Light Emitting Diode IV Characterization

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.

- Light Emitting Diodes (Neamen, 14.5)
- Band Theory of Solids (Serway, 42.5)
- Semiconductor Devices (Serway, 42.7)

Introduction

In this lab you will build a basic current-voltage characterization tool and determine the IV response of a set of light emitting diodes (LEDs) of various wavelengths. Based on the turn-on voltage of the LEDs you will be able to determine Planck's constant, h.

Theory

LEDs are essentially a unique type of pn junction diode. The important distinction in the case of the LED though is that recombination of carriers results in the emission of a photon. The wavelength of the photon is dependent on the energy band gap, E_g, of the semiconductor material from which the diode is fabricated.

Recall that the current, I, across a biased diode is given by

$$I = I_s(e^{(e\Delta V/k_B t)} - 1) \tag{1}$$

Under forward bias, the potential barrier between the p and n regions is reduced, as is the depletion width of the junction. As eV approaches E_g, majority carrier electrons from the n-type material with sufficient energy are able to diffuse across the transition region into the p-type material, where they are, in effect, an injection current of minority carriers. The same occurs for holes from the p-type region diffusing across the transition region into the n-type material.

These "injection currents" result in an excess of minority carriers into the p- and n type regions, and the carriers must eventually recombine with majority carriers. If the semiconductor is a "direct band gap" material, the recombination can occur directly via the transition of an electron from conduction to the valence band and the excess energy from the transition is released in the form of a photon, as shown in Figure 1.

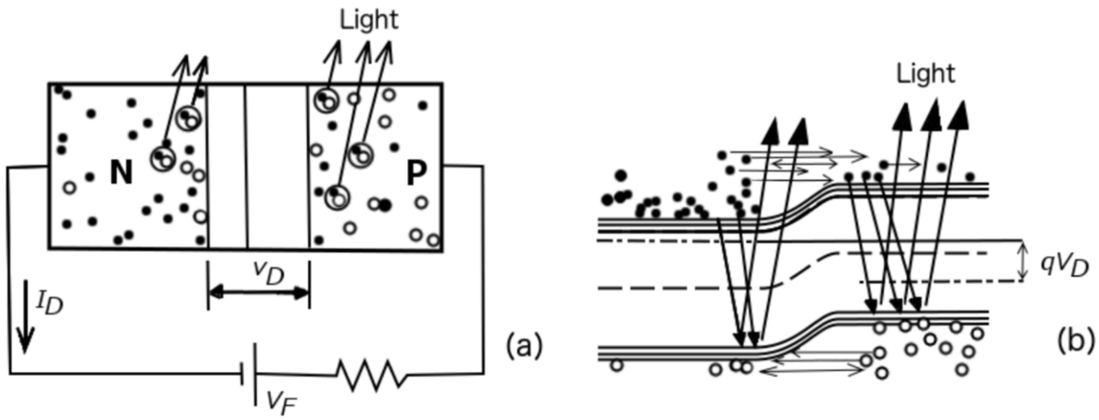


Figure 1: (a) PN junction LED circuit showing doped regions, transition region, and charge recombination. (b) Band diagram illustrating flow and recombination, as well as photon emission. Modified from [1].

The energy of the emitted photon then depends on the semiconductor's band gap according to

$$E_g = h\nu = hc/\lambda \tag{2}$$

where ν is the photon frequency, λ is the photon's wavelength, h is Planck's constant, and c is the speed of light. Furthermore, the "turn-on" or threshold voltage of the LED, V_t, at which forward conduction begins, is related to E_g according to

$$eV_t = E_g \tag{3}$$

where e is the unit charge of an electron.

LED Band Gap Energies

A great deal of engineering has gone into the production of semiconductor materials such that LEDs cover a broad set of wavelengths and exhibit a wide range of optical intensities. In many cases they have very favorable power consumption properties when compared to conventional light sources.

Figure 2 shows the band gaps of several semiconductor materials. Many III-V semiconductors exhibit convenient band gap energies corresponding to IR, visible, and UV wavelengths. Wafers of GaAs (E_g=1.43eV) and InP (E_g=1.35eV) are widely commercially available, though in smaller wafer diameters and production volumes than Si.

Interestingly, tertiary (e.g. InxGa1-xAs) and quaternary (e.g. InxGa1-xAsyP1-y) alloys of III-V semiconductor materials can be made to exhibit various tailored wavelengths, and in many cases these alloys can be grown directly onto GaAs or InP substrates. This opens up a multitude of LED applications.

