Ponderosa Planet Rover Challenge

ACEBOTT ESP32 Smart Car with Camera

Arduino IDE

October 2024
Ponderosa Hacks Workshop
Adapted from ACEBOTT ESP32 Manual - https://www.acebott.com/

I. Introduction: Welcome to Ponderosa Planet

The shuttle landed on Planet Ponderosa, revealing a vibrant landscape that sparked excitement in our crew. It was similar to earth in terms of atmosphere and gravity but was densely covered by mountains and tall trees, comparable to Flagstaff's ponderosa pines.

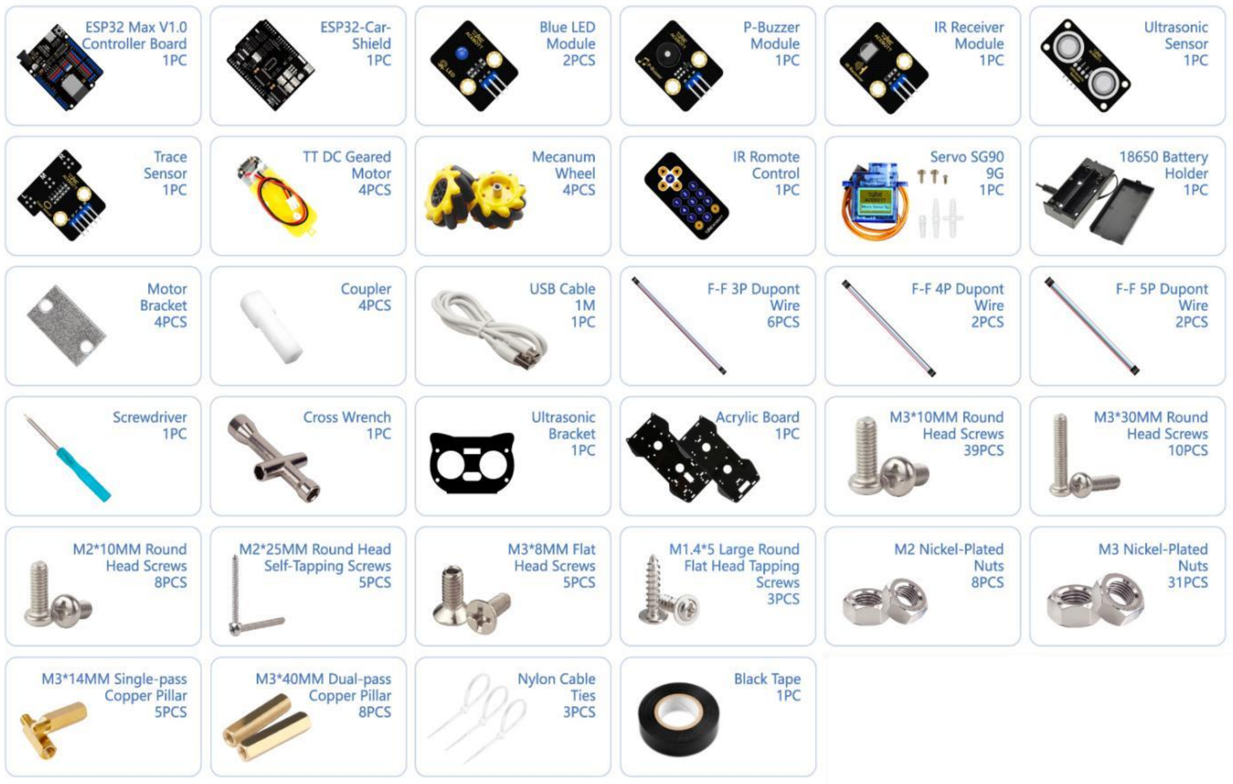
Let's go explore...

Challenge #1 : Assemble the Rover

While others set up camp, I joined Engineer Lumi to assemble the rover.

"Let's get this ready!" Lumi cheered, and we quickly pieced together the rover, eager to explore.

Lumi points to the kit on the front table and says, "Before you start the assembly, you need to check whether the kit is ready according to the table below. This is the first step to successfully assemble the smart car."



Tip: have some scientists work on building rover while the others start on the next challenge to set up for programming to rover.

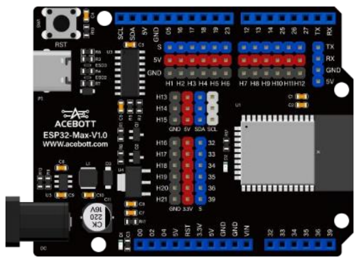
Challenge #2. Configure Rover Controller Board (ESP32)

Nervous and excited, I said, "Another assignment? I can't wait to go explore the planet!"

"This controller board is a shell right now," Lumi said, pointing to the ESP32. "To make it work like a brain, we need to program it to tell the ESP32 what to do."

"How do you program it?" I asked, puzzled.

Lumi then pointed to the ESP32 on the desk in front of him and said, "To program the ESP32 control board, we need to install software that can edit the program on the computer and upload ESP32 resources and programs to the board. We can use Arduino IDE for this."

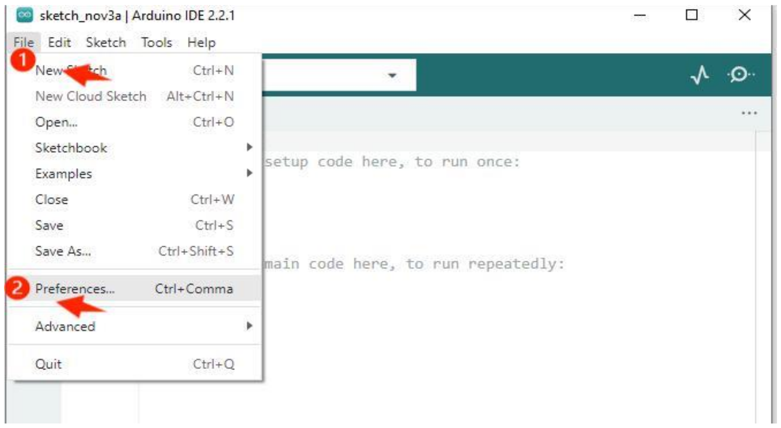


A. Install Arduino IDE onto your computer. Use the links below for instructions based on your computer Operating System:

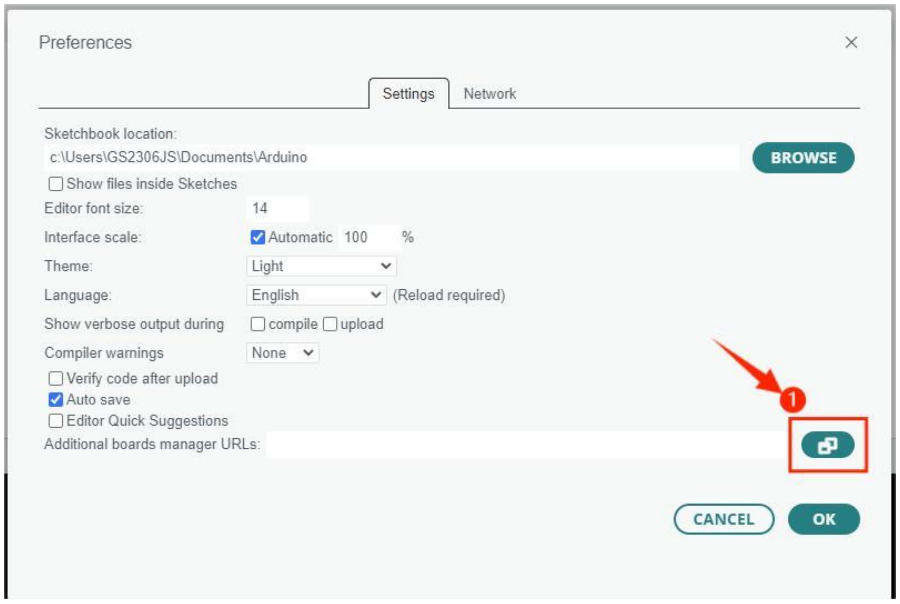
- For Windows, click here.
- For Mac, click here.

