

MSP430F149 ボード

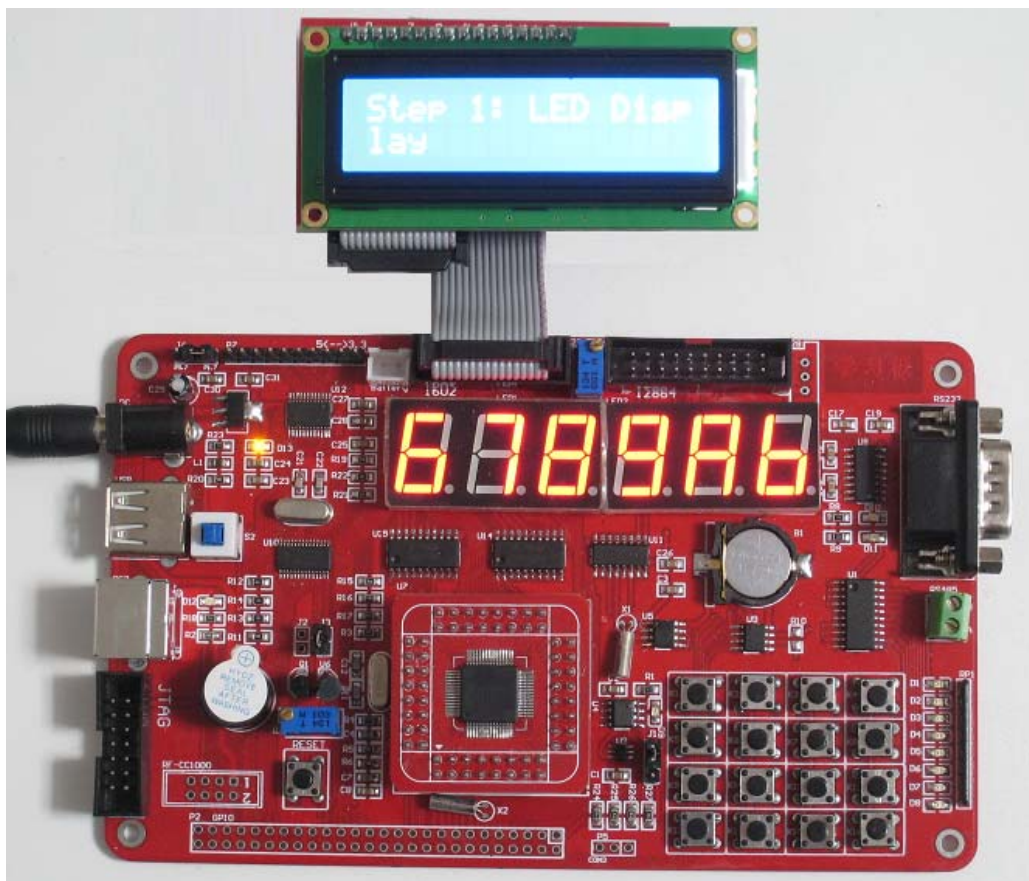
マニュアル

株式会社日昇テクノロジー

<http://www.csun.co.jp>

info@csun.co.jp

2010/12/26



[copyright@2010](#)



・ 修正履歴

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2010/12/26

※ この文書の情報は、文書を改善するため、事前の通知なく変更されることがあります。最新版は弊社ホームページからご参照ください。

「<http://www.csun.co.jp>」

※ (株)日昇テクノロジーの書面による許可のない複製は、いかなる形態においても厳重に禁じられています。



第一章 MSP430F149 ボードの概要.....	5
1.1 概要仕様	5
1.2 機能配置図.....	6
1.2.1 部品レイアウト.....	6
1.2.2 ジャンパー及びコネクタのレイアウト	8
1.3 ハードウェアの構成.....	8
1.3.1 MSP430F149.....	8
1.3.2 RS232.....	9
1.3.3 RS485.....	10
1.3.4 EEPROM.....	10
1.3.5 リアルタイムクロック	10
1.3.6 DAC	11
1.3.7 温度センサー	11
1.3.8 12864 液晶インタフェース	12
1.3.9 1602 液晶インタフェース	12
1.3.10 7SEG LED	13
1.3.11 レベル変換回路.....	14
1.3.12 PS2 インタフェース	14
1.3.13 USBインタフェース	15
1.3.14 LED	16
1.3.15 マトリックスキーボード	16
1.3.16 ADC回路	17
1.3.17 ブザー回路.....	17
1.3.18 リソースコントロール回路.....	17
1.3.19 RF微弱モジュールインタフェース	18
1.3.20 GPIO.....	19
1.3.21 JTAGインタフェース.....	19
1.4 サンプルソースについて	20
1.4.1 基本テスト.....	20
1.4.2 モジュール基本プログラム.....	20
1.4.3 総合テスト.....	21
第二章 IAR環境でソフトウェアの開発.....	22
2.1 新しいプロジェクトの作成.....	22
2.2 ソースファイルの追加.....	24
2.3 プロジェクトオプションの設定.....	24



2.4 コンパイル	28
2.5 Make.....	29
2.6 デバッグ	29

※ 使用されたソースコードは<http://www.csun.co.jp/>からダウンロードできます。



第一章 MSP430F149 ボードの概要

TI 社が提供する超低消費電力マイクロコントローラの MSP430 ファミリは、さまざまなアプリケーション向けに幅広いペリフェラルを備えた数多くのデバイスで構成されています。

1.1 概要仕様

■MSP430F149 搭載

MSP430F149 は下記主な特徴を持っています：

- ・ 16 ビット超低消費電力マイクロコントローラ
- ・ 1.8V～3.6V 幅広い給電電圧
- ・ 5 つの低消費電力モード、低消費電力モードからアクティブ・モードへのウェークアップが 6 μ s 以内で行われる
- ・ 2 KB RAM、60 KB フラッシュ
- ・ ハードウェア乗算器
- ・ 12 ビット ADC
- ・ 2x USART/SPI
- ・ 2xPWM 16-Bit タイマー (TimerA3, TimerB7)
- ・ 0.8 μ A リアルタイムモード
- ・ 0.1 μ A RAM 保持
- ・ WatchDog など

■カーネルボードと拡張ボードの分離設計となっており、使い便利

■拡張ボードに豊富な機能を搭載

- ・ 三種類の給電方式：外部電源、USB 給電、バッテリー給電
- ・ 8-bit双方向レベル変換I/F (5V--3.3V, 3.3V--5V)
- ・ USB I/F (USB2.0、USB1.1両方サポート)
- ・ PS2 I/F
- ・ JTAG I/F
- ・ ブザーI/F
- ・ RF 無線モジュール I/F
- ・ 12-Bit温度センサー
- ・ 12-Bit ADC I/F
- ・ 1602液晶モジュールI/F
- ・ 12864液晶モジュールI/F
- ・ 6×7SEG 表示モジュール
- ・ RTC リアルタイムクロック+ボタンバッテリー
- ・ 256×8-BitのEEPROM

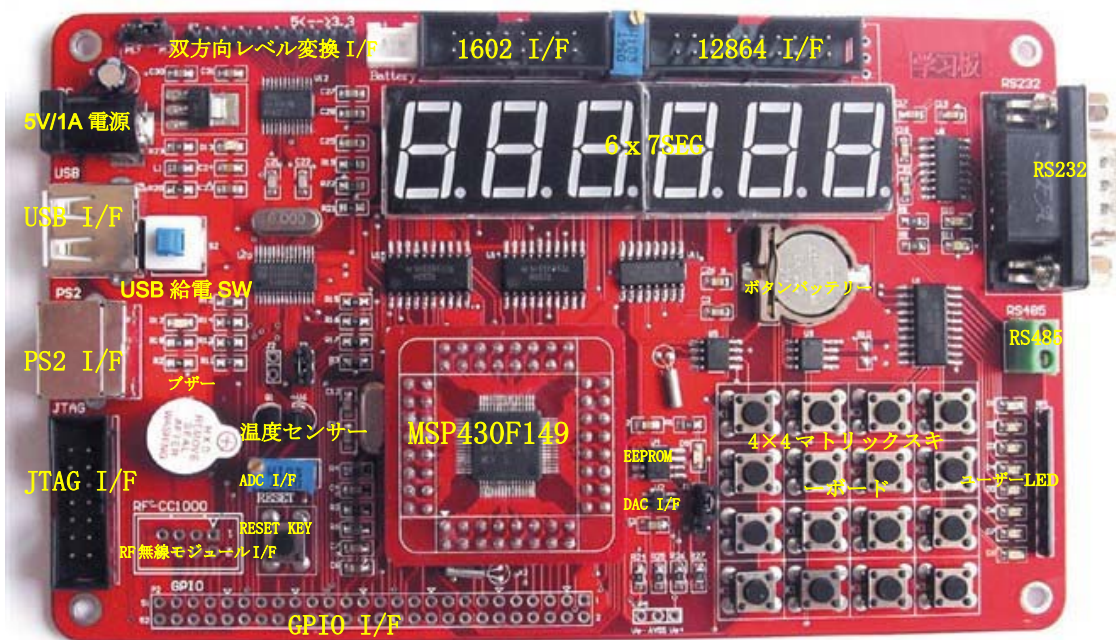
- ・ 8-Bit DAC
- ・ 4×4 マトリックスキーボード（独立キーとしても利用可）
- ・ RS232 I/F
- ・ 簡易 RS485 I/F
- ・ ユーザーLED x 8
- ・ MCU の全ての IO を引き出されている

