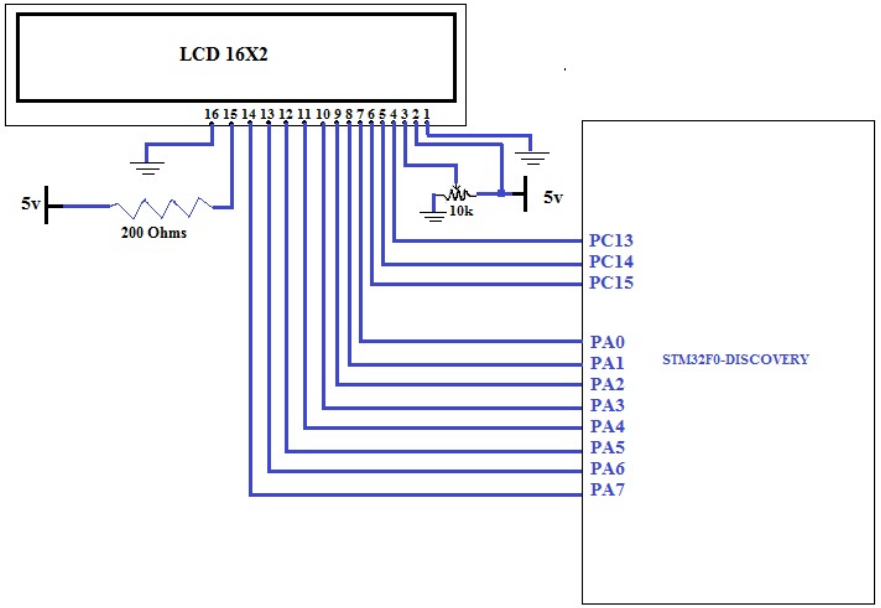
LCD and Temperature Sensors

Connect LCD to ARM

PIN CONFIGURATION FOR 16X2 LCD

PIN 1 - VSS	PIN9 - D2
PIN 2 - VCC	PIN10 - D3
PIN 3 - VEE	PIN11 - D4
PIN4 - RS	PIN12 - D5
PIN5 - R/W	PIN13 - D6
PIN6 - EN	PIN14 - D7
PIN7 - D0	PIN15 - BACKLIGHT +
PIN8 - D1	PIN16 - BACKLIGHT GND



Command	Code
Clear Display, Cursor to Home	0x0001
Cursor to Home	0x0002
Entry Mode:	
Cursor Decrement, Shift off	0x0004
Cursor Decrement, Shift on	0x0005
Cursor Increment, Shift off	0x0006
Cursor Increment, Shift on	0x0007
Display Control:	
Display, Cursor, and Cursor Blink off	0x0008
Display on, Cursor and Cursor Blink off	0x000C
Display and Cursor on, Cursor Blink off	0x000E
Display, Cursor, and Cursor Blink on	0x000F
Cursor / Display Shift: (nondestructive move)	
Cursor shift left	0x0010
Cursor shift right	0x0014
Display shift left	0x0018
Display shift right	0x001C
Display Function (2 rows for 4-bit data; big)	0x002C
Display Function (2 rows for 4-bit data; small))	0x0028
Display Function (1 row for 4-bit data; big)	0x0024
Display Function (1 row for 4-bit data; small)	0x0020

RS	R/W	Mode
0	0	Write Command (Write IR)
0	1	Read Status (Read IR)
1	0	Write DR
1	1	Read DR

