

```

// Sample LED binary counter using AXI Timer

// Timer registers
volatile int *TCSR0 = (int *)0x41C00000; // Control and status register
volatile int *TLR0 = (int *)0x41C00004; // Load register
volatile int *TCR0 = (int *)0x41C00008; // Counter register (unused here)

// Green LEDs Register
volatile int *ledData = (int *)0x40000000; // LED output register
volatile int *ledTri = (int *)0x40000004; // LED tristate register

int main(void)
{
    // Setup code (Timer)
    *TCSR0 = 0x0000; // Clear the timer configuration

    *TLR0 = 0x027BC85A; // Timer runs at 83.3MHz. Half of that is 4.16MHz
                        // Preloading the timer with 4.16 million and counting
                        // down to 0 means it will take exactly 0.5sec

    *TCSR0 = 0b00000110010; // Setting the following bits for functionality:
                            // Bit 1: Count down mode
                            // Bit 4: Auto-reload TLR0 -> TCR0 (upon reaching
                            // 0)
                            // Bit 5: Initially load TLR0 -> TCR0 (now)

    // Setup code (LEDs)
    *ledTri = 0x0000;
    *ledData = 0x0000;
    volatile int count = 0;

    // Enable timer:
    *TCSR0 = 0b00010010010; // Changing the following bits on TCSR0:
                            // Bit 5 off: Timer can't run when loading from
                            // TLR0
                            // Bit 7 on: Timer enable

    // Infinite loop
    while(1)
    {
        // Poll for timer complete flag
        if ((*TCSR0 & 0b00100000000) == 0b00100000000)
        {
            // Do time related operation
            if(count >= 0xF) { count = 0; }
            else { count++; }

            // Reset the timer flag by setting the flag bit
            *TCSR0 |= 1 << 8;
        }
    }
}

```

```
        // Misc code to run that isn't timer related
        *ledData = count;
    }

    return 0;
}
```