**DIGITAL SERVOAMPLIFIER**

# TGA-24-9/20

## Description of the set up software S.C.D.

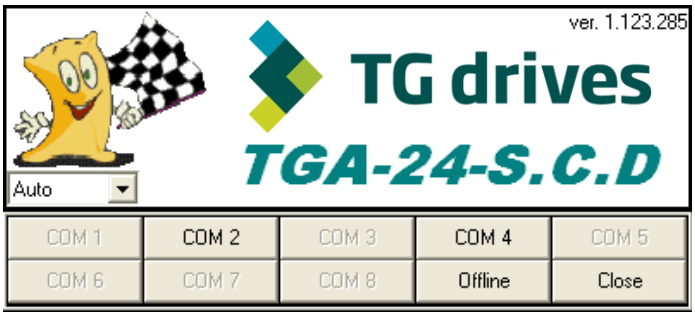
Edition 01/2014


TG drives

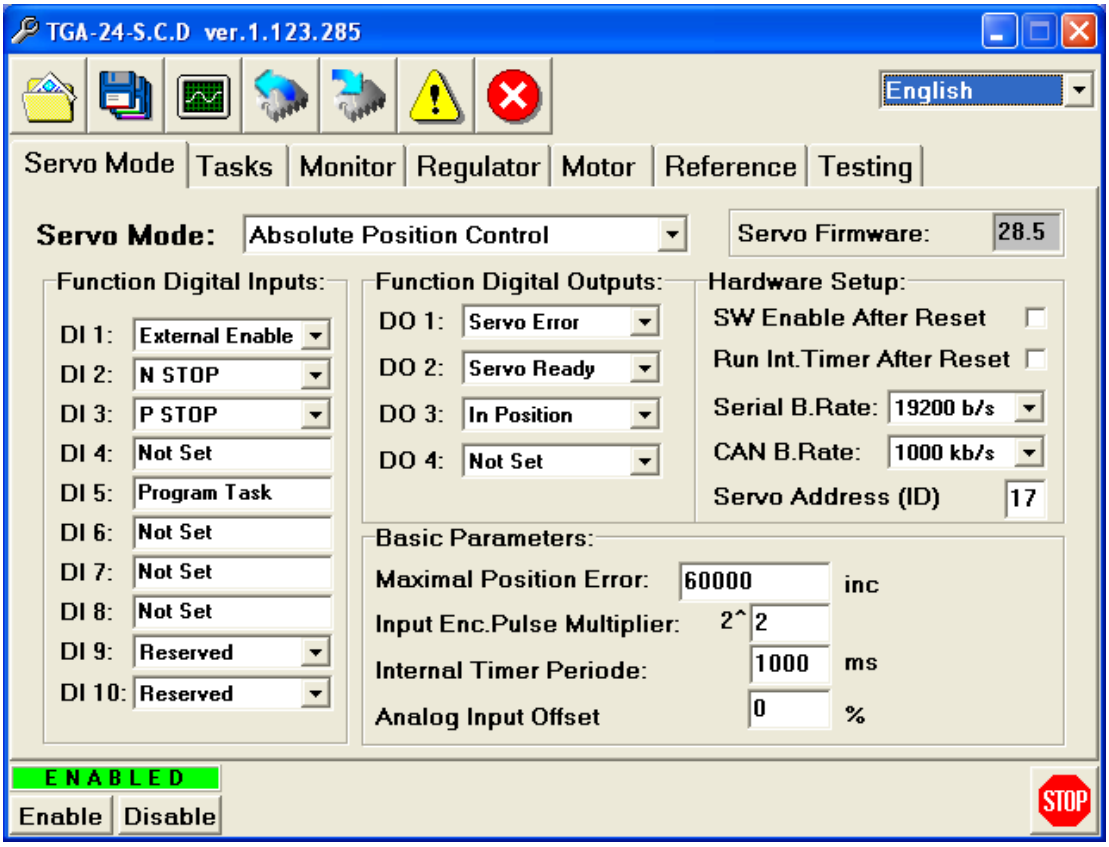
# Contents

# 1 Introduction

Servo-controller TGA is suitable to control AC synchronous servomotor..
The TGA controller has pre-program several functions and operating modes usable for your applications. The function can be set by set-up software S.C.D. (Service, Control, Drive).
To install program S.C.D to your computer start the program SETUP_TGA_1_XXX.exe. The program S.C.D. communicate with servo-controller TGA through serial line RS232. The controller has to be connected with PC by communication cable (see Hardware manual). If your PC is not fitted by serial port RS232 you can use suitable converter (e.g. USB-RS232).
After starting program S.C.D. the Window offering communication port (COM1-COM8) or Offline work. After selection correct port the communication between S.C.D. program and a TGA should be set. The program should find correct communication speed automatically.




After successfully setting of the communication or offline mode the Windows will appear.



## 2 Function icons

Load parameters from the file to the servo-controller. The servo-controller has to be in DISABLE mode. To memorize the parameters after switching off the controller the order SAVE to the EEPROM should be executed.

Save parameters to the file.

Oscilloscope.

Reading parameters saved in EEPROM from the servo-controller.

Saving parameters to the EEPROM. Actual setting will be memorized after switching off the servo-controller.

Clear fault.

Reset servo-controller.


Stop of the servomotor (mode Digital Speed = 0 is executive)


On/Off the power stage of the servo-controller.

# 3 Description of the functions

## 3.1 MENU-Servo Mode

### 3.1.1 Servo Mode

The Operation Mode of the Servo Controller:

