Current-Voltage Curve Tracer

Texas Tech University

May 2017

Abstract

This paper describes a system which plots the leakage current with respect to voltage across the drain-source of MOSFETS, collector-emitter of BJTS and P-N junction of diodes. The system incorporates three subsystems comprised of LabVIEW as the control center for issuing commands and analyzing data, a PIC microcontroller for gathering data of a DUT, device under test, and MySQL for remote data storage and analyzation via TCP/IP protocol. The result of system execution yields a graphical display that describes the DUT’s leakage current as voltage is increased.

Table of Contents

Abstract.................................................................................................................................1

Table of Contents..................................................................................................................2

List of Figures.......................................................................................................................3

List of Tables.........................................................................................................................4

1. Introduction.......................................................................................................................5

2. Data Accusation System ..................................................................................................6

3. Data Processing System ……….....................................................................................13

4. Remote Data Storage System..........................................................................................16

5. Conclusion.......................................................................................................................19

6. Appendix A – References................................................................................................20

7. Appendix B – Notes………………………………………………………....…………21

List of Figures

Figure 1.1-System Data Flow………………………………………………………………5

Figure 2.1-Controller Section (DAS)… ………………………………...…………………6

Figure 2.2-Power Section (DAS)… …………………………………………………...…..7

Figure 2.3-Drain DAC Current Amplifier ……….…………………………………...……8

Figure 2.4- Main Function Code Flowchart ………………………...……………………11

Figure 2.5- Bit Shifting Function ………………………..………….……………………13

Figure 3.1- LabVIEW Data Acquisition …………..…………….……………………….14

Figure 3.2 - LabVIEW Data Sorting ………………...…………….……………………..14

Figure 3.3-LabVIEW SQL Transmission ……….……………………….………………15

Figure 4.1 – Remote Server Hosting ……………………………….……….……………16

Figure 4.2 – PHP Graph ……………………………………...………………..…………17

Figure 4.3 – HTML Display on RDSS ……………………………………….…………..18

List of Tables

Table I-Expected Bit Value…………………………………………………………….…10

1. Introduction

A system was designed to emulate the following scenario: to graphically represent, locally and remotely, the characteristic leakage current of a MOSFET, BJT, P-N diode with respect to an applied voltage when placed under test by the system. The system, typically called a current-voltage curve tracer, is constrained to testing voltages up to 50V and able to record leakage currents up to 10A. The overall system is comprised of three subsystems that work in tandem: The Data Accusation System (DAS), The Data Processing System (DPS), The Remote Data Storage System (RDSS). Figure 1.1 depicts the direction of flow for inputs and outputs of each system within the entire system.

