SHDesigns Timer B Library for Dynamic C

2-24-2005 S. Henion

1.0 Description

The timberb.lib is used to generate a periodic interrupt. This interrupt is used to call a user-defined function at periodic intervals.

Timer B is a 10-bit free-running counter with several clock modes:

Clock Mode	*Max Delay @ 22mHz*	*Max Delay at 44MHz*
PCLK/2	92.48us	46.24us
PCLK/16	740.7us	370.4us
Clocked by Timer A1.	23.7ms	11.85ms

Timer A1 is also used by the serial ports if baud rates are under 2700 (22mHz). This limits the range if it is in use. Even using A1 as a prescaler, long delays can not be achieved.

The library adds an interrupt counter to increase the available range. This increases the maximum delay to about 1500 seconds.

2.0 Using the library.

The timerb.lib file needs to be added to the lib.dir Add the full path to the file if it is not in a directory underneath the compiler directory.

2.1 Defines

There are three #defines used by the library:

TIMER_DEBUG – if defined, the lib will output debug info on stdio.

TIMERB_IMMEDIATE – If this is defined, the user routine will be call almost immediately when the timer is started. Then it will be called later at the specified delay. This is useful for routines that are a start/stop period.

FAST_TIMERB – If this is defined, then only AF, HL and DE are saved in the interrupt. This makes the routine faster. The user function must save any additional registers it uses.

TIMER_INT_ENA – if defined, interrupts are re-enabled when calling the user function. This allows other interrupts to occur (serial interrupts.)

2.2 Functions

The main function is:

int TimerBInit(unsigned long usec,char intlevel)

where:

usec – is the period in microseconds between calls to the user function.
Intlevel – The interrupt level of the isr 1-3.

The function will return 0 if the time is supported. If the usec value is outside the range the timer supports.

float TimerBRate(void)

This will return the actual timer value in microseconds. Exact times are not possible due to limitations in the counter frequency. The value can be used to calculate the error from the specified time.

void TimerBUninit(void)

This stops timer interrupts.

2.3 User routine

If FAST_TIMERB is defined the routine would normally be an asm function as follows:

```
#asm root nodebug
TimerBRoutine::
;
; user assembly code
;
        ret
#endasm
```

A 'C' version can be used:

```
nodebug root void TimerBRoutine()
{
}
```

Either can be used if FAST_TIMERB is not defined. If FAST_TIMERB is defined, then no 'C' functions can be called unless the registers are saved first.

The “nodebug” option is important. Otherwise the ISR is much slower and the IDE does not debug interrupts well.
Adding a define:

```
#define         TIMER_INT_ENA
```

Will allow other interrupts to occur while your function is being called. Your function must be shorter than the time period.