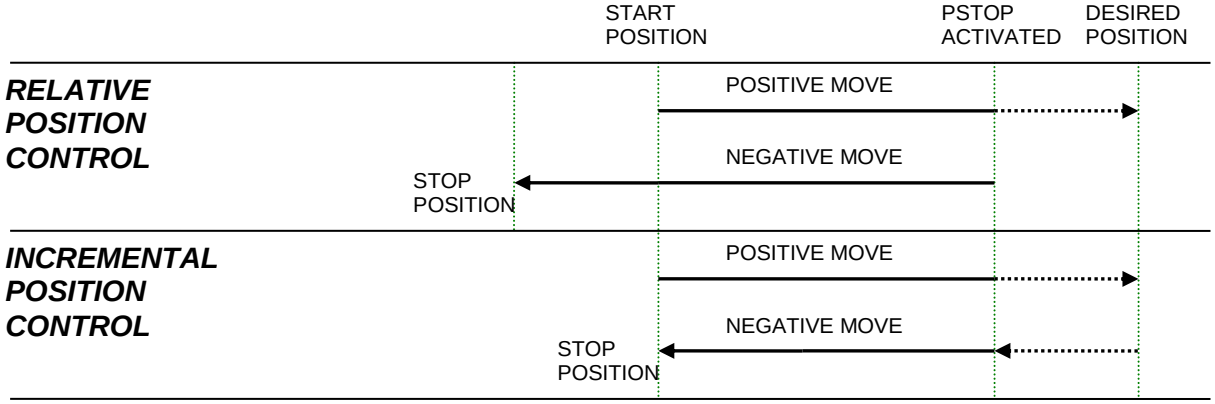
| | |
|---|---|
| Digital Torque Control | The Torque Control with digital set-point (RS 232, CAN BUS) |
| Digital Speed Control | The Speed Control with digital set-point (RS 232, CAN BUS) |
| Analogue Torque Control | The Torque Control with analogue set-point (analogue input 0-5V- terminal X2, range 0-10V is possible to set on the card).<br>0V-is maximal negative value of the desired value (see MENU Motor)<br>2,5V-zero torque. Zero can be shifted by Analogue Input Offset<br>5V-is maximal positive value (MENU Motor) |
| Analogue Speed Control | The Speed Control with analogue set-point (analogue input 0-5V- terminal X2, range 0-10V is possible to set on the card).<br>0V-is maximal negative value of the desired value (see MENU Motor)<br>2,5V-zero torque. Zero can be shifted by Analogue Input Offset<br>5V-is maximal positive value (MENU Motor) |
| Absolute Position Control | The movement to an absolute target position, referred to the reference point. Reference point can be zero position from resolver after switching on the servo-controller or actual position can be set to zero in mode REFERENCE.<br>The maximum range of movement is limited to $+/-2^{31}$ increments (+/- 32768 motor turns). The internal profile generator calculates every 1 ms new reference position and speed according to set deceleration, acceleration and maximal speed. |
| Incremental Position Control | The movement relative to last target (set-point) position.<br>The maximum range of the setting value is limited to $+/-2^{30}$ increments. The internal profile generator calculates every 1 ms new reference position and speed according to set deceleration, acceleration and maximal. |
| Relative Position Control | The movement relative to actual start position.<br>The maximum range of the setting value is limited to $+/-2^{30}$ increments. The internal profile generator calculates every 1 ms new reference position and speed according to set deceleration, acceleration and maximal. |
| Stepper Motor Control | Stepper Motion Control. The drive accepts signals Pulse and Direction generated by a control system on terminal X2. The pulses are possible to multiply by a multiplied constant. Compare to a standard stepper motor the servo-drive cannot lose pulses, thanks to feedback. In case of overloading the servo-drive indicate an error. After setting the mode it is necessary to SAVE parameters to EEPROM and RESET the servo-controller. |
| Encoder Control | Encoder Follower-electronic gearing. A master incremental encoder or encoder output of a master servo-amplifier is connected to terminal X2. The pulses are possible to multiply by multiplied constant.<br>After setting the mode it is necessary to SAVE parameters to EEPROM and RESET the servo-controller. |
| Trajectory CAN Control | Required position is generated by a control system and is sent to the servo-controller through CAN BUS. Upgrade of position should to be in period 0,7 - 2 ms. |

### 3.1.2 Digital Inputs

The servo-controller TGA has pre-program some functions. The functions are optional.

| Input | Function | Description |
|---|---|---|
| DI1 | External enable | If input is „High“ then torque to the servomotor is enabled. |
| DI2 | N STOP | Negative direction of the motion is blocked. Servomotor will stop on Emergency Ramp. Not active in analogue and digital torque modes. |
| DI3 | P STOP | Positive direction of the motion is blocked. Servomotor will stop on Emergency Ramp. Not active in analogue and digital torque modes. |
| DI4 | | Function is selectable in MENU Tasks |
| DI5 | | Function is selectable in MENU Tasks |
| DI6 | | Function is selectable in MENU Tasks |
| DI7 | | Function is selectable in MENU Tasks |
| DI8 | | Function is selectable in MENU Tasks |
| DI9 | | Reserved |
| DI10 | | Reserved |

**NSTOP and PSTOP function in RELATIVE and INCREMENTAL POSITION modes**




### 3.1.3 Digital outputs

| Output | Function | Description |
|---|---|---|
| D01 | Servo Error | If an error occurred the output is “High“. The faults can happened:<br>Resolver error<br>Overheating of the controller<br>Over heating of the servo motor<br>Short circuit on power stage or cable or motor<br>Over loading of the servomotor<br>Position error<br>Under voltage<br>Disable (if digital input DI1 is set) |
| D02 | Servo ready | The output is „High“ if torque to the servomotor is enabled. |
| D03 | In position | The output is „High“ if motor reach required position. |
| D04 | Speed=0 | The output is „High“ if motor has speed=0. |