Lab 9.1

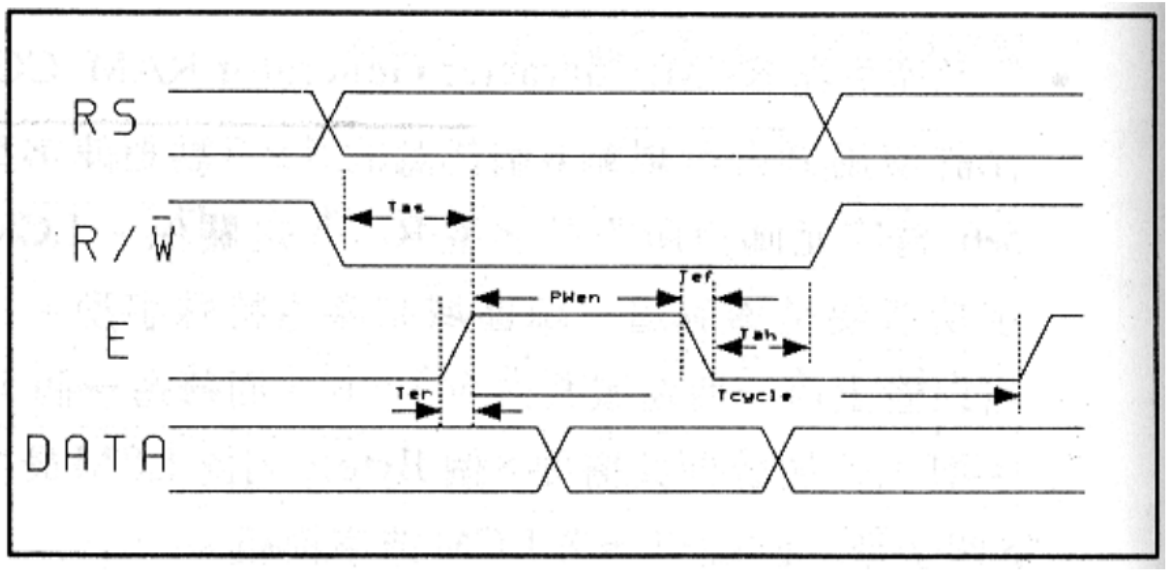
- 請在LCD上顯示自己的組別（兩位數），並且讓它從左到右依序顯示，每間格0.3秒到下一個動畫（請使用Systick_Handler）

```
int main() {
init();
SysTick_Config(1300000UL);
while (1) {}
return 0;
}
```

```
void SysTick_Handler() {
...
}
```

```
void init() {
for (int i = 0; i < 8; i++) {
TM_GPIO_Init(GPIOA, LCD_dataPin[i], TM_GPIO_Mode_OUT, TM_GPIO_OType_PP,
TM_GPIO_PuPd_UP, TM_GPIO_Speed_Medium);
}

TM_GPIO_Init(GPIOB, LCD_VSPin, TM_GPIO_Mode_OUT, TM_GPIO_OType_PP,
TM_GPIO_PuPd_UP, TM_GPIO_Speed_Medium);
TM_GPIO_Init(GPIOB, LCD_RSPin, TM_GPIO_Mode_OUT, TM_GPIO_OType_PP,
TM_GPIO_PuPd_UP, TM_GPIO_Speed_Medium);
TM_GPIO_Init(GPIOB, LCD_RWPin, TM_GPIO_Mode_OUT, TM_GPIO_OType_PP,
TM_GPIO_PuPd_UP, TM_GPIO_Speed_Medium);
TM_GPIO_Init(GPIOB, LCD_ENPin, TM_GPIO_Mode_OUT, TM_GPIO_OType_PP,
TM_GPIO_PuPd_UP, TM_GPIO_Speed_Medium);

init_LCD();
}
```

```
void init_LCD() {
WriteToLCD(0x38, 1); // Function Setting
WriteToLCD(0x06, 1); // Entering Mode
WriteToLCD(0x0C, 1); // Display on
WriteToLCD(0x01, 1); // Clear Screen
WriteToLCD(0x80, 1); // Move to top left
}
```




```
void WriteToLCD(int input, int isCmd)
{
...
TM_GPIO_SetPinLow(GPIOB, LCD_RSPin);
...
TM_GPIO_SetPinHigh(GPIOB, LCD_RSPin);

TM_GPIO_SetPinLow(GPIOB, LCD_RWPin);

for (int index = 0 ; index < 8 ; index++)
{
...
TM_GPIO_SetPinHigh(GPIOA, LCD_dataPin[index]);
...
TM_GPIO_SetPinLow(GPIOA, LCD_dataPin[index]);
...
}

TM_GPIO_SetPinHigh(GPIOB, LCD_ENPin);
delay_ms(10);
TM_GPIO_SetPinLow(GPIOB, LCD_ENPin);
delay_ms(10);
}
```

