

音を利用したインスタレーション

-打楽器の新たなインタフェース(POMPOM)の制作-

担当教員：長嶋 洋一 教授
デザイン研究科 1332010 リュジュンヒー

目次

- 1 . コンセプト
- 2 . 外見デザイン
 - ① CADデータ
 - ② 実物
- 3 . 内部の仕組み
 - ① 電子回路とCAD
 - ② プログラミング



コンセプト

- 打楽器が持つ演奏方法は叩くだけである。ここに新しいインタフェースを通じて、幅広い演奏方法を持つことでよりアクティブに音を楽しむ、音を利用したインスタレーションをデザインし、制作することを目標とした。



外見デザイン

○ 基本形の選定



- アクティブ的な演奏のため主な演奏方法は「叩く」ことであること。
- 「叩く」、「回す」、「振る」動作をしやすい大きさの楽器であること。
- 「叩く」、「回す」、「振る」動作のため、演奏の時に両手が自由であること。



外見デザイン

○ ボタンの配置



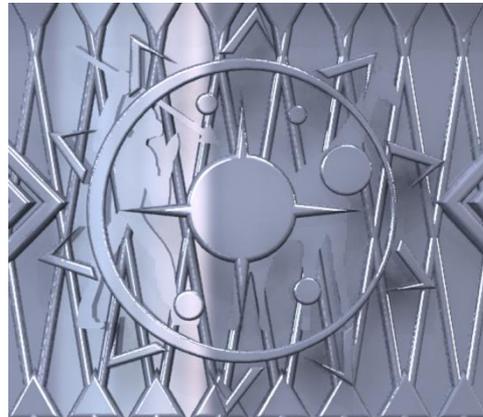
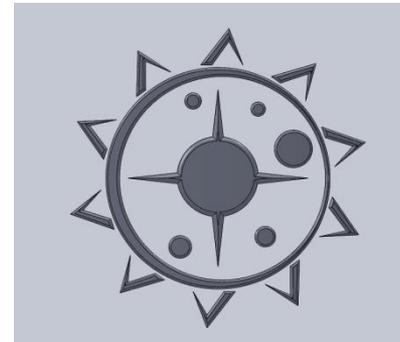
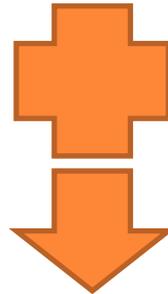
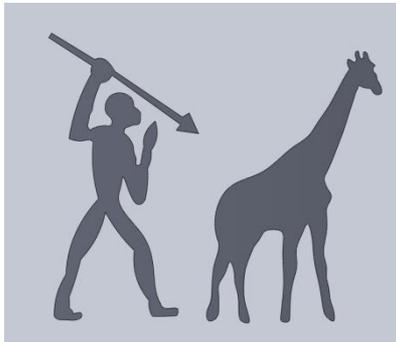
- 計9個のボタンを叩く。
- 回転板を回す。
- 本体を持って振る。



