第05次實習課

學生：林培瑋

2024 Advanced Mixed-Operation System (AMOS) Lab.

Tamkang University
Department of Electrical and Computer Engineering
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan (R.O.C.)

目錄

微處理機概論 – HW1

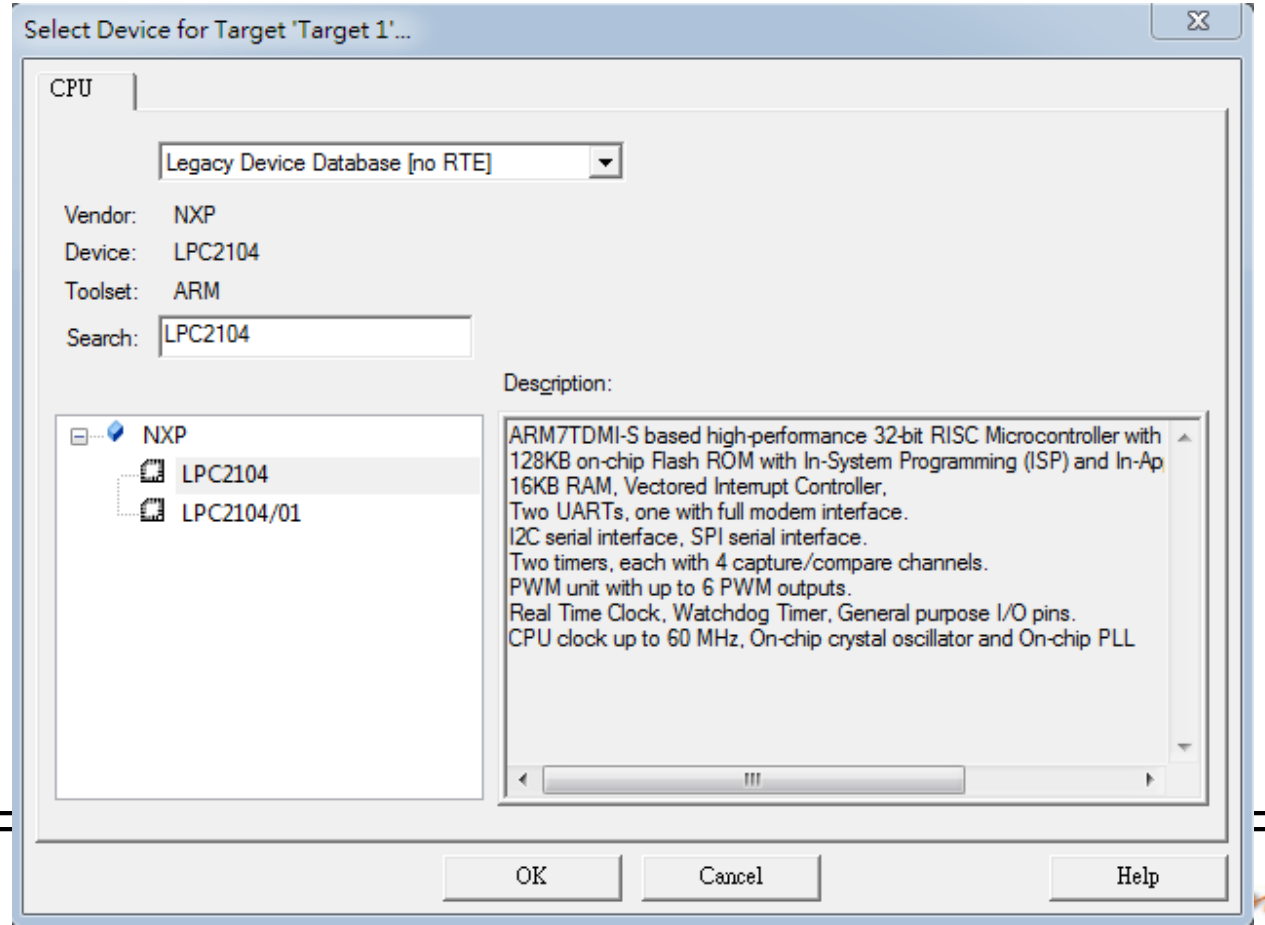
繳交期限：4/9 23:59

(用word檔上傳至iclass)

請提前上傳，以免iclass塞車

作業題目

- 課本p.348-349
- ARM7TDMI: LPC2104



程式(1/3)

```
            AREA    UARTDEMO, CODE, READONLY
PINSEL0     EQU     0xE002C000        ; controls the function of pins
U0START     EQU     0xE000C000        ; start of UART0 register
LCR0        EQU     0xC               ; line control register for UART0
LSR0        EQU     0x14              ; line status register for UART0
RAMSTART    EQU     0x40000000        ; start of onboard RAM for 2104
            ENTRY
start
            LDR     sp, =RAMSTART     ; set up stack pointer
            BL      UARTConfig        ; initialize/configure UART0
            LDR     r1, =CharData     ; starting address of characters
Loop
            LDRB    r0, [r1], #1      ; load character, increment address
            CMP     r0, #0            ; null terminated?
            BLNE    Transmit          ; send character to UART
            BNE     Loop              ; contiune if not a '0'
done        B       done              ; otherwise we're done
```

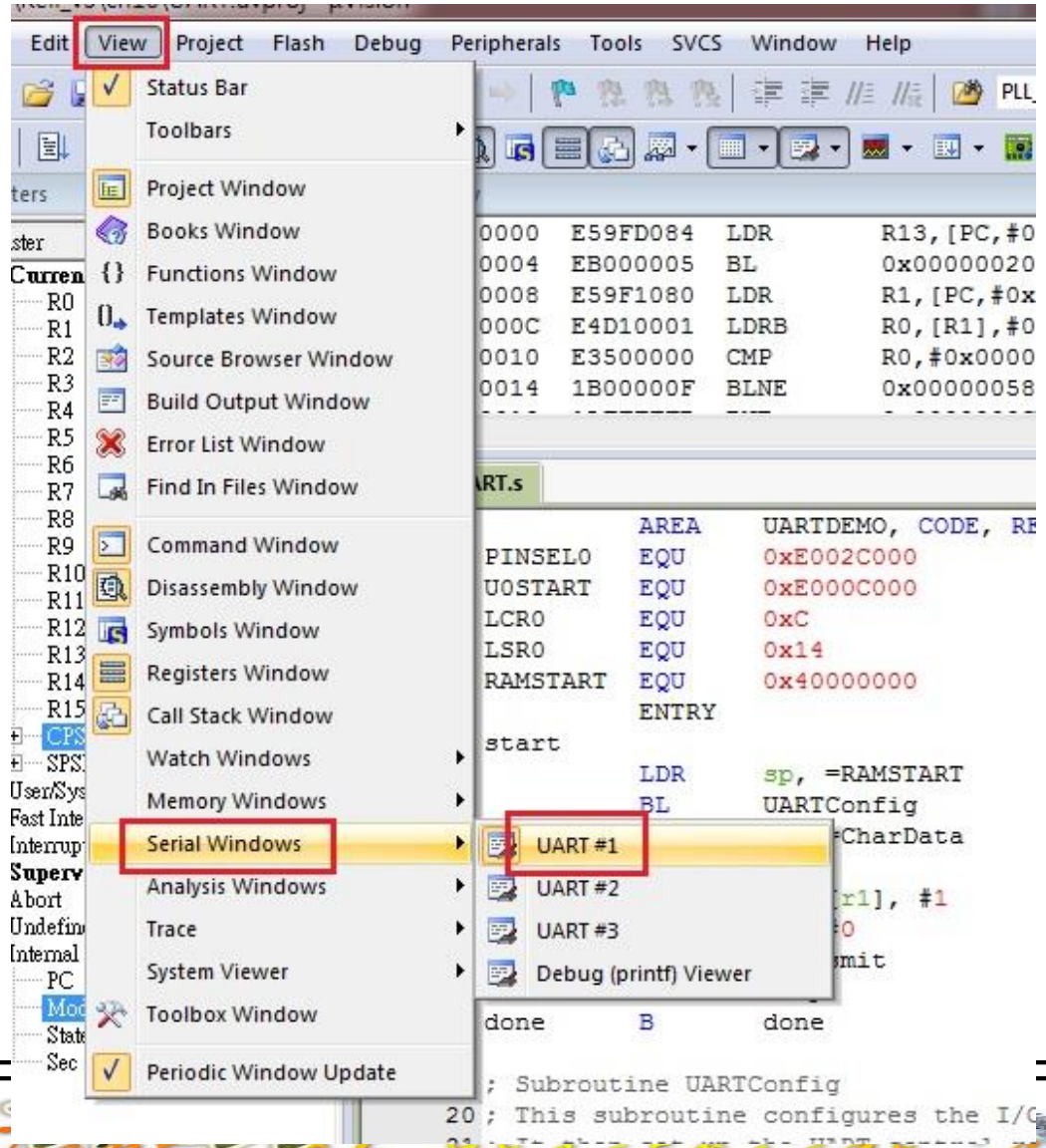
程式(2/3)

```
19
20 UARTConfig
21          STMIA   sp!, {r5, r6, LR}
22          LDR     r5, =PINSEL0        ; base address of register
23          LDR     r6, [r5]            ; get contents
24          BIC     r6, r6, #0xF        ; clear out lower nibble
25          ORR     r6, r6, #0x5        ; sets p0.0 to Tx0 and P0.1 to Rx0
26          STR     r6, [r5]            ; r/modify/w back to register
27          LDR     r5, =U0START
28          MOV     r6, #0x83           ; set 8 bits, no parity, 1 stop bit
29          STRB    r6, [r5, #LCR0]     ; write control byte to LCR
30          MOV     r6, #0x61           ; 9600 baud @15 MHz VPB clock
31          STRB    r6, [r5]            ; store control byte
32          MOV     r6, #3              ; set DLAB=0
33          STRB    r6, [r5, #LCR0]     ; Tx and Rx buffers set up
34          LDMDB   sp!, {r5, r6, PC}
35
```

