

ユーザーズマニュアル

PIC32 開発ツールキット

MX360-DK



Cyber MELON

株式会社 インターネット

Copyright©2008 Internet Co., Ltd. All Rights Reserved

ユーザーズマニュアル履歴

Rev.	改訂日付	内 容
1.0	2008/11/07	初版リリース
1.01	2009/1/5	誤り修正
1.02	2009/2/6	9 ページ動作チェック RS232C は MX360-ST 専用
1.03	2009/2/11	温度条件、字句微修正

☆本マニュアルの最新版は当社ホームページからダウンロードいただけます。

目次

1. はじめに.....	5
2. 概要	
2.1 特長.....	9
2.2 仕様.....	11
2.3 液晶表示サンプル.....	12
3. ハードウェア	
3.1 ブロック図.....	13
3.2 外観・接続図.....	14
3.3 各部名称と機能.....	15
3.4 コネクタ・ピン配置.....	16
3.5 TTL レベルでのシリアル通信について.....	18
3.6 回路図.....	18
4. ソフトウェア	
4.1 開発フロー.....	19
4.2 開発環境.....	20
4.3 サンプルプロジェクトのビルド.....	21
4.4 FLASH への書き込み.....	22
4.5 MX360 ライブラリの使用.....	25
4.6 ライブラリの仕様と制限事項.....	53
4.7 ライブラリのライセンスについて.....	54

5. その他

5.1 電気的特性.....	55
5.2 オプション製品.....	56
5.3 特注仕様.....	56

1. はじめに

このたびはサイバーメロン・マイコンボードシリーズ **MX360-DK** 開発ツールキットをお買いあげいただきまして、誠に有り難うございます。

MX360-DK は高性能の 32 ビット PIC マイコンに各種 I/O 機能を備え学習、製品開発、組み込みなどに最適な汎用製品です。

ご使用に当たりましては本書を良くお読みいただき正しい取り扱い方法をご理解の上ご使用いただきますようお願い致します。

ご注意事項



1. 本製品の仕様、および本書の内容に関して事前の予告なく変更することがありますのでご了承ください。
2. 本製品の使用によるお客様の損害、および第三者からのいかなる請求につきましても当社はその責任を負いかねますので予めご了承ください。
3. 本製品に付属のソフトウェア・ライブラリおよびサンプルプログラムはその動作を完全保証するものではありません。製品に組み込んで使用される場合にはユーザー様にて十分なテストと検証をお願いします。
ソフトウェアの最新版はユーザー登録後、当社ホームページからダウンロードしていただけます。
4. 本製品および本書に関し、営利目的での複製、引用、配布は禁止されています。
5. 本マニュアルは MX360-ST (MX360 ボードと電源のみのセット) と共通になっております。
MX360-DK だけに適用され、MX360-ST には適用されない項目は **DK** マークで示します

ご使用に当たって



1. 梱包品の内容をまずご確認ください。
2. ご使用になる前に下記の安全についての注意を必ずお読みください。
3. 通電する前に、本製品の使い方を十分ご確認ください、正しい接続と設定をご確認ください。

安全についてのご注意



1. 本製品を医療機器など人命に関わる装置や高度な信頼性・安全性を要求される装置へ搭載することはご遠慮ください。
その他の装置に搭載する場合でもユーザー様にて十分な信頼性試験・評価をおこなった上で搭載してください。
また非常停止や緊急時の制御は外部の独立した回路にておこなってください。
2. 本製品の改造使用は発熱、火災などの原因となり危険ですのでご遠慮ください。
3. 本製品のマニュアル記載環境以外でのご使用は故障、動作不良などの原因になりますのでご遠慮ください。
4. 本製品は導電部分が露出しておりますので、金属パーツなどショートの可能性のあるもの、液体のこぼれる可能性のある場所の近くでの使用はお控えください。
また装置に組み込む場合も絶縁に関しては十分な注意を払ってください。
5. 電源は必ず本製品専用（指定）のものをお使いください。
電圧、極性、プラグ形状など異なるものをご使用になりますと故障の原因となるばかりでなく、火災など重大事故に繋がる危険性があります。
6. 本製品に触れる前には体から静電気を除去してください。
7. 本製品には落下など強い衝撃を与えないでください。

使用環境



- 以下の環境でのご使用はお控えください。
 - ・強い電磁界や静電気などのある環境
 - ・直射日光の当たる場所、高温になる場所
 - ・氷結や結露のある場所、湿度の異常に高い場所
 - ・薬品や油、塩分などのかかる場所
 - ・可燃性の気体、液体などに触れる場所
 - ・振動の多い場所、本製品が静止できない場所
 - ・基板のショートを引き起こす可能性のある場所

規格取得など



- 本製品は UL CSA 規格、CCC 認証などいっさい取得しておりません。装置に組み込む場合は各安全規格への適合性をユーザー様でご確認いただき、対応して頂きますようお願いいたします。
- また本製品は RoHS (特定有害物質の使用制限指令)に対応しておりません。

製品保証と修理



- ・本製品の保証は商品到着後 10 日以内の初期不良のみ無償交換とさせていただきます。
本マニュアルに記載するテスト手順にて正しく動作しない場合はただちに電源を切って、当社ホームページのサポートからご連絡ください。 折り返し交換手順をご案内いたします。
- ・保証期間中であってもユーザー様の責となる故障（落下や電源の誤接続など）は有料修理になります。
- ・その他の故障やクレームにつきましても当社ホームページ <http://www.cyber-melon.com> サポートコーナーよりご連絡ください。

梱包内容をご確認ください



DK マークの項目は MX360-ST (ボードと電源のみ) には含まれません。

1. MX360 ボード本体 1



2. 説明書 CD-ROM 1



3. 電源アダプター 1



DK 4. S T N液晶表示器 (本体に接続済み) 1



DK 5. RS232C ケーブル 1



DK 6. SD カード 1



動作チェック



付属の 5V 電源アダプターを MX360 本体に接続して通電してください。

- ・まずボード上 2 カ所の赤い LED が 2 回点滅してボードの初期化が完了したことを知らせます。
- ・ **DK** MX360-DK (液晶表示器付き) の場合は” MMC Card not found” というエラーが一瞬出たあと、液晶パネルにフォントとグラフィックのデモを 5 秒ずつ交互に表示します。
- ・ **DK** 付属の SD カードをパソコンで FAT32 にフォーマットし、付属 CD-ROM の Font フォルダから “kfont12.fnt” ファイルを SD カードのルートにコピーして、MX360 カードソケットに挿入してください。その時点でリセットボタン (黒) を押すと、こんどは漢字のデモを含めた画面 (2.3 の内容) が液晶パネルに 5 秒ずつ表示されます。

■ 以下は MX360-ST 専用の **RS232C 通信テスト** です。

- ・ 付属の RS232C ケーブルでパソコンと接続してください。
パソコンのターミナルソフト (ハイパーターミナル、TeraTerm など) を立ち上げて
Baud rate:38400 , Data: 8bit, Stop: 1bit, Parity: None, 無手順に設定します。
MX360 ボードのリセットボタン (黒) を押すと “Welcome to MX360” がターミナルウィンドウに表示されます。
- ・ PC のターミナルから何か文字を打ち込んで (この時点ではエコーバックしません) CR (Enter) を押してください。
この時点で今まで打ち込んだ 1 行分がエコーバックされます。

もし動作異常が認められた場合は電源をはずして当社ホームページ <http://www.cyber-melon.com> のサポートから症状をご連絡ください。
対処方法をメールまたは電話でご連絡いたします。

2. 概要

MX360 シリーズは Microchip 社の高性能 32 ビット PIC マイコン に SD カードソケット、PS2 キーボード用ポート、RS232C コネクタ、ICSP コネクタなどの I/O を搭載し組み込みアプリケーションの開発を可能にしたボードです。

ブートローダー書き込み済みで、添付の Flash 書き込みソフトでパソコンから RS232C ケーブルで開発したプログラムを転送、書き込みできます。

MX360-DK はこれに STN グラフィック液晶表示器と豊富なソフトウェアライブラリを加えた開発ツールキットで、ASCII/漢字表示や簡易グラフィック、ロングファイルネーム対応の FAT32 ファイルシステムを使った実用的で高度なアプリケーションがすぐ簡単に開発できます。