B. Once Arduino is installed and opened, follow the steps below:

1. Go to File >Preferences


sketch_nov3a | Arduino IDE 2.2.1
File
Edit
Sketch
Tools
Help
1
New Sketch Ctrl+N
New Cloud Sketch Alt+Ctrl+N
Open... Ctrl+O
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
2
Preferences... Ctrl+Comma
Advanced
Quit Ctrl+Q
setup code here, to run once:
main code here, to run repeatedly:

2. Add the development board management address URLs by clicking on the link icon shown below.


Preferences

Settings | Network

Sketchbook location: c:\Users\GS2306JS\Documents\Arduino

☐ Show files inside Sketches

Editor font size: 14

Interface scale: ☑ Automatic 100 %

Theme: Light

Language: English (Reload required)

Show verbose output during ☐ compile ☐ upload

Compiler warnings: None

☐ Verify code after upload

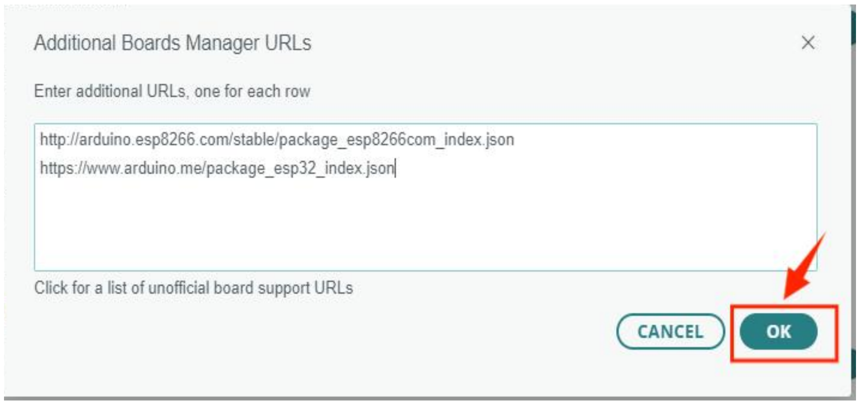
☑ Auto save

☐ Editor Quick Suggestions

Additional boards manager URLs:

3. Copy and paste the following URLs and add it to the "Additional Boards Manager URLs" section and press “OK” to exit out of both boxes.

```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
https://www.arduino.me/package_esp32_index.json
```

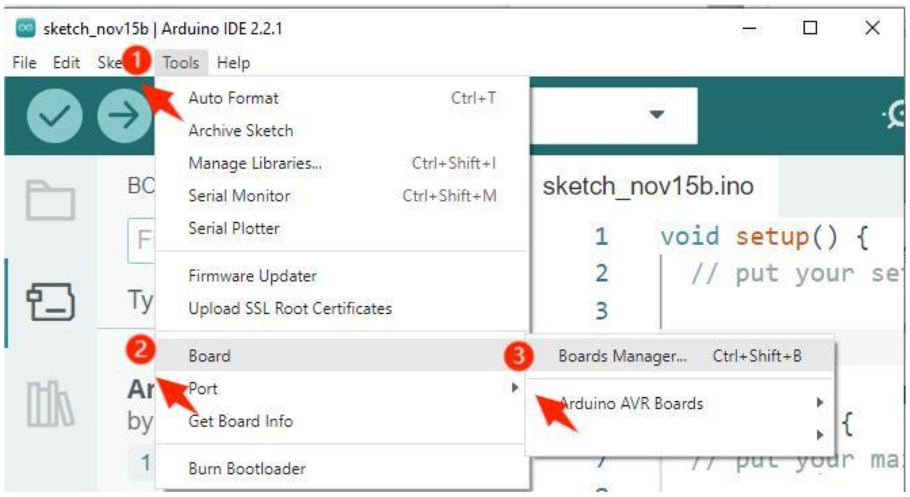

Additional Boards Manager URLs

Enter additional URLs, one for each row

```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
https://www.arduino.me/package_esp32_index.json
```

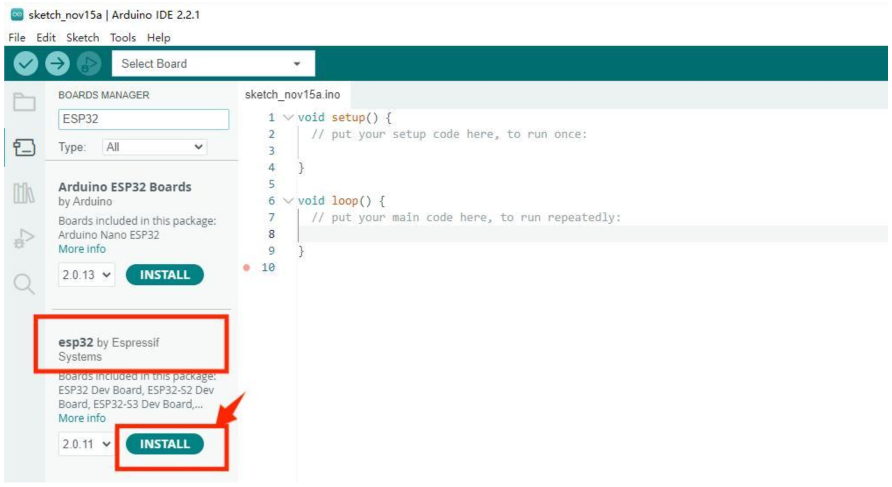
Click for a list of unofficial board support URLs

4. Click Tools > Board > Boards Manager.



```
void setup() {
  // put your se
```

5. Search for "ESP32" in the BOARDS MANAGER's search bar and install “esp32 by Espressif Systems.”

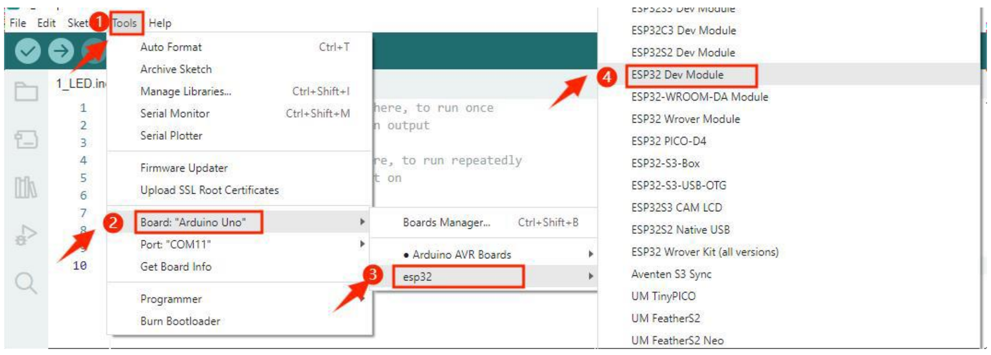


Arduino ESP32 Boards by Arduino
Boards included in this package: Arduino Nano ESP32
More info

esp32 by Espressif Systems
Boards included in this package: ESP32 Dev Board, ESP32-S2 Dev Board, ESP32-S3 Dev Board,...
More info

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

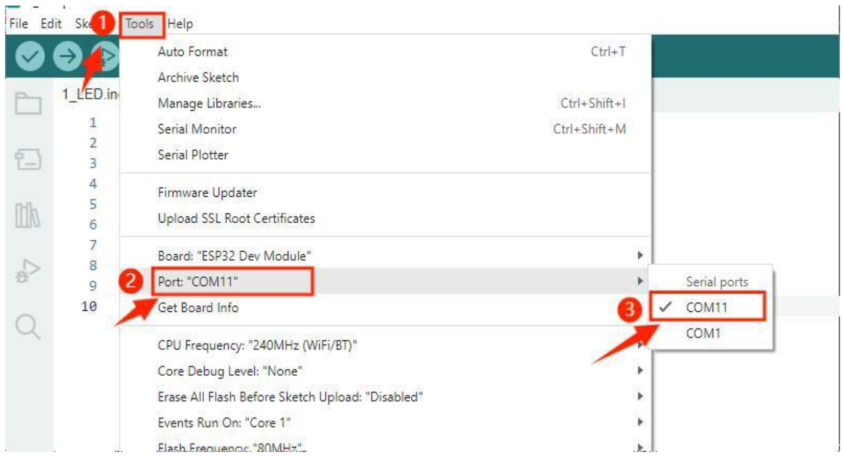
6. Once installation is complete, close the Arduino IDE program and reopen it. When you go to Tools > Board. You will see the esp32 board appear.



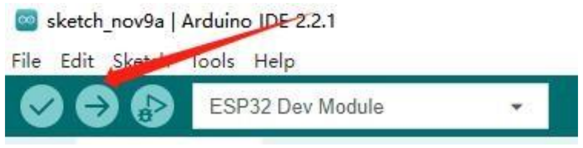
```
here, to run once
n output

re, to run repeatedly
t on
```

Lumi said happily: "Very good! Next, connect the ESP32 board to your computer using the USB cable, and then select the corresponding port number on the Arduino IDE, and finally upload the program to the board."

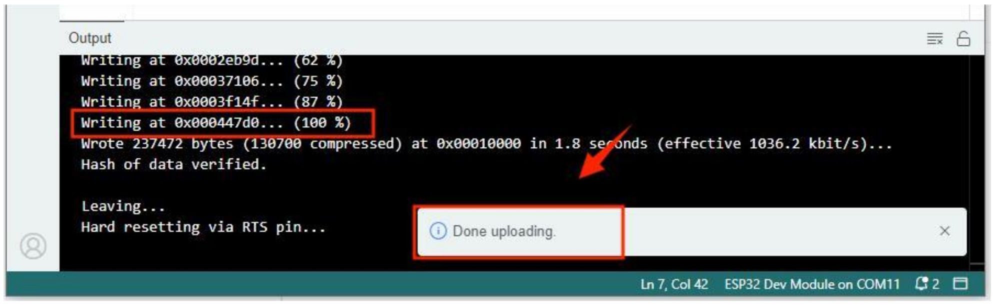
NOTE: Your COM number may be different than the one shown in the screenshot.


File Edit Sketch Tools Help
1
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
Firmware Updater
Upload SSL Root Certificates
Board: "ESP32 Dev Module"
2
Port: "COM11"
Get Board Info
CPU Frequency: "240MHz (WiFi/BT)"
Core Debug Level: "None"
Erase All Flash Before Sketch Upload: "Disabled"
Events Run On: "Core 1"
Flash Frequency: "80MHz"
3
Serial ports
COM11
COM1
1_LED.ino

7. Make sure the battery pack on the rover is set to **OFF**. To upload the program, press the arrow button at the top in Arduino IDE.



