

The Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET)

Physics Topics

If necessary, review the following topics and relevant textbook sections from Neamen “Semiconductor Physics and Devices”, 4th Ed. and Serway “Physics for Scientists and Engineers”, 11th Ed.

- Section 8.1.5, especially equation 8.27 (Neamen)
- Band Theory of Solids (Serway, 42.5)
- Semiconductor Devices (Serway, 42.7)

Introduction

Recall that in Lab 3 we studied the current versus voltage properties of a forward biased diode. The diode consisted of a PN semiconductor junction (in practice we used an NPN BJT transistor but focused our attention on the Base-Emitter PN junction). By capturing the current versus voltage characteristics of the junction we were able to determine Boltzmann’s constant, and we were also able to use the temperature dependency of the diode’s reverse bias current, I_o , to determine the band gap of the silicon diode.

The development of the PN junction set the stage for the fabrication of many subsequent “active” semiconductor devices. One of the most important such devices is the metal-oxide-semiconductor field effect transistor (MOSFET). Whereas transistors in general are important because of their versatility, behaving as switches, amplifiers, or oscillators depending on their configuration, MOSFETs in particular exhibit extremely beneficial low-power switching properties when compared against alternative transistor types. As such they have become absolutely fundamental in computing and memory applications, where their physical dimensions have been scaled down to allow the fabrication of many millions of MOSFET transistors on individual semiconductor chips or integrated circuits.

In this lab we will measure several characteristics of a discrete MOSFET transistor and compare the properties to the expected parameters provided by the manufacturer.

MOSFET Structure

Figure 1 illustrates the profile of an n-channel MOSFET. Two n-doped regions, the drain and the source, are embedded in a p-type semiconductor substrate.

On the surface of the semiconductor is a layer of insulator (SiO₂ in the case of Si substrates). Openings etched or masked into the insulator allow the deposition of metallic electrodes contacting n-doped drain and source regions. Note that beneath the gate electrode the insulator layer remains intact, isolating the gate and substrate. Voltages ap-

plied to this “gate” electrode control the flow of current between the source and drain by introducing or depleting charge carrier states in the substrate region beneath the gate.

Theory of Operation

The MOS Capacitor

The MOSFET's behavior and operation is based in large part on the physics of the MOS capacitor. We'll initially ignore the presence of the source and drain, leaving only the capacitor-like gate, insulator, and semiconductor structure. We'll briefly consider how a voltage applied to the gate modifies the energy bands of the semiconductor substrate in the vicinity of the area beneath the gate, and how this in turn influences the carrier densities and depletion regions therein. When V_G is 0V, the metal and semiconductor Fermi-levels align. The semiconductor's Fermi-level is $q\phi_F$ electron volts below its intrinsic level, E_i , and is indicative of how strongly p-type the substrate is, as we recall that the concentration of majority carriers in doped p-type semiconductor is given by:

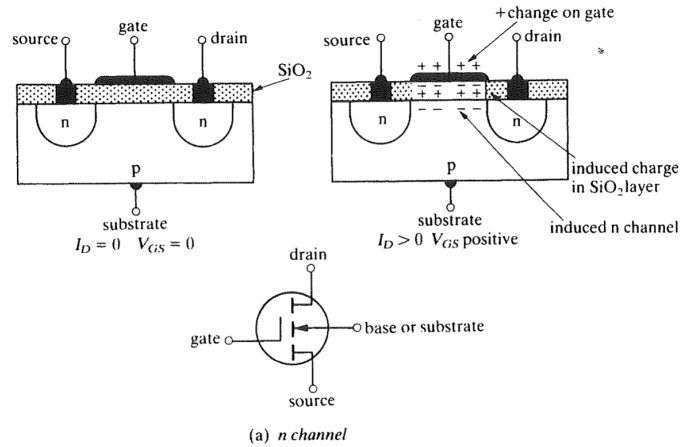


Figure 1: Physical structure of an enhancement mode n-channel MOSFET, and schematic symbol [1].

