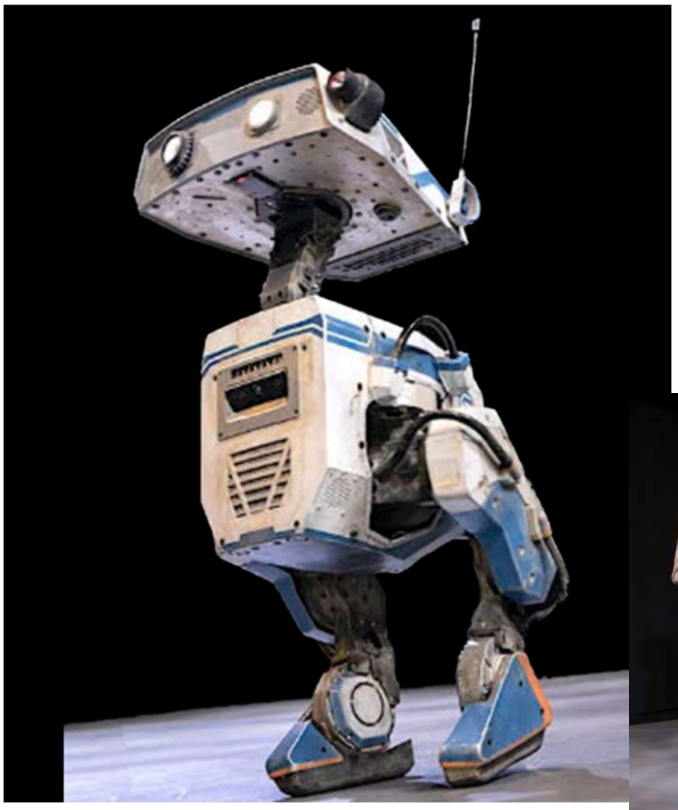
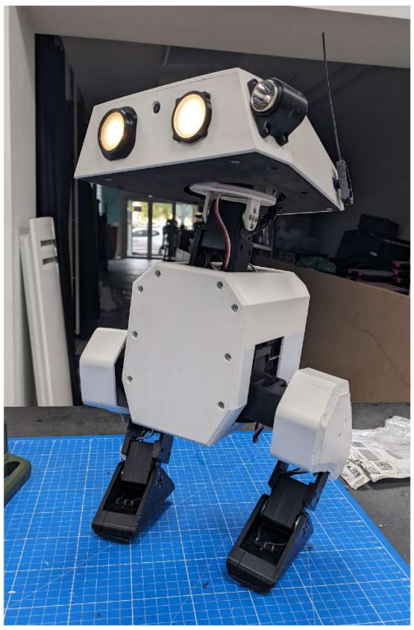
BDX Assembly Notes V1

Start 12-4-25:





Mission:

- To build a Companion Robot with WiFI connectivity
- To build a Plate Form to add to and learn from
- To be mobile, robot
- Duckling type Bi-ped
- Some sort of Joy Stick controller Xbox1 or Steam deck
- Animate movements
- To have AI type vision, and move to object x
- To have Alexa type voice interface plus AI ChatGPT link

- https://www.youtube.com/watch?v=16LuvR2CARA&ab_channel=CrunchLabs
- https://www.youtube.com/watch?v=7_LW7u-nk6Q&ab_channel=DisneyResearchHub
- https://youtu.be/XPI6HNrd7ml
- https://youtu.be/My0S-hsBC2M
- https://www.facebook.com/gmrhodes13/videos/1574771059862852/?idorvanity=1330064834544532
- FB: https://www.facebook.com/groups/1330064834544532/discussion/preview
- Discord: https://discord.gg/UtJZsgfQGe
- Github : https://github.com/apirrone/Open_Duck_Mini

BDX Mini Specifications:

Basic Start, as per the design, then my extras

- Total of 14 x 7.4V 19kg serial Servos
- Wave Share Servo Board
- RPi Zero 2 W
- 5amp Buck (Rpi)
- 25/50amp BEC (Servos)
- IMU BNO055 DFrobotics
- I2S Amp Audio output (R2D2)
- Auto Reset fuse
- Voltage display
- On/Off Switch
- XT60 Connector
- Li-po Battery
- 3 x Bearings
- 70 x m3 x 5 inserts

My Extras

- LCD Eyes
- Rpi5
- Microwave sensor – Det movement through walls
- Sonar Object avoidance - Ultra Sonics
- AI Camera
- Software Defined radio Receive – SDR
- Easy Wi-Fi Connection (embedded config page, browse to like Kia)
- reSpeaker with esp32 (Voice recognition and audio output)
- Chat GPT AI

Parts

BOM

https://docs.google.com/spreadsheets/d/1gq4iWWHEJVgAA_eemkTEsshXqrYlFxXAPwO515KpCJc/edit?fbclid=IwY2xjawJnvdNleHRuA2FlbQIxMAABHlXfITFoIASPVftpWyNDc4kmpx5yOoABXVeLFmbzP1Ezn754cJjJ7bYBixwG_aem_1k7gThmfu40zYomnexQ21A&gid=0#gid=0

16 x Actuators (includes 2 spare) $24ea	$322.85 usd	pay pal	Alibaba	in stock
2 x Servo Power PCB	$15.98 usd	pay pal	Waveshare	in stock
Pi Zero 2 W 1GHz quad core	$32.95	visa43.77	pbtech	in stock
UBEC-25A/50Amp 3-18S 80V	$64	visa	aliex	29-5-25
IMU BNO055 (DFrobotics)	$11.50	visa	aliex	29-5-25
I2S amp	$1.13	visa	aliex	29-5-25
Battery Meter	$11	visa	aliex	29-5-25
3 x Bearing 4.20ea	$12.6($23)	NZ Miniature bearing		29-5-25
70+ M3 x 5mm inserts 10/pack	$12.5	Gogo tronics NZ		in stock
Type C USB-C to Micro USB lead				
2 x Micro Sw				in stock
2 x 9g MG90 Servo				in stock
5amp Buck & PSU Bits				in Stock
Magnets for Bat Pod				in stock
Projector LED				in stock

My Extras

Sonar HC-SR04 x 2	$3.99ea	epartners
Microwave sensor RCWL-0516	$2.99	epartners
AI Pi Camera 12.3 MP Sony IMX500	$139	pbtech
Pi Zero Camera cable	$10	pbtech
ReSpecker Lite with ESP32S3	$58	seeed studio - nzd
Software Defined Radio	$	Ali

Supplies Links

Actuators

https://www.alibaba.com/product-detail/Robot-Servo-STS3215-7-4V-19kg_1600052037414.html?spm=a2700.galleryofferlist.normal_offer.d_title.26cb13a0on0spo

STS3215 7.4V 19kg.cm C001 Magnetic Encoder 360 Double Shaft Dual Axis TTL Multi Dual Mode Smart Serial Bus Servo NZD $23.97

Specs for above not found, but Waveshare STS3215 HS 20Kg has these specs

Voltage:	5 - 8.4 volts
Stall current:	2.4amp
Standby:	250mA
Size :	45x25x36.5

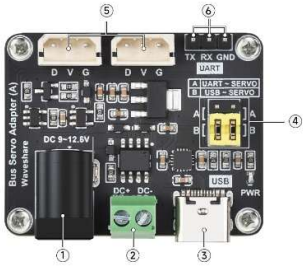
Therefore 14 Actuators x 250mA = 3.5Amp at rest. Plus RPi and other bits.

14 x 2.4Amp = 33Amp If all stalled.

Need to test this before final Buck choice. What is ideal current of all electronics?

