介面實驗

實驗七

類比訊號處理

班級：機械 3A

學號：108303013

姓名：黃鉦淳

日期：111/01/15

介面實驗工作日誌

實驗七　　111 年 01 月 15 日

組別		姓名	黃鉦淳	學號	108303013
實驗起始時間		110/12/17		費時	1 個月
實驗結束時間		111/01/15			
所遭遇問題	電路不會接				
解決方法	狂問學長				
完成項目及心得	短時間不想碰電路了亏				
調查	□是否有看課程講解影片 是否實用？有何建議？		□是否有看實驗教學影片 是否實用？有何建議？		

一、程式碼

實驗一 反向放大電路及非反向放大電路實作

Language : C

```
#include "c4mlib.h"

#define arrSize 21

void ADCPostPro_step(void *VoidStr_p);
void SPIDACTrm_step(void *VoidStr_p);

void ADC_init();
void SPI_init();
void timer3_init();

typedef struct
{
    uint16_t *InData_p;             /*Pointer points to the Input Data Source */
    uint16_t *DataList_p;           /* Pointer points to the buffer array */
    uint8_t DataLength;             /* Length of Datalist */
    uint8_t DataCount;              /*Data Count of the data in list*/
    uint8_t TaskId;                 // The TaskId got after registered
    uint8_t NextTaskNum;              // Number of Next Task
    uint8_t *NextTask_p;            // pointer to the List of TaskId for NextTasks
    volatile uint8_t TrigCount; // Triggered Counter
} ADCPostProStr_t;

typedef struct
{
    uint16_t OutData;
    uint16_t *OutData_p;            /*Pointer points to the Output Data Source */
    uint16_t *DataList_p;           /* Pointer points to the Out buffer array */
    uint8_t DataLength;             /* Length of Datalist */
    uint8_t DataCount;              /*Data Count of the data in list*/
    uint8_t TaskId;                 // The TaskId got after registered
    uint8_t NextTaskNum;              // Number of Next Task
```

```
    uint8_t *NextTask_p;            // pointer to the List of TaskId for NextTasks
    volatile uint8_t TrigCount; // Triggered Counter
} DACPreProStr_t;

typedef struct
{
    uint8_t Mode;       /* Transmit Mode*/
    uint8_t CardId;     /* Card Identification Number */
    uint8_t RegAdd;     /* Register Id of the register of this Card*/
    uint8_t Bytes;      // Bytes of the Data to Transmit;
    uint16_t *Data_p; // Data to be Transmit
    uint8_t Counter;
    uint8_t TaskId;                 // The TaskId got after registered
    uint8_t NextTaskNum;            // Number of Next Task
    uint8_t *NextTask_p;            // pointer to the List of TaskId for NextTasks
    volatile uint8_t TrigCount; // Triggered Counter
} SpiDacTrmStr_t;

#define ADCPOSTPRO_LAY(ADCPostPro_str, ListNum, _NextTaskNum,
InDataAdd) \
    uint16_t ADCPostPro_str##_DataList[ListNum];                                \
    uint8_t ADCPostPro_str##_NextTaskList[_NextTaskNum];
\
    ADCPostProStr_t ADCPostPro_str =
\

{                                                                               \
            .InData_p = InDataAdd,
\
            .DataList_p = ADCPostPro_str##_DataList,                            \
            .DataLength = ListNum,
\
            .DataCount = 0,
\
            .TaskId = 0,
\
            .NextTaskNum = _NextTaskNum,
\
```

```
                .NextTask_p = ADCPostPro_str##_NextTaskList,
\
                .TrigCount = 0}

#define DACPrePro_LAY(DACPreProStr, ListNum, _NextTaskNum) \
    uint16_t DACPreProStr_DataList[ListNum];                    \
    uint8_t DACPreProStr_NextTaskList[_NextTaskNum];            \
    DACPreProStr_t DACPreProStr = {                             \
        .DataList_p = DACPreProStr_DataList,                    \
        .DataLength = ListNum,                                  \
        .DataCount = 0,                                         \
        .TaskId = 0,                                            \
        .OutData_p = 0,                                         \
        .NextTaskNum = _NextTaskNum,                            \
        .NextTask_p = DACPreProStr_NextTaskList}

#define SPIDACTrm_LAY(SpiDacTrmStr, _NextTaskNum)
\
    uint8_t SpiDacTrmStr_NextTaskList[1] = {0};
\
    SpiDacTrmStr_t SpiDacTrmStr =
{                                                                   \
        .Mode = 2,      /* Transmit Mode*/
\
        .CardId = 1,    /* Card Identification Number */
\
        .RegAdd = 80, /*參考  操作控制參數快查表  */
\
        .Bytes = 2,     /*參考  操作控制參數快查表  */
\
        .Data_p = 0,    /*Data to be Transmit */
\
        .Counter = 0,
\
        .TaskId = 0,                                /*The TaskId got after
registered */                   \
        .NextTaskNum = _NextTaskNum,                /*Number of Next Task
*/                                  \
```

```
        .NextTask_p = SpiDacTrmStr_NextTaskList, /*pointer to the List of TaskId
for NextTasks*/ \
        .TrigCount = 0}

int main()
{
    C4M_DEVICE_set();

    //設定 Timer
    TIM3_HW_LAY();
    hardware_set(&TIM1_3HWSet_str);

    //設定 Timer3 中斷
    TIMHWINT_LAY(TIMINT_Str, 3, 1);
    timer3_init();

    //設定 ADC
    ADC_HW_LAY();
    hardware_set(&ADCHWSet_str);
    ADC_init();

    //設定 ADC 中斷
    ADCHWINT_LAY(ADCHWINT_Str, 0, 2);
    hardware_set(&ADCHWINT_Str);

    //設定 SPI
    SPI_init();

    unsigned int ADC_readData;

    ADCPOSTPRO_LAY(ADC_PostPro_Str, arrSize, 0, &ADC_readData);
    DACPrePro_LAY(DAC_PostPro_Str, arrSize, 0);
    SPIDACTrm_LAY(SPI_DAC_Str, 0);

    HMI_snget_matrix(HMI_TYPE_UI16, 1, DAC_PostPro_Str.DataLength,
DAC_PostPro_Str.DataList_p);
    SPI_DAC_Str.Data_p = DAC_PostPro_Str.DataList_p;
```

```
	//設定降頻器配置
	int period[2];
	for (int i = 0; i < sizeof(period) / sizeof(int); i++)
		period[i] = 10;
	// int period = 20;

	FREQREDU_LAY(FreqRedu_Str, 1, 2, &OCR3A, 2, &period);

	int8_t freq_TaskID;
	freq_TaskID = FreqRedu_reg(&FreqRedu_Str, &SPIDACTrm_step,
&SPI_DAC_Str, 1, 0);

	FreqRedu_en(&FreqRedu_Str, freq_TaskID, ENABLE);

	uint8_t TaskID[3];

	//將降頻器登入進 Timer 中斷
	TaskID[0] = HWInt_reg(&TIMINT_Str, &FreqRedu_step, &FreqRedu_Str);
	HWInt_en(&TIMINT_Str, TaskID[0], ENABLE);

	//將 ADC 結果登入進 ADC 中斷
	RT_REG_IO_LAY(ADC_result_Str, 0, &ADCL, 2, (uint8_t
*)&ADC_readData);
	TaskID[1] = HWInt_reg(&ADCHWINT_Str, &RealTimeRegGet_step,
&ADC_result_Str);
	// HWInt_en(&ADCHWINT_Str, TaskID[1], ENABLE);

	//將 ADC_postPro_step 結果登入進 ADC 中斷
	TaskID[2] = HWInt_reg(&ADCHWINT_Str, &ADCPostPro_step,
&ADC_PostPro_Str);
	// HWInt_en(&ADCHWINT_Str, TaskID[2], ENABLE);

	sei();

	while (1)
	{
		;
	}
```

```
    return 0;
}

void ADC_init()
{
    //設定外部參考電壓
    REGFPT(&ADMUX, 0xC0, 6, 1);

    //設定 10 位元轉換靠右
    REGFPT(&ADMUX, 0x20, 5, 0);

    //設定非連續或觸發轉換
    REGFPT(&ADCSRA, 0x20, 5, DISABLE);

    //設定 ADC1 F1 輸入
    REGFPT(&DDRF, 0x02, 0, 0);

    //設定致能 ADC
    REGFPT(&ADCSRA, 0x80, 7, ENABLE);

    //設定致能 ADC Interrupt
    REGFPT(&ADCSRA, 0x08, 3, ENABLE);

    //設定 ADC 時脈  clk/128
    REGFPT(&ADCSRA, 0x07, 0, 7);

    //設定 ADC1 單通道
    REGFPT(&ADMUX, 0x1f, 0, 1);
}

void SPI_init()
{
    REGFPT(&DDRD, 0x20, 5, 0x01); //設定 PD5 為 MCP4922 晶片選擇
    REGFPT(&DDRB, 0x07, 0, 0x07); //設定 PB1(SCK) / PB2(MOSI)為輸出

    //主板設定
    REGFPT(&SPSR, 0x01, SPI2X, 1); //設定雙倍工作時脈
```

```
    REGFPT(&SPCR, 0x03, SPR0, 0);    // SPI_FreqDivide_4
    REGFPT(&SPCR, 0x04, CPHA, 0);    //前收後送
    REGFPT(&SPCR, 0x08, CPOL, 0);    //設定前緣為上
    REGFPT(&SPCR, 0x10, MSTR, 1);    //設定為主板
    REGFPT(&SPCR, 0x20, DORD, 0);    //高位元先送
    REGFPT(&SPCR, 0x40, SPE, 1);     // SPI 致能
}

void timer3_init()
{
    // normal mode
    REGFPT(&TCCR3A, 0x03, 0, 0);
    // normal mode
    REGFPT(&TCCR3B, 0x18, 3, 1);
    //設定 timer 時脈  clk/1024
    REGFPT(&TCCR3B, 0x07, 0, 4);

    OCR3A = 269;
    //設定 timer3A 致能
    REGFPT(&ETIMSK, 0x10, 4, 1);
}

void ADCPostPro_step(void *VoidStr_p)
{
    volatile ADCPostProStr_t *Str_p = (ADCPostProStr_t *)VoidStr_p;

    Str_p->DataList_p[Str_p->DataCount] = *(Str_p->InData_p);

    if ((Str_p->DataCount + 1) == (Str_p->DataLength))
    {
        cli();
        // HMI_snput_matrix(HMI_TYPE_UI16, 1, Str_p->DataLength, Str_p-
>DataList_p);
        Str_p->DataCount = 0;
        sei();
    }
    else
        Str_p->DataCount++;
```

```
}

void SPIDACTrm_step(void *VoidStr_p)
{
	printf("SPIDACTrm_step\n");
	SpiDacTrmStr_t *Str_p = (SpiDacTrmStr_t *)VoidStr_p; /*Typeset Structure pointer*/

	REGFPT(&PORTD, 0x20, 0, 0);
	ASA_SPIM_trm(Str_p->Mode, Str_p->CardId, Str_p->RegAdd, Str_p->Bytes, Str_p->Data_p + Str_p->Counter, 0);
	REGFPT(&PORTD, 0x20, 5, 1);

	//觸發 ADC 轉換
	REGFPT(&ADCSRA, 0x40, 6, 1);

	if ((Str_p->Counter + 1) < arrSize)
		Str_p->Counter++;
	else
		Str_p->Counter = 0;
}
```

