Designer Systems.co.uk
PRODUCT DESIGN AND MANUFACTURING

18 Channel Servo Driver Shield

for Arduino™ and Raspberry-PI™

Technical Data

DS-SCX18.S

Features

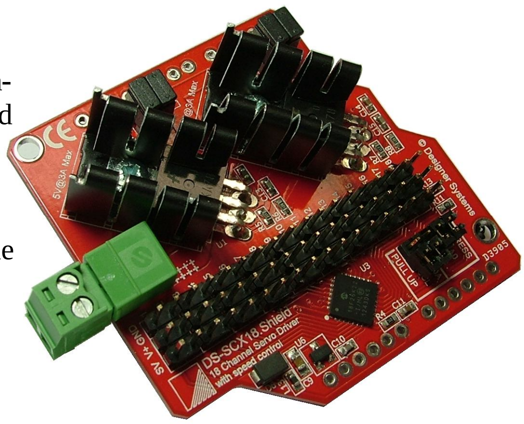
- **Arduino™ UNO Shield standard form factor for simple integration into any Arduino project**
- **Frees up the Arduino™ IO lines normally used for servo control**
- **I^{2}C interface for simple connection to Arduino or Raspberry-PI**
- **Dual on-board 5Volt 3Amp regulators, with heatsinking, for servo power with enable jumpers**
- **Sixteen (16) level speed control for each servo**
- **Movement complete and soft-start complete status for each servo**
- **Global activation control ensures all servos start moving together, important for multi-legged robots**
- **I^{2}C address links allow up to four [4] shields to be used together to provide up to 72 servos**
- **LVD, RoHS and WEEE compliant product**

Description

The Designer Systems DS-SCX18.S is an eighteen [18] channel RC servo driver with advanced servo control features.
Specifically targeted at the Arduino UNO board user [all other Arduino boards supported] and the Raspberry-PI the SCX18.S features high speed I^{2}C communication for easy project integration and smooth speed control.

Once connected each servo can be positioned (with speed control), enabled, reversed and soft-started by simply writing a value to an internal register over the connected I^{2}C interface.
The SCX18.S caters for the majority of servos by providing a wide pulse width range of 0.50mS to 2.50mS with 8uS per step accuracy and also provides global activation of new servo position, soft-start & movement complete registers for superior control.

The SCX18.S provides dual high power regulators capable of supplying 5V @ 3A to the connected servos from an external Li-Pol or



Ni-MH battery pack of 7.2 to 8.4V and features a disable link to allow the use of low voltage battery packs of 3.6 to 4.8V.
The on-board I^{2}C pull-ups are jumper configurable to allow disconnection when connecting to the Raspberry-PI, which has its own pull-ups.

Applications

The SCX18.S has applications in robotics, including quadruped, hexapod and octopod robots, process control & sensor manipulation when used in conjunction with standard RC servos.

CONTROL MODULES

Selection Guide

Description	Part Number
18 Channel Servo Driver Shield	DS-SCX18.S
HS311 Standard hobby servo	DS-HS311

Raspberry-PI, Arduino, NANO, UNO & MEGA are trademark.

Power requirements

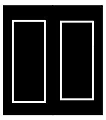
The DS-SCX18.S takes the power necessary for operation (approx. 2-25mA) from an external battery or power adaptor or power from the Arduino or Raspberry-PI board.
The SCX18 provides three PCB pads, two marked ‘GND’ and one marked ‘Vin’ in the same format as that present on the UNO board, which should be connected to negative and positive battery/power supply terminals respectively. The input voltage range is 4.75 - 16VDC with the internal circuitry being protected against power supply reversal.

The two on-board servo regulators provide a clean regulated 5V supply for the connected servos at a maximum current of 3Amps each, the left hand regulator powering servos 1 to 9 and the right hand regulator servos 10-18. The following table indicates if a regulator needs to be enabled or disabled dependant on supply voltage:

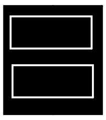
Status	Battery pack/supply voltage
Disabled	3.6V to 4.8V
Enabled	7.2V to 8.5V

The servo regulators are enabled or disabled with on-board double links just above the regulator heat-sinks which can be configured as follows:

Enabled (ON):



Disabled (OFF):



Refer to the voltage rating of the servos you wish to use before configuring the servo regulator.

