Application Note

AN000605

AS7341 EVAL KIT Flicker Detection

Introduction Light Flicker Detection

v1-00 • 2021-Feb-01

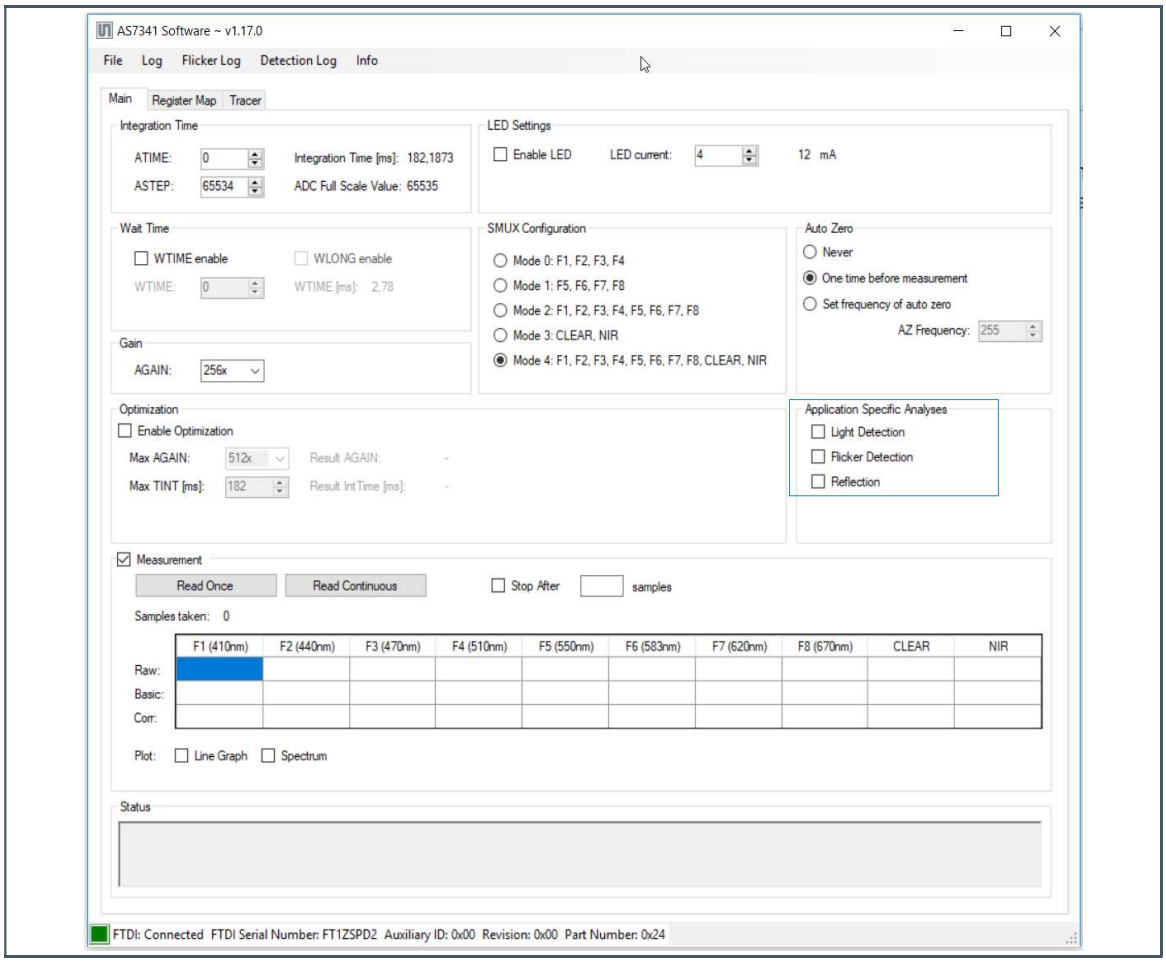
Content Guide

1 General

This document describes the Flicker function in the AS7341 Eval Kit. See the AS7341 Eval Kit manual for other hardware and software details, and/or find the basics for the 11-channel spectral sensor AS7341 in the datasheet.

Connect the AS7341 Eval Kit to a PC and start the GUI “AS7341 Demo”. A window will open, where you will find the function “Flicker Detection” located on the right side of the GUI, in the middle of the window within the block “Application Specific Analyzes”. Select this function to detect flicker frequencies based on hardware and software procedures.

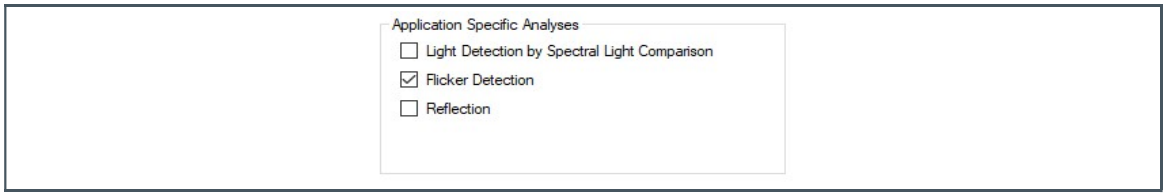
Figure 1 :
Main Window of AS7341 GUI with Flicker Function as Analyzes Function



