
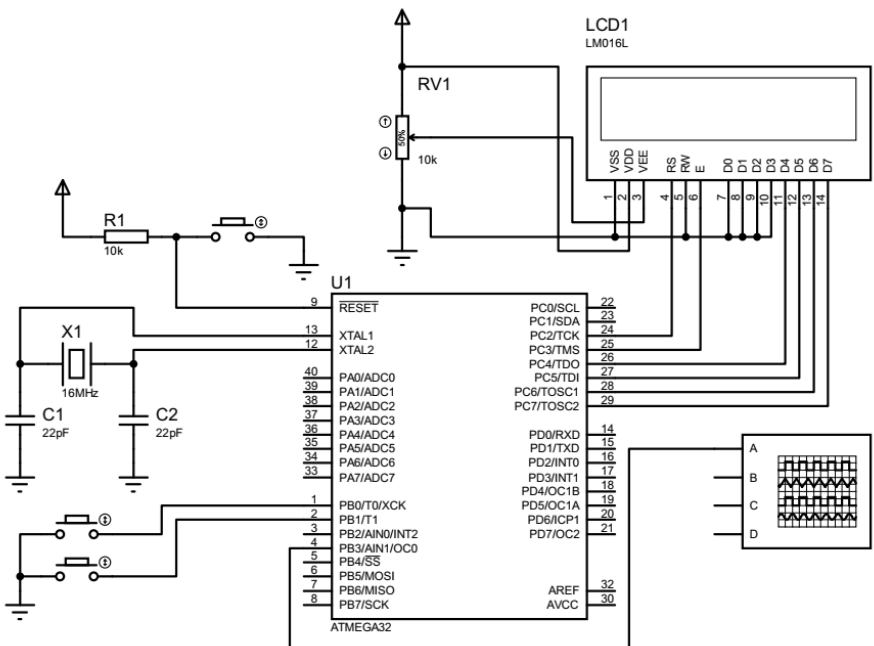




| | | | | | | | | | | | | | | | |
|---|--|--|------------|--|-------|--|-------|---|---------|-----------------|-------|--|---------|---|---------|
|  | ใบงานที่ 6 | | หน้าที่ 1 | | | | | | | | | | | | |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | | หน่วยที่ 3 | | | | | | | | | | | | |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป | | | | | | | | | | | | | | |
| ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์ | | | | | | | | | | | | | | | |
| <p>จุดประสงค์การเรียนรู้การสอน</p> <p>จุดประสงค์ทั่วไป</p> <ol style="list-style-type: none">1. เพื่อให้มีทักษะในการออกแบบและต่อวงจรสัญญาณนาฬิกาด้วยไมโครคอนโทรลเลอร์2. เพื่อให้มีทักษะในการออกแบบและต่อวงจร ไฟวิง วงจรนับ บนอุปกรณ์ LED ,7-Segment และ LCD3. เพื่อให้มีทักษะในการใช้งานโปรแกรม Proteus ในการจำลองการทำงานของวงจรสัญญาณนาฬิกาและวงจรมอนิเตอร์แสดงผลบนอุปกรณ์ LED ,7-Segment และ LCD4. เพื่อให้มีเจตคติที่ดีต่อการเรียนรู้เรื่อง การสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์ <p>จุดประสงค์เชิงพฤติกรรม</p> <ol style="list-style-type: none">1. เตรียมเครื่องมือ วัสดุ และอุปกรณ์ได้ถูกต้อง2. ออกแบบวงจรสัญญาณนาฬิกาได้3. ต่อวงจรสัญญาณนาฬิกาได้4. ออกแบบวงจรมอนิเตอร์ได้5. ต่อวงจรมอนิเตอร์ได้6. ใช้โปรแกรม Proteus เพื่อต่อวงจรทดสอบการทำงานของวงจรสัญญาณนาฬิกาและวงจรมอนิเตอร์ได้7. ปฏิบัติงานได้อย่างถูกต้อง ประณีต รอบคอบ ปลอดภัย และเสร็จภายในเวลาที่กำหนด อย่างมีคุณธรรม จริยธรรม <p>คุณธรรม จริยธรรม ค่านิยมและคุณลักษณะที่พึงประสงค์</p> <ol style="list-style-type: none">1. มีความซื่อสัตย์สุจริต2. มีความอดทนอดกลั้น3. มีความรักสามัคคีในการปฏิบัติงาน4. มีมนุษยสัมพันธ์ในการทำงาน5. มีการคำนึงถึงความปลอดภัยในการปฏิบัติงาน <p>เครื่องมือ วัสดุ และอุปกรณ์การปฏิบัติงาน</p> <table><tr><td>1. ใบความรู้เรื่องการสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลฯ</td><td>1 ชุด</td></tr><tr><td>2. ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์</td><td>1 ชุด</td></tr><tr><td>3. บอร์ดใช้งานไมโครคอนโทรลเลอร์ AT89C51ED2 , PIC16F877 และ ATMEGA32 อย่างละ</td><td>1 บอร์ด</td></tr><tr><td>4. มัลติมิเตอร์</td><td>1 ตัว</td></tr><tr><td>5. เครื่องโปรแกรมไมโครคอนโทรลเลอร์ตระกูล AVR</td><td>1 บอร์ด</td></tr><tr><td>6. เครื่องโปรแกรมไมโครคอนโทรลเลอร์ตระกูล PIC16F</td><td>1 บอร์ด</td></tr></table> | | | | 1. ใบความรู้เรื่องการสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลฯ | 1 ชุด | 2. ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ | 1 ชุด | 3. บอร์ดใช้งานไมโครคอนโทรลเลอร์ AT89C51ED2 , PIC16F877 และ ATMEGA32 อย่างละ | 1 บอร์ด | 4. มัลติมิเตอร์ | 1 ตัว | 5. เครื่องโปรแกรมไมโครคอนโทรลเลอร์ตระกูล AVR | 1 บอร์ด | 6. เครื่องโปรแกรมไมโครคอนโทรลเลอร์ตระกูล PIC16F | 1 บอร์ด |
