1100
Control	RW	12	bit 0 = fault reset bit 2 = EtherCAT communication reset
State	R	16	- specifies the actual status of the I/O unit: 0 = does not communicate, no communication was established or the module is faulty 1 = I/O module is operational (no faults) - for more details, EtherCATState register can be used
Error	R	20	uses the code of the first detected fault, if the I/O unit is in fault; the value meaning depends on the particular type of the unit (see the I/O unit manual)
Number_In	R	24	the number of bytes used for digital inputs (1 byte = 8 digital inputs)
In0	R	28	digital input octet In0 (logical input slot 0)
In1	R	32	digital input octet In1 (logical input slot 1)
In2	R	36	digital input octet In2 (logical input slot 2)
In3	R	40	digital input octet In3 (logical input slot 3)
In4	R	44	digital input octet In4 (logical input slot 4)
In5	R	48	digital input octet In5 (logical input slot 5)
In6	R	52	digital input octet In6 (logical input slot 6)
In7	R	56	digital input octet In7 (logical input slot 7)
Ext_In0	R	60	digital input octet Ext_In0 (logical input slot 8)
Ext_In1	R	64	digital input octet Ext_In1 (logical input slot 9)
Ext_In2	R	68	digital input octet Ext_In2 (logical input slot 10)
Ext_In3	R	72	digital input octet Ext_In3 (logical input slot 11)
Ext_In4	R	76	digital input octet Ext_In4 (logical input slot 12)
Ext_In5	R	80	digital input octet Ext_In5 (logical input slot 13)
Ext_In6	R	84	digital input octet Ext_In6 (logical input slot 14)
Ext_In7	R	88	digital input octet Ext_In7 (logical input slot 15)
Number_Out	R	92	the number of bytes used for digital outputs (1 byte = 8 digital outputs)
Out0	RW	96	digital output octet Out0 (logical output slot 0)
Out1	RW	100	digital output octet Out1 (logical output slot 1)
Out2	RW	104	digital output octet Out2 (logical output slot 2)
Out3	RW	108	digital output octet Out3 (logical output slot 3)
Out4	RW	112	digital output octet Out4 (logical output slot 4)
Out5	RW	116	digital output octet Out5 (logical output slot 5)
Out6	RW	120	digital output octet Out6 (logical output slot 6)
Out7	RW	124	digital output octet Out7 (logical output slot 7)
Ext_Out0	RW	128	digital output octet Ext_Out0 (logical output slot 8)
Ext_Out1	RW	132	digital output octet Ext_Out1 (logical output slot 9)
Ext_Out2	RW	136	digital output octet Ext_Out2 (logical output slot 10)
Ext_Out3	RW	140	digital output octet Ext_Out3 (logical output slot 11)
Ext_Out4	RW	144	digital output octet Ext_Out4 (logical output slot 12)
Ext_Out5	RW	148	digital output octet Ext_Out5 (logical output slot 13)
Ext_Out6	RW	152	digital output octet Ext_Out6 (logical output slot 14)
Ext_Out7	RW	156	digital output octet Ext_Out7 (logical output slot 15)
Number_Analn	R	160	number of connected analog inputs
AI0	R	164	analog input AI0
AI1	R	168	analog input AI1
AI2	R	172	analog input AI2

name	access	offset	description
AI3	R	176	analog input AI3
Ext_AI0	R	180	analog input Ext_AI0
Ext_AI1	R	184	analog input Ext_AI1
Ext_AI2	R	188	analog input Ext_AI2
Ext_AI3	R	192	analog input Ext_AI3
Number_AnaOut	R	196	number of connected analog outputs
AO0	RW	200	analog output AO0
AO1	RW	204	analog output AO1
AO2	RW	208	analog output AO2
AO3	RW	212	analog output AO3
Ext_AO0	RW	216	analog output Ext_AO0
Ext_AO1	RW	220	analog output Ext_AO1
Ext_AO2	RW	224	analog output Ext_AO2
Ext_AO3	RW	228	analog output Ext_AO3
Number_MeasureAmpl	R	232	number of connected strain gauges
Ext_MeasureAmpl0	R	236	output voltage of strain gauge 0
Ext_MeasureAmpl1	R	240	output voltage of strain gauge 1
Ext_MeasureAmpl2	R	244	output voltage of strain gauge 2
Ext_MeasureAmpl3	R	248	output voltage of strain gauge 3
Ext_MeasureAmpl4	R	252	output voltage of strain gauge 4
Ext_MeasureAmpl5	R	256	output voltage of strain gauge 5
Ext_MeasureAmpl6	R	260	output voltage of strain gauge 6
Ext_MeasureAmpl7	R	264	output voltage of strain gauge 7
EtherCATState	R	268	I/O unit communication status from EtherCAT point of view: 0x01 = Init 0x02 = Pre-Operational 0x04 = Safe-Operational 0x08 = Operational

SDO object registers

name	access	offset	description
Control	RW	272	SDO object write and read control: 0 = ready 1 = writing request 2 = reading request
Status	R	276	communication progress or a message whether the communication was successful: 0 = communication was successful 1 = communication is in progress 2 = communication error
NumberByte	W	280	SDO object size (bytes)
Index	W	284	address from where to read the data or where to write the data
SubIndex	W	288	address from where to read the data or where to write the data
Data	RW	292	data to write or read, or error code
SdoResponseTries	R	296	counter of unsuccessful communications – internal variable

Other registers

name	access	offset	description
Rx_Bufer	RW	300	not used
Tx_Bufer	RW	332	not used
Type_Specific_Data	RW	364	I/O unit specific data
SysTimeDifference	R	396	last recorded difference between local time and reference time [ns]