

RL78/G23

タイマ機能 サンプルスケッチ (Arduino™ スケッチ)

要旨

本アプリケーションノートでは、RL78/G23-64p Fast Prototyping Board (FPB) 用 Arduino ライブラリのタイマ機能を用いて、任意の時間経過後に圧電サウンダでメロディを再生させる制御方法を説明します。

動作確認デバイス

評価ボード : RL78/G23-64p Fast Prototyping Board

商標・他社 TM

Arduino は Arduino SA の商標です。

目次

1. システム概要	3
1.1 圧電サウンダ	4
1.2 I ² C 接続キャラクタ LCD モジュール	4
2. 動作確認環境	5
3. 開発環境構築	6
3.1 ボードの接続	6
3.2 使用端子一覧	7
3.3 Arduino™ IDE のセットアップ	8
4. ソフトウェア説明	11
4.1 サンプルコードの概要	11
4.2 サンプルコードに使用するスケッチ例	12
4.2.1 MTimer2 / FlashLed	12
4.2.2 Wire / i2c-lcd	13
4.3 API 関数	14
4.3.1 I ² C 接続キャラクタ LCD モジュール制御関数	15
4.4 サンプルスケッチの動作確認手順	16
4.5 フローチャート	19
4.5.1 メイン処理フローチャート	19
4.5.2 呼び出し関数処理フローチャート	21
4.5.3 関数詳細	24
4.6 応用例	25
4.6.1 設定時間変更	25
4.6.2 カウント終了メロディ変更	25
5. 注意事項	26
5.1 Windows デバイス マネージャーに COM ポートが表示されない	26
5.2 RL78/G23-64p Fast Prototyping Board に正しく書き込みができない	27
6. サンプルコード	28
7. 参考ドキュメント	28
改訂記録	29

1. システム概要

本システムは、RL78/G23-64p Fast Prototyping Board、圧電サウンダ、I²C 接続キャラクタ LCD モジュールで構成されています。プログラムの作成と書き込みには Arduino™ IDE を使用します。また本システムでは、FPB のスイッチ(SW1)押下後に LCD モジュールに残り時間をカウントダウンで表示し、残り 0 秒になると圧電サウンダからメロディを再生します。

本システムで使用するサンプルコードのブロック構成を以下に示します。

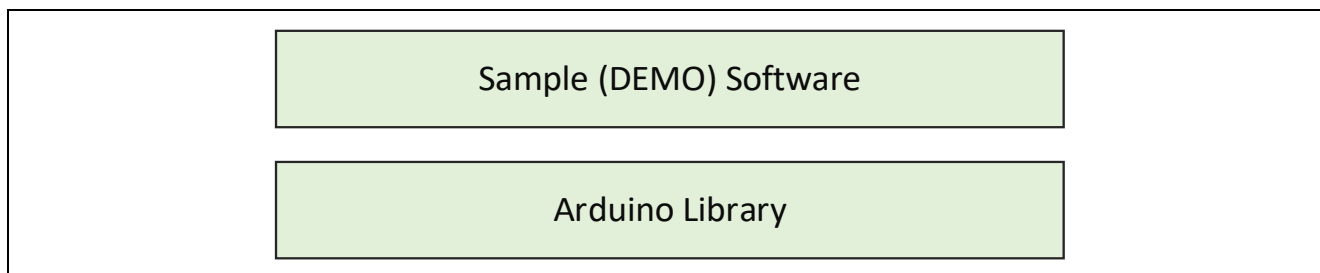


図 1-1 ソフトウェアブロック図

本システム構成の概略図を以下に示します。

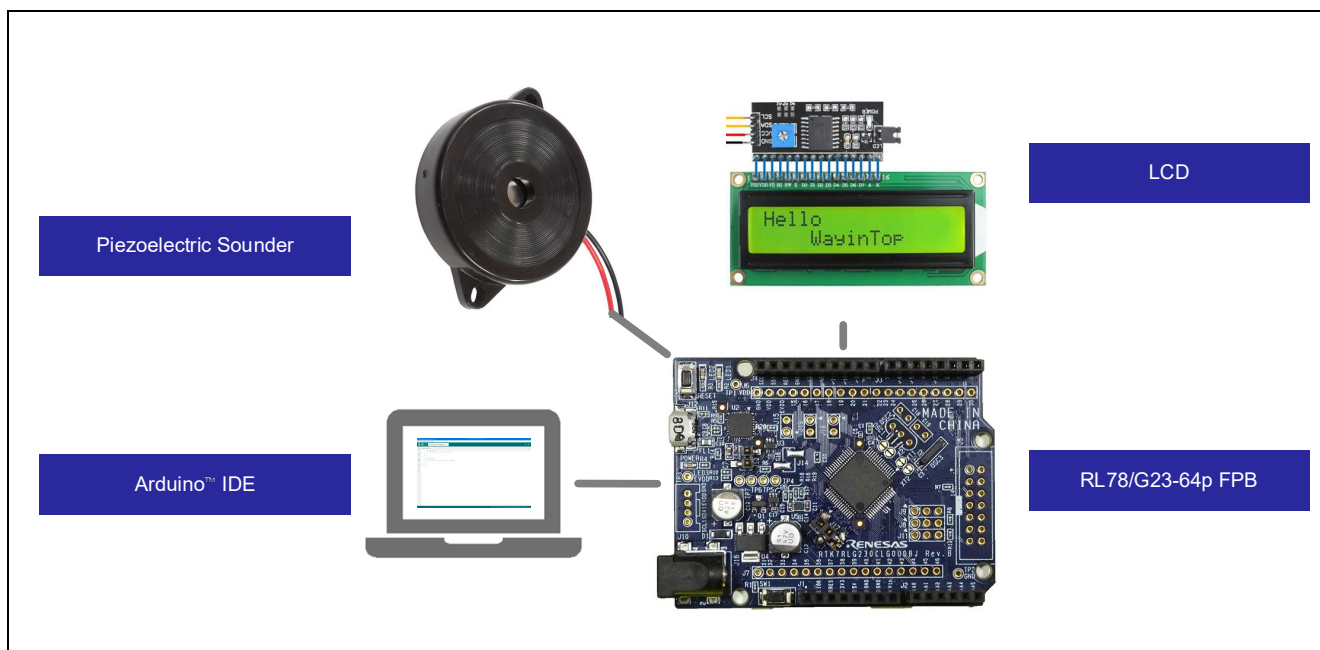


図 1-2 システム構成図

1.1 圧電サウンダ

圧電サウンダは、入力される信号に応じて発音する発音部品で、本システムでは、FPB からの特定周波数の矩形波を供給することで、任意の音を発生させます。

圧電サウンダを制御する矩形波の周波数は以下の通りです。

表 1-1 圧電サウンダで使用する矩形波の周波数

音名	周波数[Hz]
C (ド)	523
D (レ)	587
E (ミ)	659
F (ファ)	698
G (ソ)	784
A (ラ)	880
B (シ)	988

1.2 I²C 接続キャラクタ LCD モジュール

I²C 接続キャラクタ LCD モジュールは、FPB から I²C 通信でコマンドを送信することで LCD に任意の文字を表示させることができます。本システムでは Arduino IDE の RL78/G23-64p Fast Prototyping Board 用のスケッチ例で使用している LCD モジュールを使用しています。

以下は、本システムで使用する液晶モジュールです。



図 1-3 液晶モジュール

本システムでは使用する端子数が 2 本で済む I²C 接続を使用していますが、Arduino では直接 LCD ドライバを制御してキャラクタ表示できるスケッチ例も用意されています。制御に必要な端子数は増えますが、同様の表示が実現可能です。

2. 動作確認環境

本システムの動作確認環境は、以下のとおりです。

表 2-1 動作確認環境（ハードウェア）

項目	内容
評価ボード	RL78/G23-64p Fast Prototyping Board – RTK7RLG230CLG000BJ
圧電サウンダ	SPT15
I ² C 接続キャラクタ LCD モジュール	WayinTop 1602
動作電圧	5V

表 2-2 動作確認環境（ソフトウェア）

項目	内容	バージョン
OS	Windows 10	-
統合開発環境（IDE）	Arduino™ IDE	2.3.2
標準ライブラリ	RL78/G23-64p FPB ライブラリ	2.3.1

3. 開発環境構築

ボードの接続方法と Arduino™ IDE のセットアップを説明します。

本システムでは Arduino™ IDE 2.3.2 を使用しています。Arduino™ IDE 2.3.2 以降をインストールしていない場合は、インストールしてください。

<https://www.arduino.cc/en/software>

3.1 ボードの接続

図 3-1 のように PC と評価ボードを USB で接続します。また、評価ボードと圧電サウンダ、I²C 接続キャラクタ LCD モジュールをそれぞれジャンパワイヤで接続します。

本システムでは、評価ボードへの電源供給は USB を使用します。評価ボードの回路を確認し、必要に応じてジャンパを設定してください。

本システムでは、評価ボードのジャンパを以下のように設定します。

表 3-1 評価ボードのジャンパ設定

ジャンパ	設定	機能
J8	1-2 ショート	COM port デバッグ使用
J9		
J11		
J13	オープン	マイコンへの 5V 電源供給
J17	1-2 ショート	

