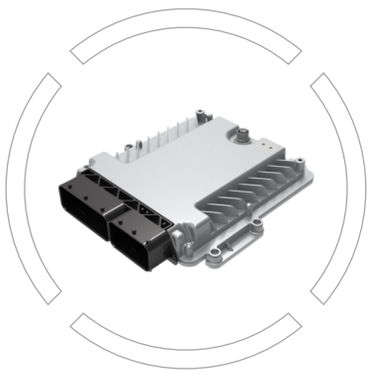
THM HYDRAULICS

T680 Series

Safety Controller



Index

Description

The T680 Safety Controller is an integral part of advanced system solutions. It provides the foundation for functional safety in safety-critical applications across the industrial, robotics, and construction machinery sectors. The controller is ideal for high-performance communication, multi-I/O control, and functional safety system projects.

The product is powered by an Infineon AURIX™ processor that meets the requirements of SIL 3 (IEC 61508) and ASIL D (ISO 26262) safety standards. This processor delivers the highest performance within Infineon' s 32-bit AURIX™ family, featuring up to six processor cores operating at 300 MHz each. This architecture enhances system performance, increases productivity, and improves overall control safety.

The T680 Safety Controller is designed according to the Category 3 safety architecture recommended by the EN 280 standard and complies with EN ISO 13849-1:2023, achieving Performance Level d (PL d). An independent CPU core is dedicated to system safety, enabling secure communication between modules over the internal data bus and supporting real-time diagnostic monitoring. This design ensures signal integrity and enhances data storage reliability.

The T680 Safety Controller features a 100 Mbps Ethernet interface with write functionality and is equipped with four-channel CAN FD communication as standard. CAN FD increases network bandwidth and improves error frame detection rates while maintaining compatibility with existing CAN hardware and software, particularly at the physical layer. This approach reduces the load of traditional CAN network topologies, improves communication efficiency, and shortens system debugging time.

All safety controllers support the widely adopted CODESYS V3.5 SP16 programming environment. This enables remote over-the-air (OTA) upgrades for both applications and firmware, simplifying maintenance, reducing total cost of ownership throughout the product lifecycle, and accelerating digital transformation initiatives.

Applications

- Safety-critical applications for the industrial, robotics and construction machinery industries.
- Aerial Work Platforms
- Lifting machinery
- Harbour and offshore machinery
- Material handling machinery
- Industrial, Robotics
- Construction Machinery

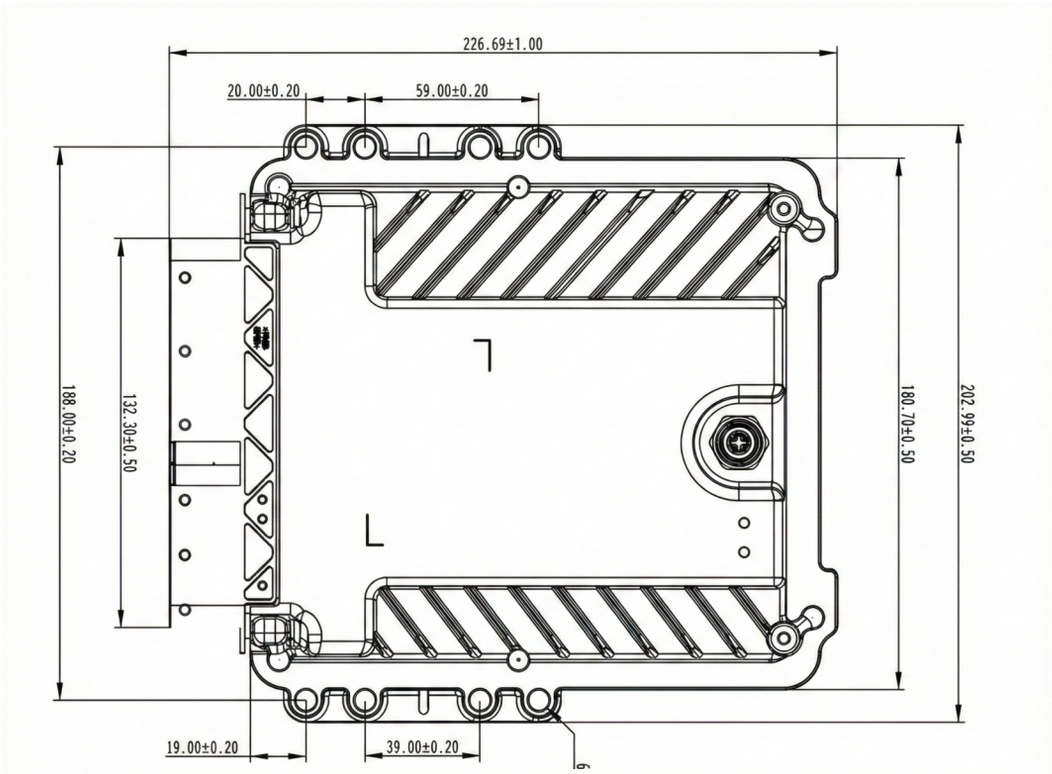
Features

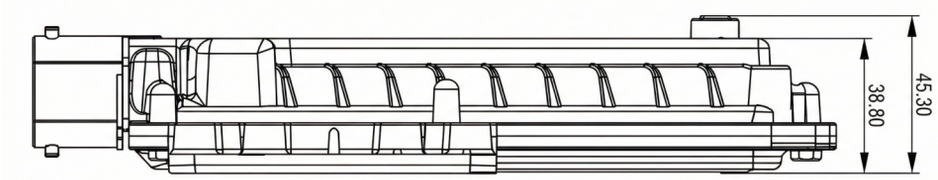
- 6-core processor, 300 MHz main frequency.
- High-speed SPI interface ferroelectric memory (FRAM) with up to 1 billion write/erase cycles. Dual-parameter backup storage improves data reliability.
- Four-channel CAN FD interface with speeds up to 2 Mbps, backward compatible with CAN 2.0A and CAN 2.0B protocols.
- High-performance CAN transceivers with enhanced port voltage resistance capability.
- Ethernet interface supports program debugging and firmware flashing, improving development efficiency.
- Enhanced power performance with reverse polarity protection.
- Supports remote OTA upgrades, including differential upgrade capability.
- Highly integrated design with 48 outputs and 76 inputs, totaling 124 I/O ports.14 × 2.5 A + 4 × 4 A High-side PWM/DO output ports for driving high-power loads.6 × 3 A + 4 × 4 A Low-side PWM/DO output ports. These can be combined with High-side outputs to form safety circuits and improve system safety levels.10 × 2.5 A + 4 × 4 A DO outputs, all with diagnostic functions.2 × AO (0–5 V) analog outputs.20-channel 4–20 mA / 0–5 V / DI_H input ports for convenient analog signal acquisition.
- Achieves ISO 13849 PL d functional safety level.
- Software developed using CODESYS V3.5, supporting static code analysis, software version management, and related functions.

Technical Specifications

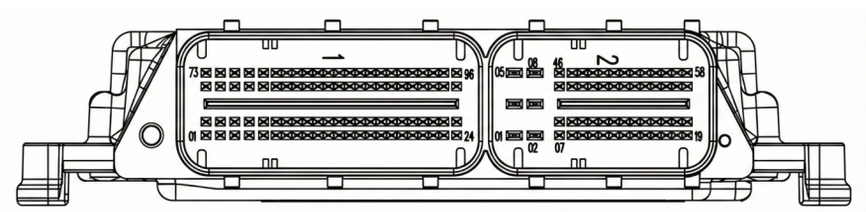
Term (in a mathematical formula)	Parameters
Working Voltage:	8 - 36V
Operating Temperature:	-40...85°
Storage Temperature:	-40...85°
IP Rating:	IP67 (with matching connecting harness)
CPU Frequency:	6-core 300MHz
Memory Capacity:	
RAM:	2.5 MB
FLASH:	16 MB
FRAM:	32 kBytes
Power supply anti-reverse connection:	yes
Moisture Resistance:	
(according to IEC 60068-2-30Db)	95% (+25°C... +55°C)
Size:	204.5 x 203 x 40.5 mm
Matching Connector:	96 Pins TE 9-1563173-9 58 Pins TE 2456937-1
Weight:	1.0 kg
Moisture resistance:	95%(25°C~+55°C) IEC 60068-2-30Db
Alternating heat and humidity:	(IEC 60068-2-30Db) 95 % (25°C~ +55°C)
Sinusoidal vibration:	(ISO 16750-3) 10-500 Hz, 0.35 mm / 5G, 10 cycles each axis, X, Y, Z in each axis, 10 cycles (2 hours)
Broadband random vibration:	ISO 16750-3 13.4 m/s^2, X, Y, Z each axis, 4 hours, normal work.
Shock:	IEC60068-2-27 3rd order, 3 specified coordinate directions, 30G, 18ms half sine wave pulse
Electrostatic discharge immunity:	ISO 10605:2008 ± 15KV air discharge and ±8KV contact discharge, the operation is normal after the test.
Conducted transient immunity:	ISO 7637-2:2011 test pulse 1, -150V, operating state: C test pulse 2a, +112V, operating state: A test pulse 2b, +10V, operating state: C test pulse 3a, -220V, operating state: A test pulse 3b, +150V, operating state: A
Load dumping:	ISO16750-2, chp4.6.4 Grade C
Electromagnetic radiation immunity:	ISO 11452-4：2011 30 V/m radiated field; 60 mA current injection (BCI)

Product Dimensions (Dimensions in mm)





Pin Definitions



Pin: XYY
X: Connector number
Y: Pin number

Pin Definitions

Pin		Functional Description
101	OUT_16	High-side PWM/DO output with current measurement up to 4A
102	OUT_36	High-side PWM/DO output with current measurement up to 2.5A
103	OUT_20	High-side DO output with current measurement up to 4A
104	OUT_34	AO Output, 0~5V
105	OUT_33	AO Output, 0~5V
106	OUT_22	High-side DO output with current measurement up to 2.5A
107	OUT_21	High-side DO output with current measurement up to 2.5A
108	IN_68	Function 1: Low effective frequency input (0...10K) | Function 2: High effective frequency input (0...10K) | Function 3: High effective switching input | Function 4: Low effective switching input
109	IN_75	DSM frequency input (0...9K) / Low active frequency input (threshold 5V/9V) / Low active switching input
110	IN_73	DSM frequency input (0...9K) / Low active frequency input (threshold 5V/9V) / Low active switching input / Pin110 and Pin111 form a double pulse input
111	IN_74	DSM frequency input (0...9K) / Low active frequency input (threshold 5V/9V) / Low active switching input / Pin110 and Pin111 form a double pulse input
112	IN_71	DSM frequency input (0...9K) / Low active frequency input (threshold 5V/9V) / Low active switching input / Pin112 and Pin113 form a double pulse input
113	IN_72	DSM frequency input (0...9K) / Low active frequency input (threshold 5V/9V) / Low active switching input / Pin112 and Pin113 form a double pulse input
114	IN_42	Function 2: Low effective switch input
115	IN_43	Function 1: High valid switching input | Function 2: Low valid switching input, the default is low valid switching input.
116	IN_63	Function 1: 0...50KΩ resistive analog | Function 2: Low active switching
117	IN_64	Function 1: 0...50KΩ resistive analog | Function 2: Low active switching
118	IN_65	Function 1: 0...50KΩ resistive analog | Function 2: Low active switching
119	IN_40	0...32 V analog input/highly effective switching input (threshold voltage configurable)
120	IN_41	0...32 V analog input/highly effective switching input (threshold voltage configurable)
121	IN_62	Function 1: 0...50KΩ resistive analog | Function 2: Low active switching
122	GND	GND
123	GND	GND
124	GND	GND
125	OUT_17	High-side PWM/DO output with current measurement up to 4A
126	OUT_15	High-side PWM/DO output with current measurement up to 4A
127	OUT_35	High-side PWM/DO output with current measurement up to 2.5A
128	OUT_19	High-side DO output with current measurement up to 4A
129	OUT_18	High-side PWM/DO output with current measurement up to 4A
130	OUT_13	High-side PWM/DO output with current measurement up to 2.5A
131	OUT_14	High-side PWM/DO output with current measurement up to 2.5A
132	IN_70	Highly effective frequency inputs (500...10K), ports 132/133 are used as double pulse and can be multiplexed as highly effective inputs / 0...32V analog inputs
133	IN_69	Highly effective frequency inputs (500...10K), ports 132/133 are used as double pulse and can be multiplexed as highly effective inputs / 0...32V analog inputs
134	IN_18	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
135	IN_17	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
136	IN_15	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
137	IN_16	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
138	IN_52	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
139	IN_38	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
140	IN_44	Function 1: High valid switching input | Function 2: Low valid switching input, the default is low valid switching input.
141	IN_39	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
142	IN_23	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input

Pin Definitions

143	IN_22	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
144	IN_20	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
145	GND	AGND,GND_S2
146	GND	AGND,GND_S1
147	IN_04	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
148	IN_06	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
149	OUT_09	High-side PWM/DO output with current measurement up to 2.5A
150	OUT_11	High-side PWM/DO output with current measurement up to 2.5A
151	OUT_05	High-side PWM/DO output with current measurement up to 2.5A
152	OUT_07	High-side PWM/DO output with current measurement up to 2.5A
153	OUT_01	High-side PWM/DO output with current measurement up to 2.5A
154	OUT_03	High-side PWM/DO output with current measurement up to 2.5A
155	IN_45	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
156	IN_46	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
157	IN_10	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
158	IN_08	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
159	IN_07	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
160	CAN4_L	CAN4_L
161	CAN4_H	CAN4_H
162	CAN3_L	CAN3_L
163	CAN3_H	CAN3_H
164	IN_33	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
165	IN_61	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
166	IN_34	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
167	IN_58	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
168	IN_53	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
169	IN_59	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
170	IN_60	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
171	IN_56	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
172	IN_47	Function 1: High valid switching input | Function 2: Low valid switching input, the default is low valid switching input.
173	OUT_10	High-side PWM/DO output with current measurement up to 2.5A
174	OUT_12	High-side PWM/DO output with current measurement up to 2.5A
175	OUT_06	High-side PWM/DO output with current measurement up to 2.5A
176	OUT_08	High-side PWM/DO output with current measurement up to 2.5A
177	OUT_02	High-side PWM/DO output with current measurement up to 2.5A
178	OUT_04	High-side PWM/DO output with current measurement up to 2.5A
179	OUT_41	Low-side PWM/DO output with current measurement up to 5A
180	OUT_42	Low-side PWM/DO output with current measurement up to 5A
181	OUT_43	Low-side PWM/DO output with current measurement up to 3A
182	OUT_44	Low-side PWM/DO output with current measurement up to 3A
183	OUT_37	Low-side PWM/DO output with current measurement up to 5A
184	OUT_38	Low-side PWM/DO output with current measurement up to 5A
185	OUT_39	Low-side PWM/DO output with current measurement up to 3A
186	OUT_40	Low-side PWM/DO output with current measurement up to 3A

Pin Definitions

187	OUT_46	Low-side PWM/DO output with current measurement up to 3A
188	OUT_45	Low-side PWM/DO output with current measurement up to 3A
189	OUT_24	High-side DO output with current measurement up to 2.5A
190	OUT_23	High-side DO output with current measurement up to 2.5A
191	IN_48	Function 1: High valid switching input | Function 2: Low valid switching input, the default is low valid switching input.
192	IN_49	Function 1: High valid switching input | Function 2: Low valid switching input, the default is low valid switching input.
193	OUT_26	High-side DO output with current measurement up to 2.5A
194	OUT_25	High-side DO output with current measurement up to 2.5A
195	OUT_48	High-side DO output with current measurement up to 2.5A
196	IN_50	Function 1: High valid switching input | Function 2: Low valid switching input, the default is low valid switching input.
201	UBP	UBP
202	GND	GND
203	UBP	UBP
204	UBP	UBP
205	UBP	UBP
206	UBP	UBP
207		GND
208	IN_67	High/low active frequency inputs (0...10K), ports 208/213 are used as double pulse and can be multiplexed as high/low active inputs
209	IN_03	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
210	IN_14	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
211	IN_19	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
212	IN_28	Function 1: First group, the same group can be configured as high valid switching input or low valid switching input at the same time, the software configuration will take effect after reboot | Function 2: 0~10V analog input
213	IN_66	High/low active frequency inputs (0...10K), ports 208/213 are used as double pulse and can be multiplexed as high/low active inputs
214	IN_09	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
215	CAN1_H	CAN1_H
216	CAN1_L	CAN1_L
217	IN_36	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
218	IN_54	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
219	VSS_1	
220		GND
221	OUT_47	High-side PWM/DO output with current measurement up to 2.5A
222	IN_13	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
223	IN_05	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
224	IN_21	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
225	IN_27	Function 1: First group, the same group can be configured as high valid switching input or low valid switching input at the same time, the software configuration will take effect after reboot | Function 2: 0~10V analog input
226	IN_37	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
227	IN_57	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
228	INH	External cut-off of main switch, >4.5V allows output, <1V prohibits output, disconnection can't output
229	IN_55	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
230	IN_76	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input

Pin Definitions

231	IN_35	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
232	VSS_2	Vref10V,1A drive capability
233	GND	GND
234	IN_26	Function 1: First group, the same group can be configured as high valid switching input or low valid switching input at the same time, the software configuration will take effect after reboot | Function 2: 0~10V analog input
235	IN_24	Function 1: Highly effective switching input | Function 2: 0~5V analog input | Function 3: 4~20mA analog input
236	IN_02	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
237	IN_30	Function 1: Second group, the same group can be configured as high valid switching input or low valid switching input at the same time, the software configuration will take effect after reboot.
238	IN_12	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
239	IN_31	Function 1: Second group, the same group can be configured as high valid switching input or low valid switching input at the same time, the software configuration will take effect after reboot.
240	KL15	KL15
241	OUT_28	High-side DO output with current measurement up to 4A
242	OUT_29	High-side DO output with current measurement up to 2.5A
243	OUT_27	High-side DO output with current measurement up to 4A
244	OUT_31	High-side DO output with current measurement up to 2.5A
245	UBS	UBS
246	GND	GND
247	IN_25	Function 1: First group, the same group can be configured as high valid switching input or low valid switching input at the same time, the software configuration will take effect after reboot | Function 2: 0~10V analog input
248	IN_29	Function 1: Second group, the same group can be configured as high valid switching input or low valid switching input at the same time, the software configuration will take effect after reboot.
249	IN_51	Function 1: High valid switching input | Function 2: Low valid switching input, the default is low valid switching input.
250	IN_01	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
251	IN_32	Function 1: Second group, the same group can be configured as high valid switching input or low valid switching input at the same time, the software configuration will take effect after reboot.
252	IN_11	Function 1: Highly effective switching input with software configurable threshold voltage (max.10V) | Function 2: 0~10V analog input
253	CAN2_H	CAN2_H
254	CAN2_L	CAN2_L
255	VSS_3	Vref5V,250mA drive capability
256	OUT_30	High-side DO output with current measurement up to 2.5A
257	OUT_32	High-side DO output with current measurement up to 2.5A
258	UBS	UBS

Diagram of Electrical connections

20A* fuse, the fuse size of the power output is selected according to the total load current of the controller

Hardware-Ports

Ports	Name	Definition
101	OUT_16	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
102	OUT_36	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
103	OUT_20	High-side Switching Output | Maximum Current 4A
104	OUT_34	AO voltage output, 0~5V
105	OUT_33	AO voltage output, 0~5V
106	OUT_22	High-side Switching Output | Maximum Current 2.5A
107	OUT_21	High-side Switching Output | Maximum Current 2.5A
108	IN_68	High / low valid frequency input (0~10KHZ) | High / low valid switch input
109	IN_75	DSM frequency input (0...9KHZ) | Low active frequency input (threshold 5V/ 9V) | Low active switching input
110	IN_73	DSM frequency input (0...9KHZ) | Low active frequency input (threshold 5V/ 9V) | Low active switching input | Dual pulse input +IN_74
111	IN_74	DSM frequency input (0...9KHZ) | Low active frequency input (threshold 5V/ 9V) | Low active switching input | Dual pulse input +IN_73
112	IN_71	DSM frequency input (0...9KHZ) | Low active frequency input (threshold 5V/ 9V) | Low active switching input | Dual pulse input +IN_72
113	IN_72	DSM frequency input (0...9KHZ) | Low active frequency input (threshold 5V/ 9V) | Low active switching input | Dual pulse input +IN_71
114	IN_42	Low effective switching input | Secondary cut-off switch, below 0.7V allow output, above 2V prohibit output
115	IN_43	Low/high active switching inputs
116	IN_63	0...50K Ω resistive analog | low effective switching inputs
117	IN_64	0...50K Ω resistive analog | low effective switching inputs
118	IN_65	0...50K Ω resistive analog | low effective switching inputs
119	IN_40	Highly effective switching inputs | 0~32V analog inputs
120	IN_41	Highly effective switching inputs | 0~32V analog inputs
121	IN_62	0...50K Ω resistive analog | low effective switching inputs
122	GND	GND
123	GND	GND
124	GND	GND
125	OUT_17	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
126	OUT_15	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
127	OUT_35	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
128	OUT_19	High-side Switching Output | Maximum Current 4A
129	OUT_18	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
130	OUT_13	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
131	OUT_14	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
132	IN_70	Highly Effective Frequency Input (500~10KHZ) | Highly Effective Switch Input | 0~32V analog Input | Dual Pulse Input +IN_69
133	IN_69	Highly Effective Frequency Input (500~10KHZ) | Highly Effective Switch Input | 0~32V analog Input | Dual Pulse Input +IN_70
134	IN_18	High/Low Valid Switching Input | 0~10V analog Input
135	IN_17	High/Low Valid Switching Input | 0~10V analog Input
136	IN_15	High/Low Valid Switching Input | 0~10V analog Input
137	IN_16	High/Low Valid Switching Input | 0~10V analog Input
138	IN_52	Highly effective switching input | 0~5V analog input | 4~20mA analog input
139	IN_38	Highly effective switching input | 0~5V analog input | 4~20mA analog input
140	IN_44	Low/high active switching inputs
141	IN_39	Highly effective switching input | 0~5V analog input | 4~20mA analog input