外見デザイン

○ 外見の様子

アフリカの民族的デザインを参考にした柄を重ねて、
見る人からの距離によって異なる印象を与える。



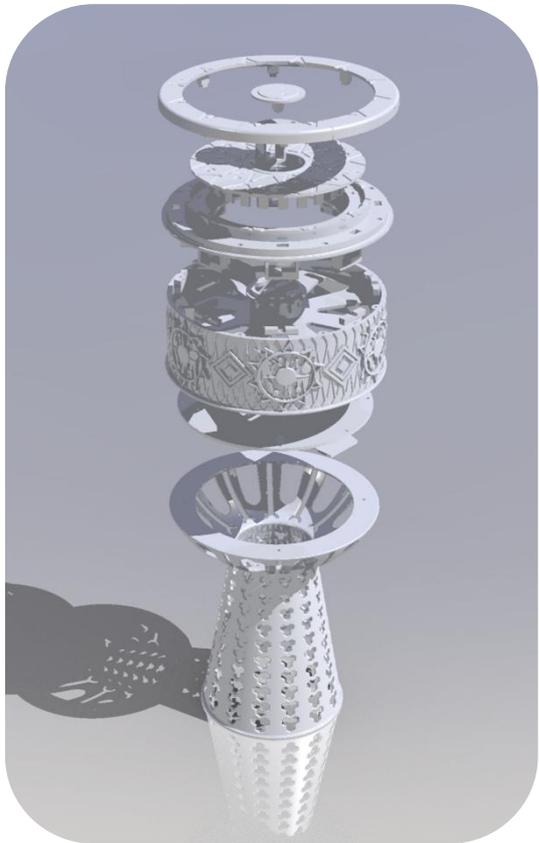
外見デザイン

- CADデータ



外見デザイン

- CADデータ



