SB Servo: Digital Serial Servo Motors

User Manual

OPEN-SOURCE DIGITAL SERVO MOTOR

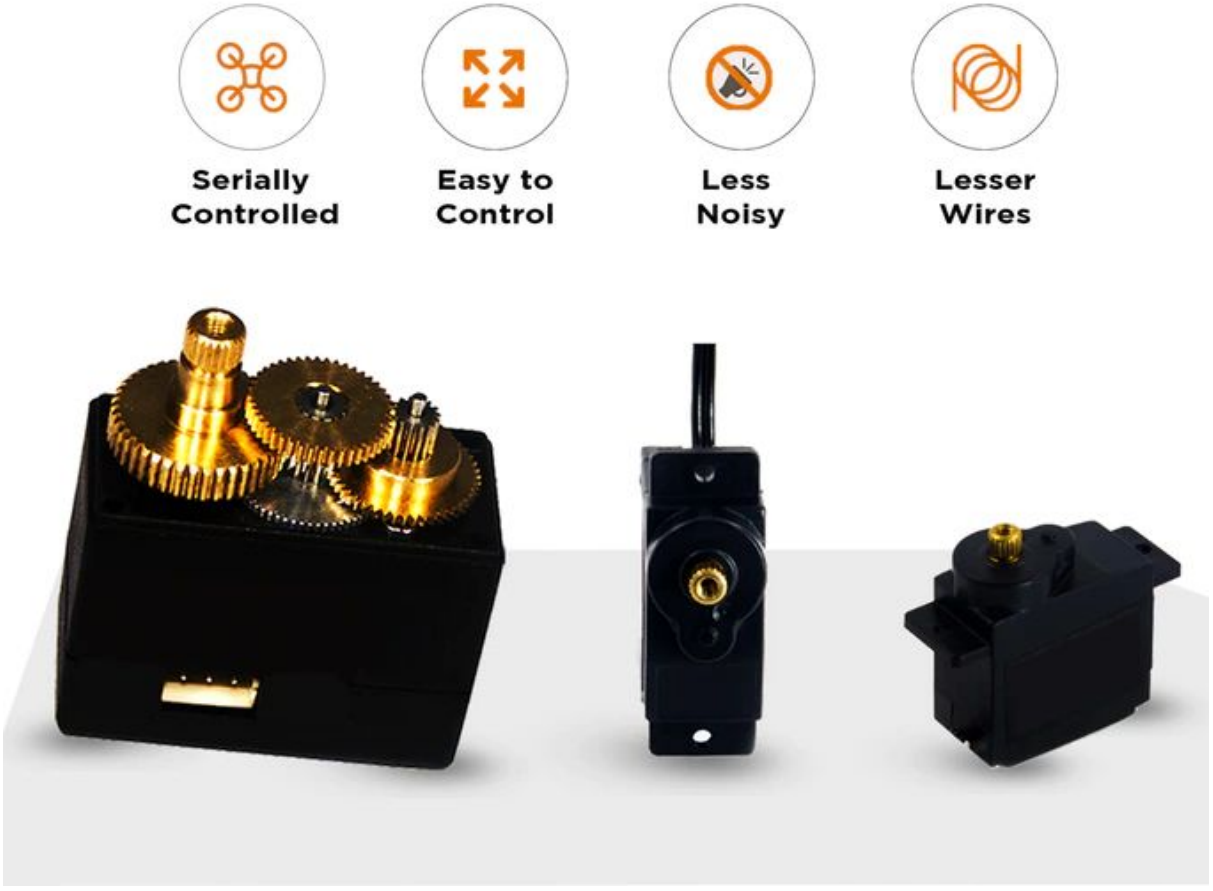


Table Of Contents

Introduction

SB Serial Servos are a series of high torque motors made with metal gears. These motors are very precise and easy to control which makes them an excellent choice for all the robotic applications. These are digital servo motors which can be controlled by any hardware using serial communication.

