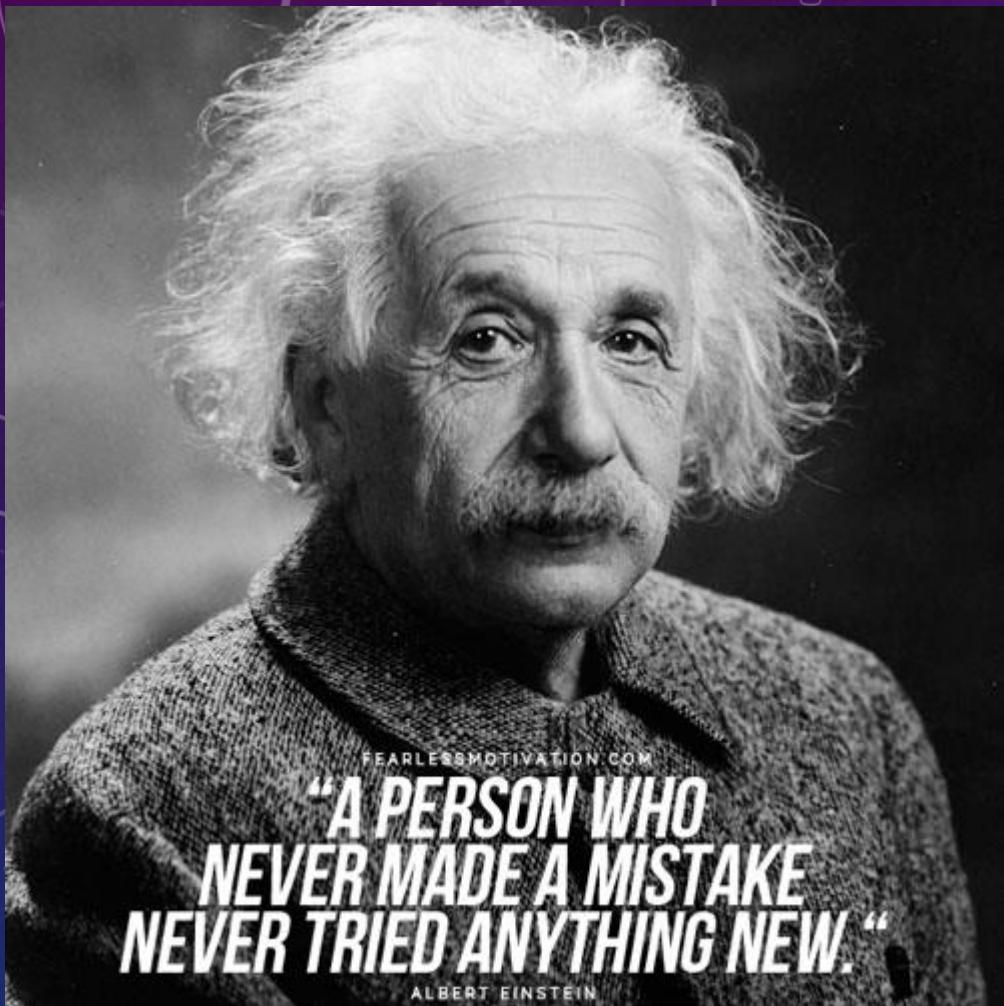


NPN-NMOS TRANSISTOR TRACER

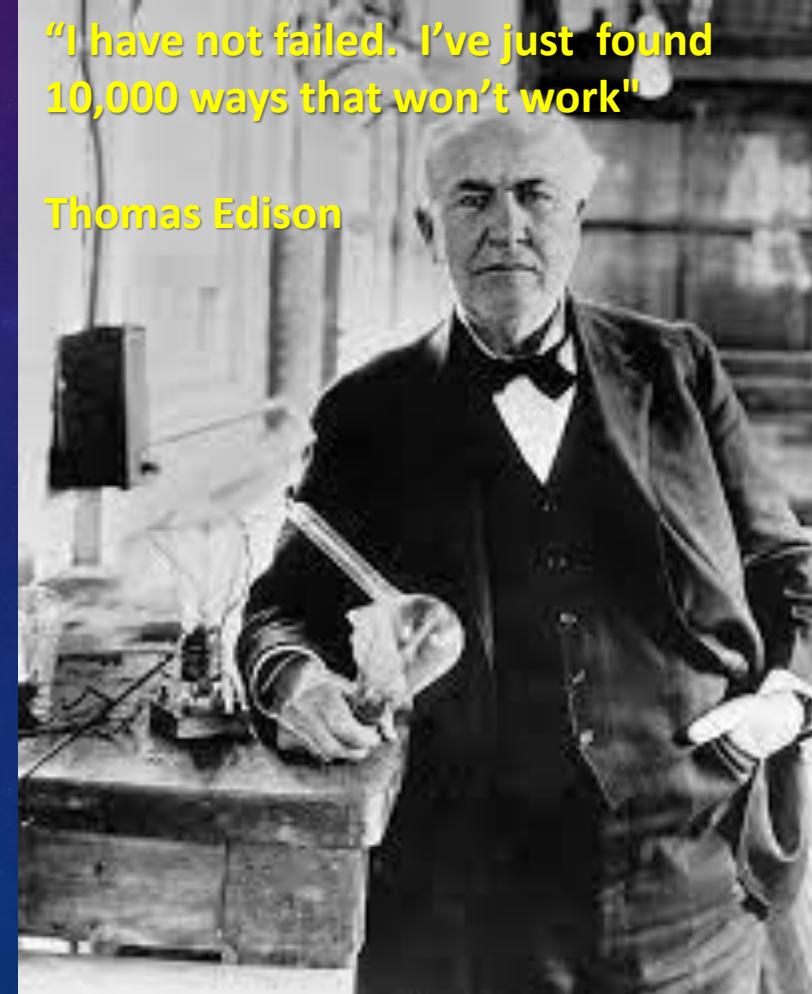
DAVE VE3OOI
MARCH 2023



This Project is a
Textbook Example
of Making Mistakes

**"I have not failed. I've just found
10,000 ways that won't work"**

Thomas Edison



THE PROBLEM

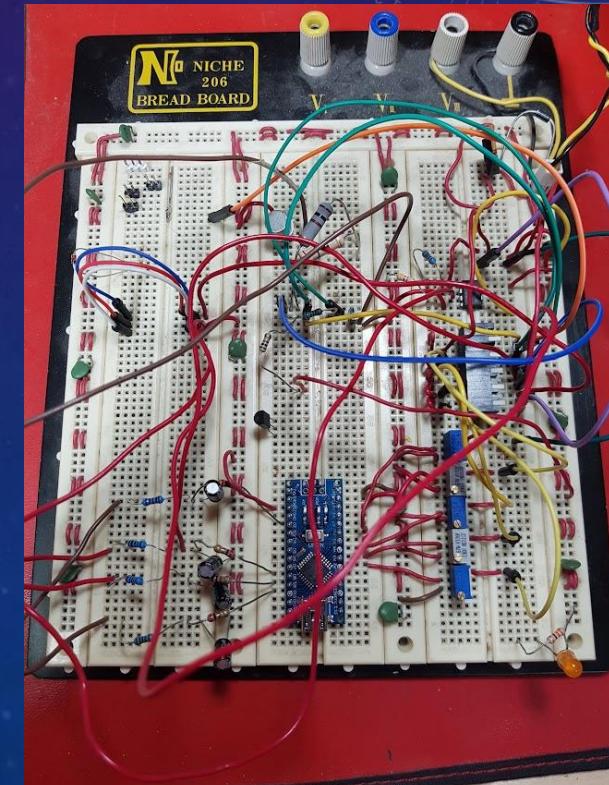
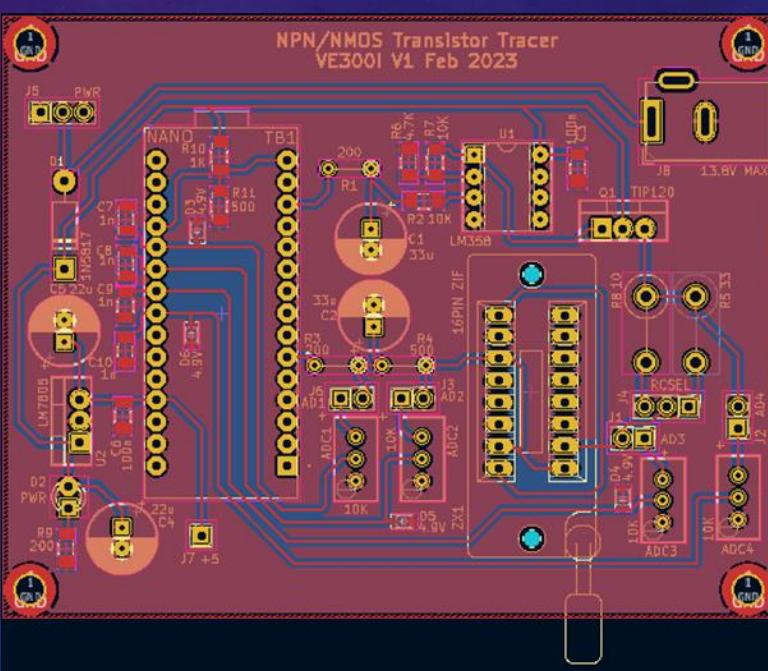
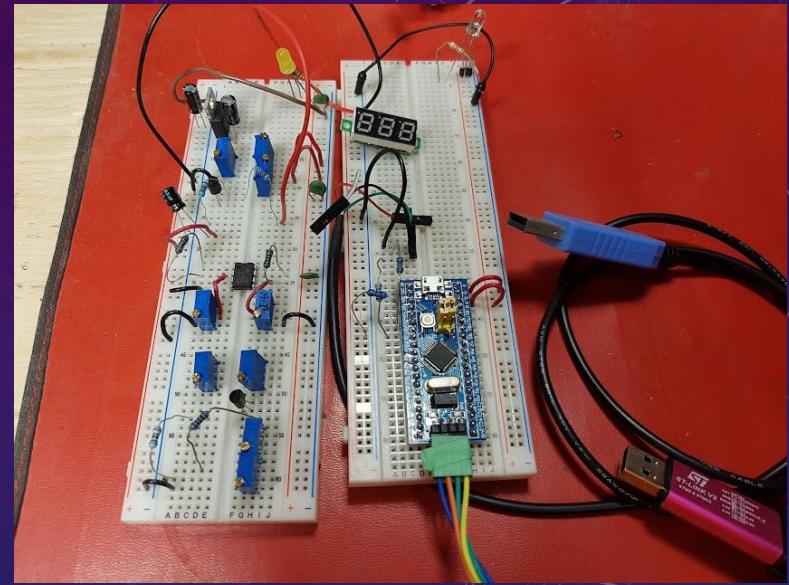
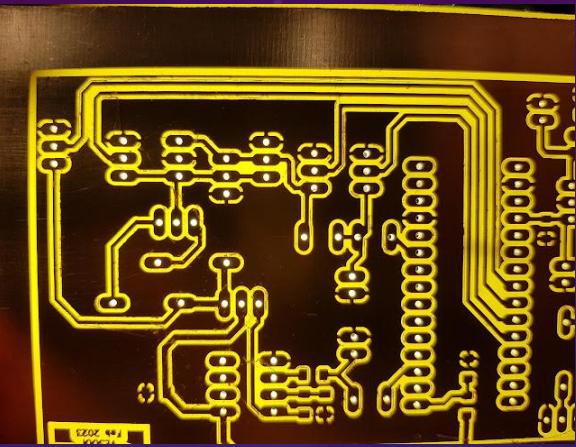
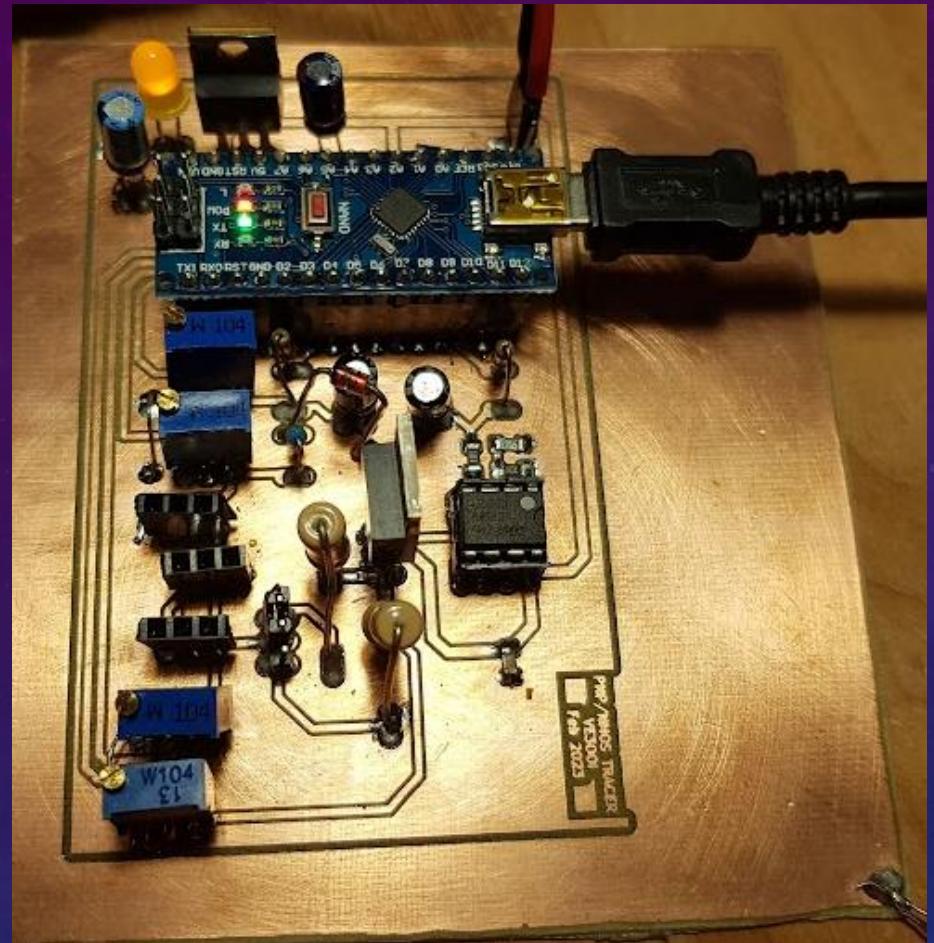
1. You have a load that needs 350mA of current to work properly.
 - ✓ What transistor should you use?
 - ✓ What base resistor do you need to drive 100mA?
2. You need to have a quiescent current of 200mA through a MOSFET.
 - ✓ What bias do you apply to the MOSFET to get 200mA?
3. You need to have a reverse voltage protection diode with lowest voltage drop.
 - ✓ Which diode do you use from a bag of diodes you ordered from Digikey.
4. You are building a Class A RF amplifier and you need to set the bias point of the transistor so that it 100% linear.
 - ✓ What is the optimum bias point to use?
 - ✓ What base current do you need to get the desired output current?

AGENDA

1. The Transistor Tracer Hardware Version 1
2. Python Graphing Program
3. Summary of Various Transistor Curves (Not a Tutorial)
4. Data Sheet Comparison with Transistor Tracer Graphs
 - ✓ Diodes
 - ✓ NPN Transistors
 - ✓ Enhancement Mode MOSFETS
5. Changes in Version 2

* This is not a Transistor Tutorial. Its assumed you understand how a BJT and MOSFET operates and terms (e.g., V_{be} , V_{gs} , V_{ce} , V_{ds} , etc). Only a short summary if given

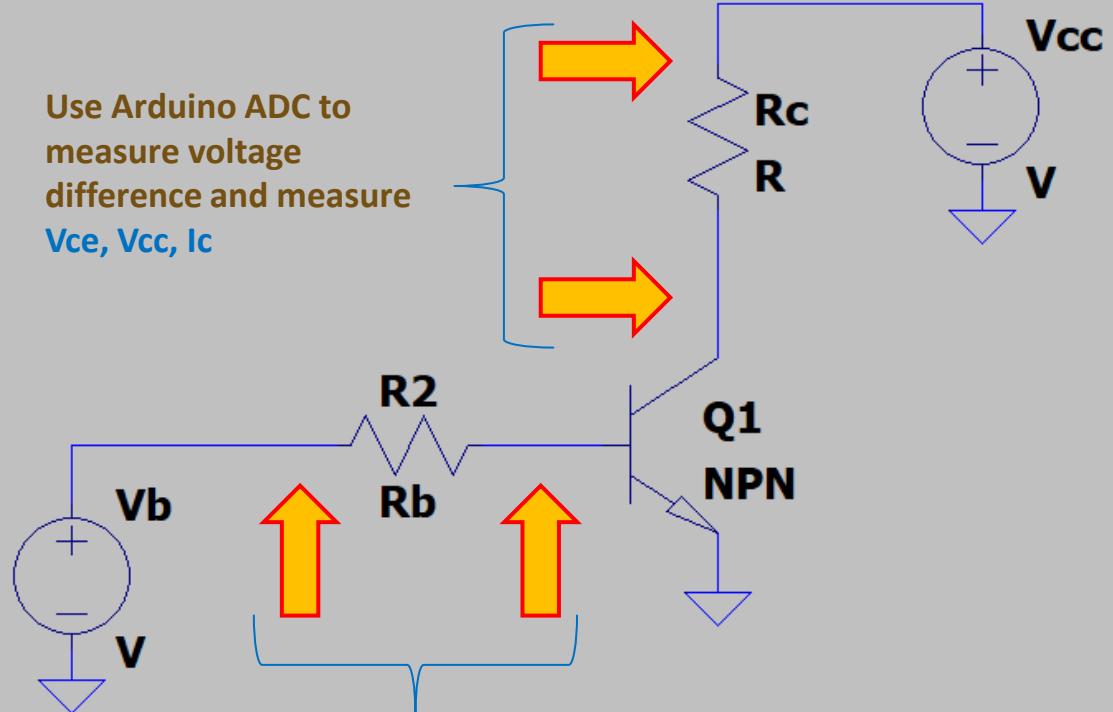
TRANSISTOR TRACER V1 EVOLUTION



Drop PNP and PCHAN MOS Functionality

HOW IT WORKS

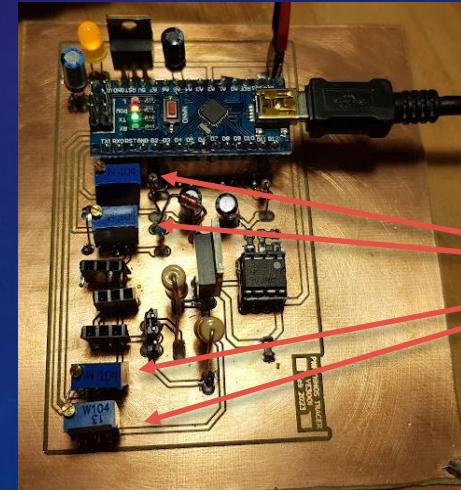
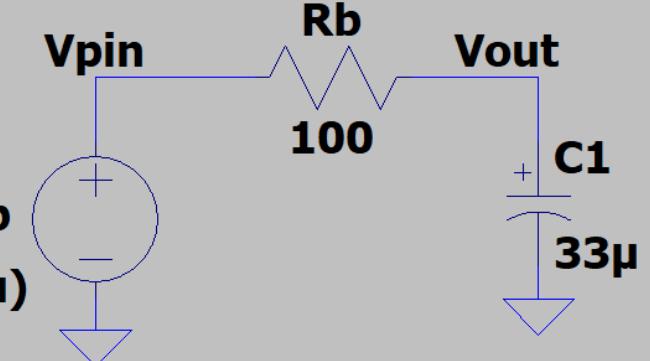
Use Arduino ADC to measure voltage difference and measure V_{ce} , V_{cc} , I_c



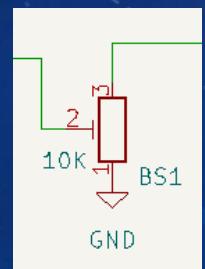
Use Arduino ADC to measure voltage difference and measure V_{be} , I_b , V_b

Use Arduino PWM and a LPF to generate a DC voltage

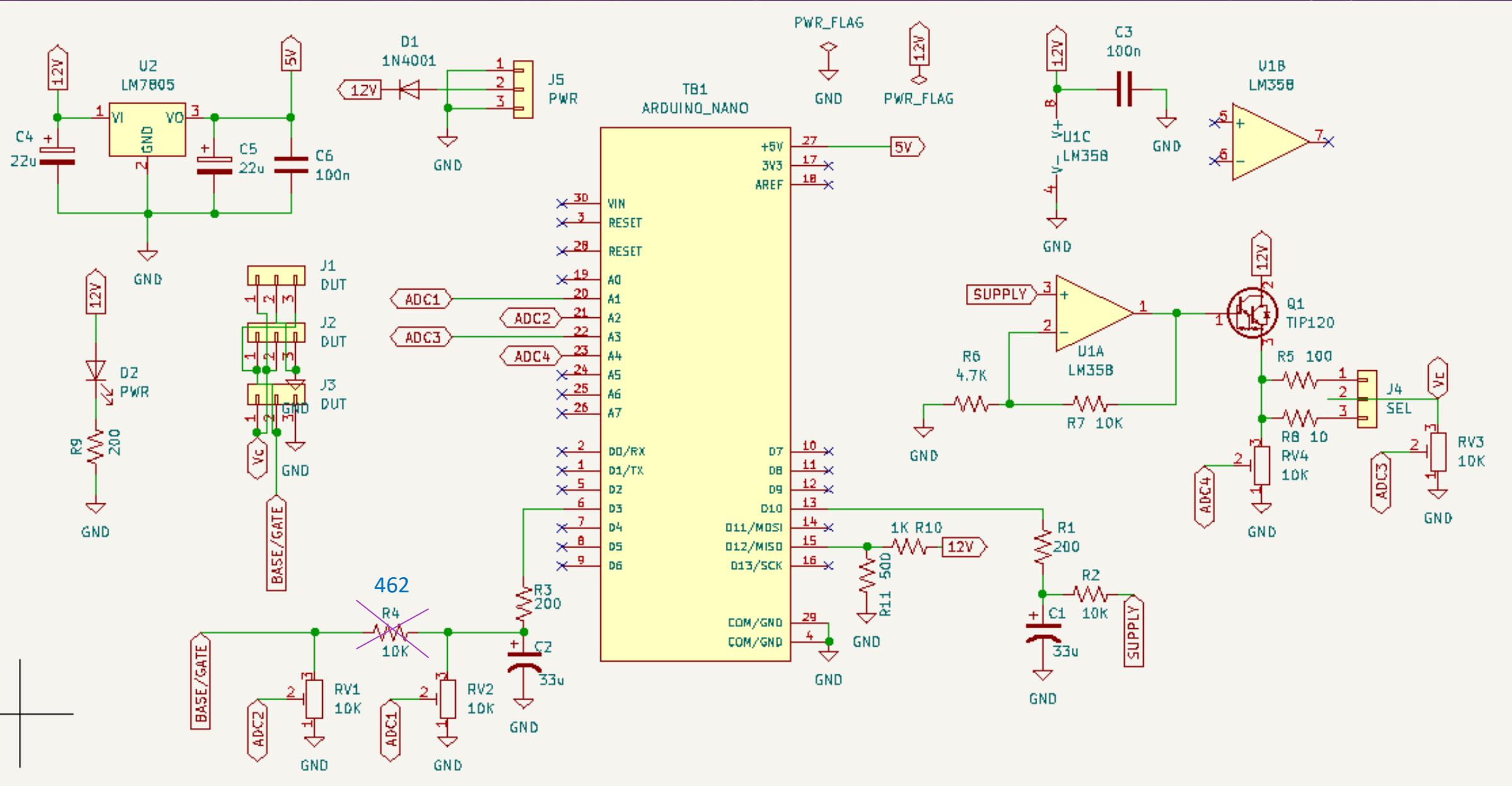
PULSE(0 5 0 1u 1u 16u 32u)



Pots uses as a resistive divider to keep ADC voltages below 5V

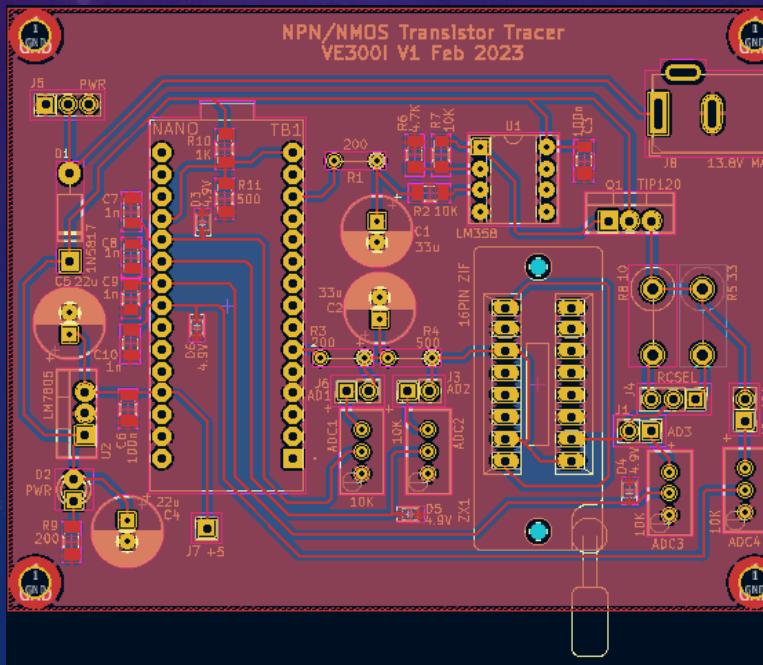
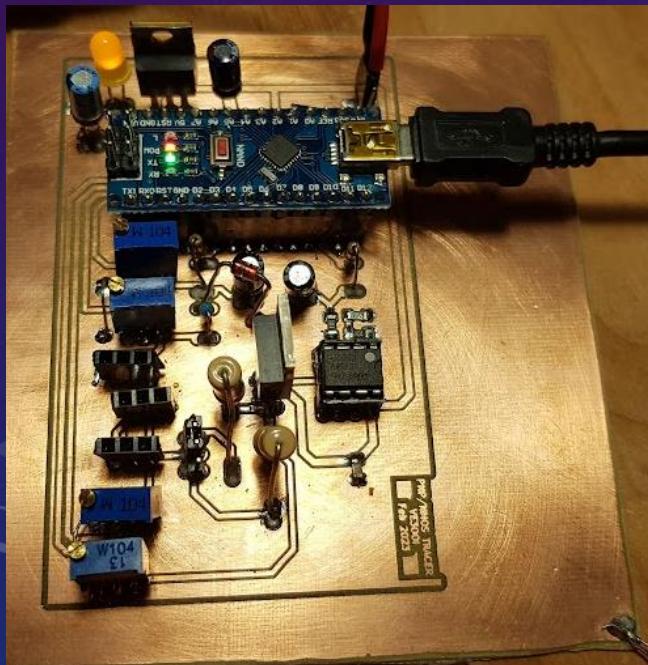
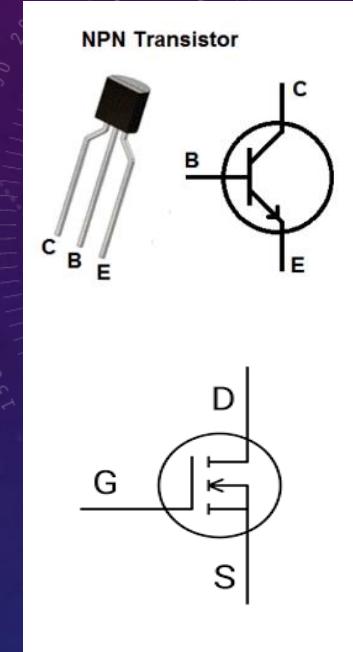


SCHEMATIC



VERSION 1: SPEEDS AND FEEDS

- ✓ **NPN** BJTs with EBC, BEC, BCE
- ✓ **NMOS** (N Channel Enhancement Mode MOSFET) with GDS, GSD, DGS
- ✓ Vcc 14V, Max Collector/Drain Voltage: 11@930mA
- ✓ Max Base/Gate voltage: 4.3@7mA

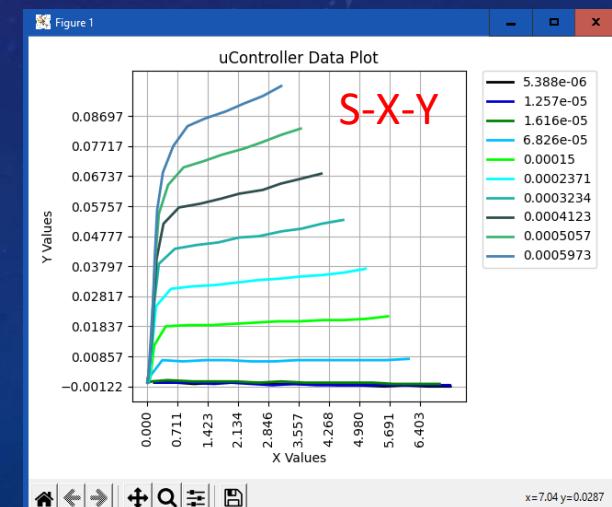
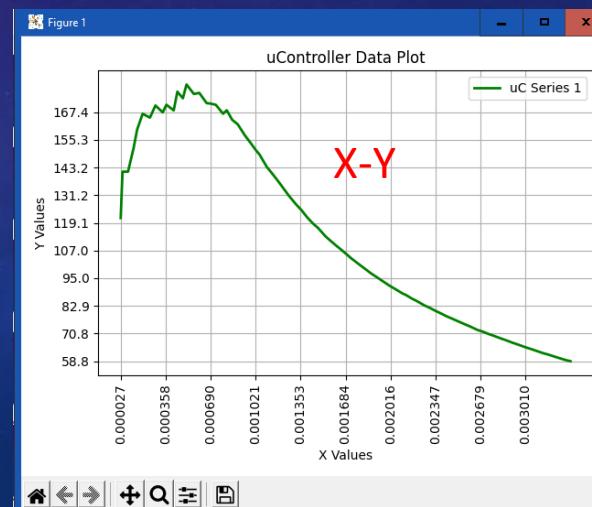
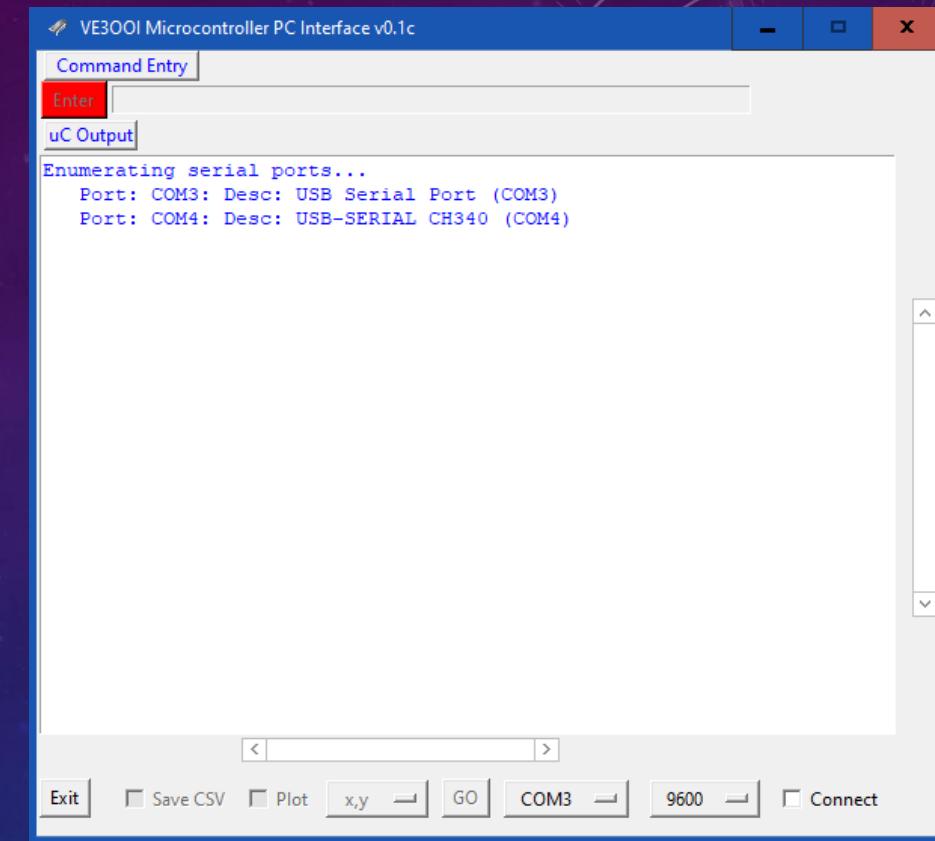


PC INTERFACE – PYTHON MICROCONTROLLER CONSOLE

MICROCONTROLLER CONSOLE

- Data acquisition program written in python for PC
 - ✓ https://github.com/drajnauth/uC_Console
 - ✓ Need to install matplotlib, numpy & serial to use Python script
 - ✓ Can run .exe file. Rename .ex_ to .exe 
- Used to acquire and plot data from a microcontroller
- Data needs to be written to serial console in a specific format
 - Data to be captured has "#" at start of line
 - Finish data set with a "|"
 - Create a new series with a "="
- Currently supports
 - Plot after all data received: x-y plots, s-x-y series
 - Plot as data arrives (live plot): y, x-y, s-x-y

