



Simple Servo™ USER Instructions

PLEASE PAY CAREFUL ATTENTION TO THE FOLLOWING INFORMATION

DO NOT APPLY MORE THAN 6V TO THE UNIT

Power the unit from either a +5V PSU or battery pack (6V max).

Usually the board is powered via the 3 way terminal block. There are a couple of different connection options depending on your application.

Option 1 single 6V battery pack

1. GND terminal to your battery negative
2. +Ve and +5V should both be connected to the battery positive

Option 2 6V battery pack for the Servo motors and a second logic supply derived from your micro board.

1. GND connect this to Battery **and** Logic supply ground
2. +Ve connects to battery pack positive
3. +5V connects to your logic supply positive

In addition to the above there is space on the PCB to solder 0.1" Molex connectors. LOGIC PSU (CN1)

This is strictly a 5V input that powers the microcontroller

SERVO PSU (CN3)

This supplies the servo voltage which may vary from motor to motor

INTRODUCTION

This product provides the following servo motor control features;

PIN Control (CW/CCW/ZERO) allows use of simple toggle switches or logic levels for control

ANALOG Control (Use a pot or analogue voltage to control the motor)

SERIAL Control (Use serial data [TTL] to control the position or speed)

POSITION EEPROM Memory for Zero position (retained through power off)

MODE Control (change mode on the fly)

TOGGLE between two preset positions with simple PIN control

PIN SPEED control and STOP when using motor in continuous rotation mode

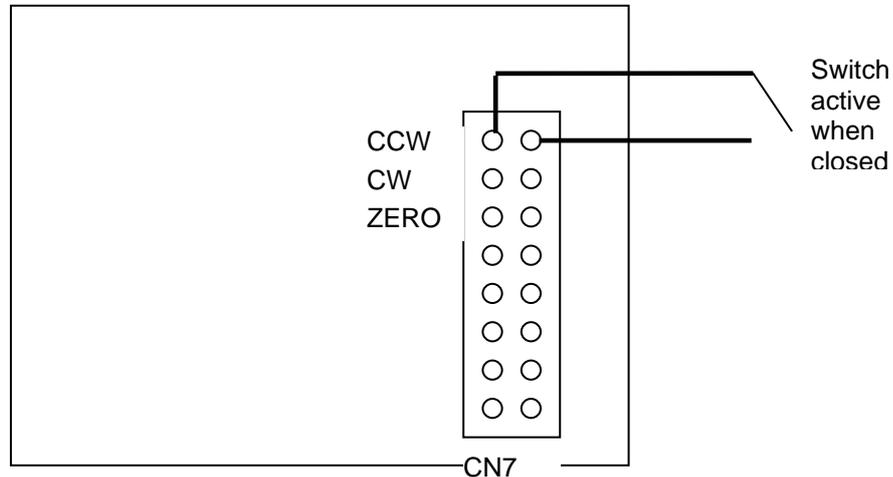
High resolution (5us/0.4°)

LEARN BUTTON onboard the PCB for learning zero/stop position

ACTIVE ROBOTS

Note on terminology:

The CPU port pins are normally pulled high (using internal pull-ups) and are active when brought low (GND), we refer to this (brought low) as Active – as shown below.



