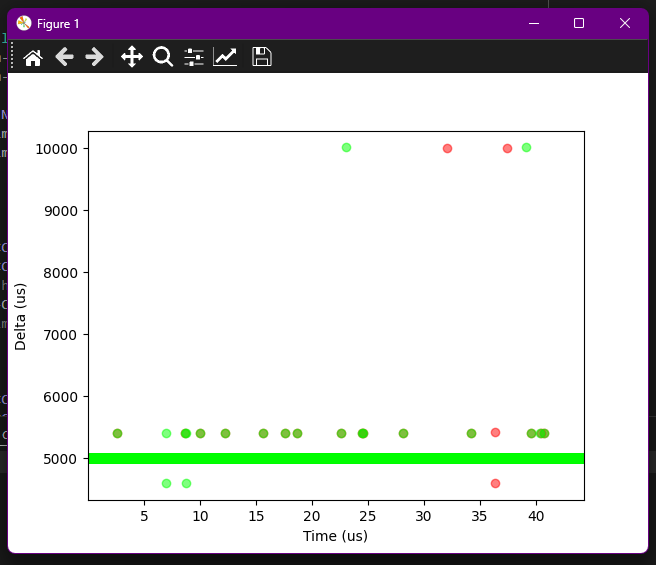
MIDDS Notes

# Fitting the coarse counter

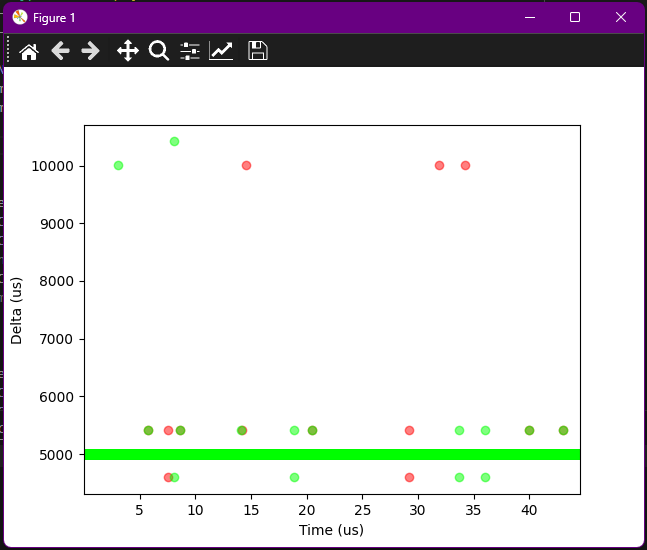


Input frequency of 100 Hz, detecting both rising and falling edges.

Using margin of < 40000.

Intermediate readings of SR register are **disabled**.

Error in X-axis label, should be seconds.

