ChipFiles

19 February 2018 06:28

This is the process for the production of MicroChip PIC chipfiles for Great Cow BASIC.

The outcome of this process is a microcontroller specific DAT file. These files are located in the chipfiles folder of the Great Cow BASIC installation.

A microcontroller specific DAT file is required for every microcontroller that Great Cow BASIC supports. The microcontroller specific DAT file contains information with respect registers, register bits, configuration options, memory, interrupts and microcontroller specific constraints (if any). This microcontroller specific DAT file is for Great Cow BASIC only - this is a good resource but the microcontroller specific DAT file should not be manually edited.

All issues with respect to the microcontroller specific DAT file should be report via the forum.

This document will explain how we produce these microcontroller specific DAT file.

Tools

1. MPLAB-X IDE - we will be using component of the MPASMX sub-directory. Great Cow BASIC uses *.INC files as the primary data source for microcontroller configuration.
2. PreProcess.BAT- the first part of the Great Cow BASIC conversion process. This uses GAWK.EXE - an Open Source application. PreProcessIncFile.bat is used for the first step of the conversion . The associated source script file is PreProcessIncFile.awk. PreProcessIncFile.awk is the script that is completes the conversion.
3. GETCHIPDATA.EXE - a Great Cow BASIC specific application for conversion.
4. Chipdata.cvs - a Great Cow BASIC specific data source. This file contains critical data with respect to microcontroller specific data.
5. CriticalChanges.txt - a Great Cow BASIC specific data source. This file contains critical date with respect to microcontroller specific data. This file contains corrections to known bugs in the INC source files and how to correct them.

Usage:

1. Update the source INC file from the latest MPLAN-X installation by copying the file from the source folder to the Incfiles \original folder .
   Copy C:\Program Files (x86)\Microchip\MPLABX\v4\mpasmx\p1*.inc [target]DAT\incfiles\OrgFiles
2. Execute PreProcess.bat. This will update or create the inc files in the [target]DAT\incfiles folder.
3. Update CriticalChanges.txt for any known corrections.
4. Update the Chipdata.cvs file with the correct chip data
5. Execute getchipdata.exe > chipfiles/outputlog.txt
6. Copy to chipfiles to distribution
7. Update SVN:DAT

Details of Chipdata.cvs

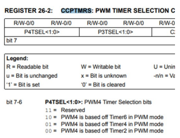
PWMTimerVariant can be set to 0, 1 or 2
Values will be set in the .dat using the following rule.

Search for CCPTMRS and examine.

PWMTimerVariant = 1. When CCPTMRS or CCPTMRSX has PTxSEL a value of 0x00 =Timer 2 and where PTxSEL=0x11 is reserved.

The Default (no value present). When PTxSEL as value=0x01 = Timer2 and PTxSEL=0x00 to be reserved, OR, this register /bit combination is not present. Which is the general case.

PIC16F1614
PWMTimerVariant = 1
Where timer2,4,6 = 0,1,2



REGISTER 26-2: CCPTMRS: PWM TIMER SELECTION C

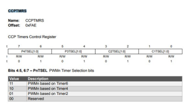
R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0
P4TSEL<1:0>		P3TSEL<1:0>		C
bit 7				

Legend:

R = Readable bit	W = Writable bit	U = Unin
u = Bit is unchanged	x = Bit is unknown	-n/n = Va
'1' = Bit is set	'0' = Bit is cleared	

bit 7-6 **P4TSEL<1:0>:** PWM4 Timer Selection bits
11 = Reserved
10 = PWM4 is based off Timer6 in PWM mode
01 = PWM4 is based off Timer4 in PWM mode
00 = PWM4 is based off Timer2 in PWM mode

PIC18(L)F27/47K40
PWMTimerVariant = 2
Where timer 2,4,6 = 1,2,3



CCPTMRS

Name: CCPTMRS
Offset: 0xFAE

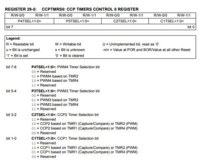
CCP Timers Control Register

t	7	6	5	4	3	2	1	0
	P4TSEL[1:0]		P3TSEL[1:0]		C2TSEL[1:0]		C1TSEL[1:0]	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
t	0	1	0	1	0	1	0	1

Bits 4:5, 6:7 – PnTSEL PWMn Timer Selection bits

Value	Description
11	PWMn based on Timer6
10	PWMn based on Timer4
01	PWMn based on Timer2
00	Reserved

PIC16(L)F19156
PWMTimerVariant = 2
Where timer2,4=1,2
Datasheet is wrong?



