

數位系統技術



UART/USB 傳輸

Ren-Der Chen (陳仁德)

Department of Computer Science and
Information Engineering

National Changhua University of Education

E-mail: rdchen@cc.ncue.edu.tw

Spring, 2025

實驗內容

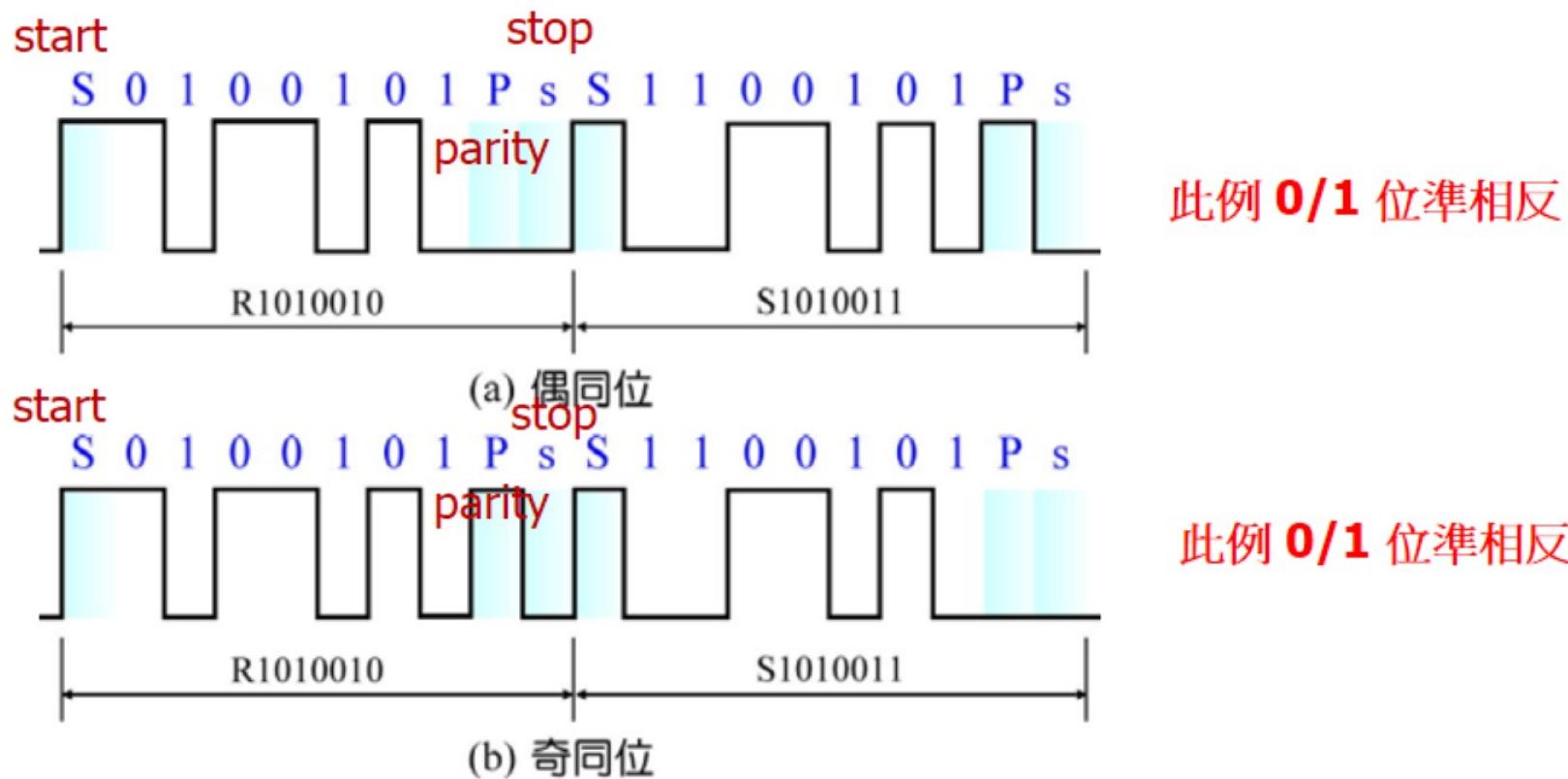
- 學習如何將DE10實驗板之資料透過FTDI232介面，以串列(serial)方式傳送至PC端，並由RealTerm觀察。
- 學習如何在PC端由RealTerm輸入資料，透過FTDI232介面，以串列(serial)方式傳送至DE10實驗板。
- UART (Universal Asynchronous Receiver/Transmitter):
將資料透過串列(serial)通訊和並列(parallel)通訊間作傳輸轉換。

串列位元流規範

| 信號框架 | 位元數 | 位元值(正邏輯) | 功 能 |
|--------------------|---------|----------|---|
| 標誌位元 Mask Bit | - | 1(Mask) | 表示電路停滯中。 |
| 起始位元 Start Bit | 1 | 0(Space) | 通知接收者的同步訊號，計時由1→0開始，每次接收皆由此位元取得同步。 |
| 資料位元 Data Bit | 5、6、7、8 | 依資料而定 | 要傳送或接收的資料位元，依系統目的的不同，資料長度有5、6、7、8位元可選用。 |
| 同位位元 Parity Bit | 1或0 | - | (None)為加快傳輸速度，通常會選不使用同位位元。 |
| | | 0或1 | 使用奇同位(Odd Parity)，即資料位元與同位位元所含的位元1的個數為奇數個；或偶同位(Even Parity)，即資料位元與同位位元所含的位元1的個數為偶數個，以為做位元檢查之用，可簡易判斷位元傳輸中是否有受干擾。 |
| | | 0(Space) | 此位元永遠設0(Space)，較少用。 |
| | | 1(Mask) | 此位元永遠設1(Mask)，較少用。 |
| 停止位元 Stop Bit | 1、1.5、2 | 1(Mask) | 此次傳輸已結束，一方面做為下一筆傳輸之間隔符號。其中1.5位元是為資料長度為5時所專用。 |