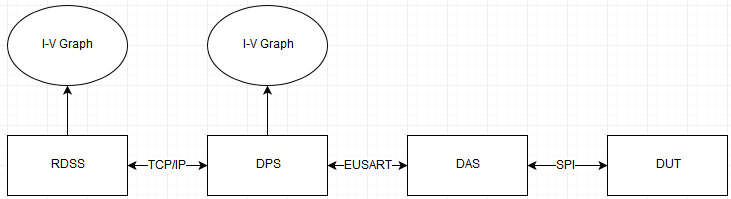


Figure 1.1-System Data Flow

2. Data Accusation System

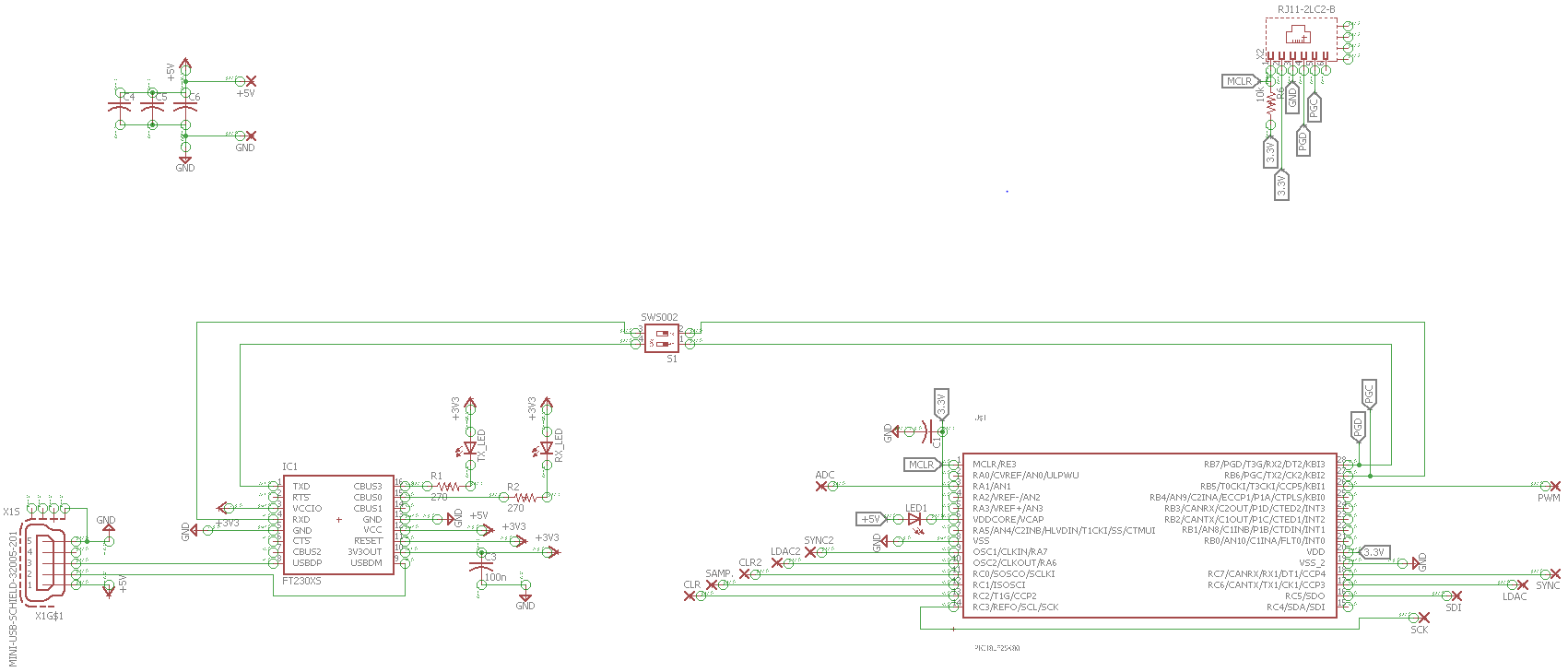
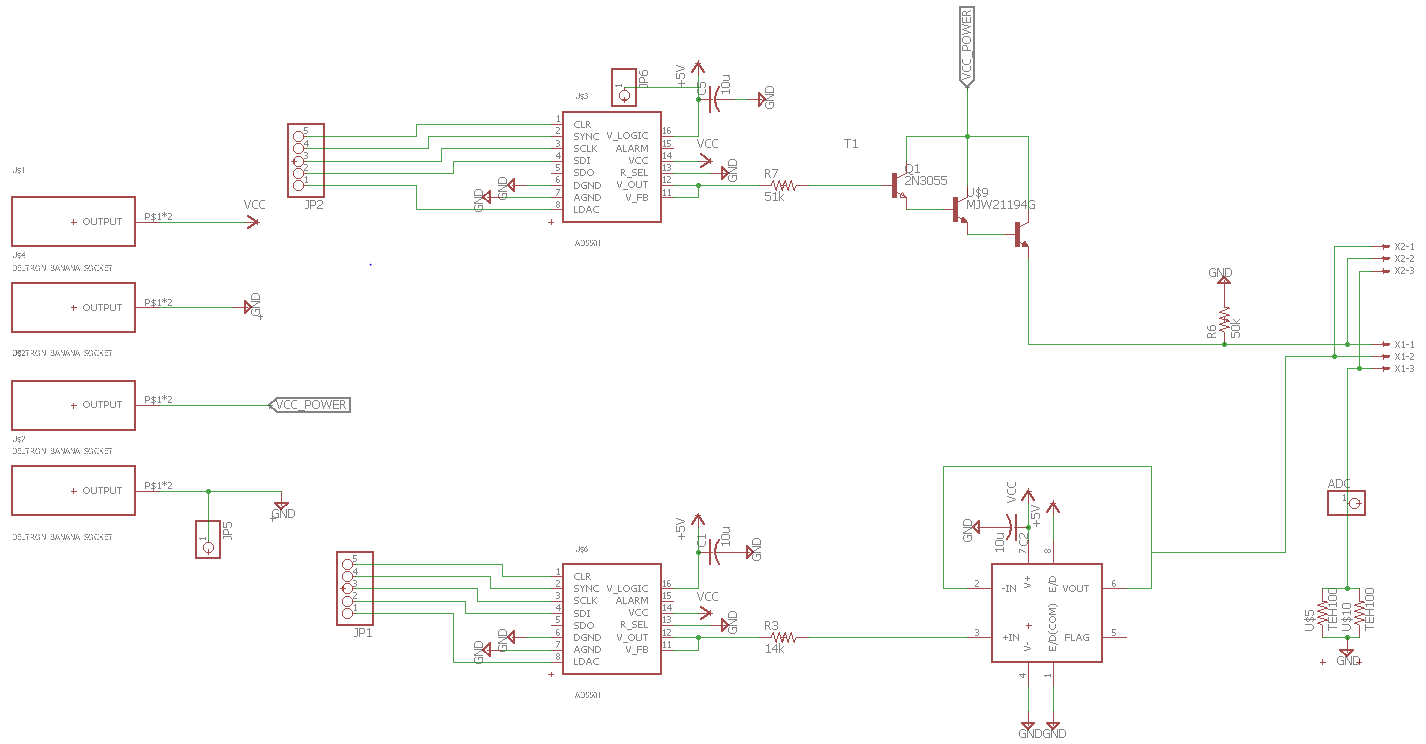
The circuit implemented to increase voltage while simultaneously measuring leakage current of a DUT is comprised of two separate entities that act as one circuit. The splitting of the DAS system is necessary to provide physical isolation between the low voltage controller section and the high voltage peripheral section of the system. Such isolation ensures that each section can operate within their necessary voltage requirements and still maintain electronic, as well as human, safety considerations. Figure 2.1 and Figure 2.2 display the implemented DAS system previously described.

