

自由部門：発表順番号（登録番号） 16（20054）

タイトル：「CanGakki ー組み替え可能な知育管楽器ー」

学校名：八戸工業高等専門学校

学生氏名： 類家 健永

西谷 宥人

上沢 悠三

古舘 源貴

指導教員： 釜谷 博行

送信側 arduino

```
#include <Wire.h>

//wind sensor
#define BME280_ADDRESS 0x76
unsigned long int hum_raw,temp_raw,pres_raw;
signed long int t_fine;

uint16_t dig_T1;
  int16_t dig_T2;
  int16_t dig_T3;
uint16_t dig_P1;
  int16_t dig_P2;
  int16_t dig_P3;
  int16_t dig_P4;
  int16_t dig_P5;
  int16_t dig_P6;
  int16_t dig_P7;
  int16_t dig_P8;
  int16_t dig_P9;
  int8_t  dig_H1;
  int16_t dig_H2;
  int8_t  dig_H3;
  int16_t dig_H4;
  int16_t dig_H5;
  int8_t  dig_H6;

double temp_act_ave = 0.0, press_act_ave = 0.0,hum_act_ave =0.0;

//choose sound
volatile int dic=1;

void setup()
{
  //wind sensor
```

```

uint8_t osrs_t = 1;           //Temperature oversampling x 1
uint8_t osrs_p = 1;           //Pressure oversampling x 1
uint8_t osrs_h = 1;           //Humidity oversampling x 1
uint8_t mode = 3;              //Normal mode
uint8_t t_sb = 6;              //Tstandby 10ms
uint8_t filter = 0;            //Filter off
uint8_t spi3w_en = 0;          //3-wire SPI Disable

uint8_t ctrl_meas_reg = (osrs_t << 5) | (osrs_p << 2) | mode;
uint8_t config_reg      = (t_sb << 5) | (filter << 2) | spi3w_en;
uint8_t ctrl_hum_reg    = osrs_h;

byte i;
double temp_act = 0.0, press_act = 0.0,hum_act=0.0;
signed long int temp_cal;
unsigned long int press_cal,hum_cal;
pinMode(13,OUTPUT); // led check

Serial.begin(9600);
Serial.println("MBE280Samplev2 by Yas. Sensor Init value Please Wait 5 SEC.....
");
Wire.begin();

writeReg(0xF2,ctrl_hum_reg);
writeReg(0xF4,ctrl_meas_reg);
writeReg(0xF5,config_reg);
readTrim();
// Sensor Wait 3
delay(3000);
// Sensor ave 5 times
for( i=0 ; i<5 ; i++){
    readData();

    temp_cal = calibration_T(temp_raw);
    press_cal = calibration_P(pres_raw);
    hum_cal = calibration_H(hum_raw);

```

```

        temp_act = (double)temp_cal / 100.0;
        press_act = (double)press_cal / 100.0;
        hum_act = (double)hum_cal / 1024.0;
        temp_act_ave += temp_act, press_act_ave += press_act, hum_act_ave +=
hum_act;
        delay(200);
    }
    temp_act_ave = temp_act_ave/5 , press_act_ave = press_act_ave/5 , hum_act_ave =
hum_act_ave/5;
    Serial.print("TEMP ave : ");
    Serial.print(temp_act_ave);
    Serial.print(" DegC  PRESS ave : ");
    Serial.print(press_act_ave);
    Serial.print(" hPa  HUM ave : ");
    Serial.print(hum_act_ave);
    Serial.println(" %");
    delay(1000);

    //choose sound
    attachInterrupt(0,dics,FALLING);

    //switch
    pinMode(4,INPUT);
    pinMode(5,INPUT);
    pinMode(6,INPUT);
    pinMode(7,INPUT);
    pinMode(8,INPUT);
    pinMode(9,INPUT);
    pinMode(10,INPUT);
    pinMode(11,INPUT);
}

void loop()
{
    while(dic==1)

