An-Najah National University
Faculty of Engineering and IT

جامعة النجاح الوطنية
كلية الهندسة وتكنولوجيا

Computer Engineering Department
Microcontroller Lab (10636496)

Report Grading Sheet

Instructor Name: Hikmat Darawsheh	**Experiment: 9**
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	

An-Najah National University
Faculty of Engineering and IT

جامعة النجاح الوطنية
كلية الهندسة وتكنولوجيا

Contents

Abstract

In this experiment we learnt how to use stepper motors and how they work and how to control them to move in a certain direction (clock wise or counter clock wise) or do full step or half step (180 degree or 360 degree).

Introduction

The stepper motor in this experiment is connected to the JA PORTB (RB7-RB10). It is a two-phase stepper, with a 1.8◦ stepping angle for the first phase (full step) and a 0.9◦ stepping angle for the second phase (half step). The activator can regulate the direction and delay of the object to change its speed.

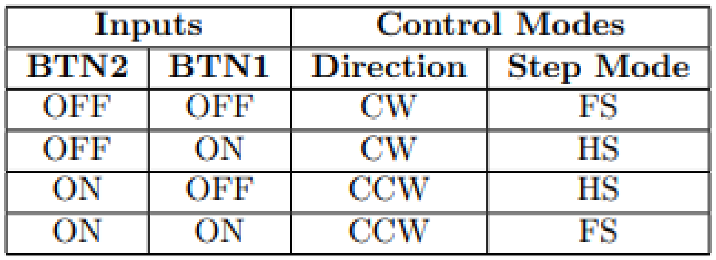
Materials

- Material: ChipKITTM Pro MX7 processor board with USB cable.
- Microchip MPLAB R X IDE.
- MPLAB R XC32++ Compiler.
- MPLAB Harmony Framework.
- Tera Term.
- PmodSTEPTM
- Stepper Motor (5V-12V, 25Ω, unipolar or bi-polar)

Methods

To conduct this experiment, we made a new project and adjusted the pin settings accordingly (RG6, RG7) to enable CN. we changed the Direction and the Step mode depending on the value of the two buttons as follows

.



Inputs		Control Modes	
BTN2	**BTN1**	**Direction**	**Step Mode**
OFF	OFF	CW	FS
OFF	ON	CW	HS
ON	OFF	CCW	HS
ON	ON	CCW	FS

Table 12: Task: Stepper motor controls.

Experimental Results

We add these states on app.h

```
typedef enum

{        /* Application's state machine's initial state. */

APP_STATE_INIT=0,

APP_STATE_SERVICE_TASKS,

         fullStep,

         halfStep,

         CWFS,

         CWHS,

         CCWHS,

         CCWFS,

} APP_STATES;
```

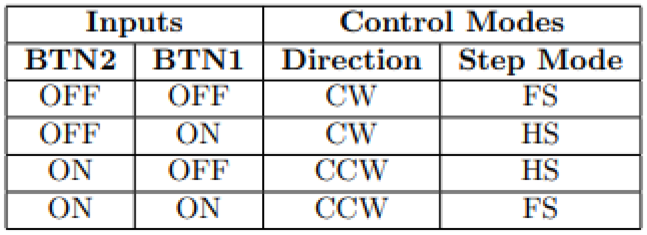
We initialized both the arrays for 1.8 and 0.9 and their directions (CW and CCW):

```
APP_DATA appData;

int fullarr[4] ={0x08,0x01,0x04,0x02};

int halfarr[8] ={0x02,0x06,0x04,0x05,0x01,0x09,0x08,0x0A};

int i=0;int j=0;int k=0;
```

In the system_interrupt.c we configure the code when an interrupt happens on the buttons and depending on the values of the buttons, The state of the program (the direction and the step mode of the stepper motor) as follows:

```
void __ISR(_CHANGE_NOTICE_VECTOR, ipl1AUTO) _IntHandlerChangeNotification(void)
{
   /* TODO: Add code to process interrupt here */
   int btn1 = PORTGbits.RG6;
   int btn2 = PORTGbits.RG7;
  if(btn1 == 0 && btn2 == 0 ){
      appData.state = CWFS;
   }
   else if(btn1 == 1 && btn2 == 0 ){
      appData.state = CWHS;
   }
   else if(btn1 == 0 && btn2 == 1 ){
      appData.state = CCWHS;
   }
   else if(btn1 == 1 && btn2 == 1 ){
      appData.state = CCWFS;
   }
   PLIB_INT_SourceFlagClear(INT_ID_0, INT_SOURCE_CHANGE_NOTICE);
}
```

to match the table as follows:



Inputs		Control Modes	
BTN2	BTN1	Direction	Step Mode
OFF	OFF	CW	FS
OFF	ON	CW	HS
ON	OFF	CCW	HS
ON	ON	CCW	FS

Table 12: Task: Stepper motor controls.

and for the code in app.c as follows:

```
case fullStep:
    {       i=0;j=0;k=0;
      for(j=0;j<50;j++){
        for(k=0;k<4;k++){
          PORTB = fullarr[k] << 7;
          for(i=0;i<500000;i++){}
        }       }
      appData.state = APP_STATE_SERVICE_TASKS;
      break;     }
    case halfStep:
    {
      i=0;j=0;k=0;
      for(j=0;j<25;j++){
        for(k=0;k<8;k++){
          PORTB = halfarr[k] << 7;
          for(i=0;i<500000;i++){}
        }       }
      appData.state = APP_STATE_SERVICE_TASKS;
      break;
```

```
}      case CWFS:
{          i=0;j=0;k=0;
   for(j=0;j<50;j++){
      for(k=0;k<4;k++){
         PORTB = fullarr[k] << 7;
         for(i=0;i<500000;i++){}
      }          }
   appData.state = APP_STATE_SERVICE_TASKS;
   break;
}
case CWHS:
{          i=0;j=0;k=0;
   for(j=0;j<25;j++){
      for(k=0;k<4;k++){
         PORTB = fullarr[k] << 7;
         for(i=0;i<500000;i++){}
      }          }
   appData.state = APP_STATE_SERVICE_TASKS;
   break;
}
case CCWHS:
{          i=0;j=0;k=0;
   for(j=0;j<25;j++){
      for(k=0;k<8;k++){
         PORTB = halfarr[k] << 7;
         for(i=0;i<500000;i++){}
```

```
        }        }
      appData.state = APP_STATE_SERVICE_TASKS;
      break;      }
   case CCWFS:
   {         i=0;j=0;k=0;
      for(j=0;j<50;j++){
         for(k=0;k<8;k++){
            PORTB = halfarr[k] << 7;
            for(i=0;i<500000;i++){}
         }        }
      appData.state = APP_STATE_SERVICE_TASKS;
      break;      }
   /* TODO: implement your application state machine.*/
   /* The default state should never be executed. */
   default:      {
      break;     }
```

Discussion

We used interrupts handle the changes that happens to the buttons to Direction and the Step mode for the Stepper motor to either go CW or CCW or half step or full step.

Conclusion

This lab we learnt how to work with stepper motor and how to control its direction as well as the step mode by using buttons through interrupts.