

STEM Duコントローラ
RDC-ESP32 TYPE 1・2・3
プレイステーション4
無線コントローラの接続

九州国際大学
水井研究室

事前準備

STEM Duコントローラ RDC-ESP32R2 開発環境のインストール



The screenshot shows the JAPAN ROBOTECH website. The main navigation bar includes links for Home, RDC, and various programming resources. The content area is titled "1. プログラミング環境をインストールする" (1. Install the programming environment). It provides instructions on where to find the latest installation guide and includes a link to "プログラミング環境のインストール" (Install the programming environment). Below this, it says "2. パソコンとRDCを接続する" (2. Connect the PC and RDC) and provides instructions on using a micro-USB cable. An image of the RDC-ESP32R2 board is shown at the bottom left.



The screenshot shows the STEM Education Research Center website. The header includes the center's name in English and Japanese, along with a list of fields: Engineering, Mathematics, Technology, Science, and Art. The main content area is titled "STEM Duコントローラのプログラミング環境" (STEM Du controller programming environment). It explains that two preparations are needed for programming and provides a list of updates. The updates include: 2022/5/7 macOS Monterey (12.3.x) support, 2022/1/2 Arduino 1.8.19 compatibility, 2022/1/2 STEM Du library changes, and 2021/6/10 board manager support.

参考:

JAPAN ROBOTECH webサイト(プログラミング環境をインストール)

<https://sites.google.com/site/japanrobotech2/start>

埼玉大学 教育学部 野村研究室(RDC-ESP32のボードマネージャをインストール)

http://neo.stem-edulab.org/page_20200820030156/page_20201016230943

PS4/3コントローラを準備



DualShock4

- ・アナログ入力:6軸
- ・スイッチ入力:16
- ・6軸加速度センサ:1個
- ・サウンド出力:1
- ・振動モータ出力:1
- ・フルカラーLED表示

PS4コントローラ (DualShock4)
Bluetooth接続



PS3コントローラ (DualShock3)
Bluetooth接続

市販ゲームコントローラは、アナログ入力や多くのボタン(CH)入力に対応し、耐久性が高く入手しやすい！

作業の流れ

目的: PS4/3コントローラ(DualShock4/3) をBluetooth接続

1. ESP32のBluetooth MACアドレスを調査
(コントローラがBluetooth経由でRDC本体を探すため)
2. MACアドレスをコントローラへ書込み
(コントローラへRDC本体のMACアドレスを書込む)
3. PS4-esp32ライブラリをArduino IDEにインストール
(Arduino上で, コントローラを使うためのライブラリを準備)
4. サンプルプログラムで動作確認
(コントローラの動作確認)

作業の流れ(イメージ)

目的: ESP32とPS4コントローラを, Bluetooth接続したい

1. ESP32が持つ固有の MACアドレスを調査

・ESP32開発ボード
Wi-FiとBluetooth内蔵



RDC-ESP32
TYPE1・2・3背面



SixaxisPairTool

2. 通信相手であるESP32の MACアドレスを, SixaxisPairToolアプリで, PS4コントローラへ書き込む



3. PS4コントローラは通信相手のMACアドレスを 知っているのので, ペアリングしたことになる! PS4コントローラの電源ONで操作可能に



1. ESP32のBluetooth MACアドレスを調査

コントローラ中央の「PS」ボタンが押されると、コントローラに記憶されたMACアドレスの相手とのBluetooth通信確立を試みる。
ここでは先ず、RDC本体のMACアドレスを調査する

下記ソースコードを実行し、
ツール → シリアルモニタからSTEM Du RDC－ESP32のMACアドレスを確認

```
void setup(void)
{
  Serial.begin(115200);
  uint8_t bt_mac[6];
  esp_read_mac(bt_mac, ESP_MAC_BT);
  Serial.printf("Bluetooth Mac Address => %02X:%02X:%02X:%02X:%02X:%02X\r\n",
    bt_mac[0], bt_mac[1], bt_mac[2], bt_mac[3], bt_mac[4], bt_mac[5]);
}

void loop() { }
```