```

```

{
    double temp_act = 0.0, press_act = 0.0,hum_act=0.0;
    signed long int temp_cal;
    unsigned long int press_cal,hum_cal;

    readData();

    temp_cal = calibration_T(temp_raw);
    press_cal = calibration_P(pres_raw);
    hum_cal = calibration_H(hum_raw);
    temp_act = (double)temp_cal / 100.0;
    press_act = (double)press_cal / 100.0;
    hum_act = (double)hum_cal / 1024.0;
// FUITA hantei
    if ( (temp_act > temp_act_ave ) && (press_act <= (press_act_ave - 0.25)) &&
(hum_act > (hum_act_ave + 1.0)))
    {
        origin();
    }
    delay(50);
}

while(dic==2)
{
    double temp_act = 0.0, press_act = 0.0,hum_act=0.0;
    signed long int temp_cal;
    unsigned long int press_cal,hum_cal;

    readData();

    temp_cal = calibration_T(temp_raw);
    press_cal = calibration_P(pres_raw);
    hum_cal = calibration_H(hum_raw);
    temp_act = (double)temp_cal / 100.0;
    press_act = (double)press_cal / 100.0;
    hum_act = (double)hum_cal / 1024.0;

```

```

// FUITA hantei v2
    if ( (temp_act > temp_act_ave ) && (press_act <= (press_act_ave - 0.50)) &&
(hum_act > (hum_act_ave + 1.0)))
    {
        tranp();
    }
    delay(50);
}
while(dic==3)
{
    double temp_act = 0.0, press_act = 0.0,hum_act=0.0;
    signed long int temp_cal;
    unsigned long int press_cal,hum_cal;

    readData();

    temp_cal = calibration_T(temp_raw);
    press_cal = calibration_P(pres_raw);
    hum_cal = calibration_H(hum_raw);
    temp_act = (double)temp_cal / 100.0;
    press_act = (double)press_cal / 100.0;
    hum_act = (double)hum_cal / 1024.0;
// FUITA hantei v2
    if ( (temp_act > temp_act_ave ) && (press_act <= (press_act_ave - 0.50)) &&
(hum_act > (hum_act_ave + 1.0)))
    {
        flute();
    }
    delay(50);
}
}

void readTrim()
{
    uint8_t data[32],i=0;
    Wire.beginTransmission(BME280_ADDRESS);
    Wire.write(0x88);

```

```

Wire.endTransmission();
Wire.requestFrom(BME280_ADDRESS,24);
while(Wire.available()){
    data[i] = Wire.read();
    i++;
}

Wire.beginTransmission(BME280_ADDRESS);    // Add 2014/04/06
Wire.write(0xA1);                          // Add 2014/04/06
Wire.endTransmission();                    // Add 2014/04/06
Wire.requestFrom(BME280_ADDRESS,1);        // Add 2014/04/06
data[i] = Wire.read();                     // Add 2014/04/06
i++;                                       // Add 2014/04/06

Wire.beginTransmission(BME280_ADDRESS);
Wire.write(0xE1);
Wire.endTransmission();
Wire.requestFrom(BME280_ADDRESS,7);        // Fix 2014/04/06
while(Wire.available()){
    data[i] = Wire.read();
    i++;
}

dig_T1 = (data[1] << 8) | data[0];
dig_T2 = (data[3] << 8) | data[2];
dig_T3 = (data[5] << 8) | data[4];
dig_P1 = (data[7] << 8) | data[6];
dig_P2 = (data[9] << 8) | data[8];
dig_P3 = (data[11]<< 8) | data[10];
dig_P4 = (data[13]<< 8) | data[12];
dig_P5 = (data[15]<< 8) | data[14];
dig_P6 = (data[17]<< 8) | data[16];
dig_P7 = (data[19]<< 8) | data[18];
dig_P8 = (data[21]<< 8) | data[20];
dig_P9 = (data[23]<< 8) | data[22];
dig_H1 = data[24];
dig_H2 = (data[26]<< 8) | data[25];

```

```

    dig_H3 = data[27];
    dig_H4 = (data[28]<< 4) | (0x0F & data[29]);
    dig_H5 = (data[30] << 4) | ((data[29] >> 4) & 0x0F); // Fix 2014/04/06
    dig_H6 = data[31]; // Fix 2014/04/06
}

void writeReg(uint8_t reg_address, uint8_t data)
{
    Wire.beginTransmission(BME280_ADDRESS);
    Wire.write(reg_address);
    Wire.write(data);
    Wire.endTransmission();
}

void readData()
{
    int i = 0;
    uint32_t data[8];
    Wire.beginTransmission(BME280_ADDRESS);
    Wire.write(0xF7);
    Wire.endTransmission();
    Wire.requestFrom(BME280_ADDRESS,8);
    while(Wire.available()){
        data[i] = Wire.read();
        i++;
    }
    pres_raw = (data[0] << 12) | (data[1] << 4) | (data[2] >> 4);
    temp_raw = (data[3] << 12) | (data[4] << 4) | (data[5] >> 4);
    hum_raw = (data[6] << 8) | data[7];
}

signed long int calibration_T(signed long int adc_T)
{
    signed long int var1, var2, T;

