

Module ASR Arduino チュートリアル

1. 準備作業

- 1. 環境構築: [Arduino IDE 入門ガイド](#)を参考にIDEをインストールし、使用する開発ボードに応じてボードマネージャと必要なライブラリをインストールしてください。
- 2. 使用ライブラリ:
 - [M5Unified](#)
 - [M5GFX](#)
 - [M5Unit-ASR](#)
- 3. 使用ハードウェア:
 - [CoreS3-SE](#)
 - [Module ASR](#)
 - [Unit Dual Button](#)
 - [Grove to 4P](#)



2. 注意事項

ピン互換性

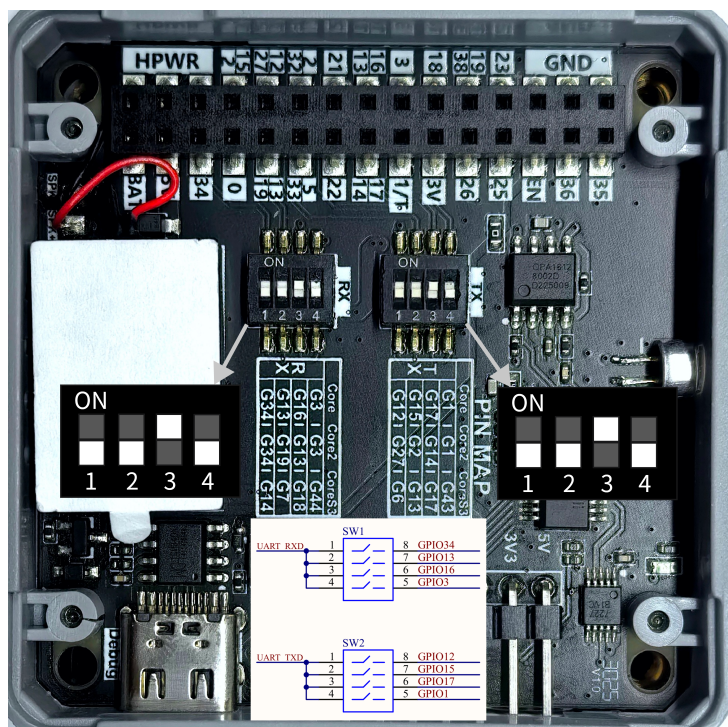
各ホストのピン配置が異なるため、ご使用前に製品ドキュメントの[ピン互換表](#)を参照し、実際の配線に合わせてサンプルプログラムを修正してください。

3. サンプルプログラム

- 本チュートリアルで使用するメインコントローラはCoreS3-SEで、Module ASRと組み合わせて音声インタラクションを実現します。使用前に下図を参考に、DIPスイッチを指定の位置に切り替えてください。

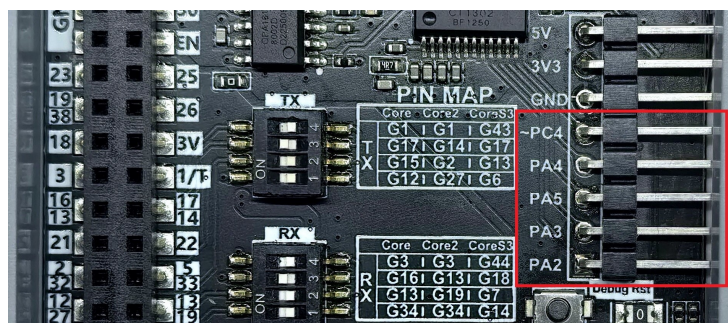
3.1 DIPスイッチ

Module ASRはUART通信を採用しています。実際の配線に合わせてプログラム内のピン定義を修正してください。接続後の対応UART IOはG18 (RX)、G17 (TX)です。DIPスイッチの実物位置は下図の通りです:



3.2 IO端子

- Module ASRは5つのIO端子を備え、音声コマンドで各IOの出力レベルを制御したり、IO入力で特定の音声応答をトリガーできます。IO端子は下図の通りです:



3.3 サンプルコード

Module ASRは音声・シリアルによるウェイクアップやコマンド入力をサポートし、音声およびシリアルで現在のコマンド情報を返します。デフォルトファームウェアにはいくつかのプリセットコマンドが含まれており、下記の**turn off**などに限りません。詳細は[Module ASR出荷時プリセットコマンド](#)をご参照ください。

```
1  #include <M5Unified.h>
2  #include <unit_asr.hpp>
3
4  ASRUnit asr;
5
6  void turnOnHandler()
7  {
8      Serial.println("turn on command received!");
9      M5.Display.clear();
10     M5.Display.drawCenterString("turn off!", 160, 110);
11 }
12
13 void turnOffHandler()
14 {
15     Serial.println("turn off command received!");
16     M5.Display.clear();
17     M5.Display.drawCenterString("turn off!", 160, 110);
18 }
19
20 void wakeupHandler()
21 {
22     Serial.println("wakeup command received!");
23     M5.Display.clear();
24     M5.Display.drawCenterString("wakeup!", 160, 110);
25 }
26
27 void PA2_HighHandler()
28 {
29     Serial.println("set PA2 level to high");
30     M5.Display.clear(TFT_GREEN);
31     M5.Display.drawCenterString("PA2 level: high", 160, 110);
32 }
33
34 void PA2_LowHandler()
35 {
36     Serial.println("set PA2 level to low");
37     M5.Display.clear(TFT_YELLOW);
38     M5.Display.drawCenterString("PA2 level: low", 160, 110);
39 }
40
41 void AddHandler()
42 {
43     Serial.println("Add command received!");
44     M5.Display.clear();
45     M5.Display.drawCenterString("Add", 160, 110);
46 }
47
48 void PA4_InputHandler()
49 {
50     Serial.println("PA4 key pressed");
```