1. ESP32のBluetooth MACアドレスを調査

正しく選択されていますか？

ボードマネージャ
を選択



STEM Du ESP32
のインストール
を確認



The screenshot shows the Arduino IDE 2.2.1 interface. The 'ボードマネージャ' (Board Manager) tab is active, displaying the 'stem du esp32' package by Espressif Systems. The package version '2.0.11' is selected, and the 'インストール' (Install) button is visible. Below it, the 'STEM Du ESP32 by Tairo NOMURA' package is listed with version '2.0.0' installed. The 'sketch_jan4a.ino' file is open in the editor, showing a sketch that reads the Bluetooth MAC address and prints it to the Serial Monitor. The Serial Monitor is open at the bottom, showing the output of the sketch, which includes the Bluetooth MAC address '40:22:D8:3B:74:FE'.

```
1 void setup() {  
2   // put your setup code here, to run once:  
3   Serial.begin(115200);  
4   uint8_t bt_mac[6];  
5   esp_read_mac(bt_mac, ESP_MAC_BT);  
6   Serial.printf("Bluetooth Mac Address => %02X:%02X:%02X:%02X:%02X:%02X\r\n", bt_mac[0], bt_mac[1], bt_mac[2], bt_mac[3], bt_mac[4], bt_mac[5]);  
7 }  
8  
9 void loop() {  
10  // put your main code here, to run repeatedly:  
11  
12 }  
13
```

出力 シリアルモニタ x
メッセージ ('COM9'のSTEM Du RDC-ESP32 R2にメッセージを送信するにはEnter) LFのみ 115200 baud
ets Jul 29 2019 12:21:46
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config:0: SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13516
load:0x40080400,len:3604
entry 0x400805f0
Bluetooth Mac Address => 40:22:D8:3B:74:FE

シリアルモニタを起動(「ツール」内)



シリアルモニタに、STEM Du ESP32の
BluetoothMACアドレスが表示

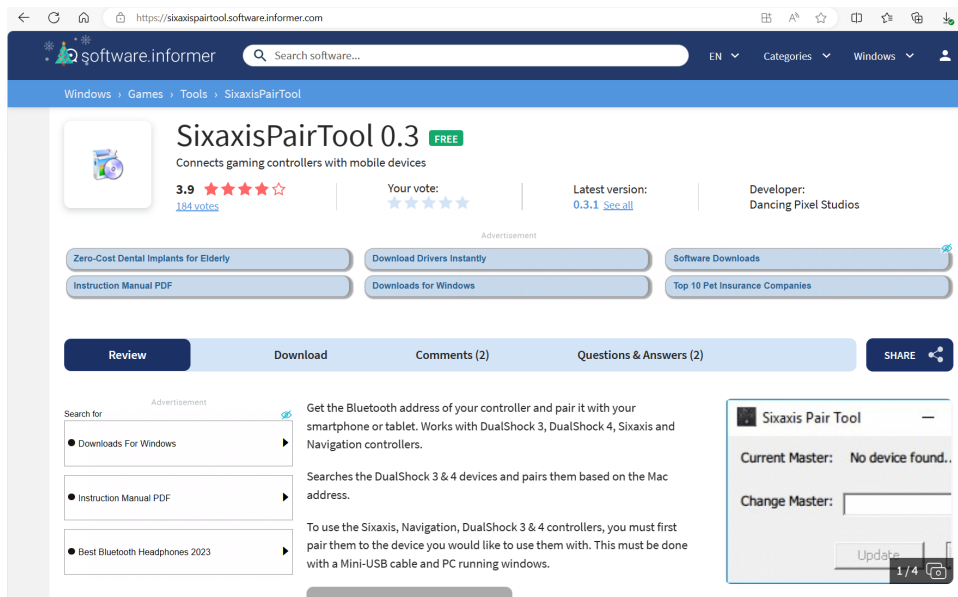


このMACアドレスをメモ(次の章で使います)

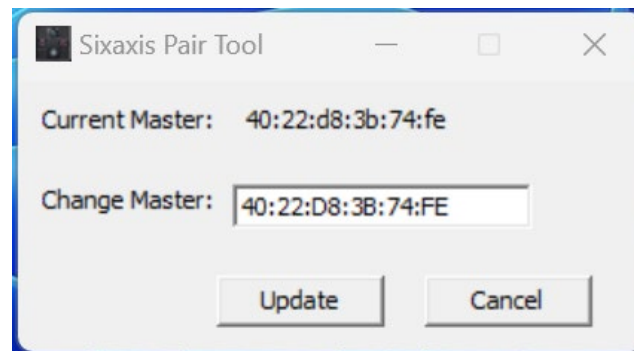
2. MACアドレスをコントローラへ書込み

調査したRDC本体のMACアドレス(前章)を,
PCにUSBケーブルで接続したコントローラへ書込む.
そこで, webサイト経由で「SixaxisPairTool」をダウンロード→インストール
このアプリはUSBケーブルで接続したコントローラに対し,
登録されたMACアドレスの確認と, その変更が行える

