ALGONQUIN COLLEGE

CST8227 INTERFACING

Lecture 2

Agenda:

- Lab2 Summary, LED operation (forward biased versus reverse biased – diagram on whiteboard)

- ARM chip in more detail (data sheet) + Teensy schematics + excel file
- Memory on board Teensy
- Bootloader
- What's that part?

- Lab 3
- Pulse Width Modulation
- Tri-colour LED calculations + datasheets
- Interrupts

- HA#2 now uploaded
- Older edition textbooks uploaded to Brightspace

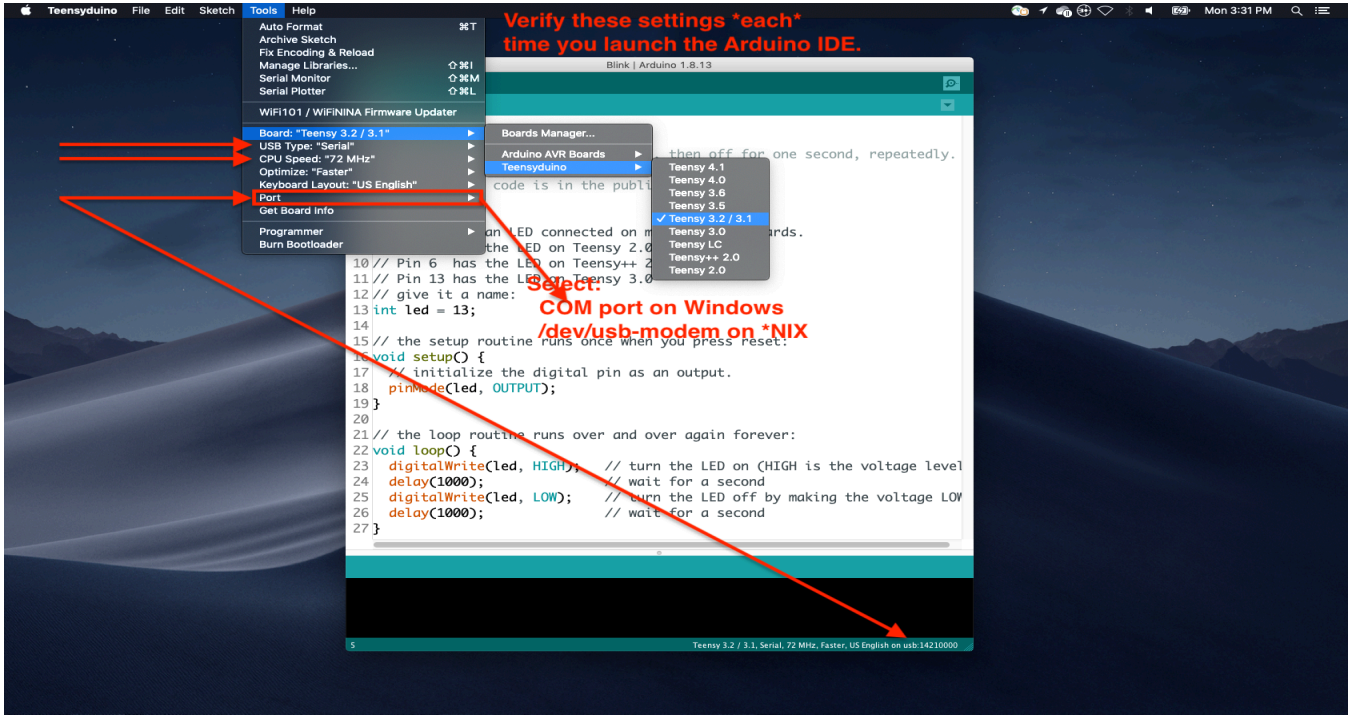
A little bit of trivia….

- What does the acronym ARM represent?
  - Advanced Risc Machine
- How much memory does the Halfkay bootloader occupy?
  - One half of a K (500 bytes)

