DELAY GENRATION FOR PIC MICROCONTROLLER

PIC micro controllers have default internal 4Mhz clock which then automatically (internally) gets divided by four and becomes of 1Mhz so, each instruction cycle take 1uSec. Time to get executed, subroutines takes two instruction cycles.

To generate delay follow this procedure :

call and **goto** and any **subroutine** functions have **2uSec** delay **other functions** have **1uSec** delay. Generally any delay generation program/routine may contain two basic blocks,

MULTIPLICATION OF
BASIC DELAY SUBROUTINE
BY NECESSARY NUMBER OF TIMES
(This is used only when large delay is needed)

This block contains at least one **call** and one **goto** function. Each of 2uS delay and other functions.

MAIN DELAY ROUTINE

This block may contain one or more **goto** functions and other functions

To calculate delay you just have to multiply the number or numbers used for decrement in **decfsz** function with each other and with corresponding number of uSeconds (2 for 2uS which is for goto or call or any subroutine) taken by the functions.

Example Code :

Let's make a code for flashing an LED for five seconds. We will use delay routine of one second and then multiply it by five.

```
 List p=16f690
        #include <p16f690.inc>
        __CONFIG   _MCLRE_ON & _CP_OFF & _WDT_OFF & _INTRC_OSC_NOCLKOUT
        ERRORLEVEL - 302
;*********************CREATE REGISTER STORAGE BLOCKS*****************
        cblock 0x20
                temp
                d1
                d2
                d3
                multiply
        endc
;***********************SETUP START POINT******************************
        org 0x00
        goto Main
;***********************SETUP THE CONSTANTS****************************
        Main:
        bsf STATUS,RP0              ;switch to BANK 1
        movlw b'00000000'
        movwf TRISC                         ;set port c all output
        movwf TRISA                         ;set port a all output, for precaution
        movlw TRISB                 ;set port a all output, for precaution
        bcf STATUS,RP0              ;switch back to BANK 0
;************************MAIN LOOP*******************************
        Loop:
        movlw b'00000001'
        movwf PORTC
        call Multi              ;2 * start from Multi  subroutine
        movlw b'00000000'
        movwf PORTC
        call Multi
        goto Loop
        Multi:                  ; 5 x Delay (1 sec)
        movlw d'5'
        movwf multiply
        Loop1:
        call Delay              ;2      * Delay ( 2 * 124,000)
        decfsz multiply         ;1
        goto Loop1              ;2      *
        retlw 00h               ;1      * the 2 of call Multi =  gives TOTAL DELAY OF 992000uS =
                                                                 1 SECOND
```

```
Delay:
movlw d'62'   ; 62uS
movwf d1
movlw d'50'   ; 50uS
movwf d2
movlw d'5'    ; 5uS
movwf d3
Loop2:
decfsz d1     ;62    *
goto Loop2    ;2     *
decfsz d2     ;50    *
goto Loop2    ;2     *
decfsz d3     ;5     *
goto Loop2    ;2              =      124,000      and *
retlw 00h     ;1
end
```

Here, the total is archived by the following procedure :

No.	Function	Time Taken (in micro seconds)	Final Time
1	Loop calls Multi which takes 2uS	2	2 X
2	In Multi Loop1 calls Delay and Loop 1 also contains one goto	2(for call) x 2(for goto)	2 X 2 = 4
3	Delay after putting values in d1,d2,and d3 enters into Loop2 each decfsz takes corresponding time, for d1 62uS, for d2, 50uS, for d3 5uS. Between this decfsz functions there are goto functions each of them of 2uS.	Total time = 62 x2(for goto)x50x2(for goto)x5x2(for goto)x1(for retlw) = 124,000	124 X 100 X 10 X 1 = 124000
4	With retlw the code returns to Loop 1 of Multi and this happens for five times	Above time x 5	2 X 4 X 124000 = 992000 (aprox. 1 Second) X 5 = 5 second

NOTE:

- **Exact calculation of time can be done by little more accurate mathematical effort.**
- **For easy ready to use code for delay generation see this link.**
  http://www.piclist.com/techref/piclist/codegen/delay.htm

By: CHRIS