# **要旨**

このアプリケーションでは、ADT7410 TRZというデジタル温度センサーを使用して環境の温度を測定します。 IIC介してセンサーと通信します。

マイコン内で一般的に使用される通信方式を利用して、よく使用されるICセンサーを制御するプログラムを作成します。これにより、センサーの制御とデータ取得が把握しやすくなります。

プログラミングの目標が達成できるためにプログラミングの主な機能が以下に記載します。

・センサーで取得したデータをUSBでPC ターミナルに表示。

・ボードに搭載されているLCDに環境の温度を表示。

・100msec毎に温度センサーからデータを取得。

・10回取得した後に平均して、PC及びLCDにデータを表示。

・PC に表示するデータは10進数して、小数点第一位まで表示させる。

・Switch1を押下毎に温度センサーの取得タイミングを変更する。100msec->10msec->100msec->10msec...

**目次**

[**要旨 1**](#_Toc160013164)

[**1. 仕様 3**](#_Toc160013165)

[**１.１ADT7410TRZの動作環境 4**](#_Toc160013166)

[**2. 動作環境 6**](#_Toc160013167)

[**RL78/G14マイコン・トレーニング・キット 6**](#_Toc160013168)

[**3. 周辺機能 7**](#_Toc160013169)

[**4. ハードウェア 8**](#_Toc160013170)

[**4.1　ハードウェア構成 8**](#_Toc160013171)

[**4.2 使用する端子一覧 9**](#_Toc160013172)

[**５．ソフトウェア 10**](#_Toc160013173)

[**5.1 動作概要 10**](#_Toc160013174)

[**5.1.1 ADT7410TRZ のデータシートの参考 11**](#_Toc160013175)

[**5.2 ソフトウェア構成 11**](#_Toc160013176)

[**5.3 アプリケーション 12**](#_Toc160013177)

[**5.3.1関数一覧 12**](#_Toc160013178)

[**5.3.2定数一覧 13**](#_Toc160013179)

[**5.3.3 関数一覧仕様 14**](#_Toc160013180)

[**5.3.3.1 Comm\_Processing\_Initialization() 14**](#_Toc160013181)

[**5.3.3.2 PCと接続 14**](#_Toc160013182)

[**5.3.3.3 InititialiseLCD 15**](#_Toc160013183)

[**5.3.3.4 DisplayLCD 15**](#_Toc160013184)

[**5.3.3.5 TemperatureSensor\_Init 16**](#_Toc160013185)

[**5.3.3.6 TemperatureSensor\_Get 16**](#_Toc160013186)

[**5.3.3.7 Switch\_TimeChange 17**](#_Toc160013187)

[**5.3.3.8 Average\_10tGetData 17**](#_Toc160013188)

[**6.アプリケーションのセットアップ説明 18**](#_Toc160013189)

[**6.1　環境セットアップ 18**](#_Toc160013190)

# 1. **仕様**

表 1に周辺機能と用途、以下に記載します。

表 1使用する周辺機能と用途

|  |  |
| --- | --- |
| 周辺機能 | 用途 |
| IIC通信インターフェース | センサーとマイコンの間でデータを送受信するため |
| UART通信インターフェース(SCI) | 温度センサーから収集したデータをPCにUART経由で送信して PC端末にデータを表示する |
| LCDディスプレイ | 温度センサーから収集したデータをLCDディスプレイに表示 |
| タイマー | 温度センサーからデータを取得する時間を設定しデータ収集の一貫性と正確性を確保する |

