MagBot

Think. Build. Code. Play.

Rev: 1.2

SETUP

BUILD

OPERATE

CODE

CALIBRATION

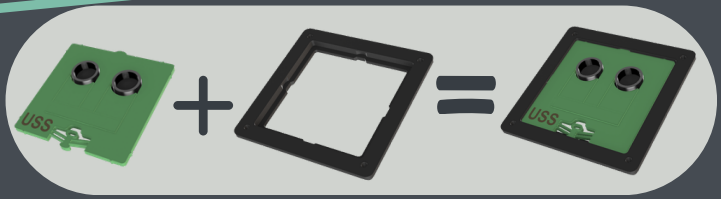
TABLE OF CONTENTS

INTRO

Welcome to your new learning robot! This robot was designed for kids, students, and makers of all ages to explore electronics, coding, and problem-solving in a fun, quick and hands-on way.

The robot grows with you: first you play, then you learn how it works, and later you can code it, customize it, and invent your own upgrades.

Top Plate → 5x buttons + RGB
Front Plate → Ultrasonic sensor
Right Plate → Potentiometer
Left Plate → Power switch
Back Plate → Battery pack
Bottom Plate → Main bot (ESP32 microcontroller, 2x motors, sensors, and selector switch)



This modular design means you can rebuild and upgrade your robot anytime.

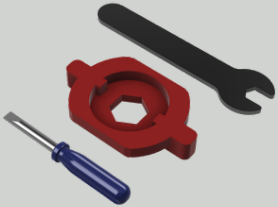
WHAT'S IN THE BOX

MagBot

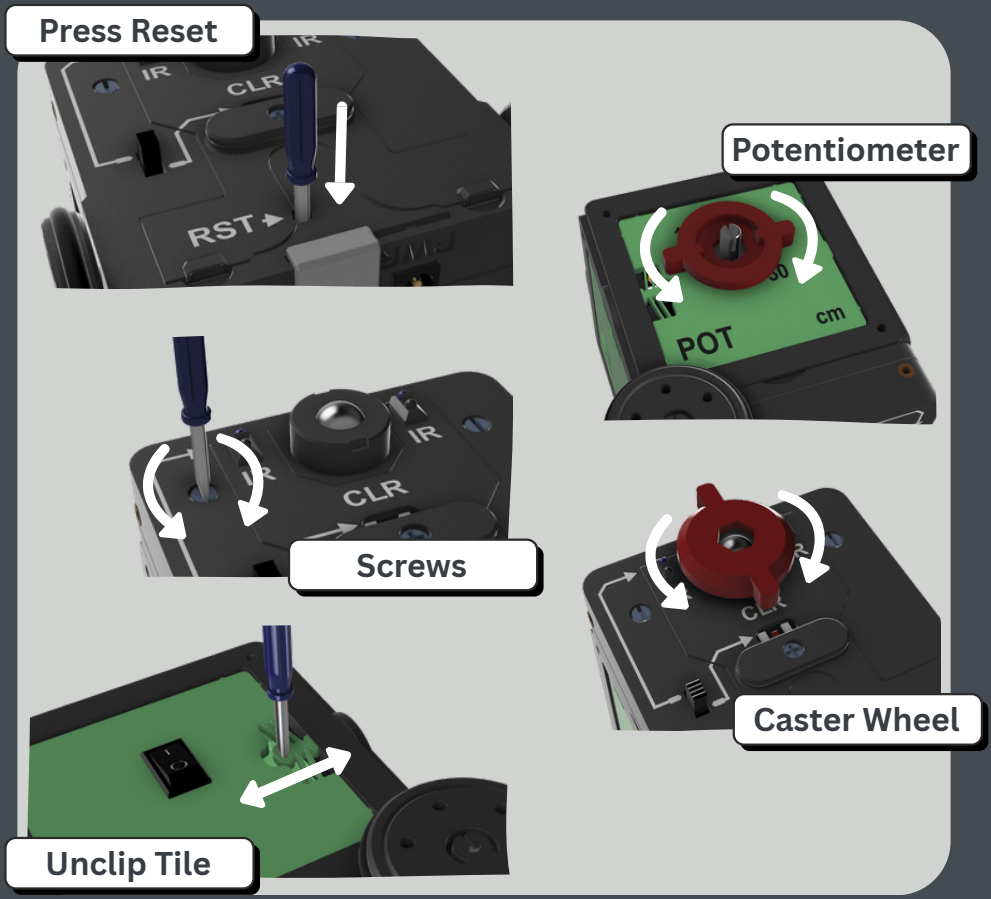
NFC Tag

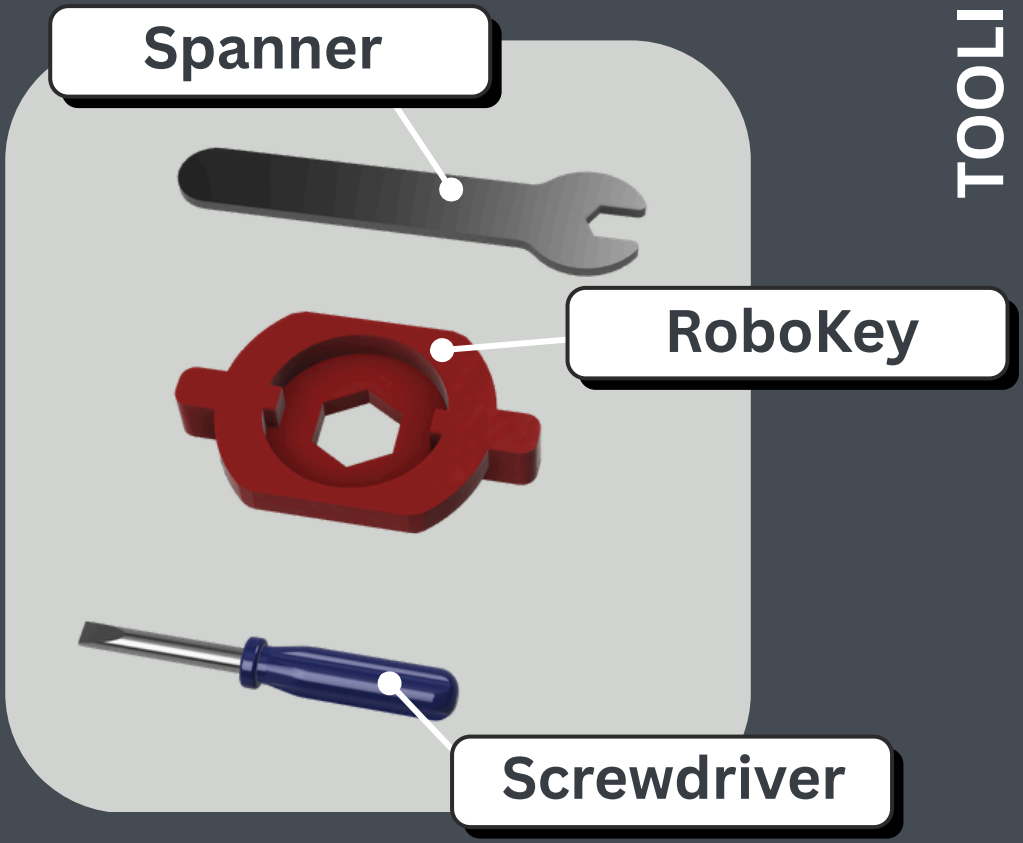
Cables

Tools



Battery


Press Reset
IR
CLR
RST
Potentiometer
POT
cm
Screws
IR
CLR
Caster Wheel
Unclip Tile


Spanner
RoboKey
Screwdriver

1. Barrel 5.5x2.1mm connector, used to power the MagBot from the rechargeable Battery.

2. Female 2Pin Jumper Connection for any other type of battery in the 6-10V range, ideally aimed for a 9V battery.

3. Male 2Pin Jumper Connection when Bridged connects power from either of the power connections (1) or (2) to the rest of the MagBot, powering it on.

4. Female 5Pin Jumper Connection, connected as follows:
   - A1 - Turn Right Button (Yellow)
   - A2 - Turn Left Button (Orange)
   - GND - (Black)
   - 3.3V - (White)
   - D13 - Action Button (Brown)

5. Boost Converter, supplying the motor controller with a constant 12V line.

6. Female 4Pin Jumper Connection, that is connected as follows to the Ultra Sonic Sensor (USS):
   - 5V - (White)
   - D3 - Trig (Yellow), ~ PWM
   - D2 - Echo (Orange), 5->3.3V Logic
   - GND - (Black)

7. Buck Converter, supplying the ESP32 and all 5V pins with a constant 5V line.

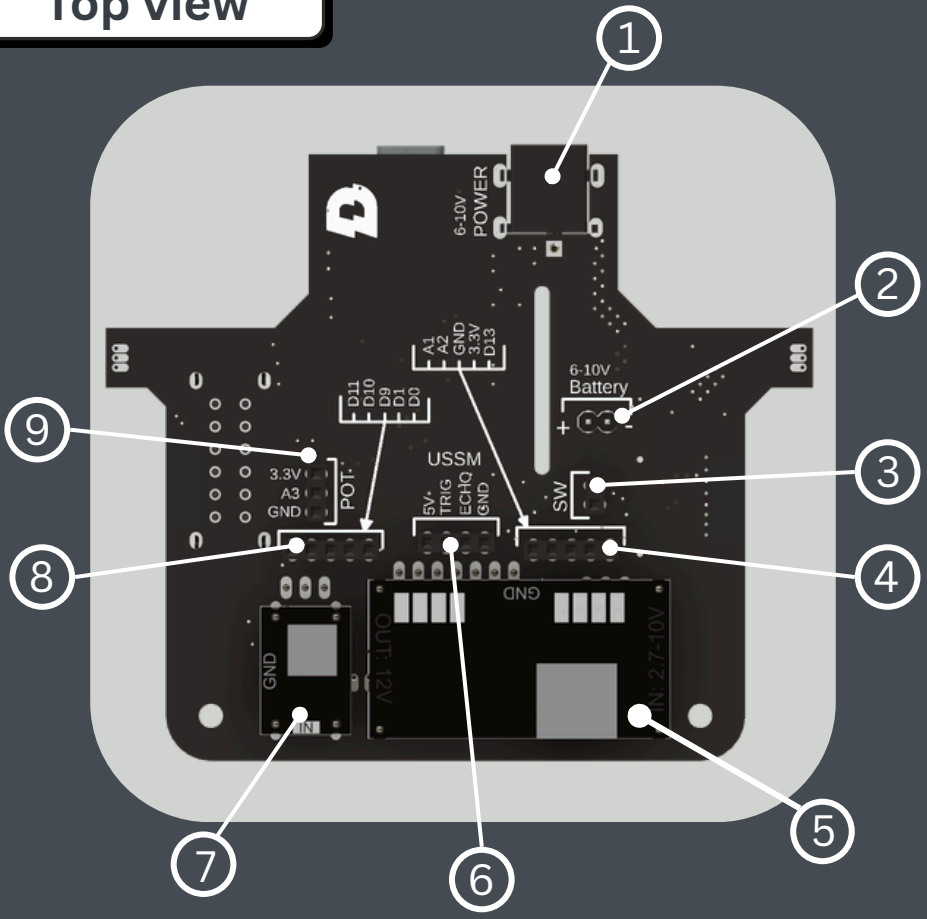
8. Female 5Pin Jumper Connection, connected as follows:
   - D11 - Blue LED (Blue), ~ PWM
   - D10 - Green LED (Green), ~ PWM
   - D9 - Red LED (Red), ~ PWM
   - D1 - Back Button (Grey)
   - D0 - Forward Button (Purple)

9. Female 3Pin Jumper Connection, connected as follows:
   - 3.3V - (White)
   - A3 - Potentiometer (Orange)
   - GND - (Black)

~PWM: Pulse Width Modulation, is a way to control how much power a device gets by switching it on and off very quickly. Used to control a motor's speed, LEDs brightness or a Buzzers tone.

