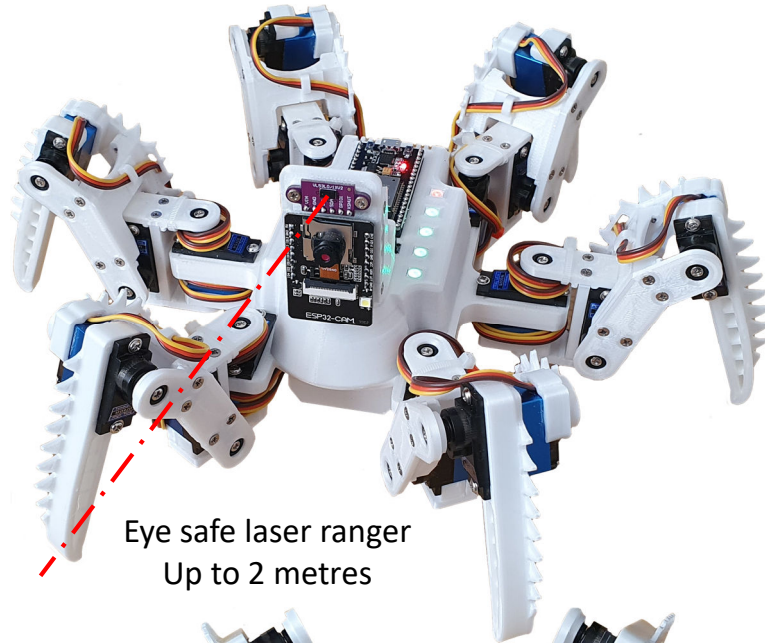


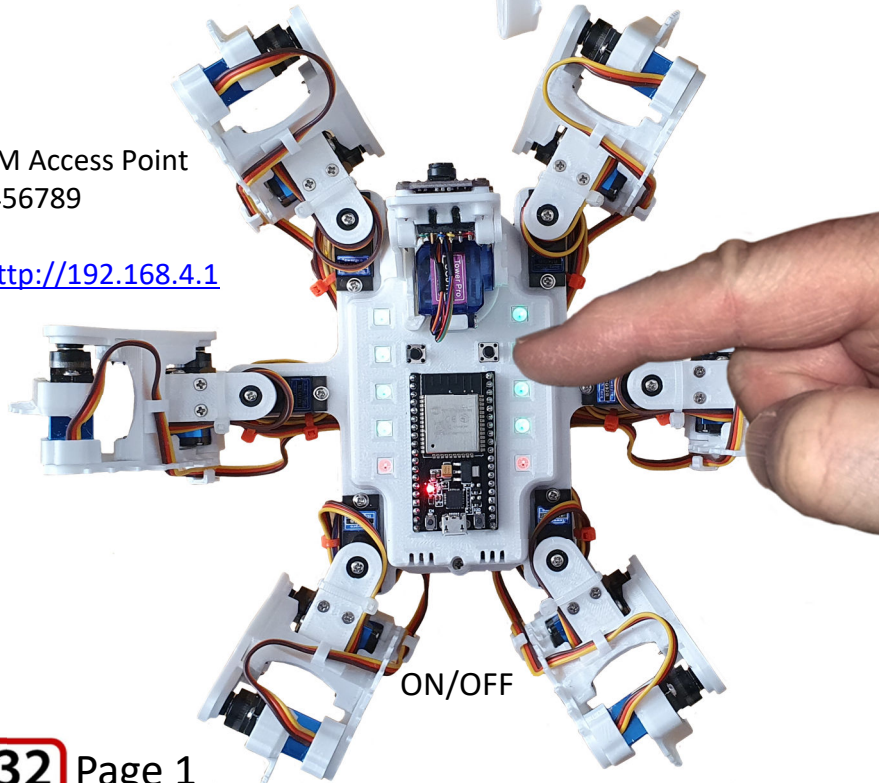
# SpidaBot – Button Functions



Eye safe laser ranger  
Up to 2 metres

Connect using:  
SSID: ESP32-CAM Access Point  
Password: 123456789

Browser URL: <http://192.168.4.1>



ON/OFF

## Tech:

- ESP32 microcontroller
- 20 x Servo motors
- VL53L0X TOF laser range finder
- 2.4GHz ESP-NOW wireless control
- 10 RGB LEDs – battery monitoring/motion
- 2 x 3.7v 3000mAh batteries
- 3-D printed construction

## Features:

- Safe start, with LED blink sequence.
- A button press will cause it to stand up, in ranging mode, and respond to further button presses.
- 1 - LEDs display laser ranging
- 2 - backs away from approaching objects
  - - returns to start point after 5 seconds
- 3 - target tracking at a fixed 20 cm distance
- Controller 'live' recording, and replay action
- Battery Low sensing with cut-off.
- ESP32 RESET button returns it to safe mode.

