＜編集日：R2 2017.10.14＞

I2C-OS / Read&Write Controller for I2C device
I2C-OS 基板 製作マニュアル

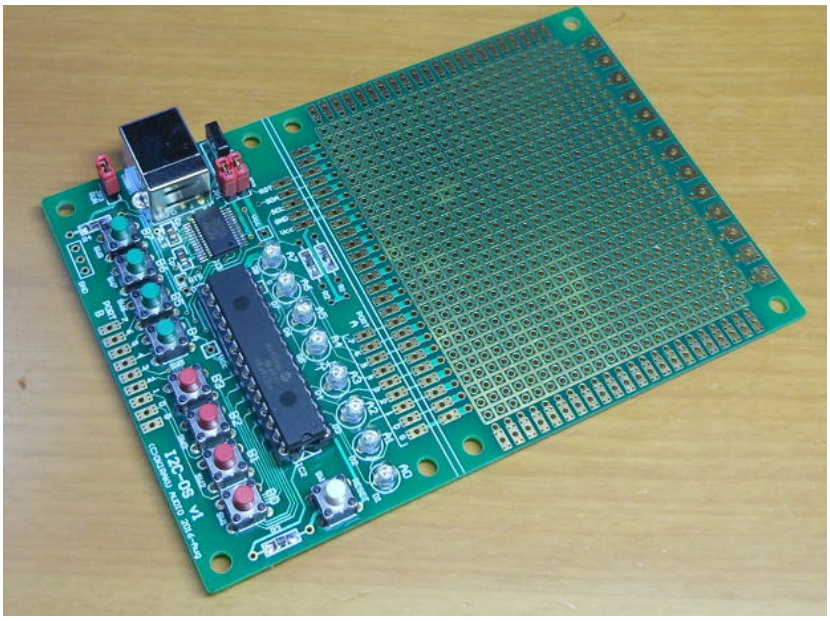
＜注意＞

本キットをつかって生じた感電、火災等の一切のトラブルについては、当方は責任を負いませんのでご了承ください。また、基板、回路図、マニュアル等の著作権は放棄していませんので、その一部あるいは全体を無断で第３者に対して使用することはできません。

１．はじめに

本基板は PC から USB を介してターミナルソフトにて、I2C でソフト制御のデバイスを動作させるための基板です。I2C の制御線のほかにもリセット用の信号もあり DAC などの I2C デバイスを容易に動作させることが可能と思います。さらに、制御部だけでなく部品を実装するためのユニバーサル面も併設していますので簡単な回路をこの基板上で組み上げることにも適しています。

さらに、本基板は PIC マイクロコントローラへの制御コマンドの書き込み基板でもあります。すなわち、本基板上で制御コマンドを書き込んだ PIC を別の基板で動作させることができます。I2C 制御のデバイスを動作させるための PIC の作成に便利かと思います。



完成例

２．仕様(Specification)

表　主な仕様(Specification)

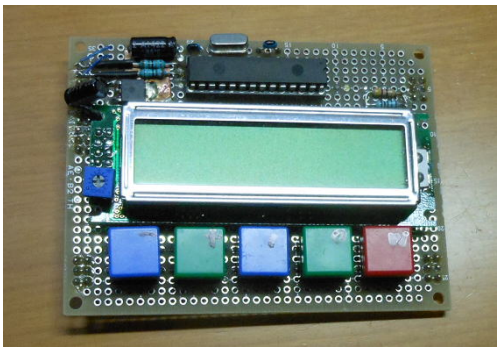
機能 Function	I2C 制御用のデバイスの評価基板、および I2C 制御コマンドの PIC への書き込み機能
仕様＆特徴 Spec. and features.	・ USB を介し PC より I2C デバイスを制御。 ・ PIC の IO として入力スイッチ８個と出力 LED を８個搭載。 ・ 制御線として I2C（SDA,SCL）のほか RESET 機能もあり。 ・ PIC の EEPROM を利用してコマンドプログラム機能有り。
必要電源 POWER	・USB より供給（５V)あるいは、外部より任意の電圧を供給可能。
基板仕様	FR4、厚さ 1.6mm、銅箔厚 70μm、金メッキ、サイズは巻末

OKIRAKU AUDIO 2017 All copyrights reserved

３．開発経緯（開発コンセプト）

I2C はフィリップス社で開発されたシリアルバスであり、抵抗でプルアップされた双方向のオープンコレクタ信号線が 2 本のシンプルな構成にもかかわらず、最大で 127 個のデバイスとの通信ができる優れた規格です。

しかしながら、その制御はやや複雑であることから通常は I2C の制御ライブラリを有する高級言語（C 言語など）でプログラムを組んだ上で使用する必要があります。そのため都度プログラムの作成が必要となりかなり面倒でした。うまく動かなくてもプログラムにバグがあるのか、それとも I2C デバイスの使い方に問題があるのかの切り分けができません。そのため、I2S デバイスを制御するための専用の基板をユニバーサルで作成したことがあります。



ユニバーサルで組んだ I2S 制御基板

これはこれで便利に使っていますが、キースイッチの操作回数も多いことと、もともと PC が近くに使える環境でもあることから、直接 PC のキーボードで入力するようにした方が便利であることから、今回の開発に至っています。PC との接続は簡単にすませるため USB 接続として、基板側に FT232 を用いて USB からシリアル信号に変換して基板上の制御用 PIC と通信を行っています。

さらに評価対象の I2C デバイスとの制御を確認したのちに、そのまま他の基板に PIC が移植できれば便利です。そこでデバイスの確認に用いた制御コマンドなどを PIC 内の EEPROM に格納しておき、PIC にはその内容を解釈して実行できるプログラム（OS）も搭載しておけば、あらたにプログラムを作成する必要もありません。今回はこういった機能を実現させるために基板ならびに制御用の PIC ソフトを開発しました。

機能をまとめると、この基板および PIC の機能を大きくわけると３つあります。

表　本基板および PIC の機能

機能	内容	動作条件
I2C デバイス評価 （基板＋PIC）	PC と本基板を USB 接続して、I2C デバイスに直接書き込み読み出しを行います。本基板からは I2C 制御線（SDA, SCL）のほかに RESET 信号も出ています。簡単な回路であれば基板上のユニバーサル上に組むことも可能です。	JP3 開放
I2C デバイス動作コマンド書き込み （基板＋PIC）	I2C デバイスの制御コマンド（プログラム）を PIC 内部の EEPROM に書き込みます。制御コマンドは PC より伝送します。プログラムは簡単なコマンドラインの集合体になっています。	JP3 開放
PIC 単体実行 （基板＋PIC あるいは PIC 単体）	I2C デバイスの制御コマンド（プログラム）を書き込んだ PIC を単体で動作させることが可能です。そのため本基板は複数の PIC 書き込みのプラットフォームとしても使用可能です。	JP3 接続 あるいは PIC の Pin13(C2)を直接 GND に接続

OKIRAKU AUDIO 2017 All copyrights reserved

４．端子機能

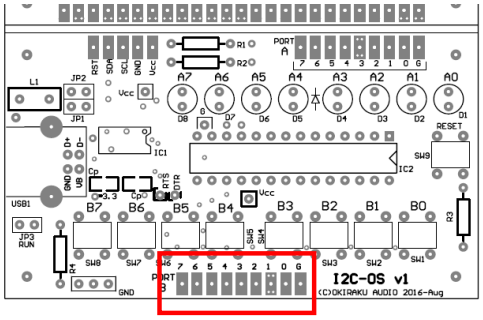
(1)基板端子機能

(i)PORT-B

PIC の PORT-B は外部入力に使用します。主な用途はスイッチを接続することを想定しており（そのため PORT-B は PIC 内部でプルアップ）、制御プログラムを使用してスイッチ操作に応じて I2C デバイスとの通信や PIC の PORT-A の制御をすることができます。

表　基板端子機能（PORT-B）

	No	内容
PORT-B	0	PIC の B0 に接続
	1	PIC の B1 に接続
	2	PIC の B2 に接続
	3	PIC の B3 に接続
	4	PIC の B4 に接続
	5	PIC の B5 に接続
	6	PIC の B6 に接続
	7	PIC の B7 に接続
	G	GND 接続

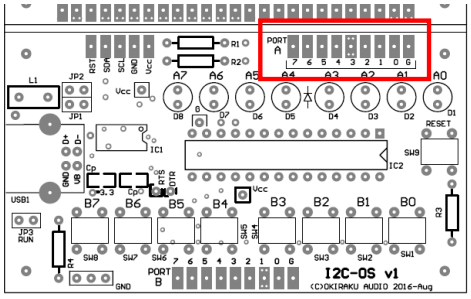


(ii)PORT-A

PIC の PORT-A は外部出力に使用します。主な用途は LED を接続することを想定しており、制御プログラムを使用してスイッチ操作に応じてその状況を表示することができます。LED 以外にもリレー制御に用いても便利でしょう。

