
	ใบงานที่ 14		หน้าที่ 1										
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8										
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ												
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI													
<p>จุดประสงค์การเรียนรู้การสอน</p> <p>จุดประสงค์ทั่วไป</p> <ol style="list-style-type: none">1. เพื่อให้มีทักษะในการใช้งานการรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI2. เพื่อให้มีทักษะในเขียนโปรแกรมไมโครคอนโทรลเลอร์เพื่อรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI กับอุปกรณ์ภายนอก3. เพื่อให้มีทักษะในการใช้งานโปรแกรม Proteus ในการจำลองการทำงานของไมโครคอนโทรลเลอร์เพื่อรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI กับอุปกรณ์ภายนอก4. เพื่อให้มีเจตคติที่ดีต่อการเรียนรู้เรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI <p>จุดประสงค์เชิงพฤติกรรม</p> <ol style="list-style-type: none">1. เตรียมเครื่องมือ วัสดุ และอุปกรณ์ได้ถูกต้อง2. ต่อบอร์ดเพื่ออ่านค่าฐานเวลาจริงกับอุปกรณ์ DS1307 ด้วยระบบบัส I2C ได้3. ต่อบอร์ดเพื่ออ่านค่าข้อมูล ADC จากอุปกรณ์ MCP3208 ด้วยระบบบัส SPI ได้4. เขียนโปรแกรมเพื่อติดต่ออุปกรณ์ภายนอกผ่านระบบบัส I2C และ SPI ด้วยไมโครคอนโทรลเลอร์ได้5. ใช้โปรแกรม Proteus เพื่อจำลองการทำงานของอุปกรณ์ DS1307 และ MCP3208 ร่วมกับไมโครคอนโทรลเลอร์ได้6. 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. ใบความรู้เรื่องการรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI</td><td>1 ชุด</td></tr><tr><td>2. ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์</td><td>1 ชุด</td></tr><tr><td>3. มัลติมิเตอร์</td><td>1 ตัว</td></tr><tr><td>4. สายต่อวงจร</td><td>40 เส้น</td></tr><tr><td>5. คอมพิวเตอร์ที่ติดตั้งโปรแกรม Proteus 7.8SP2</td><td>1 เครื่อง</td></tr></table>				1. ใบความรู้เรื่องการรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI	1 ชุด	2. ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์	1 ชุด	3. มัลติมิเตอร์	1 ตัว	4. สายต่อวงจร	40 เส้น	5. คอมพิวเตอร์ที่ติดตั้งโปรแกรม Proteus 7.8SP2	1 เครื่อง
1. ใบความรู้เรื่องการรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI	1 ชุด												
2. ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์	1 ชุด												
3. มัลติมิเตอร์	1 ตัว												
4. สายต่อวงจร	40 เส้น												
5. คอมพิวเตอร์ที่ติดตั้งโปรแกรม Proteus 7.8SP2	1 เครื่อง												

	ใบงานที่ 14		หน้าที่ 2
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

6. บอร์ดฐานเวลาจริง DS1307	1 บอร์ด
7. ไอซี MCP3208	1 ตัว
8. ตัวต้านทานปรับค่าได้ 10kΩ	1 ตัว

ข้อควรระวัง

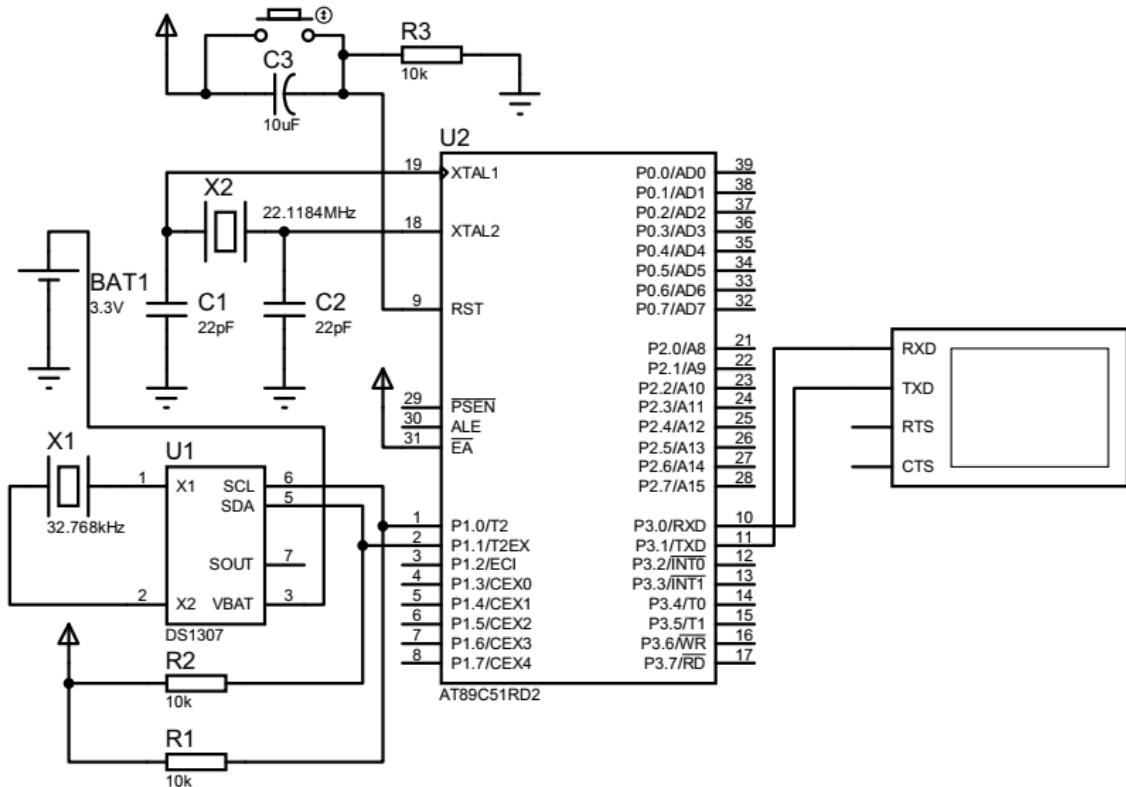
ขณะปฏิบัติการทดลองห้ามหยอกล้อเล่นกัน และห้ามนำสายต่อลัดวงจรระหว่างขั้วไฟ Vcc 5V กับ GND

ข้อเสนอแนะ


ควรตรวจสอบความพร้อมของเครื่องมือและอุปกรณ์ทุกครั้งก่อนใช้งาน รวมถึงควรศึกษาและปฏิบัติตามขั้นตอนการปฏิบัติงานอย่างเคร่งครัด

