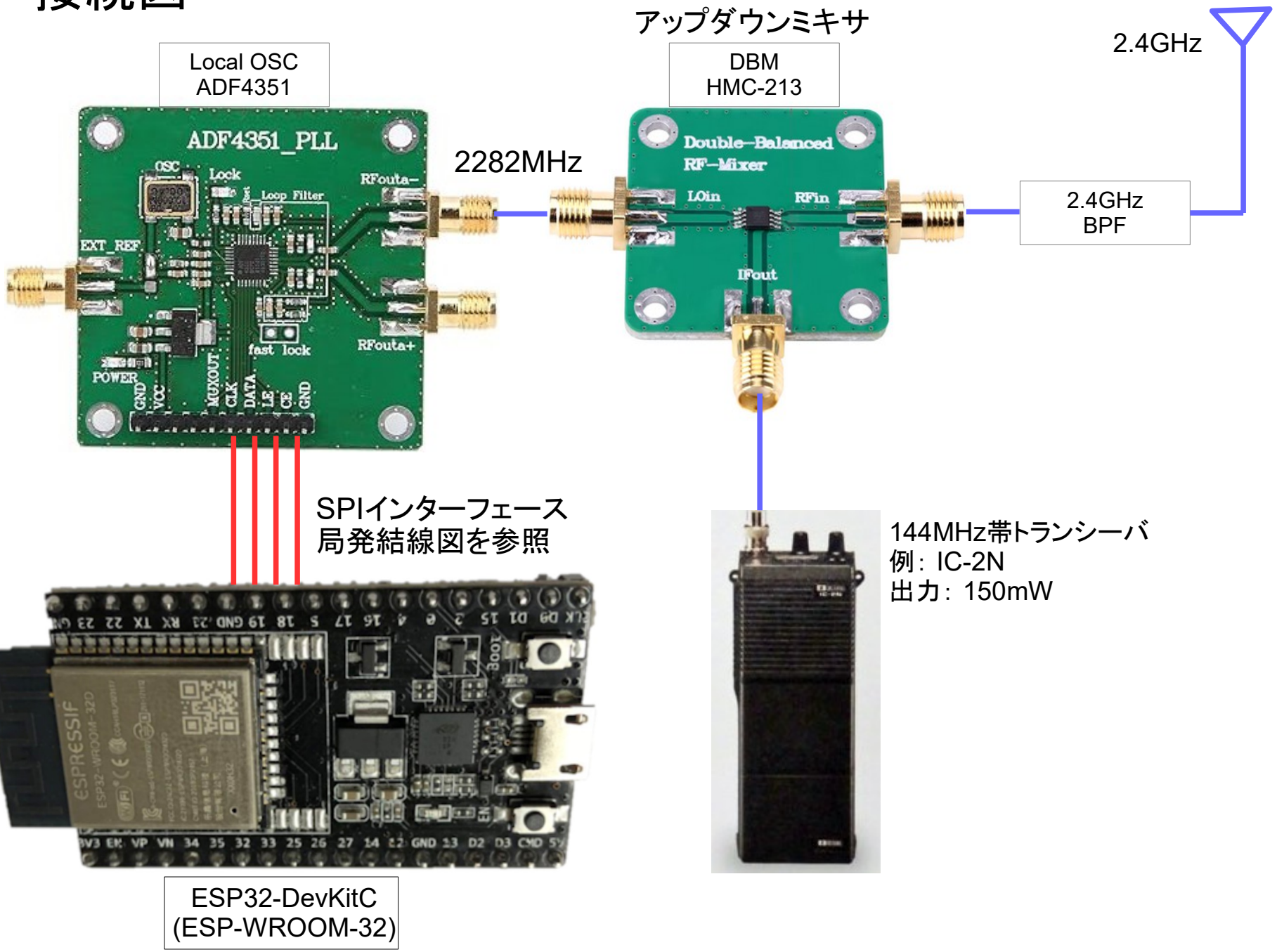
2.4GHz簡易トランシーバの実験

従来、マイクロ波用のトランスバータを製作する場合、局発部に多段にわたる逓倍回路が必要となります。 これが自作を困難にしていると思います。
ここでは、ADF4351_PLL基板とマイコン基板を使用して手軽に局発部（マイクロ波発振）を作り、双方向ミキサーを通して簡単に2.4GHz トランシーバを実験してみます。