表　基板端子機能(PORT-A)

	No	内容
PORT-A	0	PIC の A0 に接続
	1	PIC の A1 に接続
	2	PIC の A2 に接続
	3	PIC の A3 に接続
	4	PIC の A4 に接続
	5	PIC の A5 に接続
	6	PIC の A6 に接続
	7	PIC の A7 に接続
	G	GND 接続

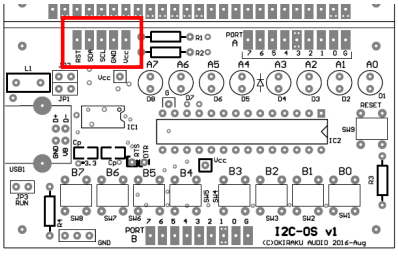


(iii)I2C 制御線

I2C デバイスと接続するメインの信号線になります。

表　基板端子機能（I2C 制御線）

No	内容
RST	RST 信号出力（PIC の C5 に接続）
SCL	I2C 信号の SCL（PIC の C3 に接続）
SDA	I2C 信号の SDA（PIC の C4 に接続）
GND	電源 GND
Vcc	電源 Vcc



(2)ジャンパー設定

本基板には JP1～JP3 の３つのジャンパーがあります。

(i)JP1,JP2

JP1,JP2 は基板内部の電源に関する設定です。下記の組み合わせで使用します。

表　JP1.2 の設定

JP2	JP1	基板上の Vcc	内容
短絡	短絡	５V	USB より PIC に電源を供給します。そのため外部電源 Vcc は 5V となります。
開放	短絡	外部 VCC	PIC に独立して電源供給して使用します。大電流(>500mA)が必要な場合に設定します。IC1（FT232）の動作電圧が 5V となるため、外部電圧（Vcc）も 5V を供給します。
短絡	開放	外部 VCC	PIC に独立して電源供給して使用します。IC1（FT232）の動作電圧は外部電圧（Vcc）となります。大電流(>500mA)が必要な場合や PIC を任意の電圧で使用する場合に設定します。外部電圧 Vcc は 3～5V で可変することができます。たとえば Vcc に 3.3V を供給することで、PIC は 3.3V 動作となり、I2C デバイスが 3.3V 動作の時などに整合させることができます。
開放	開放	外部 VCC	IC1(FT232)を実装しない場合の設定です。PIC と周辺スイッチ、LED だけをつかう場合を想定しています。電源電圧は 3.3V～５V の範囲で使用することができます（PIC の電源電圧範囲に依存)。

(iii)JP3 (RUN)

JP3 は PIC コントローラの動作モードを設定します。JP3 は PIC の Pin13(C2)に接続されており、このピンの状態により PIC の動作モードを変更しています。

表　JP3　(RUN)

JP2	内容
開放 (OPEN)	PC との通信により I2C 制御線のコントロールならびに PIC への制御コマンドを書き込みします。
短絡 (SHORT)	PIC を独立的に動作させます。EEPROM に書かれた制御コマンドを実行します。PC は必要ではありません。

OKIRAKU AUDIO 2017 All copyrights reserved

４．部品表例

部品表例を示します。

表　部品表例

区分	部品No	規格	仕様	個数	備考
抵抗	R1,2	炭素皮膜 1/4W	1kΩ	2	2012 チップ抵抗も可 I2C プルアップ用。
	R3,4	炭素皮膜 1/4W	47kΩ	2	2012 チップ抵抗も可 プルアップ用。
	Rb	ﾁｯﾌﾟ抵抗	1kΩ	8	2012,1608 サイズ LED の電流制限抵抗。
ｺﾝﾃﾞﾝｻ	Cp	ﾁｯﾌﾟｺﾝﾃﾞﾝｻ	0.1uF	5	2012,1608 サイズ パスコン。
	Cb	ﾁｯﾌﾟｺﾝﾃﾞﾝｻ	1uF	2	3216 サイズ パスコン。
ダイオード	D1-8	LED	φ3 赤色	8	3.3V 使用時は青、白色 LED は不可 (Vf が 3V を超えるため)。
インダクタ	L1	インダクター	10～100uH	1	適当でよい。ジャンパーでも可。
IC	IC1	USB-RS232 変換	FT232RL	1	SSOP28 秋月電子で購入可
	IC2	PIC	28PIN タイプ	1	SDIP-28 プログラム済 PIC18F26K22 ベース
スイッチ	SW1-9	タクトスイッチ		9	
コネクタ	USB1	USB-B コネクタ	USB-B	1	標準サイズ 秋月電子で購入可
基板			I2C-OS	1	

※ハッチング部がキットのオプション。

OKIRAKU AUDIO 2017 All copyrights reserved

５．接続方法

以下にいくつかの接続例を示します。

(1) 標準接続

5V 動作の I2C デバイスと接続して評価（および PIC への制御コマンド書き込み）を行う場合の標準的な接続法です。あるいは I2C デバイスへの接続はなく、単に PIC の制御コマンドの書き込みだけ行う場合の接続になります。

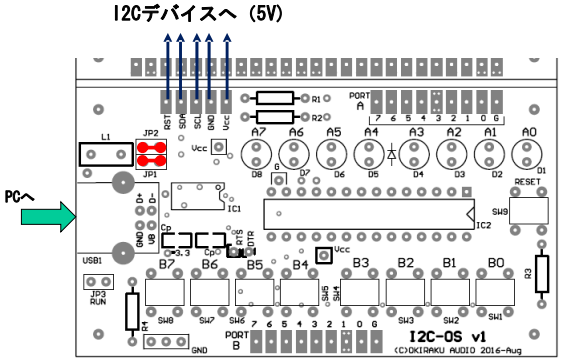


図　標準的な接続

（電源は USB より供給。外部 I2C デバイスは５V。JP1,2 を接続。JP3 は開放）

(2)外部給電してデバイス評価（および PIC への制御コマンド書き込み）を行う場合

たとえば､3.3V 動作 I2C デバイスを評価する場合､あるいは I2C デバイスを含む回路が 5V だが大電流(500mA 以上)を必要とする場合の接続例です。USB の 5V 給電では対応できない場合の接続方法です。

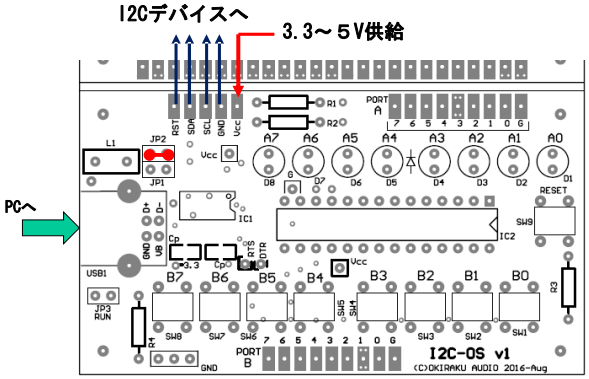


図　外部給電で使用する場合

（JP2 を接続。JP1,3 は開放）

(3)内部給電で 3.3V 動作 I2C デバイスの評価（および PIC への制御コマンド書き込み）を行う場合

FT232 の LDO（電圧レギュレータ）出力を外部に供給します。したがってデバイスの消費電流は 50mA 以下に抑える必要があります（PIC ならびに LED, プルアップ抵抗での消費もあるので、実質は 20mA 以下）。あまり推奨できる方法ではありません。消費電流がごくわずかな I2C デバイスの評価に適用可能です。

