Anwendungsbeispiel

- Eine Applikation auf einem MSP430 soll einen freilaufenden Zähler periodisch mit der Frequenz von 1 Hz inkrementieren und seinen Inhalt auf einer vierstelligen Siebensegmentanzeige darstellen.
- Die vierstelligen Siebensegmentanzeige wird mit einem Treiberbaustein AS1108 gesteuert.
- Die Schnittstelle zwischen dem MSP430 und dem AS1108 ist eine P2P-SPI-Verbindung. Laut dem Datenblatt für AS1108 beträgt die SPI-Taktperiode mindestens 100 ns, was einer Taktfrequenz von 10 MHz entspricht. In dieser Applikation soll der SPI-Takt allerdings mit nur ca. 100 kHz laufen.

Anwendungsbeispiel

- Auszug auf der Initialisierungssequenz von UCA

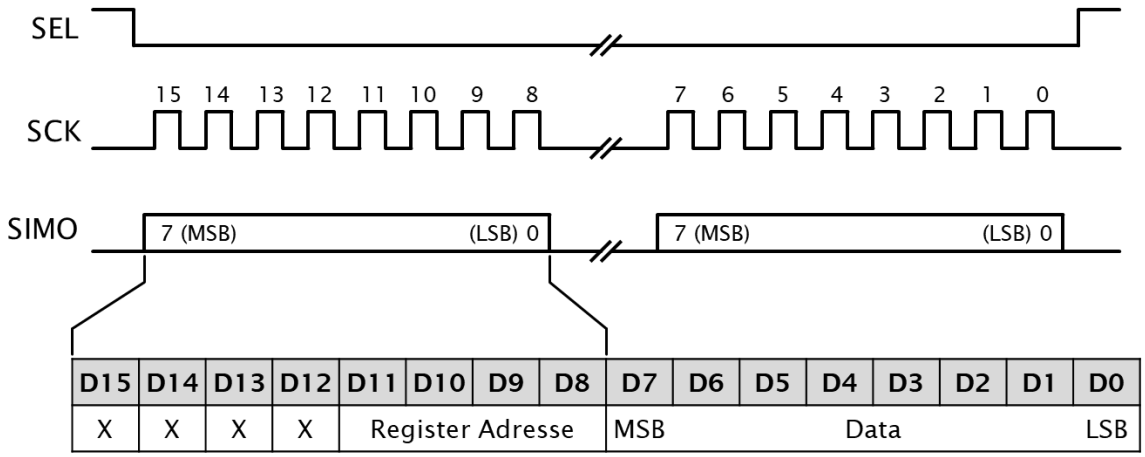
Beispiel 8.1

```
// Port 2: Pin 3          => SPI.CS output, idle High
// Port 2: Pin 4, 5 and 6 => SPI
...

#pragma FUNC_ALWAYS_INLINE(UCA1_init)
GLOBAL Void UCA1_init(Void) {
// set up Universal Serial Communication Interface A
   SETBIT(UCA1CTLW0, UCSWRST);  // UCA1 software reset
   UCA1BRW = 6;                 // prescaler

// in Übereinstimmung mit dem SPI-Timing-Diagramm von AS1108
   UCA1CTLW0 = UCCKPH           // 15: clock phase select: rising edge
             | 0                // 14: clock polarity: inactive low
             | UCMSB            // 13: MSB first
             | 0                // 12: 8-bit data
             | UCMST            // 11: SPI master mode
             | UCMODE_0         // 10-9: mode select: 3-pin SPI
             | UCSYNC           // 8:  synchronous mode enable
             | UCSSEL__ACLK     // 7-6: clock source select
             | 0;               // 0: release the UCA0 for operation

// ggf. weitere lokale Variablen initialisieren

}
```

Anwendungsbeispiel

Beispiel 8.1

- Schreibzugriffe auf den AS1108 basierend auf dem Timing



D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	Register Adresse				MSB	Data						LSB

```
Void UCA1_emit(const UChar adr, const UChar val) {
   UChar ch = UCA1RXBUF;          // RXBUF auslesen, UCRXIFG := 0, UCOE := 0
   CLRBIT(P2OUT, BIT3);           // Select aktivieren
   UCA1TXBUF = adr;               // Ausgabe einer Registeradresse

   while (NOT TSTBIT(UCA1IFG, UCRXIFG)) ;

   ch = UCA1RXBUF;
   UCA1TXBUF = val;               // Ausgabe eines Datums

   while (NOT TSTBIT(UCA1IFG, UCRXIFG)) ;

   ch = UCA1RXBUF;
   SETBIT(P2OUT, BIT3);           // Select deaktivieren
}
```

Anwendungsbeispiel

- Initialisierungssequenz für den AS1108

Beispiel 8.1

```
#define INITSIZE 10

typedef struct {
   UChar adr;
   UChar val;
} TFrame;

LOCAL const TFrame init[INITSIZE] = {
   { 0x0E, 0x0C },  // internal oscillator, enable B/HEX decoding, enable SPI
   { 0x0C, 0x81 },  // shutdown register := normal mode
   { 0x0F, 0x00 },  // normal mode
   { 0x01, 0x00 },
   { 0x02, 0x00 },
   { 0x03, 0x00 },
   { 0x04, 0x00 },
   { 0x09, 0xFF },  //
   { 0x0A, 0x03 },  // intensity 7/32
   { 0x0B, 0x03 }   // display all numbers
};

...

UInt i;
for(i=0; i LT INITSIZE; i++) {
   UCA1_emit(init[i].adr, init[i].val);
}
```

