

VisualC++によるYNUSBI/Oボードへのアクセスプログラムの作成手順

以下に VC++の空プロジェクトで C プログラムを作成し YNUSBI/O ボードでのアクセス方法を説明します。手順はかなり複雑です。

1 Visual Studio を立ち上げる。VisualC++ - Win32 選択し Win32 コンソールアプリケーションを選択する。図 1

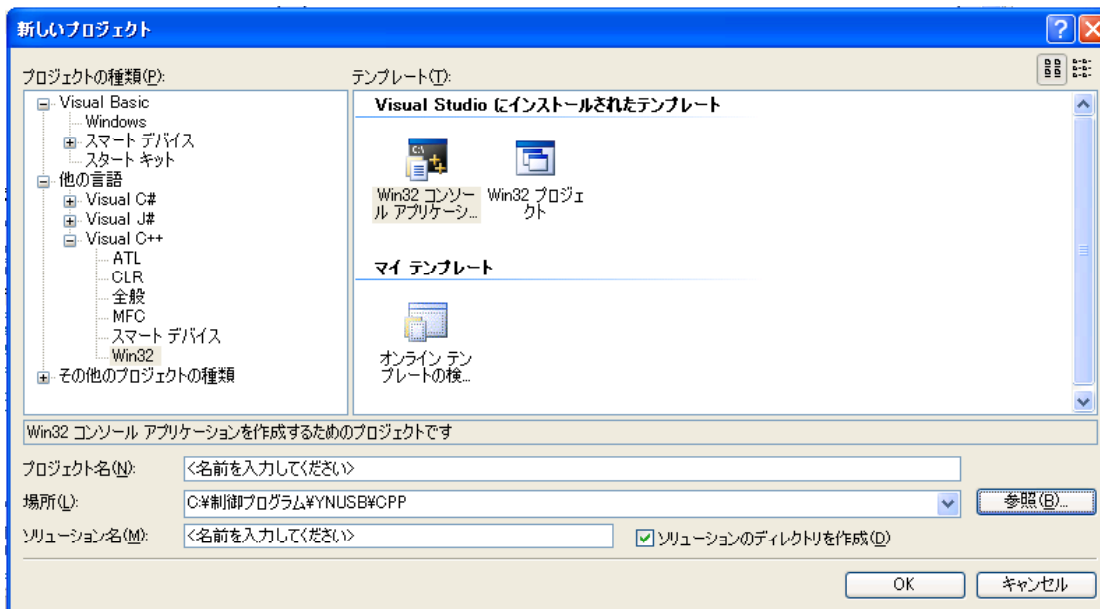


図 1

2 プロジェクト名を入力する。場所はあらかじめフォルダを作成しておく。図 2

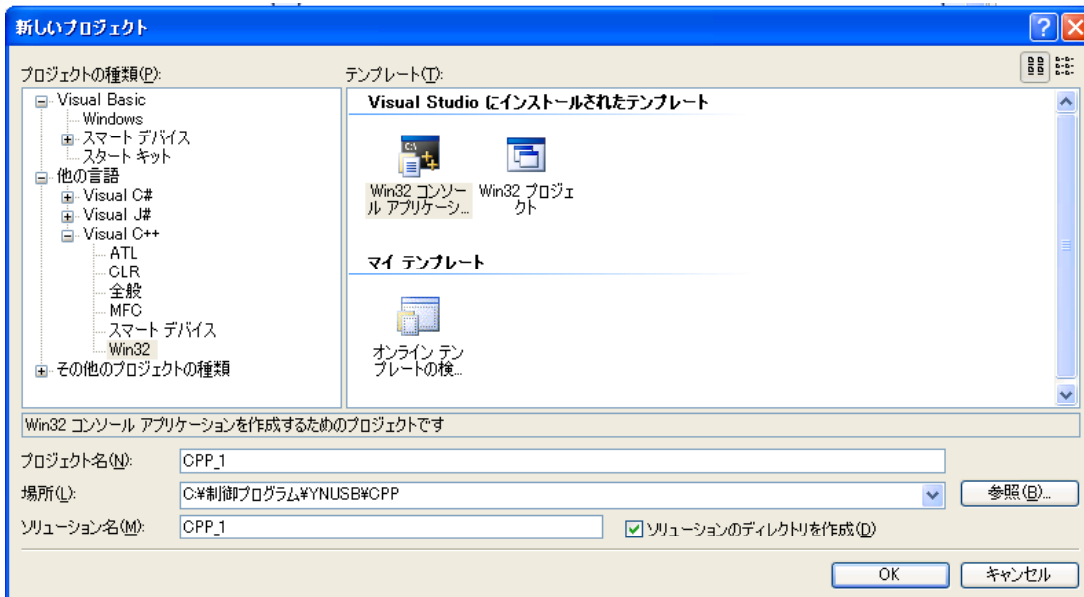