YOU MUST PROGRAM uC TO SEND DATA



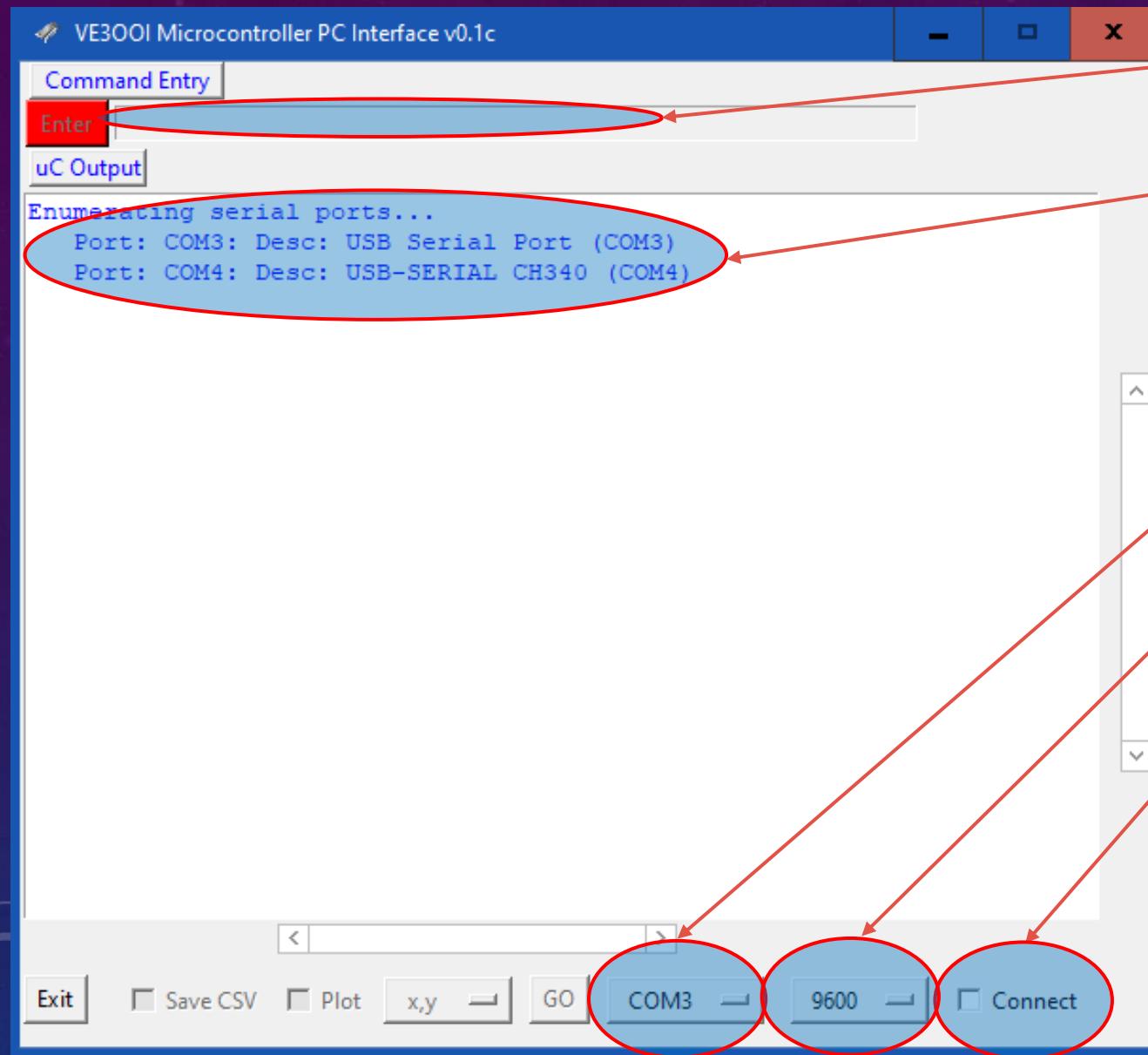
RUNNING THE EXE FILE

 Shack_Icon.png	3/30/2023 2:00 PM	PNG File	165 KB
 uC_Console_TKv0.1c.ex_	3/30/2023 2:22 PM	EX_File	31,015 KB
 uC_Console_TKv0.1c.py	3/30/2023 2:43 PM	Python File	52 KB
 uC_Console_TKv0.1c.pyw	3/30/2023 2:43 PM	Python File (no co...)	52 KB



 uC_Console_TKv0.1c.exe	3/30/2023 2:21 PM	Application	31,015 KB
--	-------------------	-------------	-----------

MICROCONTROLLER CONSOLE: CONNECTION



Enter command to be sent to uC
(same as Arduino Serial Console)

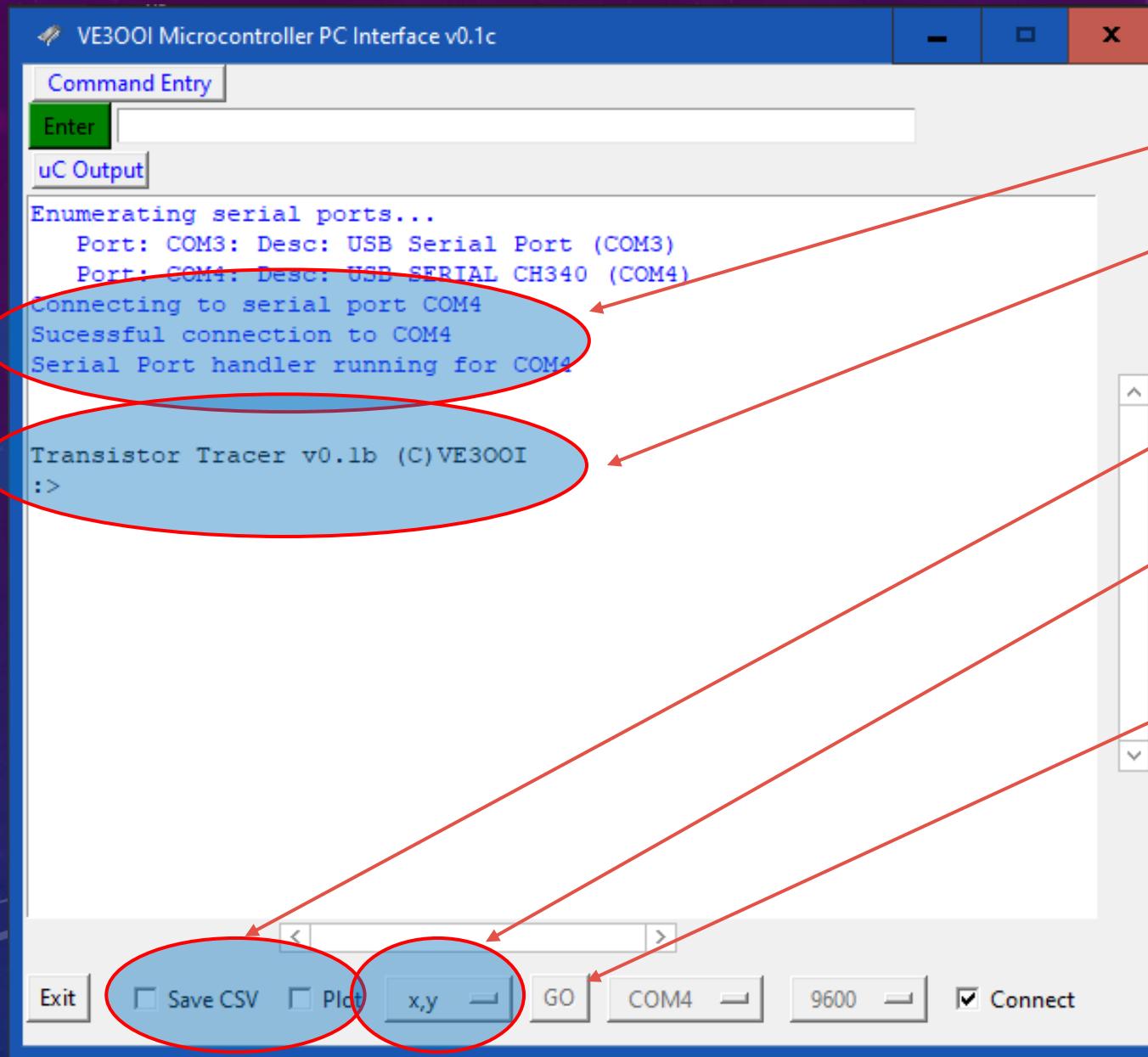
Serial Ports available to connect to

Select com port from drop down

Select baud rate from drop down box

Select the “connect” checkbox to
connect to the uC

MICROCONTROLLER CONSOLE: USAGE



Connection successful message

Text from uC is black. Text from program is blue

Select if you want to save data to a csv file or plot the data

Select the type of plot or the data series to save to csv file

When data has been successfully captures, the "GO" button will be enabled and you can press "GO" to generate the plot or save the csv file

HOW IT WORKS

1. The Microcontroller must have a CLI
2. The uC must append "#" to numbers and send other control characters

```
# // x,y numbers  
| // end control characters  
= // switch control character  
// i.e., new xy plot)
```

```
:> tni32  
Rc = 32.00  
:> BJT Input Sweep  
Sweeping Vb 0.00 V to 4.60 V  
Vc/Vd: 4.6000 Vb/Vg 0.0000 - 4.60 inc 0.23  
#0.0000000000000000,0.000000000000  
#0.18862840652465820312,0.000000000000  
#0.43115062713623046875,0.000000000000  
#0.66019940376281738281,0.006315693378  
#0.71409320831298828125,0.044209775924  
#0.74104013442993164062,0.077472376823  
#0.75451364517211914062,0.093472108840  
#0.78146052360534667968,0.103577196598  
#0.78146052360534667968,0.111156010627  
#0.79493398666381835937,0.118313789367  
#0.80840740203857421875,0.124629485607  
#0.80840740203857421875,0.130103087425  
#0.82188091278076171875,0.135576665401  
#0.82188091278076171875,0.140208148956  
#0.83535432815551757812,0.145260715484  
#0.83535432815551757812,0.149471163749  
#0.83535432815551757812,0.153260574340  
#0.84882774353027343750,0.157049989700  
#0.84882774353027343750,0.160418343544  
#0.86230125427246093750,0.163365659713  
|  
:>
```

```
Plotting Enabled for x,y Data:  
tni32  
Rc = 32.00  
:> BJT Input Sweep  
Sweeping Vb 0.00 V to 4.60 V  
Vc/Vd: 4.6000 Vb/Vg 0.0000 - 4.60 inc 0.23  
Captured: 0.0, 0.0  
Captured: 0.1886284065246582, 0.0  
Captured: 0.4311506271362305, 0.0  
Captured: 0.6601994037628174, 0.005473584175  
Captured: 0.7140932083129883, 0.043788747787  
Captured: 0.7410401344299317, 0.077472376823  
Captured: 0.7545136451721192, 0.093472118377  
Captured: 0.7814605236053467, 0.103998255729  
Captured: 0.7814605236053467, 0.111577057838  
Captured: 0.7949339866638183, 0.118734836578  
Captured: 0.8084074020385742, 0.125050520896  
Captured: 0.8084074020385742, 0.130524110794  
Captured: 0.8218809127807617, 0.135576665401  
Captured: 0.8218809127807617, 0.140629196166  
Captured: 0.8353543281555176, 0.145681738853  
Captured: 0.8353543281555176, 0.149471163749  
Captured: 0.8353543281555176, 0.153681621551  
Captured: 0.8488277435302735, 0.1570499897  
Captured: 0.8488277435302735, 0.160418367385  
Captured: 0.862301254272461, 0.163365659713  
Captured 20 data pairs for 0 series. Press 'GO' to plot  
:>
```

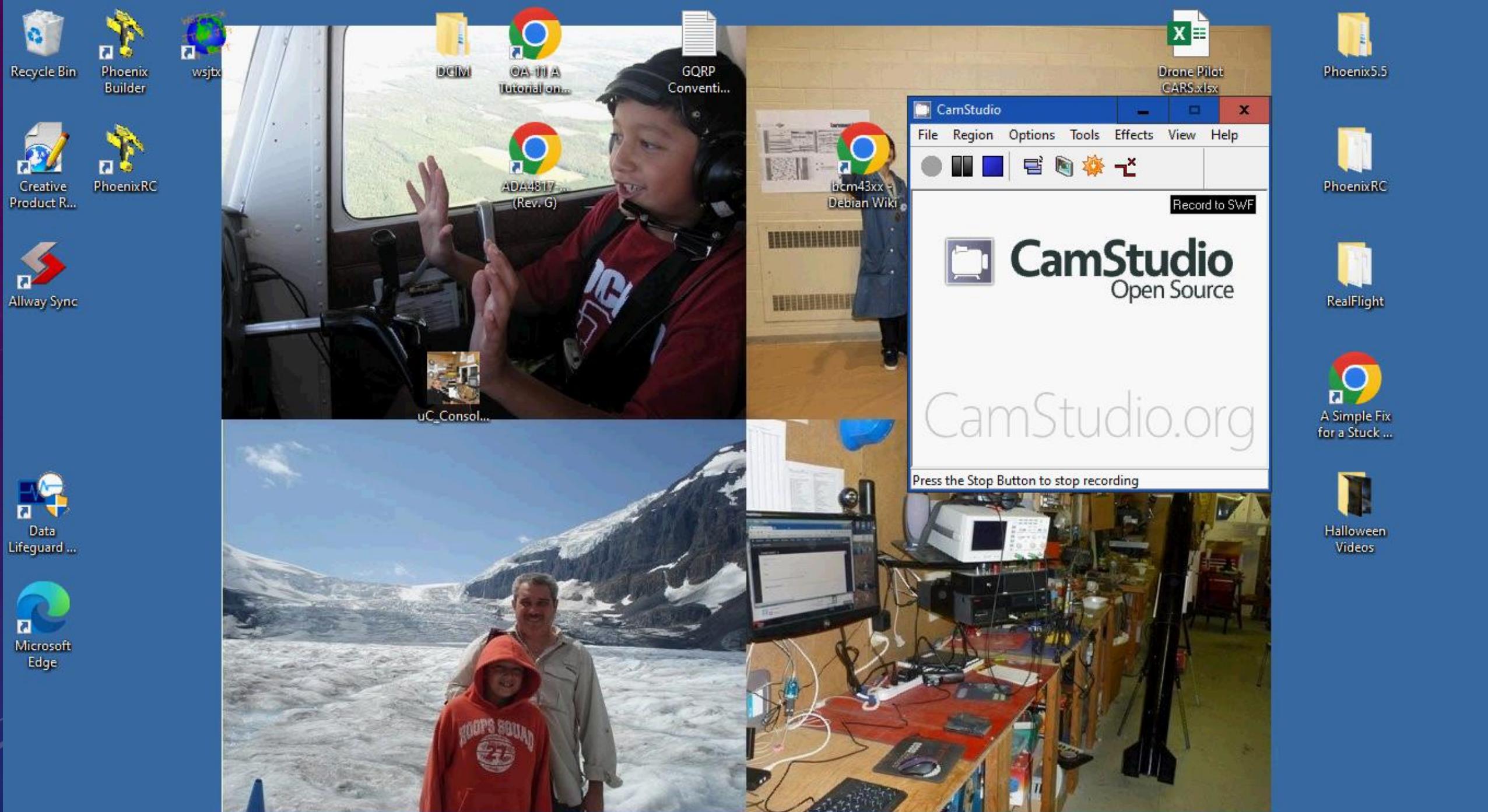
HOW IT WORKS

1. The Microcontroller must have a CLI
2. The uC must append "#" to numbers and send other control characters

```
# // x,y numbers  
| // end control characters  
= // switch control character  
// i.e., new xy plot)
```

tno 32
Rc = 32.00
> BJT Output Sweep
Sweeping Vb 0.00 V to 4.60 V
BJT NPN Output Sweep
Vb 0.0000 - 2.3000
Vc 0.0100 - 4.6000
#0.000000000000,0.000000000000,0.000000000000
#0.000000000000,0.848827743530,-0.000421045351
#0.000000000000,1.724602460861,0.000000000000
= #0.000013473452,0.000000000000,0.000000000000
#0.000000000000,0.848827743530,-0.000421045351
#0.000000000000,1.724602460861,0.000000000000
= #0.000013473480,0.000000000000,0.000000000000
#0.000013473480,0.835354328155,0.000000000000
#0.000013473480,1.724602460861,-0.000421043491
= #0.000080840768,0.000000000000,0.000000000000
#0.000040420351,0.026946914196,0.001263136625
#0.000026946902,0.579358625411,0.004631500720
= #0.000242522192,0.000000000000,0.000000000000
#0.000229048728,0.000000000000,0.000842091083
#0.000161681470,0.107787656784,0.017683911323
= #0.000431150627,0.000000000000,0.000000000000
#0.000417677164,0.000000000000,0.000842091083
#0.000377256870,0.067367286682,0.019368094444
= #0.000646725940,0.000000000000,0.000000000000
#0.000633252525,0.000000000000,0.000842091083
#0.000579358673,0.053893828392,0.019789141054
= | :>

Series Id (not used to plot)
X data point
Y data point
Switch to new XY plot
End of data. Enable plot



CLI MANUAL

ARDUINO SOFTWARE (FIRMWARE)

- This software was compiled using PlatformIO IDE. As is it will NOT compile with Arduino IDE.
- For Arduino IDE, simply rename the main.cpp to an Arduino program name (*.ino). Then create a directory with the same name and store all .cpp and .h files in that directory

main.cpp	3/10/2023 4:45 PM	CPP File	18 KB
main.h	2/28/2023 9:35 PM	H File	1 KB
nfet.cpp	3/7/2023 7:32 PM	CPP File	5 KB
nfet.h	3/1/2023 5:44 PM	H File	1 KB
npn.cpp	3/6/2023 8:55 PM	CPP File	6 KB
npn.h	3/5/2023 8:14 PM	H File	1 KB
trace.cpp	3/9/2023 6:42 PM	CPP File	13 KB
trace.h	3/10/2023 1:38 PM	H File	4 KB
UART.cpp	2/25/2023 8:49 PM	CPP File	5 KB
UART.h	2/25/2023 8:49 PM	H File	1 KB

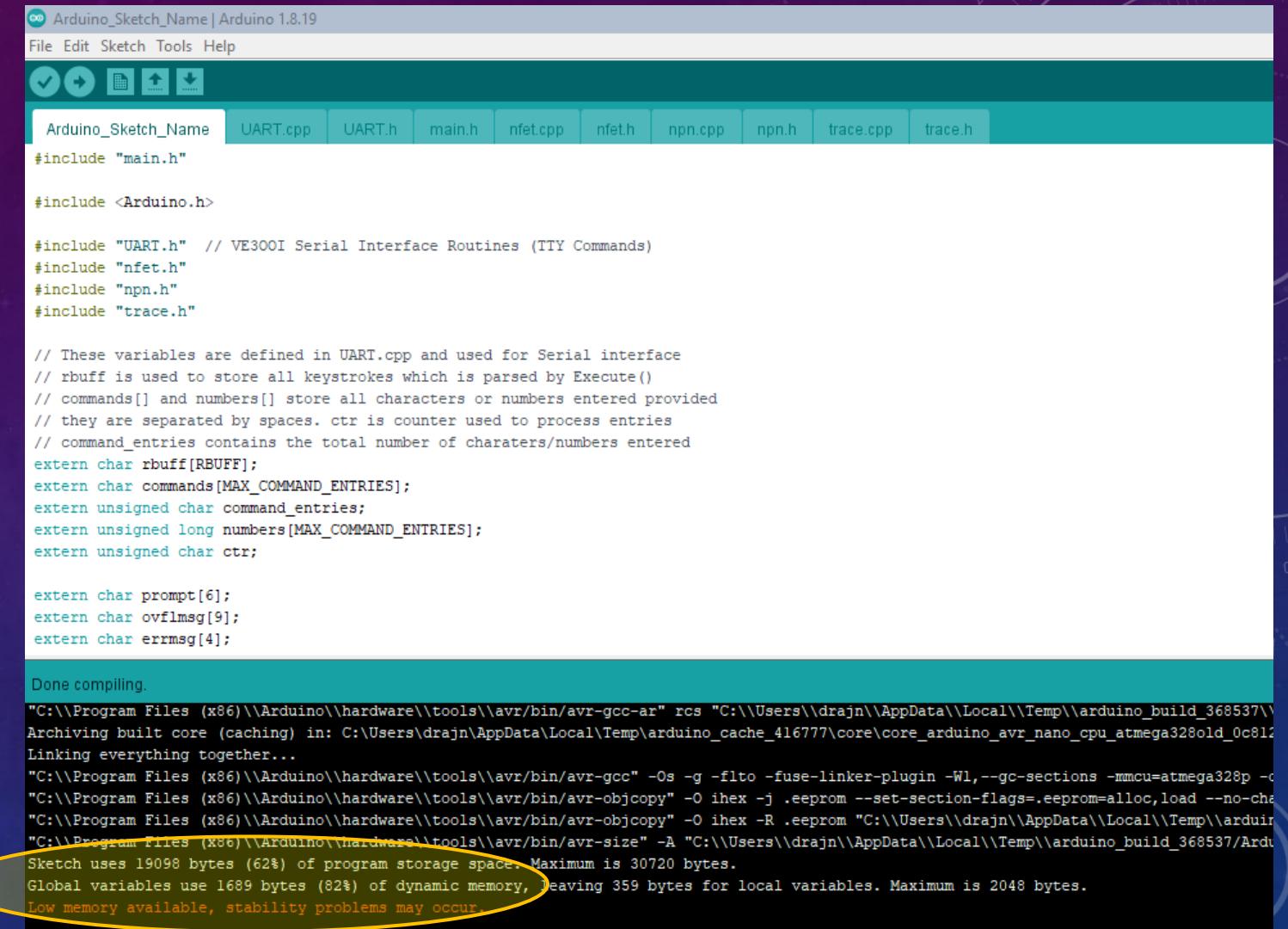


A yellow bracket on the right side of the slide groups the two tables, indicating they represent the same data structure.

Arduino_Sketch_Name	3/27/2023 4:32 PM	File folder
Arduino_Sketch_Name.ino	3/10/2023 4:45 PM	Arduino file
main.h	2/28/2023 9:35 PM	H File
nfet.cpp	3/7/2023 7:32 PM	CPP File
nfet.h	3/1/2023 5:44 PM	H File
npn.cpp	3/6/2023 8:55 PM	CPP File
npn.h	3/5/2023 8:14 PM	H File
trace.cpp	3/9/2023 6:42 PM	CPP File
trace.h	3/10/2023 1:38 PM	H File
UART.cpp	2/25/2023 8:49 PM	CPP File
UART.h	2/25/2023 8:49 PM	H File

ARDUINO IDE VS PLATFORM.IO

	Arduino	PlatformIO
RAM	1689	1414
FLASH	23334	19098



The screenshot shows the Arduino IDE interface with a sketch named "Arduino_Sketch_Name". The code includes includes for main.h, Arduino.h, UART.h, nfet.h, npn.h, and trace.h. It defines extern variables for rbuf, commands, command_entries, numbers, ctr, prompt, ovflmsg, and errmsg. A message at the bottom states: "Done compiling. Sketch uses 19098 bytes (62%) of program storage space. Maximum is 30720 bytes. Global variables use 1689 bytes (82%) of dynamic memory, leaving 359 bytes for local variables. Maximum is 2048 bytes. Low memory available, stability problems may occur." A yellow oval highlights the memory usage statistics at the bottom.

```
#include "main.h"

#include <Arduino.h>

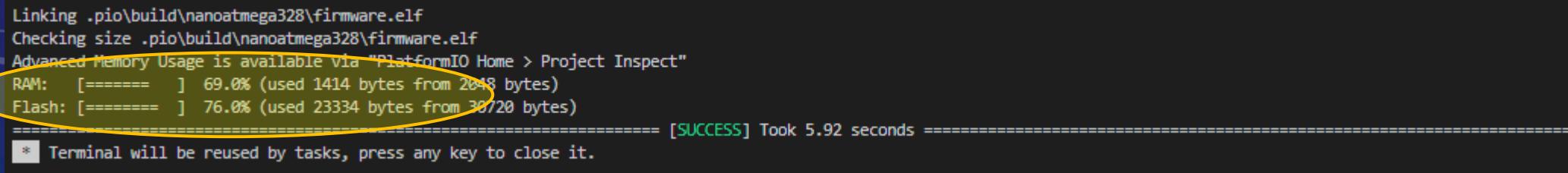
#include "UART.h" // VE300I Serial Interface Routines (TTY Commands)
#include "nfet.h"
#include "npn.h"
#include "trace.h"

// These variables are defined in UART.cpp and used for Serial interface
// rbuf is used to store all keystrokes which is parsed by Execute()
// commands[] and numbers[] store all characters or numbers entered provided
// they are separated by spaces. ctr is counter used to process entries
// command_entries contains the total number of characters/numbers entered
extern char rbuf[RBUFF];
extern char commands[MAX_COMMAND_ENTRIES];
extern unsigned char command_entries;
extern unsigned long numbers[MAX_COMMAND_ENTRIES];
extern unsigned char ctr;

extern char prompt[6];
extern char ovflmsg[9];
extern char errmsg[4];

Done compiling.

"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-gcc-ar" rcs "C:\Users\drajn\AppData\Local\Temp\arduino_build_368537\core\core_arduino_avr_nano_cpu_atmega328old_0c812d.o"
Archiving built core (caching) in: C:\Users\drajn\AppData\Local\Temp\arduino_cache_416777\core\core_arduino_avr_nano_cpu_atmega328old_0c812d.a
Linking everything together...
"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-gcc" -Os -g -flto -fuse-linker-plugin -Wl,--gc-sections -mmcu=atmega328p -c
"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-objcopy" -O ihex -j .eeprom --set-section-flags=.eeprom=alloc,load --no-ch
"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-objcopy" -O ihex -R .eeprom "C:\Users\drajn\AppData\Local\Temp\arduino
"C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-size" -A "C:\Users\drajn\AppData\Local\Temp\arduino_build_368537\Ardu
Sketch uses 19098 bytes (62%) of program storage space. Maximum is 30720 bytes.
Global variables use 1689 bytes (82%) of dynamic memory, leaving 359 bytes for local variables. Maximum is 2048 bytes.
Low memory available, stability problems may occur.
```



The screenshot shows the PlatformIO terminal output. It displays memory usage statistics: RAM usage is 69.0% (1414 bytes from 2048 bytes), and Flash usage is 76.0% (23334 bytes from 30720 bytes). The output ends with "[SUCCESS] Took 5.92 seconds" and a note that the terminal will be reused by tasks.

```
Linking .pio\build\nanoatmega328\firmware.elf
Checking size .pio\build\nanoatmega328\firmware.elf
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [=====] 69.0% (used 1414 bytes from 2048 bytes)
Flash: [=====] 76.0% (used 23334 bytes from 30720 bytes)
=====
===== [SUCCESS] Took 5.92 seconds =====
* Terminal will be reused by tasks, press any key to close it.
```

CLI GUIDE: SUMMARY

```
H          // Help - Command summary  
  
C          // Display current Calibration  
C [R|M|D] // Calibrate  
D          // Set default values  
E [I|P]   // Configure EEPROM  
M [R]     // Measure Voltages  
P [3|10 [255] // Set PWM on pin 3 or pin 10  
R          // Reset  
S [V] [B|C] [0] [mV] // Set voltage on Base or Vcc pin.  
S [I] [uA]    // Set current on Base  
S [U] [D|M|U] // Set default unit to decimal, Milla or Micro for currents  
S [M] [B|C] [min] [max] // Set min max voltage to sweep  
S [R] [B|C] [R] // Set Rb or Rc resistor values  
S [P] [B|C] [n] // Set number of data point in sweep between max and min  
T [NI|NIV|NO|NB|NBC|NR|NG] // Trace NPN Transistor  
T [FI|FIV|FO|FK|FG|FR]    // Trace N Chan MOSFET  
T [DI|DO]      // Trace Diode
```

CLI GUIDE: CALIBRATION

Display current Calibration values

C

Connect the Tracer to Vcc with empty socket (nothing connected) to determine the voltage sweep range for base/gate voltage and collector/drain voltage. It also calculated the conversion constant to convert collect/drain (pin 10) PWM voltage to actual output Vcc voltage delivered to DUT

C M

Connect the Tracer to Vcc with empty socket (nothing connected) to determine the voltage sweep range for base/gate voltage and collector/drain voltage. It also calculated the conversion constant to convert collect/drain (pin 10) PWM voltage to actual output Vcc voltage delivered to DUT

C D

Connect the Tracer to Vcc with empty socket (nothing connected). It generates a PWM voltage on collector/drain pin defined by a calibration voltage constant (see below) and then measures and displays all ADC values 10 times. This is one way to calibrate ADC pots. Press the reset button or enter the R command to stop the display.

C R

BUILD TIPS

- 1. DO NOT CONNECT ARDUINO AND POWER UP BEFORE CALIBRATION IS DONE**
2. Do you best to get parts as labelled.
3. POTS show incorrect values. They should all be 100K
4. The TIP120 can be replaced with a 2SC5706 power transistor
5. The 1N5817 can be replaced with any 1 Amp Shottky Diode
6. The board can be powered from either J8 or J5. Maximum voltage is 14 V. Exceeding this could damage your Arduino
7. R1, R3, R6 and R7 resistors cannot be substituted.
8. Capacitors C1, C2, C7, C8, C9 and C10 cannot be substituted. Other capacitors can be used similar values.
9. The D3, D4, D5, and D6 4.9V Zener diodes are optional and are there to protect Arduino ADC from overvoltage
10. Resistors R10 and R11 form a voltage divider to read Vcc voltage. If you change these values ensure the ratio will allow 14V to deliver about 4.8V to the Arduino. Measure and note the values of the resistors you use. You will need to enter them in the Arduino code
11. R4, R5 and R8 set the maximum current to the base or to the collector. You can substitute values here. Keep R4 close to the 500R (i.e. within 10 or 20R). R5 and R8 must be 2W resistors. They dissipate several hundred milliamps. Measure and note the values of the resistors you use. You will need to enter them in the Arduino code
12. Solder all SMT parts
13. Solder all through hole components from the underside
14. Arrange the pots such that turning clockwise increases the resistance between the wiper and the grounded leg.
15. Once you're finished remove all jumpers from header pins. Do NOT connect the Arduino. Power up and check voltages. Proceed to calibration.

CALIBRATION & CONFIGURATION FOR VERSION 1

1. **DO NOT CONNECT ARDUINO AND POWER UP THE BOARD BEFORE CALIBRATION IS DONE**
2. Before powering up Arduino, you need to calibrate pots to not damage the Arduino.
3. Remove Arduino. Remove jumpers from all trim pots jumper headers pins.
4. Connect Vcc to the board and measure the voltage at the +5V terminal pin (J7).
5. Connect a jumper wire from +5V terminal to each pot's jumper "+" pin (J1, J2, J6, J7). "+" is marked on the silkscreen
6. Adjust each pot until you get 1.75V output (exactly) at the pot's wiper. Use one consistent VOM.
7. Insert Arduino and load the code. Power up.
8. Connect via TTY console to Arduino. Enter the **C R** command to display all ADC values from each pot
9. Note that each pot has ADC1, ADC2, ADC3, ADC4 marked on the silkscreen. Each jumper associate with the pot is marked as AD1, AD2, AD3, AD4. This corresponds to the ADC number on the Arduino.
10. Connect 5V to the same jumper in step 3. Note the ADC number of the jumper and pot.
 - a. Adjust the pot's wiper (very fine adjust needed) until the corresponding ADC displayed by the Arduino is **EXACTLY** 364. Repeat for all jumpers/pots.
 - b. Enter the **R** command to reset the Arduino when done
11. Complete and record the following calculations for each pot:
 - a. Calculate **VOLTAGE_DIVIDER** = $V(\text{wiper})/V(\text{applied})$ e.g., $1.75/5 = 0.357$
 - b. Calculate **ADC_CONVERSION** = $\text{ADC_Value}/V(\text{wiper})$ e.g., $364/1.75 = 208.0$
 - c. Calculate **VOLTAGE_CONVERSION** = **VOLTAGE_DIVIDER** * **ADC_CONVERSION** e.g., $0.357*208 = 78.0$
12. Take the average **VOLTAGE_CONVERSION** and update the **#define ADC_CONVERSION ((float)78.0)** in the **defines.h** file
13. You need to measure the resistance of the Base and Collector resistors. Update the following defines in the **defines.h** file

```
#define RC_RESISTORHIGH 32  
#define RC_RESISTORLOW 10  
#define RB_RESISTORLOW 462
```
14. Reset the Arduino and measure the max voltage for the base and collectors. Ensure no jumpers are connected. Enter the **P 3 255** command and measure and record the voltage after the 200R resistor connected in D3 pin. Reset the Arduino and repeat for pin 10. Enter the **P 10 255** and measure and record the voltage after the 200R resistor connected to D10 pin. Update the following defines in the **defines.h** file
 1. **#define MAX_VC_VOLTAGE 4.6**
 2. **#define MAX_VB_VOLTAGE 4.6**
15. Compile and load the Arduino program before continuing. The above defines must be in place.
16. Complete the following. DO NOT RESET OR POWER OFF ARDUINO UNTIL ALL STEPS ARE COMPLETE. Failure will result in redoing below
 - a. Set the base resistor value using the **S R B nnnn** command where **nnnn** is the resistor value. E.g., **S R B 462**
 - b. Set the jumper J4 (RCSEL) to the resistor for the sweep. Use the **S R C nnnn** command to define the Collector resistor **nnnn** that the jumper selected. E.g., **S R C 10**
 - c. With Vcc applied and nothing connected to the ZIF socket (socket is empty), run the **C M** command to have the Arduino figure out the max voltages available to it during a sweep. If you change Vcc, this needs to be repeated.
17. Final calibration step is to get the Arduino to measure the min and max voltages available when generating a trace. Insert jumpers in J1, J2, J6, & J7. Ensure there is nothing connected to the ZIF socket. Enter the **C M** command. This takes a few minutes to complete.
18. Update the EEPROM with the configuration by entering the **E W** command. The Arduino will now power up with the values you defined above.
19. Your transistor curve trace (V1) is now ready.