2 AS7341 Demo Software Flicker Functions

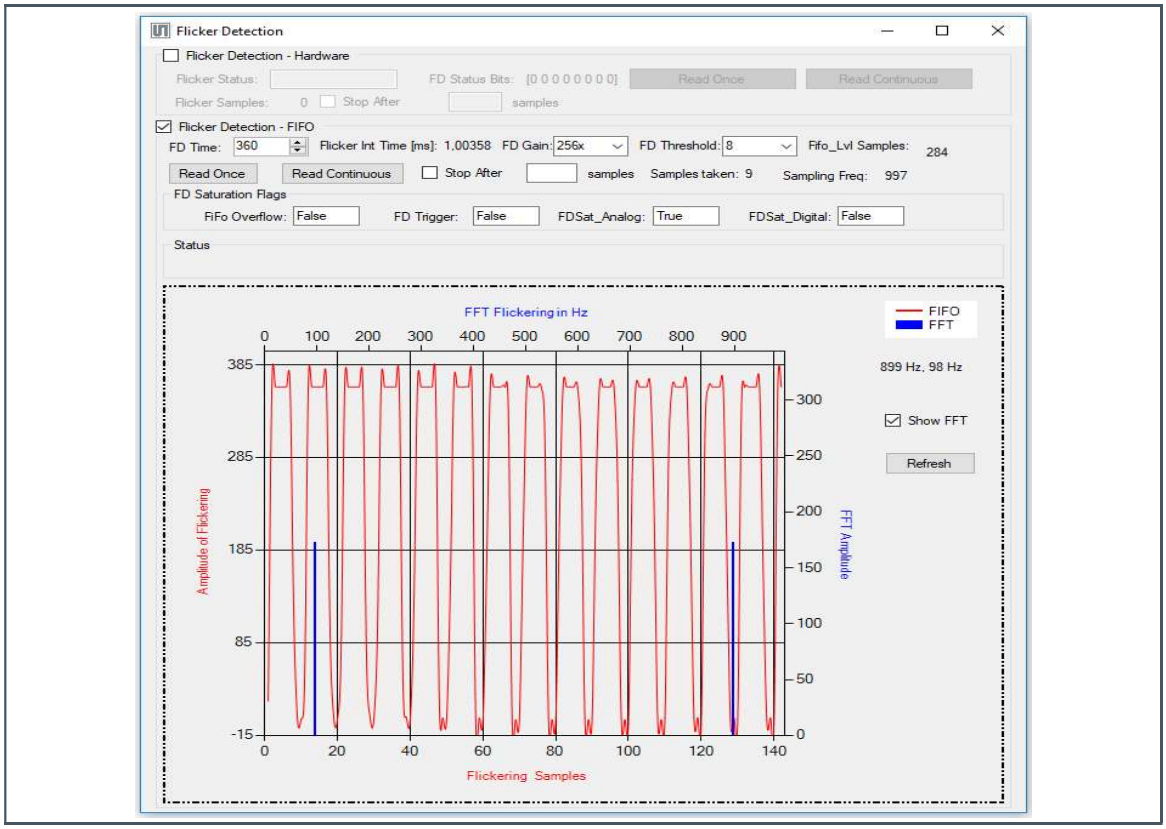
A window titled “Flicker Detection” pops up when clicking the Flicker Detection checkbox in the Application Specific Analyses section of the main window.

Figure 2:
Flicker Detection in Application Specific Analyses Group Box of Main Window



This disables the ALS measurements in the main window. In the GUI, two types of flicker detection methods are implemented – Hardware (on-chip Goertzel) and the FIFO method.

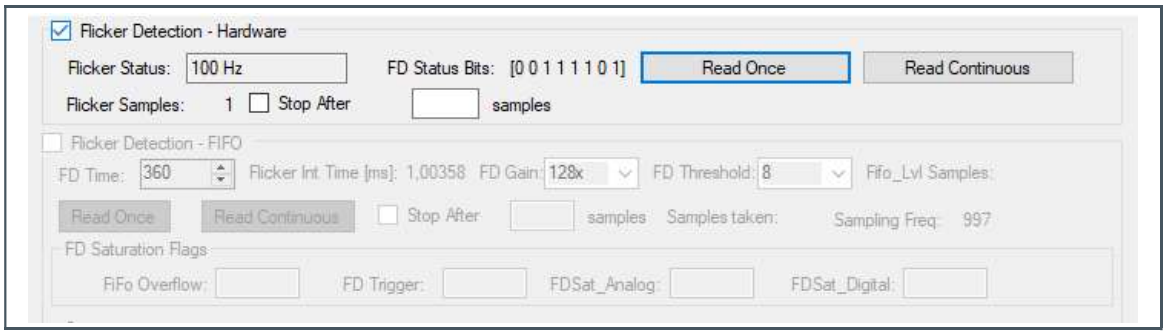
Figure 3 :
Flicker Detection of AS7341 Demo Software



2.1 Hardware Method

Selecting the “Flicker Detection – Hardware” checkbox enables the hardware flicker detection and disables the FIFO method of flicker detection, as shown in Figure 4 below.

Figure 4 :
Flicker Detection – Hardware (on chip Goertzel)



Hardware flicker detection detects the presence of 100Hz and 120 Hz flickering, using a Goertzel version of the FFT algorithm running on the AS7341. The “Read Once” button displays the status of the flicker in “Flicker status”. The FD status register is converted to binary bits and shown as “FD Status Bits”. Clicking the “Read Continuous” button once measures FD Status continuously in a loop. To measure mentioned samples times, select the “Stop after” checkbox and enter the number of samples to be measured in the textbox. Upon clicking the “Read Continuous” button, the mentioned samples will be measured in a loop. “Flicker Samples” shows the number of finished measurements.

2.2 FIFO Method

By default, the FIFO method of flicker detection (FD) is selected. The respective integration time is calculated based on the formula ((Fd_time +1)*2.78 µs) and displayed in ms. The Gain can be selected from the selection list. Similarly, FD Threshold can be selected. If the FIFO level is greater than FD Threshold, a FIFO interrupt (fint) is raised. FD Time and FD Gain define the sampling rate, amplitude of counts, and the maximum number counts.

Clicking the “Read Once” button updates the Flicker detection graph with current samples of FIFO data bytes. The output of the flicker channel in counts (raw data) is plotted against the corresponding sample number in the primary Y-axis (left) and X-axis respectively (bottom), as the red-colored graph. Clicking the “Read Continuous” button will keep updating the FD samples. If the “Stop after” checkbox is selected and the number of samples mentioned, then the mentioned samples start measurement after clicking “Read Continuous”. The “Samples taken” shows the number of finished measurements. In a single cycle, “FIFO_lvl Samples” gives the maximum number of FIFO entries (every 2 Bytes), read-out either before the overflow flag goes high or the maximum sample taken is less than 250. The sample levels add up back to back on consecutive measurements. Sampling Frequency is the reciprocal of the Flickering integration time. The “FD Saturation flags” shows the status of the saturation flag bit, like FIFO overflow, FD trigger, FDSat_Analog, and FDSat_Digital.