When V_G is 0V, the metal and semiconductor Fermi-levels align. The semiconductor's Fermi-level is $q\phi_F$ electron volts below its intrinsic level, E_i , and is indicative of how strongly p-type the substrate is, as we recall that the concentration of majority carriers in doped p-type semiconductor is given by:

$$p = N_A \exp\left(\frac{E_F - E_V}{kT}\right) \quad (1)$$

When $V_G < 0V$ is applied at the gate, the Fermi-level in the metal increases by qV_G . This has the effect of depositing negative charges at the gate, which in turn attracts additional holes to the oxide-semiconductor interface. As such, the semiconductor bands are bent near the interface and the Fermi-level, E_F , and valence band, E_v , are closer to each other in energy due to the increased majority carrier density. The device is said to be in an “accumulation” state. Its capacitance is given by

$$C_{ox} = \frac{\epsilon_{ox}}{d_{ox}} \quad (2)$$

where ϵ_{ox} is the permittivity of the insulator layer and d_{ox} is the insulator thickness.

When $V_G > 0V$, things become more interesting. The redistribution of carrier states cause the bands of the semiconductor to bend near the interface such that the Fermi-level and E_i become farther apart. That is, $V_G > 0V$ causes positive charge to build at the gate, and in turn induces a reduction or “depletion” state in the p-type semiconductor in the area near the oxide-semiconductor interface. The effect is analogous to the depletion region

