micromagic systems

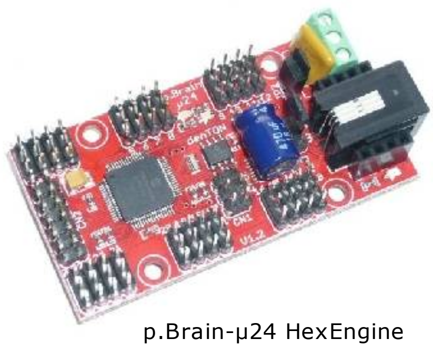
Animatronic & Puppet control systems for Film & Television

***p.Brain-HexEngine  V1.21*** *(Updated 12/10/09)*
***Configuration Guide***


## Contents



p.Brain-ds24 HexEngine
(Requires p.Brain-SMB)



p.Brain-µ24 HexEngine

## Description

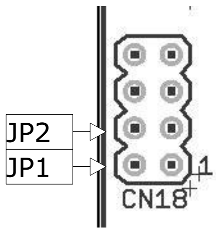
The *HexEngine* is a software module designed for the *p.Brain-ds24* or *p.Brain-μ24* hardware modules. The *HexEngine* is a user configurable locomotion engine for hexapod robots. *HexEngine* is configured via a serial port (RS-232 or TTL level) using standard terminal software such as HyperTerminal or TerraTerm (As HyperTerminal is no longer included with Vista, I recommend using TeraTerm Pro available here: TeraTerm Pro V2.3). Once the *HexEngine* is configured for the target hexapod platform, the hexapod locomotion can be controlled using simple direction commands over either the RS-232 or TTL serial port. The idea of the *HexEngine* is to allow you to get your hexapod up and running with minimum fuss, allowing you to concentrate on obstacle avoidance or other high level control applications.

## Key Features

- Configuration via terminal port or control port via optional blue tooth link
- Locomotion control via terminal port or control port
- Configurable body geometry
- Configurable leg geometry
- Multiple Servo output re-map configurations
- Gait configurations
- X,Y,Z Body rotation and translation
- Walk in any direction
- Servo rotation calibration
- Servo reverse for each leg joint
- Leg calibration offsets
- Four selectable gaits patterns (can be changed during locomotion )
- Two selectable leg move styles, giving a total of eight individual gaits.
- Eight supported baud rates
- Three PWM servo ranges, standard, extended and full (500 to 2500uS)
- Dedicated PWM hardware for precision 1uS PWM resolution

## Port Configuration – (p.Brain-ds24)

There are two serial ports on the *p.Brain-ds24* which are used for locomotion control and/or terminal configuration. Using jumpers JP1 and JP2 on CN18 of the *p.Brain-SMB* there are four possible combinations for port configuration:




| | JP2 | JP1 | UART1 TTL | UART2 RS232 |
|---|---|---|---|---|
| 1 | OFF | OFF | Locomotion port | Configuration port |
| 2 | OFF | ON | | Locomotion & Configuration port |
| 3 | ON | OFF | Locomotion & Configuration port | |
| 4 | ON | ON | Configuration Port | Locomotion Port |

The baud rate for the TTL port is set by 'CBR' and the RS-232 baud is set by 'TBR' within the configuration menu. The default baud rates are: TTL port 9.6kbps, RS-232 port 115.2kbps.
*Note: All port communications are **8 bit, No Parity, 1 Stop bit (8N1)***

## Port Configuration – (p.Brain-μ24)

There are two serial ports on the *p.Brain-μ24* which are used for locomotion control and/or terminal configuration. UART2 is an TTL serial port which can be used for locomotion control only. UART1 is connected to either the RS-232 RJ11 port or the optional ESD200 bluetooth adaptor using JP1. This port defaults to PIP locomotion control mode, but can also be used for the hexapod configuration mode / terminal port.

## Terminal Connection

In order to configure your *HexEngine*, you will require terminal software such as HyperTerm or Tera Term, a free serial port and a suitable RS232 lead. If you are using the *p.Brain-SMB* in conjunction with the *p.Brain-ds24* or the *p.Brain-μ24*, there is an RJ11 4/4 to DB9 RS232 cable available from micromagic systems (MSR-RJ11-232). If you wish to make your own cable you will need to use the following connections:

| RJ11 4/4 Connector | DB9 Female Connector | Description |
|---|---|---|
| 1 | 3 | RS232 RX |
| 2 | 2 | RS232 TX |
| 3 | 5 | GND |
| 4 | | Unused |

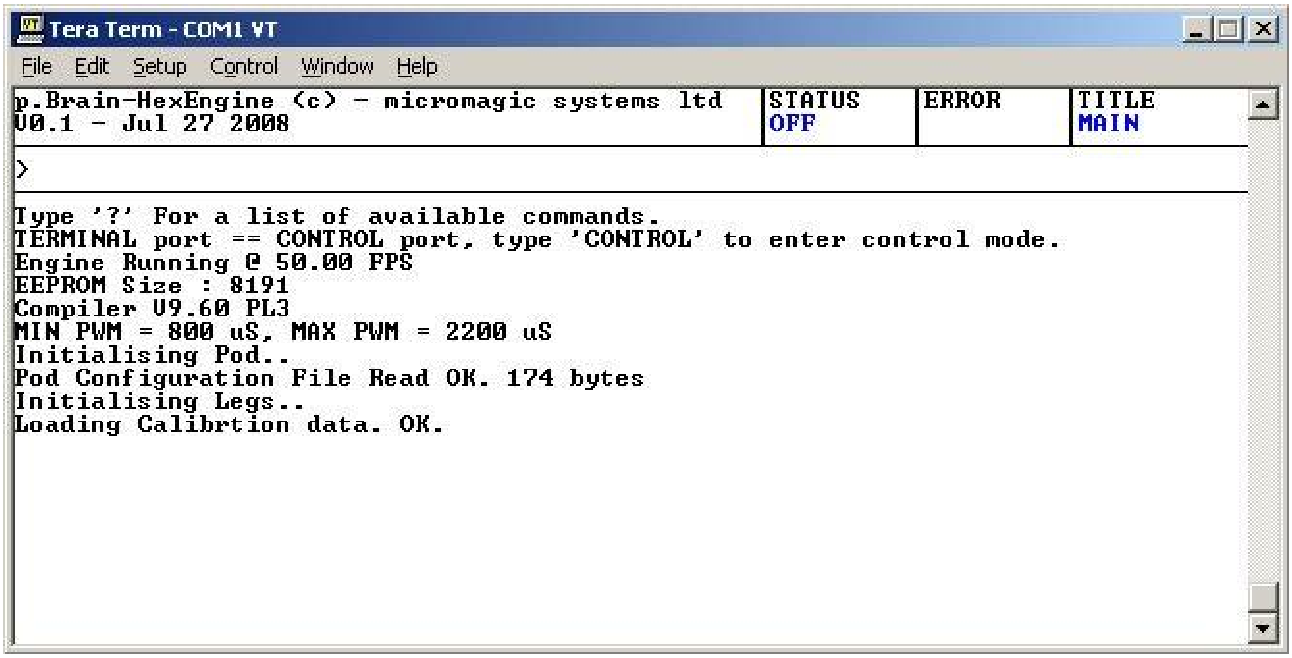
### *p.Brain-RJ232*

Micromagic systems RS-232 DB9 to RJ11 serial port adaptor.

**p.Brain-ds24:**
It is also possible to configure the *HexEngine* via UART1's TTL interface, in order to do this you will require a suitable TTL interface converter to your host system or a blue tooth module such as the ESD200 connected to UART1. If you are using the *p.Brain-SMB* PCB, there is a suitable socket for the ESD200, please see the p.Brain-SMB user guide for installation guidance.

**p.Brain-μ24:**
It is possible to configure the *HexEngine* using either the RS-232 interface, the ESD200 bluetooth adaptor, or a TTL compatible serial port using the ESD200 socket. See the *p.Brain-μ24* user guide for ESD200 installation and configuration.

## Boot Screen




Upon boot, you should see a boot screen similar to the one above. At the top is the title bar, which displays software name and version, hexapod status, error codes and page title. Upon boot various information is listed such as the frame rate, EEPROM size, PWM range and compiler version.

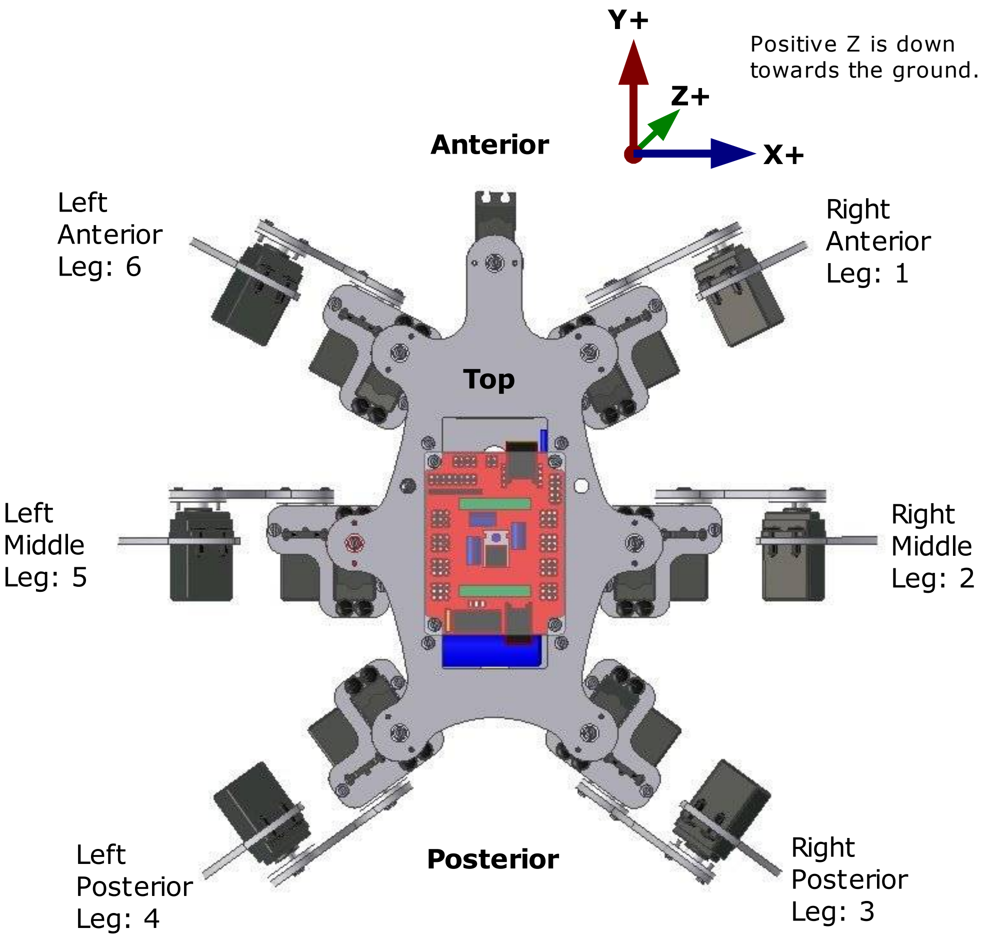
The *HexEngine* will also load the Pod (short for hexapod) configuration file and Pod calibration file. If there is a problem with either of these files you will be informed during boot. The configuration file holds all the geometry, gait and default boot values for the pod. If this is the first time the *HexEngine* has been run, it will have loaded up with a default set of geometry values, so the first thing to do is to enter the configuration menu.

