Lecture 02

Using LEDs and the Teensy

Agenda for Today's Lecture

01 Passsive Component Recap

02 LED's Intro for Lab 1.1 (1st out of 4 parts)

03 Binary and Hex

04 Teensy Intro for Lab 1.2 (2nd out of 4 parts)

05 Pulse Width Modulation (PWM)

Announcements

- 10 people haven't picked up a kit yet. Sign up for pickup on the sheet (bottom of canvas 510 home page)
  - Check kits for missing components. (e.g., headers)
- Lab 1 will go out soon for those who want to start early.
- If you have questions you may raise hand, or you may post to chat to class or to TA who is monitoring
  - Questions may be anonymous if desired.
  - Some questions will be answered during lecture.

C Programming Language?

- Recitation Friday will cover basics of C programming.
- Quick online tutorials (do this before Friday)

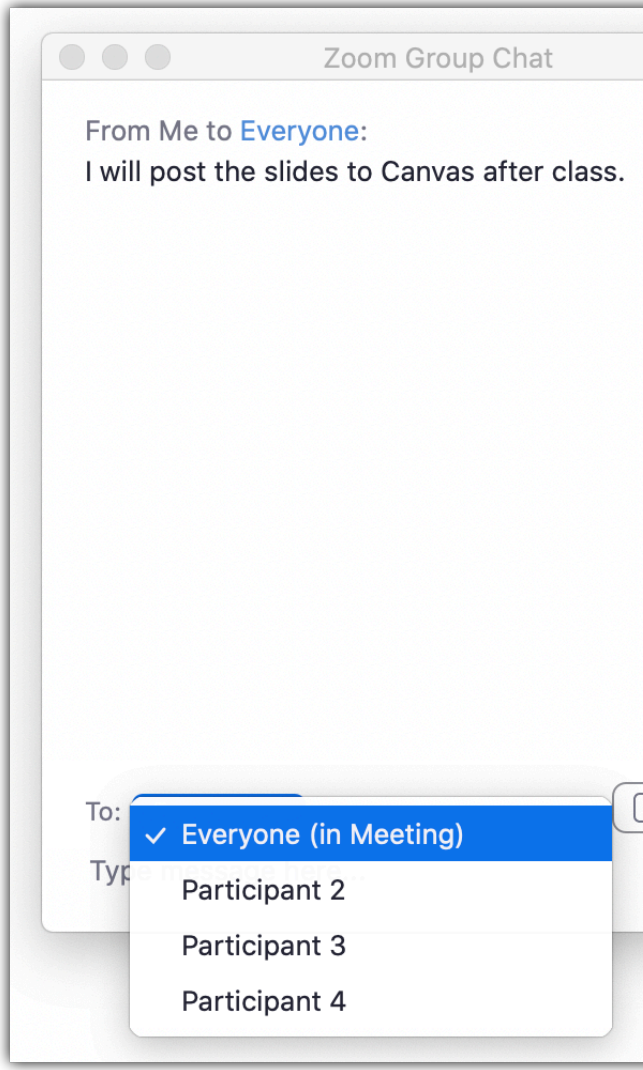
A. http://www.tutorialspoint.com/cprogramming/
B. http://www.learn-c.org/

Class participation (graded)

- QUESTIONS IN GREEN
- Either live annotate or *private* chat to me.
- We will check CHAT logs to verify class participation
- We are looking for any answer, not necessarily the right one.
- Possible extra pts for corrections to early wrong ones.
- For asynchronous video watchers:
- Get link to the "Activity Sheet" on Canvas Assignments
- Follow along and (re)submit finished sheet to Canvas

In Private Chat to me:

- What times of day will you prefer to do 510 Lab work?
  - Ex: Evenings
  - Ex: Weekend afternoons
  - Ex: 2PM to 5PM

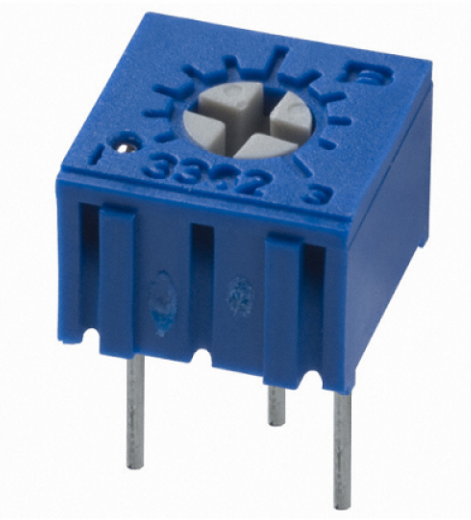


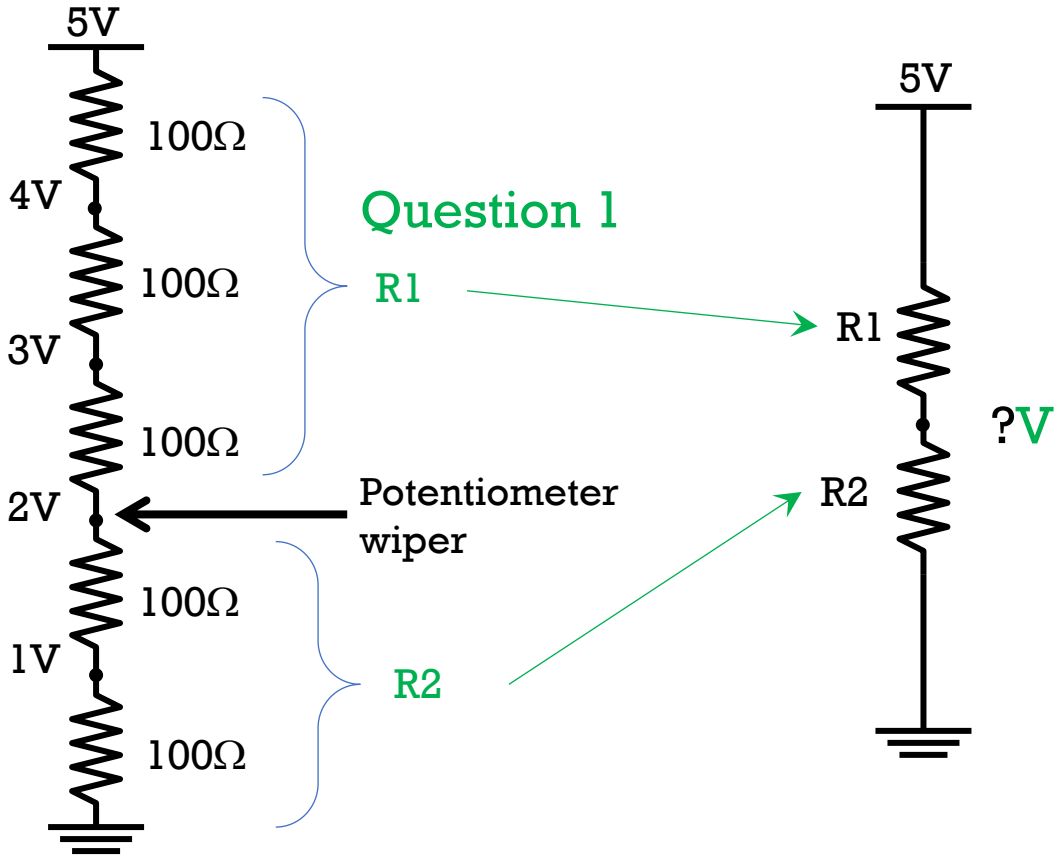
Click on To: "Everyone"
select "Mark Yim" <privately>

01

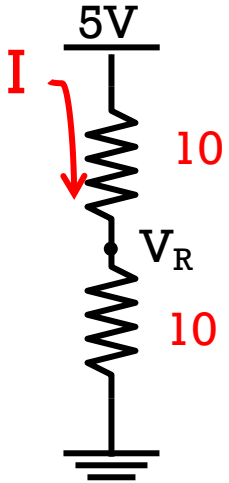
Passive Component Recap

Voltage dividers/potentiometers





Voltage divider



1/4W Resistor

$$V_R = 5\,\frac{R_1}{R_1 + R_2} = 2.5$$

$$I = \frac{5}{R_1 + R_2}$$

Power in Watts = Volts * Amps

$$\text{Power} = VI = 2.5\,\frac{5}{R_1 + R_2} \qquad R_1 = 10,\ R_2 = 10$$

$$\text{Power} = 12.5/20 = 0.625\text{W}$$

For 1/4W Resistors, W < 0.25 ->

$$W = 12.5/50$$

For nominal 5V cases
Use 1/4W resistors
~25Ω << 1MΩ

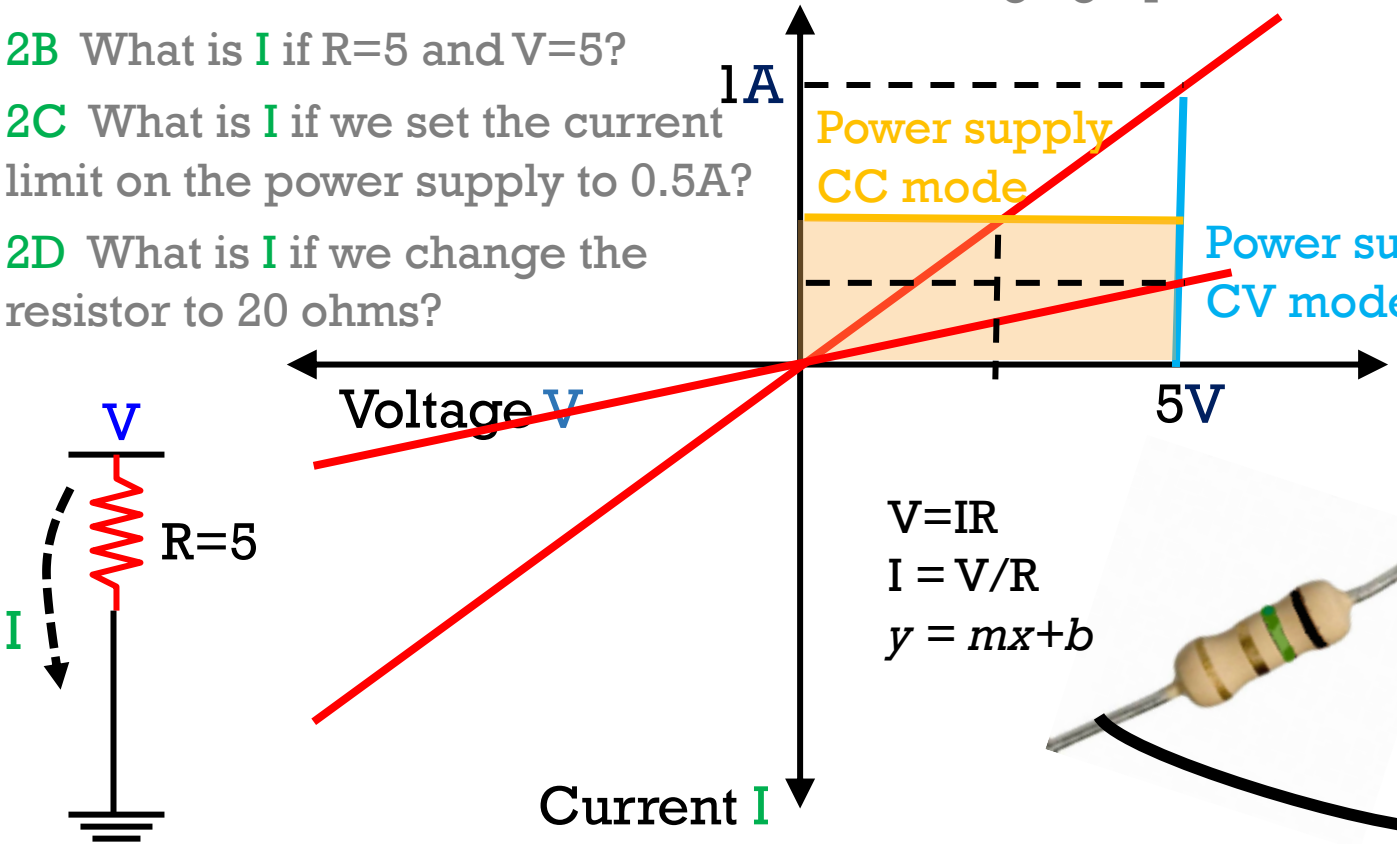
Resistance on Voltage/Current graph

Question 2A Draw and hold the current/voltage graph for a resistor

2B What is I if R=5 and V=5?