### 3.1.4 Servo Firmware

Number of the firmware

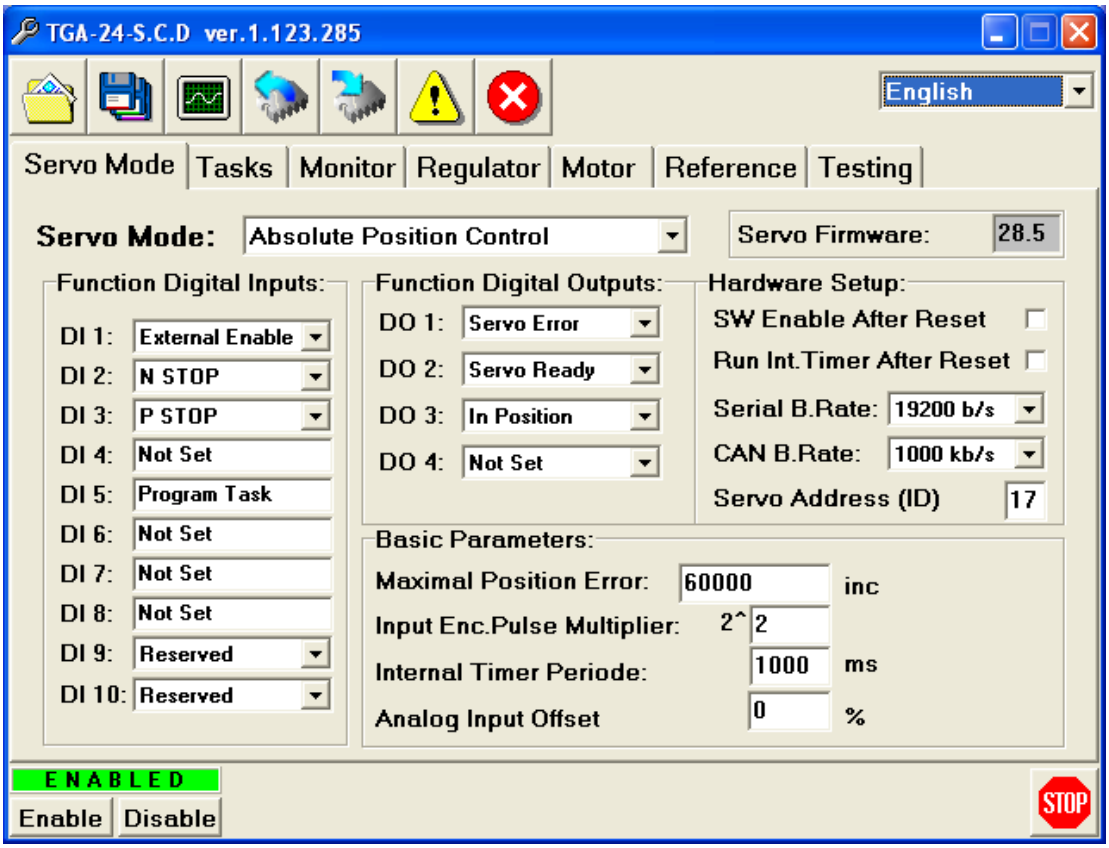
### 3.1.5 Hardware setup

| | |
|---|---|
| SW Enable After Reset | If set, torque to the motor is enabled after reset. |
| Run Int. Timer After Reset: | If set, internal timer is started after Reset (see Tasks). |
| Serial B. Rate | Setting communication rate of the serial line.<br>9600 b/s, 19200 b/s, 38400 b/s, 57600 b/s.<br>New rate is accepted after Saving the parameters to EEPROM and Reset. |
| CAN B. Rate | Setting communication rate of the CAN BUS.<br>125 kb/s, 250 kb/s, 500 kb/s, 1000 kb/s<br>New rate is accepted after Saving the parameters to EEPROM and Reset. |
| Servo Address (ID) | Setting of the address 1-63 |

### 3.1.6 Basic Parameters

| | |
|---|---|
| Maximal Position Error | Maximal allowable position error in increments. |
| Input. Enc. Pulse Multiplier | Binary exponent of the multiplier of the input pulses (ENCODER Control, Stepper Motor Control). If the value is 0 then pulses are multiplied 1. |
| Internal Input Period | Setting of the Period of the internal timer. (See MENU Tasks) |
| Analogue Input Offset | Shifting of the analogue input |

## 3.2 MENU Tasks




It is possible to program 10 functions to the servo controller. The functions can be started by rising or falling edge of a digital input, internal timer, or signal In position or Speed=0.

### 3.2.1 Task Enable

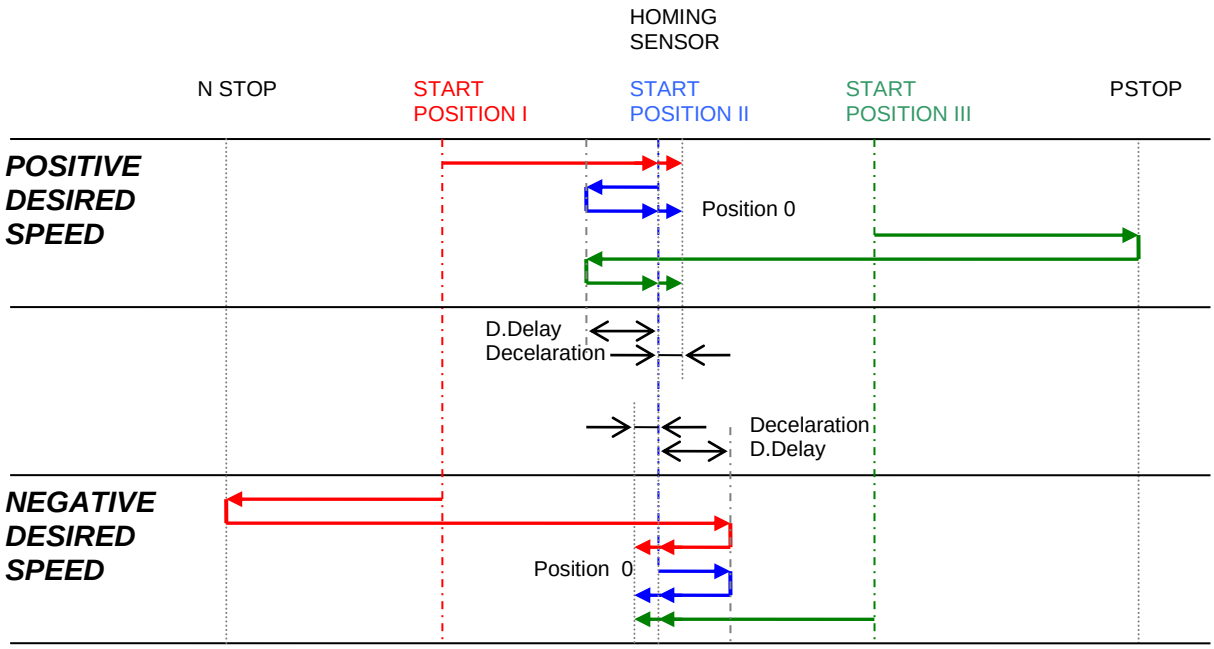
Enabling required task.

### 3.2.2 Task Function

It is possible to select the functions::

| | |
|---|---|
| Absolute Position | Absolute movement |
| Incremental Position | Relative movement to the last target position |
| Relative Position | Relative movement to the last actual position |
| Stop Motion | Stop motion |
| Set Ref. Position | Set to zero actual position |
| Digital Speed | Speed control |
| Digital Torque | Torque control |
| Servo Reset | Reset s |
| Enable Internal Timer | Enabling internal timer |
| Disable Internal Timer | Disabling internal timer |
| Homing | Start Homing. Settable only in TASK No1.Reference sensor has to be connected to Input 4. When N STOP and P STOP functions are activated, motor reverse automatically.<br>D.delay - time for go down from reference sensor when sensor is activated after start.<br>Speed and direction of the homing is settable by **Speed**. |

**TASK HOMING description**




### 3.2.3 Task Parameters

According to chosen function some windows are appeared to set parameters of the function.

| | |
|---|---|
| Delay | Delay of the start of a function after a edge come. |
| Position/Speed/Current | Target position in increments valid for absolute, incremental and relative position modes.<br>Desired speed in speed mode<br>Desired current in torque mode |
| Speed | Maximal speed in position modes. |
| Acc | Acceleration (in ms to 3000 rpm). |
| Dec | Deceleration (in ms to 3000 rpm) |

Required function can be started by the digital inputs or virtual signals.

