

LoRaWAN スケッチ開発

KashiwaGeeks

山口知昭

Github : [ty4tw](#)

KashiwaGeeks（代表 原田貴義氏）

毎月第二土曜日 13:00-17:00 KOILでもくもく会開催

- 参加者は多くても10名
- 今年のもくもく会での学習内容

1. 配色の基礎知識
2. IoTはフロンティアか
3. MQTT-SNについて
4. ポインターについて理解する最後のチャンス
5. LoRaWAN仕様書の読み方
6. Arduinoの開発環境を一緒に調べてみよう
7. Arduinoを作ってみよう
8. OSについて語ろう

など

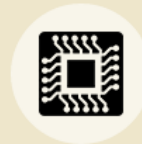


The Eclipse Paho project provides open-source client implementations of MQTT and MQTT-SN messaging protocols aimed at new, existing, and emerging applications for the Internet of Things (IoT).

[Download Now »](#)


For Constrained Networks

IoT systems need to deal with frequent network disruption and intermittent, slow, or poor quality networks. Minimal data costs are crucial on networks with millions and billions of connected devices.



Devices and Embedded Platforms

Devices and edge-of-network servers often have very limited processing resources available. Paho understands small footprint clients and corresponding server support.

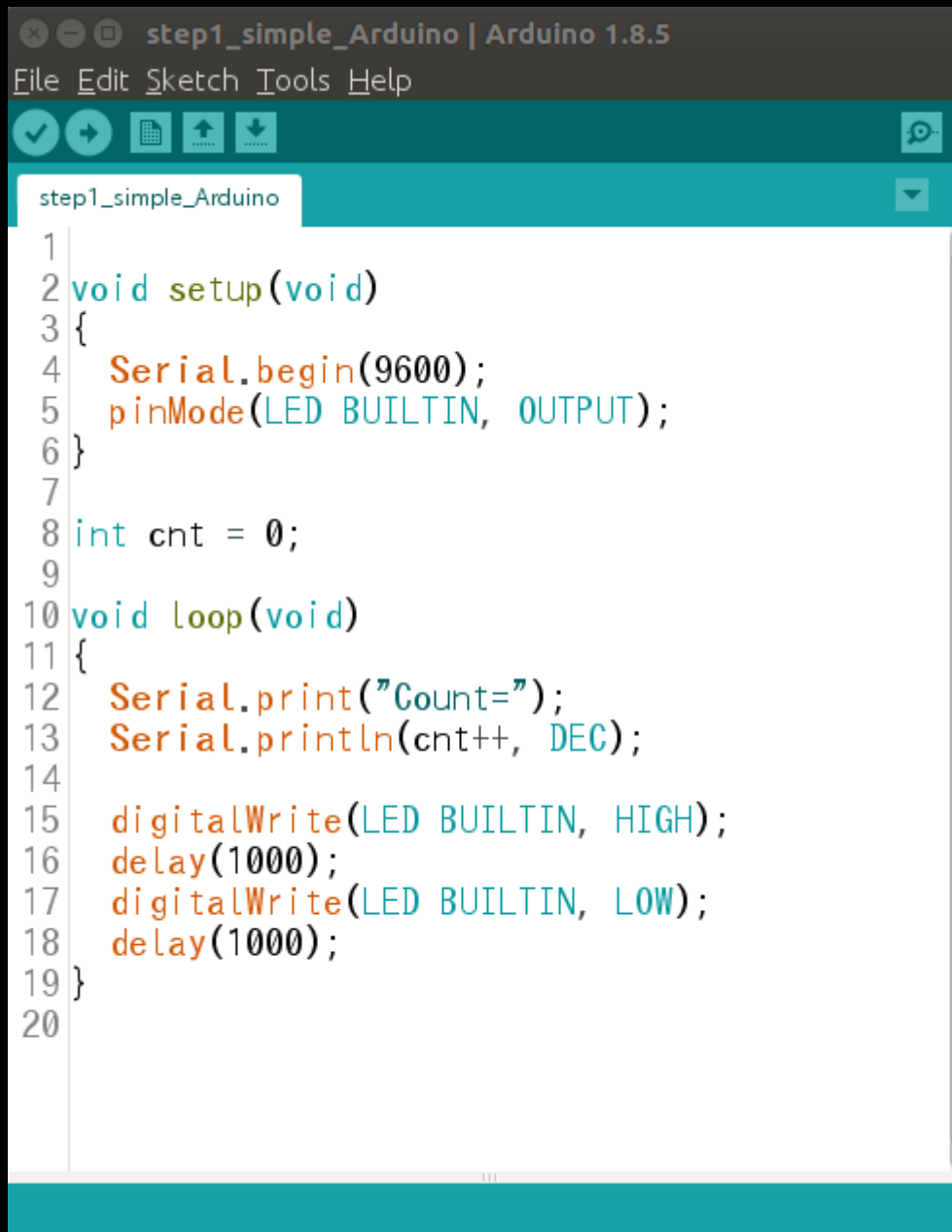


Reliable

Paho focuses on reliable implementations that will integrate with a wide range of middleware, programming and messaging models.



