```
' ============================================================================================================
'
'   File..... DynamicVP_DEF.SXB
'   Compiler.. SXB 2.00.16
'   Purpose... Library to add virtual peripherals to SXB 2.x applications
'   Author.... Peter Verkaik
'   E-mail.... verkaik6@zonnet.nl
'   Started... Feb 5, 2009
'   Updated... Mar 21, 2009
'
' ============================================================================================================

' ------------------------------------------------------------------------------------------------------------
' Requirements
' ------------------------------------------------------------------------------------------------------------
' This library requires the following libraries:
' SXB_ConMac.SXB, PortPins_DEF.SXB + PortPins_INC.SXB

' ------------------------------------------------------------------------------------------------------------
' Library Description
' ------------------------------------------------------------------------------------------------------------
' This library provides code that implements dynamic VP loading.
' There are up to six VP banks, each VP bank occupies an SX ram bank (16 bytes)
' and each VP bank can be used for any of the included VP drivers.
' VP drivers can be started and stopped. Once stopped, the VP bank of
' the stopped driver can be re-used by starting a new driver.
'
' Virtual Peripherals are started using vpHandle = dvpStartVP label_of_declaration
' The returned value must be kept as it is required by subsequent calls to VP functions.
' If the returned value is 0 then there are no free VP banks and the VP is not started.
'
' Virtual Peripherals are stopped using dvpStopVP vpHandle
' The VP bank occupied by the Virtual Peripheral is released so another VP can be started.
'
' Up to dvpVPbanks (declared in VAR section) Virtual Peripherals can run simultaneously.
' But you can define as many VP's as memory serves. For example, you can declare 3 Serial Transmit
' VP's and use only 1 VP bank. By starting a Serial Transmit VP only when you need to send data
' the one VP bank can be shared by all Serial Transmit VP's (or other types of VP).
'
' You can have multiple instances of the same type of VP.

'************************************************************************************************
'CON section
'************************************************************************************************

'Timer ID's
dvpT8_0         con     $00     ' 8bit timer
dvpT8_1         con     $01     ' 8bit timer
dvpT8_2         con     $02     ' 8bit timer
dvpT8_3         con     $03     ' 8bit timer
dvpT8_4         con     $04     ' 8bit timer
dvpT8_5         con     $05     ' 8bit timer
dvpT8_6         con     $06     ' 8bit timer
dvpT8_7         con     $07     ' 8bit timer
dvpT16_0        con     $10     '16bit timer (uses T8_0 and T8_1)
dvpT16_1        con     $12     '16bit timer (uses T8_2 and T8_3)
dvpT16_2        con     $14     '16bit timer (uses T8_4 and T8_5)
dvpT16_3        con     $16     '16bit timer (uses T8_6 and T8_7)

'Serial pinmodes
dvpDTHT         con     $00     'datapin true,   handshakepin true
dvpDIHT         con     $20     'datapin invert, handshakepin true
dvpDTHI         con     $40     'datapin true,   handshakepin invert
dvpDIHI         con     $60     'datapin invert, handshakepin invert

'Serial Transmit/Receive VP baudrates
dvp115200       con     INTERRUPTRATE/115200
dvp76800        con     INTERRUPTRATE/76800
dvp57600        con     INTERRUPTRATE/57600
dvp38400        con     INTERRUPTRATE/38400
dvp28800        con     INTERRUPTRATE/28800
dvp19200        con     INTERRUPTRATE/19200
dvp14400        con     INTERRUPTRATE/14400
dvp9600         con     INTERRUPTRATE/9600
dvp7200         con     INTERRUPTRATE/7200
dvp4800         con     INTERRUPTRATE/4800
dvp3600         con     INTERRUPTRATE/3600
dvp2400         con     INTERRUPTRATE/2400
dvp1800         con     INTERRUPTRATE/1800
dvp1200         con     INTERRUPTRATE/1200

'Serial data formats
dvpE71          con     %00001010       ' even parity, 7 databits, 1 stopbit , 1 startbit, 10 bits total
dvpE72          con     %00001111       ' even parity, 7 databits, 2 stopbit , 1 startbit, 11 bits total
dvpE81          con     %00001101       ' even parity, 8 databits, 1 stopbit , 1 startbit, 11 bits total
dvpO71          con     %00010010       ' odd  parity, 7 databits, 1 stopbit , 1 startbit, 10 bits total
dvpO72          con     %00010111       ' odd  parity, 7 databits, 2 stopbit , 1 startbit, 11 bits total
dvpO81          con     %00010101       ' odd  parity, 8 databits, 1 stopbit , 1 startbit, 11 bits total
dvpN72          con     %00000010       ' no   parity, 7 databits, 2 stopbit , 1 startbit, 10 bits total
dvpN81          con     %00000000       ' no   parity, 8 databits, 1 stopbit , 1 startbit, 10 bits total
dvpN82          con     %00000101       ' no   parity, 8 databits, 2 stopbit , 1 startbit, 11 bits total
