
	ใบงานที่ 13		หน้าที่ 1
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
จุดประสงค์การเรียนรู้การสอน			
จุดประสงค์ทั่วไป			
1. เพื่อให้มีทักษะในการรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire ด้วยไมโครคอนโทรลเลอร์			
2. เพื่อให้มีทักษะในเขียนโปรแกรมไมโครคอนโทรลเลอร์เพื่อรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
3. เพื่อให้มีทักษะในการใช้งานโปรแกรม Proteus ในการจำลองการรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire ด้วยไมโครคอนโทรลเลอร์			
4. เพื่อให้มีเจตคติที่ดีต่อการเรียนรู้เรื่อง รับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
จุดประสงค์เชิงพฤติกรรม			
1. เตรียมเครื่องมือ วัสดุ และอุปกรณ์ได้ถูกต้อง			
2. ต่อบอร์ดวงจรเพื่อทดสอบรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire ด้วยไมโครคอนโทรลเลอร์ได้			
3. เขียนโปรแกรมควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire ด้วยไมโครคอนโทรลเลอร์ได้			
4. ใช้โปรแกรม Proteus เพื่อต่อบอร์ดวงจรทดสอบการทำงานการรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire ด้วยไมโครคอนโทรลเลอร์ได้			
5. ปฏิบัติงานได้อย่างถูกต้อง ประณีต รอบคอบ ปลอดภัย และเสร็จภายในเวลาที่กำหนด อย่างมีคุณธรรม จริยธรรม			
คุณธรรม จริยธรรม ค่านิยมและคุณลักษณะที่พึงประสงค์			
1. มีความซื่อสัตย์สุจริต			
2. มีความอดทนอดกลั้น			
3. มีความรักสามัคคีในการปฏิบัติงาน			
4. มีมนุษยสัมพันธ์ในการทำงาน			
5. มีการคำนึงถึงความปลอดภัยในการปฏิบัติงาน			
เครื่องมือ วัสดุ และอุปกรณ์การปฏิบัติงาน			
1. ใบความรู้เรื่องรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire		1 ชุด	
2. ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์		1 ชุด	
3. โปรแกรม HyperTerminal		1 ชุด	
4. มัลติมิเตอร์		1 ตัว	
5. สายต่อบอร์ดวงจร		40 เส้น	
6. คอมพิวเตอร์ที่ติดตั้งโปรแกรม Proteus 7.8SP2		1 เครื่อง	

	ใบงานที่ 13		หน้าที่ 2
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire

7. DS18S20 Temperature Sensor

2 ตัว

8. ตัวต้านทานค่า 10k Ω

2 ตัว

ข้อควรระวัง

ขณะปฏิบัติการทดลองห้ามหยอกล้อเล่นกัน และห้ามนำสายต่อลัดวงจรระหว่างขั้วไฟ Vcc 5V กับ GND

