

RL78/ G10

IICA0（スレーブ機能編）

要旨

本サンプルコードでは、IICA0 機能を用いた IIC バスのスレーブ機能の実現方法を示します。

対象デバイス

RL78/G10(R5F10Y47)

1. IIC バスの基本的な内容

IIC バスでは、通信の制御は IIC バスのマスタが行います。スレーブはマスタからの指示に従って、データの送受信を行います。スレーブができることは、マスタの送信データに対して、ACK 応答するか NACK 応答するかとマスタとの同期をとるために SCL 信号をロウ・レベルに引いて、マスタにウェイトをかけることだけです。ただし、マスタによっては、ウェイト機能に対応していないこともあるので、注意が必要です。

それでは、スレーブは絶対にマスタに従うだけかというそうではなく、通信の細かなプロトコルはマスタの指示に従いますが、その上位の階層では、逆にマスタはスレーブの仕様に合わせる必要があります。これは、スレーブは IIC バスを介して、何らかの機能をマスタに提供しますが、提供する機能はスレーブが規定しているからです。

そのため、最初にスレーブとして、「どのような機能を提供するか」を明確に定義します。それによって、マスタが通信を行うことになります。

このように、スレーブでは、提供する機能で IIC バスからのアクセスに対する処理が決まってしまうので、マスタのように IIC バスの制御を階層化することは難しいです。ここからは、マスタからの指示に応じて処理が変化します。

2. IIC バスのスレーブとしての基本的な仕様

2.1 IIC バスに関する仕様

IIC バスに関する仕様は以下のようになります。

- ・ 接続する IIC バス : ファースト・モードと標準モード
- ・ スレーブ・アドレス : 0x60（コード生成時に指定）
- ・ 拡張コード対応 : 対応しない（無視して通信から退避）
- ・ レジスタ構成 : スレーブ・アドレスに続く 8 ビットで使用する機能を指定します。

レジスタ・アドレス	R/W	機 能
0x00 ~ 0x01	Write	LED表示データ書き込み
0x00 ~ 0x01	Read	A/D変換結果読み出し
0x80 ~ 0xFF	Read/Write	RAMへの読み書き

2.2 スレーブ機能の仕様

スレーブの機能として、以下の 3 つの機能を提供します。3 つの機能はレジスタのアドレスと送信か受信かで切り替わります。

- ・ LED 表示機能 : 表示データは 2 個の 8 ビットのデータで、SW で切り替えて表示します。
- ・ A/D 変換機能 : 2 チャンネルのアナログ入力の 16 サンプル分の移動平均を送信します。

- ・ RAM 機能 : 128 バイトのデータの任意のアドレスからの読み書きができます。

2.2.1 LED 表示機能

LED 表示器としては、8 ビットのデータを表示するために 8 個の LED を使用しています。表示可能なデータは 2 バイトで、SW 入力により 2 バイトのデータのどちらかを指定できるようにします。SW を押していないときには、レジスタ・アドレス 0x00 のデータ、押しているときにはアドレス 0x01 のデータを表示します。

表示は上位 4 ビットと下位 4 ビットの 2 回に分けた時分割で表示します。表示の周期は 10ms とします。マスタからの表示データはマスタがストップ・コンディションを発行した時点で表示データとして確定します。確定したデータは 50ms 以内に表示可能になります。

2.2.2 A/D 変換機能

2 チャンネルのアナログ入力を変換し、最新の 16 回分の移動平均を得ることが可能です。A/D 変換の仕様は以下の通りです。RL78/G10 の A/D コンバータにはスキャン・モードや連続変換機能がないので、ソフトウェアで処理します。

- ・ アナログ入力 : チャンネル 3～4 の 2 チャンネル
- ・ 変換方式 : ソフトウェアでスキャン・モード連続変換を実現
- ・ 変換分解能 : 10 ビット
- ・ 変換時間 : 4.6 μ 秒／チャンネル
- ・ バッファ : 16 データ／チャンネル（合計 64 バイト）

