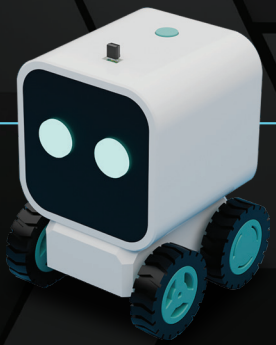
LEGENDIUM

NANO SPARK
Interactive • Programmable Robot

NANO ROBOT
USER INSTRUCTION MANUAL



A fun and educational robotics experience that lets young innovators explore how technology thinks, reacts, and interacts.

Build • Code • Play • Create

TABLE OF CONTENTS

1. Introduction 01
2. Safety Instructions 01
   2.1 Important: Read Before You Begin 01
3. Parts Included 01
   3.1 Mechanical Components 01
   3.2 Electronic Components 03
4. Assembly Instructions of NanoRobot 06
   4.1 Attach Nano Spark Free Wheel to Base Bottom Part 06
   4.2 Attach Nano Spark Control Wheel to DC Motor 07
   4.3 Attach DC Motor to Base Bottom Part 08
   4.4 Place Charging Module on Base Bottom Part 09
   4.5 Install Battery Holder and Battery 09
   4.6 Attach Motor Driver to Motor Clip 10
   4.7 Mount Motor Clip and Connect Motor Wires 12
   4.8 Place UBEC on Base Bottom Part 13
   4.9 Install Power Distribution Board and Connections 14
   4.10 Attach Mounting Clips and Head Bottom to Base Top Part 17
   4.11 Attach Nano Spark Base Top (with Head Bottom) to Nano Spark Base Bottom 19
   4.12 Install Arduino Nano + Expansion Board onto Head Bottom 21
   4.13 Connect Power to Arduino Nano 22
   4.14 Connect Motor Control Signals from Expansion Board to Motor Driver 23
   4.15 Install All Head Sensors 24
   4.16 Attach Eyes Clip to Nano Spark Face Plate 31
   4.17 Connect the RGB LED Cables (do this first, while parts are loose) 32
   4.18 Insert the RGB LED Modules into the Face Plate 33
   4.19 Attach the Face Plate to the Head Bottom 34
   4.20 Assemble Nano Spark Head Top to Head Bottom 35
5. Arduino Nano Pin Connection to Various Sensors 37
   5.1 Pin Assignments 37
   5.2 Unconnected Pins 37
6. Getting Started with Arduino Nano 38
   6.1 Installing the Arduino IDE 38
   6.2 Installing the CH340 Driver 39
   6.3 Connecting the Arduino Nano 40
   6.4 Selecting the Board and Processor 40
   6.5 Selecting the Port 41
   6.6 Uploading Your First Sketch 41
7. Demo Code 42
8. Contact & Support 50

1. Introduction

Welcome to the **Nano Robot** assembly guide! This manual will help you easily assemble, wire, and set up your robot using clear explanations and step-by-step instructions designed for young learners.

By the end of this guide, you will:

- Assemble the entire robot structure.
- Install and connect electronic components.
- Understand each module and its purpose.
- Learn how to upload code using Arduino Nano.
- Test and operate the robot in 3 modes: **IR Control Mode, Music Mode**, and **LDR Mode.**

2. Safety Instructions

2.1 Important: Read Before You Begin

- Work on a clean, flat table to prevent losing small parts.
- Handle electronic parts gently-avoid bending pins or pulling wires too hard.
- Ensure the robot is powered OFF while assembling.
- Do not touch exposed metal contacts with wet hands.
- Use only recommended batteries and charging equipment.
- If something feels stuck, don't force it - check alignment instead.

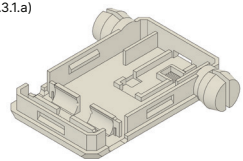
3. Parts Included

3.1 Mechanical Components

Images and detailed views of all mechanical parts are provided in the original part reference file. These include:

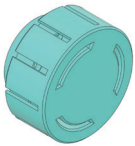
a. Nano Spark Base Bottom

(Fig.3.1.a)



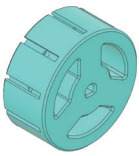
b. Nano Spark Free Wheel

(Fig.3.1.b)



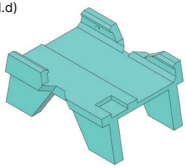
c. Nano Spark Control Wheel

(Fig.3.1.c)



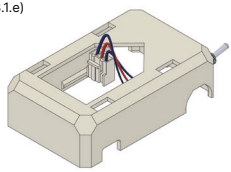
d. Motor Clip

(Fig.3.1.d)



e. Nano Spark Base Top

(Fig.3.1.e)

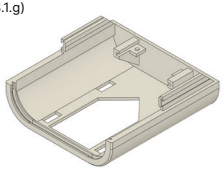


f. Mounting Clip

(Fig.3.1.f)

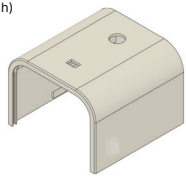
g. Nano Spark Head Bottom

(Fig.3.1.g)



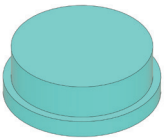
h. Nano Spark Head Top

(Fig.3.1.h)



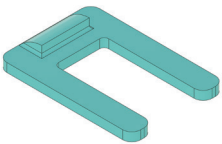
i. Base Bottom

(Fig.3.1.i)



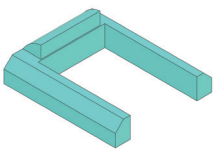
j. Button Cover

(Fig.3.1.j)



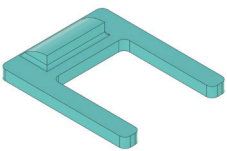
k. LDR Cover

(Fig.3.1.k)



l. TSOP Cover

(Fig.3.1.l)

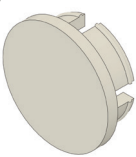


m. Nano Spark Face Plate

(Fig.3.1.m)

n. Eyes Clip

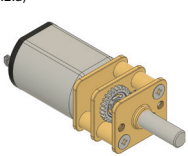
(Fig.3.1.n)



3.2 Electronic Components

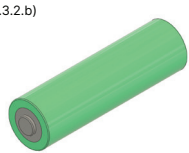
a. DC Motors

(Fig.3.2.a)



b. Battery (14500 Li-on)

(Fig.3.2.b)

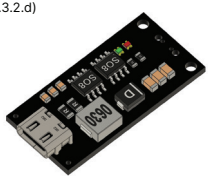


c. Battery Holder

(Fig.3.2.c)

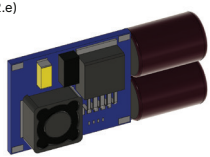
d. Charging Module

(Fig.3.2.d)



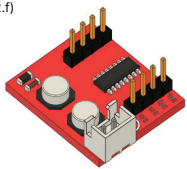
e. UBEC (Voltage Converter)

(Fig.3.2.e)



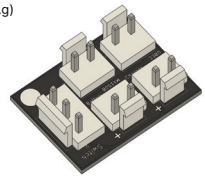
f. Motor Driver (MX1508)

(Fig.3.2.f)



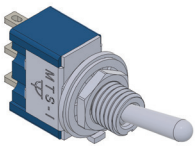
g. Power Distribution Board

(Fig.3.2.g)



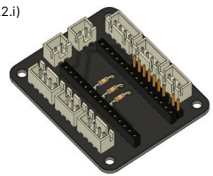
h. Toggle Switch

(Fig.3.2.h)



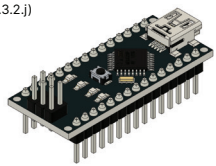
i. Expansion Board

(Fig.3.2.i)



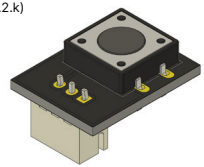
j. Arduino Nano

(Fig.3.2.j)



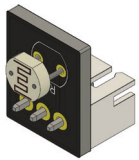
k. Button Module

(Fig.3.2.k)



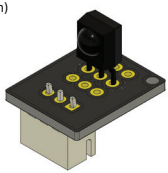
l. LDR Module

(Fig.3.2.l)



m. TSOP Module

(Fig.3.2.m)

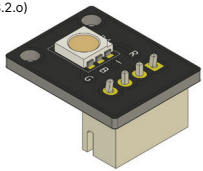


n. Buzzer Module

(Fig.3.2.n)

o. RGB LED Modules

(Fig.3.2.o)



p. IR Remote

(Fig.3.2.p)

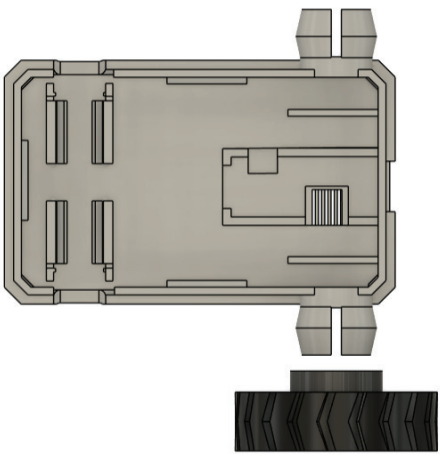
4. Assembly Instructions of NanoRobot

4.1. Attach Nano Spark Free Wheel to Base Bottom Part

a. Place one **Nano Spark Free Wheel** flat on a stable, clean surface with the hole side facing upward.

b. Take the **Nano Spark Base Bottom** part and align one of the **axle stubs** (the small protruding pins on the base) directly over the hole in the Free Wheel as shown in the figure (Fig.1.a).