| 1. ใบความรู้เรื่องการสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลฯ | 1 ชุด | | | | | | | | | | | | | | |
| 2. ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ | 1 ชุด | | | | | | | | | | | | | | |
| 3. บอร์ดใช้งานไมโครคอนโทรลเลอร์ AT89C51ED2 , PIC16F877 และ ATMEGA32 อย่างละ | 1 บอร์ด | | | | | | | | | | | | | | |
| 4. มัลติมิเตอร์ | 1 ตัว | | | | | | | | | | | | | | |
| 5. เครื่องโปรแกรมไมโครคอนโทรลเลอร์ตระกูล AVR | 1 บอร์ด | | | | | | | | | | | | | | |
| 6. เครื่องโปรแกรมไมโครคอนโทรลเลอร์ตระกูล PIC16F | 1 บอร์ด | | | | | | | | | | | | | | |


| | | | |
|--|--|-----------|------------|
|  | ใบงานที่ 6 | | หน้าที่ 2 |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | | หน่วยที่ 3 |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป | | |
| ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์ | | | |
| 7. ออสซิลโลสโคป | | 1 เครื่อง | |
| 8. สายต่อวงจร | | 40 เส้น | |
| 9. คอมพิวเตอร์ที่ติดตั้งโปรแกรม Proteus 7.8SP2 | | 1 เครื่อง | |
| 10. อุปกรณ์แสดงผลต่าง ๆ ได้แก่ LED, 7-Segment และ Character LCD 16x2 | | | |
| 11. ตัวต้านทานค่า 470Ω และ 10kΩ | | | |
| 12. สวิตช์ Button | | | |
| ข้อควรระวัง | | | |
| ขณะปฏิบัติการทดลองห้ามหยอกล้อเล่นกัน และห้ามนำสายต่อลัดวงจรระหว่างขั้วไฟ Vcc 5V กับ GND | | | |
| ข้อเสนอแนะ | | | |
| ควรตรวจสอบความพร้อมของเครื่องมือและอุปกรณ์ทุกครั้งก่อนใช้งาน รวมถึงควรศึกษาและปฏิบัติตามขั้นตอนการปฏิบัติงานอย่างเคร่งครัด | | | |
| ลำดับขั้นการปฏิบัติงาน (Step Operation) | | | |
| 1. การสร้างสัญญาณนาฬิกาด้วยไมโครคอนโทรลเลอร์ และแสดงผลการทำงานบน Character LCD | | | |
| 1.1 ให้นักศึกษาใช้ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ ร่วมกับบอร์ดใช้งานไมโครคอนโทรลเลอร์ ATMEGA32 เพื่อต่อวงจรในรูปที่ 1.1 เพื่อทดลองการสร้างสัญญาณนาฬิกาที่ความถี่ 3.9kHz ออกที่ขา OC0 ด้วยโมดูล Timer0 โหมด PWM และสามารถปรับค่า Duty Cycle ของสัญญาณได้ด้วยสวิตช์ที่ขา PB0 และ PB1 | | | |
|  | | | |
| รูปที่ 1.1 รูปวงจรการสร้างสัญญาณนาฬิกาที่ความถี่ 3.9kHz ด้วยไมโครคอนโทรลเลอร์ ATMEGA32 | | | |

