Beste Öztop 2375624
Deniz Karakay 2443307 28.11.2022

EE447 Introduction to Microprocessors
Preliminary Work-3

Question-1

Write a C function that which sends a GPIO port the necessary signals to demonstrate the Full Step Mode in both directions (clockwise or counterclockwise).

In order to obtain the behavior described above, we first tried the C code given in page 6 of the Experimental Manual. We have managed to turn the LED on and off (with green color, PF3) for discrete time instants. Then, we tried to light up 4 different colors (0x0E for white, 0x08 for green -PF3, 0x04 for blue-PF2, 0x02 for red-PF1). The resultant code is as below:

```
#include "TM4C123GH6PM.h"

int t = 0;

void init_func(void){
      SYSCTL->RCGCGPIO |= 0x20;
      GPIOF->DIR |= 0x0E;
      GPIOF->DEN |= 0x0E;

      SysTick->LOAD = 3199998;
      SysTick->CTRL = 7;
      SysTick->VAL = 0;
}


void SysTick_Handler (void){
      if(t < 2){
            GPIOF->DATA ^= 2;
      }
      else if(t < 4){
            GPIOF->DATA ^= 4;
      }
      else if(t < 6){
            GPIOF->DATA ^= 8; //0x00001000
      }
      else if(t < 8){
            GPIOF->DATA ^= 14; //0x00001110
      }
      else{
            t=0;
            return;
      }
      t+=1;

}

int main(void){
      init_func();
      while(1);
}
```

Beste Öztop 2375624
Deniz Karakay 2443307 28.11.2022

The following section of code is then written to turn the step motor in both clockwise and counterclockwise directions. To do so, we are making one out of four poles ON at a time just like we have lit ON and OFF four colors. We have set the CW and CCW rotations to occur continuously for this part for certain time intervals to obtain at least 1 full rotation for each direction (the counter values below are set to satisfy this, and hve been found manually by observing the step motor behavior).

```
#include "TM4C123GH6PM.h"

int direct = 0;
void init_func(void){
	SYSCTL->RCGCGPIO |= 0x02;
	GPIOB->DIR |= 0x0F;
	GPIOB->DEN |= 0x0F;
	GPIOB->DATA |= 0x01;

	SysTick->LOAD = 99999;
	SysTick->CTRL = 7;
	SysTick->VAL = 0;
}

void step_motor_rotate(int dir){
	if(dir == 0){ // CW
		GPIOB->DATA *= 2;
		if(GPIOB->DATA == 8){
			GPIOB->DATA = 1;
			return;
		}
	}
	else if(dir ==1){ //CCW
		GPIOB->DATA /= 2;
		if(GPIOB->DATA == 1){
			GPIOB->DATA = 8;
			return;
		}

	}
}

void SysTick_Handler (void){
	step_motor_rotate(direct);
}

int main(void){
	init_func();
	int counter = 0;
	while(1){
		if(counter == 1000000){
			GPIOB->DATA = 1;
			direct = 0;
		}
		else if(counter == 2000000){
			GPIOB->DATA = 8;
			direct = 1;
		}else if(counter == 2000001){
			counter = 0;
		}
		counter += 1;
	}
}
```

Direction-0 indicates the clockwise rotation, the pins 0,1,2,3 are ON consecutively

Direction-1 indicates the counterclockwise rotation, the pins 3,2,1,0 are ON consecutively