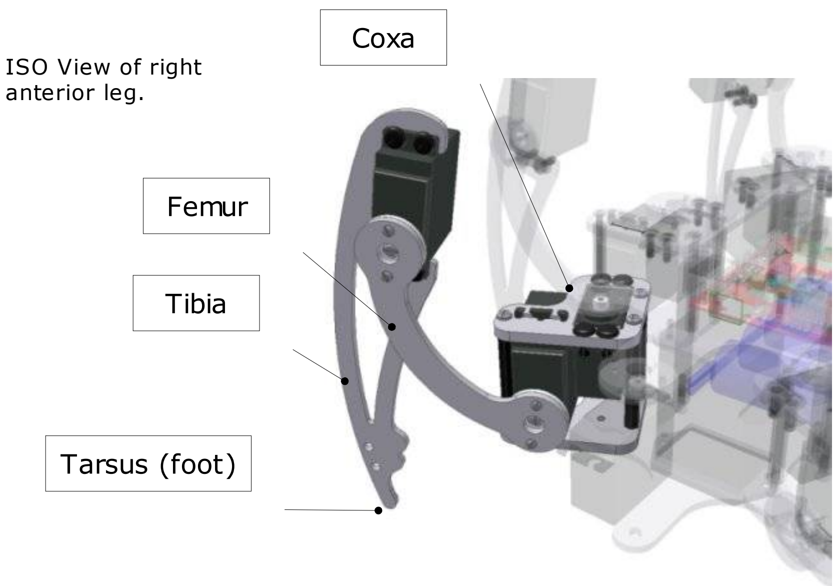
If you are using a *p.Brain-μ24*, to enter configuration mode type “@@@”, do not type any other key for at least one second. ( To exit configuration mode, return to the MAIN menu and type “PIP Control” followed by the ENTER key.

Type '?' followed by the ENTER key to see a list of commands for this menu screen.

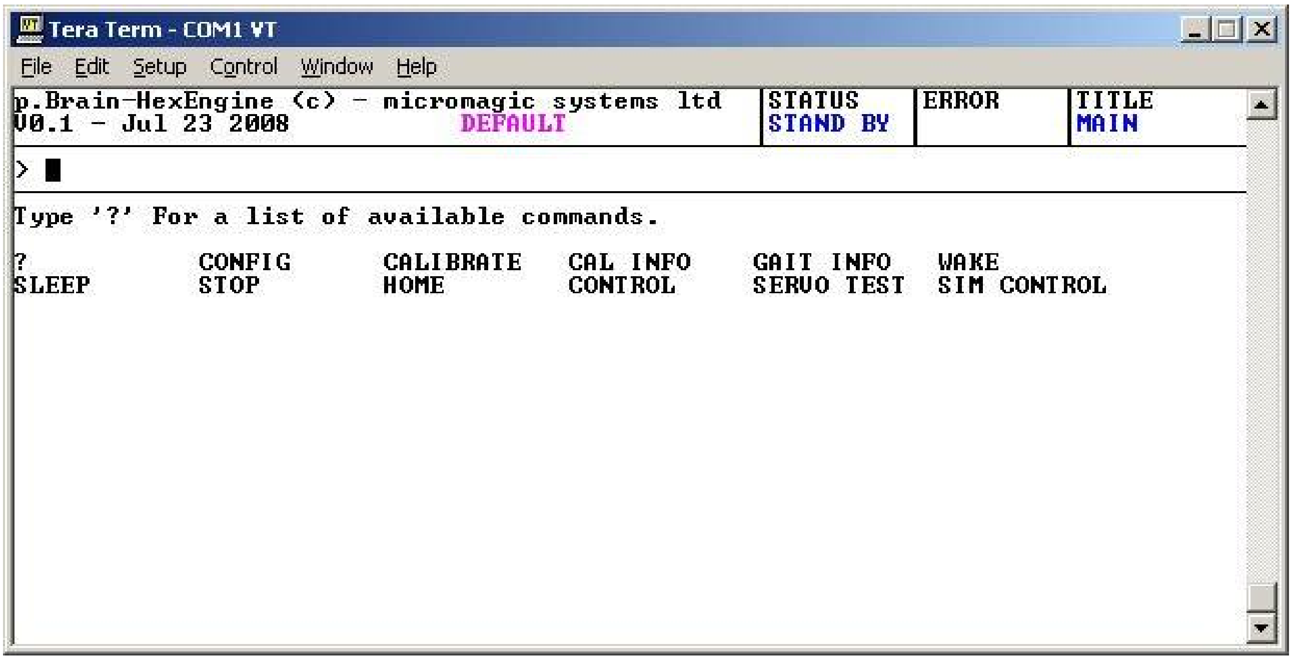
## Hexapod (Pod) Geometry

Before you start configuring your hexapod, take time to study the following drawings to help you understand the basic geometry of a hexapod. *Note: print this out and keep it handy*.



## Main Menu




After the initial boot message the hexapod is ready for locomotion control or one of the commands listed in the main menu. Locomotion control only works within the main menu and the gait info screen, if the hexapod is powered up and one of the configuration screens is entered such as 'CONFIG' the hexapod will power down.

## Main Menu Commands Table

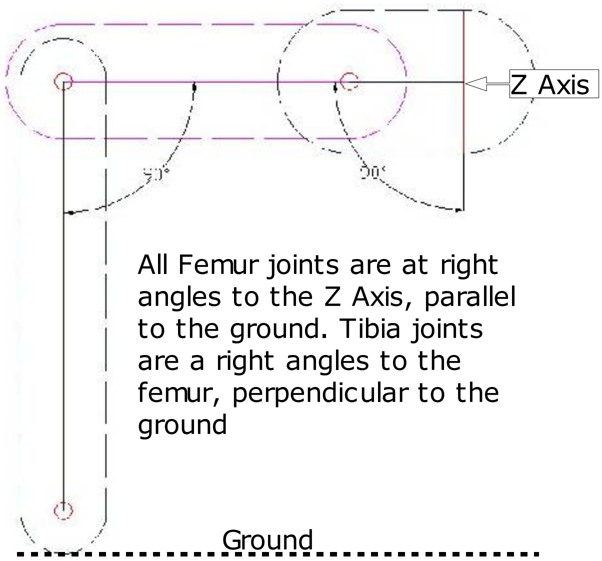
| Command | Description |
|---|---|
| ? | List all available commands within this screen |
| CONFIG | Enter configuration screen |
| CALIBRATE | Enter servo calibration screen |
| CAL INFO | Display calibration data |
| LEG INFO | Display tarsus position and leg angles |
| GAIT INFO | Enter gait information screen |
| WAKE | Wake the hexapod |
| SLEEP | Hexapod into standby |
| STOP | Stop the Hexapod |
| HOME | Reset Hexapod legs to home position |
| SERVO TEST | Enter servo test screen |
| PIP CONTROL | Enter (Packet Interface Protocol) locomotion control mode |
| SIM CONTROL | Enter Simple locomotion control mode |
| PORT TEST | Enter I/O port test screen |
| PS2 TEST | View PS2 Gamepad connection status and values |
| | |

## Main Menu Commands

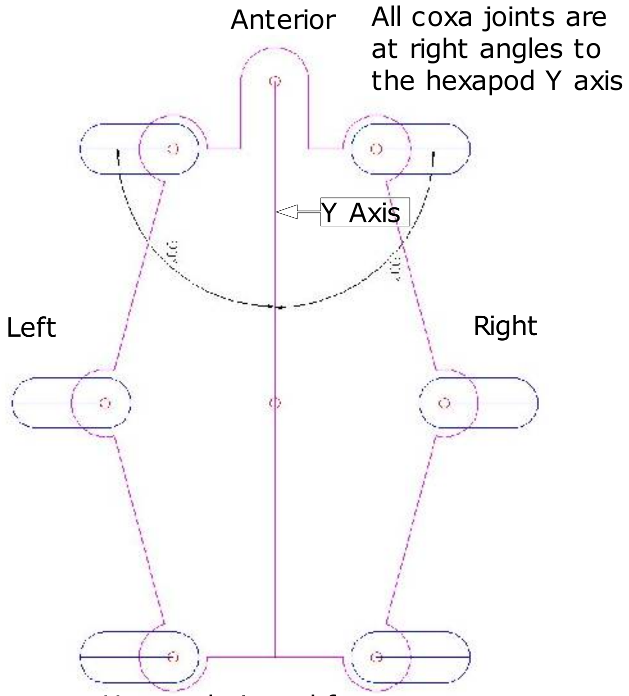
### Calibrate

In this screen the servo calibration offsets can be set for each leg joint. Servo calibration offset can be between -127 to 127 uS. Before beginning calibration, it is best to move your hexapod legs into the approximate calibration position, and to stand your hexapod off of the ground so that none of the legs come in contact with the floor. ***Be prepared for sudden servo moves. Also if your configuration file is not set-up correctly servos may drive into the body or other legs, so always be ready to stop the servo refresh with the '-' key or switch off the servo power.***

### *Calibration Position Diagram*




Right Anterior leg viewed from front.




Hexapod viewed from top.

When entering calibration mode global servo refresh is switched off, and is only switched on when the '+' key is pressed. To switch off global servo refresh press the '-' key. You will see a picture of the hexapod at the right side of the screen, with the current leg highlighted in red, to change selected leg use the number keys '1' to '6'. when selecting a leg, all individual servo enable bits are switched off. To enable a servo and start calibration press one of the servo jog keys Coxa: 'A', 'Z', Femur: 'S', 'X', Tibia: 'D', 'C'. The relevant servo will spring into life, and you can adjust its position using the appropriate keys. The calibration position is shown in Picture 5. Once you have a leg calibrated save the data using the 'F' key and move to the next until all 6 legs are done. You can also check the leg movement by pressing the '0' key. This will move the selected leg to the centre angles defined by ACA, MCA, PCA, CFA & CTA. To return to the calibration position press key '9'

Once you have finished calibrating all legs, press 'Q' to quit, you will be prompted to save the calibration data.

*Note: It's a good idea to spend time getting your calibration and servo ranges correct, the better the calibration, the smoother your hexapod will move. You can check how good your calibration is working by using the 'Wake' command from the main menu to check the hexapods neutral standing position.*

**Cal Info**
This command will list the current servo calibration offsets

**Leg Info**
This command will list the leg tarsus (foot) position in X,Y & Z coordinates, along with the leg joint angles.