図 2

3 図 2 で OK をクリック。図 3 になる。次へをクリック。図 4 になる。

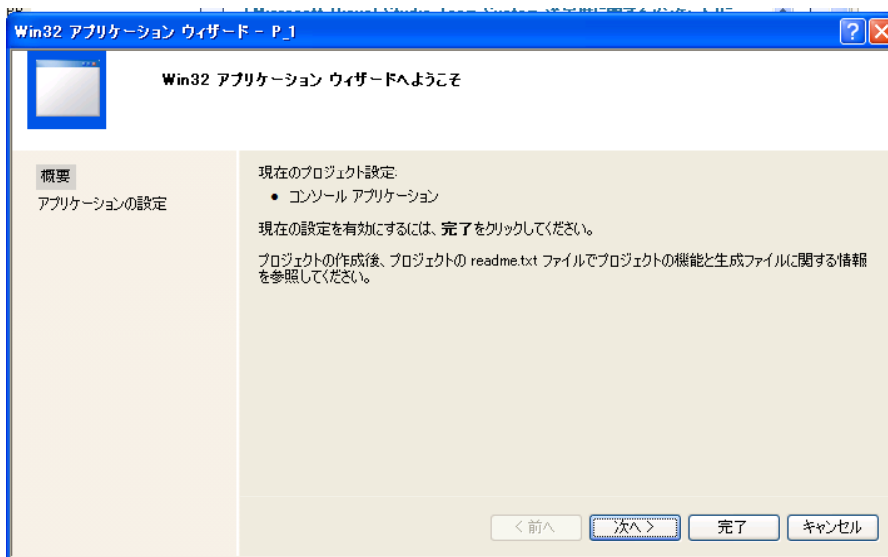


図 3

4 図 4 のようにコンソールアプリケーションと空のプロジェクトを選択し完了をクリック

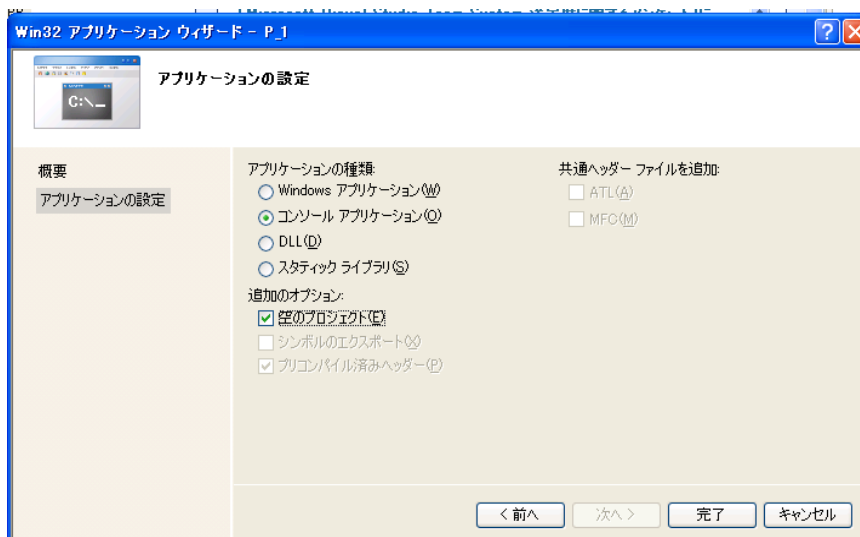


図 4

5 ソリューションエクスプローラー名 (図 5 (a)) を右クリックする。図 5 (b) のようにサブメニューが出るのでプロパティをクリック

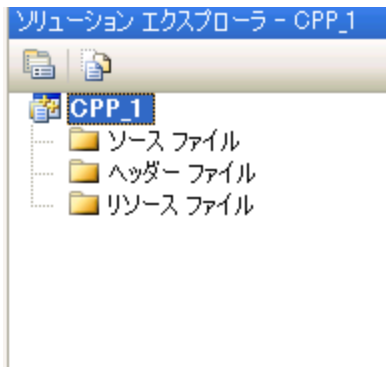


図 5 (a)