8. When the progress displays 100%, the program has been uploaded successfully. Great job!



```
Writing at 0x0002eb9d... (62 %)
Writing at 0x00037106... (75 %)
Writing at 0x0003f14f... (87 %)
Writing at 0x000447d0... (100 %)
Wrote 237472 bytes (130700 compressed) at 0x00010000 in 1.8 seconds (effective 1036.2 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

II. Explore Planet Ponderosa

“Finally, we are done with the rover assembly and configuration. Are you ready to explore our new world?” Lumi asked.

The power of a smart car comes from its motors. To control the car is to control the motors, so you need to supply enough voltage for the motor to turn. The motion of the smart car is not only related to the motor, but also related to its wheels.

Lumi continues, "Smart cars can be difficult to control due to the variety of motion modes. In order to make it easier for you to control the motion of the smart car, we have prepared a powerful library for you, which provides a lot of command tools to control the motion of the smart car. You just need to install this library into the Arduino IDE development environment, and you can easily control the movement of your smart car."

Challenge #3: Install Movement Commands for Rover

Let's get the rover setup with basic movement controls so we can travel through the terrain on Ponderosa Planet.

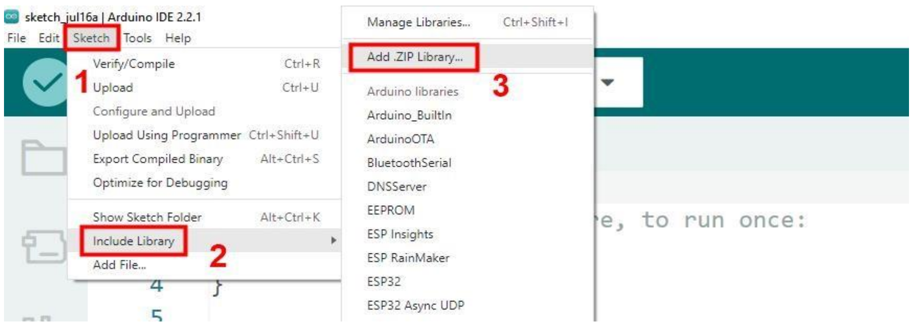
1. Downloads the library zip files from the Ponderosa Hacks Resources site https://ponhack24.github.io/Resources/

   Acebott.zip

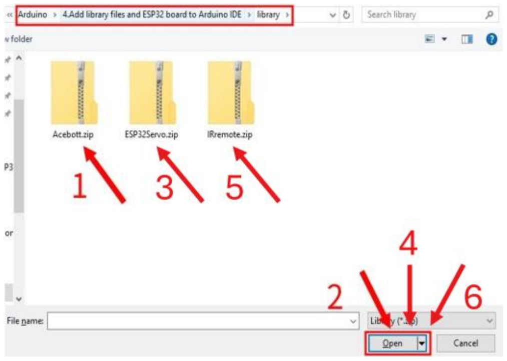
   ESP32Servo.zip

   IRremote.zip

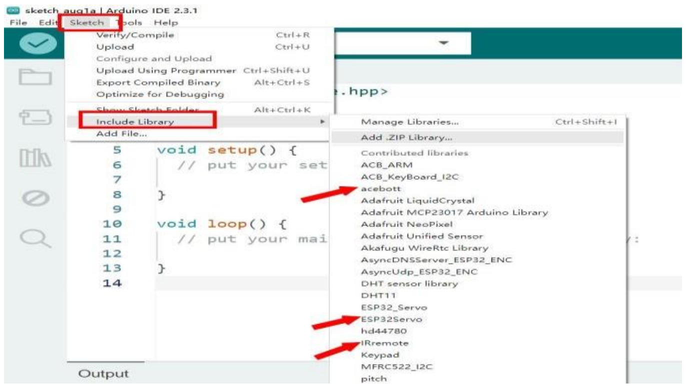
2. Open Arduino IDE, click "sketch" → Click "Include Library" → click "Add ZIP Library"



3. Select each folder individually, then click "Open" button. (Steps 1+2, 3+4, 5+6)

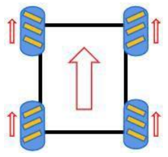
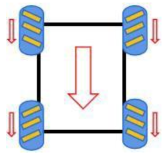
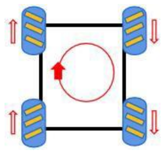
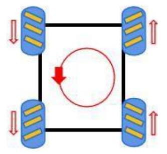
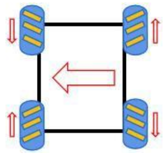
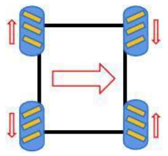
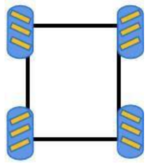


4. Check to make sure library files have added successfully.



```
void setup() {
  // put your set

}

void loop() {
  // put your mai

}
```

"OK, let's see what commands we have for motion control of the smart car! You just need to give different parameters, for different smart car motion states." See table below. The speed of the smart car can be selected from 0 to 255, with 255 as the fastest speed.

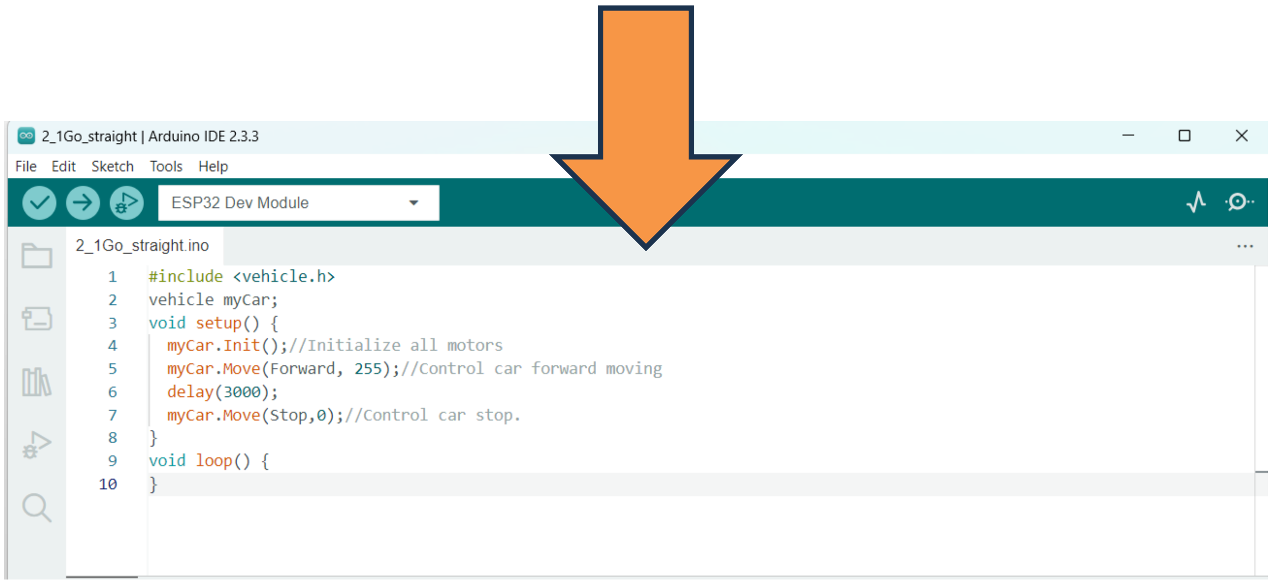
Motion states	Action diagram	Control command
Forward		myCar.Move(Forward,speed)
Backward		myCar.Move(Backward,speed)
Clockwise		myCar.Move(Clockwise,speed)
Contrarotate		myCar.Move(Contrarotate,speed)
Move_Left		myCar.Move(Move_Left,speed)
Move_Right		myCar.Move(Move_Right,speed)
Stop		myCar.Move(Stop,speed)

Motion Test #1: Make Rover Go Forward

Lumi continued, "From the control commands, we can see that if the smart car is going to go straight, the first parameter of the Move function needs to be 'forward' so that all four wheels of the smart car are in the same direction. Let's code the program in Arduino IDE and upload it to the rover to see how it works."

Copy and paste the code to move the rover forward into Arduino IDE sketch area.