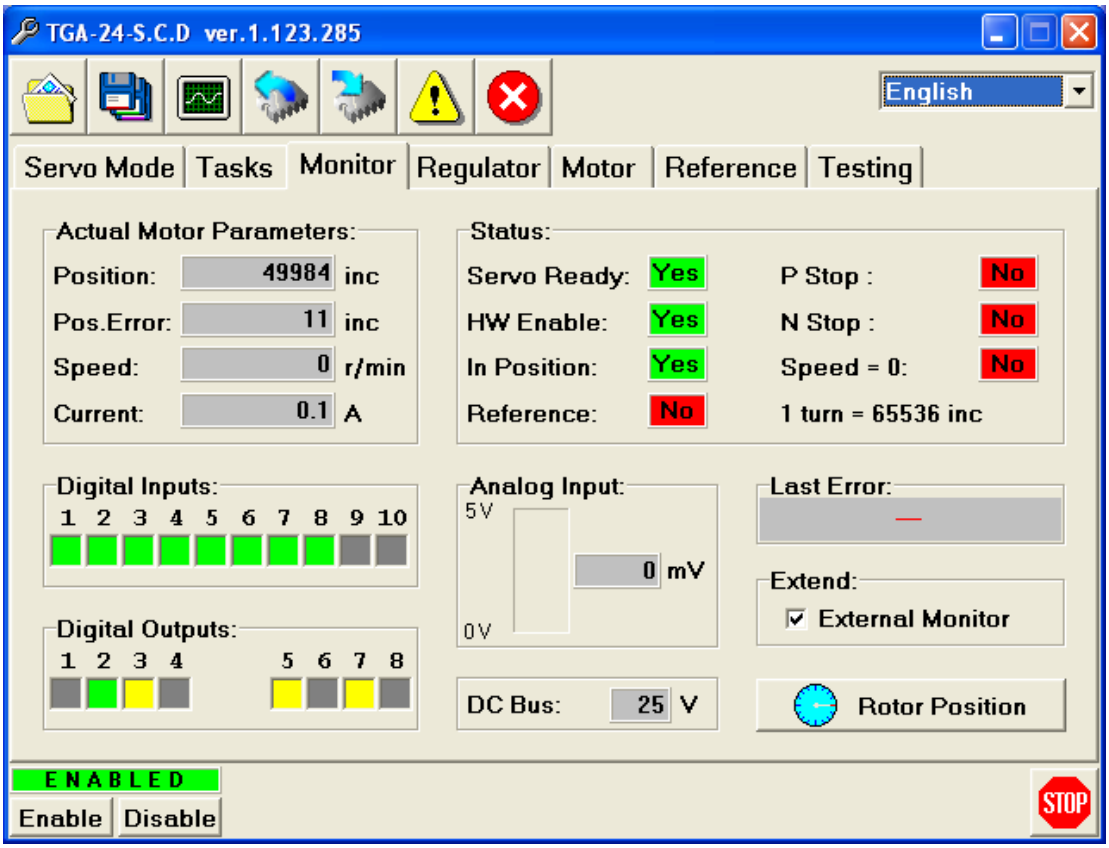
### 3.2.4 Control input
It is possible to chose digital inputs 4 to 8.

### 3.2.5 Virtual signal

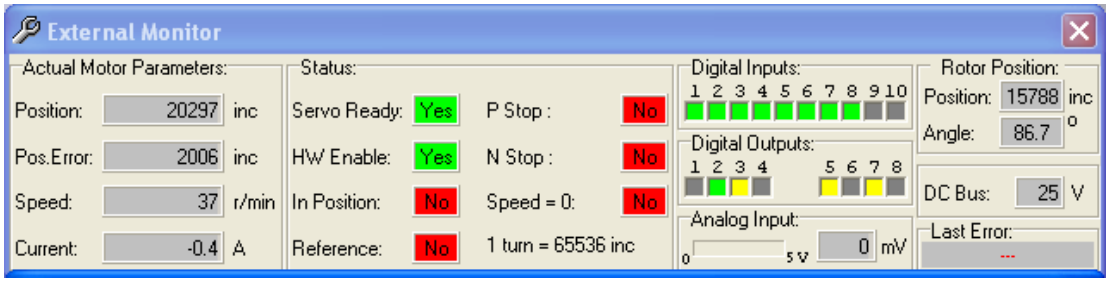
| | |
|---|---|
| Internal timer | The function is started by a edge of the internal timer. |
| In position | The function is started by internal signal In position.. |
| Speed = 0 | The function is started by internal signal Speed=0. |

### 3.2.6 Run Edge
Every function can be start by rising or falling edge of the control input or signal.

## 3.3 MENU Monitor




Menu Monitor allows monitoring status and parameters of a servo-controller (inputs, outputs, input voltage, faults) and a servomotor (position, position error, speed, current, faults). To see the value in other modes it is possible to enable External Monitor and External Monitor window is opened.

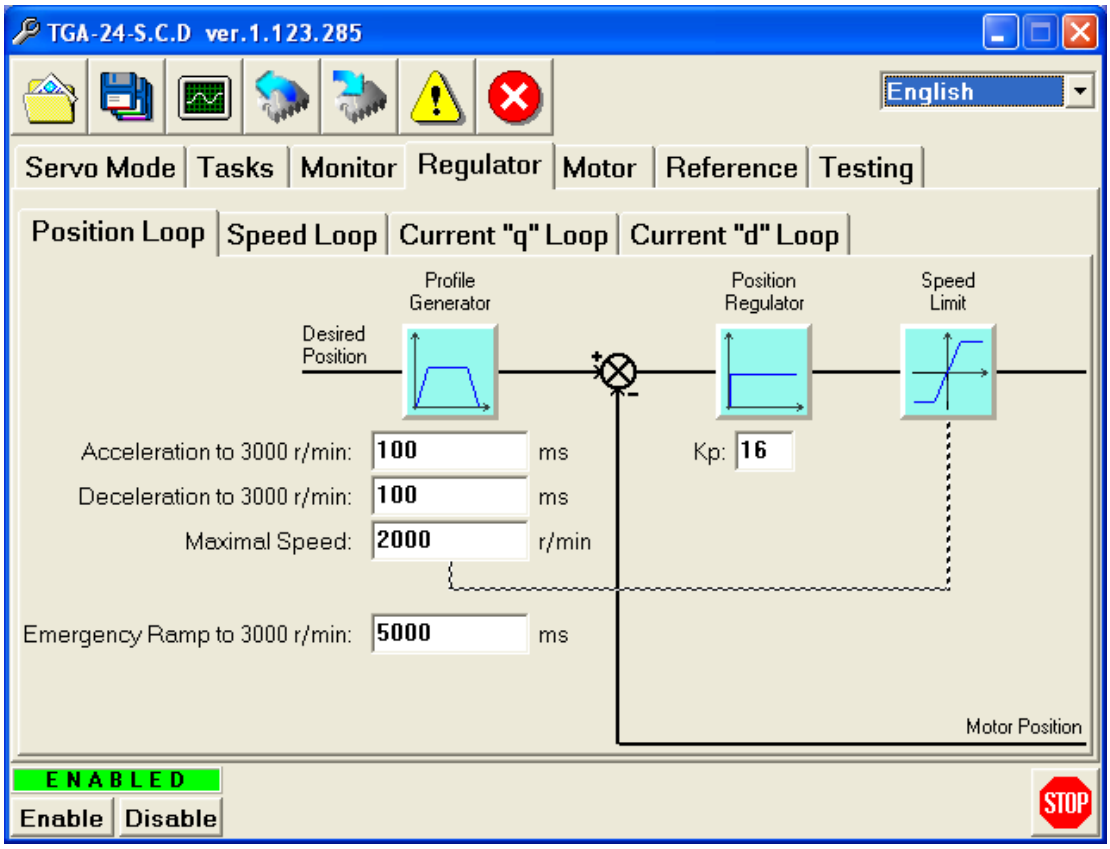


### 3.4 MENU Regulator

Parameters of the regulators are possible to set in the Menu. The TGA servo-controller has cascade regulation structure.