Figure 2.1-Controller Section (DAS)

Figure 2.2-Power Section (DAS)

The circuit in Figure 2.1 transmits and receives binary data to and from a PC host running the LabVIEW DPS subsystem code via a USB Mini port. The FT230XS, IC1, handles the conversion of incoming USB differential signals and the outgoing microcontroller EUSART data stream in a bidirectional manner. Emulation of the PIC18LF25K80 as a detectable COM port and the PC as a EUART connection allows for both devices to communicate in their standard fashion without communication error or manual configuration of the bit stream on both devices. The datasheet of the FT230XS can be found in the Appendix [2].

The PIC18LF25K80 microcontroller, IC2, is the central component of the DAS operation. The datasheet of the PIC18LF25K80 can be found in the Appendix [1]. Controlling of peripheral devices and data collection is directed by the microcontroller through instruction of C code that is generated by MPLABX-IDE and programmed through an ICD3, In Circuit Debugger, that is attached to the RJ11 port. During programming the DPST switch, located between the FT230XS and PIC18LF25K80, is open circuited to ensure the programming data is isolated from the FT230XS. A schottky diode between the main power supply and the supply pins of the microcontroller, VDD, adds supply isolation during programming allowing for only the microcontroller to be powered by the ICD3 programmer.

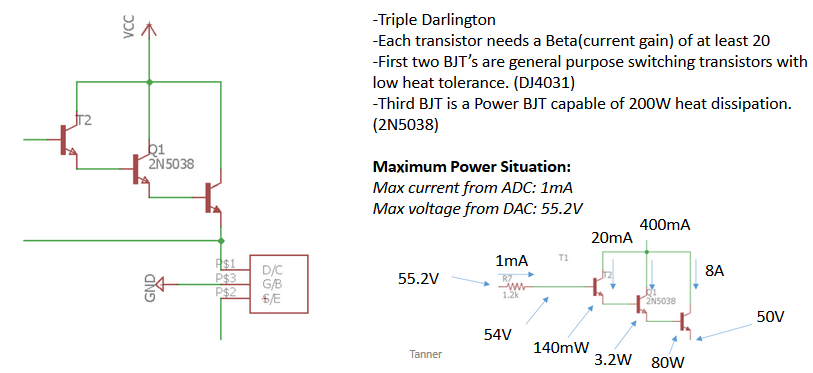
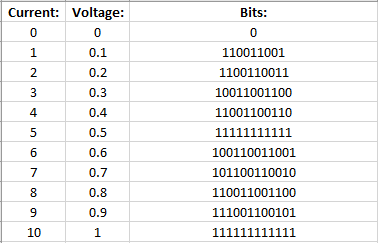
The circuit in Figure 2.2 describes the power section of the DAS. Male header pins supply signal connection between two external DACs and the respective pins on the microcontroller in Figure 2.1. A Darlington triple buffers the drain DAC to increase output current capability to the DUT. A secondary DAC controls the gate voltage of the DUT when testing a MOSFET or BJT and is not operated during diode testing. The output of the gate DAC is buffered by the OPA454 operational amplifier to increase output current supplied to the gate of DUT. The configuration of the drain DAC’s current amplifier was unnecessary for the gate DAC being that only 25mA is required for efficient operation. The datasheet of the OPA454 can be found in the Appendix [3]

