

# 放射線計測機模型の詳細

## 使用材料

- ・ ELEGOO UNO R3 (Arduino) × 1
- ・ キャラクタディスプレイ (LCD1602) × 1
- ・ アクティブブザー (LCD) × 1
- ・ 可変抵抗 (10K $\Omega$ ) (VR) × 1
- ・ 抵抗器 (100 $\Omega$ ) (R) × 2
- ・ タクトスイッチ × 1
- ・ スライドスイッチ × 1

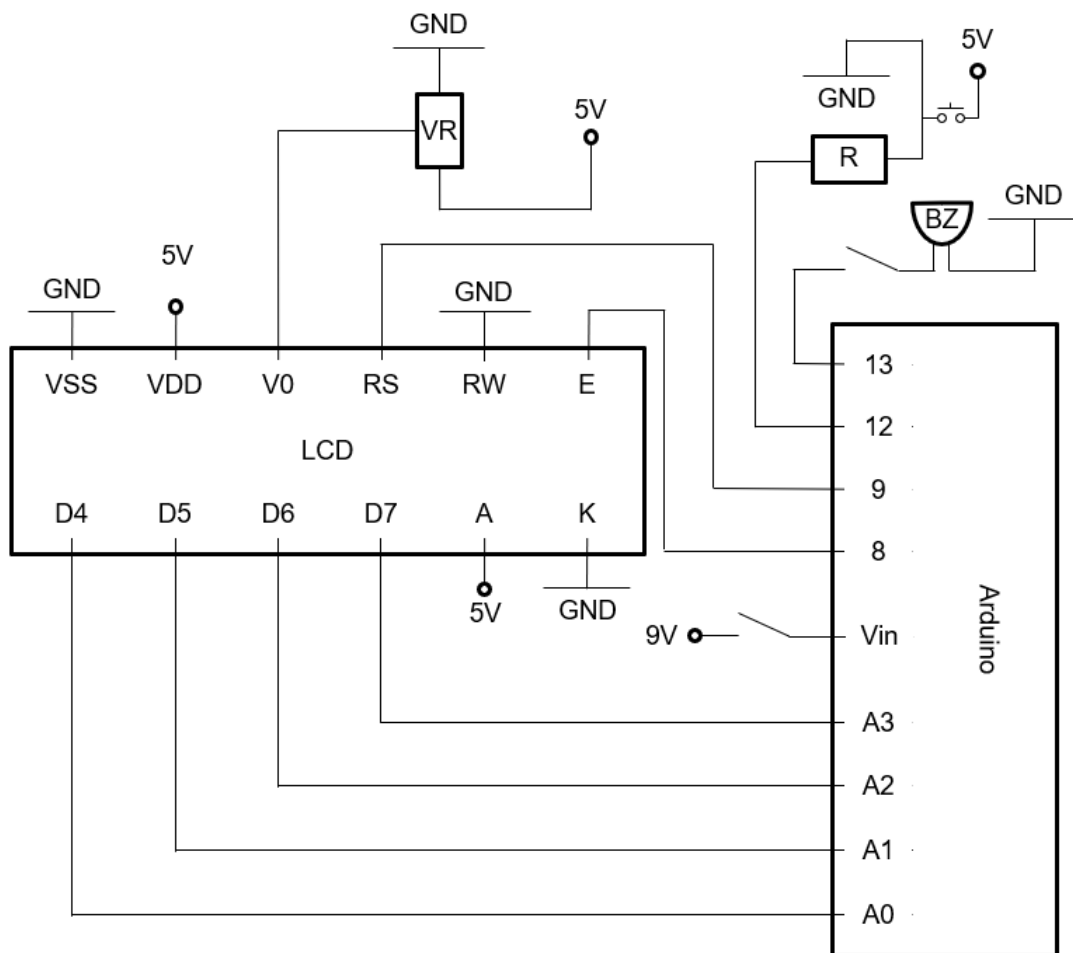


図3. 放射線計測機模型の回路図

以下にArduinoのコードを記す

```
#include <Wire.h>
#include <LiquidCrystal.h>

//ディスプレイ表示に関する数値
long randomNumber;//ランダム生成の数値
long countVolume;//表示する元値
long DcountVolume;//表示する元値の30秒経過後の基準値

long ones;//表示する数字の1の位
long pointOnes;//表示する数字の0.1の位

char onesPlace;//表示する数字の1の位(キャラクタディスプレイで表示できる値)
char pointOnesPlace;//表示する数字の0.1の位(キャラクタディスプレイで表示できる値)

int trialCount;//表示する手前の試行回数
int count;//時間経過を計測する試行回数

LiquidCrystal lcd(9, 8, A0, A1, A2, A3); /*RS E D4 D5 D6 D7*/

void setup() {
    while (digitalRead(12) == HIGH) { //12のボタンが押されるまで待機
    }
    lcd.begin(15, 2);
    Serial.begin(9600);
    randomSeed(analogRead(5));
}

void loop() {
    char onesPlace[8];
    char pointOnesPlace[9];
    randomNumber = random(1, 30); // 1から30の乱数を生成
    countVolume = randomNumber;//countVolumeの初期値をrandomNumberに設定

    //徐々にカウントアップさせる為のコード
    for(count=0; count<40; count++){
        for (trialCount=0 ;trialCount<5 ;trialCount++){
            randomNumber = random(1, 100); // 1から100の乱数を生成
            Serial.println(randomNumber);
        }
    }
}
```

```

if( randNumber%5==0){
    countVolume++;
    tone(13,2216,50);//ブザーの発音(pin,周波数,音の長さ)
}
if( randNumber%7==0){
    countVolume--;
}
}

//countVolumeをキャラクタディスプレイで表示できる数へ変換
ones = countVolume/10;
ones = 0.5; //次のラウンドを小数点以下切り捨てにするために0.5を減じる
round(ones);
sprintf(onesPlace,"%d",ones);

point0nes = countVolume%10;//10で割った余り(1の位)を求める
round(point0nes);
sprintf(point0nesPlace,"%d",point0nes);

//キャラクタディスプレイの表示に関するコード
//1行目
lcd.setCursor(0, 0);
lcd.print("Count Volume");

//2行目
lcd.setCursor(0, 1);
lcd.print( onesPlace );

lcd.setCursor(1, 1);
lcd.print( "." );

lcd.setCursor(2, 1);
lcd.print( point0nesPlace );

lcd.setCursor(4, 1);
lcd.print( "micro Gy/h" );

delay(400);
}

//30秒経過後に数字の揺らぎを小さくするためのコード
DcountVolume = countVolume;

```

```