#### 3.4.1 Position loop

| | |
|---|---|
| Kp | Proportional gain of the position regulator. Higher value reduces position error. If the value is too high an oscillation of the servomotor can appeared. |
| Acceleration to 3000 r/min | Maximal allowed acceleration |
| Deceleration to 3000 r/min | Maximal allowed deceleration |
| Maximal speed | Maximal speed |
| Emergency Ramp | Acceleration time when N STOP and P STOP are happened |




#### 3.4.2 Speed Loop

| | |
|---|---|
| Kp | Proportional gain of the speed regulator |
| Ki | Integrating gain of the speed regulator |
| Imax | Maximal current Iq (Q- active- current component) |

#### 3.4.3 Current “q“ Loop

Regulator of Q-active current component.

| | |
|---|---|
| Kp | Proportional gain of the iq |
| Ki | Integrating gain of the iq |

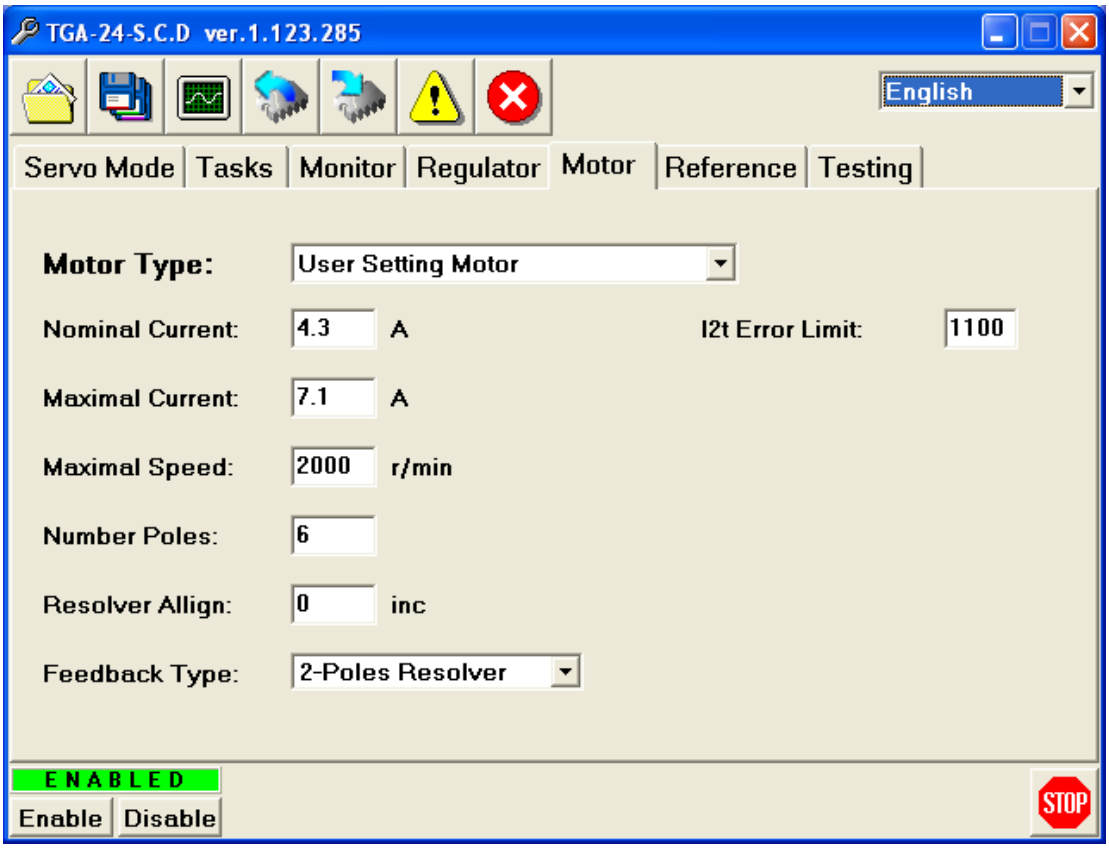
#### 3.4.4 Current “d“ Loop

Regulator of the D-reactive current component

| | |
|---|---|
| Kp | Proportional gain of the id |
| Ki | Integrating gain of the id |

### 3.5 MENU Motor

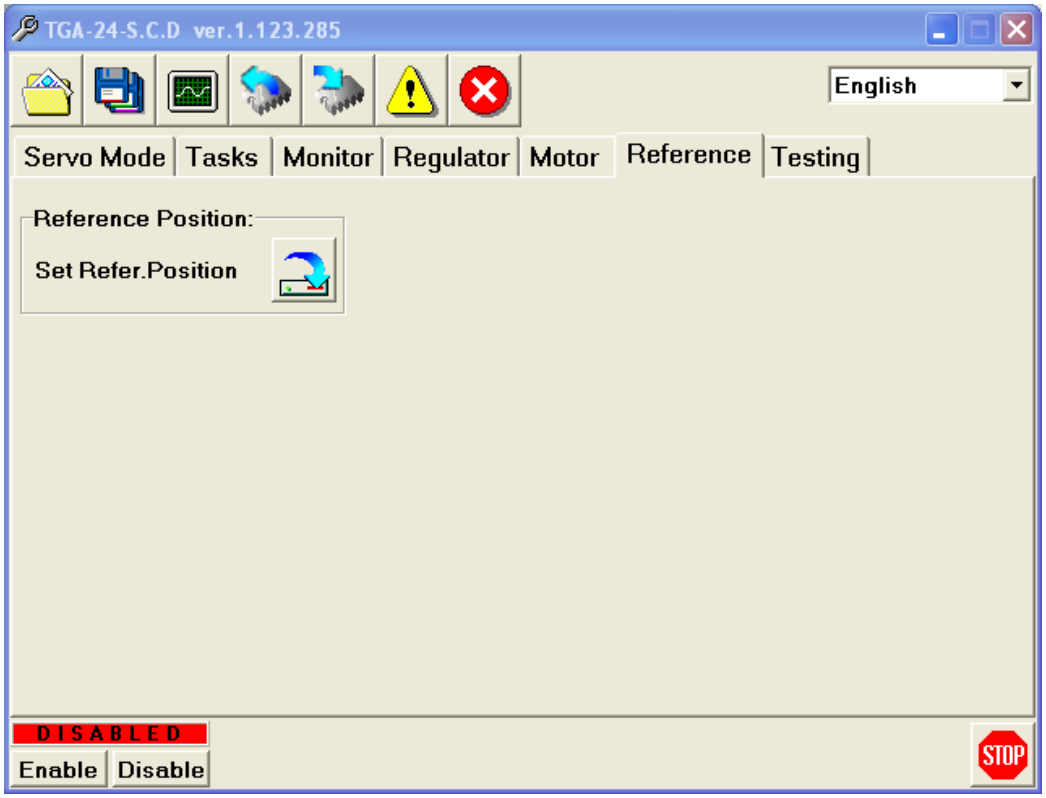
Menu “Motor” allows selecting pre-defined motor or setting parameters manually. When some pre-defined motor is chosen all parameters from Menu Motor and Regulator are set automatically.