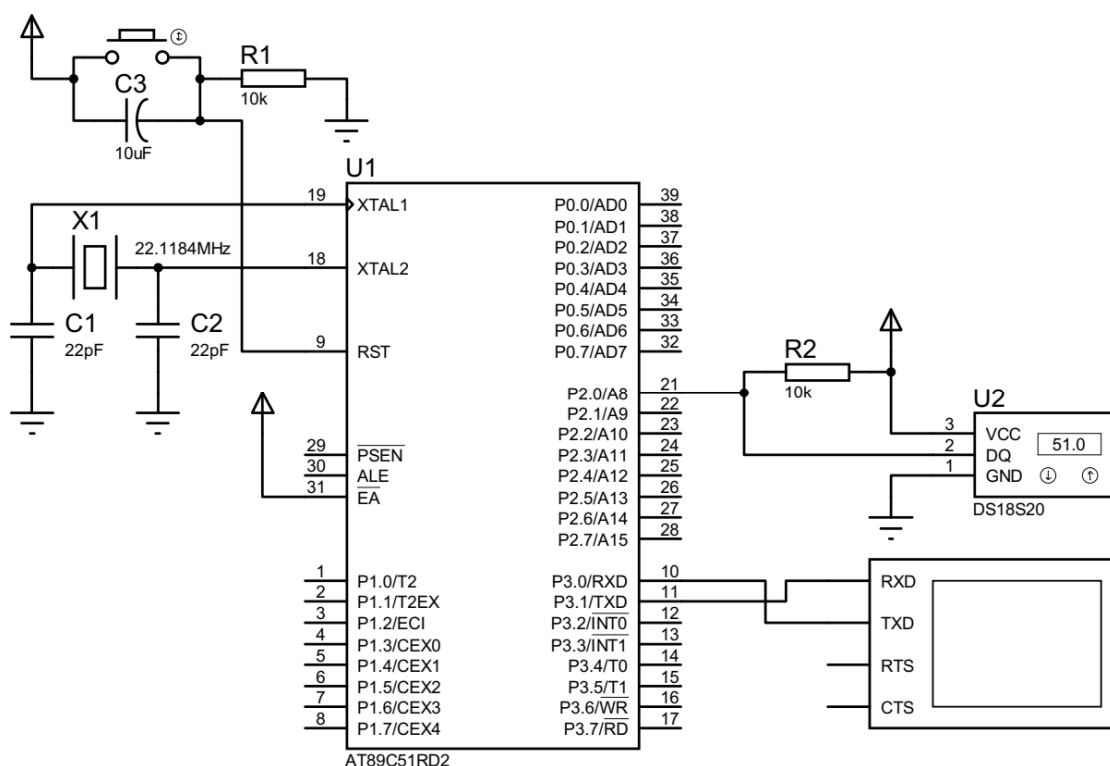
ข้อเสนอแนะ

ควรตรวจสอบความพร้อมของเครื่องมือและอุปกรณ์ทุกครั้งก่อนใช้งาน รวมถึงควรศึกษาและปฏิบัติตามขั้นตอนการปฏิบัติงานอย่างเคร่งครัด


ลำดับขั้นการปฏิบัติงาน (Step Operation)

1. การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire ด้วยไมโครคอนโทรลเลอร์ AT89C51ED2


1.1 ให้นักศึกษาใช้ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ ร่วมกับบอร์ดใช้งานไมโครคอนโทรลเลอร์ AT89C51ED2 เพื่อต่อวงจรในรูปที่ 1.1 เพื่อทดลองการสร้างเครื่องวัดอุณหภูมิด้วยเซนเซอร์ DS18S20 ที่มีพอร์ตการสื่อสารเป็นแบบ One-Wire และแสดงผลบนโปรแกรม Hyper Terminal ที่ติดตั้งบนเครื่องคอมพิวเตอร์




รูปที่ 1.1 รูปร่างวงจรวัดอุณหภูมิด้วย DS18S20 แสดงผลผ่านพอร์ต UART ด้วย AT89C51ED2

	ใบงานที่ 13	หน้าที่ 3
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	


ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire
<p>1.2 จากวงจรในข้อที่ 1.1 ให้นักศึกษาเขียนโปรแกรมตามตัวอย่างต่อไปนี้โดยใช้โปรแกรม Keil uVision3 ตามขั้นตอนในใบความรู้เรื่องภาษาซีกับไมโครคอนโทรลเลอร์ และทำการ Build โปรแกรมจนกว่าไม่เกิด Error เพื่อให้ได้ไฟล์นามสกุล .hex เพื่อนำไปใช้งานต่อไป</p> <pre>//Xtal 22.1184MHz #include <at89c51xd2.h> #include <intrins.h> #define XTAL 22118400 sbit onewire = P2^0; // Bit data 1-wire bus unsigned char dot; unsigned int temp; unsigned char ascii[] = "0123456789ABCDEF"; void delay(int time){ unsigned char x; for(;time>0;time--) for(x=200;x>0;x--); } void delayOneWire(unsigned int usecond){ while(usecond--); } unsigned char ow_reset(){ unsigned char presence; unsigned int i; onewire = 0; // pull DQ line low delayOneWire(480); // leave it low for 480us onewire = 1; // allow line to return high delayOneWire(10); // wait for presence i = 60; do{ presence = onewire; // get presence signal i--; // wait for end of timeslot }while((i>0)&&(presence>0)); i = 120 - i; delayOneWire(i); // presence signal returned 0=presence, 1 = no part return presence; }</pre>