**Gait Info**
This screen displays leg phase, tarsus position (X,Y,Z), leg angles, body rotation and translation, and drive variables. The screen is updated twice a second. Very useful for debugging. To exit this screen press any key. *Note: This screen can be very useful the first time you configure your hexapod. If you are unsure if you have configured the geometry correctly, have a look at the tarsus positions and leg angles within this screen to see if they are what you would expect. You can also disconnect the servo power source and run the hexapod through some moves without the servos moving to check the leg angles first.*

**Wake**
This will wake the hexapod up from standby mode, e.g. Stand the hexapod up into its neutral standing position. This is for testing purposes.

**Sleep**
This will put the hexapod into standby mode.

**Stop**
This will stop the hexapod if it happens to be moving.

**Home**
If the hexapod is standing, and the legs are not in their neutral position, this will move the legs back to their neutral position, e.g. Relax the hexapod.

**Control**
When both the locomotion and configuration port are configured to use the same UART, (e.g. Mode 2 & 3 of Port Configuration Table ) upon boot the *HexEngine* will default to configuration mode. In order to enter locomotion mode, use the 'CONTROL' command. To return to configuration mode once in locomotion mode, type '@@@' three times with no enter key followed by a one second pause.

**Servo Test**
This enables direct driving of each servo using the cursor keys left & right to select a servo, and the cursor key up & down to change the servo position. The servo position is listed in micro seconds. There are three pre-defined servo positions using the number keys '1', '2' & '3' which are equal to 1000uS, 1500uS, 2000uS respectively. Pressing one of these keys will drive the servo from its current position to the desired position over a one second interval. Pressing any key during the automated move will stop the servo in the current position. Use the 'Q' key to quit servo test.

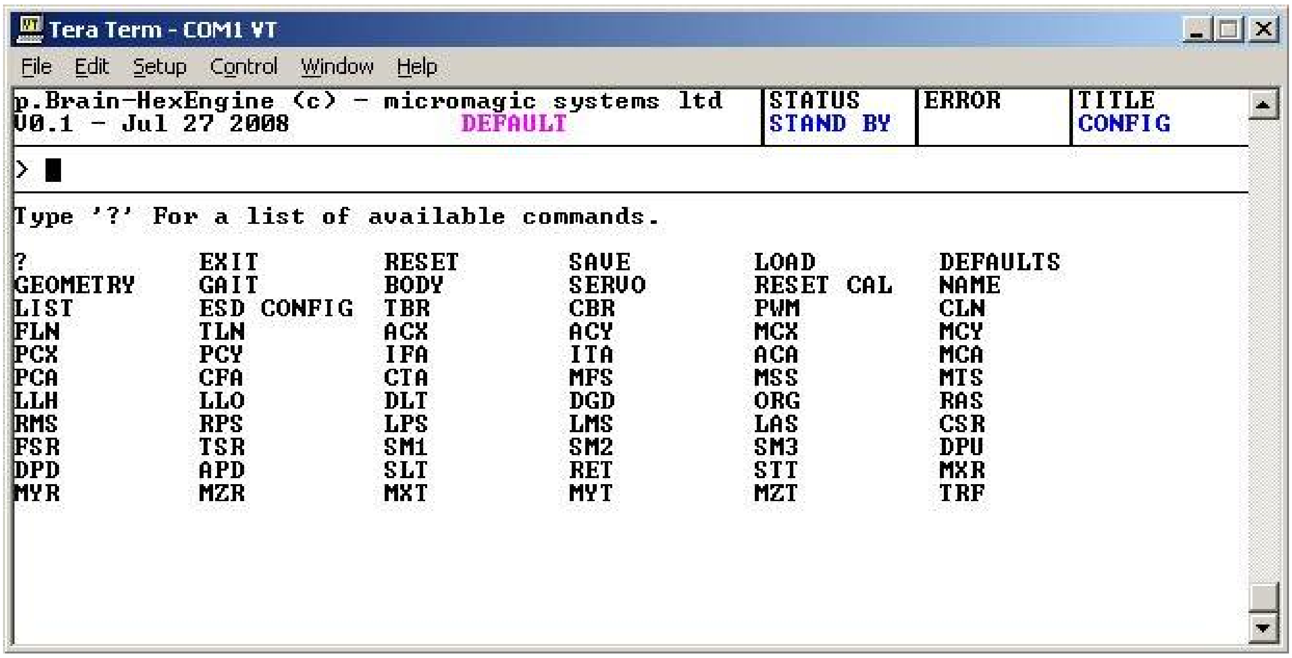
**Sim Control**
The Sim Control screen offers a very simple locomotion interface to the hexapod for testing and control using single key commands, which allows the hexapod to be controlled using a simple terminal interface. The ENTER key is not required after a command key, for example, to walk forward press and hold the 'w' key, upon releasing the key the hexapod will automatically stop. Upon entering 'Sim Control' the following keys control the hexapod:

| Key | Description |
|---|---|
| + | Power up hexapod |
| - | Power down hexapod |
| SPACE | Stop hexapod |
| ! | Emergency stop hexapod ( Shuts off servos instantly ) |
| w | Walk forward |
| s | Walk backwards |
| a | Turn Left |
| d | Turn Right |
| q | Crab Left |
| e | Crab Right |
| 1 | Wave gait 1 (slowest) |
| 2 | Wave gait 2 |
| 3 | Wave gait 3 (In my opinion, the best!) |
| 4 | Tripod gait (fastest ) |
| 5 | On Road gait ( fast, fluid ) |
| 6 | Off Road gait (slower, better ground clearance ) |
| 7 | Decrease leg transfer speed by 0.1 seconds |
| 8 | Increase leg transfer speed by 0.1 seconds |
| 9 | Reset leg transfer speed to power on default (DLT) |
| r | Reset legs to neutral position |
| b | Switch on full 3D balance mode |
| c | Switch off full 3D balance mode |
| ESC | Return to main menu |

**Port Test**
This screen displays the digital and analogue I/O status. The port is configured using the DIO and ADC configuration commands.

**PS2 Test**
This screen displays the PS2 gamepad status and variables, such as buttons and the two analogue joystick values. See PS2 Gamepad Control for further information.

## Configuration Menu




Within this menu all of the geometry, gait and power up settings can be configured. A full list of commands can be viewed by typing '?' followed by the enter key. Most commands are three letters long and are followed by the ENTER key. Commands longer than three letters are used to view parameters and/or change menu's. For example type 'EXIT' to return to the MAIN menu. *Note: When entering the configuration menu, the hexapod will automatically power down. The Pod cannot be run while in configuration mode.*

To query a configuration setting, type the three character command followed by a carriage return (denoted as <cr> in examples). The command will be repeated followed by the current setting. Example to query tibia length:

```
>TLN<cr>

TLN=100.00
```

To change a configuration setting, type the three character command followed by '=' followed by the desired value. If the value is in the valid range for the command, the command and value are echoed back, if the value is outside the valid range for the command, the value is clamped to the nearest limit. Example to set tibia length:

```
> TLN=120.0<cr>

TLN=120.0
```

Example of range clamping:

```
> TLN=0<cr>

TLN=25
```

## Configuration Menu Commands Table

KEY:
COMMANDS
P.BRAIN CONFIGURATION
GEOMETRY CONFIGURATION
GAIT CONFIGURATION
SERVO CONFIGURATION
BEHAVIORAL CONFIGURATION
BODY MOTION CONFIGURATION

***Table 1***

| Command | Description | Units | Min | Max |
|---|---|---|---|---|
| ? | List all available commands within this screen | | | |
| EXIT | Exit configuration menu back to main menu | | | |
| RESET | Soft reset the *p.Brain-ds24* | | | |
| SAVE | Save all configuration data | | | |
| LOAD | Load all configuration data | | | |
| DEFAULTS | Reset all configuration data to default values | | | |
| GEOMETRY | Displays a List of leg geometry | | | |
| GAIT | Displays a list of gait configuration | | | |
| BODY | Displays a list of body geometry | | | |
| SERVO | Display all servo configuration settings | | | |
| RESET CAL | Resets all leg offset calibration data to 0.0 | | | |
| LIST | LIST all configuration details | | | |
| ESD CONFIG | Configure ESD200 blue tooth module | | | |
| MSRH01 | Reset all configuration data to MSR-H01 kit values | | | |
| PHOENIX | Reset all configuration data to PHOENIX kit values | | | |
| | | | | |
| NAME | Pod file name | string | 0 | 16 |
| TBR | Terminal port BAUD Rate | Integer | 0 | 7 |
| CBR | Control port BAUD Rate | Integer | 0 | 7 |
| PWM | Servo Pulse Width Modulation Range Limit | | 0 | 2 |
| | | | | |
| CLN | Coxa length | mm | 0 | 300 |
| FLN | Femur length | mm | 25 | 300 |
| TLN | Tibia Length | mm | 25 | 300 |
| ACX | Right Anterior Coxa X | mm | 0 | 250 |
| ACY | Right Anterior Coxa Y | mm | 0 | 250 |
| MCX | Right Middle Coxa X | mm | 0 | 250 |
| MCY | Right Middle Coxa Y ( Must be = 0.0 ) | mm | 0 | 0 |
| PCX | Right Posterior Coxa X | mm | 0 | 250 |

| Command | Description | Units | Min | Max |
|---|---|---|---|---|
| PCY | Right Posterior Coxa Y | mm | -250 | 0 |
| IFA | Initial Femur angle | Degrees | 0 | 90 |
| ITA | Initial Tibia angle | Degrees | -180.0 | -90.0 |
| ACA | Anterior Coxa Centre angle | Degrees | -45.0 | +45.0 |
| MCA | Middle Coxa Centre angle | Degrees | -45.0 | +45.0 |
| PCA | Posterior Coxa Centre angle | Degrees | -45.0 | +45.0 |
| CFA | Centre Femur Angle | Degrees | -45.0 | +45.0 |
| CTA | Centre Tibia Angle | Degrees | -45.0 | +135.0 |
| | | | | |
| RAS | Right anterior servo reverse bits | Bits | 000 | 111 |
| RMS | Right middle servo reveres bits | Bits | 000 | 111 |
| RPS | Right posterior servo reverse bits | Bits | 000 | 111 |
| LPS | Left posterior servo reverse bits | Bits | 000 | 111 |
| LMS | Left middle servo reverse bits | Bits | 000 | 111 |
| LAS | Left anterior servo reverse bits | Bits | 000 | 111 |
| CSR | Coxa servo range | Degrees | 0.0 | 180.0 |
| FSR | Femur servo range | Degrees | 0.0 | 180.0 |
| TSR | Tibia servo range | Degrees | 0.0 | 180.0 |
| SM1 | Servo Remap SMB Hexapod Standard | | | |
| SM2 | Servo Remap Inverted SMB Hexapod Standard | | | |
| SM3 | Servo Remap SMB Standard servo numbering | | | |
| | | | | |
| MFS | Maximum forward leg stroke (Walk) | mm | 5 | 250 |
| MSS | Maximum sideways leg stroke (Crab) | mm | 5 | 250 |
| MTA | Maximum turn angle (Turn) | Degrees | 2 | 30 |
| LZR | Landing Zone Radius | mm | 0 | 150 |
| LLH | Leg lift height (normal mode) | mm | 10 | 250 |
| LLO | Leg lift height (Off road mode) | mm | 10 | 250 |
| DLT | Default leg transition speed | seconds | 0.3 | 10.0 |
| DGD | Default gait delay | Integer | 0 | 3 |
| ORG | Off road gait enable | bit | 0 | 1 |
| B3D | Full 3D balance gesture | bit | 0 | 1 |
| B3T | 3D balance rotation gain | real | 0.0 | 2.0 |
| B3R | 3D balance translation gain | real | 0.0 | 2.0 |
| | | | | |
| DPU | Default power up height | mm | 10 | 250 |
| DPD | Default power down height | mm | 0 | 250 |