2C What is I if we set the current limit on the power supply to 0.5A?

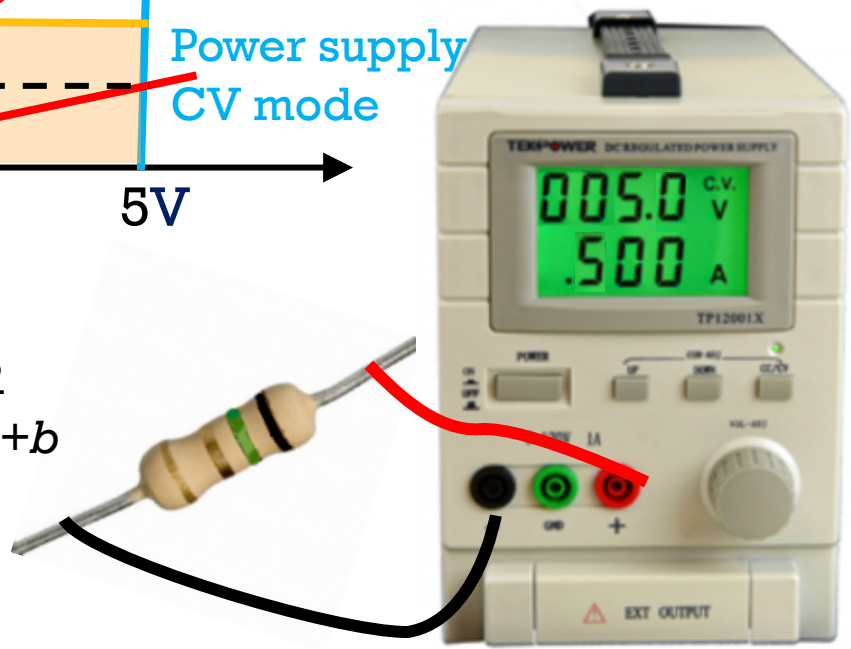
2D What is I if we change the resistor to 20 ohms?



V=IR

I = V/R

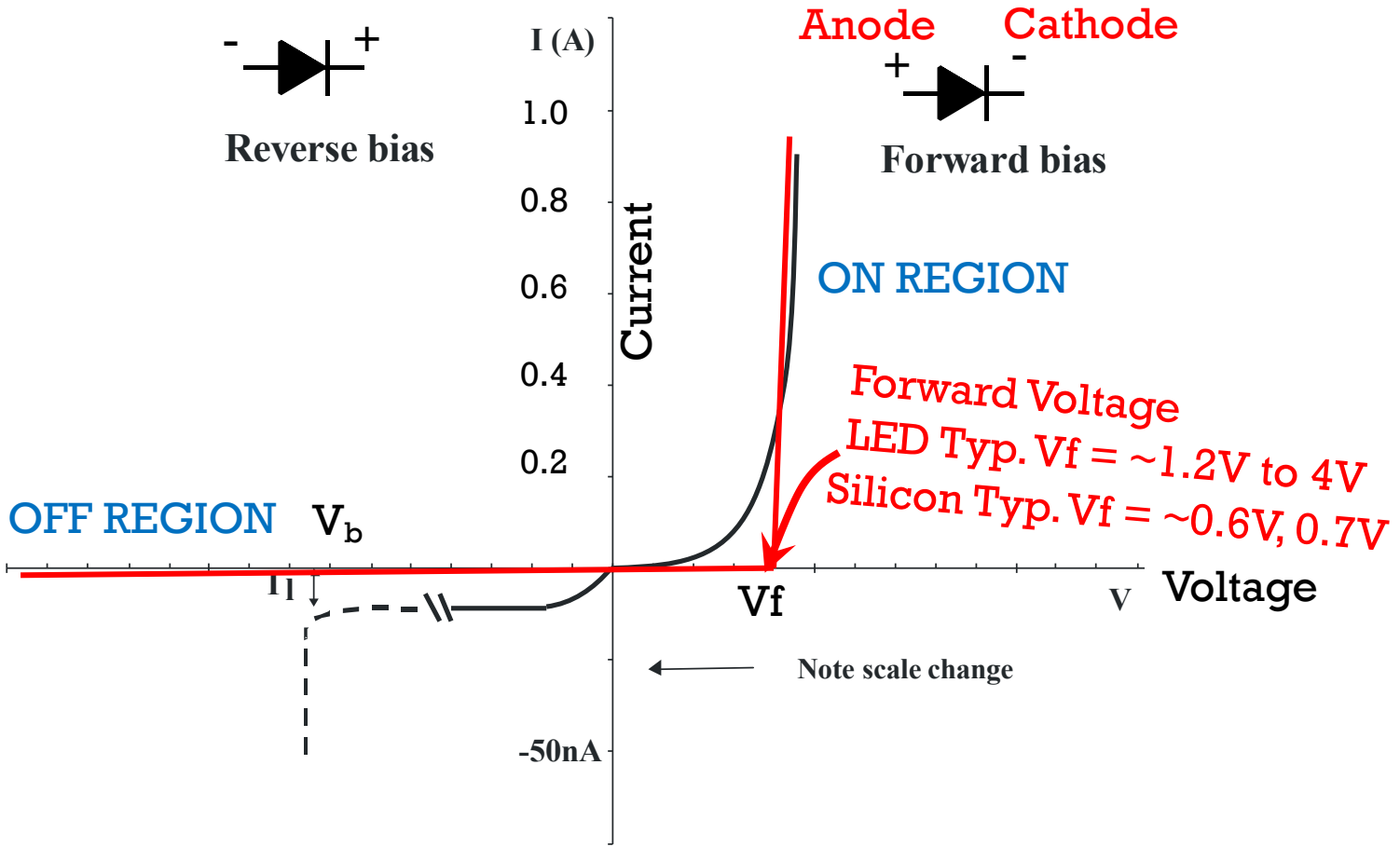
$y = mx+b$



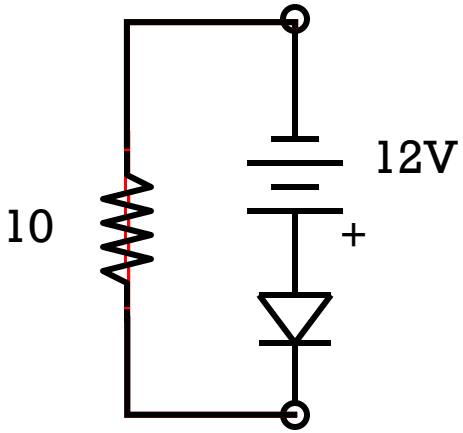
02

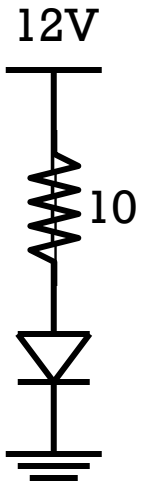
Intro for Lab 1.1 LED's

Standard Diode Behavior



Question 3: C

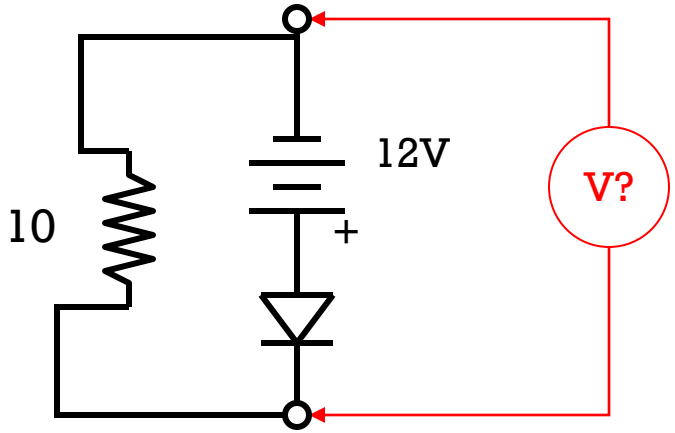




Why do we need a resistor?

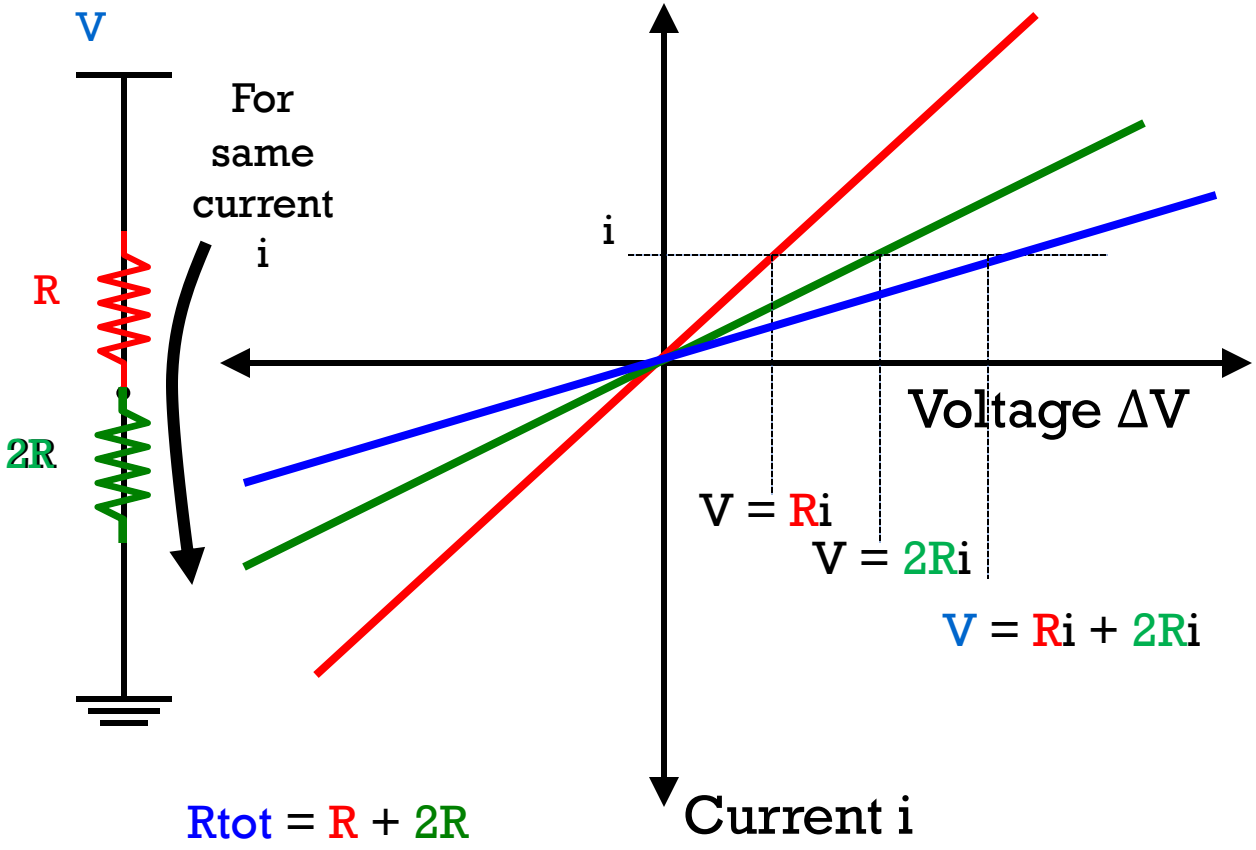
Is the LED on or off?