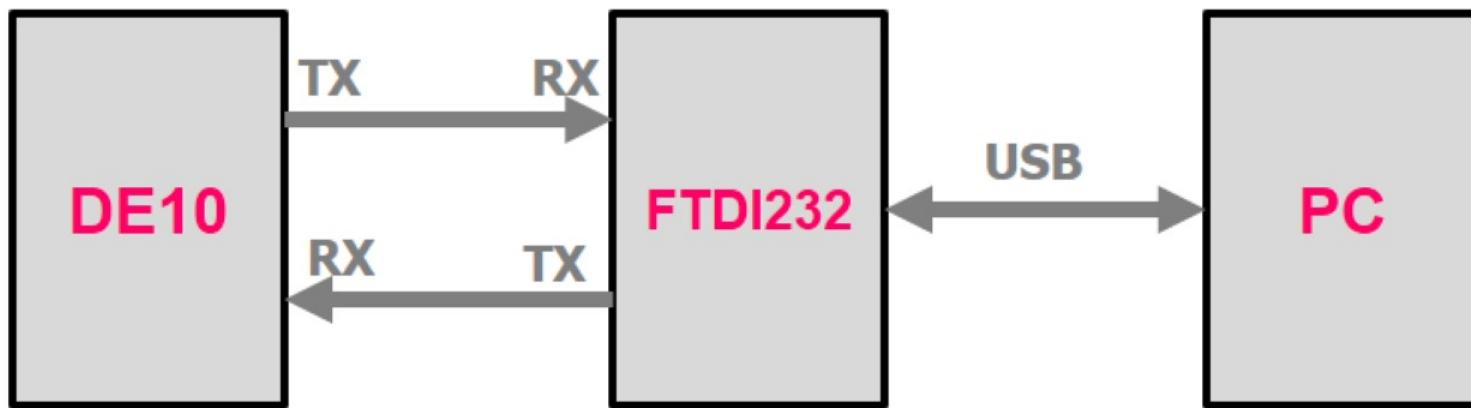
資料傳輸

- 通訊協定: 速率(baud rate)-資料位元數-同位位元-停止位元數
 - 以 **9600(bps)-7-x-1** 為通訊協定 (x 為同位位元) , 傳送"R" (1010010)及"S"(1010011) , 由低位元開始傳輸 。



