

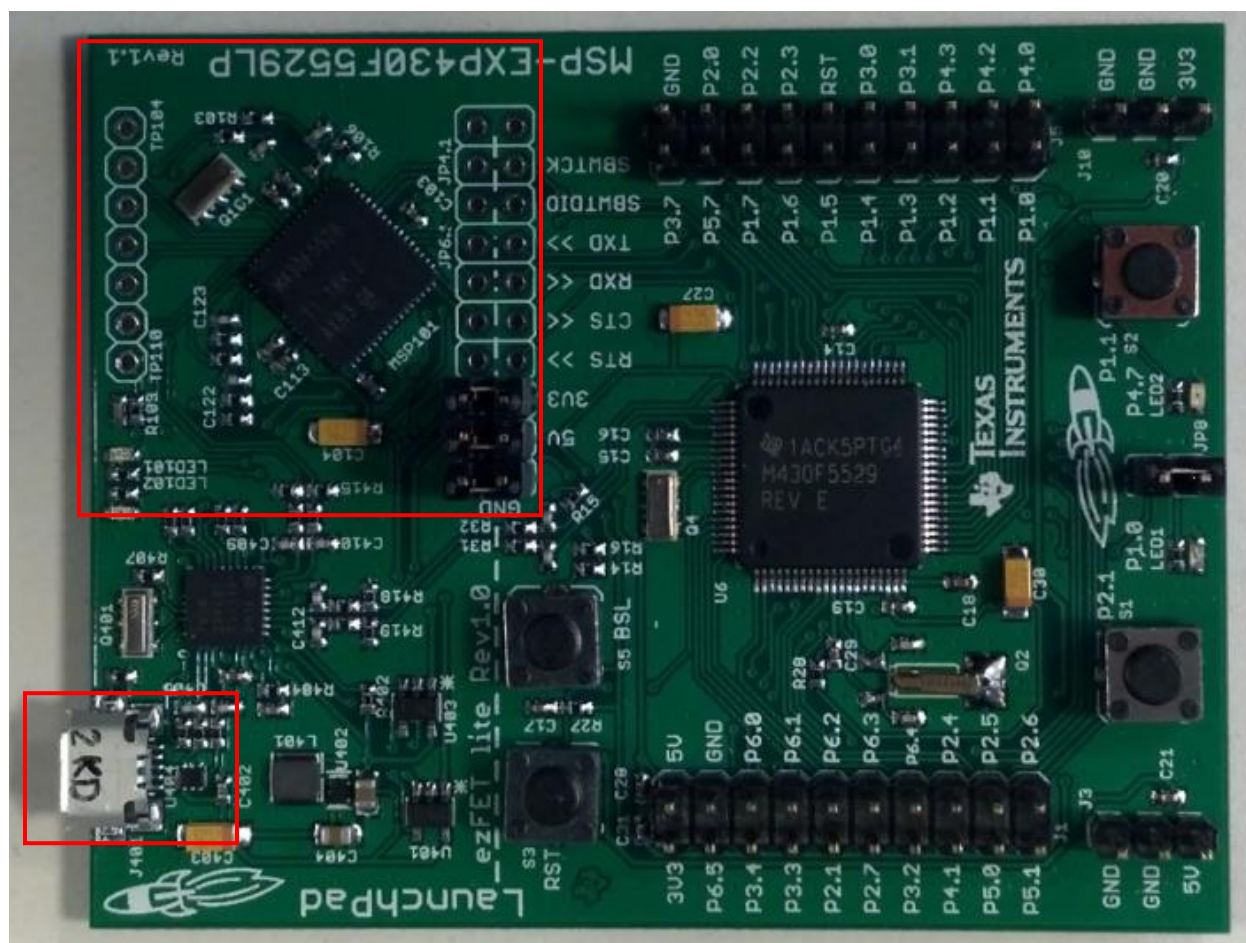
## ***eZ-FET lite On-Board Emulation Test Specification***

*MSP430 Tools*

This document describes the production test and programming of the eZ-FET lite on-board emulation. The eZ-FET lite on-board emulation is derived from the eZ-FET on-board emulation, with some features removed.