Question 4: C



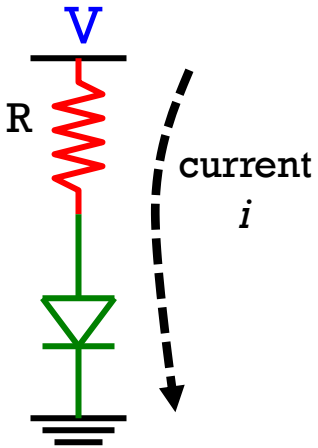
What is the voltage V?

Devices in series

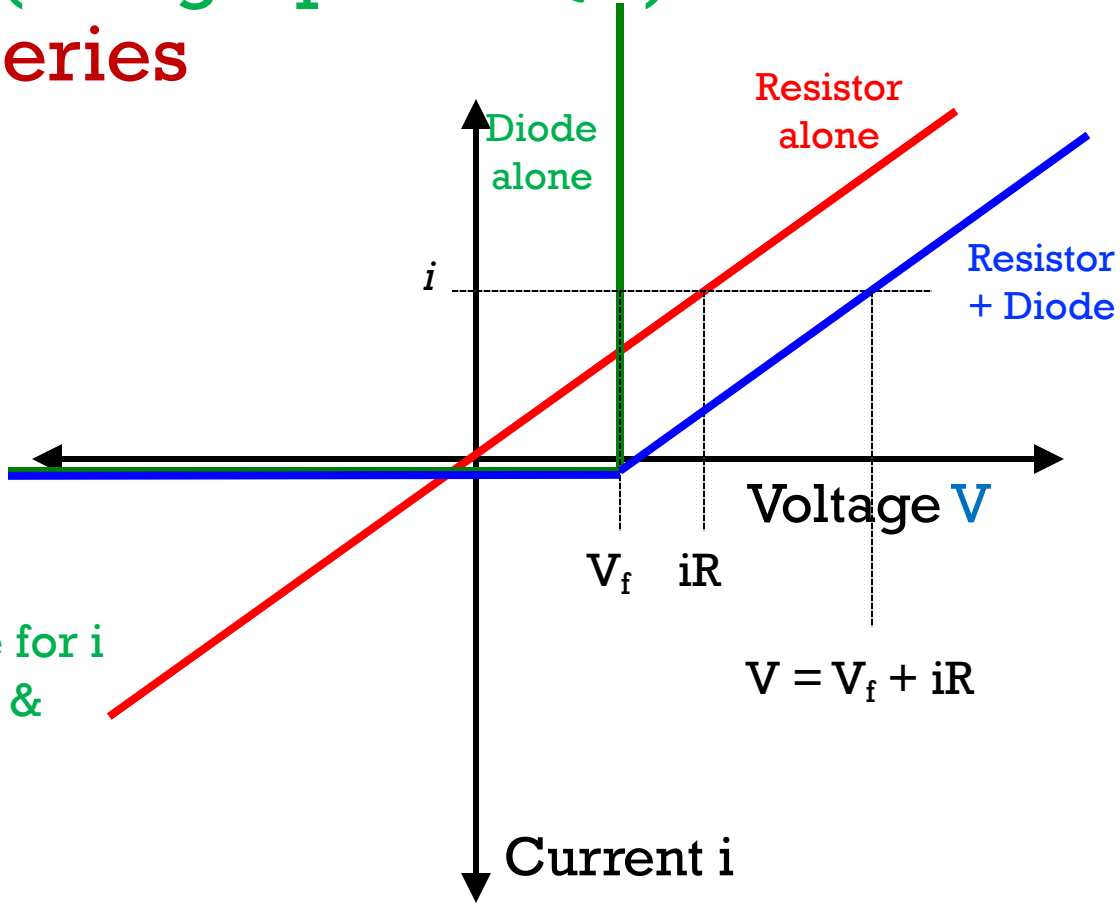


Question 5: (use graph for Q2)

Devices in series



Draw and hold the curve for i as V varies for the diode & resistor in series.



Diodes in series

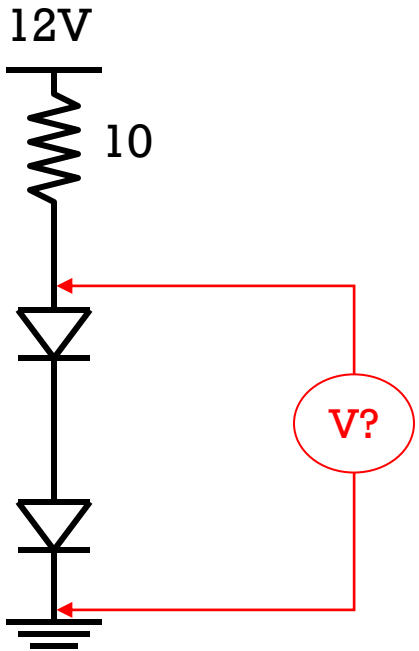
The amount of light from an LED generally increases with current.

Q6: What happens to total amount of light emitted if we put 2 in series?

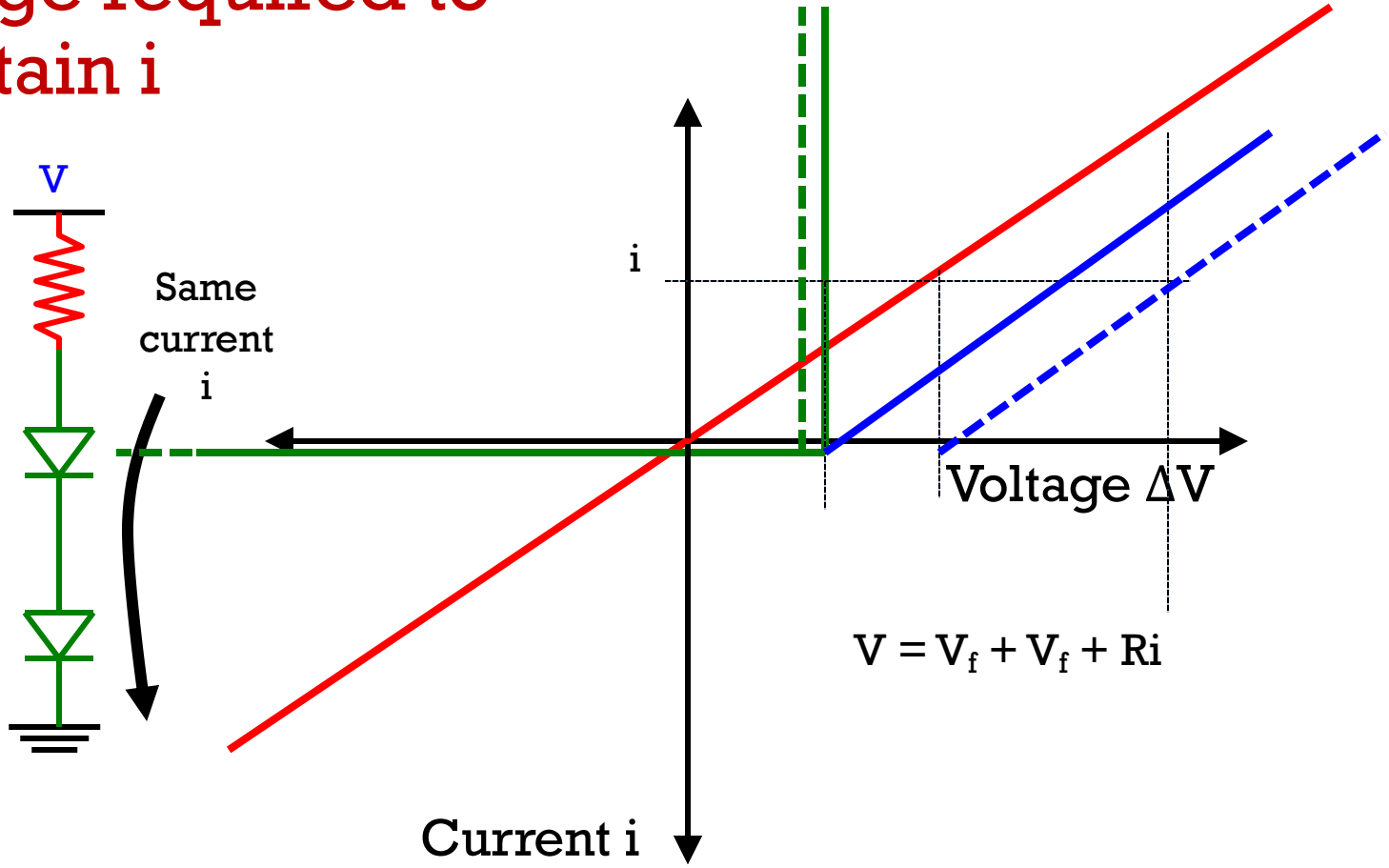
A) Double the light as one LED with res?

B) Same light as one LED (1/2 from each)?

C) More than one, but less than two?