| Command | Description | Units | Min | Max |
|---|---|---|---|---|
| APD | Auto power down enable | bit | 0 | 1 |
| SLT | Sleep Time Out | Seconds | 1 | 255 |
| RET | Relax Time Out | Seconds | 1 | 255 |
| STT | Stop Time Out | Seconds | 0.5 | 10.0 |
| RTR | Auto Reset Translate & Rotate | bit | 0 | 1 |
| SRC | Servo Range Check | bit | 0 | 1 |
| ASF | Auxiliary Servo Filter | Integer | 0 | 25 |
| ADC | External Port Analogue / Digital Configure | string | 8 | 8 |
| DIO | External Port Digital Direction Configure I/O | string | 8 | 8 |
| PIP | PIP Packet Mode | bit | 0 | 1 |
| TXD | Command Port Transmission Delay | Integer | 0 | 10 |
| PS2 | PS2 Gamepad Control Mode | bit | 0 | 1 |
| | | | | |
| MXR | Maximum X Rotation | degrees | 0 | 20 |
| MYR | Maximum Y Rotation | degrees | 0 | 20 |
| MZR | Maximum Z Rotation | degrees | 0 | 20 |
| MXT | Maximum X Translation | mm | 0 | 100 |
| MYT | Maximum Y Translation | mm | 0 | 100 |
| MZT | Maximum Z Translation | mm | 0 | 100 |
| TRF | Translate & Rotate Filter | real | 0.0 | 50.0 |
| PTR | Post-Translate Body | bit | 0 | 1 |
| PAM | Head PAN servo mid position | Integer | 500 | 2500 |
| PA+ | Head PAN servo max position | Integer | 500 | 2500 |
| PA- | Head PAN servo min position | Integer | 500 | 2500 |
| TIM | Head TILT servo mid position | Integer | 500 | 2500 |
| TI+ | Head TILT servo max position | Integer | 500 | 2500 |
| TI- | Head TILT servo min position | Integer | 500 | 2500 |
| TWM | Head TWIST servo mid position | Integer | 500 | 2500 |
| TW+ | Head TWIST servo max position | Integer | 500 | 2500 |
| TW- | Head TWIST servo min position | Integer | 500 | 2500 |
| GRM | Head GRIPPER servo mid position | Integer | 500 | 2500 |
| GR+ | Head GRIPPER servo max position | Integer | 500 | 2500 |
| GR- | Head GRIPPER servo min position | Integer | 500 | 2500 |
| LPT | Load PAN/TILT/TWIST mid position upon wake | bit | 0 | 1 |
| HSI | Head Servos Invert Bits | bits | 0 | 111 |
| AHL | Auto Head Leveling | bit | 0 | 1 |
| PTR | Post Translate Body | bit | 0 | 1 |

# Configuration Menu Commands

## P.BRAIN CONFIGURATION

### NAME
Set or read the name string of the p.Brain. Example:

```
NAME=MSR-H01<cr>
```

### TBR
Set the baud rate for the RS232 terminal port UART2 on the dsPIC, see Table 2 for available baud rates. *Note: The first time the HexEngine is run, the terminal baud rate should be 115200kbps. All port communications are 8 Bit, No parity, 1 Stop bit ( **8N1** )*

***Table 2***

| Configuration Value | Baud Rate | Configuration Value | Baud Rate |
|---|---|---|---|
| 0 | 1200 | 4 | 19200 |
| 1 | 2400 | 5 | 38400 |
| 2 | 4800 | 6 | 57600 |
| 3 | 9600 | 7 | 115200 |

### CBR
Set the baud rate for the TTL terminal port UART1 on the dsPIC, see Table 2 for available baud rates. The baud rate error is listed next to the baud rate, with higher baud rate errors, problems may occur if the host system also has a baud rate error of the opposite magnitude. For the first time the *HexEngine* is run, the control baud rate should be 9600kbps. *Note: All port communications are 8 Bit, No parity, 1 Stop bit ( **8N1** ). If you change the CBR setting and you are using an ESD200 blue tooth module, you will need to run the ESD CONFIG command to configure the ESD200 to the new baud rate.*

### PWM
Set the global maximum and minimum PWM servo output range. There are three settings, standard, extended and full. Most servo will operate quite happily with standard and extended ranges, however, not all servos will tolerate the full range. Most analogue servos will accept full range signals, but may have mechanical stops which prevent full range use, many digital servos have full range mechanical capabilities, but cut out if the signal goes beyond the extended range limits. It is rare that a hexapod will need the full range limits, and therefore I suggest using either standard or extended limits.

| Setting | PWM Minimum | PWM Maximum | Approx. Rotation (degrees) |
|---|---|---|---|
| 0 | 1000 | 2000 | +/- 50 |
| 1 | 800 | 2200 | +/- 70 |
| 2 | 600 | 2400 | +/- 90 |
| 3 | 500 | 2500 | +/- 100 |

## GEOMETRY CONFIGURATION

**CLN**
Coxa Length. The distance from the centre of the coxa servo spline, to the centre of the femur servo spline. (See Diagram 2)

**FLN**
Femur Length. The length from the centre of the femur servo spline, to the centre of the tibia servo spline. (See Diagram 2)

**TLN**
Tibia Length. The length from the centre of the tibia servo spline to the end of the tibia/foot. (See Diagram 2)

**ACX**
Anterior Coxa X. Distance from anterior coxa servo centre to centre of body in X direction. This is the same for both left and right anterior coxa joints. *Note: The right side of the hexapod is positive X direction. (See Diagram 1)*

**ACY**
Anterior Coxa Y. Distance from anterior coxa servo centre to centre of body in Y direction. This is the same for both left and right anterior coxa joints. *Note: The anterior of the hexapod is positive Y direction. (See Diagram 1)*

**MCX**
Middle Coxa X. Distance from middle coxa servo centre to centre of body in X direction. This is the same for both left and right middle coxa joints. *(See Diagram 1)*

**MCY**
Middle Coxa Y. Distance from middle coxa servo centre to centre of body in Y direction. This is the same for both left and right middle coxa joints. *Note: The middle coxa joint must be in the middle of the body, therefore this value must be 0.0. (See Diagram 1)*

**PCX**
Posterior coxa X. Distance from posterior coxa servo centre to centre of body in X direction. This is the same for both left and right posterior coxa joints. *(See Diagram 1)*

**PCY**
Posterior Coxa Y. Distance from posterior coxa servo centre to centre of body in Y direction. This is the same for both left and right posterior coxa joints. *Note: The anterior of the hexapod is positive Y direction. (See Diagram 1)*

**IFA**
Initial Femur Angle. This is the femur angle when the hexapod is in its neutral standing position. *Note: This value is only used during the hexapod power up in order to set the initial leg angles and placement. The DPU setting is also used to calculate the initial standing position, therefore the final standing femur angle may differ from the value set for IFA.*

**ITA**
Initial Tibia Angle. This is the tibia angle when the hexapod is in its neutral standing position. *Note: This value is only used during the hexapod power up in order to set the initial leg angles and placement. The DPU is also used to calculate the initial standing position, therefore the final standing femur angle may differ from the value set for IFA.*

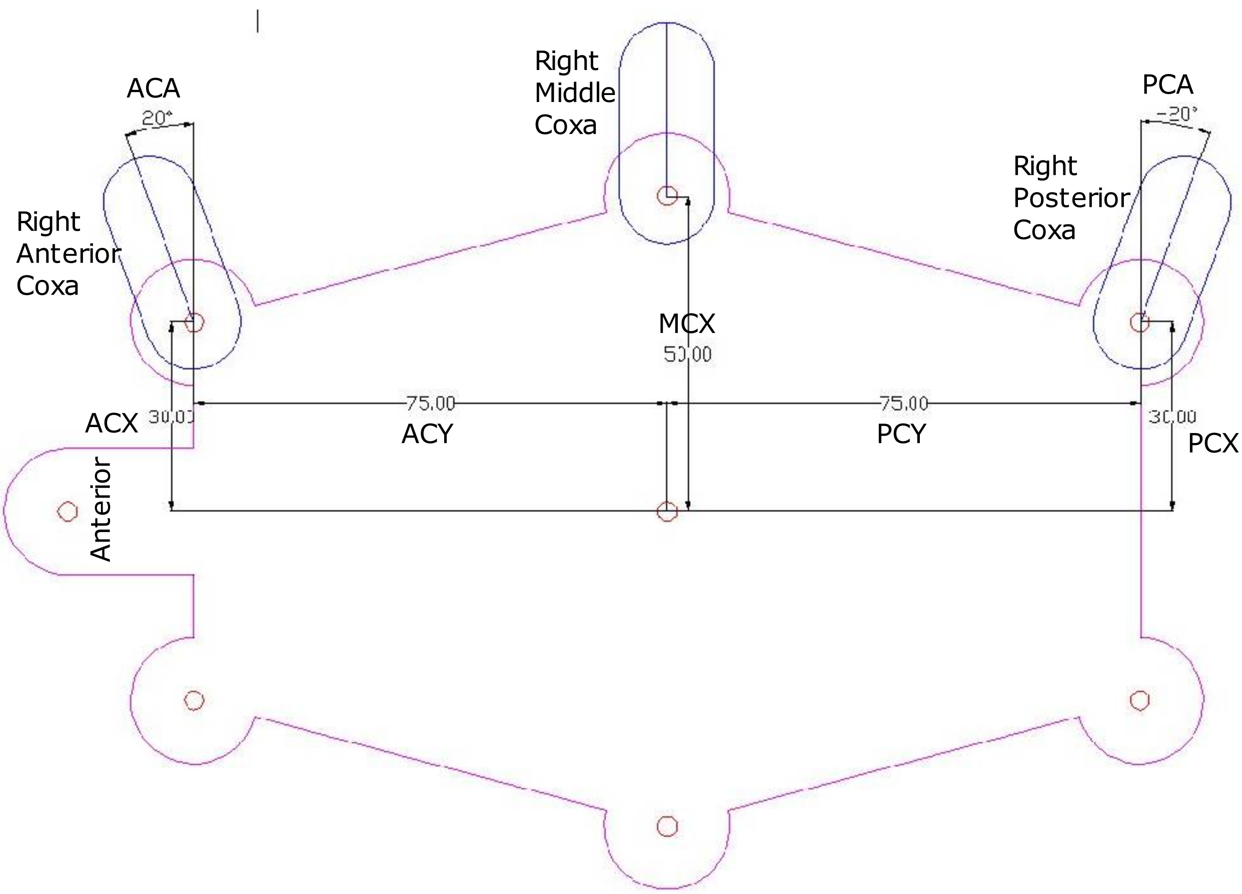
**ACA**
Anterior Coxa Angle. This is the anterior coxa angle when the hexapod is at its neutral standing position, also the centre position for coxa swing. *Note: This angle is set for the right coxa, and is automatically mirrored for the left coxa.* *(See Diagram 1)*

