

Hubo II CAN Message Protocols

TX Message ID

Motor Command Message ID	0x01
Sensor Command Message ID	0x02
Reference Message ID	0x10 + BNO

RX Message ID

FT Sensor data Message ID	0x40 + SBNO
Tilt Sensor/IMU data Message ID	0x50 + SBNO
Encoder value Message ID	0x60 + BNO
Status Message ID	0x150 + BNO
Board Information Message ID	0x190 + BNO + BOFF
Board Para & Current Message ID	0x1C0 + BNO

(NOTE)

BNO: Board Number

SBNO: Board Number for Sensor Boards. SBNO=BNO-0x2F

BOFF=0 for BNO < 0x30

BOFF=0x80 for BNO >= 0x30

CAN packet coding example

Mail Box ID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
MesgID	DL	4-Byte-INT				2-Byte-INT		1-BYTE-INT	-

DL: Data length

4_Byte_INT = Byte0 | (Byte1 < <8) | (Byte1 < <16) | (Byte1 < <24)

2_Byte_INT = Byte4 | (Byte5 < <8)

1_Byte_INT = Byte6

Board Description Table

Board Name	Board No (BNO)	SBNO	Description	No of Axis	Board Type	Firmware No
JMC0	0x00(0)	-	Right Hip Yaw/Right Hip Roll	2	2	1
JMC1	0x01(1)	-	Right Hip Pitch	1	1	1
JMC2	0x02(2)	-	Right Knee	1	1	1
JMC3	0x03(3)	-	Right Ankle Pitch/Right Ankle Roll	2	2	1
JMC4	0x04(4)	-	Left Hip Yaw/Left Hip Roll	2	2	1
JMC5	0x05(5)	-	Left Hip Pitch	1	1	1
JMC6	0x06(6)	-	Left Knee	1	1	1
JMC7	0x07(7)	-	Left Ankle Pitch/Left Ankle Roll	2	2	1
JMC8	0x08(8)	-	Right Shoulder Pitch/Right Shoulder Roll	2	2	1
JMC9	0x09(9)	-	Right Shoulder Yaw/Right Elbow	2	2	1
JMC10	0x0A(10)	-	Left Shoulder Pitch/Left Shoulder Roll	2	2	1
JMC11	0x0B(11)	-	Left Shoulder Yaw/Left Elbow	2	2	1
JMC12	0x0C(12)	-	extra	-	-	-
JMC13	0x0D(13)	-	extra	-	-	-
JMC14	0x0E(14)	-	Smart Power Controller	-	9	4
JMC15	0x0F(15)	-	extra	-	-	-
						-
EJMC0	0x20(32)	-	Right Wrist Yaw/Right Pitch	2	3	1
EJMC1	0x21(33)	-	Left Wrist Yaw/Left Wrist Pitch	2	3	1
EJMC2	0x22(34)	-	Nect Yaw/Neck 1/Neck 2	3	3	1
EJMC3	0x23(35)	-	Waist	1	1	1
EJMC4	0x24(36)	-	Right Finger0/Finger1/Finger2/Finger3/Finger4	5	5	1
EJMC5	0x25(37)	-	Left Finger0/Finger1/Finger2/Finger3/Finger4	5	5	1
EJMC6	0x26(38)	-	Extra	-	-	-
				-	-	-
FT0	0x30(48)	1	Right Foot F/T sensor	-	6	2
FT1	0x31(49)	2	Left Foot F/T Sensor	-	6	2
IMU0	0x32(50)	3	IMU sensor 0	-	7	3
IMU1	0x33(51)	4	IMU sensor 1	-	7	3
IMU2	0x34(52)	5	IMU sensor 2	-	7	3
FT3	0x35(53)	6	Right Wrist F/T Sensor	-	8	5
FT4	0x36(54)	7	Left Wrist F/T Sensor	-	8	5

1. Motor Control Board

Command Message for Motor Boards (Message ID=0x01)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Set & Req. Board Info. ¹⁾	BNO	0x01	CANR	-	-	-	-	-
Req. Board Status ²⁾	BNO	0x02	-	-	-	-	-	-
Req. Encoder Position ³⁾	BNO	0x03	FES	-	-	-	-	-
Req. Current ⁴⁾	BNO	0x04	-	-	-	-	-	-
Reset Encoder to zero	BNO	0x06	CH	-	-	-	-	-
Set position gain0	BNO	0x07	Kp0		Ki0		Kd0	
Set position gain1	BNO	0x08	Kp1		Ki1		Kd1	
Set current gain0	BNO	0x09	KPt0		KDt0		KF0	
Set current gain1	BNO	0x0A	KPt1		KDt1		KF1	
Turn ON/OFF Driver(HIP)	BNO	0x0B	HIP_EN	-	-	-	-	-
Open loop PWM(1, 2CH)	BNO	0x0D	PUL_ON	DIR0	DUTY0	DIR1	DUTY1	-
for Finger(5CH)	BNO	0x0D	PUL_ON	D_DT0	D_DT1	D_DT2	D_DT3	D_DT4
for Neck(3CH)	BNO	0x0D	PUL_ON	D_DT0	D_DT1	D_DT2	-	-
Turn ON Controller	BNO	0x0E	-	-	-	-	-	-
Turn OFF Controller	BNO	0x0F	-	-	-	-	-	-
Set Control mode	BNO	0x10	FBC	-	-	-	-	-
Go to Home Offset	BNO	0x11	CH&D	SDR	H_OFFSET(확인)			
Set Dead Zone	BNO	0x20+CH	DZone	-	-	-	-	-
Req. Board Parameters ⁵⁾	BNO	0x24	PARM	-	-	-	-	-
Set Home Search Para.	BNO	0x30+CH	SRL	SDR	OFFSET			
Set Encoder Resolution	BNO	0x38+CH	ENC_RE		-	-	-	-
Set Max. Acc.& Vel.	BNO	0x40+CH	MACC		MVEL		-	-
Set Lower Position Limit	BNO	0x50+CH	MPS	MPOS1				-
Set Upper Position Limit	BNO	0x56+CH	MPS	MPOS2				-
Set Home Vel. & Acc.	BNO	0x60+CH	HMA	HMV1	HMV2	SRM	LIMD	-
Set Gain Override	BNO	0x6F	GOVW0	GOVW1	GDUR		-	-
Set New Board Number	BNO	0xF0	NEW_BNO	CANR	-	-	-	-
Set Jam & PWM Sat. lim.	BNO	0xF2	JAM_LIM		PWM_LIM		LIMD	JAMD
Set Error Bound	BNO	0xF3	I_ERR		B_ERR		E_ERR	
Initialize Board	BNO	0xFA	-	-	-	-	-	-

Reference Message for Motor Boards (Message ID=0x10 + BNO)

Description	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
Send Position Reference	REF0				REF1			
for Finger(5CH)	REFR0	REFR1	REFR2	REFR3	REFR4			
for Neck(3CH)	REFR0	REFR1	REFR2					
<p>(NOTE)</p> <p>1. REF0=Byte0 (Byte1<<8) (Byte2<<16) (Byte3<<24) REF1=Byte4 (Byte5<<8) (Byte6<<16) (Byte7<<24)</p> <p>2. Reference for Finger and Neck is given by differential value of position, REFRx.</p> <p>Single byte REFRx is coded by 2's Complementary binary. To decode a negative number, (int)VAL = REFRx – 0x100 .</p> <p>3. If the control mode is "position" the reference value is POSITION in encoder unit.</p> <p>4. If the control mode is "current", the reference value is Current in 0.01A unit for finger board or 0.05A unit for the other boards.</p>								