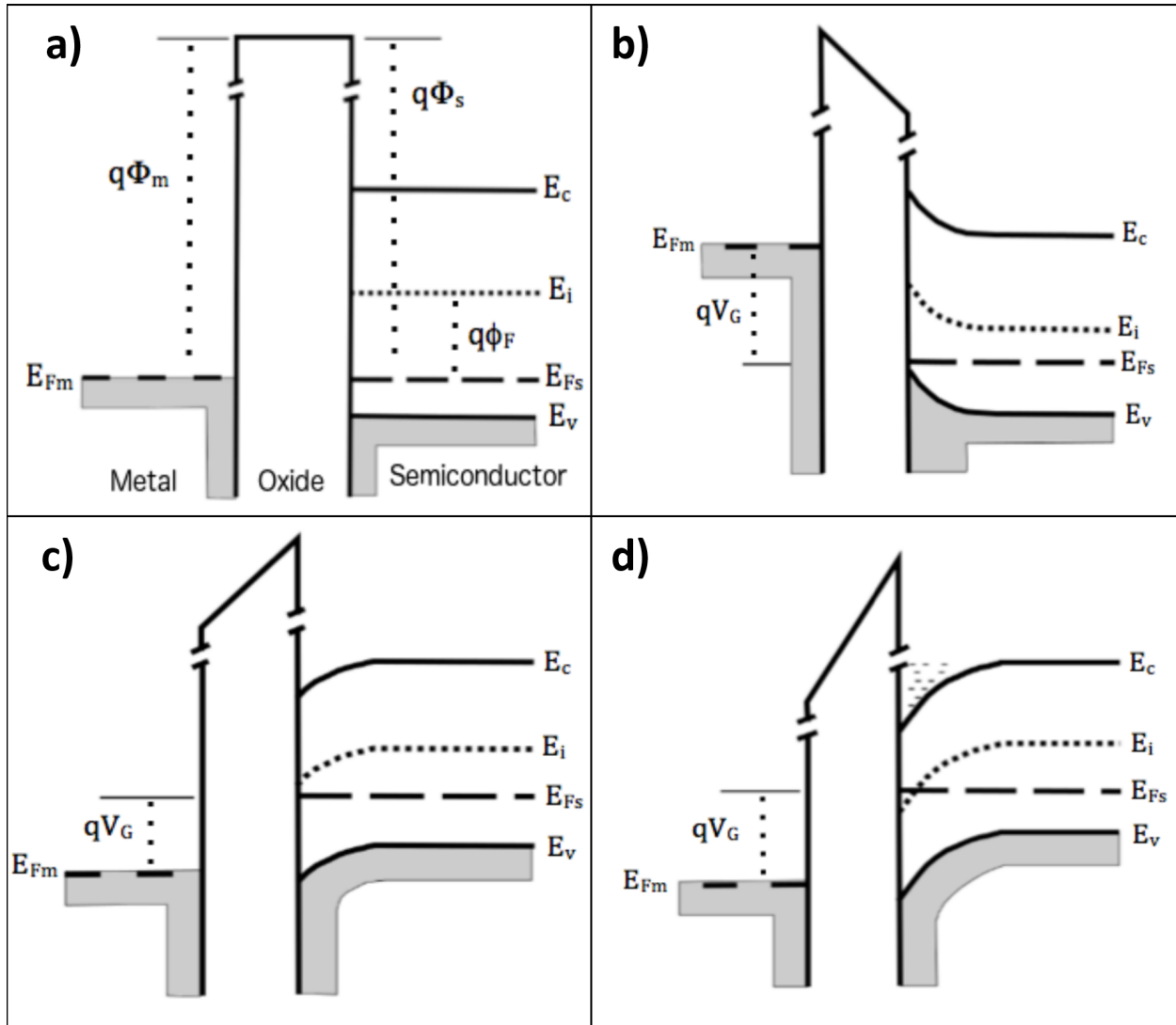


Figure 2: Ideal MOS under various voltage conditions.

- a) Flat band, $V_G = 0V$ b) Accumulation state, $V_G < 0V$
c) Depletion, $V_G > 0V$ d) Inversion, $V_G \gg 0V$

between a pn+ junction¹, the width of which is (2),

$$W = \left[\frac{2\epsilon_s(\phi_s)}{qN_a} \right]^{1/2} \quad (3)$$

Here, ϕ_s is the potential difference across the depletion region (i.e. energy band bending, as shown in Figure 3), where $V_G = V_i + \phi_s$ and V_i represents the potential difference across the oxide insulator layer.

As a result, the capacitance becomes like a series combination of the oxide capacitance and that due to the depletion region width,

$$C_d = \frac{\epsilon_s}{W} \quad (4)$$

where ϵ_s is the permittivity of the depletion layer.

As V_G becomes even larger, eventually E_F becomes greater than E_i . In this situation the region of semiconductor near the oxide interface becomes “inverted,” meaning that conduction band carrier states become filled with minority electrons forming an n-channel. (Realize that when we add drain and source electrodes to this MOS capacitor, we can induce current conduction across the newly formed n-channel just by applying a V_{DS} !)

At this point the depth of the semiconductor depletion layer is at a maximum, and the total capacitance of the device is at a minimum,

$$C = \left[\frac{1}{C_{ox}} + \frac{1}{C_d} \right]^{-1} = \frac{C_{ox}C_d}{C_{ox} + C_d} \quad (5)$$

The gate voltage required to induce this inversion state is defined as the device’s “threshold voltage,” V_T . V_T represents the point at which the MOSFET becomes conductive, and it can be controlled in device fabrication by tailoring the material parameters and physical dimensions of the MOSFET. As V_G becomes greater than V_T the inverted MOS capacitor enters a state of “strong inversion,” where $\phi_s = 2\phi_F$.

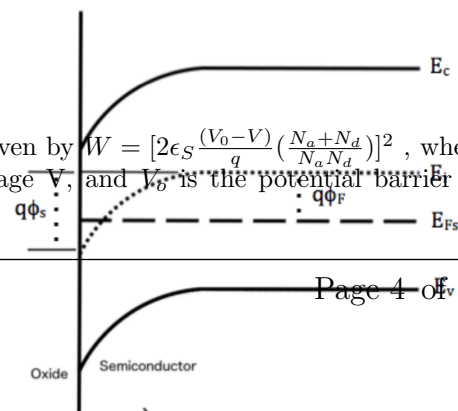
Measurements of V_T , C_{ox} , and C_d can thus be used to characterize many of the physical parameters of the device, like oxide thickness and doping densities. (That is, C-V measurements can serve as a useful characterization and reliability testing tool.)