■ 外形寸法: 155mm×95mm※突起物は除く

■ 回路図などの資料を公開

■ サンプルソース沢山

1.2 機能配置図



1.2.1 部品レイアウト

外形寸法: 155mm×95mm、全ての部品が正面に配置されています。

正面：

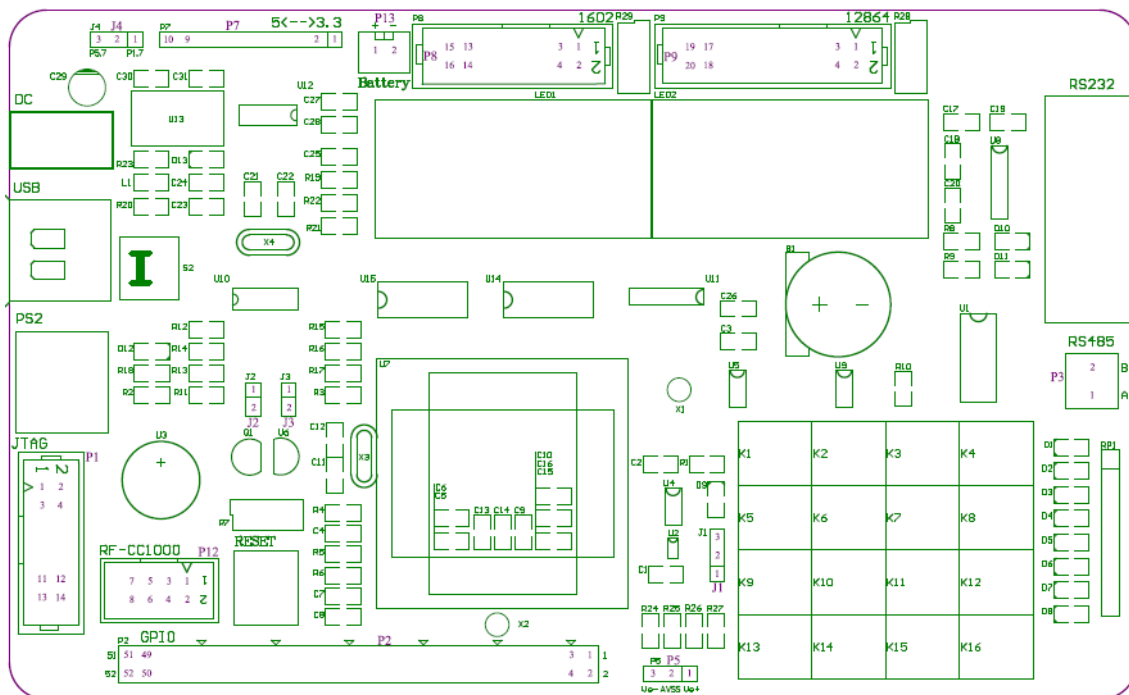


低価格、高品質が不可能？

日昇テクノロジーなら可能にする



1.2.2 ジャンパー及びコネクタのレイアウト



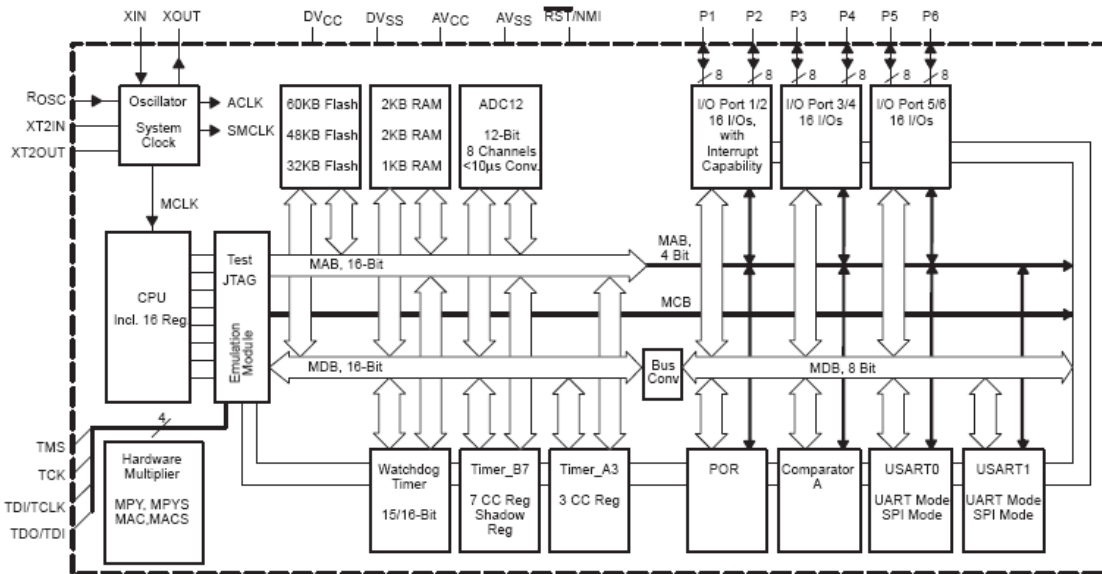
1.3 ハードウェアの構成

部品名称	デバイス	詳細
MCU	MSP430F149	16 ビット超低消費電力マイクロコントローラ
USB	PDIUSBD12	USB2.0サポート、フルスピード (12Mbit/s) サポート
リアルタイム クロック	DS1302	精度は秒まで、閏年自動補充、32Byte 不揮発性の RAM
温度センサー	DS18B20	1-Wire インタフェース、9-Bit~12-Bit プログラマブル分解率
EEPROM	AT24C02	IICインタフェース、256×8-Bit容量
DA 変換	DAC5571	IIC インタフェース、188KSPS 快速更新速度、8-Bit 解像度
レベル変換	SN74LVC4245	3.3V~5V 双方向レベル変換
RS232	MAX3232	3.3V、2 チャンルの RS232 レベル変換をサポート
RS485	SN65176B	半二重モード、送信機も受信機もスリープ状態出力可

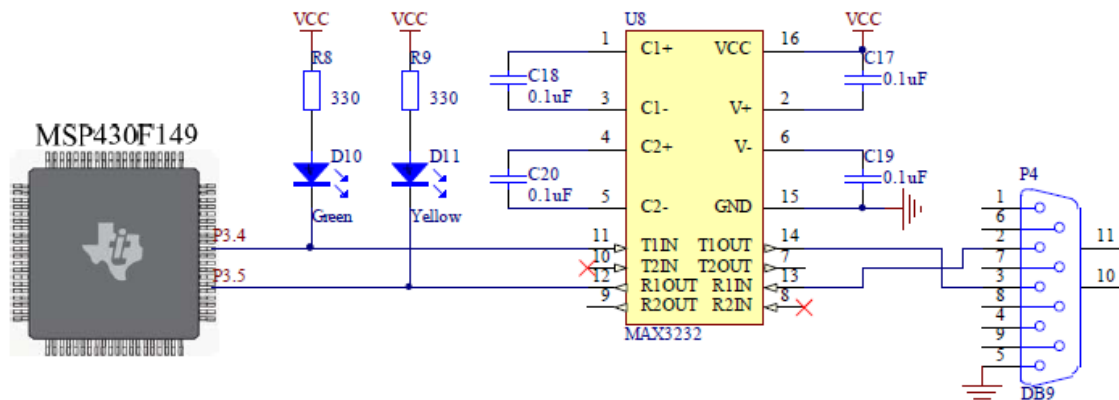
1.3.1 MSP430F149

内部構造：

MSP430x14x



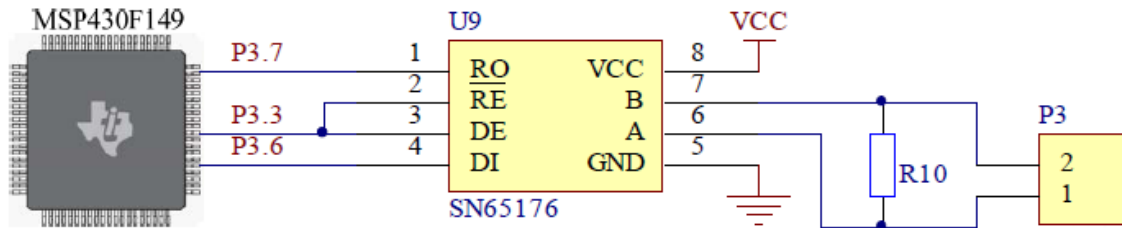
1.3.2 RS232



MSP430F149 には二つの UART を提供しております。本ボードでは UART0 を利用しています。MCU は UTXD0 (P3. 4) を通じて PC にデータを送信して URXD0 (P3. 5) で PC からのデータを受信します。D10 と D11 二つの LED は通信状態を表します。