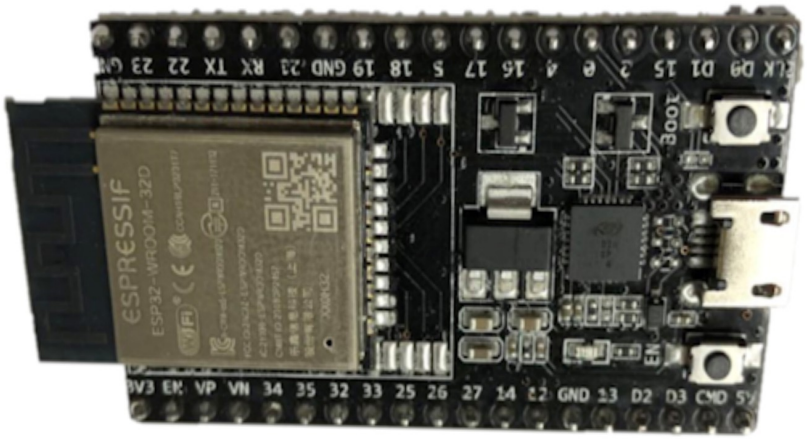
1. ）2.4GHz 簡易トランシーバ　接続図
2. ）局発(Local OSC) - 1　PLL基板とマイコン基板について
3. ）局発(Local OSC) - 2　ADF4351_PLL基板とESP32結線図
4. ）局発(Local OSC) - 3　周波数（レジスタ値）決定ツール
5. ）局発(Local OSC) - 4　ソースプログラム①
6. ）局発(Local OSC) - 5　ソースプログラム②
7. ）Arduino IDE　プログラム統合開発環境