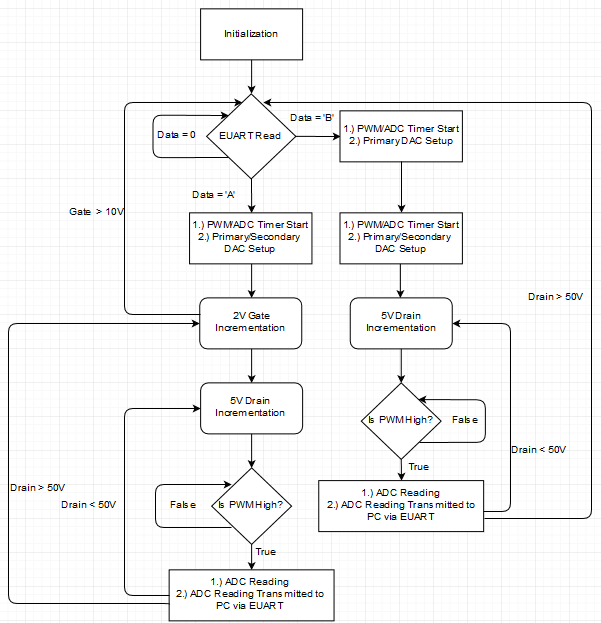
Figure 2.3-Drain DAC Current Amplifier

In reference to Figure 2.2, the AD5501 is a twelve bit Digital to Analog Converter, DAC, attached to the SPI OUT, CLK, LDAC, CLR, SYNC lines of the microcontroller. The datasheet for the AD5501 can be found in the Appendix [4]. The maximum sourcing current from the DAC is amplified by a Darlington triple giving the DUT the ability to sink ten amps if required. Figure 2.5 gives further analysis to the current amplifier. With a current gain of at least 20 and a current limiting resistor of 51kΩ, the DAC will not be required to source more than the maximum 1mA allowed for the part. The 2N5038, a power transistor, is capable of dissipating 200 watts of heat during operation, making external cooling unnecessary due to the system voltage/current requirements. The datasheet for the 2N5038 can be found in the Appendix [5].

The TEH100, a .1-ohm sense resistor, supplies feedback by means of a voltage to indirectly measure the leakage current of the DUT. The datasheet for the TEH100 can be found in the Appendix [6]. The scaling of the voltage feedback ranges from 0 to 5V when 10A are being passed by the DUT. The feedback voltage is captured by the microcontroller’s 12-bit ADC providing a resolution of 4095 steps or 1.2 mV per step. Low thermal drift and high heat dissipation abilities will ensure accurate feedback results to the microcontroller. Table I describes the expected data bits returned for a specific voltage measured.

Table I-Expected Bit Values

The 50V supply voltage is handled by two separate power supplies. One 50V power supply directly, and solely, powers the Darlington triple which directly supplies the DUT. The second 50V supply provides all ICs with 50V requirements independent of the DUT’s 50V supply. The necessity to split the 50V supply stems from the voltage drop occurring across DUT when testing occurs. If the supplies are unified and said voltage drop occurs, all ICs that require 50V are starved. Such destabilization of the supply can cause massive hardware error and erroneous operation.

Figure 2.4-Main Function Code Flowchart

In Figure 2.3, the current iteration of the main function code written for the microcontroller is displayed. When the microcontroller receives a byte on the EUART line, it is compared between two cases. Case “A” executes an algorithm specific to MOSFETs and BJTs, while case “B” executes an algorithm specific to diodes.

In the case of an ASCII “A” being received the microcontroller, initialization of the PWM and ADC timers occurs; then, writing of the control bits to the gate and drain DACs proceeds. Nested for loops then implement an algorithm designed to sweep a range of gate voltages while also sweeping a range of drain voltages. The gate loop begins at 0V and increments by 2V until reaching 10V. During each gate iteration, the drain loop begins at 0V and increments by 5V until reaching 50V. Each loops count value directly controls an external DAC that tracks and updates accordingly via SPI communication synchronized by a PWM timer.

In the case of an ASCII “B” being received the microcontroller, initialization of the PWM and ADC timers occurs; then, writing of the control bits to drain DAC proceeds. A modified version of the algorithm described in case “A” then begins to execute in a single for loop. The loop begins at 0V and increments by 1V until reaching 50V. Like case “A”, the count value directly controls an external DAC that tracks and updates accordingly via SPI communication that is synchronized by a PWM timer. The requirement of a gate voltage is forgone being that a diode is under test and does not require gate voltage to operate. Incrimination is also scaled down to better improve resolution of the data collected.

The PWM timer applies a 30% duty cycle to the transmission of SPI commands and synchronizes all outgoing signals to 200Hz. When the PWM timer count is reset to 0, SPI transmissions set the external DACs to the respective loop voltage. The drain DAC is then reset to 0V once the PWM timer’s value reaches the 30% of the count value. During the PWM process ADC readings are synchronized by monitoring the PWM count value. Once the count value reaches 15% of the overall value the ADC samples the external voltage, the sampled data is transmitted back to the host computer via EUART.

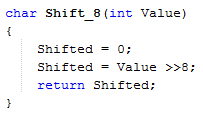
Data written on both EUSART and SPI lines are broken up into MSB, Most Significant Byte, and LSB, Least Significant Byte. Two transmissions are used when data larger than eight bits is required to be sent out from the microcontroller. Figure 2.4 demonstrates the splitting of a twenty-four-bit integer value into two eight-bit char variable. The last eight bits of the integer value are allowed to be truncated being that under no circumstance does the microcontroller need to send out data larger than sixteen bits.

