

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

รหัสวิชา 30127-2004 (2-3-3) ดิจิทัลและไมโครคอนโทรลเลอร์

Digital And Microcontroller

1

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

1. รูปแบบข้อมูลของการสื่อสารแบบ I2C
2. รูปแบบข้อมูลของการสื่อสารแบบ SPI
3. รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต I2C ของไมโครคอนโทรลเลอร์
 - 3.1 รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต I2C ของ PIC16F887
 - 3.2 รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต I2C ของ ATMEGA32
4. รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต SPI ของไมโครคอนโทรลเลอร์
 - 4.1 รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต SPI ของ AT89C51ED2
 - 4.2 รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต SPI ของ PIC16F887
 - 4.3 รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต SPI ของ ATMEGA32

Digital And Microcontroller

2

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

4. การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ I2C ของไมโครคอนโทรลเลอร์
 - 4.1 การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ I2C ของ AT89C51ED2
 - 4.2 การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ I2C ของ PIC16F887
 - 4.3 การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ I2C ของ ATMEGA32
5. การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ SPI ของไมโครคอนโทรลเลอร์
 - 5.1 การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ SPI ของ AT89C51ED2
 - 5.2 การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ SPI ของ PIC16F887
 - 5.3 การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ SPI ของ ATMEGA32

Digital And Microcontroller

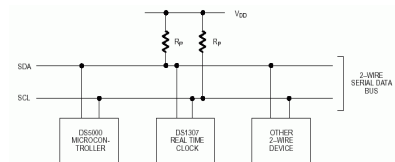
3

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

1. รูปแบบข้อมูลของการสื่อสารแบบ I2C

I2C Bus (I²C) ย่อมาจาก Inter Integrate Circuit Bus (IIC) นิยมเรียกสั้นๆ ว่า I²C (ไอ-แอสคิว-ซี-บัส) เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) แบบ Half Duplex ถูกพัฒนาโดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ Serial data (SDA) และสาย Serial clock (SCL)

1.1 ลักษณะการเชื่อมต่ออุปกรณ์แบบ I2C Bus

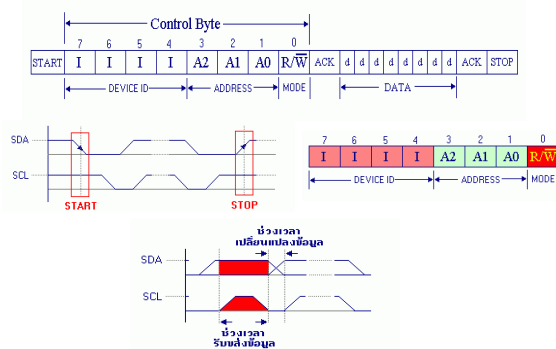


Digital And Microcontroller

4

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

1.2 การเขียน-อ่านข้อมูลกับอุปกรณ์แบบ I2C Bus



Digital And Microcontroller

5

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

2. รูปแบบข้อมูลของการสื่อสารแบบ SPI

SPI ย่อมาจาก Serial Peripheral Interface คือรูปแบบการสื่อสารข้อมูลแบบอนุกรมแบบซิงโครนัส พัฒนาขึ้นมาโดยโมโตโรล่าเพื่อใช้ในการสื่อสารระยะใกล้ อาศัยสัญญาณนาฬิกาเป็นตัวกำหนดจังหวะการรับส่งข้อมูล สามารถส่งข้อมูลไปยังปลายทางและรับข้อมูลจากปลายทางกลับมาในครั้งเดียวกัน (Full Duplex) การสื่อสารอนุกรมแบบ SPI จะแบ่งอุปกรณ์ออกเป็น 2 ฝั่ง คือ Master เป็นตัวควบคุมการรับส่งข้อมูล และ Slave เป็นอุปกรณ์ที่รอรับคำสั่งจาก Master ในบัสการสื่อสารแบบอนุกรมแบบ SPI สามารถมี Slave ได้มากกว่า 1 ตัว

Digital And Microcontroller

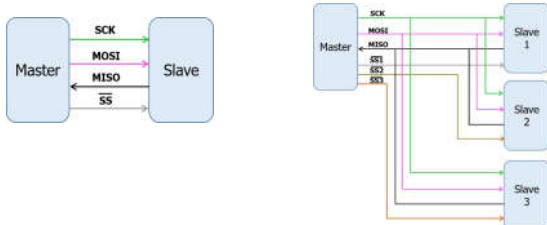
6

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

2.1 ขาสัญญาณและลักษณะการเชื่อมต่ออุปกรณ์ด้วยระบบบัส SPI

การสื่อสารอนุกรมแบบ SPI ใช้สายสัญญาณทั้งหมดจำนวน 4 เส้น มีรายละเอียดดังนี้

- SCK (Clock Data) ใช้สำหรับส่งสัญญาณนาฬิกาจาก Master ไปยัง Slave Shared
- MISO (Master In Slave Out) ใช้สำหรับรับข้อมูลจาก Slave Shared
- MOSI (Master Out Slave In) ใช้สำหรับส่งข้อมูลจาก Master ไปยัง Slave Shared
- SS/CS (Slave Select/Chip Select) ใช้สำหรับเลือก Slave ที่ต้องการใช้งาน Not Shared