DE10-FTDI232-PC 資料對傳

資料以串列方式由
DE10傳至FTDI232



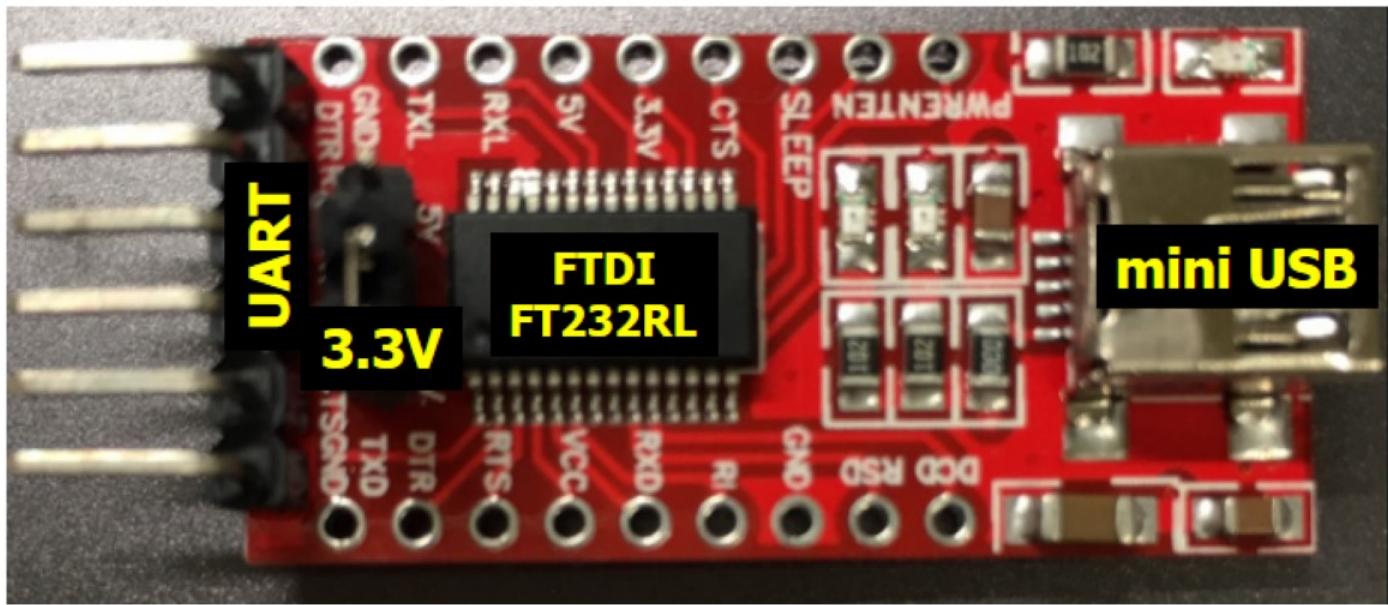
資料以串列方式由
FTDI232傳至DE10

資料透過USB介面式在
FTDI232與PC間互傳

FTDI232

- USB to serial UART interface

DTR
RX
TX
VCC
CTS
GND



DTR: data terminal ready

RX: receive data

TX: transmit data

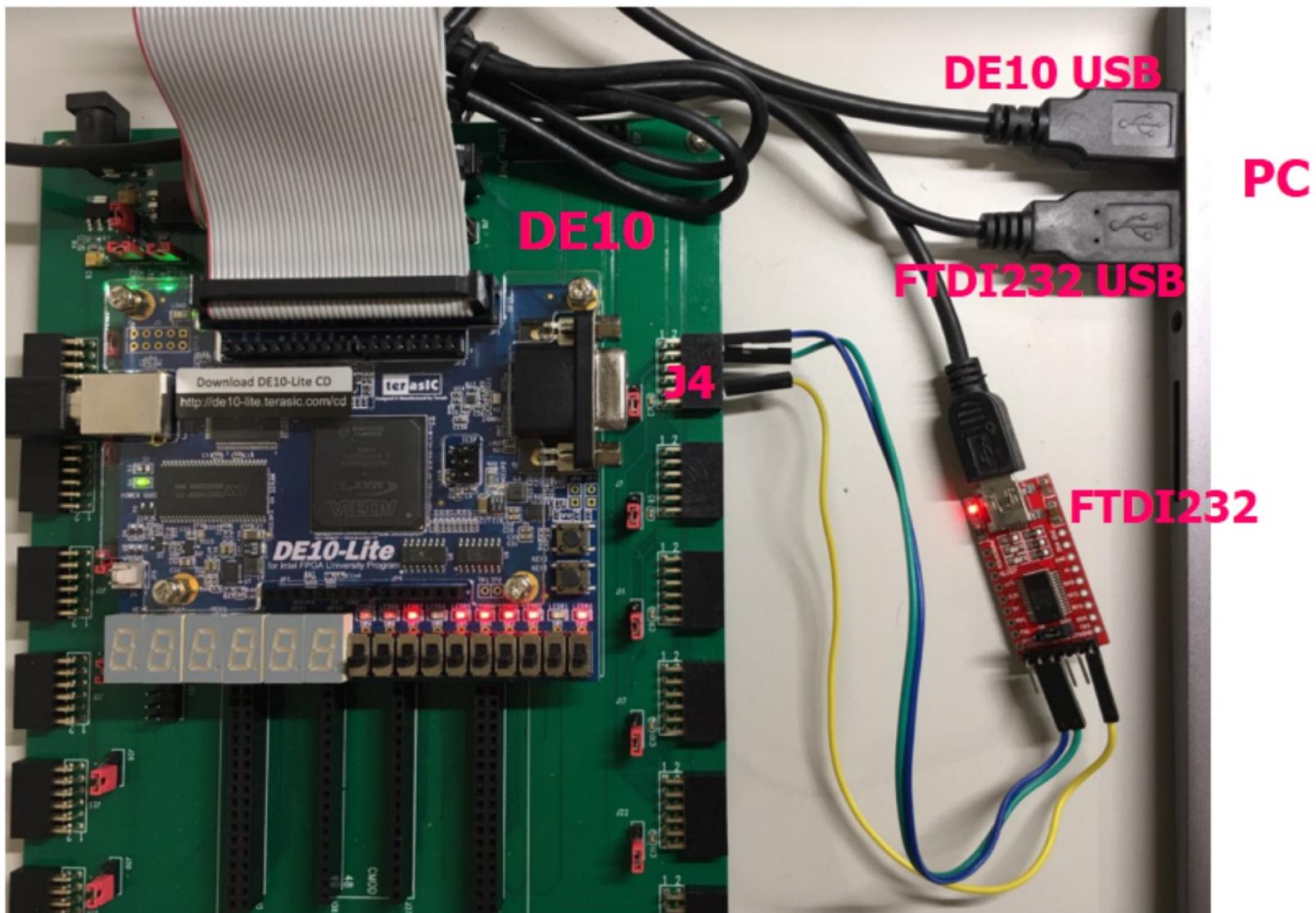
VCC: power supply (3.3V)

CTS: clear to send

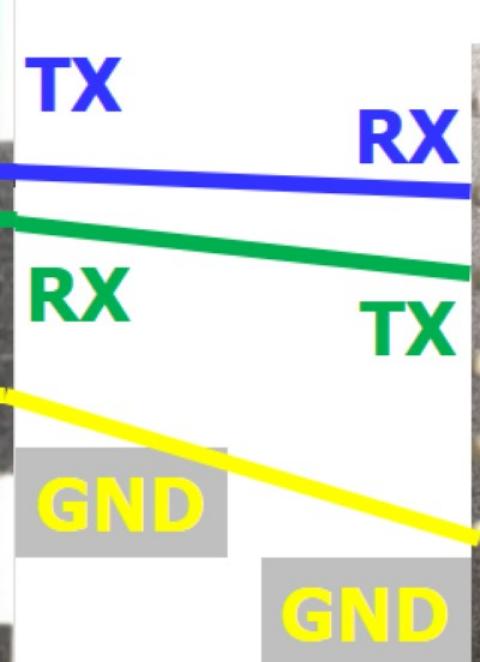
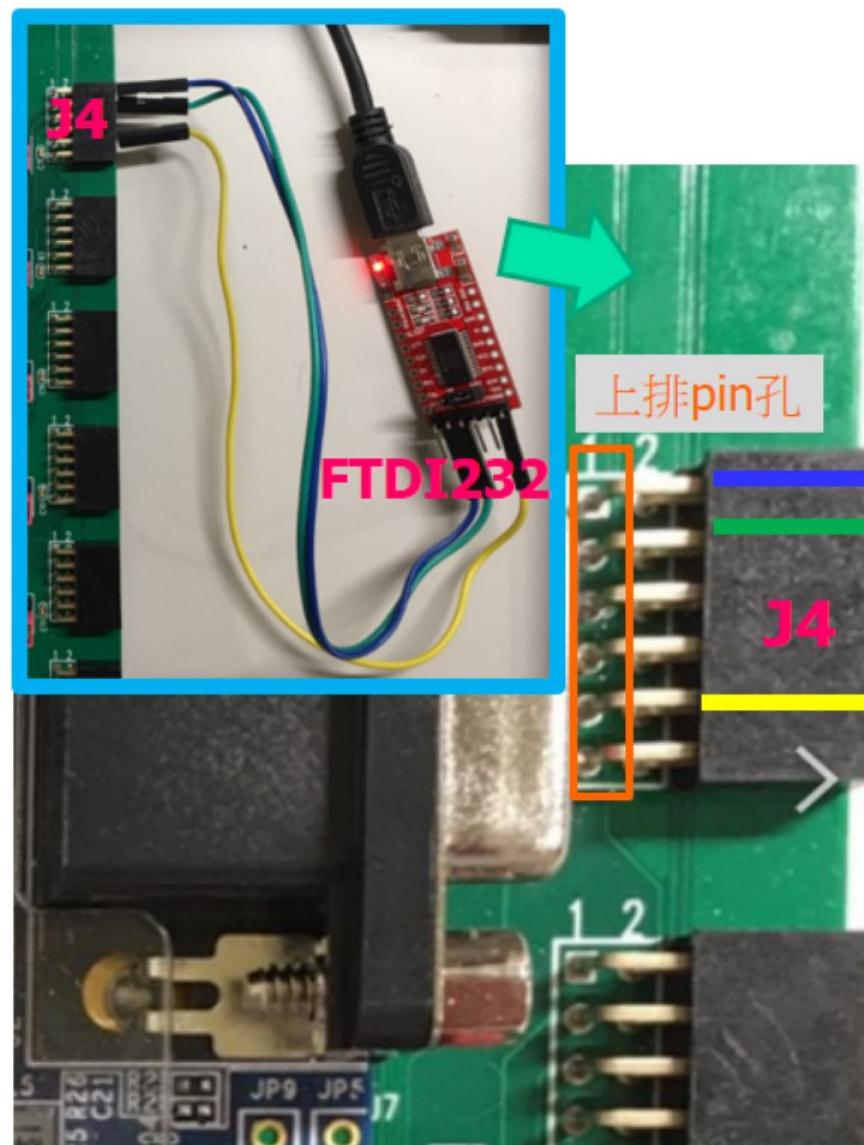
GND: ground



