# EMUZ80技術資料

EMUZ80はZ80とPIC18F47Q43で動く2チップ構成のコンピュータです。最少の安価な部品で完成し、 BASICが動きます。プログラムは簡単に書き換えられ、即座に動作を解析することができます。



# 目次

EMUZ80の概要 3
部品表 4
回路図 5
ソルダバッドの処理 6
ACアダプタ 7
プログラムの書き込み 8
●書き込みの準備 8
●標準プログラムの書き込み 9
●標準プログラムの書き込み 9
●標準プログラムの実験と解析 10
アドレスマップ 11
USB-シリアル変換ケーブル/アダプタ 12
端末ソフトの設定 13
起動の確認 14
別途配布物 1 15



Tools Window Help

• 👸 • 1	*	<u> </u>	• 🖏 🌃	• 🚾 PC:	0x0 n ov	zdcc ∶W:0x	:0 : bank 0	¥	How do I? Keywo	rd(s)
	🖭 mair	nc ×								
	Source	History	🔝 🔛 📓	- 🔊 - 🔍	₹ <b>- - - -</b>	l 📭   🔗 🛛	🌜 🐁 🔤	u 🖭 🥚 🗆	Mar 📑 📴	
	93	0xF	E,OxOA,	- 77	CP	OAH				
	94	0x2	3,0x03,	11	JR	Z,STOP				
	95	0x2	3,	11	INC	HL				
	96	0x1	B,OxEE,	11	JR	PUTCH				
	97	0x1	3,OxFE,	// STOP	: JR	STOP				
	98	0x4	3,0x45,0x40	),Ox4C,Ox4F	,0x2C,//	DB '	HELLO, W	ORLD',ODH,OA	Н	
	99	0x2	),Ox57,Ox4P	,0x52,0x4C	,0x44,			Abocookaa Taa	- T \/T	
	100	0x0	),OxOA,					13:9600ps - Ter	a ierm vi	
	101	- */					ファイル(F)	編集(E) 設知	を(S) コントロール(O)	ウィンドウ(W)
	102	- 11	CALL/RET t	est			ADRS:00	)00,RD:0,WR	:1,READ:0031	
	103	0x	31,0x00,0x9	30,//	LD SP,	STACK	ADRS:00	)01,RD:0,WR	:1,READ:0000	
	104	0x	CD,OxOB,OxO	00,// LOOP:	CALL SUB	1	ADRS:00	02,RD:0,WR	:1,READ:0090	
	105	0x	CD,OxOC,OxO	)0,//	CALL SUB	2	ADRS:00	03,RD:0,WR	:1,READ:00CD	
	106	0x	18,0xF8,	11	JR LOO	P	ADRS:00	)04,RD:0,WR	:1,READ:000B	
	107	0x	09,	// SUB1:	RET		ADRS:00	05,RD:0,WR	:1,READ:0000	
	108	0x	09,	// SUB2:	RET		ADRS:8F	FF,RD:1,WR	:0,SAVE:0000	
	100						ADD0.0D	CE DD.1 MD	0. SVNE-0008	

## EMUZ80の概要

EMUZ80は部品代1000円で完成するZ80のコンピュータです。ROM、RAM、シリアル (UART) はPIC18F47Q43でエ ミュレートします。標準プログラムだと、Z80は2.5MHz/最悪7ウェイトで動き、BASICを実行します。標準プログラム はいっそうの高速化を目指して改良する余地があります。ウデに自信のある人はぜひ挑戦してください。

●本体の部品─部品表にしたがってご自身で揃え、プリント基板の部品番号が一致する位置に取り付けてください。
 ❷Snap接続端子─PIC18F47Q43にプログラムを書き込むときSnapを取り付けます。

● DCジャック―電圧5V、電流500mA以上、内径2.1 mm、センタープラスのACアダプタを接続してください。
 ●シリアル端子―TTL-232R-5V または同等のUSB-シリアル変換ケーブル/アダプタでパソコンと接続してください。



③DCジャック ④シリアル端子

部品表

本体の部品は下に示す部品表にしたがって揃えてください。部品表の部品番号とプリント基板の部品番号を照合し、 所定の位置に取り付けると組み立て完了です。部品表には実験や解析の作業を想定した部品が含まれます。そうした 部品は、ご利用のジャンパー線などに合わせて変更し、あるいは省略することを検討してください。

