Gertech AUTOMATION

GMA-CANopen Series Multi-turn Absolute Encoder

Features

- Clock structure, Cooper-Gear-Type Multi-turn Absolute Encoder;
- Permanent magnet and magnetic induction element are adopted, and through magnetoelectric conversion, converting the angular displacement of the shaft into electrical pulse signal output.
- Stable and High Reliable With a long working life;
- Housing Diameter:58mm;
- Hollow Shaft Diameter:12mm;
- Resolution: Turns:Max16bit, Single Turn:Max. 16bit, Total Max. 29bits
- CANopen Interface;
- Supply Voltage: 8-30v;
- Widely used in various fields of automatic control and measurement system, such as machinery manufacturing, shipping, textile, printing, aviation, military industry Testing machine, elevator, etc.
- Vibration-resistant, corrosion-resistant, pollution-resistant;

Technical Parameter:

Item	Technical Parameter	Signal	Colour
Size	Housing Dia.:58mm; Shaft Dia.:10mm	Dc 8-30v	Red
Output Interface/ Method	CANopen	0v	Black
Resolution	Turns:Max16bit, Single Turn:Max. 16bit, Total Max. 29bits	CANopen H	Blue
Supply Voltage	8-30v;	CANopen L	White
Max. Frequency Response	350kHz		
Max. Rotary Speed	5000rpm		
Start Torque	9.8x 10^{-4} N•M		
Max. Shaft Loading	Axial: 29.4N, Radial:19.6N;		
Weight	0.2kg		
Grade of Protection	IP 67		
Working Temperature	−40~85℃		

***Customer can select their preferable the Output method, Supply Voltage and Required Resolution(Turns/PPR);**

CANopen Interface

The encoder follows the "encoder device line rule class2", and is generally used as a slave device. For information not covered in this manual,please refer to the relevant sections of the documents "CIA Standard Specification 301" and "CIA Standard Specification 406" (these two specifications can be obtained from www.can-cia.org).

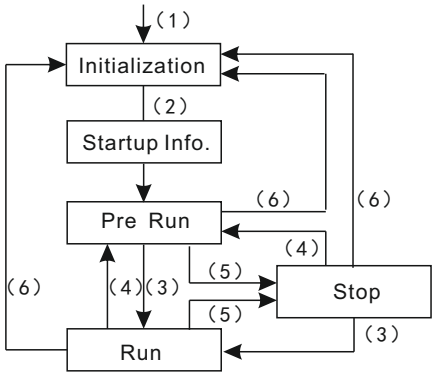
EDS Document

EDS files are provided to customers together with CANopen encoder. Please install EDS files on the main controller of CANopen before using CANopen encoder.

State Device

The CANopen device can be applied in different working states. By sending specific NMT messages , it can switch between different working states. The status diagram is as follows:

Item	Description
(1)	Power On
(2)	Initialization Done, start information sent automatically
(3)	NMT message "start remote node"
(4)	NMT message "pre run"
(5)	NMT message "close remote node"
(6)	N MT message "reset node" or "reset communication"



GA-CANopen Series Multi-turn Absolute Encoder

Initialization

This is the state that CANopen device enters for the first time after power on or hardware reset. After reading the parameter information stored in EPROM and completing the basic device initialization, CANopen device (encoder) automatically sends the startup information to the main controller to enter the "pre operation" state.

Pre Run

In this state, it can communicate through SDO. Because PDO does not exist, it cannot communicate through PDO. The configuration and parameter configuration of PDO can be completed by configuration program. The encoder can be put into operation directly by sending "start remote node".

Run

In this state, all communication objects are available. The slave device can send process data through PDO according to the parameter settings in the object dictionary. The main controller can access the object dictionary through PDO. The main controller sends "pre operation" message to make it enter "pre operation" state.

Stop

In this state, the slave device is forced to stop all communication (except the monitoring node). Communication through SDO through PDO is also. By sending specific NMT message, the master controller can directly enter into pre run or run state from the encoder.