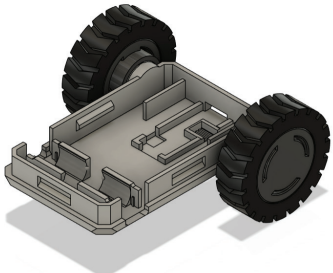
(Fig.1.a)



c. Gently press the **Base Bottom** part straight down so the **axle stub** inserts fully into the wheel's hole until the wheel seats completely against the base.

d. Flip the assembly over (or work from the other side), place the second Free Wheel flat on the surface with its hole facing upward, then align and press the remaining axle stub on the **Base Bottom** part into this wheel's hole as shown in the figure (Fig.1.b).

(Fig.1.b)



e. After both wheels are attached, gently spin each one by hand to confirm they rotate smoothly and freely without wobbling or tightness.

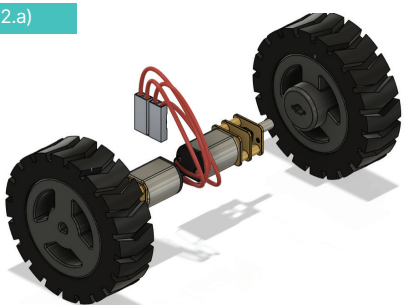
4.2. Attach Nano Spark Control Wheel to DC Motor

a. Examine the **DC Motor** shaft it has a flat "D" shape (one side is flattened).

b. Look at the center hole of the **Control Wheel** it also has a matching "D" shape.

c. Align the flat side of the **Control Wheel's** D-hole exactly with the flat side of the motor shaft (rotate the wheel until the two flats are perfectly parallel) as shown in the figure (Fig.2.a).

(Fig.2.a)



d. Once aligned, gently but firmly push the **Control Wheel** straight onto the motor shaft until it seats completely and the end of the shaft is flush with (or slightly protrudes from) the outer face of the wheel.

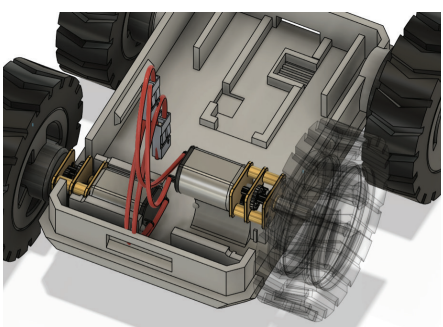
e. Lightly try to twist the wheel by hand there should be no play or slippage; the D-shape prevents the wheel from spinning on the shaft.

4.3. Attach DC Motor to Base Bottom Part

a. Hold the **DC Motor** with the shaft and **Control Wheel** pointing outward (away from the base).

b. Align the motor so that the flat side rests downward directly on the flat bottom surface of the press-fit motor mount (the cylindrical slot at the front of the **Nano Spark Base Bottom** part) as shown in the figure (Fig.3.a).

(Fig.3.a)



c. Gently press the motor straight down into the mount using even pressure on both sides of the motor body. Continue pushing until the motor's top.

d. Try to wiggle the motor gently with your fingers there must be no movement, rocking, or play. A correctly seated motor will feel completely solid.

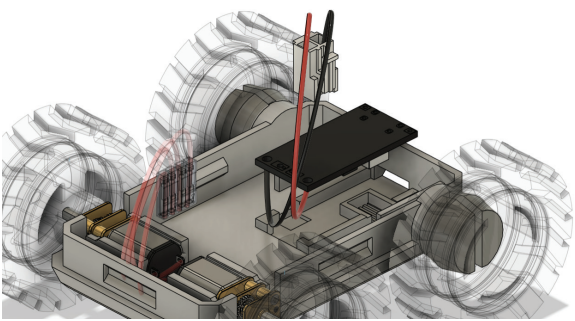
4.4. Place Charging Module on Base Bottom Part

a. Take the **Charging Module** and turn it upside-down (USB-C port facing downward).

b. Rotate the module so that its wires (red and black) exit toward the front of the base (the same direction as the **DC motor** and **Control Wheel**).

c. Align the **Charging Module** with the corresponding rectangular slots in the **Nano Spark Base Bottom** part (located just behind the battery holder area) as shown in the figure (Fig.4.a).

(Fig.4.a)



d. Gently lower the module straight down into position.

e. Press evenly on both ends until it fully seats with a light click and sits perfectly flush with the surrounding surface.

f. Lightly tug upward to confirm it is locked securely with no movement.

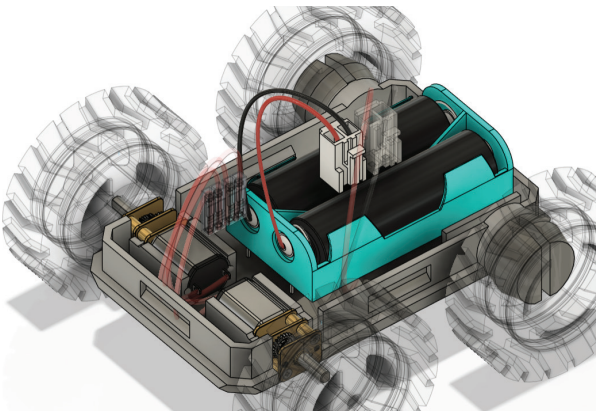
4.5. Install Battery Holder and Battery

a. Insert the **battery** into the **Battery Holder** (observe correct polarity if marked; usually the spring side is negative).

b. Position the loaded **Battery Holder** onto the designated area above the **Charging Module** on the **Nano Spark Base Bottom** part.

c. Orient the **Battery Holder** so that the wires coming out of it face toward the DC motors (this keeps the wires short and neat) as shown in the figure (Fig.5.a).

(Fig.5.a)



d. Confirm the holder is secure with no wobble and that the wires are not pinched or stretched.

4.6. Attach Motor Driver to Motor Clip

a. Examine the **Motor Clip**. It has three raised supports with horizontal grooves:

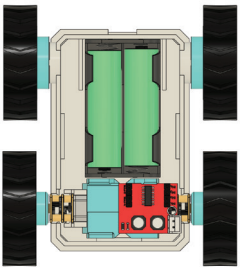
i. One end has grooves/supports on both sides (fully supported end).

ii. The opposite end has a groove and support on only one side (the partially open end).

b. Identify the **JST connector** (small white power plug) on the **Motor Driver** board.

c. Orient the **Motor Driver** board so that the **JST connector** will end up on the partially open side (the side with only one support) once insertion is complete as shown in the figure (Fig.6.a).

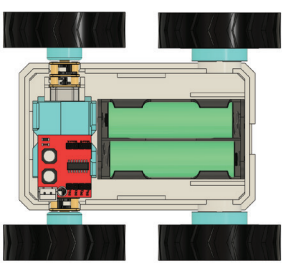
(Fig.6.a)



d. Start sliding from the partially open end (the end with only one groove/support). Carefully insert the edge of the **Motor Driver** board into the single available groove first, then align the other two edges with the remaining grooves as you push.

e. Slide the board from the partially open end toward the fully supported end (typically right-to-left or left-to-right) depending on your view follow the reference image direction as shown in the figure (Fig.6.b)

(Fig.6.b)



until it stops firmly against the two supports at the fully supported end.

f. Confirm correct installation:

i. The board sits completely flat and secure with almost no play.

ii. The **JST connector** protrudes freely on the partially open side (the side missing the second support) and is easily accessible.

iii. The opposite edge of the board is fully backed by the two supports.