The IDE

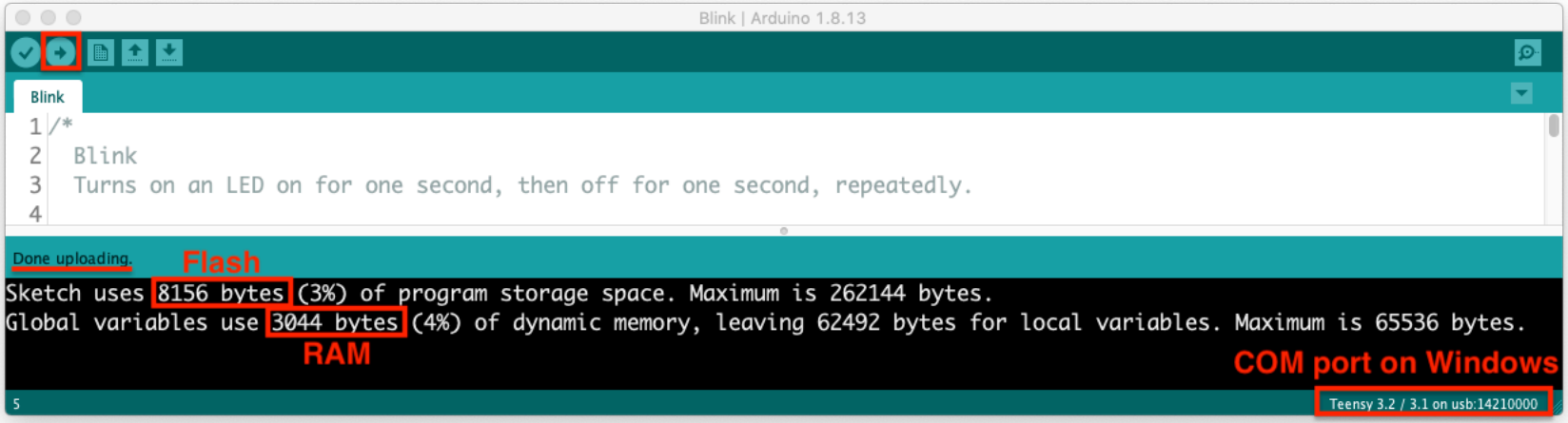
Serial Monitor displays serial data being sent from the Arduino board

Tools Menu

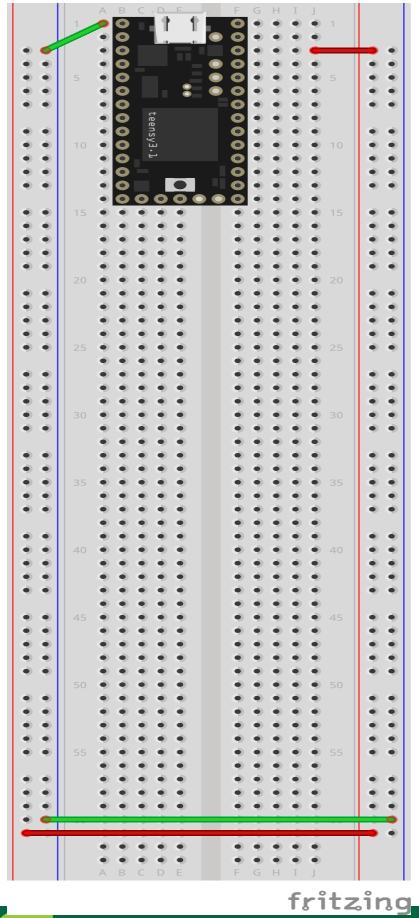


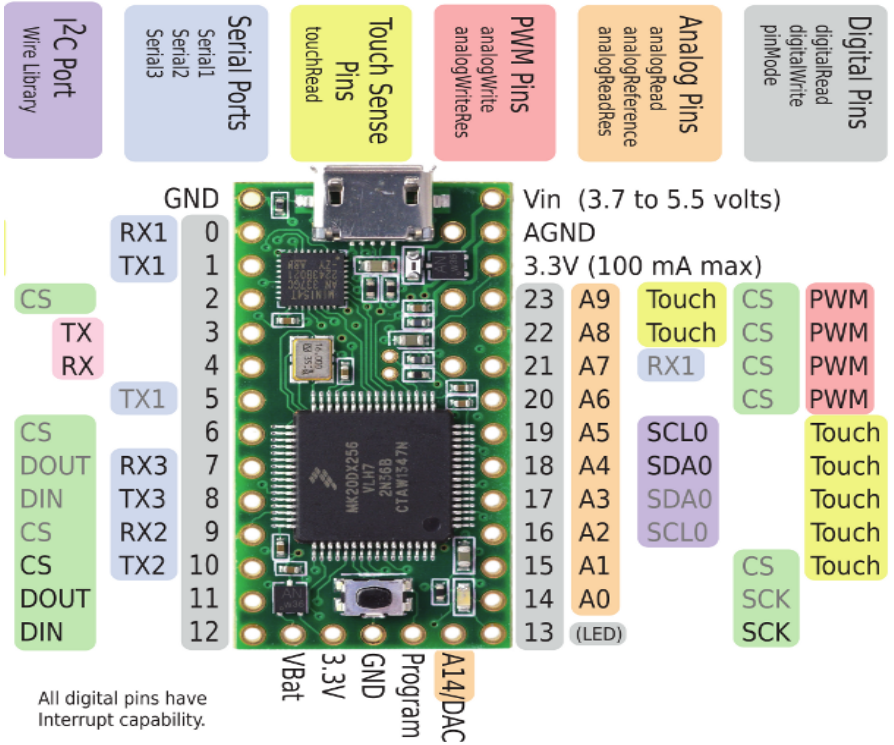
- Board selection sets the file and fuse settings used by the burn bootloader command [and possibly] parameters (e.g. CPU speed and baud rate) used when compiling and uploading sketches;
- USB Type is created with the Teensyduino Add-in; can select the type of device Teensy will simulate when it runs your sketch.

