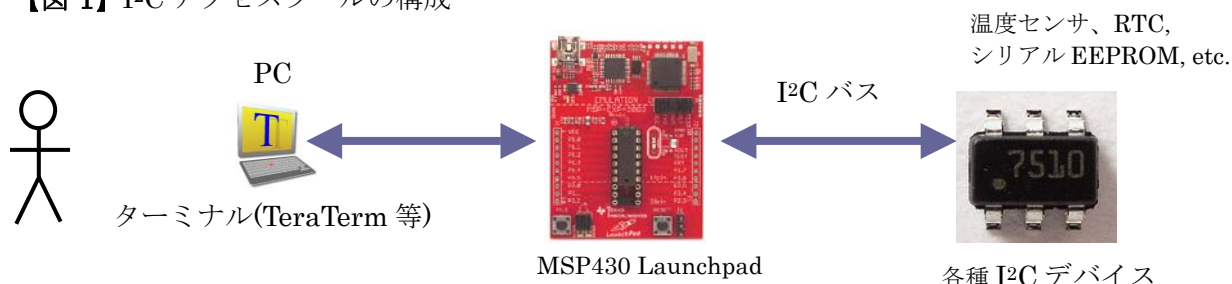


MSP430 Launchpad による I²C デバイス アクセスツール

espilab

各種センサ IC やシリアル EEPROM、LCD モジュールなど、マイコン周辺 I/O を拡張する上で、今や I²C バスは欠かせない存在になりました。ただ、これらの I²C デバイスに対して、MS-DOS 時代の SYMDEB のように、試しに read/write してみるといった手軽な手段がありません。そこで、シリアル端末と TI 製マイコン評価ボードの MSP430 Launchpad を利用して、I²C デバイスにアクセスできるツールを作りましたので、紹介いたします。(図-1)

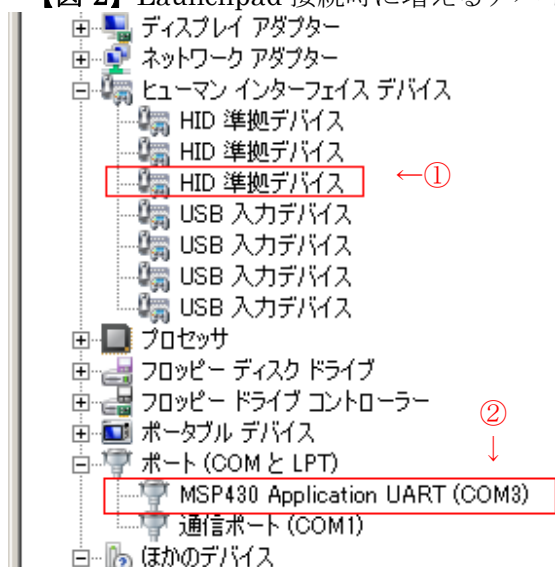
【図-1】 I²C アクセスツールの構成



MSP430 Launchpad のシリアル機能について

TI 製マイコン評価ボードの MSP430 Launchpad (以下 "Launchpad") を、必要なドライバ類をインストール済みの PC と USB ケーブルで接続すると、2 つのデバイスとして認識されます。(図-2)

【図-2】 Launchpad 接続時に増えるデバイス



ターゲットマイコンへのプログラムの書き込みや実行・停止などのデバッグ(エミュレーション)機能は、HID デバイス①を通して制御されます。

「ポート」に現れているデバイス②「MSP430 Application UART (COMx)」は、シリアルポートとしてターミナルソフト等でアクセス可能です。

Launchpad 上のジャンパ J3 には、TXD, RXD と書かれたシルク文字があり、試しにターミナルソフトでキーを叩きながらオシロスコープで波形を観測すると、片方には PC からの送信データが届いている事が分かります。

上記シリアル信号をターゲットマイコンのシリアルポートで受け取り、ターミナルの相手をするようなプログラムを MSP430 マイコン上に組めばよいのですが、ここで何点かの注意事項があります。

Launchpad の Rev.1.4 までと Rev.1.5 以降の違い

まず、今回の目的のためには UART 機能と I²C バス機能が同時に使えるマイコンが必要です。

Launchpad は現在も販売中ですが、途中で製品バージョンが変わっています。(図-3)

