

拡張機能とは・・・I/Oが足りなくなったときやPICを連携して制御したいときに利用する機能

CPUマルチ拡張・・・同一CPUを接続し、通信によりI/Oを増やす方法  
(XやYで拡張、拡張側のCPUはラダーを記載できない。)

Expander 拡張・・・MCP23008やMCP23017をI2Cで接続し、拡張する方法 (XやYで拡張)

I2C拡張・・・同一シリーズCPUをI2Cで接続して、BデバイスやWデバイスをやり取りする方法 (CPUそれぞれでラダーが記述できる)

接続距離が短い。また、デバイスが増えると通信速度が極端に遅くなる。4台位が限界です (P452-I2CINF-LB が該当)

CAN拡張・・・458と458、4585と4585・4680といったCPUをCANで接続して、BデバイスやWデバイスをやり取りする方法 (CPUそれぞれでラダーが記述できる) 接続距離が長く、通信速度は速い

(P452-CAN-LB P18F452-LB には 18F458 が対応、P18F4525-LB は 18F4585 が対応)

CPU・機能	MCP23008 MCP23017 使用	同一CPU(マルチ) 連結による拡張	I2C通信	CAN通信
PIC16Fシリーズ	◎	×	×	×
PIC18Fシリーズ	◎	◎	◎	×
PIC18F458 (258)	◎	◎	◎	◎