Servo Drive usb type C

https://www.waveshare.com/bus-servo-adapter-a.htm

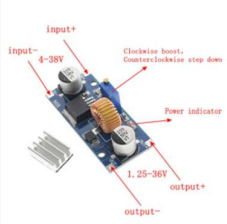


Pi Zero

https://www.pbtech.co.nz/product/SEVRBP0368/Raspberry-Pi-Zero-2-W-with-Soldered-Male-Header-1G

5amp Buck

https://epartners.co.nz/collections/power-modules-dc-converters?srsltid=AfmBOor-sGhwIkbYTm8gYA2_flamaC9nYuehv7Ziv5Ltd5Sl86CmevDA

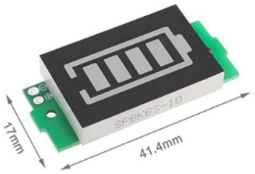


UBEC-25A/50Amp 3-18S 80V

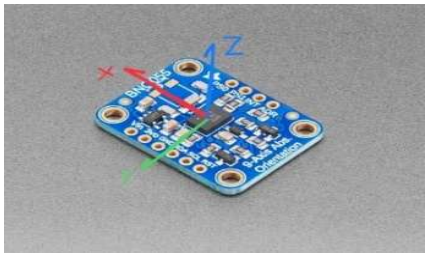
https://www.rchobbies.co.nz/hobbywing-30606000-ubec-25a-3-18s-80v/

Battery Meter

https://www.amazon.co.uk/Battery-Capacity-Indicator-Module-Compatible/dp/B0B1HL1VKV



IMU GY-BNO055 9 Axis Gyroscope Module Board Acceleration Sensor

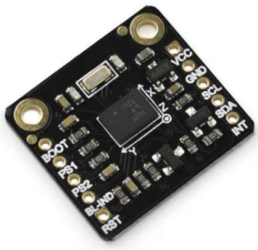


https://www.aliexpress.com/item/1005005499172228.html?spm=a2g0o.productlist.main.5.25f4ZlLTZlLTiA&algo_pvid=c21698a1-3e39-40f0-9dc3-6052789cec54&algo_exp_id=c21698a1-3e39-40f0-9dc3-6052789cec54-4&pdp_ext_f=%7B%22order%22%3A%22111%22%2C%22eval%22%3A%221%22%7D&pdp_npi=4%40dis%21NZD%2112.36%2112.36%21%21%217.13%217.13%21%40212e520f17468388385524436e5453%21112000033321745339%21sea%21NZ%211616880998%21X&curPageLogUid=fIkV0tVK987i&utparam-url=scene%3Asearch%7Cquery_from%3A

NOT THIS ONE: Cheap supply, arrived dead.

This has default address 0x29, Gnd add pin to get 0x28.

Use DFRobotics one. Costs a little more but reliable, and works in BDX



https://www.dfrobot.com/product-2142.html

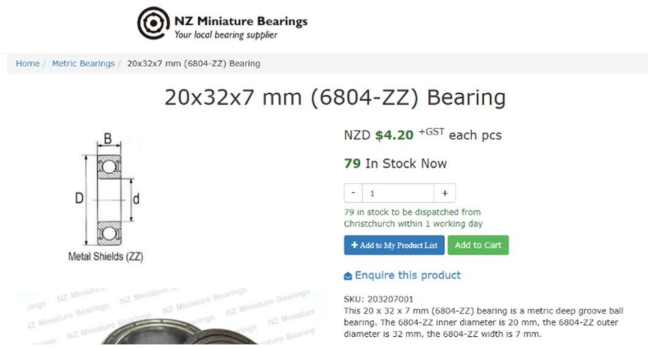
MAX 98357 I2S AMP

https://www.aliexpress.com/item/1005008589448305.html?src=google&pdp_npi=4%40dis!NZD!1.82!1.13!!!!!%40!12000045855045254!ppc!!!&src=google&albch=shopping&acnt=615-992-9880&isdl=y&slnk=&plac=&mtctp=&albbt=Google_7_shopping&aff_platform=google&aff_short_key=_oFgTQeV&gclsrc=aw.ds&&albagn=888888&&ds_e_adid=&ds_e_matchtype=&ds_e_device=c&ds_e_network=x&ds_e_product_group_id=&ds_e_product_id=en1005008589448305&ds_e_product_merchant_id=561337230&ds_e_product_country=NZ&ds_e_product_language=en&ds_e_product_channel=online&ds_e_product_store_id=&ds_url_v=2&albcp=22478631344&albag=&isSmbAutoCall=false&needSmbH

ouyi=false&gad_source=1&gad_campaignid=22478632454&gbraid=0AAAAA_TvRHpO5yhCOtrdPAIPYS-GyG0sb&gclid=CjwKCAjwz_bABhAGEiwAm-P8YZbpmW1idClZoJPsh5j8SpDdX3EpPHZvzU-Qi_oz3Td-CutAiGGxtBoCrWgQAvD_BwE

Bearing

https://www.nzminiaturebearings.co.nz/product/20x32x7-mm-6804-zz-bearing.html



M3 Inserts

https://sparks.gogo.co.nz/catalog/Hardware-270/Fasteners-and-Washers-296/Metric-Thread-Inserts-226/M3-3mm-Thread-Insert-Nut-Extra-Grip-Centering-Heat-Set-10-Inserts--Pack-903.html?srsltid=AfmBOoqPZqb_lpVytuu0jjyiusd9JpS017JHsY19FI3IkixYJB3i60y4

M3 (3mm) Thread Insert Nut Extra Grip Centering Heat Set 10 Inserts / Pack

$2.50

Choose Size 3mm Long

NZ Stocked Ready To Ship
Buy now, estimated **NZ urban delivery** **Monday / Tuesday**.
Rural adds 1-2 days.

Price	$2.50 Per Bag of 10
In Stock	Approx 9 Bags of 10

Microwave Sensor RCWL-0516

https://epartners.co.nz/products/ga2004?_pos=1&_psq=Microwave+sensor+RCWL-0516&_ss=e&_v=1.0

Sonar

https://epartners.co.nz/products/vx2004?_pos=1&_sid=f093761f6&_ss=r

TFT LCD - HD IPS Color 240x240 1.28 inch round (eye)

https://www.youtube.com/watch?v=pmCc7z_Mi8I&ab_channel=TheLastOutpostWorkshop

Epartners $17.99

Pi Camera (PBtech)

https://www.pbtech.co.nz/product/SEVRBP0544/Raspberry-Pi-5-Official-AI-Camera-123-MP-Sony-IMX5

Raspberry Pi 5 Official AI Camera 12.3 MP Sony IMX500 Intelligent Vision Sensor, 7.857 mm Sensor Size, Integrated RP2040 $133.85

https://www.pbtech.co.nz/product/SEVRBP0413/Raspberry-Pi-Accessories-FPC-Cable-for-Camera-15pi $11

ReSpeaker Lite

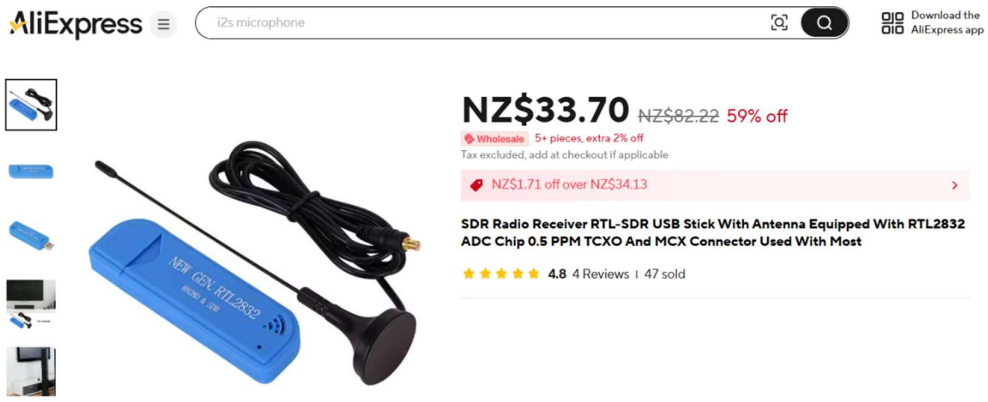
https://www.seeedstudio.com/ReSpeaker-Lite-Voice-Assistant-Kit-p-5929.html

https://wiki.seeedstudio.com/respeaker_lite_pi5/

https://www.hackster.io/idreams/build-your-own-amazon-echo-using-a-rpi-and-respeaker-hat-7f44a0