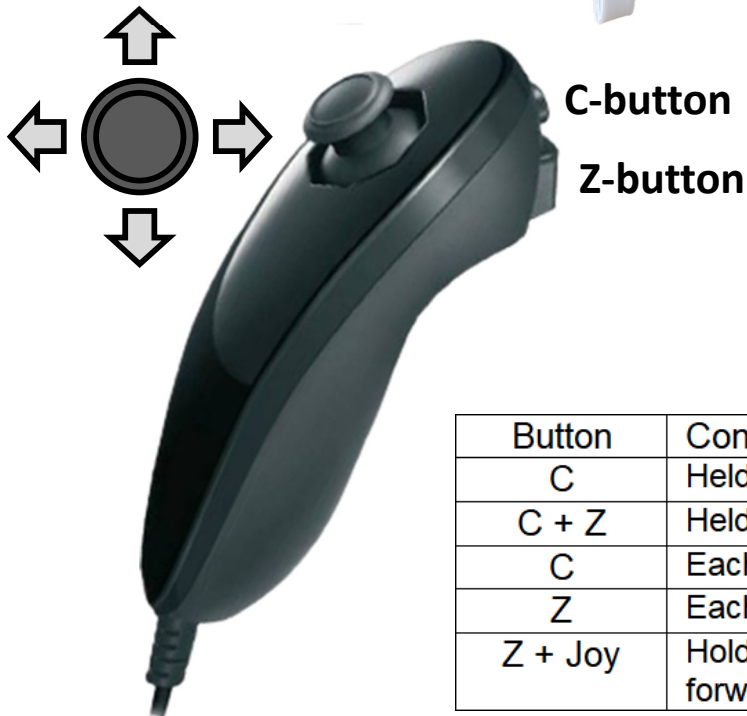
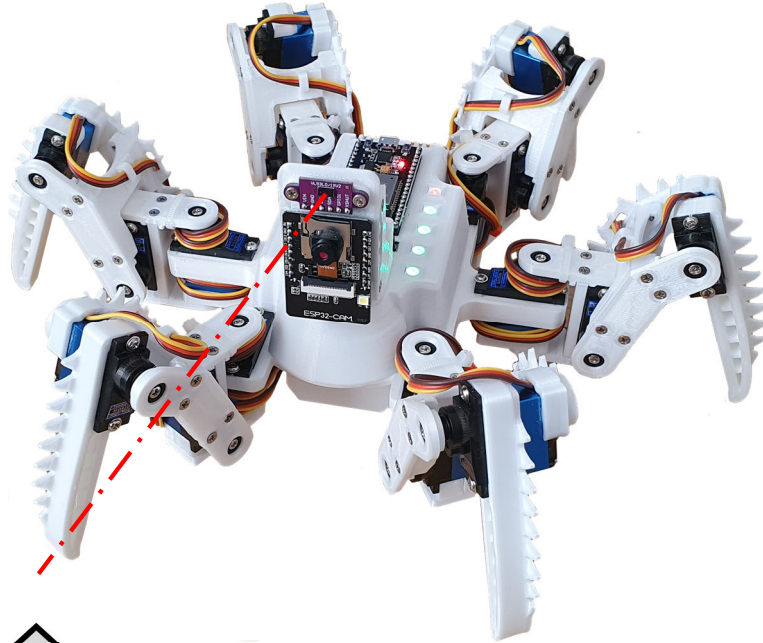
## Enhancements:

- Scope within coding.

## In Use:

- Legs will initially pull in when power is applied.
- Switch power OFF when not in use.