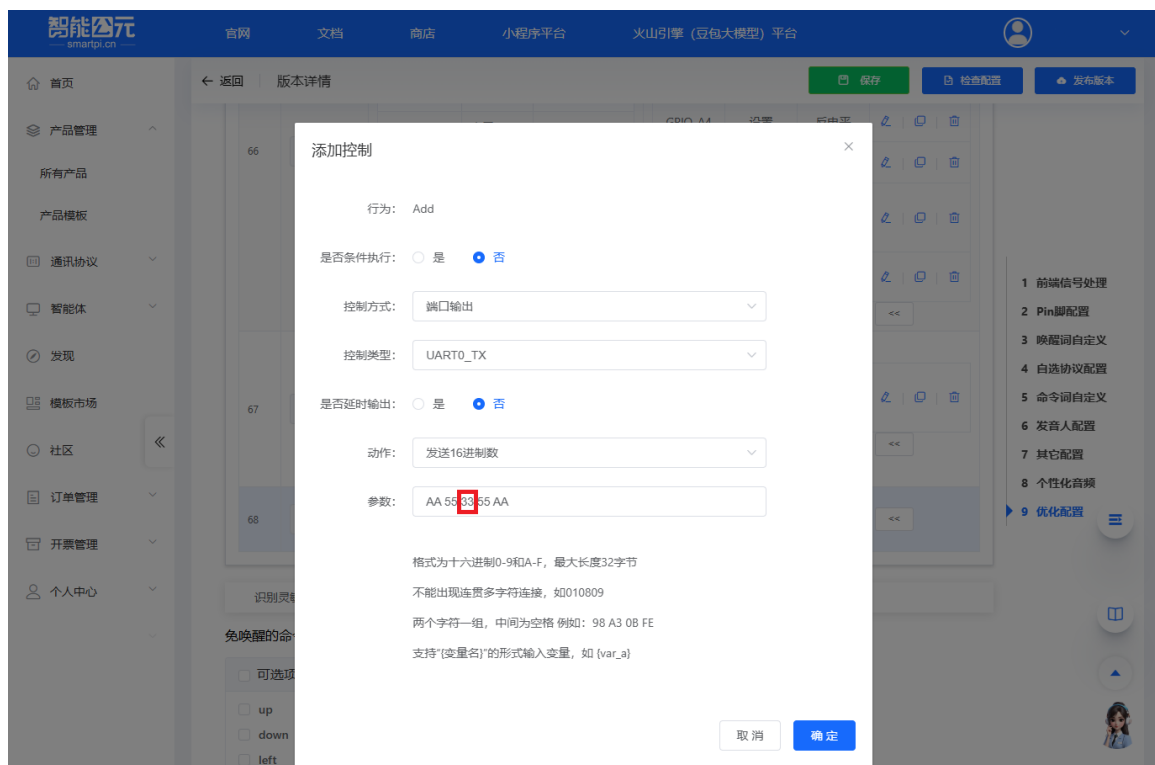
```

51     M5.Display.clear();
52     M5.Display.drawCenterString("PA4 key pressed", 160, 110);
53 }
54
55 void setup()
56 {
57     M5.begin();
58     M5.Display.setFont(&fonts::FreeMonoBold12pt7b);
59     Serial.begin(115200);
60     asr.begin(&Serial1, 115200, 18, 17);
61     asr.addCommandWord(0x14, "turn on", turnOnHandler);
62     asr.addCommandWord(0x15, "turn off", turnOffHandler);
63     asr.addCommandWord(0x31, "Hi,ASR", wakeupHandler);
64     asr.addCommandWord(0x32, "hello", wakeupHandler);
65     asr.addCommandWord(0x50, "PA2 high level", PA2_HighHandler);
66     asr.addCommandWord(0x51, "PA2 low level", PA2_LowHandler);
67     asr.addCommandWord(0xFF, "Hi,M Five", wakeupHandler);
68     asr.addCommandWord(0x33, "Add", AddHandler);
69     asr.addCommandWord(0x5B, "PA4 input", PA4_InputHandler);
70     asr.printCommandList();
71     M5.Display.clear();
72     M5.Display.drawCenterString("Say", 160, 60);
73     M5.Display.drawCenterString("\nHi,M Five\n", 160, 80);
74     M5.Display.drawCenterString("\nHello\n", 160, 100);
75     M5.Display.drawCenterString("\nHi,ASR\n", 160, 120);
76     M5.Display.drawCenterString("to wakeup!", 160, 140);
77 }
78
79 void loop()
80 {
81     M5.update();
82     if (asr.update()) {
83         Serial.println(asr.getCurrentCommandWord());
84         Serial.printf("0x%X\n", asr.getCurrentCommandNum());
85         Serial.println(asr.getCurrentRawMessage());
86         Serial.println((asr.checkCurrentCommandHandler()));
87     }
88 }

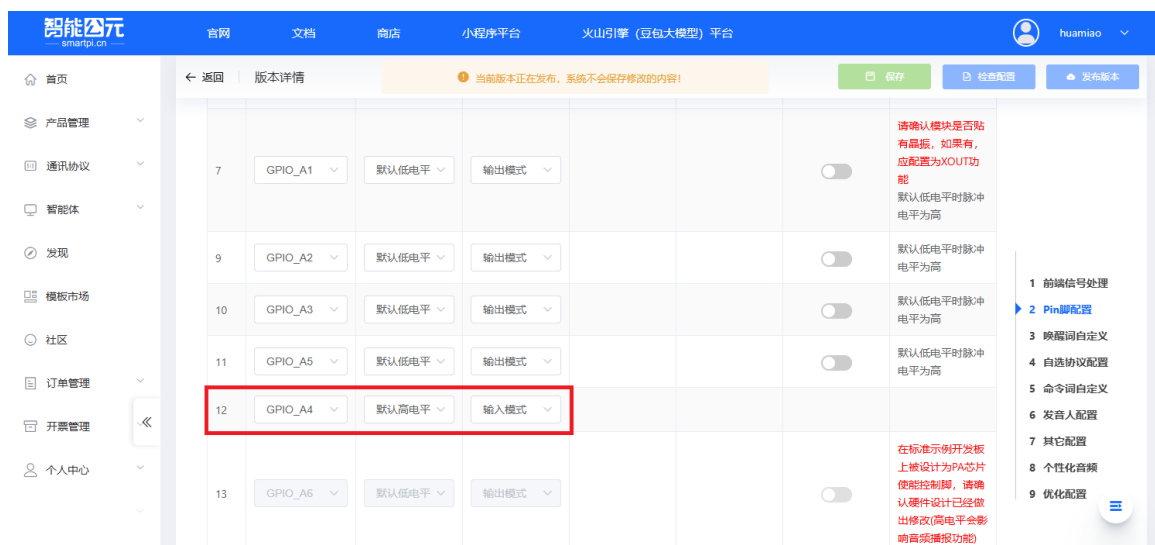
```