OPERATION: SWEEPS

1. The transistor tracer sweeps a set of PWM voltages.
 1. There is a base/gate PWM voltage (Pin 3) that is applied directly to the transistor under test.
 2. There is a Vcc PWM voltage (Pin 10) that is amplified by an opamp and a high-power transistor
2. Since the base/gate PWM voltage is fed directly to the transistor under test, the base/gate voltage is essentially the output voltage to the transistor
3. However, since the Vcc PWM voltage is amplified, the output voltage fed to the collector/drain of the transistor under test is different.
4. When sweeping Vcc values, only the PWM voltage is used. **When you set voltage ranges, you are setting the PWM voltage that the Arduino outputs.**
5. You can manually set the output voltage supplied to the transistor under test using the S V command. However, this is a one-time setting. Its not used in a sweep. **BE CAREFULL SETTING VOLTAGES MANUALLY** – you could destroy a resistor or the transistor under test if you exceed its maximum ratings

REMEMBER: When you set voltage ranges, you are setting the PWM voltage that the Arduino outputs. You are NOT setting the output voltage fed to the transistor under test

OPERATION: NPN SWEEPS

1. The **T** command is used to complete Traces. The **T N** command is used to complete NPN Transistor traces
2. The following traces uses the configured min and max voltages. You can adjust sweep voltages and number of data points (resolution) to be taken during a sweep. Use the **S M** command to adjust min & max voltages for and **S P** to set data points.
3. The following below outlines the various traces:
 - I. Input Trace: **T N I**
 - ✓ This command sets the Vcc voltage to the maximum available voltage and then sweeps the base voltage from configured min to max at define inc (resolution). It prints out Vbe vs Ic
 - II. Input Trace: **T N I V**
 - ✓ This command is the same as above except it prints out Vbe vs Vce
 - III. Output Trace: **T N O**
 - ✓ This command sweeps the base voltage from configurated min to max with inc. For each base voltage, it then sweeps Vcc from configured min to max at defined inc. For each base voltage it print out Vce vs Ic in the form Ib, Vce, Ic
 - IV. Beta Trace: **T N B**
 - ✓ This command sets the Vcc voltage to the maximum available voltage and then sweeps the base voltage from configured min to max at define inc (resolution). It calculates transistor beta and prints out Ib vs Beta
 - V. Beta Trace: **T N B C**
 - ✓ This command is the same as above except it prints out prints out Ic vs Beta
 - VI. Beta Trace: **T N R**
 - ✓ This command sets the Vcc voltage to the maximum available voltage and then sweeps the base voltage from configured min to max at define inc (resolution). It calculates transistor re (intrinsic resistance) and prints out Ic vs re
 - VII. Beta Trace: **T N G**
 - ✓ This command sets the Vcc voltage to the maximum available voltage and then sweeps the base voltage from configured min to max at define inc (resolution). It calculates transistor transconductance and prints out Ic vs g_m

OPERATION: N-CHANNEL ENHANCEMENT MODE MOSFET SWEEPS

1. The **T** command is used to complete Traces. The **T F** command is used to complete N channel Mosfet traces
2. The following traces uses the configured min and max voltages. You can adjust sweep voltages and number of data points (resolution) to be taken during a sweep. Use the **S M** command to adjust min & max voltages for and **S P** to set data points.
3. The following below outlines the various traces:

I. Input Trace: **T F I**

- ✓ This command sets the Vcc voltage to the maximum available voltage and then sweeps the gate voltage from configured min to max at define inc (resolution). It prints out V_{gs} vs I_d

II. Input Trace: **T F I V**

- ✓ This command is the same as above except it prints out V_{gs} vs V_{ds}

III. Output Trace: **T F O**

- ✓ This command sweeps the base voltage from configurated min to max with inc. For each gate voltage, it then sweeps Vcc from configured min to max at defined inc. For each gate voltage it print out V_{ds} vs I_d in the form I_d, V_{gs}, I_d

IV. Intrinsic Resistance Trace: **T F R**

- ✓ This command sets the Vcc voltage to the maximum available voltage and then sweeps the gate voltage from configured min to max at define inc (resolution). It calculates transistor r_e (intrinsic resistance) and prints out I_d vs R_{ds}

V. Transconductance Trace: **T F G**

- ✓ This command sets the Vcc voltage to the maximum available voltage and then sweeps the gate voltage from configured min to max at define inc (resolution). It calculates transistor transconductance and prints out I_d vs g_m

VI. Transconductance Trace: **T F G**

- ✓ This command is the same as above except it it calculates transistor transconductance constant K and prints out I_d vs K

OPERATION: DIODE SWEEPS

1. The **T** command is used to complete Traces. The **T D** command is used to complete diode traces
2. The following traces uses the configured min and max voltages. You can adjust sweep voltages and number of data points (resolution) to be taken during a sweep. Use the **S M** command to adjust min & max voltages for and **S P** to set data points.
3. Note that the diode **MUST** be oriented correctly in the socket.
4. The following below outlines the various traces:

I. Input Trace: **T D I**

- ✓ The diode must be connected between the base/gate socket and ground. The cathode is connected to emitter/source socket (ground). For a Zener diode, its reversed, the cathode is connected to the base. This command sets the Vcc voltage to the maximum available voltage and then sweeps the base/gate voltage from configured min to max at define inc (resolution). It prints out V_b vs I_d

II. Output Trace: **T D O**

- ✓ The diode will need a resistor connected between the cathode and the emitter/source socket (ground). The resistor is selected to NOT exceed the maximum current the diode can handle. The resistor also protect the Arduino pin used for base PWM output (pin 3). **Failure to add a resistor could cause damage to the Arduino.** This command set the base/gate voltage Arduino pin to zero to sink current. It then sweeps the Vcc voltage from configurated min to max with inc. For each Vcc voltage, it measure the voltage drop across the collector/drain to the base/gate sockets along with current. It print out V_f vs I



**Special Jig Required
for Output Sweep**

OPERATION: CUSTOM SWEEPS

1. The **S** command is used to set trace values.
 2. For base/gate voltage there is a min voltage, max voltage
 3. It sets the PWM voltage output from the Arduino. **IT DOES NOT SET THE ACTUAL OUTPUT VOLTAGE DELIVERED TO THE DUT**
 4. Values **MUST** be entered as milla volts (mV) e.g., 4V entered at 4000
 5. The following below outlines the various configuration options:
- I. Base/Gate Sweep: **S M B xxxx yyyy**
- ✓ This command sets the minimum voltage (xxxx) and the maximum voltage (yyyy) for the base/gate PWM pin. The increment use for trace is automatically calculated to the number of data point defined. Default is min: 0 and maximum determined by **C M** calibration command.
- II. Vcc Sweep: **S M C xxxx yyyy**
- ✓ This command sets the minimum voltage (xxxx) and the maximum voltage (yyyy) for the Vcc PWM pin. The increment use for trace is automatically calculated to the number of data point defined. Default is min: 0 and maximum determined by **C M** calibration command.
 - ✓ Input sweeps use the maximum Vcc voltage. Therefore, you can manually reduce the Vcc voltage (e.g., for $V_{ce(sat)}$) used for input sweeps by setting the maximum Vcc voltage.

You need to enter the **E W** command to update the EEPROM configuration values or else these definition will be lost at next restart of the Arduino

OPERATION: CUSTOM SWEEPS

1. The **S** command is used to set trace values.
2. You can configure the number of data point to be taken during a typical sweep. Some sweeps such as the beta, gm, K and Rds and re sweeps typically take 25% of the increment.
3. Example:
 1. If the min is 0.01V and the max is 4.6V
 2. If you set the number of point to 10, then the increment uses during a sweep is $(4.6 - 0.01)/20 = 0.45$
 3. Therefore, during a sweep, 10 measurements are taken with each increasing by 0.45V from min to max.
4. The following below outlines the various configuration options:

I. Base/Gate Sweep: **S P B xxxx**

- ✓ This command sets the number of data points to measure during a sweep of the base/gate voltage. The increment used for sweep is automatically calculated. This can give a smaller step size (e.g., increment of 0.1V instead of 0.4V). Default is 10 data points for base/gate.
- ✓ The range of allowed data points is defined in the **defines.h** file as:

```
#define MIN_NUM_VB_VOLTAGE 5  
#define MAX_NUM_VB_VOLTAGE 20  
#define NUM_VB_VOLTAGE 10           // Default number of points
```

II. Vcc Sweep: **S P C xxxx**

- ✓ This command sets the minimum voltage (xxxx) and the maximum voltage (yyyy) for the Vcc PWM pin. The increment use for trace is automatically calculated to the number of data point defined. Default is 15 data points for Vcc.
- ✓ The range of allowed data points is defined in the **defines.h** file as:

```
#define MIN_NUM_VC_VOLTAGE 5  
#define MAX_NUM_VC_VOLTAGE 20  
#define NUM_VC_VOLTAGE 15          // Default number of points
```

You need to enter the **E W** command to update the EEPROM configuration values or else these definition will be lost at next restart of the Arduino

OPERATION: EEPROM MANAGEMENT

The E command is used to manage configuration values stored in the EEPROM.

- Configuration values stored in the EEPROM will be restored whenever the Arduino is power on.
- The following below outlines the various configuration options:

I. Read EEPROM: E R

- ✓ This command reads and overwrites the current running configuration with the configuration stored in the EEPROM.
Any unsaved changes will be lost (i.e., changes to voltage ranges, points, resistor values, max voltages, etc)

II. Write EEPROM: E W

- ✓ This command writes the current configuration to the EEPROM. It overwrites the EEPROM.

III. Initialize EEPROM: E I

- ✓ This command sets the default configuration values, calculates the max voltages and then writes the default configuration to the EEPROM. Think of this as a factory reset.

OPERATION: MISC MANAGEMENT

There are a few other MISC command that can be used. For example, troubleshooting.

I. Set Voltages: **S V B xxxx or S V C xxxx**

- ✓ This set the PWM voltage at pin 3 (base voltage) and pin 10 (Vcc voltage). The voltage is given by xxxx and is in milla volts (i.e., mV). If 0 (0h and not zero) is added after the **B** or **C** then, the Arduino tries to set the output voltage delivered to the transistor under test. Note that this value could be off by up to 0.5 volts.

II. Set PWM: **P xx yyy**

- ✓ This command manually sets the PWM output of pin 3 or pin 10 of the Arduino and is used for troubleshooting.
- ✓ xx is the pin number (3 or 10) and yyy is the PWM value (1-255). A higher PWM values will output a higher voltage. A value of 128 will give ½ of the maximum output voltage.

III. Measure: **M**

- ✓ This reads and display all the voltages and currents. It assumed you have manually set a voltage using the **P** or **V** commands

IV. Reset: **R**

- ✓ This resets the Arduino. Its not the same as a power on.

V. Set Defaults: **D**

- ✓ This command sets the default parameter to the running config. It does not change the EEPROM config.

VI. Set current units for graph: **S U D** or **S U M** or **S U U**

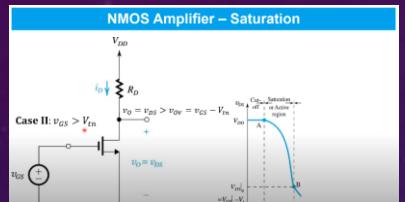
- ✓ This command sets units for current values produced. **S U D** is used for normal decimal, **S U M** is used for milla and **S U U** is used for micro. Default is **D**

MEASUREMENTS OF NPN BJT TRANSISTORS

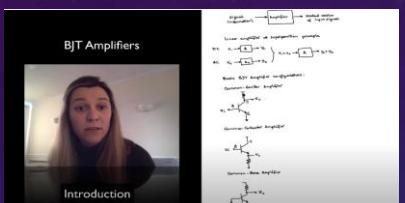
IF YOU ARE LOOKING FOR A TUTORIAL...



W2AEW: Back to Basics: Basics of the Bipolar Transistor - NPN and PNP
<https://www.youtube.com/watch?v=lrAUC4ZoNYY&t=1263s>



Microelectronic Prof. Tony Chan Carusone: Transistor Amplifier Basic Principles
<https://www.youtube.com/watch?v=lrAUC4ZoNYY&t=1263s>



Mateo Aboy: Series of Videos on BJT and FET Amplifiers
https://www.youtube.com/playlist?list=PLZvLSclgk4yJkRHB_eJ10zo2exTY_irR0



Electronics with Professor Fiore: Semiconductor Devices
https://www.youtube.com/playlist?list=PLxuejeK2BP_dtn1ijj1lZKrIs2gHPPsyp



MOS FET Modification Transistor Curve Tracer HeathKit
<https://www.youtube.com/watch?v=rFmDUPVdYSc&t=4953s>

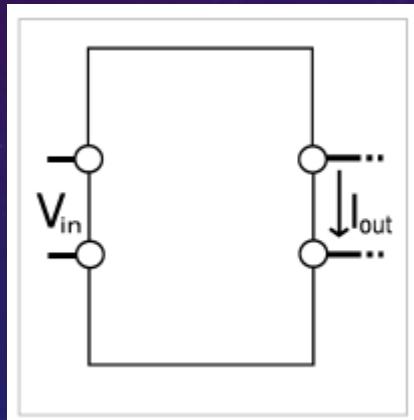
TRANSCONDUCTANCE

Transconductance (for **transfer conductance**), also infrequently called **mutual conductance**, is the electrical characteristic relating the **current** through the output of a device to the **voltage** across the input of a device. Conductance is the reciprocal of resistance.

Transconductance is a contraction of *transfer conductance*. The old unit of conductance, the mho (ohm spelled backwards), was replaced by the SI unit, the **siemens**, with the symbol S (1 siemens = 1 ampere per volt).

Transconductance is very often denoted as a conductance, g_m , with a subscript, m, for *mutual*. It is defined as follows:

$$g_m = \frac{\Delta I_{\text{out}}}{\Delta V_{\text{in}}}$$



A voltage applied at the input causes a current at the output.
Transconductance is a measure of the slope. It's like a "transfer function"

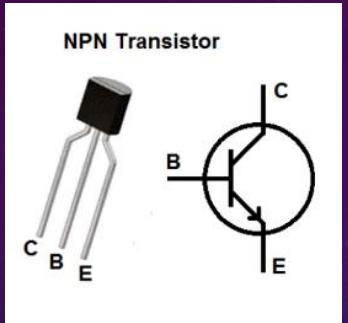
This is a very important concept for MOSFETs

It is used with NPN Transistors, but they are more a current device

A transfer function is a convenient way to represent a linear, time-invariant system in terms of its input-output relationship. It is obtained by applying a Laplace transform to the differential equations describing system dynamics, assuming zero initial conditions. In the absence of these equations, a transfer function can also be estimated from measured input-output data.

TRANSISTOR OPERATION

NPN



Feed Small Current into Base

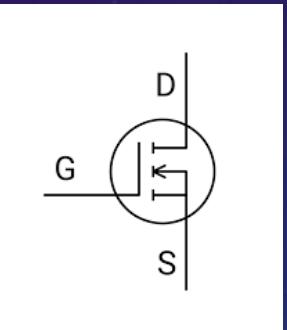
NPN TRANSISTOR

β

Get a LARGE current between Collector and Emitter

Datasheets today use h_{fe} instead of Beta. Beta sounds “cooler”

N-CH
MOSFET



Feed Voltage above Threshold

NMOS TRANSISTOR

g_m

Get a LARGE current between Drain and Source

IF YOU ARE LOOKING FOR A TUTORIAL...



W2AEW: Back to Basics: Basics of the Bipolar Transistor - NPN and PNP
<https://www.youtube.com/watch?v=lrAUC4ZoNYY&t=1263s>



Microelectronic Prof. Tony Chan Carusone: Transistor Amplifier Basic Principles
<https://www.youtube.com/watch?v=lrAUC4ZoNYY&t=1263s>



Mateo Aboy: Series of Videos on BJT and FET Amplifiers
https://www.youtube.com/playlist?list=PLZvLScIgk4ylkRHB_eJ10zo2exTY_irRO



Electronics with Professor Fiore: Semiconductor Devices
https://www.youtube.com/playlist?list=PLxuejeK2BP_dtn1jj1ZKrls2gHPPsyP



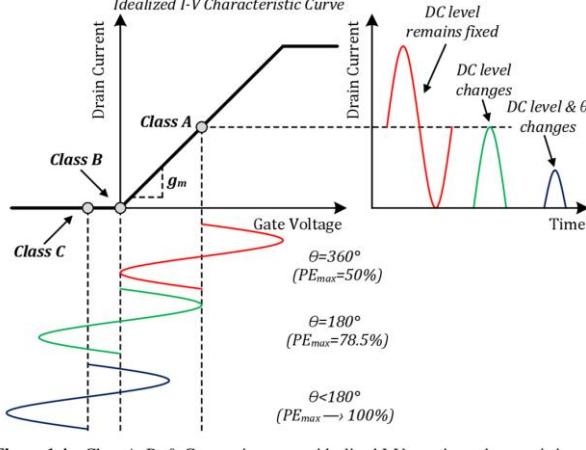
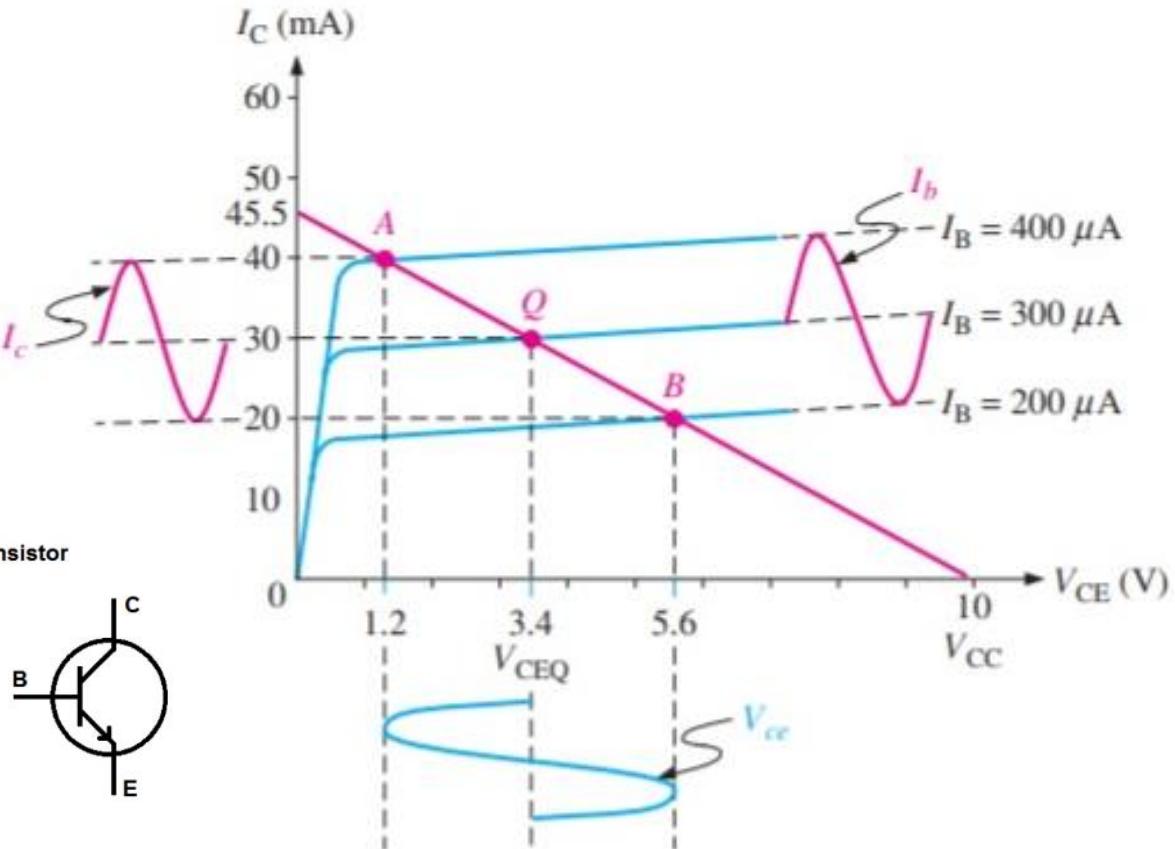
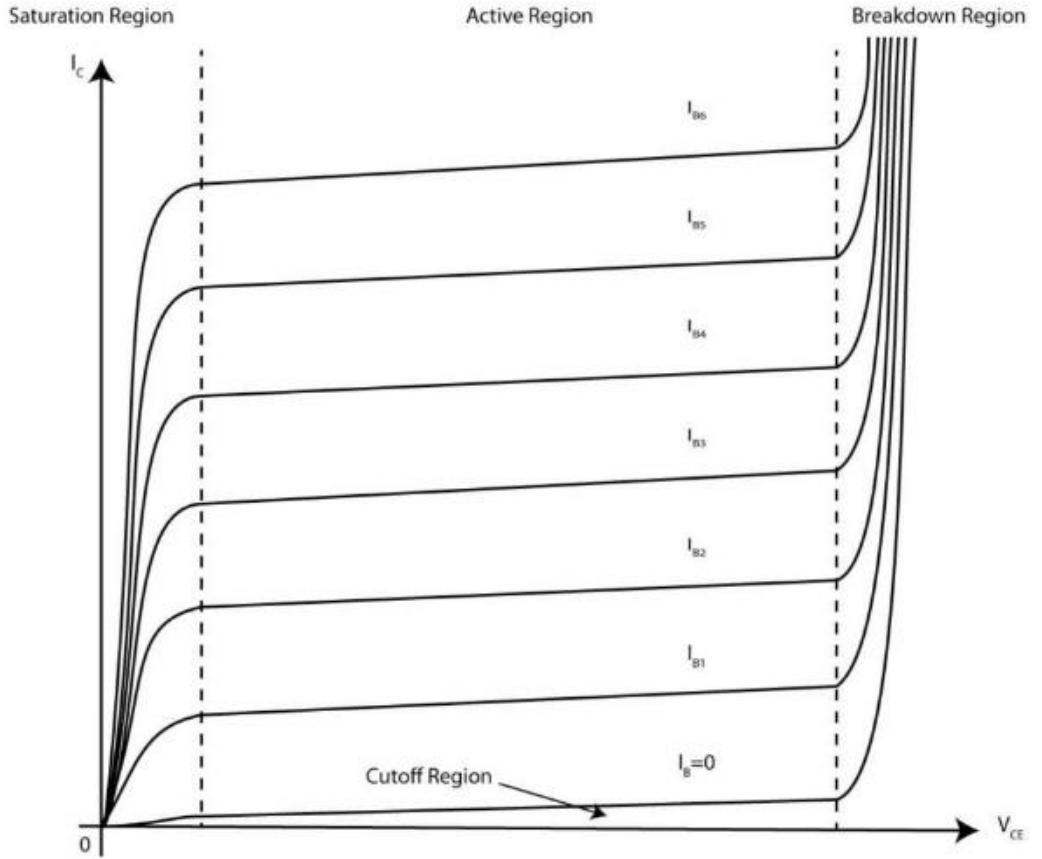
MOS FET Modification Transistor Curve Tracer HeathKit
<https://www.youtube.com/watch?v=rFmDUPvdYsc&t=4953s>

TRANSISTOR REGIONS: VOLTAGE & CURRENT

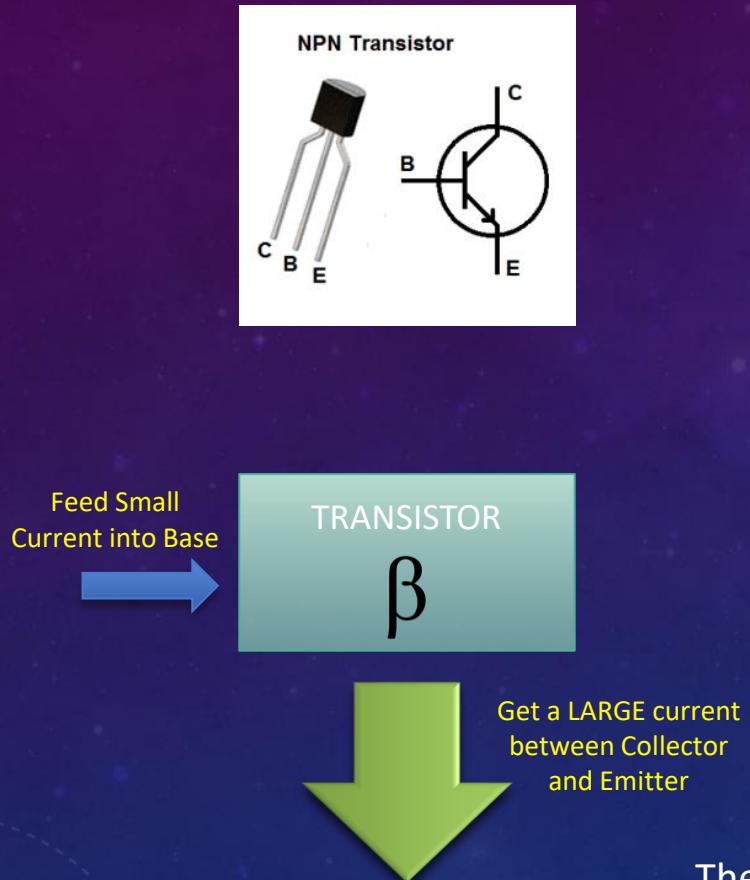
The **Breakdown Region** of a transistor is the region where the collector voltage, V_{CC} , is so large that the **collector-base diode breaks down**, causing a large, undesired collector current to flow.

When the collector-base voltage is too large, the collector-base diode breaks down, so that the collector conducts electricity. So even though the base of the transistor doesn't receive any current, the transistor still conducts across the collector. This is called the breakdown region.

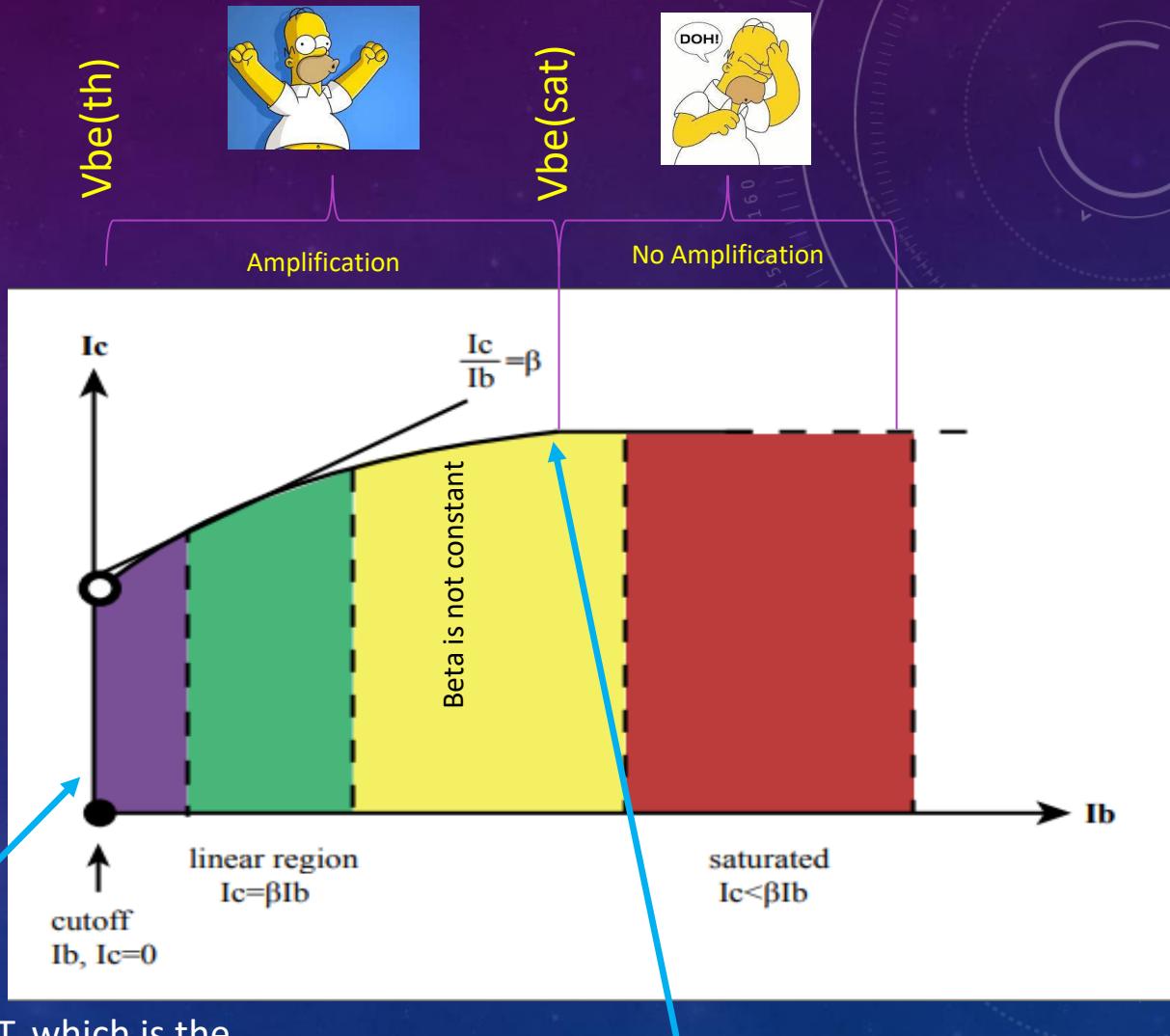
The breakdown region should always be avoided in transistor circuits. This is achieved by not placing too much bias voltage on the collector.



TRANSISTOR REGIONS: CURRENT

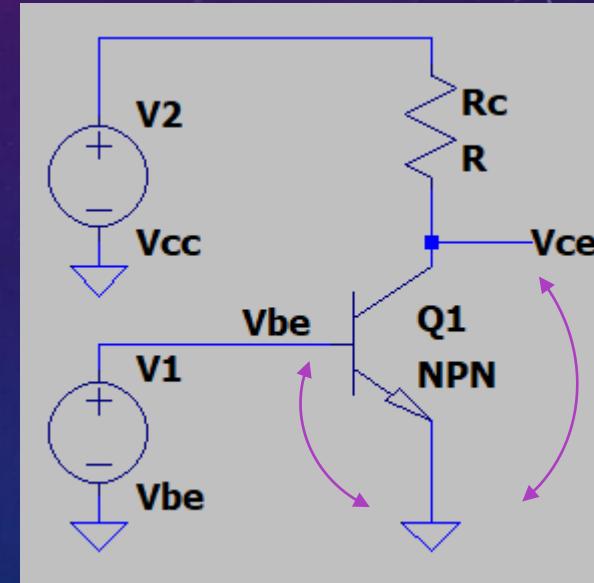
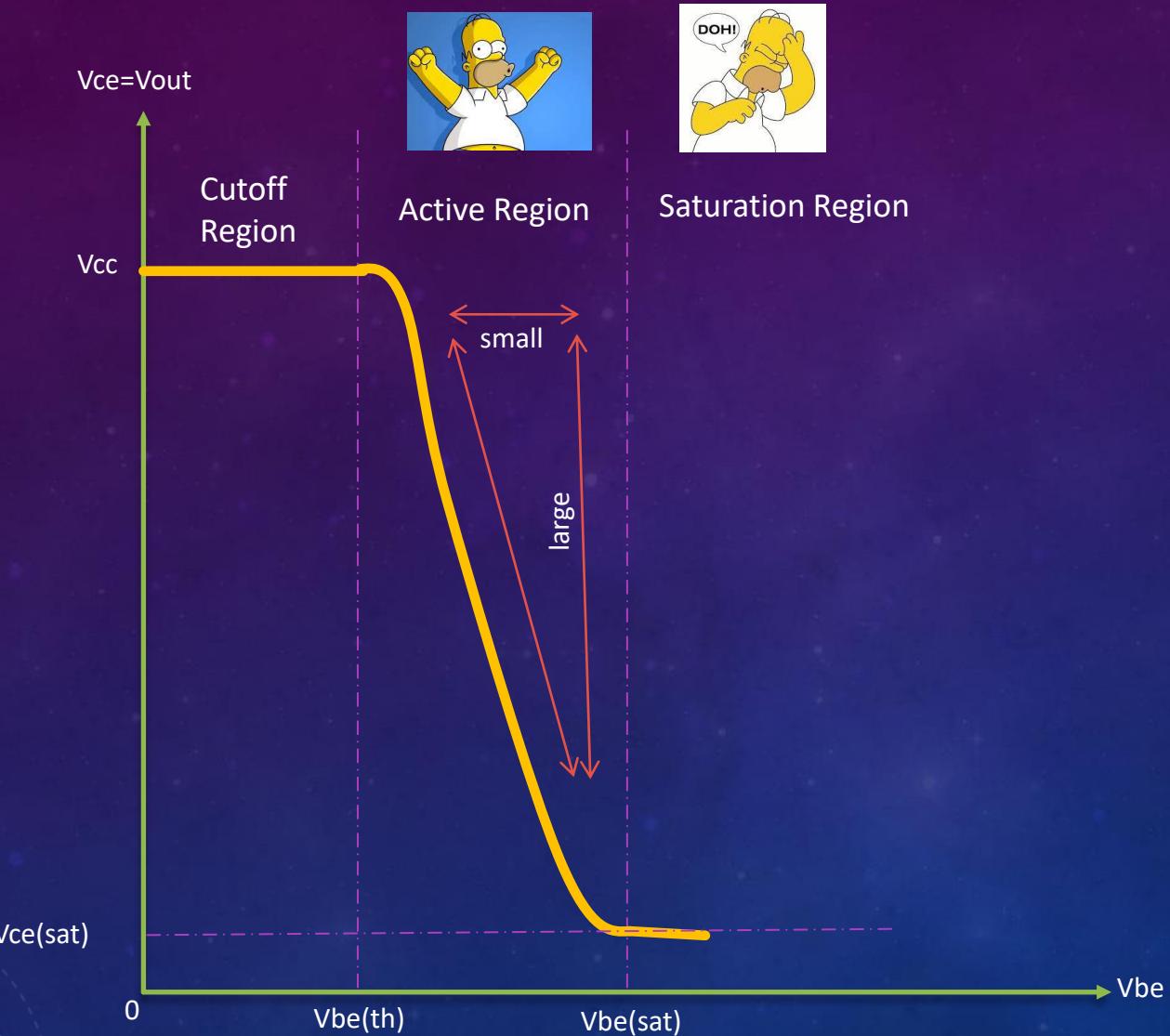


There is “ $V_{be(th)}$ ” for BJT, which is the threshold voltage (actually current) at which device starts conducting. ***Not specified on data sheet!!***



There is $V_{be(sat)}$ for BJT, which is the maximum voltage at which collector current is constant.

