

MPC-RAS64DIO-I

取扱説明書



EMBEDDED TECHNOLOGY
Corporation

はじめに

1. 製品の保証について

・無償修理

製品ご購入後1年間は無償で修理いたします。
(但し、下記「有償修理」に該当するものを除く)

・有償修理

- 1) 製品ご購入後1年を経過したもの。
- 2) 製品購入1年以内で故障の原因がお客様の取り扱い上のミスによるもの。
- 3) 製品購入1年以内で故障の原因がお客様の故意によるもの。

・免責事項

当社製品の故障、不具合、誤動作あるいは停電によって生じた損害等の纯粹経済損失につきましては、当社は一切その責任を負いかねますので、あらかじめご了承ください。

2. 製品について

- ・当社製品はカタログ仕様範囲内において、使用部品、回路図等、予告無く変更することがあります。
- ・当社製品は部品メーカーの製造中止等によりやむを得ず製品の供給を続けることが出来なくなることがあります。
- ・当社製品の無断での複製を禁止します。
- ・当社製品は一般商工業用として設計されており生命、財産に関わるような状況下で使用されることを意図して設計、製造されたものではありません。本製品の故障、誤動作が人命を脅かしたり、人体に危害を与えたりする恐れのある用途（生命維持、監視のための医療用）、および高い信頼性が要求される用途（航空・宇宙用、運輸用、海底中継器、原子力制御用、走行制御用、移動体用）にはご利用されないようご注意ください。すべての電子機器はある確率で故障が発生します。当社製品の故障により、人畜や財産が被害を受けたり、火災事故や社会的損害が生じたりしないように安全設計をお願いします。また長時間連続運転や仕様外の環境でのご使用は避けてください。但し、長時間運転でご使用された場合の故障につきましては通常どおりの修理保証（1年以内無償、1年以上有償）が受けられます。

3. カタログ、取扱説明書の記載事項について

- ・当社製品のカタログ及び取扱説明書は予告無く変更する場合があります。
- ・取扱説明書に記載されている内容及び回路図の一部又は全部を無断での転載、転用を禁止します。
- ・本資料に記載された情報、回路図は機器の応用例であり動作、性能を保証するものではなく、実際の機器への搭載を目的としたものではありません。またこれらの情報、回路を使用することにより起因する第三者の工業所有権、知的所有権、その他権利侵害に関わる問題が生じた際、当社はその責を負いませんのであらかじめご了承ください。

4. 海外への輸出について

- ・当社製品を使用した機器を海外へ持ち出される場合、当社製品のCOCOMパラメーターシートが必要です。その都度お申しつけ頂ければパラメーターシートを発行いたします。

5. 本書に記載された使用条件の範囲内でご使用願います。使用条件の範囲を超えたご使用の場合は本製品の保証は致しかねますのであしからずご了承ください。

改訂履歴

改訂日	改訂項目	ページ	改訂箇所
2016/7	初版		
2022/6/17	第2版	20	拡張ボードとラズベリーパイの連結を追記
//	//	13	9-2. I/O アドレスマップの記述を修正 I/O の表現を PORT に変更

目次

	ページ
1. 概要	5
2. 特徴	5
3. ご注文型番	5
4. 仕様	6
5. ブロック図	7
6. 基板寸法図、配置図	8
7. アドレス対応表	8
8. ピンアサイン	9
9. I2C通信データフォーマット	12
10. 入出力回路	14
11. 設定	15
12. コネクタ	16
13. I2C通信の使用方法	16
14. サンプルプログラム	18
15. 拡張ボードとラズベリーパイの連結	20

1. 概要

RAS64DIOはラズベリーパイ向けに開発した拡張入出力ボードです。
入出力信号は双方向バスで、コマンドにより port 単位で入力または出力に設定でき、
入出力合計で 64 ビットとなります。
ラズベリーパイとの通信方式は I2C です。

2. 特徴

入力部

◎ 入力電圧範囲 3.3V~5V CMOS ロジック信号

入力信号は全て+5Vプルアップ又は GND プルダウンが選択できます。
(出荷時は全 port プルアップ)