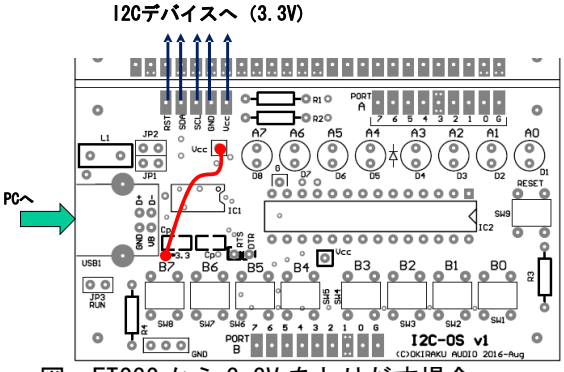


図　FT232 から 3.3V をとりだす場合

（JP1,2,3 は開放．基板上の「3.3」から「Vcc」へジャンパ接続）

OKIRAKU AUDIO 2017 All copyrights reserved

(4)制御コマンド書き込み済 PIC を単体で動作（電源は USB から供給）

I2C デバイスが 5V の場合で、PIC を単体で動かす場合の接続です。USB は PC からの 5V 電源供給のみに使用します。

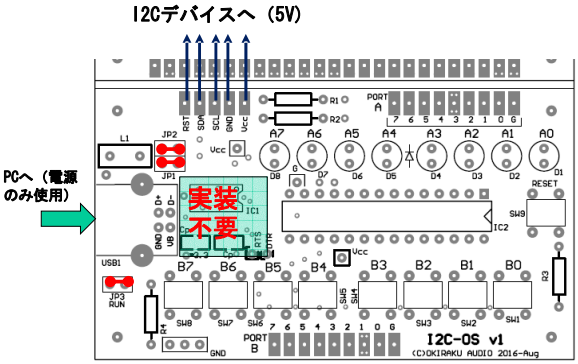


図　USB 給電で PIC 単体で動作させる場合
（JP1, 2, 3 は接続。IC1(FT232)は実装不要）

(5)制御コマンド書き込み済 PIC を単体で動作（電源は外部から供給）

本基板を独立させて使用する場合の接続になります。外部 Vcc より PIC に 3.3～5V 電源を供給します。

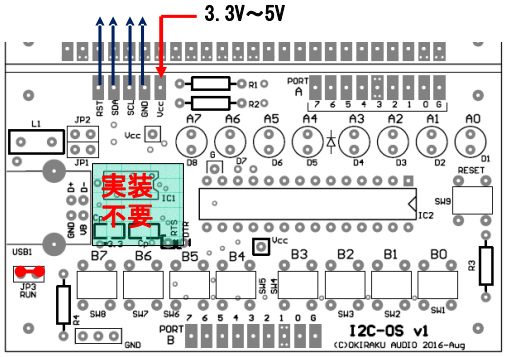


図　外部給電で PIC 単体で動作させる場合
（JP3 は接続。USB コネクタや FT232(IC1)は実装不要）

(6)PIC を他の基板で使用

制御コマンドプログラム済 PIC を他の基板で用いる場合の接続です。必要な接続は PIC の C2(Pin13)を GND、MCLR(Pin1)を Vcc に接続しておきます。また SDA, SCL へのプルアップ抵抗（1～1.5kΩ）も必要です。

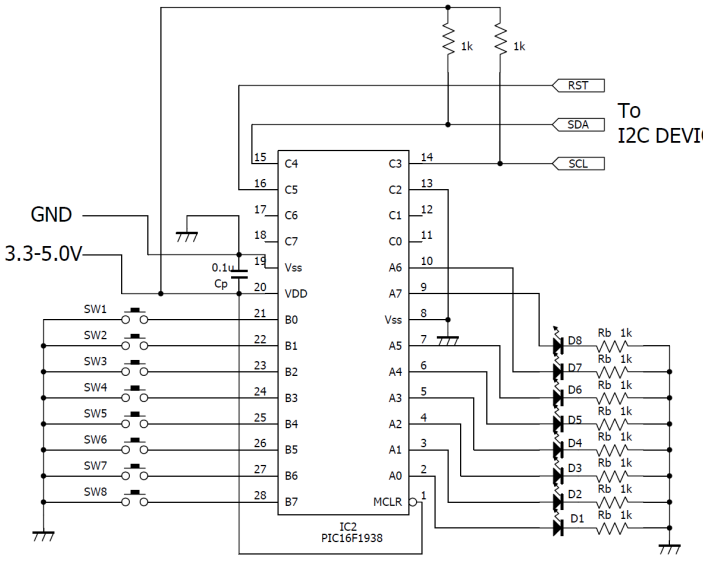
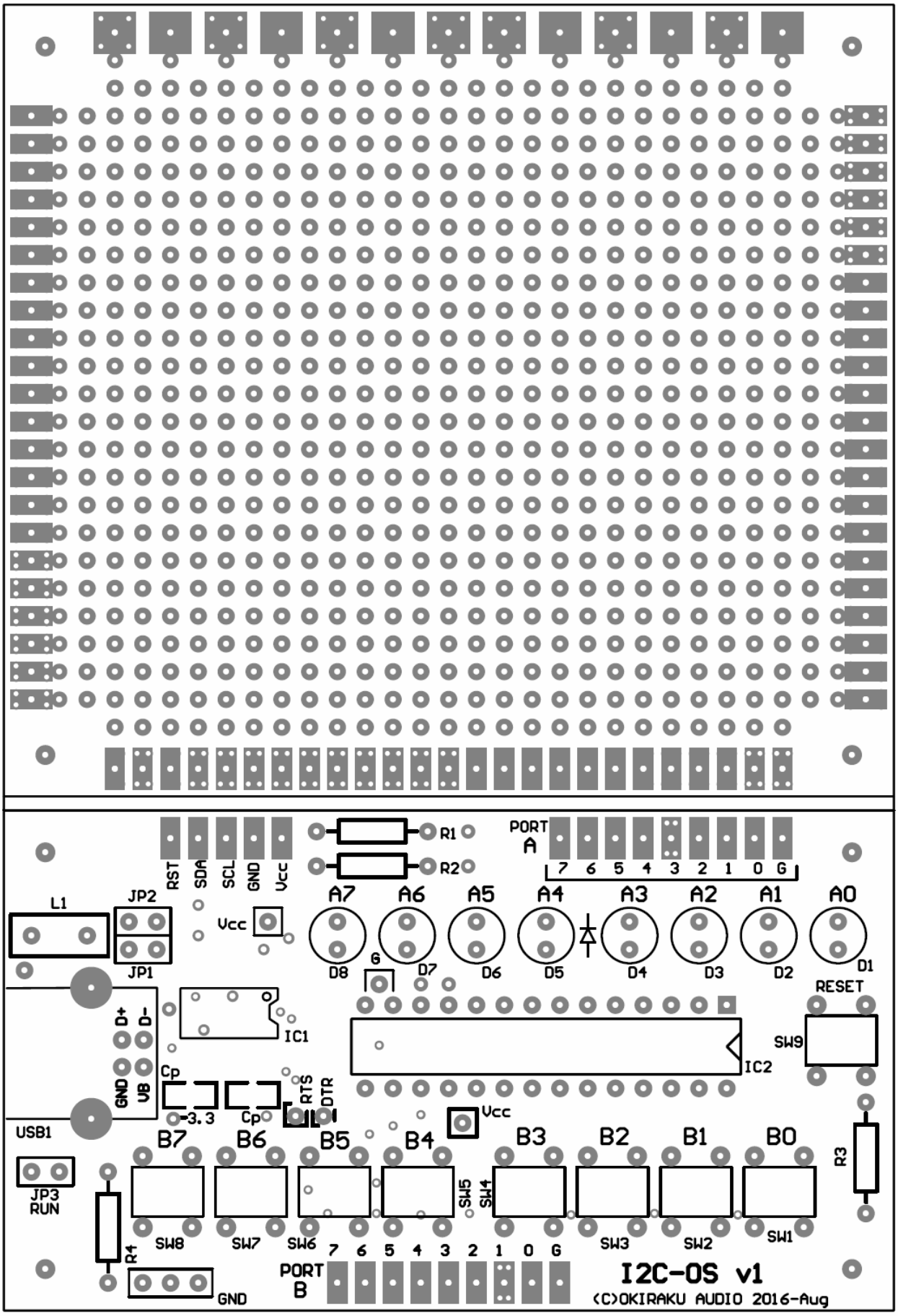