This document describes the IP revision **eZ-FET lite 1.10**

The test and programming sequence described in this document needs to be executed first. The remaining part of the target board (if any) need to be tested after the board passes the eZ-FET lite test. Please note that the target domain test is not within the scope of this Production Setup Package.



**Figure 1: Picture of MSP-EXP430F5529 Experimenters Board with previous eZ-FET lite 1.00 On-Board Emulation (eZ-FET lite components inside red rectangles)**

## Pre-Test Requirements

- a. Board needs to be fully assembled
- b. 100% Automated Optical Inspection of assembled PCB

## Required Test Equipment

- a. USB Micro-B cable to connect the eZ-FET lite to the Windows PC
- b. MSP-FET430UIF (with MSP430.DLLv3) and USB cable
  - a. Get the latest MSP-FET430UIF DLL from [www.ti.com/mspds](http://www.ti.com/mspds)
  - b. In case your MSP-FET430UIF is still operating with DLLv2 firmware, please follow the upgrade steps described on [www.ti.com/mspds](http://www.ti.com/mspds)
- c. MSP-FET430UIF Windows drivers
  - a. Included in Code Composer Studio 5.1 (or higher) or IAR Embedded Workbench for MSP430 5.40.2 (or higher)
  - b. Also available from [www.ti.com/mspds](http://www.ti.com/mspds)
- d. Custom JTAG adapter cable to connect MSP-FET430UIF to the eZ-FET
  - a. Pinout according to Table 2
- e. Personal computer with 32-bit or 64-bit Windows OS (GUI tested on Windows XP 32-bit, Windows 7 64-bit)
- f. Microsoft Visual C++ 2010 Redistributable Package (x86)
  - a. Available from <http://www.microsoft.com/en-us/download/details.aspx?id=5555>

## Test Guidelines

All tests will guide the test engineer through the test process.

The device is not switched to 'field'/deployment mode until tests have been completed/passed and the results recorded.

If a test fails once, the test should be repeated once. If it still fails, the board is label as defect and put aside for further investigation.

## Test Steps

The following steps are in order and it is assumed that they are executed in this order.

If a test fails, retry once. If it still fails. Label board as defect and put aside for further investigation.

**Table 1. Test Steps**

#	Step Description	Pass criteria	Time [s]
1	Connect blank eZ-FET via USB cable to PC	USB enumerates as HID component	
2	Connect MSP-FET430UIF to USB and JTAG cable to on-board connector (Table 2)		
3	Start GUI. Select log file and firmware images (Expert only. These settings come pre-configured in the ez-flasher.cfg text file).		
4	Press "Test and Program MSP430 eZ-FET" button		
5	Press 'Ok' if asked to update UIF firmware. Only required once if there is a firmware update available.	"MSP-FET430UIF Firmware updated successfully, retry downloading"	
6	Automatic functional test execution. If the test fails GUI displays message and red icon is on.		
7	Automatic programming of final production code	GUI "Production code running" message and green indicator label. Only Green LED on UIF on.	
8	Disconnect JTAG cable.		

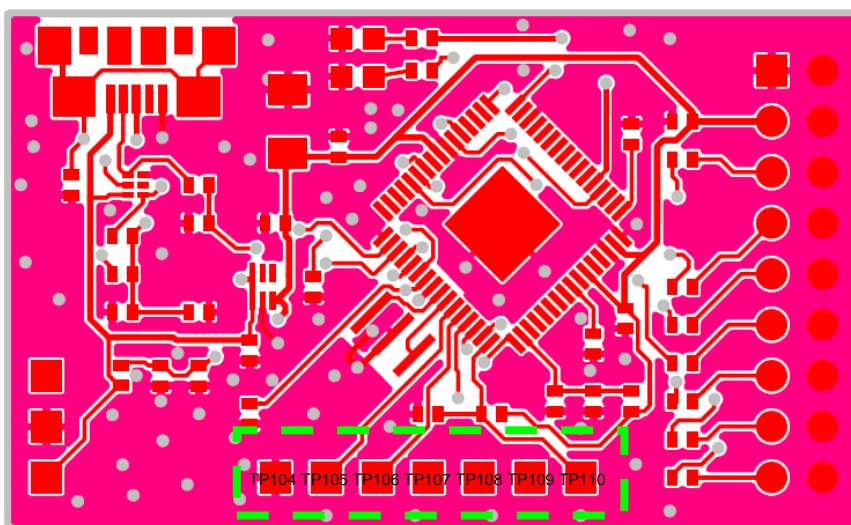
A log file is created automatically by the test program.

