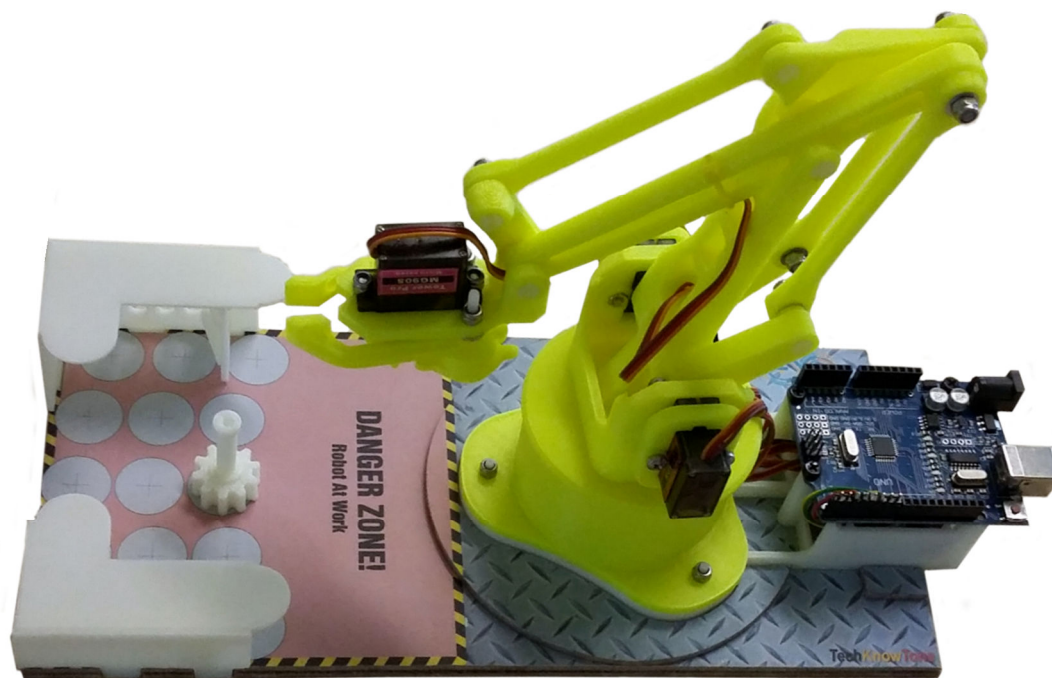


Reach Robot

Servo Calibration



Why do we need to calibrate the servos?

- No two servos are the same
- Servo pulse timings can vary widely (nominally $1000\mu\text{s}$ – $2000\mu\text{s}$)
- Servos can be damaged if not setup correctly
- Course calibration must be performed prior to the assembly process
- Course calibration determines min/max servo mechanical limits
- Fine calibration determines min/max robot mechanical limits
- The Arduino C++ code needs limit values in order to work accurately

Servo calibration is performed in three stages:

- Course calibration, performed prior to assembly
- Pre-set ensures mechanical parts are assembled correctly
- Fine calibration, performed during testing



HJ Servo Consistency Tester



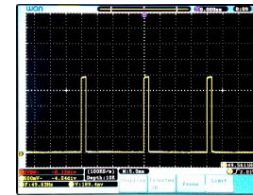
Select:

- Variable (Pot) pulse width 800 – 2200 μ s (default mode)
- Fixed constant pulse width 1500 μ s
- Sweep (Pot) pulse width from 800 -> 2200 -> 800 μ s

Pulse Width:

This is actually pulse frequency (Hz)

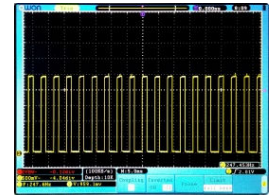
- 50H = 50 Hz - run MG90S at this frequency (default)
- 125H = 125 Hz
- 250H = 250 Hz



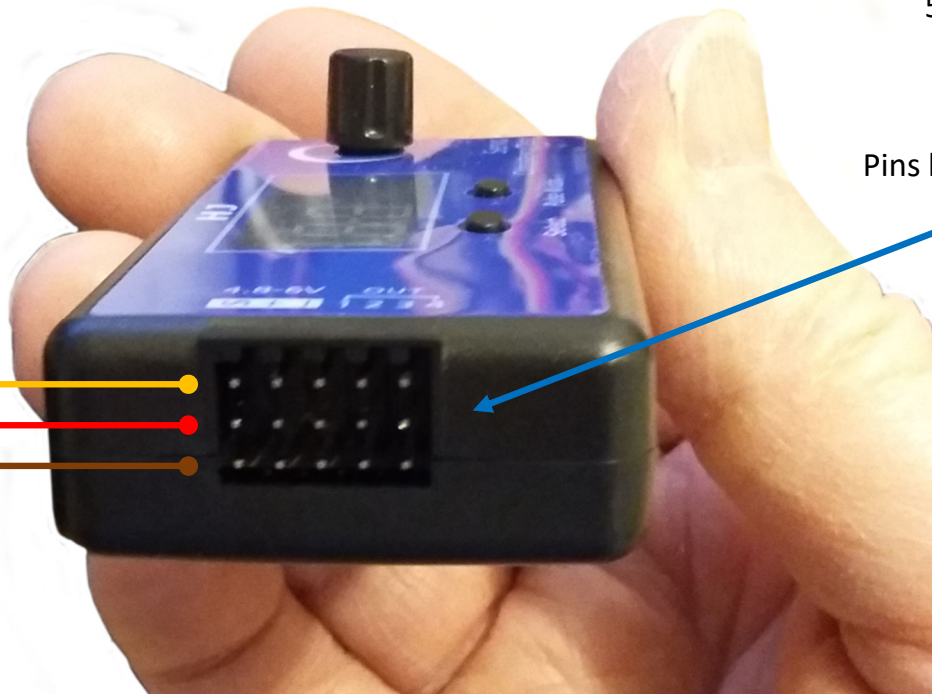
50Hz



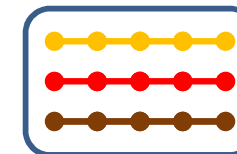
125Hz



250Hz



Pins have common connections

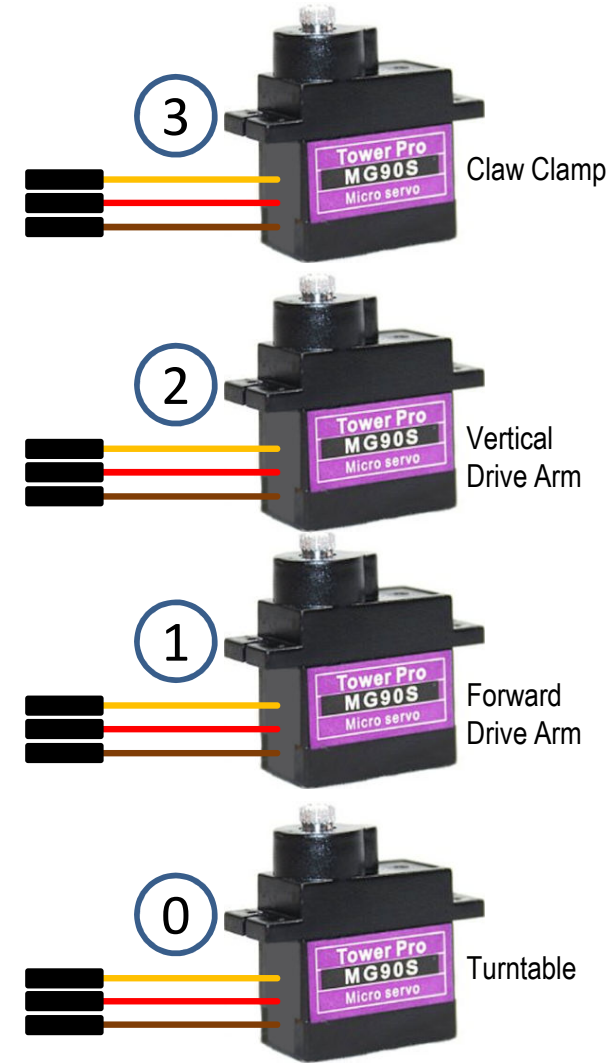


Signal
+4.8 - 6V
GND

Course calibration:

Confirms servos consistently provide adequate movement:

- Number your servos so that you can always identify them
- Use the consistency tester to determine their min/max limits
- Only connect **ONE** servo at a time whilst power is **OFF**
- Select the middle 1500µs output setting
- Attach an Angle Gauge and long control arm to observe movement
- Turn the Angle Gauge to align the 90° mark with the control arm pointer
- Cycle the servo for 2 minutes at full rate, to observe warm-up offset (drift)
- Record and reset the gauge, record your results in a table like the one below:



Servo 0

Turntable

Warm-up offset	_____ °	Angular range	Å _____ °
Angle at 2200µs	_____ °	1500-2200 range	Å+ _____ °
Angle at 800µs	_____ °	800-1500 range	Å- _____ °

Servo 1

Forward Drive Arm

Warm-up offset	_____ °	Angular range	Å _____ °
Angle at 2200µs	_____ °	1500-2200 range	Å+ _____ °
Angle at 800µs	_____ °	800-1500 range	Å- _____ °

Servo 2

Vertical Drive Arm

Warm-up offset	_____ °	Angular range	Å _____ °
Angle at 2200µs	_____ °	1500-2200 range	Å+ _____ °
Angle at 800µs	_____ °	800-1500 range	Å- _____ °

Servo 3

Claw Clamp

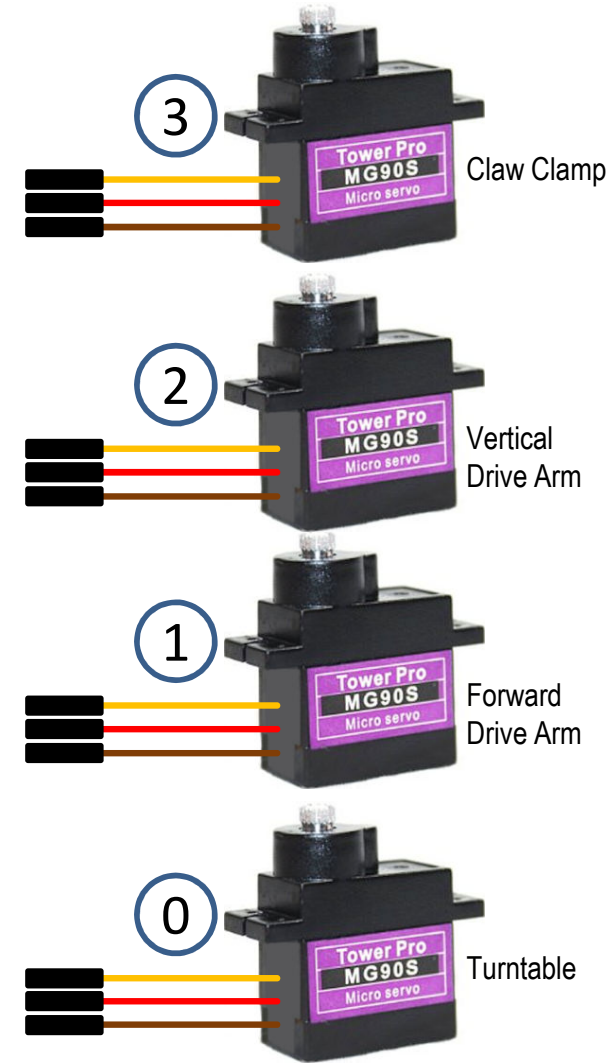
Warm-up offset	_____ °	Angular range	Å _____ °
Angle at 2200µs	_____ °	1500-2200 range	Å+ _____ °
Angle at 800µs	_____ °	800-1500 range	Å- _____ °

EXAMPLE

Course calibration:

Confirms servos consistently provide adequate movement:

- Number your servos so that you can always identify them
- Use the consistency tester to determine their min/max limits
- Only connect **ONE** servo at a time whilst power is **OFF**
- Select the middle 1500µs output setting
- Attach an Angle Gauge and long control arm to observe movement
- Turn the Angle Gauge to align the 90° mark with the control arm pointer
- Cycle the servo for 2 minutes at full rate, to observe warm-up offset (drift)
- Record and reset the gauge, record your results in a table like the one below:



Servo 0

Turntable

Warm-up offset	<u>0</u> °	Angular range	Å	<u>136</u> °
Angle at 2200µs	<u>154</u> °	1500-2200 range	Å+	<u>64</u> °
Angle at 800µs	<u>18</u> °	800-1500 range	Å-	<u>72</u> °

Servo 1

Forward Drive Arm

Warm-up offset	<u>-1.5</u> °	Angular range	Å	<u>133</u> °
Angle at 2200µs	<u>154</u> °	1500-2200 range	Å+	<u>64</u> °
Angle at 800µs	<u>21</u> °	800-1500 range	Å-	<u>69</u> °

Servo 2

Vertical Drive Arm

Warm-up offset	<u>-1</u> °	Angular range	Å	<u>135</u> °
Angle at 2200µs	<u>155</u> °	1500-2200 range	Å+	<u>65</u> °
Angle at 800µs	<u>20</u> °	800-1500 range	Å-	<u>70</u> °

Servo 3

Claw Clamp

Warm-up offset	<u>-2</u> °	Angular range	Å	<u>138</u> °
Angle at 2200µs	<u>155</u> °	1500-2200 range	Å+	<u>65</u> °
Angle at 800µs	<u>17</u> °	800-1500 range	Å-	<u>73</u> °

Servo Pre-set For Assembly:

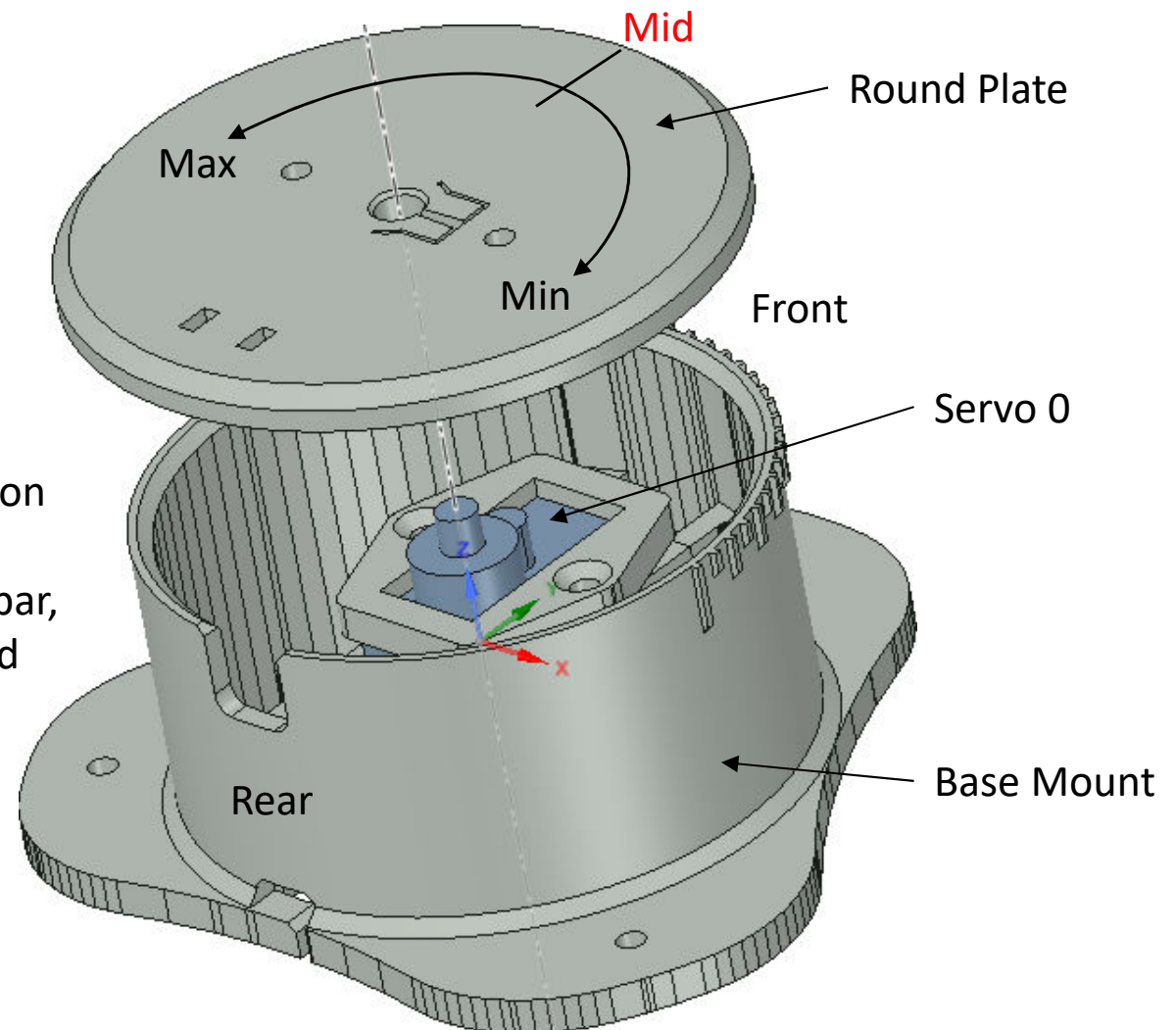
This ensures the attached mechanical part will have sufficient range.