図　PIC を他の基板で使用する場合。スイッチや LED は必要な場合実装。

OKIRAKU AUDIO 2017 All copyrights reserved

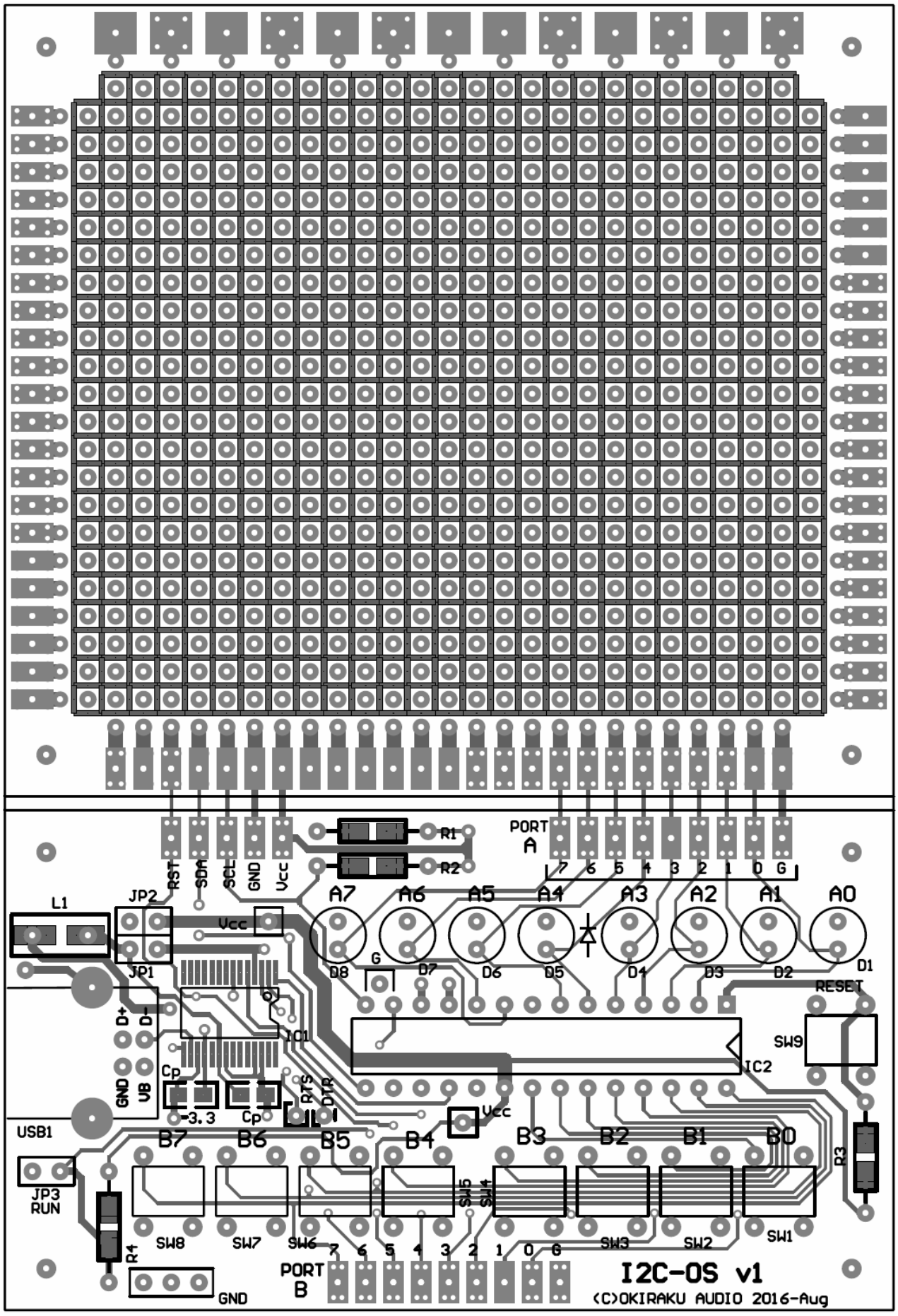
６．基板パターン

(1)基板シルク



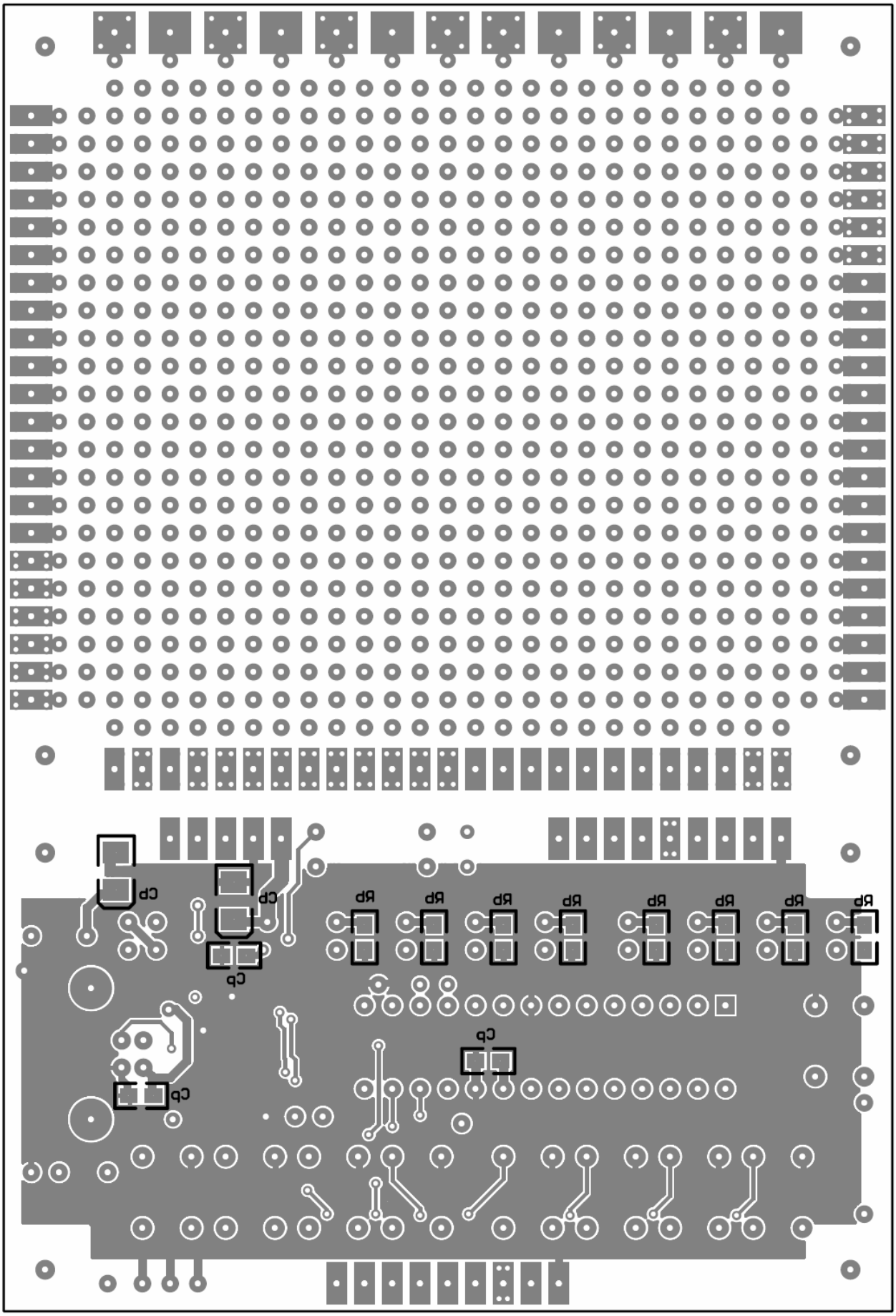
OKIRAKU AUDIO 2017 All copyrights reserved