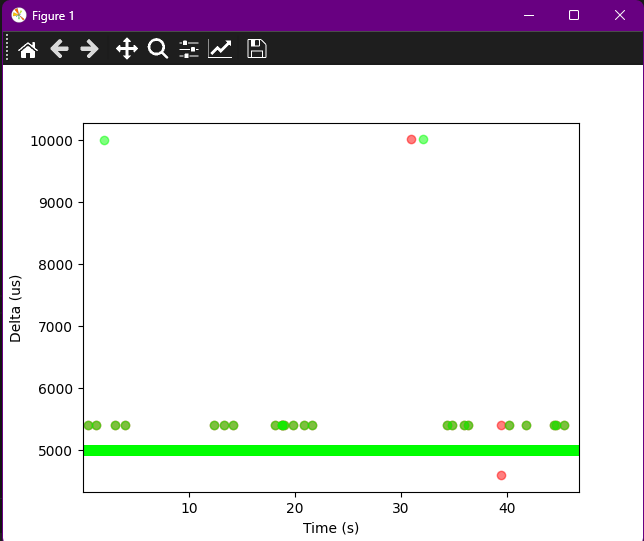


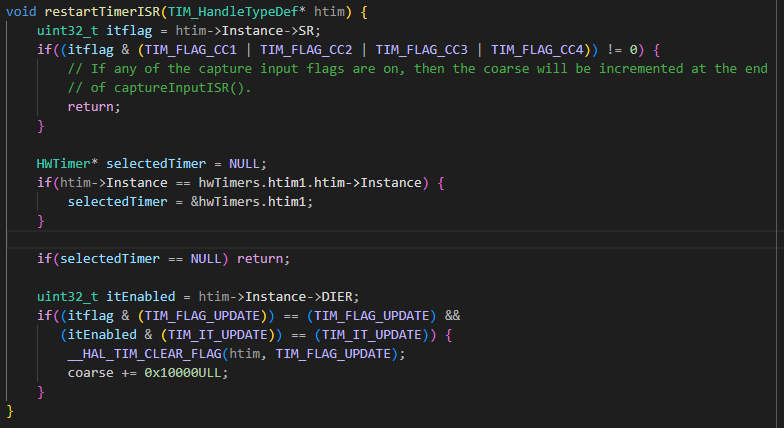
Input frequency of 100 Hz, detecting both rising and falling edges.

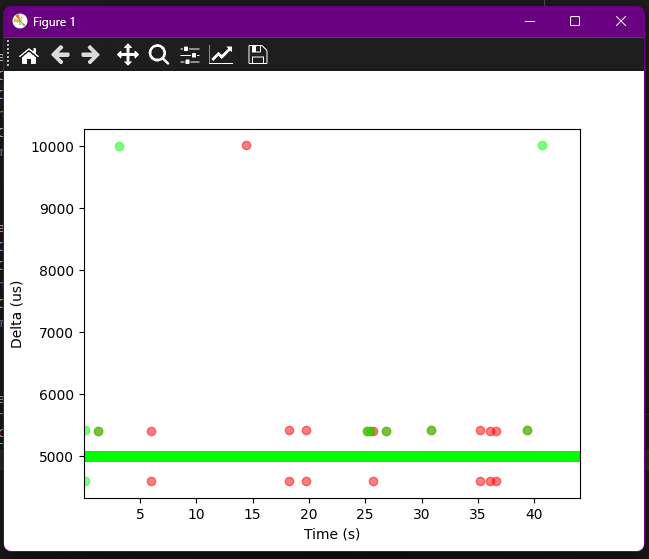
Using margin of < 40000.

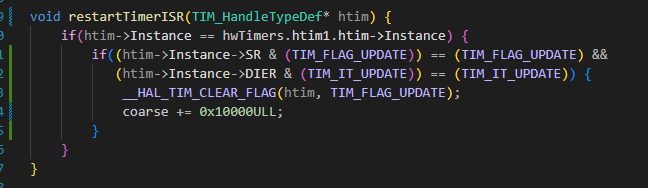
Intermediate readings of SR register are **enabled**.

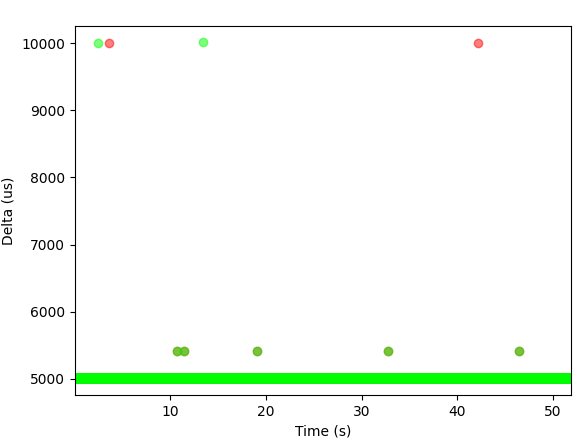
Error in X-axis label, should be seconds.