Figure 2.5-Bit Shifting Function

3. Data Processing System

The Data Processing Subsystem receives the 12-bit data captured by the Data Acquisition Subsystem and displays it graphically on the host machine in real time. Figure 3.1 displays the LabVIEW code responsible for acquiring the incoming data. Figure 3.2 displays the LabVIEW code responsible for sorting and combining multiple arrays of data into one graph. Finally Figure 3.3 displays the LabVIEW code responsible for transmitting the results of the LabVIEW code in Figure 3.2 to the RDSS.

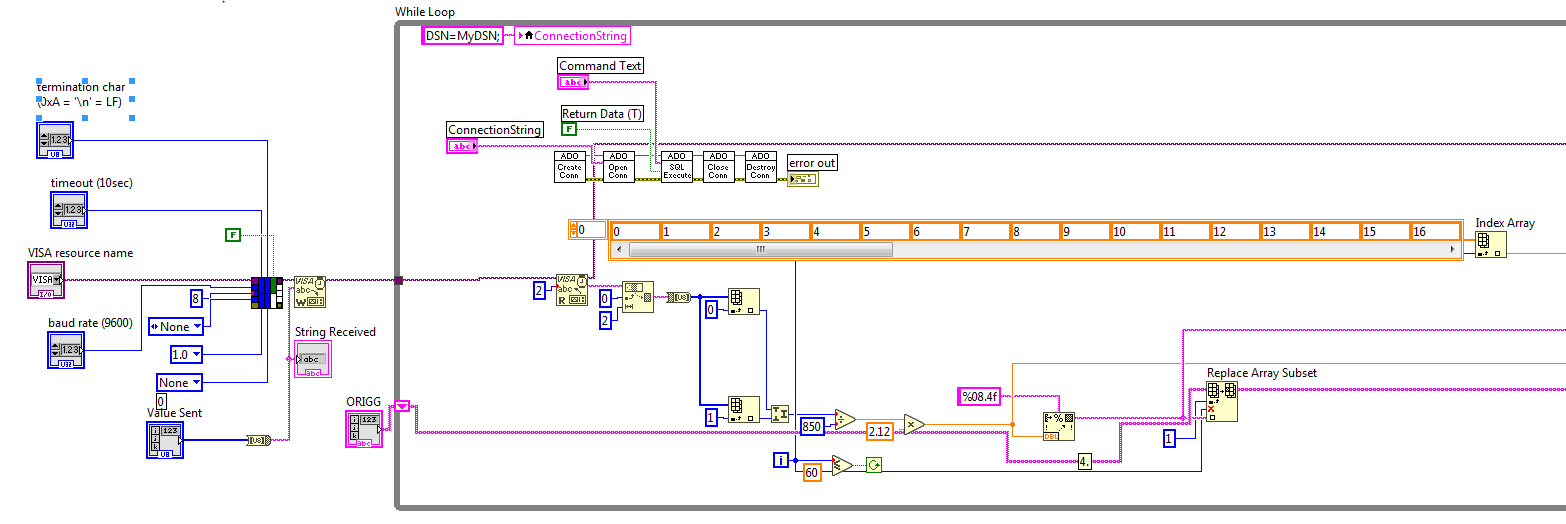
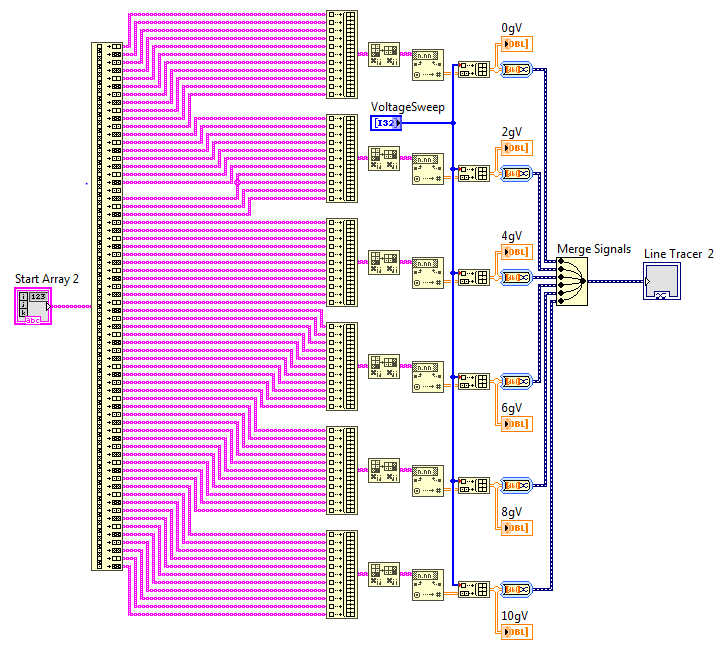
Figure 3.1- LabVIEW Data Acquisition [7]

Figure 3.2 - LabVIEW Data Sorting [7]

