Note re use of 18F46K22

JG Dec 16 2015

1. CLOCK ISSUES

We have a 32.768KHz crystal as the secondary oscillator

Current plan is to CALIBRATE the HFINTOSC to enable its use for timing-sensitive operations – specifically the BAUD CLOCK (which requires 921.6K \* 2^n) and the timing of the stepper motor pulses for determination of RPM (less critical than BAUD CLOCK)

In order to accomplish these tasks (therby avoiding the use of a second crystal!) we use TMR5 and its gating function as follows:

To cal:

Set up TMR 6 to accept the 1MHz HFINTOSC / 4 with 1:16 prescaler (15625Hz nominal)  
Setup the T5G input to be PR6-match-TMR6 with PR6 = 224 [69.4 Hz (14.4msec) nominal]  
Ensure Timer5 gating function is in TOGGLE mode  
Use the 32KHz (no /4) as input to TMR5  
Count 32KHz pulses over the gating period (NOMINAL 32768\*1E-6\*64\*(PR6+1) = 471.8)  
Tune the oscillator until 32768\*(1/Fosc)\*64\*225 = 512 or Fosc = 921.6KHz

1. MEASURING GEN\_RPM

Given this calibration of the internal oscillator and use of the gating function of TMR5, we can easily determine the speed of rotation.

Use the signal from the external signal conditioner as the T5G input. The period of this conditioned signal is 1.2/RPM or, e.g., 600msec at slowest 2RPM and 2.4msec at fastest 500RPM.

Gate the HFINTOSC (calibrated) signal which is cal’d to be 921.6KHz nominal (in actuality +/- 0.5% or so)

Setup Timer5 to count Fosc/4 div 4 for a period ranging from 2.4 to 240msec.

Results will thus range from to 34560 (slowest) to 138 (fastest)

1. FCMEN (may not be needed if crystal is not used!)

Fail-safe clock enable functionality *could* be utilized to force a valve closure and shutdown.

Note that should the 1.8432MHz (or 3.6864MHz) crystal oscillator fail, there would be not ability to communicate with the mote. HOWEVER, we could force a mote reset, which would in turn, eventually, inform the controller that the sprinkler is down for the count.

This is a low priority functionality, however, we could, for now, enable the FCMEN fuse and keep the interrupt disabled pending final decisions as to how to treat the error condition.

1. PRICLKEN

To enable there to be ZERO power going into the primary oscillator, set this fuse AND manage the PRISD internal SFR bit. See p33 of datasheet. Fuse is PRIMARY\_SW.

1. PLLCFG

This config bit CLEAR allows PLL to be turned on under software control, rather than being on all the time. Make sure that OSCTUN:6 is never set. See p37 of datasheet.

1. Brown out

We believe that the brown out reset functionality can be disabled

1. MUX’d I/O lines

Note the following fuses:

“CCP2D2” should put PWM2B onto D2

“TIMER3B5” allows B5 to be TIMER3 input, but is not used. But cannot be at C0 which is dedicated to 2ndry oscillator

“CCP3E0” should put PWM3A onto E0..

“CCP2B3” should put PWM2A onto B3.

1. Stack Issues

There is both a NORMAL and FAST REGISTER stack. Never thought about this in previous versions. Is this relevant now?

Also the FUSE to cause a stack over/under reset should be set to avoid erratic behavior and upon reset, the STKPTR under/over bits should be checked to ensure that there is some handling of such an error.

1. CODE PROTECTION

Probably want to enable code protection FUSE in production, but not during prototype development

1. WATCHDOG

Decided to allow it to be enabled or disabled in software. Be sure to deal with WDT reset in code whenever functionality is enabled. (2 minutes time out, WDT\_32768, WDT\_SW fuses)

1. Compiler output re fuses (Dec16 2015)

Configuration Fuses:

Word 1: C100 XT NOPLLEN PRIMARY\_SW FCMEN IESO

Word 2: 3E18 PUT NOBROWNOUT BORV19 WDT\_SW WDT32768

Word 3: A000 CCP2B3 NOPBADEN CCP3E0 NOHFOFST TIMER3B5 CCP2D2 MCLR

Word 4: 0081 STVREN NOLVP NOXINST NODEBUG

Word 5: C00F NOPROTECT NOCPB NOCPD

Word 6: E00F NOWRT NOWRTC NOWRTB NOWRTD

Word 7: 400F NOEBTR NOEBTRB