(2) 基板部品面パターン

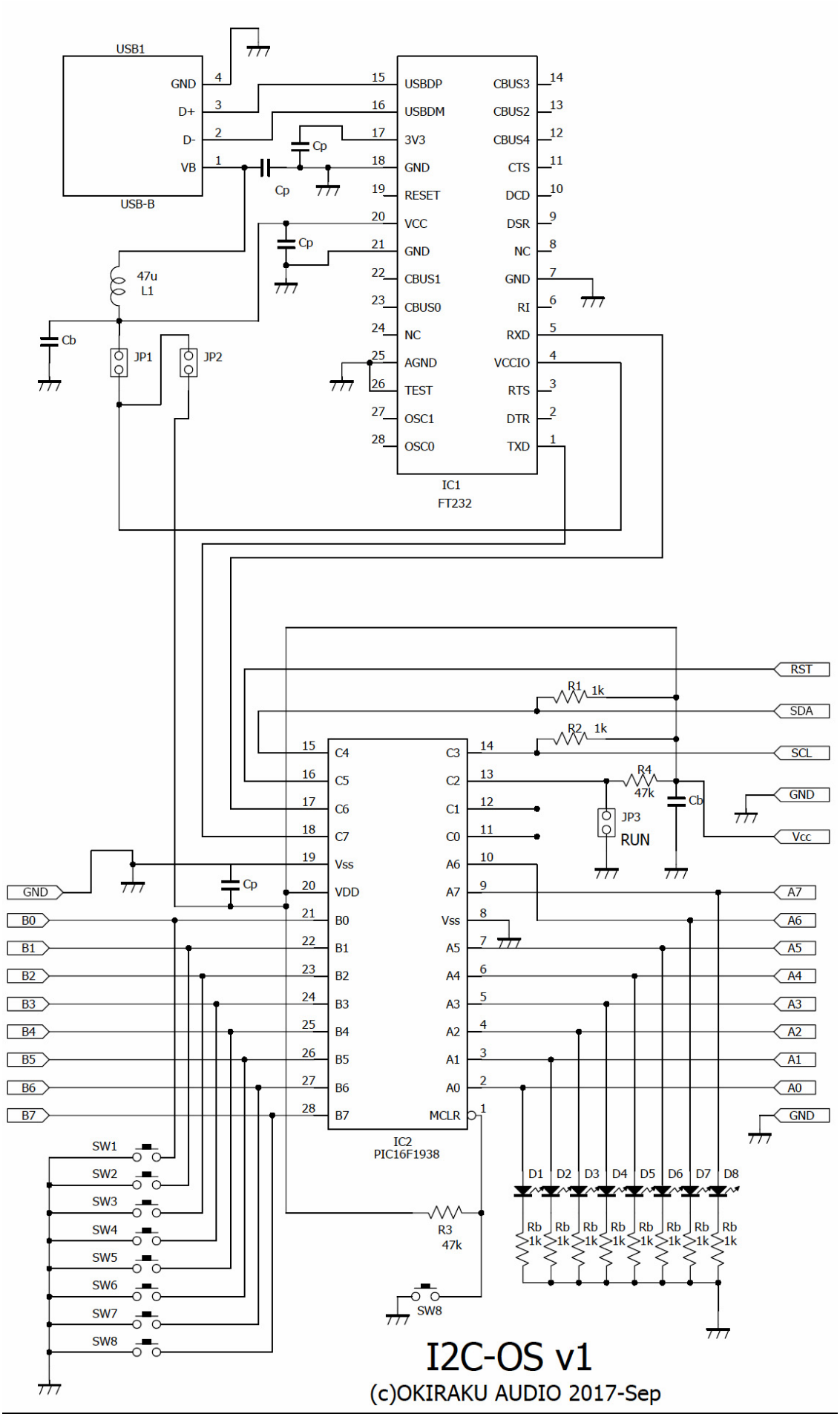


OKIRAKU AUDIO 2017 All copyrights reserved

(3) 半田面パターン



7．回路図



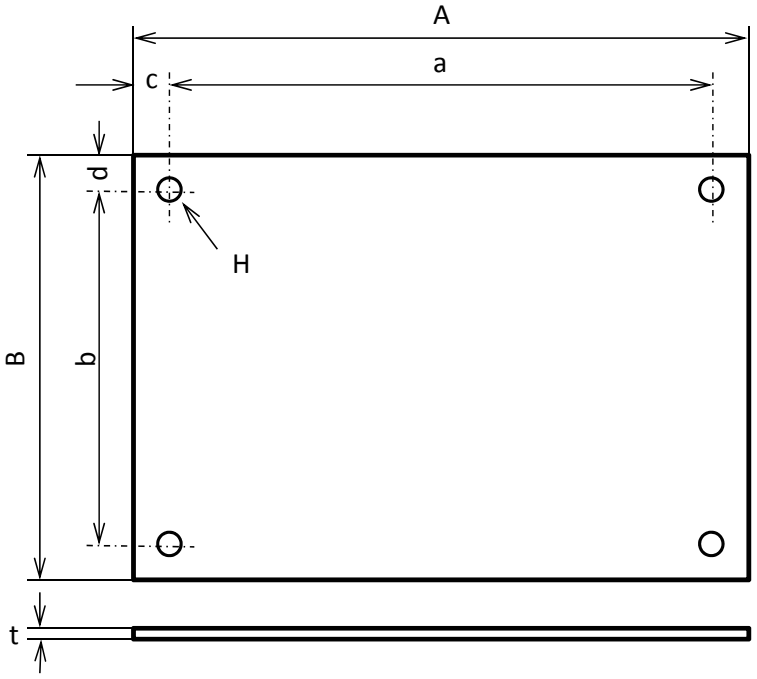
OKIRAKU AUDIO 2017 All copyrights reserved

８．基板寸法

本基板サイズは”STD“になります。

表　寸法 単位 mm/(mil)　※1mil=25.4/1000mm

	name	A	B	t	H	a	b	c,d
	STD-S	119.4 (4700)	43.2 (1700)	1.6	3.5 (138)	111.8 (4400)	35.6 (1400)	3.8 (150)
✔	STD	119.4 (4700)	81.3 (3200)	1.6	3.5 (138)	111.8 (4400)	73.7 (2900)	3.8 (150)
	STD-H	81.3 (3200)	59.7 (2350)	1.6	3.5 (138)	73.7 (2900)	52.1 (2050)	3.8 (150)
	WIDE	144.8 (5700)	101.6 (4000)	1.6	3.5 (138)	137.2 (5400)	94.0 (3700)	3.8 (150)
	None							



OKIRAKU AUDIO 2017 All copyrights reserved

９．【重要】基板の修正

v1 基板では一か所要修正箇所があります（修正しない場合は A0 の LED が点灯できません)。下記を参考に修正してください。A0 の LED（D1）を使用しない場合は修正の必要はありません。

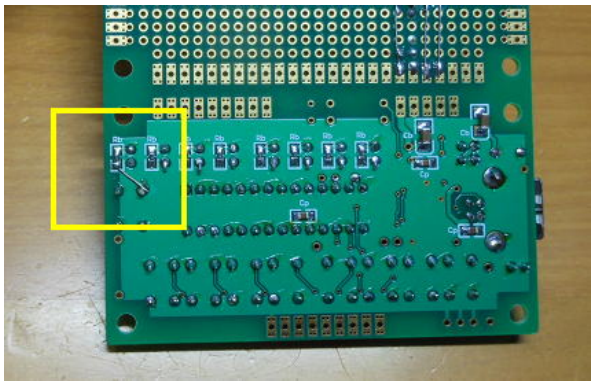


図　修正位置

図　修正用ジャンパー線

OKIRAKU AUDIO 2017 All copyrights reserved

１０．使い方（ソフトの使用方法）

以下では、WINDOS がインストールされている PC との接続を想定して使用方法を記します。

(1)準備

(i)PC との接続

USB ケーブルを用いて本基板と PC と接続します。基板上の FT232RL は自動認識されて必要なソフトが自動でインストールされます（認識されない場合はメーカサイトからドライバーをダウンロードしてください。いままで XP,Win8,Win10 ではすべて自動認識されました）。

メーカサイト：ｈｔｔｐ：／／ｗｗｗ．ｆｔｄｉｃｈｉｐ．ｃｏｍ／

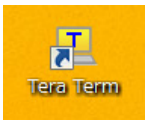
(ii)ターミナルソフトのインストール

用いるターミナルソフトは種類を選びません。すでにインストールされているターミナルソフトがあればそれを用いればいいでしょう。

通信条件は **9600baud，８ビット、ノンパリティ、１スタートビット、１ストップビット**です。

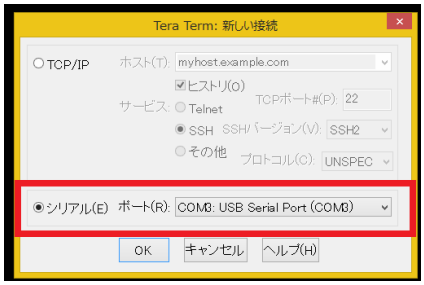
以下では TERA TERM を用いた場合のインストールと設定例について示します。TERA TERM は WEB で“TERATERM”で検索すれば簡単に見つけられます。https://ja.osdn.net/projects/ttssh2/ からダウンロードします。このマニュアルを執筆していたときの最新版は teraterm-4.96.exe でした。

ダウンロード＆実行するとディスクトップに以下のアイコンが作成されます。



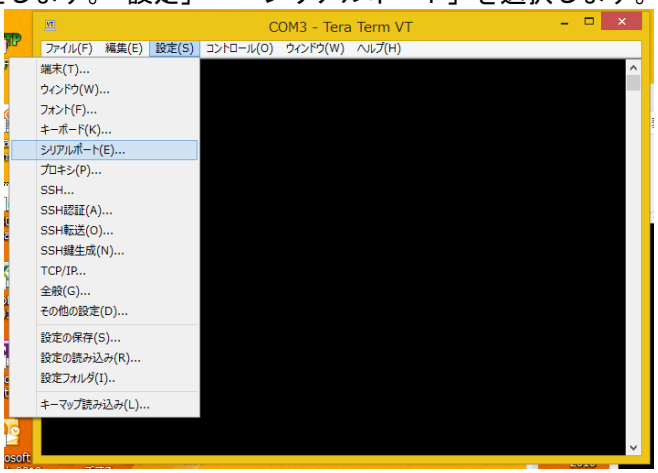
TeraTerm アイコン

このアイコンを最初に実行すると「新しい接続」の設定が現れます。TCP/IP が既定値となっていますが、ここではシリアルポートを選択します。



「新しい接続」の設定。シリアルを選択。

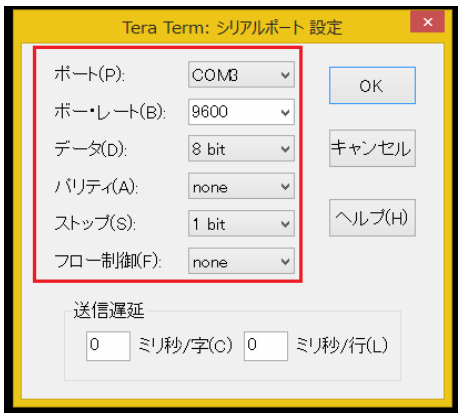
そのあと、通信条件を設定します。「設定」→「シリアルポート」を選択します。



通信条件の確認を選択

通信条件が 9600baud、8Bit、パリティなし、１ストップビット、フロー制御なしになっていることを確認します。なお接続ポートは COM3 になっていますが、これはお使いの PC によって変わります。