The interrupt subroutine calls for our rotation function

Beste Öztop 2375624
Deniz Karakay 2443307 28.11.2022

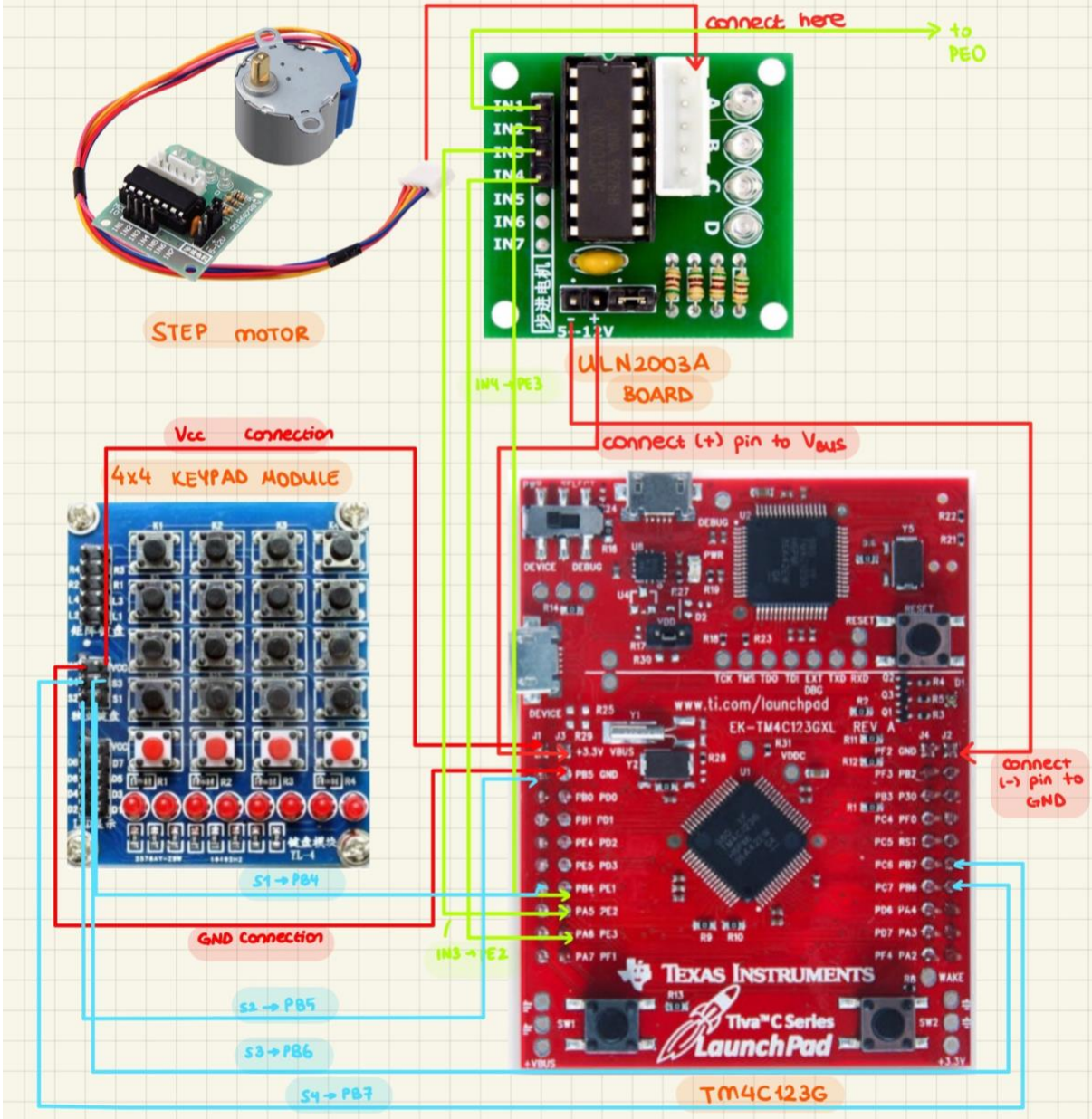
Question-2

Question - 2

2. (10%) Now, you will design a system that has two inputs from push buttons and provides a step to the stepper motor upon input. One button is to provide a step for clockwise rotation and the other is for counterclockwise rotation. You will use 4 buttons of the 4x4 Keypad Module introduced in Experiment-2. Draw the necessary connections between TM4C123G, ULN2003A's board, 4x4 Keypad Module and stepper motor.

* Just like the previous experiment, we can utilize PORT-B (4,5,6,7 pins) to control the 4 buttons of the keypad module. The following connections are to be done...

→ Choose I = 0111 to indicate CW rotation. (Push button-1 is pressed)
→ Choose I = 1011 to indicate CCW rotation. (Push button-2 is pressed)