UART 通信利用しなくて P3. 4 と P3. 5 は普通の GPIO として利用する場合、D10 と D11 二つの LED はユーザー LED として利用できます。

1.3.3 RS485



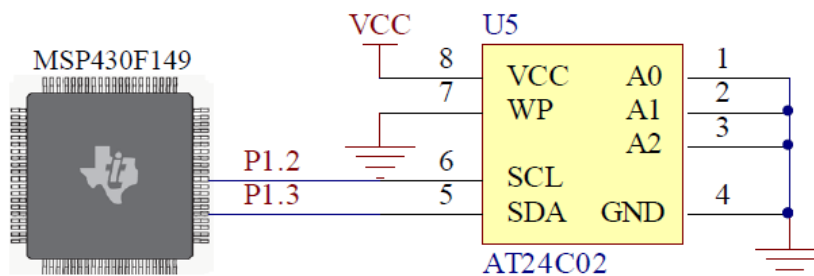
P3. 3がHighレベル時、SN65176Bはデータ發送モード。

P3. 3がLowレベル時、SN65176Bはデータ受信モード。

本ボードではUART1を利用してRS485の通信を実現しています。MCUはUTXD1 (P3. 6)を通じてRS485バスにデータを送信してURXD1 (P3. 7)でRS485バスからのデータを受信します。

※MCUはデータ送信する前に必ずP3. 3をHighレベルに設定してSN65176Bをデータ發送モードに設定する必要。そうする事で、UTXD1 (P3. 6)を通じてRS485バスにデータを送信できます。同じ通り、MCUはデータ受信する前に必ずP3. 3をLowレベルに設定する必要。

1.3.4 EEPROM



AT24C02 の主な仕様：

動作電圧は 2.7V~5.5V、容量は 256×8-Bit、書き込み回数は 100 万回、データ保存期間は 100 年以上。

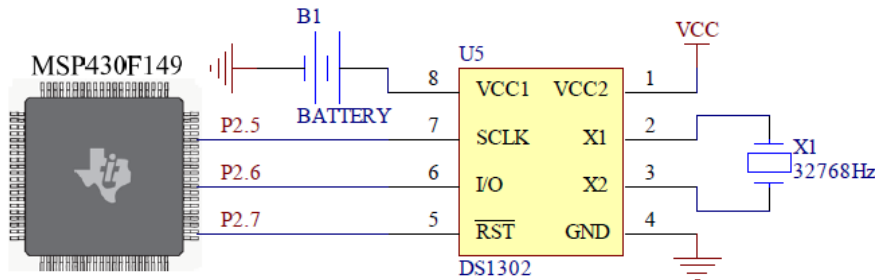
MCU の GPIO ポート P1. 2、P1. 3 が AT24C02 の SCL、SDA と繋いで IIC バスを構成。MSP430F149 の内部に IIC がないので IO ポートで模擬して EEPROM の読み書きを実現します。

EEPROM のアドレスライン A0~A2 は全て Low レベルで、IIC 通信時 EEPROM のスレーブアドレスは唯一で A0~A2 は全て 0 です。

1.3.5 リアルタイムクロック

DS1302 は秒、分、時、日、週、月、年を計算できます。また 2100 年までの閏年の日付を補充できます。2.0V~5.5V の給電、三本線制シリアル通信インタフェース、また 31Byte の RAM を持っています。DS1302 は二つの給電源をサポートします。VCC2 はメイン電源と繋いで、VCC1 は予備バッテリーと繋いでいます。VCC2 の電源を切っても、自動的に VCC1 の給電に

切り替えますので、システムの電源が落ちても DS1302 は正確に時間を計算して RAM 上に保存しているデータを持つことができます。

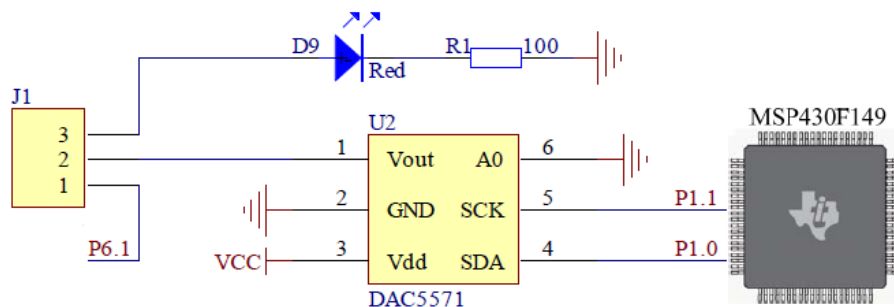


MCU 側は GPIO の P2. 5、P2. 6、P2. 7 ポートを DS1302 の SCLK、I/O、RST を繋いでいます。DS1302 のピン 1 はボードの 3.3V 電源と繋いでメイン電源となります。ピン 8 は CR1220 型番のボタンバッテリーのプラス端と繋いで予備電源になります。

1.3.6 DAC

DAC5571は低消費電力、シングルチャネル、8-Bit解像度のバッファ電圧出力DACです。リセット後の電圧は0V。2.7V~5.5Vの給電電圧、IICインタフェースでMCUと接続しています。

3.4Mbps通信速度をサポート、サンプリング率188KSPS。

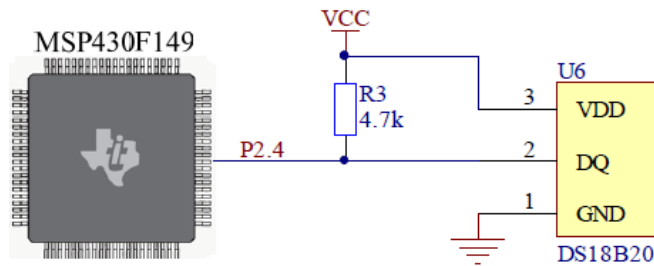


MCUのP1.0、P1.1ポートとDAC5571のSDA、SCKポートを接続して、GPIOでIICを模擬してDACの操作をします。J1の2と3をショートすると、DACの出力でLED D9を点灯します。D9の明るさでVoutの電圧の変化が分かります。J1の2と1をショートすると、MP430内部のADCでDACの出力をサンプリングして変換します。直接電圧計でJ1の2とGNDの電圧を測るとDACからの出力電圧値が正確に分かります。

1.3.7 温度センサー

DS18B20 の主な仕様：

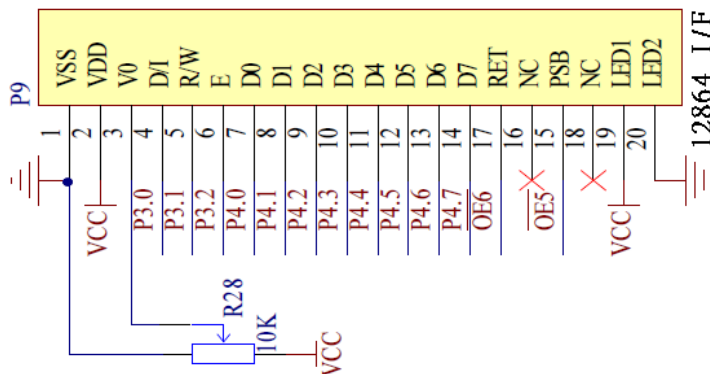
温度検査範囲：-55℃~+125℃（-10℃~+85℃の間に±0.5℃の精度）、1-Wire インタフェース、9-Bit~12-Bit プログラマブル分解率



MCU の I/O ポートで 1-Wire プロトコルで DS18B20 の読み書きを実現します。

1.3.8 12864 液晶インタフェース

本ボードは 2.54 ピッチ 20 ピンのコ 12864 液晶コネクタを提供しております。ST7920 ドライバの任意の 12864 モジュールを接続できます。

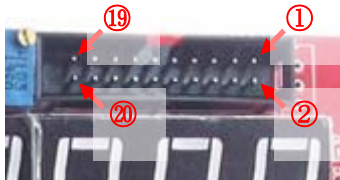


15 番 PSB はデータ転送モード選択位で、High レベルの場合はパラレルモード、Low レベルの場合はシリアルモードになります。これは 74HC595 (U11) の QC ポートと繋いでいて、QC ポートの制御によりデータ転送モードを選択できます。

17 番の RET はリセット、74HC595 (U11) の QB ポートと繋いでいて、QB ポートの制御によりリセット状態を実現します。

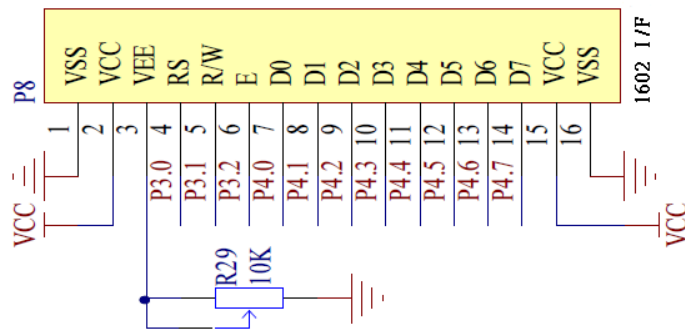
注意事項：

- 1、液晶の操作の前、必ずデータ転送モードを設定して、一回リセット動作が必要。
- 2、液晶のコントロールポートのロジック電圧は 3.3V、バックライトの電圧も 3.3V。
- 3、インタフェースのピン配置：