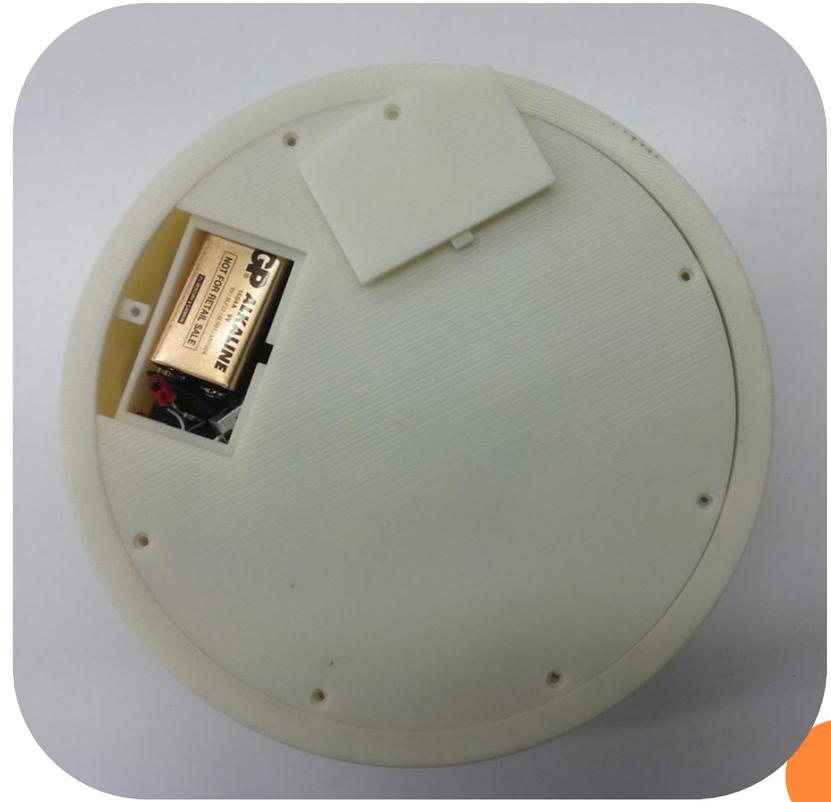
外見デザイン

○ 実物



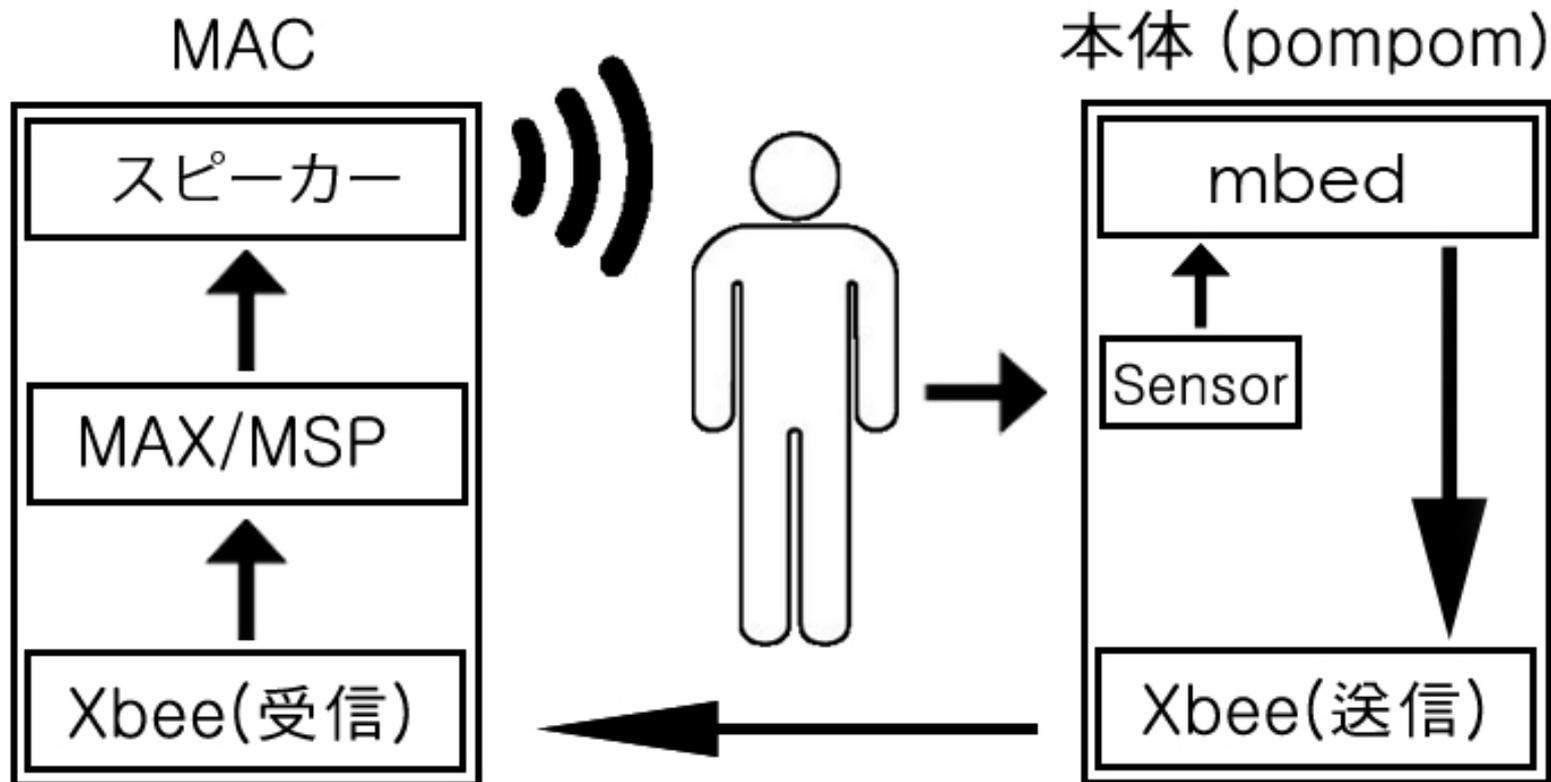
外見デザイン

○ 実物



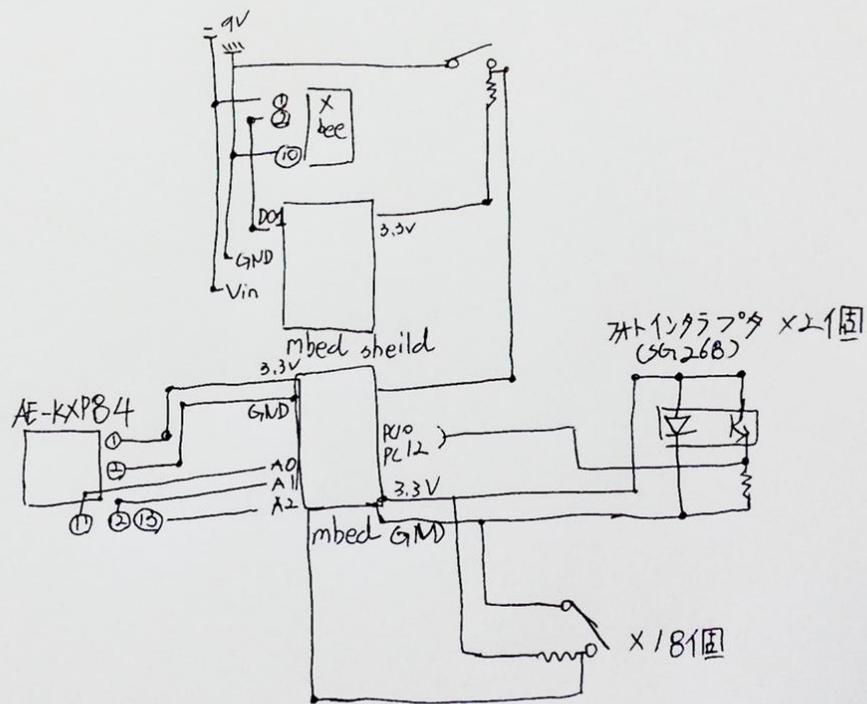
内部の仕組み

○ 電子回路