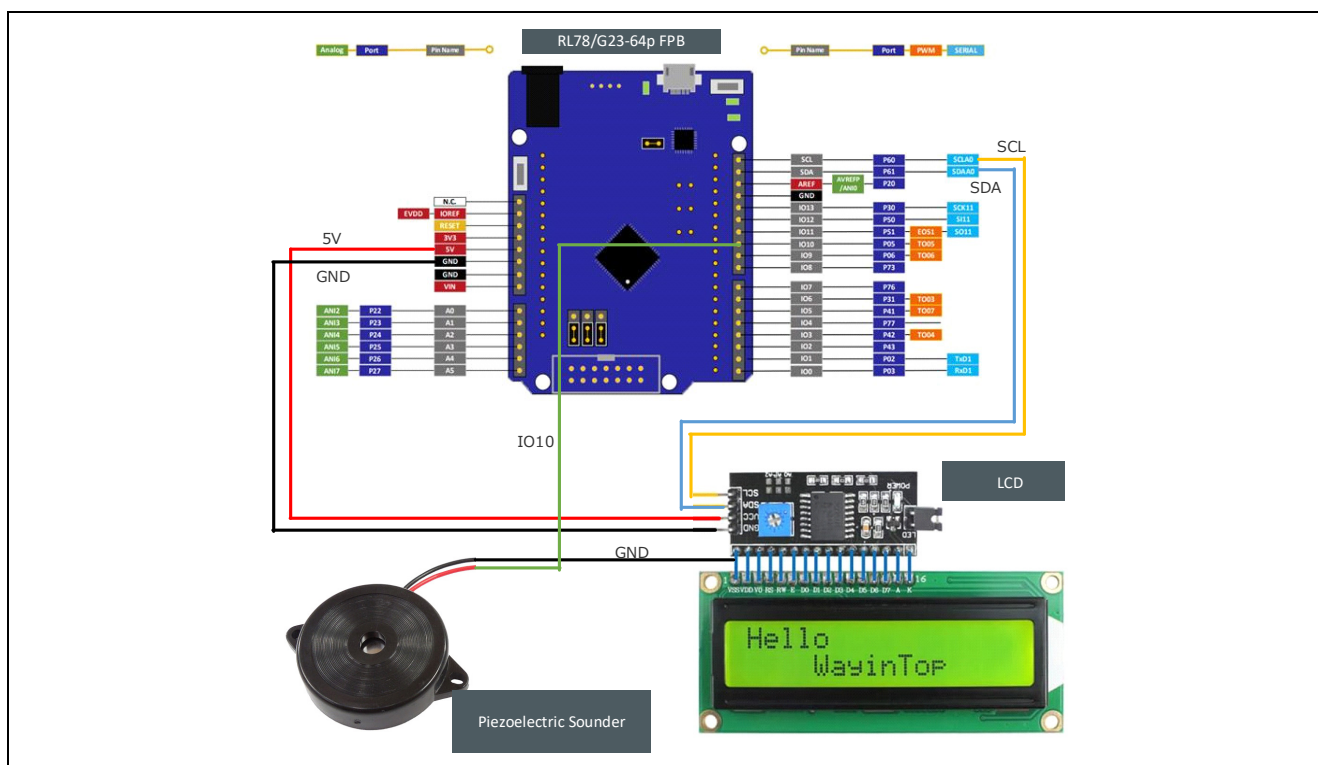


図 3-1 PC と評価ボード、圧電サウンダ、I²C 接続キャラクタ LCD モジュールと電源の接続

3.2 使用端子一覧

本システムの使用端子を以下に示します。

表 3-2 本システムの使用端子一覧

項目	Arduino™ 信号名	マイコンの端子番号	端子
I2C	SDA	18	P61
	SCL	17	P60
Tone	IO10	31	P05
SW1	IO26	9	P137/INTP0
VDD	5V	-	-
GND	GND	14	-

各ボードの詳細な端子の説明は、以下のマニュアルを参照してください。

- RL78/G23-64p Fast Prototyping Board ユーザーズマニュアル (R20UT4814)

3.3 Arduino™ IDE のセットアップ

本章で Arduino™ のセットアップ手順を説明します。

備考. セットアップ手順は、[クイックスタートガイド](#)・[renesas/Arduino Wiki](#)・[GitHub](#)に記載されている手順と同様です。また上記サイトでは、LED を点滅させるサンプルスケッチが記載されています。必要に応じて、参照してください。

1. Arduino™ IDE を起動します。
2. [ツール] – [ボード:] – [ボードマネージャ] を選択します。

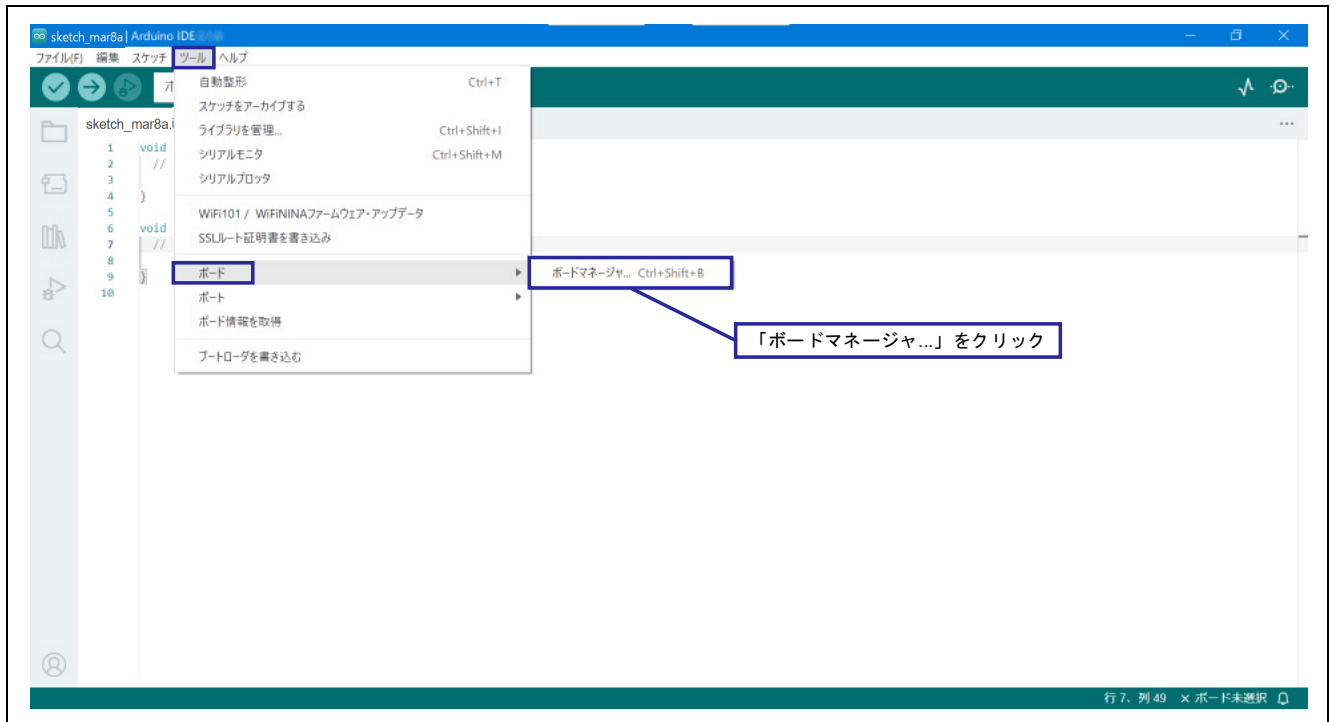


図 3-2 ボードマネージャの選択

3. [タイプ] は“全て”を選択し、検索欄に“RL78/G23”と入力し、表示された [RL78/G23-64p Fast Prototyping Board] の [インストール] をクリックします。

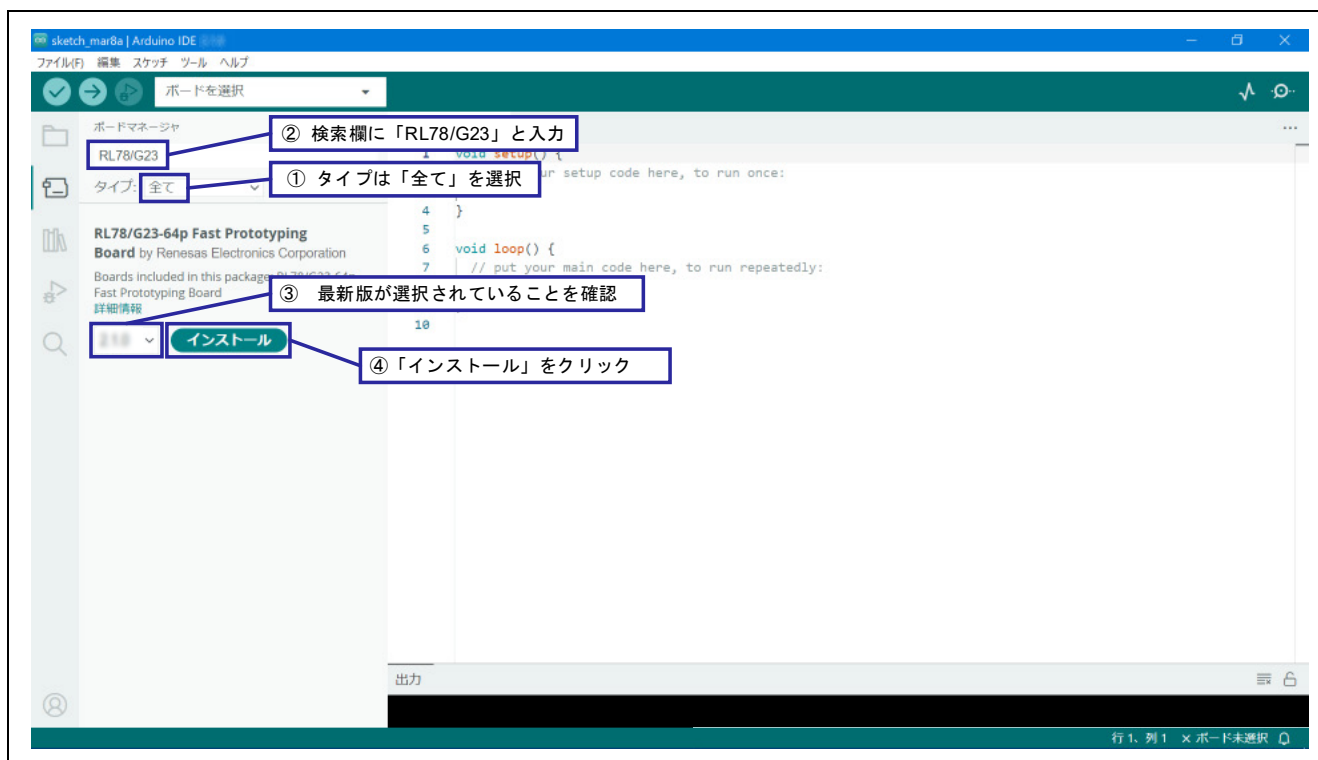


図 3-3 ボードマネージャのインストール

4. [ツール] - [ポート] から評価ボードに割り当てられたシリアルポートを選択します。
COM ポート番号は、Windows のデバイス マネージャーから確認できます。