There are two servo variants available for SB-SS Motors:

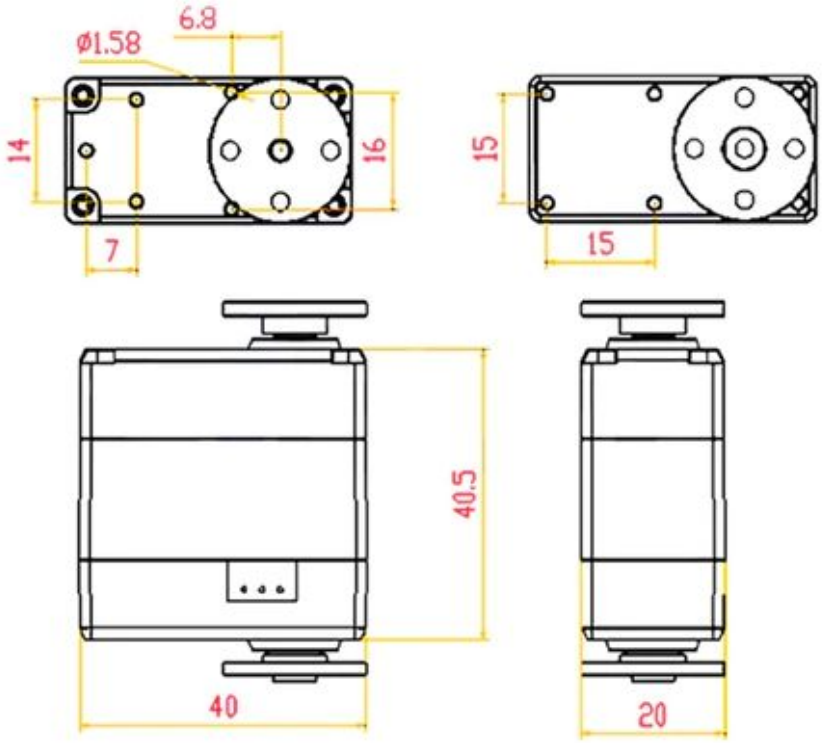
- SB-SS023 - It has a max torque of 2.3 Kg.cm and is helpful in lightweight robotic projects. This servo has dimensions of 23.2 x 12 x 25.5 mm.

- SB-SS15 has a max torque of 15 Kg.cm with dimensions 40 x 20 x 40.5 mm is for heavier applications