変換結果はレジスタ・アドレス 0x00～0x01 で読み出すことができます。10 ビットの変換結果を上位 2 ビット、下位 8 ビットの順で読み出せます。チャンネル 4 の下位 8 ビットを読み出したあとは、図 2.1 に示すようにチャンネル 0 に戻ります。

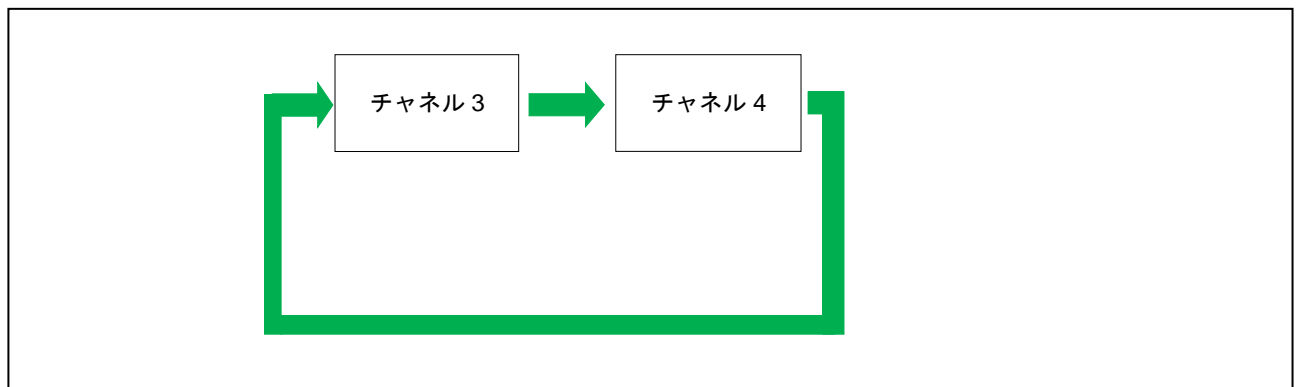


図 2.1 A/D 変換結果の読み出し

2.2.3 RAM 機能

128 バイトの一時保管に使用できる 128 バイトの領域です。初期状態では、0x00～0x7F のデータが格納されています。スレーブ・アドレスに続けて 8 ビットで指定するレジスタのアドレスが 0x80～0xFF の場合に RAM へのアクセスとなります。受信したデータは直ちに RAM に書き込まれます。RAM へのアクセスは自動的にアドレスが更新されていきます。アドレス 0x7F へのアクセスの次はアドレス 0x00 へのアクセスとなります。

