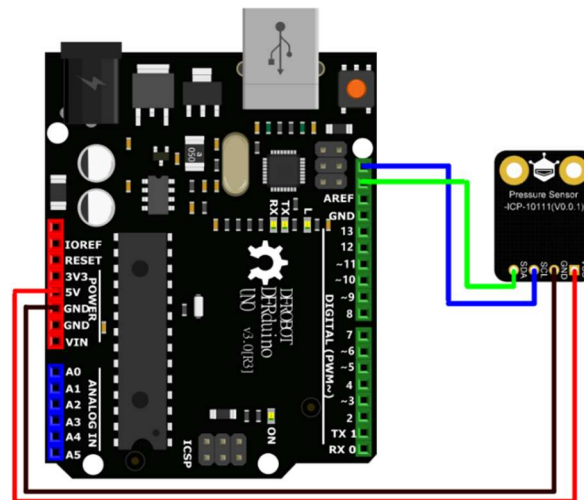


1. 测试 Sensor_ICP10111.ino

将 ICP10111 传感器参考如下连线图连接至 Arduino，并且上传当前文件夹中的 Sensor_ICP10111.ino 至 Arduino



串口监视器中正常出现读数，说明传感器正常。传感器可以被连接至 Beacon 测试。

```
Arduino Uno
Sensor_SGP40.ino  sensirion_arch_config.h  sensirion_voc_algorithm.c  sensirion_voc_algorithm.h

1  #include <Wire.h>
2
3  extern "C" {
4  #include "sensirion_arch_config.h"
5  #include "sensirion_voc_algorithm.h"
6  };
7
8  #define MODULE_I2C_ADDRESS ((uint8_t)0x59) // 传感器的 I2C 地址，此处的 sc
9
10 #define TEST_OK 0xD400
11
12 #define CMD_HEATER_OFF_H 0x36
13 #define CMD_HEATER_OFF_L 0x15
14 #define CMD_HEATER_OFF_SIZE 2
15
16 #define CMD_MEASURE_TEST_H 0x28
17 #define CMD_MEASURE_TEST_L 0x0E
18 #define CMD_MEASURE_TEST_SIZE 2
19
20 #define CMD_SOFT_RESET_H 0x00
21 #define CMD_SOFT_RESET_L 0x06
22 #define CMD_SOFT_RESET_SIZE 2
23
24 #define CMD_MEASURE_RAW_H 0x26
25 #define CMD_MEASURE_RAW_L 0x0F

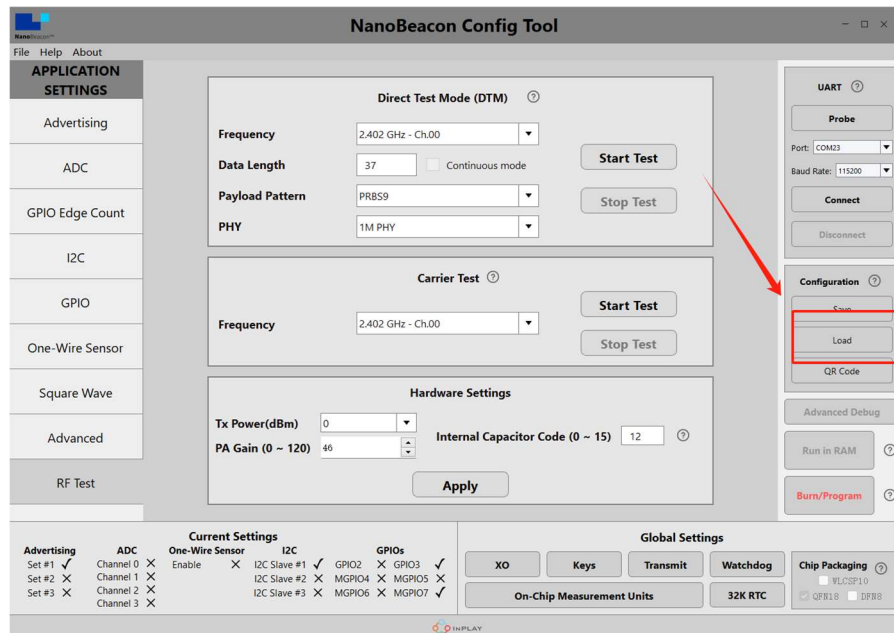
Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM4')

vocIndex = 119
74 F0 B8
vocIndex = 119
75 25 B8
vocIndex = 118
75 50 B3
vocIndex = 115
75 75 C0
vocIndex = 113
75 A9 B9
vocIndex = 110
75 DC B4
vocIndex = 107
```