# SpidaBot – RC Demo Functions



## Tech:

- ESP32 microcontroller
- 20 x Servo motors
- VL53L01X TOF laser range finder
- 2.4GHz wireless control
- 10 RGB LEDs – battery monitoring/motion
- 2 x 3.7v 3000mAh batteries
- 3-D printed construction

## Features:

- Safe start, legs moves, with LED indicators.
- Controlled via Wii Nunchuk, or Classics.
- Walks and turns in both directions.
- Performs pre-set moves – bow and wave.
- Variable speed motion via joystick.
- Monitor+ display modes.
- Battery Low sensing with cut-off.
- Data can be returned via the wireless link.

## Enhancements:

- None.

Button	Conditions and responses
C	Held initially for > 1 second to make the robot stand and become 'active'.
C + Z	Held for > 2 seconds will return the robot to an 'inactive' safe state.
C	Each press will increase the responsiveness of the robot from 1 – 5 (max).
Z	Each press will decrease the responsiveness of the robot from 5 – 1 (min).
Z + Joy	Holding Z will change the right/left walking modes from turning to walking sideways, forwards will become a 'bow' and reverse will become a 'hello wave'.

# SpidaBot – Wii Classic Pro Functions

- L** HELD, + **START** initiates RECORD mode.
- L** pressed stops active RECORD mode.
- START** Plays a recording.
- ANY** Stops a playing recording.

- R** HELD, connects WiFi
- R+Z<sub>R</sub>** HELD disconnects WiFi
- When connected:
- R** Increases 'Gear' value
- Z<sub>R</sub>** Decreases 'Gear' value
- Select + R** Gear Max mode
- Select + Z<sub>R</sub>** Gear Min mode

- ^** Rise up, when standing
- <** Camera LED decrease/off
- >** Camera LED on/increase
- v** Lower down, when raised

- X** Not defined
- Y** Not defined
- a** Sit back and wave, demo move.
- b** Forward bow, demo move.

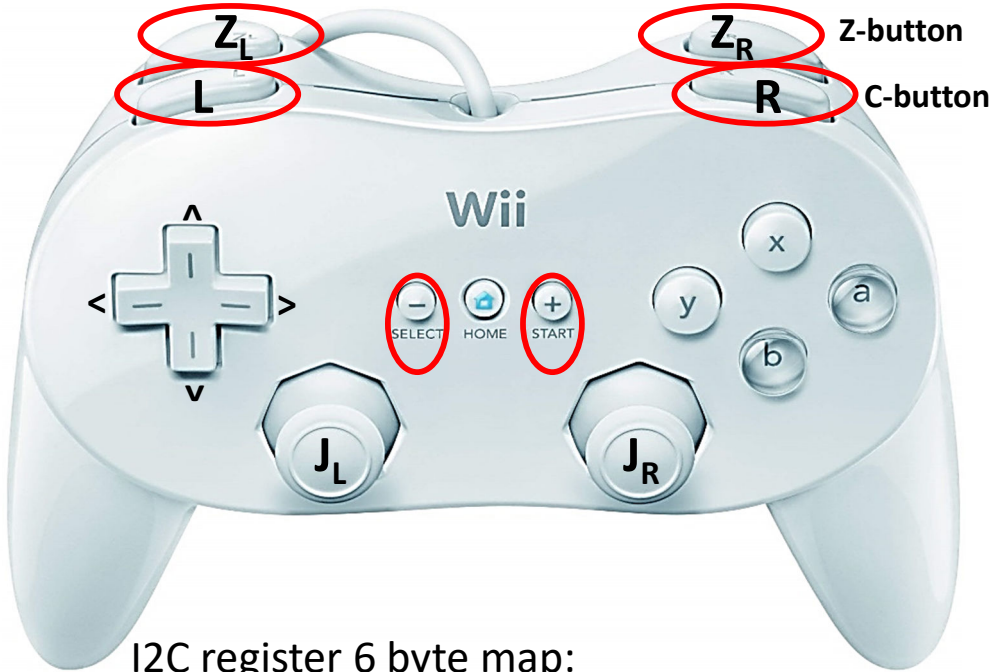


Z-button }  
C-button } Like Wii Nunchuk

**J<sub>L</sub>** Controls Camera Pan and Tilt.  
X-axis controls pan right/left.  
Y-axis controls tilt forwards/back.

**J<sub>R</sub>** Controls SpidaBot in normal walking mode.  
X-axis controls steering and turning.  
Y-axis controls forward/reverse drive.  
**Z<sub>L</sub>** Held in for sideways walking & steer.  
**Z<sub>R</sub>** Held in for pitch and yaw movements.