Arduinoスケッチはダサイ

A screenshot of the Arduino IDE interface. The title bar shows 'step1_simple_Arduino | Arduino 1.8.5'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for saving, running, uploading, and downloading. The main editor area shows a sketch named 'step1_simple_Arduino' with the following code:

```
1
2 void setup(void)
3 {
4   Serial.begin(9600);
5   pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 int cnt = 0;
9
10 void loop(void)
11 {
12   Serial.print("Count=");
13   Serial.println(cnt++, DEC);
14
15   digitalWrite(LED_BUILTIN, HIGH);
16   delay(1000);
17   digitalWrite(LED_BUILTIN, LOW);
18   delay(1000);
19 }
20
```

- ✓ Serial.print("Count=");
- ✓ Serial.println(cnt++, DEC);
- ✓ pinMode();
- ✓ digitalWrite();

ArduinoでLoRaWAN 大丈夫？

```
step2_simple_LoRaWAN | Arduino 1.8.5
File Edit Sketch Tools Help

step2_simple_LoRaWAN S
1 #define LoRa SEND INTERVAL 5000 // LoRa送信間隔（ミリ秒）
2
3 unsigned long beforetime = 0L;
4 float bme temp = 0;
5 float bme humi = 0;
6 float bme press = 0;
7
8
9 void setup() {
10   Serial.begin(9600);
11   Serial.println("LoRa TEMP/HUMI/PRESS Send for KiwiTech");
12
13   // ここでLoRaWANの初期設定
14 }
15
16
17 void loop() {
18   if (millis() - beforetime > LoRa SEND INTERVAL) {
19     sendTemp(); // ここでLoRaWANでデータを送る
20     beforetime = millis();
21   }
22 }
23
24
25 //
26 // 温度湿度気圧を LoRa送信する
27 //
28 void sendTemp()
29 {
30   sprintf(cmdline, "lorawan tx cnf %d %04x%04x%06lx", port, temp, humi, press);
31   if (!sendCmd2(cmdline, true, "tx ok", "rx", SERIAL WAIT TIME)) {
32     return false;
33   }
34   return true;
35 }
36
37
```