4.7 Mount Motor Clip and Connect Motor Wires

a. Take the **Motor Clip** that already has the **Motor Driver** correctly inserted (JST connector protruding from the partially open side the side with only one support).

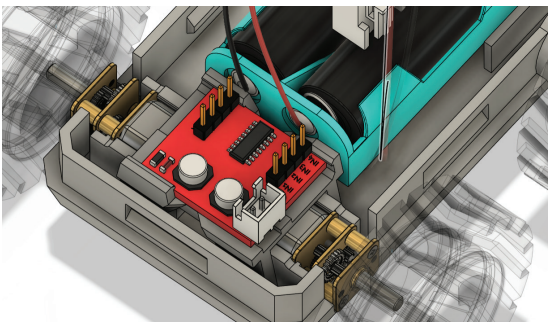
b. Position the entire **Motor Clip** assembly over the **DC motor** (the motor is already press-fitted in the base).

i. Orient the clip so that the **JST connector** faces outward and sits close to the outer edge of the **Nano Spark Base Bottom** part (this gives easy access for the power cable later).

ii. The opposite side of the clip (the fully supported side with two grooves) will face inward, toward the center of the base.

c. Lower the **Motor Clip** straight down until it seats over the two small **motor clamps/posts** on the base (they are located on both sides of the press-fitted motor) as shown in figure (Fig.7.a).

(Fig.7.a)

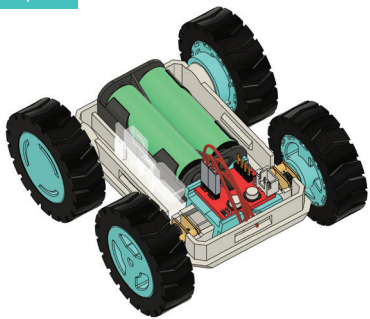


d. Route the wires:

i. Take the red and black wires from the **Charging Module and Battery Holder** and pass them upward through the small groove/notch on the **Motor Clip** that is on the inward side (the fully supported side with two supports).

ii. Take the two thin **Motor Wires** coming from the **DC motor** and route them upward through the small groove on the opposite side (the partially open side where the **JST** is located) as shown in the figure (Fig.7.b).

(Fig.7.b)



e. Connect the **motor wires** to the **Motor Driver**:

i. Insert the wire labeled or coming from **Motor A terminal** into the **Motor-A pins** on the **Motor Driver**.

ii. Insert the wire labeled or coming from the **Motor B terminal** into the **Motor-B pins** on the **Motor Driver**.

4.8. Place UBEC on Base Bottom Part

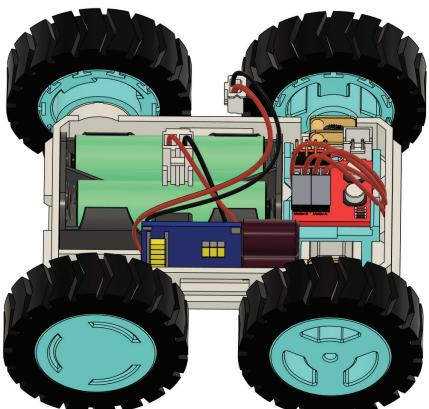
a. Identify the **UBEC** (the small voltage regulator board with two capacitors and input/output wires).

b. Look at the **battery holder** already installed the power switch will later be on the right side of the **battery** (when viewing the base with the DC motor at the front).

c. Place the **UBEC** on the left side of the **battery holder** (the side opposite to where the switch will be).

d. Orient the **UBEC** so that the two large **capacitors** (the tall cylindrical components) face toward the front, i.e., closest to the **Motor Driver** and **DC motor** as shown in the figure (Fig.8.a).

(Fig.8.a)



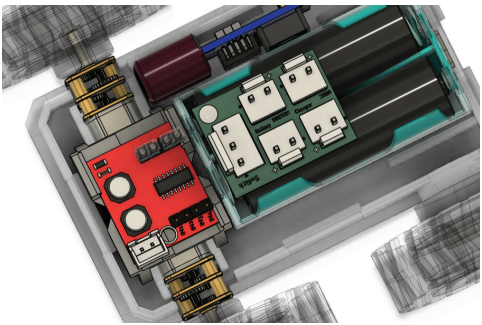
i. This is the only correct orientation: the **capacitors** will be near the motor area, while the input/output wires or pads will naturally point toward the rear/center for easier connection later.

e. Gently press the **UBEC** straight down into its designated slot or onto the double-sided tape/foam pad (if provided). It should sit flush and level.

4.9 Install Power Distribution Board and Connections

a. Place the **Power Distribution Board** directly on top of the **battery** as shown in the figure (Fig.9.a).

(Fig.9.a)



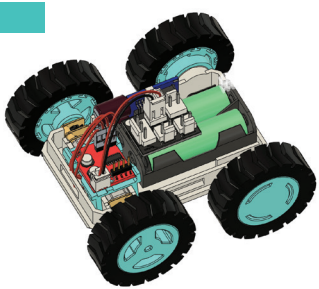
b. Connect power to the **Motor Driver**:

i. Take the provided **JST cable** (one end is **JST-VH**; the other end is **JST-XH**).

ii. Plug the **JST-VH** end into the "**Motor Driver**" header on the **Power Distribution Board.**

iii. Plug the **JST-XH** end into the white **JST connector** on the **Motor Driver** (the **connector** that protrudes near the edge of the base from the partially open side of the **Motor Clip**) as shown in the figure (Fig.9.b).

(Fig.9.b)



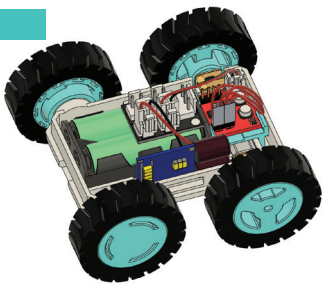
iv. Ensure both sides click securely.

c. Connect the **UBEC**:

i. Locate the **UBEC** input wires (the two wires exiting on the same side as the two large capacitors).

ii. Plug these **input wires (capacitor-side)** into the **"UBEC" connector** on **the Power Distribution Board** (red → +, black →-) as shown in the figure (Fig.9.c).

(Fig.9.c)

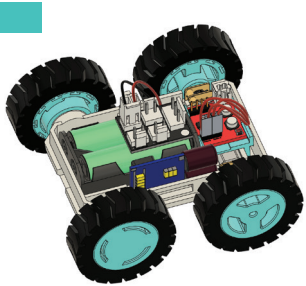


d. Connect the **Charging Module**:

i. Take the red and black wires from the **Charging Module** (routed through the Motor Clip's inward groove).

ii. Plug them into the "CHARGER" connector on the **Power Distribution Board** (red → +, black →-), as shown in the figure (Fig.9.d).

(Fig.9.d)

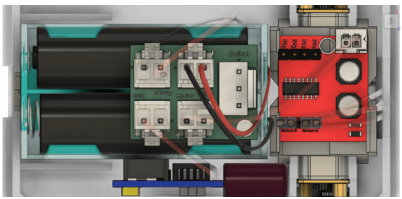


e. Connect the **Battery Holder**:

i. Take the red and black wires from the **Battery Holder**.

ii. Plug them into the "**BATTERY**" connector on the **Power Distribution Board** (red → +, black →-) as shown in the figure (Fig.9.e).

(Fig.9.e)



4.10. Attach Mounting Clips and Head Bottom to Base Top Part

a. Take the four small black **Mounting Clips.**

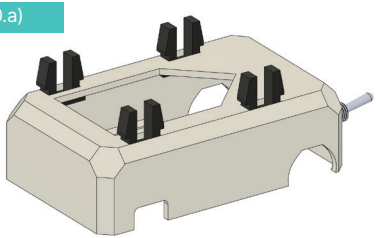
b. On the **Nano Spark Base Top part** (upper shell):

i. Locate the four rectangular slots around the central opening.

ii. From underneath the **Base Top**, push each **Mounting Clip** upward into its slot until it clicks securely.

iii. Correct orientation: after insertion, the open side of the clip faces outward (toward the outer edge of the robot), and the small hook/lip on the top of the clip protrudes slightly outward over the edge of the opening (not inward) as shown in the figure (Fig.10.a).

