Texas Instruments

Application Report
SPNA108–April 2007

Catalog TMS470 Watchdogs

Keith Engler *TMS470 Applications*

ABSTRACT

The TMS470 family of ARM7™ microcontrollers contains an analog watchdog (AWD). The AWD operation is described in the *TMS470R1 System Module Reference Guide* (SPNU189), and how to set up the AWD is described in the *Analog Watchdog Resistor, Capacitor, and Discharge Interval Selection Constraints* application report (SPNA089). The digital watchdog (DWD) is described in the *TMS470R1 Digital Watchdog Reference Guide* (SPNU244).

This application report is focused on the AWD and, therefore, it should be considered an addendum to the two AWD documents; specifically, highlighting how to determine if the TMS470 has an internal pulldown within the AWD. This document also references a spreadsheet to calculate the analog watchdog resistor and capacitor values based on the desired AWD period.

Contents

List of Figures

List of Tables

Disclaimer

This document is not intended to replace the TMS470R1B1M data manual nor is it in any way a device specification, and values in this document cannot be ensured. If any discrepancy is found in this document when comparing it to the TMS470R1B1M data manual, the TMS470R1B1M data manual should prevail.

1 Overview

This document refers specifically to the TMS470 family of catalog ARM7 microcontroller devices listed in Table 1. All of the devices in Table 1 have the analog watchdog (AWD), but they vary as to whether they require an external pulldown and whether they have the digital watchdog (DWD). This table also defines the upper and lower constant current limits of the internal pulldown, the AWD threshold, and the AWD source-to-drain resistance. These are important parameters when calculating the AWD period.

The TMS470 family of catalog ARM7 microcontroller devices has evaluation boards available for purchase. The AWD circuitry is designed into these boards with a jumper to disable the AWD. The R and C values are not connected for the convenience of creating a custom time period. The AWD period spreadsheet accompanying this document can be used to calculate the correct R and C values.

Table 1. TMS470 Catalog Devices

DEVICE	IPD	INTERNAL PULLDOWN (μA)		AWD THRESHOLD (V)		R_{DS} (Ω)	DWD
		MIN	MAX	MIN	MAX		
TMS470R1A64	Y	5	40	1.35	1.8	45	N
TMS470R1A128	Y	5	40	1.35	1.8	45	N
TMS470R1A256	Y	5	40	1.35	1.8	45	N
TMS470R1A288	N	N/A	N/A	1.35	1.8	45	Y
TMS470R1A384	N	N/A	N/A	1.35	1.8	45	N
TMS470R1A512	Y	5	40	1.35	1.8	45	N
TMS470R1A768	Y	5	40	1.35	1.8	45	N
TMS470R1AB1M	N	N/A	N/A	1.35	1.8	45	Y

Each device's data manual contains a Terminal Functions table that lists whether or not the AWD has an internal pulldown (IPD). Table 2 is an excerpt from the TMS470R1A128 data manual and shows that the AWD pin does have an IPD. Table 3 is also an excerpt from the TMS470R1A128 data manual and shows the constant current range associated with the pulldown. This information is in the data sheets for all devices listed in Table 1.

Table 2. AWD Pin Description

WATCHDOG/REAL-TIME INTERRUPT (WD/RTI)				
AWD	50	3.3-V I/O	**IPD (20 μA)**	Analog watchdog reset. The AWD pin provides a system reset if the WD KEY is not written in time by the system, providing an external RC network circuit is connected. If the user is not using AWD, TI recommends that AWD be connected to ground or pulled down to ground by an external resistor. For more details on the external RC network circuit, see the *TMS470R1x System Module Reference Guide* (literature number SPNU189) and the application note *Analog Watchdog Resistor, Capacitor and Discharge Interval Selection Constraints* (literature number SPNA005).