IDE and Memory

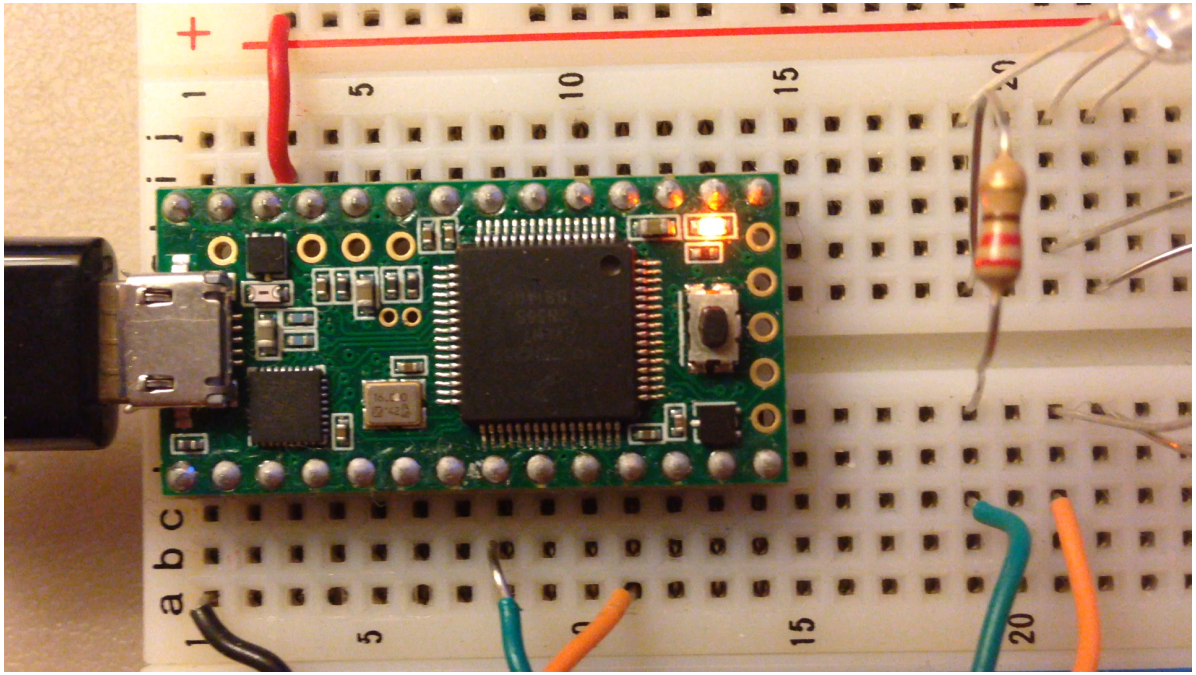


Prelab #2 Schematic



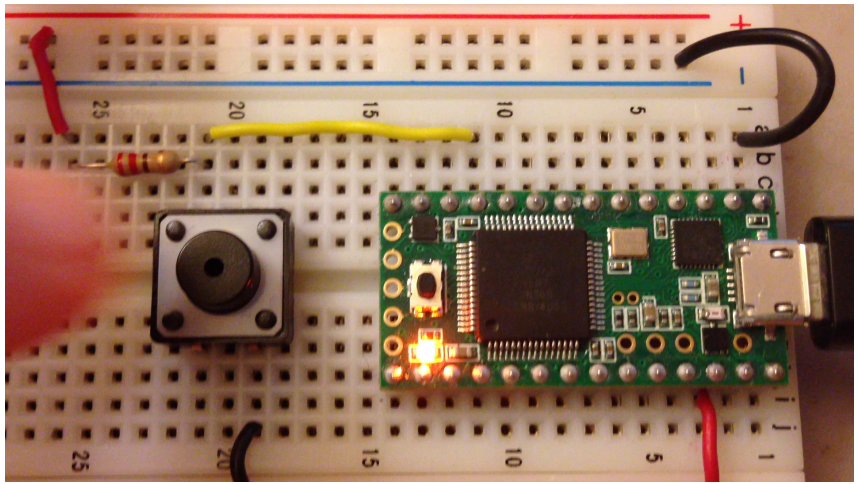


Lab 2 Demo 1

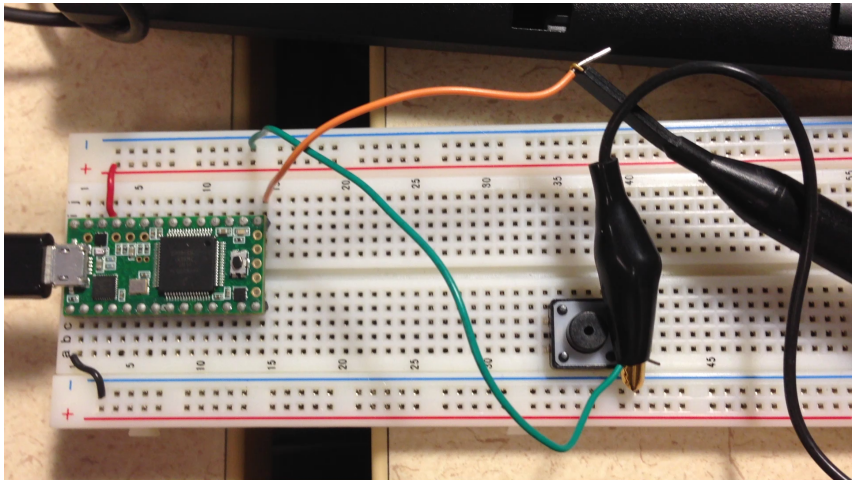


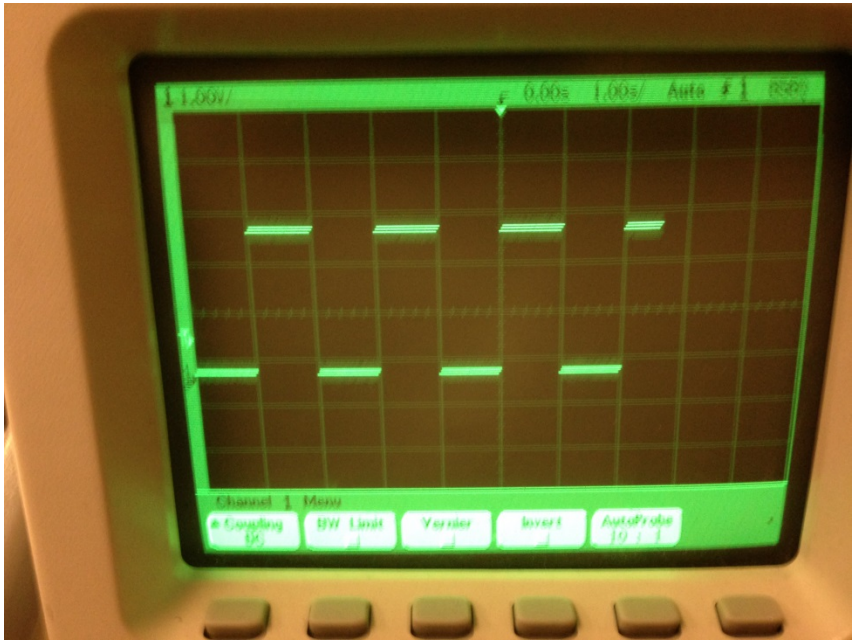
Blink: No Interrupt

```
int pbIn = 9;
int ledOut = 13;       // The output LED pin
boolean state = LOW;      // The input state

void setup()
{
  // Set up the digital Pin 9 to an Input and Pin 13 to an Output
  pinMode(pbIn, INPUT);
  pinMode(ledOut, OUTPUT);
}

void loop()
{
  state = digitalRead(pbIn);      //Read the state of button
  digitalWrite(ledOut, state);    //write the state to LED

  //Simulate a long running process or complex task
  for (int i = 0; i < 1000; i++)
  {
    // do nothing but waste some time
    delay(10);
  }
}
```



Oscilloscope, Blink, delay = 1000 ms





Sketch Basics

- Teensyduino (and Arduino) source files are known as ***sketches*** and are saved with the extension .ino
- The build process ultimately leads to a .hex file (hexadecimal) which is the code that is uploaded to the Teensy board (the ARM chip).

```
//initialize variables, pin modes, etc. calibrations.  Runs once.
void setup(){
    pinMode(led, OUTPUT); // initialize a digital pin as an output or input.
}

// the loop routine runs infinitely
void loop(){
    digitalWrite(led, HIGH); // sets a digital pin to binary 1 or binary 0
    delay(200);              //argument in  miliseconds
    digitalWrite(led, LOW); // sets a digital pin to binary 1 or binary 0
    delay(200);              //argument in  miliseconds
}
```

digitalWrite(), analogWrite(),

digitalWrite(pin, value)

- Writes a HIGH or LOW value to a specified digital pin.

