

Arduino Nesso N1 LoRa 通信

Arduino Nesso N1 LoRa 通信相关 API 与案例程序。

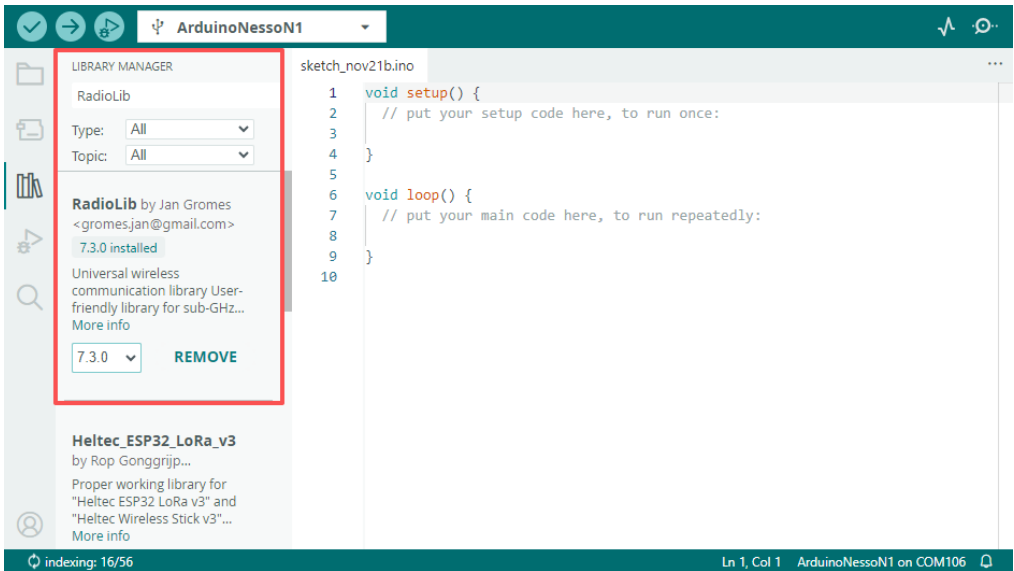
注意

Arduino Nesso N1 使用 SX1262 LoRa 通信芯片，开发时请注意选择相应的类。

案例程序

编译要求

- M5Stack 板管理版本 >= 3.2.5
- 开发板选项 = ArduinoNessoN1
- M5GFX 库版本 >= 0.2.17
- M5Unified 库版本 >= 0.2.11
- RadioLib 库版本 >= 7.3.0



