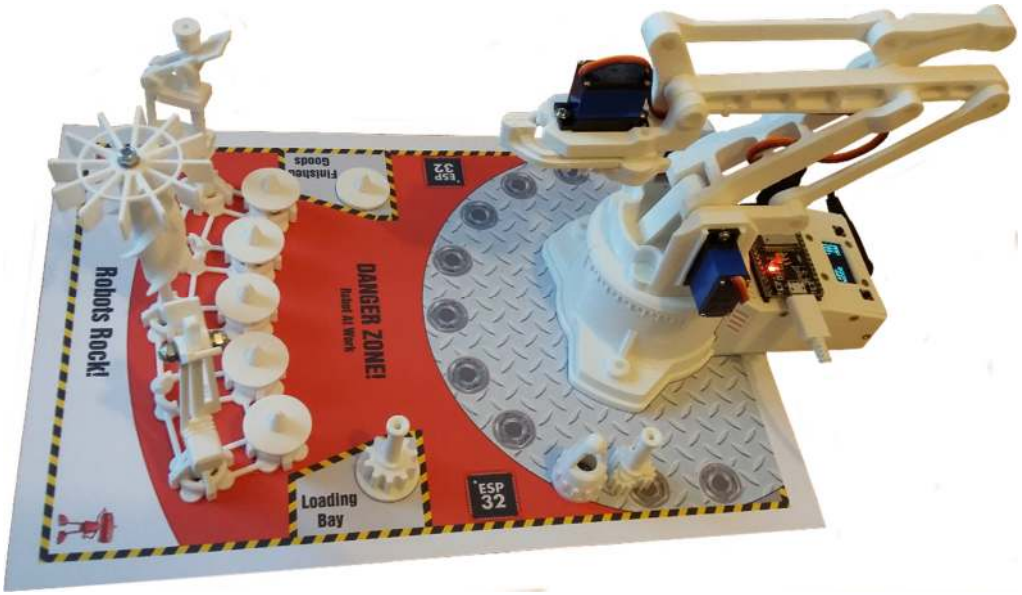
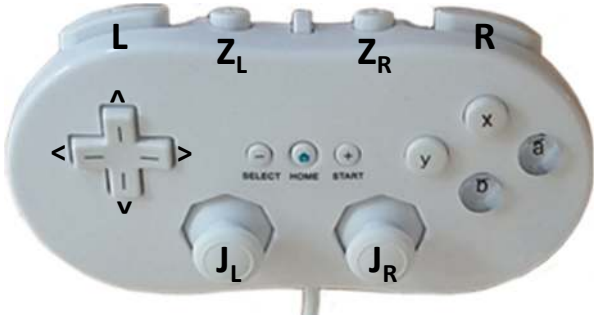
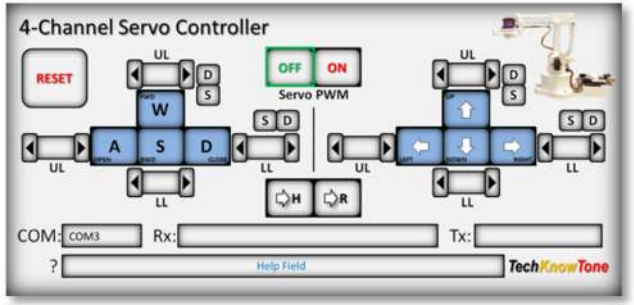
Reach Robot Mk1 – Demo Functions





Note:
If no controller is connected, a long press on SW0 will take the robot into 'demo' mode. There after any button press will take it to the rest position.



Tech:

- ESP32 microcontroller
- 4 x Servo motors
- I2C wired Wii Classic controller
- 7.4v batteries / adapter
- 3-D printed construction