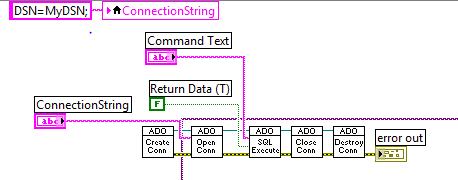
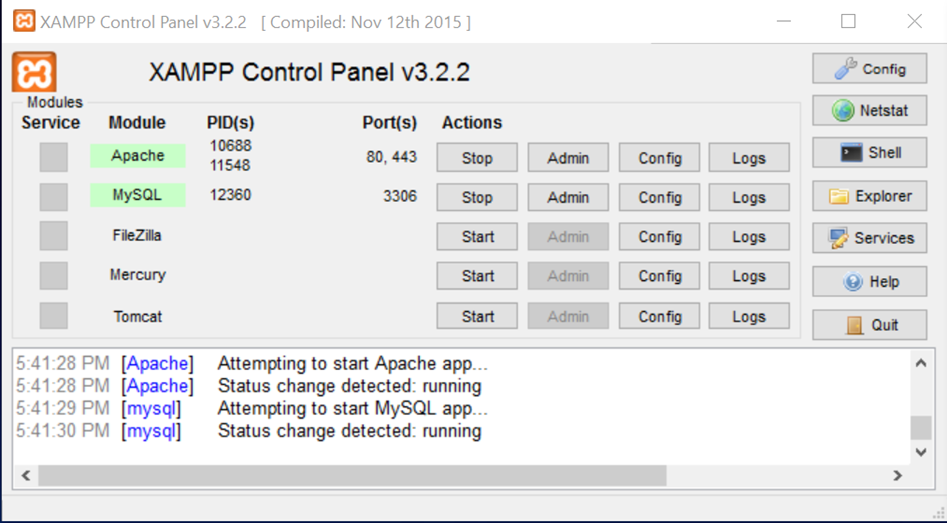


Figure 3.3-LabVIEW SQL Transmission [7]

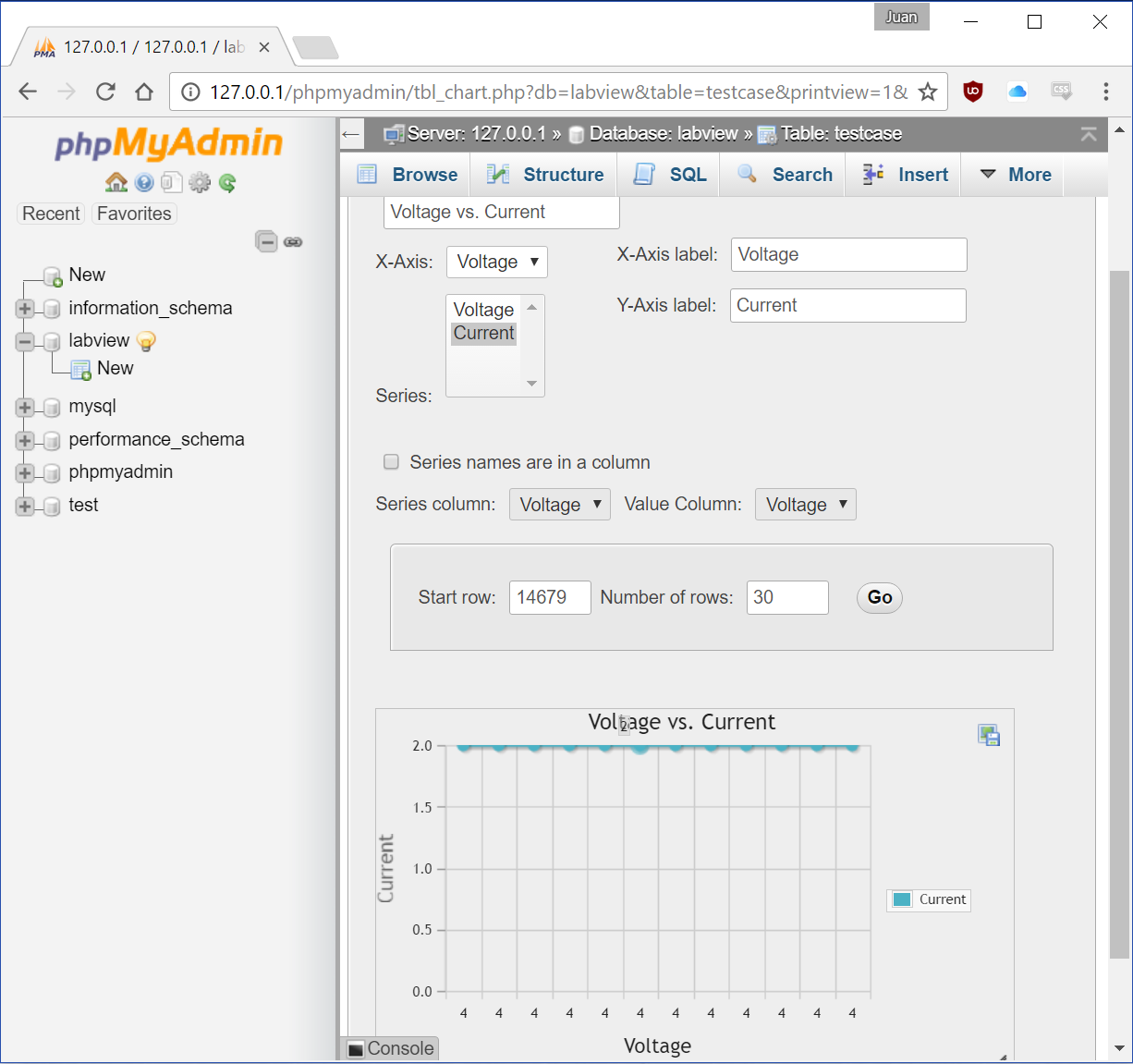
When first initialized, the DPS creates a VISA port in which data is sent and received from the microcontroller. The DPS writes an ASCII “A” or “B” depending on the required operating procedure to begin the data acquisition process on the microcontroller. As bytes are received they are type-casted from string data types into integer data types. Once two integers have been received, they are combined into one integer value with MSB and LSB being respectively placed during the combination step. The combined integer value is then divided by 850, the resolution of the ADC scaled to 5V, to return the integers to the analog values measured. The analog values are then multiplied by 2.12 to recover the indirectly measured current by the ADC. The decimal values are then sent to a LabVIEW module, depicted in Figure 3.2, that automatically sorts the data into multiple arrays. The sorting procedure allows the arrays of data to be categorized into individual lines to be graphed. In the case of a MOSFET or BJT, this is a necessity to recognize which sets of data identify with a certain gate voltage. The Remote Data Storage System works in tandem with the DPS to transmit and process the incoming data to a remote SQL database, Figure 3.3 describes the SQL transmission process.

