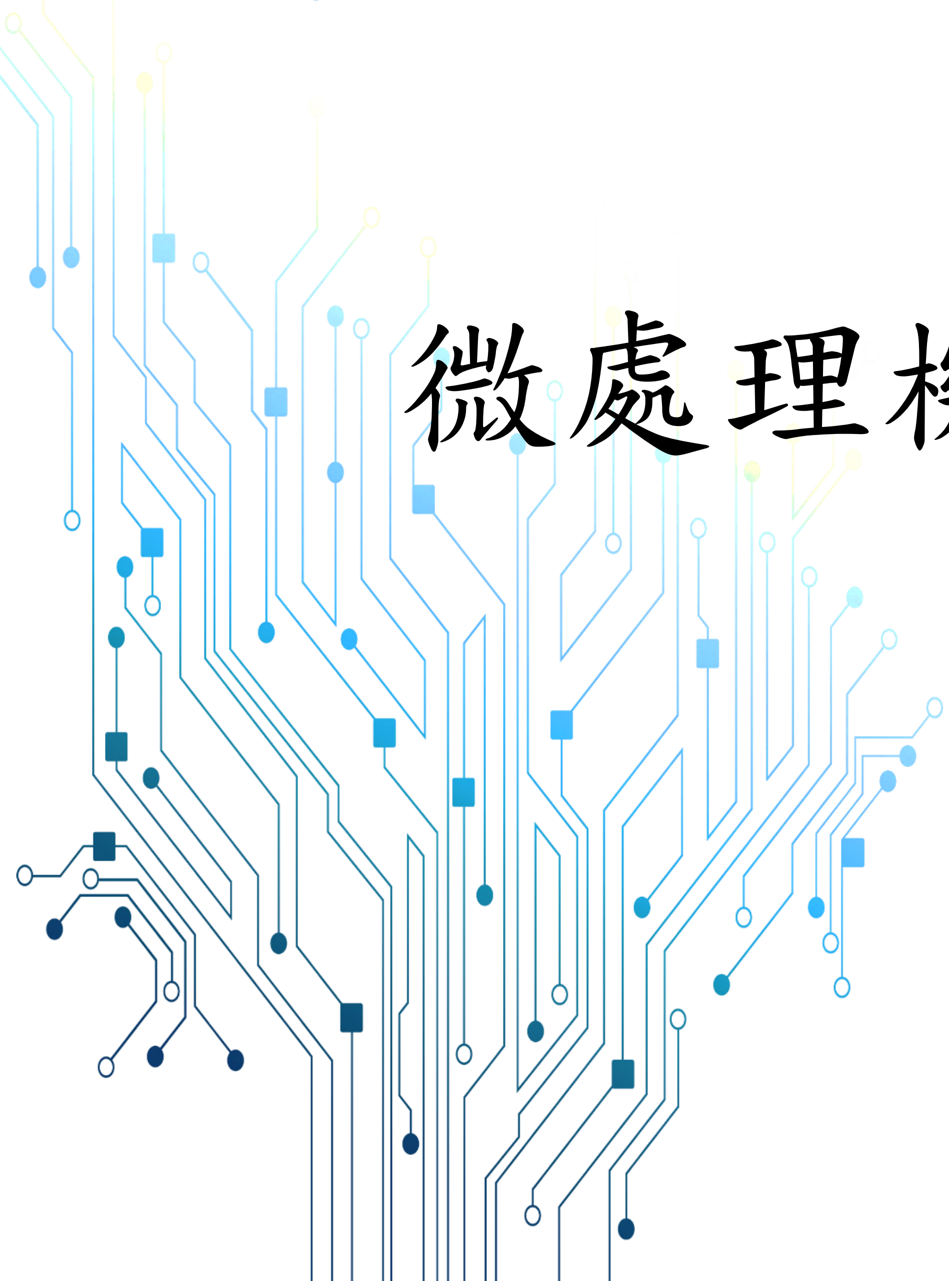
國立中正大學
National Chung Cheng University



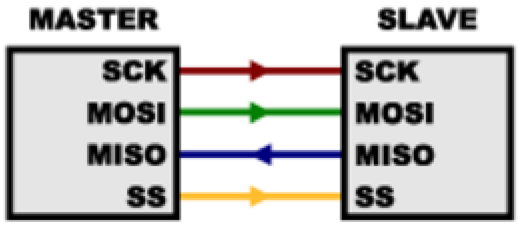
# 微處理機系統與介面技術

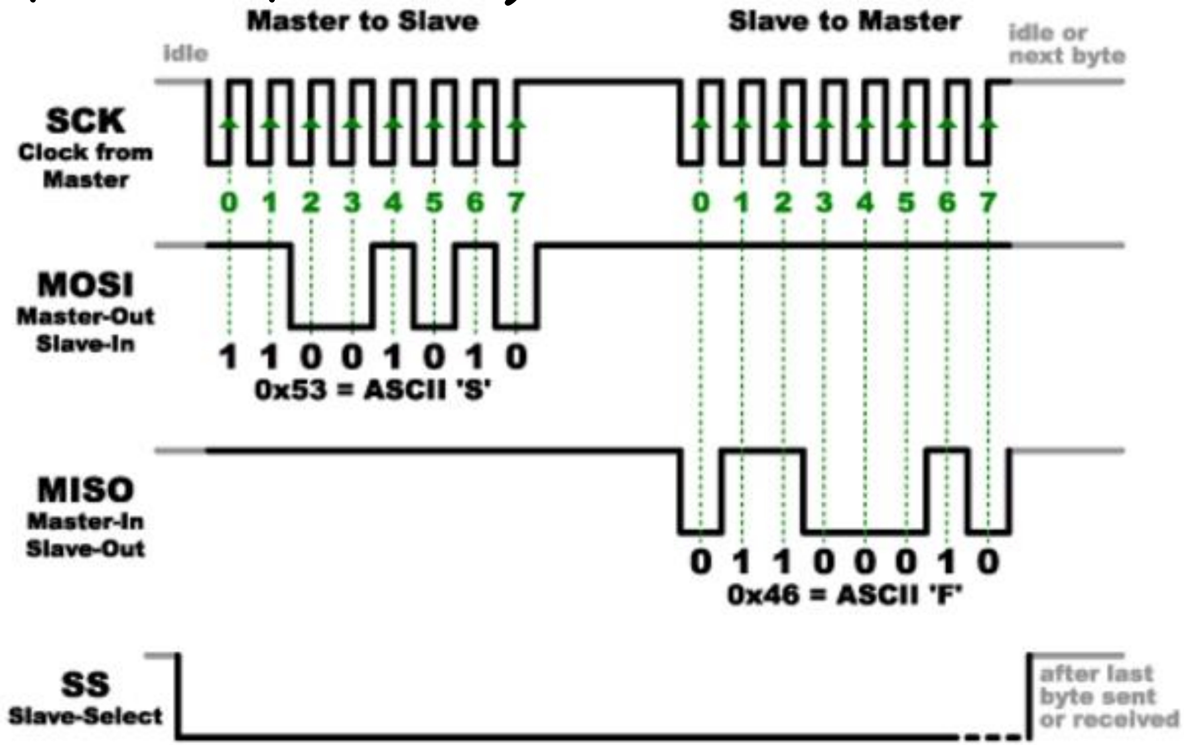
## LAB 4 — SPI

www.neatlab.tw
neat
Laboratory
CCU@Taiwan 中正大學

# SPI - Serial Peripheral Interface

- Synchronous serial data communication(can operate in full duplex)
- 4 wire communication(SS,CLK,MOSI,MISO)
- NUC140 has four set SPI

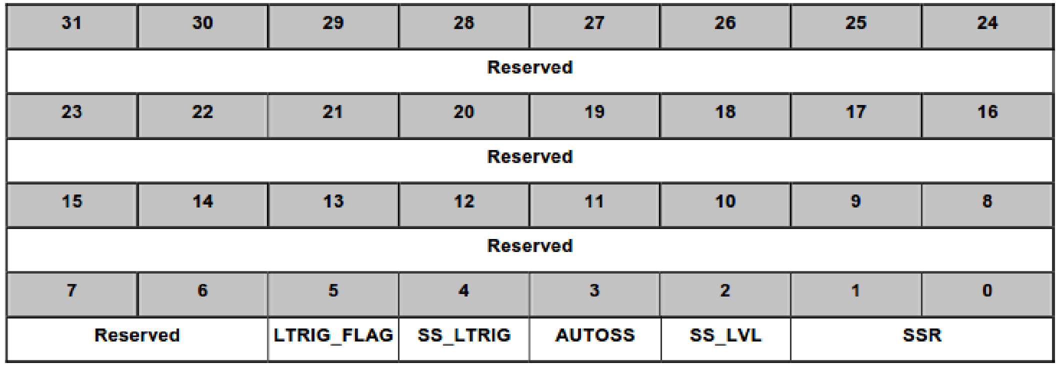