## eZ-FET lite Debug Connector

The eZ-FET lite emulation MCU can be accessed through a JTAG interface using the 4-wire JTAG protocol. The following table shows the connections between a standard 14-pin JTAG cable (output from MSP-FET430UIF) and the eZ-FET test pins.

**Table 2. eZ-FET Debug Connector Pinout**

Standard 14pin JTAG connector		On-board connector	
Signal Name	Pin Number	Pin Number	Signal Name
TDO	1	TP106	EZFET_TDO
TDI	3	TP107	EZFET_TDI
TMS	5	TP108	EZFET_TMS
TCK	7	TP109	EZFET_TCK
TEST	8	TP105	EZFET_TEST
GND	9	TP104	GND
RST	11	TP110	EZFET_RST



**Figure 2: eZ-FET lite 1.10 PCB Overview (green rectangle shows test points TP104 – TP110)**



**Figure 3: Connection between MSP-FET430UIF and eZ-FET lite (Picture shows MSP-EXP430F5529 Experimenters Board with eZ-FET lite 1.0)**

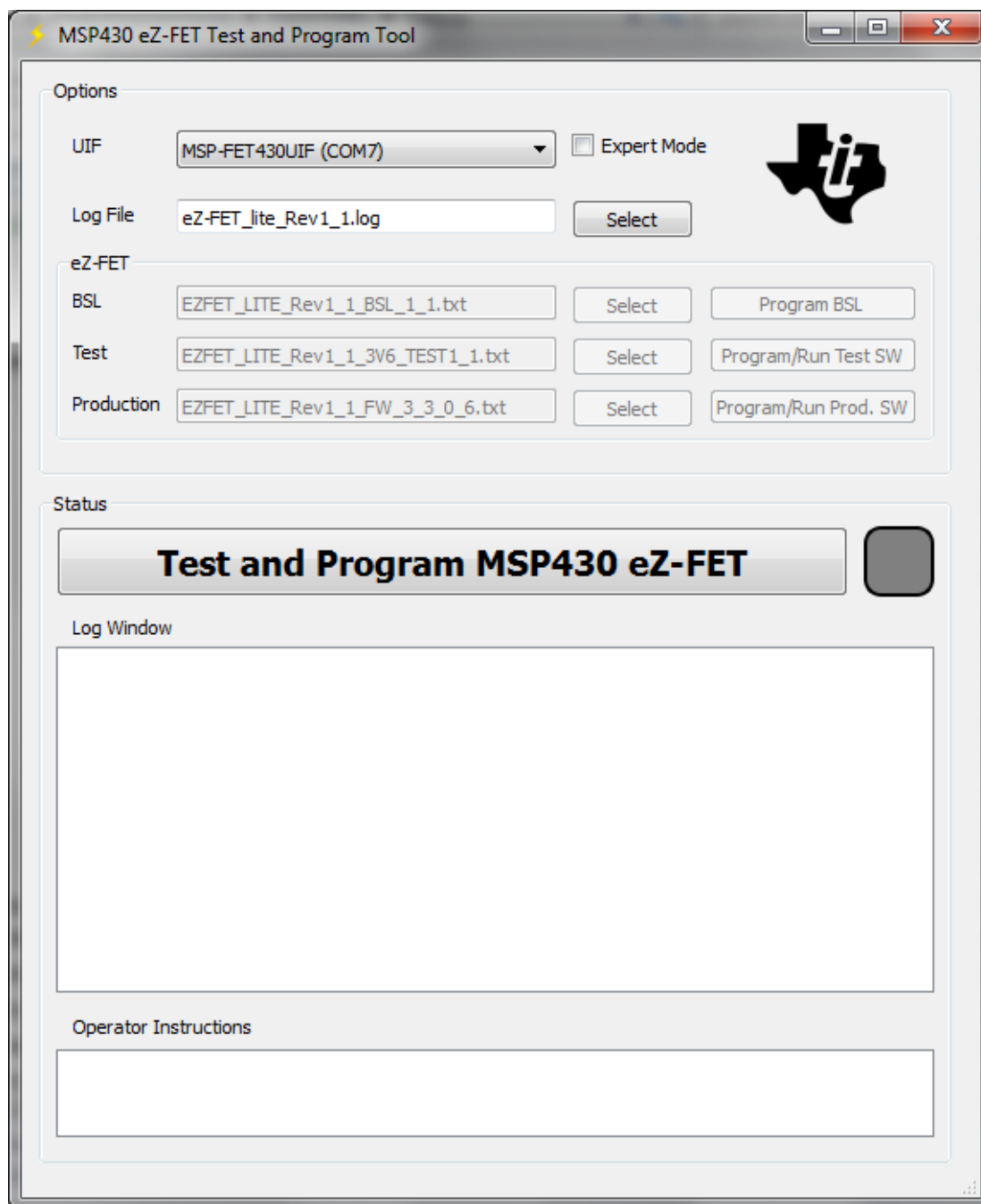
## **eZ-FET lite Test and Programming GUI**