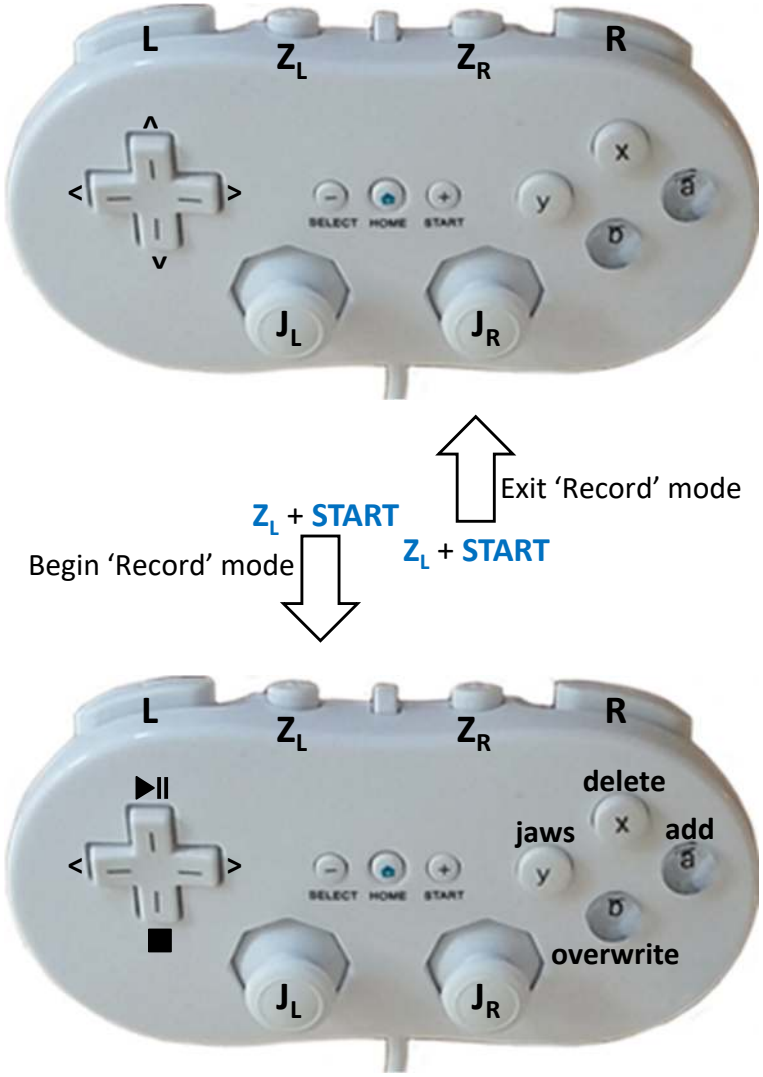
Features:

- Safe start, move to rest
- Controlled via Wii controller
- Turns, reaches and opens jaws
- Performs pre-set moves
- XYZ programmed via Windows app

Wii Controller (Normal Mode):

- **HOME** – moves to ready position
- **L** + **HOME** – performs main demo 2
- J_L – joystick X moves jaws, Y moves reach
- J_R – joystick X rotates L/R, Y moves up/down
- **L** or **R** – halves joystick demands
- **SELECT** – moves to reset/rest position
- **L** + **SELECT** – performs main demo 1
- **START** – moves to floor position
- **L** + **START** – performs main demo 3
- **y** – toggles step/continuous print servo values
- Z_L or Z_R – quarters joystick demands
- Z_L + **START** – toggle Record/Normal modes
- **<** or **^** or **>** or **v** – perform reach moves L/C/R/D

