Hitec Robot Servos

2007-09-23

HSR-5498SG
HSR-8498HB
HSR-5980SG *
HSR-5990TG *

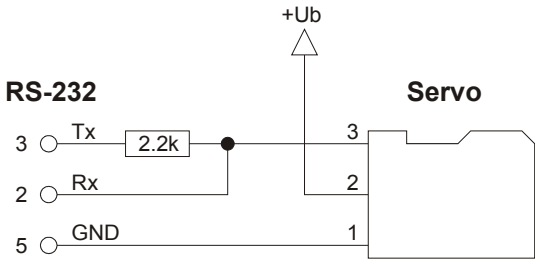
(*) not tested

Technical specifications

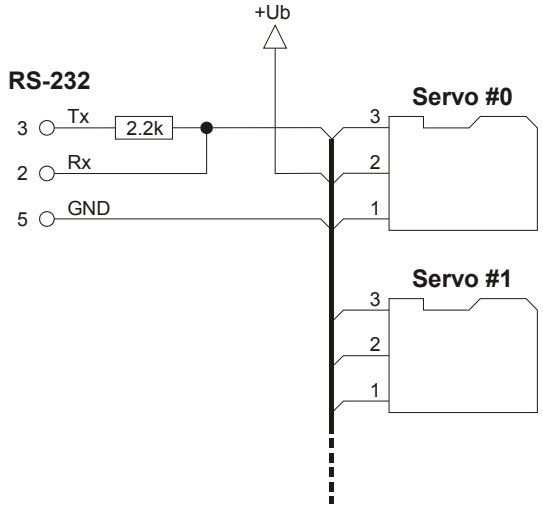
(according to Hitec homepage)

		HSR-8498HB	***HSR-5498SG***	***HSR-5980SG***	**HSR-5990TG**
					
				Economical version of 5990TG: Steel gear, no heatsink.	
Motor			cored metal brush	coreless metal	coreless metal
Gear		carbonite	metal	metal	titanium
Torque 6.0 V	Ncm	7.4	11	24	24
7.4 V		9.0	13.5	30	30
Speed 6.0 V	Sec/60°	0.2	0.22	0.17	0.17
7.4 V		0.18	0.19	0.14	0.14
Dimension (LxBxH)	mm	40 x 20 x 47	40 x 20 x 37	40 x 20 x 37	40 x 20 x 37
Weight	g	55	59.8	68	68
Price	€	52.90	59.00	89.90	139.00

Serial line connection



Transmit and send data are multiplexed on one communication line by a wired-or connection. With this connection scheme, the host must keep the RS-232 line high to receive data, since the servos communication output is an open-collector without internal pull-up resistor. The signal levels in the above connection scheme do not conform to the RS-232 specification. If your RS-232 fails to receive data from the servo you must use a RS-232 level converter.
Up to 256 servos can be connected parallel on one communication line (wired-or bus). Note that before multiple servos can be used parallel on one bus, different id's must be assigned to the individual servos, which is only possible with a single servo connection.



Communication protocol

Communication parameters:

Baudrate	19.2 kbaud
Databits	8
Stopbits	2
Parity	none
Handshake	none

Each command frame has 7 bytes. The last 2 bytes contain the answer from the servo controller. Since the controller can only pull down the communication line, the host must pull up the line by sending 0x00. Different answers from multiple servos on one communication line will result in a bus collision.

	0	*1*	*2*	*3*	*4*	*5*	*6*
Host:	0x80	Command	Param1	Param2	Checksum	0x00	0x00
Servo:	High-Z					Return1	Return2

The checksum byte adds up the sum of bytes 0:3 to a multiple of 256:

```
Sum[0:4] % 256 == 0x00
Checksum = 256 - (0x80 + Command + Param1 + Param2) % 256
```