2024/11/12　JA4CXX / 山本俊通

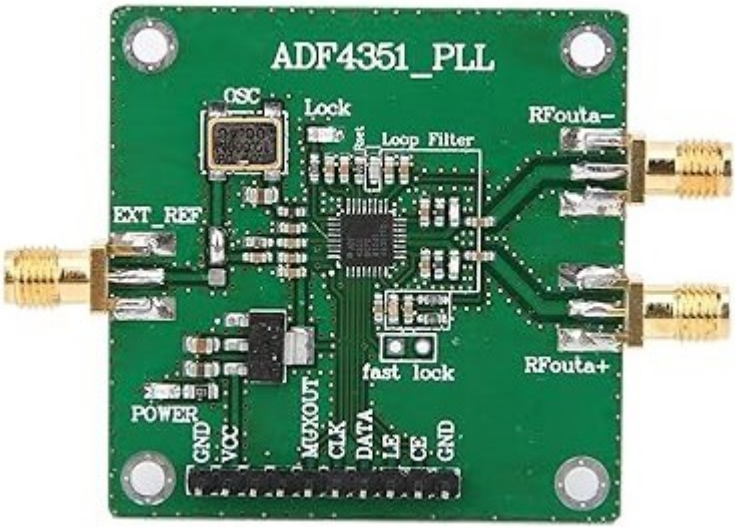
2.4GHz 簡易トランシーバ 接続図



局発(Local OSC) – 1 使用したPLL基板とマイコン基板について

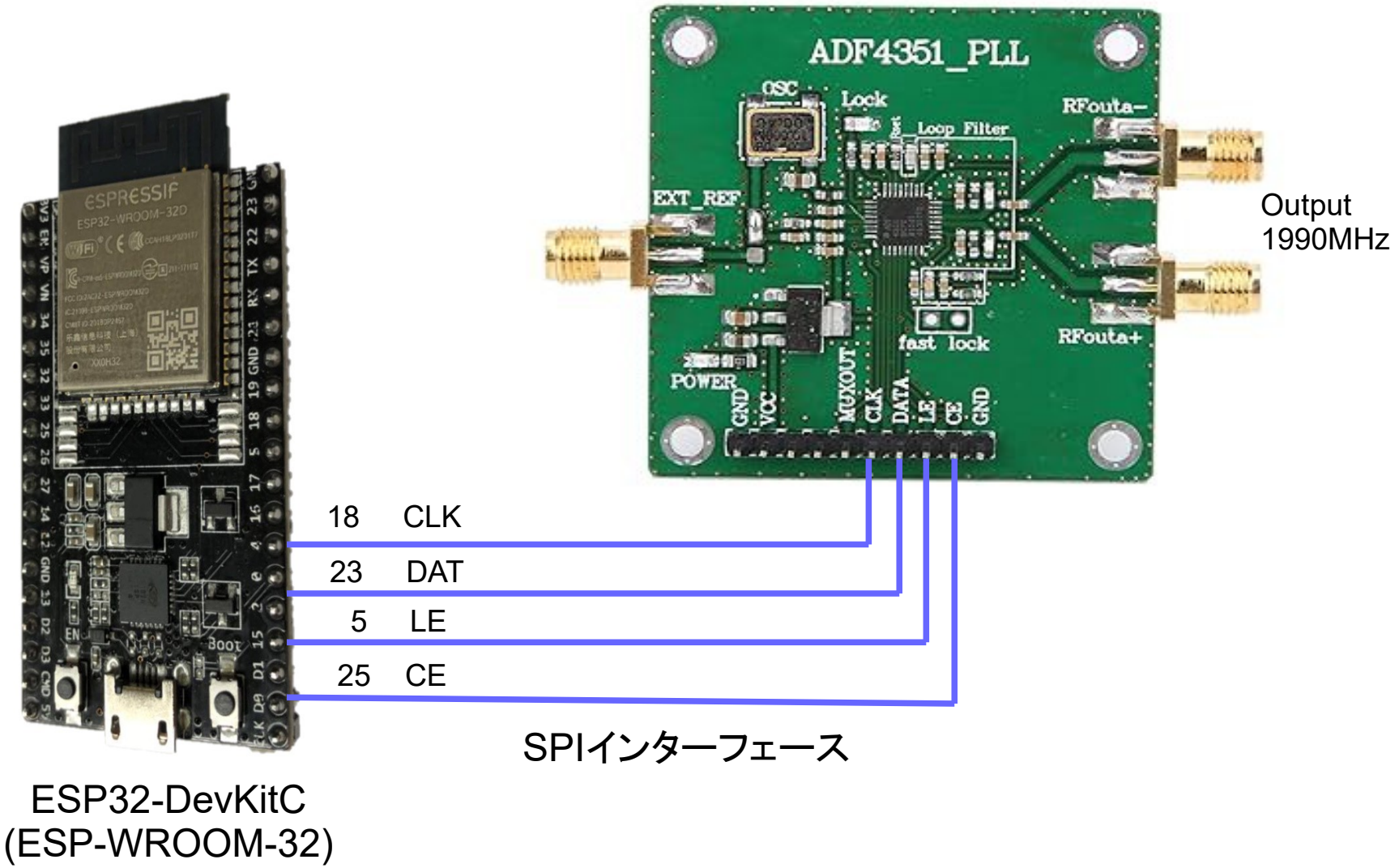


ESP32-DevKitCはESP-WROOM-32と呼ばれる
マイコンを搭載したワイヤレスモジュールです。
WiFiとBluetoothを搭載したMCUで、動作温度が
-40℃から＋125℃まで安定した動作が可能な
堅牢な設計、超低消費電力設計、さまざまな
モジュールを内蔵した高度な統一性を意識した
設計などの特徴があります。
ESP32マイコンはデュアルコア、動作クロックは
240MHz、RAMは520KB、フラッシュメモリは64MBと
優れたスペックとなっており、マイコンとしては
十分すぎるほどの性能です。