The MOSFET Transistor

So having considered the states of the MOS capacitor, what of the MOSFET transistor physics?

With the addition of n-doped source and drain regions (or p-doped regions for a p-channel MOSFET), recognize that we’ve introduced two pn junctions into the device structure. If we consider the MOS capacitor’s accumulation and depletion states as we vary V_G ,

¹Recall that for a biased pn junction the depletion width is given by $W = [2\epsilon_s \frac{(V_0 - V)}{q} \left(\frac{N_a + N_d}{N_a N_d} \right)]^{1/2}$, where $V_0 - V_i$ is the change in the potential barrier due to bias voltage V , and V_0 is the potential barrier at equilibrium.



the substrate remains p-type and all that is altered is the concentration of majority p-type carriers in the area beneath the gate. As such, voltage applied from drain to source, V_{DS} , is equivalent to reverse biasing the drain-substrate np junction. The potential barrier is increased, the width of the depletion boundary between drain and substrate is widened, and no drain current flows.

However, as V_G becomes greater than V_T we've seen that inversion occurs, and suddenly the drain and source become conductively connected via the induced n-channel. Current can flow freely if a voltage V_{DS} is applied. The MOSFET is switched into its conductive mode.

We would still like to consider how the device behaves in this inversion state as we vary V_{DS} . While V_{DS} is small, the current from drain to source is observed to be roughly proportional to V_{DS} , and the MOSFET acts somewhat like an ohmic resistive load. In this scenario the device is particularly useful for switching applications. (Millions of MOSFET switches are used in computer memory and processors.)

As V_{DS} is increased for a given $V_G > V_T$ eventually all carriers generated in the channel layer are quickly swept from drain to source. As part of your report you'll comment on how this is reflected in the measurements you'll make of I_D versus V_{DS} .

Drain Current and Transconduction

In the non-saturation region, the drain current can be expressed as

$$I_D = K_n[2(V_{GS} - V_{TN})V_{DS} - V_{DS}^2] \quad (6)$$

$$K_n = \frac{W\mu_n C_{ox}}{2L} \quad (7)$$

Where K_n is called the transconduction parameter. In the linear region, where V_{DS} is small, the drain current can be approximated as

$$I_D = K_n[2(V_{GS} - V_{TN})V_{DS}] \quad (8)$$

This equation can be used to determine V_T and K_n when a small V_{DS} is applied.

In the saturation region, the drain current can be expressed as

$$I_D = K_n(V_{GS} - V_{TN})^2 \quad (9)$$

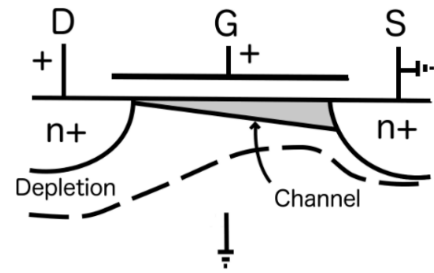


Figure 4: The MOSFET in profile. $V_G \gg 0V$ causes “strong inversion” and an n-channel forms, linking a conductive path from drain to source. The MOSFET is in “on” state.

$$K_n = \frac{W\mu_n C_{ox}}{2L} \quad (10)$$

This equation can potentially be used to determine V_T and K_n when a large V_{DS} is applied. This equation assumes the output impedance is infinite, or the output current I_D doesn't change when V_{DS} is large. This may not be a good model for a realist MOSFET under all conditions.

Another quantity of interest is the transconductance gain g_{fs}

$$g_{fs} = \frac{\partial I_{DS}}{\partial V_{GS}} \quad (11)$$

Pre-Lab Questions

Please complete the following questions prior to coming to lab. They will help you prepare for both the lab and the pre-lab quiz (Found on D2L).

- 1.) Read through the entire lab writeup before beginning
- 2.) What is the **specific** goal of this lab? Exactly what question(s) are you trying to answer? Be as specific as possible. ("To learn about topic X..." is **not** specific!)
- 3.) What **specific** measurements or observations will you make in order to answer this question?