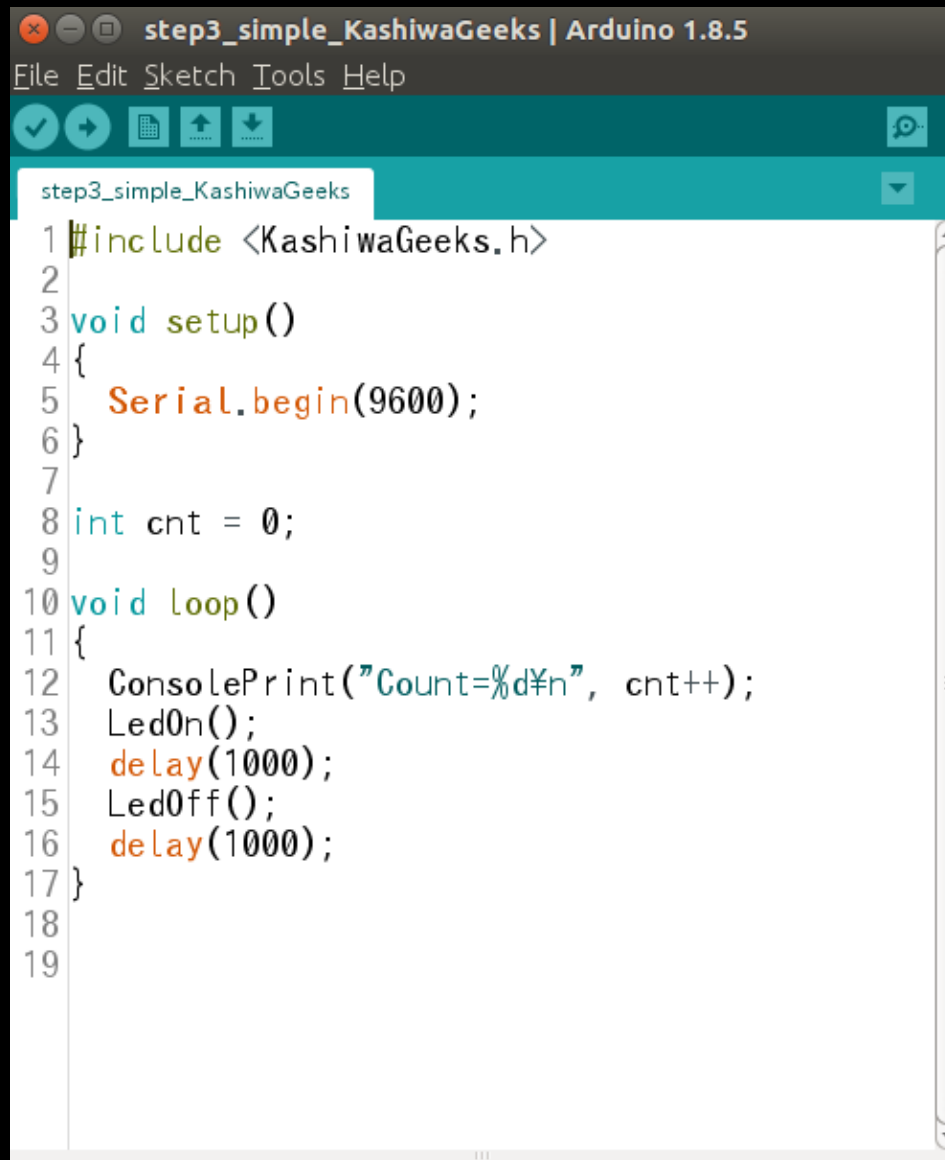
✓ loop()で実行の制御は見通しが悪い。

✓ ループしてるだけでバッテリー消費

✓ sprintf(buff, "lorawan tx cnf ", ...);
sendCmd();

KashiwaGeeks登場

#include <KashiwaGeeks.h>

A screenshot of the Arduino IDE interface. The title bar reads 'step3_simple_KashiwaGeeks | Arduino 1.8.5'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for saving, running, and other functions. The main text area shows the following code:

```
1 #include <KashiwaGeeks.h>
2
3 void setup()
4 {
5   Serial.begin(9600);
6 }
7
8 int cnt = 0;
9
10 void loop()
11 {
12   ConsolePrint("Count=%d\n", cnt++);
13   LedOn();
14   delay(1000);
15   LedOff();
16   delay(1000);
17 }
18
19
```

新コマンド

ConsolePrint("format", ...);
ConsolePrint(F("format"), ...);
LedOn()
LedOff()

がしかし、

1. loop()の見通しの悪さ
2. バッテリー消費

は改善されていない

心配御無用 三役登場

```
step4_sleep_power_save
1 #include <KashiwaGeeks.h>
2
3 void start(void)
4 {
5     Serial.begin(57600);
6     //DisableDebug();
7
8     /* seetup WDT interval to 1, 2, 4 or 8 seconds
9      * Default interval is 1 second. */
10    //setWDT(8);    // set to 8 seconds
11 }
12
13 int cnt = 0;
14
15 void func1()
16 {
17     ConsolePrint("Count=%d\r\n", cnt++);
18     DebugPrint(F("Debug Count=%d\r\n"), cnt++);
19     delay(1000);
20 }
21
22
23 //=====
24 //      Execution interval
25 //      TASK( function, interval by second )
26 //=====
27 TASK_LIST = {
28 TASK(func1, 0, 5),
29 //TASK(func2, 2, 4),
30 //TASK(func3, 3, 6),
31 END OF TASK_LIST
32 };
33
34
```

```
1 void start(void)
2 {
3     Serial.begin(57600);
4     //DisableDebug();
5
6     /* seetup WDT interval to 1, 2, 4 or 8 seconds
7      * Default interval is 1 second. */
8     //setWDT(8); // set to 8 seconds
9 }
```

```
TASK_LIST = {
TASK(func1, 0, 5),
//TASK(func2, 2, 4),
//TASK(func3, 3, 6),
END_OF_TASK_LIST
};
```

ここでクイズです。

残された課題は何？

答えは

```
sprintf( buff, "format", ... );
```

```
sendCmd();
```

Class ADB922S 登場

1.3. How to Use ADB922S Arduino Shield

ADB922S Arduino Shield equips with a USB-UART bridge and can be powered by USB. Figure 1 and following table depict this module and its connector.