TRANSISTOR REGIONS: VOLTAGE



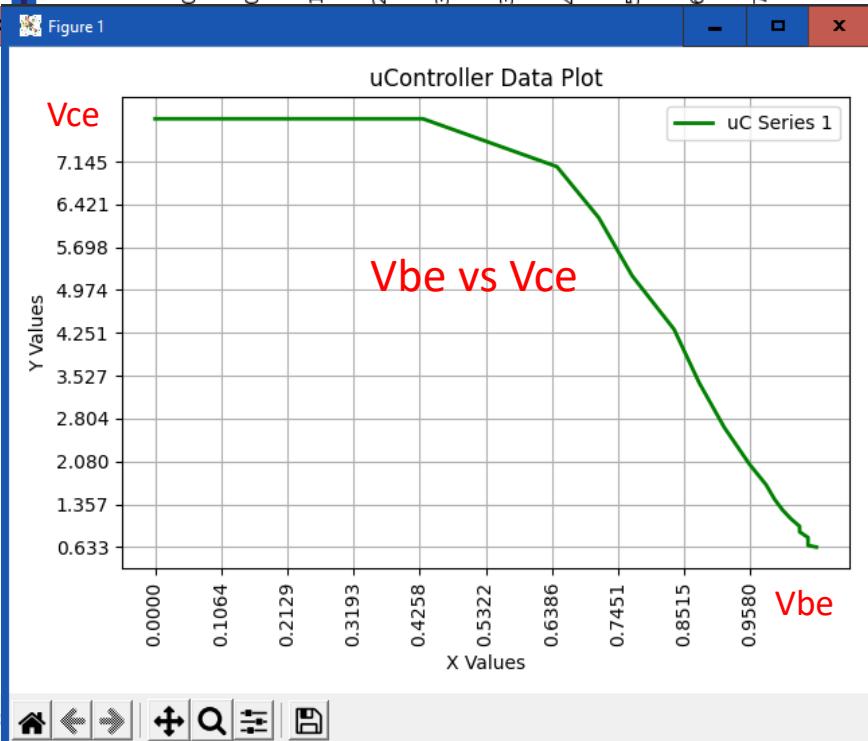
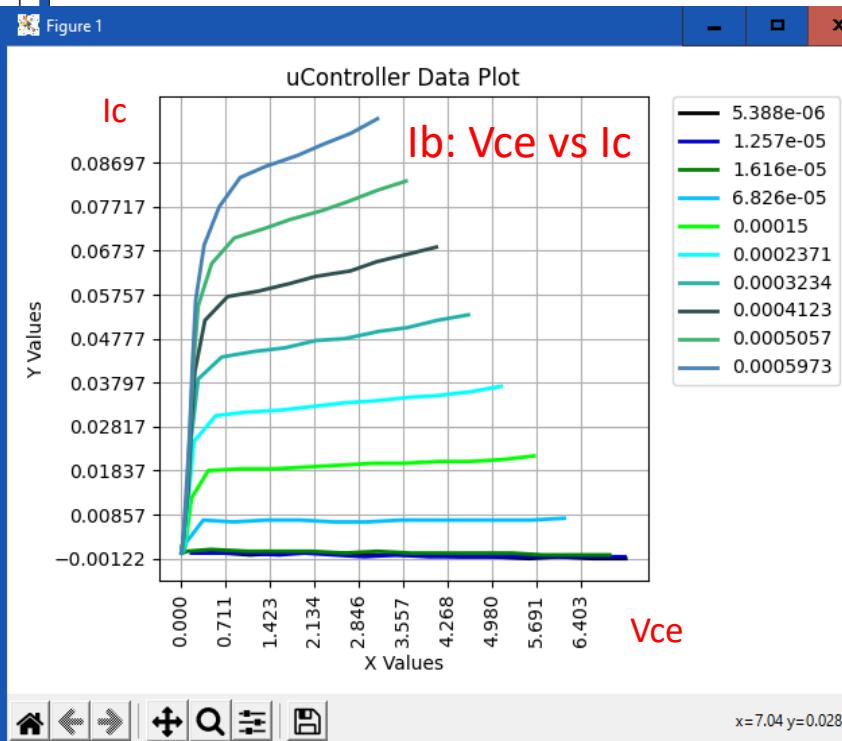
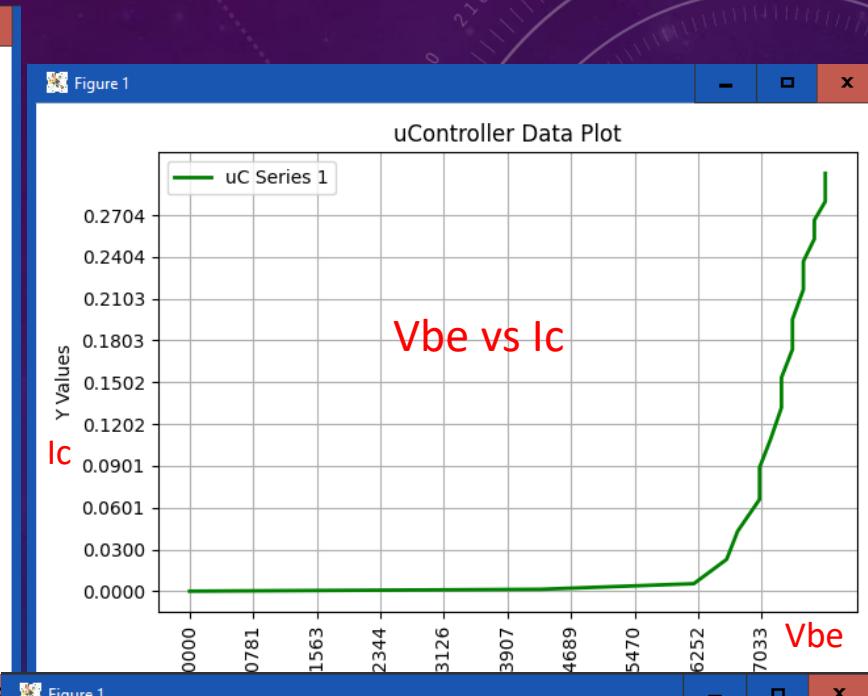
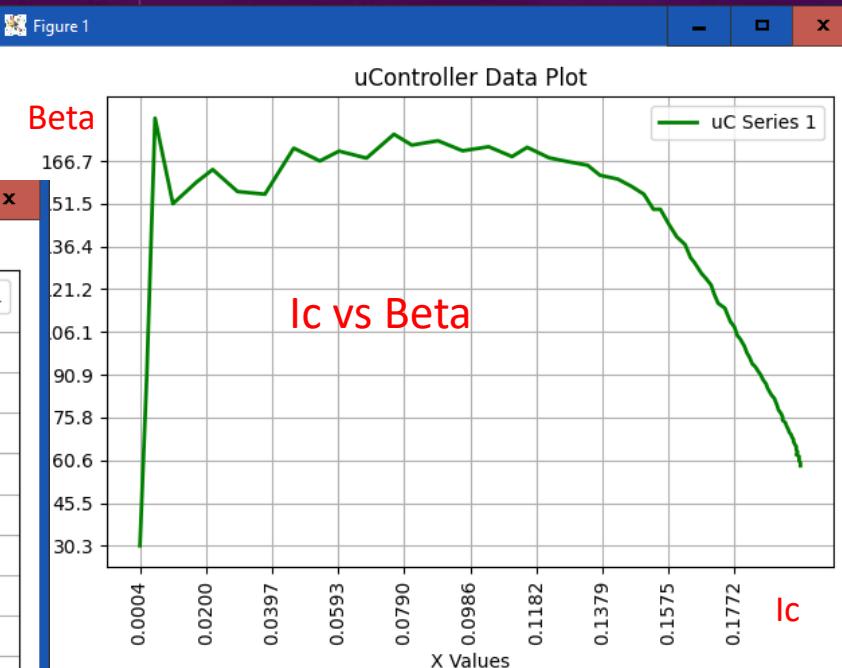
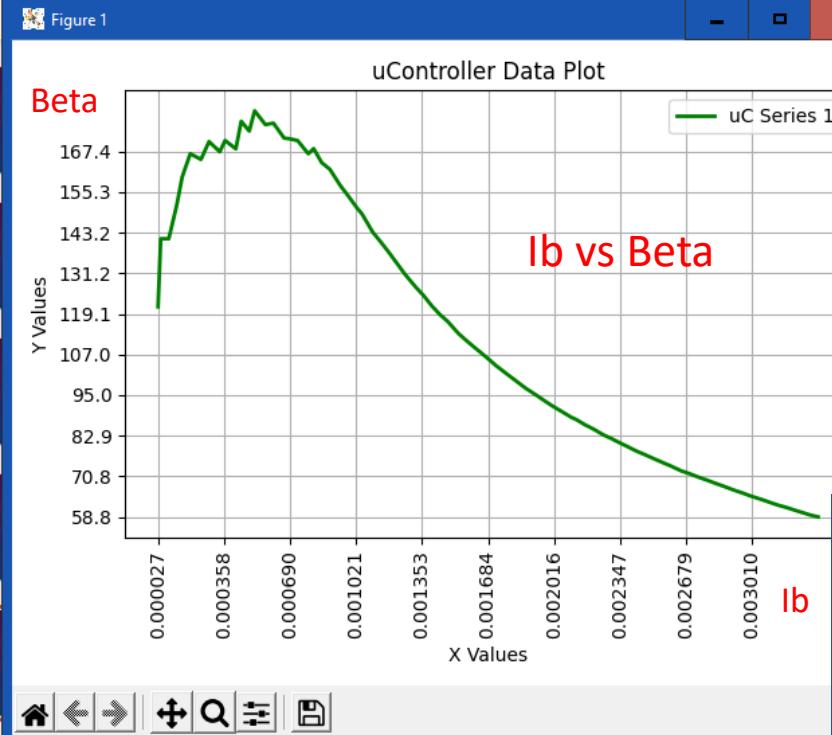
This is a tool that you need to understand what you are doing.

I have only scratched the surface of what this can do AND how to use it

Following charts are based on my limited understanding

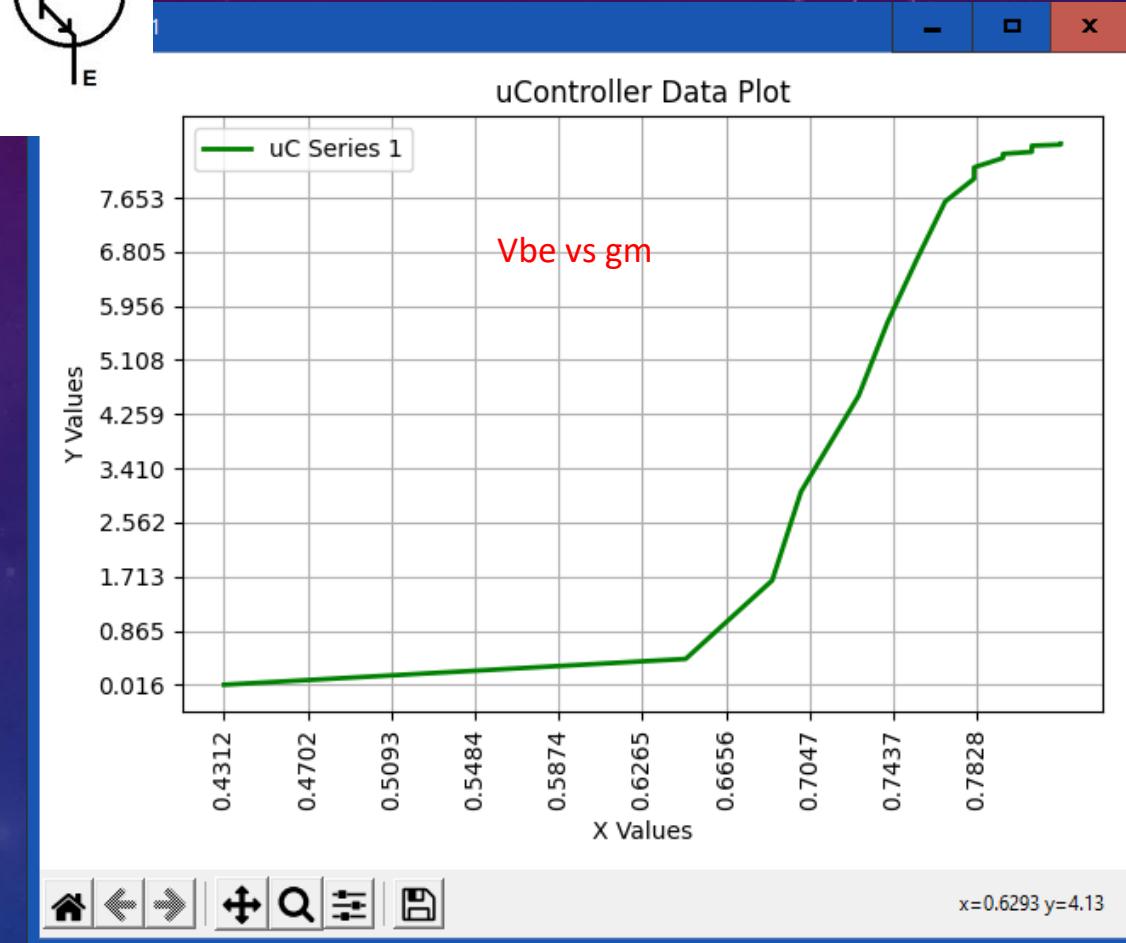
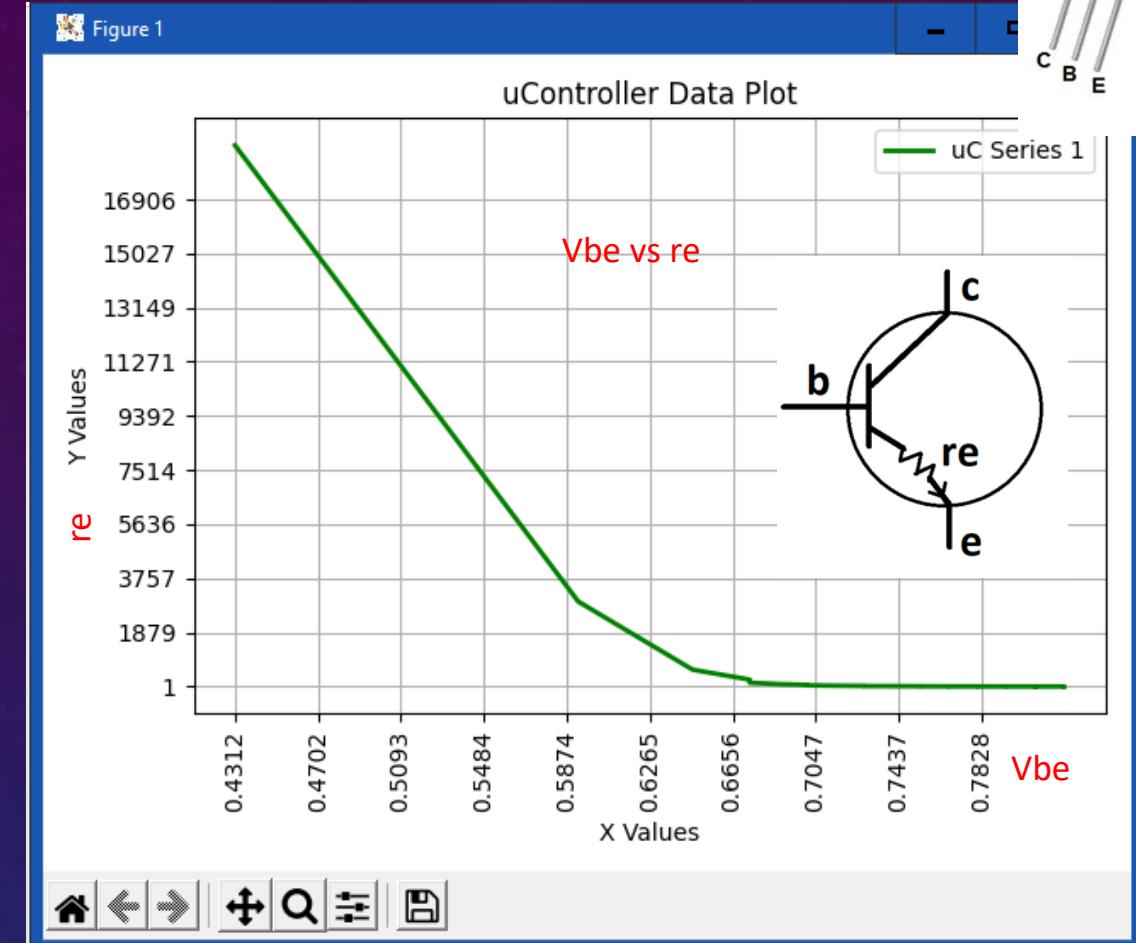


NPN PLOTS



NPN PLOTS

Figure 1



x=0.6293 y=4.13

MEASUREMENTS OF DIODES AND BJT DEVICES

DIODE PLOTS: 1N914

V_f: 1.6168

Figure 1

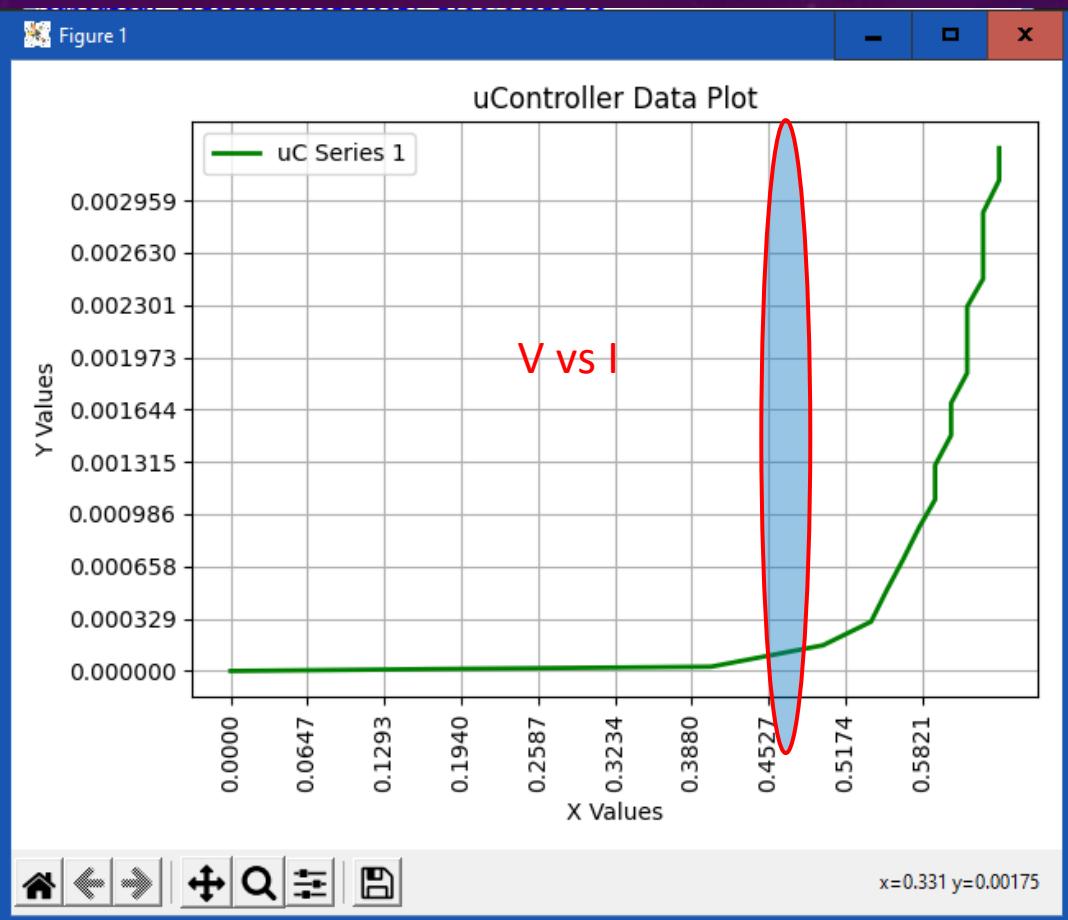
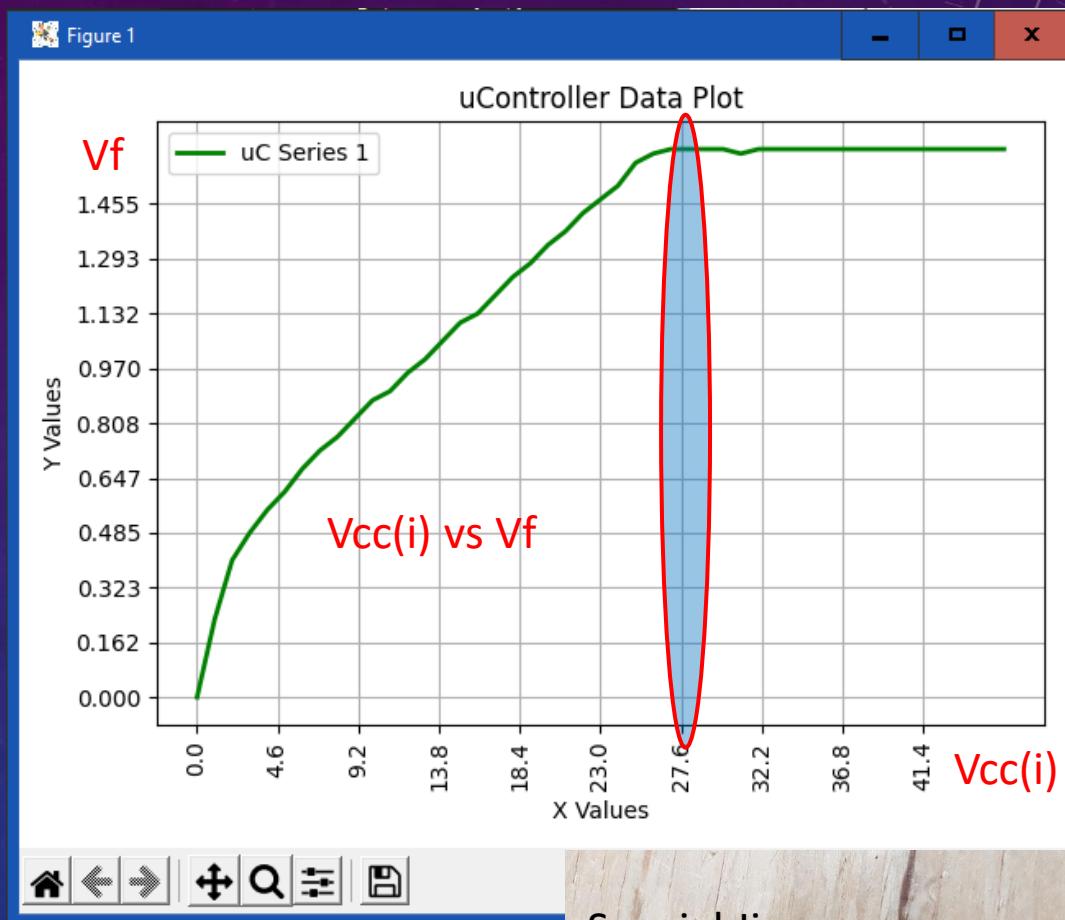


Figure 1

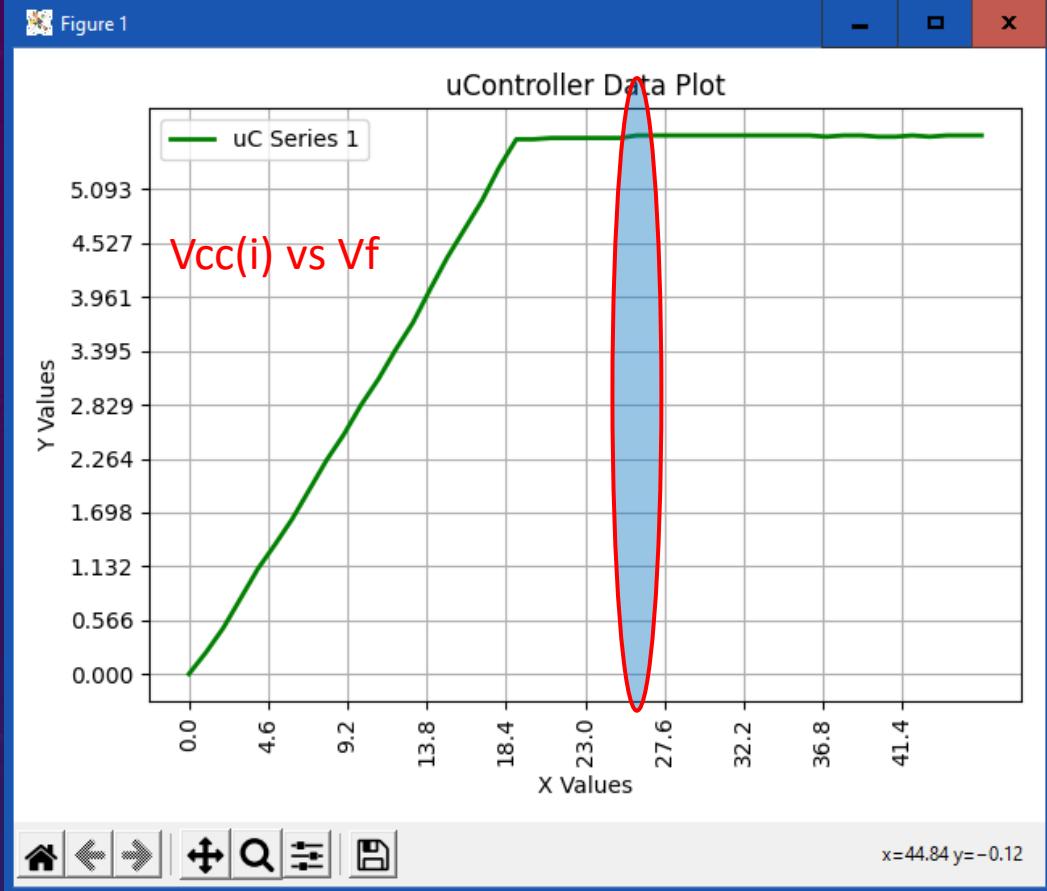


PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Forward voltage	I _F = 10 mA	V _F			1	V
Breakdown voltage	I _R = 100 µA	V _(BR)	100			V
Peak reverse current	V _R = 75 V	I _R			5	µA
	V _R = 20 V, T _j = 150 °C	I _R			50	µA
	V _R = 20 V	I _R			25	nA
Diode capacitance	V _R = 0, f = 1 MHz	C _D			4	pF
Reverse recovery time	I _F = 10 mA, I _R = 1 mA, V _R = 6 V, R _L = 100 Ω	t _{rr}			4	ns

Special Jig
Required



DIODE PLOTS: 1N752 (5.6Z)

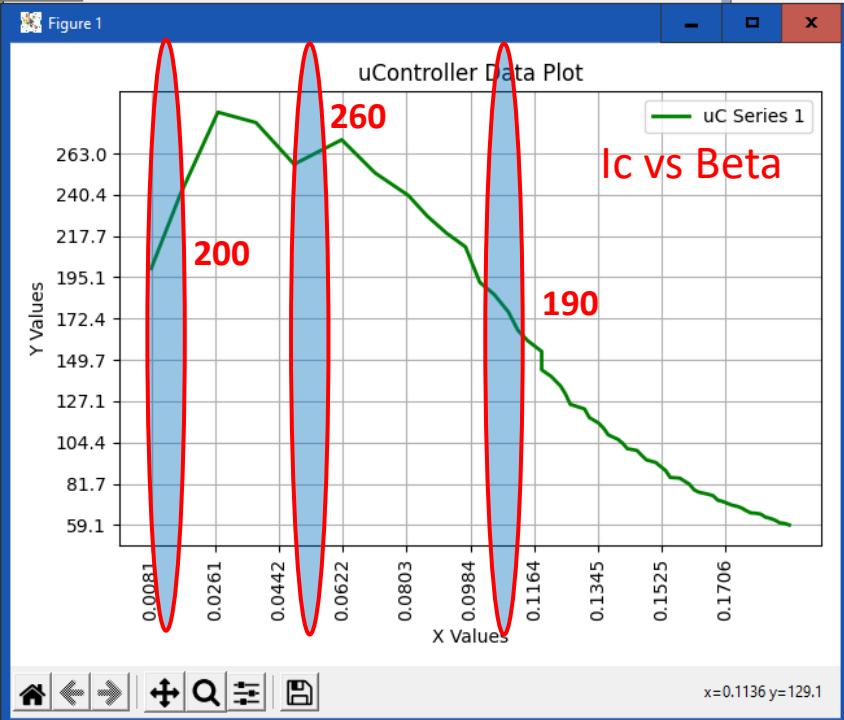


ELECTRICAL CHARACTERISTICS* @ 25°C

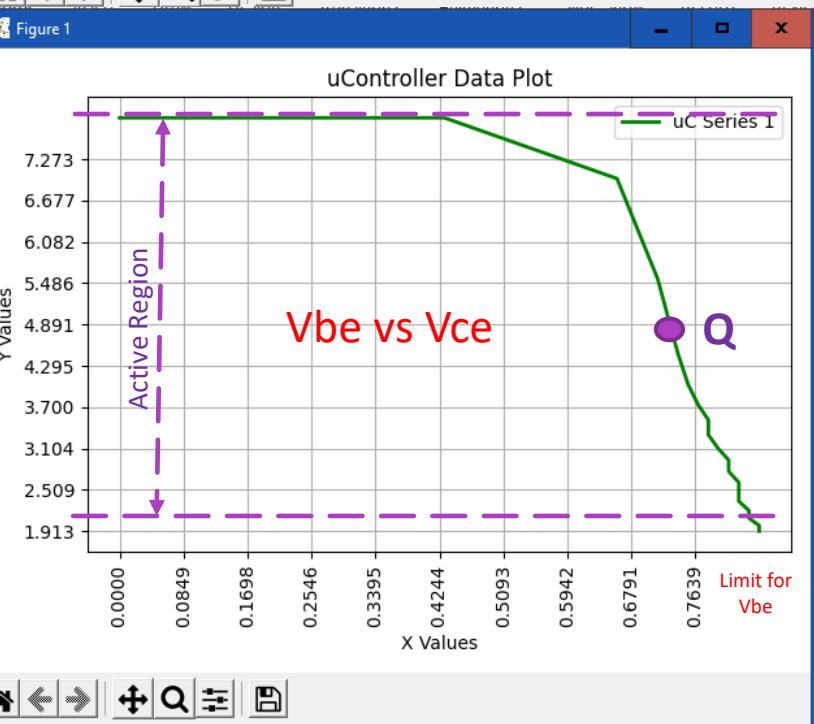
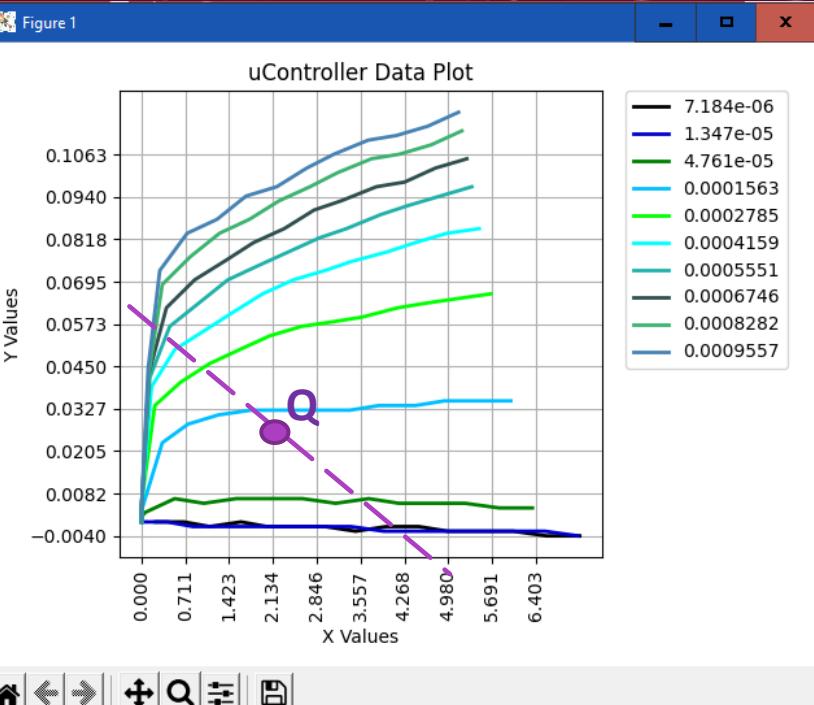
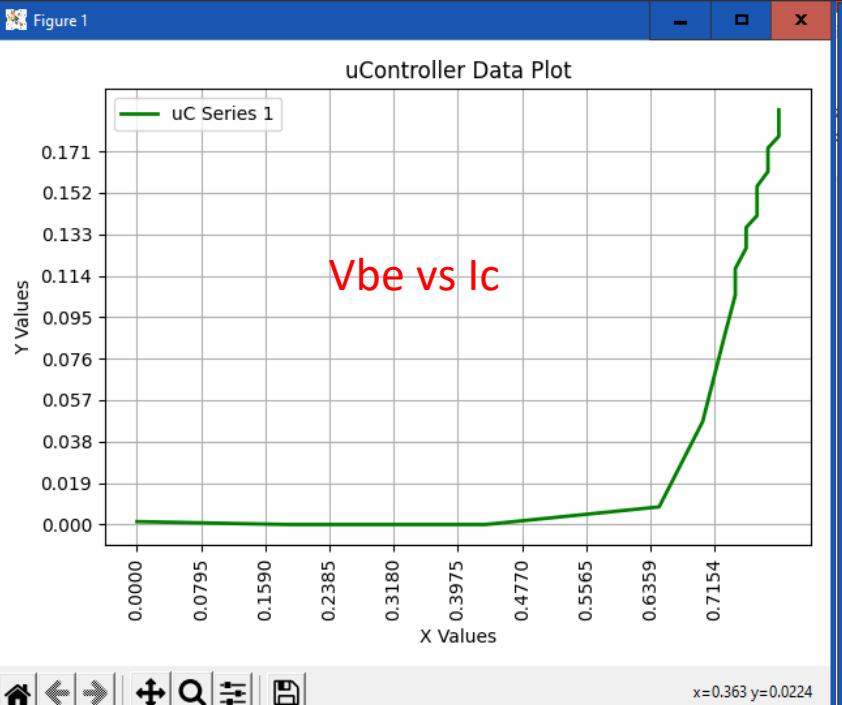
JEDEC TYPE NO. (NOTE1)	NOMINAL ZENER VOLTAGE V _Z @ I _{ZT} (NOTE 2)	ZENER TEST CURRENT I _{ZT}	MAXIMUM ZENER IMPEDANCE Z _{ZT} @ I _{ZT} (NOTE 3)	MAXIMUM REVERSE CURRENT I _R @ V _R = 1 VOLT		MAXIMUM ZENER CURRENT I _{ZM} (NOTE 4)	TYPICAL TEMP COEFF. OF ZENER VOLTAGE α _{VZ}
				VOLTS	mA	OHMS	μA
1N4370	2.4	20	30	100	200	150	-.085
1N4371	2.7	20	30	75	150	135	-.080
1N4372	3.0	20	29	50	100	120	-.075
1N746	3.3	20	28	10	30	110	-.066
1N747	3.6	20	24	10	30	100	-.058
1N748	3.9	20	23	10	30	95	-.046
1N749	4.3	20	22	2	30	85	-.033
1N750	4.7	20	19	2	30	75	-.015
1N751	5.1	20	17	1	20	70	+/-0.10
1N752	5.6	20	11	1	20	65	+.030

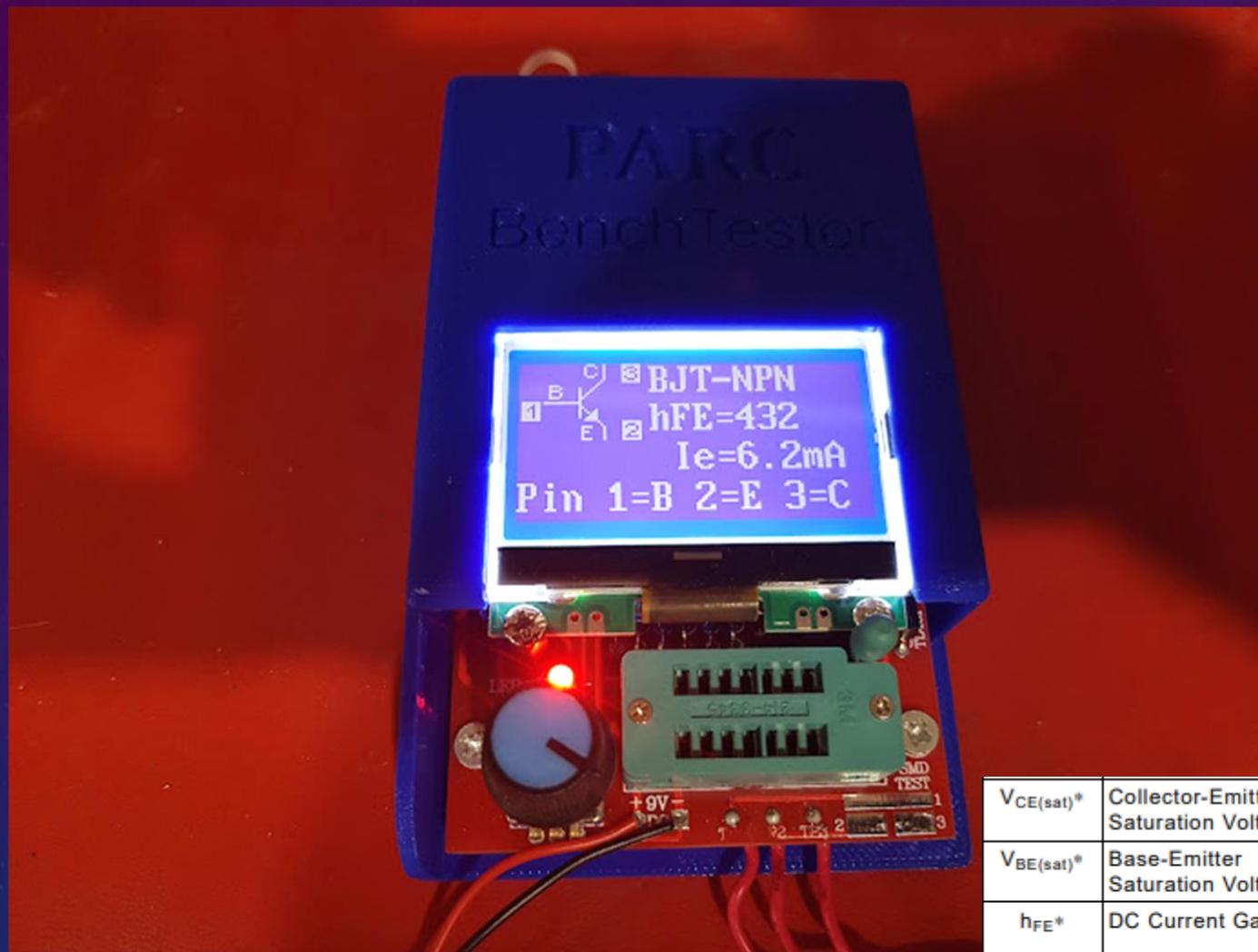


2N3904



$V_{CE(sat)*}$	Collector-Emitter Saturation Voltage	$I_C = 10 \text{ mA}$	$I_B = 1 \text{ mA}$		0.2	V
		$I_C = 50 \text{ mA}$	$I_B = 5 \text{ mA}$		0.2	V
$V_{BE(sat)*}$	Base-Emitter Saturation Voltage	$I_C = 10 \text{ mA}$	$I_B = 1 \text{ mA}$	0.65	0.85	V
		$I_C = 50 \text{ mA}$	$I_B = 5 \text{ mA}$		0.95	V
h_{FE}^*	DC Current Gain	$I_C = 0.1 \text{ mA}$	$V_{CE} = 1 \text{ V}$	60		
		$I_C = 1 \text{ mA}$	$V_{CE} = 1 \text{ V}$	80		
		$I_C = 10 \text{ mA}$	$V_{CE} = 1 \text{ V}$	100		
		$I_C = 50 \text{ mA}$	$V_{CE} = 1 \text{ V}$	60	300	
		$I_C = 100 \text{ mA}$	$V_{CE} = 1 \text{ V}$	30		

