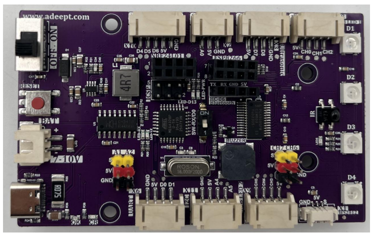
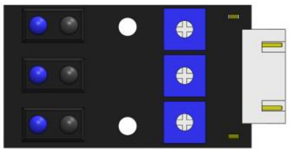
Lesson 11 Introduction to the Line-Tracking Module

11.1 Overview

This course focuses on the application of the line tracking module, aiming to guide learners to use Adeept Robot Control Board and Arduino development environment to master the working principle, hardware connection method, code writing and debugging skills of the line tracking module, and achieve project development based on line tracking technology.

11.2 Required Components

Components	Quantity	Picture
Adeept Robot Control Board	1	
Type-C USB Cable	1	
Line Tracking module	1	

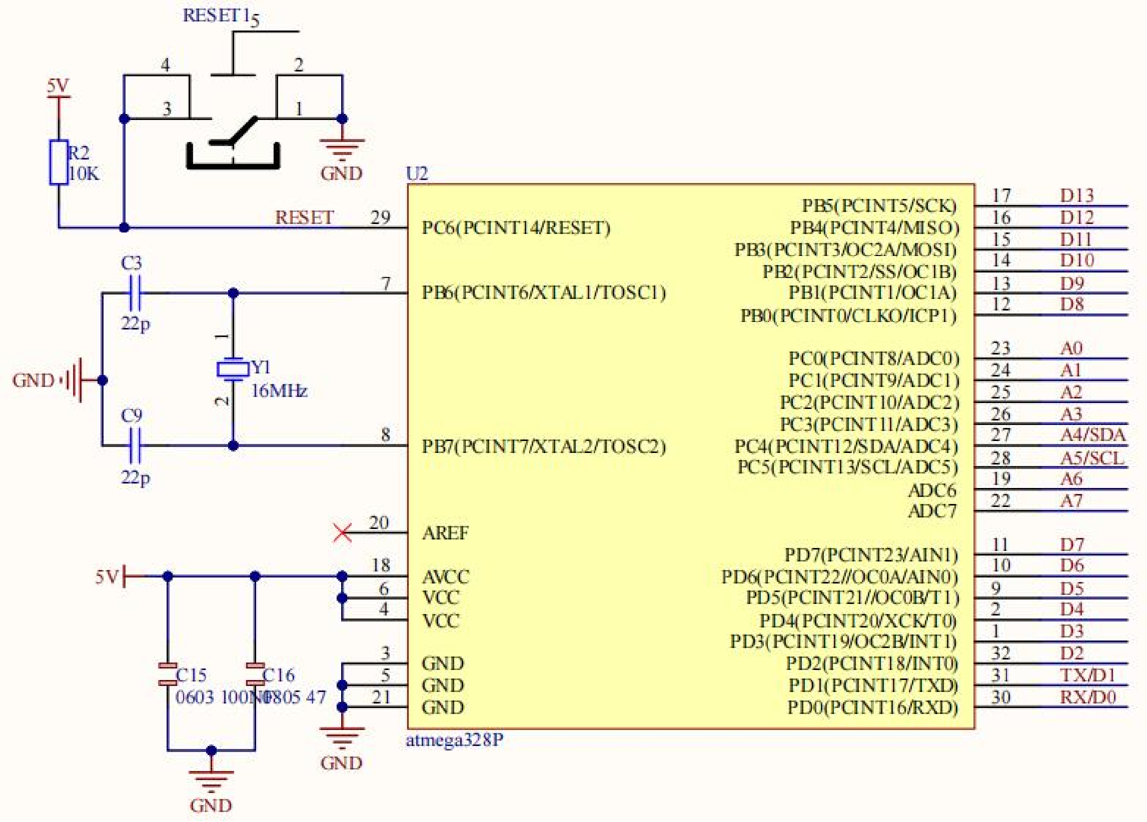
11.3 Principle Introduction

"Line Tracking"--- On the white paper with black lines drawn, the black lines can be judged according to the intensity of the reflected light received due to the different reflection coefficients of the black lines and the white paper to light.

A more common detection method, infrared detection method, is used in the line tracking module. Infrared detection method is to use the characteristics of infrared rays that have different reflection properties on physical surfaces of different colors. During the running of the program, infrared light is continuously emitted to the ground. When the infrared light encounters the white ground, diffuse emission occurs, and the reflected light is received by the receiving tube; if it encounters a black line, the infrared light is absorbed, and the receiver of the line tracking module Signal cannot be received. The line tracking module we provide is a three-channel infrared tracking module, which contains 3 sets of sensors, each of which consists of an infrared emitting LED and an infrared receiver.