部品番号	型番	数量	仕様	販売店
IC1	Z80 CPU	1	マイクロプロセッサ	オレンジピコ、若松通商
IC2	PIC18F47Q43	1	マイクロコントローラ	オレンジピコ、秋月電子通商
D1	1N4148	1	小信号スイッチングダイオード	オレンジピコ、秋月電子通商、若松通商
LED1	OSRRH23133A	1	$\varphi$ 3mm一般LED各色	オレンジピコ、秋月電子通商
R1、R2	10kΩ (1/4W)	2	カーボン抵抗/金属皮膜抵抗	オレンジピコ、秋月電子通商
R3	1.5k Ω (1/4W)	1	カーボン抵抗 <sup>[注1]</sup>	オレンジピコ、秋月電子通商
C1 ~ C3	0.1 μ F (50V)	3	積層セラミックコンデンサ	オレンジピコ、秋月電子通商
C4	10 μ F (16V)	1	電解/タンタルコンデンサ	オレンジピコ、秋月電子通商
CON1、CON2	PH-1x40RG(2)	1	1列L型ピンヘッダ <sup>[注2]</sup>	オレンジピコ、秋月電子通商
CON3	MJ-179PH	1	2.1mm φ標準DCジャック	オレンジピコ、秋月電子通商
S1	SS-12D00-G5	1	スライドスイッチ	オレンジピコ、秋月電子通商
S2	DTS-6-V	1	小型タクトスイッチ	オレンジピコ、秋月電子通商
_	FH-1x20	4	20ピン1列ピンソケット <sup>[注3]</sup>	オレンジピコ、秋月電子通商
	2227-40-06	2	40 ピンICソケット 600mil	オレンジピコ、秋月電子通商

[注1] R3の抵抗値は使用するLEDの輝度に応じて調整してください。

[注2] CON1は8ピン、CON2は6ピンのところで折って使います。

[注3] ご利用のジャンパー線によってはピンヘッダを取り付けてください。実験や解析をしない場合は不要です。

[通販サイト]

秋月電子通商—http://akizukidenshi.com/

オレンジピコーhttps://store.shopping.yahoo.co.jp/orangepicoshop/

若松通商—http://wakamatsu.co.jp/biz/

※2022年3月15日時点の情報です。

回路図

回路図を下に示します。部品番号は部品表およびプリント基板のシルク印刷と一致しています。Z80とPIC18F47Q43 の両端のピンソケットは、ここに描いていませんが、各端子が隣のピンと接続し、シルク印刷した信号を取り扱います。 重要な処理の多くをPIC18F47Q43のプログラムが実行するため、標準プログラムのソースとあわせてご覧ください。



CON1-Snap 接続端子、CON2-シリアル端子 J1 (クローズ)-NMI-VCC、J2 (クローズ)-BUSREQ-VCC、J3 (オープン)-BUSPOWER

## ソルダバッドの処理

プリント基板のハンダ面に3か所のソルダバッドがあります。これらは原則として何もしないでください。J1とJ2は 不使用端子を無効に固定しています。J3は電源をUSBのバスパワーからとるときクローズしますが、その場合、多く のUSB-シリアル変換アダプタは電源スイッチを入れた瞬間にリセットします(TTL-232R-5Vは正常に動作します)。



J3 (オープン) — BUSPOWER

# ACアダプタ

電源は原則としてACアダプタからとります。電圧5V、電流500mA以上、内径2.1 mm、センタープラスのACアダプタ をDCジャックに接続してください。粗悪な製品は通電時に一瞬、電圧が5Vを超えて回路を壊す恐れがありますから、 信頼のおける製品を使ってください。EMUZ80は秋月電子通商で販売しているGF12-US0520で動作確認しています。





# プログラムの書き込み

PIC18F47Q43に標準プログラムを書き込むとZ80がBASICを実行します。また、標準プログラムを改編して高速化を 図ったり、配列rom[]に独自のプログラムを置いてZ80の動作を検証したりすることができます。これらの作業のもっ とも簡単な手順を以降に示します。

#### ●書き込みの準備

PIC18F47Q43の書き込みにはSnapを使います。次のとおりパソコン→Snap→EMUZ80を接続してください。 ●EMUZ80を組み立ててPIC18F47Q43を取り付けます。Z80が取り付けてあっても大丈夫です。 ●CON1にSnapを取り付けます。Snapとプリント基板は▽の位置を一致させてください。 ●SnapのケーブルをパソコンのUSBポートに接続します。 ④EMUZ80の電源を入れます。