Hardware-Ports

142	IN_23	Highly effective switching input | 0~5V analog input | 4~20mA analog input
143	IN_22	High/Low Valid Switching Input | 0~10V analog Input
144	IN_20	High/Low Valid Switching Input | 0~10V analog Input
145	AGND	AGND,GND_S2
146	AGND	AGND,GND_S1
147	IN_4	High/Low Valid Switching Input | 0~10V analog Input
148	IN_6	High/Low Valid Switching Input | 0~10V analog Input
149	OUT_9	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
150	OUT_11	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
151	OUT_5	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
152	OUT_7	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
153	OUT_1	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
154	OUT_3	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
155	IN_45	Low/high active switching inputs
156	IN_46	Low/high active switching inputs
157	IN_10	High/Low Valid Switching Input | 0~10V analog Input
158	IN_8	High/Low Valid Switching Input | 0~10V analog Input
159	IN_7	High/Low Valid Switching Input | 0~10V analog Input
160	CAN4_L	CAN4 low
161	CAN4_H	CAN4 high
162	CAN3_L	CAN3 low
163	CAN3_H	CAN3 high
164	IN_33	Highly effective switching input | 0~5V analog input | 4~20mA analog input
165	IN_61	Highly effective switching input | 0~5V analog input | 4~20mA analog input
166	IN_34	Highly effective switching input | 0~5V analog input | 4~20mA analog input
167	IN_58	Highly effective switching input | 0~5V analog input | 4~20mA analog input
168	IN_53	Highly effective switching input | 0~5V analog input | 4~20mA analog input
169	IN_59	Highly effective switching input | 0~5V analog input | 4~20mA analog input
170	IN_60	Highly effective switching input | 0~5V analog input | 4~20mA analog input
171	IN_56	Highly effective switching input | 0~5V analog input | 4~20mA analog input
172	IN_47	Low/high active switching inputs
173	OUT_10	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
174	OUT_12	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
175	OUT_6	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
176	OUT_8	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
177	OUT_2	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
178	OUT_4	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
179	OUT_41	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
180	OUT_42	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
181	OUT_43	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 3A
182	OUT_44	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 3A
183	OUT_37	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
184	OUT_38	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
185	OUT_39	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 3A

Hardware-Ports

186	OUT_40	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 3A
187	OUT_46	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 3A
188	OUT_45	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 3A
189	OUT_24	High-side Switching Output | Maximum Current 2.5A
190	OUT_23	High-side Switching Output | Maximum Current 2.5A
191	IN_48	Low/high active switching inputs
192	IN_49	Low/high active switching inputs
193	OUT_26	High-side Switching Output | Maximum Current 2.5A
194	OUT_25	High-side Switching Output | Maximum Current 2.5A
195	OUT_48	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
196	IN_50	Low/high active switching inputs
201	+UBP	UBP
202	GND	GND
203	+UBP	UBP
204	+UBP	UBP
205	+UBP	UBP
206	+UBP	UBP
207	GND	GND
208	IN_67	High/low effective frequency input (0~10KHZ) | High/low effective switch input | Dual pulse input +IN_66
209	IN_3	High/Low Valid Switching Input | 0~10V analog Input
210	IN_14	High/Low Valid Switching Input | 0~10V analog Input
211	IN_19	High/Low Valid Switching Input | 0~10V analog Input
212	IN_28	High/Low Valid Switching Input | 0~10V analog Input
213	IN_66	High/low effective frequency input (0~10KHZ) | High/low effective switch input | Dual pulse input +IN_67
214	IN_9	High/Low Valid Switching Input | 0~10V analog Input
215	CAN1_H	CAN1 high
216	CAN1_L	CAN1 low
217	IN_36	Highly effective switching input | 0~5V analog input | 4~20mA analog input
218	IN_54	Highly effective switching input | 0~5V analog input | 4~20mA analog input
219	VSS1	Vref 5V, 250mA drive capability
220	GND	GND
221	OUT_47	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
222	IN_13	High/Low Valid Switching Input | 0~10V analog Input
223	IN_5	High/Low Valid Switching Input | 0~10V analog Input
224	IN_21	High/Low Valid Switching Input | 0~10V analog Input
225	IN_27	High/Low Valid Switching Input | 0~10V analog Input
226	IN_37	Highly effective switching input | 0~5V analog input | 4~20mA analog input
227	IN_57	Highly effective switching input | 0~5V analog input | 4~20mA analog input
228	INH	[Emergency stop switch] External cut-off main switch, >4.5V allows output, <1V prohibits output
229	IN_55	Highly effective switching input | 0~5V analog input | 4~20mA analog input
230	IN_76	Highly effective switching input | 0~5V analog input | 4~20mA analog input
231	IN_35	Highly effective switching input | 0~5V analog input | 4~20mA analog input
232	VSS2	Vref 10V, 1A drive capability
233	GND	GND
234	IN_26	High/Low Valid Switching Input | 0~10V analog Input
235	IN_24	Highly effective switching input | 0~5V analog input | 4~20mA analog input
236	IN_2	High/Low Valid Switching Input | 0~10V analog Input
237	IN_30	High/Low Valid Switching Input | 0~10V analog Input
238	IN_12	High/Low Valid Switching Input | 0~10V analog Input
239	IN_31	High/Low Valid Switching Input | 0~10V analog Input
240	KL15	KL15
241	OUT_28	High-side Switching Output | Maximum Current 4A

Hardware-Ports

242	OUT_29	High-side Switching Output | Maximum Current 2.5A
243	OUT_27	High-side Switching Output | Maximum Current 4A
244	OUT_31	High-side Switching Output | Maximum Current 2.5A
245	+UBS	UBS
246	GND	GND
247	IN_25	High/Low Valid Switching Input | 0~10V analog Input
248	IN_29	High/Low Valid Switching Input | 0~10V analog Input
249	IN_51	Low/high active switching inputs
250	IN_1	High/Low Valid Switching Input | 0~10V analog Input
251	IN_32	High/Low Valid Switching Input | 0~10V analog Input
252	IN_11	High/Low Valid Switching Input | 0~10V analog Input
253	CAN2_H	CAN2 high
254	CAN2_L	CAN2 low
255	VSS3	Vref 5V, 250mA drive capability
256	OUT_30	High-side Switching Output | Maximum Current 2.5A
257	OUT_32	High-side Switching Output | Maximum Current 2.5A
258	+UBS	UBS

Input

No.	Ports	Name	Definition
1	250	IN_1	High/Low Valid Switching Input | 0~10V analog Input
2	236	IN_2	High/Low Valid Switching Input | 0~10V analog Input
3	209	IN_3	High/Low Valid Switching Input | 0~10V analog Input
4	147	IN_4	High/Low Valid Switching Input | 0~10V analog Input
5	223	IN_5	High/Low Valid Switching Input | 0~10V analog Input
6	148	IN_6	High/Low Valid Switching Input | 0~10V analog Input
7	159	IN_7	High/Low Valid Switching Input | 0~10V analog Input
8	158	IN_8	High/Low Valid Switching Input | 0~10V analog Input
9	214	IN_9	High/Low Valid Switching Input | 0~10V analog Input
10	157	IN_10	High/Low Valid Switching Input | 0~10V analog Input
11	252	IN_11	High/Low Valid Switching Input | 0~10V analog Input
12	238	IN_12	High/Low Valid Switching Input | 0~10V analog Input
13	222	IN_13	High/Low Valid Switching Input | 0~10V analog Input
14	210	IN_14	High/Low Valid Switching Input | 0~10V analog Input
15	136	IN_15	High/Low Valid Switching Input | 0~10V analog Input
16	137	IN_16	High/Low Valid Switching Input | 0~10V analog Input
17	135	IN_17	High/Low Valid Switching Input | 0~10V analog Input
18	134	IN_18	High/Low Valid Switching Input | 0~10V analog Input
19	211	IN_19	High/Low Valid Switching Input | 0~10V analog Input
20	144	IN_20	High/Low Valid Switching Input | 0~10V analog Input
21	224	IN_21	High/Low Valid Switching Input | 0~10V analog Input
22	143	IN_22	High/Low Valid Switching Input | 0~10V analog Input
23	142	IN_23	Highly effective switching input | 0~5V analog input | 4~20mA analog input
24	235	IN_24	Highly effective switching input | 0~5V analog input | 4~20mA analog input
25	247	IN_25	High/Low Valid Switching Input | 0~10V analog Input
26	234	IN_26	High/Low Valid Switching Input | 0~10V analog Input
27	225	IN_27	High/Low Valid Switching Input | 0~10V analog Input
28	212	IN_28	High/Low Valid Switching Input | 0~10V analog Input
29	248	IN_29	High/Low Valid Switching Input | 0~10V analog Input
30	237	IN_30	High/Low Valid Switching Input | 0~10V analog Input
31	239	IN_31	High/Low Valid Switching Input | 0~10V analog Input
32	251	IN_32	High/Low Valid Switching Input | 0~10V analog Input
33	164	IN_33	Highly effective switching input | 0~5V analog input | 4~20mA analog input
34	166	IN_34	Highly effective switching input | 0~5V analog input | 4~20mA analog input
35	231	IN_35	Highly effective switching input | 0~5V analog input | 4~20mA analog input
36	217	IN_36	Highly effective switching input | 0~5V analog input | 4~20mA analog input
37	226	IN_37	Highly effective switching input | 0~5V analog input | 4~20mA analog input
38	139	IN_38	Highly effective switching input | 0~5V analog input | 4~20mA analog input
39	141	IN_39	Highly effective switching input | 0~5V analog input | 4~20mA analog input
40	119	IN_40	Highly effective switching inputs | 0~32V analog inputs
41	120	IN_41	Highly effective switching inputs | 0~32V analog inputs
42	114	IN_42	Low effective switching input | Secondary cut-off switch, below 0.7V allow output, above 2V prohibit output
43	115	IN_43	Low/high active switching inputs
44	140	IN_44	Low/high active switching inputs
45	155	IN_45	Low/high active switching inputs
46	156	IN_46	Low/high active switching inputs
47	172	IN_47	Low/high active switching inputs
48	191	IN_48	Low/high active switching inputs
49	192	IN_49	Low/high active switching inputs
50	196	IN_50	Low/high active switching inputs
51	249	IN_51	Low/high active switching inputs
52	138	IN_52	Highly effective switching input | 0~5V analog input | 4~20mA analog input
53	168	IN_53	Highly effective switching input | 0~5V analog input | 4~20mA analog input
54	218	IN_54	Highly effective switching input | 0~5V analog input | 4~20mA analog input
55	229	IN_55	Highly effective switching input | 0~5V analog input | 4~20mA analog input
56	171	IN_56	Highly effective switching input | 0~5V analog input | 4~20mA analog input
57	227	IN_57	Highly effective switching input | 0~5V analog input | 4~20mA analog input
58	167	IN_58	Highly effective switching input | 0~5V analog input | 4~20mA analog input
59	169	IN_59	Highly effective switching input | 0~5V analog input | 4~20mA analog input
60	170	IN_60	Highly effective switching input | 0~5V analog input | 4~20mA analog input
61	165	IN_61	Highly effective switching input | 0~5V analog input | 4~20mA analog input
62	121	IN_62	0...50KÑ resistive analog | low effective switching inputs
63	116	IN_63	0...50KÑ resistive analog | low effective switching inputs
64	117	IN_64	0...50KÑ resistive analog | low effective switching inputs
65	118	IN_65	0...50KÑ resistive analog | low effective switching inputs
66	213	IN_66	High/low effective frequency input (0~10KHZ) | High/low effective switch input | Dual pulse input +IN_67
67	208	IN_67	High/low effective frequency input (0~10KHZ) | High/low effective switch input | Dual pulse input +IN_66
68	108	IN_68	High / low valid frequency input (0~10KHZ) | High / low valid switch input
69	133	IN_69	Highly Effective Frequency Input (500~10KHZ) | Highly Effective Switch Input | 0~32V analog Input | Dual Pulse Input +IN_70
70	132	IN_70	Highly Effective Frequency Input (500~10KHZ) | Highly Effective Switch Input | 0~32V analog Input | Dual Pulse Input +IN_69
71	112	IN_71	DSM frequency input (0...9KHZ) | Low active frequency input (threshold 5V/9V) | Low active switching input | Dual pulse input +IN_72

Input

72	113	IN_72	DSM frequency input (0...9KHZ) | Low active frequency input (threshold 5V/9V) | Low active switching input | Dual pulse input +IN_71
73	110	IN_73	DSM frequency input (0...9KHZ) | Low active frequency input (threshold 5V/9V) | Low active switching input | Dual pulse input +IN_74
74	111	IN_74	DSM frequency input (0...9KHZ) | Low active frequency input (threshold 5V/9V) | Low active switching input | Dual pulse input +IN_73
75	109	IN_75	DSM frequency input (0...9KHZ) | Low active frequency input (threshold 5V/9V) | Low active switching input
76	230	IN_76	Highly effective switching input | 0~5V analog input | 4~20mA analog input

Output

No.	Ports	Name	Definition
1	153	OUT_1	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
2	177	OUT_2	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
3	154	OUT_3	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
4	178	OUT_4	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
5	151	OUT_5	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
6	175	OUT_6	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
7	152	OUT_7	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
8	176	OUT_8	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
9	149	OUT_9	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
10	173	OUT_10	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
11	150	OUT_11	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
12	174	OUT_12	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
13	130	OUT_13	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
14	131	OUT_14	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
15	126	OUT_15	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
16	101	OUT_16	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
17	125	OUT_17	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
18	129	OUT_18	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
19	128	OUT_19	High-side Switching Output | Maximum Current 4A
20	103	OUT_20	High-side Switching Output | Maximum Current 4A
21	107	OUT_21	High-side Switching Output | Maximum Current 2.5A
22	106	OUT_22	High-side Switching Output | Maximum Current 2.5A
23	190	OUT_23	High-side Switching Output | Maximum Current 2.5A
24	189	OUT_24	High-side Switching Output | Maximum Current 2.5A
25	194	OUT_25	High-side Switching Output | Maximum Current 2.5A
26	193	OUT_26	High-side Switching Output | Maximum Current 2.5A
27	243	OUT_27	High-side Switching Output | Maximum Current 2.5A
28	241	OUT_28	High-side Switching Output | Maximum Current 4A
29	242	OUT_29	High-side Switching Output | Maximum Current 4A
30	256	OUT_30	High-side Switching Output | Maximum Current 2.5A
31	244	OUT_31	High-side Switching Output | Maximum Current 2.5A
32	257	OUT_32	High-side Switching Output | Maximum Current 2.5A
33	105	OUT_33	AO voltage output, 0~5V
34	104	OUT_34	AO voltage output, 0~5V
35	127	OUT_35	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A

Output

36	102	OUT_36	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
37	183	OUT_37	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
38	184	OUT_38	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
39	185	OUT_39	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 3A
40	186	OUT_40	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 3A
41	179	OUT_41	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
42	180	OUT_42	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 4A
43	181	OUT_43	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 3A
44	182	OUT_44	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 3A
45	188	OUT_45	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 3A
46	187	OUT_46	Low-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 3A
47	221	OUT_47	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A
48	195	OUT_48	High-side Switching Output | PWM Duty Cycle Output | PWM Constant Current Output | Maximum Current 2.5A

CAN

No.	Ports	Name	Definition
1	215	CAN1_H	CAN1 high
2	216	CAN1_L	CAN1 low
3	253	CAN2_H	CAN2 high
4	254	CAN2_L	CAN2 low
5	163	CAN3_H	CAN3 high
6	162	CAN3_L	CAN3 low
7	161	CAN4_H	CAN4 high
8	160	CAN4_L	CAN4 low

UBP/UBS/GND

No.	Ports	Name	Definition
1	240	KL15	KL15
2	201	+UBP	UBP
3	203	+UBP	UBP
4	204	+UBP	UBP
5	205	+UBP	UBP
6	206	+UBP	UBP
7	245	+UBS	UBS
8	258	+UBS	UBS
9	122	GND	GND
10	123	GND	GND
11	124	GND	GND
12	202	GND	GND
13	207	GND	GND
14	220	GND	GND
15	233	GND	GND
16	246	GND	GND
17	145	AGND	AGND,GND_S2
18	146	AGND	AGND,GND_S1

STOP

No.	Ports	Name	Definition
1	228	INH	[Emergency stop switch] External cut-off main switch, >4.5V allows output, <1V prohibits output
2	114	IN_42	Low effective switching input | Secondary cut-off switch, below 0.7V allow output, above 2V prohibit output

Reference power supply 5V/10V

No.	Ports	Name	Definition
1	219	VSS1	Vref 5V, 250mA drive capability
2	232	VSS2	Vref 10V, 1A drive capability
3	255	VSS3	Vref 5V, 250mA drive capability

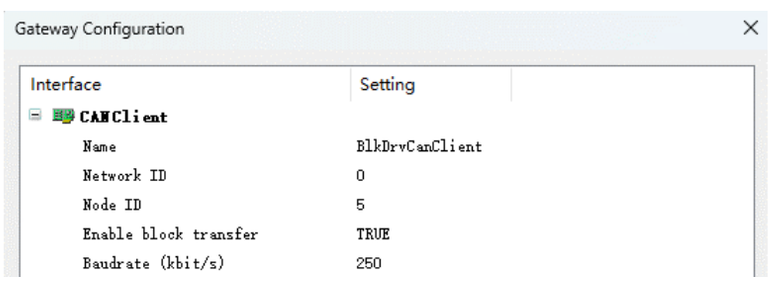
CAN Debugging Online

The T680 supports online on-line debugging of CAN port and downloading of CoDeSys application.
By default, CAN1 is used as the debug port, CAN1H is Pin215 and CAN1L is Pin216.

No.	Ports	Name	Definition
1	215	CAN1_H	CAN1 high
2	216	CAN1_L	CAN1 low
3	253	CAN2_H	CAN2 high
4	254	CAN2_L	CAN2 low
5	163	CAN3_H	CAN3 high
6	162	CAN3_L	CAN3 low
7	161	CAN4_H	CAN4 high
8	160	CAN4_L	CAN4 low

If you need to configure the CAN debug port, please refer : Sofeware——>Communication Related——>CAN

The default controller Node-ID is 1 and the baud rate is 250kbps.
Add CANClient interface to Gateway Configuration in codesys, Network ID is 0, NodeID can be set to another ID that is not the same as the server ID, such as 2, 3, 4, 5... etc., baud rate is 250, add configuration setting to enable block transfer (faster speed, will make bus load rate high).



After the configuration is complete go to the Gateway PLC directory in the codesys installation directory and locate the CODESYSControl.cfg file and find the path to the configuration file for the codesys gateway, for example:

[SysFile]
Windows.WorkingDirectory=C:\ProgramData\CODESYS\CODESYSGatewayV3\[folder name]\

Go to the above directory, open the file Gateway.cfg, open the configuration parameters corresponding to CAN online, refer to the red font as below, save the configuration file and restart the gateway after the configuration is completed. after restarting, you can check the startup logs of the gateway in the configuration file directory to make sure the startup is successful. The following log shows that the PCAN device has been found and the CAN interface has been successfully opened.

Note: When Can online is turned on, gateway will broadcast every 10s to discover the controller of the bus. If no controller is found, you can wait for 10s before scanning. If the controller is still not found, please check the can line.

[ComponentManager]
;--> Additional components for CmpBlkDrvCanClient

CAN Debugging Online

Component.1=CmpBlkDrvCANClient
Component.2=CmpCAASdoClient
Component.3=CmpCAACanL2
Component.4=CmpPCANBasicDrv
;Component.5=CmpCANFoxDrv
;Component.6=CmpIxxatCANDrv
;<-- Additional components for CmpBlkDrvCanClient
;--> Additional components for CmpBlkDrvUsb
;Component.6=CmpUsbMpdUsbLib
;<-- Additional components for CmpBlkDrvUsb

[CmpCAASdoClient]
MaxSegmentsPerCycle=32 ; for block download: maximum of sent segments per cycle; 0 means infinity
If you use PCAN as an online tool, you need to open CmpPCANBasicDrv, if you use IXXAT, you need to open CmpIxxatCANDrv, and block other can drivers such as CmpPCANBasicDrv, CmpCANFoxDrv, etc.

Ethernet online

The T680 supports Ethernet on-line debugging and downloading of the CoDeSys application.
The default IP address and gateway information is as follows:

typology	(be) worth	instructions
default IP address	192.168.1.72	modifiable
subnet mask	255.255.255.0	

Before connecting, you need to modify the Ethernet property configuration of the computer used for connecting so that the computer and the controller are in the same LAN segment,
Restart Gataway after configuration is complete to scan device online.

Hardware-Ports

SN	Development environment (computer)	Note
1	CODESYS V3.5.18.50	Minimum Version Requirements
2	JSHR Std Library, 1.0.0.7	Developed by Varex Controls
3	ServiceTool V1.3.2	Service Supervisor developed

Library

JSHR Std Library consists of 8 parts:
1. Input, this section focuses on describing the port configuration methods, port status acquisition methods for various types of input ports of the T680 controller;
2. Output, this section focuses on describing the port configuration methods for various types of outputs of the T680 controller, output port control methods, diagnostic methods for output ports, and so on.
3. Communication: This section focuses on describing the usage of the T680 controller CAN communication.
4. Diag: This section focuses on describing the various diagnostic configurations and diagnostic result definitions for the T680 controller.
5. Storage: This section focuses on describing the usage of SPI NorFlash (not open for use yet) and FRAM built into the T680 controller.
6. System: This section focuses on describing the system information of the T680 controller, such as: product serial number, version number, operating voltage, PCB temperature and other information.
7. Misc: This section focuses on describing other information and control schemes of the T680 controller, such as: control of LEDs, operating status of KL15, enabling of 5V outputs etc.
8. Library Information: This section provides information about the version of this library function, version number, whether it is a released version, etc.

Input General

1.1.1. Location in the library file:

You can view the enumeration type definitions, structure definitions, library functions, etc. introduced below through Library Manager → JSHR Std Library → Input. As shown in Fig. 1 (CODESYS project screenshot of our T680 series controller as an example):

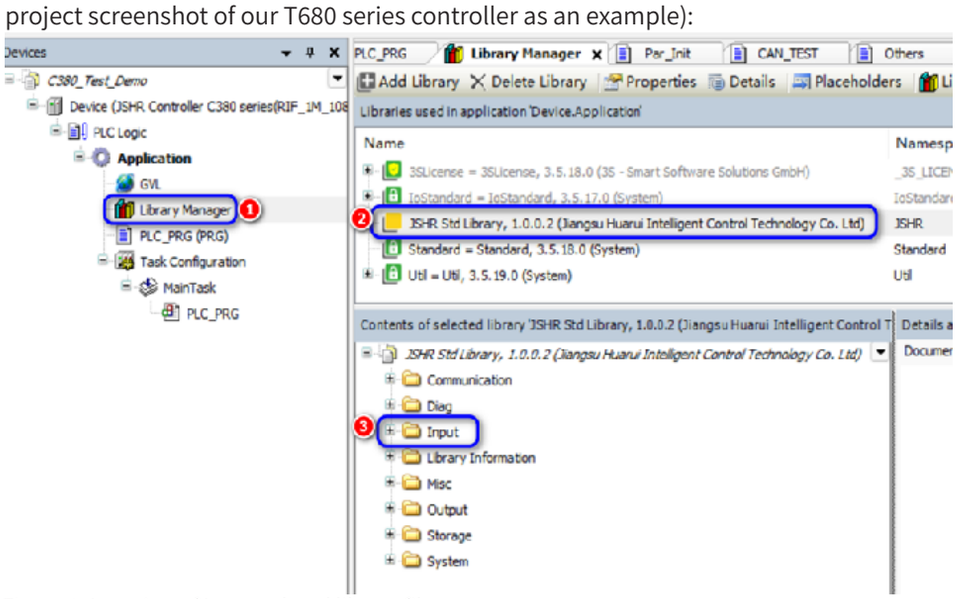


Figure 1: Location of Input-related library files

Enumeration Type Definition

1.1.1.1. IN_Channel_Type type

IN_Channel_Type defines a channel name for each input channel, which can be used to configure the channel parameters, read the input status, and read the input value in the program.
In T680, please refer to section 4.1 for the correspondence between physical port and channel name.