https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi

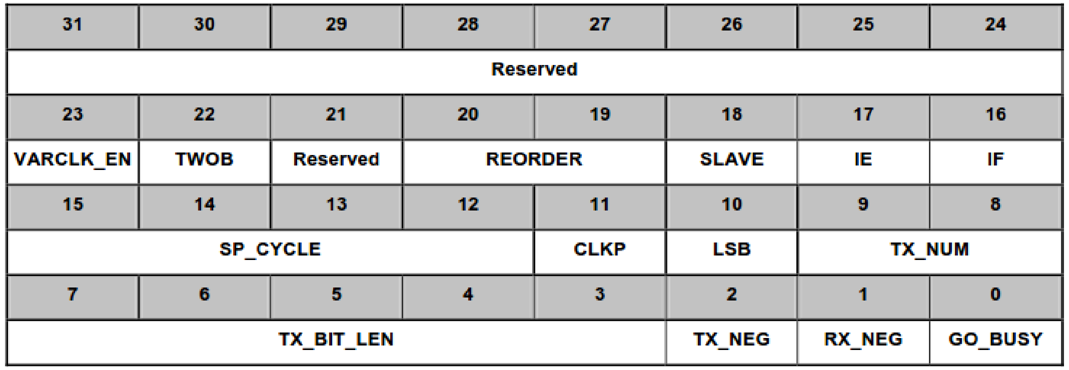
# SPI register

- SPI->SSR: SPI slave select register
  - SS_LVL
  - AUTOSS
  - SSR



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| Reserved | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Reserved | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Reserved | | LTRIG_FLAG | SS_LTRIG | AUTOSS | SS_LVL | SSR | |