| | | | |
|--|--|--|------------|
|  | ใบงานที่ 6 | | หน้าที่ 3 |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | | หน่วยที่ 3 |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป | | |
| ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์ | | | |
| <p>1.2 จากวงจรในข้อที่ 1.1 ให้นักศึกษาเขียนโปรแกรมตามตัวอย่างต่อไปนี้โดยใช้โปรแกรม AVR Studio 6.2 ตามขั้นตอนในใบความรู้เรื่องภาษาซีกับไมโครคอนโทรลเลอร์ และทำการ Build โปรแกรมจนกว่าไม่เกิด Error เพื่อให้ได้ไฟล์นามสกุล .hex เพื่อนำไปใช้งานต่อไป</p> <pre>#include <avr/io.h> #define F_CPU 16000000UL #include <util/delay.h> #define lcd_ddr DDRC #define lcd_port PORTC #define rs_lcd 2 #define en_lcd 3 #define swUp 0 #define swDown 1 const unsigned char ascii[16] = "0123456789ABCDEF"; void lcd_busy(int time) { lcd_port &= ~(1<<en_lcd); for(;time>0;time--) _delay_us(500); lcd_port = (1<<en_lcd); } void lcd_command(unsigned char cmd) { lcd_port &= ~(1<<rs_lcd); lcd_port = (lcd_port & 0x0f) (cmd & 0xf0); lcd_busy(2); lcd_port = (lcd_port & 0x0f) (cmd << 4); lcd_busy(2); lcd_port = (1<<rs_lcd); }</pre> | | | |

| | | | |
|---|--|--|------------|
|  | ใบงานที่ 6 | | หน้าที่ 4 |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | | หน่วยที่ 3 |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป | | |
| ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์ | | | |
| <pre>void lcd_putc(unsigned char dat) { lcd_port = (1<<rs_lcd); lcd_port = (lcd_port & 0x0f) (dat & 0xf0); lcd_busy(2); lcd_port = (lcd_port & 0x0f) (dat << 4); lcd_busy(2); lcd_port = (1<<rs_lcd); } void lcd_puts(char *str) { while(*str != '\0') lcd_putc(*str++); } void lcd_init() { lcd_ddr = 0b11111100; lcd_port = (1<<en_lcd); lcd_port = (1<<rs_lcd); lcd_command(0x33); lcd_command(0x32); lcd_command(0x28); lcd_command(0x0c); lcd_command(0x06); lcd_command(0x01); _delay_ms(500); }</pre> | | | |

| | | |
|---|--|------------|
|  | ใบงานที่ 6 | หน้าที่ 5 |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | หน่วยที่ 3 |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป | |
| ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์ | | |
| <pre>void showDuty(unsigned char duty) { lcd_command(0xc7); lcd_puts(" "); lcd_command(0xc7); if(duty < 10){ lcd_putc(ascii[duty]); lcd_putc('%'); }else if(duty < 100){ lcd_putc(ascii[duty/10]); lcd_putc(ascii[duty%10]); lcd_putc('%'); }else{ lcd_putc(ascii[duty/100]); lcd_putc(ascii[(duty%100)/10]); lcd_putc(ascii[duty%10]); lcd_putc('%'); } } void pwmOC0_init(){ TCCR0 = 0x62; OCR0 = (255 * 10) / 100; } void changDutyOC0(unsigned char duty){ OCR0 = (255 * duty) / 100; }</pre> | | |