5->3.3V Logic: These pins have a voltage divider connected to them to ensure 5V signals are turned in 3.3V logic for the ESP32 as some sensors and modules required 5V to operate.

Top view


1
2
3
4
5
6
7
8
9
6-10V
POWER
A1
A2
GND
3.3V
D13
6-10V
Battery
+
-
D11
D10
D9
D1
D0
3.3V
A3
GND
POT
USSM
5V
TRIG
ECHO
GND
SW
GND
GND
OUT: 12V
IN: 2.7-10V
IN

10. USB-C Port, used for programming the MagBot.

11. Reset Button, might be required to press while troubleshooting issues.

12. Hall sensor used to determine the position and speed of the Right side wheel.

13. Selector switch, connected to the A4 (SDA) and A5 (SCL) pins of the ESP32; it connects these pins to either the SDA and SCL pins of the Color sensor or individual analog pins of the Left and Right IR sensors.

14. Female 3Pin Jumper Connection, connected to the Rigth IR sensor as follows:
- OUT - A4 pin via the selector switch
- 3.3V
- GND

15. Connects to the Right side Motor.

16. Female 7Pin Jumper Connection, connected to the Color sensor as follows:
- 3.3V
- GND
- SCL - A5 via selector switch
- SDA - A4 via selector switch
- LED

17. Connects to the Left side Motor.

18. Female 3Pin Jumper Connection, connected to the Left IR sensor as follows:
- OUT - A5 pin via the selector switch.
- 3.3V
- GND

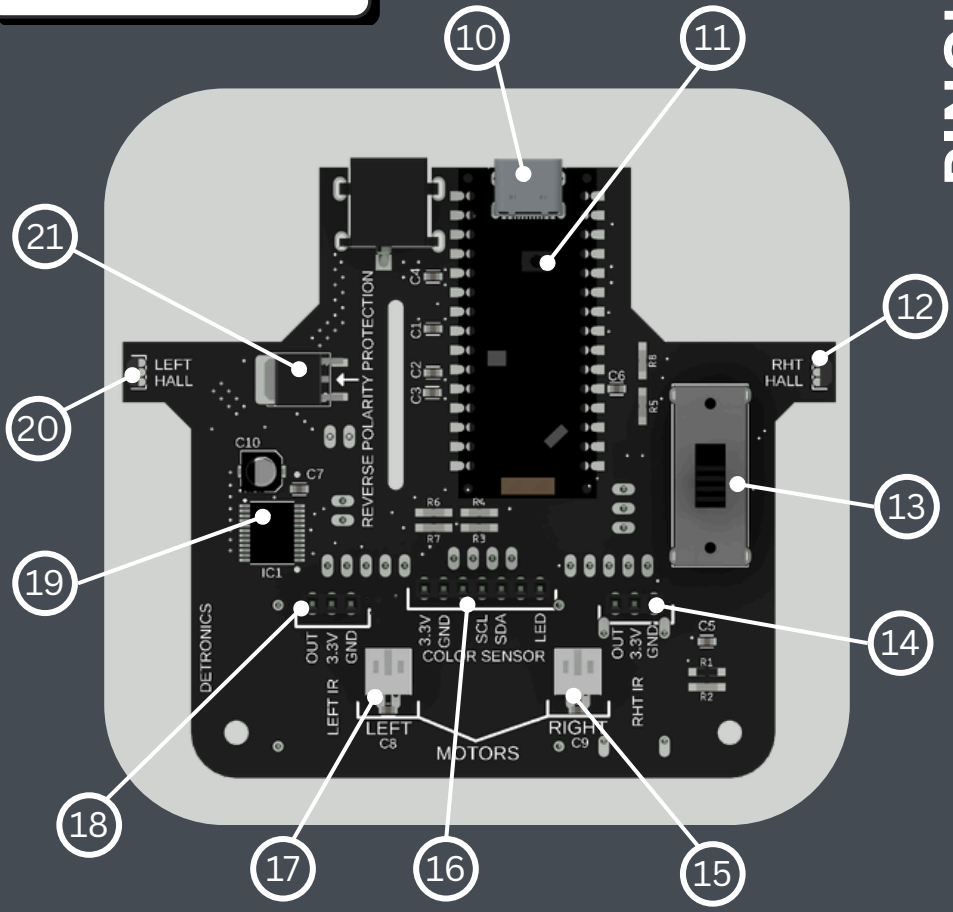
19. Motor controller connects to:
- D12 - IN2 of the LHS motor
- D4 - IN1 of the LHS motor
- D5 - ENA, controlling the speed of the LHS motor, ~ PWM
- D7 - IN3 of the RHS motor
- D8 - IN4 of the RHS motor
- D6 - ENB, controlling the speed of the RHS motor, ~ PWM

20. Hall sensor used to determine the position and speed of the Right side wheel.

21. P-Type MOSFET used for reverse polarity protection, thus the MagBot will not get damaged if the battery is connected the wrong way.

I²C: (Inter-Integrated Circuit) is a way for the robot to talk to many sensors and devices using only two wires. One wire SCL (Serial Clock) carries the clock and keeps everything in sync, while the other SDA (Serial Data) carries the data or the actual messages. Each device on the bus has its own address, so the robot knows who it's talking to.

Bottom view


10
11
21
12
20
13
19
14
18
17
16
15
LEFT HALL
RHT HALL
REVERSE POLARITY PROTECTION
C10
C7
IC1
C4
C1
C2
C3
C6
R8
R5
R6
R4
R7
R3
DETRONICS
OUT
3.3V
GND
LEFT IR
3.3V
GND
SCL
SDA
LED
COLOR SENSOR
OUT
3.3V
GND
RHT IR
C5
R1
R2
LEFT
C8
MOTORS
RIGHT
C9

Wire Colors

- White: VCC (Power) – either 5 V or 3.3 V, depending on the sensor
- Black: Ground (GND)
- Red: RGB LED – Red channel
- Blue: RGB LED – Blue channel
- Green: RGB LED – Green channel
- Orange: Multiple Signal inputs
- Yellow: Multiple Signal inputs
- Purple: Button inputs
- Brown: Button inputs
- Grey: Used for 1x button and the power switch

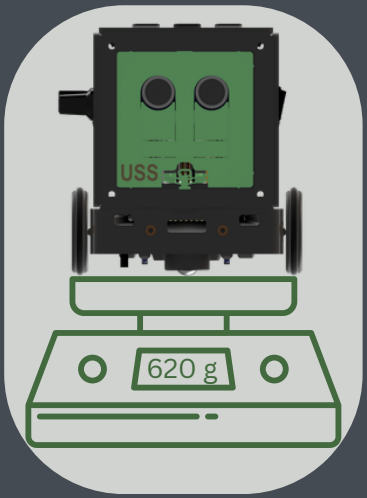
Tip: Always check which wires are connected before making changes or upgrades. Mixing up VCC and GND can damage sensors or the ESP32.

Battery Indicator

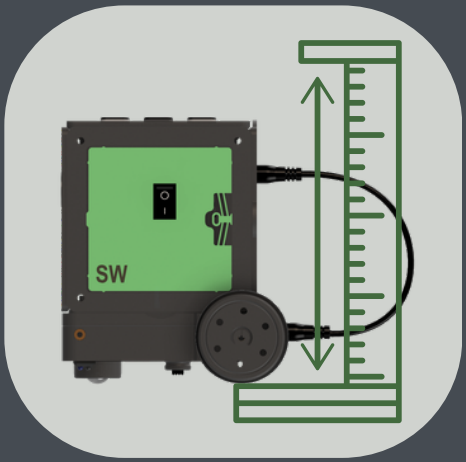
On Start Up:

%	Voltage	Flash Sequence	Speed
100%	8.2 - 8.4V		Full speed
83%	7.65 - 8.19V		Full speed
67%	7.36 - 7.64V		Full speed
50%	7.07 - 7.35V		Slower speed
33%	6.79 - 7.06V		Slower speed
17%	6.50 - 6.78V		Slower speed
0%	5.80 -6.49V		No further code