| | |
|---|---|
| Nominal Current | Nominal current of the motor (r.m.s. value) |
| Maximal current | Maximal allowable current (r.m.s. value). |
| Maximal speed | Nominal speed |
| Number of Poles | Number of poles |
| Resolver Align | Setting of the commutation angle |
| Feedback Type | Only 2-poles resolver |
| I2t Error Limit | Setting of theI2t characteristic. Value 1100 allows 2 times nominal current for 5 s. |

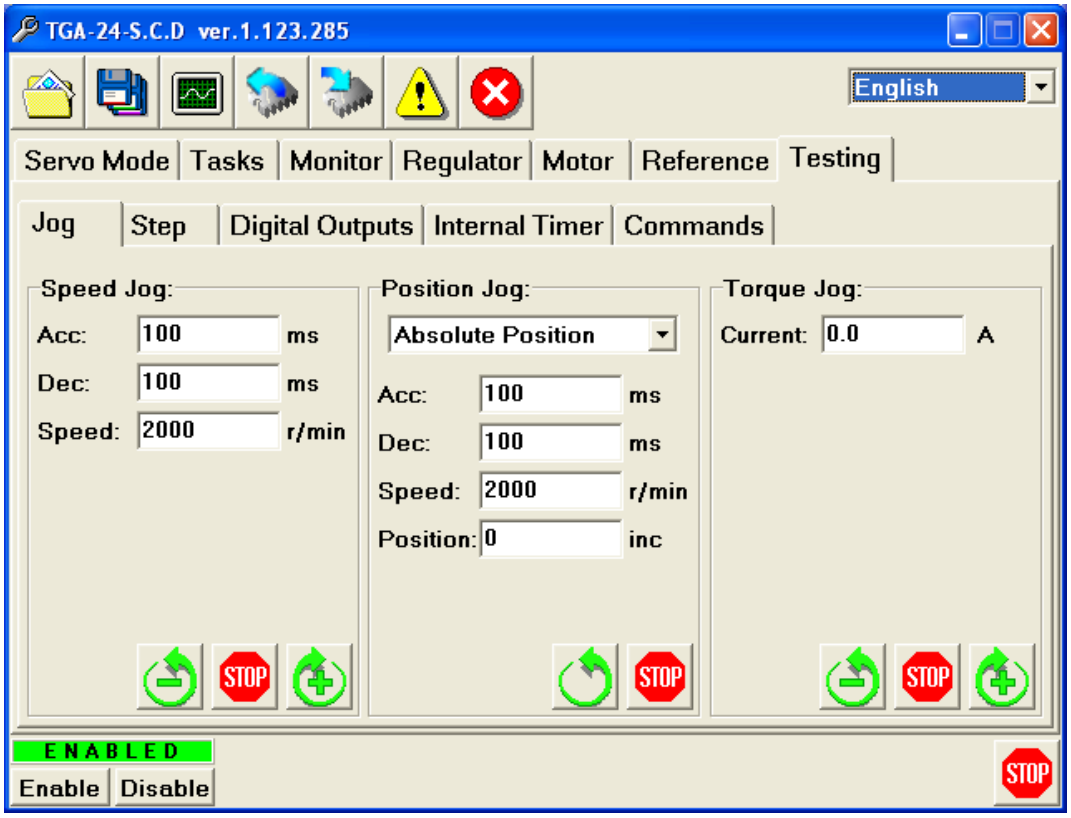
### 3.6 MENU Reference

Setting internal position counter to zero.




### 3.7 MENU Testing

Menu Testing allows testing your application through PC.



The modes is possible to test in **window Jog**:

Digital torque control
Digital speed control
Absolute Position
Relative Position
Incremental Position

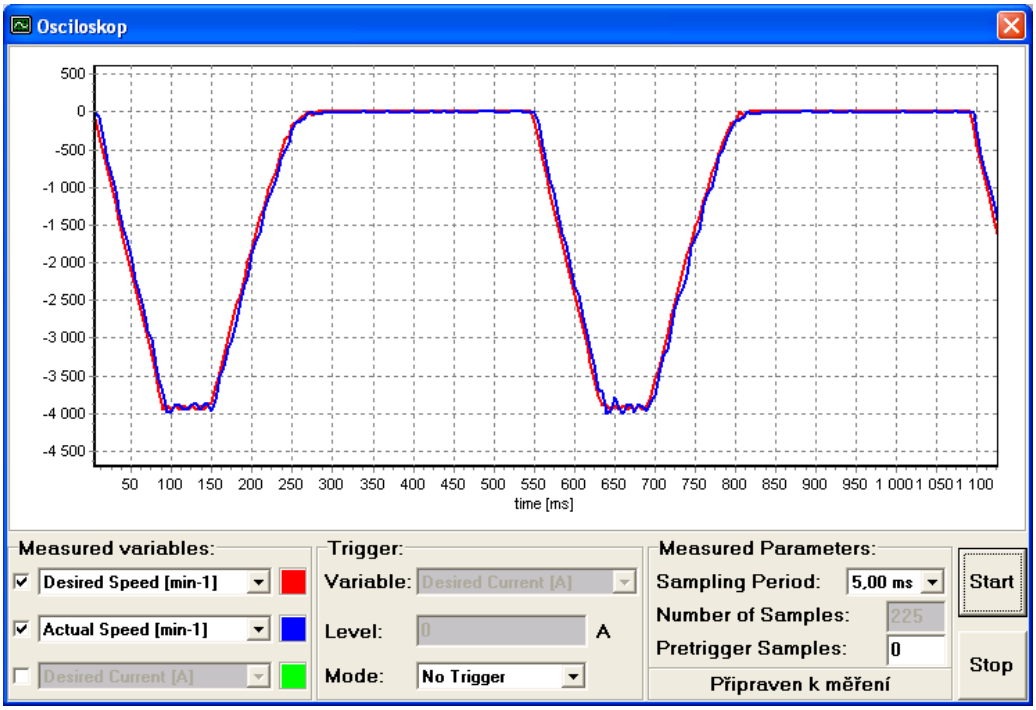
**Window "Step"** allows testing motion modes in step mode.

**Window "Digital outputs"** allows setting and resetting digital outputs.

**Window "Internal Timer"** allows starting and stopping internal timer. A drive will execute the function set in Menu **Tasks**.
**Commands** direct start of a predefined motion task 1-10.