```

```
'*************************************************************************************************
'SUB/FUNC/TASK declarations
'*************************************************************************************************

' ================================================================================================

'{$ifused dvpStartVP}
  'Description: Start a VP driver
  'Use:         bytevar = dvpStartVP vpDeclaration
  'Parameters:  (word) Address of VP declaration
  'Returns:     (byte) vpbank if succes, 0 if no free vpbanks
  '{$uses ppDIR_Input}
  '{$uses ppDIR_Output}
  '{$uses ppPIN_Low}
  '{$uses ppPIN_High}
' ------------------------------------------------------------------------------------------------
  dvpStartVP            FUNC    1,2
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpStopVP}
  'Description: Stop a VP driver
  'Use:         dvpStopVP vpbank
  'Parameters:  (byte) vpbank previously returned by dvpStartVP
  'Returns:     none
' ------------------------------------------------------------------------------------------------
  dvpStopVP             SUB     1
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpTMR_SetTick}
  'Description: Set timer tick
  'Use:         dvpTMR_SetTick vpbank,value
  'Parameters:  (byte) vpbank, (word) value
  'Returns:     none
' ------------------------------------------------------------------------------------------------
  dvpTMR_SetTick        SUB     3,3,byte,word
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpTMR_Mark}
  'Description: Mark timer
  'Use:         dvpTMR_Mark vpbank,timerID
  'Parameters:  (byte) vpbank, (byte) timerID (dvpT8_0 to dvpT8_7, dvpT16_0 to dvpT16_3)
  'Returns:     none
  'Returns:     none
' ------------------------------------------------------------------------------------------------
  dvpTMR_Mark           SUB     2
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpTMR_Timeout}
  'Description: Test for timer timeout
  'Use:         dvpTMR_Timeout vpbank,timerID,timeoutValue
  'Parameters:  (byte) vpbank, (byte) timerID (dvpT8_0 to dvpT8_7, dvpT16_0 to dvpT16_3), (word) timeoutValue
  'Returns:     (byte) 0 if no timeout, $FF if timeout
' ------------------------------------------------------------------------------------------------
  dvpTMR_Timeout        FUNC    1,4,4,byte,byte,byte,word
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpDAC_SetValue}
  'Description: Set DAC value
  'Use:         dvpDAC_SetValue vpbank, value
  'Parameters:  (byte) vpbank, (byte) value
  'Returns:     none
' ------------------------------------------------------------------------------------------------
  dvpDAC_SetValue       SUB     2
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpPWM_SetValue}
  'Description: Set PWM value
  'Use:         dvpPWM_SetValue vpbank, lowtime_LSB, lowtime_MSB, hightime_LSB, hightime_MSB
  '             lowtime >= 2, hightime >= 2, pwm frequency = INTERRUPTRATE/(lowtime+hightime)
  'Parameters:  (byte) vpbank, (byte) lowtime_LSB, (byte) lowtime_MSB, (byte) hightime_LSB, (byte) hightime_MSB
  'Returns:     none
' ------------------------------------------------------------------------------------------------
```

```
  dvpPWM_SetValue       SUB     5
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpADC_GetValue}
  'Description: Get ADC value
  'Use:         dvpADC_GetValue vpbank
  'Parameters:  (byte) vpbank
  'Returns:     (byte) value
' ------------------------------------------------------------------------------------------------
  dvpADC_GetValue       FUNC    1,1
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpSTX_SendBufferEmpty}
  'Description: Test if buffer is empty and nothing is being sent
  'Use:         dvpSTX_SendBufferEmpty vpbank
  'Parameters:  (byte) vpbank
  'Returns:     (byte) 0 if not empty, $FF if empty
' ------------------------------------------------------------------------------------------------
  dvpSTX_SendBufferEmpty        FUNC    1,1
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpSTX_SendByte}
  'Description: Send a byte
  'Use:         dvpSTX_SendByte vpbank, value
  'Parameters:  (byte) vpbank, (byte) value
  'Returns:     none
  '{$uses dvpSER_EvenParity}
' ------------------------------------------------------------------------------------------------
  dvpSTX_SendByte       SUB     2
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpSRX_ByteAvailable}
  'Description: Test if byte received
  'Use:         dvpSRX_ByteAvailable vpbank
  'Parameters:  (byte) vpbank
  'Returns:     (byte) 0 if no byte received, $FF if byte received
' ------------------------------------------------------------------------------------------------
  dvpSRX_ByteAvailable  FUNC    1,1
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpSRX_ReceiveByte}
  'Description: Get byte
  '             There is no validity check on received parity bits.