REGISTER 29-5: CCPTMRS0: CCP TIMERS CONTROL 0 REGISTER

R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1
P4TSEL<1:0>		P3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>	
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Reset
'1' = Bit is set	'0' = Bit is cleared	

bit 7-6 **P4TSEL<1:0>:** PWM4 Timer Selection bit
00 = Reserved
01 = PWM4 based on TMR2
10 = PWM4 based on TMR4
11 = Reserved

bit 5-4 **P3TSEL<1:0>:** PWM3 Timer Selection bit
00 = Reserved
01 = PWM3 based on TMR2
10 = PWM3 based on TMR4
11 = Reserved

bit 3-2 **C2TSEL<1:0>:** CCP2 Timer Selection bit
11 = Reserved
10 = CCP2 based on TMR1 (Capture/Compare) or TMR2 (PWM)
01 = CCP2 based on TMR1 (Capture/Compare) or TMR4 (PWM)
00 = Reserved

bit 1-0 **C1TSEL<1:0>:** CCP1 Timer Selection bit
11 = Reserved
10 = CCP1 based on TMR1 (Capture/Compare) or TMR2 (PWM)
01 = CCP1 based on TMR1 (Capture/Compare) or TMR4 (PWM)
00 = Reserved

Data File format

```
'GCBASIC/GCGB Chip Data File
'Chip: 16F819
'Main Format last revised:   14/07/2017
'Header Format last revised: 28/03/2019

[ChipData]
Prog=2048
EEPROM=256
RAM=256
I/O=16
ADC=5
MaxMHz=20
IntOsc=8, 4, 2, 1, 0.5, 0.25, 0.125
Pins=18
Family=14
ConfigWords=1
PSP=0
MaxAddress=511
'Microcontroller specific configuration to create variants. Parameters used with specific libraries, the compiler or user programs. All sourced from chip data database

'Used within  user programs
Stacks=8

'Used within user programs
UserIDAddress=8192
UserIDLength=4

[Interrupts]
ADCReady:ADIE,ADIF
CCP1:CCP1IE,CCP1IF
EEPROMReady:EEIE,EEIF
....

[Registers]
INDF,0
TMR0,1
PCL,2
STATUS,3
...
...

[Bits]
TMR1IF,PIR1,0
TMR2IF,PIR1,1
...
...

[FreeRAM]
20:7F
A0:EF
120:16F

[NoBankRAM]
70:7F

[Pins-DIP]
...

[ConfigOps]
OSC=LP,XT,HS,EC,EXTCLK,INTOSCIO,INTRC_IO,INTOSCCLK,INTRC_CLKOUT,EXTRCIO,EXTRC_IO,EXTRCCLK,EXTRC_CLKOUT
WDTE=OFF,ON

[Config]
FOSC_LP,1,16364
LP_OSC,1,16364
FOSC_XT,1,16365
```

SMTClockSourceVariant can be set to 1, 2 or 3.
Values will be set in the .dat the following rule applies for SMTxCLK

Search for SMTxCLK

SMTClockSourceVariant = 0.
No SMT Clock source.

SMTClockSourceVariant = 1.

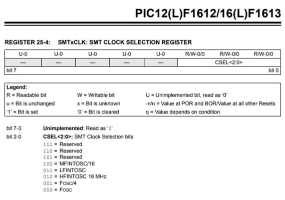
Microcontroller does not support MFINTOSC (500KHz)
SMT_MFINTOSC_16 = 4
SMT_LFINTOSC = 3
SMT_HFINTOSC = 2 '16MHz regardless of FOSC
SMT_FOSC4 = 1
SMT_FOSC = 0

SMTClockSourceVariant = 2.

SMT_AT1_perclk = 6
SMT_MFINTOSC = 5
SMT_MFINTOSC_16 = 4
SMT_LFINTOSC = 3
SMT_HFINTOSC = 2 '16MHz regardless of FOSC
SMT_FOSC4 = 1
SMT_FOSC = 0

SMTClockSourceVariant = 3.