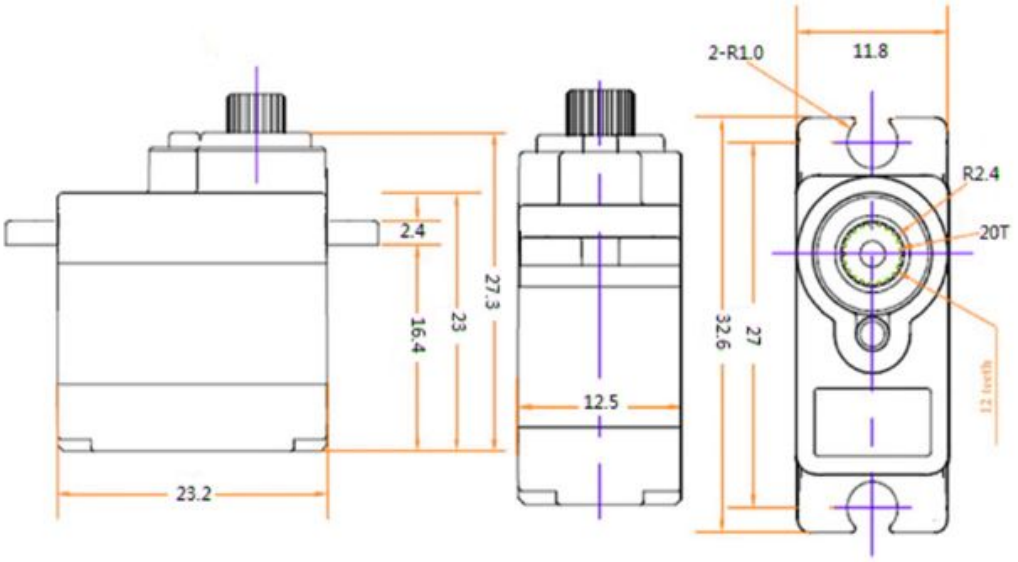
Features and Benefits

Dimensions	23.2X12.0X25.5mm	40.0X20.0X40.5mm
Weight	12.5g(0.44oz)	56g(1.98oz)
Gear Type (Ratio)	Metal(217:1)	Metal(275:1)
No of I/O Ports	0	2
Bus Interface	TTL Level	TT Level
Position Sensor	Potentiometer(300°/1024)	Potentiometer(300°/1024)
Operating Angle	300°(Servo Mode)	300°(Servo Mode)
Communication System	UART Communication	UART Communication
Protocol Type	Half duplex UART	Half duplex UART
ID	254 ID (0~253)	254 ID (0~253)
Communication Speed	38400bps - 1 Mbps	38400bps - 1 Mbps
Feedback	Position, Temperature, Load, Speed, Input Voltage	Position, Temperature, Load, Speed, Input Voltage
Bearing Type	2BB	2BB
Output Shaft	20T(4.88 mm)	25T
Operating Voltage	6V~8.4V	6V~8.4V
Torque(Kg.cm)	0.76(6V)	15(6V)
Stall Current(A)	1A(6V)	1.5A(at 6V)
Operating Speed(RPM)	143(6V)	63(6V)
Connector(Wire Length)	3P&5264 (15cm)	3P&5264 (15cm)

Dimensions



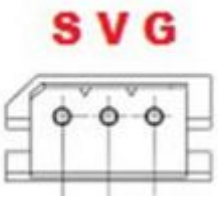
SB-SS 15



SB-SS 023

Serial Communication Protocol

The SB Servos works on a proprietary serial communication protocol. The communication mode is the asynchronous serial mode. The Servo works on the TTL logic level and connection requires three lines; positive (+), negative(-) and the signal line.



Multiple servos can be connected to the control network and each servo is having a unique ID on the network for communication. The Master/Controller sends the packets and only the servo matching the ID can receive the packet and send the response accordingly.

Instruction Frame Format

The master or controller will send the instructions to the servo motors with the below frame format:

Initial	ID	Data Length	Instruction	Parameter	CheckSum
0xFF 0xFF	Servo ID	Length	Instruction Code	Param 1 + ... Param N	CheckSum

Start Byte: Start Byte of any packet consists of 2 bytes of 0xFF
ID Number: Each servo has an ID number and it ranges from 0 to 253 (In Hex - 0x00 to 0xFD). ID number 254 (i.e. 0xFE) is a broadcast ID. If the controller sends the broadcast ID in the frame then all the servos will receive the instruction and no response is returned from servo motors except the PING instructions.
Data Length: Data Length is equal to the number of parameters added with 2 i.e. N + 2.
Instruction Code: This is the operation instructions code, Check the instruction code table below.

Parameters: This is the additional information required by the instruction and it supports the maximum two-byte parameter to represent a memory value.
Checksum: Checksum is used to verify the transmitted and received data. The checksum calculation method for SB-SS servos is-

Checksum = ~ (ID + Length + Instruction Code + Parameter1 + ... Parameter N)

If the sum value in parentheses exceeds 255, the lowest byte will be taken, and "~" means reverse the bits.

Response Frame Format

The servo motor will send the response frame for all the instructions if the response bit in the memory table is not reset. If the Response Bit is set then the SB-SS servo will only respond to read Instruction. The response frame format is:

Initial	ID	Data Length	Current State	Parameter	CheckSum
0xFF 0xFF	Servo ID	Length	Error	Param 1 + ... + Param N	CheckSum

- The response frame contains the ERROR status of the servo. If the ERROR value is 0 this means there is no ERROR in the response.
- During the read instruction (READ DATA), parameters 1 up to N Parameters holds the read information.

Instruction Codes

The instruction codes are used to specify the request type to the SB-SS servo motor. The request type can be ping, read, write, reg write, action, synchronize write, and reset. Majorly instruction codes will be used for reading and writing some data from and to the memory control table.
The following instruction codes are available for SB Serial Servo Motors.