## **１.１ADT7410TRZの動作環境**

図 1 ADT7410TRZの動作環境は以下に記載します。

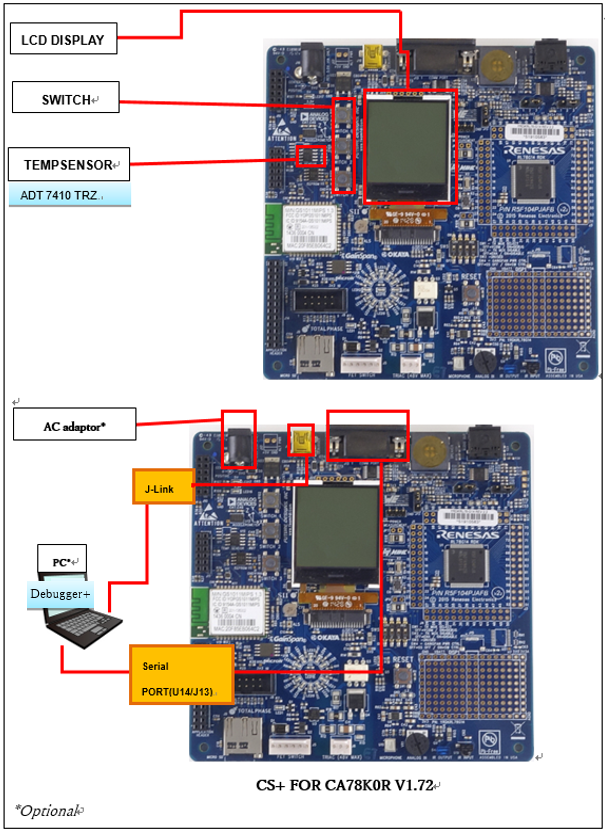


図 1 ADT7410TRZの動作環境

表 2 ADT7410TRZの動作環境を下に記載します。

表 2 ADT7410TRZの動作環境

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function | Connector | Pin | Signal | Condition |
| IIC | U12 | P60,P61 | R\_IICA0\_Master\_Receive R\_IICA0\_Send\_Receive | MTK RL78G14 上に搭載されているアプリケーション関数 |
| UART | TxD1-J14.2 RxD1-J14.3 | P02  P03 | R\_UART1\_Receive R\_UART1\_Send | MTK RL78G14 上に搭載されているアプリケーション関数 |
| LCD SPI インターフェース | LCD-CS  LCD-RS  MOSI  SCK  RSTOUT | P145 (pin 98)  P146 (pin 73)  P72 (pin 38)  P70(Pin40)  P130(Pin91) | DisplayLCD | MTK RL78G14 上に搭載されているアプリケーション関数 |

**1.1.1スイッチ設定**

・１０ｍｓデータを取得する時スイッチ状態　(スイッチ1→4 )

|  |  |  |  |
| --- | --- | --- | --- |
| SW1 (リセットボタン) | SW2 | SW3 | SW4 |
| ON | OFF | OFF | ON |

・１０0ｍｓデータを取得するスイッチの状態　( スイッチ1→4 )

|  |  |  |  |
| --- | --- | --- | --- |
| SW1 (リセットボタン) | SW2 | SW3 | SW4 |
| ON | OFF | OFF | ON |

# **2. 動作環境**

本アブリケーションのサンプルのプログラミングは以下に環境を想定しています。

表 3動作環境を下に記載します。

表 3動作環境

| 項目 | 内容 |
| --- | --- |
| 使用ボード | RL78/G14マイコン・トレーニング・キット |
| MPU | Rl78 グループ R01DS0053EJ0350 |
| 動作周波数 | CPU：RL78/G14(R5F104PJAFB、32MHz) |
| 動作電圧 | 3.3V |
| 統合開発環境 | ADI社製  RENESAS製  CS+ For CA78K0R V1.72 |
| エミュレータ | ADI社製  J-Link |

# **3. 周辺機能**

シリアルIIC とシリアルポート及び　それぞれのデータ受送信とコミュニケーション方式はルネサスのRL７８グループの資料のループ・ユーザーズマニュアル ハードウェア編を参照してください。

<https://www.renesas.com/jp/ja/document/apn/rl78g14-rl78g1c-rl76l12-rl78l13-rl78l1c-group-i2c-bus-single-master-control-software-using-iica>

他にはルネサスのホームページでも役に立つ資料が載せているので参照してください。

ADT7410TRZ(デジタル温度センサー)のデータシートは、こちらのリンクを参照してください。

<https://www.analog.com/media/en/technical-documentation/data-sheets/adt7410.pdf>

UARTの詳細はこちらのリンクを参照してください。

<https://www.renesas.com/jp/ja/document/apn/rl78g12-serial-array-unit-uart-communication-cc-rl-rev110>