[SixaxisPairTool Download - Searches the DualShock 3 & 4 devices and pairs \(informer.com\)](https://sixaxispairtool.software.informer.com)



SixaxisPairTool Ver0.3.1
(2024年1月)



アプリを起動し,
PCがUSB接続したコントローラ
を認識できた表示

2. MACアドレスをコントローラへ書込み

PCとコントローラを, USBケーブルで接続

1. PCとコントローラを
USBケーブルで接続
2. Windows11の設定 →
Bluetooth とデバイス →
デバイスを確認
3. 「**Wireless Controller**」
の表示を確認



PCがUSB接続されたコントローラを,
プラグアンドプレイ(自動認識)できていれば,
「Wireless Controller」として確認できる

2. MACアドレスをコントローラへ書込み

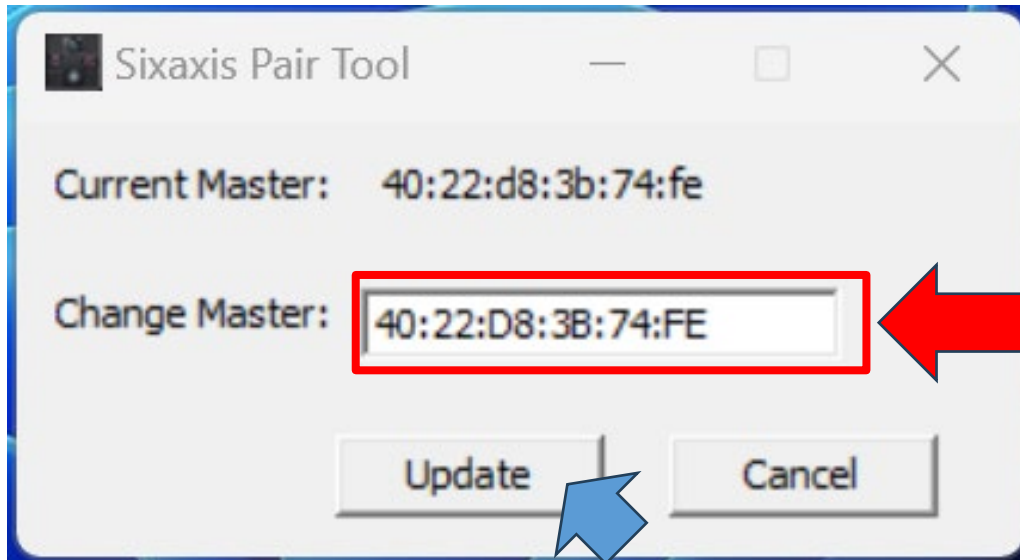
「SixaxisPairTool」を起動し、
コントローラへRDC本体のMACアドレスを書込む



