第04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.)

P347 16.2.4 Writing the Data to the UART

16.2.4 Writing the Data to the UART

Now that the UART is configured to send and receive data, we can try writing some data out of the part. In this case, we'll send some character data—the short message "Watson. Come quickly!" The subroutine for this task is written so that the calling routine can send a single character at a time. When the subroutine receives the character, it's placed into the transmit buffer, but only after the processor checks to ensure the previous character has been transmitted. Who's reading 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:

-> Read bit 5 of LSR to see if bit 5 = 1 or 0.

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

0b0010 0000

NE

同步機制

錯誤寫法 (因為其他未使用到的位元初始值為0，才可以使用CMP。)

建議使用TST 搭配 EQ

UART 0

0xE000C000	U0RBR (DLAB=0)	U0 Receiver buffer register	8-bit data							RO	un–defined
	U0THR (DLAB=0)	U0 Transmit holding register	8-bit data							WO	NA
	U0DLL (DLAB=1)	U0 Divisor latch LSB	8-bit data							R/W	0x01
0xE000C004	U0IER (DLAB=0)	U0 Interrupt enable register	0	0	0	0	0	En. Rx Line Status Int.	Enable THRE Int.	En. Rx Data Av.Int.	R/W	0
	U0DLM (DLAB=1)	U0 Divisor latch LSB	8 bit data								R/W	0
0xE000C008	U0IIR	U0 Interrupt ID register	FIFOs Enabled		0	0	IIR3	IIR2	IIR1	IIR0	RO	0x01
	U0FCR	U0 FIFO control register	Rx Trigger		-	-	-	U0 Tx FIFO Reset	U0 Rx FIFO Reset	U0 FIFO Enable	WO	0
0xE000C00C	U0LCR	U0 Line control register	DLAB	Set break	Stick parity	Even parity select	Parity enable	Nm. of stop bits	Word length select		R/W	0
0xE000C014	U0LSR	U0 Line status register	Rx FIFO Error	TEMT	THRE	BI	FE	PE	OE	DR	RO	0x60
0xE000C01C	U0LSR	U0 Scratch pad register	8-bit data								R/W	0