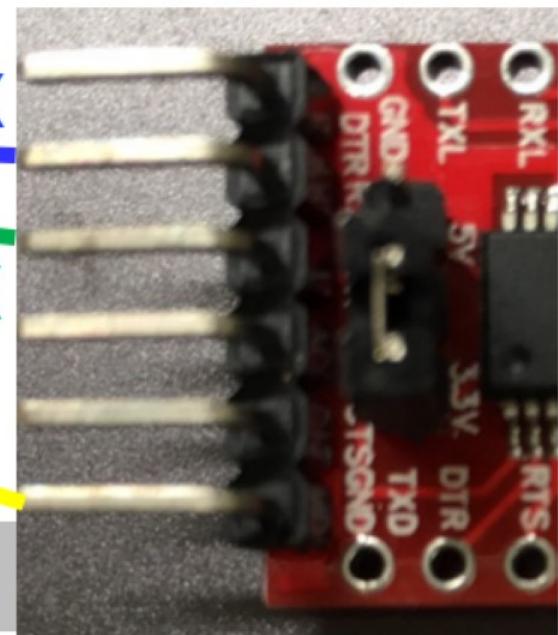
系統架構



細部接線



FTDI232



RealTerm (1/2)

- <https://sourceforge.net/projects/realterm/>
- 串列信號終端機工具，安裝於電腦上，傳送與接收串列信號。

Home / Browse / Development / Serial / RealTerm: Serial/TCP Terminal



RealTerm: Serial/TCP Terminal

Serial and TCP terminal for engineering and debugging
Brought to you by: crun

★★★★★ 30 Reviews

Downloads: 3,757 This Week

 Download Get Updates Share This

RealTerm (2/2)

裝置管理員

檔案(F) 動作(A) 檢視(V) 說明(H)



rdchen_asus_nb

> IDE ATA/ATAPI 控制器
> Intel(R) Dynamic Platform anc

> 人性化介面裝置

> 生物特徵辨識裝置

> 列印佇列

> 印表機

> 存放控制器

> 安全性裝置

> 系統裝置

> 相機

> 音效、視訊及遊戲控制器

> 音訊輸入與輸出

> 處理器

> 軟體元件

> 軟體裝置

> 通用序列匯流排控制器

> 連接埠 (COM 和 LPT)

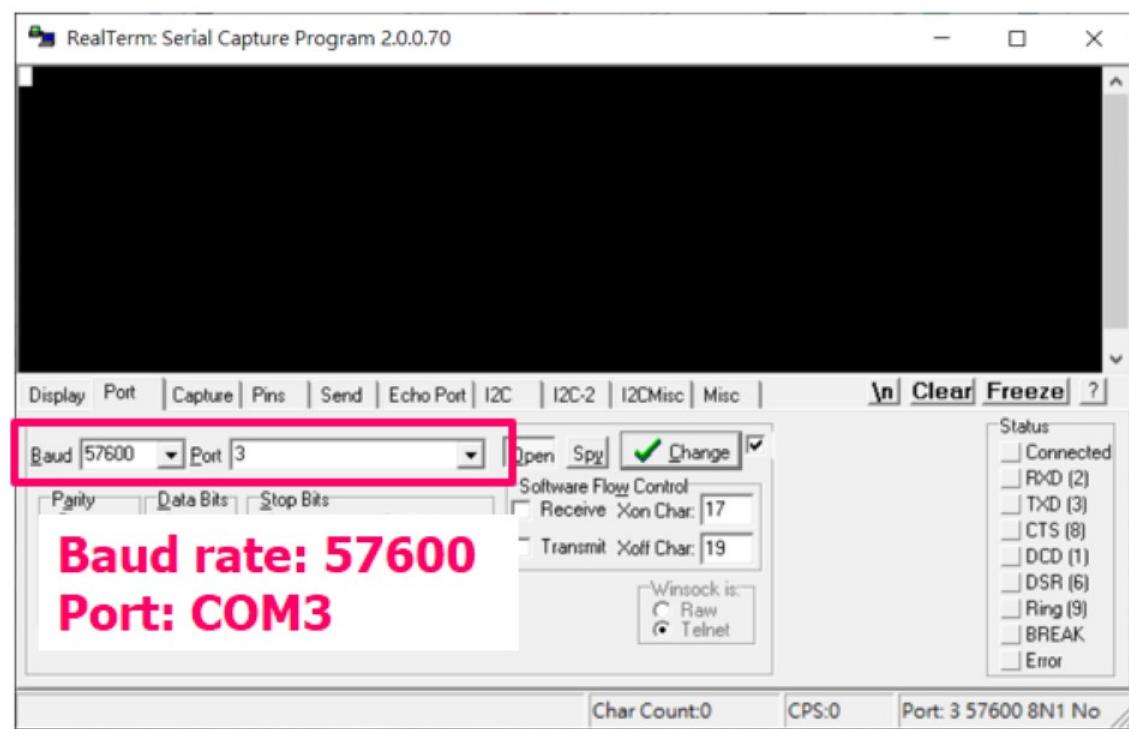
USB Serial Port (COM3)

韌體

滑鼠及其他指標裝置



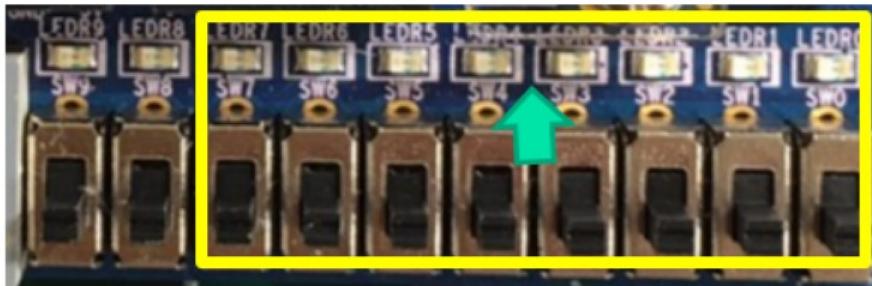
要用此身分方能執行！



實驗一：DE10 -> FTDI232 -> PC

- 由DE10實驗板8個switch之上下位置表示輸入之8-bit信號值(上為1，下為0) · 8個led燈顯示輸入之信號值 · 亮燈代表1。
- 將輸入之信號透過FTDI232以串列方式傳送至PC · 由Realterm觀察接收到之信號。

Output



Input



Output

Design 1: UART_TX (1/5)