2.1 特長

- ・ 32 ビット 75.6MHz 動作、命令プリフェッチ、乗除算 1 サイクルハードウェア搭載の強力マイクロプロセッサ。
- ・ ADC, SPI, I2C, UART, PWM, InputCapture, DMA, GPIO, Timer 内蔵。
- ・ I/O ピンは 5V 入力トレラント。
- ・ 浮動小数点や三角関数も扱える math ライブラリ。
- ・ 殆どの I/O ピンをコネクタなどに引き出し済み。
- ・ ブートローダーを書き込み済み。添付の書き込みソフトで開発したプログラムが書き込めるので別途書き込み器が不要。
- ・ CONFIG ワードの設定、割り込み設定など、初心者にも敷居の高い内容を含むサンプルソースコードを提供。サンプルプロジェクトを改造することで、すぐに確実に動作するプログラムを作成できます。

以下は **DK** MX360-DK (開発ツールキット) のみの対応です。

- ・ STN モノクロ液晶表示器 (128x64 dot) を搭載。
- ・ 同梱のソフトウェア・ライブラリでロングファイルネーム対応の FAT32 ファイルシステム、JIS 第一水準漢字フォント、漢字フォントドライバ、簡易グラフィックドライバ、RS232C 通信、PS2 キーボード入力など豊富な機能をすぐに利用可能。
- ・ サンプルプログラムの応用で簡単に漢字・簡易グラフィックターミナル

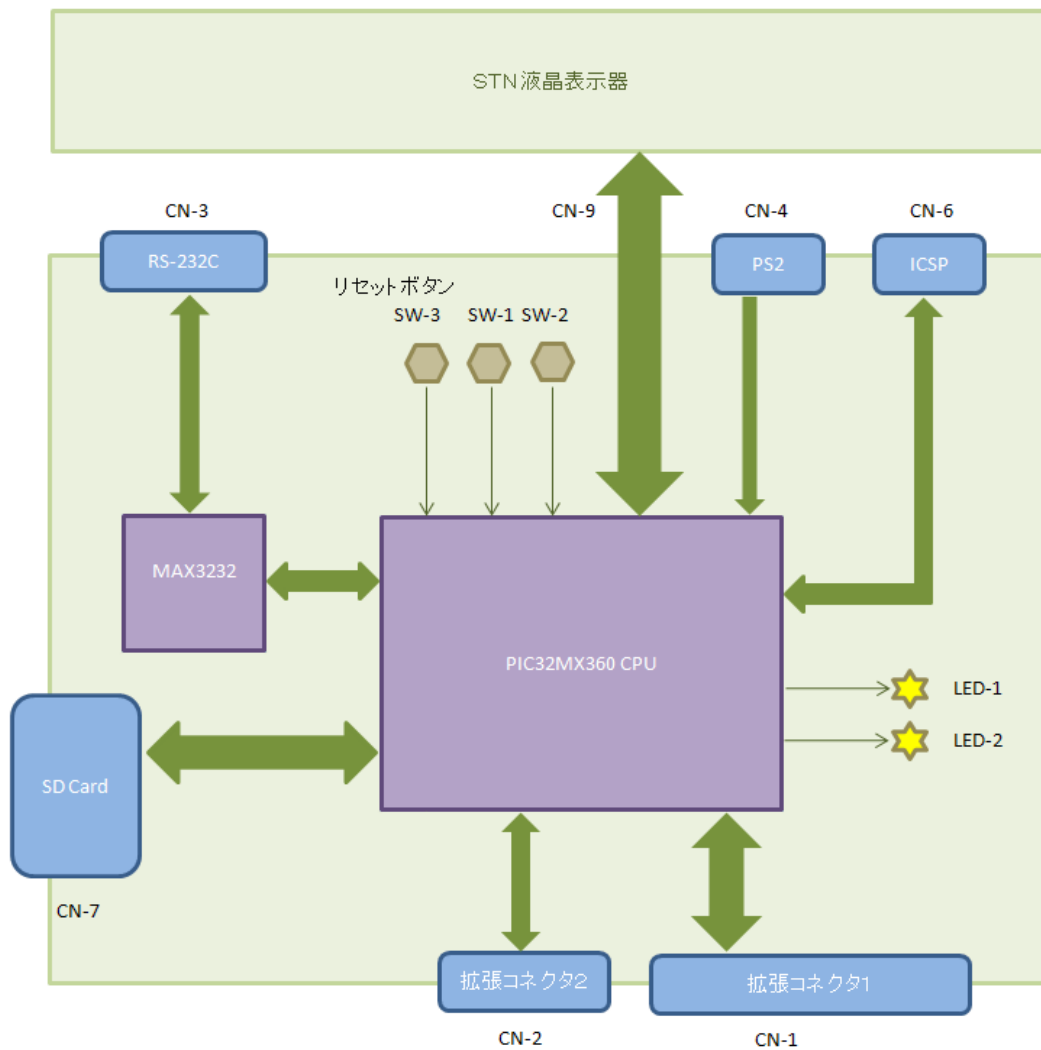
2.2 ボード仕様

項目	仕様
CPU	PIC32MX360F512L
	Flash ROM 容量 512KB, 内蔵 RAM 容量 32KB
動作クロック	内部 75.6MHz, 周辺バスクロック 18.9MHz (プログラムにより変更可能)
電源電圧	DC5V 入力 (CPU 周辺動作 3.3V)
入出力	RS232C コネクタ
	PS2 コネクタ (キーボード読み込み専用)
	MMC/SD カードソケット
	ICSP (インサーキット・シリアルプログラミング) コネクタ
	STN 液晶表示器用コネクタ
	タクトスイッチ (2個)
	LED (2個)
	リセットスイッチ
	汎用 I/O コネクタ 1 (50ピン)
	汎用 I/O コネクタ 2 (14ピン) 含アナログ系
基板寸法	99.1(W) x 79.4(D) x 16.8(H) mm (基板のみ)
	99.1(W) x 83.9(D) x 37.4(H) mm (突起部、液晶含)
重量	138g (液晶表示器含む)