SixaxisPairTool



アプリを起動



メモしたRDC本体の
MACアドレスを**入力**
→「Update」をクリック

3. PS4-esp32ライブラリをArduino IDEにインストール

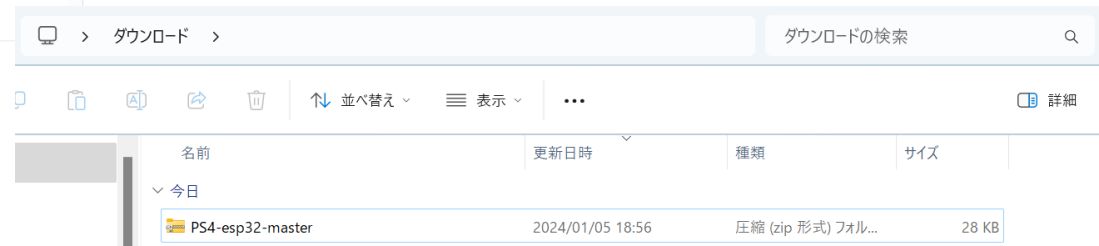
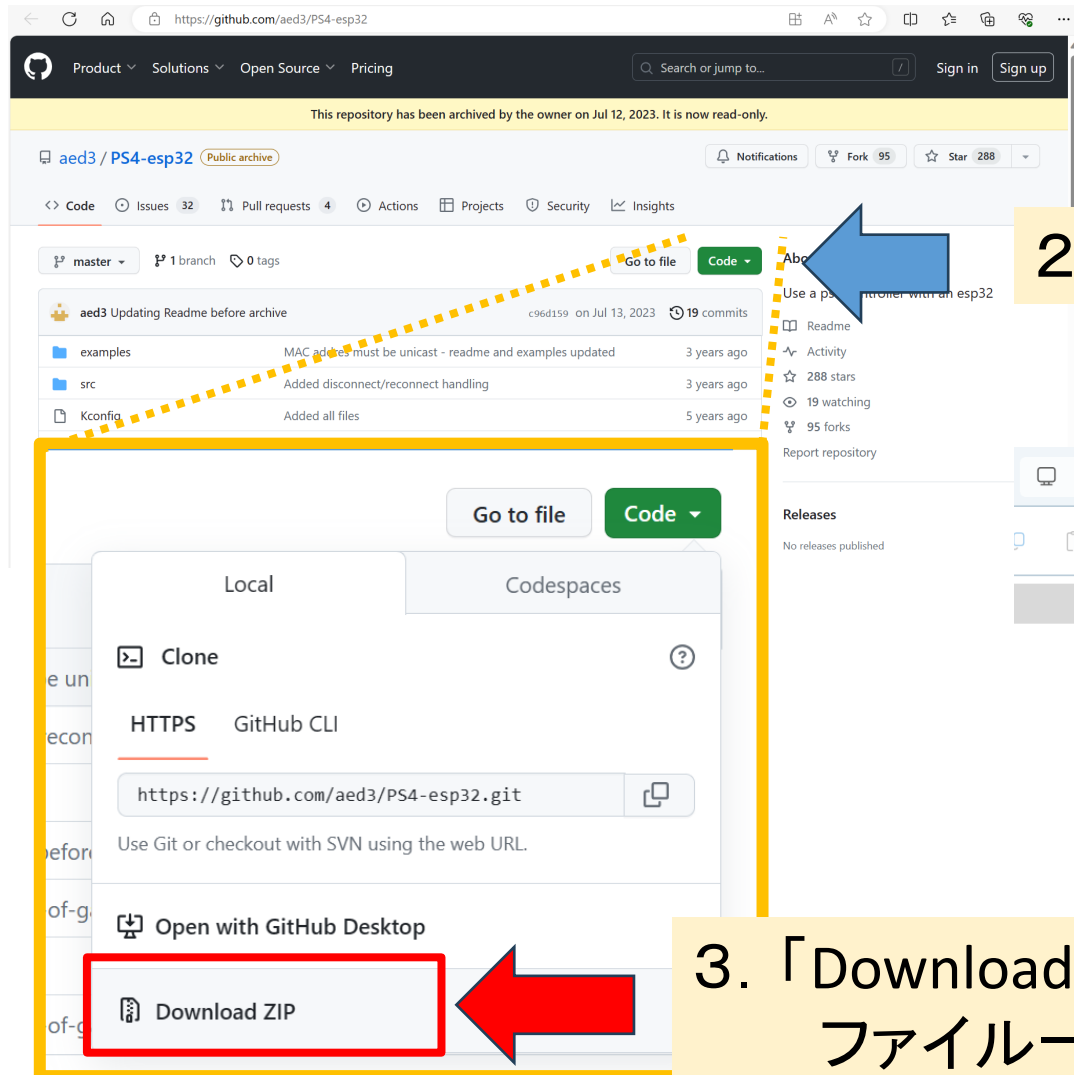
1. 次のURLを開く

