國立中正大學
National Chung Cheng University

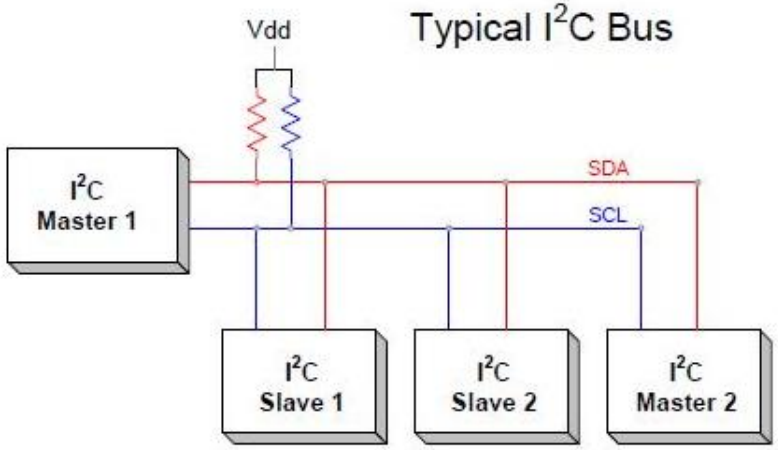
微處理機系統與介面技術

LAB 6 – I2C

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

I2C - Inter-Integrated Circuit

- Multiple Master/Multiple Slave synchronous communication
- Two wires: SDA/SCL
- I2C0
  - I2C0SDA/GPA8 : pin 12
  - I2C0SCL/GPA9 : pin 11
- I2C1
  - I2C1SDA/GPA10 : pin 10
  - I2C1SCL/GPA11 : pin 9



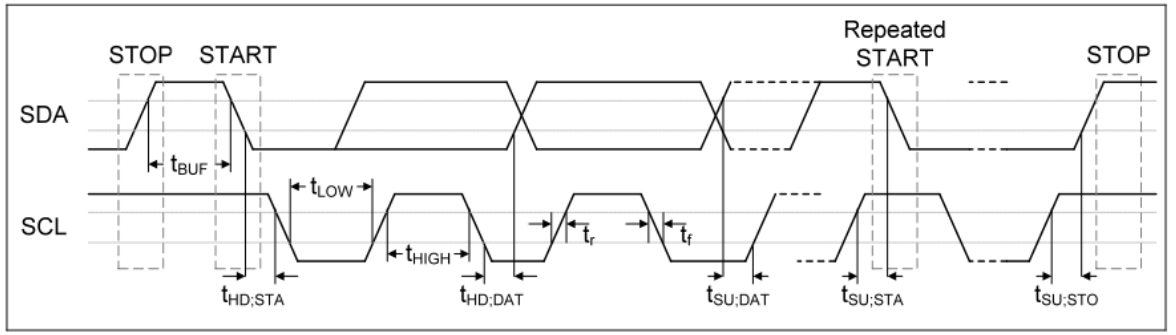


Figure 5-18 I²C Bus Timing

- https://learn.sparkfun.com/tutorials/i2c

I2C register configuration

- I2CON: I2C Control Register
  - EI
  - ENS1
  - STA, STO, SI, AA
- I2CDAT: I2C Data Register
- I2CSTATUS: I2C Status Register

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
EI	ENS1	STA	STO	SI	AA	Reserved	

I2C operation

- STA – start bit
- STO – stop bit
- SI – I2C interrupt flag
  - SI flag is set by hardware if new I2C status present
  - Write 1 to clear this bit
- AA - Acknowledge

