Preliminary Work

Deniz Soysal 2305332

November 2024

1)

MCU	KEYPAD
PA0	S1
PA1	S2

Table 1: Microcontroller keypad connections

MCU	MOTOR
PB0	OUT0
PB1	OUT1
PB2	OUT2
PB3	OUT3

Table 2: Microcontroller-motor connections

2)

```
#include "TM4C123GH6PM.h"

void GPIO_Init(void) {
    // enable the clock
    SYSCTL->RCGCGPIO |= (1 << 0) | (1 << 1);
    while (!(SYSCTL->PRGPIO & ((1 << 0) | (1 << 1))));  // Wait until ports are ready

    // configure Port A pins
    GPIOA->DIR &= ~0x03;
    GPIOA->DEN |= 0x03;
    GPIOA->PUR |= 0x03;

    // configure Port B pins
    GPIOB->DIR |= 0x0F;
    GPIOB->DEN |= 0x0F;
    GPIOB->DATA &= ~0x0F;
}
```

3)

```
#include "TM4C123GH6PM.h"

void Stepper_Control(int step_no) {
    // Ensure step_no is valid (1 to 4)
    if (step_no < 1 || step_no > 4) return;

    // Clear current outputs (PB0-PB3) without affecting other pins
    GPIOB->DATA &= ~0x0F; // Clear PB0-PB3

    // Set outputs based on the step number
    switch (step_no) {
        case 1:
            GPIOB->DATA |= 0x01;
            break;
        case 2:
            GPIOB->DATA |= 0x02;
            break;
        case 3:
            GPIOB->DATA |= 0x04;
            break;
        case 4:
            GPIOB->DATA |= 0x08;
            break;
        default:
     break;
    }
}
```

4)

```
#include "TM4C123GH6PM.h"

void GPIO_Init(void);
void Stepper_Control(int step_no);
void Delay(int ms);


int step = 1;

int main() {
    GPIO_Init();

    while (1) {
        // Check if Button S1 (Clockwise) is released
        if ((GPIOA->DATA & 0x01) == 0) { // PA0 is low (button pressed)
            while ((GPIOA->DATA & 0x01) == 0); // Eait release
            Delay(20); // debounce delay
            //one step clockwise
            Stepper_Control(step);
            step = (step % 4) + 1; // increment step
        }

        // Check if Button S2 (Counterclockwise) is released
        if ((GPIOA->DATA & 0x02) == 0) { // PA1 is low (button pressed)
            while ((GPIOA->DATA & 0x02) == 0); // wait for release
            Delay(20); // debounce delay
            // one step counterclockwise
            step = (step == 1) ? 4 : (step - 1); // decrement step
            Stepper_Control(step);
        }
    }
}


void GPIO_Init(void) {
    // enable clock for Ports A and B
    SYSCTL->RCGCGPIO |= (1 << 0) | (1 << 1);
    while (!(SYSCTL->PRGPIO & ((1 << 0) | (1 << 1)))); // for clock stabilization

    // configure PA0 and PA1
    GPIOA->DIR &= ~0x03;
    GPIOA->DEN |= 0x03;
    GPIOA->PUR |= 0x03;
    // configure PB0-PB3
    GPIOB->DIR |= 0x0F;
    GPIOB->DEN |= 0x0F;
    GPIOB->DATA &= ~0x0F;
}

// Motor Control
void Stepper_Control(int step_no) {
    // ensure step_no is valid (1 to 4)
    if (step_no < 1 || step_no > 4) return;

    // clear current outputs (PB0-PB3)
    GPIOB->DATA &= ~0x0F;

    switch (step_no) {
        case 1:
            GPIOB->DATA |= 0x01;
            break;
        case 2:
            GPIOB->DATA |= 0x02;
            break;
        case 3:
            GPIOB->DATA |= 0x04;
            break;
        case 4:
            GPIOB->DATA |= 0x08;
            break;
    }
}

// delay
void Delay(int ms) {
    int i, j;
    for (i = 0; i < ms; i++) {
        for (j = 0; j < 3180; j++); // approximate 1 ms delay on 16 MHz clock
    }
}
```

5)

MCU	KEYPAD	function
PA0	S1	speed up
PA1	S2	slow down
PA2	S3	clockwise
PA3	S4	counter-clockwise

Table 3: Controller-keypad connections

MCU	MOTOR
PB0	OUT0
PB1	OUT1
PB2	OUT2
PB3	OUT3

Table 4: Controller-motor connections

- VBUS (5V from TM4C123G board) powers the ULN2003A and stepper motor.
- GND is shared among the TM4C123G board, ULN2003A, and the stepper motor.

