

PHAT-SV16 TECHNICAL DESCRIPTION

pHAT Module Series

Document: pHAT-SV16 Technical Description

Issue: 1

Date: 18th July 2022



Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Designer Systems Ltd.

11 Castle Street, Truro, Cornwall TR1 3AF, United Kingdom.

Tel: +44 (0) 1872 262000

Email: sales@designersystems.co.uk

For more information, please visit:

<http://www.designersystems.co.uk>

For technical support, or to report documentation errors, please visit:

<http://www.designersystems.co.uk/robotics>

Or email to: support@designersystems.co.uk

GENERAL NOTES

DESIGNER SYSTEMS OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. DESIGNER SYSTEMS MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. DESIGNER SYSTEMS DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF DESIGNER SYSTEMS LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

Copyright © Designer Systems Ltd. 2022. All rights reserved.

About the Document

History

Revision	Description	Date	Author
Draft A	Create	29/4/22	DIO
Issue 1	First release	18/7/22	-

Contents

	Page No.
1 Introduction	6
2 Product Concept	7
2.1 General Description	7
2.2 Key Features	8
3 Application.....	9
3.1 Installation.....	9
3.2 Servo Motor Connection.....	9
3.3 Operation	9
3.4 Indication	9
3.5 Pin Assignment	10
3.5.1 Raspberry-Pi Header	10
3.5.2 Servo Headers.....	10
3.6 Power Supply	11
3.6.1 Servo Power Terminal.....	11
3.6.2 Power Supply Pins.....	11
3.7 I ² C Interface.....	12
3.7.1 I ² C Interface Pins.....	12
3.7.2 I ² C Communication	12
3.7.3 I ² C Registers.....	13
3.7.4 I ² C Servo Control Register Format.....	23
3.7.5 I ² C Servo Status Register Format.....	23
3.7.6 Servo Movement Complete Determination	23
3.7.7 I ² C Global Update Register	24
3.7.8 I ² C Write Example	24
3.8 Application Software	25
3.8.1 Demonstration Software.....	25
4 Electrical Characteristics	26
4.1 Absolute Maximum Ratings	26
4.2 Operating Conditions.....	26
4.3 Current Consumption	27
5 Mechanical.....	28
5.1 Dimensions.....	28
6 References	29
6.1 I ² C protocols	29
7 Appendix.....	30
8 Compliance	31

Tables

Table 1: Key Features	8
-----------------------------	---

Table 2: Status Indication	9
Table 3: Power Supply Pins.....	11
Table 4: I ² C Interface Pins.....	12
Table 5: I ² C Address Settings	12
Table 6: I ² C Registers.....	13
Table 7: I ² C Servo Control Register Format.....	23
Table 8: I ² C Servo Status Register Format.....	23
Table 9: Absolute Maximum Ratings	26
Table 10: Normal Operating Conditions	26
Table 11: Current Consumption	27
Table 12: Related Documents.....	30
Table 13: Terms and Abbreviations	30

Figures

Figure 1: Dimensions.....	28
Figure 2: I ² C Write protocol.....	29
Figure 3: I ² C Read protocol.....	29

1 Introduction

This document defines the pHAT-SV16 Servo driver module and describes the hardware interface that is connected to the customers Raspberry-Pi application.

This document can help customers quickly understand module interface specifications, electrical and mechanical details, as well as other related information of the module. Associated with the quick start guide and demo software, customers can use this document to easily set up the module.

2 Product Concept

2.1 General Description

The Designer Systems pHAT-SV16 is a fully featured sixteen [16] servo motor driver with advanced control features. Specifically designed for the Raspberry-Pi Zero (other Raspberry-Pi boards fully supported) the pHAT-SV16 features high speed I2C communication for easy project integration.

pHAT-SV16 supports the majority of analogue 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.

pHAT-SV16 also provides address selection jumpers to allow up to four [4] boards to be connected to a single Raspberry-Pi board allowing control of up to sixty-four [64] servos.

pHAT-SV16 has applications in robotics, including quadruped, hexapod and octopod robots, process control & sensor manipulation when used in conjunction with standard analogue RC servos.