SDR

https://www.aliexpress.com/item/1005006007202943.html?src=google&pdp_npi=4%40dis%21NZD%2170.74%2129.59%21%21%21%21%21%40%211200000352941269 73%21ppc%21%21%21&src=google&albch=shopping&acnt=298-731-3000&isdl=y&slnk=&plac=&mtctp=&albbt=Google_7_shopping&aff_platform=google&aff_short_key=UneMJZVf&gclsrc=aw.ds&albagn=888888&ds_e_adid=&ds_e_matchtype=&ds_e_device=c&ds_e_network=x&ds_e_product_group_id=&ds_e_product_id=en1005006007202943&ds_e_product_merchant_id=109375829&ds_e_product_country=NZ&ds_e_product_language=en&ds_e_product_channel=online&ds_e_product_store_id=&ds_url_v=2&albcp=21627925371&albag=&isSmbAutoCall=false&needSmbHouyi=false&gad_source=1&gclid=Cj0KCQjw2N2_BhCAARIsAK4pEkWgIQIjw0H-Vy4fK03qxy5wEGTLdBWWtH2Xiay8GRUhjvRNdtLs5dsaAqV8EALw_wcB&gatewayAdapt=glo2fra



Prints

V2 Frame (github)

-	2 x foot_top.stl	Y	DONE	
-	2 x foot_side.stl	Y	DONE	
-	2 x foot_bottom_pla.stl	B	DONE	
-	2 x foot_bottom_tpu.stl x 2 (TPU)	B	DONE Proto Putty	
-	4 x knee_to_ankle_left_sheet.stl	B	DONE	
-	4 x knee_to_ankle_right_sheet.stl	B	DONE	
-	4 x leg_spacer.stl	W	DONE	
-	1 x left_roll_to_pitch.stl	Y	DONE	
-	1 x right_roll_to_pitch.stl	Y	DONE	
-	2 x roll_motor_bottom.stl	W	DONE	
-	2 x roll_motor_top.stl	W	DONE	
-	1 x trunk_bottom.stl	B	DONE	
-	1 x trunk_top.stl	B	DONE	
-	1 x neck_left_sheet.stl	B	DONE	
-	1 x neck_right_sheet.stl	B	DONE	
-	1 x head_pitch_to_yaw.stl	B	DONE	
-	1 x head_yaw_to_roll.stl	W	DONE	
-	1 x head_roll_mount.stl	B	DONE (V2 Head file)	

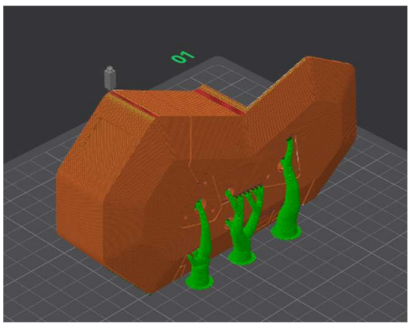
V2 Baseplate _Head_FullV1.step

-	1x Top2	W	DONE	
-	1 x LHS	W	DONE	
-	1 x RHS	W	DONE	
-	1 X face	W	DONE	
-	1 x base rear	B	DONE	
-	1 x base front	B	DONE	
-	1 x L eye	Y	DONE	
-	1 x L eye lens	B	DONE	
-	1 x R eye	Y	DONE	
-	1 x R eye lens	B	DONE	
-	1 x Bearing Housing	G	DONE	
-	1 x Servo top	G	DONE	
-	1 x Servo side 1	G	DONE	
-	1 x Servo Side 2	G	DONE	
-	1 x Spk Mount	G	DONE	
-	1 x Cable Port	W	DONE	
-	1 x left_antenna_holder.stl	Y	DONE	
-	1 x right_antenna_holder.stl	Y	DONE	
-	1 x Projector Cover.stl	B	DONE	

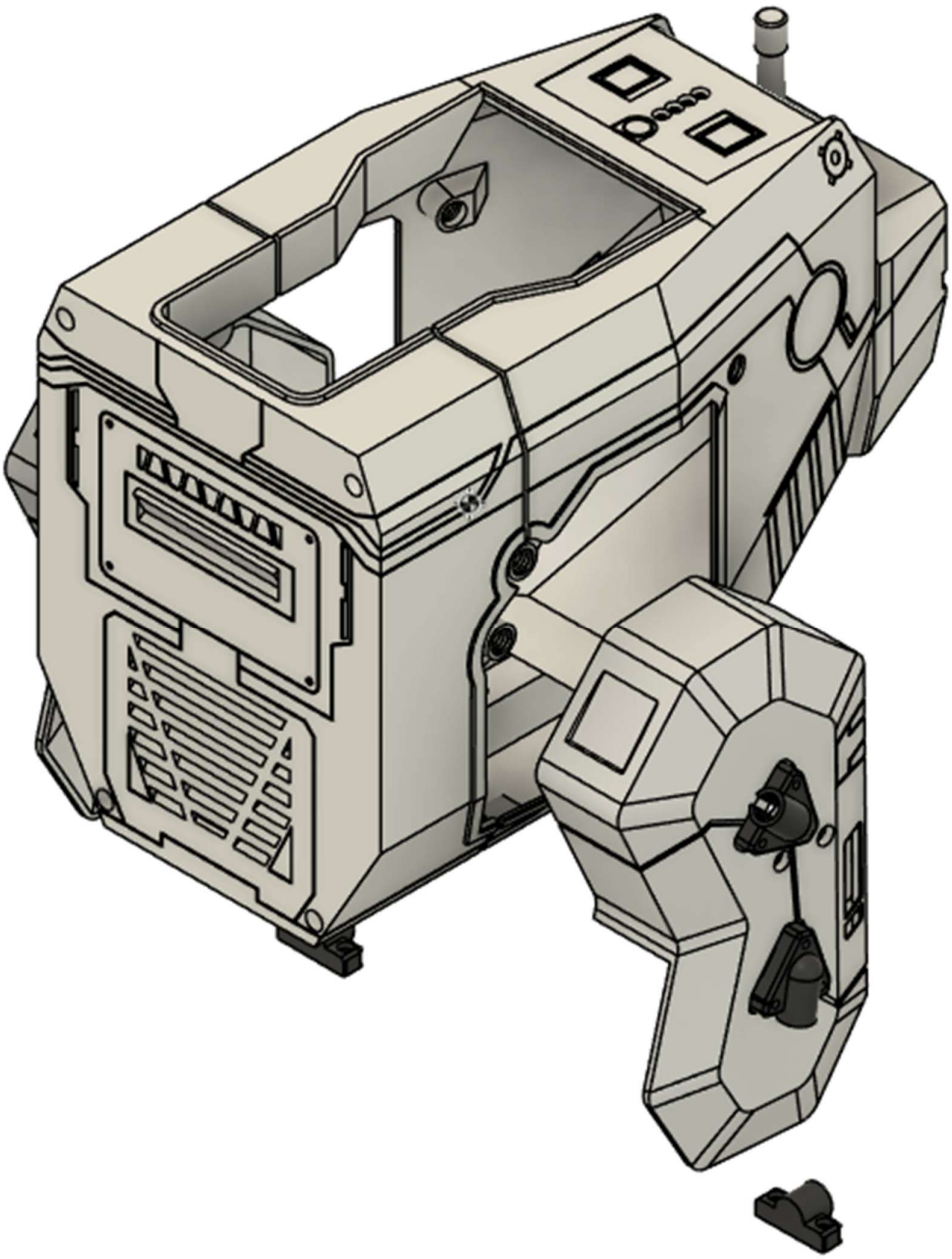
V2 Parks Skins (Mr B Pattrion)

-	1 x Rear cover - Temp base	W	DONE	3hr
-	1 x front_cover.stl	W		10
-	1 x rear_cover.stl	W		12 hr
-	1 x Battery Pod	W	DONE	3.5
-	1 x left_leg cover.stl	W	DONE	4.5
-	1 x right_leg cover.stl	W	DONE	4.5
-	2 x Foot Plugs	G	DONE	10m
-	4 x Upper foot plugs	G	DONE	14m