ลำดับขั้นตอนการปฏิบัติงาน (Step Operation)

- การอ่านค่าฐานเวลาจริงกับอุปกรณ์ DS1307 ด้วยระบบบัส I2C โดยใช้ไมโครคอนโทรลเลอร์ AT89C51ED2
 - ให้นักศึกษาใช้ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ ร่วมกับบอร์ดใช้งานไมโครคอนโทรลเลอร์ AT89C51ED2 เพื่อต่อวงจรในรูปที่ 1.1 เพื่อทดลองการอ่านค่าฐานเวลาจริงจาก DS1307



รูปที่ 1.1 รูปวงจรการอ่านค่าฐานเวลาจริงจาก DS1307 ด้วย AT89C51ED2

	ใบงานที่ 14	หน้าที่ 3
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

1.2 จากวงจรในข้อที่ 1.1 ให้นักศึกษาเขียนโปรแกรมตามตัวอย่างต่อไปนี้โดยใช้โปรแกรม Keil uVision3 ตามขั้นตอนในใบความรู้เรื่องภาษาซีกับไมโครคอนโทรลเลอร์ และทำการ Build โปรแกรมจนกว่าไม่เกิด Error เพื่อให้ได้ไฟล์นามสกุล .hex เพื่อนำไปใช้งานต่อไป

```
#include <at89c51xd2.h>

#define XTAL 22118400
#define scl  P1_0
#define sda  P1_1


unsigned char ascii[] = "0123456789ABCDEF";


void delay(int time){
    unsigned char x;
    for(;time>0;time--)
        for(x=200;x>0;x--);
}


unsigned char uart_getc(){
    while(RI == 0);
    RI = 0;
    return SBUF;
}


void uart_putc(unsigned char dat){ //Send 1 Character
    SBUF = dat;
    while(TI == 0);
    TI = 0;
}

void uart_puts(unsigned char *str){ //Send String
    while(*str != '\0') uart_putc(*str++);
}
```

	ใบงานที่ 14	หน้าที่ 4
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI		
<pre>void uart_init(int baud){ //Use XTAL 22.1184MHz SCON = 0x50; //RX Enable Data 8 bit Baud rate variable TMOD = (TMOD & 0x0f) 0x20; // Timer1 mode 8 Auto reload TH1 = 256 - ((XTAL/384)/baud); TL1 = TH1; TR1 = 1; } void i2c_delay(){ unsigned char i; for(i=15;i>0;i--); } void i2c_clk(){ i2c_delay(); scl = 1; i2c_delay(); scl = 0; } void i2c_start(){ if(scl) scl = 0; sda = 1; scl = 1; i2c_delay(); sda = 0; i2c_delay(); scl = 0; }</pre>		

	ใบงานที่ 14		หน้าที่ 5
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>void i2c_stop(){ if(scl) scl = 0; sda = 0; i2c_delay(); scl = 1; i2c_delay(); sda = 1; } bit i2c_wrdata(unsigned char dat){ bit data_bit; unsigned char i; for (i=0;i<8;i++) { data_bit = dat & 0x80; sda = data_bit; i2c_clk(); dat = dat<<1; } sda = 1; i2c_delay(); scl = 1; i2c_delay(); data_bit = sda; scl = 0; i2c_delay(); return (data_bit); }</pre>			

	ใบงานที่ 14		หน้าที่ 6
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>unsigned char i2c_rddata(){ bit rd_bit; unsigned char i,dat; dat = 0x00; for (i=0;i<8;i++){ i2c_delay(); scl = 1; i2c_delay(); rd_bit = sda; dat = dat<<1; dat = dat rd_bit; scl = 0; } sda = 1; i2c_delay(); i2c_clk(); scl = 1; return (dat); } void i2cWriteByte(unsigned char control,unsigned char addr,unsigned char dat){ bit err = 0; do{ i2c_start(); err = i2c_wrdata(control); if(err == 0){ err = i2c_wrdata(addr); if(err == 0){ err = i2c_wrdata(dat); i2c_stop(); } } }while(err); }</pre>			


	ใบงานที่ 14	หน้าที่ 7
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	


ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI
--

```
unsigned char i2cReadByte(unsigned char control,unsigned char addr){
    bit err = 0;
    unsigned char dat;
    do{
        i2c_start();
        err = i2c_wldata(control);
        if(err == 0){
            err = i2c_wldata(addr);
            if(err == 0){
                i2c_start();
                err = i2c_wldata(control+1);
                if(err == 0){
                    dat = i2c_rddata();
                    i2c_stop();
                }
            }
        }
    }while(err);
    return (dat);
}

void i2c_init(){
    scl = 1;
    sda = 1;
}

void showSec(unsigned char dat){
    uart_puts("SEC = ");
    uart_putc(ascii[dat/0x10]);
    uart_putc(ascii[dat%0x10]);
    uart_puts("\n\r");
}
```