出力部

SN74HC574PWR を搭載し

出力電圧 5V、電流 20mA (短絡) を出力します。

共通部

◎ 基板サイズ

Raspberry Pi4 MODEL-B,Raspberry Pi3 MODEL-B、

Raspberry Pi 2 MODEL-B と同サイズ

◎ラズベリーパイとの接続

ラズベリーパイの GPIO 拡張コネクタと接続し、I2C 又は SPI により通信します。
ラズベリーパイの拡張コネクタに接続する際は、中間基板として MPC-RASST が
必要になる場合があります。

3. ご注文型番

MPC-RAS64DIO-I : I2Cプロトコル

MPC-RAS64DIO-S : SPIプロトコル

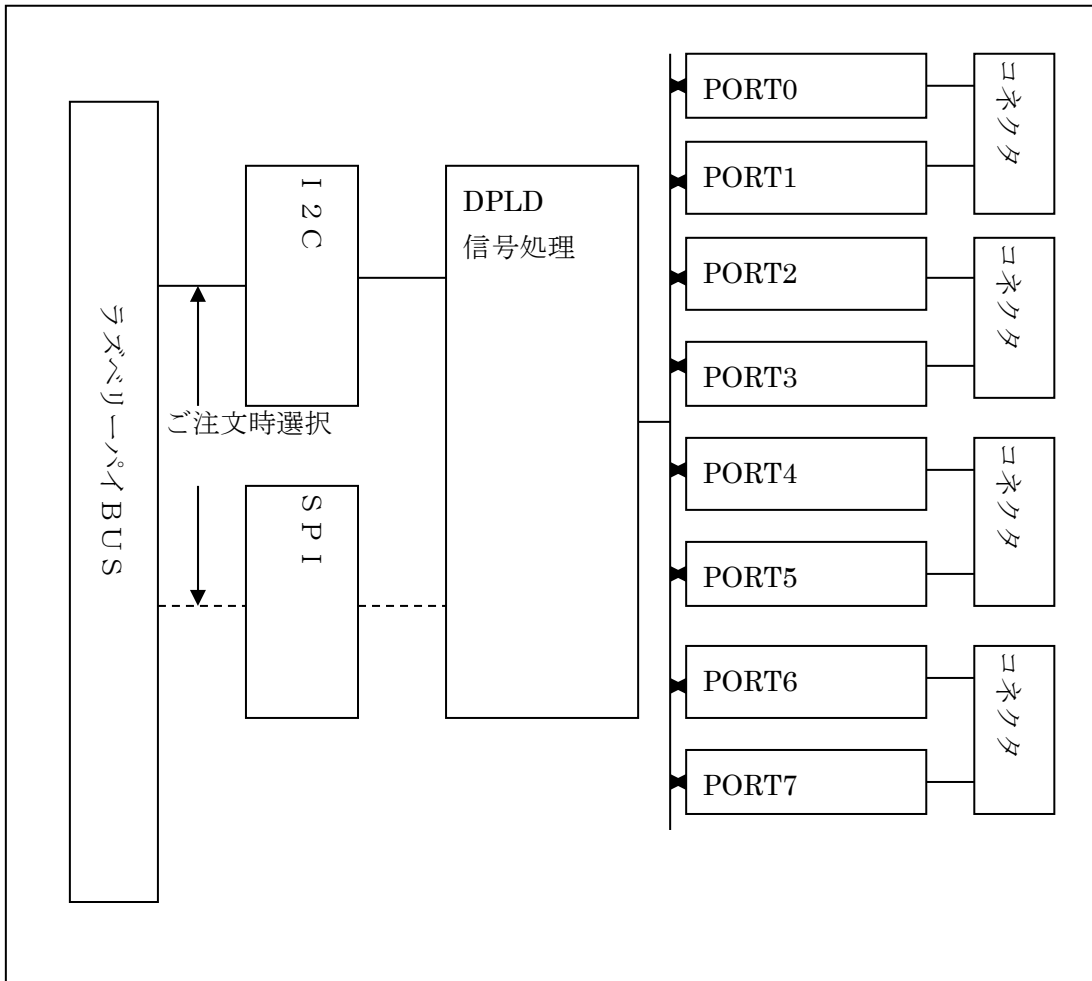
MPC-RAS-CA2030P-24

外部接続ケーブル (基板側=コネクタ、先バラ 長さ 30cm) : ¥900- (本)
電線 : UL1007 AWG24