- SPI->CNTRL: SPI control and status register
  - SLAVE
  - CLKP
  - TX_NUM, TX_BIT_LEN
  - TX_NEG, RX_NEG
  - GO_BUSY



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
| Reserved | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| VARCLK_EN | TWOB | Reserved | REORDER | | SLAVE | IE | IF |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| SP_CYCLE | | | | CLKP | LSB | TX_NUM | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TX_BIT_LEN | | | | | TX_NEG | RX_NEG | GO_BUSY |

# ADXL SPI configuration

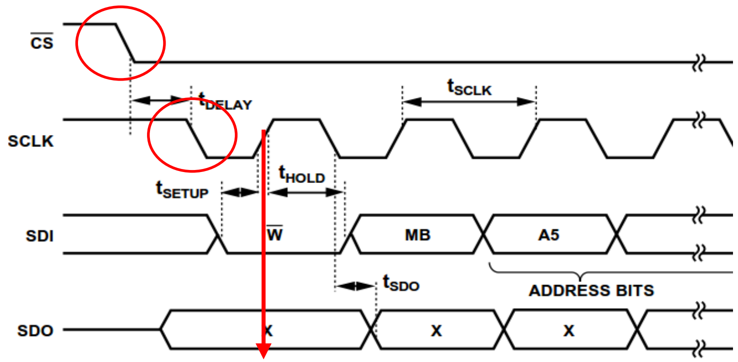


Figure 37. SPI 4-

- SPI->DIVIDER: Set SPI clock(DIVIDER)
- SPI->SSR
  - SS line is active at low-level edge(SSR.SS_LVL)
  - Disable auto ss(SSR.AUTOSS)
- SPI->CNTRL
  - Set SPI as master mode(CNTRL.SLAVE)
  - CLK is idle at high(CNTRL.CLKP)
  - CPOL=1, CPHA=1(CNTRL.TX_NEG, RX_NEG)
  - 8 bit data length for each word transmit(CNTRL.TX_BIT_LEN)
  - One word in one transfer(CNTRL.TX_NUM)

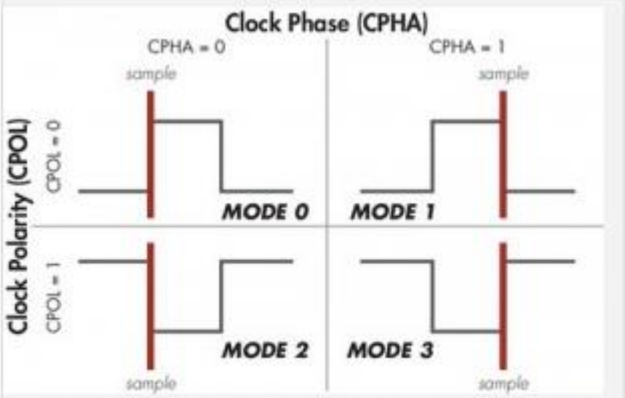


Figure 2: Clock Polarities and Clock Phases of SPI Modes

# ADXL345 register map

- Initial ADXL345
  - POWER_CTL(0x2D): 0x08
  - DATA_FORMAT(0x31): 0x0B
  - FIFO_CTL(0x38): 0x80
- ADXL data register
  - DATAX0(0x32), DATAX1(0x33)
  - DATAY0(0x34), DATAY1(0x35)
  - DATAZ0(0x36), DATAZ1(0x37)