註) PIC16F88は、MCP23008の拡張のみサポートしています。

註) PIC16Fシリーズは、MCP23008とMCP23017の拡張をサポートしていますが、この2種類を同時に使用することはできません。

PIC18Fシリーズは、MCP23008とMCP23017の拡張を同時に使用することはできます。

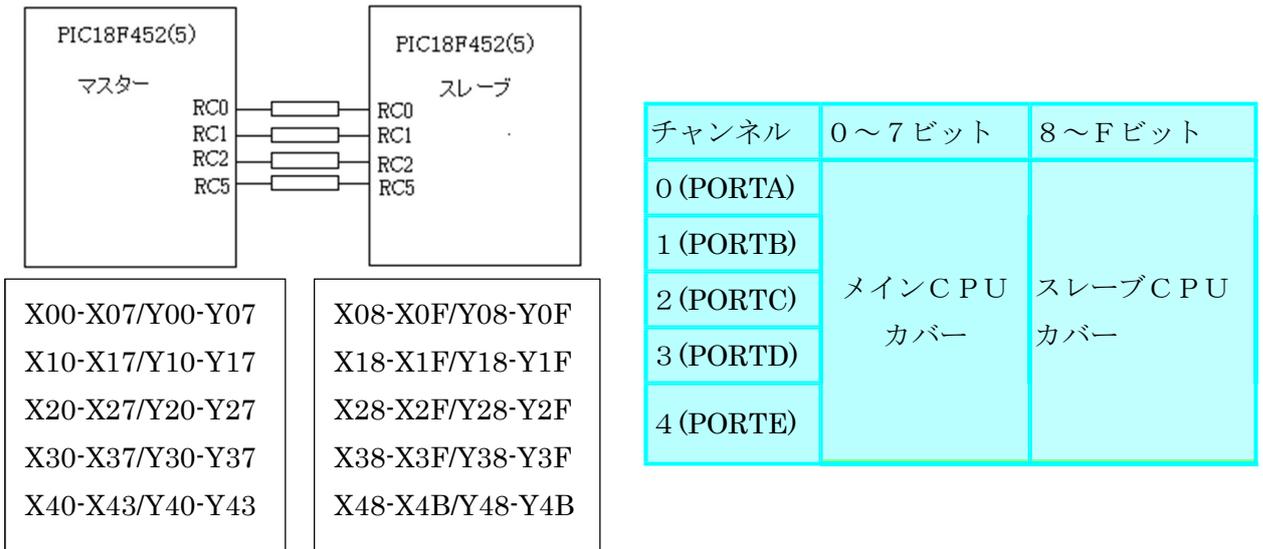
その結果、合計88(-2)ピンのI/Oの拡張ができます。

註) 拡張、I2C、CAN通信を使用するには別売りライブラリーが必要です

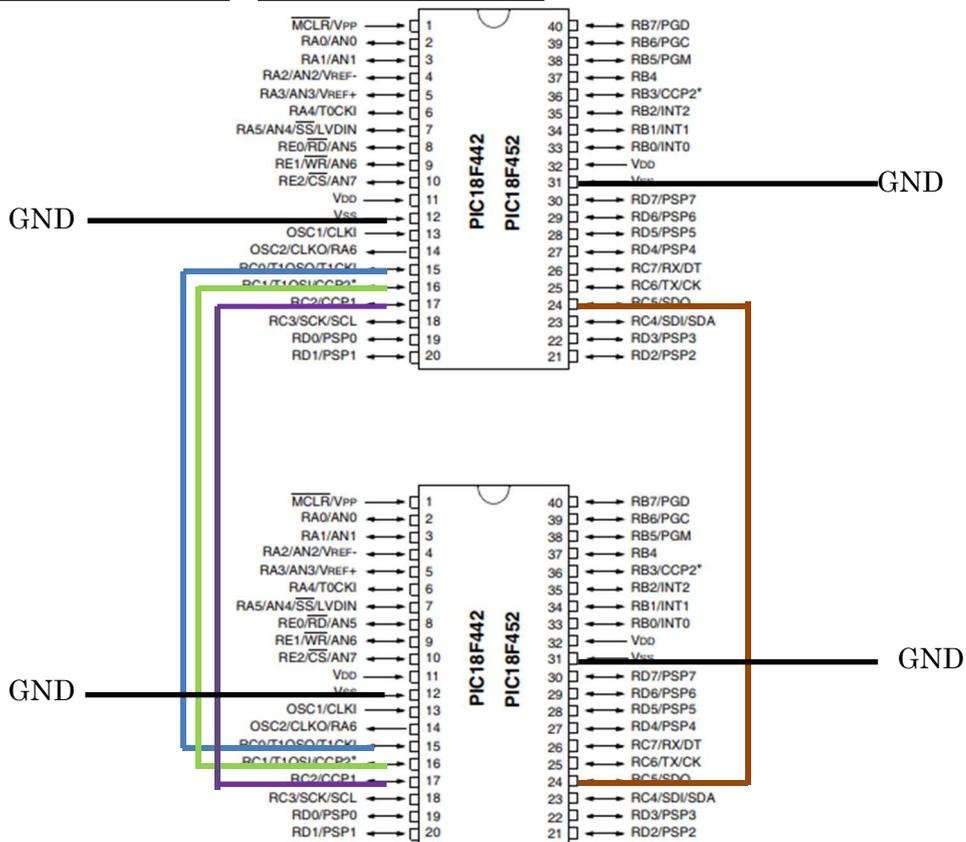
1. CPU (マルチ拡張) 同一PIC素子を使用する場合

PIC18Fシリーズすべてに摘要されます。接続抵抗の抵抗値は0が原則です。マスター、スレーブとも、どのピンを入力に使用するか、出力に使用するかピン毎に任意です。

ハードウェアの設定・・・RC0,RC1,RC2,RC5 を通信で使用します (IO では使用不可)

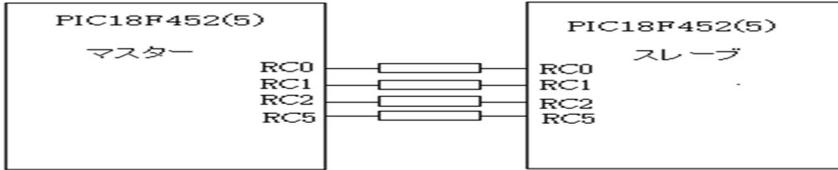


チャンネル	0～7ビット	8～Fビット
0 (PORTA)	メインCPU カバー	スレーブCPU カバー
1 (PORTB)		
2 (PORTC)		
3 (PORTD)		
4 (PORTE)		



※配線は通信部分のみ記載しています。  
