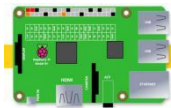
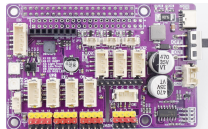



## Lesson 13 How to use the Light Tracking Module

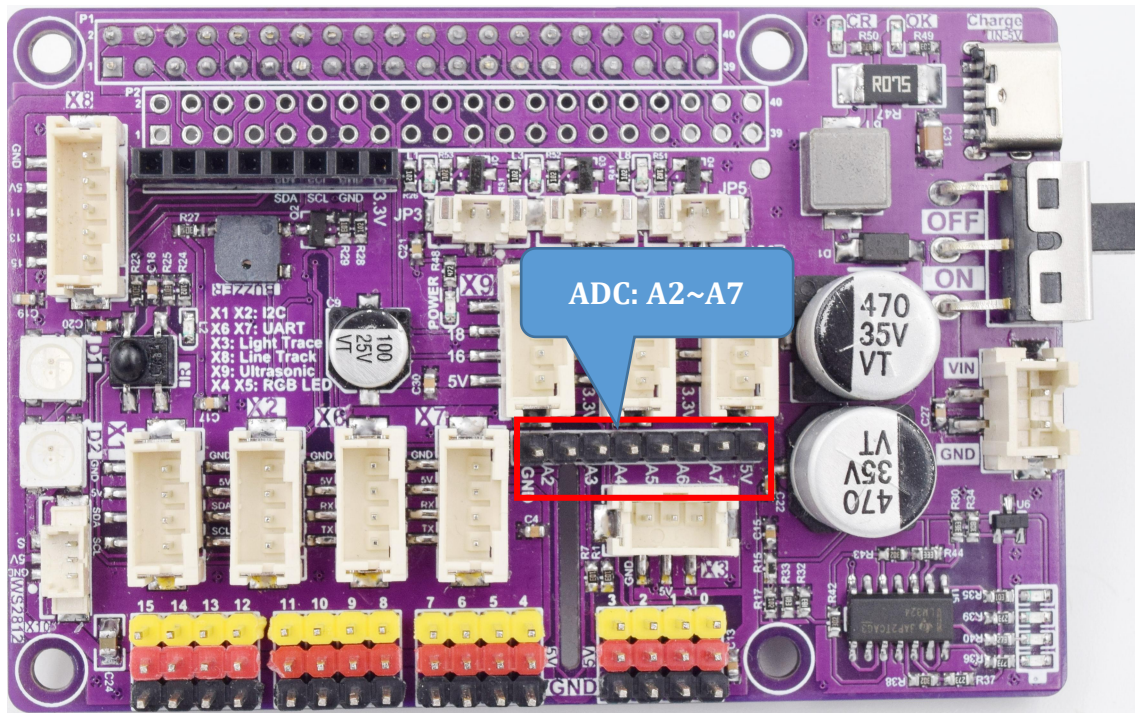
In this lesson, we will learn how to use the Light Tracking Module.

### 13.1 Components used in this Course

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

### 13.2 The Introduction of the Light Tracking Module

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:



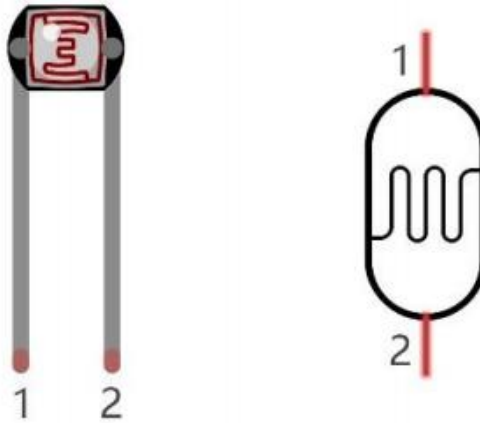
The address occupied by the ADS7830 chip on Adeept Robot HAT V3.1 is **0x48**.

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 lighttracing 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.

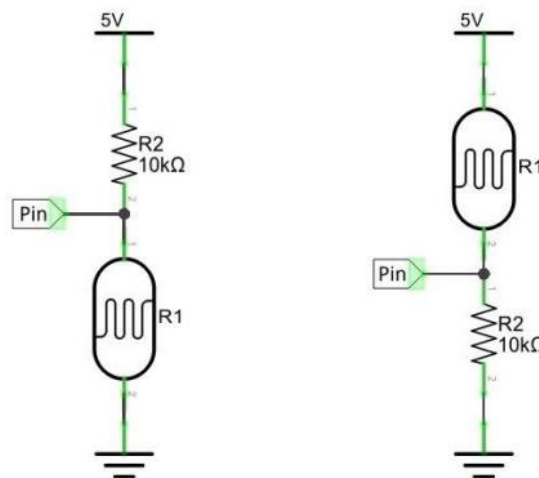
ADC is an electronic integrated circuit used to convert analog signals such as voltages to digital or binary form consisting of 1s and 0s. The range of our ADC on Raspberry Pi is 8 bits, that means the resolution is  $2^8=256$ , and it represents a range (at 5V) will be divided equally to 256 parts.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



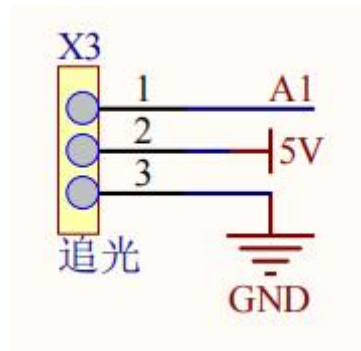
The circuit below is used to detect the change of a photoresistor's resistance value:



In the above circuit, 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.