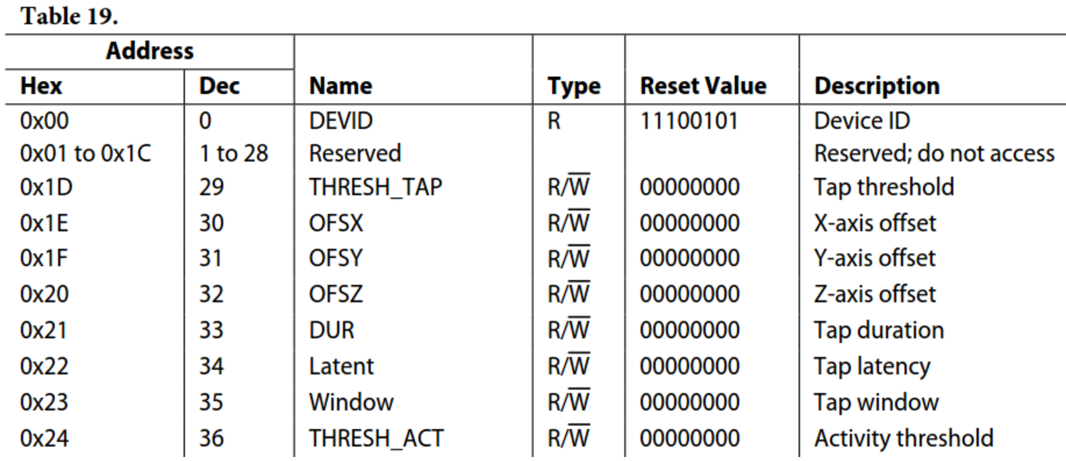
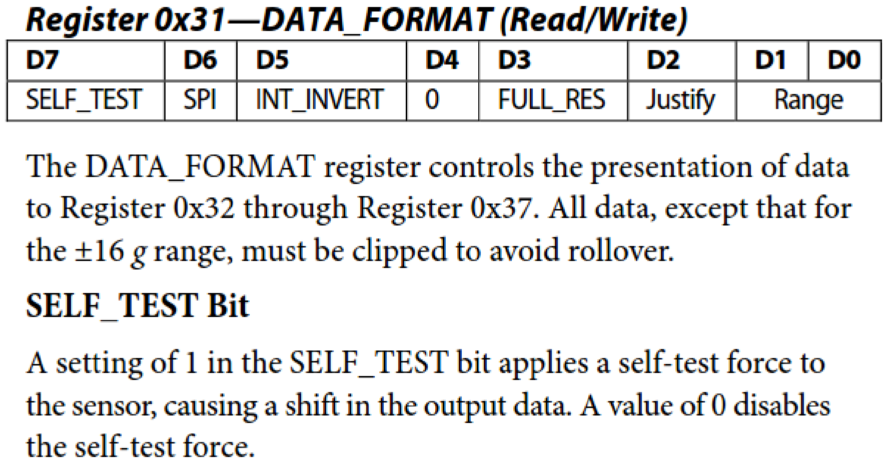


Table 19.

| Address | | Name | Type | Reset Value | Description |
|---|---|---|---|---|---|
| Hex | Dec | | | | |
| 0x00 | 0 | DEVID | R | 11100101 | Device ID |
| 0x01 to 0x1C | 1 to 28 | Reserved | | | Reserved; do not access |
| 0x1D | 29 | THRESH_TAP | R/W̄ | 00000000 | Tap threshold |
| 0x1E | 30 | OFSX | R/W̄ | 00000000 | X-axis offset |
| 0x1F | 31 | OFSY | R/W̄ | 00000000 | Y-axis offset |
| 0x20 | 32 | OFSZ | R/W̄ | 00000000 | Z-axis offset |
| 0x21 | 33 | DUR | R/W̄ | 00000000 | Tap duration |
| 0x22 | 34 | Latent | R/W̄ | 00000000 | Tap latency |
| 0x23 | 35 | Window | R/W̄ | 00000000 | Tap window |
| 0x24 | 36 | THRESH_ACT | R/W̄ | 00000000 | Activity threshold |



***Register 0x31—DATA_FORMAT (Read/Write)***

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| SELF_TEST | SPI | INT_INVERT | 0 | FULL_RES | Justify | Range | |

The DATA_FORMAT register controls the presentation of data to Register 0x32 through Register 0x37. All data, except that for the ±16 *g* range, must be clipped to avoid rollover.

**SELF_TEST Bit**

A setting of 1 in the SELF_TEST bit applies a self-test force to the sensor, causing a shift in the output data. A value of 0 disables the self-test force.