1.1.1.2. IN_Mode_Type type

IN_Mode_Type defines the signal type of the input channel. the T680 controller supports a variety of input signal types, and each input port supports two or more types of signals, see Table 1 for the specific types of signal inputs supported by the port.
The input types and enumeration variable values defined in IN_Mode_Type are as follows:

Table 1: IN_Mode_Type enumeration variable definitions

serial number	Input Type	Enumerated variable values	Type Description
1	IN_DIH	0	Highly effective switching inputs
2	IN_DIL	1	Low effective switching input
3	IN_AIV	2	Voltage type analog input
4	IN_AIC	3	Current type analog input
5	IN_AIR	4	Resistive analog input
6	IN_AIT	5	analog input for temperature measurement
7	IN_FIH	6	High effective frequency signal input
8	IN_FIL	7	Low effective frequency signal input
9	IN_ENCODE	8	Quadrature encoder input
10	IN_FIDC	9	Duty Cycle Input Mode
11	IN_RESERVE1	10	Reserved input mode 1
12	IN_RESERVE1	11	Reserved input mode 2

1.1.1.3. In_ValueFormat_Type type

In the T680 controller, you can get the values of input signals through a series of interface functions. For input signals, the T680 controller can provide two types of input signal values: 1) Digital values 2) Physical values.
Digital value: the value after analog-to-digital conversion (ADC);
Physical quantity value: the actual physical quantity value, e.g.: mA, mV, Ω, etc. are given visually.

Input General

IN_ValueFormat_Type defines the type of the input signal value.

Table 2: IN_ValueFormat_Type Enumeration Variable Definitions

serial number	Input Type	Enumerated variable values	Type Description
1	IN_DIGIT_VALUE	0	Digital values (internal use)
2	IN_PHY_VALUE	1	physical value
3	IN_RESERVE_VALUE	2	Type of reservation

For different types of input ports (In_MODE_Type), IN_DIGIT_VALUE and IN_PHY_VALUE correspond as follows:

Table 3: Correspondence between IN_Mode_Type and IN_ValueFormat_Type

serial number	In_MODE_Type	IN_PHY_VALUE	IN_DIGIT_VALUE
1	IN_DIH	High level: 1, remaining states: 0	Internal use
2	IN_DIL	Low level: 1, remaining states: 0	Internal use
3	IN_AIV	Outputs values in mV	Internal use
4	IN_AIC	Outputs values in uA	Internal use
5	IN_AIR	Outputs values in Ω	Internal use
6	IN_AIT	Outputs values in °C	Internal use
7	IN_FIH	Outputs values in units of 0.1 Hz	*No such interface*
8	IN_FIL	Outputs values in units of 0.1 Hz	*No such interface*
9	IN_ENCODE	Outputs a value in 2x edge steps.	*No such interface*
10	IN_FIDC	Output 0~10000, corresponding to 0%~100% duty cycle	*No such interface*
11	IN_RESERVE1	reserve	reserve
12	IN_RESERVE1	reserve	reserve

1.1.2. IN_TS structure

The IN_TS structure is used in Input related functions to get additional information about the

STRUCT **IN_TS**

InType_em	IN_Mode_Type	
Di_u8	BYTE	digital input result,valid:1,invalid:0
Ai_u16	WORD	ADC result
AiV_u16	WORD	voltage input result [mV]
AiC_u16	WORD	current input result [uA]
AiR_u16	WORD	resister input result [ohm]
AiT_i16	INT	temperature reselt [degC]
FiDC_u16	WORD	Duty cycle of frequency input result [0~10000]:[0%~100%]
Error_u16	WORD	channel error status,0-No error,others-error
Fi_u32	DWORD	frequency input [0.1Hz]
FiSum_u64	LWORD	frequency input sum
Enc_i64	LINT	Encode input result

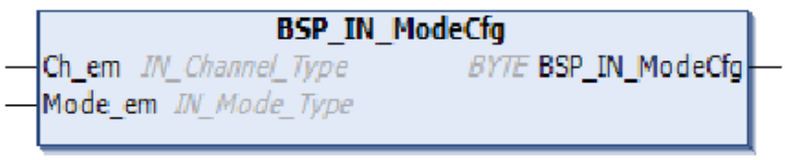
BSP_IN_Mode Cfg Function

1.1.1.1. BSP_IN_ModeCfg Function

1. Function Description

Used to set the type of signal input for the specified port.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	IN_Channel_Type	Input port channel name , see: Section 4.2 for details.
Mode_em	IN_Mode_Type	Input signal type, for details see: section 5.2.2.2

Output Variables:

output variable	Variable type	exegesis
BSP_IN_ModeCfg	BYTE	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 4: Channel (Ch_em) does not support this mode (Mode_em) 5: Driver parameter error

4. Precautions for use

1. It is important to check the type of the specified input port before using it, otherwise, the desired return value may not be obtained.

5. DIH/DIL Mode Configuration Group Correspondence

1. There is a group linkage relationship for DIH/DIL configuration, i.e., enabling DIH/DIL is always configured according to the group relationship in Table 4.

2. For example, when IN_1 is configured as DIH, IN_2~IN_11 will be configured as DIH mode by linkage.

Table 1: Correspondence between IN_Mode_Type and IN_ValueFormat_Type

form	Corresponding channel range	exegesis
1	IN_1 ~ IN_11	When DIH/DIL mode is configured by group members, it takes effect by group relationship
2	IN_12 ~ IN_22	ibid
3	IN_25 ~ IN_28	ibid
4	IN_29 ~ IN_32	ibid
5	IN_43 ~ IN_46	ibid
6	IN_47 ~ IN_51	ibid

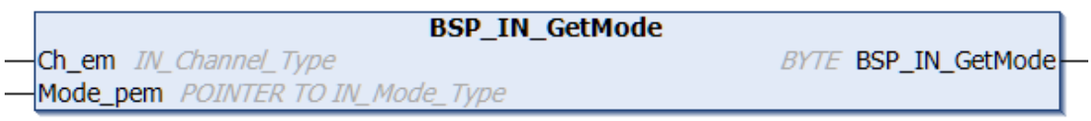
BSP_IN_Get Mode function

1.1.1.1. BSP_IN_Get Mode function

1. Function Description

Used to get the signal input type of the specified port.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	IN_Channel_Type	Input port channel name , see: Section 4.2 for details.
Mode_em	POINTER TO IN_Mode_Type	Return signal type, see: section 5.2.2.2 for details.

Output Variables:

output variable	Variable type	exegesis
BSP_IN_GetMode	BYTE	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 2: Pointer variable (ValueFormat_em) is NULL

4. Precautions for use

not have

BSP_IN_AI_Get Value function

1.1.1.1. BSP_IN_AI_Get Value function

1. Function Description

Used to acquire the AD value or physical value of the analog input from the specified port and store the acquisition result in the specified address space.

2. function block diagram

3. Input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	IN_Channel_Type	Input port channel name see: Section 4.2 for details.
ValueFormat_em	IN_ValueFormat_Type	The type of value expected from the specified port, see section 5.2.2.3 for details
Value_pu16	POINTER TO WORD	Pointer to the variable storing the result of the fetch

Output Variables:

output variable	Variable type	exegesis
BSP_IN_AI_GetValue	BYTE mode	Return Value: 0: read successfully 1: Channel (Ch_em) Error 2: Pointer variable (ValueFormat_em) is NULL 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this

4. Precautions for use

1. The type of input port must be configured before it can be used.
2. This function is used to read the port that is configured as an analog input.

5. usage example

```
(* Example declaration *)
AI1ADC: DWORD.
AI1Res: BYTE.
(* Example in ST *)
AI1Res : = JSHRBSP_IN_AI_GetValue(JSHR.IN_Channel_Type.
                                  JSHR.IN_ValueFormat_Type.IN_DIGIT_VALUE,
                                  ADR (AI1ADC)).
IF  AI1Res <> 0   THEN
    (* Error handling mechanisms *)
END_IF
```

BSP_IN_DI_Get Value function

1.1.1.1. BSP_IN_DI_Get Value function

1. Function Description

Used to acquire the switching input status from the specified port and store the acquisition result into the specified address space.

2. function block diagram

3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	IN_Channel_Type	Input port channel name see: Section 4.2 for details.
Valid_pu8	POINTER TO BYTE	Pointer to the variable storing the result of the fetch

Output Variables:

output variable	Variable type	exegesis
BSP_IN_DI_GetValue	BYTE mode	Return Value: 0: read successfully 1: Channel (Ch_em) Error 2: Pointer variable (Valid_pu8) is NULL 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this

4. Precautions for use

1. The type of input port must be configured before it can be used.
2. This function is used to read the port that is configured as a switching input.
3. For highly valid switching inputs, when accessing a high level, the acquisition result is 1, and other states are 0.
4. For low valid switching inputs, when accessing the low level, the acquisition result is 1, and other states are 0.

BSP_IN_DI_Water Mark Set Function

1.1.1.1. BSP_IN_DI_WaterMarkSet Function

1. Function Description

In the T680 controller, some analog input ports can be multiplexed as switching input ports. When multiplexed as a switching input port, the threshold voltage for the switching input must be set.

2. function block diagram

3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	IN_Channel_Type	Input port channel name see: Section 4.2 for details.
LowerStateThreshold_u16	WORD	Lower limit voltage, when the input voltage is lower than this parameter, it is recognised as low level. Default value: 1500 (corresponds to 1.5V)
UpperStateThreshold_u16	WORD	Upper limit voltage, when the input voltage is higher than this parameter, it is recognised as a high level. Default value: 2500 (corresponds to 2.5V)

Output Variables:

output variable	Variable type	exegesis
BSP_IN_DI_WaterMarkSet	BYTE	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 2: Lower limit value is less than the lowest limit value 3: Upper limit value greater than maximum limit value 4: Channel does not support this mode 5: It is not permissible for the lower limit value to be greater than the upper limit value

4. Precautions for use

1. The type of input port must be configured before it can be used.
2. This function applies to analog and switching multiplexed ports when configured as switching inputs.
3. The lower threshold voltage LowerStateThreshold_u16 shall be lower than the upper threshold voltage UpperStateThreshold_u16;
4. Lower limit voltage min: 150mV;
5. Upper limit voltage max:

BSP_IN_DI_Water Mark Set Function

4. Precautions for use
1. The type of input port must be configured before it can be used.
2. This function applies to analog and switching multiplexed ports when configured as switching inputs.
3. The lower threshold voltage LowerStateThreshold_u16 shall be lower than the upper threshold voltage UpperStateThreshold_u16;
4. Lower limit voltage min: 150mV;
5. Upper limit voltage max:

a) When the port type is 0~5V: the maximum value of upper limit voltage is 4850mV;
b) When the port type is 0~10V: the maximum value of upper limit voltage is 9850mV;
c) When the port type is 0~32V: the maximum value of upper limit voltage is 31450mV;

BSP_IN_GetExtensions function

1.1.1.1. BSP_IN_GetExtensions function
1. Function Description
Used to get different types of return values for the specified port.
2. function block diagram

3. input-output variable
Input Variables:

input variable	Variable type	exegesis
Ch_em	IN_Channel_Type	Input port channel name, see: Section 4.2 for details.
In_ps	POINTER TO IN_TS	Get the type of the value, see: Section 5.2.3 for details

Output Variables:

output variable	Variable type	exegesis
BSP_IN_GetExtensions	BYTE	Return Value: 0: read successfully 1: Channel (Ch_em) Error

4. Precautions for use
1. It is important to check the type of the specified input port before using it, otherwise, the desired return value may not be obtained.
2. For internal testing use only.

BSP_IN_ResetError function

1.1.1.1. BSP_IN_ResetError function
1. Function Description
Reset input channel error.
2. function block diagram

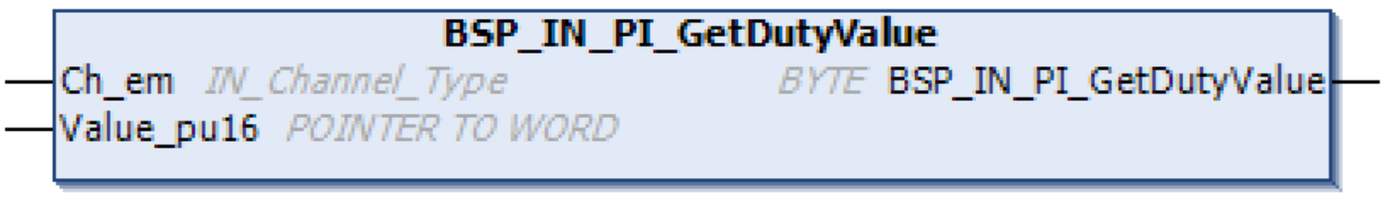
3. input-output variable
Input variables: none
Output Variables:

output variable	Variable type	exegesis
BSP_IN_ResetError	BYTE	Return Value: 0: Reset successful

4. Precautions for use
The T680 does not support this function at this time.

BSP_IN_PI_GetDutyValue function

1.1.1.1. BSP_IN_PI_GetDutyValue function
1. Function Description
Used to obtain the duty cycle of the frequency signal input of the specified channel.
2. function block diagram



3. input-output variable
Input Variables:

input variable	Variable type	exegesis
Ch_em	IN_Channel_Type	Input port channel name, see: Section 4.2 for details.
Value_pu16	POINTER TO WORD	Pointer to the variable storing the result of the fetch

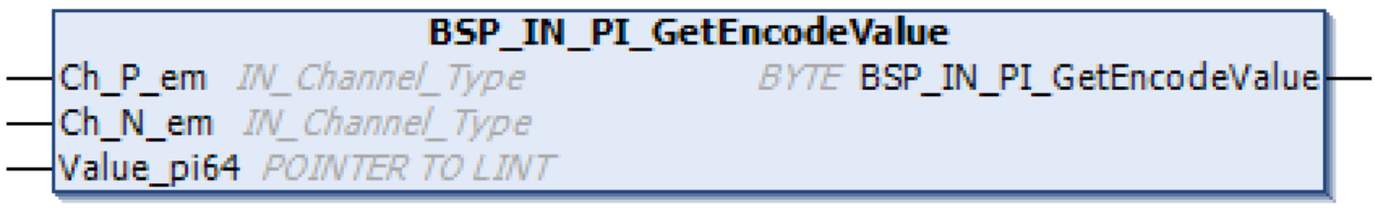
Output Variables:

output variable	Variable type	exegesis
BSP_IN_PI_GetDutyValue	BYTE	Return Value: 0: read successfully 1: Channel error 2: Pointer variable is NULL 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode

4. Precautions for use
1. Before use, you must verify that the port number is entered correctly;
2. Before use, you must verify that the specified channel is configured for frequency input mode.

BSP_IN_PI_GetEncodeValue function

1.1.1.1. BSP_IN_PI_GetEncodeDirector function
1. Function Description
Used to obtain the direction of the quadrature coded input signal.
2. function block diagram



3. input-output variable
Input Variables:

input variable	Variable type	exegesis
Ch_P_em	IN_Channel_Type	A-phase input port channel name see: Section 4.2 or Table 2 for details
Ch_N_em	IN_Channel_Type	B-phase input port channel name see: Section 4.2 or Table 2 for details
Value_pi8	POINTER TO BYTE	Pointer to the variable storing the result of the fetch

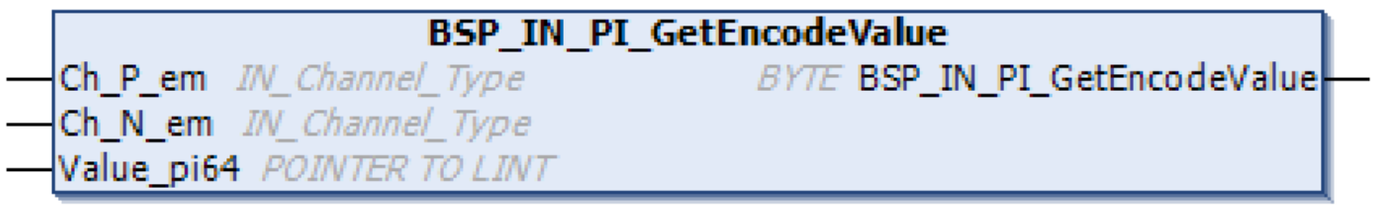
Output Variables:

output variable	Variable type	exegesis
BSP_IN_PI_GetEncodeDirector	BYTE	Return Value: 0: read successfully 1: Channel (Ch_P_em, Ch_N_em) error 2: pointer variable (Value_pi64) is NULL 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode

4. Precautions for use
1. In the T680 controller, there are 4 sets of quadrature coded input signals. When using them, you need to pay attention to whether the port matching is correct and the port type is configured correctly.
2. 0: positive direction, 1: negative direction.

BSP_IN_PI_GetEncodeDirector function

1.1.1.1. BSP_IN_PI_GetEncodeDirector function
1. Function Description
Used to obtain the direction of the quadrature coded input signal.
2. function block diagram



3. input-output variable
Input Variables:

input variable	Variable type	exegesis
Ch_P_em	IN_Channel_Type	A-phase input port channel name see: Section 4.2 or Table 2 for details
Ch_N_em	IN_Channel_Type	B-phase input port channel name see: Section 4.2 or Table 2 for details
Value_pi8	POINTER TO BYTE	Pointer to the variable storing the result of the fetch

Output Variables:

output variable	Variable type	exegesis
BSP_IN_PI_GetEncodeDirector	BYTE	Return Value: 0: read successfully 1: Channel (Ch_P_em, Ch_N_em) error 2: pointer variable (Value_pi64) is NULL 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode

4. Precautions for use
1. In the T680 controller, there are 4 sets of quadrature coded input signals. When using them, you need to pay attention to whether the port matching is correct and the port type is configured correctly.
2. 0: positive direction, 1: negative direction.

BSP_IN_PI_GetFreValue function

1.1.1.1. BSP_IN_PI_GetFreValue function
1. Function Description
Used to obtain the frequency value of the frequency input signal.
2. function block diagram

3. input-output variable
Input Variables:

input variable	Variable type	exegesis
Ch_em	IN_Channel_Type	Input port channel name, see: section 4.2 for details.
Value_pu32	POINTER TO DWORD	Pointer to the variable storing the result of the fetch

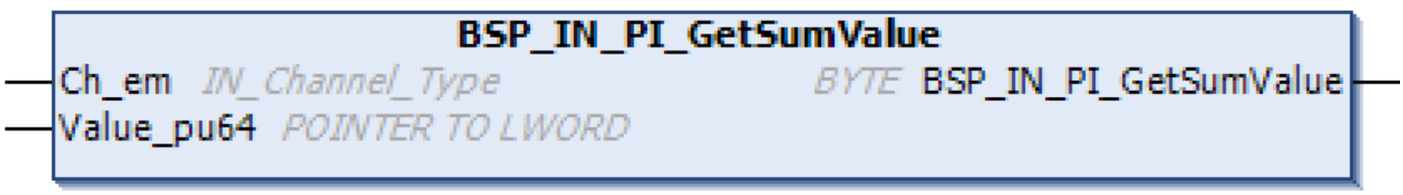
Output Variables:

output variable	Variable type	exegesis
BSP_IN_PI_GetFreValue	BYTE	Return Value: 0: read successfully 1: Channel error 2: Pointer variable is NULL 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode

4. Precautions for use
1. In the T680 controller, there are 10 frequency input signals. When using it, you need to pay attention to whether the port matching is correct and the port type is configured correctly.
2. The frequency value obtained here is the value after the actual frequency value x 10.

BSP_IN_PI_GetSumValue function

1.1.1.1. BSP_IN_PI_GetSumValue function
1. Function Description
Used to get the number of pulse inputs for the specified port.
2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	IN_Channel_Type	Input port channel name, see: section 4.2 for details.
Value_pu32	POINTER TO DWORD	Pointer to the variable storing the result of the fetch

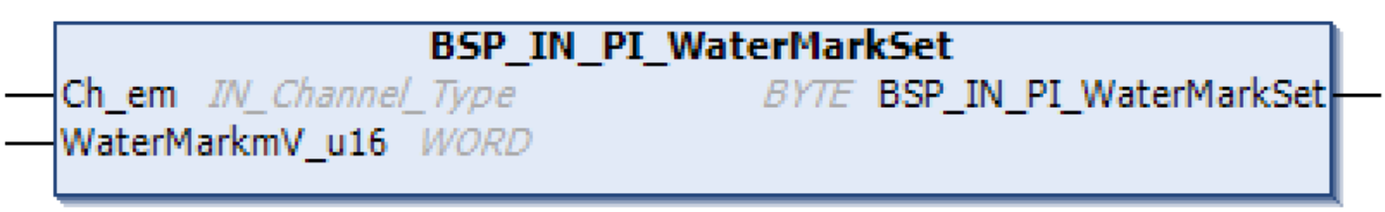
Output Variables:

output variable	Variable type	exegesis
BSP_IN_PI_GetFreValue	BYTE	Return Value: 0: read successfully 1: Channel error 2: Pointer variable is NULL 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode

4. Precautions for use
1. In the T680 controller, there are 10 frequency input signals. When using it, you need to pay attention to whether the port matching is correct.
2. The result obtained is the total number of pulses.

BSP_IN_PI_WaterMarkSet Function

1.1.1.1. BSP_IN_ResetError function
1. Function Description
Reset input channel error.
2. function block diagram



3. input-output variable
Input variables: none
Output Variables:

output variable	Variable type	exegesis
BSP_IN_ResetError	BYTE	Return Value: 0: Reset successful

4. Precautions for use

The T680 does not support this function at this time.

Output general

1.1.1. Location in the library file

You can view the enumeration type definitions, library functions, etc. described below via Library Manager → JSHR Std Library → Output. As shown in Figure 2:

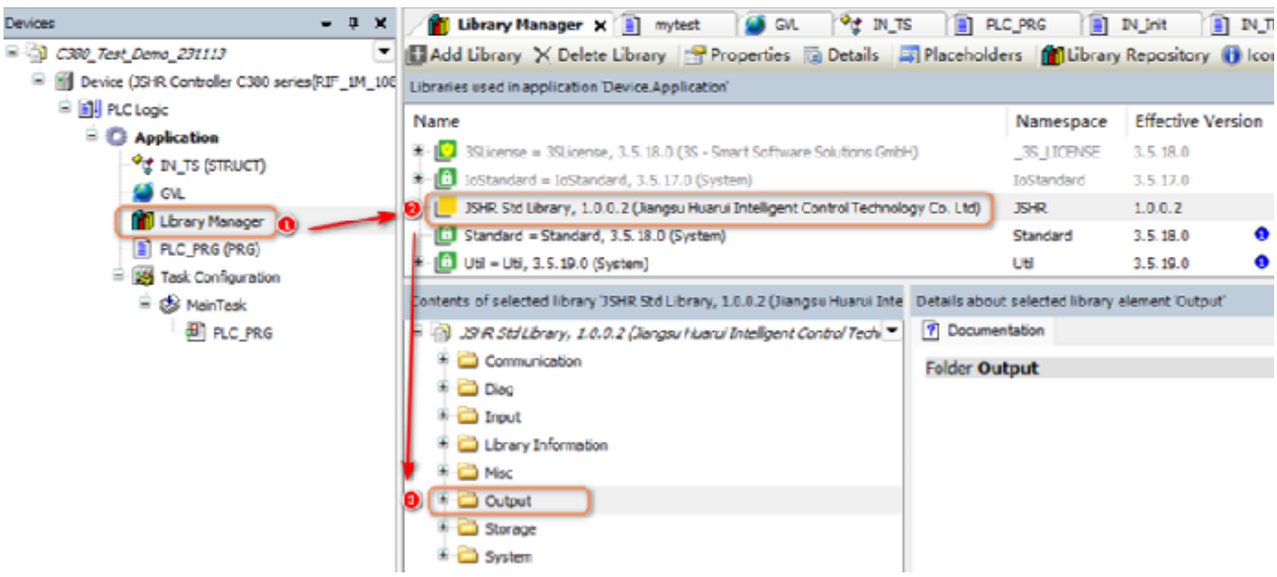


Figure 2: Location of Output-related library files

1.1.2. Enumerated Definitions

1.1.2.1. OUT_Channel_Type type

OUT_Channel_Type defines a channel name for each output channel, which can be used to configure the channel parameters, set the output status, and monitor the output status in the program.