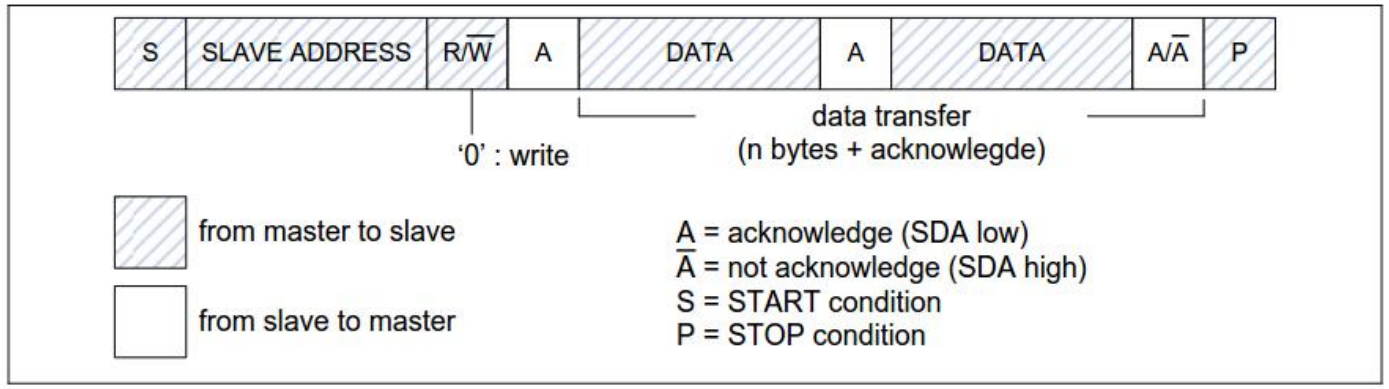
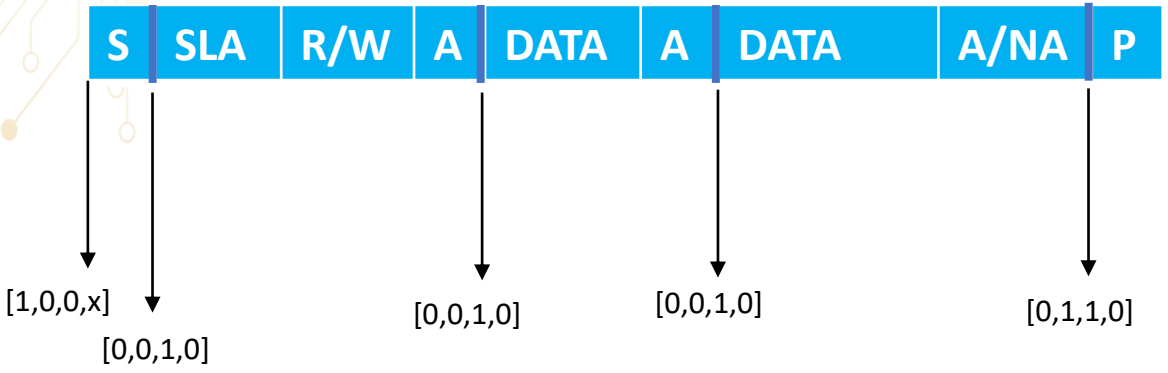


Figure 5-20 Master Transmits Data to Slave



SI flag is set by Hardware

[STA,STO,SI,AA]

ADXL345 – 3 axis Digital Accelerometer

- Pin configuration
  - SDA --------- M0 SDA(GPA8)
  - SCL --------- M0 SCL(GPA9)

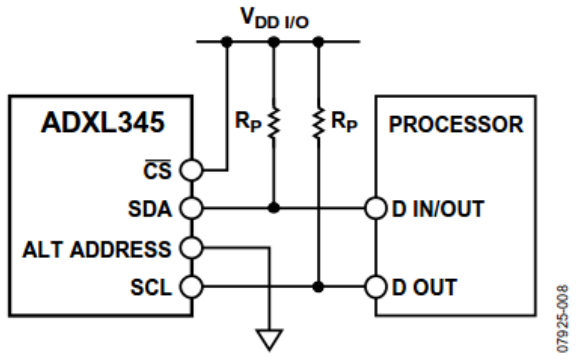


Figure 40. I²C Connection Diagram (Address 0x53)

- Don't care: VS, INT1, INT2
- CS, SCL, SDA is already pull high for this module
- SDO is already pull low

- 如果R4有被拆掉的話,要把SDO接地(要附圖)

ADXL345 register map

- ADXL slave address(0x53)
  - Write address: 0xA6
  - Read address: 0xA7
- 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					
Hex	Dec	Name	Type	Reset Value	Description
0x00	0	DEVID	R	11100101	Device ID
0x01 to 0x1C	1 to 28	Reserved			Reserved; do not access
0x1D	29	THRESH_TAP	R/$\overline{W}$	00000000	Tap threshold
0x1E	30	OFSX	R/$\overline{W}$	00000000	X-axis offset
0x1F	31	OFSY	R/$\overline{W}$	00000000	Y-axis offset
0x20	32	OFSZ	R/$\overline{W}$	00000000	Z-axis offset
0x21	33	DUR	R/$\overline{W}$	00000000	Tap duration
0x22	34	Latent	R/$\overline{W}$	00000000	Tap latency
0x23	35	Window	R/$\overline{W}$	00000000	Tap window
0x24	36	THRESH_ACT	R/$\overline{W}$	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.