Instruction Name	Function Description	Value	Parameter Length
PING	Check the current status	0x01	0
READ	Read the parameters from the memory table	0x02	2
WRITE	Write the parameters to the memory table	0x03	>= 1
REG WRITE	The parameters are effective only after ACTION instruction code is received	0x04	Not Less than 2
ACTION	It triggers the REG WRITE INSTRUCTION	0x05	0
SYNC WRITE	For simultaneous controlling of multiple servos	0x83	Not Less than 2
RESET	Reset Memory table to factory settings	0x06	0

Table: Instruction Code Table

PING Instruction

Function Read the working state of the servo
Length 0x02
Instruction 0x01
Parameter None

The PING command uses the broadcast address(254) or a single ID to check the status of the motor.

Example: Read the working status of servo motor with ID 1
Instruction Frame:- FF FF 01 02 01 FB

Initial	ID	Data Length	Instruction	CheckSum
0xFF 0xFF	0x01	0x02	0X01	0xFB

Response: The response frame consists of an additional parameter, i.e Current State. If this bit is not 0 look at the Error Table.
Response Frame: FF FF 01 02 00 FC

Initial	ID	Data Length	Current State	CheckSum
0xFF 0xFF	0x01	0x02	0X00	0xFC

READ Instruction

Function Reads data from the servo memory control table
Length 0X04
Instruction 0X02
Parameter 1 Head address of read-out segment of data
Parameter 2 Length of data bytes to read from memory control table

Example: Read the current position of the servo with ID 1 (low byte before, high byte after).
Two bytes are read from address 0X38 in the control table.
Instruction frame: FF FF 01 04 02 38 02 BE initial ID

Initial	ID	Data Length	Instruction	Parameter	CheckSum
0xFF 0xFF	0x01	0x04	0X02	0x38 0x02	0xBE

Response: The response frame now consists of the parameter section which has our requested data, i.e. Position.

Response Frame: FF FF 01 04 00 18 05 DD

Initial	ID	Data Length	Current State	Parameter	CheckSum
0xFF 0xFF	0x01	0x04	0X00	0x18 0x05	0xDD

The parameter has 2-byte data:

Low Byte L - 0X18

High Byte H - 0X05

Two-byte synthesis of 16-bit data 0X0518, using the decimal representation of the current location of 1304.

WRITE Instruction

Function Write data to the servo memory control table
Length N+3 (N is the parameter length)
Instruction 0X03
Parameter 1 Head address of data write segment
Parameter 2 The first data written
Parameter 3 Second data
Parameter N+1 Number N Data

Example: Set the IDs of all the servo motors to 1.
The address of ID number is 5 in the control table, so write 1 at address 5. The ID of the sending instruction package uses the broadcast ID (0xFE).

Instruction frame: FF FF FE 04 03 05 01 F4

Initial	ID	Data Length	Instruction	Parameter	CheckSum
0xFF 0xFF	0xFE	0x04	0X03	0x05 0x01	0xF4

Because broadcasting ID is used to send instructions, there will be no data return.
In addition, the memory table EPROM has a protective lock switch, which needs to be turned off before modifying the ID, otherwise, the written ID number will be erased after the power is off.

Example: Control the ID1 servo to rotate to 2048 at a speed of 1000 seconds. In the control table, the first address of the target location is 0X2A, so six consecutive bytes of

data are written at the address 0X2A, namely position data 0X0800 (2048), time data 0X0000 (0), speed data 0X03E8 (1000). The ID of the sending instruction package uses a non-broadcast ID (0xFE), so the servo will return to the status package when the instruction is received.