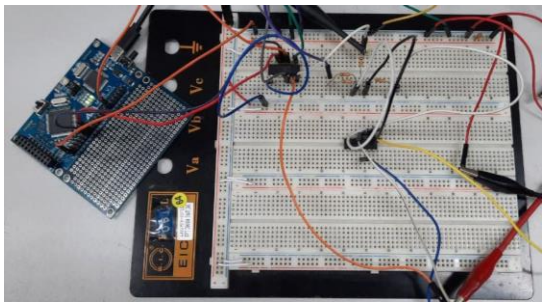
Language : Matlab

```
clear;clc;close;
[port] = remo_open(6);

% 0 2V
% 0 4096
outputVoltage = linspace(0, 1, 21);
data = uint16(outputVoltage * (2^12/2));
remo_snput_matrix(port, data);

while 1
    [data] = remo_get_msg(port);
    disp(data);
end

remo_close(port);
```

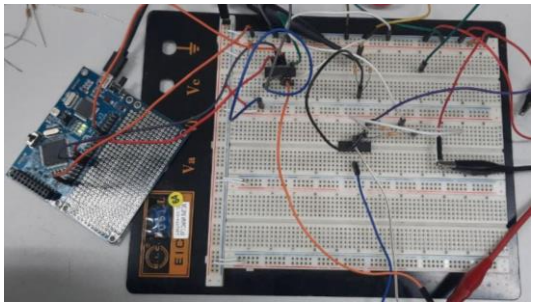
二、實驗數據

1.電路圖

實驗一 反向放大電路及非反向放大電路實作

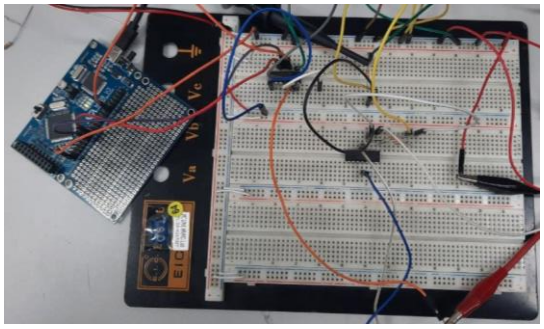


反向放大器

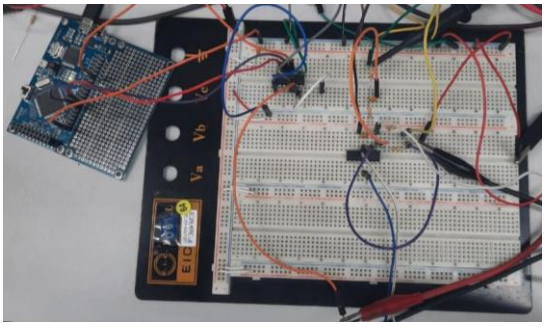


非反向放大器

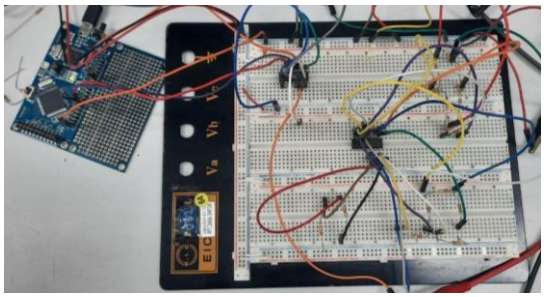
實驗二 低通濾波器實作



一階低通濾波器

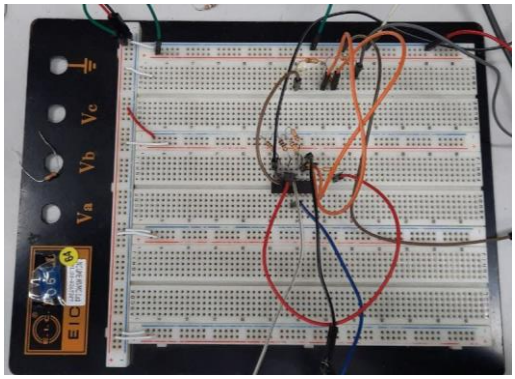


二階低通濾波器

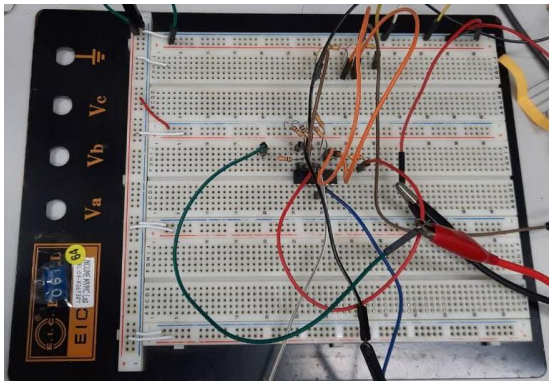


二階差動濾波器

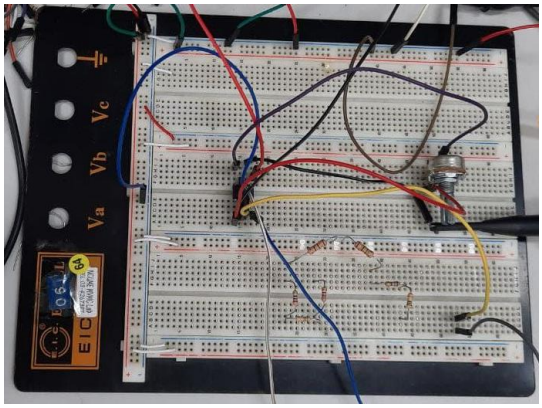
實驗三 開迴路增益實測



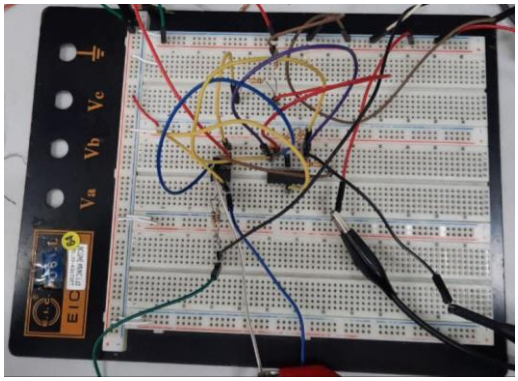
實驗四 增益頻寬積實測



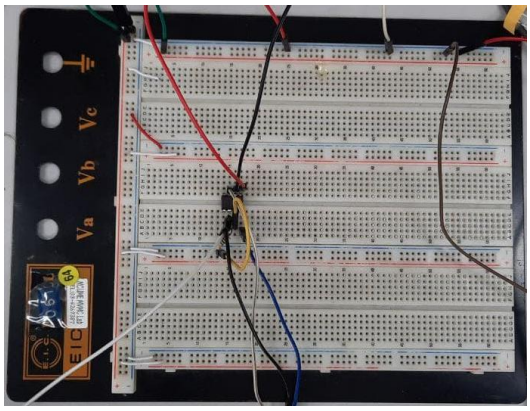
實驗五 儀表放大器電路實測



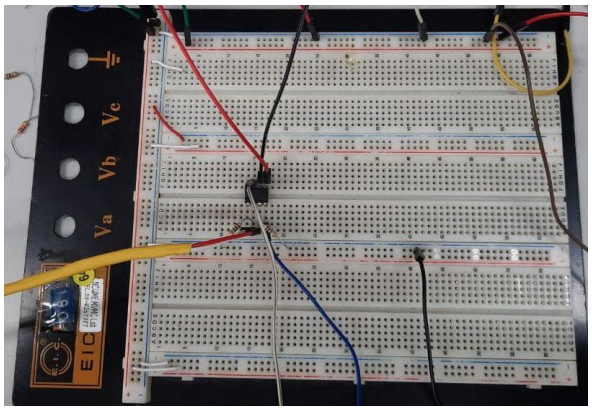
實驗六 CMRR量測實驗



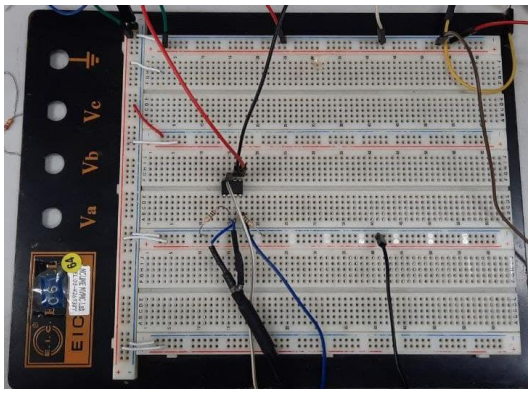
實驗七電磁波雜訊拾取實驗



單線被覆線



平行被覆線(不相交)



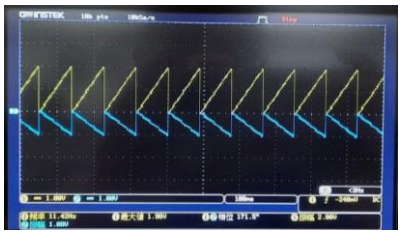
平行被覆線(相交)