# **4. ハードウェア**

## **4.1　ハードウェア構成**

以下にこのアプリケーションのハードウェア構成を示します

図 2ハードウェア構成(MTK-RL78G14)を以下に記載します。

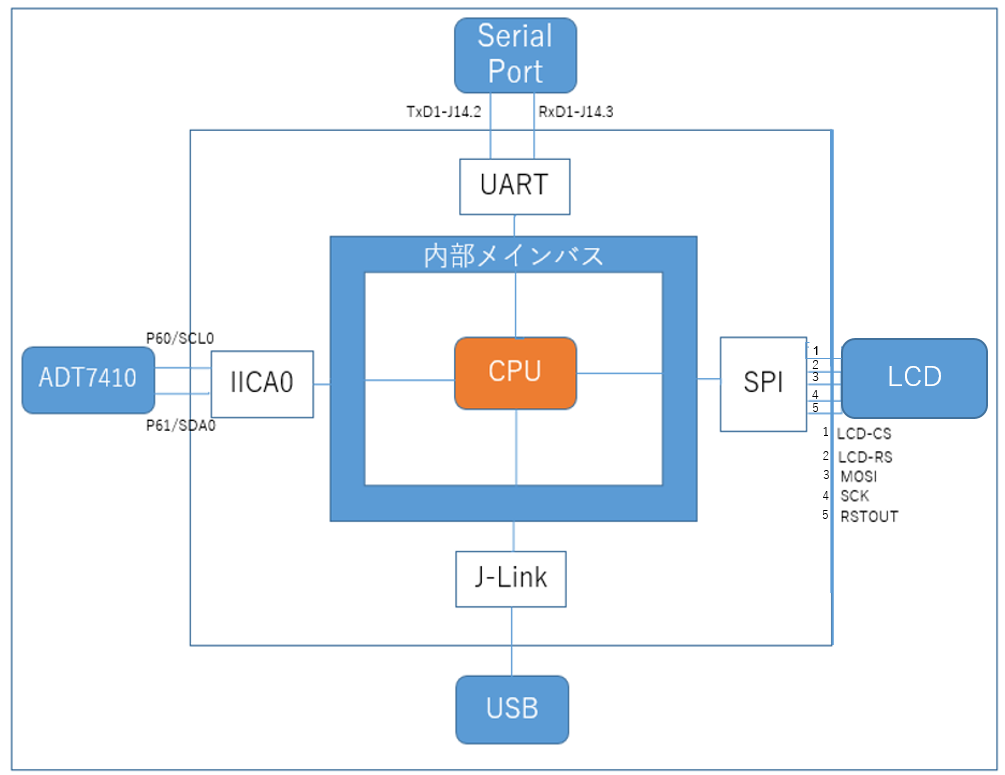


図 2ハードウェア構成(MTK-RL78G14)

## **4.2 使用する端子一覧**

　表 4端子一覧を以下に記載します。

表 4端子一覧

|  |  |  |
| --- | --- | --- |
| 端子名 | 入出力 | 機能 |
| R\_IICA0\_Master\_Send | 出力 | デジタル温度センサーにデータ送信 |
| R\_IICA0\_Master\_Receive | 入力 | デジタル温度センサーからデータ受信 |
| R\_UART1\_Send | 出力 | ターミナルへのデータ送信 |
| R\_UART1\_Receive | 入力 | デジタル温度センサーからデータ受信 |
| G\_CSI21\_SendingData | 出力 | LCDにデータを送信 |

# **５．ソフトウェア**

## **5.1 動作概要**

現在、このアプリケーションはデジタル温度センサーからIICインターフェースを介してデータを取得し、シリアルポートのUARTを通じてPCに表示します。  
　例としては：  
　->３０．１　をPCに表示する次は１００ｍsec毎にデータをアップデートしたら  
　->３２．３  
　->４０．３　  
課題の仕様によってSWITCHとも接続して動作させます。

#### 5.1.1 ADT7410TRZ のデータシートの参考

こちらのリンクをクリックしてADT7410データシートを参照してください。

<https://www.analog.com/media/en/technical-documentation/data-sheets/adt7410.pdf>

## **5.2 ソフトウェア構成**

図 3ソフトウェア構成を以下に記載します。

Application

SPI

Device Driverir

UART

SPI

IICA0

**MTK RL78G14**

Temp Sensor

PCターミナル

LCD

図 3ソフトウェア構成

## **5.3 アプリケーション**

### **5.3.1関数一覧**

表 5関数一覧は以下に記載します。

表 5関数一覧

|  |  |  |
| --- | --- | --- |
| レイヤー / ブロック | 関数名 | 見出し |
| アプリケーション | Temperature\_Display() | 5.4.3.1 |
| R\_iica0\_callback\_master\_receiveend() | 5.4.3.2 |
| InititialiseLCD() | 5.4.3.3 |
| DisplayLCD() | 5.4.3.4 |
| TemperatureSensor\_Init() | 5.4.3.5 |
| TemperatureSensor\_Get() | 5.4.3.6 |
| Switch\_TimeChange() | 5.4.3.7 |
| Average\_10tGetData() | 5.4.3.8 |