Instruction frame: FF FF 01 09 03 2A 00 08 00 E8 03 D5

Initial	ID	Data Length	Instruction	Parameter	Check Sum
0xFF 0xFF	0x01	0x09	0X03	0x2A 0x00 0x08 0x00 0x00 0xE8 0x03	0xBE

Response: The returned current state is 0, indicating that the servo has received the instructions correctly and correctly and has begun to execute them.
Response Frame: FF FF 01 02 00 FC

Initial	ID	Data Length	Current State	CheckSum
0xFF 0xFF	0x01	0x02	0x00	0xFC

- REG WRITE Instruction

The REG WRITE instruction is similar to the WRITE DATA except that the execution time is different. When the REG WRITE instruction frame is received, the received data is stored in the buffer reserve and the Registered Instruction Register is set at 1. When ACTION instruction is received, the stored instruction is finally executed.

Length N+3	(N is the number of data to be written)
Instruction	0X04
Parameter 1	The header address of the area where the data is to be written
Parameter 2	The first data to be written
Parameter 3	The second data to be written
Parameter N+1	The Nth Data to Write

Example: Control ID1 to ID10 servo to rotate to 2048 position at 1000 per second. The only ID in the following instruction package receives instructions on the bus and returns. Other ID numbers are not returned on the bus.

ID 1 Asynchronous Write Instruction Pack：FF FF 01 09 04 2A 00 08 00 00 E8 03 D4
ID 1 Retun Pack：FF FF 01 02 00 FC
ID 2 Asynchronous Write Instruction Pack：FF FF 02 09 04 2A 00 08 00 00 E8 03 D3
ID 3 Asynchronous Write Instruction Pack：FF FF 03 09 04 2A 00 08 00 00 E8 03 D2
ID 4 Asynchronous Write Instruction Pack：FF FF 04 09 04 2A 00 08 00 00 E8 03 D1
ID 5 Asynchronous Write Instruction Pack：FF FF 05 09 04 2A 00 08 00 00 E8 03 D0
ID 6 Asynchronous Write Instruction Pack：FF FF 06 09 04 2A 00 08 00 00 E8 03 CF
ID 7 Asynchronous Write Instruction Pack：FF FF 07 09 04 2A 00 08 00 00 E8 03 CE
ID 8 Asynchronous Write Instruction Pack：FF FF 08 09 04 2A 00 08 00 00 E8 03 CD
ID 9 Asynchronous Write Instruction Pack：FF FF 09 09 04 2A 00 08 00 00 E8 03 CC
ID10 Asynchronous Write Instruction Pack：FF FF 0A 09 04 2A 00 08 00 00 E8 03 CB

ACTION Instruction

Action Instruction is used to trigger the REG WRITE instruction.

Length	0X02
Instruction	0X05
Parameter	None

ACTION instructions are useful for controlling multiple servos at the same time. When controlling multiple servos, the ACTION command enables the first and last servos to perform their respective actions simultaneously without delay. When the action command is sent to multiple servos, the broadcast ID (0xFE) is used, so no data frame will be returned when the command is sent. Example 6: After issuing the asynchronous writing instructions that control ID1 to ID10 servo to rotate at 2048 position at a speed of 1000 seconds, the following instruction packages (FF FF FE 02 05 FA) need to be sent when the asynchronous writing instructions need to be executed. All servos on the bus receive this instruction and run the asynchronous writing instruction received before.

SYNC WRITE Instruction

Used to control multiple servos.

ID	0XFE
Length	(L + 1) * N + 4 (L: Length of data, N: Servo Number)
Instruction	0X83
Parameter 1	Head address of write data
Parameter 2	Length of write data(L)
Parameter 3	First servo Number
Parameter 4	Write the first data of the first servo
Parameter 5	Write the L data of the first servo
Parameter L+3	Write the second data of the first servo
Parameter L+4	The second Servo ID number

Parameter L+5 Write the first data of the second servo
Parameter L+6 Write the second data of the second servo
...
Parameter 2L+4 Write the L data of the second servo ...
...

Unlike the REG WRITE + ACTION instruction, the real-time performance is higher. A SYNC WRITE instruction can modify the control table contents of multiple servos at one time, while the REG WRITE + ACTION instruction can be implemented step by step. Nevertheless, when using SYNC WRITE instructions, the length of the data written must be the same as the first address of the data saved.