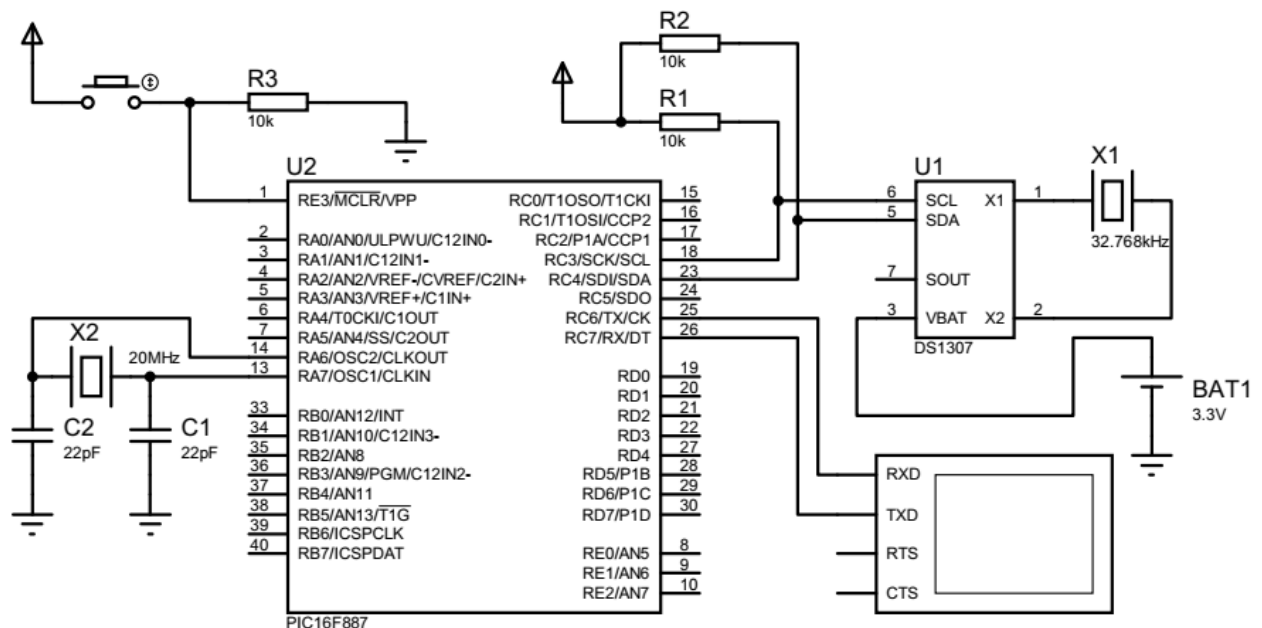
	ใบงานที่ 14		หน้าที่ 8
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>void main(){ unsigned char sec=0,buf=0; uart_init(9600); i2c_init(); uart_puts("Test DS1307 Read Write Sec\n\r"); i2cWriteByte(0xd0,0,sec); showSec(sec); while(1){ sec = i2cReadByte(0xd0,0); if(sec != buf){ buf = sec; showSec(sec); } delay(1000); } }</pre>			
<p>1.3 จากข้อที่ 1.2 ให้นักศึกษาใช้โปรแกรม Proteus ISIS ทำการจำลองการทำงานของวงจรเปรียบเทียบกับการทำงานของวงจรจริง โดยการโหลดไฟล์นามสกุล .hex ลงบนตัวไมโครคอนโทรลเลอร์ AT89C51ED2 ด้วยโปรแกรม Flip 3.4.7 และป้อนแหล่งจ่ายให้บอร์ดทดลอง พร้อมกับเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์เข้ากับเครื่องคอมพิวเตอร์และใช้โปรแกรม HyperTerminal เพื่อทดสอบการทำงานของวงจร และบันทึกผลการทดลอง</p> <p>บันทึกผลการทดลอง</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>			

	ใบงานที่ 14	หน้าที่ 9
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

2. การอ่านค่าฐานเวลาจริงกับอุปกรณ์ DS1307 ด้วยระบบบัส I2C โดยใช้ไมโครคอนโทรลเลอร์ PIC16F887

2.1 ให้นักศึกษาใช้ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ ร่วมกับบอร์ดใช้งานไมโครคอนโทรลเลอร์ PIC16F887 เพื่อต่อวงจรในรูปที่ 2.1 เพื่อทดลองการอ่านค่าฐานเวลาจริงจาก DS1307




รูปที่ 2.1 รูปวงจรการอ่านค่าฐานเวลาจริงจาก DS1307 ด้วย PIC16F887


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


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


const unsigned char ascii[] = "0123456789ABCDEF";