Communication Object

There are four kinds of communication messages:
-Network management NMT: the NMT master controller controls the NMT status of the NMT slave device.
-Process data object PDO: used to transfer real-time data.-Service data object SDO: object dictionary for direct access to CANopen devices.
-Special function object:
-Synchronous transmission (sync): provides a basic network synchronization mechanism. With this service,main controller can send real-time data.
-Emergency: use this object whenever an error event occurs.
-Node guard: used to view the operation status of the slave device.

Relationship between device state and communication object:

	Initialization	Pre Run	Run	Stop
NMT		X	X	X
PDO			X	
SDO		X	X	
Sync			X	
Emerg	X	X	X	
Boot-up				
Nodeg		X	X	X

Predefined connection instructions

Main Controller→Encodrer(Broadcast)		
Communication Object COB Type	**Function Code (Binary)**	**COB-ID(Hexadecimal)**
NMT	0000	001
Sync	0001	080
Main Controller→Encodrer(Point to Point)		
Emerg	0001	081-0FF
PD01(Send)	0011	181-1FF
PD02(Send)	0101	281-2FF
SDO(Send)	1011	581-5FF
SDO(Receive)	1110	601-67F
Node monitoring		701-77F

〞Communication object type" (send / receive) is from the perspective of device (encoder).
The start message uses the cob-id of the node monitoring object.

GA-CANopen Series Multi-turn Absolute Encoder

NMT Object

NMT Structure

COB-ID(11Bit)		2 Bytes CAN Data	
Function Code	Node ID	Order	Encoder ID
0000	0	NMD Function	Encoder ID

If the encoder ID is 00h, NMT message is sent to all network accessing nodes.

NMT Function

Order(Hexadecimal)	NMT function	Node State
01	Start Remote Node	Run
02	Stop Remote Node	Stop
80	Enter Pre Run	Pre Run
81	Reset Node	Pre Run
82	Reset Communaction	Pre Run

Start Boot-up Object

Boot-up message structure:

COD-ID(Hexadecimal)	1 Byte CAN Data
700+Node ID	00

POD Object

PDO(Send) message consists of four CAN data bytes, to transmit the position value of the encoder

POD Structure

Identification		4 Bytes CAN Data			
COD-ID(Hexadecimal)		Byte 0	Byte 1	Byte 2	Byte 3
Function Code	Node ID	2^{7}-2^{0}	2^{15}-2^{8}	2^{23}-2^{16}	2^{31}-2^{24}
		Low Byte	…	…	High Byte

The three types PDO are defined as follows:

PDO1 cyclic mode: asynchronous transmission. The absolute rotary encoder automatically sends current process data cyclically without querying the host. The cycle time can be set programmatically. The value is between 1 and 65536. The unit is ms. (see "cycle time: 6200h object"). Circular mode can be used (or disabled) by setting the highest position of cob-id used by PDO1 (object 1800h, sub item 1) to "0" ("1").

PDO2 synchronous mode: synchronous transmission. Synchronization refers to sending PDO after main controller synchronizaton.

Synchronous message is a cob with a highest priority sent by the master controller to all slave devices. After receiving the synchronous message, the encoder will send back the process data. Each slave device transmits data according to its own node ID. If the encoder does not respond to all synchr--onization commands,

it should be set to respond after “n” units synchronization commands. The value of PDO2 “n” can be set by sub item 2 of object 1801h.

The value of PDO3, “n”can be set through the sub item 2 of object 1802h.

The synchronization mode can be used (or disabled) by setting the highest position of cob-id used by PDO (object 1801h or 1802h, sub item 1) to "0" ("1").

Note:Multiple transport modes can coexist.

SDO Object

SDO messages are used to query or change the parameters of the encoder, which are included in the object dictionary. The maximum of CAN data is 4 bytes, and the other 4 bytes are used for command, index, sub index and other fields. When the controller sends an SDO message to the encoder, the encoder will send the response information to the main controller (If error happens, error will be alarmed).