### **5.3.2定数一覧**

表 6 定数一覧は以下に記載します。

表 6 定数一覧

|  |  |  |
| --- | --- | --- |
| 定数名 | 設定値 | 内容 |
| Tx\_buf | 1 | マスターがIICA0でデータ転送のレジスターのアドレス(単位: Byte) |
| Rx\_buf | 1 | マスターがIICA0でデータ転送のレジスターのアドレス(単位: Byte) |
| DATA\_LENGTH\_BYTE | 2 | 出力最大温度センサー(単位: Byte) |

### **5.3.3 関数一覧仕様**

#### 5.3.3.1 Temperature\_Display

Temperature\_Display()

|  |  |
| --- | --- |
| 概要 | LCDに現在の温度を表示します。 |
| ヘッダファイル | r\_cg\_serial.h |
| 宣言 | void Temperature\_Display(); |
| 内容 | この関数は温度を液晶画面に表示します。 |
| 引数 | なし |
| リターン値 | なし |

#### 5.3.3.2 R\_iica0\_callback\_master\_receiveend

R\_iica0\_callback\_master\_receiveend()

|  |  |
| --- | --- |
| 概要 | この関数は、i2cbuff配列から受信した温度データを取り出し、temperature\_Reading構造体に格納します。 |
| ヘッダファイル | r\_cg\_serial.h |
| 宣言 | static void r\_iica0\_callback\_master\_receiveend(void) |
| 内容 | この関数は、i2cbuff配列から受信した温度データを取り出して、temperature\_Reading構造体に保存します。 |
| 引数 | なし |
| リターン値 | なし |

#### 5.3.3.3 InititialiseLCD

InititialiseLCD()

|  |  |
| --- | --- |
| 概要 | ボード搭載のLCDの初期化を設定 |
| ヘッダファイル | glyph￥LCD.h |
| 宣言 | void InitialiseLCD(void) |
| 内容 | この関数でLCDの初期化を設定 |
| 引数 | なし |
| リターン値 | なし |

#### 5.3.3.4 DisplayLCD

DisplayLCD()

|  |  |
| --- | --- |
| 概要 | LCD に取得したデータを表示する |
| ヘッダファイル | glyph￥LCD.h |
| 宣言 | void DisplayLCD(uint8\_position, const \* string ) |
| 内容 | LCD に取得したデータを表示する |
| 引数 | Uint8\_t position 文字が書いてあるライン const uint8\_t \*string　　　　　　　　ラインの内容 |
| リターン値 | なし |

#### 5.3.3.5 TemperatureSensor\_Init

TemperatureSensor\_Init

|  |  |
| --- | --- |
| 概要 | デジタル温度センサーの初期化を設定 |
| ヘッダファイル | ｒ\_cg\_serial.h |
| 宣言 | void TemperatureSensor\_Init(void) |
| 内容 | デジタル温度センサーの設定 |
| 引数 | なし |
| リターン値 | なし |

#### 5.3.3.6 TemperatureSensor\_Get

TemperatureSensor\_Get()

|  |  |
| --- | --- |
| 概要 | デジタル温度センサーのデータを取得 |
| ヘッダファイル | r\_cg\_serial.h |
| 宣言 | int16\_t TemperatureSensor(void) |
| 内容 | データを取得する |
| 引数 | なし |
| リターン値 | なし |

#### 5.3.3.7 Switch\_TimeChange

Switch\_TimeChange()

|  |  |
| --- | --- |
| 概要 | この関数は、時間変更のスイッチを処理します。 |
| ヘッダファイル | r\_cg\_serial.h |
| 宣言 | void Switch\_TimeChange(); |
| 内容 | この関数は、時間変更のスイッチが押されたときに呼び出され、時間の変更を処理します |
| 引数 | なし |
| リターン値 | なし |

#### 5.3.3.8 Average\_10tGetData

Average\_10tGetData()

|  |  |
| --- | --- |
| 概要 | 温度センサーからデータを10回取得した後に平均します。 |
| ヘッダファイル | なし |
| 宣言 | double Average\_10tGetData(); |
| 内容 | 10個のデータを取得し、それらの値の平均値を計算します。 |
| 引数 | なし |
| リターン値 | 計算された平均値 (double型) |

# **6.アプリケーションのセットアップ説明**

## **6.1　ソフトウェアの環境セットアップ**

1 最初はもしCSまだインストールしてなかったら、このドキュメントとフォルダーに搭載しているのCS+のインストール方法をセットアップしておいてください。

写真

2テラタームもインストールしてセットアップしておいてください。

3次にプロジェクトをインポートするときは下の順番を進んでください。

④ハードウェアのセットアップは下の写真のようにセットアップしてください。設定順番も下のように進んでください。

4最後に実行とデバッグするときも下の順番に実施してください。