Parameters

- pin: the μC pin number
- ex. int pin = 13;
- value: HIGH or LOW

(also 1 or 0, where 1 = HIGH and 0 = LOW)

Returns

- Nothing

analogWrite(pin, value)

- Writes an analog value (PWM wave) to a pin.
- Can be used to light a LED at varying brightness or drive a motor at various speeds.
- The frequency of the PWM signal pins is approximately 488.28 Hz

Parameters

- pin: the pin to write to.
- value: the duty cycle: between 0 (always off) and 255 (always on).

Returns

- nothing

μC = microcontroller

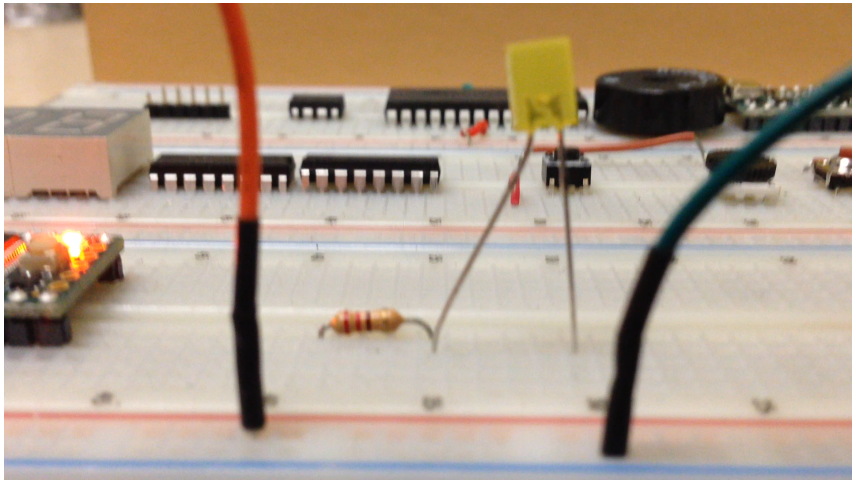
delay(unsigned long)

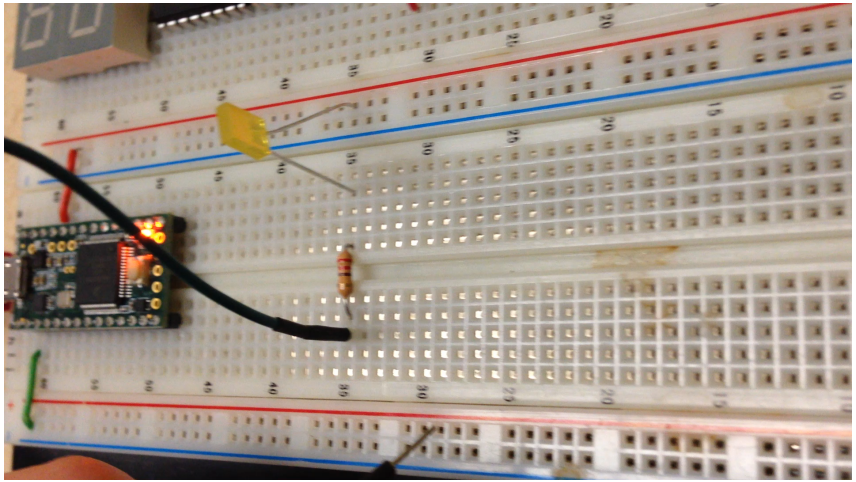
- Pauses the program for the amount of time (in milliseconds) specified as parameter. (There are 1000 milliseconds in one second.)

- **Parameters:** the number of milliseconds to pause (*unsigned long*)
- **Returns**: Nothing

- **delay():** No other reading of sensors, mathematical calculations, or pin manipulation can go on during the delay function, so in effect, it brings most other activity to a halt
- **Avoid the use of delay()** for timing of events longer than 10's of milliseconds unless the Arduino sketch is very simple.
- A better solution is to use millis()

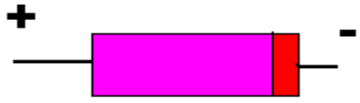
How to Wire an LED

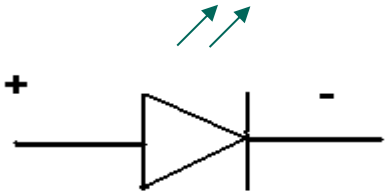




What is a [light emitting] diode?