SPESIFICATION



Max Speed 17.2 cm/s

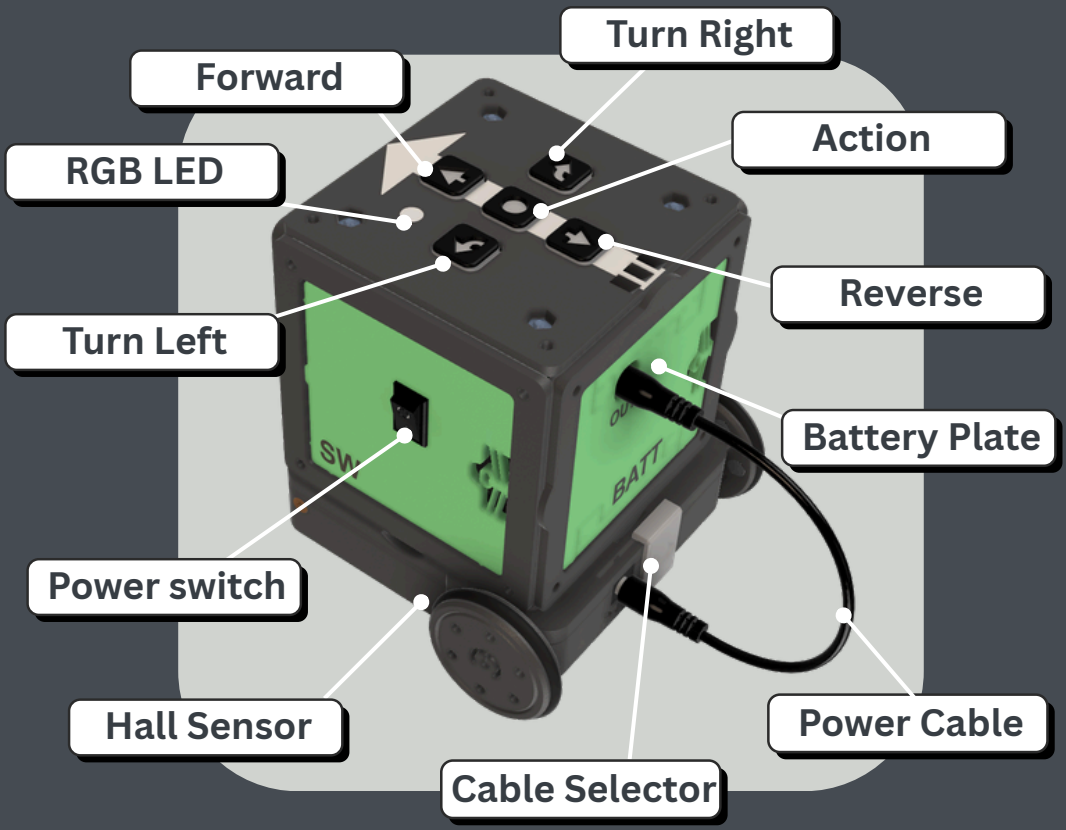


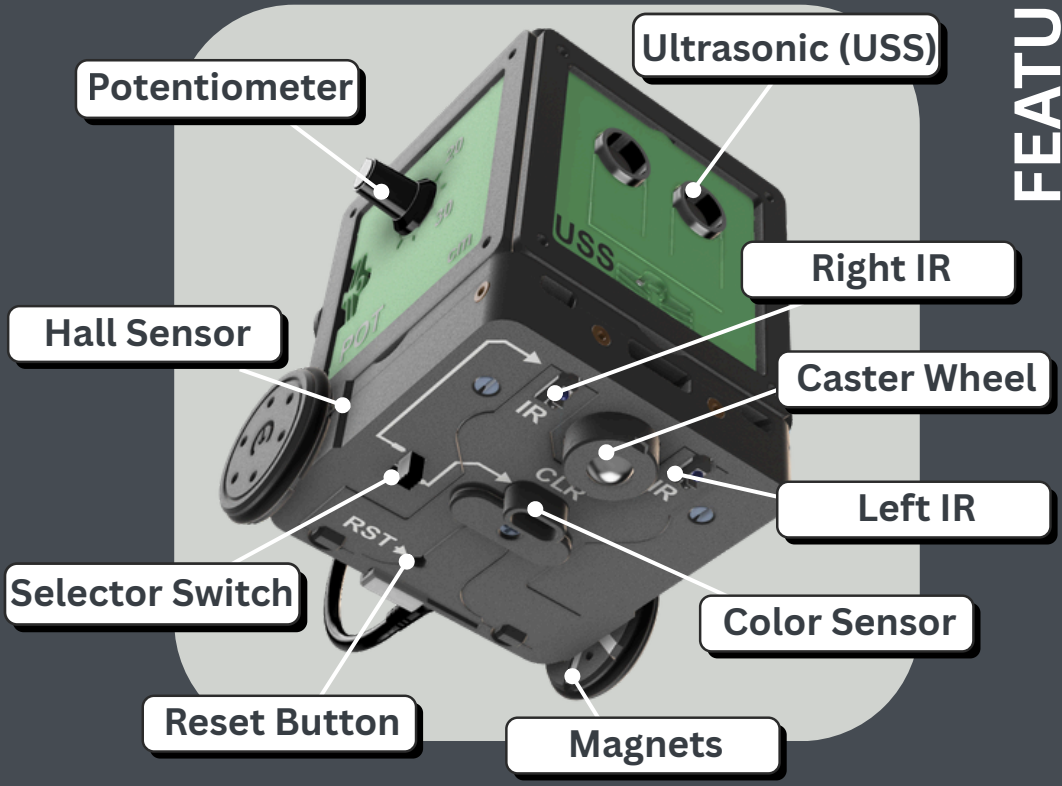
Dimensions:
(W × D × H) 120 × 105 × 122 mm
(excluding power cable at back)

Ground Clearance IR sensor: 5 mm; Color sensor: 3 mm

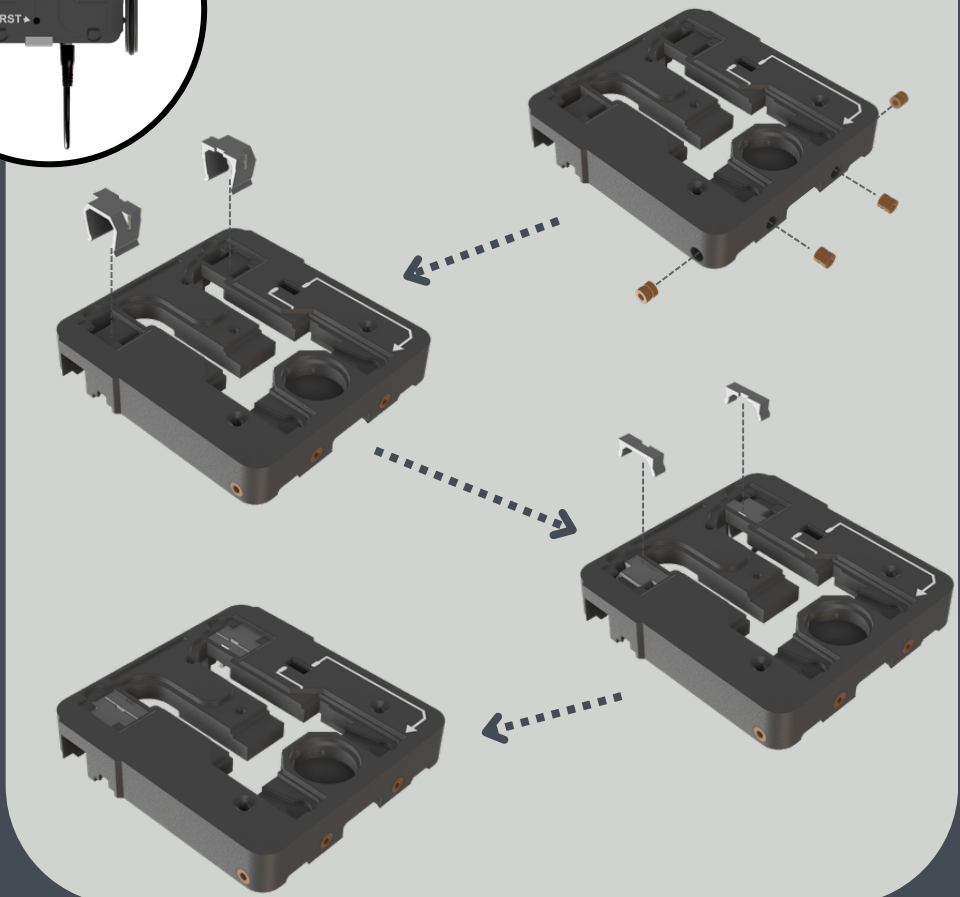
ESP32-S3-Nano:

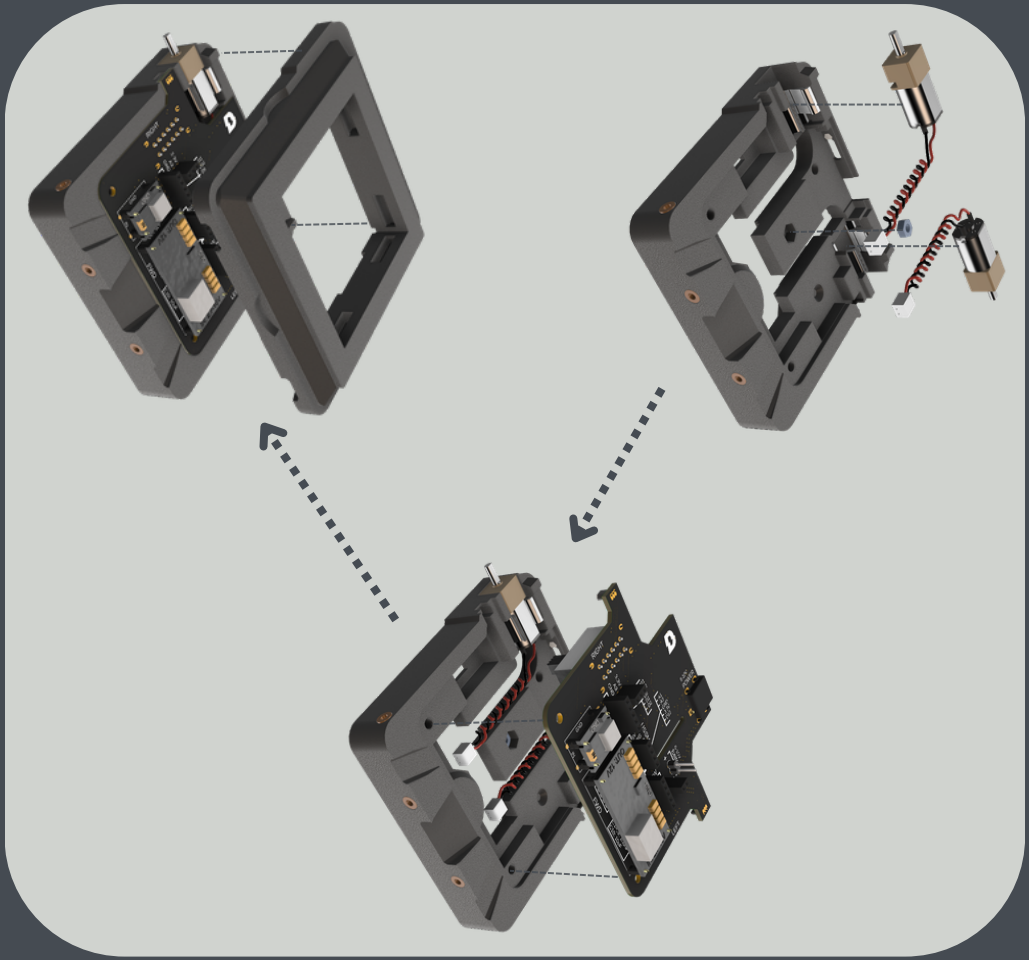
- Operating Voltage: 3.3V
- Pins:
  - Analog = 8
  - Digital I/O = 14
  - PWM = 5
  - I2C = 1
- Microcontroller: Dual-core, up to 240 MHz
- Storage:
  - 512 KB RAM;
  - 384 KB ROM;
  - 8 MB PSRAM;
  - 16 MB Flash
- Wireless: Integrated 2.4 GHz Wi-Fi and Bluetooth LE
- Programming: Supports both Arduino and MicroPython; compatible with Arduino IoT Cloud and HID emulation (keyboard/mouse)


Turn Right
Forward
Action
RGB LED
Reverse
Turn Left
Battery Plate
Power switch
Hall Sensor
Power Cable
Cable Selector
BATT