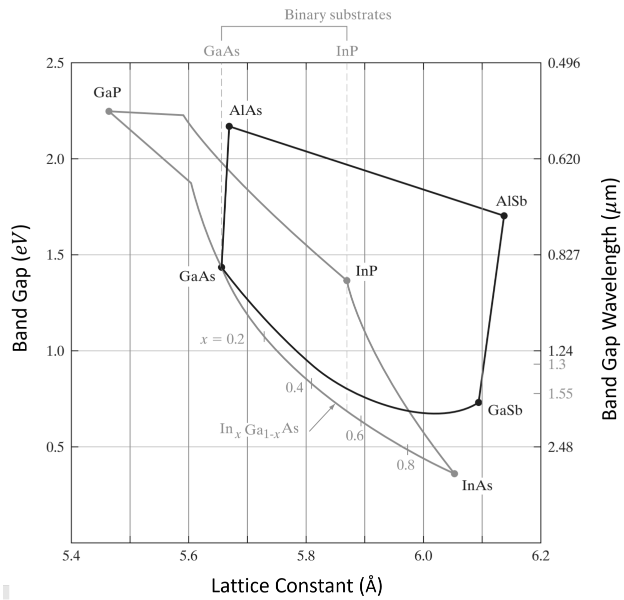


Figure 2: Band gap energies and emission wavelengths of several III-V semiconductors and alloys. Modified from [2].

Direct vs Indirect Band Gap Semiconductors

Note that only semiconductors exhibiting “direct” transitions can emit photons efficiently, and so only these materials are used in the fabrication of LEDs. A direct transition means that the carrier recombination process involves only the emission of a photon. Many group III-V and II-VI binary exhibit direct transitions.

Notably, Si and Ge, which have many favorable electronic properties, exhibit indirect carrier recombination, and are therefore poorly suited to light-emitting applications.

In indirect transitions, the recombination of charge carriers is a multi-step process in which energy is thermally transferred to an acoustic phonon and then to a photon, or occurs entirely non-optically through phonon generation alone.

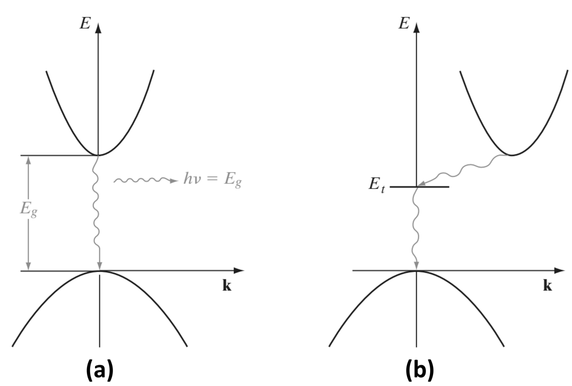


Figure 3: (a) Direct transition with photon emission. (b) Indirect emission. Transition requires a change in wave vector, k, which gives up energy to the lattice (vibrational phonons) instead of emitting a photon. Modified from [2].

(Attempts have been made to engineer light emitting impurities into Si and thereby merge the electronic benefits of Si with photonics, which, if successful, could merge with advanced Si device fabrication techniques and electronic properties to create fully integrated opto-electronic devices.)

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?

Apparatus

- Arduino microcontroller (Adafruit Metro Mini)
- Adafruit MCP4745 12-bit 5V DAC breakout board
- Adafruit INA219 DC High-Side Current Sensor breakout board
- MCP6002 dual op-amp (CMOS)
- TIP41C or 2N3904 BJT transistor (note different pinouts), or equivalent.
- Electrical prototyping board
- Jumper wires

Experiment

Assemble an IV characterization tool using an Arduino microcontroller, 5V digital to-analog converter (DAC), and current sensor, as in Figure 4.

Note: You will need to finish constructing this circuit using jumper wires. When you are done the experiment please remove the jumper wires and leave other wires in place.

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

The goal of this experiment is to measure the current across an LED as the voltage applied to the LED increases from 0V toward 5V. You will need to write a short program to control the voltage supplied by the DAC and to take current measurements.