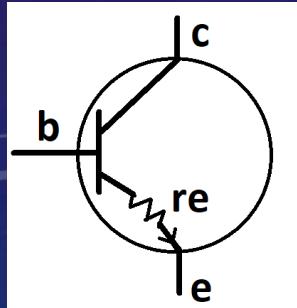
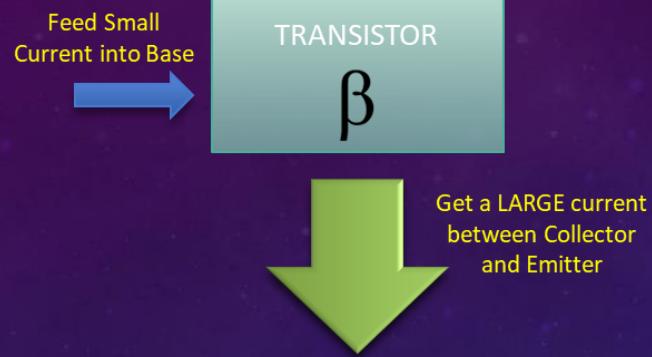




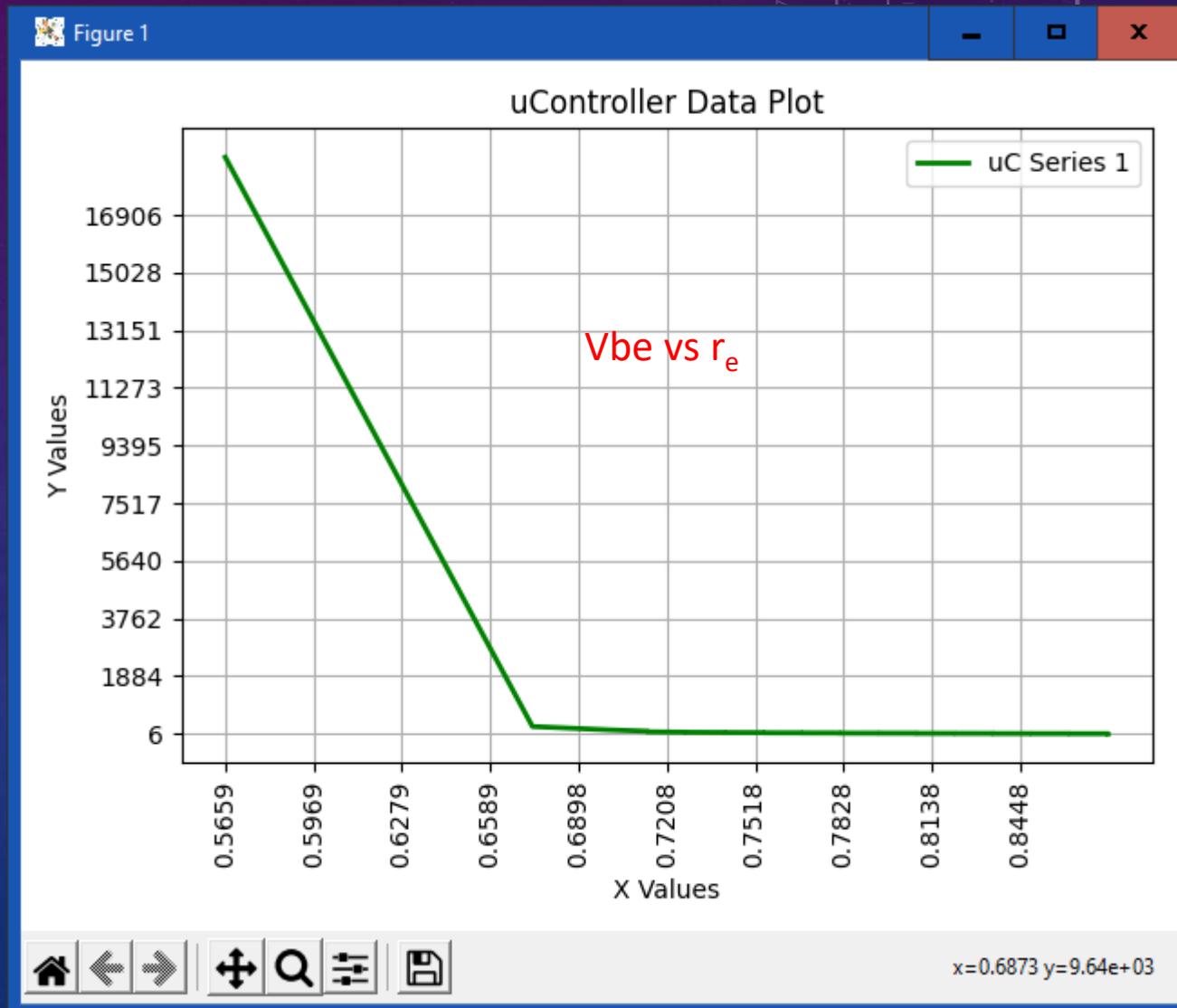
432 Really?!

$V_{CE(sat)}^*$	Collector-Emitter Saturation Voltage	$I_C = 10 \text{ mA}$ $I_C = 50 \text{ mA}$	$I_B = 1 \text{ mA}$ $I_B = 5 \text{ mA}$			0.2	V
$V_{BE(sat)}^*$	Base-Emitter Saturation Voltage	$I_C = 10 \text{ mA}$ $I_C = 50 \text{ mA}$	$I_B = 1 \text{ mA}$ $I_B = 5 \text{ mA}$	0.65		0.85 0.95	V
h_{FE}^*	DC Current Gain	$I_C = 0.1 \text{ mA}$ $I_C = 1 \text{ mA}$ $I_C = 10 \text{ mA}$ $I_C = 50 \text{ mA}$ $I_C = 100 \text{ mA}$	$V_{CE} = 1 \text{ V}$ $V_{CE} = 1 \text{ V}$ $V_{CE} = 1 \text{ V}$ $V_{CE} = 1 \text{ V}$ $V_{CE} = 1 \text{ V}$	60 80 100 60 30		300	

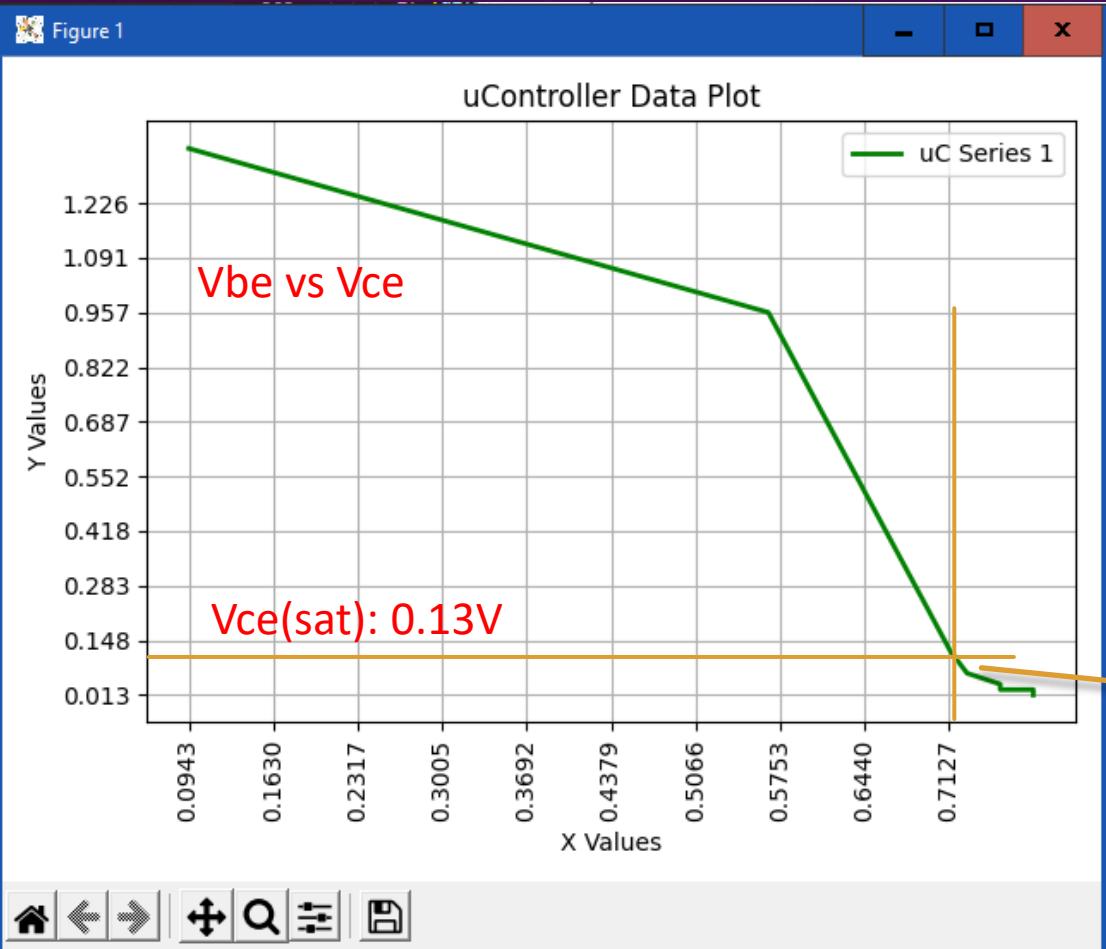
AMENDMENT (TIME TRAVEL) – NEW PLOT



r_e is like a base controlled resistor that controls current between collector to emitter



AMENDMENT (TIME TRAVEL)



Previous trace with V_{cc} set to ~8V
This trace with V_{cc} set to about ~1.5V.



V _{CE(sat)*}	Collector-Emitter Saturation Voltage	I _C = 10 mA	I _B = 1 mA	0.2	V
		I _C = 50 mA	I _B = 5 mA	0.2	V
V _{BE(sat)*}	Base-Emitter Saturation Voltage	I _C = 10 mA	I _B = 1 mA	0.65	V
		I _C = 50 mA	I _B = 5 mA	0.85	V
h _{FE*}	DC Current Gain	I _C = 0.1 mA	V _{CE} = 1 V	300	
		I _C = 1 mA	V _{CE} = 1 V	60	
		I _C = 10 mA	V _{CE} = 1 V	80	
		I _C = 50 mA	V _{CE} = 1 V	100	
		I _C = 100 mA	V _{CE} = 1 V	60	
				30	

AMENDMENT (TIME TRAVEL)

```
:> NPN Beta/Re/Gm Sweep
Sweep Vb/Vg: 0.14 - 4.60 points: 20 delta: 0.22
Fixed Vc/Vd: 3.71
Data format error. Skipping capturing line: Ic,beta
Captured: 0.009431362152, 161.69908142
Captured: 0.029641532897, 254.0994873
Captured: 0.055241203308, 315.70007324
Captured: 0.078146076202, 297.73342895
```

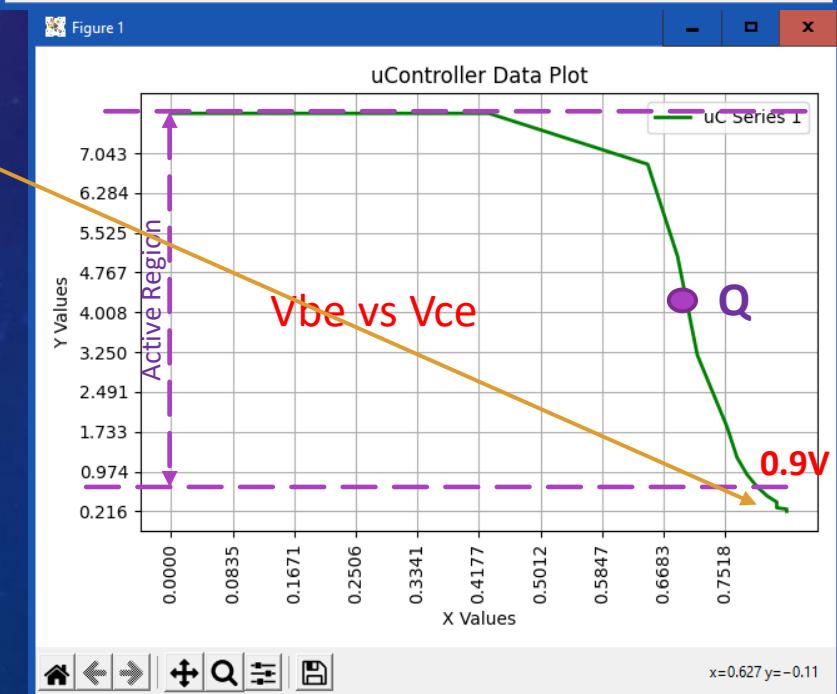
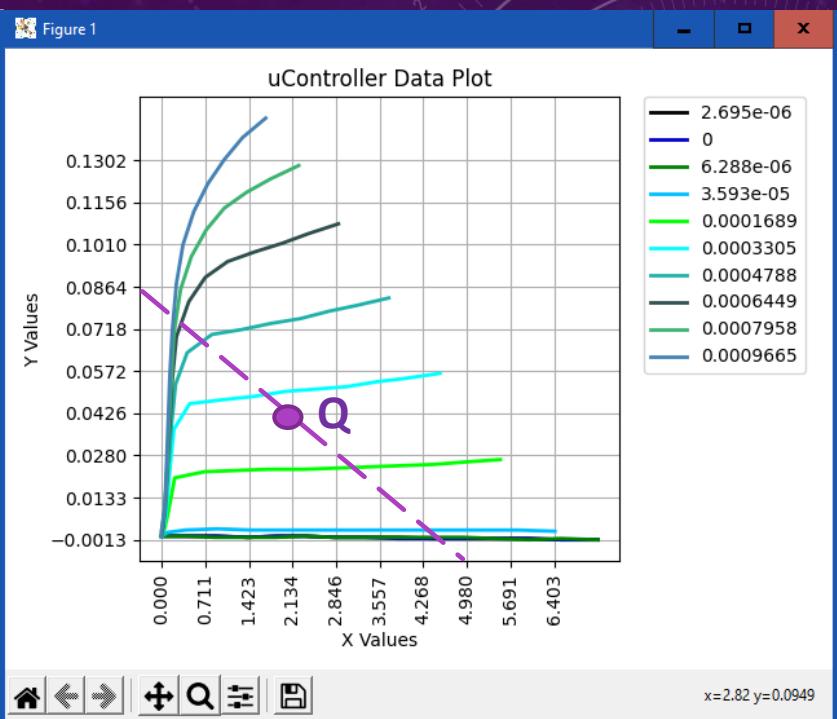
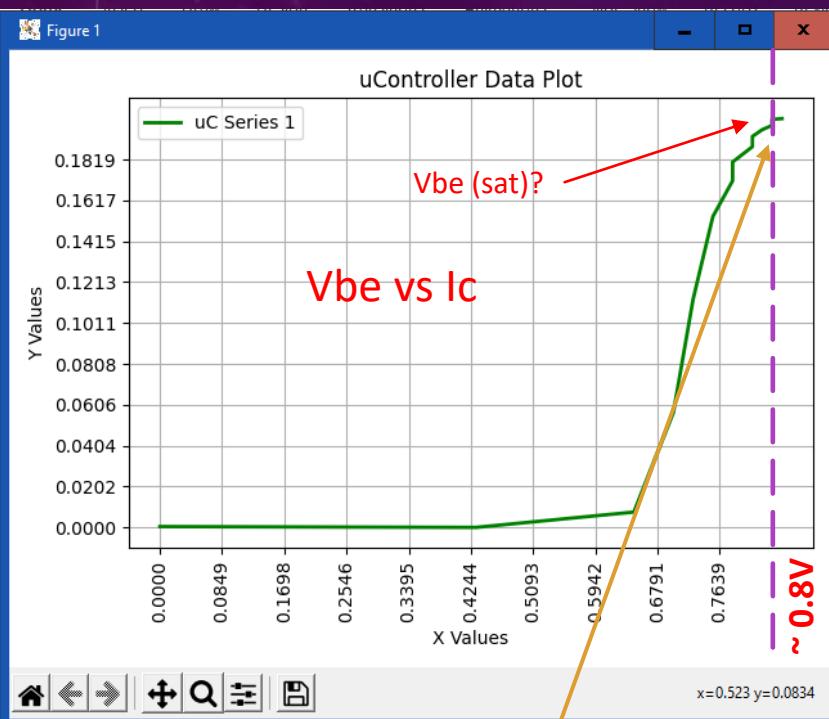
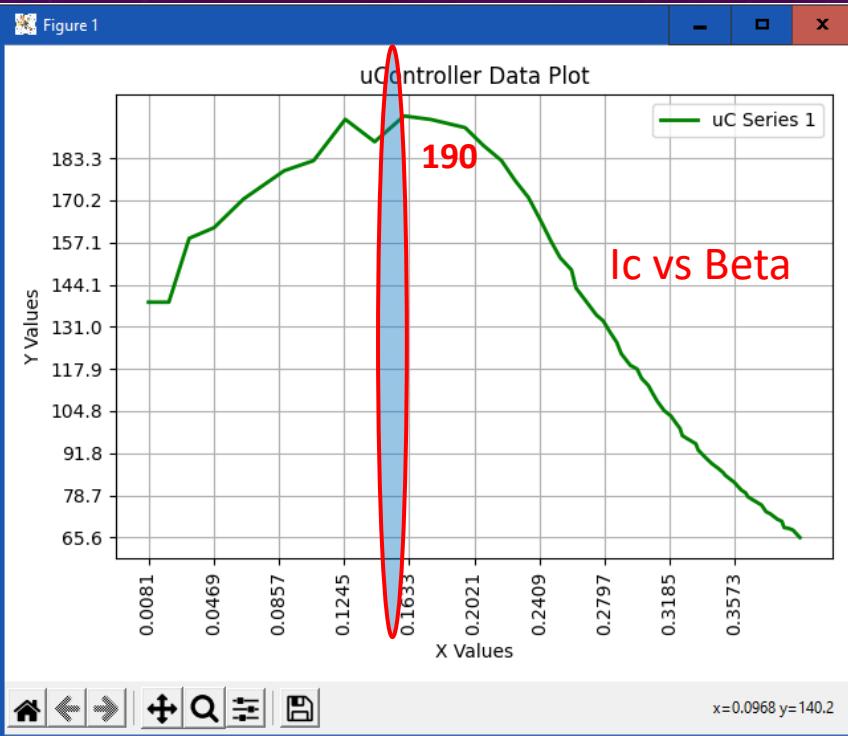
Bottom Line:

- ✓ Need to fiddle with ranges for sweep
- ✓ Also help to understand what you are doing!!

Notice the steps.
Datasheet Beta at 10mA
With default sweep jump about 20mA!!

$V_{CE(sat)}^*$	Collector-Emitter Saturation Voltage	$I_C = 10 \text{ mA}$ $I_C = 50 \text{ mA}$	$I_B = 1 \text{ mA}$ $I_B = 5 \text{ mA}$			0.2 0.2	V V
$V_{BE(sat)}^*$	Base-Emitter Saturation Voltage	$I_C = 10 \text{ mA}$ $I_C = 50 \text{ mA}$	$I_B = 1 \text{ mA}$ $I_B = 5 \text{ mA}$	0.65		0.85 0.95	V V
h_{FE}^*	DC Current Gain	$I_C = 0.1 \text{ mA}$ $I_C = 1 \text{ mA}$ $I_C = 10 \text{ mA}$ $I_C = 50 \text{ mA}$ $I_C = 100 \text{ mA}$	$V_{CE} = 1 \text{ V}$ $V_{CE} = 1 \text{ V}$ $V_{CE} = 1 \text{ V}$ $V_{CE} = 1 \text{ V}$ $V_{CE} = 1 \text{ V}$	60 80 100 60 30		300	

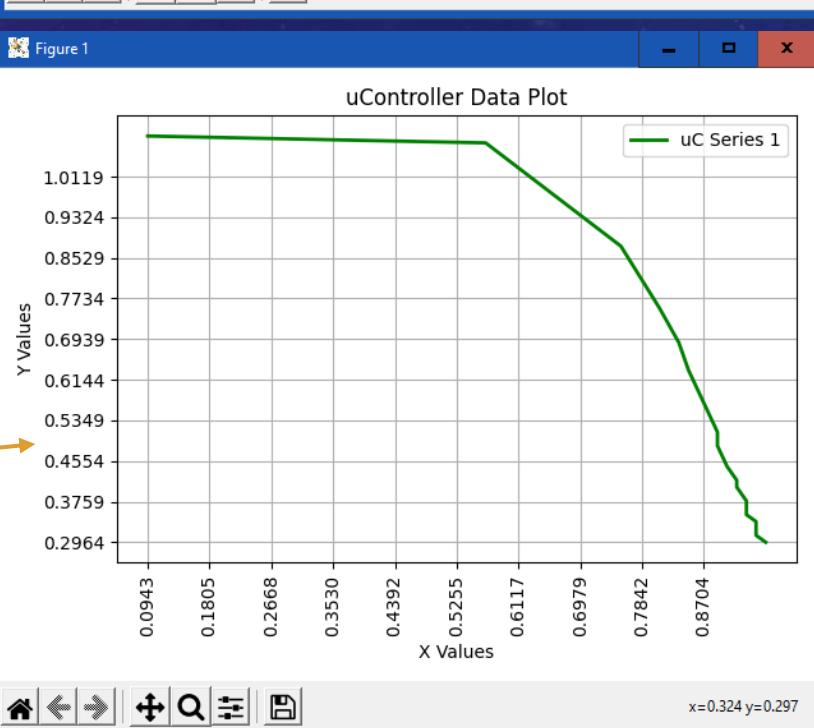
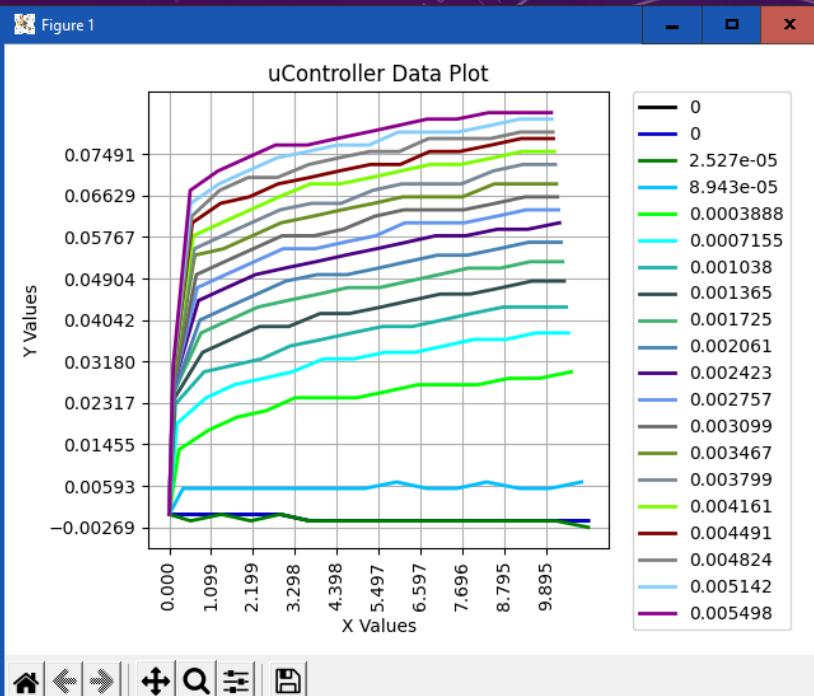
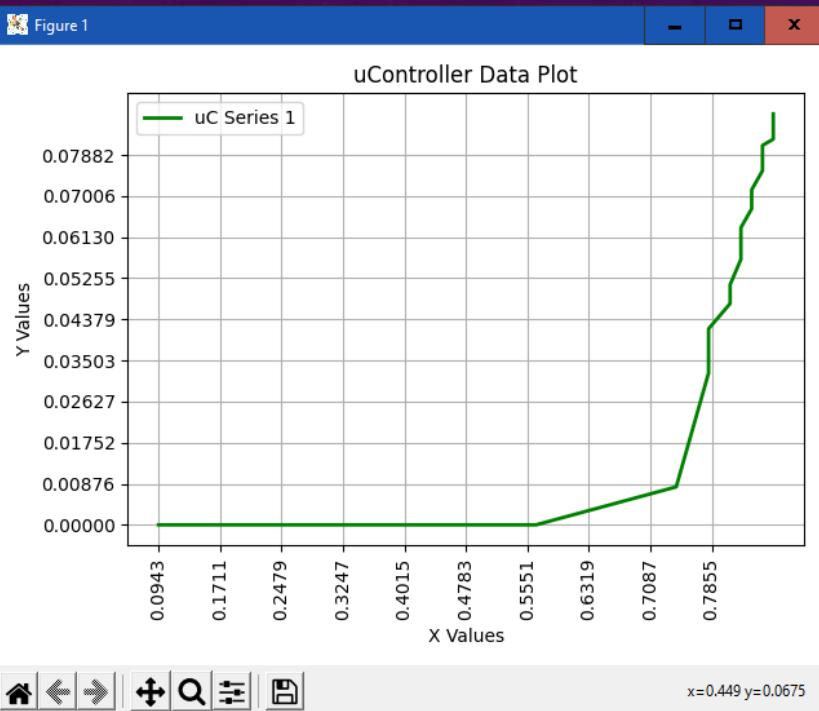
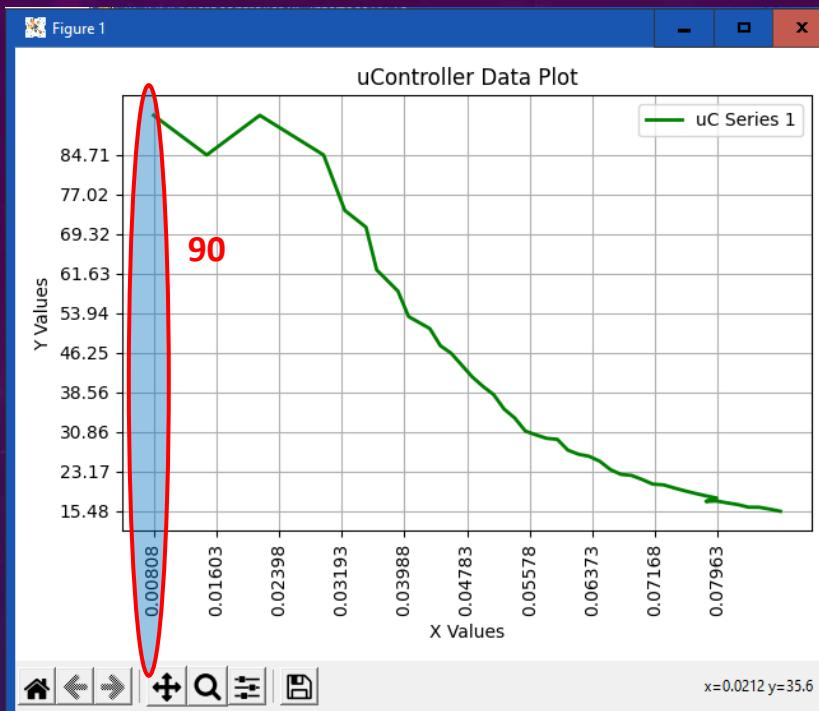
2N2222A



Obsolete Product

$V_{CE(sat)}^*$	Collector-Emitter Saturation Voltage	I _c = 150 mA I _b = 15 mA	I _c = 500 mA I _b = 50 mA			0.3	V
$V_{BE(sat)}^*$	Base-Emitter Saturation Voltage	I _c = 150 mA I _b = 15 mA	I _c = 500 mA I _b = 50 mA	0.6		1.2	V
h_{FE}^*	DC Current Gain	I _c = 0.1 mA V _{CE} = 10 V	I _c = 1 mA V _{CE} = 10 V	35		2	V
		I _c = 10 mA V _{CE} = 10 V	I _c = 150 mA V _{CE} = 10 V	50			
		I _c = 150 mA V _{CE} = 10 V	I _c = 500 mA V _{CE} = 10 V	75			
		I _c = 150 mA V _{CE} = 10 V	I _c = 150 mA V _{CE} = 1 V	100	300		
		I _c = 500 mA V _{CE} = 10 V	I _c = 10 mA V _{CE} = 10 V	40			
		I _c = 150 mA V _{CE} = 1 V	I _c = 10 mA V _{CE} = 10 V	50			
		T _{amb} = -55 °C		35			

MPSH10



DC Current Gain
($I_C = 4.0 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$)

Collector-Emitter Saturation Voltage
($I_C = 4.0 \text{ mA}$, $I_B = 0.4 \text{ mA}$)

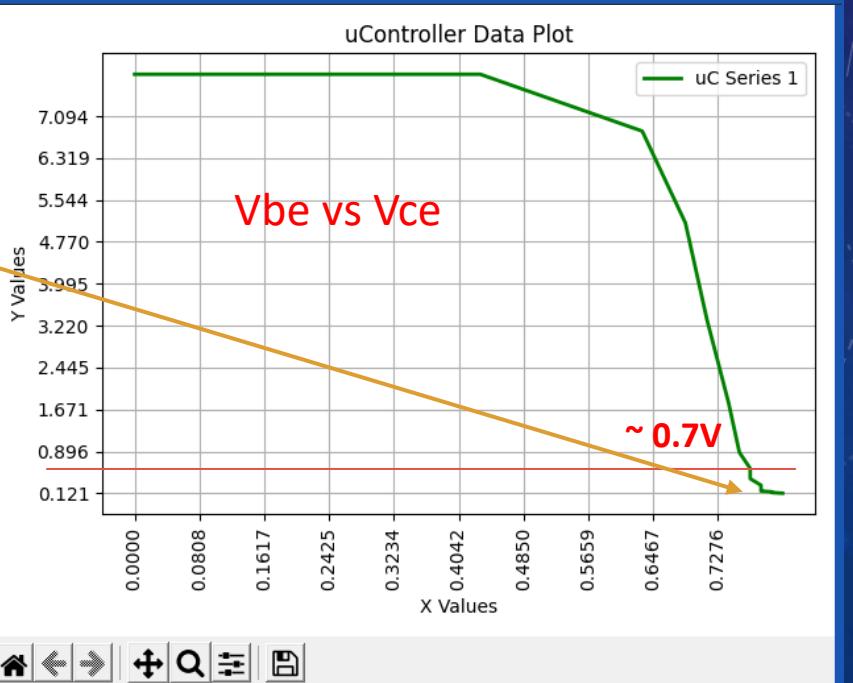
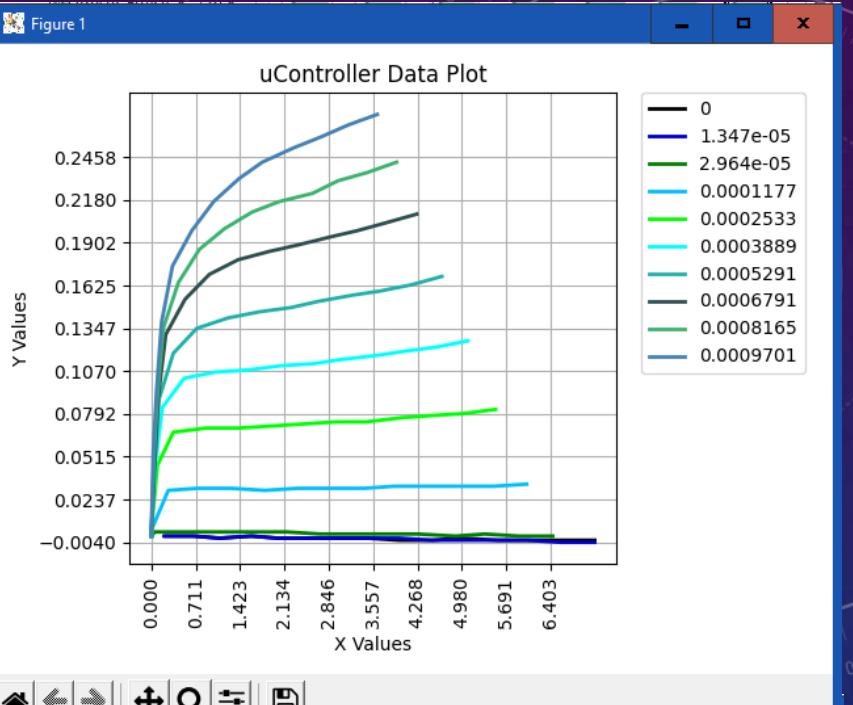
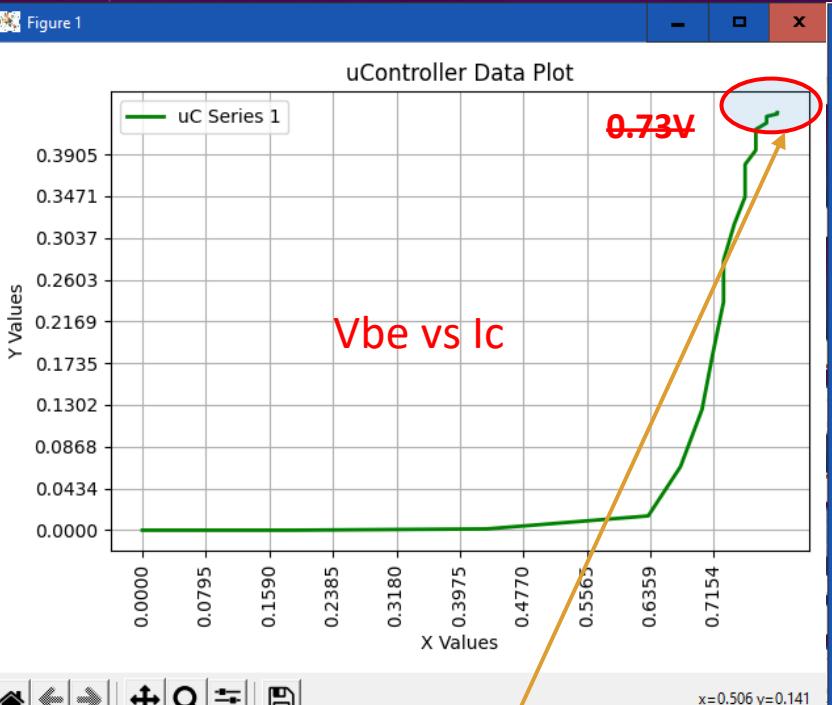
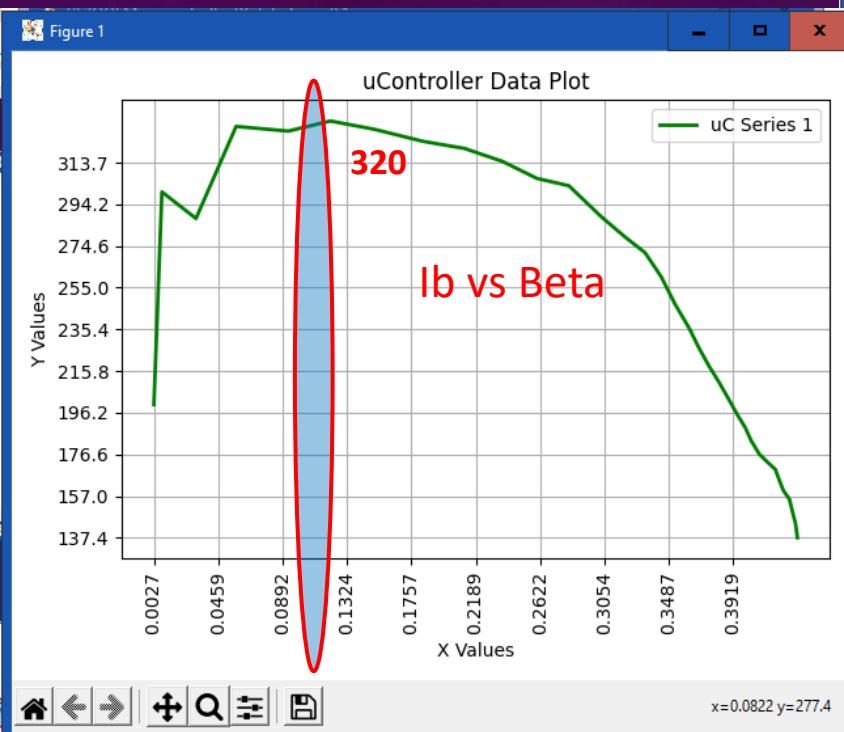
Base-Emitter On Voltage
($I_C = 4.0 \text{ mA}$, $V_{CE} = 10 \text{ Vdc}$)