  'Use:         dvpSRX_ReceiveByte vpbank
  'Parameters:  (byte) vpbank
  'Returns:     (byte) byte received
' ------------------------------------------------------------------------------------------------
  dvpSRX_ReceiveByte    FUNC    1,1
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpSER_OddParity}
  'Description: Calculate odd parity
  'Use:         dvpSER_OddParity value
  'Parameters:  (byte) value
  'Returns:     (byte) Odd parity bit in __PARAM1.7
  'Modifies:    __PARAM1
  '{$uses dvpSER_EvenParity}
' ------------------------------------------------------------------------------------------------
  dvpSER_OddParity      FUNC    1,1
' ------------------------------------------------------------------------------------------------
'{$endif}

' ================================================================================================

'{$ifused dvpSER_EvenParity}
  'Description: Calculate even parity
  'Use:         dvpSER_EvenParity value
  'Parameters:  (byte) value
  'Returns:     (byte) Even parity bit in __PARAM1.7
  'Modifies:    __PARAM1
' ------------------------------------------------------------------------------------------------
  dvpSER_EvenParity     FUNC    1,1
' ------------------------------------------------------------------------------------------------
```

```
'{$endif}

' ============================================================================================================

'Description: Declare a Serial Transmit VP
'Use:         \dvpSTX_Declare name,dataport,datapin,hsport,hspin,pinmode,baud,buffer,format
'Parameters:  name    : name that will be used as label for the generated datatable
'             dataport: port to which the datapin belongs (ppPortA,ppPortB,ppPortC,ppPortD,ppPortE)
'                       You may add $80 to set the pin latch register high.
'             datapin : pin used for data output (ppPin0,ppPin1,ppPin2,ppPin3,ppPin4,ppPin5,ppPin6,ppPin7)
'             hsport  : port to which the handshake pin belongs (0 if no handshake)
'             hspin   : pin used for handshake input (0 if no handshake)
'             pinmode : data and handshake true/invert (dvpDTHT,dvpDTHI,dvpDIHT,dvpDIHI)
'             baud    : baudrate (dvp115200 .. dvp1200) (value equals INTERRUPTRATE/baud)
'             buffer  : name of bytebuffer (0 if no buffer)
'             format  : byteprotocol (dvpE71,dvpE72,dvpE81,dvpO71,dvpO72,dvpO81,dvpN72,dvpN81,dvpN82)
'Returns:     none
' ------------------------------------------------------------------------------------------------------------
dvpSTX_Declare macro name,dataport,datapin,hsport,hspin,pinmode,baud,buffer,format
' ------------------------------------------------------------------------------------------------------------

' ============================================================================================================

'Description: Declare a Serial Receive VP
'Use:         \dvpSRX_Declare name,dataport,datapin,hsport,hspin,pinmode,baud,buffer,format
'Parameters:  name    : name that will be used as label for the generated datatable
'             dataport: port to which the datapin belongs (ppPortA,ppPortB,ppPortC,ppPortD,ppPortE)
'                       You may add $80 to set the pin latch register high.
'             datapin : pin used for data input (ppPin0,ppPin1,ppPin2,ppPin3,ppPin4,ppPin5,ppPin6,ppPin7)
'             hsport  : port to which the handshake pin belongs (0 if no handshake)
'             hspin   : pin used for handshake output (0 if no handshake)
'             pinmode : data and handshake true/invert (dvpDTHT,dvpDTHI,dvpDIHT,dvpDIHI)
'             baud    : baudrate (dvp115200 .. dvp1200) (value equals INTERRUPTRATE/baud)
'             buffer  : name of bytebuffer (0 if no buffer)
'             format  : byteprotocol (dvpE71,dvpE72,dvpE81,dvpO71,dvpO72,dvpO81,dvpN72,dvpN81,dvpN82)
'Returns:     none
' ------------------------------------------------------------------------------------------------------------
dvpSRX_Declare macro name,dataport,datapin,hsport,hspin,pinmode,baud,buffer,format
' ------------------------------------------------------------------------------------------------------------

' ============================================================================================================

'Description: Declare an Analogue to Digital VP
'Use:         \dvpADC_Declare name,outport,outpin,inport,inpin
'Parameters:  name    : name that will be used as label for the generated datatable
'             outport : port to which the outpin belongs (ppPortA,ppPortB,ppPortC,ppPortD,ppPortE)
'                       You may add $80 to set the pin latch register high.