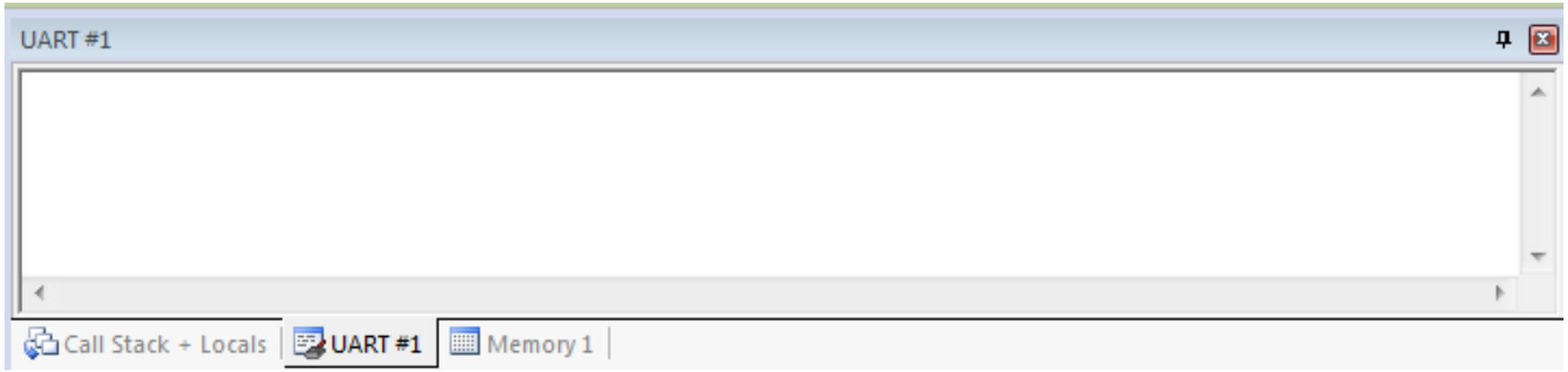
程式(3/3)

```
Transmit
            STMIA   sp!, {r5, r6, LR}
            LDR     r5, =U0START
wait
            LDRB    r6, [r5, #LSR0]    ; get staus of buffer
改成TST     CMP     r6, #0x20          ; buffer empty?
            BEQ     wait               ; spin until buffer's empty
            STRB    r0, [r5]
            LDMDB   sp!,{r5, r6, PC}
CharData
            DCB     "Watson. Come quickly!",0
            END
```

開啟UART視窗

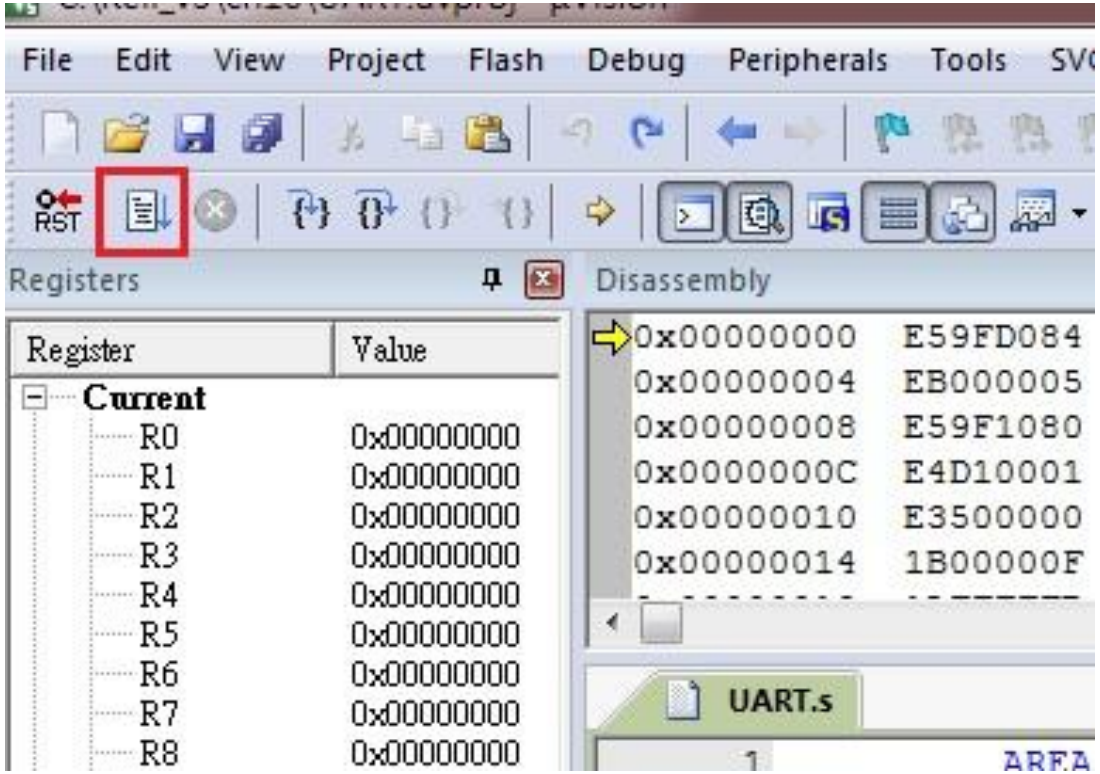

Edit
View
Project
Flash
Debug
Peripherals
Tools
SVCS
Window
Help
Status Bar
Toolbars
Project Window
Books Window
Functions Window
Templates Window
Source Browser Window
Build Output Window
Error List Window
Find In Files Window
Command Window
Disassembly Window
Symbols Window
Registers Window
Call Stack Window
Watch Windows
Memory Windows
Serial Windows
Analysis Windows
Trace
System Viewer
Toolbox Window
Periodic Window Update
UART #1
UART #2
UART #3
Debug (printf) Viewer