Due to compact form factor, ultra-low power consumption and extended temperature range, pHAT-SV16 is a best choice for a wide range of robotic applications.

The module fully complies with the RoHS directive of the European Union

2.2 Key Features

The following table describes the key features of the pHAT-SV16.

Table 1: Key Features

Features	Details
Power Supply	- Supply Voltage: 4.5 ~ 5.5VDC - Typical Supply Voltage: 5.0VDC
Servo motor	16 x Analogue (standard)
Servo Voltage	5.0 ~ 7.4V
Servo Power	Limited by external supply (recommend 3A minimum)
Servo Control	- Enable/Disable/Reverse - Soft Start - Position with speed control - Position complete flag
Servo Connection	16 x 3pin header (standard)
I ² C Speed	400kHz max.
Environmental	- Operating Temperature -20°C to 50°C - Storage Temperature -45°C to 100°C
Dimensions	75 x 30 x 10mm
Weight	14g approx.

3 Application

3.1 Installation

The module should be attached to the Raspberry-Pi board using a 20+20 2.54mmP pin header/socket (not supplied).

3.2 Servo Motor Connection

The servo motor connections are marked from '1' to '16' and consist of three pins marked 'GND', 'V+' and 'SV'. Analogue servos should be connected to these pins, noting polarity, BLACK is normally GND, see section 2.7.2 for pinning.

3.3 Operation

When power is applied to the pHAT-SV16, from the connected Raspberry-Pi board, the module will power-up and is then ready for operation.

3.4 Indication

The STATUS indicator is used to provide visual feedback of the current SV16 condition. There are two (2) conditions as follows.

Table 2: Status Indication

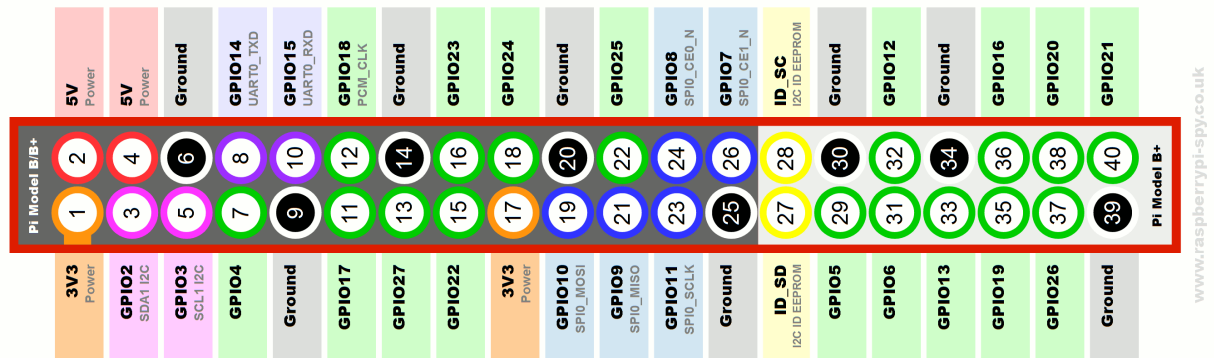
Indication	Description
ON	Power applied and ready
Flashing fast	New servo position requested and in-progress

These conditions will change as new servo positions are requested.

3.5 Pin Assignment

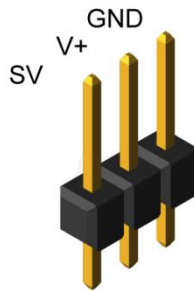
3.5.1 Raspberry-Pi Header

The SV16 module connects to the Raspberry-Pi 40pin header, pinout as follows:



3.5.2 Servo Headers

The 16 x Servo motor headers are 3way 2.54mm pitch, pinout as follows:



3.6 Power Supply

3.6.1 Servo Power Terminal

The SV16 provides a 3.5mm pitch pluggable servo power terminal that should be connected to an external 3 ~ 8.5V DC supply of at least 2Amps. If all 16 servos are connected and are in a hold condition (physical load applied) the current at this terminal may rise to > 6Amps.



