```
{{ * SEE END OF FILE FOR TERMS OF USE
┌─────────────────────────────────────┬──────────────────┬───────┬─────────────┐
│ Name of Program                     │  Author          │ Ver   │ Date        │
├─────────────────────────────────────┼──────────────────┼───────┼─────────────┤
│ Spin_PASM_Style_Guide.spin          │  A. Hapgood      │ v1.00 │ 03-DEC-2013 │
├─────────────────────────────────────┴──────────────────┴───────┴─────────────┤
│ Provides the author with a template and style guide for Spin and PASM        │
│ programming.                                                                 │
│                                                                              │
│ Stack Space Required ( Longs ): Number of stack longs required by the object │
│                                                                              │
│ Use DD-MMM-YYYY date format.                                                 │
│                                                                              │
│ Use _v0.00 as a suffix for draft documents. Change suffix to v1.00 when      │
│ content is complete.                                                         │
│ First Release:  v1.00                                                        │
│ Minor Revision: v1.01, v1,02, etc.                                           │
│ Major Revision: v2.00, v3.00, etc.  Minor release reset to zero              │
│                                                                              │
│* See OBJ section for required objects                                        │
└──────────────────────────────────────────────────────────────────────────────┘

┌────────────────────────────────────────────────────────┬───────┬─────────────┐
│ Changes                                                │ Ver   │ Date        │
├────────────────────────────────────────────────────────┼───────┼─────────────┤
│ Brief description of changes                           │ v#.## │ DD-MMM-YYYY │
│ New stack space requirement if different               │       │             │
│ Author Name                                            │       │             │
└────────────────────────────────────────────────────────┴───────┴─────────────┘
12345678901234567890123456789012345678901234567890123456789012345678901234567890
└────────────────────────────── 80 columns wide ───────────────────────────────┘
}}
'├─ Two blank lines at the end of each block to provide whitespace
'│
CON
' ↑ Capitalize block designators

' │ Pre-defined constants are lowercase and
' │ preceeded by an underscore if one-time settable
' ↓                 ↓ keep equals signs lined up with colons in OBJ section
  _clkmode         = xtal1 + pll16x
  _xinfreq         = 5_000_000
' Predefined and one-time settable constants are bold faced automatically

' ↓ User defined constants; UPPER_CASE and no preceeding underscore
  CONSTANT1        = 1
  CONSTANT2        = 2
  CONSTANT3        = 3
  CLK_BURST_R      = $BF                'Clock - Burst Mode - Read

                       '│ delimit long constant values with
                       '↓ underscores to arrange into logical groups
  RAM_BURST_R      = %11_11111_1        'RAM - Burst Mode - Read ($FF)
  RAM_BURST_W      = %11_11111_0        'RAM - Burst Mode - Write ($FE)
```

```
VAR
'↓ Two spaces between left margin and code
  long  Variables                      'List purpose of variables
  word  Are                            'Short lines are commented like this and
  byte  CamelCase                      'may span multiple lines
                                      '↑ keep comments lined up
'Prefix variable names with a lowercase p when
'they are used as address pointers. pWolfNippleChips, etc.


OBJ
 '↓ Object names are CamelCase
  NameHere         : "objFileName"     'prefix with obj and CamelCase remainder
  CamelCase        : "objCamelCase"
  IsUsdForObj      : "objIsUsedForObjects"
                  '↑ keep colons lined up with equals signs in CON section


PUB Init : ResultVar | temp, i
''Use single line documentation comment to briefly describe function
{{PUB and PRI function names are CamelCase
  Result variable are CamelCase
  Keep local variable names short and all lower case
  Use Init for initialization actions, if needed
  Use documentation comments when describing usage information
}}

''RETURNS: description of what is returned here


  Code goes here


PUB Main
''Use main if this is a top object


  Code goes here

         '↓ Space before & after ↓
PUB Start( CogNum, pParam, pStack )'← parameters are CamelCase
''Use start and stop if this object launches a new cog and is not a top object


  Case CogNum
            'start specified cog
    0..7  : coginit
            'start next available cog
    Other : cognew
           '↑ spin functions are lowercase
```

```
PUB Stop
{{Use multi-line comments when there is more
  than one line of documentation or comments
}}


  Code goes here


PUB More
''Some is good, more is better and too much is just right


  More code happens here


PUB Examples
{{Use multi-line comments when describing theory}}


  NewFreq := _xinfreq                  'Short line comments should all line up
                                       'and can be more than one line

  'Long lines get comments before them
  NewFreq := _xinfreq * ( !PLLx #> || ( RCx == 0 ) ) + 12_000_000
  '                      ↑ spaces between brackets and variables

  'Unwieldy equations can utilize spacing to clarify logical groups
  NewFreq := _xinfreq * ( PLLx #> || ( RCx == 0 ) )  +  12_000_000
  Variable := ObjName.Call( constant( pi / 2 + 3 ) )
  ObjName.Str( string( CONSTANT1, "String here!" ) )


PRI Internal
''Use PRIs when needed, especially if this is not a top level object


  DoStuff := Here


CON
''Constants that will be used in assembly go here


DAT
''This section for spin only

  'Indirectly addressed or program modified constants go here
  LabelName        byte    "string", $AF, $ED
```

```
DAT
''Assembly code goes at the bottom
''Global labels are CamelCase
''Local labels are :lowercase

                  '↓ org statement lined up with instructions
                   org       0

                  '| instructions are lowercase
                  '|
                  '|         | destination operand
                  '|         |
                  '↓         ↓     ↓ one space between destination and source