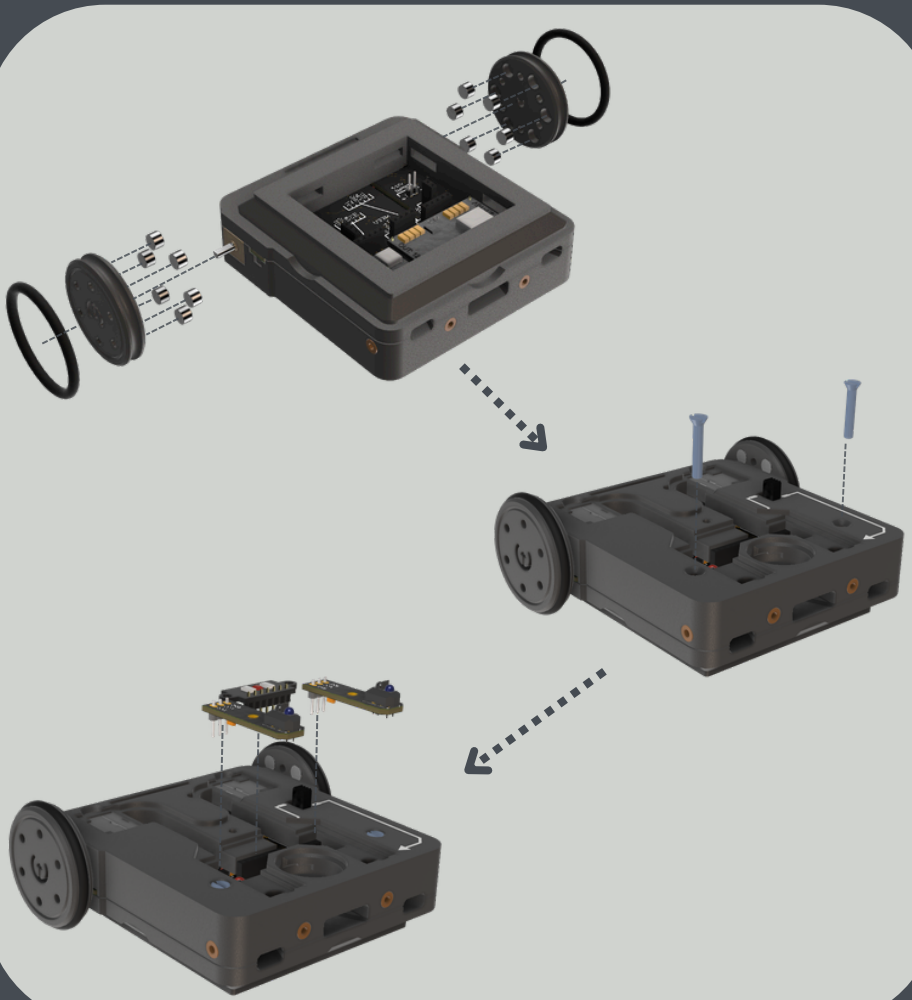

Potentiometer
Ultrasonic (USS)
Right IR
Hall Sensor
Caster Wheel
Left IR
Selector Switch
Color Sensor
Reset Button
Magnets
USS
POT
IR
CLK
IR
RST

