2005 年　卒業研究論文

レーザーを使った３次元スキャナの作成

㈱国立小山工業高等専門学校
電子制御工学科　金野研究室
5年　青柳信吾

Keywords: USB PIC LASER Dual-CPU OpenGL 3D-Graphics

レーザーを使った3次元スキャナの作成

CONTENTS

1.　はじめに

2.　測定原理

3.　全体の構成

4.　回路の構成

5.　作成

6.　通信用PICの仕様

7.　駆動用PICの仕様

8.　デバイスの認識

9.　ソフトウェアの目標

10.　表示用ソフトウェアの作成

11.　制御用ソフトウェアの作成

12.　実行結果とパソコンの操作方法

13.　結論

14.　今後の課題

15.　参考･引用元

APPENDIX

付録 A　ボトルを測定した測定データ

付録 B　汎用USBドライバUUSBD用の設定ファイル　UUSBD.INF

付録 C　OpenGLプロッティングツール「glOutput.exe」の入出力仕様

付録 D　「glOutout.exe」のソースコード

付録 E　デバイス用PICのソースコード（測定部）　mesure.asm

付録 F　デバイス用PICのソースコード（表示・通信部）　ASM

F.1　メインファイル　dev1pjt.asm

F.2　インクルードファイル　GETMEM.H

F.3　インクルードファイル　STDMAC.H

F.4　インクルードファイル　USBSYS.H

F.5　インクルードファイル　DSCRPT42.H

F.6　インクルードファイル　USBMAIN.ASM

付録 G　デバイス用 CPLD のソースコード　Verilog-HDL

付録 H　PC側プログラムのソースコード VBProject

H.1　フォーム Form1.frm

H.2　標準モジュール　Module1.bas

付録 I　最終発表用資料

奥付・その他

1. はじめに

近年、パソコンによる3次元表示はとても高性能になってきている。風景の描画や、人物の表現、霧や雨などの自然現象の描画ができるようになった今、パソコンによる3次元表示は実物と見間違えるほどである。しかし、このようなパソコンの高性能な表示機能を利用し、実際の物体をパソコン上で表現しようとすると問題が発生する。実際の物体をパソコン上にデータ化するための機器（以降これを3次元スキャナという）が非常に高価であるために、個人では利用できないということである。

そのため実際の物体の形状をパソコンに取り込むための装置、3次元スキャナを作成しようと考え、作成を試みた。 目標としては、物体をスキャンし物体の形状をパソコン上にデータ化する事を目標とし、これを確認できる表示システムも同時に模索する。

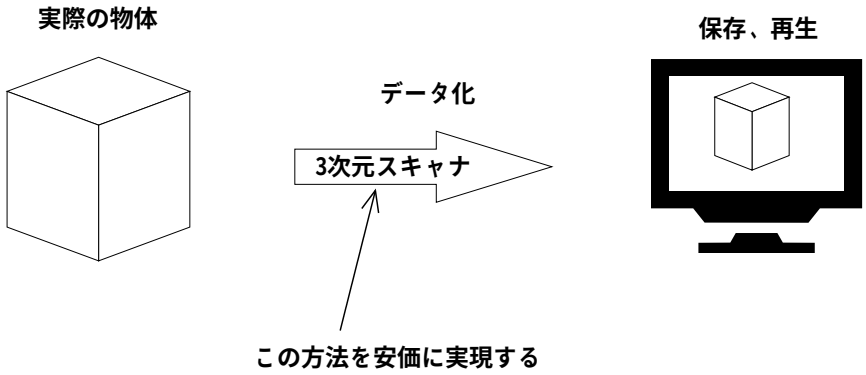


図1.1　この研究の目的

2．測定原理

物体の形状を測定するのには、次の図2.1に示す測定原理を利用する。

レーザーを測定対象物に照射すると物体表面でレーザー光は乱反射される。レーザーが乱反射された点と、受光素子、受光スリットが直線上に無い場合は受光素子に乱反射された光が入ることはない。

次に、レーザーの照射角度を変えていくと、レーザーの乱反射点と、受光スリット、受光素子が直線に並び受光素子の出力が変化する。このときのレーザーの照射角と、受光素子とレーザー光源間の距離により、受光素子と測定対象物（このときにレーザーが当たっている点が測定対象点となる）との距離を求めることができる。

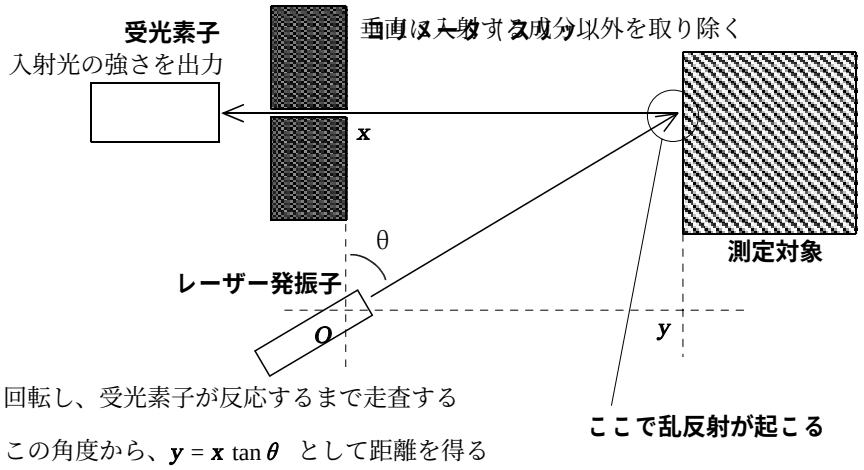


図2.1　距離測定の原理

また、今回の測定には自作のスリットを利用した。そのため、乱反射点と受光素子、受光スリットが直線に並んでいなくても、図に示すようにΔθの区間で受光素子が光を検知してしまう。これを回避するために、図2.2に示す微分法を利用した。センサー出力は、図2.2にあるようにセンサーが反応する付近では山形に変化する。この最高点がスリットが本当の線だった場合の測定点（理想的な測定点）である。そのため、センサーの出力を微分し傾きが負になる前の測定値を真の値とし、真の角度θを判定し、距離を求める。

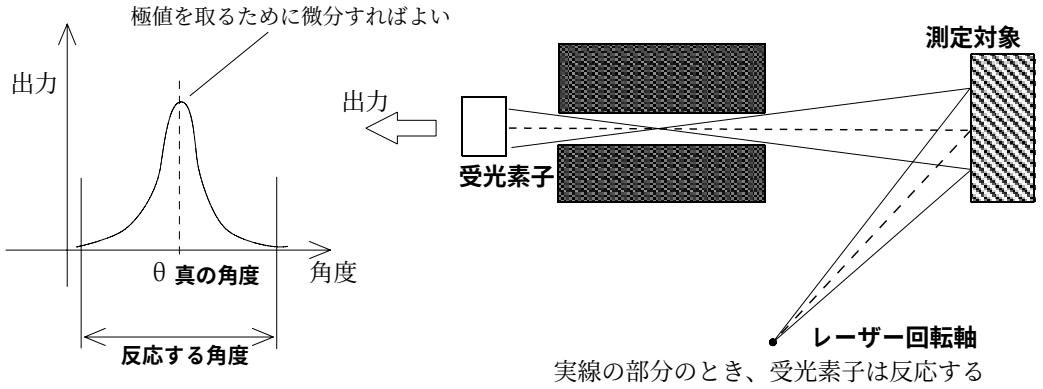


図2.2　受光部の特性向上の方法

測定対象はターンテーブルの上に乗せ、1点の距離を測定するごとにターンテーブルをシフトさせることで対象物の断面を得る。1回転分スキャンした後、受光素子、受光スリットを含めた測定ユニットを上方向にシフトさせることによって高さ方向のスキャンを行い、測定対象物を円柱座標系で取り込むことができる。

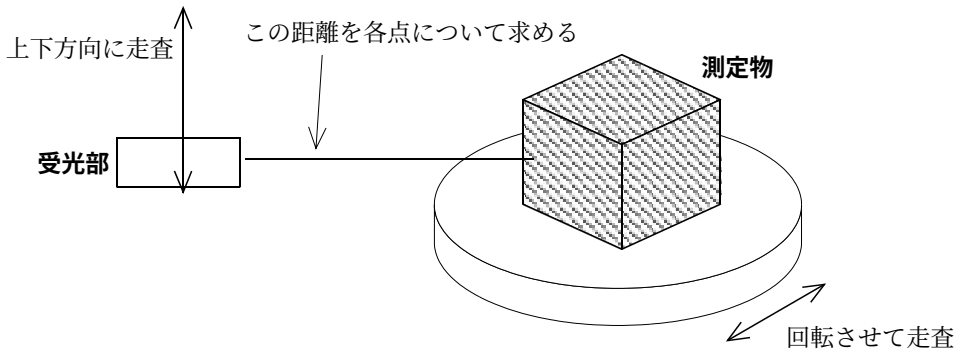


図2.3　物体の形状を得る方法

3．全体の構成

3次元スキャナの全体構成を次図に示す。3次元スキャナはスキャン機構や測定機構としての駆動機能と、ＵＳＢ通信用を行うための通信機能がある。これらは同時に動作すると効率的であるため、駆動機能と通信機能の制御部を二つに分けた。またパソコン側から操作するために、通信部はパソコンからの命令を駆動部に橋渡しする機能も持っている。

通信部はＰＩＣとUSBドライバICのUSBN9604と表示用のLCDによって構成されている。パソコンとPICのUSB通信はUSNBN9604が仲立ちすることで、PICでは作ることのできない12Mbpsの信号や、作動信号の作成解読などを可能にしている。通信部にはプログラムを作りやすいこともあり、PIC16F877を利用した。またLCDを駆動するとピンの数が足りなくなることから、LCD駆動用の端子を3本のシリアル信号線とし、シリアルパラレル変換することでPICが使うピンを節約している。

駆動部（測定機能も持つ）はPICが主となっており、PICがセンサー出力の解析や、ステッピングモータ用の基準信号作成を行っている。モータを複数個駆動させるため、PICからはステッピングモータ用の信号線は1組しか出さず、CPLDによるステッピングモータセレクタを通すことで2個のステッピングモータを駆動させている。駆動用のＰＩＣは制御は通信用ＰＩＣは制御からの命令を受けるピンが１２本、ステッピングモータの制御信号を作成するピンが5本、センサー入力など測定用の端子が4本と多くのピンを使い、また動作速度が求められることからPIC18F452（18シリーズ、40pin）の物を使った。

