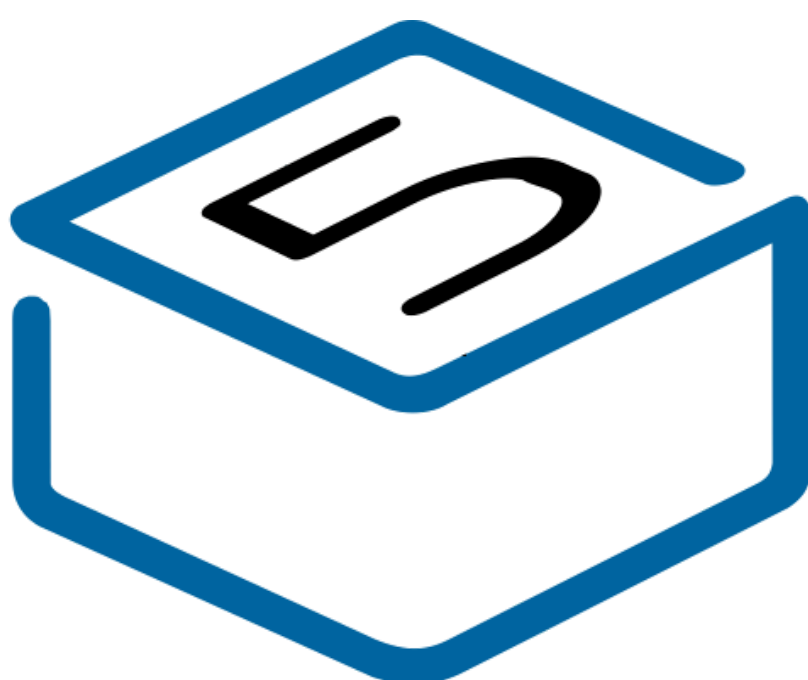


Arduino Nesso N1

model: TPX00227



M5STACK

2025

CONTENTS

1.OUTLINE	3
1.1. Arduino Nesso N1.....	3
2.SPECIFICATIONS.....	4
2.1. Specifications.....	4
2.2. Module Size.....	4
3.Quick start.....	5
3.1. SCAN Wi-Fi.....	5
3.2. SCAN BLE Device.....	6
3.3. SCAN LoRa Device.....	7
4. FCC Warning.....	8
Arduino Install.....	9

1. OUTLINE

Arduino Nesso N1 is a fully functional IoT development board that uses the ESP32C6 main controller and comes with an external 16M Flash storage, ensuring ample program space; it is equipped with a 1.14-inch **ST7789v2 touchscreen** for intuitive display and interaction; it features an integrated **SX1262** LoRa module (915MHz), paired with an **FPC antenna** and a detachable **LoRa** antenna, supporting long-range wireless communication; it integrates a **BMI270** sensor for motion and environmental detection, and includes a **buzzer**, an IR emitter, and 3 user buttons, providing versatile feedback and control; moreover, it offers **Grove** interfaces and **EXT IO** expansion ports, along with a built-in **200mAh** battery, making it an ideal solution for smart home, remote monitoring, industrial automation, and other applications.

1.1. Arduino Nesso N1

1. Communication Capabilities

- Main Controller: ESP32C6
- Wireless Communication: Wi-Fi (2.4GHz), Bluetooth Low Energy (BLE) 5.0
- LoRa Module: SX1262 (915MHz), equipped with an FPC antenna and a detachable LoRa antenna for long-range communication

2. Processor and Performance

- Processor Model: ESP32C6 (RISC-V architecture)
- Clock Speed: Up to 160MHz (typical)
- Performance Features: Integrated Wi-Fi, BLE, and LoRa capabilities for versatile IoT applications

3. Display and Input

- Display: 1.14-inch ST7789v2 color touchscreen for clear visual output and direct user interaction
- User Buttons: 3 programmable buttons for power control and custom functions
- Additional Input: IR emitter for remote control or signaling

4. Memory

- External Flash: 16MB external Flash for program storage and data logging
- On-Chip Memory: Internal SRAM (capacity depends on the ESP32C6 specifications)

5. Power Management

- Battery: Built-in 200mAh LiPo battery for portable operation
- Charging: USB input (Type-C or similar, depending on board design) for battery charging and power supply
- Power Protection: On-board circuitry to safeguard against overcurrent or overvoltage

6. GPIO Pins and Programmable Interfaces

- Grove Ports: 2 Grove connectors for rapid sensor and module integration
- EXT IO Expansion: Additional pins for extended I/O usage
- Interface Protocols: Supports standard interfaces such as I²C, SPI, UART, etc.