右下角可以看到UART視窗



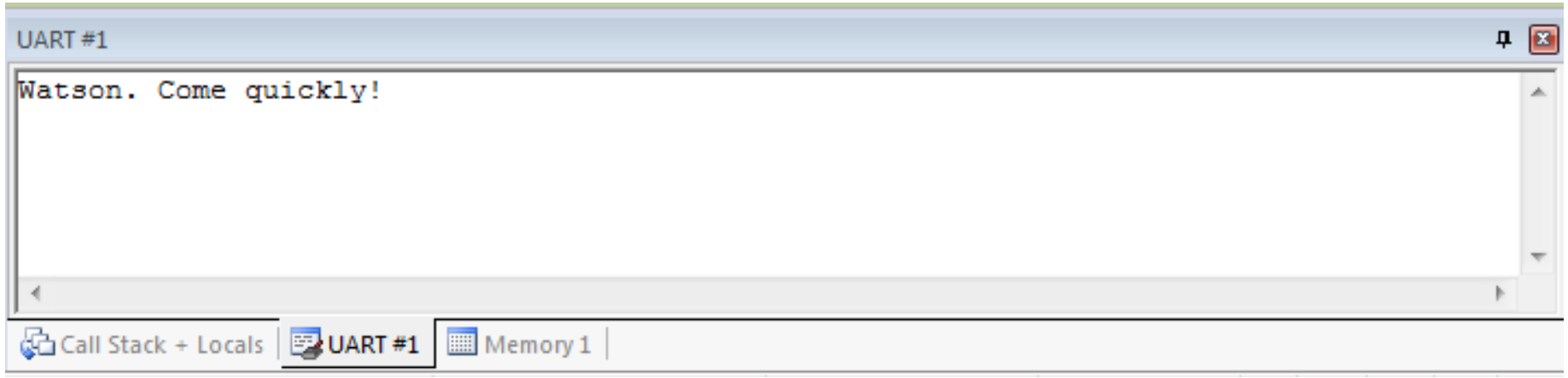
執行程式

- 執行程式使用”Run”或按”F5”



執行結果

- 結果圖成功印出字串



第一部分作業內容

- (1) 執行課本p.348-349程式

  印出字串”**TKU-ECE+學號+英文名字**”

  Ex: TKU-ECE 407443210 Michael

(2) 改寫上述程式使能印出其反向字串

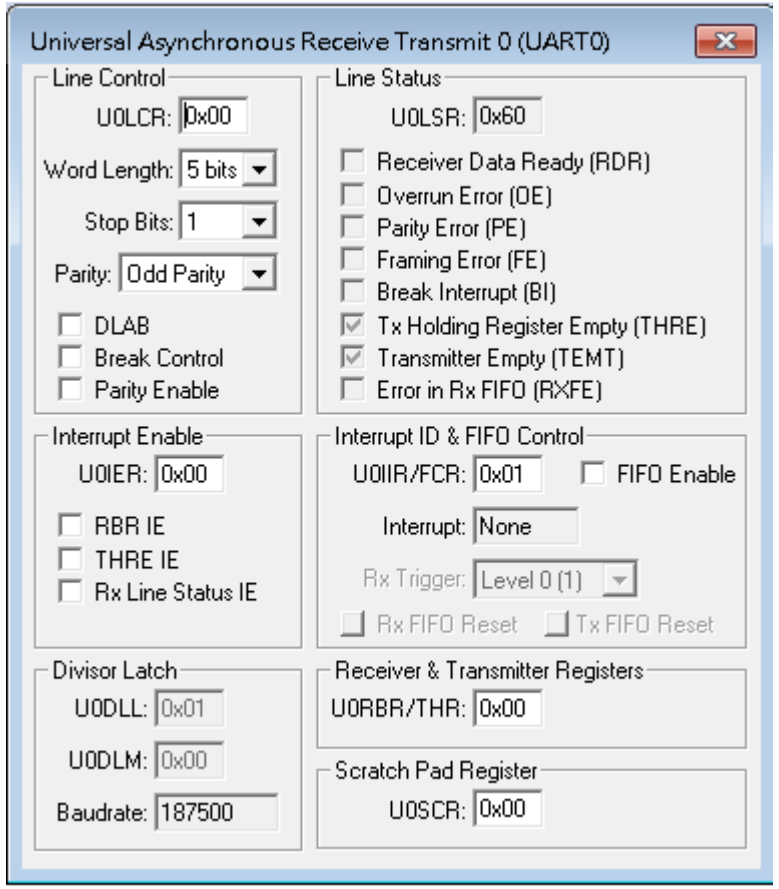
  Ex: leahciM 012344704 ECE-UKT

(3) Calculate Keil Tool System Clock Frequency and rewrite the initialization to show about 9600 baud on the divisor latch window

第一部分作業內容

- 截圖清楚，要截到UART#1及如下頁所示之UART0視窗圖。
- 請交完整結報word檔。(包括按F5、按F11及按F10執行效果的心得）

UART0視窗


Universal Asynchronous Receive Transmit 0 (UART0)
Line Control
U0LCR: 0x00
Word Length: 5 bits
Stop Bits: 1
Parity: Odd Parity
DLAB
Break Control
Parity Enable
Line Status
U0LSR: 0x60
Receiver Data Ready (RDR)
Overrun Error (OE)
Parity Error (PE)
Framing Error (FE)
Break Interrupt (BI)
Tx Holding Register Empty (THRE)
Transmitter Empty (TEMT)
Error in Rx FIFO (RXFE)
Interrupt Enable
U0IER: 0x00
RBR IE
THRE IE
Rx Line Status IE
Interrupt ID & FIFO Control
U0IIR/FCR: 0x01
FIFO Enable
Interrupt: None
Rx Trigger: Level 0 (1)
Rx FIFO Reset
Tx FIFO Reset
Divisor Latch
U0DLL: 0x01
U0DLM: 0x00
Baudrate: 187500
Receiver & Transmitter Registers
U0RBR/THR: 0x00
Scratch Pad Register
U0SCR: 0x00

第二部分

- 加入上課所講的Receive (主程式)

```
Loop
        LDRB r0, [r1],#1 ; load character, increment address
        CMP r0,#0 ; null terminated?
        BLNE Transmit ; send character to UART
        BNE Loop ; continue if not a ??

        MOV  r9,#20
Loop1
        BL Receive
        STRB  r0, [r1],#1
        SUBS  r9,r9,#1
        BNE Loop1
done    B done
```

加入上課所講的Receive (副程式)

```
Transmit
        STMIA sp!,{r5,r6,lr}
        LDR r5, = UOSTART
wait
        LDRB r6,[r5,#LSRO] ; get status of buffer
        TST r6,#0x20 ; buffer empty?
        BEQ wait ; spin until buffer   empty
        STRB r0,[r5]
        LDMDB sp!,{r5,r6, pc}

Receive
        STMIA sp!,{r5,r6,lr}
        LDR r5, = UOSTART
wait1
        LDRB r6,[r5,#LSRO] ; get status of buffer
        TST r6,#1 ;data ready
        BEQ wait1 ;until data ready
        LDRB r0,[r5]
        LDMDB sp!,{r5,r6, pc}
```

打開圖中的Peripherals

D:\Keil Tool\789.uvproj - µVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
Current	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	

Project Registers

Disassembly

```
0x00000000  E59FD0BC  LDR       R13,[PC,#0x00BC]
    11:         BL UARTConfig ; initialize/configure UART0
0x00000004  EB00000A  BL        0x00000034
    12:         LDR r1, = CharData ; starting address of characters
    13: Loop
```

147.s

```
 5 LCR0     EQU 0xC
 6 LSR0     EQU 0x14
 7 RAMSTART EQU 0x40000000
 8          ENTRY
 9 start
10          LDR  sp,=RAMSTART
11          BL UARTConfig ; initialize/configure UART0
12          LDR r1, = CharData ; starting address of characters
13 Loop
14          LDRB r0, [r1],#1 ; load character, increment address
15          CMP r0,#0 ; null terminated?
16          BLNE Transmit ; send character to UART
17          BNE Loop ; continue if not a ??
18
19          MOV  r9,#20
20 Loop1
21          BL Receive
```

Command

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 212 Bytes (0%)