-	1 x Butt Plug	y	DONE	8m



Leg covers print better in this orientation



Parks Skins for V2 frame, from MrB Patreon

Foot Pad Mould

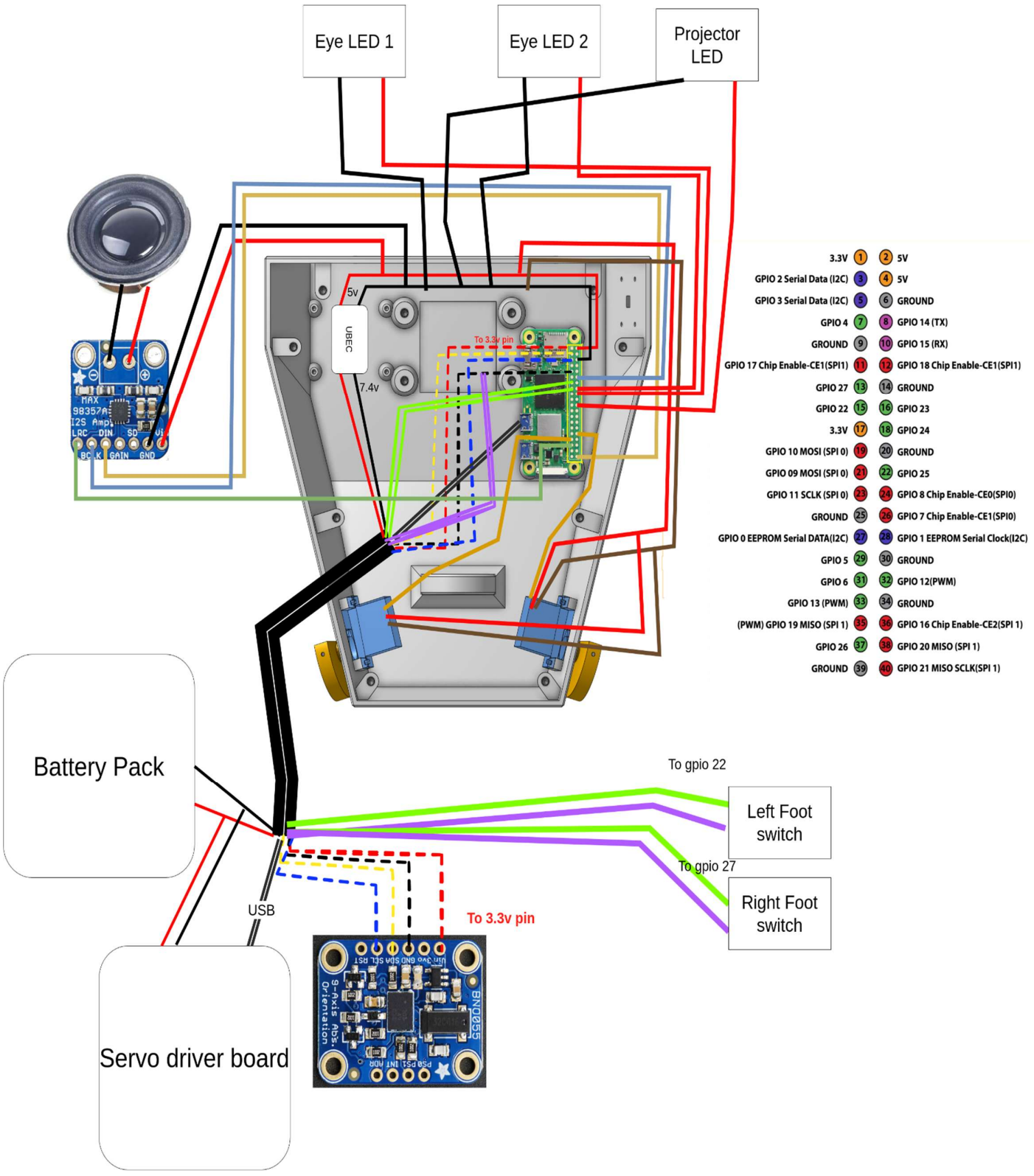
https://www.youtube.com/watch?v=7fwytA5r2Mw&t=127s&ab_channel=TKOR

Silicon + Corn flour + food colouring

Food colouring should be good chemical one, not cheap water base

Takes days to dry, do not disturb it, let dry in mould (5 days)

Wiring


Eye LED 1
Eye LED 2
Projector LED
5v
UBEC
7.4v
To 3.3v pin
3.3V 1 2 5V
GPIO 2 Serial Data (I2C) 3 4 5V
GPIO 3 Serial Data (I2C) 5 6 GROUND
GPIO 4 7 8 GPIO 14 (TX)
GROUND 9 10 GPIO 15 (RX)
GPIO 17 Chip Enable-CE1(SPI1) 11 12 GPIO 18 Chip Enable-CE1(SPI1)
GPIO 27 13 14 GROUND
GPIO 22 15 16 GPIO 23
3.3V 17 18 GPIO 24
GPIO 10 MOSI (SPI 0) 19 20 GROUND
GPIO 09 MOSI (SPI 0) 21 22 GPIO 25
GPIO 11 SCLK (SPI 0) 23 24 GPIO 8 Chip Enable-CE0(SPI0)
GROUND 25 26 GPIO 7 Chip Enable-CE1(SPI0)
GPIO 0 EEPROM Serial DATA(I2C) 27 28 GPIO 1 EEPROM Serial Clock(I2C)
GPIO 5 29 30 GROUND
GPIO 6 31 32 GPIO 12(PWM)
GPIO 13 (PWM) 33 34 GROUND
(PWM) GPIO 19 MISO (SPI 1) 35 36 GPIO 16 Chip Enable-CE2(SPI 1)
GPIO 26 37 38 GPIO 20 MISO (SPI 1)
GROUND 39 40 GPIO 21 MISO SCLK(SPI 1)
Battery Pack
To gpio 22
Left Foot switch
To gpio 27
Right Foot switch
USB
To 3.3v pin
Servo driver board

Neck Wires

1. Blk 0.6 – Gnd
2. Red 0.6 - +7.4v Servos
3. Wht 0.4 – Servo PWM sig from Waveshare pcb

4. Blk USB Cable - Servo Coms
5. Red_Mouse – Right Foot Switch GPIO 27
6. Wht_Mouse – Left Foot Switch GPIO 22
7. Org/Wht – I2C SCL for IMU GPIO 3
8. Grn/Wht – I2C SDA for IMU GPIO 2
9. Org 0.2/0.4 – +5v from Buck
10. Org - +3v for IMU
11. Grn - GND for IMU

I2C bus signal cross-talk and electromagnetic compatibility can be greatly improved by proper twisting of the cable wires. Twisted pairs is especially important for sensor wiring. 10 turns for each pair SCL/+5V and SDA/GND per 30cm cable length.

Cut up Cat5 patch cable, should be stranded(soft).

IMU **Address: 0 x 028 on some units 0x029 then gnd add pin to get 28**

- Vin 3+ - Org
- GND - Grn
- SCL - Org/Wht GPIO 3
- SDA - Grn/Wht GPIO 2
- ADD

- INT
- BOOT
- REST

I2S mono Amp 9 DB Gain

- LRC - BLU GPIO 19 MISO SPI 1
- BCLK - PINK GPIO 18 CE SPI 1
- DIn - YLW GPIO 21 MISCO SPI 1
- Gain - add 100k res to GND = 15db gain else is 9db default
- SD
- GND - BLK
- Vin - ORG +5V

Original Speaker: 37mm ➔ 50mm Dif: 13mm/2 = 6.5mm required top and bottom

Antenna SG90 Servo

- Signal - ORG LHS GPIO 12 (PWM)
- Signal - ORG RHS GPIO 13 (PWM)
- +5v - RED
- GND - BRN

Eyes

- LH EYE GPIO 24
- RH EYE GPIO 23

Use 2 x 220R = < 15mA

Projector (Flash light led, keep the lens)