Pasm               mov       pAddr, par              'get parameter pointer
                   mov       pVar1, pAddr
                   add       pAddr, #4               'prep for next long
                   mov       pVar2, pAddr  wz,wc
                                        '↑ ↑ ↑ comma but no space
                                        '| | | between effects fields
                                        '| |
                                        '| |effects fields are lowercase
                                        '|
                                        '| 2 spaces between source operand
                                        '| and effects field (lower case)

                   rdlong    Var1, pVar1  wz
  if_nz            rdlong    Var2, pVar2
 '↑ conditions start in column 3 ( leaves room for if_nc_and_nz )
 '| conditions are predefined so are lower_case

                   'increment Var1 and Var2 by 1 in an endless loop
:loop              call      #IncVars                'inc Var1 and Var2 by 1

                   wrlong    Var1, pVar1             'update variables
                   wrlong    Var2, pVar2             'in hub RAM

                   jmp       #:loop

IncVars            add       Var1, #1
                   add       Var2, #1
IncVars_ret        ret
```

```
'******************************************************************************
'INITIALIZED VARIABLES
'******************************************************************************
'Start PASM variable names on column 1; in line with PASM labels
'List pin variables first followed by other variables
'Initialized pin variables are UPPERCASE and should be short
'Other initialized variables are CamelCase
'Use 0-0 to indicate a variable is initialized elsewhere

CLK                long      0-0
CS                 long      |< 16                   'pin assignment

PwmCycle           long      16_000                  '5,000 cps @ 80 mHz
SampleHex          long      $F0F0F0F0               'hexadecimal

                   '32 bit binary assignment with delimiters
SampleBin          long      %0_00100_000_00000000_001110_000_001101

'******************************************************************************
'UNINITIALIZED VARIABLES
'******************************************************************************
'Uninitialized variables are CamelCase
                   'single line comments can go here
pAddr              res       1                       'or here
pVar1              res       1                       'pointer vars are prefixed
pVar2              res       1                       'with a lowercase p
Var1               res       1
Var2               res       1


                   fit       496
                  '↑         ↑ values line up with destination operand
                  '| fit directive, res and long lined up with instructions
```

```
{{
┌──────────────────────────────────────────────────────────────────────────────┐
│                            THEORY OF OPERATION                               │
├──────────────────────────────────────────────────────────────────────────────┤
│ Long descriptive information goes here, especially when describing theory.   │
│ This section should be in the last block.                                    │
│                                                                              │
│                                                                              │
│                                                                              │
│                                                                              │
│                                                                              │
└──────────────────────────────────────────────────────────────────────────────┘
┌─────────────────┬───────────────┐
│   Valid Clock   │ CLK Register  │   This chart can be explained
│      Modes      │    Value      │   with comments over here
├─────────────────┼───────────────┤
│ RCFAST          │ 0_0_0_00_000  │
├─────────────────┼───────────────┤
│ RCSLOW          │ 0_0_0_00_001  │
└─────────────────┴───────────────┘
┌─────────────────┬───────────────┐
│ Tab Stops       │ Value         │   Settings from
├─────────────────┼───────────────┤   Edit - Preferences - Operation
│ CON Block       │ 2,19,39,79    │
├─────────────────┼───────────────┤
│ VAR Block       │ 2,8,39,79     │
├─────────────────┼───────────────┤
│ OBJ Block       │ 2,19,39,79    │
├─────────────────┼───────────────┤
│ PUB/PRI Block   │ 2,39,79       │
├─────────────────┼───────────────┤
│ DAT Block       │ 2,19,29,53,79 │
└─────────────────┴───────────────┘
┌──────────────────────────────────────────────────────────────────────────────┐
│                          TERMS OF USE: MIT License                           │
├──────────────────────────────────────────────────────────────────────────────┤
│ Permission is hereby granted, free of charge, to any person obtaining a copy │
│ of this software and associated documentation files (the "Software"), to     │
│ deal in the Software without restriction, including without limitation the   │
│ rights to use, copy, modify, merge, publish, distribute, sublicense, and/or  │
│ sell copies of the Software, and to permit persons to whom the Software is   │
│ furnished to do so, subject to the following conditions:                     │
│                                                                              │
│ The above copyright notice and this permission notice shall be included in   │
│ all copies or substantial portions of the Software.                          │
│                                                                              │
│ THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR   │
│ IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,     │
│ FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL      │
│ THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER   │
│ LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING      │
│ FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS │
│ IN THE SOFTWARE.                                                             │
└──────────────────────────────────────────────────────────────────────────────┘
12345678901234567890123456789012345678901234567890123456789012345678901234567890
└────────────────────────────── 80 columns wide ───────────────────────────────┘
}}
```