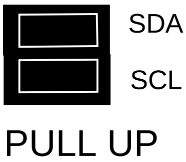
Connection of the external supply - battery / mains adaptor - to the SCX18 module is through a two (2) way pluggable screw terminal block marked ‘3-8.5VDC @ 6A Max’.

Note: This supply is NOT reverse connection protected but is marked with a series of ‘+++++’ signs to denote positive.

I²C connection

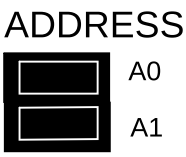
The I²C connections are marked ‘SDA’ and ‘SCL’ and allow connection to the Arduino UNO board ‘ANALOG IN’ pins 4 and 5 or the Rasperberry-PI GPIO port pins 3 and 5 (see Fig. 2.0) or another I²C Master device.

The DS-SCX18.S is fitted with pull-up jumpers that can be configured to provide the source current necessary for I²C communication. The following jumpers should normally be set when using the UNO board, as long as the I²C bus does not have existing pull-up’s provided by another device. **These jumpers <u>MUST</u> be removed when using the Raspberry-PI:**



I²C communication

Up to four DS-SCX18.S modules may be connected to the same UNO / Raspberry-PI board or I²C bus and accessed individually using their own individual address.
The address is configured with the following jumpers:



The following table shows how the jumpers are placed for the different binary addresses:

Address xx	A0	A1
00 (default)	ON	ON
01	OFF	ON
10	ON	OFF
11	OFF	OFF

The binary address (xx) above is used in conjunction with the device ID 11101xxD to form the complete device address i.e. if both jumpers are left connected (default) then the device address would be $1110100D_{binary}$.
The ‘D’ bit determines if a read or a write to the SCX18 is to be performed. If the ‘D’ bit is set ‘1’ then a register read is performed or if clear ‘0’ a register write.

To access individual registers a device write must be undertaken by the I²C Master which consists of a Start condition, device ID (‘D’ bit cleared), register to start write, one or more bytes of data to be written and a stop condition (see Figure 1.0 for I²C write protocol).

There are 37 individual registers that can be written within the SCX18 that control output as follows:

SCX I2C address

	N_7	N_6	N_5	N_4	N_3	N_2	N_1	N_0
1.	1	1	1	0	1	X	X	0

XX = SCX18 address

Register address

R0	U	U	B	B	B	B	B	B

B..B = 0 to 37
U..U = unused on this implementation

Servo 1 position

R1	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 1 control

R2	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 2 position

R3	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 2 control

R4	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 3 position

R5	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 3 control

R6	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 4 position

R7	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 4 control

R8	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 5 position

R9	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 5 control

R10	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 6 position

R11	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 6 control

R12	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 7 position

R13	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 7 control

R14	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 8 position

R15	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 8 control

R16	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 9 position

R17	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 9 control

R18	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 10 position

R19	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 10 control

R20	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 11 position

R21	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 11 control

R22	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 12 position

R23	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 12 control

R24	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 13 position

R25	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 13 control

R26	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 14 position

R27	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 14 control

R28	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 15 position

R29	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 15 control

R30	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 16 position

R31	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 16 control

R32	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 17 position

R33	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 17 control

R34	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo 18 position

R35	P	P	P	P	P	P	P	P

P..P = Servo position 0 to 255 (0.5mS to 2.50mS)

Servo 18 control

R36	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

Servo global enable register

R37	X	X	X	X	X	X	X	X

X..X = Any value

Each control register consists of four control bits and a four bit speed control value defined as follows:

Bit (A) $128_{decimal}$ is the operate bit which when set activates the servo being controlled.
Bit (B) $64_{decimal}$ is the reverse bit which reverses the position value for the servo being controlled.
Bit (C) $32_{decimal}$ is the soft-start bit which when set on servo first activation, see operate bit above, feeds position pulses to the servo in a ramping manner until position is attained.
Bit (D) $16_{decimal}$ is the speed control enable bit which when set applies the speed value 0 to 15, contained in the four bits (SSSS) $1,2,4,8_{decimal}$, to the servo being controlled.