**MCA**
Middle Coxa Angle. This is the middle coxa angle when the hexapod is at its neutral standing position, also the centre position for coxa swing. *(See Diagram 1)*

**PCA**
Posterior Coxa Angle. This is the posterior coxa leg angle when the hexapod is at its neutral standing position, also the centre position for the coxa swing. *(See Diagram 1)*

## *Diagram 1 - Body Geometry Details*

The following diagram viewed from the top of the hexapod indicates the measuring points for the body geometry settings. *Note: MCX and MCA are set to zero and are not shown on the diagram.*

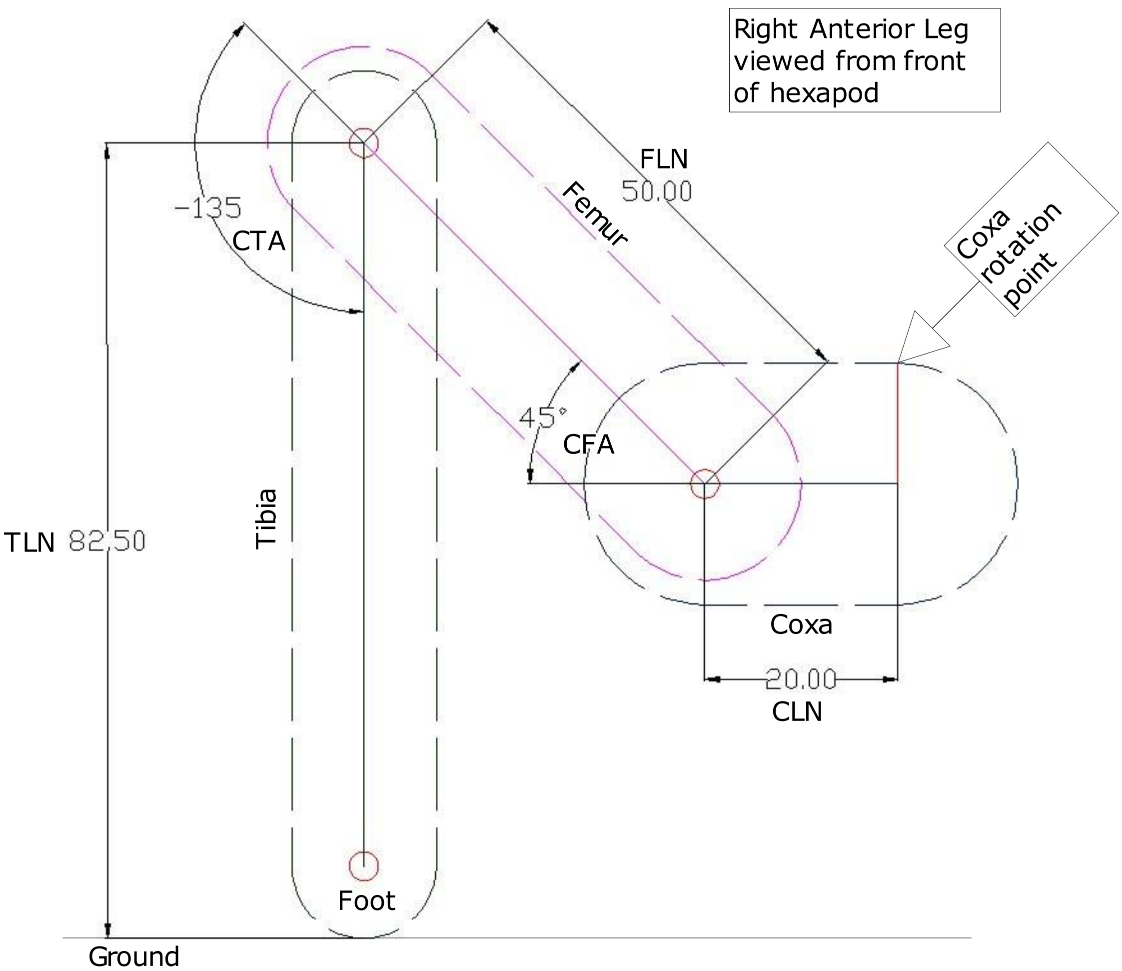


**CFA**
Centre Femur Angle. This setting defines the centre position of the femur, for example, most of my hexapods use a femur leg swing range of 0.0 to 90.0 degrees, so the CFA is set to 45.0 degrees. Essentially this is where the servo PWM signal will be at its neutral of 1500uS. (See Diagram 2)

**CTA**
Centre Tibia Angle. This setting defines the centre position of the tibia, for example, most of my hexapods use a tibia leg swing range of -90 to -180.0 degrees, so the CFA is set to -135.0 degrees. Essentially this is where the servo PWM signal will be at its neutral of 1500uS. (See Diagram 2)

## *Diagram 2 – Leg Geometry Details*

These angles may seem strange at first, this is due to the way the Inverse Kinematic model was created. If you are looking at the right anterior leg from the front, when the femur is pointing straight out from the body parallel to the ground the femur angle is 0.0 degrees. Now if the tibia is also pointing straight out from the body parallel to the ground the tibia angle is 0.0 degrees. Now if we rotate the femur 45.0 degrees clockwise and  tibia 135 degrees anticlockwise the foot should now points towards the ground. This is the position most of my hexapods use for the CFA & CTA angles.



## GAIT CONFIGURATION

**MFS**
Maximum Forward Stroke: This defines the maximum forward & backwards leg stroke the hexapod can take. This is one of the configuration settings that determines the speed of the hexapod. If in doubt start with a low figure such as 10, then try your hexapod out going forwards at full speed, depending on its size the hexapod should be taking fairly small steps approximately 10mm long. *Note: There are no checks to see if this value is within the physical range of the hexapods legs, therefore setting an unrealistic value may result in servo collisions & or IK maths errors.*

**MSS**
Maximum Side Stroke: This defines the maximum side stroke / crabbing stroke the hexapod can take. This is one of the configuration settings that determines the speed of the hexapod. If in doubt start with a low figure such as 10, then try your hexapod out going sideways at full speed, depending on its size the hexapod should be taking fairly small steps approximately 10mm long. *Note: There are no checks to see if this value is within the physical range of the hexapods legs, therefore setting an unrealistic value may result in servo collisions & or IK maths errors.*

**MTA**
Maximum Turn Angle: This defines the maximum turning speed of the hexapod. Start with a value of 10 degrees and test, high values will be clamped by the LZR setting, and may produce stange walking patterns. *Note: There are no checks to see if this value is within the physical range of the hexapods legs, therefore setting an unrealistic value may result in servo collisions & or IK maths errors.*

**LZR**
Landing Zone Radius: This value can be seen as a circle with its centre at the neutral/resting point of the foot of each leg. If this value is non zero, when a new leg step position is calculated, the foot must fall within this working radius, if it does not, the foot will be limited to the extremity of the LZR radius.

**LLH**
Leg Lift Height: This value determines the height the leg is lifted when the hexapod is taking a step. *Note: There are no checks to see if this value is within the physical range of the hexapods legs, therefore setting an unrealistic value may result in servo collisions & or IK maths errors.*

**LLO**
Leg Lift Off-Road: This value determines the height the leg is lifted when the hexapod is taking a step using the off-road gait method. *Note: There are no checks to see if this value is within the physical range of the hexapods legs, therefore setting an unrealistic value may result in servo collisions & or IK maths errors.*

**DLT**
Default Leg Transition Speed: This value sets the initial transition speed of the leg to take a step from A to B. This values is in seconds, and therefore a lower value gives a faster leg transition speed. The leg transition speed can be changed in real time using the hexapod control API. *Note: The maximum speed for the leg to move is determined by the servos being used, fast leg speed may damage some servos!*

**DGD**
Default Gait Delay: This value sets the initial gait delay. The gait delay determines the way the legs are sequenced as the hexapod moves (See Table 3). The gait delay can also be changed in real time using the hexapod control interface.

***Table 3***

| Gait Delay | Gait Style | Legs in contact with ground |
|---|---|---|
| 3 | Wave 1 | 5 |
| 2 | Wave 2 | 4 |
| 1 | Wave 3 | 4 |
| 0 | Tripod | 3 |

**ORG**
Off Road Gait: This switch enables the off road walking gait. In this mode to take a step the legs follow this sequence: protract -> move -> retract. This gives a much more robot like move but does offer better obstacle clearance. When not using the off-road gait, to take a step the legs protract and retract during the move, this gives a much faster more fluid look.

**B3D**
Full 3D balance: This option switches on the full 3D balance gesture engine. With this enabled, the hexapod body moves according to the position of the legs, which gives a more natural body movement. With this option disabled, the hexapod body is kept stable as the legs move.

**B3T**
3D balance tranlation gain: This value controls the level of body translation in 3D balance mode. Setting to 0.0 will switch off tranlation, setting to 2.0 will double the standard amount.

**B3R**
3D balance rotation gain: This value controls the level of body rotation in 3D balance mode. Setting to 0.0 will switch off rotation, setting to 2.0 will double the standard amount.

## SERVO CONFIGURATION

**RAS**
Right Anterior Servo: This value sets the individual servo reverse bits for the right anterior leg. The value is entered in the form: Coxa, Femur, Tibia (CFT), e.g. a value of 000 means none of the servos are reversed, a value of 010 means only the femur servo is reversed. (See Diagram 3)

**RMS**
Right Middle Servo: This value sets the individual servo reverse bits for the right middle leg. The value is entered in the form: Coxa, Femur, Tibia (CFT), e.g. a value of 000 means none of the servos are reversed, a value of 010 means only the femur servo is reversed. (See Diagram 3)

**RPS**
Right Posterior Servo: This value sets the individual servo reverse bits for the right posterior leg. The value is entered in the form: Coxa, Femur, Tibia (CFT), e.g. a value of 000 means none of the servos are reversed, a value of 010 means only the femur servo is reversed. (See Diagram 3)

**LPS**
Left Posterior Servo: This value sets the individual servo reverse bits for the left posterior leg. The value is entered in the form: Coxa, Femur, Tibia (CFT), e.g. a value of 000 means none of the servos are reversed, a value of 010 means only the femur servo is reversed. (See Diagram 3)

**LMS**
Left Middle Servo: This value sets the individual servo reverse bits for the left middle leg. The value is entered in the form: Coxa, Femur, Tibia (CFT), e.g. a value of 000 means none of the servos are reversed, a value of 010 means only the femur servo is reversed. (See Diagram 3)

**LAS**
Left Middle Servo: This value sets the individual servo reverse bits for the left anterior leg. The value is entered in the form: Coxa, Femur, Tibia (CFT), e.g. a value of 000 means none of the servos are reversed, a value of 010 means only the femur servo is reversed. (See Diagram 3)

**CSR**
Coxa Servo Range: This setting specified in degrees, determines the servo travel range for the coxa servos. For example, a standard PWM servo signal is from 1 to 2 milliseconds in length and would drive a standard servo 90 degrees. However, most servos differ slightly between makes and models and so this value can be trimmed to suit. If this value is not set correctly, the *HexEngine* will not produce accurate movements. *Note: See* ***Table 4*** *for a list of known servos and their appropriate range values.*

**FSR**
Femur Servo Range: This setting specified in degrees, determines the servo travel range for the femur servos. See CSR for a more detailed explanation.