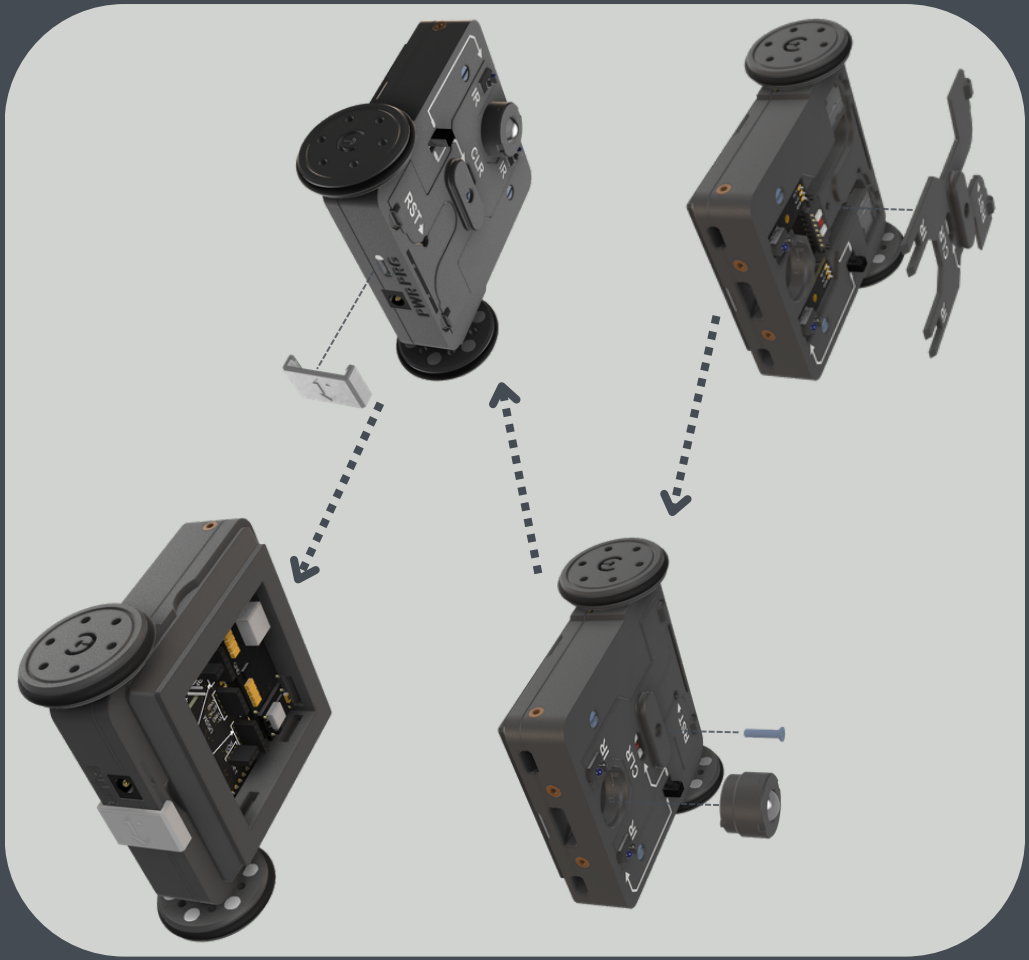
Bottom Plate





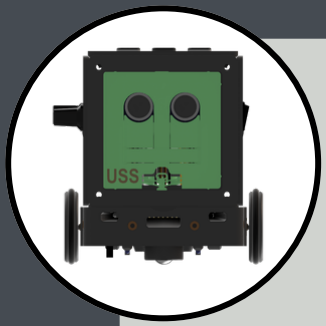
BOTTOM PLATE BREAKDOWN

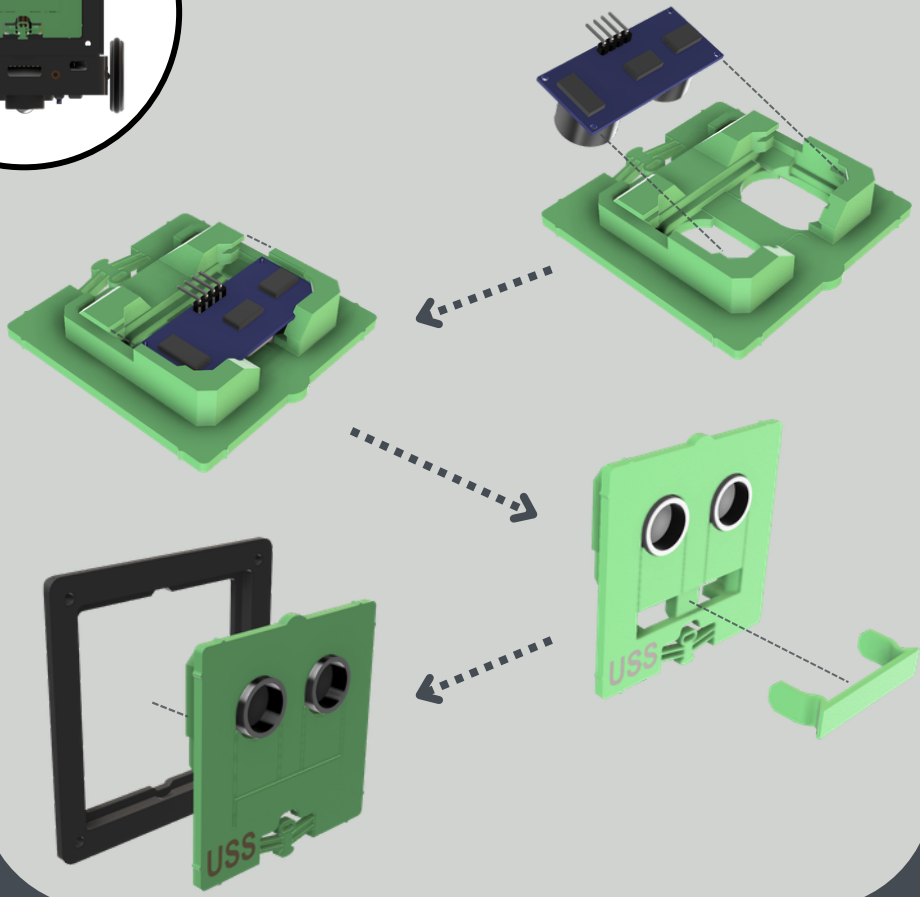


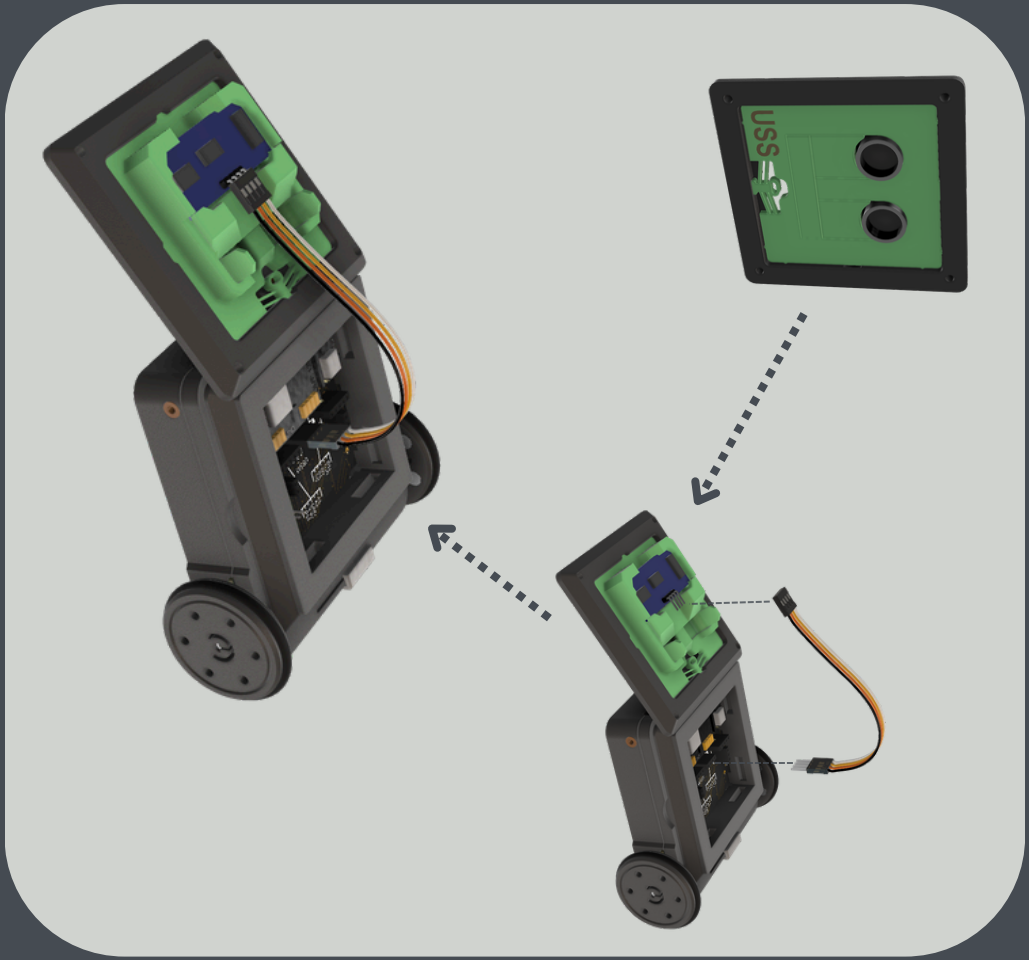


BOTTOM PLATE BREAKDOWN

Front Plate

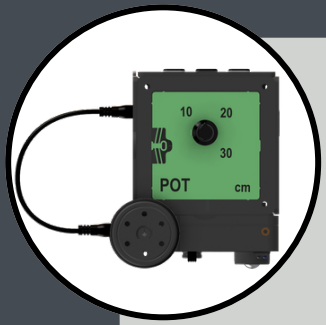


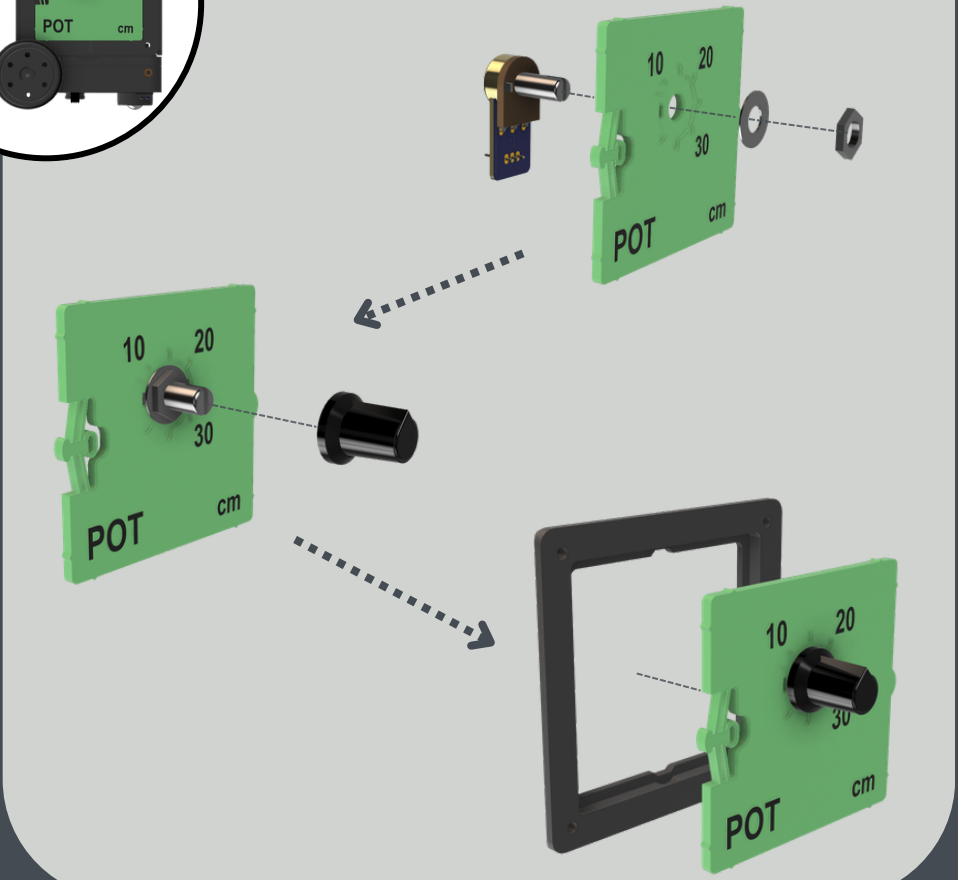




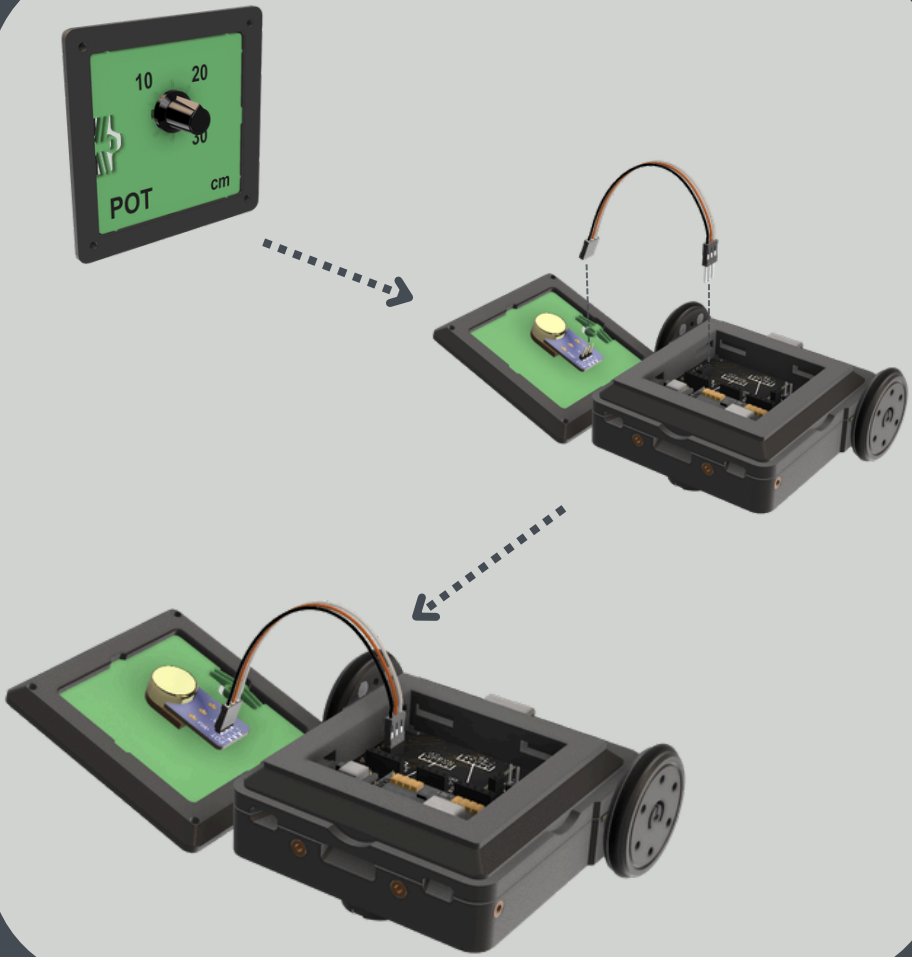
FRONT PLATE BREAKDOWN

Right Plate

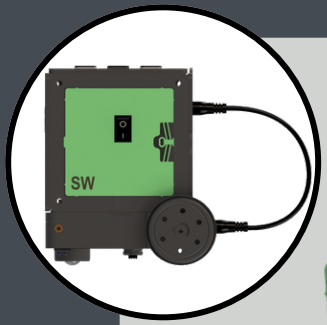


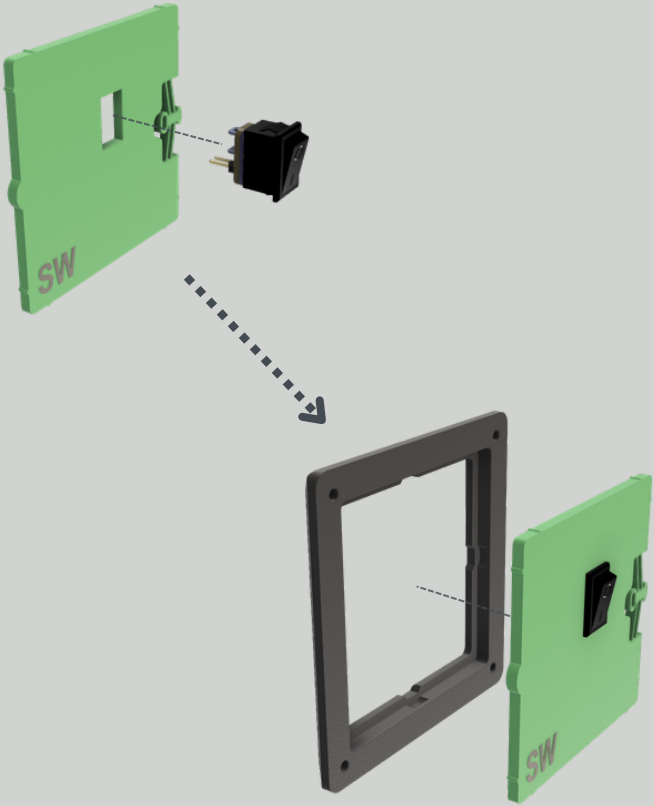


RIGHT PLATE BREAKDOWN

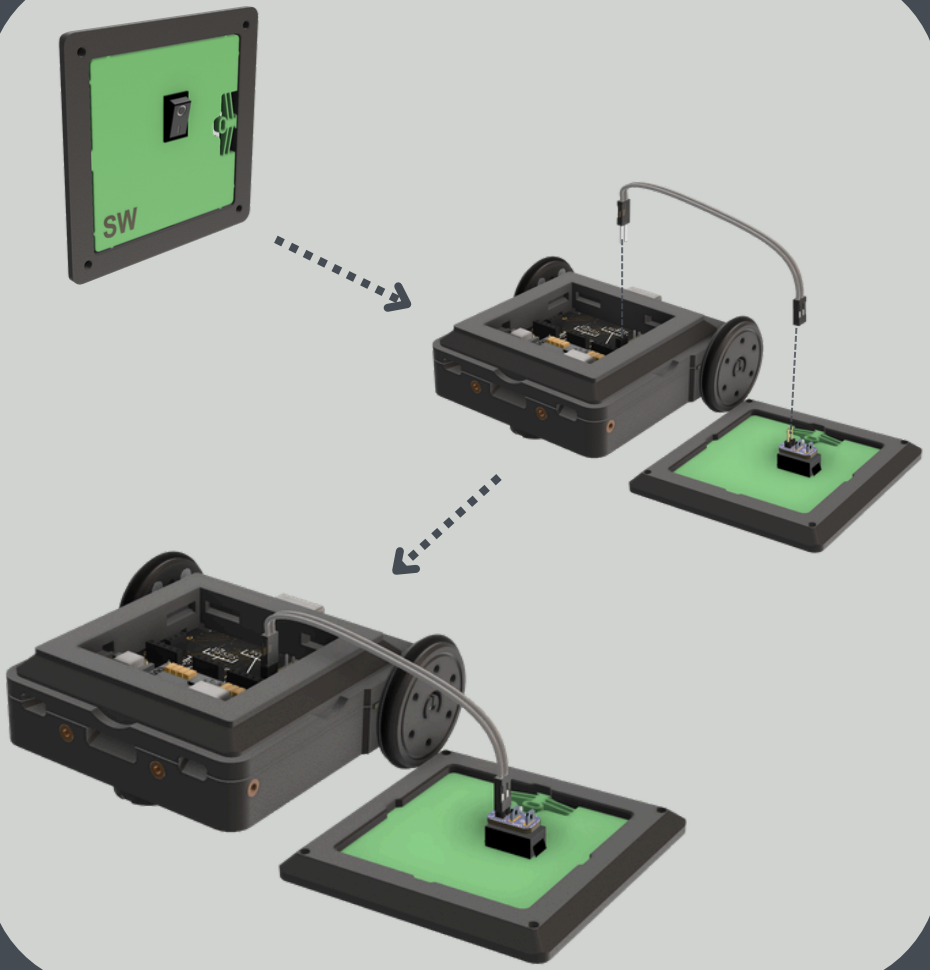


Left Plate

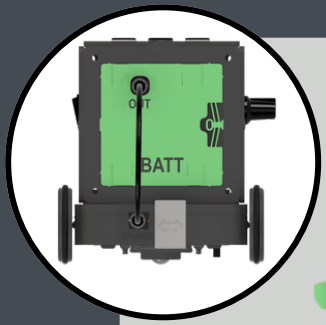


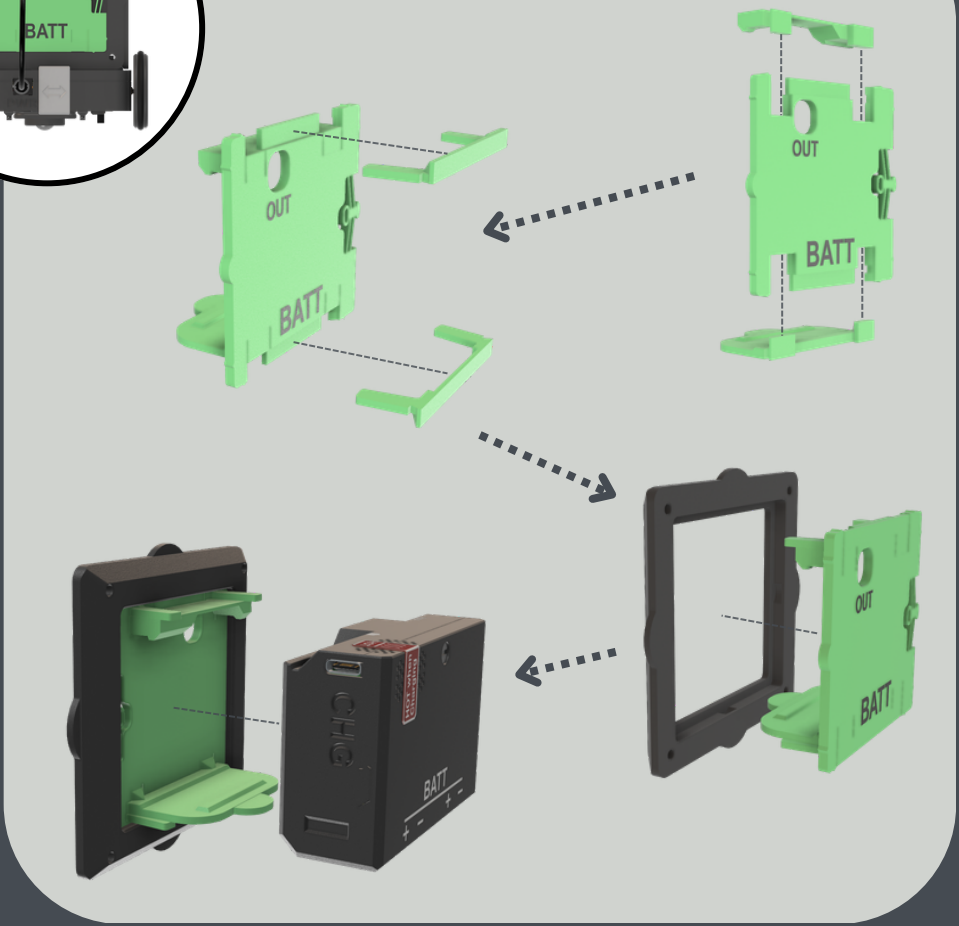


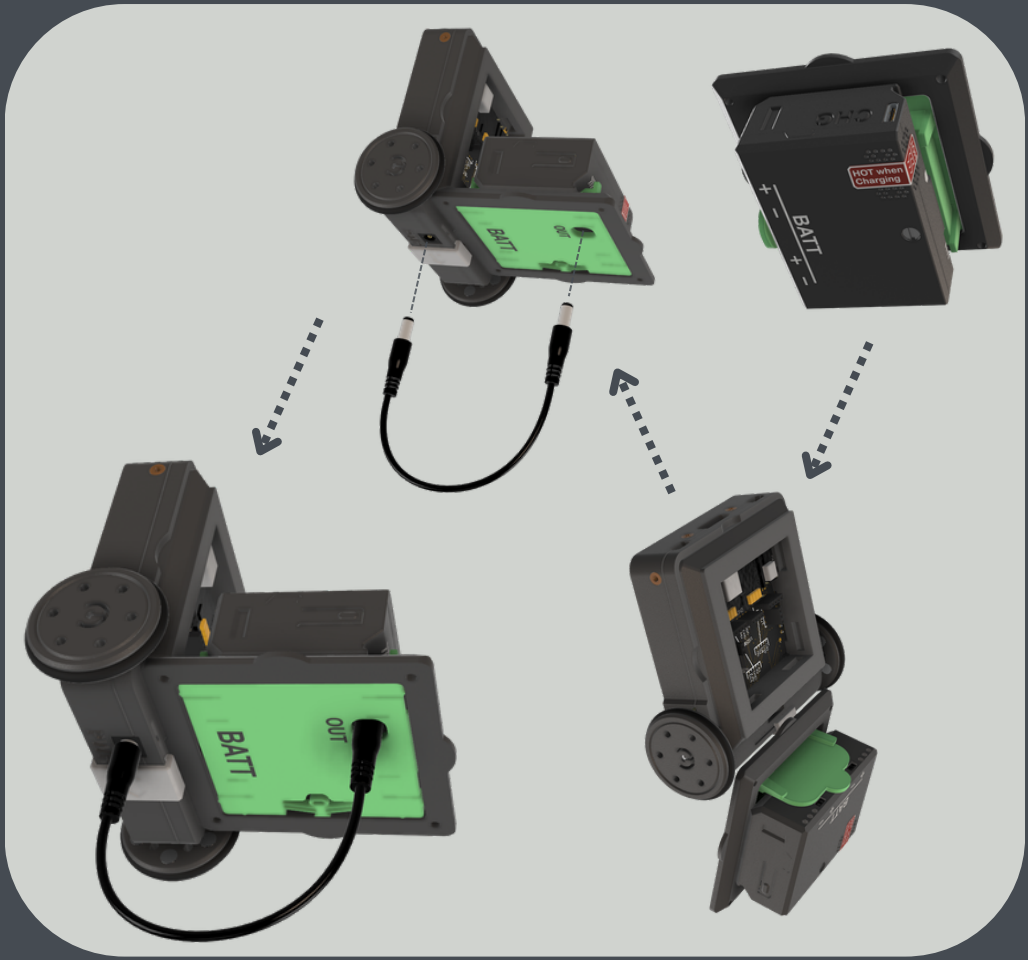
LEFT PLATE BREAKDOWN



Back Plate

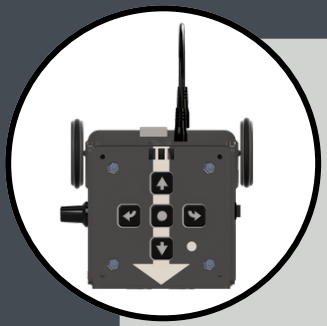


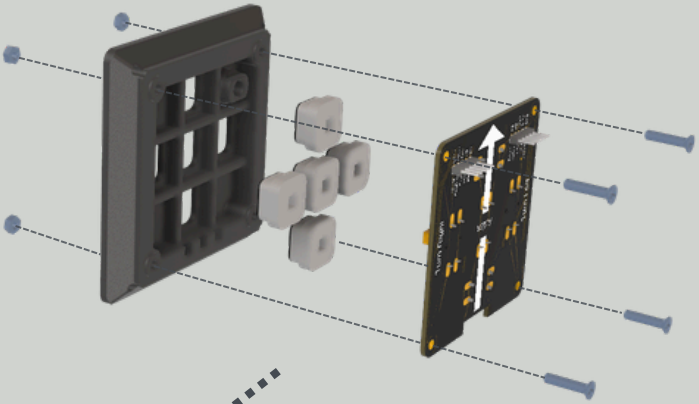


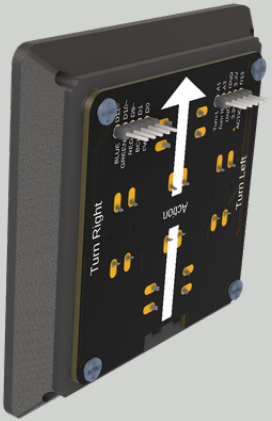


BACK PLATE BREAKDOWN

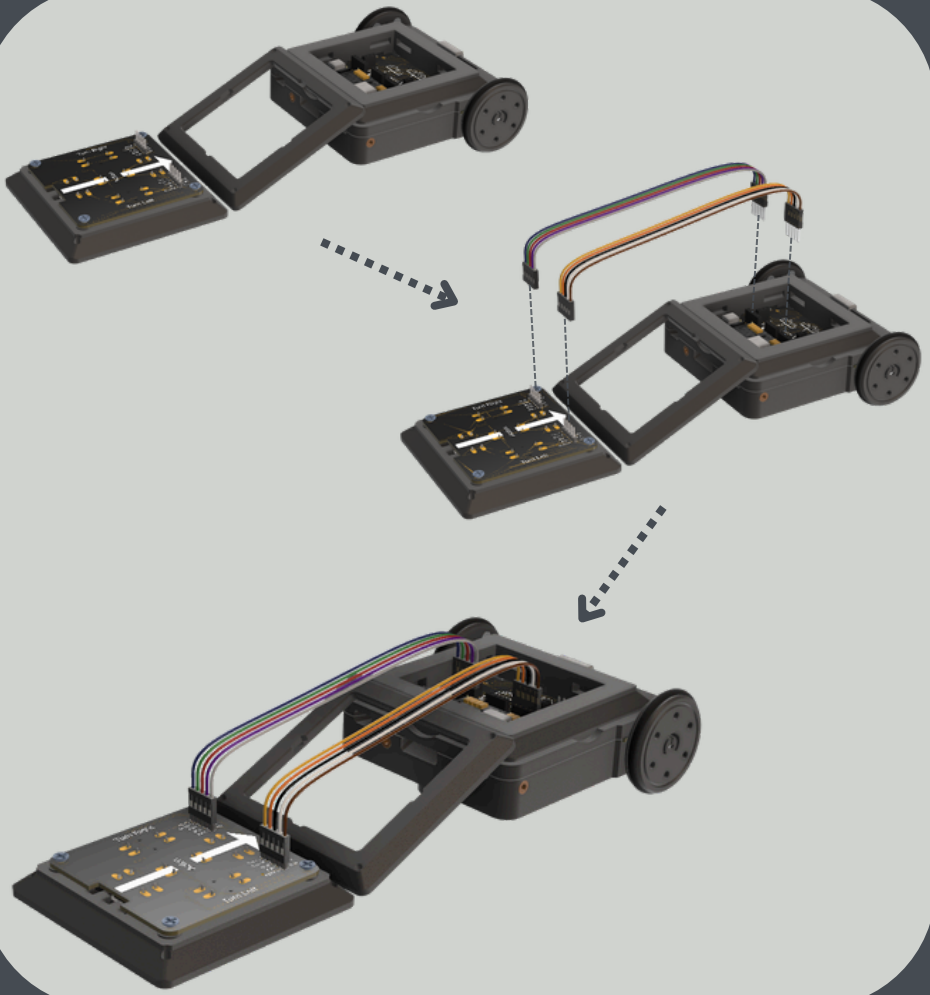
Top Plate



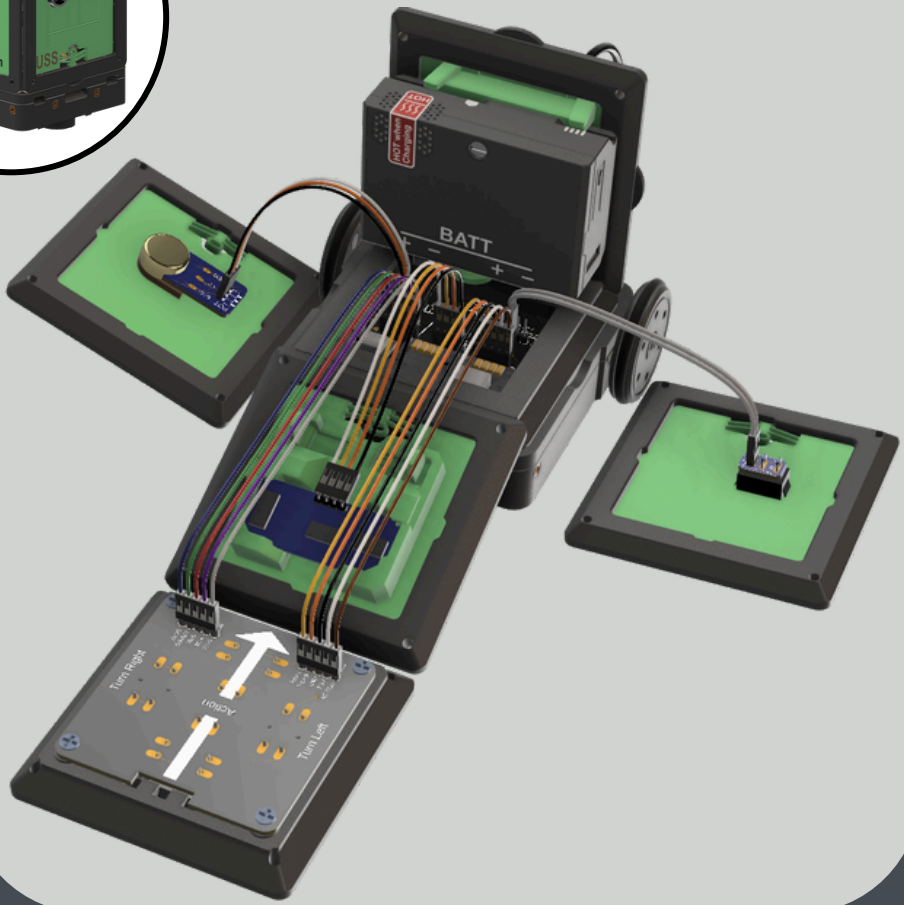




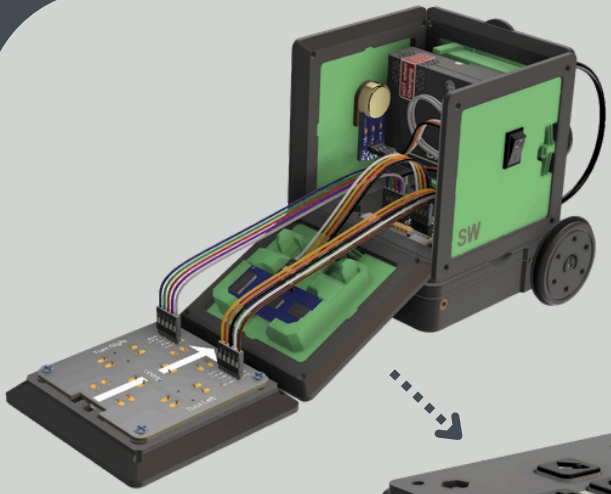
TOP PLATE BREAKDOWN



Final Assembly



FINAL ASSEMBLY

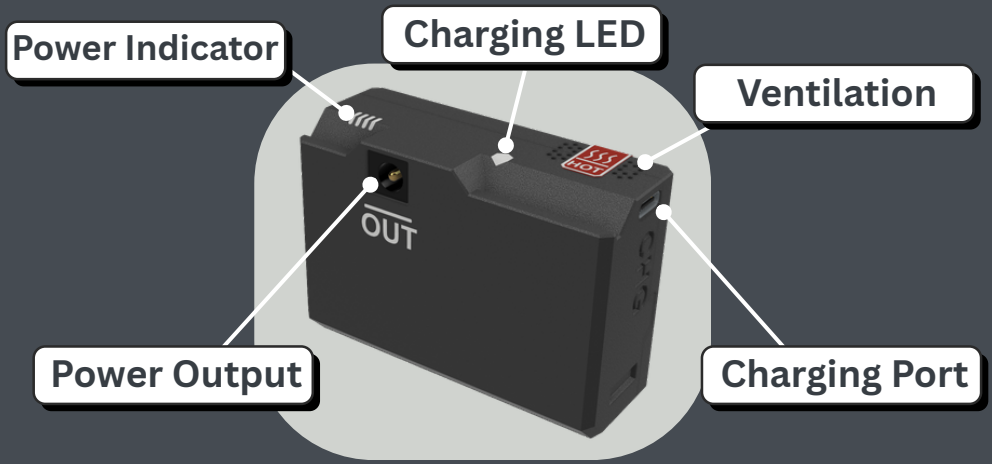


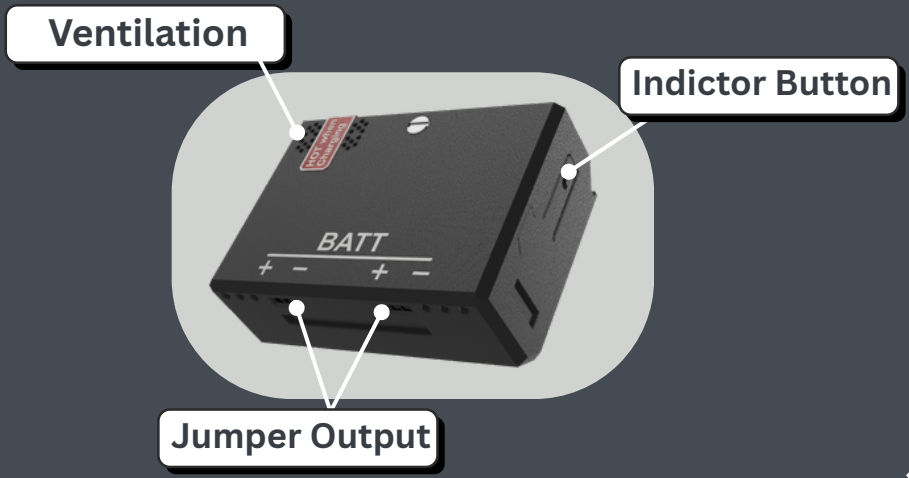
Battery:

- Rechargeable 2S 18650 cell
- Input Charging Voltage = 5V
- Charging Voltage = 8.4 - 9V
- Output Voltage = 6.4 - 8.4 V
- Capacity = 3000 mAh
- Balance charging and overcharged protection with BMS: HX-2S-JH20

1A version = Standard, and charges in +- 5H, can be charged via the supplied USB cable from either phone charger or from a computer USB port.