WARNING: This power input is NOT reverse polarity protected. Servo damage may result if the supply is incorrectly connected.

3.6.2 Power Supply Pins

The pHAT-SV16 provides a supply input and multiple ground connections on the 20+20 header that connect to the 5.0V supply on the Raspberry-Pi board. The table below describes the module supply and ground pins.

Table 3: Power Supply Pins

Pin Name	Pin No	Description	Min	Typ.	Max	Unit
V+	2, 4	Power Supply	4.5	5.0	6.0	V
Ground	6,9,14,20, 25,30,34, 39	Power Ground				

3.7 I²C Interface

3.7.1 I²C Interface Pins

The SV16 provides I²C data (SDA) and clock (SCL) connections on the 20+20 header that connect to the SDA and SCL on the Raspberry-Pi board. The table below describes the module I²C pins.

Table 4: I²C Interface Pins

Pin Name	Pin No	I/O	Description	Comment
SDA	3	DIO	I2C Data	3.3V level
SCL	5	CO	I2C Clock	3.3V level

The SV16 does NOT have I²C pullups but relies on the pullups present on the Raspberry-Pi board. When not connecting to a Raspberry-Pi board external pullups of 4.7Kohms should be connected on SDA and SCL to a 3.3V supply.

3.7.2 I²C Communication

Up to four SV16 modules may be connected to the same Raspberry-Pi board or I²C bus and accessed individually using their own individual address.

The following table shows how the pads are soldered for the different binary addresses.

Table 5: I²C Address Settings

Address (xx)	A0	A1
00	OPEN	OPEN
01	SHORT	OPEN
10	OPEN	SHORT
11	SHORT	SHORT

The binary address (xx) above is used in conjunction with the device ID 11001xxD (0xB8_{hex}) to form the complete device address i.e. if both jumpers are left unconnected (default) then the device address would be 1100100D_{binary}.

The 'D' bit determines if a read or a write to the SV16 is to be performed. If the 'D' bit is set '1' then a register read is performed or if clear '0' a register write.

3.7.3 I²C Registers

To write individual registers a device write must be undertaken by the Raspberry-Pi 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 2 for I²C write protocol).

To read individual data and status registers a device write then read must be undertaken by the Raspberry-Pi which consists of a Start condition, device ID ('D' bit clear), 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 register specified and terminated with a Stop condition. The RV also auto increments the register specified for every additional read requested by the Master I²C device, which allows more than one register to be read in one transaction. This allows for example all the servo status registers to be read in one transaction (see Figure 3 for I²C read protocol).

There are 33 individual registers that can be written and 17 registers that can be read within the SV16 as follows:

Table 6: I²C Registers

Register name	Type	Register address		Description								
		Hex	Binary									
Register address	w	00	00000000	Start register to write to								
Servo 1 position	w	01	00000001	Servo position (8uS resolution)								
Servo 1 control	w	02	00000010	Servo 1 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 2 position	w	03	00000011	Servo position (8uS resolution)								

Register name	Type	Register address		Description								
		Hex	Binary									
Servo 2 control	w	04	00000100	Servo 2 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 3 position	w	05	00000101	Servo position (8uS resolution)								
Servo 3 control	w	06	00000110	Servo 3 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 4 position	w	07	00000111	Servo position (8uS resolution)								
Servo 4 control	w	08	00001000	Servo 4 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 5 position	w	09	00001001	Servo position (8uS resolution)								

Register name	Type	Register address		Description								
		Hex	Binary									
Servo 5 control	w	0A	00001010	Servo 5 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 6 position	w	0B	00001011	Servo position (8uS resolution)								
Servo 6 control	w	0C	00001100	Servo 6 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 7 position	w	0D	00001101	Servo position (8uS resolution)								
Servo 7 control	w	0E	00001110	Servo 7 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 8 position	w	0F	00001111	Servo position (8uS resolution)								