Experiment

In this experiment you will characterize the behaviour of a MOSFET transistor by measuring drain current, I_D , under various V_G and V_{DS} conditions, and you will compare the results observed against values provided from manufacturer data sheets.

Recall that in Lab 3 we measured the I-V curve for a PN junction by manually varying a current supply and recording the resultant current and voltage across the diode. In this laboratory we will need to collect similar curves. However, we will automate the process somewhat by building an I-V curve tracing tool using an Arduino microcontroller and certain voltage supply and current sensing "breakout boards."²

Initially you will wire your equipment and use a short program to acquire a graph of I_D as a function of V_{DS} for various gate voltages, V_G .

As well, you'll revise your program to acquire a graph of I_D vs V_G for a fixed large value of V_{DS} . From this graph you'll determine the transconductance gain of your transistor (often written g_{fs}) and compare against the datasheet specification.

Finally, based on your characterization of your MOSFET transistor, you will design a simple transistor switch to provide power to a small load device like a light bulb or motor.

²Note, the objective of the lab is to determine properties of the MOSFET transistor. The Arduino I-V curve tracer you will build and program serves as a tool to help achieve that end goal. However the tool should not be a main focus of your attention when composing your report.

Apparatus

- MOSFET transistor (2N7000, ZVN2106, or similar)
- Arduino (Adafruit Metro Mini, or suitable alternative version)
- Adafruit MCP4745 12-bit 5V DAC breakout board (2 required)
- Adafruit INA219 DC High-Side Current Sensor breakout board
- MCP4002 dual op-amp (CMOS)
- TIP41C BJT transistor and external power supply
- Electrical prototyping board
- Barrel jack receiver
- 9V power supply (barrel jack)
- Jumper wires

I-V Curve Tracer Circuit Layout

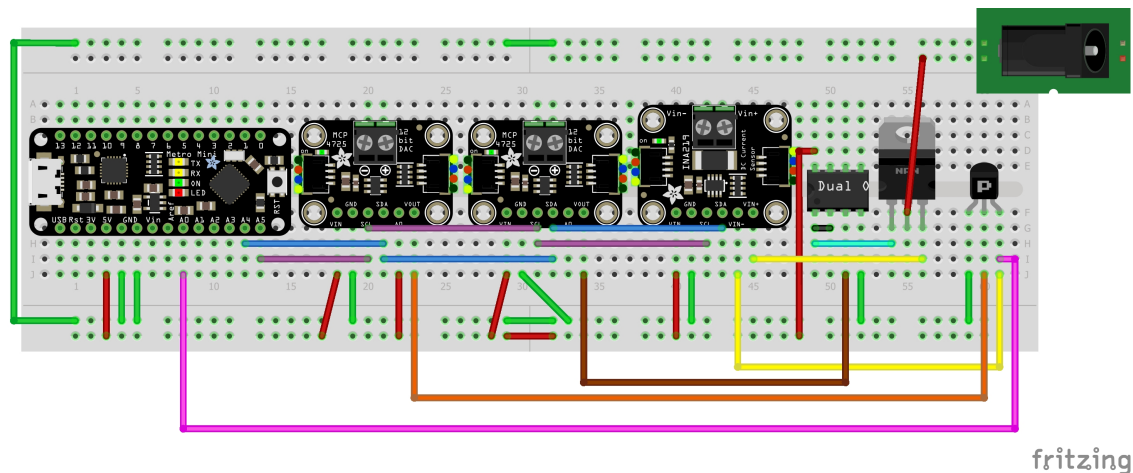


Figure 5: Layout of electrical components and wiring for I-V curve tracing application.(Version shown uses Adafruit Metro Mini)

Arduino-based I-V Curve Tracer - Configuration and Principles of Operation

The goal of the first part of the experiment is to capture I-V curves where we monitor drain current, I_D , while varying drain-source bias voltage, V_{DS} , for various gate voltages, V_{GS} .