4A version = Can charge in just over an hour. CAUTION: Do Not charge via a computer, and only use a USB cable and adapter that is rated for 5A or higher.


Power Indicator
Charging LED
Ventilation
OUT
Power Output
Charging Port


Ventilation
Indictor Button
BATT
Jumper Output

Charging LED

Once the USB-C cable is Connected:

Slow Flashing: Not Charging

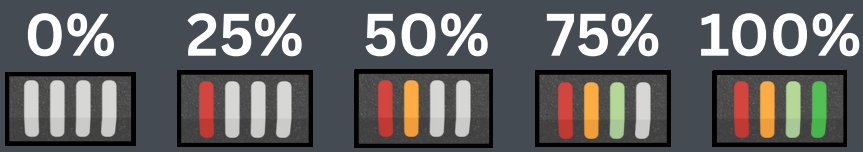
Not Connected

Charging

Finished

Power Indicator

Once the Indicator Button is Pressed:



Below 6.45V (red light flashing)

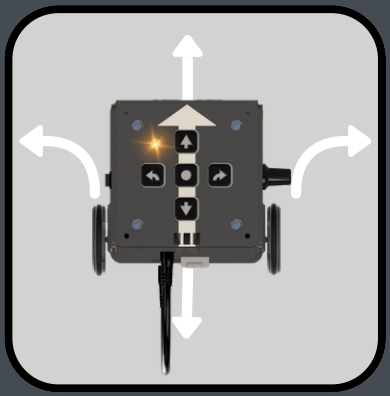
6.45–6.7V

6.7–7.2V

7.2–7.9V

7.9–8.4V

Stand by



How to do it:

- Turn On.
- Wait for Standby mode.
- Turn the Potentiometer to select the step length.
- Short Press any of the four directions.

What it Does:

This is the Starting Mode, where you take control! Press any of the direction buttons to make the MagBot move or turn in that direction — it's the perfect way to get familiar with your robot's movements.

Color Identifier

How to do it:

- Select Color Sensor.
- Turn On.
- Wait for Yellow LED.
- Short Press Action Button.

What it Does:

In this mode, also known as Chameleon Mode, the MagBot looks at the color beneath it and changes its RGB light to match that color.

Color Actions

How to do it:

- Select Color Sensor.
- Turn On.
- Wait for Standby mode.
- Place on Color.
- Short Press Action Button.
- Wait for Rainbow LED.
- Long Press Action Button.

What it Does:

In this mode, the MagBot checks the color beneath it and performs the action linked to that color.