- LED - WHT GPIO 25

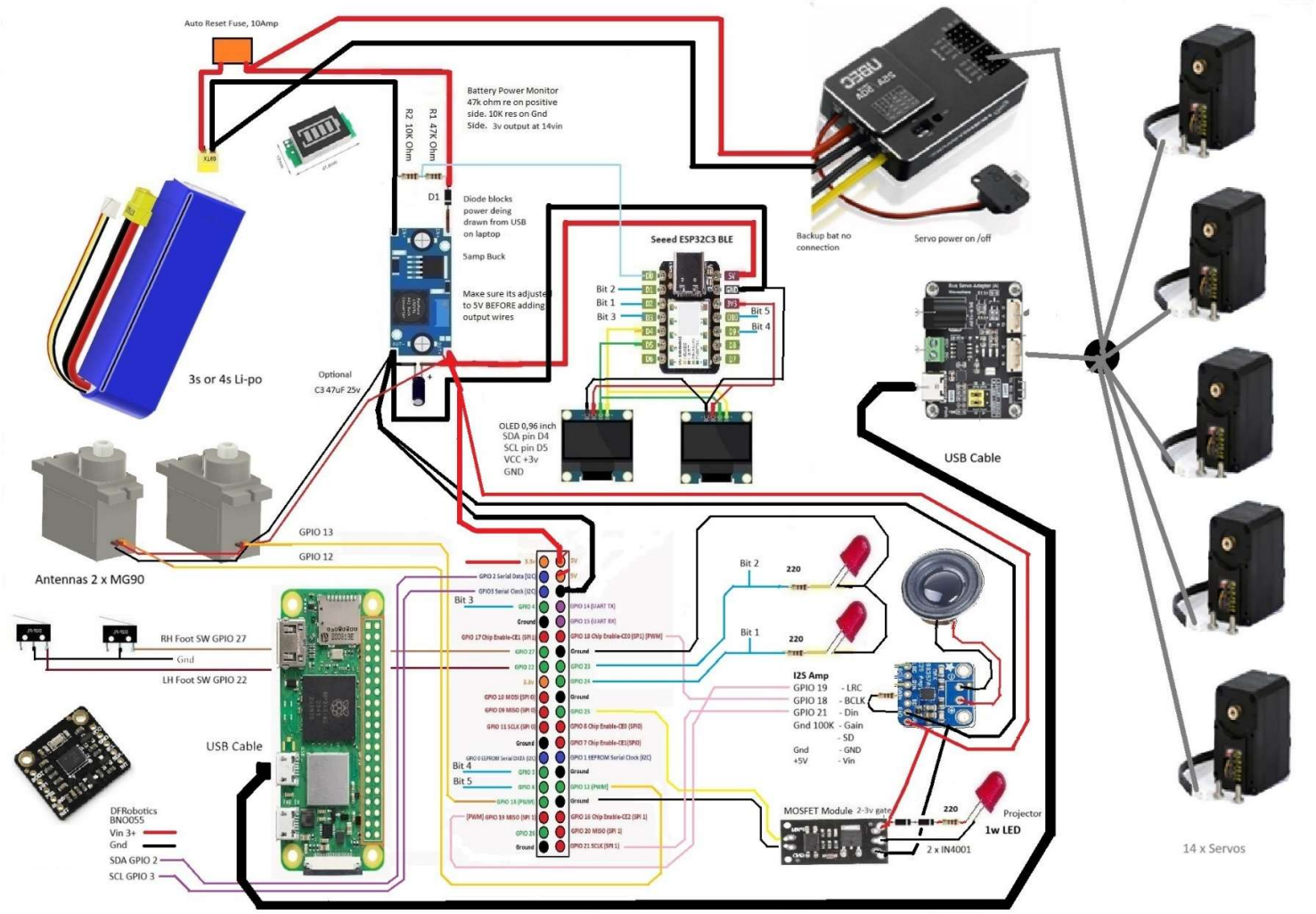
1W high power LED 3.2-3.4v @ 300mA

GPIO max current source is 16mA per pin. 50mA total all GPIO pins combined

Use Mosfet on 5v supply with 3v gate to GPIO. Use 2 x IN4001 to drop Mosfet output voltage to 3v and limit with 220R = 4.5mA. Works good.

Notes:-

- Star wire servos from central point, not looped from servo to servo
- Use a high current UBEC 25/50amp set to 7.4v for servos. Ubec can have an On/Off sw
- Use a separate 5amp Buck set to 5v :
  - Feed Buck direct from battery, **not** off the 7.4v. Servo line
  - 1 x Pi = 2.5amp
  - 2 x SG90 Servo
  - 1 x I2S SPI Amp
  - 1 x Projector 1w 300mA high power LED (use flash light led)
  - 1 x ESP32
- Auto reset fuse - 10amp start with
- Li-Po Battery & XT60 connector
- LED Volt meter display, mounts on rear cover
- Servo Power On/Off mounts in Battery Pod (NC = Off) Part of Ubec


Auto Reset Fuse, 10Amp
Battery Power Monitor
47k ohm re on positive
side. 10K res on Gnd
Side. 3v output at 14vin
R1 47K Ohm
R2 10K Ohm
D1
Diode blocks
power deing
drawn from USB
on laptop
5amp Buck
Make sure its adjusted
to 5V BEFORE adding
output wires
Seeed ESP32C3 BLE
Bit 2
Bit 1
Bit 3
Bit 5
Bit 4
Backup bat no
connection
Servo power on /off
3s or 4s Li-po
Optional
C3 47uF 25v
OLED 0,96 inch
SDA pin D4
SCL pin D5
VCC +3v
GND
USB Cable
GPIO 13
GPIO 12
Antennas 2 x MG90
RH Foot SW GPIO 27
Gnd
LH Foot SW GPIO 22
USB Cable
DFRobotics
BNO055
Vin 3+
Gnd
SDA GPIO 2
SCL GPIO 3
Bit 2
220
Bit 1
220
I2S Amp
GPIO 19 - LRC
GPIO 18 - BCLK
GPIO 21 - Din
Gnd 100K - Gain
- SD
Gnd - GND
+5V - Vin
Bit 3
Bit 4
Bit 5
MOSFET Module 2-3v gate
220
Projector
1w LED
2 x IN4001
14 x Servos

RPi Zero 2w - Pi OS Lite (64-bit) Arm64 RTOS

- 2.5amp PSU
- Mini HDMI
- Micro USB
- Wifi SSID: B818_C937
- WiFi PW: 95A5FHQ8yDQ
- IP:192.168.1.250
- Mask: 255.255.255.0
- GW: 192.168.1.1
- WiFi SSID: bdx
- Host: bdxv2
- UN:richn
- PW:richn01
- SSH: richn@bdxv2
- sftp://richn@bdxv2/ in File Explorer address bar

RPi 5 - PINNS Arm64 – Screen: HDMI 3

- NOT FULLY SETUP WIP
- 2.5amp PSU
- Mini HDMI its not
- Micro USB
- Wifi SSID:
- WiFi PW:
- IP:192.168.1.
- Mask: 255.255.255.0
- GW: 192.168.1.1
- WiFi broad cast ID:
- UN:
- PW:
- SSH:
- PINNs Dual Boot, press shift or JS click.
  - Place holder: Ubuntu 24.04 LTS
  - Raspberrian full GUI (64-bit)
  - Place Holder: what was it

REGULAR Commands

workon open-duck-mini-runtime	// set environment, every time
cd Open_Duck_Mini_Runtime	// change to working directory
cd scripts	// place to go
ls – la	// list files <DIR>
. ./bdx	// bash file that does the 4 lines above
cp <filename> <filename>	// copy file on pi
rm <filename>	// remove file
cd ~/	// got user home/root
ifconfig	//ip config
i2cdetect -y 1	// Show i2c devices lan1 or 2
nano ~/.bashrc	// edit environment
sudo raspi-config	// pi configuration
lsusb ttyACM0	// Linux USB port
nmcli device show	// ipconfig
sudo shutdown -r now	// use: ./restart
sudo shutdown -h now	// use: ./halt

Secure File copy commands for SSH

Copy to pi. Folder : project

scp myfile.txt <username>@<pi_ip_address>:project/

copy from pi to PC

scp <username>@<pi_ip_address>:myfile.txt .

60 essential Linux commands

https://www.hostinger.com/tutorials/linux-commands?utm_medium=ppc&utm_campaign=Generic-Tutorials-DSA-VPS|NT:Se|LO:Other-ASIA-t1&gad_source=1&gad_campaignid=22523926723&gbraid=0AAAADMy-hZ9nZ4nzwbY0ehsrIviZhuKi&gclid=CjwKCAjw_pDBBhBMEiwAmY02No21tSi_GRGcolFSK5iglZF99afnY4mi5KC-VX9c_-uBB_aKmM5rBBoCbt4QAvD_BwE

Mini Duck – Software Install

The follow is at this git hub link, unedited

https://github.com/apirrone/Open_Duck_Mini_Runtime/tree/v2

Update the system and install necessary stuff

sudo apt update

sudo apt upgrade

sudo apt install git

sudo apt install python3-pip

sudo apt install python3-virtualenvwrapper

(optional) sudo apt install python3-picamzero

Add this to the end of the .bashrc:

nano ~/.bashrc // edit environment

```
export WORKON_HOME=$HOME/.virtualenvs

export PROJECT_HOME=$HOME/Devel

source /usr/share/virtualenvwrapper/virtualenvwrapper.sh
```

Enable I2C

sudo raspi-config -> Interface Options -> I2C

TODO set 400KHz ?

