

Arduinoを使った3Dデータの作成 Grasshopper+Fireflyを使った モデリング手法

Creating 3D data using Arduino Modeling technique with Grasshopper + Firefly

真壁 友
MAKABE Tomo

キーワード：Arduino, Profileprojector, Grasshopper, Firefly,
3D CAD

This report introduces the connection method of Profileprojector (Nikon V-12) and CAD Software. Profileprojector is a machine that optically measure. This can read the coordinates of the X-Y. It is possible to reproduce the model to use the CAD from the coordinate data. However, the X-Y coordinate data must be input manually. This Software brings the coordinate data to automatically CAD from Profileprojector to use a plug-in software(Grasshopper).

○ Arduino

3DプリンタやCNCフライス盤の発達により3次元加工が従来よりも身近になった。3D CADを使い立体図形をモデリングし3Dプリンタで出力する、これは従来は無かった物作りの手法である。しかしこの手法が広く普及しているかは疑問である。低価格3Dプリンタが登場した当初は「夢の3Dプリンタ」とも呼ばれたがそれほど浸透していない。出力時間や出力精度の問題もあるが、それ以外の理由として3D CADのソフトウェアの使いこなしが問題となる。3D CADは操作の習得までに時間がかかる。奥行きのある立体物を平面のコンピュータ画面で作るため敷居が高い。しかし3Dプリンタで自分がイメージする形状を出力するためには3D CADを使い、自分でモデルを作り出す必要がある。インターネットで公開されている3Dデータ等もあるが、他人が用意した部品を出力するだけになり3Dプリンタを活かし切れた使い方とは言えない。

3D CADではコンピュータディスプレイ上でマウス等の入力装置を使い行う。平面の装置を使い立体を作るために困難が生じる。一部には3Dペンといった物を使い立体造形を行う補助道具もあるが、まだ一般的にはなっていない。本研究では様々なデバイスを3D CADに接続しユーザの補助を行うことの基礎的な試行である。様々なデバイスを接続するために汎用性が高くユーザも多いArduinoを使い外部機器を3D CADソフトに接続する。Arduinoは安価で種類も豊富なマイコンボードであり多

くのユーザにより利用されている。このArduinoを3D CADソフトウェアのRhinocerosに接続するためにはGrasshopperを使う。Grasshopperはアルゴリズムで形状を作り出すRhinocerosのplug-inである。GUI画面で機能モジュールを線で繋ぐ要領でプログラミングが可能な環境で手軽に取り組める。Rhinocerosだけでは困難な形状を数学的な手法により作る手助けをしてくれるplug-inである。近年、建築を始め様々な分野で活用されている。このGrasshopperの拡張機能としてFireflyというアドオンの機能がある。これをGrasshopper上から呼び出すことによりArduinoと通信を行いデータの取得が出来る。

今回は精密なデータを扱う目的で万能投影機と3D CADの連動を試みた。万能投影機はステージに置いた部品を光学的に拡大しその形状をスクリーンに投影する装置であり、部品の形状確認に使われる装置である。レンズは10倍から100倍程度が一般的であり、テレセントリック工学系により歪む事無くスクリーンに部品形状を投影できる。100倍のレンズを使った場合にはスクリーン上の1mmが実物の0.01mmに相当する。スクリーン上に定規やガイドを当てて測定する他に、スクリーン中央のクロスラインを基準にした測定も可能である。X-Y微動ステージを使い1 μ m単位の寸法を読み取る事が出来る。最近ではCCD等のイメージセンサを使用した画像測定器が主流となっていて万能投影機は前世代の測定機器とも言える。しかし画像測定器は自動で各部の寸法を読み取るといった機能を持っているがために高価であり導入が困難な場合もある。万能投影機は安価であり中古で20万円程度で手に入れることも可能である。しかし万能投影機は座標値を出すだけになり、そこから寸法を求め図面にするためにはかなりの労力が必要と成る。この部分を半自動化するためにも万能投影機と3D CADソフトのRhinocerosを連携させる意味がある。