PINS of Adeept Robot Control Board	Arduino(X10)
S3	D4
S2	D5
S1	D6
VCC	5V
GND	GND

11.4 Wiring Diagram



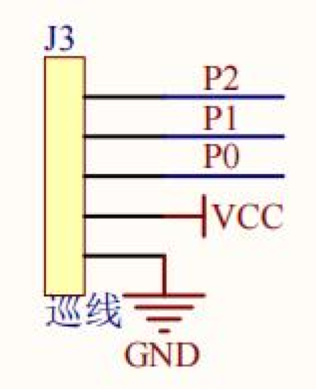
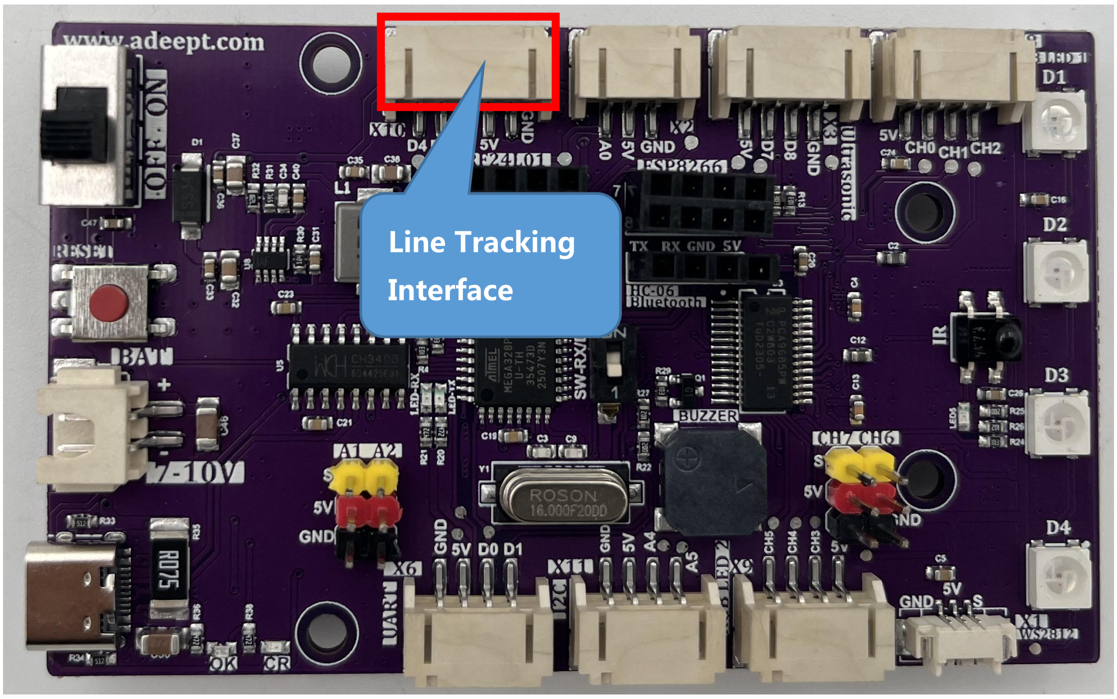
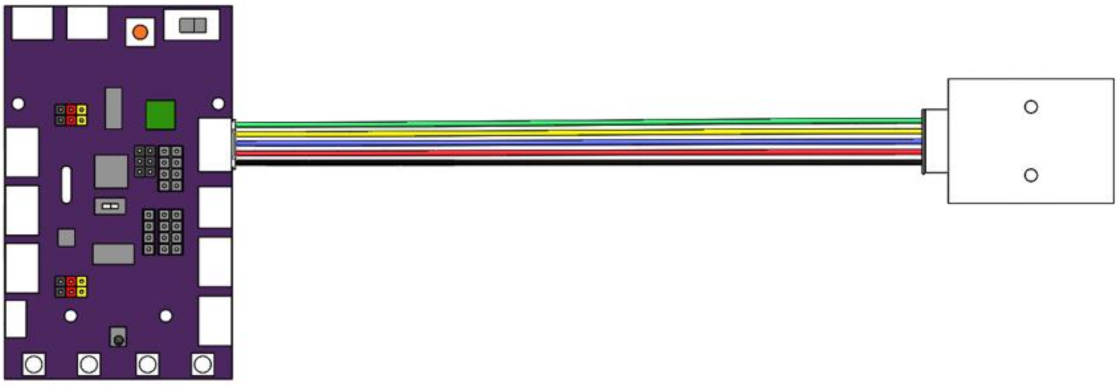