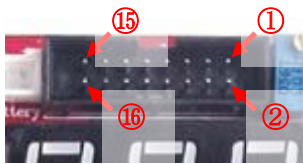


1.3.9 1602 液晶インタフェース

本ボードは 2.54 ピッチ 16 ピンのコ 1602 液晶コネクタを提供しております。

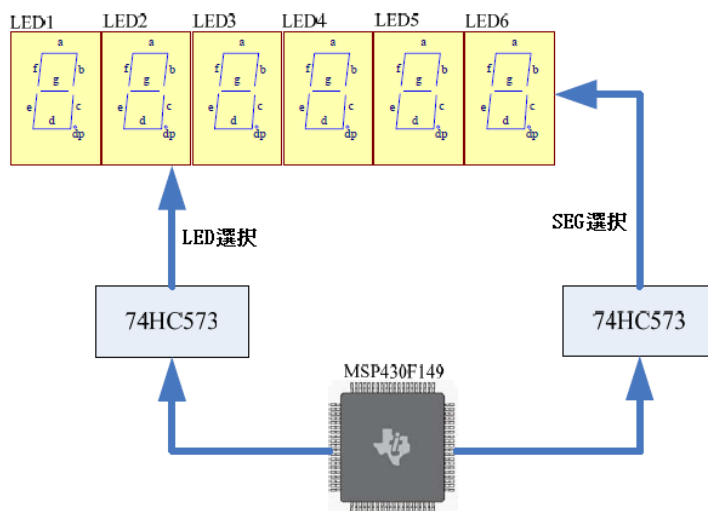


インタフェースのピン配置：



1.3.10 7SEG LED

MSP430F149 は低消費電力で、IO のドライブ能力が高くないため、7SEG LED と MCU の間に 74HC573 を入れています。



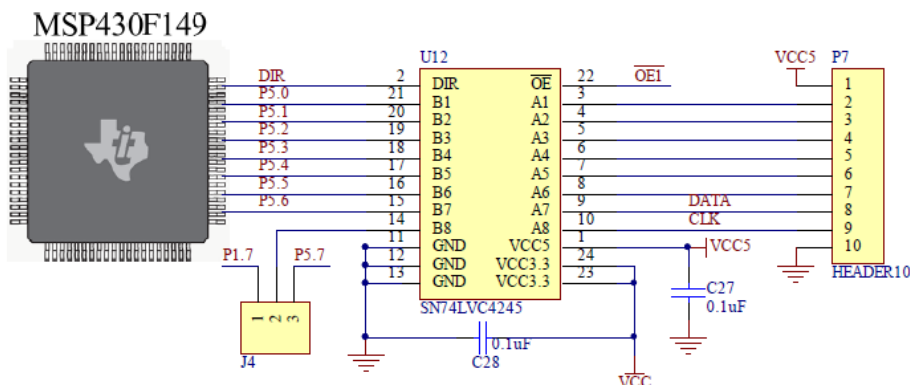
LED選択信号	LED1	LED2	LED3	LED4	LED5	LED6
MCUポート	P5.0	P5.1	P5.2	P5.3	P5.4	P5.5

LED 選択信号は Low レベル有効。

SEG選択信号	a	b	c	d	e	f	g	dp
MCUポート	P4.0	P4.1	P4.2	P4.3	P4.4	P4.5	P4.6	P4.7

SEG 選択信号は High レベル有効。

1.3.11 レベル変換回路



A1~A8 は 5V レベルデータポートと繋いで、B1~B8 は 3.3V レベルデータポートと繋がります。
OE は Output Enable 信号 (Low レベル有効)、DIR は変換方向を制御する。

DIR が Low の場合データ方向は $B \rightarrow A$ 。

DIR が High の場合データ方向は $A \rightarrow B$ 。

SN74LVC4245 の B1~B7 は MCU の P5.0~P5.6 ポートと繋いでいますが、B8 はジャンパーで P5.7 或いは P1.7 と繋がります。P1.7 と繋いだときは PS2 からのクロック信号で割り込み発生できます。PS2 利用しない場合は P5.7 と繋いで、8bit のレベル変換となります。

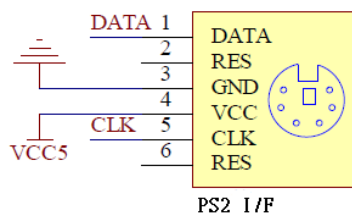
74LVC4245 の OE と DIR 信号は 74HC595 のパラレル出力と繋いでいます。

注意事項：レベル変換機能利用しない場合は必ずこのモジュールの出力を禁止する必要があります（0E を High レベルに設定して置く）。でなければ他の機能に影響を与える恐れがあります。

1.3.12 PS2 インタフェース

本ボードは標準 6 ピンインタフェースの PS2 を提供しています。標準キーボード、マウスを接続できます。

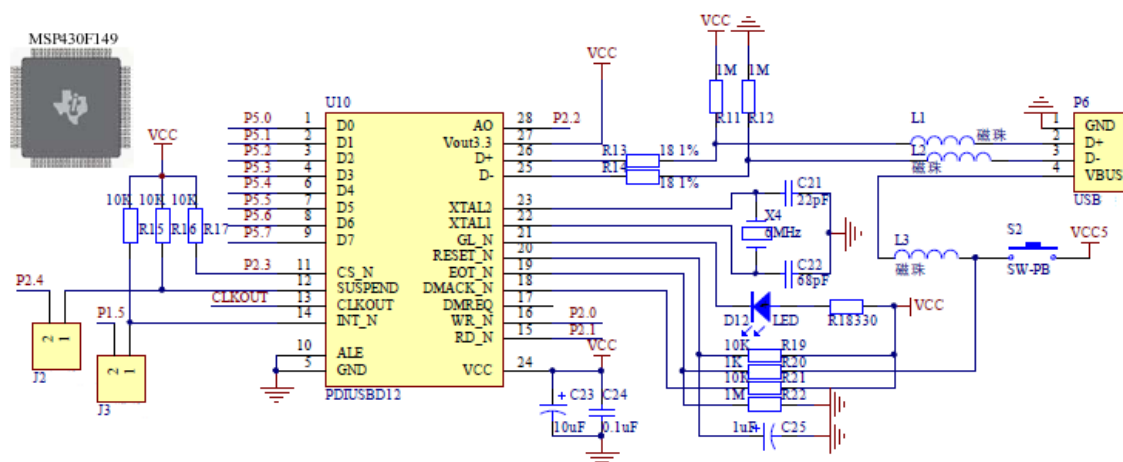
キーボード、マウスは 5V 給電で、MSP430F149 は 3.3V で動きますので、間にレベル変換が必要です。PS2 の CLK 信号を SN74LVC4245 の A8 ポートと接続して PS2 の DATA 信号を SN74LVC4245 の A7 ポートと接続しています。SN74LVC4245 を利用してレベル変換を実現しています。回路図は下記：



注意事項:PS2 を利用する前に 74LVC4245 の OE を Low、DIR ポートを High に設定する必要。

1.3.13 USBインターフェース

PHILIPS 社製の PDIUSB12 を利用して USB ドライバ実装しています。

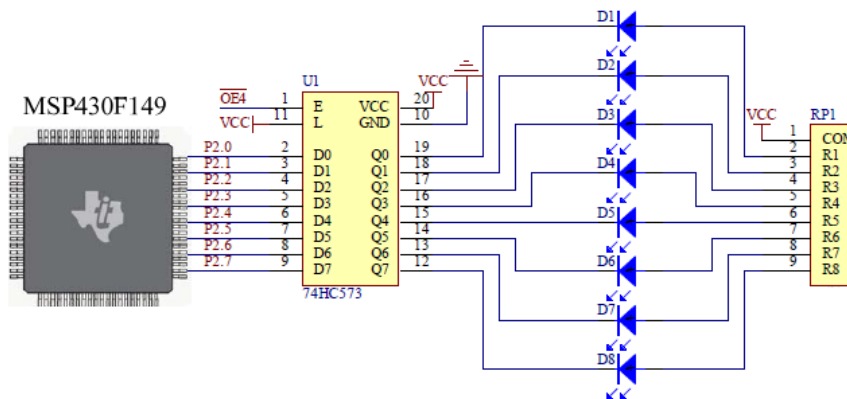


PDIUSB12 のデータポートは MCU の P5 ポートと繋いで、コントロールポートは P2.0→WR_N, P2.1→RD_N, P2.2→AO, P2.3→CS_N と繋いでいます。PDIUSB12 の INT_N ポートはジャンパーを通じて MCU の P1.5 ポートと繋いで、MCU に割り込み要求時に利用します。ジャンパーを利用するのは INT_N ポートを Low レベル設定時キーボード回路に影響しない様にする為です。USB 回路利用しない時は J3 のジャンパーをオープンして置くのをお勧めです。