7. Others

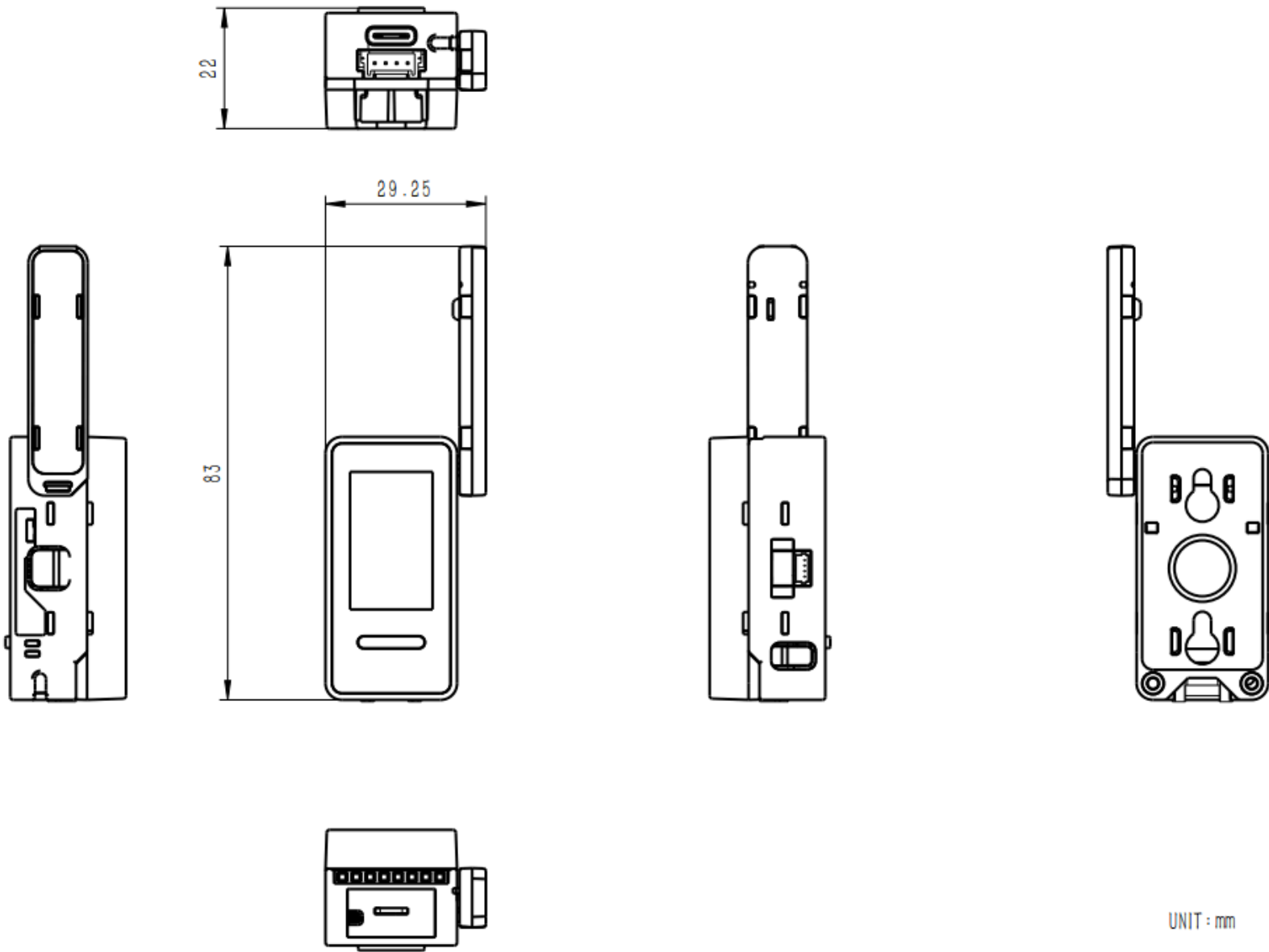
- Sensors: BMI270 for motion and orientation detection
- Audio Feedback: Buzzer for alerts and notifications
- Form Factor: Compact design with a dedicated antenna slot for easy installation
- Use Cases: Ideal for smart home systems, remote monitoring, industrial automation, and other IoT applications

2. SPECIFICATIONS

2.1. Specifications

Parameter and Specification	Value
MCU	ESP32C6 (RISC-V up to 160MHz)
Communication Capability	Wi-Fi (2.4GHz), BLE 5.0, LoRa (SX1262, 915MHz)
Supply Voltage	5V input (via USB)
Flash Storage	16MB external Flash
Display	1.14 inch ST7789v2 color touchscreen
Sensor	BMI270 (motion & orientation)
Buzzer	On-board buzzer
IR Emitter	On-board IR LED
Expansion Interface	2 Grove ports, EXT IO expansion
Battery	200mAh LiPo
Operating Temperature	0-40° C
Wi-Fi Frequency	2.4GHz
BLE Working Frequency	2.4GHz
Module Size	83.0 x 29.25 x 22mm
Manufacturer	M5Stack Technology Co., Ltd

2.2. Module Size



3. QUICK START

Before you do this step, look at the text in the final appendix: Installing Arduino

3.1. Print WiFi information

- 1. Open Arduino IDE (Refer to https://docs.m5stack.com/en/arduino/arduino_ide for the installation guide for the development board and software)
- 2. Select the ESP32C6 DEV Module board and the corresponding port, then upload the code
- 3. Open the serial monitor to display the scanned WiFi and signal strength information

Nesso_WiFi.ino

```
1  #include "WiFi.h"
2
3  void setup() {
4      Serial.begin(115200);
5      WiFi.mode(WIFI_STA);
6      WiFi.disconnect();
7      delay(100);
8
9      Serial.println("Scanning for WiFi networks...");
10     int n = WiFi.scanNetworks();
11     if (n == 0) {
12         Serial.println("No networks found.");
13     } else {
14         Serial.print(n);
15         Serial.println(" networks found.");
16         for (int i = 0; i < n; ++i) {
17             Serial.print(i + 1);
18             Serial.print(": ");
19             Serial.print(WiFi.SSID(i));
20             Serial.print(" (");
21             Serial.print(WiFi.RSSI(i));
22             Serial.print(")");
23             Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN) ? " " : "*");
24             delay(10);
25         }
26     }
27     Serial.println("");
28 }
29
30 void loop() {
31     // put your main code here, to run repeatedly:
32 }
33
```

COM20

17:20:58.755 -> Scanning for WiFi networks...
17:20:58.755 -> 35 networks found.
17:20:58.755 -> 1: M5-UiFlow-Zone (-34) *
17:20:58.801 -> 2: XLOT (-34) *
17:20:58.801 -> 3: M5-R&D (-39) *
17:20:58.801 -> 4: WiFi_ADF4 (-39) *
17:20:58.801 -> 5: DIANJIXZ (-45) *
17:20:58.848 -> 6: Xiaomi_32BD (-47) *
17:20:58.848 -> 7: M5-UiFlow-Zone (-53) *
17:20:58.848 -> 8: M5-UiFlow-Zone (-54) *
17:20:58.848 -> 9: CenturyLink2842 (-55) *
17:20:58.848 -> 10: M5-UiFlow-Zone (-56) *
17:20:58.895 -> 11: esp-shui (-56) *
17:20:58.895 -> 12: CMCC-FSNg (-57) *
17:20:58.895 -> 13: YUESHIQI-602 (-57) *
17:20:58.895 -> 14: ChinaNet-hZsm (-57) *

☒ Autoscroll ☒ Show timestamp

you've pressed Ctrl but nothing was sent. Should you select a line ending?