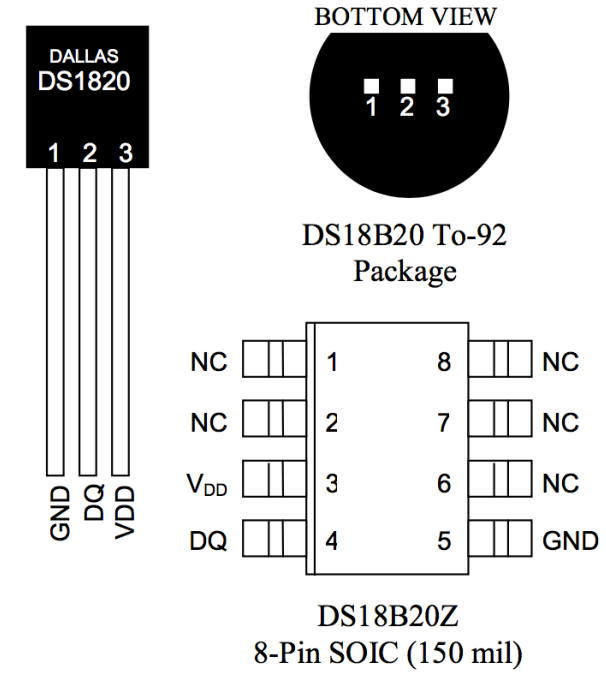
DS18B20

Programmable Resolution 1-Wire Digital Thermometer

Digital Thermometer

- Unique 1-Wire interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply
- range is 3.0V to 5.5V
- Zero standby power required
- Measures temperatures from -55°C to +125°C. Fahrenheit equivalent is -67°F to +25
- ±0.5°C accuracy from -10°C to +85°C
- Thermometer resolution is programmable from 9 to 12 bits
- Converts 12-bit temperature to digital word in 750 ms (max.)

PIN ASSIGNMENT

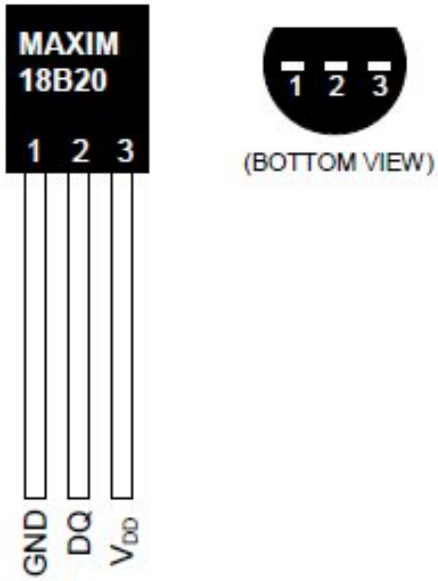


PIN DESCRIPTION

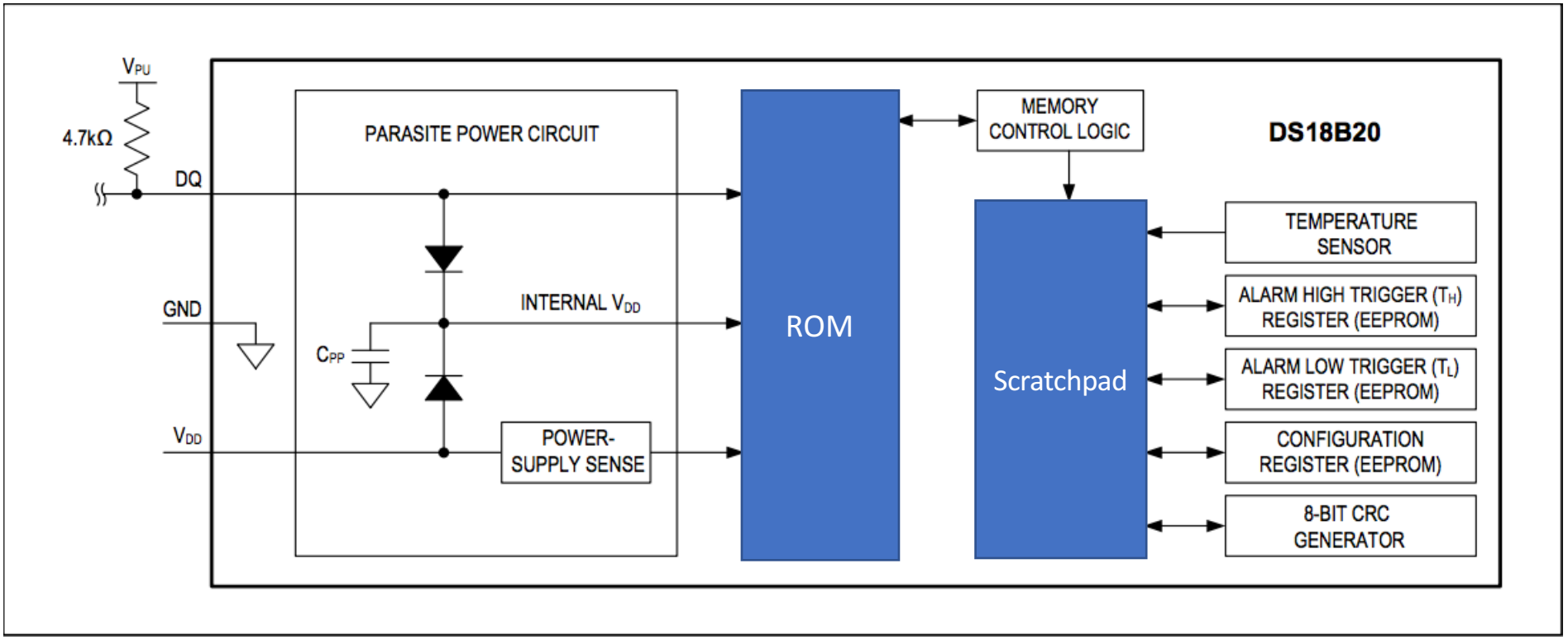
GND - Ground
DQ - Data In/Out
V_{DD} - Power Supply Voltage
NC - No Connect