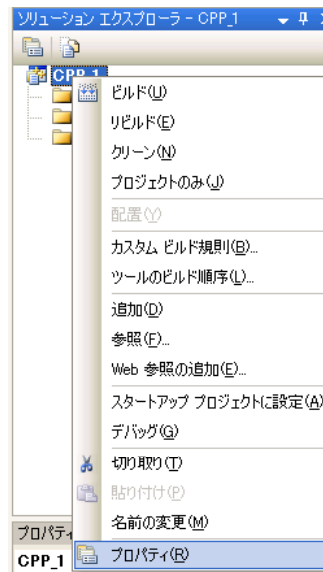


図 5 (b)

6 図 6 のプロパティページの画面でリンクを開きシステムをクリック。図 7 のようにコンソールに設定されているか確認する。

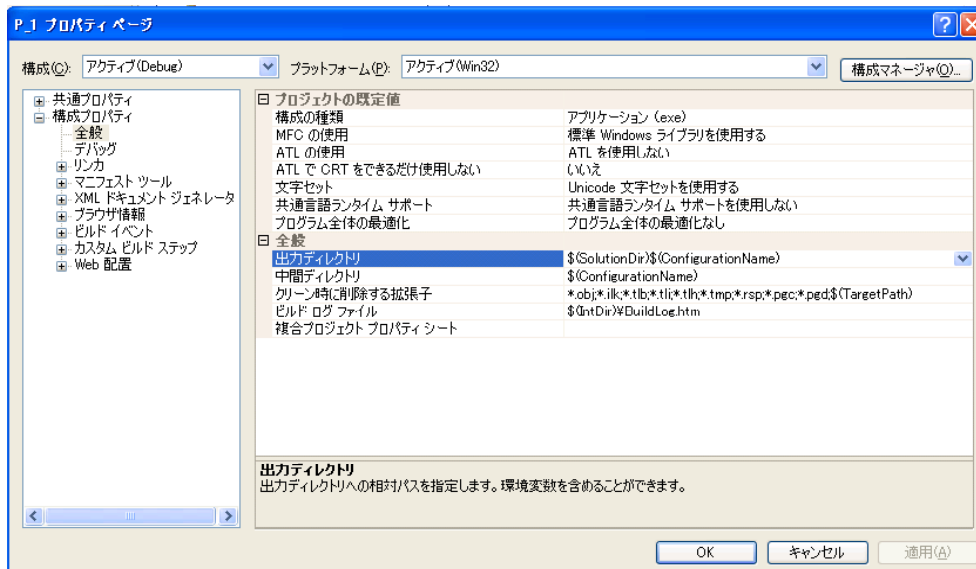


図 6

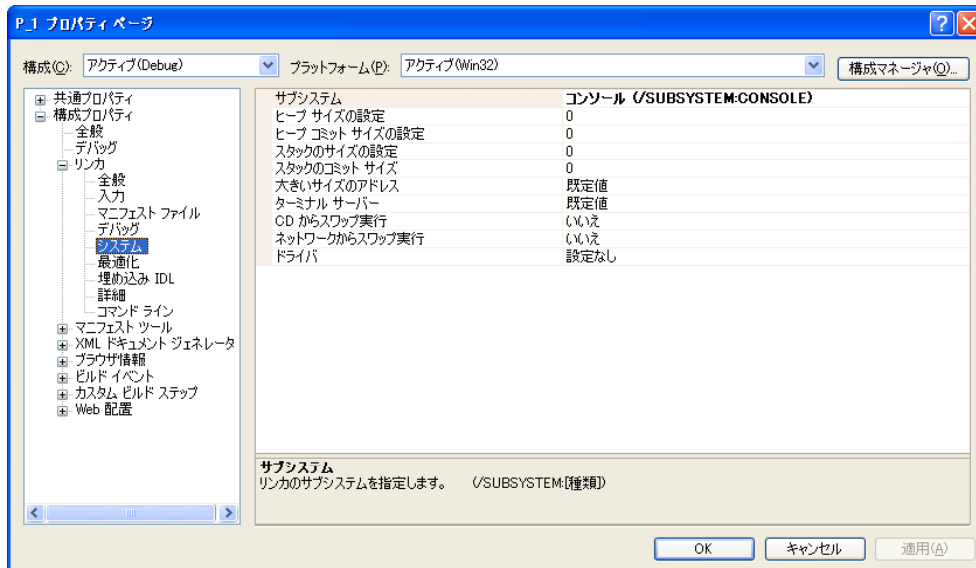


図 7

7 プロジェクト - 新しい項目の追加をクリック。

8 図 9 画面になるので Visual C++ から C++ ファイル(Cpp) を選択し追加をクリック

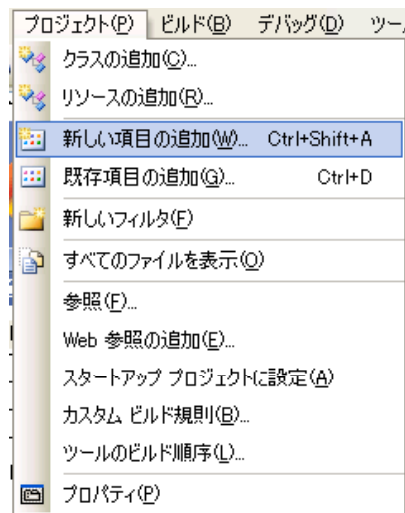


図 8

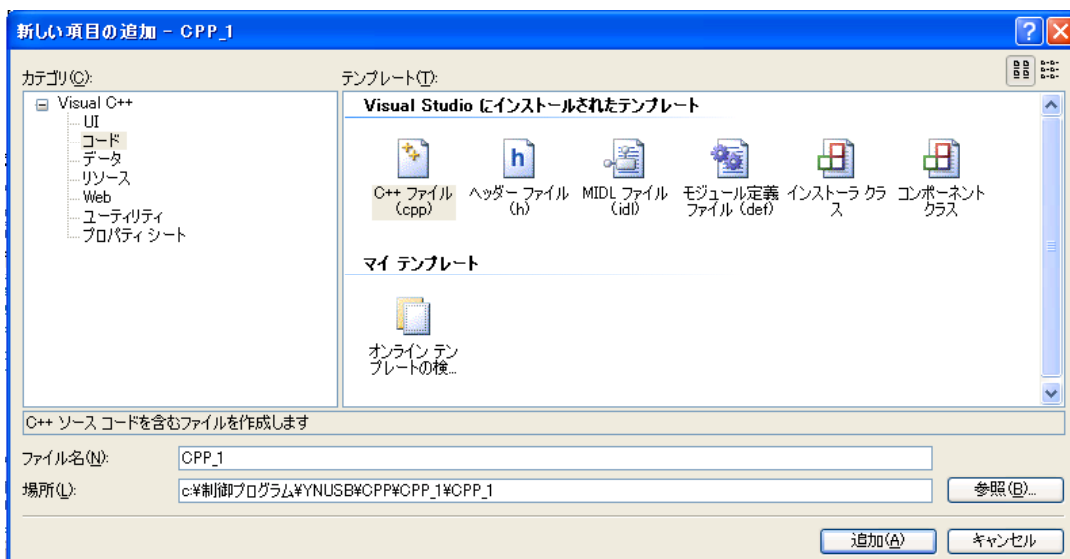


図 9

9 図10のエディタ画面になる。雛形は何もない。最後にあるリスト1を打ち込みます。