```
#include <vehicle.h>
vehicle myCar;
void setup() {
  myCar.Init();//Initialize all motors
  myCar.Move(Forward, 255);//Control car forward moving
  delay(3000);
  myCar.Move(Stop,0);//Control car stop.
}
void loop() {
}
```


2_1Go_straight | Arduino IDE 2.3.3

File Edit Sketch Tools Help

ESP32 Dev Module

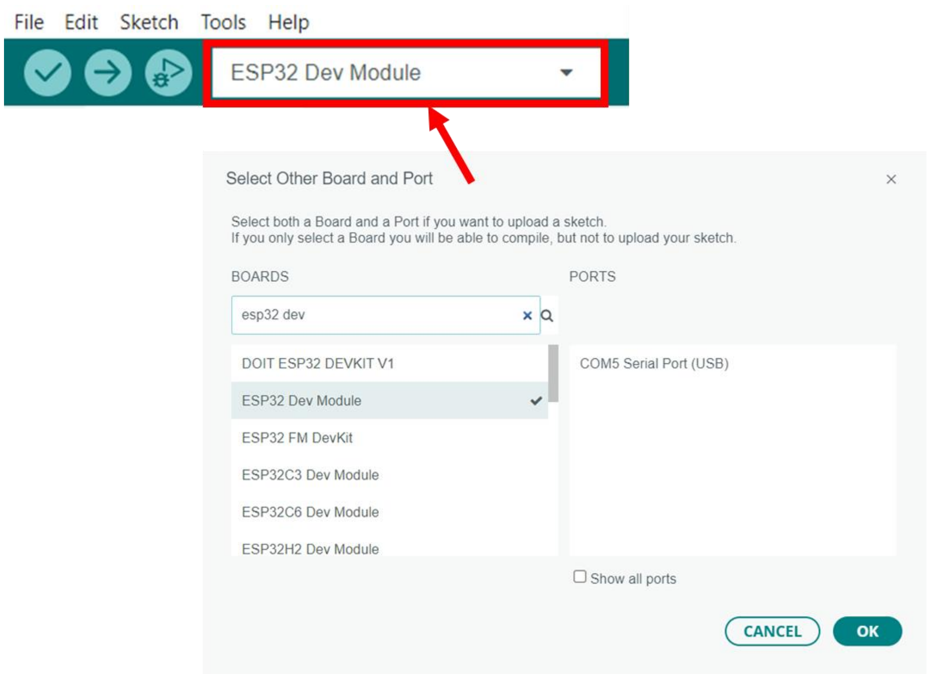
2_1Go_straight.ino

```
#include <vehicle.h>
vehicle myCar;
void setup() {
  myCar.Init();//Initialize all motors
  myCar.Move(Forward, 255);//Control car forward moving
  delay(3000);
  myCar.Move(Stop,0);//Control car stop.
}
void loop() {
}
```

NOTE: Keep the TX and RX pin connections to the camera module disconnected, otherwise the flash upload will fail. We will connect these pins in a later challenge.

Make sure the ESP32 rover board is connected to your computer with the USB cable and that the "ESP32 Dev Module" is selected in Arduino IDE with the correct Port.

See screenshot below:



Select Other Board and Port

Select both a Board and a Port if you want to upload a sketch.
If you only select a Board you will be able to compile, but not to upload your sketch.

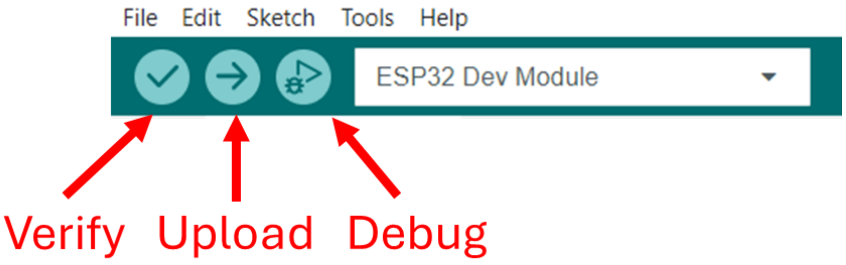
BOARDS	PORTS
esp32 dev	
DOIT ESP32 DEVKIT V1	COM5 Serial Port (USB)
ESP32 Dev Module ✓	
ESP32 FM DevKit	
ESP32C3 Dev Module	
ESP32C6 Dev Module	
ESP32H2 Dev Module	

☐ Show all ports

CANCEL OK

Use the "Verify" function to compile the code to see if there are any issues. Then use "Upload" to send the program code to the ESP32 control board in the rover. Debug can be used if errors are encountered.

Make sure the rover's battery pack switch is set to **OFF** while uploading the program.



Arduino IDE will show progress in the "Output" section and will take some time to compile code, connect to the board, and upload program.

Once complete, disconnect the USB cable and make sure the rover is on the floor before switching the battery pack power **ON** to execute the program code. In this case, the rover should move forward and stop.

I upload the code, and saw the smart car start to move forward a short distance before it stopped so I asked Lumi, "How do I change the distance the smart car travels?"

Lumi explains, "Distance is speed times time! When the speed is determined, you try changing the time parameter to see what happens." I modified the time parameters according to the tips given by Lumi, was able to adjust the distance to go forward.

Try changing parameters in Arduino IDE and upload the code to see the effects.

Motion Test #2: Make Rover Turn Right

Lumi continued, "From the control commands, we can see that if the smart car is going to go straight, the first parameter of the Move function Just like going straight, you can control the Angle of rotation by changing the rotation time.

```
#include <vehicle.h>
vehicle myCar;
void setup() {
  myCar.Init();//Initialize all motors
  myCar.Move(Clockwise , 255);//Control car counterclockwise rotate
  delay(750);//Modify the parameters in red
  myCar.Move(Stop,0);//Control car stop.
}
void loop() {
}
```

Just like before, (1) copy and paste the code into Arduino IDE, (2) connect the ESP32 board to the computer with the USB cable, (3) make sure battery pack switch is OFF, (4) and “Upload” the program.

When the cable is disconnected and battery pack switched back ON, does the rover turn right? Try adjusting the speed and time parameters to see what happens.

Motion Test #3: Make Rover Move Right

Ponderosa Planet has sharp obstacles to navigate, and thanks to the wheels on our rover, it can drive horizontally, or sideways, to the right. Test it out with this code.

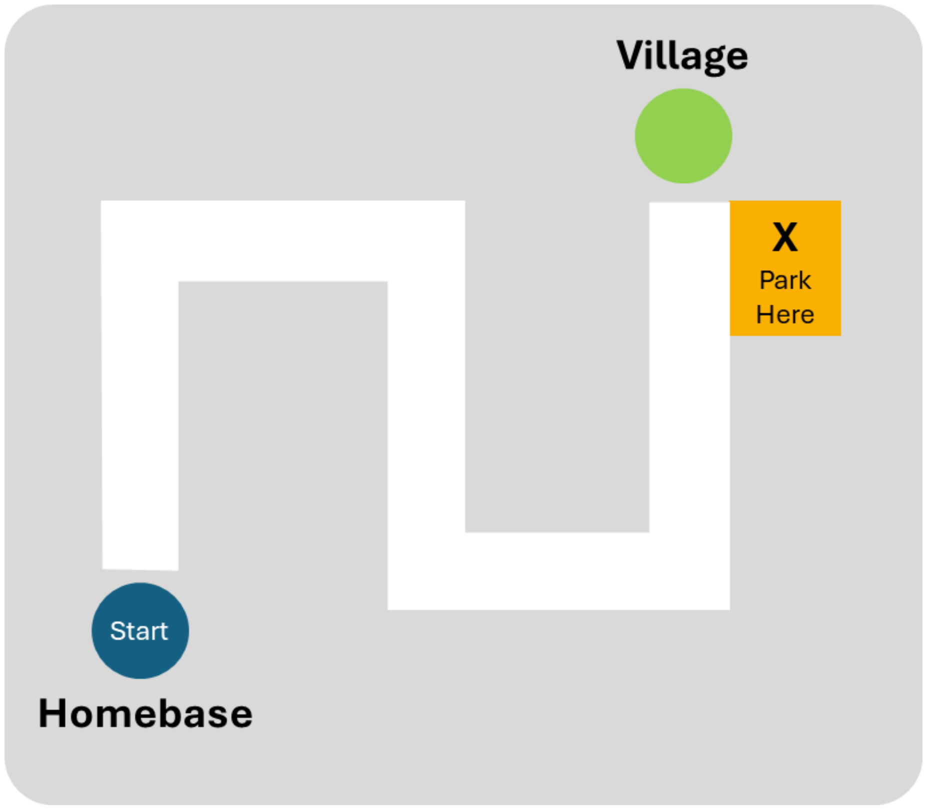
```
#include <vehicle.h>
vehicle myCar;
void setup() {
  myCar.Init();//Initialize all motors
  myCar.Move(Move_Right , 255);//Control car counterclockwise rotate
  delay(750);//Modify the parameters in red
  myCar.Move(Stop,0);//Control car stop.
}
void loop() {
}
```

Try out other commands to move the vehicle.

Challenge #4: Navigate to the Village

All of a sudden, an alarm was blaring all around me. Lumi said "It's time to put your new skills into action. We've got to get the rover programed to head to the Pondi Village as soon as possible. It sounds like there is an emergency and all explorer teams need to get there right away!"

Now that you know the basics to program the rover to move. Can you use the motion commands to make the rover follow a complete instruction set to repeatably move from one location to another?



To create the complete instruction set, code the directions and times like this:

```
#include <vehicle.h>
vehicle myCar;

void setup() {
  myCar.Init();//Initialize all motors
  myCar.Move(Forward, 255);//Control car forward moving
  delay(3000);
  myCar.Move(Clockwise , 255);//Control car counterclockwise rotate
  delay(750);//Modify the parameters in red
  myCar.Move(Forward, 255);//Control car forward moving
  delay(1500);
  myCar.Move(Clockwise , 255);//Control car counterclockwise rotate
  delay(750);//Modify the parameters in red
  myCar.Move(Forward, 255);//Control car forward moving
  delay(3000);
  myCar.Move(Contrarotate , 255);//Control car contrarotate
  delay(750);//Modify the parameters in red
  myCar.Move(Forward, 255);//Control car forward moving
  delay(1500);
  myCar.Move(Contrarotate , 255);//Control car contrarotate
  delay(750);//Modify the parameters in red
  myCar.Move(Forward, 255);//Control car forward moving
  delay(3000);
  myCar.Move(Move_Right , 255);//Control car left moving
  delay(750);//Modify the parameters in red
  myCar.Move(Stop,0);//Stop car
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

I uploaded my program and continued to debug and modify the parameters. Finally, the smart car completed the upgrade of the automatic driving function.