GA-CANopen Series Multi-turn Absolute Encoder

SDO Structure

Identifier		4 Bytes CAN Data				1-4 Bytes CAN Data			
Identifier Hex.		0	1	2	3	4	5	6	7
Function Code	Node ID	Order	Indexes		Subindex	Process Data			
		1 Byte	Low Byte	High Byte	1 Byte	Low Byte	...	...	High Byte

Order

The content of command byte is transmitted through can network in the form of message.
There are three message forms:
-Settings: send configuration parameters to the device;
-Request: the main controller is used to read the data of the device;
-Alarm: the encoder is used to send error information to the main controller (for example, index does not exist, parameter is invalid, etc.)

Object Dictionary

Each object is represented as follows:
Index - sub index object name [data type, property]
-Indexes and subindexes are identified in hexadecimal.
-Properties: RO = read-only, RW = read-write.

-Unsigned 16 Data Type:

Process Data Byte	
Byte 4	Byte 5
Low Byte	High Byte

-Unsigned 32 Data Type:

Process Data Byte			
Byte 4	Byte 5	Byte 6	Byte 7
Low Byte	...	...	High Byte

Standard Object(DS301)

Index - Subindex	Object Name	[Data Type, Property]
1000-00	Device Type Default value: 0001 0196h=Single Turn Absolute Encoder, DS 406 0002 0196h=Multi-turn Absolute Encoder, DS 406	[Unsigned32,ro]
1001-00	Error register If a bit of the register is set to "1", its corresponding error has occurred. Default value: 00h	[Unsigned 8, ro]
1003 -00 -01 -02 . . . -04	Predefined error fields This object saves errors that the device has generated. Number of errors [unsigned 8, RW] Write 00h to clear the error history. Last error An earlier error	 [unsigned 32, RO] [unsigned 32, RO]
1005-00	Cob ID synchronous message Default value: 0000 0080h	[unsigned 32, RW]
1008-00	Equipment manufacturer name Default: "ADK"	[string, RO]
1009-00	Hardware version	[string, RO]
100A-00	Software version	[string, RO]
100C-00	Monitoring time Default: 03e8h (MS)	[unsigned 16, RW]
100D-00	Life time factor Default value: 05H "node life time" = "0bj_100c" * "0bj_100d". When "node life time" is not equal to 0, "node monitoring" is enabled. If the slave device is not monitored in the service life, the "service life monitoring event" will be triggered. The red LED indicates the monitoring error, and the content of 1001h and 1003h objects will be updated. In order to eliminate errors, "reset node" message should be sent.	[unsigned 8, RW]
1010-01	Storage parameter This object holds all parameters to non-volatile memory. The signal written is "save". **Controller → encoder (write)**	[unsigned 32, RW]