(Fig.10.a)

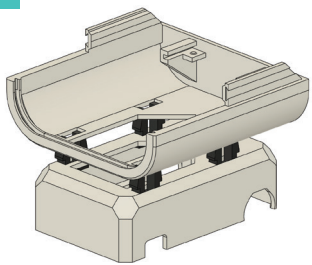


c. Take the **Nano Spark Head Bottom**.

i. Find the arrow-shaped marker (or triangular notch) on the bottom surface.

ii. This arrow points toward the rear of the **Head Bottom** so when correctly installed, the arrow will point toward the back of the robot (the side opposite the Control Wheel / front of the robot) as shown in the figure (Fig.10.b).

(Fig.10.b)

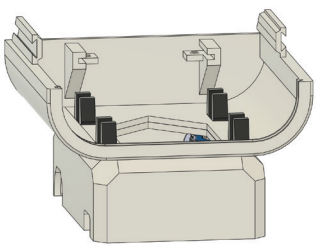


d. Lower the **Head Bottom** into the **Base Top** part.

i. Rotate it so the arrow points directly backward (align it with the rear of the **Base Top / opposite** the C**ontrol Wheel** side).

ii. Once aligned, press firmly and evenly downward all around the rim of the **Head Bottom** as shown in the figure (Fig.10.c).

(Fig.10.c)



iii. While pressing down, use your fingers from underneath the **Base Top** to gently push the four **Mounting Clips** inward if needed - this helps the rim of the **Head Bottom** slide under the clip hooks more easily.

iv. You will hear/feel four clear clicks as the edge of the **Head Bottom snaps** under the outward-facing hooks of the four **Mounting Clips**.

e. Check:

i. The **Head Bottom** sits perfectly flush and cannot be lifted.

ii. The arrow on the **Head Bottom** clearly points to the rear.

iii. All four clips are fully engaged with their hooks visible on the outside.

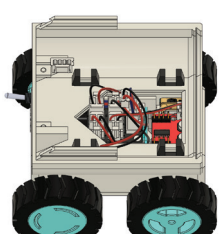
4.11. Attach Nano Spark Base Top (with Head Bottom) to Nano Spark Base Bottom

a. Prepare the **Base Top**:

i. Make sure the **Nano Spark Head Bottom** is already locked in place and its arrow points toward the rear of the **Base Top**.

b. Route the cables from the **Base Bottom** upward through the large central hole before closing as shown in the figure (Fig.11.b)

(Fig.11.b)



i. **Power Distribution Board** lift it gently and pass the entire board upward through the hole.

ii. **Switch cable** (the cable with the toggle switch) feed it through the same hole.

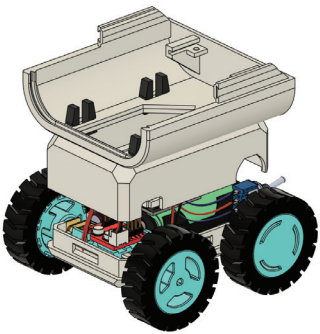
iii. **UBEC** output cable (a **JST-XH** or loose red/black wires) also passes through the hole.

c. Align and close the shell:

i. Hold the **Base Top** above the **Base Bottom**.

ii. Lower the **Base Top** straight down while keeping the cables inside. You will feel the edges start to engage as shown in the figure (Fig.11.a).

(Fig.11.a)



d. Press firmly all around the edge (front, sides, and rear) until you hear multiple clicks the top and bottom are now locked together with no gaps as shown in the figure (Fig.11.c).

(Fig.11.c)

e. Final internal placement inside the **Base Top.**

i. Take the **Power Distribution Board** and place it in the arrow-shaped recess/gap at the front of the **Head Bottom** (the arrow on the Head Bottom points toward the rear, so the wide part of the arrow-shaped gap is at the front the board fits perfectly there).

ii. Connect the **switch cable** to the designated "SWITCH" header on the **Power Distribution Board**.

4.12. Install Arduino Nano + Expansion Board onto Head Bottom

a. First, mount the **Arduino Nano** onto the **Expansion Board** (do this before sliding anything into the head):

i. Take the **Expansion Board** and place it on a clean surface with the component side up.

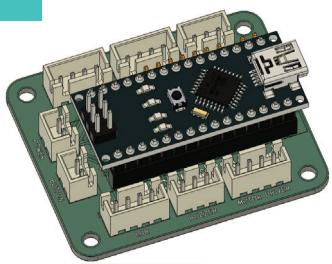
ii. Look for the white silkscreen outline and the small "USB" mark on the board.

iii. Align the **Arduino Nano** so that its mini-B USB port faces the same direction as the "USB" mark (this will point toward the rear of the robot later).

iv. Carefully line up all the Nano's pins with the two rows of female headers on the **Expansion Board**.

v. Press the **Arduino Nano** straight down evenly until it seats fully all pins must be completely inserted with no bent pins and the Nano sitting flat and parallel to the **Expansion Board** as shown in the figure (Fig.12.a).

(Fig.12.a)



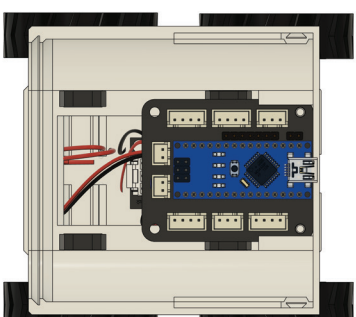
b. Now slide the combined **Arduino Nano + Expansion Board** into the **Head Bottom**:

i. Hold the **Nano Spark Head Bottom** (already fixed inside the Base Top). You will see two parallel horizontal grooves/slots on the top surface.

ii. Take the assembled **Expansion Board** (with **Arduino Nano** on top) and orient it so the mini-B USB port of the **Arduino Nano** faces the rear (the same direction the arrow on the Head Bottom is pointing).

iii. Starting from the front, slide the edges of the **Expansion Board** into the two grooves and push gently but firmly toward the rear until it stops and clicks/locks into place. The board should sit perfectly flat and centered as shown in the figure (Fig.12.b).

(Fig.12.b)



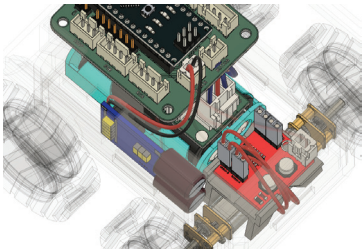
4.13. Connect Power to Arduino Nano

a. Locate the **UBEC** output cable it is a short cable with a 2-pin **JST-XH connector** (small white plug, 2.54 mm pitch).

b. Find the matching 2-pin **JST-XH** header on the **Expansion Board** (clearly labeled “POWER”).

c. Plug the **JST-XH connector** from the **UBEC** output directly into this header on the **Expansion Board** as shown in the figure (Fig.13.a).

(Fig.13.a)



i. The connector is keyed/polarized, so it only fits one way.

ii. Push until it clicks firmly into place.

d. Give the connector a very gentle tug to confirm it is fully locked and cannot pull out accidentally.

4.14. Connect Motor Control Signals from Expansion Board to Motor Driver

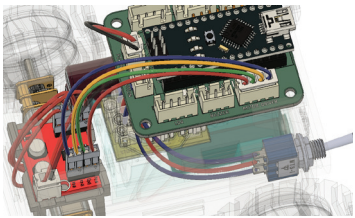
a. Take the provided motor control cable:

i. One end has a single 4-pin **JST-XH connector**.

ii. The other end splits into two separate 2-pin Dupont female connectors (one pair for IN1/IN2, the other for IN3/IN4).

b. Plug the 4-pin JST-XH connector firmly into the "**MOTOR DRIVER**" header on the **Expansion Board** (it is keyed and only fits one way) as shown in the figure (Fig.14.a).

(Fig.14.a)



c. Route the cable neatly downward through the central opening toward the **Motor Driver** as shown in the figure (Fig.14.a).

d. Connect the two Dupont pairs to the **Motor Driver** (MX1508 module):

i. The first 2-pin Dupont connector → push onto the **IN1 and IN2 pins** (polarity does not matter at this stage it can be swapped later if direction is reversed).

ii. The second 2-pin Dupont connector → push onto the **IN3 and IN4 pins.**

e. Ensure all connectors are fully seated:

i. **JST-XH** end is clicked securely on the **Expansion Board**.

ii. Both Dupont pairs are pushed all the way down onto the **Motor Driver** pins with no bent or floating pins.

f. Gently tug each connector to confirm that nothing comes loose.