Next, before we boarded the smart car to the village, Lumi reminded me: "Remember to bring your computer, it will be your most important tool for the rest of the journey."

Challenge #5: Through the Dark Forest

By the time you finish building and mapping a path for the rover, it's already dark out. Turn on the rover's LED light so you can see your way through the path.

"All electronic components need power and energy to work properly, and LEDs are no exception," Lumi explains. "If you want to light up an LED, you need to send power into it."

LED Light Test #1: Turn on the LED Lights

```
#define leftLed 12
#define rightLed 2
void setup(){
  pinMode(leftLed, OUTPUT);
  pinMode(rightLed, OUTPUT);
}
void loop(){
  digitalWrite(leftLed,HIGH);//Turn on left led, high power
  digitalWrite(rightLed,HIGH);//Turn on right led, high power
}
```

At full power we can make the LED light turn on. What happens if I reduce the amount of power I give to the LEDs? If I reduce the amount of energy I give to the LED, its brightness should decrease. I wonder: "How to change the controller board output voltage?"

Lumi replied: "ESP32 controller board can use PWM technology to change the output voltage, the following is the corresponding program, upload the program to see if the brightness of the lights has changed."

LED Light Test #2: Change LED Brightness

```
#define leftLed 12
#define rightLed 2
void setup(){
  pinMode(leftLed, OUTPUT);
  pinMode(rightLed, OUTPUT);
  //Change red parameters, range 0-255 to display different brightness
  analogWrite(leftLed,50);
  analogWrite(rightLed,50);
}
void loop(){
}
```

Lumi continued, "You can change the parameters in the parentheses in the program, which ranges *from 0-255, corresponding to the output voltage range of 0-5V."*

Lumi looked at me constantly modifying and debugging the voltage parameters in the code to achieve different brightness of the lights, so he said: "Congratulations, you have mastered the method of controlling the brightness, next, can you change the brightness of the lights gradually from the darkest to the brightest, and then from the brightest back to the darkest?"

LED Light Test #3: Breathing Lights

```
#define leftLed 12
#define rightLed 2
const int fadeDelay = 10; // Latency per brightness change (ms)

void setup(){
 pinMode(leftLed, OUTPUT);
 pinMode(rightLed, OUTPUT);
}

void loop(){
 for (int brightness = 0; brightness <= 255; brightness++)
 {
  analogWrite(leftLed,brightness);
  analogWrite(rightLed,brightness);
  delay(fadeDelay);
 } //gradually Turn on the headlights
 for (int brightness = 255; brightness >= 0; brightness--)
 {
  analogWrite(leftLed,brightness);
  analogWrite(rightLed,brightness);
  delay(fadeDelay);
 }//Gradually extinguish the headlights
}
```

With the upload of the program, the headlights have the effect of breathing lights. Great job!

Challenge #6：Avoid Obstacles in Boulder Canyon

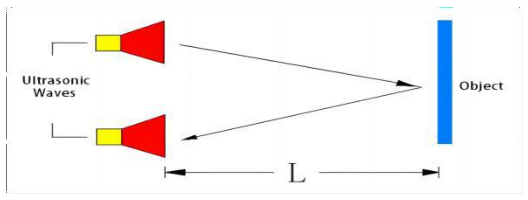
After using our new communication device to talk walk the Pondi natives in the village, we got a call from home base telling us to go back. When we started driving, we noticed the ground was littered with mini volcanoes sputtering hot lava. We needed a way to easily navigate back home.

"As long as we make the smart car have the ability to avoid obstacles automatically, we can safely pass through the volcanos," Lumi said.

"How can a smart car detect volcanoes in front of it?" I wondered.

Lumi replied: "We have installed ultrasonic sensors in our smart car, which is like the eyes of the smart car, so that the smart car can sensitively detect obstacles in front of it. There are two probes on the ultrasonic sensor: one acts as a transmitter to send the ultrasonic signal, and the other acts as a receiver to receive the reflected signal," Lumi explains.

Lumi continued: "The ultrasonic transmitter sends an ultrasonic wave in a certain direction, it travels through the air, it hits an obstacle on the way, and it returns immediately, and the ultrasonic receiver receives the reflected wave. By converting the calculation, the distance L between the ultrasonic wave and the obstacle in front is obtained."



I asked, "How do we get this distance L?"

"We're going to get this data through programming," Lumi said, and then showed the distance measurement task information.

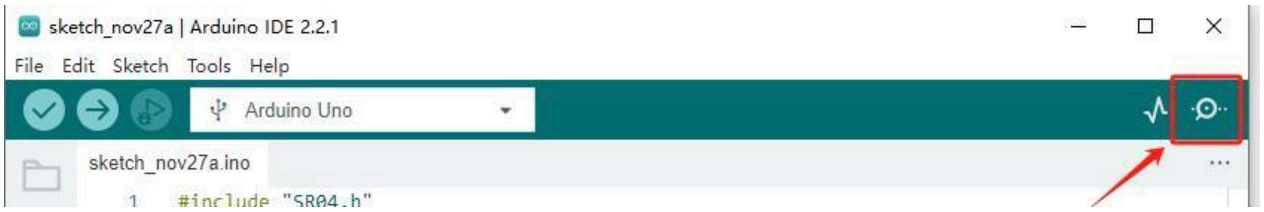
I am confused and ask: "serial port monitor?"

Lumi explained, "The serial monitor is the operating platform for serial data communication between the computer and the controller board. It can print and display the data transmitted from *the controller board to the computer, and it can also send data directly to the controller. It is a very important tool."*

"How do I turn on the serial monitor to see the distance data?" I asked.

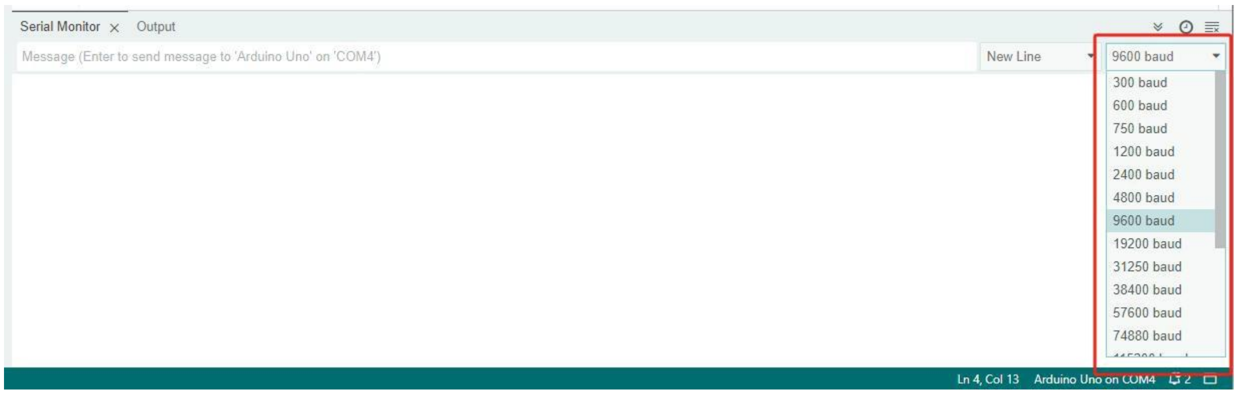
Test #1: Display Ultrasonic Distance

The port for the serial monitor and the USB port to upload Arduino code are the same, so click the magnifying glass icon in the top right corner of Arduino IDE to see the serial monitor data.


sketch_nov27a | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
sketch_nov27a.ino

```
#include "SR04.h"
```

After opening the serial port monitor, set the appropriate baud rate, baud rate is the speed of data communication. Click on the Baud rate drop-down menu to select the appropriate baud rate as needed, here we need 9600.

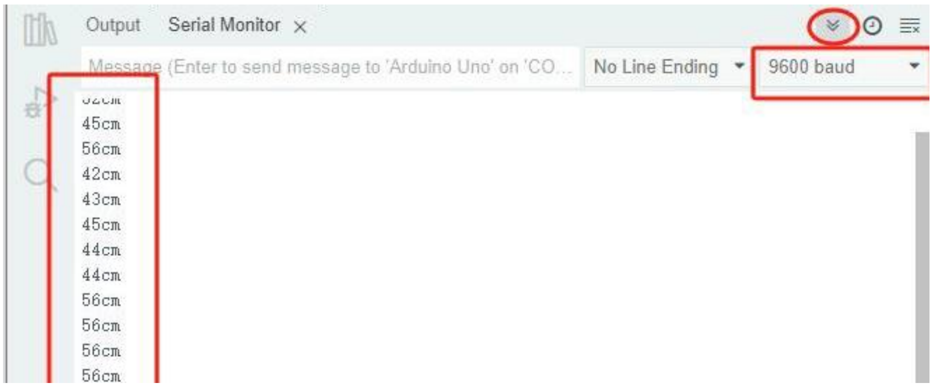


I followed the steps to turn on the serial monitor and said to Lumi, "I have turned on the serial monitor, how do I program to get the distance of the ultrasound?"