In the T680, the physical port and channel names correspond as in Section 4.3

1.1.2.2. OUT_Mode_Type type

OUT_Mode_Type defines the signal type of the output channel. the T680 controller supports a variety of output signal types, see Table 5 for the specific signal output types supported by the port.

The output types and enumerated variable values defined in OUT_Mode_Type are as follows:

Table 6: OUT_Mode_Type Enumeration Variable Definitions

serial number	Input Type	Enumerated variable values	Type Description
1	OUT_DOH	0	High level switching output
2	OUT_DOL	1	Low level switching output
3	OUT_POH	2	High Level PWM Duty Cycle Outputs
4	OUT_POL	3	Low Level PWM Duty Cycle Outputs
5	OUT_POH_CL	4	High level PWM constant current output
6	OUT_POL_CL	5	Low level PWM constant current output
7	OUT_AOC	6	Current-type analog output (T680 does not have such a port)
8	OUT_AOV	7	Voltage type analog output
9	OUT_FO	8	Frequency output (T680 does not have such a port)
10	OUT_RESERVE1	9	Reserved output mode 1
11	OUT_RESERVE2	10	Reserved output mode 2

1.1.3. OUT_TS structure

In Output related functions, the OUT_TS structure is used to get additional information about the input channel. The structure is defined as follows topic.

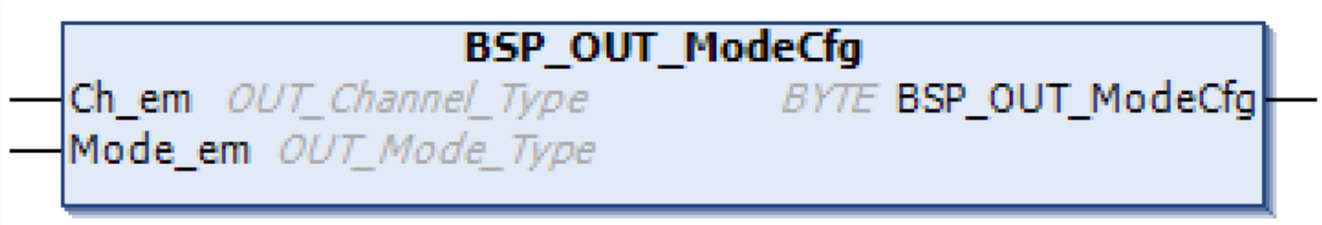
Output general

STRUCT **OUT_TS**

名称		
OUT_Mode_em	OUT_Mode_Type	channel mode
OUT_Set_DiagEnable_u8	BYTE	Out diagnostic used or not
PO_Set_DitherEnable_u8	BYTE	Out dither used or not
PO_Set_DitherFrequency_u8	BYTE	PO dither Frequency
PO_Set_DitherStep_u8	BYTE	PO dither STEP
OUT_Return_Code_u8	BYTE	Output return code
DO_FB_OutState_u8	BYTE	DO Feedback state
DO_Set_Output_u8	BYTE	DO Set Output
AOV_Set_Voltage_u16	WORD	AOV set voltage
AOC_Set_Current_u16	WORD	AOC set current
POCL_Set_Current_u16	WORD	POCL set current
PO_Set_Frequency_u16	WORD	PO set frequency
PO_Set_Duty_u16	WORD	PO set Duty
OUT_FB_Current_u16	WORD	Out feedback current
AOV_FB_Voltage_u16	WORD	AOV feedback voltage
OUT_DiagCode_u16	WORD	Out Error Code

BSP_OUT_ModeCfg Function

1.1.1.1. BSP_OUT_ModeCfg Function
1. Function Description
Sets the output port mode.
2. function block diagram



3. input-output variable
Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name, see: section 4.3 for details
Mode_em	OUT_Mode_Type	Port signal output type, see: Section 5.3.2.2 for details.

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_ModeCfg	BYTE	Return Value: 0: Successful setup 1: Channel error 4: Channel does not support this mode

4. Precautions for use
It is important to check the type of the specified output port before using it; otherwise, the desired state may not be output.

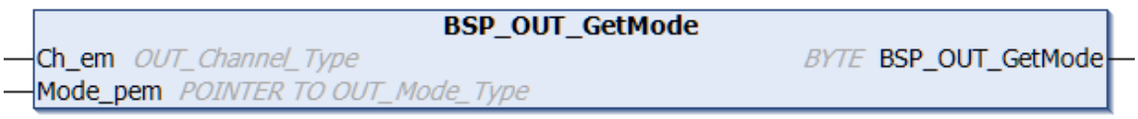
BSP_OUT_GetMode function

1.1.1.1. BSP_OUT_GetMode function

1. Function Description

Used to get the signal output type of the specified port.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name, see: section 4.3 for details
Mode_em	POINTER TO OUT_Channel_Type	Port signal output type, see: Section 5.3.2.2 for details.

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_GetMode	BYTE	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 2: Pointer variable (ValueFormat_em) is NULL

4. Precautions for use

not have

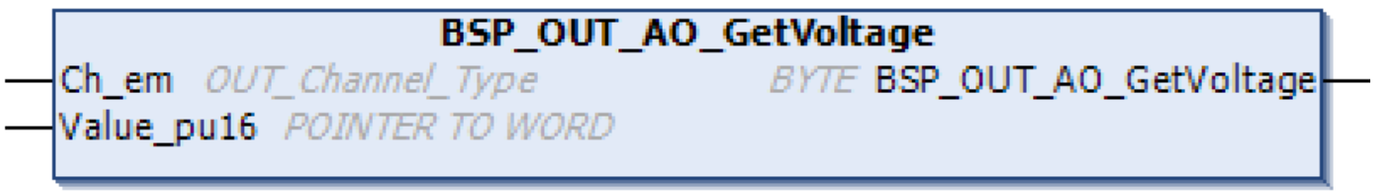
BSP_OUT_AO_GetVoltage function

1.1.1.1. BSP_OUT_AO_GetVoltage function

1. Function Description

Get the output voltage of the AO port.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name see: section 4.3 for details
Value_pu16	POINTER TO WORD	Pointer to the variable storing the result of the fetch

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_AO_GetVoltage	BYTE	Return Value: 0: Acquisition success

4. Precautions for use

1. Get results in mV.

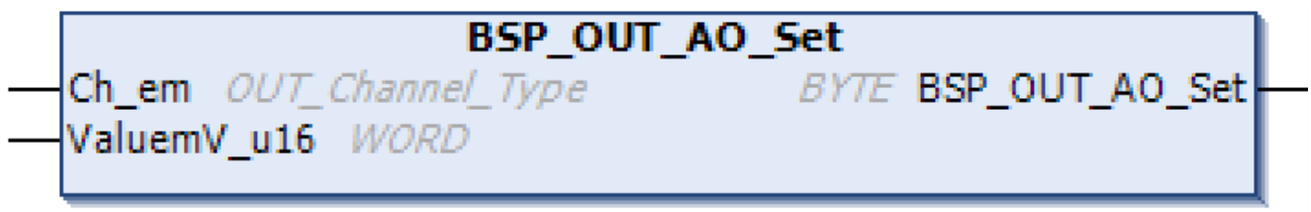
BSP_OUT_AO_Set Function

1.1.1.1. BSP_OUT_AO_Set Function

1. Function Description

Sets the output voltage of the AO port.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name see: section 4.3 for details
ValuemV_u16	Word	Sets the port output voltage in mV. Output range: 0~5000. Corresponding physical value: 0~5V

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_AO_GetVoltage	BYTE	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 2: Setting parameters greater than the upper limit

4. Precautions for use

1. Ch_em takes the correct value, otherwise there will be setup recognition.

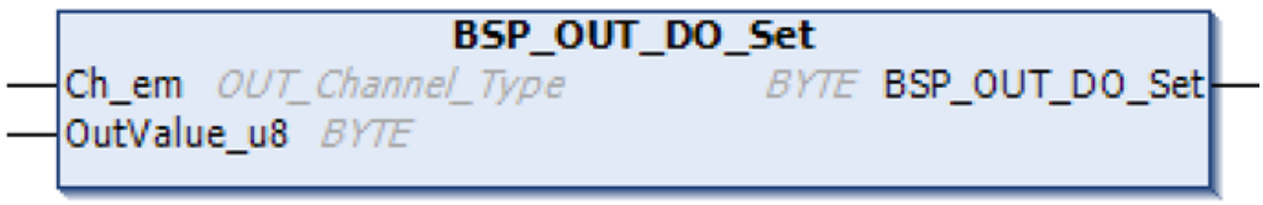
BSP_OUT_DO_Set Function

1.1.1.1. BSP_OUT_DO_Set Function

1. Function Description

Sets the output state of the switching output port.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name, see: section 4.3 for details
OutValue_u8	BYTE	Sets the state of the switching output port. 1: With output 0: no output

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_DO_Set	BYTE	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 2: Invalid set value (OutValue_u8) 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode 5: Channel failure, unable to output, please power off and restart or disable port diagnostics after releasing the port failure.

4. Precautions for use

1. Ch_em takes the correct value.

2. Before use, ensure that the port specified by this Ch_em is configured for switching output mode.

BSP_OUT_GetCurrent function

1.1.1.1. BSP_OUT_GetCurrent function

1. Function Description

Read the port output current value.

2. function block diagram

3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name, see: section 4.3 for details
Value_pu16	POINTER TO WORD	Pointer to the variable storing the result of the fetch Unit for obtaining results: mA

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_GetCurrent	BYTE	Return Value: 0: read successfully 1: Channel (Ch_em) Error 2: Pointer variable (Value_pu16) is NULL 4: Channel does not support this mode

4. Precautions for use

1. Ch_em takes the correct value.

BSP_OUT_GetDOStatus function

1.1.1.1. BSP_OUT_GetDOStatus function

1. Function Description

Reads the output status of the switching output port.

2. function block diagram

3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name, see: section 4.3 for details
Value_pu8	POINTER TO BYTE	Pointer to the variable storing the result of the fetch Get result = 0: no output from port Get result = 1: port has output

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_GetDOStatus	BYTE	Return Value: 0: read successfully 1: Channel (Ch_em) Error 2: Pointer variable (Value_pu8) is NULL 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode

4. Precautions for use

1. Before use, ensure that the port specified by this Ch_em is configured for switching output mode.

BSP_OUT_GetExtensions function

1.1.1.1. BSP_OUT_GetExtensions function

1. Function Description

Used to get different types of return values for the specified port.

2. function block diagram

3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Input port channel name, see: Section 4.3 for details.
Out_ps	POINTER TO OUT_TS	Get the type of the value, see: section 5.3.3 for details

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_GetExtensions	BYTE	Return Value: 0: read successfully 1: Channel (Ch_em) Error

4. Precautions for use

1. It is important to check the type of the specified input port before using it, otherwise, the desired return value may not be obtained.

2. For internal testing use only.

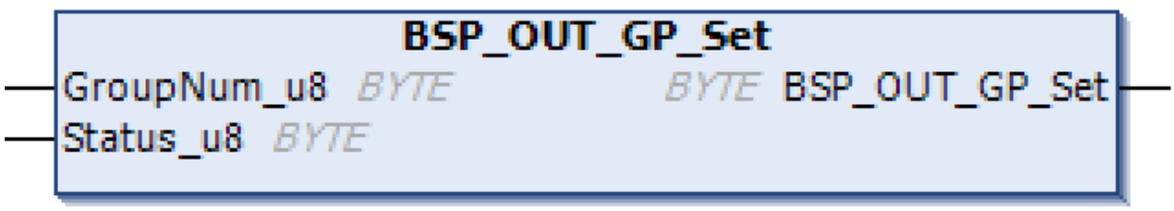
BSP_OUT_GP_Set function

1.1.1.1. BSP_OUT_GP_Set function

1. Function Description

Sets the output state of the group switch.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
GroupNum_u8	BYTE	Group switch serial number 0, 1
Status_u8	BYTE	Setting the Group Switch Status 0: Group switch off 1: Group switch on

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_GP_Set	BYTE	Return Value: 0: Successful setup Other values: setup failure

4. Precautions for use

The group switch correspondences in the T680 controller are as follows:

Group switch 0: OUT_1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 47.

Group switch 1: OUT_2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 48.

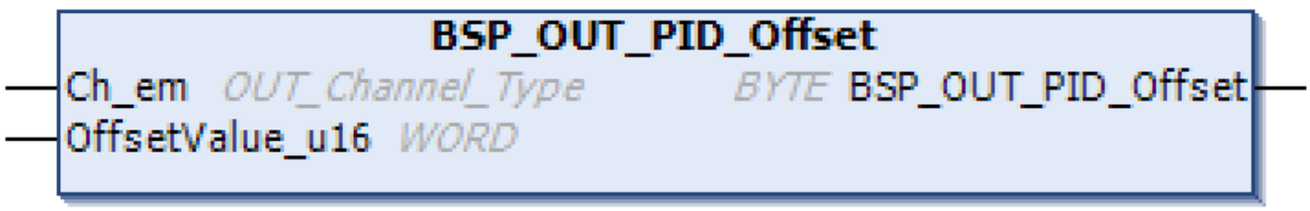
BSP_OUT_PID_Offset function

1.1.1.1. BSP_OUT_PID_Offset function

1. Function Description

Sets the offset parameter of the PID.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name, see: section 4.3 for details
OffsetValue_u16	WORD	Offset value (0~10000)

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_PID_Offset	WORD	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 2: Parameter (OffsetValue_u16) is greater than upper limit 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode

4. Precautions for use

It is important to check the type of the specified output port before using it; otherwise, the desired state may not be output.

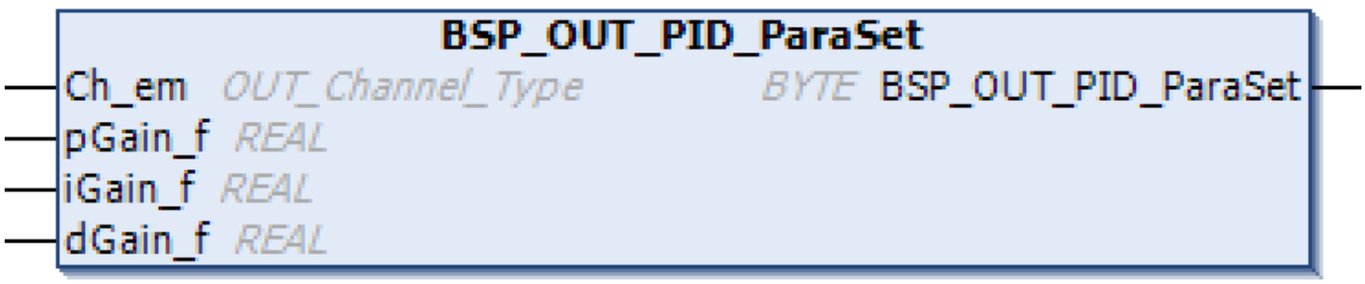
BSP_OUT_PID_ParaSet function

1.1.1.1. BSP_OUT_PID_ParaSet function

1. Function Description

Set the Kp, Ki, and Kd parameters for PID regulation.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name, see: section 4.3 for details
pGain_f	REAL	Set the Kp parameter for PID regulation, default value: 2.5
iGain_f	REAL	Set the Ki parameter for PID regulation, default value: 0.03
dGain_f	REAL	Set the Kd parameter for PID regulation, default value: 0

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_PID_ParaSet	BYTE	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode

4. Precautions for use

It is important to check the type of the specified output port before using it; otherwise, the desired state may not be output.

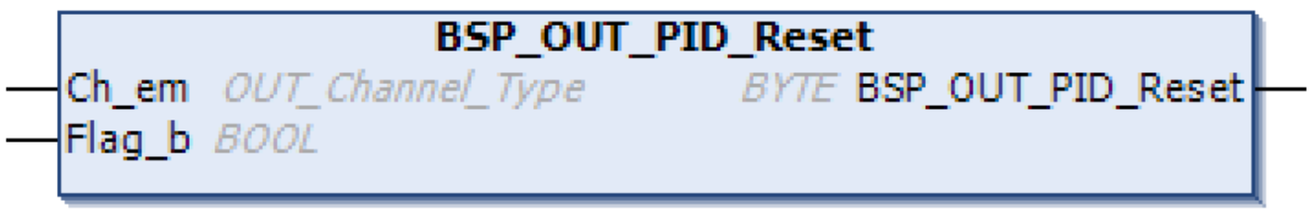
BSP_OUT_PID_Reset function

1.1.1.1. BSP_OUT_PID_Reset function

1. Function Description

PID regulation reset.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name, see: section 4.3 for details
Flag_b	BOOL	Flag_b=FALSE, no reset Flag_b=TRUE, reset

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_PID_Reset	BYTE	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 2: Invalid parameters 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode

4. Precautions for use

It is important to check the type of the specified output port before using it; otherwise, the desired state may not be output.

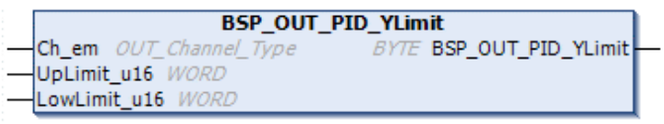
BSP_OUT_PID_YLimit function

1.1.1.1. BSP_OUT_PID_YLimit function

1. Function Description

PID output upper limit setting.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name, see: section 4.3 for details
UpLimit_u16	WORD	PID output upper limit value, default value: 10000
LowLimit_u16	WORD	PID output lower limit value, default value: 0

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_PID_YLimit	BYTE	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 2: The upper limit value is greater than the maximum upper limit of 10000 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode 5: The lower limit value is not allowed to be greater than the upper limit value

4. Precautions for use

It is important to check the type of the specified output port before using it; otherwise, the desired state may not be output.

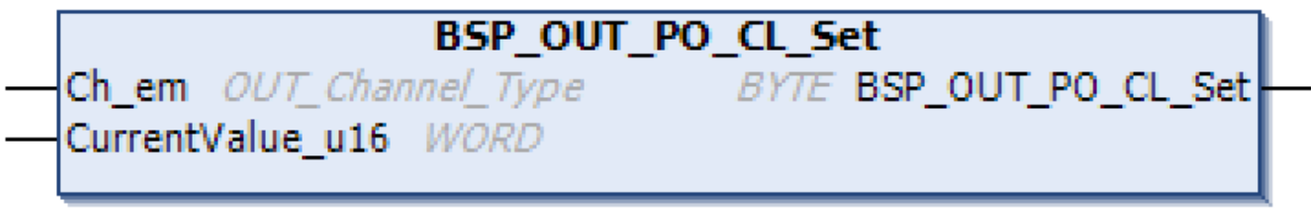
BSP_OUT_PO_CL_Set function

1.1.1.1. BSP_OUT_PO_CL_Set function

1. Function Description

Sets the PWM port output current.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name see: section 4.3 for details
CurrentValue_u16	WORD	Output current value in mA

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_PO_CL_Set	BYTE	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 2: CurrentValue_u16 is greater than the upper limit 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode

4. Precautions for use

It is important to check the type of the specified output port before using it; otherwise, the desired state may not be output.

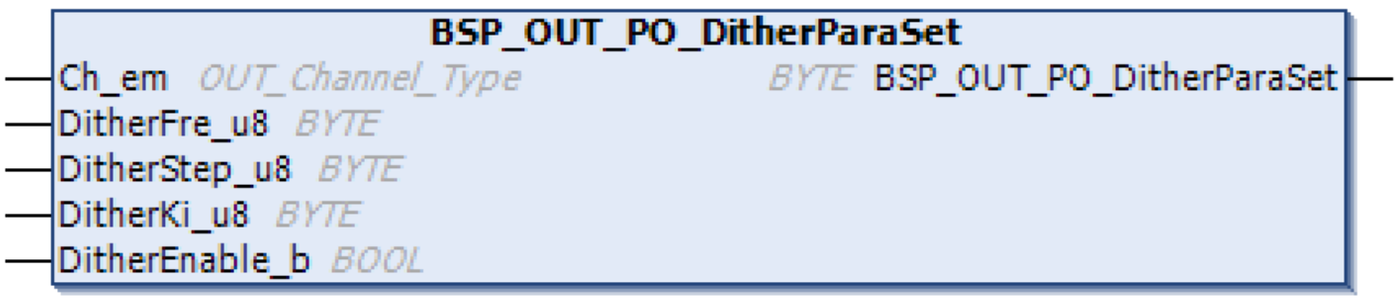
BSP_OUT_PO_DitherParaSet function

1.1.1.1. BSP_OUT_PO_DitherParaSet function

1. Function Description

Sets the dither parameter for the PWM port output.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name, see: section 4.3 for details
DitherFre_u8	BYTE	Tremor frequency: 50, 100, 120
DitherStep_u8	BYTE	Chattering step : 0~30
DitherKi_u8	BYTE	Chatter Ki value for current speed regulation
DitherEnable_b	BOOL	=TRUE Enables the chatter function =FALSE Disables the chatter function

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_PO_DitherParaSet	BYTE	Return Value: 0: Successful setup 1: Channel error 2: Chatter frequency is not supported 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this mode 5: Tremor amplitude value greater than the maximum upper limit of 30

4. Precautions for use

The T680 does not support this function.