unsigned char uart_getc(){
    unsigned char dat;
    while (!RCIF);
    dat = RCREG;
    return dat;
}
```

	ใบงานที่ 14	หน้าที่ 10
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI		
<pre>void uart_putc(unsigned char c){ //Send 1 Character while(!TRMT); TXREG = c; } void uart_puts (char *s) { //Send String while (*s) { uart_putc(*s); s++; } } void uart_init(unsigned int baud){ //Xtal = 20MHz unsigned int SpeedUart; SpeedUart = ((_XTAL_FREQ/ baud)/16) - 1; SPBRG = SpeedUart; SPBRGH = SpeedUart >> 8; SPEN = 1; //Serial Port Enable & Continuous Enable CREN = 1; TXEN = 1; //Tx Enable & High Speed mode BRGH = 1; } void I2C_Init(long i2c_clk_freq){ SSPCON = 0x28; // configure MSSP module to work in I2C mode SSPADD = (_XTAL_FREQ/(4 * i2c_clk_freq)) - 1; // set I2C clock frequency SSPSTAT = 0; } void I2C_Start(){ while ((SSPSTAT & 0x04) (SSPCON2 & 0x1F)); // wait for MSSP module to be free (not busy) SEN = 1; // initiate start condition }</pre>		

	ใบงานที่ 14	หน้าที่ 11
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI		
<pre>void I2C_Repeated_Start(){ while ((SSPSTAT & 0x04) (SSPCON2 & 0x1F)); // wait for MSSP module to be free (not busy) RSEN = 1; // initiate repeated start condition } void I2C_Stop(){ while ((SSPSTAT & 0x04) (SSPCON2 & 0x1F)); // wait for MSSP module to be free (not busy) PEN = 1; // initiate stop condition } void I2C_Write(unsigned char i2c_data){ while ((SSPSTAT & 0x04) (SSPCON2 & 0x1F)); // wait for MSSP module to be free (not busy) SSPBUF = i2c_data; // update buffer } unsigned char I2C_Read(unsigned char ack){ unsigned char _data; while ((SSPSTAT & 0x04) (SSPCON2 & 0x1F)); // wait for MSSP module to be free (not busy) RCEN = 1; while ((SSPSTAT & 0x04) (SSPCON2 & 0x1F)); // wait for MSSP module to be free (not busy) _data = SSPBUF; // read data from buffer while ((SSPSTAT & 0x04) (SSPCON2 & 0x1F)); // wait for MSSP module to be free (not busy) // send acknowledge pulse ? (depends on ack, if 1 send, otherwise don't send) if(ack) ACKDT = 0; else ACKDT = 1; ACKEN = 1; return _data; // return data read }</pre>		

	ใบงานที่ 14		หน้าที่ 12
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>void i2cWriteByte(unsigned char ctl,unsigned char addr,unsigned char dat){ I2C_Start(); // start I2C I2C_Write(ctl); // RTC chip address I2C_Write(addr); // send register address I2C_Write(dat); // send register address I2C_Stop(); // stop I2C } unsigned char i2cReadByte(unsigned char ctl,unsigned char addr){ unsigned char dat; I2C_Start(); // start I2C I2C_Write(ctl); // RTC chip address I2C_Write(addr); // send register address I2C_Repeated_Start(); // restart I2C I2C_Write(ctl+1); // initialize data read dat = I2C_Read(0); // read data from addr I2C_Stop(); // stop I2C return dat; } void showSec(unsigned dat){ uart_puts("SEC = "); uart_putc(ascii[dat/0x10]); uart_putc(ascii[dat%0x10]); uart_puts("\n\r"); }</pre>			

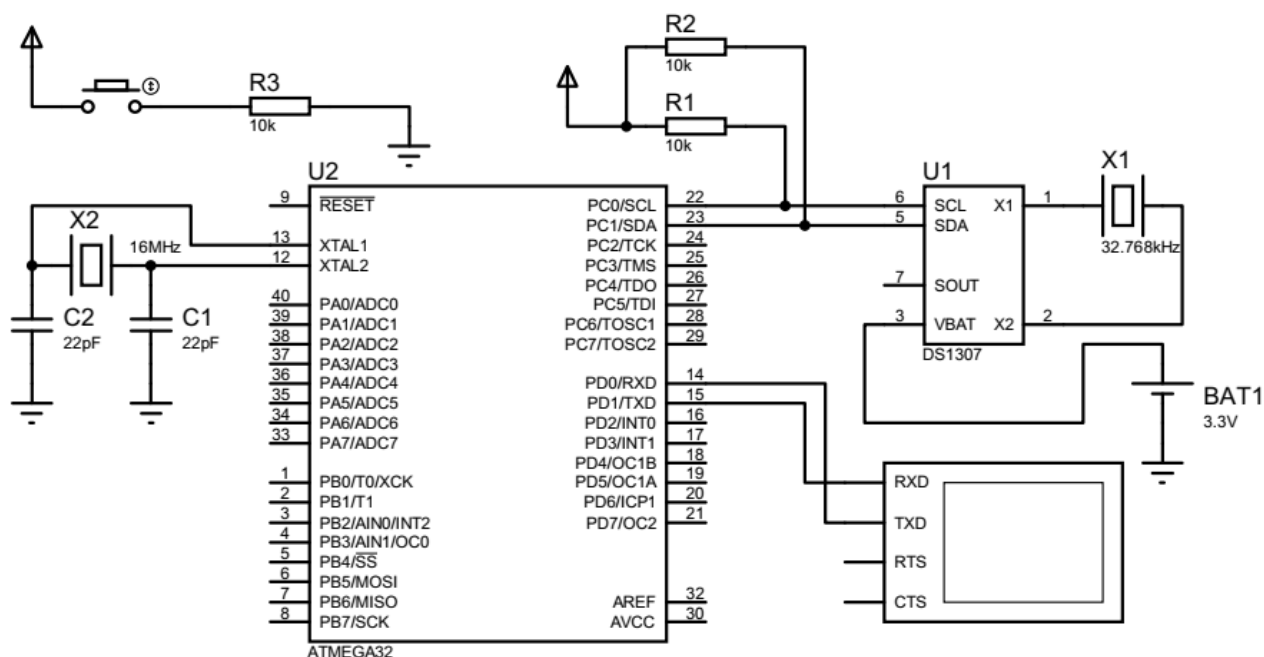
	ใบงานที่ 14		หน้าที่ 13
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>void main() { unsigned char sec=0,buf; uart_init(9600); I2C_Init(10000); // initialize I2C bus with clock frequency of 10kHz i2cWriteByte(0xd0,0,sec); showSec(sec); while(1){ sec = i2cReadByte(0xd0,0); if(buf != sec){ buf = sec; showSec(sec); } __delay_ms(100); } }</pre>			
<p>2.3 จากข้อที่ 2.2 ให้นักศึกษาใช้โปรแกรม Proteus ISIS ทำการจำลองการทำงานของวงจรเปรียบเทียบกับการทำงานของวงจรจริง โดยการโหลดไฟล์นามสกุล .hex ลงบนตัวไมโครคอนโทรลเลอร์ PIC16F887 ด้วยโปรแกรม MPLAB X ร่วมกับเครื่องโปรแกรม Pickit3 และป้อนแหล่งจ่ายให้บอร์ดทดลอง พร้อมกับเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์เข้ากับเครื่องคอมพิวเตอร์และใช้โปรแกรม HyperTerminal เพื่อทดสอบการทำงานของวงจรและบันทึกผลการทดลอง</p> <p>บันทึกผลการทดลอง</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>			

	ใบงานที่ 14		หน้าที่ 14
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

3. การอ่านค่าฐานเวลาจริงกับอุปกรณ์ DS1307 ด้วยระบบบัส I2C โดยใช้ไมโครคอนโทรลเลอร์ ATMEGA32

3.1 ให้นักศึกษาใช้ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ ร่วมกับบอร์ดใช้งานไมโครคอนโทรลเลอร์ ATMEGA32 เพื่อต่อวงจรในรูปที่ 3.1 เพื่อทดลองการอ่านค่าฐานเวลาจริงจาก DS1307



รูปที่ 3.1 รูปวงจรการอ่านค่าฐานเวลาจริงจาก DS1307 ด้วย ATMEGA32

3.2 จากวงจรในข้อที่ 3.1 ให้นักศึกษาเขียนโปรแกรมตามตัวอย่างต่อไปนี้โดยใช้โปรแกรม AVR Studio 6.2 ตามขั้นตอนในใบความรู้เรื่องภาษาซีกับไมโครคอนโทรลเลอร์ และทำการ Build โปรแกรมจนกว่าไม่เกิด Error เพื่อให้ได้ไฟล์นามสกุล .hex เพื่อนำไปใช้งานต่อไป