【図-3】 Launchpad に同梱されているチップと機能概要

MSP430 LaunchPad	添付チップ	Flash	RAM	Timer_A	COMP_A+	ADC10	USI	USCI	Temp	Package
							I2C/SPI	I2C/SPI/UART	Sensor	
旧タイプ	MSP430G2211	2KB	128	1	1					14-PDIP
	MSP430G2231	2KB	128	1		1	1		○	14-PDIP
新タイプ	MSP430G2452	8KB	256	1	1	1	1		○	20-PDIP
	MSP430G2553	16KB	512	2	1	1		1	○	20-PDIP

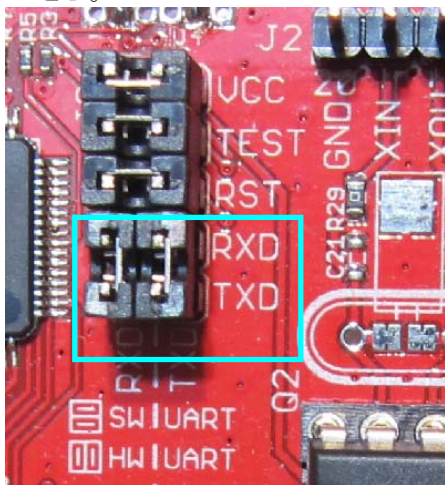
Rev 1.4 までと、2012 年から販売された Rev 1.5 以降では、同梱されているマイコンチップが違います。この中で、UART と I²C を同時に使えるのは、MSP430G2553 (以下,G2553)になりますので、今回はこれを使います。

また、Launchpad の基板にも注意点があります。G2553 を使うとき、エミュレータ回路の TXD と、マイコン側の TXD ピンとがぶつかっているのです。この配線ミスには TI も気づいたようで、Rev 1.5 からは、J3 のヘッダピンに付けるジャンパの向きを変えて挿す事で、TXD と RXD を交叉させて、エミュレータ回路とマイコンとが通信できるように修正されています。 そのように設定したジャンパの様子を写真-1 に示します。 なお、Launchpad の回路図は公開されています(→資料(1))

なお、Launchpad を購入する際、Rev 1.5 かそうでないかは、パッケージに貼られたシールで確認できます。(写真-2)

