

TOF测距传感器VL53L1X用户手册



版本V1.0

目录

| | |
|-------------------------|----|
| 模块概述..... | 3 |
| 简介..... | 3 |
| 芯片说明..... | 3 |
| 规格参数..... | 4 |
| 外壳尺寸, 单位MM..... | 5 |
| 引脚说明..... | 5 |
| 应用示例-Arduino UNO..... | 5 |
| 硬件及连接..... | 5 |
| 程序及结果-Arduino IDE..... | 6 |
| 程序及结果-Mind+ 暂未支持..... | 8 |
| 程序及结果-Mixly 暂未支持..... | 8 |
| 附录..... | 9 |
| 附录1-UNO接口说明..... | 9 |
| 附录2-Arduino如何导入库? | 9 |
| 附录3-Mind+如何导入扩展库? | 10 |

| | |
|-------------------------|----|
| 附录4-Mixly如何导入扩展库? | 10 |
| 版本说明 | 11 |
| 联系我们 | 11 |

模块概述

简介

TOF测距传感器，采用VL53L1X芯片设计制作。通过I2C总线接口通信，可提供距离检测及输出，精确测距范围可达4米，快速测距频率可达50Hz。

与传统的测距传感器不同，VL53L1X 采用 ST 最新的 ToF 技术，集成了物理红外滤波器和光学元件，无论目标颜色和反射率如何，都可以进行距离测量，抗干扰能力更强。

TOF测距传感器具有统一的兼容乐高积木的外壳，可轻松完成乐高积木的拼接，实现创意设计。

芯片说明



VL53L1X是先进的飞行时间 (ToF) 激光测距传感器。它是市面上较快的微型ToF传感器，快速测距频率高达50 Hz，可在4m范围内精确测距。

它采用微型回流焊封装，集成了SPAD接收阵列、940nm不可见1类激光发射器、物理红外滤光片和光学器件，在各种环境照明条件（带各种盖片选项）下实现最佳的测距性能。

与传统的IR传感器不同，VL53L1X使用ST最新一代ToF技术，可测量在任何目标颜色和反射率时的绝对距离。

它还能通过对接收阵列上兴趣区 (ROI) 的大小进行编程来降低传感器FoV。

VL53L1X 具有三种距离模式(DM): 短距离、中距离和长距离。长距离模式允许达到4 m的最长可能测距距离。然而这个最大测距距离受环境光影响。近距离模式对环境光的免疫力更强，其最大测距距离通常限于1.3 m。

特征

- 完全集成的小型化模块
 - 尺寸：4.9x2.5x1.56 mm
 - 发射器：940 nm不可见激光器（1类）
 - SPAD（单光子雪崩二极管）接收阵列配集成透镜
 - 运行高级数字固件的低功耗微控制器
- 引脚排列兼容VL53L0X FlightSense™测距传感器
- 快速、精确的远程测距
 - 测量距离可达400 cm
 - 高达50 Hz的测距频率
- 典型全视野（FoV）： 27 °
- 接收阵列上兴趣区（ROI）的大小可编程，可以降低传感器FoV
- 接收阵列上兴趣区（ROI）的位置可编程，从主机进行多区域操作控制
- 方便集成
 - 单回流焊元件
 - 可以隐藏在各种盖片材料下
 - 软件驱动和代码示例用于交钥匙测距应用
 - 单电源（2v8）
 - I²C接口（可达400 kHz）
 - 更多VL53L1X详情请参考数据手册。

规格参数

工作电压：DC 3.3-5V

IC：VL53L1X

通信方式：I2C，地址0x29

VL53L1X 检测距离：4~400cm

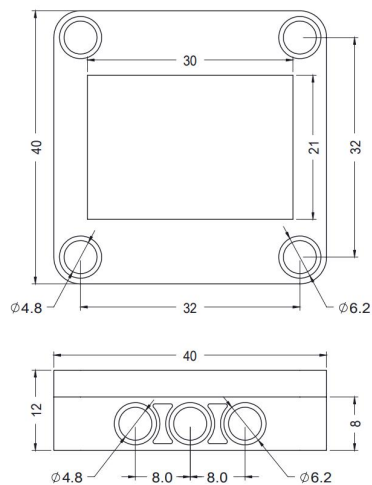
测距精度：±5%

测距时间(min)：20ms (高速模式)，200ms(高精度模式)

