**Project 25：****Human Induction Lamp**

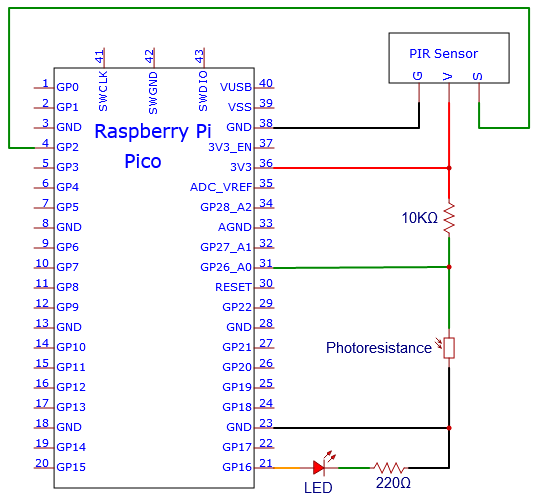
1. **Introduction**

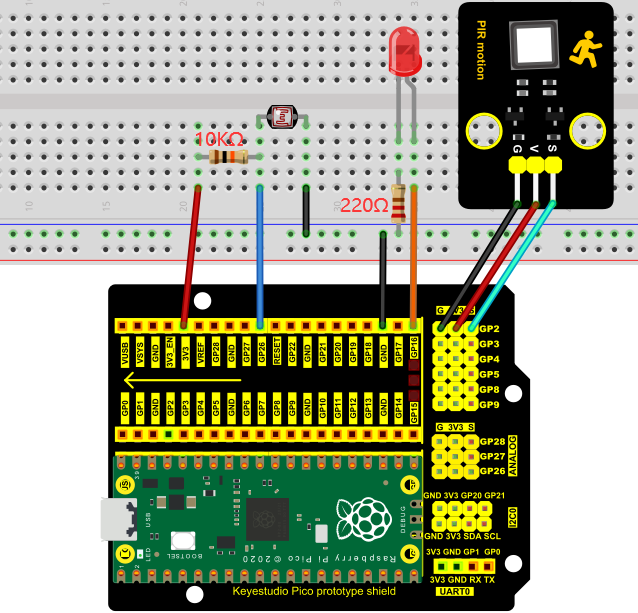
With the development of science and technology, the use of human induction lamp that usually used in the dark corridor area is very common in our real life, such as the corridor of the community, the bedroom of the room, the garage of the dungeon, the bathroom and so on. The human induction lamp are generally composed of a PIR Motion Sensor, a lamp, a photoresistor sensor and so on. In this project, we will learn how to use a PIR Motion Sensor, LEDs, and a photoresistor to make a human induction lamp .

1. **Components Required**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 17a6d7f241a04d4e932cb06b758197c4 | _DSC2552 | | 37 | 红 (4) | 10k欧 |
| Raspberry Pi Pico\*1 | Raspberry Pi Pico Expansion Board\*1 | | Photoresistor\*1 | Red LED\*1 | 10KΩResistor\*1 |
|  | 1 | 220欧 | 杜邦线-1 |  |  |
| Breadboard\*1 | PIR Motion Sensor\*1 | 220ΩResistor\*1 | F-F Dupont Wires | Jumper Wires | USB Cable\*1 |

1. **Circuit Diagram and Wiring Diagram**

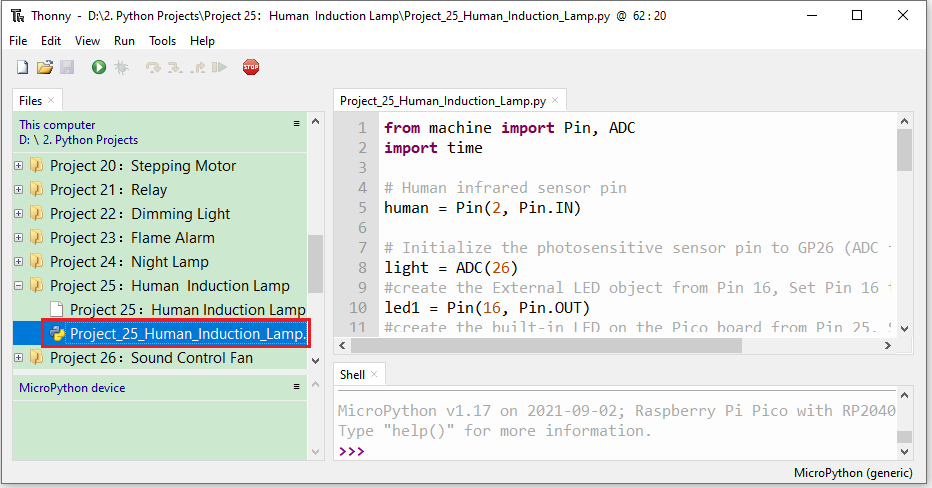




1. **Test Code**

The code used in this project is saved in the file KS3020 Keyestudio Raspberry Pi Pico Learning Kit Ultimate Edition\2. Windows System\1. Python\_Tutorial\2. Python Projects\Project 25：Human Induction Lamp. You can move the code to anywhere, for example, we can save the code in theDisk(D), the route is D:\2. Python Projects.