4.15. Install All Head Sensors

a. Work with the **Nano Spark Head Top** lying flat on the table (component side up) throughout the whole process.

b. Phase 1 Identify and connect the correct cable to each sensor first.

i. LDR Module (Refer figure (Fig.15.a)).

1. Cable: one end 3-pin **JST-XH** → connects to **LDR sensor** other end 4-pin **JST-XH** (one pin is blank/unused) → will later go to **Expansion Board**.

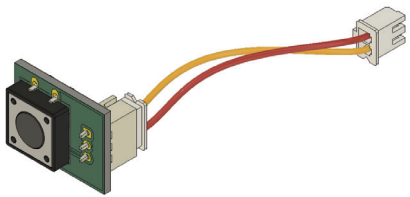
(Fig.15.a)

ii. Button Module (Refer figure (Fig.15.b), (Fig.15.c))

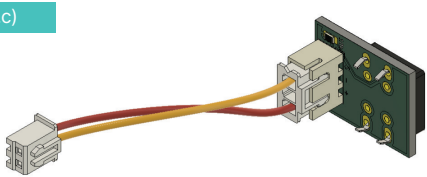
1. Cable: one end 3-pin JST-XH with one pin missing/blocked → connects to Button Module, the other end 2-pin JST-XH → will go to Expansion Board.

2. Plug the 3-pin end (with the blocked pin) into the **Button Module**.

(Fig.15.b)



(Fig.15.c)

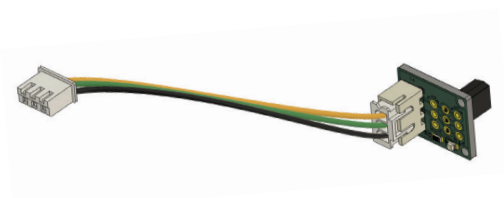


iii. TSOP Module (Refer figure (Fig.15.d), (Fig.15.e)).

1. Cable: both ends are identical 3-pin **JST-XH.**

2. Plug one 3-pin **JST-XH** into the **TSOP. Module**.

(Fig.15.d)

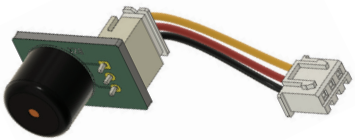


(Fig.15.e)

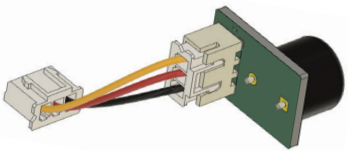
iv. Buzzer Module (Refer figure (Fig.15.f), (Fig.15.g))

1. Cable: both ends are identical **3-pin JST-XH.**
2. Plug one 3-pin **JST-XH** into the Buzzer.

(Fig.15.f)



(Fig.15.g)



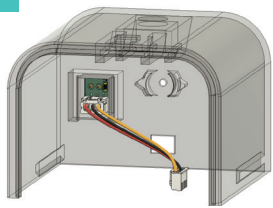
c. All four sensors now have their cables attached and are still loose.

d. Phase 2 Mount the sensors into the Head Top (exact order).

i. LDR

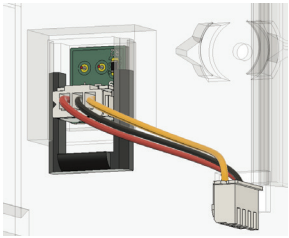
1. Insert the LDR from inside, so the light sensor comes out of the front round hole as shown in the figure (Fig.15.h).

(Fig.15.h)



2. Slide the **smallest clip** from the side until it clicks as shown in the figure (Fig.15.i).

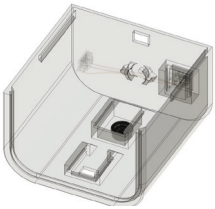
(Fig.15.i)



ii. Button

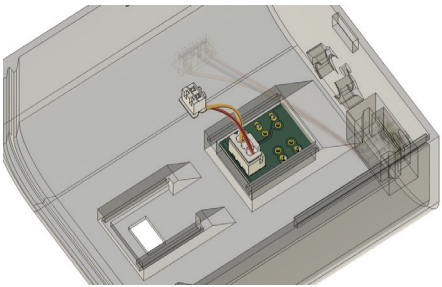
1. Drop the **Button Cap** into the large top-rear hole (it just sits there) as shown in the figure (Fig.15.j).

(Fig.15.j)



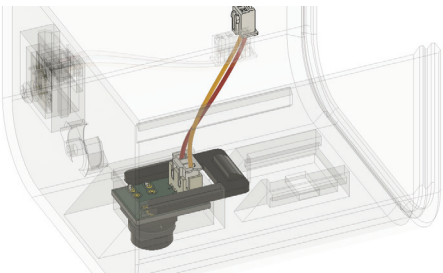
2. Push the **Button Module** up from underneath. as shown in the figure (Fig.15.k).

(Fig.15.k)



3. Slide the **largest clip** from the side (lift the Head Top slightly if needed) until it locks both sides. as shown in the figure (Fig.15.l).

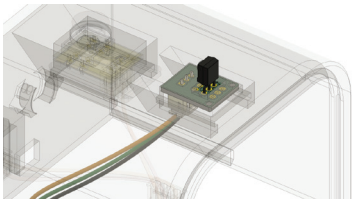
(Fig.15.l)



iii. TSOP

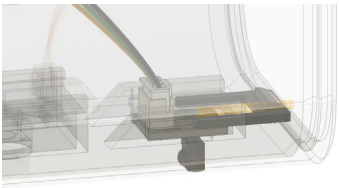
1. Insert the TSOP into the rectangular front holder (sensor facing outward). as shown in the figure (Fig.15.m).

(Fig.15.m)



2. Slide the medium TSOP clip/cover from the side until it clicks. as shown in the figure (Fig.15.n).

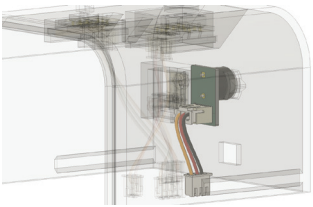
(Fig.15.n)



iv. Buzzer (last)

1. Push the Buzzer straight down into the groove right next to the LDR until it snaps in as shown in the figure (Fig.15.o).

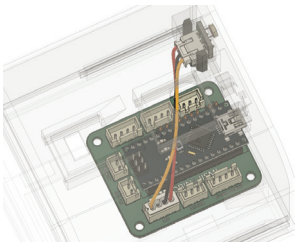
(Fig.15.o)



e. Phase 3 Connect the free ends of the cables to the Expansion Board.

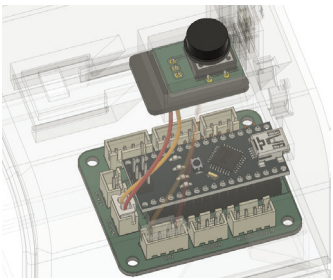
i. **LDR** → plug the 4-pin **JST-XH** (with one blank pin) into the **4-pin LDR header** on the **Expansion Board** (it only fits one way).as shown in the figure (Fig.15.p).

(Fig.15.p)



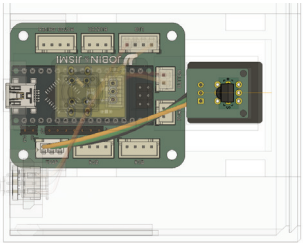
ii. **Button** → plug the 2-pin **JST-XH** into the 2-pin **Button header** on the **Expansion Board** as shown in the figure (Fig.15.q).

(Fig.15.q)



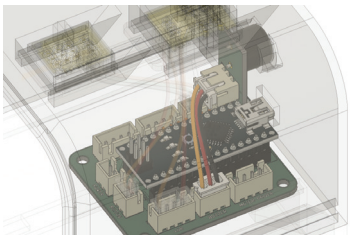
iii. **TSOP** → plug the remaining 3-pin **JST-XH** into the 3-pin **TSOP header** on the **Expansion Board** as shown in the figure (Fig.15.r).

(Fig.15.r)



iv. **Buzzer** → plug the remaining 3-pin **JST-XH** into the 3-pin **BUZZER header** on the **Expansion Board** as shown in the figure (Fig.15.s).

(Fig.15.s)



4.16. Attach Eyes Clip to Nano Spark Face Plate

a. Place the **Nano Spark Face Plate** flat on the table with the front (outer/curved) side facing up.

b. Take the two Eye pieces (each has four small flexible clips or legs on the back).