Newline

Before you do this step, look at the text in the final appendix: Installing Arduino

3.2. Print BLE information

- 1. Open Arduino IDE (Refer to https://docs.m5stack.com/en/arduino/arduino_ide for the installation guide for the development board and software)
- 2. Select the ESP32C6 DEV Module board and the corresponding port, then upload the code
- 3. Open the serial monitor to display the scanned BLE and signal strength information

Nesso_BLE_ino.ino

```
1  #include "BLEDevice.h"
2
3  class MyAdvertisedDeviceCallbacks: public BLEAdvertisedDeviceCallbacks {
4      void onResult(BLEAdvertisedDevice advertisedDevice) {
5          Serial.print("Advertised Device: ");
6          Serial.println(advertisedDevice.toString().c_str());
7      }
8  };
9
10 void setup() {
11     Serial.begin(115200);
12     Serial.println("Starting BLE scan...");
13
14     BLEDevice::init("");
15
16     BLEScan* pBLEScan = BLEDevice::getScan();
17     pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallbacks());
18     pBLEScan->setActiveScan(true); // Active scan uses more power, but get
    results faster
19     pBLEScan->start(10, false); // Scan for 10 seconds
20 }
21
22 void loop() {
23     // Do nothing here
24 }
25
```

COM18

16:32:55.340	->	Advertised Device: Name: , Address: 29:b2:79:b9:a3:a0, manufacturer data: 060001092022f2ad5527637974d01222aa793bcbcb9fc4c359e2392776a, rssi: -95
16:32:55.340	->	Advertised Device: Name: , Address: 68:ab:bc:a6:82:56, manufacturer data: 8f030a108212005482a6bcab6881, rssi: -72
16:32:55.387	->	Advertised Device: Name: , Address: 4c:11:0b:4a:ac:06, manufacturer data: 4c0010052818e6dfc1, txPower: 8, rssi: -78
16:32:55.387	->	Advertised Device: Name: , Address: c4:23:5c:6d:7f:cc, manufacturer data: 4c0012020003, rssi: -78
16:32:55.387	->	Advertised Device: Name: , Address: 7c:c2:94:11:dd:b3, manufacturer data: 8f030a10bb1900b1dd1194c27c81, rssi: -90
16:32:55.434	->	Advertised Device: Name: , Address: 69:9a:a5:ca:0e:76, manufacturer data: 4c001007381fa49766f208, txPower: 12, rssi: -87
16:32:55.481	->	Advertised Device: Name: , Address: 68:8a:2d:9d:69:9a, manufacturer data: 4c000719010e202b778f01000a5a7b38b9d862679f9aa0147c93dfb9a3, rssi: -92
16:32:55.481	->	Advertised Device: Name: , Address: 46:21:43:b4:e4:8f, manufacturer data: 4c0009081302c0a802531b581608006aad6eb4cfc9d7, rssi: -86
16:32:55.481	->	Advertised Device: Name: , Address: 68:13:24:e2:c9:a6, rssi: -94, serviceData: 0x00000000
16:32:55.528	->	rer data: 4c0012020000, rssi: -75
16:32:55.528	->	Advertised Device: Name: , Address: 4d:7a:15:80:e0:e4, manufacturer data: 4c0016080083cf28ec2b91b1, rssi: -75
16:32:55.575	->	Advertised Device: Name: , Address: 0d:4f:0e:0f:b8:6b, manufacturer data: 06000109202270c24b9ec6b7806f55379bea22271ecd7e87c71f99cb35, rssi: -92
16:32:55.575	->	Advertised Device: Name: , Address: 43:85:45:a1:4f:84, manufacturer data: 4c000908130cc0a81f071b5813080a88ba7d27f9c700, rssi: -81
16:32:55.622	->	Advertised Device: Name: , Address: a4:c1:38:8d:a7:00, rssi: -74, serviceData: 0x00000000
16:32:55.622	->	Advertised Device: Name: , Address: fa:e7:06:2b:fd:91, manufacturer data: 4c0012023503, rssi: -78
16:32:55.714	->	Advertised Device: Name: , Address: c3:3e:25:29:00:03, manufacturer data: 4c0012020003, rssi: -74
16:32:55.714	->	Advertised Device: Name: , Address: 52:88:46:95:91:08, manufacturer data: 4c00160800d660375f0003bf, rssi: -73
16:32:55.806	->	Advertised Device: Name: , Address: 6a:c3:bb:88:c2:0b, manufacturer data: 4c0010050e18874880, txPower: 12, rssi: -89
16:32:55.991	->	Advertised Device: Name: , Address: 4b:c9:66:74:75:f0, manufacturer data: 4c00100607194fa9cd38, txPower: 12, rssi: -87
16:32:55.991	->	Advertised Device: Name: , Address: 24:e8:e2:9b:75:46, manufacturer data: 4c0013080a4d1f30f2970b00, rssi: -91
16:32:56.038	->	Advertised Device: Name: , Address: 64:3d:63:13:1f:b0, manufacturer data: 4c00100607194fa9cd38, txPower: 12, rssi: -82
16:32:56.129	->	Advertised Device: Name: , Address: c1:55:39:b6:23:30, manufacturer data: 4c0012020000, rssi: -69
16:32:56.184	->	Advertised Device: Name: , Address: 41:a0:2a:ea:27:15, manufacturer data: 4c00160800579e01df5e3cae, rssi: -94
16:32:56.184	->	Advertised Device: Name: , Address: dd:3a:2f:71:cc:4f, manufacturer data: 4c0012020003, rssi: -90
16:32:56.265	->	Advertised Device: Name: , Address: f1:79:78:04:24:72, manufacturer data: 4c0012020003, rssi: -84
16:32:56.265	->	Advertised Device: Name: , Address: 73:d0:c7:76:2d:cd, manufacturer data: 4c0010073f1be2cc95d138, txPower: 7, rssi: -77
16:32:56.405	->	Advertised Device: Name: , Address: 75:d9:97:51:7d:8e, manufacturer data: 4c001007211fb4e4ccdc78, txPower: 12, rssi: -84
16:32:56.452	->	Advertised Device: Name: , Address: e4:84:07:a4:3e:e9, rssi: -91
16:32:56.452	->	Advertised Device: Name: , Address: 2e:da:35:f1:e5:1c, manufacturer data: 0600010f2022042879d9cedeb21fc16d6033b9bb7deb6b4e88513f2830, rssi: -95
16:32:56.452	->	Advertised Device: Name: , Address: cd:4e:ff:37:55:dd, manufacturer data: 4c0012020002, rssi: -91
16:32:56.500	->	Advertised Device: Name: , Address: 71:ab:11:45:16:08, manufacturer data: 4c0010053b18f2b4c3, txPower: 12, rssi: -87
16:32:56.545	->	Advertised Device: Name: , Address: 4e:bb:9b:58:79:b4, manufacturer data: 4c00160800c1b1dbbac7dd93, rssi: -66
16:32:56.590	->	Advertised Device: Name: , Address: dc:5d:0a:32:f6:cd, manufacturer data: 4c0012020000, rssi: -88
16:32:57.096	->	Advertised Device: Name: , Address: 65:c0:b9:6e:b8:49, manufacturer data: 4c0010052298728c65, txPower: 8, rssi: -89
16:32:57.329	->	Advertised Device: Name: , Address: 63:70:68:f2:c1:6f, manufacturer data: 4c00160800bb73dcc3dc3fa9, rssi: -86
16:32:57.329	->	Advertised Device: Name: , Address: d5:24:79:0c:93:f0, manufacturer data: 4c0012020001, rssi: -87
16:32:57.699	->	Advertised Device: Name: , Address: 42:bc:23:c2:3a:25, manufacturer data: 4c000c0e007f2849c2940c9d352a1085d4dc1006431d064dde18, rssi: -94
16:32:58.026	->	Advertised Device: Name: , Address: c4:8f:62:41:70:9d, manufacturer data: 4c0012020000, rssi: -94
16:32:58.026	->	Advertised Device: Name: , Address: d6:1e:a5:0c:5b:4e, manufacturer data: 4c001219395de24f1f2dd0ff3eb13c218d86153fee2b613140f7a80194, rssi: -73
16:32:58.213	->	Advertised Device: Name: , Address: fb:01:b0:e5:b4:ed, manufacturer data: 4c0012020002, rssi: -68
16:32:58.351	->	Advertised Device: Name: , Address: cd:55:86:51:87:a7, manufacturer data: 4c0012020003, rssi: -78
16:32:58.537	->	Advertised Device: Name: , Address: d2:e8:b8:38:e8:06, manufacturer data: 4c0012025401, rssi: -98
16:32:58.583	->	Advertised Device: Name: , Address: d0:17:51:8f:06:7e, manufacturer data: 4c0012026e00071106d0de3ee5e0414d36927a38cec0059ba4, rssi: -88