Set the usb serial latency timer

cd /etc/udev/rules.d/

sudo touch 99-usb-serial.rules

sudo nano 99-usb-serial.rules

copy the following line in the file

SUBSYSTEM=="usb-serial", DRIVER=="ftdi_sio", ATTR{latency_timer}="1"

Set the udev rules for the motor control board

TODO

Setup xbox one controller over bluetooth

Turn your xbox one controller on and set it in pairing mode by long pressing the sync button on the top of the controller.

Run the following commands on the rasp :

bluetoothctl

scan on

Wait for the controller to appear in the list, then run :

pair <controller_mac_address>

trust <controller_mac_address>

connect <controller_mac_address>

The led on the controller should stop blinking and stay on.

You can test that it's working by running

python3 mini_bdx_runtime/mini_bdx_runtime/xbox_controller.py (this does not seam to work, until the VE is created).

Speaker wiring and configuration

Follow this tutorial

For now, don't activate /dev/zero when they ask

https://learn.adafruit.com/adafruit-max98357-i2s-class-d-mono-amp?view=all

Install the runtime

Make a Virtual Environment and activate it

mkvirtualenv -p python3 open-duck-mini-runtime

workon open-duck-mini-runtime (all further work gets done in this VE, start this on startup)

Clone this repository on your rasp, cd into the repo, then :

git clone https://github.com/apirrone/Open_Duck_Mini_Runtime

cd Open_Duck_Mini_Runtime

git checkout v2

pip install -e .

Test the IMU

sudo apt-get install i2c-tools

i2cdetect -y 1 // imu is on address 0X28

python3 mini_bdx_runtime/mini_bdx_runtime/raw_imu.py

You can also run python3 scripts/imu_server.py on the robot and python3 scripts/imu_client.py --ip <robot_ip> on your computer to check that the frame is oriented correctly.

To find the ip address of the robot, run ifconfig on the robot

Test motors

This will allow you to verify all your motors are connected and configured.

python3 scripts/check_motors.py

Make your duck_config.json

Copy example_config.json in the home directory of your duck and rename it duck_config.json.

cp example_config.json ~/duck_config.json

In this file, you can configure some stuff, like registering if you installed the expression features, installed the imu upside down or and other stuff. You also write the joints offsets of your duck here

Find the joints offsets (Chk notes at HW Setup & Testing, from this point on)

This script will guide you through finding the joints offsets of your robot that you can then write in your duck_config.json

This procedure won't be necessary in the future as we will be flashing the offsets directly in each motor's eeprom.

cd scripts/

python3 find_soft_offsets.py

Run the walk !

Download the latest policy checkpoint and copy it to your duck.

cd scripts/

python v2_rl_walk_mujoco.py --onnx_model_path <path_to>/BEST_WALK_ONNX_2.onnx

- The commands are :

- A to pause/unpause

- X to turn on/off the projector

- B to play a random sound

- Y to turn on/off head control (very experimental, I don't recommend trying that, it can break your duck's head)

- left and right triggers to control the left and right antennas

- LB (new!) press and hold to increase the walking frequency, kind of a sprint mode ðŸ™‚

Additional Environment Improvements, not part of Git files

Convection has .sh file extenstion for bash script files, works without, fuck typing

This file is run at start up.
It can be edited to include starting the environment or the walk.py

nano ~/.bashrc // edit environment

add

```
workon open-duck-mini-runtime
cd /home/richn/Open_Duck_Mini_Runtime/scripts

# walk or head puppet here, xbox On before BDX power up
```

. ./bdx // set environment, and land in work folder

nano bdx

```
#!/bin/bash

cd /home/richn/Open_Duck_Mini_Runtime/scripts

workon open-duck-mini-runtime

ls -l

Ctrl X and save
```

chmod u+x bdx

To Use:

source ./bdx

or

.<space> ./bdx

./shutdownpi //use this line to initiate

cp bdx shutdownpi

nano shutdownpi

```
#!/bin/bash

sudo shutdown -h now

Ctrl X and save
```

chmod u+x shutdownpi

./restartpi

cp shutdownpi restartpi

nano restartpi

#!/bin/bash

sudo shutdown -r now

Ctrl X and save

chmod u+x restartpi

copy restartpi and shutdownpi need to be in the working folder, eg copy into scripts

Also worth coping these files from

/Open_Duck_Mini_Runtime/mini_bdx_runtime/mini_bdx_runtime

to

/scripts. (save changing directories during testing).

antenas.py

feet_contacts.py

projector.py

raw.imu.py

sounds.py

xbox controller.py

HW Setup & Testing

Write Servo IDs

Individually connect each servo and program ID, use masking tape to label servo with ID.