その他にも、電源等をつなぐ必要があります。

- ソフトウェアの設定



スレーブ（拡張）側には PIC18F452-LB 使用の場合は ¥8F5¥E08¥ 中の EXSLAV6.HEX を

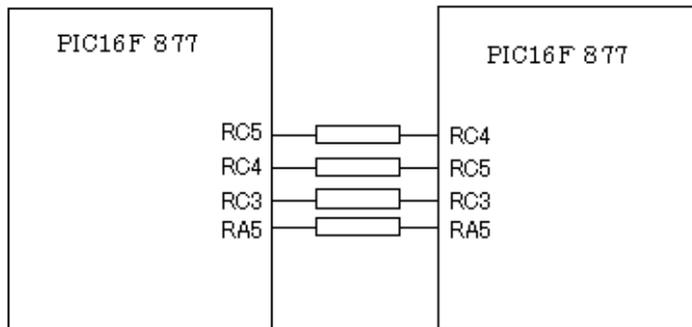
PIC18F4525-LB 使用の場合は ¥8F55¥E08¥ 中の EX55SLAV6.HEX を書き込んでください。

スレーブ（拡張）側はマスター側と I/O やり取りの動作のみをします。

※スレーブ側にラダーは作成できません。

※ 887 や 877 の接続回路

887 や 877 のマニュアルには以下のように記載されています。ただ、引継ぎした時点の HP では、PIC16 シリーズのマルチ CPU 拡張は未サポートのようです。P1 の水色の表が、該当します。多分、動くとは思いますが・・・。



## 2. Expander 拡張の使用方法 (MCP32008、MCP32017)

「連枝」(れんり) では I/O の拡張用素子として MCP23008、MCP23017 を使用し、ラダー図上で X、Y の各チャンネルの 8~F (23008)、0~F (23017) を使用するだけで外部に接続された各々 8 ピン、16 ピンの I/O をアクセスすることができます。各ピンを入力 (X) に使用するか出力 (Y) に使用するかは任意です。拡張は I2C と同じ回線を使用します。

