```
Init_UART0_Polling PROC {}
        PUSH {R1-R2}

        PORT_PCR_SET_PTB2_UART0_RX EQU (PORT_PCR_ISF_MASK :OR:
        PORT_PCR_MUX_SELECT_2_MASK)
        PORT_PCR_SET_PTB1_UART0_TX EQU (PORT_PCR_ISF_MASK :OR:
        PORT_PCR_MUX_SELECT_2_MASK)


        SIM_SOPT2_UART0SRC_MCGFLLCLK EQU (1 << SIM_SOPT2_UART0SRC_SHIFT)

        SIM_SOPT5_UART0_EXTERN_MASK_CLEAR EQU \
           (SIM_SOPT5_UART0ODE_MASK :OR: \
            SIM_SOPT5_UART0RXSRC_MASK :OR: \
            SIM_SOPT5_UART0TXSRC_MASK)

        UART0_BDH_9600 EQU 0x01
        UART0_BDL_9600 EQU 0x38

        UART0_C1_8N1             EQU 0x00
        UART0_C3_NO_TXINV          EQU 0x00
        UART0_C4_OSR_16          EQU 0x0F
        UART0_C4_NO_MATCH_OSR_16  EQU UART0_C4_OSR_16
        UART0_C5_NO_DMA_SSR_SYNC  EQU 0x00

        UART0_C2_T_R EQU (UART0_C2_TE_MASK :OR: UART0_C2_RE_MASK)

        UART0_S1_CLEAR_FLAGS EQU \
           (UART0_S1_IDLE_MASK :OR: \
            UART0_S1_OR_MASK   :OR: \
            UART0_S1_NF_MASK   :OR: \
            UART0_S1_FE_MASK   :OR: \
            UART0_S1_PF_MASK)

        UART0_S2_NO_RXINV_BRK10_NO_LBKDETECT_CLEAR_FLAGS EQU \
           (UART0_S2_LBKDIF_MASK :OR: UART0_S2_RXEDGIF_MASK)

        LDR   R1, =SIM_SOPT2
        LDR   R2, =SIM_SOPT2_UART0SRC_MASK
        LDR   R3, [R1]
        BICS  R3, R3, R2
        LDR   R2, =SIM_SOPT2_UART0SRC_MCGFLLCLK
        ORRS  R3, R3, R2
        STR   R3, [R1]
```

```
; Set UART0 for external connection
LDR   R1, =SIM_SOPT5
LDR   R2, =SIM_SOPT5_UART0_EXTERN_MASK_CLEAR
LDR   R3, [R1]
BICS  R3, R3, R2
STR   R3, [R1]

; Enable UART0 module clock
LDR   R1, =SIM_SCGC4
LDR   R2, =SIM_SCGC4_UART0_MASK
LDR   R3, [R1]
ORRS  R3, R3, R2
STR   R3, [R1]

; Enable PORT B module clock
LDR   R1, =SIM_SCGC5
LDR   R2, =SIM_SCGC5_PORTB_MASK
LDR   R3, [R1]
ORRS  R3, R3, R2
STR   R3, [R1]

; Configure PORTB pins
LDR   R1, =PORTB_PCR2
LDR   R2, =PORT_PCR_SET_PTB2_UART0_RX
STR   R2, [R1]

LDR   R1, =PORTB_PCR1
LDR   R2, =PORT_PCR_SET_PTB1_UART0_TX
STR   R2, [R1]

; Disable UART0
LDR   R1, =UART0_BASE
MOVS  R2, #UART0_C2_T_R
LDRB  R3, [R1, #UART0_C2_OFFSET]
BICS  R3, R3, R2
STRB  R3, [R1, #UART0_C2_OFFSET]

; Set baud rate
MOVS  R2, #UART0_BDH_9600
STRB  R2, [R1, #UART0_BDH_OFFSET]

MOVS  R2, #UART0_BDL_9600
STRB  R2, [R1, #UART0_BDL_OFFSET]
```

```
      MOVS  R2, #UART0_C1_8N1
      STRB  R2, [R1, #UART0_C1_OFFSET]

      MOVS  R2, #UART0_C3_NO_TXINV
      STRB  R2, [R1, #UART0_C3_OFFSET]

      MOVS  R2, #UART0_C4_NO_MATCH_OSR_16
      STRB  R2, [R1, #UART0_C4_OFFSET]

      MOVS  R2, #UART0_C5_NO_DMA_SSR_SYNC
      STRB  R2, [R1, #UART0_C5_OFFSET]

      MOVS  R2, #UART0_S1_CLEAR_FLAGS
      STRB  R2, [R1, #UART0_S1_OFFSET]

      MOVS  R2, #UART0_S2_NO_RXINV_BRK10_NO_LBKDETECT_CLEAR_FLAGS
      STRB  R2, [R1, #UART0_S2_OFFSET]

      ; Enable UART0
      MOVS  R2, #UART0_C2_T_R
      STRB  R2, [R1, #UART0_C2_OFFSET]


      POP  {R1-R3, LR}
      BX   LR


PutChar        PROC {}

    PUSH    {R1-R3, LR}
    LDR       R1, =UART0_BASE
    MOVS    R2, #UART0_S1_RDRF_MASK

PollRx
    LDRB    Rm, [R1, #UART0_S1_OFFSET]
    ANDS    Rm, Rm, R2
    BEQ     PollRx

    LDRB    R0, [R1, #UART0_D_OFFSET]
    POP     {R1-R3, LR}
    BX      LR
    ENDP
```

```
GetChar      PROC {}

        LDR     R1, =UART0_BASE
        MOVS    R2, #UART0_S1_TDRE_MASK

PollTx
        LDRB    Rm, [R1, #UART0_S1_OFFSET]
        ANDS    Rm, Rm, R2
        BEQ     PollTx

        STRB    R0, [R1, #UART0_D_OFFSET]
        POP     {R1-R3, LR}
        BX      LR
        ENDP
```