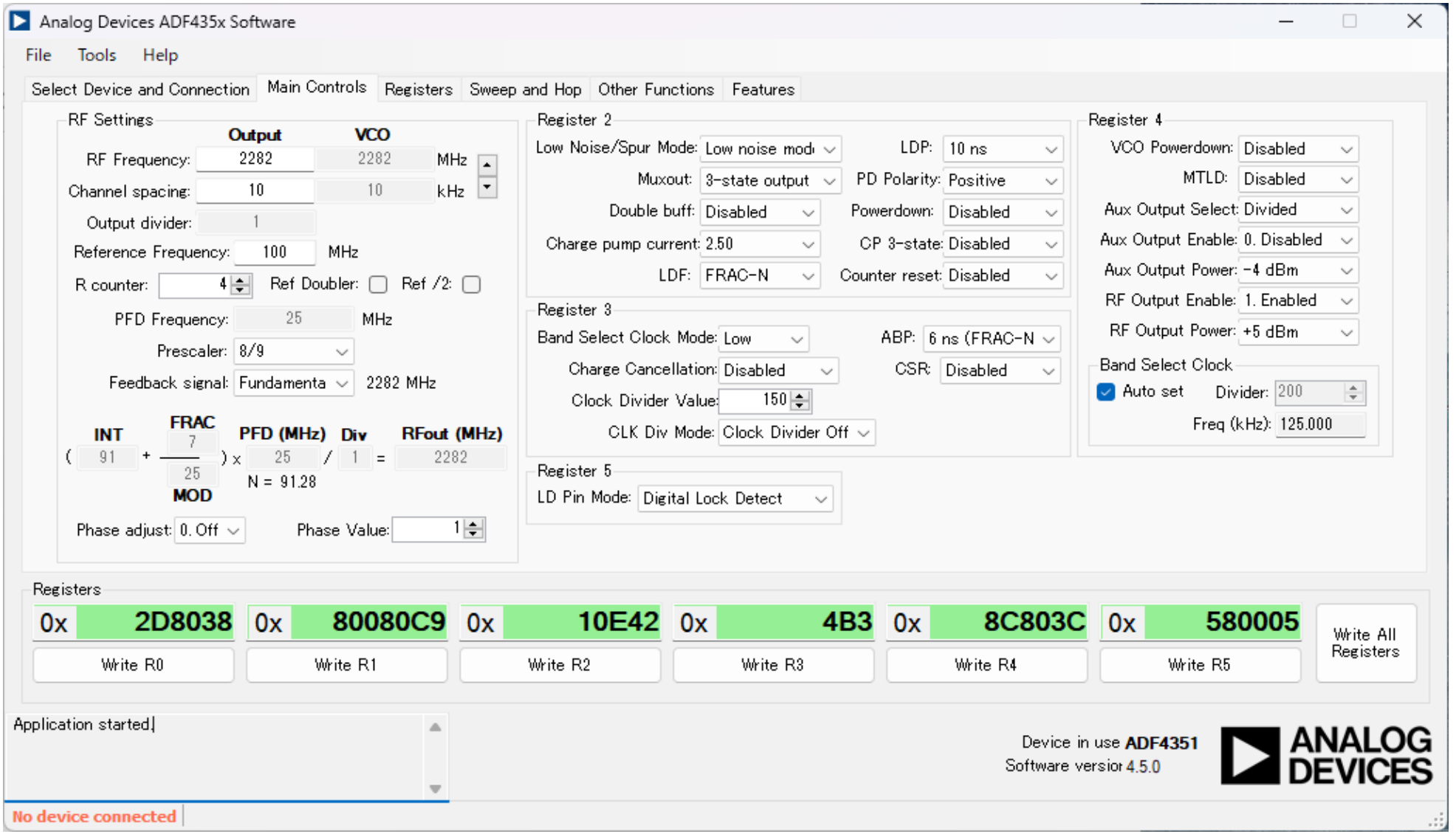
ADF4351は、外部のループ・フィルタと外部からの
リファレンス周波数を使うことによって、フラクショナルN
（分数分周型）またはインテジャーN（整数分周型）の
PLL周波数シンセサイザを実現することができます。
ADF4351は、2200MHz～4400MHzの基本出力周波数
としての電圧制御発振器（VCO）を内蔵しています。
さらに、1/2/4/8/16/32または64の分周回路は、35 MHzまで
低いRF出力周波数を発生させることができます
全ての内蔵レジスタの制御は、簡単な3線インターフェースを
介して行われます。
このデバイスは、3.0V～3.6Vの電源範囲で動作します。