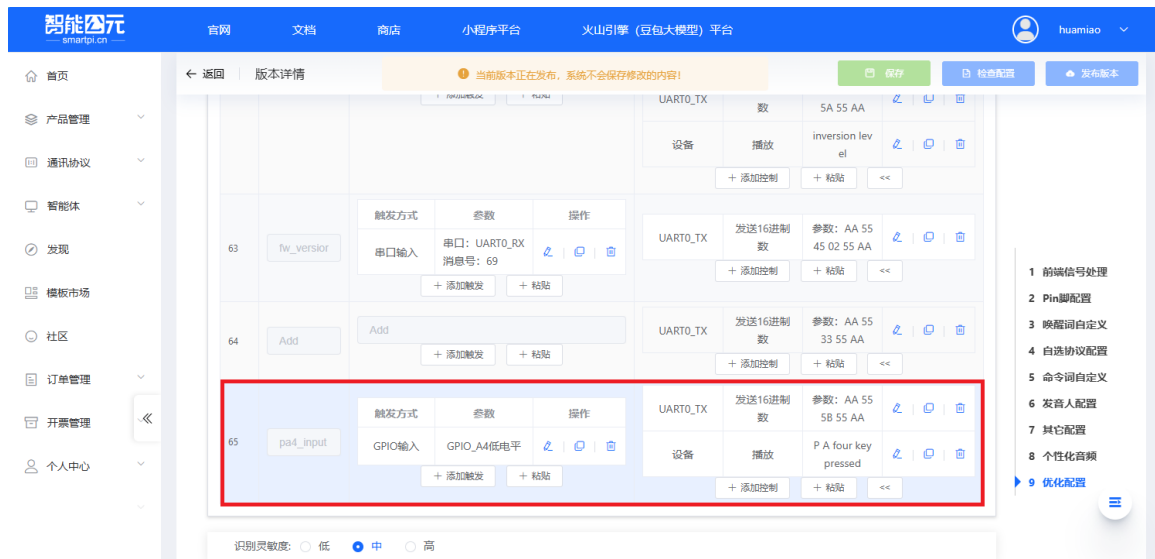
3.4 カスタムコマンド設定

- Module ASRはカスタムファームウェアを再生成・書き込みすることでのみ新しいインタラクションコマンドの追加や変更が可能です。
詳細は[Module ASRカスタムファームウェア生成と書き込みチュートリアル](#)をご参照ください。
- 上記例ではカスタム音声コマンドAddのフレームはAA 55 33 55 AAで、コマンドコードは0x33です。ファームウェア設定は下図の通り。例ではaddCommandWord関数でコマンドをテーブルに追加し、コールバックを登録しています。



- 上記例のPA4 inputはカスタムコマンドで、フレームはAA 55 5B 55 AA、コマンドコードは0x5Bです。ファームウェア設定ではピン設定でGPIO_A4を入力モードに設定する必要があります。詳細は下図の通り。例ではaddCommandWord関数でコマンドをテーブルに追加し、コールバックを登録しています。





4. コンパイルと書き込み

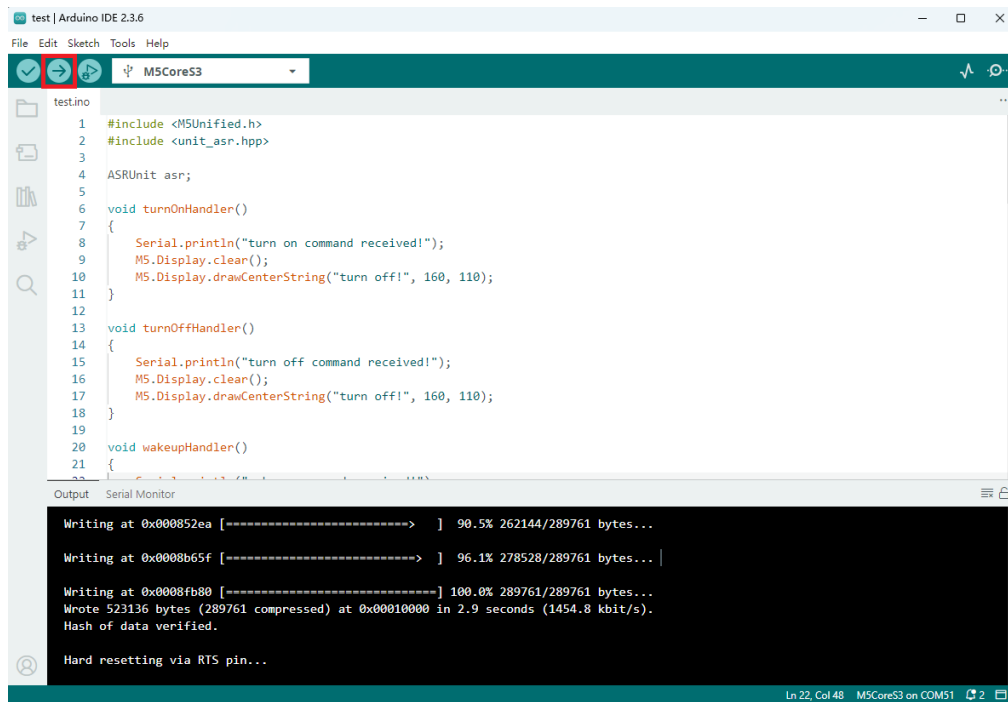
- 1. ダウンロードモードに入る：CoreS3-SEのリセットボタンを約2秒間長押しし、内部の緑色LEDが点灯したら離します。これでデバイスはダウンロードモードに入り、書き込み待機状態になります。