RAM のデータを読み出す場合のアクセス方法を図 2.4 に示します。

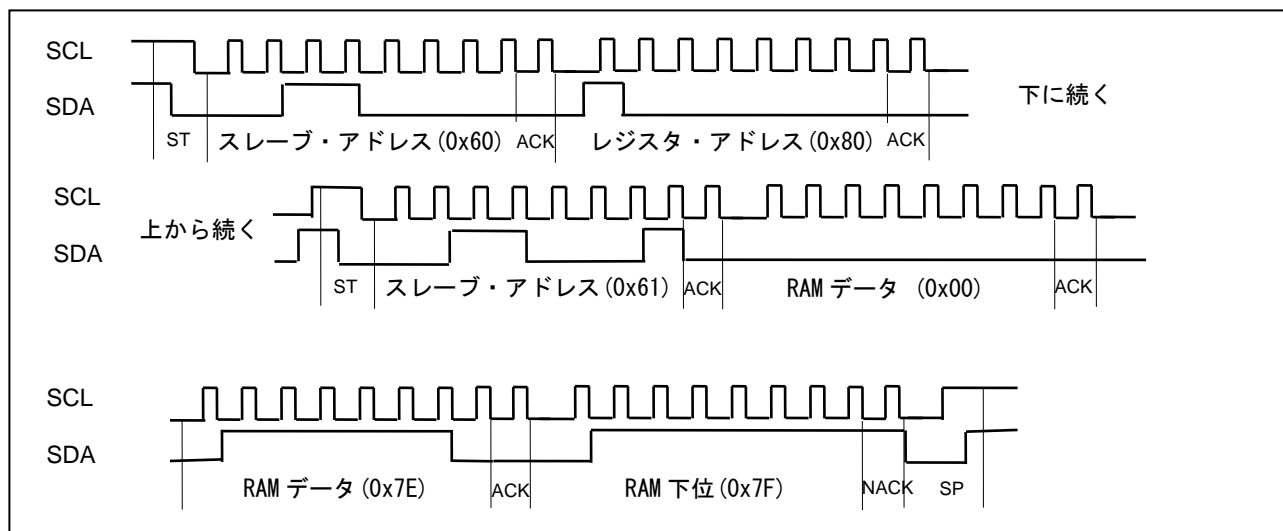


図 2.4 RAM データの読み出しタイミング

この例では、最初にスタート・コンディション（ST）、スレーブ・アドレス（0x60）に続けて、レジスタ・アドレス 0x80（＝RAM のアドレス 0x00）を指定しています。

その後に、リスタートして、読み出しでスレーブを選択（0x61）することで、指定したアドレス 0x00 番地の RAM の値を読み出します。この図では、アドレス 0x00 の値は 0x00 になっています

一番下のタイミングでは RAM アドレス 0x7E の値を読み出し、0x7F の値を読み出したところで、マスタが NACK 応答を戻してきたので、スレーブは通信完了として通信から退避します。最後に、ストップ・コンディション（SP）で IIC バスを開放して通信を完了します。

2.3.4 RAM へのデータ書き込み

RAM にデータを書き込む場合のアクセス方法を図 2.5 に示します。

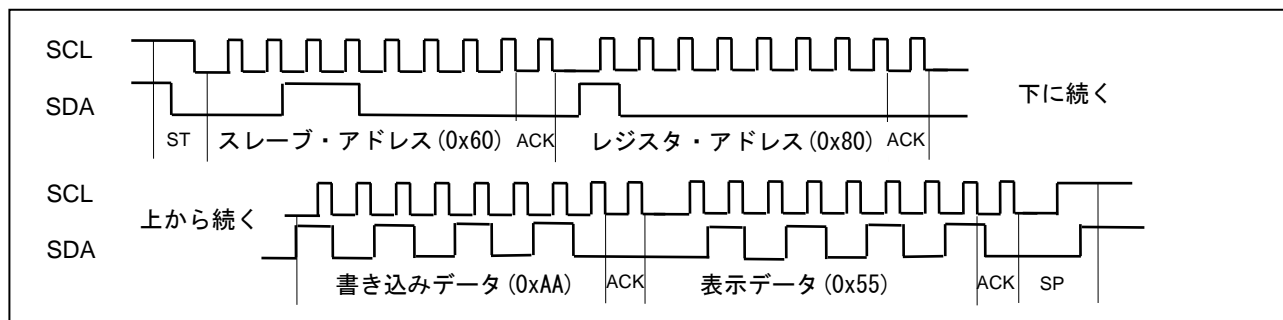


図 2.5 RAM へのデータ書き込みタイミング

この例では、最初にスタート・コンディション（ST）、スレーブ・アドレス（0x60）に続けて、レジスタ・アドレス 0x80（＝RAM のアドレス 0x00）を指定しています。

その後、0x00 番地への書き込みデータとして 0xAA を次の番地への書き込みデータとして 0x55 を送信しています。2 バイトのデータを送信して通信を完了し、ストップ・コンディションを発行して IIC バスを開放しています。

3. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 3.1 動作確認条件

項目	内容
使用マイコン	RL78/G10 (R5F10Y47)
動作周波数	<ul style="list-style-type: none"> ● 高速オンチップ・オシレータ (HOCO) クロック : 20MHz ● CPU/周辺ハードウェア・クロック : 20MHz
動作電圧	3.3V (3.02V~5.5V で動作可能) SPOR 電圧 : 2.84V
統合開発環境	ルネサス エレクトロニクス製 CS+ V3.01.00
コンパイラ	ルネサス エレクトロニクス製 CC-RL V1.01.00
使用ボード	R 5 F 1 0 Y 4 7 A S P 使用 R L 7 8 マイコンモジュール LED 高輝度 1 0 ポイント赤色バーLEDアレイ O S X 1 0 2 0 1 - R (モジュールの 10 個中 8 個使用)、SW 等を搭載