# SpidaBot – Wii Classic Pro Functions



I2C register 6 byte map:

Bit								
7	6	5	4	3	2	1	0	
RX<4:3>					LX<5:0>			RxWiFi[0]
RX<2:1>					LY<5:0>			RxWiFi[1]
RX<0>	LT<4:3>				RY<4:0>			RxWiFi[2]
LT<2:0>					RT<4:0>			RxWiFi[3]
BDR	BDD	BLT	B-	BH	B+	BRT	1	RxWiFi[4]
BZL	BB	BY	BA	BX	BZR	BDL	BDU	RxWiFi[5]

LX,LY are the left Analog Stick X and Y (0-63), RX and RY are the right Analog Stick X and Y (0-31), and LT and RT are the Left and Right Triggers (0-31). The left Analog Stick has twice the precision of the right Stick.

BD{L,R,U,D} are the D-Pad direction buttons. B{ZR,ZL,A,B,X,Y,+,H,-} are the discrete buttons. BL{LT,RT} are the digital button click of LT and RT. All buttons are 0 when pressed.

## Wii Classic Pro functions:

SpidaBot system default is Wi-Fi RC disconnected.

- **R** – held in, will connect SpidaBot over ESP-NOW Wi-Fi.
- **R + Z<sub>R</sub>** – held in, will disconnect RC over Wi-Fi.
  - short press will move robot towards REST.

When connected:

- **R** – increased Gear value, if in full control.
- **Z<sub>R</sub>** – decreased Gear value, if in full control.
- **a** – demonstrates a sit and wave movement.
- **b** – demonstrates a bow movement.
- **B- + R** - gives **Full** control, speeds 1 – 5.
- **B- + L** - gives **Min** control, speed 1.
- **J<sub>R</sub>** – controls walking actions, forward, back & turn.
- **J<sub>L</sub>** – controls camera pan and tilt.
- Digital keypad:
  - **^** - raises the height, when standing or walking.
  - **v** – lowers the height, when standing or walking.
  - **>** - turns on/increases camera LED brightness.
  - **<** - turns off/decreases camera LED brightness.
- **Z<sub>L</sub>** – held in, J<sub>R</sub> joystick functions to walk sideways.
- **Z<sub>R</sub>** – held in, J<sub>R</sub> joystick functions change to pitch and yaw.

# SpidaBot – Wii Classic Functions

- Z<sub>L</sub>** HELD, + **START** initiates RECORD mode.
- Z<sub>L</sub>** pressed stops active RECORD mode.
- START** Plays a recording.
- ANY** Stops a playing recording.

- R** HELD, connects WiFi
- R+Z<sub>R</sub>** HELD disconnects WiFi
- When connected:
- R** Increases 'Gear' value
- Z<sub>R</sub>** Decreases 'Gear' value
- Select + R** Gear Max mode
- Select + Z<sub>R</sub>** Gear Min mode

**Z-button** (L, Z<sub>L</sub>, Z<sub>R</sub>, R)

**C-button** (R)

- ^** Rise up, when standing
- <** Camera LED decrease/off
- >** Camera LED on/increase
- v** Lower down, when raised