Once all the required position & control registers have been set a write to the R37 (Global enable register) must be made to activate all the new positions.

Example.

To set the first nine servos to new positions with servos 1 to 4 running at speed 0 and servos 5 to 8 running at speed 5 in reverse mode, first write:

Byte 1 (SCX18 Adr) $1110100\mathbf{0}_{binary}$
Byte 2 (Register 0) $0_{decimal}$
Byte 3 (Register 1) $30_{decimal}$
Byte 4 (Register 2) $144_{decimal}$, 90_{hex}
Byte 5 (Register 3) $35_{decimal}$
Byte 6 (Register 4) $144_{decimal}$, 90_{hex}
Byte 7 (Register 5) $40_{decimal}$
Byte 8 (Register 6) $144_{decimal}$, 90_{hex}
Byte 9 (Register 7) $45_{decimal}$
Byte 10 (Register 8) $144_{decimal}$, 90_{hex}
Byte 11 (Register 9) $127_{decimal}$
Byte 12 (Register 10) $213_{decimal}$, $D5_{hex}$
Byte 13 (Register 11) $130_{decimal}$
Byte 14 (Register 12) $213_{decimal}$, $D5_{hex}$
Byte 15 (Register 13) $140_{decimal}$
Byte 16 (Register 14) $213_{decimal}$, $D5_{hex}$
Byte 17 (Register 15) $150_{decimal}$
Byte 18 (Register 16) $213_{decimal}$, $D5_{hex}$

then to activate write:

Byte 1 (SCX18 Adr) $1110100\mathbf{0}_{binary}$
Byte 2 (Register 0) $37_{decimal}$
Byte 3 (Register 37) $0_{decimal}$

To read the status registers a device write then read must be undertaken by the OOPic / I^2C Master. The write consists of a Start condition, device ID ('D' bit cleared), register to start read and a Stop condition. This is followed by a read, which consists of a Start condition, device ID ('D' bit set), followed by data from the status register and terminated with a Stop condition (see Figure 1.1 for I^2C read protocol).

Status registers

There are 18 registers that can be read within the SCX18 as follows:

SCX I2C Address

	N_7	N_6	N_5	N_4	N_3	N_2	N_1	N_0
1.	1	1	1	0	1	X	X	1

XX = SCX18 address

Servo 1 status

R0	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 2 status

R1	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 3 status

R2	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 4 status

R3	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 5 status

R4	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 6 status

R5	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 7 status

R6	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 8 status

R7	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 9 status

R8	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 10 status

R9	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 11 status

R10	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 12 status

R11	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 13 status

R12	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 14 status

R13	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 15 status

R14	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 16 status

R15	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 17 status

R16	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Servo 18 status

R17	A	B	C	D	0	0	0	0

A = Operation (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start (0 – Complete 1 – In-progress)
D = Movement (0 – Complete 1 – In-progress)

Firmware version

R18	M	M	M	M	V	V	V	V

M..M = Firmware major revision number 1-15
V..V = Firmware minor revision number 1-15

Bit (A) $128_{decimal}$ is the operate bit which when set indicates that the servo is operational.
Bit (B) $64_{decimal}$ is the reverse bit which when set indicates that position values written to the servo will be reversed.

Bit (C) $32_{decimal}$ is the soft-start bit which when set indicates that soft-start is in progress.

Movement complete determination
Bit (D) $16_{decimal}$ is cleared to indicate if the current servo movement has completed.
This indication is not derived from mechanical or electrical feedback from the servo being controlled but is a function of the current servo speed selected and position.
When the slowest servo speed (0) is selected the determination of movement completion is at its best. This is because the positional change of the servo between its current and final position has been split into many sub-positions which must be attained before the final position is reached. These many sub-positions ensure that the mechanical position closely relates to the position requested by the pulse width and therefore the determination of final position (movement complete) will closely relate to mechanical position. As servo speed is increased the error between mechanical position and pulse width position increases and movement completion accuracy is degraded.

