Pico Mere Mortal Video Game Driver

The “Pico Mere Mortal Video Game Driver” is an interrupt driven video driver.

Sound is generated by the video driver.

The screen is made up of 16(h) by 11(v) tiles.

Each tile is made up of an 8x8 bitmap.

There are 16 tiles in a tileset. You can define 1 or 2 tilesets. If using two tilesets, the top line will use tileset 1, and the rest of the screen will use tileset 2. This is to allow for a top line that display a score value. Since the digits 0 to 9 will require 10 tiles.

Here is the driver memory map

For a single tileset (with TILESETS_1 defined}

```
$000 - $1FF = MMVGM Driver Code
$200 - $27F = Tile Set 1 (Bitmaps for tile set)
$280 - $7FE = Optional subroutines and Game program
```

For two tilesets (top line uses it's own tileset)

```
$000 - $1FF = MMVGM Driver Code
$200 - $27F = Tile Set 1 (Line 0 bitmaps for tile set)
$280 - $2fF = Tile Set 2 (Lines 1-10 bitmaps for tile set)
$300 - $7FE = Optional subroutines and Game program
```

Here are the I/O pins that are defined by the driver

```
LEDs            PIN RA OUTPUT

JStkUp          PIN RB.0 INPUT PULLUP  ' Joystick input pins
JStkDown        PIN RB.1 INPUT PULLUP  ' Joystick input pins
JStkLeft        PIN RB.2 INPUT PULLUP  ' Joystick input pins
JStkRight       PIN RB.3 INPUT PULLUP  ' Joystick input pins
JStkFire        PIN RB.4 INPUT PULLUP  ' Joystick input pins
IO_0            PIN RB.5 INPUT PULLUP
IO_1            PIN RB.6 INPUT PULLUP
IO_2            PIN RB.7 INPUT PULLUP

AVPort          PIN RC OUTPUT          ' Connects to Audio/Video R2R DACs
AVBlack         PIN RC.2 OUTPUT        ' Make high for black output
AVWhite         PIN RC.3 OUTPUT        ' Make high (along with black) for white output
```

Here are the constants that are defined by the driver

```
Pressed         CON 0                  ' Buttons go to zero when pressed

VidMode_VSync   CON 0                  ' Video modes: 0 = Generating Vertical Sync
VidMode_TBlank  CON 1                  '              1 = Generating Top Blank Lines
VidMode_Active  CON 2                  '              2 = Generating Active Video Lines
VidMode_BBlank  CON 3                  '              3 = Generating Bottom Blank Lines

Line0           CON 0                  ' Offset for video line 0
Line1           CON 16                 ' Offset for video line 1
Line2           CON 32                 ' Offset for video line 2
Line3           CON 48                 ' Offset for video line 3
Line4           CON 64                 ' Offset for video line 4
Line5           CON 80                 ' Offset for video line 5
Line6           CON 96                 ' Offset for video line 6
Line7           CON 112                ' Offset for video line 7
Line8           CON 128                ' Offset for video line 8
Line9           CON 144                ' Offset for video line 9
Line10          CON 160                ' Offset for video line 10
```

Here are the variables that are defined by the driver

```
videoLine0      VAR BYTE(8)            ' 1st line of text (line 0)
videoLine1      VAR BYTE(8)            ' 2st line of text (line 1)
videoLine2      VAR BYTE(8)            ' 3rd line of text (line 2)
videoLine3      VAR BYTE(8)            ' 4th line of text (line 3)
videoLine4      VAR BYTE(8)            ' 5th line of text (line 4)
videoLine5      VAR BYTE(8)            ' 6th line of text (line 5)
videoLine6      VAR BYTE(8)            ' 7th line of text (line 6)
videoLine7      VAR BYTE(8)            ' 8th line of text (line 7)
videoLine8      VAR BYTE(8)            ' 9th line of text (line 8)
videoLine9      VAR BYTE(8)            ' 10th line of text (line 9)
videoLine10     VAR BYTE(8)            ' 11th line of text (line 10)
```

Here are the subroutines that are included in the driver

```
PutTile       SUB 2, 3                 ' X, Y, TileID or Pos, TileID
GetTile       FUNC 2,1,2               ' X, Y or Pos; Returns TileID and RAM location
Snd           SUB 3                    ' Freq,Duration,Volume
Delay         SUB                      ' Delays for about "x" milliseconds
WaitSync      SUB 0                    ' Waits for the active video field to be done
```

Here are the optional subroutines that CAN be loaded

* Note that loading these subroutines will use a portion of the game code space.

```
'{$IFDEF LOAD_LINEADDR}
LineAddr      FUNC 1,1                 ' Given line# returns address
'{$ENDIF}

'{$IFDEF LOAD_CLEARSCREEN}
ClearScreen   SUB 1, 2                 ' TileID or TileID, Start Line
'{$ENDIF}

'{$IFDEF LOAD_SCROLLUP}
ScrollUp      SUB 0,1                  ' optional tile to clear new line
'{$ENDIF}

'{$IFDEF LOAD_SCROLLDOWN}
ScrollDown    SUB 0,1                  ' optional tile to clear new line
'{$ENDIF}

'{$IFDEF LOAD_SCROLLLEFT}
ScrollLeft    SUB 1                    ' tile to clear new line
'{$ENDIF}

'{$IFDEF LOAD_SCROLLRIGHT}
ScrollRight   SUB 1                    ' tile to clear new line
'{$ENDIF}

'{$IFDEF LOAD_VALUEINC}
ValueInc      SUB 1                    ' Position
'{$ENDIF}

'{$IFDEF LOAD_VALUEDEC}
ValueDec      SUB 1                    ' Position
'{$ENDIF}

'{$IFDEF LOAD_PLOTXY}
PlotXY            SUB 2                ' x,y (x=0 to 31, y=0 to 21)
'{$ENDIF}

'{$IFDEF LOAD_UNPLOTXY}
UnPlotXY          SUB 2                ' x,y (x=0 to 31, y=0 to 21)
'{$ENDIF}
```

There are a number of conditional defines that are used to configure the driver

```
'{$DEFINE TILESETS_1}
```

This will configure the driver for a single tileset used for all video lines. Using this define will allow more space for the game program.

```
'{$DEFINE TILESETS_2}
```

This will configure the driver to use two tilesets. Tileset 1 will be used for the top line, and tileset 2 will be used for the rest of the screen. This is the default mode.

```
'{$DEFINE LOAD_PIXEL_TILESET_1}
```

This will load the bitmaps for tileset 1 that work with the PlotXY and UnPlotXY commands.

```
'{$DEFINE LOAD_PIXEL_TILESET_2}
```

This will load the bitmaps for tileset 2 that work with the PlotXY and UnPlotXY commands.