```



```

    var1 = (((adc_T >> 3) - ((signed long int)dig_T1<<1))) * ((signed long int)dig_T2)) >>
11;
    var2 = (((((adc_T >> 4) - ((signed long int)dig_T1)) * ((adc_T>>4) - ((signed long
int)dig_T1))) >> 12) * ((signed long int)dig_T3)) >> 14;

    t_fine = var1 + var2;
    T = (t_fine * 5 + 128) >> 8;
    return T;
}

```

```

unsigned long int calibration_P(signed long int adc_P)
{
    signed long int var1, var2;
    unsigned long int P;
    var1 = (((signed long int)t_fine)>>1) - (signed long int)64000;
    var2 = (((var1>>2) * (var1>>2)) >> 11) * ((signed long int)dig_P6);
    var2 = var2 + ((var1*((signed long int)dig_P5)<<1);
    var2 = (var2>>2)+(((signed long int)dig_P4)<<16);
    var1 = (((dig_P3 * (((var1>>2)*(var1>>2)) >> 13)) >>3) + (((signed long int)dig_P2)
* var1)>>1))>>18;
    var1 = (((32768+var1))*((signed long int)dig_P1))>>15);
    if (var1 == 0)
    {
        return 0;
    }
    P = (((unsigned long int)(((signed long int)1048576)-adc_P)-(var2>>12)))*3125;
    if(P<0x80000000)
    {
        P = (P << 1) / ((unsigned long int) var1);
    }
    else
    {
        P = (P / (unsigned long int)var1) * 2;
    }
    var1 = (((signed long int)dig_P9) * ((signed long int)(((P>>3) * (P>>3))>>13)))>>12;
    var2 = (((signed long int)(P>>2)) * ((signed long int)dig_P8))>>13;

```

```

    P = (unsigned long int)((signed long int)P + ((var1 + var2 + dig_P7) >> 4));
    return P;
}

unsigned long int calibration_H(signed long int adc_H)
{
    signed long int v_x1;

    v_x1 = (t_fine - ((signed long int)76800));
    v_x1 = (((((adc_H << 14) -(((signed long int)dig_H4) << 20) - (((signed long
int)dig_H5) * v_x1)) +
                ((signed long int)16384)) >> 15) * ((((((v_x1 * ((signed long int)dig_H6))
>> 10) *
                (((v_x1 * ((signed long int)dig_H3)) >> 11) + ((signed long int) 32768))) >>
10) + (( signed long int)2097152)) *
                ((signed long int) dig_H2) + 8192) >> 14));
    v_x1 = (v_x1 - (((((v_x1 >> 15) * (v_x1 >> 15)) >> 7) * ((signed long int)dig_H1)) >>
4));
    v_x1 = (v_x1 < 0 ? 0 : v_x1);
    v_x1 = (v_x1 > 419430400 ? 419430400 : v_x1);
    return (unsigned long int)(v_x1 >> 12);
}

void dics()
{
    dic++;
    if(dic>=4)
    {
        dic = 1;
    }
    delay(100);
}

void origin()
{
    if(digitalRead(4)==LOW)

```

```

{
    Serial.write('a');
}
else if(digitalRead(5)==LOW)
{
    Serial.write('b');
}
else if(digitalRead(6)==LOW)
{
    Serial.write('c');
}
else if(digitalRead(7)==LOW)
{
    Serial.write('d');
}
else if(digitalRead(8)==LOW)
{
    Serial.write('e');
}
else if(digitalRead(9)==LOW)
{
    Serial.write('f');
}
else if(digitalRead(10)==LOW)
{
    Serial.write('g');
}
else if(digitalRead(11)==LOW)
{
    Serial.write('h');
}
}

```

```

void tranp()
{
    if(digitalRead(4)==LOW&&digitalRead(5)==LOW&&digitalRead(6)==LOW)
    {