- Select the correct servo. ie. Servo 0 for turntable
- Connect servo to consistency tester and set appropriate value
- Attach mechanical part to the servo in the position indicated



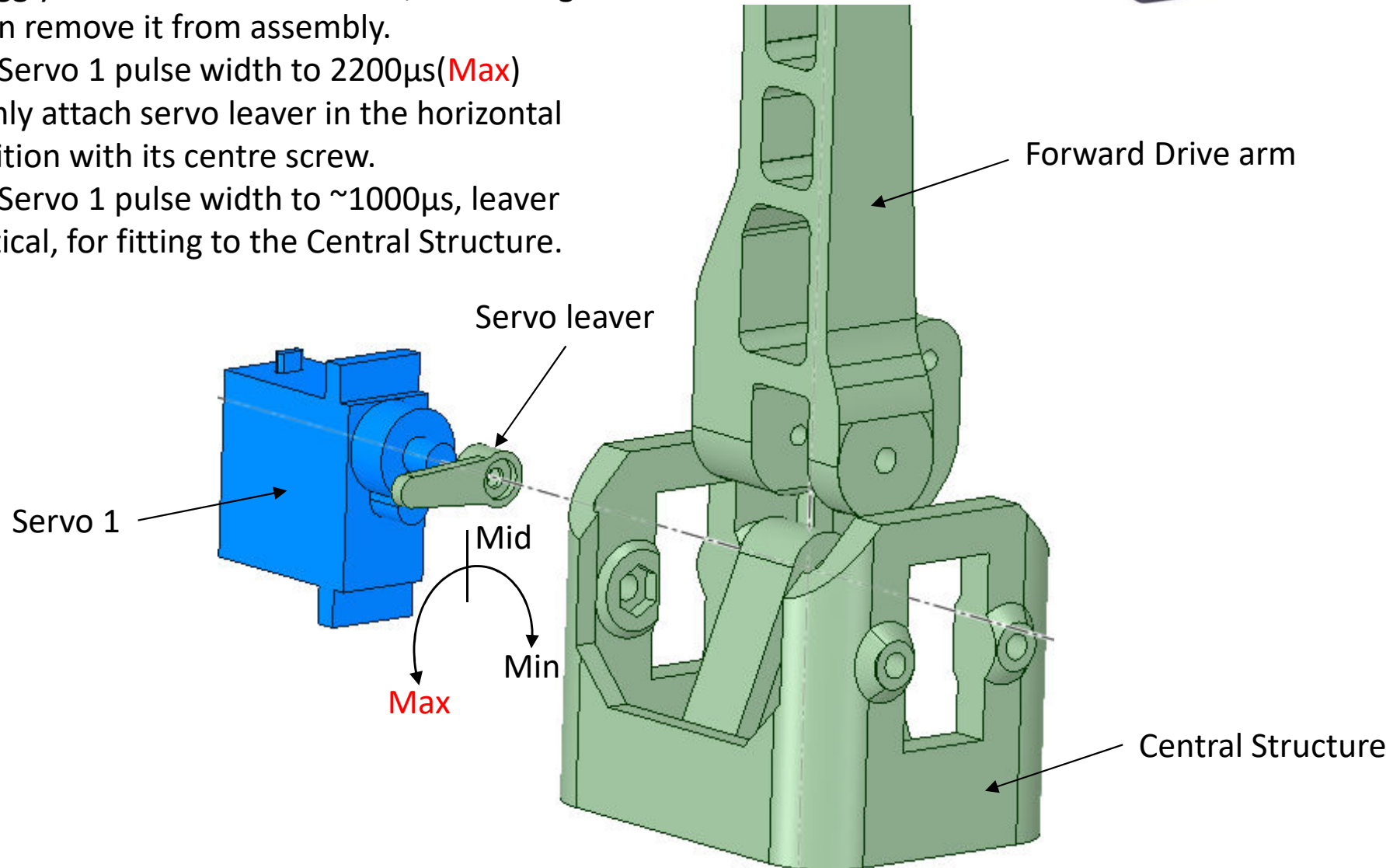
Servo 0 - Pre-set For Assembly:

- Attach Servo 0 to the Base Mount
- Set Servo 0 pulse width to 1500 μ s (**Mid**)
- Attach round plate in forward facing position shown using a long thin screwdriver
- Alternatively connect twin armed control bar, with arms facing front/rear and glue Round Plate onto the control bar.



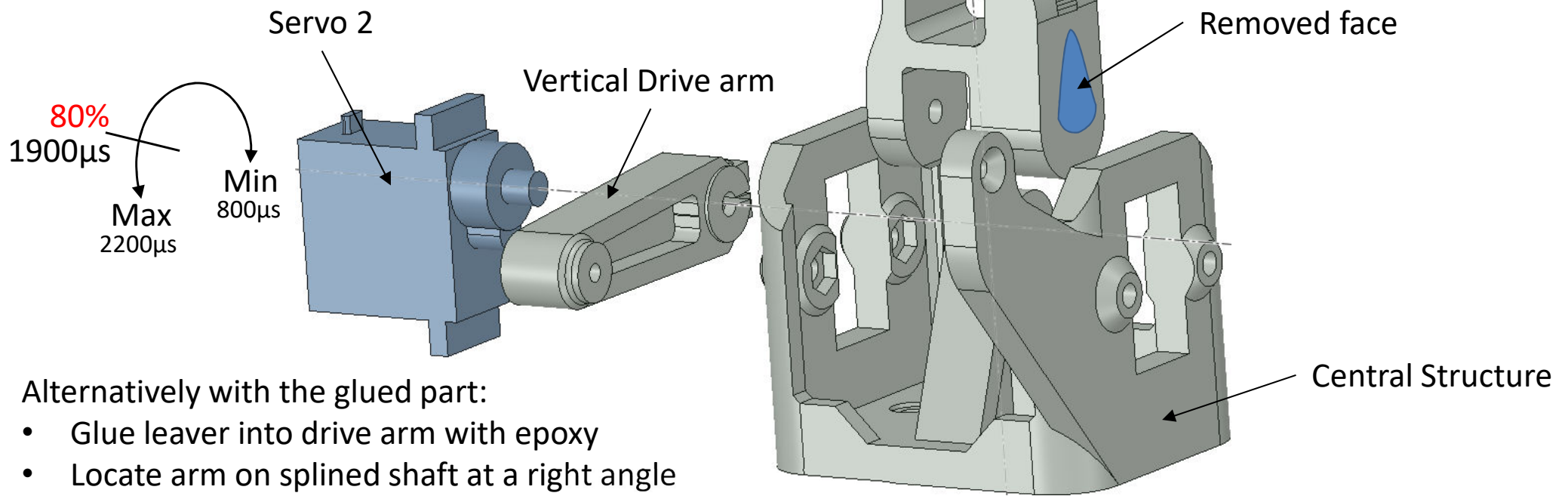
Servo 1 - Pre-set For Assembly:

- Ensure Servo 1 with lever attached will engage snugly with Forward Drive arm, but not tight.
- Then remove it from assembly.
- Set Servo 1 pulse width to 2200 μ s(**Max**)
- Firmly attach servo lever in the horizontal position with its centre screw.
- Set Servo 1 pulse width to \sim 1000 μ s, lever vertical, for fitting to the Central Structure.