2.實驗結果

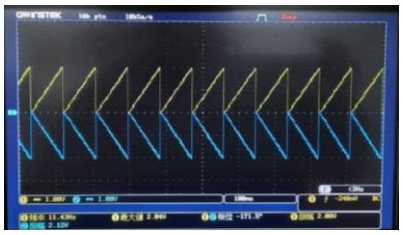
實驗一 反向放大電路及非反向放大電路實作

■ 反向放大電路線性實驗

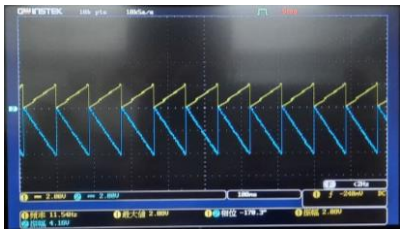
1. DAC提供電壓輸入，示波器輸出



DAC 輸入-5kΩ



DAC 輸入-10kΩ



DAC 輸入-20kΩ

■ 反向放大電路頻率響應實驗

1. 輸入電壓4.8V，不同負載電阻電壓

負載電阻-10Ω

負載電阻-20Ω

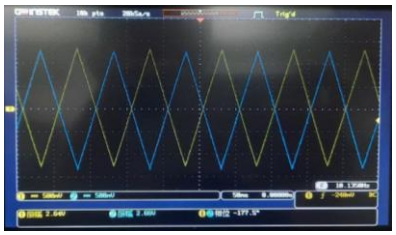
負載電阻-100Ω

負載電阻-1kΩ

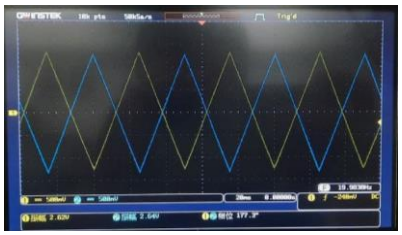
負載電阻-10kΩ

負載電阻-1MΩ

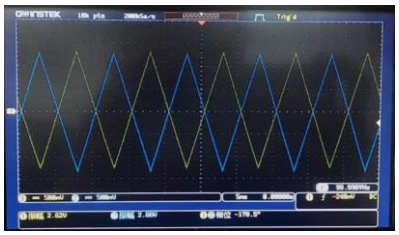
2. 輸入產波器之弦波，示波器輸出



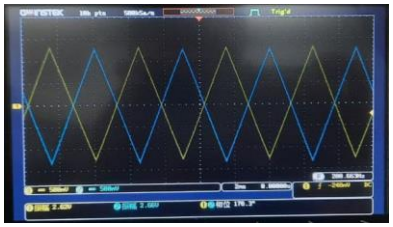
頻率-10Hz



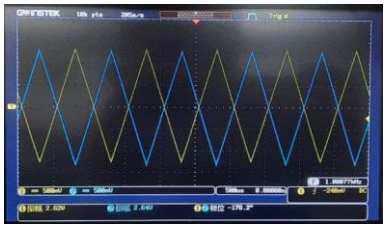
頻率-20Hz



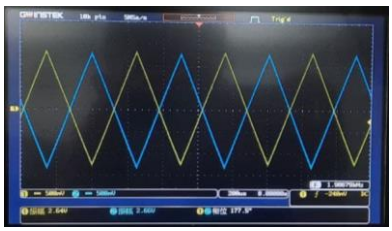
頻率-100Hz



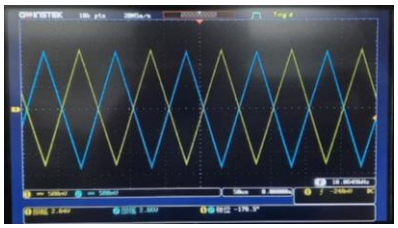
頻率-200Hz



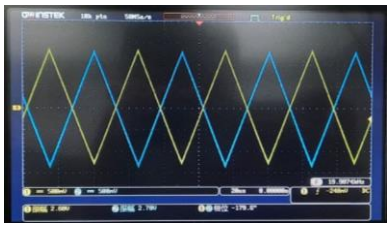
頻率-1kHz



頻率-2kHz



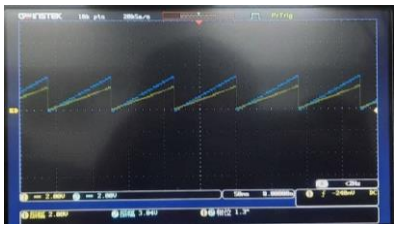
頻率-10kHz



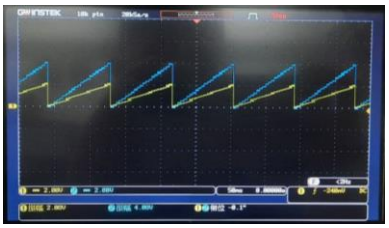
頻率-20kHz

■ 非反向放大器電路線性實驗

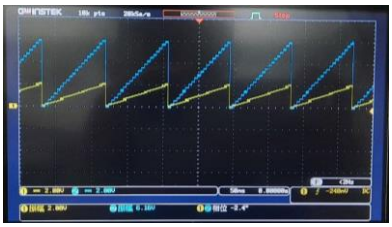
1. DAC提供電壓輸入、示波器輸出



DAC 輸入-5kΩ



DAC 輸入-10kΩ



DAC 輸入-20kΩ

■ 非反向放大電路頻率響應實驗

1. 輸入電壓4.8V，不同負載電阻電壓

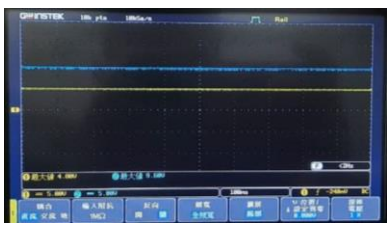
負載電阻-10Ω

負載電阻-20Ω

負載電阻-100Ω

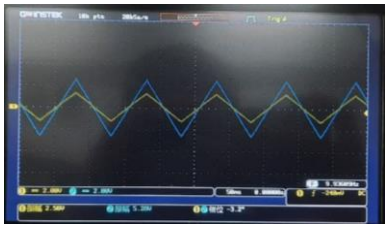
負載電阻-1kΩ

負載電阻-10kΩ

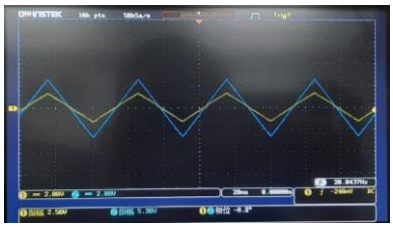


負載電阻-1MΩ

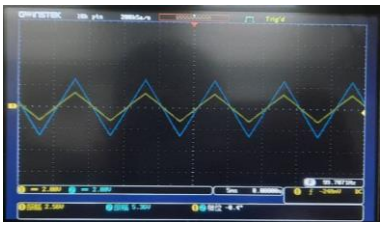
2. 輸入產波器之弦波，示波器輸出



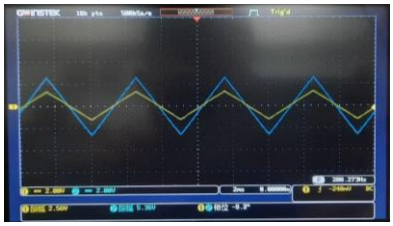
頻率-10Hz



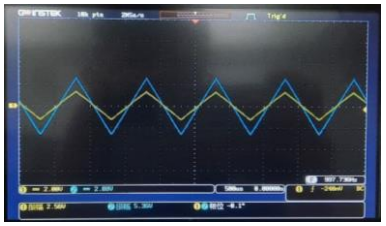
頻率-20Hz



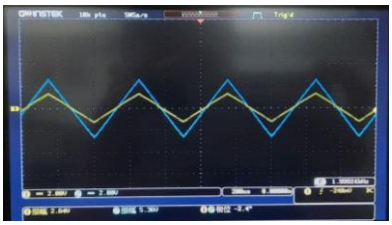