Register name	Type	Register address		Description								
		Hex	Binary									
Servo 8 control	w	10	00001000	Servo 8 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 9 position	w	11	00000111	Servo position (8uS resolution)								
Servo 9 control	w	12	00010010	Servo 9 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 10 position	w	13	00010011	Servo position (8uS resolution)								
Servo 10 control	w	14	00010100	Servo 10 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 11 position	w	15	00010101	Servo position (8uS resolution)								

Register name	Type	Register address		Description								
		Hex	Binary									
Servo 11 control	w	16	00010110	Servo 11 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 12 position	w	17	00010111	Servo position (8uS resolution)								
Servo 12 control	w	18	00011000	Servo 12 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 13 position	w	19	00011001	Servo position (8uS resolution)								
Servo 13 control	w	1A	00011010	Servo 13 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 14 position	w	1B	00011011	Servo position (8uS resolution)								

Register name	Type	Register address		Description								
		Hex	Binary									
Servo 14 control	w	1C	00011100	Servo 14 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 15 position	w	1D	00011101	Servo position (8uS resolution)								
Servo 15 control	w	1E	00011110	Servo 15 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Servo 16 position	w	1F	00011111	Servo position (8uS resolution)								
Servo 16 control	w	20	00100000	Servo 16 control <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>S</td><td>S</td><td>S</td><td>S</td></tr></table> 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)	A	B	C	D	S	S	S	S
A	B	C	D	S	S	S	S					
Global update	w	21	00100001	Global update control								

Register name	Type	Register address		Description								
		Hex	Binary									
Servo 1 status	r	00	00000000	Servo 1 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 2 status	r	01	00000001	Servo 2 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 3 status	r	02	00000010	Servo 3 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 4 status	r	03	00000011	Servo 4 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					

Register name	Type	Register address		Description								
		Hex	Binary									
				Servo 5 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 5 status	r	04	00000100									
				Servo 6 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 6 status	r	05	00000101									
				Servo 7 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 7 status	r	06	00000110									
				Servo 8 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 8 status	r	07	00000111									

Register name	Type	Register address		Description								
		Hex	Binary									
Servo 9 status	r	08	00001000	Servo 9 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 10 status	r	09	00001001	Servo 10 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 11 status	r	0A	00001010	Servo 11 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 12 status	r	0B	00001011	Servo 12 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					

Register name	Type	Register address		Description								
		Hex	Binary									
Servo 13 status	r	0C	00001100	Servo 13 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 14 status	r	0D	00001101	Servo 14 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 15 status	r	0E	00001110	Servo 15 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Servo 16 status	r	0F	00001111	Servo 16 status <table><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table> 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)	A	B	C	D	0	0	0	0
A	B	C	D	0	0	0	0					
Firmware version	r	10	00010000	Firmware (Bit 0-3 = minor version, Bit 4-5 = major version)								

3.7.4 I²C Servo Control Register Format

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

Table 7: I²C Servo Control Register Format

Bit	Bit value	Function	Description
A	128 (0x80)	Operate bit	When set activates the servo being controlled
B	64 (0x40)	Reverse bit	When set reverses the position value for the servo being controlled
C	32 (0x20)	SoftStart bit	When set on servo first activation, see operate bit above, feeds position pulses to the servo in a ramping manner until position is attained
D	16 (0x10)	Speed bit	When set applies the speed value 0 to 15
SSSS	0 – 15 (0x00 – 0x0F)	Speed value	Speed value 0 to 15

3.7.5 I²C Servo Status Register Format

Each servo status register consists of four status bits defined as follows:

Table 8: I²C Servo Status Register Format

Bit	Bit value	Function	Description
A	128 (0x80)	Operation bit	Set when servo is operational
B	64 (0x40)	Reverse bit	Set when servo is in reverse operation
C	32 (0x20)	SoftStart bit	Set when SoftStart is in operation
D	16 (0x10)	Movement bit	Set when servo movement is in-progress

3.7.6 Servo Movement Complete Determination

Servo status register 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.