# SPI Write operation

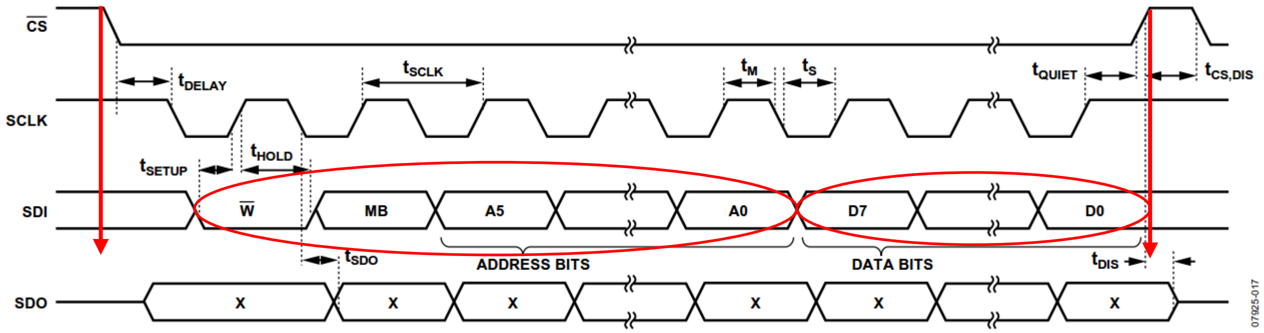
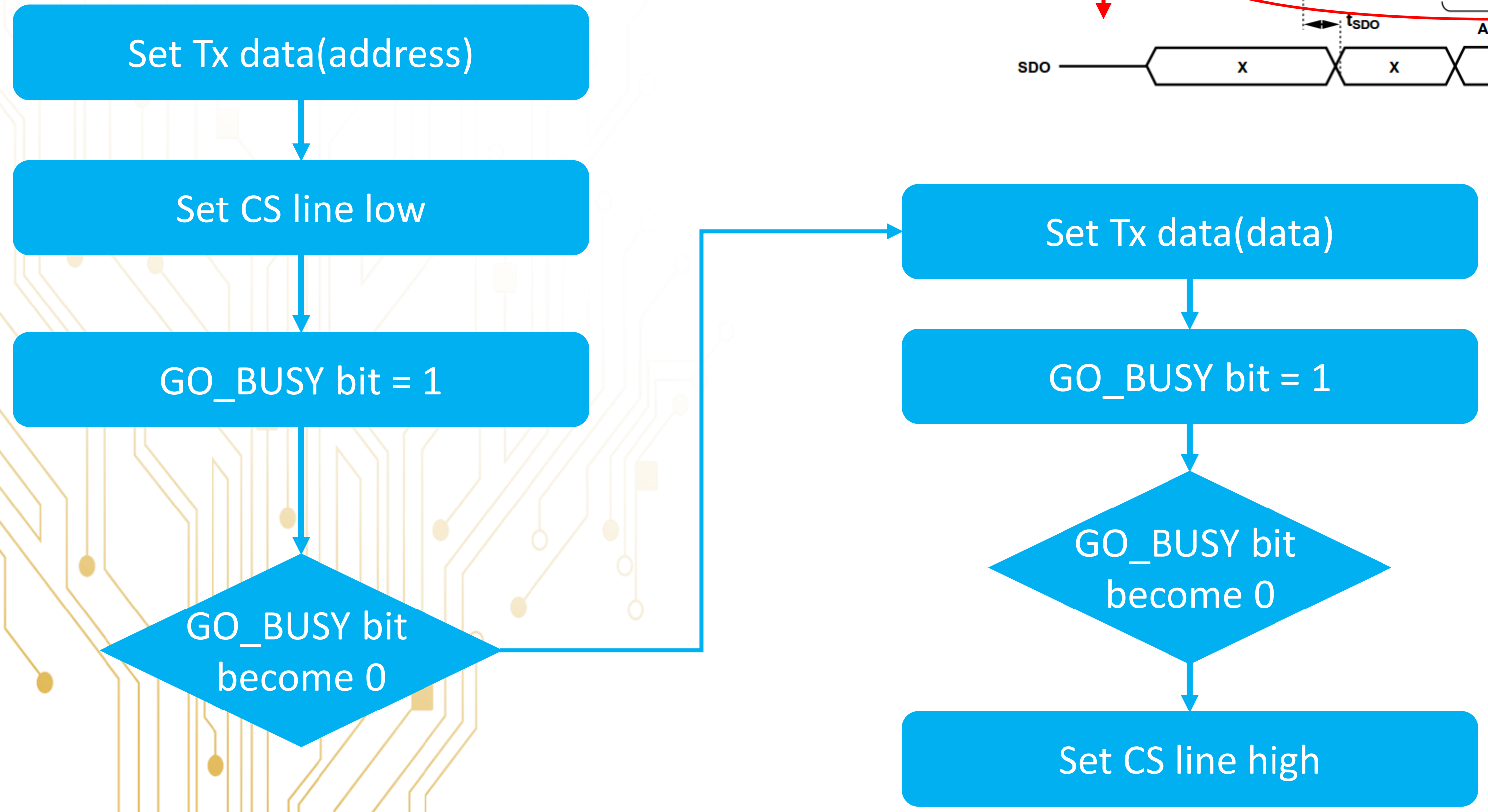


Figure 37. SPI 4-Wire Write



Set Tx data(address)

Set CS line low

GO_BUSY bit = 1

GO_BUSY bit become 0

Set Tx data(data)