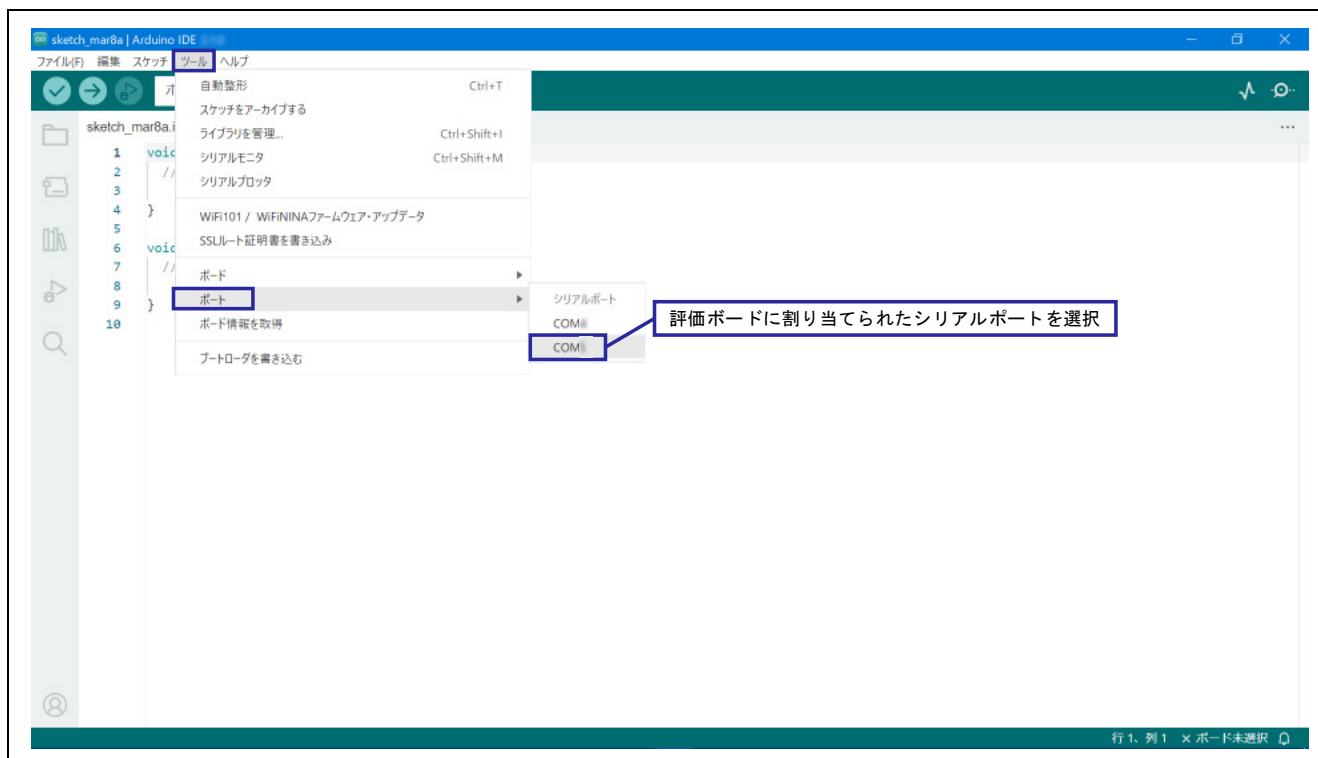


図 3-4 シリアルポートの選択

5. [ツール] – [ボード:] – [RL78/G23-64p Fast Prototyping Board] – [RL78-G23 Fast Prototyping Board 64Pin] を選択します。

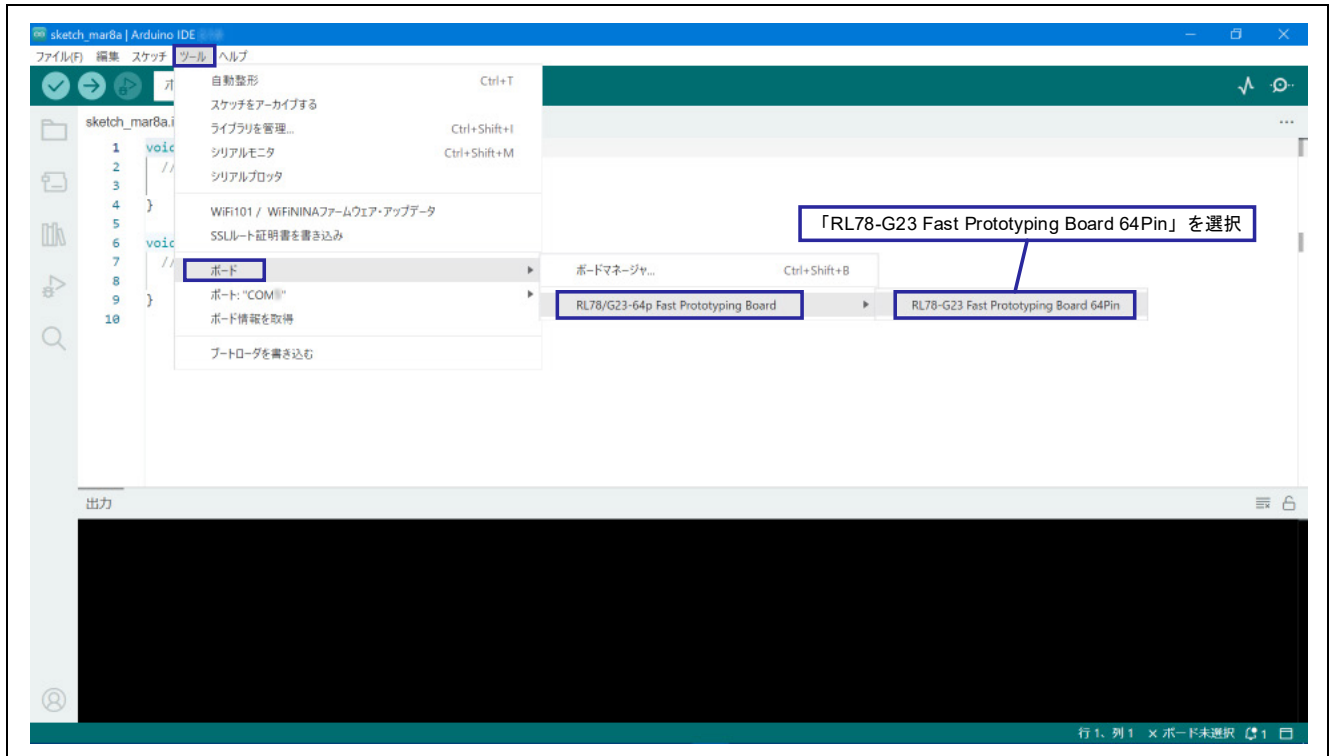


図 3-5 ボードの選択

4. ソフトウェア説明

4.1 サンプルコードの概要

本サンプルコードは、MSTimer2 を使用しサウンドを再生するまでの残り時間をカウントダウンする周期処理、Tone を使用しメロディを再生する処理、Wire を使用し I²C 接続キャラクタ LCD モジュールに残り時間を表示する処理で構成された Arduino™ IDE 上で実行するサンプルスケッチになります。ファイル構成を以下に示します。

使用する API 関数は「4.3 API 関数」を、サンプルスケッチの詳細は「4.4 サンプルスケッチの動作確認手順」を参照してください。

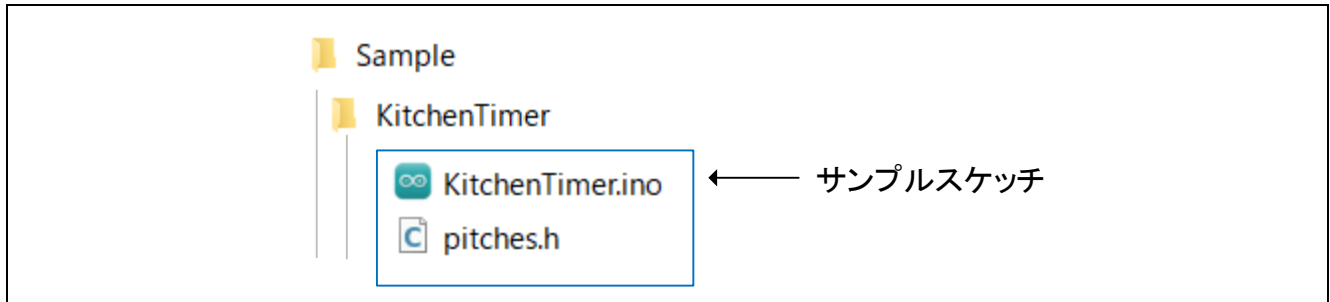


図 4-1 サンプルコードのファイル構成

4.2 サンプルコードに使用するスケッチ例

本サンプルコードでは、Arduino™ IDE で提供されているスケッチ例を流用しています。スケッチ例の参照方法を以下に示します。

4.2.1 MTimer2 / FlashLed

MTimer2 を使用して、RL78/G23-64p Fast Prototyping Board の LED を 0.5 秒間隔で点滅させるスケッチ例です。以下の手順で参照します。

