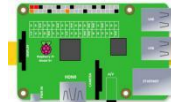




Lesson 14 Light Tracking

14.1 Overview

In this lesson, you'll learn to implement light - tracking functionality with a Raspberry Pi, Adeept Robot HAT V3.2, and a Light Tracking module. It covers components, principle, wiring, program demonstration, and code explanation.

14.2 Required Components

| Components | Quantity | Picture |
|-----------------------|----------|---|
| Raspberry Pi | 1 |  |
| Adeept Robot HAT V3.2 | 1 |  |
| Light Tracking Module | 1 |  |

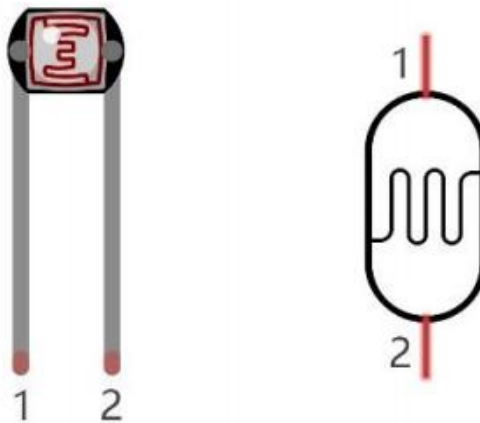
14.3 Principle Introduction

Light Tracking Module is composed of two photoresistors. The photoresistor is very sensitive to the amount of light present. We can use this feature to make a light tracing car. The car is controlled to turn toward the light source by reading the ADC values of the two photoresistors at the head of the car.

ADC is an electronic integrated circuit used to convert analog signals such as voltages to digital or binary form consisting of 1s and 0s. The range of our ADC on Raspberry Pi is 8 bits, that means the resolution is $2^8=256$, and it represents a range (at 5V) will be divided equally to 256

parts. The range of analog values corresponds to ADC values. So the more bits the ADC has, the denser the partition of analog will be and the greater the precision of the resulting conversion.

A photoresistor is simply a light sensitive resistor. It is an active component that decreases resistance with respect to receiving luminosity (light) on the component's light sensitive surface. A photoresistor's resistance value will change in proportion to the ambient light detected. With this characteristic, we can use a photoresistor to detect light intensity. The photoresistor and its electronic symbol are as follows.



When a photoresistor's resistance value changes due to a change in light intensity, the voltage between the photoresistor and resistor R1 will also change. Therefore, the intensity of the light can be obtained by measuring this voltage.

When the brightness of the light received by the photoresistor R1 and R2 is the same, the voltage at SIG is 5/2, which is 2.5V. Therefore, when the brightness of the light received by the two photoresistors is different, the voltage at SIG will be greater or less than 2.5V, and the range is 0-5V. The ADC value range corresponding to the SIG voltage range is 0-256.

Due to the problem of resistance accuracy, when the module is under the same light intensity, the detected value may deviate, but it does not affect the basic functions of the module.

| PINS of Raspberry Pi | Line Tracking Module |
|----------------------|----------------------|
| A1 | SIG |
| VCC | VCC |
| GND | GND |

14.4 Wiring Diagram

Adeept Robot HAT V3.1 expands 8 ADC pins from I2C pins through the ADS7830 chip. The ADC A1 pin is used for the Light Tracking interface, the A0 pin is used to detect battery power, and the other 6 pins are on the board, as shown below:

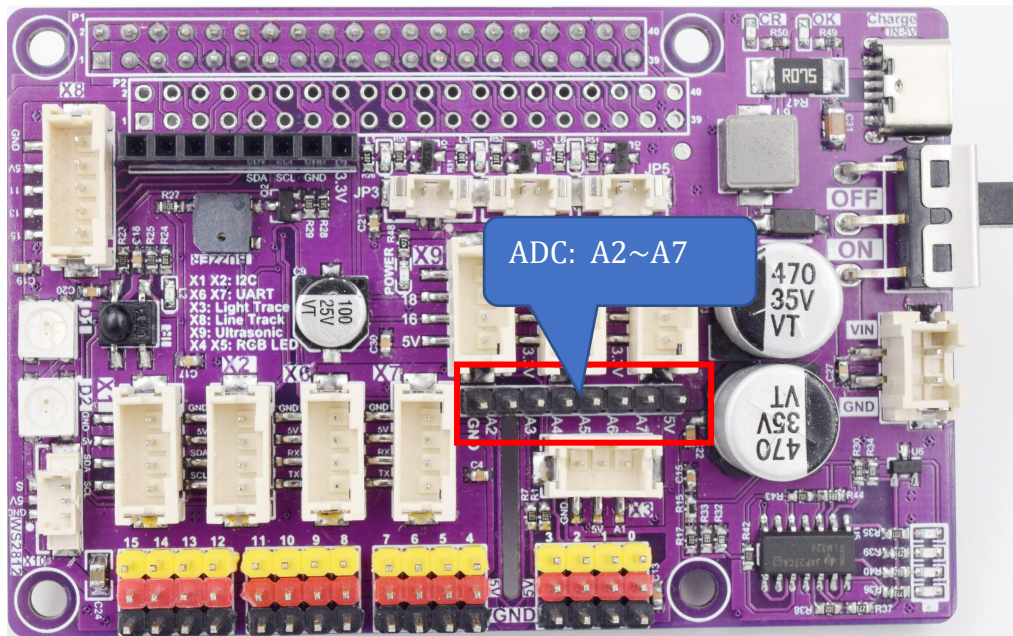
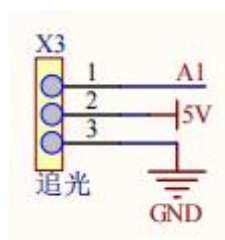
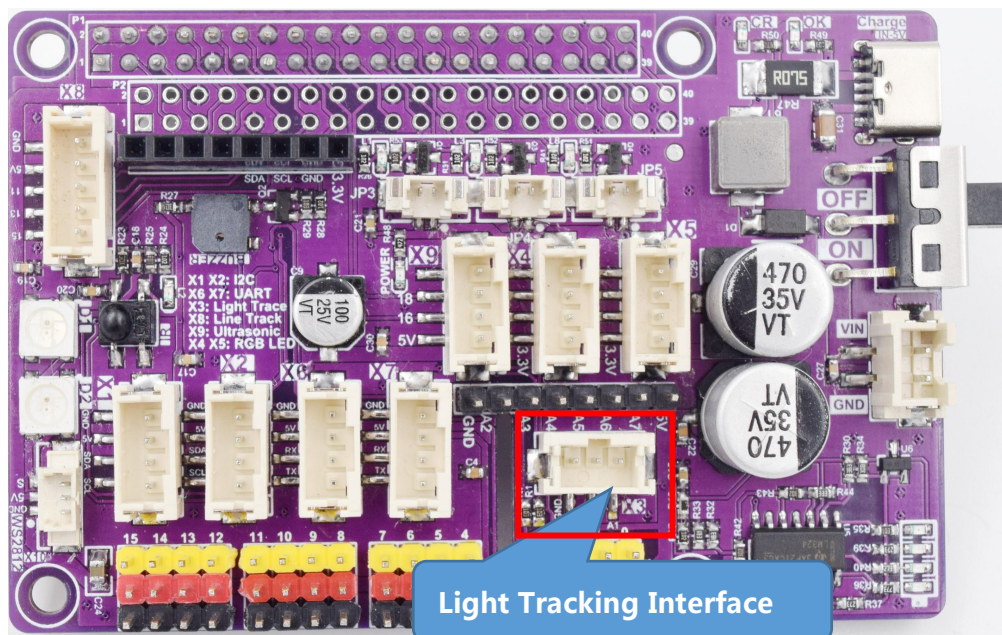


Figure as below :





14.5 Demonstration

1. **Remotely log:** Remotely log in to the Raspberry Pi terminal.
2. **Navigate to the Program Folder:** Enter the following command in the terminal and press Enter to access the folder where the program is located:

```
cd Adept_4WD_Smart_Car_for_RPi/Examples/08_Light_Tracking/
```

```
pi@raspberrypi:~ $ cd Adept_4WD_Smart_Car_for_RPi/Examples/08_Light_Tracking/
pi@raspberrypi:~/Adept_4WD_Smart_Car_for_RPi/Examples/08_Light_Tracking $
```

3. **View Directory Contents:** Type "ls" in the terminal and press Enter. This will display all the files in the current directory, ensuring that the "**LineTracking.py**" file is present:

```
ls
```

```
pi@raspberrypi:~/Adept_4WD_Smart_Car_for_RPi/Examples/08_Light_Tracking $ ls
LightTracking.py
```

4. **Run the Program:** Enter the command and press **Enter** to run the program:

```
sudo python3 LightTracking.py
```

```
pi@raspberrypi:~/Adeept_4WD_Smart_Car_for_RPi/Examples/08_Light_Tracking $ sudo python3 LightTracking.py
Light Tracking Value: 124
Light Tracking Value: 123
Light Tracking Value: 132
Light Tracking Value: 114
Light Tracking Value: 211
Light Tracking Value: 227
Light Tracking Value: 229
Light Tracking Value: 59
Light Tracking Value: 28
Light Tracking Value: 23
```

5. Observation and Termination:After running the program successfully, Use light to shine on "LDR1" or "LDR2" at both ends of the light tracking module, and the value on the screen will show the maximum value (close to 255), or the value will decrease, with the minimum value being 0 (close to 0).When you want to terminate the running program, you can press the "**Ctrl + C**" shortcut key on the keyboard.

14.6 Code

Complete code refer to [LightTracking.py](#)

```
01 #!/usr/bin/env/python
02 # File name   : LightTracking.py
03 # Website    : www.Adeept.com
04 # Author     : Adeept
05 # Date      : 2025/03/7
06
07 import time
08 import smbus
09
10
11 class ADS7830(object):
12     def __init__(self):
13         self.cmd = 0x84
14         self.bus=smbus.SMBus(1)
15         self.address = 0x48 # 0x48 is the default i2c address for ADS7830 Module.
16
17     def analogRead(self, chn): # ADS7830 has 8 ADC input pins, chn:0,1,2,3,4,5,6,7
18         value = self.bus.read_byte_data(self.address, self.cmd|(((chn<<2 | chn>>1)&0x07)<<4))
19         return value
20
21 if __name__ == "__main__":
22     adc = ADS7830()
23     while True:
24         adc_value = adc.analogRead(1)
25         print(f"Light Tracking Value: {adc_value}")
26         time.sleep(0.5)
```

Code explanation

Initialization Stage:

Connect the ADS7830 module (address 0x48) via the I2C protocol. Define 8 ADC channels.

Loop Control Process:

Stage 1: Read the analog value of the photosensitive resistor connection pin through chan1.value and convert it to an ADC value.

Stage 2: Print them to the terminal

Stage 3: Delay the program by 500ms to avoid frequent data output and facilitate observation of data changes.