OKIRAKU AUDIO 2017 All copyrights reserved



通信条件を確認
(9600baud、8Bit、Non-Parity、1StopBit、None-Flow control)

設定ができれば、接続環境は完了です。リターンキーを押せばコマンドプロンプトの＊が表示されるはずです。また基板上のリセットボタン（RESET）を押せば以下の表示となるはずです（表示のバージョンはリリース版と異なる場合があります）。

最後に「設定（S)」→「設定の保存（S)」を行っておけば、以降はこの確認は不要になります。

オープニングメッセージとコマンドプロンプト表示

簡単なコマンドとして”A”あるいは”a”を入力して、そのあとに２桁の 16 進数を入力すれば、その値に応じた基板上の LED が表示されます。A コマンドは PORT-A への出力コマンドです。

A コマンドで”44”を入力すると、LED の A6,A2 が点灯します（16 進で 0x44 に相当する LED が点灯)。

なお PIC に搭載されているコマンド一覧を表示せるには HELP 表示である”？”を押します。

例）A コマンドの入力例。LED の A6,A2 が点灯します。

次項では各コマンドについて説明します。

（２）直接コマンド入力方法

コマンドプロンプト＊のあとにコマンドとパラメータを入力します。コマンドは英文字１文字です。パラメータがある場合は 16 進数２桁で入力します。パラメータの確定はリターンキーで行い、修正は BS にて行います。パラメータを入力せず、リターンキーのみを押すとそのコマンドはキャンセルされます。

次表は直接入力コマンドの説明になります。

表　直接入力コマンド一覧

コマンド＋パラメータ []は 16 進２桁の定数	説明
W [ADRS][REG][DATA]	※Write I2C device I2C デバイスに書き込みを行います。 [ADRS]：I2C デバイスのアドレス [REG]：レジスタ（第１パラメータ） [DATA]：データ（第２パラメータ）
R [ADRS][REG]	※Read I2C device I2C デバイスから読み出しみを行います。パラメータを入力するとデータが表示されます。 [ADRS]：I2C デバイスのアドレス [REG]：レジスタ（第１パラメータ）
+ ([DATA])	※Increment [REG] at previous W/R W/R コマンドをレジスタ値をインクリメント(+1)して実行します。I2C アドレスとレジスタの再入力が不要です。W コマンド場合のみデータを入力します。 ([DATA])：データ（第２パラメータ）
- ([DATA])	※Decrement [REG] at previous W/R W/R コマンドをレジスタ値をデクリメント(-1)して実行します。I2C アドレスとレジスタの再入力が不要です。 W コマンド場合のみデータを入力します。 ([DATA])：データ（第２パラメータ）
. ([DATA])	※Unchanged [REG] at previous W/R W/R コマンドをレジスタ値の変更なしで再実行します。I2C アドレスとレジスタの再入力が不要です。W コマンド場合のみデータを入力します。 ([DATA])：データ（第２パラメータ）
A [DATA]	※A-PORT of PIC direct set PIC の PORT-A（出力専用）にデータを出力します。 [DATA])：PORT-A への出力データ
B	※B-PORT read PIC の PORT-B（入力専用）の状態を表示します。
K	※negative level RST seq.(H->L->H) リセット出力を“L”レベルにしたのち、”H”レベルに戻します。“L”の保持時間は 1ms です。一般のリセットはこのシーケンスになります。
J	※positive level RST seq.(L->H->L) リセット出力を“H”レベルにしたのち、”L”レベルに戻します。“H”の保持時間は 1ms です。
?	※HELP コマンド一覧が表示されます。さらに？を押すとプログラムで使用する制御コマンドが引き続き表示されます。

OKIRAKU AUDIO 2017 All copyrights reserved

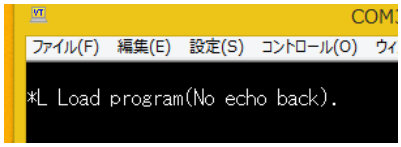
制御コマンドを PIC にプログラム時の関連するコマンドは次表のとおりです。

表　制御コマンドプログラム関連コマンド一覧

コマンド	内容
L	※Load program (no echo back) 制御コマンドプログラムを入力あるいはそのファイルをターミナルソフトから転送します(*1)。入力された内容のエコーバックはありません。 正常なプログラムのロードが終わるとエラーなしが表示され、自動的に EEPROM に制御コマンドが書き込まれます。 　プログラムにエラーがある場合はエラー表示を行い停止します。なお、エラーが発生するとプログラムはエラー以降の余分なコマンドを解釈しきれず、ハングする可能性もあるので、リセットにて復帰してください。
P	※display contents of Program 書き込まれたプログラムの内容を表示します。
X	※eXecute program 書き込まれたプログラムを実行します。JP3 を短絡させて PIC を起動した場合と同じ動作になります。
Z	※execute program with monitoring 書き込まれたプログラムを実行します。PC 側に実行内容を逐次表示します。プログラムの実行確認に使用します。
?	HELP コマンド一覧が表示されますが、さらに？を押すとプログラムで使用する制御コマンドが引き続き表示されます。