One Wire Communication

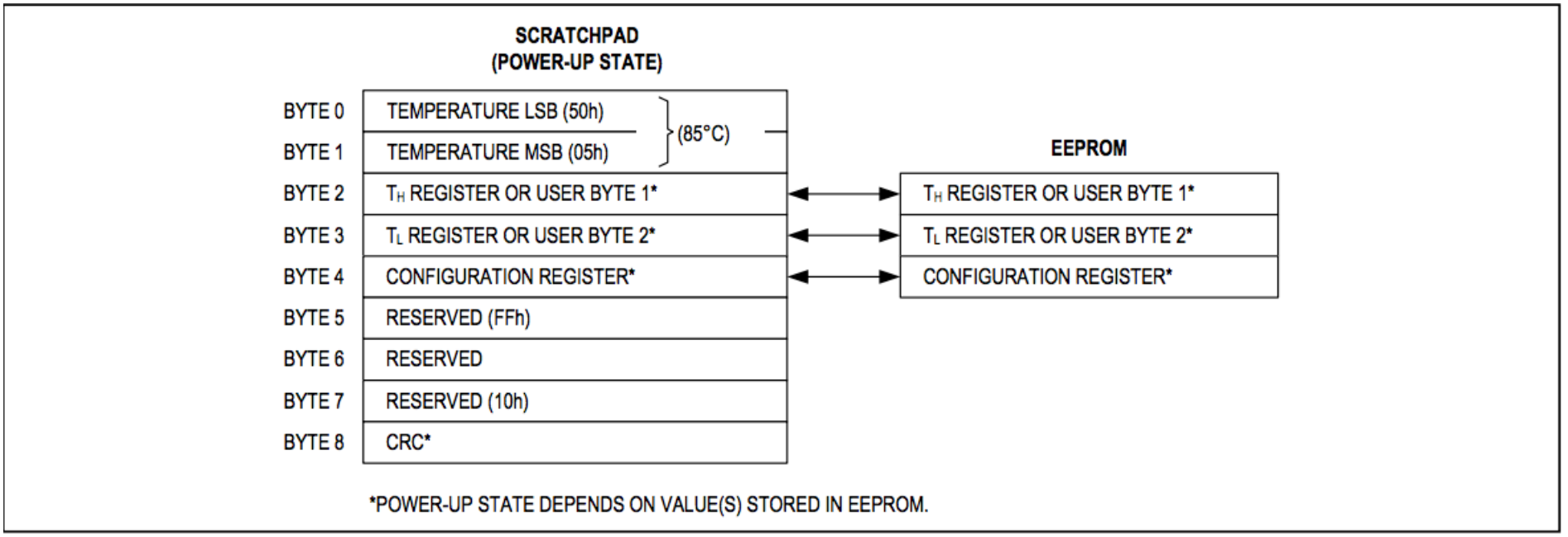
- 透過DQ以及時序的搭配來達到input以及output都透過一條線即可



Block Diagram



Scratchpad Memory



Temperature Representation

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	BIT 15	**BIT 14**	**BIT 13**	**BIT 12**	**BIT 11**	**BIT 10**	**BIT 9**	**BIT 8**
MS BYTE	S	S	S	S	S	2^6	2^5	2^4

S = SIGN

特性

- 64-bit ROM 儲存裝置的特別辨識碼（以便識別多個溫度計）
- Scratchpad 擁有 2-byte 溫度暫存器、警示暫存器（過高溫或是過低溫）及設定暫存器（精度設定），後三者為與 EEPROM 溝通所以是非揮發的。
- 溫度精度可以調成為 9 bits, 10 bits, 11 bits 及 12 bits
  - 分別為 0.5, 0.25, 0.125 及 0.0625
  - 預設為 12 bits

控制流程

- 每次要完成一個Operation都必須經過三個步驟
  - Initialization
  - ROM Command
  - DS18B20 Function Command

