A picture containing text, sign, clipart

Description automatically generated

DCA Testing

Hardware Clocks & Timers

2023-03-22

Table of Contents

1. Purpose……………………………………………………………………………………………………………………………… 2
2. Overview……………………………………………………………………………………………………………………………. 2
3. Scope…………………………………………………………………………………………………………………………………. x
   1. SYSTICK ……………………………………………………………………………………………….………… x
   2. RTC ……………………………………………………………………………………..………………..…..…. x
   3. TICK TIMER ………………………………………………………………………………………………..…. x
   4. SAMPLE TIMER……………………………………………………………………………………………... x
   5. WDT TIMER ………………………………………………………………………………………………….. x
4. Results………………………………………………………………………………………………………………………………. x
5. Analysis……………………………………………………………………………………………………………………………… x
6. **Purpose**

The internal core clock frequency used on the hardware module is 8Mhz, while a second high precision 32kHz clock is also used for timers. The 32bit timer (RTC) is running at 8Mhz, the 16bit Tick Timer used by the TCPIP stack is running at high precision 32kHz, the 32bit Sample timer used by the sensor interrupt for tick count is running at 8Mhz, the 16bit timer used by the WDT is running at high precision 32Khz.

1. **Overview**

The accuracy and drift of each timer is an important issue due to the time sensitivity of each task or event attached. RTC handles the timestamps used for data, events, and time synchronization. Tick Timer is used for the TCPIP send and receive. Tick Timer is also piggy backed for a 1sec callback event. The Sample timer is the fastested timer used. This is not used for a timestamp, but is used for delta time difference in ticks. This converts to nanosecond speeds. The WDT timer is not as important for accuracy, however the functionality of the WDT is important to prevent a customers hardware from going into a fail state indefinitely.

* Screenshots/pictures

1. **Scope**

The entirety of the API endpoints will be tested. Each endpoint for data, and user functionality will be tested through the browser/POST. All of these endpoints will be tested rigorously.

3.1 SYSTICK

The 24bit, SYSTICK timer running at 8MHz is not used in our application explicitly. Any background stacks using this are not doing any time sensitive work. This won’t be tested.

3.2 RTC

The RTC is a 32bit timer, running at 8Mhz/8=1Mhz. Each full count of the RTC equals 1second. An interrupt function is attached to this count timer, which increments the stored time() value inside the embedded system. This is used for time synchronization between the server and hardware.

Every test after this one will rely on the accuracy of this timer as it’s used to write to debug with a timestamp. Any event or data sent to server uses this value.

The tests for this timer will to ensure the Set\_Timestamp(), and GetTimestamp() are not returning errors, and the results are expected. These expected results are setting the timestamp with a known value (from server) and an immediate call of Get\_Timestamp() will reflect the value. This RTC value is read from the Count register.

3.3 TICK TIMER

The tick timer is running at 32kHz high precision. This is used explicitly by the TCPIP stack async functionality. This timer is also piggy backed at a 1sec interrupt to perform data calculation, status led code presentation, and timer pumps.

The test for this relies on the previous timer (RTC)’s accuracy. The test for this will be to debug store in memory a timestamp every interrupt. This test will occur over 1 hour periods as a heartbeat re-sync event in the case of no data recorded is called every hour. This will test the drift over an hour period as this is the longest it will possibly go without a resync.

3.4 SAMPLE TIMER

This timer is used to measure the delta difference between each sensor interrupt to calculate average. While this timer is used to read TICKS, the timer is ran at 8Mhz with no prescaler. The fastest speed this timer can run is 125ns. While this timer is 32bit, it will be reset every second to ensure the average over a 1sec period never has a register count roll over causing inaccuracy. The test for this will be to debug store the current ticks at the 1sec interrupt, reset the count register, and read it back ensuring it’s zero. This timer is so quick, it will be difficult to internally test it without an external clock source.

3.5 WDT

This timer is used to prevent a customer’s hardware from going into an unexpected state, indefinitely. Every iteration of the main loop, the watchdog is fed. This timer runs using the 32kHz high precision timer, and is prescaled to run at 1sec intervals. The WDT peripheral is initialized to reset the hardware after 8 seconds without being fed.

The test for this is to take the timestamp before the watchdog timer is started, commenting out the watchdog feed, and comparing the reset timestamp. This difference should be approx (8\*3600)ms. The accuracy of this isn’t highly important as it’s reset many times a second, however the deviation shouldn’t be more than +-3600ms

1. **Results**

|  |  |  |  |
| --- | --- | --- | --- |
| Number of Test Cases Planned | Number of Test Cases Executed | Number of Tests Cases Passed | Number of Test Cases Failed |
| 12 | 12 | 12 | 0 |

Tests were carried out for each scope one at a time, three times each, for a total of 18 test cases.

1. **Analysis**

All test cases were passed without issue. Every timer ran as expected with little to no drift when sync’d with server. The drift on the on the RTC with 1 hour between re-syncs was less than 1000ms (1second). This accuracy is enough to ensure data sent to the server is on the correct x-bound timestamps.