3.7.7 I²C Global Update Register

Once all the required position & control registers have been set a write to the 0x21_{hex} (Global update) must be made to activate all the new positions.

3.7.8 I²C Write 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 (SV16 Adr)	11001000 _{binary}
Byte 2 (Register 0x00)	0 _{decimal}
Byte 3 (Register 0x01)	30 _{decimal}
Byte 4 (Register 0x02)	144 _{decimal} , 90 _{hex}
Byte 5 (Register 0x03)	35 _{decimal}
Byte 6 (Register 0x04)	144 _{decimal} , 90 _{hex}
Byte 7 (Register 0x05)	40 _{decimal}
Byte 8 (Register 0x06)	144 _{decimal} , 90 _{hex}
Byte 9 (Register 0x07)	45 _{decimal}
Byte 10 (Register 0x08)	144 _{decimal} , 90 _{hex}
Byte 11 (Register 0x09)	127 _{decimal}
Byte 12 (Register 0x0A)	213 _{decimal} , D5 _{hex}
Byte 13 (Register 0x0B)	130 _{decimal}
Byte 14 (Register 0x0C)	213 _{decimal} , D5 _{hex}
Byte 15 (Register 0x0D)	140 _{decimal}
Byte 16 (Register 0x0E)	213 _{decimal} , D5 _{hex}
Byte 17 (Register 0x0F)	150 _{decimal}
Byte 18 (Register 0x10)	213 _{decimal} , D5 _{hex}

then to activate write:

Byte 1 (SV16 Adr)	11001000 _{binary}
Byte 2 (Register 0x00)	21 _{hex}
Byte 3 (Register 0x21)	0 _{decimal}

3.8 Application Software

3.8.1 Demonstration Software

Raspberry-Pi demonstration software written in Python is available to download from the website www.designersystems.co.uk/robotics

4 Electrical Characteristics

4.1 Absolute Maximum Ratings

Absolute maximum ratings for power supply and voltage on digital pins of the module are listed in the following table.

Table 9: Absolute Maximum Ratings

Parameter	Min.	Max.	Unit
Power Supply Voltage (V+)	-0.3	6.0	V
Input Voltage on SDA and SCL	-0.3	3.6	V
Storage temperature	-45	100	°C

4.2 Operating Conditions

Normal operational conditions are listed in the following table.

Table 10: Normal Operating Conditions

Parameter	Min.	Typ.	Max.	Unit
Power Supply Voltage (V+)	4.5	5.0	5.5	V
Power Supply Current			20	mA
Servo Voltage	3.0	7.4**	8.5	V
Servo Current*		1.0		A
Input voltage on SDA and SCL		3.3		V
Operating Temperature	-20	25	50	°C

* 16 Servo running with no physical load

** Standard Lithium-Ion battery pack

4.3 Current Consumption

Normal values for current consumption are listed in the following table.