Electrical Characteristics (T$_A$ = 25^{o}C Typical)

Parameter	Minimum	Maximum	Units	Notes
Supply Voltage (Servo power)	3.8	12	V	1,2
Supply Current (Servo power)	1	2x 2800	mA	3
Supply Voltage (on-board VCC)	4.75	16	V	
Supply Current (on-board VCC)	2	25	mA	4
I2C pull-up resistance	-	4700	Ω	
I^{2}C speed	-	400	kHz	

Absolute Maximum Ratings

Parameter	Minimum	Maximum	Units	Notes
Supply Voltage (Servo power)	-0.5	+30	V	5
Supply Current (Servo power)	0	4.5	A	

Environmental

Parameter	Minimum	Maximum	Units
Operating Temperature	0	70	^{o}C
Storage Temperature	-10	80	^{o}C
Humidity	0	80	%
Dimensions	Length 56.25mm, Width 53.5mm, Height 15mm		
Weight	25g		
Immunity & emissions	See statement on page 8		

Notes:
1.Servo voltage below 5V requires that the servo regulator be disabled (see above).
2.Voltages above 6-12V may require force cooling of the heat-sinks if servo load is high.
3.Values given are based on maximum and minimum loading for each regulator.
4.Values given are for servos being not driven and driven.
5.Value given is based on maximum and minimum loading.

Calculating binary bit values:

The registers used above use the binary notation to allow the control of servo operation, reversal, soft-start & speed selection. Each register is made up of eight (8) bits, which can be set or cleared to produce the desired operation, the individual bits having a value associated with them as follows:

128	64	32	16	8	4	2	1

If we take for example one of the servo control registers we can see it is made up of four (4) separate bits A, B, C & D plus a four bit value SSSS:

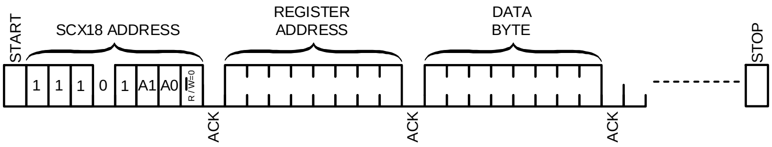
Servo 1 control

R1	A	B	C	D	S	S	S	S

A = Operate (0 – Servo disabled 1 – Servo enabled)
B = Reverse (0 – Servo normal 1 – Servo reversed)
C = Soft-start control (0 – Disabled 1 – Enabled)
D = Speed control (0 – Disabled 1 – Enabled)
S..S = Servo speed value 0 to 15 (0 = slowest)

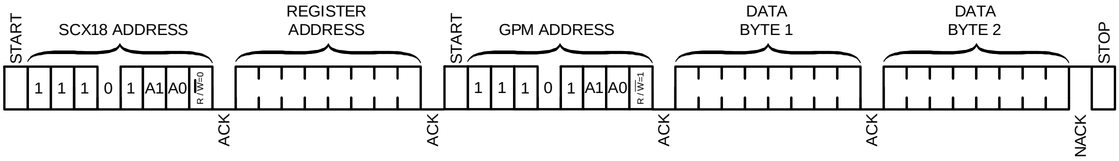
Each bit is defined to control a particular function for the servo it controls, so if for example we wanted to enable servo 1 we would need to set bit ‘A’ which controls the servo operation. We know from the bit values defined above that the value associated with the ‘A’ bit is 128, so by writing this value to register 1 we can enable servo 1. If we need to enable additional functions such as the speed control - ‘D’ - as well as the servo enable, the value of this bit is added to the value written to the register i.e. 128 + 16 = 144. In addition we could also add a speed value of 5 that would make the total value 128 + 16 + 5 = 149.

Figure 1.0 (I^2C write protocol)