<https://github.com/aed3/PS4-esp32>

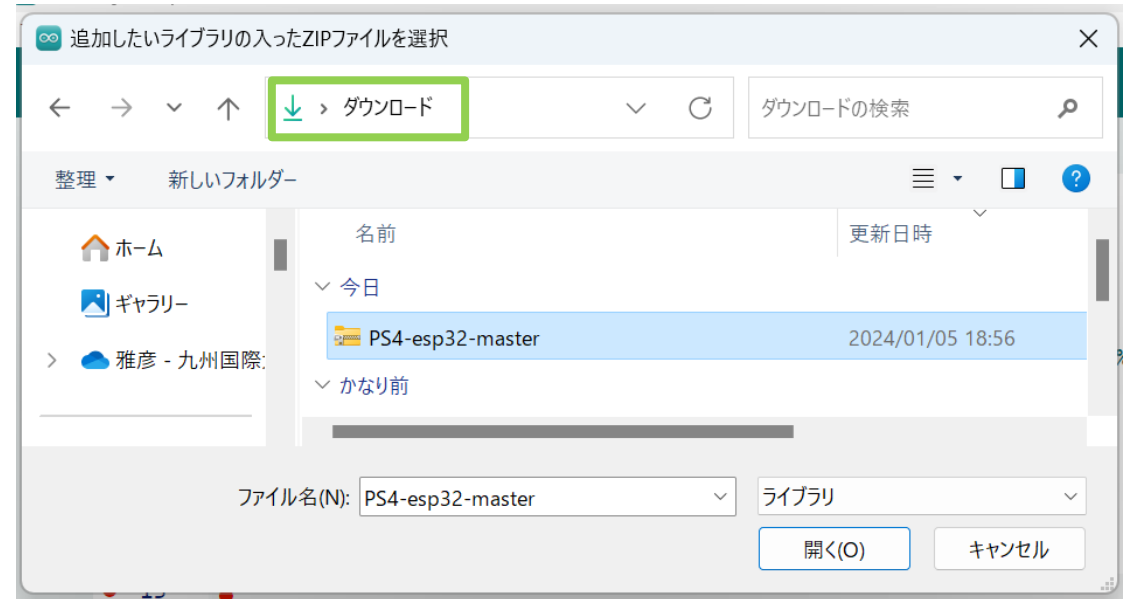
2. 「Code▼」をクリック

ダウンロードのフォルダに,
「PS4-ESP32-maste.zip」ファイルを確認

3. 「Download ZIP」の文字をクリックし,
ファイル式をダウンロード



3. PS4-esp32ライブラリをArduino IDEにインストール



1. Arduino IDEを起動し
スケッチ →
ライブラリをインクルード →
.ZIP形式のライブラリをインストール
を選択

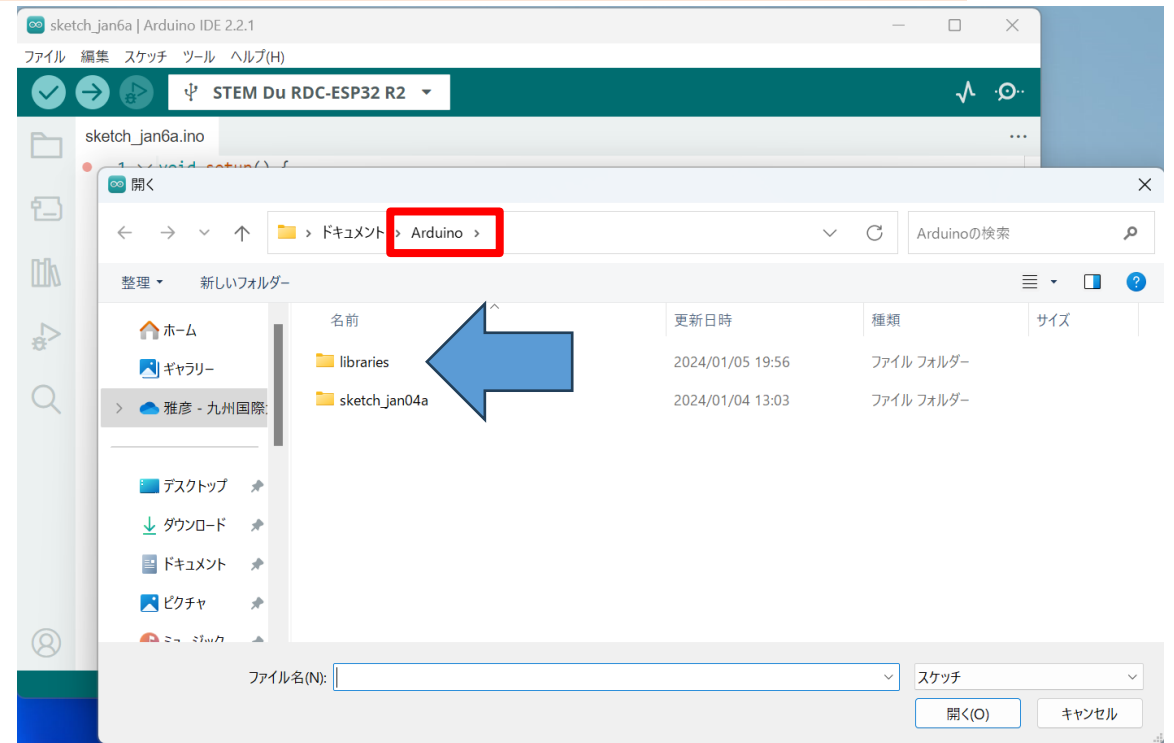
2. ダウンロードフォルダを選択し、
先ほど入手した
「PS4-ESP32-maste.zip」ファイルを
を選択して、「開く」をクリック