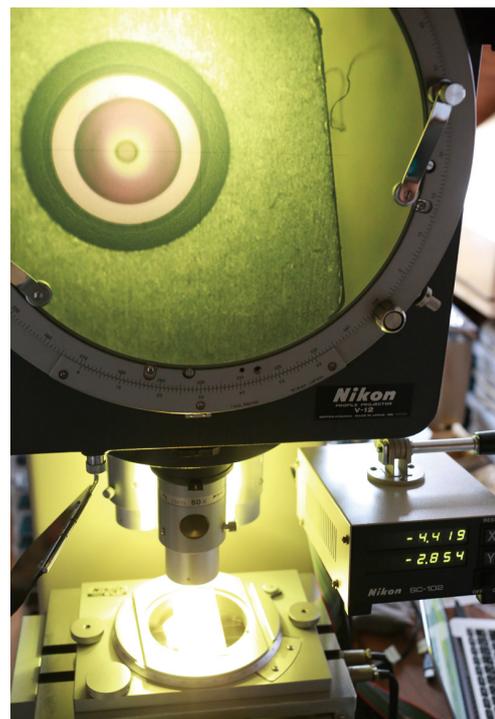


Fig - 1 万能投影機とX-Yカウンタ

万能投影機は Nikon V-12 を使用。X-Y カウンタは Nikon SC-102。カウンタが微動テーブルの X/Y 軸のリニアエンコーダの移動量を読み取り表示をしている。このカウンタは出力ポートから X と Y を出力しているが出力形式は X/Y の値では無くカウンタの 7 セグ LED の表示内容に対応したデータになっている。そこでこのポートから出力される信号を Arduino で読み取り、数値に変換しパソコンへシリアル通信で送信する。カウンタと Arduino を接続するシールド (Arduino の拡張基板) の回路図は【Fig-2】の通りである。将来の拡張を考えスイッチと LED を入れたが今回は値確定のため 1 つのボタンしか使っていない。