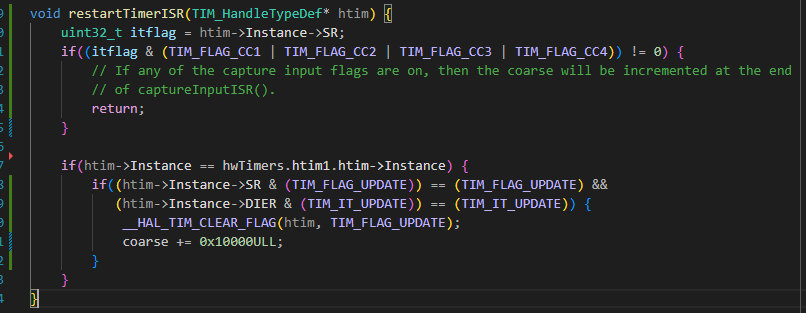


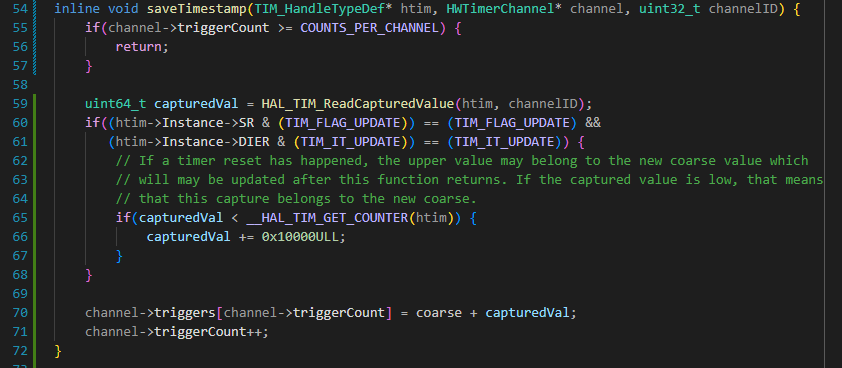




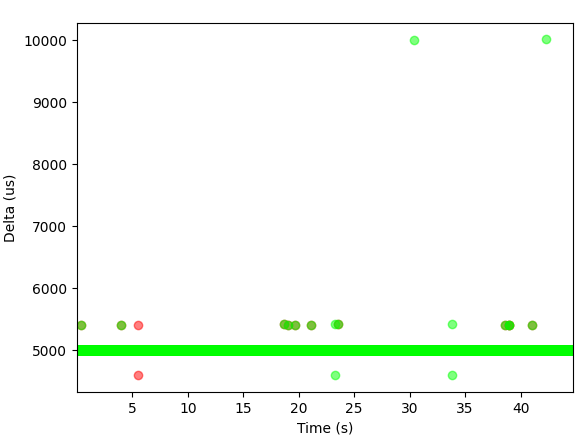
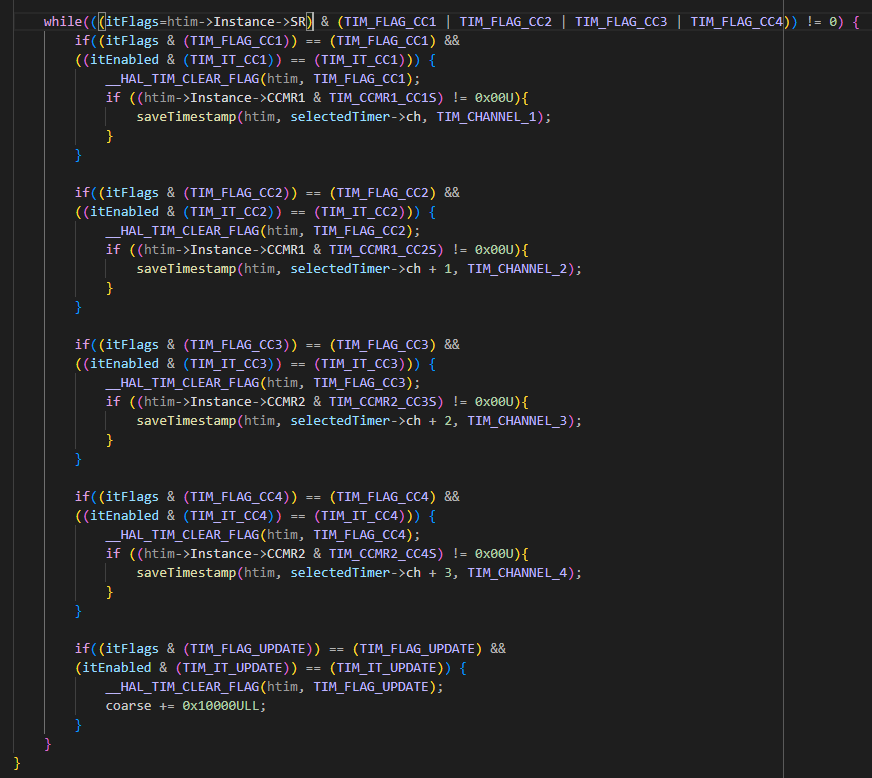