※ PICkit で書き込む場合は CON1の▽側6ピンを使います。 ただし、 PICkit では書き込みテストをしていません。

#### ●標準プログラムの書き込み

標準プログラムはコンパイル済み HEX ファイル emuz80\_pic.hex で提供します。 HEX ファイルの書き込みには MPLAB X IPEを使い、次のように操作します。 なお、 MPLAB X IPE は MPLAB X IDE とともにインストールされます。

😰 MPLAB IPE v6.00	)	– 🗆 X
File Settings View	Tools Window Help	
Op Advanced	Mode	
D Hold in Re	cat	Results
Release from the second s	om Reset	
PM3 Com	nunication	Checksum: N/A 62
Device:	PIC18F47Q43 • 2 Apply 3	Pass Count: 18
Test		Fail Count: 12
1001:	Snap S.No : BUR183076188 • 4 Connect 5	Total Count: 30
Prog	am 📴 Erase 🖓 Read	Verify Blank Check
① [Settings] → [	Release from Reset] を選択	
2 PIC18F47Q43 7	を選択	
€ [Apply] をクリ	ック	
● Snapを選択		
<b>じ</b> [Connect] をク	<b>リック</b>	
Device:	PIC18F47Q43	Pass Count: 18
Test		Fail Count: 12
1001:	Snap S.No : BUR183076188	Total Count: 30
Prog	am 🕢 🖵 Frase	Verify Blank Check
		, only and the second second
Hex File: C:\\	Jsers\Tetsuya\Downloads\emuz80-main\emuz80-main\emuz80_pic.hex	Browse Clear selection
SQTP File: Clic	k on browse to select a SOTP file	Browse Clear selection
Output - IPE ×		
2022-03-03 10:23	m C:#Users≇letsuya≇Downloads≇emuz8U-main≇emuz8U-main≇emuz8U_pic.hex ?:51 +0900 - Hex file loaded successfully.	<u>^</u>
		~
	Tool: Snap S.No : BUR188076188 Device: PIC18F47Q43 Environment	NA PIC18F-Q_DFP 1.13.211 Tool Pack Version:Latest

⑥[Brows]をクリックしてemuz80\_pic.hexを選択

● [Program] をクリック

#### ◉標準プログラムの改編

MPLAB X IDE でプロジェクト emuz80.Xを開くと標準プログラムを改編することができます。標準プログラムはすべ ての手順を main.c に記述していますから、必要に応じ、これを書き換えます。[Make and Program Devices] をクリッ クすると書き換えた main.c をビルドして書き込みます。

X MPLAB X IDE v6.00 - emuz80 : default	
File Edit View Navigate Source Refactor	Production Debug Team Tools Window Help
👚 🞦 🞴 🖣 🦻 🦿 🛛 defau	h 🔍 🕼 * 🕆 * 🖄 * 🕨 * 🙀 * 🖓 * 🖉 * 🖓 🖬 * 🕲 🚾 🥮 🛒   1
Projects × Files Services	main.c ×     Make and Program Device (Project emuz80)
⊡	Source History 👕 📴 🐻 - 🐻 - 🔍 - 🔍 🖓 🖓 🖓 🖓 🎧 斗 🖓 😓 😒
⊕ 🕞 Important Files ⊕ 🕞 Linker Files	65 66 #define Z80_CLK 2500000UL // Z80 clock frequency(Max 16MHz)
Source Files	67 68 #define ROM_SIZE 0x4000 // ROM size 16K bytes
Libraries	69 #define RAM_SIZE 0x1000 // RAM size 4K bytes 70 #define RAM_TOP 0x8000 // RAM start address
'±™' [[] Loadables	71 #define UART_DREG 0xE000 // UART data register address 72 #define UART_CREG_0xE001 // UART_control_register address
	<ul> <li>extern const unsigned char rom[]; // Equivalent to ROM, see end of this fil</li> <li>unsigned char ram[RAM_SIZE]; // Equivalent to RAM</li> </ul>
	78
	80 struct {
	82 unsigned char h; // Address high 8 bits