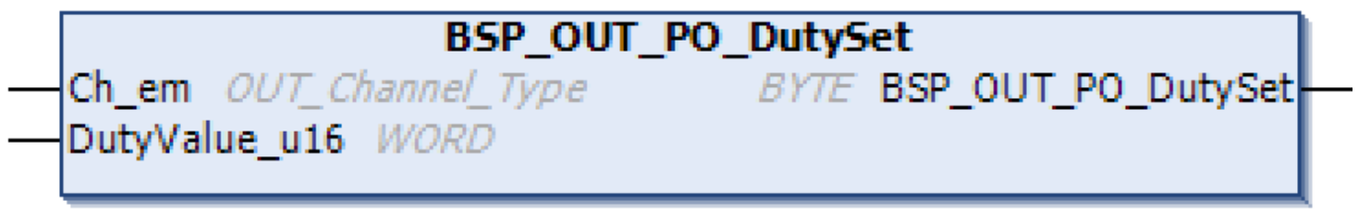
BSP_OUT_PO_DutySet function

1.1.1.1. BSP_OUT_PO_DutySet function

1. Function Description

Sets the duty cycle parameter of the PWM port.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel number, for details see: section 4.3
DutyValue_u16	WORD	Output Duty Cycle Output: 0~10000 Correspondence: 0~100% duty cycle

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_PO_DutySet	BYTE mode	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 2: Set duty cycle (DutyValue_u16) greater than the maximum limit 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this 5: Channel failure can not output, please troubleshooting

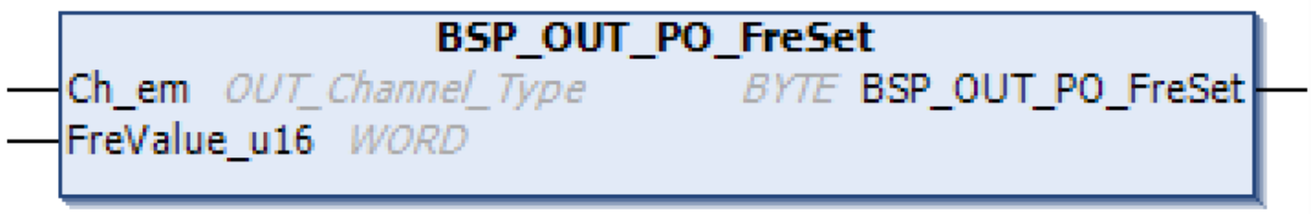
BSP_OUT_PO_FreSet Function

1.1.1.1. BSP_OUT_PO_FreSet Function

1. Function Description

Sets the frequency parameter of the PWM port.

2. function block diagram



3. Input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel number, for details see: section 4.3
FreValue_u16	WORD	Frequency range: 100~1000Hz

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_PO_FreSet	BYTE mode	Return Value: 0: Successful setup 1: Channel (Ch_em) Error 2: Set the frequency (FreValue_u16) greater than the maximum upper limit of 1000 or less than the minimum lower limit of 10 3: The mode of the channel configuration does not match the result of the acquired interface 4: Channel does not support this 5: The channel is turned on for chattering, no frequency change is allowed

4. Group correspondence

form	Corresponding channel range	exegesis
1	OUT_1~6	When configuring frequencies for group members, configure by group
2	OUT_7~14	ibid
3	out_15~18, 35, 36, 47, 48	ibid
4	OUT_37, 41	ibid
5	OUT_38, 42	ibid
6	OUT_39, 43	ibid
7	OUT_40, 44, 45	ibid
8	OUT_46	ibid

BSP_OUT_ResetError function

1.1.1.1. BSP_OUT_ResetError function

1. Function Description

Reset output channel error.

2. function block diagram

3. input-output variable

Input variables: none

Output Variables:

output variable	Variable type	exegesis
BSP_OUT_ResetError	BYTE	Return Value: 0: Reset successful

4. Precautions for use

The T680 does not support this function.

Communication general

You can view the enumeration type definitions, structure definitions, library functions, etc. Described below via Library Manager → JSHR Std Library → Communication. As shown in Figure 3:

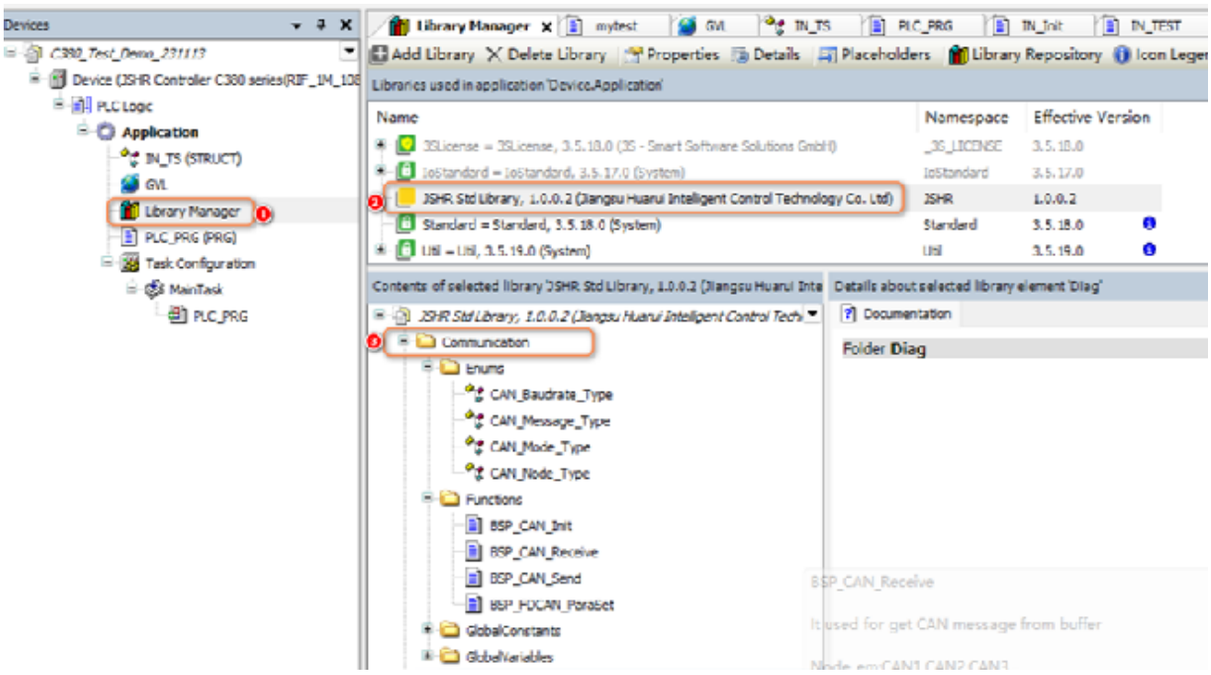


Figure 3: Location of Communication-related library files

1.1.1. Enumeration Type Definition

1.1.1.1. CAN_Baudrate_Type Type

Some of the baud rates supported by the CAN bus are defined here. The baud rates and enumeration values correspond as follows:

Table 7: Correspondence between baud rate and enumeration value

CAN bus baud rate	Enumerated variable names	enumerated value	note
10kbit/s	Can_10k	10	
50 kbit/s	Can_50k	50	
100 kbit/s	Can_100k	100	
125 kbit/s	Can_125k	125	
250 kbit/s	Can_250k	250	
500 kbit/s	Can_500k	500	
800 kbit/s	Can_800k	800	
1M bit/s	Can_1000k	1000	

1.1.1.2. CAN_Message_Type Type

The type of CAN bus message is defined here. The correspondence between message types and enumeration values is as follows:

Table 8: Correspondence between baud rate and enumeration value

Communication general

CAN message type	Enumerated variable names	enumerated value	note
Conventional CAN, standard frame	Std	1	
Conventional CAN, Extended Frame	Ext	2	
Conventional CAN, Standard Remote Frame	Std_Rmt	4	
Conventional CAN, Extended Remote Frame	Ext_Rmt	8	
CAN FD, standard frame	FD_Std	16	T680 is not supported at this time
CAN FD, extended frame	FD_Ext	32	T680 is not supported at this time

1.1.1.3. CAN_Mode_Type type

The communication type of the CAN bus is defined here. The correspondence between the communication type and the enumeration value is as follows:

Table 9: Correspondence between communication types and enumeration values

CAN communication type	Enumerated variable names	enumerated value	note
Conventional CAN	ClassicCAN	0	
CAN FD	CAN FD	1	T680 is not supported at this time

1.1.1.4. CAN_Node_Type type

The channel number of the CAN bus is defined here, and the correspondence between the channel number and the enumeration value is as follows topic.

Table 10: Correspondence between channel number and enumeration value

channel number	Enumerated variable names	enumerated value	note
CAN1 channel	CAN1	0	
CAN2 channel	CAN2	1	
CAN 3 channels	CAN3	2	
CAN 4 channels	CAN4	3	
CAN 5 channels	CAN5	4	T680 is not supported
CAN 6 channels	CAN6	5	T680 is not supported
CAN 7 channels	CAN7	6	T680 is not supported
CAN 8 channels	CAN8	7	T680 is not supported

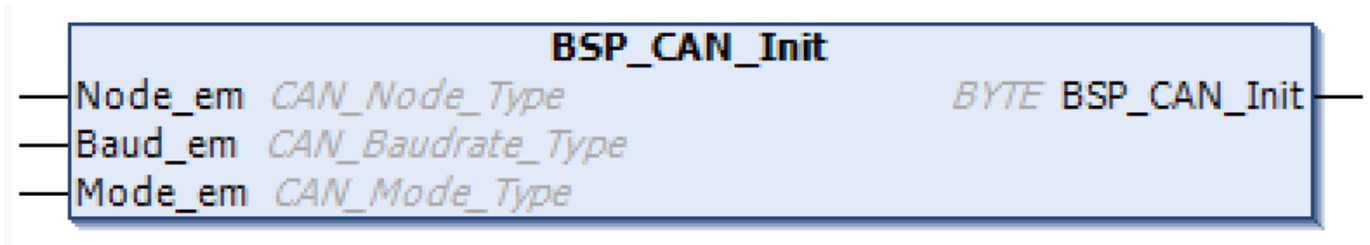
BSP_CAN_Init function

1.1.1.1. BSP_CAN_Init function

1. Function Description

CAN bus initialisation.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Node_em	CAN_Node_Type	CAN channel number, see: section 4.4 for details
Baud_em	CAN_Baudrate_Type	CAN bus baud rate, see: section 5.4.2.1.
Mode_em	CAN_Mode_Type	CAN operating mode, see: section 5.4.2.3 for more information.

Output Variables:

output variable	Variable type	exegesis
BSP_CAN_Init	BYTE	Return Value: 0: Successful setup 1: Channel (Node_em) error 2: Baud rate (Baud_em) not supported 3: External can bus abnormality, initialisation failure, please troubleshoot the can bus

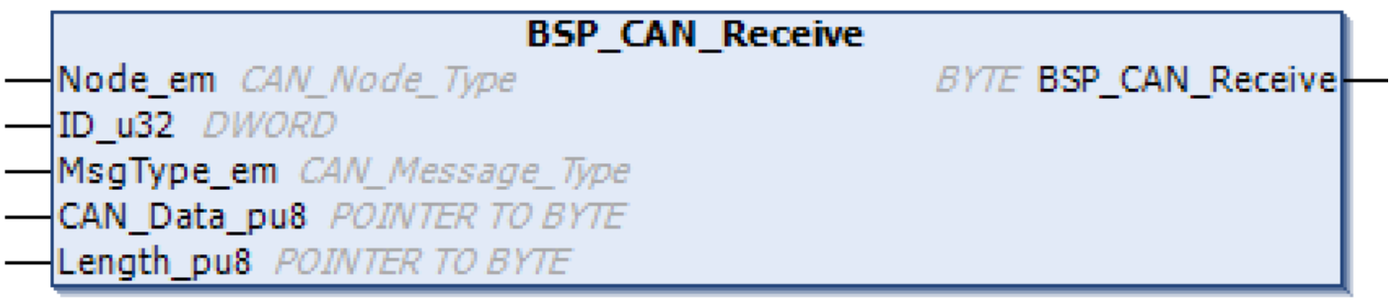
BSP_CAN_Receive function

1.1.1.1. BSP_CAN_Receive function

1. Function Description

This function provides the method of CAN bus data reception.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Node_em	CAN_Node_Type	CAN channel number, see: section 4.4 for details
ID_u32	DWORD	CANID number
MsgType_em	CAN_Message_Type	CAN message type, see: section 5.4.2.2 for details.
CAN_Data_pu8	POINTER TO BYTE	Pointer to the variable storing the result of the acquisition, here the CAN bus message information is acquired.
Length_pu8	POINTER TO BYTE	Pointer to the stored message length (DLC) variable

Output Variables:

output variable	Variable type	exegesis
BSP_CAN_Receive	BYTE	Return Value: 0: Received successfully 1: Channel (Node_em) error 2: This data was not received 3: Receive list full, support to receive up to 128 different frames (ID same type different also considered as different frames)

4. usage example

```
(* Example declaration *)
CAN1_Rx_Len_u8 : BYTE.
CAN1_RxDATA_au8 : ARRAY [0..7] OF BYTE.
(* Example in ST *)
IF 0 =
JSHR.BSP_CAN_Receive(JSHR.CAN_Node_Type.CAN1,16#201,JSHR.CAN_Message_Type.Std,ADR(CAN1_RxDATA_au8),
ADR(CAN1_Rx_Len_u8)) ADR(CAN1_Rx_Len_u8))THEN
      // receive OK,and handler data, User todo if received
END_IF
```

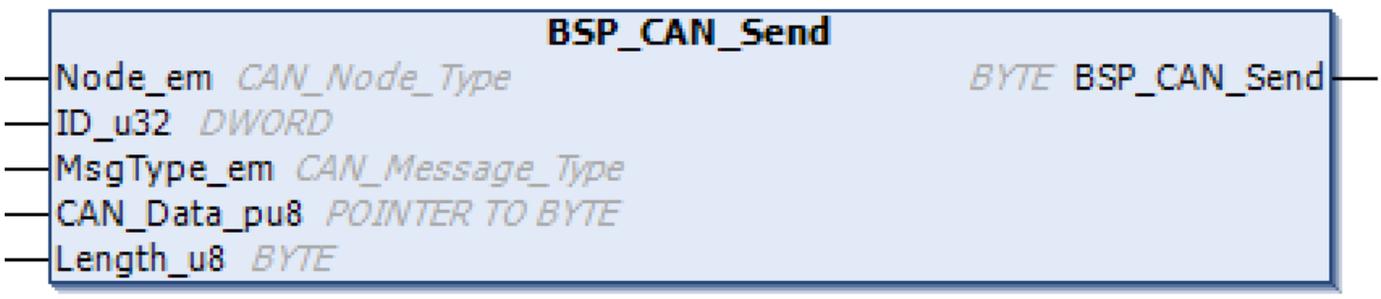
BSP_CAN_Send function

1.1.1.1. BSP_CAN_Send function

1. Function Description

This function provides the method of CAN bus data sending.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Node_em	CAN_Node_Type	CAN channel number, see: section 4.4 for details
ID_u32	DWORD	CANID number
MsgType_em	CAN_Message_Type	CAN message type, see: section 5.4.2.2 for details.
CAN_Data_pu8	POINTER TO BYTE	Pointer to the variable storing the result of the acquisition, here the CAN bus message information is acquired.
Length_pu8	BYTE	Message length DLC

Output Variables:

output variable	Variable type	exegesis
BSP_CAN_Send	BYTE	Return Value: 0: Sent successfully 1: Channel (Node_em) error 2: Frame type (MsgType_em) error 3: Length (Length_pu8) error 4: data buffer (CAN_Data_pu8) is empty 5: Send buffer is full, please reduce the amount of sending at the same time or increase the sending interval. 7: Hardware Busy

4. usage example

BSP_CAN_Send function

```
(* Example declaration *)
CAN1_TxDATA_au8 : ARRAY [0..7] OF BYTE.
(* Example in ST *)
CAN1_TxDATA_au8[0] := 16#01;
CAN1_TxDATA_au8[1] := 16#02;
CAN1_TxDATA_au8[2] := 16#03;
CAN1_TxDATA_au8[3] := 16#04;
CAN1_TxDATA_au8[4] := 16#05;
CAN1_TxDATA_au8[5] := 16#06;
CAN1_TxDATA_au8[6] := 16#07;
CAN1_TxDATA_au8[7] := 16#08;
IF 0 <>
JSHR.BSP_CAN_Send(JSHR.CAN_Node_Type.CAN1,16#181,JSHR.CAN_Message_Type.Ext,ADR(CAN1_TxDATA_au8),8)
THEN
      //Send Error Handling
    ;
END_IF
```

BSP_FDCAN_ParaSet Function

1.1.1.1. BSP_FDCAN_ParaSet Function

1. Function Description

This function provides a reference method for the CAN FD bus. (Note: The T680 does not support this function at this time)

2. function block diagram

3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Node_em	CAN_Node_Type	CAN channel number, see: section 4.4 for details
ESI_u8	BYTE	Error State Indicator Setting Enable =0: Do not enable the Error State Indicator. =1: Enable Error State Indicator
BRS_u8	BYTE	Set BitRate Switch Enable =0: Do not enable BitRate Switch =1: Enable BitRate Switch
BRSValue_u16	WORD	baud

Output Variables:

output variable	Variable type	exegesis
BSP_FDCAN_ParaSet	BYTE	Return Value: 0: Successful setup

Diagnose Related

1.1.1. Location in the library file

You can view the library functions etc. described below via Library Manager → JSHR Std Library → Diag. As shown in Figure 4:

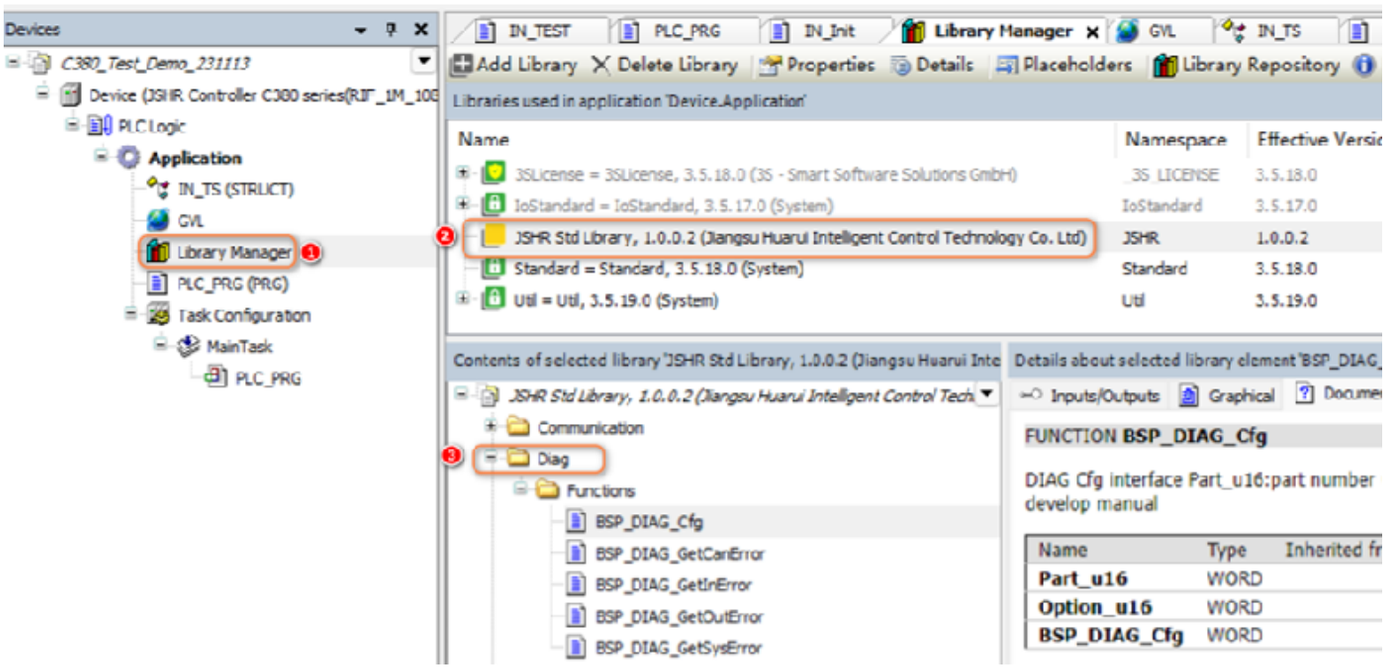


Figure 4: Location of Diag-related library files

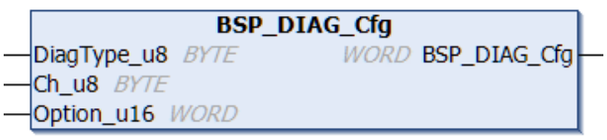
BSP_DIAG_Cfg Function

1.1.1.1. BSP_DIAG_Cfg Function

1. Function Description

Get the diagnostic information configured for the port. When the output is connected to a smaller load and the current is small, it will cause the port to fail, in which case the port enters the protection state and cannot be used for output, which can be used by configuring the diagnostic function of the corresponding channel to be turned off. Note that overload protection is not affected by this function.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
DiagType_u8	BYTE	Diagnostic Class Number 1: Enter INPUT 2: Output OUTPUT
Ch_u8	BYTE	Specific channel number for diagnostic class IN_1~IN_76 OUT_1~OUT_48
Option_u16	WORD	Diagnostic Options 0: Normal mode 1: Diagnostic model 2: Safe mode (default)

Output Variables:

output variable	Variable type	exegesis
BSP_DIAG_Cfg	WORD	Return Value: 0: Configuration successful 1: Channel error 2: Incorrect configuration value 3: Type numbering is not supported

4. Precautions for use

1. Normal Mode: In this mode, the internal pull-up diagnostic pulse of the output channel is turned off and the controller does not actively cut off the output.
2. Diagnostic Mode: In this mode, the internal pull-up diagnostic pulse of the output channel is enabled, and the controller does not actively cut off the output when a diagnostic error occurs.
3. Safe Mode: In this mode, the internal pull-up diagnostic pulse of the output channel is enabled and the controller will actively cut off the output when a diagnostic error occurs;
4. When this function is used to set the output channel diagnostic type, the following sets of correspondences exist for the pull-up diagnostic pulses used for internal diagnostics:

Group 1: OUT_1, 3, 5, 7, 9, 11
Group 2: OUT_2, 4, 6, 8, 10, 12
Group 3: OUT_13, 15, 17, 31, 35, 47
Group 4: OUT_14, 16, 18, 32, 36, 48
Group 5: OUT_19, 21, 23, 25, 27, 29
Group 6: OUT_20, 22, 24, 26, 28, 30

For example, according to the above relationship group, when OUT_1 is set to "Normal Mode", OUT_3, 5, 7, 9, 11 will be set to normal mode at the same time.