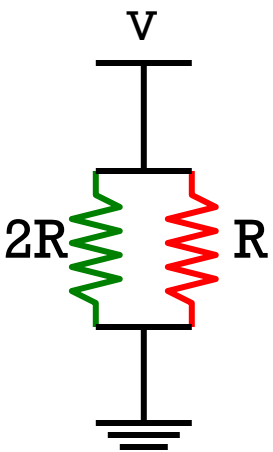
Voltage required to maintain i

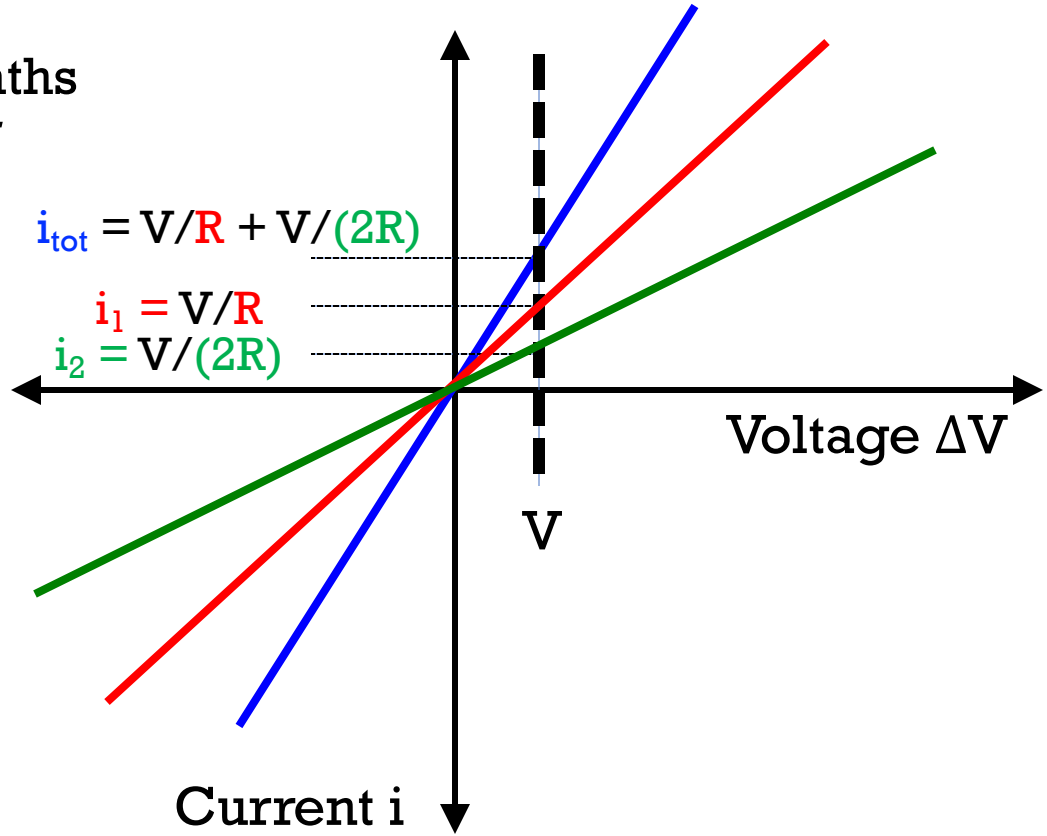


Parallel case?

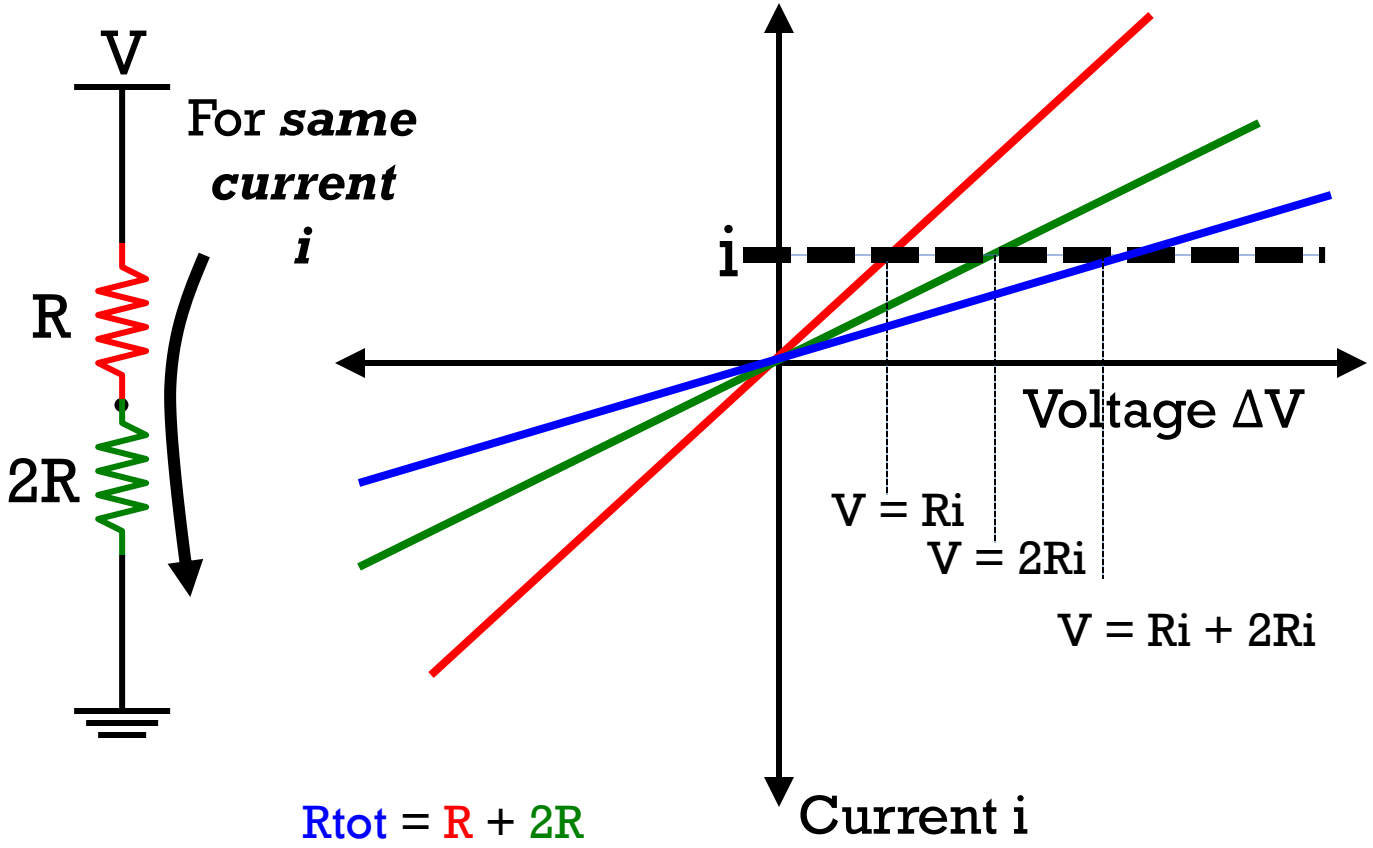
Two current paths
but ***same V***

i_1 , i_2



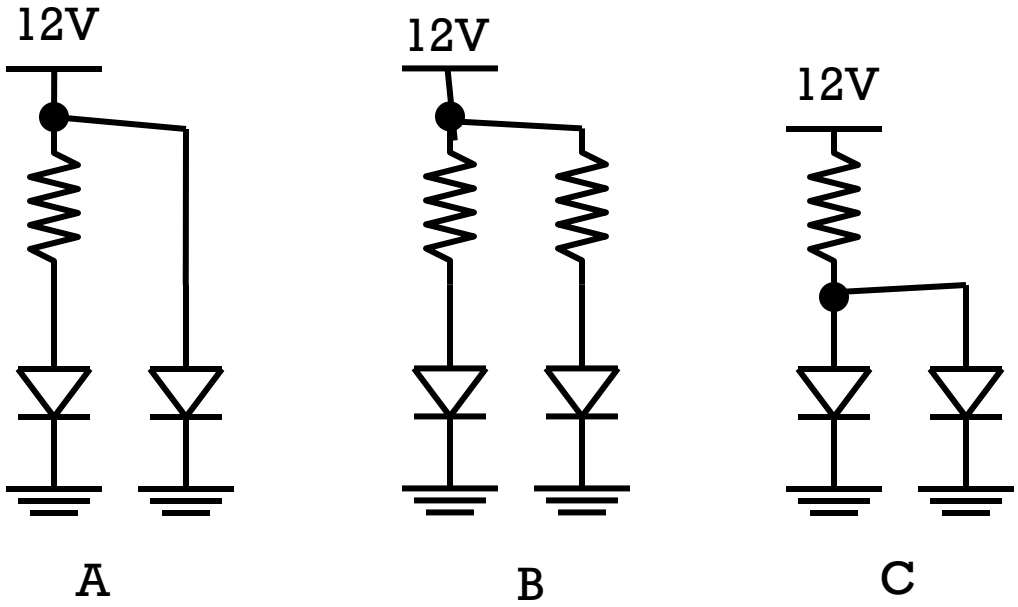


Devices in series



Rtot = R + 2R

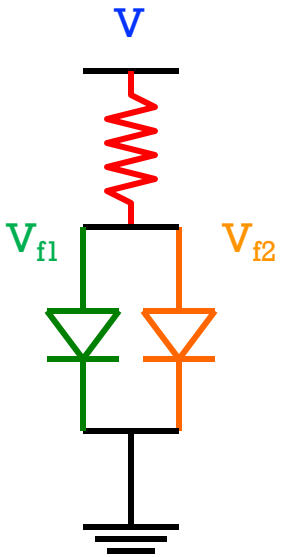
Diodes in Parallel

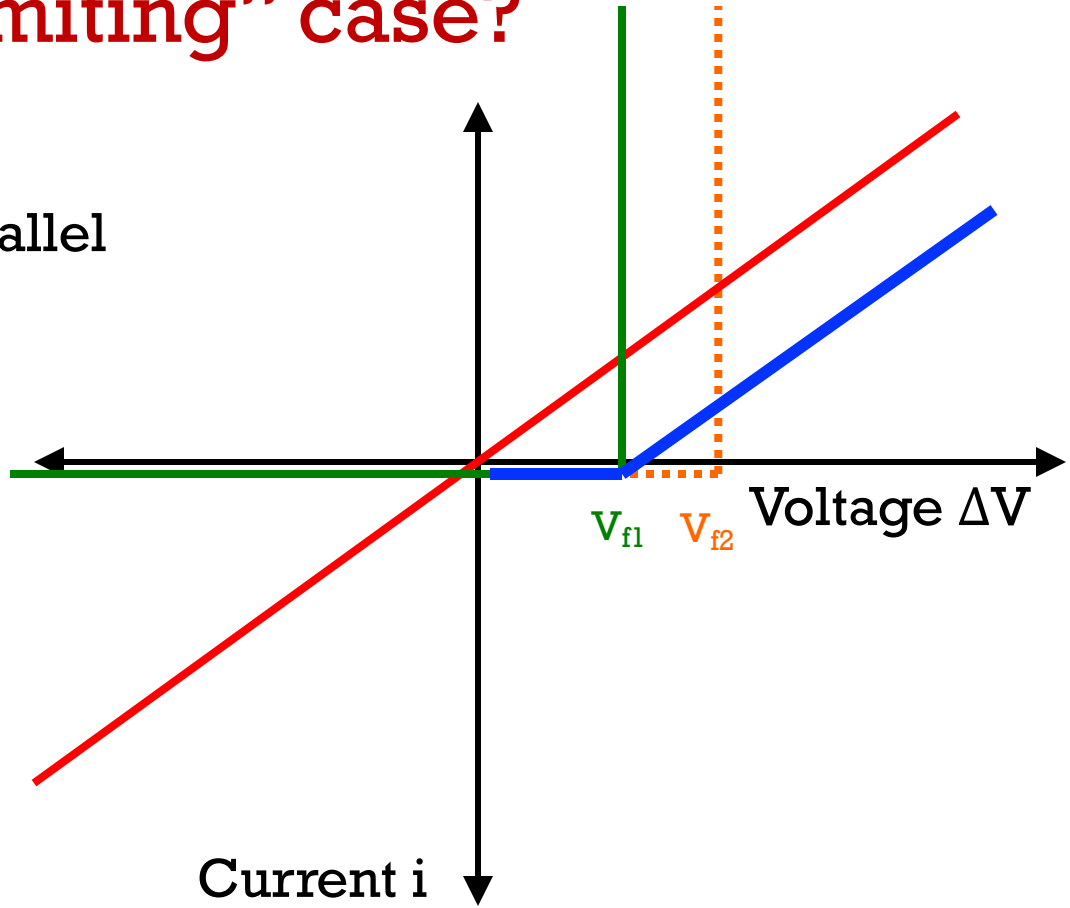


If we have 2 diodes with different Vf which circuit will light both diodes? Circle and hold the correct answer.