頻率-100Hz



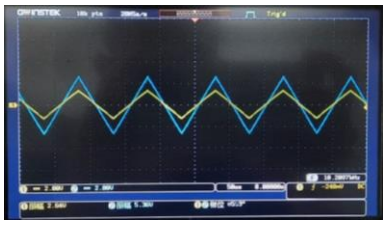
頻率-200Hz



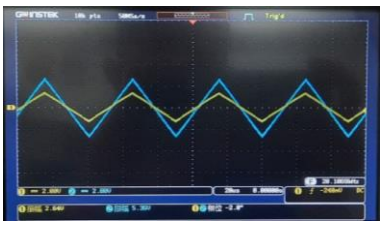
頻率-1kHz



頻率-2kHz



頻率-10kHz



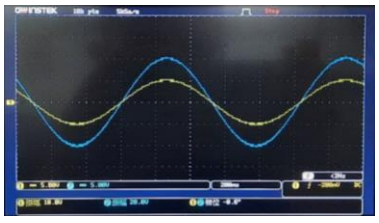
頻率-20kHz

實驗二 低通濾波器實作

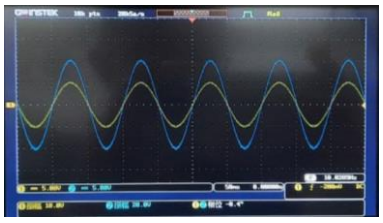
一階低通濾波器設計實作

■ 正弦波反應

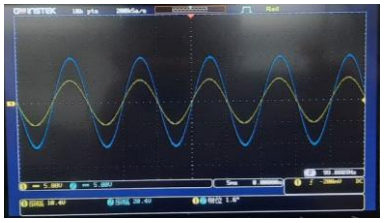
1. 頻率響應



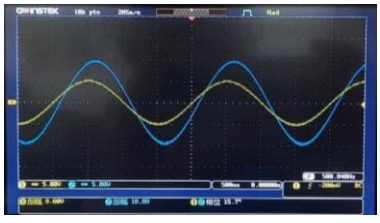
頻率-1Hz



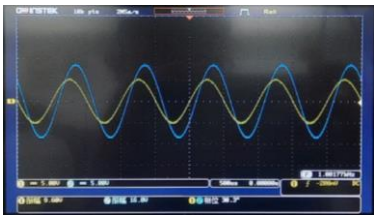
頻率-10Hz



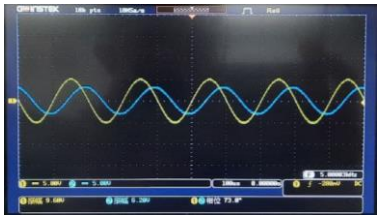
頻率-100Hz



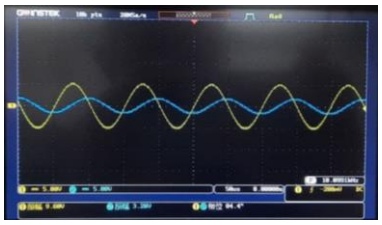
頻率-500Hz



頻率-1kHz

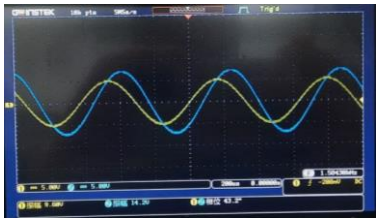


頻率-5kHz

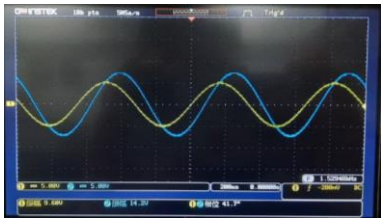


頻率-10kHz

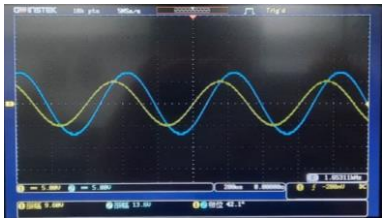
2. 截止頻率



截止頻率

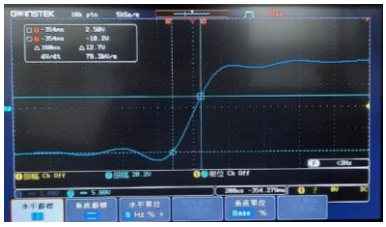


截止頻率前

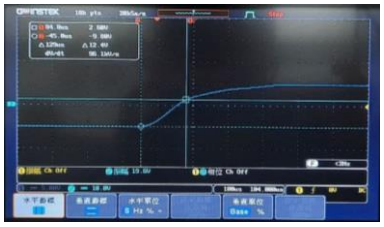


截止頻率後

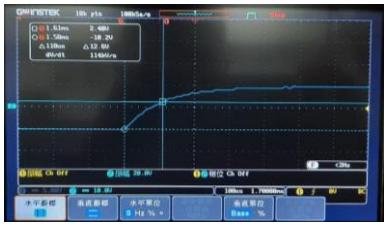
■ 方波反應



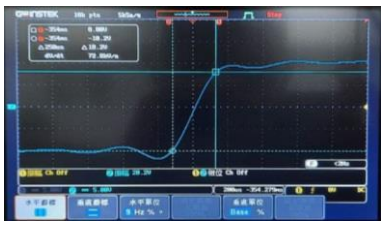
頻率 1Hz-時間常數



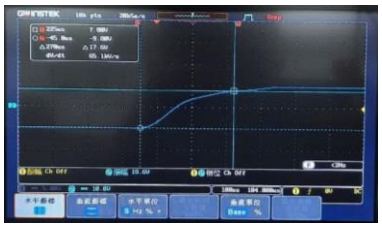
頻率 10Hz-時間常數



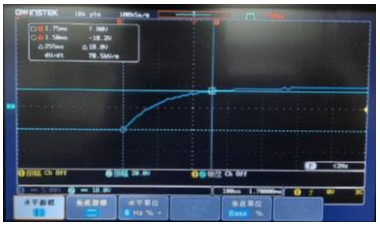
頻率 100Hz-時間常數



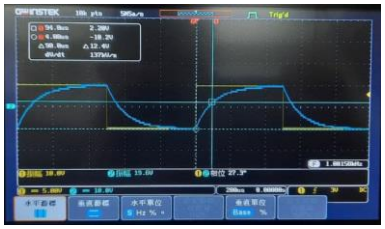
頻率 1Hz-穩定時間



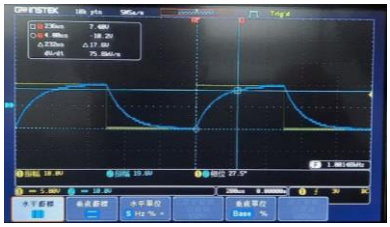
頻率 10Hz-穩定時間



頻率 100Hz-穩定時間



頻率 1kHz-時間常數

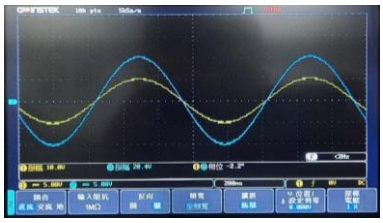


頻率 1kHz-穩定時間

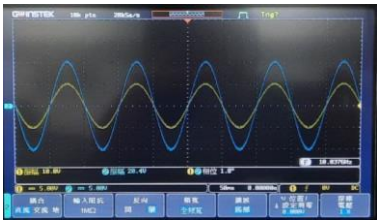
■ 自行設計

1. 正弦波反應

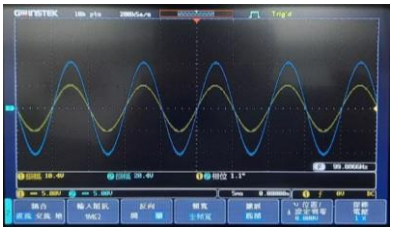
A. 頻率響應



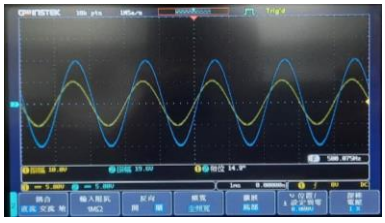