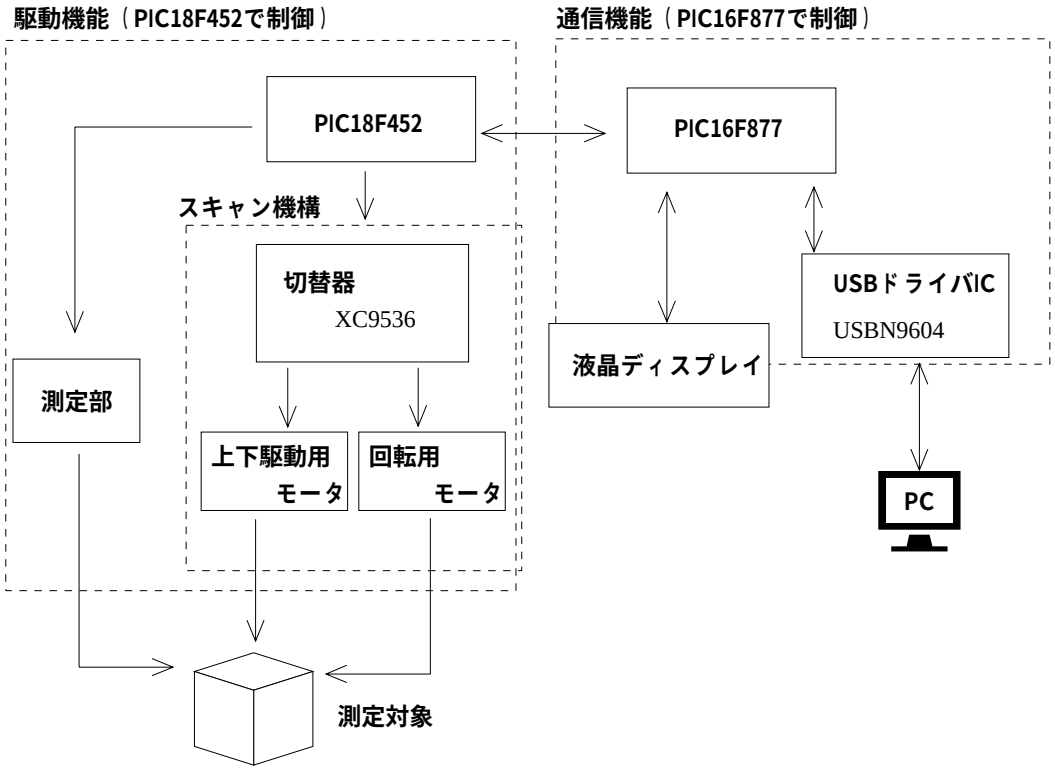


図3.1　全体の構成

4. 回路の構成

以上を元に作成した回路図を次に示す。

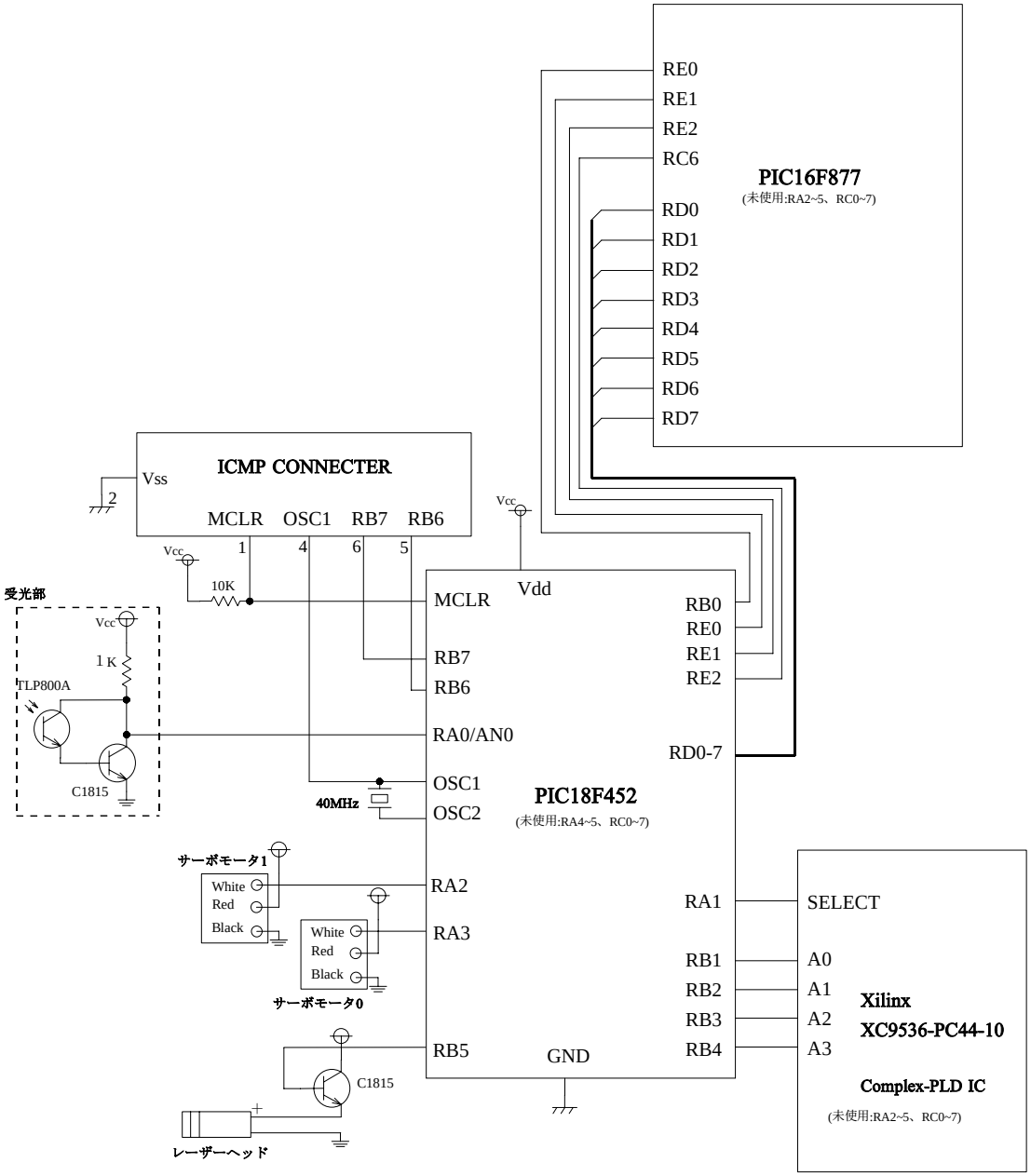


図4.1　このシステムの回路図（1／4　測定部）

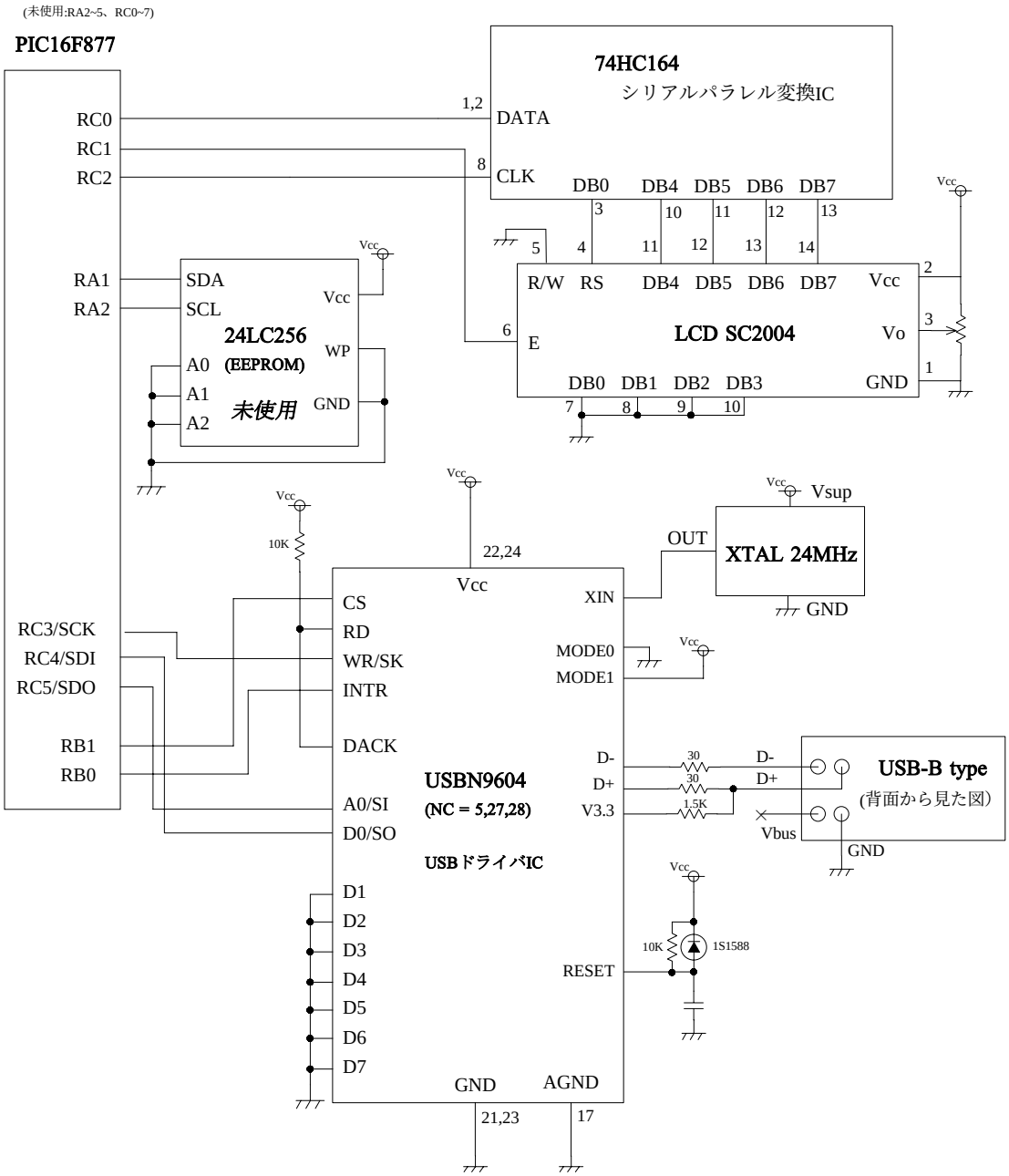


図4.2　このシステムの回路図（2／4　通信部）

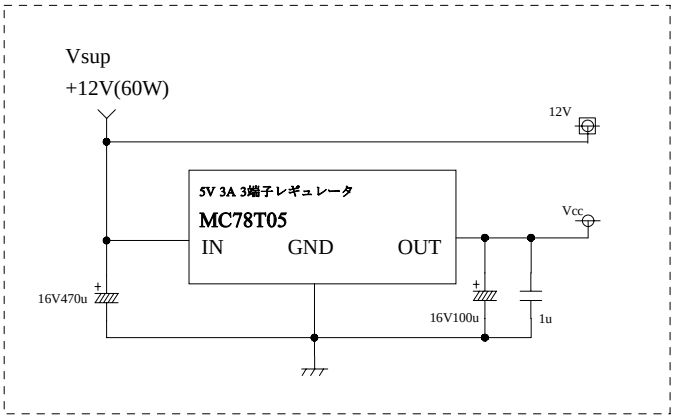


図4.3　このシステムの回路図（3／4　電源部）

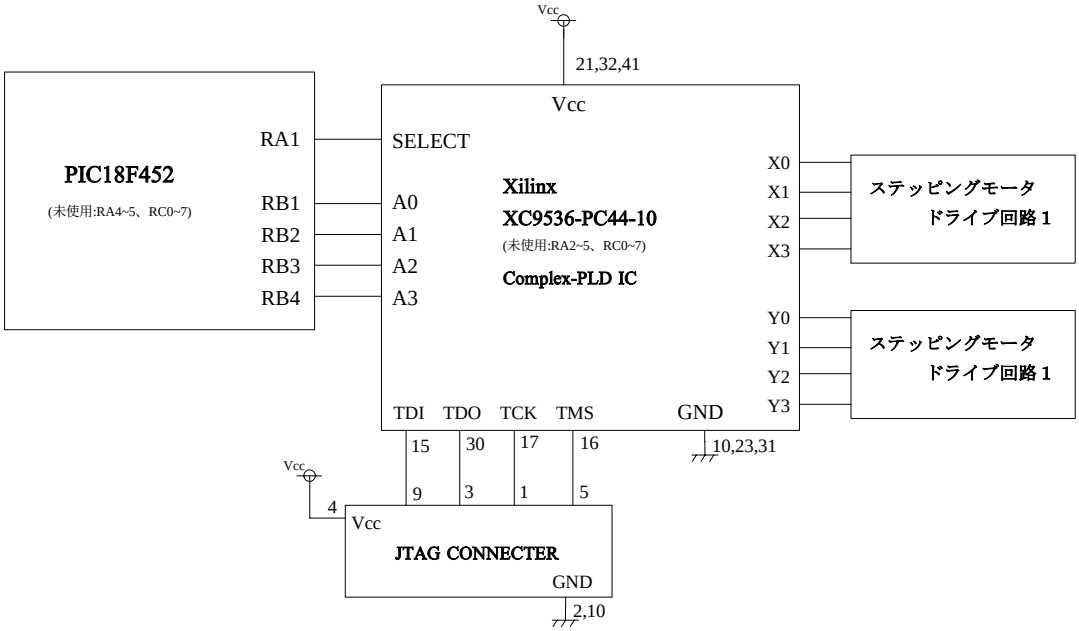


図4.4　このシステムの回路図（4／4　モータ駆動信号切り替え部）

回路作成中、通信部駆動部両方のPICがPORTDを出力に設定すると過電流によりPICが壊れてしまうのではないかと考え、PORTDの間に保護用の抵抗3kΩを入れていた。しかし、PICの入力ポートはDFFで入力を得る仕様なので、途中に保護用の抵抗を入れてしまうと、正常に動かない。実際、保護用の抵抗をいれた場合、PORTDを入力（TRISD=0xFF）にした状態でPORTD=0xFFとすると、PORTDに5[V]が出力され、抵抗の他端が0[V]になるという不思議な状況が起こっている。そのため、プログラム的に両方がOutputになることを回避し、保護用の抵抗は取り外した。

USBN9604とPICとの通信にはSSP(SPIモード)を利用している。SPIモードとは、マスタとスレーブの関係が存在す

るシリアル通信のことである。USBN9604はこの通信方法か、もしくはパラレルによる通信が可能であるが、PICを使う場合はPIC自体にSPIモジュールが組み込まれているものもあるので、SPIモジュールが組み込まれているPICを使う場合にはシリアル通信を使うことをお勧めする。パソコンからUSBによりデータがUSBNに送られてくると、USBNは外部（ここではPIC）に割り込み信号をINTR(4番ピン)より出す。これを受けた外部の回路は、USBNにどのような情報が送られてきたのかなどを問い合わせ、データの入出力に関する指示を行うことでPICとパソコンがUSB通信ができるようになる、という仕組みである。そのためUSBNのINTRピンはPICのPORTBの0番ピンと接続されており、USBNBのINTR信号によってPICがRB0割り込みを開始するようになっている。

測定に使う受光部は赤外フォトトランジスタを利用している。このフォトトランジスタはフォトインタラプタ（マウスのホイールの回転検出などに使う物）を分解したものなので、感度はそれほど良い物ではない。そのため、2SC1815とダーリントン接続とすることで感度を上げている。これにより、安価で非常にすぐれたセンサーとして使うことができる。また、測定用のレーザーヘッドは電流を多く使用するために、PICの出力ピンからの電流供給では駆動することが出来ない。そのため、トランジスタのスイッチングを利用して電流を確保した。

回転台操作用のステッピングモータを駆動させる回路は、秋月電商で販売されているNPNダーリントントランジスタアレイを利用した。これは大電力のダーリントントランジスタ4素子が内蔵されたパッケージであり保護抵抗や逆起電圧対策のダイオードなどが含まれたものである。そのため、このトランジスタアレイをそのまま利用すればよいような回路としてある。

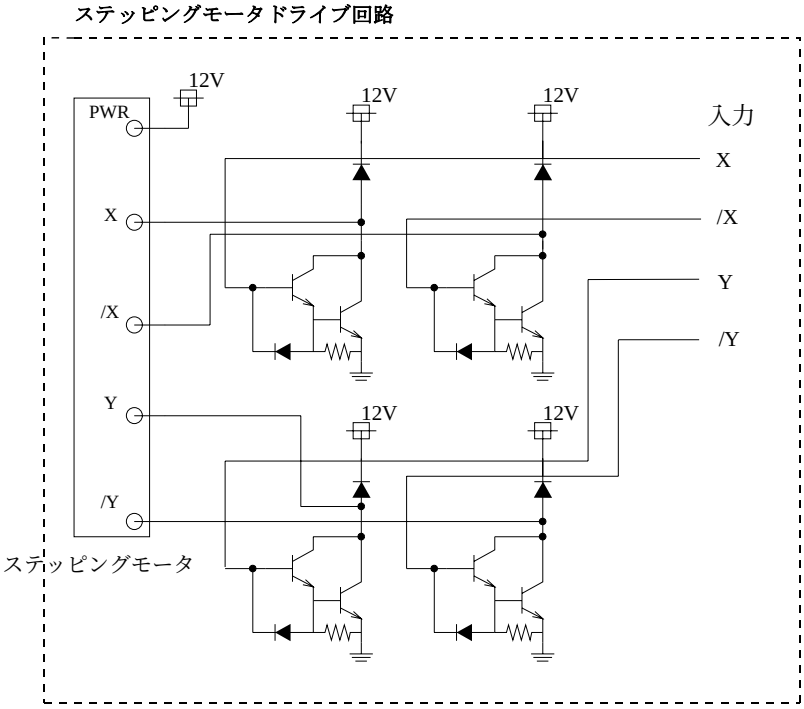


図4.5　ステッピングモータドライブ回路

また、測定部を上下させるための機構としてインクジェットプリンタのヘッド機構を利用した。このヘッドに使われているモーターは4端子のステッピングモーター（電源が無いタイプのもの）である。これを制御するために、ステッ

ピングモータドライバLB1845を利用した。これは大電力用のステッピングモータドライバである。LB1845はステッピングモータに流す電流の制限や、制御用の信号を出力する間隔などを周辺回路のパラメータにより決めることができる。今回は出力を常にEnableにし、駆動用のステッピングパルスはPICから回転台駆動用に出力しているものと同じ物をLB1845に入力している。

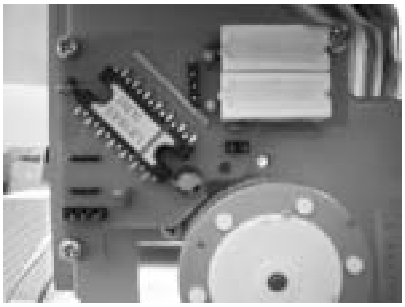


図4.6　上下機構専用に設けたモータドライブ回路

5．作成

図に示した回路図よりサンハヤト感光基盤（紙フェノールタイプ K12）より基盤を作成した。これに使用した部品を表に示す。

表5.1　3次元スキャナに使用した部品一覧

回路図中記号	名称	品名･型番	数量	単価
U1	PIC	PIC18F452	1	600
U2	PIC	PIC16F877	1	500
U3	CPLD	XC9636-44c-10	1	650
U4	USBドライバIC	UUSBD9604	1	500
X1	セラミック発振子	40MHz	1	60
X2	セラミック発振子	20MHz	1	60
X3	水晶発振子	24MHz	1	60
U5	3端子レギュレータ	78T05	1	300
U6	トランジスタアレイ	FT5754M	1	200
U7	モータドライバ	LB1845	1	400
U8	大電力用トランジスタ	2SD1061	1	10
U9	シリアルパラレル変換	74HC164	1	40
R	抵抗	30Ω 1/4W	2	2
R	抵抗	1.5kΩ 1/4W	1	1
R	抵抗	1kΩ 1/4W	8	8
R	抵抗	10kΩ 1/4W	5	5
R	セメント抵抗	0.2Ω 5W	2	100
VR	可変抵抗	10kΩ	1	100
C	電解コンデンサ	16V 330uF	1	10
C	電解コンデンサ	10V 100uF	2	20
C	電解コンデンサ	6.3V 100uF	1	10
C	積層セラミック	0.1uF	2	20
C	セラミックコンデンサ	100pF	2	100
D	小信号ダイオード	1SS1588	1	5
Tr1	トランジスタ	2SC1815	2	20
LED	LED	赤色φ3	3	12
J1	USBレセプタクル	タイプB	1	50
J2	DCジャックφ3.5	DCジャックφ3.5	1	30
	ピンヘッダ	1列	81	400
	ピンヘッダ	2列	19	200
	ピンフレーム	1列	31	400
S1	レーザーヘッド	650nm赤 5mW以下	1	300
S2	フォトトランジスタ	TLP800A	1	200
LCD	液晶表示器	LCD2004BL	1	2000

M1	サーボモータ	FUTABA S3001	1	3000
M2	ステッピングモータ	TS3103N124	1	450
	スイッチングアダプタ	12V 5A	1	1800
M3	インクジェットヘッド	STEPタイプ	1	100
SW	DIPスイッチ	1回路2接点	1	50
SW	マイクロスイッチ	1回路2接点	1	300
合計金額				13073