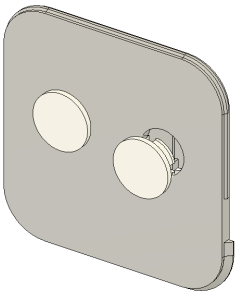
c. From the front side (the side that will be visible on the robot):

i. Align the left Eye directly over the left eye opening.

ii. Gently push it straight down into the hole. The two clips will flex inward as they pass through, then snap outward behind the Face Plate with an audible click as shown in the figure (Fig.16.a).

iii. Repeat the exact same process for the right Eye.

(Fig.16.a)



4.17. Connect the RGB LED Cables (do this first, while parts are loose)

a. Each **RGB LED Module** has a 4-pin **JST-XH male connector**.

b. You have two **4-pin JST-XH extension cables** (female on both ends).

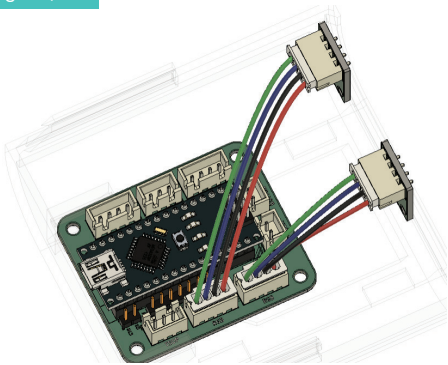
c. For each **RGB LED Module** (Refer figure (Fig.17.a):

i. Plug one end of an extension cable into the **RGB LED Module's connector**.

ii. Plug the other end into one of the two **RGB headers on the Expansion Board**.

iii. Repeat for the second module. → Both RGB LEDs are now fully connected and have enough cable length.

(Fig.17.a)



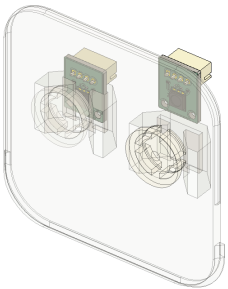
4.18 Insert the RGB LED Modules into the Face Plate

a. Take the **Face Plate** (eyes already installed).

b. From the inside/back, slide each **RGB LED Module** into its pocket directly behind the eyes (LEDs facing forward toward the eyes) as shown in the figure (Fig.18.a).

c. Push gently until the boards sit flat and are held by the side clips/grooves.

(Fig.18.a)



4.19 Attach the Face Plate to the Head Bottom

a. Correct orientation of the parts (Refer figure (Fig.19.a), (Fig.19.b)):

i. The **Head Bottom** has a long horizontal groove at the very front edge.

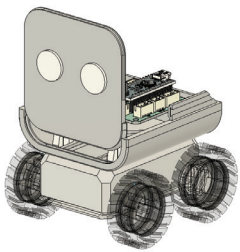
ii. The **Face Plate** has a matching raised horizontal rail/lip along its bottom inner edge.

b. Attaching (Refer figure (Fig.19.c), (Fig.19.d)):

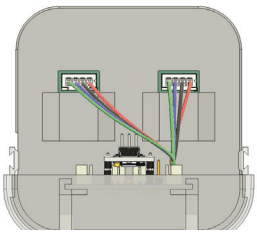
i. Hold the **Face Plate** with the eyes facing you.

ii. Slide the **Face Plate** to the **Head Bottom** such that raised rail on the bottom of the **Face Plate** hooks into the groove on the front of the **Head Bottom.**

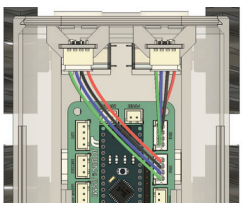
(Fig.19.a)



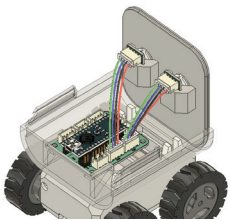
(Fig.19.b)



(Fig.19.c)



(Fig.19.d)



4.20 Assemble Nano Spark Head Top to Head Bottom

a. Place the robot body upright (Control Wheel = front, charging port = rear).

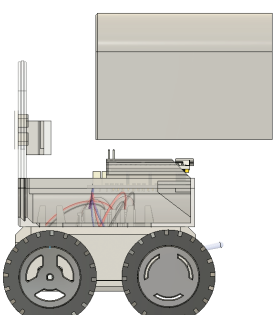
b. Hold the fully equipped **Head Top** so that the **Face Plate** faces exactly forward (same direction as the **Control Wheel**).

c. Look at the **Head Bottom part** (that is already locked inside the Base Top with the four **mounting clips**) as shown in the figure (Fig.20.a).

i. On the left and right walls of this **Head Bottom**, you will see two short vertical rails.

ii. Each rail starts roughly at the rear (back) of the **Head Bottom** and runs forward only to about the center of the plate (so they cover only the rear half of the total length).

(Fig.20.a)

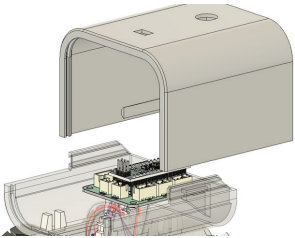


d. Correct starting position (Refer figures (Fig.20.b), (Fig.20.c)):

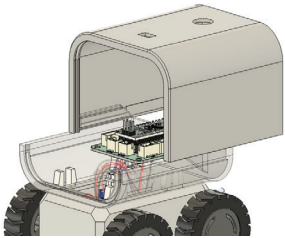
i. Bring the bottom edge of the **Head Top** to approximately the middle length of the **Head Bottom.**

ii. Slightly adjust **Head Top** until the internal vertical grooves on the left and right sides of the Head Top line up perfectly with the rear (starting) ends of the two short rails of the **Head Bottom.**

(Fig.20.b)



(Fig.20.c)



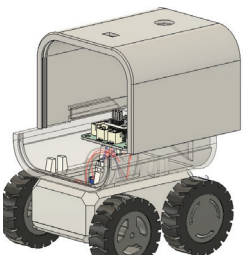
e. When the grooves are perfectly aligned with the rear ends of the rails as shown in the figure (Fig.20.d).

i. Keep the **Head Top** level and slide it straight downward (more like sliding it forward-and-down along the rails).

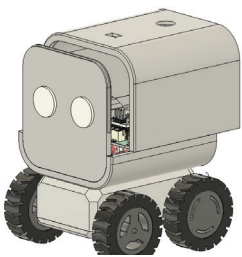
ii. The short rails will guide the **Head Top** smoothly.

iii. Push evenly until the Head Top drops fully into place, and you hear/feel two clear clicks at the front - the locking tabs have snapped securely under the top edge of the **Face Plate** as shown in the figure (Fig.20.e). Resulting in completing the **Nano Spark Robot** as shown in the figure (Fig.20.f & Fig.20.g).

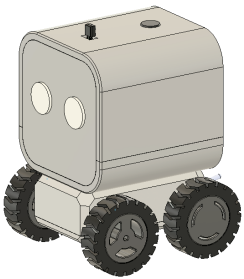
(Fig.20.d)



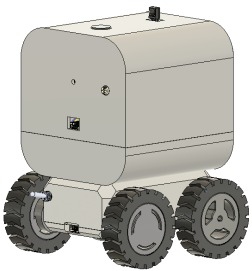
(Fig.20.e)



(Fig.20.f)



(Fig.20.g)



5. Arduino Nano Pin Connection to Various Sensors

5.1 Pin Assignments:

- D2: Connected to the button for input detection.
- D4: Connected to the buzzer for audio output.
- D5: Connected to Motor A positive terminal for motor control.
- D6: Connected to Motor A negative terminal for motor control.
- D7: Connected to the blue LED for visual output.
- D8: Connected to the red LED for visual output.
- D9: Connected to Motor B positive terminal for motor control.
- D10: Connected to Motor B negative terminal for motor control.
- D11: Connected to the green LED for visual output.
- D12: Connected to the TSOP module for infrared signal reception.
- A7: Connected to the LDR (Light Dependent Resistor) for light sensing.

5.2 Unconnected Pins:

- **D3, D1, A0, A1, A2, A3, A4, A5, A6:** Not connected but available for use, with headers provided for potential future connections to additional sensors or components.

6. Getting Started with Arduino Nano

The **Arduino Nano** is a compact, breadboard-friendly microcontroller board based on the ATmega328P, ideal for beginners and advanced users alike. We'll cover everything from installing the Arduino IDE to configuring the board, including specific instructions for boards with the CH340 USB-to-serial chip (common in many **Arduino Nano** variants, including official ones).

