

# アルドウイーノ シーエルアイ ブイエス コード かいてきかいはつ **Arduino CLI + VS CODEで快適開発**

アルドウイーノ シーエルアイ ラッパー フォーブイエス コード  
Arduino CLI Wrapper for VS CODEで  
アルドウイーノ かいはつ  
「Arduino の開発をもっとカンタンに！」



Lang-ship  
たなかまさゆき

# アルドゥイーノ Arduinoをプログラムするには？

アルドゥイーノ アイディーエイ

- Arduino IDE

プラットホームアイオー

- PlatformIO

ウォクウィ

- Wokwi

アルドゥイーノ シーエルアイ

- Arduino CLI

じょうきいがい

上記以外にいろいろありますが、

アルドゥイーノ アイディーエイ プラットホームアイオー りょう

Arduino IDE かPlatformIOを利用

することが多いです。

# アルドウイーノ アイディーエイ Arduino IDE

- いちばんひょうじゅんてき かいはつかんきょう とうごうかいはつかんきょう  
一番 標 準 的な開発環 境であり、オールインワンの統合開発環 境
- ジーユーアイ りょう せってい かのう  
GUI を利用してボード設定などが可能
- プラットホームやライブラリに関して複数のバージョンを共 有する  
ことができる
- 少し古いプロジェクトを久しぶりにコンパイルすると  
いろいろバージョンアップしていく動かないことが起こりがち！

# プラットホームアイオー PlatformIO

- 組み込み環境用の統合開発環境
- プラットホームやライブラリに関するバージョン指定が可能
- ボード定義がなかつたり、GUIで変更できない
- 公式 ESP32 for Arduinoが2系で更新停止
- 3系以降は pioarduino を利用
- 新製品の M5Stack ボードを動かすのが大変

# ウォクワイ **Wokwi**

- 実機がなくてもブラウザ上でエミュレーションして動作
- Rustも利用可能
- 使えるボードが少ない
- **M5Unified のビルドがタイムアウトになりがち**

# アルドウイーノ シーエルアイ Arduino CLI

- アルドウイーノ アイディーエイ ないよう うご  
• Arduino IDE の内部で動いているすごいやつ
- ジーアールピーシー れんけいかのう  
• コマンドラインか gRPC で連携可能
- かん してい かのう  
• プラットホームやライブラリに関してバージョン指定が可能
- せってい せいび むずか  
• 設定ファイルの整備が難しい

# かくちょうきのうつく なので拡張機能を作ってみた

The screenshot shows the Arduino CLI Wrapper extension page on the Visual Studio Marketplace. The extension icon is a green microcontroller chip. The title is "Arduino CLI Wrapper" by TANAKA Masayuki, with 182 installs and 5 stars. A description says "Run Arduino CLI commands from VS Code". Below are buttons for "無効にする" (Disable), "アンインストール" (Uninstall), and "自動更新" (Auto Update). Navigation tabs include "詳細", "機能", "変更ログ", and "拡張機能パック". The main content area is titled "Arduino CLI Wrapper (VS Code Extension)" and includes a "日本語READMEはこちら" link. It describes the extension as running Arduino CLI commands from the command palette, status bar, and Explorer view, streaming logs to a pseudo terminal, supporting sketch.yaml profiles, and updating IntelliSense includePath during builds. On the right, there's an "インストール" section with details: 識別子: tanakamasayuki.vscode-arduino-cli-wrapper, バージョン: 1.5.0, 最終更新日: 2025-10-02, 時: 12:50:10, サイズ: 949.98KB. Below is a "Marketplace" section showing the item was published on 2025-09-10 at 15:14:50 and last released on 2025-10-02 at 12:05:44.