bit 5(Transmit Holding Register Empty) -> if 1:empty；0:not empty。

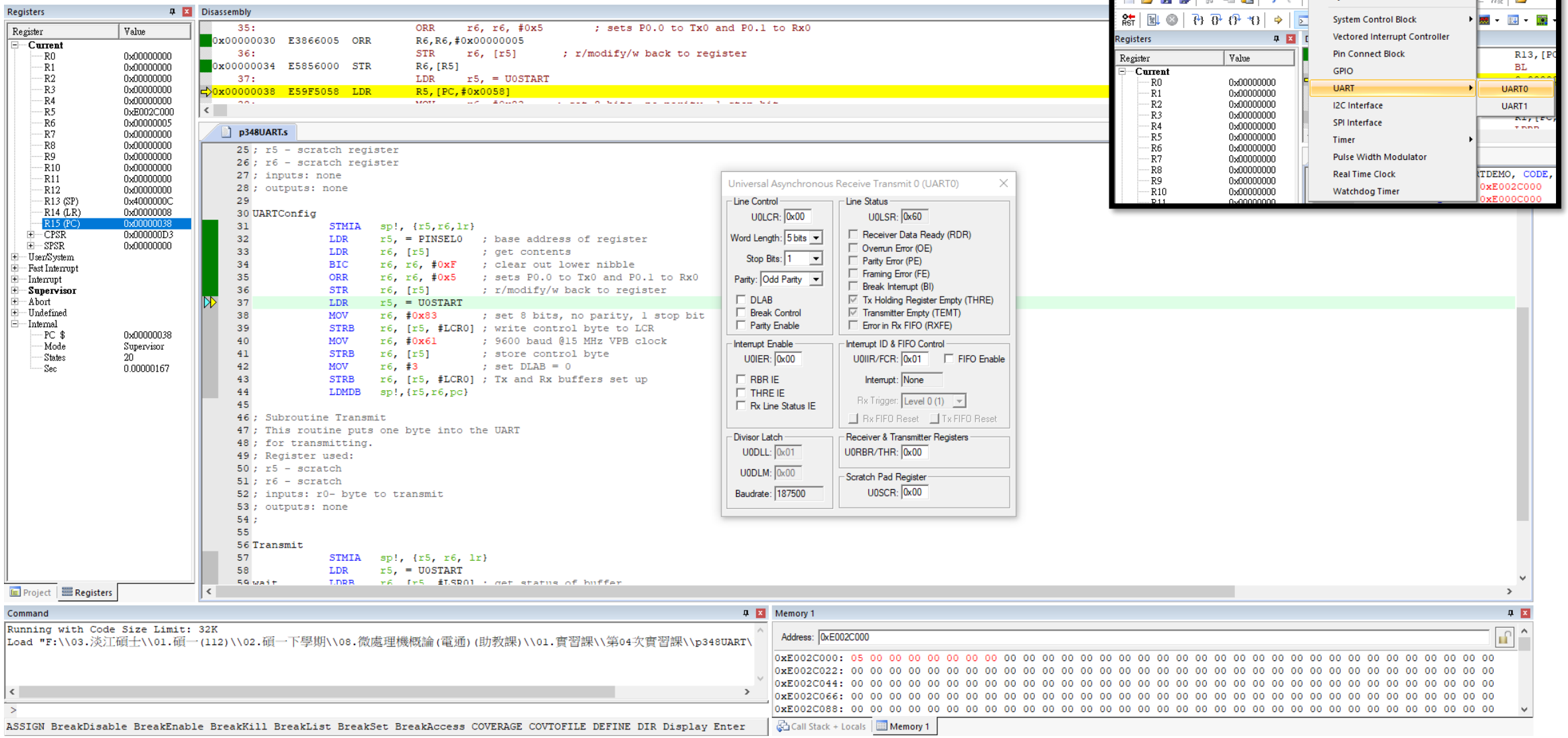
→類似於C語言中的put character

Keil Tool-UARTConfig

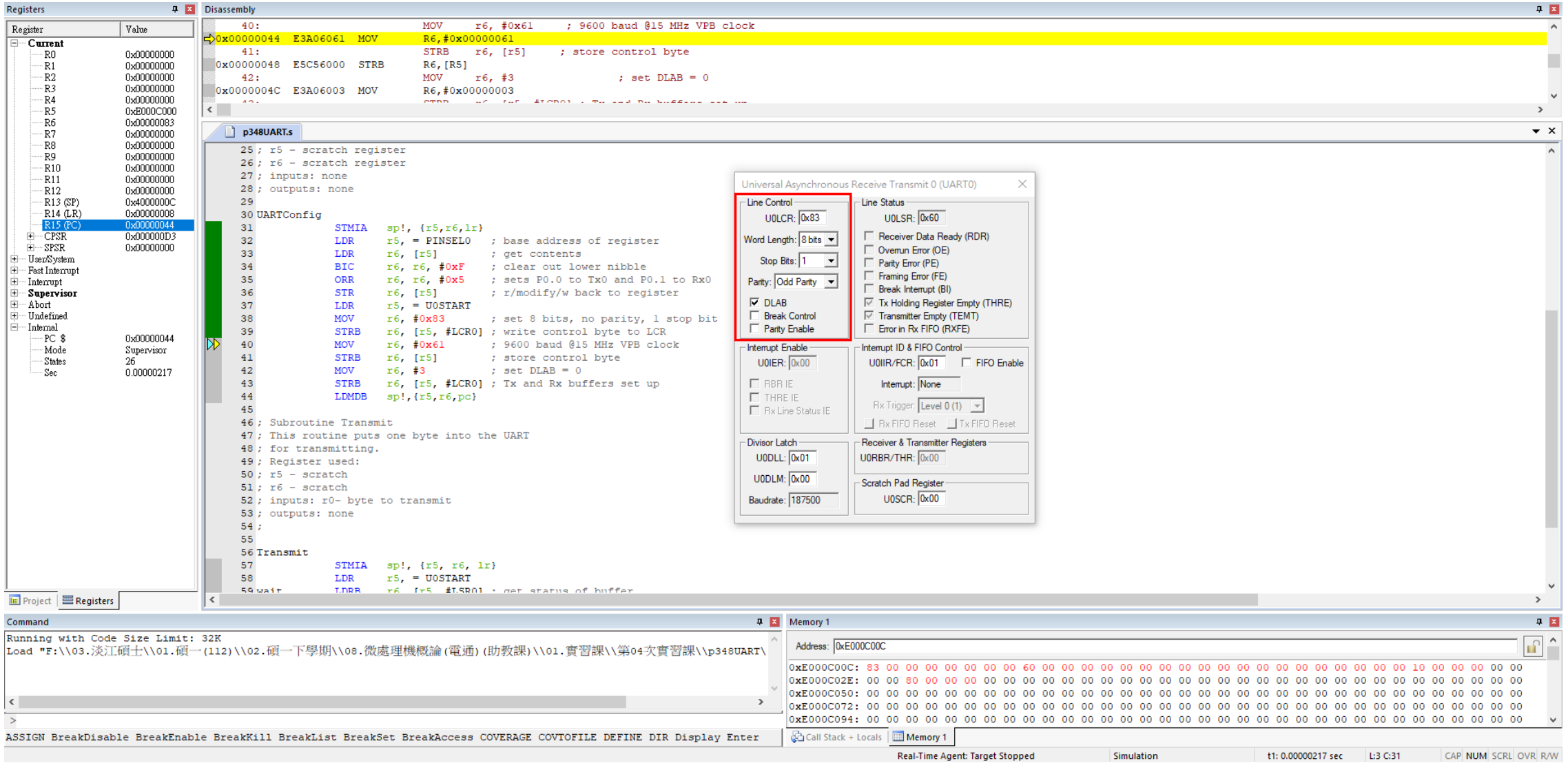
2024 Advanced Mixed-Operation System (AMOS) Lab.

