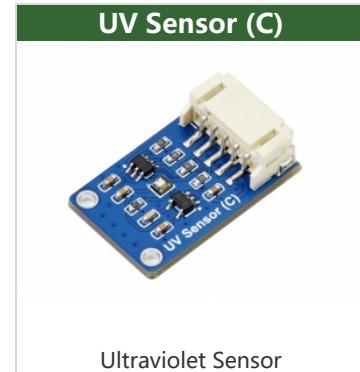


Overview

V Sensor (C) is an I2C digital UV sensor expansion board equipped with LTR390-UV-01, which can be used to measure UV and human visible light and can output light intensity data. UV Sensor (C) has the characteristics of sensitive sensing, fast response, and small size. Its core sensor LTR390-UV-01 is integrated on an expansion board (27 × 20mm), which can be easily integrated into the device.



Product Features

- Using LTR390-UV-01, can measure ultraviolet and visible light of the human eye.
- Built-in ADC, can directly output light intensity through the I2C interface, not easy to be disturbed by noise.
- With interrupt output and programmable upper and lower thresholds.
- On-board level conversion circuit, compatible with 3.3V/5V working level.
- Provide complete supporting information manual (Raspberry/Arduino/STM32 sample program and user manual, etc.)

Specifications

- Operating voltage: 3.3V/5V
- Sensor: LTR390-UV-01
- Response wavelength: 280 - 430nm
- Communication interface: I2C (fixed address: 0 x 53)
- Product size: 27 × 20mm
- Fixed aperture: 2.0mm

Applications

- Ultraviolet tester
- Outdoor ultraviolet detector
- Germicidal lamp

Interface Description

- Pin function

Pin Number	Identification	Pin Description
1	VCC	3.3V/5V power supply positive
2	GND	Power Ground
3	SDA	I2C data line
4	SCL	I2C clock line
4	INT	Interrupt output

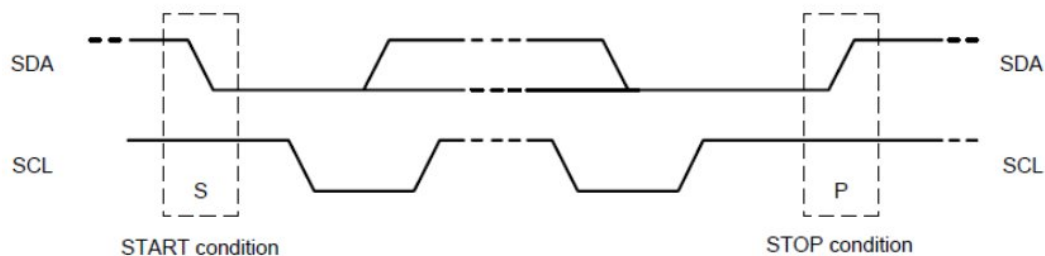
Hardware Description

Chip

This product adopts the LTR-390UV-01 UV sensor, which is an integrated low voltage I2C ambient light sensor (ALS) and UV light sensor (UVS), using a single microchip (2 x 2mm) lead-free surface mount package. The sensor converts light intensity into a digital output signal that can be directly used with the I2C interface. It provides a linear ALS response over a wide dynamic range, making it ideal for applications in high-brightness environments. The sensor has programmable interrupts with a hysteretic response to events, eliminating the need to poll the sensor for readings, and improving system efficiency.

Communication Protocol

I2C communication requires a data cable and a clock wire. Normally, the I2C bus has three signals during transmission: start, end, and response.



Start signal: When SCL is at a high level, SDA transitions from a high level to a low level to start transmitting data.

End signal: When SCL is at a high level, SDA transitions from a low level to a high level, ending data transfer.

Response signal: After receiving the 8-bit data, the IC that receives the data sends a specific low-level pulse to the IC that sends the data, indicating that the data has been received.