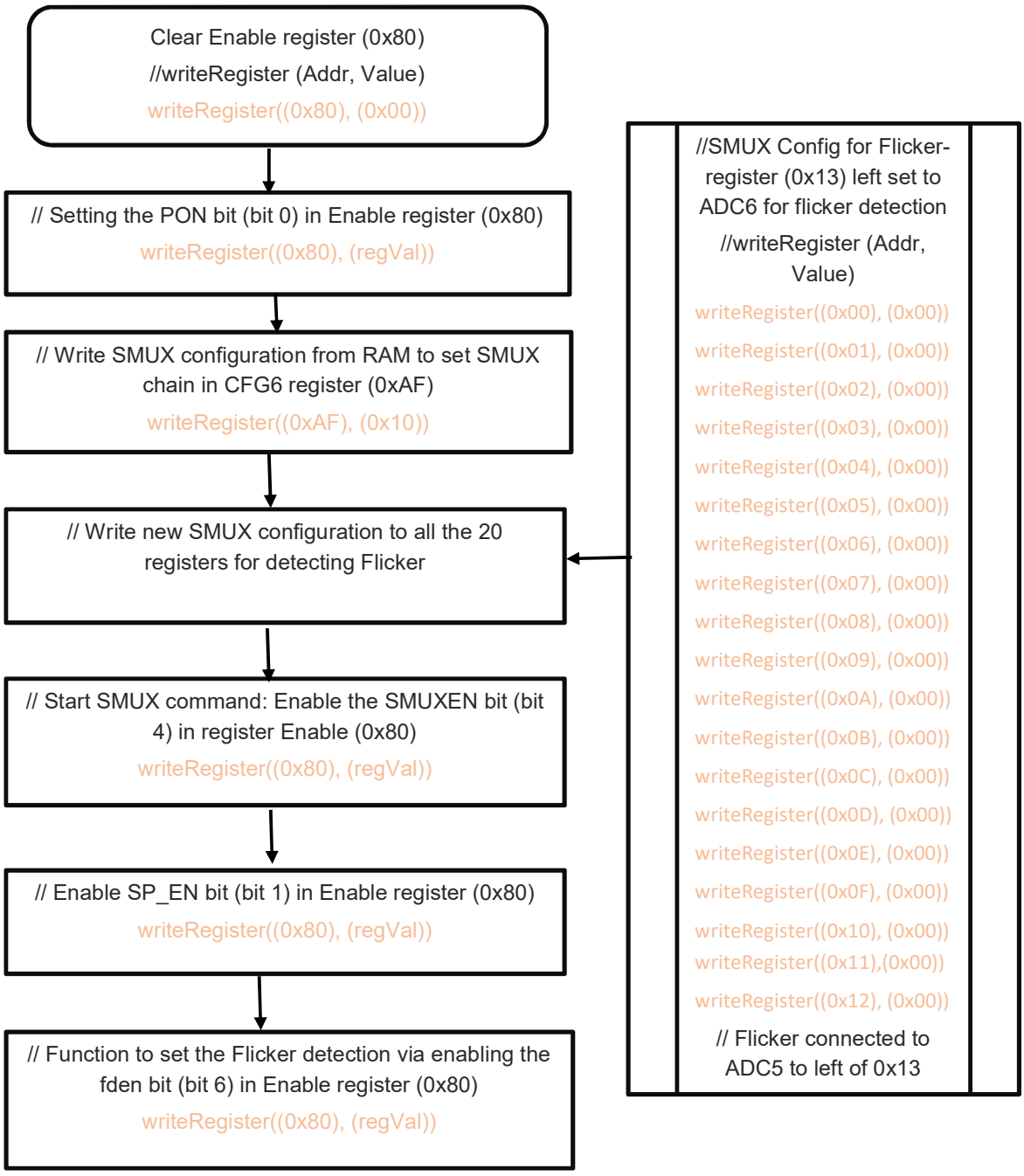
“FIFO Overflow” indicates an overflow of the FIFO data and the information that is lost. “FD trigger” indicates if there is not enough time to get and process flicker data. This can be solved by increasing the time for sampling. “FDSat_Analog” specifies that the intensity of ambient light has exceeded the maximum integration level for the analog circuit for flicker detection. “FDSat_Digital” shows that digital saturation is reached. The maximum cut-off value of digital saturation depends on the FD integration time.

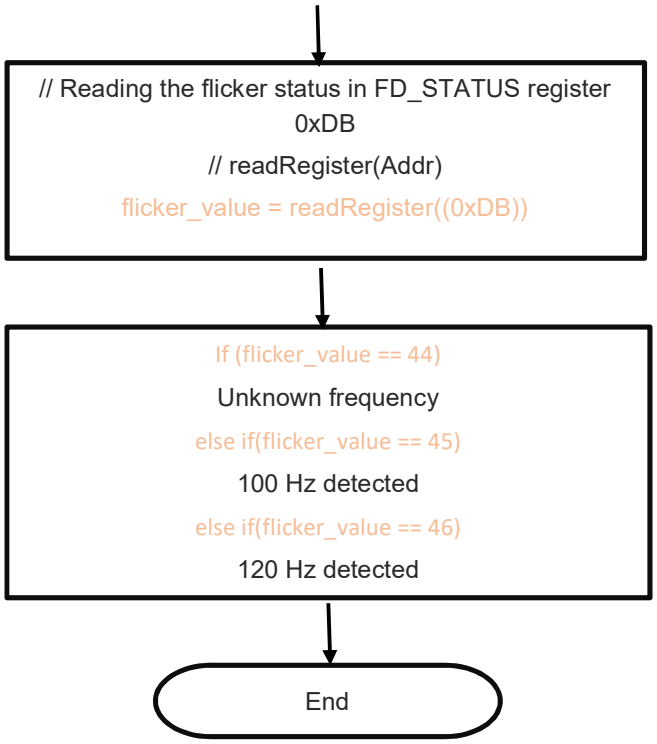
Fourier analysis converts a signal from its original domain (often time) to a representation in the frequency domain. Selecting the FFT checkbox computes the Fast Fourier Transformation and enumerates the discrete results graphically. The FFT plot is represented in blue with the amplitude of FFT on the secondary Y-axis (top) and Frequency represented on the secondary X-axis (right). The “Refresh” button clears out the last measurement data and plot.

3 Flowchart – Hardware and FIFO Methods

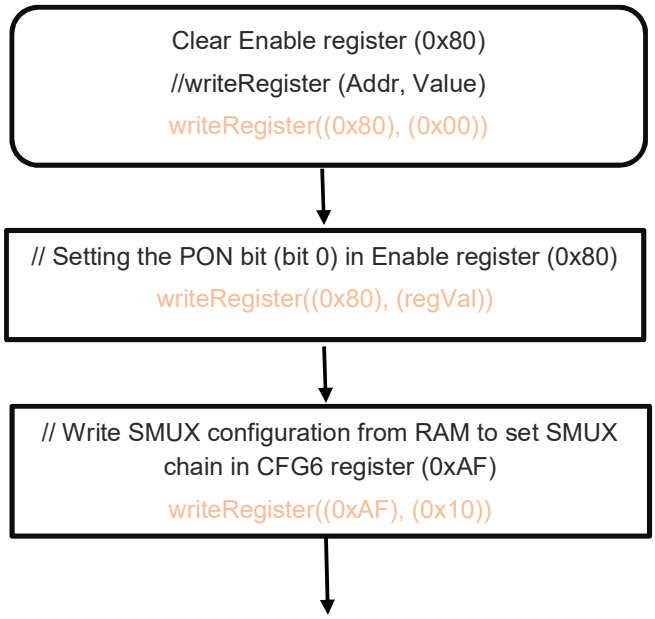
This section describes the flow of Register configuration and setting the Hardware and FIFO flicker measurements in AS7341 sensors.