内部の仕組み

○ 回路図



① 「叩く」スイッチが9個×2
(2個の時間差で叩く強さを
計算する)

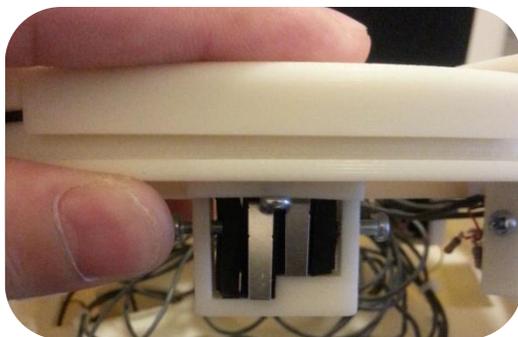
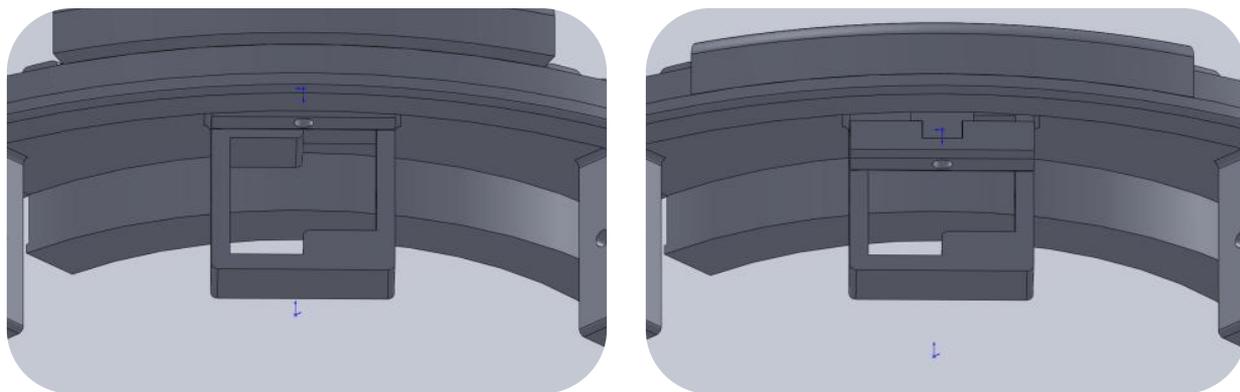
② 「回す」フォトインタラ
プタ(SG268)×2