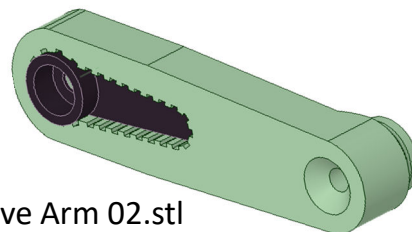
Servo 2 - Pre-set For Assembly:

- Attach Servo 2 to the Central Structure
- Have Vertical Drive arm attached loosely
- Set Servo 2 pulse width to 1900 μ s (80% range value)
- Calculated from previously recorded values
- Attach Vertical Drive arm in horizontal position shown



Alternatively with the glued part:

- Glue lever into drive arm with epoxy
- Locate arm on splined shaft at a right angle
- Fit and tighten centre screw
- Feed into the Central Structure



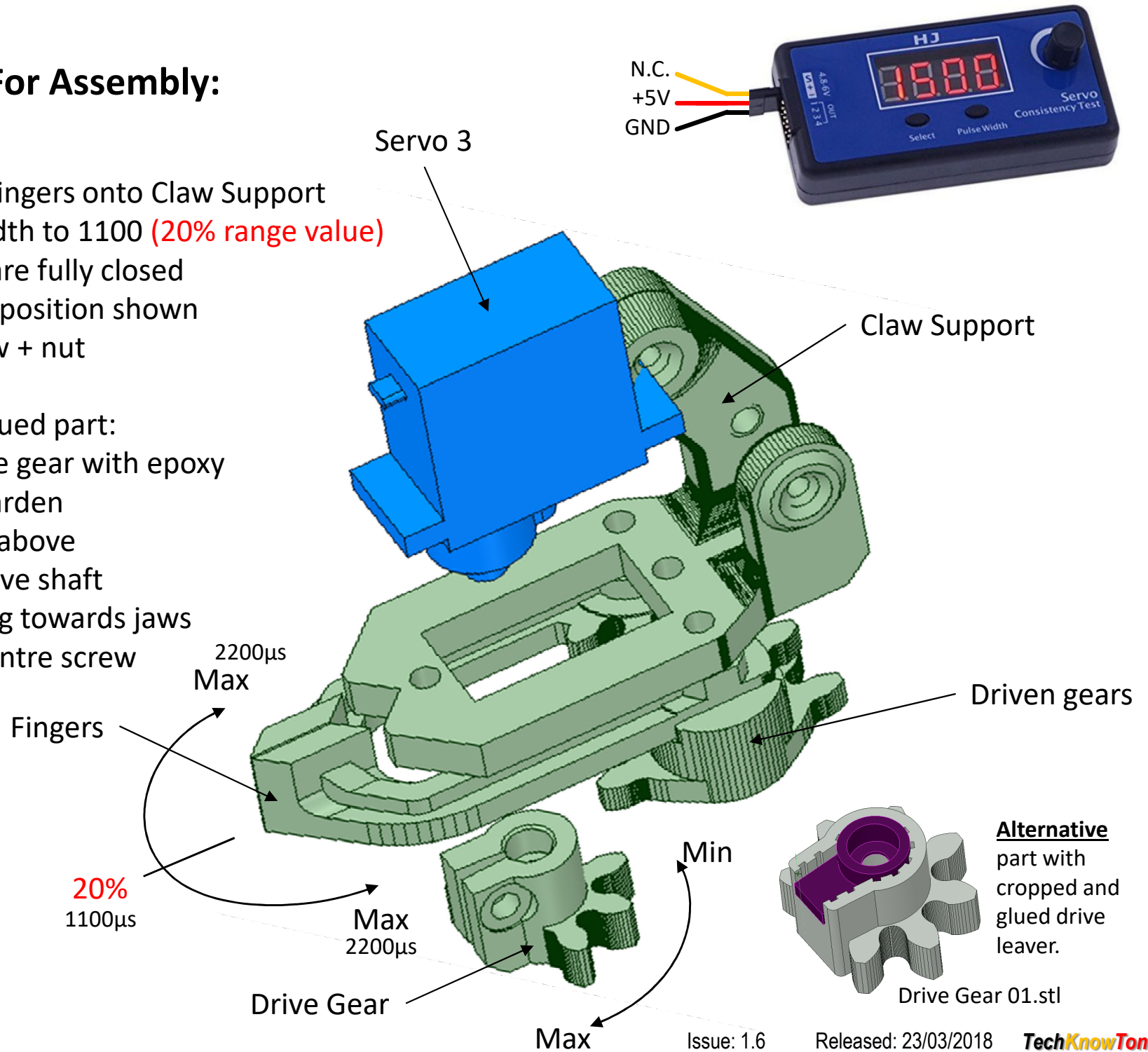
Alternative part with servo lever glued into Vertical Drive arm.

Servo 3 - Pre-set For Assembly:

- Mount Servo 3 and fingers onto Claw Support
- Set Servo 3 pulse width to 1100 (20% range value)
- Ensure claw fingers are fully closed
- Attach Drive Gear in position shown
- Tighten locking screw + nut

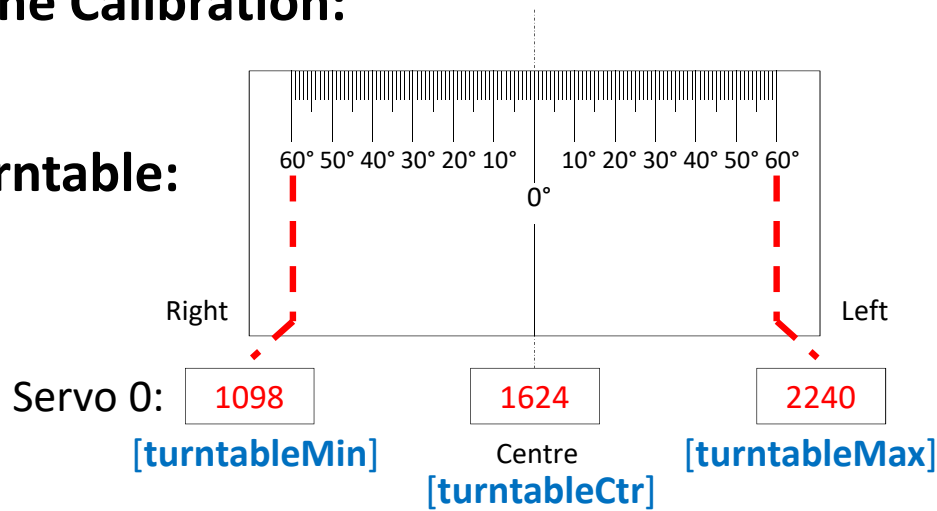
Alternatively with the glued part:

- Glue lever into drive gear with epoxy
- Allow glue to fully harden
- Set Servo 3 angle as above
- Locate on splined drive shaft
- Cropped lever facing towards jaws
- Insert and tighten centre screw