SMT_REF_CLK = 7
SMT_SOSC = 6
SMT_MFINTOSC_16 = 5
SMT_MFINTOSC = 4
SMT_LFINTOSC = 3
SMT_HFINTOSC = 2 '16MHz regardless of FOSC
SMT_FOSC = 1
SMT_FOSC4 = 0



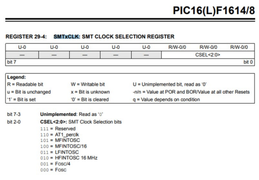
PIC12(L)F1612/16(L)F1613

REGISTER 25-4: SMTxCLK: SMT CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CSEL<2:0>		
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	q = Value depends on condition

bit 7-3 **Unimplemented**: Read as '0'
bit 2-0 **CSEL<2:0>:** SMT Clock Selection bits
111 = Reserved
110 = Reserved
101 = Reserved
100 = MFINTOSC/16
011 = LFINTOSC
010 = HFINTOSC 16 MHz
001 = Fosc/4
000 = Fosc



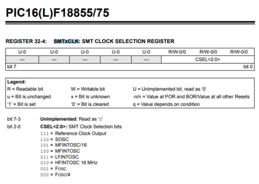
PIC16(L)F1614/8

REGISTER 29-4: SMTxCLK: SMT CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CSEL<2:0>		
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	q = Value depends on condition

bit 7-3 **Unimplemented**: Read as '0'
bit 2-0 **CSEL<2:0>:** SMT Clock Selection bits
111 = Reserved
110 = AT1_perclk
101 = MFINTOSC
100 = MFINTOSC/16
011 = LFINTOSC
010 = HFINTOSC 16 MHz
001 = Fosc/4
000 = Fosc



PIC16(L)F18855/75

REGISTER 32-4: SMTxCLK: SMT CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CSEL<2:0>		
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	q = Value depends on condition

bit 7-3 **Unimplemented**: Read as '0'
bit 2-0 **CSEL<2:0>:** SMT Clock Selection bits
111 = Reference Clock Output
110 = SOSC
101 = MFINTOSC/16
100 = MFINTOSC
011 = LFINTOSC
010 = HFINTOSC 16 MHz
001 = Fosc
000 = Fosc/4

ConfigBaseLoc to be set in the .dat the following rule applies:
Value are specific to the microcontroller and may need to be calculated.

Search for 'memory map' and examine if the CONFIG words are at 300000h

Applies to 18f only and it is calculated at the last location for the config memory address minus 15

An example, 18F67J50 the last location is 0x1FFFF, so the ConfigBaseLoc is 0x1FFFF - 15 = 0x1FFF0

Default location for an 18F is from 0x300000.

IntOSCCONFormat to be set in the .dat the following rule applies:
Values are 1. The default is empty.

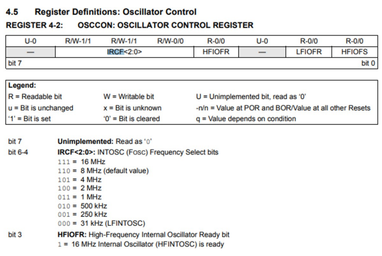
Search for HFIOFS bit

Applies to PIC only.
Examples are PIC10F322 and PIC18F13K22. This is required to ensure the clock oscillator is set correctly. This chip will have HFIOFS bit. The OSSCON bits for IRCFx are different for specific chips. The IntOSCCONFormat enables system.h to set the correct frequency bits.

IntOSCCONFormat = 1

111 = 111 16 Mhz
110 = 110 8 Mhz
. . .
000 = 0 31 kHz

OSCCON TABLE for IntOSCOONFormat =1



4.5 Register Definitions: Oscillator Control

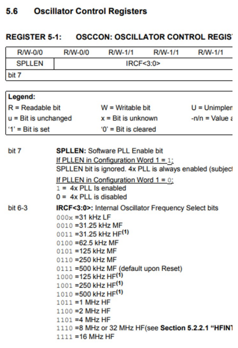
REGISTER 4-2: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R/W-1/1	R/W-1/1	R/W-0/0	R-0/0	U-0	R-0/0	R-0/0
—	IRCF<2:0>			HFIOFR	—	LFIOFR	HFIOFS
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	q = Value depends on condition