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

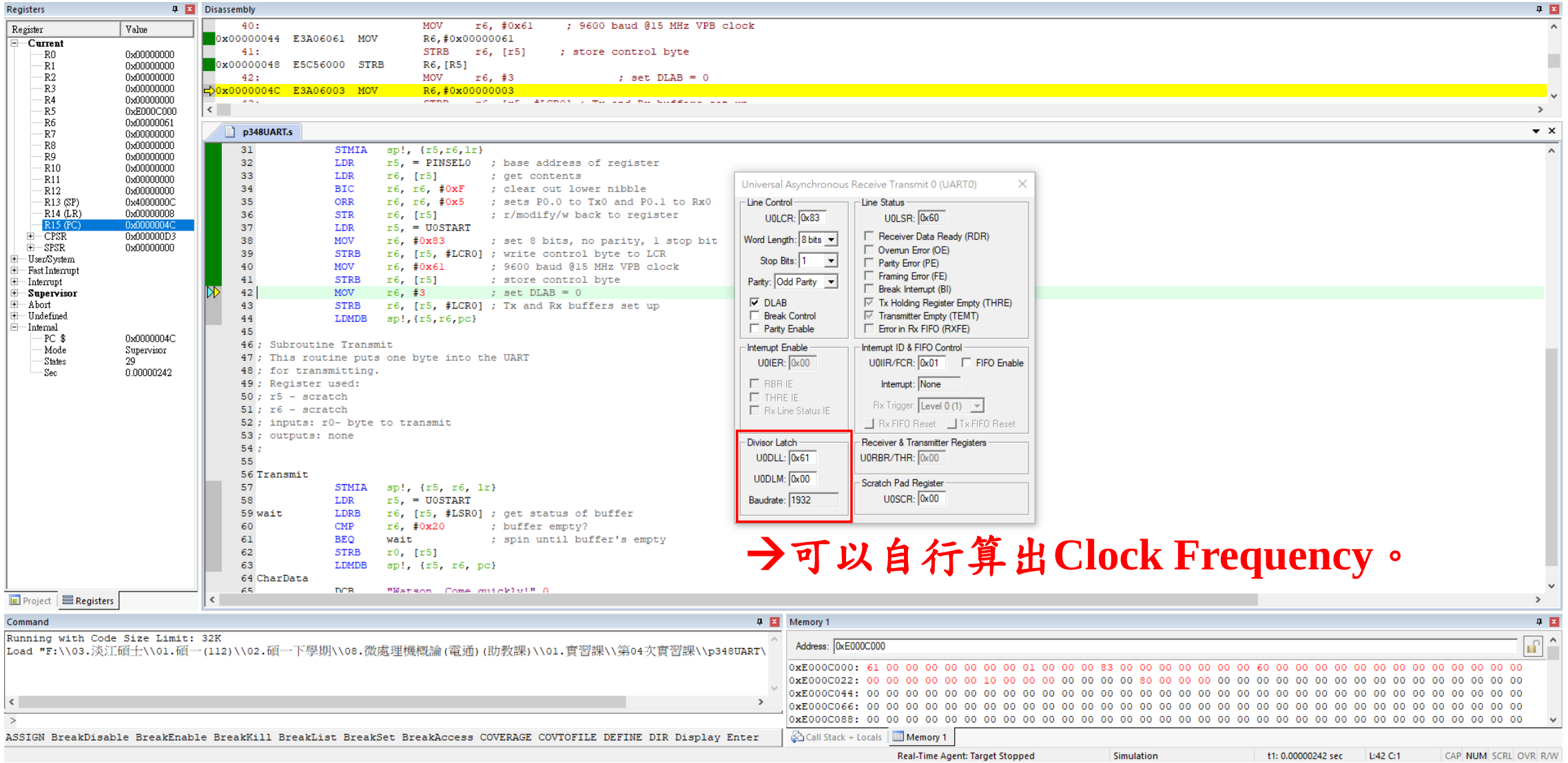
Keil Tool-P348(16.2.5 Putting the Code Together)



