 比較三種方式執行的差異

第一次作業

第一部分

1. 正常印出字串

2. 實現字串反轉

反轉字串副函式

UARTConfig

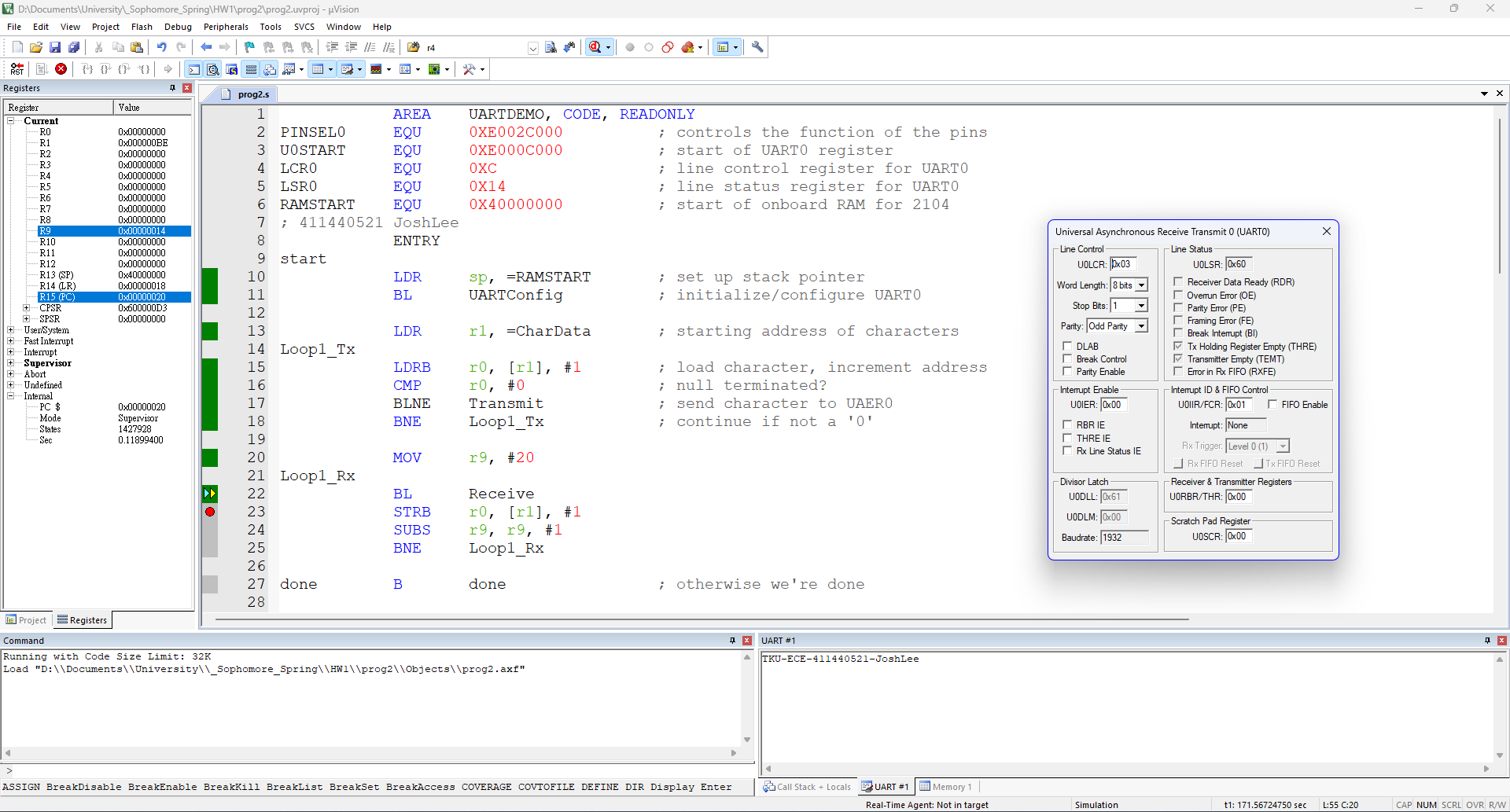
1. 設定Tx和Rx
2. 設定LCR的功能
3. 設定BaudRate = 9375 (因為有捨入誤差)

Transmit程式碼

|  |  |
| --- | --- |
| F5執行時會將字串完整印出 | F10執行時會吃兩個字元 |
|  |  |
| F11則會卡無窮迴圈，無法印出字串 |  |
|  |  |