BS \\789\147.s\10, 1
```

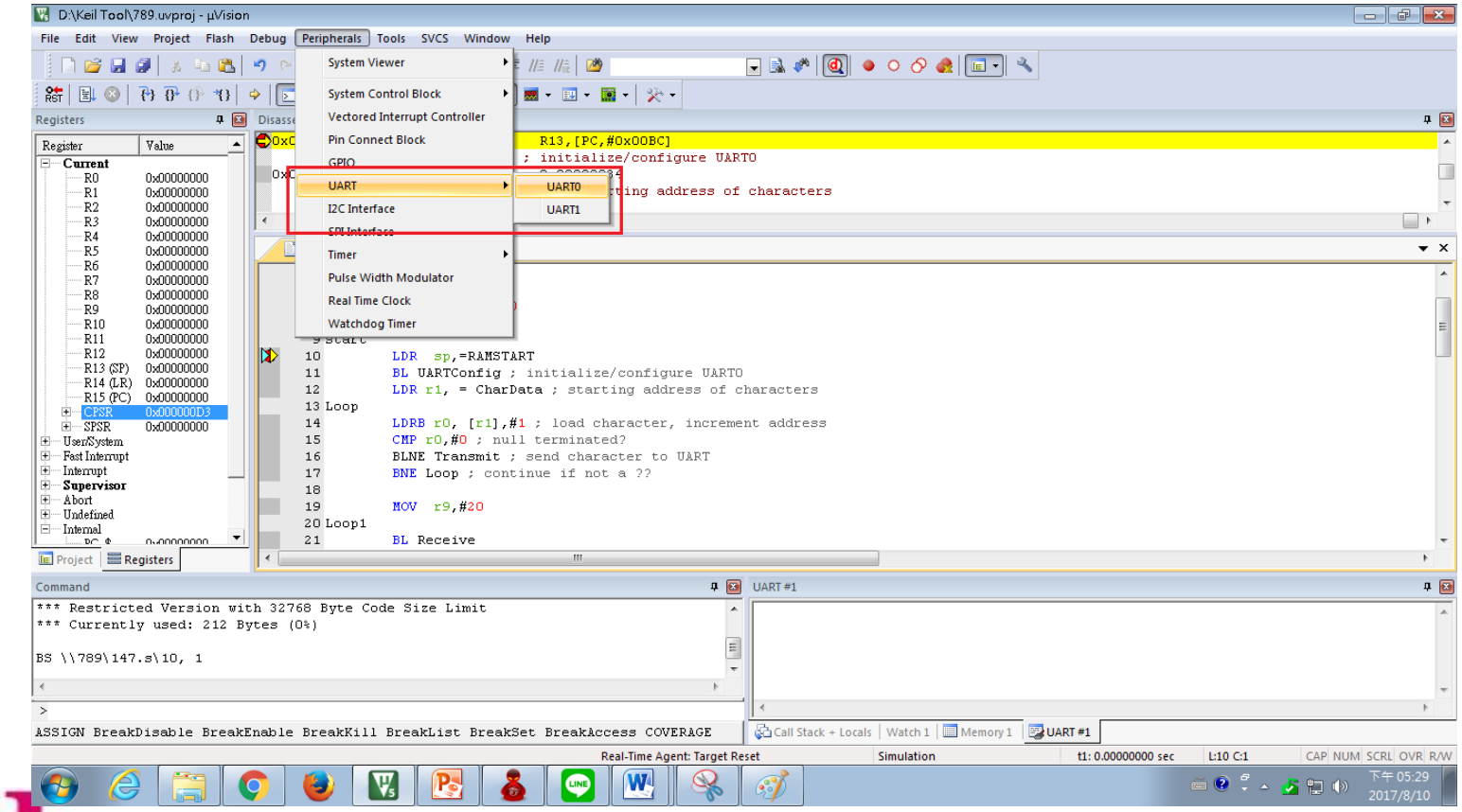
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE

UART #1

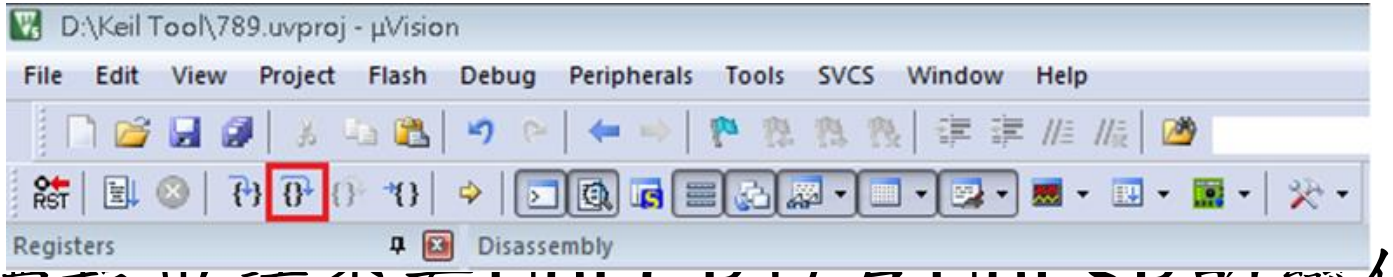
Call Stack + Locals | Watch 1 | Memory 1 | UART #1

Real-Time Agent: Target Reset | Simulation | t1: 0.00000000 sec | L:10 C:1 | CAP NUM SCRL OVR R/W

找到UART點選UART0



- 接著點選圖中的指令或是F10 and F11一直到結束



- 再點選的過程中請查看U0LCR以及U0LSR的變化，並思考其中代表的意義。
- 加入三種錯誤的偵測，沒有錯誤的時候才可receive

第二部分作業內容

- 印出字串 ”**TKU-ECE+學號+英文名字**”
- 截圖清楚，要截到UART0視窗圖以及UART#1視窗圖。
- 請繳交包含第一部分與第二部分的完整結報word檔。

- 繳交內容：按照結報格式編寫完整，把結報word檔上傳iclass對應作業位置。
- 繳交期限： 4/9 23:59
- Word檔名：

  微處理機概論_學號_姓名_HW1
- 程式**需在Keil Tool程式視窗內展示**且展示程式與執行結果的**每一截圖需看的到學號姓名**否則不計分。

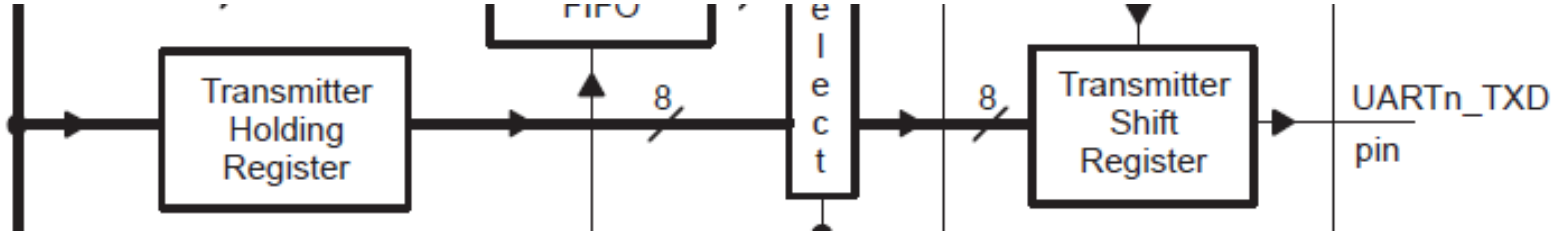
HW1重點整理

- 第一部分：
  - (3) Calculate Keil Tool **System Clock Frequency** and rewrite the initialization to **show about 9600 baud on the divisor latch window**
  - 先算出**System Clock Frequency**，再改寫”設定速率”的程式碼，使其在**divisor latch window**上顯示9600 baud
  - 須以註解或手算方式來呈現**System Clock Frequency**
- 第二部分：
  - Receive 主程式→SUB**S** r9, r9, #1
  - 代表運算後的結果會影響狀態暫存器 CPSR
    - 也可以寫成
    - SUB r9, r9, #1
    - CMP r9, #0

在ARM架構中，CPSR寄存器的NZCV標誌位佔有4個bit，分別代表以下意義：

- N（Negative，負數）：第31位，如果計算結果是負數，則該位被設置為1，否則為0。
- Z（Zero，零）：第30位，如果計算結果為零，則該位被設置為1，否則為0。
- C（Carry，進位）：第29位，用於無符號算術操作，如果發生進位，則該位被設置為1，否則為0。
- V（Overflow，溢出）：第28位，用於有符號算術操作，如果發生溢出，則該位被設置為1，否則為0。

UART技術手冊 p8、p30



6	TEMT		Transmitter empty (TEMT) indicator.
			In non-FIFO mode:
		0	Either the transmitter holding register (THR) or the transmitter shift register (TSR) contains a data character.
		1	Both the transmitter holding register (THR) and the transmitter shift register (TSR) are empty.
			In FIFO mode:
		0	Either the transmitter FIFO or the transmitter shift register (TSR) contains a data character.
		1	Both the transmitter FIFO and the transmitter shift register (TSR) are empty.
5	THRE		Transmitter holding register empty (THRE) indicator. If the THRE bit is set and the corresponding interrupt enable bit is set (ETBEI = 1 in IER), an interrupt request is generated.
			In non-FIFO mode:
		0	Transmitter holding register (THR) is not empty. THR has been loaded by the CPU.
		1	Transmitter holding register (THR) is empty (ready to accept a new character). The content of THR has been transferred to the transmitter shift register (TSR).
			In FIFO mode:
		0	Transmitter FIFO is not empty. At least one character has been written to the transmitter FIFO. You can write to the transmitter FIFO if it is not full.
		1	Transmitter FIFO is empty. The last character in the FIFO has been transferred to the transmitter shift register (TSR).

Keil Tool-P348、349(F11)

❖ 第一次迴圈

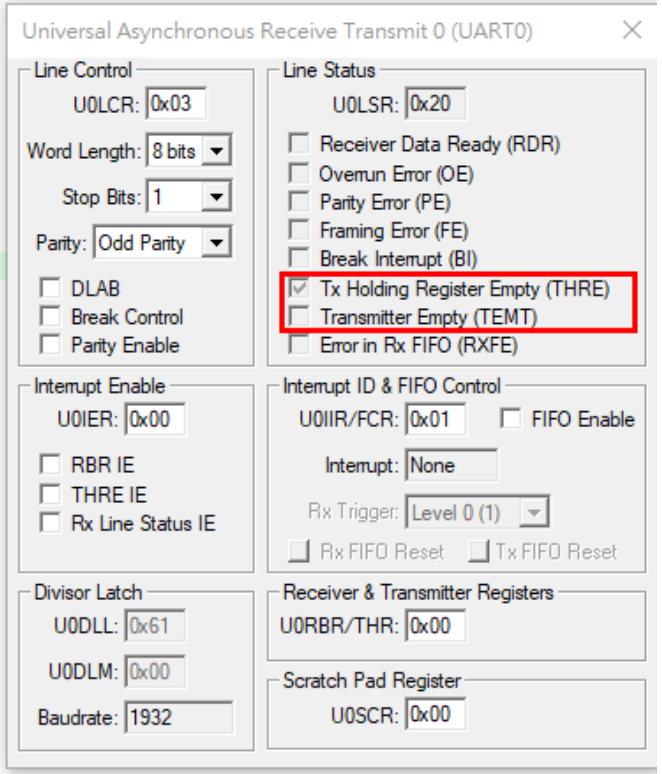
→如果TSR是空的，則資料會自動從THR傳到TSR。

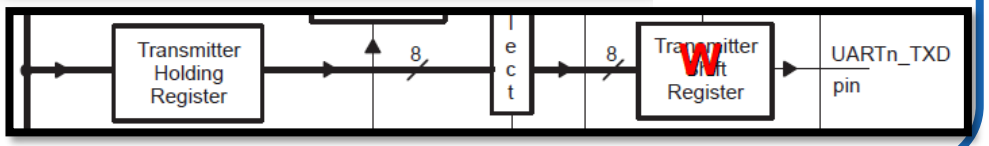
p348UART.s