Table 11: Current Consumption

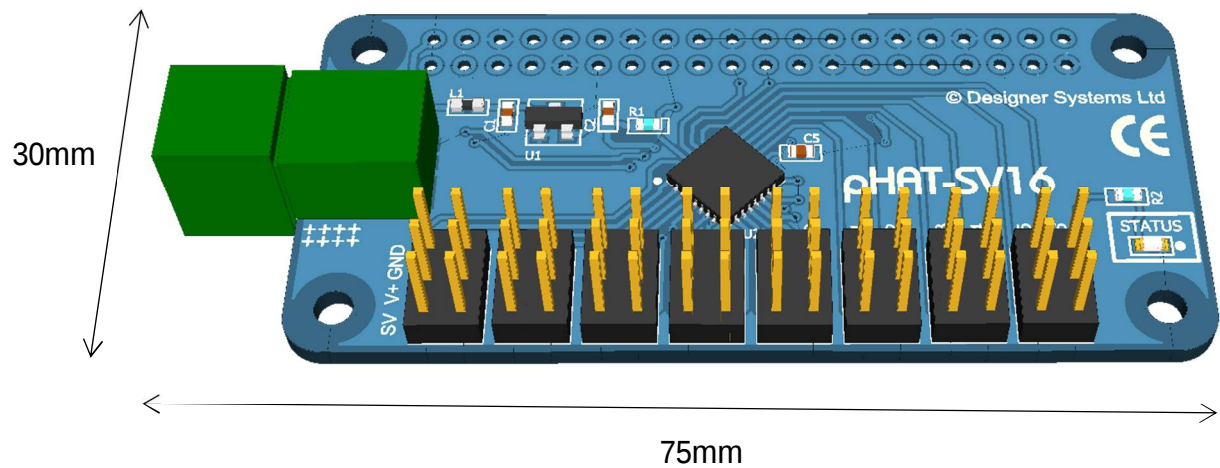
Parameter	Min.	Typ.	Max.	Unit
Supply Current – All servos running		20		mA
Servo Power – All servos running and loaded			6	A

5 Mechanical

5.1 Dimensions

Mechanical drawing – all dimensions in millimetres.

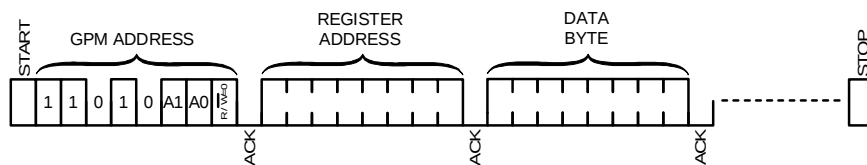
Figure 1: Dimensions



6 References

6.1 I²C protocols

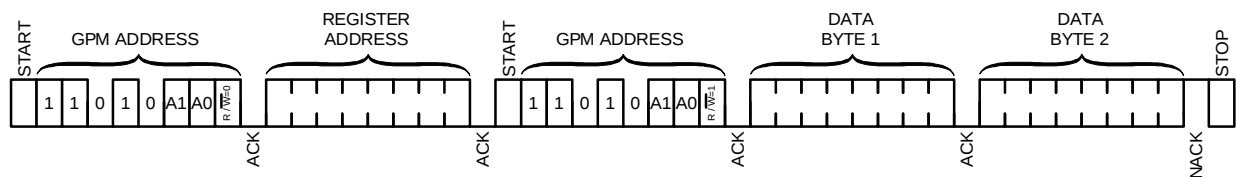
Figure 2: I²C Write protocol



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 SV16 until the 'STOP' condition.

Figure 3: I²C Read protocol



'DATA BYTE 1 & 2' are register values returned from the SV16. Each byte written is acknowledged 'ACK' by the SV16, every byte read is acknowledged 'ACK' by the I2C Master. A Not-acknowledge 'NACK' condition is generated by the I2C Master when it has finished reading.

7 Appendix

Table 12: Related Documents

Document Name	Remark
---------------	--------

Table 13: Terms and Abbreviations

Abbreviation	Description
I ² C	Inter-Integrated Circuit
ESD	Electrostatic Discharge

8 Compliance



WEEE Consumer Notice

This product is subject to Directive 2012/19/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 2011/65/EC (RoHS 2) and 2015/863/WU (RoHS 3) 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), Bis(2-Ethylhexyl) phthalate (DEHP), Benzyl butyl phthalate (BBP), Dibutyl phthalate (DBP) and Diisobutyl phthalate (DIBP).



REACH Compliance

This product complies with Regulation 1907/2006 covering the Registration, Evaluation, Authorisation and restriction of Chemicals (REACH). Designer Systems Ltd confirms that none of its products or packaging contain any of the 174 Substances of Very High Concern (SVHC) on the REACH Candidate List in a concentration above the 0.1% by weight allowable limit.