

Assembly Manual for v4.1

MeArm.io

Arduino & RasPi Edition

Manual Revision 4.1 by Joshua Vanderwall March 2019

This assembly manual is based on a work by Scott Pierce

CONTENTS

Introduction.....	2
Setting up the Arduino.....	3
Setting up the Raspberry Pi.....	5
Setting up the Servos with Raspberry Pi.....	8
Calibrating the Servo Motors.....	10
Building the MeArm.....	11
References.....	24

INTRODUCTION

This manual is for the assembly of the MeArm Version 4.1. It is my intention that with this document you can build the MeArm with nothing but a screwdriver and enthusiasm. You need no prior knowledge and will finish as a robot builder -- ready to take the first steps towards mastering coding, electronics, robotics or whatever your plans for world domination require.

This version is for use with an Arduino or Raspberry Pi board to control your MeArm. You will also need a power supply capable of delivering 5-6V at 2A, a breadboard and a few wires to connect everything together.

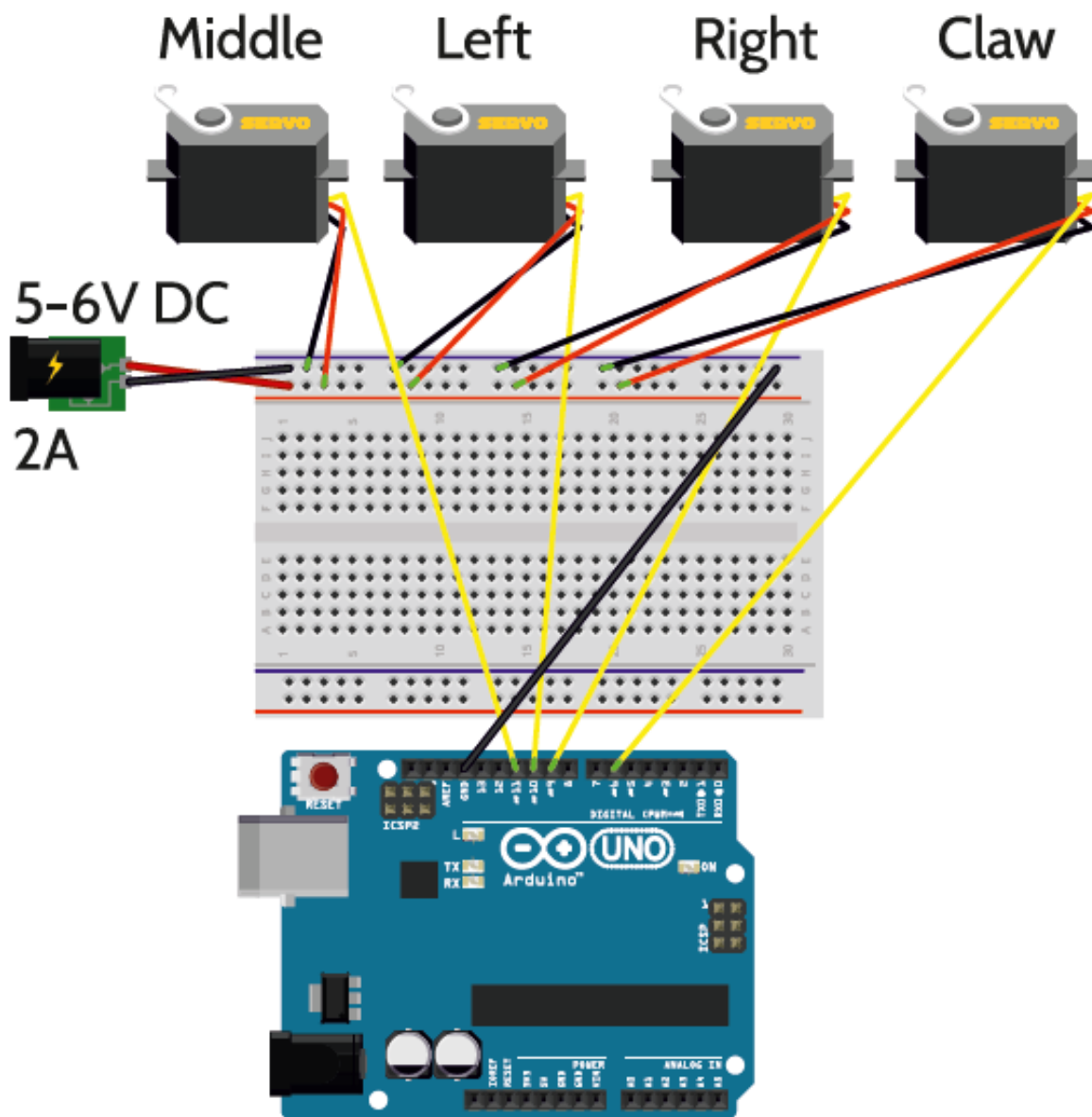
Thank you for joining us on the MeArm Project.

All the best!

Content: Carl Monk, Scott Pierce

Composition and Edits: Peter Jansen, Urs Utzinger, David Lesser, Joshua Vanderwall

Setting up the Arduino



fritzing

The Bread Board (white thing with holes) is used here as a “power rail” and can easily be dispensed with. Here it connects the Servo Motors to their own power supply, which is tied to a common ground on the Arduino. You can not power four servo motors through the Arduino.

WARNING! Exceeding 6V can irreparably damage the servo motors.

Setting up the Arduino

Step 1.

Using the configuration on the previous page, connect the servo motors to power and your Arduino.

Step 2.

Download and install the Arduino IDE

(<https://www.arduino.cc/en/Main/Software>). Install the drivers for your chosen Arduino board.

Step 3.

Connect your Arduino to your PC via USB Cable.

Step 4.

Start the Arduino IDE or go to codebender.cc/sketch:148456

Step 5.

Type:

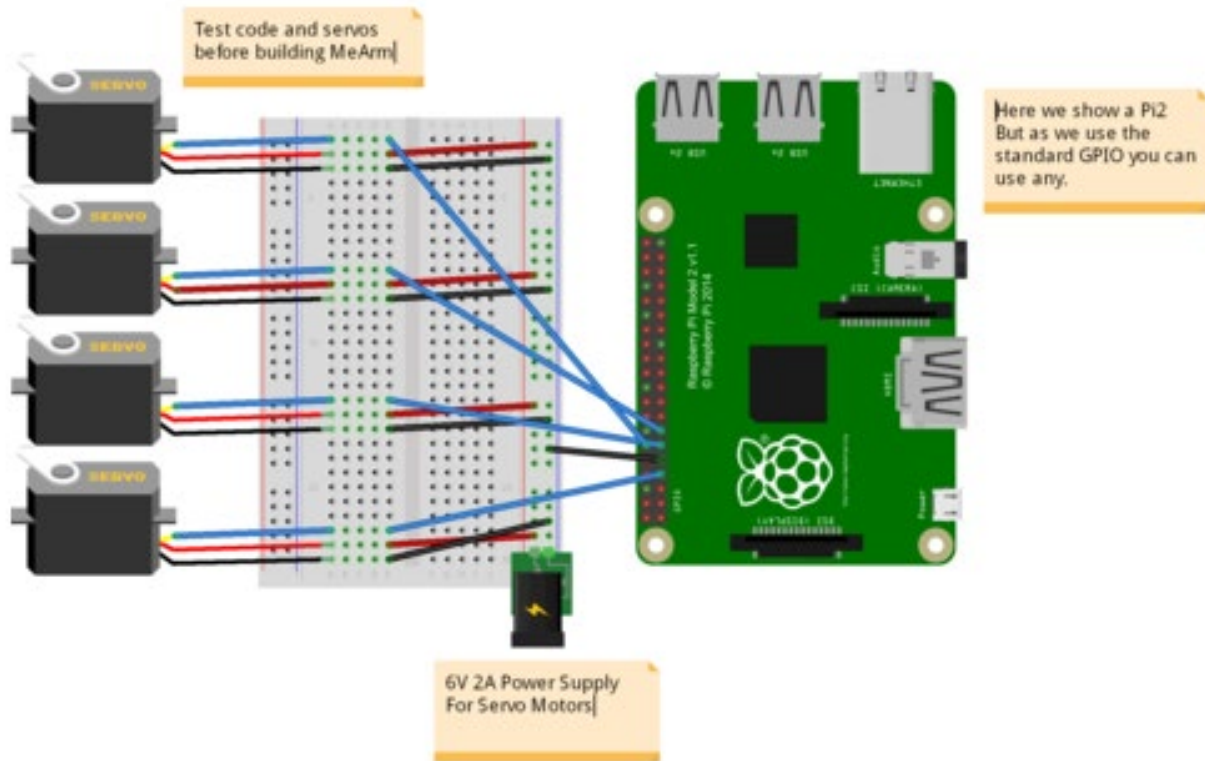
```
#include <Servo.h>

Servo middle, left, right, claw; // creates 4 "servo objects"

void setup()
{
  Serial.begin(9600);
  middle.attach(11); // attaches the servo on pin 11 to the middle object
  left.attach(10); // attaches the servo on pin 10 to the left object
  right.attach(9); // attaches the servo on pin 9 to the right object
  claw.attach(6); // attaches the servo on pin 6 to the claw object
}

void loop()
{
  middle.write(90); // sets the servo position according to the
  value(degrees)
  left.write(90); // does the same
  right.write(90); // and again
  claw.write(35); // yes you've guessed it.
  delay(300); // doesn't constantly update the servos which can fry them.
```