Digital And Microcontroller

7

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

2.2 รูปแบบสัญญาณในระบบบัส SPI

รูปแบบสัญญาณ SPI มี 4 รูปแบบ แตกต่างกันที่ขั้วขั้วสัญญาณนาฬิกา (Clock Polarity) และ เฟส (Phase) ดังนี้

2.1 เมื่อ CPHA=0 และ CPOL=0 สัญญาณนาฬิกา (Clock) ในสถานะปกติจะเป็น Low และ จะรับ-ส่งข้อมูลที่ขอบขาขึ้นของสัญญาณนาฬิกา (Rising Edge Clock)

2.2 เมื่อ CPHA=0 และ CPOL=1 สัญญาณนาฬิกา (Clock) ในสถานะปกติจะเป็น High และ จะรับ-ส่งข้อมูลที่ขอบขาลงของสัญญาณนาฬิกา (Falling Edge Clock)

2.3 เมื่อ CPHA=1 และ CPOL=0 สัญญาณนาฬิกา (Clock) ในสถานะปกติจะเป็น Low และ จะรับ-ส่งข้อมูลที่ขอบขาลงของสัญญาณนาฬิกา (Falling Edge Clock)

2.4 เมื่อ CPHA=1 และ CPOL=1 สัญญาณนาฬิกา (Clock) ในสถานะปกติจะเป็น High และ จะรับ-ส่งข้อมูลที่ขอบขาขึ้นของสัญญาณนาฬิกา (Rising Edge Clock)

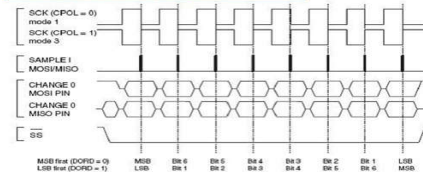
Digital And Microcontroller

8

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

Mode	Clock Polarity (CPOL)	Clock Phase (CPHA)	Output Edge	Data Capture
SPI_MODE0	0	0	Falling	Rising
SPI_MODE1	0	1	Rising	Falling
SPI_MODE2	1	0	Rising	Falling
SPI_MODE3	1	1	Falling	Rising

SPI Transfer with CPHA = 1 (Setup First)

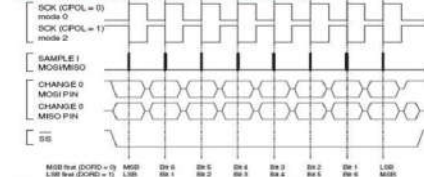


Digital And Microcontroller

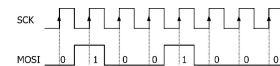
9

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

SPI Transfer with CPHA = 0 (Sampling First) (Popular)



2.3 การใช้งานการสื่อสารอนุกรมแบบ SPI

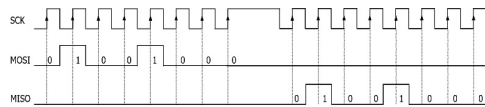


รูปตัวอย่างสัญญาณการส่งอักษร 'H' แบบ SPI

Digital And Microcontroller

10

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI



รูปตัวอย่างการส่งอักษร 'H' และรับข้อมูลอักษร 'H' ด้วยการสื่อสารอนุกรมแบบ SPI

3. รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต I2C ของไมโครคอนโทรลเลอร์

3.1 รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต I2C ของ PIC16F887 (RC3, RC4)

- SSPSTAT คือรีจิสเตอร์ที่ใช้แสดงสถานะและกำหนดลักษณะการทำงานของพอร์ต I2C
- SSPCON คือรีจิสเตอร์ที่ใช้ควบคุมการทำงานของพอร์ต I2C
- SSPCON2 คือรีจิสเตอร์ที่ใช้ควบคุมการทำงานของพอร์ต I2C
- SSPBUFF คือรีจิสเตอร์ที่ใช้สำหรับเก็บข้อมูลที่เกิดจากการรับส่งข้อมูลพอร์ต I2C
- SSPMASK คือรีจิสเตอร์ที่ใช้สำหรับการเปรียบเทียบแอดเดรสของการสื่อสารด้วยระบบบัส I2C
- SSPADDD คือรีจิสเตอร์ที่ใช้สำหรับกำหนดค่าความเร็วในการสื่อสารด้วยระบบบัส I2C

Digital And Microcontroller

11

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

3.2 รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต I2C ของ ATMEGA32 (PC0, PC1)

TWBR คือรีจิสเตอร์ที่ใช้ในการกำหนดค่าเร็วในการสื่อสารด้วยระบบบัส I2C

TWCR คือรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของพอร์ต I2C

TWDR คือรีจิสเตอร์ที่ใช้ในการเก็บค่าข้อมูลของการรับส่งด้วยระบบบัส I2C

TWAR คือรีจิสเตอร์ที่ใช้ในการกำหนดตำแหน่งแอดเดรสของอุปกรณ์ ในการนี้ที่ใช้ ไมโครคอนโทรลเลอร์ ATMEGA32 ทำหน้าที่เป็นอุปกรณ์ตัวลูก (Slave) ในการสื่อสารด้วยระบบบัส I2C

4. รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต SPI ของไมโครคอนโทรลเลอร์

4.1 รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต SPI ของ AT89C51ED2 (P1.1, P1.5, P1.6 และ P1.7)

SPCON คือรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของพอร์ต SPI และกำหนดค่า ความเร็วของการสื่อสาร

SPSTA คือรีจิสเตอร์ที่ใช้ในการแสดงสถานะ และควบคุมการทำงานของพอร์ต SPI

SPDAT คือรีจิสเตอร์ที่ใช้ในการเก็บค่าข้อมูลของการรับส่งด้วยระบบบัส SPI

Digital And Microcontroller

12

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

- 4.2 รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต SPI ของ PIC16F887 (RA5 , RC3 , RC4 และ RC5) SSPSTAT คือรีจิสเตอร์ที่ใช้แสดงสถานะและกำหนดลักษณะการทำงานของพอร์ต SPI SSPCON คือรีจิสเตอร์ที่ใช้ควบคุมการทำงานของพอร์ต SPI SSPBUF คือรีจิสเตอร์ที่ใช้สำหรับเก็บข้อมูลที่เกิดจากการรับส่งข้อมูลที่พอร์ต SPI
- 4.3 รีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานพอร์ต SPI ของ ATMEGA32 (PB4 , PB5 , PB6 และ PB7) SPCR คือรีจิสเตอร์ที่ใช้ควบคุมการทำงานของพอร์ต SPI SPSR คือรีจิสเตอร์ที่ใช้แสดงสถานะการทำงานของพอร์ต SPI และเพิ่มความเร็วในการติดต่อสื่อสารด้วยพอร์ต SPI อีก 2 เท่า
- SPDR คือรีจิสเตอร์ที่ใช้สำหรับเก็บข้อมูลที่เกิดจากการรับส่งข้อมูลที่พอร์ต SPI

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

5. การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ I2C ของไมโครคอนโทรลเลอร์
- 5.1 การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ I2C ของ AT89C51ED2
- การเขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ I2C ของ AT89C51ED2 จะเป็นการเขียนฟังก์ชันที่เรียกว่า Software I2C
- ```
void i2c_delay() {
 void i2c_clk() {
 void i2c_start() {
 void i2c_stop() {
 bit i2c_wrdData(unsigned char dat){ }
 unsigned char i2c_rdata() { }
 void i2cWriteByte(unsigned char control,unsigned char addr,unsigned char dat){ }
 unsigned char i2cReadByte(unsigned char control,unsigned char addr){ }
 void init_i2c() { }
```

## การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

- 5.2 การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ I2C ของ PIC16F887
- การเขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ I2C ของ PIC16F887 จะเป็นการเขียนฟังก์ชันที่เรียกว่า Hardware I2C
- ```
void I2C_Init(long i2c_clk_freq){ }
void I2C_Start() { }
void I2C_Repeated_Start() { }
void I2C_Stop() { }
void I2C_Write(unsigned char i2c_data){ }
unsigned char I2C_Read(unsigned char ack){ }
void i2cWriteByte(unsigned char ctl,unsigned char addr,unsigned char dat){ }
unsigned char i2cReadByte(unsigned char ctl,unsigned char addr){ }
```

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

- 5.3 การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ I2C ของ ATMEGA32
- การเขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ I2C ของ ATMEGA32 จะเป็นการเขียนฟังก์ชันที่เรียกว่า Hardware I2C
- ```
void i2c_init(uint32_t freq){ }
void i2cWriteByte(char address, char reg, char data){ }
unsigned char i2cReadByte(char address, char reg){ }
```
6. การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ SPI ของไมโครคอนโทรลเลอร์
- 6.1 การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ SPI ของ AT89C51ED2
- เนื่องจากการทำงานของพอร์ต SPI แบบ Hardware ของ AT89C51ED2 มีปัญหาทำให้ไม่สามารถส่งงานให้ทำงานได้ ดังนั้น การติดต่ออุปกรณ์ด้วยพอร์ต SPI ของ AT89C51ED2 จึงกระทำแบบ Software SPI
- ```
void spi_init() { //Software SPI CPOL = 0 , CPHA = 0
    unsigned char spi_write(char dat){ }
```

การรับส่งข้อมูลอนุกรมในรูปแบบ I2C และ SPI

- 6.2 การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ SPI ของ PIC16F887
- การเขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ SPI ของ PIC16F887 จะเป็นการเขียนฟังก์ชันที่เรียกว่า Hardware SPI
- ```
void SPInit() { }
unsigned char SPIWrite(unsigned char data){ }
```
- 6.3 การใช้เขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ SPI ของ ATMEGA32
- การเขียนฟังก์ชันภาษาซีเพื่อควบคุมการรับส่งข้อมูลอนุกรมในรูปแบบ SPI ของ ATMEGA32 จะเป็นการเขียนฟังก์ชันที่เรียกว่า Hardware SPI
- ```
void SPInit() { }
uint8_t SPIWrite(uint8_t data){ }
```