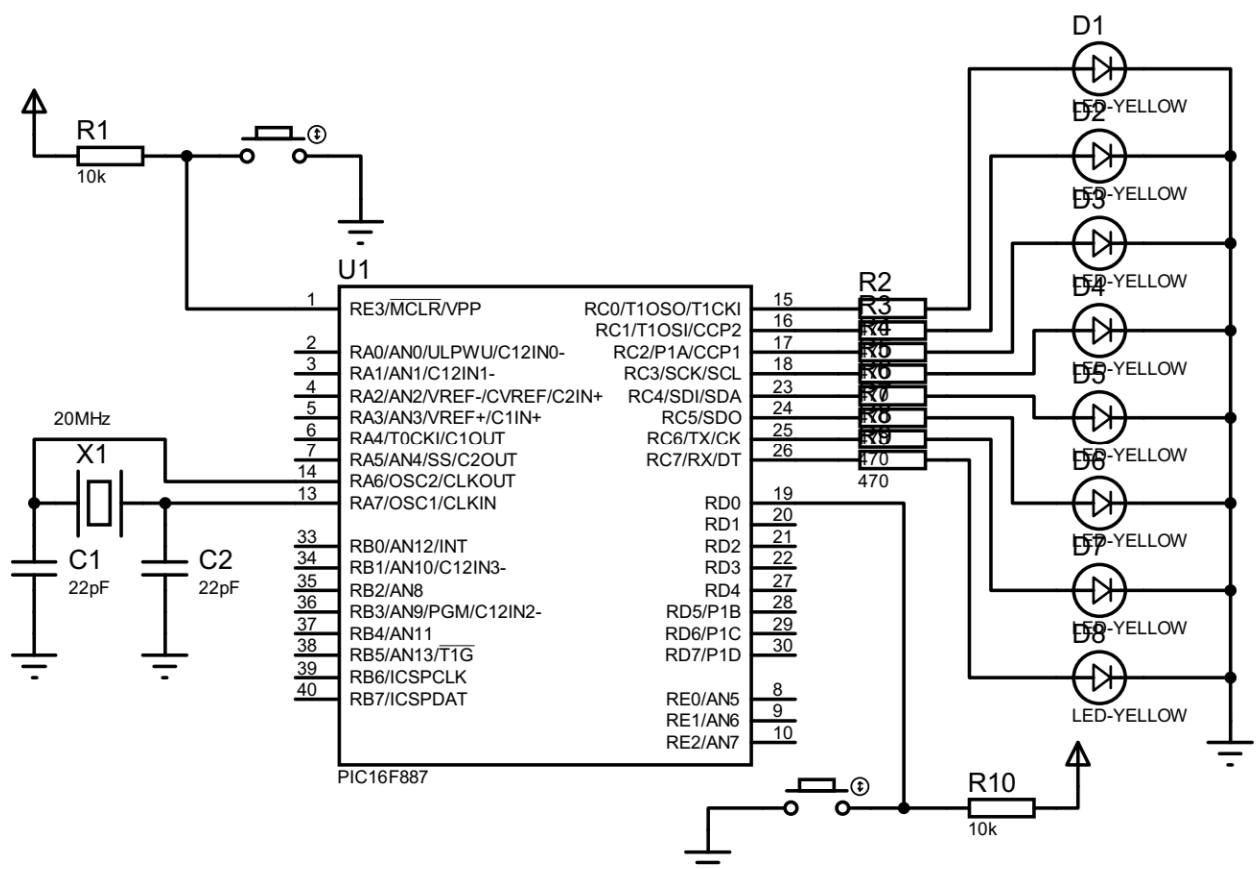
| | | | |
|---|--|--|------------|
|  | ใบงานที่ 6 | | หน้าที่ 6 |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | | หน่วยที่ 3 |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป | | |
| ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์ | | | |
| <pre>int main() { unsigned char duty = 10; DDRB = 0b00001000; DDRB &= ~(1<<swUp); DDRB &= ~(1<<swDown); PORTB = 0b00000011; lcd_init(); lcd_command(0x80); lcd_puts("GenClock 3.9kHz "); lcd_command(0xc0); lcd_puts("Duty = 10%"); pwmOC0_init(); while(1) { if((PINB & (1<<swUp)) == 0){ duty++; if(duty > 100) duty = 100; showDuty(duty); changDutyOC0(duty); _delay_ms(300); } if((PINB & (1<<swDown)) == 0){ duty--; if(duty == 255) duty = 0; showDuty(duty); changDutyOC0(duty); _delay_ms(300); } } return 0; }</pre> | | | |