PICは通信部と駆動部二つのPICがそれぞれのPORTDによって結合しているために、基盤上には横に配置するのではなく、PICの上にPICが乗った構造にしたほうが、基盤が作成しやすく、また回路面積も小さく配線経路も短くすることができる。そのため、完成品をみてわかるようにPICが重なって配置されるように、通信用の基盤の上に駆動用回路を乗せている。

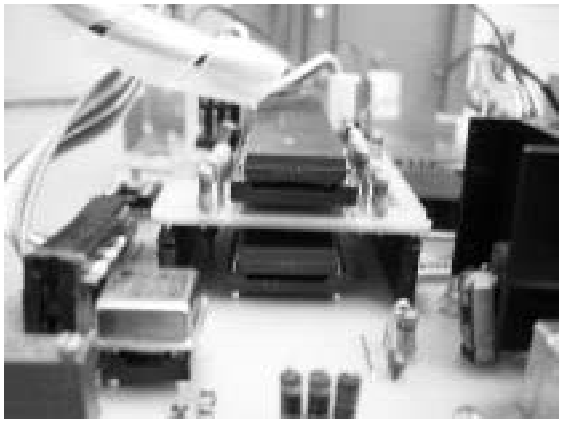


図5.1　PICを重ねることで回路面積を縮小する

PICやCPLDを基盤上に配置したままで書き換えが可能なように、CPLDからはJTAG端子をPICからはICMP端子を出している。JTAG端子とはCPLDで書き換え専用に設けられた端子であり、ここにパソコンから出したJTAG信号を入力することによりCPLDの書き換えが可能となる。JTAGポートのピン配置図は統一されており、Xilinx社のダウンロードケーブル（パソコンからCPLDにJTAGコードを転送するためのケーブル）の説明書にピン配置が記載されている。これに合わせて作成すればよい。

PICはプログラムメモリの書き換えとしてPIC18F452、PIC16F877の場合RB7、RB6とMCLR、OSCを利用する。これらをPIC書き換え専用の端子として基盤上に配置したのがICMP方式によるPICの書き換えである。PICをボード上に配置したまま、これらのピンに書き込み用の信号を加えるとPICを基盤からはずすことなく書き換えることが可能となる。実際、ICMP方式によるPICの書き込みを利用してみたところ、PICを書き換えの度にはずす手間がなくは非常に便利でPIC自体の損傷も少なくなったので良いのではないかと感じている。しかし、ICMP端子は作成者により全く仕様が異なることや、MicroChip社が定めるモジュラージャック利用ではモジュラージャックの値段が高いという問題点もあり、汎用性に欠ける。もしICMP方式を利用できる基盤をつくるのならば、しっかりと仕様を決め、それに合わせて基盤を作成するのが良いだろう。

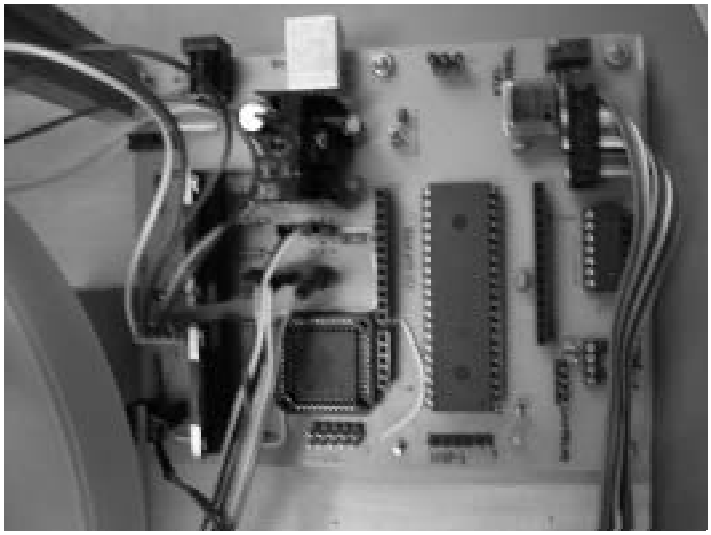


図5.2　PICとCPLDの書き込み用端子　(PIC、CPLDの下のピン)

今回、作成中に USB が認識しないトラブルがあった。そのため、USBN9604 の問題であることも考え USBN の交換も行ったが、実のところ原因はパソコン側にあった。そのため、USBN9604 を交換した際に傷つけた箇所が基盤上に多々あるが、これらは交換用についた傷であり、深い意味は全くない。また USBN を交換してわかったことだが USBN は熱にも強く、破壊、ねじれにも強い。そのため、回路のトラブルがあった場合 USBN 自体が壊れていることは最後に確認すれば十分である。

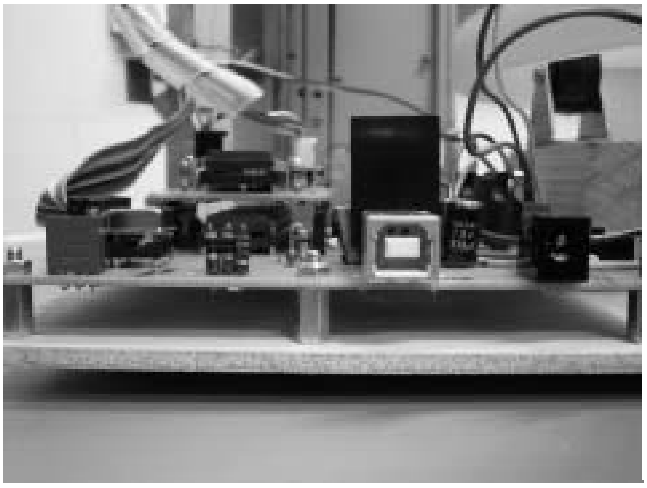


図5.3　USBポートとUSB回路 (USBポート左、抵抗が3本並んでいる箇所の基盤表裏)

測定用のレーザー光源として、秋月電商で販売されていた発光波長650nmの可視光赤色レーザーを利用した。しかし、可視光のフォトトランジスタは高いことや、どうしても可視光は日常的に存在することもあり、赤外領域のフォトトランジスタ (感度ピーク800nm) を選択した。ここでレーザーの発光ピークとフォトトランジスタの受光ピークが違うのが疑問に思える方もいるかもしれない。しかし受光感度800nmのフォトトランジスタは実際800nmのみが測定可能というわけではなく800nm前後も測定が可能であるため、この二つの素子の組み合わせとして利用している。また、レーザー光源として赤外線を利用することも考えたが赤外線レーザーが高いことと、可視光にすることでどこにレーザーは照射されているのかを確認する目的もあり、可視光のレーザーを利用している。

図5.4　測定部　(左:レーザー発振子　右:受光部　下:回転台)

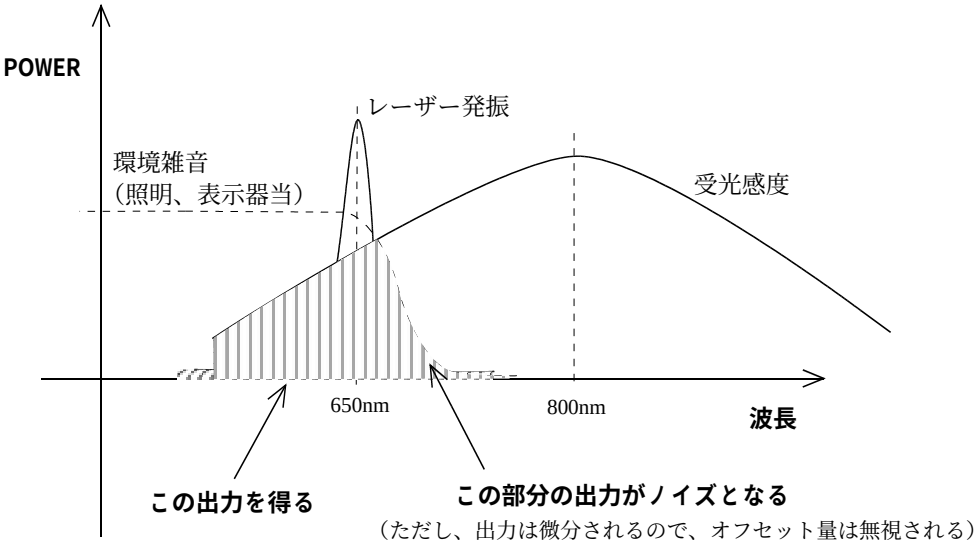


図5.5　発光波長と受光波長

6．通信用PICの仕様

通信用ＰＩＣは、パソコンからＵＳＢを通じて命令コマンドを受け取り、それに合わせた処理を行い、パソコンにデータを返すようになっている。パソコンとＰＩＣを結ぶＵＳＢのパイプ構成はＵＳＢの基本パイプ7本を次の表のような利用用途として定めている。

表6.1　今回定めたUSBパイプの仕様

Pipe	Input / Output	バッファサイズ	利用用途	処理ルーチンラベル
-	双方向	8バイト	システム接続	USBMAIN.ASM 内
1	Input	8バイト	命令受付	DO_RX1
2	Output	8バイト	予備	USB_TX1
3	Input	8バイト	予備	DO_RX2
4	Output	8バイト	予備	USB_TX2
5	Input	16バイト	予備	DO_RX2
6	Output	16バイト	データ返送	USB_TX3

デバイスがパソコンと接続されたとき、まずパソコン側からデバイス情報の要求が行われる。これに対してPICはデバイスの製造番号など、デバイスを認識するための情報を送り返す。これらのシステム的処理はUSBPipe0で行われることになっているため、表6.1に示すようにPipe0はシステム用に確保した。

次にスキャナとしての機能を実現する通信として、Pipe1（PCが送信したデータをPICが受信するパイプ）とPipe6（PICが送信したデータをPCが受信するパイプ）を利用した。まずPipe1により、PCからの命令を受信する。このデータをPICが受け取るとLCD画面上にデータを表示し、さらに送られてきたデータの命令部分を抜き出し駆動用のPICへと送信する。その後駆動用のPICが応答を返すまで待機し、駆動用PICの処理結果を受け取る。最後に、駆動用PICから受け取った処理結果をUSBを通じてPCに送信し、1回の通信が完了する。

以上の流れをフロー図として図に示す。

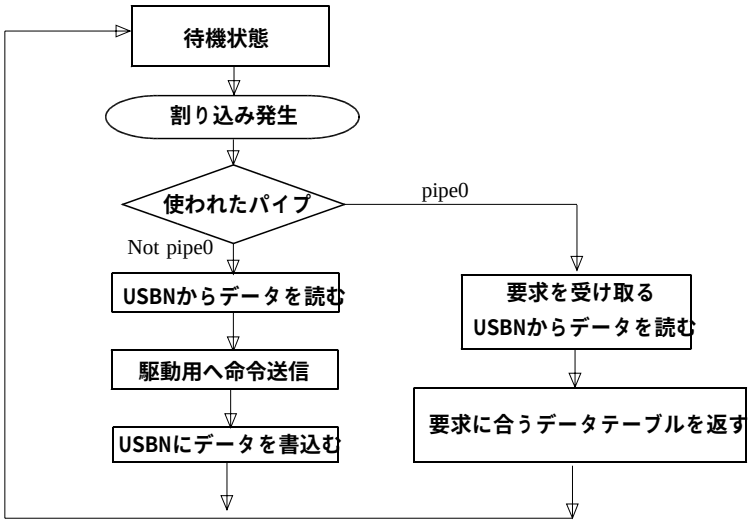


図6.1　PIC16F877のフローチャート（簡略）

7．駆動用PICの仕様

駆動用のPICは通信用PICからの命令を受け取り、それに合わせた処理を行い、通信用PICへ処理結果を返す。駆動用PICと外部回路（駆動部以外の回路）を接続するピン配置は次の表のようになっている。

表7.1　駆動用PICの入出力仕様

ピン	方向（※）	説明
PORTB 0	入力	命令クロック端子 データ入出力端子に命令をセットし、このピンを立ち上げると命令を受け付ける。BUSYが1の状態では反応しない。 また、このピンが1である場合、データ入出力端子から出力されているデータはホールドされる。
PORTE 0	出力	BUSYフラグ 駆動用PICが処理を行っている間、ここが1になる。
PORTE 1	－	PORTD方向設定端子（現在未利用）
PORTE 2	－	PORTD方向設定端子（現在未利用）
PORTD	双方向	データ入出力端子 BUSYフラグが1、命令クロックが1の間は出力に、 それ以外のときは入力ピンとして動作する。

※駆動用PICからみたデータの方向

また駆動用PICに渡す命令一覧を次表に示す。

表7.2　駆動用PICの命令一覧

命令（Hex）	Output（Hex）	動作	処理ルーチンラベル
0x01	測定値	測定を行い、測定結果を返す	STANDBYLP01
0x02	0	回転台を順方向に回転させる	FUNC01
0x03	0	回転台を逆方向に回転させる	FUNC02
0x04	0 上げられない場合 1	測定部を上方向に移動させる	FUNC03
0x05	0	測定部を下方向に移動させる	FUNC04
0x06	0	初期化	FUNC05

駆動用PICのフロー図を次図に示す。

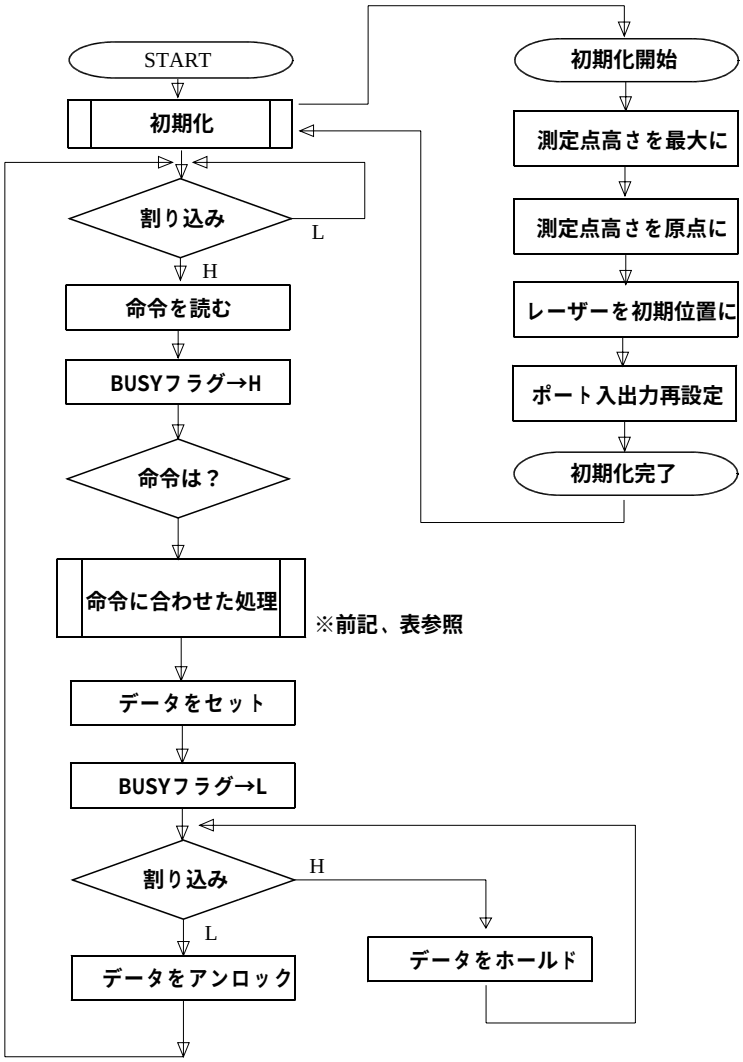


図7.1　PIC18F452のフローチャート

8. デバイスの認識

完成したデバイスをパソコンとＵＳＢケーブルで接続し、デバイスの電源をＯＮにすると、次のような画面がでて機器が接続された事がわかる。

図8.1　ＵＳＢ機器の認識画面

その後、実際にパソコンに周辺機器をインストールするのと同手順で機器のインストールを行う。機器のインストールは次の通りに行う。まず、新しいハードウェアの検索ウィザードが表示される（図10.2）ので「**次へ**」を押す。