	ใบงานที่ 13		หน้าที่ 4
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
<pre>bit read_bit (void){ unsigned char i; bit dat; // Bit for keep data onewire = 0; // Clear bit 1-wire _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay onewire = 1; // Set bit 1-wire _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay dat = onewire; // Read data bit keep in dat for(i=0;i<16;i++) _nop_();// Delay 64 microsec return(dat); // Return dat } unsigned char ReadByte(){ unsigned char i,j,dat; // For counter and keep data dat = 0; // Clear data for(i=0;i<8;i++){ // For loop Read data 1 byte(8 time) j = read_bit(); // Keep data bit to j dat = (j<<7) (dat>>1); // Shift left j 7 time OR } return(dat); // Return dat }</pre>			

	ใบงานที่ 13		หน้าที่ 5
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire	
<pre>void WriteByte(unsigned char com){ unsigned char j,i; // For counter bit send; // Bit send data 1-wire for(j=0;j<8;j++){ // For loop Write data 1 byte(8 time) send = com & 0x01; // LSB bit keep to send com = com>>1; // Shift Right 1 time if(send) { // Check send = 1? onewire = 0; // Clear bit 1-wire _nop_(); // Delay delay 4 microsec _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay onewire = 1; // Set bit 1-wire for(i=0;i<16;i++) _nop_(); //delay 64 microsec }else{ onewire = 0; // Clear bit 1-wire for(i=0;i<16;i++) _nop_(); // delay 64 microsec onewire = 1; // Set bit 1-wire _nop_(); // Delay delay 4 microsec _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay _nop_(); // Delay } } }</pre>	

	ใบงานที่ 13		หน้าที่ 6
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
<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 ReadTemp_DS18B20() { unsigned char get[10],i; int temp_lsb, temp_msb; while(ow_reset()); WriteByte(0xCC); // Write command skip Rom DS1820 WriteByte(0x44); // Write command Convert Temperature</pre>			