LED D12 は PDIUSB12 から制御しています。MCU と PC を接続時に点滅して通信状態を示します。接続後は点灯していて、接続有効状態を示します。

USB でボードに給電できます。USB ケーブルで PC と繋いで、USB 電源 SW でオン/オフ制御できます。

1.3.14 LED

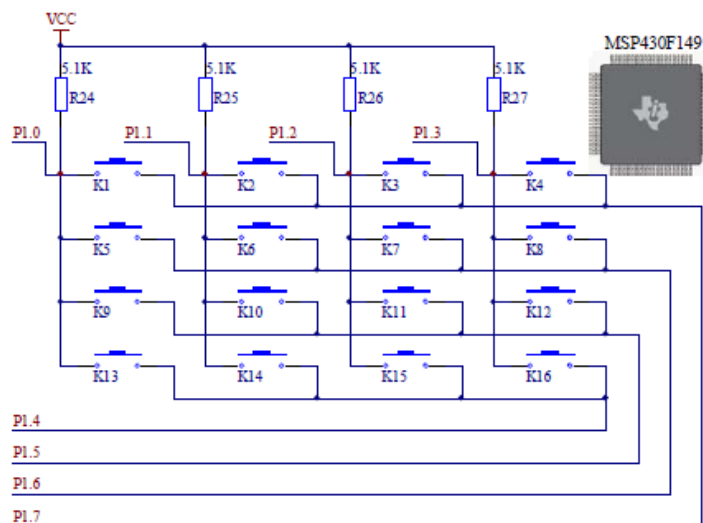


74HC573 を通じて MCU の P2 ポートの各 IO で 8 つの LED を制御しています。IO が Low レベルの時相応の LED が点灯して、High レベルの時消灯します。

74HC573 の Enable ポートは 74HC595 の OE4 と繋いでいて、このポートが 74HC573 の出力が有効か制御します。LED 回路利用する場合、OE4 を Low レベルに設定する必要があります。利用しない場合は、他の回路から LED を影響しない様に OE4 を High レベルに設定するのをお勧めします。

1.3.15 マトリックスキーボード

本ボードでは 4x4 のマトリックスキーボードと 4 つの独立キーを複用する回路を提供しております。例えば P1.7 が Low レベルでしたら K1~K4 は P1.0~P1.3 ポートと繋いでいる独立キーとなります。独立キーが必要ない場合、P1 ポートをスキャンすると 4x4 のマトリックスキーボードとなります。



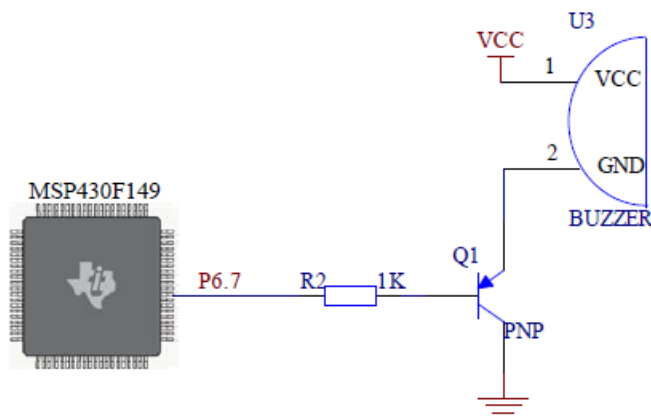
1.3.16 ADC回路

MSP430F149 内部では 12-Bit の ADC を持っていて、外部に 8 チャンネルを提供しております。IO の P6.0~P6.7 ポートになります。本ボードでは P6.0 ポートは既に利用しています R7 のピン 2 と繋いでいて、ボリューム調整で P6.0 ポートの電圧を調整できます。P6.1 ポートはジャンパー J1 の 1 と 2 をショートするかで DAC5571 の出力ポートに接続するか決められます。他のチャンネルポートは空いていますのでユーザーの方で自由に利用できます。

また、MCU 内部搭載している ADC は外部参考電圧入力もサポートします。P5 で接続できます。P5 の各ピン（右側口で囲まれているのが 1）と MCU の外部参考回路のピンの対応関係：1--VeREF+、2--AVSS、3--VeREF-。

注意事項：外部参考電圧を利用する場合、プログラミング時必ず内部参考電源を閉じる必要。MCU が損害される恐れがありますので、同時に存在できません。

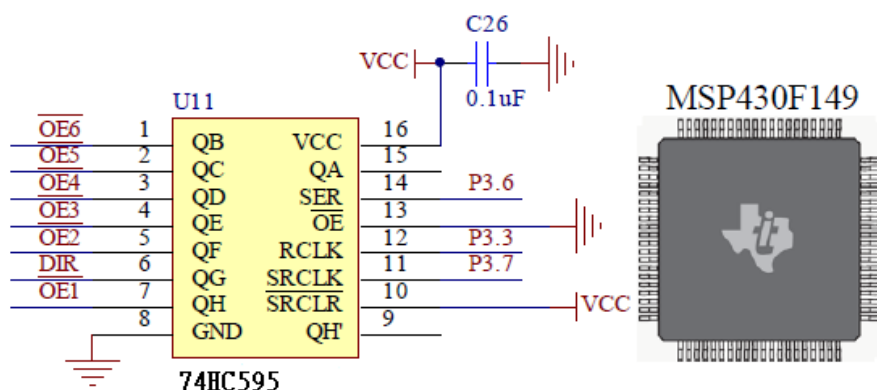
1.3.17 ブザー回路



P6.7 が High レベルを出力するとブザーが鳴って、Low レベルを出力するとブザーが閉じられます。

1.3.18 リソースコントロール回路

本ボードは 74HC595 を利用してリソースの複用を実現しています。3 つの IO で複数のチップの Enable を制御します。



QH~QB ポートと各リソースとの対応関係は下記の表：

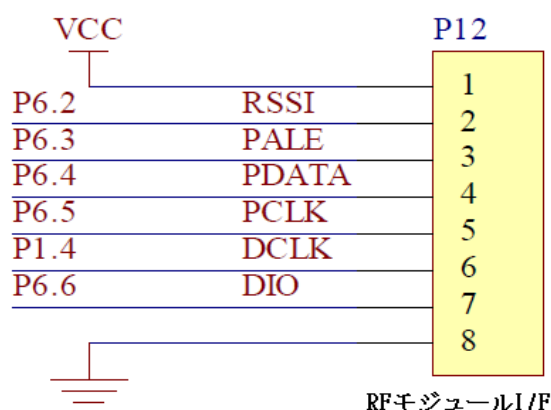
ポート	レベル	コントロール
QH	High	74LVC4245出力禁止、5Vポートと3.3Vポートを切断させる
	Low	74LVC4245 出力許可、5V ポートと 3.3V ポートのレベル変換許可
QG	High	レベル変換方向：5V→3.3V
	Low	レベル変換方向：3.3V→5V
QF	High	U14出力禁止、7SEG LEDのセグ選択信号を禁止
	Low	U14 出力許可、7SEG LED のセグ選択信号を許可
QE	High	U15 出力禁止、7SEG LED の LED 選択信号を禁止
	Low	U15 出力許可、7SEG LED の LED 選択信号を許可
QD	High	U1 出力禁止、LED ドライブ信号を禁止
	Low	U1 出力許可、LED ドライブ信号を許可
QC	High	12864液晶のデータ通信方式をパラレルに設定
	Low	12864 液晶のデータ通信方式をシリアルに設定
QB	High	12864を動作状態に設定
	Low	12864 をリセット状態に設定

※QH が High レベルの場合 QG は無効です。

MCU 側の 3 つの IO の制御については各サンプルソースにある BoardConfig.h ファイルの BoardConfig() 関数をご参照ください。

1.3.19 RF微弱モジュールインタフェース

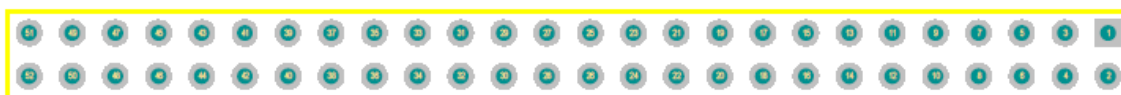
本ボードは CC1000 (433MHz) モジュールを基の RF 微弱モジュールインタフェースを提供しています。