测距角度：27°

激光波长：940nm

外壳尺寸，单位MM



引脚说明



应用示例-Arduino UNO

硬件及连接

⚠ 注意：硬件应用于示例演示，可能需要另购；如有不明请咨询本司客服！

Legoduino Pro主板、TOF测距传感器、线材。

TOF测距传感器的 G、V、SDA、SCL分别连接 主板的IIC端口GND、VCC、

SDA(A4)、SCL(A5)引脚。

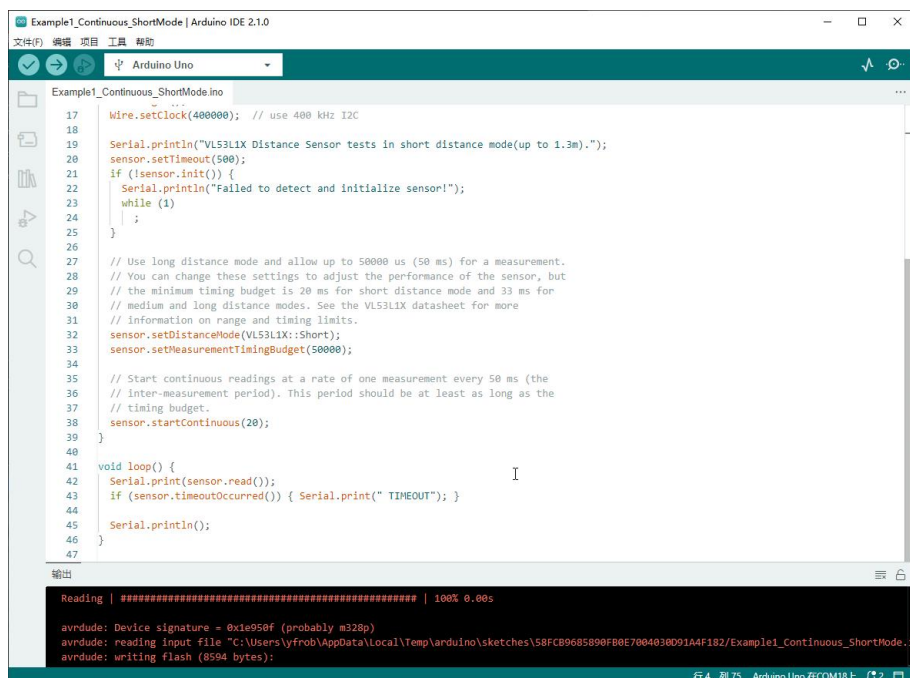


程序及结果-Arduino IDE

⚠ 注意：程序需要添加库文件“vl53l1x-arduino-master.zip”，否则无法正常编译，添加方式见附录2。

例程一、短距离模式下测量距离

打开例程“Example1_Continuous_ShortMode”编译上传，此程序为短距离模式，最大测距为1.3M。



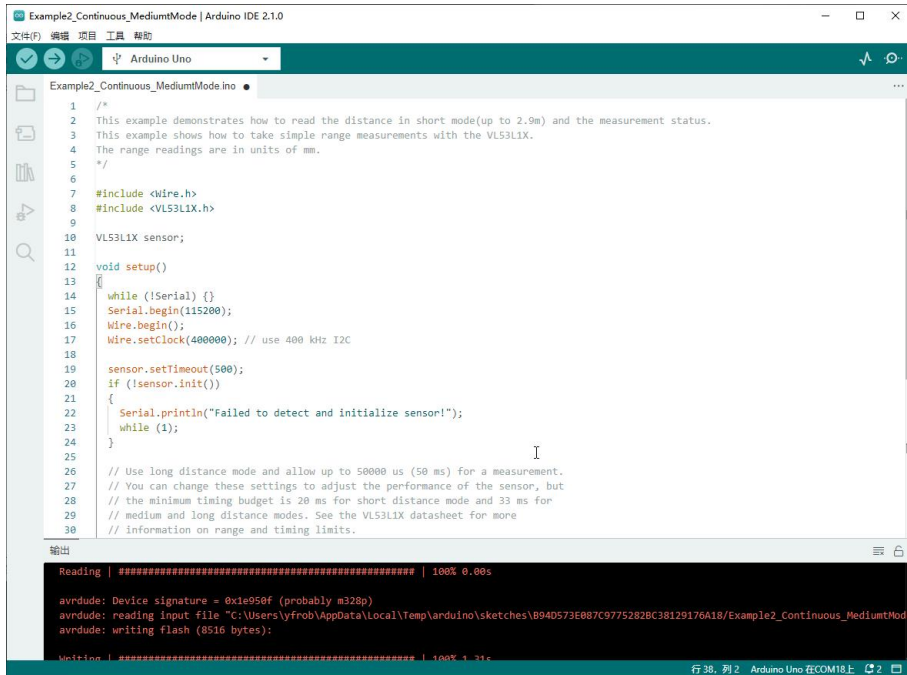
```

Example1_Continuous_ShortMode.ino
17 Wire.setClock(400000); // use 400 kHz I2C
18
19 Serial.println("VL53L1X Distance Sensor tests in short distance mode(up to 1.3m).");
20 sensor.setTimeout(500);
21 if (!sensor.init()) {
22   Serial.println("Failed to detect and initialize sensor!");
23   while (1)
24     ;
25 }
26
27 // Use long distance mode and allow up to 50000 us (50 ms) for a measurement.
28 // You can change these settings to adjust the performance of the sensor, but
29 // the minimum timing budget is 20 ms for short distance mode and 33 ms for
30 // medium and long distance modes. See the VL53L1X datasheet for more
31 // information on range and timing limits.
32 sensor.setDistanceMode(VL53L1X::Short);
33 sensor.setMeasurementTimingBudget(50000);
34
35 // Start continuous readings at a rate of one measurement every 50 ms (the
36 // inter-measurement period). This period should be at least as long as the
37 // timing budget.
38 sensor.startContinuous(20);
39 }
40
41 void loop() {
42   Serial.print(sensor.read());