Setting up the Raspberry Pi



The breadboard used in the image is just to make things look tidy. In reality you can just attach all the grounds and all the power lines and wire them directly to the power, then bring the PWM lines (the servo control wires that come from the GPIO) directly into the servos. Don't connect the 6V power to the GPIO (other than connecting the ground wire). Also it's not advised to draw the power for the servos through the GPIO, they can draw up to an amp each and the Pi isn't set up for that. Better safe than have to buy a new Pi!

Setting up the Raspberry Pi

STEP 1: Get to the command line on your Pi. Either boot to it or use a terminal in your GUI. We used a fresh install of Raspian.

```
pi@raspberrypi ~ $ git clone
git://github.com/richardghirst/PiBits.git
Cloning into 'PiBits'...
remote: Reusing existing pack: 359, done.
remote: Total 359 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (359/359), 362.62 KiB | 311.00 KiB/s,
done.
Resolving deltas: 100% (154/154), done.
Checking connectivity... done.
pi@raspberrypi ~ $ cd PiBits/ServoBlaster/user
pi@raspberrypi ~/PiBits/ServoBlaster/user $ make servod
gcc -Wall -g -O2 -o servod servod.c -lm

pi@raspberrypi ~/PiBits/ServoBlaster/user $ sudo ./servod -idle-
timeout=2000
```

```
Board revision: 2
Using hardware: PWM
Using DMA channel: 14
Idle timeout: 2000
Number of servos: 8
Servo cycle time: 20000us
Pulse increment step size: 10us
Minimum width value: 50 (500us)
Maximum width value: 250 (2500us)
Output levels: Normal
Using P1 pins: 7,11,12,13,15,16,18,22
Using P5 pins:
Servo mapping:
0 on P1-7 GPIO-4
1 on P1-11 GPIO-17
2 on P1-12 GPIO-18
3 on P1-13 GPIO-27
4 on P1-15 GPIO-22
5 on P1-16 GPIO-23
6 on P1-18 GPIO-24
7 on P1-22 GPIO-25
```

Setting up the Raspberry Pi

STEP 2: If everything is working ok, you'll be able to send the command

```
pi@raspberrypi ~/PiBits/ServoBlaster/user $ echo 0=50% > /dev/servoblaster
```

NOTE: This will send servo 0 (the one attached to GPIO 4) to 50% of its range. Changing to `echo 1=20% > /dev/servoblaster` will send servo 1 to 20% of its 0 → 180 degree range.

STEP 3: Use Python and something called Tkinter (which should be installed on your Pi already). Create a new file using your favorite file editor (it should be GVIM - it will make you more popular, stronger and better looking). I called it MeArm.py. Add the following code to it and save.

```
| #!/usr/bin/env python
from Tkinter import * #allows us to make a GUI with Tkinter
import os

root = Tk()
```


Setting up the Servos with Raspberry Pi

STEP 1: Use the following code to setup the servos using lists

```
SNums = [0,1,2,3] #Numbers of the Servos we'll be using in
ServoBlaster
SName = ["waist","Left","Right","Claw"] #Names of Servos
AInis = [90,152,90,60] #Initial angle for Servos 0-3
AMins = [0,60,40,60] #Minimum angles for Servos 0-3
AMaxs = [180,165,180,180] #Maximum angles for Servos 0-3
ACurs = AInis #Current angles being set as the initial angles
Step = 5
for i in range(4):
    print(SNums[i],AInis[i],AMins[i],AMaxs[i],ACurs[i])

os.system('sudo /home/pi/PiBits/ServoBlaster/user/servod -idle-
timeout=2000') #This line is sent to command line to start the
servo controller

#inc listens for all key presses. On certain presses in the if
statements below it either calls a process to add or subtract
from the current servo angle.
def inc(event):
    print "pressed", repr(event.char)
    if repr(event.char) == "'a'":
        AAdd(0)
    if repr(event.char) == "'d'":
        ASub(0)
    if repr(event.char) == "'w'":
        AAdd(1)
    if repr(event.char) == "'s'":
        ASub(1)
    if repr(event.char) == "'j'":
        AAdd(2)
    if repr(event.char) == "'l'":
        ASub(2)
    if repr(event.char) == "'i'":
        AAdd(3)
    if repr(event.char) == "'k'":
        ASub(3)
def callback(event):
    frame.focus_set()
def AAdd(Servo):
    if ACurs[Servo] < AMaxs[Servo]:
        ACurs[Servo] = ACurs[Servo]+Step
    # micro = (1000 + (ACurs[Servo] * 5.555))
    micro = (1000 + (ACurs[Servo] * 8.3333))
```

```

print(ACurs[Servo],micro)
os.system("echo %d=%dus > /dev/servoblaster" %
(SNums[Servo],micro))
else:
print "Max Angle Reached",SName[Servo],"Servo"
def ASub(Servo):
if ACurs[Servo] > AMins[Servo]:
ACurs[Servo] = ACurs[Servo]-Step
# micro = (1000 + (ACurs[Servo] * 5.555))
micro = (1000 + (ACurs[Servo] * 8.3333))
print(ACurs[Servo],micro)
os.system("echo %d=%dus > /dev/servoblaster" %
(SNums[Servo],micro))
else:
print "Min Angle Reached",SName[Servo],"Servo"
frame = Frame(root, width=500, height=300)
boxtext = Label(root, text="Click this box for keyboard command
of the MeArm. Use the a d s w j l i and k keys for control.")
boxtext.pack()
frame.bind("<Key>",inc)
frame.bind("<Button-1>", callback)
frame.pack()
root.mainloop()

```

STEP 2: Using a terminal or the command line type the following

```
pi@raspberrypi ~ $ python MeArm.py
```

STEP 3: All being well you should now have a pop-up box that tells you to click inside it to control your MeArm!

Calibrating the Servo Motors

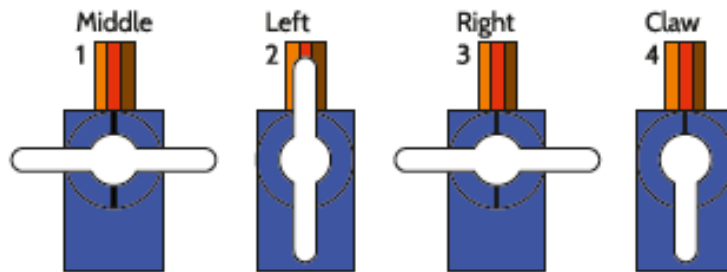
STEP 1: Follow the Setup steps for either Arduino or Raspberry Pi

STEP 2: Using a marker, make a line across the Servo Motor Body and spindle (as shown)



STEP 3: Attach the Servo Arms as shown below using a small machine screw from the servo fixings packs. Leave these attached until you are required to use them in building the MeArm. Be careful not to move the position of the spindle now that they are calibrated.