Table 3. Pulldown Current Range

I_I	Input current (I/O pins)	I_{IL} Pulldown	$V_I = V_{SS}$	-1	1	μA
		I_{IH} Pulldown	$V_I = V_{CCIO}$	5	40	
		I_{IL} Pullup	$V_I = V_{SS}$	-40	-5	
		I_{IH} Pullup	$V_I = V_{CCIO}$	-1	1	
		All other pins	No pullup or pulldown	-1	1	

2 Watchdogs

The DWD module monitors software operation and implements a system reset function upon CPU disruption. If the software enters into an improper loop or if the CPU becomes temporarily disrupted, the DWD timer underflows to assert a system reset. Once activated, the DWD can only be deactivated by a system reset.

The AWD generates a system reset when the voltage passes the threshold voltage (see Table 1). The watchdog reset is caused by a high voltage level on the AWD pin. Essentially, the external circuitry charges the AWD pin with a typical RC-charge curve. As the voltage crosses some threshold, the pin recognizes the external voltage as a logical 1 and creates a watchdog reset. The watchdog may be cleared by writing 0x0E5 and then 0x0A3 to the WKEY register. When the correct values are written, the AWD drains the external capacitor and resets the external RC delay. If an incorrect value is written to the WKEY register, a watchdog reset occurs immediately.

The DWD allows for a longer watchdog time period compared to the AWD with an internal pulldown. The periods of the DWD and the AWD with an external pulldown period can reach seconds, but the AWD with internal pulldown period is in the low millisecond range.

The internal pulldown acts to discharge the AWD pin, preventing it from getting too high. If the pulldown strength is too great and the external resistor is too large, the pin might never trip a watchdog reset. Therefore, a suitably low resistor must be chosen so that the watchdog can trip. If the AWD is not used, it can be grounded. The low resistance value forces either a large capacitor or a small watchdog timeout period. The problem with this watchdog approach is that the voltage is asymptotic, meaning that it spends a much time very close to the trip point and noise can possibly trip the watchdog.

3 Feeding the Watchdogs

The AWD may be cleared by writing 0x0E5 and then 0x0A3 to the WKEY register. When the correct values are written, the analog watchdog drains the external capacitor and resets the external RC delay.

The DWD may be cleared by writing 0xE51A and then 0xA35C to the DWKEY register. When the correct values are written, the DWD is reloaded and starts counting again.

The code in Figure 1 shows a method of feeding both the DWD and the AWD.

```
// DWD Module
__IO_REG32(DWKEY,   0xFFFFFF68,__READ_WRITE);
__IO_REG32(DWPRLD,  0xFFFFFF64,__READ_WRITE);
__IO_REG32(DWCTRL,  0xFFFFFF60,__READ_WRITE);

void Feed_Watchdogs()
{
  WKEY=0x0e5;   // AWD: WKEY is enabled for reset by the next 0x0A3
  WKEY=0x0a3;   //      Analog Watchdog is Reset
  DWKEY=0xe51a; // DWD: DWKEY is enabled for reloading by the next 0x0A35C
  DWKEY=0xa35c; //      Digital Watchdog is Reloaded
}
```

Figure 1. Function to Feed the DWD and AWD

The code in Figure 2 shows an example program setting up the DWD and toggling a GIO line. This program then continuously feeds to the DWD so that it does not cause the system reset. The GIO line goes back to zero upon a system reset.

```
void main(void)
{
  PCR = CLKDIV_2;                          // ICLK = SYSCLK / 2
  GCR = ZPLL_CLK_DIV_PRE_1;                // SYSCLK = 8 x fOSC
  PCR |= PENABLE;                          // Enable peripherals

  int count = 0;

  // You must first initialize the DWD preload register before the DWD will operate.
  // The DWD preload register is a 12-bit programmable register that
  // provides the expiration time for the DWD counter.
  // texp = 2^13 * (DWPRLD + 1) / OSCIN
  // where DWPRLD = 0,1,2,..,(212 - 1) and texp is the counter expiration time.
  DWPRLD = 99;  //Preload value for DWD, texp=0.109227
  DWCTRL = 0xACED5312; //Load complement of the default value (0x5312ACED)
                       //After the preload value has been loaded,
                       //enable DWD by writing a key to the DWD control register
                       //Once enabled; the DWD counter starts decrementing.
 for (;;)
  {
    volatile unsigned long i;
    count= count + 1;
    if(count == 50)
    {
      GIODCLRB = 0x00000001;                 // Reset GIOB0 output
      GIODIRB = 0x00000001;                  // GIOB0 to output direction
      GIODOUTB= 0x00000001;                  // GIOB0 High
    }
    for ( i = 0; i < 800000; i++ );
    if(count > 60)
    {
      Feed_Watchdog();
    }
  }
}
```

Figure 2. Main Code to Set Up DWD

4 How the Watchdog Reset Affects GIO Lines

The *Digital Watchdog Reference Guide* (SPNU244) shows that writing an incorrect value to the DWKEY causes a system reset.