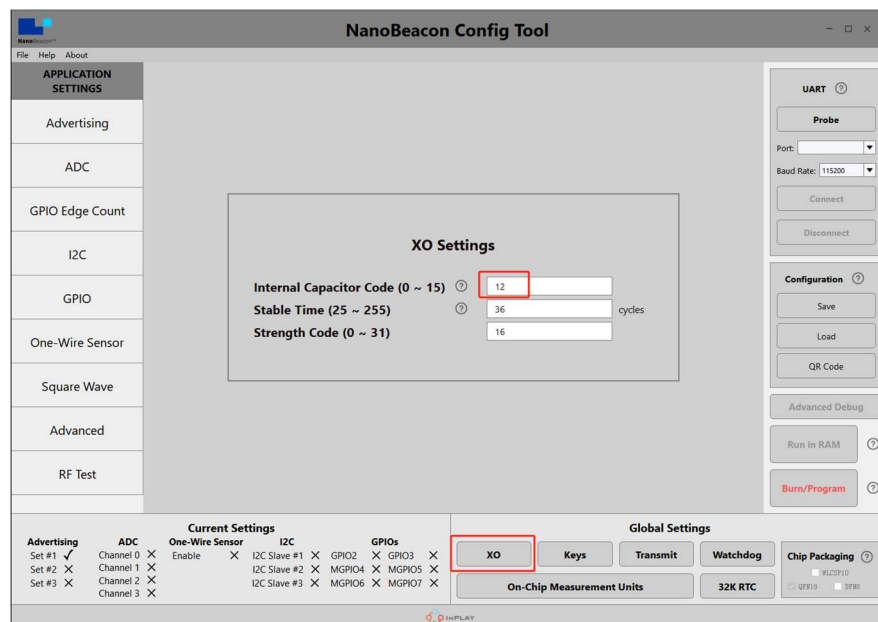
2. 烧录 Beacon 并且连接传感器

请您使用 USB-TTL 转换器将.cfg 文件烧录进 Beacon。

NanoBeacon Config Tool 中可以 Load 本文件夹中的 SGP40.cfg 文件。



检查 XO 电容配置是否为 12

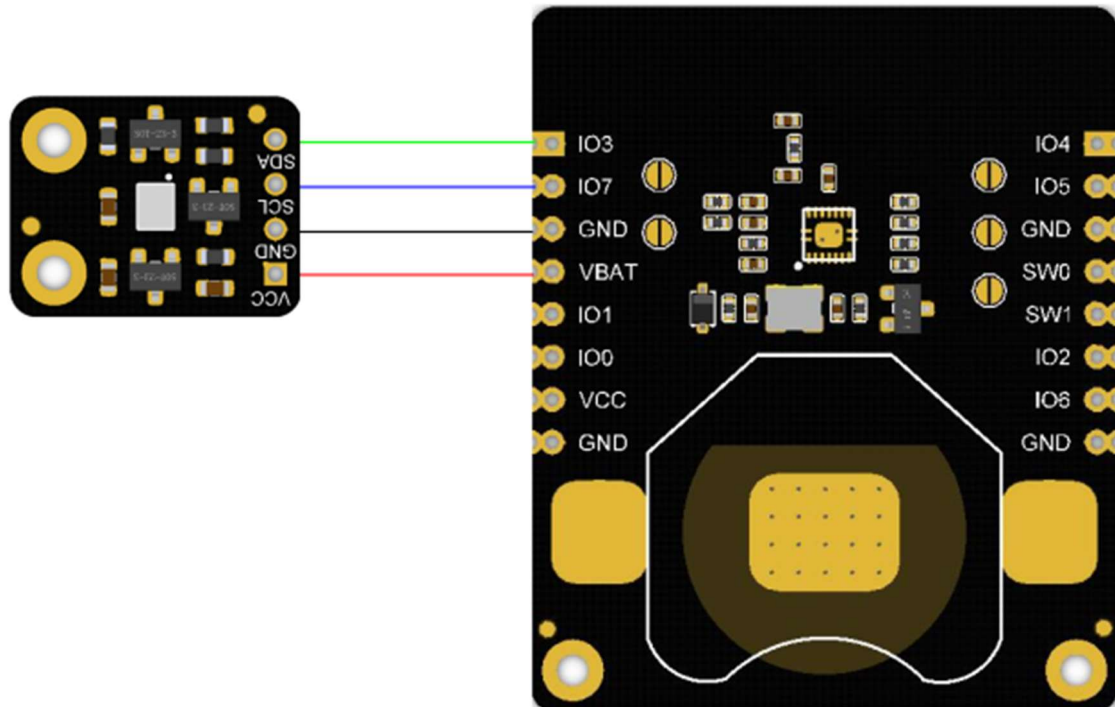


烧录流程请参考 Beacon 的 wiki:

https://wiki.dfrobot.com.cn/_SKU_TEL0168_Fermion_BLE_%E4%BC%A0%E6%84%9F%E5%99%A8%E4%BF%A1%E6%A0%87#target_4