COB-ID	Order	Index		Sub Index	Data Byte			
600+ID	23	10	10	01	73	61	76	65

GA-CANopen Series Multi-turn Absolute Encoder

Index - Subindex	Object Name [Data Type, Property]
	Encoder → controller (confirm) COB-ID: 580+ID; Order: 60; Index: 10 10; Sub Index: 01; Data Byte: 00 00 00 00
1011-01	Restore default parameter [unsigned 32, RW] This object is used to restore the default values of all parameters. When the written signal is "load", after setting reset, the restored default value takes effect. **Controller → Encoder (write)** COB-ID: 580+ID; Order: 23; Index: 11 10; Sub Index: 01; Data Byte: 6F 6C 61 64 **Encoder → Controller (confirm)** COB-ID: 580+ID; Order: 60; Index: 11 10; Sub Index: 01; Data Byte: 00 00 00 00 **Controller → Encoder (Reset Node)**: COB-ID: 000; Order: 81; Slave Device ID: ID **Encoder → Controller (Boot up)**: COB-ID: 700+ID; Order: 00 In order to save the default values, the function of "store parameters" need to be performanced (see object 1010h).
1014-00	EMCY COB-ID [Unsigned32,rw] Default value 80h + node ID This object defines the cob-id of the emcy writing service
1018 -01 -02 -03	Identifying objects Sales firm [unsigned 32, RO] Manufacturer [unsigned 32, RO] Revision number [unsigned 32, RO]
1800 -01 -02	PD 01 Transmission Parameter this type of PDO asynchronously transmits the position value of the encoder. For the setting of the cycle timer, see object 6200h COB ID of PDO1 [unsigned 32, RW] Default value: 4000 0180h + node ID (COB ID without RTR) Transmission type [unsigned 8, RW] Default: FEH (asynchronous transfer)
1801 -01 -02	PDO2 transmission parameter synchronously transmits the position value of encoder. Cod ID of PDO2 [unsigned 32, RW] Default value: 4000 0280h + node ID (COB ID without RTR) Transmission type [unsigned 8, RW] Default value: 01h (synchronous transmission) For the n-value that requires n synchronous signals, it can be set in sub index 2 of object 1801h.
1802 -01 -02	PDO3 transmission parameter synchronously transmits the position value of encoder. Cob ID of PDO3 [unsigned 32, RW] Default value: C000 0380h + node ID (PDO3 disabled, no RTR) Transmission type [unsigned 8, RW] Default value: 01h (synchronous transmission) For the n value of N response signals, it can be set in sub index 2 of object 1802h. Note: It can be used (disabled) by setting the highest bit of cob ID (object 180xh, sub index 1) of PDO to "0" (or "1").
1A01-01	PDO1 mapping parameter [unsigned 32, RW] Default value: 6004 0020h This object follows ds406 and contains the position value of the encoder.
1A01-01	PDO2 mapping parameter [unsigned 32, RW] See object 1a00h, sub index 1.
1A01-01	PDO3 mapping parameter [unsigned 32, RW] See object 1a00h, sub index 1.

Manufacturer Related Objects

Index - Subindex	Object Name [Data Type, Property]
2104-00	Min. Value of Limit switch [unsigned 32, RW] Default value: 0000 0010h If the position value is less than the value specified by the object 2104h, 12 bits of the object 6500h are set to "1". This function can be turned on by setting the 12 bits of the object 6000h to "1".