3.3. Print LoRa information

- 1. Open Arduino IDE (Refer to https://docs.m5stack.com/en/arduino/arduino_ide for the installation guide for the development board and software)
- 2. Select the ESP32C6 DEV Module board and the corresponding port, then upload the code
- 3. Open the serial monitor to display the scanned LoRa and signal strength information

```
1  #include <Arduino.h>
2  #include <M5GFX.h>
3  #include <RadioLib.h>
4  #include <Wire.h>
5
6  #include <lgfx/v1/panel/Panel_ST7789.hpp>
7  #include <lgfx/v1/touch/Touch_FT5x06.hpp>
8
9  #include "driver/gpio.h"
10 #include "freertos/FreeRTOS.h"
11 #include "freertos/task.h"
12
13
14 #include "BMI270-Sensor-API/bmi270.h"
15 #include "TinyIRSender.hpp"
16 // bit operation
17 #define setbit(x, y) x |= (0x01 << y)
18 #define clrbit(x, y) x &= ~(0x01 << y)
19 #define reversebit(x, y) x ^= (0x01 << y)
20 #define getbit(x, y) ((x) >> (y) & 0x01)
21
22 // I2C device addr
23 #define PI4IO_M_ADDR 0x43
24 #define PI4IO_B_ADDR 0x44
25 #define BQ27220_ADDR 0x55
26 #define AW32001_ADDR 0x49
27
28 // PI4IO registers
29 #define PI4IO_REG_CHIP_RESET 0x01
30 #define PI4IO_REG_IO_DIR 0x03
31 #define PI4IO_REG_OUT_SET 0x05
32 #define PI4IO_REG_OUT_H_IM 0x07
33 #define PI4IO_REG_IN_DEF_STA 0x09
34 #define PI4IO_REG_PULL_EN 0x0B
35 #define PI4IO_REG_PULL_SEL 0x0D
36 #define PI4IO_REG_IN_STA 0x0F
37 #define PI4IO_REG_INT_MASK 0x11
38 #define PI4IO_REG_IRQ_STA 0x13
39 // SPI
40 #define CONFIG_MISO_GPIO 22
41 #define CONFIG_SCLK_GPIO 20
42 #define CONFIG_MOSI_GPIO 21
43
```

transmission finished!

[LoRa]

StickC6L #32

rss: -75.0dBm, snr:5.75dB

Frequency error: -220.03 hz

[LoRa]