4. ハードウェア説明

4.1 ハードウェア構成例

図 5.1 に本サンプルコードで使用するハードウェア構成例を示します。

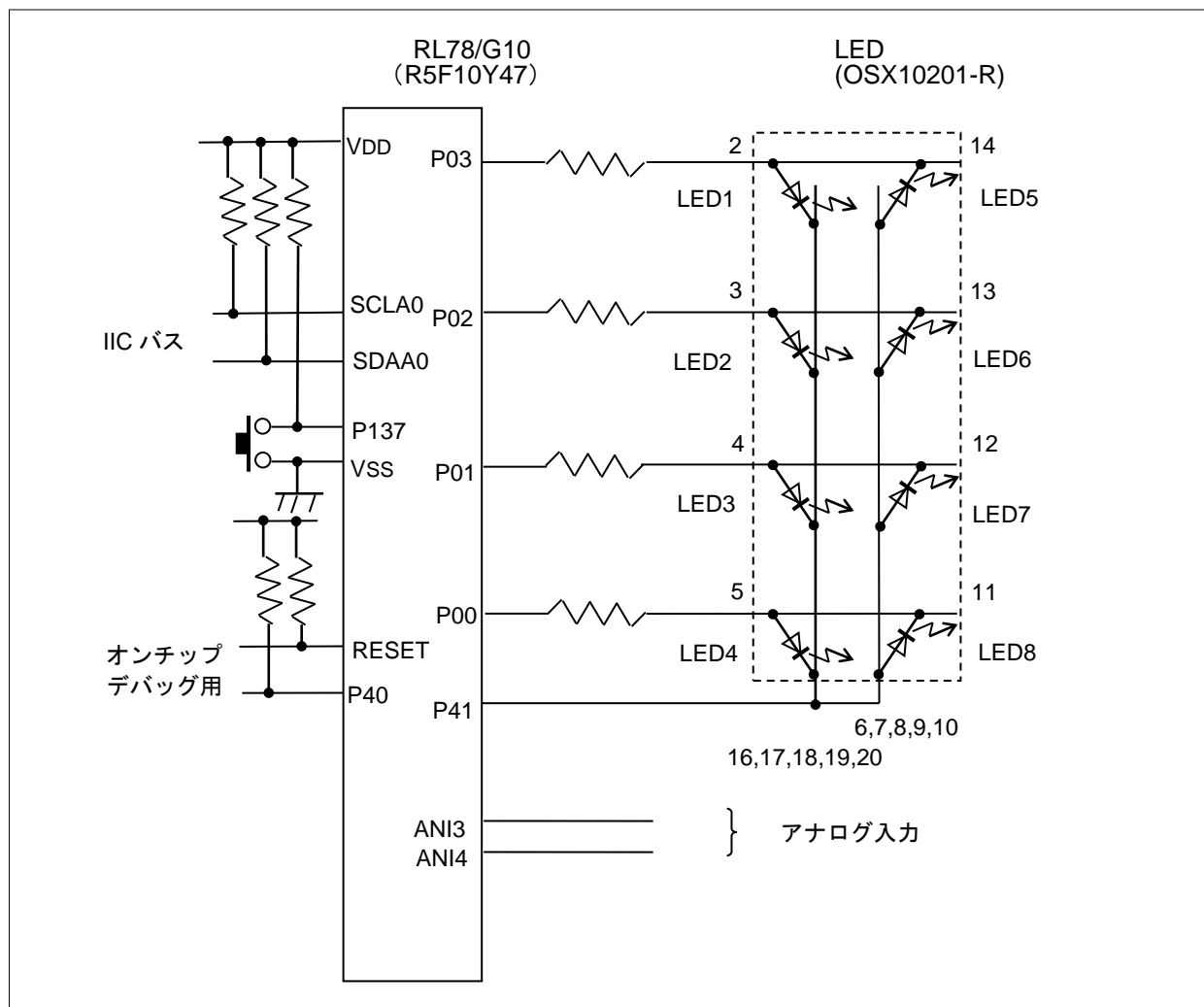


図 5.1 ハードウェア構成

注意 1 この回路イメージは接続の概要を示す為に簡略化しています。実際に回路を作成される場合は、端子処理などを適切に行い、電気的特性を満たすように設計してください（入力専用ポートは個別に抵抗を介して VDD 又は VSS に接続して下さい）。