1. USB Device Connector
2. Reset Button
3. UART TX/RX LED (D4, D5)
4. ADB922S
5. Antenna Connector

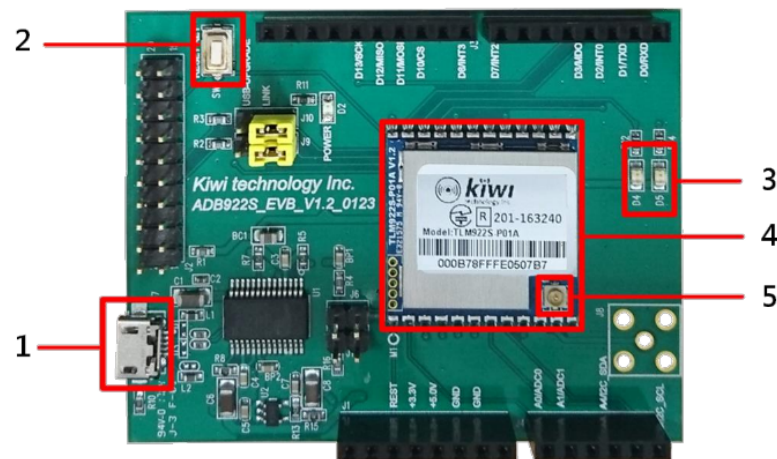


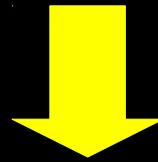
Figure 1 ADB922S Arduino Shield

```

step6_power_save_LoRaWAN S
35 #include <KashiwaGeeks.h>
36
37 ADB922S LoRa;
38
39 //=====
40 //          Initialize Device Function
41 //=====
42 #define BPS 9600          9600
43 #define BPS 19200        19200
44 #define BPS 57600        57600
45 #define BPS 115200       115200
46
47 void start()
48 {
49     /* Setup console */
50     Serial.begin(BPS 57600);
51     //DisableConsole();
52     //DisableDebug();
53
54     ConsolePrint(F("**** Start*****\n"));
55
56     /* setup ADB922S */
57     if ( LoRa.begin(BPS 9600) == false )
58     {
59         while(true)
60         {
61             LedOn();
62             delay(300);
63             LedOff();
64             delay(300);
65         }
66     }
67
68     /* join LoRaWAN */
69     LoRa.connect();
70
71
72     /* for BME280 initialize */
73     //bme.begin();
74
75     /* setup WDT interval to 1, 2, 4 or 8 seconds */
76     //setWDT(8);    // set to 8 seconds
77 }
78

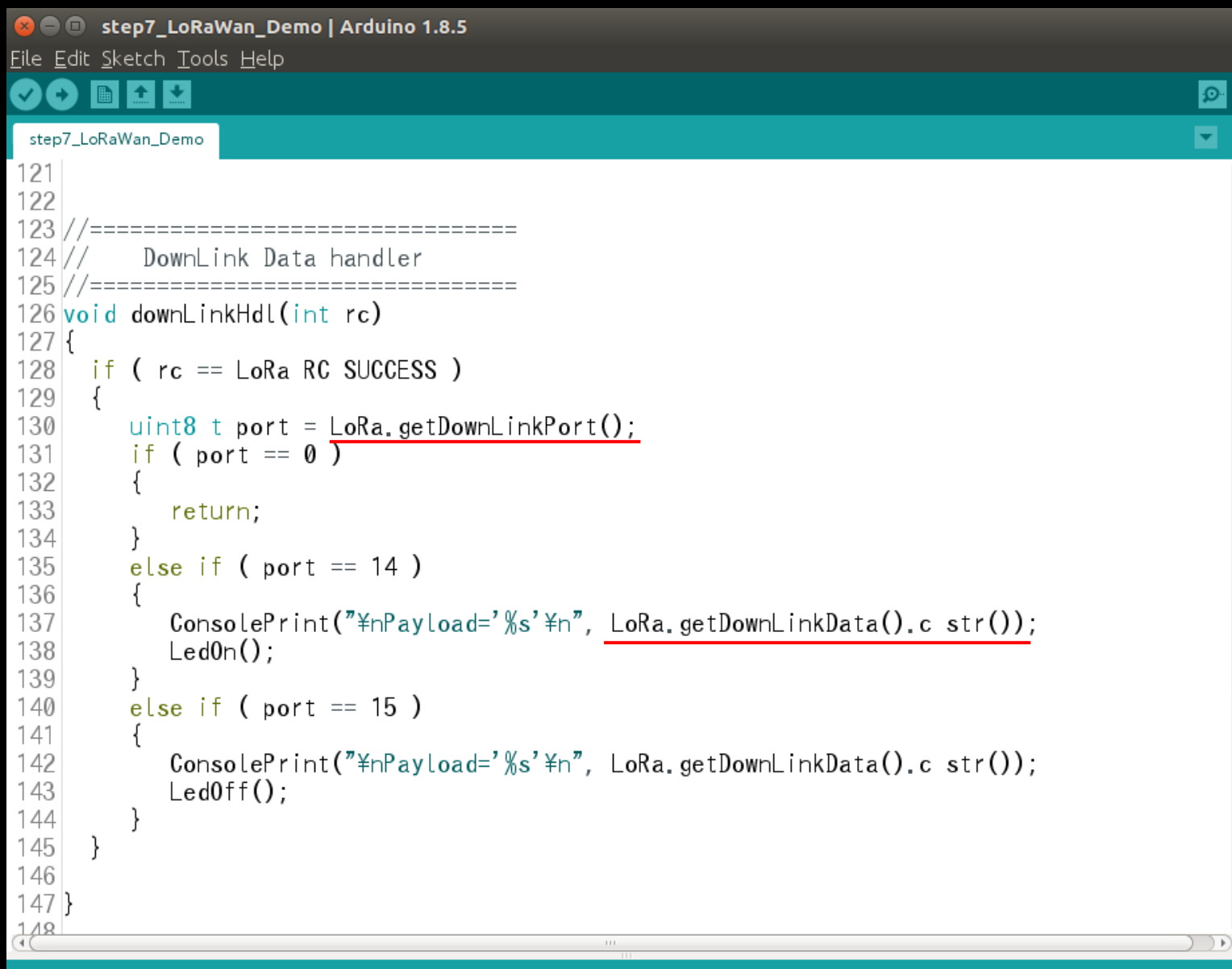
```

```
257
258 sprintf(cmdline, "lorawan tx ucnf %d %04x%04x%06lx", port, temp, humi, press);
259 sendCmd2(cmdline, true, "tx_ok", "rx", SERIAL_WAIT_TIME);
260
261 |sprintf(cmdline, "lorawan tx cnf %d %04x%04x%06lx", port, temp, humi, press);