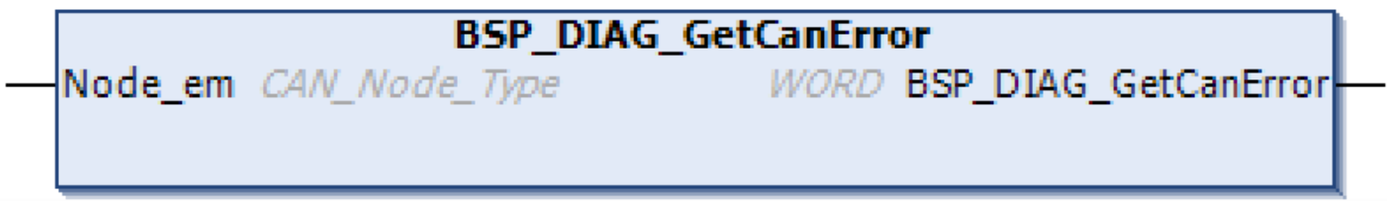
BSP_DIAG_GetCanError function

1.1.1.1. BSP_DIAG_GetCanError function

1. Function Description

Get the diagnostic information of the CAN channel.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Node_em	CAN_Node_Type	CAN channel number, see: section 4.4 for details

Output Variables:

output variable	Variable type	exegesis
BSP_DIAG_GetCanError	WORD	Return Value: 0: No error 1: bus off 2: Waring 4: Stuff error 16: Tx FIFO Error 32: Rx FIFO Error

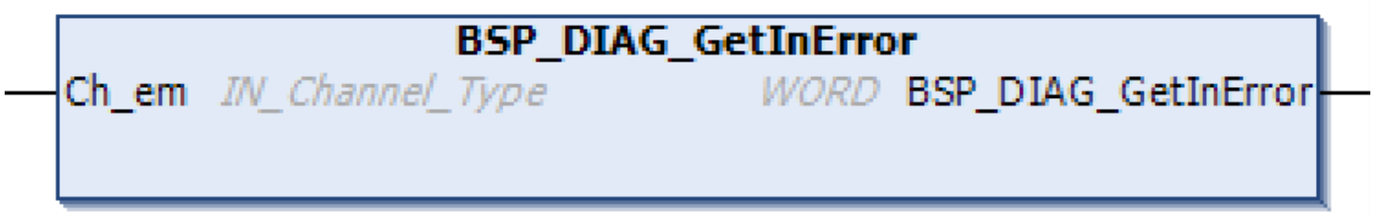
BSP_DIAG_GetInError function

1.1.1.1. BSP_DIAG_GetInError function

1. Function Description

Get the diagnostic information of the input channel.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	IN_Channel_Type	Input port channel name see: Section 4.2 for details.

Output Variables:

output variable	Variable type	exegesis
BSP_DIAG_GetInError	WORD	Return Value: 0: No faults 1: Current overload protection 2: Resistance greater than the measuring range 4: Resistive reference voltage out of range

BSP_DIAG_GetOutError function

1.1.1.1. BSP_DIAG_GetOutError function

1. Function Description

Get the diagnostic information of the output channel.

2. function block diagram

3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	OUT_Channel_Type	Output port channel name for details see: section 4.3.2.1

Output Variables:

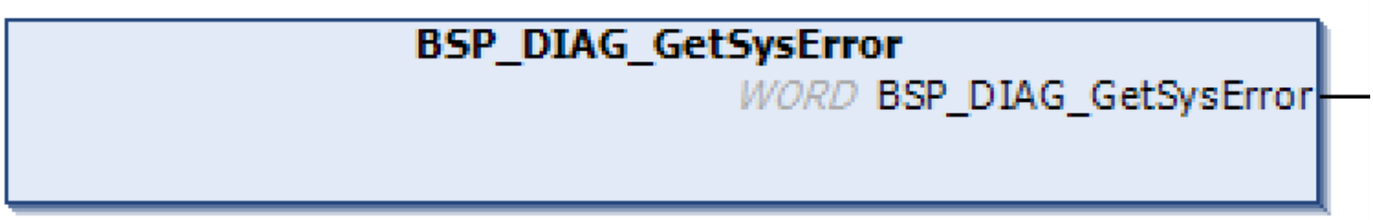
output variable	Variable type	exegesis
BSP_DIAG_GetOutError	WORD	Return Value: 0: read successfully 1: Short circuit to ground 2: Short circuit to power supply 4: Open Road 5: Short or open to ground (indistinguishable) 6: Short or open to power supply (indistinguishable) 16: Overload protection 32: Emergency stop activated, stop diagnostics 64: Deviation from set value (PWM constant current or AO)

4. Precautions for use

The channel does not allow output after detecting a fault, and the Clear Fault interface can be invoked after confirming that the fault has been removed. If a reset fault is always performed if a fault exists, repeated faults and resets can cause damage to the controller or load.

BSP_DIAG_GetSysError function

1.1.1.1. BSP_DIAG_GetSysError function
1. Function Description
Obtain diagnostic information about the system.
2. function block diagram



3. input-output variable
Input Variables: none

Output Variables:

output variable	Variable type	exegesis
BSP_DIAG_GetSysError	WORD	Return Value: 0: No faults 1: Component loading failure 2: Loss of product certification 4: Serial number does not match

Parameter storage general

1.1.1. Location in the library file

You can view the library functions etc. described below via Library Manager → JSHR Std Library → Storage. As shown in Figure 4:

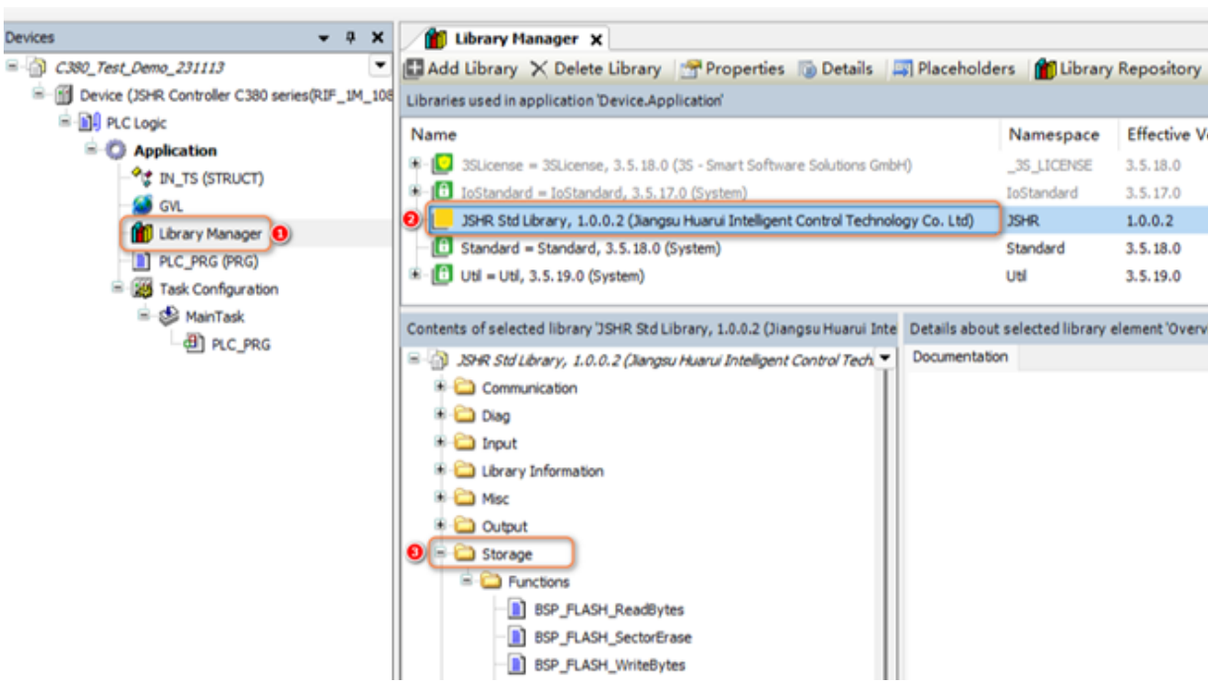


Figure 5: Location of Storage-related library files

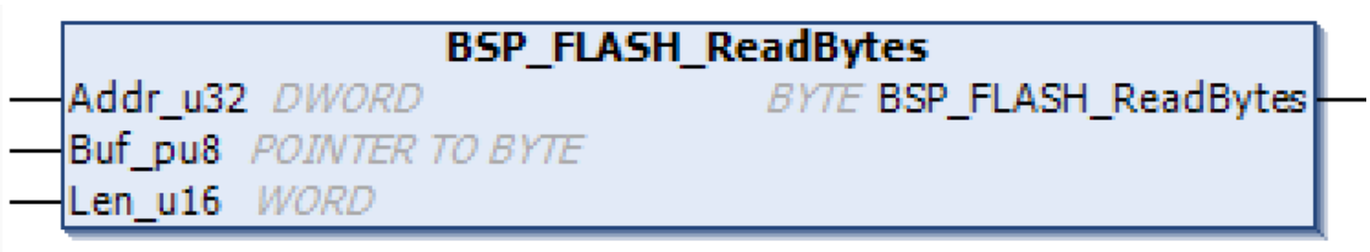
BSP_FLASH_ReadBytes function

1.1.1.1. BSP_FLASH_ReadBytes function

1. Function Description

Reads the contents of a file stored in Flash in the form of bytes. (Note: The T680 does not support this function at the moment)

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Addr_u32	DWORD	starting address Value range: 0~4*1024*1024-1
Buf_Pu8	POINTER TO BYTE	Pointer to the variable storing the result of the fetch
Len_u16	WORD	Read the length of the byte

Output Variables:

output variable	Variable type	exegesis
BSP_FLASH_ReadBytes	BYTE	Return Value: 0: read successfully Other values: read failure

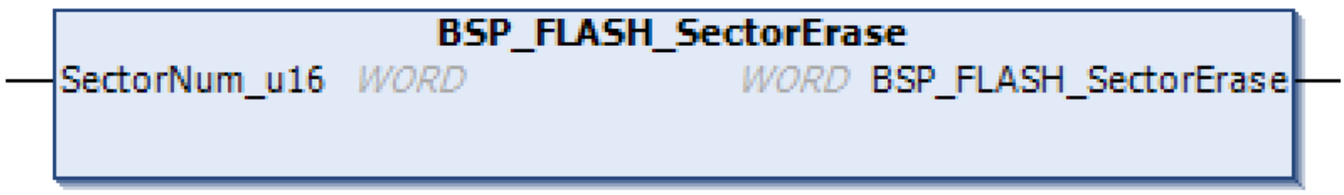
BSP_FLASH_SectorErase function

1.1.1.1. BSP_FLASH_SectorErase function

1. Function Description

Erases the contents of the specified partition in Flash. (Note: The T680 does not support this function for the time being.)

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
SectorNum_u16	WORD	Partition number 0~1023

Output Variables:

output variable	Variable type	exegesis
BSP_FLASH_SectorErase	WORD	Return Value: 0: Erase successful 4: Sector larger than maximum 5: Hardware busy

4. Precautions for use

1. In the T680 controller, there is 4MB of Flash divided into 1024 partitions. The correspondence between partition number and address is as follows:

The size of each sector is 4KB, and must be erased before writing, the maximum erase time per sector is 250ms, and the maximum number of erase times is 100000.

BSP_FLASH_WriteBytes function

1.1.1.1. BSP_FLASH_WriteBytes function

1. Function Description

Writes byte-ordered contents to the Flash memory. (Note: The T680 does not support this function at this time.)

2. function block diagram

3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Addr_u32	DWORD	memory start address Value range: 0~4*1024*1024-1
Buf_Pu8	POINTER TO BYTE	Pointer to the starting address of the memory contents
Len_u16	WORD	Length of storage bytes

Output Variables:

output variable	Variable type	exegesis
BSP_FLASH_WriteBytes	BYTE	Return Value: 0: Write successful 1: Hardware error 2: Hardware busy 3: Buffer pointer is NULL 4: Length overrun

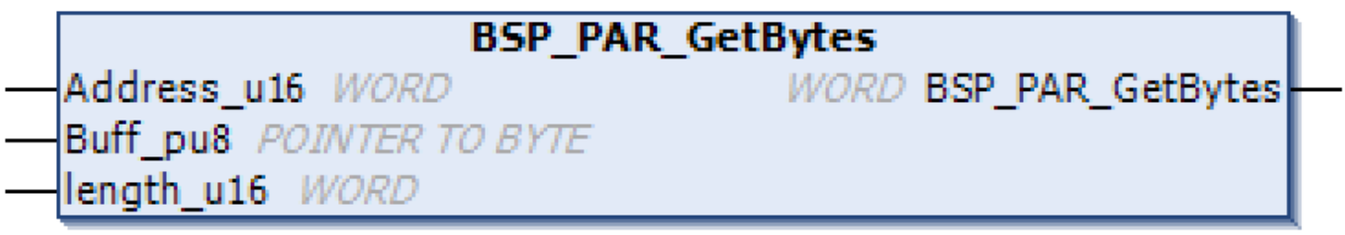
BSP_PAR_GetBytes function

1.1.1.1. BSP_PAR_GetBytes function

1. Function Description

Reads the contents of the file stored in FRAM as bytes.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Address_u16	WORD	starting address Value range: 0~2047
Buff_Pu8	POINTER TO BYTE	Pointer to the variable storing the result of the fetch
Length_u16	WORD	Read the length of the byte

Output Variables:

output variable	Variable type	exegesis
BSP_PAR_GetBytes	WORD	Return Value: 0: read successfully 1: Failure 2: Address overruns 3: Buffer is NULL 4: Length overrun

BSP_PAR_GetDWords function

1.1.1.1. BSP_PAR_GetDWords function

1. Function Description

Reads the contents of the file stored in FRAM as a DWORD.

2. function block diagram

3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Address_u16	WORD	starting address Value range: 0~2047
Buff_Pu32	POINTER TO DWORD	Pointer to the variable storing the result of the fetch
Length_u16	WORD	Number of DWORD variables to read

Output Variables:

output variable	Variable type	exegesis
BSP_PAR_GetDWords	WORD	Return Value: 0: read successfully 1: Failure 2: Address overruns 3: Buffer is NULL 4: Length overrun

BSP_PAR_GetInitStatus function

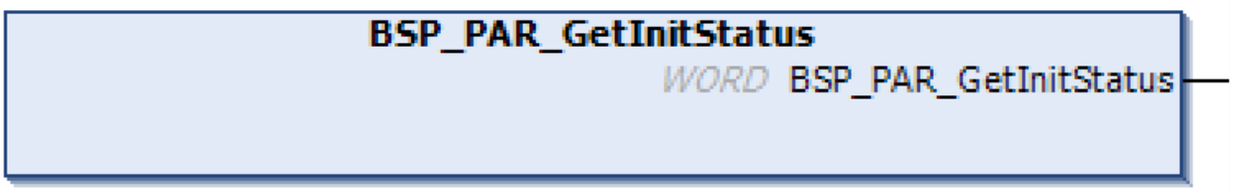
1.1.1.1. BSP_PAR_GetInitStatus function

1. Function Description

Reads the initialisation status of the FRAM parameter storage area.

To ensure that the data is valid, validation of the validity of the data stored in the FRAM needs to be performed at power-up.

2. function block diagram



3. input-output variable

Input Variables:

not have

Output Variables:

output variable	Variable type	exegesis
BSP_PAR_GetInitStatus	WORD	

The return value is defined as follows:

bit0 = 0: byte main memory checksum is correct

bit0 = 1: byte main memory checksum failure

bit1 = 0: byte backup storage area checksum is correct

bit1 = 1: byte Backup memory checksum failure

bit2 = 0: Word main memory checksum correctly

bit2 = 1: Word main memory checksum failure

bit3 = 0: Word backup storage area checksum correctly

bit3 = 1: Word backup store checksum failure

bit4 = 0: Real main memory checksum is correct

bit4 = 1: Real main memory checksum failure

bit5 = 0: Real backup storage area checksum correctly

bit5 = 1: Real backup store checksum failure

bit6: Reservations

bit7: Reservations

bit8 = 0: voltage normal

bit8 = 1: Voltage abnormality while reading Byte main memory area

bit9 = 0: voltage normal

bit9 = 1: Voltage exception when reading byte backup storage area

bit10 = 0: voltage normal

bit10 = 1: Voltage abnormality while reading WORD main memory area

bit11 = 0: voltage normal

bit11 = 1: Voltage abnormality while reading WORD backup storage area

bit12 = 0: voltage normal

bit12 = 1: Voltage abnormality while reading REAL main memory area

bit13 = 0: voltage normal

bit13 = 1: Voltage abnormality while reading REAL backup storage area

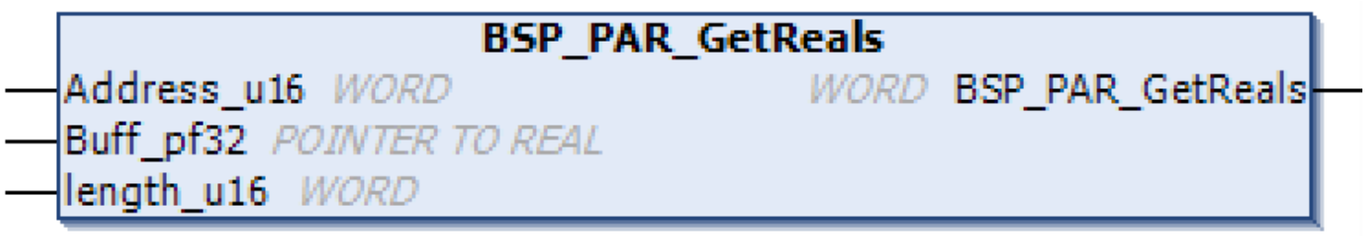
BSP_PAR_GetReals function

1.1.1.1. BSP_PAR_GetReals function

1. Function Description

Reads the contents of a file stored in FRAM as Real.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Address_u16	WORD	starting address Value range: 0~2047
Buf_Pf32	POINTER TO REAL	Pointer to the variable storing the result of the fetch
Length_u16	WORD	The number of REAL variables to read

Output Variables:

output variable	Variable type	exegesis
BSP_PAR_GetReals	WORD	Return Value: 0: read successfully 1: Failure 2: Address overruns 3: Buffer is NULL 4: Length overrun

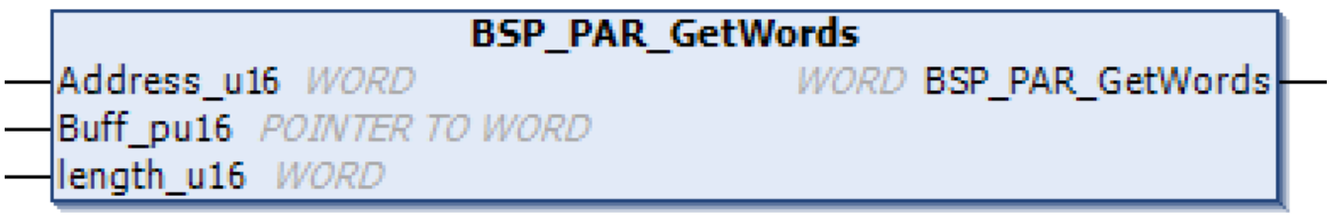
BSP_PAR_GetWords function

1.1.1.1. BSP_PAR_GetWords function

1. Function Description

Reads the contents of a file stored in FRAM as word.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Address_u16	WORD	starting address Value range: 0~2047
Buf_Pu16	POINTER TO WORD	Pointer to the variable storing the result of the fetch
Length_u16	WORD	Number of WORD variables to read

Output Variables:

output variable	Variable type	exegesis
BSP_PAR_GetReals	WORD	Return Value: 0: read successfully 1: Failure 2: Address overruns 3: Buffer is NULL 4: Length overrun

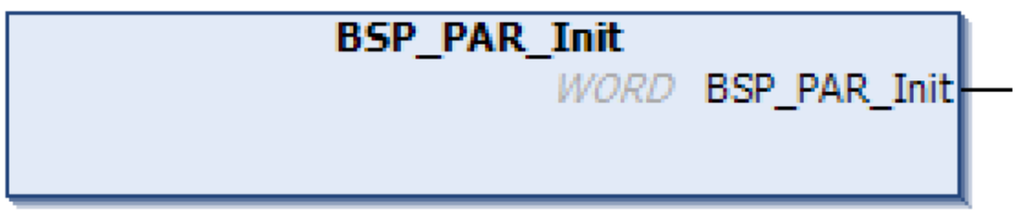
BSP_PAR_Init Function

1.1.1.1. BSP_PAR_Init function

1. Function Description

Parameter management initialisation, firmware power-up will be initialised, such as parameter management initialisation fails, you can re-call this interface in APP to initialise again, if it still fails, the saved parameters are invalid, please do not use, and prompt the user.

2. function block diagram



3. input-output variable

Input Variables: None

Output Variables:

output variable	Variable type	exegesis
BSP_PAR_Init	WORD	

Return value reference: section 4.6.2.6 Return value of BSP_PAR_GetInitStatus.