NOTE: These images are not to scale. You will have two large servos and two small servos.



STEP 4: Move onto Building the MeArm

Building the MeArm

Please Note: The MeArm relies on self tapping screws, do not over tighten or you could strip the hole or break the part.

Do not attempt to force parts together as excessive force can break them.







Parts

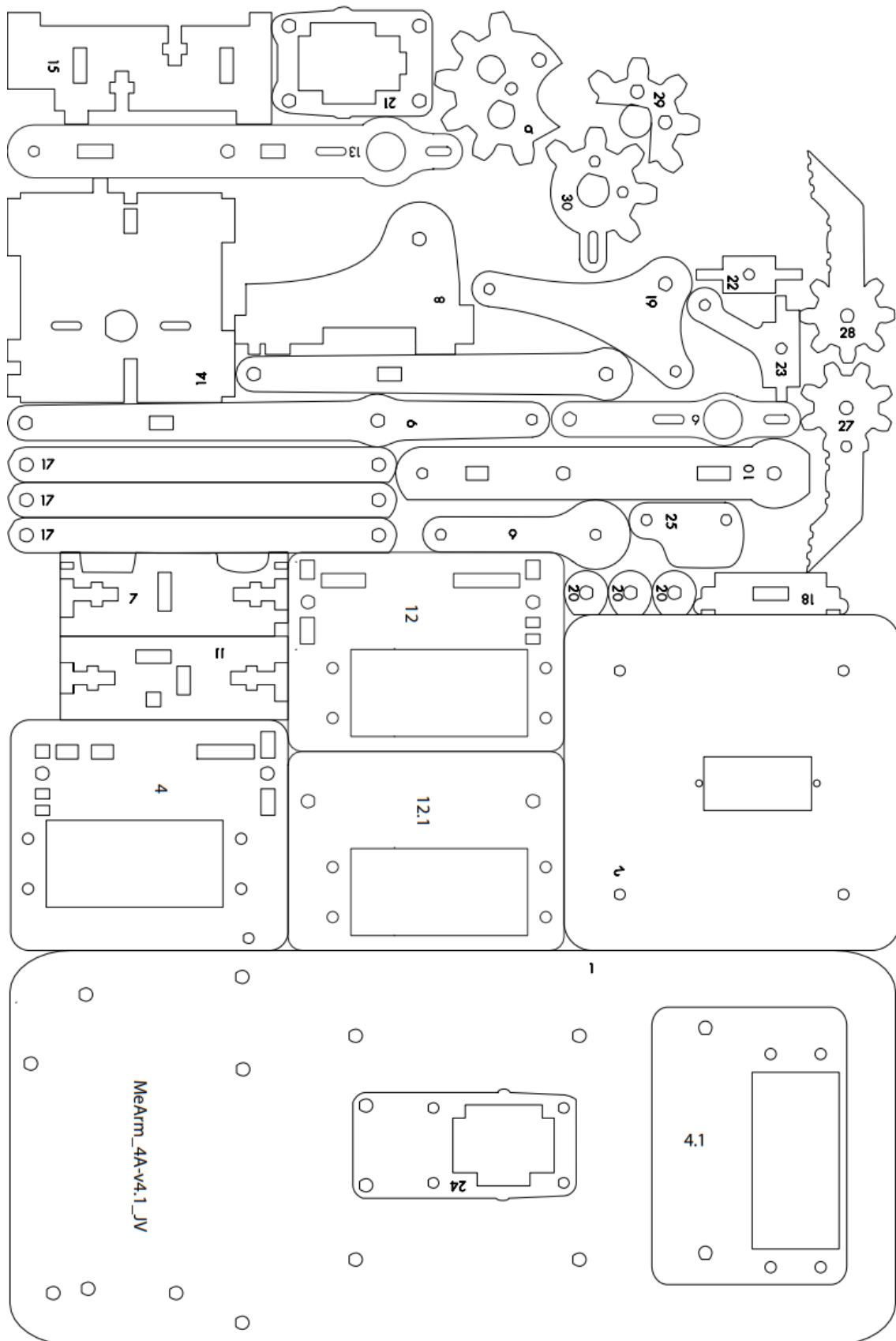
Building the MeArm Requires the following fasteners. These profiles are actual size (if printed A5 sized paper).

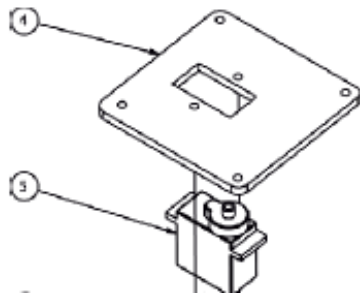
The acrylic parts list is shown for reference. Small differences may occur as the product is improved over time.

Construction requires a small cross head screwdriver and 2.5mm hex key, which is supplied.

Additional screws for the servo arms are included in the servo packs.

10x – M3 Nut	
6x – M3 x 6	
15x – M3 x 8	
3x – M3 x 10	
8x – M3 x 12	
4x – M3 x 20	

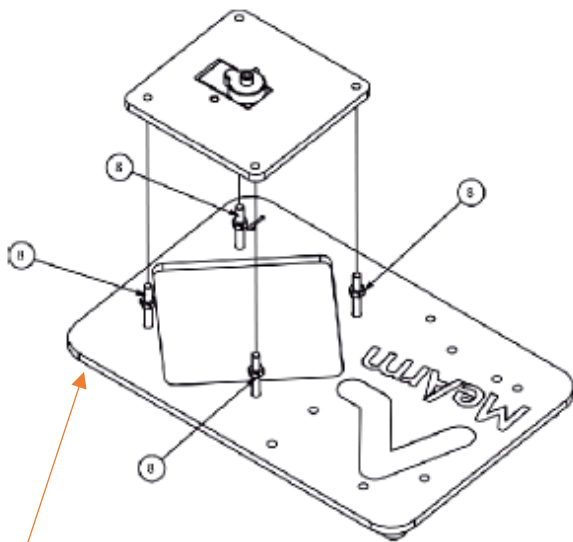




Step 1: Insert MIDDLE Servo Motor from the top side of Pivot Servo Plate

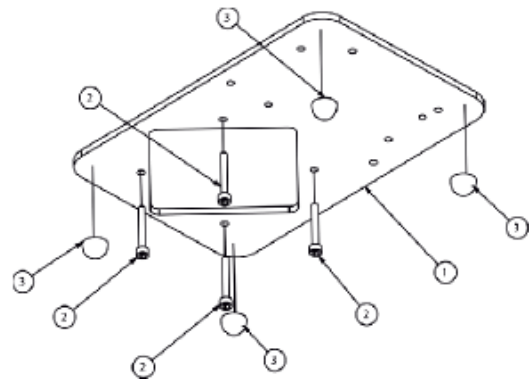
Step 2 – 4: Insert 2 x Servo Mount Screws shown as item 7

PARTS LIST			
ITEM	PART NAME	QTY	PART NUMBER
1	Base Plate	1	1
2	M3 x 20mm Screw	4	
3	Pivot Servo Plate	1	2
4	Servo Motor	1	
5	M3 x 8mm Screw	2	
6	M3 Nut	4	

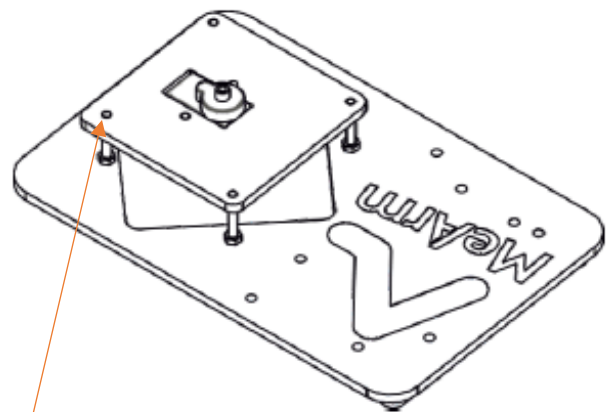


Step 6. Thread (1x) Nut half way down each of the (4x) 20mm Screws.
Step 7. Screw each of the 20mm Screws into the mating holes on the Pivot Servo Plate until the end of the screw is flush with the top of the plate