Reach Robot Mk1 – Wii Classic Functions

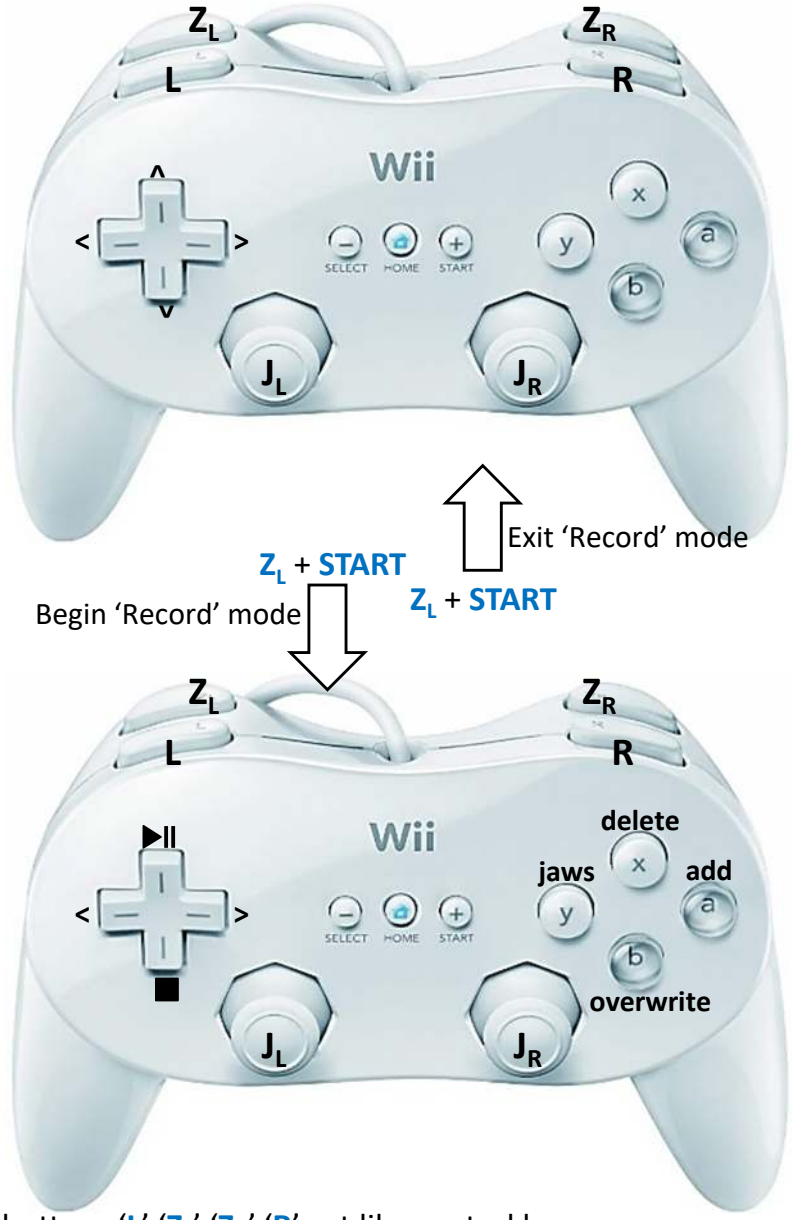


Note:
Front buttons '**L**' 'Z_L' 'Z_R' '**R**' act like control keys

Wii Controller (Record Mode):

▶II	- play/pause the recorded move sequence - (held) play cycle repeatedly until stopped
■	- stop 'play' and go to 'start'
<	- move to previous recorded position, cycle
>	- move to next recorded position, cycle
SELECT	- moves to reset/rest position, overwrites current
HOME	- moves to ready position , overwrites current
START	- moves to floor position , overwrites current
a	- add a new point in the sequence, as current
b	- insert/increment a delay, in 100ms steps - if held down auto-increment up to 9.9 seconds
x	- snapshot current position in memory
y	- close/open jaws 'n' times (assumes open to start)
J_L	- X open/closes jaws, Y moves reach fwd/bckwd
J_R	- X rotates L/R, Y moves up/down
L + (< or >)	- insert a slow down/speed up command
L + **x**	- recall, overwrite and move to the snapshot point
L + **y**	- decrement stored clap count (9 – 0)
L + **b**	- decrement delay, in 100ms steps to zero
L + **a**	- appends the current position as last point
L + **R**	- export recorded move sequence to serial port
(**L** or **R**) + J_n	- hold **L** or **R** to half joystick demands
Z_L + **START**	- RECORD mode enter/exit toggle, Hold to clear
Z_L + **a**	- delete current point
(Z_L or Z_R) + J_n	- hold **L** or **R** to quarter any joystick demands