Return Message From Motor Boards

Description	MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
1) Send Board Info	0x190+BNO + BOFF	CANR	1-	BTY	VER0	VER1	VER2	VER3	-
2) Send Board Status	0x150+BNO	STAT00	STAT01	STAT02	STAT03	STAT10	STAT11	STAT12	STAT13
for Neck & Finger	""	STAT0	STAT1	STAT2	STAT3	STAT4	-	-	-
3) Send Encoder Posi.	0x60+BNO	M0_POS				M1_POS			
for Neck	""	M0_POS		M1_POS		M2_POS			
for Finger, FES=0	""	M0_POR		M2_POR		M2_POS			
for Finger, FES=1	""	M3_POS		M4_POS					
4) Send Current	0x1C0+BNO	B0	B1	B2	B3	B4	B5	B6	B7
5) Send Board Para.	0x1C0+BNO	B0	B1	B2	B3	B4	B5	B6	B7

Default values

BNO	Joint	Limit		PID Gain			DZ	Homeset1			Homeset2				Encoder		
		MPOS1	MPOS2	Kp	Kd	Ki		OFF	HLIM	HLD	HV1	HV2	HMA	SM	ERS	AS	MD
0	RHY	-90000	90000	200	500	0	44	0	20	1	40	50	0.1	0	4000	1	1
	RHR	-150000	120000	200	500	0	44	4500	20	1	40	50	0.1	0	4000	1	0
1	RHP1	-165000	130000	200	500	0	44	-4000	20	0	40	50	0.1	0	4000	1	1
	RHP2																
2	RKN1	-30000	220000	200	500	0	44	-13000	20	1	40	50	0.1	0	4000	1	0
	RKN2																
3	RAP	-170000	170000	200	500	0	44	0	20	1	40	50	0.1	0	4000	1	0
	RAR	-78000	78000	200	500	0	44	12000	20	1	40	50	0.1	0	4000	1	1
4	LHY	-90000	90000	200	500	0	44	0	20	0	40	50	0.1	0	4000	1	1
	LHR	-120000	150000	200	500	0	44	-4500	20	0	40	50	0.1	0	4000	1	0
5	LHP1	-165000	130000	200	500	0	44	-4000	20	0	40	50	0.1	0	4000	1	0
	LHP2																
6	LKN1	-30000	220000	200	500	0	44	-13000	20	1	40	50	0.1	0	4000	1	1
	LKN2																
7	LAP	-170000	170000	200	500	0	44	0	20	1	40	50	0.1	0	4000	1	1
	LAR	-78000	78000	200	500	0	44	-8000	20	0	40	50	0.1	0	4000	1	1
8	RSP	-310000	280000	100	250	0	28	0	20	1	40	50	0.1	0	4000	1	1
	RSR	-180000	10000	100	250	0	28	13333	20	1	40	50	0.1	0	4000	1	1
9	RSY	-150000	100000	100	250	0	28	0	20	1	40	50	0.1	0	4000	1	1
	REB	-190000	0	100	250	0	28	0	20	1	40	50	0.1	0	4000	1	1
10	LSP	-310000	280000	100	250	0	28	0	20	1	40	50	0.1	0	4000	1	0
	LSR	-10000	180000	100	250	0	28	-11666	20	0	40	50	0.1	0	4000	1	1
11	LSY	-100000	150000	100	250	0	28	0	20	0	40	50	0.1	0	4000	1	1
	LEB	-190000	0	100	250	0	28	10000	20	1	40	50	0.1	0	4000	1	0
32	RWY	-100000	180000	2000	5000	0	140	0	100	0	100	100	1	1	4000	1	0
	RWP	-70000	90000	2000	5000	0	140	100000	100	0	100	100	1	2	4000	1	0
33	LWY	-180000	100000	2000	5000	0	140	0	100	1	100	100	1	1	4000	1	0
	LWP	-70000	90000	2000	5000	0	140	100000	100	0	100	100	1	2	4000	1	1
34	NKY	-4000	4000	40	100	0	90	0	30	1	2	2	1	1	128	1	1
	NK1	-1300	1050	40	100	0	90	-1470	30	1	2	2	1	2	128	1	0
	NK2	-1300	1050	40	100	0	90	-1470	30	1	2	2	1	2	128	1	0
35	WST	-170000	170000	200	500	0	44	0	20	0	40	50	0.1	0	4000	1	0
36	RH0	-4800	4400	50	100	0	0	-5000	100	1	20	20	1	2	64	1	0
	RH1	-4800	4400	50	100	0	0	-5000	100	1	20	20	1	2	64	1	1
	RH2	-4800	4400	50	100	0	0	-5000	100	1	20	20	1	2	64	1	1
	RH3	-4800	4400	50	100	0	0	-5000	100	1	20	20	1	2	64	1	0
	RH4	-4800	4400	50	100	0	0	-5000	100	1	20	20	1	2	64	1	0
37	LH0	-4800	4400	50	100	0	0	-5000	100	1	20	20	1	2	64	1	1
	LH1	-4800	4400	50	100	0	0	-5000	100	1	20	20	1	2	64	1	0
	LH2	-4800	4400	50	100	0	0	-5000	100	1	20	20	1	2	64	1	0
	LH3	-4800	4400	50	100	0	0	-5000	100	1	20	20	1	2	64	1	1
	LH4	-4800	4400	50	100	0	0	-5000	100	1	20	20	1	2	64	1	1

MPOS1: MAX_POS1, MPOS2: MAX_POS2, DZ: D_ZONE, OFF: OFFSET, HLIM: Home_Lim, HLD: Lim_dir, HV1: V_max1, HV2: V_max2, HMA: A_max, SM: Search Mode, ERS: Encoder resolution, AS: Auto scale, MD: Motor Dirction

BNO	Joint	Speed Limits		Jam and Power Saturation Limit			Error Limit	
		Vmax	Amax	JAM_LIM(sec10)	JAMD(%)	PWM_LIM(sec10)	I_ERR	B_ERR
0	RHY	1500 (22500rpm)	20 (3	50	5	20000	10000
	RHR	1500	20	3	50	5	20000	10000
1	RHP1	1500	20	3	50	5	20000	10000
	RHP2							
2	RKN1	1500	20	3	50	5	20000	10000
	RKN2							
3	RAP	1500	20	3	50	5	20000	10000
	RAR	1500	20	3	50	5	20000	10000
4	LHY	1500	20	3	50	5	20000	10000
	LHR	1500	20	3	50	5	20000	10000
5	LHP1	1500	20	3	50	5	20000	10000
	LHP2							
6	LKN1	1500	20	3	50	5	20000	10000
	LKN2							
7	LAP	1500	20	3	50	5	20000	10000
	LAR	1500	20	3	50	5	20000	10000
8	RSP	1500	20	3	50	5	20000	10000
	RSR	1500	20	3	50	5	20000	10000
9	RSY	1500	20	3	50	5	20000	10000
	REB	1500	20	3	50	5	20000	10000
10	LSP	1500	20	3	50	5	20000	10000
	LSR	1500	20	3	50	5	20000	10000
11	LSY	1500	20	3	20	5	20000	10000
	LEB	1500	20	3	50	5	20000	10000
32	RWY	-100000	180000	2000	5000	0	0	10000
	RWP	-70000	90000	2000	5000	0	100000	10000
33	LWY	-180000	100000	2000	5000	0	0	10000
	LWP	-70000	90000	2000	5000	0	100000	10000
34	NKY	-4000	4000	40	100	0	0	30
	NK1	-1300	1050	40	100	0	-1470	30
	NK2	-1300	1050	40	100	0	-1470	30
35	WST	-170000	170000	200	500	0	0	20
36	RH0	-4800	4400	50	100	0	-5000	100
	RH1	-4800	4400	50	100	0	-5000	100
	RH2	-4800	4400	50	100	0	-5000	100
	RH3	-4800	4400	50	100	0	-5000	100
	RH4	-4800	4400	50	100	0	-5000	100
37	LH0	-4800	4400	50	100	0	-5000	100
	LH1	-4800	4400	50	100	0	-5000	100
	LH2	-4800	4400	50	100	0	-5000	100
	LH3	-4800	4400	50	100	0	-5000	100
	LH4	-4800	4400	50	100	0	-5000	100

Detail Description of Command Message

1. Set and Request Board Information(RBI: 0x01)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x01	CANR	-	-	-	-	-

CANR: CAN rate in msec.