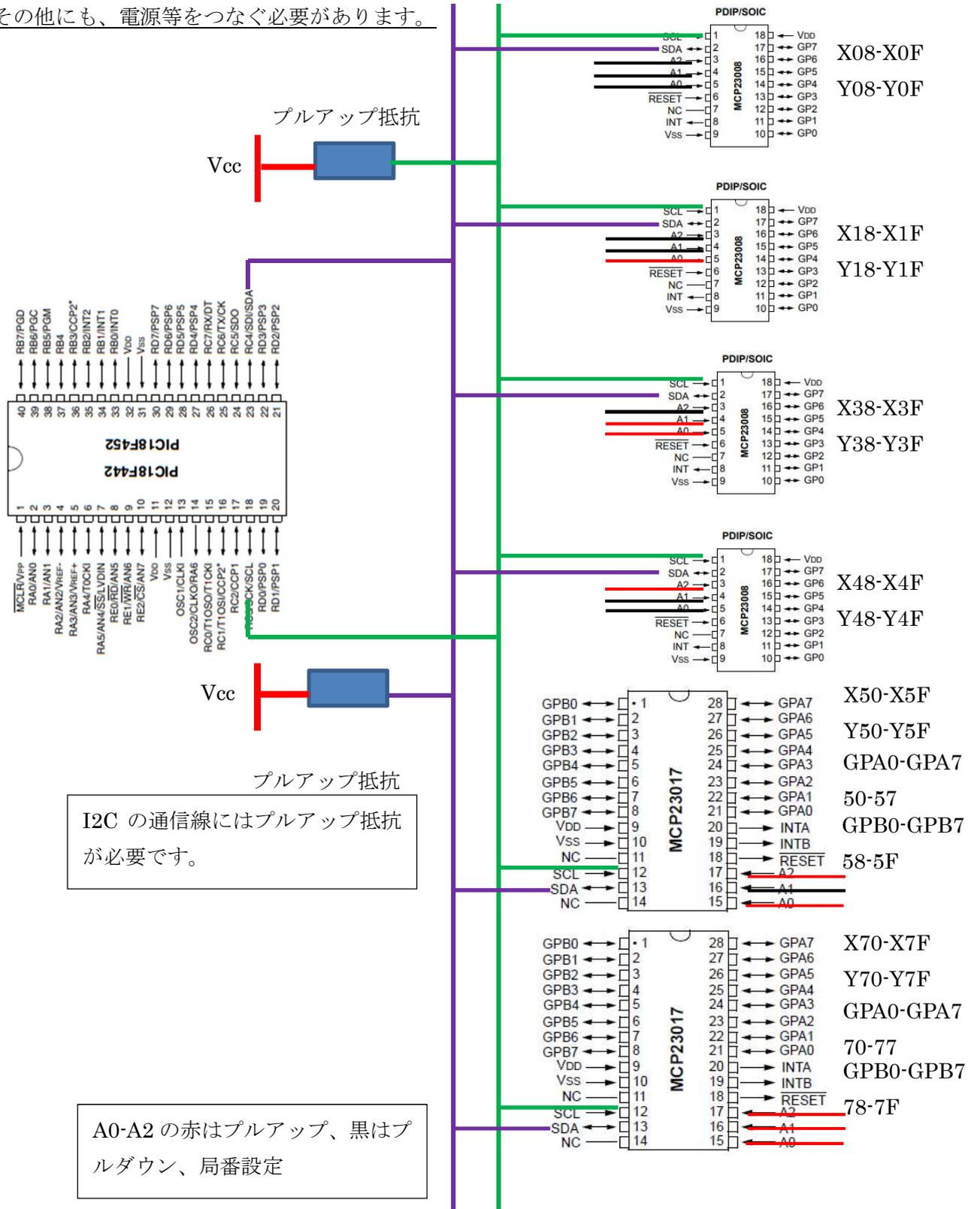
従ってこの回線を利用すれば、MCP23008 または MCP23017 をロボットの端末として使用できます。足りなければ 2 つ以上を同一場所に使用してピン数を倍にすることもできます (これら拡張用素子を制御できるのは、1 箇の CPU のみですが、I2C 通信の性質上、拡張用素子は、I2C 通信ライン上であれば、どこにでも置くことができます。)

### ・ハードウェアの設定

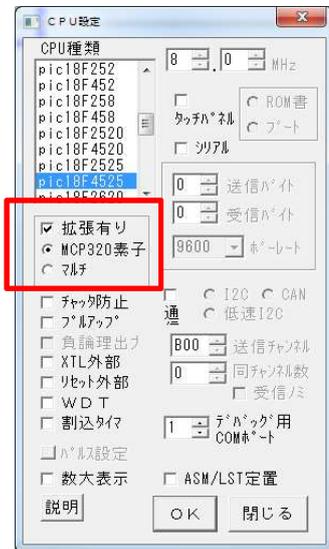
CPU 搭載の I2C 通信ポートに Expander IC を接続する。  
MCP32008 や MCP32017 を番号に合わせて、局番設定する。  
以下の表の ? は、X または Y が入ります。