Controls When Using a STANDARD 90/180° MOTOR

PIN CONTROLS

CCW - Move Counter Clockwise while input active

CW - Move Clockwise while input active

ZERO - Move to zero position while input active

Speed Settings

SPD1 ○ ○ SLOW
SPD2 ○ ○

SPD1 MEDIUM
SPD2 ○ ○

SPD1 ○ ○ FAST
SPD2

Mode - Link Fitted = Analogue/POT mode control

- No Link = Pin mode control

Learn - 1. **Learn Button**, press learn button/or learn new zero position

- 2. **Learn Pin**, allows an external learn switch to be fitted

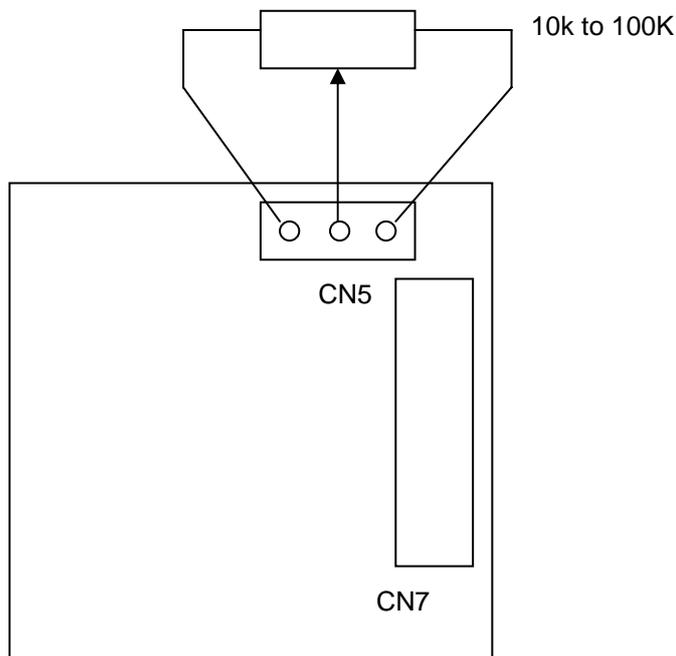
ACTIVE ROBOTS

Note You can be in PIN or POT mode to learn or set up a new centre position, this input should only be temporarily brought low and then released – IE using a normally open button or switch.

POT CONTROL

Fit link to the Mode pins

The servo will now be controlled by the POT input, connect as shown below. In pot mode the centre pin of CN5 becomes an analogue input 2.5V = centre position, which produces a 1500us pulse output to the servo.



TIP – by fitting fixed resistors in the series with the ends of the pot you can adjust the stop positions.

Toggleing between Two positions

1. First in PIN mode position the servo using the CW/CCW controls then press the learn button (1st position)
2. Change to pot mode and position the servo using a potentiometer (2nd Position)
3. Link out the ZERO pins with a jumper
4. Switch in/out the MODE pins to toggle between the position

ACTIVE ROBOTS

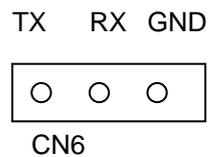
SERIAL CONTROL MODE

To use serial data to control ensure you are in PIN mode (Remove Link from MODE pins)

The serial comms setup is:

9600 Baud, 8 Bits, No Parity, 1 stop bit

The Connections are to CN6 and are TTL level



Protocol:

Send &Hxx wait 100ms Send &Hxx

Where xx is **11** for fully CW, **128** is middle and **244** is fully CCW.

After the second second byte is transmitted the simple servo will echo back the transmitted byte.

Code example in Basic (Bascom-AVR)

```
$baud = 9600
```

```
Dim Tx_data As Byte
```

```
Dim Rx_data As Byte
```

```
Tx_data = 128 'goto centre/1500us pulse
```

```
Rx_data = 0
```

```
Print Chr(Tx_data); 'tell servo board data is coming
```

```
Waitms 100 'wait a 100ms
```

```
Print Chr(Tx_data); 'this is the data the servo board will use
```

```
'could also use 'Put' instead of 'Print' as we don't need Carriage Return
```

```
'above is minimum needed to control servo
```

```
'below also looks for errors
```

```
Try_again: 'jump label
```

```
Waitms 20 'wait 20ms 'servo board should have responded by now
```

```
Rx_data = Inkey() 'optional – you can wait to check that data has been received
```

```
If Rx_data <> Tx_data then 'error in comms if nor equal
```

```
Print Chr$(Tx_data) 'try transmitting again
```

```
Goto Try_again
```

```
End If
```

ACTIVE ROBOTS

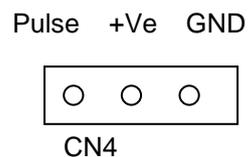
Notes on Using Continuous Rotation Servo Motors

In Pin MODE it is usual to fit the CENTRE link, this ensures when the CW and CCW are not 'active' the motor will stop and not keep rotating.

For this to work correctly you must setup the centre position (or stop position with continuous rotating motors) using the LEARN BUTTON/PIN

The speed select pins SPD1 and SPD2 now control the acceleration not speed of the motor.

Servo Motor Connector



Typical Motor wire colours

Gnd - BLACK or BROWN
+Ve - RED
Pulse – WHITE or ORANGE

The Pulse output is at TTL levels

Typical Applications

Testing Servo motors
Because its has high resolution (5us/0.4°)
Will not go outside 1 to 2ms range

Controlling servo using simple toggle switches
Switch control inputs for all functions, this makes it easy to use for beginners or those that don't have the time to set-up more sophisticated controls

Remote control
Small, POT/analogue or serial input control