

簡易信号発生ユニット(ALC装備)

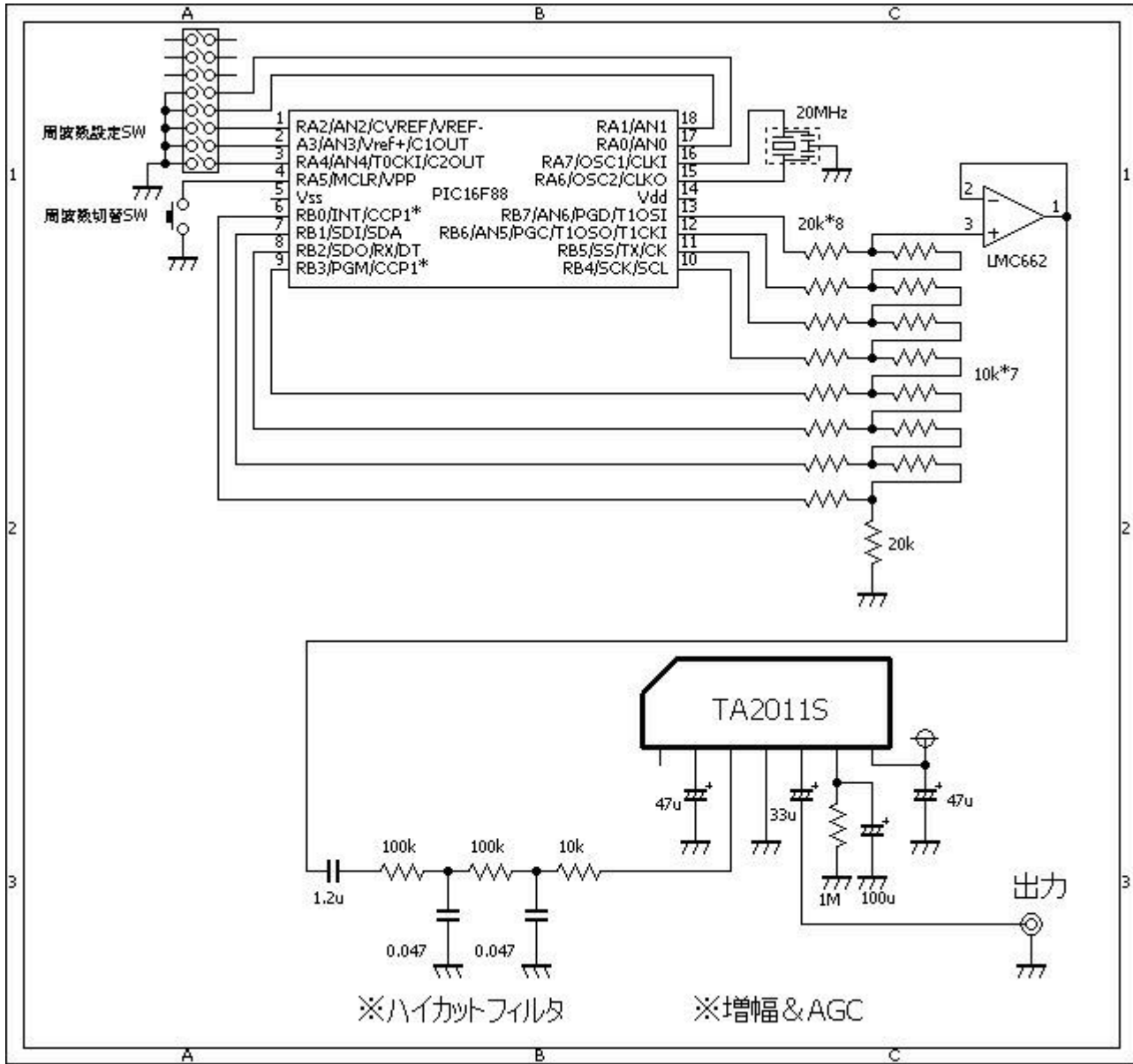
概要

真空管アンプを作成した場合、そのアンプの周波数特性を自動的に測定したいと考えているのですが、そのために先ず必要となるのが信号発生器です。理想的には10Hz~100kHzまでを出力が一定のまま連続的にスイープできれば良いのですが、今回は周波数の範囲を低域限定(10Hz~100Hz)とし、出力が一定になるようにALC(自動レベル制御)回路を付加しました。ALCについては、「自動レベル制御(ALC)の実験」を参照ください。