4. Remote Data Storage System

When initialized an SQL connection is established between a remote server and the computer that is hosting the LabVIEW module. As bytes are received from the COM port, a copy is sent to the SQL interface for transmission. Each byte received is sent out from the RDSS via a TCP connection contained within the SQL module of LabVIEW. Once the final transmission has occurred, the SQL connection is closed and RDSS processing continues on a remote server. Figure 4.2 displays the server hosting software.

Figure 4.1 – Remote Server Hosting [8]

XAMPP is used to host the PHP backend and HTML front end that together act as the remote processing server for the RDSS. The XAMPP provides the LabVIEW module with an IP, essentially an address, and port in which to transmit data using TCP standards. The PHP backend, when receiving data, begins to store the byte information into an array with MSB and LSB in their respective configuration. PHP’s mathematical abilities allow for the 12-bit data to be converted into the indirectly measured current in the same algorithm as the DPS; In which; the data obtained by the DAS and received by the DPS is effectively reconstructed by the RDSS. In Figure 4.3, further PHP coding takes the decimal data and plots it on a graph that will be utilized by the HTML frontend.

Figure 4.2 – PHP Graph [9]

The HTML front end handles all the graphical elements with no mathematical or data processing operations taking place in the code. The PHP generated graph is imported by the HTML code as an object which is then displayed to the viewer. The exported graph generated by the backend is of an image format; therefore, importation and updating of the graph to the end user is made simple for the HTML front end by simply refreshing the exported graph as needed, updating the website image with a new image instead of actual processing occurring on the frontend. Figure 4.4 shows an example of the HTML frontend.