'             outpin  : pin used to keep inpin near 50% Vdd threshold (ppPin0,ppPin1,ppPin2,ppPin3,ppPin4,ppPin5,
    ppPin6,ppPin7)
'             inport  : port to which the inpin belongs
'             inpin   : pin used to trigger the outpin
'Returns:     none
' ------------------------------------------------------------------------------------------------------------
dvpADC_Declare macro name,outport,outpin,inport,inpin
' ------------------------------------------------------------------------------------------------------------

' ============================================================================================================

'Description: Declare a Digital to Analogue VP
'Use:         \dvpDAC_Declare name,outport,outpin,value
'Parameters:  name    : name that will be used as label for the generated datatable
'             outport : port to which the outpin belongs (ppPortA,ppPortB,ppPortC,ppPortD,ppPortE)
'                       You may add $80 to set the pin latch register high.
'             outpin  : pin used for pwm output(ppPin0,ppPin1,ppPin2,ppPin3,ppPin4,ppPin5,ppPin6,ppPin7)
'             value   : initial value of DAC output (0-255)
'Returns:     none
' ------------------------------------------------------------------------------------------------------------
dvpDAC_Declare macro name,outport,outpin,value
' ------------------------------------------------------------------------------------------------------------

' ============================================================================================================

'Description: Declare a Pulse Width Modulation VP
'             PWM frequency = INTERRUPTRATE/(lowtime+hightime)
'Use:         \dvpPWM_Declare name,outport,outpin,lowtime,hightime
'Parameters:  name    : name that will be used as label for the generated datatable
'             outport : port to which the outpin belongs (ppPortA,ppPortB,ppPortC,ppPortD,ppPortE)
'                       You may add $80 to set the pin latch register high.
'             outpin  : pin used for pwm output(ppPin0,ppPin1,ppPin2,ppPin3,ppPin4,ppPin5,ppPin6,ppPin7)
'             lowtime : number of interrupt ticks output is low (>= 2)
'             hightime: number of interrupt ticks output is high (>= 2)
'Returns:     none
' ------------------------------------------------------------------------------------------------------------
dvpPWM_Declare macro name,outport,outpin,lowtime,hightime
' ------------------------------------------------------------------------------------------------------------

' ============================================================================================================

'Description: Declare a Timer VP
'             Each Timer VP supports eight 8bit timers (dvpT8_0 to dvpT8_7) and four 16bit timers
'             (dvpT16_0 to dvpT16_3). The 16bit timers overlap the 8bit timers.
```

```
'             The passed value sets the timer tick to <value/INTERRUPTRATE> seconds.
'             For 8bit timers the value is truncated to 8bits.
'Use:         \dvpTMR_Declare name,value
'Parameters:  name : name that will be used as label for the generated datatable
'             value: 16bit divider value (0-$FFFF)
'Returns:     none
' ----------------------------------------------------------------------------------------------------
dvpTMR_Declare macro name,value
' ----------------------------------------------------------------------------------------------------

' ====================================================================================================

'Description: Interrupt part that calls driver code.
'             It must be located in the same codepage as the driver code.
'Use:         \dvpIsr
'Parameters:  none
'Returns:     none
' ----------------------------------------------------------------------------------------------------
dvpIsr macro
' ----------------------------------------------------------------------------------------------------

' ====================================================================================================

'Description: Interrupt part of VP driver code.
'             It must be located in the lower half of a codepage.
'Use:         \dvpCore
'Parameters:  none
'Returns:     none
' ----------------------------------------------------------------------------------------------------
dvpCore macro
' ----------------------------------------------------------------------------------------------------

' ====================================================================================================

'****************************************************************************************************
' End of library DynamicVP_DEF.SXB
'****************************************************************************************************
```