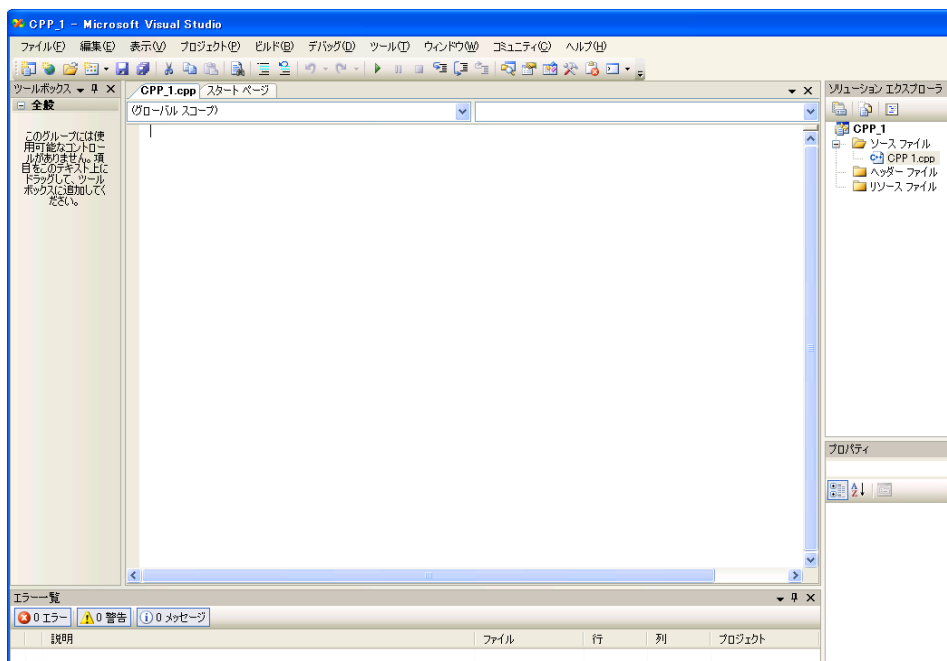
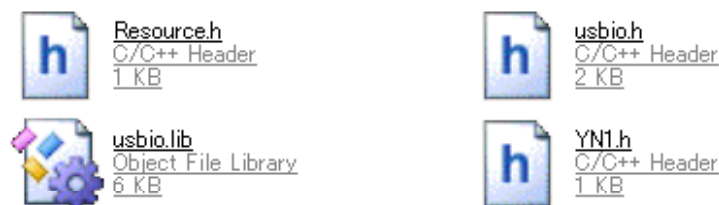


図 1 0

1 0 ここからがYUUSB I/Oボードでアクセスするための準備です。行うことは次の通りです。

(1) 図 1 1 の 4 つのファイルを C P P _ 1 のディレクトリにコピーする。このファイルは C D のサンプルプログラムの heda フォルダにあります。



(2) usbio.lib を追加する。

図 1 1

1 1 usbio.lib の追加手順を説明します。(大切 忘れないように)

(1) 5 のところの手順と同じにプロパティページを開きます。開いた画面で図 1 2 のようにリンカ - 入力を選択します。追加の依存ファイルを選択し「OK」をクリックし、右のボタンをクリックします。