動作原理

- PICで正弦波データ(72バイト使用)を連続的にD/A(ラダー抵抗方式)出力する。
ディップSWで周波数を10Hz単位で設定(10段階、最大100Hz)できるようにする。
- その出力をハイカットフィルタに通す。(低い周波数よりも高い周波数ではレベルがかなり低くなる)
- ALC(自動レベル制御)回路を通して出力レベルを一定にする。

回路図



ソースコード

sin3.c

```

//*****
*
static unsigned char tbl[72] = {
    127,
    138,
    149,
    160,
    170,
    181,
    191,
    200,
    209,

```

217,
224,
231,
237,
242,
246,
250,
252,
254,
254,
254,
252,
250,
246,
242,
237,
231,
224,
217,
209,
200,
191,
181,
170,
160,
149,
138,
127,
116,
105,
94,
84,
73,
63,
54,
45,
37,
30,
23,
17,
12,
8,
4,
2,
0,
0,
0,
2,
4,
8,
12,

```
17,  
23,  
30,  
37,  
45,  
54,  
63,  
73,  
84,  
94,  
105,  
116,  
};  
  
//*****  
*  
  
void sin10hz()  
{  
    unsigned char cnt;  
    while (1) {  
        for (cnt = 0; cnt < 72; cnt++) {  
            PORTB = tbl[cnt];  
            Delay_us(1387);  
        }  
        if (PORTA.F5 == 0)  
            return;  
    }  
}  
  
void sin20hz()  
{  
    unsigned char cnt;  
    while (1) {  
        for (cnt = 0; cnt < 72; cnt++) {  
            PORTB = tbl[cnt];  
            Delay_us(693);  
        }  
        if (PORTA.F5 == 0)  
            return;  
    }  
}  
  
void sin30hz()  
{  
    unsigned char cnt;  
    while (1) {  
        for (cnt = 0; cnt < 72; cnt++) {  
            PORTB = tbl[cnt];  
            Delay_us(460);  
        }  
    }  
}
```

```
        if (PORTA.F5 == 0)
            return;
    }
}

void sin40hz()
{
    unsigned char cnt;
    while (1) {
        for (cnt = 0; cnt < 72; cnt++) {
            PORTB = tbl[cnt];
            Delay_us(345);
        }
        if (PORTA.F5 == 0)
            return;
    }
}

void sin50hz()
{
    unsigned char cnt;
    while (1) {
        for (cnt = 0; cnt < 72; cnt++) {
            PORTB = tbl[cnt];
            Delay_us(275);
        }
        if (PORTA.F5 == 0)
            return;
    }
}

void sin60hz()
{
    unsigned char cnt;
    while (1) {
        for (cnt = 0; cnt < 72; cnt++) {
            PORTB = tbl[cnt];
            Delay_us(228);
        }
        if (PORTA.F5 == 0)
            return;
    }
}

void sin70hz()
{
    unsigned char cnt;
    while (1) {
        for (cnt = 0; cnt < 72; cnt++) {
            PORTB = tbl[cnt];
            Delay_us(195);
        }
    }
}
```

```
    }
    if (PORTA.F5 == 0)
        return;
}

void sin80hz()
{
    unsigned char cnt;
    while (1) {
        for (cnt = 0; cnt < 72; cnt++) {
            PORTB = tbl[cnt];
            Delay_us(170);
        }
        if (PORTA.F5 == 0)
            return;
    }
}

void sin90hz()
{
    unsigned char cnt;
    while (1) {
        for (cnt = 0; cnt < 72; cnt++) {
            PORTB = tbl[cnt];
            Delay_us(151);
        }
        if (PORTA.F5 == 0)
            return;
    }
}

void sin100hz()
{
    unsigned char cnt;
    while (1) {
        for (cnt = 0; cnt < 72; cnt++) {
            PORTB = tbl[cnt];
            Delay_us(137);
        }
        if (PORTA.F5 == 0)
            return;
    }
}

//*****
*

void main() {
    CMCON = 0b00000111; // コンパレータは使用しない。
    ANSEL = 0b00000000; // A/D変換は使用しない。
}
```

