成功大學資訊工程系微處理機實習上機考二

1. **嚴禁抄襲。若查證屬實，抄襲者與提供抄襲者皆視為零分。**
2. **考試時間: 19：00～21：00。**
3. **請以 C 或 assembly 撰寫，違者不予計分。**

第一題：Timer

題目說明：

程式開始時，LED1 & LED2 同時閃爍。

➢ 限制

- LED1 閃爍頻率：LED2 閃爍頻率＝2：1。
- 時間間隔的部分請以 TIMER 實作。
- LED on/off 在 TIMER ISR 中實作。

➢ 評分標準

LED1 & LED2 同時開始	10%
時間間隔比例正確	20%

➢ 提示

- 不會寫的話可以開始放假了～

第二題：UART & DMA

題目說明：

MSP430 透過 UART 接收從電腦發送的字元，並利用 DMA Channel 0 將字元存起來。

當輸入的字元數量為 5 時，利用 DMA Channel 1 將之前輸入的字元反向印出。

➢ 限制

- 只能使用 DMA ISR。

➢ 評分標準

UART	10%
DMACTL0	10%
DMA Channel 0	20%
DMA Channel 1	30%

➢ 提示

- Edge Triggers or Level Triggers
- Transfer Modes

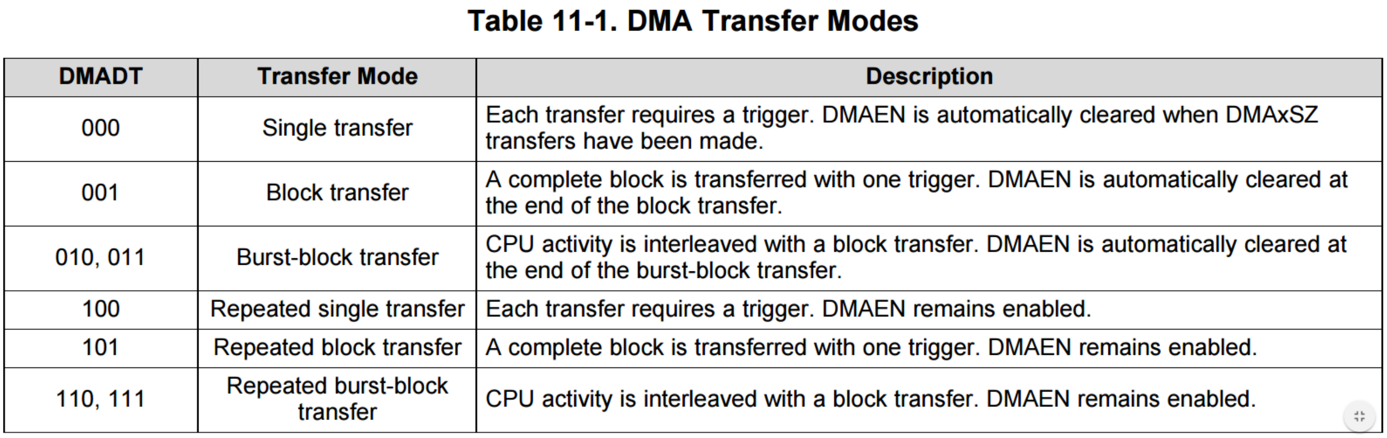

Table 11-1. DMA Transfer Modes

DMADT	Transfer Mode	Description
000	Single transfer	Each transfer requires a trigger. DMAEN is automatically cleared when DMAxSZ transfers have been made.
001	Block transfer	A complete block is transferred with one trigger. DMAEN is automatically cleared at the end of the block transfer.
010, 011	Burst-block transfer	CPU activity is interleaved with a block transfer. DMAEN is automatically cleared at the end of the burst-block transfer.
100	Repeated single transfer	Each transfer requires a trigger. DMAEN remains enabled.
101	Repeated block transfer	A complete block is transferred with one trigger. DMAEN remains enabled.
110, 111	Repeated burst-block transfer	CPU activity is interleaved with a block transfer. DMAEN remains enabled.

範例：

input	13579	Apple	Loser
output	97531	elppA	resoL

```
#include "io430.h"

unsigned char rxbuf[5];
int main( void )
{
  // Stop watchdog timer to prevent time out reset
  WDTCTL = WDTPW + WDTHOLD;
  ????? = ??????;                       //Pin selection
  UCA1CTL1 |= ??????;                   // **Put state machine in reset**
  UCA1CTL1 |= ??????;                   // CLK = SMCLK
//===========set the baud-rate register====
  UCA1BR0 = ??????;
  UCA1BR1 = ??????;
  UCA1MCTL = ??????;
//===========================================
  UCA1CTL1 &= ??????;                  // **Initialize USCI state machine**
  // Setup DMA0 & DMA1
  DMACTL0 = ????? + ?????;//triggered by TxIFG(DMA1) & RxIFG(DMA0) ,hint: datasheet
  DMA0CTL = ?????;
  DMA1CTL = ?????;

  DMA0SZ = 5;
  DMA1SZ = 5;

  DMA0SA = ?????;
  DMA0DA = ?????;
  DMA1SA = ?????;
  DMA1DA = ?????;

  P1DIR=BIT0;
  P4DIR|=BIT7;
  P1SEL=0;

   __bis_SR_register(LPM4_bits + GIE);    // LPM0 with interrupts enabled
    __no_operation();
  return 0;
}
// DMA Interrupt Service Routine
#pragma vector=DMA_VECTOR
__interrupt void DMA_ISR(void)
{
  switch(__even_in_range(DMAIV,16))
  {
    case 0: break;
    case 2:                            // DMA0IFG = DMA Channel 0
      P1OUT ^= BIT0;                    // Toggle P1.0
      ????? |= ??????
      break;
    case 4: // DMA1IFG = DMA Channel 1
      P4OUT ^= BIT7;
      while (!(UCA1IFG&UCTXIFG));            // USCI_A1 TX buffer ready?
       UCA1TXBUF = '\r';               // TX -> RXed character
      while (!(UCA1IFG&UCTXIFG));
       UCA1TXBUF = '\n';
      break;
    default: break;
  }
}
```