Default: 5(ms)

Return:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x190+BNO + BOFF	CANR	1	BTY	VERSION			-	

CANR: Echo CANR to confirm CAN rate.

BTY: Board Type:

- 1: 1CH – 2 Motor BLDC Board
- 2: 2CH – 2 Motor BLDC Board
- 3: 3CH DC motor Board for Neck joint or 2CH DC Board for Wrist
- 4: 1CH – 1 Motor BLDC Board for Waist
- 5: 5CH -5 Motor Board for Hand(Finger)
- 6: F/T Sensor for Foot
- 7: Firmware for IMU Board
- 8: F/T Sensor for Wrist
- 9: Smart Power Control Board

VERSION is consisted of 7 digit decimal number as below:

D6	D5	D4	D3	D2	D1	D0
Firmware No	Version code					

Firmware No:

- 1: Firmware for Motor Board
- 2: Firmware for F/T Sensor for Foot
- 3: Firmware for IMU Board
- 4: Firmware for Smart Power Board
- 5: Firmware for F/T Sensor for Wrist

Version code:

D5D4: yy

D3D2: mm

D1D0: dd

2. Request Board Status and Error Flags (RBS: 0x02)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x02	-	-	-	-	-	-

Action: Request the Board to send the Status and Error flags. The Board automatically sends Status and Error flags without request if any change of the status is detected.

Return:

For 1CH and 2CH Board:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x150+BNO	STAT00	STAT01	STAT02	STAT03	STAT10	STAT11	STAT12	STAT13

STATx0

b7	b6	b5	b4	b3	b2	b1	b0
HME _x				LIM _x	MOD _x	RUN _x	HIP _x

HIP_x: 1: Motor x driver ON, 0: OFF

RUN_x: 1: Motor x Controller ON, 0:OFF

MOD_x: 1: Motor x Current Feedback mode, 0: Position Feedback mode

MIM_x: 1: Motor x Limit switch ON, 0: Limit switch OFF

HME: See 16.

STATx1

b7	b6	b5	b4	b3	b2	b1	b0
-	MO1 _x	MO0 _x	FLT _x	ENC _x	BIG _x	PWM _x	JAM _x

JAM_x: 1: Motor x JAM detected, 0: Normal

PWM_x: 1: Motor x PWM saturation detected, 0: Normal

BIG_x: 1: Motor x Position error is bigger than BERR, 0: Normal

ENC_x: 1: Motor x Encoder failure detected, 0: Normal

FLT_x: 1: Motor x Fault signal from motor driver x detected, 0: Normal

MO0: 1: Motor0 fail for type 1 board. 0: Normal

MO1: 1: Motor1 fail for type 1 board. 0: Normal

STATx2

b7	b6	b5	b4	b3	b2	b1	b0
TP _x		ABS _x	TMP _x	ACC _x	VEL _x	PS2 _x	PS1 _x

PS1_x: 1: Motor x Lower limit error, 0: Normal

PS2_x: 1: Motor x Upper limit error, 0: Normal

VEL_x: 1: Motor x Over velocity error, 0: Normal

ACC_x: 1: Motor x Over acceleration error, 0: Normal

TMP_x: 1: Motor x Over temperature error, 0: Normal

TP_x: Reserved for Motor x

STATx3: Reserved for Motor x

For 5CH (Finger) and 3CH (Neck) Boards:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x150+BNO	STAT0	STAT1	STAT2	STAT3	STAT4	STATB	-	-

STATx

b7	b6	b5	b4	b3	b2	b1	b0
ENCx	BIGx	PWMx	JAMx	LIMx	MODx	RUNx	HIPx

HIPx: 1: Motor x driver ON, 0: OFF

RUNx: 1: Motor x Controller ON, 0:OFF

MODx: 1: Motor x Current Feedback mode, 0: Position Feedback mode

LIMx: 1: Motor x Limit switch ON, 0: Limit switch OFF

JAMx: 1: Motor x JAM detected, 0: Normal

PWMx: 1: Motor x PWM saturation detected, 0: Normal

BIGx: 1: Motor x Position error is bigger than BERR, 0: Normal

ENCx: 1: Motor x Encoder failure detected, 0: Normal

STATB

b7	b6	b5	b4	b3	b2	b1	b0
SPA	SPA	SPA	SPA	SPA	SPA	SPA	-

3. Request Encoder Position (REP: 0x03)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x03	FES	-	-	-	-	-

FES: Finger Position Send flag. FES is ignored for other case than finger.

0: Send M0_POS, M1_POS, M2_POS for finger position

1: Send M3_POS, M4_POS for finger position

Return:

For 1CH and 1CH Board:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x60+BNO	M0_POS				M1_POS			

For 3CH neck board:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x60+BNO	M0_POS		M1_POS		M2_POS		-	-

For 5CH finger board:

FES=0:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x60+BNO	M0_POS		M1_POS		M2_POS		-	-

FES=1:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x60+BNO	M3_POS		M4_POS		-	-	-	-

4. Request Current value (RCU: 0x04)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x04	-	-	-	-	-	-

Retrun:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x1C0+BNO	B0	B1	B2	B3	B4	B5	B6	B7

For Board Type 1(CH0 only), 2, 4, 3(Wrist):

B0 = Motor0_CUR >> 2

B1 = Motor1_CUR >> 2

B2 = TEMP >> 2

B3 = (Motor0_CUR & 0x03) | (Motor0_CUR & 0x03) << 2 | (TEMP & 0x03) << 4

B4 = Motor0_JUL >> 2

B5 = Motor0_JUL >> 2

B6 = SPARE >> 2

B7 = (Motor0_JUL & 0x03) | (Motor0_JUL & 0x03) << 2 | (SPARE & 0x03) << 4

(NOTE: Motorx_CUR: Motorx Current in 10mA unit. Actual current is calculated by
Ampere = Motorx_CUR/100.

Motorx_JUL: Heat generated by Motor x in Joule.

For Board Type 3(Neck)

B0 ~ B3: same as above

B4 = Motor2_CUR >> 2

B5 = Motor2_CUR * 0x03

(NOTE: Motorx_CUR: Motorx Current in 4mA unit. Actual current is calculated by
Ampere = Motorx_CUR/250.

For Board Type 5(Finger)

B0 = Motor0_CUR >> 2

B1 = Motor1_CUR >> 2

B2 = Motor2_CUR >> 2

B3 = (Motor0_CUR & 0x03) | (Motor0_CUR & 0x03) << 2 | (Motor2_CUR & 0x03) << 4

B4 = Motor3_CUR >> 2

B5 = Motor4_CUR >> 2

B6 = SPARE >> 2

B7 = (Motor3_CUR & 0x03) | (Motor4_CUR & 0x03) << 2 | (SPARE & 0x03) << 4

(NOTE: Motorx_CUR: Motorx Current in 1mA unit. Actual current is calculated by
Ampere = Motorx_CUR/1000.

5. Reset Encoder to Zero (REZ: 0x06)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x06	CH	-	-	-	-	-

CH: Channel No.

CH= 0 ~ 4 according to the board.

CH= 0xF selects ALL Channel

Action:

1. Set encoder(s) to Zero.
2. Initialize internal parameters.
3. Reset Fault and Error Flags.

6. Set Motor Position Gain 0 (SMG0: 0x07)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x07	Kp0		Ki0		Kd0	

Sets Channel 0's PID gains.

For 3CH-Neck-Board and 5CH-Finger-Board this command sets ALL Channels' PID gains.

7. Set Motor Position Gain 1 (SMG1: 0x08)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x08	Kp1		Ki1		Kd1	

Sets Channel 1's PID gains.

8. Set Motor Current Gain 0 (SMC0: 0x09)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x09	Kpt0		Kdt0		Kf0	

Sets Channel 0's PI and filter gains.