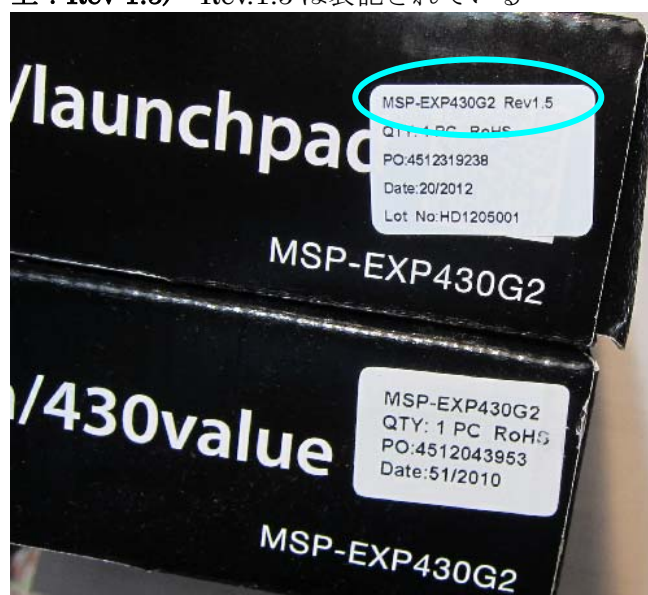
【写真-1】 TXD,RXD をひねるジャンパ接続

青枠の 2 つのジャンパは向きを 90° 変えて差し込む。



【写真-2】 パッケージ外箱のシール (下 : Rev1.4

上 : Rev 1.5) Rev.1.5 は表記されている



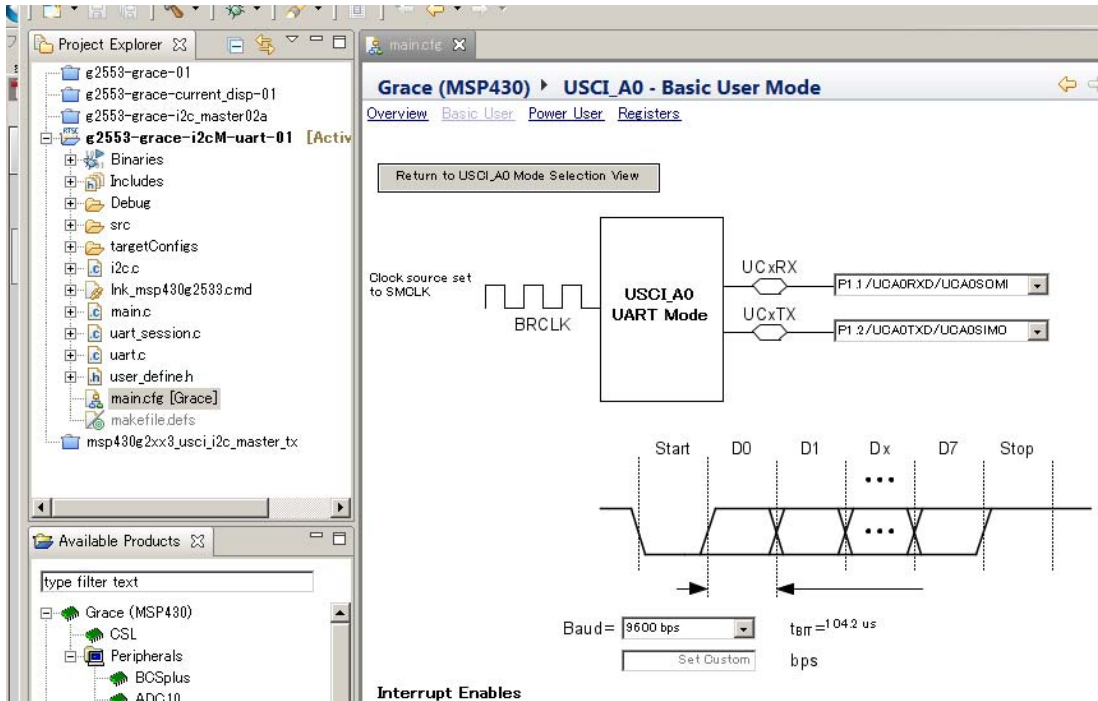
プログラム開発には CCS & Grace を利用

プログラミングには Code Composer Studio v5.5 (以下 CCS)と、CCS 上で使える自動コード生成ツール、「Grace」を利用しました。 GUI 画面で、マイコン内部の各モジュールのうち、使用するユニットを選択し、どのような動作モードで使用するか、クロック設定はどうするか等のパラメータを設定すると、その指定通りの初期化ルーチンや、割り込みルーチンの枠組みのコードを生成してくれます。

UART や I²C に実際にデータ送信/受信するルーチンについては、サンプルコードの説明はありますが、自分で組む必要があります。 他社製の類似のワンチップマイコンの開発ツールでは、正しく

動く送受信ルーチンのコードまで自動生成され、プログラマは関数を呼ぶだけでよいといった物もありますが、Grace はまだそこまではサポートしていないようです。今後のバージョンアップによる発展が期待されます。画面の一例を 図-4 に示します。