チャンネル 局番	0~7 ビット	8~F ビット
0	メイン CPU カバー	MCP32008-0 カバー(?08-?0F)
1		MCP32008-1 カバー(?18-?1F)
2		MCP32008-2 カバー(?28-?2F)
3		MCP32008-3 カバー(?38-?3F)
4		MCP32008-4 カバー(?48-?4F)
5	MCP23017-0 カバー (GPA0-GPA7 : ?50-?57) (GPB0-GPB7 : ?58-?5F)	
6	MCP23017-1 カバー (GPA0-GPA7 : ?60-?67) (GPB0-GPB7 : ?68-?6F)	
7	MCP23017-2 カバー (GPA0-GPA7 : ?70-?77) (GPB0-GPB7 : ?78-?7F)	

※配線は通信部分と局番部分のみ記載しています。  
 その他にも、電源等をつなぐ必要があります。



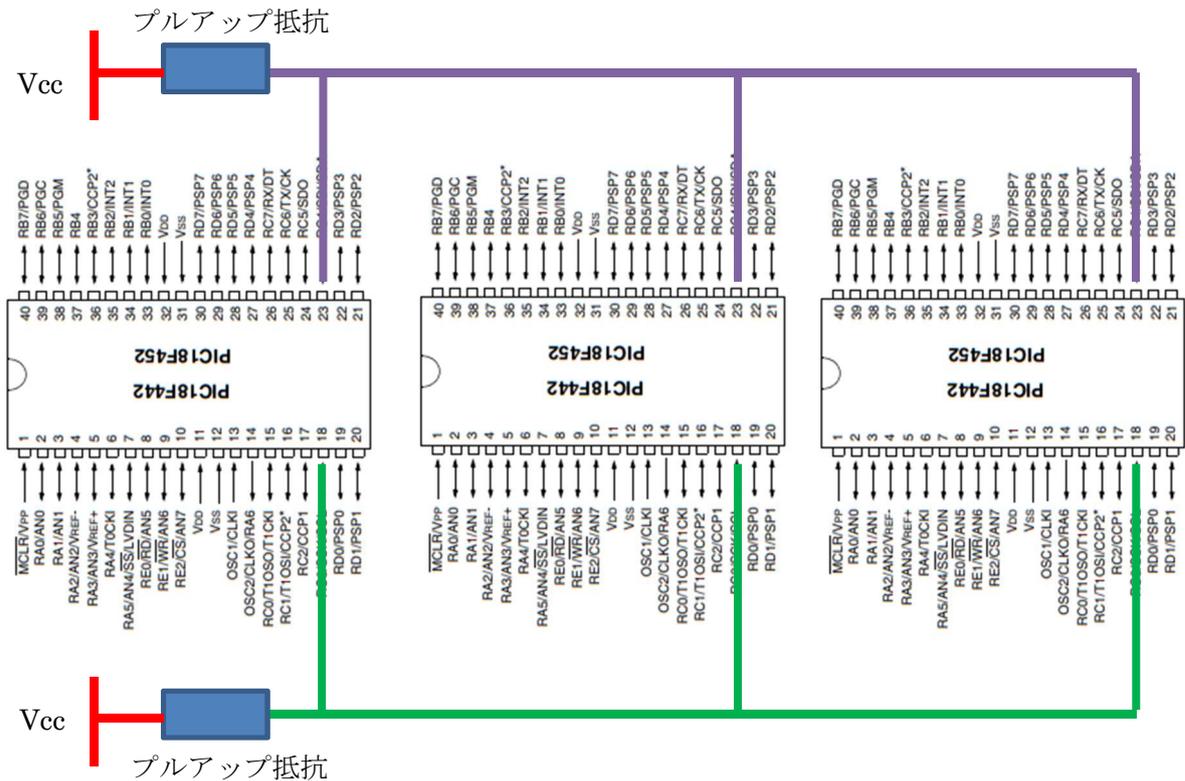
## ソフトウェア上の設定



### 3. I2C拡張 ( P452-I2CINF-LB )

- ・ハードウェアの設定

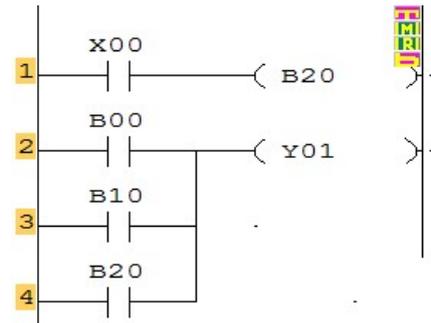
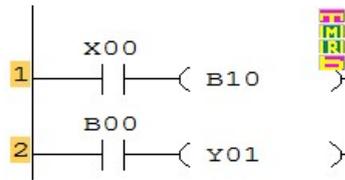
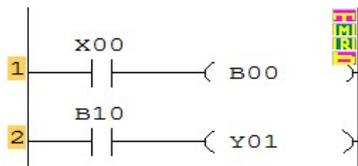
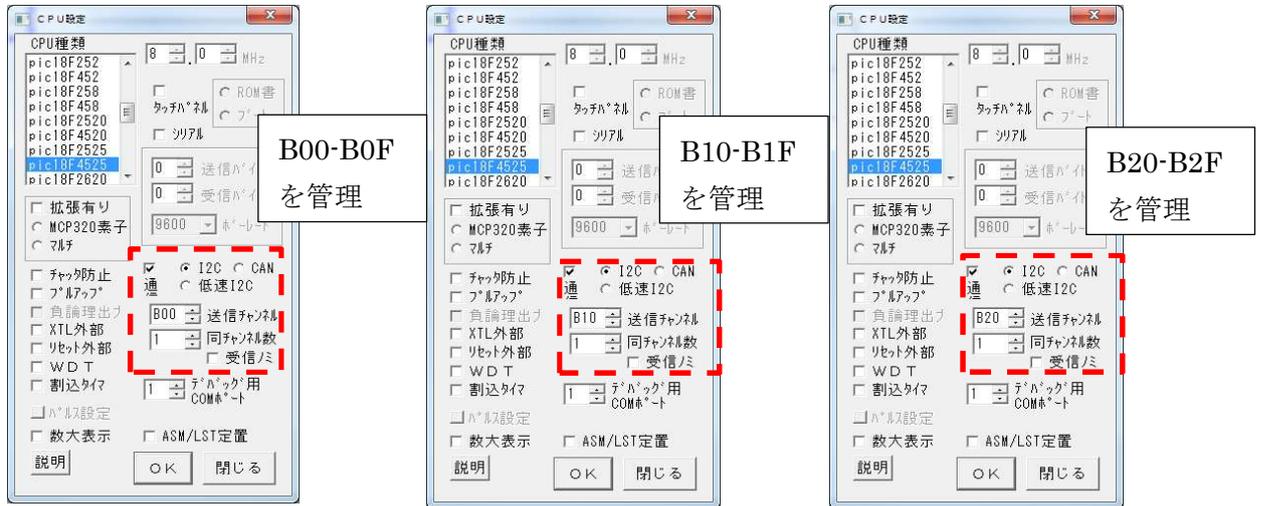
I2Cの通信ができるように配線します。プルアップの抵抗値については、回路の状態により変更します。(最大4個程度です。それ以上配線できますが、動作が不安定です。)