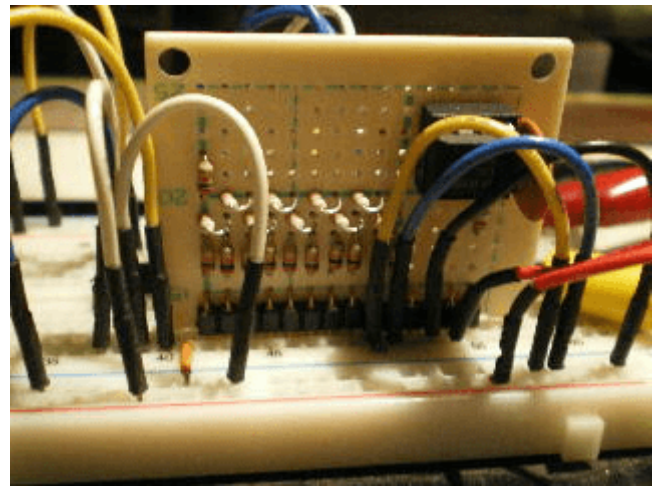
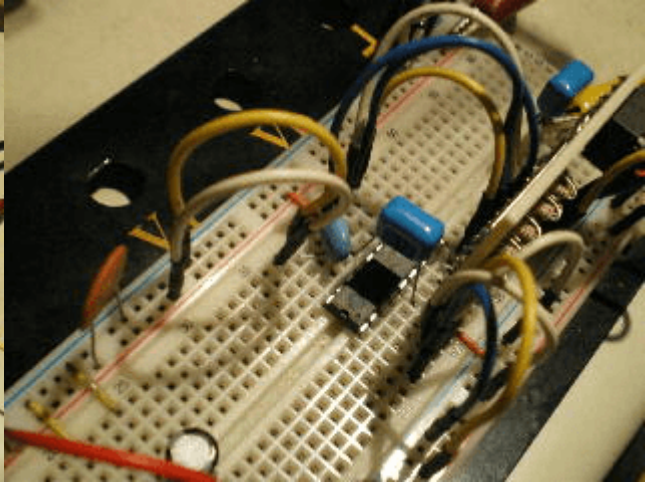
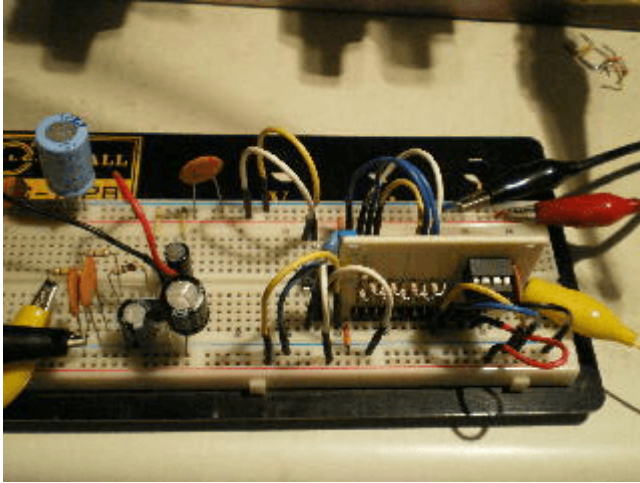
```
TRISA = 0b11111111;
TRISB = 0b00000000;
while (1) {
    switch (PORTA & 0b00011111) {
        case 0:
            break;
        case 1:
            sin10hz();
            break;
        case 2:
            sin20hz();
            break;
        case 3:
            sin30hz();
            break;
        case 4:
            sin40hz();
            break;
        case 5:
            sin50hz();
            break;
        case 6:
            sin60hz();
            break;
        case 7:
            sin70hz();
            break;
        case 8:
            sin80hz();
            break;
        case 9:
            sin90hz();
            break;
        case 10:
            sin100hz();
            break;
        default:
            break;
    }
}

//*****
*
```