3. ハードウェア

3.1 ブロック図

MX360 ボード・ブロックダイアグラム

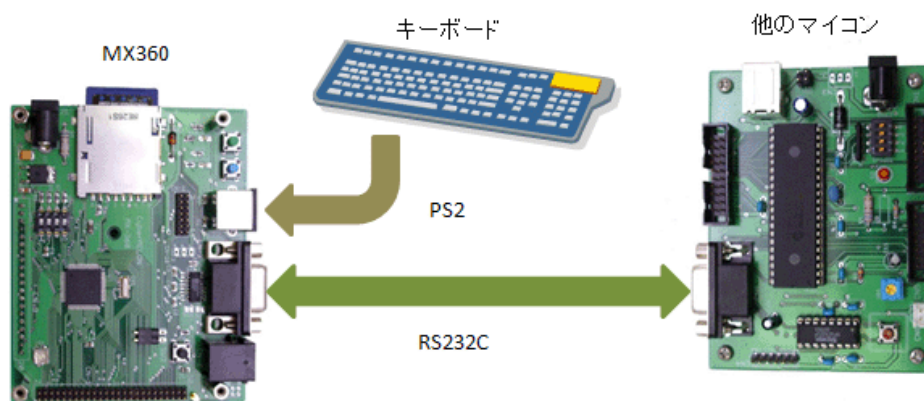


3.2 外観・接続図

DK MX360-DK の外観

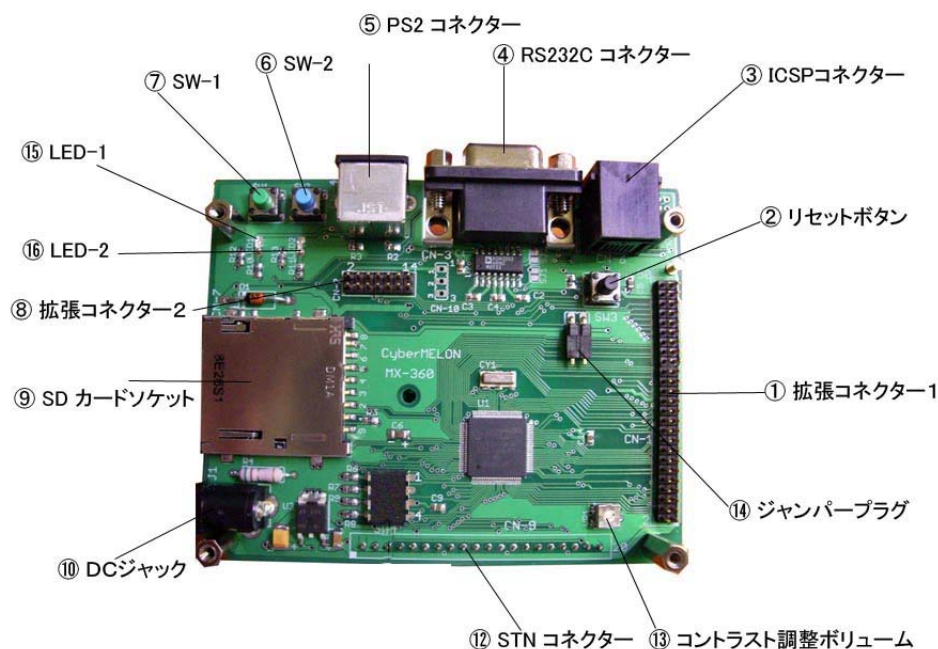


プログラムの書き込みはP Cから RS232C で可能



他のマイコンから漢字・グラフィック端末、カードリーダーとして MX360 を使う場合など

3.3 各部名称と機能



- | | |
|-----------------|------------------------|
| ① 拡張コネクタ-1 | 詳細は 3.4 を参照してください。 |
| ② リセットボタン | CPU をリセットします。 |
| ③ ICSP コネクタ | 外部プログラマー/デバッガ-用です。 |
| ④ RS232C コネクタ | 9 ピン D-SUB メスのコネクタ-です。 |
| ⑤ PS2 コネクタ | キーボード入力専用です。 |
| ⑥ SW-2 | 汎用スイッチ入力 |
| ⑦ SW-1 | 汎用スイッチ入力 |
| ⑧ 拡張コネクタ-2 | 詳細は 3.4 を参照してください。 |
| ⑨ SD カードソケット | MMC/SD カード挿入用。 |
| ⑩ DC ジャック | +5V DC 入力。 |
| ⑪ | |
| ⑫ STN コネクタ | 裏に STN 液晶表示器を接続します。 |
| ⑬ コントラスト調整ボリューム | 液晶表示器のコントラストを調整します。 |
| ⑭ ジャンパープラグ | RS232C の受信をイネ-ブルします。 |
| ⑮ LED-1 | 汎用 LED。 |
| ⑯ LED-2 | 汎用 LED。 |

3.4 コネクタ・ピン配置

CN-1 拡張コネクタ-1

PIN	機能	接続先	PIN	機能	接続先
1	GND		2	GND	
3	RE8	U1-18	4	RE9	U1-19
5	RB0	U1-25	6	RB1	U1-24
7	RB2	U1-23	8	RB3	U1-22
9	RB4	U1-21	10	RB5	U1-20
11	RB6	U1-26	12	RB7	U1-27
13	RB8	U1-32	14	RB9	U1-33
15	RB10	U1-34	16	RB11	U1-35
17	RB12	U1-41	18	RB13	U1-42
19	RB14	U1-43	20	RB15	U1-44
21	RF2	U1-52	22	RF3	U1-51
23	RF6	U1-55	24	RF7	U1-54
25	RF8	U1-53	26	+5V	
27	RG0	U1-90	28	RG1	U1-89
29	RG2	U1-57	30	RG3	U1-56
31	RD0	U1-72	32	RD1	U1-76
33	RD2	U1-77	34	RD3	U1-78
35	RD4	U1-81	36	RD5	U1-82
37	RD6	U1-83	38	RD7	U1-84
39	RD8	U1-68	40	RD9	U1-69
41	RD10	U1-70	42	RD11	U1-71
43	RD12	U1-79	44	RD13	U1-80
45	RD14	U1-47	46	RD15	U1-48
47	+3.3V		48	+3.3V	
49	GND		50	GND	

CN-2 拡張コネクタ-2

PIN	機能	接続先	PIN	機能	接続先
1	GND		2	GND	
3	RA0	U1-17	4	RA1	U1-38
5	RA2	U1-58	6	RA3	U1-59
7	RA4	U1-60	8	RA5	U1-61
9	RA6	U1-91	10	RA7	U1-92
11	RA9	U1-28	12	RA10	U1-29
13	+3.3V		14	+3.3V	

CN-3 RS232C コネクタ-

PIN	機能	接続先	PIN	機能	接続先
1			2	TxD	U2-14
3	RxD	U2-13	4		
5	GND		6		
7	CTS	SJ2	8	RTS	SJ1
9					

CN-4 PS2 コネクタ-

PIN	機能	接続先	PIN	機能	接続先
1	Data	R2	2	TxD	U2-14
3	GND		4	+5V	
5	Clock	R3	6		

CN-6 ICSP コネクタ-

PIN	機能	接続先	PIN	機能	接続先
1	MCLR-/Vpp	U1-13	2	VDD	+3.3V
3	GND		4	PGD	U1-25
5	PGC	U1-24	6	未使用	

CN-9 STN 液晶表示器 コネクタ

PIN	機能	接続先	PIN	機能	接続先
1	GND		2	+5V	
3	VO	VR3	4	D/I	U1-6
5	R/W	U1-7	6	E	U1-8
7	DB0	U1-93	8	DB1	U1-94
9	DB2	U1-98	10	DB3	U1-99
11	DB4	U1-100	12	DB5	U1-3
13	DB6	U1-4	14	DB7	U1-5
15	CS1	U1-73	16	CS2	U1-74
17	RESET	U1-9	18	Vout	VR3
19	A	R4	20	K	GND

CN-10 RS232C コネクタ (未実装)

PIN	機能	接続先	PIN	機能	接続先
1	GND		2	TxD	U1-50
3	RxD	U1-49			

3.5 TTLレベルでのシリアル通信について

CN-10 を実装すれば MAX3232 を介さない TTL レベルでのシリアル通信が可能になります。

この場合は MAX3232 からの Rx 信号と衝突しないようジャンパープラグ J1 (リセットスイッチの左下) を必ずはずしてください。

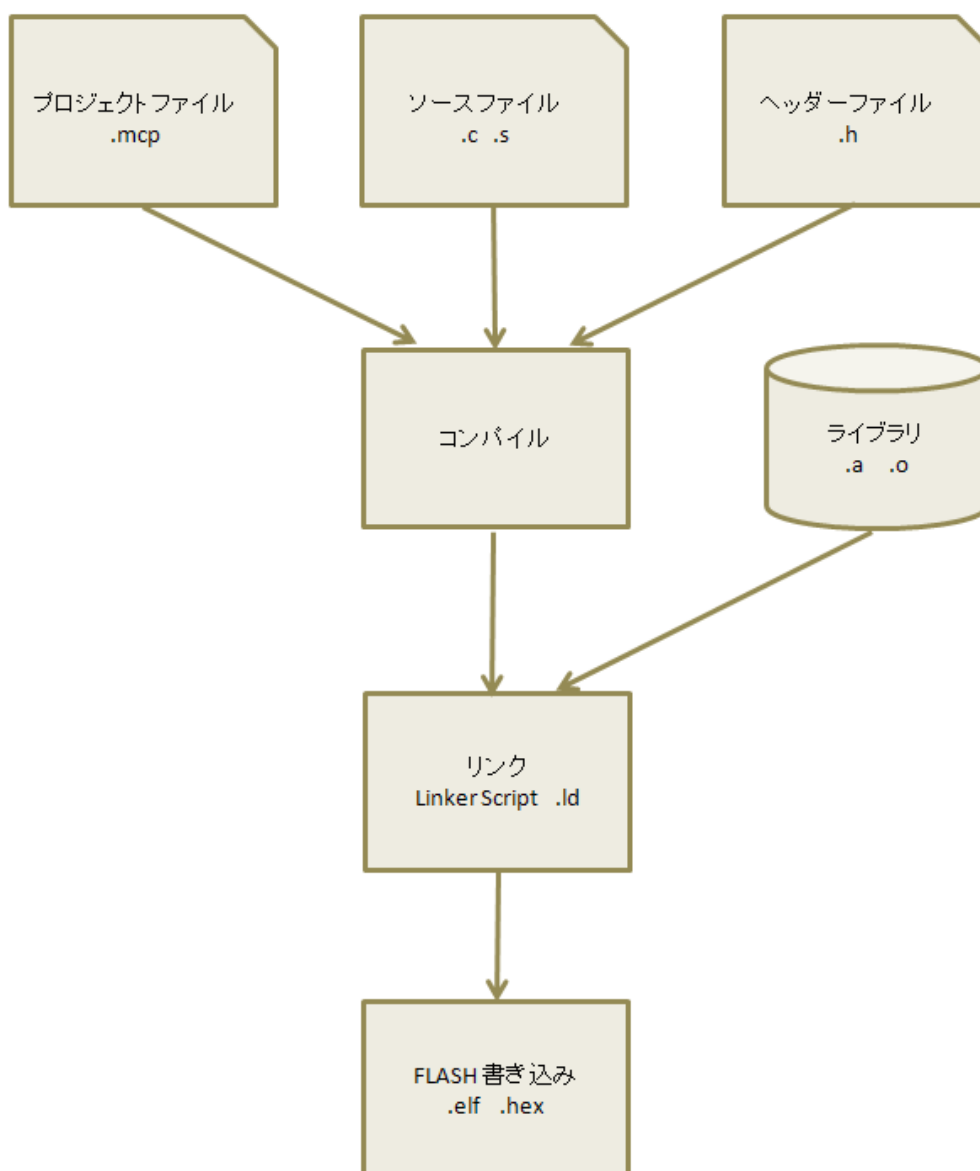
尚、このケースでは CTS, RTS を使ったハードウェアによるハンドシェイクは使えません。