The goal of the second part of the experiment is to measure I_D while varying V_{GS} as V_{DS} is held at a fixed voltage. The configuration of the circuit is the same for both parts of the experiment; only code changes are required.

Transistors are often static sensitive, so it is best to avoid touching their pins. Furthermore, it is important to know which pin of a MOSFET is which. For the MOSFETs used in this lab, the pins are as shown in Figure 6, though it is always good to check the component sheet to be certain.

To configure the electrical components, one DAC board will be used to supply our V_{DS} voltage (dac_vds). Note that we route our dac_vds V_{out} pin through an MCP6002 op-amp stage, followed by a TIP41C BJT transistor. The op-amp stage serves to “isolate” any DAC output resistance from influencing voltages levels at later stages of the circuit (i.e. the DAC could act as a parallel resistor and create a “voltage divider” effect). However, despite that the op-amp stage isolates our dac_vds voltage, the MCP6002 chip cannot “source” enough current to properly drive our MOSFET. To compensate, a TIP41C transistor stage is added to increase the current available to the MOSFET. This is at the expense of a small drop in our voltage due to the base-emitter PN junction.

After the BJT stage, a wire connects to the V+ input on the INA219 Current Sensor board, and a wire from the V- output of the INA219 connects to the MOSFET’s drain. The MOSFET’s source pin is connected to ground.

The second DAC board is used to supply our V_{GS} voltage (dac_vgs). There is no need to “isolate” this DAC because the MOSFET gate provides extremely high (effectively infinite) insulation, or resistance, so no voltage dividing effect is expected. The V_{out} pin from dac_vgs is therefore wired directly to the gate pin of our MOSFET.

Important Equipment Note: The BJT requires a power source greater than the 5V provided by the MetroMini. You will use a 9V power supply. This power supply should only be plugged in when you are actively collecting data. If you are making changes to your code/wiring you should unplug the power supply.

Arduino Code

We will need two short programs to capture the MOSFET IV Curves.

The first program can be found on D2L as MOSFET_IV.txt. You should alter the code to collect a data sets for several different gate voltages between 0 and 5V.

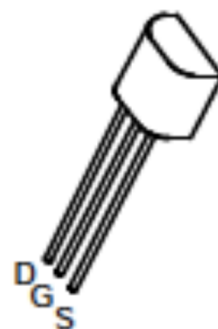


Figure 6: MOSFET pinout for 2N7000 and ZVN2106

Your second program will share the same structure as the first, however it will monitor I_D vs V_{GS} for a fixed large value of V_{DS} . As a reminder, the wiring doesn't need to be changed for this portion of the experiment. Instead, you will be changing the code provided to hold V_{DS} constant (at 5V) and sweep V_{GS} from 0 – 5V while measuring I_D . **Note: You can't measure V_{GS} using the INA219, how else can you print out the voltage?**

Important Things to Remember

- Select the correct board in the Arduino app (“Arduino Mini”)
- Select the port you are uploading to (USB)
- **ONLY** power the circuit when you are collecting data

Presentation of Results

You will capture a set of I_D vs V_{DS} data from your program by opening an Arduino serial terminal. Copy and paste the data for a given V_{GS} condition into a spreadsheet. You should be able to plot several I_D vs V_{DS} curves on the same plot to illustrate the effect of changing V_{GS} . (You'll use the plot to help design a basic MOSFET transistor switch.)

Once you've modified your program to capture I_D vs V_{GS} (for a large value of V_{DS}) you'll similarly capture your results by pasting the data from a serial terminal into a spreadsheet. You'll use this graph to determine both V_T and g_{fs} for your transistor. You will then determine whether it falls within manufacturer specification.

Analysis

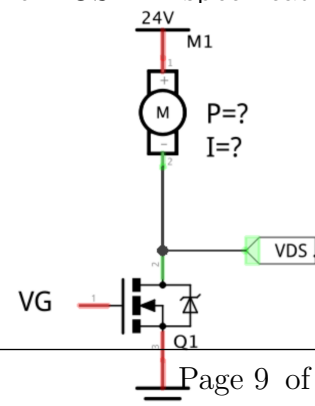
Comment on whether your I_D vs V_{DS} curves and I_D vs V_{GS} curve are as expected. Explain the shape of the curves of I_D vs V_{DS} as V_{DS} increases.

