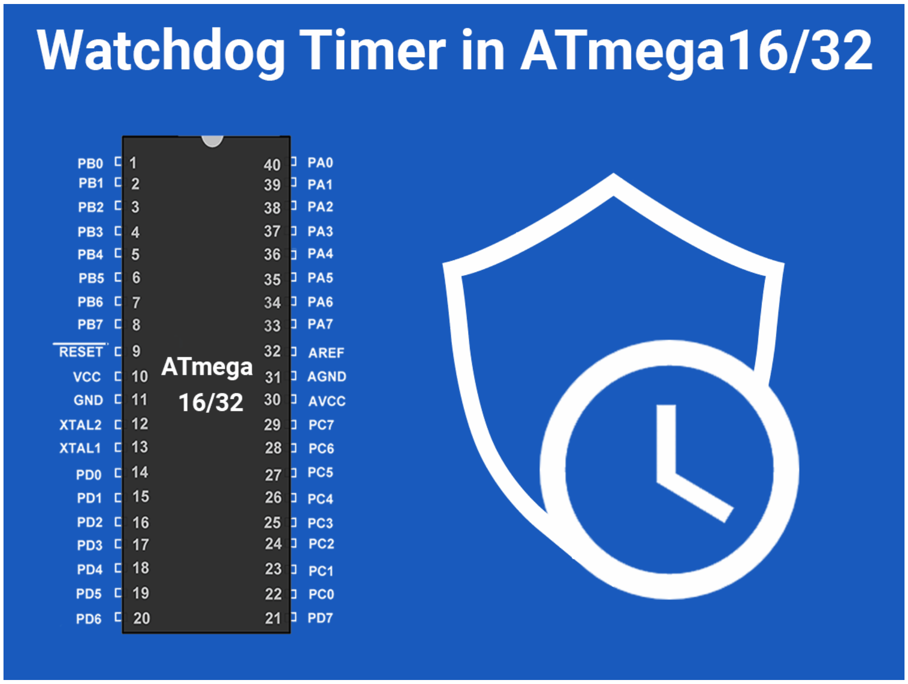
Watchdog Timer in AVR ATmega16/ATmega32



Introduction to Watchdog

- Watchdog Timer (WDT) can be helpful to automatically reset the system whenever a timeout occurs.
- A system reset is required for preventing the failure of the system in a situation of a hardware fault or program error.
- There are countless applications where the system cannot afford to get stuck at a point (not even for a small duration of time). For example, in a radar system, if the system hangs for 5 minutes, it can result in serious repercussions (an enemy plane or missile may go undetected resulting in huge losses).
- The system should be robust enough to automatically detect the failures quickly and reset itself in order to recover from the failures and function normally without errors.
- One can manually reset the system to recover from errors. But it is not always feasible to manually reset the system, especially once it has been deployed.
- To overcome such problems, a watchdog timer is necessary to automatically reset the system without human intervention.

How does Watchdog Timer work?

- The watchdog timer is loaded with a timeout period which is dependent on the application.
- The watchdog timer starts its counting independent of the system clock i.e. it has a separate internal oscillator to work independently of a system clock.

- The watchdog timer cleared through software each time before the timeout period occurs.
- Whenever software failed to clear the watchdog timer before its timeout period, then watchdog timer resets the system.
- For this purpose, the watchdog timer is used to overcome software failures in real-time applications.
- The watchdog timer is also used to wake up the microcontroller from sleep mode.

To enable, disable, and check the status of the watchdog timer following registers are used.

MCU Control and Status Register (MCUCSR):

7	6	5	4	3	2	1	0
JTD	**ISC2**	**-**	**JTRF**	**WDRF**	**BORF**	**EXTRF**	**PORF**

WDRF: Watchdog Reset Flag

- This bit is used to check the status of the watchdog timer.
- WDRF is set if a watchdog reset occurs.

Watchdog Timer Control Register (WDTCR):

7	6	5	4	3	2	1	0
-	**-**	**-**	**WDTOE**	**WDE**	**WDP2**	**WDP1**	**WDP0**

WDTOE: (Watchdog Turn-off Enable)

- This bit is used to disable the watchdog timer, this bit sets only when the WDE bit is written to logic 0. Otherwise, the watchdog timer is not disabled.

WDE: (Watchdog Enable)

- This bit is used to enable the watchdog timer. Watchdog timer enables when this bit is logic 1 and to disable set to logic 0.

1: Enable the watchdog timer

0: Disable the watchdog timer

To disable the watchdog timer –

1. First, write logic 1 to WDTOE and WDE.
2. Within the next four clock cycles, write a logic 0 to WDE. This disables the

Watchdog timer.

WDP2, WDP1 & WDP0

- WDP2, WDP1 & WDP0 bits represent the watchdog timer pre-scaling, the different pre-scaling timeout period is shown in the below table.

