

delay

/*

lowlevel delay routines

Designed by Shane Tolmie of KeyGhost corporation. Freely distributable.

Questions and comments to shane@keyghost.com

Want to see 4Mb of Hi-Tech C FAQ and sample source code?

<http://www.keyghost.com/htpic>

(also has tiny device to record keystrokes in hardware on PC)

For Microchip 12C67x, 16C7x, 16F87x and Hi-Tech C

Example C:

```
#define PIC_CLK 8000000
```

```
#include "delay.h"
```

```
unsigned int timeout_int, timeout_char;
```

```
timeout_char=timeout_char_us(1147);
```

```
while(timeout_char-- && (RA1==0)); //wait up to 1147us for port RA1  
to go high
```

```
// - this is the max timeout
```

```
timeout_int=timeout_int_us(491512);
```

```
while(timeout_int-- && (RA1==0)); //wait up to 491512us for port  
RA1 to go high
```

```
// - this is the max timeout
```

```
dly250n; //delay 250ns
```

```
dly1u; //delay 1us
```

```
DelayUs(40); //do not do DelayUs(0) or else it bombs :)
```

```
DelayUs(255); //max
```

*/

```
#ifndef __DELAY_H
```

```
#define __DELAY_H
```

```
#define PIC_CLK 4000000
```

```
unsigned char delayus_variable;
```

```
#if (PIC_CLK == 4000000) || (PIC_CLK == 3686400) //3686400 is not  
entirely accurate, but it works
```

```
#define dly125n please remove; for 32Mhz+ only
```

ページ(1)

```

                delay
                #define dly250n please remove; for 16Mhz+ only
                #define dly500n please remove; for 8Mhz+ only
                #define dly1u asm("nop")
                #define dly2u dly1u;dly1u
#elif (PIC_CLK == 8000000)
                #define dly125n please remove; for 32Mhz+ only
                #define dly250n please remove; for 16Mhz+ only
                #define dly500n asm("nop")
                #define dly1u dly500n;dly500n
                #define dly2u dly1u;dly1u
#elif ( (PIC_CLK == 16000000) || (PIC_CLK == 16257000) )
                #define dly125n please remove; for 32Mhz+ only
                #define dly250n asm("nop")
                #define dly500n dly250n;dly250n
                #define dly1u dly500n;dly500n
                #define dly2u dly1u;dly1u
#elif (PIC_CLK == 20000000)
                #define dly200n asm("nop")
                #define dly400n dly250n;dly250n
                #define dly2u dly400n;dly400n;dly400n;dly400n;dly400n
#elif (PIC_CLK == 32000000)
                #define dly125n asm("nop")
                #define dly250n dly125n;dly125n
                #define dly500n dly250n;dly250n
                #define dly1u dly500n;dly500n
                #define dly2u dly1u;dly1u
#else
                #error please define pic_clk correctly
#endif

//****
//delay routine

#if PIC_CLK == 4000000 || (PIC_CLK == 3686400) //3686400 is not
entirely accurate, but it works
                #define DelayDivisor 4
                #define WaitFor1Us asm("nop")
                #define Jumpback asm("goto $ - 2")
#elif PIC_CLK == 8000000
                #define DelayDivisor 2
                #define WaitFor1Us asm("nop")
                #define Jumpback asm("goto $ - 2")
#elif ( (PIC_CLK == 16000000) || (PIC_CLK==16257000) )
                #define DelayDivisor 1
                #define WaitFor1Us asm("nop")
                #define Jumpback asm("goto $ - 2")
#elif PIC_CLK == 20000000
                #define DelayDivisor 1

```

```

                delay
        #define WaitFor1Us asm("nop")
        #define WaitFor1Us asm("nop")
        #define Jumpback asm("goto $ - 2")
#elif PIC_CLK == 3200000
        #define DelayDivisor 1
        #define WaitFor1Us asm("nop"); asm("nop"); asm("nop");
asm("nop"); asm("nop")
        #define Jumpback asm("goto $ - 6")
#else
        #error please define pic_clk correctly
#endif

```

```

#define DelayUs(x) { ¥
char) (x/DelayDivisor); ¥
                delayus_variable=(unsigned
                WaitFor1Us; } ¥
                asm("decfsz _delayus_variable, f"); ¥
                Jumpback;

```

/*

timeouts:

C code for testing with ints:

```

                unsigned int timeout;
                timeout=4000;
                PORT_DIRECTION=OUTPUT;
                while(1)
                {
                        PORT=1;
                        timeout=8000;
                        while(timeout-- >= 1); //60ms @
8Mhz, opt on, 72ms @ 8Mhz, opt off
                        PORT=0;
                }

```

Time taken:	optimisations on:	16cyc/number loop,
8us @ 8Mhz		
	optimisations off:	18cyc/number
loop, 9us @ 8Mhz		
	with extra check ie:	&& (RB7==1),
+3cyc/number loop, +1.5us @ 8Mhz		

C code for testing with chars:

similar to above

delay
 Time taken: optimisations on: 9cyc/number loop,
 4.5us @ 8Mhz
 with extra check ie: && (RB7==1),
 +3cyc/number loop, +1.5us @ 8Mhz

Formula: rough timeout value = (<us desired>/<cycles per
 loop>) * (PIC_CLK/4.0)

```

To use:            //for max timeout of 1147us @ 8Mhz
                   #define LOOP_CYCLES_CHAR            9
                   //how many cycles per loop, optimizations on
                   #define timeout_char_us(x)           (unsigned
char) ((x/LOOP_CYCLES_CHAR)*(PIC_CLK/4.0))
                   unsigned char timeout;
                   timeout=timeout_char_us(1147);
                   //max timeout allowed @ 8Mhz, 573us @ 16Mhz
                   while((timeout-- >= 1) && (<extra
condition>));     //wait
  
```

```

To use:            //for max 491512us, half sec timeout @ 8Mhz
                   #define LOOP_CYCLES_INT            16
                   //how many cycles per loop, optimizations on
                   #define timeout_int_us(x)           (unsigned
int) ((x+/LOOP_CYCLES_INT)*(PIC_CLK/4.0))
                   unsigned int timeout;
                   timeout=timeout_int_us(491512);
                   //max timeout allowed @ 8Mhz
                   while((timeout-- >= 1) && (<extra
condition>));     //wait
  
```

```

*/
#define LOOP_CYCLES_CHAR            9
                                  //how many cycles per loop, optimizations on
#define timeout_char_us(x)           (unsigned
char) ((x/LOOP_CYCLES_CHAR)*(PIC_CLK/4000000.0))

#define LOOP_CYCLES_INT            16
                                  //how many cycles per loop, optimizations on
#define timeout_int_us(x)           (unsigned
int) ((x/LOOP_CYCLES_INT)*(PIC_CLK/4000000.0))

#endif
  
```