43   if (sensor.timeoutOccurred()) { Serial.print(" TIMEOUT"); }
44
45   Serial.println();
46 }
47
输出
Reading | ##### | 100% 0.00s
avrdude: Device signature = 0x1e950f (probably m328p)
avrdude: reading input file "C:\Users\yfrob\AppData\Local\Temp\arduino\sketches\58FCB9685890FB8E7804930091A4F182\Example1_Continuous_ShortMode.1
avrdude: writing flash (8594 bytes):
行 4, 列 75 Arduino Uno 在 COM18E

```

例程二、中距离模式下测量距离

打开例程“Example2_Continuous_MediumtMode”编译上传，此程序为中距离模式，最大测距为2.9M。



```

1  /*
2  This example demonstrates how to read the distance in short mode(up to 2.9m) and the measurement status.
3  This example shows how to take simple range measurements with the VL53L1X.
4  The range readings are in units of mm.
5  */
6
7  #include <Wire.h>
8  #include <VL53L1X.h>
9
10 VL53L1X sensor;
11
12 void setup()
13 {
14   while (!Serial) {}
15   Serial.begin(115200);
16   Wire.begin();
17   Wire.setClock(400000); // use 400 kHz I2C
18
19   sensor.setTimeout(500);
20   if (!sensor.init())
21   {
22     Serial.println("Failed to detect and initialize sensor!");
23     while (1);
24   }
25
26   // Use long distance mode and allow up to 50000 us (50 ms) for a measurement.
27   // You can change these settings to adjust the performance of the sensor, but
28   // the minimum timing budget is 20 ms for short distance mode and 33 ms for
29   // medium and long distance modes. See the VL53L1X datasheet for more
30   // information on range and timing limits.