Initialization

- 透過 One Wire Protocol 告訴 DS18B20 我們準備要下指令給它了

ROM Command

- 若同時在匯流排上有很多 DS18B20 我們可以透過 ROM Command 來找尋特定的 DS18B20。
- 也可以用來偵測是否有任一 Device 有超溫或是溫度過低的現象。
  - Search ROM [0xF0]
  - Read ROM [0x33]
  - Match ROM [0x55]
  - Skip ROM [0xCC]
  - Alarm Search [0xEC]
- 若只有一個 DS18B20，我們可以使用 Skip ROM 這個指令

DS18B20 Function Command

- 這些指令是用來讀寫 Scratchpad Memory 的。
  - Convert T [0x44]
  - Write Scratchpad [0x4E]
  - Read Scratchpad [0xBE]
  - Copy Scratchpad [0x48]
  - Recall E^2 [0xB8]
  - Read Power Supply [0xB4]

Convert T[emperature]

- 開始讀取環境溫度並且利用ADC轉換
- 溫度將會放在 Scratchpad Memory 的前兩個 byte
- 不同精度需要不同的轉換時間
  - 9-bit -> 93.75 ms
  - 10-bit -> 187.5 ms
  - 11-bit -> 375 ms
  - 12-bit -> 750 ms
- 若轉換完成後，透過one wire 讀取一個 bit 會得到 1，反之則代表還正在進行溫度轉換。

Write Scratchpad

- 分別寫入 3 bytes 的資料到 Scratchpad 裡
  - T_H: 高溫 threshold
  - T_L: 低溫 threshold
  - Configuration:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
S	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Figure 5. T_H and T_L Register Format

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0	R1	R0	1	1	1	1	1

Figure 10. Configuration Register

Table 2. Thermometer Resolution Configuration

R1	R0	RESOLUTION (BITS)	MAX CONVERSION TIME	
0	0	9	93.75ms	(t_{CONV}/8)
0	1	10	187.5ms	(t_{CONV}/4)
1	0	11	375ms	(t_{CONV}/2)
1	1	12	750ms	(t_{CONV})