局発(Local OSC) – 2 ADF4351_PLL基板とESP32結線図



局発(Local OSC) – 3 周波数(レジスタ値)設定ツール　出力周波数2282MHz

ADF4351のレジスタ値決定ツール(ADF435x software)。
レジスタのR5からR0まで順次書き込みます。



局発(Local OSC) – 4 ソースプログラム①

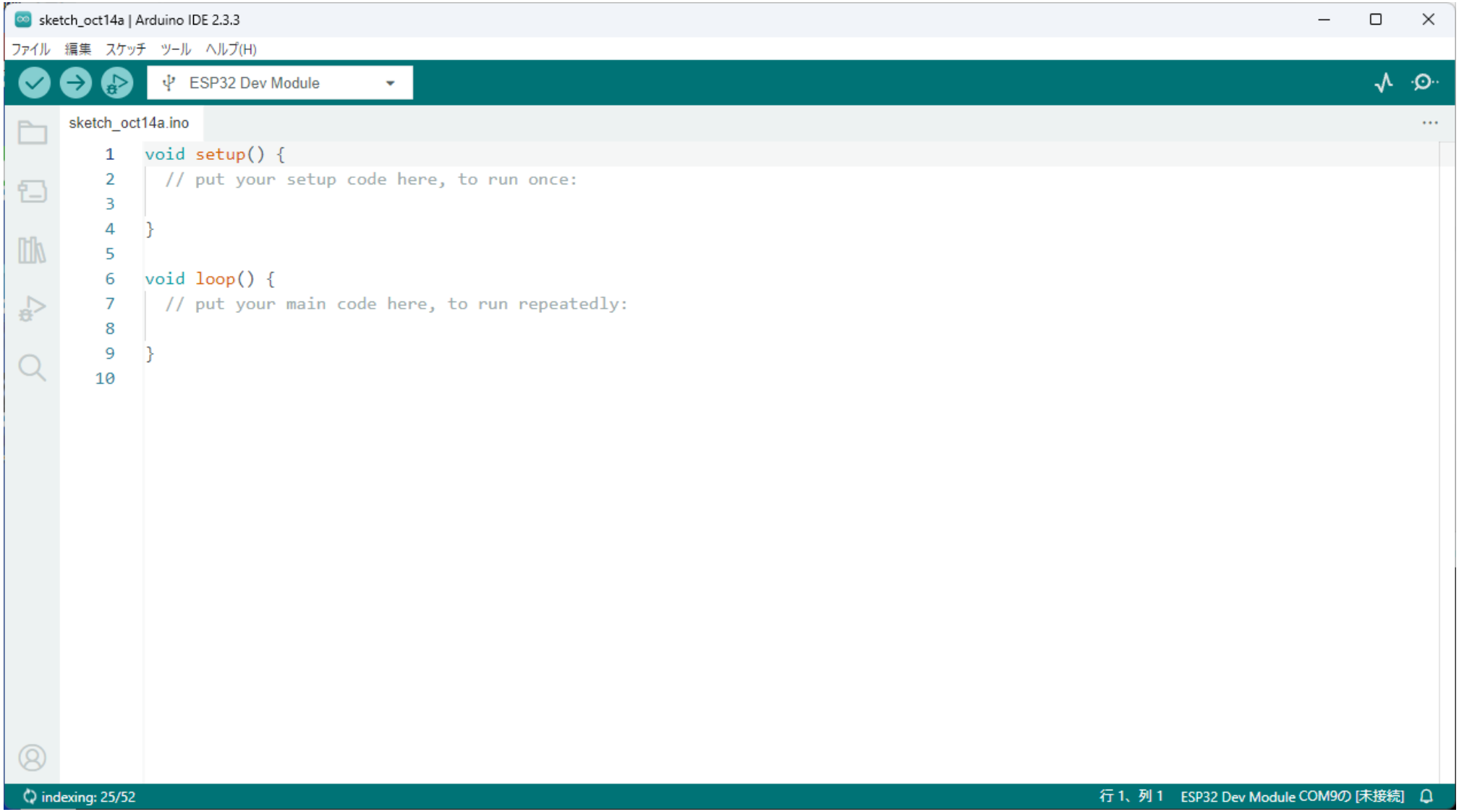
https://github.com/Ryushane/ADF4351-ESP32/tree/master から
ライブラリをダウンロードし、ADF4351.cpp、ADF4351.h を
下記のソースファイルと同じディレクトリに配置します。

```
////////////////////////////////////////
//        ADF4351 RF Generator                    /
//        Spot-OSC-2282MHz.ino                    /
//        2023-2-16 - 2024-11-11     JA4CXX         /
//// ADF4351 ////////////////////////////
//        MCU-Pin        ADF4351 Board-Pin
//        IO = 18          CLK
//        IO = 23          DAT
//        IO = 5     LE
//        IO = 25          CE
//--------------------------------------
//        Board Power +5V
//// SSD1306 ////////////////////////////
//        MCU-Pin        SSD1306 Board-Pin
//        IO = 21          SDA
//        IO = 22          SCL
//--------------------------------------
//        Board Power +3.3V
////////////////////////////////////////
#include "ADF4351.h"
#include <SPI.h>
#include "SSD1306.h"//ディスプレイ用ライブラリを読み込み

#define clock 18
#define data 23
#define LE 5
#define CE 25

ADF4351  adf4351(clock,data,LE,CE); // declares object PLL of type ADF4351
SSD1306  display(0x3c, 21, 22);     //SSD1306インスタンスの作成（I2Cアドレス,SDA,SCL）
```