```

输出

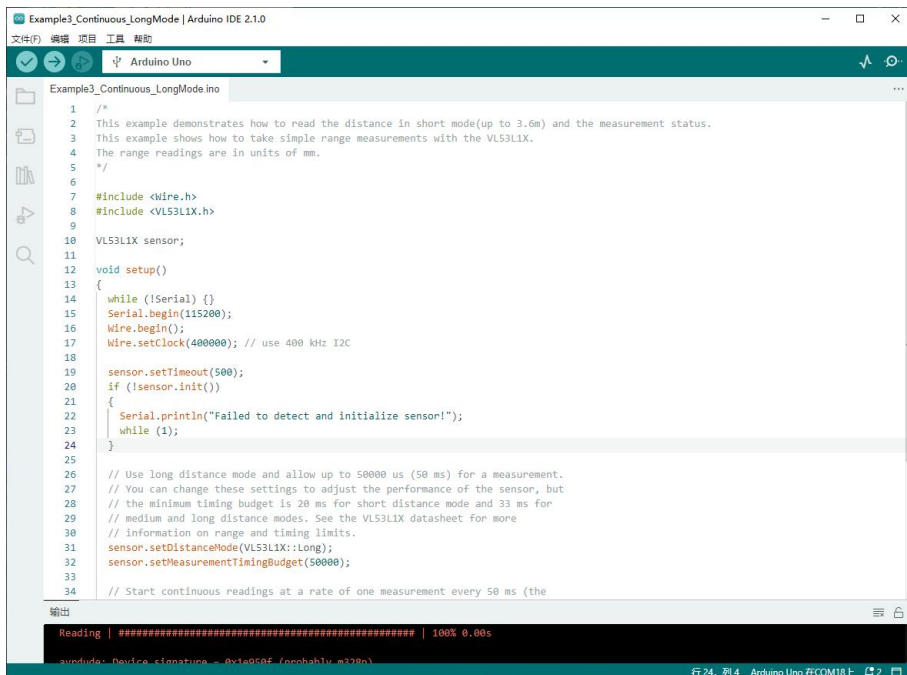
```

Reading | ##### | 100% 0.00s
avrdude: Device signature = 0x1e950f (probably m328p)
avrdude: reading input file "C:\Users\yfrog\AppData\Local\Temp\arduino\sketches\B94D573E087C9775282BC38129176A18/Example2_Continuous_MediumMode.ino"
avrdude: writing flash (8516 bytes):

```

例程三、长距离模式下测量距离

打开例程“Example3_Continuous_LongMode”编译上传，此程序为长距离模式，最大测距为4M（最大测距时受环境光线影响）。



```

1  /*
2  This example demonstrates how to read the distance in short mode(up to 3.6m) and the measurement status.
3  This example shows how to take simple range measurements with the VL53L1X.
4  The range readings are in units of mm.
5  */
6
7  #include <Wire.h>
8  #include <VL53L1X.h>
9
10 VL53L1X sensor;
11
12 void setup()
13 {
14   while (!Serial) {}
15   Serial.begin(115200);
16   Wire.begin();
17   Wire.setClock(400000); // use 400 kHz I2C
18
19   sensor.setTimeout(500);
20   if (!sensor.init())
21   {
22     Serial.println("Failed to detect and initialize sensor!");
23     while (1);
24   }
25
26   // Use long distance mode and allow up to 50000 us (50 ms) for a measurement.
27   // You can change these settings to adjust the performance of the sensor, but
28   // the minimum timing budget is 20 ms for short distance mode and 33 ms for
29   // medium and long distance modes. See the VL53L1X datasheet for more
30   // information on range and timing limits.
31   sensor.setDistanceMode(VL53L1X::Long);
32   sensor.setMeasurementTimingBudget(50000);
33
34   // Start continuous readings at a rate of one measurement every 50 ms (the