### 3.8 OSCILLOSCOPE

Oscilloscope allows to graphical monitoring of the actual values of some parameters. Because of oscilloscope has not automatic scaling of each separate parameter it is recommended to monitor parameters with similar values (e.g. actual and desired speed). The zoom is activated by movement of the mouse to right-down and simultaneous pushing of the left switch.




| | |
|---|---|
| Desired Current (A) | Actual value of the desired current. |
| Desired Position (inc) | Actual value of the desired position |
| Desired Speed ($min^{-1}$) | Actual value of the desired speed |
| Actual Current „q" (A) | Actual value of the Q-active-current component |
| Actual Current „d" (A) | Actual value of the D-reactive-current component |
| Actual Position (inc) | Actual value of the position |
| Actual Speed ($min^{-1}$) | Actual speed |
| Position Error (inc) | Actual position error |
| Rotor Angle (inc) | Actual rotor position within one turn |
| DC Bus (V) | Actual value of the supply voltage |
| Analogue In (V) | Actual value of the analogue input |

## 3.9 FAULT DESCRIPTION

**Resolver error**
Check resolver cable, resolver on the motor.

**Overheating of the controller**
Temperature of the heat sink increased 70°C

**Over heating of the servo motor**
Temperature of the winding of the motor increased 140°C Check mechanism.

**Short circuit on power stage, cable or motor**
Check motor cable, winding of the motor.

**Over loading of the servomotor**
Check mechanism

**Position error**
Actual position is out of the allowable range. Check mechanism, power cable.

**Under voltage**
Voltage dropped under 15 VDC. Check power supply.

**Disable (if digital input DI1 is set)**
Input 1 is not activated.

# 4 Communication Protocol RS 232/485 description

The RS-232 protocol is a set of simple binary structures and conventions enabling a data/code exchange between a personal computer (PC) and a target controller board. It uses raw 8 bits, no parity, serial transfer at a standard speed (9600 bps, 19200, 38400, 57600bps by default).

The communication model is based on a master-slave relationship, where the PC sends a message with a command and its arguments, and the target responds immediately (within a specified time) with the operation status code and return data. The target never initiates communication; its responses are specified and always of a fixed (known) length (This is true on a logical level. On a link level, there is a replication of special start-of-message bytes, details of which follow).

## 4.1 Command Message

A command message is always sent from the PC to the target.

| start-of-message (1 BYTE) | command (1 BYTE) | data length (1BYTE) | data part (variable length) | checksum (1 BYTE) |
|---|---|---|---|---|

| | |
|---|---|
| start of message (SOM) | The special character defined as ASCII '+' code (0x2b) |
| command | A one byte command code (see bellow) |
| data length | The length of data part |
| data part | Variable length data |
| checksum | Two's complement checksum; computed by taking the two's complement of the sum of all bytes of a message after the SOM. |

The following code example calculates checksum for a message in standard format before it is transmitted:

```
typedef unsigned char BYTE;

struct {
        BYTE cmd;
        BYTE len;
        BYTE data[N];
        BYTE _space_for_checksum;
} message;

// prepare message
// ...

// calculate checksum
BYTE *p = &message;
BYTE sum = *p++;                    // cmd field
for(int i=0; i<=message.len; i++)
        sum += *p++;                // add len and data
// store checksum after last valid data byte
*p = 0x100 - sum;

// transmit SOM byte
// ....

// transmit message and checksum (replicating each occurence of SOM byte)
// ....
```

The start-of-message (SOM) character receives special treatment from the link protocol layer. When received, it should reset the receiver's state machine and initialise it for reception of new message. Since the data being transferred across an RS-232 line is in binary format, a byte with value equal to SOM may be contained in the message body, which could cause an undesirable reinitialization of the receiver. This is why each occurrence of SOM byte in length, data or checksum part of a message is signalled by duplicating this byte. On the other side, the receiver resets its state machine only when the SOM byte it receives is followed by a non-SOM byte. If the receiver receives two consecutive SOM bytes, it merges them to a single one.

## 4.2 Response Message

A response message is always sent from the target to the PC.

The format of response messages is shown in the following table:

| start-of-message (1 BYTE) | status code (1 BYTE) | data part (known length) | checksum (1 BYTE) |
|---|---|---|---|

| | |
|---|---|
| start of message | The special character defined as ASCII '+' code |
| status code | The one byte operation status code (see bellow) |
| variable length data | the length depends on status code value. An error response message (status MSB set) carries no data; a success response message carries known data to the PC (the data length is predetermined) |
| checksum | Two's complement checksum; computed by taking the two's complement of the sum of all bytes of a message after the SOM |

## 4.3 Commands description:

### 4.3.1 To read a Word

| | | |
|---|---|---|
| Start of the message (1byte): | 2BH | |
| Command: | D1H | |
| Address (2 byte): | | |
| Check Sum: | | |
| | | |
| Answer (5 Bytes): | | |
| Start of the message (1byte): | 2BH | |
| Status (1 Byte): | 00H | No error |
| | 81H | Unknown command code |
| | 82H | Command checksum error |
| | 83H | Command is too long |
| Data (2 Bytes): | | |
| Check Sum: | | |

### 4.3.2 To read a DWord

| | | |
|---|---|---|
| Start of the message (1Byte): | 2BH | |
| Command: | D2H | |
| Address (2 Byte): | | |
| Check sum: | | |
| | | |
| **Answer (7 Bytes):** | | |
| Start of the message (1Byte): | 2BH | |
| Status (1 Byte): | 00H | No error |
| | 81H | Unknown command code |
| | 82H | Command checksum error |
| | 83H | Command is too long |
| Data (4 Byte): | | |
| Check sum: | | |

### 4.3.3 To write a Word

| | |
|---|---|
| Start of the message (1Byte): | 2BH |
| Command: | 02H |
| Number of the transferring Bytes: | 05H |
| Number of the transferring Dates: | 02H |
| Address (2 Byte): | |
| Dates (2 Byte): | |
| Check sum: | |

*Example:*
To write to address 01A6H a Word it has to be send:
2B,02,05,02,A6,01,LB,HB,CS

**Answer (3 Bytes):**

| | | |
|---|---|---|
| Start of the message (1Byte): | 2BH | |
| Status (1 Byte): | 00H | No error |
| | 81H | Unknown command code |
| | 82H | Command checksum error |
| | 83H | Command is too long |
| Check sum: | | |

### 4.3.4 To write a DWord

| | |
|---|---|
| Start of the message (1Byte): | 2BH |
| Command: | 02H |
| Number of the transferring Bytes: | 07H |
| Number of the transferring Dates: | 04H |
| Address (2 Byte): | |
| Dates (4 Bytes): | |
| Check sum: | |

**Answer (3 Bytes):**

| | | |
|---|---|---|
| Start of the message (1Byte): | 2BH | |
| Status (1 Byte): | 00H | No error |
| | 81H | Unknown command code |
| | 82H | Command checksum error |
| | 83H | Command is too long |
| Check sum: | | |

*Note:*
List of the variables and their addresses is enclosed.

# 5 CAN bus-communication profile for TGA-24-9/20

Main advantage of the TGA servo-controller is its simple installation and using. So the CAN protocol was developed according to the idea. You do not need to implement any CANopen driver to your PC or PLC, but if you already have it you can use the driver as well.