2 VDD は SPOR 電圧 (V_{SPOR}) 以上にしてください。

4.2 使用端子一覧

表 5.1に使用端子と機能を示します。

表 5.1 使用端子と機能

端子名	入出力	内容
SDA00	入出力	IIC 通信データ信号
SCL00	入出力	IIC 通信クロック信号
P03～P00	出力	LED へのデータ出力
P41	出力	LED の点灯タイミング出力
ANI3～ANI4	入力	アナログ信号入力
P137	入力	SW 入力

5. ソフトウェア説明

5.1 動作概要

本アプリケーションノートでは、内蔵周辺機能の初期設定だけは CS+のコード生成機能を利用します。

内蔵周辺機能の初期設定が完了したら、データの初期化を行い、A/Dコンバータと LED 点灯用タイマを起動します。A/D 変換と LED 点灯は割り込みによりバックグラウンドで処理します。IIC バスでの通信についても、IICA0 割り込みによりバックグラウンドで処理します。

メイン処理では、2 チャネル分の A/D 変換完了を待ちます。その間に IIC バスでストップ・コンディション検出したら、LED 点灯用のデータ受信バッファから点灯制御用バッファに転送します。2 チャネル分の A/D 変換が完了したら、移動平均値を IIC バスの送信用バッファに転送します。送信用バッファに転送されたデータは、マスタからの指示に対応して IIC バスに送信されます。

A/D 変換完了割り込み処理では、各チャネルの変換結果を加算していきます。加算する値が 16 以上になると、最も古い値を最新の値に置き換えていきます。チャネル 3～4 の変換が完了すると、メイン処理に A/D 変換完了を通知します。

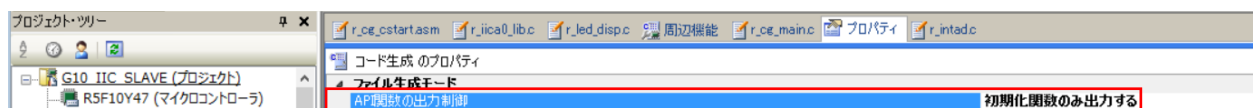
5ms のタイマ割り込みでは、点灯データを上位 4 ビット、下位 4 ビットの順に 2 時分割で点灯させます。また、50ms に 1 回、SW の状態を確認して、2 つの点灯データのどちらを使用するかを指定します。

IICA0 割り込みでは、マスタからのアドレスを指定するレジスタの値、送信／受信の指定に応じた処理を行います。詳細は、「2.1 IIC バスに関する仕様」を参照してください。IICA0 割り込み処理では、メイン処理に対してストップ・コンディションを検出したことを通知することで、IIC バスの通信が完了したことを示します。

以上のように、ほとんどの処理は割り込みベースで行われます。メイン処理は、割り込み処理で準備されたデータを別の割り込み処理でできるように、データを必要なバッファに設定するだけです。

5.2 コード生成での設定内容

プロパティの「ファイル生成モード」の「API 関数の出力制御」を「初期化関数のみ出力する」に設定し、以下の設定を行います。



CA78K0R からの継承性で、CC-RL78 のプロパティの「コンパイル・オプション」の「その他」の「他コンパイラからの以降支援機能を使用する」を「はい (CA78K0R) (-convert_cc=ca78k0r)」に設定しておきます。

(d) リセット要因確認

- ・リセット要因を確認する関数を入力する：チェックを」外す

端子割り当て設定	クロック設定	ブロック図	オンチップ・デバッグ設定	リセット要因確認
関数出力設定				
<input type="checkbox"/> リセット要因を確認する関数を入力する				