Figure as below :

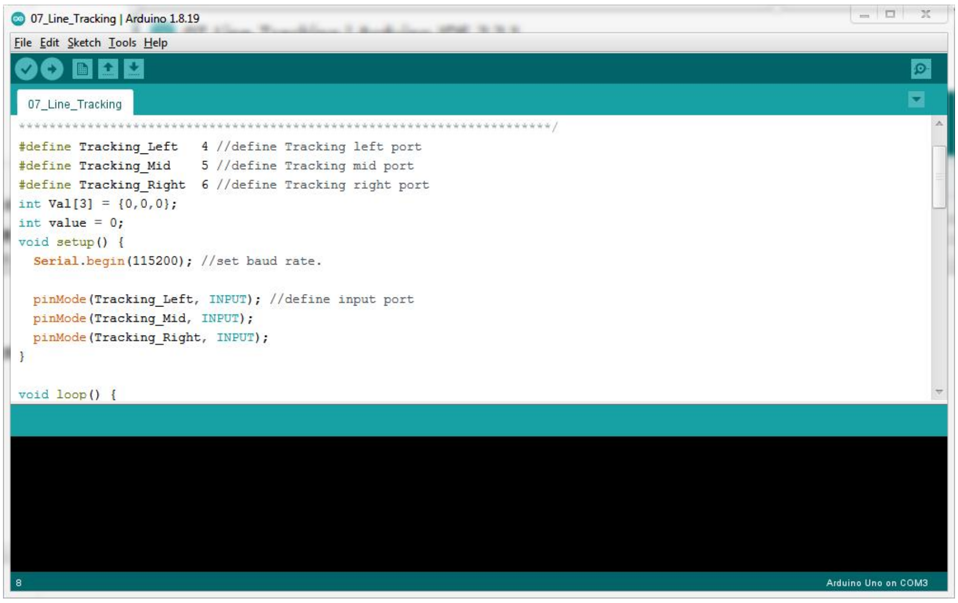




11.5 Demonstration

1. Connect your computer and Adeept Robot Control Board (Arduino Board) with a USB cable.

2. Open “**07_Line_Tracking**” folder in “/Code”, double-click “07_Line_Tracking.ino”.


07_Line_Tracking | Arduino 1.8.19
File Edit Sketch Tools Help
07_Line_Tracking
Arduino Uno on COM3

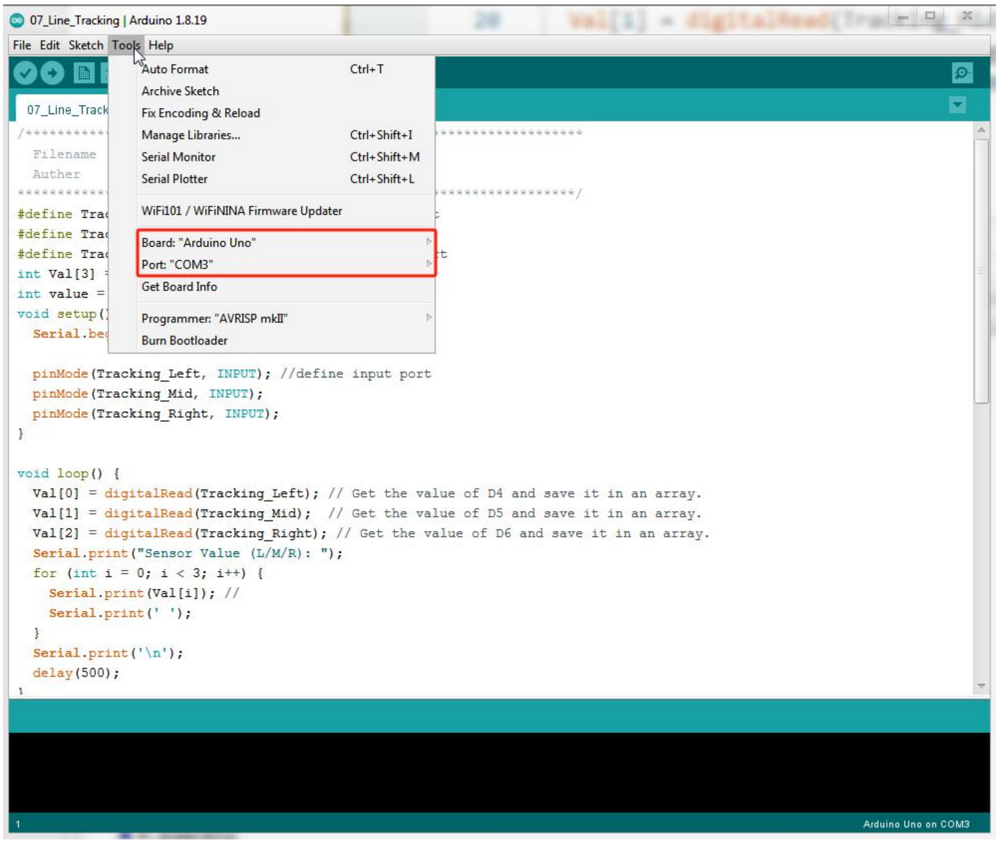
```
***************************************************************************/
#define Tracking_Left   4 //define Tracking left port
#define Tracking_Mid    5 //define Tracking mid port
#define Tracking_Right  6 //define Tracking right port
int Val[3] = {0,0,0};
int value = 0;
void setup() {
  Serial.begin(115200); //set baud rate.

  pinMode(Tracking_Left, INPUT); //define input port
  pinMode(Tracking_Mid, INPUT);
  pinMode(Tracking_Right, INPUT);
}

void loop() {
```

