

# HEXWALKER INSECT (MOD025)

## Contents:

- Lynxmotion EH1 Hexwalker Kit (x1)
- radio control servo (x3)
- microswitch (x2)
- 10k resistor (x2)
- battery box (x1)

## Required (purchase separately):

- AXE024 PICAXE-08M Servo Driver (x1)  
or AXE031 PICAXE-18X 21 Channel Servo Driver (x1)
- AA alkaline cells (x4)



## Burnt? - Don't Panic!

The 'burnt' look of the plastic is normal, and is a result of the laser cutting process used to manufacture the parts. The burn/smoke marks are actually only on the thin outer protective layer, which needs to be peeled off before use! Note that each piece of plastic has this protective cover on both sides (it may be transparent or white). To remove simply use a sharp point to lift the edge and then peel away from the plastic.

## Assembly

The illustrated assembly guide for the EH1 model is online at [www.lynxmotion.com](http://www.lynxmotion.com)  
For spare or missing assembly parts please contact [www.lynxmotion.com](http://www.lynxmotion.com) directly.

## Servo Test program (PICAXE-08M on AXE024)

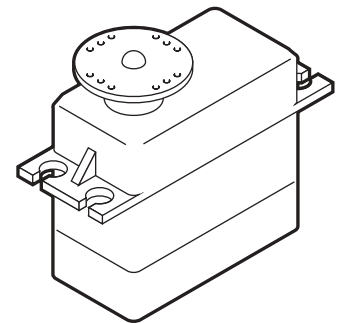
To centre the servos (for assembly) on the AXE024, type in and download this program:

```
servo 1, 150
servo 2, 150
servo 4, 150
stop
```

## Servo Test program (PICAXE-18X on AXE031, using channels 1,2,3)

To centre the servos (for assembly) on the AXE031, type in and download this program:

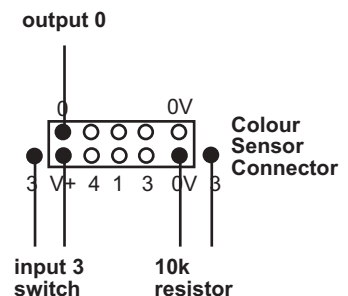
```
i2cslave $c2, i2cslow, i2cbyte
writei2c 63, (128)
writei2c 64, (128)
writei2c 65, (128)
stop
```



## Information

Example programs for the EH1 hexwalker are available for download at [www.picaxe.co.uk](http://www.picaxe.co.uk)  
The programs are also automatically installed within the /samples subfolder of the Programming Editor software.

The included resistor and switches can be used to build a 'obstacle detection system'. For the AXE024 the switches (NO and C contacts) must be connected in parallel, and then connected to the driver board (input3) as shown in the diagram. For the AXE031 the switches can be connected to any of the spare PICAXE-18X inputs (see AXE031 datasheet).



## ' EH1 Hexwalker PICAXE-08M Program (AXE024)

```
' www.picaxe.co.uk
' Program to make the EH1 Hexwalker walk forwards
' until input switch activated. Then reverse and
' turn before going off forwards again.

'*** Input Output Pins
'0 (not used)
'1 centre servo
'2 right side servo
'3 bumper switch
'4 left side servo

'*** Program constants
' Delay to slow down servo motion (30ms default)
symbol servo_delay = 30
' Number of steps backwards and turning
symbol no_back_steps = 8
symbol no_turn_steps = 5

'*** Initialisation
init:
' set servos to known position
    servo 1,120
    servo 2,120
    servo 4,120
    pause servo_delay

' set interrupt to occur when input3 high
    setint %00001000,%00001000

'*** Move forward
move_forward:
    for b1 = 120 to 180
        servo 1,b1
        pause servo_delay
    next b1
    for b1 = 120 to 180
        servo 4,b1
        servo 2,b1
        pause servo_delay
    next b1
    for b1 = 180 to 120 step -1
        servo 1,b1
        pause servo_delay
    next b1
    for b1 = 180 to 120 step -1
        servo 4,b1
        servo 2,b1
        pause servo_delay
    next b1

    goto move_forward
```

```
'*** Interrupt sub
' When interrupt occurs back up and then turn
interrupt:
'Backwards
    for b2 = 1 to no_back_steps
        for b1 = 180 to 120 step -1
            servo 1,b1
            pause servo_delay
        next b1
        for b1 = 120 to 180
            servo 4,b1
            servo 2,b1
            pause servo_delay
        next b1
        for b1 = 120 to 180
            servo 1,b1
            pause servo_delay
        next b1
        for b1 = 180 to 120 step -1
            servo 4,b1
            servo 2,b1
            pause servo_delay
        next b1
    next b2

'Turn
    for b2 = 1 to no_turn_steps
        for b1 = 180 to 120 step -1
            servo 1,b1
            pause servo_delay
        next b1
        let b2 = 180
        for b1 = 120 to 180
            servo 4,b1
            servo 2,b2
            let b2 = b2 - 1
            pause servo_delay
        next b1
        for b1 = 120 to 180
            servo 1,b1
            pause servo_delay
        next b1
        let b2 = 120
        for b1 = 180 to 120 step -1
            servo 4,b1
            servo 2,b2
            let b2 = b2 + 1
            pause servo_delay
        next b1
    next b2

'Reset interrupt
    setint %00001000,%00001000
    return
```