Basic

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

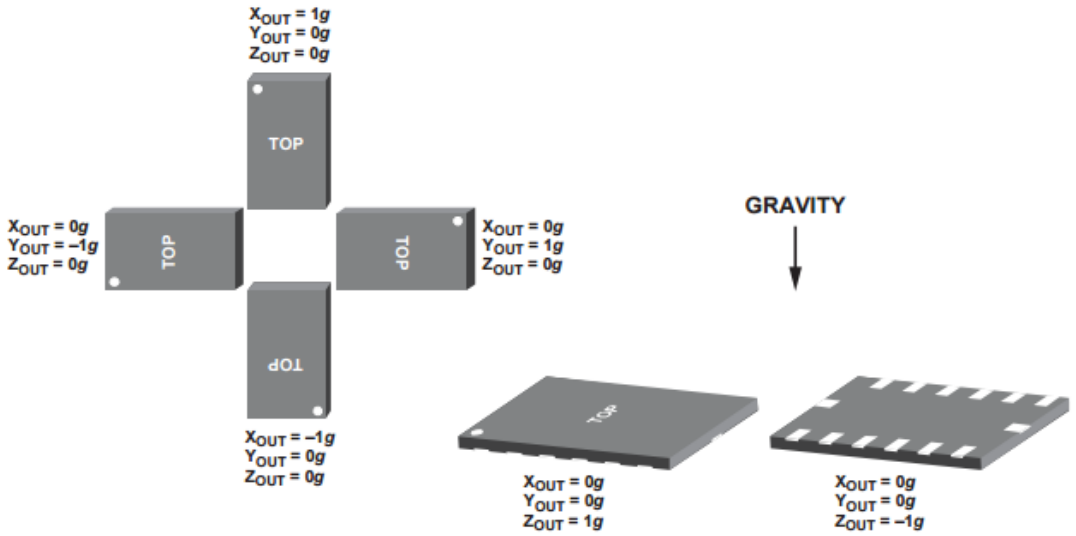
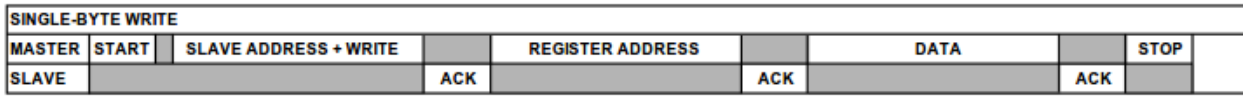


Figure 58. Output Response vs. Orientation to Gravity

```
COM8 - PuTTY
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
```

I2C Polling Architecture

- Polling the SI flag ,do next step when SI flag is set
  - STA ➔ polling SI ➔ SLA+W ➔ polling SI ➔ data(register addr) ➔ polling SI…
  - Remember to clear the SI for each status



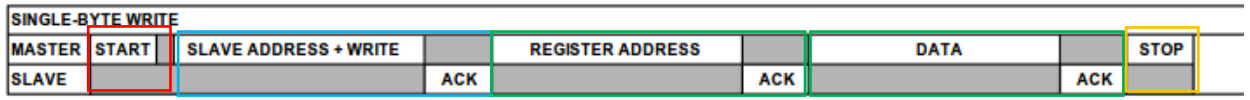
SINGLE-BYTE WRITE										
MASTER	START		SLAVE ADDRESS + WRITE		REGISTER ADDRESS		DATA		STOP	
SLAVE				ACK		ACK		ACK		