NOTE: This Base Plate has been reconfigured under the same PART NUMBER

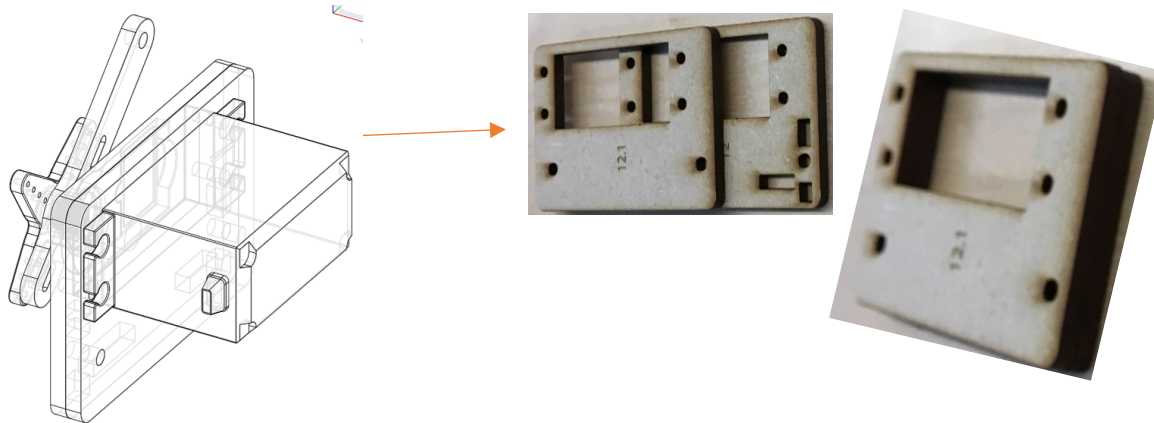


Step 5. Insert (4) 20mm Screw though from underside of Base Plate



Step 8. Tighten the Nuts rest of the way down (finger tight is fine).

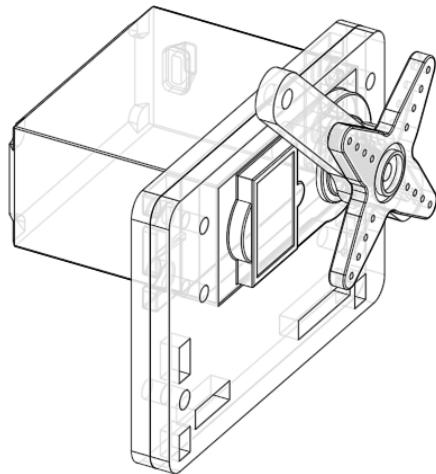
NOTE: Be sure the screw ends sit flush with the top of the Pivot Servo Plate or even slightly lower (to catching on the Servo Base)



Note: The above figure is comprised of Parts 12 and 12.1.

Step 9 – 11: Use Left Arm Servo plates 12 and 12.1 and mount the Servo to the rectangular hole. To do this, use 4x M3 x 10mm.

The two screws mounting the plate to the structure are M3 x 12mm with 2x M3 nuts.



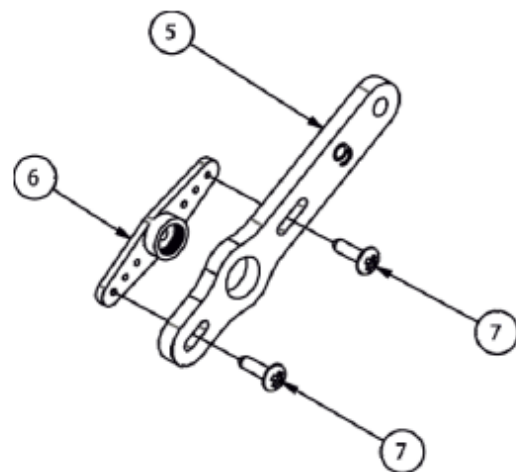
Note: The image above is made up of two brackets stacked together using PART 12 and PART 12.1.

Step 13, Attach the Long Servo Arm Extension assembly to the calibrated LEFT servo motor (using the Brains Board Guide as a reference for position).

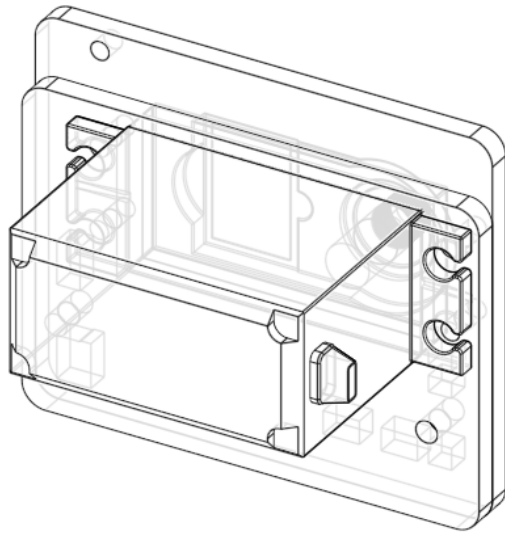
Step 14. Use the machine screw from the servo pack to secure in position.

Step 15, (Optional) Power the Arduino and check that the Long Servo Arm Extension assembly points to approximately 9 o'clock.

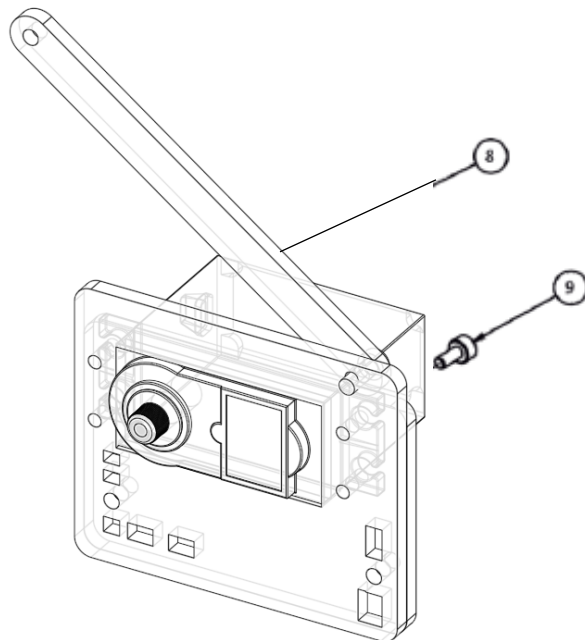
PARTS LIST			
ITEM	PART NAME	QTY	PART NUMBER
1	Left Arm Servo Plate	2	12 and 12.1
2	M3 x 8mm Screw	2	
3	Large S3003 Servo	1	
4	Large Servo Arm	1	6
5	Servo Double Arm	1	
6	Servo Mount Screw	2	
7	Servo Screw	1	



Step 12. Attach the Servo Arm to the Long Servo Arm Extension using the sharp screws in the servo pack. These will self tap with a little pressure.



Note: The above figure shows Parts 4 and 4.1 combined
Step 16 – 17: Use Right Arm Servo plates 4 and 4.1 and mount the Servo to the rectangular hole. To do this, use 4x M3 x 10mm. The two screws mounting the plate to the structure are M3 x 12mm with 2x M3 nuts.

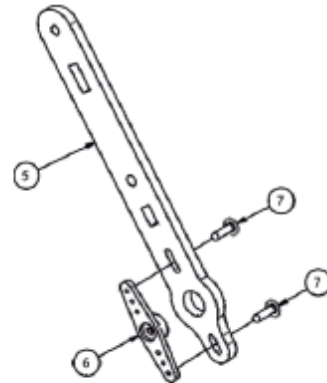


Step 19. Insert 6mm Screw into Parallel linkage and attach to Right Arm Servo Plate

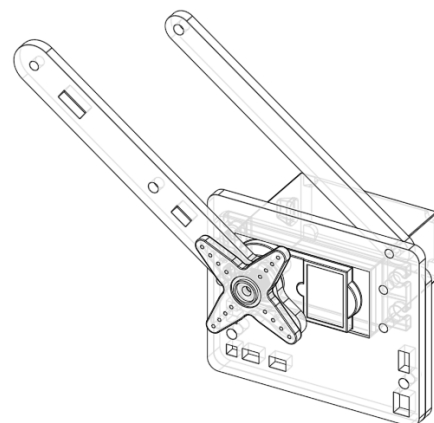
Note: The image above comprised of PART 4 and PART 4.1.