3.6 回路図

MX360 ボードの回路図は付属 CD-ROM の Documents フォルダの MX360_sch.pdf をご覧ください。

4. ソフトウェア

4.1 開発フロー



- ★ MX360 ライブラリはサンプルプロジェクトの中にオブジェクトファイル(.o)として含まれていますので、これをコピーしてご利用ください。

4.2 開発環境

統合開発環境(MPLAB IDE)をマイクロチップ社のホームページ

<http://www.microchip.com>

から無料でダウンロードすることができます。

最新版はトップページから左の Design タブの” MPLAB IDE” をクリックして
MPLAB Integrated Development Environment のページに行き IDE downloads
をクリックしてダウンロードし、IDE をインストールしてください。

OS は Windows のみの対応となります。

プログラムの開発にはこの他に C32 コンパイラが必要です。C32 コンパイラは
無料の Student Edition が評価用に使えます。

マイクロチップ社のトップページから左の Design タブの “Development Tools”
をクリックし、飛んだページの Software “MPLAB C Compilers “をクリック
してコンパイラのダウンロードページを表示します。

ページの下の方にある “MPLAB C Compiler for PIC32 MCUs vX.XX - Student Edition”
をクリックしてダウンロード、インストールしてください。(X.XX は最新のバージョン)
Student Edition のダウンロードにはその前に簡単なユーザー登録が必要です。

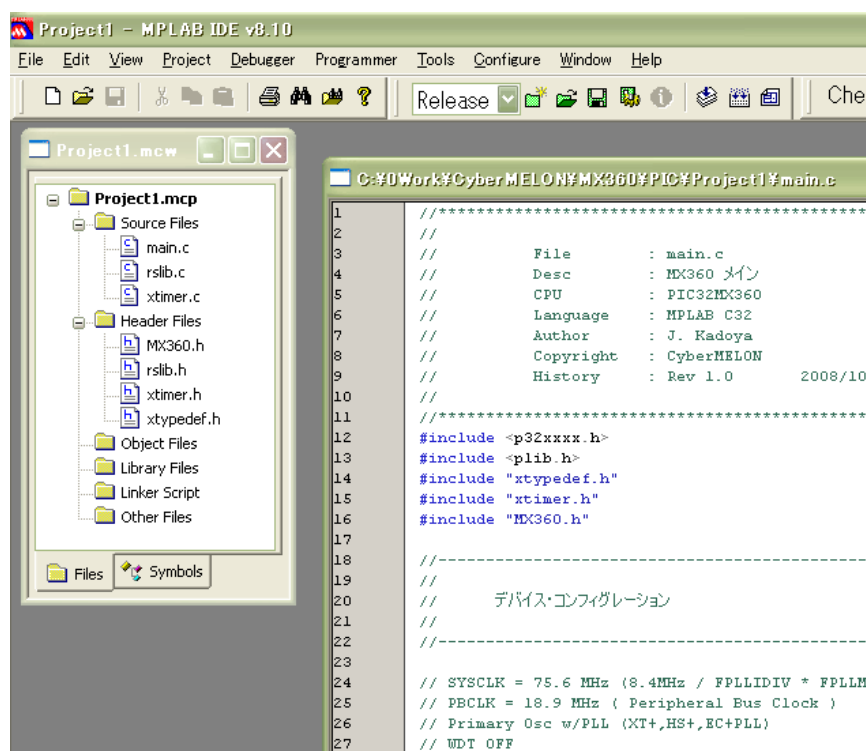
Student Edition は一定期間を過ぎるとコンパイラの最適化など一部の機能が
使えなくなりますが、コンパイルは継続して無料で可能です。

4.3 サンプルプロジェクトのビルド

- CD-ROM の “Sample Project” フォルダをどこかハードディスクの適当な場所にコピーしてください。
- プロジェクトフォルダの中の .mcp ファイルをダブルクリックすると MPLAB IDE が立ち上がります。

または MPLAB IDE を立ち上げて **Project** メニューの **Open** でプロジェクトファイルを選択します。 **View** メニューの **Project** および **Output** にチェックを入れると図のようにプロジェクトファイルの一覧が表示されます。

ソースファイルをダブルクリックするとエディターで編集できるようになります。



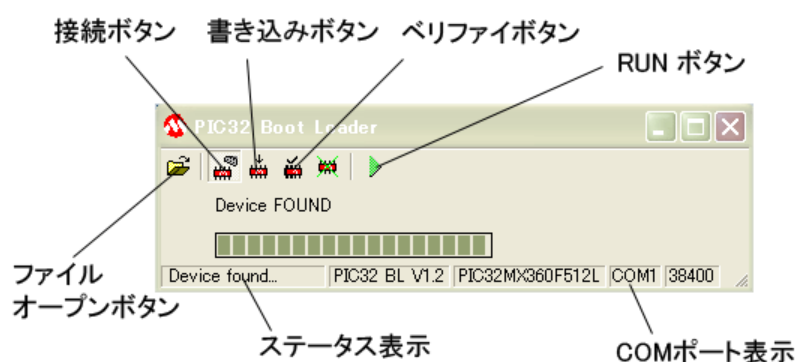
- **Project** メニューの “**Make**” または “**Build All**” をクリックするとビルドがおこなわれます。
- ビルドが成功すれば Output ウィンドウに **BUILD SUCCEEDED** が表示されます。
- エラーの場合は Output ウィンドウのエラーの箇所をダブルクリックするとソース上のエラーの箇所に飛ぶことができます。

4.4 FLASH への書き込み

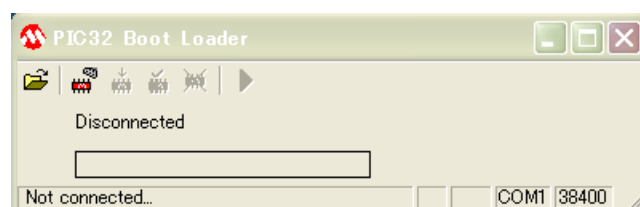
作成したプログラムを MX360 マイコンの FLASH ROM に書き込んでみます。

CD-ROM の “Flash Writer” フォルダにある PIC32BL.exe をハードディスクの適切な場所にコピーして実行します。

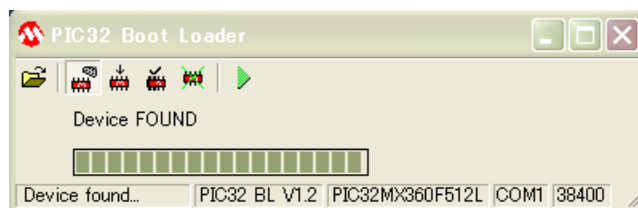
以下に各部の名称を示します。



- MX360 をパソコンと RS232C ケーブルで接続します。パソコンに RS232C 端子がない場合は USB - シリアル変換ケーブルなどをご用意ください。
- PIC32BL.exe (以下ライターソフトと呼びます) の COM ポート表示部を右クリックして使用する COM ポートを選択してください。