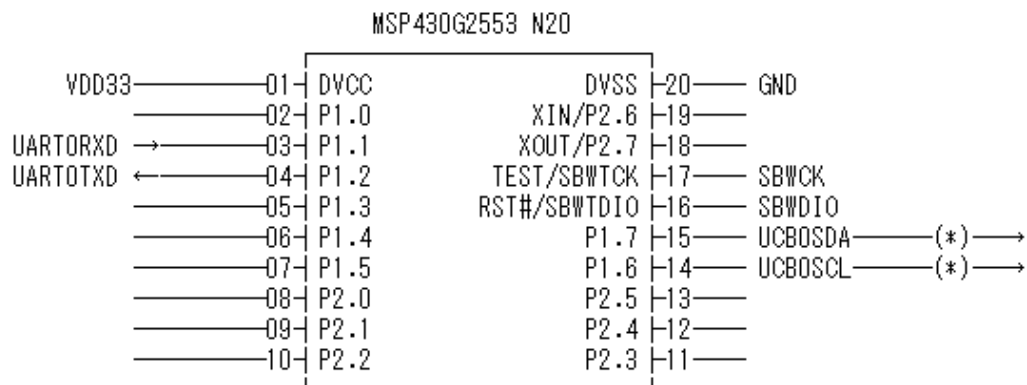
【図-4】 CCS 上の Grace の様子 USCI_A0 の設定画面例。



回路図...のようなもの

Launchpad を利用し、I²C は 2 線しか使わないので、回路図は罫線を使ってメモした程度です。G2553 の 14 ピンから SCL、15 ピンから SDA を引き出し、アクセスしたいデバイスに接続します。プルアップ抵抗が必要です。

【図-5】 回路図



(*) → 10KΩ でプルアップ

I²C アクセスツールの使いかた

プログラムのソースコード一式については、筆者 Web サイトにて公開いたします。(文末 資料(2))
ここでは、プログラムの使用感のみ紹介したいと思います。シリアル EEPROM (BR24L64, ローム) に書き込み／読み出し操作をする例を示します。

TeraTerm などのターミナルソフトで Launchpad の COM ポートに 9600bps でアクセスし、リタ

エンキーを叩くとプロンプト”>”がでます。「? <enter> 」と入力するとヘルプメニューが出ます。(図-6a)

Launchpad 側は I²C マスタです。最初に、対象のデバイスのスレーブアドレスを設定します。今回の EEPROM の I²C アドレス 7bit は、”1010000” であり、これを MSB 側に詰めて “A0” を指定します。(図-6b)

<pre>【図-6a】 ヘルプメニュー >? a <adr(hex)> : set slave address s xx [xx] .. : set byte(s) data to write c <bytes(hex)> : set count bytes to read r : read access w : write access p : show current parameters 0 : reset i2c I/F P : output stop condition ? : show help ></pre>	<pre>【図-6b】 スレーブアドレス設定 >a a0 Set slave address = 0xA0 ></pre>
--	---

この EEPROM は書き込み手順としては以下のようにアドレスとデータ列を送信すれば書き込めます。

(Write アクセスで) <アドレス上位バイト>, <アドレス上位バイト>, <データ 1 バイト目>, <データ 2 バイト目>, ...
読み出したい場合は、2 バイトの先頭アドレスをまず Write アクセスで書き込み、続いて Read アクセスすれば、指定したアドレスからの内容が順番に読み出せます。

(Write アクセスで) <アドレス上位バイト>, <アドレス上位バイト>

(Read アクセスで) <データ 1 バイト目>, <データ 2 バイト目>, ...

EEPROM に書き込んで読み出している例を示します。

<pre>【図-6c】 EEPROM へ書き込みと読み出し例 >s 00 00 ←書き込みする 2 バイトを設定 data for write (hex):00 00 , 0x02 bytes >w ←書き込む (アドレス 0000h が書かれた) >c 8 ←読み出しバイト数を設定 read counter = 0x08 bytes >r ←読み出し FF FF FF FF FF FF FF FF ←0000h 番地から 8 バイトの内容 >s 00 00 12 34 56 ←書き込むバイト列を設定 data for write (hex):00 00 12 34 56 , 0x05 bytes >w ←書き込む (アドレス 0000h から 12h, 34h, 56h が書き込まれたはず)</pre>	<pre>>s 00 00 data for write (hex):00 00 , 0x02 bytes >w ←アドレス設定 0000h を書き込む >r ←読み出し 12 34 56 FF FF FF FF FF FF ←書き込まれていた ></pre>
---	--

I²C デバイスが自由に使えるとなると、通販や秋葉原で売られている各種デバイスにも興味がわいてきます。筆者の手元では I²C 接続できる温度センサや LCD モジュールなどを使用しています。これから、色々なデバイスを利用してみたいと思います。

えすぴー・ラボラトリー

twitter: @espilab

参考資料・関連リンク

(1) MSP-EXP430G2 LaunchPad Experimenter Board User's Guide (SLAU318D)

(2) 筆者サイト <http://espilab.ddo.jp/pool/download.html>