|  |
| --- |
| SPI 1. SPI란?  - SPI는 Serial Peripheral Interface의 약자로서 직렬 통신의 한 방법 이며 동기식 통신을 한다.(모토로라에서 제안)  - Data 송수신에는 3개의 선이 필요하며 Full-duplex통신이 가능하다  (Full-duplex란 data 송신부와 수신부가 동시에 data전송, 즉 교환이 가능하다는 의미이다. 이와 대비되는 뜻으로는 Half-duplex가 있으며 간단하게 Full-duplex는 전화기, Half-duplex는 무전기로 보면 된다.)  - Single Master 구조를 가지며 Slave는 Multi-slave로 여러 개를 가질 수 있는데 이때는 selection을 위해 선이 더 필요하다.  - 주로 근거리 통신, 즉 chip간 통신에 사용된다.  2. SPI 통신의 구조  - 4개의 SPI통신용 핀      1) SPISOMI : SPI slave-output/master-input pin          (slave에서는 전송, master에서는 수신을 하는 pin)      2) SPISIMO : SPI slave-input/master-output pin          (slave에서는 수신, master에서는 전송을 하는 pin)      3) /SPISTE : SPI Slave transmit-enable pin      4) SPICLK : SPI serial-clock pin ( SPI통신은 동기식 이므로 CLK signal이 필요)  - SPI 동작구조  http://pds21.egloos.com/pds/201107/31/72/d0109072_4e351f6185363.jpg  위의 그림은 SPISTE, 즉 enable pin을 제외한 master와 slave간의 data 송수신 구조만 나타낸 것이다. 옆의 그림에서 ss는 multi-slave에서 slave를 선택하는 pin이다.  그림에서 알 수 있듯이 master와 slave 모두 본인의 MSB부터 전송하며 받은 data는 LSB부터 채워나감을 알 수 있다. 이 때 동작 구조를 선택할 수 있는데      1) master만 slave로 data전송          (master가 data전송, dummy 수신)      2) master와 slave간 상호 전송          (master와 slave의 data가 교환)      3) slave만 master로 data전송          (slave만 data전송, dummy 수신)  3. 28x의 SPI의 특징  - Master/Slave모드 모두 지원  - 자체 분주기로 125가지 통신속도 조절 가능  - 통신 데이터 크기 조절 가능(1~16bit)  - 4가지 방식으로 동기 클럭을 설정  ( 1) Rising edge without delay  2) Rising edge with delay  3) Falling edge without delay  4) Falling edge with delay )  - polling 또는 interrupt 방식으로 data 송수신 가능  - 16레벨의 FIFO를 가진다. |
| I2C  1. I2C란?  - Inter-Integrated Circuit의 약자로 동기식 직렬통신의 한 방식이다.  - 통신에는 2개의 선이 필요하며 Half-duplex 통신이 가능하다.  (Half-duplex는 무전기와 같이 data 동시교환이 불가능하다.)  - Multi-master mode를 지원하기 때문에 여러 개의 주변회로와 연결이 가능하다.  - 이 역시 SPI와 마찬가지로 chip간 근거리 통신에 자주 이용된다.  - I2c는 마이크로프로세서와 저속의 주변장치 사이의 통신을 용도로 Philips 에서 개발한 규격  - TWI(Two Wire Interface)라는 방식으로 인터페이싱을 한다. (데이터, 클럭 각각 1핀씩 총 2핀)  - 양방향 오픈드레인선인 SCL(Serial Clock)  2. I2C의 동작구조  http://pds21.egloos.com/pds/201107/31/72/d0109072_4e351ff2b286b.jpg  I2C는 기본적으로 위와 같은 구조이다. Multi-master와 Multi-slave 모두 같은 통신 라인에 연결 되어있다.  1) 라인에 연결된 마스터들 중 하나가 Start신호를 보내면 Bus를 점유      (만약 마스터들이 동시에 start신호를 보내면 낮은 slave address를 보낸 master가 우선순위를 가진다.)  2) Bus를 점유한 master가 slave address와 Read/Write 정보를 Bus에 전송  3) master가 보낸 address와 일치하는 slave가 ACK신호를 전송  4) ACK신호를 받은 마스터가 slave에 data전송, 전송 후 Bus 점유를 해제      I2C의 인터럽트로는 7개의 기본 인터럽트와 2개의 FIFO인터럽트가 가 있다. 7개의 기본인터럽트는 여러 개가 한 번에 걸릴 수      없고 Arbiter에 의해 한 개가 선택되어 걸린다.      그리고 FIFO 인터럽트는 기본 인터럽트와는 별개 이므로 기본 인터럽트와 중복되어 걸릴 수 있다. |