Lumi responded: "Very good, use the following code for ultrasonic detection to quickly let the car know the distance of objects ahead."

```
#include <ultrasonic.h>
ultrasonic myUltrasonic;
int UT_distance = 0;

void setup() {
  Serial.begin(9600);//The serial port monitor is initialized with baud rate of 9600
  myUltrasonic.Init();
}
void loop() {
  UT_distance = myUltrasonic.Ranging(Trig_PIN, Echo_PIN);
  //the distance of the ultrasonic detection
  Serial.print(UT_distance);
  // The serial port shows the distance of ultrasonic detection
  Serial.println("cm");
  delay(1000);
}
```

After I uploaded the program, I opened the serial port monitor, set the baud rate, and found that the serial port monitor continuously display the detected distance.



```
45cm
56cm
42cm
43cm
45cm
44cm
44cm
56cm
56cm
56cm
56cm
```

At the same time, Lumi's voice said, "Great, you have mastered the ultrasonic detecting function, now we can start implementing the obstacle avoidance function. Think about what the smart car needs to do if it encounters an obstacle. Lumi then presented information about side task two.

I immediately replied, "Just turn around in place."

Lumi said, "That's good. The logic is clear. Do you remember how to control the smart car *turning in place and the Angle of rotation?"*

I replied confidently, "Of course, you can control the rotation angle of the car by changing the rotation time."

Lumi responded, "Great, so let's make it obstacle-avoiding."

Test #2: Obstacle Avoidance

Use the following code to make the rover turnaround, 180 degrees, to avoid obstacles.

```
#include <vehicle.h>
#include <ultrasonic.h>
vehicle myCar;
ultrasonic myUltrasonic;
int UT_distance = 0;

void setup(){
 myCar.Init();
 myUltrasonic.Init();
}

void loop()
{
 UT_distance = myUltrasonic.Ranging(Trig_PIN, Echo_PIN);
 if (UT_distance <= 25){
 //The distance is less than 25cm to achieve the effect of turning
  myCar.Move(Contrarotate, 180);
  delay(1500); //The turning time is modified to realize the rotation of different angles
  myCar.Move(Stop, 0);
 }
 else {
 //If the distance is greater than 25, move forward
 myCar.Move(Forward, 150);
 }
}
```

I uploaded the program, modified the turning time of the smart car and constantly debugged the program, so that the smart car obtained the function of obstacle avoidance.

III: Rescue Mission

"Headquarters - What's wrong? We got your call and came back as fast as we could."

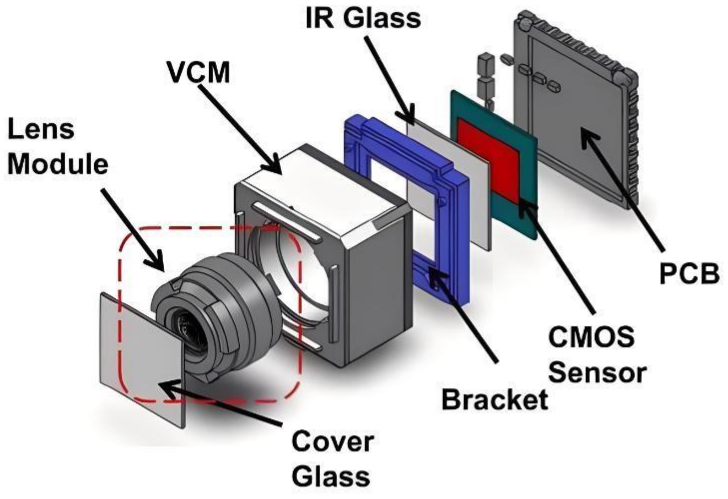
"A team was out collecting dirt samples and they came across the entrance to a cave system. It sounds like Pondi natives are trapped and it's too dangerous to send humans into the caves. Who knows what could be living down there."

"What should we do?"

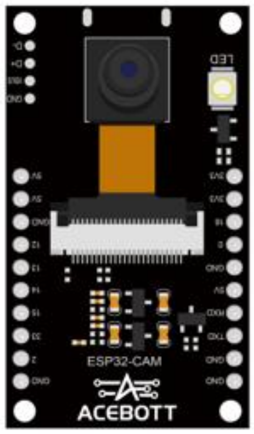
Lumi calmly replied, "Don't worry, we can use the rover camera and control it remotely."

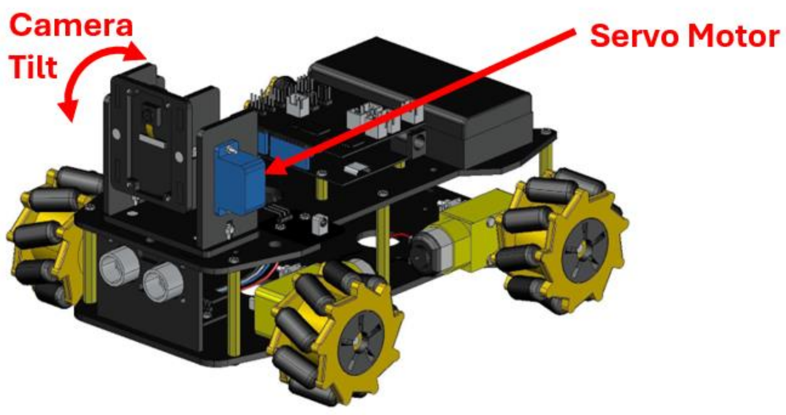
Challenge #7： Enable WiFi Camera Function

The structure of a camera mainly includes a lens, voice coil motor (VCM), filter, image sensor, and Printed Circuit Board (PCB) circuit board. Electrical signals are captured and converted to be able to see images on a screen.



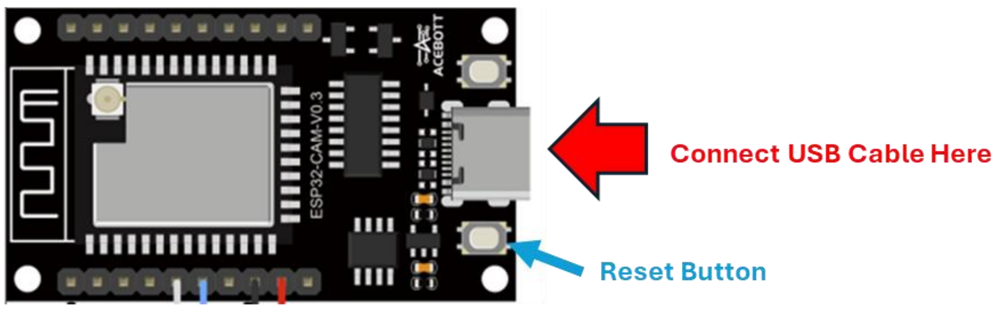
The ACEBOTT Camera module installed in Challenge #1 was assembled with a servo motor to tilt the camera up and down for greater viewing angles.





There is a lot of programing code that needs to be uploaded in order to enable the camera and the WiFi capabilities.

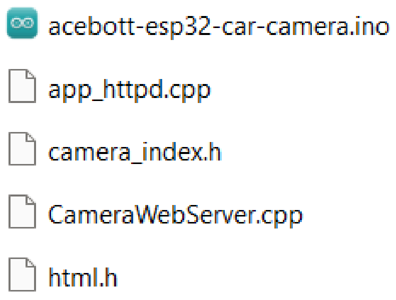
1. Switch the battery pack to **OFF**. Let's first load all the programing for the car to body. Download the following program folder from Ponderosa Hacks Resources: acebott-esp32-car-body program. Double-click on the "acebott-esp32-car-camera.ino" file and it should open up in Arduino IDE. Connect the ESP32 board with the USB cable and upload the program.
2. Next, let's get the camera setup. Connect the USB cable from your computer to the board on the camera module. You can now connect the 4 pins to the ESP32 board (left off in Challenge #1).



3. Download the following program folder from Ponderosa Hacks Resources and save to your computer:

acebott-esp32-car-camera

In the folder, you should see the following files.