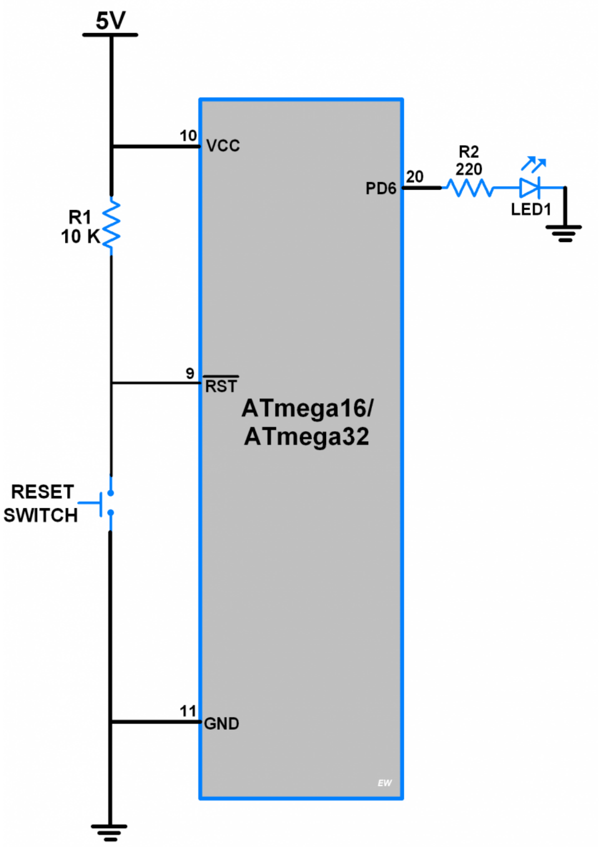
WDP2	WDP1	WDP0	Typical Time-out at VCC = 5.0V
0	0	0	16.3ms

WDP2	WDP1	WDP0	Typical Time-out at VCC = 5.0V
0	0	1	32.5ms
0	1	0	65ms
0	1	1	0.13s
1	0	0	0.26s
1	0	1	0.52s
1	1	0	1.0s
1	1	1	2.1s

Example

- Here, we are going to design a simple application that demonstrates the use of watchdog timer in ATmega16/ATmega32 based on AVR.
- In this application, the watchdog timer resets the main program after a 2.1-sec timeout. LED turns ON and OFF after every 2.1-sec due Watchdog timer.

Circuit diagram



ATmega16/32 Watchdog Timer Hardware Connections

ATmega16/32 Watchdog timer program

```
        WDTCR = (1<<WDE)|(1<<WDP2)|(1<<WDP1)|(1<<WDP0);
}

void WDT_OFF()
{
        /*
    This function use for disable the watchdog timer.
    */
        WDTCR = (1<<WDTOE)|(1<<WDE);
        WDTCR = 0x00;
}

int main(void)
{
        WDT_ON();                /* Enable the watchdog timer */
        LED_DDR |= 0xC0;
        LED_PORT |= (1<<6);      /* Set PD6 pin to logic high */
        _delay_ms(1000);         /* Wait for 1 second */
        LED_PORT &= ~(1<<6);     /* Clear PD6 pin */

    while(1);
}
```

Components Used

Powered By (https://www.mouser.in?utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

ATmega 16 ATmega 16	X 1	🛒 (https://www.mouser.in/ProductDetail/Microchip-Technology-Atmel/ATMEGA16L-8PU?qs=%2Fha2pyFaduiGCJtTvs2wv8fVZbVAalLu7Iq%2FgITS0tALAx6fMenLvg%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0) Datasheet (/components/atmega-16/1/datasheet)
Atmega32 Atmega32	X 1	🛒 (https://www.mouser.in/ProductDetail/Microchip-Technology-Atmel/ATMEGA32-16PU?qs=aqrrBurbvGdpkmgj7RWmsQ%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0) Datasheet (/components/atmega32/1/datasheet)

Components Used

LED 5mm

LED 5mm

X 1

(https://www.mouser.in/ProductDetail/Lite-On/LTL-307EE?qs=Yz4wJs0d%252BpgyXm%2FpkMp2pg%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Datasheet (/components/led-5mm/1/datasheet)

Downloads

ATmega16 watchdog timer project file

Download (/api/download/platform-attachment/315)

Comments

Comment

amrmagdi50 (/users/amrmagdi50/profile)
2019-10-01 02:05:57

Can you tell me please why do we delay 1 sec shouldn't we w8 yill the wdtc to do the 2.1 sec reset by itself

Reply Like 1

rogilelee (/users/rogilelee/profile)
2021-06-22 17:10:41

since watchdog timer will restart after 2.1 sec.

so once led is on then delay is 1 sec

so time left - 2.1sec - 1sec = 1.1 sec.

so led will be off for 1.1 sec then watchdog timer activated.

as watchdog timer is not effected by delay.

Reply Like 1

About Us (/about)
Business Offering (/business-services)
Host Platform (/launch-platform)
Contact Us (/contactus)

Connect On:

Facebook(https://www.facebook.com/electronicwings)

LinkedIn(https://www.linkedin.com/company/electronicwin

Youtube(https://www.youtube.com/channel/UCNdqkukBtk4

Instagram(https://www.instagram.com/electronicwings_co
igshid=1cip10jijttko)

Terms of Service (/terms-of-service)
Cookies Policy (/cookie-policy)
Privacy Policy (/privacy-policy)