```

输出

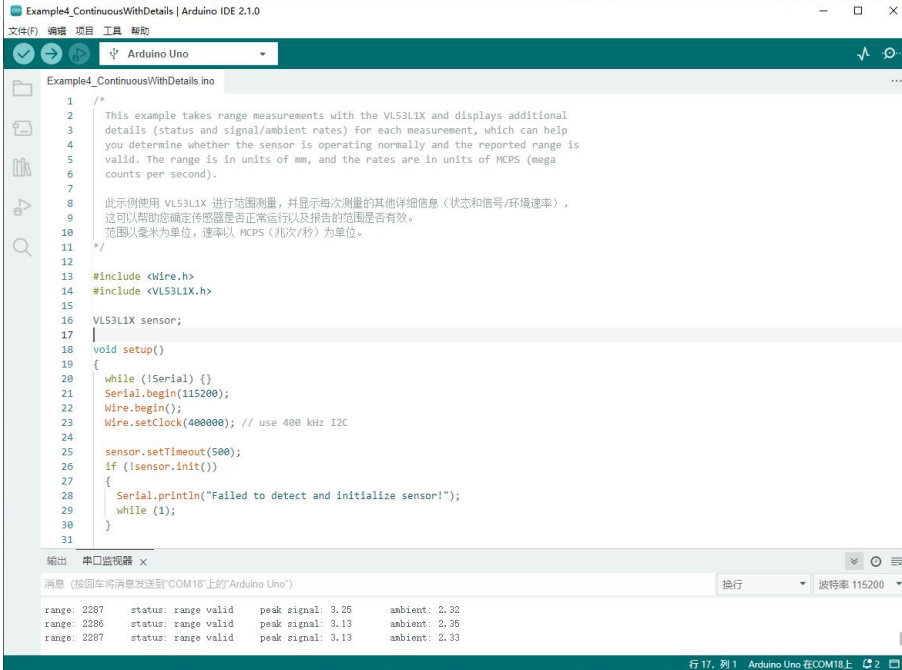
```

Reading | ##### | 100% 0.00s
avrdude: Device signature = 0x1e950f (probably m328p)

```

例程四、中距离模式下测量距离

打开例程“Example4_ContinuousWithDetails”编译上传，此程序为长距离模式，最大测距为4M（最大测距时受环境光线影响），并显示传感器是否正常工作及测距结果是否有效的细节。



```

1  /*
2  This example takes range measurements with the VL53L1X and displays additional
3  details (status and signal/ambient rates) for each measurement, which can help
4  you determine whether the sensor is operating normally and the reported range is
5  valid. The range is in units of mm, and the rates are in units of MCPS (mega
6  counts per second).
7
8  此示例使用 VL53L1X 进行范围测量，并显示每次测量的其他详细信息（状态和信号/环境速率），
9  这可以帮助您确定传感器是否正常运行以及报告的范围是否有效。
10  范围以毫米为单位，速率以 MCPS（兆次/秒）为单位。
11  */
12
13 #include <Wire.h>
14 #include <VL53L1X.h>
15
16 VL53L1X sensor;
17
18 void setup()
19 {
20   while (!Serial) {}
21   Serial.begin(115200);
22   Wire.begin();
23   Wire.setClock(400000); // use 400 kHz I2C
24
25   sensor.setTimeout(500);
26   if (!sensor.init())
27   {
28     Serial.println("Failed to detect and initialize sensor!");
29     while (1);
30   }
31 }

