

ライブラリで効率的な Leafony開発

慶應義塾大学

中川 修哉

Leafony ライブラリを公開しました🎉

<https://github.com/Leafony/LeafonySTM32>

The screenshot shows the GitHub repository page for **Leafony / LeafonySTM32**. The repository is public and has 3 unwatchers, 0 forks, and 0 stars. The main branch is **main** with 1 branch and 3 tags. The repository contains 20 commits, with the latest commit **c837498** on April 22. The repository is an Arduino Library for Leafony STM32, with documentation at docs.leafony.com/. The repository includes a README.md file, a LICENSE file, and a keyword.txt file. The repository also has a **Spell Check** badge showing **passing**.

File	Commit	Time
.github/workflows	Update spell-check.yml	9 months ago
docs	first commit	9 months ago
examples/Basics	Support accel	4 months ago
src	Support accel	4 months ago
tools	first commit	9 months ago
LICENSE	first commit	9 months ago
README.md	typo	4 months ago
keyword.txt	first commit	9 months ago
library.json	first commit	9 months ago
library.properties	version update	6 months ago

LeafonySTM32

Spell Check **passing**

About

Arduino Library for Leafony STM32

docs.leafony.com/

arduino **leafony**

Readme

MIT license

0 stars

3 watching

0 forks

Releases 3

v0.0.3 **Latest**

on Apr 22

[+ 2 releases](#)

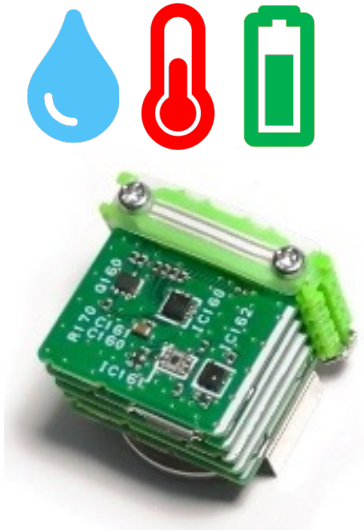
Packages

No packages published

[Publish your first package](#)

Contributors 2

多くのサンプルスケッチが登場



環境センシング



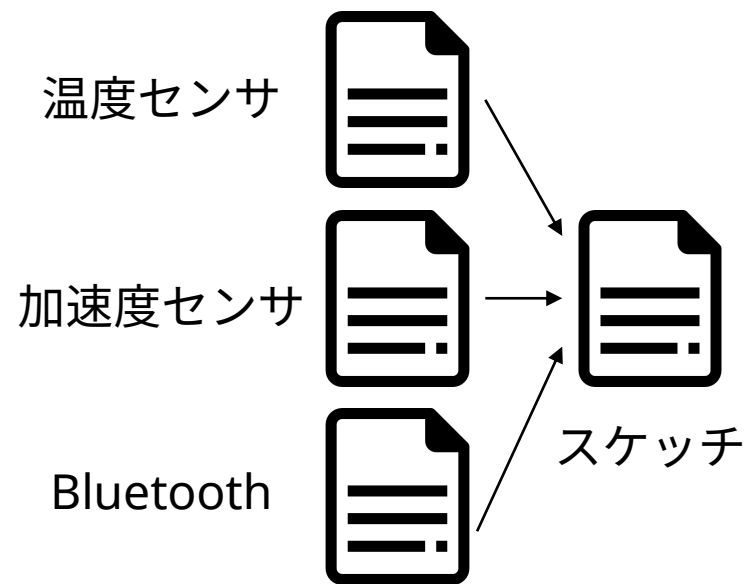
Bluetooth通信



イベントが発生したとき
だけ動作するシステム

問題点

- ICごとにライブラリが必要
- リーフが増えると読み込むライブラリ数も増大
- コードが長く読みにくくなる



他社製品の場合

- M5 Stack
 - 専用のライブラリが存在
 - BluetoothやWi-Fiなどの複雑な記述もライブラリで楽々
 - 記述方法が決まるため、安定した動作が可能

Leafony ライブラリを公開しました🎉

<https://github.com/Leafony/LeafonySTM32>

The screenshot shows the GitHub repository page for **Leafony / LeafonySTM32**. The repository is public and has 3 unwatchers, 0 forks, and 0 stars. The main branch is **main** with 1 branch and 3 tags. The repository contains 20 commits, with the latest commit **c837498** on April 22. The repository is an Arduino Library for Leafony STM32, with documentation at docs.leafony.com/. It includes a README, MIT license, 0 stars, 3 watchers, and 0 forks. There are 3 releases, with the latest being **v0.0.3** on April 22. The repository also has 2 contributors and no published packages.

