



Computer Engineering Department
Microcontroller Lab (10636496)
Report Grading Sheet

| | | | |
|---|----------------------|--------------|---------------|
| Instructor Name: Hikmat Darawsheh | Experiment: 8 | | |
| Academic Year: 2024 | Performed on: | | |
| Semester: | Submitted on: | | |
| Student Names: | | | |
| 1- Wala' Essam Ashqar | 2- Sadeen Hawash | | |
| 3- | 4- | | |
| 5- | 6- | | |
| Evaluation Criterion | CLO | Grade | Points |
| Abstract and Aims Aims and idea of the experiment are clearly stated in simple words | | 10 | |
| Introduction, Apparatus and Procedures Introduction is complete and well-written, all grammar/spelling correct, Appropriate background information related to the principles of the experiment is provided. The list of apparatus and procedures are also provided | | 15 | |
| Experimental Results, Calculations and Discussion Results analyzed correctly. Experimental findings adequately and specifically summarized, in graphical, tabular, and/or written form. Comparison of theoretical predictions to experimental results, including discussion of accuracy and error analysis as needed. | | 50 | |
| Conclusions Conclusions summarize the major findings from the experimental results with adequate specificity. Highlighting the most important results | | 15 | |
| Appearance Title page is complete, page numbers applied, content is well organized, correct spelling, fonts are consistent, good visual appeal. You have also to use reference for the information you provide | | 10 | |
| Total | | 100 | |



Contents

| | |
|---|-------------------|
| Abstract..... | 2 |
| Introduction..... | 2 |
| Materials..... | 2 |
| Methods..... | 2 |
| Experimental Results..... | 3 |
| Discussion..... | 5 |
| Conclusion..... | 5 |

Abstract

In this experiment we learnt how to generate a proportional output using the output compare resource on the PIC32MX processor so that we can implement Digital-to-Analog Conversion (DAC) with pulse width modulation (PWM). The proportional output will be used to control the speed of a DC motor.

Introduction

By turning on the CN (change notice) for a RG6 and RG7 pins (button 1,2), we can use interrupts in PIC32 code so that whenever this pin changes, it will trigger an interrupt, which will halt the code's execution and switch to the interrupt service routine (ISR), which we should implement to handle the interrupts to change the speed of the DC motor.



Materials

- Material: ChipKITTM Pro MX7 processor board with USB cable.
- Microchip MPLAB R X IDE.
- MPLAB R XC32++ Compiler.
- MPLAB Harmony Framework.
- Tera Term.
- Digilent DC Motor
- Digilent PmodDHB1

Methods

To conduct this experiment, we made a new project and adjusted the pin settings accordingly (RG6, RG7) to enable CN. After correctly setting the clock, we initialized the timer and the Output Compare (OC) and then we changed the values of the speed of the DC motor in the ISR code depending on the values of the buttons.

Experimental Results

We initialized the timer and OC in the app.c as follows:

```
case APP_STATE_INIT:

    {        appInitialized = true;

        if (appInitialized)

        { DRV_TMR0_Start();

            DRV_OC0_Start();

            appData.state = APP_STATE_SERVICE_TASKS;

        }        break;

    }
```



In the system_interrupt.c we configure the code when an interrupt happens on the buttons and depending on the values of the buttons, we change the speed of the DC motor accordingly as follows:

And we set the values of the OC to match the table below

| BTN1 | BTN2 | PWM |
|------|------|-----|
| OFF | OFF | 25% |
| ON | OFF | 50% |
| OFF | ON | 75% |
| ON | ON | 95% |

```
extern APP_DATA appData;

int flag=0;

int i;

int speed=80;

void __ISR(_CHANGE_NOTICE_VECTOR, ip11AUTO)_IntHandlerChangeNotification(void)
{
    for(i=0;i<500000;i++);

    /* TODO: Add code to process interrupt here */

    if(PORTGbits.RG6==0 && PORTGbits.RG7==0)
    {
        speed=50;

        LATG=0x8FFF;
    }

    else if(PORTGbits.RG6==1 && PORTGbits.RG7==0 )
    {
        speed=100;

        LATG=0x1FFF;
    }
}
```



```
else if(PORTGbits.RG6==0 && PORTGbits.RG7==1)
{
    speed=150;
    LATG=0x8FFF;
}
else if(PORTGbits.RG6==1 && PORTGbits.RG7==1)
{
    speed=200;
    LATG=0x8FFF;
}
DRV_OC0_PulseWidthSet(speed);
//appData.state = APP_STATE_SERVICE_TASKS;
PLIB_INT_SourceFlagClear(INT_ID_0, INT_SOURCE_CHANGE_NOTICE);
}
void __ISR(_TIMER_2_VECTOR, IPL1_AUTO) IntHandlerDrvTmrInstance0(void)
{
    PLIB_INT_SourceFlagClear(INT_ID_0, INT_SOURCE_TIMER_2);
}
void __ISR(_OUTPUT_COMPARE_2_VECTOR, IPL1_AUTO) _IntHandlerDrvOCInstance0(void)
{
    PLIB_INT_SourceFlagClear(INT_ID_0, INT_SOURCE_OUTPUT_COMPARE_2);}
```



Discussion

We used interrupts handle the changes that happens to the buttons to control the speed of the DC motor.

Conclusion

This lab taught us about interrupts, how to set them up, and how to use them to control other component such as OC and change the speed of the DC motor and how we could control the speed of a motor by changing the duty cycle for the OC.