```
LSR0        EQU     0x14            ; line status register for UART0
RAMSTART    EQU     0x40000000      ; start of onboard RAM for 2104
            ENTRY
start
            LDR     sp, = RAMSTART  ; set up stack pointer
            BL      UARTConfig      ; initialize/configure UART0
            LDR     r1, = CharData  ; starting address of characters
Loop
            LDRB    r0, [r1],#1     ; load character, increment address
            CMP     r0,#0           ; null terminated?
            BLNE    Transmit        ; send character to UART
            BNE     Loop            ; continue if not a '0'
done        B       done            ; otherwise we  e done

; Subroutine UARTConfig
; This subroutine configures the I/O pins first. It
; then sets up the UART control register. The
; parameters
; are set to 8 bits, no parity and 1 stop bit.
; Registers used:
; r5 - scratch register
; r6 - scratch register
; inputs: none
; outputs: none

UARTConfig
            STMIA   sp!, {r5,r6,lr}
            LDR     r5, = PINSEL0   ; base address of register
            LDR     r6, [r5]        ; get contents
            BIC     r6, r6, #0xF    ; clear out lower nibble
            ORR     r6, r6, #0x5    ; sets P0.0 to Tx0 and P0.1 to Rx0
            STR     r6, [r5]        ; r/modify/w back to register
            LDR     r5, = U0START
            MOV     r6, #0x83       ; set 8 bits, no parity, 1 stop bit
            STRB    r6, [r5, #LCR0] ; write control byte to LCR
```





Keil Tool-P348、349(F11)

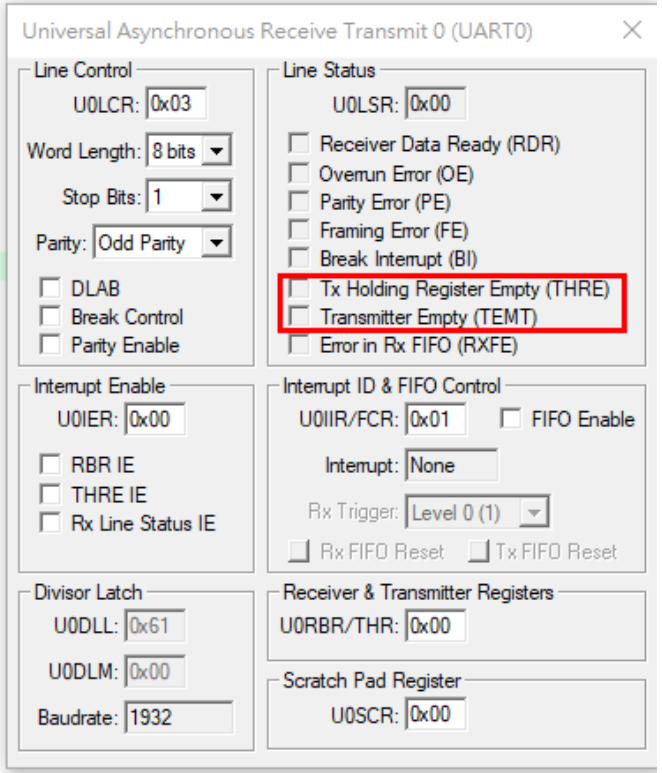
❖ 第二次迴圈

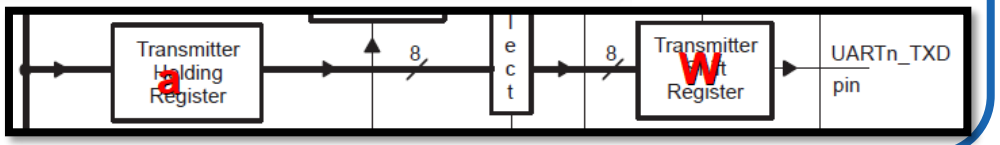
p348UART.s