acebott-esp32-car-camera.ino

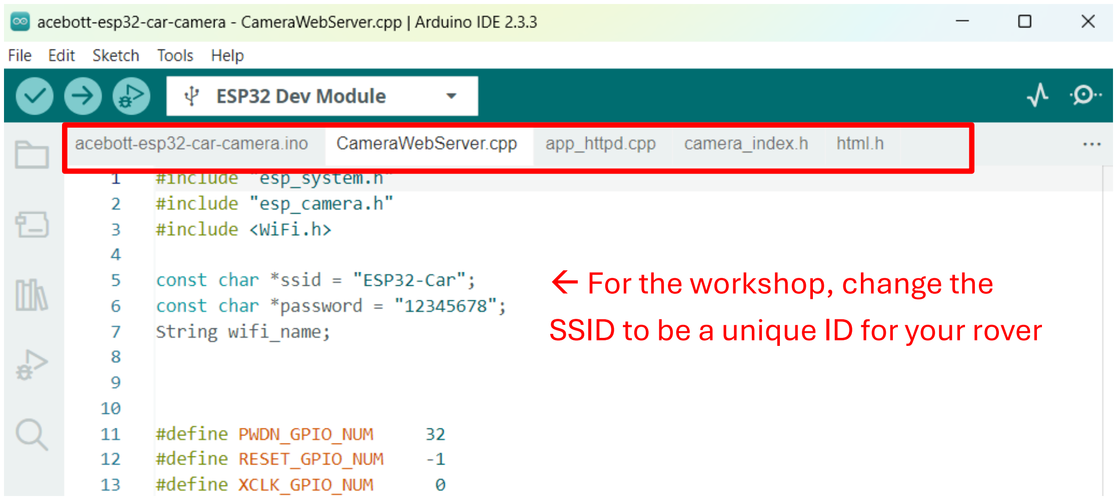
app_httpd.cpp

camera_index.h

CameraWebServer.cpp

html.h

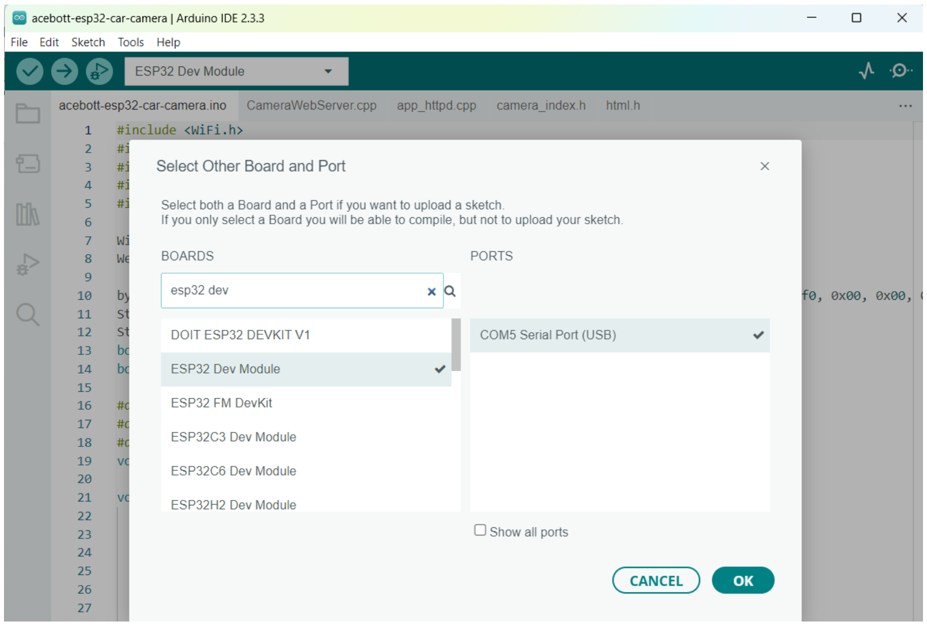
4. Double-click on the “acebott-esp32-car-camera.ino” file and it should open up in Arduino IDE along with the other files in the folder. Note the WiFi info in the CameraWebServer.cpp file.


acebott-esp32-car-camera - CameraWebServer.cpp | Arduino IDE 2.3.3
File Edit Sketch Tools Help
ESP32 Dev Module
acebott-esp32-car-camera.ino
CameraWebServer.cpp
app_httpd.cpp
camera_index.h
html.h

```
#include "esp_system.h"
#include "esp_camera.h"
#include <WiFi.h>

const char *ssid = "ESP32-Car";
const char *password = "12345678";
String wifi_name;



#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
```

← For the workshop, change the SSID to be a unique ID for your rover

5. Make sure to select the “ESP32 Dev Module” and correct COM port for your computer.


acebott-esp32-car-camera | Arduino IDE 2.3.3
File Edit Sketch Tools Help
ESP32 Dev Module
acebott-esp32-car-camera.ino
CameraWebServer.cpp
app_httpd.cpp
camera_index.h
html.h

```
#include <WiFi.h>
```

Select Other Board and Port

Select both a Board and a Port if you want to upload a sketch.
If you only select a Board you will be able to compile, but not to upload your sketch.

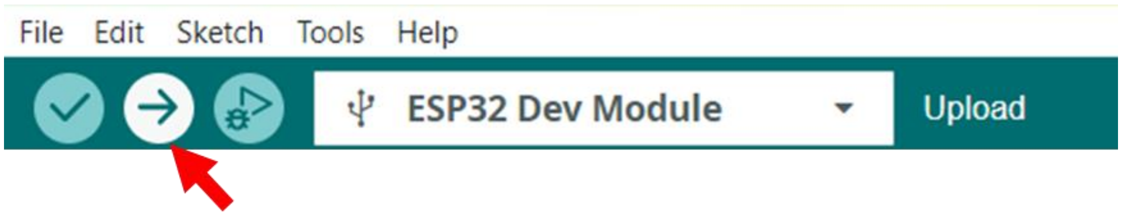
BOARDS: esp32 dev
- DOIT ESP32 DEVKIT V1
- ESP32 Dev Module ✓
- ESP32 FM DevKit
- ESP32C3 Dev Module
- ESP32C6 Dev Module
- ESP32H2 Dev Module

PORTS:
- COM5 Serial Port (USB) ✓

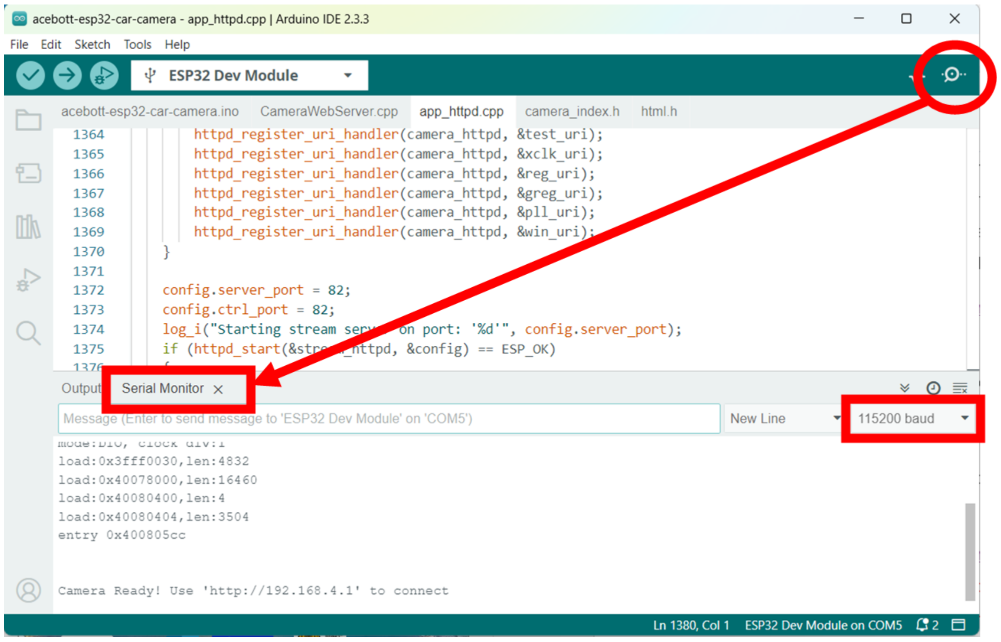
☐ Show all ports

CANCEL OK

6. Press the upload button to send program to the camera module.



7. Once output shows 100% complete. Open the Serial Monitor by clicking the magnifying glass in top right corner. Also, set baud to 115200.

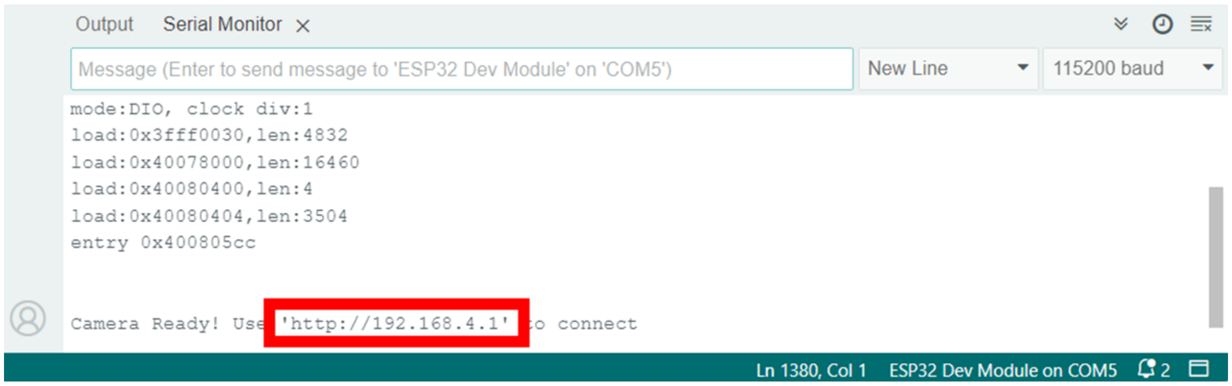

acebott-esp32-car-camera - app_httpd.cpp | Arduino IDE 2.3.3
File Edit Sketch Tools Help
ESP32 Dev Module
acebott-esp32-car-camera.ino
CameraWebServer.cpp
app_httpd.cpp
camera_index.h
html.h
Output
Serial Monitor
New Line
115200 baud
Ln 1380, Col 1 ESP32 Dev Module on COM5

```
        httpd_register_uri_handler(camera_httpd, &test_uri);
        httpd_register_uri_handler(camera_httpd, &xclk_uri);
        httpd_register_uri_handler(camera_httpd, &reg_uri);
        httpd_register_uri_handler(camera_httpd, &greg_uri);
        httpd_register_uri_handler(camera_httpd, &pll_uri);
        httpd_register_uri_handler(camera_httpd, &win_uri);
    }

    config.server_port = 82;
    config.ctrl_port = 82;
    log_i("Starting stream server on port: '%d'", config.server_port);
    if (httpd_start(&stream_httpd, &config) == ESP_OK)
```