Note: These instructions assume you're using a genuine or compatible **Arduino Nano**. If your board is a clone, the steps should still apply, but verify hardware details. The guide is platform-agnostic where possible, with OS-specific notes.

6.1. Installing the Arduino IDE

The Arduino Integrated Development Environment (IDE) is the software you'll use to write, compile, and upload code (called "sketches") to your board.

a. Go to the official Arduino download page: https://www.arduino.cc/en/software.

b. Select the version compatible with your operating system (Windows, macOS, or Linux). For Windows, choose the installer (.exe) for ease.

c. Download the file and run the installer. Follow the on-screen prompts to complete the installation.

i. On Windows: You may need to approve driver installations during setup.

ii. On macOS: Drag the app to your Applications folder.

iii. On Linux: Extract the archive and run the install.sh script if provided, or follow distribution-specific instructions.

d. Launch the Arduino IDE to verify it's working. You should see a blank sketch window.

For more details on the IDE, including preferences and features, refer to the official Arduino documentation.

6.2. Installing the CH340 Driver

Many **Arduino Nano** boards use the **CH340 chip** for USB-to-serial communication. If your computer doesn't recognize the board (e.g.,no new COM port appears), you'll need to install the **CH340 driver**. Official Arduino boards may include this in the IDE setup, but for clones or older systems, manual installation is required.

Windows 7/10/11

a. Download the **CH340 driver** executable from a reliable source, such as SparkFun:
https://cdn.sparkfun.com/assets/learn_tutorials/8/4/4/CH341SER.EXE.

b. Run the executable.

c. Click "Uninstall" first (to remove any conflicting versions).

d. Then click "Install".

e. Restart your computer if prompted.

f. Connect your **Arduino Nano** via USB and check Device Manager (search for it in the Start menu). Look under "Ports (COM & LPT)" for a new "USB-Serial CH340 (COMX)"entry.

macOS

a. Open Terminal (Go > Applications > Utilities > Terminal).

b. Navigate to the Extensions folder:

i. For macOS 10.9+: cd /Library/Extensions.

ii. For macOS 10.8 or below: cd /System/Library/Extensions.

c. List files with ls | grep usb to check for existing **CH340 drivers**.

d. Remove old drivers (if present) with administrative privileges: sudo rm -rf /Library/Extensions/usb.kext and sudo rm -rf /Library/Extensions/usbserial.kext (enter your password when prompted).

e. Download the macOS driver:
https://cdn.sparkfun.com/assets/learn_tutorials/8/4/4/CH341SER_MAC.ZIP.

f. Unzip the file and run the .pkg installer. Follow the prompts.

g. Restart your Mac.

h. Connect the board and verify in System Information (Apple Menu > About This Mac > System Report > USB) that the CH340 device appears.

Linux (Ubuntu/Debian-based, e.g., Raspberry Pi Raspbian)

a. Update your system: Open a terminal and run sudo apt-get update & sudo apt-get upgrade.

b. **The CH340** driver is often included in modern kernels, but if not:

i. Download the Linux driver: https://cdn.sparkfun.com/assets/learn_tutorials/8/4/4/CH341SER_LINUX.ZIP.

ii. Unzip and navigate to the folder: cd path/to/unzipped/folder.

iii. Build and install: make clean, make, sudo make load.

iv. If issues arise, unload the module: sudo rmmod ch341.

c. Plug in the board and check with dmesg | grep ch34 for recognition (look for "ch34x" or similar).

d. Add permissions if needed: sudo usermod -a -G dialout $USER (replace $USER with your username), then sudo chmod a+rw /dev/ttyUSB0.

e. Log out and back in for changes to take effect.

After installation, open the Arduino IDE, connect the board, and select the port (see Section 5). If the board isn't recognized, try a different USB cable or port.

6.3. Connecting the Arduino Nano

a. Use a Mini-B USB cable (not Micro-USB) to connect the Nano to your computer's USB port.

b. The board should power on, indicated by the red power LED (on the bottom for Nano 2.x or top for Nano 3.0).

c. If the LED doesn't light, check the cable and USB port. No

6.4. Selecting the Board and Processor

a. Open the Arduino IDE.

b. Go to **Tools > Board > Arduino AVR Boards > Arduino Nano.**

i. If "Arduino AVR Boards" isn't listed, install it via **Tools > Board > Boards Manager**. Search for "AVR" and install version 1.8.6 or later.

c. Select the processor:

i. For newer Nano boards (manufactured after January 2018): **Tools > Processor > ATmega328P.**

ii. For older Nano boards (before January 2018) or if you encounter upload errors: **Tools > Processor > ATmega328P (Old Bootloader).**

iii. If unsure, try uploading with each option until it succeeds.

6.5. Selecting the Port

a. With the board connected, go to **Tools > Port**.

b. Select the appropriate port:

i. On Windows: Look for "COMX (**Arduino Nano**)" where X is a number e.g., COM3. Use Device Manager to confirm.

ii. On macOS: Select "/dev/cu.usbserial-XXXX" or similar.

iii. On Linux: Select "/dev/ttyUSB0" or "/dev/ttyACM0".

c. If no port appears, double-check the CH340 driver installation (Section 2), cable, or try restarting the IDE/computer.

6.6. Uploading Your First Sketch

a. Open a sample sketch: **File > Examples > 01.Basics > Blink**. This blinks the built-in LED on pin 13.

b. Click the **Upload** button (right arrow icon) or press Ctrl+U (Windows/Linux) / Cmd+U (macOS).

c. Watch the RX/TX LEDs flash during upload. The status bar should show "Done uploading."

d. If successful, the onboard LED will blink every second. If there's an error (e.g., "avrdude: stk500_recv(): programmer is not respond-ing"), verify board/processor/port selections or try the old bootloader option.

e. Common troubleshooting:

i. Ensure no other software is using the port.

ii. Press the reset button on the Nano just before uploading.

iii. For CH340 issues, reinstall the driver.