4. サンプルプログラムで動作確認

サンプルプログラム

「PS4ReceiveData.ino」を開き, シリアルモニタから
コントローラ操作による入力値を確認

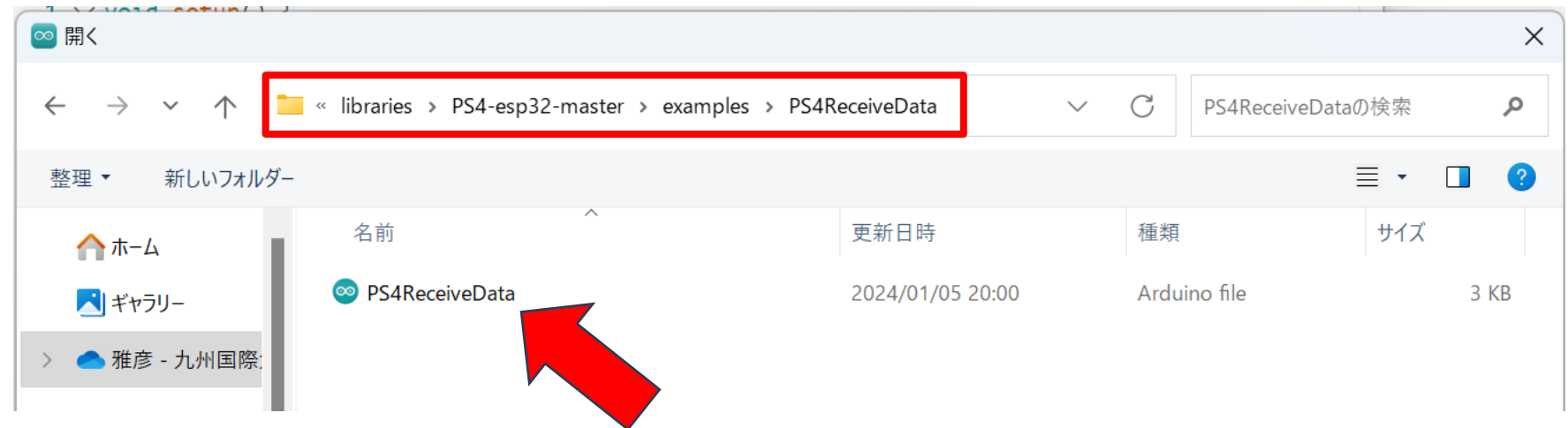
1. Arduino IDE を起動し,
ファイル → 開く
から, 「**Arduino**」のフォルダを探す
2. このフォルダ内の「libraries」内に
先ほど導入した「PS4-ESP32-master」
フォルダを探す