在烧录完成后，参考下图连接 Beacon 和传感器。

注：我们的.cfg 示例文件默认 SCL->GPIO7, SDA->GPIO3。

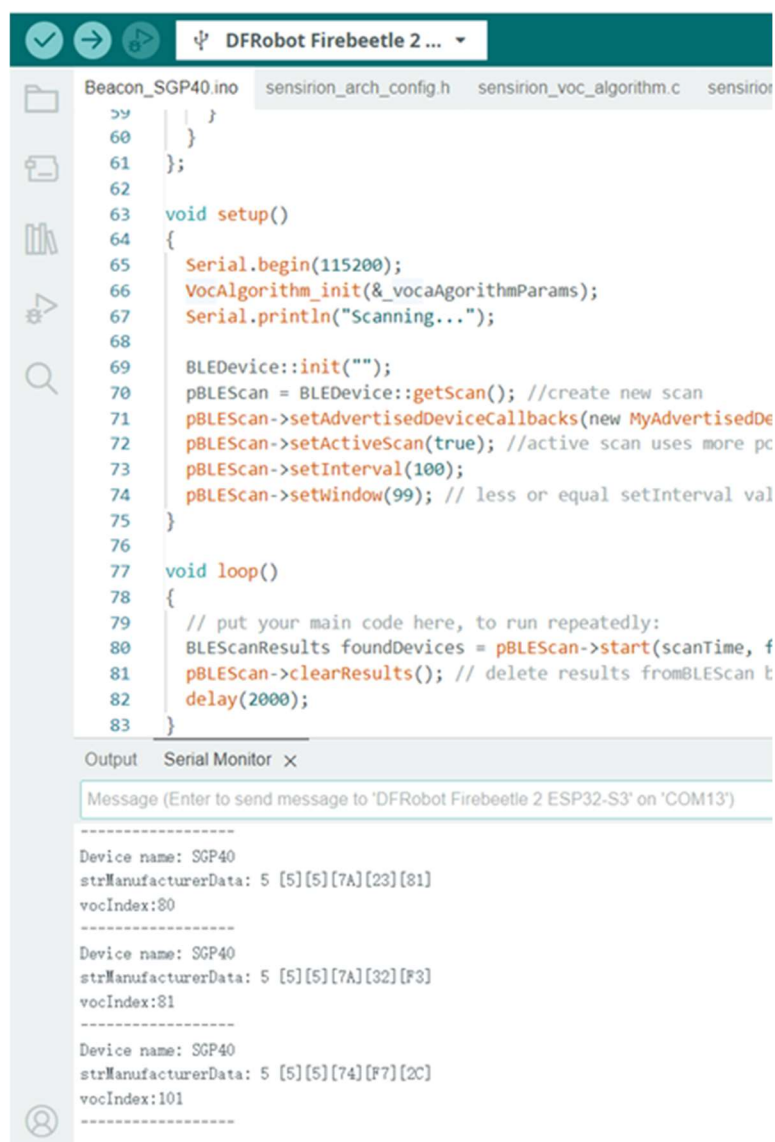


3. 上传 ESP32 代码并获取读数

将同目录下的 Beacon_SGP40.ino 上传至 ESP32 主板。

并且将 Beacon 和传感器供电，供电方式可选 CR2032 纽扣电池，或者 VCC 和 GND 输入 3.3V。

您将会看到串口监视器中打印相关数据。



The screenshot shows the Arduino IDE interface. The top toolbar includes icons for checking, running, and uploading code, along with a dropdown menu showing 'DFRobot Firebeetle 2 ...'. The file explorer on the left shows the project files: Beacon_SGP40.ino, sensirion_arch_config.h, sensirion_voc_algorithm.c, and sensirion_voc_algorithm.h. The main editor displays the Beacon_SGP40.ino file with the following code:

```
59   }
60   }
61 };
62
63 void setup()
64 {
65   Serial.begin(115200);
66   VocAlgorithm_init(&_vocaAlgorithmParams);
67   Serial.println("Scanning...");
68
69   BLEDevice::init("");
70   pBLEScan = BLEDevice::getScan(); //create new scan
71   pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallback());
72   pBLEScan->setActiveScan(true); //active scan uses more power
73   pBLEScan->setInterval(100);
74   pBLEScan->setWindow(99); // less or equal setInterval value
75 }
76
77 void loop()
78 {
79   // put your main code here, to run repeatedly:
80   BLEScanResults foundDevices = pBLEScan->start(scanTime, false);
81   pBLEScan->clearResults(); // delete results from BLEScan object and start over
82   delay(2000);
83 }
```

Below the code editor is the 'Serial Monitor' window, which is currently empty. The title bar shows 'Output' and 'Serial Monitor' with a close button. The message input field contains the text: 'Message (Enter to send message to "DFRobot Firebeetle 2 ESP32-S3" on "COM13")'.