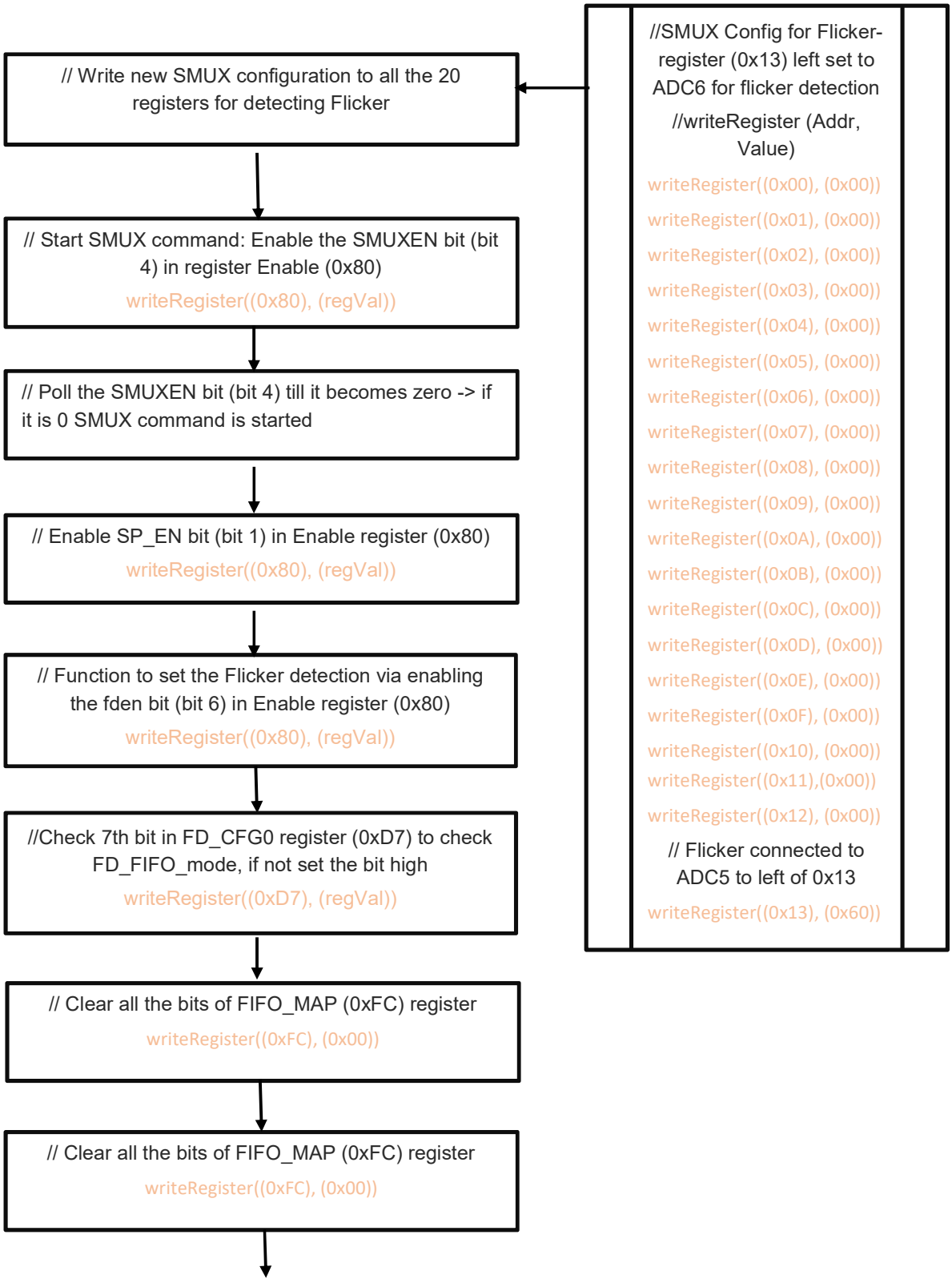
3.1 Hardware Method

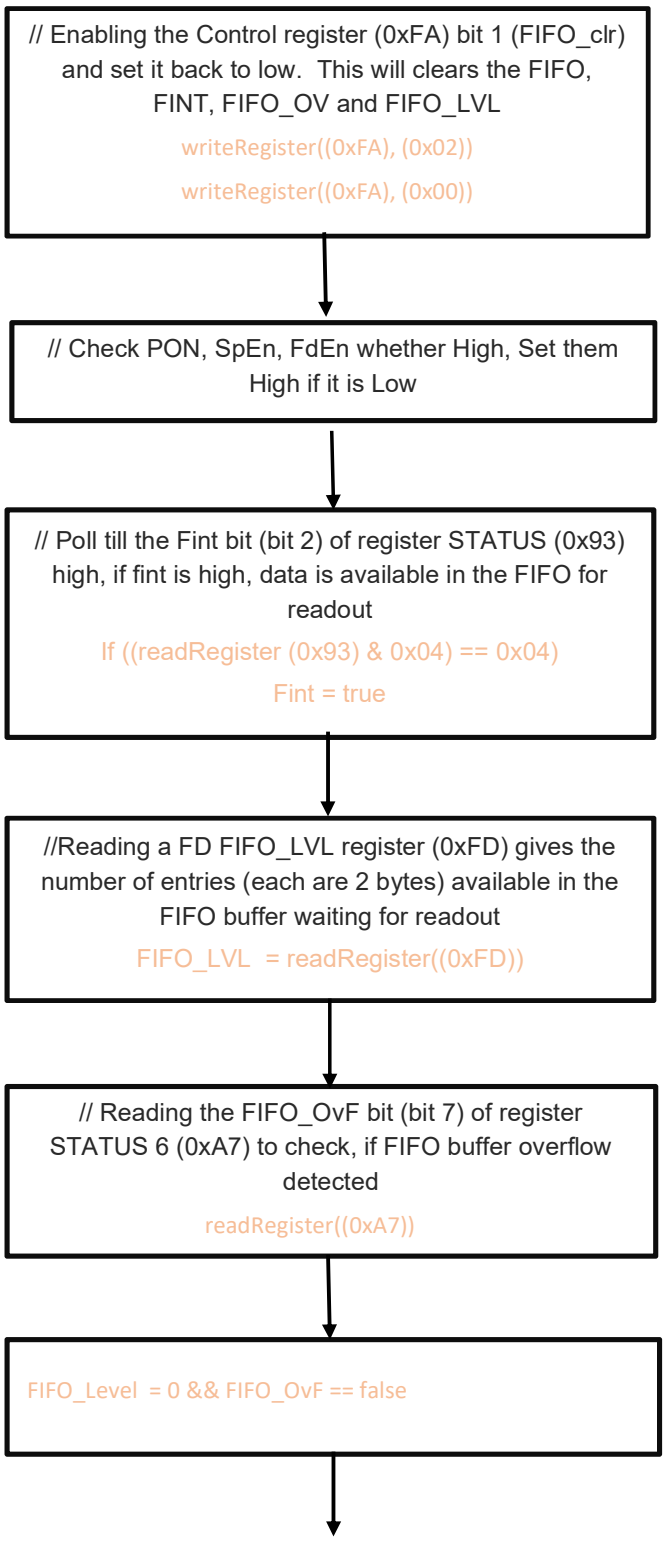


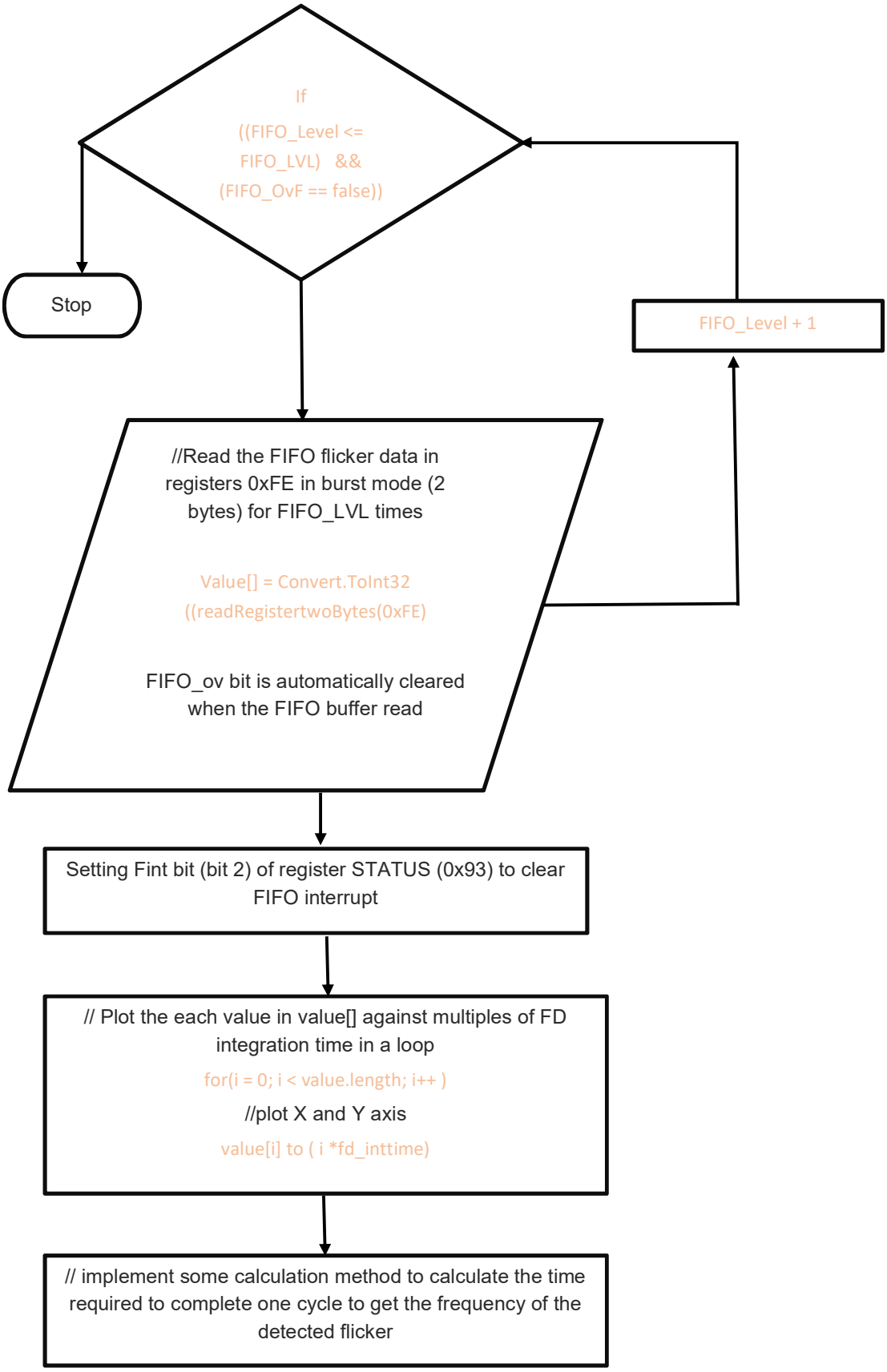


3.2 FIFO Method




// Write new SMUX configuration to all the 20 registers for detecting Flicker
//SMUX Config for Flicker- register (0x13) left set to ADC6 for flicker detection
//writeRegister (Addr, Value)
writeRegister((0x00), (0x00))
writeRegister((0x01), (0x00))
writeRegister((0x02), (0x00))
writeRegister((0x03), (0x00))
writeRegister((0x04), (0x00))
writeRegister((0x05), (0x00))
writeRegister((0x06), (0x00))
writeRegister((0x07), (0x00))
writeRegister((0x08), (0x00))
writeRegister((0x09), (0x00))
writeRegister((0x0A), (0x00))
writeRegister((0x0B), (0x00))
writeRegister((0x0C), (0x00))
writeRegister((0x0D), (0x00))
writeRegister((0x0E), (0x00))
writeRegister((0x0F), (0x00))
writeRegister((0x10), (0x00))
writeRegister((0x11),(0x00))
writeRegister((0x12), (0x00))
// Flicker connected to ADC5 to left of 0x13
writeRegister((0x13), (0x60))
// Start SMUX command: Enable the SMUXEN bit (bit 4) in register Enable (0x80)
writeRegister((0x80), (regVal))
// Poll the SMUXEN bit (bit 4) till it becomes zero -> if it is 0 SMUX command is started
// Enable SP_EN bit (bit 1) in Enable register (0x80)
writeRegister((0x80), (regVal))
// Function to set the Flicker detection via enabling the fden bit (bit 6) in Enable register (0x80)
writeRegister((0x80), (regVal))
//Check 7th bit in FD_CFG0 register (0xD7) to check FD_FIFO_mode, if not set the bit high
writeRegister((0xD7), (regVal))
// Clear all the bits of FIFO_MAP (0xFC) register
writeRegister((0xFC), (0x00))
// Clear all the bits of FIFO_MAP (0xFC) register
writeRegister((0xFC), (0x00))