MINIMUM

MAXIMUM

Symbol	Min	Max	Unit
h_{FE}	60	-	-
$V_{CE(\text{sat})}$	-	0.5	Vdc
$V_{BE(\text{on})}$	-	0.95	Vdc

BC337



DC Current Gain
(I_C = 100 mA, V_{CE} = 1.0 V)

BC337
BC337-25
BC337-40

h_{FE}

	100	-	630	-
100	-	-	630	-
160	-	-	400	-
250	-	-	630	-
60	-	-	-	-

Base-Emitter On Voltage
(I_C = 300 mA, V_{CE} = 1.0 V)

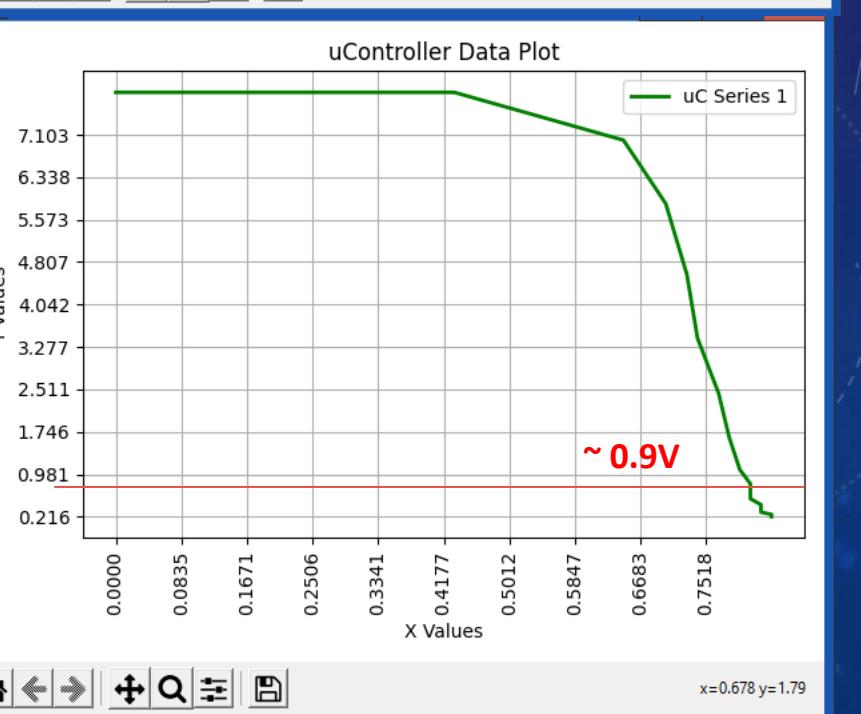
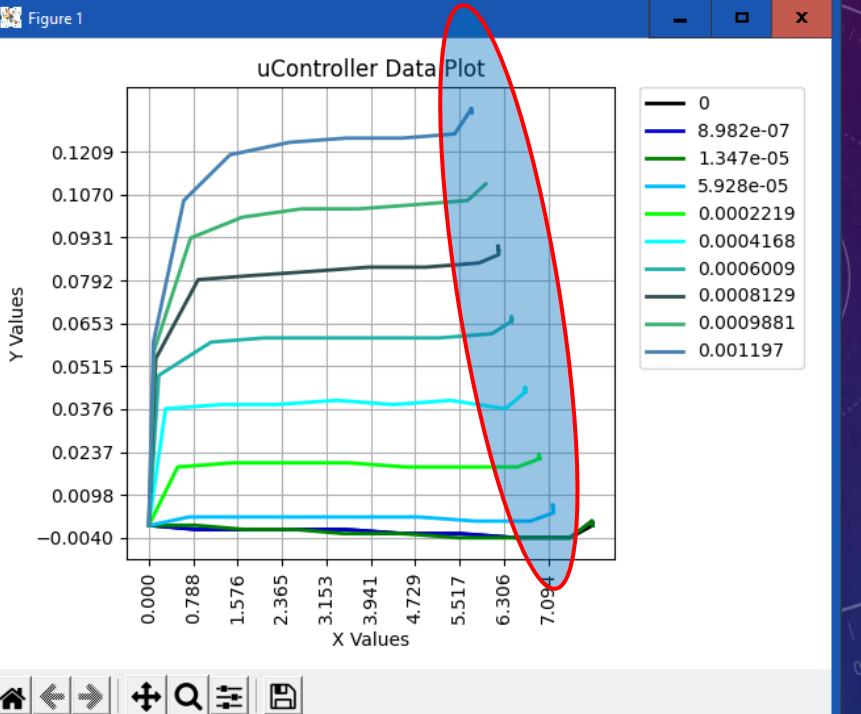
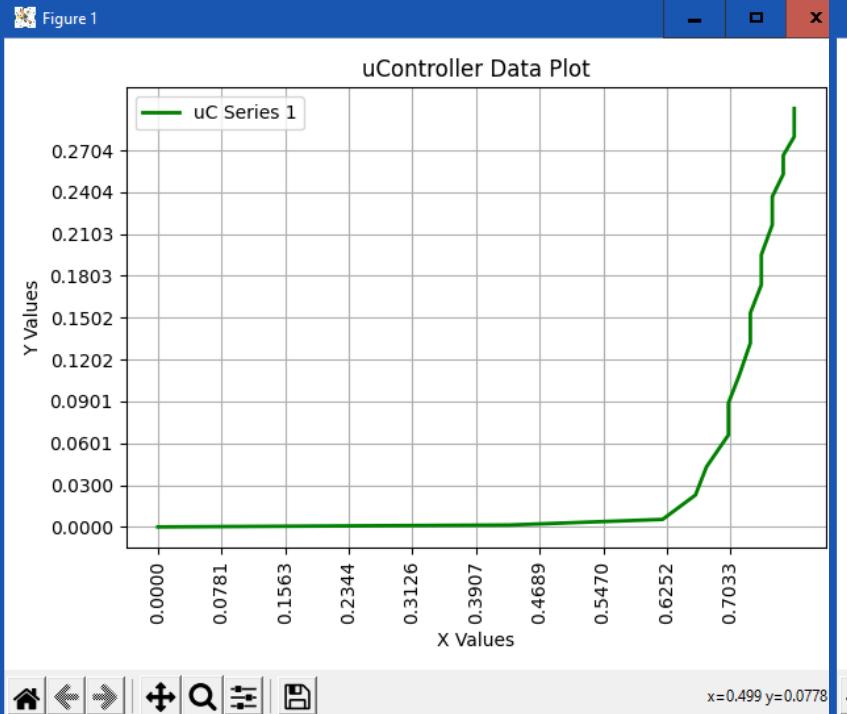
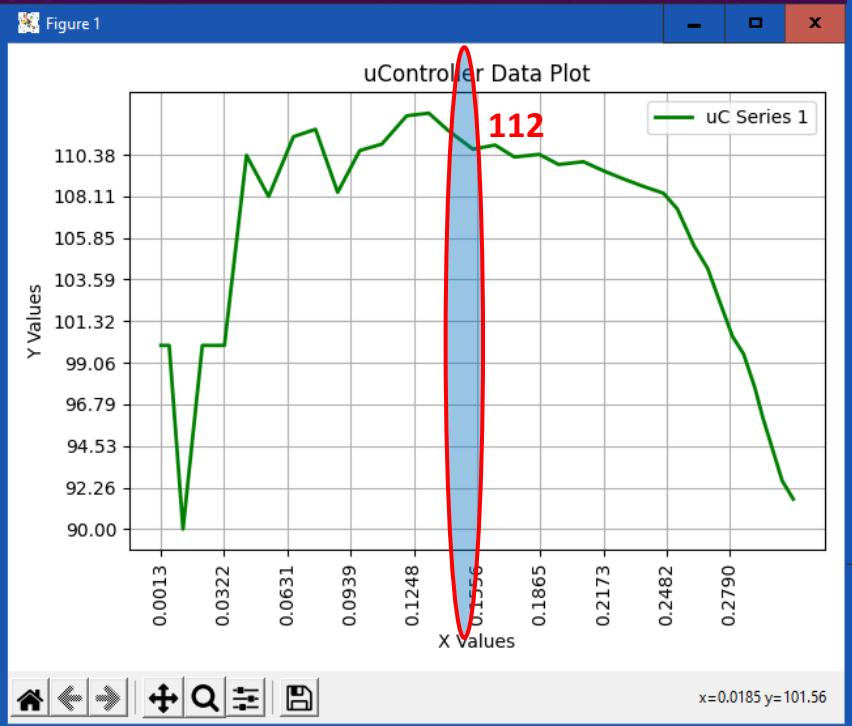
V_{BE(on)}

Collector-Emitter Saturation Voltage
(I_C = 500 mA, I_B = 50 mA)

V_{CE(sat)}

	-	-	1.2	V _{dc}
V _{BE(on)}	-	-	1.2	V _{dc}
V _{CE(sat)}	-	-	0.7	V _{dc}

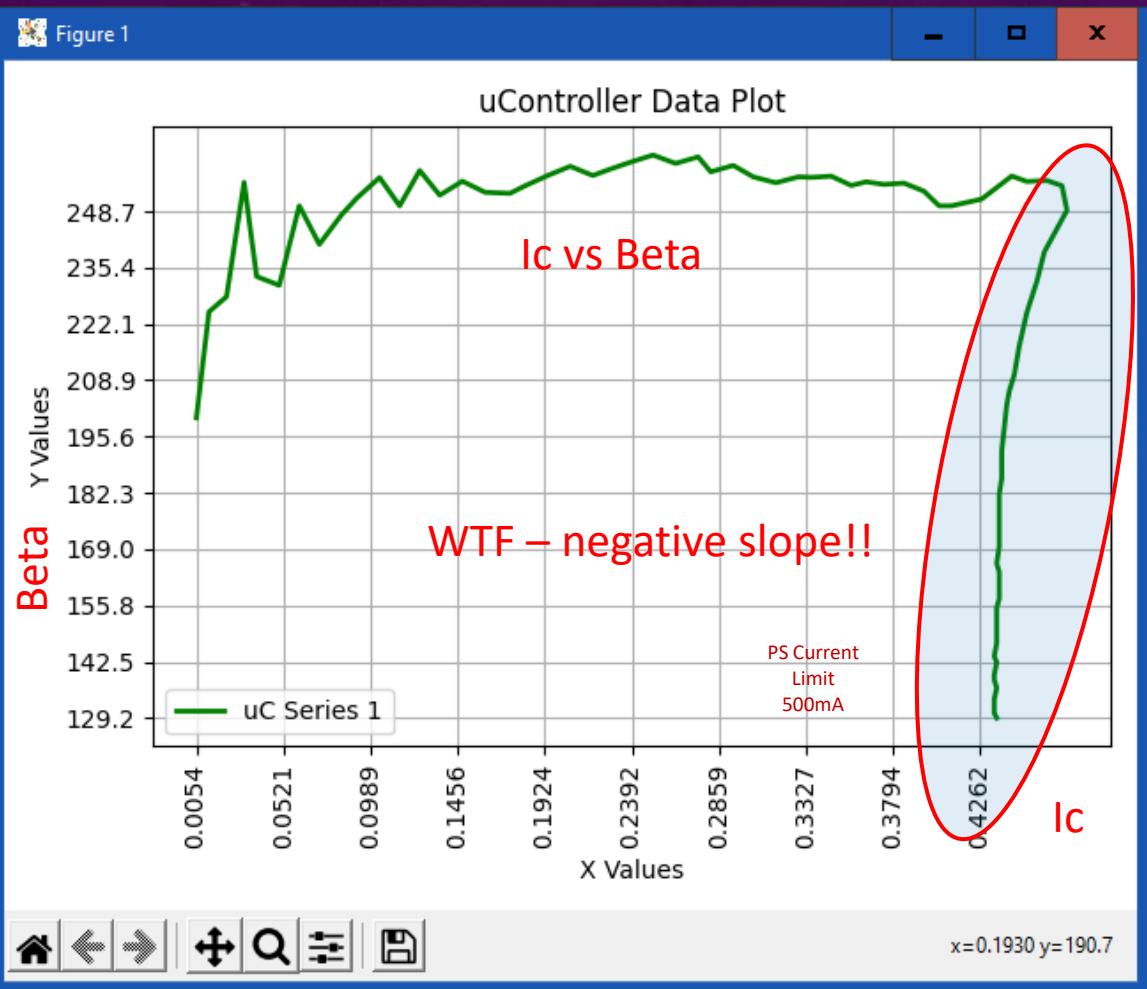
2N2219A



$V_{CE(sat)*}$	Collector-Emitter Saturation Voltage	$I_C = 150 \text{ mA}$ $I_B = 15 \text{ mA}$			0.3	V
		$I_C = 500 \text{ mA}$ $I_B = 50 \text{ mA}$			1	V
$V_{BE(sat)*}$	Base-Emitter Saturation Voltage	$I_C = 150 \text{ mA}$ $I_B = 15 \text{ mA}$	0.6		1.2	V
		$I_C = 500 \text{ mA}$ $I_B = 50 \text{ mA}$			2	V
h_{FE}^*	DC Current Gain	$I_C = 0.1 \text{ mA}$ $V_{CE} = 10 \text{ V}$	35			
		$I_C = 1 \text{ mA}$ $V_{CE} = 10 \text{ V}$	50			
		$I_C = 10 \text{ mA}$ $V_{CE} = 10 \text{ V}$	75			
		$I_C = 150 \text{ mA}$ $V_{CE} = 10 \text{ V}$	100		300	
		$I_C = 500 \text{ mA}$ $V_{CE} = 10 \text{ V}$	40			
		$I_C = 150 \text{ mA}$ $V_{CE} = 1 \text{ V}$	50			
		$I_C = 10 \text{ mA}$ $V_{CE} = 10 \text{ V}$	35			
		$T_{amb} = -55 \text{ }^{\circ}\text{C}$				

BD139

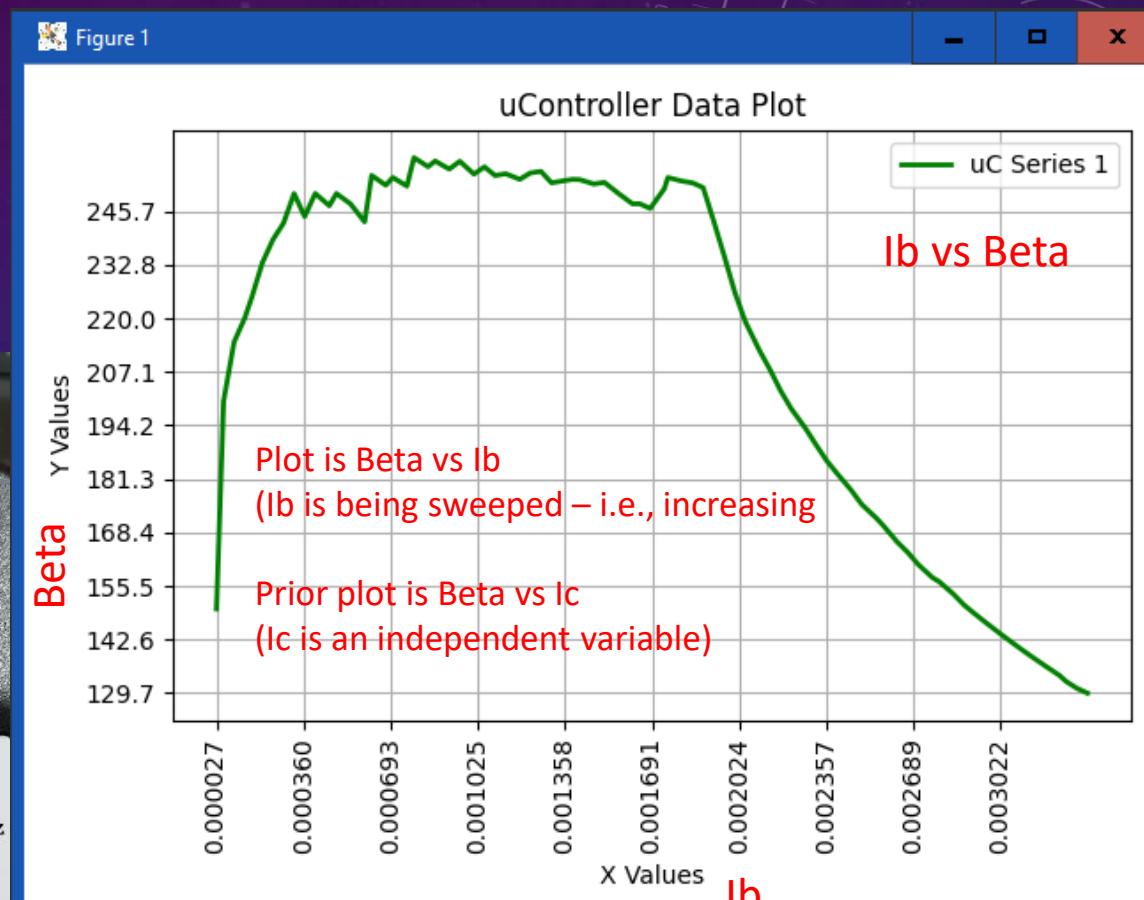
Figure 1



AC ADAPTER
A-88
AC100-240V 50-60Hz
DC10V ==500mA
CHARGING (RED)
COMPLETED (GREEN)

You really need to understand what you are doing

Figure 1



BD139

 Figure

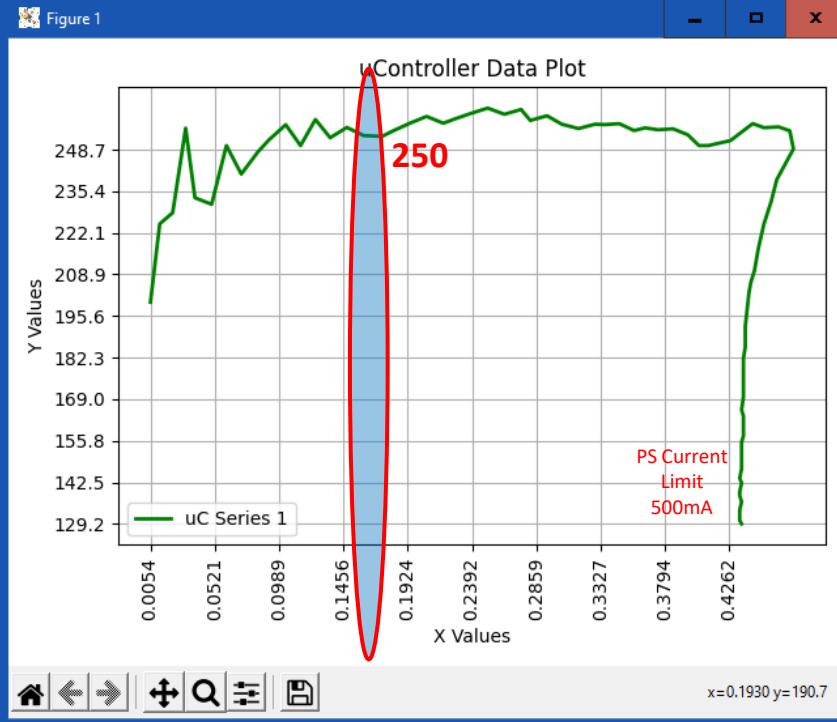


 Figure 1

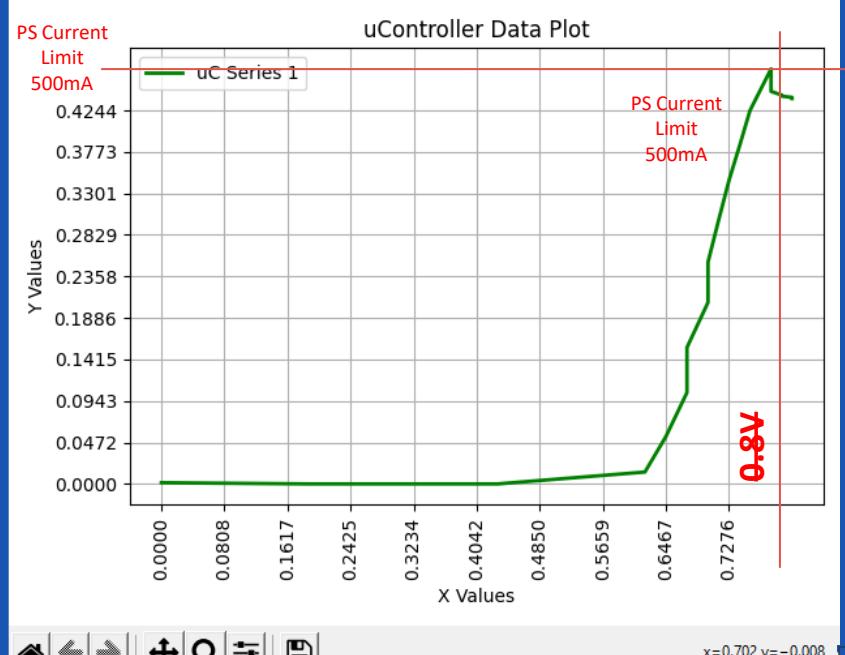


 Figure 1

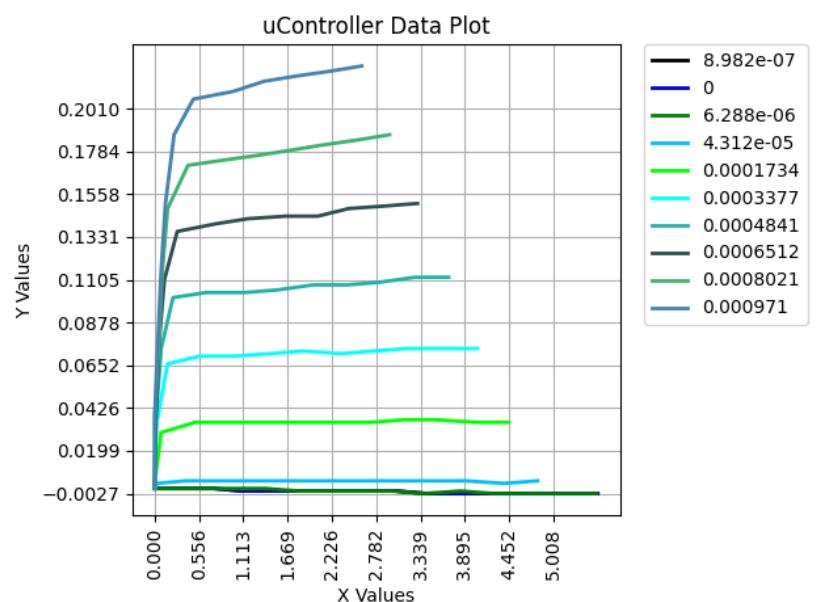
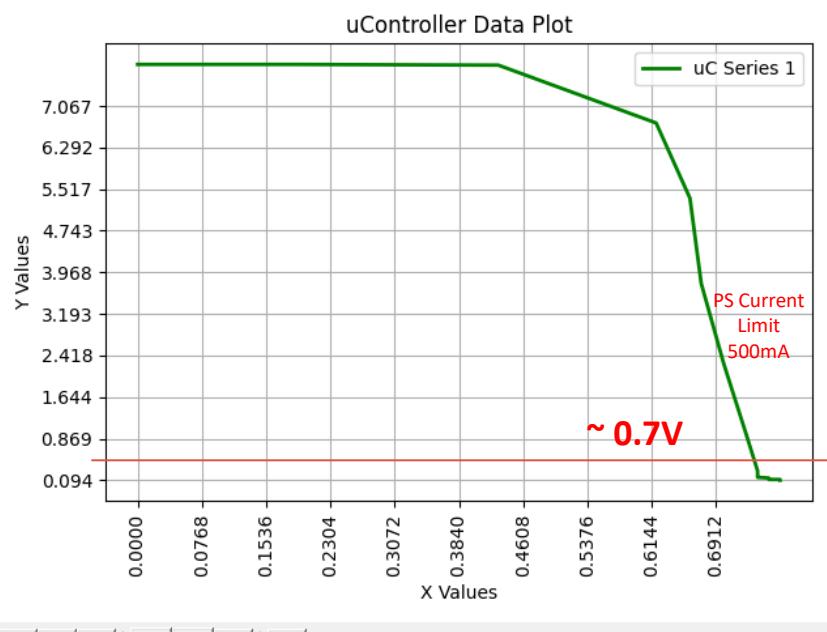
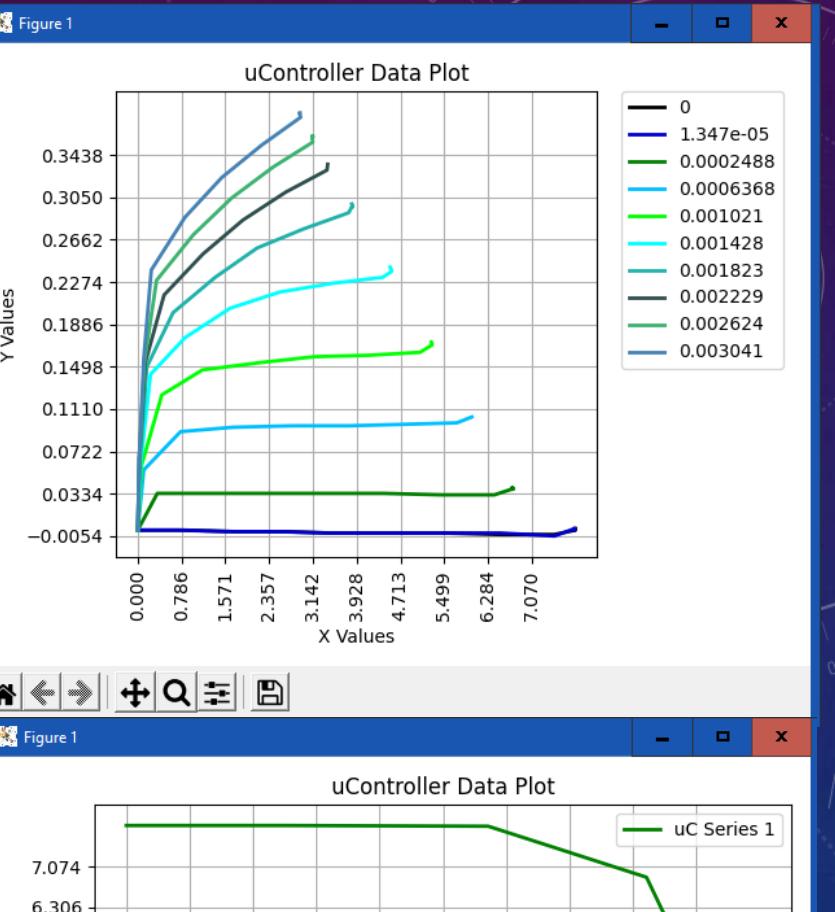
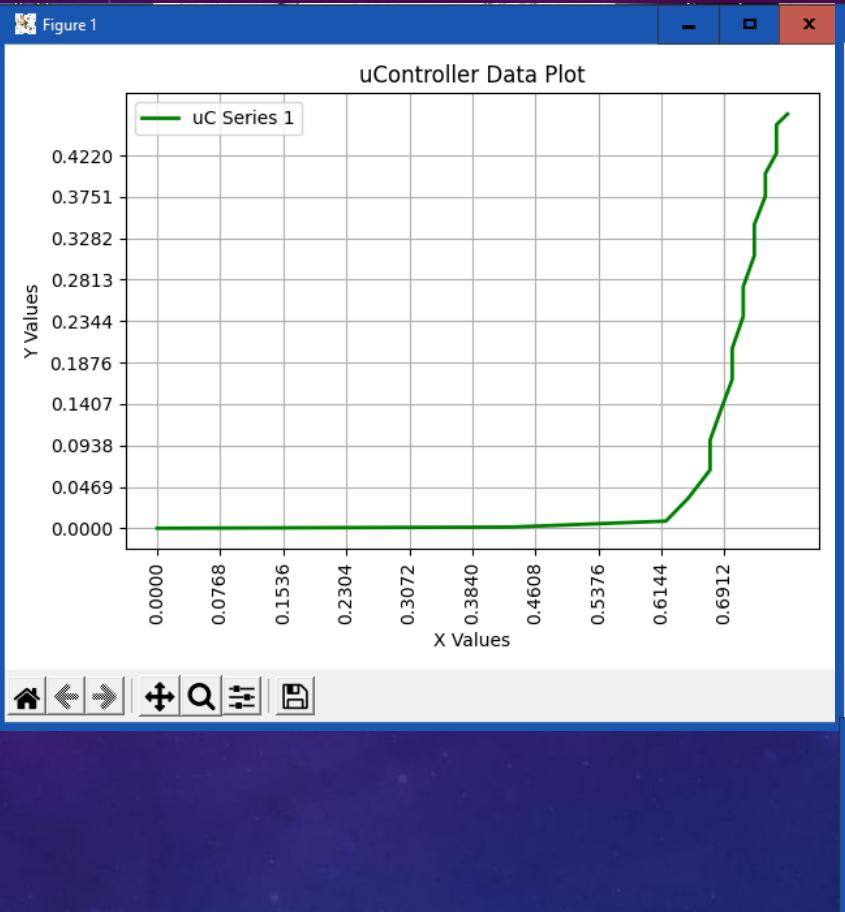
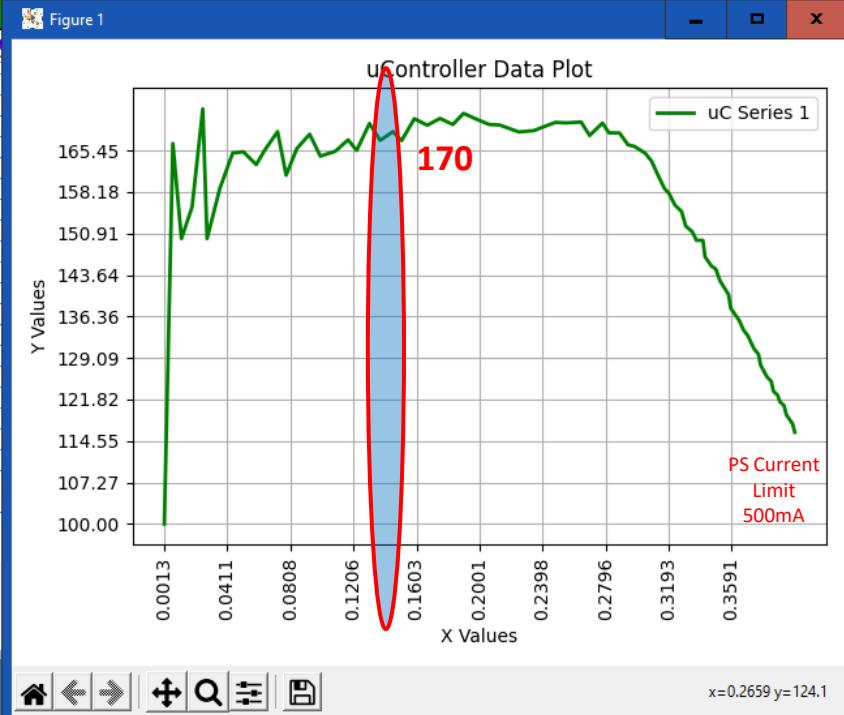