- 以 **57600-8-x-1** 為通訊協定 (無同位位元)。
- 將八個bit輸入信號 **i_TX_Byte[7:0]** 以序列方式傳至輸出 **o_TX_Bit**，每一個clock傳一個bit。

```
module UART_TX (
    input   i_clk,
    input   i_rst,
    input [7:0] i_TX_Byte,
    output  o_TX_Bit
);
    parameter [1:0] TX_START_ST = 2'd0;
    parameter [1:0] TX_DATA_ST  = 2'd1;
    parameter [1:0] TX_STOP_ST  = 2'd2;
```

信號傳輸時，電路之狀態

1. TX_START_ST: 傳送start信號
2. TX_DATA_ST: 傳送資料信號
3. TX_STOP_ST: 傳送stop信號

Design 1: UART_TX (2/5)

```
reg [1:0] r_SM_Main = TX_START_ST;  
reg [2:0] r_Byte_Idx = 3'd0;  
  
reg r_TX_Bit = 1'b1;  
assign o_TX_Bit = r_TX_Bit;
```

1. r_SM_Main: 紀錄電路之狀態
2. r_Byte_Idx: 資料信號之bit index
3. r_TX_Bit: 傳送至輸出o_TX_Bit之信號

Design 1: UART_TX (3/5)

```
// TX state machine
always @(posedge i_clk or posedge i_rst) begin
    if (i_rst == 1'b1) begin
        r_Byte_Idx = 3'd0;
        r_TX_Bit = 1'b1;
        r_SM_Main = TX_START_ST;
    end
    else begin
        case (r_SM_Main)
            TX_START_ST: begin
                r_Byte_Idx = 3'd0;          啟動資料傳輸 ,
                r_TX_Bit = 1'b0 // start  start信號設為0
                r_SM_Main = TX_DATA_ST;
            end // TX_START_ST      跳至資料傳輸狀態
        endcase
    end
end
```

Design 1: UART_TX (4/5)

```
TX_DATA_ST: begin
    r_TX_Bit = i_TX_Byte [ ] ; // send
    // 傳送資料信號之某個bit
    if (r_Byte_Idx < 3'd7) begin
        r_Byte_Idx = r_Byte_Idx + 1;
        r_SM_Main = [ ];
    end
    else
        r_SM_Main = [ ];
end // TX_DATA_ST
// 若資料信號尚未傳完，則在下一週期繼續傳送下一個bit，電路狀態維持在TX_DATA_ST
若資料信號已傳完 (r_Byte_Idx==3'd7)，則在下一週期進入TX_STOP_ST狀態
```

Design 1: UART_TX (5/5)

```
TX_STOP_ST: begin
    r_TX_Bit =  // stop
    r_SM_Main = 
end // TX_STOP_ST

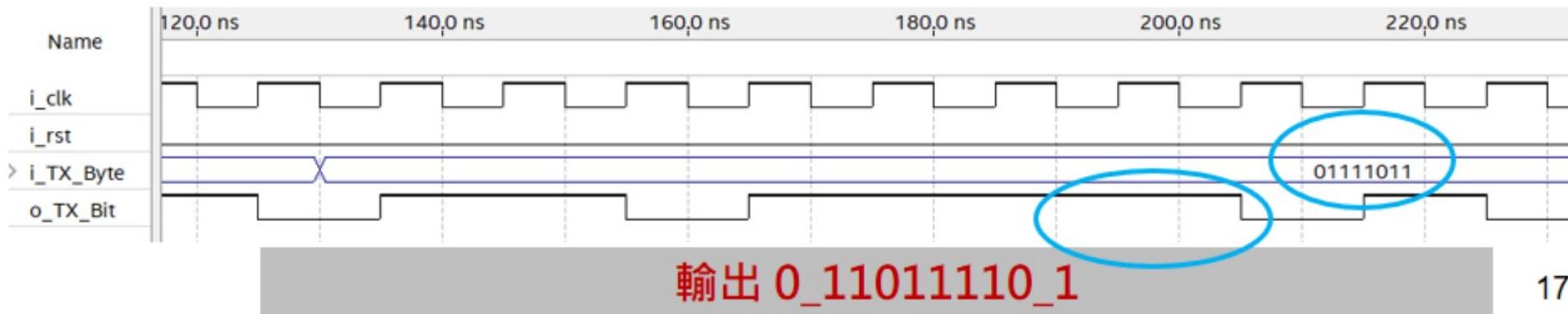
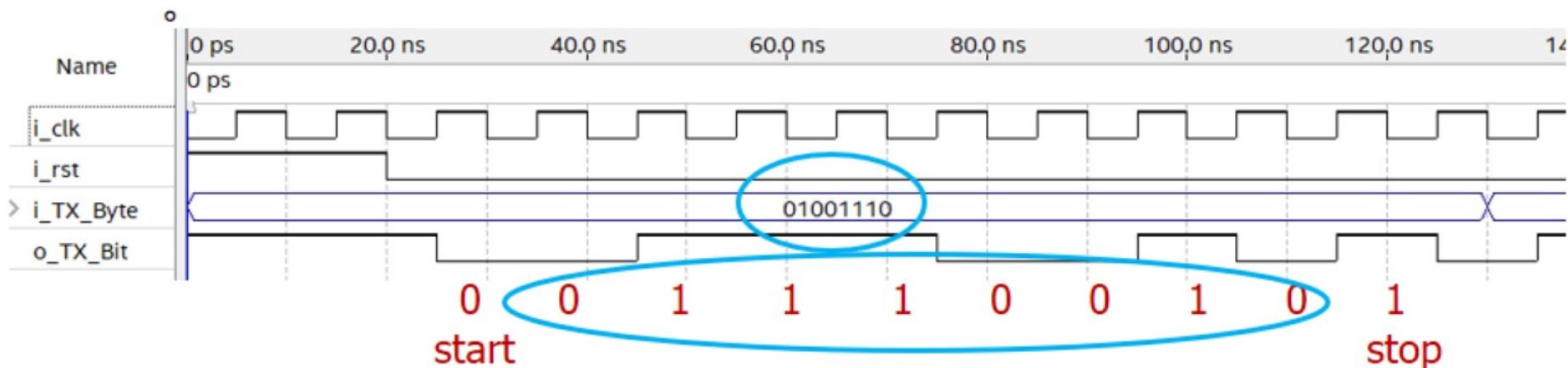
default:
    r_SM_Main = TX_START_ST;

endcase
end
end
```