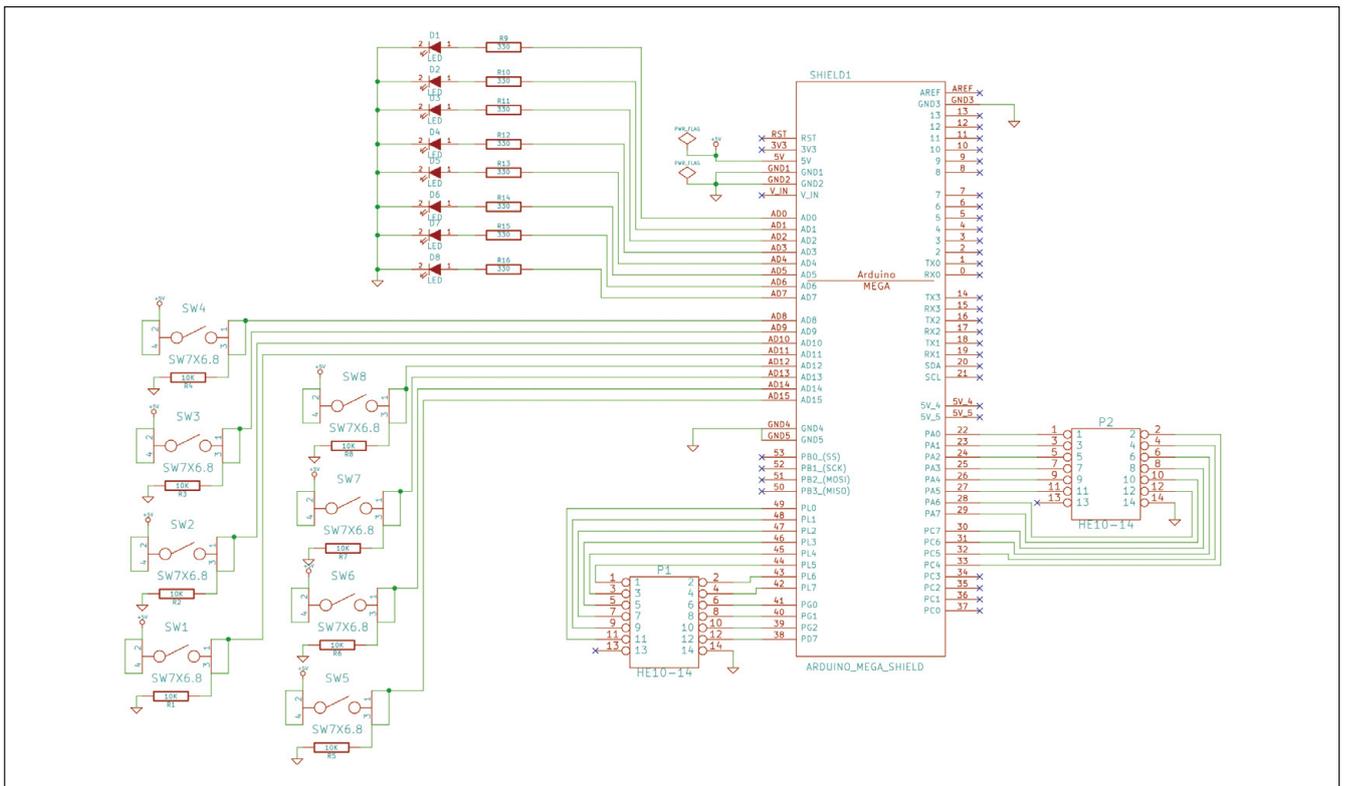


Fig-2 回路図

カウンタと Arduino Mega を接続するシールド基板の様子は Fig-3 である。リボンケーブルの先にカウンタが接続され、USB ケーブルはパソコンへと接続されている。Arduino のファームウェアのソースコードは紙面では長くなるので以下 URL に掲載する。
http://www.mkbtm.jp/nid/sc102_arduino/sc-102_data/Processing_unit-2.ino

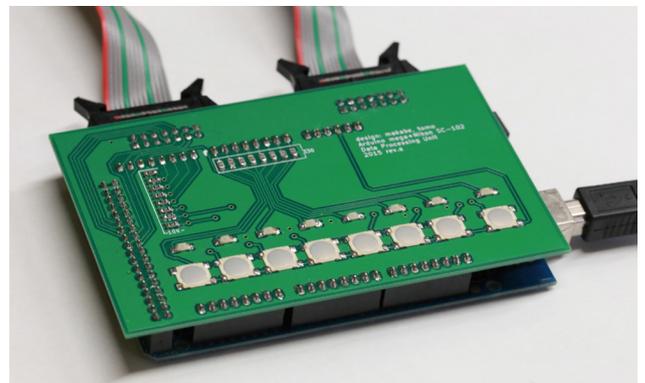


Fig-3 カウンタ接続用 Arduino Mega 用シールド

Grasshopper のプログラムは Fig-4 のようになる。Arduino から送信されたデータを Firefly で受信して処理を行う。データは 1 行でデータの区切りを ; (セミコロン)、X と Y の区切りを , (カンマ) にしている。このデータをリスト形式に変換して Rhinocross のポイント形式に変換して画面に表示を行う。リストから座標への変換と描画部分は Grasshopper 内の C# で行っている。C# のコードは Fig-5 の通りである。

C# のコードで分かるように Z 軸の値は 0 にしている。万能投影機は X と Y 軸方向は測定出来るが、Z 軸高さ方向は測定する事が出来ない。高さ方向はマイクロメータ等で測定し CAD データを作成する必要がある。