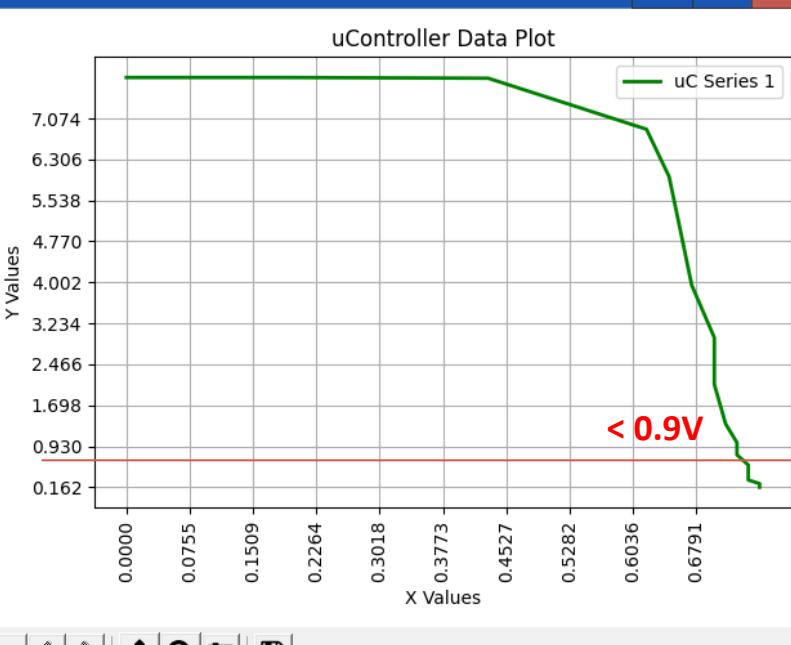
 Figure 1



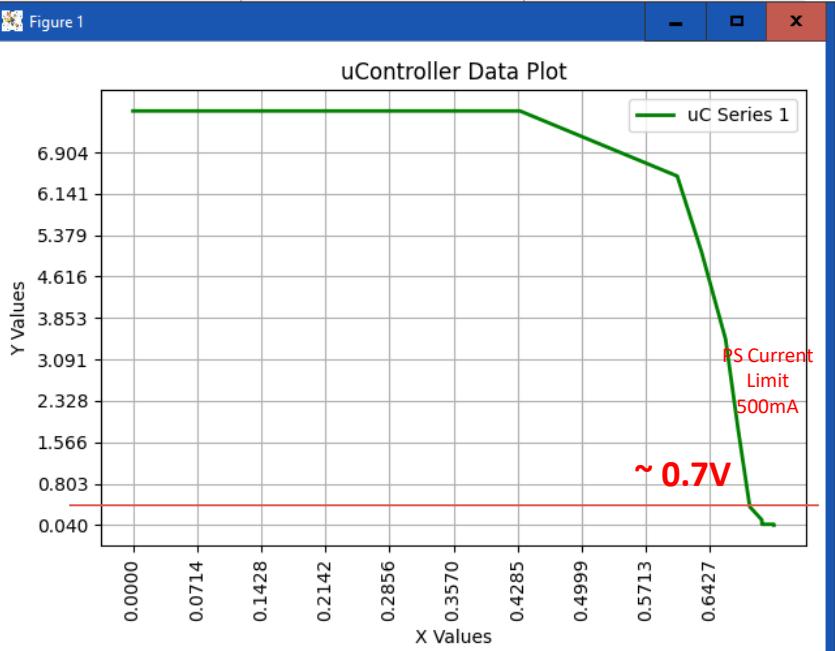
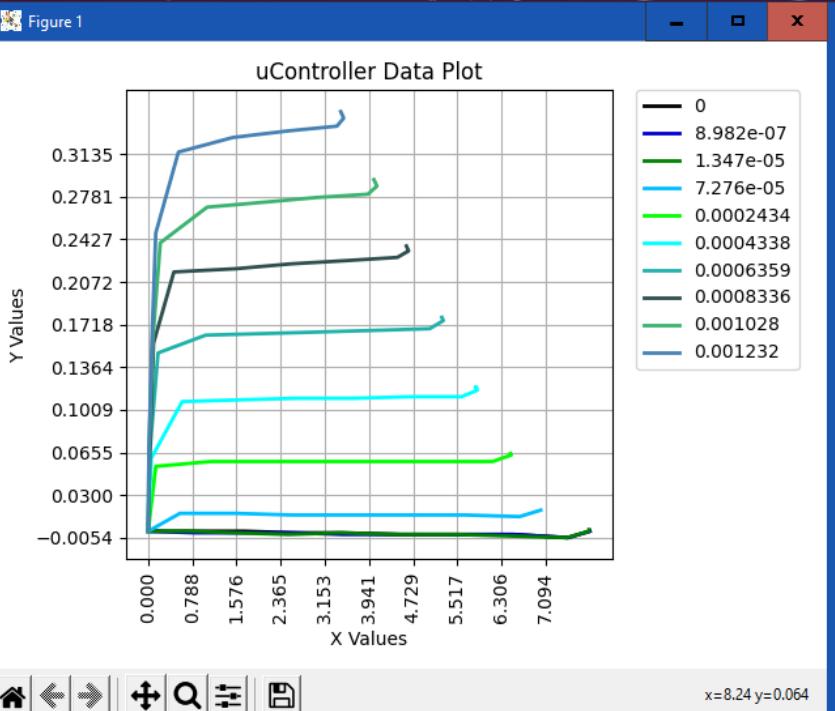
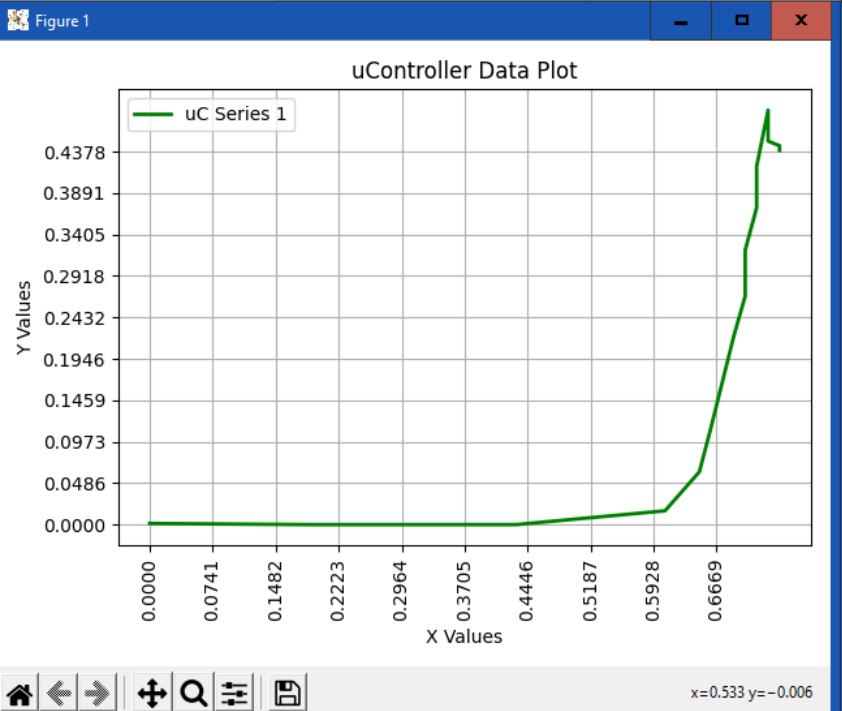
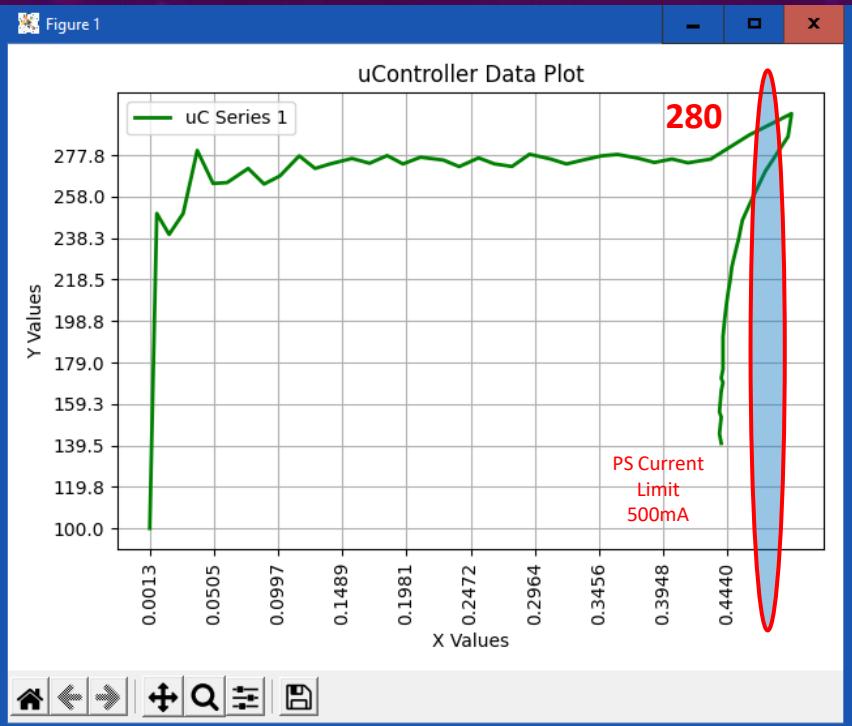
2N3050



SYMBOL	TEST CONDITIONS		MIN	MAX	MIN	MAX	UNITS
$V_{CE(SAT)}$	$I_C = 150\text{mA}$, $I_B = 15\text{mA}$		-	1.4	-	0.3	V
$V_{BE(SAT)}$	$I_C = 150\text{mA}$, $I_B = 15\text{mA}$		-	1.7	0.6	1.0	V
$V_{BE(ON)}$	$V_{CE} = 2.5\text{V}$, $I_C = 150\text{mA}$		-	1.7	-	1.0	V
β_{FE}	$V_{CE} = 2.5\text{V}$, $I_C = 150\text{mA}$		25	-	25	-	
β_{FE}	$V_{CE} = 10\text{V}$, $I_C = 150\text{mA}$		50	250	50	250	



2SC5706

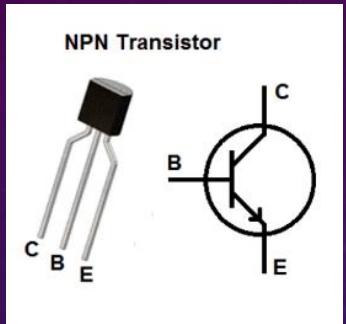


Parameter	Symbol	Conditions	Ratings			Unit
			min	typ	max	
Collector Cutoff Current	I_{CBO}	$V_{CB}=(-)40V, I_E=0A$			(-)1	μA
Emitter Cutoff Current	I_{EBO}	$V_{EB}=(-)4V, I_C=0A$			(-)1	μA
DC Current Gain	h_{FE}	$V_{CE}=(-)2V, I_C=(-)500mA$	200		560	
Gain-Bandwidth Product	f_T	$V_{CE}=(-)10V, I_C=(-)500mA$		(360)400		MHz
Output Capacitance	C_{ob}	$V_{CB}=(-)10V, f=1MHz$		(24)15		pF
Collector-to-Emitter Saturation Voltage	$V_{CE(sat)1}$	$I_C=(-)1A, I_B=(-)50mA$	(-115)90	(-195)135		mV
	$V_{CE(sat)2}$	$I_C=(-)2A, I_B=(-)100mA$	(-255)160	(-430)240		mV
Base-to-Emitter Saturation Voltage	$V_{BE(sat)}$	$V_{CE}=(-)2V, I_B=(-)100mA$	(-)0.89	(-)1.2		V

MEASUREMENTS OF N-CHANNEL ENHANCEMENT MODE MOSFETS

TRANSISTOR OPERATION

NPN



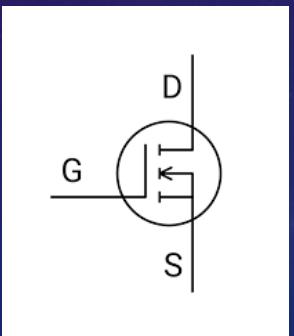
Feed Small Current into Base

NPN TRANSISTOR
 β

Get a LARGE current between Collector and Emitter

Datasheets today use h_{fe} instead of Beta. Beta sounds “cooler”

N-CH
MOSFET



Feed Voltage above Threshold

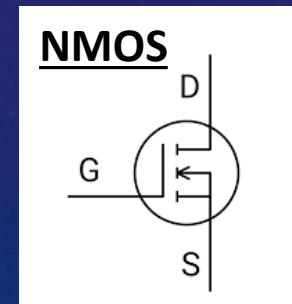
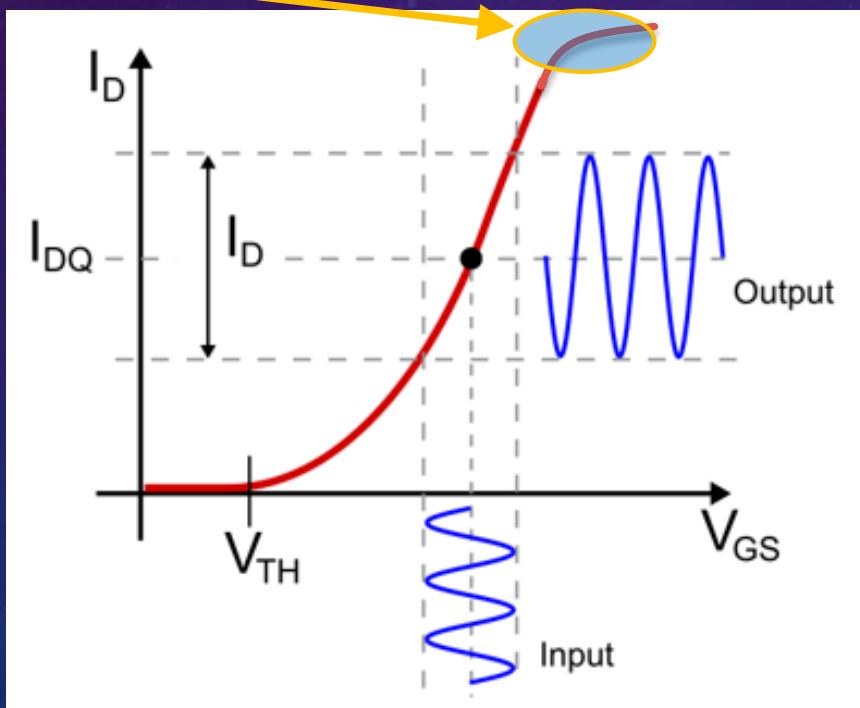
NMOS TRANSISTOR
 g_m

Get a LARGE current between Drain and Source

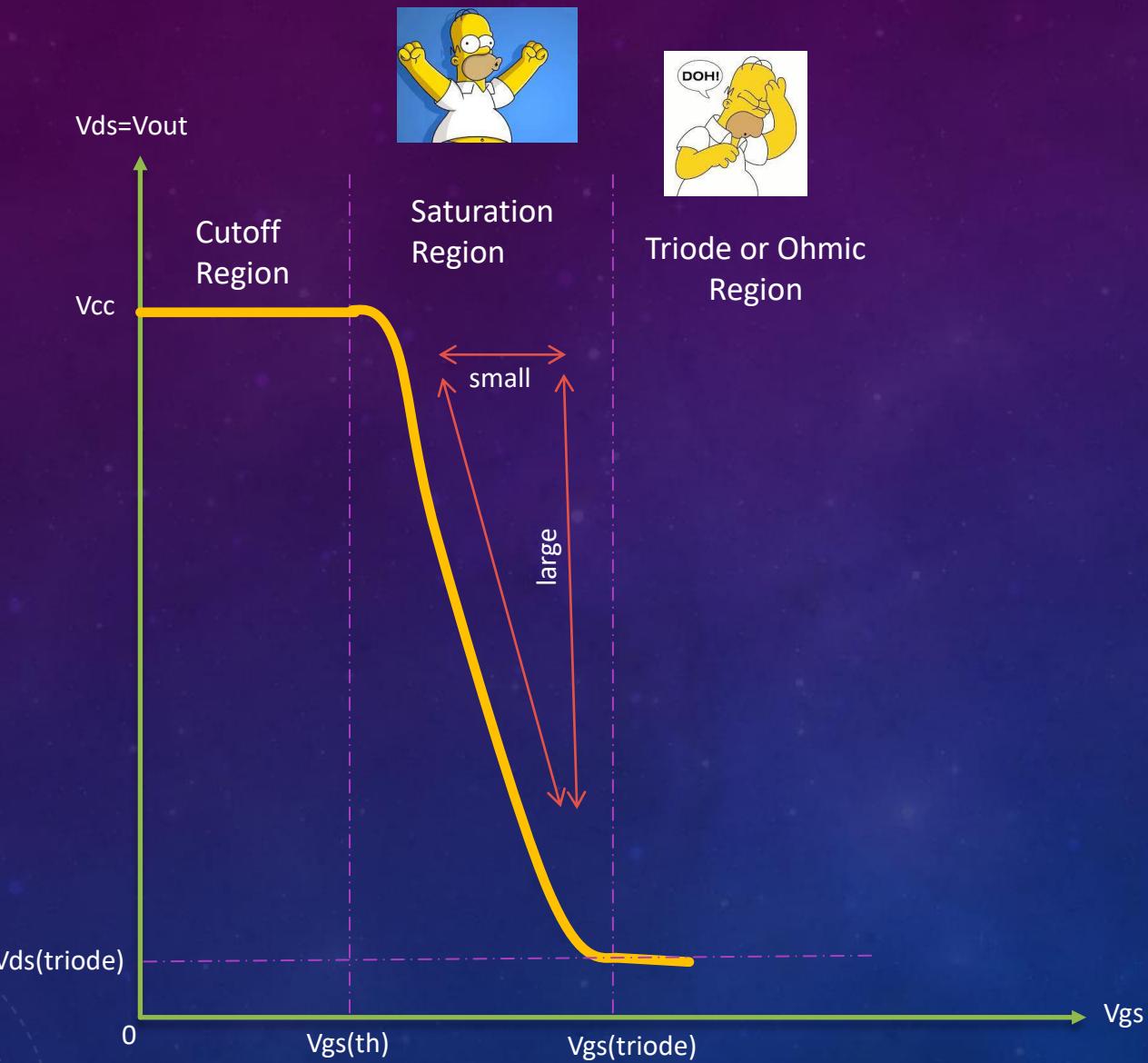
NMOS MOSFETS

Similar to BJT:

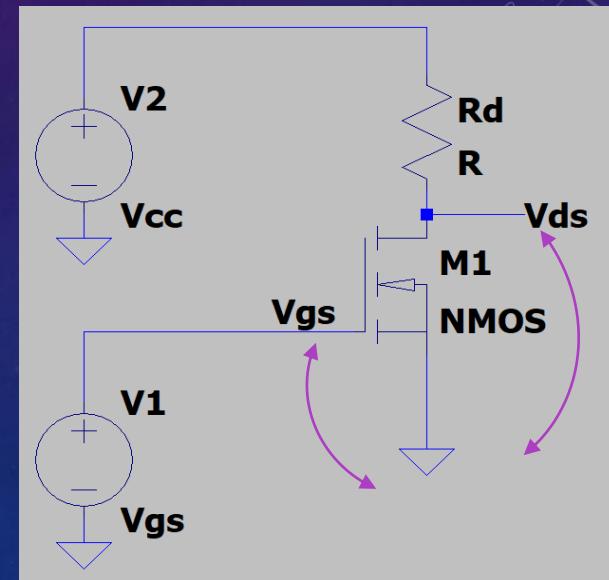
- There is V_{be} (turn on) for BJT, there is $V_{gs(on)}$ or $V_{gs(th)}$. This is threshold voltage at which device starts conducting
- There is a maximum V_{be} or V_{gs} where current is at **maximum** and constant. $V_{be(sat)}$ and $V_{gs(triode?)}$. **For MOSFET, Not specified on data sheet!!**



MOSFET REGIONS: VOLTAGE



Triode or Ohmic Region



MOSFETS

- Similar Input and Output charts to BJT:
 - However, the input curve will show saturation if V_{gs} is high enough
- No concept of Beta (current amplification). MOSFET is a voltage device. Voltage controls current.
- The gain depends on g_m (transconductance) and K
- K is based on channel parameters inside MOSFET

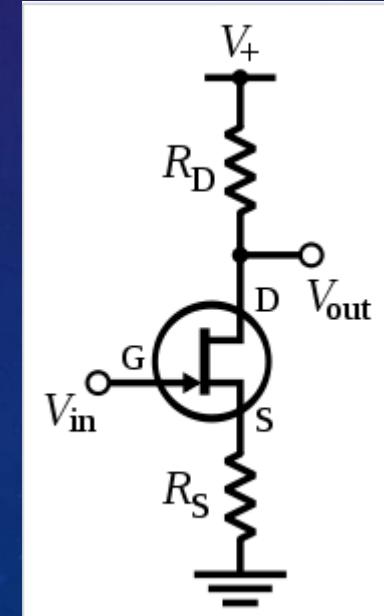
$$I_d = \frac{K}{2} (V_{gs} - V_t)^2 \text{ where } K = \mu C_{ox} \frac{W}{L}$$

$$K = 2 \frac{I_d}{(V_{gs} - V_t)^2}$$

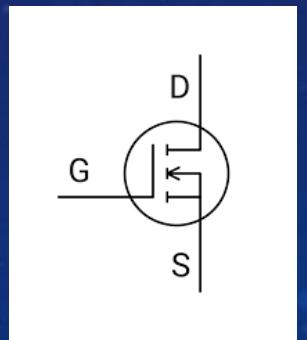
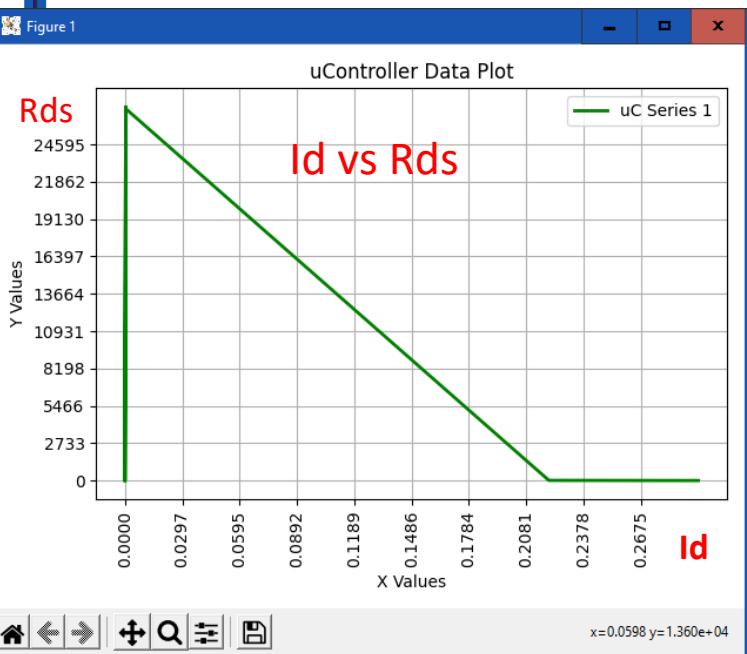
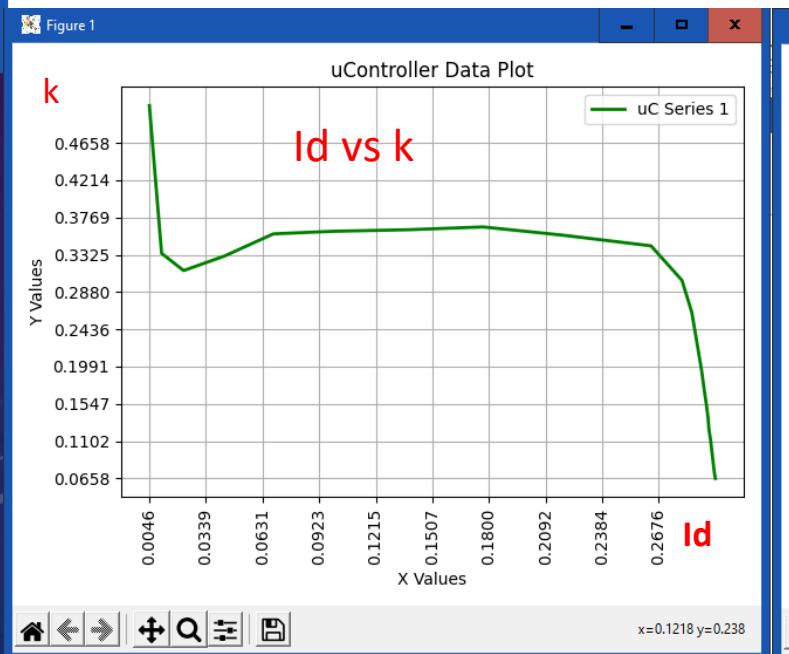
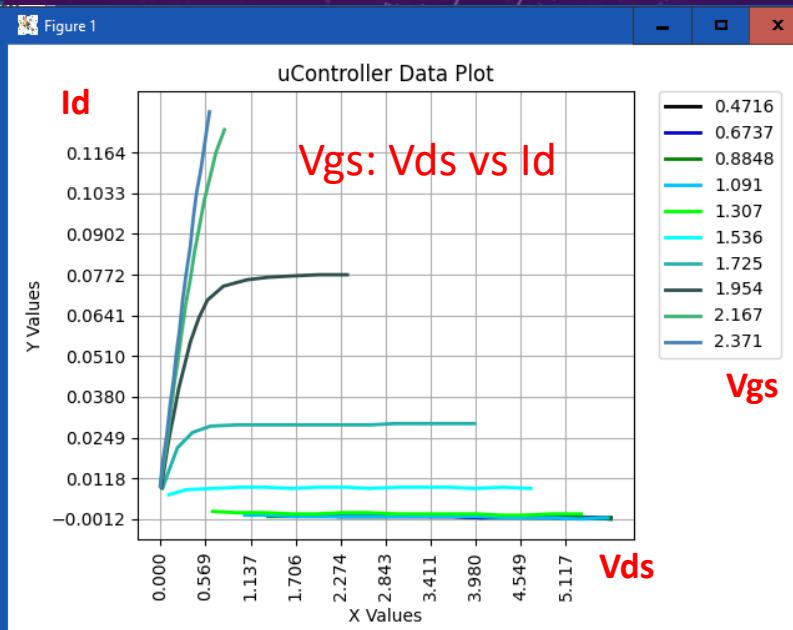
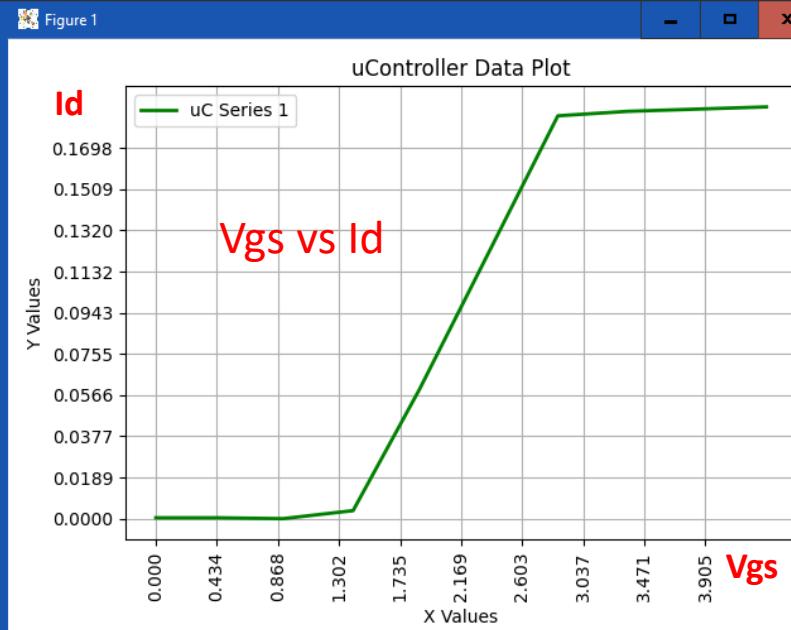
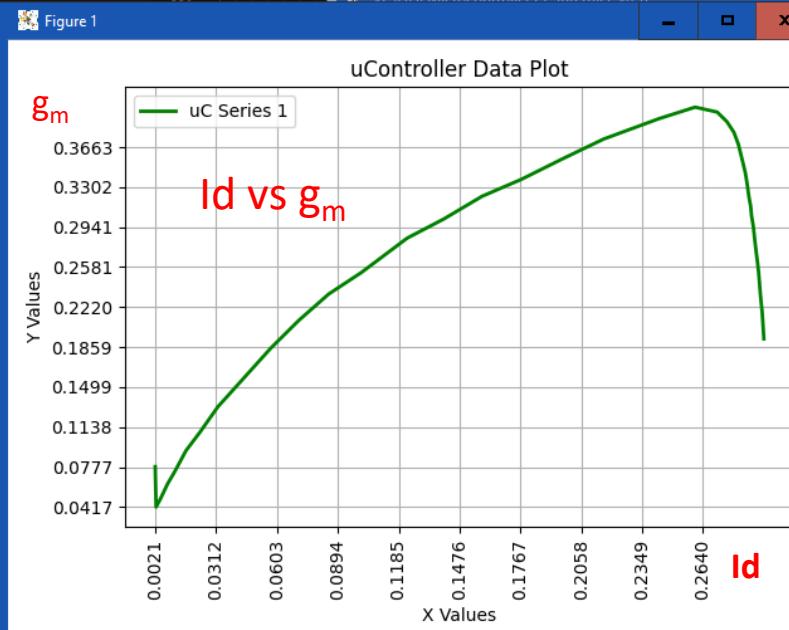
$$g_{m(sat.)} = \frac{\partial I_{D(sat.)}}{\partial V_{GS}} = \frac{W\mu_s C_{ox}}{L} (V_{GS} - V_{th})$$

Datasheets typically provide transconductance g_m

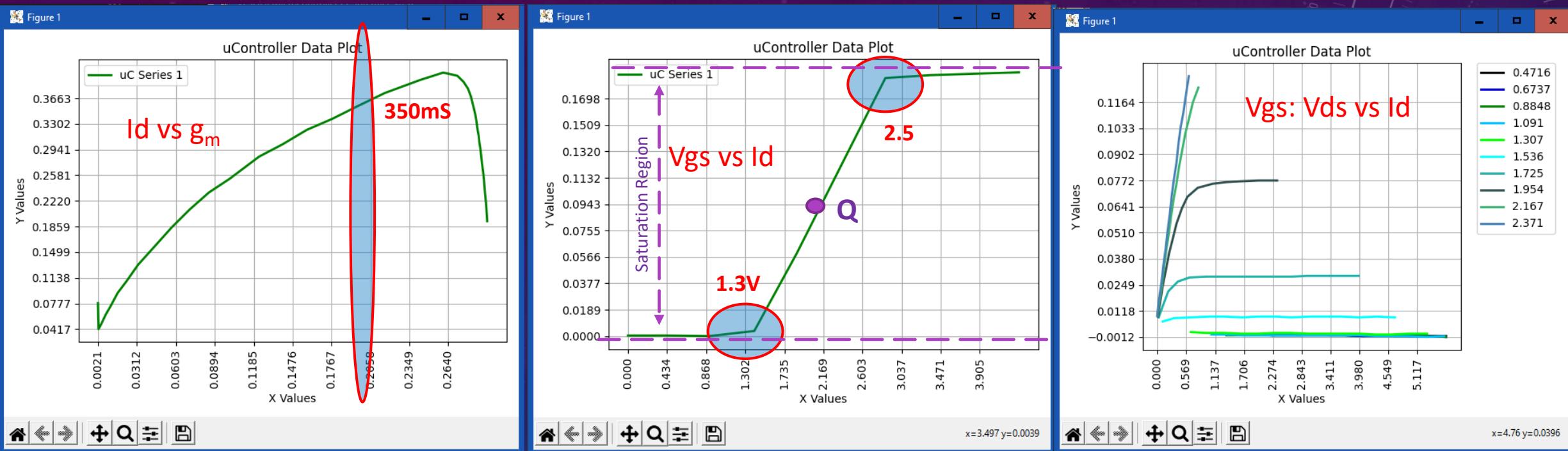
	Definition	Expression
Current gain	$A_i \triangleq \frac{i_{out}}{i_{in}}$	∞
Voltage gain	$A_v \triangleq \frac{v_{out}}{v_{in}}$	$-\frac{g_m R_D}{1 + g_m R_S}$
Input impedance	$r_{in} \triangleq \frac{v_{in}}{i_{in}}$	∞
Output impedance	$r_{out} \triangleq \frac{v_{out}}{i_{out}}$	R_D



NMOS PLOTS



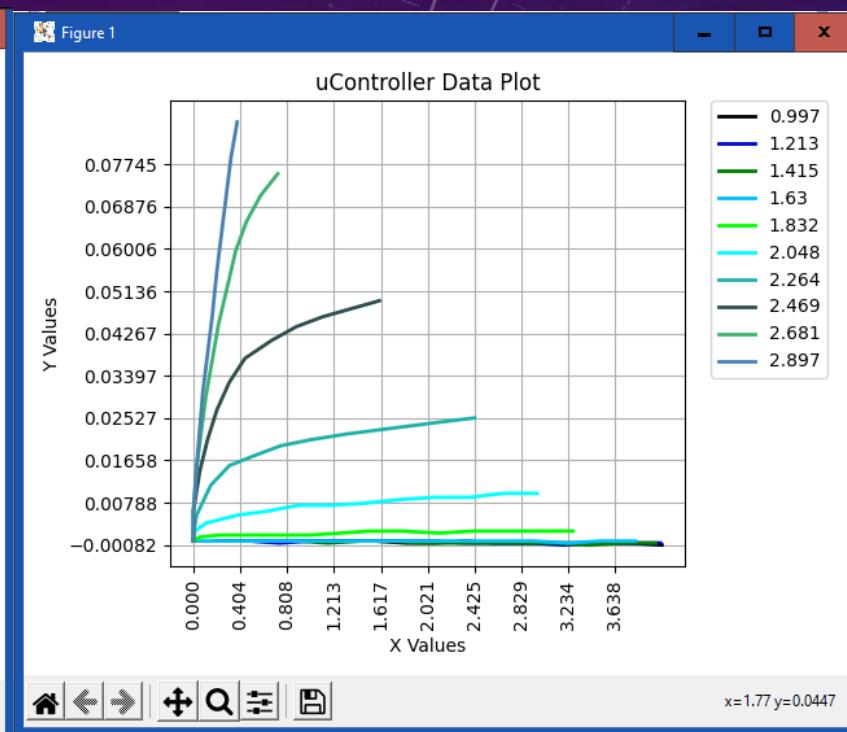
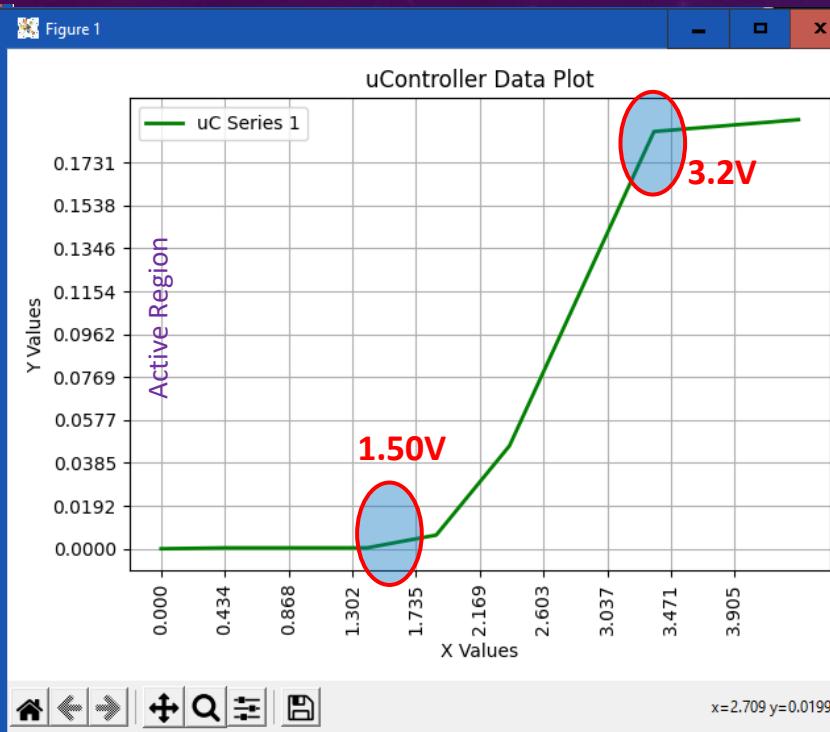
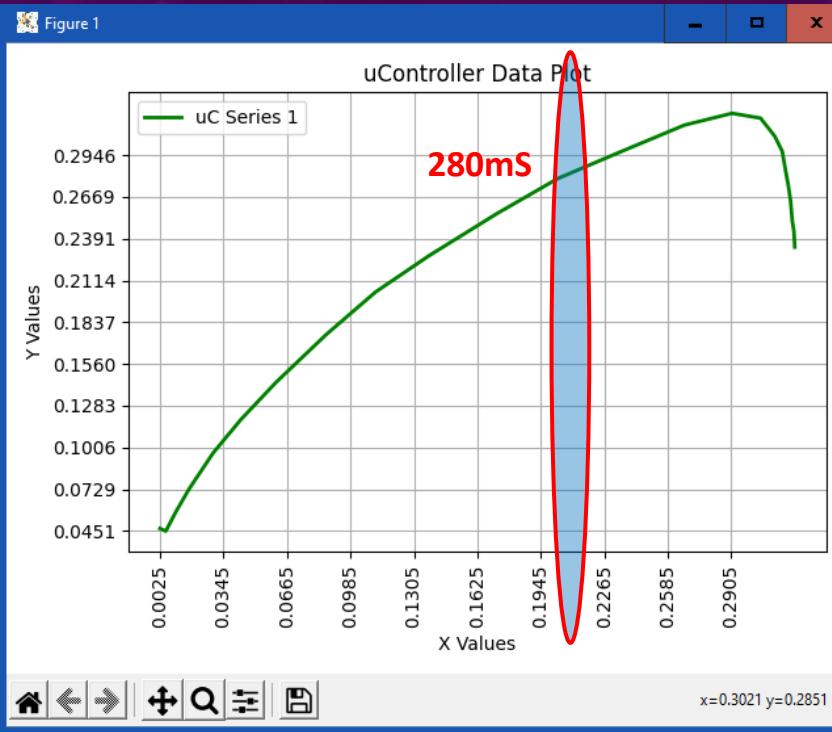
2N7000