Copy Scratchpad

- 將 T_H, T_L 及 Configuration 從 Scratchpad Memory 寫到 EEPROM 裡面（非揮發性記憶體）。

Recall E^2

- 將 T_H, T_L 及 Configuration 從 EEPROM （非揮發性記憶體）讀到 Scratchpad Memory 裡面。

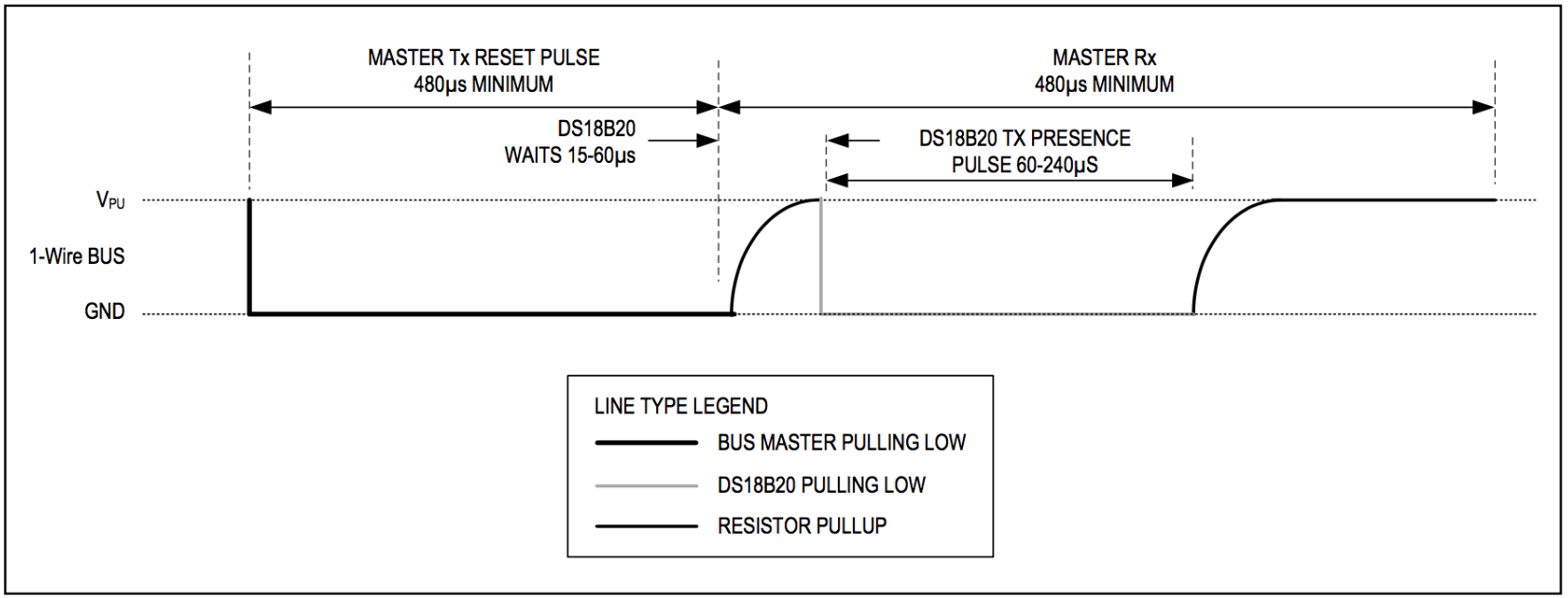
Read Power Supply

- 若接線方式為 Parasitic Mode (寄生方式)，則在 one wire 1-bit 讀取時會拿到 low
- 若接線方式為 External Power Supply，則在 one wire 1-bit 讀取時會拿到 high

One Wire Protocol (Initialization)

- Master 把電壓壓低(0)最少 480 us
- Master 釋放 wire（改成input）並且因為pull-up resistor所以會電壓拉高
- 當 DS18B20 偵測到正緣，它會等待 15us 到 60us 然後把電壓壓低 60us 到240us
- 當我們偵測到 DS18B20 有把電壓壓低，我們便知道 DS18B20 已經初始化了

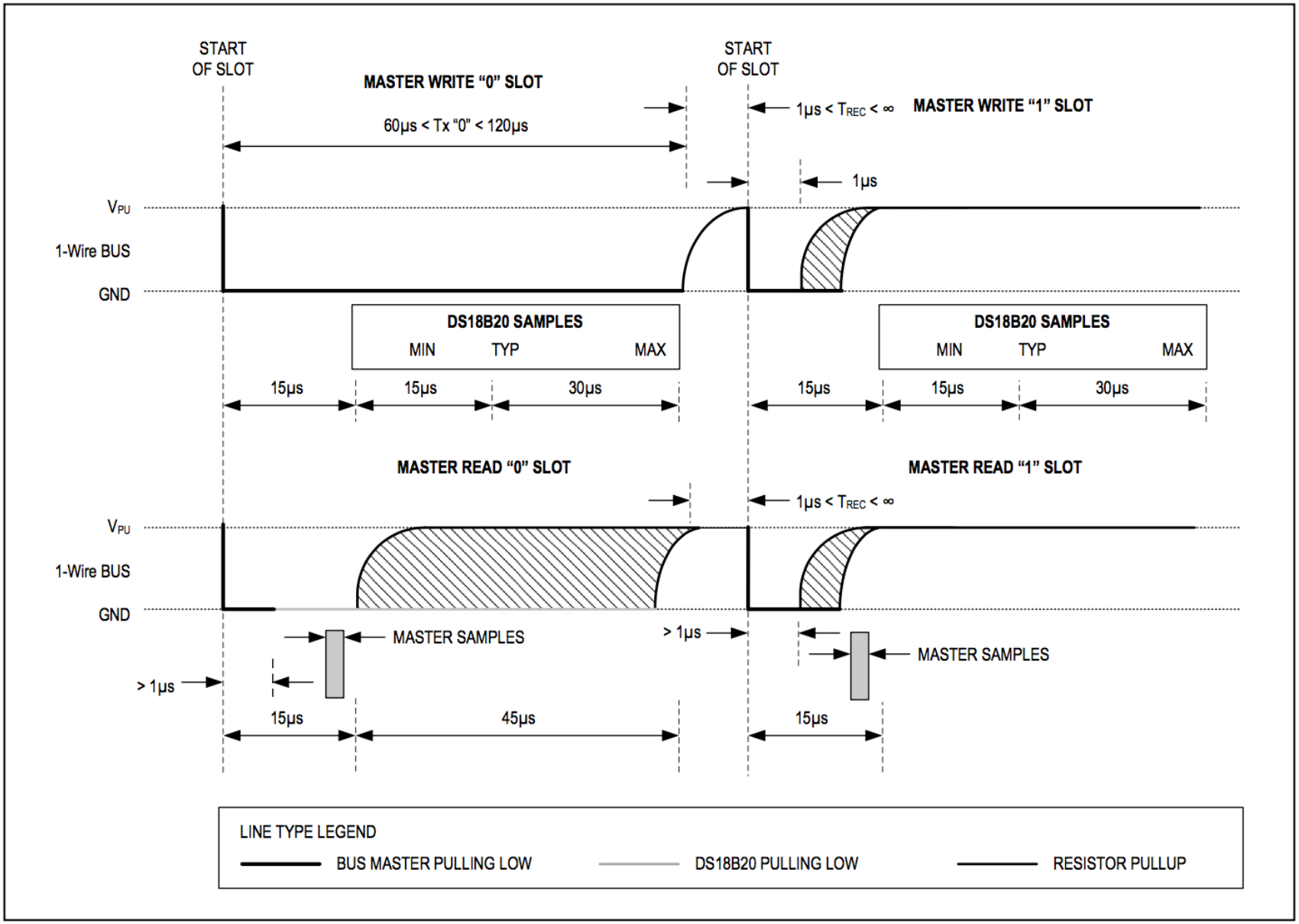
One Wire Protocol (Initialization)



```
uint8_t OneWire_Reset(OneWire_t* OneWireStruct) {
...

/* Line low, and wait 480us */
ONEWIRE_INPUT(OneWireStruct);
ONEWIRE_LOW(OneWireStruct);
ONEWIRE_OUTPUT(OneWireStruct);
ONEWIRE_DELAY(480);

/* Release line and wait for 70us */
ONEWIRE_INPUT(OneWireStruct);
ONEWIRE_DELAY(70);

/* Check bit value */
...

/* Delay for 410 us */
ONEWIRE_DELAY(410);

...
}
```