#### ●独自のプログラムの実験と解析

Z80は配列rom[]に記述したプログラムを実行します。ここへ独自のプログラムを書くことで実験や解析ができます。

X MPLAB X IDE v6.00 - emuz80 : default		
<u>File Edit View Navigate Source Refactor</u>	Production <u>D</u> ebu	ug Tea <u>m T</u> ools <u>W</u> indow <u>H</u> elp
👚 🔚 🖶 🍯 🍼 🛛 defau	lt .	- 監・背・豫・ト・聖・聖・筍 豚・⑩ 🚥 🧇 🛒 !
Projects × Files Services  emuz 80 emuz 80 file Header Files files files	Source Histor	ry 🖀 📴 🖙 🐺 -  ସ୍ 😎 🖓 🕾 🚍 斗   🔗 😓   🗐 🗐   👄 🗉   🕮 🚅 🖡 st unsigned char rom[ROM SIZE] = {
	289	// HELLO
main.c	290	0x31, 0x00, 0x90, 0x21, 0x31, 0x00, 0x7e, 0xfe,
🕀 庙 Libraries	291	0x00, 0x28, 0x06, 0xcd, 0x19, 0x00, 0x23, 0x18,
🗄 🚋 Loadables	292	Oxf5, Oxcd, Ox26, OxOO, Oxcd, Ox19, OxOO, Ox18,
	293	Oxf8, Oxf5, Ox3a, OxO1, OxeO, Oxcb, Ox4f, Ox28,
	294	Oxf9, Oxf1, Ox32, OxOO, OxeO, Oxc9, Ox3a, OxO1,
	295	OxeO, Oxcb, Ox47, Ox28, Oxf9, Ox3a, OxOO, OxeO,
	296	0xc9, 0x48, 0x45, 0x4c, 0x4c, 0x4f, 0x2c, 0x20,
	297	0x57, 0x4f, 0x52, 0x4c, 0x44, 0x0d, 0x0a, 0x00,
	298	
	299	// EMUBASIC
	300	Oxf3, Ox31, Oxed, Ox80, Oxc3, Ox41, OxOO, Oxff,
	301	Oxc3, Ox34, OxOO, Oxff, Oxff, Oxff, Oxff, Oxff,
	302	Oxc3, Ox1b, OxOO, Oxff, Oxff, Oxff, Oxff, Oxff,
	303	Oxc3, Ox2e, OxOO, Ox3a, OxO1, OxeO, Oxcb, Ox47,
11	904	NY98 NYEG NY85 NY90 NY50 NYEE NYEE NYEE

## アドレスマップ

標準プログラムのアドレスマップを下に示します。メモリアドレスのみが存在し、IOアドレスはありません。ROM領 域は先頭が0000H、サイズは16Kバイトです。RAM領域は先頭が8000H、サイズは4Kバイトです。UARTはE000Hと E001Hに割り当ててあり、Z80はこのアドレスでPIC18F47Q43のレジスタを読み書きすることになります。



# USB-シリアル変換ケーブル/アダプタ

EMUZ80はUSB-シリアル変換ケーブル/アダプタでパソコンと接続し、端末ソフトで操作します。USB-シリアル変換 ケーブルはFTDIのTL-232R-5Vを想定しています。ほかに、Arduino Pro Mini 5V用のUSB-シリアル変換アダプタが使 えます。EMUZ80はFTDIのTTL-232R-5VとsparkfunのCH340G (バスパワーには非対応)で動作確認しています。



シリアル端子にはEMUZ80側の信号名が印刷されています。これとUSB-シリアル変換ケーブル/アダプタの送受信信 号がたすき掛けになるように接続します。すなわち、TXD 之RXD、5V 之VCC、GND 之GND となるのが正常です。なお、 信号電圧 3.3V/5V 対応のUSB-シリアル変換アダプタは、信号電圧をあらかじめ5V に設定しておいてください。

	RES CON2	SET .		RESE CON2
	NC	RE3 @	• untyreds	
	TXD	RAO 📺		
	RXD	RA1		
	5V	RA2		5V
	NC NC	RA3 🕹		
	GND	RA1 9	DISER TELIAS	
	μ Δ D1	RA5		L & D1
		1 RE0		
TTL-232R-5Vの接続例		RE1	CH340Gの接続例	

## 端末ソフトの設定

端末ソフトの通信方式は非同期シリアル、通信速度は9600bps、通信形式はデータ長8ビット、パリティなし、ストッ プビット1です。また、BASICのプログラムのアップロードに備えて300m秒/行の遅延設定をしておくといいでしょう。 端末ソフトがTeraTermの場合、[設定]→[シリアルポート]と選択して下に示すとおり設定します。