GO_BUSY bit = 1

GO_BUSY bit become 0

Set CS line high

Tips: You can see NuMicro_SPI.ppt p.17 for the example code

# SPI Read operation

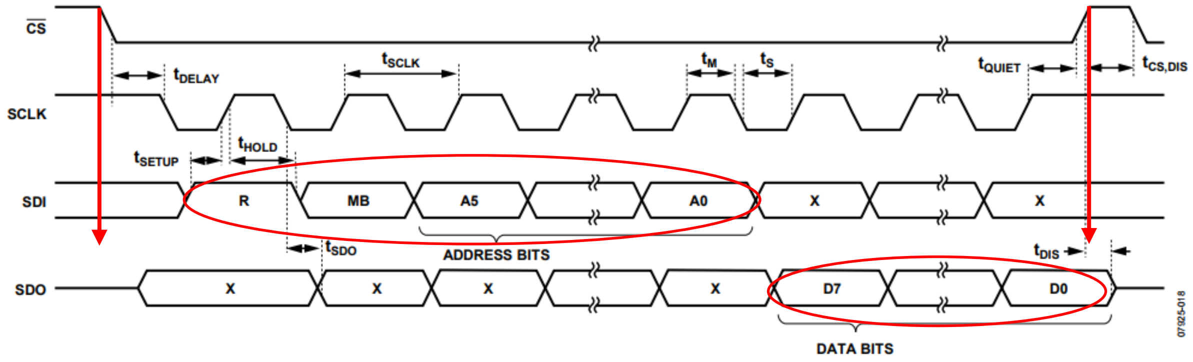
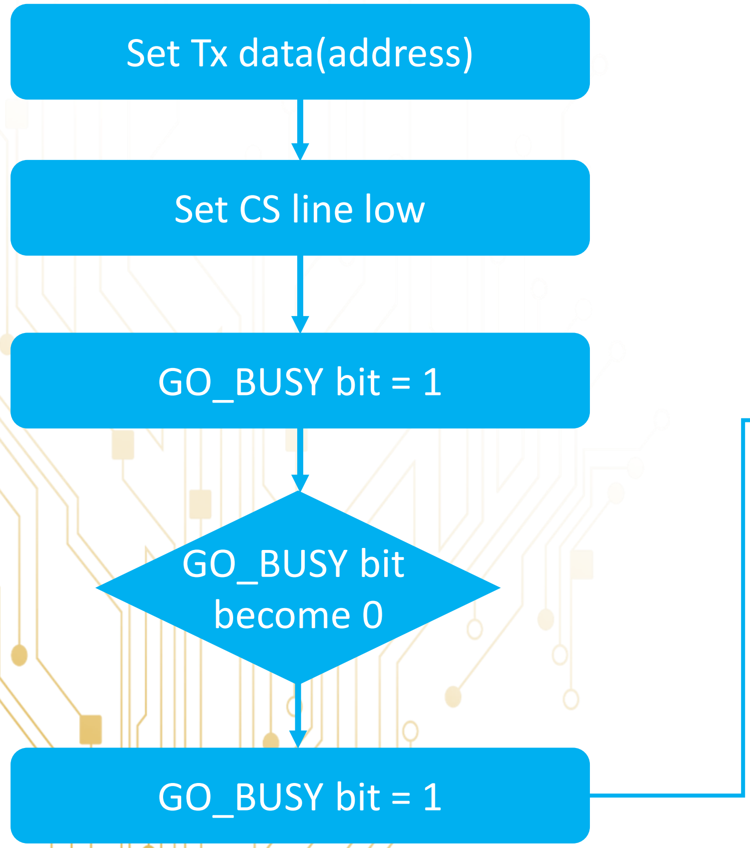


*Figure 38. SPI 4-Wire Read*



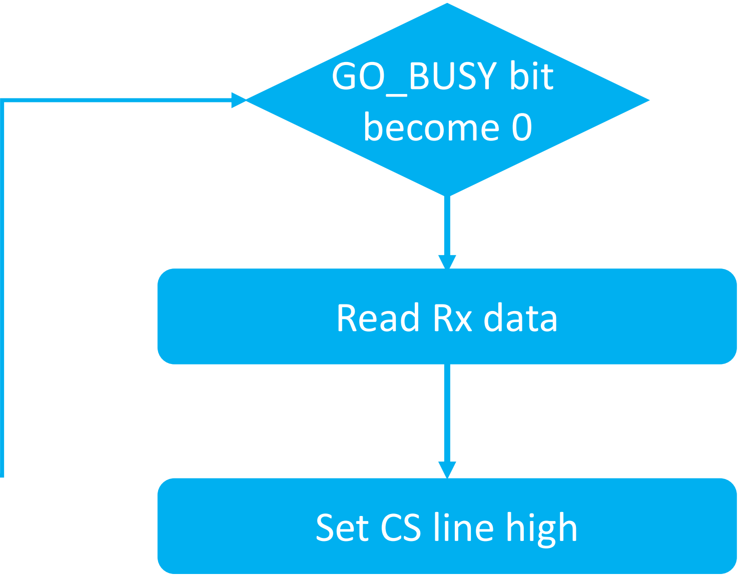
Set Tx data(address)

Set CS line low

GO_BUSY bit = 1

GO_BUSY bit become 0

GO_BUSY bit = 1



GO_BUSY bit become 0

Read Rx data

Set CS line high

Tips: You can see NuMicro_SPI.ppt p.17 for the example code

# ADXL SPI Read/Write

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| R/W | MB | A5 | A4 | A3 | A2 | A1 | A0 |

- Data format
  - Read/Write bit + Multiple-byte bit + 6 bits address
- Configure 0x2D(0x0010_1101) as address, single-byte Read
  - Read + ~~MB~~ + address ➔ 0x1010_1101