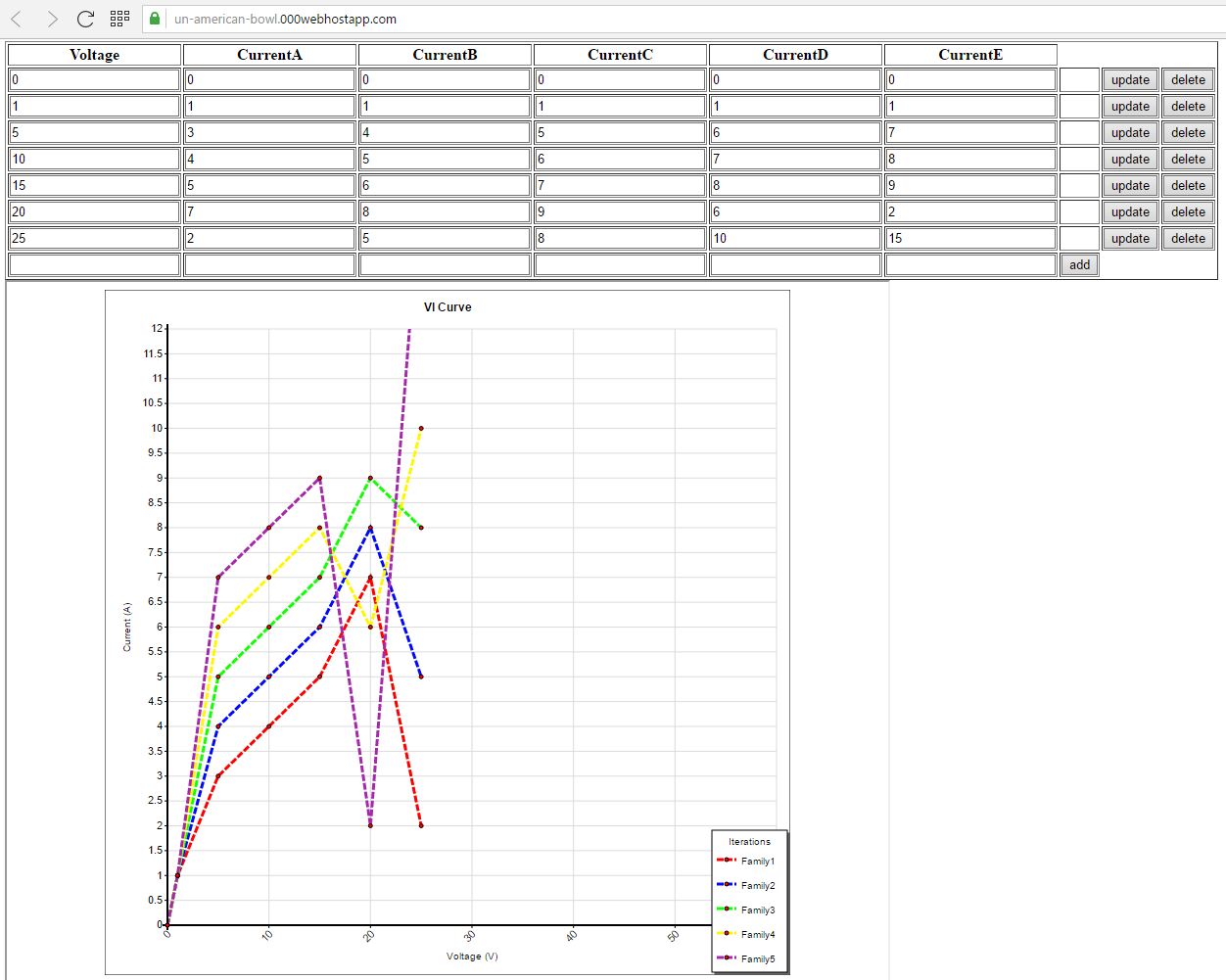


Figure 4.3-HTML Display on RDSS [9]

5. Conclusion

The implementation of a system in which the leakage current, with respect to an increasing voltage, is measured through the drain-source of a MOSFET, collector-emitter of a BJT and anode-cathode of a P-N diode has been successful. The three subsystems: DAS, DPS and RDSS all work in tandem to obtain, process and represent data to the end user in a graphical format both locally and remotely. The finalized system meets the project constraints of obtaining 50V during testing and measuring up to 10 amps of leakage current.

Appendix A:

References:

[1] PIC18LF25K80

<http://www.mouser.com/ds/2/268/01308d-33524.pdf>

[2] FT230XS

<http://www.mouser.com/ds/2/163/DS_FT230X-5395.pdf>

[3] OPA454

<http://www.ti.com/lit/ds/symlink/opa454.pdf>

[4] AD5501

<http://www.mouser.com/ds/2/609/AD5501-877117.pdf>

[5] 2N5038

<http://www.mouser.com/ds/2/308/2N5038-D-105429.pdf>

[6] TEH100

<http://www.mouser.com/Search/ProductDetail.aspx?R=TEH100MR100JEvirtualkey58810000virtualkey588-TEH100MR100JE>

Appendix B:

Safety:

During experimentation, the presence of high voltages and currents requires that precautions be taken in order to prevent electrocution of electronics and/or individuals near the device. When the device has +50 volts present, no contact must be made with the hardware, nor must metal be near the surface of the hardware to ensure arching associated with high voltages does not occur. The presence of high currents compounds the need of isolation away from the device when active. Appropriate grounding attire should be worn to avoid death by cardiac arrest if electrocution was to occur. The surrounding environment should have signs to warn other individuals to remain clear of the area and a dividing barrier should be present to set a perimeter.

Ethics:

When designing a device, one must not cut corners or be reckless with decision making as to save monetary value or time. When dealing with a potentially dangerous device, it is vital that everything be done correctly and to standard to avoid and accident from occurring. Academic honesty must also be applied to engineering ethics, as the presenting of one’s ideas that did not originate from the presenter constitutes as stealing. It is not only ethically incorrect to plagiarize in any engineering process it is morally wrong to do such.

Final Budget:

