Published on *LabJack* (https://labjack.com)

23.0 Watchdog [T-Series Datasheet]

Overview

The Watchdog system can perform various actions if the T-Series device does not receive any communication within a specified timeout period.

A typical usage is to first use the IO Config system to set the power-up defaults for everything as desired, and then configure the watchdog to reset the device on timeout. For example, consider a software program that enables the watchdog to reset the T-Series device with a 60 second timeout, and then the software has a loop that talks to the device once per second. If something goes wrong with the software, or some other problem that causes communication to stop, the T-Series device will reset every 60 seconds until communication resumes.

The watchdog timeout can be set as low as 1 second, but such a low value is usually not a good idea. For example, when a USB device resets it takes a little time for USB to re-enumerate and software to be able to talk to the device again, so you could get in a situation where the device keeps resetting so often that you can't start talking to it again. This might require using the reset-to-factory jumper—see 11.0 SPC for details.

The timeout period is reset when a response to a command-response packet is sent to the host. Alternatively, "strict" mode can be enabled. When strict mode is enabled the timeout period is reset by writing the key value to WATCHDOG_STRICT_CLEAR. The key can be set by writing WATCHDOG_STRICT_KEY_DEFAULT. While strict is disabled writing any value to WATCHDOG_STRICT_CLEAR will clear the watchdog. Note that writing to WATCHDOG_STRICT_CLEAR will clear the watchdog when the write is processed, not when a response packet is sent.

Normal spontaneous stream data does not reset the watchdog timeout. The timeout period is reset when a response to a command-response packet is sent to the host, so if streaming you must do some periodic command-response communication to reset the watchdog timeout.

Register Listing

Use the following registers to configure the watchdog system:

Watchdog Registers

Name	Start Address	Type	Access
WATCHDOG_ENABLE_DEFAULT Write a 1 to enable the watchdog or a 0 to disable. The watchdog must be disabled before writing any of the other watchdog registers (except for WATCHDOG_STRICT_CLEAR).	61600	UINT32	R/W
WATCHDOG_ADVANCED_DEFAULT A single binary-encoded value where each bit is an advanced option. If bit 0 is set, IO_CONFIG_SET_CURRENT_TO_FACTORY will be done on timeout. If bit 1 is set, IO_CONFIG_SET_CURRENT_TO_DEFAULT will be done on timeout.	61602	UINT32	R/W
WATCHDOG_TIMEOUT_S_DEFAULT When the device receives any communication over USB/Ethernet/WiFi, the watchdog timer is cleared. If the watchdog timer is not cleared within the timeout period, the enabled actions will be done.	61604	UINT32	R/W
WATCHDOG_STARTUP_DELAY_S_DEFAULT This specifies the initial timeout period at device bootup. This is used until the first time the watchdog is cleared or timeout ... after that the normal timeout is used.	61606	UINT32	R/W
WATCHDOG_STRICT_ENABLE_DEFAULT Set to 1 to enable strict mode.	61610	UINT32	R/W
WATCHDOG_STRICT_KEY_DEFAULT When set to strict mode, this is the value that must be written to the clear register.	61612	UINT32	R/W
WATCHDOG_STRICT_CLEAR When running in strict mode, writing the key to this register is the only way to clear the watchdog. Writing to this register while not using strict mode will clear the watchdog.	61614	UINT32	W
WATCHDOG_RESET_ENABLE_DEFAULT Timeout action: Set to 1 to enable device-reset on watchdog timeout.	61620	UINT32	R/W
WATCHDOG_DIO_ENABLE_DEFAULT Timeout action: Set to 1 to enable DIO update on watchdog timeout.	61630	UINT32	R/W

Name	Start Address	Type	Access
WATCHDOG_DIO_STATE_DEFAULT The state high/low of the digital I/O after a Watchdog timeout. See DIO_STATE	61632	UINT32	R/W
WATCHDOG_DIO_DIRECTION_DEFAULT The direction input/output of the digital I/O after a Watchdog timeout. See DIO_DIRECTION	61634	UINT32	R/W
WATCHDOG_DIO_INHIBIT_DEFAULT The inhibit mask of the digital I/O after a Watchdog timeout. See DIO_INHIBIT	61636	UINT32	R/W
WATCHDOG_DAC0_ENABLE_DEFAULT Timeout action: Set to 1 to enable DAC0 update on watchdog timeout.	61640	UINT32	R/W
WATCHDOG_DAC0_DEFAULT The voltage of DAC0 after a Watchdog timeout.	61642	FLOAT32	R/W
WATCHDOG_DAC1_ENABLE_DEFAULT Timeout action: Set to 1 to enable DAC1 update on watchdog timeout.	61650	UINT32	R/W
WATCHDOG_DAC1_DEFAULT The voltage of DAC1 after a Watchdog timeout.	61652	FLOAT32	R/W

&print=true

Example

The most common way to use Watchdog is to write:

```
WATCHDOG_ENABLE_DEFAULT=0
WATCHDOG_TIMEOUT_S_DEFAULT=60
WATCHDOG_RESET_ENABLE_DEFAULT=1
WATCHDOG_ENABLE_DEFAULT=1
```

If the device does not receive any communication for 60 seconds, the watchdog will cause the device to reset. So if nothing is talking to the device, it will reset every 60 seconds. In conjunction, you would often use the IO Config system to configure the power-up defaults as desired.