図8.2　ハードウェアの検出

次に、「**デバイスに最適なドライバを検索する（推奨）**」を選択し、「**次へ**」を押す。

図8.3　ハードウェアの検出

ドライバファイルの特定画面が表示されるので、「**場所を指定**」のチェックボックスのみにチェックを入れ、「**次へ**」を押す。

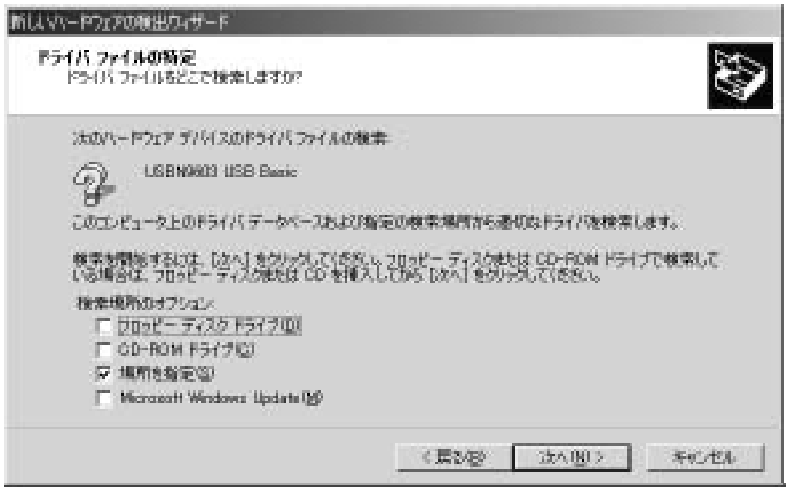


図8.4　ドライバファイルの特定

製造元が指定したドライバがどこにあるのかを聞かれるので、この画面で「**参照**」をクリックする。

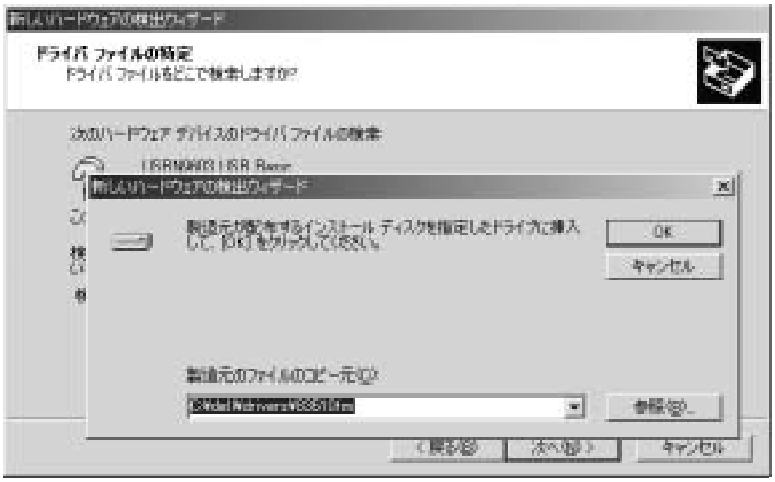


図8.5　ドライバファイルの位置の指定

このデバイスドライバを求められるウィンドウでは、このようにuusbd.sysとuusbd.infのデバイスドライバが含まれているフォルダを指定すればよい。

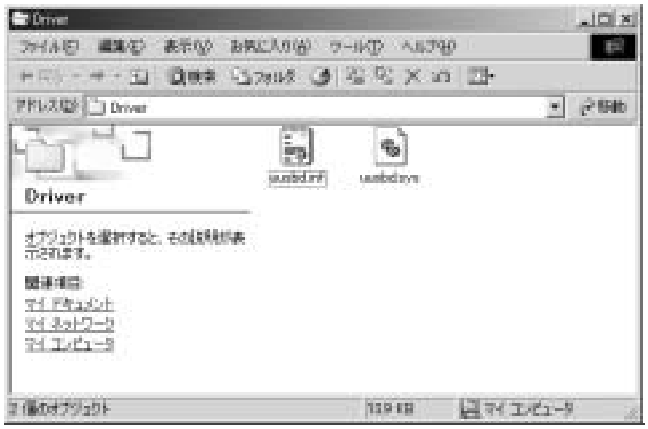


図8.6　ドライバファイルが含まれているフォルダ　(例)

参照画面でこのように表示されるので、上記のようなフォルダを開き、そこに含まれている、UUSBD.INFを指定(セットアップ情報ファイル)して、「開く」を押す。

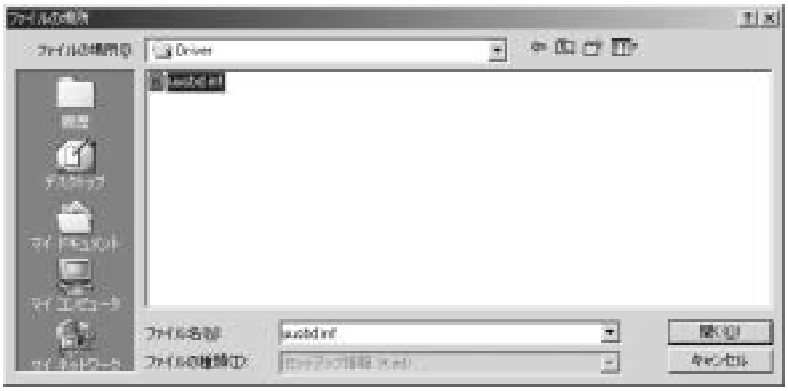


図8.7　場所を指定する画面

これにより、ドライバファイルの位置指定(図10.5)のテキストボックスにセットアップファイルのパスがかかれるので、「OK」をクリックする。

ドライバファイルが改めて検索され、USBの機器名称と、USBのドライバファイルへのパスが図10.8のように表示される。これらのパスと機器名称が正しければ、「次へ」をクリックする。

もしもこの画面が表示されない場合は、デバイスドライバ内の情報が間違っている(uusbd.inf内に記述されているベンダーコード、プロダクトコードと機器が発行したベンダーコード、プロダクトコードが合致しない)可能性があるので、その点を見直すようにしてほしい。

図8.8　場所を指定する画面

こうして、ハードウェアのインストールが次のような画面がでて、完了となる。「完了」をクリックすればよい。

図8.9　ハードウェアドライバのインストール完了画面

こうしてハードウェアの認証は完了する。一度だけインストールしてしまえば次からの接続のときはこのようなインストールは必要無い。また、機器接続中はタスクバー上に図10.10の右下の記号が表示され、機器が接続されていることがわかる。

図8.10　ハードウェア接続中のアイコン

9. ソフトウェアの目標

ソフトウェアを開発するにあたり、大まかな仕様として次のような物にする。

・操作性が良い物
・パソコンの利点である高性能は3次元表示を生かすこと
・研究用のものなので、ソフトウェアの機能更新が容易にできるもの

そのため、ソフトウェアを表示部、処理部の二つに分け、別々のソフトとして操作性、表示性能を向上させていく事にし、処理用のソフトウェアから表示用のソフトウェアをすぐに呼び出せる仕様にすることで操作性も向上させるようにする。構想としては、次図に示す通りである。

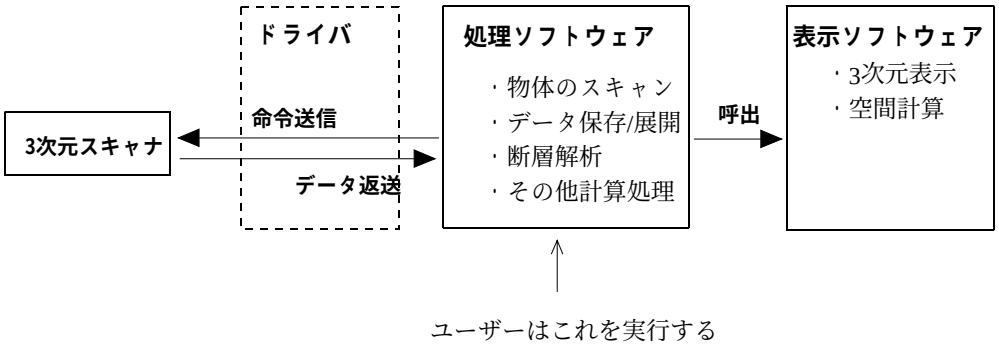


図9.1　ソフトウェアの構成

10．表示用ソフトウェアの作成

表示用のソフトウェアは、3次元スキャナのデバイスにより得られた3次元データを有効に使うために、視点の回転、透過などの表示が行えるものにする。

そこで今回はSilicon Graphics社の開発した3次元グラフィックスに関する機能を充実させたプログラムインターフェースであるOpenGLを利用した。OpenGLは現在Windows標準となっている表示機能の一つであり、このインターフェースを使いプログラムすることによって立体の投影などの計算量が多くなる3次元表示をＯＳが、対応するハードウェアがパソコンに接続されている場合（RadeonやGeForce等の高性能なグラフィックボードなど）はハードウェアによる表示計算を行い、高速に描画することを可能にするものである。今回は、VisualC++6.0を使い、「Win32 Console Apprication」からOpenGLを利用した。

また、OpenGLはそのまま利用すると命令も多くなかなか扱いにくいため、今回はOpenGLを使いやすくするための専用ライブラリ「GLUT」も利用した。GLUTはプログラム作成に使うであろう大まかな機能を使いやすくまとめた関数群である。

GLUTの使い方は次図のように扱う。

まずはじめにGLUTを「glutInit」により初期化する。これによりglutが利用できるようになる。次に、表示部の設定を行い、ウィンドウサイズの設定などを行った後、glutCreateWindow関数を使い、ウィンドウを作成する。その後、マウスの処理を書いた関数名、キーボードの処理を書いた関数名、再描画用の処理を書いた関数名など実際のアプリケーションに必要な処理を記述した関数を「glutXXXXFunc」によって登録する（XXXXは個別に名前がある。詳しくはソース参照のこと）。最後に、描画にシェーディングを利用するなどの描画に関する設定を有効にしていきOpenGLが利用できる。初期状態ではすべてのオプションはDisableとなっているために、使う機能だけをEnableにすればよい。以上の流れでOpenGLによる3次元描画を扱うことができる。

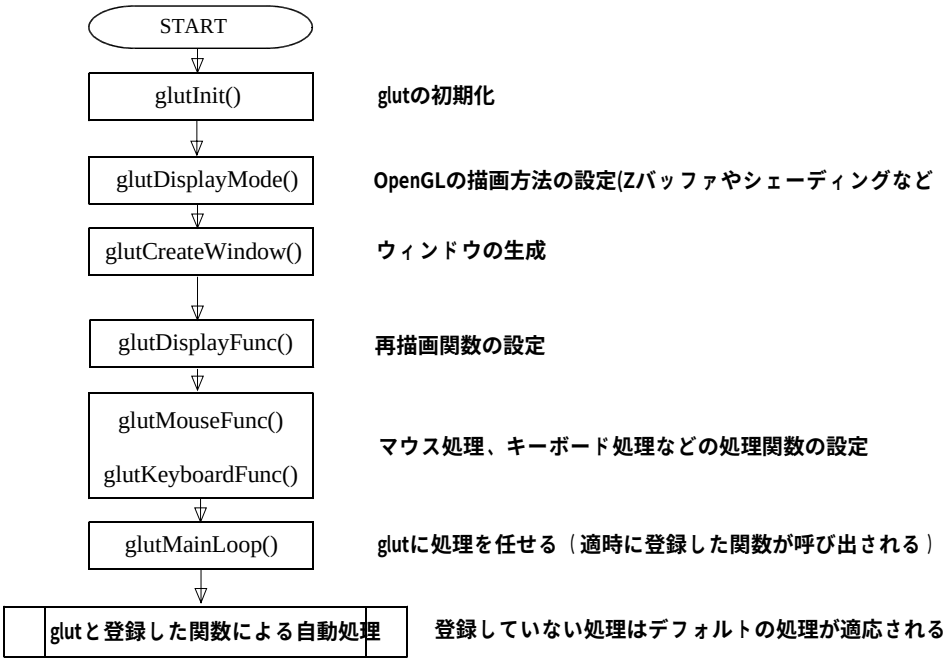


図10.1　glutを利用するための流れ

実際のソースは付録Dの通りであり、詳しい動作などについてはコメントを参照していただきたい。
表示用ソフトウェアのフロー図は図に示す。

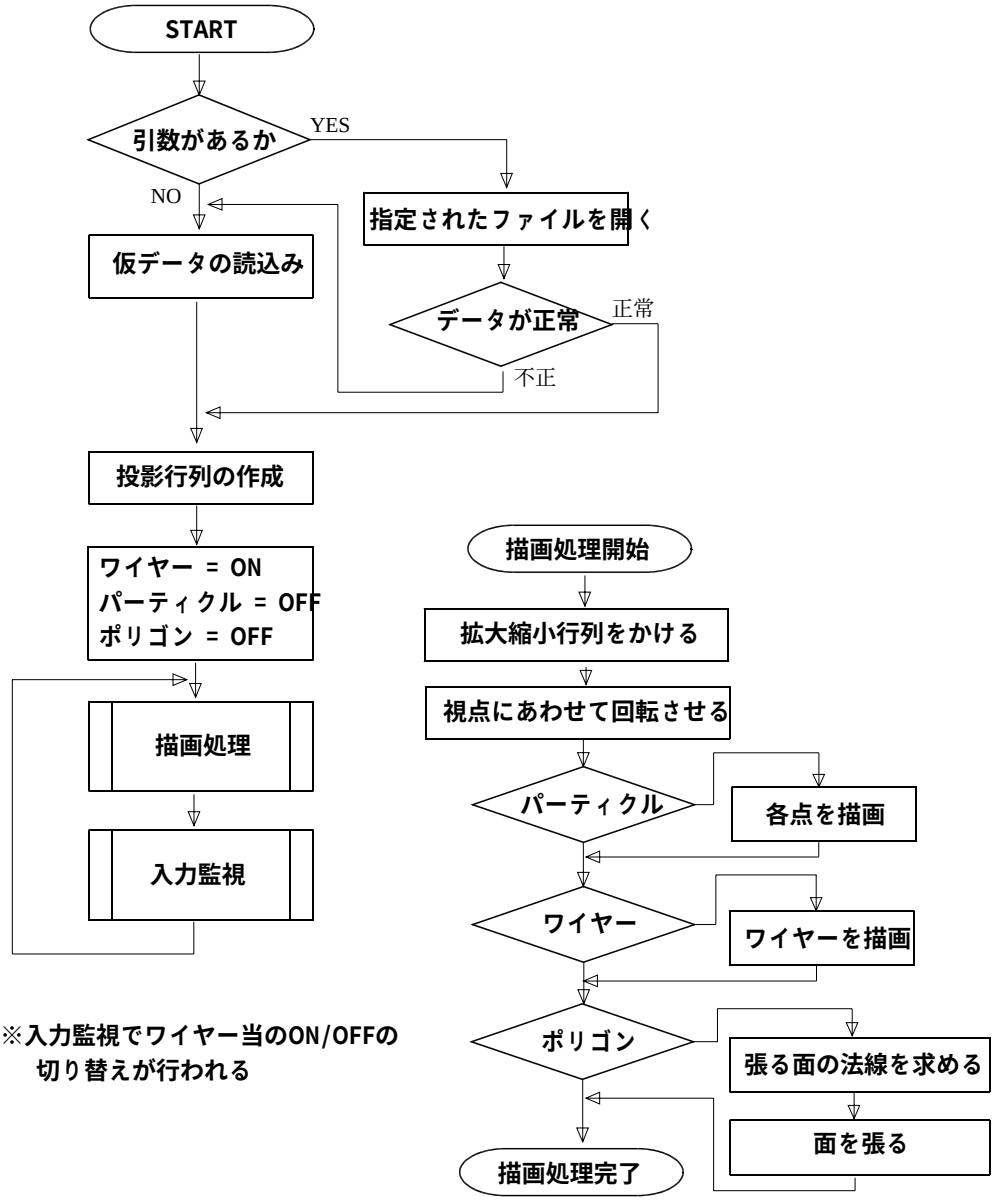


図10.2　表示用ソフトウェアのフローチャート

11. 制御用ソフトウェアの作成

制御用のソフトウェアは開発しやすい事など考えて VisualBasic により開発した。制御用ソフトウェアは GUI から受けた命令により、デバイスとの通信、外部表示プログラム（１２節で紹介したもの）へのデータの送信、データをファイルに保存するなどの処理を行うものとする。これを行うための状態遷移図を図に示す。