Example: Writing position 0X0800 time 0X0000 and speed 0X03E8 for ID1-ID4 with four servo header addresses 0X2A (low byte in front, high node in back)
Instruction frame: FF FF FE 20 83 2A 06 01 00 08 00 00 E8 03 02 00 08 00 00 E8 03 03 00 08 00 00 E8 03 04 00 08 00 00 E8 03 58

Initial	ID	Data Length	Instruction	Parameter	Check Sum
0xFF 0xFF	0xFE	0x20	0X83	0x2A 0x06 0x01 0x00 0x08 0x00 0x00 0xE8 0x03 0x02 0x00 0x08 0x00 0x00 0xE8 0x03 0x03 0x00 0x08 0x00 0x00 0xE8 0x03 0x04 0x00 0x08 0x00 0x00 0xE8 0x03	0x58

As the broadcasting ID is used to send instructions, no data is returned.

- RESET Instruction

Reset the data in the memory control table to factory defaults.
Length 0X02
Instruction 0X06
Parameter None

Example: Reset servo ID number is 0.
Instruction frame: FF FF 01 02 06 F6

Initial	ID	Data Length	Instruction	Check Sum
0xFF 0xFF	0x00	0x02	0x06	0xF7

Response Frame: FF FF 01 02 00 FC

Initial	ID	Data Length	Current State	Check Sum
0xFF 0xFF	0x01	0x02	0x00	0xFC

Memory Mapping

Error State

All the packets send back an acknowledgement packet which contains a *"Current State"* byte. This byte is used to represent the working state of the motor. If the servo is working in the normal state a 0x00 packet is received. If the current working state is abnormal, i.e. there is an issue with some functionality, it will be reflected by the current state byte. Below is the error state table that represents the information on the current working state of the servo.

Data	Name	Description
0x00	Normal	Servo is working in a normal state
0x01	Overvoltage and Undervoltage	The voltage is over the specified range
0x04	Overheated	The temperature is over the specified range
0x20	Overload	In the position mode, the output torque is less than the load when running

Table: Instruction Code Table

Memory Control Table

The SB Servo has a controller IC which is accessed through the Serial UART interface. When we are performing a read or write operation in the servo motor we are actually reading from or writing at the address at RAM or EEPROM.
As mentioned in the Instruction Code section's various Instructions, we need to specify the memory address in the parameters. The Control Parameter and its respective address is mentioned in the below table:

Address	Control Parameter	Access	Initial Value	Location
0 (0x00)	---	---	---	EEPROM
1 (0x00)	---	---	---	
2 (0x00)	---	---	---	
3 (0x03)	Software Version (H)	Read	---	
4 (0x04)	Software Version (L)	Read	---	
5 (0x05)	Servo ID	Read / Write	00 (0x00)	
6 (0x06)	Baud Rate	Read / Write	00 (0x00)	
7 (0x07)	Return Delay Time	Read / Write	00 (0x00)	
8 (0x08)	Answer Status Level	Read / Write	01 (0x01)	
9 (0x09)	Minimum Angle Limit (H)	Read / Write	00 (0x00)	
10 (0x0A)	Minimum Angle Limit (L)	Read / Write	00 (0x00)	
11 (0x0B)	Maximum Angle Limit (H)	Read / Write	03 (0x03)	
12 (0x0C)	Maximum Angle Limit (L)	Read / Write	255 (0xFF)	
13 (0x0D)	Maximum Temperature Limit	Read / Write	80 (0x50)	
14 (0x0E)	Highest Input Voltage	Read / Write	250 (0xFA)	
15 (0x0F)	Lowest Input Voltage	Read / Write	50 (0x32)	
16 (0x10)	Max Torque (H)	Read / Write	03 (0x03)	
17 (0x11)	Max Torque (L)	Read / Write	255 (0xFF)	