1. Arduino™ IDE を起動します。
2. [ファイル] – [スケッチ例] – [MTimer2] – [FlashLed] を選択します。

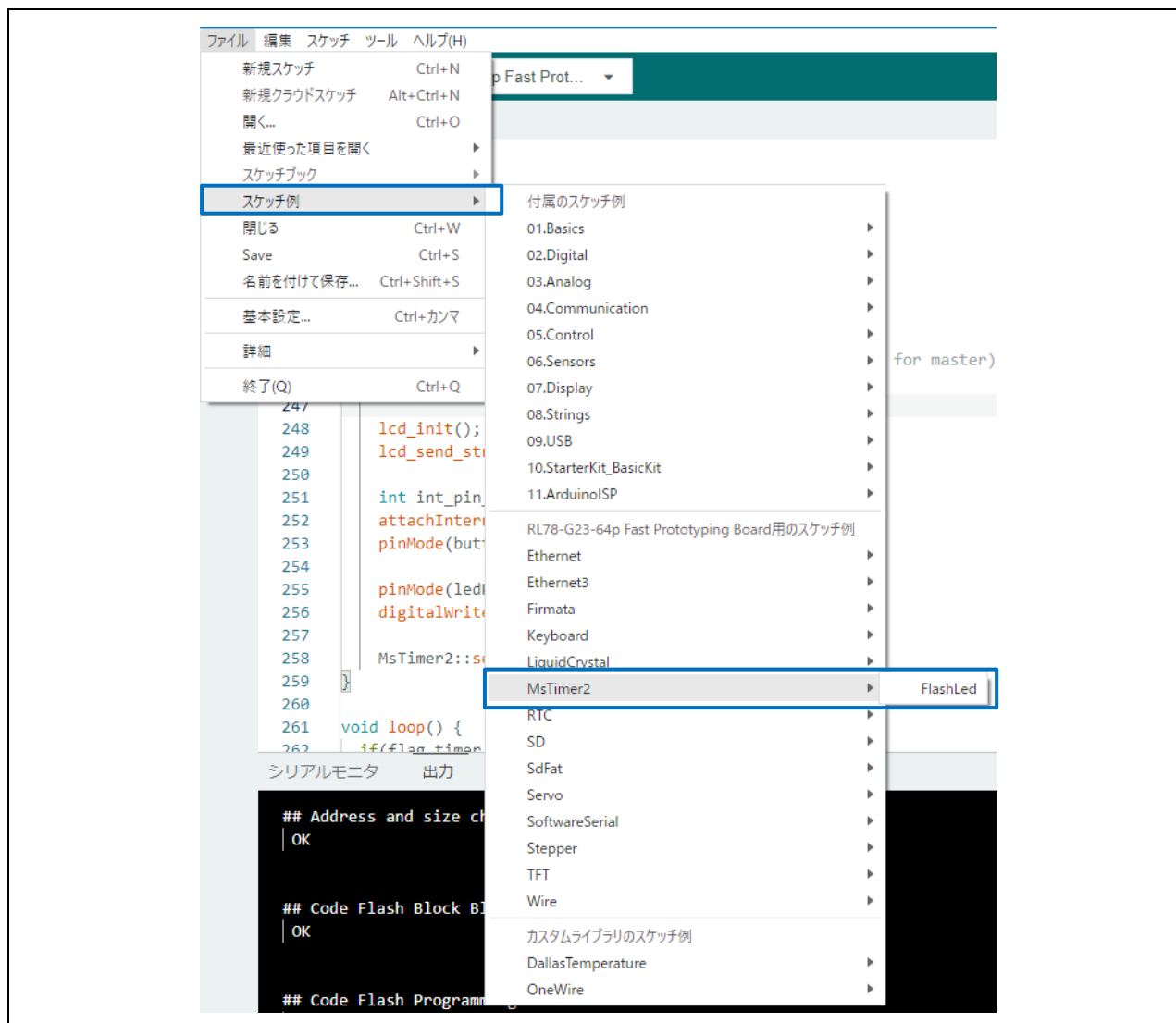


図 4-2 MTimer2/FlashLed の選択

4.2.2 Wire / i2c-lcd

Wire を使用して、I²C 接続キャラクタ LCD モジュールに「Hello RL78!」を表示させるスケッチ例です。以下の手順で参照します。

1. Arduino™ IDE を起動します。
2. [ファイル] – [スケッチ例] – [Wire] – [i2c-lcd] を選択します。

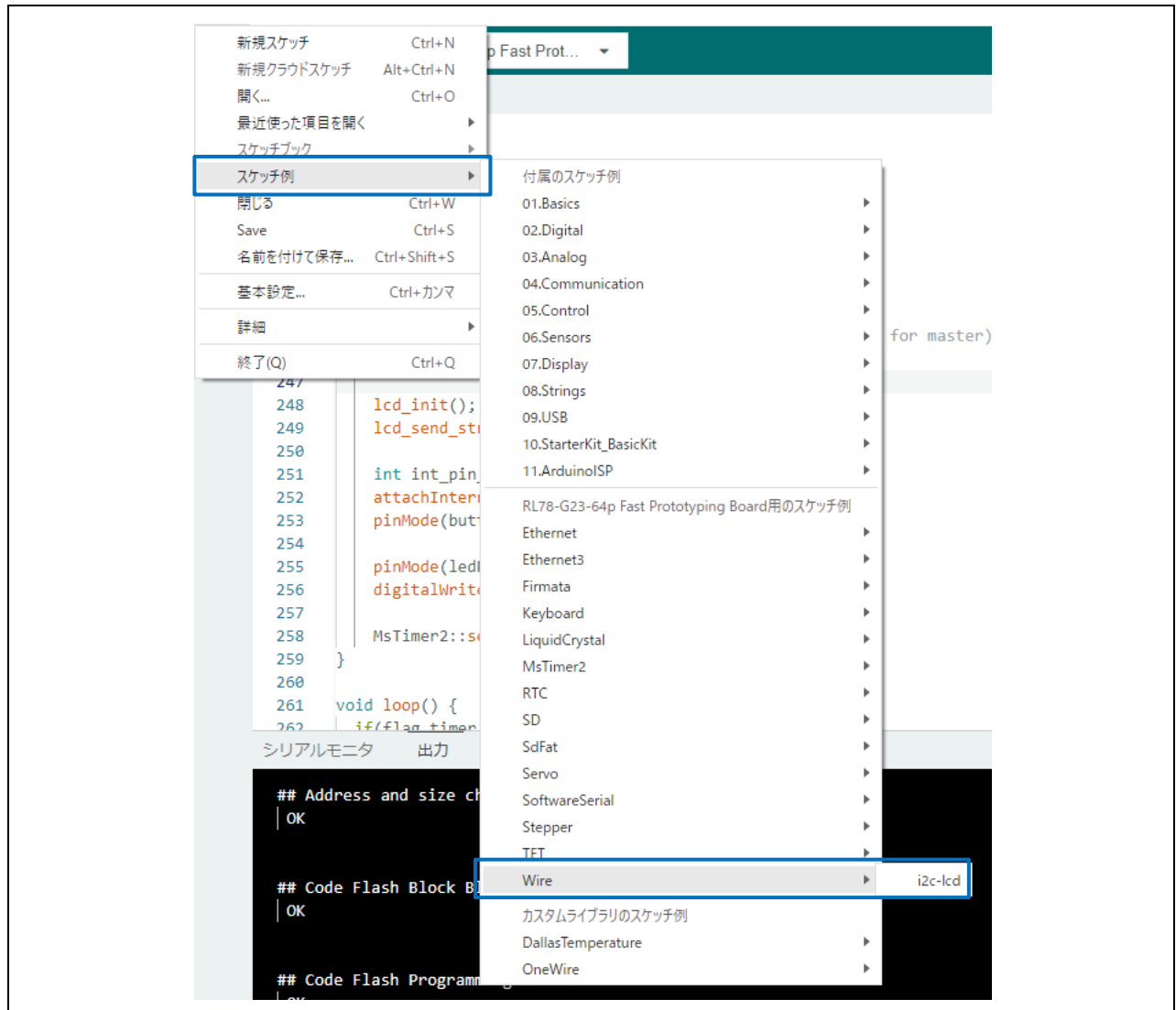


図 4-3 Wire/i2c-lcd の選択

4.3 API 関数

本サンプルコードでは、digitalWrite 関数と、Tone 関連の関数、attachInterrupt 関数と MsTimer2 ライブラリ、Wire ライブラリを使用しています。以下に各ライブラリの API 関数を示します。

表 4-1 使用関数一覧

API 関数名	機能
digitalWrite(pin,value)	デジタル端子に HIGH/LOW を出力
tone(pin, frequency, duration)	デジタル端子に指定された周波数の矩形波を出力
noTone()	矩形波の生成を停止
Wire.begin(address)	Wire ライブラリを初期化 本サンプルでは I2C バスにマスタとして接続
Wire.setClock(clockFrequency);	I2C 通信のクロック周波数を変更
Wire.beginTransmission(address)	相手先に対して送信を開始するための準備
Wire.write(value)	送信バッファの末尾にデータを追加
Wire.endTransmission()	相手先に送信シーケンスを発行し、I2C 通信開始
MsTimer2::set(ms, function)	指定した周期間隔 (ms) で処理する関数を登録します。
MsTimer2::start()	set 関数で登録した周期処理を開始します。
MsTimer2::stop()	実行中の周期処理を停止します。
attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)	外部端子割り込みのモード設定とサービスルーチンの登録

各ライブラリの関数仕様は以下の Arduino™ ホームページを参照してください。

[API リスト](#)・[renesas/Arduino Wiki](#)・[GitHub](#)

[digitalWrite\(\)](#) - [Arduino Reference](#)

[tone\(\)](#) - [Arduino Reference](#)

[noTone\(\)](#) - [Arduino Reference](#)

[Wire](#) - [Arduino Reference](#)

[MsTimer2](#) - [Arduino Reference](#)

[attachInterrupt\(\)](#) - [Arduino Reference](#)