<https://marketplace.visualstudio.com/items?itemName=tanakamasayuki.vscode-arduino-cli-wrapper>

# おも きのう 主な機能

アルドウイーノ アイディィーエイ  
**(Arduino IDE にある機能)**

コンパイル/アップロード/シリアルモニター/デバッグ

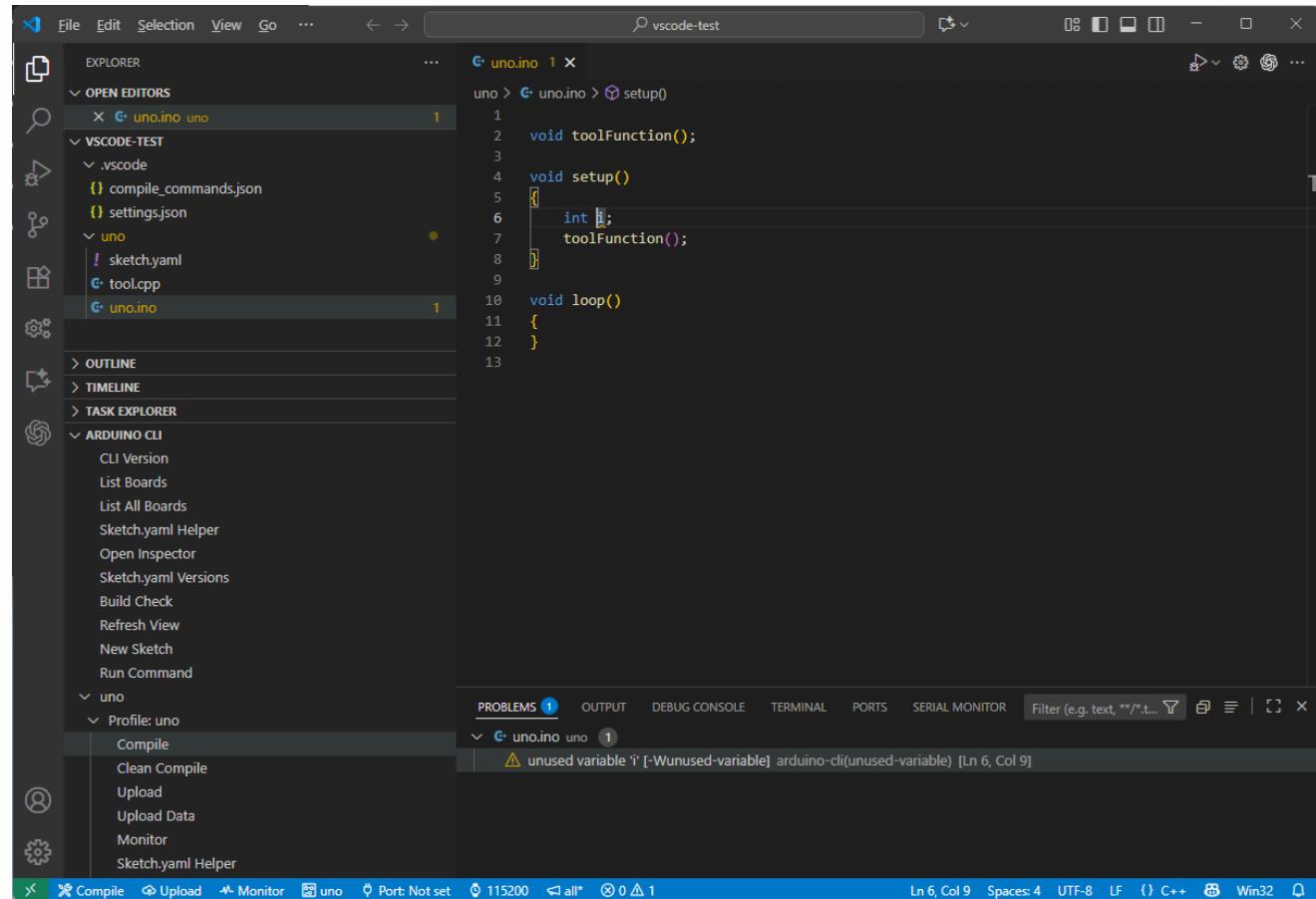
プラットホームアイオー  
**(PlatformIO にある機能)**

バージョン指定/複数ボード対応/インスペクター/data転送

どくじきのう  
**(独自機能)**

設定ファイル補助/スケッチ例検索/ WSL 連携/Wokwi連携

# きのうしようかい ブイエス コード れんけい 機能紹介 - VS CODEとの連携



シーシープラスプラスかくちょうきのう れんけい  
