



SFRA LIB Release Notes

V1_50_00_00	<ul style="list-style-type: none"> Added FPU64 library objects (.lib)
V1_40_00_00	<ul style="list-style-type: none"> Added support for plotting closed loop transfer function SFRA GUI is modularized to enable easy integration Added support to change the speed of the sweep by passing a parameter to the configuration routine of the SFRA Following are the changes required to use this library compared to the previous version in the user code using this library, exact steps may vary depending on your project structure. <ol style="list-style-type: none"> Configuration: Three new parameters are passed to the configuration routine to enable measurement of closed loop transfer function. NULL can be passed for the pointers if closed loop function does not need to be plotted. <p>previous</p> <pre>SFRA_F32_config(&sfra1, SFRA_ISR_FREQ_HZ, SFRA_AMPLITUDE, SFRA_FREQ_LENGTH, SFRA_FREQ_START, SFRA_FREQ_STEP_MULTIPLY, plantMagVect, plantPhaseVect, olMagVect, olPhaseVect, freqVect);</pre> <p>now:</p> <pre>SFRA_F32_config(&sfra1, SFRA_ISR_FREQ_HZ, SFRA_AMPLITUDE, SFRA_FREQ_LENGTH, SFRA_FREQ_START, SFRA_FREQ_STEP_MULTIPLY, plantMagVect, plantPhaseVect,</pre>

```
oiMagVect,  
oiPhaseVect,  
NULL,  
NULL,  
freqVect,  
1);
```

2. Also additional header file must be include for NULL definition

```
#include <stddef.h>
```

3.Link the SCI Comms GUI file from sfra/gui/source instead of sfra/gui/

4. Add include path for sfra/gui/include in the include paths

5. Delete all these variables from the solution file for the SFRA GUI as due to modularization of the SFRA GUI these are now part of the SCI comms GUI, this also redcues the steps to include SFRA GUI

Remove

```
//  
//need to keep SFRA GUI related variables here  
// otherwise they might conflict when combining mutiple solution files  
//  
//  
//-- SFRA GUI Related Variables ----  
//  
int16_t SerialCommsTimer;  
int16_t CommsOKflg;  
//  
//Flag for reinitializing SFRA variables  
//  
int16_t initializationFlag;  
  
//  
//GUI support variables  
// sets a limit on the amount of external GUI controls--increase as  
// necessary  
//16 textbox controlled variables  
//  
int16_t *varSetTxtList[16];  
//  
//16 button controlled variables  
//  
int16_t *varSetBtnList[16];  
//  
//16 slider controlled variables  
//  
int16_t *varSetSlidrList[16];  
//  
//16 variables sendable to GUI  
//  
int16_t *varGetList[16];  
//  
//16 arrays sendable to GUI  
//
```

```
int32_t *arrayGetList[16];
//
//16 32-bit textbox or label controlled variables
//
uint32_t *dataSetList[16];
```

6. Add the sfra_gui_scicomms_driverlib.h include along with the sfra header file

```
#include "sfra_f32.h"
#include "sfra_gui_scicomms_driverlib.h"
```

7. Delete the following piece of code if it exists, all of this now is handled by the modularized SFRA GUI

```
if(initializationFlag == 1)
{
initializationFlag = 0;
sfra1.start = 1;
}
```

8. Delete the old configuration , shown below which is very long

```
SFRA_GUI_config(SCI_VBUS_CLK, SFRA_GUI_SCI_BAUDRATE);
```

```
initializationFlag = 0;
//
//"Set" variables
// assign GUI Buttons to desired flag addresses
//
varSetBtnList[0] = (int16_t *)&initializationFlag;

//
//"Get" variables
//-----
// assign a GUI "getable" parameter address
//
varGetList[0] = (int16_t *)&( sfra1.vecLength);
varGetList[1] = (int16_t *)&( sfra1.status);
varGetList[2] = (int16_t *)&( sfra1.freqIndex);

//
//"Setable" variables
//-----
// assign GUI "setable" by Text parameter address
//
dataSetList[0] = (uint32_t *)&( sfra1.freqStart);
dataSetList[1] = (uint32_t *)&( sfra1.amplitude);
dataSetList[2] = (uint32_t *)&( sfra1.freqStep);

//
// assign a GUI "getable" parameter array address
//
arrayGetList[0] = ((int32_t *) freqVect);
arrayGetList[1] = ((int32_t *) olMagVect);
arrayGetList[2] = ((int32_t *) olPhaseVect);
```

```

arrayGetList[3] = ((int32_t *) plantMagVect);
arrayGetList[4] = ((int32_t *) plantPhaseVect);
arrayGetList[5] = ((int32_t *) &( sfra1.freqStart));
arrayGetList[6] = ((int32_t *) &( sfra1.amplitude));
arrayGetList[7] = ((int32_t *) &( sfra1.freqStep));