One Wire Protocol (Read 1 bit)

- 每一次的 read operation 最少都要有 60us
- 連續兩次之間要間格最少 1us
- 由 Master 把 DQ 拉低最少 1us 然後釋放DQ(改成input) 表示開始讀取。
- DS18B20 將會開始放 0 (low) 或是 1 (high) 在DQ上
- DS18B20 所送出來的 data 只會有效 15us（從Master把DQ拉低開始算），也就是說，Master 必須要在 15us 內把值讀進來


START
OF SLOT
MASTER WRITE "0" SLOT
60µs < Tx "0" < 120µs
START
OF SLOT
1µs < TREC < ∞
MASTER WRITE "1" SLOT
1µs
VPU
1-Wire BUS
GND
DS18B20 SAMPLES
MIN
TYP
MAX
15µs
15µs
30µs
DS18B20 SAMPLES
MIN
TYP
MAX
15µs
15µs
30µs
MASTER READ "0" SLOT
MASTER READ "1" SLOT
1µs < TREC < ∞
VPU
1-Wire BUS
GND
MASTER SAMPLES
> 1µs
> 1µs
MASTER SAMPLES
15µs
45µs
15µs
LINE TYPE LEGEND
BUS MASTER PULLING LOW
DS18B20 PULLING LOW
RESISTOR PULLUP

One Wire Protocol (Write 1 bit)

- 每一次的 write operation 最少都要有 60us
- 連續兩次之間要間格最少 1us
- Master Write 1
  - 由 Master 把 DQ 拉低
  - 15us 內釋放DQ(改成input)
- Master Write 0
  - 由 Master 把 DQ 拉低
  - 維持至少 60us