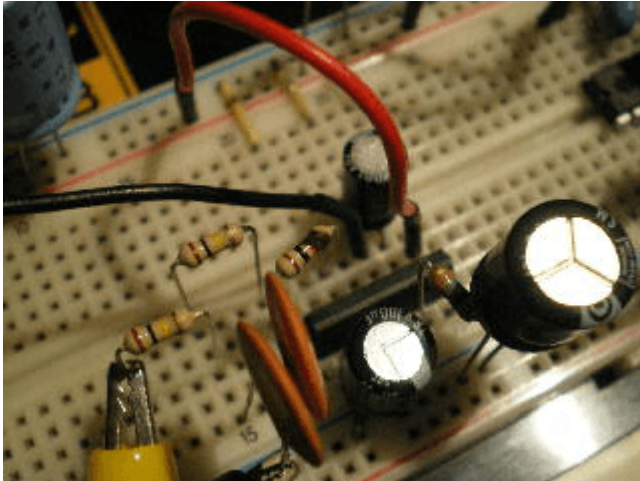
動作確認

いつものブレッドボードで確認しました。左:ALC部分、右側の手前がD/A部分、奥がPIC部分です。

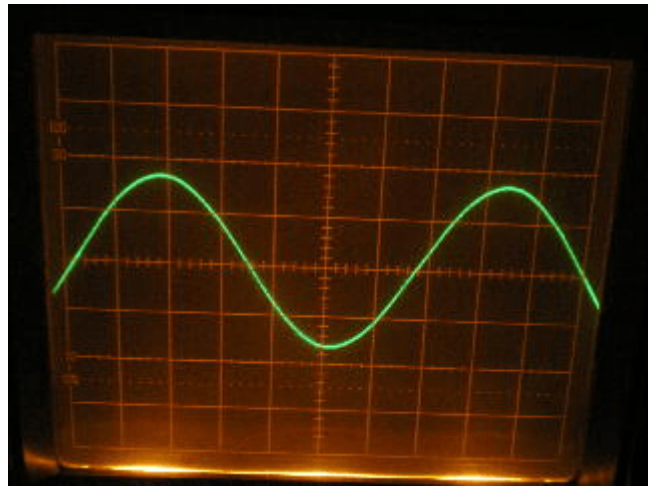
右:PIC部分です□20MHzのセラミック発振子も見えます。



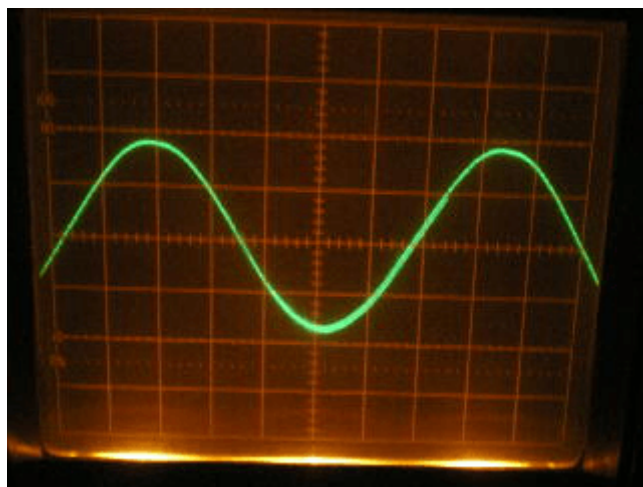
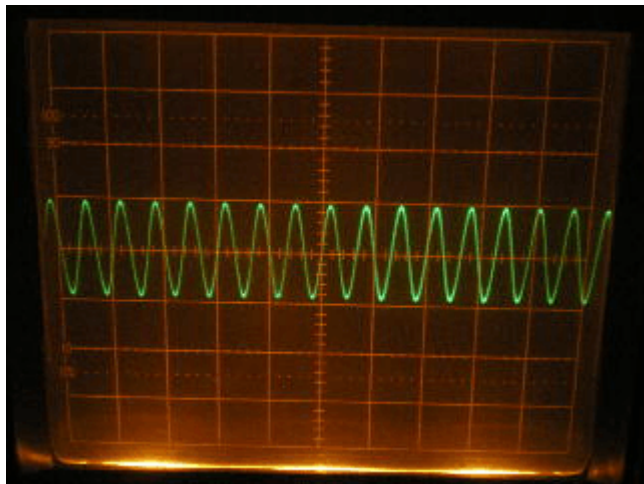
左:D/A(ラダー抵抗方式)部分です。 右:ALC部分です。



ハイカットフィルタの直後の波形です。 <左:10Hz> ハイカットフィルタの直後の波形です。 <

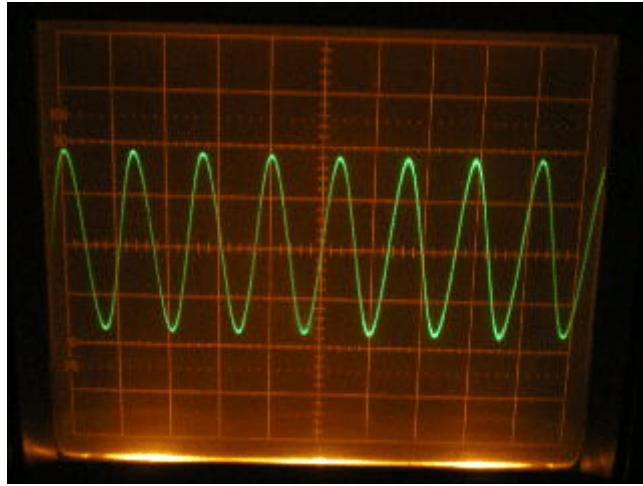


右:100Hz>波形がかなり小さくなりました。

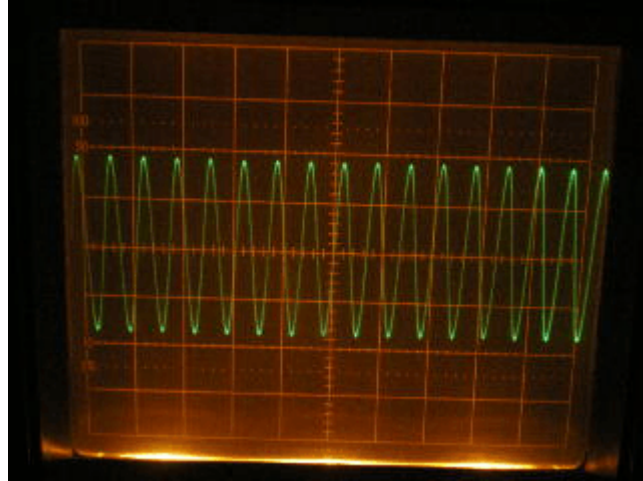


ALC直後の波形です□<10Hz>出力一定!





ALC直後の波形です□<50Hz>出力一定!



ALC直後の波形です□<100Hz>出力一定!



如何ですか?これで低域限定ですがアンプの周波数特性の自動計測の第一歩が出来ました 😊

著作権表示 **copyright notice**

このページは稲崎様の閉鎖したHPのコピーで、著作権は稲崎様にあります。詳細 This page is a copy of Mr. Inasaki's closed website, and the copyright is held by him. [Details](#)

From:

<http://www.deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://www.deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:56>

Last update: **2025/10/17 14:29**

