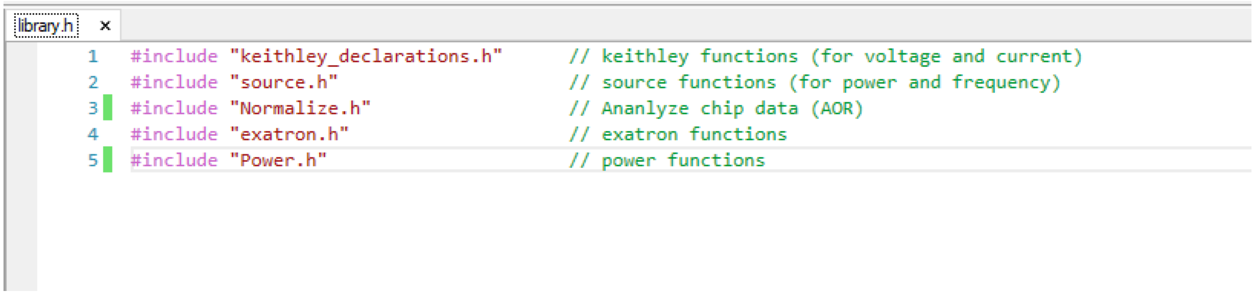
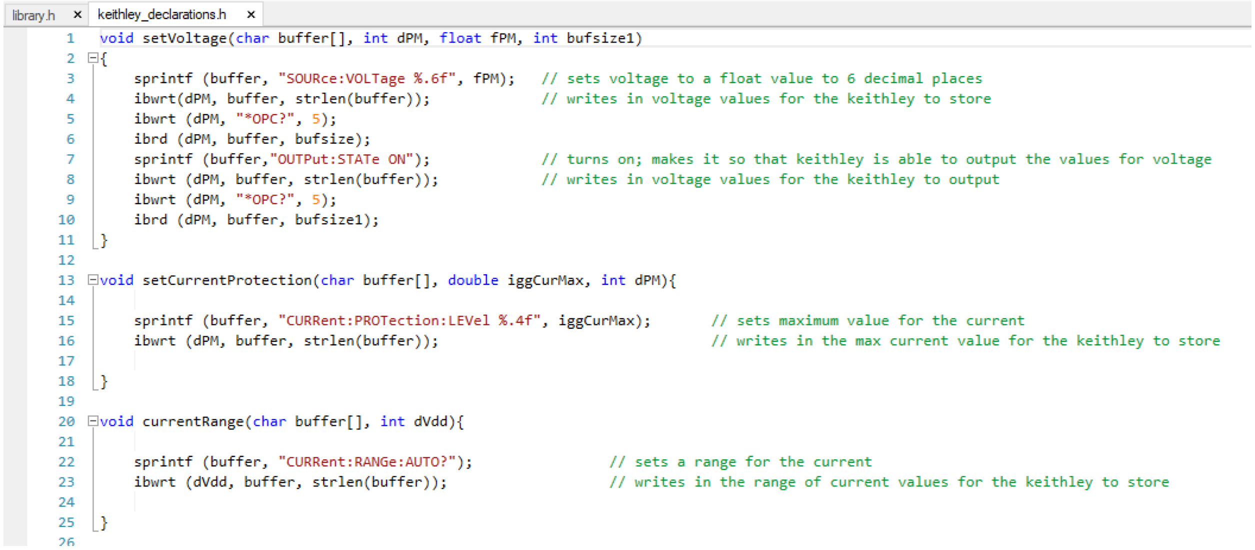
Ananya Gopalan
Engineering Internship
Formerly Custom MMIC, now Qorvo
7/3/2020