```
#include <avr/io.h>
```

```
#define F_CPU 16000000
```

```
#include <util/delay.h>
```

```
const uint8_t ascii[] = "0123456789ABCDEF";
```

```
unsigned char uart_getc(){
```


```
    unsigned char dat;
```


```
    while ( !(UCSRA & (1<<RXC)));
```


```
    dat = UDR;
```

```
    return dat;
```

```
}
```

	ใบงานที่ 14		หน้าที่ 15
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>void uart_putc(unsigned char c){ //Send 1 Character while(!(UCSRA & (1 << UDRE))); UDR = c; } void uart_puts (char *s){ //Send String while (*s) { uart_putc(*s); s++; } } void uart_init(unsigned int baud){ //Xtal = 16MHz uint32_t SpeedUart; SpeedUart = (uint32_t) (F_CPU/((uint32_t)16*baud)) - 1; UBRRL = SpeedUart; UBRRH = SpeedUart >> 8; UCSRB = (1<<RXEN) (1<<TXEN); UCSRC = (1<<URSEL) (3<<UCSZ0); } void i2c_init(uint32_t freq){ TWBR =(uint32_t)((((uint32_t)(F_CPU/freq) - 16) / 8); } void showTime(uint8_t sec){ uart_puts("SEC "); uart_putc(ascii[sec/0x10]); uart_putc(ascii[sec%0x10]); uart_puts("\n\r"); }</pre>			

	ใบงานที่ 14	หน้าที่ 16
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI		
<pre> void i2cWriteByte(char address, char reg, char data){ TWCR = (1<<TWINT) (1<<TWSTA) (1<<TWEN); // send a start bit on i2c bus while(!(TWCR & (1<<TWINT))); // wait for confirmation of transmit TWDR = address; // load address of i2c device TWCR = (1<<TWINT) (1<<TWEN); // transmit while(!(TWCR & (1<<TWINT))); // wait for confirmation of transmit TWDR = reg; TWCR = (1<<TWINT) (1<<TWEN); // transmit while(!(TWCR & (1<<TWINT))); // wait for confirmation of transmit TWDR = data; TWCR = (1<<TWINT) (1<<TWEN); // transmit while(!(TWCR & (1<<TWINT))); // wait for confirmation of transmit TWCR = (1<<TWINT) (1<<TWEN) (1<<TWSTO); // stop bit } unsigned char i2cReadByte(char address, char reg){ char read_data = 0; TWCR = (1<<TWINT) (1<<TWSTA) (1<<TWEN); // send a start bit on i2c bus while(!(TWCR & (1<<TWINT))); // wait for confirmation of transmit TWDR = address; // load address of i2c device TWCR = (1<<TWINT) (1<<TWEN); // transmit while(!(TWCR & (1<<TWINT))); // wait for confirmation of transmit TWDR = reg; // send register number to read from TWCR = (1<<TWINT) (1<<TWEN); // transmit while(!(TWCR & (1<<TWINT))); // wait for confirmation of transmit TWCR = (1<<TWINT) (1<<TWSTA) (1<<TWEN); // send repeated start bit while(!(TWCR & (1<<TWINT))); // wait for confirmation of transmit TWDR = address+1; // transmit address of i2c device with readbit set TWCR = (1<<TWINT) (1<<TWEA) (1<<TWEN); // clear transmit interrupt flag while(!(TWCR & (1<<TWINT))); // wait for confirmation of transmit TWCR = (1<<TWINT) (1<<TWEN); // transmit, nack (last byte request) while(!(TWCR & (1<<TWINT))); // wait for confirmation of transmit read_data = TWDR; // and grab the target data TWCR = (1<<TWINT) (1<<TWEN) (1<<TWSTO); // send a stop bit on i2c bus return read_data; } </pre>		

	ใบงานที่ 14		หน้าที่ 17
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

```
int main(){
    uint8_t buf=0,sec=0;
    uart_init(9600);
    i2c_init(100000);
    i2cWriteByte(0xd0,0,sec);
    showTime(sec);
    while(1){
        sec = i2cReadByte(0xd0,0);
        if(buf != sec){
            buf = sec;
            showTime(sec);
        }
        _delay_ms(100);
    }
    return 0;
}
```

3.3 จากข้อที่ 3.2 ให้นักศึกษาใช้โปรแกรม Proteus ISIS ทำการจำลองการทำงานของวงจรเปรียบเทียบกับการทำงานของวงจรจริง โดยการโหลดไฟล์นามสกุล .hex ลงบนตัวไมโครคอนโทรลเลอร์ ATMEGA32 ด้วยโปรแกรม AVR Studio 6.2 ร่วมกับเครื่องโปรแกรม AVR STK500V2 และป้อนแหล่งจ่ายให้บอร์ดทดลอง พร้อมกับเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์เข้ากับเครื่องคอมพิวเตอร์และใช้โปรแกรม HyperTerminal เพื่อทดสอบการทำงานของวงจร และบันทึกผลการทดลอง

บันทึกผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....


.....

.....

.....

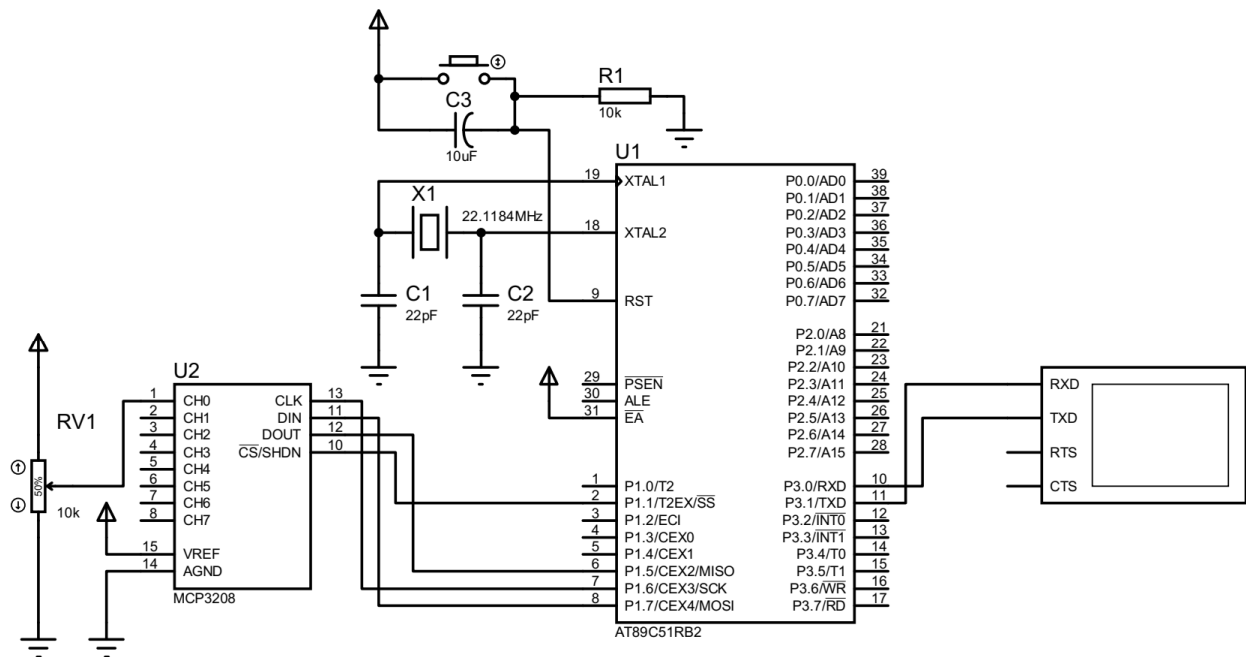
.....

.....

	ใบงานที่ 14		หน้าที่ 18
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI


4. การอ่านค่าข้อมูล ADC จากอุปกรณ์ MCP3208 ด้วยระบบบัส SPI โดยใช้ไมโครคอนโทรลเลอร์ AT89C51ED2
- 4.1 ให้นักศึกษาใช้ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ ร่วมกับบอร์ดใช้งานไมโครคอนโทรลเลอร์ AT89C51ED2 เพื่อต่อวงจรในรูปที่ 1.1 เพื่อทดลองการอ่านค่าข้อมูล ADC จากอุปกรณ์ MCP3208





รูปที่ 4.1 รูปวงจรการอ่านค่าข้อมูล ADC จากอุปกรณ์ MCP3208 ด้วย AT89C51ED2


- 4.2 จากวงจรในข้อที่ 4.1 ให้นักศึกษาเขียนโปรแกรมตามตัวอย่างต่อไปนี้โดยใช้โปรแกรม Keil uVision3 ตามขั้นตอนในใบความรู้เรื่องภาษาซีกับไมโครคอนโทรลเลอร์ และทำการ Build โปรแกรมจนกว่าไม่เกิด Error เพื่อให้ได้ไฟล์นามสกุล .hex เพื่อนำไปใช้งานต่อไป