PARTS LIST

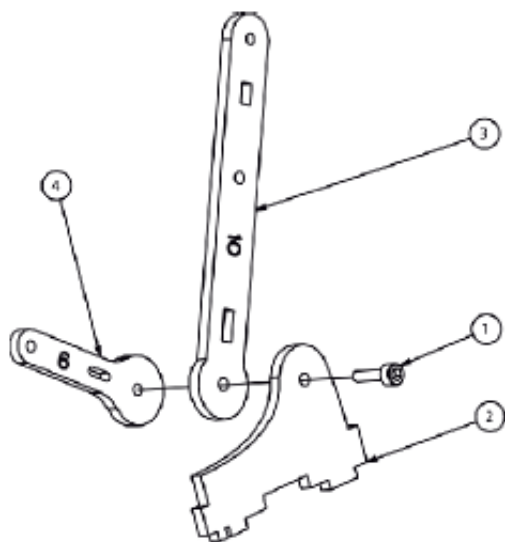
ITEM	PART NAME	QTY	PART NUMBER
1	Right Arm Servo Plate	2	4 and 4.1
2	M3 x 8mm Screw	2	
3	Large S3003 Servo	1	
4	Right Arm Base Joint	1	13
5	Servo Double Arm	1	
6	Servo Mount Screw	2	
7	Parallel Linkage	1	
8	M3 x 6mm Screw	1	
10	Servo Screw	1	



Step 18. Attach the Servo Arm to the Right Arm Base Joint using the sharp screws in the servo pack. These will self tap with a little pressure.

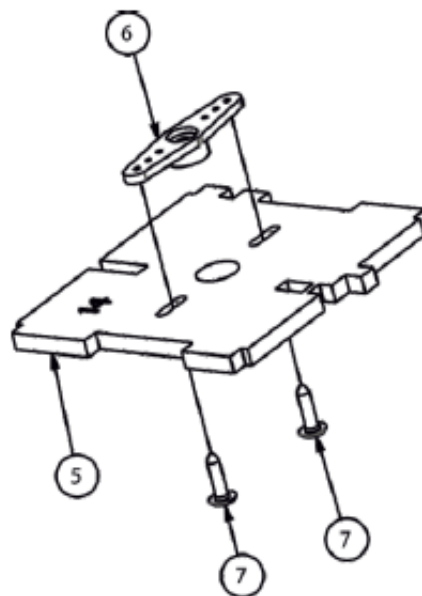


Step 20. Attach the Right Arm Base Joint assembly to the calibrated RIGHT servo motor (using the Brains Board Guide as a reference for position).
Step 21. Use the machine screw from the servo pack to secure in position



Step 23. Fix parts shown together using 10mm Screw. These should move freely. This assembly is known as the pig.

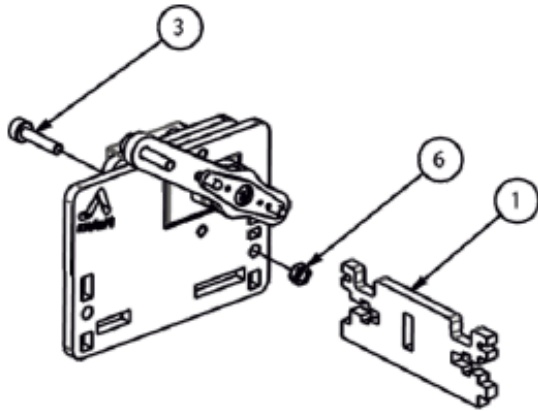
RTS LIST			
ITEM	PART NAME	QTY	
1	M3 x 10mm Screw	1	
2	Left Arm Mount Tablet	1	8
3	Left Arm Base Joint	1	10
4	Short Servo Arm	1	9
5	Arm Bottom Plate	1	14
6	Servo Double Arm	1	
7	Servo Mount Screw	2	



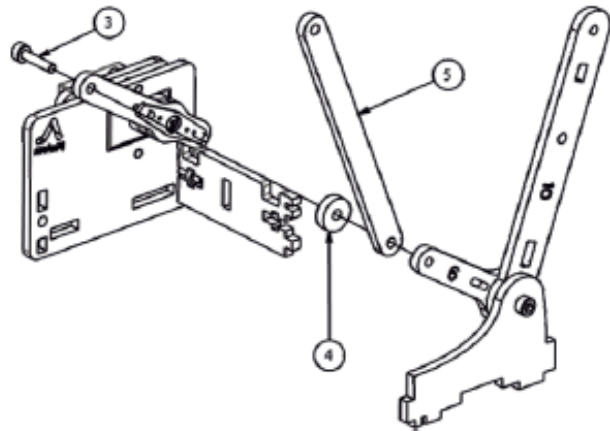
Step 24. Attach the Servo Arm to the Arm Bottom Plate using the sharp screws in the servo pack. These will self tap with a little pressure.

Step 25. Insert the machine screw to attach to the servo later into the Servo Arm as insertion later is a little tricky.

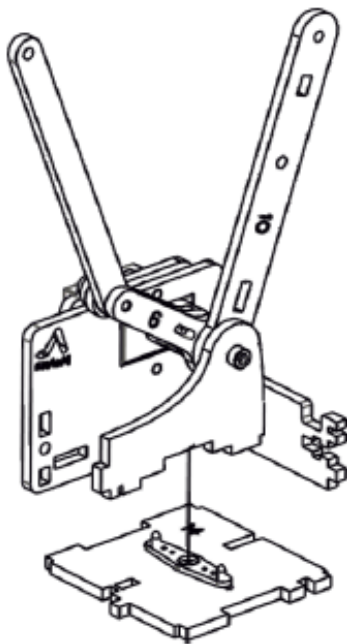
ITEM	PART NAME	QTY	PART NUMBER
1	Front Arm Base Cross Member	1	7
2	Back Arm Base Cross Member	1	11
3	M3 x 12mm Screw	3	
4	Spacer	1	20
5	Parallel Linkage	1	17
6	M3 Nut	2	



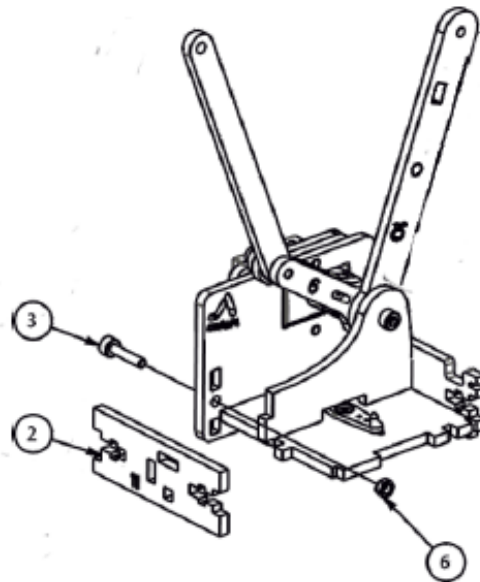
Step 26. Insert (1) 12mm Screw, add nut and Front Arm Base Cross Member and tighten. Stop before you break it.



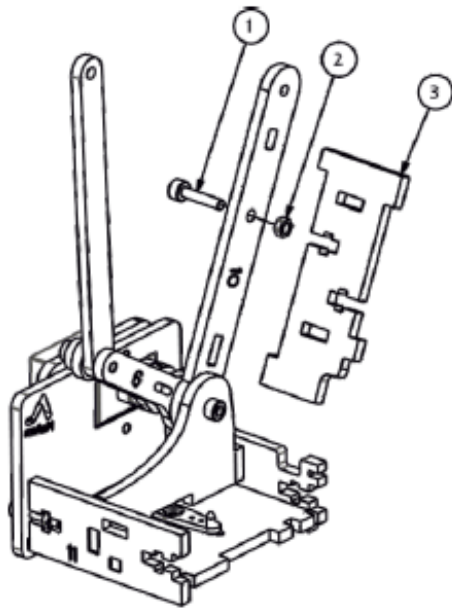
Step 27. Insert the Pig into the Front Arm Base Cross Member and using 12mm Screw attach the Pig to the Long Servo Arm Extension



Step 28. Insert the Arm Bottom Plate into the Left Arm Servo Plate at a diagonal angle and lever into place on the Pig Section.