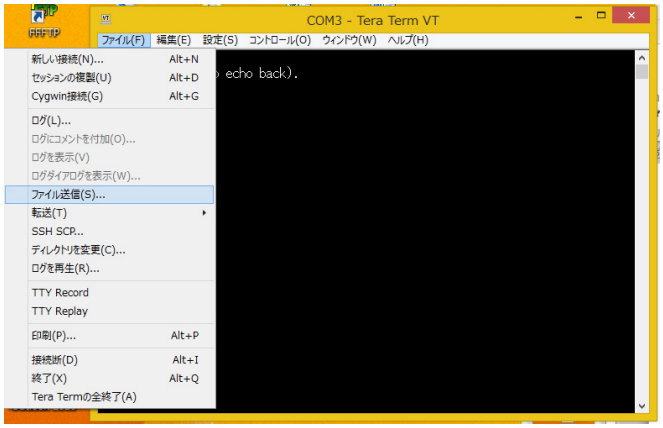
(*1) プログラムファイルの転送方法

　コマンド L を入力すると下記の表示となります。この状態になれば、ファイル転送等でプログラムを転送します。

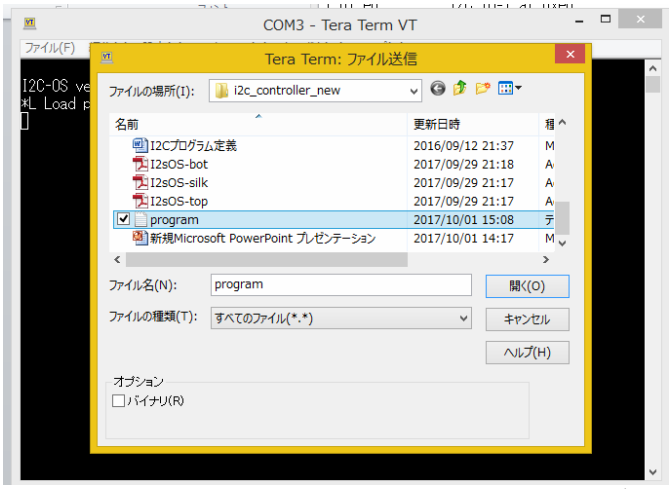


制御コマンドプログラムの転送待ちの状態

TeraTerm の場合は「ファイル（F)」→「ファイル送信（S)」にて送信するファイルを選択して転送します。

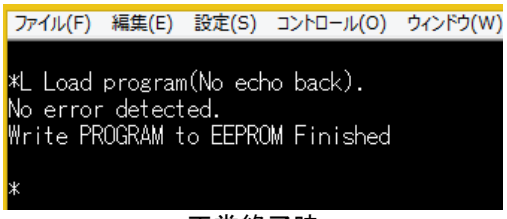


「ファイル（F)」→「ファイル送信（S)」にてファイル送信を選択



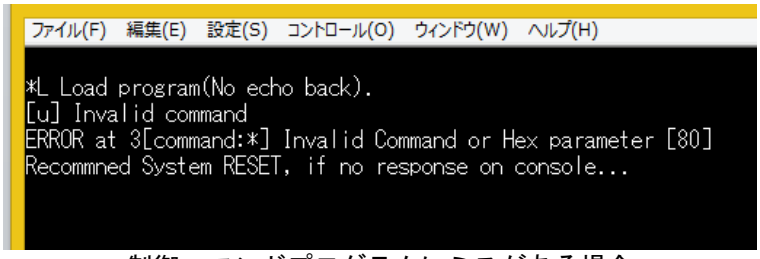
転送するファイルを選択して「開く（O)」をクリックすれば転送開始。

プログラム内容が正常で、転送が終了すれば下記のようにエラーなし、EEPROM 書き込み終了が表示されます。



正常終了時

プログラム内容にエラーがある場合はエラー表示を行います。このとき、システムがハングする場合がありますので RESET により復帰させます。



制御コマンドプログラムにミスがある場合

<u>（３）制御コマンドによるプログラムの作成方法</u>

プログラムはテキストエディタ等を使用して作成します。

構造は１行１実行となっており、１行に複数の実行を記述することはできません。またテキスト行の先頭文字は制御コマンドでなければなりません。制御コマンドとパラメータ間はスペースで区切ります。

【基本文法】

制御コマンド（１文字）　パラメータ１　パラメータ２・・・　（コメント）

パラメータはすべて２桁の 16 進数です（“f1”,”4e”など)。不要な文字や文字数が異なるとエラーが発生します。

次表に制御コマンドを示します。

なお記述できる制御プログラムの容量は最大で 250 行です。一般的な使用であればこれで十分です。PIC 内部での処理フロ―は PORT-B の状態変化（基板上では B0～B7 ボタンの ON/OFF）があれば、関連するコマンド（表のハッチン部）を制御コマンドプログラムの記述順に合致するものを探して実行しています。

OKIRAKU AUDIO 2017 All copyrights reserved

表　制御コマンド一覧（プログラム） ※制御コマンドは大文字、小文字を区別しています。

コマンド＋パラメータ []は 16 進２桁の定数	制御コマンド説明
N	※Reset with LOW level(NORMAL RESET) リセットシーケンスにおけるリセットレベルを “L”に設定します。通常はこの場合が多いと思います。指定しない場合は自動的にこの設定になります。
n	※Reset with HIGH level(Reverse RESET) リセットシーケンスにおけるリセットレベルを “H”に設定します。
s[PORTA]	※Initial A-PORT before RESET PIC の PORT-A の初期状態を設定します。 例）起動直後は PORT-A は全て L レベルにする。 s 00
J[MASK]	PORT-B 入力の一部をジャンパーピン扱いとします。これを設定することにより、ジャンパーピンの設定に対応した制御コマンド(I2C データ）を送信することができます。 例）B7,B6 をジャンパーピンとして設定する。 J C0 例）B7,6,5,4 をジャンパーピンとして設定する。 J F0
I[ID][ADRS]	※SET I2C-DEVICE[ID] address to [ADRS] 使用する I2C デバイスの ID とアドレスを定義します。本 PIC で使用できる I2C デバイスは最大で４個です。ID は 00,01,02,03 から選択します。 例）ID:00 にアドレス 4E、ID:02 に 8C を設定 I 00 4e I 02 8c
W[TIME]	※Wake up time for power on [TIME]*100ms PIC へ電源投入後の起動時間を設定します。この時間を経過したのちに制御コマンドプログラムの実行を開始します。設定 TIME に 100mS を乗じた値が起動開始時間になります。 例)1 秒後に起動 W 0A
w[TIME]	※wait-time after reset [TIME]*10ms PIC が起動したのち、リセットコマンドを実行します。リセット後の制御コマンドプログラムの実行までの待機時間を指定します。設定 TIME に 10mS を乗じた値が起動開始時間になります（要はリセット後の I2C デバイスの起動を待つ意味の wait です)。 例)リセット後 10mS 後に制御コマンドプログラム起動 w 01
i[ID][REG][DATA][DELAY2]	※I2C DATA SEND after RESET リセット送出後の I2C デバイスにデータを送出します。これはリセット後に事前に I2C デバイスに各種パラメータを設定する場合に使用します。実行後に DELAY2（×10mS）で示された時間のウエイトを設けます。 例） I 00 00 0F 00 I 00 01 10 00 I 01 01 03 00
B[BT][ID][REG][DATA][DELAY2]	※B-PORT Bit[BT] ON(LOW LEVEL)　I2C DATA out PIC の PORT-B の指定されたボタン[BT,Bit]が押された場合（レベル LOW に変化)に I2C デバイスにデータを送出します。実行後に DELAY2(×10mS)で示された時間のウエイトを設けます。 例）B:00 ボタンが押された場合、デバイス ID:01 のアドレス 10 にデータ C0 を送出する。実行後のディレイはなし。 B 00 01 10 C0 00

表　制御コマンド一覧（プログラム）つづき

コマンド＋パラメータ []は 16 進２桁の定数	制御コマンド説明
b[BT][ID][REG][DATA][DELAY2]	※B-PORT Bit[BT] OFF(HIGH LEVEL) I2C DATA out PIC の PORT-B の指定されたボタン[BT,Bit]が離された場合（レベル HIGH に変化）に I2C デバイスにデータを送出します。 例）B:00 ボタンが押された場合、デバイス ID:01 のアドレス 10 にデータ 30 を送出します。実行後に DELAY2（×10mS）で示された時間のウエイトを設けます。実行後のディレイはなし。 b 00 01 10 30 00
P[BT][PORTA][DELAY2]	※B-PORT Bit[BT] ON(LOW LEVEL) A-PORT set & DELAY PIC の PORT-B の指定されたボタン[BT,Bit]が押された場合（レベル LOW に変化）に PORT-A にデータを書き込みます。その後 DELAY2（×10mS）で示された時間のウエイトを設けます。 例）B:07 ボタンが押された場合、PORT-A に 0F を書き込む。ディレイ時間はなし。 P 07 0F 00
p[BT][PORTA][DELAY2]	※B-PORT Bit[BT] OFF(HIGH LEVEL) A-PORT set & DELAY PIC の PORT-B の指定されたボタン[BT,Bit]が離された場合（レベル HIGH に変化）に PORT-A にデータを書き込みます。その後 DELAY2（×10mS）で示された時間のウエイトを設けます。 例）B:07 ボタンが離された場合、PORT-A に 00 を書き込む。ディレイ時間は 10mS（DELAY2 は 01）。 p 07 00 01
S[PORTA][DELAY2]	※A-PORT set after RESET&DELAY リセット後の PIC の PORT-A の状態を設定します。 ディレイは設定後のウエイト時間（×10mS）を設定します。 例)起動直後に PORT-A の LED を流れ表示させる。表示後のディレイは 40ms とする。 S 01 04　PORT-A Bit 0 ON S 02 04　PORT-A Bit 1 ON S 04 04　PORT-A Bit 2 ON S 08 04　PORT-A Bit 3 ON S 10 04　PORT-A Bit 4 ON S 20 04　PORT-A Bit 5 ON S 40 04　PORT-A Bit 6 ON S 80 04　PORT-A Bit 7 ON S 00 04　ALL LED OFF
M[MASK][ID][REG][DATA]	※I2C DATA SEND for JUMPER SETTING J コマンドで設定したジャンパーピンの設定(H/L)に変化があった場合に I2C デバイスにデータを送出します。事前に J コマンドでどのピン（PORT-B のピン）をジャンパーピンに定義するかを指定しなければいけません。 例）B7,6 が開放(H)されたらデバイス ID:01 のアドレス 10 にデータ C0 を送出する。 M C0 01 10 C0
C	※Commnet このラインはコメント行になります。 例） C COMMENT LINE
E or e	※END OF PROGRAM プログラムコードの終了を示します。なお、E あるいは e のあとには改行を入れないでください。 例） e

（４）プログラム例

以下に制御コマンドのプログラム例を示します。

例１）押したボタンの位置の LED が点灯する（ボタンを離しても点灯は続く）。

```
P 00 01 00          PUSH B0 LED0(PORT-A=01) ON NO-DELAY
P 01 02 00          PUSH B1 LED1(PORT-A=02) ON
P 02 04 00          PUSH B2 LED2(PORT-A=04) ON
P 03 08 00          PUSH B3 LED3(PORT-A=08) ON
P 04 10 00          PUSH B4 LED4(PORT-A=10) ON
P 05 20 00          PUSH B5 LED5(PORT-A=20) ON
P 06 40 00          PUSH B6 LED6(PORT-A=40) ON
P 07 80 00          PUSH B7 LED7(PORT-A=80) ON
E
```