Reach Robot Mk1 – Wii Classic Pro Functions

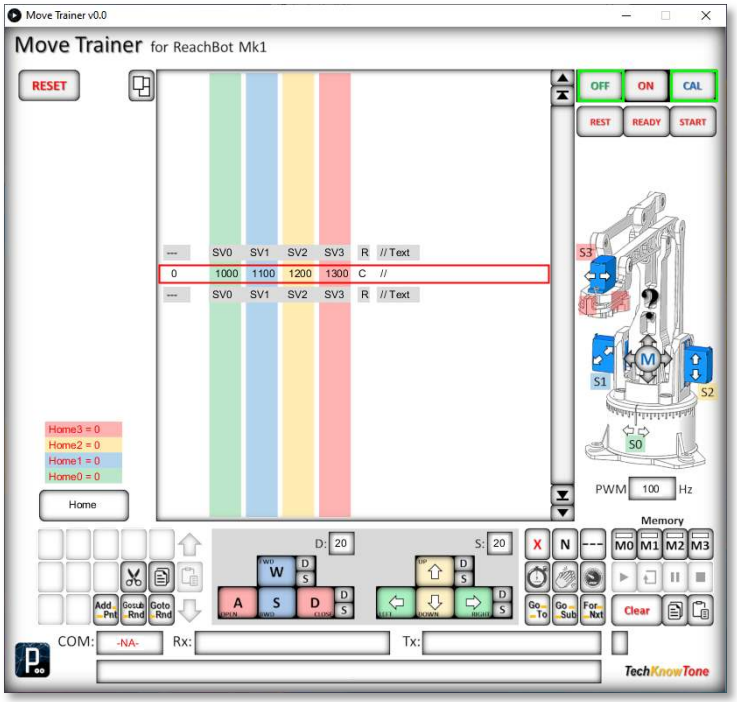


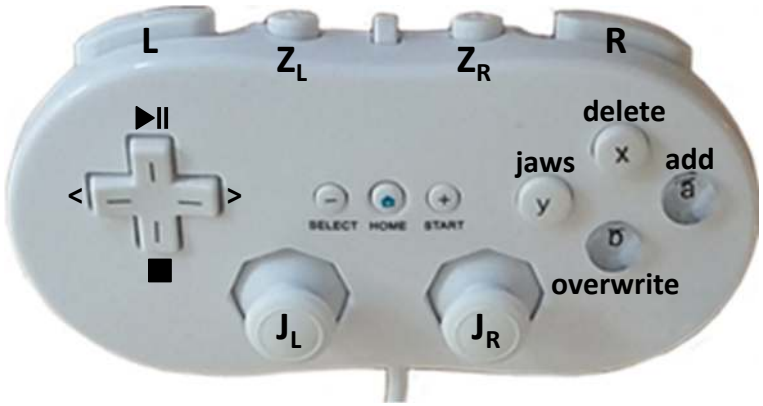
Note:
Front buttons '**L**' 'Z_L' 'Z_R' '**R**' act like control keys

Wii Controller (Record Mode):

▶II	- play/pause the recorded move sequence - (held) play cycle repeatedly until stopped
■	- stop 'play' and go to 'start'
<	- move to previous recorded position, cycle
>	- move to next recorded position, cycle
SELECT	- moves to reset/rest position, overwrites current
HOME	- moves to ready position , overwrites current
START	- moves to floor position , overwrites current
a	- add a new point in the sequence, as current
b	- insert/increment a delay, in 100ms steps - if held down auto-increment up to 9.9 seconds
x	- snapshot current position in memory
y	- close/open jaws 'n' times (assumes open to start)
J_L	- X open/closes jaws, Y moves reach fwd/bckwd
J_R	- X rotates L/R, Y moves up/down
L + (< or >)	- insert a slow down/speed up command
L + **x**	- recall, overwrite and move to the snapshot point
L + **y**	- decrement stored clap count (9 – 0)
L + **b**	- decrement delay, in 100ms steps to zero
L + **a**	- appends the current position as last point
L + **R**	- export recorded move sequence to serial port
(**L** or **R**) + J_n	- hold **L** or **R** to half joystick demands
Z_L + **START**	- RECORD mode enter/exit toggle, Hold to clear
Z_L + **a**	- delete current point
(Z_L or Z_R) + J_n	- hold **L** or **R** to quarter any joystick demands

Reach Robot Mk1 – Windows App





Note:
Front buttons '**L**' 'Z_L' 'Z_R' '**R**' act like control keys

Wii Controller (App Mode):