3. Select development board and serial port.

Board: Tools--->Board--->Arduino AVR Boards--->Arduino Uno

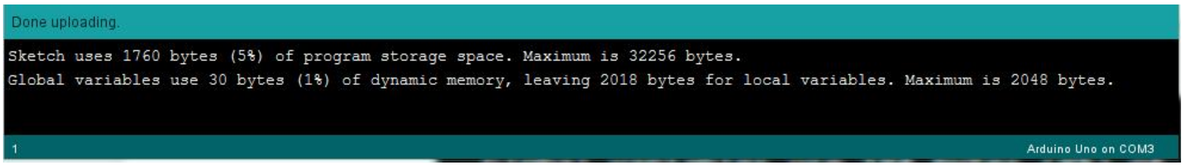
Port: Tools --->Port--->COMx

Note: The port number will be different in different computers.


07_Line_Tracking | Arduino 1.8.19
File Edit Sketch Tools Help
07_Line_Track
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Port: "COM3"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Arduino Uno on COM3

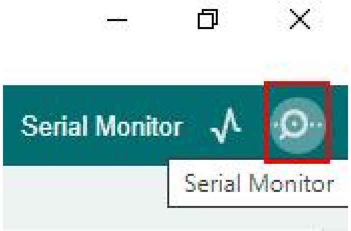
```
/***********                                     *******************
  Filename
  Auther
***********                                     ******************/
#define Trac
#define Trac
#define Trac                                    t
int Val[3] =
int value =
void setup()
  Serial.be

  pinMode(Tracking_Left, INPUT); //define input port
  pinMode(Tracking_Mid, INPUT);
  pinMode(Tracking_Right, INPUT);
}

void loop() {
  Val[0] = digitalRead(Tracking_Left); // Get the value of D4 and save it in an array.
  Val[1] = digitalRead(Tracking_Mid);  // Get the value of D5 and save it in an array.
  Val[2] = digitalRead(Tracking_Right); // Get the value of D6 and save it in an array.
  Serial.print("Sensor Value (L/M/R): ");
  for (int i = 0; i < 3; i++) {
    Serial.print(Val[i]); //
    Serial.print(' ');
  }
  Serial.print('\n');
  delay(500);
}
```

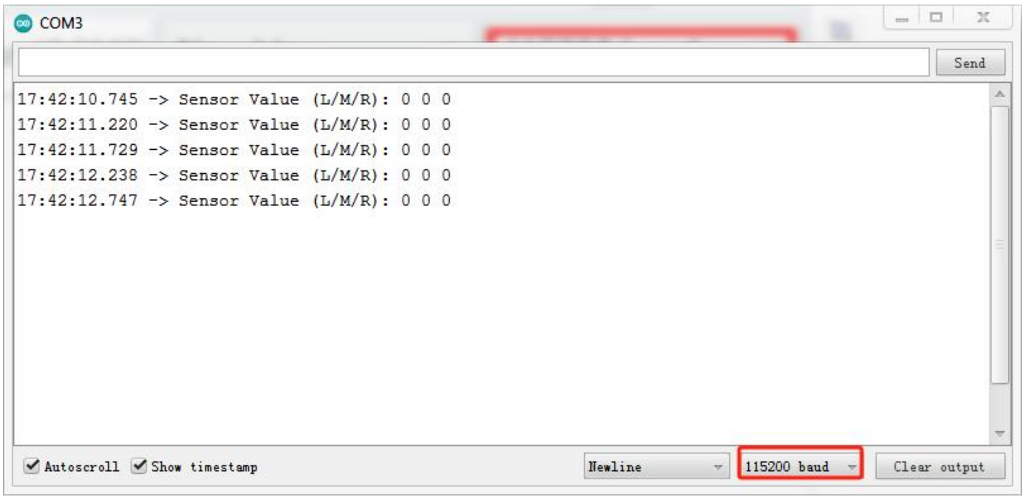
4. After opening, click → to upload the code program to the Arduino. If there is no error warning in the console below, it means that the Upload is successful.



```
Done uploading.
Sketch uses 1760 bytes (5%) of program storage space. Maximum is 32256 bytes.
Global variables use 30 bytes (1%) of dynamic memory, leaving 2018 bytes for local variables. Maximum is 2048 bytes.
1                                                                                        Arduino Uno on COM3
```