```
PINSEL0     EQU     0xE002C000      ; controls the function of the pins
U0START     EQU     0xE000C000      ; start of UART0 registers
LCR0        EQU     0xC             ; line control register for UART0
LSR0        EQU     0x14            ; line status register for UART0
RAMSTART    EQU     0x40000000      ; start of onboard RAM for 2104
            ENTRY
start
            LDR     sp, = RAMSTART  ; set up stack pointer
            BL      UARTConfig      ; initialize/configure UART0
            LDR     r1, = CharData  ; starting address of characters
Loop
            LDRB    r0, [r1],#1     ; load character, increment address
            CMP     r0,#0           ; null terminated?
            BLNE    Transmit        ; send character to UART
            BNE     Loop            ; continue if not a '0'
done        B       done            ; otherwise we  e done

; Subroutine UARTConfig
; This subroutine configures the I/O pins first. It
; then sets up the UART control register. The
; parameters
; are set to 8 bits, no parity and 1 stop bit.
; Registers used:
; r5 - scratch register
; r6 - scratch register
; inputs: none
; outputs: none

UARTConfig
            STMIA   sp!, {r5,r6,lr}
            LDR     r5, = PINSEL0   ; base address of register
            LDR     r6, [r5]        ; get contents
            BIC     r6, r6, #0xF    ; clear out lower nibble
            ORR     r6, r6, #0x5    ; sets P0.0 to Tx0 and P0.1 to Rx0
            STR     r6, [r5]        ; r/modify/w back to register
```

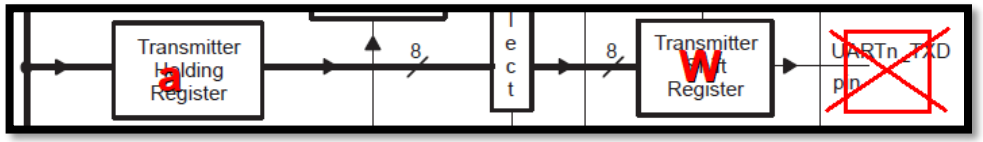




Keil Tool-P348、349(F11)

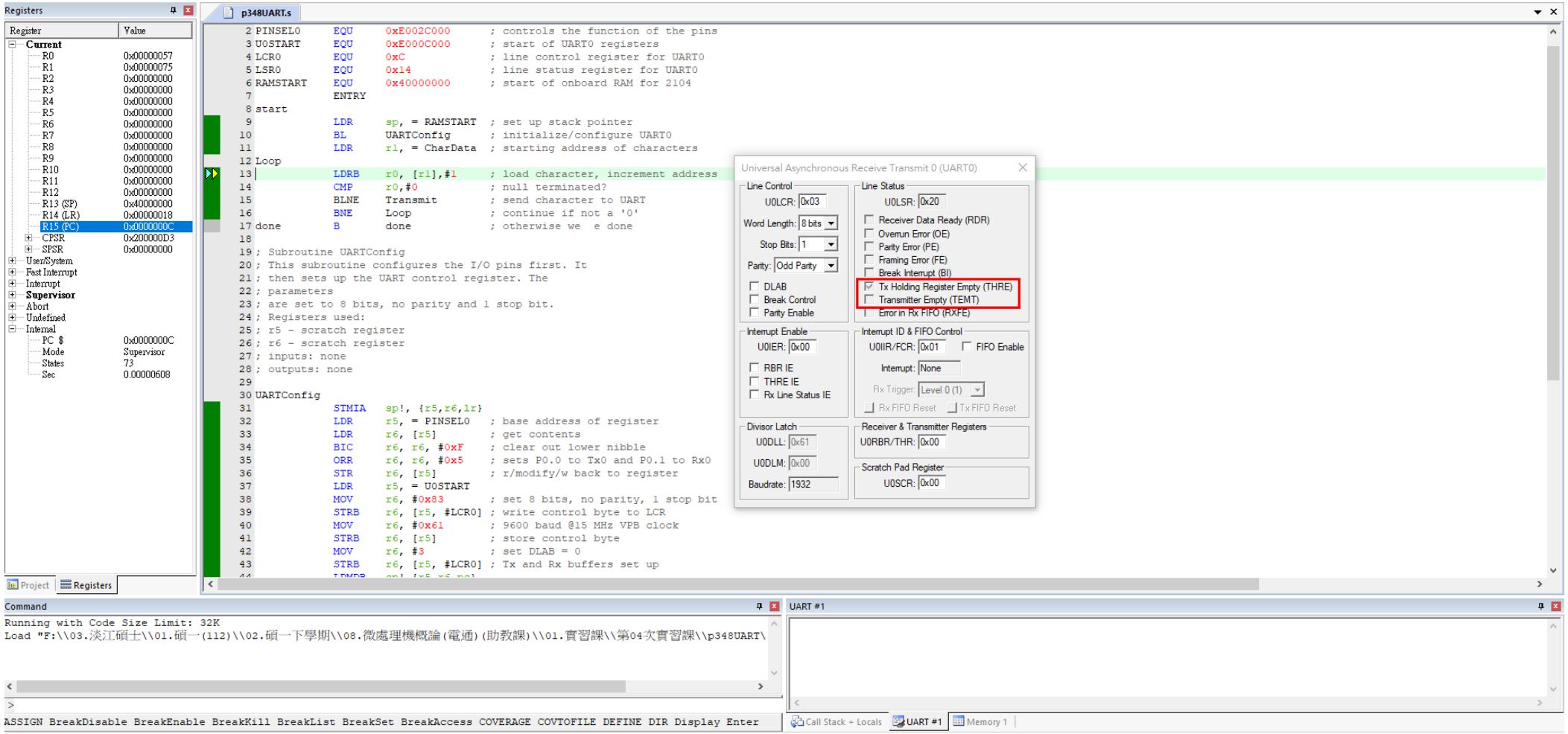
- 第三次迴圈-無窮迴圈(wait)。
- UART#1視窗無法顯示，因為使用F11**沒有模擬的輸出端**，使傳出去的值卡在Transmit holding register裡面，當在判斷THRE的狀態時，就會因為有值在裡面不斷的迴圈**(紅色框框)**。