4. サンプルプログラムで動作確認


3.Arduino フォルダ内の
libraries → PS4-ESP32-master →
examples と続けて開く

4. examples フォルダ内の
PS4ReceiveData フォルダを開き
Arduino file である
PS4ReceiveData
を開く



4. サンプルプログラムで動作確認

5. PS4ReceiveData のコードから、
PS4.begin("xx:xx:xx:xx:xx:xx");
へ、メモしたRDC本体の
MACアドレスを入力する

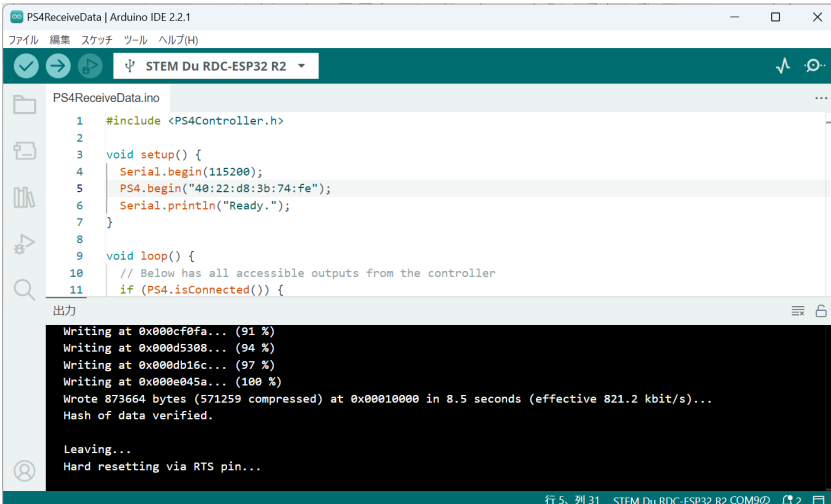
6. 検証  → エラーが無ければ
書き込み  → 完了を待つ



```
PS4ReceiveData | Arduino IDE 2.2.1
ファイル 編集 スケッチ ツール ヘルプ(H)
STEM Du RDC-ESP32 R2
PS4ReceiveData.ino
1 #include <PS4Controller.h>
2
3 void setup() {
4   Serial.begin(115200);
5   PS4.begin("40:22:d8:3b:74:fe");
6   Serial.println("Ready.");
7 }
8
9 void loop() {
10  // Below has all accessible outputs from the controller
11  if (PS4.isConnected()) {
12    if (PS4.Right()) Serial.println("Right Button");
13    if (PS4.Down()) Serial.println("Down Button");
14    if (PS4.Up()) Serial.println("Up Button");
15    if (PS4.Left()) Serial.println("Left Button");
16  }
17 }
```

出力

行 5、列 31 STEM Du RDC-ESP32 R2 COM9の



```
PS4ReceiveData | Arduino IDE 2.2.1
ファイル 編集 スケッチ ツール ヘルプ(H)
STEM Du RDC-ESP32 R2
PS4ReceiveData.ino
1 #include <PS4Controller.h>
2
3 void setup() {
4   Serial.begin(115200);
5   PS4.begin("40:22:d8:3b:74:fe");
6   Serial.println("Ready.");
7 }
8
9 void loop() {
10  // Below has all accessible outputs from the controller
11  if (PS4.isConnected()) {
12    if (PS4.Right()) Serial.println("Right Button");
13    if (PS4.Down()) Serial.println("Down Button");
14    if (PS4.Up()) Serial.println("Up Button");
15    if (PS4.Left()) Serial.println("Left Button");
16  }
17 }
```