StickC6L #33

rss: -75.0dBm, snr:6.00dB

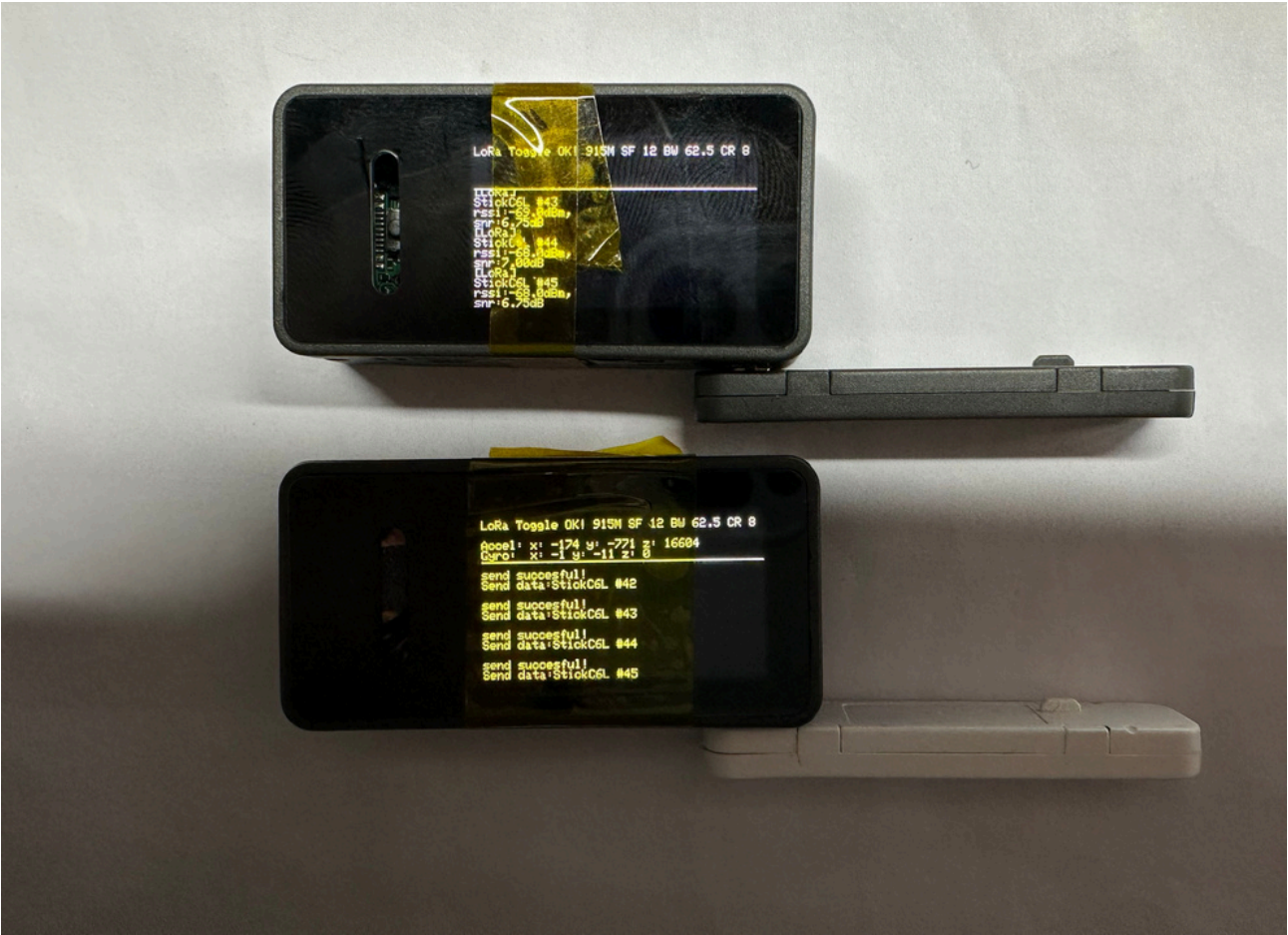
Frequency error: -222.69 hz

[LoRa]

StickC6L #34

rss: -74.0dBm, snr:6.50dB

Frequency error: -225.84 hz



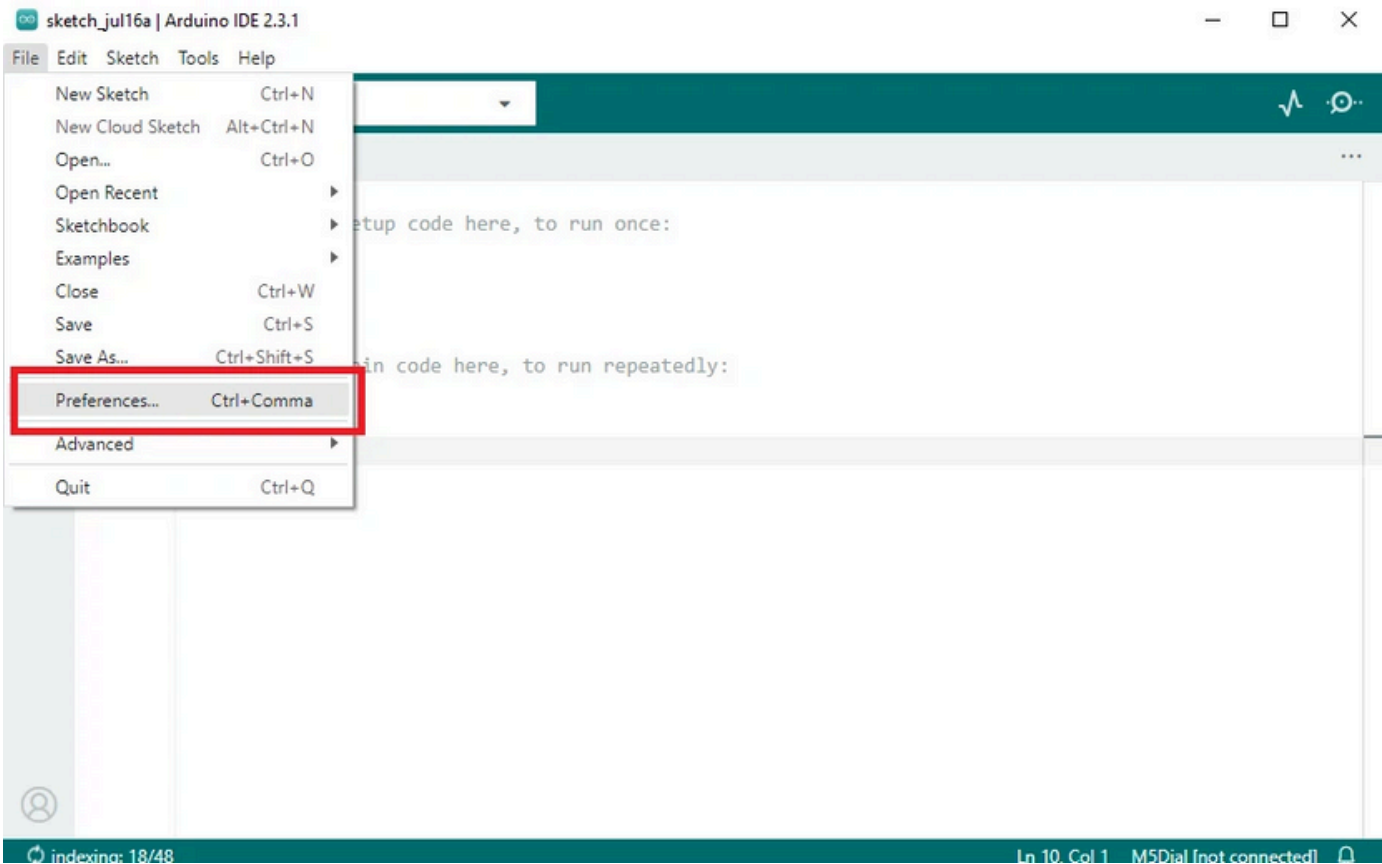
Arduino Install

一. Installing Arduino IDE(<https://www.arduino.cc/en/Main/Software>)

Click to visit the Arduino official website , and select the installation package for your operating system to download.