頻率-1Hz



頻率-10Hz

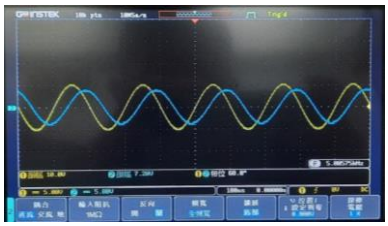


頻率-100Hz



頻率-500Hz

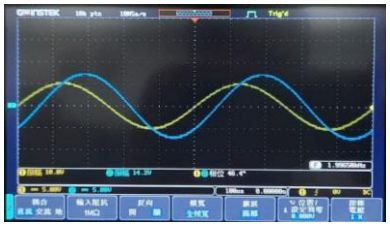
頻率-1kHz



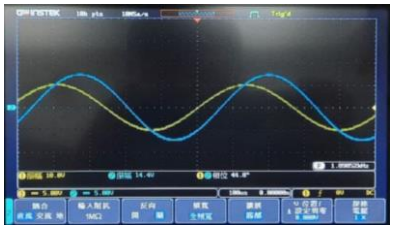
頻率-5kHz

頻率-10kHz

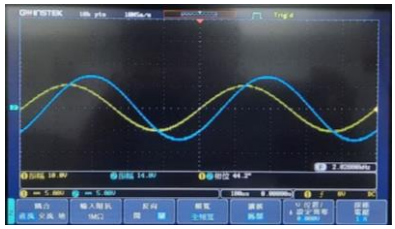
B. 截止頻率



截止頻率

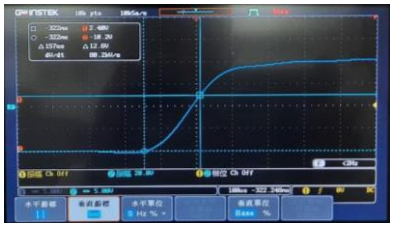


截止頻率前

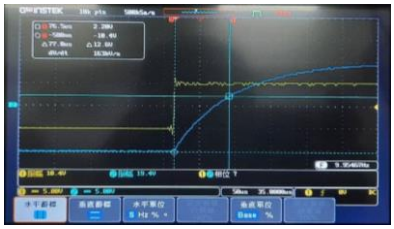


截止頻率後

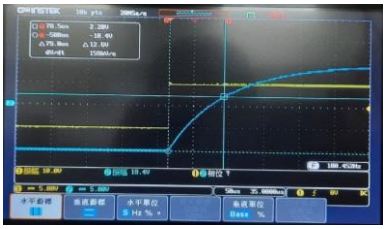
2. 方波反應



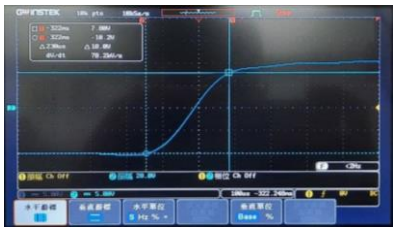
頻率 1Hz-時間常數



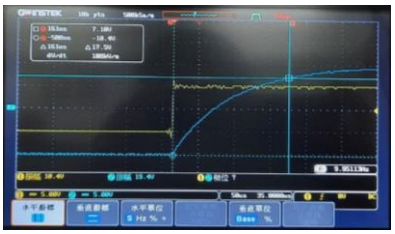
頻率 10Hz-時間常數



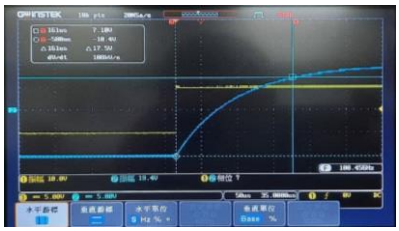
頻率 100Hz-時間常數



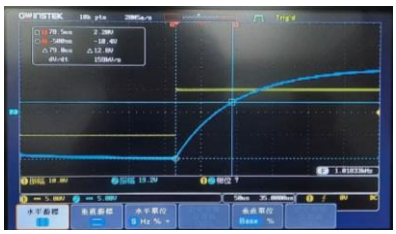
頻率 1Hz-穩定時間



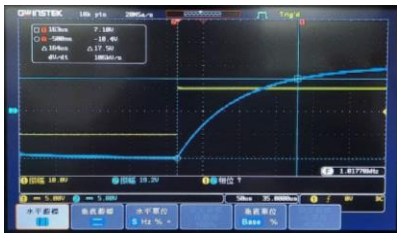
頻率 10Hz-穩定時間



頻率 100Hz-穩定時間



頻率 1kHz-時間常數

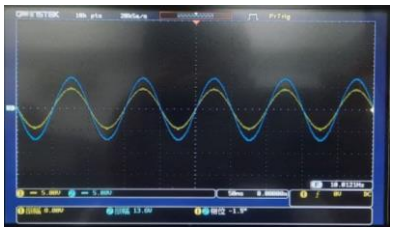


頻率 1kHz-穩定時間

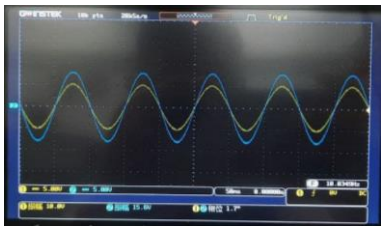
二階低通濾波器設計實作

■ 正弦波反應

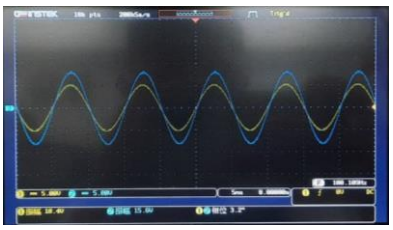
1. 頻率響應



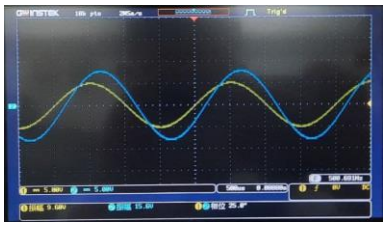
頻率-1Hz



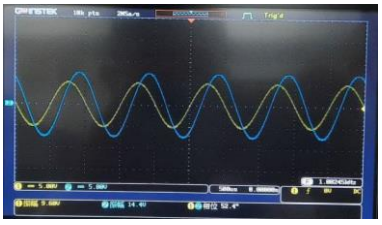
頻率-10Hz



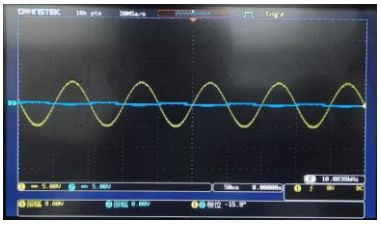
頻率-100Hz



頻率-500Hz

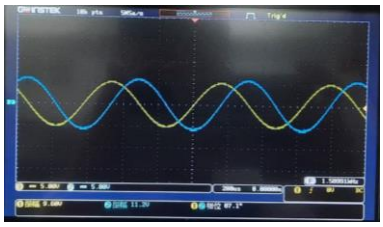


頻率-1kHz

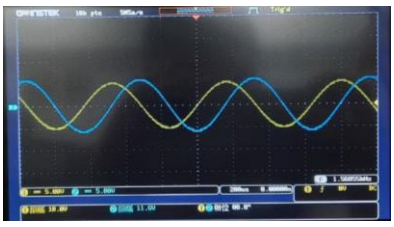


頻率-10kHz

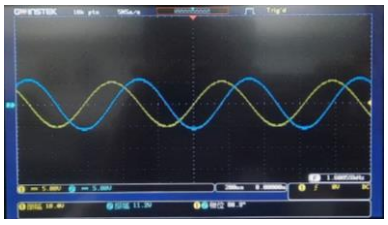
2. 截止頻率



截止頻率

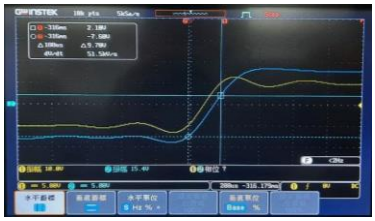


截止頻率前

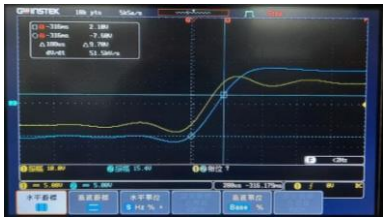


截止頻率後

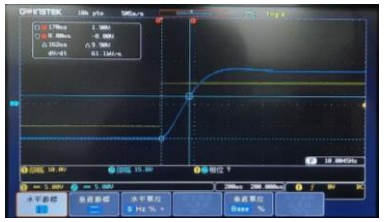
■ 方波反應



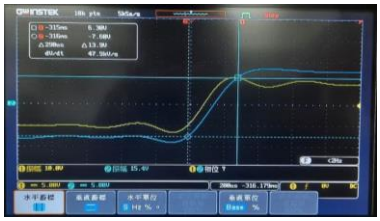
頻率 1Hz-時間常數



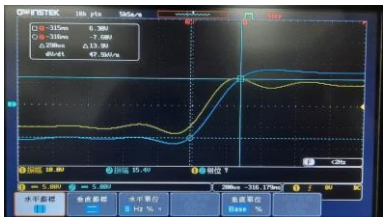
頻率 10Hz-時間常數



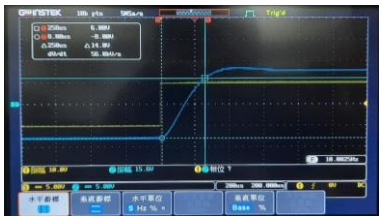
頻率 100Hz-時間常數



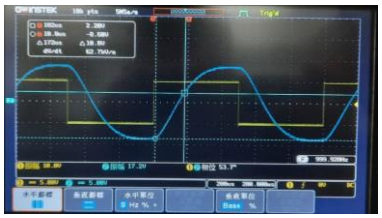
頻率 1Hz-穩定時間



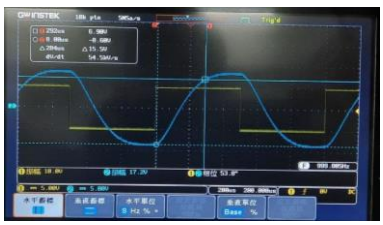
頻率 10Hz-穩定時間



頻率 100Hz-穩定時間



頻率 1kHz-時間常數

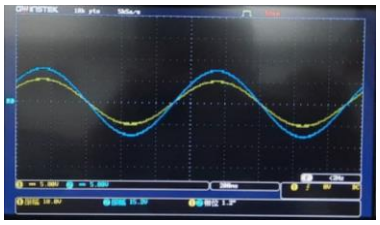


頻率 1kHz-穩定時間

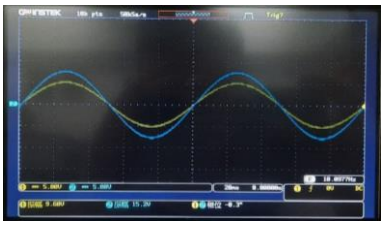
■ 自行設計

1. 正弦波反應

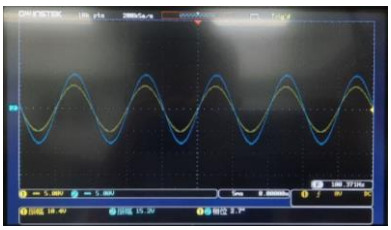
A. 頻率響應



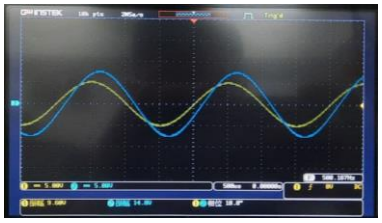
頻率-1Hz



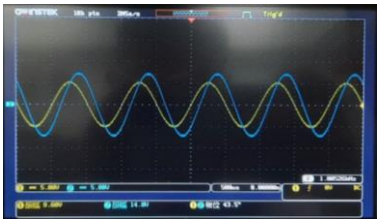
頻率-10Hz



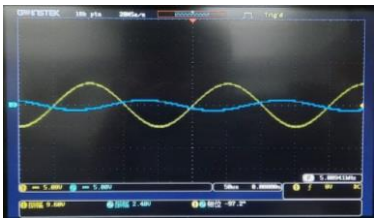
頻率-100Hz



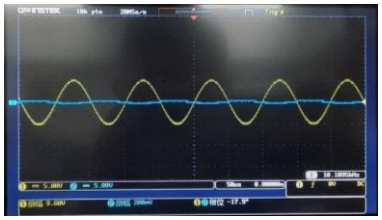
頻率-500Hz



頻率-1kHz

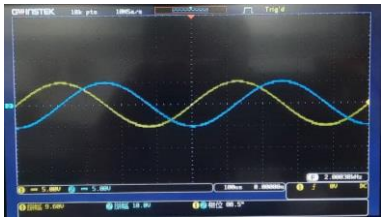


頻率-5kHz



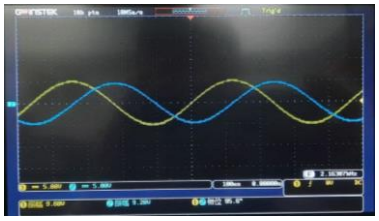
頻率-10kHz

B. 截止頻率



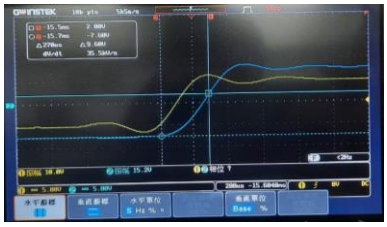
截止頻率

截止頻率前

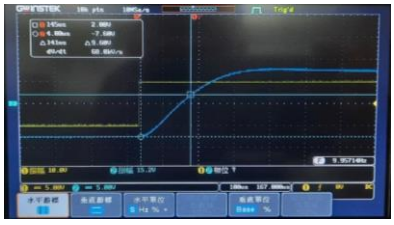


截止頻率後

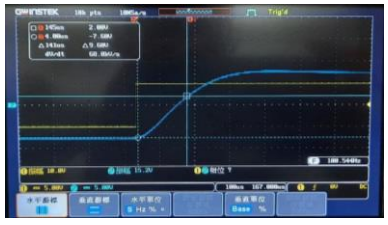
2. 方波反應



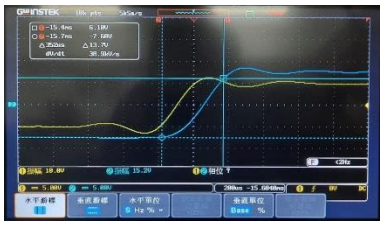
頻率 1Hz-時間常數



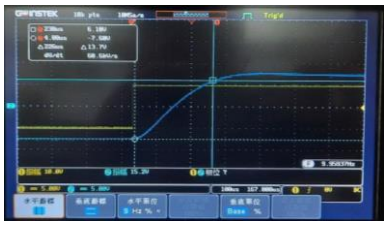
頻率 10Hz-時間常數



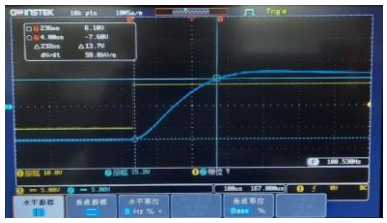
頻率 100Hz-時間常數



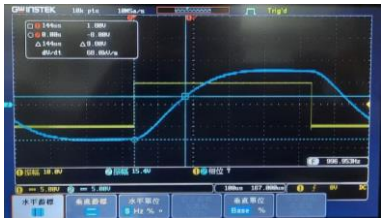
頻率 1Hz-穩定時間



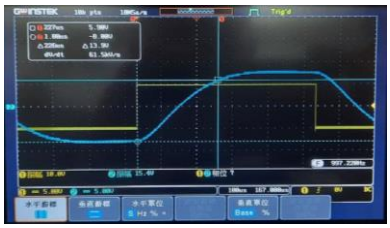
頻率 10Hz-穩定時間



頻率 100Hz-穩定時間



頻率 1kHz-時間常數

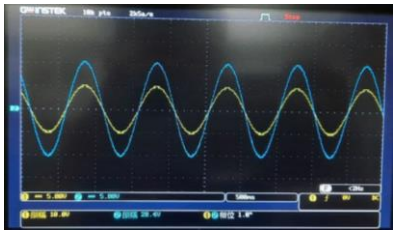


頻率 1kHz-穩定時間