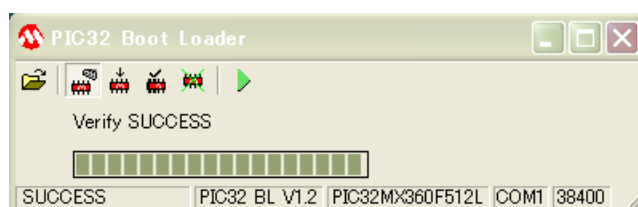
- ライターソフトの接続ボタンを押すとバーグラフが動き始めます。その状態で MX360 のリセットボタンを押すとボード上の LED が速く点滅しブートローダーがデータ転送待ちになったことを示します。またライターソフトの接続表示が “Device found” になって通信が確立されたことを示します。また通信によって取得されたデバイスの型名 (PIC32MX360F512) とブートローダーのバージョンを表示します。



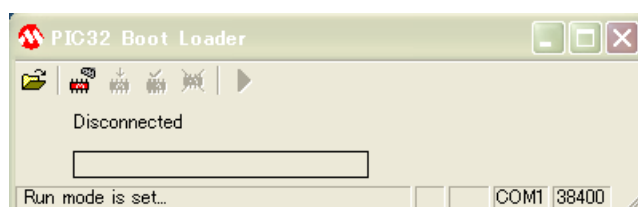
- ファイル・オープンボタンをクリックして書き込みたい HEX ファイルを選択します。
- ライターソフトの書き込みボタンをクリックすると、上書きして良いか?の確認ダイアログが出ますので OK をクリックします。ライターソフトは選択した HEX ファイルを MX360 に転送して書き込みを開始します。



- 書き込みが終了すると VERIFY をおこない、正しく書けていれば図のような SUCCESS 表示になります。



- 三角の RUN ボタンをクリックするか、RS232C ケーブルをはずしてボードのリセットボタンを押すと、書き込んだプログラムを実行します。



一度プログラムをライターソフトで書き込むと、FLASH の末尾にチェックコードが記録され、以後リセットが押されるとブートローダーがこれをチェックしてライターソフトから接続要求がなければ自動的にユーザープログラムにジャンプします。

純正プログラマ使用時のご注意



MPLAB ICD-2, ICD-3, REAL ICE など 外部ツールで書き込み、またはイレースを一度でもおこないますと MX360 のブートローダー領域を上書きして壊します。万一誤ってブートローダー領域を壊してしまった場合は別途書き込みサービス（有料）をおこなっておりますので、当社ホームページ <http://www.cyber-melon.com> のサポートよりご連絡ください。

もちろん外部ツールでプログラミングをおこなう場合はブートローダー領域を壊しても何ら問題ありません。

ブートローダーは MX360 付属の書き込みソフトでプログラムを書き込む場合にのみ必要です。

尚、ブートローダーのオブジェクトファイルは付属 CD-R の “Hex Files” フォルダに “BootLdr.hex” という名前で格納されております。

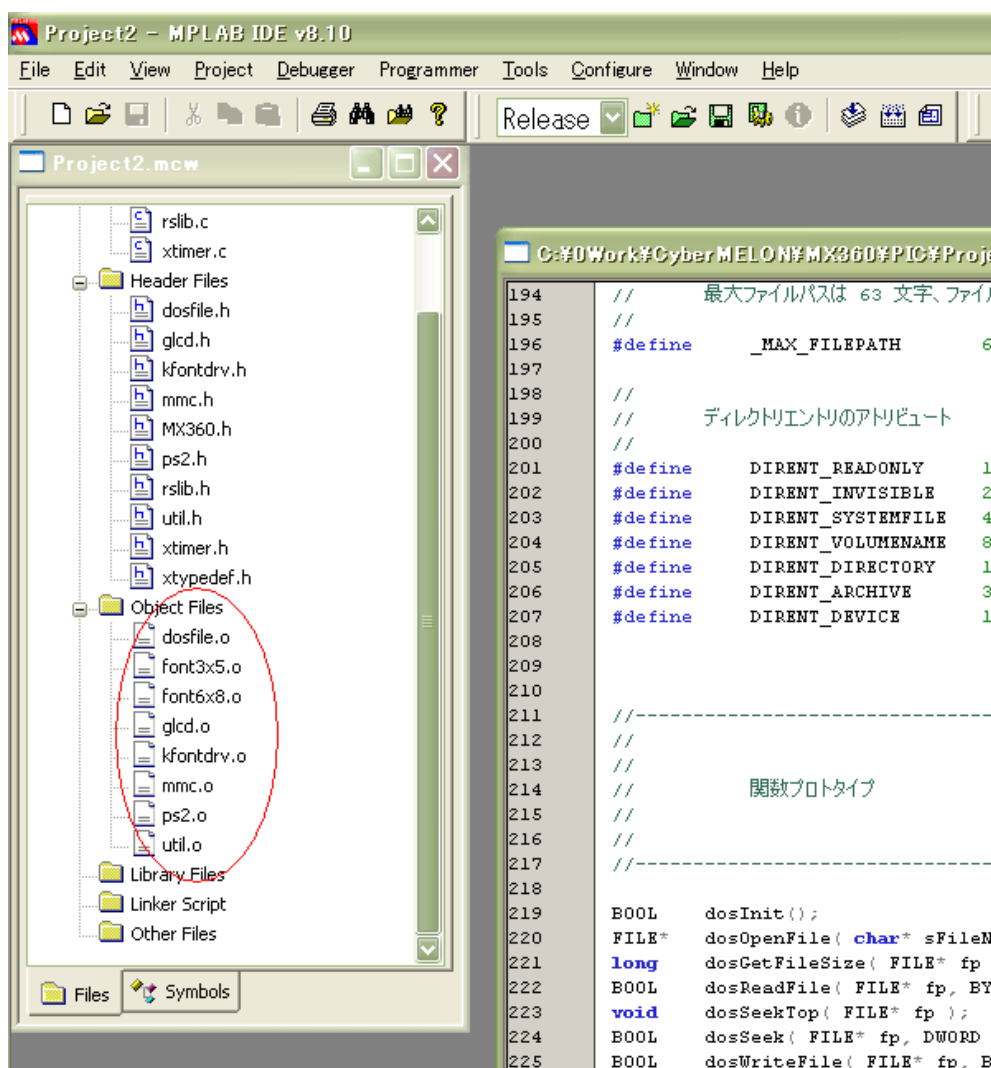
尚、MX360-DK（開発ツールキット）の場合は このフォルダに “MX360Demo.hex” という名前で書き込み済みのデモプログラムが格納されています。

もしデモプログラムに戻したい場合はこのファイルをライターで書き込んでください。

4.5 **DK** MX360 ライブラリの使用

- MX360-DK 開発ツールキットのサンプルプロジェクト Project2 には SD カードから FAT32 ファイルシステムを使うための関数や漢字フォントドライバーなどのライブラリが含まれています。(MX360-ST には含まれませんが MX360-DK を先にご購入の場合、コピーして MX360-ST でお使いいただくことは可能です)

Project2.mcp を開くと以下のオブジェクトがプロジェクトに組み込まれているのがわかります。ライブラリはアーカイブ化されておりませんので、拡張子 .o のオブジェクトファイルを必要に応じて 下図のようにプロジェクトウィンドウの **Object Files** に追加して使います。他のプロジェクトで使用するときもオブジェクトと対応するヘッダーファイルをコピーしてお使いください。尚、ライブラリの最新版は当社ホームページよりダウンロードしていただけます。



以下に各ライブラリオブジェクトの関数と機能を説明します。

変数の Type(型)の表記についてはプロジェクトに含まれている `xtypedef.h` ファイルを参照してください。

ライブラリ	ヘッダー	概略機能	参照章/頁
<code>dosfile.o</code>	<code>dosfile.h</code>	ロングファイルネーム対応 FAT32 ファイルシステムを実装。	4.5.1 /27
<code>mmc.o</code>	<code>mmc.h</code>	MMC/SD カードの初期化とセクターリード・ライトをおこなう。	4.5.2 /32
<code>glcd.o</code>	<code>glcd.h</code>	STN液晶表示器にアスキーキャラクターや簡易グラフィックを表示する。	4.5.3 /34
<code>font6x8.o</code> <code>font3x5.o</code>	なし	ASCII 文字フォント。 <code>glcd.o</code> モジュールから使用。	4.5.4 /38
<code>kfontdrv.o</code>	<code>kfontdrv.h</code>	漢字フォントドライバーモジュール。	4.5.5 /39
<code>ps2.o</code>	<code>ps2.h</code>	PS2 キーボード入力モジュール。	4.5.6 /42
<code>rslib.o</code>	<code>rslib.h</code>	UART 通信モジュール。	4.5.7 /43
<code>xtimer.o</code>	<code>xtimer.h</code>	タイマーモジュール。	4.5.8 /46
<code>util.o</code>	<code>util.h</code>	汎用ユーティリティーモジュール。	4.5.9 /47