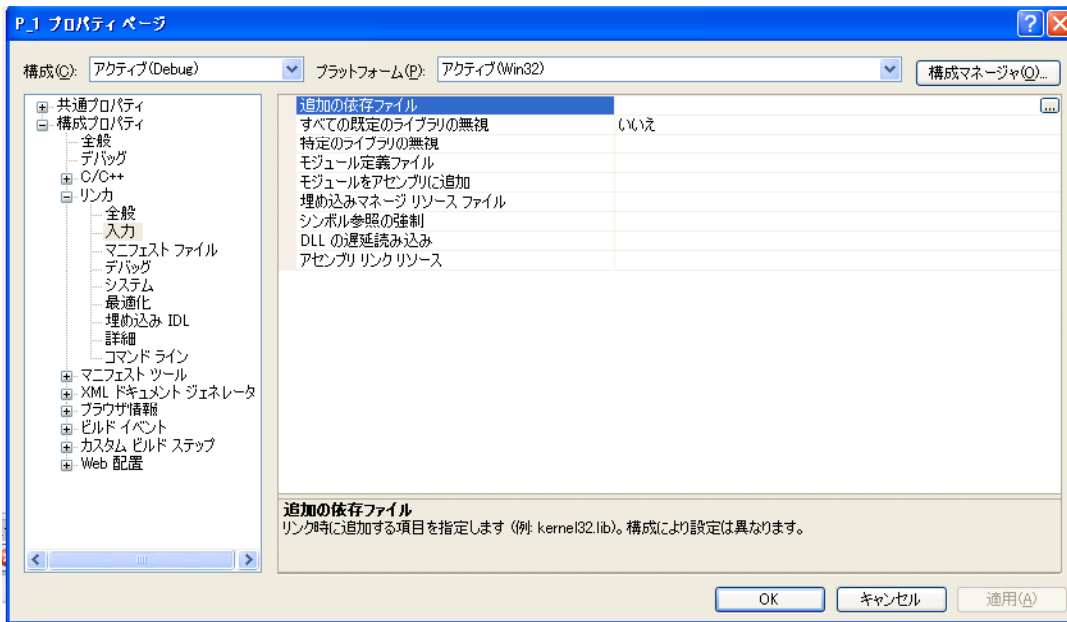


図 1 2

(2) 図 1 3 の追加の依存ファイルの画面になるので、図 1 4 のように「usbio.lib」と打ち込みます。「OK」をクリックすると図 1 5 のように usbio.lib が追加されます。