C/C++ 拡張機能と連携  
インテリセンス  
し、IntelliSenseやビルド後  
ひょうじ  
のエラー表示などいまどきの  
ひょうじゅんてき とうごうかいはつかんきょう  
標準的な統合開発環境を  
じつけん  
実現。  
ワーニング オール  
デフォルトWarningは all  
がい  
で、ワークスペース外のファ  
イルは無視します。

# きのうしようかい ブイエス コード れんけい 機能紹介 - VS CODEとの連携 2

きほんてき 基本的なエディタとしての機能もArduino IDE よりかなり進化。

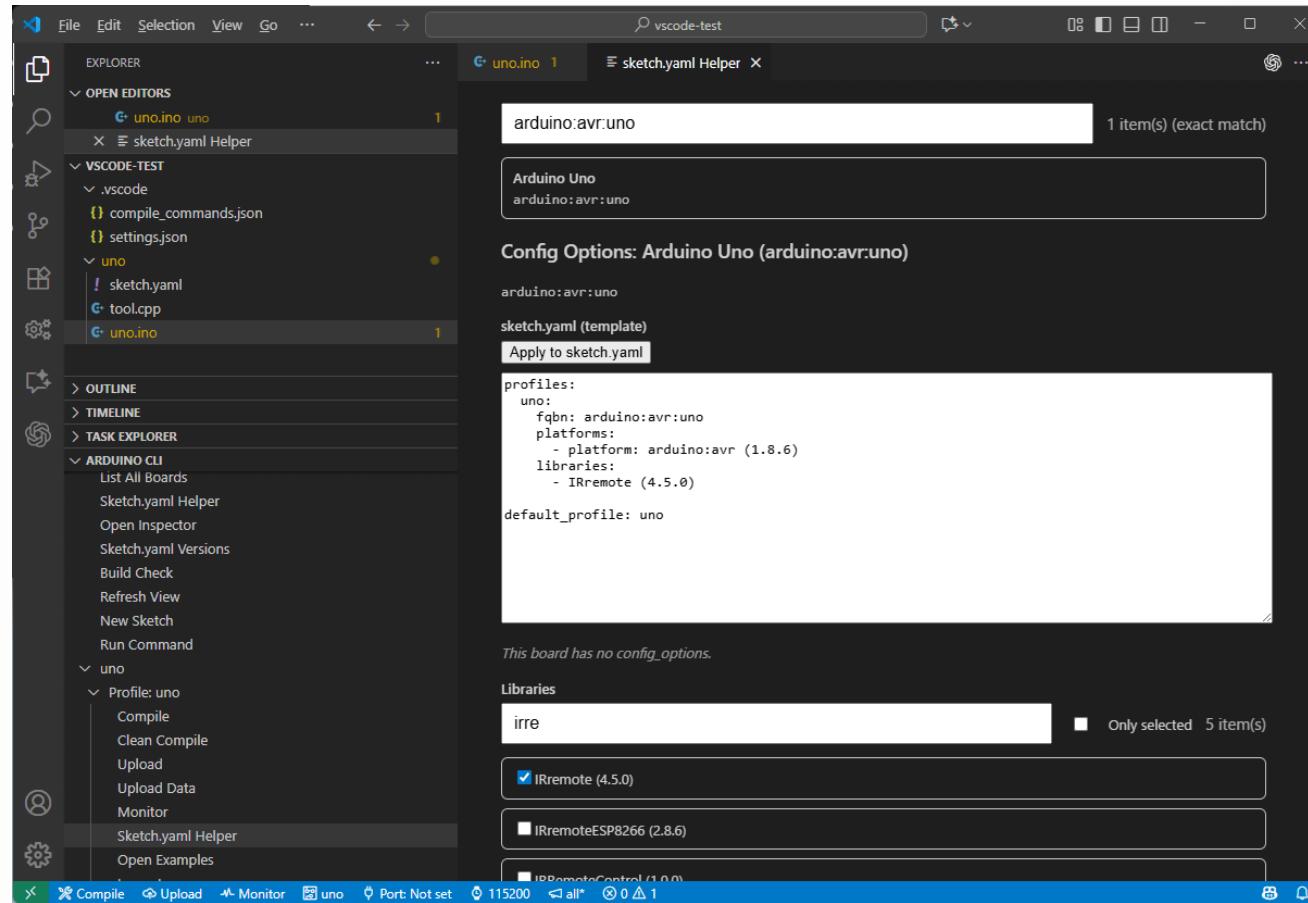
- ギットれんけい Git連携
- エーアイれんけい AI 連携
- ふくすう どうじ かんり 複数のプロジェクトを同時に管理