局発(Local OSC) – 5 ソースプログラム②

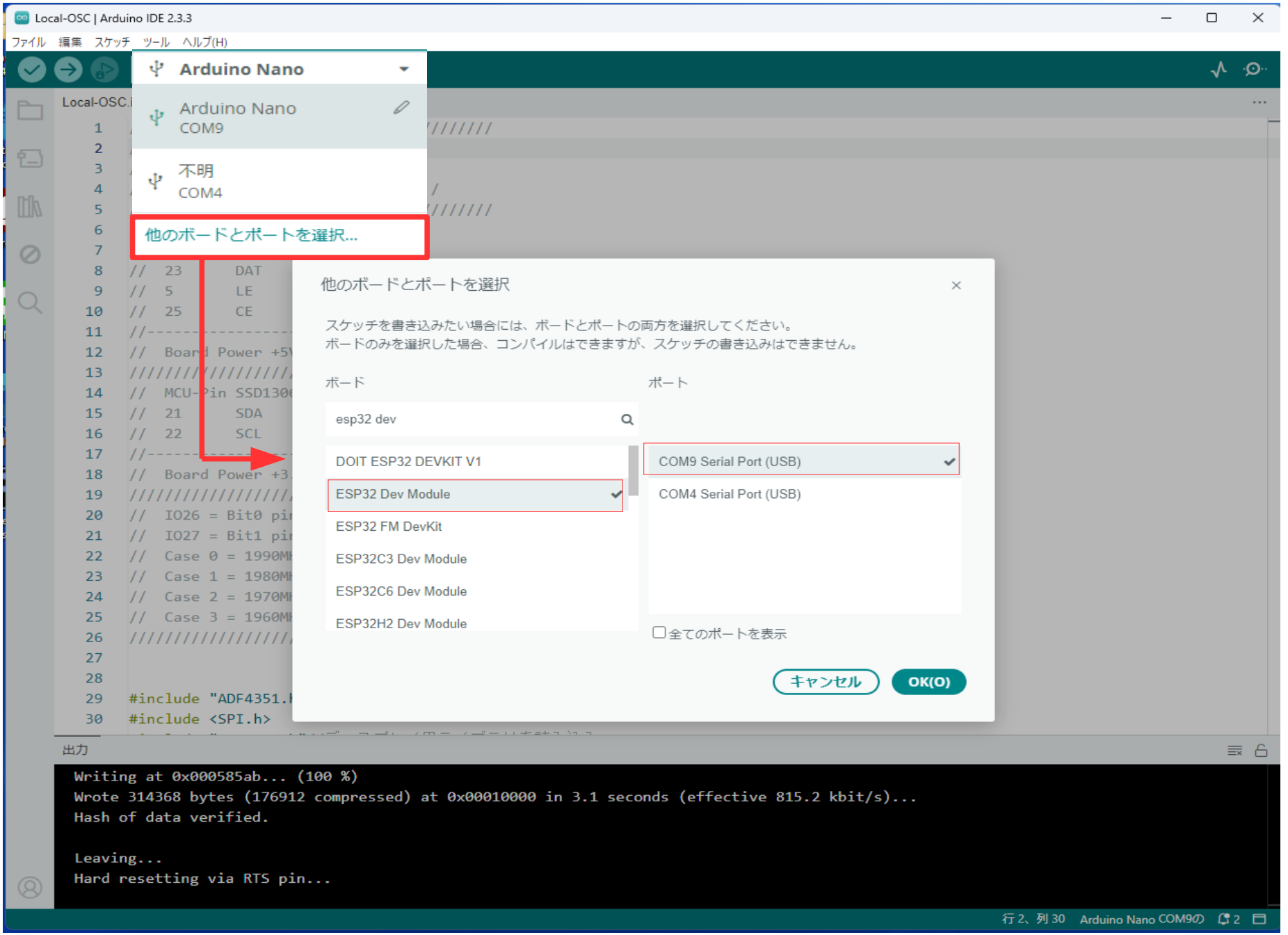
```
 void setup() {
  Serial.begin(115200);
  Serial.println("Program started!");
  adf4351.begin();

  display.init();    //ディスプレイを初期化
  display.setFont(ArialMT_Plain_16);    //フォントを設定 ArialMT_Plain_10, ArialMT_Plain_16, ArialMT_Plain_24
  display.drawString(0, 0, "Spot Oscillator");    //(0,0)の位置にWiFi connectedを表示
  display.drawString(0, 48, "JA4CXX");

//発振周波数（2282MHz）<--- 2427MHz - 2282MHz = 145MHz
      adf4351.WriteRegister(0x00580005);  // -R5- default value LD working mode
      adf4351.WriteRegister(0x8C803C);    // -R4- output divider = auto // band select clock divider
      adf4351.WriteRegister(0x000004B3);  // -R3- default Antibacklash pulse width
      adf4351.WriteRegister(0x00010E42);  // -R2- Low noise mode, R Counter=4
      adf4351.WriteRegister(0x80080C9);   // -R1- prescaler=8/9, MOD=2
      adf4351.WriteRegister(0x2D8038);    // -R0- int = 91 frac = 7
      Serial.println("All registers have benn written!");
      display.drawString(0, 24, "Freq: 2282MHz");
      Serial.println("Freq: 2282MHz");
      display.display();   //指定された情報を描画
}

 void loop() {
   // put your main code here, to run repeatedly:
}
```