For 3CH-Neck-Board and 5CH-Finger-Board this command sets ALL Channels' PID gains.

9. Set Motor Current Gain 1 (SMC1: 0x0A)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x0A	Kpt1		Kdt1		Kf1	

Sets Channel 1's PI and filter gains.

10. Motor Driver Enable/Disable (MDE: 0x0B)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x0B	HIP_EN	-	-	-	-	-

HIP_EN

1: Set HIP_EN bit enables Motor driver for all channels. Also DISABLES position feedback.

0: Clear HIP_EN bit disables Motor driver for all channels.

11.Reserved (0x0C)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x0C		-	-	-	-	-

12.Open loop PWM duty command in Percent(%) (PDU: 0x0D)

For 1CH and 2CH boards

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x0D	PUL_ON	DIR0	DUTY0	DIR1	DUTY1	B4

For 5CH boards

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x0D	PUL_ON	D_DT0	D_DT1	D_DT2	D_DT3	D_DT4

For 3CH boards

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x0D	PUL_ON	D_DT0	D_DT1	D_DT2	-	-

DIRx: 1: CW, 0:CCW

DUTYx: Percent PWM Duty. $-100 < \text{DUTYx} < 100$

D_DTx:

b7	b6	b5	b4	v3	b2	b1	b0
DIRx		DUTYx					

Action:

1. Disable Motor position feedback.
2. PUL_ON:
 - 1: Pulse out to run motor in specified PWM duty and direction
 - 0: Enforce Zero duty to stop motor. Back EMF will break the motor.

13.Turn on the Feedback Controller (CON: 0x0E)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x0E	-	-	-	-	-	-

Action: Activate Feedback Controller. There are two control modes: Position feedback and Current feedback. This command also enables motor driver.

14..Turn off the Feedback Controller (COF: 0x0F)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x0F	-	-	-	-	-	-

Action: De-activate Feedback Controller. This command also disables motor drivers.

15..Set Control mode (SCM: 0x10)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x10	FBC	-	-	-	-	-

Action: Select one of two feedback controllers: Position and Current.

FBC:

1: Current Control

0: Position Control

16.FIND the Limit and GO to the Offset (LGO: 0x11)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x11	CH&D	HO0H	HO0L	HO1H	HO1L	HH01

CH&D

b7	b6	b5	b4	b3	b2	b1	b0
CH				x	x	/DT	SDR

CH: Channel number, CH=0xF selects ALL channel

/DT:

0: Get Offset value from Byte 3 ~ Byte 7. (See 18)

1: Ignore data in Byte 3 ~ Byte 7.

SDR: Search direction. (See 18). If /DT is set, SDR is ignored.

HH01

b7	b6	b5	b4	b3	b2	b1	b0
S0	HH0			S1	HH1		

OFFSET0= Sign(S0)*(HO0L | HO0H<<8 | HH0<<16)

OFFSET1= Sign(S1)*(HO1L | HO1H<<8 | HH1<<16)

Sign(Sx) = 1 if Sx=0

$$\text{Sign}(S_x) = 0 \text{ if } S_x=1$$

Action:

1. Find the Limit switch and Index signal to get absolute position.
2. Go to the Offset position from the Index.
3. Set Encoder position value to Zero.
4. Activate position feedback controller.
5. Either Limit switch or Index signal is not found it sets fault bit and deactivates position controller.
6. LED displays the search and go status as HME.

NOTE: There are three Find-the-Limit modes. See 2.

Return: Returns HME with other status flags. See 2.

HME:

- 0x1: Start to find the Limit switch.
- 0x2: Limit switch-ON is found.
- 0x3: Limit switch-OFF is found and go backward to get ON.
- 0x4: Index signal detected.
- 0x5: Start to move to Offset position.
- 0x6: Arrived at Offset position. -> Success!

- 0xD: Fail to find backward limit switch signal.
- 0xE: Fail to detect Index signal
- 0xF: Fail to find Limit switch

17.Set Dead zone (SDZ: 0x20 + CH)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x20+CH	DZone	-	-	-	-	-

DZone: 0 ~ 255

Action: Set the value of Dead zone to remove FET's PWM null.

18.Set Home search parameter (SHP: 0x30 + CH)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x30+CH	SRL	SDR	OFFSET			

SRL: Search Limit. SRL is maximum number of turns to find limit switch.

SDR: Search direction. (1 or 0)

OFFSET: Offset from Index position.

Action: Set the value for Home search parameter.

19.Set Encoder resolution (SER: 0x38 + CH)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x38+CH	ENC_RE		-	-	-	-

ENC_RE:

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MDR	AuS	ENC_RES													

ENC_RES: Encoder Resolution

AuS: Auto Scale. AuS=1 sets Auto scale.

MDR: Motor Direction (1: CW, 0:CCW)

20.Set Maximum Acceleration and Velocity(SMAV: 0x40 + CH)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x40+CH	MACC		MVEL		-	-

MACC: Maximum Acceleration.

MVEL: Maximum Velocity.

Action: Set the value for Maximum Velocity and Acceleration.

21.Set Lower position limit (SLP: 0x50 + CH)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x50+CH	MPS	MPOS1				-

MPS

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	MPSS	MPSE

MPSS: MPOS1 update (1), MPOS1 ignore (0)

MPSE: Lower Position Limit Enable (1), Lower Position Limit Disable (0),

(NOTE) MPSE is set 1 when power ON.

MPOS1: Lower position limit

Action: Set the value for Lower position limit .

22.Set Upper position limit(SUP: 0x56 + CH)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
-------	--------	--------	--------	--------	--------	--------	--------	--------

0x01	BNO	0x50+CH	MPS	MPOS2	-
------	-----	---------	-----	-------	---

MPS

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	MPSS	MPSE

MPSS: MPOS2 update (1), MPOS2 ignore (0)

MPSE: Upper Position Limit Enable (1), Upper Position Limit Disable (0),

(NOTE) MPSE is set 1 when power ON.

MPOS1: Lower position limit

Action: Set the value for Upper position limit.

23.Set Home search Acceleration and Velocity(SHV: 0x60 + CH)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x60+CH	HMA	HMV1	HMV2	SRM	LIMD	-

HMA: Home Search Acceleration by 100. Actual acceleration will be calculated by $X_{acc} = HAM/100$.

HMV1: Maximum velocity to reach Limit switch.

HMV2: Maximum velocity to reach Offset position

SRM: Search Mode

0: Limit switch & Index

1: Limit switch only

2: No limit switch. LIMD is used to detect jam at the mechanical limit.

LIMD: PWM duty(%) for mechanical limit detection. This value will be used for Limit search mode 2(SRM=2). LIMD can be set by SJP command

Action: Set the values.

24.Gain override (GOV: 0x6F)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x6F	GOVW0	GOVW1	GDUR	-	-	-

GOVW0: Percent (%) override value of controller gain for channel 0.

GOVW1: Percent (%) override value of controller gain for channel 1.

GDUR: Duration in msec.

(NOTE:: $100 < GOVWx < 100$)

Action: Overall gain of the PID position controller changes to GOVW0x in GDUR ms from the current value.

25.Set Board number with NEW_BNO number(SNB: 0xF0)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0xF0	NEW_BNO	CANR	-	-	-	-

NEW_BNO: New Board number number

CANR: CAN rate

Action: Set the board number with NEW_BNO.

26.Set JAM and PWM saturation limit (SJP: 0xF2)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0xF2	JAM_LIM		PWM_LIM		LIMD	JAMD

JAM_LIM: JAM detection time in msec

PWM_LIM: PWM saturation detection time in msec.

LIMD: PWM duty(%) for Limit detection. This value will be used for Limit search mode 2. LIMD can be set by SHV command

JAMD: JAM limit in % duty

27.Set Error Bound (SEB: 0xF3)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0xF3	I_ERR		B_ERR		T_ERR	

I_ERR: Maximum Input difference error.

B_ERR: Maximum error.