4.5.1 dosfile.o (ヘッダーファイル: dosfile.h)

概要: ロングファイルネーム対応 FAT32 ファイルシステムを実装します。

ファイルのオープン、リード、ライト、シーク、クローズなどを行います

関数名	dosInit	
機能	ファイルシステムの初期化	
引数	タイプ	内容
なし		
戻り値	BOOL	TRUE: 成功、FALSE: 失敗
備考	ファイルシステム使用前に一度だけコールしてください。	

関数名	dosOpenFile	
機能	ファイルを名前でオープン	
引数	タイプ	内容
sFileName	char*	ファイル名
sMode	char*	リード時: "r" ライト時 "w"
戻り値	FILE*	成功: ファイル構造体へのポインタ 失敗: NULL
備考		

関数名	dosGetFileSize	
機能	ファイルのサイズを得る	
引数	タイプ	内容
fp	FILE*	dosOpenFile で返されたファイル構造体ポインタ
戻り値	long	ファイルのサイズ (バイト数) を返す
備考		

関数名	dosReadFile	
機能	ファイルの内容を読み込む	
引数	タイプ	内容
fp	FILE*	dosOpenFile で返されたファイル構造体ポインタ
pBuf	BYTE*	データを受け取るバッファへのポインタ
nLength	int	読み込ませるバイト数
戻り値	BOOL	TRUE: 成功、FALSE: 失敗
備考	現在の SEEK 位置から続けて読み込む	

関数名	dosSeek	
機能	指定位置までシークする	
引数	タイプ	内容
fp	FILE*	dosOpenFile で返されたファイル構造体ポインタ
dwPos	DWORD	シークするポジション (先頭からのバイト数)
戻り値	BOOL	TRUE: 成功、FALSE: 失敗
備考		

関数名	dosSeekTop	
機能	ファイル先頭にシークする	
引数	タイプ	内容
fp	FILE*	dosOpenFile で返されたファイル構造体ポインタ
戻り値		なし
備考		

関数名	dosWriteFile	
機能	ファイルの書き込み	
引数	タイプ	内容
fp	FILE*	dosOpenFile で返されたファイル構造体ポインタ
pBuf	BYTE*	データバッファへのポインタ
nLength	int	書き込ませるバイト数
戻り値	BOOL	TRUE: 成功、FALSE: 失敗
備考	現在の SEEK 位置から続けて書き込む。同名のファイルが既にカレントディレクトリに存在する場合は上書きされる。	

関数名	dosCloseFile	
機能	ファイルのクローズ	
引数	タイプ	内容
fp	FILE*	dosOpenFile で返されたファイル構造体ポインタ
戻り値	BOOL	TRUE: 成功、FALSE: 失敗
備考	リードモードでオープンした場合もメモリー領域を開放するために必ず最後にコールしてください	

関数名	dosDeleteFile	
機能	ファイルのクローズ	
引数	タイプ	内容
sFileName	char*	ファイル名
戻り値	BOOL	TRUE: 成功、FALSE: 失敗
備考		

関数名	dosOpenDirectory	
機能	カレントディレクトリの移動	
引数	タイプ	内容
sDirName	char*	ディレクトリ名
戻り値	BOOL	TRUE: 成功、FALSE: 失敗
備考	<p>この名前の下ディレクトリ階層に移動します。</p> <p>ディレクトリ名が NULL の場合はルートディレクトリに移動します。</p> <p>以後、ファイルのオープン、削除などはこのディレクトリの中でおこなわれます。</p>	

関数名	dosCreateDirectory	
機能	新しいディレクトリの作成	
引数	タイプ	内容
sDirName	char*	ディレクトリ名
戻り値	BOOL	TRUE: 成功、FALSE: 失敗
備考	ディレクトリの作成によってカレントディレクトリは移動しません。 作成したディレクトリに移動するには dosOpenDirectory で カレントディレクトリを変更してください。 同名のディレクトリが同じ階層に既に存在するとエラーになります。	

4.5.2 mmc.o (ヘッダーファイル: mmc.h)

概要: MMC/SD カードの初期化とセクターリード・ライトをおこなうモジュールで dosfile.o から内部で使われますので dosfile.o を使用するときは必ず mmc.o もプロジェクトに追加してください。

関数名	mmcInit	
機能	マルチメディアカードの初期化	
引数	タイプ	内容
なし		
戻り値	int	0: 成功、> 0 エラーコード
備考	dosInit の前に一度だけコールしてください	

mmc エラーコード

エラー名	数値	内容
MMC_ERR_SUCCEEDED	0	エラーなし (成功)
MMC_ERR_REPLY	1	コマンドに応答がない
MMC_ERR_TIMEOUT	2	タイムアウト
MMC_ERR_DATA_WRITE	3	書き込みエラー

関数名	mmcReadSector	
機能	セクターの読み込み	
引数	タイプ	内容
iStartSector	int	開始セクター番号
iSectorLength	int	読み込むセクター数
pBuf	BYTE*	データを受け取るバッファへのポインタ
戻り値	BOOL	TRUE: 成功、FALSE: 失敗
備考		

関数名	mmcSectorWrite	
機能	セクターの書き出し	
引数	タイプ	内容
iStartSector	int	開始セクター番号
iSectorLength	int	書き出すセクター数
pBuf	BYTE*	データバッファへのポインタ
戻り値	BOOL	TRUE: 成功、FALSE: 失敗
備考		

関数名	mmcIsReady	
機能	マルチメディアカードが使える状態か調べる	
引数	タイプ	内容
なし		
戻り値	BOOL	TRUE: 使用可能、FALSE: 使用不可
備考	MX360 ではホットプラグはサポートしない	

制限事項

MX360 の MMC/SD カードはカードの検出をおこなっておりません。

従いましてホットプラグ（電源投入中のカードの抜き差し）にも対応できません。

4.5.3 glcd.o (ヘッダーファイル: glcd.h)

概要: STN液晶表示器にアスキーキャラクターや簡易グラフィックを表示するためのモジュールです。

関数名	glcdInit	
機能	液晶表示器の初期化	
引数	タイプ	内容
なし		
戻り値		なし
備考	使用前に一度だけコールしてください	

関数名	glcdClear	
機能	液晶表示を全画面消去	
引数	タイプ	内容
なし		
戻り値		なし
備考		

関数名	glcdPutChar	
機能	ASCII 文字の表示	
引数	タイプ	内容
code	BYTE	文字コード(0x20.. 0xDF)
x	short	X 方向座標 (文字単位 0..20)
y	short	Y 方向座標 (文字単位 0..7)
戻り値		なし
備考		

関数名	gLCDPutString	
機能	ASCII 文字列の表示	
引数	タイプ	内容
pMsg	char*	文字列 (0 ターミネート)
x	short	X 方向開始座標 (文字単位 0..20)
y	short	Y 方向開始座標 (文字単位 0..7)
戻り値		なし
備考		

関数名	gLCDPutDot	
機能	点の描画	
引数	タイプ	内容
bOnOff	BYTE	0: ドット非表示、1: ドット表示
x	short	X 方向グラフィック座標 (0..127)
y	short	Y 方向グラフィック座標 (0..63)
戻り値		なし
備考		

関数名	gLCDPutSmallChar	
機能	小 ASCII フォント文字の表示	
引数	タイプ	内容
code	BYTE	文字コード ('0' ~ '9', 'A' ~ 'F' のみ)
x	short	X 方向グラフィック座標 (0..127)
y	short	Y 方向グラフィック座標 (0..63)
戻り値		なし
備考		

関数名	glcdPutCharRev	
機能	ASCII 文字の反転表示	
引数	タイプ	内容
code	BYTE	文字コード(0x20.. 0xDF)
x	short	X方向座標 (文字単位 0..20)
y	short	Y方向座標 (文字単位 0..7)
戻り値		なし
備考		

関数名	glcdRevChar	
機能	文字の反転	
引数	タイプ	内容
x	short	X方向座標 (文字単位 0..20)
y	short	Y方向座標 (文字単位 0..7)
戻り値		なし
備考	既に描画済みの文字を白黒反転する	