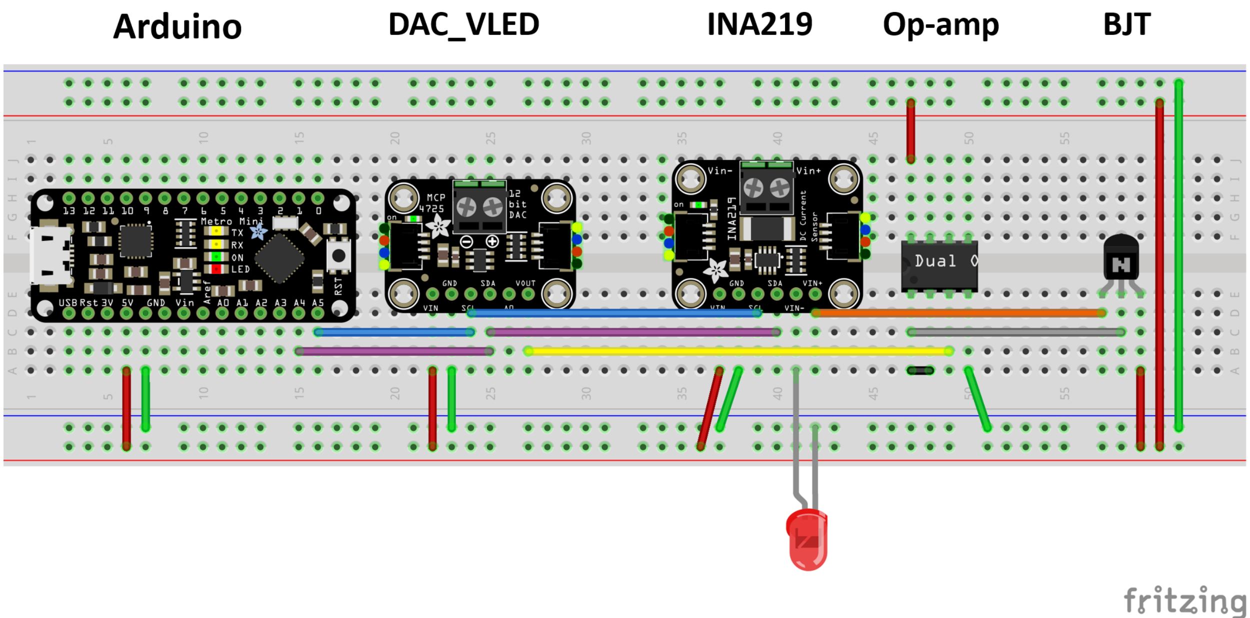


Figure 4: IV characterization tool. MCP4725 DAC supplies voltage (0-5V) to an MCP6002 op-amp isolation buffer. Voltage out of MCP6002 is routed through a BJT transistor current source (since MCP6002 can source only 20mA). INA219 current sensor measures current and voltage to LED, which is wired from INA219 to GND.

Electronic Components

1.) A DAC board will be used to supply our V_{DS} voltage (dac_vds).

2.) The supplied voltage is routed out from the dac_vds (V_{out}) pin through an MCP6002 op-amp stage.The op-amp serves to "isolate" any DAC output resistance from influencing voltage levels at later stages of the circuit (i.e. the DAC could act as a parallel resistor and create a "voltage divider" effect).

3.) The voltage is then routed out of the op-amp and into a 2n3904 npn BJT transistor. The role of the transistor is increasing the current available to the op-amp ensuring it can stabilize the signal. This comes as the expense of a small drop in voltage due to the base-emitter PN junction.

4.) After the BJT stage, a wire connects to the V+ input of the INA219 current sensor board, and the LED will complete the circuit connecting the V- output of the INA219 to ground.

5.) The INA219 will then be able to read the current going into the LED and the voltage at the positive pin of the LED relative to ground.

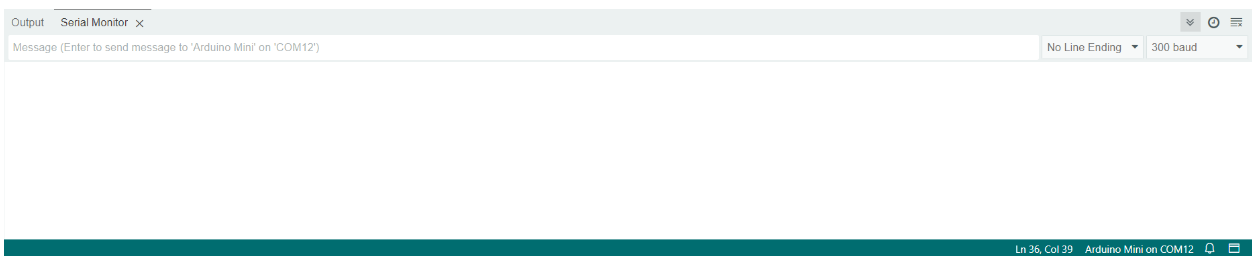
Note: Transistors are often static sensitive, so it is best to avoid touching their pins.