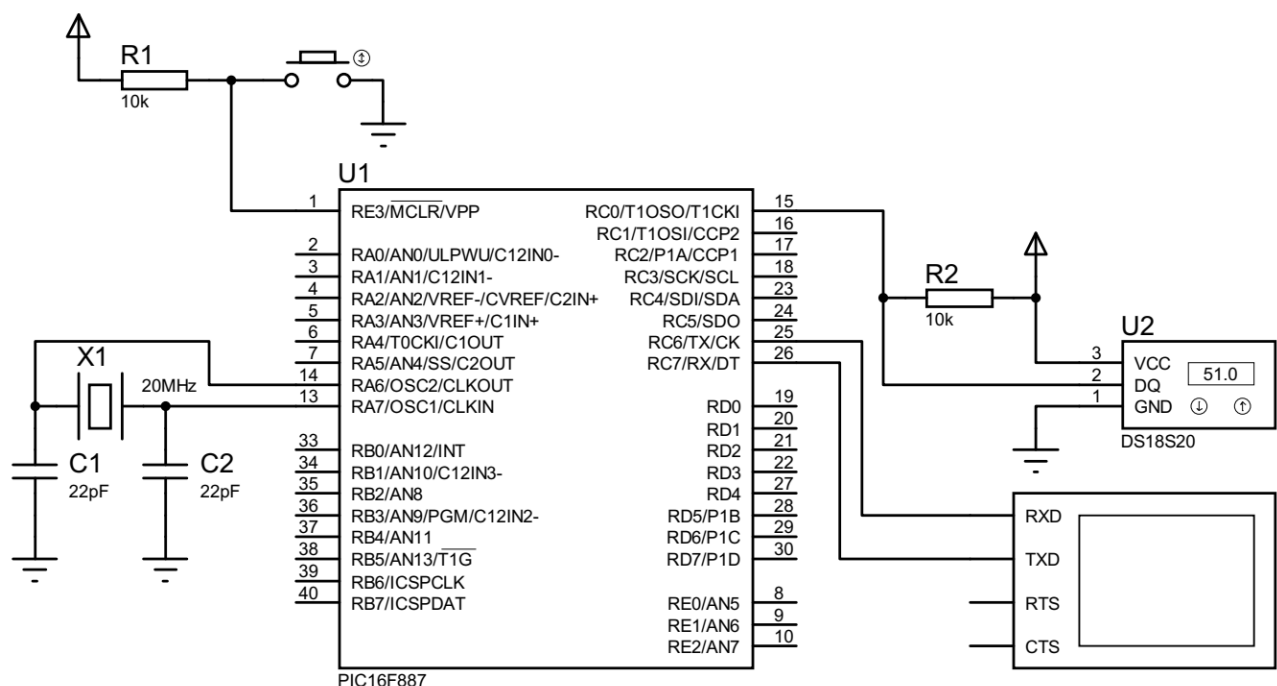
	ใบงานที่ 13		หน้าที่ 7
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
<pre>while(ow_reset()); WriteByte(0xCC); // Write command skip Rom DS1820 WriteByte(0xBE); // Write command Read Temperature for(i=0;i<9;i++) get[i] = ReadByte(); // Read Temperature and keep to temp while(ow_reset()); temp_msb = get[1]; temp_lsb = get[0]; dot = 0; if((temp_lsb&0x01)==1) dot = 5; if(temp_msb == 0x00) temp_lsb = temp_lsb >> 1; else{ temp_lsb = (~temp_lsb)+1; temp_lsb = temp_lsb >> 1; temp_lsb = (-1) * temp_lsb; } temp = temp_lsb; } void main(){ uart_init(9600); uart_puts("Test UART 9600bps & OneWire DS18B20\n\r"); uart_puts("Please any key to Start\n\r"); uart_getc(); uart_puts("\n\r"); while(1){ ReadTemp_DS18B20(); uart_putc(ascii[temp/100]); uart_putc(ascii[(temp%100)/10]); uart_putc(ascii[temp%10]); uart_putc('.'); uart_putc(ascii[dot]); uart_puts("\n\r"); delay(1000); } }</pre>			

	ใบงานที่ 13		หน้าที่ 9
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire

2. การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire ด้วยไมโครคอนโทรลเลอร์ PIC16F887


2.1 ให้นักศึกษาใช้ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ ร่วมกับบอร์ดใช้งานไมโครคอนโทรลเลอร์ PIC16F887 เพื่อต่อวงจรในรูปที่ 2.1 เพื่อทดลองการสร้างเครื่องวัดอุณหภูมิด้วยเซนเซอร์ DS18S20 ที่มีพอร์ตการสื่อสารเป็นแบบ One-Wire และแสดงผลบนโปรแกรม Hyper Terminal ที่ติดตั้งบนเครื่องคอมพิวเตอร์





รูปที่ 2.1 รูปวงจรวัดอุณหภูมิด้วย DS18S20 แสดงผลผ่านพอร์ต UART ด้วย PIC16F887

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


```
#define _XTAL_FREQ 20000000
#include <xc.h>
#define onewire RC0 // Bit data 1-wire bus
#define TRISC1wire TRISC0
unsigned char dot;
unsigned int temp;
const unsigned char ascii[] = "0123456789ABCDEF";
```