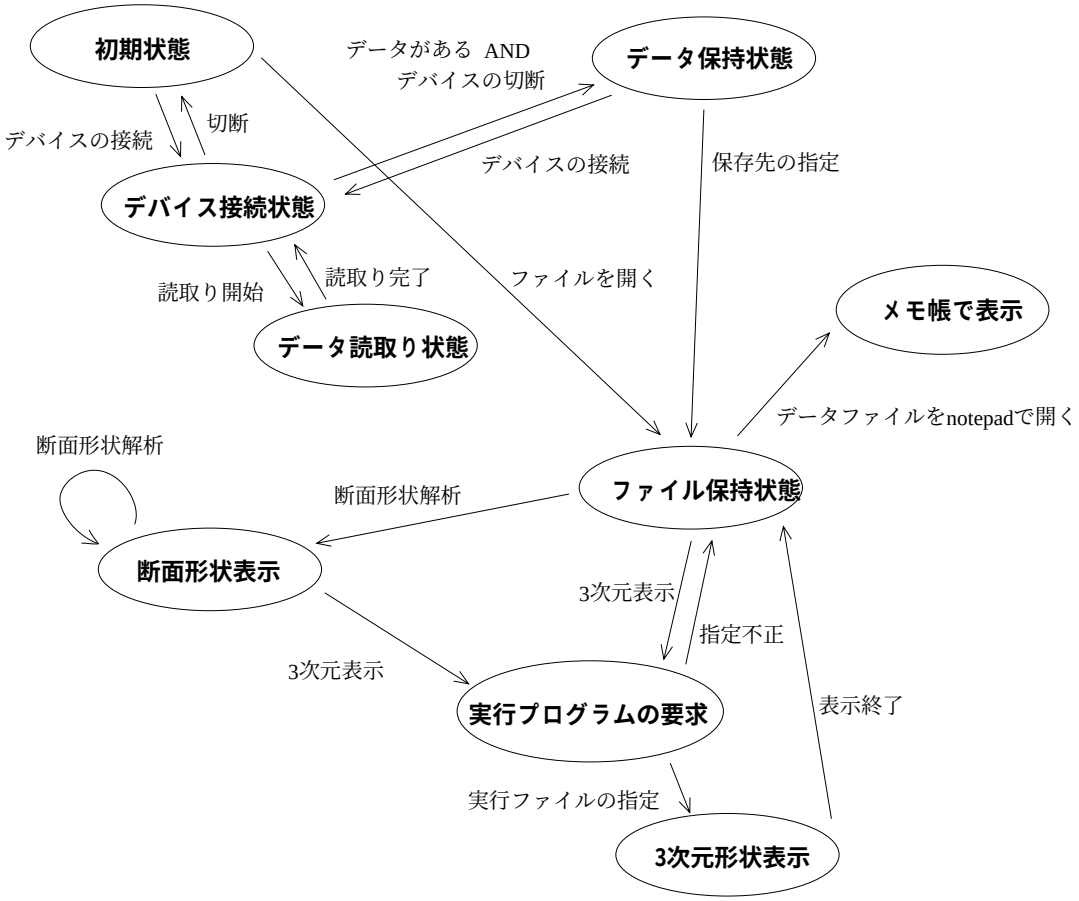


図11.1　制御用ソフトウェアの状態遷移図（概略）

12. 実行結果とパソコンの操作方法

実際の実行画面を下図に示す。

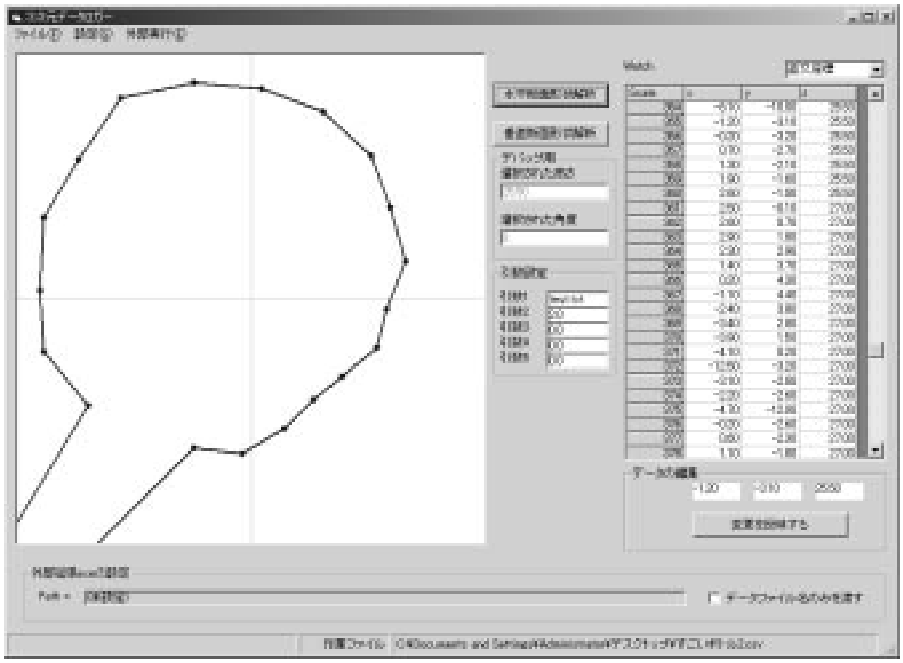


図12.1　実際にソフトを実行している画面 (1)

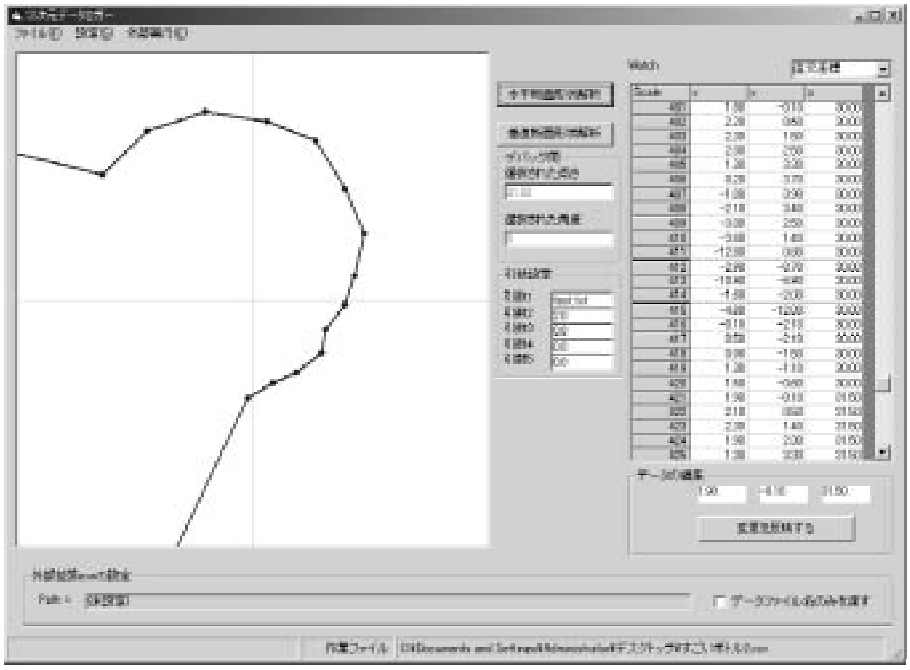


図12.2　実際にソフトを実行している画面 (1)

操作方法を以下で説明する。まず、測定器をパソコンに接続する。この状態で、パソコン側プログラム「**VBProject.exe**」を起動する。起動すると、次のような画面が表示される。

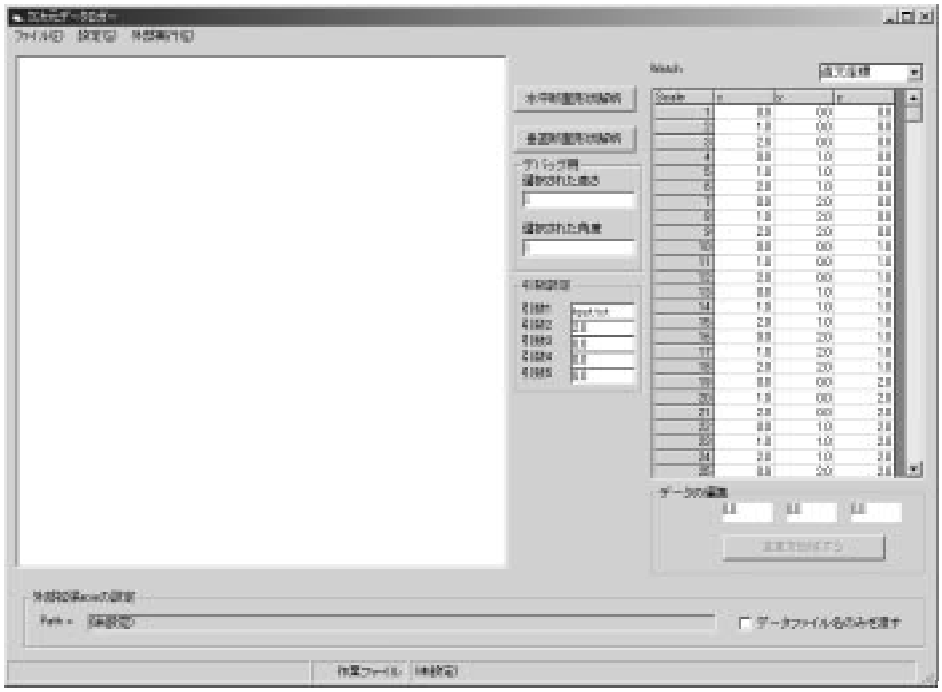


図12.3　VBPojectを起動した状態

まずはじめに、表示用のプログラムの登録を行う。外部拡張 exe の設定というフレーム内の「**(未設定)**」と書かれているテキストボックスを**ダブルクリック**すると、外部拡張 exe を選択する画面が表示される。そこで、「**glOutput.exe**」を選択し、「**開く**」を押す

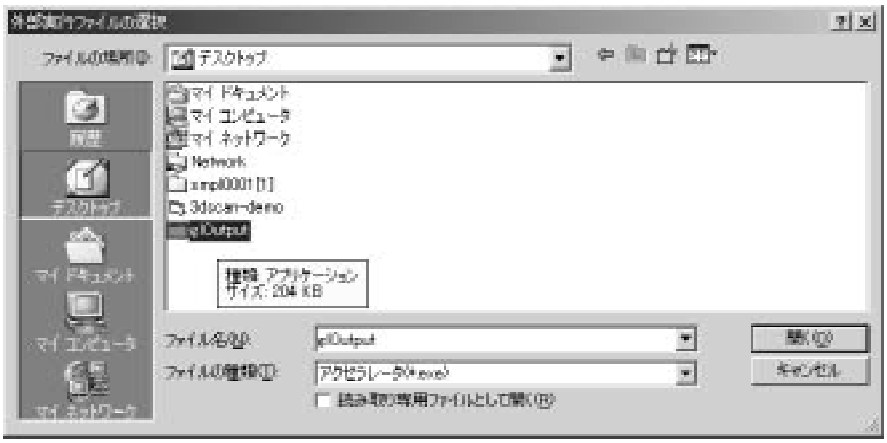


図12.4　アクセラレータ実行ファイルの設定

以上で初期設定は終了である。測定機器が接続されていることを確認し、メニューの「**ファイル**」ー「**デバイスを接続する**」をクリックし、デバイスを接続する。

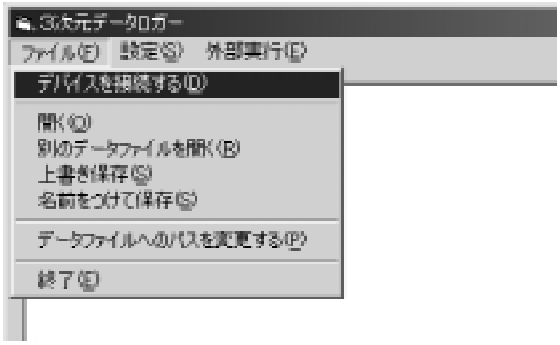


図12.5　デバイスの接続

すると、次のようにフォームが変更される。この状態になれば、プログラムが正しく機器を認識していることになる。この状態で、**測定を行いたい段の数**を指定して、「**測定開始**」を押せば測定が開始される。測定にかかる時間は１点あたり３～６秒であり、１周（１段）測定を行うのに１分～２分かかる。**測定中はできるだけ、パソコンにふれないようにしていただきたい。**

図12.6　測定画面

測定が完了したら、切断ボタンを押すことで再びデータ画面（起動状態）に戻る。最後に、「**外部実行**」ー「**外部実行アプリケーションの起動**」を押す。

図12.7　表示の実行

データファイルを保存する先を聞かれるので適当な位置に、適当な名前を付けて保存する。ここで保存したデータが表示用のプログラムに渡される。これにより、測定した物体が画面上に次のように表示される。

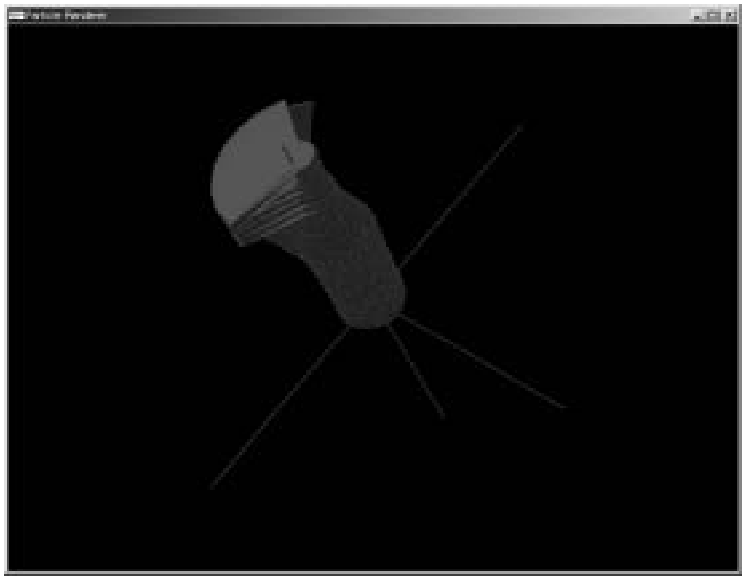


図12.8　3次元測定データの表示

データの断面形状の解析を行いたい場合も説明する。

VBProjectの右に出ているデータテーブルからスライスしたい断面の高さを指定する。高さ0.3のところでスライスしたい場合は高さ0.3が含まれているデータセルをクリックする。その後、「水平断面形状解析」をクリックすれば、次のような表示になり水平断面をみることができる。