# きのうしようかい スケッチ ヤムル 機能紹介 - sketch.yaml

```
profiles:  
  m5stack_atom:  
    fqbn: esp32:esp32:m5stack_atom:UploadSpeed=750000,DebugLevel=verbose  
    platforms:  
      - platform: esp32:esp32 (3.3.1)  
        platform_index_url: https://espressif.github.io/arduino-esp32/package_esp32_index.json  
    libraries:  
      - M5Unified (0.2.8)  
      - M5GFX (0.2.14)
```

設定ファイルでボードの設定から、プラットホームのバージョン  
指定、利用するライブラリのバージョン指定まで可能です。  
ただこれを手で管理するのは無理！

# きのうしようかい せってい 機能紹介 - 設定ヘルパー



ヘルパー画面からプラットホームとボード、ボード定義とライブラリの指定が可能。  
現在指定しているバージョンと最新バージョンの差分チェック機能も搭載。

# きのうしようかい 機能紹介 - スケッチ例検索

```

/*
 * BlinkRGB
 *
 * Demonstrates usage of onboard RGB LED on some ESP dev boards.
 *
 * Calling digitalWrite(RGB_BUILTIN, HIGH) will use hidden RGB driver.
 *
 * RGBLedWrite demonstrates control of each channel:
 * void rgbLedWrite(uint8_t pin, uint8_t red_val, uint8_t green_val, uint8_t blue_val)
 *
 * WARNING: After using digitalWrite to drive RGB LED it will be impossible to drive the same pin
 * with normal HIGH/LOW level
 */
//#define RGB_BRIGHTNESS 64 // Change white brightness (max 255)

// the setup function runs once when you press reset or power the board

void setup() {
    // No need to initialize the RGB LED
}

// the Loop function runs over and over again forever
void loop() {
    #ifdef RGB_BUILTIN
        digitalWrite(RGB_BUILTIN, HIGH); // Turn the RGB LED white
        delay(1000);
        digitalWrite(RGB_BUILTIN, LOW); // Turn the RGB LED off
        delay(1000);

        rgbleWrite(RGB_BUILTIN, RGB_BRIGHTNESS, 0, 0); // Red
        delay(1000);
        rgbleWrite(RGB_BUILTIN, 0, RGB_BRIGHTNESS, 0); // Green
        delay(1000);
        rgbleWrite(RGB_BUILTIN, 0, 0, RGB_BRIGHTNESS); // Blue
        delay(1000);
        rgbleWrite(RGB_BUILTIN, 0, 0, 0); // Off / black
        delay(1000);
    #endif
}