```
#include <at89c51xd2.h>
#define XTAL 22118400
#define ss P1_1
#define mosi P1_7
#define miso P1_5
#define sck P1_6
unsigned char ascii[] = "0123456789ABCDEF";
void delay(int time){
    unsigned char x;
    for(;time>0;time--){
        for(x=200;x>0;x--);
    }
}
```

	ใบงานที่ 14		หน้าที่ 19
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>unsigned char uart_getc(){ while(RI == 0); RI = 0; return SBUF; } void uart_putc(unsigned char dat){ //Send 1 Character SBUF = dat; while(TI == 0); TI = 0; } void uart_puts(unsigned char *str){ //Send String while(*str != '\0') uart_putc(*str++); } void uart_init(int baud){ //Use XTAL 22.1184MHz SCON = 0x50; //RX Enable Data 8 bit Baud rate variable TMOD = (TMOD & 0x0f) 0x20; // Timer1 mode 8 Auto reload TH1 = 256 - ((XTAL/384)/baud); TL1 = TH1; TR1 = 1; } void spi_init(){ //Software SPI CPOL = 0 , CPHA = 0 ss = 0; mosi = 0; miso = 1; sck = 0; }</pre>			

	ใบงานที่ 14		หน้าที่ 20
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>unsigned char spi_write(char dat){ signed char i; unsigned char datO=0; for(i=7;i>=0;i--){ datO <<= 1; if(miso) datO = 1; if(dat & (1<<i)) mosi = 1; else mosi = 0; sck = 1; sck = 0; } return datO; } unsigned int spiReadMCP3208(unsigned char ch){ unsigned int dat; unsigned char cmd = 6; cmd = (ch >> 2); ch <<= 6; ss = 0; spi_write(cmd); dat = spi_write(ch); dat = (dat << 8) spi_write(0xff); ss = 1; return (dat &= 0x0fff); } void showADC(unsigned int dat){ uart_puts("ADC = "); uart_putc(ascii[dat/0x100]); uart_putc(ascii[(dat%0x100)/0x10]); uart_putc(ascii[dat%0x10]); uart_puts("\n\r"); }</pre>			

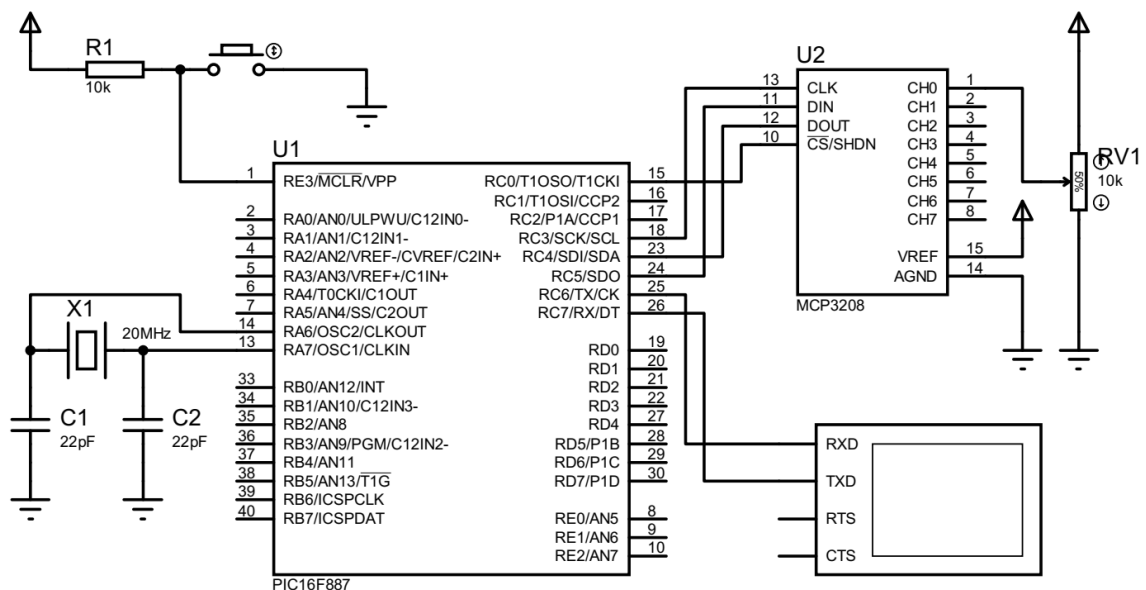
	ใบงานที่ 14		หน้าที่ 21
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>void main(){ unsigned int adc,buf; uart_init(9600); spi_init(); uart_puts("Test spi port\n\r"); adc = spiReadMCP3208(0); showADC(adc); buf = adc; while(1){ adc = spiReadMCP3208(0); if(adc != buf){ buf = adc; showADC(adc); } delay(1000); } }</pre>			
<p>4.3 จากข้อที่ 4.2 ให้นักศึกษาใช้โปรแกรม Proteus ISIS ทำการจำลองการทำงานของวงจรเปรียบเทียบกับการทำงานของวงจรจริง โดยการโหลดไฟล์นามสกุล .hex ลงบนตัวไมโครคอนโทรลเลอร์ AT89C51ED2 ด้วยโปรแกรม Flip 3.4.7 และป้อนแหล่งจ่ายให้บอร์ดทดลอง พร้อมกับเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์เข้ากับเครื่องคอมพิวเตอร์ และใช้โปรแกรม HyperTerminal เพื่อทดสอบการทำงานของวงจร และบันทึกผลการทดลอง</p> <p>บันทึกผลการทดลอง</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>			

	ใบงานที่ 14		หน้าที่ 22
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

5. การอ่านค่าข้อมูล ADC จากอุปกรณ์ MCP3208 ด้วยระบบบัส SPI โดยใช้ไมโครคอนโทรลเลอร์ PIC16F887


5.1 ให้นักศึกษาใช้ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ ร่วมกับบอร์ดใช้งานไมโครคอนโทรลเลอร์ PIC16F887 เพื่อต่อวงจรในรูปที่ 5.1 เพื่อทดลองการอ่านค่าข้อมูล ADC จากอุปกรณ์ MCP3208





รูปที่ 5.1 รูปวงจรการอ่านค่าข้อมูล ADC จากอุปกรณ์ MCP3208 ด้วย PIC16F887


5.2 จากวงจรในข้อที่ 5.1 ให้นักศึกษาเขียนโปรแกรมตามตัวอย่างต่อไปนี้โดยใช้โปรแกรม MPLAB X ตามขั้นตอนในใบความรู้เรื่องภาษาซีกับไมโครคอนโทรลเลอร์ และทำการ Build โปรแกรมจนกว่าไม่เกิด Error เพื่อให้ได้ไฟล์นามสกุล .hex เพื่อนำไปใช้งานต่อไป