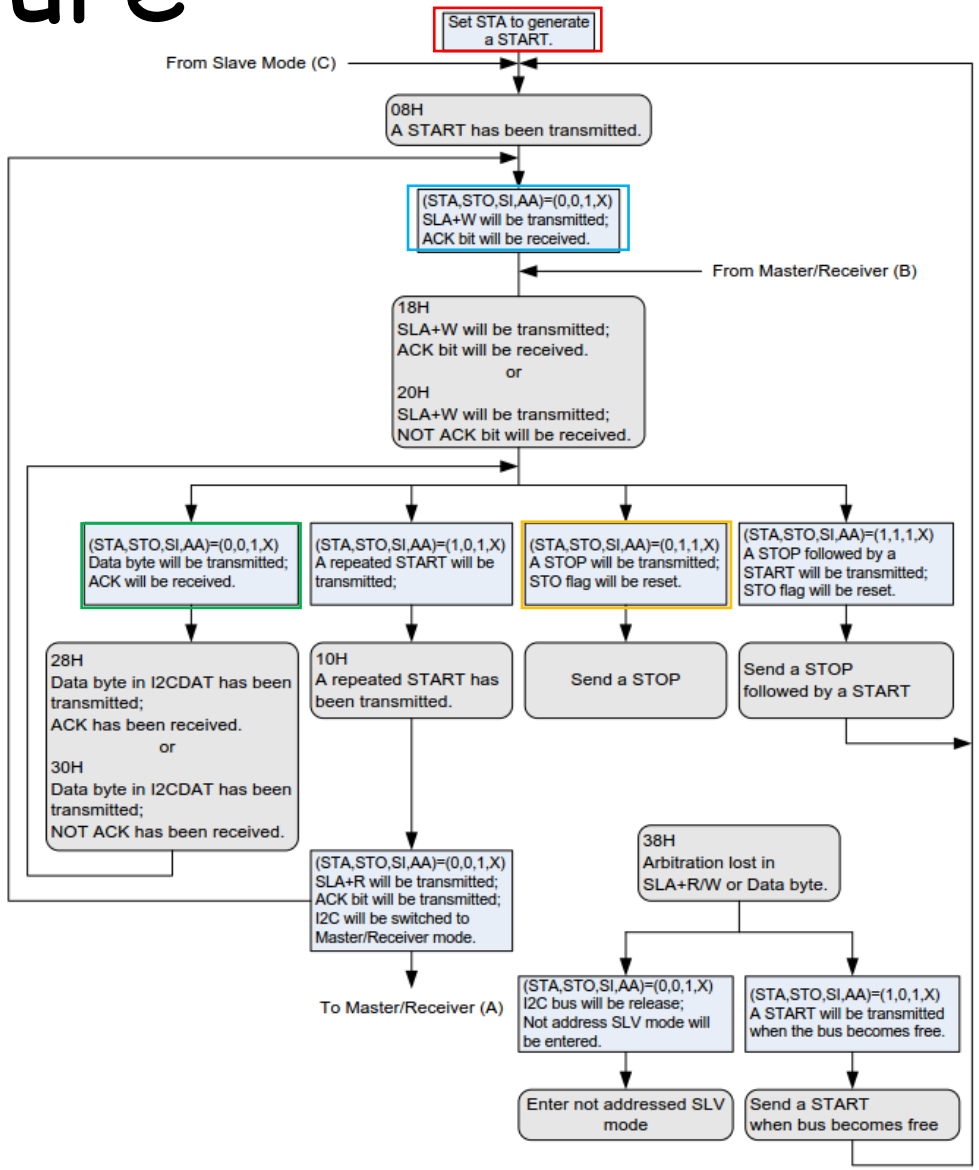
- Data format Hint: xaxis = ((DataX1 << 8) | DataX0)
- Tips: You can reference the NuMicro_I2C.ppt p.17,18 to see I2C Read/Write
- Write Read/Write first, then you can implement adxl_init, adxl_readDataX…

I2C Interrupt Architecture

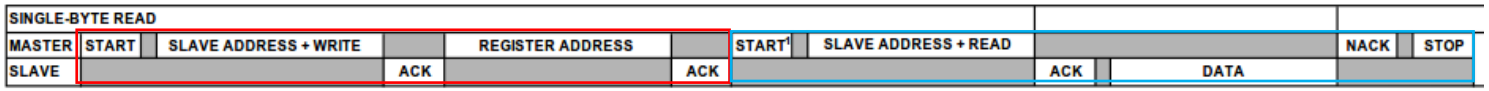
- Doing the I2C step in IRQHandler
  - Status = 0x08(Start)

    next step ➔ Write slave address + write bit
- Following the step with I2C flow chart
  - Master Transmitter/Receiver Mode
- I2C status flow chart is in the Technical Reference pdf p.216,217





Tips



Write flow chart

Read flow chart

- 範例程式: I2C_Master
  - Function pointer
  - Just look at the status what we need, ex. 0x08 Start
- Easy test: you can read the adxl register 0x00 to test I2C communication is correct or not, it will return 0xE5 if your I2C is right
- Be careful for the type cast uint8_t to the float
- Volatile for the waiting flag, maybe for the data
- ADXL Read need to go through two flow chart!!!

Demo

- Place: 創新大樓515 找助教 林子華(進門最後一排最裡面)
- Demo Time: (二)(三)下午三點~五點
- Report deadline: 12/18(五)
- Report title format: LABx_ID_Name
- 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%