4.3.1 I²C 接続キャラクタ LCD モジュール制御関数

I²C 接続キャラクタ LCD モジュールを制御する関数を以下に示します。

(1) LCD モジュール初期化

内容	LCD モジュールを初期化します。	
使用 API 関数	lcd_init ()	
使用例	lcd_init ()	LCD モジュールを初期化

(2) LCD モジュール制御コマンド送信

内容	LCD モジュールを制御するコマンドを送信します。	
使用 API 関数	lcd_send_cmd(cmd)	
cmd	制御コマンド	
使用例	lcd_send_cmd (0x01)	LCD 表示をクリアするコマンドを送信

(3) LCD モジュール表示データ送信

内容	LCD モジュールに表示するキャラクタデータを送信します。	
使用 API 関数	lcd_send_data(data)	
data	キャラクタデータ	
使用例	lcd_send_data (0x30)	LCD に"0"を表示させるデータを送信

(4) LCD モジュール表示クリア

内容	LCD モジュール表示をクリアします。	
使用 API 関数	lcd_clear()	
使用例	lcd_clear ()	LCD 表示をクリア

(5) LCD 表示位置指定

内容	LCD の文字表示位置を指定します。	
使用 API 関数	lcd_put_cur(row, col)	
row	縦軸の表示位置	
col	横軸の表示位置	
使用例	lcd_put_cur(0, 3)	1 行目の 4 文字目を指定

(6) LCD モジュール表示文字列送信

内容	LCD モジュールに表示する文字列データを送信します。	
使用 API 関数	lcd_send_string(str)	
str	表示文字列	
使用例	lcd_send_str ("Hello")	LCD に「Hello」を表示

