EMBEDDS

≡

Using watchdog timer in your projects

BY ADMIN   AVR TUTORIAL

All AVR microcontrollers have an internal watchdog timer that can be successfully used in your projects. Atmega328 and other modern AVR microcontrollers have the so-called Enhanced Watchdog Timer (WDT). It has few beneficial features, including a separate 128kHz clock source, reset the microcontroller, and generates an interrupt.
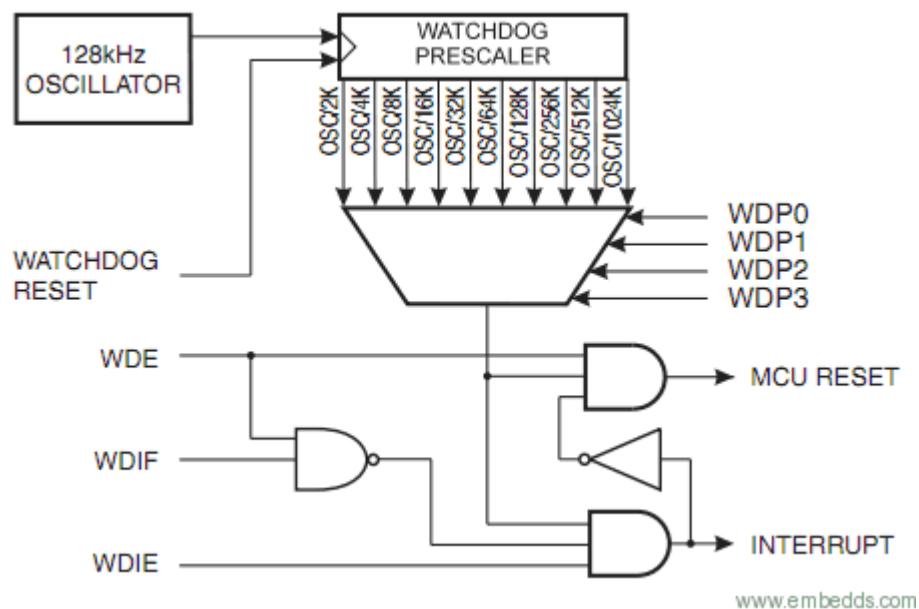
The watchdog timer is nothing more than a simple counter that gives a pulse when it counts up from the hardware perspective. This pulse can either generate an interrupt or reset MCU (or do both). The watchdog timer can be reset to zero at any time with simple **WDR** command, and this is where the fun begins. If you enabled the watchdog timer, you have to take care and reset it before it fills up and resets MCU. Otherwise, if your program hangs or sticks in some infinite loop

•

Watchdog clock and prescaller

The watchdog timer is clocked with a separate 128kHz oscillator. It is a low power oscillator that isn't accurate enough for precise timing operations. Upon individual MCU fuse settings, this clock also can serve as the system clock. Then watchdog becomes synchronous with the main clock. But this is a particular case we won't consider. For better flexibility, the watchdog timer has a clock prescaller system, similar to other timers. Depending on WDP[3..0] settings in the WDTCSR register, the watchdog can be set to timeout from 16ms up to 8s.

Watchdog Interrupt mode

The watchdog timer interrupt is a valuable source of an interrupt as it depends on a separate clock source. So it can be used to wake up MCU from sleep. You can construct a very efficient battery-driven system that needs to do some task every 8s and then get back to sleep. Or simply WDT interrupts can stop routines that take too long to execute. I bet you can see more uses here…

And there is a particular case where watchdog can generate both: interrupt and system reset. By using this feature, you may have the ability to preserve valuable parameters before the reset.

Programming AVR watchdog timer

There are two ways to set up a watchdog timer: programming WDTON fuse when flashing MCU, and another is during program flow. If the WDTON fuse is programmed, it cannot be disabled in software, and it only works in system reset mode (no interrupt available). This might be used to ensure system security. The only thing we can change here is a prescaller.

Software watchdog programming gives more flexibility and functionality. As we mentioned, you can use reset and interrupt modes. To start the watchdog timer or change its prescaller, some programming sequence is needed. This helps to ensure a secure program flow. These are:

1. Disable interrupts;

2. Reset watchdog timer with WDR command;

3. Write logic one to WDCE and WDE bits of WDTCSR simultaneously;

4. Then, in four cycles, write logic one to enable (or zero to disable) watchdog timer or change

manipulate watchdog timer. In this case, we can call any of the three functions that will take care of the proper execution sequence:

```
#define wdt_reset(); //resets watchdog timer
#define wdt_disable(); //disables watchdog timer
#define wdt_enable(timeout); //sets a watchdog prescaller
```

Depending on the selected microcontroller, prescallers may be set to one of the following values:

```
#define WDTO_15MS 0
#define WDTO_30MS 1
#define WDTO_60MS 2
#define WDTO_120MS 3
#define WDTO_250MS 4
#define WDTO_500MS 5
#define WDTO_1S 6
#define WDTO_2S 7
#define WDTO_4S 8
#define WDTO_8S 9
```

Let's write a simple program that will use a watchdog timer in interrupt and reset modes simultaneously. To do this, we simply run the program without resetting the watchdog timer. In the main routine, we will blink LED at approximately 0.5Hz, and when watchdog interrupts occurs, it will give a burst of five 0.1Hz signals. After reset, LED will light for a half-second. These will give us a clue about ongoing processes. Keep in mind that after watchdog interrupt counter has to count up once more for a reset to occur.