```

输出 串口监视器 x

消息 (按回车将消息发送到“COM18”上的 Arduino Uno)

```

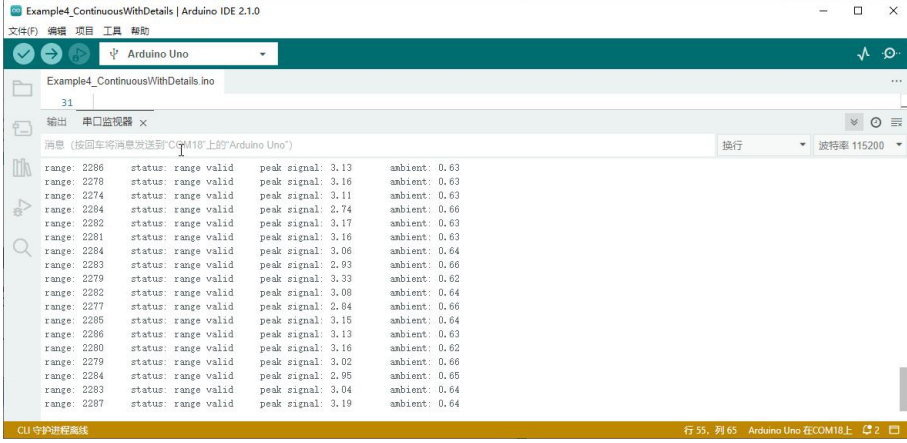
range: 2287 status: range valid peak signal: 3.25 ambient: 2.32
range: 2286 status: range valid peak signal: 3.13 ambient: 2.35
range: 2287 status: range valid peak signal: 3.13 ambient: 2.33

```

行 17, 列 1 Arduino Uno 在 COM18 上

测距结果

观察串口打印测得值；range：测距值（mm）；status：数据是否有效；peak signal/ambient两个值暂未研究出是何意思，有兴趣的自行研究下。



```

31

```

输出 串口监视器 x

消息 (按回车将消息发送到“COM18”上的 Arduino Uno)

```

range: 2286 status: range valid peak signal: 3.13 ambient: 0.63
range: 2278 status: range valid peak signal: 3.16 ambient: 0.63
range: 2274 status: range valid peak signal: 3.11 ambient: 0.63
range: 2284 status: range valid peak signal: 2.74 ambient: 0.66
range: 2282 status: range valid peak signal: 3.17 ambient: 0.63
range: 2281 status: range valid peak signal: 3.16 ambient: 0.63
range: 2284 status: range valid peak signal: 3.06 ambient: 0.64
range: 2283 status: range valid peak signal: 2.93 ambient: 0.66
range: 2279 status: range valid peak signal: 3.33 ambient: 0.62
range: 2282 status: range valid peak signal: 3.08 ambient: 0.64
range: 2277 status: range valid peak signal: 2.84 ambient: 0.66
range: 2285 status: range valid peak signal: 3.15 ambient: 0.64
range: 2286 status: range valid peak signal: 3.13 ambient: 0.63
range: 2280 status: range valid peak signal: 3.16 ambient: 0.62
range: 2279 status: range valid peak signal: 3.02 ambient: 0.66
range: 2284 status: range valid peak signal: 2.95 ambient: 0.65
range: 2283 status: range valid peak signal: 3.04 ambient: 0.64
range: 2287 status: range valid peak signal: 3.19 ambient: 0.64