// Enabling the Control register (0xFA) bit 1 (FIFO_clr) and set it back to low. This will clears the FIFO, FINT, FIFO_OV and FIFO_LVL
writeRegister((0xFA), (0x02))
writeRegister((0xFA), (0x00))
// Check PON, SpEn, FdEn whether High, Set them High if it is Low
// Poll till the Fint bit (bit 2) of register STATUS (0x93) high, if fint is high, data is available in the FIFO for readout
If ((readRegister (0x93) & 0x04) == 0x04)
Fint = true
//Reading a FD FIFO_LVL register (0xFD) gives the number of entries (each are 2 bytes) available in the FIFO buffer waiting for readout
FIFO_LVL = readRegister((0xFD))
// Reading the FIFO_OvF bit (bit 7) of register STATUS 6 (0xA7) to check, if FIFO buffer overflow detected
readRegister((0xA7))
FIFO_Level = 0 && FIFO_OvF == false


If
((FIFO_Level <=
FIFO_LVL) &&
(FIFO_OvF == false))
Stop
FIFO_Level + 1
//Read the FIFO flicker data in registers 0xFE in burst mode (2 bytes) for FIFO_LVL times
Value[] = Convert.ToInt32
((readRegistertwoBytes(0xFE)
FIFO_ov bit is automatically cleared when the FIFO buffer read
Setting Fint bit (bit 2) of register STATUS (0x93) to clear FIFO interrupt
// Plot the each value in value[] against multiples of FD integration time in a loop
for(i = 0; i < value.length; i++)
//plot X and Y axis
value[i] to (i *fd_inttime)
// implement some calculation method to calculate the time required to complete one cycle to get the frequency of the detected flicker

4 Flicker Detection by FIFO Method of AS7341 Using GUI

4.1 Description with Examples

For illuminating the FIFO method with an example,

LED light source is considered:-

White LED with 1.8V VDC, PWM frequency 300 Hz, Duty cycle 60%.

Figure 5:
Flickering Samples of LED Light Source Measured using FIFO Method

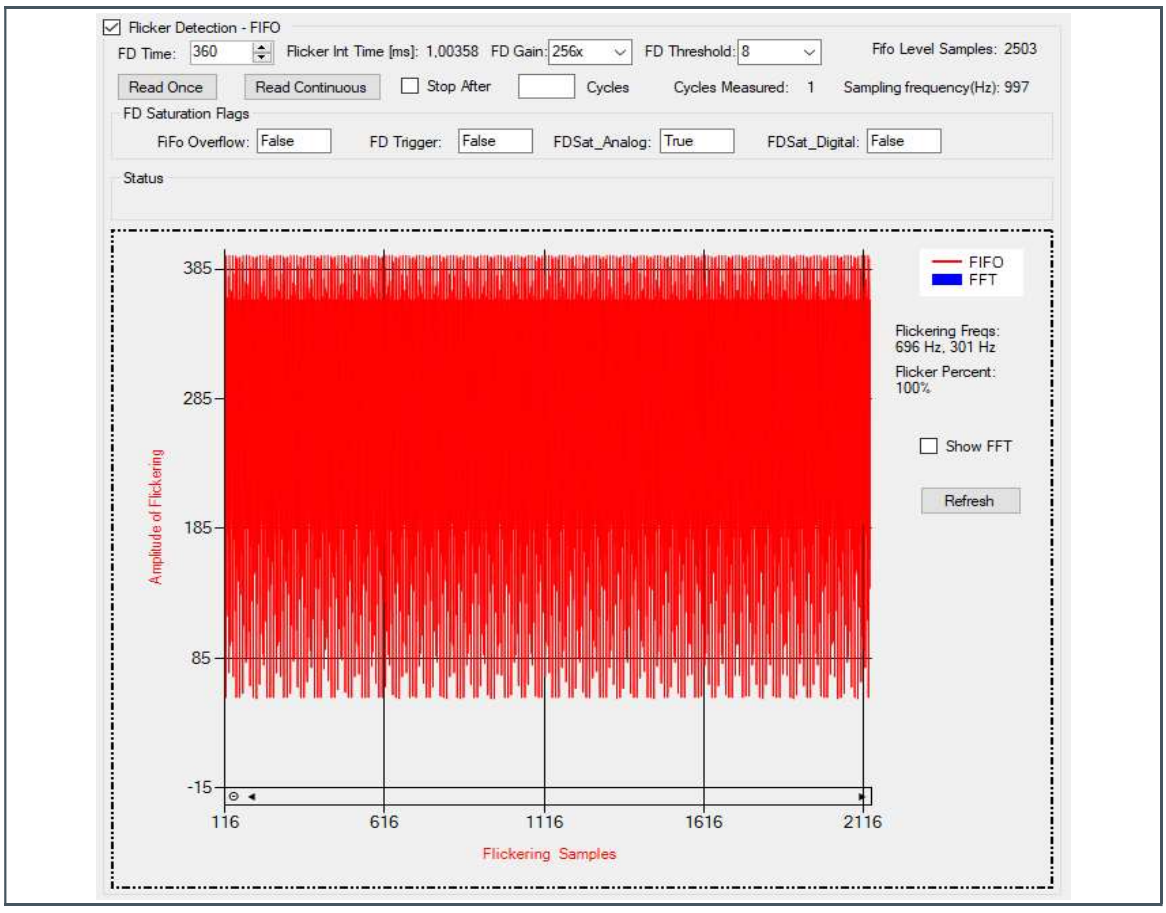
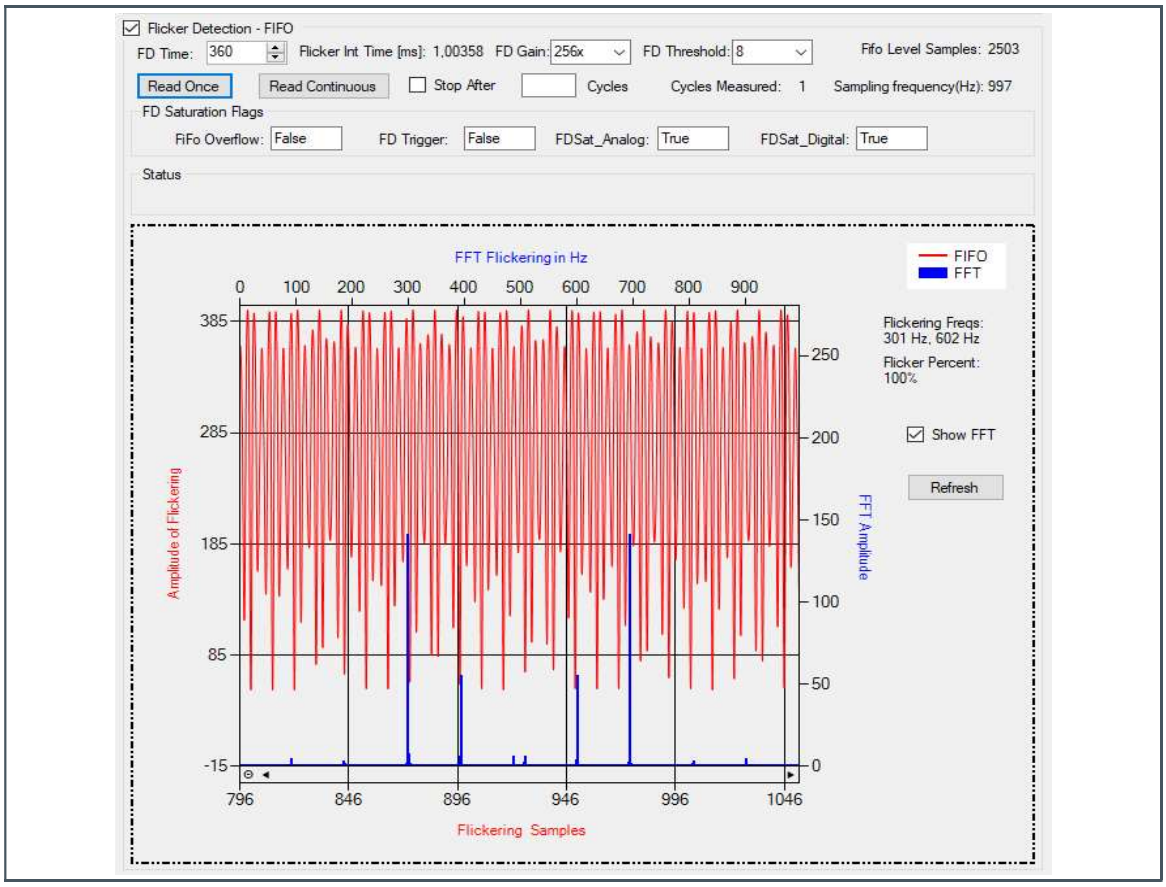