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/wdt.h>
#include <util/delay.h>
//Prepare LED pin
void PortInit(void)
```

```c
}
//initialize watchdog
void WDT_Init(void)
{
//disable interrupts
cli();
//reset watchdog
wdt_reset();
//set up WDT interrupt
WDTCSR = (1<<WDCE)|(1<<WDE);
//Start watchdog timer with 4s prescaller
WDTCSR = (1<<WDIE)|(1<<WDE)|(1<<WDP3);
//Enable global interrupts
sei();
}
//Watchdog timeout ISR
ISR(WDT_vect)
{
    //Burst of fice 0.1Hz pulses
    for (uint8_t i=0;i<4;i++)
    {
        //LED ON
        PORTD|=(1<<PD2);
        //~0.1s delay
        _delay_ms(20);
        //LED OFF
        PORTD&=~(1<<PD2);
        _delay_ms(80);
    }
}
int main (void)
{
    //Initialize port
    PortInit();
    //initialize watchdog
    WDT_Init();
    //delay to detet reset
    _delay_ms(500);
    while(1)
    {
        //LED ON
```

```
        _delay_ms(500);
    }
}
```

You can test this program with standard Arduino Duemilanove or Uno board by plugging LED between Digital 2 and GND pins:
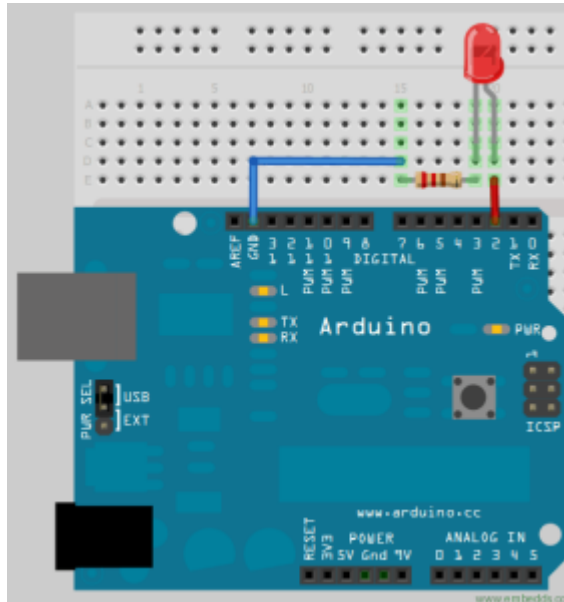
Or you can use any AVR development board with modern AVR.

As you can see, we enabled watchdog reset and interrupted modes almost without wdt.h library usage because it doesn't support interrupt mode programming yet. So we had to initialize by ourselves while following the rules. There is another watchdog library version developed by *Curt Van Maanen*. It can be downloaded here in case you decide to give it a try.

And finally, if you decide to debug your programs with a hardware debugger, be sure to disable the watchdog timer as it runs on its clock source while the debugged program runs much slower than normal. You don't want your application to be reset during any debugging step.

TAGGED   AVR Tutorial, change prescaller, watchdog timer, watchdog timer interrupt.

« Pong-like LED clock

Small tutorial on Interfacing capacitance
sensor to PIC »

Pingback: Electronics-Lab.com Blog » Blog Archive » Using watchdog timer in your projects

Dan
JULY 31, 2011 AT 12:45 AM

Thank you for this. I was experiencing a lot of frustration until I saw this post.

wesam
FEBRUARY 5, 2012 AT 5:53 PM

Hi,
I have set the WDT very short time, so when arduino start it will reset agan and could not access it via usb any more.....

what can i Do?????

thanks

Catprog
JULY 17, 2013 AT 12:00 PM

@wesam

use the following code

#include

void setup(){

wdt_reset(); // First thing, turn it off
wdt_disable();
Serial.begin(57600);

Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.
To find out more, including how to control cookies, see here: Privacy Policy

Close and accept

```
}
```

have your adunio unpluged and start the upload. just before it finishes compiling plug it in.

This is how I fixed that watchdog issue.

prasant

APRIL 14, 2017 AT 4:36 PM

in program above you have included wdt.h header file but you have written all function defination menually . By the way thanks for your tutorial .Really helpful to me 🙂

SEARCH

AVR Projects   Arduino   PIC Projects   Misc   Business   Technology   Other MCU Projects

Linux board projects   FPGA Projects   ARM Cortex   Internet   MSP430 Projects   Games

Software   Education   PIC32   ARM7 Projects   AVR Tutorial   MSC-51 Projects   PCB   Health

Handy Circuits   Guide   Lifestyle   68HC Projects   ARM Cortex Tutorial   BASIC Stamp   Finances

Casino   ChipKIT Projects   MSP430 Tutorial   Raspberry Pi   ZiLOG   Marketing

RECENT COMMENTS

Ed Woodrick on Is Investing in High-End Ethernet Cable Worth it for Programmers?

admin on Order PCBs from PCBWay and get a free mask to fight COVID-19

Ed on Order PCBs from PCBWay and get a free mask to fight COVID-19

John Jennings on Programming STM32-Discovery using GNU tools. Linker script

Ashok on Programming STM32-Discovery using GNU tools. Startup code

RECENT TUTORIALS

IoT based Temperature and Humidity Monitoring System using Raspberry pi Board

How to Teach Kids to Program with Raspberry Pi

Why You Should Buy a Raspberry Pi

Programming AVR I2C interface

Using Direct Memory Access (DMA) in STM32 projects

-

How to make pancakes healthier with Kratom?

The Top 5 Photo Editing Software Choices

CRM vs. ERP — Do You Know the Difference?

Five Tips To Be Successful In Data Science

TOP POSTS & PAGES

Programming AVR I2C interface

Blinking the LED with MSP430

Multichannel ADC using DMA on STM32

ADC on Atmega328. Part 1

Using Direct Memory Access (DMA) in STM32 projects

Copyright © Embedded projects from around the web|Privacy Policy

POWERED BY PARABOLA & WORDPRESS.