	ใบงานที่ 13		หน้าที่ 10
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
<pre>void delay_us(int us){ for(;us>0;us--) __delay_us(1); } char ow_reset(){ char DQ; int i; TRISC1wire = 0; onewire = 0; // Clear bit 1-wire __delay_us(480); // Delay 480 microsec onewire = 1; // Set bit 1-wire __delay_us(16); // Delay 16 microsec TRISC1wire = 1; i = 240; do{ DQ = onewire; // get presence signal i--; __delay_us(1); }while((i>0)&&(DQ>0)); TRISC1wire = 0; onewire = 1; i = 480 - i; delay_us(i); // wait for end of timeslot return DQ; // returned 0 = presence, 1 = no part } char read_bit(void){ TRISC1wire = 0; onewire = 0; // pull onewire low to start timeslot __delay_us(10); onewire = 1; // then return high __delay_us(5); TRISC1wire = 1; return(owire); // return value of DQ line }</pre>			

	ใบงานที่ 13		หน้าที่ 11
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
<pre>void write_bit(unsigned char DQ){ TRISC1wire = 0; onewire = 0; // pull DQ low to start timeslot __delay_us(10); if(DQ==1) onewire = 1; // return DQ high if write 1 __delay_us(65); // hold value for remainder of timeslot onewire = 1; __delay_us(15); } unsigned char ReadByte(void){ unsigned char i; unsigned char value = 0; for(i=0;i<8;i++){ if(read_bit()) value =0x01<<i; // reads byte in, one byte at a time and then shifts it left __delay_us(60); // wait for rest of timeslot TRISC1wire = 0; onewire = 1; __delay_us(15); } return value; } void WriteByte(char val){ unsigned char i; unsigned char temp; TRISC1wire = 0; for (i=0; i<8; i++){ // writes byte, one bit at a time temp = val>>i; // shifts val right 'i' spaces temp &= 0x01; // copy that bit to temp write_bit(temp); // write bit in temp into } }</pre>			

	ใบงานที่ 13		หน้าที่ 12
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
<pre>unsigned char uart_getc(){ unsigned char dat; while (!RCIF); dat = RCREG; return dat; } 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; }</pre>			

	ใบงานที่ 13		หน้าที่ 13
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
<pre>void ReadTemp_DS18B20() { unsigned char get[10],k; int temp_lsb, temp_msb; while(ow_reset()); WriteByte(0xCC); // Skip ROM WriteByte(0x44); // Start Conversion while(ow_reset()); WriteByte(0xCC); // Skip ROM WriteByte(0xBE); // Read Scratch Pad for(k=0; k<9; k++) get[k] = ReadByte(); temp_msb = get[1]; temp_lsb = get[0]; ow_reset(); dot = 0; if((temp_lsb&0x01)==1) dot = 5; if(temp_msb == 0x00) temp_lsb = temp_lsb >> 1; else{ temp_lsb = (~temp_lsb)+1; temp_lsb = temp_lsb >> 1; temp_lsb = (-1) * temp_lsb; } temp = temp_lsb; }</pre>			