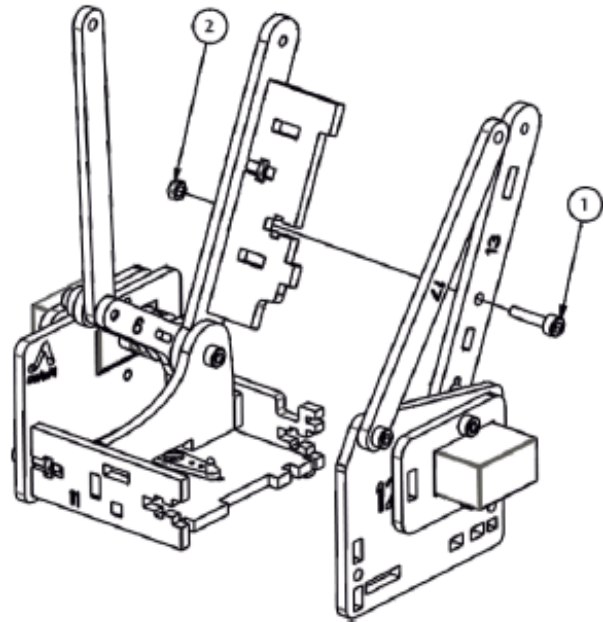


Step 29. Insert 12mm screw, add nut and Rear Arm Base Cross Member and lever carefully onto the rear of the Pig and the Arm Bottom Plate. Leave this screw slightly loose to aid attaching this assembly to the base later.



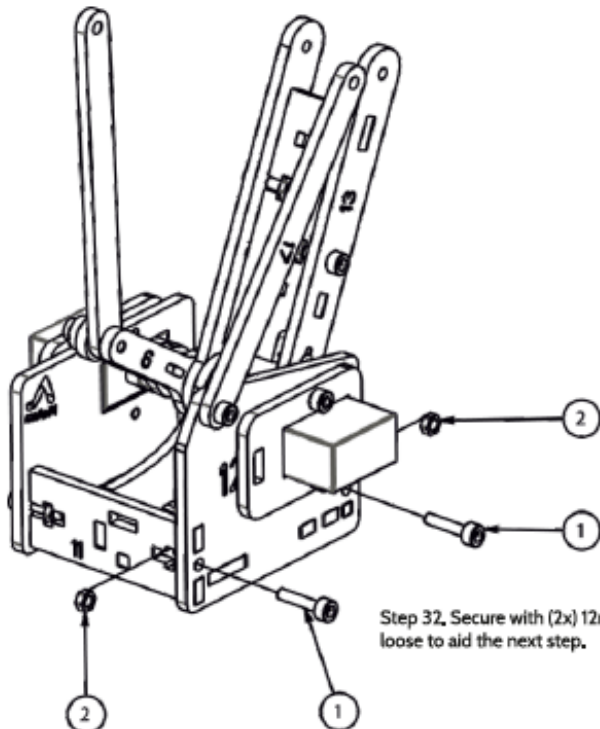
Step 30. Using 12mm Screw and Nut attach Main Arm Cross Web

RTS LIST			
ITEM	PART NAME	QTY	PART NUMBER
1	M3 x 12mm Screw	4	
2	M3 Nut	4	
3	Main Arm Cross Web	1	15



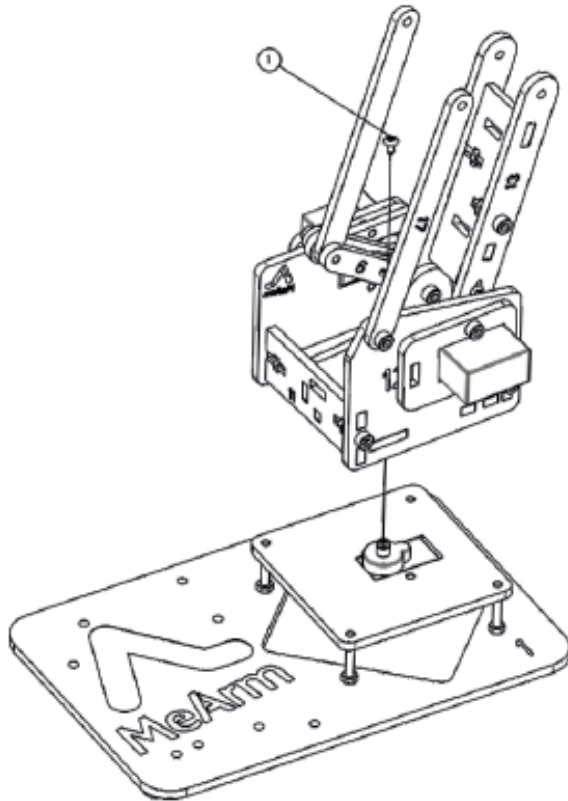
Step 31. Bring the right and left assemblies together securing with a 12mm Screw and Nut in the Main Arm Cross Web.

This part is tricky. Be gentle and have some patience. Small adjustments of the parts will allow it all to push together.



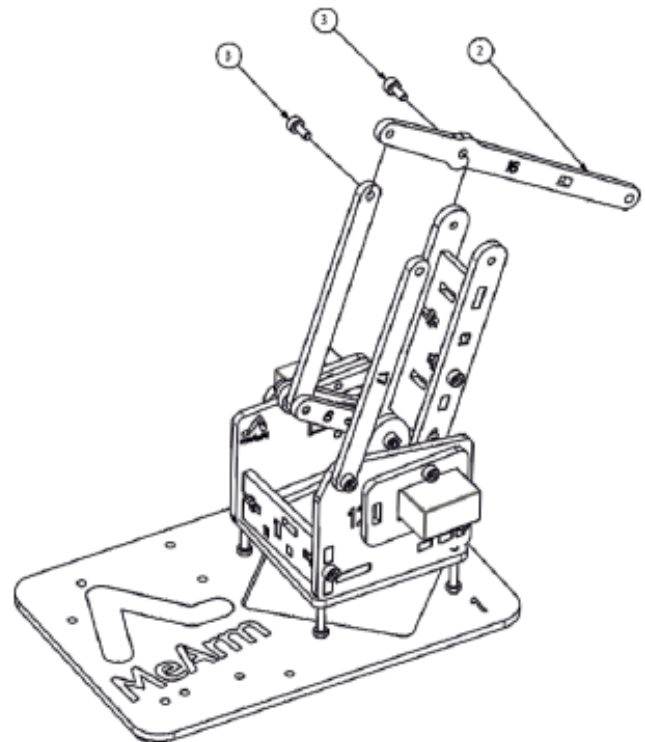
Step 32, Secure with (2x) 12mm Screws and (2x) Nuts, Leave a little loose to aid the next step.

ITEM	PART NAME	QTY	PART NUMBER
1	Servo Screw	1	
2	Left Wrist Joint	1	16
3	M3 x 6mm Screw	2	



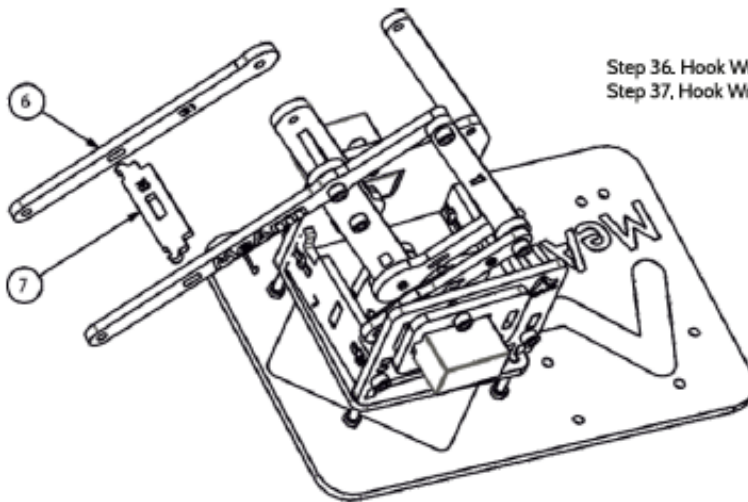
Step 33. Attach Arm Assembly to Base using small machine screw from Servo Pack, which should have been left in the Servo Arm from Step 25.