Command set

Command	*Param1*	*Param2*	*Return1*	*Return2*	*Description*
0..0x7F (id)	pos_high	pos_low	0x00	0x00	Set servo #id target position
0xE1	addr	0x00	data	0x03	Read EEPROM
0xE2	addr	data	0x03	0x03	Write EEPROM
0xE3	addr	0x00	data	0x03	Read memory
0xE4	addr	data	0x03	0x03	Write memory
0xE5	0x00	0x00	pos_high	pos_low	Read position
0xE6	pos_high	pos_low	0x00	0x00	Set target position
0xE7	0x00	0x00	version	id	Read version & id
0xE8	0x00	0x00	pulsewidth	voltage	Read pulsewidth & voltage
0xE9	0..0x7F (id)	speed	pos_high	pos_low	Set servo #id speed & read position
0xEA	0x00	1 / 2 / 3	0x03	0x06	Select control parameter set
0xEB	0x00	1 / 0	0x03	0x06	Set go / stop
0xEC					
0xED					
0xEE					
0xEF	0x00	0x00	0x03	0x06	Release

Command description

Command	**0..0x7F** (id)	**Set servo #id target position** This allows to set the position of a single servo. **Note:** This command re-enables ALL released motors on the bus (see command 0xEF).
Param1	pos_high	
Param2	pos_low	
Return1	0x00	
Return2	0x00	

Command	**0xE1**	**Read EEPROM** Read one byte from the EEPROM. Readable addresses are 0..0xFF. Addresses >0x2C always read as 0x00. Known EEPROM addresses are tabulated below.
Param1	addr	
Param2	0x00	
Return1	data	
Return2	0x00	

Command	**0xE2**	**Write EEPROM** Write one byte to the EEPROM. Writable addresses are 0..0x2C. When writing addresses > 0x2C every 3rd write attempt will fail, and return 0x00,0x00 ### ? ###
Param1	addr	
Param2	data	
Return1	0x03	
Return2	0x03	

Command	**0xE3**	**Read memory** Read one byte from the controllers volatile memory. Readable addresses are 0..0xFF. Known memory addresses are tabulated below.
Param1	addr	
Param2	0x00	
Return1	data	
Return2	0x03	

Command	**0xE4**	**Write memory** Write one byte to the controllers volatile memory. The addressable memory range is 0..0xFF, but not all addresses are writable. When a write attempt fails the controller returns 0x00,0x00. ### *TODO: Tabulate writable addresses.* ###
Param1	addr	
Param2	data	
Return1	0x03	
Return2	0x03	

Command	**0xE5**	**Read position** Read the current position. Use command 0xE9 to address multiple motors on one bus.
Param1	0x00	
Param2	0x00	
Return1	pos_high	
Return2	pos_low	

Command	**0xE6**	**Set target position** This sets the target position for all motors on the bus. Use command 0..0x7F to address multiple motors.
Param1	pos_high	
Param2	pos_low	
Return1	0x00	
Return2	0x00	

Command	**0xE7**	**Read version & id**
Param1	0x00	
Param2	0x00	
Return1	version	
Return2	id	

Command	**0xE8**	**Read pulsewidth & voltage** Reads the driving pulsewidth applied to the motor and the supply voltage. Voltage scale = [0.03522 V]
Param1	0x00	
Param2	0x00	
Return1	pulsewidth	
Return2	voltage	

Command	**0xE9**	**Set servo #id speed & read position** This command sets the speed of a single servo on the bus and reads its current position. Valid speed range is 1..255.
Param1	0..0x7F (id)	
Param2	speed	
Return1	pos_high	
Return2	pos_low	

Command	**0xEA**	**Select control parameter set** This command sets the control parameters (P-gain,D-gain,Dead-band) according to the parameter sets 1-3 stored in the EEPROM. Values other than 1-3 always select parameter set 3. Parameter set 1 is loaded at power-on.
Param1	0x00	
Param2	1 / 2 / 3	
Return1	0x03	
Return2	0x06	

Command	**0xEB**	**Set go/stop** Sending a 0 will suspend subsequent changes of the target position until sending a value >0. This command does NOT disable the position control. To disable the position control mechanism use command 0xEF.
Param1	1 / 0	
Param2	0x00	
Return1	0x03	
Return2	0x06	

Command	**0xEF**	**Release** This command stops the servos position control mechanism. The joint can be moved by hand afterwards. **Note:** Any following move command on ANY servo on the bus will re-enable all servos. **A re-enabled servo will move to the last set target position with highest possible speed.**
Param1	0x00	
Param2	0x00	
Return1	0x03	
Return2	0x06	

Bug: When a new target position is sent to a motor the trajectory generator uses the last trajectory set-point as starting point, which may be different from the actual position in case the joint was moved by hand after a release command(0xEF). A re-enabled motor will therefore start to move with highest speed in the direction of the old target position until it catches up with the new trajectory.