T_MAX: Max Temperature Warning Temperature

28.Initialize the Board(IBR: 0xFA)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0xFA	-	-	-	-	-	-

Action: Initialize the board with default parameters.

Return:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x190+BNO + BOFF	CANR	1	BTY	VERSION			-	

Returns after completion of saving the value at the memory

29.Request Parameters (RPA: 0x24)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x24	PARM	-	-	-	-	-

Action: Returns value requested by PARM, where CN is Channel number and

OF=0 for CN=0,1,2

OF=5 for CN=3, 4

Return:

A. $PARM = CN * 6 + 1 + OF$:

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	Kp		Ki		Kd		EncL	EnCH

EnCH:

b7	b6	b5	b4	b3	b2	b1	b0
MDR	AuS	EnCH1					

$ENC_RES = EncL + (EnCH1 < 8)$;

AuS: Auto Scale. AuS(1) sets Auto scale.

MDR: Motor Direction (1: CW, 0:CCW)

B. $PARM = CN * 6 + 2 + OF$:

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	DZONE		HSD	HSM	HSL0	HSL1	HO0	HO1

DZONE: Dead zone

HSD: Home Search Direction

HSM: Home Search Mode

Home Search Limit = $HSL0 \mid (HSL1 < 8)$

C. $PARM = CN * 6 + 3 + OF$:

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	HO2	HO3	MP10	MP11	MP12	MP13	MP20	MP21

Home Offset = $HO0 \mid (HO1 < 8) \mid (HO2 < 16) \mid (HO3 < 24)$

Lower Position Limit = $MP10 \mid (MP11 < 8) \mid (MP12 < 16) \mid (MP13 < 24)$

D. $PARM = CN * 6 + 4 + OF$:

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	MP22	MP23	MAL	MAH	MVL	MVH	SML	SMH

Upper Position Limit = MP20 | (MP21<<8) | (MP22<<16) | (MP23<<24)

Maximum Acceleration = MAL | (MAH<<8);

Maximum Velocity = MVL | (MVH<<8);

Maximum PWM = SML | (SMH<<8);

E. PARM=CN*6 + 5 + OF:

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	CLL	CLH	RSRV	RSRV	KPtL	KPtH	KDtL	KDtH

Current Limit = CLL | (CLH<<8);

Kpt = KPtL | (KPtH<<8);

Kdt = KDtL | (KDtH<<8);

F. PARM=CN*6 + 6+ OF:

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	KFtL	KFtH	PATR2	PATR3	PATR4	PATR5	PATR6	PATR7

KFt = KFtL | (KFtH<<8);

(SUM of Bytes from A to F)&0xFF = 0

G. PARM=20:

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	BNO	RSRV	CANL	CNAH	BTY	RSRV	HMAL	HMAH

BNO: Board Number

CANL | (CANH<<8): CAN rate

BTY : Board Type

HMAL | (HMAH<<8) : Maximum Acceleration for Home limit search

H. PARM=21

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	HMV1L	HMV1	HMV2L	HMV2H	JAML	JAMH	PWML	PWMH

HMV1L | (HMV1H<<8) : Maximum Velocity for Home limit search

HMV2L | (HMV2H<<8) : Maximum Velocity to Offset position

JAML | (JAMH<<8): JAM error detection time. unit: 0.1sec.

PWML | (PWMH<<8): PWM saturation error detection time. unit: 0.1sec.

I. PARM=22

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	LIMD	PWMD	IERRL	IERRH	BERRL	BERRH	EERRL	EERRH

LIMD: PWM duty(%) for Limit detection. This value will be used for Limit search mode 2.

PWMD: PWM duty(%) for JAM detection.

IERRL | (IERRH<<8): Maximum Input difference error.

BERRL | (BERRH<<8): Maximum error.

EERRL | (EERRH<<8): Maximum error. for encoder failure

2. FT Sensor Board

Command Message for Sensor Boards (Message ID=0x01)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Set & Req. Board Info.	BNO	0x01	CANR	-	-	-	-	-
Req. Board Parameters	BNO	0x24	PARM	-	-	-	-	-
Req. to execute NULL	BNO	0x81	EFS	-				
Set FT matrix constant0	BNO	0xA0	SFT00		SFT01		SFT02	
Set FT matrix constant1	BNO	0xA1	SFT10		SFT11		SFT12	
Set FT matrix constant2	BNO	0xA2	SFT20		SFT21		SFT22	
Set Inclino. scale factor	BNO	0xA5	SIF0		SIF1		SIF2	
Set Board Number(BNO) & filter freq.	BNO	0xA8	NEW_BNO	FREQ10		-	-	-
Initialize Board	BNO	0xFA	0xAA	-	-	-	-	-

Read Message for Sensor Boards (Message ID=0x02)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Req. FT ¹⁾ and Tilt in digit ²⁾	SBNO	0x00		-	-	-	-	-
Req. FT data ¹⁾ w/scale and Tilt ⁴⁾ w/scale	SBNO	0x02	-	-	-	-	-	-
Req. FT data ³⁾ w/scale and Tilt ²⁾ in digit	SBNO	0x03	-	-	-	-	-	-
Req. FT data ¹⁾ in digit and Tilt ⁴⁾ w/scale	SBNO	0x04	-	-	-	-	-	-
Req. FT data in digit ¹⁾	SBNO	0x11	-	-	-	-	-	-
Req. FT data w/scale ³⁾	SBNO	0x12	-	-	-	-	-	-
Req. Tilt data in digit ²⁾	SBNO	0x21	-	-	-	-	-	-
Req. Tilt data/ scale ⁴⁾	SBNO	0x22	-	-	-	-	-	-
Req. Gyro & Temp data ⁵⁾	SBNO	0x13						

SBNO: If SBNO=0xFF all sensor boards accept the command.

Return Message from Sensor Boards (SBNO = BNO – 0x2F)

Description	MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
1) Send FT data	0x40+SBNO + BOFF	MX		MY		FZ		-	-
2) Send Tilt data	0x50+SBNO	SX		SY		SZ		-	-
3) Send FT data	0x40+SBNO	Mx100		My100		Fz10		-	-
4) Send Tilt data	0x50+SBNO	Sx100		Sy100		Gz100		-	-
5) Send Gyro-rate & Temperature	0x50+SBNO	GZL		Temp10		-	-	-	-

$M_x = M_{x100}/100$: M_x is a Moment x Nm
 $M_y = M_{y1000}/100$: M_y is a Moment y Nm
 $F_z = F_{z10}/10$: F_z is a Force z in N.
 $S_x = S_{x100}/100$ S_x is a Acceleration in m/sec^2
 $S_y = S_{y100}/100$ S_y is a Acceleration in m/sec^2
 $G_z = G_{z100}/100$ G_z is a Gravitational Acceleration in g ($9.8m/sec^2$)
 SX, SY, SZ are non-scaled digit data from 3-axis Accelerometer.
 MX, MY, FZ are non-scaled digit data from 3-axis F/T sensor.

Temp = Temp10/10 : Temperature in degree C.

$$\begin{Bmatrix} M_x \\ M_y \\ F_z \end{Bmatrix} = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.001 \end{pmatrix} * \begin{bmatrix} SFT00 & SFT01 & SFT02 \\ SFT10 & SFT11 & SFT12 \\ SFT20 & SFT21 & SFT22 \end{bmatrix} * \begin{Bmatrix} MX \\ MY \\ FZ \end{Bmatrix}$$

$$S_{x100} = SIF0 * S_x * 1e-3$$

$$S_{y100} = SIF1 * S_y * 1e-3$$

$$G_{z100} = SZ / SIF2 * 1e3$$

Default Value of SFTxy

	Foot Sensor			Wrist Sensor		
	SFTx0	SFTx1	SFTx0	SFTx1	SFTx2	SFTx2
x=0	-250	0	0	12	0	0
x=1	0	-250	0	0	-12	0
x=2	0	0	-3000	0	0	-250

Default Value of SIFx

SIF0	SIF1	SIF2
600	600	8000

Detail Description of Command Message

1. Set and Request Board Information(RBI: 0x01)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x01	CANR	-	-	-	-	-

CANR: CAN rate in msec.