The eZ-FET lite test and programming GUI is a graphical user-interface program that allows performing automatic tests and downloading production code into the eZ-FET lite emulation through a MSP-FET430UIF programming tool. Please note that you cannot use another eZ-FET lite to program the eZ-FET lite.

The eZ-FET lite test and programming GUI is based on the MSP430 Debug Stack (MSP430.dll) and has been developed with Qt 5.0.2 for Windows 32-bit (VS 2010) and Microsoft Visual Studio C++ Express 2010.

The user can easily select which test and production code to download on the device (in TI text format) and receive feedback about its electrical status, only with the push of a button. An “expert mode” is provided to have more control over the test sequence.

For convenience, all of the messages shown in the GUI status box are logged to the selected file with a timestamp and the message type (INFO, ERROR). Each programming sequence is appended to the file and delimited with a line, so old logs are never overwritten. Error codes have been taken from MSP430.DLL.

**Figure 4: Initial GUI screen**

On startup the GUI reads the configuration file “ez-flasher.cfg” where a number of filenames are predefined. Modify them according to your requirements.

**Configuration File**

```
# -----
#Log File
# -----
log_file=msp-exp430f5529.log

# -----
# eZ-FET
# -----
#BSL
bsl_file=EZFET_BSL_1_00.txt
#Test Firmware
emulation_test_file=EZFET_TEST_3V6_1_00.txt
#Production Firmware
emulation_production_file=EZFET_FW_1_00.txt
```

**Table 3. eZ-FET lite USB Bootstrap Loader Firmware**

IP	Revision	BSL firmware
eZ-FET lite	1.10	EZFET_LITE_Rev1_1_FW_3_3_0_6.txt

**Table 4. eZ-FET lite Test Firmware**

IP	Revision	VCC voltage	Test firmware
eZ-FET lite	1.10	2.8V	EZFET_LITE_Rev1_1_2V8_TEST1_1.txt
eZ-FET lite	1.10	3.0V	EZFET_LITE_Rev1_1_3V0_TEST1_1.txt
eZ-FET lite	1.10	3.3V	EZFET_LITE_Rev1_1_3V3_TEST1_1.txt
eZ-FET lite	1.10	3.6V	EZFET_LITE_Rev1_1_3V6_TEST1_1.txt

Please select the correct test firmware according to your supply voltage defined by IC101.

**Table 5. eZ-FET lite Production Firmware**

IP	Revision	BSL firmware
eZ-FET lite	1.10	EZFET_LITE_Rev1_1_BSL_1_1.txt

### Example Log File

2/4/2013 1:39:51 PM: START  
2/4/2013 1:39:51 PM: eZ-FET Test: DCDC MCU=0 (PASS)  
2/4/2013 1:39:51 PM: eZ-FET Test: Lot ID=1184520229 (PASS)  
2/4/2013 1:39:51 PM: eZ-FET Test: Die X-Pos=7, Die Y-Pos=29 (PASS)  
2/4/2013 1:39:51 PM: eZ-FET Test: XT2=3964928Hz (PASS)  
2/4/2013 1:39:51 PM: eZ-FET Test: VBUS=4955mV (PASS)  
2/4/2013 1:39:51 PM: eZ-FET Test: VOUT=3625mV (PASS)  
2/4/2013 1:39:51 PM: All eZ-FET tests have passed.

