介面實驗

實驗五

ASA DAC 介面卡開發及使用

班級：機械 2A

學號：108303013

姓名：黃鉦淳

日期：110/8/4

介面實驗工作日誌

實驗五　　　　　　　　　　　　　　110 年 8 月 5 日

組別		姓名	黃鉦淳	學號	108303013
實驗起始時間		110/8/3		費時	2 天
實驗結束時間		110/8/4			

所遭遇問題	DAC 的實驗被包裝的好困難
解決方法	狂問學長
完成項目‧及心得	DAC 比 ADC 好懂
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、流程圖
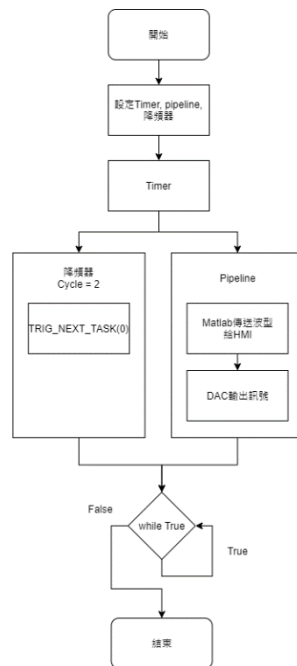
程式碼

Language : C
#include "c4mlib.h"

void DACPrePro_step(void *VoidStr_p);
void SPIDACTrm_step(void *VoidStr_p);
void trigger_pipeline();

void SPI_init();
void timer3_init();

uint8_t arrSize;

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

```c
        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 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 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*/
\
```

```c
        .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 */
\
        .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                            /*Triggered Counter */
\
    }

int main()
{
    C4M_DEVICE_set();

    HMI_snget_matrix(HMI_TYPE_UI8, 1, 1, &arrSize);

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

    //設定Timer3中斷
    TIMHWINT_LAY(TIMINT_Str, 3, 2);

    timer3_init();

    //設定SPI
    SPI_init();

    //設定降頻器配置
    int period[2] = {10800, 10800};
```

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

    uint8_t freq_TaskID;
    freq_TaskID = FreqRedu_reg(&FreqRedu_Str, &trigger_pipeline, NULL,
1, 0);

    FreqRedu_en(&FreqRedu_Str, freq_TaskID, ENABLE);

    DACPrePro_LAY(DAC_PostPro_Str, arrSize, 0);
    SPIDACTrm_LAY(SPI_DAC_Str, 0);

    SPI_DAC_Str.Data_p = DAC_PostPro_Str.DataList_p;

    //設定Pipeline
    PIPELINE_LAY(2, 4, 10);

    //單/雙通道資料後處理
    uint8_t pipeline_TaskID[2];
    pipeline_TaskID[0] = Pipeline_reg(&SysPipeline_str, &DACPrePro_step,
&DAC_PostPro_Str, NULL);
    pipeline_TaskID[1] = Pipeline_reg(&SysPipeline_str, &SPIDACTrm_step,
&SPI_DAC_Str, NULL);

    uint8_t TaskID[2];

    //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pipeline登入進Timer中斷
    TaskID[1] = HWInt_reg(&TIMINT_Str, &Pipeline_step, &SysPipeline_str);
    HWInt_en(&TIMINT_Str, TaskID[1], ENABLE);

    sei();

    while (1)
    {
```

```
            ;
        }
        return 0;
    }

void trigger_pipeline()
{
    TRIG_NEXT_TASK(0);
}

void SPI_init()
{
    REGFPT(&DDRF, 0x01, 0, 0x01); //B0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, 5);

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

```c
void DACPrePro_step(void *VoidStr_p)
{
    DACPreProStr_t *Str_p = (DACPreProStr_t *)VoidStr_p;
    HMI_snget_matrix(HMI_TYPE_UI16, 1, Str_p->DataLength, Str_p->DataList_p);
    TRIG_NEXT_TASK(1);
}

void SPIDACTrm_step(void *VoidStr_p)
{
    SpiDacTrmStr_t *Str_p = (SpiDacTrmStr_t *)VoidStr_p; /*Typeset Structure pointer*/

    for (uint8_t i = 0; i < arrSize; i++)
    {
        REGFPT(&PORTF, 0x01, 0, 0);
        ASA_SPIM_trm(Str_p->Mode, Str_p->CardId, Str_p->RegAdd, Str_p->Bytes, Str_p->Data_p + i, 0);
        REGFPT(&PORTF, 0x01, 0, 1);
        _delay_ms(10);
    }
}
```

Language : Matlab

```matlab
clear;clc;close;
[port] = remo_open(8);

var = input('enter arrSize = ');
remo_snput_matrix(port,uint8(var));

t = linspace(0,2*pi,uint8(var));

y = sin(t);
y = uint16((y + 1) / 2 * (2^12 -1));

while 1
    remo_snput_matrix(port,y);
end

remo_close(port);
```
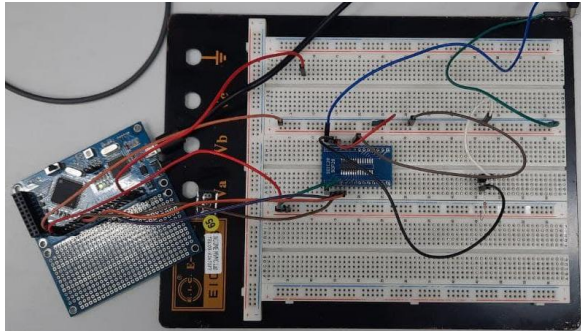
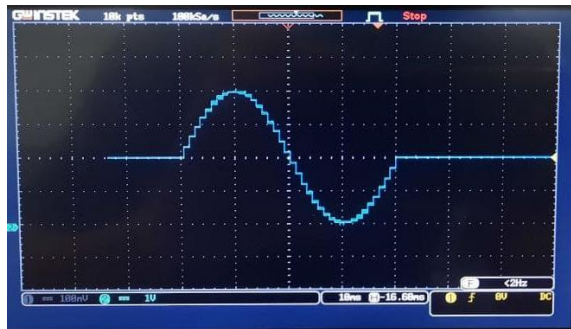# 三、實驗數據

1.電路圖

2.實驗結果

V$_{pp}$ 4V 週期0~2π

四、實驗問題

1. 請問您在這個實驗中您學到了什麼?
ADC與DAC的轉換其實是反向的,以及可以藉由程式進行DAC轉換。轉換完之後之解析度會隨之下降,那是為了讓程式執行上較快速。

2. 請問 D/A 輸出是否能夠有斜率的上昇或則下降,為什麼?
否,因為在最小誤差範圍(LSB)內,皆為同一數值。