Default: 5(ms)

Return:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x190+BNO + BOFF	CANR	1	BTY	VERSION			-	

BTY, VERSION: See Motor Board section for details.

2. Request to execute NULL the sensors (RNL: 0x81)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x81	EFS	-	-	-	-	-

EFS: 0x00 NULL F/T sensors

: 0x02 NULL Inclinometers

3. Set FT sensor Matrix coefficient0 (SFC0: 0xA0)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0xA0	SFT00		SFT01		SFT02	

4. Set FT sensor Matrix coefficient1 (SFC1: 0xA1)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0xA1	SFT10		SFT11		SFT12	

5. Set FT sensor Matrix coefficient2 (SFC2: 0xA2)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0xA2	SFT20		SFT21		SFT22	

SFT 00 ~ SFT22: Set FT sensor coefficient matrix

Return:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x190+BNO + BOFF	CANR	1	BTY	VERSION			-	

Returns after completion of saving the value at the memory

6. Set Inclinator(Accelerometer) scale factor (SAS: 0xA5)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0xA5	SIF0		SIF1		SIF2	

(NOTE)

$$Sx100 = SIF0 * SX * 1e-3$$

$$Sy100 = SIF1 * SY * 1e-3$$

$$Gz100 = SZ / SIF2 * 1e3$$

Return: same as above.

7. Set Board Number and Filter frequency (SNBF: 0xA8)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0xA8	NEW_BNO	FREQ10		-	-	-

NEW_BNO: Set the Board number with NEW_BNO.

FREQ = FREQ/10: FREQ is a cut off frequency of the first order Low pass filter.

8. Initialize with default value(IDF: 0xFA)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0xFA	0xAA	-	-	-	-	-

Action: Initialize the Board.

Return: Same as above.

9. Request Parameters (RPA: 0x24)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0x24	PARM	-	-	-	-	-

Action: Returns value requested by $PARM = CN * 6 + 1 + OF$, where CN is Channel number.

OF=0 for CN=0,1,2

OF=5 for CN=3, 4

Return:

A. PARM=1:

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	SFT00		SFT01		SFT02		FREQ	

B. PARM=2:

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	SFT10		SFT11		SFT12		SPARE	

C. PARM=3:

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	SFT20		SFT21		SFT22		SPARE	

FREQ: Set FT sensor coefficient matrix

$$\begin{Bmatrix} M_x \\ M_y \\ F_z \end{Bmatrix} = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.001 \end{pmatrix} * \begin{bmatrix} \text{SFT00} & \text{SFT01} & \text{SFT02} \\ \text{SFT10} & \text{SFT11} & \text{SFT12} \\ \text{SFT20} & \text{SFT21} & \text{SFT22} \end{bmatrix} \begin{Bmatrix} M_x \\ M_y \\ F_z \end{Bmatrix}$$

(NOTE)

1. M_x , M_y , F_z are measured in digit value from A/D converter. M_x and M_y are moment in Nm, and F_z is force in N.
2. $\text{Freq} = \text{FEEQ}/10$: Freq is cut off frequency of 1st order low pass filter.

D. PARM=4:

MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x1C0 +BNO	SIF0		SIF1		SIF2		SPARE	

SIF0: Scale factor for Accelerometer SX

SIF0: Scale factor for Accelerometer SY

SIF0: Scale factor for Accelerometer SZ

(NOTE)

$$S_x = \text{SIF0} * S_x * 1e-5 \text{ (degree)}$$

$$S_y = \text{SIF1} * S_y * 1e-5 \text{ (degree)}$$

$$S_z = (S_z - \text{SIF2}) / \text{SIF2} \text{ (g/g)}$$

S_x , S_y , S_z are measured in digit value from Accelerometer. S_x and S_y are inclined angle and S_z is g multiple in z-direction.

3. Power Control Board

Command Message for Power Control Board (Message ID=0x01)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Set & Req. Board Info. ¹⁾	BNO	0x01	CANR	-	-	-	-	-
Set Switch function	BNO	0x81	SFUNC	-	-	-	-	-
Request Alarm	BNO	0x82	ALRM					
Request Beep	BNO	0x83	BDUR					
Req. Voltage and current	BNO	0xE0	-	-	-	-	-	-
Req. Time and Status	BNO	0xE1	-	-	-	-	-	-

1. Set and request Board Information(RBI: 0x01)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Set & Req. Board Info. ¹⁾	BNO	0x01	CANR	-	-	-	-	-

CANR: CAN rate in msec.

Default: 5(ms)

Return:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x190+BNO + BOFF	CANR	-	BTY	VERSION			-	

BTY, VERSION: See Motor Board section for details.

2. Set Switch function(SSF: 0x81)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Set & Req. Board Info. ¹⁾	BNO	0x81	SFUNC	-	-	-	-	-

SFUNC

- 0x00: SEC_TIME reset to Zero
- 0x01: Set 12VSEN and LEDR -> Turn On 12V for sensor.
- 0x02: Clear 12VSEN and LEDR -> Turn Off 12V for sensor.
- 0x04: Set M_BTN_ON -> Execute power on sequence to turn on 48V.
- 0x05: Set M_BTN_OFF -> Execute power off sequence to turn off 48V
- 0x07 Set BEEPF -> Beeper Enable
- 0x08 Clear BEEPF and Turn off Beep. -> Beeper Disable
- 0x0A: Set TGL_PC1 -> Turn on PC1.
- 0x0B Clear TRL_PC1 -> Turn off PC1.
- 0x0C: Set TGL_PC2 -> Turn on PC2.
- 0x0D Clear TRL_PC2 -> Turn off PC2.

3. Request Alarm(ALM: 0x82)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Set & Req. Board Info. ¹⁾	BNO	0x82	ALRM	-	-	-	-	-

ALRM

0x00: Alarm OFF

0x01: Alarm sound 1, Beeper Enable

0x02: Alarm sound 2, Beeper Enable

0x03: Alarm sound 3, Beeper Enable

0x04: Alarm sound 4, Beeper Enable

4. Request Alarm(ALM: 0x82)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Set & Req. Board Info. ¹⁾	BNO	0x82	BDUR	-	-	-	-	-

BDUR: Beeps for BDUR*0.1 second.

5. Request voltage and current(RVC: 0xE0)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Req. Voltage and Current	BNO	0xE0	-	-	-	-	-	-

Return

Description	MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
Send Voltage and Current	0x60+BNO	VOLT100		AMP100		WATT100		-	-

BNO=0x14

VOLT = VOLT100/100: Main voltage (48V) measured in V.

CURR= AMP100/100: Main power (48V) current in Ampere.

WATT= WATT100/10: Accumulated Power in WH.

6. Request Time and Status information(RTD: 0xE1)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Req. Time & status info.	BNO	0xE1	-	-	-	-	-	-

Return

Description	MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
Send Time	0x60+BNO	SMF	0x00	SEC_TIME	-	-	-	-	-

SMF

b7	b6	b5	b4	b3	b2	b1	b0
SW_12_SEN	SW_12_DR	SW_48_O	SW_48_R	SW_PC1	SW_PC2	BEEPF	BATT

BATT:	1: Battery operated,	0: External Power operated
BEEPF:	1: Beeper activated,	0: Beeper disabled
SW_PC2:	1: PC2 is ON	0: OFF
SW_PC1:	1: PC1 is ON	0: OFF
SW_48_R	1: 48V Pre-switch is ON.	0: OFF.
SW_48_O	1: 48V Main-switch is ON.	0: OFF.
SW_12_DR	1: 12V for motor driver is ON	0: OFF
SW_12_SEN	1: 12V for sensor is ON	0: OFF

SEC_TIME: Time elapsed since power ON or since reset.