<table>
<tr><th>Index - Subindex</th><th>Object Name [Data Type, Property]</th></tr>
<tr><td>2105-00</td><td>Max Value of Limit Switch [unsigned 32, RW]
Default value: 003F FFF0H
If the position value is greater than the value specified by the object 2105h, 13 bits of the object 6500h are set to "1".
This function can be turned on by setting the 13 bits of the object 6000h to "1".</td></tr>
<tr><td>3000-00</td><td>Baud rate [unsigned 8, RW]
This object defines the bit rate of the device, as shown in the following table
<table>
<tr><th>Data</th><th>Baud Rate</th></tr>
<tr><td>00h</td><td>20Kb/s</td></tr>
<tr><td>01h</td><td>50Kb/s</td></tr>
<tr><td>02h</td><td>100Kb/s</td></tr>
<tr><td>03h</td><td>125Kb/s</td></tr>
<tr><td>04h</td><td>250Kb/s</td></tr>
<tr><td>05h</td><td>500Kb/s</td></tr>
<tr><td>06h</td><td>800Kb/s</td></tr>
<tr><td>07h</td><td>1000Kb/s</td></tr>
</table>
To change the baud rate:
set the object 3000h,
then send the command "reset node"
(or "reset communication"), and finally store the parameters.
Controller → Encoder
<table>
<tr><th>COB-ID</th><th>Order</th><th colspan="2">Index</th><th>Sub Index</th><th>Data</th></tr>
<tr><td>600+ID</td><td>2F</td><td>00</td><td>30</td><td>00</td><td>See The List 00 00 00</td></tr>
</table>
Encoder → Controller (confirm)
<table>
<tr><th>COB-ID</th><th>Order</th><th colspan="2">Index</th><th>Sub Index</th><th>Data</th></tr>
<tr><td>580+ID</td><td>60</td><td>00</td><td>30</td><td>00</td><td>00 00 00 00</td></tr>
</table>
Controller → Encoder (Reset Node)
<table>
<tr><th>COB-ID</th><th>Order</th><th>Slave Device ID</th></tr>
<tr><td>000</td><td>81</td><td>ID</td></tr>
</table>
Encoder → Controller (Boot up)
<table>
<tr><th>COB-ID</th><th>Order</th></tr>
<tr><td>700+ID</td><td>00</td></tr>
</table>
The stored the parameters (see object 1010h).
If it cannot be stored, the original baud rate will be used after power on again.</td></tr>
<tr><td>3001-01</td><td>Node ID [unsigned 8, RW] this object defines the node identifier of the device.
Default value: 01h
The steps to change the node address are: set the object for 3001h, send the command "reset node",
and store the parameters.
Controller → Encoder
<table>
<tr><th>COB-ID</th><th>Order</th><th colspan="2">Index</th><th>Sub Index</th><th>Data</th></tr>
<tr><td>600+ID</td><td>2F</td><td>01</td><td>30</td><td>00</td><td>New Node ID 00 00 00</td></tr>
</table>
Encoder → Controller (confirm)
<table>
<tr><th>COB-ID</th><th>Order</th><th colspan="2">Index</th><th>Sub Index</th><th>Data</th></tr>
<tr><td>580+ID</td><td>60</td><td>01</td><td>30</td><td>00</td><td>00 00 00 00</td></tr>
</table>
Controller → Encoder (Reset Node)
<table>
<tr><th>COB-ID</th><th>Order</th><th>Slave Device ID</th></tr>
<tr><td>000</td><td>81</td><td>Original ID</td></tr>
</table>
Encoder → Controller (Boot up)
<table>
<tr><th>COB-ID</th><th>Order</th></tr>
<tr><td>700+ID</td><td>00</td></tr>
</table>
The stored parameters (see object 1010h).
If it cannot be stored, the original node ID will be used after power on again</td></tr>
</table>