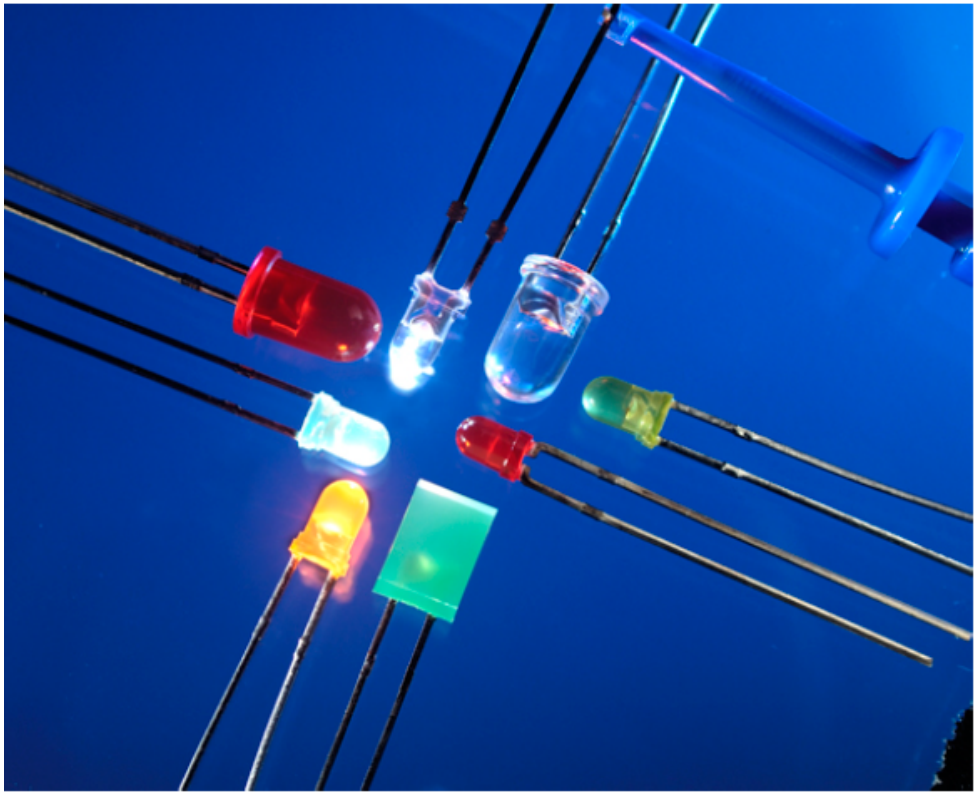
What is “limiting” case?

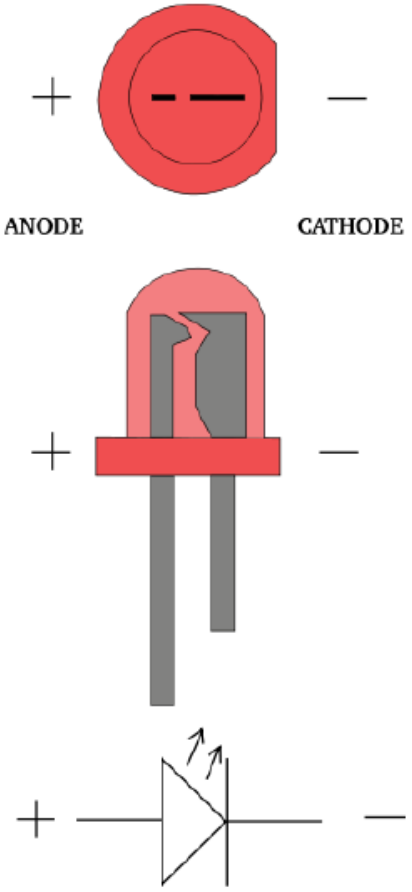
- Circuits in parallel





LED Packages





DataSheet

LN77L

GaAlAs Infrared Light Emitting Diode

For optical control systems

■ Features

- High-power output, high-efficiency: P_O = 18 mW (typ.)
- Fast response and high-speed modulation capability: f_C = 20 MHz (typ.)
- Wide directivity: θ = 20° (typ.)
- Transparent epoxy resin package

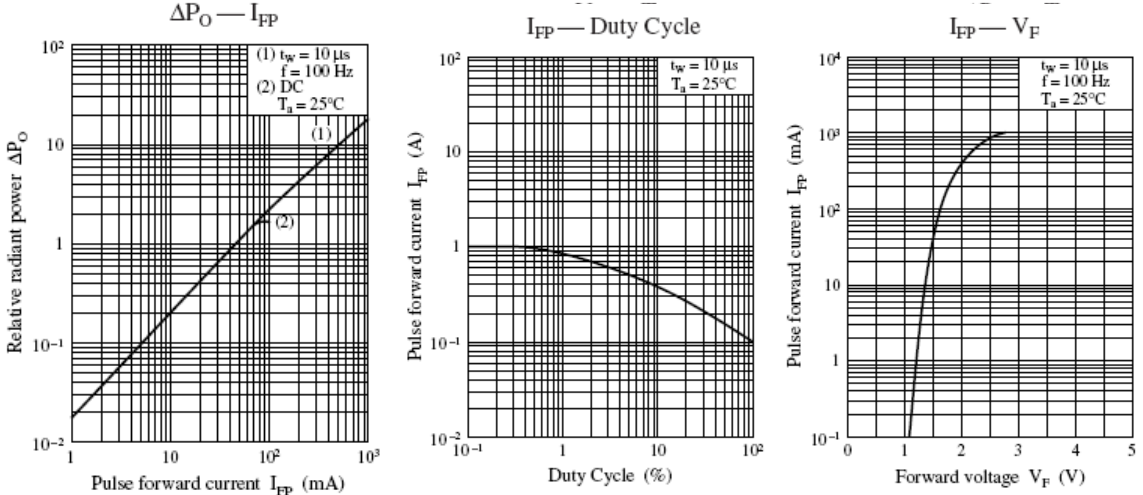
■ Electrical-Optical Characteristics T_a = 25°C ± 3°C

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Forward voltage	V_F	I_F = 100 mA		1.6	1.9	V
Reverse current	I_R	V_R = 3 V			10	μA
Radiant power *	P_O	I_F = 50 mA	10	18		mW
Peak emission wavelength	λ_P	I_F = 50 mA		860		nm
Spectral half band width	$\Delta\lambda$	I_F = 50 mA		40		nm
Half-power angle	θ	The angle when the radiant power is halved		20		°

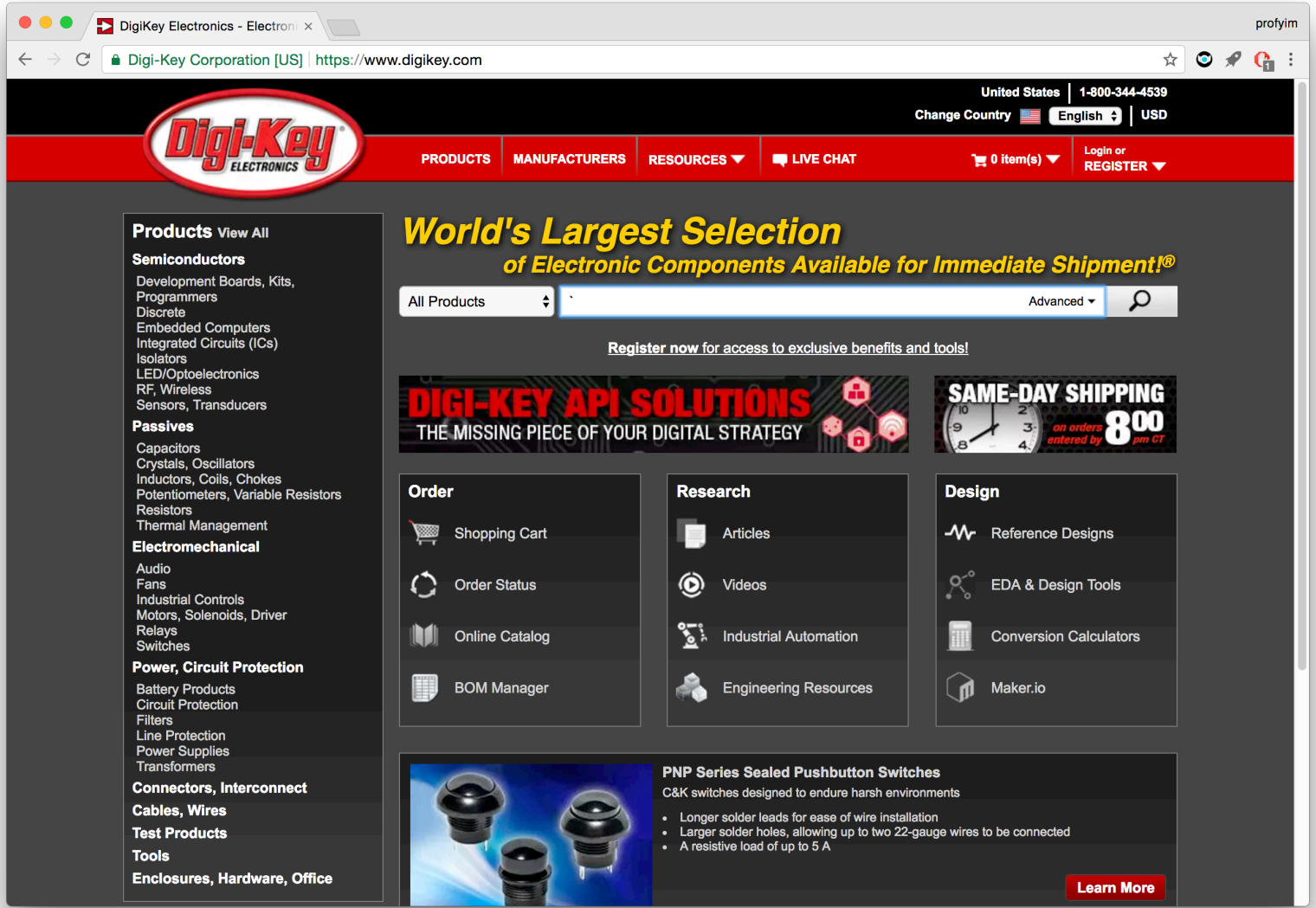
DataSheet

■ Electrical-Optical Characteristics $T_a = 25°C \pm 3°C$

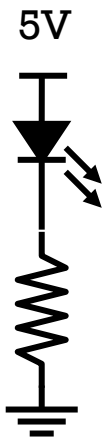
Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Forward voltage	V_F	$I_F = 100$ mA		1.6	1.9	V
Reverse current	I_R	$V_R = 3$ V			10	μA
Radiant power *	P_O	$I_F = 50$ mA	10	18		mW
Peak emission wavelength	λ_P	$I_F = 50$ mA		860		nm
Spectral half band width	$\Delta\lambda$	$I_F = 50$ mA		40		nm
Half-power angle	θ	The angle when the radiant power is halved		20		°