```
#define _XTAL_FREQ 20000000
#include <xc.h>
#define SS_TRIS TRISC0
#define CS_HIGH() (RC0 = 1)
#define CS_LOW() (RC0 = 0)
const unsigned char ascii[] = "0123456789ABCDEF";
unsigned char uart_getc(){
    unsigned char dat;
    while (!RCIF);
    dat = RCREG;
    return dat;
}
```

	ใบงานที่ 14	หน้าที่ 23
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI		
<pre>void uart_putc(unsigned char c){ //Send 1 Character while(!TRMT); TXREG = c; } void uart_puts (char *s) { //Send String while (*s) { uart_putc(*s); s++; } } void uart_init(unsigned int baud){ //Xtal = 20MHz unsigned int SpeedUart; SpeedUart = ((_XTAL_FREQ/ baud)/16) - 1; SPBRG = SpeedUart; SPBRGH = SpeedUart >> 8; SPEN = 1; //Serial Port Enable & Continuous Enable CREN = 1; TXEN = 1; //Tx Enable & High Speed mode BRGH = 1; } void SPIInit(){ TRISCbits.TRISC3 = 0; // Setting Serial Clock as Output TRISCbits.TRISC4 = 1; // Master Input Slave Output (MISO) - SDI as input TRISCbits.TRISC5 = 0; // Master Output Slave Input (MOSI) - SDO as output SS_TRIS = 0; // Slave Select (SS) as output CS_HIGH(); // SS = HIGH //Master Mode, CPOL = 0, CPHA = 0 ,Freq SPI = XTAL/64 SSPSTAT = 0x00; SSPCON = 0b00100010; }</pre>		

	ใบงานที่ 14		หน้าที่ 24
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>unsigned char SPIWrite(unsigned char data){ SSPBUF = data; while(!SSPSTATbits.BF); // Wait for transmission complete return (SSPBUF); } unsigned int readADCEX(unsigned char ch){ unsigned char cmd=6; unsigned int dat; cmd = (ch >> 2); ch <<= 6; CS_LOW(); SPIWrite(cmd); dat = SPIWrite(ch); dat = (dat << 8) SPIWrite(0xFF); CS_HIGH(); return (dat & 0xffff); } void showADC(unsigned int dat){ uart_puts("ADC = "); uart_putc(ascii[dat/0x100]); uart_putc(ascii[(dat%0x100)/0x10]); uart_putc(ascii[dat%0x10]); uart_puts("\n\r"); }</pre>			

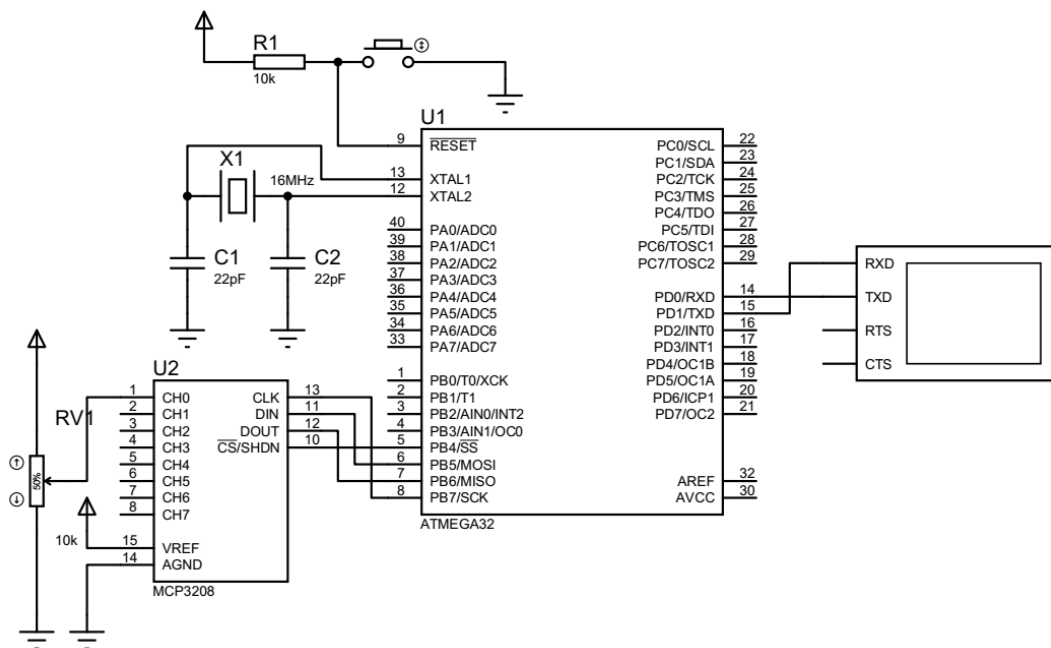
	ใบงานที่ 14		หน้าที่ 25
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>void main(){ unsigned int adc,buf; uart_init(9600); SPIInit(); uart_puts("Test spi port\n\r"); adc = readADCEX(0); showADC(adc); buf = adc; while(1){ adc = readADCEX(0); if(adc != buf){ buf = adc; showADC(adc); } __delay_ms(200); } }</pre>			
<p>5.3 จากข้อที่ 5.2 ให้นักศึกษาใช้โปรแกรม Proteus ISIS ทำการจำลองการทำงานของวงจรเปรียบเทียบกับการทำงานของวงจรจริง โดยการโหลดไฟล์นามสกุล .hex ลงบนตัวไมโครคอนโทรลเลอร์ PIC16F887 ด้วยโปรแกรม MPLAB X ร่วมกับเครื่องโปรแกรม Pickit3 และป้อนแหล่งจ่ายให้บอร์ดทดลอง พร้อมกับเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์เข้ากับเครื่องคอมพิวเตอร์และใช้โปรแกรม HyperTerminal เพื่อทดสอบการทำงานของวงจรและบันทึกผลการทดลอง</p> <p>บันทึกผลการทดลอง</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>			

	ใบงานที่ 14		หน้าที่ 26
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

6. การอ่านค่าข้อมูล ADC จากอุปกรณ์ MCP3208 ด้วยระบบบัส SPI โดยใช้ไมโครคอนโทรลเลอร์ ATMEGA32


6.1 ให้นักศึกษาใช้ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ ร่วมกับบอร์ดใช้งานไมโครคอนโทรลเลอร์ ATMEGA32 เพื่อต่อวงจรในรูปที่ 6.1 เพื่อทดลองการอ่านค่าข้อมูล ADC จากอุปกรณ์ MCP3208




รูปที่ 6.1 รูปวงจรการอ่านค่าข้อมูล ADC จากอุปกรณ์ MCP3208 ด้วย ATMEGA32


6.2 จากวงจรในข้อที่ 6.1 ให้นักศึกษาเขียนโปรแกรมตามตัวอย่างต่อไปนี้โดยใช้โปรแกรม AVR Studio 6.2 ตามขั้นตอนในใบความรู้เรื่องภาษาซีกับไมโครคอนโทรลเลอร์ และทำการ Build โปรแกรมจนกว่าไม่เกิด Error เพื่อให้ได้ไฟล์นามสกุล .hex เพื่อนำไปใช้งานต่อไป