```
Transmit
            STMIA   sp!, {r5, r6, lr}
            LDR     r5, = U0START
wait        LDRB    r6, [r5, #LSR0] ; get status of buffer
            TST     r6, #0x20       ; buffer empty?
            BEQ     wait            ; spin until buffer's empty
            STRB    r0, [r5]
            LDMDB   sp!, {r5, r6, pc}
CharData
            DCB     "Watson. Come quickly!",0
            END
```



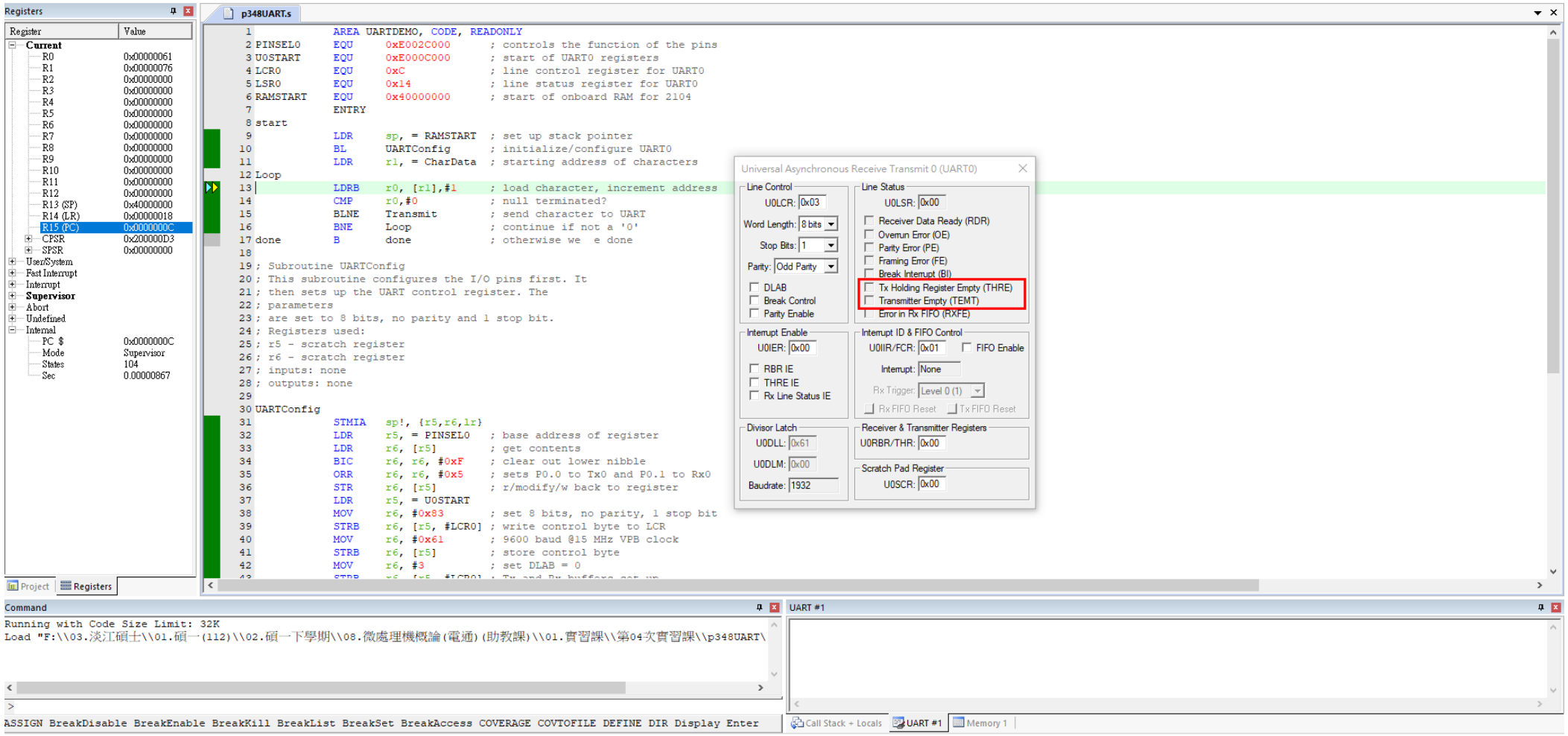
Keil Tool-P348、349(F10)

❖ 第一次迴圈



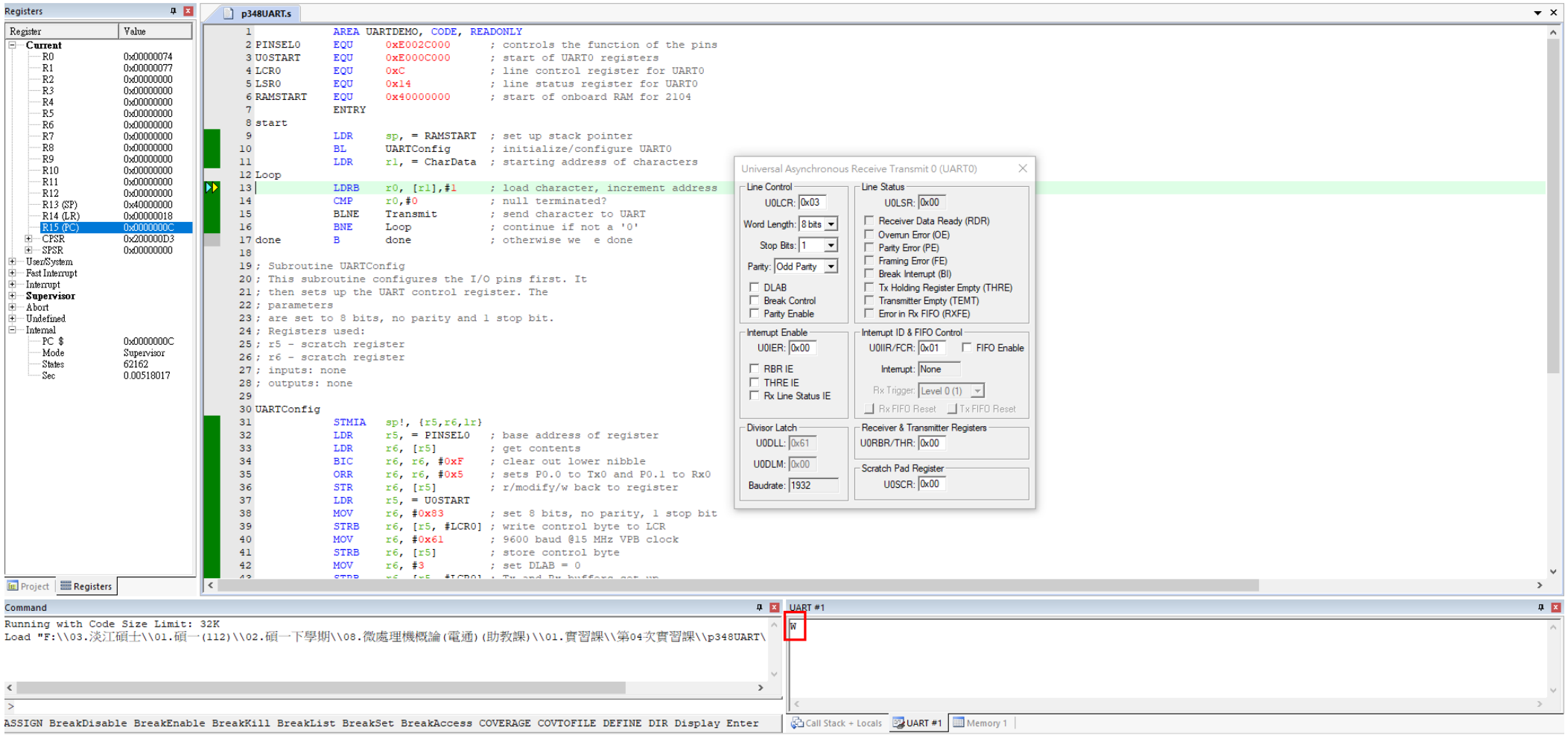
Keil Tool-P348、349(F10)

❖ 第二次迴圈



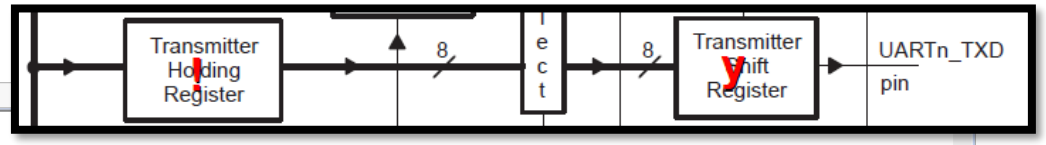
Keil Tool-P348、349(F10)

❖ 第三次迴圈



Keil Tool-P348、349(F10)

❖ 最後一次迴圈



Registers

Register	Value
Current	
R0	0x00000000
R1	0x0000008A
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x40000000
R14 (LR)	0x00000018
R15 (PC)	0x0000001C
CPSR	0x600000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC $	0x0000001C
Mode	Supervisor
States	1179680
Sec	0.09830667

Project | Registers

p348UART.s

```
LCR0        EQU     0xC             ; line control register for UART0
LSR0        EQU     0x14            ; line status register for UART0
RAMSTART    EQU     0x40000000      ; start of onboard RAM for 2104
            ENTRY
start
            LDR     sp, = RAMSTART  ; set up stack pointer
            BL      UARTConfig      ; initialize/configure UART0
            LDR     r1, = CharData  ; starting address of characters
Loop
            LDRB    r0, [r1],#1     ; load character, increment address
            CMP     r0,#0           ; null terminated?
            BLNE    Transmit        ; send character to UART
            BNE     Loop            ; continue if not a '0'
done        B       done            ; otherwise we  e done

; Subroutine UARTConfig
; This subroutine configures the I/O pins first. It
; then sets up the UART control register. The
; parameters
; are set to 8 bits, no parity and 1 stop bit.
; Registers used:
; r5 - scratch register
; r6 - scratch register
; inputs: none
; outputs: none