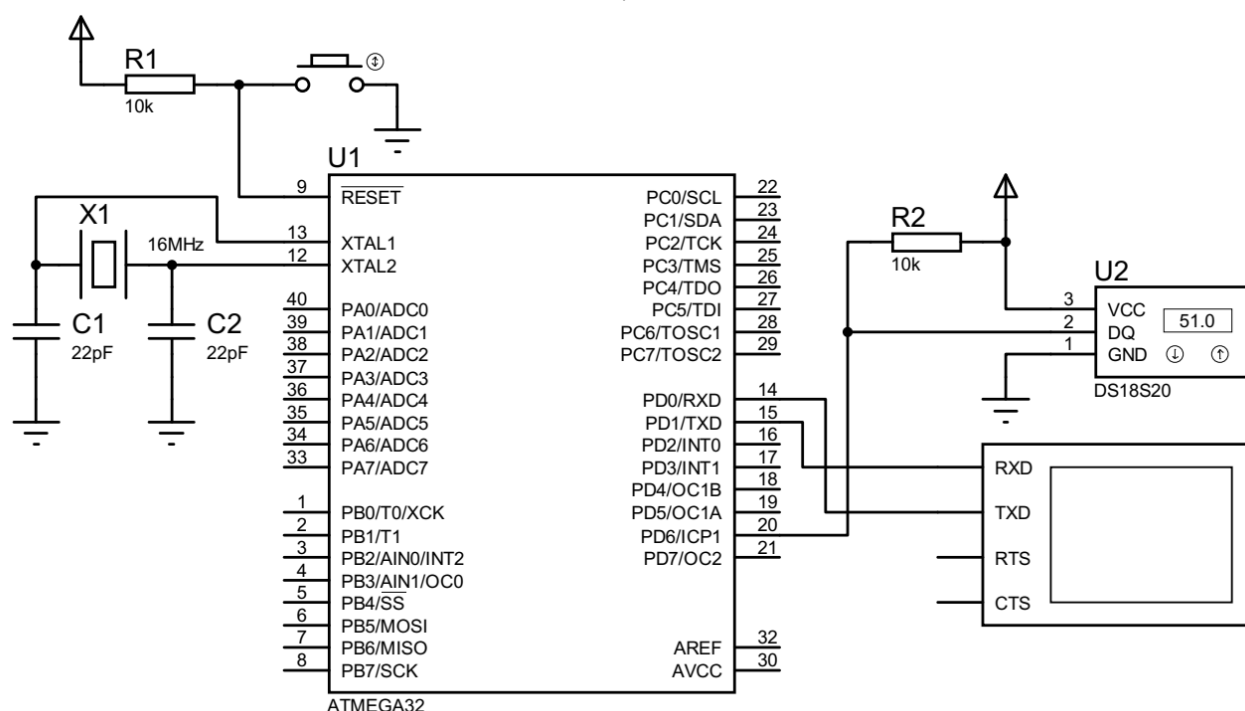
	ใบงานที่ 13		หน้าที่ 14
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
<pre>int main(){ uart_init(9600); uart_puts("Test UART 9600bps & OneWire DS18B20\n\r"); uart_puts("Please any key to Start\n\r"); uart_getc(); uart_puts("\n\r"); while(1){ ReadTemp_DS18B20(); uart_putc(ascii[temp/100]); uart_putc(ascii[(temp%100)/10]); uart_putc(ascii[temp%10]); uart_putc('.'); uart_putc(ascii[dot]); uart_puts("\n\r"); __delay_ms(1000); } return 0; }</pre>			
<p>2.3 จากข้อที่ 2.2 ให้นักศึกษาใช้โปรแกรม Proteus ISIS ทำการจำลองการทำงานของวงจรเปรียบเทียบกับการทำงานของวงจรจริง โดยการโหลดไฟล์นามสกุล .hex ลงบนตัวไมโครคอนโทรลเลอร์ PIC16F887 ด้วยโปรแกรม MPLAB X ร่วมกับเครื่องโปรแกรม Pickit3 และป้อนแหล่งจ่ายให้บอร์ดทดลอง หลังจากนั้นให้ใช้โปรแกรม Hyper Terminal เชื่อมต่อกับบอร์ดทดลอง แล้วทำการตั้งค่า comport ด้วยอัตราบอดเรต 9600bps และทำการกดปุ่ม คีย์บอร์ดใด ๆ ก็ได้ 1 ครั้งแล้วให้สังเกตผลการทำงานของบอร์ดทดลอง เพื่อบันทึกผลการทดลอง</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 13		หน้าที่ 15
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		

ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire

3. การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire ด้วยไมโครคอนโทรลเลอร์ ATMEGA32

3.1 ให้นักศึกษาใช้ชุดทดลองวงจรดิจิทัลและไมโครคอนโทรลเลอร์ ร่วมกับบอร์ดใช้งานไมโครคอนโทรลเลอร์ ATMEGA32 เพื่อต่อวงจรในรูปที่ 3.1 เพื่อทดลองการสร้างเครื่องวัดอุณหภูมิด้วยเซนเซอร์ DS18S20 ที่มีพอร์ตการสื่อสารเป็นแบบ One-Wire และแสดงผลบนโปรแกรม Hyper Terminal ที่ติดตั้งบนเครื่องคอมพิวเตอร์