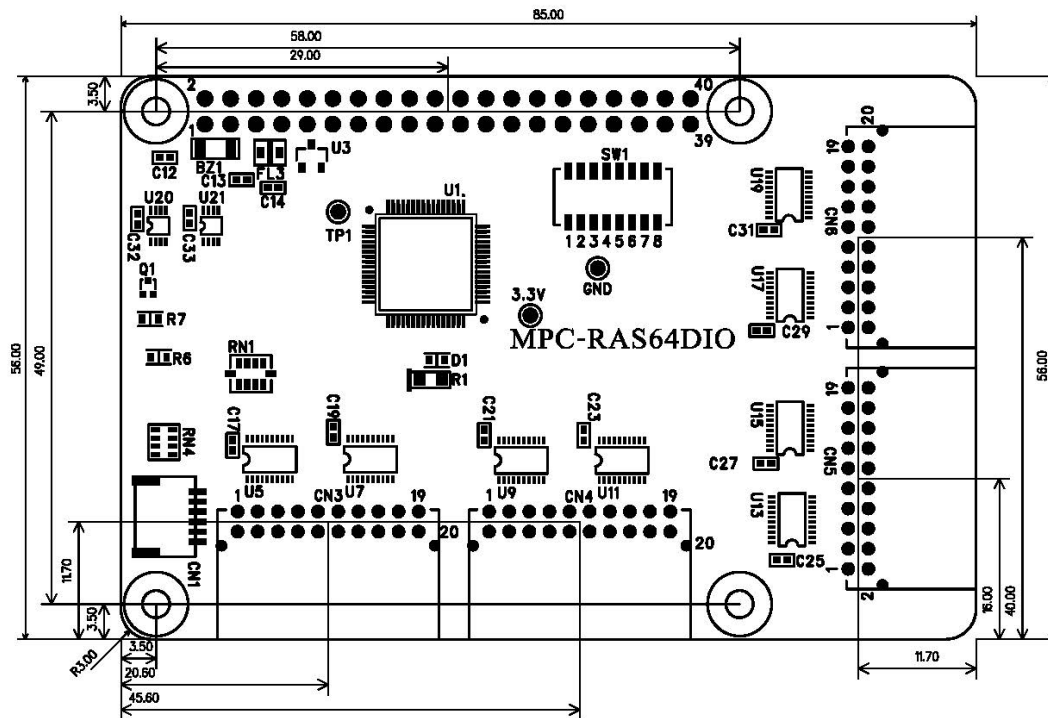
4. 仕様

項目	内容
入力点数	64 ビット (出力と共有)
出力点数	64 ビット (入力と共有)
I/O アドレス	port アドレス 0=PORT0 1=PORT1 2=PORT2 3=PORT3 4=PORT4 5=PORT5 6=PORT6 7=PORT7 8=コントロール (入出力設定) I2C: I2C アドレス(6,5bit)は DIP スイッチ により設定
入力部 IC	TC74LCX541FK
出力部 IC	SN74HC574PWR
入力信号 VIH VIL	2V~5.5V 0~0.8V
出力信号 VOH VOL	4.9V~5V (IOH=-6mA) 0~0.1V (IOL=20 μ A) ~0.33V (IOL=6mA)
電源	+5V (バスより供給)
消費電流	110mA (無負荷動作時)
外部インターフェースコネクタ (入力部、出力部共同)	基板側 S20B-PADSS-1 (JST) × 4
使用温度範囲	-20°C~60°C