发送端

```
1  #include "M5Unified.h"
2  #include <RadioLib.h>
3
4  #define LORA_MOSI_PIN 21
5  #define LORA_MISO_PIN 22
6  #define LORA_SCK_PIN 20
7  #define LORA_IRQ_PIN 15
8  #define LORA_CS_PIN 23
9  #define LORA_BUSY_PIN 19
10
11 // SX1262: NSS, DIO1, NRST, BUSY
12 SX1262 radio = new Module(LORA_CS_PIN, LORA_IRQ_PIN, RADIOLIB_NC, LORA_BUSY_PIN);
13
14 // save transmission state between loops
15 int transmissionState = RADIOLIB_ERR_NONE;
16
17 // flag to indicate that a packet was sent
18 volatile bool transmittedFlag = false;
19
20 // this function is called when a complete packet is transmitted by the module
21 // IMPORTANT: this function MUST be 'void' type and MUST NOT have any arguments!
22 #if defined(ESP8266) || defined(ESP32)
23 ICACHE_RAM_ATTR
24 #endif
25
26 void setFlag(void) {
27     // we sent a packet, set the flag
28     transmittedFlag = true;
29 }
30
31 M5Canvas canvas(&M5.Display);
32
33 void setup() {
34     M5.begin();
35     Serial.begin(115200);
36     M5.Display.setRotation(1);
37
38     // LED_BUILTIN at E1.P7
39     auto& ioe = M5.getIOExpander(0);
40
41     //LORA_RESET
42     ioe.digitalWrite(7, false);
43     delay(100);
44     ioe.digitalWrite(7, true);
45     delay(100);
46
47     ioe.digitalWrite(5, true); //LORA_LNA_ENABLE
48     ioe.digitalWrite(6, true); //LORA_ANTENNA_SWITCH
49
50     canvas.createSprite(M5.Display.width(), M5.Display.height());
```

```

51 canvas.setTextColor(GREEN);
52 canvas.setTextScroll(true);
53 // initialize SX1262
54 Serial.print(F("[SX1262] Initializing ... "));
55 canvas.println("[SX1262] Initializing ... ");
56 canvas.pushSprite(0, 0);
57
58 // frequency, bandwidth, spreading factor, coding rate, sync word, power, preamble length, TCXO r
59 // useRegulatorLDO
60 int state = radio.begin(868.0, 125.0f, 12, 5, 0x34, 22, 20, 3.0, true);
61 if (state == RADIOLIB_ERR_NONE) {
62     Serial.println(F("success!"));
63 } else {
64     Serial.print(F("failed, code "));
65     Serial.println(state);
66     while (true) {
67         delay(10);
68     }
69 }
70
71 // set the function that will be called when packet transmission is finished
72 radio.setPacketSentAction(setFlag);
73
74 // start transmitting the first packet
75 Serial.print(F("[SX1262] Sending first packet ... "));
76
77 // you can transmit C-string or Arduino string up to 256 characters long
78 transmissionState = radio.startTransmit("Hello world from M5Stack!");
79 }
80
81 // counter to keep track of transmitted packets
82 int count = 0;
83
84 void loop() {
85     // check if the previous transmission finished
86     if (transmittedFlag) {
87         // reset flag
88         transmittedFlag = false;
89
90         if (transmissionState == RADIOLIB_ERR_NONE) {
91             // packet was successfully sent
92             Serial.println(F("transmission finished!"));
93             canvas.println("OK!");
94             canvas.pushSprite(0, 0);
95             // NOTE: when using interrupt-driven transmit method, it is not possible to automatically mea
96             // transmission data rate using getDataRate()
97
98         } else {
99             Serial.print(F("failed, code "));
100             Serial.println(transmissionState);
101         }
102
103     // clean up after transmission is finished. This will ensure transmitter is disabled, RF switch

```

```
104 // etc.
105 radio.finishTransmit();
106
107 // wait a second before transmitting again
108 delay(1000);
109
110 // send another one
111 Serial.print(F("[SX1262] Sending another packet ... "));
112
113 // you can transmit C-string or Arduino string up to 256 characters long
114 String str = "Hello Arduino Nesso N1 #" + String(count++);
115 transmissionState = radio.startTransmit(str);
116 canvas.println("Send:" + str);
117 canvas.pushSprite(0, 0);
118 }
119 delay(100);
120 }
```