🔟 COM	6:9600bps	- Tera Terr	n VT				_	Х
ファイル(F)	編集(E)	設定(S)	コントロール(O)	ウィンドウ(W)	ヘルプ(H)			 
ファイ)ル(F)	[編集(E)	設定(S) 端 フ イ フ ポ キー シリ ブロ SSF SSF SSF SSF TCF	コントロール(O) た(T) ンドウ(W) ント(F) ボード(K) アルポート(E) キシ(P) れ 記証(A) 封転送(O) 封鍵生成(N)	יאר איז	∧Jµ7(H)			^
		全 紀 その 設 灯 設 灯	&(G) (他の設定(D) をの保存(S) をの読み込み(R). をフォルダ(I) マップ読み込み(L					
								~
Tera Term:	シリアルポー	ト設定と接	<del>と</del> 続			×		
ボー	-ト(P):		COM6	~	祖左の接結を再設守の	(NI)		

示ート(P):	COM6	$\sim$	現在の接続を再設定(N)
スピ <i>ー</i> ド(E):	9600	~	
データ(D):	8 bit	$\sim$	キャンセル
バリティ(A):	none	$\sim$	
ストップビット(s):	1 bit	$\sim$	ヘルプ(H)
フロー制御(F):	none	~	
□ 达1言)差处	E		
0	= ] ミリ秒/字(g	<u>c</u> ) <u>300</u>	ミリ秒/行(_)
0 Device Friendly Nat Device Instance ID: Device Manufactur Provider Name: wcl Driver Date: 8–8–20 Driver Version: 3.4.2	= ] ミリ秒/字(( me: USB-SE : USB¥VID_1: er: wch.cn h.cn 014 2014.8	<u>)</u> 300 RIAL CH34 A86&PID_7	】ミリ科シバテ( <u>し</u> ) 40 (COM6) 523¥6&7563C70&0&4

## 起動の確認

EMUZ80をパソコンと接続し、端末ソフトを開き、電源を入れると、BASICの起動メッセージが表示されます。この 時点で、部品の取り付けかたに間違いがなく、プログラムのUARTとROMも正しく動作していることがわかります。 また、空き容量が3710バイトと表示されていれば、RAMも正しく動作しています。これでEMUZ80は完成です。



BASICがASCIIART.BASの実行を完了する時間はPIC18F47Q43のプログラムの速度を測る目安になります。標準プログ ラムは23分23秒です。PIC18F47Q43のプログラムの改良を試み、BASICがこれより早くASCIIART.BASの実行を完了 したら、それは成功したといえます。なお、各種コンピュータの実測値がこちらのサイトにまとめられています。

ASCIIART(マンデルブロ集合)ベンチマーク—http://haserin09.la.coocan.jp/asciiart.html

### 別途配布物一覧

Githubのリポジトリで関連のファイルを提供しています。Githubおよびgitに不案内な人は次のURLを開き、[Code] ボタンから [Downloard ZIP] を選択してダウンロードするのが簡単です。

●Githubリポジトリ―https://github.com/vintagechips/emuz80

Oアイルリスト

README.md — EMUZ80 のごく簡単な説明

emuz80\_pic.hex―標準プログラム→PIC18F47Q43に書き込んでください

emuz80\_techdata.pdf―この文書

ASCIIART.BAS—BASICで実行可能なマンデルブロ集合プログラム(よくベンチマークで使われます)

emuz80.X フォルダ―標準プログラムの MPLAB X IDE プロジェクト

examplesフォルダ—EMUZ80で実行可能なZ80コード

→\*.txtの記述を配列 rom[]の先頭に挿入し、再ビルドし、PIC18F47Q43 に書き込んでください

・HELLOフォルダ—HELLO, WORLDとエコーバックのテスト用

・CALLRET フォルダ—ROM、RAM、UARTのテスト用→動作はロジックアナライザなどで観測してください

・EMUBASICフォルダ―標準プログラムの配列 rom[]に書き込まれている BASIC

Eagle416フォルダ—Eagle v4.16が生成したファイル

※標準プログラムより高速なプログラムがここへ追加される場合があります。

EMUZ80技術資料 2022年3月15日初版発行 著者—鈴木哲哉 Copyright © 2022 Tetsuya Suzuki CC BY-NC-SA 3.0