```

CLI 守护进程离线

行 55, 列 65 Arduino Uno 在 COM18 上

程序及结果-Mind+ 暂未支持

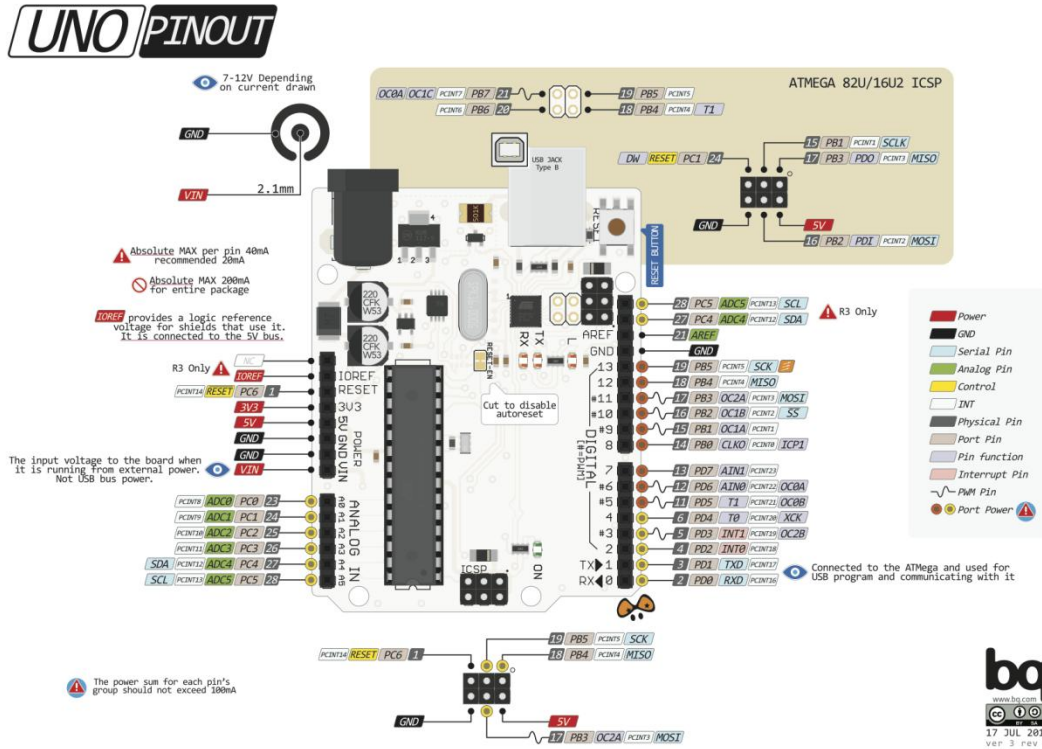
更多YFRobot扩展库参考：[YFRobot库 For Mind+ - YFRobotwiki](#)

程序及结果-Mixly 暂未支持

更多YFRobot扩展库参考：[YFRobot三方库ForMixly1.0/2.0 - YFRobotwiki](#)

附录

附录1-UNO接口说明



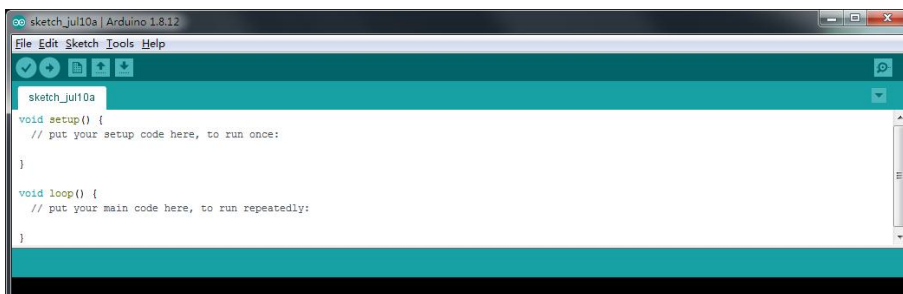
注：UNO官方版本和兼容版本大部分功能都相同

附录2-Arduino如何导入库？

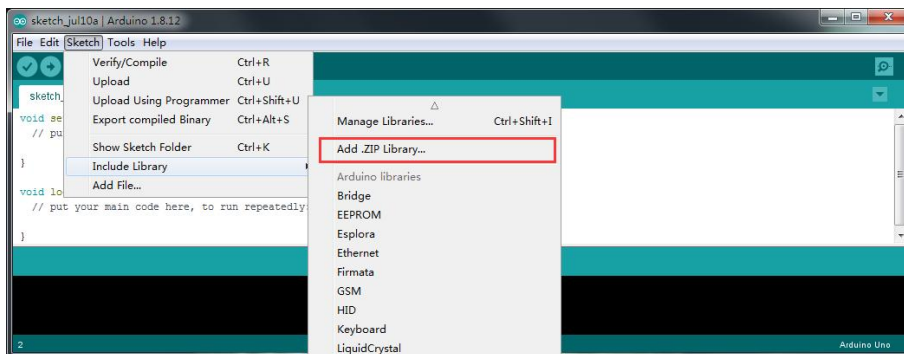
教程中有些需要使用库；如何将其导入到自己的Arduino IDE编译器中？

Arduino IDE导入库视频教程：[点击视频链接查看。](#)