## 5.1 Control and parameterisation

There is implemented only one communication protocol for control and parameterisation.

A PLC or PC is always master. The PLC or PC sends the message and the TGA servo-controller answering within 1-2 ms (synchronous transmission).

### 5.1.1 Message description sent by master (PLC, PC)

**DLC=7**

| ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 0x600+node | ControlByte | Address | | Data | | | | Data length |

**ID** – identifier of the message in hex code
**node**- address (id) of the servo-controller 1-63 (0x01-0x3F)

**Byte0** Control Byte = 9 write parameter
Control Byte = 8 read parameter

**Byte1 + 2** Address of the parameter (see the list of the parameters)

**Byte3 - 6** Data - data of the parameter (see the list of the parameters)

**Byte7** Length of the data 1 – single-word data (16 bits)
2 – double words data (32 bits)

*Note:*
If you want to read a parameter (**Byte0**=8) the **Byte3-Byte6** should be 0.

### 5.1.2 Message description sent by TGA (slave)

**DLC=7**

| ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 0x580+node | ControlByte | Address | | Data | | | | Data length |

**ID** – identifier of the message in hex code
**node** -address (id) of the servo-controller 1-63 (0x01-0x3F)

**Byte0** ControlByte=9 write parameter
ControlByte=8 read parameter

**Byte1+2** Address of the parameter (see parameters list)

**Byte3-6** Data- data of the parameter (see parameter list)

**Byte7** Length of the data 1 – single word data (16 bits)
2 – double words data (32 bits)

*Note:*
In case of the writing parameters (**Byte0=9**) the TGA answering by the same values in **Byte0-Byte7** as received.

In case of the reading parameters (**Byte0=8**) the TGA answering by the same values in **Byte0-Byte2** and **Byte7**. The **Byte3-Byte6** are filled by actual values.

*Example:*

*1/ Write desired speed 600 rpm. to the servo-controller with address=3*

| | |
|---|---|
| Speed calculation: | 600 rpm. = 171798691units = 0xA3D70A3 |
| ID | 0x603 |
| Address of the desired speed | 0x194 |
| Data | 0A 3D 70 A3 |
| Data length | 0x2 |

Master (PLC,PC)

| ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 603 | 09 | 01 | 94 | 0A | 3D | 70 | A3 | 02 |

Answer of the TGA

| ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 583 | 09 | 01 | 94 | 0A | 3D | 70 | A3 | 02 |

*2/ Read actual position from the servo-controller with address=16*

| | |
|---|---|
| Actual servo position is -10000 increments | |
| ID | 0x60F |
| Address of the actual position: | 0x196 |
| Data | -10000 incr. = 0xFFFFD8F0 |
| Data length | 0x2 |

Master (PLC,PC)

| ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 60F | 08 | 01 | 96 | 00 | 00 | 00 | 00 | 02 |

Answer of the TGA

| ID | 5.1.2.1 Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 58F | 08 | 01 | 96 | FF | FF | D8 | F0 | 02 |

## 5.2 Remote frame (monitoring):

### 5.2.1 Message description sent by master (PLC, PC)

**DLC=0**

| ID |
|---|
| 0x180+node |

**ID** – identifier of the message in hex code
**node**- address (id) of the servo-controller 1-63 (0x01-0x3F)

### 5.2.2 Message description sent by TGA (slave)

**DLC=7**

| ID | byte 0 | byte 1 | byte 2 | byte 3 | byte 4 | byte 5 | byte 6 | byte 7 |
|---|---|---|---|---|---|---|---|---|
| 0x180+node | Status | Actual position | | | | Dig. inputs | | Dig. outputs |

**ID** – identifier of the message in hex code
**node**- address (id) of the servo-controller (0x01 – 0x3F)

**Byte0** Status of the servo-controller

| |
|---|
| *bit 0 = Enable OK*<br>*bit 1 = In Position*<br>*bit 2 = Trajectory CAN Control is activated*<br>*bit 3 = Speed=0*<br>*bit 4 = Emergency stop 1*<br>*bit 5 = Emergency stop 2*<br>*bit 6 = Fault* |

**Byte1-4** Actual rotor position range $-2^{31}$ to $(2^{31}-1)$ (1 turn = 65536 increments)

**Byte5-6** Actual digital inputs state (negative value)

**Byte5**: *bit0 = in1*
. .
*bit7 = in8*

**Byte6** reserved

**Byte7**

Actual digital outputs state
**Byte7**: *bit0 = out1*
. .
*bit3 = out4*

# 6 Aplication notes

## 6.1 Speed Mode

- Enable the power stage of the servo-controller

  Write *2 Byte* decimal value *2* on address *$1ED*

- Switch mode

  Speed mode $*2002*

  Write *4 Byte* hexadecimal value $*2002* on address *$1EF*

- Set acceleration

  $a = 50\ n.s^{-2}$ ; *50 . 178957 = 8947850*

  Write *4 Byte* decimal value *8947850* on address *$1D4*

- Set deceleration

  $a = 50\ n.s^{-2}$ ; *50 . 178957 = 8947850*

  Write *4 Byte* decimal value *8947850* on address *$1D6*

- Set maximal speed

  $v_{max}$ = 3000 $min^{-1}$ = 50 $s^{-1}$ ; 50 . 17179869,184 = 858993459

  Write *4 Byte* decimal value 858993459 on address *$1D2*

- Set desired speed

  v = -1500 $min^{-1}$ = -25 $s^{-1}$ ; -25 . 17179869,184 = -429496730

  Write 4 *Byte* signed decimal value -429496730 on address *$194*

## 6.2 Position mode

- Enable the power stage of the servo-controller

  Write *2 Byte* decimal value *2* on address *$1ED*

- Switch mode

  Absolut position mode $4004

  Write *4 Byte* hexadecimal value $*4004* on address *$1EF*

- Set acceleration

  $a = 50\ n.s^{-2}$ ; *50 . 178957 = 8947850*

  Write *4 Byte* decimal value *8947850* on address *$1D4*

- Set deceleration

  *a = 50 n.s$^{-2}$ ; 50 . 178957 = 8947850*

  Write *4 Byte* decimal value *8947850* on address *$1D6*

- Set maximal speed

  $v_{max}$ = 3000 min$^{-1}$ = 50 s$^{-1}$ ; 50 . 17179869,184 = 858993459

  Write *4 Byte* decimal value 858993459 on address *$1D2*

- Set desired position

  Desired position 655360 inc ( 1r. = 65536 inc )

  Write *4 Byte* signed decimal value 655360 on address *$192*



| | |
|---|---|
| Company name | **TG Drives, s.r.o.** |
| Legal form | limited liability company |
| Office address | Olomoucká ev.č. 84, CZ - 627 00 Brno |
| Invoicing address | Jeneweinova 37, CZ - 617 00 Brno |
| IČO | 60738821 |
| DIČ | CZ60738821 |
| Phone | +420548141811 |
| Fax | +420548141890 |
| E-mail | info@tgdrives.cz |
| Web | www.tgdrives.com |