Digikey.com


DigiKey Electronics - Electroni
profyim
Digi-Key Corporation [US] | https://www.digikey.com
United States | 1-800-344-4539
Change Country
English
USD
Digi-Key ELECTRONICS
PRODUCTS
MANUFACTURERS
RESOURCES
LIVE CHAT
0 item(s)
Login or REGISTER
Products View All
Semiconductors
Development Boards, Kits, Programmers
Discrete
Embedded Computers
Integrated Circuits (ICs)
Isolators
LED/Optoelectronics
RF, Wireless
Sensors, Transducers
Passives
Capacitors
Crystals, Oscillators
Inductors, Coils, Chokes
Potentiometers, Variable Resistors
Resistors
Thermal Management
Electromechanical
Audio
Fans
Industrial Controls
Motors, Solenoids, Driver
Relays
Switches
Power, Circuit Protection
Battery Products
Circuit Protection
Filters
Line Protection
Power Supplies
Transformers
Connectors, Interconnect
Cables, Wires
Test Products
Tools
Enclosures, Hardware, Office
World's Largest Selection
of Electronic Components Available for Immediate Shipment!®
All Products
Advanced
Register now for access to exclusive benefits and tools!
DIGI-KEY API SOLUTIONS
THE MISSING PIECE OF YOUR DIGITAL STRATEGY
SAME-DAY SHIPPING
on orders entered by 8:00 pm CT
Order
Shopping Cart
Order Status
Online Catalog
BOM Manager
Research
Articles
Videos
Industrial Automation
Engineering Resources
Design
Reference Designs
EDA & Design Tools
Conversion Calculators
Maker.io
PNP Series Sealed Pushbutton Switches
C&K switches designed to endure harsh environments
Longer solder leads for ease of wire installation
Larger solder holes, allowing up to two 22-gauge wires to be connected
A resistive load of up to 5 A
Learn More

Q7: LED resistor problem



- You have an LED part number WP7113PBC. You want it is as bright as safely possible from a 5V supply. What resistor will work best? What specifications are important from the spec sheet?
- Hint: the more current through the diode, the brighter it will be, but if you have more sustained current than the tested forward current (I_F), the diode may burn out. Don't forget the diode drop. Always be conservative.

Spec sheet:
http://www.us.kingbright.com/images/catalog/SPEC/WP7113PBC-A.pdf

03

Binary, Hex

Binary numbers (base 2)

00=0

01=1

10=2

11=3

In this example:
2-bit number
2 digits, each binary
total of 4 possible.

MSB LSB

0000=0

0001=1

0010=2

0011=3

0100=4

0101=5

0110=6

0111=7

1000=8

... up to 15

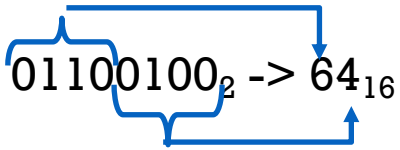
MSB = Most significant bit
LSB = Least significant bit

In this example:
4-bit number
4 digits, each binary

Hexadecimal numbers (base 16)

- "Digits" represent 0-15. Use 0-9, A-F
- Compact way to represent binary numbers
- Represents exactly 4 bits:
- Chunk 8 bit "bytes" into two 4 bit hex "nibbles"

100_{10} -> 01100100_2 -> 64_{16}



'C' notation

asm notation

- Hex notation represented by 0x_ _ or $_ _

  0x64 or $64

Binary - Hexadecimal

- 0000=0
- 0001=1
- 0010=2
- 0011=3
- 0100=4
- 0101=5
- 0110=6
- 0111=7
- 1000=8
- 1001=9
- 1010=A
- 1011=B
- 1100=C
- 1101=D
- 1110=E
- 1111=F

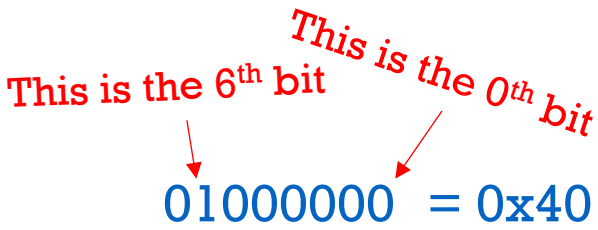
1111|1111 = 0xFF

11000011 = 0xC3

11011110 10101101 = ?

10111110 11101111 = ?

Question 8: Translate binary to hex



04

Intro Lab 1.2 and Teensy

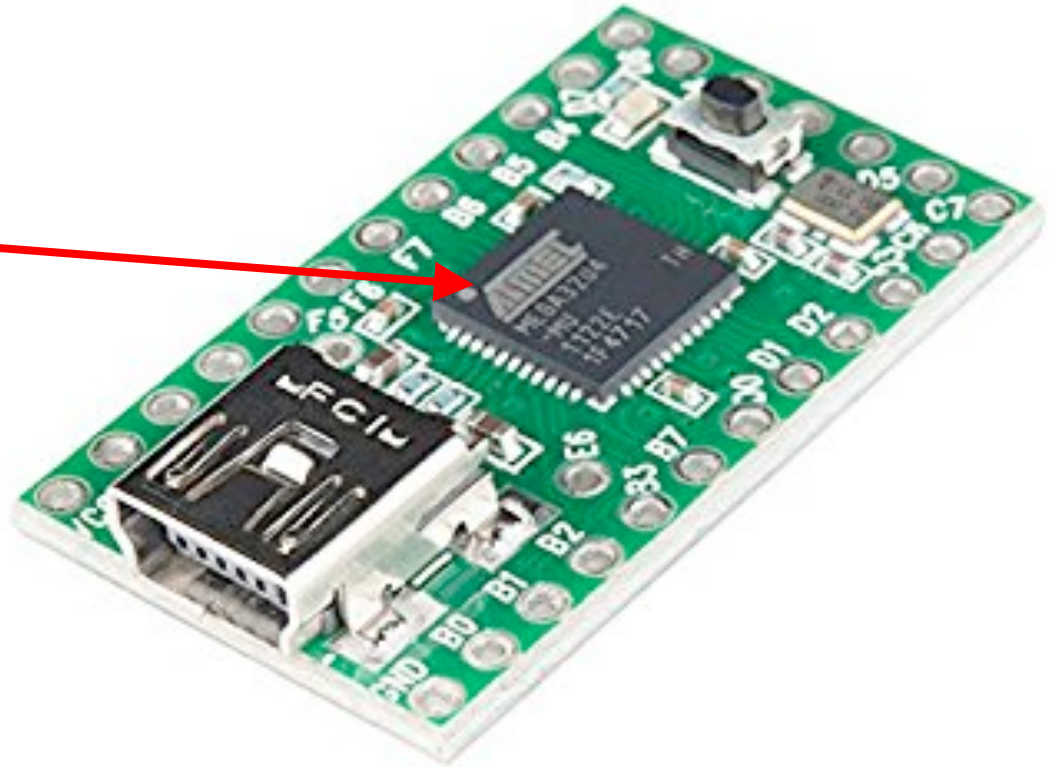
Microcomputers/Microcontrollers

- Are very common and can be found many places like:
  - Cell phones
  - Automobiles
  - Home appliances
- They interface with sensors and/or actuators
- They need to be programmed to function

The Teensy 2.0

(ATmega 32 U4)

From PJRC.com



ATmega32 U4 8-bit Microcontroller

- Can operate @ 5 V
- 32K of program space, 2.5K RAM
- 16 MHz clock speed,
- Peripherals:
  - USB 2.0 interface
  - USART / SPI
  - 16-Bit Timer
  - 12-channel 10-bit ADC
  - 26 GPIO (24 available on Teensy)

Easy development
Note USB Mini

Communication with other devices

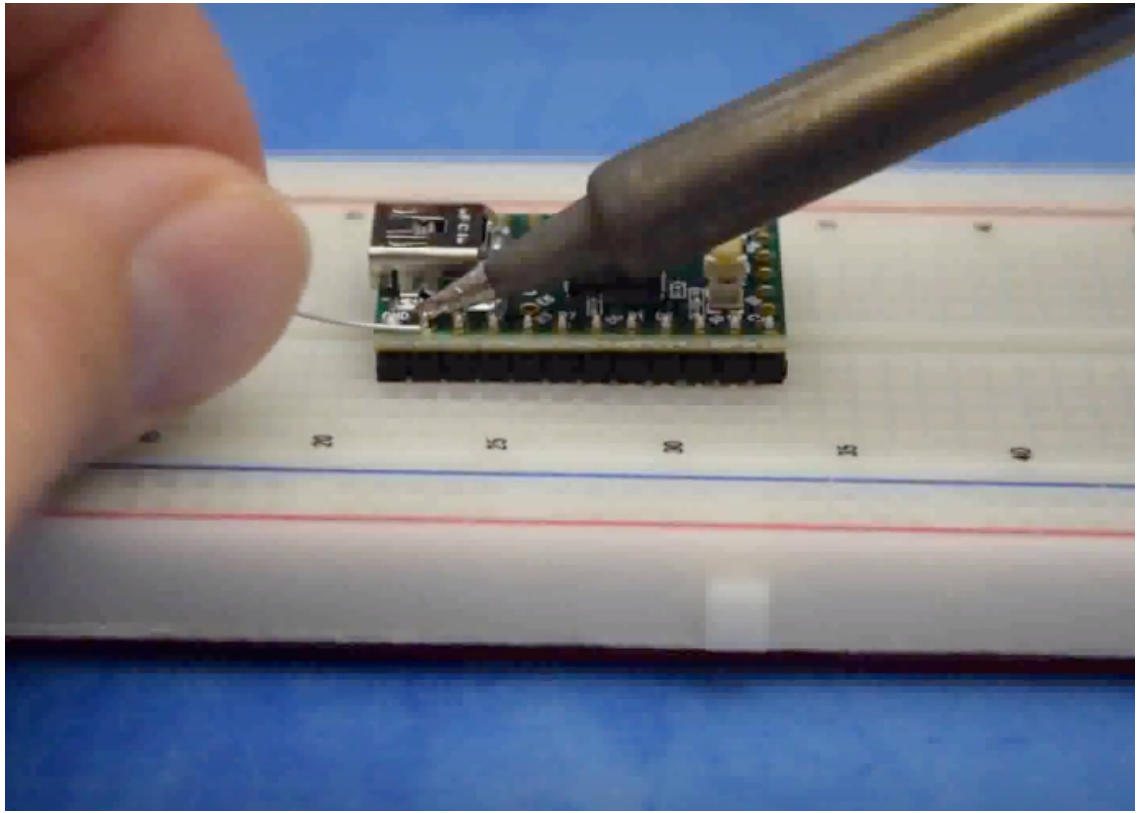
Timer + PWM

Analog input

Digital input/output

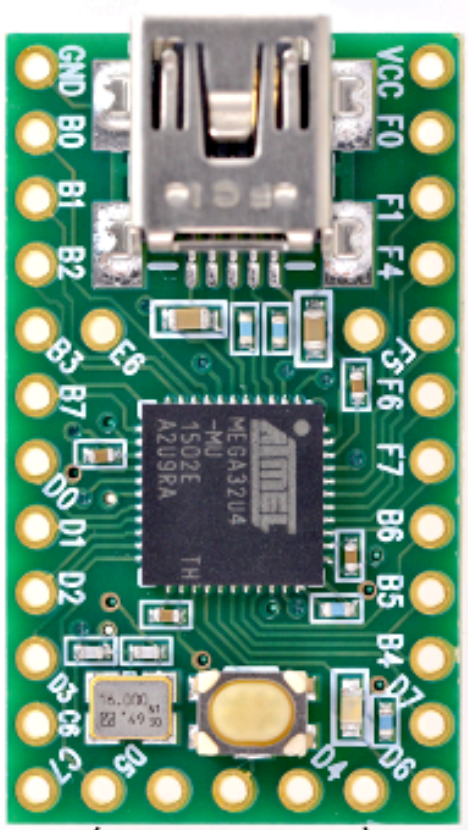
Soldering the Teensy 2.0

3 min video
on good soldering
tips, how to solder
your teensy



https://www.youtube.com/watch?v=XYIqK4uzNYQ&t=78s

Teensy 2.0 Pinout



GND
(interior) SS PB0
PE6 SCLK PB1
AIN0 MOSI PB2
INT6 MISO PB3
RTS OC1C OC0A PB7
OC0B SCL INT0 PD0
SDA INT1 PD1
RXD1 INT2 PD2
TXD1 INT3 PD3
$\overline{\text{OC4A}}$ OC3A PC6
OC4A ICP3 PC7
CTS XCK1 PD5