Where appropriate, compare your results to manufacturer specification: determine the slope of the plot of I_D vs V_{GS} for $V_{GS} > V_T$ (linear region). Compare against the tolerances provided in the datasheet. (Note that transistor parameters can vary dramatically, and therefore values of V_T and g_{fs} may be significantly different than typical values and still fall within specification.)

Note that in our code we have intentionally “pulsed” our MOSFET on for a few milliseconds at a time to take measurements, and then powered V_{DS} or V_G off for some length of time. Why is this recommended for our measurements? Refer to the MOSFET specification sheet in your answer.

Wrap Up - MOSFET Switch

Based on your I_D vs V_{DS} plot and your value of V_T , conceptually design a simple transistor switch as shown in Figure 7. The intent of the switch circuit is to turn on a resistive load like a light bulb or small motor. Assume the power demand from



the load is equal to the last two digits of your student number (if working in pairs, select one partner's student number), i.e. for student number 123426, use a power load of 6W. Assume a supply voltage, V_{dd} , of 24V.

Recall that $P=IV$. Select a suitable current (and therefore V_G) from your I_D vs V_{DS} plot to supply the appropriate power to the circuit load.

How much power will be dissipated across the MOSFET drain/source? Will the MOSFET survive? Explain.

Report

Labs will be completed in groups, you will enroll in a group with your lab partner at the beginning of each lab session. Each group will submit a single report through the assignment section on D2L.

- **Introduction**

- What is the experiment's objective?

- **Theory**

- You may be able to show a derivation of the physics you're investigating, or you may want to reference a source that provides a description/equation representing the physics you're investigating.
- You may want to provide graphs that illustrate or predict how you expect the system under study to behave.

- **Procedure**

- Explain the systematic steps required to take any measurements.

- **Results and Calculations**

- Tabulate your measurements in an organized manner.
- Based on your procedure, you should know what your tables
- Provide examples of any calculations.

- **Discussion and Conclusions**

- Discuss the main observations and outcomes of your experiment.
- Summarize any significant conclusions.

- **References**

- **(Appendices)**

References

- 1.) Simpson, Robert E. (1987). *Introductory Electronics for Scientists and Engineers*. 2nd ed.
- 2.) Streetman, Ben G. (1995). *Solid State Electronic Devices*. Upper Saddle River, NJ: Prentice-Hall, 4th ed.

Appendix 1: Introduction to Arduino

Arduino is a computing platform that consists of a hardware device (a microcontroller known as the board) and a software package to operate it. Arduino users can write programs that read information from sensors and can control output devices like motors or lights. The platform is relatively easy to use even for individuals with minimal programming experience.

The Arduino programming language is based on C/C++ and is used to write code and communicate with the board. An Arduino program, also known as a sketch, is written in an open source software package known as the integrated development environment (IDE). The Arduino IDE is available online and can be downloaded from the following link: <https://www.arduino.cc/en/Main/Software>.

The IDE allows users to Verify their program before it is uploaded to the board to ensure that they are free of any syntax errors. Figure A1 above shows the interface of IDE and a simple sketch used to blink a light emitting diode (LED).

When your sketch is complete and compiles successfully, it can be uploaded to the board using the Upload button shown in Figure A1.

Installing Arduino Libraries

In some cases when additional circuit components are used, special library files might be required. In this experiment, you will be using an INA219 DC current sensor and an MCP 4725 digital-to-analog converter (DAC). Each of these “breakout boards” give the Arduino extra capabilities but require installation of an Arduino driver library file which can be found and downloaded from the following link:

<https://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use/arduino-libraries>.

When downloading the libraries is complete, they must be moved into Arduino’s library folder, usually located under Documents > Arduino > libraries. A name change might be necessary to match the name used in the sketch script.