| SCI 1. SCI란?  - SCI는 Serial Communication Interface의 약자로 비동기식 직렬통신의 한 방식  - 일반적으로 UART라고 불리기도 하며 1:1 통신에 특화된 면모를 보인다.  - TX, RX 두 개의 선을 사용하며 Full-duplex 통신을 지원한다.  - 주로 장거리 통신(보드 간 통신)에 이용하므로 chip의 신호만으로는 그 강도가 약하기 때문에 별도의 트랜시버를 이용해서 신호를 증폭해 통신한다. (우리가 알고 있는 RS-232C, RS-422 등은 이런 트랜시버에 속한다.)  2. SCI의 동작구조  http://pds22.egloos.com/pds/201107/31/72/d0109072_4e352052f1833.jpg  SCI의 구조는 간단하게 위 그림과 같다. 이 경우 Data 송수신이 별도로 이루어지는데 SCI의 경우는 2개의 Shift register가 있으므로 Data의 상호 교환이 가능하다. SPI와는 달리 SCI는 LSB부터 전송이 되는데 TX shift register의 LSB가 TX핀으로 넘어가고 수신부 RX핀으로 전달된다. 이 전송된 비트는 RX shift register의 MSB로 넘어가게 된다.  SCI의 경우는 비동기식 직렬 통신이기 때문에 통신 속도(Baud rate)를 맞춰줘야 하는데 예를 들어 왼쪽의 SCI이 9600bps(baud per second)의 통신 속도를 가지면 오른쪽의 SCI도 9600bps의 통신 속도를 가져야 한다.  3. SCI의 특징  - 2개의 통신용 핀을 가진다. (SCITXD = TX, SCIRXD = RX)  - 자체 분주기로 65535가지로 통신속도 조절이 가능하다.  - 통신 데이터 크기 조절이 가능하다.  - 4가지 방식의 error detection flag를 가진다.  - NRZ(none return to zero) format을 가진다.  (NRZ란 예를 들어 1101의 data를 보내면 1101 그대로 보내는 것이며 반대로 RZ는 10100010 data사이에 0을 return해 data간 구분이 쉽도록 하는 것이다.)  - polling 방식, interrupt방식 모두로 송 수신이 가능하다.  - 16비트 송수신 FIFO를 가지고 있다.  - 자동 Baud rate 감지 기능을 가진다.(한 쪽의 통신속도를 check해서 자동으로 맞춰준다.) |
| ECAN / CAN  1. eCAN이란?  - Enhanced Controller Area Network의 약자로 비동기식 직렬 통신의 한 방식이다.  - 2개의 선으로 통신하며 SCI와는 달리 Half-duplex 통신이 가능하다.  - 멀티마스터-멀티슬레이브 구조가 가능해 N:N 통신이 가능하다.  - Differential 신호방식을 사용하므로 노이즈에 강하다.  - 주로 중장거리 보드 간 통신에 사용되므로 이 역시 별도의 트랜시버를 필요로 한다.  2. eCAN의 동작 구조  http://pds21.egloos.com/pds/201107/31/72/d0109072_4e3520ee3f9a2.jpg  CAN통신은 일반적으로 위와 같은 구조를 가진다. CAN을 나온 data는 트랜시버를 거쳐 CAN H, CAN L로 data가 넘어가게 되는데 이 트랜시버를 거치며 사용하는 신호 방식인 differential 신호 방식이 CAN통신의 특징이다.  트랜시버에는 TX에서 나온 data를 받아서 Inverting을 하는 데 1번 한 것을 CAN H에, 2번 한 것을 CAN L에 전달하고 이 두 신호의 전위차로 원 신호를 파악하게 된다. 좀 더 자세하게 살펴보기 위해 그래프를 살펴본다.  http://pds22.egloos.com/pds/201107/31/72/d0109072_4e35213f537da.jpg  이 그래프는 TX에서 101이 트랜시버를 거쳐서 나왔을 때를 나타낸 것인데 CAN H와 CAN L의 전위차가 작으면 1, 전위차가 크면 0으로 하여 원 신호인 101을 파악할 수 있음을 알 수 있다. 그리고 이러한 신호 방식은 noise가 생기더라고 전위 차이는 바뀌지 않기 때문에 noise에 굉장히 강한 모습을 나타낸다.  http://pds22.egloos.com/pds/201107/31/72/d0109072_4e35214f3229a.jpg  그리고 CAN통신에서 실제로 주고 받는 data는 이 Mail box에 저장되고 보내지는데 main box에는 각각의 ID가 있어서 이 ID가 address역할을 하여 특정 mail box의 data를 송수신 할 수 있다.  CAN에서 실제로 전달되는 메시지 구조는 ID, Data를 제외하고 Start, ACK 등 여러 가지가 포함되는데 이는 따로 저장되어 있지 않고 개발자가 설정하면 자동으로 추가되어 전달된다.  data를 주고받는 방식은 SPI와 비슷하게 master에서 ID와 data를 같이 송신하면 slave에서 ID를 check하고 만약 ID가 일치하면 data를 받아들이게 된다. |
|  |