Current Work Progress

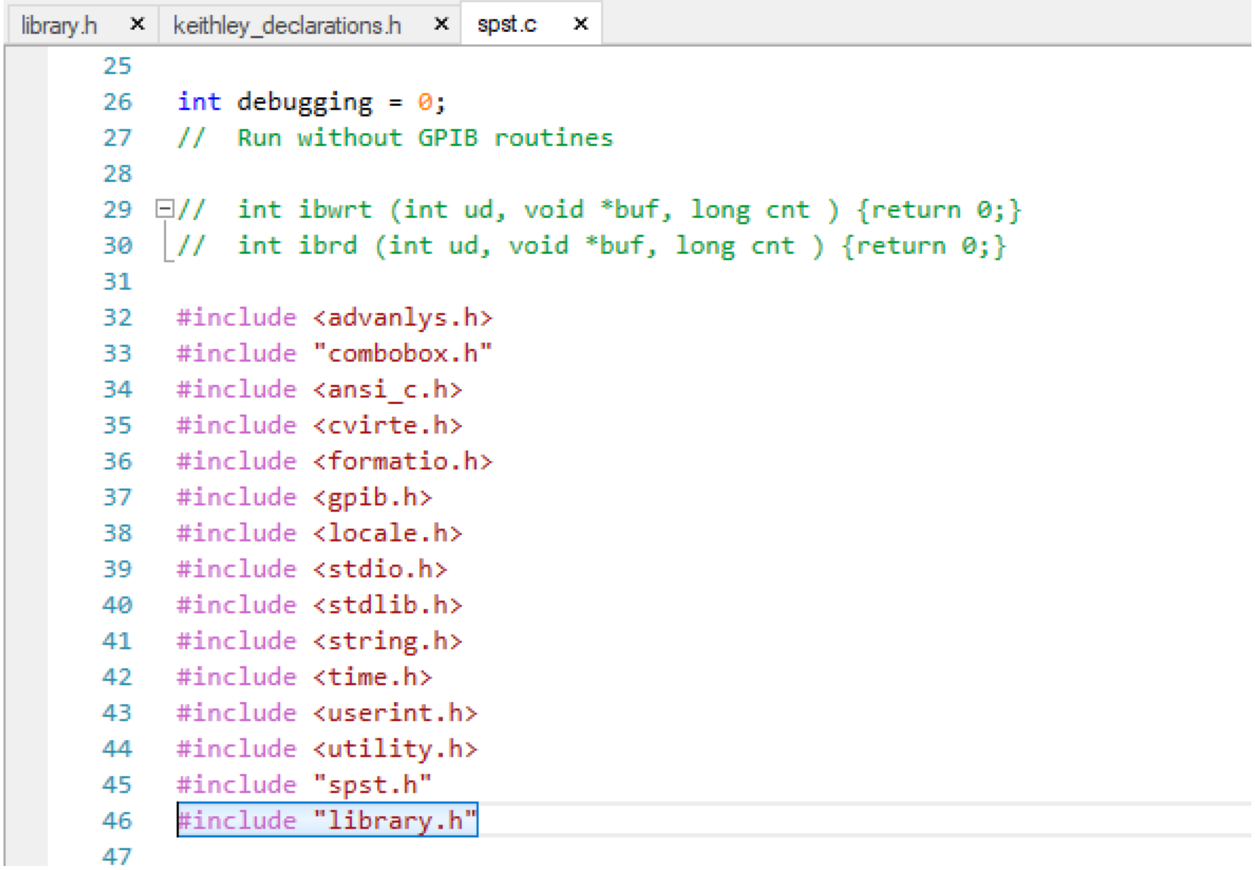
- I created a graphical user interface for the keithley machine in LabWindows/CVI
    - User can set the starting and ending voltage values
    - The program reads voltage and current values and increments the voltage by the user inputed step number.
    - The current and voltage values are printed to a text file which the user can save
- Created a library which is implemented into every program
    - This library encompassing the different functions used by each machine
    - A different header file was created for every machine so in the future, when programing a machine one can call upon the function instead of rewriting that block of code
        - For example, every universal code block for the keithley machine--such as reading the current and setting limits for the max current the machine allows--is included in the keithley header file.
        - Similarly, a different header file was created for the source and power meters, the exatron machine, and the AOR functionality.
    - When running or creating a program the user/programmer is easily able to download the git library repository, as the library is already implemented in every program
    - For a new program, the library is easily implemented through one line of code
    - Benefits: Makes the program more readable, organized, unified, succinct, easily accessible, and collaborative (as the code is more easily understood and written)
- Created a separate repository on GitHub for every program and the library implementations
    - Makes collaborations on programs/projects much easier
    - Gives the users the ability to restrict access to only specific personal (adds to the security of the project)
    - Shows the iterations of the project or code so a programmer can easily debug the program
    - Makes projects much more organized
- Implemented AOR functionality into a myriad of programs
    - Similar to the nice figure production program that Grace was working on last year, I implemented a similar algorithm to compute AOR calculations in other preexisting production programs.
        - These programs also use a rolling window to get rid of erroneous microchips and keep only the best of the group of microchips being tested in real time