```

```

        Serial.write('o');
    }
    else if(digitalRead(5)==LOW&&digitalRead(6)==LOW)
    {
        Serial.write('n');
    }
    else if(digitalRead(4)==LOW&&digitalRead(6)==LOW)
    {
        Serial.write('m');
    }
    else if(digitalRead(4)==LOW&&digitalRead(5)==LOW)
    {
        Serial.write('l');
    }
    else if(digitalRead(6)==LOW)
    {
        Serial.write('k');
    }
    else if(digitalRead(5)==LOW)
    {
        Serial.write('j');
    }
    else if(digitalRead(4)==LOW)
    {
        Serial.write('i');
    }
    else
    {
        Serial.write('p');
    }
}

void flute()
{
    if(digitalRead(4)==LOW)
    {

```

```

        Serial.write('q');
    }
    else if(digitalRead(5)==LOW)
    {
        Serial.write('r');
    }
    else if(digitalRead(6)==LOW)
    {
        Serial.write('s');
    }
    else if(digitalRead(7)==LOW)
    {
        Serial.write('t');
    }
    else if(digitalRead(8)==LOW)
    {
        Serial.write('u');
    }
    else if(digitalRead(9)==LOW)
    {
        Serial.write('v');
    }
    else if(digitalRead(10)==LOW)
    {
        Serial.write('w');
    }else if(digitalRead(11)==LOW)
    {
        Serial.write('x');
    }
}

```

受信側 arduino

```
#include <MozziGuts.h>
#include <Oscil.h>
#include <tables/cos2048_int8.h>

Oscil <COS2048_NUM_CELLS, AUDIO_RATE> aSin(COS2048_DATA);

#define CONTROL_RATE 64

char stp;
char memo;

void setup(){
  Serial.begin(9600);
  startMozzi(CONTROL_RATE); // :)
  aSin.setFreq(0);
  Serial.println("Start reading.");
}

void updateControl(){
  // put changing controls in here

  delay(100);
  stp = (char)Serial.read();
  if(Serial.read() == -1)stp='z';
  Serial.print("stp=");
  Serial.println((char)stp);
  Serial.print("memo=");
  Serial.println((char)memo);
  if(stp>0){
    switch(stp){
      case 'a':
        if(stp != memo)aSin.setFreq(523);
        break;
```

```

    case 'b':
    if(stp != memo)aSin.setFreq(587);
    break;

    case 'c':
    if(stp != memo)aSin.setFreq(659);
    break;

    case 'd':
    if(stp != memo)aSin.setFreq(698);
    break;

    case 'e':
    if(stp != memo)aSin.setFreq(784);
    break;

    case 'f':
    if(stp != memo)aSin.setFreq(880);
    break;

    case 'g':
    if(stp != memo)aSin.setFreq(988);
    break;

    case 'h':
    if(stp != memo)aSin.setFreq(1047);
    break;

    case 'z':
    if(stp != memo)aSin.setFreq(0);
    break;
}
}
memo = (char)stp;
}

```

```
int updateAudio(){
    return aSin.next(); // return an int signal centred around 0
}

void loop(){
    audioHook(); // required here
}
```


Web サイト

トップページ (html.erb)

```
<header class="page-header wrapper">
  <h1><a href="/"></a></h1>
  <nav>
    <ul class="main-nav">
      <li><a href="/start">はじめに</a></li>
      <li><a href="/make">各楽器の作り方</a></li>
      <li><a href="/movie">練習動画一覧</a></li>
      <li><a href="/ongen">追加音源</a></li>
    </ul>
  </nav>
</header>

<div class="home-content wrapper">
  <h2 class="page-title">CanGakki 公式サポートサイト</h2>
  <p><span class="top-arrow">↓</span>まずはこのボタンを押そう</p>
  <a class="botton" href="/start">はじめに</a>
</div>
</div>
```