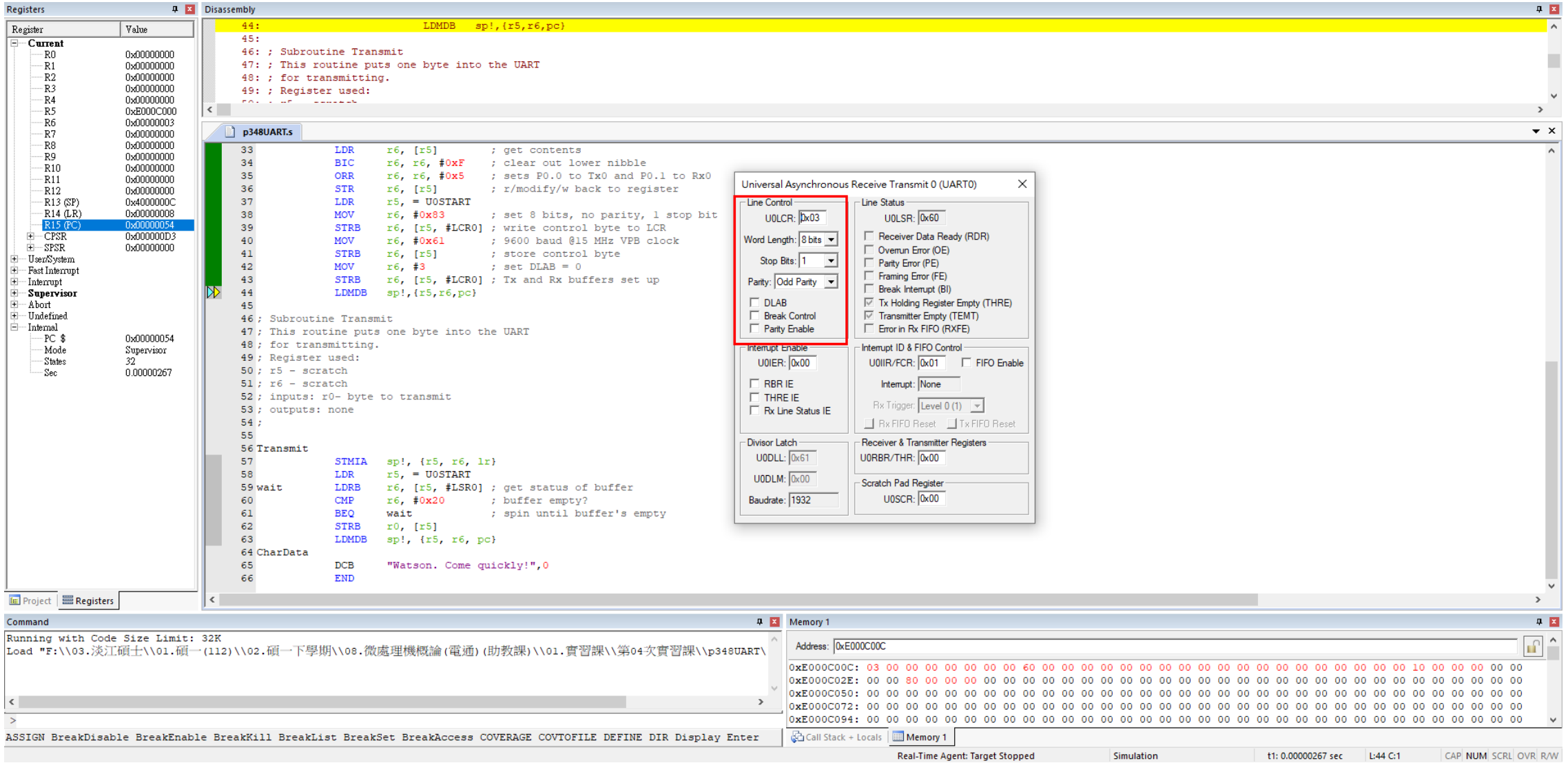
Keil Tool-P348(16.2.5 Putting the Code Together)