RF モジュールを利用しない場合は、このインタフェースは普通の IO として利用できます。

1.3.20 GPIO

本ボードではMCUの48のピンを全て引き出して2.54ピッチの拡張ピンで提供しています。
ピン配置は下記の様：



□で囲まれているのが1番。同じ行なのはすべて奇数で1～51番、下の行は全て偶数で、1番の真下は2番で、2～52番の順となります。

拡張 IO と MCU のピン対応関係：

1	GND	10	P1.7	19	P3.0	28	P4.1	37	P5.2	46	P6.3
2	VCC	11	P2.0	20	P3.1	29	P4.2	38	P5.3	47	P6.4
3	P1.0	12	P2.1	21	P3.2	30	P4.3	39	P5.4	48	P6.5
4	P1.1	13	P2.2	22	P3.3	31	P4.4	40	P5.5	49	P6.6
5	P1.2	14	P2.3	23	P3.4	32	P4.5	41	P5.6	50	P6.7
6	P1.3	15	P2.4	24	P3.5	33	P4.6	42	P5.7	51	GND
7	P1.4	16	P2.5	25	P3.6	34	P4.7	43	P6.0	52	VCC
8	P1.5	17	P2.6	26	P3.7	35	P5.0	44	P6.1		
9	P1.6	18	P2.7	27	P4.0	36	P5.1	45	P6.2		

1.3.21 JTAGインタフェース

本ボードでは TI 社の 14PIN (2.54 ピッチ) 標準 JTAG インタフェースを提供しています。
各ピンの定義：



1	2	3	4	5	6	7	8	9	10	11	12	13	14
TDO	VCC	TDI		TMS		TCK		GND		RST			

1.4 サンプルソースについて

Examples.zip に下記サンプルソースは含まれている。

1.4.1 基本テスト

Examples#0

- ADC
- Clock Out
- Comparator
- FLASH
- HFXT
- LPM3
- SoftwareFLL
- TimerA3
- TimerB7
- ToggleIO
- UART
- WatchDog

1.4.2 モジュール基本プログラム






#Examples#1

- 3.3--5V
- ADC
- BUZZER
- Cry1602
- Cry12864
- DAC
- DS18B20
- DS1302
- EEPROM
- KEY
- LED
- MyUcos-II
- Nixietube
- PS2
- RS232
- RS485
- USB



1.4.3 総合テスト

Examples#2

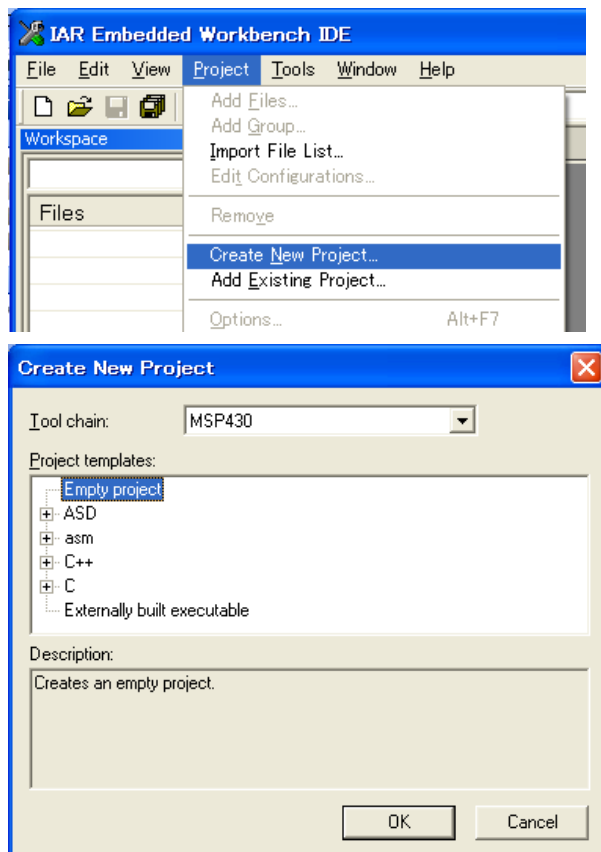
-  AD_DA
-  Interface
-  KeyBorad
-  SSCOM Test
-  Temp_Time

第二章 IAR環境でソフトウェアの開発

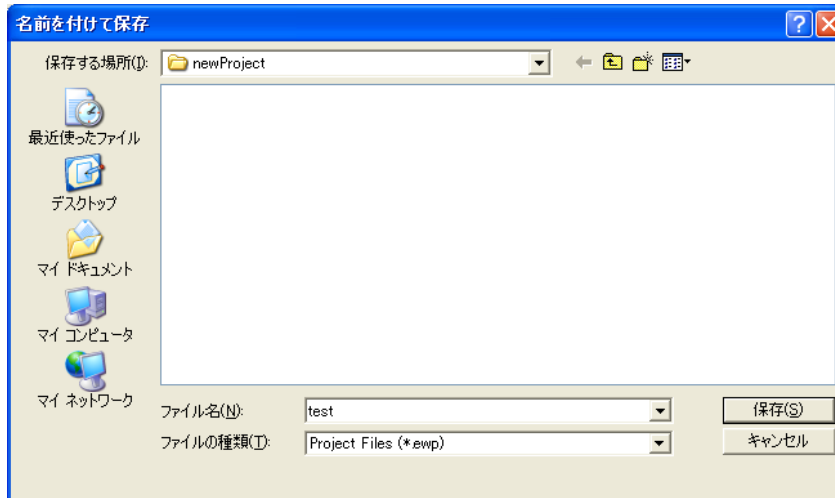
IAR Embedded Workbench (Ver4.20) 環境で説明します。

2.1 新しいプロジェクトの作成

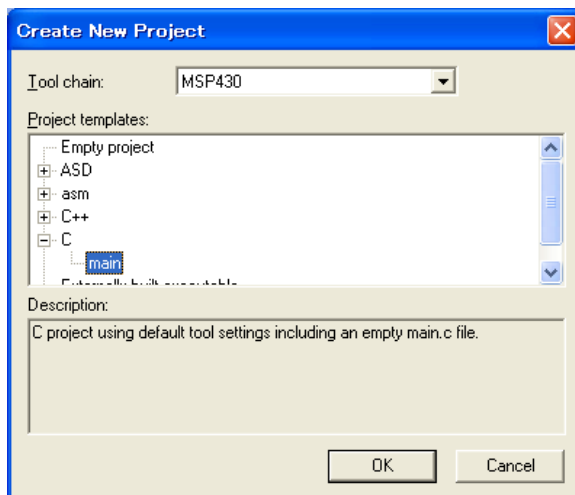
Project→Create New Project をクリックする。



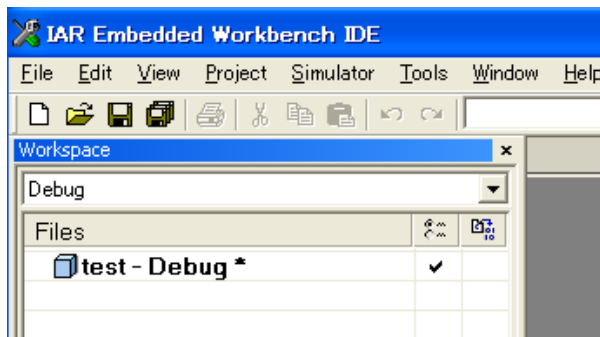
Empty project を選択して OK をクリックして、フォルダを選択し、test というプロジェクトを作成する。



Empty project を選択する代わりに+C を開いて main を選択して OK を押しても同じです。



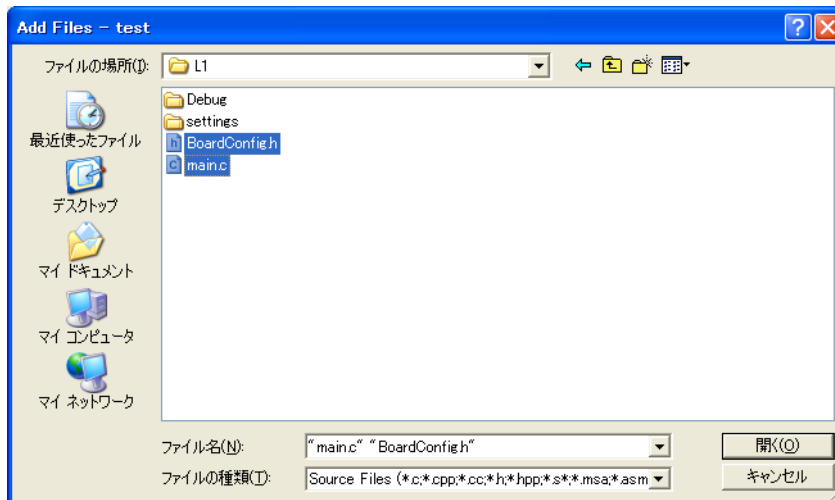
Workspace に test というプロジェクトがあります。Debug モードと Release モードがありますが、最初シミュレーションとデバッグが必要な段階では Debug モードに設定する必要があります。



File→Save Workspace でプロジェクトが保存されます。

2.2 ソースファイルの追加

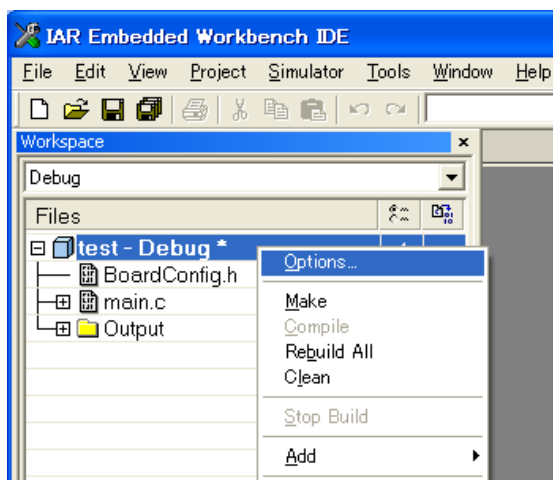
既に編集したファイルを追加する場合は Project→Add Files で追加する。