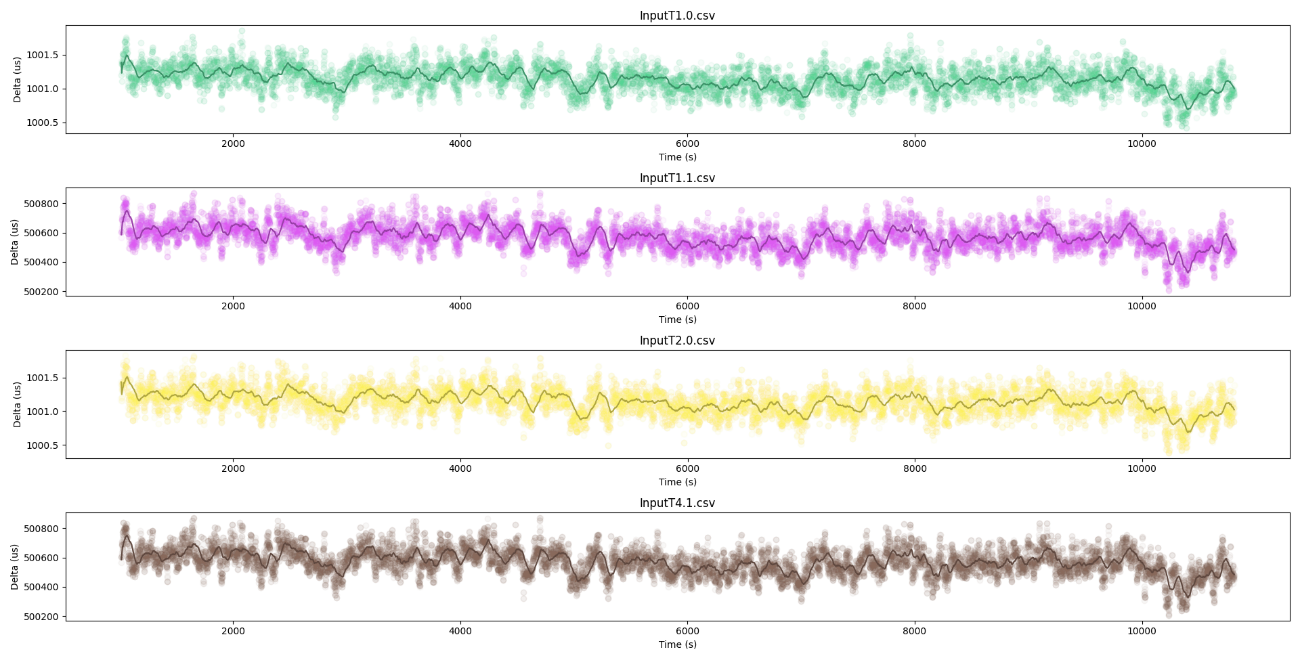


I thought that by checking if the captured value was less than the trigger count one could definitely know if the trigger happened AFTER the timer rollout. Looking good!