รูปที่ 3.1 รูปวงจรวัดอุณหภูมิด้วย DS18S20 แสดงผลผ่านพอร์ต UART 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 dot;
```

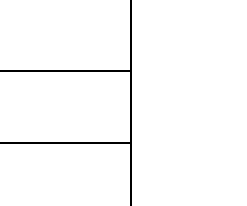
```
unsigned int temp;
```


	ใบงานที่ 13	หน้าที่ 16
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire		
<pre> #define BIT_DQ 6 // 1-Wire Interface (DS18B20 Temp. Sensor) #define SET_IN_DQ (DDRD &= ~(1<<BIT_DQ)) #define SET_OUT_DQ (DDRD = (1<<BIT_DQ)) #define OUT_HIGH_DQ (PORTD = (1<<BIT_DQ)) #define OUT_LOW_DQ (PORTD &= ~(1<<BIT_DQ)) #define IN_DQ (PIND & (1<<BIT_DQ)) void delay_us(unsigned int time_us){ for (;time_us>0; time_us--) _delay_us(1); } unsigned char ow_reset(void){ unsigned char presence; unsigned int i; SET_OUT_DQ; // Set output port delay_us(1); OUT_LOW_DQ; // pull DQ line low delay_us(480); // leave it low for 480us OUT_HIGH_DQ; // allow line to return high delay_us(15); // wait for presence SET_IN_DQ; // Set input port i = 120; do{ presence = IN_DQ; // get presence signal _delay_us(1); // wait for end of timeslot i--; }while((i > 0)&&(presence > 0)); i = 480 - i; delay_us(i); return(presence); // presence signal returned 0=presence, 1 = no part } </pre>		

	ใบงานที่ 13		หน้าที่ 17
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
<pre>unsigned char read_bit(void){ SET_OUT_DQ; // Set output port OUT_LOW_DQ; // pull DQ low to start timeslot delay_us(10); // delay_us 15us from start of timeslot OUT_HIGH_DQ; // then return high SET_IN_DQ; // Set input port delay_us(5); return(IN_DQ); } void write_bit(char bitval){ OUT_LOW_DQ; // pull DQ low to start timeslot delay_us(10); if(bitval == 1) // return DQ high if wirte 1 OUT_HIGH_DQ; delay_us(65); // hold value for remainder of timeslot (100us) OUT_HIGH_DQ; delay_us(15); } unsigned char ReadByte(void){ unsigned char i; unsigned char value = 0; for(i=0; i<8; i++) { if(read_bit()) // reads byte in,one byte at a time value = 0x01<<i; // and then shifts it left delay_us(60); // wait for rest for timeslot SET_OUT_DQ; OUT_HIGH_DQ; delay_us(15); } return(value); }</pre>			

	ใบงานที่ 13		หน้าที่ 18
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
<pre>void WriteByte(char val){ unsigned char i; unsigned char temp; SET_OUT_DQ; // Set output port for(i=0; i<8; i++) { // writes byte, one bit at a time temp = val>>i; // shifts val right 'i' spaces temp &= 0x01; // copy that bit to temp write_bit(temp); // write bit in temp into } } 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++; } }</pre>			

	ใบงานที่ 13		หน้าที่ 19
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004		หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