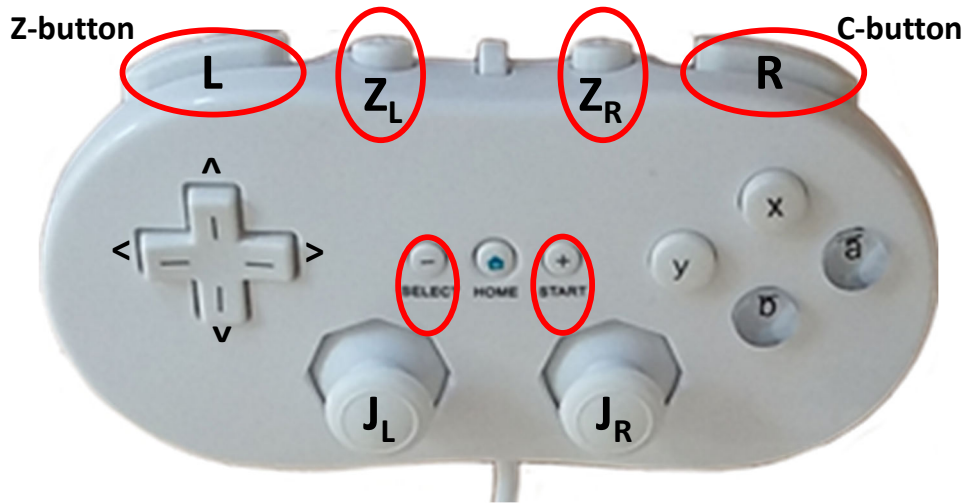
- X** Not defined
- Y** Not defined
- a** Sit back and wave, demo move.
- b** Forward bow, demo move.

**J<sub>L</sub>** Controls Camera Pan and Tilt.  
X-axis controls pan right/left.  
Y-axis controls tilt forwards/back.

**J<sub>R</sub>** Controls SpidaBot in normal walking mode.  
X-axis controls steering and turning.  
Y-axis controls forward/reverse drive.

- Z<sub>L</sub>** Held in for sideways walking & steer.
- Z<sub>R</sub>** Held in for pitch and yaw movements.

# SpidaBot – Wii Classic Functions



I2C register 6 byte map:

Bit								
7	6	5	4	3	2	1	0	
RX<4:3>				LX<5:0>				RxWiFi[0]
RX<2:1>				LY<5:0>				RxWiFi[1]
RX<0>	LT<4:3>			RY<4:0>				RxWiFi[2]
LT<2:0>				RT<4:0>				RxWiFi[3]
BDR	BDD	BLT	B-	BH	B+	BRT	1	RxWiFi[4]
BZL	BB	BY	BA	BX	BZR	BDL	BDU	RxWiFi[5]

LX,LY are the left Analog Stick X and Y (0-63), RX and RY are the right Analog Stick X and Y (0-31), and LT and RT are the Left and Right Triggers (0-31). The left Analog Stick has twice the precision of the right Stick.

BD{L,R,U,D} are the D-Pad direction buttons. B{ZR,ZL,A,B,X,Y,+,-,H,-} are the discrete buttons. BL{LT,RT} are the digital button click of LT and RT. All buttons are 0 when pressed.

## Wii Classic functions:

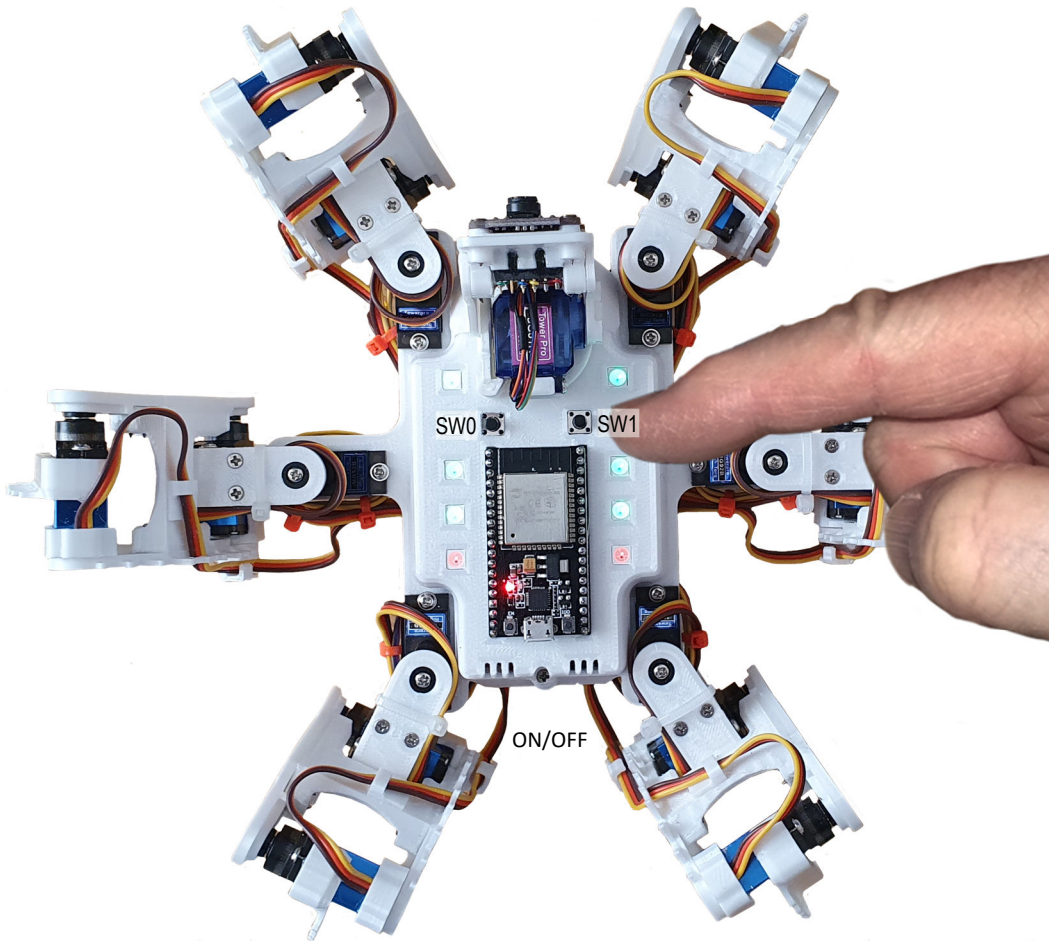
SpidaBot system default is Wi-Fi RC disconnected.