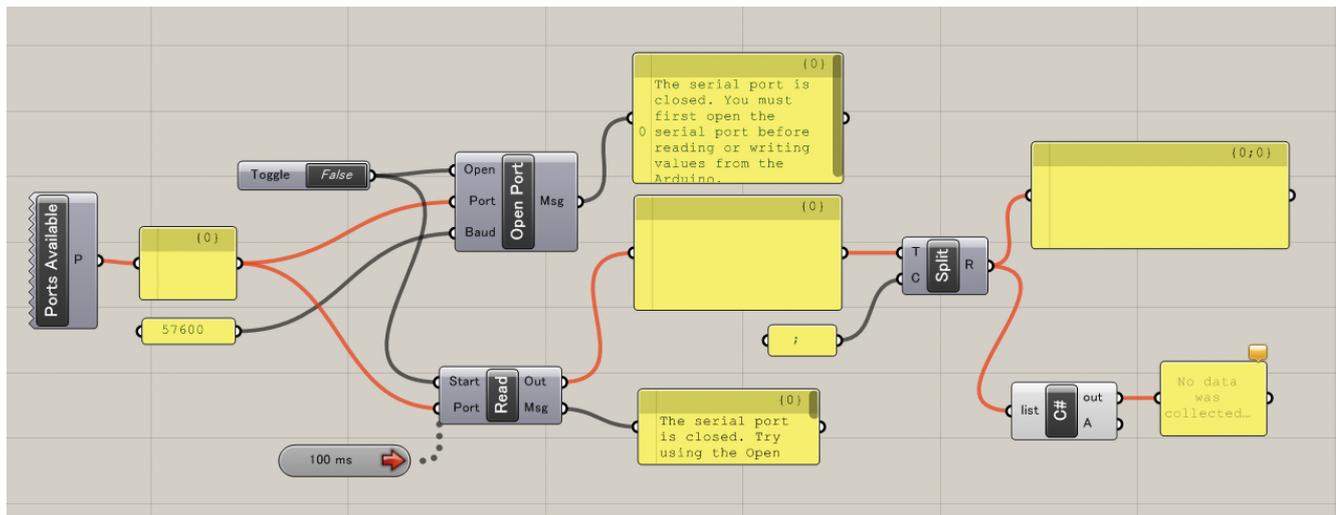


Fig - 4 Arduino データ受信用 Grasshopper プログラム

```
private void RunScript(List<string> list, ref object A)
{
    List<Point3d> pts = new List<Point3d>();

    // 文字列を double に変換するため。-0 ~ 9 だけを残して余計な文字を消す正規表現
    System.Text.RegularExpressions.Regex re = new System.Text.RegularExpressions.Regex(@"^[^-0-9^\.]*");

    foreach (string point in list) // 受け取った list を一行ずつ処理
    {
        if (point != ""){
            double[] pointValue = {0.0,0.0,0.0};

            string[] stArrayData = point.Split(','); // リストの行をカンマで分解して配列に入れる。
            int i = 0;

            foreach (string s in stArrayData) // 配列に入った値を順番に処理
            {
                if (s != ""){
                    string dataString = s;
                    dataString = dataString.Trim(); // 前後のスペースを削除 (いらない?)
                    dataString = re.Replace(dataString, ""); // 正規表現をつかって置換

                    double ss = double.Parse(dataString);
                    //double ss = Convert.ToDouble(test);
                    Print(ss.ToString());

                    pointValue[i] = ss;
                    i++;
                }
            }
            Point3d tempPt = new Point3d(pointValue[0], pointValue[1], 0);
            pts.Add(tempPt);
        } //if (point != "")
        A = pts;
    } // end of foreach
}
}
```

Fig - 5 Grasshopper 内 C# スクリプト

応用例：

このシステムの活用例として時計部品の製作を紹介する。キャリパー ETA6497 のガンギ車、4 番車、3 番車のブリッジ Fig-7 の形状を読み取り同じ部品を CNC フライス盤で切削加工する。サイズは 18mm×16mm 程度である。



Fig-7 ETA6497 のブリッジ

作業手順は次の通りである。

- 1 万能投影機のスクリンを見ながら微動テーブルを動かしてポイントとなる点を記録させていく (Fig-8)。この作業により Rhinoceros の画面上に点がプロットされる。円弧であれば最低 3 点、直線部分であれば 2 点を読み取る必要がある。Fig-9
- 2 点を元にして輪郭形状を作成する。Fig-10
- 3 部品の厚みを測定し立体形状にする。Fig-11
- 4 切削パスの作成
- 5 CNC フライス盤による切削加工