Vcc
PF0 ADC0 (interior) AREF
PF1 ADC1
PF4 ADC4
PF5 ADC5
PF6 ADC6
PF7 ADC7
PB6 ADC13 OC1B OC4B
PB5 ADC12 OC1A $\overline{\text{OC4B}}$
PB4 ADC11
PD7 ADC10 T0 OC4D
PD6 ADC9 T1 $\overline{\text{OC4D}}$
PD4 ADC8 ICP1

Vcc
GND
RST

(LED on PD6)

Example Program

Starting comment, explain what program is

```
/* Blink.c - Blinks internal LED  */
#include <avr/io.h>

int main(void)
{
    DDRD = 0x40;
        /* insert your hardware initialization here */
    for(;;){
      int i;
      PORTD ^= 0x40;
      for (i=0;i<30000; i++) ;
    }
    return 0;          /* never reached */
}
```

Includes another file into program

Initial setup

Puts a 1 in the 6th bit of DDRD register

Loop forever

XORs the value in the 6th bit of PORTD register
Causing pin D6 to alternate 5V or 0V

Runs ";" no-op, 30,000 times

Binary - Hexadecimal

- 0000=0
- 0001=1
- 0010=2
- 0011=3
- 0100=4
- 0101=5
- 0110=6
- 0111=7
- 1000=8
- 1001=9
- 1010=A
- 1011=B
- 1100=C
- 1101=D
- 1110=E
- 1111=F

1111|1111 = 0xFF

11000011 = 0xC3

This is the 6th bit

This is the 0th bit

01000000 = 0x40

```
PORTD = 0x40; // set 1 to 6th bit, 0 all others
PORTD = 0x00; // clear all bits to 0.
```

Example Program from Lab 1.2 (Blinky.c)

```
#include "teensy_general.h"

int main(void)
{
    teensy_clockdivide(0); //set the clock speed
    teensy_led(ON);        // turn on the on board LED
    teensy_wait(1000);     // wait 1000 ms when at 16 MHz

    for(;;){     /* insert your main loop code here */
        teensy_led(TOGGLE);// switch the led state
        teensy_wait(1000);
    }
    return 0;    /* never reached */
}
```

Includes another file into program

Initial setup

Loop forever

Calls no-ops, but calibrated to take 1ms per input value.

"teensy_general.h" Header File

```
[…]
#define teensy_led(val)     set(DDRD,6);\
        if(val==OFF){set(PORTD,6);}\
        else if(val==ON){clear(PORTD,6);}\
        else if(val==TOGGLE){toggle(PORTD,6);}

[…]
#define OFF         0
#define ON          1
#define TOGGLE      2

[…]
#define set(reg,bit)          reg |= (1<<(bit))
#define clear(reg,bit)        reg &= ~(1<<(bit))
#define toggle(reg,bit)       reg ^= (1<<(bit))
```

Allows macro to continue past the end

#define Macros - some examples

```
#define OFF       0
#define ON        1
#define ever      (;;)
#define runXtimes(x)  for (int i=0; i<(x); i++)
```

Subtle Bug in `teensy_LED`:

Normal use:

```
teensy_LED(ON); -> teensy_LED(1);
teensy_LED(OFF);-> teensy_LED(0);
```

Breaks under this case:

```
int condition1 = 1, condition2 = 0;
teensy_LED(condition1 || condition2);
```

Note, this would not happen if `teensy_LED` was a subroutine.

#define Macros - recommendations

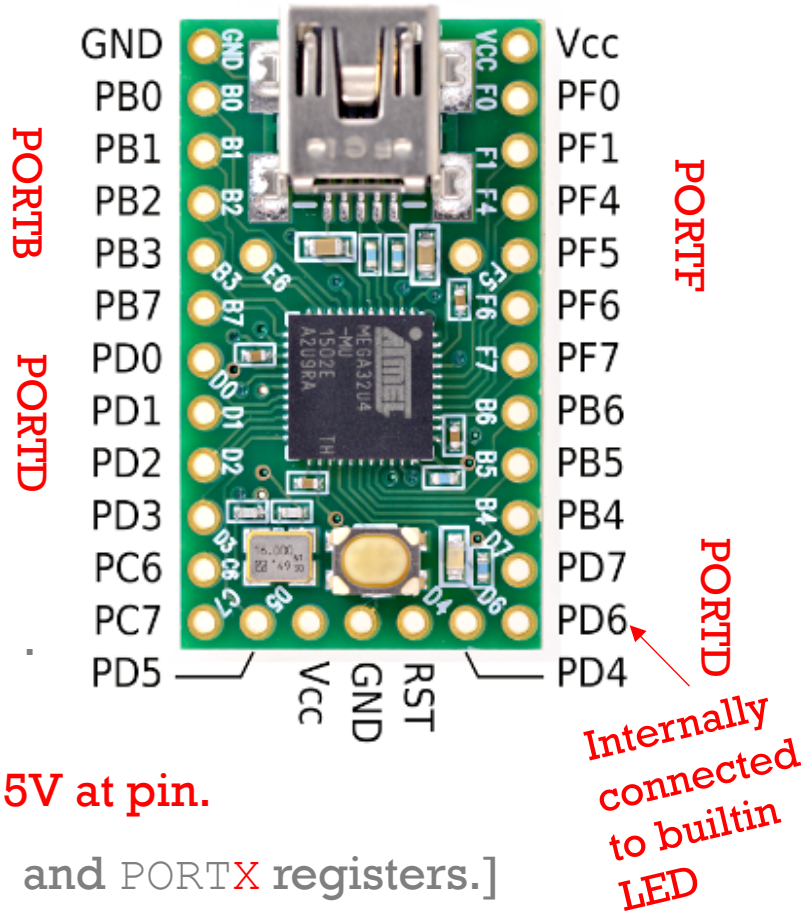
- Use macros where constants are involved (no runtime costs)

```
#define set(reg,bit)          reg |= (1<<(bit))
#define clear(reg,bit)     reg &= ~(1<<(bit))
#define toggle(reg,bit)       reg ^= (1<<(bit))
```

- Macros take more program space than subroutines.
- Macros run slightly faster than subroutines.
- Macros can lead to unexpected bugs.
- Use macros sparingly – e.g. short statements that will make things clearer.
- Macros can sometimes be more confusing than subroutines.

Blinking external LED

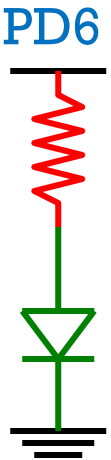
1. Choose any pin P**X#**
   - Example: P**B2**
2. Hookup LED to pin
   - current limited so port is not blown
3. In code, set `DDRX bit#` to be 1.
   - Example `DDRB = 0x02;`
4. Alternate writing 0 and 1 at `PORTX bit#` .
   - Example `PORTB ^= 0x02;`

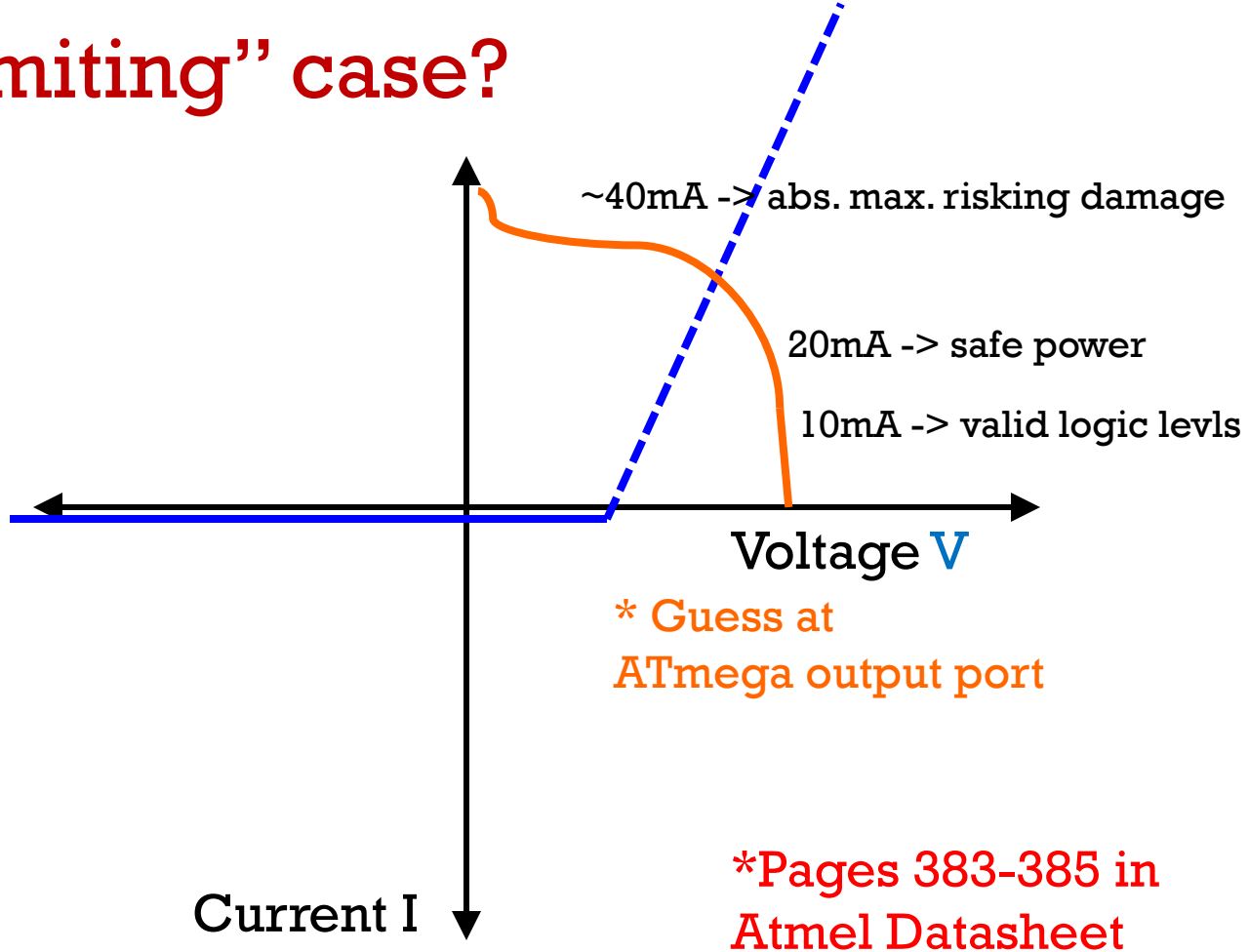


To debug: write 1 to port, use DMM to verify 5V at pin.