▶II	- play/pause the recorded move sequence - (held) play cycle repeatedly until stopped
■	- stop 'play' and go to 'start'
<	- move to previous recorded position, cycle
>	- move to next recorded position, cycle
SELECT	- moves to reset/rest position, overwrites current
HOME	- moves to ready position , overwrites current
START	- moves to floor position , overwrites current
a	- add a new point in the sequence, as current
b	- insert/increment a delay, in 100ms steps - if held down auto-increment up to 9.9 seconds
x	- snapshot current position in memory
y	- close/open jaws 'n' times (assumes open to start)
J_L	- X open/closes jaws, Y moves reach fwd/bckwd
J_R	- X rotates L/R, Y moves up/down
L + **x**	- recall, overwrite and move to the snapshot point
L + **y**	- decrement stored clap count (9 – 0)
L + **b**	- decrement delay, in 100ms steps to zero
L + **a**	- appends the current position as last point
L + **R**	- export recorded move sequence to serial port
(**L** or **R**) + J_n	- hold **L** or **R** to half joystick demands
Z_L + **START**	- RECORD mode enter/exit toggle , Hold to clear
Z_L + **a**	- delete current point
(Z_L or Z_R) + J_n	- hold **L** or **R** to quarter any joystick demands

Note: connecting the robot to the Windows App prevents the robot from entering local Record Mode, as the app extends the functionality of the stored data, which can be copied and used in robot C++ code.

Reach Robot Mk1 – Windows App Functions

insert a ‘Blank’ line as a separator or extended text line

insert a pause delay, from 100ms to 9.9 seconds.

insert a jaw clap sequence from 1 to 10 claps.

insert a speed adjustment from 10% to 300% of normal speed.

insert a go to line command, to branch to a specific line.

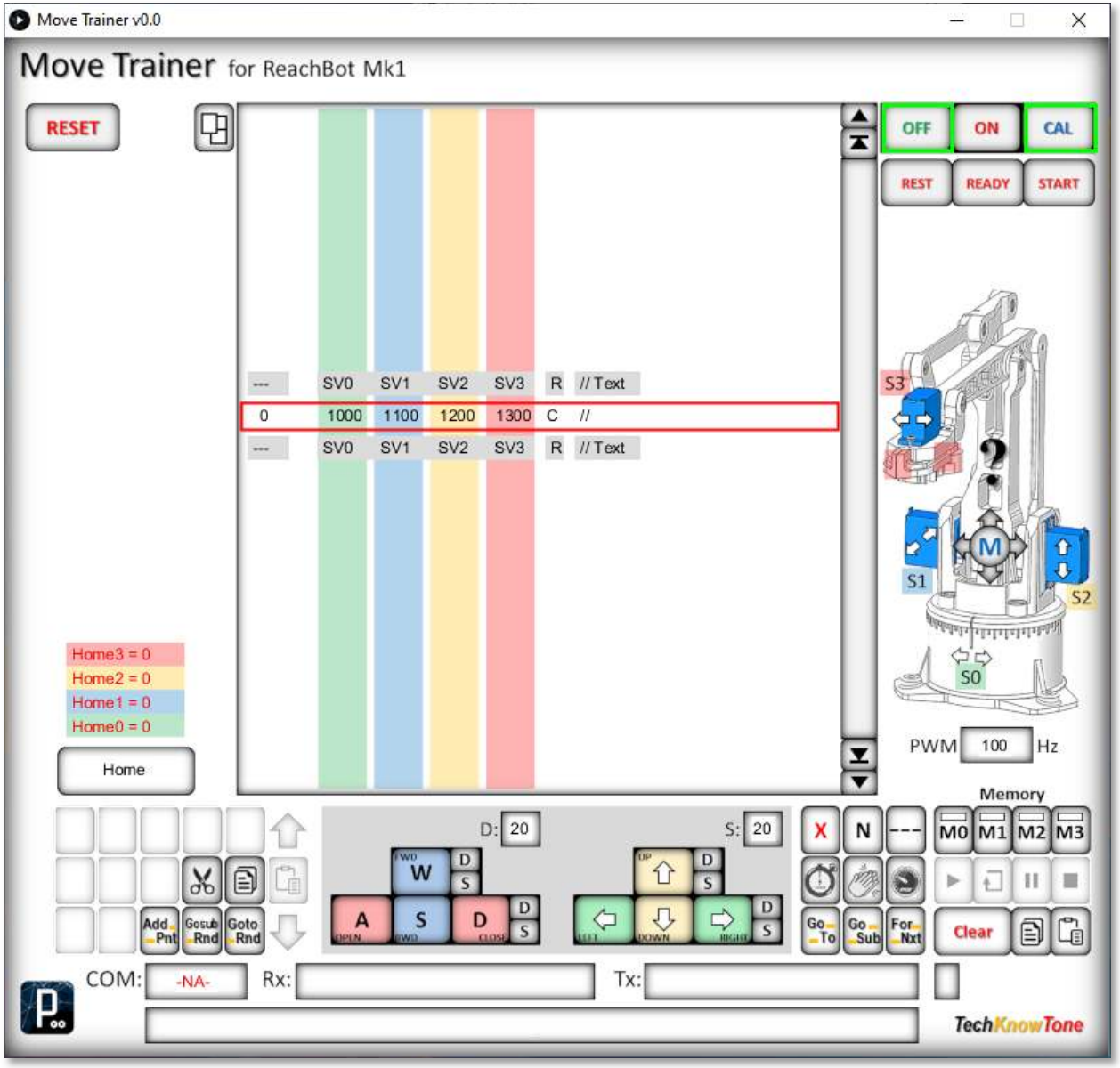
insert a go to subroutine line, from which you can return later.