Getting Started with the Arduino IDE

Open the Arduino IDE, you will be working with a 2.x version of the IDE. Once the IDE opens wait 30 seconds to a minute for it to download any updates.

In the top left of the IDE you will see a check mark button (Verify), an arrow button (Upload), a play button (debug, you won't need to use this) and a drop down menu. Click the drop down menu and select the COM port labeled as USB. A pop-up will appear, select Ardunio Mini as the board type and click OK.

Next you will need to change the baud rate of the serial monitor/plotter. To do this select tools in the top left corner of your screen, then select serial plotter. The bottom of your screen should now change to display:



In the top right of this window you will see a baud setting (in the screen shot this is set to 300 baud), click that drop down and select 9600 baud. This will sync your serial monitor/plotter with the output of the Arduino.

Arduino Code

You will write a short program to capture the LED IV Curves.

Download the file LED_IV.txt from D2L and use it as a starting point.

Use a "for loop" to step the DAC voltage from 0-5V in small voltage increments. Pulse the LED on only briefly (a millisecond or so) and be sure to turn the LED off for at least 10 times longer than it was on to allow cooling. Include an instruction in the loop to end the loop if the current exceeds 100mA. More detailed instructions on how to accomplish these steps are included in the .txt file as comments.

When you are ready to test your code press the Verify button. The IDE will notify you if there are any issues with compiling the code, look at any lines referenced in the error log and see if you can fix the issue. If you are unable to debug your code you can ask your TA for help.

Once your code is successfully compiling you can upload it to the board and test it. If wired correctly you should see your LED begin to flash growing brighter and then turning off again.

To monitor the data that is being collected you can open a serial plotter, in the IDE select the tools drop-down and click Serial Plotter. You will see a live plot of the voltage and current in your system. As your LED starts to flash you should see a dramatic increase in current measured across the INA.

Once you are confident in your results you should export voltage and current data for a complete run. Close the serial plotter and open the serial monitor. Turn off auto-scroll by clicking the auto-scroll button  just above the baud setting. Now you can copy (cmd+c) and paste (cmd+v) data directly out of the Serial Monitor and into excel for further plotting and analysis.

Plot your data in excel and confirm that it is coherent and clearly displays the turn on voltage of the LED.

If you are satisfied with your data you can move on to a new colour of LED. You will collect data for a total of 6 LEDs with unique wavelengths. LED wavelengths are supplied in the Appendix.

Note: You don't have to change your code when you move to the next LED, simply swap the LED and press the "Clear Output" button  found beside the autoscroll button.

Analysis

Plot current as a function of voltage for each LED data set, it is recommended that you make your current axis logarithmic. You can now identify the "turn-on" voltage, where the LED just begins to exhibit current above noise level,this voltage will be taken as the band gap energy of the LED. In your report you should identify the criteria you used to determine the turn on voltage and include a single plot with all LED data plotted as current vs voltage.

Plot your results for E_g as a function of LED photon frequency, ν. Include appropriate error bars. From the plot E_g vs ν determine Planck's constant and compare to its accepted value.

Wrap Up

The following questions are designed to make sure that you understand the physics implications of the experiment and also to extend your knowledge of the physical concepts covered.Your report should answer these questions in the noted section in a seamless manner.

1.) [Theory] Explain what "turn on" voltage means as it relates to semiconductor diodes, specifically noting what will be observed in the case of an LED. What is physically occurring in the diode as this voltage is crossed. **Note:** You may need to refer back to Lab 3 and apply your knowledge of semiconductor physics to adequately explain.

2.) [Procedure] In the apparatus section there are two different transistors mentioned.

Look up the pin out of both transistors and comment on how your wiring would change if you had used a TIP41C transistor.

3.) [Discussion] Why is it important to put a *break* in the “for loop” when a certain current threshold is reached? What would happen to the circuit if you allowed the Arduino to sweep the full 5V (or even further)?