5. ブロック図



6. 基板寸法図、配置図



7. アドレス対応表

番地	外部コネクタとの対応
0	CN3 2~9番 PORT0 (1=内部 VCC, 10=内部 GND)
1	CN3 12~19番 PORT1 (11=内部 VCC、20=内部 GND)
2	CN4 2~9番 PORT2 (1=内部 VCC、10=内部 GND)
3	CN4 12~19番 PORT3 (11=内部 VCC、20=内部 GND)
4	CN5 2~9番 PORT4 (1=内部 VCC, 10=内部 GND)
5	CN5 12~19番 PORT5 (11=内部 VCC、20=内部 GND)
6	CN6 2~9番 PORT6 (1=内部 VCC、10=内部 GND)
7	CN6 12~19番 PORT7 (11=内部 VCC、20=内部 GND)

8. ピンアサイン

CN3

ピン番号	信号	ピン番号	信号
1	VCC_5V	2	PORT 00
3	PORT 01	4	PORT 02
5	PORT 03	6	PORT 04
7	PORT 05	8	PORT 06
9	PORT 07	10	GND
11	VCC_5V	12	PORT 10
13	PORT 11	14	PORT 12
15	PORT 13	16	PORT 14
17	PORT 15	18	PORT 16
19	PORT 17	20	GND

CN4

ピン番号	信号	ピン番号	信号
1	VCC_5V	2	PORT 20
3	PORT 21	4	PORT 22
5	PORT 23	6	PORT 24
7	PORT 25	8	PORT 26
9	PORT 27	10	GND
11	VCC_5V	12	PORT 30
13	PORT 31	14	PORT 32
15	PORT 33	16	PORT 34
17	PORT 35	18	PORT 36
19	PORT 37	20	GND

CN5

ピン番号	信号	ピン番号	信号
1	VCC_5V	2	PORT 40
3	PORT 41	4	PORT 42
5	PORT 43	6	PORT 44
7	PORT 45	8	PORT 46
9	PORT 47	10	GND
11	VCC_5V	12	PORT 50
13	PORT 51	14	PORT 52
15	PORT 53	16	PORT 54
17	PORT 55	18	PORT 56
19	PORT 57	20	GND

CN6

ピン番号	信号	ピン番号	信号
1	VCC_5V	2	PORT 60
3	PORT 61	4	PORT 62
5	PORT 63	6	PORT 64
7	PORT 65	8	PORT 66
9	PORT 67	10	GND
11	VCC_5V	12	PORT 70
13	PORT 71	14	PORT 72
15	PORT 73	16	PORT 74
17	PORT 75	18	PORT 76
19	PORT 77	20	GND

CN2 (ラズベリーパイとの 40 ピン GPIO インターフェースコネクタ)

ピン番号	信号	ピン番号	信号
1	未使用	2	+5V
3	I2C_SDA	4	+5V
5	I2C_SCL	6	GND
7	未使用	8	未使用
9	GND	10	未使用
11	未使用	12	未使用
13	未使用	14	GND
15	未使用	16	未使用
17	未使用	18	未使用
19	SPIO_MOSI	20	GND
21	SPIO_MISO	22	未使用
23	SPIO_SCLK	24	SPIO_CE0
25	GND	26	SPIO_CE1
27	未使用	28	未使用
29	未使用	30	GND
31	未使用	32	未使用
33	未使用	34	GND
35	未使用	36	未使用
37	未使用	38	未使用
39	GND	40	未使用

9. I2C通信データフォーマット

1 バイト目

ST	7	6	5	4	3	2	1	0	ACK
st	スレーブアドレス								
	DIPSW-8	DIPSW-7	“1” 固定	A3	A2	A1	A0	R/W	ACK

2 バイト目

7	6	5	4	3	2	1	0	ACK
D7	D6	D5	D4	D3	D2	D1	D0	ACK

DIPSW-8 : 8ビット目 (オフ:”1”、オン:”0”)

DIPSW-7 : 7ビット目 (オフ:”1”、オン:”0”)

A0~A2 : port アドレスビット

A3 : コントロールビット (’1’=入出力方向設定、’0’=ポートアドレス設定)

R/W : ”0”で送信 “1”で受信 (ホストからみて)