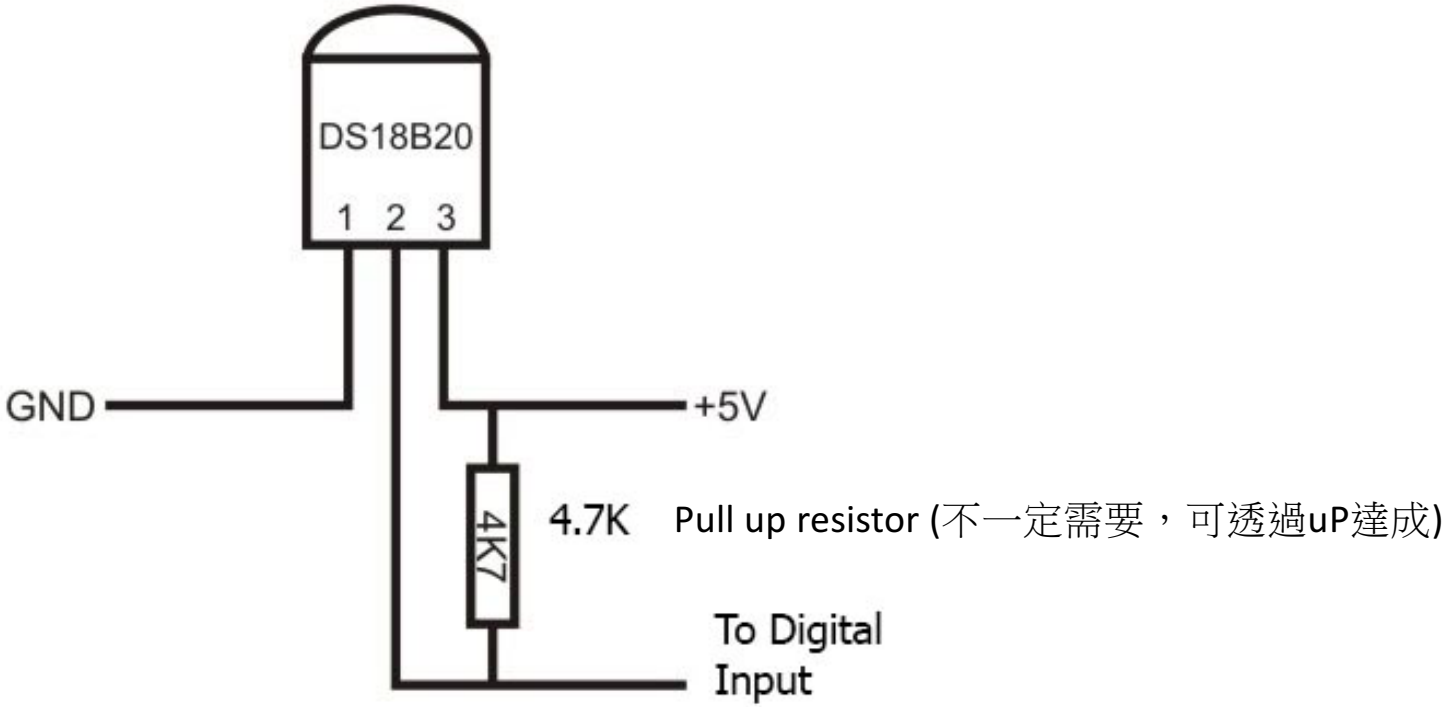
One Wire Protocol (Note)

- 所有的 Operation 都是透過負緣觸發（從高電位變成低電位的時候），因此請注意原本的電位。

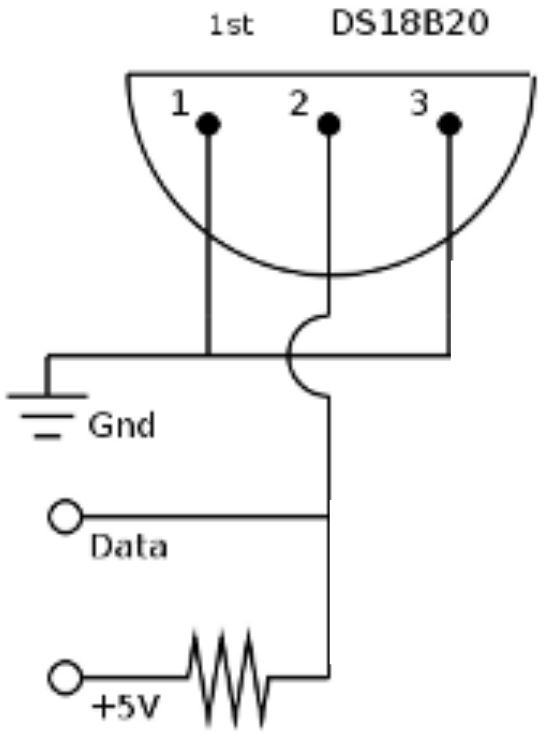
- 由於 DQ 有接一個 Pull-up resistor，若將 DQ 設置為 input 時，空角位的狀態會是 high，也就是當對方沒有給任何 input 時則為 high，當對方給 low 時，則為 low，對方給 high 時，還是 high

```
void OneWire_WriteBit(OneWire_t* OneWireStruct, uint8_t bit) {
ONEWIRE_INPUT(OneWireStruct);
if (bit) {
/* Set line low */
ONEWIRE_LOW(OneWireStruct);
ONEWIRE_OUTPUT(OneWireStruct);
...

/* Bit high */
ONEWIRE_INPUT(OneWireStruct);
...
} else {
/* Set line low */
ONEWIRE_LOW(OneWireStruct);
ONEWIRE_OUTPUT(OneWireStruct);
...
}
ONEWIRE_INPUT(OneWireStruct);
}
```

接線 (External Power Supply)



接線 (Parasitic Mode)



Reference

- http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf (Datasheet)