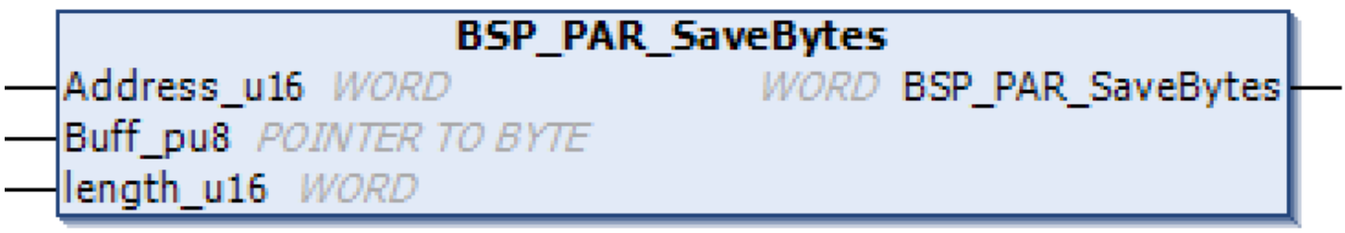
BSP_PAR_SaveBytes function

1.1.1.1. BSP_PAR_SaveBytes function

1. Function Description

Writes BYTES contents into FRAM storage.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Address_u16	WORD	starting address Value range: 0~2047
Buff_Pu8	POINTER TO BYTE	Pointer to the starting address of the memory contents
Length_u16	WORD	Store the length of BYTES

Output Variables:

output variable	Variable type	exegesis
BSP_PAR_SaveBytes	WORD	Return Value: 0: Write successful 1: Failure 2: Address overruns 3: Buffer is NULL 4: Length overrun 5: Parameter management initialisation failure, write invalid 6: Write Protect

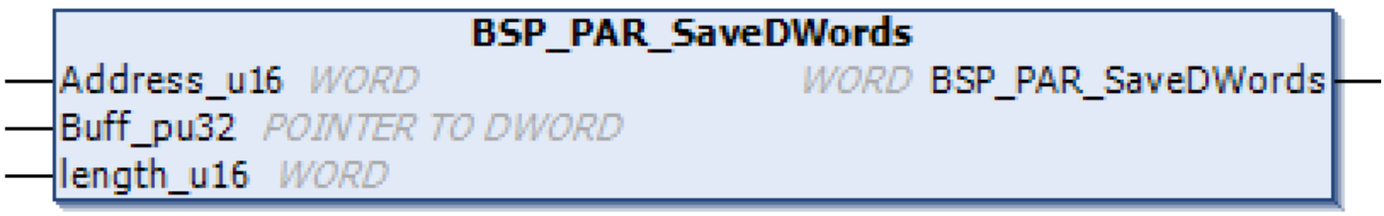
BSP_PAR_SaveDWords function

1.1.1.1. BSP_PAR_SaveDWords function

1. Function Description

Writes the contents of DWORDS into FRAM storage.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Address_u16	WORD	starting address Value range: 0~2047
Buff_Pu32	POINTER TO DWORD	Pointer to the starting address of the memory contents
Length_u16	WORD	Stores the length of the DWORDS

Output Variables:

output variable	Variable type	exegesis
BSP_PAR_SaveDWords	WORD	Return Value: 0: Write successful 1: Failure 2: Address overruns 3: Buffer is NULL 4: Length overrun 5: Parameter management initialisation failure, write invalid 6: Write Protect

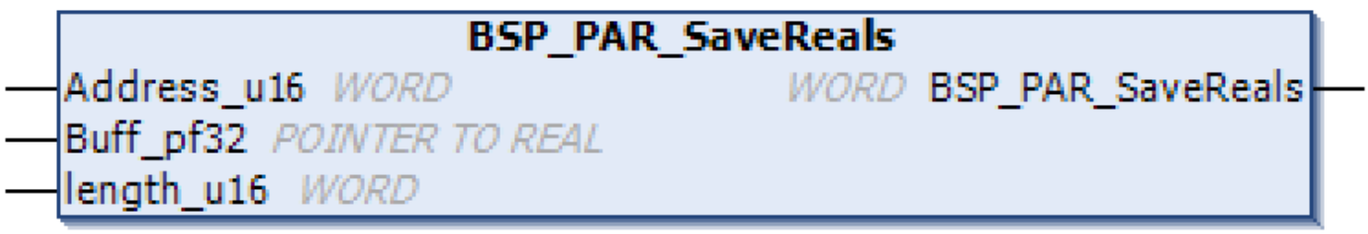
BSP_PAR_SaveReals function

1.1.1.1. BSP_PAR_SaveReals function

1. Function Description

Write the contents of REALS to the FRAM storage.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Address_u16	WORD	starting address Value range: 0~2047
Buff_Pu32	POINTER TO REAL	Pointer to the starting address of the memory contents
Length_u16	WORD	Store the length of the REALS

Output Variables:

output variable	Variable type	exegesis
BSP_PAR_SaveReals	WORD	Return Value: 0: Write successful 1: Failure 2: Address overruns 3: Buffer is NULL 4: Length overrun 5: Parameter management initialisation failure, write invalid 6: Write Protect

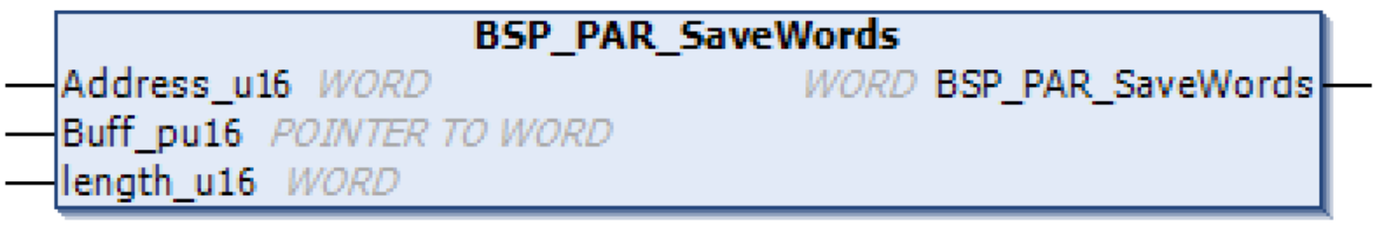
BSP_PAR_SaveWords function

1.1.1.1. BSP_PAR_SaveWords function

1. Function Description

Writes the contents of WORDS into FRAM storage.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Address_u16	WORD	starting address Value range: 0~2047
Buff_Pu16	POINTER TO WORD	Pointer to the starting address of the memory contents
Length_u16	WORD	Storing the length of WORDS

Output Variables:

output variable	Variable type	exegesis
BSP_PAR_SaveWords	WORD	Return Value: 0: Write successful 1: Failure 2: Address overruns 3: Buffer is NULL 4: Length overrun 5: Parameter management initialisation failure, write invalid 6: Write Protect

System general

1.1.1. Location in the library file

You can view the library functions etc. described below by Library Manager → JSHR Std Library → System. As shown in Figure 6:

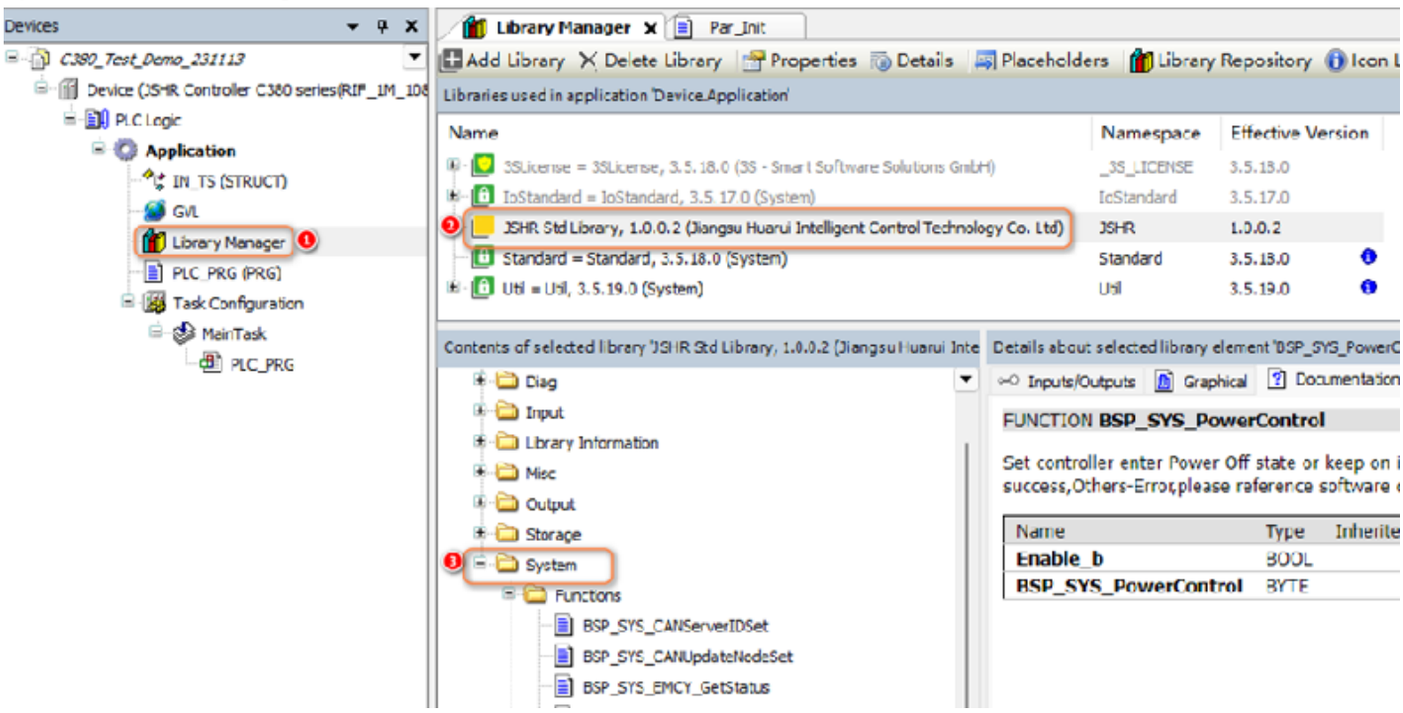


Figure 6: Location of System-related library files

1.1.2. Structure Type Definition

1.1.2.1. SYS_Product_Information Type

Product information for the T680 controller is defined here.

Structures are defined as follows:

```
TYPE SYS_Product_Information.
STRUCT
CustomID_au8 : ARRAY0[.7] OF BYTE; (* Customer ID number)*
JshrID_au8 : ARRAY0[.7] OF BYTE; (* Product ID number)*
ProductSN_au8 : ARRAY0[.15] OF BYTE; (* ProductSN number *)
FactoryCode_au8 : ARRAY0[.7] OF BYTE; (* Factory ID number)*
ProductType_au8 : ARRAY0[.7] OF BYTE; (* Product Type *)
END_STRUCT
END_TYPE
```

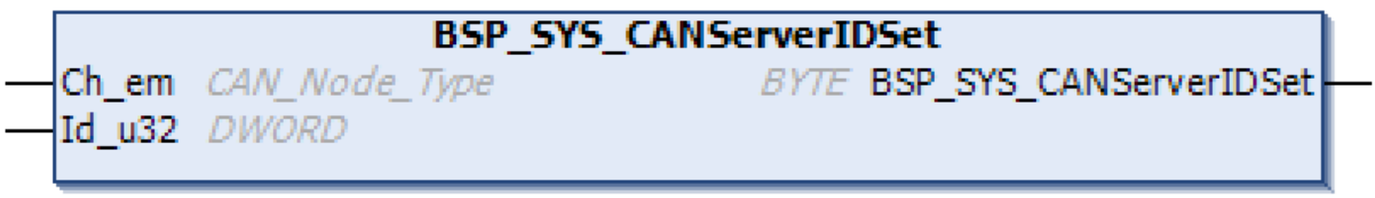
BSP_SYS_CANServerIDSet Function

1.1.1.1. BSP_SYS_CANServerIDSet Function

1. Function Description

Set the CAN communication service node number for multiple controllers on the same bus for online debugging.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	CAN_Node_Type	CAN online commissioning channel number, see: section 4.4 for details
Id_u32	DWORD	CANID for CAN online commissioning (1~127)

Output Variables:

output variable	Variable type	exegesis
BSP_SYS_CANServerIDSet	BYTE	Return Value: 0: Successful setup

4. Precautions for use

The CAN ID of the GateWay of codesys cannot be the same as the CAN ID of the controller.

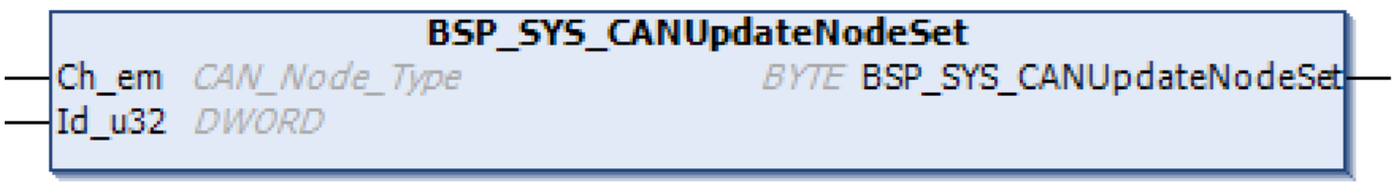
BSP_SYS_CANUpdateNodeSet function

1.1.1.1. BSP_SYS_CANUpdateNodeSet function

1. Function Description

Set the CAN channel number and communication ID used for program upgrade using CAN communication.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_em	CAN_Node_Type	CAN channel number, see: section 4.4 for details
Id_u32	DWORD	CANID for upgrading

Output Variables:

output variable	Variable type	exegesis
BSP_SYS_CANUpdateNodeSet	BYTE	Return Value: 0: Successful setup Other values: setup failure

4. Precautions for use

This feature is not available at this time.

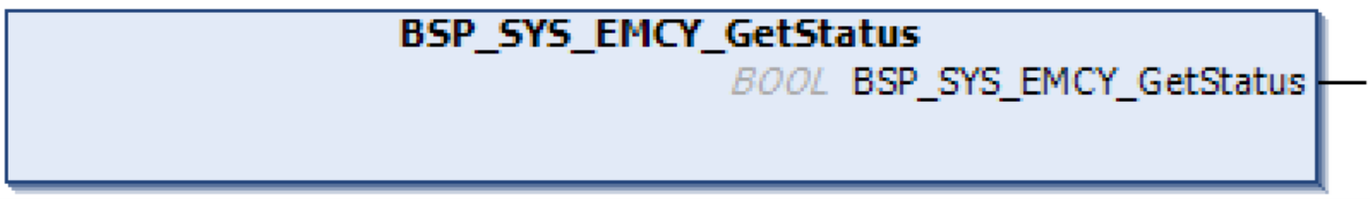
BSP_SYS_EMCY_GetStatus function

1.1.1.1. BSP_SYS_EMCY_GetStatus function

1. Function Description

Get the status of the controller's external emergency stop switch.

2. function block diagram



3. input-output variable

Input Variables: None

Output Variables:

output variable	Variable type	exegesis
BSP_SYS_EMCY_GetStatus	BYTE	Return Value: 0: Emergency stop not activated 1: Emergency stop activated, all outputs cut off

4. Precautions for use

None.

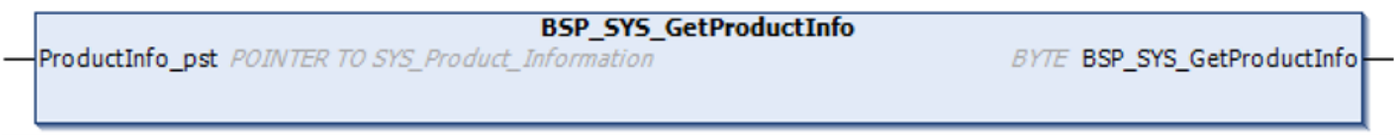
BSP_SYS_GetProductInfo function

1.1.1.1. BSP_SYS_GetProductInfo function

1. Function Description

Obtain product information for the controller, for more information refer to: Section 4.7.2.1.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
ProductInfo_pst	POINTER TO SYS_Product_Information	Pointer to the starting address of the memory contents

Output Variables:

output variable	Variable type	exegesis
BSP_SYS_GetProductInfo	BYTE	Return Value: 0: Acquisition success 2: The variable pointer is NULL

BSP_SYS_GetRtcTime function

1.1.1.1. BSP_SYS_GetRtcTime function

1. Function Description

Gets the RTC time in the controller, which is the number of seconds since 0:0:0 on 1 January 1970.

(Note: the T680 does not support this function)

2. function block diagram

3. input-output variable

Input Variables: None

Output Variables:

output variable	Variable type	exegesis
BSP_SYS_GetRtcTime	DWORD	This value is the number of seconds since 0:0:0 on 1 January 1970

BSP_SYS_GetTemperatureValue function

1.1.1.1. BSP_SYS_GetTemperatureValue function

1. Function Description

Get the internal temperature of the controller.

2. function block diagram

3. input-output variable

Input Variables: None

Output Variables:

output variable	Variable type	exegesis
BSP_SYS_GetTemperatureValue	INT	Temperature range: -40℃~120℃

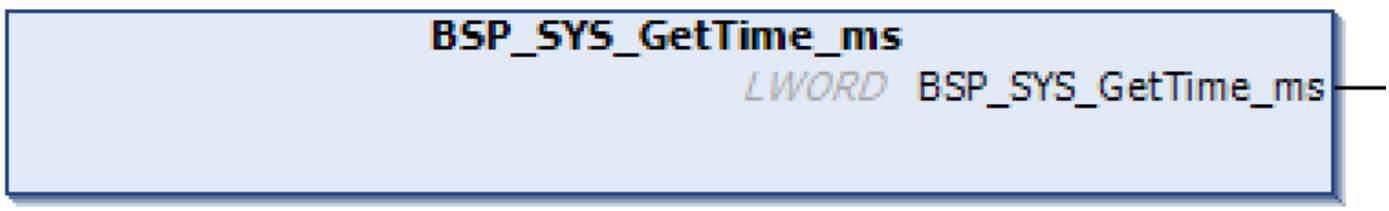
BSP_SYS_GetTime_ms function

1.1.1.1. BSP_SYS_GetTime_ms function

1. Function Description

Gets the time in ms since the controller was powered up.

2. function block diagram



3. input-output variable

Input Variables: None

Output Variables:

output variable	Variable type	exegesis
BSP_SYS_GetTime_ms	LWORD	Unit: ms

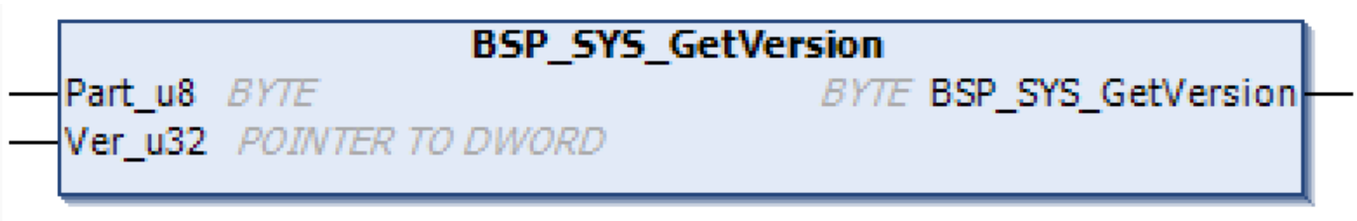
BSP_SYS_GetVersion function

1.1.1.1. BSP_SYS_GetVersion function

1. Function Description

Get controller version information.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Part_u8	BYTE	Part Number
Ver_U32	POINTER TO DWORD	Pointer to the starting address of the variable storing version information

Output Variables:

output variable	Variable type	exegesis
BSP_SYS_GetVersion	BYTE	Return Value: 0: Acquisition success 2: The variable pointer is NULL

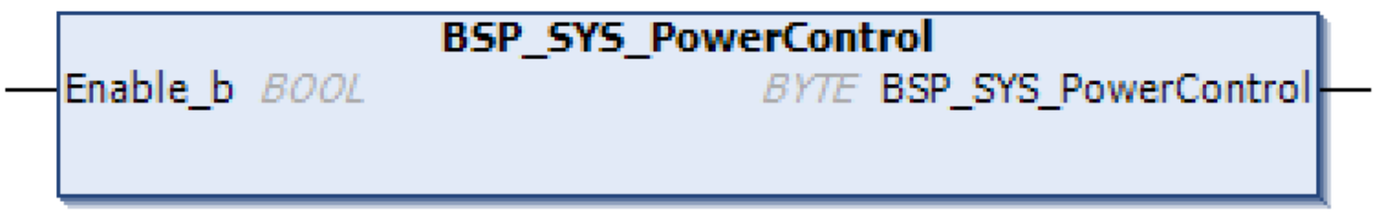
BSP_SYS_PowerControl Function

1.1.1.1. BSP_SYS_PowerControl Function

1. Function Description

Setting up the KL15 power supply affects the system.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Enable_b	BOOL	With normal power supply to the UBS: = TRUE: System shutdown when KL15 has no power = FALSE, when KL15 is not powered, the system still works normally

Output Variables:

output variable	Variable type	exegesis
BSP_SYS_PowerControl	BYTE	Return Value: 0: Successful setup Other values: setup failure

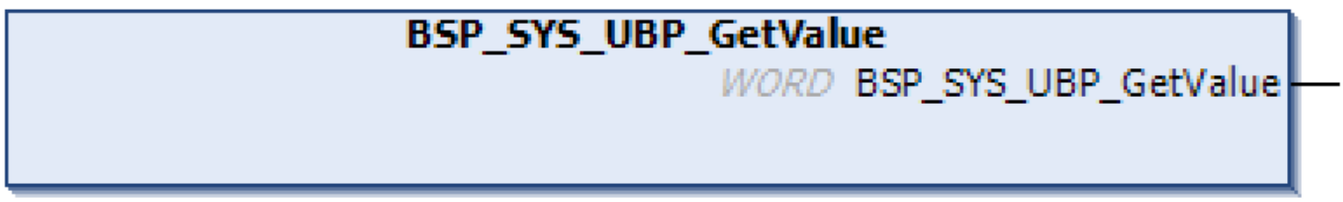
BSP_SYS_UBP_GetValue function

1.1.1.1. BSP_SYS_UBP_GetValue function

1. Function Description

Get the UBP supply voltage value.

2. function block diagram



3. input-output variable

Input Variables: None

Output Variables:

output variable	Variable type	exegesis
BSP_SYS_UBP_GetValue	WORD	Get UBP supply voltage value in mV

4. Precautions for use

None.

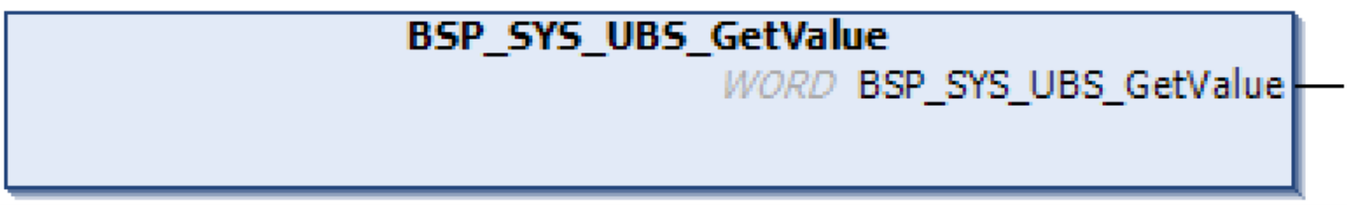
BSP_SYS_UBS_GetValue function

1.1.1.1. BSP_SYS_UBS_GetValue function

1. Function Description

Get the UBS supply voltage value.

2. function block diagram



3. input-output variable

Input Variables: None

Output Variables:

output variable	Variable type	exegesis
BSP_SYS_UBS_GetValue	WORD	Get the UBS supply voltage value in mV.

Misc general

1.1.1. Location in the library file

You can view the library functions etc. described below by Library Manager → JSHR Std Library → Misc. As shown in Figure 7:

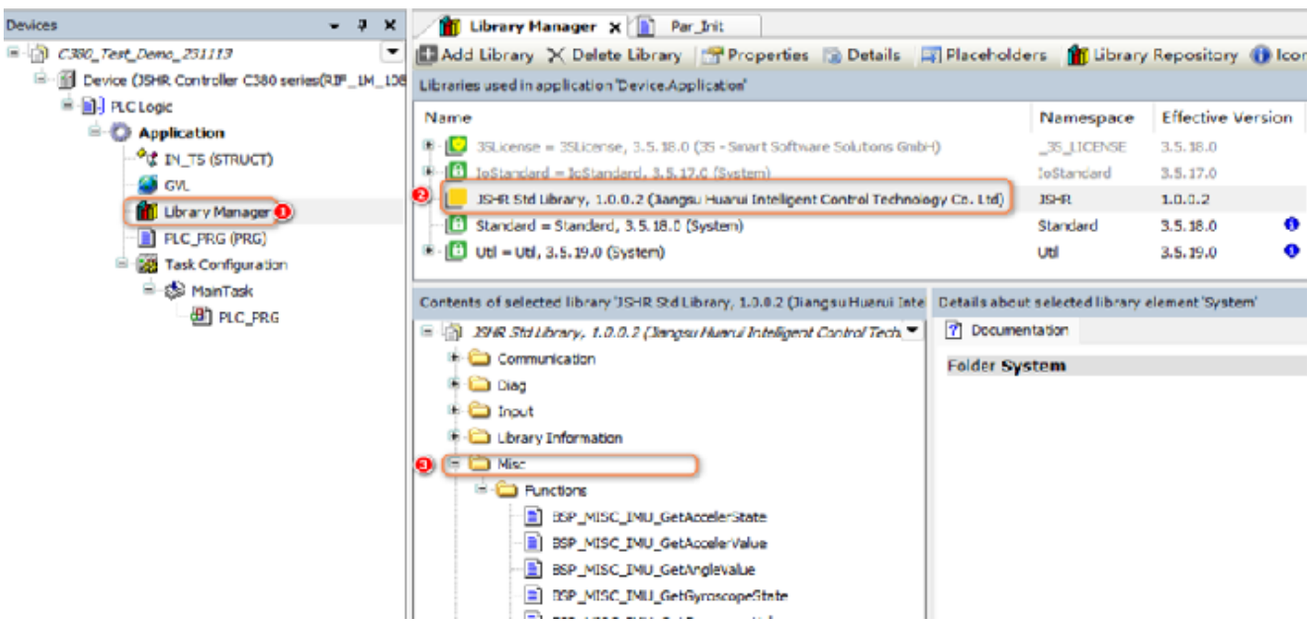


Figure 7: Location of relevant library files

BSP_MISC_IMU_GetAccelerState function

1.1.1.1. BSP_MISC_IMU_GetAccelerState function

1. Function Description

Get the status of the controller's built-in IMU chip.