UARTConfig
            STMIA   sp!, {r5,r6,lr}
            LDR     r5, = PINSEL0   ; base address of register
            LDR     r6, [r5]        ; get contents
            BIC     r6, r6, #0xF    ; clear out lower nibble
            ORR     r6, r6, #0x5    ; sets P0.0 to Tx0 and P0.1 to Rx0
            STR     r6, [r5]        ; r/modify/w back to register
            LDR     r5, = U0START
            MOV     r6, #0x83       ; set 8 bits, no parity, 1 stop bit
            STRB    r6, [r5, #LCR0] ; write control byte to LCR
            MOV     r6, #0x61       ; 9600 baud @15 MHz VPB clock
            STRB    r6, [r5]        ; store control byte
            MOV     r6, #3          ; set DLAB = 0
            STRB    r6, [r5, #LCR0] ; Tx and Rx buffers set up
            LDMDB   sp!,{r5,r6,pc}
```

Universal Asynchronous Receive Transmit 0 (UART0)

Line Control
U0LCR: 0x03
Word Length: 8 bits
Stop Bits: 1
Parity: Odd Parity
DLAB
Break Control
Parity Enable

Line Status
U0LSR: 0x00
Receiver Data Ready (RDR)
Overrun Error (OE)
Parity Error (PE)
Framing Error (FE)
Break Interrupt (BI)
Tx Holding Register Empty (THRE)
Transmitter Empty (TEMT)
Error in Rx FIFO (RXFE)

Interrupt Enable
U0IER: 0x00
RBR IE
THRE IE
Rx Line Status IE

Interrupt ID & FIFO Control
U0IIR/FCR: 0x01
FIFO Enable
Interrupt: None
Rx Trigger: Level 0 (1)
Rx FIFO Reset
Tx FIFO Reset

Divisor Latch
U0DLL: 0x61
U0DLM: 0x00
Baudrate: 1932

Receiver & Transmitter Registers
U0RBR/THR: 0x00

Scratch Pad Register
U0SCR: 0x00

Command

```
Running with Code Size Limit: 32K
Load "F:\\03.淡江碩士\\01.碩一(112)\\02.碩一下學期\\08.微處理機概論(電通)(助教課)\\01.實習課\\第04次實習課\\p348UART\
```

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter

UART #1

```
Watson. Come quickl
```

Call Stack + Locals | UART #1 | Memory 1

Keil Tool-P348、349(F5)

→兩種情況THR、TSR的值會被傳出來

1. 程式結束時(F5)
2. 重新存取U0START時(Receive)

```
AREA UARTDEMO, CODE, READONLY
PINSEL0     EQU     0xE002C000      ; controls the function of the pins
U0START     EQU     0xE000C000      ; start of UART0 registers
LCR0        EQU     0xC             ; line control register for UART0
LSR0        EQU     0x14            ; line status register for UART0
RAMSTART    EQU     0x40000000      ; start of onboard RAM for 2104
            ENTRY
start
            LDR     sp, = RAMSTART  ; set up stack pointer
            BL      UARTConfig      ; initialize/configure UART0
            LDR     r1, = CharData  ; starting address of characters
Loop
            LDRB    r0, [r1],#1     ; load character, increment address
            CMP     r0,#0           ; null terminated?
            BLNE    Transmit        ; send character to UART
            BNE     Loop            ; continue if not a '0'
done        B       done            ; otherwise we  e done

; Subroutine UARTConfig
; This subroutine configures the I/O pins first. It
; then sets up the UART control register. The
; parameters
; are set to 8 bits, no parity and 1 stop bit.
; Registers used:
; r5 - scratch register
; r6 - scratch register
; inputs: none
; outputs: none

UARTConfig
            STMIA   sp!, {r5,r6,lr}
            LDR     r5, = PINSEL0   ; base address of register
            LDR     r6, [r5]        ; get contents
            BIC     r6, r6, #0xF    ; clear out lower nibble
            ORR     r6, r6, #0x5    ; sets P0.0 to Tx0 and P0.1 to Rx0
            STR     r6, [r5]        ; r/modify/w back to register
            LDR     r5, = U0START
            MOV     r6, #0x83       ; set 8 bits, no parity, 1 stop bit
            STRB    r6, [r5, #LCR0] ; write control byte to LCR
            MOV     r6, #0x61       ; 9600 baud @15 MHz VPB clock
            STRB    r6, [r5]        ; store control byte
            MOV     r6, #3          ; set DLAB = 0
```

Line Control: U0LCR: 0x03; Word Length: 8 bits; Stop Bits: 1; Parity: Odd Parity; DLAB; Break Control; Parity Enable

Line Status: U0LSR: 0x60; Receiver Data Ready (RDR); Overrun Error (OE); Parity Error (PE); Framing Error (FE); Break Interrupt (BI); Tx Holding Register Empty (THRE); Transmitter Empty (TEMT); Error in Rx FIFO (RXFE)

Interrupt Enable: U0IER: 0x00; RBR IE; THRE IE; Rx Line Status IE

Interrupt ID & FIFO Control: U0IIR/FCR: 0x01; FIFO Enable; Interrupt: None; Rx Trigger: Level 0 (1); Rx FIFO Reset; Tx FIFO Reset

Divisor Latch: U0DLL: 0x61; U0DLM: 0x00; Baudrate: 1932

Receiver & Transmitter Registers: U0RBR/THR: 0x00

Scratch Pad Register: U0SCR: 0x00

Command:
```
Running with Code Size Limit: 32K
Load "F:\\03.淡江碩士\\01.碩一(112)\\02.碩一下學期\\08.微處理機概論(電通)(助教課)\\01.實習課\\第04次實習課\\p348UART\
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE DEFINE DIR Display Enter
```

UART #1:
```
Watson. Come quickly!
```

p347 poll & busy waiting

ing this data? In the simulation tools, there is a serial window that can accept data from a UART, driving the necessary handshake lines that are normally attached to the receiver. The assembly code for our transmitter routine looks like the following:

只要狀態沒成立，CPU就要一直等待，沒辦法做其他事情。

LOOP

POLL(CPU主動詢問周邊晶片)

```
        LDR     r5, =U0START
wait    LDRB    r6,[r5,#LSR0]     ; get status of buffer
        CMP TST r6,#0x20          ; buffer empty?
        BEQ     wait              ; spin until buffer's empty
        STRB    r0,[r5]
```

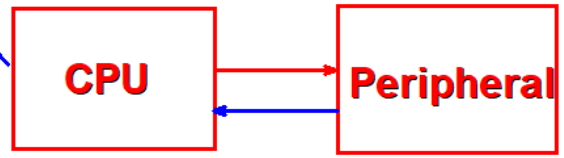
BUSY WAITING
(CPU的狀態)

改進：周邊在任何狀態發生改變時，主動通知CPU，省去BUSY WAITING時間。

INTERRUPT

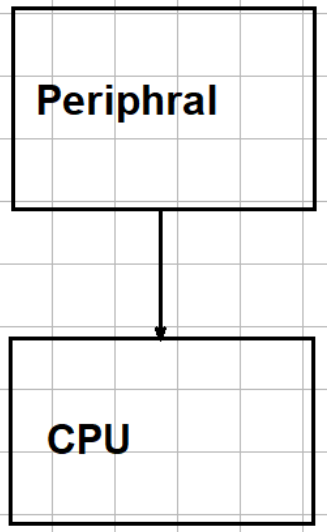
編號 → 讓CPU知道什麼情況發生時，執行哪一個程式。

1. BUFFER EMPTY
2. DATA READY



例：門鈴壞掉->busy waiting

p347 poll & busy waiting



INTERRUPT(並告知CPU編號)

Interrupt number

藉由Interrupt vector table找到對應執行程式的初始位置
並在記憶體裡找到對應的程式並執行

1. first instruction address
2. address

Memory

TSR -> Terminate and Stay Resident(常駐程式)

只要是 ISR -> Interrupt Service Routine

Q&A

Thanks for your attention !!