CT /MP

Adresscodierung:

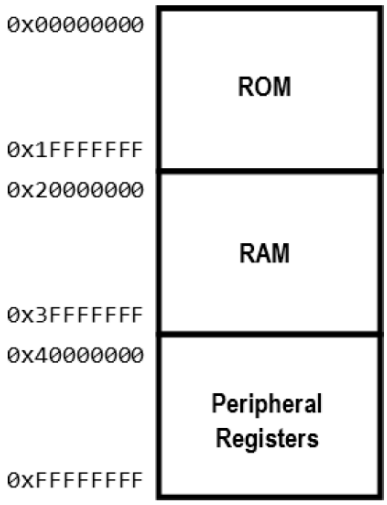
1 Kbyte = 1024 Byte, 1Mbyte = 1024 Kbyte = 1.048.576 Byte

Größe:

$$ROM = 2 * 2^{(7*4)} = 2 * 268.435.456\ Adressen = 536.870.912\ Byte$$
$$= 524.288\ KByte = 512\ MByte = 0{,}5\ GByte$$

Erklärung:

Die Größe der Speicher berechnet sich durch die Anzahl der Adressen.
Diese können so berechnet werden:
Zwei hoch die Anzahl der Hex-Stellen, die sich komplett ändern
(0 → F) mal 4, hier: 7 Stellen * 4
Dieses Ergebnis multipliziert mit den Stellen die sich nicht komplett ändern.
Hier: 2 (0000 und 0001)



Memory Map

UART (Universal Asynchronous Receiver Transmitter)

Übertragungsstandard

TTL/CMOS-Level : 0=0V, 1=3,3V, unipolar, positive Logik
RS232 : 0=+3..+15V, 1=-3..-15V, bipolar, negative/invertierte Logik

Format

D/P/S
Anzahl der Datenbits : **5,6,7,8,9**
Paritätsbit : **N**=None/keins, **O**=Odd/Ungerade, **E**=Even/Gerade
Anzahl der Stoppbits : **1** oder **2**

z.B.: **7O1** = 7 Datenbits, ein Paritätsbit für ungerade Parität und ein Stopbit.

Paritätsbit

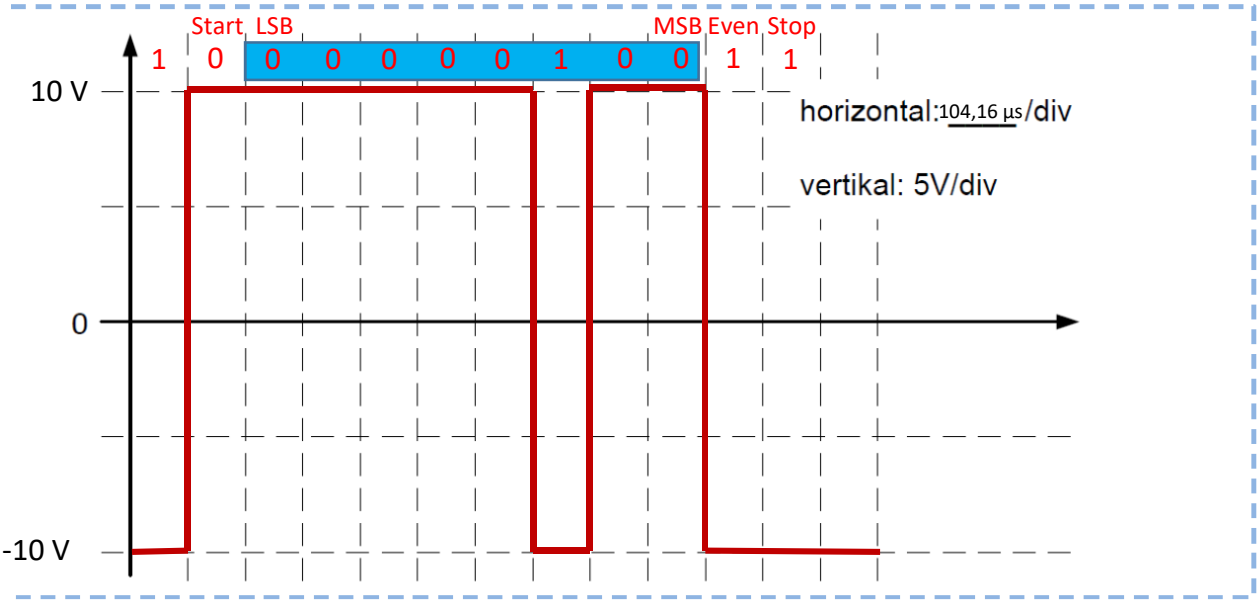
N = Kein Paritätsbit
E = Gerade. Nach dem Hinzufügen des Paritätsbits enthält der Datenrahmen (ohne Start/Stopp-Bit) eine gerade Anzahl von Einsen. (000 0000 → Parität = 0)
O = Ungerade. Nach dem Hinzufügen des Paritätsbits enthält der Datenrahmen (ohne Start/Stopp-Bit) eine ungerade Anzahl von Einsen. (000 0000 → Parität = 1)

Empfangsfehler

- Paritätsfehler: (Rx detektiert ein anderes Paritätsbit als erwartet)
  - Ungerade Anzahl Bitfelder treten während der Übertragung auf.
  - Rx und Tx sind auf verschiedene Paritätsmodi, Bitraten oder Anzahl Datenbit eingestellt.
- Framing Fehler: (Stoppbit ist nicht 1)
  - Bitfehler tritt während der Übertragung auf.
  - Rx und Tx haben verschiedene Bitraten oder Rahmenformate.

