

BlinkingLED


November 2010

Description

The purpose of this project is to blink LD2 at a frequency determined by a CPU timer. This LED is connected to the output of GPIO34.

Hardware Setup

1 ADC-A6 COMP3(+VE)	2 ADC-A2 COMP1 (+VE)	3 ADC-A0	4 3V3
5 ADC-A4 COMP2 (+VE)	6 ADC-B1	7 EPWM-4B GPIO-07	8 TZ1 GPIO-12
9 SCLA GPIO-33	10 ADC-B6 COMP3(-VE)	11 EPWM-4A GPIO-06	12 ADC-A1
13 SDAA GPIO-32	14 ADC-B0	15 EPWM-3B GPIO-05	16 5V0 (Disabled by Default)
17 EPWM-1A GPIO-00	18 ADC-B4 COMP2 (-VE)	19 EPWM-3A GPIO-04	20 SPISOMIA GPIO-17
21 EPWM-1B GPIO-01	22 ADC-A5	23 EPWM-2B GPIO-03	24 SPISIMOA GPIO-16
25 SPISTEA GPIO-19	26 ADC-B2 COMP1 (-VE)	27 EPWM-2A GPIO-02	28 GND
29 SPICLKA GPIO-18	30 GPIO-34 (LED)	31 PWM1A-DAC (Filtered)	32 GND

 No connection

Software Setup

Add the following variables to a watch window (View -> Watch)

CpuTimer0Regs.PRD.all (format = decimal) - This variable controls the period at which the LED will blink. By default this variable is set to mSec500 which is equivalent to 80000000. Its value should be set according to the equation:

$$ledBlinkPrd = (80M / 4)[low-speed\ clk\ frequency] * 2[\#\ of\ LED\ states] / <desired\ LED\ blink\ frequency\ (Hz)>$$

Overview

By default, the program uses CpuTimer0 and sets it to run at 500ms. The timer's period is set by the value written into the PRD bit of the CpuTimer0 register:

```
CpuTimer0Regs.PRD.all = mSec500;
```

Since there are two different LED states, the LED will blink once every second.

The program will then run in an infinite loop until the CPU timer interrupt flag is received. Once received, GPIO34's output value will toggle.

```
for(;;) //infinite loop
{
    // check to see if CpuTimer0 has reached its period value
    if(CpuTimer0Regs.TCR.bit.TIF == 1)
    {
        CpuTimer0Regs.TCR.bit.TIF = 1;        // clear flag

        //-----
        GpioDataRegs.GPBTOGGLE.bit.GPIO34 = 1;    // Toggle LD2
        //-----
    }
}
```