```

```

CommsOKflg = 0;
SerialCommsTimer = 0;

```

Add the below , new SFRA GUI init code

```

//
//configures the SCI channel for communication with SFRA host GUI
//to change SCI channel change #defines in the settings.h file
//the GUI also changes a LED status, this can also be changed with #define
//in the file pointed to above
//
SFRA_GUI_config(SFRA_GUI_SCI_BASE,
SCI_VBUS_CLK,
SFRA_GUI_SCI_BAUDRATE,
SFRA_GUI_SCIRX_GPIO,
SFRA_GUI_SCIRX_GPIO_PIN_CONFIG,
SFRA_GUI_SCITX_GPIO,
SFRA_GUI_SCITX_GPIO_PIN_CONFIG,
SFRA_GUI_LED_INDICATOR,
SFRA_GUI_LED_GPIO,
SFRA_GUI_LED_GPIO_PIN_CONFIG,
&sfra1,
1);

```

9. Modify the settings,h file and/or add the following defines ,

```

//
// SFRA related
//
#define SFRA_GUI_SCI_BASE SCIA_BASE
#define SCI_VBUS_CLK 50000000
#define SFRA_GUI_SCI_BAUDRATE 57600

#define SFRA_GUI_SCIRX_GPIO 28
#define SFRA_GUI_SCITX_GPIO 29

#define SFRA_GUI_SCIRX_GPIO_PIN_CONFIG GPIO_28_SCIRXDA
#define SFRA_GUI_SCITX_GPIO_PIN_CONFIG GPIO_29_SCITXDA

// if the following #define is set to 1 SFRA GUI indicates status on LED
// otherwise LED code is ignored
#define SFRA_GUI_LED_INDICATOR 1
#define SFRA_GUI_LED_GPIO 31
#define SFRA_GUI_LED_GPIO_PIN_CONFIG GPIO_31_GPIO31

```

10. Delete the following declarations, these are now part of the modularized SFRA GUI (sfra_scicomms_driverlib.h) header file and are not needed in user code.

```

void SFRA_GUI_config(uint32_t vbus_clk, uint32_t sci_baud);

```

	<pre>void SFRA_GUI_serialHostComms();</pre> <p>11. Delete the following, the modularized SFRA GUI handles this now and add the following text.</p> <pre>SFRA_GUI_serialHostComms(); SerialCommsTimer++;</pre> <pre>SFRA_F32_runBackgroundTask(&sfra1); SFRA_GUI_runSerialHostComms(&sfra1);</pre>
V1_30_01_00	<ul style="list-style-type: none"> Fixed extern in scicommsgui file, the variable *arrayGetList[] is externed as int16_t whereas it points to int32_t. It is not changes to extern to a int32_t. The issue generated error in the new SDK project for type mis-match.
V1_30_00_00	<ul style="list-style-type: none"> Major update to library interface to comply with coding standard for C2000 SW Removed deprecated examples that don't work in SDK framework Added STB project using f28004x to test SFRA standalone Removed all dependence to IQmath library Updated documentation with new lit number for the SDK release of the library <p>Following are the changes required in a typical application for SFRA Lib usage when migrating from an older version to this new library.</p> <ol style="list-style-type: none"> The header file name has changed from SFRA_F_Include.h to SFRA_F32.h, this is done to be compliant to new directory structure and file name convention. The directory structure has changed and simplified, the new directory is as below (you will need to change the include path under compiler options to point correctly, also modify the linker options to point to the correct file):

- sfra
 - lib
 - sfra_f32.lib
 - sfra_f32_tmu.lib
 - include
 - sfra_f32.h
 - gui
 - sfra_gui_scicomms_driverlib.c

3. The **SFRA object** will be SFRA_F32,

~~volatile SFRA_F sfra1;~~

SFRA_F32 sfra1;

4. May need to re-define for consistency in solution.h file

~~#define FREQ_STEP_MULTIPLY (float)1.07~~

#define SFRA_FREQ_STEP_MULTIPLY (float)1.07

5. Name of inject, collect and background functions have changed from

~~SFRA_F_INJECT~~ -> SFRA_F32_inject

~~SFRA_F_COLLECT~~ -> SFRA_F32_collect

~~SFRA_F_BACKGROUND~~ -> SFRA_F32_runBackgroundTask

6. Additional type casting may be required due to volatile and non volatile argument changes

SFRA_F32_collect((float*)&gi_out, (float*)&ac_cur_sensed);