cd scripts

python3 configure_motor.py --id <id> //eg --id 20

"left_hip_yaw": 20

"left_hip_roll": 21

"left_hip_pitch": 22

"left_knee": 23

"left_ankle": 24

"neck_pitch": 30

"head_pitch": 31

"head_yaw": 32

"head_roll": 33

"right_hip_yaw": 10

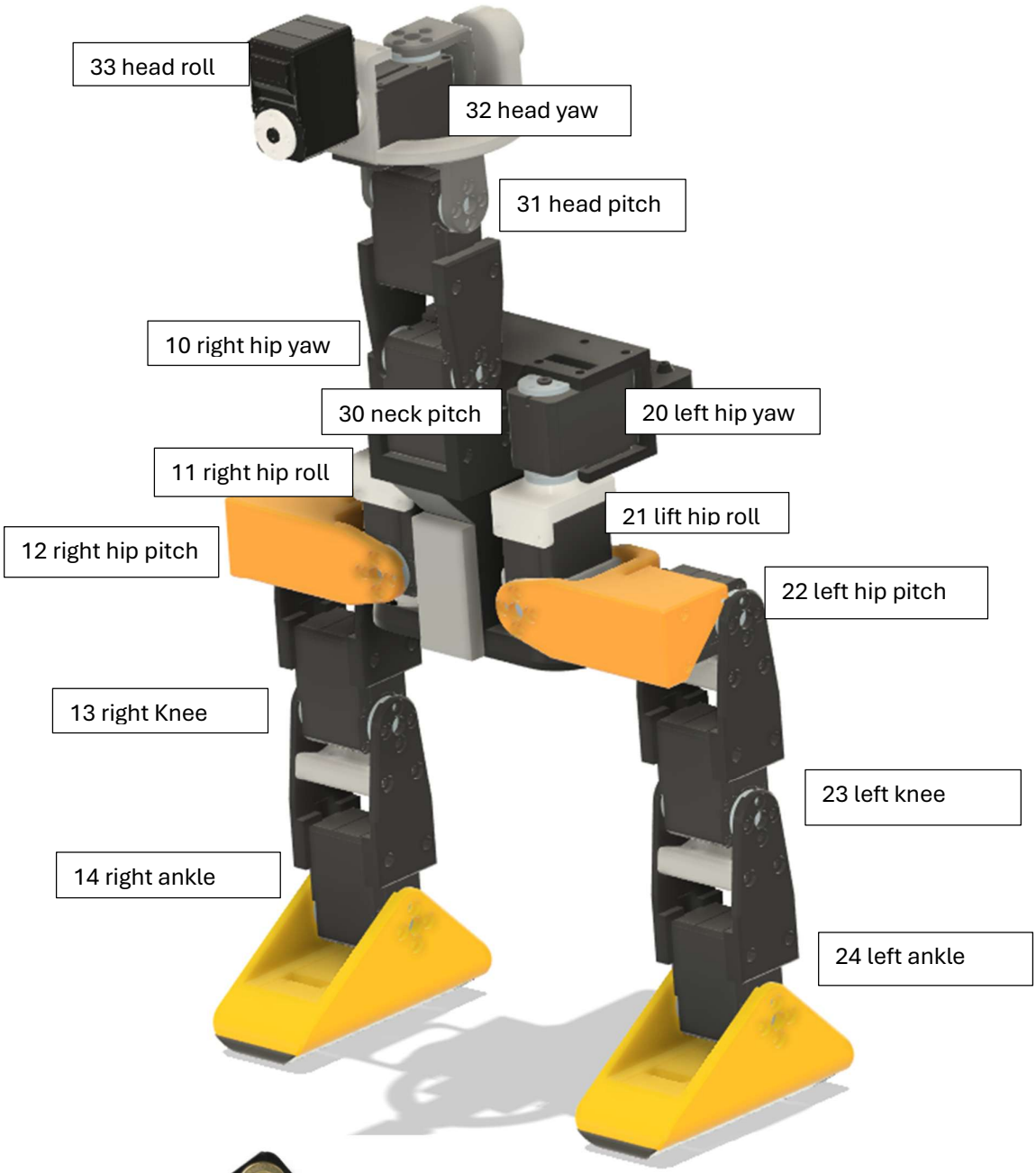
"right_hip_roll": 11

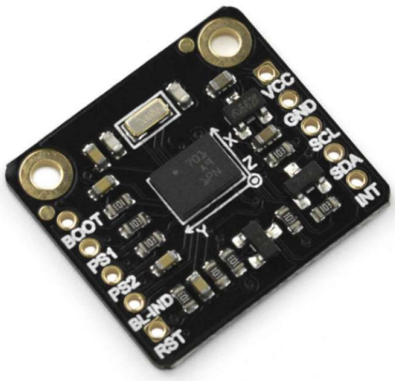
"right_hip_pitch": 12

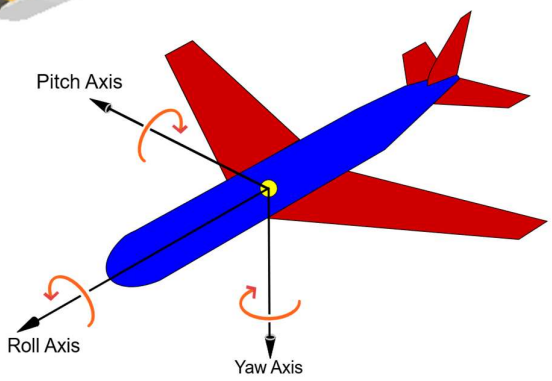
"right_knee": 13

"right_ankle": 14

NOTE: Use cad file to ID Servo wires exit point and orientation. Servo 12/22 cables exit rear







Right side up: Y Forward, X Right, Z Up

Upside down: Y Backward, X Right, Z Down.

Testing

Wire up Servos and confirm operation: // Servo Power Switch On (On = NO)

python3 check_voltage.py // useful to confirm operation of all 14 servos

python3 check_motors.py // when asked, do not need to set offsets yet

Test Xbox Controller

PS3 would not work. New Xbox wireless controller, did not work.

Needs firmware upgrade. Search Disc Xbox to find details

To reset Xbox controller, bat out, 1-15sec bat in, hold sync button until controller vibrates

- On PC, Goto Microsoft store, look for Xbox Accessories and download
- Plug Xbox Ctrl in via USB
- Apply upgrade via Xbox Accessories software. Back to Pri

- If controller is enrolled need to run bluetoothctrl
- # Bluetoohctrl
- remove <mac address>
- Agent on
- Scan on
- Connect <mac address>
- Trust <mac>
- Bond <Mac>
- Info <mac>
- devices
- Might have to power cycle/reboot a couple times to get it to take

python3 xbox_controler.py // test controller

Test IMU (Address: 0 x 28)

i2cdetect -y 1 // Ii2c scanner

python3 raw_imu.py // View raw output

Right side up

Y Forward

X Right

Z Up

Upside down

Y Backward

X Right

Z Down

Test Feet Switches

python3 feet_contacts.py // Need these working B4 it will run

Test Antennas // edit duck_config.json option = true

python3 antennas_control_test.py // turn controller on first

python3 antennas.py // no controller, random

Test Sound

requires sound driver loaded, and edit duck_config.json option = true

python3 sounds.py // play random sound

Test Projector // edit duck_config.json option = true

python3 projector.py // Projector on/off

Calibration and getting it going

Set Servo Offsets (pre set BDX to sit like above drawing for correct offset position)

- Legs straight on mounting base, like drawing
- Neck vertical
- Head horizontal with floor
- Servo Power switch on

python3 check_motors.py // Write offsets

python3 find_soft_offsets.py

the output offsets need to be copied and edited into duck_config.json

This procedure may need to be done more than once, to get things right for walking

IMU Calibration

python3 calibrate_imu.py

Calibration status: (sys, gyro, accel, mag) // they all need to equal 3

CALIBRATION DONE

offsets_accelerometer (-117, -14, -28)

offsets_gyroscope (4, 1, 0)

offsets_magnetometer (29, -104, -408)

Saved imu_calib_data.pkl

1. Jaime — 4/24/25, 1:23 PM

   You need to move the duck physically and lay it on all of its sides, so front, left, back, right, top, bottom until all numbers say 3.

2. **shaboomi — 4/25/25, 12:45 PM**

   its all working now, I did the following: Follow the Adafruit "calibration dance" for each sensor Gyroscope (G): keep the board perfectly still until you see the G-value go to 3.
   Accelerometer (A): place the board flat on each of its six faces, holding each orientation still so the A-value increments up to 3.
   Magnetometer (M): wave the board through a "figure-8" motion in all three dimensions until the M-value climbs to 3.
   System (S): once G, A and M are 3, the system S will automatically move to 3.

   Adafruit Learning System Keep an eye on the printed tuple and only move on when each component hits 3.

https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/bno055-sensor-calibration-circuitpython

TurnOff/On

This is use full, for testing. Turn on will move bdx to standing pos. Can then take off mount before walk. Need to hold it on walk startup else falls over.

Turn off will relax the servos. Do it on the stand. Else falls on floor

python3 turn_on.py

python3 turn_off.py

Test Head puppet **Turn Xbox controller on first**

Will assume standing position

python3 head_puppet.py

-Xbox controller:

- Left Stick moves head axis Pitch/Yaw
- Right Stick - Left/Right Head axis Roll
- Right Stick - Up/Down Neck Pitch #30 – **un-completed code, commented out**
- Y = Mode select (head control, noobled for head puppet)
- X = Projector
- B = Sound
- A = Eyes (RN Dev)
- L_Trig = Right Antena? test
- R_Trig = Left Antena? test
- LED Eyes OP, will be on random blink timer

Test Walk **Turn Xbox controller on first**

python3 v2_rl_walk_mujoco.py --onnx_model_path ~/Open_Duck_Mini_Runtime/BEST_WALK_ONNX_2.onnx

- Xbox controller:

- Y = Head control on/off (switch between walk). (very experimental, I don't recommend trying that, it can break your duck's head)
- X = turn projector LED on/off
- B = play a random sound
- A = pause/unpause (prints to terminal)
- left and right triggers control the left and right antennas
- LB (new!) press and hold to increase the walking frequency, kind of a sprint mode. ~ freq factor = 1.3 else 1.0
- Dpad Up = ~freq offset +0.5
- Dpad Down ~ Freq offset -0.5

Notes:

Panic/crash out if it detects movement errors, ie its on the stand, needs to be on a surface.

Could do a dance if IMU not calibrated or upside down (was funny).