D0~D7 : データ

※R/W ビットは、ラズベリーパイのコマンド等で自動的に設定されます。

9. 1 スレーブアドレス

当ボードはスレーブアドレスを 10port 占有します

1) 先頭アドレスの設定

I2C アドレス bit	port の入出力方向設定時	port データ入力/出力時
6	DIP-SW8、DIP-SW7	DIP-SW8、DIP-SW7
5	(下図参照)	(下図参照)
4	“1” 固定	“1” 固定
3	1 (コントロールデータ)	”0” (ポートアドレス)
2	”0”固定	port アドレス
1	”0”固定	port アドレス
0	”0”固定	port アドレス

2) DIP-SW (ディップスイッチ(SW1)) 8,7 の設定

DIP-SW ビット 8	DIP-SW ビット 7	I2C 占有アドレス
0	0	10~19 (HEX)
0	1	30~39 (HEX)
1	0	50~59 (HEX)
1	1	設定禁止

DIP-SW はオフで”1”、オンで”0”

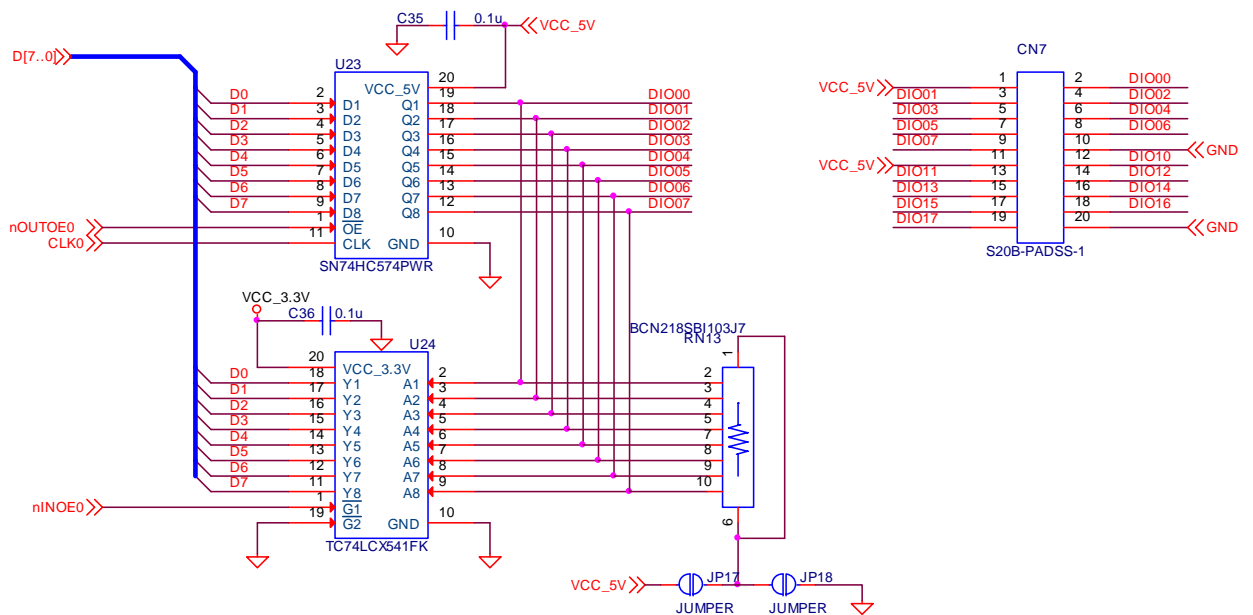
9. 2 I/O アドレスマップ

A3	A2	A1	A0	R/W	動作
0	0	0	0	0	PORT0 へのデータ書き込み
0	0	0	1	0	PORT1 へのデータ書き込み
0	0	1	0	0	PORT2 へのデータ書き込み
0	0	1	1	0	PORT3 へのデータ書き込み
0	1	0	0	0	PORT4 へのデータ書き込み
0	1	0	1	0	PORT5 へのデータ書き込み
0	1	1	0	0	PORT6 へのデータ書き込み
0	1	1	1	0	PORT7 へのデータ書き込み
0	0	0	0	1	PORT0 データの読み込み
0	0	0	1	1	PORT1 データの読み込み
0	0	1	0	1	PORT2 データの読み込み
0	0	1	1	1	PORT3 データの読み込み
0	1	0	0	1	PORT4 データの読み込み
0	1	0	1	1	PORT5 データの読み込み
0	1	1	0	1	PORT6 データの読み込み
0	1	1	1	1	PORT7 データの読み込み
1	0	0	0	0	I/O-0~7 のコントロールバイト