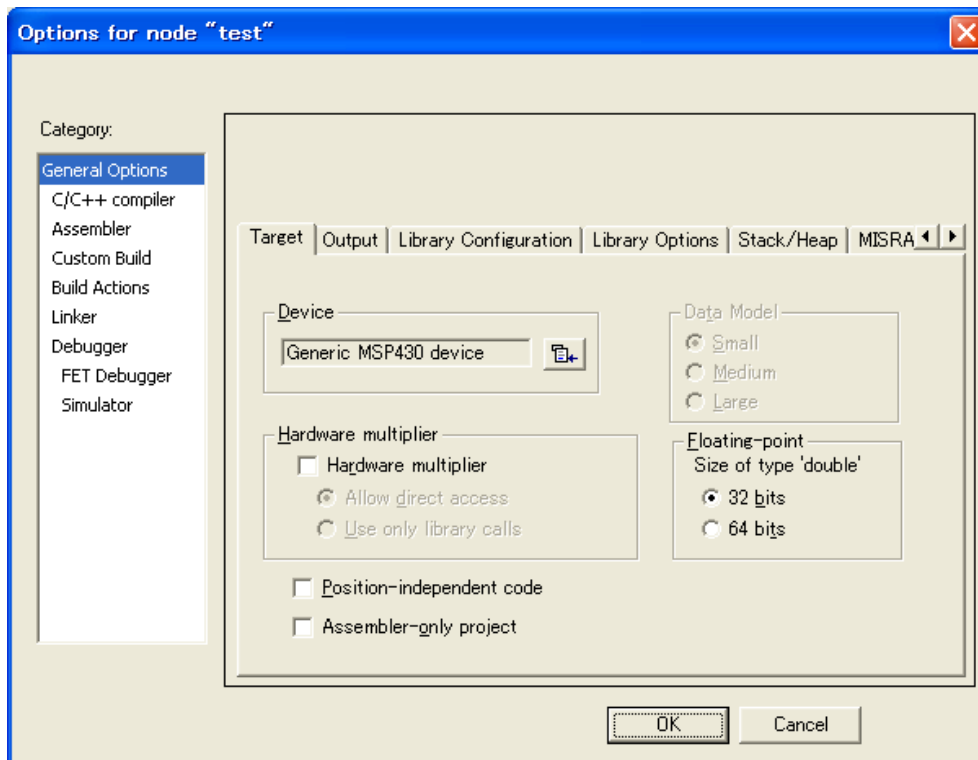


新たに編集する場合は File→New→File で新しいファイルを作成して、編集した後、File→Save で保存すれば良いです。


2.3 プロジェクトオプションの設定

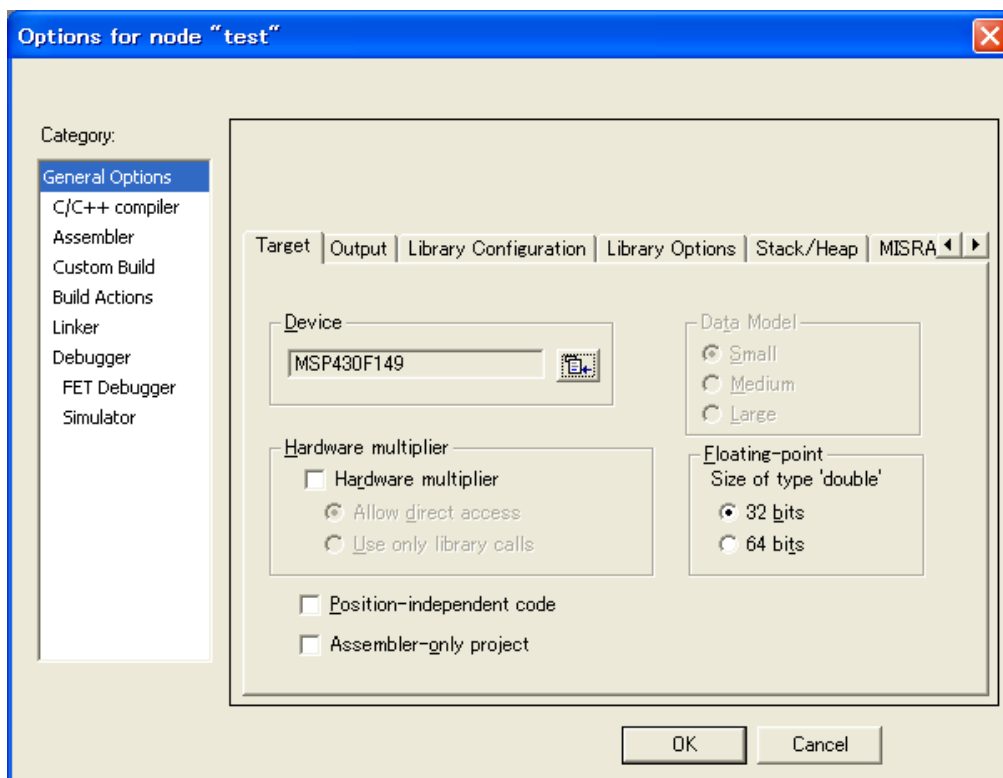
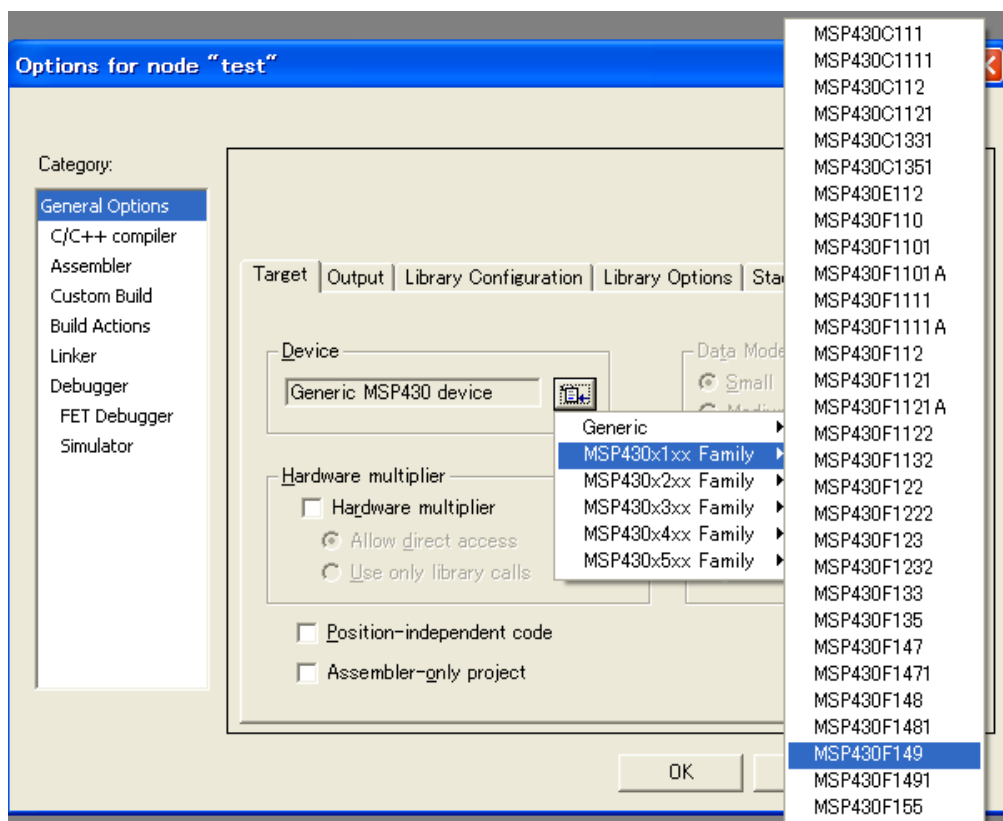
Project→Options をクリックする、或いは Workspace のプロジェクト名で右クリックする。