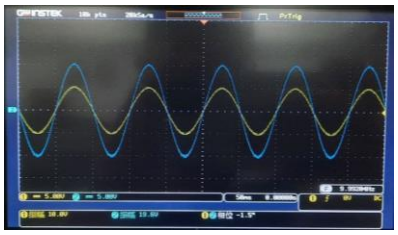
二階差動濾波器設計實作

■ 正弦波反應

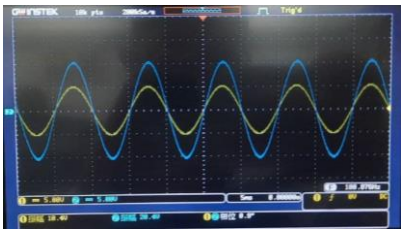
1. 頻率響應



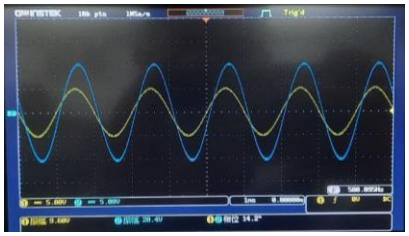
頻率-1Hz



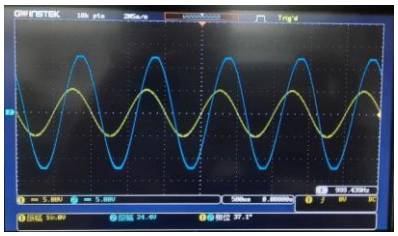
頻率-10Hz



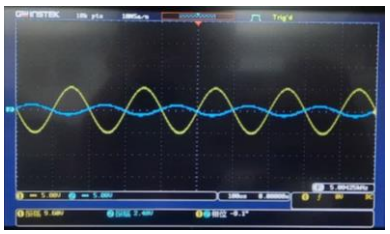
頻率-100Hz



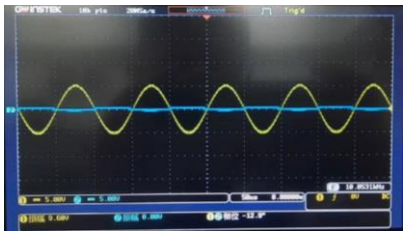
頻率-500Hz



頻率-1KHz



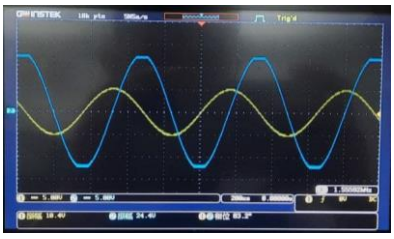
頻率-5KHz



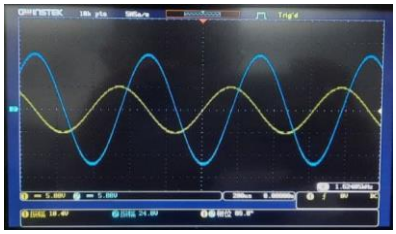
頻率-10kHz

2. 截止頻率

截止頻率

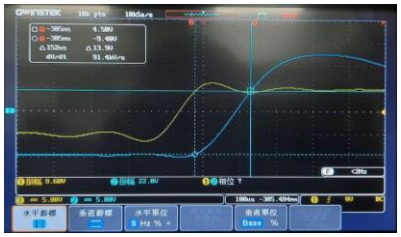


截止頻率前

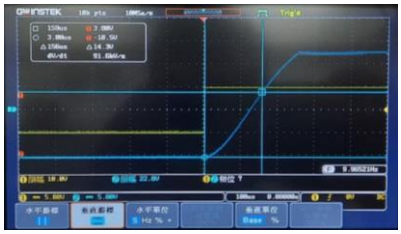


截止頻率後

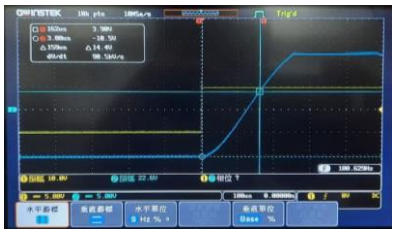
■ 方波反應



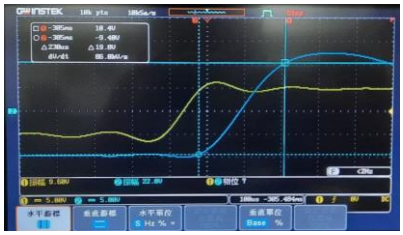
頻率 1Hz-時間常數



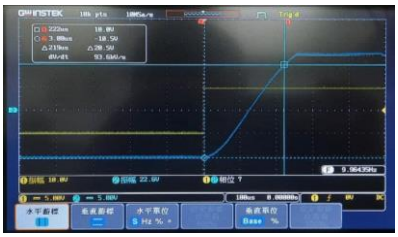
頻率 10Hz-時間常數



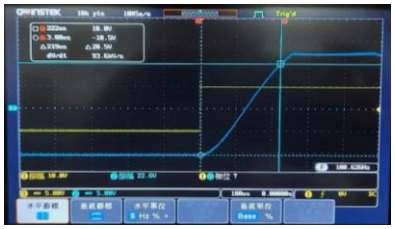
頻率 100Hz-時間常數



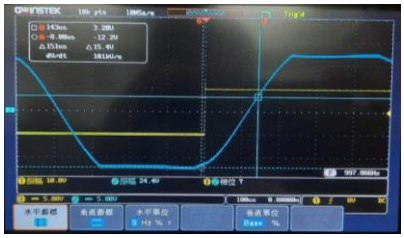
頻率 1Hz-穩定時間



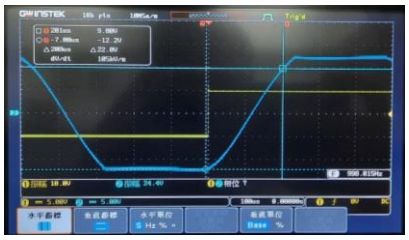
頻率 10Hz-穩定時間



頻率 100Hz-穩定時間



頻率 1kHz-時間常數

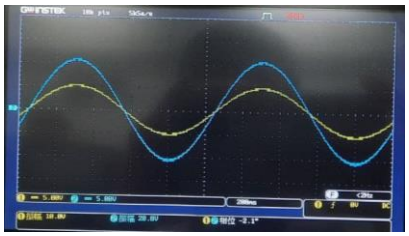


頻率 1kHz-穩定時間

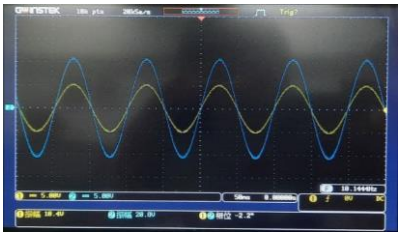
■ 自行設計

1. 正弦波反應

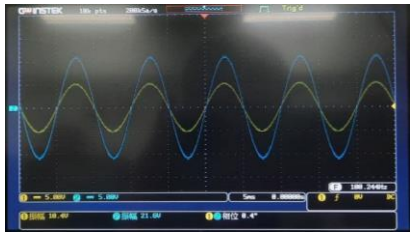
A. 頻率響應



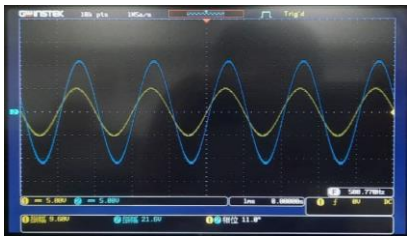
頻率-1Hz



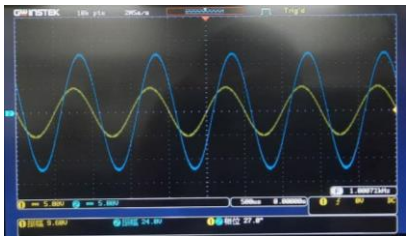
頻率-10Hz



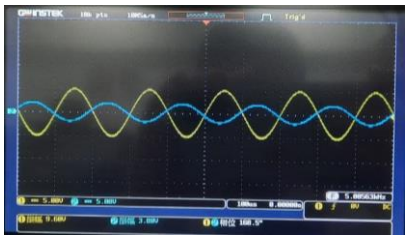
頻率-100Hz



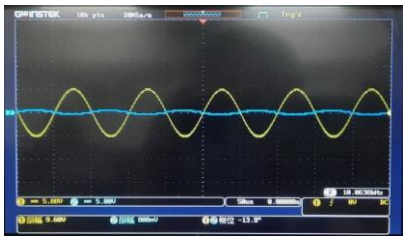
頻率-500Hz



頻率-1kHz

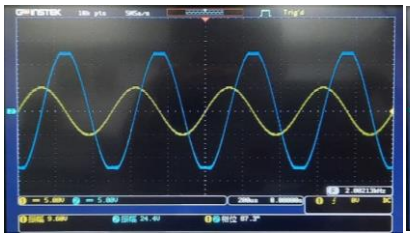


頻率 5kHz