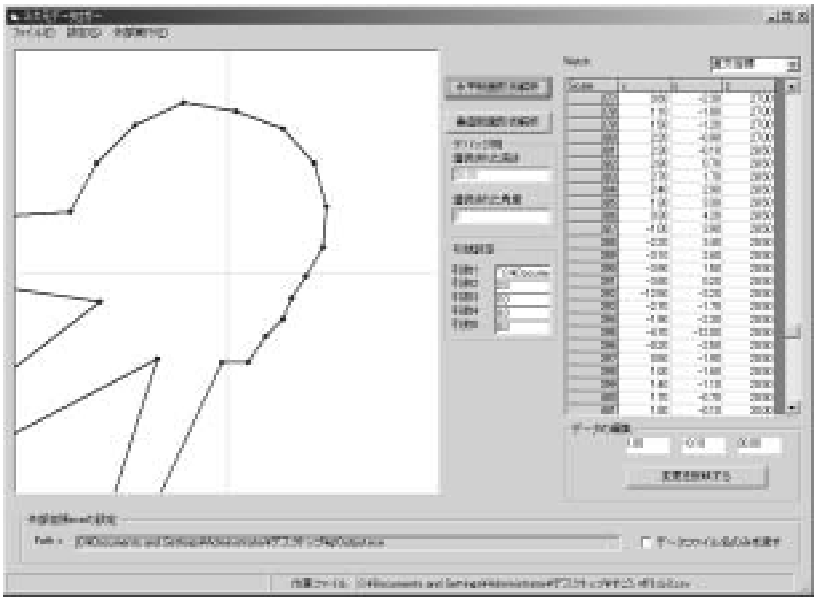


図12.9　断面形状の表示

13．結論

パソコンとUSBで接続する機器（3次元スキャナ）を作成し、節14で示すように、実際に物体の形状を読み取り、表示することができた。また、この3次元スキャナは13000円程度で作成することができた。一般的なフラットヘッドスキャナ（写真などを取り込むもの）は1万円～4万円であることから、比較的安価に実現できたと考えており、安価に実現するという目標も達成できた。

しかし、節16で詳しく述べるが、現状では形状の測定に若干の問題を抱えている。

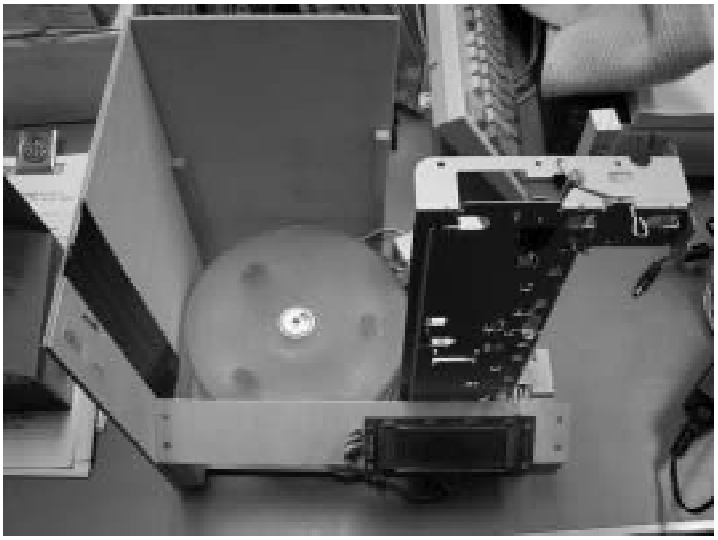
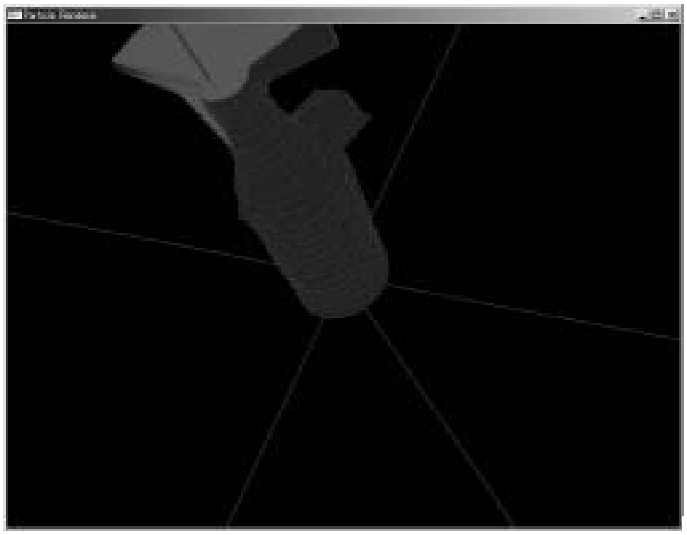


図13.1　完成した3次元スキャナ

14. 今後の課題

作成したスキャナで取り込んだデータと実際の形状を図に示す。見比べるとわかるように、物体の先端（細くなっている部分）を正確にデータ化できていないことがわかる。



(a)　パソコン上で再生した形状

(b)　実際の物体

図14.1　形状の比較

このような測定失敗が発生する理由として、測定原理の問題が考えられる。レーザーを物体に照射し、その反射を測定する現在の方法では反射が弱い面に対しての測定ができない。例えば、図に示すようなレーザーの入射線が測定対象面とほぼ水平な場合や、直線方向の反射が著しく低下する面の交差部分、透明な対象物に対しての測定は不可能となる。

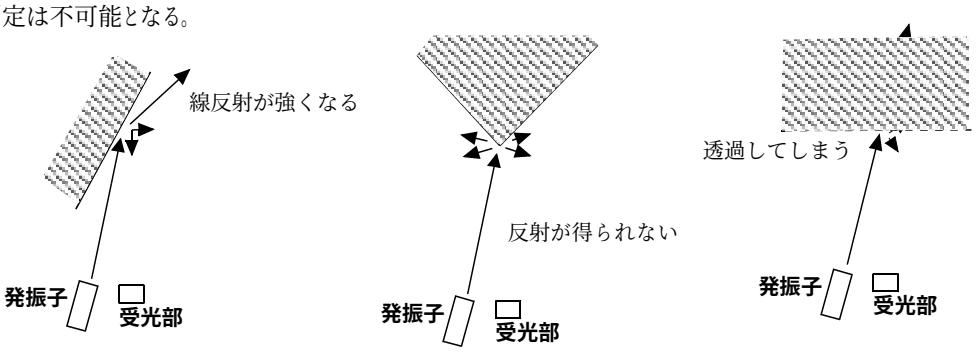


図14.2　測定のできない面

さきほどのボトルの先端で測定不良が発生しているのは、上図に示すように入射したレーザー光がボトルの面とほぼ水平に当たっているからである。そのため、このような現象の起きないボトルの太い部分では、形状を正常に読み取れていることが確認できる。

これを解決する手段として、複数の反射光の観測点を設ける方法や、乱反射成分のみを観察するのではなく直線反射光を観測する方法などが考えられる。今後はこれらの方法により、さまざまな面に対して測定を行い、どの程度有効かを調べる必要がある。

またパソコン用の処理用ソフトウェアにおいては、デバイスとの通信中にフリーズする問題がある。これは、パソコンから命令を送信し、その命令の返答があるまでをループしながら待機するからである。これには、デバイスからの返答の監視としてスレッドを利用するなどの方法により解決していきたいと考えている。