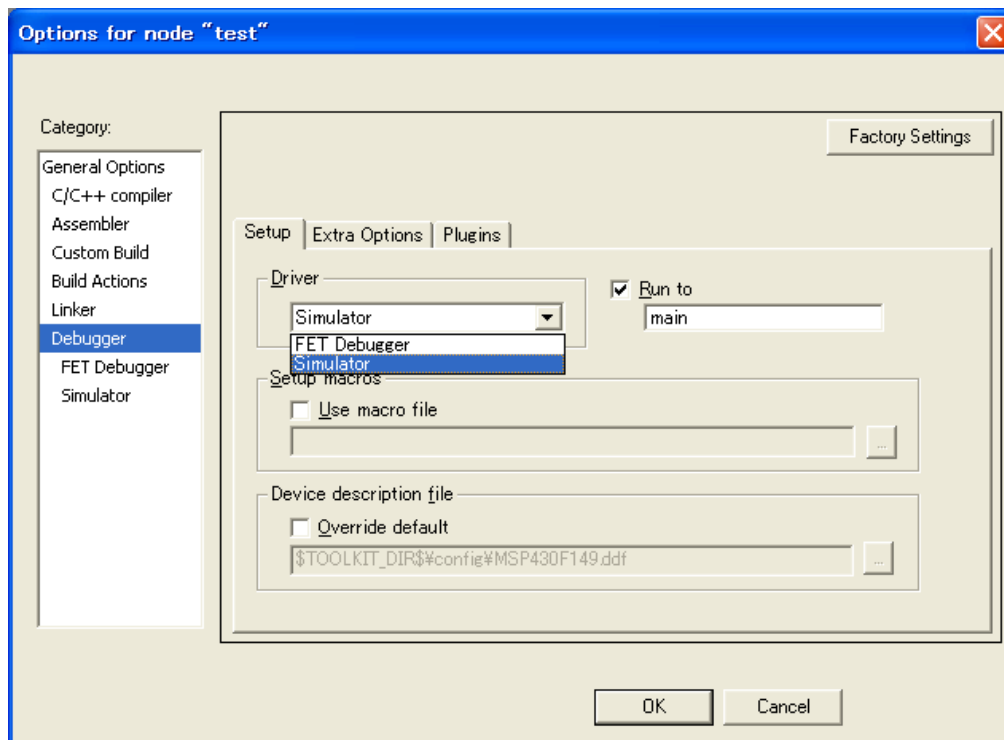


デフォルトの設定で大部分のニーズに満たしますが、下記 2 箇所だけ変更すれば良いです。

- 1、General Options の Target タブの Device 欄の  をクリックして対応している CPU の型番を選択します。



2、Category の Debugger をクリックします。

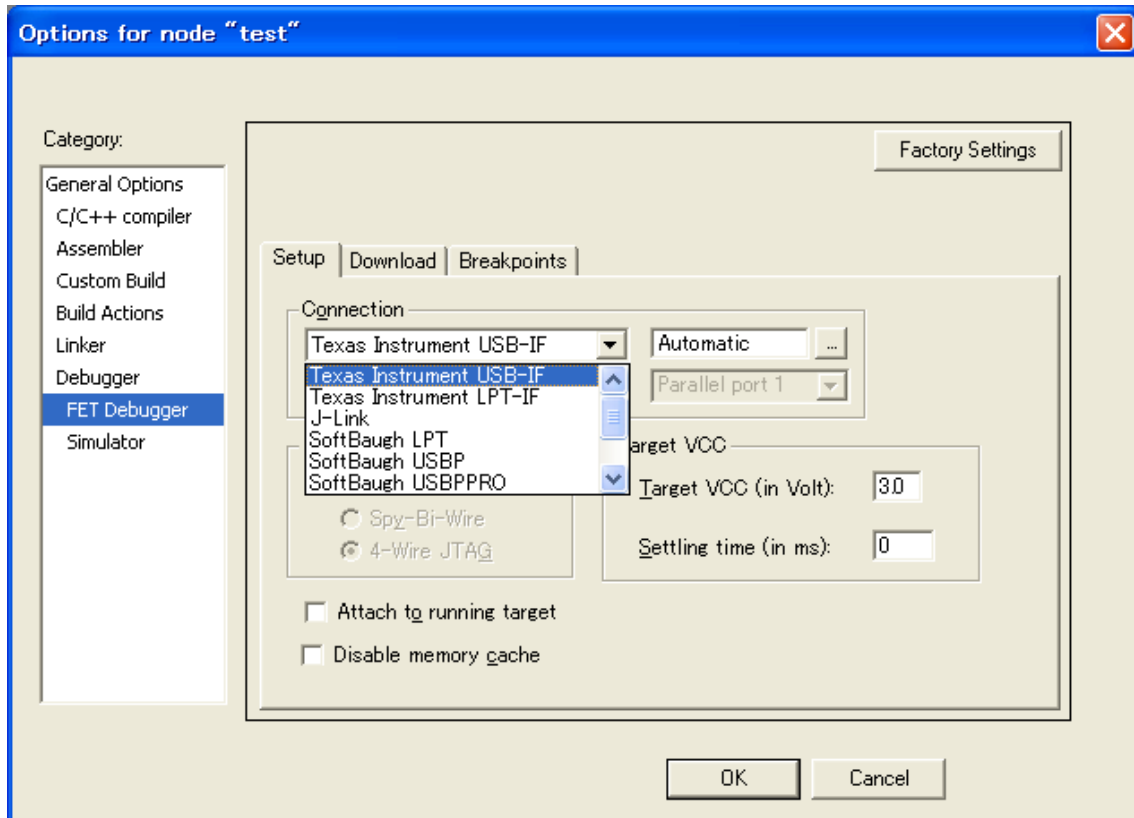


Driver の選択欄に Simulator と FET Debugger があります。

Simulator はソフトウェアでシミュレーション用で、FET Debugger はエミュレーターで PC と MCU を繋いでデバッグ用です。

シミュレーターだけしたい場合は Simulator を設定して OK をクリックしてオプション設定完了です。

デバッグ必要な場合は FET Debugger を設定して、Category の Debugger の FET Debugger をクリックします。




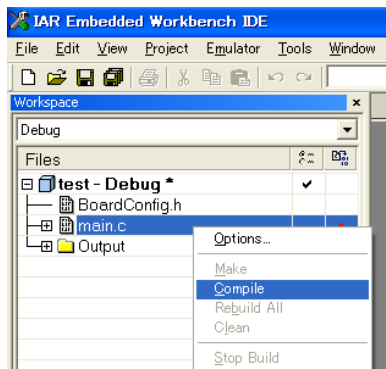
弊社提供しているMSP430 USBエミュレーター（製品紹介URL：

<http://www.csun.co.jp/SHOP/2010082301.html>）を利用する場合はデフォルトのTexas Instrument USB-IFを選択してOKをクリックします。


本ボード付属しているLPT簡易エミュレーターを利用する場合はTexas Instrument LPT-IFを選択してOKをクリックします。

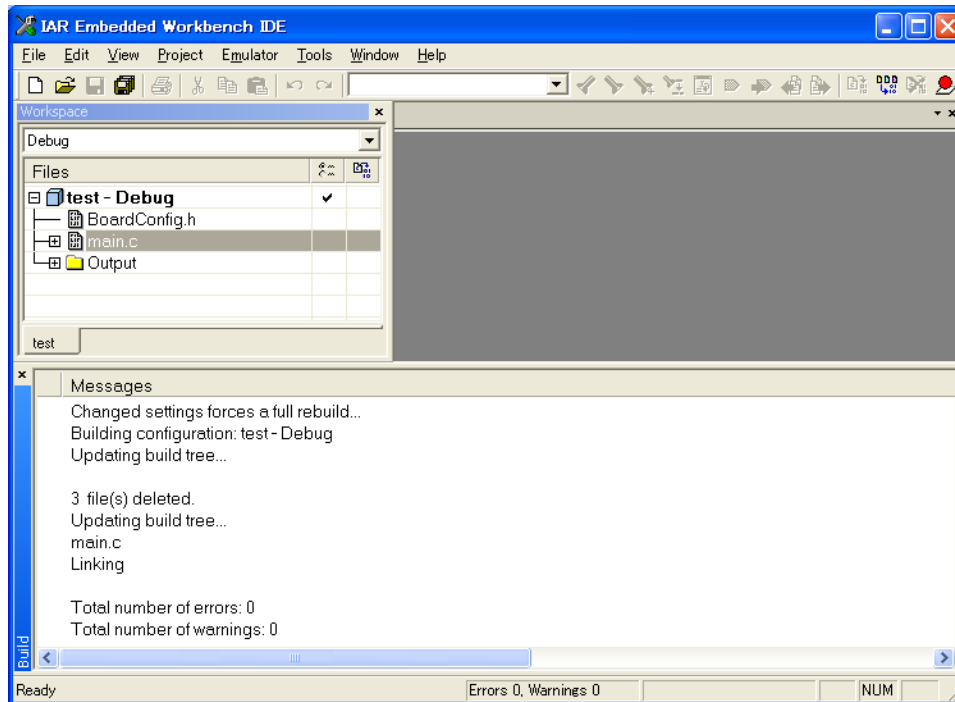
2.4 コンパイル

上記設定完了後、任意のソースファイル(.c、.cpp、.cc、.s、.asm、.msa)を右クリックしてCompileを選択、或いはProject→Compileをクリック、或いはをクリックしてコンパイルします。




2.5 Make

全てコンパイル通して Project→Make 或いは  をクリックして Make します。



2.6 デバッグ

Make 完了後デバッグが出来ます。Project→Download and Debug 或いは  をクリックしてデバッグモードに入ります。

デバッグに関してもっと詳しい情報は Help などの資料をご参照ください。