出力

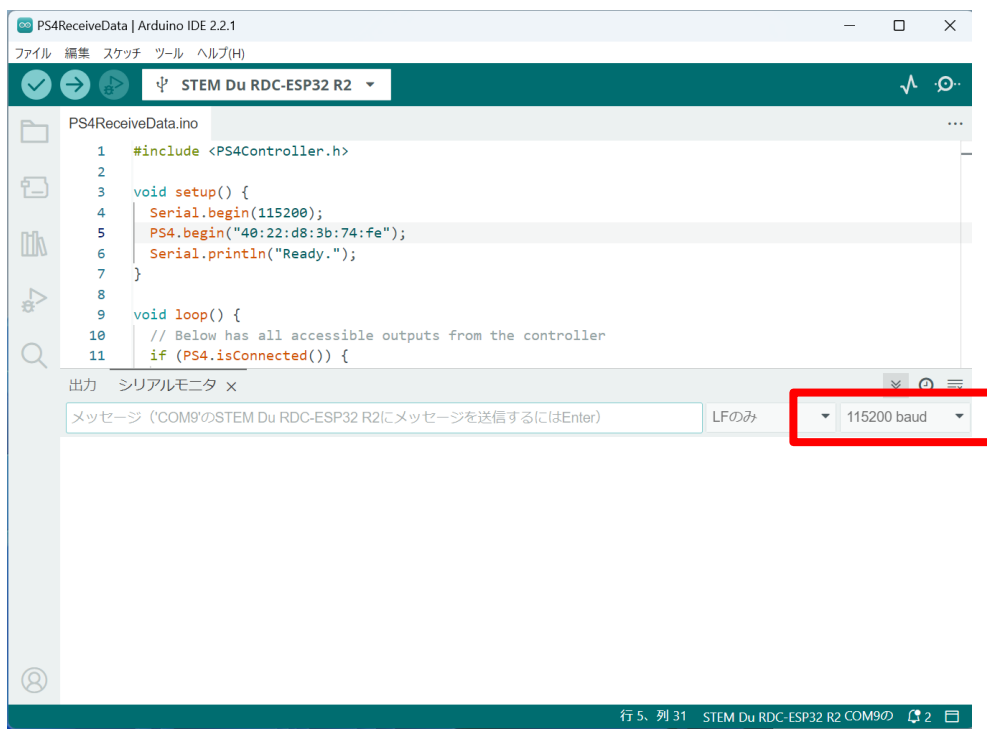
Writing at 0x000cf0fa... (91 %)
Writing at 0x000d5308... (94 %)
Writing at 0x000db16c... (97 %)
Writing at 0x000e045a... (100 %)
Wrote 873664 bytes (571259 compressed) at 0x0010000 in 8.5 seconds (effective 821.2 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...

行 5、列 31 STEM Du RDC-ESP32 R2 COM9の

4. サンプルプログラムで動作確認

7.「ツール」→ シリアルモニタ
を起動する

8.シリアルモニタの通信速度から
「115200baud」を選択



4. サンプルプログラムで動作確認

9. コントローラの「PSボタン」をクリックすると
背面LEDが点滅し, RDC本体とのBluetooth通信
を開始する. 接続されると, 背面LEDは点灯し続ける

10. シリアルモニタに表示される,
コントローラ操作の入力値を確認する

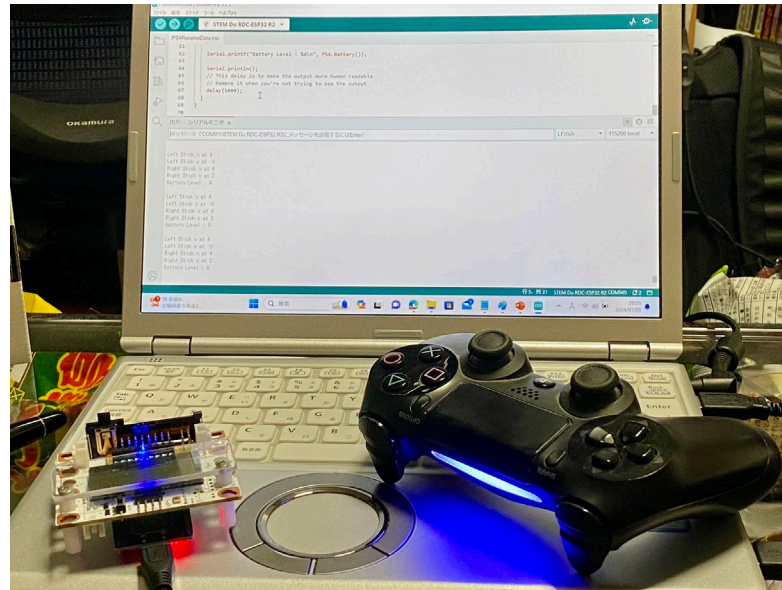


出力 シリアルモニタ x

メッセージ (COM9)のSTEM Du RDC-ESP32 R2

```
Left Stick x at -4  
Left Stick y at -4  
Right Stick x at -6  
Right Stick y at -2  
Battery Level : 8
```

```
Left Stick x at -4  
Left Stick y at -4  
Right Stick x at -6  
Right Stick y at -3  
Battery Level : 8
```



RDC本体の電源が切れると
通信が途切れ,
コントローラの電源は
自動的に切れる
(背面LEDが切れる)