以上で終わりです。

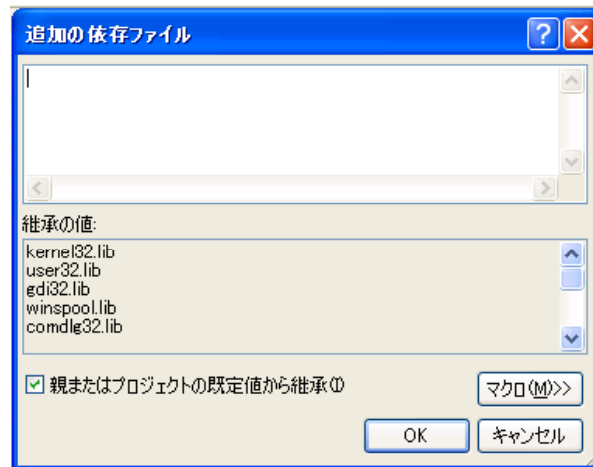


図 1 3

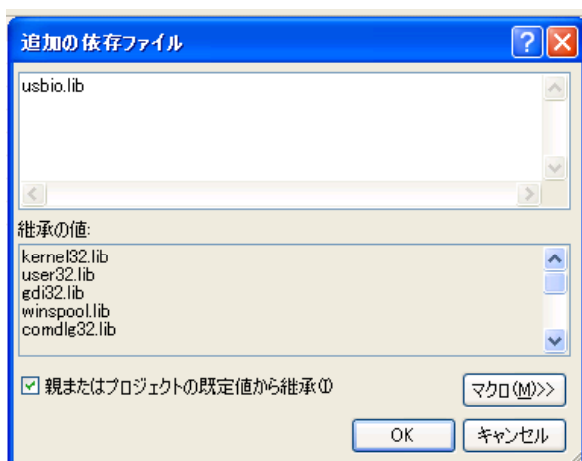


図 1 4

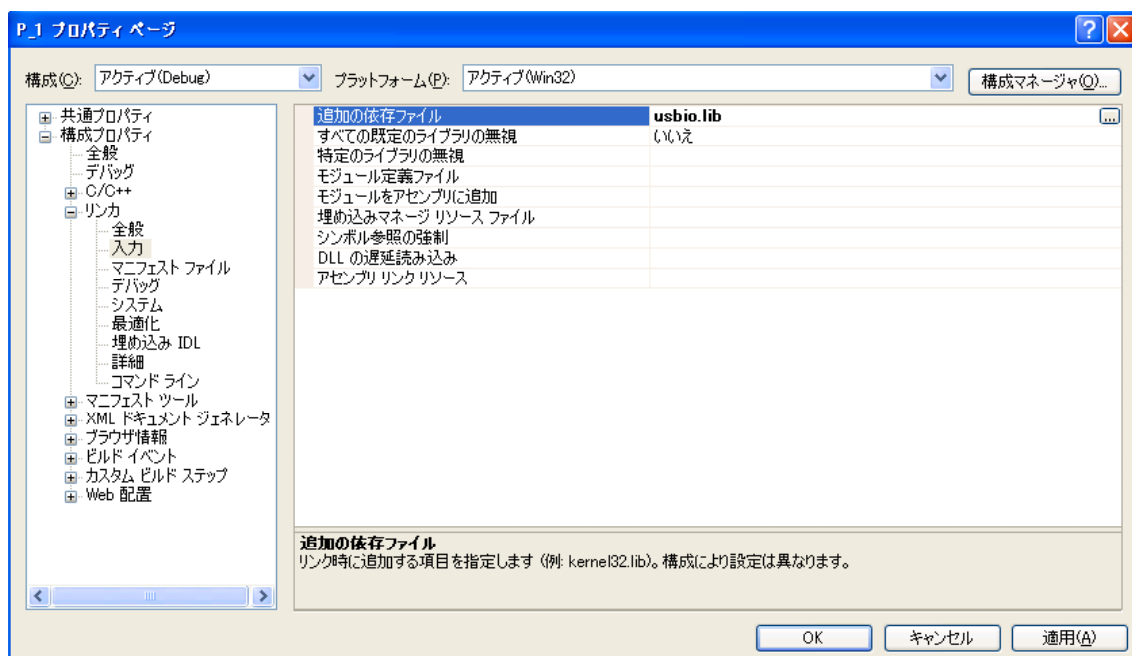


図 1 5

```
//CPP_1
//YNUSB+Y98TEST 5V系I/Oテストプログラム
//スイッチの入力とLEDの点滅

#include <windows.h>
#include <stdio.h>
#include "usbio.h"
#include "YN1.h"

void wait(void)
{
    long t = 40000000;
    while(t--);
}
```

```

}

void main(void)
{
    int ret;
    int data;
    int c = 20;
    ret = usbio_start(0);
    if(ret != -1)
        printf("OPEN¥n");
    else
    {
        printf("NOT OPEN¥n");
        ret = 1;
    }

    printf("スイッチを入力し LED に出力します。スイッチ全部 ON で次に移ります。¥n");
    data= inporta(0);
    while(data != 0xff)
    {
        data= inporta(0);
        outportb(0,data);
        outportc(0,data);
    }
    printf("LED の点滅を始めます。10回点滅で終了します。¥n");
    while(c--)
    {
        outportb(0,data);
        outportc(0,data);
        wait();
        data = ~data;
    }
    ret = usbio_end(0);
    printf("プログラムが終了します。¥n");
    printf("Enter キーを押してください。¥n");
    getchar();
}

```

リスト 1

「参考」 P1_P1 のファイル構成は図 16 のようになります。