Beste Öztop 2375624
Deniz Karakay 2443307 28.11.2022

Question-3

*According to your hardware design in step-2, write a C program that, in an infinite loop, gives a step **upon the release** of one button and gives a step in the opposite direction **upon the release** of the other button. You may assume that the other button is never pushed until the pressed button is released. The response of the motor should be after the button release. **You should be aware of bouncing inherent in the buttons**.*

```
#include "TM4C123GH6PM.h"

int direct = -1;
void init_func(void){
	SYSCTL->RCGCGPIO |= 0x12; //Make both PORT B and PORT E open
	GPIOE->DIR |= 0x0F;
	GPIOE->DEN |= 0x0F;
	GPIOE->DATA |= 0x00;

	GPIOB->DIR |= 0x0F;
	GPIOB->DEN |= 0xFF;
	//GPIOB->DATA |= 0xFF;


	/*GPIOE->AMSEL &=~0x00;
	GPIOE->PCTL &=~0x00000000;
	GPIOE->DIR |= 0x0F;
	GPIOE->AFSEL &=~0x00;
	GPIOE->DEN |= 0xFF;
	GPIOE->DATA|= 0xF3;
*/
	SysTick->LOAD = 159999;
	SysTick->CTRL = 7;
	SysTick->VAL = 0;
}

void step_motor_rotate(int dir){
	if(dir == 0){ // CW
		GPIOE->DATA *= 2;
		if(GPIOE->DATA == 8){
			GPIOE->DATA = 1;
			return;
		}
	}
	else if(dir ==1){ //CCW
		GPIOE->DATA /= 2;
		if(GPIOE->DATA == 1){
			GPIOE->DATA = 8;
			return;
		}

	}
}

void SysTick_Handler (void){
	step_motor_rotate(direct);
}



int main(void){
	init_func();
	int state = -1;
	while(1){
		while (GPIOB->DATA == 0xE0){
					GPIOE->DATA = 1;
					state = 0;
		}
		while (GPIOB->DATA == 0xD0){
					GPIOE->DATA = 8;
							state = 1;
		}
			if(state!=-1){
			direct = state;
		}

	}
	}
```

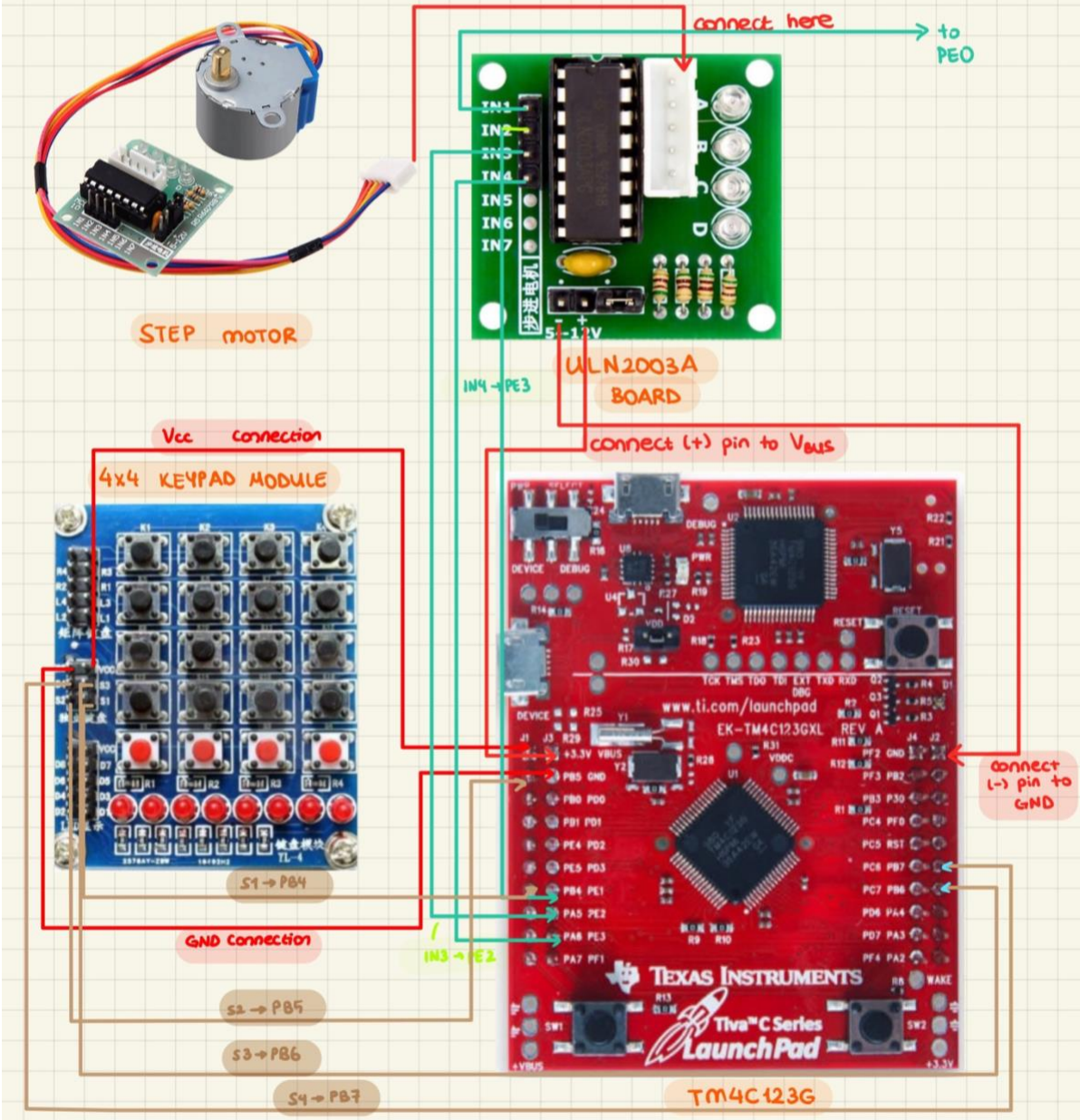
Beste Öztop 2375624
Deniz Karakay 2443307 28.11.2022