4.IMU Board

Command Message for IMU Board (Message ID=0x01)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Set & Req. Board Info. ¹⁾	BNO	0x01	CANR	-	-	-	-	-
Req. to execute NULL	BNO	0x81	-	-	-	-	-	-
Req. to execute Calib.	BNO	0x82	-	-	-	-	-	-
Req. parameters	BNO	0x24	-	-	-	-	-	-
Set board with NEW_BNO	BNO	0xA8	NEW_BNO	-	-	-	-	-

Read Message for IMU Boards (Message ID=0x02)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Req. Angle and Rate ¹⁾	SBNO	0x00	1	-	-	-	-	-

Return Message from IMU Boards (SBNO = BNO – 0x2F)

Description	MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
1) Send FT data	0x50+SBNO	ANGL_X		ANGL_Y		RATE_X		RATE_Y-	

Detail Description of Command Message

1. Set and request Board Information(RBI: 0x01)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Set & Req. Board Info. ¹⁾	BNO	0x01	CANR	-	-	-	-	-

CANR: CAN rate in msec.

Default: 5(ms)

Return:

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x190+BNO + BOFF	CANR	-	BTY	VERSION			-	

BTY, VERSION: See Motor Board section for details.

2. Request to Execute the sensor NULL(RNL: 0x81)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Set & Req. Board Info. ¹⁾	BNO	0x81		-	-	-	-	-

Action:

- Nulling is a process by which zero-levels of rate gyros are determined. When the Nulling is commanded, IMU gathers data from rate gyros for one second then average it for zero-levels. Therefore, the IMU must be in static state during the Nulling process.
- Nulling must be conducted before sending a message which requests angle and rate. IMU does not respond to the request before the Nulling process is finished.

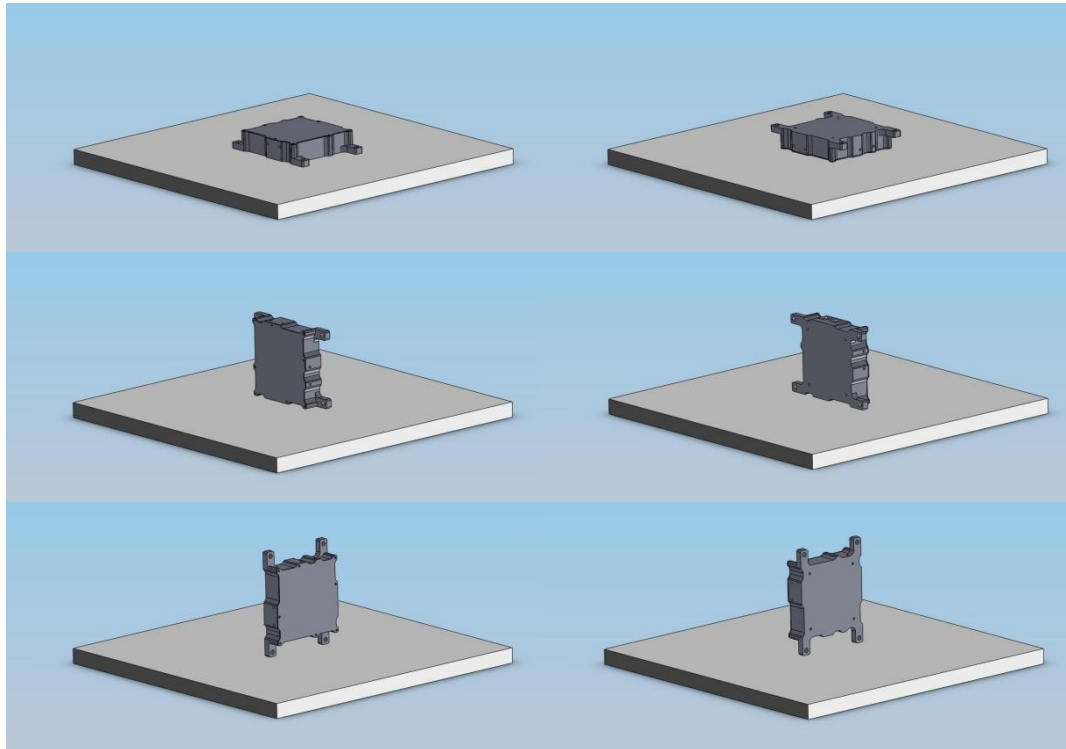
3. Request to Execute Calibration(REC: 0x82)

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Req. to execute Calib.	BNO	0x82		-	-	-	-	-

Action:

- The accelerometer in IMU measures absolute acceleration in all three directions. Before the first time use, IMU must be calibrated to determine the scale and the bias of the accelerometer. Calibration is conducted for 3 axis in six directions (x, -x, y, -y, z, -z) as follows:
 - a. Connect the IMU unit.
 - b. Place the IMU on a leveled plate.

- c. Send REC (or Push the button "3" in HUBO-i).
- d. Wait 10 seconds.
- e. Place the IMU on other faces and repeat the process, c and d.
- f. After all the faces are done, disconnect the IMU from the power and restart it.



4. Request Parameters

Description	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Req. Parameters. ¹⁾	BNO	0x24	PRF	-	-	-	-	-

Return:

PRF=1

Description	MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
Send Time	0x1C0+BNO	ACC_X_GAIN		ACC_Y_GAIN		ACC_Z_GAIN		-	-

PRF=2

Description	MsgID	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
Send Time	0x1C0+BNO	ACC_X_BAIS		ACC_Y_BAIS		ACC_Z_BAIS		-	-

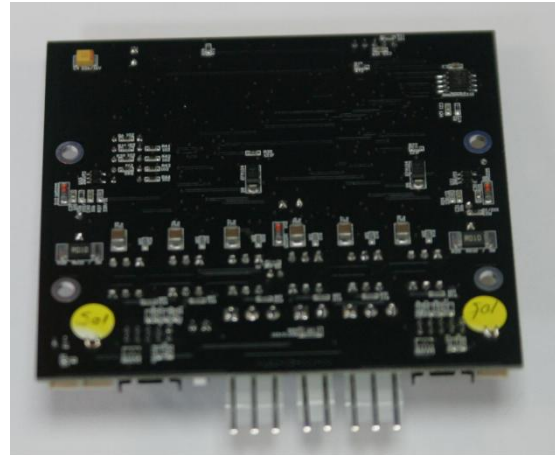
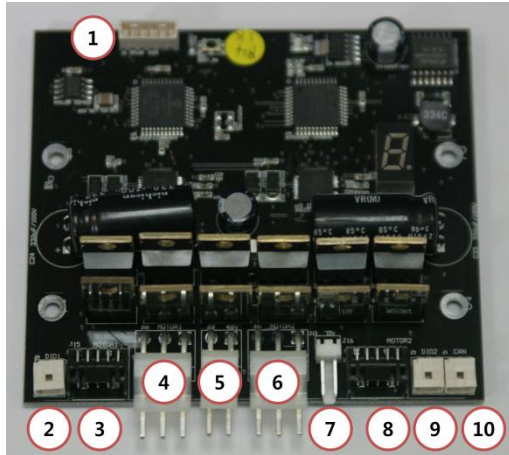
5. Set Board Number (0xA8)

MsgID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	BNO	0xA8	NEW_BNO	-	-	-	-	-

Action: Set the board number with NEW_BNO.