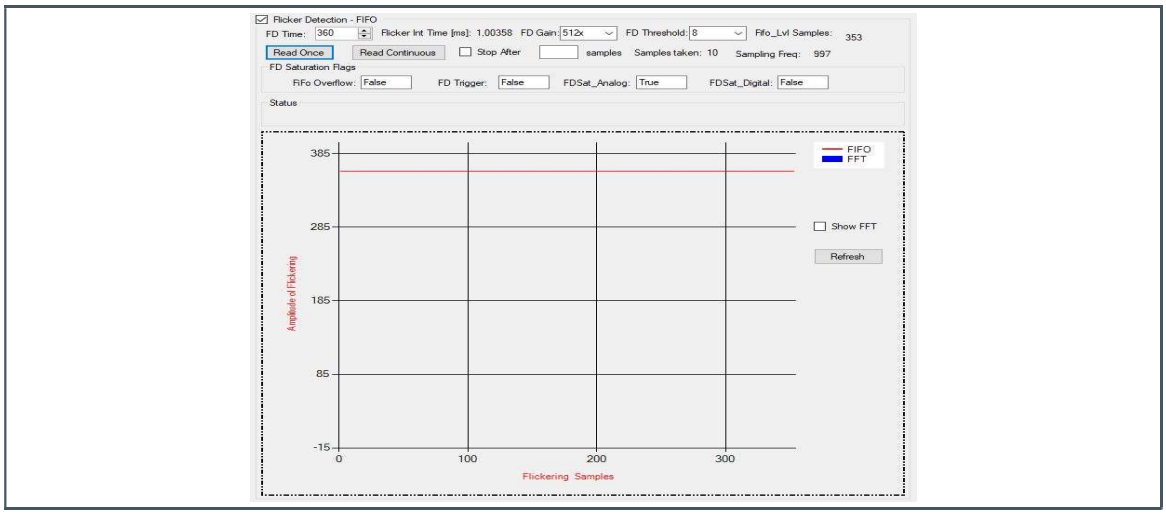
Figure 6 :
FIFO Flicker Detection and FFT Plot of LED Light Source



5 Factors Determining a Better Flicker Detection

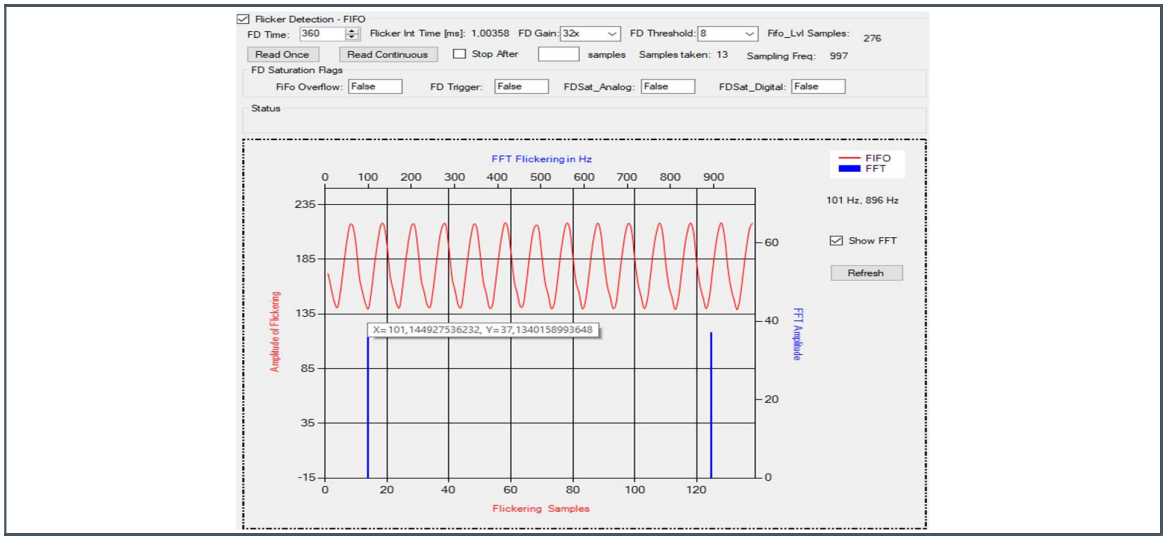
Example 1: 100Hz light source with FD Gain set to 512x. The sensor has sensed an analog saturation condition that is symbolized in the “FDSat_Analog” flag register in this case.