Hardware Description

Chip

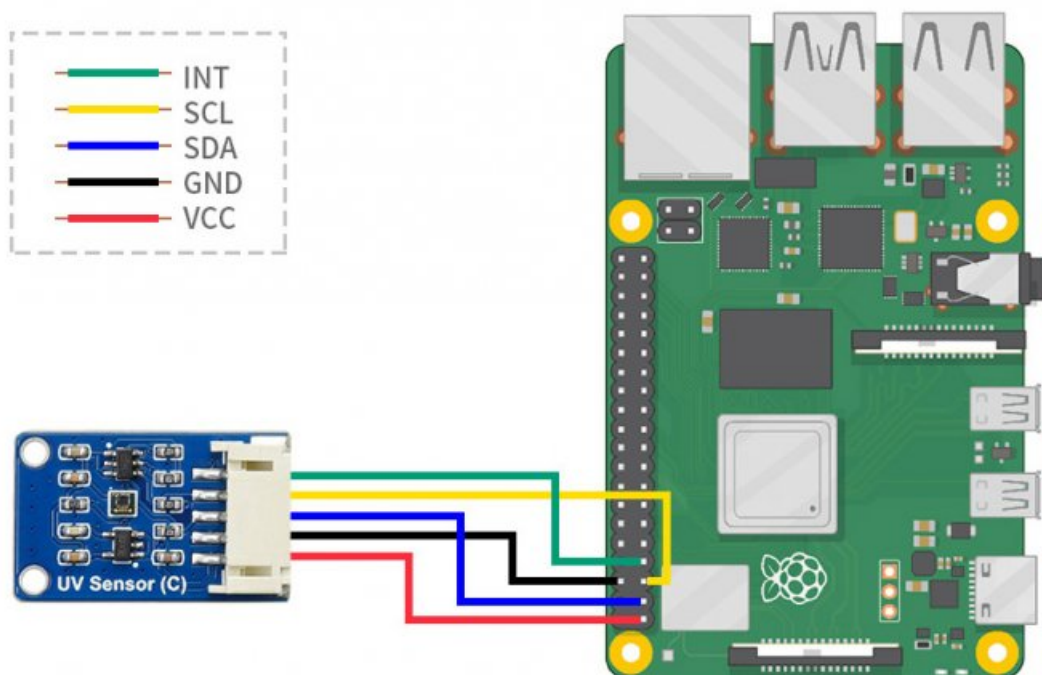
This product adopts the LTR-390UV-01 UV sensor, which is an integrated low voltage I2C ambient light sensor (ALS) and UV light sensor (UVS), using a single microchip (2 x 2mm) lead-free surface mount package. The sensor converts light intensity into a

digital output signal that can be directly used with the I2C interface. It provides a linear ALS response over a wide dynamic range, making it ideal for applications in high-brightness environments. The sensor has programmable interrupts with a hysteretic response to events, eliminating the need to poll the sensor for readings, and improving system efficiency.

Working with Raspberry Pi

This demo uses a Raspberry Pi 4 Model B, which provides RPI (Python) library demos.

Hardware connection



Enable I2C interface

- Execute in terminal:

```
sudo raspi-config  
#Select Interfacing Options -> I2C -> yes to start the i2c kernel driver.
```

```
Raspberry Pi Software Configuration Tool (raspi-config)  
  
1 Change User Password Change password for the current user  
2 Network Options Configure network settings  
3 Boot Options Configure options for start-up  
4 Localisation Options Set up language and regional settings to match your location  
5 Interfacing Options Configure connections to peripherals  
6 Overclock Configure overclocking for your Pi  
7 Advanced Options Configure advanced settings  
8 Update Update this tool to the latest version  
9 About raspi-config Information about this configuration tool  
  
<Select> <Finish>  
  
Raspberry Pi Software Configuration Tool (raspi-config)
```

```
P1 Camera    Enable/Disable connection to the Raspberry Pi Camera
P2 SSH       Enable/Disable remote command line access to your Pi using SSH
P3 VNC       Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI       Enable/Disable automatic loading of SPI kernel module
P5 I2C       Enable/Disable automatic loading of I2C kernel module
P6 Serial    Enable/Disable shell and kernel messages on the serial connection
P7 1-Wire    Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins
```

<Select>

<Back>

```
Would you like the ARM I2C interface to be enabled?
```

<Yes>

<No>

- then restart the raspberry pi:

```
sudo reboot
```

Installing Libraries

- Install BCM2835, open the Raspberry Pi terminal, and run the following command

```
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.60.tar.gz
tar zxvf bcm2835-1.60.tar.gz
cd bcm2835-1.60/
sudo ./configure
sudo make
sudo make check
sudo make install
```

- install wiringpi

```
sudo apt-get install wiringpi
#For Raspberry Pi 4B may need to be upgraded:
cd /tmp
wget https://project-downloads.drogon.net/wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
gpio -v
```

Download and run the test demo

```
sudo apt-get install p7zip-full
wget https://files.waveshare.com/upload/a/ab/UV_Sensor_C_Code.7z
7z x UV_Sensor_C_Code.7z -r -o./UV_Sensor_C_Code
sudo chmod 777 -R UV_Sensor_C_Code
```

python program

Execute in the UV_Sensor_C_Code directory:

```
cd UV_Sensor_C_Code/RPI
sudo python LTR390.py
```

- Note: The sensor has 2 working modes, measuring light intensity and UV light respectively.
- Note: The upper and lower thresholds of the interrupt are set by the program. If it is higher than or lower, it will trigger, and the INT pin will output a high pulse.