9. 3 コントロールバイト (port の入出力方向を設定)

A3		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
“1”		port7 入力	port6 入力	Port5 入力	port 4 入力	port3 入力	port2 入力	port1 入力	port0 入力
“0”		port7 出力	port6 出力	port5 出力	port4 出力	port3 出力	port2 出力	port1 出力	port0 出力

10. 入出力回路



出力回路はフリップフロップで記憶され、入力はコネクタから直接バッファに入ります。

出力モードに設定されていても出力データを読み取ることができます。

回路図中の nOUTOE0 信号に”0”が入力されると U23 のフリップフロップの出力はイネーブル状態となります。

入出力信号は全てプルアップ、又はプルダウンが設定できますが、出荷時はプルアップされます。

1 1.設定

1 1. 1 ディップスイッチ

SW1

Bit8	bit 7	bit 6	bit 5	bit	bit 3	bit 2	bit 1
S2	S1	未使用	未使用	未使用	未使用	未使用	未使用

I2C インターフェースのアドレス設定ビットです。

1 1. 2 ジャンパースイッチ（出荷時は全 port プルアップ）

port	ジャンパー	プルアップ	プルダウン	注意事項
port 0	JP1	ショート	オープン	JP1,1P2 両方ショートは禁止
	JP2	オープン	ショート	
port 1	JP3	ショート	オープン	JP3,1P4 両方ショートは禁止
	JP4	オープン	ショート	
port 2	JP5	ショート	オープン	JP5,1P6 両方ショートは禁止
	JP6	オープン	ショート	
port 3	JP7	ショート	オープン	JP7,1P8 両方ショートは禁止
	JP8	オープン	ショート	
port 4	JP9	ショート	オープン	JP9,1P10 両方ショートは禁止
	JP10	オープン	ショート	
port 5	JP11	ショート	オープン	JP11,1P12 両方ショートは禁止
	JP12	オープン	ショート	
port 6	JP13	ショート	オープン	JP13,1P14 両方ショートは禁止
	JP14	オープン	ショート	
port 7	JP15	ショート	オープン	JP15,1P16 両方ショートは禁止
	JP16	オープン	ショート	

同一 port で両方のジャンパーをショートすると、+5V と GND が短絡します。

1 2. コネクタ

CN1 : JTAG 信号コネクタ

SM06B-NSHSS-TB (メーカー : 日本圧着端子製造株)

CN2 : ラズベリーパイ GPIO コネクタ

PC104-40HK (株式会社エンベデッドテクノロジー)

CN3、CN4、CN5、CN6 : 外部 I/O 接続コネクタ

S20B-PADSS-1 (メーカー : 日本圧着端子製造株)

1 3. I2C 通信の使用方法

例) ディップスイッチ 7,8 を両方 ON("00") にした時

- 1) ポート 3 に "AA"hex を出力する
- 2) ポート 0~3 : 出力、ポート 4~7 : 入力に設定する
- 3) ポート 7 からデータを読み込む

1) ポートへの出力データの書き込み (ポート 3 に "AA"hex を出力する)

①スレーブアドレスの設定

コントロールビットに "0" を設定し、ポートアドレスビットに "3" を設定する。

"0x13"

スレーブアドレス						
0	0	1	0	0	1	1
dip8	dip7	固定	コントロールビット	ポートアドレスビット "3"		

②データ "0xaa" を書く

1	0	1	0	1	0	1	0
"A"				"A"			

ラズベリーパイコマンド例 : "i2cset -y 1 0x13 0xaa"

※-y の後ろの数字は "1" もしくは "0" が入ります。

2) コントロールバイト (ポートの入力/出力) の設定
 (ポート 0~3 : 出力、ポート 4~7 : 入力に設定する)

①コントロールビットに”1”を設定 “0x18”

スレーブアドレス						
0	0	1	1	0	0	0
dip8	dip7	固定	コントロールビット	固定	固定	固定

②コントロールバイトデータの設定 (ポート 0~3 : 出力、ポート 4~7 : 入力の例)
 “0xf0”