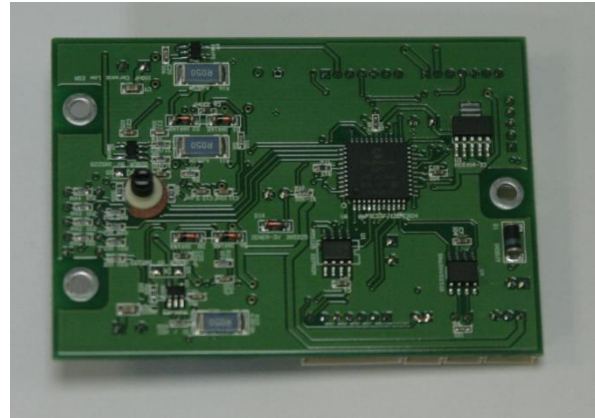
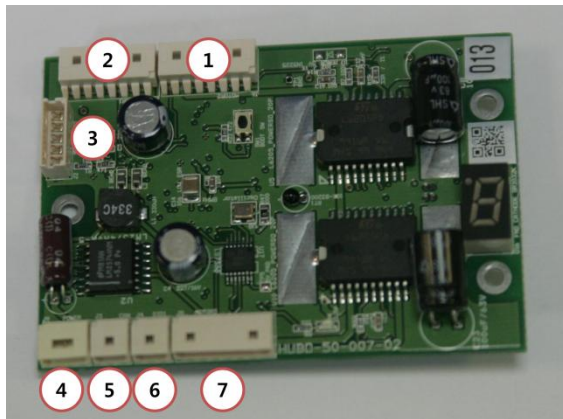
Connection Lines

1. 2ch BLDC motor controller(Lower body, shoulder pitch and roll)



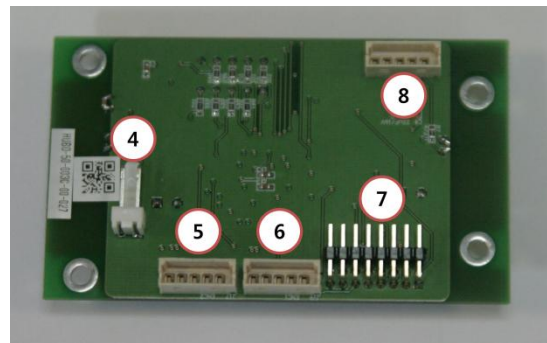
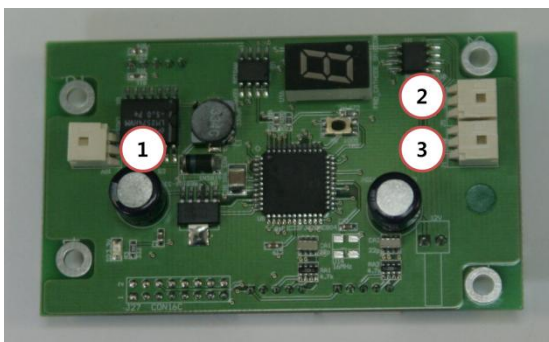
1. MPLAB debugger line
2. Limit switch 1
3. Encoder line 1
4. Motor 1
5. 48V motor power line
6. Motor 2
7. 12V board power line
8. Encoder line2
9. Limit switch 2
10. CAN line

2. 3ch DC motor controller (Neck)



1. Motor 2
2. Motor 1
3. MPLAB debugger line
4. 48V-12V-GND line
5. CAN line
6. Limit switch 1
7. Motor 3

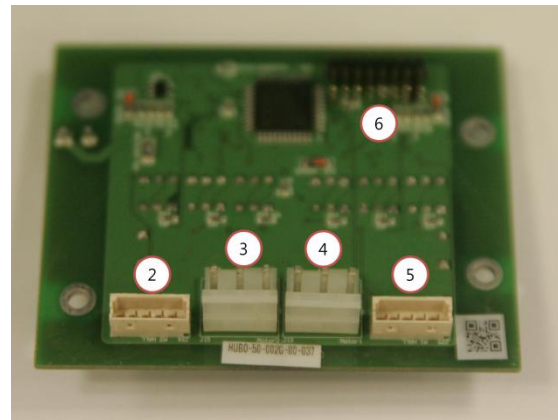
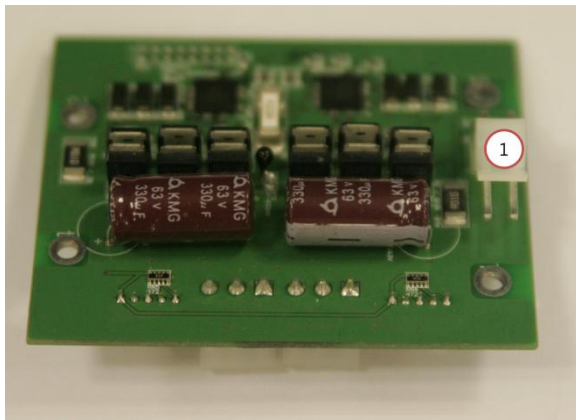
3. 2ch BLDC motor controller (Shoulder yaw and elbow)



1. Limit switch 1
2. Limit switch 2
3. CAN line

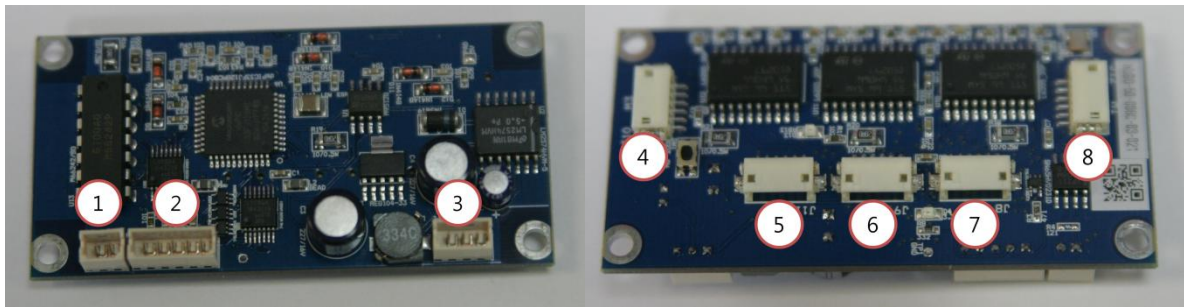
4. 12V board power line
5. Encoder line 2
6. Encoder line 1
7. Controller-Amp connect line
8. MPLAB debugger line

4. 2ch BLDC motor amp. (Shoulder yaw and elbow)



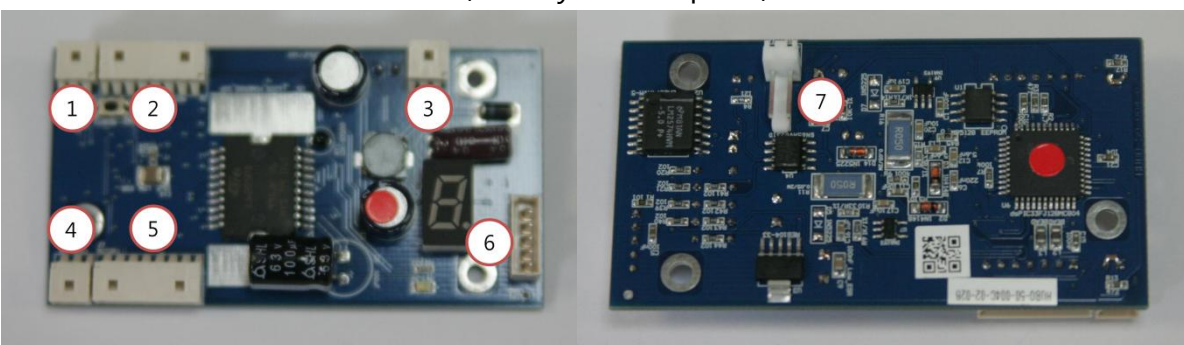
1. 48V motor power line
2. Hall sensor 2
3. Motor 2
4. Motor 1
5. Hall sensor 1
6. Controller-Amp connect line

5. 5ch DC motor controller (Hand)



1. CAN line
2. MPLAB debugger line
3. 12V-12V-GND line
4. Motor 3
5. Motor 4
6. Motor 0
7. Motor 2
8. Motor 1

6. 2ch DC motor controller (Wrist yaw and pitch)



1. Limit switch 2
2. Motor 2
3. CAN line
4. Limit switch 1

5. Motor 1
6. MPLAB debugger line
7. 12V board power line