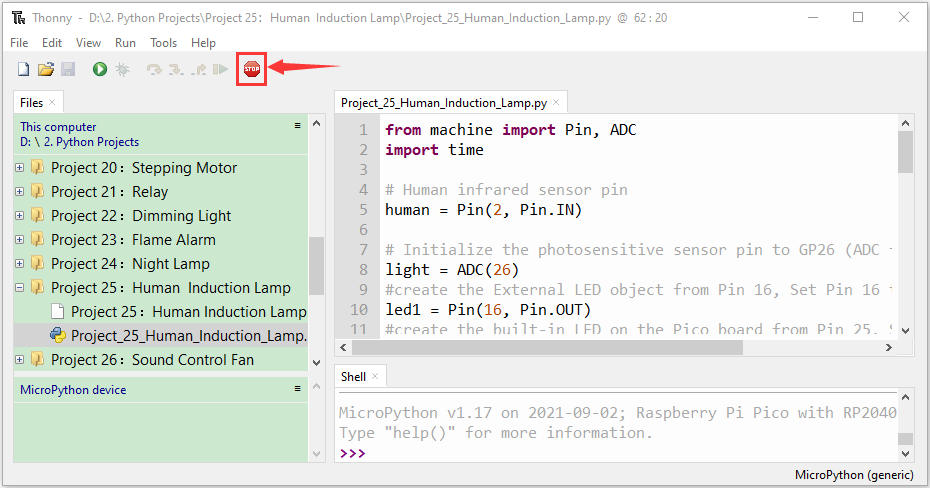
Open“Thonny”, click“This computer”→“D:”→“2. Python Projects”→“Project 25：Human Induction Lamp”. And double left-click the“Project\_25\_Human\_ Induction\_Lamp.py”.



|  |
| --- |
| from machine import Pin, ADC  import time    # Human infrared sensor pin  human = Pin(2, Pin.IN)    # Initialize the photosensitive sensor pin to GP26 (ADC function)  light = ADC(26)  #create the External LED object from Pin 16, Set Pin 16 to output  led1 = Pin(16, Pin.OUT)  #create the built-in LED on the Pico board from Pin 25, Set Pin 25 to output  led2 = Pin(25, Pin.OUT)    # Turn off the External LED  def led1\_off():  led1.value(0)    # Turn on the External LED  def led1\_on():  led1.value(1)    # Open the built-in LED on the Pico board  def led2\_on():  led2.value(1)    # Close the built-in LED on the Pico board  def led2\_off():  led2.value(0)    # Read the current analog value of the photosensitive sensor, range [0, 1023]  # The stronger the light intensity, the smaller the value.  def get\_value():  return int(light.read\_u16() \* 1024 / 65536)      def detect\_someone():  if human.value() == 1:  return True  return False    abc = 0    while True:  val = get\_value()  # print('val=', val)    if val >= 500:  led2\_on()  if detect\_someone() == True:  abc += 1  led1\_on()  print("value=", abc)  time.sleep(1)  else:  if abc != 0:  abc = 0  led1\_off()  else:  led2\_off()  led1\_off()    time.sleep(0.1) |

1. **Test Result**

Ensure that the Raspberry Pi Pico is connected to the computer，click“Stop/Restart backend”.



Click “Run current script”, the code starts executing, we will see that When your hand covers the light-sensitive part of the photoresistor to simulate darkness, the Raspberry Pi Pico's built-in LED will light up. Then shake it in front of the PIR motion sensor with your other hand, the external LED will light up, too, and after a delay of a few seconds, the external LED will automatically turn off.

At the same time, the "Shell" window of Thonny IDE will print the delay time when the external LED lights up . If the sensitive part of the photoresistor is not covered, you can see that the the Raspberry Pi Pico's built-in LED lights go out , at this time, shake in front of the PIR motion sensor with your hand, the external LED is off. Press“Ctrl+C”or click“Stop/Restart backend”to exit the program.