1	1	1	1	0	0	0	0
Port7	port6	port5	port4	port3	port2	port1	port0
入力	入力	入力	入力	出力	出力	出力	出力

ラズベリーパイコマンド例 : “i2cset -y 1 0x18 0xf0”

※-y の後ろの数字は”1”もしくは”0”が入ります。

3) ポートからのデータ入力 (ポート 7 からデータを読み込む)

スレーブアドレス						
0	0	1	0	1	1	1
dip8	dip7	固定	コントロールビット	ポート “7”		

X	x	x	x	x	x	x	x
ポート 7 の読取りデータ							

ラズベリーパイコマンド例 : “i2cget -y 1 0x17”

※ -y の後ろの数字は”1”もしくは”0”が入ります。

注意 :

- ・ コントロールバイトの設定で「出力」に設定するポートには、必ずコントロールバイト設定の前に「項 1) ポートへの出力データの書き込み」を行って下さい。
- ・ コントロールバイトの設定で「出力」に設定されているポートを読んだ場合は、現在出力されているデータが読まれます。

1 4. サンプルプログラム

```
/***RAS-64DIO-I2C SAMPLE PROGRAM***/

#include<stdio.h>
#include<stdlib.h>
#include<linux/i2c-dev.h>
#include<fcntl.h>
#include<string.h>
#include<sys/ioctl.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<unistd.h>

int fd;
char *i2cdev = "/dev/i2c-1";
unsigned char buf[16];

// I/O control address to configure I/O ports "input" or "output"
unsigned char IOcontrol_address = 0x18;
// I/O port address : 0x10~0x17
unsigned char IO_address[8] = {0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17};
//LED DATA
unsigned char data[8] = {0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80};

int i2cwrite(char *buf, int len)
{
    if((write(fd, buf, len)) != len) {
        printf("Err:- i2cwrite\n");
        return(1);
    }
    return(0);
}

int i2cread(int offset, char *buf, int len)
{
    *buf = (char)offset;
    if(read(fd, buf, len) != len) {
        printf("Err:- i2cread\n");
        return(1);
    }
    return(0);
}

int main(int arg, char *argv[])
{
    int i = 0;
```

```

printf("**** RAS-64DI0-I2C SAMPLE ****\n");
if((fd = open(i2cdev, O_RDWR)) < 0) {
    printf("Can't open i2c\n");
    exit(1);
}

//Write led data to all I/Oport//
for(i = 0; i < 8; i++) {
    if(ioctl(fd, I2C_SLAVE, IO_address[i]) < 0) {
        printf("Unable to bus access to talk to slave\n");
        exit(1);
    }
    buf[0] = data[i];

    if(i2cwrite(buf, 1)) {
        exit(1);
    }
}

//Write "0x00" to I/Ocontrol_address to configure all I/O ports "OUTPUT"//
if(ioctl(fd, I2C_SLAVE, IOcontrol_address) < 0) {
    printf("Unable to get bus access to talk to slave\n");
    exit(1);
}
buf[0] = 0x00;
if(i2cwrite(buf, 1)) {
    printf("can't write 0x18\n");
    exit(1);
}

//Read all I/O ports//
for(i = 0; i < 8; i++) {
    if(ioctl(fd, I2C_SLAVE, IO_address[i]) < 0) {
        printf("Unable to bus access to talk to slave\n");
        exit(1);
    }

    if(i2cread(fd, buf, 1)) {
        printf("error %d\n", i);
        exit(1);
    }
    printf("%02x\n", buf[0]);
}

close(fd);
return 0;
}

```

15. 拡張ボードとラズベリーパイとの連結

(写真は RASPBERRY PI 2 MODEL B+です)