Keil Tool-P348(16.2.5 Putting the Code Together)



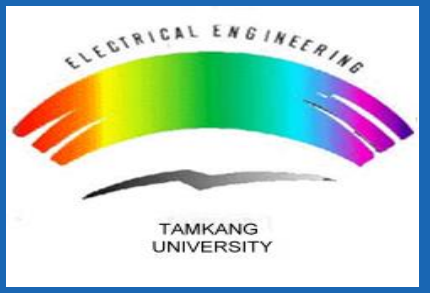
Keil Tool-P348(16.2.5 Putting the Code Together)



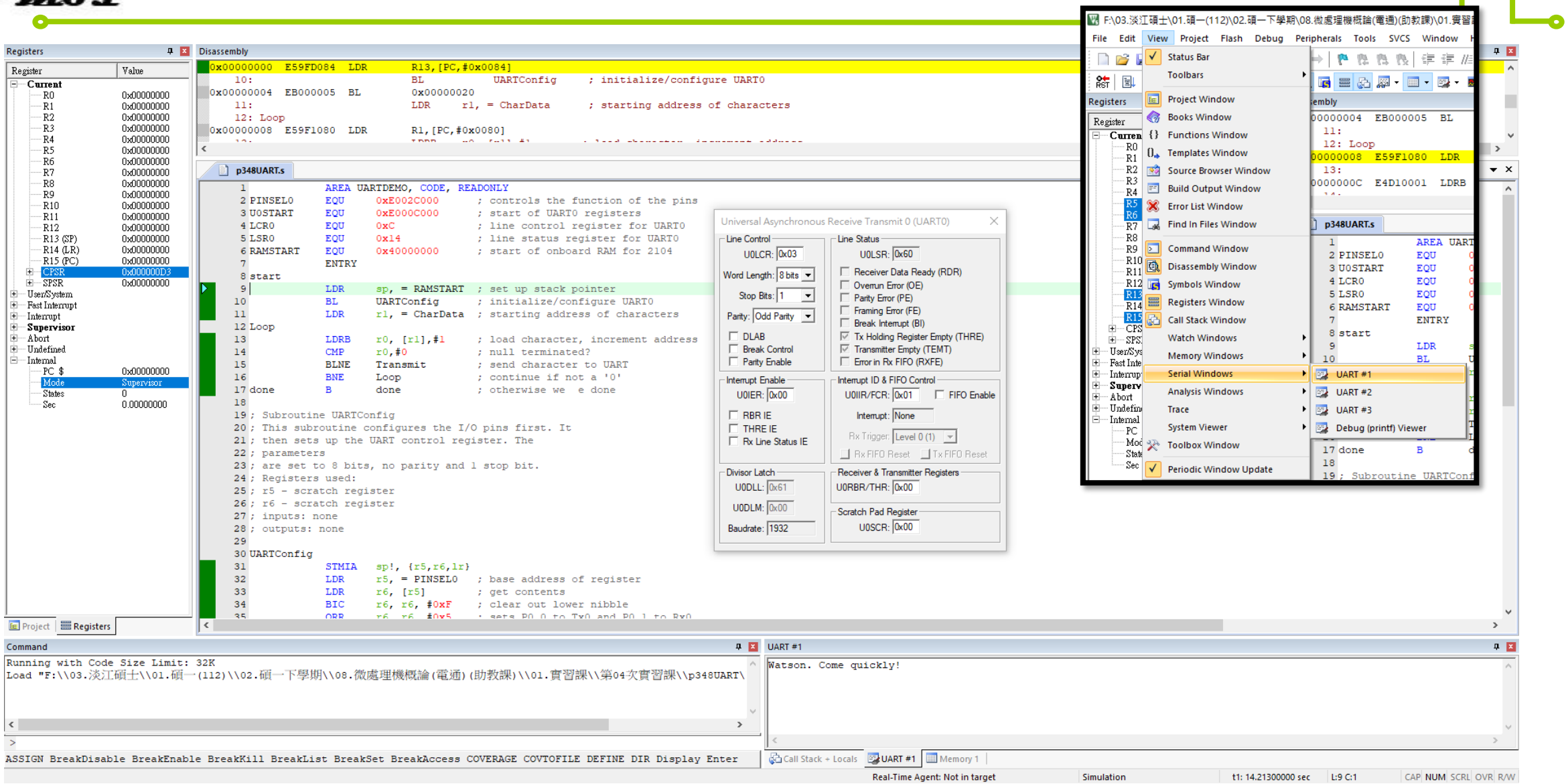
Keil Tool-Transmit

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.)