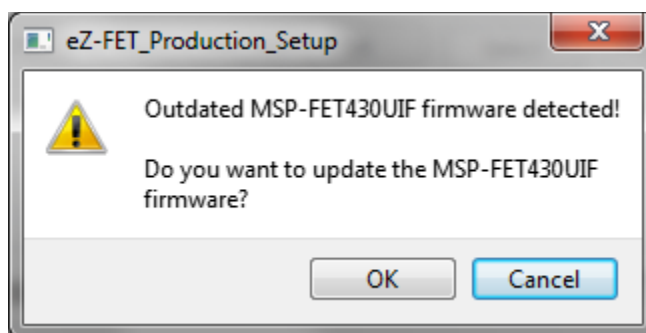


Figure 5: MSP-FET430UIF needs firmware update – Please select “OK”

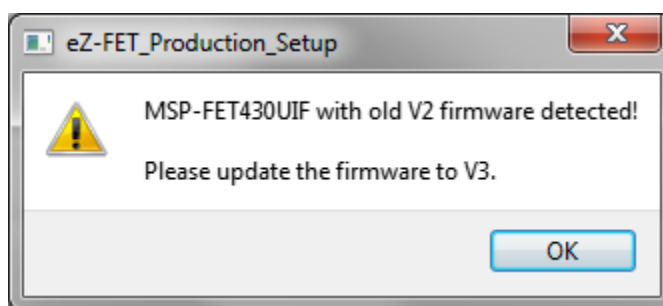
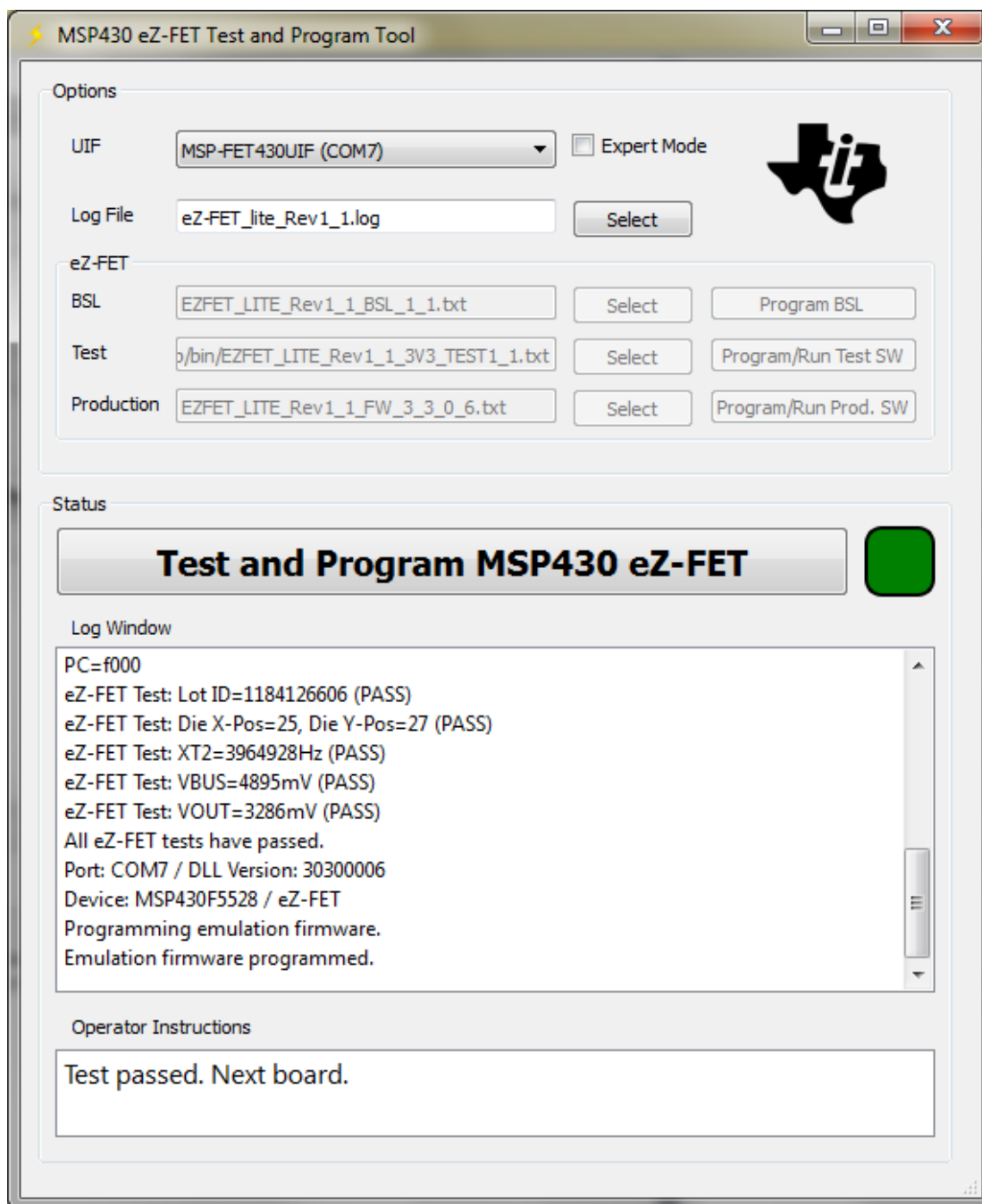