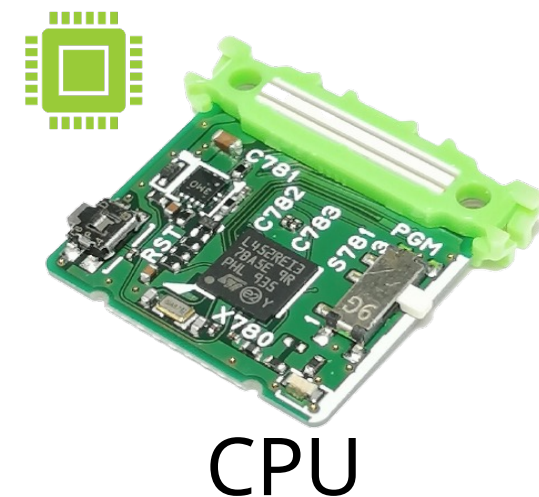
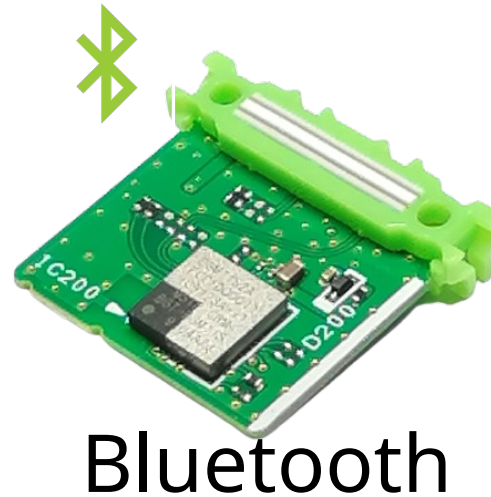
File	Commit	Time
.github/workflows	Update spell-check.yml	9 months ago
docs	first commit	9 months ago
examples/Basics	Support accel	4 months ago
src	Support accel	4 months ago
tools	first commit	9 months ago
LICENSE	first commit	9 months ago
README.md	typo	4 months ago
keyword.txt	first commit	9 months ago
library.json	first commit	9 months ago
library.properties	version update	6 months ago

LeafonySTM32

Spell Check: **passing**

Leafonyライブラリのコンセプト

もっと簡単に省電力に



リーフごとに電力を最適化
ライブラリを**使うだけで省電力に**

省電力化するには...

➤ 通信

- ❖ BluetoothのBeaconを用いたデータ通信
- ❖ 送信中だけスリープから復帰

➤ センサ

- ❖ サンプル周波数を調整
- ❖ 不要なタイミングではスリープに入れる

➤ CPU

- ❖ 動作周波数をコントロールして電力を下げる
- ❖ 複数のスリープモードを活用

細かな調整が
電力に大きく影響

センサリーフの例

- デフォルト設定を省電力モードに
- センサ取得後は即座にスリープするように

leafony.sensors.init() :
 センサを省電力モードに設定

leafony.sensors.getHumid() :
 湿度センサを読み出して即スリープ

```
1  #include <LeafonySTM32.h>
2
3  static LeafonySTM32 leafony;
4
5  void setup() {
6      leafony.begin();
7      leafony.sensors.init();
8  }
9
10 void loop() {
11     Serial.println(leafony.sensors.getHumid());
12     delay(1000);
13 }
```

You, 2 months ago • first commit

コードの比較

省電力化の
ための設定

```
9 //
10 #include <Wire.h>
11 #include <ClosedCube_OPT3001.h>
12
13 #define OPT3001_ADDRESS 0x45 // ADDR pin = VCC
14
15 ClosedCube_OPT3001 illum;
16
17 float dataIllum;
18
19 void setup() {
20     // initialize serial communication at 115200 second per second:
21     Serial.begin(115200);
22     // initialize i2c communication with OPT3001:
23     Wire.begin();
24
25     delay(10);
26
27     OPT3001_Config illumConfig;
28     OPT3001_ErrorCode illumErrorConfig;
29
30     illum.begin(OPT3001_ADDRESS);
31
32     illumConfig.RangeNumber = B1100; // automatic full scale
33     illumConfig.ConversionTime = B1; // conversion time = 800ms
34     illumConfig.ModeOfConversionOperation = B11; // continous conversion
35     illumConfig.Latch = B0; // hysteresis-style
36
37     illumErrorConfig = illum.writeConfig(illumConfig);
38
39     if(illumErrorConfig != NO_ERROR){
40         illumErrorConfig = illum.writeConfig(illumConfig); //retry
41     }
42 }
43
44 void loop() {
45     OPT3001 result = illum.readResult();
46
47     if(result.error == NO_ERROR){
48         dataIllum = result.lux;
49     }
50
51     Serial.println("Lum[lx] = " + String(dataIllum));
52
53     delay(1000);
54
55 }
```

Before

```
1 #include <LeafonySTM32.h>
2
3 static LeafonySTM32 leafony;
4
5 void setup() {
6     leafony.begin();
7     leafony.sensors.init();
8 }
9
10 void loop() {
11     Serial.println(leafony.sensors.getHumid());
12     delay(1000);
13 }
```