Objects specified in the equipment regulations (DS 406)

<table>
<tr><th>Index - Subindex</th><th>Object Name [Data Type, Property]</th></tr>
<tr><td>6000-00</td><td>Running Parameters
<table>
<tr><th>Bit</th><th>Function</th><th>Bit=0</th><th>Bit=1</th></tr>
<tr><td>0</td><td>Encoder counting direction</td><td>CW</td><td>CCW</td></tr>
<tr><td>1</td><td colspan="3">N/A</td></tr>
<tr><td>2</td><td>Zoom function</td><td>Prohibit</td><td>Usable</td></tr>
<tr><td>3-11</td><td colspan="3">N/A</td></tr>
<tr><td>12</td><td>Min. Value of Position</td><td>Prohibit</td><td>Usable</td></tr>
<tr><td>13</td><td>Min. Value of Position</td><td>Prohibit</td><td>Usable</td></tr>
<tr><td>14-15</td><td colspan="3">N/A</td></tr>
</table>
Default value: 0000H
-The encoder counting direction defines whether the counting value increases or decreases when the rotation axis rotates clockwise or anticlockwise from the encoder axis.
-Zoom function: if this function is disabled, the physical resolution is used (see objects 6501h and 6502h);
if this function is used, the transmitted position value has the following relationship with the actual position value
posTx=obj_6001 Real Pos obj_6002/Obj_6501</td></tr>
<tr><td>6001-00</td><td>Resolution per revolution [unsigned 32, RW]
This object determines the number of steps for each revolution of the encoder. I
f the second digit of 6000h is "1",then the object is enabled.
To prevent counting errors, make sure that the result of obj_6501 is an integer.
obj_6001</td></tr>
<tr><td>6002-00</td><td>Total measurement range [unsigned 32, RW]
This object determines the total number of steps in the measurement range.
If the second position of 6000h is "1", the object is used.
Give an example:
The "total hardware resolution" of the multi turn encoder is 16777216, and the 6501h value of the object is 4096,
If the value of object 6001h is 2048, then obj_6002 = (16777216 x 2048)/4096 = 8388608;</td></tr>
<tr><td>6003-00</td><td>Preset value
The preset value is the set position value, which will be reached when the encoder shaft rotates to a certain angle.
To prevent operational errors, the default value must not exceed "total hardware resolution.".</td></tr>
<tr><td>6004-00</td><td>Position value [unsigned 32, RO]
This object defines the position output values of the communication objects 1800h, 1801h and 1802h.</td></tr>
<tr><td>6200-00</td><td>Cycle time [unsigned 16, RW]
Default value: 0064h (100ms).
Cycle timer is used in asynchronous communication to adjust the cycle interval of pd01 (object 1800h) transmission.</td></tr>
<tr><td>6500-00</td><td>Running Status [Unsigned16, ro]
<table>
<tr><th>Bit</th><th>Function</th><th>Bit=0</th><th>Bit=1</th></tr>
<tr><td>0</td><td>Encoder counting direction</td><td>CW</td><td>CCW</td></tr>
<tr><td>1</td><td colspan="3">N/A</td></tr>
<tr><td>2</td><td>Zoom function</td><td>Prohibit</td><td>Usable</td></tr>
<tr><td>3-11</td><td colspan="3">N/A</td></tr>
<tr><td>12</td><td>Less than the minimum value or not (object 2104h)</td><td>Prohibit</td><td>Usable</td></tr>
<tr><td>13</td><td>Less than the minimum value or not (object 2104h)</td><td>Prohibit</td><td>Usable</td></tr>
<tr><td>14</td><td colspan="3">N/A</td></tr>
<tr><td>15</td><td>Actual operation status</td><td>Stop/Pre Run</td><td>Runing</td></tr>
</table>
-encoder rotation direction: from the direction of the shaft, whether the encoder rotates clockwise or anticlockwise.
-limit switch min / Max: this function can be used by setting the 12 and 13 positions of the object 6000h to "1".
-actual operation status: with this function, to know the actual operation status of the encoder (see Chapter 5.2):
Bit15 = 0: "stop" or "pre run" status;
Bit15 = 1: "running" status.</td></tr>
</table>

GA-CANopen Series Multi-turn Absolute Encoder

Objects specified in the equipment regulations (DS 406)

Index - Subindex	Object Name [Data Type, Property]
6501-00	Hardware resolution per revolution [unsigned 32, RO] This object is used to define the number of resolvable steps per revolution on the hardware. To use other values, see object 6001h.
6502-00	Total turns of hardware [unsigned 32, RO] This object defines the maximum number of turns that can be measured on the hardware. "Total hardware resolution" = "obj_6501" X "obj_6502". To use other values, see 6001h and 6002h
6504-00	Supported error messages [unsigned 16, RO] Default value: 0000H (alarm not supported)
6506-00	Supported alarm information [unsigned 16, RO] Default value: 0000H (alarm not supported))
6507-00	Regulations and software version [unsigned 32, RO] Default value: 0301 0101h (software version = 1.1 encoder profile version = 3.1).
6508-00	Run time [unsigned 32, RO] Default = FFFF FFFFH (not used)
6509-00	Offset [integer32, RO] This object contains offset values, which are calculated based on preset and location values
650A-01	Production offset value [integer32, RO] This object contains the manufacturing offset value. This value gives the offset between the physical zero point and the numerical zero point on the encoder disk.
650B-00	Serial number [unsigned 32, RO] Default = FFFF FFFFH (not used)

Note:

In order for the changed parameters to take effect, you need to perform the save parameters operation (see object 1010h). If the operation of "save parameter" is not executed, the parameter will be lost when the "reset node" command, the "reset communication" command or the power is turned off.

Parameter Setting

The following are some examples of data exchange between the main controller and the encoder when setting parameters.

"Id" is used to represent the address of the encoder. The value is written in hexadecimal.