Figure 6: MSP-FET430UIF with DLLv2 firmware detected. Please use the update process described on [www.ti.com/mspds](http://www.ti.com/mspds) to update to DLLv3





**Figure 7: eZ-FET lite test and programming was successful**

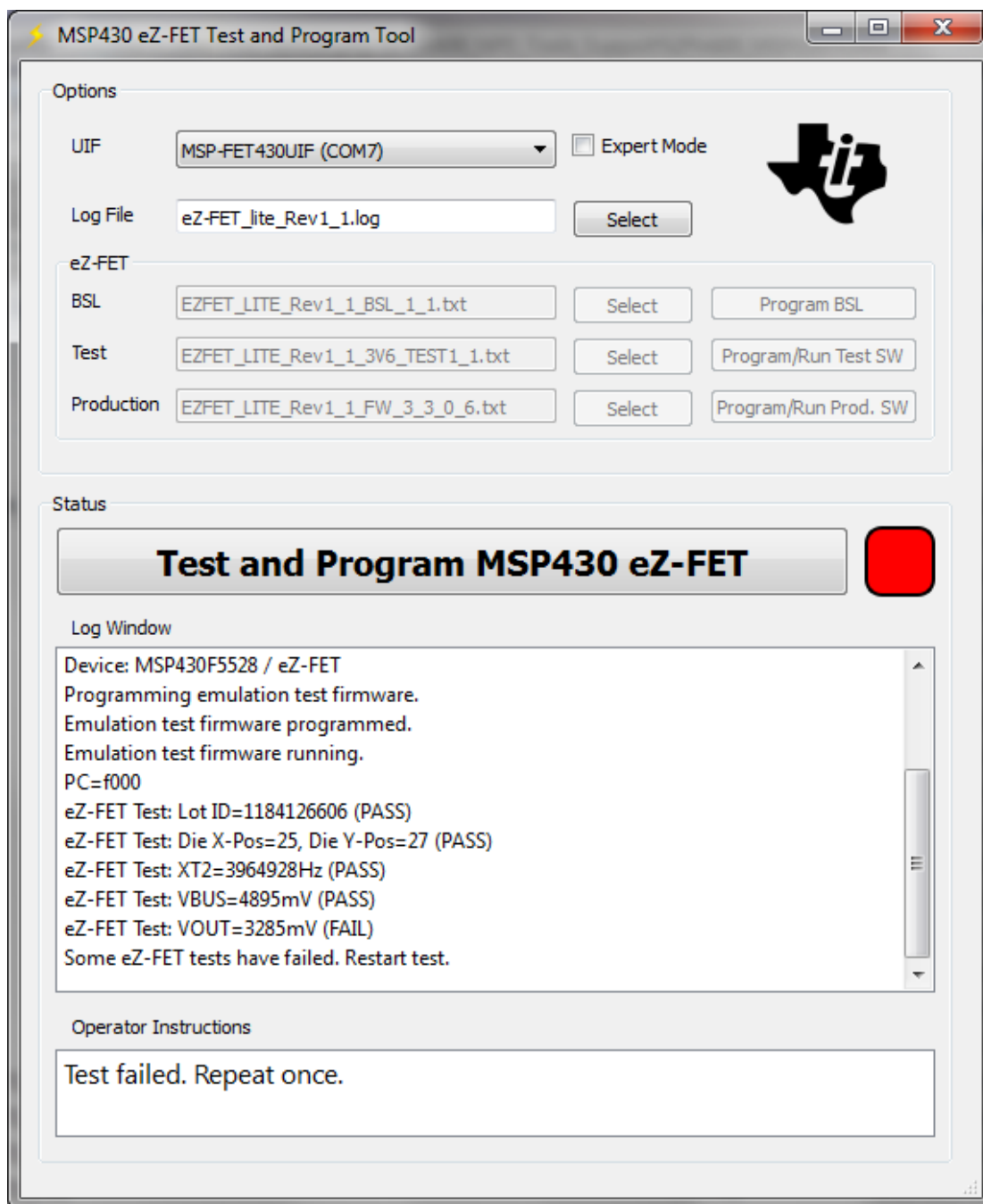


Figure 8: eZ-FET lite test and programming failed due to wrong test firmware – use 3V3 firmware to test this board

## **Firmware Production Test Code Specification**

The production test firmware performs automatic electrical tests on the eZ-FET emulation.

These tests are normally sufficient to detect the most common problems in the emulation domain, however we recommend to also test and program the target domain using the eZ-FET on-board emulation (e.g. with a MSP430-Flasher project) and only release boards from manufacturing that have passed a full test covering normal mode of usage (erase, download and run code, set breakpoints etc.).

### ***Firmware Test Functions***

- a. System Initialize
  - a. Configure GPIOs for correct termination
  - b. Set MCLK to 18 MHz, SMCLK to 4 MHz, ACLK to 32768 Hz
  - c. Set MCLK to 20 MHz
- b. Perform electrical test
  - a. Initialize fixed-location RAM variables and read TLV table for silicon die info
  - b. Test XT2 Frequency, compare to REFO
    - i. Use the internal connection from TA2 CCI2B to ACLK, with SMCLK as the timer clock, collect a few captures to ensure the crystal frequency is correct.
  - c. Test supply voltages
    - i. Perform ADC conversions on VBUS and VCC\_SUPPLY
  - d. Check if the collected values comply with the specifications
    - i. Turn red(fail)/green(pass) LEDs on accordingly