関数名	glcdDrawLine	
機能	線の描画	
引数	タイプ	内容
xStart	short	開始点のX座標(0..127)
yStart	short	開始点のY座標(0..63)
xEnd	short	終了点のX座標(0..127)
yEnd	short	終了点のY座標(0..63)
戻り値		なし
備考		

関数名	gldFlipBackgroundBuf	
機能	描画バッファの内容を液晶に表示する	
引数	タイプ	内容
戻り値		なし
備考	fontPutString12Background() で文字列をバッファに展開したあと、この関数で実際の描画をおこないます。	

関数名	gldClearBackgroundBuf	
機能	描画バッファの内容をクリアする	
引数	タイプ	内容
戻り値		なし
備考		

4.5.4 font6x8.o font3x5.o (ヘッダーファイル:なし)

概要: ASCII 文字フォントです。 glcd.o モジュールから使用します。
glcd.o モジュールを使用するには必ずプロジェクトに追加してください。

漢字フォント描画のサンプルプログラム

1. 任意の位置への描画例

```
fontPutString12ByDot( " 夜をこめて 鳥の ", 10, 0 );  
fontPutString12ByDot( "空音は はかるとも ", 10, 13 );  
fontPutString12ByDot( " よに逢坂の ", 10, 26 );  
fontPutString12ByDot( " 関はゆるさじ ", 6, 39 );  
fontPutString12ByDot( "※※ 清少納言 ※※", 10, 52 );
```

// きめ細かい位置の指定ができるが、表示は遅い

2. 高速な表示例

```
glcdClearBackgroundBuf();  
fontPutString12Background( "1 我が輩は猫である。名", 0, 0 );  
fontPutString12Background( "2 前はまだ無い。どこで", 0, 1 );  
fontPutString12Background( "3 生まれたか頓と見当が", 0, 2 );  
fontPutString12Background( "4 つかぬ。何でも薄暗い", 0, 3 );  
fontPutString12Background( "5 じめじめした所でニャ", 0, 4 );  
glcdFlipBackgroundBuf();
```

// 予め内部描画バッファに展開してから一挙に画面全体を表示する

4.5.5 kfontdrv.o (ヘッダーファイル: kfontdrv.h)

概要: 漢字フォントドライバーです。

現バージョンでは漢字フォントは SD カードのルートの **kfont12.fnt** ファイルから読み出します。

必要な作業: 付属のSDカードは工場出荷時に FAT16 でフォーマットされているのでまずお手持ちのパソコンで **FAT32** でフォーマットしなおしてください。

当ライブラリの Ver. 1.0 は FAT32 オンリーのサポートです。

そして付属 CD-ROM の Font フォルダから kfont12.fnt ファイルを SD カードのルートフォルダにコピーしてください。

関数名	fontOpenKFont12	
機能	漢字フォントドライバーの初期化	
引数	タイプ	内容
なし		
戻り値	BOOL	TRUE: 成功、FALSE: 失敗
備考	使用前に一度だけコールしてください。	

関数名	fontCloseKFont12	
機能	漢字フォントドライバーの終了処理 (メモリー解放)	
引数	タイプ	内容
戻り値		
備考	使用後に一度だけコールしてください。メモリー領域を開放します。	

関数名	fontPutString12ByDot	
機能	(漢字を含む) 文字列の任意位置への表示	
引数	タイプ	内容
pMsg	char*	Shift-JIS 文字列(0 ターミネート)
xPos	short	表示開始Xグラフィック座標(0..127)
yPos	short	表示開始Yグラフィック座標(0..63)
戻り値	なし	
備考	<p>座標は左上のドット位置。</p> <p>高速の描画が必要な場合は fontPutString12Background をお使いください。(但し表示位置は半角文字単位となります)</p>	

関数名	fontPutString12Background	
機能	(漢字を含む) 文字列の内部バッファへの高速展開	
引数	タイプ	内容
pMsg	char*	Shift-JIS 文字列(0 ターミネート)
xPos	short	表示開始Xキャラクター座標(0..20)
yPos	short	表示開始Yキャラクター座標(0..4)
戻り値	なし	
備考	<p>座標は半角文字(12x6)単位のキャラクタ位置。</p> <p>この関数で内部描画バッファに文字列を展開したあと glcdFlipBackgroundBuf をコールすることで実際の液晶パネルへの表示がおこなわれます。</p>	

関数名	fontIsKanji	
機能	文字コードが漢字かどうかの判定	
引数	タイプ	内容
pCode	BYTE*	文字コード第一バイトへのポインタ
戻り値	BOOL	TRUE: 漢字、FALSE: 非漢字
備考	第一バイトが 81~9F、第二バイトが 40~FF なら漢字	

4.5.6 ps2.o (ヘッダーファイル: ps2.h)

概要: PS2 キーボード入力モジュールです。

関数名	ps2Init	
機能	P S 2 モジュールの初期化	
引数	タイプ	内容
なし		
戻り値		なし
備考		

関数名	ps2IsKbdReady	
機能	キーボードの状態に変化があったか調べる	
引数	タイプ	内容
なし		
戻り値	BOOL	TRUE: 変化あり、FALSE: 変化なし
備考	キー ON, OFF 時両方でコードが送られてくる	

関数名	ps2GetKeyCode	
機能	キーON, OFF 時のスキャンコードを取得	
引数	タイプ	内容
なし		
戻り値	BYTE	キースキャンコード
備考	キースキャンコードは ASCII や JIS コードと関係ない キー独自のコードで ON/OFF 時でも異なります。	

4.5.7 rslib.o (ヘッダーファイル: rslib.h)

概要: UART 通信モジュールです。

関数名	rsInit	
機能	UART の初期化	
引数	タイプ	内容
なし		
戻り値		なし
備考	使用前に一度だけコールしてください。 初期値は 38400baud 8N1 無手順 Rx 割り込み ON	

関数名	rsPutCh	
機能	1 文字の出力	
引数	タイプ	内容
code	char	文字コード
戻り値		なし
備考		

関数名	rsPuts	
機能	文字列の出力	
引数	タイプ	内容
pStr	char*	送出する文字列(0 ターミネート)
戻り値		なし
備考		

関数名	rsGetCh	
機能	ポーリングによる1文字取得	
引数	タイプ	内容
なし		
戻り値	char	受信文字
備考	これを使う場合はRx 割り込みを禁止しておくこと	

関数名	rsResetBuf	
機能	受信バッファの初期化	
引数	タイプ	内容
なし		
戻り値		なし
備考		

関数名	rsIsLineEnd	
機能	CR (リターンコード) を受信したか調べる	
引数	タイプ	内容
戻り値	BOOL	TRUE: 受信済み、FALSE: 未受信
備考		

関数名	rsGetLine	
機能	CR までの 1 行を受信バッファからコピー	
引数	タイプ	内 容
pBuf	char*	コピー先のバッファ
nLength	int	コピー先バッファのバイト数
戻り値		
備考	CR までのデータをコピーして 0 ターミネート (CR はコピーしない)	

4.5.8 xtimer.o (ヘッダーファイル: xtimer.h)

概要: タイマーモジュールです。

関数名	timInit	
機能	Timer1 をオープン	
引数	タイプ	内容
なし		
戻り値		なし
備考		

関数名	timWaitMsec	
機能	指定ミリ秒待つ	
引数	タイプ	内容
msec	int	ディレイした時間 (ミリ秒数)
戻り値		なし
備考		

4.5.9 util.o (ヘッダーファイル: util.h)

概要: 汎用ユーティリティモジュールです。

printf などのコード短縮版の内容を含みます。

関数名	utFormatHex2	
機能	1 バイト(8bit)の数値を 2 桁ヘキサ文字列に変換	
引数	タイプ	内容
pBuf	char*	文字列を受け取るバッファ
n	BYTE	数値
戻り値		なし
備考		

関数名	utFormatHex4	
機能	1 ワード(16bit)の数値を 4 桁ヘキサ文字列に変換	
引数	タイプ	内容
pBuf	char*	文字列を受け取るバッファ
n	WORD	数値
戻り値		なし
備考		

関数名	utFormatHex8	
機能	1 ダブルワード(32bit)の数値を 8 桁ヘキサ文字列に変換	
引数	タイプ	内容
pBuf	char*	文字列を受け取るバッファ
n	DWORD	数値
戻り値		なし
備考		

関数名	utValHex2	
機能	2桁ヘキサ文字列をバイト(8bit)値に変換	
引数	タイプ	内容
pStr	char*	文字列 (0 ターミネート)
戻り値	BYTE	変換値
備考		