```

**Copy to Clipboard** **Copy to project /examples**

File Edit Selection View Go ... ← → ⌘ vscodeArduinoCli

Arduin Examples × Filter (folder/path/filename) rgb

ESP32 : AnalogOut/ledWrite\_RGB/ledWrite\_RGB.ino  
c:\Users\ms.tanaka\AppData\Local\Arduino15\internal\esp32

ESP32 : Camera/CameraWebServer/CameraWebServer.ino  
c:\Users\ms.tanaka\AppData\Local\Arduino15\internal\esp32

ESP32 : GPIO/BlinkRGB/BlinkRGB.ino  
c:\Users\ms.tanaka\AppData\Local\Arduino15\internal\esp32

ESP32 : RMT/RMTDriver\_Compatible/Legacy\_RMT\_Driver.ino  
c:\Users\ms.tanaka\AppData\Local\Arduino15\internal\esp32

ESP32 : RMT/RMTWrite\_RGB\_LED/RMTWrite\_RGB\_LED.ino  
c:\Users\ms.tanaka\AppData\Local\Arduino15\internal\esp32

Matter : MatterColorLight/MatterColorLight.ino  
c:\Users\ms.tanaka\AppData\Local\Arduino15\internal\esp32

Matter : MatterContactSensor/MatterContactSensor.ino  
c:\Users\ms.tanaka\AppData\Local\Arduino15\internal\esp32

Matter : MatterDimmableLight/MatterDimmableLight.ino  
c:\Users\ms.tanaka\AppData\Local\Arduino15\internal\esp32

Matter : MatterEnhancedColorLight/MatterEnhancedColorLight.ino  
c:\Users\ms.tanaka\AppData\Local\Arduino15\internal\esp32

Matter : MatterFan/MatterFan.ino  
c:\Users\ms.tanaka\AppData\Local\Arduino15\internal\esp32

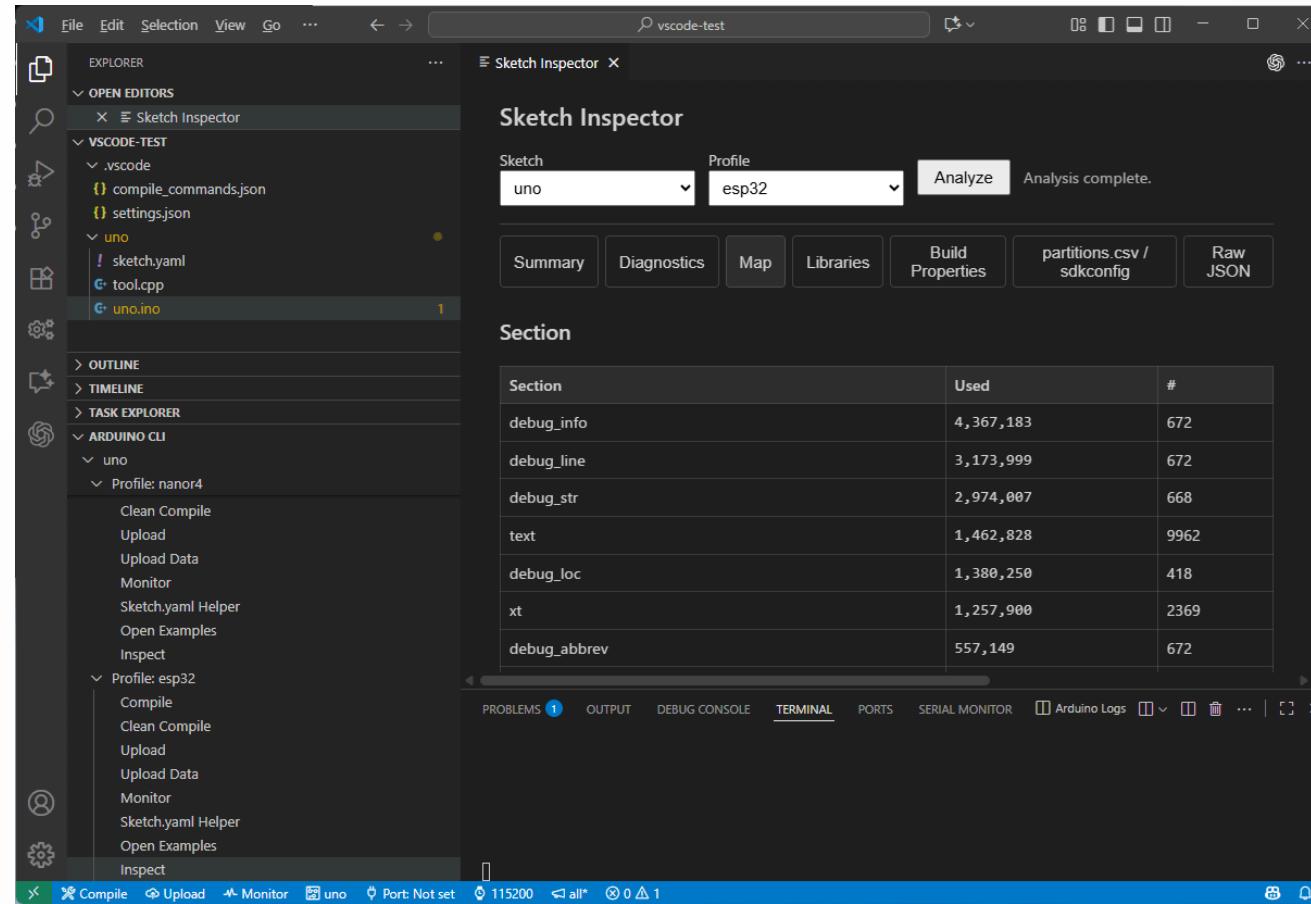
Matter : MatterOnIdentify/MatterOnIdentify.ino  
c:\Users\ms.tanaka\AppData\Local\Arduino15\internal\esp32