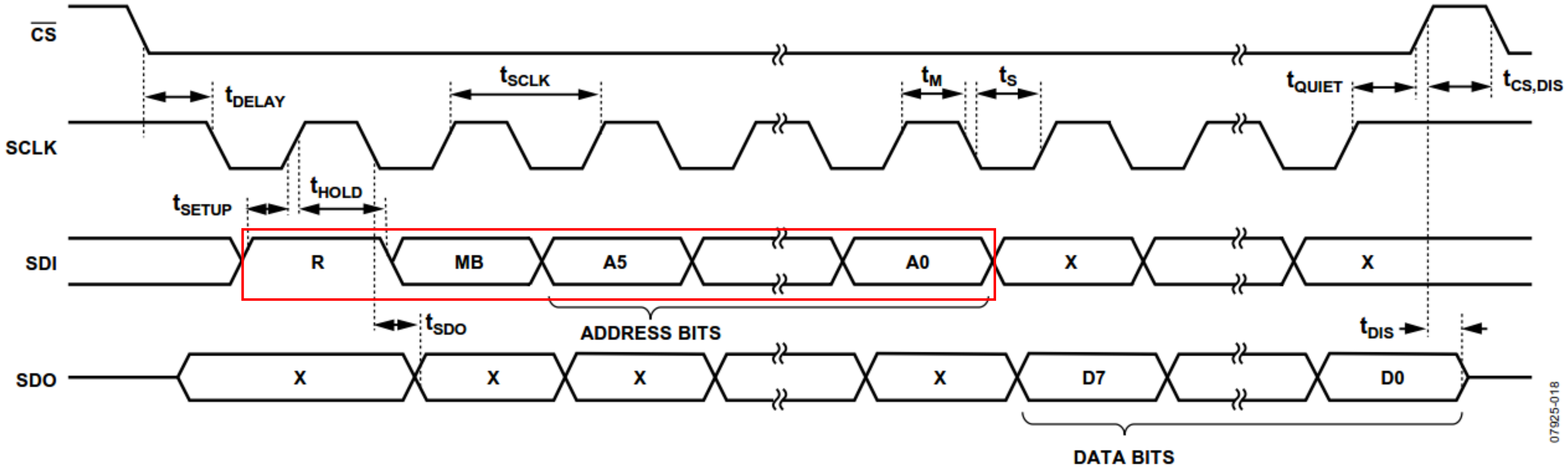


*Figure 38. SPI 4-Wire Read*

# ADXL pin configuration

- CS ------------>SPI2 CS(GPD0)
- SCL ----------->SPI2 CLK(GPD1)
- SDO ----------->SPI2 MISO(GPD2)
- SDA(SDI) ---->SPI2 MOSI(GPD3)

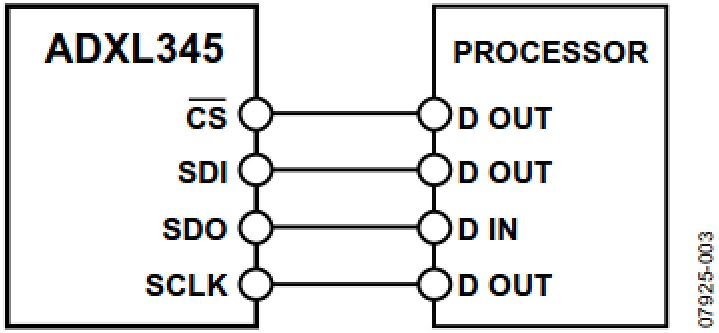
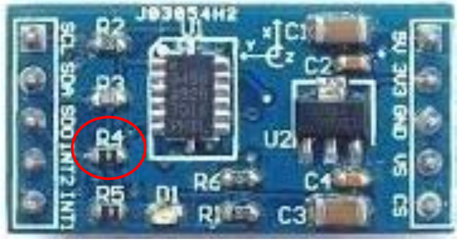
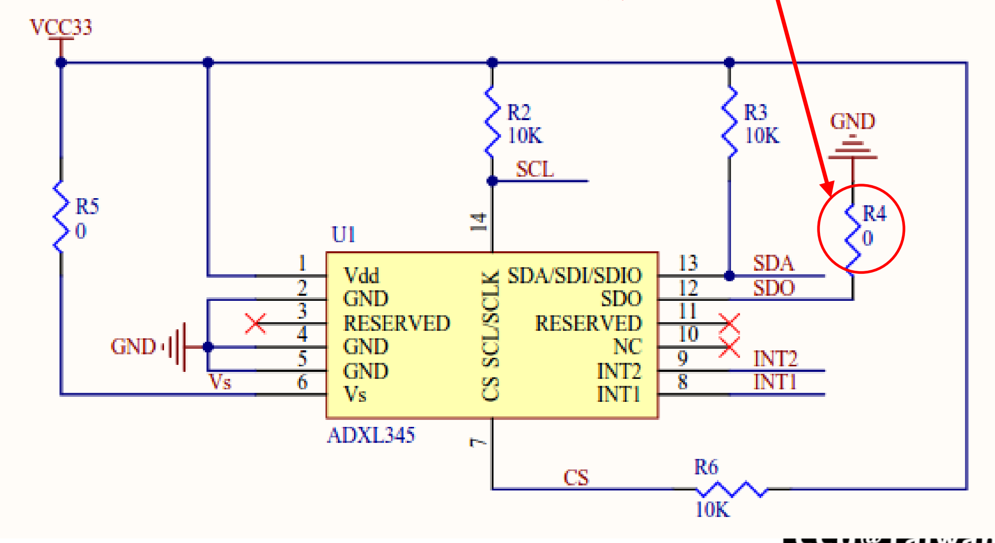


*Figure 35. 4-Wire SPI Connection Diagram*



Bad design for using SPI, we need to remove this resistor then let SPI work



- Mark: Don't use SPI0

# Basic

- Read 3 axis accelerometer and print on putty
- Need to do calibration
  - Result = (Raw data ± offset)/(256 ± offset)