楽器や練習曲の選択画面（このプログラムを基にそれぞれのページを作成）

```
header class="page-header wrapper">
  <h1><a href="/"></a></h1>
  <nav>
    <ul class="main-nav">
      <li><a href="/start">はじめに</a></li>
      <li><a href="/make">各楽器の作り方</a></li>
      <li><a href="/movie">練習動画一覧</a></li>
      <li><a href="/ongen">追加音源</a></li>
    </ul>
  </nav>
</header>
```

```

<div class="normal-content">
  <h2 class="page-title">タイトル </h2>
  <div class="make-list">
    <div class="item">
      <a href="/make_cangakki" class="botton">項目 1 </a>
    </div>
    <div class="item">
      <a href="/movie" class="botton">項目 2 </a>
    </div>
    <div class="item">
      <a href="/movie" class="botton">項目 3 </a>
    </div>
  </div>
</div>

<footer>
  <div class="wrapper">
    <p><small>2019 プロコン自由部門</small></p>
  </div>
</footer>

```

説明画面（このプログラムを基にそれぞれのページを作成）

```

<header class="page-header wrapper">
  <h1><a href="/"></a></h1>
  <nav>
    <ul class="main-nav">
      <li><a href="/start">はじめに</a></li>
      <li><a href="/make">各楽器の作り方</a></li>
      <li><a href="/movie">練習動画一覧</a></li>
      <li><a href="/ongen">追加音源</a></li>
    </ul>
  </nav>
</header>

```

```

<div class="normal-content wrapper">
  <h2 class="page-title">CanGakki の作り方</h2>
  <p>
    CanGakki は演奏したい楽器を自分で選んで組み立てないと、演奏することができません。
    このサイトでは、自分の力で楽器を作り、演奏の練習を行うまでの流れを分かりやすく記しています。
    自分一人で自由に演奏できるようになるまでこのサイトを見ながら頑張ってください！！
    早速下のボタンをクリックして最初の楽器を作ってみましょう！！
  </p>
  <div class="page-main">
    <p>
      手順 1 =====
      =====
    </p>
    
    <p>
      手順 2 =====
      =====
    </p>
    
    <p>
      手順 3 =====
      =====
    </p>
    
    <p>
      完成！！
    </p>
  </div>
  <a class="botton start-botton" href="/movie">曲の練習に進む</a>
</div>

<footer>
  <div class="wrapper">

```

<p><small>2019 プロコン自由部門</small></p>
</div>
</footer>

CSS ファイル

```
@charset "UTF-8";
```

```
html{  
  font-size:100%;  
}
```

```
body{  
  
  line-height:1.7;  
  color:#432;  
}
```

```
a{  
  text-decoration:none;  
}
```

```
img{  
  max-width:100%;  
}
```

//ヘッダー関連

```
.logo{  
  width:210px;  
  margin-top:14px;  
}
```

```
.main-nav{  
  display:flex;  
  font-size:1.25rem;  
  text-transform:uppercase;  
  margin-top:34px;  
  list-style:none;  
}
```

```
.main-nav li{  
  margin-left:36px;  
}
```

```

.main-nav a{
  color:#432;
}
.main-nav a:hover{
  color:#0bd;
}
.page-header{
  display:flex;
  justify-content:space-between;
}
//

.wrapper{
  max-width:1100px;
  margin:0 auto;
  padding:0 4%;
}
.home-content{
  text-align:center;
  margin-top:10%;
}
.home-content p{
  font-size:1.125rem;
  margin:10px 0 42px;
}
.page-title{
  font-size:4rem;
  text-transform:uppercase;
  font-weight:normal;
}
.botton{
  font-size:1.375rem;
  background:#0bd;
  color:#fff;
  border-radius:5px;
  padding:18px 32px;
}

```

```

}
.botton:hover{
    background:#0090aa;
}
#home{
    background-image :url(/images/logo.png);
    min-height:100vh;
}
#home .page-title{
    text-transform:none;
}
.big-bg{
    background-size: cover;
    background-position: center top;
    background-repeat: no-repeat;
}
footer{
    background:#432;
    text-align:center;
    padding:26px 0;
}
footer p{
    color:#fff;
    font-size:0.875rem;
}
.top-arrow{
    font-size: 2rem;
}
.normal-content{
    text-align: center;
    margin:10%;
}
.normal-content p{
    margin:5% auto;
}

```

```
.start-content p{
  font-size:1.125rem;
  margin:10px 0 42px;
}
.start-botton{
  margin:30px;
}
.make-list{
  display:grid;
  gap:26px;
  grid-template-columns:repeat(auto-fit,minmax(240px,1fr));
  margin-top:6%;
  margin-bottom:50px;
}
@media(max-width:600px){

}
```