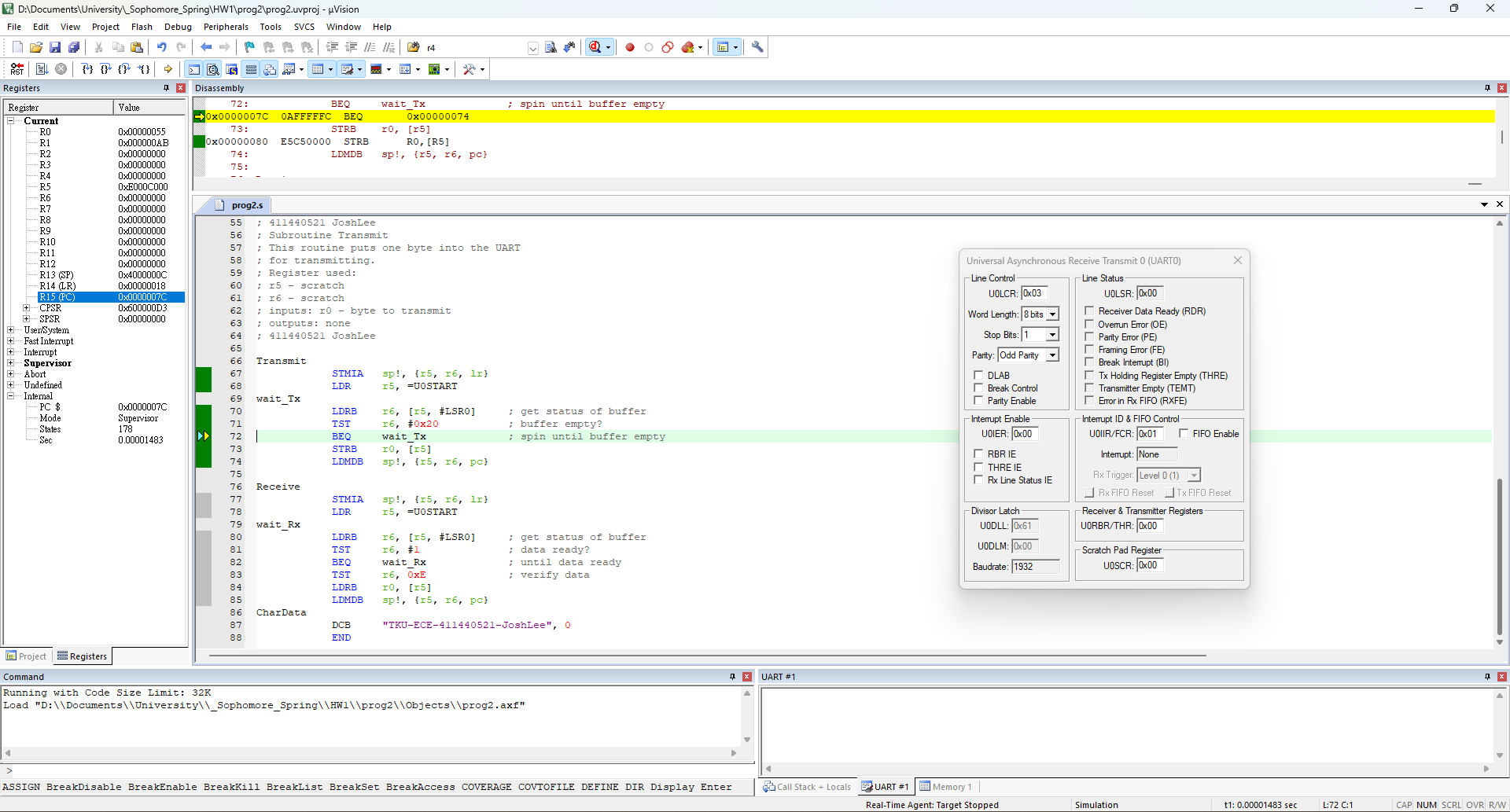
3. Calculate Keil Tool System Clock Frequency and rewrite the initialization to show about 9600 baud on the divisor latch window

|  |  |
| --- | --- |
|  | Claculate Clock Frequency  Clock Frequency = value \* Baud Rate \* 16  = 0x61 \* 1932 \* 16  = 2998464 Hz = 3 MHz |
|  | Initialization Value  Initialization Value = (Clock Frequency / Baud Rate) / 16  = (3M / 9600) / 16  = 0x14 |