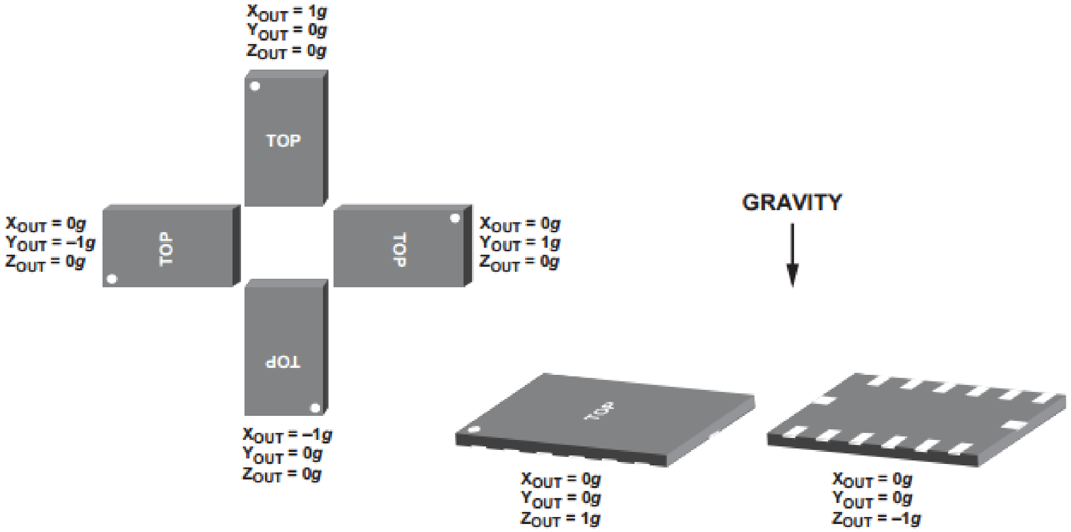
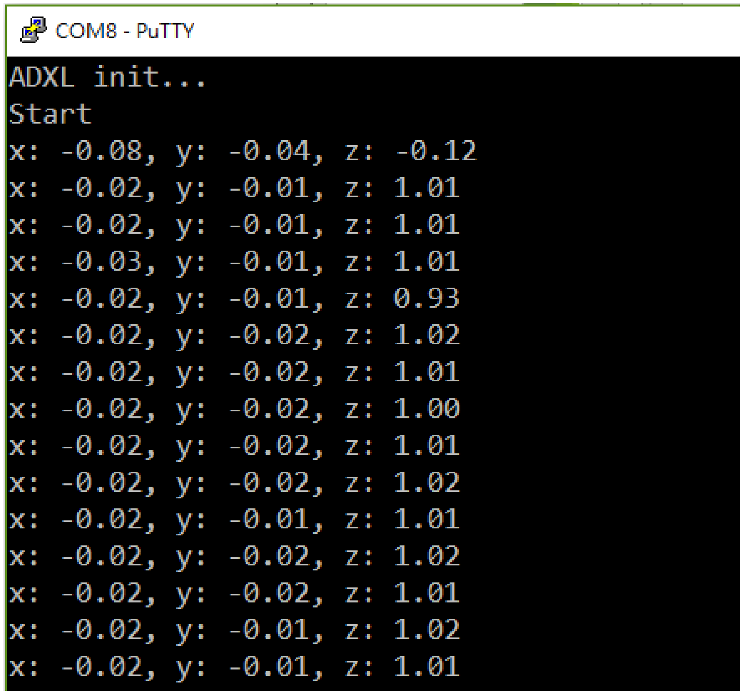


*Figure 58. Output Response vs. Orientation to Gravity*



```
ADXL init...
Start
x: -0.08, y: -0.04, z: -0.12
x: -0.02, y: -0.01, z: 1.01
x: -0.02, y: -0.01, z: 1.01
x: -0.03, y: -0.01, z: 1.01
x: -0.02, y: -0.01, z: 0.93
x: -0.02, y: -0.02, z: 1.02
x: -0.02, y: -0.02, z: 1.01
x: -0.02, y: -0.02, z: 1.00
x: -0.02, y: -0.02, z: 1.01
x: -0.02, y: -0.02, z: 1.02
x: -0.02, y: -0.01, z: 1.01
x: -0.02, y: -0.02, z: 1.02
x: -0.02, y: -0.02, z: 1.01
x: -0.02, y: -0.01, z: 1.02
x: -0.02, y: -0.01, z: 1.01
```

# Tips

- 範例程式: SPI_Loopback
- Easy test: you can read the adxl register 0x00 to test SPI communication is correct or not, it will return 0xE5 if your SPI is right
- Remember to change configuration in the SYS_init
  - Ex. CLK_SEL1(ModuleClock), GPx_MFP, ALT_MFP ➔ change SPI to SPI2
- Do not use AutoSS, SPI.c SPI.h are useful.
- Be careful for the SPI configuration !!!
- You can write the code as the example.c

# Demo

- Place: 創新大樓515 找助教 潘冠豪
- Demo Time: (二)(四)下午兩點~四點半
- Report deadline: 11/22(五)
- Report title format: LABx_ID_Name.pdf

- Demo必須在Report deadline前完成
- Demo前須先上傳程式碼(上傳main所在的.c檔即可)

# Graded

- Basic : 80%
- Report & Code : 20%