5. Click Serial Monitor, Set the baud rate as 115200.





```
17:42:10.745 -> Sensor Value (L/M/R): 0 0 0
17:42:11.220 -> Sensor Value (L/M/R): 0 0 0
17:42:11.729 -> Sensor Value (L/M/R): 0 0 0
17:42:12.238 -> Sensor Value (L/M/R): 0 0 0
17:42:12.747 -> Sensor Value (L/M/R): 0 0 0
```

☑ Autoscroll ☑ Show timestamp Newline 115200 baud

6. You will see the data obtained by the three detection terminals. (L/M/R) means (Left / Mid / Right).

11.6 Code

Complete code refer to 07_Line_Tracking.ino

```
/**********************************************************************
  Filename    : 07_Line_Tracking.ino
  Auther      : www.adeept.com
**********************************************************************/
#define Tracking_Left   4 //define Tracking left port
#define Tracking_Mid    5 //define Tracking mid port
#define Tracking_Right  6 //define Tracking right port
int Val[3] = {0,0,0};
int value = 0;
void setup() {
  Serial.begin(115200); //set baud rate.

  pinMode(Tracking_Left, INPUT); //define input port
  pinMode(Tracking_Mid, INPUT);
  pinMode(Tracking_Right, INPUT);
}

void loop() {
  Val[0] = digitalRead(Tracking_Left); // Get the value of D4 and save it in an array.
  Val[1] = digitalRead(Tracking_Mid);  // Get the value of D5 and save it in an array.
  Val[2] = digitalRead(Tracking_Right); // Get the value of D6 and save it in an array.
  Serial.print("Sensor Value (L/M/R): ");
  for (int i = 0; i < 3; i++) {
    Serial.print(Val[i]); //
    Serial.print(' ');
  }
  Serial.print('\n');
  delay(500);
}


void Track_Read(void){
  Val[0] = digitalRead(Tracking_Left); //Read the value of digital interface 4 and assign it to val[0]
  Val[1] = digitalRead(Tracking_Mid);
  Val[2] = digitalRead(Tracking_Right);
  if ((Val[0]==0)&&(Val[1]==0)&&(Val[2]==0)){ // 000
    value = 0;
    }
  else if ((Val[0]==0)&&(Val[1]==1)&&(Val[2]==0)){ // 010
    value = 1;
    }
  else if ((Val[0]==1)&&(Val[1]==0)&&(Val[2]==0)){ // 100
    value = 2;
    }
  else if ((Val[0]==1)&&(Val[1]==1)&&(Val[2]==0)){ // 110
    value = 3;
    }
  else if ((Val[0]==0)&&(Val[1]==0)&&(Val[2]==1)){ // 001
    value = 4;
    }
  else if ((Val[0]==0)&&(Val[1]==1)&&(Val[2]==1)){ // 011
    value = 5;
    }
  else if ((Val[0]==1)&&(Val[1]==1)&&(Val[2]==1)){ // 111
    value = 6;
    }
  return value;
}
```

Code explanation

Initialization Stage:

In the setup function, first use Serial. start (115200); Initialize serial communication and set the baud rate to 115200 to view sensor data in the serial monitor.

Then use the pinMode function to set the left, middle, and right sensor pins of the line tracking module to input mode, allowing Arduino to read the sensor signals.

Loop Control Process:

Stage 1: Read the status values of three sensors using the digitalRead function and store them in the array Val.

Stage 2: Print the sensor values to the serial monitor through serial communication, with each line starting with "Sensor Value (L/M/R):" followed by the values of the left, middle, and right sensors, separated by spaces and wrapped at the end.

Stage 3: Finally, The program has a delay of 500ms to avoid frequent data output and facilitate observation and analysis of data.