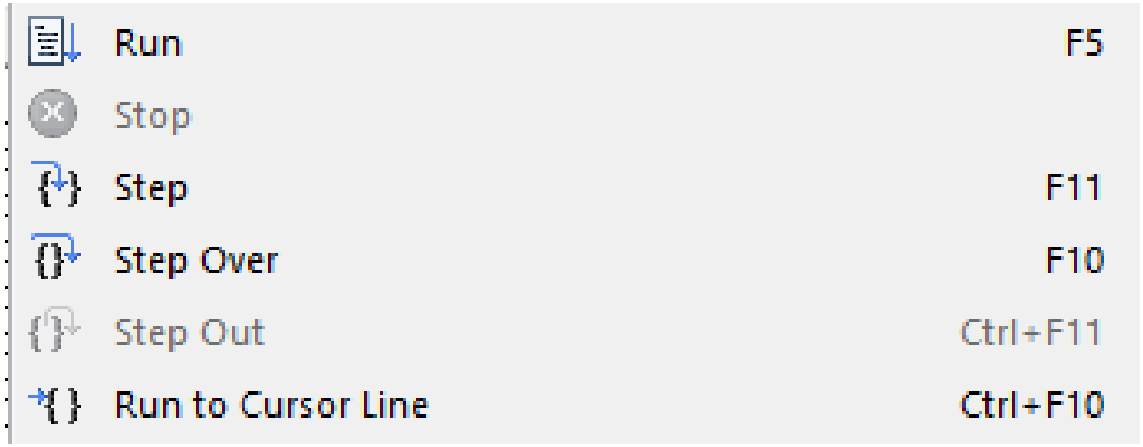


Keil Tool-P348(16.2.5 Putting the Code Together)



Debug：F11 vs. F10 vs. F5

- F11(Step)：單步執行。(一行指令一行指令執行)
- F10(Step Over)：不進副函式，但副函式的結果會呈現出來。
  (把副函式當成一步)
- F5(Run)：整個程式執行完的結果。



練習：反向印出字串

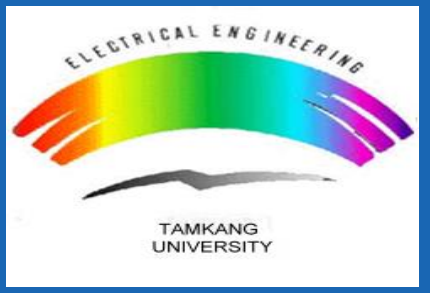
```
            AREA UARTDEMO, CODE, READONLY
PINSEL0     EQU   0xE002C000      ; controls the function of the pins
U0START     EQU   0xE000C000      ; start of UART0 registers
LCR0        EQU   0xC             ; line control register for UART0
LSR0        EQU   0x14            ; line status register 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're done
```

→先自行計算最後一個字元的位置，依序往前取值。(設COUNTER)

Receive

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.)



Reading the data from the UART buffer

```
Receive
            STMIA   sp!, {r5, r6, lr}
            LDR     r5, = U0START
wait
            LDRB    r6, [r5, #LSR0] ; get status of buffer
            TST     r6, #1          ; DR : Data Ready?
            BEQ     wait            ; spin until data ready
            STRB    r0, [r5]
            LDMDB   sp!, {r5, r6, pc}
```