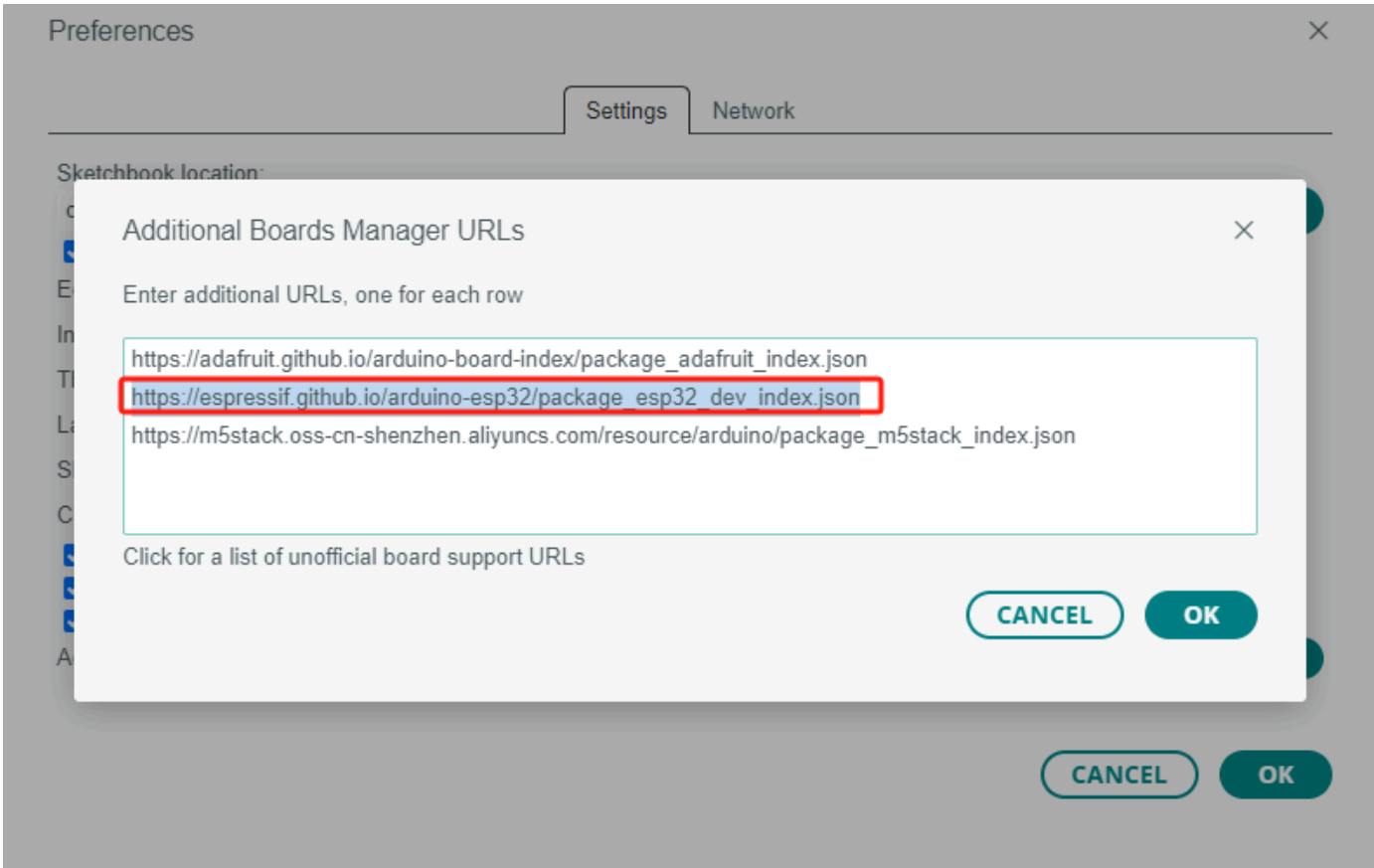
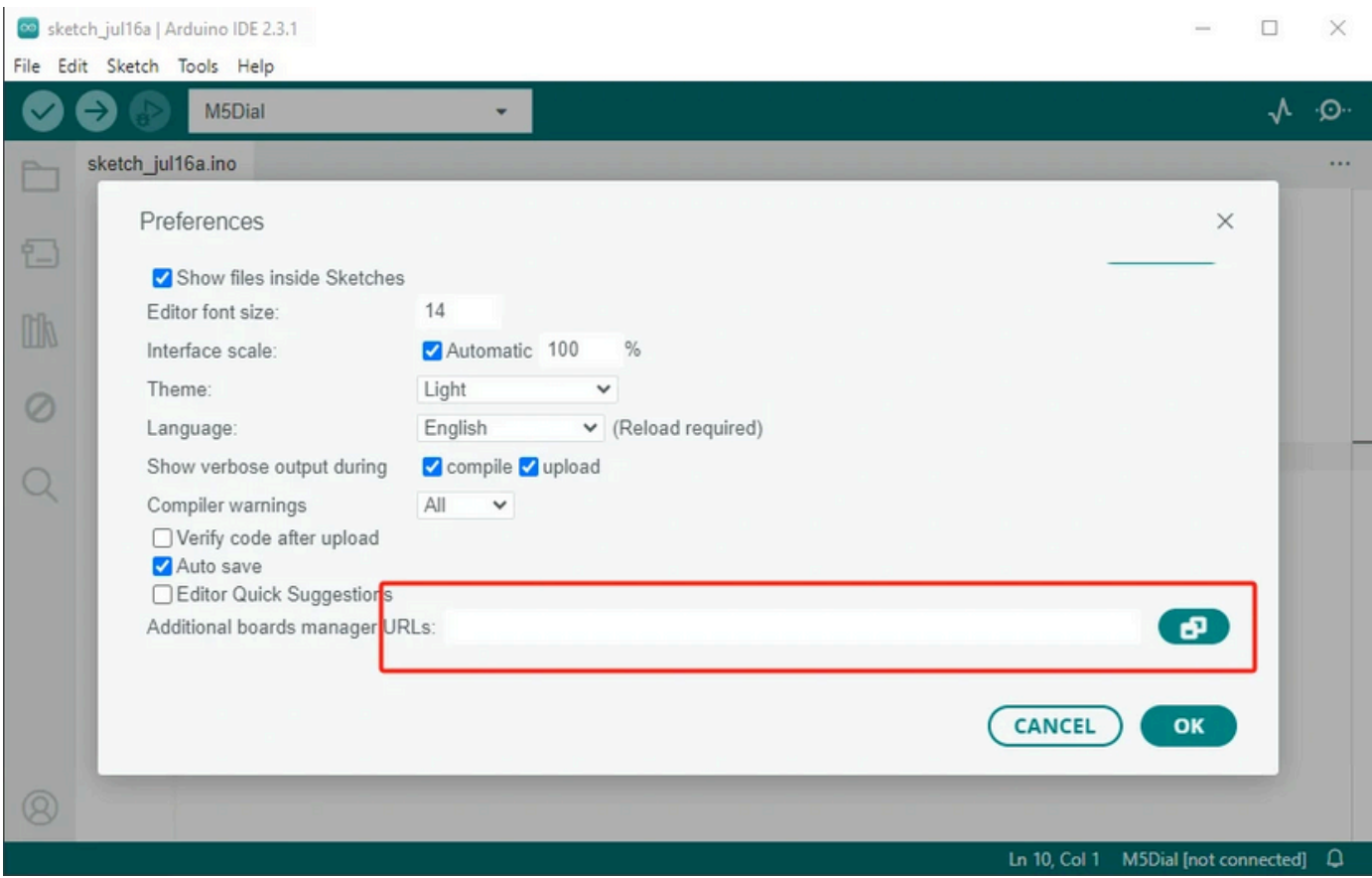
二. Installing Arduino Board Management