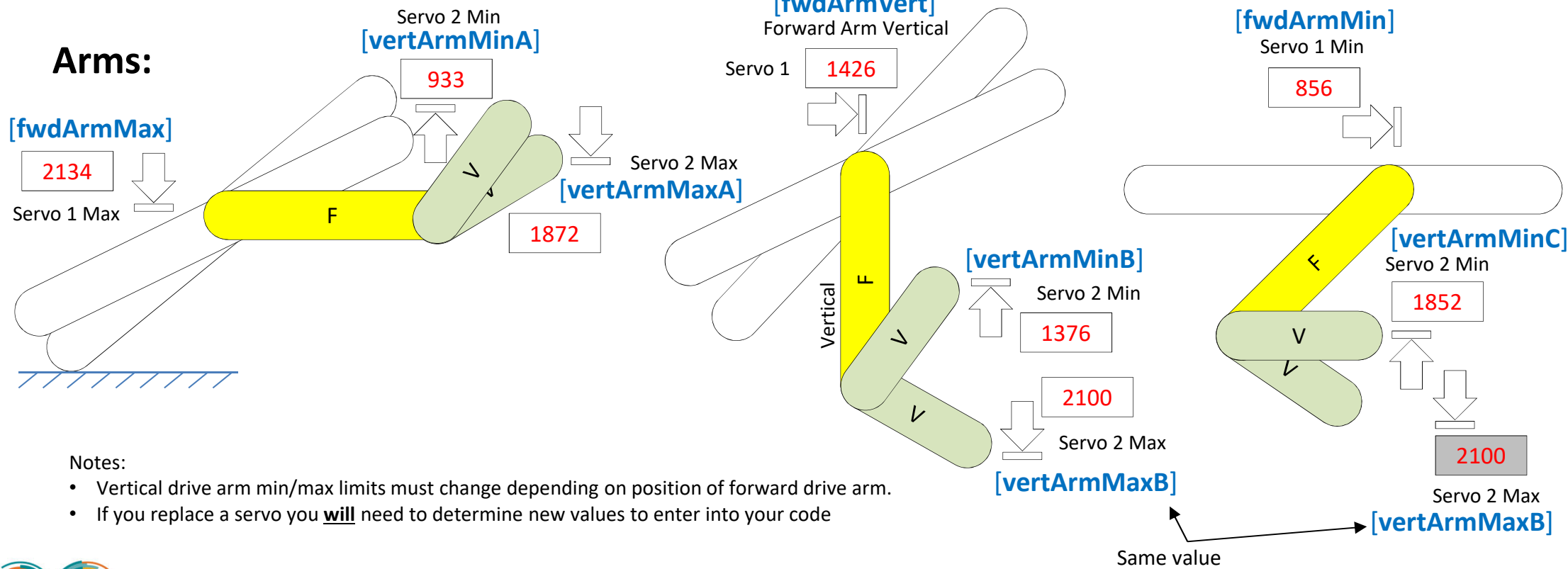


Base Servos Fine Calibration:

Turntable:



Arms:



Notes:

- Vertical drive arm min/max limits must change depending on position of forward drive arm.
- If you replace a servo you **will** need to determine new values to enter into your code

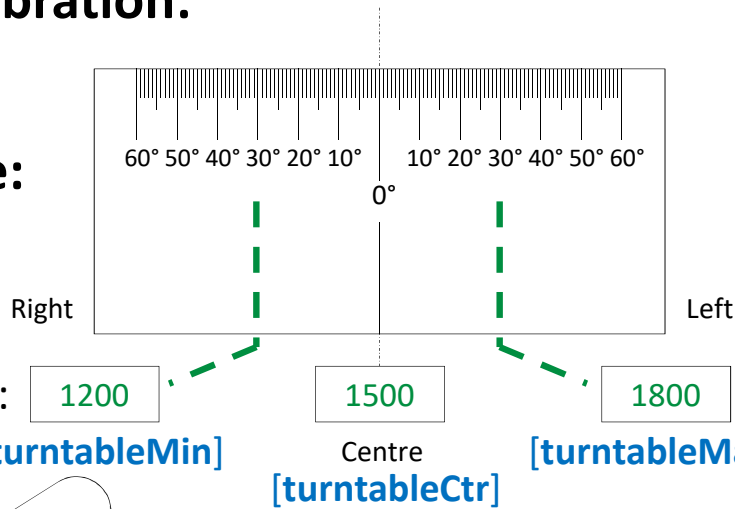
Servo Calibration Record Sheet

EXAMPLE

These are the numbers I obtained from my servos and used in the Arduino code provided. You will determine different values for your servos.

Base Servos Fine Calibration:

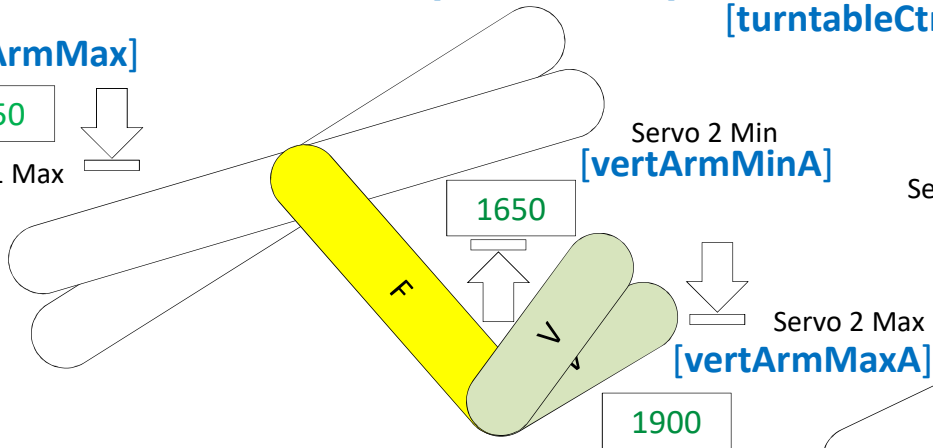
Turntable:



Arms:

[fwdArmMax]

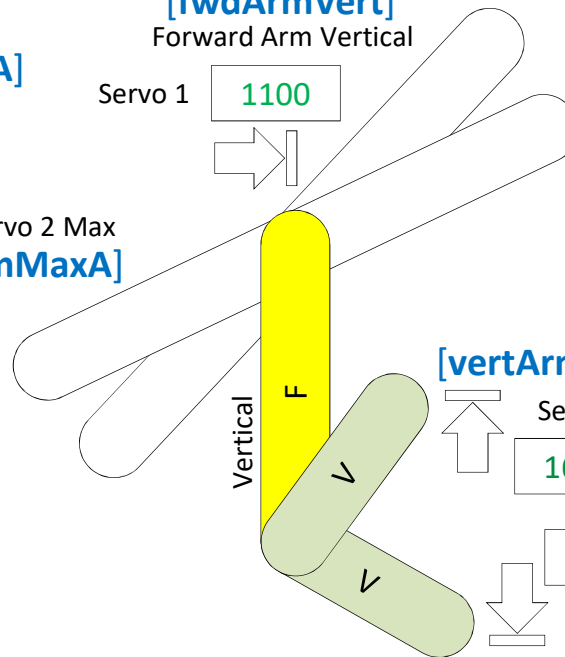
1450
Servo 1 Max



[fwdArmVert]

Forward Arm Vertical

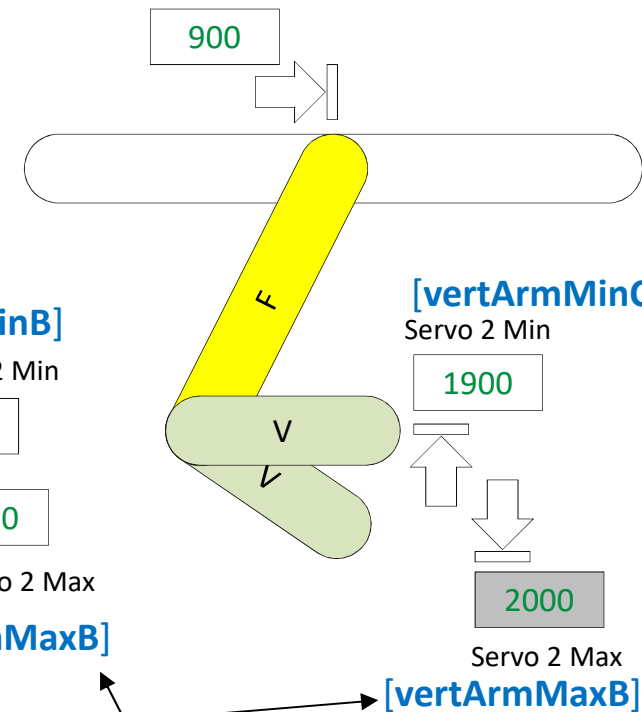
Servo 1 1100



[fwdArmMin]