(2) ポートの設定

- ・ P00～P03：入力ポートに設定
- ・ P41：出力ポートに設定（1 を出力）

Port0	Port4	Port12	Port13
P00			
<input type="radio"/> 使用しない	<input checked="" type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ
<input type="checkbox"/> N-ch <input type="checkbox"/> 1を出力			
P01			
<input type="radio"/> 使用しない	<input checked="" type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ
<input type="checkbox"/> N-ch <input type="checkbox"/> 1を出力			
P02			
<input type="radio"/> 使用しない	<input checked="" type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ
<input type="checkbox"/> 1を出力			
P03			
<input type="radio"/> 使用しない	<input checked="" type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ
<input type="checkbox"/> 1を出力			
P04			
<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ
<input type="checkbox"/> 1を出力			
P05			
<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ
<input type="checkbox"/> 1を出力			
P06			
<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ
<input type="checkbox"/> N-ch <input type="checkbox"/> 1を出力			
P07			
<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ
<input type="checkbox"/> N-ch <input type="checkbox"/> 1を出力			

Port0	Port4	Port12	Port13
P40			
<input checked="" type="radio"/> 使用しない	<input type="radio"/> 入力	<input type="radio"/> 出力	<input checked="" type="checkbox"/> 内蔵プルアップ
<input type="checkbox"/> 1を出力			
P41			
<input type="radio"/> 使用しない	<input type="radio"/> 入力	<input checked="" type="radio"/> 出力	<input type="checkbox"/> 内蔵プルアップ
<input checked="" type="checkbox"/> 1を出力			

(3) タイマの設定

(a) 一般設定 チャンネル 0：インターバル・タイマ

一般設定	チャンネル0	チャンネル1	チャンネル2	チャンネル3
機能				
チャンネル 0	インターバル・タイマ			
チャンネル 1	使用しない			
チャンネル 2	使用しない			
チャンネル 3	使用しない			

(b) チャンネル 0

- ・動作モード設定：16 ビット
 - ・インターバル時間（16 ビット）設定：5ms
- これ以外は初期状態のまま

一般設定	チャンネル0	チャンネル1	チャンネル2	チャンネル3
インターバル・タイマ設定				
インターバル時間(16ビット)		5	ms	(実際の値: 5)
<input type="checkbox"/> カウント開始時にINTTM00割り込みを発生する				
割り込み設定				
<input checked="" type="checkbox"/> タイマ・チャンネル0のカウント完了で割り込み発生(INTTM00)				
優先順位		低		

(4) インターバル・タイマの設定

初期値（使用しない）のまま

(5) クロック出力／ブザー出力の設定

初期値（使用しない）のまま

(6) ウォッチドッグ・タイマの設定

- ・ HALT/STOP モード時の動作設定：停止
- ・ ウォッチドッグ・タイマ動作設定：使用しない

ウォッチドッグ・タイマ動作設定
☒ 使用しない ☐ 使用する

HALT/STOPモード時の動作設定
☐ 許可 ☒ 停止

オーバーフロー時間設定
オーバーフロー時間 $(2^{16} - 1)/f_{IL}$ 4369.00 (ms)

割り込み設定
☒ オーバーフロー時間の75% + $3/(f_{IL} \times 4)$ 到達時にインターバル割り込みを発生する(INTWDTI)

優先順位 低

(7) A/D コンバータの設定

- ・ A/D コンバータ動作設定：使用する
- ・ コンパレータ動作設定：許可
- ・ 分解能設定：10 ビット
- ・ 動作モード設定：ANI3、ANI4 だけチェック
- ・ 変換開始チャンネル設定：ANI3
- ・ 変換時間モード：標準 1
- ・ 変換時間：4.6 (92/fCLK) (μ s)
- ・ 割り込み設定：A/D の割り込み許可をチェック（優先順位はレベル 1）