2. function block diagram

3. input-output variable

Input Variables: None

Output Variables:

output variable	Variable type	exegesis
BSP_MISC_IMU_GetAccelerState	BYTE	Return Value: 0: Success Other values: abnormal

4. Precautions for use

None.

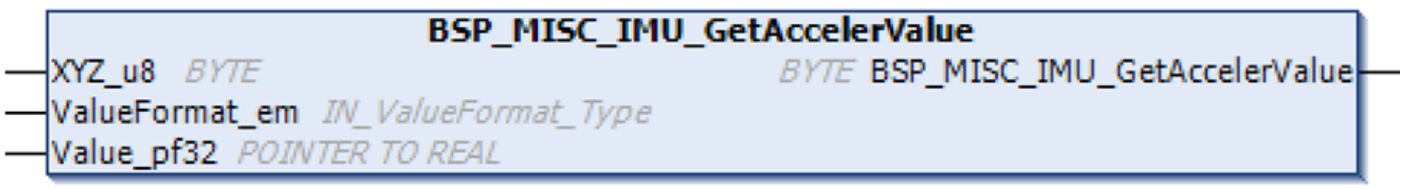
BSP_MISC_IMU_GetAccelerValue function

1.1.1.1. BSP_MISC_IMU_GetAccelerValue function

1. Function Description

Get the acceleration value of the controller's built-in IMU chip.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
XYZ_u8	BYTE	0: Get X-axis acceleration value 1: Get the Y-axis acceleration value 2: Get the Z-axis acceleration value
ValueFormat_em	IN_ValueFormat_Type	Get the value of the input signal, refer to: Section 5.2.2.3.
Value_pf32	POINTER TO REAL	Pointer to the variable storing the result of the fetch

Output Variables:

output variable	Variable type	exegesis
BSP_MISC_IMU_GetAccelerValue	BYTE	Return Value: 0: Acquisition success Other values: Failed to obtain

4. Precautions for use

None.

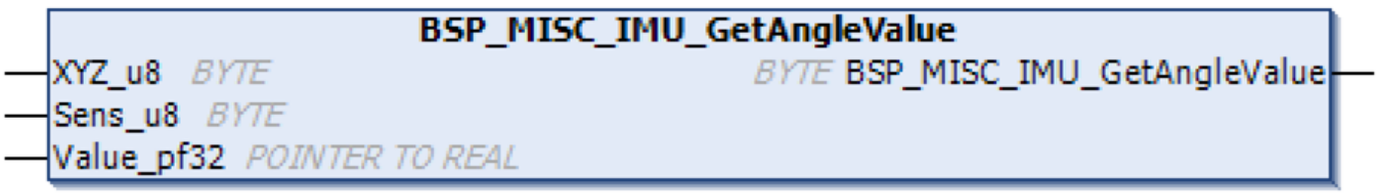
BSP_MISC_IMU_GetAngleValue function

1.1.1.1. BSP_MISC_IMU_GetAngleValue function

1. Function Description

Get the angle value of the controller's built-in IMU chip.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
XYZ_u8	BYTE	0: Get X-axis acceleration value 1: Get the Y-axis acceleration value 2: Get the Z-axis acceleration value
Sens_u8	BYTE	Sensitivity, value range: 1~255, the smaller the value, the higher the sensitivity
Value_pf32	POINTER TO REAL	Pointer to the variable storing the result of the fetch in

Output Variables:

output variable	Variable type	exegesis
BSP_MISC_IMU_GetAngleValue	BYTE	Return Value: 0: Acquisition success Other values: Failed to obtain

4. Precautions for use

None.

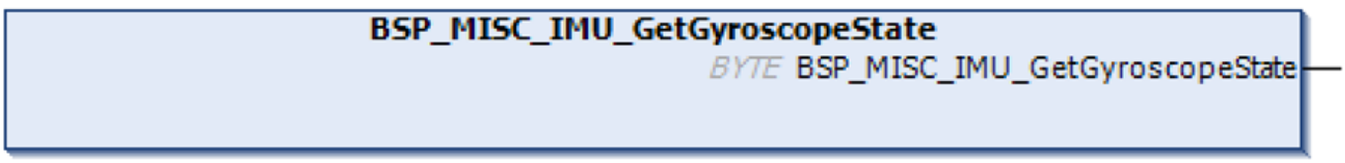
BSP_MISC_IMU_GetGyroscopeState function

1.1.1.1. BSP_MISC_IMU_GetGyroscopeState function

1. Function Description

Get the status of the controller's built-in gyroscope chip.

2. function block diagram



3. input-output variable

Input Variables: None

Output Variables:

output variable	Variable type	exegesis
BSP_MISC_IMU_GetGyroscopeState	BYTE	Return Value: 0: Successful setup Other values: setup failure

4. Precautions for use

None.

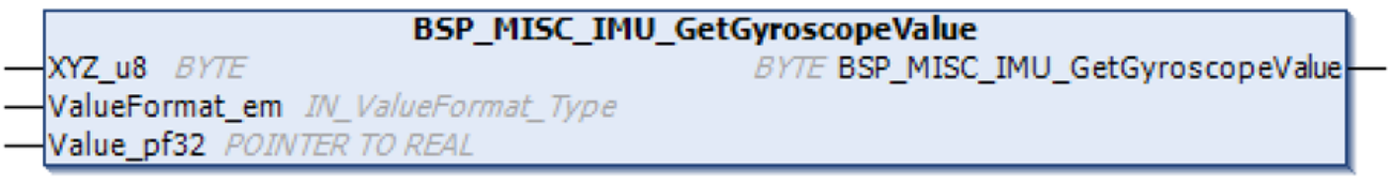
BSP_MISC_IMU_GetGyroscopeValue function

1.1.1.1. BSP_MISC_IMU_GetGyroscopeValue function

1. Function Description

Get the value of the controller's built-in gyroscope chip.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
XYZ_u8	BYTE	0: Get the X-axis angular velocity value 1: Get the Y-axis angular velocity value 2: Get the Z-axis angular velocity value
ValueFormat_em	IN_ValueFormat_Type	Get the value of the input signal, refer to: Section 4.2.2.3.
Value_pf32	POINTER TO REAL	Pointer to the variable storing the result of the acquisition, numerical unit: °/s

Output Variables:

output variable	Variable type	exegesis
BSP_MISC_IMU_GetAccelerValue	BYTE	Return Value: 0: Acquisition success Other values: Failed to obtain

4. Precautions for use

None.

BSP_MISC_KL15_GetStatus function

1.1.1.1. BSP_MISC_KL15_GetStatus function

1. Function Description

Get the status of the controller's built-in KL15 signal.

2. function block diagram

3. input-output variable

Input Variables: not have

Output Variables:

output variable	Variable type	exegesis
BSP_MISC_KL15_GetStatus	BYTE	Return Value: 0: low level 1: High level

BSP_MISC_LED_Set Function

1.1.1.1. BSP_MISC_LED_Set Function

1. Function Description

Sets the status of the controller LEDs.

2. function block diagram

3. input-output variable

Input Variables:

input variable	Variable type	exegesis
LedNum_u8	BYTE	LED indicator serial number 0 - red, 1 - green, 2 - blue
OnOff_b	BOOL	Sets the value of the indicator. TRUE: Indicator light on FALSE: Indicator light off

Output Variables:

output variable	Variable type	exegesis
BSP_MISC_LED_Set	BYTE	Return Value: 0: Successful setup Other values: setup failure

BSP_MISC_VSS_Enable function

1.1.1.1. BSP_MISC_VSS_Enable function

1. Function Description

5V supply voltage output enable.

2. function block diagram

3. input-output variable

Input Variables:

input variable	Variable type	exegesis
LedNum_u8	BYTE	Supply voltage output serial number 0- VSS1, 5V 1- VSS2, 10V 2- VSS3, 5V
Enable_b	BOOL	TRUE: Enable supply voltage output FALSE: Turns off supply voltage output

Output Variables:

output variable	Variable type	exegesis
BSP_MISC_VSS_Enable	BYTE	Return Value: 0: Successful setup Other values: setup failure

4. Precautions for use

The T680 contains 3-channel VSS supply voltage outputs as defined in Section 4.6.

※Special note:

VSS2 is a special supply voltage output, the following functions must enable VSS2 when used:

(1) When INPUT input port type is IN_DIL

(2) IN_66,69,71,73,75 when port type is IN_FIH/FIL

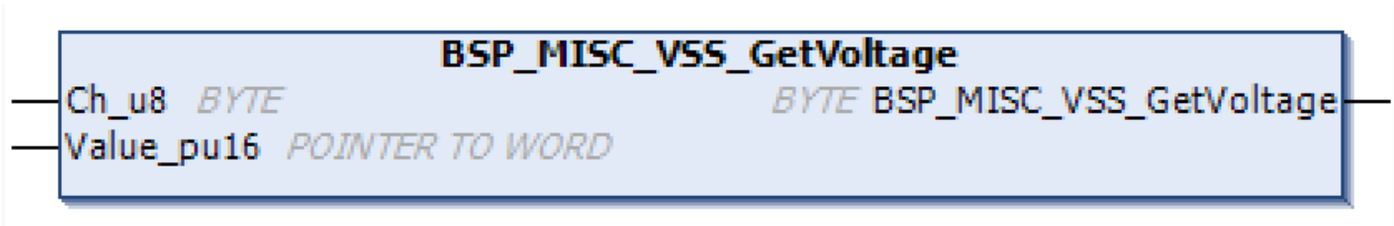
BSP_MISC_VSS_GetVoltage function

1.1.1.1. BSP_MISC_VSS_GetVoltage function

1. Function Description

5V supply voltage monitoring.

2. function block diagram



3. input-output variable

Input Variables:

input variable	Variable type	exegesis
Ch_u8	BYTE	5V supply voltage output serial number
Value_pi16	Pointer to WORD	Pointer to the variable storing the result of the acquisition, value unit: mV

Output Variables:

output variable	Variable type	exegesis
BSP_MISC_VSS_GetVoltage	BYTE	Return Value: 0: read successfully Other values: read failure

Library general

1.1.1. Location in the library file

You can view the library functions etc. described below by Library Manager → JSHR Std Library → Library Information. As shown in Figure 8:

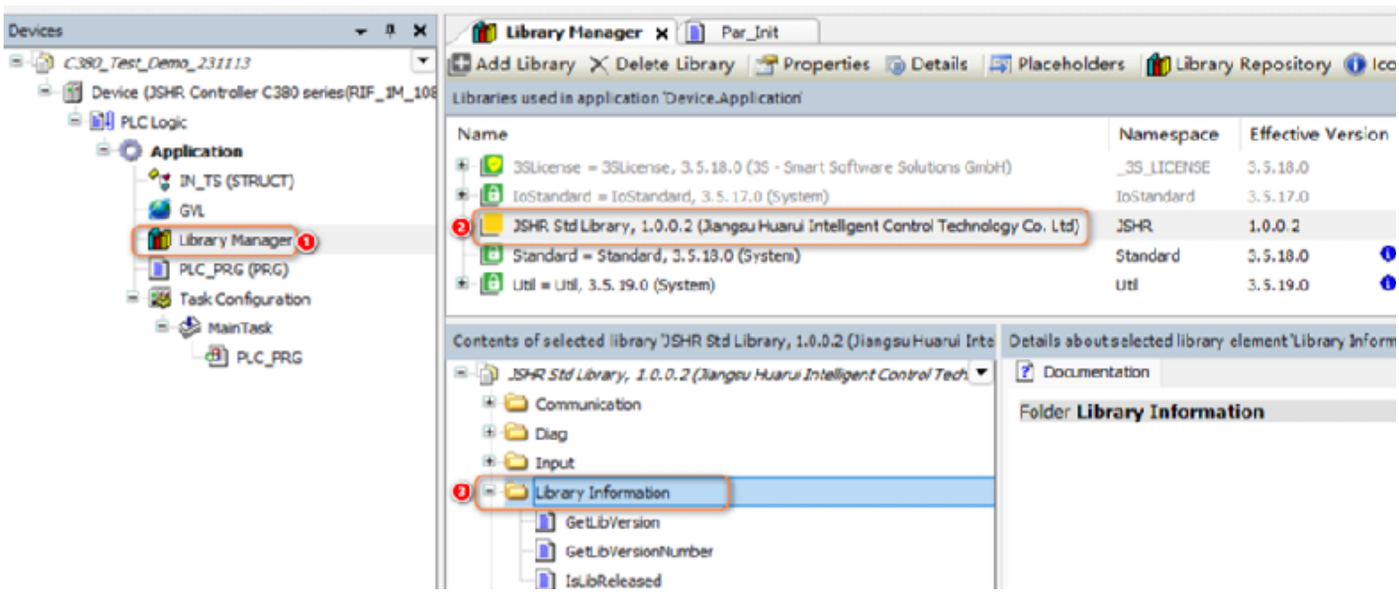


Figure 8: Location of Library Information related library files

GetLibVersion function

1.1.1.1. GetLibVersion function
1. Function Description
Get the version information of the library.
2. function block diagram

3. input-output variable
Input Variables: not have

Output Variables:

output variable	Variable type	exegesis
GetLibVersion	VERSION	Returns Library version information

GetLibVersionNumber function

1.1.1.1. GetLibVersionNumber function
1. Function Description
Get the version number of the library.
2. function block diagram

3. input-output variable
Input Variables: not have

Output Variables:

output variable	Variable type	exegesis
GetLibVersionNumber	DWORD	Returns the Library version number

IsLibReleased function

1.1.1.1. IsLibReleased function
1. Function Description
Get the release status of the Library.
2. function block diagram

3. input-output variable
Input Variables: not have

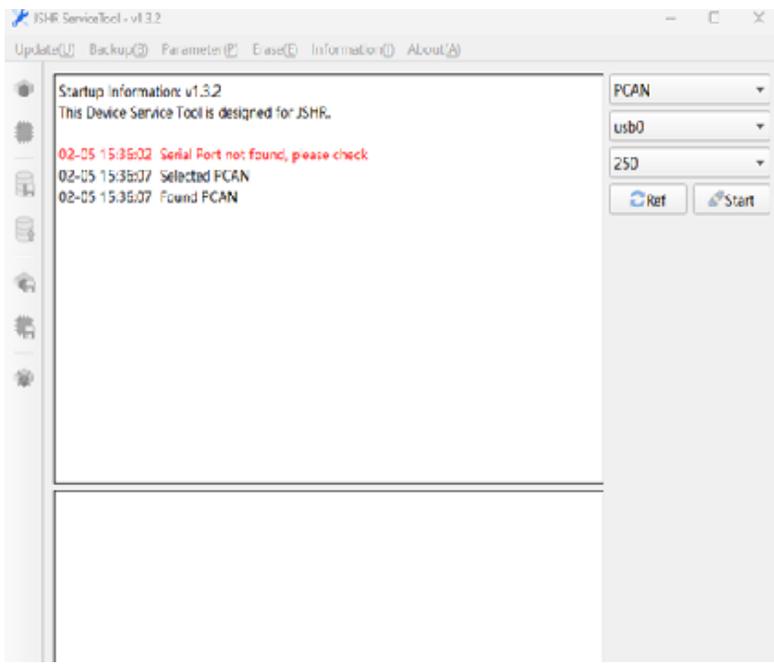
Output Variables:

output variable	Variable type	exegesis
IsLibReleased	BOOL	TRUE: Library is published. FALSE: Library is in internal testing status

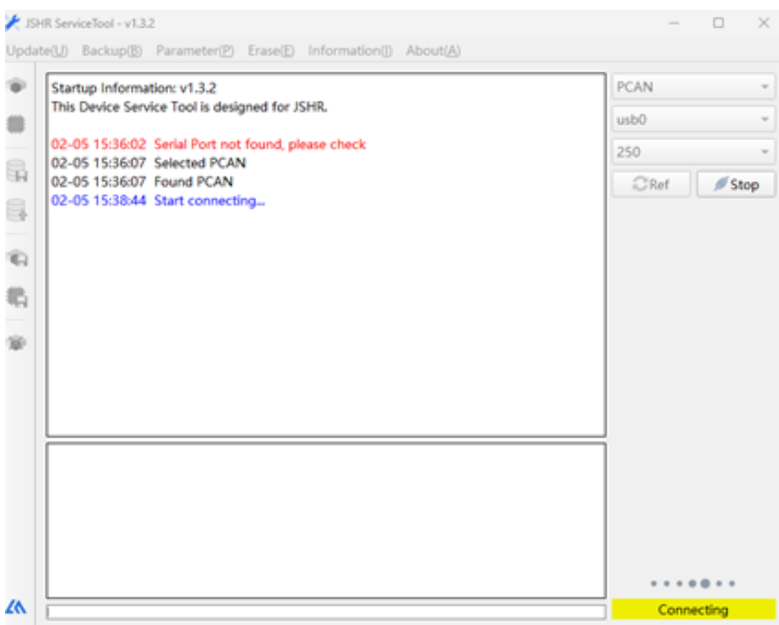
Service

1. Service Tool ,This software is for service update codesys app or firmware.(not debugging online)

2. Please select PCAN device, ,250kbps



3. Click Start button



4. Then reboot the cotroller. Wait for the link succeed.

FAQ

1.1. common problems

1.1.1. CodeSys cannot find the device

A: Please check whether the peak-can driver is installed correctly, whether the device manager can find the device, and make sure the device is not occupied by other tools except Gateway (you can refer to the Gateway startup log to make sure that Gateway starts successfully). Check if the Gateway configuration is correct, check if the controller power supply is normal and the communication line is correct. Make sure the device description file matches the controller.

1.1.2. Codesys app download or online debugging always disconnects or gives an error

A: Please check whether the line of communication is normal, whether there is interference. can the driver is compatible with the operating system or the driver does not match, replace the driver to try. If the problem still exists, you can change PC to try.

1.1.3. App doesn't run properly or restarts after download

A: Usually it is caused by memory out of bounds, dead loop and other problems in the code. Dezero will not lead to dead reboot, but there will be the problem of abnormal calculation results, you can use codesys's out-of-bounds checking and dezero protection function. If this problem occurs, you can ground the BSL, power on the boot mode, and use the erase function of the tool to erase the app, which will solve the problem.

1.1.4. CAN frames are sometimes not received

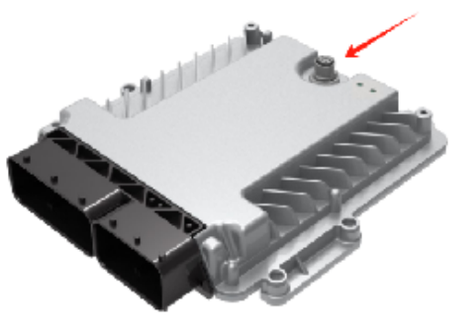
A: There are multiple function calls to receive the same ID in the program, and after the received ID is processed by one interface, it will not be received in another.

1.1.5. Simultaneous monitoring of multiple Traces occasionally results in breakpoints

A: Due to the limitation of CAN bandwidth, up to five 4-byte variables with 10ms sampling are supported for tracking when other monitoring interfaces are not opened. If you want to increase the number, you can increase the sampling interval such as 20ms, then the monitoring can reach 10 variables. This is limited by the type of variables to be monitored and the CAN transmission bandwidth. For higher sample intervals, the number of monitored variables can be further increased by using CAN for online debugging with a baud rate of 250k or 500k.

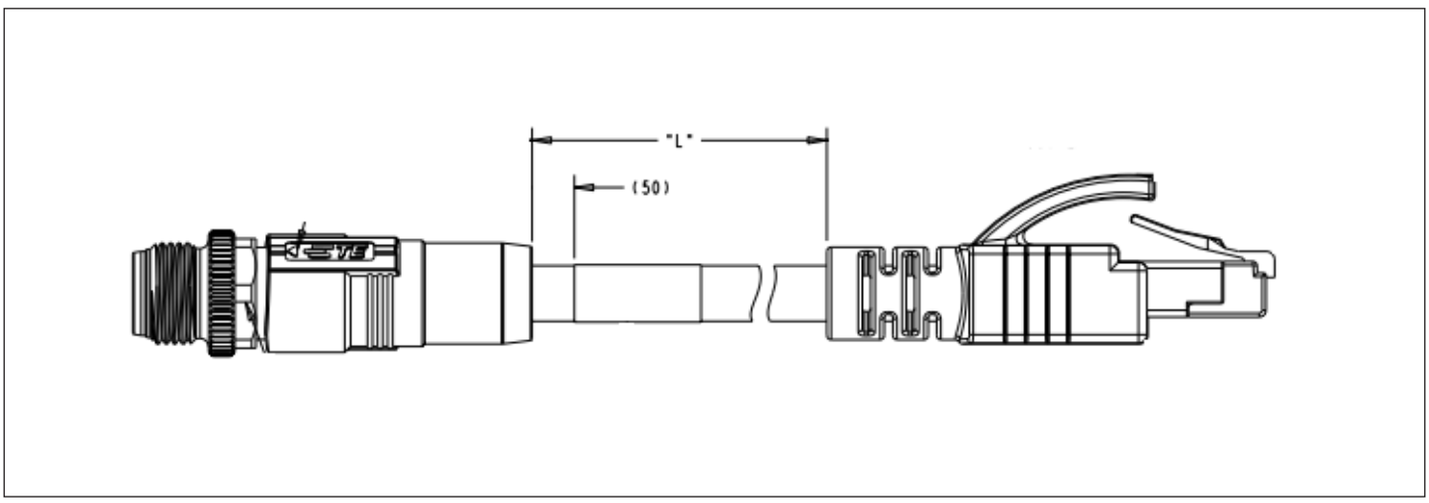
1.1.6 Pinout (connection diagram) of the 4-pin connector on the front side of the controller

A. These four pins are Ethernet, which is reserved for debug function , cannot be used for the this time coming later.



ETH Connect

PIC	Number	Define	RJ 45
	1	TD+	Pin1 white-orange
	2	RD+	Pin2 orange
	3	TD-	Pin3 white-green
	4	RD-	Pin6 green


"L"
(50)

The specified data is for product description purposes only and may not be deemed to be guaranteed unless expressly confirmed in the contract.

THM Huade Hydraulics Pvt Ltd
F-127, Phase-VIII, Focal Point,
Ludhiana-141010, Punjab (INDIA)
PH: 0161-2672777, 0161-2672778
E-mail: sales@thmhuade.com
Website: www.thmhuade.com

Follow us: