Hardware Timer

This project uses a hardware timer (see Timer1 Operation) to flash LED1 on and off. For more details on how to control the LED see Parallel I/O (PIO) Ports.

Project files: timer3.zip.

timer3.c

```c
/* timer3.c
 *
 *  Use timer1 to blink an LED
*/

#include <plib.h>

void wait();

#define LED1 BIT_0  // LED1 is connected to RD0


int main()
{
        int i;
        mPORTDSetPinsDigitalOut(LED1);   /* Make LED1 output */
        mPORTDClearBits(LED1);      /* Turn off LED1 on startup */
        while (1) {
                mPORTDToggleBits(LED1);     /* Toggle LED1 */
                wait();
        }
        return 0;
}

void wait()
{
        const int DLY = 36000;
        T1CON = 0x30; // turn timer off and set prescaller to 1:256
        TMR1 = 0;
        PR1 = 0xFFFF;
        T1CONSET = 0x8000; // start timer
        while (TMR1 < DLY) {
                // just wait
        }
        T1CONCLR = 0x8000; // stop timer
}
```

Results

I timed my LED over ten cycles and got 18.37 seconds. According to my calculation that is equivalent to a PB clock rate of 10.03 MHz.

Exercise

> Run the project and repeat the measurement of the cycle time. Calculate the PB clock rate (show your work).

Maintained by John Loomis, updated *Sat Aug 02 16:35:37 2008*