図 1 6

YNUSBI/Oボードの40ピンコネクタの信号を使用した3.3V系のプログラムはリスト2になります。5V系との違いは次の通りです。

- (1) 入出力の関数名が違う
- (2) ビット単位で入力または出力に指定できる
- (3) 入力または出力の初期設定が必要

```
//CPP3_1
//YNUSBI/O+YN3TEST 3V系テストプログラム
//
#include <windows.h>
#include <stdio.h>
#include "usbio.h"
#include "YN1.h"

void wait(void)
{
    long t = 40000000;
    while(t--);
}

void main(void)
{
    int ret;
    int data;
    int c = 20;
    ret = pio_start(0);
    if(ret != -1)
        printf("OPEN¥n");
    else
```

```

{
    printf("NOT OPEN¥n");
    ret = 1;
}
pio0_ddr(0,0);
pio1_ddr(0,0xff);
pio2_ddr(0,0xff);
pio3_ddr(0,0xff);

printf("スイッチを入力し LED に出力します。スイッチ全部 ON で次に移ります。¥n");
data= pio0_in(0);
while(data != 0xff)
{
    data= pio0_in(0);
    pio1_out(0,data);
    pio2_out(0,data);
    pio3_out(0,data);
}
printf("LED の点滅を始めます。 1 0 回点滅で終了します。¥n");
pio0_ddr(0,0xff);
while(c--)
{
    pio0_out(0,data);
    pio1_out(0,data);
    pio2_out(0,data);
    pio3_out(0,data);

    wait();
    data = ~data;
}
ret = pio_end(0);
printf("プログラムが終了します。¥n");
printf("Enter キーを押してください。¥n");
getchar();
}

```

リスト 2