4.4 サンプルスケッチの動作確認手順

本サンプルスケッチの動作確認手順を以下に示します。

事前に「3.3 Arduino™ IDE のセットアップ」を行ってください。

1. [ファイル] – [開く...]をクリックし、サンプルスケッチ KitchenTimer.ino を開きます。

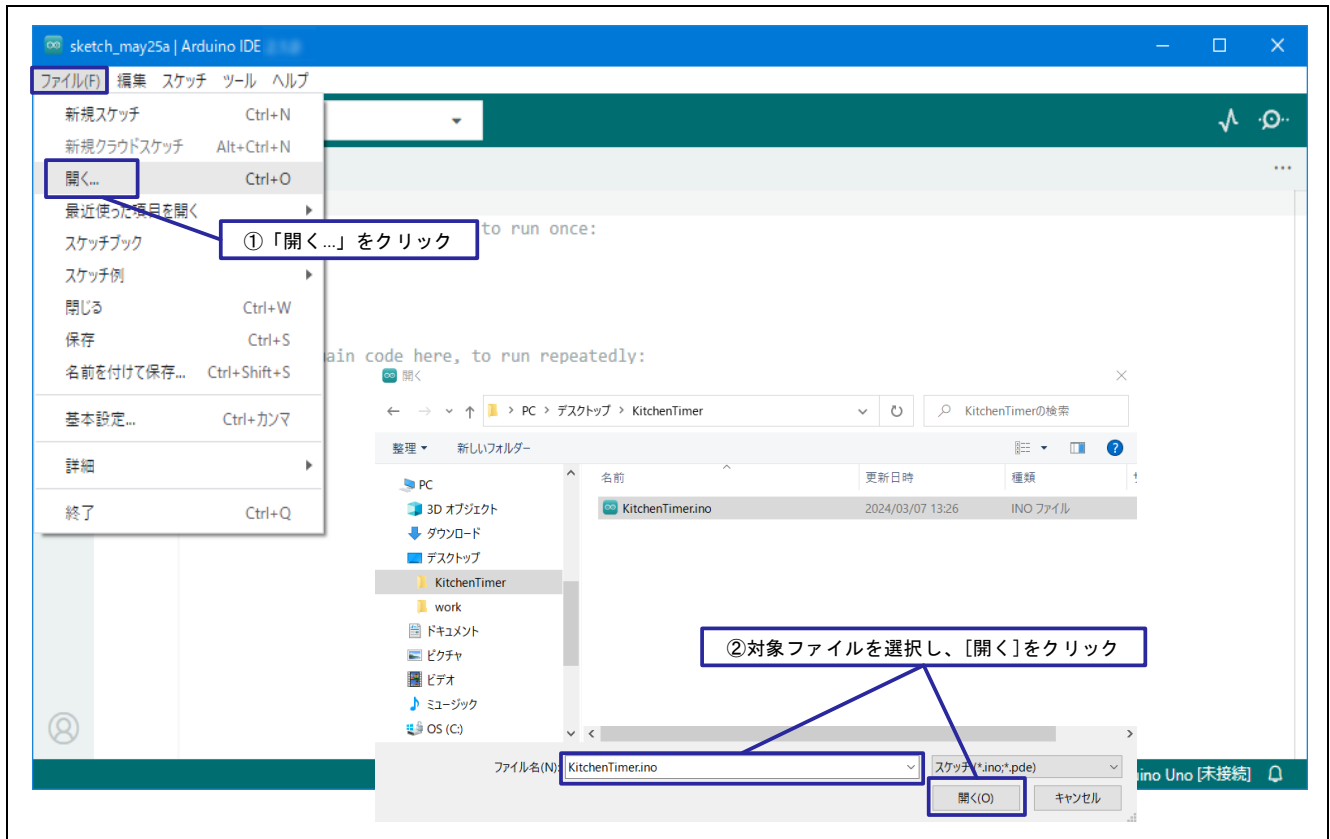


図 4-4 サンプルスケッチを起動

2. 検証のアイコンをクリックし、スケッチをコンパイルします。

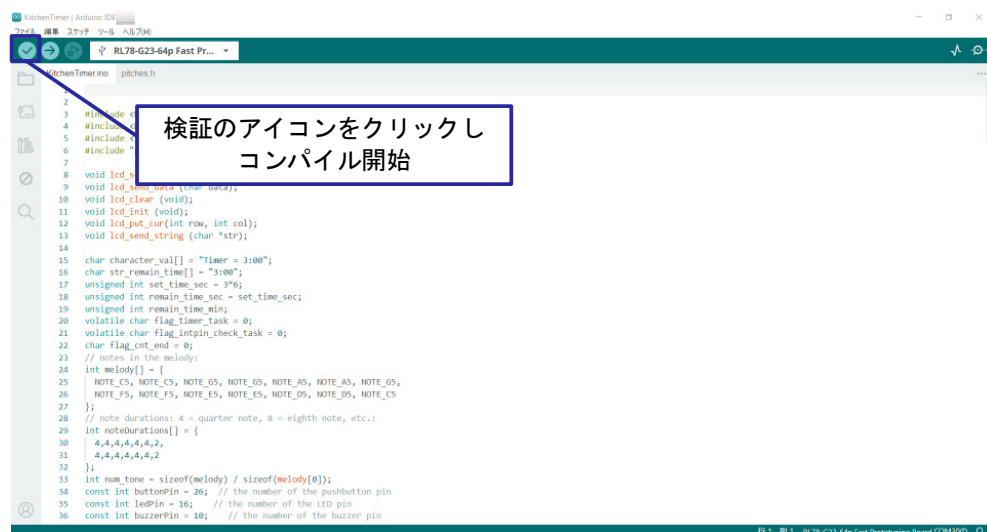


図 4-5 スケッチのコンパイル

3. コンパイル完了後、書き込みのアイコンをクリックし、デバイスに書き込みます。

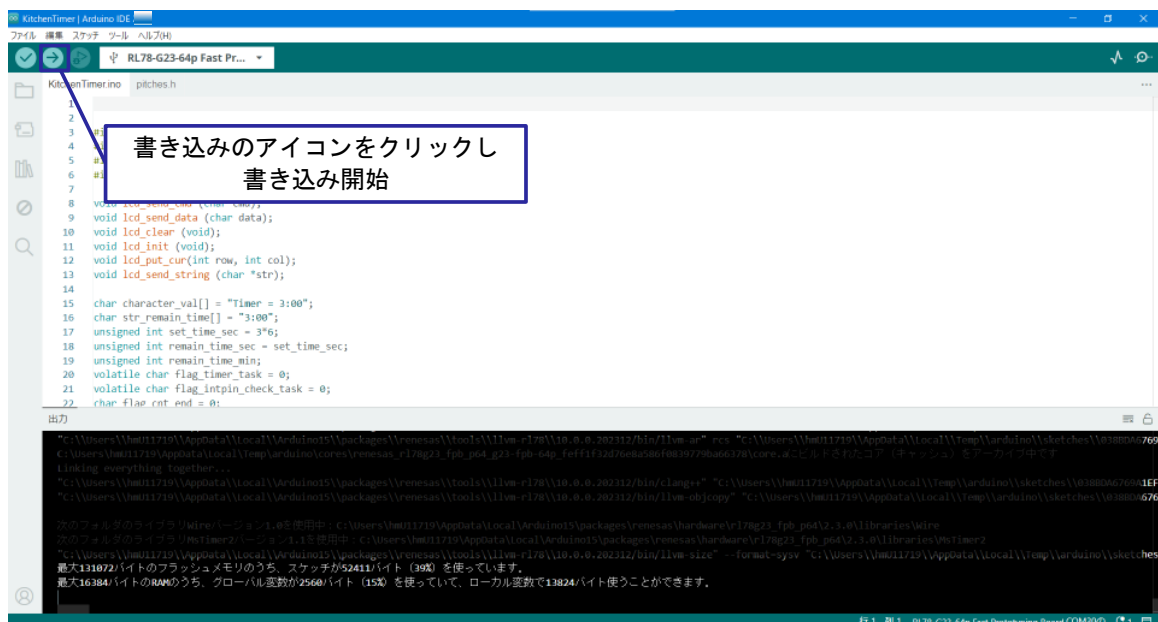


図 4-6 スケッチの書き込み

- 書き込み完了後、マイコンが動作を開始すると LCD モジュールに"Timer = 3:00"の文字が表示されます。



図 4-7 LCD モジュールの初期表示

- マイコンボード上の SW1 を押下すると、カウント動作を開始します。



6. 図 4-8 LCD モジュールのカウントダウン途中の表示

4.5 フローチャート

4.5.1 メイン処理フローチャート

以下にサンプルスケッチの処理フローを示します。

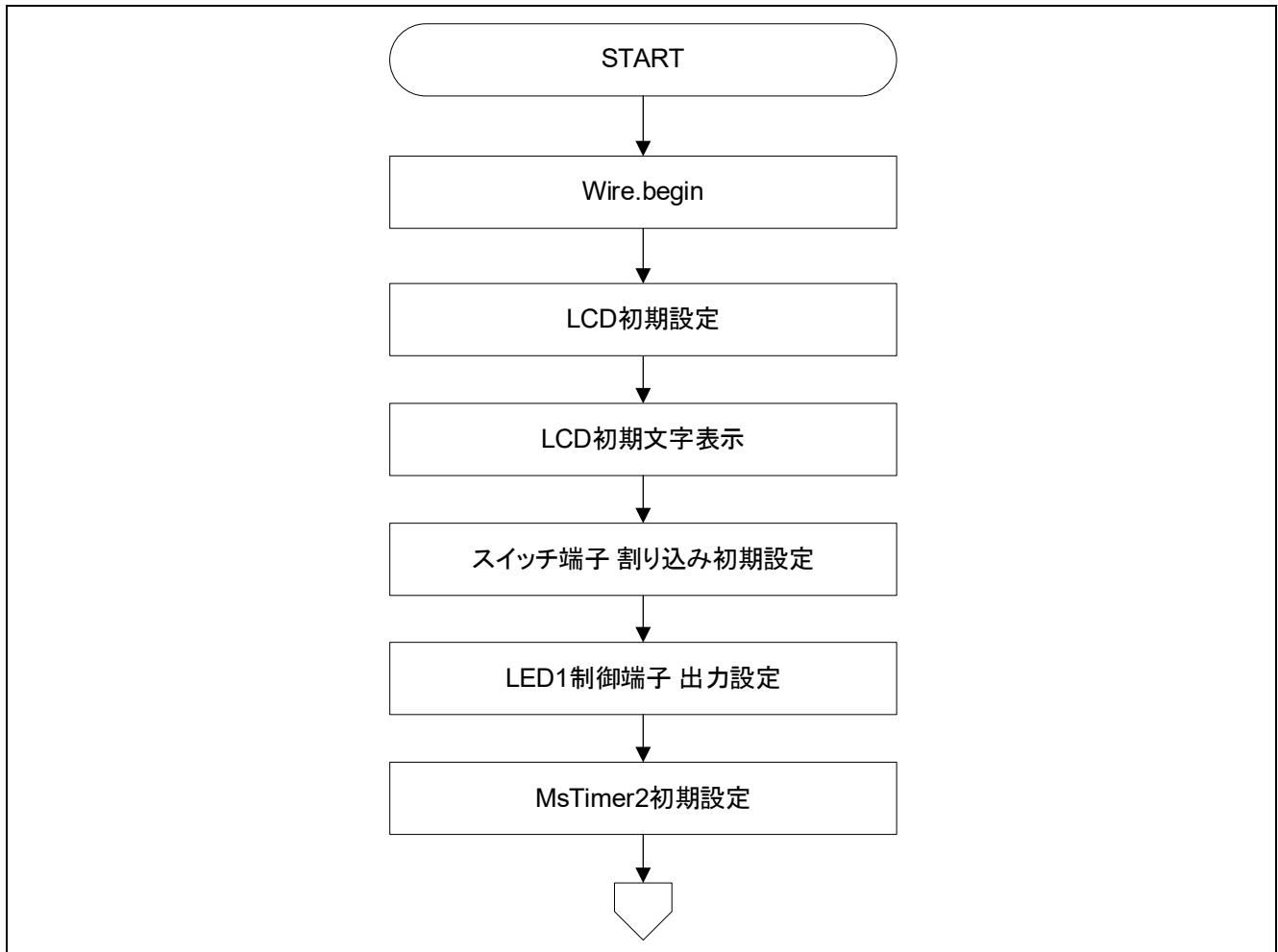


図 4-9 メイン処理フローチャート(1/2)

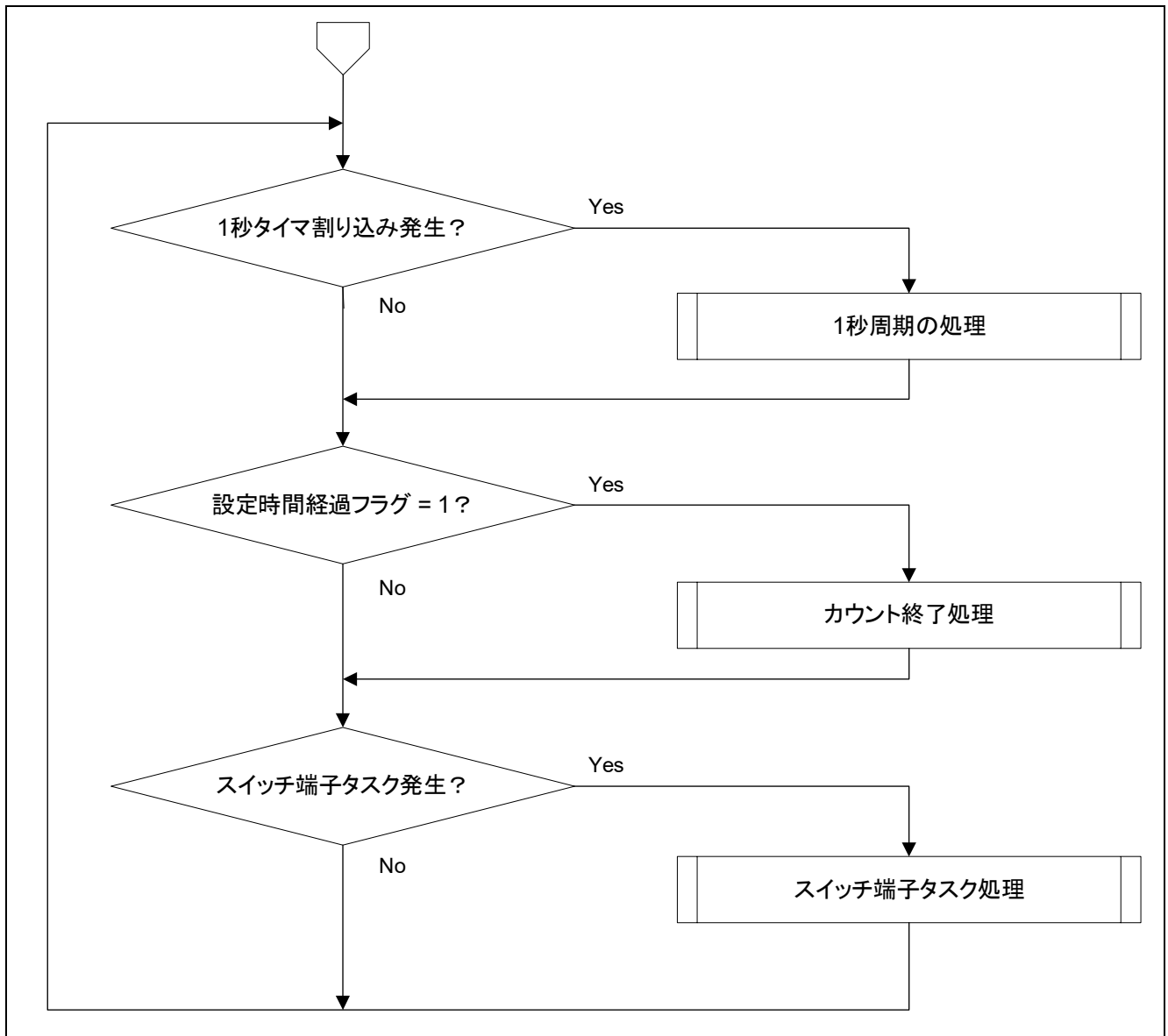


図 4-10 メイン処理フローチャート(2/2)