1. 結束資料傳輸，stop信號設為1
2. 重新回到TX_START_ST狀態，開啟下一次資料傳輸

Design 1: UART_TX Simulation Results

- 整體模擬時間(End time)設為300 ns，輸入時脈(clk)之週期設為10 ns，一開始0~20 ns，rst之值設為1，之後回到0



Design 2: UART_TX_top (1/2)

```
module UART_TX_top (
    input i_clk,
    input i_rst_b,
    input [7:0] i_TX_Byte,
    output [7:0] o_TX_Byte_b,
    output o_TX_Bit
);
```

1. i_TX_Byte[7:0]: 接至switch的8-bit輸入信號
2. o_TX_Byte_b[7:0]: 接至led的8-bit輸出信號
3. o_TX_Bit: 接至FTDI232接收端之輸出信號

Design 2: UART_TX_top (2/2)

```
assign o_TX_Byte_b = i_TX_Byte;  
  
wire rst = ~i_rst_b;  
wire clk_div;  
  
freq_div_n #(868, 10) m0  
  
wire r_TX_Bit;  
UART_TX m1  
assign o_TX_Bit = r_TX_Bit;  
endmodule
```

DE10實驗板之工作時脈為50 MHz，假設資料傳輸之 baud rate為57,600，則實驗板時脈必須先經過除頻， $50,000,000/57,600 = 868$ ，因此除頻電路需要10個bits

Pin Assignments

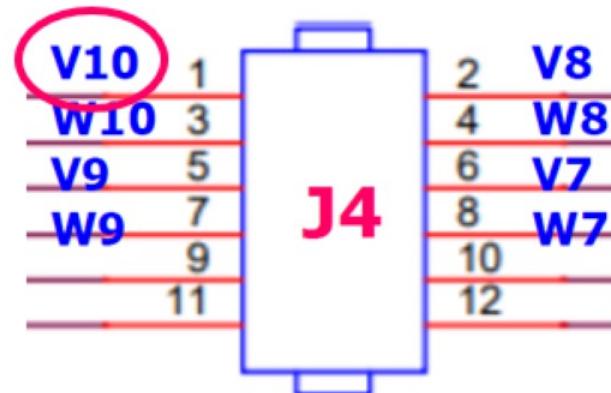
i_clk

Clocks: 50 MHZ

CLK1: P11

CLK2: N14

o_TX_Bit



i_rst_b



KEY0: B8 Push: 0

KEY1: A7 Not push: 1

i_TX_Byte



SW7 SW6 SW5 SW4 SW3 SW2 SW1 SW0
A14 A13 B12 A12 C12 D12 C11 C10

o_TX_Byte_b



1: LED on
0: LED off

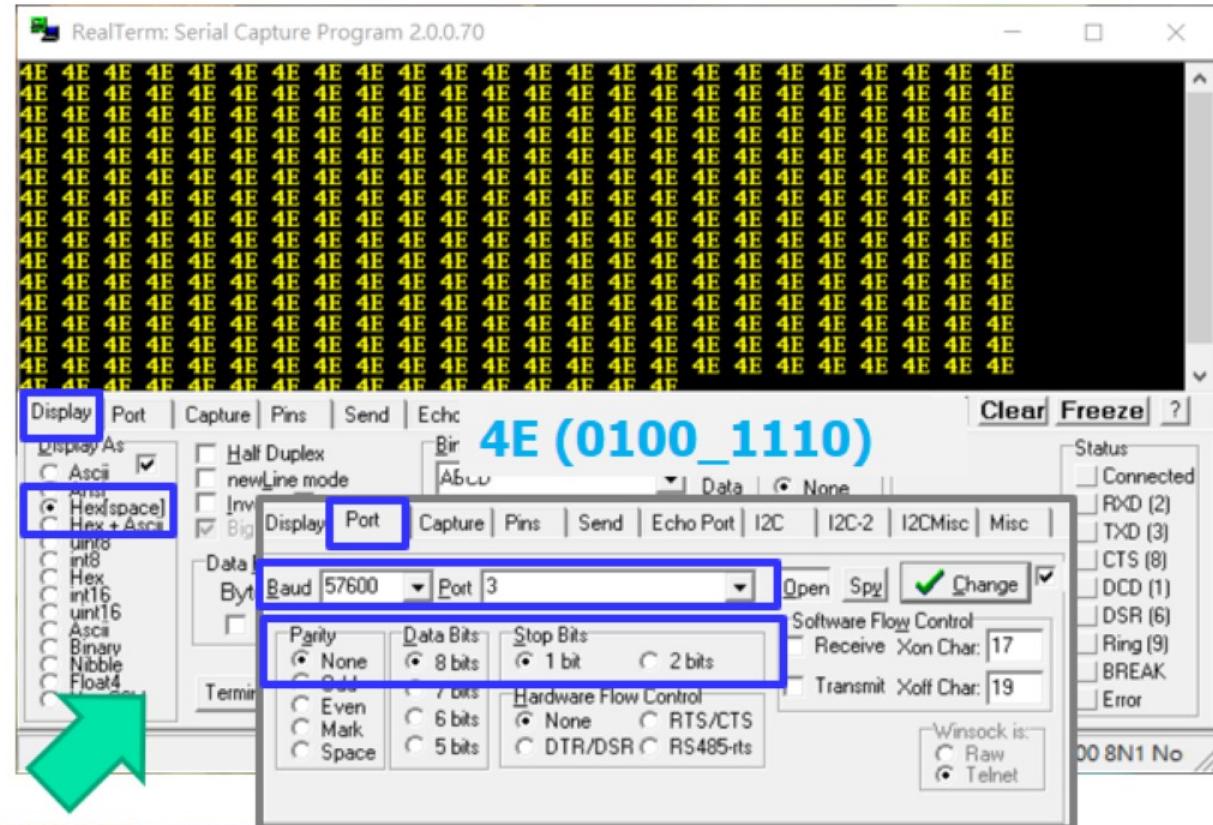
LED7 LED6 LED5 LED4 LED3 LED2 LED1 LED0
D14 E14 C13 D13 B10 A10 A9 A8