注) 下記の作業は必ず電源を切ってから行ってください。

拡張ボードとラズベリーパイの連結は、中間基板である MPC-RAS-ST-S を取り付けなければ、連結できません。

①ラズベリーパイ本体にスペーサーを取りつけます。



添付のスペーサー (※14mm) を取り付け基板底からナット止め

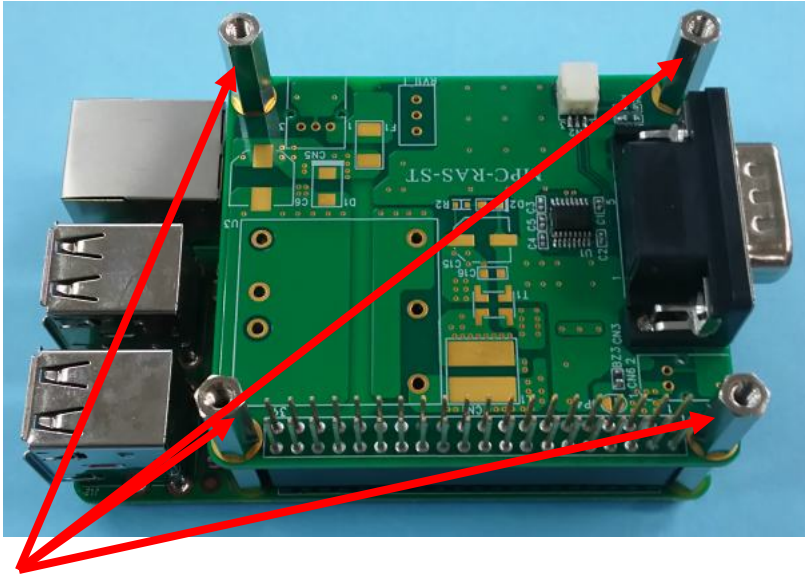
※MPC-RAS-ST に添付されているスペーサーは長さ 14mm、MPC-RAS64DIO に添付されているスペーサーは 15mmです。ここでは 14mmのスペーサーを取り付けて下さい。

②ラズベリーパイ本体の GPIO 拡張コネクタと MPC-RAS-ST の GPIO 拡張コネクタを連結します。

このラズベリーパイ本体の真上には MPC-RAS-64DIO は直接実装できませんので、本体真上には必ず MPC-RAS-ST が必要になります。(写真は MPC-RAS-ST)



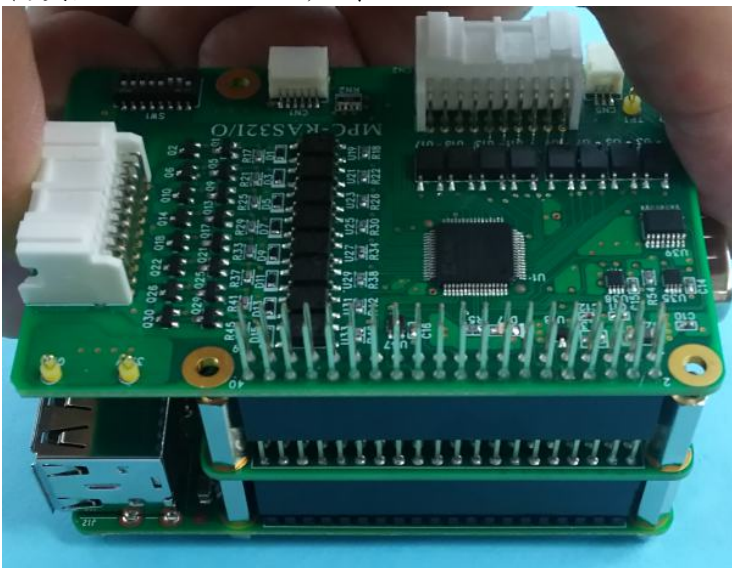
③本体の上に MPC-RAS-ST を取り付けました。



MPC-RAS-ST を本体の上に取り付けたあとスペーサー (15mm) を取りつけます

④更にその上に MPC-RAS-32IO を取りつけます。

(写真は MPC-RAS-32IO ですが、MPC-RAS-32IO と MPC-RAS-64DIO は同型です)



⑤四か所のネジをしっかりと止めて終わりです。



株式会社エンベデッドテクノロジー

〒577 大阪府東大阪市瓜生堂 3-8-13

奥田ビル 2F

電話 06-785-2713 FAX 06-785-2716

<http://www.emb-tech.co.jp>