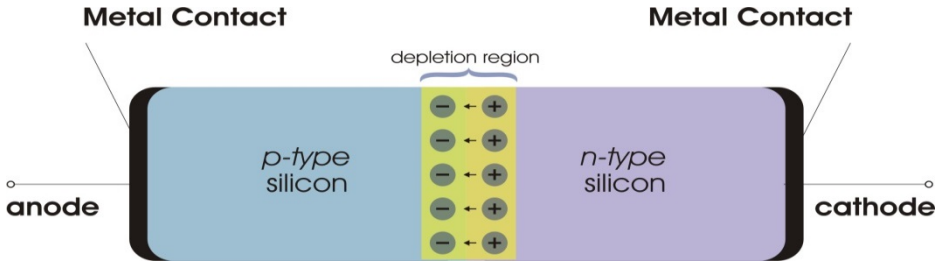
- It is a semiconductor device whose main material is silicon.
- A two terminal electronic device that permits current to flow in ***one direction only***.
- The diode conducts when a sufficiently large voltage is applied to the ***anode*** terminal.
- This value of voltage, known as V_f or the forward voltage drop, is found in device data sheets.
- Typical [small power] forward voltages for LEDs (1.4v, 2.2 v, etc.)
- When operated in this way, current flows and the diode is said to be *forward biased*.
- When a lower voltage appears at the anode, relative to the cathode, the diode will not conduct and is said to be *reverse biased*.



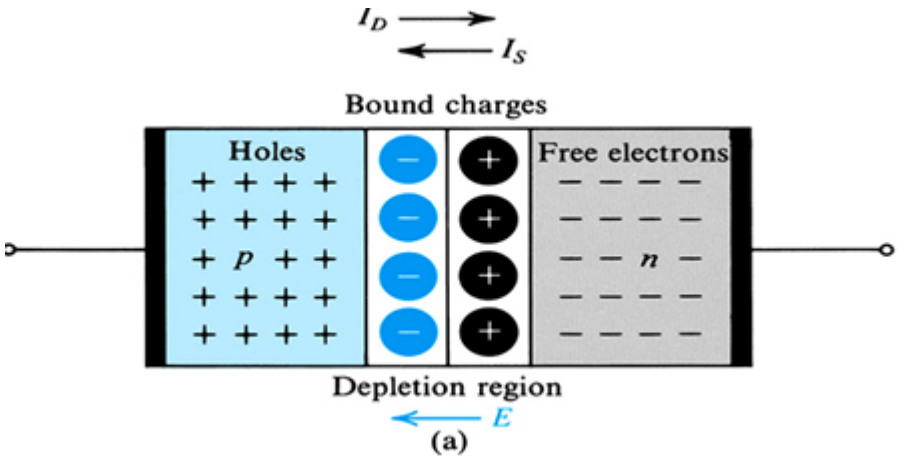


The PN junction

- Diodes are manufactured by joining two types of semiconductor materials together: positive “P “ material, and negative “N” material
- The PN junction, known as the depletion region, is the border area between the positive Si and the negative Si.
- When the PN junction is wide, the resistance of the diode is large and it is considered to be an open switch, or an open circuit.
- When the PN junction is narrow, the resistance of the diode is small and it is considered to be a short switch, or a closed circuit.



Semi-Conductor Theory



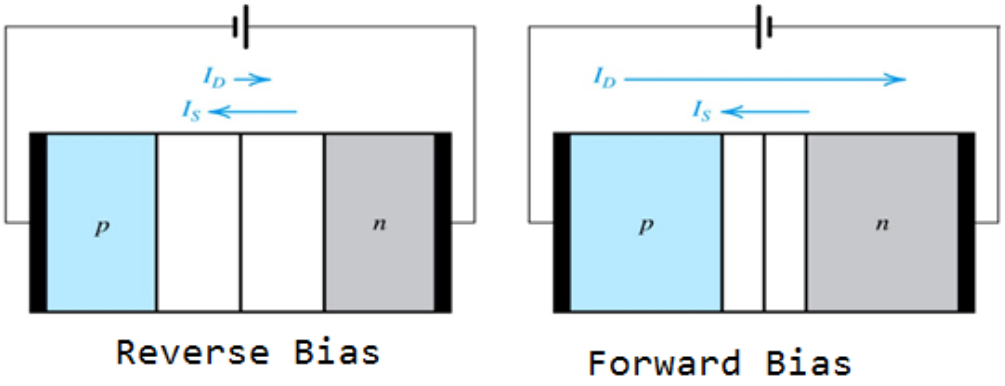
- From a sub-atomic, chemistry perspective, the “p” material has
- an excess of +ve charges and the “n” material has an excess of -ve charges.
- When the two materials are joined together, a depletion region is formed.
- This depletion region voltage potential is ultimately what amounts to the Vf specification found in LED data sheets.

Note: there is no such thing as a +ve charge, only a lack of –ve charge, however it is helpful in the understanding of flow of current to envision the existence of both charges.