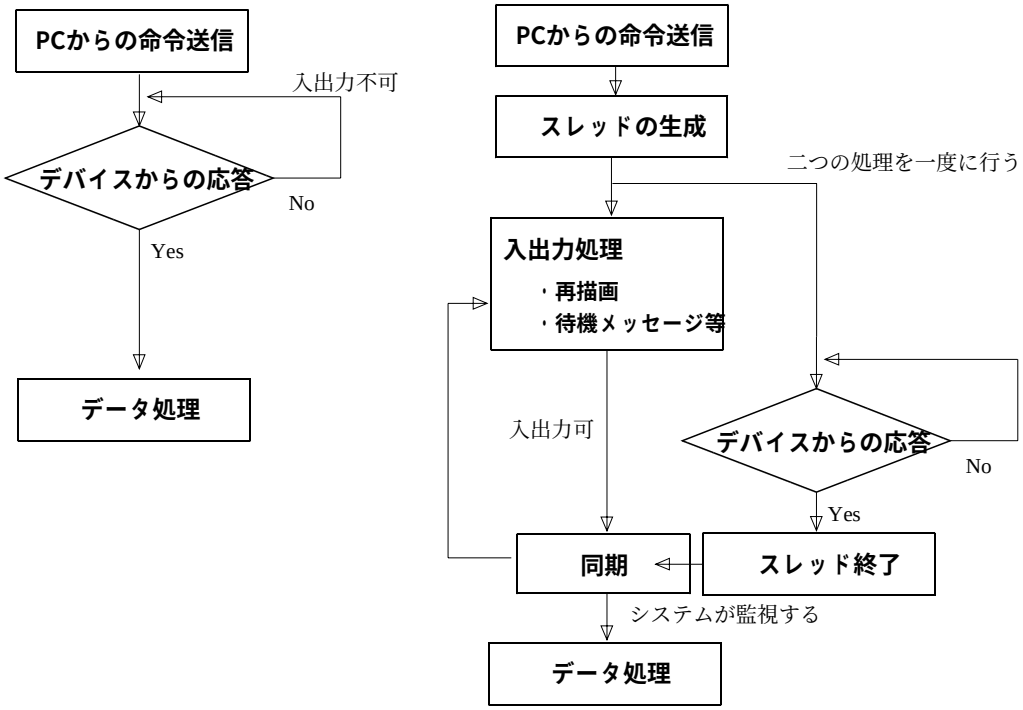


図14.3　問題解決策（スレッドの導入）

15. 参考･引用元

[1]　電子工作のための PIC18 本格活用ガイド　著：後閑哲也（技術評論社）

[2]　電子制御のための応用 PIC 活用ガイドブック 著：後閑哲也（技術評論社）

[3]　VisualBasic6.0 ステップアッププログラミング　（技術評論社）

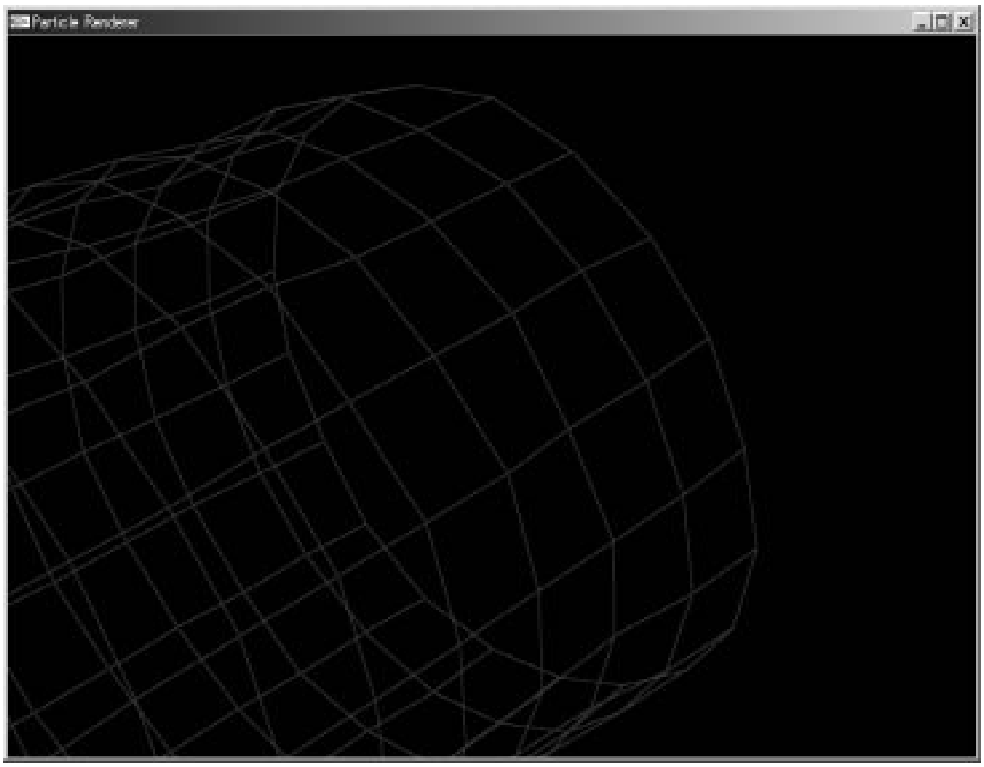
[4]　GLTU による「手抜き」OpenGL 入門　（ネットソース）

付録 A

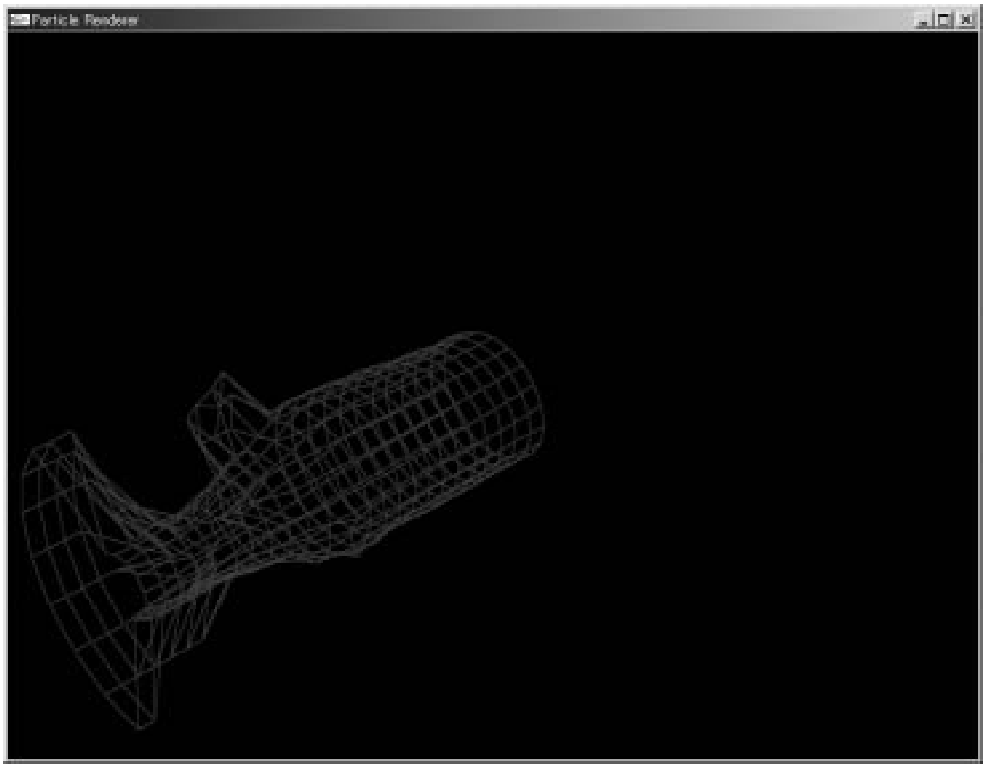
ボトルを測定した測定データ

完成した3次元スキャナにより、実際に測定対象（ボトル　下図）を測定し、
得られたデータテーブルを参考のために乗せておく。

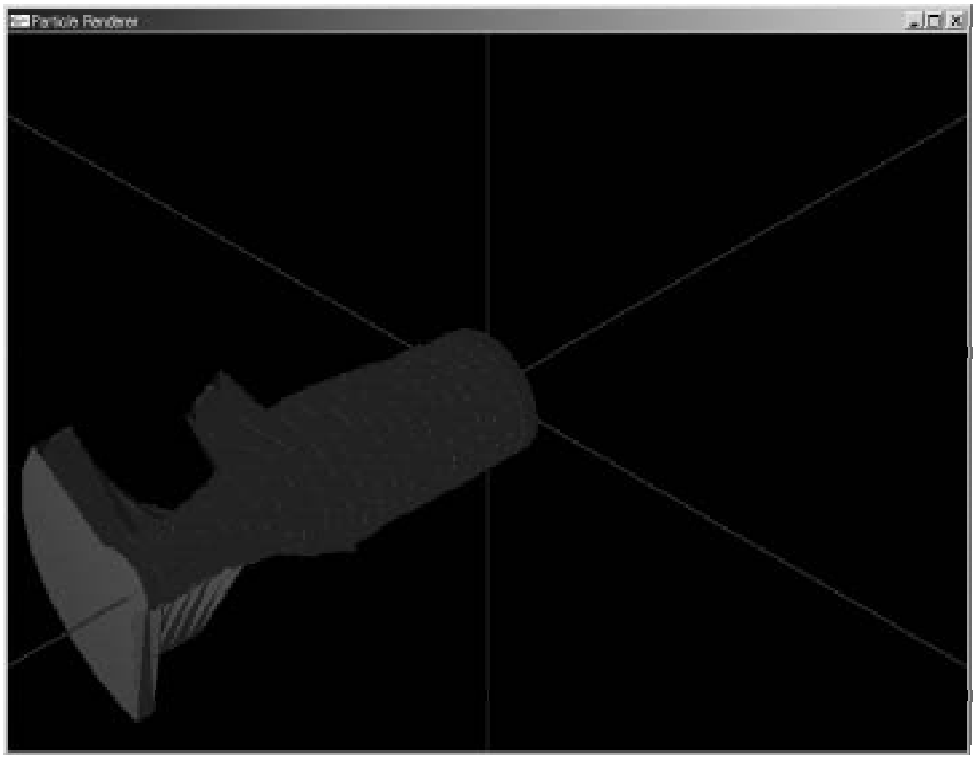
図A.1　測定対象物



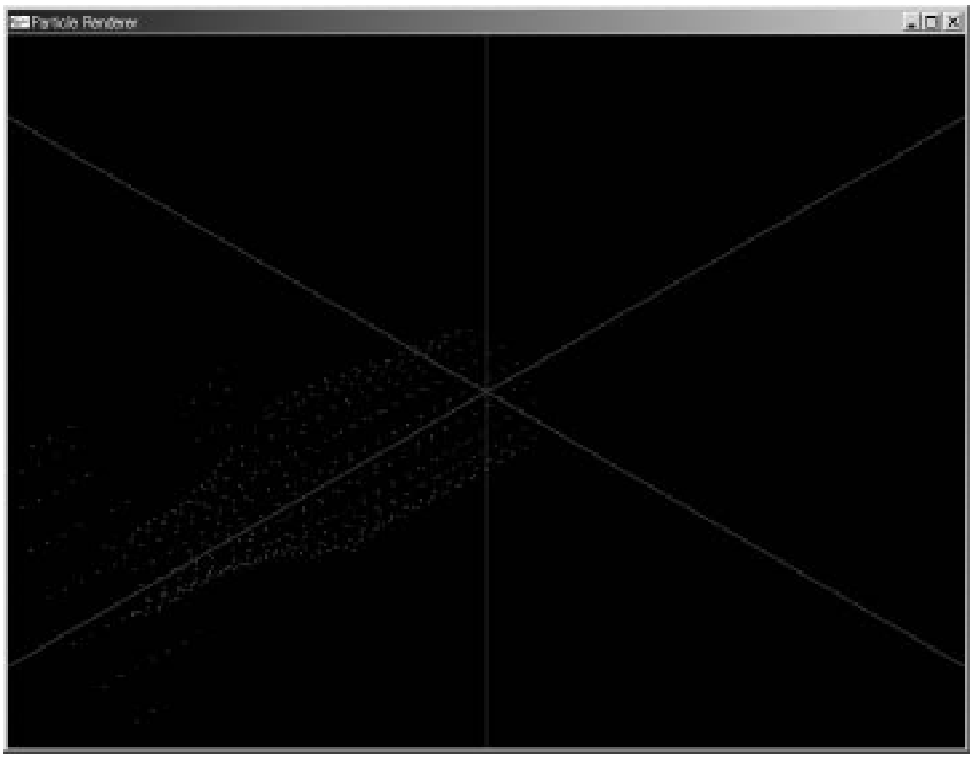
図A.2　glOutput.exeに投影した画面　（初期表示）



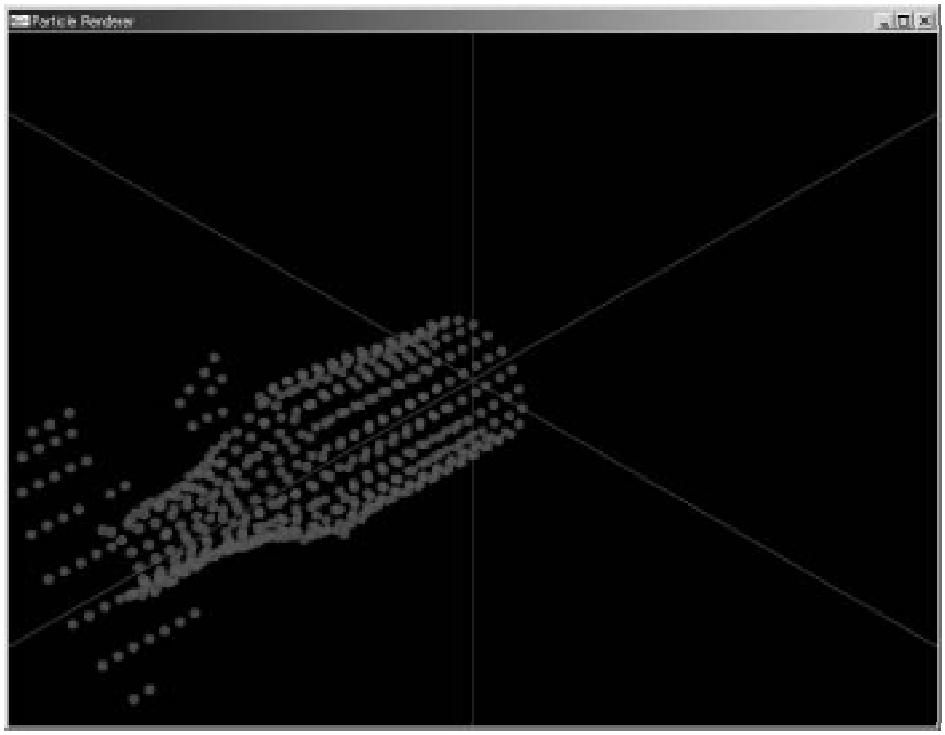
図A.3　glOutput.exeに投影した画面　（縮小表示・ワイヤーフレーム）



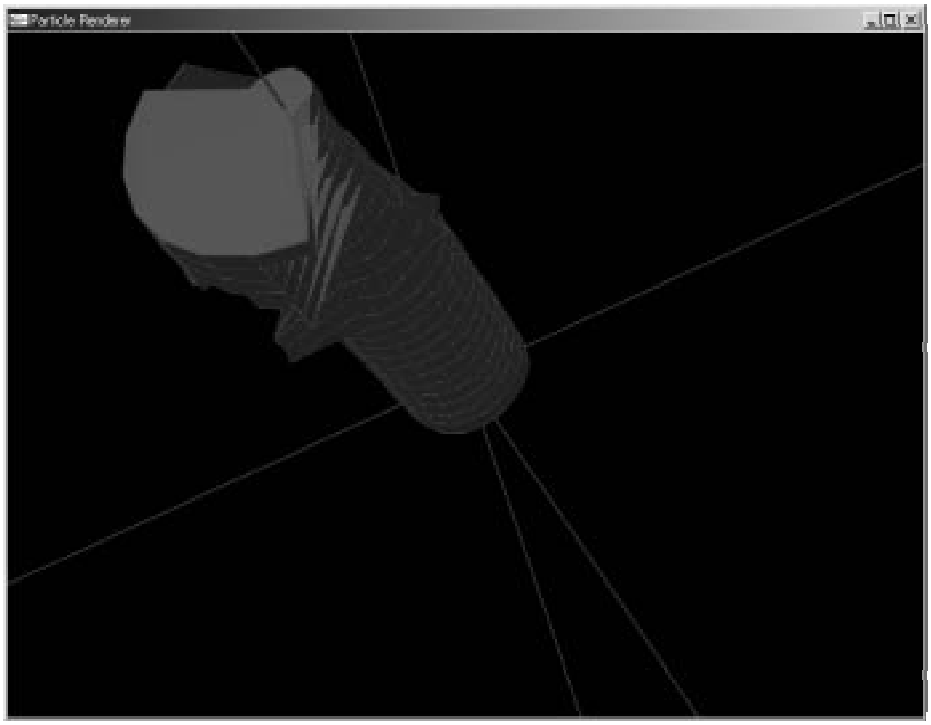
図A.4　glOutput.exeに投影した画面　（縮小・ポリゴン）



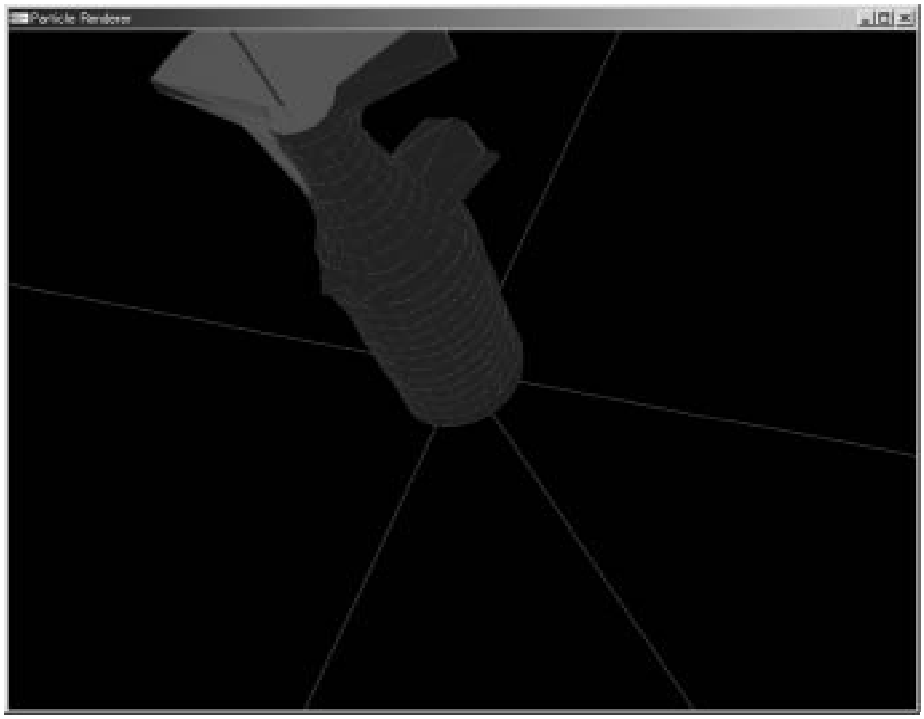
図A.5　glOutput.exeに投影した画面　（縮小・パーティクル最小）



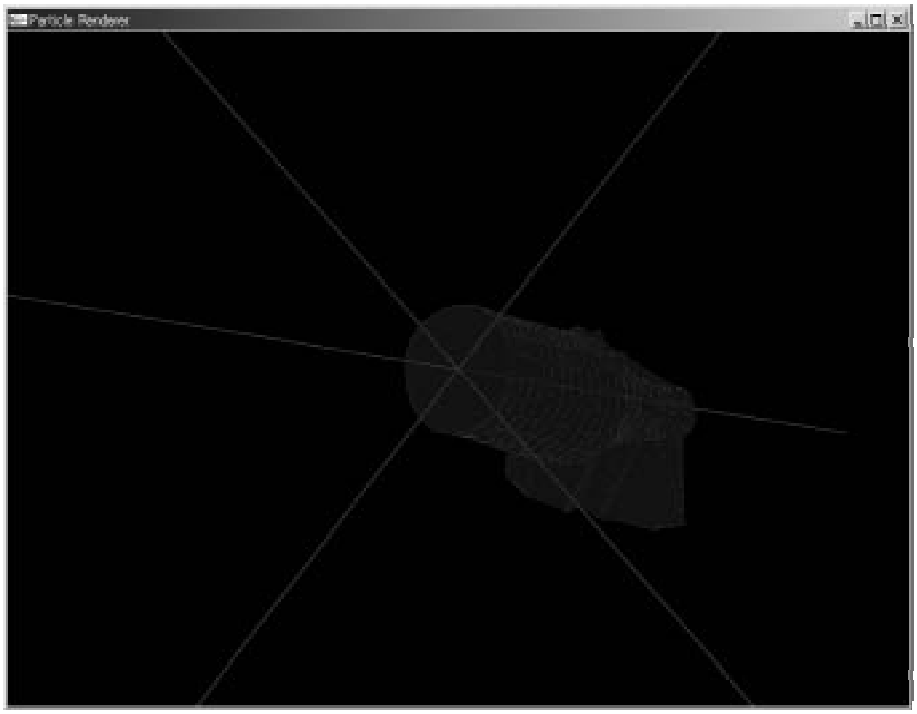
図A.6　glOutput.exeに投影した画面　（縮小・パーティクル大）



図A.7　glOutput.exeに投影した画面　（回転表示）



図A.8　glOutput.exeに投影した画面　（回転表示）



図A.9　glOutput.exeに投影した画面　（背面・回転表示）

付録 B

汎用USBドライバUUSBD用の設定ファイル　UUSBD.INF

今回USBを利用したが、これを認識させるためのドライバとして汎用USBドライバを利用した。汎用USBドライバであるUUSBDの設定ファイルの内容をここに示す。内容はUUSBDの配布元とほとんど同じ内容になっており、今回のデバイスのために次の一行が追加されているだけである。

■追加した行（36行目）

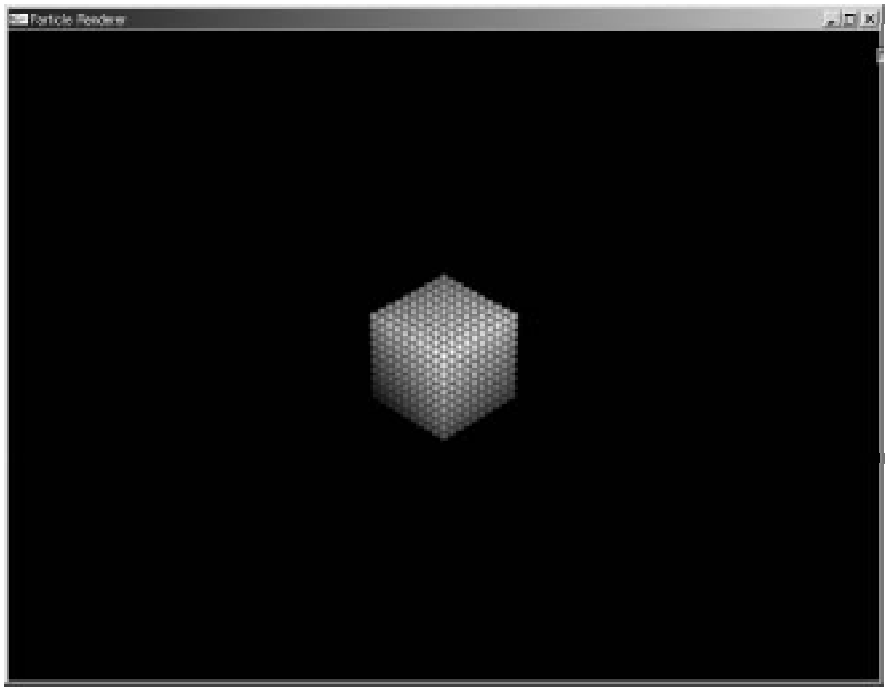
%USB\VID_09B9&PID_0042.DeviceDesc%=UUSBD.Dev, USB\VID_09B9&PID_0042

ベンダーコード09B9、プロダクトコード0042を利用するためのドライバとしている。このベンダーコード、プロダクトコードはデバイスの標準ディスクリプタテーブルの中にも同じ番号が記載されており、この番号を比較することでデバイスの認証が行われる。

付録 C

OpenGLプロッティングツール「glOutput.exe」の入出力仕様

測定結果を表示するために作成したglOutput.exeはコマンドラインから引数によるオプションをつけて起動することが可能な仕様となっている。この引数仕様をここで紹介する。引数を与えることによりglOutputのもつ、拡張性を生かすことができるので、ぜひ利用していただきたい。



図C.1　引数無しで起動した場合

glOutputに与える引数は次のような仕様となっている。識別子を利用するもの（-t、-sなどのフラグを利用するもの）ではなく、引数の場所によって引数を取り込んでいる。つまり、3番目の引数を利用したい場合は、1，2番目の引数を省略することができない仕様となっているので利用する際には十分注意していただきたい。

表C.1　glOutputの引数仕様

引数	効果
第1引数	表示するデータファイルの指定 指定されない場合glOutputと同じフォルダ内のtest.txtをデータとして表示する。 また、test.txtが存在しない場合はサンプルデータが表示される
第2引数	データの倍率が指定する。 指定がない場合はデータは等倍で表示される
第3引数	x軸方向のオフセットを指定する 指定されない場合はx=0がオフセット位置となる
第4引数	y軸方向のオフセットを指定する 指定されない場合はy=0がオフセット位置となる
第5引数	z軸方向のオフセットを指定する 指定されない場合はz=0がオフセット位置となる