A/Dコンバータ動作設定
☐ 使用しない ☒ 使用する

コンパレータ動作設定
☐ 停止 ☒ 許可

分解能設定
☒ 10ビット ☐ 8ビット

動作モード設定
☐ ANI0 ☐ ANI1 ☐ ANI2 ☒ ANI3
☒ ANI4 ☐ ANI5 ☐ ANI6

変換開始チャンネル設定
ANI3

変換時間設定
変換時間モード 標準1
変換時間 92/fCLK 4.6 (μ s)

割り込み設定
☒ A/Dの割り込み許可(INTAD)

優先順位 レベル1

(8) コンパレータの設定

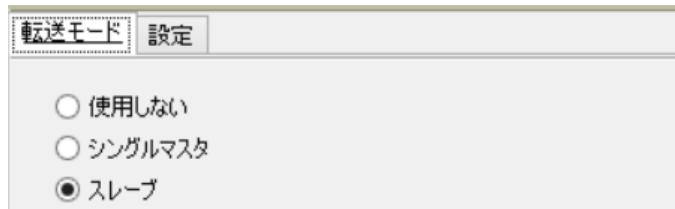
初期値（使用しない）のまま

(9) シリアルの設定

SAU 関係は全て初期値（使用しない）のまま

(10) シリアル・インタフェース IICA の設定

(a) 転送モード：スレーブに設定



(b) 設定

- ・ 自局アドレス：0x60
- ・ 動作モード設定：ファースト・モード（デジタル・フィルタ・オン）
- ・ 割り込み設定：優先順位はレベル 2



(11) 割り込みの設定

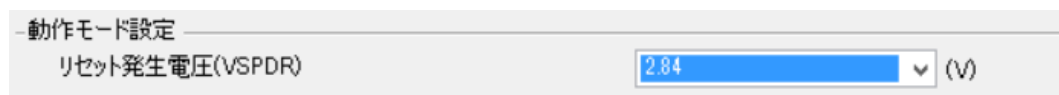
全て初期値（チェックなし）のまま

(12) キー割り込みの設定

全て初期値（チェックなし）のまま

(13) 電圧検出回路の設定

- ・ リセット発生電圧(VSPDR)：2.84（V）



定数一覧

表 6.2 にサンプルコードで使用する定数を示します。

表 6.2 サンプルコードで使用する定数

定数名	設定値	内容
POWER	4	A/D 変換のサンプル回数指定 (2 の階乗で指定)
SAMPLE	$2 \ll (\text{POWER}-1)$	A/D 変換のサンプル回数
TRUTH	1	真
FALSE	0	偽
INT_MASK	1	割り込み禁止 (マスク)
INT_ENABLE	0	割り込み許可 (マスク解除)
DETECT_STOP	0b00001000	ストップ・コンディション検出
DATA_NUMBER	2	LED に表示するデータ数
DISP_OFF_DATA	0b00001111	LED 消灯用 P0 データ
TIMING1	0b00001111	上位 4 ビット点灯用 P0 データ
TIMING2	0b00000010	下位 4 ビット点灯用 P0 データ
KEY_TIMING	10	SW 状態確認タイミング用
DATA_MASK	DATA_NUMBER-1	点灯用データ・ポインタのマスクデータ
CH_NO	2	IICA0 送信データ数 (A/D チャネル数)
DATA_MAX	2	LED 点灯用受信データ数
TX_LIMIT	DATA_MAX - 1	送信用ポインタのマスクデータ
RX_LIMIT	DATA_MAX - 1	LED 点灯用データ受信ポインタのマスクデータ
RAM_BASE	0x80	RAM 先頭のレジスタのアドレス
RAM_MASK	RAM_BASE - 1	RAM 領域のポインタのマスクデータ
COM_FLAG_MASK	0b00001111	通信フラグのクリア用データ
SLAVE_MODE	0	IICS0 の MSTS0 がスレーブ状態
DETECT	1	IIC バスでの検出状態
SELECTED	1	IICS0 の COI0 が選択状態
NOT_SELECTED	0	IICS0 の COI0 が非選択状態
RECEIVE_MODE	0	IICS0 の TRC0 が受信状態
TRANSMIT_MODE	1	IICS0 の TRC0 が送信状態
NACK	0	IICS0 の ACKD0 が ACK 未検出
F_MODE	g_status.7	IICA0 通信状態を示すビット
F_DIR	g_status.6	IICA0 の通信方向を示すビット
F_READY	g_status.5	通信レディ・ビット
F_REGADR	g_status.4	アドレス・レジスタの設定タイミング
F_SPD	g_status.3	ストップ・コンディション検出ビット
SPD_f	g_IICS_f.0	IICS0 の SPD0 ビットのコピー
STD_f	g_IICS_f.1	IICS0 の STD0 ビットのコピー
ACKD_f	g_IICS_f.2	IICS0 の ACKD0 ビットのコピー
TRC_f	g_IICS_f.3	IICS0 の TRC0 ビットのコピー
COI_f	g_IICS_f.4	IICS0 の COI0 ビットのコピー
EXC_f	g_IICS_f.5	IICS0 の EXC0 ビットのコピー
ALD_f	g_IICS_f.6	IICS0 の ALD0 ビットのコピー
MSTS_f	g_IICS_f.7	IICS0 の MSTS0 ビットのコピー