Figure 7 :
Analog Saturation at High Flicker Gain



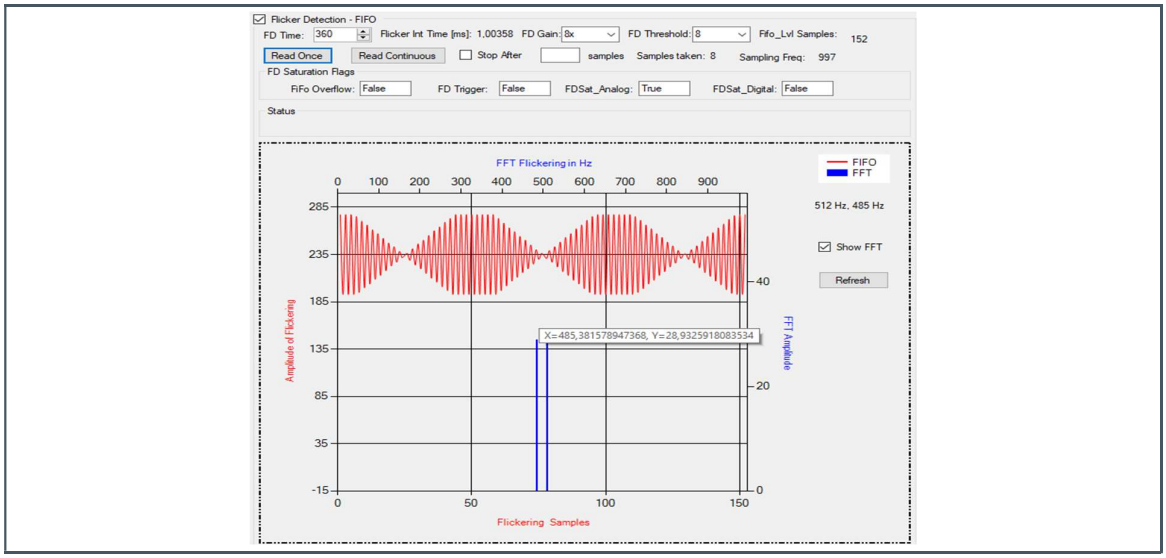
To solve the FD analog saturation, FD Gain should to be reduced. The result after reducing the gain is shown in Figure 8 below.

Figure 8 :
Optimal Gain to Get a Better Flicker Data



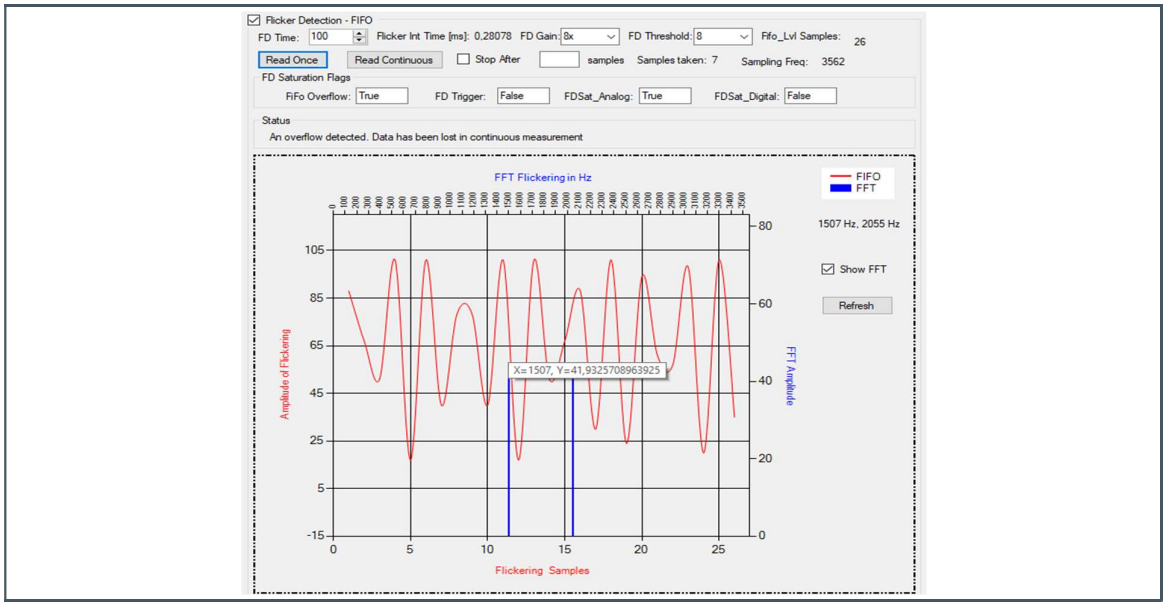
Example 2: 1500 Hz flickering light source. The maximum frequency range is not a range since it is a reciprocal of FD Integration time. Therefore, the setting should be adjusted accordingly.

Figure 9 :
Out of Range FFT Based on Selected Integration Time



Determining the optimal value of integration time will result in more precise detection of flickering, as show in Figure 10 below.

Figure 10 :
Optimal Integration Time for Better FFT



6 Revision Information

Changes from previous version to current revision v1-00	Page
Initial version	

- Page and figure numbers for the previous version may differ from page and figure numbers in the current revision.
- Correction of typographical errors is not explicitly mentioned.

7 Legal Information

Headquarters
ams AG
Tobelbader Strasse 30
8141 Premstaetten
Austria, Europe
Tel: +43 (0) 3136 500 0

Please visit our website at www.ams.com
Buy our products or get free samples online at www.ams.com/Products
Technical Support is available at www.ams.com/Technical-Support
Provide feedback about this document at www.ams.com/Document-Feedback
For sales offices, distributors and representatives go to www.ams.com/Contact
For further information and requests, e-mail us at ams_sales@ams.com