1. The Board Manager URL is used to index the development board information for a specific platform. In the Arduino IDE menu, select File -> Preferences

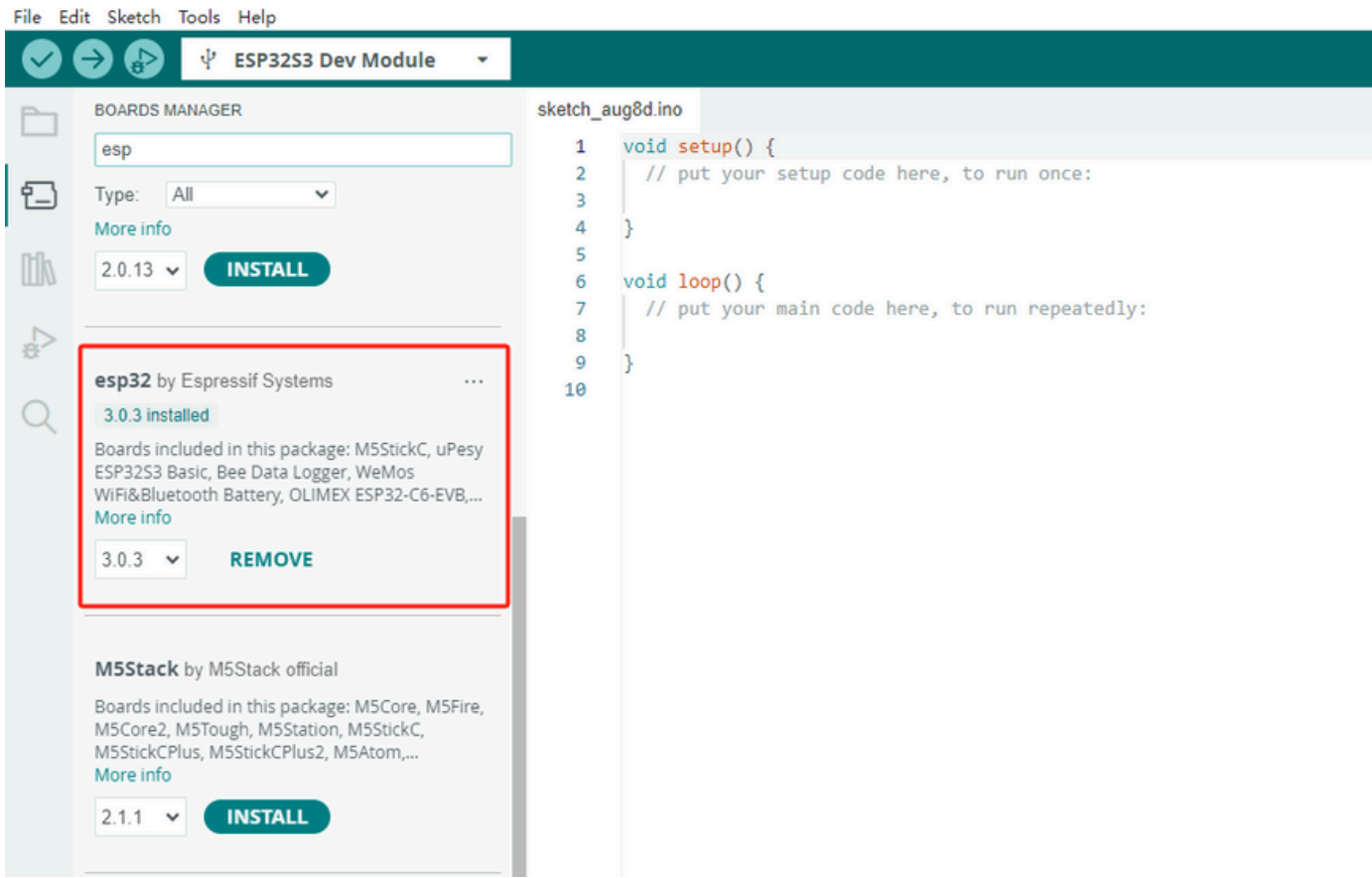


2.Copy the ESP board management URL below into the Additional Board Manager URLs: field, and save.