Servo 1 Min

900



Same value

Servo Calibration Record Sheet

DEFAULT

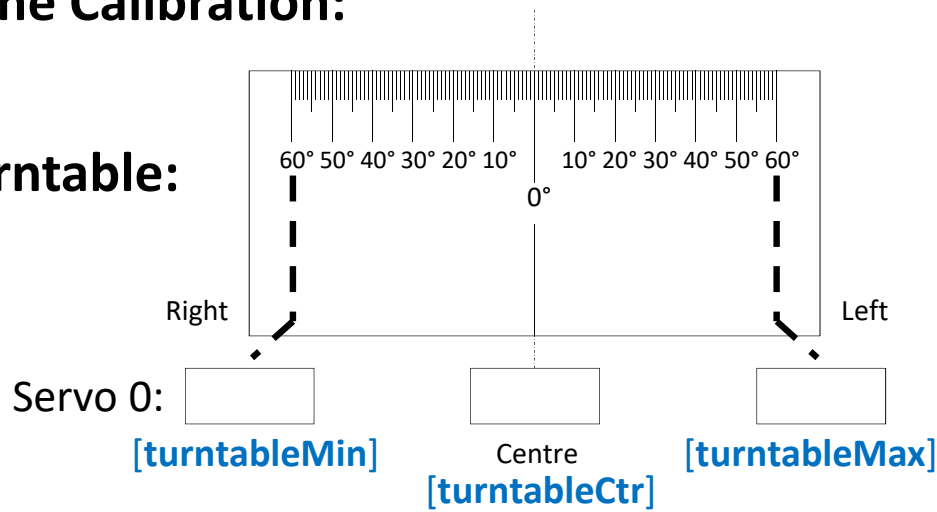
These are the numbers assigned in [Reach_Robot_Cal_00.ino](#) as default values. They are your starting point, and should be 'safe' values. You will determine different values for your servos as you calibrate your robot.

Notes:

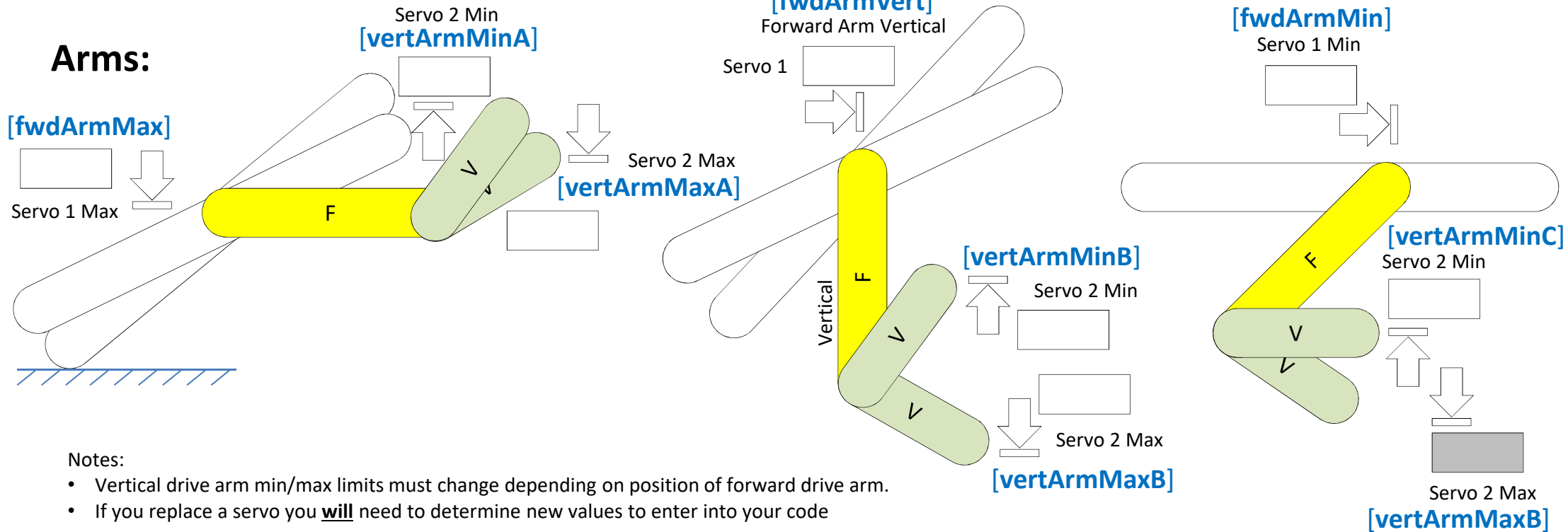
- Vertical drive arm min/max limits must change depending on position of forward drive arm.
- If you replace a servo you **will** need to determine new values to enter into your code

Base Servos Fine Calibration:

Turntable:



Arms:



Notes:

- Vertical drive arm min/max limits must change depending on position of forward drive arm.
- If you replace a servo you **will** need to determine new values to enter into your code

Servo Calibration Record Sheet

These are obtained from your servos and used in the Arduino code provided.

EXAMPLE

These are the numbers I obtained from my servos and used in the Arduino code provided. You will determine different values for your servos.

These values are determined during the 'Fine' calibration process and entered into the code as constants. Follow the instructions on the following pages to determine unique servo values for your robot.

Arduino Code Values:

- We use servo PWM values as constants in the Arduino code:

```
// Define servo calibration constants
#define fwdArmMax 2276 // forward arm Max servo value
#define fwdArmMin 1006 // forward arm Min servo value
#define fwdArmVert 1481 // forward arm vertical servo value
#define gripClose 962 // jaws closed servo value
#define gripOpen 1360 // jaws moderately open value (23%)
#define gripWide 2200 // jaws wide open value
#define turntableCtr 1461 // turntable servo centre value
#define turntableMax 2161 // turntable servo Max value
#define turntableMin 762 // turntable servo Min value
#define vertArmMaxA 1907 // vertical arm Max 'A' servo value
#define vertArmMaxB 2151 // vertical arm Max 'B' servo value
#define vertArmMinA 951 // vertical arm Max 'A' servo value
#define vertArmMinB 1333 // vertical arm Max 'B' servo value
#define vertArmMinC 1894 // vertical arm Max 'C' servo value

#define Home0 turntableCtr // home position for servo 0
#define Home1 1500 // home position for servo 1
#define Home2 1900 // home position for servo 2
#define Home3 gripOpen // home position for servo 3
#define Reset0 turntableCtr // RESET position for servo 0
#define Reset1 1500 // RESET position for servo 1
#define Reset2 1900 // RESET position for servo 2
#define Reset3 gripOpen // RESET position for servo 3
#define servoOffMax 44 // sets maximum thermal drift offset for servo 0
#define servoOffRmpDwn 60000 // sets thermal offset ramp down time in milliseconds
#define servoOffRmpUp 10000 // sets thermal offset ramps up time in milliseconds
```

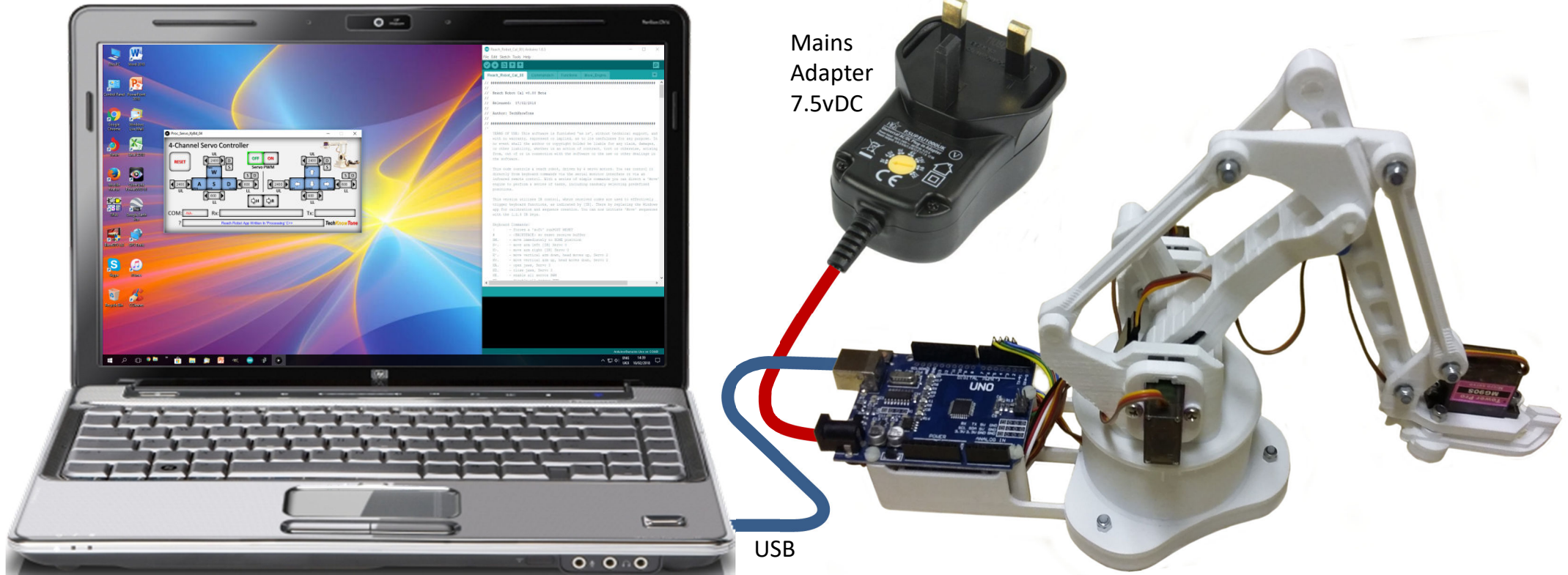


'Reset'