| | | | |
|---|--|--|------------|
|  | ใบงานที่ 6 | | หน้าที่ 8 |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | | หน่วยที่ 3 |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟlop | | |

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์

2. การเขียนโปรแกรมแสดงผลบนอุปกรณ์ LED


2.1 ให้นักศึกษาใช้ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ ร่วมกับบอร์ดใช้งานไมโครคอนโทรลเลอร์ PIC16F887 เพื่อต่อวงจรในรูปที่ 2.1 เพื่อทดลองการสร้างไฟวิ่งรูปแบบลูกตุ้มนาฬิกาบน LED 8 ดวง ที่ต่อรูปแบบ Active High





รูปที่ 2.1 วงจรไฟวิ่งบน LED 8 ดวง ที่ต่อรูปแบบ Active High ด้วยไมโครคอนโทรลเลอร์ PIC16F887

2.2 จากวงจรในข้อที่ 2.1 ให้นักศึกษาเขียนโปรแกรมตามตัวอย่างต่อไปนี้โดยใช้โปรแกรม MPLAB X ตามขั้นตอนในใบความรู้เรื่องภาษาซีกับไมโครคอนโทรลเลอร์ และทำการ Build โปรแกรมจนกว่าไม่เกิด Error เพื่อให้ได้ไฟล์นามสกุล .hex เพื่อนำไปใช้งานต่อไป

```
#define _XTAL_FREQ 20000000
#include <xc.h>
#define sw RD0
```


| | | |
|--|--|------------|
|  | ใบงานที่ 6 | หน้าที่ 9 |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | หน่วยที่ 3 |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป | |
| ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์ | | |
| <pre>const unsigned char tbLed[] = {0x18,0x28,0x48,0x88,0x48,0x28, 0x18,0x14,0x12,0x11,0x12,0x14}; int main() { char i,st=0; TRISDbits.TRISD0 = 1; TRISC = 0; PORTC = 0; while(1) { if(sw == 0){ while(sw == 0) __delay_ms(10); st = 1; } while(st){ for(i=0;i<12;i++){ PORTC = tbLed[i]; __delay_ms(400); if(sw == 0){ while(sw == 0) __delay_ms(10); st = 0; PORTC = 0; break; } } } } }</pre> | | |

| | | |
|---|--|------------|
|  | ใบงานที่ 6 | หน้าที่ 11 |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | หน่วยที่ 3 |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป | |
| ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์ | | |
| <p>3.2 จากวงจรในข้อที่ 3.1 ให้นักศึกษาเขียนโปรแกรมตามตัวอย่างต่อไปนี้โดยใช้โปรแกรม Keil uVision3 ตามขั้นตอนในใบความรู้เรื่องภาษาซีกับไมโครคอนโทรลเลอร์ และทำการ Build โปรแกรมจนกว่าไม่เกิด Error เพื่อให้ได้ไฟล์นามสกุล .hex เพื่อนำไปใช้งานต่อไป</p> <pre>#include<reg51.h> sbit sw = P1^0; code unsigned char tb7[16] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07, 0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71}; void delay(int time){ char i; for(;time>0;time--) for(i=100;i>0;i--); } void main(){ char i,st=0; P2 = 0; while(1){ if(sw == 0){ while(sw == 0) delay(100); st = 1; } while(st){ for(i=0;i<16;i++){ P2 = tb7[i]; delay(1000); if(sw == 0){ while(sw == 0) delay(100); st = 0; P2 = 0; break; } } } } }</pre> | | |