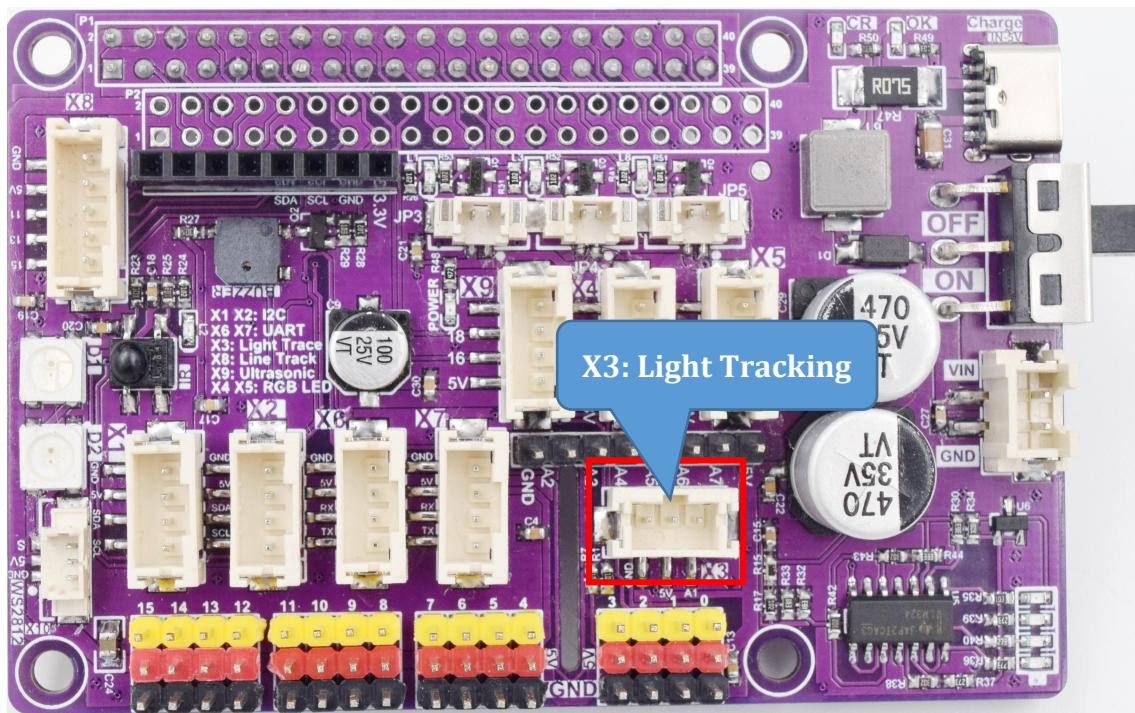
It can be seen from the circuit below that when the brightness of the light received by the photoresistor R1 and R2 is the same, the voltage is  $5/2$ , which is 2.5V. Therefore, when the brightness of the light received by the two photoresistors is different, the voltage will be

greater or less than 2.5V, and the range is 0-5V. The ADC value range corresponding to the voltage range is 0-255. (In fact, due to deviation, the maximum and minimum values may be deviated.)



### 13.3 Wiring Diagram

Figure as below:



### 13.4 How to use the Light Tracking Module



## Run the code

1. Remotely log in to the Raspberry Pi terminal.

```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Aug 29 08:17:49 2020 from 192.168.3.208

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $
```

2. See the occupied I2C address.

**i2cdetect -y 1**

```
pi@raspberrypi:~ $ i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- 48 -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- -- -- 5f --
60:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- -- -- -- -- -- -- --
```

3. Enter the command and press Enter to enter the folder where the program is located:

**cd adeept\_picar-b2/examples/**

```
pi@raspberrypi:~ $
pi@raspberrypi:~ $ cd adeept_picar-b2/examples/
pi@raspberrypi:~/adeept_picar-b2/examples $
```

4. View the contents of the current directory file:

**ls**

```

pi@raspberrypi:~/adeept_picar-b2/examples $ ls
01_LED.py      03_servo.py  05_RGB.py      07_ultra.py      09_lightTracking.py
02_buzzer.py  04_motor.py  06_ws2812.py  08_lineTracking.py
pi@raspberrypi:~/adeept_picar-b2/examples $

```

5. Enter the command and press Enter to run the program:

**sudo python3 09\_lightTracking.py**

```

pi@raspberrypi:~/adeept_picar-b2/examples $ sudo python3 09_lightTracking.py
Light Tracking Value: 115
Light Tracking Value: 116
Light Tracking Value: 115
Light Tracking Value: 115
Light Tracking Value: 115
Light Tracking Value: 115
Light Tracking Value: 115
Light Tracking Value: 115
Light Tracking Value: 115
Light Tracking Value: 115
Light Tracking Value: 115

```

6. 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).

7. When you want to terminate the running program, you can press the shortcut key "**Ctrl + C**" on the keyboard.

## 13.5 Main Code

Complete code refer to [09\\_lightTracking.py](#).

```

01 import time
02 import smbus
03
04
05 class ADS7830(object):
06     def __init__(self):
07         self.cmd = 0x84
08         self.bus=smbus.SMBus(1)
09         self.address = 0x48 # 0x48 is the default i2c address for ADS7830 Module.
10
11     def analogRead(self, chn): # ADS7830 has 8 ADC input pins, chn:0,1,2,3,4,5,6,7
12         value = self.bus.read_byte_data(self.address, self.cmd|(((chn<<2 |
13 chn>>1)&0x07)<<4))

```

```
14         return value
15
16 if __name__ == "__main__":
17     adc = ADS7830()
18     while True:
19         adc_value = adc.analogRead(1)
20         print(f"Light Tracking Value: {adc_value}")
        time.sleep(0.5)
```