實驗結果

注意1：開啟Qartus進行電路compile及program時，要將RealTerm關閉。電路順利燒錄後，再執行RealTerm。

注意2：FTDI232子板上之UART介面，要使用跳線(jumper)將電壓調整為3.3V，預設電壓可能為5V。

實驗板端FPGA

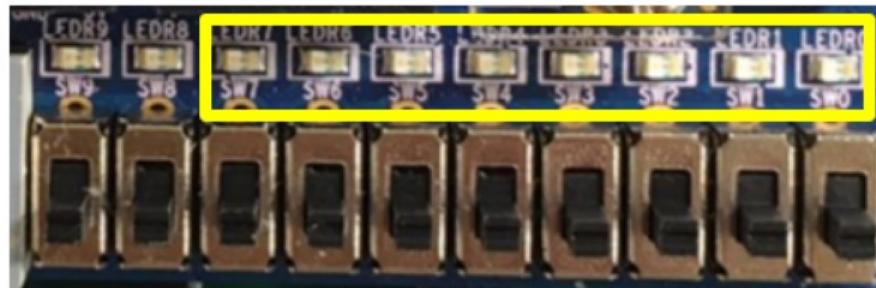


電腦端Realterm

實驗二：PC -> FTDI232 -> DE10

- 由PC端之Realterm送出8-bit信號值，透過FTDI232以串列方式傳送至DE10實驗板。
- 透過實驗板上8個led燈，觀察所接收之值，亮燈代表1。

Output



Input

Design 3: UART_RX (1/5)

- 以 **57600-8-x-1** 為通訊協定 (無同位位元)。
- 由輸入信號 **i_RX_Bit** 以序列方式接收8-bit信號，每一個 **clock** 接收一個bit，再傳至輸出信號 **o_RX_Byte[7:0]**。

```
module UART_RX (
    input   i_clk,
    input   i_rst,
    input   i_RX_Bit,
    output [7:0] o_RX_Byte
);
    parameter [1:0] RX_START_ST = 2'd0;
    parameter [1:0] RX_DATA_ST  = 2'd1;
    parameter [1:0] RX_STOP_ST  = 2'd2;
```

信號傳輸時，電路之狀態

1. RX_START_ST: 接收start信號
2. RX_DATA_ST: 接收資料信號
3. RX_STOP_ST: 接收stop信號

Design 3: UART_RX (2/5)

```
reg [1:0] r_SM_Main = RX_START_ST;  
reg [2:0] r_Byte_Idx = 3'd0;  
  
reg [7:0] r_RX_Byte = 8'd0;  
assign o_RX_Byte = r_RX_Byte;
```

1. r_SM_Main: 紀錄電路之狀態
2. r_Byte_Idx: 資料信號之bit index
3. r_RX_Byte: 傳送至輸出o_RX_Byte之信號

Design 3: UART_RX (3/5)

```
// RX state machine
always @(posedge i_clk or posedge i_rst) begin
    if (i_rst == 1'b1) begin
        r_Byte_Idx = 3'd0;
        r_RX_Byte = 8'd0;
        r_SM_Main = RX_START_ST;
    end
    else begin
        case (r_SM_Main)
            RX_START_ST: begin
                r_Byte_Idx = 3'd0;          // 若接收到start信號為0，資料傳輸啟動
                if (i_RX_Bit == [ ]) // start
                    r_SM_Main = [ ];      // 跳至資料傳輸狀態
                else
                    r_SM_Main = [ ];      // 否則維持在等待start的狀態
            end // RX_START_ST
        end
    end
end
```

Design 3: UART_RX (4/5)

```
RX_DATA_ST: begin  
    r_RX_Byte[r_Byte_Idx] = [ ] // receive  
  
    if (r_Byte_Idx < 3'd7) begin  
        r_Byte_Idx = r_Byte_Idx + [ ]  
        r_SM_Main = [ ]  
    end  
    else  
        r_SM_Main = [ ]  
end // RX_DATA_ST
```

接收資料信號之某個bit

若資料信號尚未傳完，則在下一週期繼續接收下一個bit，電路狀態維持在RX_DATA_ST

若資料信號已傳完($r_Byte_Idx == 3'd7$)，則在下一週期進入RX_STOP_ST狀態

Design 3: UART_RX (5/5)

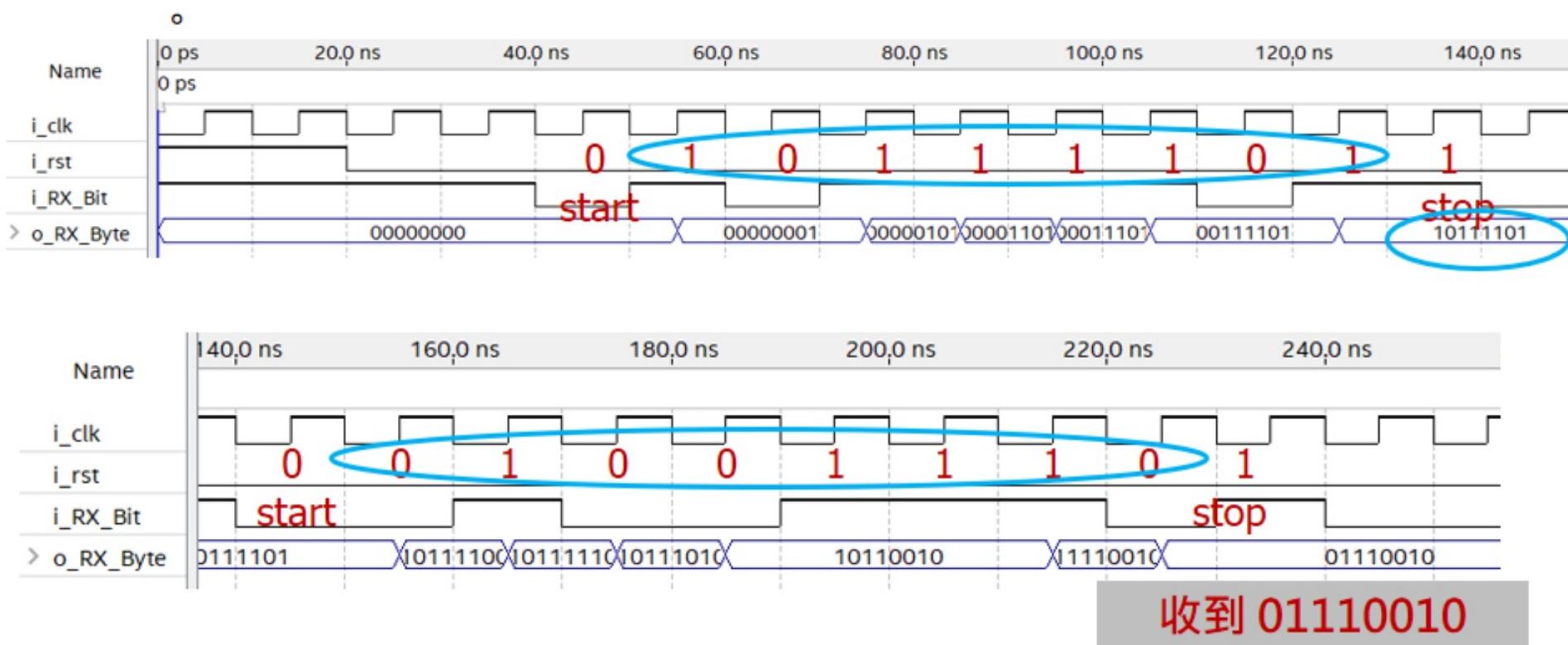
結束資料傳輸，重新回到RX_START_ST
狀態，等待下一次傳輸

```
    RX_STOP_ST: // stop
        r_SM_Main = [REDACTED]

    default :
        r_SM_Main = RX_START_ST;
    endcase
end
endmodule
```