接收端


```
1  #include "M5Unified.h"
2  #include <RadioLib.h>
3
4  #define LORA_MOSI_PIN 21
5  #define LORA_MISO_PIN 22
6  #define LORA_SCK_PIN 20
7  #define LORA_IRQ_PIN 15
8  #define LORA_CS_PIN 23
9  #define LORA_BUSY_PIN 19
10
11 // SX1262: NSS, DI01, NRST, BUSY
12 SX1262 radio = new Module(LORA_CS_PIN, LORA_IRQ_PIN, RADIOLIB_NC, LORA_BUSY_PIN);
13
14 // flag to indicate that a packet was received
15 volatile bool receivedFlag = false;
16
17 // this function is called when a complete packet is received by the module
18 // IMPORTANT: this function MUST be 'void' type and MUST NOT have any arguments!
19 #if defined(ESP8266) || defined(ESP32)
20 ICACHE_RAM_ATTR
21 #endif
22
23 void setFlag(void) {
24     // we got a packet, set the flag
25     receivedFlag = true;
26 }
27
28 M5Canvas canvas(&M5.Display);
29
30 void setup() {
31     M5.begin();
32     Serial.begin(115200);
33     M5.Display.setRotation(1);
34
35     // LED_BUILTIN at E1.P7
36     auto& ioe = M5.getIOExpander(0);
37
38     //LORA_RESET
39     ioe.digitalWrite(7, false);
40     delay(100);
41     ioe.digitalWrite(7, true);
42     delay(100);
43
44     ioe.digitalWrite(5, true); //LORA_LNA_ENABLE
45     ioe.digitalWrite(6, true); //LORA_ANTENNA_SWITCH
46
47     canvas.createSprite(M5.Display.width(), M5.Display.height());
48     canvas.setTextColor(GREEN);
49     canvas.setTextScroll(true);
50     // initialize SX1262
```

```

51 Serial.print(F("[SX1262] Initializing ... "));
52 canvas.println("[SX1262] Initializing ... ");
53 canvas.pushSprite(0, 0);
54
55 // frequency, bandwidth, spreading factor, coding rate, sync word, power, preamble length, TCXO r
56 int state = radio.begin(868.0, 125.0f, 12, 5, 0x34, 22, 20, 3.0, true);
57 if (state == RADIOLIB_ERR_NONE) {
58     Serial.println(F("success!"));
59 } else {
60     Serial.print(F("failed, code "));
61     Serial.println(state);
62     while (true) { delay(10); }
63 }
64
65 // set the function that will be called when new packet is received
66 radio.setPacketReceivedAction(setFlag);
67
68 // start listening for LoRa packets
69 Serial.print(F("[SX1262] Starting to listen ... "));
70 state = radio.startReceive();
71 if (state == RADIOLIB_ERR_NONE) {
72     Serial.println(F("success!"));
73 } else {
74     Serial.print(F("failed, code "));
75     Serial.println(state);
76     while (true) { delay(10); }
77 }
78 }
79
80 void loop() {
81     // check if the flag is set
82     if (receivedFlag) {
83         // reset flag
84         receivedFlag = false;
85
86         // you can read received data as an Arduino String
87         String str;
88         int state = radio.readData(str);
89
90         if (state == RADIOLIB_ERR_NONE) {
91             // packet was successfully received
92             Serial.println(F("[SX1262] Received packet!"));
93
94             // print data of the packet
95             Serial.print(F("[SX1262] Data:\t\t"));
96             Serial.println(str);
97
98             canvas.print(F("[SX1262] Data:\t\t"));
99             canvas.println(str);
100
101             // print RSSI (Received Signal Strength Indicator)
102             Serial.print(F("[SX1262] RSSI:\t\t"));
103             Serial.print(radio.getRSSI());

```

```

104     Serial.println(F(" dBm"));
105
106     canvas.print(F("[SX1262] RSSI:\t\t"));
107     canvas.print(radio.getRSSI());
108     canvas.println(F(" dBm"));
109
110     // print SNR (Signal-to-Noise Ratio)
111     Serial.print(F("[SX1262] SNR:\t\t"));
112     Serial.print(radio.getSNR());
113     Serial.println(F(" dB"));
114
115     canvas.print(F("[SX1262] SNR:\t\t"));
116     canvas.print(radio.getSNR());
117     canvas.println(F(" dB"));
118
119     canvas.pushSprite(0, 0);
120
121     // print frequency error
122     Serial.print(F("[SX1262] Frequency error:\t"));
123     Serial.print(radio.getFrequencyError());
124     Serial.println(F(" Hz"));
125
126
127
128     } else if (state == RADIOLIB_ERR_CRC_MISMATCH) {
129         // packet was received, but is malformed
130         Serial.println(F("CRC error!"));
131
132     } else {
133         // some other error occurred
134         Serial.print(F("failed, code "));
135         Serial.println(state);
136     }
137 }
138 }

```

运行结果

将以上两段代码分别编译上传至两个 Arduino Nesso N1，发送端和接收端将分别显示发送和接收状态。



- [RadioLib Library - GitHub](#)
- [RadioLib Wiki - GitHub](#)
- [RadioLib Docs](#)
- [RadioLib SX1262 Docs](#)