I2Cの回路については、各々のCPUのSCLピンどうし、SDAピンどうしを接続するだけです。

(SCLとSDAピンは、18F452は18、23番ピン、18F252は14、15番ピンです)。

・ソフトウェアの設定

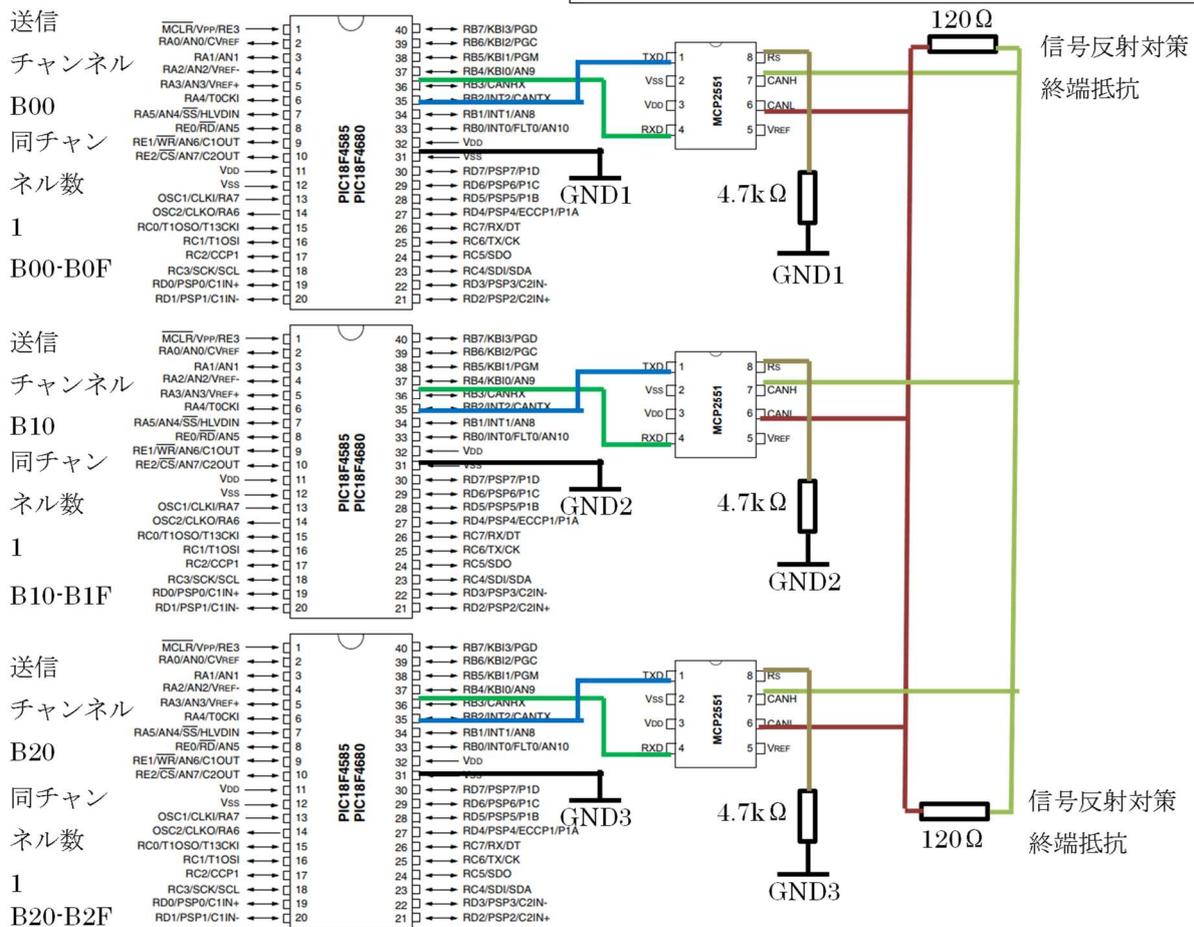


4. CANの使用法 (P452-CAN-LB [P18F452-LB は 18F458 対応、P18F4525-LB は 18F4585/4580 が対応])

・ハードウェアの設定

W0-W8(PIC18F458,258)  
W0-W32(PIC18F4680,4585)

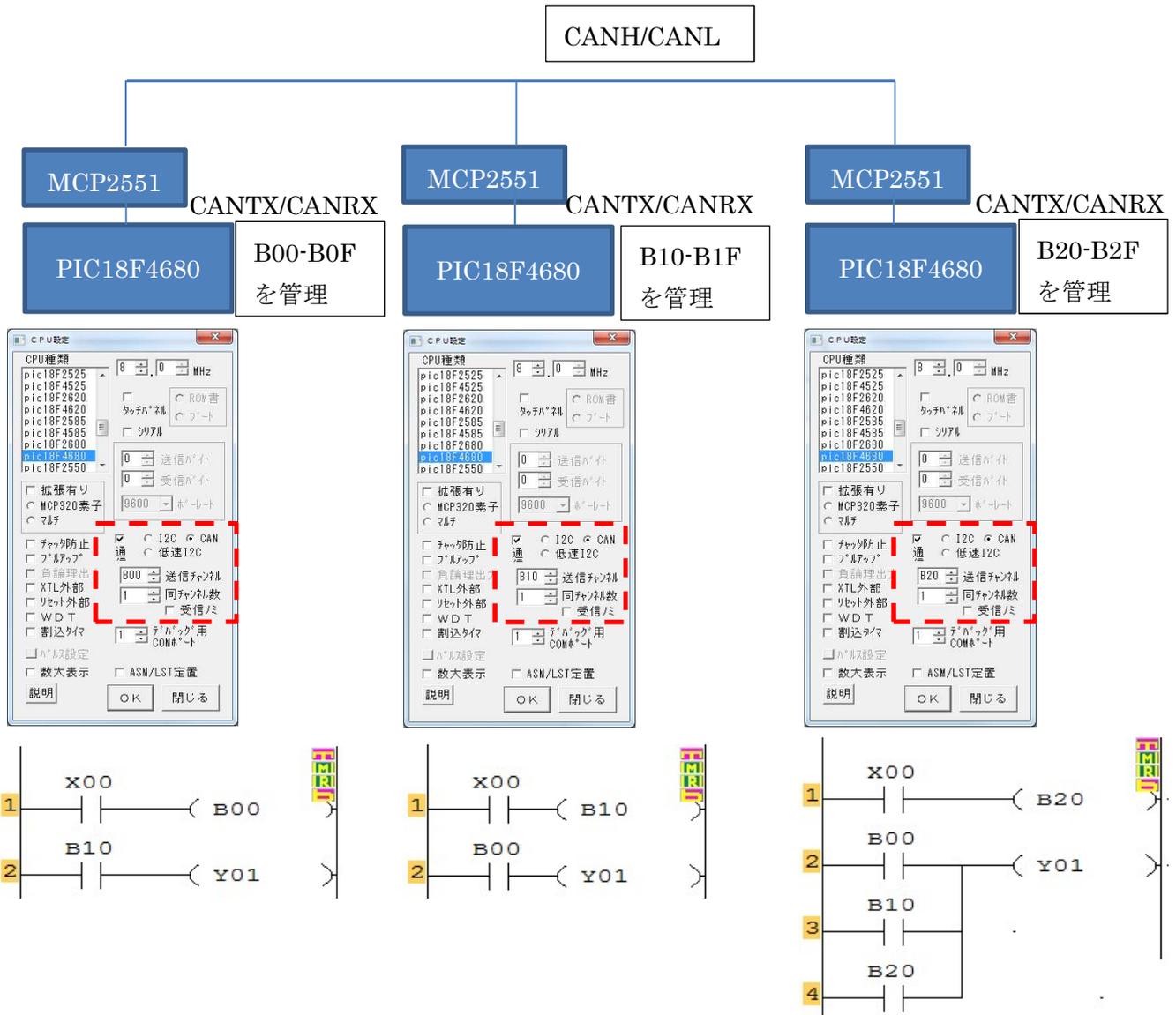
Wはすべての端末で読み書き可能。Bは読み出しはすべての端末で可能。書き換えは送信チャンネルとなっている端末のみ可能。B00は必ず必要。