- **R** – held in, will connect SpidaBot over ESP-NOW Wi-Fi.
- **L + R** – held in, will disconnect RC over Wi-Fi.  
- momentarily will move robot towards REST.

When connected:

- **R** – increased Gear value, if in full control.
- **L** – decreased Gear value, if in full control.
- **a** – demonstrates a sit and wave movement.
- **b** – demonstrates a bow movement.
- **B- + R** - gives **Full Control**, speeds 1 – 5.
- **B- + L** - gives **Min Control**, speed 1.
- **J<sub>R</sub>** – controls walking actions, forward, back & turn.
- **J<sub>L</sub>** – controls camera pan and tilt.
- Digital keypad:
  - **^** - raises the height, when standing or walking.
  - **v** – lowers the height, when standing or walking.
  - **>** - turns on/increases camera LED brightness.
  - **<** - turns off/decreases camera LED brightness.
- **Z<sub>L</sub>** – held in, **J<sub>R</sub>** joystick functions to walk sideways.
- **Z<sub>R</sub>** – held in, **J<sub>R</sub>** joystick functions change to pitch and yaw.

# SpidaBot – TEST mode



## TEST Tech:

These special test modes should only be performed with SpidaBot on its stand.

Holding down either SW0 or SW1 button switches, whilst resetting the micro, will take the code into TEST mode, in which all servo values will be set initially to their respective 'Cal' values, which will pull in the legs to their rest positions, wait for the legs to do this, and the LED battery colour will be blue.

### SW0 functions:

- Briefly pressing SW0 toggles the TEST mode display options, which are different to normal mode.
- If SW0 is held down the legs will go into a movement demo mode in which they swing through their angular limits.
- If SW0 is held down again the legs will go into a walking demo mode in which they walk slowly forward.
- If SW0 is held down again the legs will return to their default, rest positions.
- Repeating the SW0 actions will toggle between the respective modes.