Message (Enter to send message to 'ESP32 Dev Module' on 'COM5')

```
mode:DIO, clock div:1
load:0x3fff0030,len:4832
load:0x40078000,len:16460
load:0x40080400,len:4
load:0x40080404,len:3504
entry 0x400805cc


Camera Ready! Use 'http://192.168.4.1' to connect
```

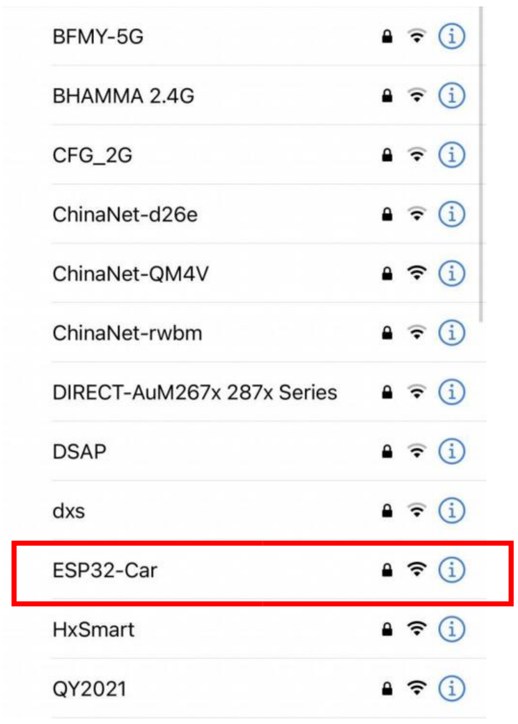
8. Then, click the white “reset” button on the camera module, next to the USB cable connection (see image in step #1). After clicking the button, data should appear in the Arduino IDE Serial Monitor. Note the http address listed to connect to the camera via a web page.



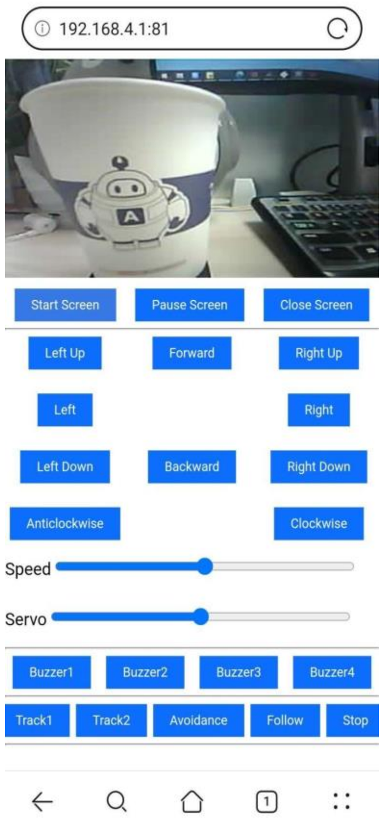
Message (Enter to send message to 'ESP32 Dev Module' on 'COM5')

```
mode:DIO, clock div:1
load:0x3fff0030,len:4832
load:0x40078000,len:16460
load:0x40080400,len:4
load:0x40080404,len:3504
entry 0x400805cc


Camera Ready! Use 'http://192.168.4.1' to connect
```

9. On your computer, go to your WiFi settings and find the SSID for your rover. Connect with the password (12345678 by default) and go to the indicated web address in a web browser. Your WiFi should be connected, and will show that there is no internet (this is the correct setup).



10. To see the camera view. make sure the battery pack on the rover is ON and use the “Start Screen” button to view camera. May need to adjust your browser window for the camera viewer. This is a great time to put the rover on the floor and test out the remote functions.

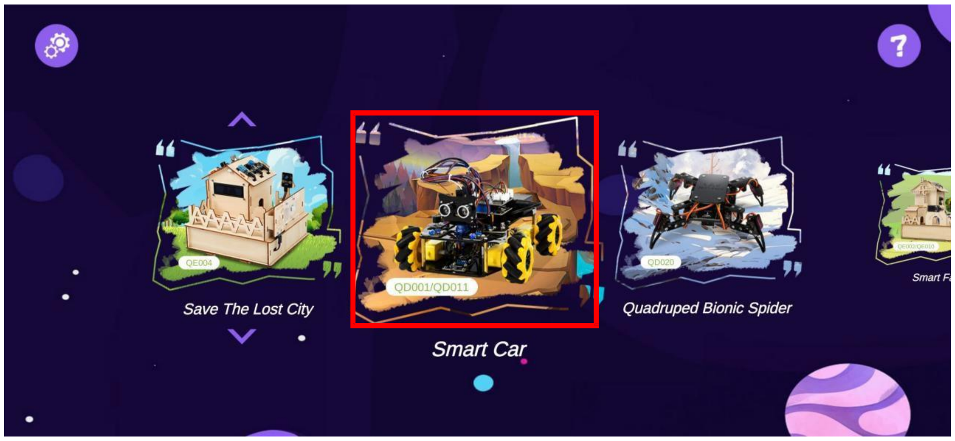


Challenge #8: ACEBOTT App Remote Control

You can also use the ACEBOTT App to remote control the rover with your mobile phone.

Step 1: Go to the Google Play or Apple store to download the ACEBOTT App on your phone.

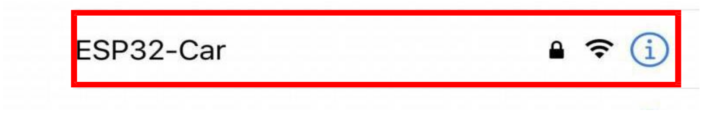
Step 2: After opening the app, you will enter the splash screen. Enter the selection screen and choose the Smar tCar option as shown below.



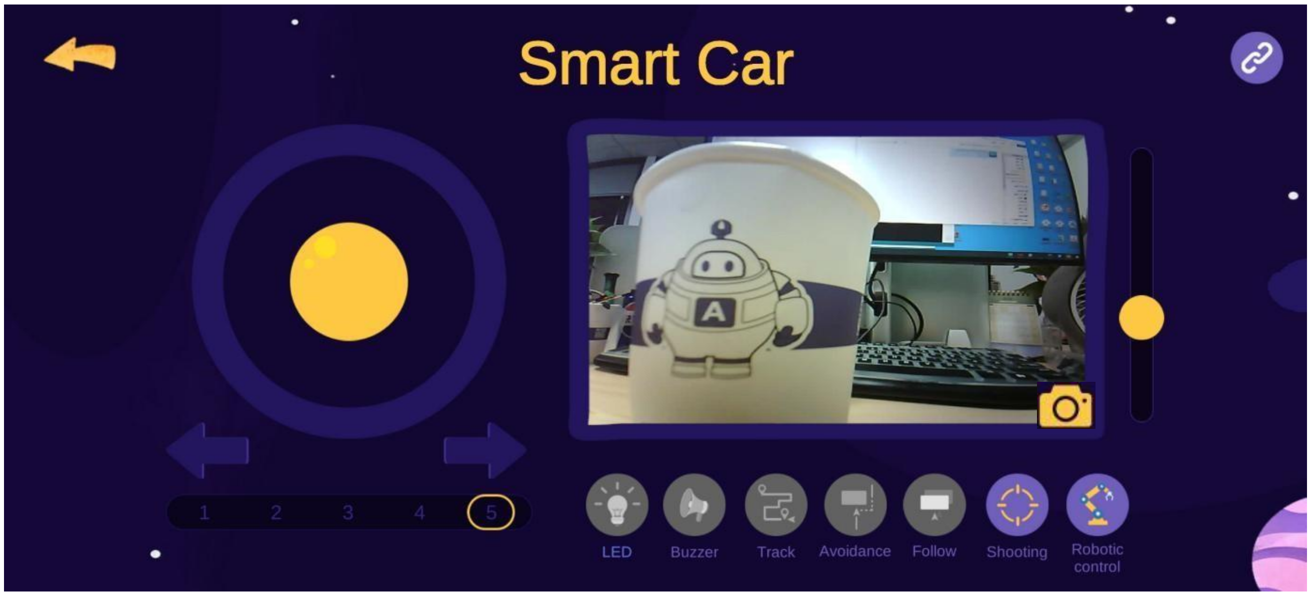
"ACEBOTT app is installed, now can I directly control the smart car?"

Lumi shook his head and said, "Not yet. The APP to control the smart car also requires the smart car to send WIFI, and the phone to connect to the WIFI of the smart car"

Step 3: Use your mobile phone to wirelessly connect to the Rover's WIFI. Connect to your Rover's SSID name, for example "ESP32-Car" wifi, with default password 12345678, as shown below.



After connecting the WiFi, click the connection icon in the upper right corner of the APP to complete the connection.



The functions of the app include: using joysticks and arrows to control the movement of the smart car, with numbers 1 to 5 below the arrows indicating the speed of the smart car's movement. On the right side, there is a live video feed, and clicking the camera icon in the bottom right corner of the video feed saves the current image displayed by the camera to the phone's photo album. Below the video feed are the corresponding hardware module functions, and on the right side of the video feed, there is a vertical slider to control the pitch angle of the camera.

Final Challenge: Rescue Pondi Villagers

Oh no! There are native Pondi trapped in the case. Use the rover to rescue as many Pondi as possible.

Good luck with the rescue mission!!