③ 「振る」3次元加速度セン
サ(AE - KXP84)(x,y,z)×1



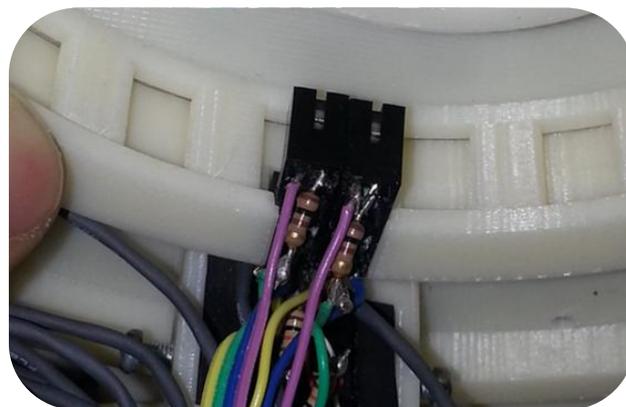
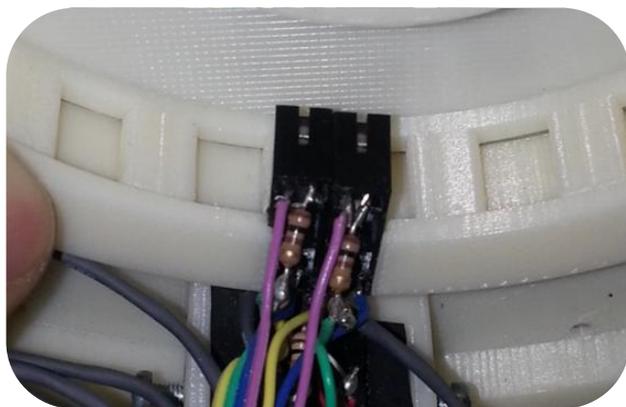
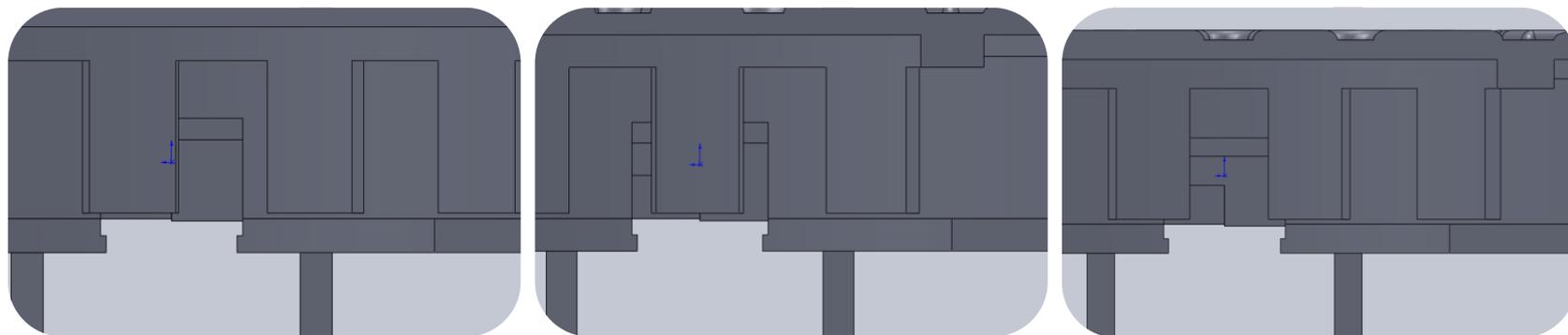
内部の仕組み

○ スイッチ



内部の仕組み

○ フォトインタラプタ



内部の仕組み

○ MBED

- MBED publishing URL

<http://developer.mbed.org/users/chickesnoup/code/pompom01/>

```
int i = 0;
if(pi_array[0] != pi_array[1]) {

    i = 20000 + pi_array[0];
    tx_message(i);
    pi_array[1] = pi_array[0];
    return 0;
}
return 0;

void send_btn(int checkbtn)
{
    int i = 10002 + (checkbtn * 1000);
    int ii = 0;
    if(check_time[0][checkbtn] == i && check_time[1][checkbtn] =
        timer[checkbtn].stop();
        ii = timer[checkbtn].read_ms();
        if(ii > 997) {
            ii = 997;
        }
        ii = i+ii;

        tx_message(ii);
        timer[checkbtn].reset();
        check_time[1][checkbtn] = 1;
    }
}
```

```
int checkBT(int checkbtn)
{
    int i = 0;

    if(SW[checkbtn][0] == 1 && SW[checkbtn][1] == 0 ) {
        timer[checkbtn].start();
        return i;
    } else if(SW[checkbtn][0] == 1 && SW[checkbtn][1] == 1 ) {
        i = 10002 + (checkbtn * 1000);
        return i;
    } else if(SW[checkbtn][0] == 0 && SW[checkbtn][1] == 0 ) {
        i = 10000 + (checkbtn * 1000);

        check_time[1][checkbtn] = 0;
        return i;
    } else if (SW[checkbtn][0] == 1 && SW[checkbtn][1] == 0 ) {
        return 0;
    }

    else {
        // i = 10003 + (checkbtn * 1);
        return 0;
    }
}

//send pi state
int send_pi(int (*pi_array))
{
    int i = 0;
```