4.5.2 呼び出し関数処理フローチャート

以下に loop 関数から呼び出す関数の処理フローを示します。

(1) timerTask 関数：

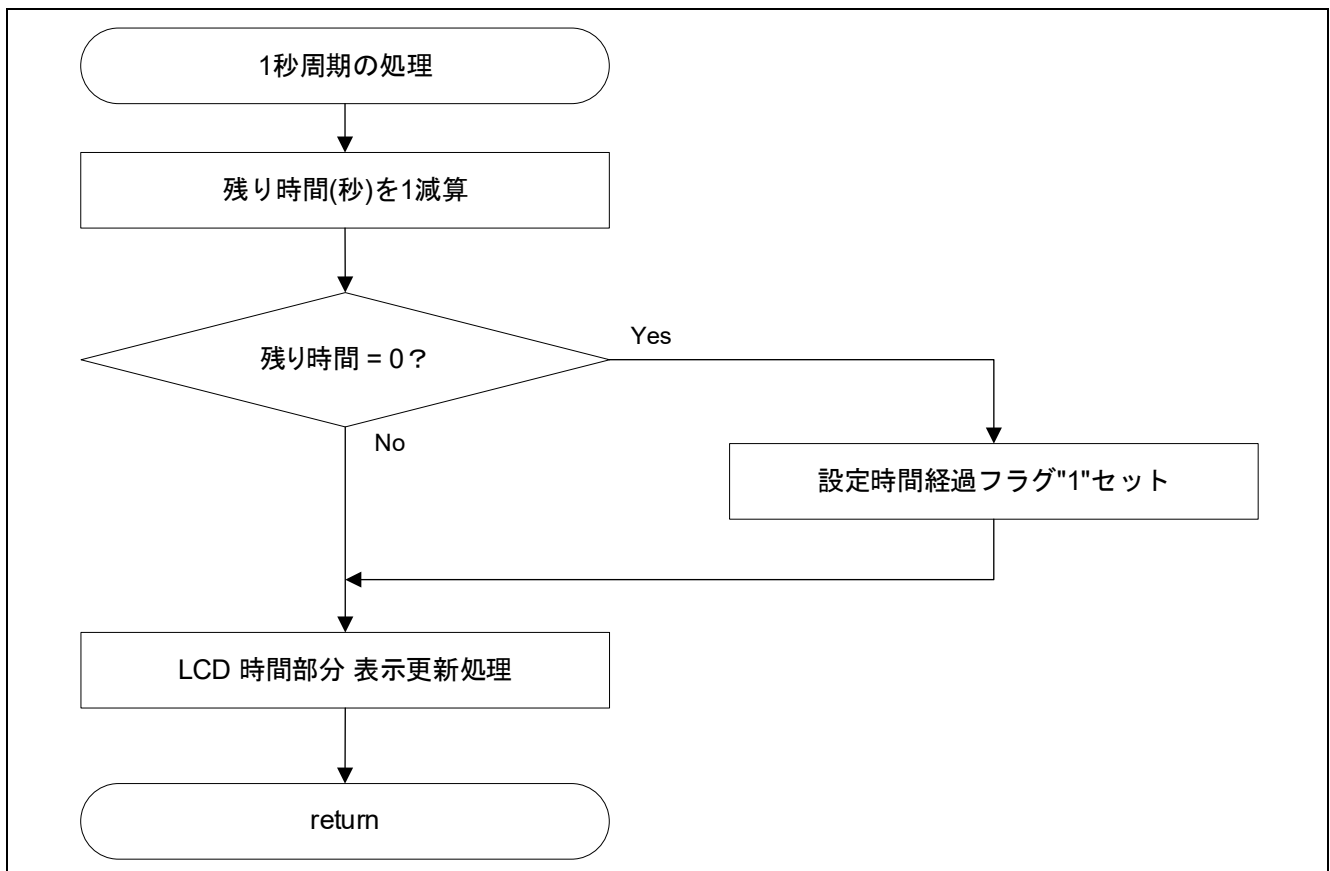


図 4-11 timerTask 関数のフローチャート

(2) timerEnd 関数 :



図 4-12 timerEnd 関数のフローチャート

(3) intpinCheck 関数 :

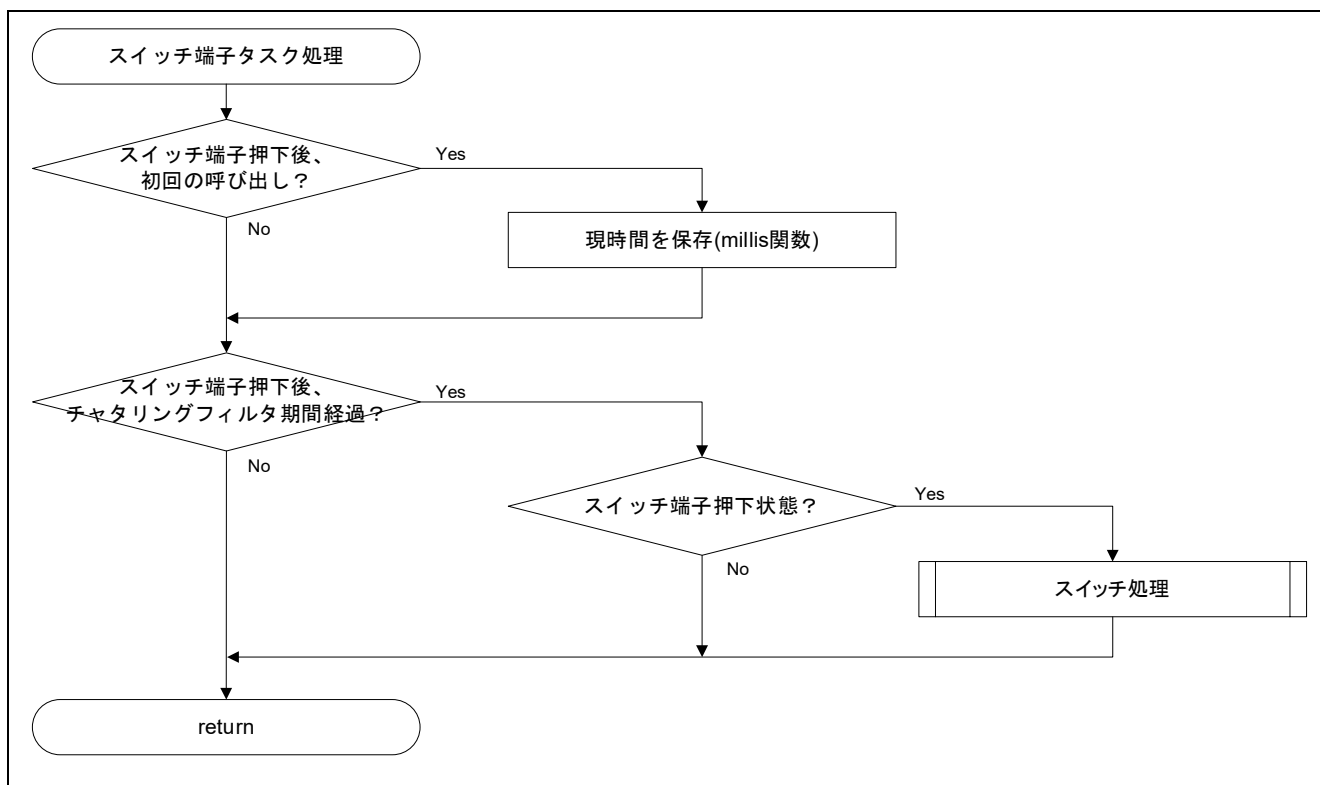


図 4-13 intpinCheck 関数のフローチャート(1/2)

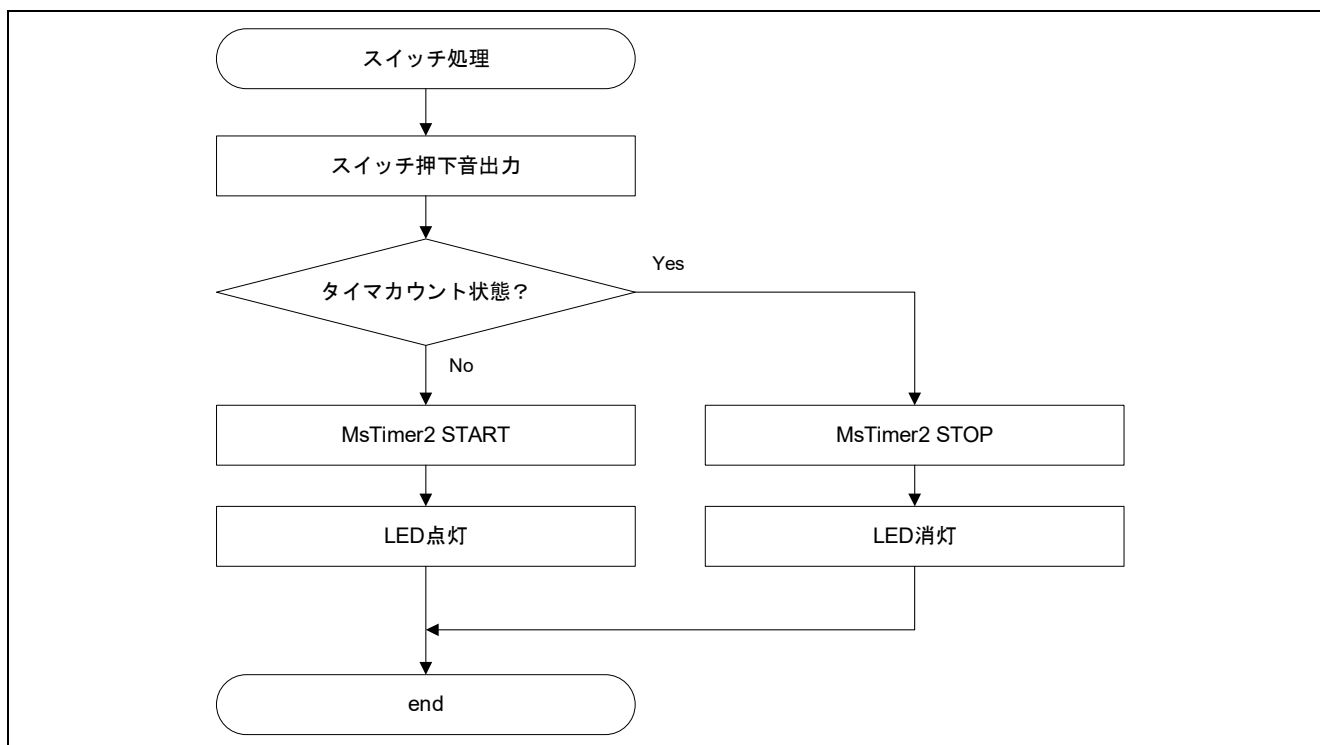


図 4-14 intpinCheck 関数のフローチャート(2/2)

4.5.3 関数詳細

呼び出し関数の詳細を以下に示します。

void timerTask (void)

概要	タイマーカウント動作の残り時間の減算と残り時間を LCD に表示する
引数	なし
戻り値	なし

void timerEnd (void)

概要	タイマーカウントの終了動作 カウント終了を示すメロディを出力し、LCD にカウント終了を表すメッセージを表示する
引数	なし
戻り値	なし

void intpinCheck (void)

概要	スイッチ押下時の処理 スイッチ押下のチャタリング除去処理をし、スイッチ押下時の音を出力、 タイマのカウント動作停止/開始を制御する
引数	なし
戻り値	なし