第一次作業

第二部分

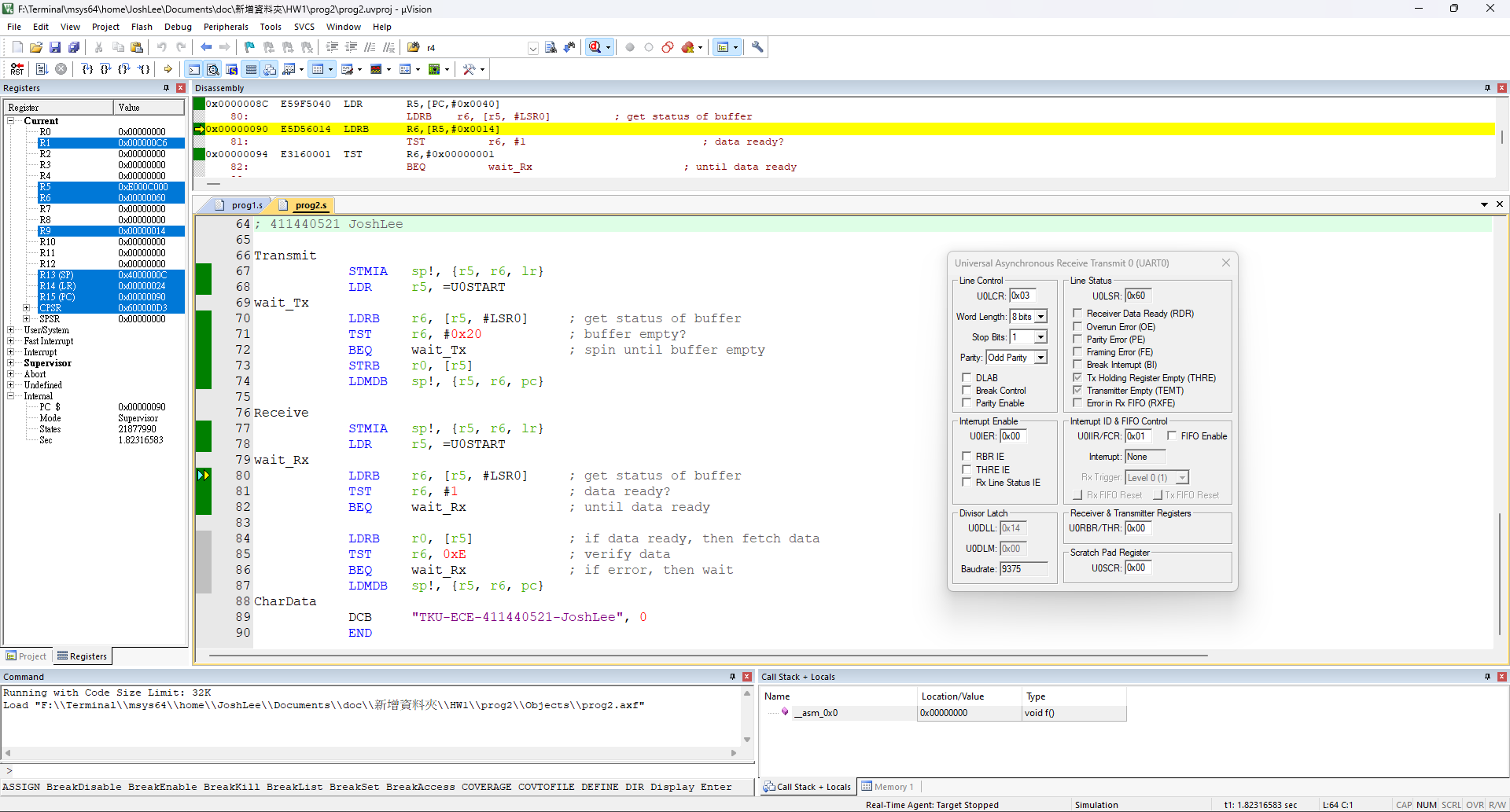


仔細觀察會發現，

F10執行時會在進到Receive副函數時，將最後兩個字元吐出。  
原因是UART在離開Tx時，Tramsmit的Buffer內還有兩個字元，但後續沒有辦法被擠出。碰巧呼叫Rx時會清除Buffer，使最後兩個字元得以一次出現。

用F10的時候，UART會正常輸出字串

1. U0LSR會恢復0x60
2. Tx Holding Register Empty (THRE) = 0
3. Transmitter Empty (TEMT) = 0



因為F11執行時，沒有接Output Device，所以不會印出字串

用F11的時候，UART不會輸出字串

1. U0LSR會變為0x00
2. Tx Holding Register Empty (THRE) = 0
3. Transmitter Empty (TEMT) = 0

沒有錯誤才可Receive Data

報告結束