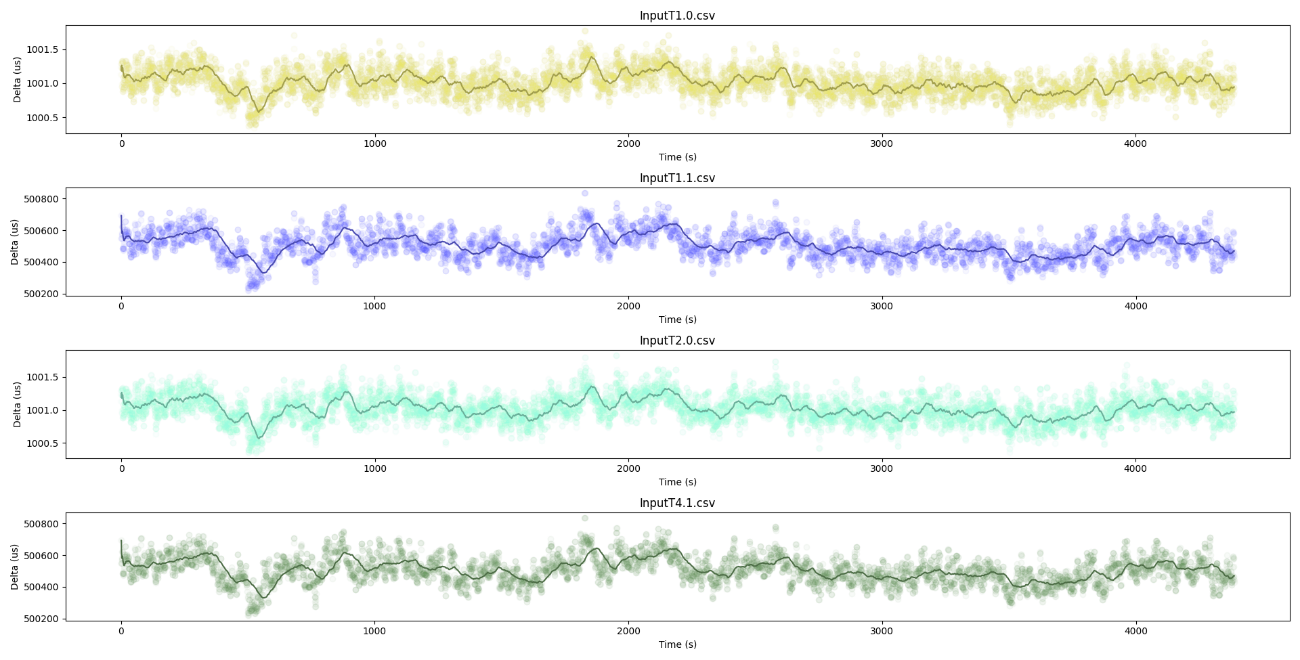
 

Inside captureInputISR() I thought that maybe by thoroughly making sure that no flag is set after exiting the ISR it would result in no overruns, but it doesn’t look to be making it and is instead ruining a little bit the measurements.

As everything tends to do, it worked out and I finally got some noiseless reliable measurements. The following are two different takes of two signals, 1Hz (T1.1 and T4.1) and 500Hz (T1.0 and T2.0), and its measured deltas (time interval between measurements). The device is both measuring rising and falling edges, that is why the delta values are double the real period of the input signal. As it can be seen, there aren’t any sudden jumps like in previous graphs.



Another take.



All these measurements are being done without the SYNC signal. There happens to be a relative deviation happening at the same time between channels. This could either be caused by the clock of the MCU or the devious quality of my signal generator…

Supposedly the signal generator has an accuracy or ±20ppm.