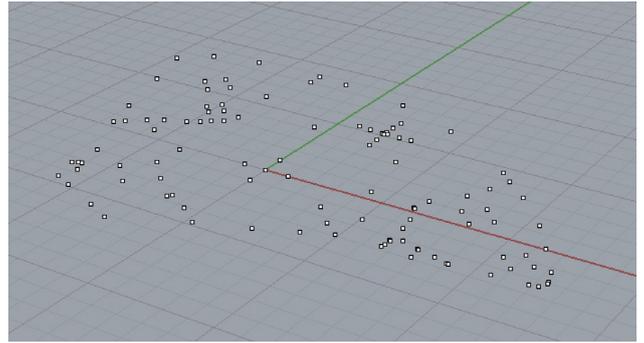


Fig-9 Rhinoceros 上の座標群

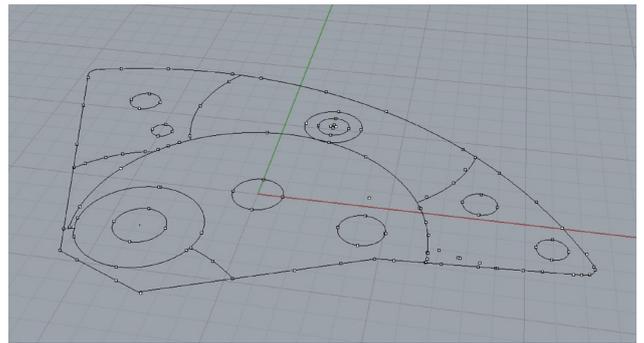


Fig-10 座標を元に作成した輪郭線

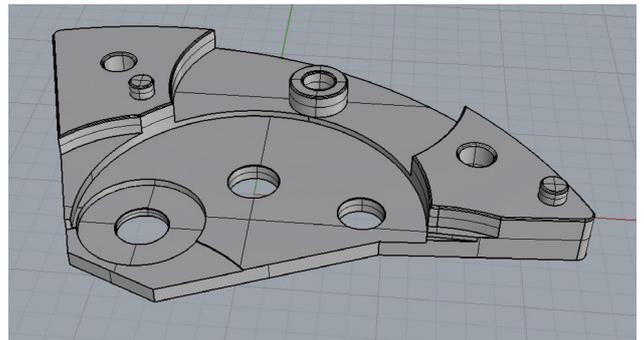


Fig-11 3D モデル

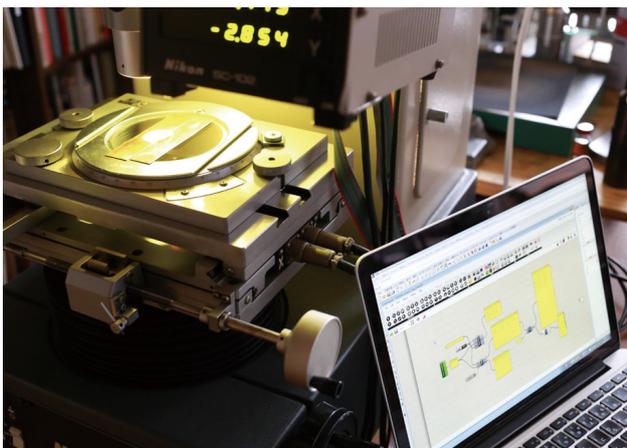


Fig-8 読み取り作業の様子

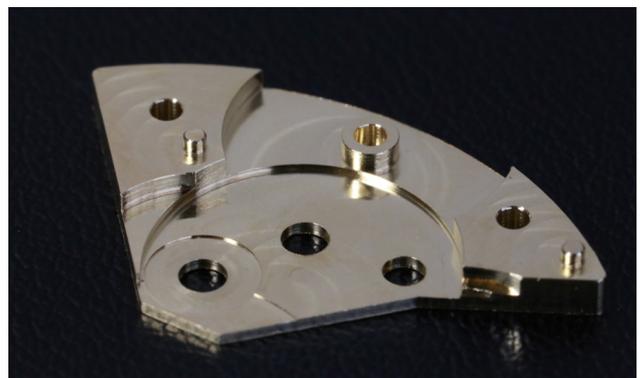


Fig-12 切削加工で作成した部品 (素材：真鍮)

完成した部品を Fig-12 に載せる。画像では軸受け部品 (ルビーの受け石) が入っていない、この後受け石を圧入で入れる事になる。

今後の展開：

今回の試行では座標を点データとして Rhinoceros で読み取り、その座標データを元にして輪郭線を作成した。この作業は経験を要する部分であり手軽な作業ではない。しかし、座標が全て画面上にプロットされているので作業時間は大幅に短縮でき、ミスを減らす事が出来る。さらに簡単に作業が可能ないように読み取り時に Arduino 側の操作で円、円弧、直線、点と読み取り対象を指定し Rhinoceros で図形オブジェクトの生成まで行うと行った改良が考えられる。そのためには Arduino の回路にハードウェアスイッチの追加が必要となる。今回の取り組みで単純な装置とソフトウェアの追加により一連の作業時間を短縮でき、Grasshopper + Firefly と Arduino の連携によるモデリングの可能性を確認する事が出来た。Arduino の特徴はユーザ数が多く、様々な拡張シールドが発売されている様に各種の入力センサーや出力機器との接続が可能な事である。今後は様々なセンサーやデバイスを使いモデリングの新たな方法を探る。

参考文献・資料

Glasshopper <http://www.grasshopper3d.com> 2016.2 閲覧

Firefly <http://www.fireflyexperiments.com/> 2016.2 閲覧

Modeling Veil: The surface-capturing textile

Yuzuru Masuda, Tomo Makabe, Akito Nakano, Akira Wakita

Re-Thinking textiles and surfaces 2014