Arduino IDE　プログラム統合開発環境

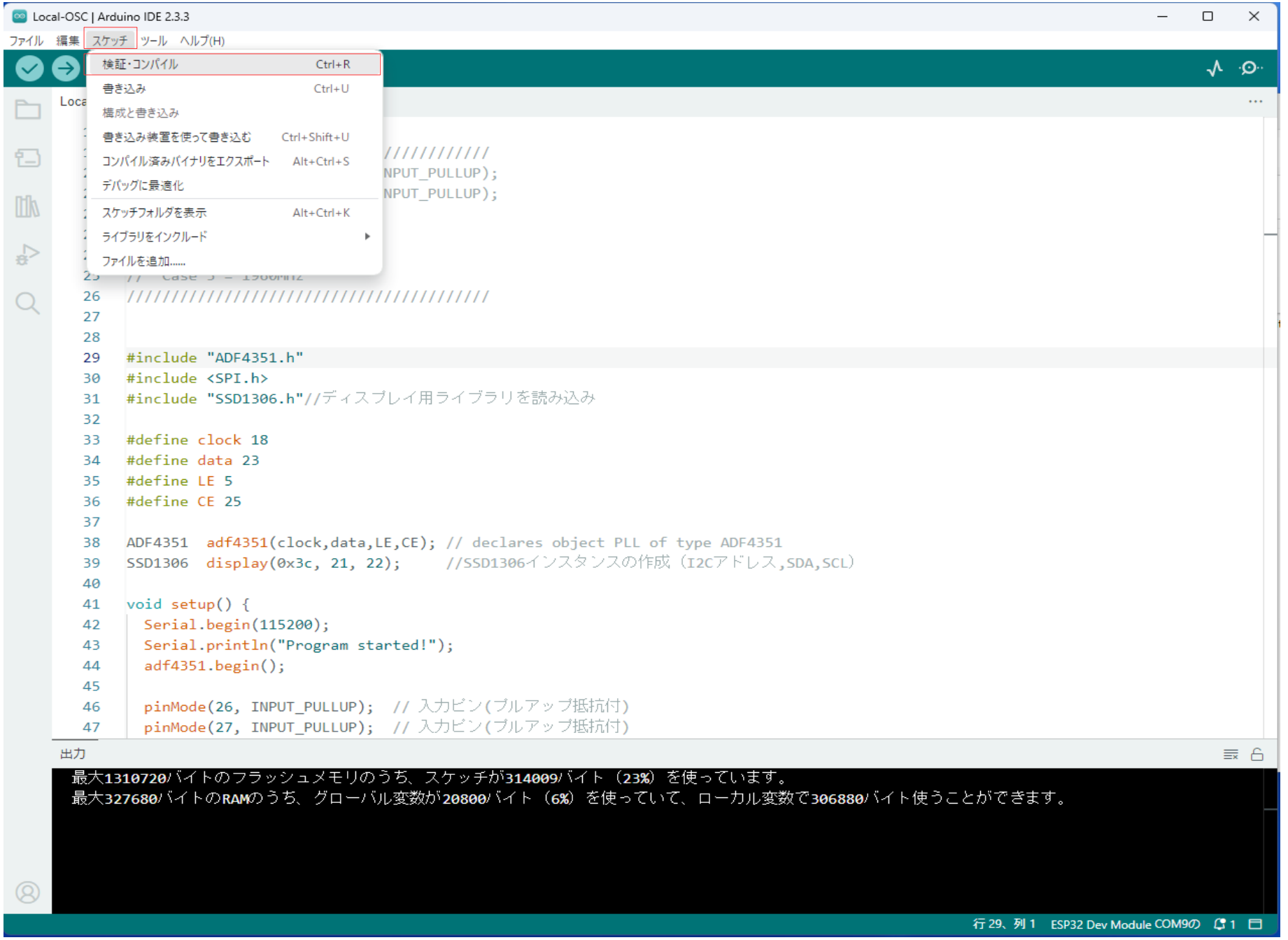


起動時

Arduino IDE プログラム統合開発環境 ー ボードとポートを選択



Arduino IDE ― プログラムをコンパイルしてバグ潰し



Arduino IDE ― プログラムを基板に書き込みます