内部の仕組み

○ MAX/MSP

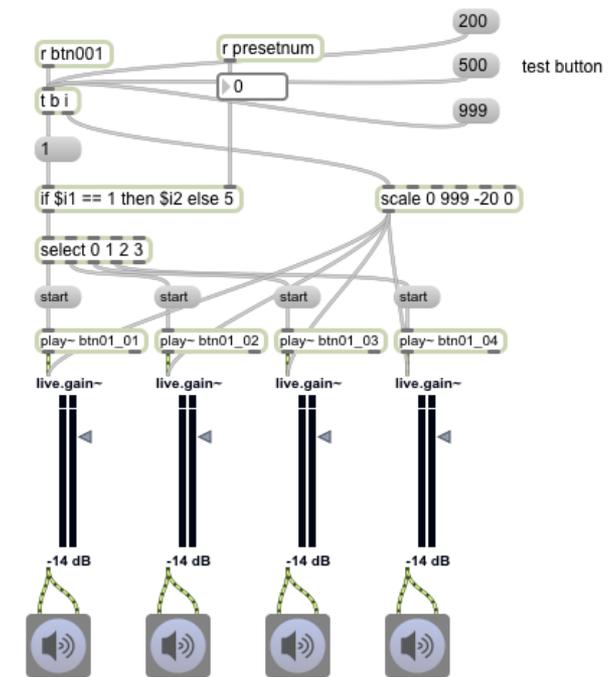
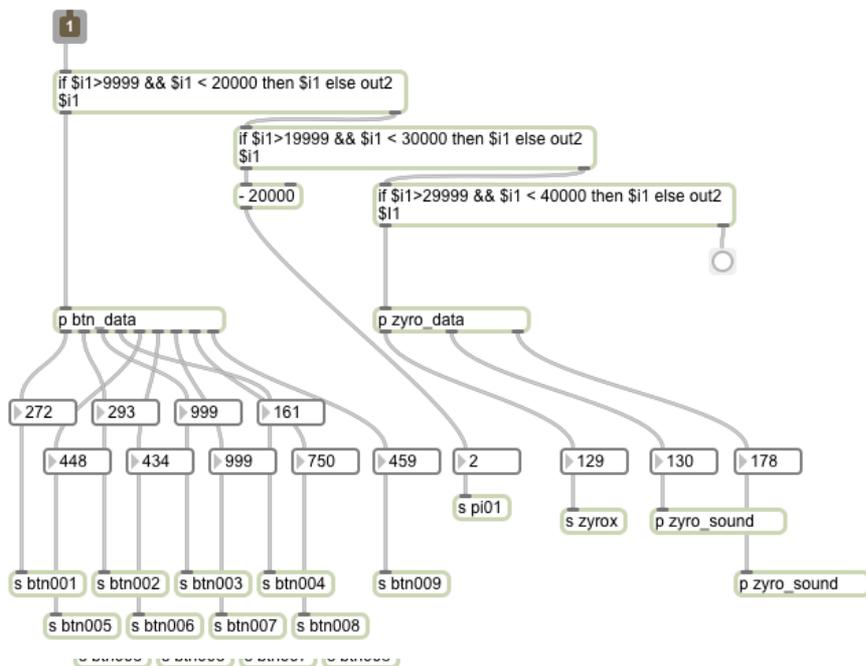
- GUIとファイル保存パッチ



内部の仕組み

○ MAX/MSP

- 情報選別パッチとフィードバックパッチ



ありがとうございます