After

少ない記述で省電力化が可能に

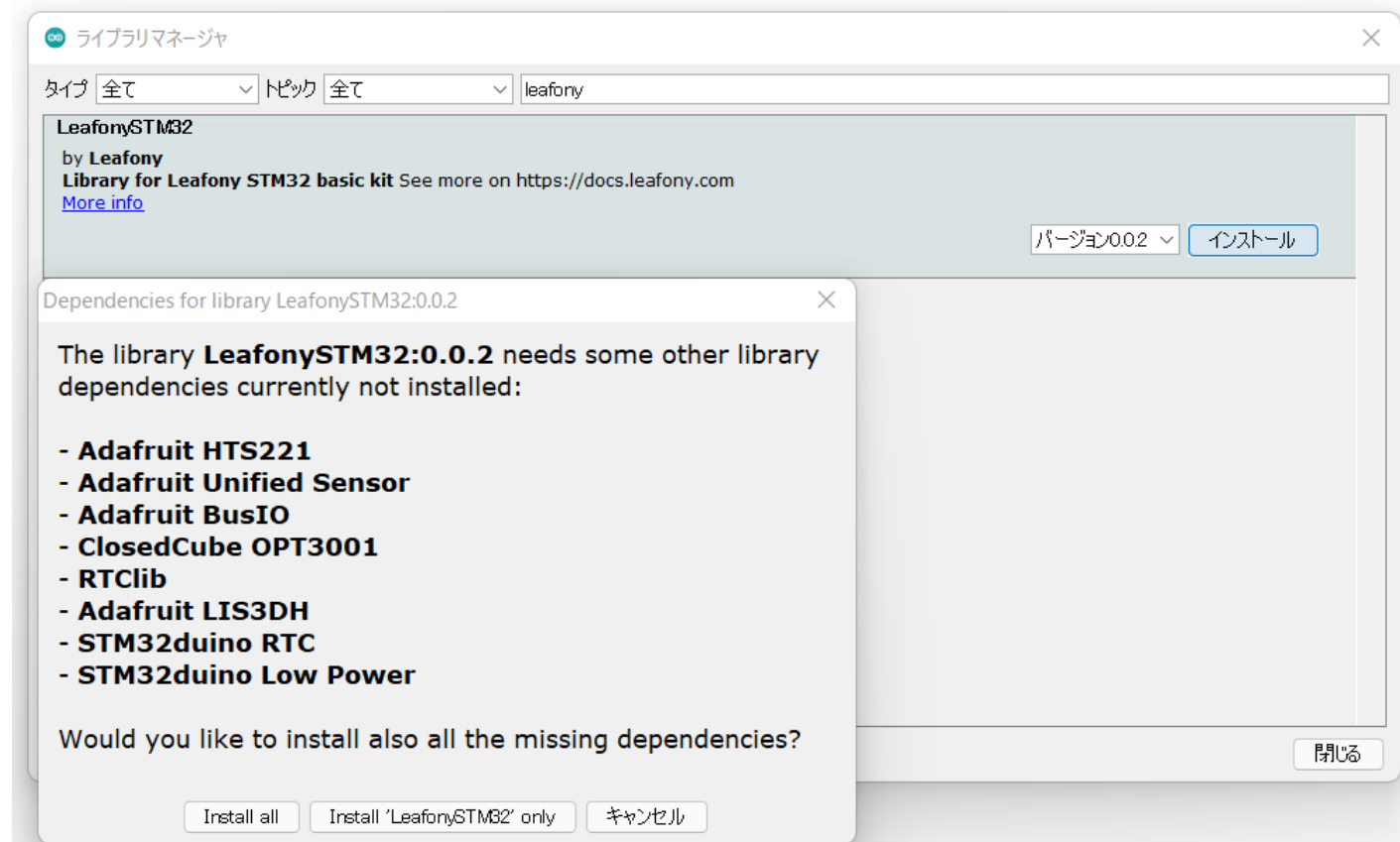
ライブラリのインストールが簡単に

Arduinoのライブラリマネージャで

「Leafony」で検索



関連するライブラリが全て
インストールされる





電力デバックスの簡易化

電力のデバッグをもっと簡単に

- 電力を下げるためには電流測定が必須
 - テスターを用意して、電流測定リーフを使う → 面倒
- ライブラリの関数を使うごとに消費電流が表示される仕組み

```
// 温度を測定
temp = leafony.sensors.getTemperature();

// センサをスリープ
leafony.sensors.sleepAllSensors();
```



```
Temperature: 24.6℃

----- Predicted Current -----
CPU: 4.5mA
Bluetooth: 3.1mA
Sensors: 0.015mA
Total: 7.6mA
-----

All sensors are going to sleep.

----- Predicted Current -----
CPU: 4.5mA
Bluetooth: 3.1mA
Sensors: 0.002mA
Total: 7.6mA
-----
```


まとめ

省電力化に特化したLeafonyのライブラリを公開

- 使うだけで省電力化が可能
- 電力のデバッグがより簡単に
- ライブラリのインストールが簡単に

```
Temperature: 24.6°C  
  
----- Predicted Current -----  
CPU: 4.5mA  
Bluetooth: 3.1mA  
Sensors: 0.015mA  
Total: 7.6mA  
-----  
  
All sensors are going to sleep.  
  
----- Predicted Current -----  
CPU: 4.5mA  
Bluetooth: 3.1mA  
Sensors: 0.002mA  
Total: 7.6mA  
-----
```

コンソール画面で
電力デバッグが可能