Step 34. (Optional) Power Arduino and check that the assembly sits at approximately 12 o'clock relative to the base.

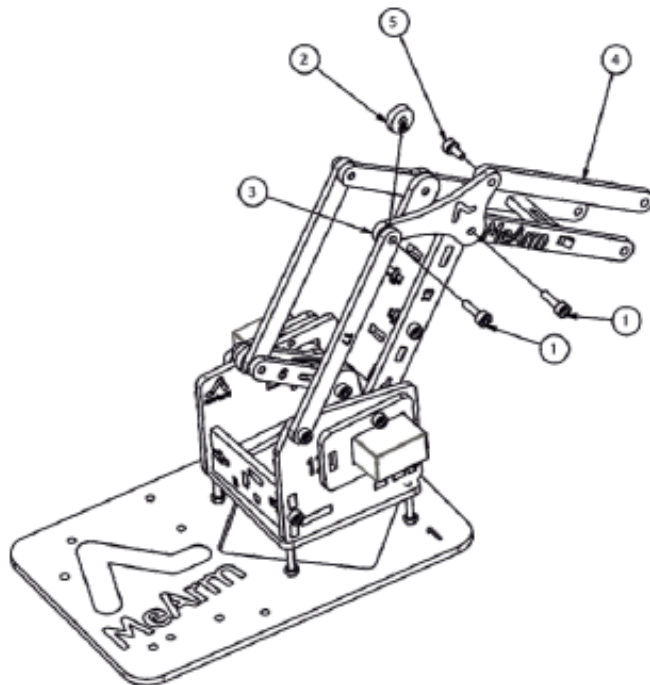


Step 35. Attach using (2) 6mm Screws.

ITEM	PART NAME	QTY	PART NUMBER
1	M3 x 10mm Screw	2	
2	Spacer	1	20
3	Parallel Linkage Connector	1	
4	Parallel Linkage	1	17
5	M3 x 6mm Screw	1	
6	Right Wrist Joint	1	31
7	Wrist Joint Connector	1	18



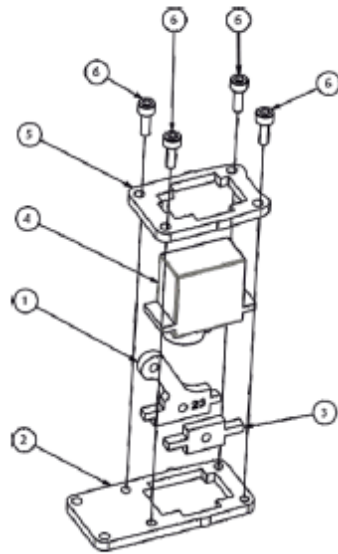
Step 36. Hook Wrist Joint Connector into Right Wrist Joint
 Step 37. Hook Wrist Joint Connector Assembly into Left Wrist Joint



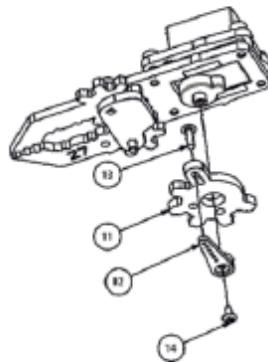
Step 38. Secure Right Wrist joint with Parallel Linkage Connector and a 10mm Screw

Step 39. Attach Parallel Linkage (from Right Servo Plate) to the Parallel Linkage Connector using 10mm Screw and a Spacer.

Step 40. Attach final Parallel Linkage to the Parallel Linkage connector using a 6mm screw.

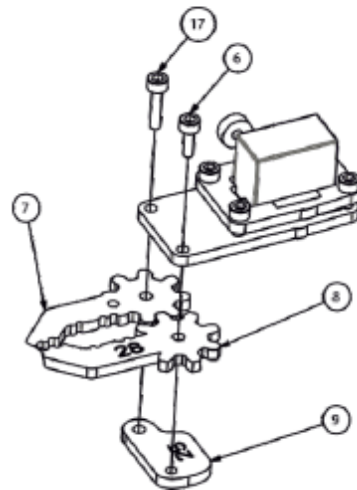


Step 41. Insert Servo into Clamp Top Servo Collar
 Step 42. Slide Right and Left Wrist Attachments into the Clamp Top Servo Collar as shown
 Step 43. Slide the Clamp Bottom Servo Mount over the Servo
 Step 44. Insert (4x) 8mm Screws into the Clamp Top Servo Collar and attach to the Clamp Bottom Servo Mount

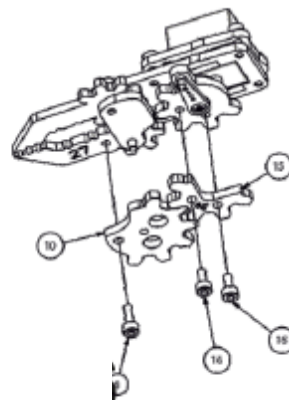


Step 46. Attach Servo Single Arm to Top Servo Gear and attach to calibrated CLAW servo (using the Brains Board Guide as reference).
 NOTE, On the calibrated CLAW Servo the Servo Arm assembly and longest lug should point towards 6 o'clock (Gripper at 12 o'clock)

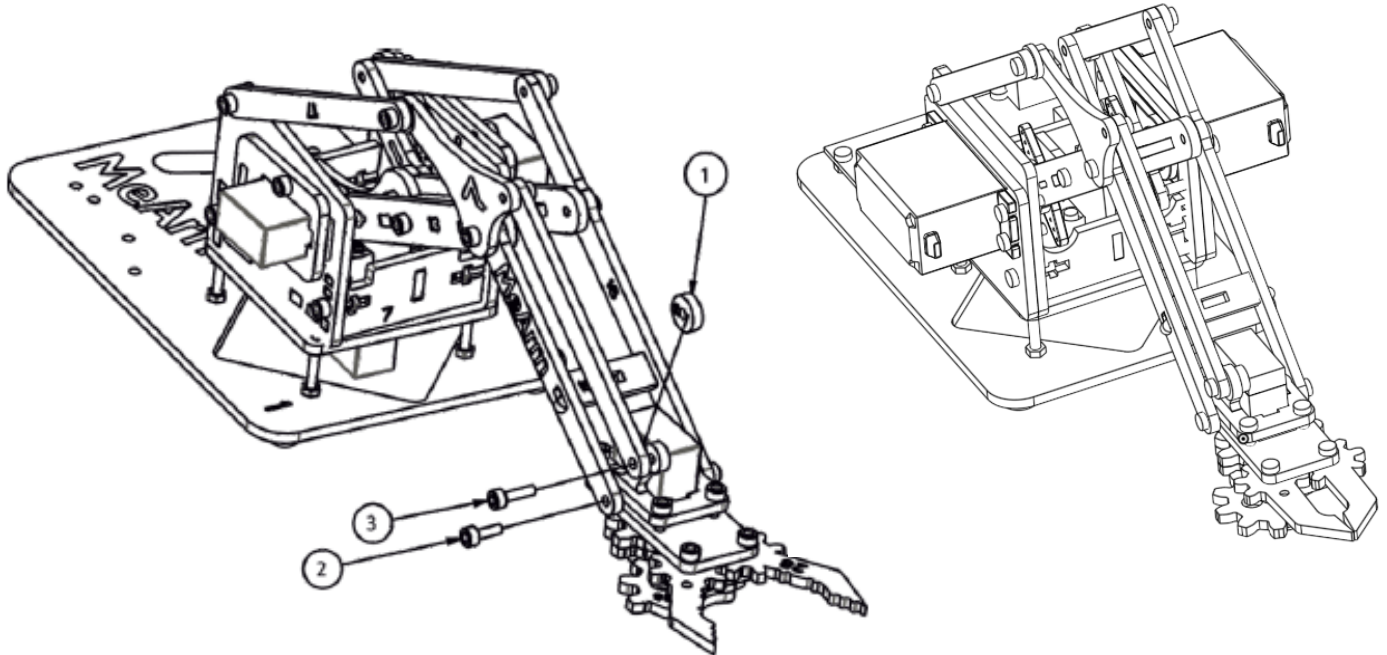
ITEM	PART NAME	QTY	PART NUMBER
1	Right Wrist Attachment	1	23
2	Clamp Bottom Servo Mount	1	24
3	Left Wrist Attachment	1	22
4	9 Gram Servo	1	
5	Clamp Top Servo Collar	1	21
6	M3 x 8mm Screw	6	
7	Right Gripper	1	27
8	Left Gripper	1	28
9	Gripper Plate	1	25
10	Gripper Gear	1	26
11	Top Servo Gear	1	30
12	Servo Single Arm	1	
13	Servo Mount Screw	1	
14	Servo Screw	1	
15	Bottom Servo Gear	1	29
16	M3 x 6mm Screw	2	
17	M3 x 12mm Screw	1	