18 (0x12)	High Voltage Flag bit	Read / Write	00 (0x00)	
19 (0x13)	Unload Condition	Read / Write	37 (0x25)	
20 (0x14)	LED Alarm Condition	Read / Write	37 (0x25)	
21 (0x15)	P	Read / Write	15 (0x0F)	
22 (0x16)	D	Read / Write	00 (0x00)	
23 (0x17)	I	Read / Write	00 (0x00)	
24 (0x18)	Minimum PWM (H)	Read / Write	00 (0x00)	
25 (0x19)	Minimum PWM (L)	Read / Write	00 (0x00)	
26 (0x1A)	Clockwise Insensitive Region	Read / Write	02 (0x02)	
27 (0x1B)	Counterclockwise Insensitive Region	Read / Write	02 (0x02)	
28 (0x1C)	Integral Restriction (H)	Read / Write	00 (0x00)	
29 (0x1D)	Integral Restriction (L)	Read / Write	00 (0x00)	
30 (0x1E)	Differential Sampling Coefficient	Read / Write	00 (0x00)	
31 - 39	---	---	---	
40 (0x28)	Torque Switch	Read / Write	00 (0x00)	RAM
41 (0x29)	---	---	---	
42 (0x2A)	Target Position (H)	Read / Write	---	
43 (0x2B)	Target Position (L)	Read / Write	---	
44 (0x2C)	Running Time (H)	Read / Write	00 (0x00)	
45 (0x2D)	Running Time (L)	Read / Write	00 (0x00)	
46 (0x2E)	Running Speed (H)	Read / Write	00 (0x00)	
47 (0x2F)	Running Speed (L)	Read / Write		
48 (0x30)	Lock Sign	Read / Write		
49 - 55	---	---	---	
56 (0x38)	Current Position (H)	Read	?	

57 (0x39)	Current Position (L)	Read	?	
58 (0x3A)	Current Speed (H)	Read	?	
59 (0x3B)	Current Speed (L)	Read	?	
60 (0x3C)	Current Load (H)	Read	?	
61 (0x3D)	Current Load (L)	Read	00 (0x00)	
62 (0x3E)	Current Voltage	Read	?	
63 (0x3F)	Current Temperature	Read	?	
64 (0x40)	REG WRITE Sign	Read	00 (0x00)	

Table: Instruction Code Table

In the instruction code table, if the control parameter has the high and low byte: "L" and "H", then its range is 0x00 - 0x3FF (0 - 1023). Example: the target position at 42 (0x2A) holds the higher byte and 43 (0x2B) holds the lower byte of the parameter.
If the control parameter takes only one byte then its range is 0x00 - 0xFE (0 - 255). The parameters stored in RAM will not be saved after power down. The parameters stored in the EEPROM can be saved after power down. "---" indicates that parameters are not accessible to the user and hence cannot be modified.

- Baud Rate (0x06)

  The Baud Rate of the servo motor can be changed by writing into the baud rate parameter in the memory control table. Baud Rate at 0x06 defaults to 0 which is 1M. The baud rate and the corresponding parameters are listed below:

Address	Hex	Actual Baud Rate	Target Baud Rate	Error (%)
0	0x00	1000000.0	1000000.0	0.00
1	0x01	500000.0	500000.0	0.00
2	0x02	250000.0	250000.0	0.00
3	0x03	128000.0	128000.0	0.00
4	0x04	115107.9	115200	0.079
5	0x05	76923.0	76800	-0.16
6	0x06	57553.9	57600	0.008

7	0x07	38461.5	38400	-0.16

- Return Delay Time (0x07)

  Sets the response time when the user reads or writes a parameter. When the command is received by the servo this parameter sets the delay to respond back to the master. It is a one-byte parameter (0~255). The time is calculated by multiplying the parameter value by 2 microseconds, If the parameter is 250, the servo will respond after 500 us. The default value of this parameter is 0, which means servo will respond as soon as the instruction is received.

- Answer status level (0x08)

  Answer status level can be used to stop the servo from responding to all the instructions. The default value of this parameter is 1.

Address	Return Response Packet
0	Other instructions do not return the acknowledge packet except the read instruction and the PING instruction
1	Returns the response packet to all instructions

- Angle Limit (0x09～0x0C)

  This parameter sets the angle range that the actuator can rotate.
  Minimum angle limit ≤ *Target angle value* ≤ *Maximum angle limit*
  Note that the minimum angle limit must be less than the maximum angle limit. If the target angle exceeds the range, it is equal to the limit value.
  The default for the minimum angle is 0x00 and for maximum angle 0x3FF.