UART 0											
0xE000C000	U0RBR (DLAB=0)	U0 Receiver buffer register	8-bit data							RO	un–defined
	U0THR (DLAB=0)	U0 Transmit holding register	8-bit data							WO	NA
	U0DLL (DLAB=1)	U0 Divisor latch LSB	8-bit data							R/W	0x01
0xE000C004	U0IER (DLAB=0)	U0 Interrupt enable register	0	0	0	0	0	En. Rx Line Status Int.	Enable THRE Int.	En. Rx Data Av.Int.	R/W	0
	U0DLM (DLAB=1)	U0 Divisor latch LSB	8 bit data								R/W	0
0xE000C008	U0IIR	U0 Interrupt ID register	FIFOs Enabled		0	0	IIR3	IIR2	IIR1	IIR0	RO	0x01
	U0FCR	U0 FIFO control register	Rx Trigger		-	-	-	U0 Tx FIFO Reset	U0 Rx FIFO Reset	U0 FIFO Enable	WO	0
0xE000C00C	U0LCR	U0 Line control register	DLAB	Set break	Stick parity	Even parity select	Parity enable	Nm. of stop bits	Word length select		R/W	0
0xE000C014	U0LSR	U0 Line status register	Rx FIFO Error	TEMT	THRE	BI	FE	PE	OE	DR	RO	0x60
0xE000C01C	U0LSR	U0 Scratch pad register	8-bit data								R/W	0

FE vs. PE vs. OE

- FE：Frame Error, PE：Parity Error, OE：Overrun Error
- Read bits 3, 2 and 1 of a word at memory address 0xE000C014(LSR)
  - If they are all 0 then END
  - Otherwise, then go back to wait

1. **Frame Error (FE) - 幀錯誤：**
   - 幀錯誤發生在接收器無法正確識別通信中的開始位和停止位之間的數據位時。在UART通信中，每個數據字節都有一個開始位和一個或多個停止位，用於標誌數據字節的開始和結束。如果接收器在接收數據時無法準確同步，就會導致幀錯誤。
   - 幀錯誤可能由於多種因素引起，例如通信速率設置不正確、噪聲干擾或線路問題。當出現幀錯誤時，接收器通常會將錯誤標誌設置為FE，並且可能會觸發相應的中斷來通知系統。
2. **Parity Error (PE) - 奇偶校驗錯誤：**
   - 奇偶校驗是一種用於檢測數據錯誤的方法。在某些UART設置中，可以配置奇偶校驗位，以使數據位的總數（包括奇偶校驗位）為奇數或偶數。在接收端，當接收到的數據的奇偶性與預期的不一致時，就會發生奇偶校驗錯誤。
   - 奇偶校驗錯誤通常表示接收到的數據可能已經被損壞或出錯。這可能是由於通信中的噪聲、干擾或其他原因引起的。
3. **Overrun Error (OE) - 溢出錯誤：**
   - 溢出錯誤發生在UART接收緩衝區無法及時處理接收到的數據時。當接收器無法處理數據流的速度，導致接收緩衝區溢出時，就會發生溢出錯誤。
   - 溢出錯誤通常由於接收數據的速度超過系統處理數據的速度而引起。這可能是因為數據傳輸速率太快、接收端處理能力不足或者其他系統性能問題。當發生溢出錯誤時，可能會丟失部分數據，這可能會導致通信錯誤或故障。

UART 0

0xE000C000	U0RBR (DLAB=0)	U0 Receiver buffer register	8-bit data							RO	un–defined
	U0THR (DLAB=0)	U0 Transmit holding register	8-bit data							WO	NA
	U0DLL (DLAB=1)	U0 Divisor latch LSB	8-bit data							R/W	0x01
0xE000C004	U0IER (DLAB=0)	U0 Interrupt enable register	0	0	0	0	0	En. Rx Line Status Int.	Enable THRE Int.	En. Rx Data Av.Int.	R/W	0
	U0DLM (DLAB=1)	U0 Divisor latch LSB	8 bit data								R/W	0
0xE000C008	U0IIR	U0 Interrupt ID register	FIFOs Enabled		0	0	IIR3	IIR2	IIR1	IIR0	RO	0x01
	U0FCR	U0 FIFO control register	Rx Trigger		-	-	-	U0 Tx FIFO Reset	U0 Rx FIFO Reset	U0 FIFO Enable	WO	0
0xE000C00C	U0LCR	U0 Line control register	DLAB	Set break	Stick parity	Even parity select	Parity enable	Nm. of stop bits	Word length select		R/W	0
0xE000C014	U0LSR	U0 Line status register	Rx FIFO Error	TEMT	THRE	BI	FE	PE	OE	DR	RO	0x60
0xE000C01C	U0LSR	U0 Scratch pad register	8-bit data								R/W	0

→參考技術手冊：https://reurl.cc/yY8MV2

Q&A

Thanks for your attention !!