Step 45. Attach Left Gripper with 8mm Screw into Gripper Plate
 Insert 12mm Screw to keep assembly in place (does not tighten yet)

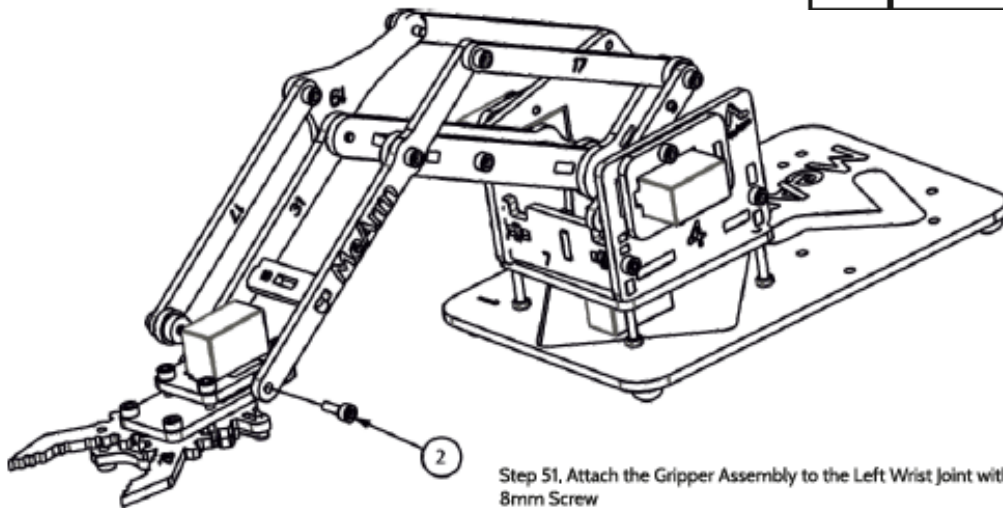


Step 47. Connect Bottom Servo Gear using 6mm Screws.
 Step 48. Attach Gripper Gear with 8mm Screw from underside and 12mm Screw (previously inserted from above). **IMPORTANT NOTE!** Jaws should be set open to 90 degrees (180 degrees to each other) when meshed with calibrated servo gear! **NOT AS SHOWN**

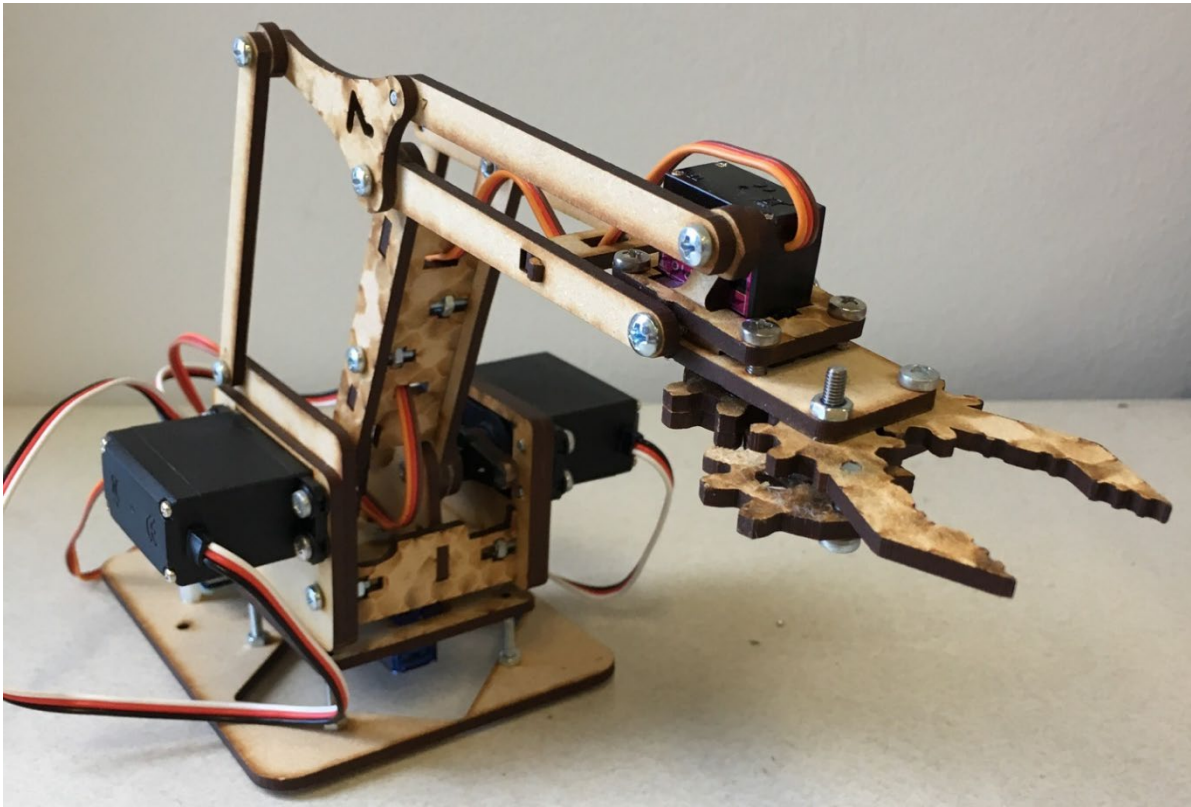


Step 49. Attach the Gripper Assembly to the Right Wrist Joint with 8mm Screw
 Step 50. Connect the Parallel Linkage with a 8mm Screw and Spacer

PARTS LIST			
ITEM	PART NAME	QTY	PART NUMBER
1	Spacer	1	20
2	M3 x 8mm Screw	3	



Step 51. Attach the Gripper Assembly to the Left Wrist Joint with 8mm Screw
 Step 52. Thread CLAW Servo cable through cable guides and attach Servo Extension to reach Arduino.
 Step 53. Attach Arduino to mounting holes in base plate.



Congratulations!

Now you've completed your MeArm! It's now ready to control with your Arduino. You can build your own controller or control directly via your Arduino. Code examples of both are available on our codebender page at <https://codebender.cc/user/MeArm> .

We'll be adding to this guide so please check back and in the mean time we hope to see you on our forums at forum.mearm.io

Many thanks for supporting the MeArm project!

Reference Material

- [1] <https://mime.co.uk/blog/2016/01/26/mearm-on-the-raspberry-pi/>
- [2] <https://learn.mime.co.uk/docs/control-your-mearm-from-raspberry-pi/>
- [3] https://learn.mime.co.uk/assets/docs/control-your-mearm-from-arduino/MeArm_v1.0_Manual_for_Arduino_v1.0.pdf
- [4] <https://www.instructables.com/id/MeArm-Robot-Arm-Your-Robot-V10/>