**TSR**
Tibia Servo Range: This setting specified in degrees, determines the servo travel range for the femur servos. See CSR for a more detailed explanation.

### Table 4

Servo movement in degrees for PWM of 1 to 2 milliseconds.

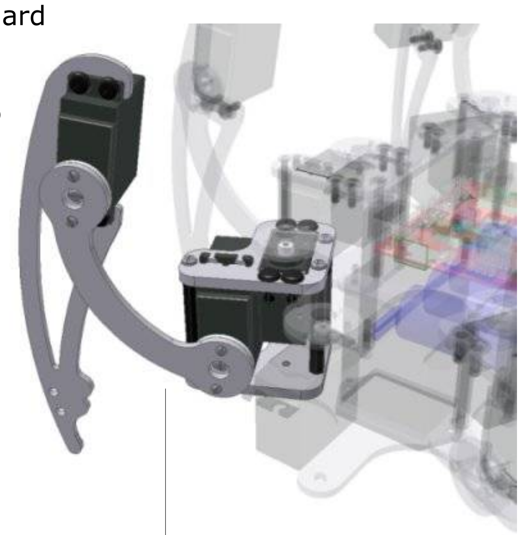
| Make | Model | Servo Range |
|---|---|---|
| Hitec | HS-475HB | 99.5 |
| Hitec | HS-645MG | 99.5 |
| Hitec | HS-225BB | 92.5 |
| Kondo | KRS788HV | 104.0 |
| | | |
| | | |
| | | |

### Diagram 3 – Servo Reverse Settings

The picture below show the right front anterior leg with the servos mounted in a standard configuration. Assuming these are Hitec servos the servo reverse setting for this leg (RAS) would be = 000. This means all the servo joints are normal travel direction. This would be the same for any type of servo which rotates in the same direction as a Hitec servo. If you were to use a servo which had reverse direction rotation, RAS would need to be 111, the exact opposite. *Note: Many of my hexapod have the posterior legs mounted in reverse to the anterior and middle legs, this is for aesthetic and sometimes mechanical reasons. In this scenario the setting for RPS & LPS would be : 111.*

The table below lists servo reverse settings for standard and reverse direction servo types. Hitec servos have been used for the purpose of defining a standard direction: Increasing the Pulse Width turns the servo horn in a clockwise direction.

| Servo Direction | Standard configuration LEG | Reverse Bit Settings |
|---|---|---|
| Standard | Right Anterior | RAS = 000 |
| Standard | Right Middle | RMS = 000 |
| Standard | Right Posterior | RPS = 000 |
| Standard | Left Posterior | LPS = 000 |
| Standard | Left Middle | LMS = 000 |
| Standard | Left Anterior | LAS = 000 |
| Reverse | Right Anterior | RAS = 111 |
| Reverse | Right Middle | RMS = 111 |
| Reverse | Right Posterior | RPS = 111 |
| Reverse | Left Posterior | LPS = 111 |
| Reverse | Left Middle | LMS = 111 |
| Reverse | Left Anterior | LAS = 111 |



Standard servo mounting & orientation for right anterior leg. Left leg configured as mirror of right.

*Note: SM1 & SM2 Settings differ for the p.Brain-SMB and p.Brain-µ24 modules, please see the relevant sections below.*

**SM1 (For p.Brain-SMB)**
Servo Remap Hexapod Standard: This setting will set the servo remap for the *p.Brain-SMB* to the standard Hexapod configuration, with the *SMB* mounted above or below the top chassis plate with the servo pins facing up. See diagram & table below for servo output configuration.

RIGHT LEG    LEFT LEG

| ***SMB* Servo Number** | **Leg / Joint** | ***SMB* Servo Number** | **Leg / Joint** |
|---|---|---|---|
| 1 | Auxiliary Servo 1 | 13 | Left Posterior Coxa |
| 2 | Auxiliary Servo 2 | 14 | Left Posterior Femur |
| 3 | Auxiliary Servo 3 | 15 | Left Posterior Tibia |
| 4 | Right Anterior Coxa | 16 | Left Middle Coxa |
| 5 | Right Anterior Femur | 17 | Left Middle Femur |
| 6 | Right Anterior Tibia | 18 | Left Middle Tibia |
| 7 | Right Middle Coxa | 19 | Left Anterior Coxa |
| 8 | Right Middle Femur | 20 | Left Anterior Femur |
| 9 | Right Middle Tibia | 21 | Left Anterior Tibia |
| 10 | Right Posterior Coxa | 22 | Auxiliary Servo 4 |
| 11 | Right Posterior Femur | 23 | Auxiliary Servo 5 |
| 12 | Right Posterior Tibia | 24 | Auxiliary Servo 6 |

**SM2 (For p.Brain-SMB)**
Servo Remap Hexapod Reversed: This setting will set the servo remap for the *p.Brain-SMB* to the inverted Hexapod configuration, with the *SMB* mounted above or below the top chassis plate with servo pins pointing down. See table below for servo output configuration.

RIGHT LEG LEFT LEG

| ***SMB* Servo Number** | **Leg / Joint** | ***SMB* Servo Number** | **Leg / Joint** |
|---|---|---|---|
| 1 | Auxiliary Servo 6 | 13 | Right Posterior Tiba |
| 2 | Auxiliary Servo 5 | 14 | Right Posterior Femur |
| 3 | Auxiliary Servo 4 | 15 | Right Posterior Coxa |
| 4 | Left Anterior Tibia | 16 | Right Middle Tibia |
| 5 | Left Anterior Femur | 17 | Right Middle Femur |
| 6 | Left Anterior Coxa | 18 | Right Middle Coxa |
| 7 | Left Middle Tibia | 19 | Right Anterior Tibia |
| 8 | Left Middle Femur | 20 | Right Anterior Femur |
| 9 | Left Middle Coxa | 21 | Right Anterior Coxa |
| 10 | Left Posterior Tibia | 22 | Auxiliary Servo 3 |
| 11 | Left Posterior Femur | 23 | Auxiliary Servo 2 |
| 12 | Left Posterior Coxa | 24 | Auxiliary Servo 1 |

**SM3 (For p.Brain-SMB & p.Brain-µ24)**
Servo Remap Standard Numbering: This setting will set the servo remap for the *p.Brain-SMB* to the standard numbering of the *SMB,* see table below.

RIGHT LEG LEFT LEG

| ***SMB* Servo Number** | **Leg / Joint** | ***SMB* Servo Number** | **Leg / Joint** |
|---|---|---|---|
| 1 | Right Anterior Coxa | 13 | Left Middle Coxa |
| 2 | Right Anterior Femur | 14 | Left Middle Femur |
| 3 | Right Anterior Tibia | 15 | Left Middle Tibia |
| 4 | Right Middle Coxa | 16 | Left Anterior Coxa |
| 5 | Right Middle Femur | 17 | Left Anterior Femur |
| 6 | Right Middle Tibia | 18 | Left Anterior Tibia |
| 7 | Right Posterior Coxa | 19 | Auxiliary Servo 1 |
| 8 | Right Posterior Femur | 20 | Auxiliary Servo 2 |
| 9 | Right Posterior Tibia | 21 | Auxiliary Servo 3 |
| 10 | Left Posterior Coxa | 22 | Auxiliary Servo 4 |
| 11 | Left Posterior Femur | 23 | Auxiliary Servo 5 |
| 12 | Left Posterior Tibia | 24 | Auxiliary Servo 6 |

### SM1 (For p.Brain-µ24)

Servo Remap Hexapod Standard: This setting will set the servo remap for the *p.Brain-µ24* to the standard Hexapod configuration, with the *µ24* mounted above or below the top chassis plate with the servo pins facing up. See diagram & table below for servo output configuration.

RIGHT LEG    LEFT LEG

| *µ24* **Servo Number** | **Leg / Joint** | *µ24* **Servo Number** | **Leg / Joint** |
|---|---|---|---|
| 1 | Auxiliary Servo 1 | 13 | Left Posterior Coxa |
| 2 | Right Anterior Coxa | 14 | Left Posterior Femur |
| 3 | Right Anterior Femur | 15 | Left Posterior Tibia |
| 4 | Right Anterior Tibia | 16 | Auxiliary Servo 4 |
| 5 | Auxiliary Servo 2 | 17 | Left Middle Coxa |
| 6 | Right Middle Coxa | 18 | Left Middle Femur |
| 7 | Right Middle Femur | 19 | Left Middle Tibia |
| 8 | Right Middle Tibia | 20 | Auxiliary Servo 5 |
| 9 | Auixiliary Servo 3 | 21 | Left Anterior Coxa |
| 10 | Right Posterior Coxa | 22 | Left Anterior Femur |
| 11 | Right Posterior Femur | 23 | Left Anterior Tibia |
| 12 | Right Posterior Tibia | 24 | Auxiliary Servo 6 |

**SM2 (For p.Brain-µ24)**
Servo Remap Hexapod Reversed: This setting will set the servo remap for the *p.Brain-µ24* to the inverted Hexapod configuration, with the *µ24* mounted above or below the top chassis plate with servo pins pointing down. See table below for servo output configuration.

RIGHT LEG LEFT LEG

| ***µ24* Servo Number** | **Leg / Joint** | ***µ24* Servo Number** | **Leg / Joint** |
|---|---|---|---|
| 1 | Auxiliary Servo 6 | 13 | Right Posterior Tiba |
| 2 | Left Anterior Tibia | 14 | Right Posterior Femur |
| 3 | Left Anterior Femur | 15 | Right Posterior Coxa |
| 4 | Left Anterior Coxa | 16 | Auxiliary Servo 3 |
| 5 | Auxiliary Servo 5 | 17 | Right Middle Tibia |
| 6 | Left Middle Tibia | 18 | Right Middle Femur |
| 7 | Left Middle Femur | 19 | Right Middle Coxa |
| 8 | Left Middle Coxa | 20 | Auxiliary Servo 2 |
| 9 | Auxiliary Servo 4 | 21 | Right Anterior Tibia |
| 10 | Left Posterior Tibia | 22 | Right Anterior Femur |
| 11 | Left Posterior Femur | 23 | Right Anterior Coxa |
| 12 | Left Posterior Coxa | 24 | Auxiliary Servo 1 |

**Pan/Tilt Head Servo Map**

| *p.Brain-SMB Servo Number* | *p.Brain-µ24 Servo Number* | *Pan/Tilt function* |
|---|---|---|
| Auxiliary 1 | Auxiliary 1 | Head Pan |
| Auxiliary 2 | Auxiliary 6 | Head Tilt |
| Auxiliary 3 | Auxiliary 2 | Head Twist |
| Auxiliary 4 | Auxiliary 5 | Head Gripper |

## BEHAVIORAL CONFIGURATION

**DPU**
Default Power Up Height: This is the neutral standing height for the hexapod. This value is used in conjunction with GFA and GTA to determine the neutral position of the hexapod. This value should be a positive, greater than DPD and greater than the distance from the centre of the coxa spline to the base of the hexapod.