·Set operation and pre operation status

NMT Message	COB-ID	Order	Node
Run	000	01	ID
Pre Run	000	80	ID

Set Single Turn Resolution(2^{16}=0001 0000h

Controller→Encoder(Setting Request)

COB-ID	Order	Index		Sub Index	Process Data			
600+ID	23	01	60	00	00	00	01	00

Encoderr→Controlle(Confirm Request)

COB-ID	Order	Index		Sub Index	Process Data			
580+ID	60	01	60	00	00	00	00	00

Set Total Resolution(2^{28}=1000 0000h)

Controller→Encoder(Setting Request)

COB-ID	Order	Index		Sub Index	Process Data			
600+ID	23	02	60	00	00	00	00	10

Encoderr→Controlle(Confirm Request)

COB-ID	Order	Index		Sub Index	Process Data			
580+ID	60	02	60	00	00	00	00	00

GA-CANopen Series Multi-turn Absolute Encoder

Set Runnning Parameters (counting direction: clockwise, scaling function: use, limit switch: disable)

Controller→Encoder(Setting Request)

COB-ID	Order	Index		Sub Index	Process Data			
600+ID	2B	00	60	00	04	00	00	00

Encoderr→Controlle(Confirm Request)

COB-ID	Order	Index		Sub Index	Process Data			
580+ID	60	00	60	00	00	00	00	00

Set preset value (preset value is 1000 = 03e8h)

Controller→Encoder(Setting Request)

COB-ID	Order	Index		Sub Index	Process Data			
600+ID	23	03	60	00	E8	03	00	00

Encoderr→Controlle(Confirm Request)

COB-ID	Order	Index		Sub Index	Process Data			
580+ID	60	03	60	00	00	00	00	00

Set synchronization counter (n = 5 = 05H)

Controller→Encoder(Setting Request)

COB-ID	Order	Index		Sub Index	Process Data			
600+ID	2F	01	18	02	05	00	00	00

Encoderr→Controlle(Confirm Request)

COB-ID	Order	Index		Sub Index	Process Data			
580+ID	60	01	18	02	00	00	00	00

Disable sync mode

Read COB-ID via PDO2

Controller→Encoder(Setting Request)

COB-ID	Order	Index		Sub Index	Process Data			
600+ID	40	01	18	01	00	00	00	00

Encoderr→Controlle(Confirm Request)

COB-ID	Order	Index		Sub Index	Process Data			
580+ID	43	01	18	01	B0	B1	B2	B3

COB-ID read by PDO2: ((B3 < 24) (B3 < 16) (B1 < 8) (B0)).
The highest position is 1: B3 = 0x80;

Set the new COB-ID with PDO2:

Controller→Encoder(Setting Request)

COB-ID	Order	Index		Sub Index	Process Data			
600+ID	23	01	18	01	B0	B1	B2	B3

Encoderr→Controlle(Confirm Request)

COB-ID	Order	Index		Sub Index	Process Data			
580+ID	60	01	18	01	00	00	00	00

GA-CANopen Series Multi-turn Absolute Encoder

Use Cycle Mode
Set Cycle Time (100ms = 64H)
Controller→Encoder(Setting Request)

COB-ID	Order	Index		Sub Index	Process Data			
600+ID	2B	00	62	00	64	00	00	00

Encoderr→Controlle(Confirm Request)

COB-ID	Order	Index		Sub Index	Process Data			
580+ID	60	00	62	00	00	00	00	00

Read COB-ID via PD 01
Controller→Encoder(Setting Request)

COB-ID	Order	Index		Sub Index	Process Data			
600+ID	40	00	18	01	00	00	00	00

Encoderr→Controlle(Confirm Request)

COB-ID	Order	Index		Sub Index	Process Data			
580+ID	43	00	18	01	B0	B1	B2	B3

COB-ID read by pd01: ((B3 < 24) (B3 < 16) (B1 < 8) (B0)).
The highest position is 0: B3 & = 0x7F;

Set the new COB-ID with PD 01:
Controller→Encoder(Setting Request)

COB-ID	Order	Index		Sub Index	Process Data			
600+ID	23	00	18	01	B0	B1	B2	B3

Encoderr→Controlle(Confirm Request)

COB-ID	Order	Index		Sub Index	Process Data			
580+ID	60	00	18	01	00	00	00	00

In order to make the changed parameters to take effect, you need to perform the save parameters operation (see object 1010h).
If the operation of "save parameters" is not performed, the parameters will be lost when the "Resurrection node" command, the "reset communication" command or the power is turned off.