Uploading Your Code and Running Your Program

Now that you have installed the Arduino IDE and the required library files, restart the IDE and plug your Arduino board using the USB cable provided.

To upload your program you’ll need to tell the IDE which type of Arduino you’re using (typically and Arduno Uno or Duemillanova). Go to the Tools menu and select your board model.

You’ll also need to tell the IDE which serial port your Arduino board is connected to. Again this can be selected under Tools > Serial Port menu. For Macintosh users, select the port that begins with /dev/cu.usbserial-. Windows users might select a COM port listed.

Finally, upload the sketch to the board. Your Arduino will continuously run your sketch until it is unplugged or a different sketch is uploaded. If you unplug your Arduino your sketch will remain installed and will automatically restart next time you power up.

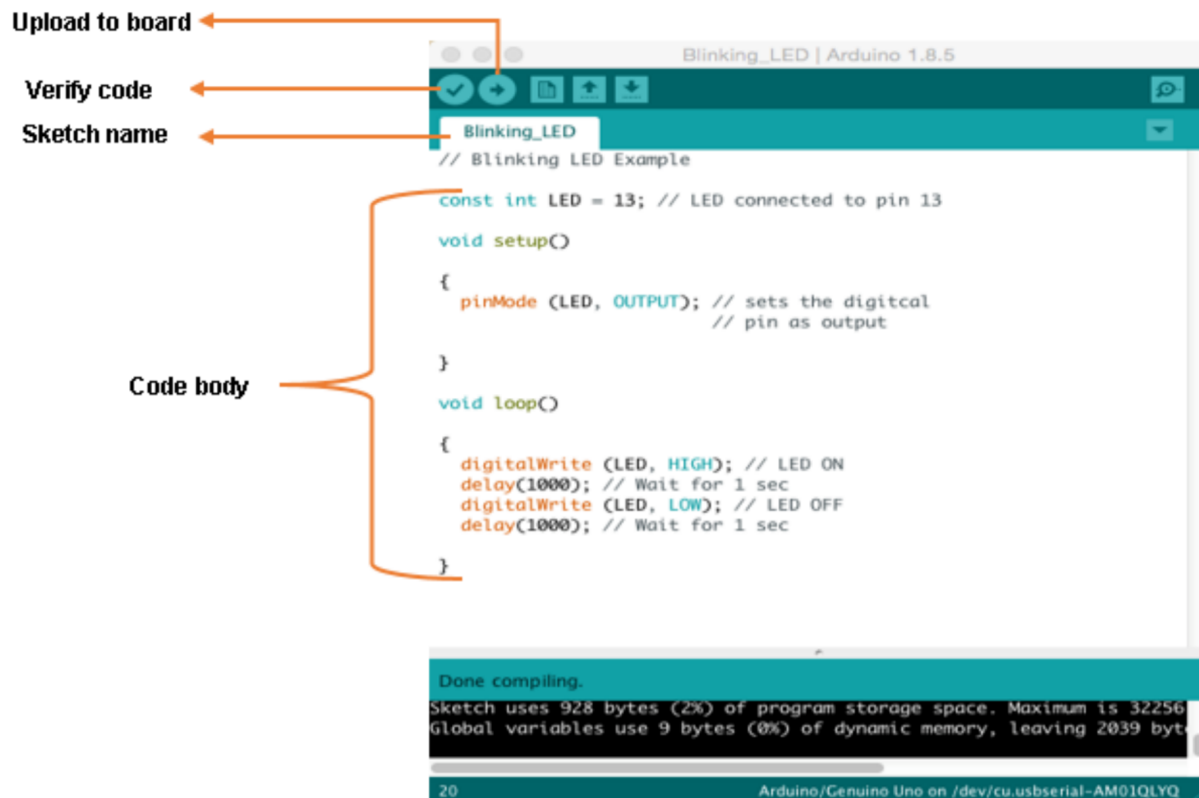


Figure 8: Interface of Arduino integrated development environment (IDE).

Go build an I-V curve tracer. Have fun!