5.3 変数一覧

表 6.3 にサンプルコードで使用する変数一覧を示します。

Type	Variable Name	Contents	Function Used
uint16_t	g_conv_data	A/D 変換データのバッファ	R_ADC_Init () r_adc_interrupt()
uint16_t	sum_data	A/D の加算用バッファ	R_ADC_Init () r_adc_interrupt()
uint8_t	Adc_End	A/D 変換完了フラグ	R_ADC_Init () r_adc_interrupt() main()
uint16_t *	gp_set_pt	A/D 変換結果格納用ポインタ	R_ADC_Init () r_adc_interrupt()
uint16_t *	gp_sum_pt	A/D 変換結果加算用ポインタ	R_ADC_Init () r_adc_interrupt() main()
uint8_t	g_disp_data_bf	LED 点灯データ	R_DISP_Init () R_Make_DISP_data () r_tau0_channel0_interrupt()
uint8_t	g_sel_data	点灯データ指定	R_DISP_Init () r_tau0_channel0_interrupt()
uint8_t	g_disp_timing	点灯タイミング	R_DISP_Init () r_tau0_channel0_interrupt()
uint8_t	g_ram_area	RAM 機能用バッファ	R_IICA0_Init () r_iica0_interrupt ()
uint8_t	g_rx_data	受信データ用バッファ	R_IICA0_Init () r_iica0_interrupt () R_IICA0_Get()
uint16_t	g_tx_data	送信データ用バッファ	R_IICA0_Init () r_iica0_interrupt () R_IICA0_Put ()
uint8_t	g_low_data_temp	送信の下位バイト保存用	r_iica0_interrupt ()
uint8_t	g_low_data_index	下位バイト送信フラグ	R_IICA0_Init () r_iica0_interrupt ()
uint8_t	gp_trx_data	IICA0 送信用ポインタ	R_IICA0_Init () r_iica0_interrupt ()
uint8_t	g_status	IICA0 通信状態フラグ	R_IICA0_Init () r_iica0_interrupt ()
uint8_t	g_IICS_f	IICS0 レジスタのコピー	r_iica0_interrupt ()

5.4 関数一覧

表 6.4 に使用する関数一覧を示します。

表 6.4 関数一覧

関数名	概要
R_ADC_Init()	A/D 変換関係の変数を初期化します
R_DISP_Init()	LED 表示関係の変数を初期化します
R_IICA0_Init()	IICA0 通信関係の変数を初期化します
R_ADC_Start()	A/D 変換を起動します
R_TAU0_Channel0_Start()	5ms のインターバル・タイマを起動します
R_IICA0_Status()	IICA0 通信状態を読み出します
R_IICA0_Get()	IICA0 受信データを読み出します
R_Make_DISP_data()	LED 発光データを設定します
R_IICA0_Put()	IICA0 送信バッファにデータを設定します
r_adc_interrupt()	A/D 変換完了割り込み処理
r_tau0_channel0_interrupt()	5ms インターバル・タイマ割り込み処理
r_iica0_interrupt	IICA0 割り込み処理