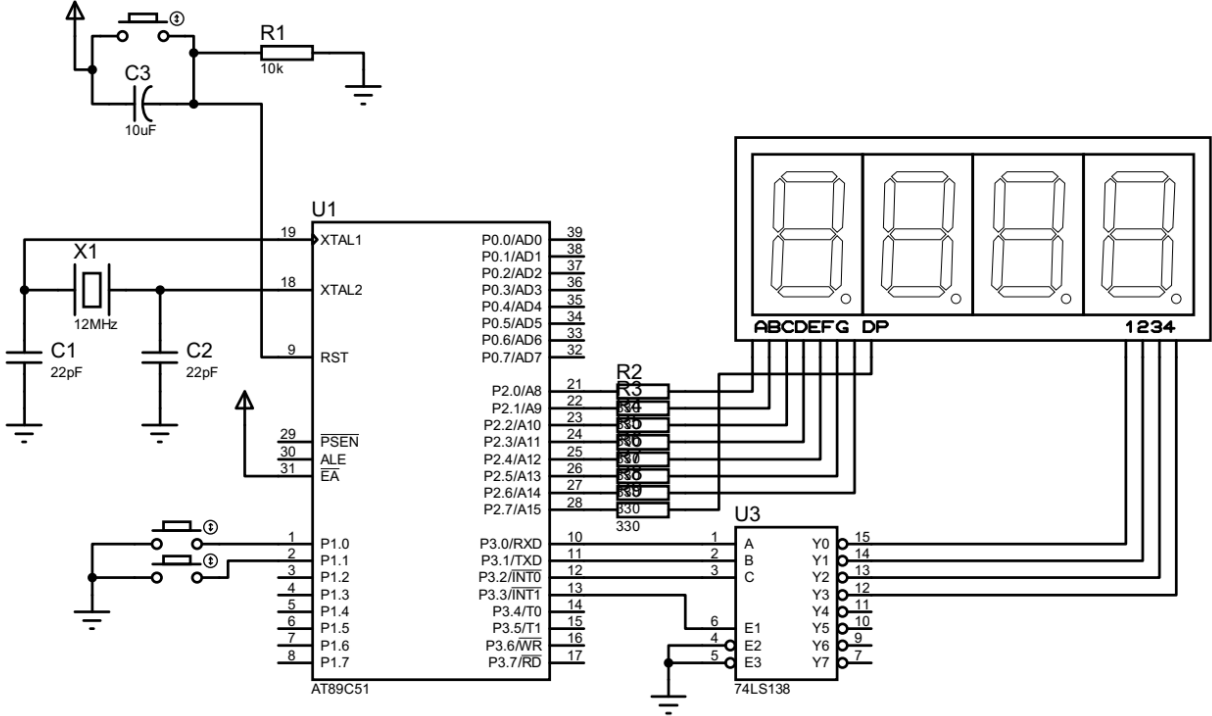
| | | |
|---|---|-------------------|
|  | ใบงานที่ 6 | หน้าที่ 12 |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | หน่วยที่ 3 |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป | |

ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์


3.3 จากข้อที่ 3.2 ให้นักศึกษาใช้โปรแกรม Proteus ISIS ทำการจำลองการทำงานของวงจรเปรียบเทียบกับการทำงานของวงจรจริง โดยการโหลดไฟล์นามสกุล .hex ลงบนตัวไมโครคอนโทรลเลอร์ AT89C51ED2 ด้วยโปรแกรม Flip 3.4.7 และป้อนแหล่งจ่ายให้บอร์ดทดลอง พร้อมทั้งทดลองกดสวิตช์ที่ขา P1.0 เพื่อทำการเริ่มการทำงาน และกดสวิตช์อีกครั้งเพื่อยุติการทำงาน หลังจากนั้นให้บันทึกผลการทดลอง


บันทึกผลการทดลอง


3.4 ให้นักศึกษาใช้ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ ร่วมกับบอร์ดใช้งานไมโครคอนโทรลเลอร์ AT89C51ED2 เพื่อต่อวงจรในรูปที่ 3.2 เพื่อทดลองการทำงานของวงจรนับเลขฐาน 10 สองหลัก