Warning objects

For the meaning of the warning information,
please refer to the section "SDO abnormal interrupt code" in "draft CIA standard 301" on www.can-cia.org.

Emergency object

An emergency object is triggered when an error occurs inside the device.

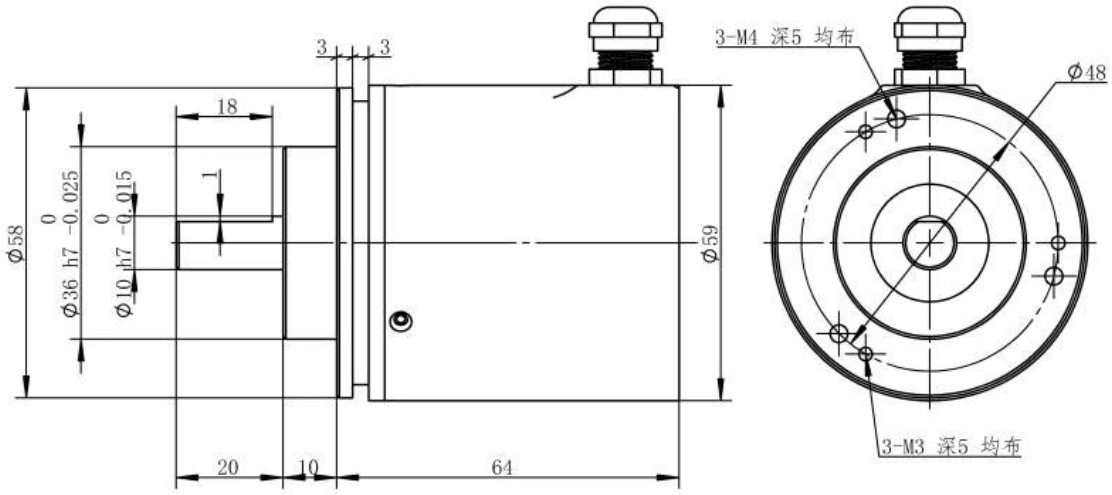
Emergency Object Structure

Identification	**CAN Data**			
COB-ID (HEX)	0	1	2	3~7
Object 1014h	Error Code		Error Register	Specific code
	Lowest position	Highest position	1001	00 00 00 00 00

-Error code defined:1000h
-node monitoring error5530h
-memory error

GA-CANopen Series Multi-turn Absolute Encoder

External Dimension



*The External Dimension can be customized as per customers demands

Note:

- ▶Adopt elastic soft connection shall be applied between encoder shaft and output shaft of user end to avoid damage of encoder shaft system due to serial movement and run out of user shaft.
- ▶Please pay attention to the allowable axle load during installation.
- ▶Make Sure that the difference Between Axial Degree of encoder shaft and user output shaft shall be no more than 0.20mm, and the deviation angle with axis shall be less than 1.5 °.
- ▶Try to avoid knocking and falling collision during installation;
- ▶Do not connect the power line and the ground wire in reverse.
- ▶The GND wire shall be as thick as possible, generally larger than φ 3.
- ▶Output lines of encoder shall not be overlapped with each other to avoid damaging output circuit.
- ▶Signal line of encoder shall not be connected to DC power supply or AC current to avoid damaging output circuit.
- ▶The motor and other equipment connected to the encoder shall be well grounded without static electricity.
- ▶Shielded cable shall be used for wiring.
- ▶Before starting the machine, carefully check whether the wiring is correct.
- ▶During long-distance transmission, the signal attenuation factor shall be considered, and the output mode with low output impedance and strong anti-interference ability shall be selected.
- ▶Avoid using in strong electromagnetic environment.