Question-4

Question-4

4. (5%) At this stage, you will design a system that has 4 inputs from push buttons to control a stepper motor. One button is for speeding up, one is for slowing down, the other two are for directions. You will use 4 buttons of the 4x4 Keypad Module introduced in Experiment-2. Draw the necessary connections between TM4C123G, ULN2003A's board, 4x4 Keypad Module and stepper motor.

* We can again use PORT-B 4,5,6,7 pins to control the 4 buttons of the keypad module.
  - → Choose I = 0111 to indicate speed up. (Push button-1 is pressed)
  - → Choose I = 1011 to indicate speed down. (Push button-2 is pressed)



Beste Öztop 2375624
Deniz Karakay 2443307 28.11.2022

Question-5

According to your hardware design in part-4, write a C program that, in an infinite loop, drives a stepper motor speed and direction of which can be controlled by external push buttons.
You may assume that no button is never pushed until a pressed button is released. The controls should be applied upon releasing the corresponding button. You should be aware of bouncing inherent in the buttons.

```
#include "TM4C123GH6PM.h"

int direct = -1;
void init_func(void){
        SYSCTL->RCGCGPIO |= 0x12; //Make both PORT B and PORT E open
        GPIOE->DIR |= 0x0F;
        GPIOE->DEN |= 0x0F;
        GPIOE->DATA |= 0x00;

        GPIOB->DIR |= 0x0F;
        GPIOB->DEN |= 0xFF;

        SysTick->LOAD = 70000;
        SysTick->CTRL = 7;
        SysTick->VAL = 0;
}

void step_motor_rotate(int dir){
        if(dir == 0){ // CW
                GPIOE->DATA *= 2;
                if(GPIOE->DATA == 8){
                        GPIOE->DATA = 1;
                        return;
                }
        }
        else if(dir ==1){ //CCW
                GPIOE->DATA /= 2;
                if(GPIOE->DATA == 1){
                        GPIOE->DATA = 8;
                        return;
                }

        }
}

void SysTick_Handler (void){
        step_motor_rotate(direct);
}



int main(void){
        init_func();
        int state = -1;
        int speedState = -1;
        while(1){
                while (GPIOB->DATA == 0xE0){
                                        GPIOE->DATA = 1;
                                        state = 0;
                }
                while (GPIOB->DATA == 0xD0){
                                        GPIOE->DATA = 8;
                                                                state = 1;
                }
                while (GPIOB->DATA == 0xB0){
                                        speedState = 2;
                }
                while (GPIOB->DATA == 0x70){
                                        speedState = 3;
                }

                if(state == 0 || state == 1){
                        direct = state;
                }

                if(speedState == 2 && SysTick->LOAD > 45000){
                                SysTick->LOAD -= 10000;
                        speedState = -1;

                }
                if(speedState == 3 && SysTick->LOAD < 100000){
                                SysTick->LOAD += 10000;
                        speedState = -1;

                }


        }
        }
```

Beste Öztop 2375624
Deniz Karakay 2443307 28.11.2022

Explanation:

In this question, just like in the 3rd one, we decided to use PORT-E pins 0,1,2,3 for output (to be given to the step motor) and PORT-B pins 4,5,6,7 for the input to be taken from the push buttons on the keyboard. Different from the other part, we used all 4 buttons in this question. The first two buttons are used to control the rotating direction of the motor. Please note that we have set an initial speed for our motor to rotate before the speed-up or speed-down commands. Then, assuming that no two buttons can be pushed at the same time, we have selected 4 cases representing the push situation of these buttons. When we want to speed up or speed down the motor, with the help of our if statement, we decrease or increase the SysTick LOAD value which is the main variable controlling our motor's speed. While doing so, we check the LOAD value to make sure that our motor never stops. Then, with the help of our while(1) infinite loop, we managed to keep the motor rotating and open to any commands that can be prompted by the user, as requested in the question.