- Maximum Temperature (0x0D)

  It is the maximum operating temperature. The default is ox50 i.e. 80 degree. The highest temperature is 80 degrees, and the setting accuracy is 1 degree.

- Highest Input Voltage (0x0E)

  It is the highest operating voltage of the servo motor. The dafault is 0xFA i.e. 85 which is equivalent to 8.5 achieved by multiplying the value with 0.1. The accuracy is 0.1V

- Lowest Input Voltage (0x0F)

  It is the lowest operating voltage of the servo motor. The default is 0x32 i.e. 50 equivalent to 5 volts.The accuracy is of 0.1V

- Max Torque (0x10～0x11)

  This parameter set the maximum output torque of the SB servo. It is a two-byte parameter which defaults to 0x03FF i.e. 1023. It is the maximum operating torque.

- LED Alarm Condition (0x13)

  Set uninstall condition, /LED alarm condition.

Data	Function
0x20	If set to 1, the overload output /LED alarm occurs when overload occurs
0x04	If set to 1, release the torque /LED alarm when overheating occurs
0x00	If set to 1, the torque /LED is released when the voltage range exceeds the voltage range

  To act in accordance with logic or principles. LED alarm condition (0X14) set to 0, turn off LED, otherwise open LED。

- Insensitive Range(0x1A～0x1B)

  The size of the dead zone position loop, clockwise and counterclockwise are set to 1 big dead zone is about 0.

- Torque Switch (0x28)

  Torque switch is used to enable or disable the torque of the servo motor. If the Torque Switch is enabled, the servo shaft will be locked.
  If the torque switch is on at “1” and off at “0”.

- Target Position (0x2A～0x2B)

  It is the target position of the servo shaft. This parameter is used to rotate the servo motor in the servo mode.
  The range of target position is 0x0000 - 0x03FF, 0x0000 corresponds to 0 degrees, 0x03FF corresponds to 200 degrees with a deviation of +2%.

- Running Time and Running Speed (0x2C～0x2F)

  These 2 parameters are used to set time and speed during servo rotation to the target location (velocity parameters than the time parameters of priority, and the writing time and speed parameters, velocity parameters are selected as control parameters).
  The time parameter unit is millisecond, the velocity parameters unit is 0.19 / sec as the 1000 speed for (1000*0.19) deg/sec. If Set to 0, the corresponding to the maximum speed is 62RPM.

- Lock Sign(0x30)

  The Lock Sign bit is used to write into the EEPROM. As there are limited write cycles for the EEPROM, this bit is used to control the EEPROM write.
  If the Lock Sign bit is set to 0, the EEPROM lock protection is locked, and the EEPROM parameter can be saved permanently.
  If the Lock Sign is 1, the EEPROM lock protection is unlocked. If the lock sign function byte is set to 1, the SB servo write speed turns slow, frequent write operations on the EEPROM parameter affect the life of the servo.

- REG WRITE sign (0x40)

  If the REG WRITE command waits for execution, it is shown as 1, and when the REG WRITE instruction is executed, it is shown as 0. The default is 0x00.

Motor Mode

SB Serial servo has a dual operating mode: the servo mode and the motor mode. It can be used for the 360-degree rotation application such as in a robotic car.

To operate the SB Serial Servo in the motor mode the minimum and the maximum angle (0x09 ~ 0x0C) should be set to 0.

The speed is controlled by writing the speed and direction into the “Running Time” control parameter of the memory control table. The size and direction are written into the 0x2C～0x2D address in the following format:

	Running Time (0x2C～0x2D)											
Bit	15~11	10	9	8	7	6	5	4	3	2	1	0
Value	0	0/1	Speed Value									

BIT 10 is the direction bit if set to ‘0’ SB Servo will rotate anticlockwise, and setting it to ‘1’ will rotate the servo clockwise rotation.

The speed of the servo is set by setting BIT0～BIT9.