[Next lecture we'll talk more about the `DDRX` and `PORTX` registers.]

What is “limiting” case?





* Guess at
ATmega output port

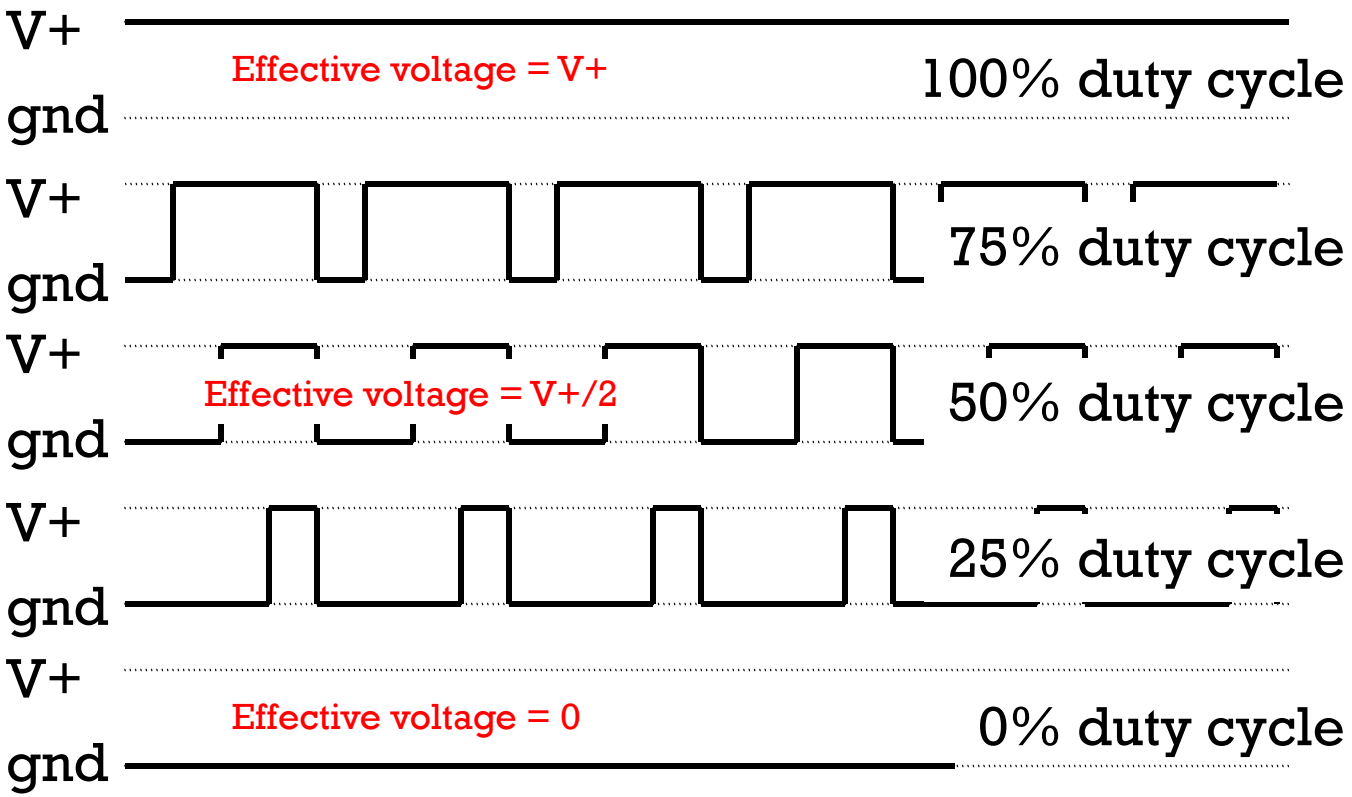
*Pages 383-385 in
Atmel Datasheet

05

PWM
Pulse Width Modulation

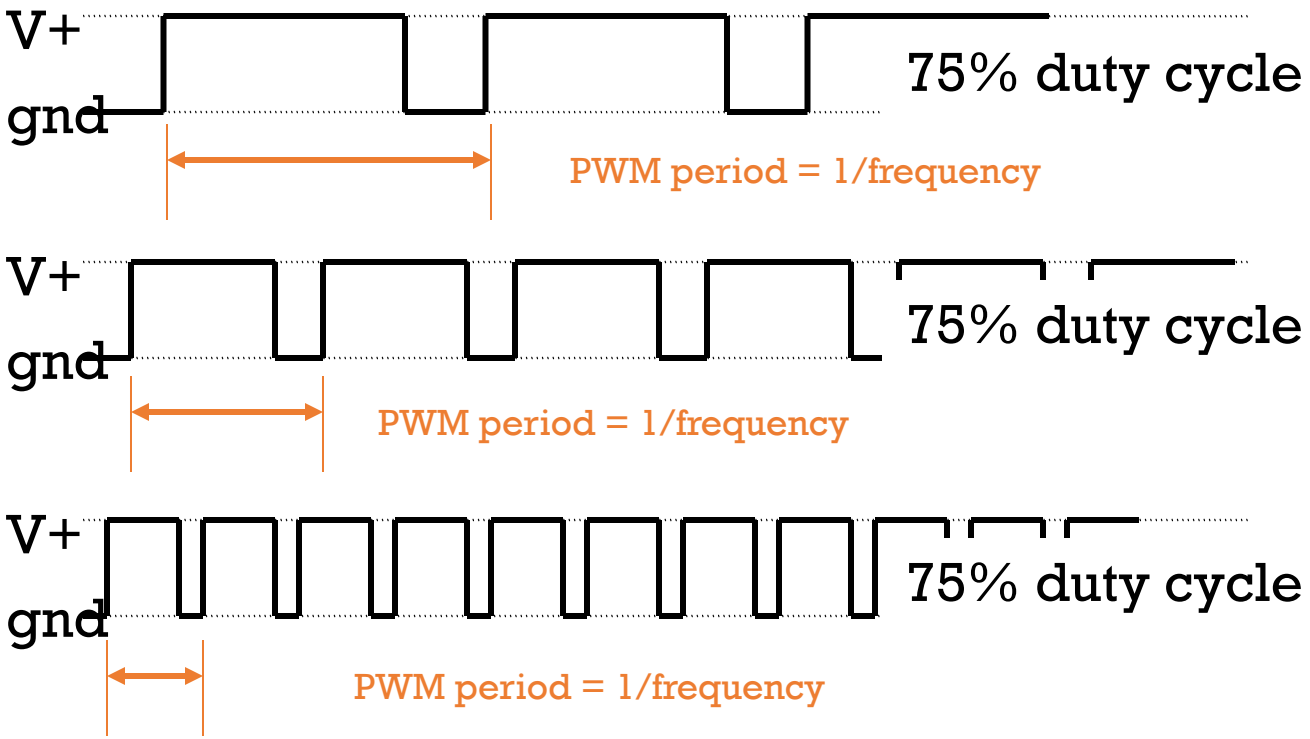
What is pulse width modulation? (PWM)

Duty Cycle



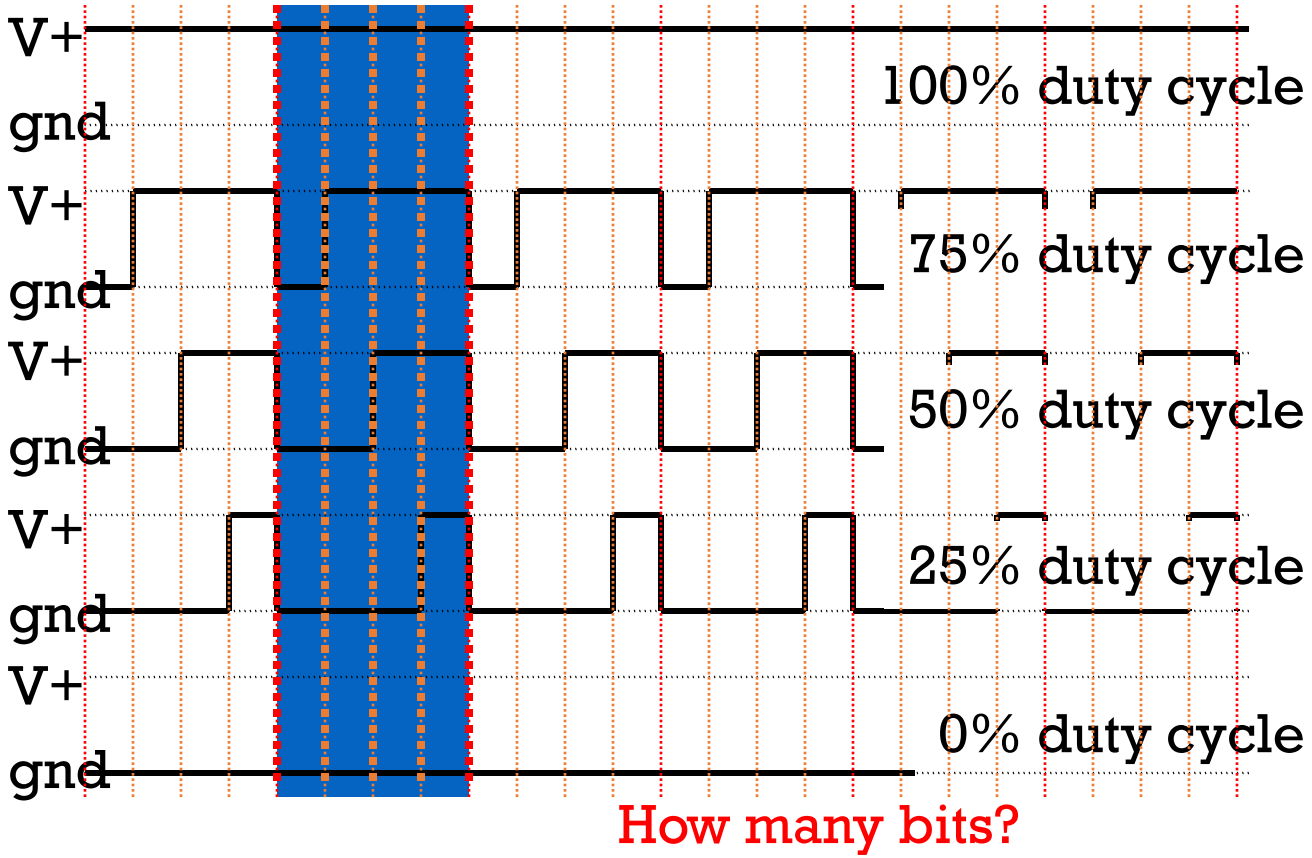
What is pulse width modulation? (PWM)

PWM frequency



What is pulse width modulation? (PWM)

Resolution



How many bits?

Summary for Lab 1.1 and 1.2:

- `set(PORTX, #)` to put 5V at pin P**X#**
- `clear(PORTX, #)` to put 0V at pin P**X#**
- Use function calls to make code easier to build on.
- Don't use "Absolute Maximum" for design purposes.
- Resistors in series with diode limits current
- Our Teensy boards sink/source up to 20mA at a pin

Answer in CHAT

Answer how you feel about each topic below with:

1. I don't understand this topic at all
2. I don't know now, but know what to do to get by
3. I understand some, but expect to get the rest later
4. I understand completely already

A. Passive components

B. LED's

C. Teensy programming