Design 3: UART_RX Simulation Results

- 整體模擬時間(End time)設為300 ns，輸入時脈(clk)之週期設為10 ns，一開始0~20 ns，rst之值設為1，之後回到0



Design 4: UART_RX_top (1/2)

```
module UART_RX_top (
    input i_clk,
    input i_rst_b,
    input i_RX_Bit,
    output [7:0] o_RX_Byte_b
);
```

1. i_RX_Bit: 接至FTDI232傳送端之輸入信號
2. o_RX_Byte_b[7:0]: 接至led的8-bit輸出信號

Design 4: UART_RX_top (2/2)

```
wire rst = ~i_rst_b;
wire clk_div;

freq_div_n #(868, 10) m0
    begin
        wire [7:0] r_RX_Byte;
        UART_RX m1
            begin
                assign o_RX_Byte_b = r_RX_Byte;
            end
    end
endmodule
```

DE10實驗板之工作時脈為50 MHz，假設資料傳輸之 baud rate為57,600，則實驗板時脈必須先經過除頻， $50,000,000/57,600 = 868$ ，因此除頻電路需要10個bits

Pin Assignments

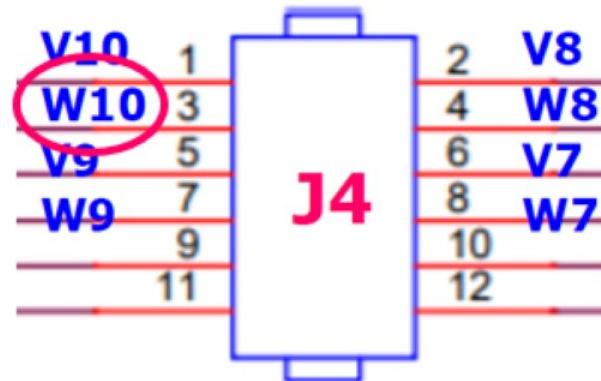
i_clk

Clocks: 50 MHZ

CLK1: P11

CLK2: N14

i_RX_Bit



i_RST_b



KEY0: B8 Push: 0

KEY1: A7 Not push: 1

o_TX_Byte_b



1: LED on
0: LED off

LED7 LED6 LED5 LED4 LED3 LED2 LED1 LED0

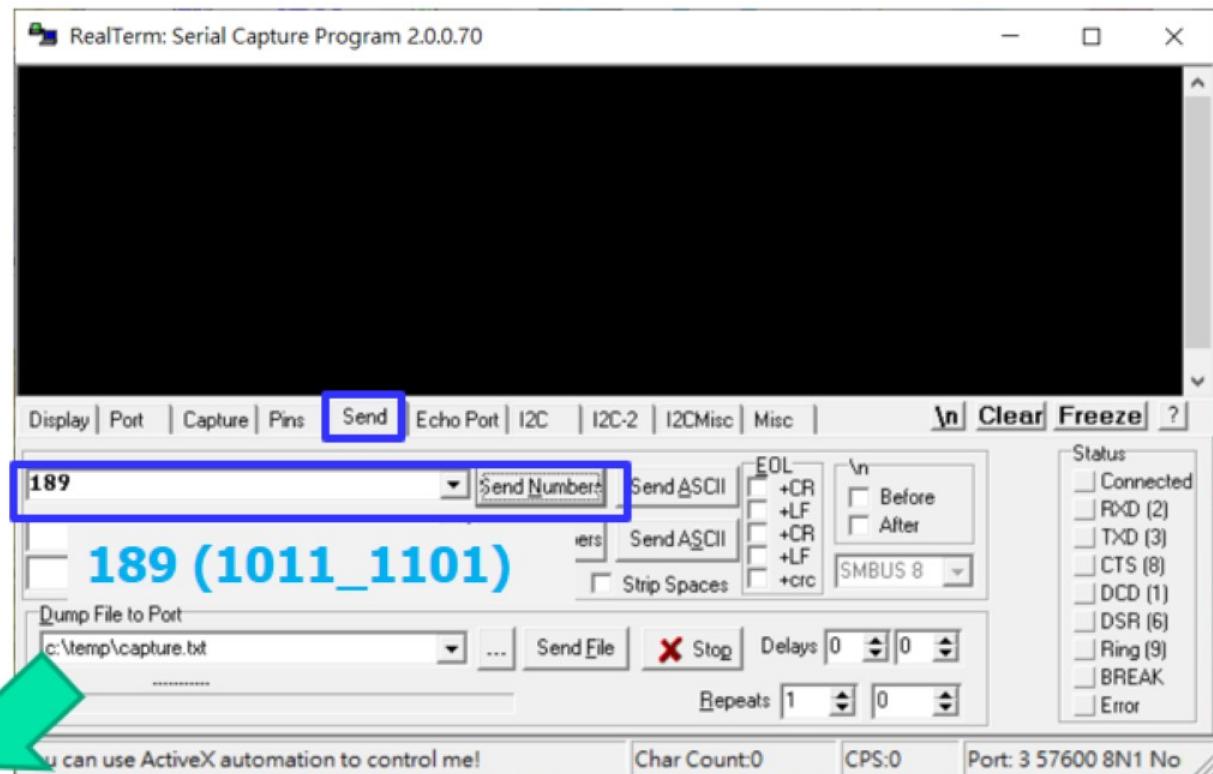
D14 E14 C13 D13 B10 A10 A9 A8

實驗結果

注意1：開啟Qartus進行電路compile及program時，要將RealTerm關閉。電路順利燒錄後，再執行RealTerm。

注意2：FTDI232子板上之UART介面，要使用跳線(jumper)將電壓調整為3.3V，預設電壓可能為5V。

實驗板端FPGA



電腦端Realterm

實驗結果驗收

- 請老師或助教驗收電路於實驗板顯示之結果