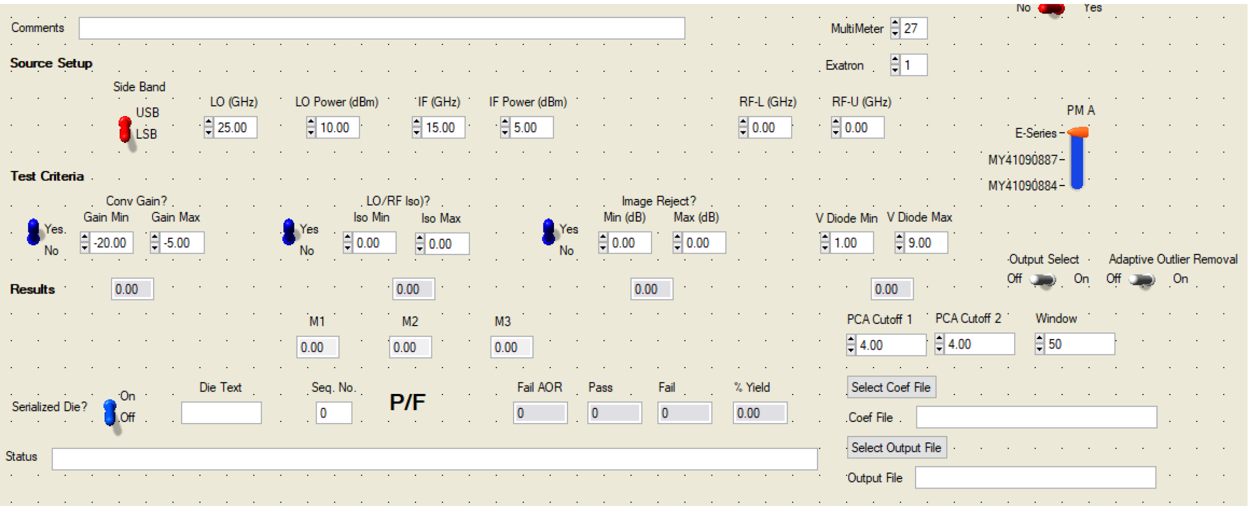
```
#include "keithley_declarations.h"      // keithley functions (for voltage and current)
#include "source.h"                     // source functions (for power and frequency)
#include "Normalize.h"                  // Ananlyze chip data (AOR)
#include "exatron.h"                    // exatron functions
#include "Power.h"                      // power functions
```



```
void setVoltage(char buffer[], int dPM, float fPM, int bufsize1)
{
    sprintf (buffer, "SOURce:VOLTage %.6f", fPM);   // sets voltage to a float value to 6 decimal places
    ibwrt(dPM, buffer, strlen(buffer));             // writes in voltage values for the keithley to store
    ibwrt (dPM, "*OPC?", 5);
    ibrd (dPM, buffer, bufsize);
    sprintf (buffer,"OUTPut:STATe ON");             // turns on; makes it so that keithley is able to output the values for voltage
    ibwrt (dPM, buffer, strlen(buffer));            // writes in voltage values for the keithley to output
    ibwrt (dPM, "*OPC?", 5);
    ibrd (dPM, buffer, bufsize1);
}

void setCurrentProtection(char buffer[], double iggCurMax, int dPM){

    sprintf (buffer, "CURRent:PROTection:LEVel %.4f", iggCurMax);       // sets maximum value for the current
    ibwrt (dPM, buffer, strlen(buffer));                                // writes in the max current value for the keithley to store

}

void currentRange(char buffer[], int dVdd){

    sprintf (buffer, "CURRent:RANGe:AUTO?");                // sets a range for the current
    ibwrt (dVdd, buffer, strlen(buffer));                   // writes in the range of current values for the keithley to store

}
```



```c

int debugging = 0;
//  Run without GPIB routines

//  int ibwrt (int ud, void *buf, long cnt ) {return 0;}
//  int ibrd (int ud, void *buf, long cnt ) {return 0;}

#include <advanlys.h>
#include "combobox.h"
#include <ansi_c.h>
#include <cvirte.h>
#include <formatio.h>
#include <gpib.h>
#include <locale.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <userint.h>
#include <utility.h>
#include "spst.h"
#include "library.h"

```



No Yes

Comments

MultiMeter 27

Exatron 1

Source Setup

Side Band USB LSB

LO (GHz) 25.00

LO Power (dBm) 10.00

IF (GHz) 15.00

IF Power (dBm) 5.00

RF-L (GHz) 0.00

RF-U (GHz) 0.00

PM A

E-Series

MY41090887

MY41090884

Test Criteria

Conv Gain? Yes No

Gain Min -20.00

Gain Max -5.00

LO/RF Iso)? Yes No

Iso Min 0.00

Iso Max 0.00

Image Reject? Yes No

Min (dB) 0.00

Max (dB) 0.00

V Diode Min 1.00

V Diode Max 9.00

Output Select Off On

Adaptive Outlier Removal Off On

Results 0.00 0.00 0.00 0.00

M1 0.00

M2 0.00

M3 0.00

PCA Cutoff 1 4.00

PCA Cutoff 2 4.00

Window 50

Serialized Die? On Off

Die Text

Seq. No. 0

P/F

Fail AOR 0

Pass 0

Fail 0

% Yield 0.00

Coef File

Output File

Status