Working with STM32

This demo has been verified on NUCLEO-F103RB (chip model STM32RBT6). If you need to transplant, please pay attention to the relevant configuration and connection method.

Hardware connection

Connection with XNUCLEO-F103RB:

UV Sensor (C)	XNUCLEO-F103RB
VCC	3.3V/5V
GND	GND
SDA	SDA/D14/PB9
SCL	SCL/D15/PB8

Experimental phenomenon

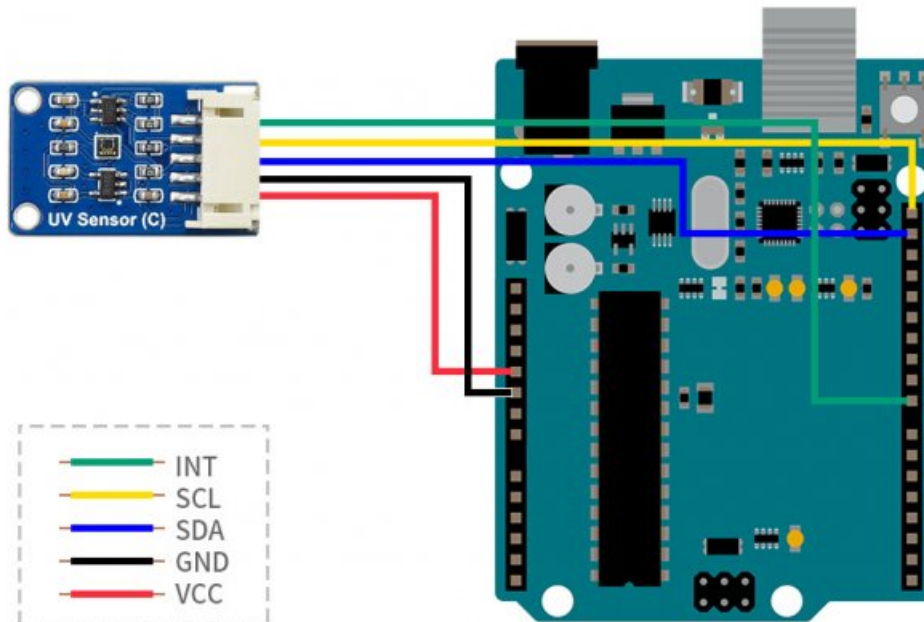
The default serial port 2 prints data, and using a UV lamp or UV pen to illuminate the sensor will print the UV value

```
[13:16:21.206]收←◆UV = 0
[13:16:21.708]收←◆UV = 0
[13:16:22.211]收←◆UV = 0
[13:16:22.714]收←◆UV = 0
[13:16:23.218]收←◆UV = 15
[13:16:23.721]收←◆UV = 17
[13:16:24.223]收←◆UV = 17
[13:16:24.726]收←◆UV = 23
[13:16:25.229]收←◆UV = 13
[13:16:25.733]收←◆UV = 16
[13:16:26.236]收←◆UV = 30
[13:16:26.738]收←◆UV = 0
[13:16:27.241]收←◆UV = 0
```

Working with Arduino

This example is tested on Arduino UNO. If other types of Arduino are used, please pay attention to whether the relevant pins are connected correctly.

Hardware connection



Install and compile the software (windows tutorial)

- [Arduino IDE installation tutorial](#)

Verifiers

Download the demo on the product encyclopedia interface, and then unzip it. This demo is written based on the Arduino library version, so the program needs to be copied to the Arduino library directory. Copy the folder `Waveshare_UV_C` in the Arduino directory in the next week's file to the libraries in the Arduino installation directory (`C:\Users\XXX\Documents\Arduino\libraries` or `C:\Program Files (x86)\Arduino\libraries`)

Open Arduino IDE: Click File (file) -> Example (example) to load the library, and check if there is the `Waveshare_UV_C` option, if so, the library is imported successfully

Open the ino project file in the example, select the corresponding model of the development board, select the corresponding COM port, compile the program, download it to the UNO, open the serial monitor, and use the UV pen or UV lamp to print the UV value.

Resources

- [Schematic](#)
- [Code](#)
- [Datasheets](#)

FAQ

Question:Why is it that after I power on as described in the tutorial, the serial port only prints a little prompt and no longer outputs any data or the output data is all 0?

Answer:

Please check whether the hardware connection is OK, especially the line sequence of SDA and SCL should not be reversed, and reconnect the sensor after power off and re-run the program.

Support

Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket, Our support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.

Working Time: 9 AM - 6 AM GMT+8 (Monday to Friday)

[Submit Now](#)