bit 7 **Unimplemented:** Read as '0'
bit 6-4 **IRCF<2:0>:** INTOSC (Fosc) Frequency Select bits
111 = 16 MHz
110 = 8 MHz (default value)
101 = 4 MHz
100 = 2 MHz
011 = 1 MHz
010 = 500 kHz
001 = 250 kHz
000 = 31 kHz (LFINTOSC)
bit 3 **HFIOFR:** High-Frequency Internal Oscillator Ready bit
1 = 16 MHz Internal Oscillator (HFINTOSC) is ready

Default OSCCON table - for comparison



5.6 Oscillator Control Registers

REGISTER 5-1: OSCCON: OSCILLATOR CONTROL REGIS

R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1
SPLLEN	IRCF<3:0>			
bit 7				

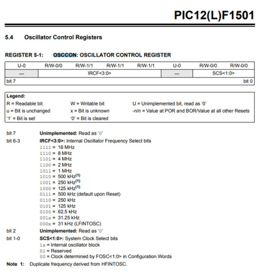
Legend:		
R = Readable bit	W = Writable bit	U = Unimpler
u = Bit is unchanged	x = Bit is unknown	-n/n = Value a
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **SPLLEN:** Software PLL Enable bit
If PLLEN in Configuration Word 1 = 1:
SPLLEN bit is ignored. 4x PLL is always enabled (subjec
If PLLEN in Configuration Word 1 = 0:
1 = 4x PLL Is enabled
0 = 4x PLL is disabled
bit 6-3 **IRCF<3:0>:** Internal Oscillator Frequency Select bits
000x =31 kHz LF
0010 =31.25 kHz MF
0011 =31.25 kHz HF(1)
0100 =62.5 kHz MF
0101 =125 kHz MF
0110 =250 kHz MF
0111 =500 kHz MF (default upon Reset)
1000 =125 kHz HF(1)
1001 =250 kHz HF(1)
1010 =500 kHz HF(1)
1011 =1 MHz HF
1100 =2 MHz HF
1101 =4 MHz HF
1110 =8 MHz or 32 MHz HF(see **Section 5.2.2.1 "HFINT**
1111 =16 MHz HF

IntOSCCONFormat = 2… there is **NO PLL** on these chips. 12f1501, 16f1503..9. Only 5 chips…
Similar to Default OSCCON but the Chipdata file would fail to set correct frequencies as PLL is not present

1111 = 16 Mhz
1110 = 110 8 Mhz

OSCCON TABLE for IntOSCOONFormat =2



PIC12(L)F1501

5.4 Oscillator Control Registers

REGISTER 5-1: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
—	IRCF<3:0>				—	SCS<1:0>	
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **Unimplemented:** Read as '0'
bit 6-3 **IRCF<3:0>:** Internal Oscillator Frequency Select bits
1111 = 16 MHz
1110 = 8 MHz
1101 = 4 MHz
1100 = 2 MHz
1011 = 1 MHz
1010 = 500 kHz(1)
1001 = 250 kHz(1)
1000 = 125 kHz(1)
0111 = 500 kHz (default upon Reset)
0110 = 250 kHz
0101 = 125 kHz
0100 = 62.5 kHz
001x = 31.25 kHz
000x = 31 kHz (LFINTOSC)
bit 2 **Unimplemented:** Read as '0'
bit 1-0 **SCS<1:0>:** System Clock Select bits
1x = Internal oscillator block
01 = Reserved
00 = Clock determined by FOSC<1:0> in Configuration Words

Note 1: Duplicate frequency derived from HFINTOSC.

ReadAD10BitForceVariant to be set in the .dat following rule applies:
Values are divisor that will shift the value READAD10, a word, by a number of bits.

Applies to improve support for 12bit ADCs force a return of 10bit values. Add ChipReadAD10BitForceVariant to the chip file where ChipReadAD10BitForceVariant is the divisor for the max value returned.