Color		Action
	Yellow	**Decelerate**
	Blue	**Turn Right**
	Cyan	**Turn 360 degrees**
	Green	**Accelerate**
	Red	**Stop**
	Purple	**Turn Left**
	Black	**Follow the Line**

Record

How to do it:

- Get Magbot into Stand by mode.
- Rotate the Potentiometer to the desired Step size.
- Long Press Action Button, the RGB LED will Turn Orange when you can let go.
- Short Press any sequence of directional buttons.

What it Does:

In this mode, you can press the four directional buttons to create a list of moves. When you press the Action button, the MagBot will perform them in order.

Execute

How to do it:

- Get Magbot into Record mode.
- Short press any list of directional buttons.
- Short Press the Action Button to start.

What it Does:

When this mode is active, the MagBot performs the list of actions you recorded earlier in Record Mode.

Free Running

How to do it:

- Select IR Sensors
- Short Press Action Button

What it Does:

In this mode, the robot drives around and avoids obstacles, turning in random directions.

Tape Following

How to do it:

- Select IR Sensors
- Place the robot on a black Line.
- Short Press Action Button
- Once a line is detected, the RGB LED will change from Blue to Cyan.

What it Does:

In this mode, the robot uses its two IR sensors to keep a black line centered between them, allowing it to follow the line wherever it goes.

Obstacle Avoidance

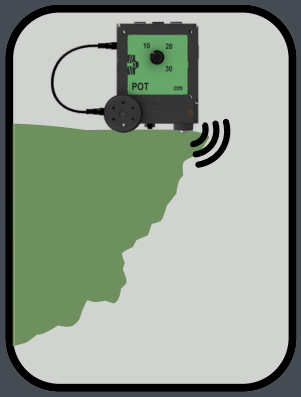
How to do it:

- Place the robot in Line following mode, or.
- Select IR Sensors.
- Short Press Action Button.