https://espressif.github.io/arduino-esp32/package_esp32_dev_index.json

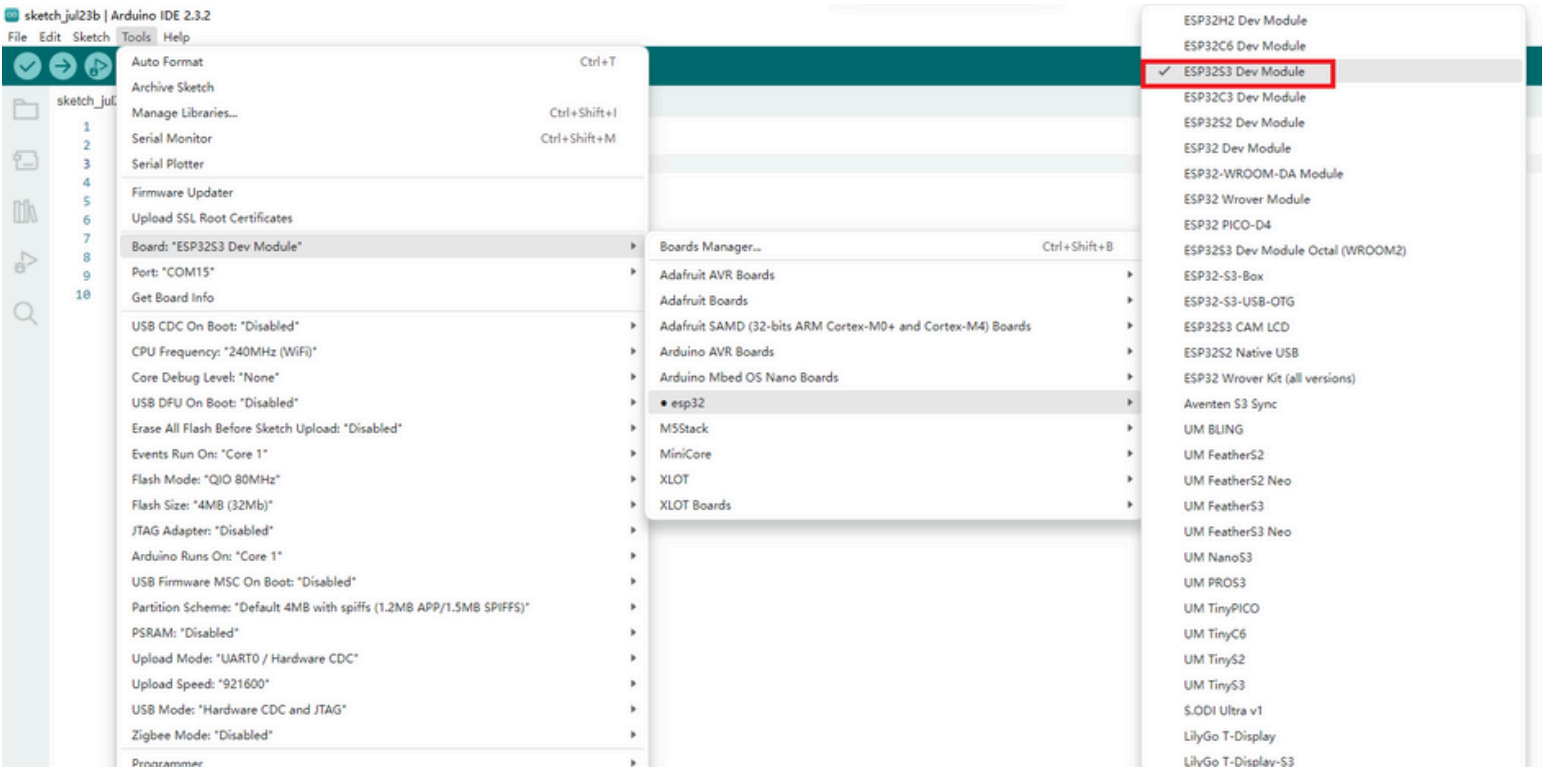


3. In the sidebar, select Board Manager, search for ESP, and click Install.



4. In the sidebar, select Board Manager, search for M5Stack, and click Install.

Depending on the product used, select the corresponding development board under Tools -> Board -> M5Stack -> {ESP32S3 DEV Module board}.



5. Connect the device to your computer with a data cable to upload the program

FCC Statement

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: 1) this device may not cause harmful interference, and 2) this device must accept any interference received, including interference that may cause undesired operation.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

ISED Statement:

This device contains licence-exempt transmitter(s)/receiver(s) that comply with Innovation, Science and Economic Development Canada's licence-exempt RSS(s). Operation is subject to the following two conditions:

- (1) This device may not cause interference; and
- (2) This device must accept any interference, including interference that may cause undesired operation of the device.

Cet appareil contient des émetteurs/récepteurs exempts de licence qui sont conformes aux CNR exemptés de licence d'Innovation, Sciences et Développement économique Canada. L'exploitation est soumise aux deux conditions suivantes :

- (1) Cet appareil ne doit pas causer d'interférence ; et
- (2) Cet appareil doit accepter toute interférence, y compris celles qui peuvent entraîner un fonctionnement indésirable de l'appareil.

Radiation Exposure Statement

SAR tests are conducted using standard operating positions according standards with the device transmitting at its highest certified power level in all tested frequency bands, although the SAR is determined at the highest certified power level, the actual SAR level of the device while operating can be well below the maximum value.

This model product has been tested and meets the RF exposure guidelines with the test distance 0mm for body worn. The highest SAR_{1g} value is 0.511 W/Kg. The use of belt clips, holsters and similar accessories should not contain metallic components in its assembly. The use of accessories that do not satisfy these requirements may not comply with RF exposure requirements, and should be avoided