insert a For… Next loop, and specify the number of loops.

insert an AddPnt[] command, to pre-load branches

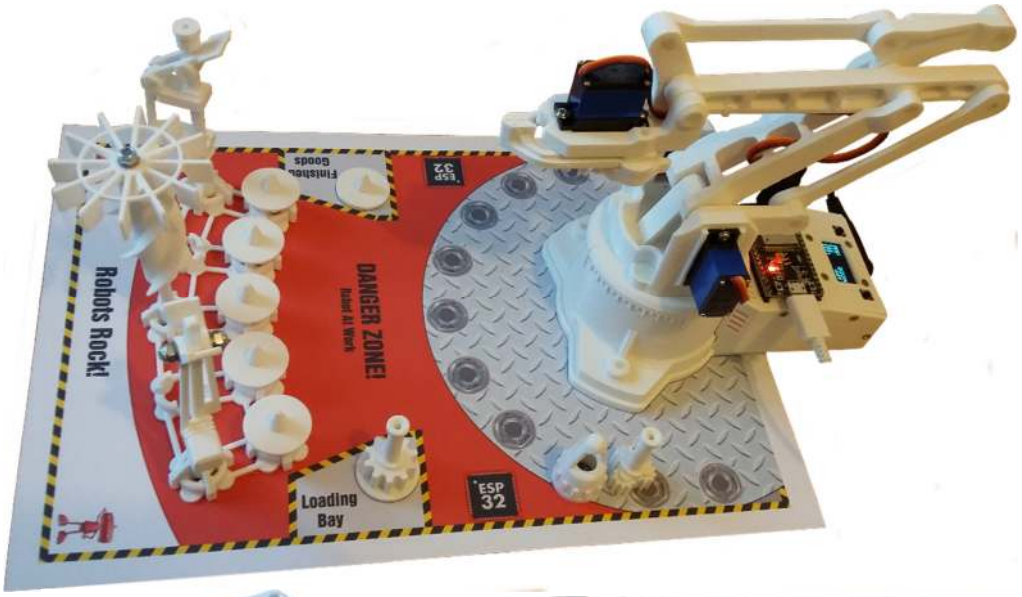
insert a random Gosub command using AddPnt[] vectors

insert a random Goto command using AddPnt[] vectors



Note: connecting the robot to the Windows App prevents the robot from entering local Record Mode, as the app extends the functionality of the stored data, which can be copied and used in robot C++ code.

Reach Robot Mk1 – WiFi Mode



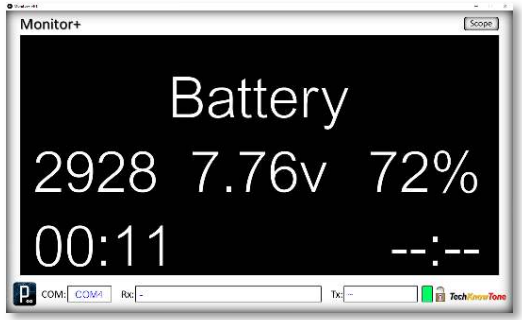
Wi-Fi:

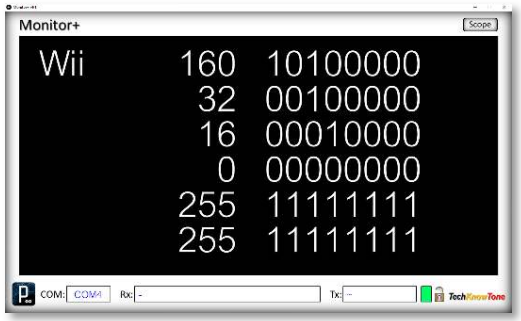
To enter Wi-Fi mode:

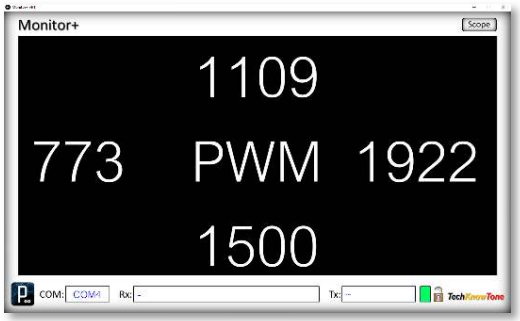
- Hold down the left button SW0
- Press and release the micros RESET button
- Release the SW0 button switch

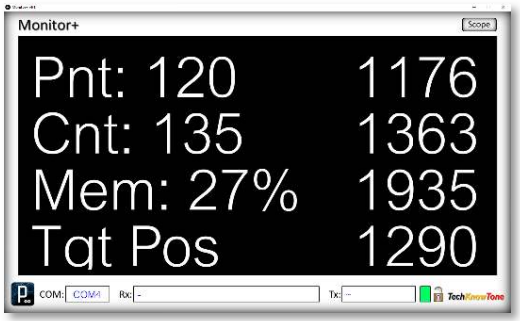
Purpose:

The purpose of Wi-Fi mode is to enable the OLED display contents to be seen on a computer screen or large monitor. You will nee a Wii Wi-Fi transceiver (see projects) and an app called Monitor+. The app connects to the Wii transceiver via the USB serial port, using a cable from the PC to the transceiver. This is a great way of showing others your robots built-in features.