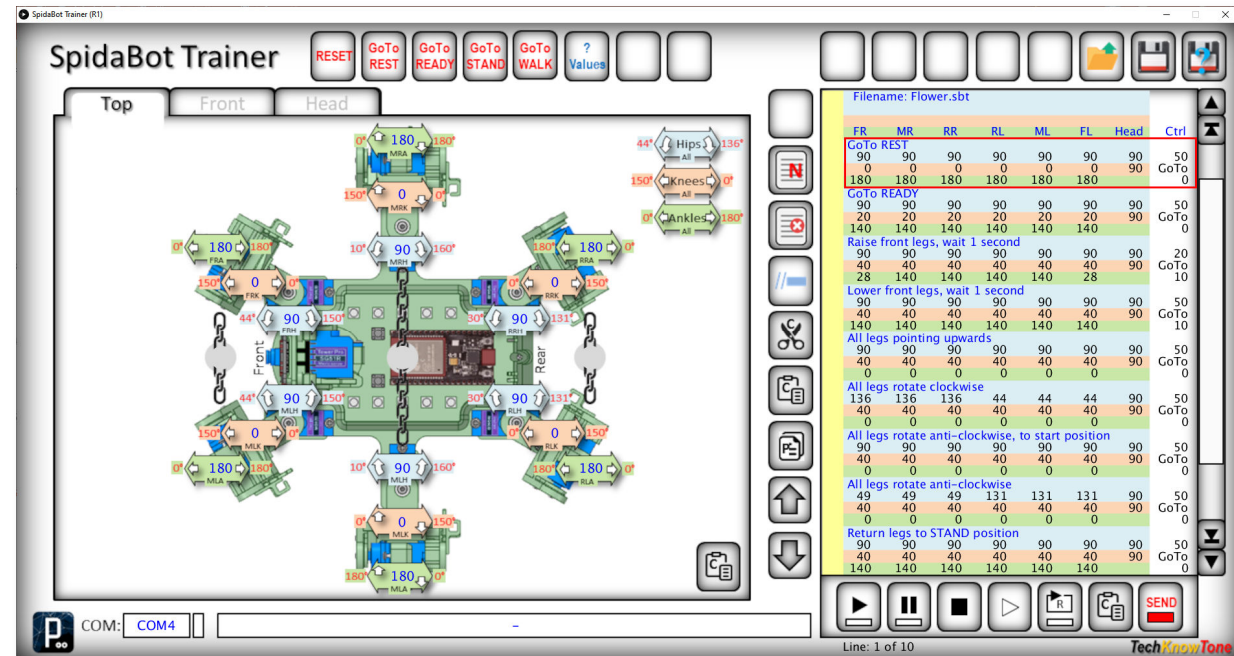
### SW1 functions:

- Briefly pressing SW1 toggles the laser range finder ON/OFF, changes the LED mode to suit, and sets the display monitor to ranging.

# SpidaBot – Trainer app

The SpidaBot Trainer app was written to enable the creation of a sequence of servo movements, legs and head. It presents three views of the robot, which help you select a particular servo for adjustment. You can click on chain link buttons to group the movement of servos, in which case they all move together as the mouse is clicked and dragged on one of them. The keyboard SHIFT key also controls the way in which they move.

The servos are grouped by colour, as **Hips**, **Knees** and **Ankles**; and these colours appear again in the right-hand list window. As you change values in the view, the corresponding values will change in the lists current line; as indicated by the red rectangle around it. You can use the editing buttons to add new lines, delete, copy, etc. As you move the mouse over them, look at the help field for information on them.



**List Window:** The top line acts as a header for the list, showing the filename and the servo references, it can not be edited. On other lines, you can include comments for each line, as reminders of what the movement does in that line. The right-hand column defines the number of steps a servo will take, when moving to the target position, the type of movement, and a pause (in 100ms units) taken before the next movement will be actions.

**PLAY buttons:** Below the list is a group of buttons. These work when the SpidaBot is connected and the SEND button is active. Note that servo movements only occur when SEND is active, as otherwise values are not sent to the SpidaBot. You can 'PLAY' your list of movements from top to bottom, or bottom to top, depending on which mouse button you click on PLAY. Or from any point in the list. The process normally stops when the last or first line is reached, but if the REPEAT button is active, then the sequence is restarted from the appropriate end. You can also STOP, and single step through the list, in a forwards and backwards direction. The Trainer app controls the overall process by sending the individual movements to the SpidaBot, and in turn, it responds with a prompt, to indicate that the movement has been implemented. When you copy a list of lines to the Windows clipboard, each line will look like this, and they can be pasted directly into your .ino code, to run as a sequence:

```
// GoTo REST
//          FLH,FLK,FLA,MLH,MLK,MLA,RLH,RLK,RLA,RRH,RRK,RRA,MRH,MRK,MRA,FRH,FRK,FRA,Steps
GoToAngles( 90, 0,180, 90, 0,180, 90, 0,180, 90, 0,180, 90, 0,180, 90, 0,180, 90, 0,180, 50);
//          HP, HT,Stp,Stp
HeadToAngle( 90, 90, 50, 50);
```