This concept is demonstrated in the code example in Figure 3. By writing an incorrect value to the WKEY register, the system reset is invoked, and it forces the GIO line to go back to zero.

```
void main(void)
{
  PCR = CLKDIV_2;                        // ICLK = SYSCLK / 2
  GCR = ZPLL_CLK_DIV_PRE_1;              // SYSCLK = 8 x fOSC
  PCR |= PENABLE;                        // Enable peripherals

  int count = 0;

  WKEY = 0x0A3;  //No Action on AWD

  for (;;)
  {
    volatile unsigned long i;
    count= count + 1;
    if(count == 50)
    {
      GIODCLRB = 0x00000001;                 // Reset GIOB0 output
      GIODIRB = 0x00000001;                  // GIOB0 to output direction
//    GIODOUTB^= 0x00000001;                 // GIOB0 Toggle
      GIODOUTB= 0x00000001;                  // GIOB0 High
//    GIODOUTB= 0x00000000;                  // GIOB0 Low
      WKEY = 0x0E5;   //WKEY is enabled for reset by the next 0x0A3.
    }
    for ( i = 0; i < 800000; i++ );
    if(count == 60)
    {
      WKEY = 0x023;   //System reset; incorrect value written to WKEY
    }
  }
}
```

Figure 3. Code to Cause AWD to Issue a System Reset

5 Calculating the AWD Period

This document references an Excel® spreadsheet that is used to calculate the AWD R and C values based on the desired AWD period. The spreadsheet calculates the R and C values for devices with internal pulldowns and without internal pulldowns.

The goal is to prevent the AWD Vmax from intersecting the AWD threshold, which is described in the *Analog Watchdog Resistor, Capacitor, and Discharge Interval Selection Constraints* application report (SPNA089). This application report refers to Vf and V0, which corresponds to the AWD Vmax and AWD Vmin, respectively, in Figure 4.

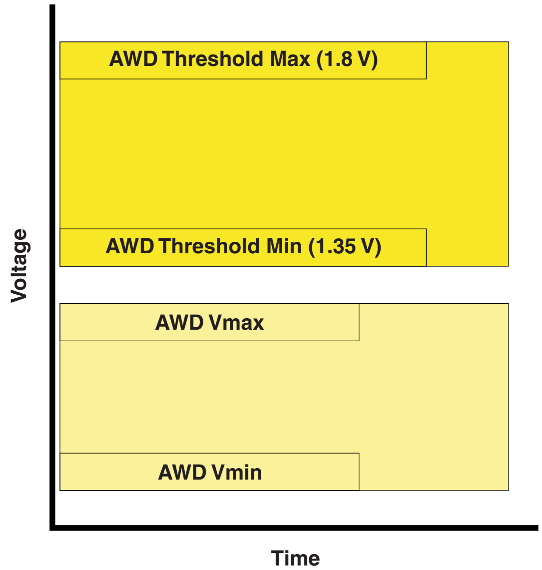

Figure 4. AWD Threshold

6 References

1. *TMS470R1x System Module Reference Guide* (SPNU189)

2. TMS470R1B1M KickStart™ Development Kit from IAR, http://focus.ti.com/docs/toolsw/folders/print/spnc010.html

3. *Analog Watchdog Resistor, Capacitor, and Discharge Interval Selection Constraints* (SPNA089)

4. *TMS470R1 Digital Watchdog Reference Guide* (SPNU244)

IMPORTANT NOTICE

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
Low Power Wireless	www.ti.com/lpw	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265