7. Init routine

New

```
SFRA_F32_reset(&sfra1);
SFRA_F32_config(&sfra1,
                SFRA_ISR_FREQ,
                SFRA_AMPLITUDE,
                SFRA_FREQ_LENGTH,
                SFRA_FREQ_START,
                plantMagVect,
                plantPhaseVect,
                olMagVect,
                olPhaseVect,
                freqVect);
```

```

SFRA_F32_resetFreqRespArray(&sfra1);

SFRA_F32_initFreqArrayWithLogSteps(&sfra1,
                                     SFRA_FREQ_START,
                                     FREQ_STEP_MULTIPLY);

```

Old

```

//Specify the injection amplitude
sfra1.amplitude=SFRA_AMPLITUDE;

SFRA_F_INIT(&sfra1);

//SFRA Related
//SFRA Object Initialization
//Specify the injection amplitude
sfra1.amplitude=SFRA_AMPLITUDE;
//Specify the length of SFRA
sfra1.Vec_Length=SFRA_FREQ_LENGTH;
//Specify the SFRA ISR Frequency
sfra1.ISR_Freq=SFRA_ISR_FREQ;
//Specify the Start Frequency of the SFRA analysis
sfra1.Freq_Start=SFRA_FREQ_START;
//Specify the Frequency Step
sfra1.Freq_Step=FREQ_STEP_MULTIPLY;
//Assign array location to Pointers in the SFRA object
sfra1.FreqVect=freqVect;
sfra1.GH_MagVect=olMagVect;
sfra1.GH_PhaseVect=olPhaseVect;
sfra1.H_MagVect=plantMagVect;
sfra1.H_PhaseVect=plantPhaseVect;

```

8. CMD file changes

```

SFRA_F_Data : > RAMGS2, ALIGN = 64, PAGE = 1
SFRA_F32_Data : > RAMGS2, ALIGN = 64, PAGE = 1

```

9. Copy the new sfra_gui_scicomms_driverlib.c file in the project.

Delete the link/copy of the old scicommsgui file.

10. Because of the change in SFRA GUI initialization you will need to make the following changes in settings.h file, note the SFRA SCI gui channel and pin is now determined in the sfra_gui_scicomms_driverlib.c file

```

#define GPIO_SFRA_SCIRX 28

```

```
#define GPIO_SFRA_SCITX 29
#define GPIO_SFRA_SCIRX_PIN_CONFIG GPIO_28_SCIRXDA
#define GPIO_SFRA_SCITX_PIN_CONFIG GPIO_29_SCITXDA
#define SCI_VBUS_CLK 50000000
#define SFRA_GUI_SCI_BAUDRATE 57600
```

You can also remove the following routine from the board.c file, as this is now part of the sfra_guicomms file

```
//TODO setupSCIconnectionForSFRA()
void setupSCIconnectionForSFRA()
{
    // setup Gpio for SCI comms for SFRA
    GPIO_setPinConfig(GPIO_SFRA_SCIRX_PIN_CONFIG);
    GPIO_setPinConfig(GPIO_SFRA_SCITX_PIN_CONFIG);
    GPIO_setQualificationMode(GPIO_SFRA_SCIRX,GPIO_QUAL_SYNC);
    ;
    GPIO_setQualificationMode(GPIO_SFRA_SCITX,GPIO_QUAL_SYNC);
    EDIS;

    // 50000000 is the LSPCLK or the Clock used for the SCI Module
    // 57600 is the Baudrate desired of the SCI module
    SCIA_Init(50000000, 57600);
}
```

11. Add declaration for

```
void SFRA_GUI_config(uint32_t vbus_clk,uint32_t baudrate);
void SFRA_GUI_serialHostComms(void);
```

remove declaration for

```
void SerialHostComms(void);
void SCIA_Init(long SCI_VBUS_CLOCKRATE, long SCI_BAUDRATE);
```

12. Rename

```
SerialHostComms-> SFRA_GUI_serialHostComms();
```

13. You may need to re-define for consistency in solution.h file.

```
#define FREQ_STEP_MULTIPLY (float)1.07
```


	<code>#define SFRA_FREQ_STEP_MULTIPLY (float)1.07</code>
V1_20_00_00 or previous releases	<ul style="list-style-type: none">• controlSUITE release and used for the first few SDK release