関数名	utValHex4	
機能	4桁ヘキサ文字列をワード(16bit)値に変換	
引数	タイプ	内容
pStr	char*	文字列 (0 ターミネート)
戻り値	WORD	変換値
備考		

関数名	utValHex8	
機能	8桁ヘキサ文字列をダブルワード(32bit)値に変換	
引数	タイプ	内容
pStr	char*	文字列 (0 ターミネート)
戻り値	DWORD	変換値
備考		

関数名	utValInt	
機能	符号付き数字文字列を値に変換	
引数	タイプ	内容
pStr	char*	文字列 (0 ターミネート)
戻り値	int	変換値
備考	atol, strtol 相当	

関数名	utToUpperString	
機能	文字列を大文字に変換	
引数	タイプ	内容
pStr	char*	文字列 (0 ターミネート)
戻り値		なし
備考		

関数名	utToLowerString	
機能	文字列を小文字に変換	
引数	タイプ	内容
pStr	char*	文字列 (0 ターミネート)
戻り値		なし
備考		

関数名	utMemoryDump	
機能	メモリーダンプ	
引数	タイプ	内容
pAddr	BYTE*	ダンプする先頭のメモリーアドレス
nLength	int	ダンプするバイト数
戻り値		
備考	ターミナル(UART2)に出力	

関数名	utStringCompareNoCase	
機能	大文字、小文字を無視して文字列の一致を検出する	
引数	タイプ	内容
pStr1	char*	第一文字列 (0 ターミネート)
pStr2	char*	第二文字列 (0 ターミネート)
戻り値	int	0: 一致 -1: 不一致
備考		

関数名	utPathSplitRight	
機能	デリミタ (一番右の) の左右で文字列を分割する	
引数	タイプ	内容
pStr1	char*	元の文字列 (0 ターミネート)
cDelim	char	デリミタ文字
pStr2	char*	分割後の右文字列を受け取るバッファ
戻り値	int	分割位置、 < 0 なら delimiter not found
備考	分割後の左文字列が pStr1 に入る	

関数名	utPutString	
機能	文字列をターミナル(UART2) に出力する	
引数	タイプ	内容
pStr	char*	文字列 (0 ターミネート)
戻り値		
備考		

4.6 ライブラリの仕様と制約事項

4.6.1 ファイルシステムの仕様と制約について

dosfile.o で実装されるファイルシステムはバージョンによって使用できるファイル名ディレクトリの最大ファイル数などに制約事項があります

またこれらの制約のうち改良可能な項目につきましてはバージョンアップまたはバグフィックスのたびに当社ホームページのダウンロード・コーナーにアップしていく予定ですので最新版をダウンロードしてお使いいただきますようお願いいたします。
Ver1.0 におきましては下記の制約があります。

■ FAT 32 の仕様

ファイルサイズはカードの容量が許す限り無制限です。

FAT16, FAT12 には対応しませんのでメディアはPCのエクスペローラまたはフォーマットツールにて FAT32 でフォーマットしてからお使いください。

■ 最大63文字のロングファイルネームに対応しておりますがファイル名やディレクトリ名に漢字は使えません。

■ 任意の深さのディレクトリ階層に対応していますが、カレントディレクトリはマルチタスクに対応しません。

複数ファイルを同時に扱うことはできますがカレントディレクトリは最後におこなった `dosOpenDirectory` のディレクトリになります。

マルチタスク下では他のタスクがディレクトリ階層を移動すると全体が影響を受けますので、複数のタスクがファイルを使う場合はセマフォなどで排他制御をした上で `dosOpenDirectory(NULL)` でルートに戻ってから必要なディレクトリに移動してファイル操作をおこなってください。

■ 1ディレクトリの最大エンタリー数

Ver1.0 では1つのディレクトリ当たりのディレクトリエンタリーは最大128です。ロングファイルネームを使うと1つのファイルで複数エンタリーを消費しますのでご注意ください。

■ ファイルの読み込みでは任意のバイト位置にシークすることができますが書き出し時には Seek は無効になり、常に最後に Write をおこなった次のバイト位置がカレントポジションになります。

dosReadFile, dosWriteFile で読み書きするサイズ (バイト数は) 1 バイト単位で任意です。内部でブロッキング/デブロッキングをおこなって適切なセクターにマージされます。

- 新しいディレクトリの作成は dosCreateDirectory で可能ですが、ディレクトリを削除する機能は Ver1.0 では持たせておりません。
パソコンで削除してください。
- ファイルの日付はカレンダークロックを持っていないため Ver1.0 では 2009 年 1 月 1 日 0 時 0 分 固定になります。
- MMC/SD カードのホットプラグ (電源を入れたままでの抜き差し) はサポートしません。
基本的にカードは挿入状態で電源を入れてください。
カード挿入でリセットがかかりますのでご注意ください。

4.6.2 漢字利用の制約について

2008 年 11 月時点で MPLAB IDE は日本語に完全対応しておりませんので、ソースにリテラルに漢字を埋め込んだ場合、文字コードによっては希に文字化けが起きます。(たとえば “申” = S-JIS 0x905C など第 2 バイトが 0x5c の文字) これは 第二バイトの 0x5c が制御文字と見なされるのが原因と思われます。RS232C から文字コードをもらって漢字を表示するターミナルソフトなどのアプリケーションでは問題ありませんが、プログラム中に漢字を組み込んだ場合、もし文字化けが起きるようでしたら、その文字だけを下記の Shift JIS コードを引数に全角 1 文字を表示する関数を使って描画してください。

```
void fontPutKanji12ByDot( WORD wCode, short xPos, short yPos )
```

また MPLAB ではフォルダー名やファイル名にも全角文字を使わない方が良いでしょう。

4.6.3 PS2 ライブラリの制約について

本システムの PS2 コネクタはキーボード入力専用です。 双方向通信の必要なマウスなどには対応できません。

PS2 キーボードから送られてくるコードは一般的なキャラクターコードとは異なり

またキーの ON, OFF 時で違いますのでインターネットなどでコード表を調べてお使いください。

4.7 ライブラリのライセンスについて

- MX360-DK 付属ライブラリは当社製品でのご使用に限定させていただきます。
ユーザー様で開発されたハードウェアなど、当社製品以外での商用でのご使用、移植は固くお断りします。
- MX360-DK 以外でも当社製品（ボード）のプログラム開発でご利用いただくことはご自由です。
- 他のシステムにてのご利用を希望される場合は別途ライセンス契約を承りますので当社ホームページ <http://www.cyber-melon.com> 「お問い合わせ」をクリックしてご連絡ください。

5. その他

5.1 電気的特性

項目	値	条件
入力 Low 電圧	0.66V max	0.2 Vdd
入力 Hi 電圧	3.17V min	0.8Vdd (5V トレラント)
出力 Low 電圧	0.4V max	IOL = 7mA@3.6V
出力 Hi 電圧	2.4V min	IOH = -12.0mA@3.6V
絶対最大シンク電流	25mA	
絶対最大ソース電流	25mA	

MX360 ボードはコントローラに PIC32MX360F を使用しております。

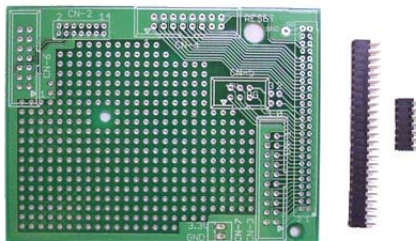
詳しくは付属 CD-ROM の Microchip フォルダにあるデータシート (英語版)

PIC32MX3xx.pdf を参照してください。

5.2 オプション製品

拡張コネクタを引き出して 2.54mm ピッチに変換するユニバーサル基板&コネクタキット **MXUV1** をオプションで用意しておりますので、各種回路プロトタイプを試作などにご利用ください。

当社ホームページの MX360 のコーナーよりご購入いただけます。



5.3 特注仕様

- ・数量によりましては MX360 プリント基板のみのパーツ販売も致します。
- ・本ボードを利用した特注仕様ソフトウェアの開発請負も可能です。

詳細は当社ホームページ <http://www.cyber-melon.com> お問い合わせコーナーよりご連絡ください。

本書の改訂版は当社ホームページの該当製品コーナーよりダウンロードしてください。

Cyber MELON

株式会社インターネット

〒665-0841

兵庫県宝塚市御殿山 2-25-39

<http://www.cyber-melon.com>

e-mail: info@cyber-melon.com

(# を @ に置き換えてください)