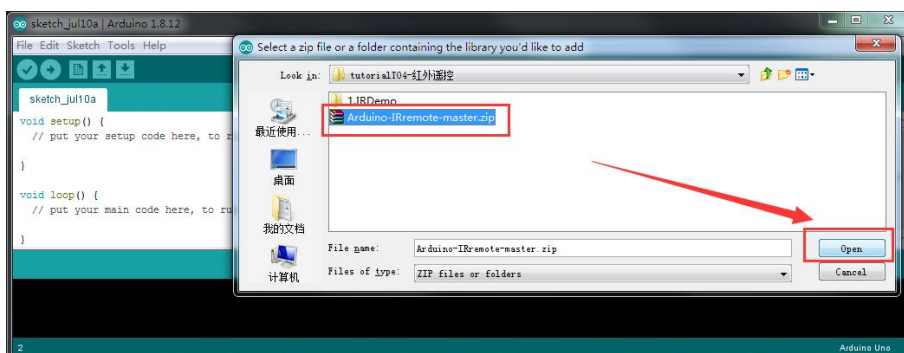
➤ 第一步：打开ARUINO IDE



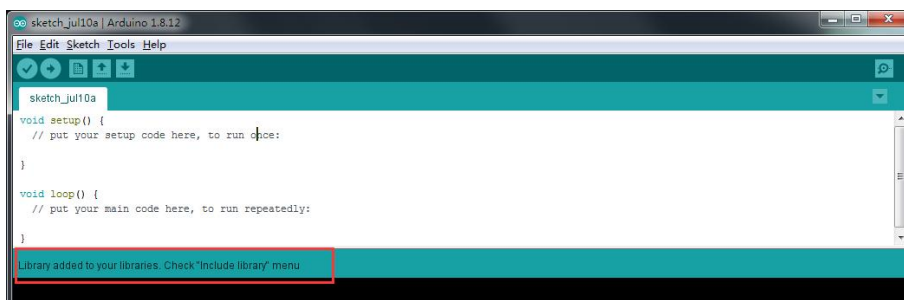
➤ 第二步：Sketch > Include Library > Add ZIP Library...



- 第三步：找到需要添加的zip格式库文件，选择并点击打开（Open）



- 第四步：导入成功后提示



附录3-Mind+如何导入扩展库？

Mind+图形化软件导入扩展库视频教程：点击[链接](#)。

附录4-Mixly如何导入扩展库？

Mixly图形化软件导入扩展库视频教程：点击[链接](#)。

版本说明

| 版本 | 日期 | 内容 |
|------|----------|----|
| V1.0 | 20230510 | 初版 |
| | | |
| | | |
| | | |
| | | |
| | | |

联系我们

| | | | |
|-----------|---|-------|--|
| YFROBOT网站 | www.yfrobot.com / www.yfrobot.com.cn | | |
| 手机 | 17696701116 (微信/QQ同号) | | |
| QQ群 | 243067479 | | |
| 邮件 | yfrobot@qq.com | | |
| 技术微信二维码 |  | 微信公众号 | YFRobotStudio  |

免责声明和版权公告

本文中的信息，包括供参考的 URL 地址，如有变更，恕不另行通知。

文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

文中提到的所有商标名称、商标和注册商标均属其各所所有者的财产，特此声明。

版权归 © 2023 YFROBOT所有。保留所有权利