262 sendCmd2(cmdline, true, "tx_ok", "rx", SERIAL_WAIT_TIME);
263
```



```
95|
96 // LoRaWANでデータを送信する
97 LoRa.sendData(port, true, F("%04x%04x%06lx"), temp, humi, press);
98 LoRa.sendDataConfirm(port, true, F("%04x%04x%06lx"), temp, humi, press);
99}
100
101 /* End of Program */
```

Down Linkデータも扱えます



```
121
122
123 //=====
124 //   DownLink Data handler
125 //=====
126 void downLinkHdl(int rc)
127 {
128   if ( rc == LoRa RC SUCCESS )
129   {
130     uint8 t port = LoRa.getDownLinkPort();
131     if ( port == 0 )
132     {
133       return;
134     }
135     else if ( port == 14 )
136     {
137       ConsolePrint("Payload='%s'", LoRa.getDownLinkData().c_str(););
138       LedOn();
139     }
140     else if ( port == 15 )
141     {
142       ConsolePrint("Payload='%s'", LoRa.getDownLinkData().c_str());
143       LedOff();
144     }
145   }
146 }
147
148
```

これも簡単にしました。
LoRa.checkDownLink();

```
step8_LoRaWan_Demo
100 //=====
101 //   DownLink Data handler
102 //=====
103 void port14(void)
104 {
105   ConsolePrint("%s\r\n", LoRa.getDownLinkData().c_str());
106   LedOn();
107 }
108
109 void port15(void)
110 {
111   ConsolePrint("%s\r\n", LoRa.getDownLinkData().c_str());
112   LedOff();
113 }
114
115 PORT LIST = {
116   PORT(14, port14), // port & callback
117   PORT(15, port15),
118   END OF PORT LIST
119 };
120
```

Class ADB922Sのメソッド

- くわしくはマニュアルで。
- ライブラリとマニュアルは
<https://github.com/ty4tw/KashiwaGeeks>

END