What it Does:

This mode is active when the MagBot is following a line or just moving around freely.

Edge Detection



How to do it:

- Select IR Sensors.
- Short Press Action Button.

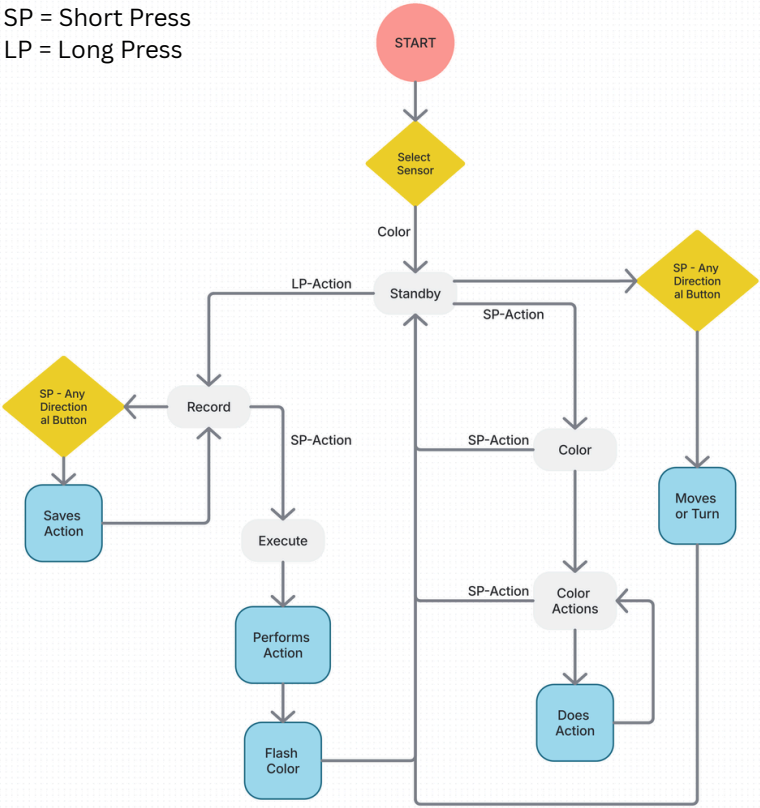
What it Does:

This mode is active only when the Selector Switch is in the IR position and moving around freely.

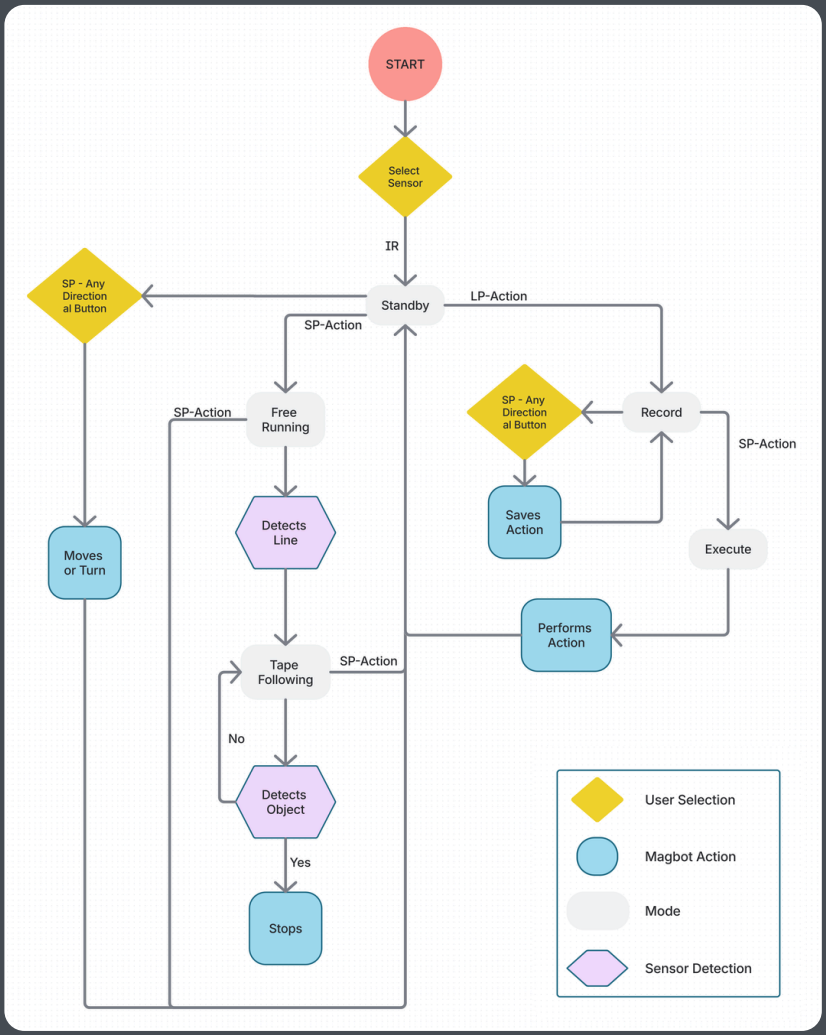
Tip: Always keep your attention on the Magbot, as it may fall off the edge, so be ready to catch it.

Operating

SP = Short Press
LP = Long Press



LOGIC FLOWCHART



Downloads

Arduino IDE

https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-downloading-and-installing/

ESP32 Nano Package

https://www.waveshare.com/wiki/ESP32-S3-Nano

Install CH340 Drivers

https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all

Setup

Arduino IDE

1. Download the Arduino IDE from the QR code.
2. Choose "Just Download", or "Contribute & Download".
3. Install Software with all default settings.

ESP32 Nano Package

1. Open the Arduino IDE.
2. Open Boards Manger -> Search "Nano ESP32" and install the latest version.

Install CH340 Drivers

1. Download CH340 Drivers for the QR code.
2. Run the .exe file
3. On the Pop-up press “Install”.

Install Libraries

1. Open the Arduino IDE.
2. Open the Library Manager.
3. Search for “Adafruit_TCS34725.h” and install.

Upload Code

Connecting MagBot

1. Unplug the Barrel power cable from the MagBot Base.
2. Slide the “Cable Selector” to reveal the USB-C connection.
3. Connect the MagBot via the supplied USB-C cable.

Arduino IDE

1. Open the Arduino IDE
2. Go to “Tools/Board:” and select “Arduino Nano ESP32”.
3. Go to “Tools/COM Port” and select the availible COM port.
4. Open or type the desired code.
5. Press Upload.
6. Wait for the “Done” Command, then unplug.
7. Slide the “Cable Selector” and connect the Power Cable.

Include Libraries

```
#include <Wire.h>
#include "Adafruit_TCS34725.h"
```

Define Pins

```
// ===== IR sensors Pins =====
#define SDA_RIGHT A4 //Either SDA for Color sensor or Right IR sensor
#define SCL_LEFT A5 //Either SCL for Color sensor or Left IR sensor

// ===== Left Motor Pins =====
#define IN1_L 4
#define IN2_L 12
#define ENA_L 6
#define HallL A6

// ===== Right Motor Pins =====
#define IN1_R 7
#define IN2_R 8
#define ENA_R 5
#define HallR A7

// ===== Control Inputs =====
#define POT  A3
```

```
// ===== Buttons =====
#define AB 13 //Action Button
#define FB 0 //Forward Button
#define BB 1 //Back Button
#define TRB A1 //Turn Right Button
#define TLB A2 //Turn Left Button

// ===== RGB LED =====
#define PIN_GREEN 9
#define PIN_RED 10
#define PIN_BLUE 11

// ===== Voltage Divider =====
#define VoltDiv A0 // Measuring the Battery Voltage

// ===== Ultra Sonic Sensor =====
#define TRIG_PIN 3
#define ECHO_PIN 2
```

void setup

```
//Turn off LEDs
analogWrite(PIN_GREEN, 0);
analogWrite(PIN_RED, 0);
analogWrite(PIN_BLUE, 0);

//Buttons
pinMode(AB, INPUT_PULLUP);
pinMode(FB, INPUT_PULLUP);
pinMode(BB, INPUT_PULLUP);
pinMode(TLB, INPUT_PULLUP);
pinMode(TRB, INPUT_PULLUP);

//RGB LED
pinMode(PIN_GREEN, OUTPUT);
pinMode(PIN_RED, OUTPUT);
pinMode(PIN_BLUE, OUTPUT);


//Voltage divider
pinMode(VoltDiv, INPUT);
```

```
//RGB LED
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);

/Potentiometer
pinMode(POT, INPUT);

//Color and IR sensors
pinMode(SDA_RIGHT, INPUT);
pinMode(SCL_LEFT, INPUT);

//Motors and Hall Sensors
pinMode(IN1_L, OUTPUT);
pinMode(IN2_L, OUTPUT);
pinMode(ENA_L, OUTPUT);
pinMode(HallL, INPUT);

pinMode(IN1_R, OUTPUT);
pinMode(IN2_R, OUTPUT);
pinMode(ENA_R, OUTPUT);
pinMode(HallR, INPUT);
```

RGB LED COLOR MEANING

Color	Mode / Meaning
Yellow	**Standby Mode** – Robot waits for input (default state).
Blue	**Free Run Mode** – Random drive, uses Ultrasonic Sensor only for obstacle avoidance.
Cyan	**Tape Follow Mode (IR)** – Follows a black line, stops at edges.
Rainbow	**Color Sensor Standby** – Reads surface color, "chameleon mode."
Green	**Execution Mode** – Plays back recorded sequence (maze solving).
Red	**Record Mode** – Records button inputs for later playback.
Purple	**Calibration Mode** – Adjust turn multipliers, speed, or sensor thresholds.

Calibrating

Color Sensor Calibration

1. Open the Arduino IDE.
2. Connect the Magbot to your computer via the USB-C cable.
3. Upload the “Color_Calibration.ino” code to the magbot
4. Open the Serial Monitor 
5. Place the Magbot on a red square, and update the values in line 24 for the RGB and Clear.
6. Move over to the Yellow square, and update the values in line 25. Do this for every color.
7. Copy lines 24 to 31.
8. Open the Magbot main code “Magbot_StdFullCode.ino” and paste the code you copied over the similar code around lines 50 to 60.
9. Upload this new code to you magbot.

Calibrating

Motor Calibration

1. Open the Arduino IDE.
2. Open the Magbot main code “Magbot_StdFullCode.ino”.
3. Around Line 64 there is a “Motor Calibration & Reference Values” section.
4. Place the Magbot down with some clearance around it, and mark the direction the Magbot is facing.
5. Turn on the Magbot and record 4x Right turns, and press the action button to execute. Mark the new direction the Magbot is facing.
6. If the Magbot turned past the first mark, reduce the “RCal” value, even going negative if required. If the Magbot did not reach the first mark, then increase the “RCal” value.
7. Connect the Magbot to your computer via the USB-C cable.
8. Upload this new code to you magbot, afterwards testing again.
9. Similar for the Left turn, record 4x Left turns, and update the “LCal” value accordingly.
10. Set the Potentiometer to max and press 1x forward, if the bot drifts left, increase the “FwdCal” value.
11. Similar, for driving backwards, if the bot drifts left, increase the “BckCal” value.

For Sales Support:

FUTUREKIDS®
www.ffcaps.co.za

Email: gerda@skillspro.co.za

For Technical Support:

DETRONICS
www.detronics.co.za

23/04/2026