Anwendungsbeispiel

- Verarbeitung der Daten in einer Timer-ISR

Beispiel 8.1

```
LOCAL UInt counter;

#pragma vector = TIMER0_A1_VECTOR
__interrupt Void TIMER0_A1(Void) {
   UInt i;
   UInt tmp = counter;
   CLRBIT(TA0CTL, TAIFG); // clear interrupt flag
   for(i=1; i LE 4; i++) {
      UChar ch = 0x0F BAND tmp;
      ch += '0';
      UCA1_emit(i, ch);
      tmp >>= 4;
   }
   counter += 1;
}
```

- Durch die for-Schleife und die beiden while-Schleifen in der UCA1_emit-Funktion blockiert die ISR für 632 us andere (auch höher priorisierte) ISR sowie andere Handler.

Anwendungsbeispiel

Beispiel 8.2

- Der Timer inkrementiert einen Zähler und erzeugt Events, die an einen Handler (in der main-Funktion) delegiert werden.
- Der Handler reagiert auf diese Evens, formatiert die Ausgabe und schreibt die Daten über die SPI-Schnittstelle in den Registersatz von AS1108 rein.

```
GLOBAL UInt counter;
...

#pragma vector = TIMER0_A1_VECTOR
__interrupt Void TIMER0_A1(Void) {
   CLRBIT(TA0CTL, TAIFG);
   counter += 1;
   Event_set(EVENT_UPDATE);
   __low_power_mode_off_on_exit();
}
```

```
GLOBAL Void main(Void) {
   CS_init();
   GPIO_init();
   Event_init();
   UCA1_init();
   TA0_init();

   while(TRUE) {
      Event_wait();
      Handler1();
      if (Event_err()) {
         SETBIT(P1OUT, BIT2);
      }
   }
}
```

```
LOCAL Void Handler1(Void) {
   UInt i;
   if (Event_tst(EVENT_UPDATE)) {
      Event_clr(EVENT_UPDATE);
      UInt tmp = counter;
      for(i=1; i LE 4; i++) {
         UChar ch = 0x0F BAND tmp;
         ch += '0';
         UCA1_emit(i, ch);
         tmp >>= 4;
      }
   }
}
```

- Die ISR selbst ist nicht mehr blockierend, aber der Handler!

Anwendungsbeispiel

- Die while-Schleifen aus der Schreibfunktion müssen durch eine Interrupt-gesteuerte Ausgabe implementiert werden.

Beispiel 8.3

```
#define DATASIZE 2
LOCAL UInt  idx;              // Index
LOCAL UChar data[DATASIZE];   // Pointer auf das Datenfeldd

GLOBAL Void UCA1_emit(const UChar adr, const UChar val) {
   idx  = 0;
   data[0] = adr;
   data[1] = val;
   SETBIT(UCA1IFG, UCRXIFG); // indirekter Aufruf der ISR
}

#pragma vector = USCI_A1_VECTOR
__interrupt Void UCA1_ISR(Void) {
   UChar ch = UCA1RXBUF;          // RXBUF auslesen, UCRXIFG := 0, UCOE := 0
   CLRBIT(P2OUT, BIT3);           // Select aktivieren
   if (idx GE DATASIZE) {
      SETBIT(P2OUT,  BIT3);       // Select deaktivieren
      Event_set(EVENT_DONE);      // Event senden
      __low_power_mode_off_on_exit();
   } else {
      UCA1TXBUF = data[idx++];    // nächstes Byte ausgeben
   }
}
```

Anwendungsbeispiel

- Die for-Schleife aus dem Handler ist mit Hilfe einer Zustandsmaschine zu implementieren.

Beispiel 8.3

```
// Datentyp eines Funktionspointers
typedef Void (* VoidFunc)(Void);

LOCAL Void State0(Void);
LOCAL Void State1(Void);

// lokale Zustandsvariablen
LOCAL VoidFunc state;
LOCAL UInt idx;
LOCAL UInt tmp;

#pragma FUNC_ALWAYS_INLINE(Handler1_init)
GLOBAL Void Handler1_init(Void) {
   state = State0;
}


#pragma FUNC_ALWAYS_INLINE(Handler1)
GLOBAL Void Handler1(Void) {
   (*state)();
}
```

```
LOCAL Void State0(Void) {
   if (Event_tst(EVENT_UPDATE)) {
      Event_clr(EVENT_UPDATE);
      tmp = counter;
      idx = 1;
      state = State1;
      Event_set(EVENT_DONE);
   }
}

LOCAL Void State1(Void) {
   if (Event_tst(EVENT_DONE)) {
      Event_clr(EVENT_DONE);
      if (idx LE 4) {
         UChar ch = 0x0F BAND tmp;
         ch += '0';
         UCA1_emit(idx, ch);
         tmp >>= 4;
         idx++;
      } else {
         state = State0;
      }
   }
}
```