รูปที่ 3.2 วงจรการนับแบบมัลติเพล็กซ์สัญญาณด้วยไมโครคอนโทรลเลอร์ AT89C51ED2

| | | |
|--|--|------------|
|  | ใบงานที่ 6 | หน้าที่ 13 |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | หน่วยที่ 3 |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป | |
| ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์ | | |
| <p>3.5 จากวงจรในข้อที่ 3.4 ให้นักศึกษาเขียนโปรแกรมตามตัวอย่างต่อไปนี้โดยใช้โปรแกรม Keil uVision3 ตามขั้นตอนในใบความรู้เรื่องภาษาซีกับไมโครคอนโทรลเลอร์ และทำการ Build โปรแกรมจนกว่าไม่เกิด Error เพื่อให้ได้ไฟล์นามสกุล .hex เพื่อนำไปใช้งานต่อไป</p> <pre>#include<reg51.h> sbit swUp = P1^0; sbit swDown = P1^1; code unsigned char tb7[16] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07, 0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71}; void delay(int time){ char i; for(;time>0;time--) for(i=100;i>0;i--); } void showNum(int num){ char i; for(i=10;i>0;i--){ P2 = tb7[num/10]; P3 = (P3 & 0xf0) 0x09; delay(10); P2 = 0; P3 &= 0xf0; delay(10); P2 = tb7[num%10]; P3 = (P3 & 0xf0) 0x0a; delay(10); P2 = 0; P3 &= 0xf0; delay(10); } }</pre> | | |

| | | | |
|--|--|--|------------|
|  | ใบงานที่ 6 | | หน้าที่ 14 |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | | หน่วยที่ 3 |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป | | |
| ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์ | | | |
| <pre>void main(){ int i,num=0; while(1){ showNum(num); if(swUp == 0){ num++; if(num > 99) num = 0; for(i=2;i>0;i--) showNum(num); } if(swDown == 0){ num--; if(num < 0) num = 99; for(i=2;i>0;i--) showNum(num); } } }</pre> | | | |
| <p>3.6 จากข้อที่ 3.5 ให้นักศึกษาใช้โปรแกรม Proteus ISIS ทำการจำลองการทำงานของวงจรเปรียบเทียบกับการทำงานของวงจรจริง โดยการโหลดไฟล์นามสกุล .hex ลงบนตัวไมโครคอนโทรลเลอร์ AT89C51ED2 ด้วยโปรแกรม Flip 3.4.7 และป้อนแหล่งจ่ายให้บอร์ดทดลอง พร้อมทั้งทดลองกดสวิตซ์ที่ขา P1.0 และขา P1.1 เพื่อทำเพิ่มค่าการนับหรือลดค่าการนับ หลังจากนั้นให้บันทึกผลการทดลอง</p> <p>บันทึกผลการทดลอง</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> | | | |

| | | | |
|--|--|------------|------------|
|  | ใบงานที่ 6 | | หน้าที่ 15 |
| | ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004 | หน่วยที่ 3 | |
| | ชื่อหน่วย สัญญาณนาฬิกา และวงจรฟลิปฟล็อป | | |
| ชื่อเรื่อง การสร้างสัญญาณนาฬิกา และการควบคุมอุปกรณ์แสดงผลด้วยไมโครคอนโทรลเลอร์ | | | |
| <p>แบบฝึกหัดท้ายการทดลอง</p> <ol style="list-style-type: none">1. ให้นักศึกษาออกแบบวงจรสร้างสัญญาณนาฬิกาที่ความถี่ 4kHz ที่สามารถปรับค่า Duty Cycle ได้ โดยใช้ไมโครคอนโทรลเลอร์เบอร์ PIC16F887 และแสดงผลค่า Duty Cycle บน Character LCD และเขียนโปรแกรมเพื่อทดสอบการทำงาน2. ให้นักศึกษาออกแบบวงจรไฟวิ่งรูปแบบไฟหยดน้ำขนาด 8 บิต แสดงผลบน LED ด้วยไมโครคอนโทรลเลอร์ AT89C51ED2 และเขียนโปรแกรมเพื่อทดสอบการทำงาน3. ให้นักศึกษาออกแบบวงจรนับเลขขนาด 4 หลัก แสดงผลบน 7-Segment แบบมัลติเพล็กซ์ ด้วยไมโครคอนโทรลเลอร์ ATMEGA32 และเขียนโปรแกรมเพื่อทดสอบการทำงาน | | | |