Workaround: (Single servo only!) Before sending a new target position, read the actual joint position and write it to memory location `0x07:0x06` (this is where the actual trajectoy set-point is stored).

EEPROM locations

Unless otherwise stated, 2-byte values are stored most significant first.

	Default settings (**)				
Address	***8498SG***	***5498SG***	***5980SG / 5990TG***		
0x00	0x50	0x50	0x14	P-gain	Parameter set 1
0x01	0xB4	0xB4	0x64		
0x02	0x03	0x04	0x04	Dead-zone	
0x03	0x1E	0x0A	0x05	D-gain	
0x04	0x01	0x02	0x02	D-gain ? (not reloaded at parameter change)	
0x05	0x01	0x01	0x01		
0x06	0xFF	0xFF	0xFF	Speed	
0x07	0x00	0xFF	0x00	Center position (**)	
0x08	0x00	0xE4	0x00		
0x09	0x02	0x02	0x02	(550) Minimum position limit.	
0x0A	0x26	0x26	0x26		
0x0B	0x09	0x09	0x09	(2450) Maximum position limit.	
0x0C	0x92	0x92	0x92		
0x0D	0x00	0x00	0x00		
0x0E	0x10	0x10	0x10		
0x0F	0x03	0x03	0x03		
0x10	0xF0	0xF0	0xF0		
0x11	0x05	0x05	0x05	(1500) Center position ?	
0x12	0xDC	0xDC	0xDC		
0x13	0xB4	0xBE	0xB4	Left limit (**)	
0x14	0xB4	0xB9	0xB4	Right limit (**)	
0x15	0x13	0x13	0x13		
0x16	0x88	0x88	0x88		
0x17	0x00	0x00	0x00		
0x18	0x00	0x00	0x00		
0x19	0x05	0x05	0x05	(1500) Center position ?	
0x1A	0xDC	0xDC	0xDC		
0x1B	0x29	0x29	0x29	Direction: 0x29: forward, 0x28: reverse (**)	
0x1C	0x28	0x28	0x28		
0x1D	0xD2	0xD2	0xD2		
0x1E	0x05	0x0A	0x0A		
0x1F	0x64	0x64	0x1E	P-gain	Parameter set 2
0x20	0xC8	0xC8	0x78		
0x21	0x04	0x05	0x04	Dead-zone	
0x22	0x32	0x0F	0x07	D-gain	
0x23	0x01	0x02	0x02	D-gain ? (not reloaded at parameter change)	
0x24	0x3C	0x3C	0x0A	P-gain	Parameter set 3
0x25	0xA0	0xA0	0x50		
0x26	0x02	0x03	0x04	Dead-zone	
0x27	0x0A	0x05	0x03	D-gain	
0x28	0x01	0x02	0x02	D-gain ? (not reloaded at parameter change)	
0x29	0x00	0x00	0x00	ID	
0x2A	0x0A	0x0A	0x0A		
0x2B	0x02	0x02	0x02		
0x2C	0xCF	0x0E	0xB3	EEPROM Checksum (*)	

****** According to HMI-Servo programmer v 1.0.2. Not tested.

***** The EEPROM Checksum adds up the location 0x00..0x2B to a multiple of 256:

```
Sum[0:0x2C] % 256 == 0
```

```
Checksum = 256 - sum[0:0x2B] % 256
```