6)

```
#include "TM4C123GH6PM.h"

// step rates
const int stepRates[] = {4, 10, 50, 100, 250, 500};
int stepRateIndex = 2; // speed
int direction = 1;     // clockwise direction
int currentStep = 1;   // sep pos

// function prototypes
void GPIO_Init(void);
void SysTick_Init(void);
void Stepper_Control(int step_no);
void Delay_us(int delay);
void SysTick_Handler(void);

// global variable for systick timing
volatile int delay_ms = 0;

int main() {
    GPIO_Init();
    SysTick_Init();

    while (1) {
        // Speed Up
        if ((GPIOA->DATA & 0x01) == 0) { // PA0 pressed
            while ((GPIOA->DATA & 0x01) == 0); // wait for release
            Delay_us(20000);                   // debounce
            if (stepRateIndex < 5) stepRateIndex++; // speed up
        }

        // Button S2 (Slow Down)
        if ((GPIOA->DATA & 0x02) == 0) { // PA1 pressed
            while ((GPIOA->DATA & 0x02) == 0);
            Delay_us(20000);
            if (stepRateIndex > 0) stepRateIndex--; // slow down
        }

        // Button S3 (Clockwise)
        if ((GPIOA->DATA & 0x04) == 0) { // PA2 pressed
            while ((GPIOA->DATA & 0x04) == 0);
            Delay_us(20000);
            direction = 1;                     // clockwise direction
        }

        // Button S4 (Counterclockwise)
        if ((GPIOA->DATA & 0x08) == 0) { // PA3 pressed
            while ((GPIOA->DATA & 0x08) == 0);
            Delay_us(20000);
            direction = -1;                    // counterclockwise direction
        }

        // motor control
        currentStep += direction; // update step position
        if (currentStep > 4) currentStep = 1;
        if (currentStep < 1) currentStep = 4;
        Stepper_Control(currentStep);

        // delay based on step rate
        Delay_us(1000000 / stepRates[stepRateIndex]);
    }
}


void GPIO_Init(void) {
    SYSCTL->RCGCGPIO |= (1 << 0) | (1 << 1); // Enable clocks for Ports A and B
    while (!(SYSCTL->PRGPIO & ((1 << 0) | (1 << 1)))); // Wait for clock ready

    // configure PA0-PA3 as inputs
    GPIOA->DIR &= ~0x0F;
    GPIOA->DEN |= 0x0F;
    GPIOA->PUR |= 0x0F;

    // configure PB0-PB3 as outputs
    GPIOB->DIR |= 0x0F;
    GPIOB->DEN |= 0x0F;
    GPIOB->DATA &= ~0x0F;
}

// systick initialization
void SysTick_Init(void) {
    SysTick->LOAD = 16000 - 1; // 1ms interval (16MHz clock)
    SysTick->VAL = 0;
    SysTick->CTRL = 0x07;      // Enable SysTick with interrupt
}

// for precise delays
void SysTick_Handler(void) {
    if (delay_ms > 0) delay_ms--;
}

// blocking Delay
void Delay_us(int delay) {
    delay_ms = delay / 1000;
    while (delay_ms > 0);
}

// Stepper Motor Control
void Stepper_Control(int step_no) {
    GPIOB->DATA &= ~0x0F; // Clear PB0-PB3
    switch (step_no) {
        case 1: GPIOB->DATA |= 0x01; break;
        case 2: GPIOB->DATA |= 0x02; break;
        case 3: GPIOB->DATA |= 0x04; break;
        case 4: GPIOB->DATA |= 0x08; break;
    }
}
```