Cashes on I2C bus errors. Need twisted pair wiring, see neck wiring

OS Error parsing issue: This is due to 1 or more servos not communicating, chk wires

Networking SSID & Wi-Fi Pwr Save Off

Jaime

I had to edit the configuration with this command:

sudo nano /etc/NetworkManager/system-connections/preconfigured.nmconnection

And then add my network inside it. just change the ssid and psk

[connection]

id=preconfigured

uuid= This shouud be unique to your pi so dont change it

type=wifi

autoconnect=true

[wifi]

mode=infrastructure

ssid= **Add your SSID here**

hidden=false

[wifi-security]

key-mgmt=wpa-psk

psk= **Add your password here.**

[ipv4]

method=auto

[ipv6]

addr-gen-mode=default

method=auto

[proxy]

WiFi Performance Improvement – Jaime 7/17/25, 3:17 PM

I kept having the wifi on the pi zero be really slow and then ..... there is a wifi power save "feature" thats turned on by default... that powers down the radio on the wifi if no activity is detected... I turned it off and Im having a better wifi experience

This checks to see if its on:

iw dev wlan0 get power_save

if it says “on”, power-save is enabled

Then this disables it:
sudo iw dev wlan0 set power_save off

Then you can modify the NetworkManager configuration to disable that for good with this command:

cat <<EOF | sudo tee /etc/NetworkManager/conf.d/wifi-powersave.conf > /dev/null

At the > enter the following and press <enter>

[connection]

wifi.powersave = 2

EOF

Drops back to cmd prompt

Walking

Github Cmd

python3 v2_rl_walk_mujoco.py --onnx_model_path ~/Open_Duck_Mini_Runtime/BEST_WALK_ONNX_2.onnx

Command line switches for Walk startup

Jaime — 7:46 AM

Pi5 python v2_rl_walk_mujoco.py --onnx_model_path /home/jaime/Open_Duck_Mini_Runtime/checkpoints/BEST_WALK_ONNX_2.onnx -p 32 --cutoff_frequency 40 --save_obs True 18650 fresh charge.

To start walk program on power up. Jaime:-
It's a service that runs on boot. Waits for the controller to connect and then starts the script. Works really well

This is possibly done in .baschrc

LCD Eyes

Serial coms link

Ideal Mode = true, No activity, eyes will move around and Blink randomly

Ideal Mode = false. Active, Eyes centred unless command to move position X,Y

Blink is random by ESP

- Serial Commands (ESP32)

#D$ Expression - Default Eyes (Square)

#T$ Expression - Tired Eyes

#A$ Expression - Angry Eyes

#R$ Expression - Happy/Relaxed Eyes

#Y$ Effect – Curiosity - Toggle

#FHn$ Effect - Flicker Horizontal - Toggle

#FVn$ Effect - Flicker Vertical- Toggle

#ID$ Effect - Invert Display - Toggle

#O$ Effect - Cyclops Mode - Toggle

#L$ Animation - Laugh Toggle

#C$ Animation - Confused Toggle

#PPn$ Position 1-8 : N NE E SE S SW W NW

#I$ Idel mode On/Off toggle (Eyes bounce around)

#WWn$ Eye Width

#HHn$ Eye Hight, Blink must be off for this to hold

#BBn$ Border radius

5 Bit bus = 0-32 Dec

Bit	1	2	3	4	5
GPIO	23	24	4	5	6

No Bus Change in x Seconds, then goes to Ideal Mode, Eyes move and blink randomly.

5 Bit Commands (0-31):

0 Ideal Mode - Eyes move and blink randomly

Expression – Pulsed **A Button** (cycle around expressions) – Latching

1 Expression - Default Eyes (Square)
2 Expression - Tired Eyes
3 Expression - Angry Eyes
4 Expression - Happy/Relaxed Eyes

Effect – Momentary - While Button held down

5 Effect - Flicker Horizontal Toggle -**Back**
6 Effect - Flicker Vertical Toggle - **Start**
7 Effect - Invert Display Toggle - **Guide**
8 Effect – Wink Toggle – **LB**
9 Effect – Blink Toggle – **RB**
10 Effect - Animation - Laugh Toggle – **LSB**
11 Effect - Animation - Confused Toggle - **RSB**
12 Effect - Curiosity Toggle –

Position - Momentary - While **Dpad** held down

13 Position – Centre - Default
14 Position - N
15 Position - NE
16 Position - E
17 Position - SE
18 Position - S
19 Position -SW
20 Position - W
21 Position - NW

Xbox - Button Assignments

- Left Stick = Left/Right Head axis Yaw
- Left Stick = Up/Down Head axis Pitch
- Right Stick = Left/Right Head axis Roll
- Right Stick = Up/Down Neck Pitch #30 – **uncompleted code, disabled**
- Y = Mode Select (select Head control, or Walk)
- X = Projector
- B = Sound

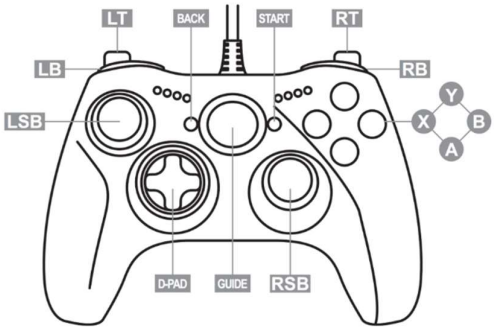
- A = Eyes Expression cycle
- L_Trig = Right Antena
- R_Trig = Left Antena
- Left Stick PB = Animation Laugh
- Right Stick PB = Animation Confused
- L Shoulder PB = Effect - Wink
- R Shoulder PB = Effect - Blink
- Select = Effect – H Flicker
- Start = Effect – Z Flicker
- Upload = Invert Display

Button # list mapping for XBox controller. Feeds xbox_controler.py

- LStick 1 = left analog stick
- RStick 2 = right analog stick

- Dpad = self.p1.get_hat(0)[1] Up = 1 Dwn = -1
- Dpad = self.p1.get_hat(0)[0] Left = -1 Right = 1
- Button 0 = A
- Button 1 = B
- Button 2 =
- Button 3 = X
- Button 4 = Y
- Button 5 = na
- Button 6 = LB
- Button 7 = RB
- Button 8 = could be LT Trigger
- Buton 9 = could be RT Trigger
- Button 10 = BACK
- Button 11 = START
- Button 12 = have seen but don't now what button it was??
- Button 13 = LSB
- Button 14 = RSB
- Button 15 = GUIDE

- left_trigger Button # = is unknown, have not been able to trap a button number. But can trap on this flag
- right_trigger Button #, as above.



The following files have been modified to make this work:

head_puppet.py

- includes eyes.py,
- includes xbox_controler.py
- Modded to pickup triggers for additional new buttons

xbox controller.py

-includes buttons.py
- Hacked to print any and all button presses with label.
- Modded to add, addition new buttons

buttons.py

- Reads the xbox controller buttons. Gives a number per button 1 -16.
- Some buttons are pre defined and do not respond eg LT/RT
- Head puppet reads buttons.trigger (this is our trap)
- Modded to define new buttons so trap works

eyes.py

- This basically a new file with an original file name. ie fits in the stack of any existing bdx's
- 5 bit bus physical GPIO output to ESP32 LCD eyes controller.
- Good ref for GPIO config.
- Triggered from head_puppet.py with button (n), sending lcd_command value to eyes.py
- Converts lcd_command to binary and outputs on GPIO

Python3 Code, Dec to Binary

```
decimal_number = 10
binary_string = bin(decimal_number)
print(binary_string) # Output: 0b1010

# To remove the "0b" prefix:
binary_without_prefix = binary_string[2:]
print(binary_without_prefix) # Output: 1010
```

Arduino C++ Code, Binary to Decimal

```
if (ISR_5bbus_Flag) {          //if  isr flag set, bits have changed
      ISR_5bbus_Flag = false;

      int Bit_1 = digitalRead(PIN_BIT1);
      int Bit_2 = digitalRead(PIN_BIT2);
      int Bit_3 = digitalRead(PIN_BIT3);
      int Bit_4 = digitalRead(PIN_BIT4);
      int Bit_5 = digitalRead(PIN_BIT5);

int sum = 0;

if (Bit_5) {
      sum = 16
}
if (Bit_4 ) {
      sum = sum + 8;
```

```
}
if (Bit_3) {
      sum = sum + 4;
}
if (Bit_2) {
      sum = sum + 2;
}
if (Bit_1) {
      sum = sum + 1;
}
PiBusValue = sum;
```

Programming Environment

VS Code for editor, with extensions for everything, config AI too. Watch you tube vids

Its unstable using SSH via VS on Pi zero. gave up

Solution

Using, WinSCP copy bdx files to local drive to make a repository

Daily

1. Modem on
2. PC On
3. BDX – On
4. Start WinSCP + connect to BDX
5. Open putty terminal via WinSCP (only need to put in PW)
6. Start BDX runtime $. ./bdx
7. Start VS, only use as editor on local repository
8. Use WinSCP to copy and past edited files to Pi zero
9. Use Putty terminal to test code

Galery

12-4-25 => 25-7-25

Add head internal

Add Projector led

Disney BDX Diagrams