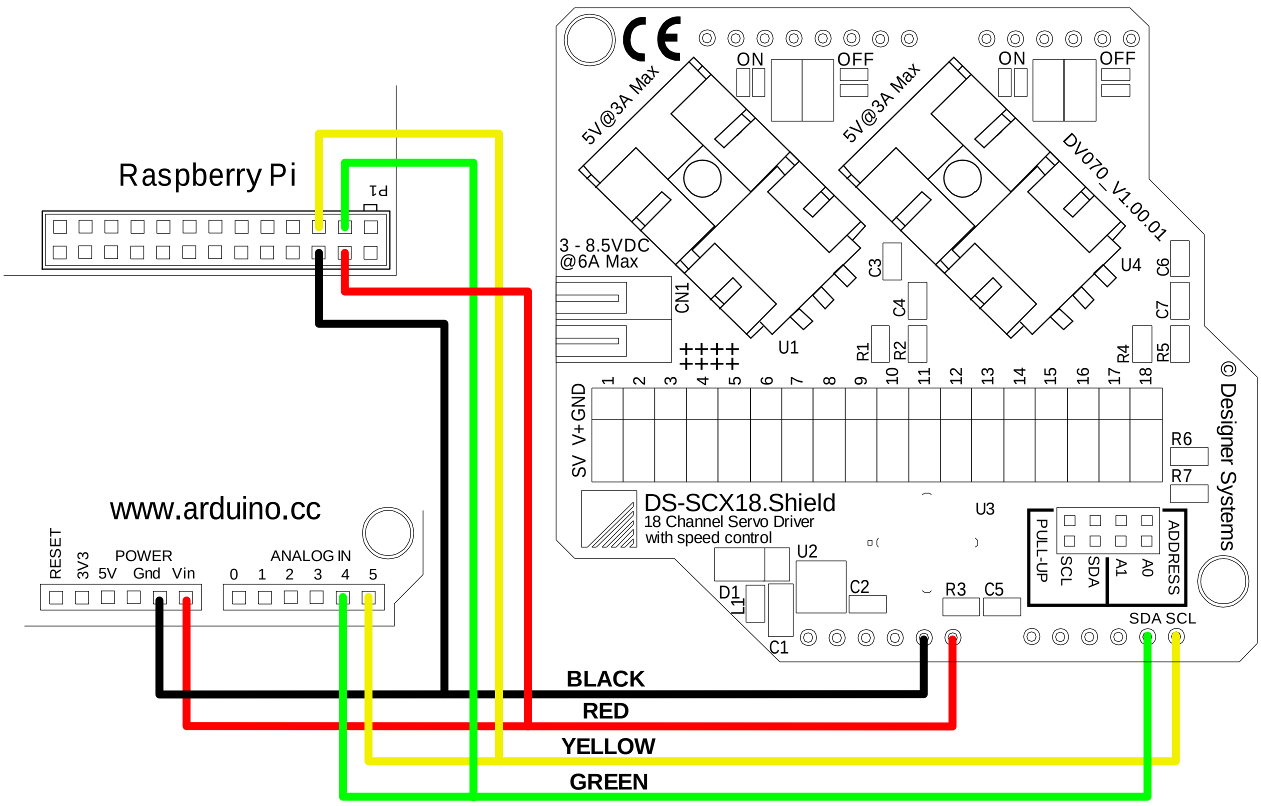
Multiple bytes may be written before the ‘STOP’ condition. Data is written into registers starting at ‘REGISTER ADDRESS’, then ‘REGISTER ADDRESS’ +1, then ‘REGISTER ADDRESS’ +2 etc.
Each byte transfer is acknowledged ‘ACK’ by the SCX18 until the ‘STOP’ condition.

Figure 1.1 (I^2C read protocol)