例２）押している間、そのボタンの位置の LED が点灯する（ボタンを離せば消灯する）。

```
P 00 01 00          PUSH B0 LED0(PORT-A=01) ON NO-DELAY
P 01 02 00          PUSH B1 LED1(PORT-A=02) ON
P 02 04 00          PUSH B2 LED2(PORT-A=04) ON
P 03 08 00          PUSH B3 LED3(PORT-A=08) ON
P 04 10 00          PUSH B4 LED4(PORT-A=10) ON
P 05 20 00          PUSH B5 LED5(PORT-A=20) ON
P 06 40 00          PUSH B6 LED6(PORT-A=40) ON
P 07 80 00          PUSH B7 LED7(PORT-A=80) ON
p 00 00 00          OFF B0  LED OFF(PORT-A=00)
p 01 00 00          OFF B1  LED OFF(PORT-A=00)
p 02 00 00          OFF B2  LED OFF(PORT-A=00)
p 03 00 00          OFF B3  LED OFF(PORT-A=00)
p 04 00 00          OFF B4  LED OFF(PORT-A=00)
p 05 00 00          OFF B5  LED OFF(PORT-A=00)
p 06 00 00          OFF B6  LED OFF(PORT-A=00)
p 07 00 00          OFF B7  LED OFF(PORT-A=00)
E
```

【解説】例２では、例１のそれぞれのボタンが押されたら LED を点灯させる P コマンドのみに加えて、ボタンが離されたときの動作としての p コマンドを追加しています。これにより見かけ上、ボタンが押されている間だけ LED が点灯することになります（ただし、ボタン操作は１つづつの場合になります)。PIC の内部処理では、ボタン操作（ON,OFF）時にすべての制御コマンドを逐次チェックし、ボタン操作に合致するコマンドを実行するため P コマンドのあとに p コマンドを記述します。

例３）B0 を押せば左向に、B1 を押せば右方向に LED がフラッシュする。

```
s 00                INITIAL LED OFF
P 00 01 0A          PUSH B0 LED0(PORT-A=01) ON DELAY100mS
P 00 02 0A          PUSH B0 LED1(PORT-A=02) ON
P 00 04 0A          PUSH B0 LED2(PORT-A=04) ON
P 00 08 0A          PUSH B0 LED3(PORT-A=08) ON
P 00 10 0A          PUSH B0 LED4(PORT-A=10) ON
P 00 20 0A          PUSH B0 LED5(PORT-A=20) ON
P 00 40 0A          PUSH B0 LED6(PORT-A=40) ON
P 00 80 0A          PUSH B0 LED7(PORT-A=80) ON
P 01 80 0A          PUSH B1 LED0(PORT-A=01) ON DELAY100mS
P 01 40 0A          PUSH B1 LED1(PORT-A=02) ON
P 01 20 0A          PUSH B1 LED2(PORT-A=04) ON
P 01 10 0A          PUSH B1 LED3(PORT-A=08) ON
P 01 08 0A          PUSH B1 LED4(PORT-A=10) ON
P 01 04 0A          PUSH B1 LED5(PORT-A=20) ON
P 01 02 0A          PUSH B1 LED6(PORT-A=40) ON
P 01 01 0A          PUSH B1 LED7(PORT-A=80) ON
E
```

OKIRAKU AUDIO 2017 All copyrights reserved

例４） CS8416(Digital Audio Receiver)を用いて 8ch の SPDIF 切替

具体的に I2C デバイスを使用するプログラム例です。出力フォーマットは RJ24。選択された SPDIF に該当する LED(PORT-B)を点灯させます。

```
N                NORMAL RESET (L=Active)
W 0A             1sec for warm up
w 05             50mS after RESET
I 00 20          CS8416 ADDRESS=20(ID:0 address 0x20)
i 00 04 80 00    CS8416 RUN and SPDIF0 select (初期設定) (ID:0 REG:4 DATA:80)
i 00 05 88 00    Right Justified 24Bit(ID:0 REG:5 DATA:88)
s 01             LED0 ON (初期設定)
B 00 00 04 80 00 B0 on SPDIF0 select (ID:0 REG:04 DATA:80)
B 01 00 04 88 00 B1 on SPDIF1 select (ID:0 REG:04 DATA:88)
B 02 00 04 90 00 B2 on SPDIF2 select (ID:0 REG:04 DATA:90)
B 03 00 04 98 00 B3 on SPDIF3 select (ID:0 REG:04 DATA:98)
B 04 00 04 a0 00 B4 on SPDIF4 select (ID:0 REG:04 DATA:a0)
B 05 00 04 a8 00 B5 on SPDIF5 select (ID:0 REG:04 DATA:a8)
B 06 00 04 b0 00 B6 on SPDIF6 select (ID:0 REG:04 DATA:b0)
B 07 00 04 b8 00 B7 on SPDIF7 select (ID:0 REG:04 DATA:b8)
P 00 01 00 00    B0 on -> LED0 ON No-DELAY
P 01 02 00 00    B1 on -> LED1 ON No-DELAY
P 02 04 00 00    B2 on -> LED2 ON No-DELAY
P 03 08 00 00    B3 on -> LED3 ON No-DELAY
P 04 10 00 00    B4 on -> LED4 ON No-DELAY
P 05 20 00 00    B5 on -> LED5 ON No-DELAY
P 06 40 00 00    B6 on -> LED6 ON No-DELAY
P 07 80 00 00    B7 on -> LED7 ON No-DELAY
e
```

例 5） DIX9211(Digital Audio Receiver)を用いて 6ch の入力切替を行う。PORT-B7,6 は SPDIF0-3 入力時の出力フォーマットを設定する。

```
C
C DAI9211 CONTROL COMMAND PROGRAM
C
C INITIALI ROUTINE
C
I 00 80          DAI9211 ADRS 0x80
N                NORMAL RESET
W 0A             WAKEUP 1sec
w 01             delay 1mS after reset
J C0             JUMPER MASK
i 00 2F 04 00    I2S format
i 00 30 12 00    PLL AUTOMATIC
i 00 34 00 00    Initial SPDIF0 select
i 00 35 00 00
i 00 36 00 00
i 00 37 00 00
i 00 60 00 00
i 00 61 10 00
i 00 6b 00 00
S 01 00            LED 0 ON (default SPDIF0 select)
C
C REPEAT ROUTINE
C
B 00 00 34 00 00  SPDIF0 select if B0 PUSH
B 00 00 35 00 00
B 00 00 36 00 00
B 00 00 37 00 00
B 00 00 60 00 00
B 00 00 61 10 00
B 00 00 6b 00 00
B 01 00 34 01 00  SPDIF1 select if B1 PUSH
B 01 00 35 01 00
```

```
B 01 00 36 01 00
B 01 00 37 00 00
B 01 00 60 00 00
B 01 00 61 10 00
B 01 00 6b 00 00
B 02 00 34 02 00  SPDIF2 select if B2 PUSH
B 02 00 35 02 00
B 02 00 36 02 00
B 02 00 37 00 00
B 02 00 60 00 00
B 02 00 61 10 00
B 02 00 6b 00 00
B 03 00 34 03 00  SPDIF3 select if B3 PUSH
B 03 00 35 03 00
B 03 00 36 03 00
B 03 00 37 00 00
B 03 00 60 00 00
B 03 00 61 10 00
B 03 00 6b 00 00
B 04 00 34 C2 00  AUXIN0 select if B4 PUSH
B 04 00 37 02 00
B 04 00 60 33 00
B 04 00 6b 33 00
B 05 00 34 C2 00  AUXIN1 select if B5 PUSH
B 05 00 37 03 00
B 05 00 6b 44 00
B 05 00 60 44 00
C
C LED SETTING
C
P 00 01 00         LED 0 ON(SPDIF0)
P 01 02 00         LED 1 ON(SPDIF1)
P 02 04 00         LED 2 ON(SPDIF2)
P 03 08 00         LED 3 ON(SPDIF3)
P 04 10 00         LED 4 ON(SPDIF4)
P 05 20 00         LED 5 ON(SPDIF5)
C
C JUMPER SETTING ROUTINE
C
M C0 00 2F 04    I2S
M 80 00 2F 05    Left Justifiedf
M 40 00 2F 03    Right Justified 16Bit
M 00 00 2F 00    Right Justified 24Bit
e
```

１１．編集履歴

Revision	DATE	CONTENT
R1	2017.9.29	初版
R2	2017.10.14	プログラム機能追加による修正

OKIRAKU AUDIO 2017 All copyrights reserved