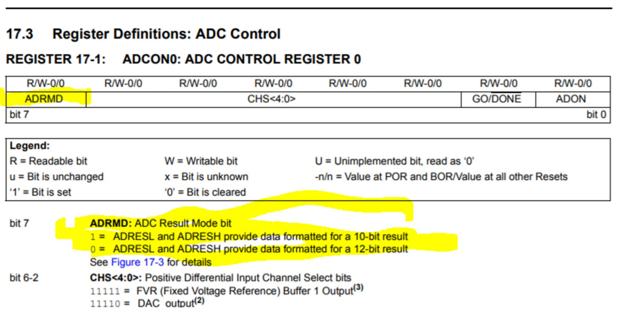
The is typically needed where the chip does not have the ADRMD bit as the . As the ADRMD bit will resolve the 10 bit issue, but, for those without ADRMD you need **ReadAD10BitForceVariant** :

```
    ;set AD Result Mode to 10-Bit
    #IFDEF Bit(ADRMD)      ;Added for 16F178x
       #IFDEF DebugADC_H
          NOP '#IFDEF Bit(ADRMD). set AD Result Mode to 10-Bit. @DebugADC_H
       #ENDIF

       SET ADRMD ON      ; WMR
    #ENDIF
```

Or

```
   #IFDEF ChipReadAD10BitForceVariant
       'Shift the data to 10bits when a 12bit ADC is returned and ADRMD is not valid
       IF ADN_PORT <> 0 then
          READAD10 = READAD10/ChipReadAD10BitForceVariant
       END IF
   #ENDIF
```



17.3 Register Definitions: ADC Control

REGISTER 17-1: ADCON0: ADC CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ADRMD	CHS<4:0>					GO/DONE	ADON
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **ADRMD:** ADC Result Mode bit
1 = ADRESL and ADRESH provide data formatted for a 10-bit result
0 = ADRESL and ADRESH provide data formatted for a 12-bit result
See Figure 17-3 for details

bit 6-2 **CHS<4:0>:** Positive Differential Input Channel Select bits
11111 = FVR (Fixed Voltage Reference) Buffer 1 Output(3)
11110 = DAC output(2)

Values of the **Family** value in the chip data files:

Family	Chips in family
12	PIC baseline chips - Most 10F, and 12F5* and 16F5* chips. (Those with 12 bit instructions)
14	PIC midrange chips - 10F3* chips, and any 12F and 16F chips other than F1* or F5* chips. (Those with 14 bit instructions)
15	PIC enhanced midrange chips - 12F1* and 16F1* chips. (Those with 14 bit instructions, but with the extra instructions and extra FSR register)
16	PIC high end chips - 18F (16 bit instruction width)
100	AVR - 90s1200, tiny11, tiny12, tiny15
110	AVR - tiny22, some 90s* parts.
120	Most AVR chips
121	Tiny4-5-9-10 and tiny102-104. Only 16 GPR's from r16-r31 and only 54 instructions.
130	AVR - mega32u6

The family is defined by GCB for the PIC chips, and is from the AVR Studio files for AVR. For each family, there is a core file specifying the instructions found on these chips and the binary equivalents. The core file is used by GCASM when assembling the program.

FamilyVariant is to handle chips that have added instructions and need different treatment in the assembler. It is used for 3 PIC families where some chips have added instructions not found in most. The default value is 0, but it can be set to anything else.

Family	FamilyVariant values
12	0: Default 1: Newer chips (16F527, 16F570) which have movlb, return and retfie instructions.
14	Not used
15	0: Default 1: Chips with more RAM that have an extra bit for the bank in the movlb instruction (6 bits rather than 5)
16	0: Default 1: Chips with more RAM (K42 and K83 series) that have a movffl instruction and 2 more bits for the address in the lfsr instruction Controls Automatic Context Save during interrupts for K42 and K83 with MVECEN = OFF, the EEPROM base address at 0x31000
AVR	Not used, although the HardwareMult parameter has a similar role in identifying which chips have a hardware multiplier.

The NoBankRAM section in the chip data file refers to the area of microprocessor RAM that can be accessed regardless of which bank is currently selected.

Example, the 16F59 does have RAM from 0x0A to 0x1F in bank 0, but only 0x0A to 0x0F can be accessed while another bank is selected. If for example bank 3 is selected, the PIC would be working with addresses between 0x60 and 0x7F. Accessing 0x6A to

0x6F would map back to 0x0A to 0x0F, but accessing 0x70 would access location 0x70 (not 0x10).