- Breakfehler: (das empfangene Signal war länger als die Dauer eines Frames)
    - Kabelbruch, Tx ausgeschaltet oder hat eine „line-break“ Anweisung gesendet.

■ Das Symol ‘@’ wird im 8/E/1 Format mit 9.6kbit/s übertragen: Skizzieren Sie das RS232-Signal des Datenrahmens und beschriften Sie die Achsen. Beschriften Sie die Bits des Rahmens, insbesondere auch LSB und MSB.



‚@‘ = 0x40 = 0100 000 , Horizontal Achse : $\frac{1}{9600\ bit/s} = 104{,}16\ \mu s$

$\text{DIVINT} = \left(\frac{\frac{Frequenz}{16}}{Boudrate}\right) runden \rightarrow z.b.: \left(\frac{\frac{16MHz}{16}}{9600\ bit/s}\right) runden = 104{,}1\overline{66} = 104$

$\text{DIVFRAC} = runden((DIVINT - Ganzzahl\ Anteil) * 2^6) \rightarrow\ z.b.: \text{runden}(0{,}167 * 2^6) = 10{,}\overline{666} = 11$

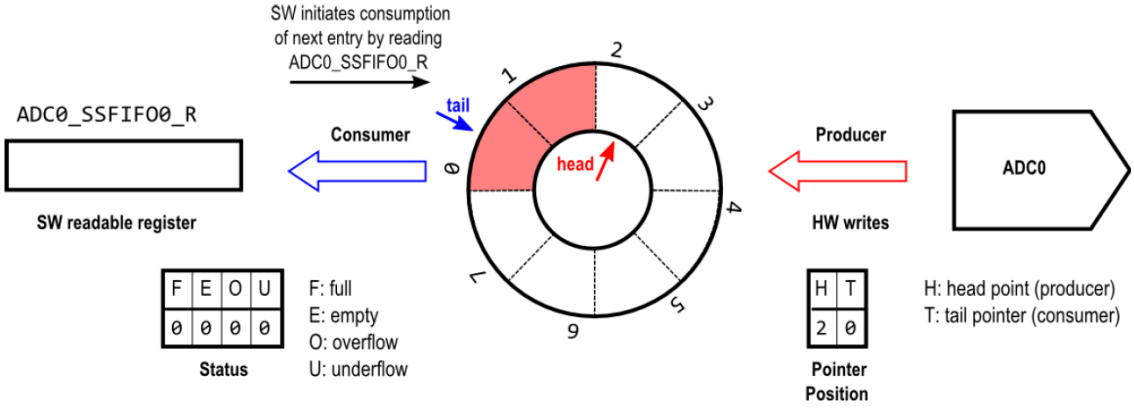
Programm für UART2-Modul:

```
SYSCTL_RCGC1_R |= 0x4;  //Betriebsspannung und Systemtakt an bei UART2
UART2_CTL_R &= ~0x1;    // UART2 deaktivieren
UART2_IBRD_R = 104;     //DIVINT der BRD
UART25_FBRD_R = 11;     //DIVFRAC der BRD
UART2_LCRH_R |= 0x66;   //Format: 8E1
UART2_CTL_R |= 0x1;     // UART2 aktivieren
```

```
Mögliche Ports: PE4(5), PD1(4), PG1(1), PD6(9). Hier nehmen wir PG1(1)
```

```
SYSCTL_RCGC2_R |= 0x40;
GPIO_PORTG_DEN_R |= 0x02;
GPIO_PORTG_AFSEL_R |= 0x02;
GPIO_PORTG_PCTL_R |= 0x0000 0010;
```

ADC / FIFO



Status

FULL:	1 = voll, 0 = nicht voll (Register erzeugt dann keine weiteren Werte)
EMPTY:	1 = leer, 0 = nicht leer (Register enthält dann „Not valid“)
Overflow:	Ist 1, wenn das Register versucht ein bereits leeren FIFO ein weiteres Mal zu lesen. T.-Pointer und H.-Pointer zeigen auf selbe Position
Underflow:	Ist 1, wenn das Register versucht ein bereits volles FIFO ein weiteres Mal zu belegen. H.-Pointer zeigt auf einen Platz unterhalb des T.-Pointers und ADC versucht in den vollen Speicher weitere Werte einzulesen.
Headpointer:	Position an die der ADU als nächstes schreibt.
Tailpointer:	Eintrag der als nächstes von der SW ausgelesen wird. Wenn der FIFO leer ist führen weitere Leseversuche zu keiner Änderung der Position des Tail-Pointers.

Bedingung um Unterlauf zu verhindern:

```
While (ADC1_SSFSTAT0_R & (1<<8));
```