'Home'

Servo Fine Calibration:



Tools needed for Servo fine calibration:

- Windows PC, with Java runtime installed
- Arduino IDE; latest version with necessary libraries
- Download Software Code .zip file containing:
 - Servo Keyboard app: [Servo_KeyBrd_Cntrl.exe](#)
 - Arduino .ino files: [Reach_Robot_Cal_00.ino](#)
- Extract files from .zip to respective folders
- Load the [Reach_Robot_Cal_00.ino](#) file into the IDE
- Immediately save it to a filename of your choice
- Something like > [My_Reach_Robot_00.ino](#)

Note: These instructions assume that you know how to use the Arduino IDE and have a basic understanding of writing C++ sketches, and transferring them to an Arduino UNO. If you do not, you will need to learn this prior to completing these calibration tasks.

Your robot will need to be powered from a 7.5v mains adapter for the servo motors to function.

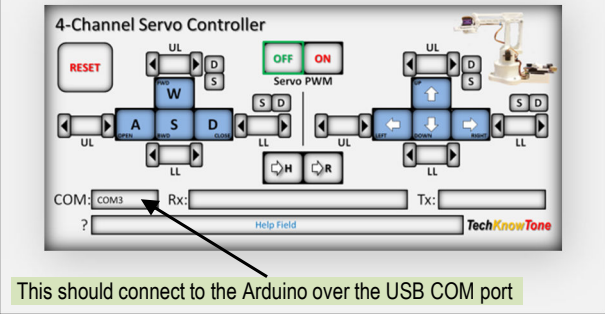
You can't use the IDE serial monitor or program the Arduino whilst the keyboard app is running, as it hogs the USB interface.

Servo Fine Calibration:

With the .ino file loaded the robot will adopt this posture... 01

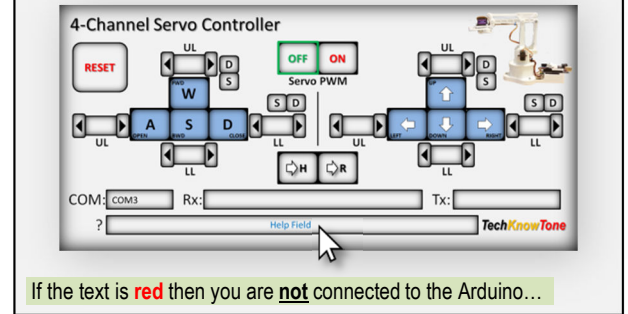


Now launch the keyboard Controller app... 02



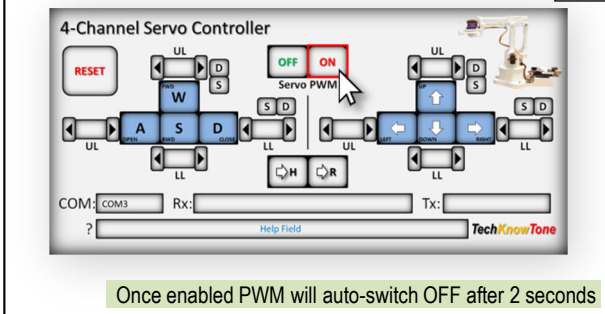
This should connect to the Arduino over the USB COM port

Move the mouse pointer to learn what buttons are for... 03



If the text is red then you are **not** connected to the Arduino...

To moved servos you must enable the PWM signals... 04



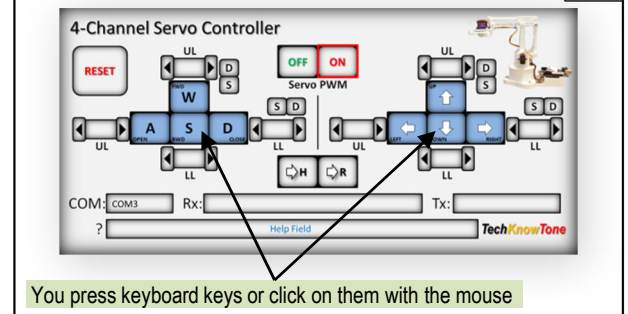
Once enabled PWM will auto-switch OFF after 2 seconds

PWM auto-switch OFF is a feature of the Arduino sketch code. 05

It serves to reduce overall power consumption, when the robot is not moving, prevents servos from over heating and ensures longer life. The PWM signals will be turned back on again automatically, whenever a servo movement is required.

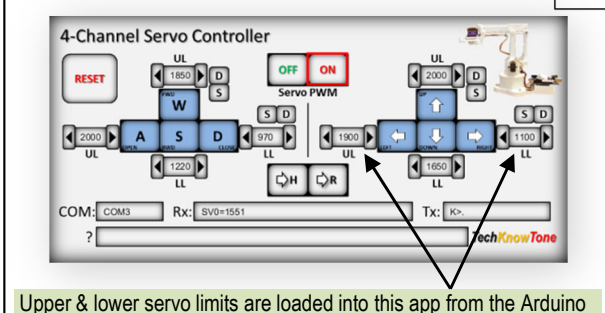


The blue coloured buttons correspond to keyboard keys... 06



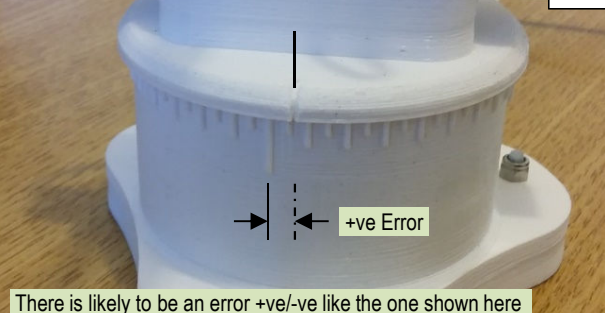
You press keyboard keys or click on them with the mouse

The robots servo movements are restricted to start with... 07



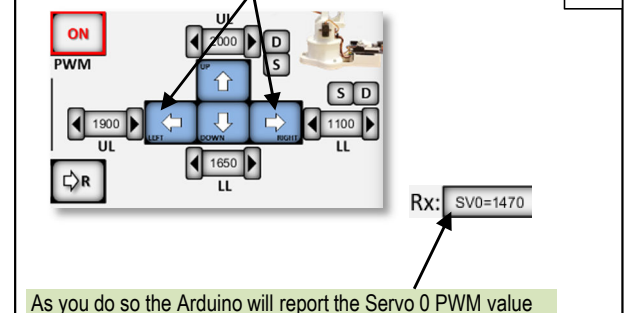
Upper & lower servo limits are loaded into this app from the Arduino

Observe the start-up centred position of the robot... 08



There is likely to be an error +ve/-ve like the one shown here

Use the left or right arrow keys to remove this error... 09



As you do so the Arduino will report the Servo 0 PWM value

Servo Fine Calibration:

The Round Plate mark and centre line should be aligned...

Rx: SV0=1461

Zero Error

Record the SV0 value needed to achieve this centre position

10

Find the default #define value in the .ino file...

```
#define turntableCtr 1500
```

Change this number to the one you have determined

```
#define turntableCtr 1461
```

Compile and send the new file to your Arduino UNO

You will need to stop/disconnect the keyboard app from USB COM!

11

Relaunch/re-connect the keyboard app...