注意

デバイスによってダウンロードモードへの入り方が異なる場合があります。詳細は[Arduino IDE 入門ガイド](#)ページ下部のデバイス書き込みガイドリストをご参照ください。



- 2. デバイスのポートを選択し、Arduino IDE左上のコンパイル・書き込みボタンをクリックして、プログラムのコンパイルと書き込みが完了するまで待ちます。



5. コマンドインタラクション

○ ウェイクアップ操作

電源投入後、ホスト画面は下図左のように表示されます。音声コマンド**Hi M Five**でモジュールをウェイクアップすると、**I'm here**と応答し、ホスト画面は下図右のように表示されます。



シリアルモニタ出力例:

```
wakeup command received!  
Hi,M Five  
0xFF  
0xAA 0x55 0xFF 0x55 0xAA  
1
```

○ プリセットコマンドインタラクション

1. 音声コマンド**Turn on**を使用すると、モジュールは**OK**と応答し、ホスト画面に**turn on!**と表示されます。
2. 音声コマンド**Turn off**を使用すると、モジュールは**OK**と応答し、ホスト画面に**turn off!**と表示されます。

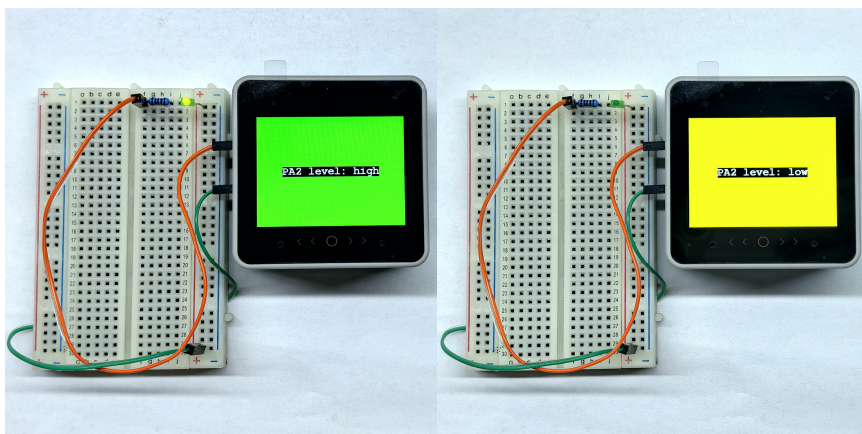


シリアルモニタ出力例:

```
turn on command received!
turn on
0x46
0xAA 0x55 0x14 0x55 0xAA
1
turn on command received!
turn on
0x46
0xAA 0x55 0x15 0x55 0xAA
1
```

○ IO制御コマンドインタラクション

1. 音声コマンド **PA2 high level** を使用すると、モジュールは **PA2 high level** と応答し、ホスト画面に **PA2 level: high** と表示され、背景色が緑色になり、PA2がHigh出力となります。
2. 音声コマンド **PA2 low level** を使用すると、モジュールは **PA2 low level** と応答し、ホスト画面に **PA2 level: low** と表示され、背景色が黄色になり、PA2がLow出力となります。
3. Unit Dual ButtonのBキー（赤色ボタン）を押すと、モジュールは **PA4 key pressed** と応答し、ホスト画面に **PA4 key pressed** と表示されます。



[Module_ASR_arduino_pa4_input.mp4](#)

シリアルモニタ出力例:

```
set PA2 level to high
PA2 high level
0x50
0xAA 0x55 0x50 0x55 0xAA
1
set PA2 level to low
PA2 low level
0x51
0xAA 0x55 0x51 0x55 0xAA
1
PA4 key pressed
PA4 input
0x5B
0xAA 0x55 0x5B 0x55 0xAA
1
```

◦ カスタムコマンドインタラクション

音声コマンドAddを使用すると、モジュールはAdd successfullyと応答し、ホスト画面にAdd successfully!と表示されます。



シリアルモニタ出力例:

```
Add command received!
Add
0x33
0xAA 0x55 0x33 0x55 0xAA
1
```