- c. Data log test results
  - a. Test data is stored in fixed RAM locations, so that the PC GUI can read them using MSP430.DLL
  - b. Pass criteria can be stored as '0xAA and fail as '0x55'
  - c. In addition to supply voltages and XT2 frequency values, Lot/Wafer ID and XY die position are collected and stored.

### ***Items not tested by the Firmware Production Test Code***

- Target domain
  - Current consumption

- Functionality
- Emulation domain
  - Field updates over USB
  - Current consumption in idle and debug mode
  - Long-time stability of oscillators etc.
  - LED functionality

### **Optical Results / LED feedback**

On-board LEDs will light up to indicate either failure or success of the test run. The table below describes the flashing patterns.

**Table 6. Optical test result feedback**

<b>Result</b>	<b>Green LED (LED102)</b>	<b>Red LED (LED101)</b>
Test FAIL	Off	On
Test SUCCESS	On	Off

### **Feedback**

Feedback is welcome. If anything is not smooth or clear, please highlight so it can be fixed in the next release.

### **Revision History**

<b>Date</b>	<b>Description</b>
6/29/2012	Initial check in
8/16/2012	Changed some details, added screenshots. Removed MSP-FET reference
8/17/2012	Adding details to make it easier at production. Porting to the EVM Development Process template. Initial release to CM for production
8/30/2012	Update Screenshots. Update references.
2/4/2013	Major update for eZ-FET 1.00 / 1.10 release
4/9/2013	Added eZ-FET lite 1.10 PCB Overview
4/16/2013	Updated pictures and screenshots, fixed typos.
5/2/2013	Removed license text, added some explanations