Rx: S0=1461, S1=1500, S2=1900, S3=1300

New default value shown here

After Resetting, press the SPACEBAR to see the servo values

12

Turn ON the PWM and invoke the sinusoidal sweep...

S-Box highlight will be blue until sinusoidal sweep is in synch, Red

13

After 3 minutes turn OFF the sinusoidal sweep as the robot passes the centre position...

Then quickly click on the go to RESET position button

14

You are likely to see an offset, caused by Servo 0 heating up as it works, resulting in a thermal offset ...

Thermal Error

We need to remove this to improve the accuracy of our robot!

15

Quickly switch ON the Servo 0 Dither function...

Dither applies a small shifting function to the Servo 0 value

16

With Dither ON adjust Servo 0 value to remove the error...

Rx: SV0=1417

Zero Error

Once happy switch Dither OFF to reveal new centre position

17

Find the default #define value in the .ino file...

```
#define servoOffMax 0
```

Subtract the latest value from the previous, $1461 - 1417 = 44$

Change the default servoOffMax number to this value...

```
#define servoOffMax 44
```

Compile and send the modified sketch file to your Arduino UNO

You will need to stop/disconnect the keyboard app from USB COM!

18

Servo Fine Calibration:

Turn ON PWM and Servo 0 sweep function, then Increase Servo 0 upper limit UL towards 2200µs...

219

Adjust Servo 0 UL value until it just reaches the +60° rotational limit at the extreme of travel. Note this value.

Decrease Servo 0 lower limit LL towards 750µs...

220

Adjust Servo 0 LL value until it just reaches the -60° rotational limit at the extreme of travel. Note this value.

Find these two default #define values in the .ino file...

221

```
#define turntableMax 1900
#define turntableMin 1100
```

Change the default numbers to your values like this...

```
#define turntableMax 2161
#define turntableMin 762
```

Compile and send the new file to your Arduino UNO

Remember to stop/disconnect the keyboard app from USB COM!

You have now determined the turntable values...

222

Turntable:

Right -60° 0° +60° Left

Servo 0: **762** **1461** **2161**

[turntableMin] **[turntableCtr]** **[turntableMax]**

These three Servo 0 values are unique to your robot!

Now find Servo 1 Forward Arm vertical PWM value...

223

You won't need to change LL/UL limits...

The Rx value should be close to the 1100µs you set the servo arm at

Record the Servo 1 value you have determined like this...

224

```
[fwdArmVert]
Forward Arm Vertical
Servo 1 1074
```

We now determine a minimum value for Servo 2...

225

Note that the Claw goes up as the arm swings down...

Switch on the sinusoidal sweep function for Servo 2...

226

This will help you observe the Servo 2 upper limit value...

Adjust the Servo 2 UL value until the arm is within 2mm...

227

Adjust the UL value slowly so as avoid a collision!

Servo Fine Calibration:

Record the Servo 2 UL value you have determined... 28

Adjust Servo 2 LL value to the point at which Forward Drive and Horizontal arms within 1mm of each other... 29

If a collision occurs the Forward Drive Arm will be pushed forwards!

Record the Servo 2 LL value you have determined... 30

Adjust Servo 2 value setting the Vertical Drive Arm horizontal 31

Servo 2 value returned from Arduino

Record the Servo 2 value you have determined... 32

Now find Servo 1 Forward Arm PWM value for this position... 33

Avoid going beyond the pinch point! Set it just above that point.

Record the Servo 1 value you have determined 34

Set Servo 1 & Servo 2 UL & LL limits as follows... 35

This will give you freedom to determine the next settings ...

Move the Forward Drive Arm to an approx. 45° angle... 36

You will need to move Servo 2 to achieve this

37 Move Servo 2 until the arms begin to pinch here...

Rx: SV2=951

Take care not to over strain the servo motor!

38 Record the Servo 2 value you have determined...

Servo 2 Min
[vertArmMinA]
951 Rx: SV2=951

39 Adjust Servo 1 to set Forward Drive Arm nearly horizontal...

Rx: SV1=2090

Triangular Link

There is a pinch point in the linkage to be avoided!

40 Record the Servo 1 value you have determined...

[fwdArmMax]
2090 Servo 1 Max
Rx: SV1=2090

41 Adjust Servo 2 to where it just pinches with the Forward Arm...

Rx: SV2=1695

Then ease it back so that it is just free!

42 Record the Servo 2 value you have determined...

Servo 2 Max
[vertArmMaxA]
1695 Rx: SV2=1695

That completes the arm measurements

43 Now set the LL & UL limits for Servo 3 to 850 & 2200µs...

Note that 'A' key will open the jaws and the 'D' key closes them

44 Close the jaws until they just touch...

Rx: SV3=962

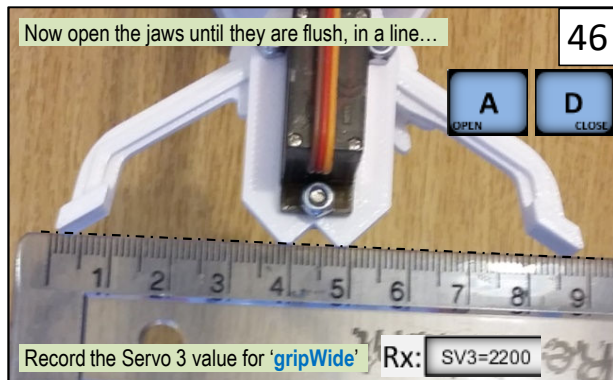
Record the Servo 3 value for 'gripClose'

Due to significant gear backlash only record the value when closing!

45 Now open the jaws to 20mm...

Rx: SV3=1360

Record the Servo 3 value for 'gripOpen'



46

You must now enter all of your recorded values in your Arduino sketch, replacing the 'default' values provided in the code.

47

This provides your robot with much greater freedom of movement, and important limits which are used by the code to ensure that the linkages work correctly.

See the code examples below comparing the default values with those determined in these measurements.

Finally change a flag value in the Arduino sketch...

49

```
int Calibrated = false;
```

The variable 'Calibrated' can now be set to true...

```
int Calibrated = true;
```

This changes the behaviour of your code for the 'Move Engine'.

Default Values

```
#define fwdArmMax 1450
#define fwdArmMin 900
#define fwdArmVert 1100
#define gripClose 970
#define gripOpen 1400
#define gripWide 2000
#define turntableCtr 1500
#define turntableMax 1900
#define turntableMin 1100
#define vertArmMaxA 1900
#define vertArmMaxB 2000
#define vertArmMinA 1650
#define vertArmMinB 1650
#define vertArmMinC 1900
```

Calibrated Values

```
// Define servo calibration constants
#define fwdArmMax 2090 // forward arm Max servo value
#define fwdArmMin 654 // forward arm Min servo value
#define fwdArmVert 1074 // forward arm vertical servo value
#define gripClose 942 // jaws closed servo value
#define gripOpen 1360 // jaws moderately open value (23%)
#define gripWide 2200 // jaws wide open value
#define turntableCtr 1461 // turntable servo centre value
#define turntableMax 2161 // turntable servo Max value
#define turntableMin 762 // turntable servo Min value
#define vertArmMaxA 1695 // vertical arm Max 'A' servo value
#define vertArmMaxB 2152 // vertical arm Max 'B' servo value
#define vertArmMinA 951 // vertical arm Max 'A' servo value
#define vertArmMinB 1333 // vertical arm Max 'B' servo value
#define vertArmMinC 1894 // vertical arm Max 'C' servo value
```

```
#define Home0 turntableCtr // home position for servo 0
#define Home1 1200 // home position for servo 1
#define Home2 1900 // home position for servo 2
#define Home3 gripOpen // home position for servo 3
#define Reset0 turntableCtr // RESET position for servo 0
#define Reset1 1200 // RESET position for servo 1
#define Reset2 1900 // RESET position for servo 2
#define Reset3 gripOpen // RESET position for servo 3
#define servoOffMax 0 // sets maximum thermal drift offset for servo 0
#define servoOffRmpDwn 60000 // sets thermal offset ramp down time in milliseconds
#define servoOffRmpUp 10000 // sets thermal offset ramps up time in milliseconds
```

48

Given what you have learnt you can now determine and enter servo values for the 'Home' and 'Reset' positions...

50



Reset is normally the rest or powered down position...

51