4.6 応用例

4.6.1 設定時間変更

外部からデータを入力したり操作をしたりする機能を実装することで、「設定時間を変更」の追加機能を実装可能です。例えば、スイッチの長押しによって、「設定時間変更モード」にエントリするように実装し、そのモード中のスイッチ短押下によって、「設定時間」をインクリメントすることで設定時間の変更が可能です。デフォルトの設定時間は、下記定義の数値を変更することによりソースコードで設定可能です。

デフォルトの設定時間の定義(分) : `#define DEFAULT_MINUTE 3` ※0 以上の整数値で設定

デフォルトの設定時間の定義(秒) : `#define DEFAULT_SECONDS 0` ※0~59 の範囲の整数値で設定

設定時間(分)を保持する変数 : `unsigned int set_time_min`

設定時間(秒)を保持する変数 : `unsigned int set_time_sec`

4.6.2 カウント終了メロディ変更

本サンプルスケッチでは、カウント終了メロディを以下の配列で定義しています。配列の内容を変更することで、別のメロディに変更することが可能です。詳細は下記 URL を参照ください。

[Play a Melody using the tone\(\) function](#)

メロディを構成する配列 : `int melody[]` ※配列のデータに一音ごと音階(周波数)を代入

メロディの一音の長さを指定する配列 : `int noteDurations[]` ※配列のデータに一音ごとの長さを代入

5. 注意事項

5.1 Windows デバイス マネージャーに COM ポートが表示されない

初めて PC と評価ボードを接続した時に、PC がポートを認識せず Windows デバイス マネージャーに COM ポートが表示されない場合があります。

表示されない場合は、評価ボードに実装されている FTDI 社製 USB-シリアル変換器 (FT232RQ) のドライバを以下の手順でインストールしてください。

1. FTDI 社のホームページから対象 OS の最新版ドライバのインストーラをダウンロードし、インストールします。

<https://ftdichip.com/drivers/vcp-drivers/>

2. インストール完了後、Windows デバイス マネージャーの「ポート(COM と LPT)」に「USB Serial Port(COMx)」が表示されます。

以下の場合、対象 COM ポートが COM5 であることが確認できます。

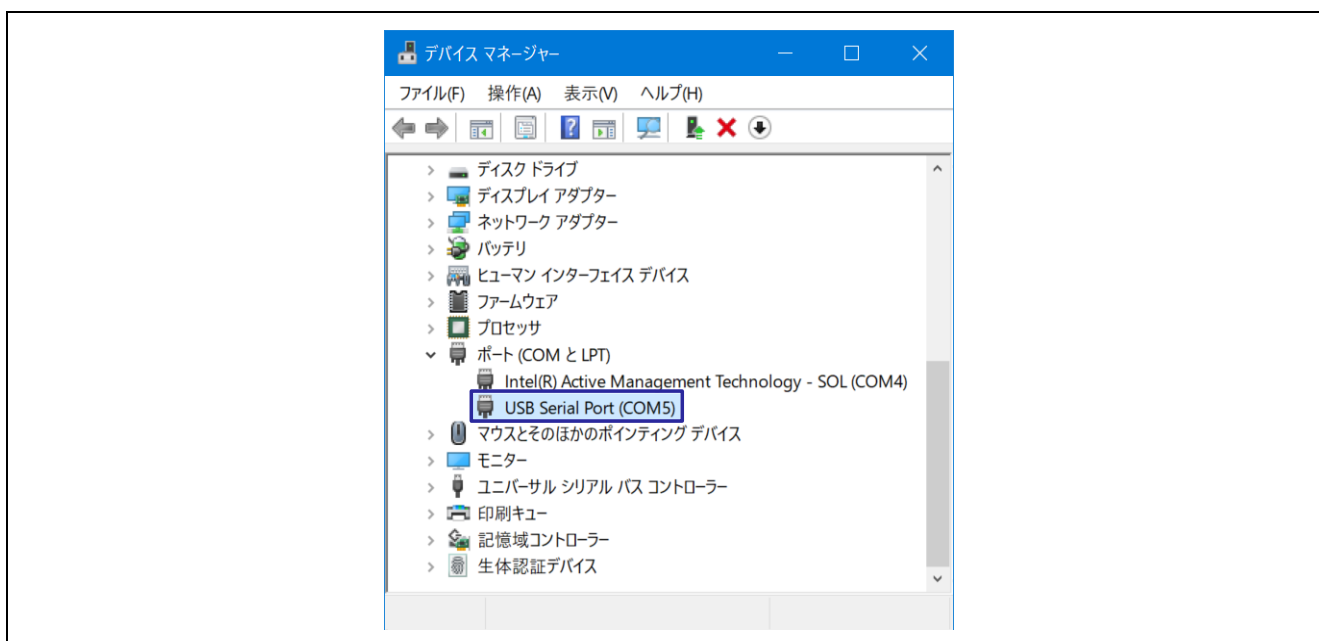


図 5-1 ドライバインストール後の Windows デバイス マネージャー

USB-シリアル変換器、COM ポートの詳細は、「RL78/G23-64p Fast Prototyping Board ユーザーズマニュアル」の「5.11 USB-シリアル変換器」、「5.12 USB-シリアル変換器リセットヘッダ」を参照してください。

5.2 RL78/G23-64p Fast Prototyping Board に正しく書き込みができない

マイコンボードに書き込む際、正しく接続できない場合があります。

接続できない場合は、Windows デバイス マネージャーから対象 COM のプラグアンドプレイ認識のチェックボックスのチェックを外してください。

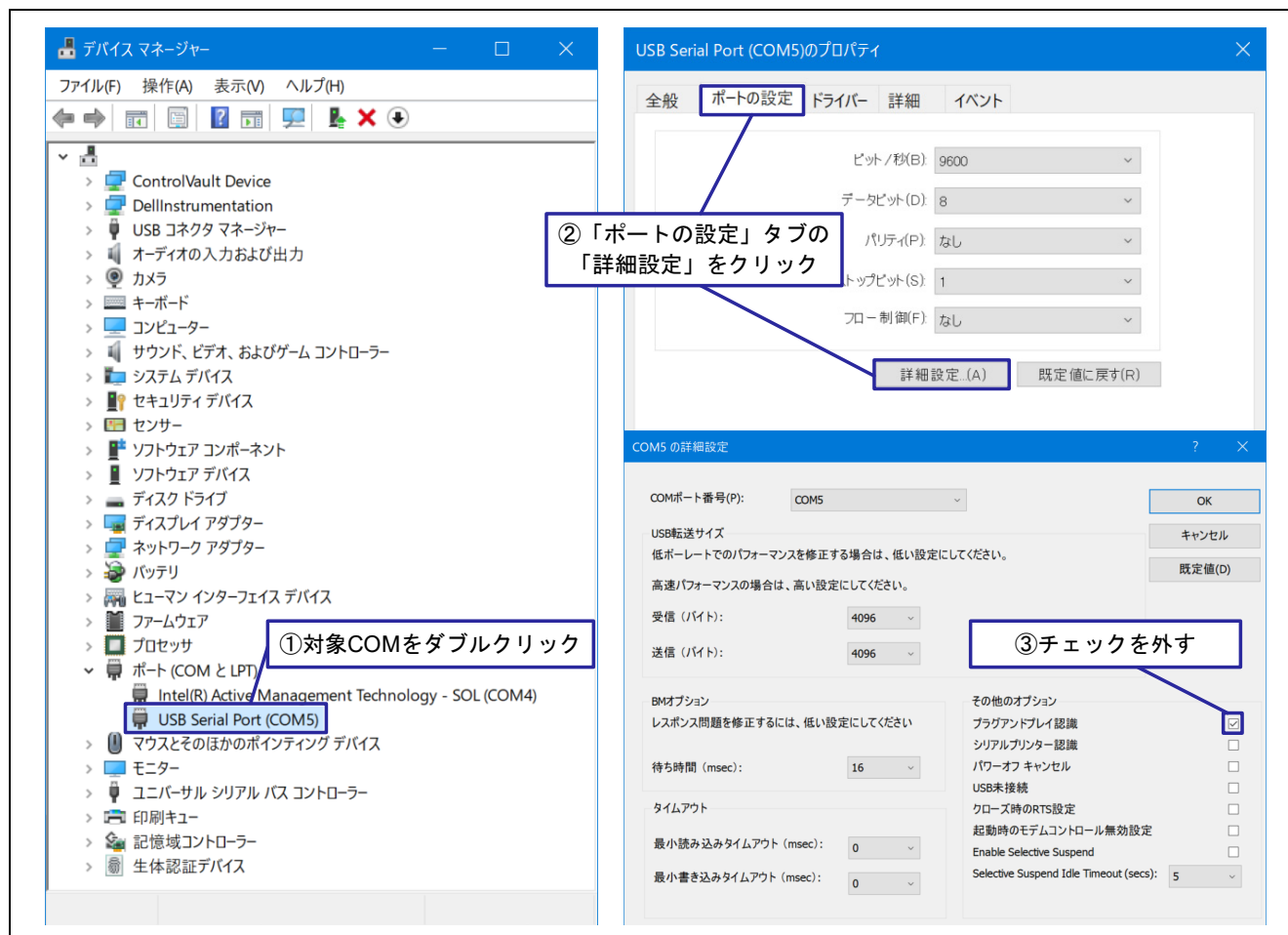


図 5-2 対象 COM の設定例

6. サンプルコード

本アプリケーションノートは、サンプルコードを用意しています。

サンプルコードはルネサス エレクトロニクスホームページから入手してください。

7. 参考ドキュメント

RL78/G23 ユーザーズマニュアル ハードウェア編 (R01UH0896)

RL78/G23-64p Fast Prototyping Board ユーザーズマニュアル (R20UT4814)

(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート／テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください)

すべての商標および登録商標は、それぞれの所有者に帰属します。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Apr.23.24	-	初版

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 V_{IL} (Max.) から V_{IH} (Min.) までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 V_{IL} (Max.) から V_{IH} (Min.) までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものとしします。
13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.5.0-1 2020.10)

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。