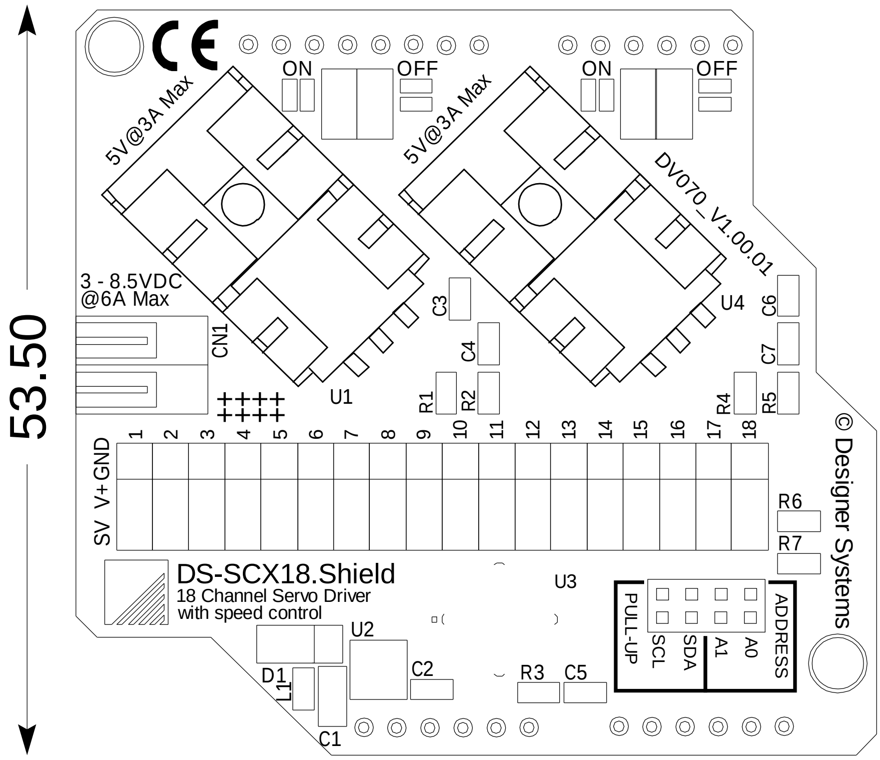


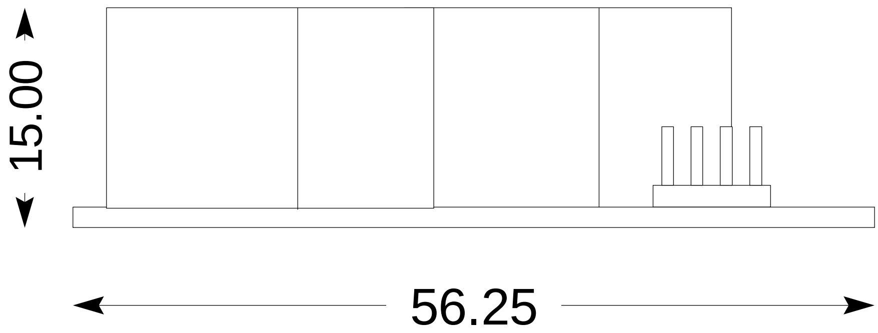
‘DATA BYTE 1 & 2’ are register values returned from the SCX18. Each byte written is acknowledged ‘ACK’ by the SCX18 , every byte read is acknowledged ‘ACK’ by the I^2C Master. A Not-acknowledge ‘NACK’ condition is generated by the I^2C Master when it has finished reading.

Figure 2.0 (Connection Schematic for Arduino UNO or Raspberry-Pi I^2C communication)



Mechanical Specifications – Units millimetres





Revision History:

1.00 Release version

WEEE Consumer Notice

This product is subject to Directive 2002/96/EC of the European Parliament and the Council of the European Union on Waste of Electrical and Electronic Equipment (WEEE) and, in jurisdictions adopting that Directive, is marked as being put on the market after August 13, 2005, and should not be disposed of as unsorted municipal/public waste. Please utilise your local WEEE collection facilities in the disposition and otherwise observe all applicable requirements. For further information on the requirements regarding the disposition of this product in other languages please visit www.designersystems.co.uk

RoHS Compliance

This product complies with Directive 2002/95/EC of the European Parliament and the Council of the European Union on the Restriction of Hazardous Substances (RoHS) which prohibits the use of various heavy metals (lead, mercury, cadmium, and hexavalent chromium), polybrominated biphenyls (PBB) and polybrominated diphenyl ethers (PBDE).

Declaration of Conformity

Apparatus name / model number DS-SCX18.S

Manufacturer Designer Systems, 11 Castle Street, Truro, Cornwall TR1 3AF, United Kingdom

Conformity via Generic Standard EN61000-1
Generic Standard EN61000-3

Description of apparatus Robotic interface peripheral

Conformity criteria For use only within commercial, residential and light industrial applications

We certify that the apparatus identified above conforms to the requirements of Council Directive 2004/108/EC & 2006/95/EC

Signed. **Date** 20/6/13

Having made this declaration the CE mark is affixed to this product, its packaging, manual or warranty.