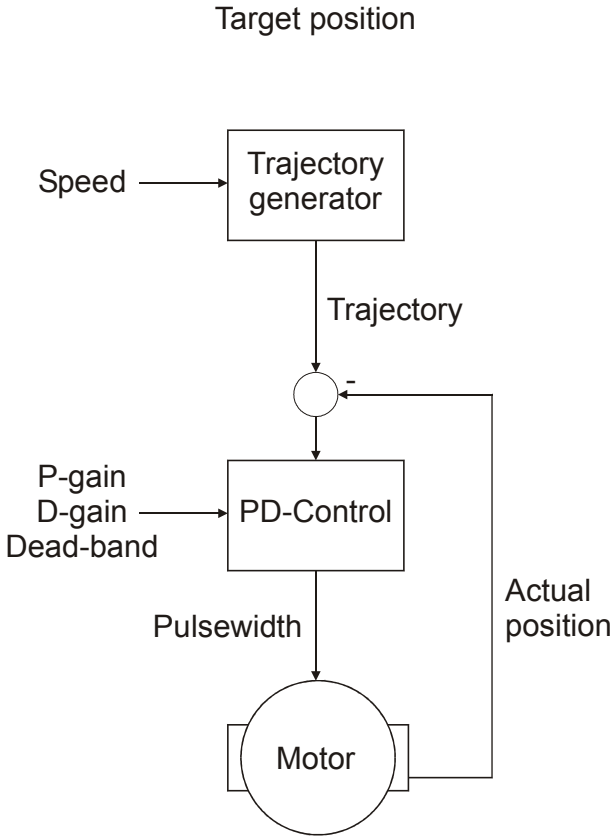
When the motor is powered on with a wrong checksum, all communication behaves normal but the motor will not move.

Memory locations

Unless otherwise stated, 2-byte values are stored most significant first.

Address	
0x06 0x07	(low/high) Current trajectory set-point
0x80 ... 0x9E	Correspond to EEPROM addresses 0x00..0x1E. Cells are loaded from EEPROM at power-on.
0xC3	Speed (sometimes updated from 0x86)
0xA5 0xA6	Target position
0xA7 0xA8	Actual position (Not reliable if motor is released ### ? ###)
0xA9 0xAA	Target position – actual position
0xC9	Go / stop

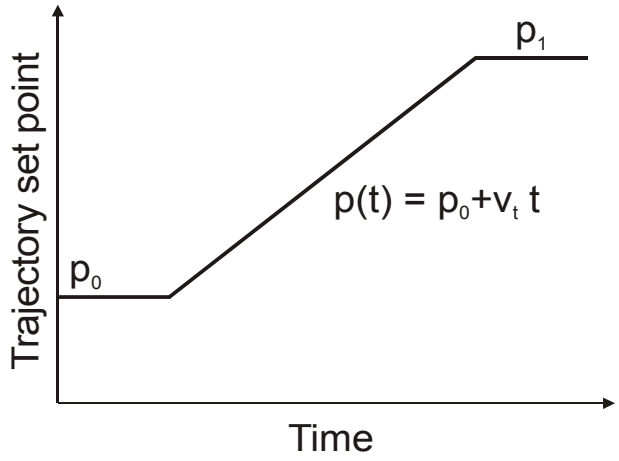
Motion control



TODO: Influence of P,D?

Trajectory generation

The trajectory generator generates a linear trajectory (constant angular velocity) starting from the last trajectory set point p_0 to the actual target position p_1:



The current trajectory set point p(t) can be read from the memory location 0x07:0x06 (least significant first!). The trajectory angular velocity v_t is proportional to the set speed parameter s. When sending commands during an ongoing motion, the addressed servo's trajectory velocity has

been found to decreases linearly with the the command frequency:

$$v_t = As\ (1 - Br)$$

v_t	trajectory angular velocity [tics/sec]
s	speed setting
r	command rate [1/sec]
A =	37.46(79) tics/sec
B =	0.001106(87) sec

The velocity reduction has been measured for commands 0xE3 (read memory) and 0xE9 (set speed/read position).

TODO: Verify velocity reduction with other commands.

Note that the maximum reachable motor velocity depends on the motor type and the supply voltage.

	Voltage		***8498HB***	***5498SG***	***5980SG/5990TG***
Maximum Velocity *	6V	[tics/sec]	3000	2727	3529
	7.4V		3333	3158	4286
Maximum possible speed setting	6V	[speed]	80	73	94
	7.4V		89	84	114

* According to Hitec webpage.