CANは差動信号なので、GNDは異なっても問題ありません。ノイズに強く、距離が取れます。

・ソフトウェアの設定

責任デバイス（管理デバイス）を指定する。送信チャンネルと同チャンネル数により、CPUが管理するBデバイスの開始番号と数が指定できる。チャンネル数1あたり16ビットを管理。Wデバイスについては、すべての端末で変更・参照可能。受信のみの端末でも、送信チャンネルは設定が必要となる。



③ラダー内で管理デバイスを変更する。

他の管理デバイスを使用する場合は、読みだす。接点として使用。

## I 2 C ・ C A N通信の追記

自動車産業では現在、殆どすべての車内の各機器の通信をC A N通信によっています。

工具のボッシュ社の提案による方法で、通信のやり取りに関する複雑な殆どの部分を I C内部の物理層で解決する方法です。

「連枝」(れんり)では、P I C 1 8 F 4 5 8にこのC A N通信機能が内蔵されているのを利用して、接点 ( Bリレー) とメモリ ( W) を、各機器相互に通信して共通の値いとして利用できるようにしました

「連枝」では I 2 CもC A Nも同じ仕様です。

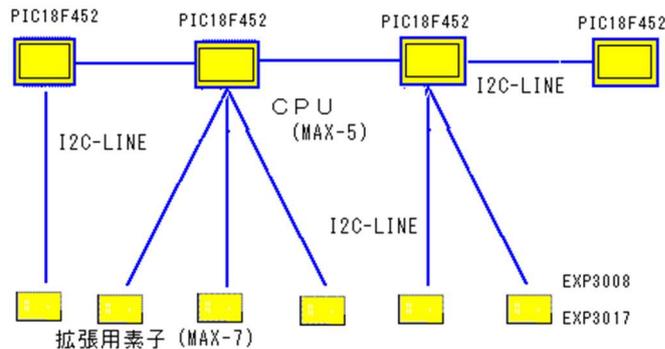
I 2 CとC A Nは通信規格が全く別のもので、一般的には前者は同一制御盤内で、後者は制御盤外の長距離の通信をします。

しかし、「連枝」(れんり)にとっては接点 ( Bリレー) とメモリ ( W) の変化を各C P U間に通信して、共通の値いとする仕様は全く同じです。

◎ユーザーはBリレーとWメモリを回路上で使用し、どちらの通信かを設定するだけでプロトコルについて何も考える必要はありません

目安としては、I 2 Cが5 0 c M以内、C A Nが1 5 M以内が考えられますが、その仕様については使用状況やそれぞれの規格が基準になります。

C A N通信にはシリアル通信のM A X 2 3 2にあたる周辺素子M C P 2 5 5 1の接続が必要ですが、I 2 C通信は各C P Uのピンどうしの接続のみで、外部周辺素子は不要です。



ラインで複数箇所で操作する場合、従来は押し釦やパイロットランプ等を実配線で配線していましたが、この従来方法だと配線が複雑かつ冗長になります。

マイコンの発達により、シリアル通信の方法がとられるようになりましたが、このためには高価なシステムを組み込むか、自ら複雑なプロトコルを組んで動作させなくてはなりませんでした。