**DPU**
Default Power Down Height: This is the resting position for the legs when the hexapod powers down. This value should be positive and greater than or equal to the distance from the centre of the coxa spline to the base of the hexapod.

**APD**
Auto Power Down: When enabled, the hexapod will automatically power down to the rest position if no control commands have been received within 30 seconds.

**SLT**
Sleep Time: If auto power down 'APD' is enabled, this variable determines how many seconds pass since the last locomotion command before the hexapod sleeps.

**RET**
Relax Time: This variable determines how many seconds pass since the last locomotion command before the hexapod returns all legs to the neutral standing position.

**SST**
Stop Time: This variable determines how many seconds pass since the last locomotion command before the hexapod stops moving. e.g. If a walk forward command is sent, the hexapod will automatically stop after 'SST' seconds if no further commands are sent. Useful for loss of communications!

**RTR**
Reset Translate & Rotate: This switch enables the automatic reset of body translation and rotation to 0, when a PIP movement command is sent. For example, if the hexapod is stood still, and the body rotated in X by 10 degrees, as soon as the hexapod receives a PIP move command, the body rotation and translation are rest to 0 which will level out the body before walking.

**SRC**
Servo Range Check: This option enables or disables the servo range check, if SRC is enabled (=1) and a servo reaches its minimum or maximum PWM value, the hexapod engine issues an error message and all signals to the servos are stopped. A wake command needs to be sent to restart the HexEngine.

**ASF**
Auxiliary Servo Filter: This value sets the amount of filtering to apply to auxiliary servo position data samed via the PIP command 'A'. Setting this value to zero, will switch off any filtering and will result in fast instant moves which can cause damage to servos if not used wisely!

**ADC**
External Port Function Configure: This string value configures the p.Brain-ds24 external port pins ( CN19 on p.Brain-SMB ) to either Digital I/O or Analogue capture. The string must be 8 characters long and can contain only 'A' or 'D' characters. For example to configure the top nibble as analogue and the bottom nibble as digital, enter the following: ADC=AAAADDDD

**DIO**
External Port I/O Configure: This string value configures the p.Brain-ds24 extrenal port pin direction ( CN19 on p.Brain-SMB ). The pin must be configured as digital using the "ADC" command in order to operate as digital I/O. The string must be 8 characters long and can contain only 'I' or 'O' characters. For example, if ADC=DDDDDDDD, to configure the top nibble as inputs and the bottom nibble as outputs, enter the following: DIO=IIIIOOOO. *Note, Digital inputs 0 to 5 have weak internal pull-up's, inputs 6 and 7 do not. There are no output current limiting resistors on the digital ports, take care not to short or overload the pins, maximum current source and sink per pin is 4 mA.*

**PIP**
Packet Interface Protocol Mode: This setting defines the PIP mode, either 0 or 1. Please see the HexEngine PIP Guide for further details on the PIP and PIP modes.

**TXD**
Transmission Delay: When controlling the HexEngine from micro processors that do not have hardware uarts such as the basic stamp series, it is necessary to add a transmission delay to packets in order to give the connected micro time to switch to data reception after a PIP transmission. This is only necessary in two way communications such as reading from the HexEngine I2C port. The TXD parameter adds a number of 0xff bytes to the header of each PIP packet.

**PS2**
PS2 Gamepad Control: Setting this value to 1 will enable the internal PS2 gamepad control module. See PS2 Gamepad Control for further details.

## BODY MOTION CONFIGURATION

**MXR**
Maximum X Rotation: This defines the maximum body X rotation in degrees. When controlling the hexapod using the control interface, binary values of 0 to 255 are scaled between -MXR and +MXR degrees.

**MYR**
Maximum Y Rotation: This defines the maximum body Y rotation in degrees. When controlling the hexapod using the control interface, binary values of 0 to 255 are scaled between -MYR and +MYR degrees.

**MZR**
Maximum Z Rotation: This defines the maximum body Z rotation in degrees. When controlling the hexapod using the control interface, binary values of 0 to 255 are scaled between -MZR and +MZR degrees.

**MXT**
Maximum X Translation: This defines the maximum body X translation in millimetres. When controlling the hexapod using the control interface, binary values of 0 to 255 are scaled between -MXT and +MXT mm.

**MYT**
Maximum Y Translation: This defines the maximum body Y translation in millimetres. When controlling the hexapod using the control interface, binary values of 0 to 255 are scaled between -MYT and +MYT mm.

**MZT**
Maximum Z Translation: This defines the maximum body Z translation in millimetres. When controlling the hexapod using the control interface, binary values of 0 to 255 are scaled between -MZT and +MZT mm.

**TRF**
Translation & Rotation Filter: This value sets the amount of filtering to apply to body translations & rotations. Setting this value to zero, will switch off any filtering and will result in fast instant moves which can cause damage to servos if not used wisely!

**PTR**
Post-Translate body: When this is set, the body translation happens after the body rotation, so if the head is pinting towards the ground, and the body is translated forward the body will translate along the new rotated axis. If this bit is clear (0), the body will translate along the original ground axis and is not effected by the body rotation.

## Head Configuration.

The following settings control the head pan, tilt, twist and grip servo limits. These servos are internally mapped to auxiliary servos, see page 25 for details. Head limits are used by the PIP command 'H' and the PS2 game pad control interface.

**PAM**
PAN Servo Middle: This value sets the pan servo middle position in micro seconds. Therefore for the standard servo centre position this would be 1500.

**PA+**
PAN Servo Maximum: This value sets the pan servo maximum position in micro seconds. Therefore for the standard servo range 45° would be approximately 2000.

**PA-**
PAN Servo Minimum: This value sets the pan servo minimum position in micro seconds. Therefore for the standard servo range -45° would be approximately 1000.

**TIM**
TILT Servo Middle: This value sets the tilt servo middle position in micro seconds. Therefore for the standard servo centre position this would be 1500.

**TI+**
TILT Servo Maximum: This value sets the tilt servo maximum position in micro seconds. Therefore for the standard servo range 45° would be approximately 2000.

**TI-**
TILT Servo Minimum: This value sets the tilt servo minimum position in micro seconds. Therefore for the standard servo range -45° would be approximately 1000.

**TWM**
TWIST Servo Middle: This value sets the tilt servo middle position in micro seconds. Therefore for the standard servo centre position this would be 1500.

**TW+**
TWIST Servo Maximum: This value sets the tilt servo maximum position in micro seconds. Therefore for the standard servo range 45° would be approximately 2000.

**TW-**
TWIST Servo Minimum: This value sets the tilt servo minimum position in micro seconds. Therefore for the standard servo range -45° would be approximately 1000.

**GRM**
GRIPPER Servo Middle: This value sets the tilt servo middle position in micro seconds. Therefore for the standard servo centre position this would be 1500.

**GR+**
GRIPPER Servo Maximum: This value sets the tilt servo maximum position in micro seconds. Therefore for the standard servo range 45° would be approximately 2000.

**GR-**
GRIPPER Servo Minimum: This value sets the tilt servo minimum position in micro seconds. Therefore for the standard servo range -45° would be approximately 1000.

**LPT**
Load Pan/Tilt: When LPT is set to 1, the pan and tilt head mid positions stored in PAM & TIM respectively are re-loaded when the hexapod wakes up, effectively resetting the head position to neutral.

**HSI**
This is the individual servo reverse bits for the three head servos in the order MSB to LSB, Pan, Tilt, Twist. Setting the associated bit to 1, will invert the servo direction, e.g. HSI=001 Would have the Twist servo inverted.

**AHL**
Auto Head Level, with this bit set to 1, the auto head levelling feature is enabled, now when the body is rotated, the head will counter rotate in an appropriate direction to keep the head level at its current position. If the head rotates in the same direction as the body, invert the associated servo using the HSI setting.

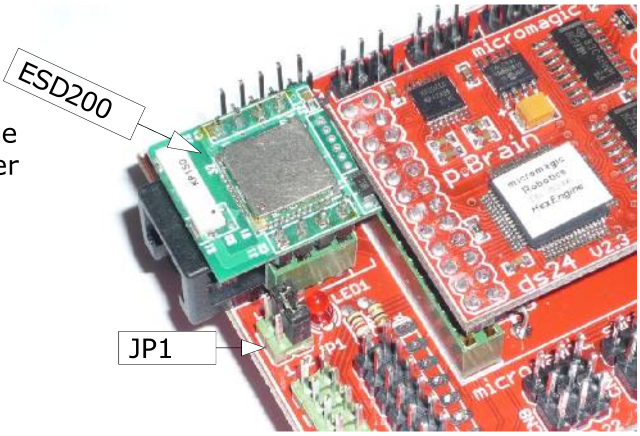
**PTR**
Post Translate Body. Usually the body translation maths happens before the body rotation functions, therefore a translation would continue along the new rotated axis. With PTR enabled, the body translation function happens after the body rotation function, this causes the body to translate along the original axis. If in doubt, give it a go and see the difference.

## ESD200 Bluetooth configuration – p.Brain-SMB

This section explains the *HexEngine* "ESD CONFIG" command. This command attempts to configure the ESD200 or ESD100 ready to work with the *HexEngine*. *Note: Have a copy of the p.Brain-SMB to hand to help installation and jumper identification.*

### ESD Installation

To install your ESD200, remove power from your *p.Brain-SMB*, connect the ESD200 to the *SMB* as indicated in the picture. Install jumper JP1-2, Have JP1-1 ready but not installed. JP1-1 connects the ESD200 transmitter to UART1 receiver on the p.Brain-ds24.




### ESD Configuration

Switch on power to your *p.Brain-SMB*, and HexEngine. Go to the "CONFIG" screen and type "ESD CONFIG"

The first prompt you will see is to reset the ESD200 to factory defaults, this is achieved by shorting JP1-1 out for a couple of seconds, and then removing the JP1-1 link. Once this has been done continue with the configuration by typing "y".

You should see a series of program operations which configure the device. After each parameter is set, a response of "OK" should appear in red on the right of the screen confirming configuration. All but the last commend "ATZ" should receive an "OK" response, if any of the responses are not "OK", try the configuration procedure again. *Note: The baud rate setting for the ESD200 is determined by the HexEngine "CBR" parameter.*

The bluetooth pass key or pin code is: 1138 *This is only true for version 1.2 and higher of the HexEngine, for versions below 1.2, please see the relative documentation.*

Once configuration is complete, your ESD200 should now be in discover mode, this means that you should be able to pair with your ESD200 using Windows bluetooth utility or similar. During the pairing procedure you will be asked for the pass key defined above. When pairing is complete, you should be able to connect you your ESD200 with the COM port assigned by your host operating system.