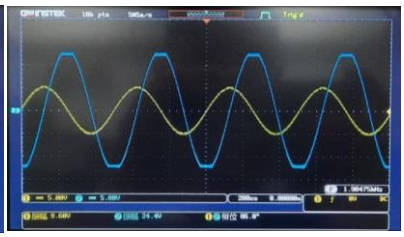


頻率-10kHz

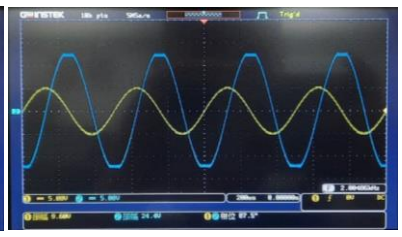
B. 截止頻率



截止頻率

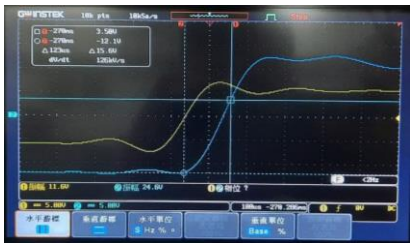


截止頻率前

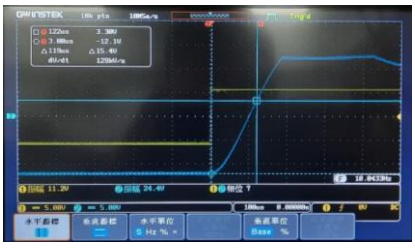


截止頻率後

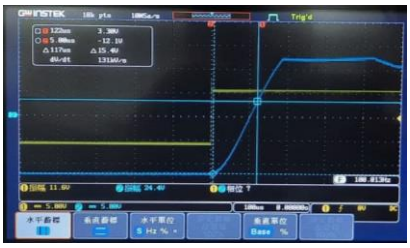
2. 方波反應



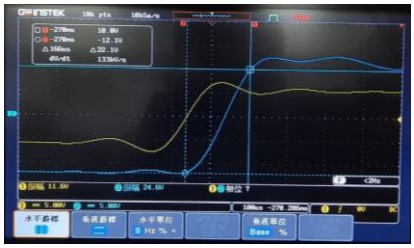
頻率 1Hz-時間常數



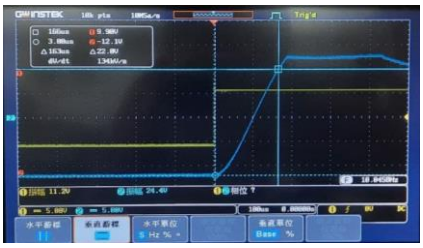
頻率 10Hz-時間常數



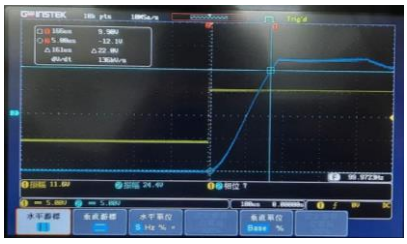
頻率 100Hz-時間常數



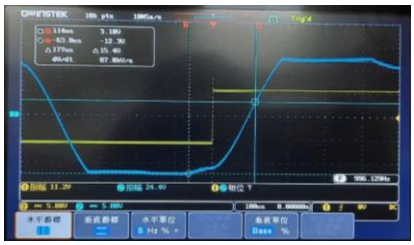
頻率 1Hz-穩定時間



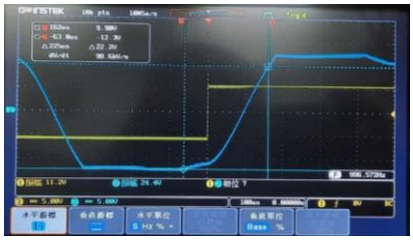
頻率 10Hz-穩定時間



頻率 100Hz-穩定時間



頻率 1kHz-時間常數



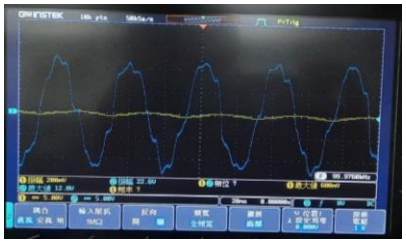
頻率 1kHz-穩定時間

開迴路增益實測

E_i=+10V

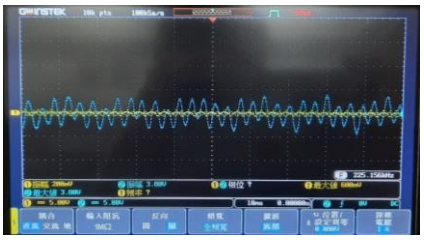
E_i=-10V

增益頻寬積實測

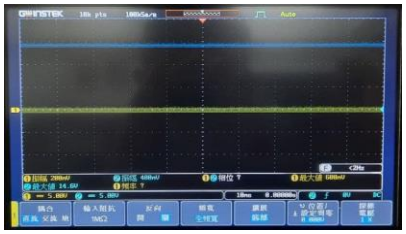


產波器 100kHz

產波器 200kHz



產波器 500kHz

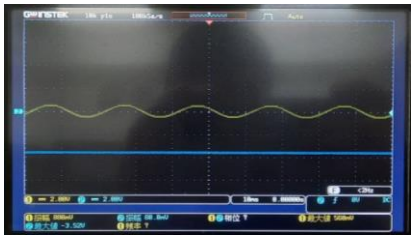


產波器 1MHz

儀表放大器電路實測

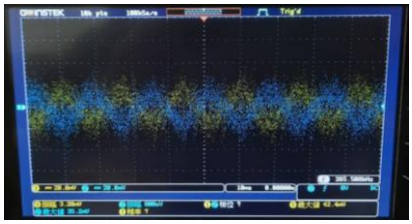
	輸入電壓	0.05[V]	0.1[V]	1[V]	10[V]
增益 Gain					
1		-26.0[mV]	-0.04[V]	-0.96[V]	-10[V]
10		-344[mV]	-640[mV]	-9.8[V]	-14.4[V]
100		-2.00[V]	-6.60[V]	-14.60[V]	-14.40[V]
1000		-14.60[V]	-14.40[V]	-14.40[V]	-14.40[V]

CMRR 量測實驗

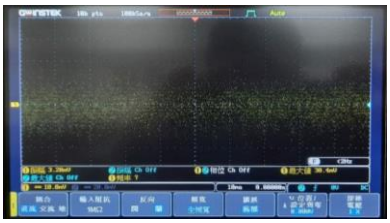


$E_0(0)$

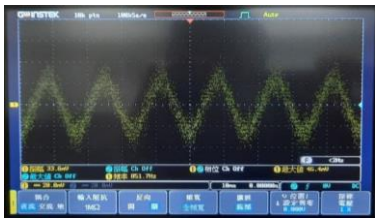
電磁波雜訊拾取實驗



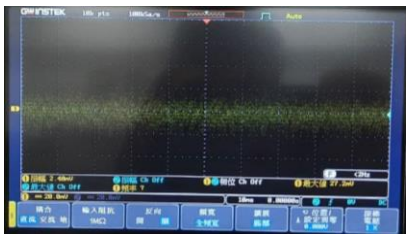
單線被覆線



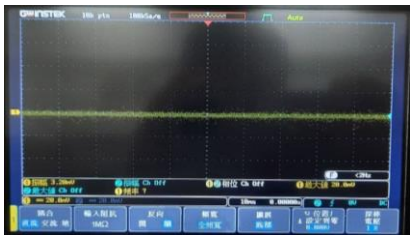
平行被覆線(不相交)



平行被覆線(相交)



RJ45(不接地)



RJ45(接地)

三、實驗問題