プログラムソース中ではこれらの引数番号が1つずつずれているが、プログラムとして得る引数は1番目がプログラム自体のパスとなるため、実際に利用する側からみた引数は表C.1に示す通りである。

次に、glOutputの操作キーについて紹介する。キー操作を表C.2に示す。

glOutputはマウス操作ではなくキーボード操作をメインに行う。マウスに対応したい場合は、プログラム中でマウス処理部を開けてあるのでここにそれぞれの機能を追加すればよい。マウス機能を追加すれば、さらに有用なソフトウェアとなるだろう。

表C.2　glOutputのキー操作仕様

キー	効果
[ESC]	プログラムを終了する
Q	z軸中心、順方向に回転させる。
W	z軸中心、逆方向に回転させる。
A	Y軸中心、順方向に回転させる。
S	Y軸中心、逆方向に回転させる。
Z	X軸中心、順方向に回転させる。
X	X軸中心、逆方向に回転させる。
D	パーティクルサイズを小さくする (パーティクル表示中のみ有効)
F	パーティクルサイズを大きくする (パーティクル表示中のみ有効)
C	物体の透明度をあげる(透明にする)
V	物体の透明度をさげる(不透明にする)
[BackSpace]	表示を縮小する
[Enter]	表示を拡大する
[Tab]	座標軸の表示非表示を切り替える
1	ワイヤーフレームの表示非表示を切り替える
2	パーティクルの表示非表示を切り替える
3	ポリゴン面の表示非表示を切り替える

付録 D
「glOutput.exe」のソースコード

プロッティングツールglOutputのソースコードを次に示す。開発はVisualC++6.0で行った。
このソースをコンパイルする場合、OpenGLライブラリであるglutを利用するため

glut.lib　を　C:\Program Files\DevStudio\Vc\lib に
glut.h　を　C:\Program Files\DevStudio\Vc\include\GL

にコピーしておかなければならない。また実行する場合にはOpenGLライブラリである

GLUT.DLLを
Windows2000,NTの場合　　C:\WINNT\system32
その他のWindowsの場合は　　C:\Windows\system32

にコピーしておく必要がある。

＜注意＞
　以上のglut関連ファイルはglut配布元のhttp://www.opengl.org/resources/libraries/glut.html　から入手することが可能である。

付録 F

デバイス用PICのソースコード(表示･通信部)　ASM

ＵＳＢ通信･表示部を制御するPICのアセンブラをここに示す。このPICにはＰＩＣ16F877を利用しているが、これはテーブルの処理を頻繁に行うために18Fを利用すると処理が煩雑になるためである。今後、このASMを18Fに対応できるようにしたいと考えている。

このASMプロジェクトではソースファイルが増えてしまったので、ファイル構成を次に示す。

```
dev1pjt.asm          メインのASM　(これがすべてをインクルードしている)
|
+- GETMEM.H                予備メモリの確保
|
+- STDMAC.H                マクロの宣言
|
+- USBSYS.H                USBN9604の内部レジスタの番地を固有名に置き換えるヘッダ
|
+- DSCRPT42.H              USB標準デバイスディスクリプタテーブル
|
+- USBMAIN.ASM             PICが行わなければならないUSBの処理
                           デバイスの認証や、データの送受信をすべて処理している。
                           データ受信後、dev1pjt内のDO_RX1 ～ DO_RX3 にジャンプする
```

USB通信を別なプロジェクトで利用したい場合はdev1pjtのように、各インクルードファイルをインクルードし、ソース内にDO_RX1～DO_RX3をソース内に設ければよい。また、デバイスデスクリプタはテーブルのアドレスによってうまく動作しないことがあるので、注意が必要である。また18Fにこのテーブルを移植した場合、テーブルのサイズが256の制限をオーバーするために正常に動作しない。

付録F．1
メインファイル　dev1pjt.asm

メインとなるアセンブラファイル。このソースがE.2～のアセンブラファイルをインクルードする。
メモリの宣言などにUSB固有のものがあるため、今回はソースを機能ごとに分解しているが、
ファイル管理を楽にしようとするならば、インクルード位置にそれぞれのソースを貼り付け統合してしまってもよい。

付録F. 2
インクルードファイル　GETMEM.H

予備メモリの確保

付録F. 3
インクルードファイル　STDMAC.H

マクロの宣言

付録F．4
インクルードファイル　USBSYS.H

USBN9604の内部レジスタの番地を固有名に置き換えるヘッダ

付録F. 5
インクルードファイル　DSCRPT42.H

USB標準デバイスデスクリプタテーブル

付録F. 6
インクルードファイル　USBMAIN.ASM

PICが行わなければならないUSBの処理である、デバイスの認証や、データの送受信をすべて処理している。
データ受信後、メインソース内のDO_RX1 ～ DO_RX3 にジャンプする。

付録 G

デバイス用CPLDのソースコード　Verilog-HDL

ANDゲート8個分の処理をCPLDに行わせた。回路面積は小さくなるものの、コスト的にはまったく意味のない行為である。CPLDは分集器や、シリアル通信用のバッファ、ッピングモータドライバなどを作成するために利用することをおすすめする。

今回あえて利用したのは、ＣＰＬＤの利用しやすさを広めるためである。

８個のゲートの操作をこのように記述することができる。

付録 H

ＰＣ側プログラムのソースコード　VBProject

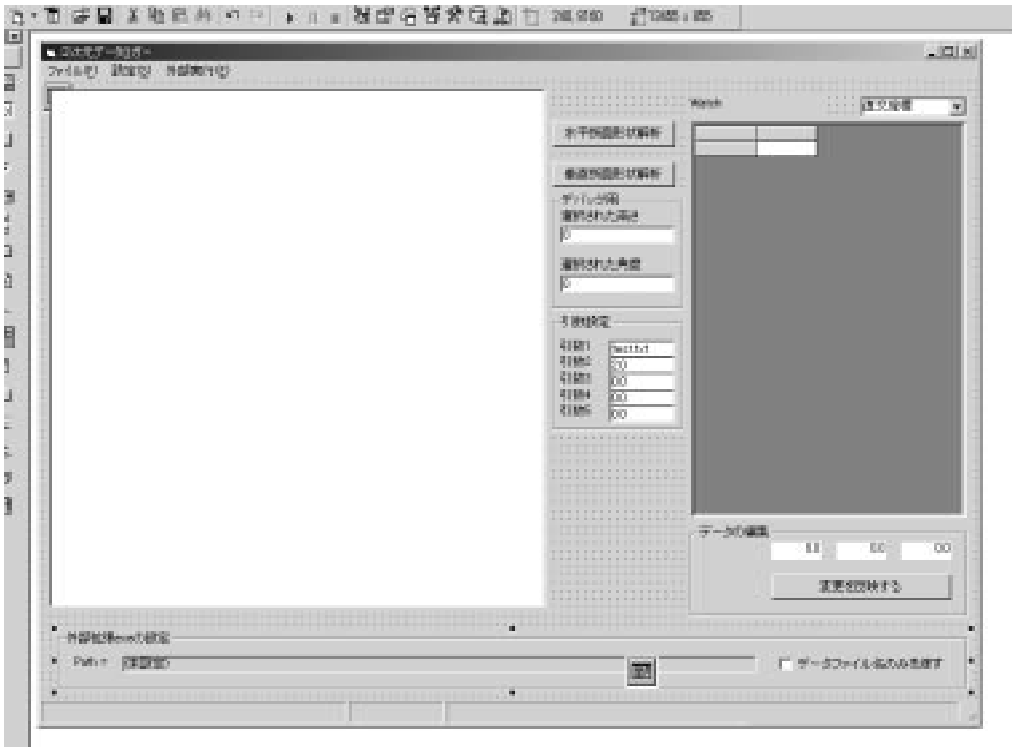
PCからデバイスに命令を送り、測定結果を表示するためのパソコン側のプログラムをここで紹介する。

開発にはVisualBasic6.0Proを利用した。各ボタンなどの挙動を指定するソースがForm1であり、USBに関するAPIの読み込みなどを行っているのがModule1である。なお、Module1を追加するには、プロジェクトウィンドウの白い部分を右クリックし、追加から標準モジュールを選べばよい。

図H.1　ファイルの構成　(VisualＢａｓｉｃの編集画面)

また、Form1はこのような構成になっている。

図H.2　フォームの構成　(VisualBasicの編集画面)



図H.3　フォームの構成　(測定Frameを取り除いたもの)

付録H．1
フォーム　Form1.frm

ユーザーが操作するプログラムのメインとなるFormである。
このプログラムはglOutputなどを呼び出したり、データファイルを保存したりすることができる。

付録H．2
標準モジュール　Module1.bas

USB通信を利用する際のAPIを宣言している。これらの宣言により、ReadFileなどの関数が利用できる。
また、USB通信用のベンダーコード、各種マスクも同時に宣言している。

付録 I

最終発表用資料

当研究を卒業研究最終発表で発表した際の配布資料とプレゼンテーションのスライドを参考までに載せておく。今後、抄録を書く際に何も参考資料がないよりは、何かしら参考になるものがあると良いと思い、載せることにした。発表内容は抄録に示す通りであり、プレゼンテーションにはアニメーション効果は使っていない。

視聴覚室のプロジェクタ（ 高さ3m、 幅5m程度 ）に投影したところ、十分に後ろからみることができた。

後書き

３月２４日卒業式も終わり※1、新居にも引っ越し※2、卒業論文が仕上がりました。さて、実際このように論文として一年間の成果をまとめてみると、本文は３０ページ強と１年間にしては１回の実験レポート以下の枚数になり、あまりやっていないことに気がついてしまいました。

USB 通信部を作成したのが 2003 年 1 月※3。２、３月は昨年の卒研生木村氏から PIC、基盤の焼き方等を学び金野研究室での前知識取得。４月に秋月電子通商で買ってきた EEPROM を PIC から操作できるプログラムを作成したりしました。５月からは研究室にあるセンサーやら駆動部品などを片っ端から PIC で動かしたり、同研究室のインさんと湿度計の作成やら基盤の焼き方などを夏休みまで練習していました。練習ついでにいろいろ卒業研究テーマを模索し、現在の「LASER」にたどり着いたのが６月頭。そうこうしているうちに大学受験もあり、６月に群馬大学情報工学部（ぁーいちおう受かりました）、７月に筑波大学第三学群情報学類（今の予定だと計算機システムコースです）を受けまして、７月後半になってしまいます。実際のところ、第一志望は筑波大学を受けるまでは北海道大学電気系だったわけですが、筑波受けた感じ「筑波でいいんじゃないだろうか」ということになりまして、筑波大学に落ち着いたわけです。しかしまぁ、筑波大学情報学類になるのならば、こんなに受験勉強はいらなかったのではないでしょうかという時間ロスが結構目立ってしまい、ほとんどの科目を勉強できた※4ことの嬉しさもあり、時間ロスに対するむなしさもありまして、北海道大学を受験してもよかったのではないかなぁと今となって残念に思っていたりします。そうこうしているうちに、中間発表になりまして、８～１０月でレーザーを使って距離が測れることのデモ機を作成したりしました。距離が測れれば、あとはデータテーブルを作成するだけ、というわけですがここで運転免許をとりはじめたりとか、文化祭で爆弾※5を作ったりしているうちに１月になってしまいます。そして最終発表へ。

このように一年間を文章に直してみると、結構なにもやっていないことがばれてしまいますが、論文の厚さと完成品の測定具合をみてみますと、一年間でいろいろできるものですねぇとか思っているところでもあります。おかげさまで、卒業研究はこの学年で１位といううれしい結果までいただきまして、ありがたい限りです。

さて、いい加減技術的な話にしないと怒られてしまいますので。

等研究では PIC、CPLD、PC などなど様々なハードを利用し、隣の研究室（鹿野研究室）の FPGA も見せて頂いたりとか、今年１年はいろいろなハードウェアを体験できました。

いろいろ使ってみてわかった結果ですが、はやりデバイス側は PIC が使いやすく安価なので制御には良いと思います。H8 等の制御器はやはり少々高価ですので、練習には使いにくいと思います。

あと、PIC での処理速度不足に悩んだ場合は CPLD、FPGA を使うのがよいでしょう。CPLD、FPGA 等は最近は金に物を言わせれば数 GH ｚ動作の物の手に入りますし、金がないよーとか、卒研ですこし速度がほしい場合はどは数千円程度で数百 MHz 動作の物が手に入ったりします。開発言語（HDL）にはいろいろな物がありますが、５年の授業で N 先生が行う Verilog-HDL を今回は利用しました。ただし、実際のところ Verilog-HDL は可読性はとてもよいのですが、自分はあまり好きではありません。もしこれから HDL を学びたいという人がいましたら、VHDL をお勧めします。あと、HDL は実際ハードウェアが動いて楽しい物ですので、コンパイラだけで遊ぶよりは千石電商あたりでダウンロードケーブルと CPLD 素子を買って LED でも光らせるほうが良いと思います。

また、今回の研究ではふれませんでしたが使ってみたい物としまして DSP があります。現在１万円程度で高速処理の DSP ボードが入手できるそうなので、アナログ信号のデジタル処理を行いたい場合は DSP をお勧めしたいです。

PC 側も様々な物にふれました。元々、自分は PC 側のプログラムしかやっていなかったので※6 PC 側のプログラムはいろいろと１年間楽しむことができました。この研究室に来まして中学校のときにやっていた VisualBasic がとても役に立ったのは事実です。ただ、実際のところ VisualBasic 単体では使いにくいもので、VisualC++で DLL 作成ができると素敵だと思います。VisualC++でシステム全体を構成するのもよいですが、プログラム効率が下がってしまいますので、研究用にはあまり向かないかなぁというのが持論です。あと、次期５年生から Java の授業が入るようですが、Java も実に重要な物ですので、覚えておくとよいと思います。

今後の金野研究室、小山工業高等専門学校共、ますます良い学校になってほしいと願っています。幸か不幸か、電子制御工学科は新しい学科ということもあり、適度に新しい機材があり、適度に新しいパソコンがあり、最新の技術に関する欲しい設備がないので買い足すことができます。あと、実験室も自由に使えますし、研究室もお邪魔してしまえはこっちのものです。時間に余裕があるのでしたら、研究室掛け持ち、なんていうのもおもしろいかもしれません（実際うちの学年にはそれらしい人が数人いたようです。金野研究室の測定器使いに来た人も多いですし、自分もこっそりと他の研究手伝ったりしていますし）。

来年からの卒業研究生活がますます良い物になることを願っています。

自分の聞く限り、レベル高い人そろっているようですので。来年以降の３月を毎年楽しみにさせていただきます

ここまで、おつきあいありがとうございました。それではまた、どこかで。

２００４年　3月　２４日

※１　3月１８日が卒業式でした。この論文提出日は３月２４日です。

※２　某筑波大学正面門より 1.5km ほど、医学から徒歩１分、7.5 畳 BT 別 K3 畳の月 3.6 万のアパート。いろいろと不具合が。

※３　２００３年１月、当時高専４年。このころは金野研究室志望が１名しかいなかったために一人でよくお邪魔していました。

※４　なぜか関東、東北、中部にある全大学の過去問は解けるようにがんばりました。

※５　偽物ですが。部外者は近づいてくれませんでした。内部の人々は喜んでくれたみたいで。

※６　２００３年１２月（高専４年の１２月）まではデバイス（ハードウェア）にさわったことがありません。抵抗、トランジスタなどなどは知識としてしか知らず、正直回路とは無縁の人でした。

レーザーを使った３次元スキャナの作成

初版　：　2005 年 3 月 24 日　印刷

著者　：　青柳　信吾

連絡先　：　citrus@104.net

読まれた方含め、作ってみたいのですが等制作に関するご質問等ありましたら連絡お願いします。

データファイルの破損、実行エラーなどの連絡もよろしくお願いします。