4.) [Appendix] Add additional comments to the code you used for data collection, your comments should explain what each line is doing and which portion of the circuit the code interacts with. Submit this commented code as an appendix in your lab report.

References

[1] Dimitrijev, Sima (2012). *Principles of Semiconductor Devices.* New York, NY: Oxford University Press, 2^{nd} ed.

[2] Streetman, Ben G. (1995). *Solid State Electronic Devices.* Upper Saddle River, NJ: Prentice-Hall, 4th ed.

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)**

Appendix 1: LED Wavelengths

Colour	Wavelength
Purple/UV	395 nm
"White"	450 nm
Blue	461 nm
Green	580 nm
Yellow	591 nm
Red	658 nm
IR	940 nm

Table 1: The approximate wavelengths of the LEDs available in the lab. "White" is in quotation marks because there is no singular wavelength for white light and as such the LED has a multi-chromatic output. However, the LED still has a turn on voltage where it begins to transmit current, that turn on voltage corresponds to the listed wavelength.

Appendix 2: 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 a Ardiuno Metro Mini). 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.

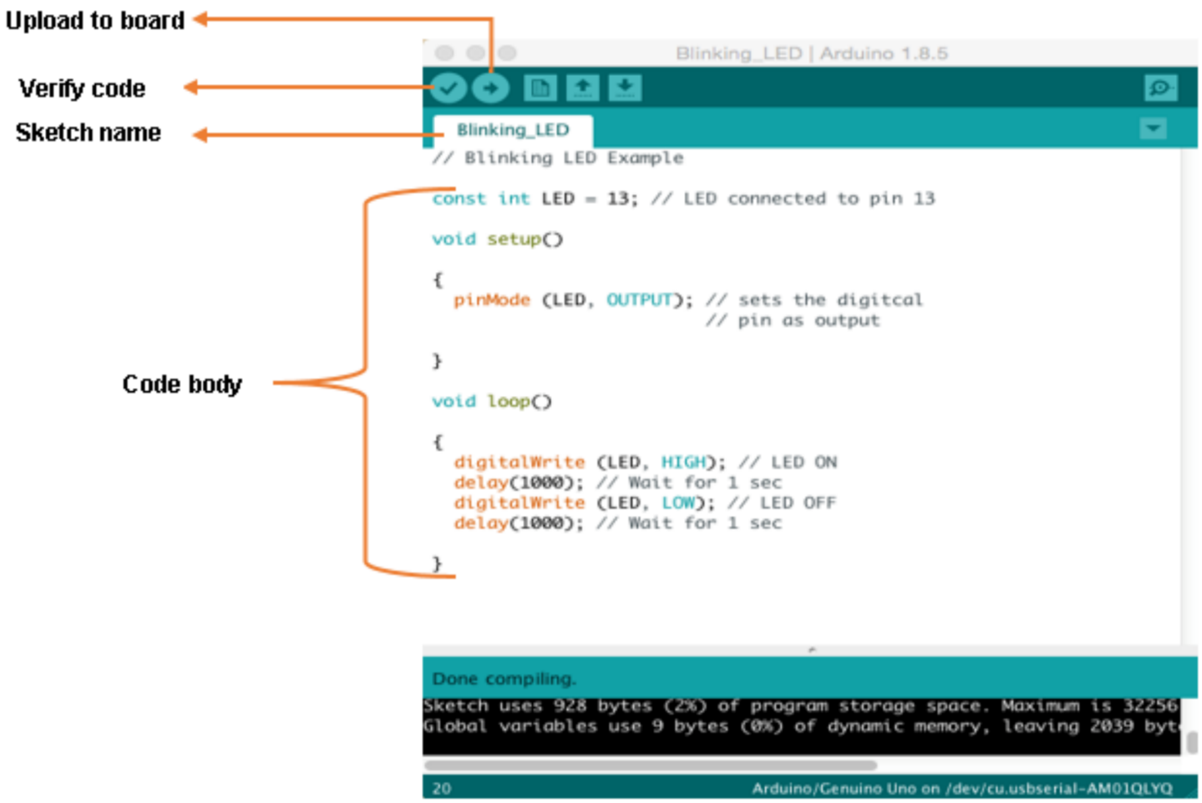


Figure 5: Interface of Arduino integrated development environment (IDE).

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.

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