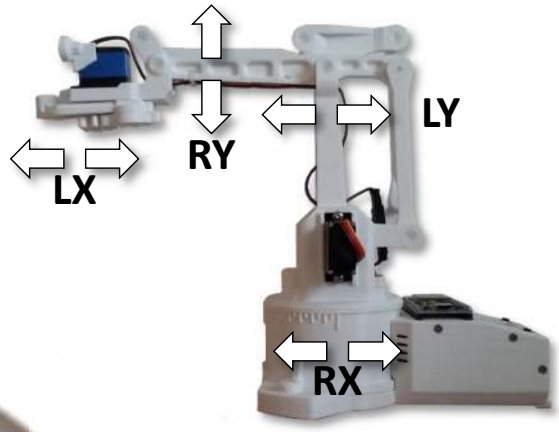




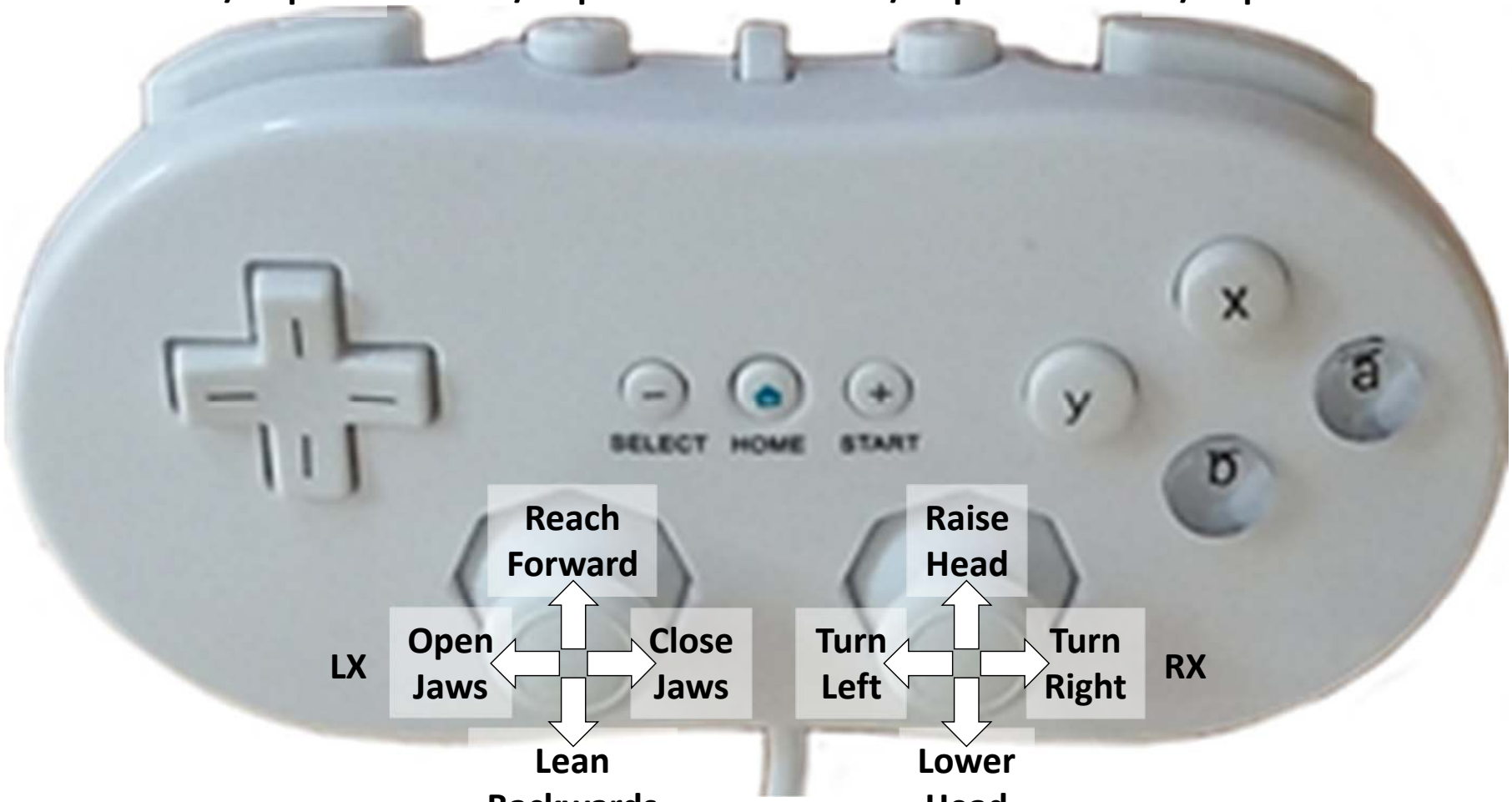




Reach Robot – Wii Functions



1/2 Speed 1/4 Speed 1/4 Speed 1/2 Speed



Reach Forward

LX

Open Jaws

Close Jaws

Lean Backwards LY

Raise Head

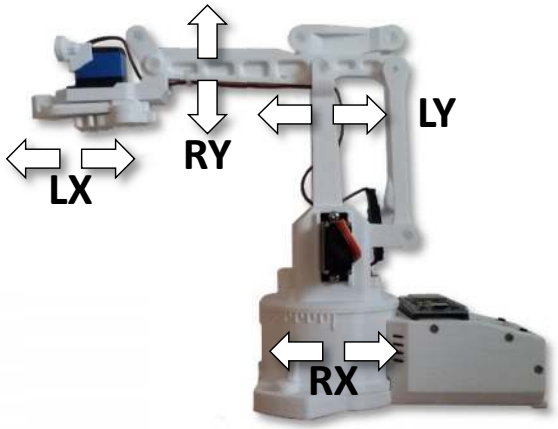
Turn Left

Turn Right

RX

Lower Head RY

Reach Robot Mk1 (Basic) – Wii Classic Pro Functions



1/4 Speed

1/4 Speed

1/2 Speed

1/2 Speed

Wii

Sit Back

Ready

Move Down

SELECT

HOME

START

x

y

a

b

Reach Forward

Raise Head

LX

Open Jaws

Close Jaws

Turn Left

Turn Right

RX

Lean Backwards LY

Lower Head RY