7. Demo Code:

```
#include <Arduino.h>  // Include Arduino core library
#include <IRremote.h>  // Include the IRremote library

// Pin definitions
#define RECV_PIN 12   // TSOP IR receiver
#define RGB_RED 8
#define RGB_GREEN 7
#define RGB_BLUE 11
#define MOTOR1_A 5    // Left motor
#define MOTOR1_B 6
#define MOTOR2_A 9    // Right motor
#define MOTOR2_B 10
#define BUTTON_PIN 2  // Button pin with pull-up
#define BUZZER_PIN 4  // Buzzer pin
#define LDR_PIN A7    // LDR pin

// IR receiver setup
IRrecv irrecv(RECV_PIN);
decode_results results;

// IR remote hex codes
const unsigned long Forward = 0xBF40FD02;  // Swapped with Backward
const unsigned long Backward = 0xB748FD02; // Swapped with Forward
const unsigned long Right = 0xFB04FD02;
const unsigned long Left = 0xB847FD02;
const unsigned long Stop = 0xFA05FD02;


// Melody definition (frequencies in Hz and durations in ms)
const int melody[] = {262, 294, 330, 349, 392, 440, 494, 523}; // C4 to C5
```

```
// Duration for motor movement and acceleration (milliseconds)
const int MOVE_DURATION = 200;
const int ACCELERATION_STEPS = 10;
const int ACCELERATION_DELAY = 20; // Time per step (ms)
const int DEBOUNCE_DELAY = 1000; // Debounce delay for button (ms)

// Mode variables
enum Mode { IR_MODE, MUSIC_MODE, LDR_MODE };
volatile Mode currentMode = IR_MODE; // Tracks current mode
volatile unsigned long lastInterruptTime = 0; // For debouncing
const int noteDurations[] = {250, 250, 250, 250, 250, 250, 250, 250}; //
Quarter notes
const int melodyLength = 8;

// LDR thresholds for color changes
const int LDR_LOW = 300;  // Low light threshold
const int LDR_HIGH = 700; // High light threshold

void toggleMode(); // Function prototype for button interrupt handler
void stopMotors(); // Function prototype to stop motors
void setRGBColor(int red, int green, int blue); // Function prototype to set
RGB color
void setup() {
  // Initialize pins
  pinMode(RGB_RED, OUTPUT);
  pinMode(RGB_GREEN, OUTPUT);
  pinMode(RGB_BLUE, OUTPUT);
  pinMode(MOTOR1_A, OUTPUT);
  pinMode(MOTOR1_B, OUTPUT);
  pinMode(MOTOR2_A, OUTPUT);
  pinMode(MOTOR2_B, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT_PULLUP); // Button with internal pull-up
  pinMode(LDR_PIN, INPUT); // LDR pin as input
```

```
  // Initialize IR receiver
  irrecv.enableIRIn();

  // Set up interrupt for button
        attachInterrupt(digitalPinToInterrupt(BUTTON_PIN),     toggleMode,
FALLING);

  // Start serial for debugging
  Serial.begin(9600);
 // Set initial RGB color for IR mode
  setRGBColor(255, 255, 255); // White for IR mode
  Serial.println("Started in IR Control Mode - RGB: White");
}

void toggleMode() {
unsigned long currentTime = millis();
  if (currentTime - lastInterruptTime > DEBOUNCE_DELAY) {
    // Cycle through modes: IR → Music → LDR → IR
    switch (currentMode) {
      case IR_MODE:
        currentMode = MUSIC_MODE;
        Serial.println("Switched to Music Mode - RGB: Purple");
        irrecv.disableIRIn(); // Disable IR receiver in music mode
        stopMotors();
        noTone(BUZZER_PIN);
        setRGBColor(128, 0, 128); // Purple for music mode
        break;
      case MUSIC_MODE:
        currentMode = LDR_MODE;
Serial.println("Switched to LDR Mode - RGB: Cyan");
        irrecv.disableIRIn(); // Disable IR receiver in LDR mode
        stopMotors();
        noTone(BUZZER_PIN);
```

```
setRGBColor(0, 255, 255); // Cyan for LDR mode
      break;
    case LDR_MODE:
      currentMode = IR_MODE;
      Serial.println("Switched to IR Control Mode - RGB: White");
      irrecv.enableIRIn(); // Re-enable IR receiver in IR mode
      stopMotors();
      setRGBColor(255, 255, 255); // White for IR mode
      break;
  }
  lastInterruptTime = currentTime;
 }
}

void setRGBColor(int red, int green, int blue) {
 analogWrite(RGB_RED, red);
 analogWrite(RGB_GREEN, green);
 analogWrite(RGB_BLUE, blue);
}

void stopMotors() {
 analogWrite(MOTOR1_A, 0);

 analogWrite(MOTOR1_B, 0);
 analogWrite(MOTOR2_A, 0);
 analogWrite(MOTOR2_B, 0);
 // Restore mode-specific RGB color after stopping motors
 switch (currentMode) {
  case IR_MODE:
    setRGBColor(255, 255, 255); // White for IR mode
    Serial.println("Stop - RGB: White (IR Mode)");
    break;
  case MUSIC_MODE:
```

```
    setRGBColor(128, 0, 128); // Purple for music mode
    Serial.println("Stop - RGB: Purple (Music Mode)");
    break;
case LDR_MODE:
    setRGBColor(0, 255, 255); // Cyan for LDR mode
    Serial.println("Stop - RGB: Cyan (LDR Mode)");
    break;
 }
}

void accelerateMotor(int motorPinA, int motorPinB, int targetSpeedA, int
targetSpeedB) {
 for (int speed = 0; speed <= max(targetSpeedA, targetSpeedB); speed +=
(255 / ACCELERATION_STEPS)) {
   if (targetSpeedA > 0) analogWrite(motorPinA, speed);
   if (targetSpeedB > 0) analogWrite(motorPinB, speed);
   delay(ACCELERATION_DELAY);
 }
}

void moveForward() {
 setRGBColor(0, 0, 255);  // Blue for forward
 Serial.println("Forward - RGB: Blue");
  accelerateMotor(MOTOR1_B, MOTOR2_B, 255, 255); // Accelerate both
motors forward
 delay(MOVE_DURATION);
 stopMotors();
}

void moveBackward() {
 setRGBColor(0, 255, 0);  // Green for backward
Serial.println("Backward - RGB: Green");
  accelerateMotor(MOTOR1_A, MOTOR2_A, 255, 255); // Accelerate both
motors
```

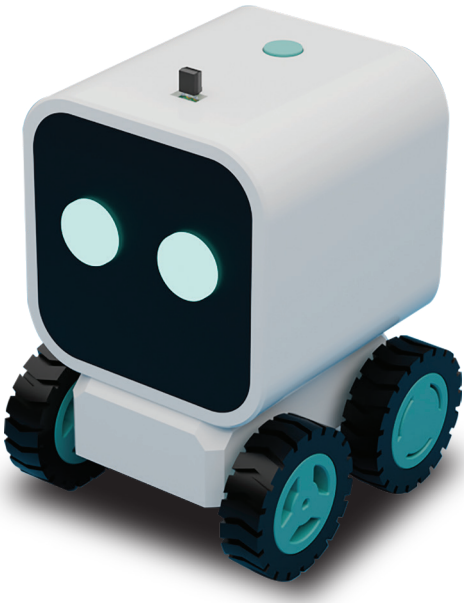
```
    setRGBColor(128, 0, 128); // Purple for music mode
    Serial.println("Stop - RGB: Purple (Music Mode)");
    break;
case LDR_MODE:
    setRGBColor(0, 255, 255); // Cyan for LDR mode
    Serial.println("Stop - RGB: Cyan (LDR Mode)");
    break;
 }
}

void accelerateMotor(int motorPinA, int motorPinB, int targetSpeedA, int
targetSpeedB) {
 for (int speed = 0; speed <= max(targetSpeedA, targetSpeedB); speed +=
(255 / ACCELERATION_STEPS)) {
   if (targetSpeedA > 0) analogWrite(motorPinA, speed);
   if (targetSpeedB > 0) analogWrite(motorPinB, speed);
   delay(ACCELERATION_DELAY);
 }
}

void moveForward() {
 setRGBColor(0, 0, 255);  // Blue for forward
 Serial.println("Forward - RGB: Blue");
  accelerateMotor(MOTOR1_B, MOTOR2_B, 255, 255); // Accelerate both
motors forward
 delay(MOVE_DURATION);
 stopMotors();
}

void moveBackward() {
 setRGBColor(0, 255, 0);  // Green for backward
Serial.println("Backward - RGB: Green");
  accelerateMotor(MOTOR1_A, MOTOR2_A, 255, 255); // Accelerate both
motors
```

```
}

void handleIRControl() {
  if (IrReceiver.decode()) {
    unsigned long irCode = IrReceiver.decodedIRData.decodedRaw-
Data;
    if (irCode != 0) {
      Serial.print("Received IR code: ");
      Serial.println(irCode, HEX);

      // Control motors based on IR code
      switch (irCode) {
        case Forward:
          moveForward();
          break;
        case Backward:
          moveBackward();
          break;
        case Right:
          turnRight();
          break;
        case Left:
          turnLeft();
          break;
        case Stop:
          stopMotors();
          break;
      }
    }
    irrecv.resume(); // Prepare for next IR signal
  } else {
    setRGBColor(255, 255, 255); // Restore white for IR mode when
idle
  }
}
```

```
void handleLDR() {
  int ldrValue = analogRead(LDR_PIN); // Read LDR value (0-1023)
  Serial.print("LDR Value: ");
  Serial.println(ldrValue);

  // Set RGB color based on LDR value
if (ldrValue < LDR_LOW) {
    setRGBColor(255, 0, 0); // Red for low light
    Serial.println("LDR - Low Light: RGB Red");
  } else if (ldrValue < LDR_HIGH) {
    setRGBColor(0, 255, 0); // Green for medium light
    Serial.println("LDR - Medium Light: RGB Green");
  } else {
    setRGBColor(0, 0, 255); // Blue for high light
    Serial.println("LDR - High Light: RGB Blue");
  }
  delay(100); // Small delay to avoid flooding Serial output
}
void loop() {
  switch (currentMode) {
    case IR_MODE:
      handleIRControl();
      break;
    case MUSIC_MODE:
      playMelody();
      break;
    case LDR_MODE:
      handleLDR();
      break;
  }
}
```

8. Contact & Support

For questions or updates, contact:
Email: customer.support@legendium.ai,
info@legendium.ai
Phone: +91 70344 99843



LEGENDIUM

NANO SPARK

Interactive • Programmable Robot

PRIMEVU

PRIMEVU INNOVATIONS PRIVATE LIMITED

Ground Floor, 28/86-E,
Jobin and Jismi IT Services LLP
Kottat, Chalakudy, Kerala, India - 680307

Email: customer.support@legendium.ai, info@legendium.ai
Phone: +91 70344 99843