### LED1

The LED next to the ESD200 (LED1) on the p.Brain-SMB board is the connection LED. This should illuminate when a connection is made between the device and a host system. *Note I have noticed in some cases the LED does not light even when a connection is established, this seems to be determined by the host software initiating the connection!?*

**ESD Manual Configurations**
The ESD200 device can be manually configured using a suitable RS-232 interface and terminal software. For further information on connecting the ESD to a serial port, please see the ESD200 user guide.

Following a factory reset, the "ESD CONFIG" command runs the following terminal commands on the ESD device: *Note: All commands are followed with a carriage return.*

| Command | Description | Reply |
|---|---|---|
| AT | See if device is connected | OK |
| AT+BTKEY=1138 | Set the PIN code | OK |
| AT+BTSEC,1,0 | Pin Authentication on Encryption off | OK |
| ATS3=1 | Fast Operation on | OK |
| AT+BTMODE,3 | Set Discovery mode on | OK |
| ATS37=8000 | Set link lost time out = 5 seconds | OK |
| AT+UARTCONFIG,x,N,1,0 | Configure Uart, x = BAUD rate e.g. 9600 | OK |
| ATZ | Soft reset | OK |

## ESD200 Bluetooth configuration – p.Brain-µ24

### ESD Installation

To install your ESD200, remove power from your *p.Brain-µ24*, connect the ESD200 to the *SMB* as indicated in the picture. Install jumper JP1 in the B position.


### ESD Configuration

Before you configure the ESD200 device, you may need to change the desired baud rate, this defaults to 115200. If you require a different baud rate, you will need to use the RJ-232 adaptor and configure the HexEngine TBR setting before proceeding. *Note, the baud rate into the ESD200 does not have to match the baud rate on the host PC bluetooth adaptor.*

Although not essential, if you have an RJ-232 adaptor lead, it is advised to plug this in during ESD200 configuration so that the progress and any configuration errors can be monitored. Connect the RJ-232 lead from the RJ11 port to a free serial port on your host PC, and start up your terminal software.

With the JP1 jumper in the 'B' position, switch on power to the p.Brain-µ24, wait until the unit has booted up, (approx 2 seconds), then press and hold the small tactile switch located under the power terminal CN3 for at least 2 seconds, then release. All going well your ESD200 will be configured in approximately 5 seconds. *If you are using the RJ-232 lead, you should see a configuration start and configuration OK message in your terminal screen.*

The bluetooth pass key or pin code is: 1138 *This is only true for version 1.2 and higher of the HexEngine, for versions below 1.2, please see the relative documentation.*

Once configuration is complete, your ESD200 should now be in discover mode, this means that you should be able to pair with your ESD200 using Windows bluetooth utility or similar. During the pairing procedure you will be asked for the pass key defined above. When pairing is complete, you should be able to connect you your ESD200 with the COM port assigned by your host operating system.

**LED1**
The blue LED next to the ESD200 is the connection LED. This should illuminate when a connection is made between the device and a host system. *Note I have noticed in some cases the LED does not light even when a connection is established, this seems to be determined by the host software initiating the connection!?*

## PS2 Gamepad Control

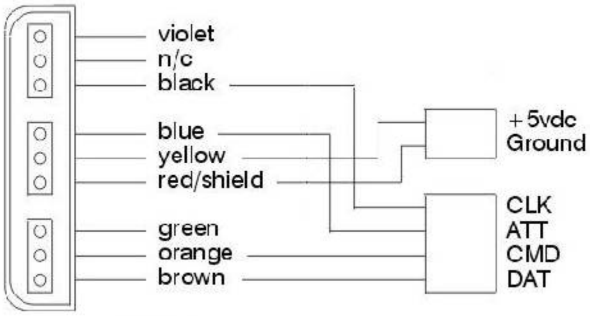
The HexEngine can have a Playstation PS2 style controller directly connected to the 8 channel I/O port. This gives a convenient way to make your hexapod into a remote controlled robot without the need for an intermediate processor.

**Connection to the PS2 controller:**

| PS2 Gamepad Signal | p.Brain-ds24 | p.Brain-SMB | p.Brain-µ24 |
|---|---|---|---|
| CLK | DIG4 (CN1 - 17) | CN19 - 15 | CN2 - 2 |
| ATT | DIG5 (CN1 - 18) | CN19 - 13 | CN2 - 4 |
| CMD | DIG6 (CN1 - 19) | CN19 - 11 | CN2 - 6 |
| DAT | DIG7 (CN1 - 20) | CN19 - 9 | CN2 - 8 |
| GROUND | N/A | CN19 - 16 | CN2 - 3 |
| +5V | N/A | CN17 – 9<br>(For 5V Power)<br>CN17 – 8<br>(For 3.3V Power) ** | CN2 – 5<br>(for 3.3V Power) ** |

*** Note: All wired & wireless PS2 gamepads we have tested work fine 3.3V, but not all will work at 5V, therefore we suggest using 3.3V from CN17 to power the gamepad. If this does not work try the 5V output on CN17.*

**PS2 Connection Cable**
To connect a PS2 gamepad to the p.Brain-SMB or p.Brain-µ24, you will need the female end of a PS2 extension cable. The cable needs to be about 8 inches long, and requires the following 6 connections:



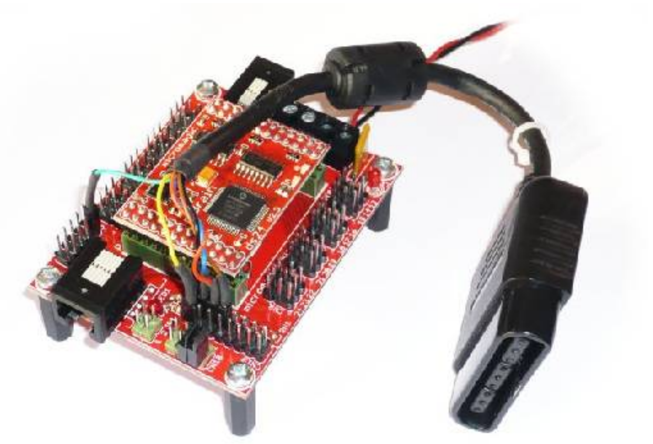

**Note!** Extension cable wire colours vary, so its best to use a continuity tester to determine the correct connections.

Once connected to the SMB/µ24, switch on the PS2 control mode within the HexEngine configuration screen by typing PS2=1, exit and save the configuration. Enter the “PS2 TEST” screen to check that your PS2 controller is connected, this screen should display the button and joystick of the controller.

**PS2 Control Functionality**

| PS2 Button | Function |
|---|---|
| Start | Power up / power down the hexapod |
| Triangle | Select tripod gait |
| Circle | Select ripple gait 3 |
| Cross | Select ripple gait 2 |
| Square | Select ripple gait 1 |
| Select + Triangle | Increase leg transition speed |
| Select + Cross | Decrease leg transition speed |
| Select + Circle | Toggle full 3D balance gesture |
| Select + Square | Toggle Off-Road gait |
| Select + R1 | Reset the hexapod legs to home position |
| Select + R2 | Reset leg transition speed to configuration default |
| Left Thumb Up | Walk Forward |
| Left Thumb Down | Walk Backwards |
| Left Thumb Left | Turn Left |
| Left Thumb Right | Turn Right |
| Right 1 (Held Down) | Enable proportional walking:<br>Forward/Backward = Right Stick Up/Down<br>Turn = Right Stick Left/Right<br>Crab = Left Stick Left/Right |
| Right 2 (Held Down) | Enable proportioanl body rotate/translate<br>X Rotate = Right Stick Up/Down<br>Y Rotate = Left Stick Left/Right<br>Z Rotate = Right Stick Left/Right<br>While holding (Left 2)<br>X Translate = Right Stick Left/Right<br>Y Translate = Right Stick Up/Down<br>Z Translate = Left Stick Up/Down |
| Left 1 ( Held Down ) | Head Pan = Left Stick Left/Right<br>Head Tilt = Left Stick Up/Down |

Image of PS2 cable connected to HexEngine
CN19 & CN17 *(Using 5V configuration)*



# Legal

Please read fully before purchasing any merchandise from micromagic systems ltd.

## PRODUCTS

In no event shall micromagic systems be liable for any claim for incidental, consequential damages, or any injuries sustained due to the use of or improper use of products and / or kits purchased out of or in connection thereof with the manufacture, sale, delivery or use of any product in this catalogue or web site. All micromagic systems products purchased should NOT be used for medical, life-saving, life-support, or any applications that could cause injury, dangerous / hazardous situations or consequential damages resulting from the use of the mechanical, hardware or software products sold or represented by micromagic systems.

All products sold by micromagic systems are for Self Learning Experiences, and for Safe Entertainment.

## Prices and Specifications

### Note

Product specifications, prices listed and availability of items in our web site and in our printed catalogue are subject to change without notice. All prices shown in our web site and in our printed catalogue are believed accurate at time of publication, but subject to change without any notice. We will always advise you of the new price increase and seek your approval before processing any orders you place.

## General Product Terms and Conditions

Call micromagic systems before returning any items.

### PRODUCT RETURNS POLICY

All sales are final.
New products are warranted for 30 days. Any return for repair or replacement must be pre-authorized by micromagic systems and under no circumstance will returns be accepted unless so authorized. Return Products under warranty must be pre-approved by MMS and sent via certified mail, prepaid and insured, for your protection. (Please note we cannot refund shipping fees). All electronic kit sales are final. Due to the fact that components of the kit may be damaged during assembly we do not accept returns or refunds on any electronic kits

If you receive damaged merchandise, you must contact micromagic systems within 2 days of receipt of your original order. Specify clearly the reason for your refusal. We will exchange returned merchandise for same new merchandise, or for the item sterling amount within 7 days once we receive the returned damaged items from you. Proof of mailing is advised, as we cannot be held responsible for loss of the returned merchandise in mail transit. All return postage is non-refundable. The merchandise, including packing and wrapping material, being returned should be in the same condition as when you received them. Please contact us via e-mail at **matt@micromagicsysstems.com**. Defective merchandise will be replaced (No cash will be refunded). We reserve the right to refuse to replace any merchandise, which our micromagic systems technicians determine to be damaged by the user, or through inappropriate use of that merchandise.

### WARRANTY POLICY

We guarantee all products except electronic kits to be free of defects in workmanship and material for 30 days from the purchase, delivery date. We will repair or replace non-electronic kits (No cash will be refunded), at our option providing there is no evidence of customer misuse or alteration to that product item.

micromagic systems carries a limited 30 day warranty on most all items, some items carry an additional number of warranty days or special restrictions. If you want specific warranty information about a product contact micromagic systems to obtain that information.

We are not able to offer any refunds or accept returns for the following items and products: Electronic Kits.

### CANCELLATION POLICY

Please be aware that if you cancel an order you may be responsible for restocking fees and / or shipping charges, including charges for return shipping. Cancelled orders are subject to a 25% or £10.00 minimum restocking fee. Orders cancelled within 24 hours of order placement will not be subject to restocking fees however this does not apply to orders with Express Shipping and Handling.