「連枝」(れんり)の I 2 C通信やC A N通信によれば、Bが頭につく Bリレーを使用す

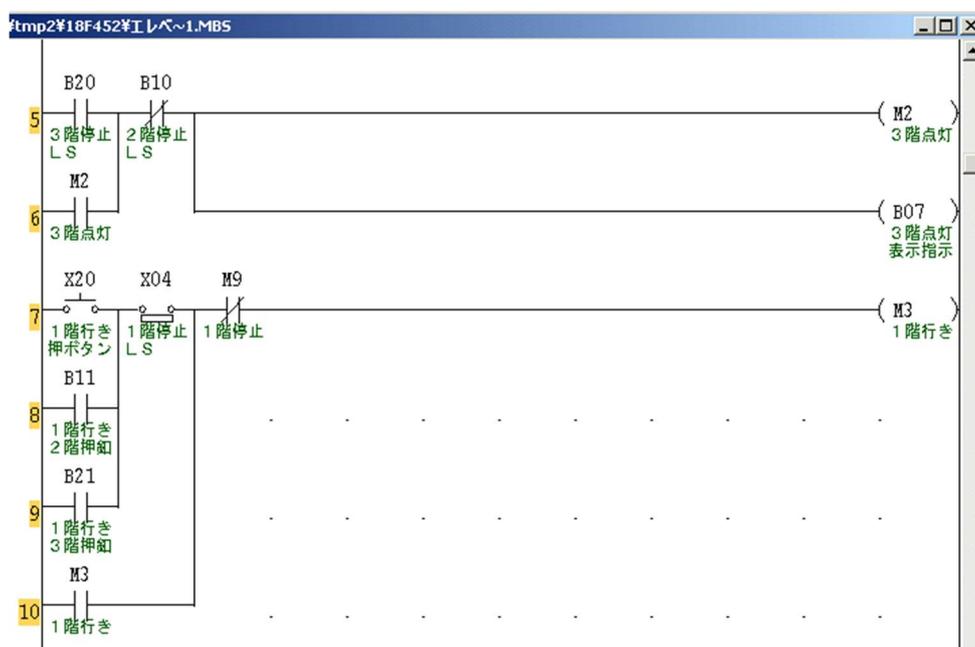
ればその変化値が、自動的に他の連結CPUに送られます。複雑なプロトコルは全て「連枝」側でコンパイルしてくれます。

◎CANでは通信失敗の場合はPIC18F458の持つCAN機能の物理層によって自動的にリトライがなされるとともに、「連枝」(れんり)のソフトによってもACKが確認されます。

◎I2CではこのCPUの物理層の持つリトライの機能や、「連枝」(れんり)のソフトによるACKの確認はありませんが、ACKの発行や2つのCPUによる通信衝突の際のアービトラージ(振り分け調整)等の多くの機能が物理的に内蔵されていて自動的に実行されます。

つまり、ユーザーとしてはBリレーを使用するだけの仕様です。他に16ビットのWメモリもそれを回路中に使用するだけで送信されます。

例として荷物用3階エレベータを、「連枝」(れんり)のCAN通信で操作する場合の回路の一部を次に示します(通常の回路は、DL版のサンプル中にあります)。

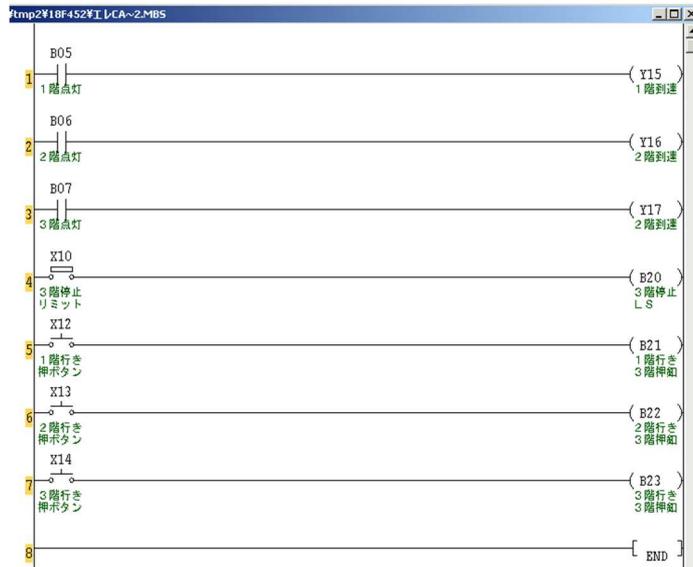


この時、6行目の出力のB07による3階ランプ点灯は各階で行わなくてはなりませんから、従来は実配線で各階に持っていかなくてはなりませんでした。

また7行目の1階行き押しボタンも各階に置かなくてはならないので、これも実配線で各階の押しボタンからの実配線が必要でした。

連枝ではそれらをすべて、接点Bとして制御盤において、あとは2本のCAN通信で各階の制御部分から持って来ればよいことになります。

3階の制御部の回路は以下ようになります。



以上で全回路です。単純な I / O だけの回路になります。

入力情報については I / O の結果を接点 B に変えるだけで、その結果が各階に通信されま  
す。

2 階も同様な回路になります。

本体制御盤と各階の通信によって、エレベータは動作します。2 階と 3 階も通信を行って  
いますがこの場合、意味はありません。

「連枝」はその他、W メモリー 16 ワードも相互に通信します。

これによって、各部所のアナログセンサーの値等を、本体を始めとする各制御部に送るこ  
とができます。

通信のプロトコル等はすべて「連枝」の変換ライブラリー (別売り) が構成してくれます。

ハード回路構成上、P I C 1 8 F 4 5 8 ( 2 5 8 ) と M C P 2 5 5 1 がメインの I C になり  
ます。両方ともマイクロチップ社製ですが、後者は R S 2 3 2 C 通信回路の M A X 2 3 2 に  
あたります。

CAN通信の回路例を示します。

18F458・CAN回路例

CAN通信

RSオンライン MCP2551 ロット5個単位 140円

(上記エレベータ回路ではありません)。