Matter : MatterTemperatureLight/MatterTemperatureLight.ino  
c:\Users\ms.tanaka\AppData\Local\Arduino15\internal\esp32

main Compile Upload Monitor m5stack\_atom Port: Not set 115200 all 0 0 0

めいなかふく  
ファイル名や中に含まれてい  
もじれつしほりこみかのう  
る文字列で絞り込みが可能な  
れいけんさくがめん  
スケッチ例検索画面です。フ  
めい  
ファイル名をクリックすると  
なかみ  
中身がプレビューするのでそ  
ばかくにんかのう  
の場で確認可能です。**特定の**  
**かんすうよびだし**  
**関数を呼び出しているスケッ**  
**チ例だけを確認可能！**

# きのうしょうかい 機能紹介 - インスペクター



じ  
ビルド時のパラメーターの  
かくにん マップ  
確認やMAPファイルの簡易  
ぶんせき かのう がめん  
分析が可能な画面です。

# きのうしようかい データ転送 機能紹介 - data転送

スケッチのあるフォルダに **データ** フォルダを作成し、そこに **転送**したい  
ファイルを入れておくことで **mkspiffs** または **mklittlefs** でイメージを  
生成し、**esptool** で **SPIFFS** パーティションに書き込みます。

アルドウイーノ アイディーエイ  
Arduino IDE のバージョンアップで利用できなくなった機能を実装  
しました。

# きのうしょうかい ダブルエスエルれんけい 機能紹介 - WSL 連携

ウインドウズじょう  
Windows上でコンパイルは非常に遅いです。そこで WSL 上でコ  
ンパイルして、Windows上でシリアルポートに転送する仕組みを作り  
ました。

データ  
デバッグやdataフォルダの転送はできませんが、通常の作業は WS  
L上で快適にコンパイルして転送まで可能です。

ありがとう  
**Thank You**



たなかまさゆき

<https://lang-ship.com/>