```
#include <avr/io.h>
#define F_CPU 16000000
#include <util/delay.h>
#define SPI_PORT PORTB
#define SPI_DDR DDRB
#define SCK_P PB7
#define MISO_P PB6
#define MOSI_P PB5
#define SS_P PB4
#define CS_HIGH() (SPI_PORT |= (1<<SS_P))
#define CS_LOW() (SPI_PORT &= ~(1<<SS_P))
```

	ใบงานที่ 14	หน้าที่ 27
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI
<pre>const uint8_t ascii[] = "0123456789ABCDEF"; unsigned char uart_getc(){ unsigned char dat; while (!(UCSRA & (1<<RXC))); dat = UDR; return dat; } void uart_putc(unsigned char c){ //Send 1 Character while(!(UCSRA & (1 << UDRE))); UDR = c; } void uart_puts (char *s){ //Send String while (*s) { uart_putc(*s); s++; } } void uart_init(unsigned int baud){ //Xtal = 16MHz uint32_t SpeedUart; SpeedUart = (uint32_t) (F_CPU/((uint32_t)16*baud)) - 1; UBRRL = SpeedUart; UBRRH = SpeedUart >> 8; UCSRB = (1<<RXEN) (1<<TXEN); UCSRC = (1<<URSEL) (3<<UCSZ0); }</pre>

	ใบงานที่ 14		หน้าที่ 28
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>void SPIInit(){ SPI_DDR = (1<<MOSI_P) (1<<SCK_P) (1<<SS_P); SPI_PORT = (1<<SS_P); CS_HIGH(); //Mater Mode, CPOL = 0, CPHA = 0 ,Freq SPI = XTAL/16 = 1MHz SPCR =(1<<SPE) (1<<MSTR) (1<<SPR0); } uint8_t SPIWrite(uint8_t data){ SPDR = data; while(!(SPSR & (1<<SPIF))); // Wait for transmission complete return (SPDR); } uint16_t readADCEx(uint8_t ch){ uint8_t cmd=6; uint16_t dat; cmd = (ch >> 2); ch <<= 6; CS_LOW(); SPIWrite(cmd); dat = SPIWrite(ch); dat = (dat << 8) SPIWrite(0xFF); CS_HIGH(); return (dat & 0xffff); } void showADC(uint16_t dat){ uart_puts("ADC = "); uart_putc(ascii[dat/0x100]); uart_putc(ascii[(dat%0x100)/0x10]); uart_putc(ascii[dat%0x10]); uart_puts("\n\r"); }</pre>			

	ใบงานที่ 14		หน้าที่ 29
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI			
<pre>int main(){ uint16_t adc,buf; uart_init(9600); SPIInit(); uart_puts("Test spi port\n\r"); adc = readADCEX(0); showADC(adc); buf = adc; while(1){ adc = readADCEX(0); if(adc != buf){ buf = adc; showADC(adc); } _delay_ms(200); } return 0; }</pre>			
<p>6.3 จากข้อที่ 6.2 ให้นักศึกษาใช้โปรแกรม Proteus ISIS ทำการจำลองการทำงานของวงจรเปรียบเทียบกับการทำงานของวงจรจริง โดยการโหลดไฟล์นามสกุล .hex ลงบนตัวไมโครคอนโทรลเลอร์ ATMEGA32 ด้วยโปรแกรม AVR Studio 6.2 ร่วมกับเครื่องโปรแกรม AVR STK500V2 และป้อนแหล่งจ่ายให้บอร์ดทดลอง พร้อมกับเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์เข้ากับเครื่องคอมพิวเตอร์และใช้โปรแกรม HyperTerminal เพื่อทดสอบการทำงานของวงจร และบันทึกผลการทดลอง</p> <p>บันทึกผลการทดลอง</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>			

	ใบงานที่ 14	หน้าที่ 30
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI		
<p>แบบฝึกหัดท้ายบททดลอง</p> <ol style="list-style-type: none">1. ให้ออกแบบวงจรนาฬิกาแสดงผลด้วยอุปกรณ์ Character LCD โดยบรรทัดแรกให้แสดง วัน/เดือน/ปี ส่วนบรรทัดที่สองให้แสดง ชั่วโมง:นาฬิกา:วินาที โดยใช้อุปกรณ์ไอซีฐานเวลาจริง DS1307 ด้วย AT89C51ED22. จากข้อที่ 5 ให้นักศึกษาแก้ไขโปรแกรมให้สามารถแสดงผลค่าข้อมูลของ ADC ในรูปแบบเลขฐาน 10 และแรงดันไฟฟ้าที่วัดได้3. ให้ออกแบบวงจรนาฬิกาแสดงผลด้วยอุปกรณ์ 7-Segment โดยให้แสดงค่า วัน.เดือน.ปี และ ชั่วโมง.นาฬิกา.วินาที โดยใช้อุปกรณ์ไอซีฐานเวลาจริง DS1307 ด้วย ATMEGA32		