<pre>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 ReadTemp_DS18B20(){ unsigned char get[10],k; int temp_lsb, temp_msb; while(ow_reset()); WriteByte(0xCC); // Skip ROM WriteByte(0x44); // Start Conversion while(ow_reset()); WriteByte(0xCC); // Skip ROM WriteByte(0xBE); // Read Scratch Pad for(k=0; k<9; k++) get[k] = ReadByte(); temp_msb = get[1]; temp_lsb = get[0]; ow_reset(); dot = 0; if((temp_lsb&0x01)==1) dot = 5; if(temp_msb == 0x00) temp_lsb = temp_lsb >> 1; else{ temp_lsb = (~temp_lsb)+1; temp_lsb = temp_lsb >> 1; temp_lsb = (-1) * temp_lsb; } temp = temp_lsb; }</pre>			

	ใบงานที่ 13	หน้าที่ 20
	ชื่อวิชา ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ	
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire		
<pre>int main(){ uart_init(9600); uart_puts("Test UART 9600bps & OneWire DS18B20\n\r"); uart_puts("Please any key to Start\n\r"); uart_getc(); uart_puts("\n\r"); while(1){ ReadTemp_DS18B20(); uart_putc(ascii[temp/100]); uart_putc(ascii[(temp%100)/10]); uart_putc(ascii[temp%10]); uart_putc('.'); uart_putc(ascii[dot]); uart_puts("\n\r"); _delay_ms(1000); } return 0; }</pre>		
<p>3.3 จากข้อที่ 3.2 ให้นักศึกษาใช้โปรแกรม Proteus ISIS ทำการจำลองการทำงานของวงจรเปรียบเทียบกับการทำงานของวงจรจริง โดยการโหลดไฟล์นามสกุล .hex ลงบนตัวไมโครคอนโทรลเลอร์ ATMEGA32 ด้วยโปรแกรม AVR Studio 6.2 ร่วมกับเครื่องโปรแกรม AVR STK500V2 และป้อนแหล่งจ่ายให้บอร์ดทดลอง หลังจากนั้นให้ใช้โปรแกรม Hyper Terminal เชื่อมต่อกับบอร์ดทดลอง แล้วทำการตั้งค่า comport ด้วยอัตราบอดเรต 9600bps และทำการกดปุ่มคีย์บอร์ดใด ๆ ก็ได้ 1 ครั้งแล้วให้สังเกตผลการทำงานของบอร์ดทดลอง เพื่อบันทึกผลการทดลอง</p> <div style="border: 1px solid black; height: 400px; width: 100%;"></div>		

	ใบงานที่ 13		หน้าที่ 21
	ชื่อวิชา	ดิจิทัลและไมโครคอนโทรลเลอร์ รหัสวิชา 30127-2004	หน่วยที่ 8
	ชื่อหน่วย การรับส่งข้อมูลอนุกรมแบบต่าง ๆ		
ชื่อเรื่อง การรับส่งข้อมูลอนุกรมในรูปแบบ UART และ One-Wire			
แบบฝึกหัดท้ายการทดลอง			
<p>1. ให้ออกแบบวงจรอ่านค่าอุณหภูมิด้วยเซ็นเซอร์ DS18S20 แล้วทำการแสดงผลบนอุปกรณ์ 7-segment และพอร์ต UART โดยใช้บอดเรต 19200bps ด้วยไมโครคอนโทรลเลอร์ AT89C51ED2 พร้อมทั้งเขียนโปรแกรมเพื่อทดสอบการทำงานของวงจร</p> <p>2. ให้ออกแบบวงจรอ่านค่าอุณหภูมิด้วยเซ็นเซอร์ DS18S20 แล้วทำการแสดงผลบนอุปกรณ์ 7-segment และพอร์ต UART โดยใช้บอดเรต 19200bps ด้วยไมโครคอนโทรลเลอร์ PIC16F887 พร้อมทั้งเขียนโปรแกรมเพื่อทดสอบการทำงานของวงจร</p> <p>3. ให้ออกแบบวงจรอ่านค่าอุณหภูมิด้วยเซ็นเซอร์ DS18S20 แล้วทำการแสดงผลบนอุปกรณ์ 7-segment และพอร์ต UART โดยใช้บอดเรต 19200bps ด้วยไมโครคอนโทรลเลอร์ ATMEGA32 พร้อมทั้งเขียนโปรแกรมเพื่อทดสอบการทำงานของวงจร</p>			

[illegible]