for(count=0; count<32767; count++){
  for (trialCount=0 ;trialCount<5 ;trialCount++){
    randomNumber = random(1, 100); // 1から100の乱数を生成
    Serial.println(randNumber);

    if( randNumber%2==0){
      countVolume = DcountVolume;
    }
    if( randNumber%3==0){
      countVolume = countVolume+1;
    }
    if(randNumber%5==0){
      countVolume = countVolume-1;
      tone(13, 2216, 50); //ブザーの発音、(pin, 周波数, 音の長さ)
    }
    if(randNumber%7==0){
      countVolume = countVolume+2;
    }
    if(randNumber%11==0){
      countVolume = countVolume-2;
      tone(13, 2216, 50); //ブザーの発音、(pin, 周波数, 音の長さ)
    }
    if(randNumber%13==0){
      countVolume = countVolume+3;
    }
    if(randNumber%17==0){
      countVolume = countVolume-3;
      tone(13, 2216, 50); //ブザーの発音、(pin, 周波数, 音の長さ)
    }
  }
  //countVolumeをキャラクタディスプレイで表示できる数へ変換
  ones = countVolume/10;
  ones = 0.5; //次のラウンドを小数点以下切り捨てにするために0.5を減じる
  round(ones);
  sprintf(onesPlace, "%d", ones);

  point0nes = countVolume%10; //10で割った余り(1の位)を求める
  sprintf(point0nesPlace, "%d", point0nes);

  //キャラクタディスプレイの表示に関するコード
  //1行目
  lcd.setCursor(0, 0);
  lcd.print("Count Volume");

```

```
//2行目  
    lcd.setCursor(0, 1);  
    lcd.print( onesPlace );  
  
    lcd.setCursor(1, 1);  
    lcd.print( "." );  
  
    lcd.setCursor(2, 1);  
    lcd.print( point0nesPlace );  
  
    lcd.setCursor(4, 1);  
    lcd.print( "micro Gy/h" );  
  
    delay(400);  
}  
}
```

## 動作内容

- ① DisitalのI2pinへ接続されたタクトスイッチが押されるまで待機する
- ② ボタンが押されたら、ランダムに1~30の数値を生成し、countVolumeに記憶する
- ③ ランダムに1~100の数値を生成し、その値が5の倍数であれば、countVolumeに1加算してブザーを鳴らし、7の倍数であればcountVolumeから1減じる
- ④ ③の動作を5回繰り返し、countVolumeをLCDで表示できる数値へ変換し、  
CountVolume  
X.Y micro Gy/h  
と表示させる。(X.YはcountVolumeを1/10した数値)
- ⑤ ③~④の動作を約30秒間繰り返す
- ⑥ 30秒経過時点のcountVolumeをDcountVolumeに記憶する
- ⑦ countVolumeへDcountVolumeの値を上書きする
- ⑧ ランダムに1~100の数値を生成し、その値が3、7、13の倍数であればそれぞれ1、2、3をcountVolumeに加算してブザーを鳴らし、5、11、17の倍数であればそれぞれ1、2、3をcountVolumeから減じる
- ⑨ ⑦の動作を5回繰り返し、countVolumeをLCDで表示できる数値へ変換し、  
CountVolume  
X.Y micro Gy/h  
と表示させる。(X.YはcountVolumeを1/10した数値)
- ⑩ 以降は⑦~⑨の動作を繰り返す