FET K Sweep
Vg 0.00 - 4.60 inc: 0.1150
Getting $V_{gs(on)}$, $I_{ds(on)}$
 $VgControl(on)$: 1.2650 $V_{gs(on)}$: 1.2665 $I_{ds(on)}$: 0.0016 $V_{gs(th)}$: 2.4926 $R_{ds(on)}$: 4.8311 k. 0.0011

On Characteristics						
$V_{GS(th)}$	Gate Threshold Voltage	$V_{DS} = V_{GS}$, $I_D = 1 \text{ mA}$	2N7000	0.8	2.1	3
		$V_{DS} = V_{GS}$, $I_D = 250 \mu\text{A}$	2N7002 NDS7002A	1	2.1	2.5
$R_{DS(ON)}$	Static Drain-Source On-Resistance	$V_{GS} = 10 \text{ V}$, $I_D = 500 \text{ mA}$	2N7000	1.2	5	
		$V_{GS} = 10 \text{ V}$, $I_D = 500 \text{ mA}$, $T_C = 125^\circ\text{C}$		1.9	9	
g_{FS}	Forward Transconductance	$V_{DS} = 10 \text{ V}$, $I_D = 200 \text{ mA}$	2N7000	100	320	ms

BS170



FET K Sweep

Vg 0.00 - 4.60 inc: 0.1150

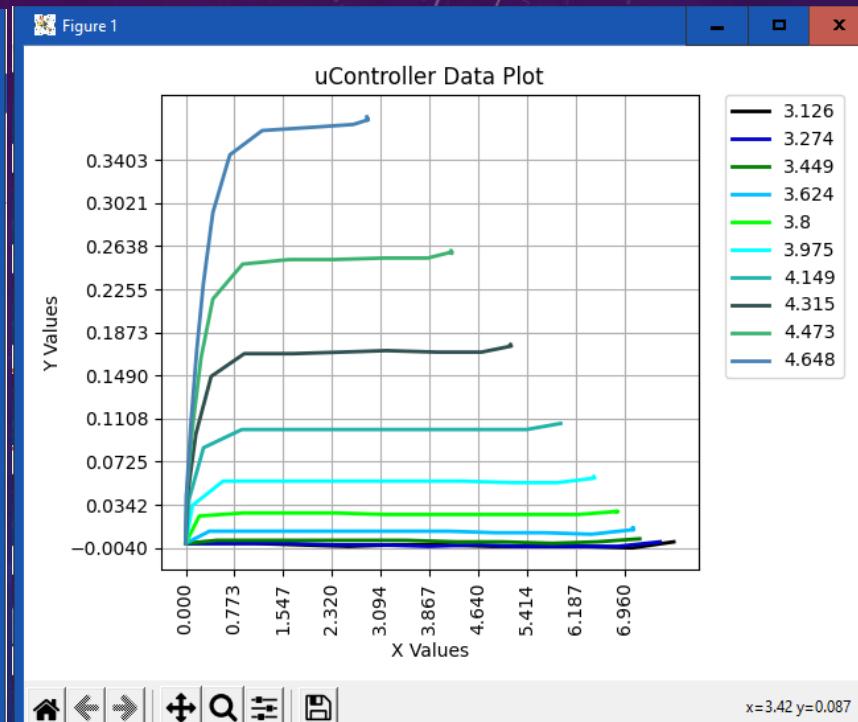
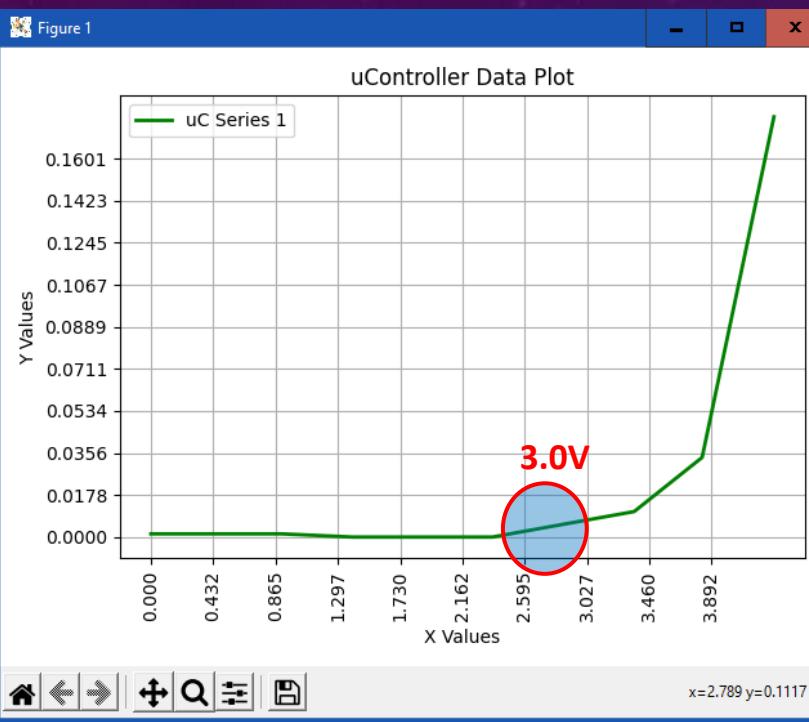
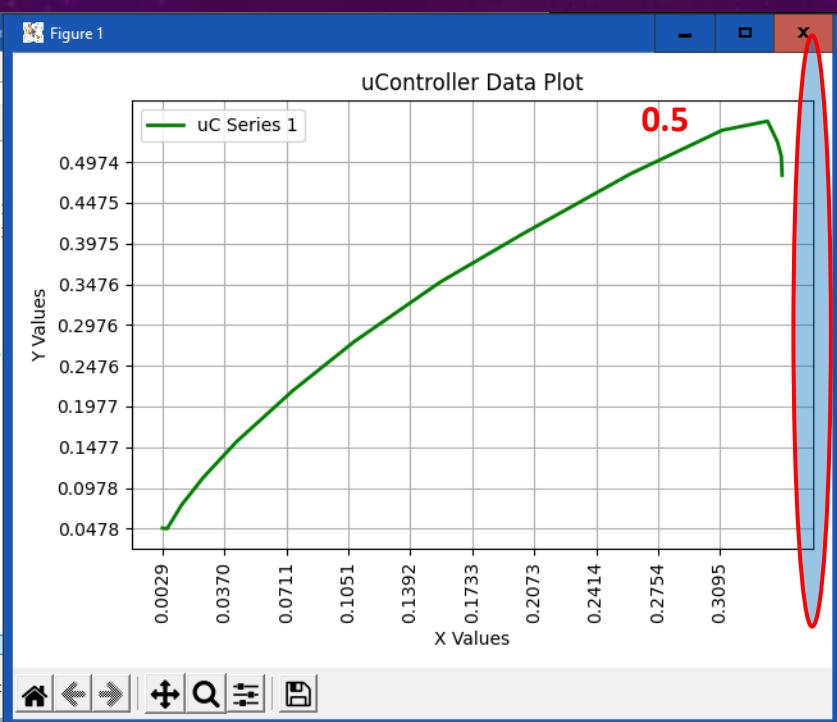
Getting Vgs(on), Ids(on)

VgControl(on): 1.4950 Vgs(on): 1.5090 Ids(on): 0.0008 Vgs(th): 3.2202 Rds(th): 4.1707 k: 0.0003

ON CHARACTERISTICS (Note 1)

$V_{GS(th)}$	Gate Threshold Voltage	$V_{DS} = V_{GS}, I_D = 1 \text{ mA}$	All	0.8	2.1	3	V
$R_{DS(ON)}$	Static Drain-Source On-Resistance	$V_{GS} = 10 \text{ V}, I_D = 200 \text{ mA}$	All	-	1.2	5	Ω
g_{fs}	Forward Transconductance	$V_{DS} = 10 \text{ V}, I_D = 200 \text{ mA}$	BS170	-	320	-	mS
		$V_{DS} \geq 2 V_{DS(on)}, I_D = 200 \text{ mA}$	MMBF170	-	320	-	

IRF510



FET K Sweep

Vg 0.00 - 4.60 inc: 0.1150

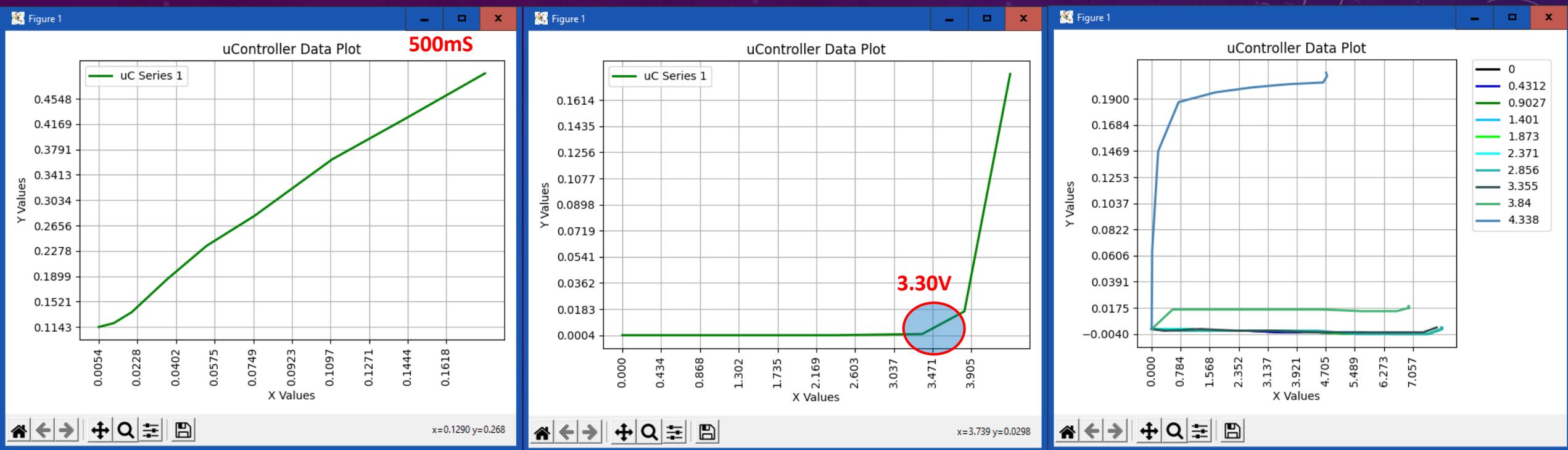
Getting Vgs(on), Ids(on)

VgControl(on): 2.9900 Vgs(on): 3.0989 Ids(on): 0.0027 Vgs(th): 4.8235 Rds(th): 0.2865 K: 0.0009

Don't have the drive to measure this MOSFET

Gate-source threshold voltage	$V_{GS(th)}$	$V_{DS} = V_{GS}, I_D = 250 \mu\text{A}$	2.0	-	4.0	V
Gate-source leakage	I_{GSS}	$V_{GS} = \pm 20 \text{ V}$	-	-	± 100	nA
Zero gate voltage drain current	I_{DSS}	$V_{DS} = 100 \text{ V}, V_{GS} = 0 \text{ V}$	-	-	25	μA
		$V_{DS} = 80 \text{ V}, V_{GS} = 0 \text{ V}, T_J = 150^\circ \text{C}$	-	-	250	
Drain-source on-state resistance	$R_{DS(on)}$	$V_{GS} = 10 \text{ V}$	$I_D = 3.4 \text{ A}^b$	-	-	0.54 Ω
Forward transconductance	g_{fs}	$V_{DS} = 50 \text{ V}, I_D = 3.4 \text{ A}^b$	1.3	-	-	S

IRFZ20



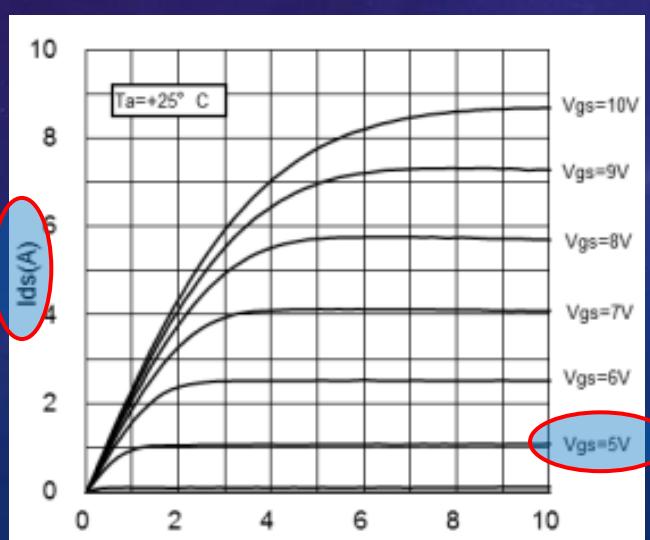
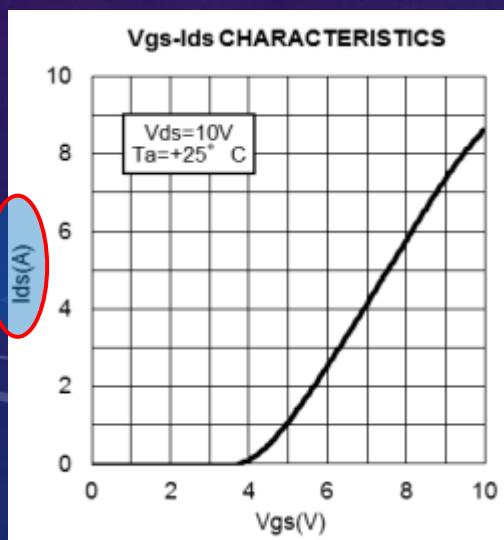
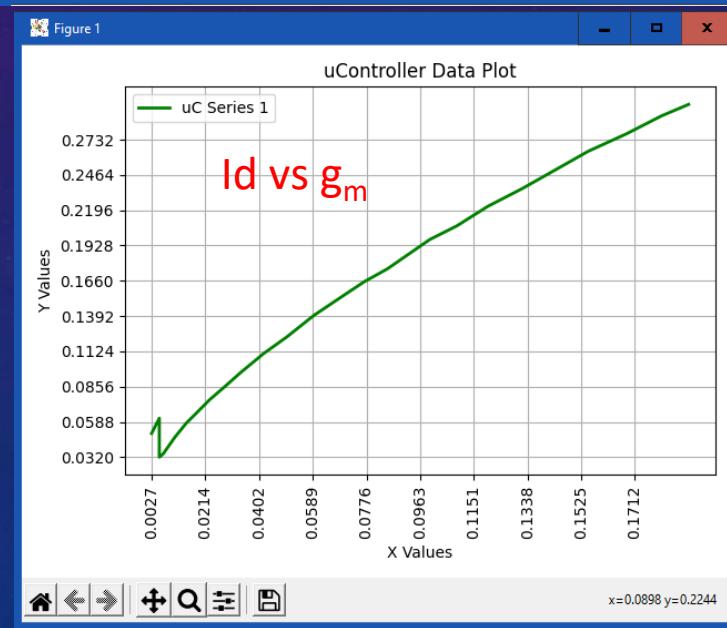
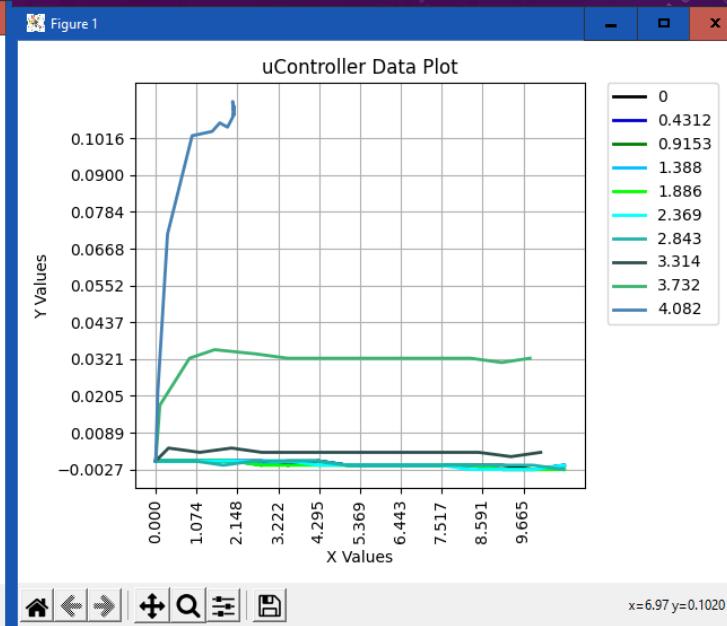
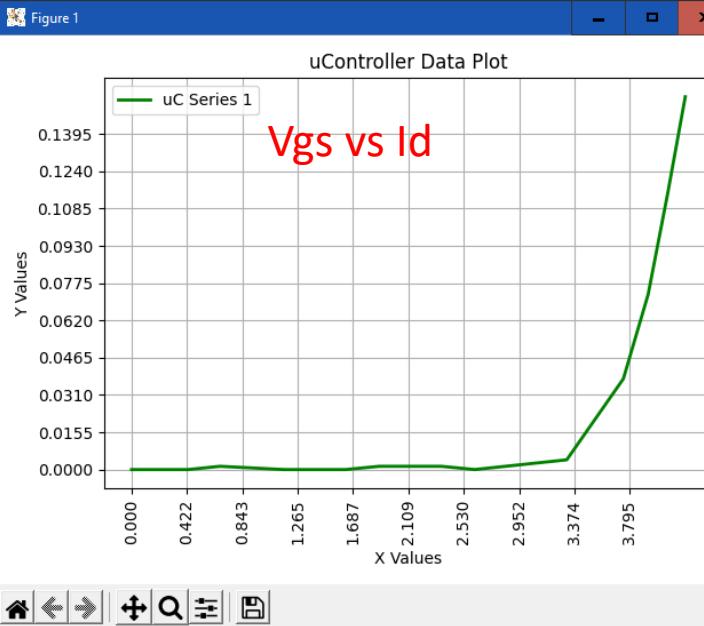
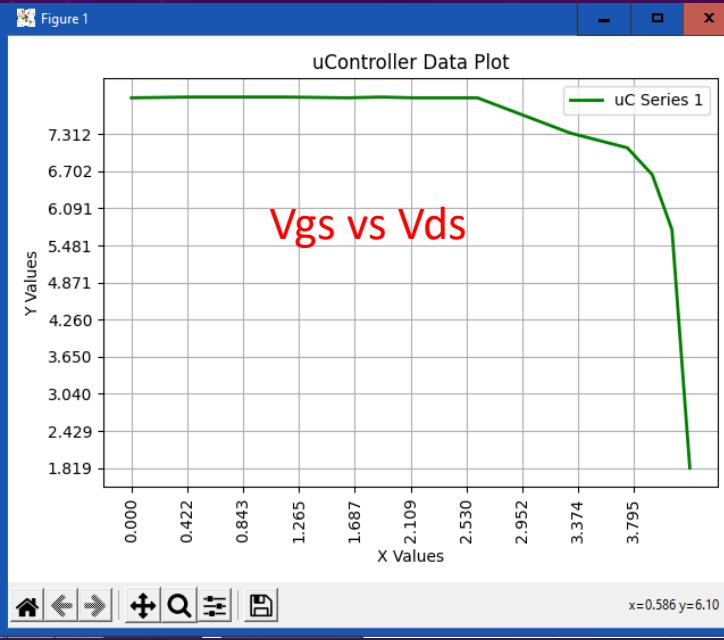
FET K Sweep
Vg 0.00 - 4.60 inc: 0.1150
Getting Vgs(on), Ids(on)

VgControl(on): 3.2200 Vgs(on): 3.3549 Ids(on): 0.0027 Vgs(th): 4.8235 Rds(th): 0.2865 k: 0.0012

Don't have the drive to measure this MOSFET

Gate-Source Threshold Voltage	$V_{GS(th)}$	$V_{DS} = 0 \text{ V}, I_D = 250 \mu\text{A}$	50	-	-	V	
Gate-Source Leakage	I_{GSS}	$V_{GS} = \pm 20 \text{ V}$	2.0	-	4.0	V	
Zero Gate Voltage Drain Current	I_{DSS}	$V_{DS} > \text{Max. Rating}, V_{GS} = 0 \text{ V}$	-	-	250	nA	
On-State Drain Current	$I_{D(on)}$	$V_{GS} = 10 \text{ V}, V_{DS} > I_{D(on)} \times R_{DS(on)} \text{ max.}$	-	-	1000	μA	
Drain-Source On-State Resistance ^b	$R_{DS(on)}$	$V_{GS} = 10 \text{ V}$	$I_D = 10 \text{ A}$	-	0.080	0.10	Ω
Forward Transconductance ^b	g_{fs}	$V_{DS} > I_{D(on)} \times R_{DS(on)} \text{ max.}, I_D = 9.0 \text{ A}$	5.0	6.0	-	S	

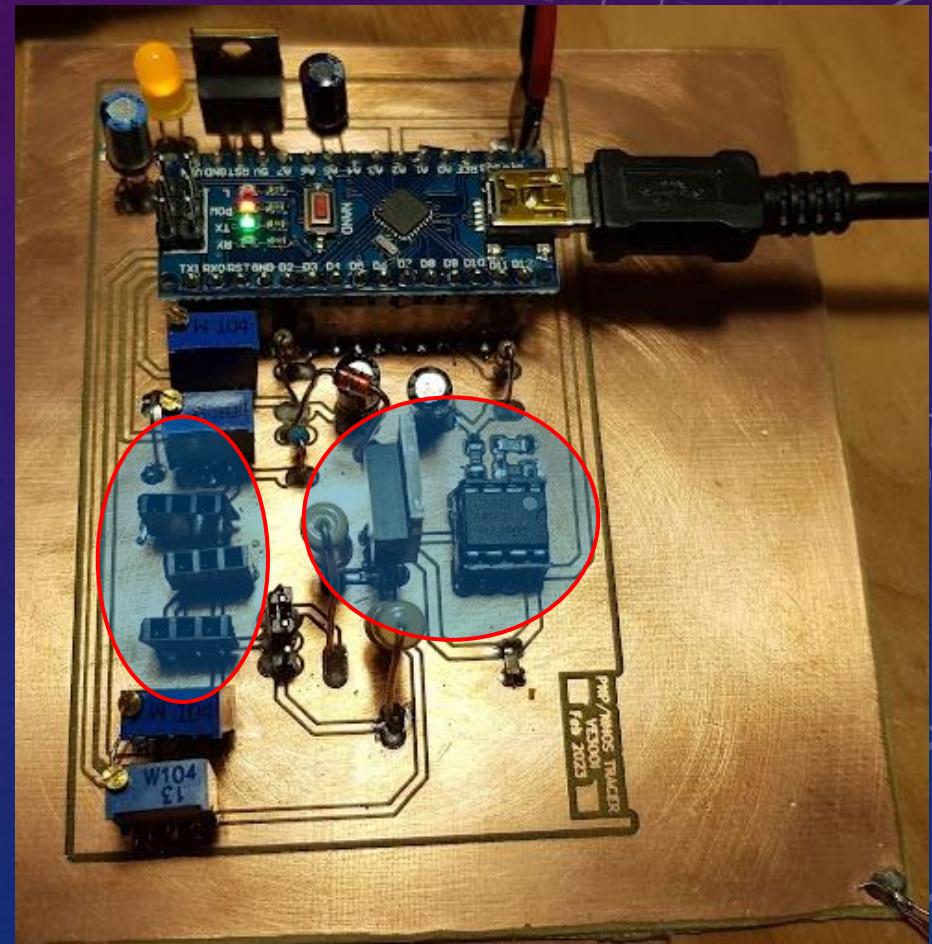
RD16HHF1



Don't have the drive to measure this MOSFET

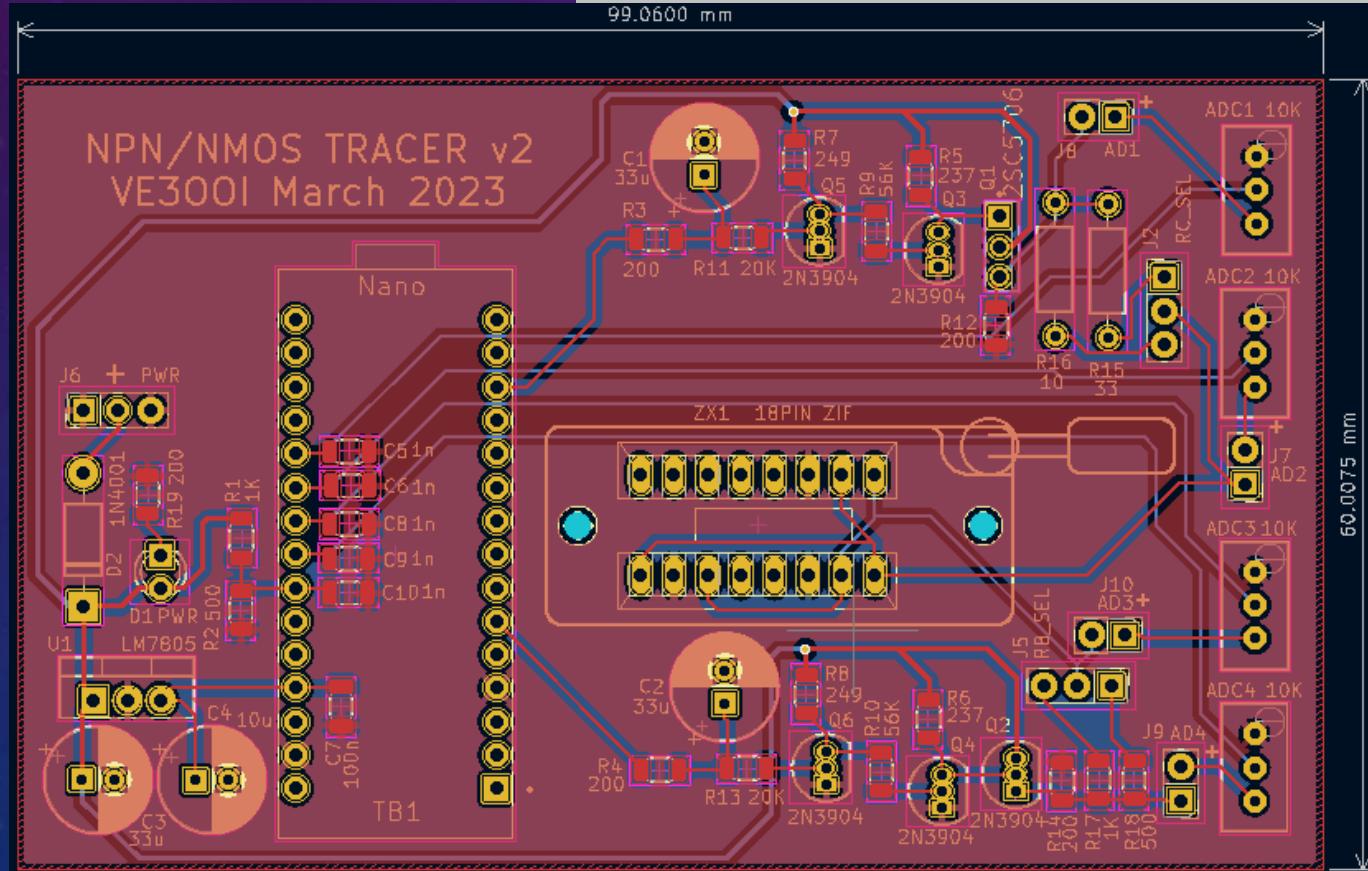
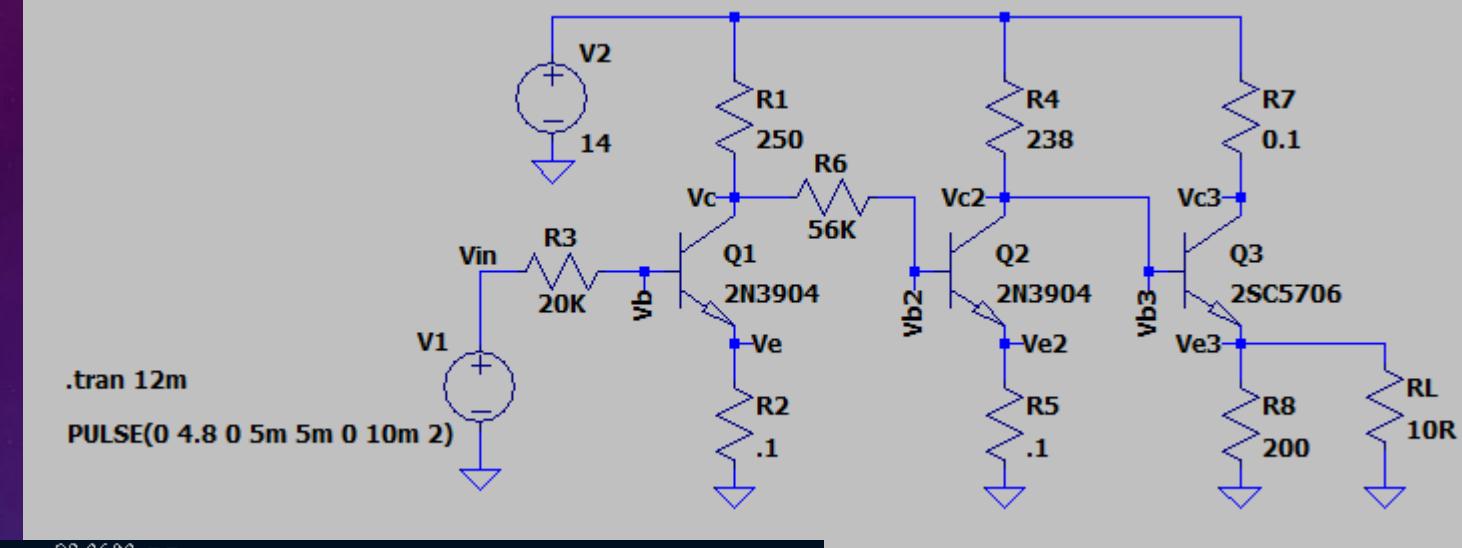
ISSUES

- Need higher current and voltage for Vcc and Vb
 - ❖ Voltage drop from Opamp and TIP120 ($\sim 8V$)
 - ❖ Using PWM ($\sim 4.3V$) for Vb
- Need better way to sweep Vcc/Vc
- Need better socket for transistors



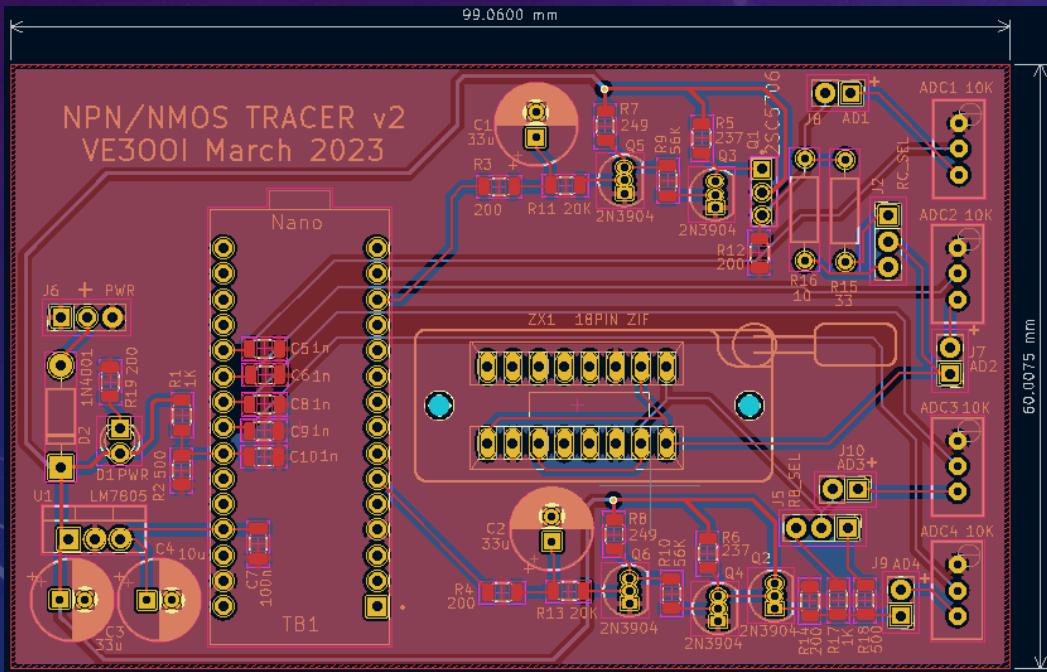
VERSION 2

- ZIF Socket
- DC Voltage Amplifier



VERSION 2: SPEEDS AND FEEDS

- ✓ **NPN** BJTs with EBC, BEC, BCE
- ✓ **NMOS** (N Channel Enhancement Mode MOSFET) with GDS, GSD, DGS
- ✓ Vcc 14V, Max Collector/Drain Voltage: 11.1@1.7A
- ✓ Vcc 14V, Max Base/Gate voltage: 11.1V@22mA



Version 1:

- ✓ Vcc 14V, Max Collector/Drain Voltage: 10@930mA
- ✓ Max Base/Gate voltage: 4.30@7mA

NPN/NMOS Transistor Tracer
VE300I V1 Feb 2023