Forward Bias versus Reverse Bias

Reverse bias:

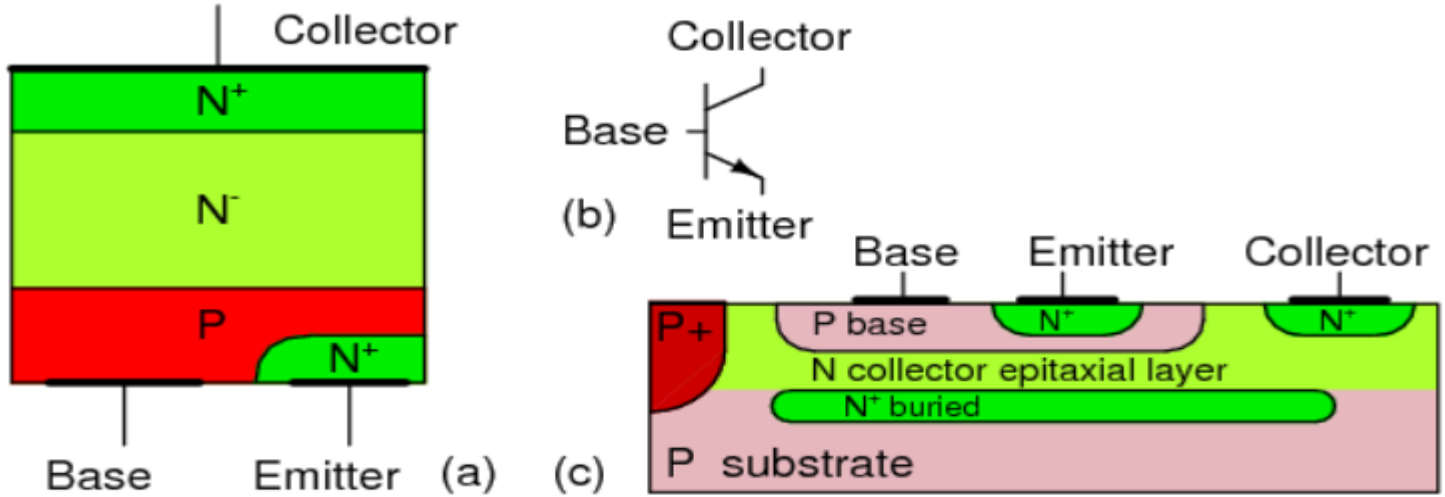
- Occurs when the voltage at the anode is less than or equal to the voltage at the cathode.
- The depletion region is wide.
- Current does not flow.
- Device causes the circuit to behave like an open circuit.



Forward bias:

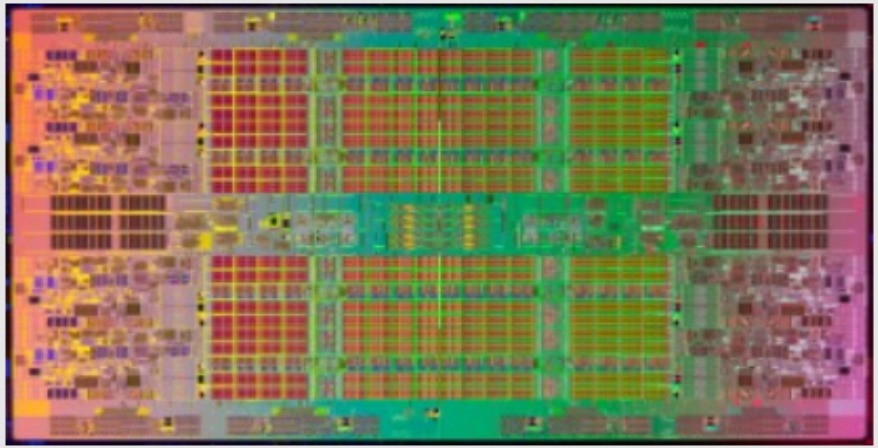
- Occurs when the voltage at the anode is greater than the voltage at the cathode by at least Vf volts.
- The depletion region is narrow.
- Current flows.
- Device behaves like a short circuit – the only voltage drop will be Vf.

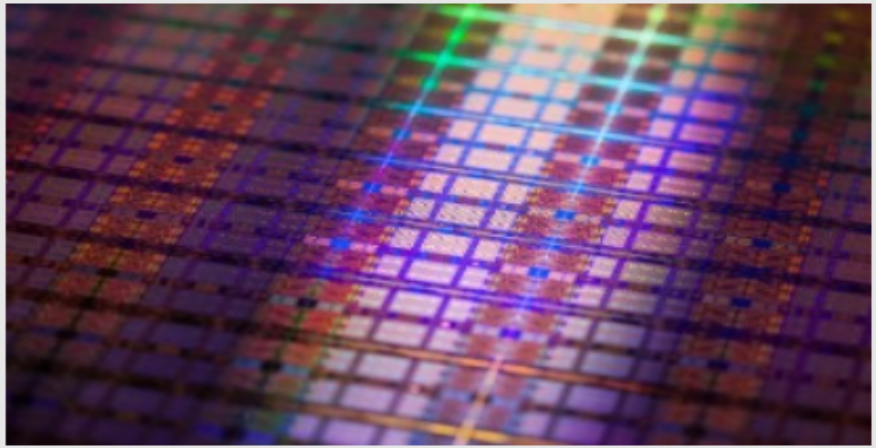
CMOS Transistor



Source: https://electronics.stackexchange.com/questions/26404/what-so-great-about-cmos

Intel Processor

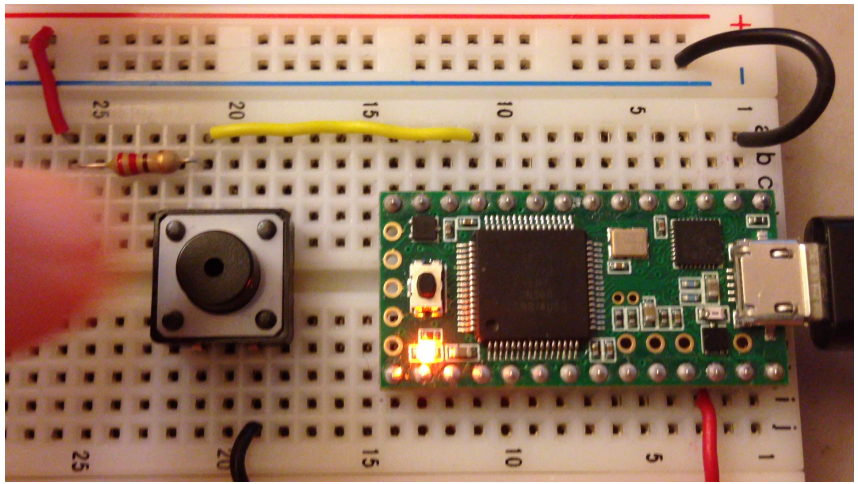




Containing 3.1 billion transistors, the Intel® Itanium® processor 9500 series is Intel's most sophisticated general purpose processors to date. It supports up to twice as many cores (8 instead of 4) than the previous-generation processor, packs up to 54 MB of on-die memory, and enables up to 2 TB of low voltage DIMMs in a four-socket configuration.

Pushbutton: No Interrupt

```
int pbIn = 9;
int ledOut = 13;       // The output LED pin
boolean state = LOW;      // The input state

void setup()
{
  // Set up the digital Pin 9 to an Input and Pin 13 to an Output
  pinMode(pbIn, INPUT);
  pinMode(ledOut, OUTPUT);
}

void loop()
{
  state = digitalRead(pbIn);      //Read the state of button
  digitalWrite(ledOut, state);    //write the state to LED

  //Simulate a long running process or complex task
  for (int i = 0; i < 1000; i++)
  {
    // do nothing but waste some time
    delay(10);
  }
}
```



digitalRead(),

Prototype: digitalRead(int)

- Reads the value from a specified digital pin, either HIGH or LOW.

Syntax

- digitalRead(pin)

Parameters

- pin: the number of the digital pin you want to read (*int*)

Returns

- HIGH or LOW

- Important uses include the polling of pins for a change in state (i.e. a user pushes a button).

Lab #2 Q&A from the PJRC forum

Q. Everytime I upload a new program should I first press the button on the board ? Or can i upload without pushing the button each time?

A. Usually you can upload without pressing the button, but that depends on your computer being able to find and communicate with the board, to request a reboot.

Q. … When i press the [reset] button and Halfkay loads, what is teensy now emulating on the USB ?

A. When you press the [reset] button, Teensy disconnects from the USB, the same as if you'd unplugged the cable. When the bootloader begins, it appears to your PC as a custom HID device. Teensy Loader talks to that to upload your code. At the end of the upload when your board reboots again, it again disconnects and then reconnects while running your program. The setting in Tools > USB Type determines what type of device it will reconnect as...

Lab #2 Q&A from the PJRC forum

Q. Does the HalfKay bootloader reside on the ARM chip ?

A. It's stored in the Mini54 chip on the board.

https://forum.pjrc.com/threads/846-Newbie-Question-about-the-quot-Teensy-Loader-quot?highlight=halfkay:

Note that the pushbutton on the Teensy 3 is actually a "program" button, not just "reset", so it starts up the bootloader which is in the secondary ARM chip on the Teensy 3 and waits for the PC to send it a new software load (eg. hex file).

Since we are programming in C…..

- In java, true and false are keywords.
- In C and C++, true and false are also keywords.
- *But* in C and C++, true = 1 and false = 0.
- When programming in the Arduino environment,

  HIGH == true == 1

  and ….

  LOW == false == 0

```
digitalWrite(anode, 1);
digitalWrite(anode, true);
digitalWrite(anode, HIGH);
```

All work!!!

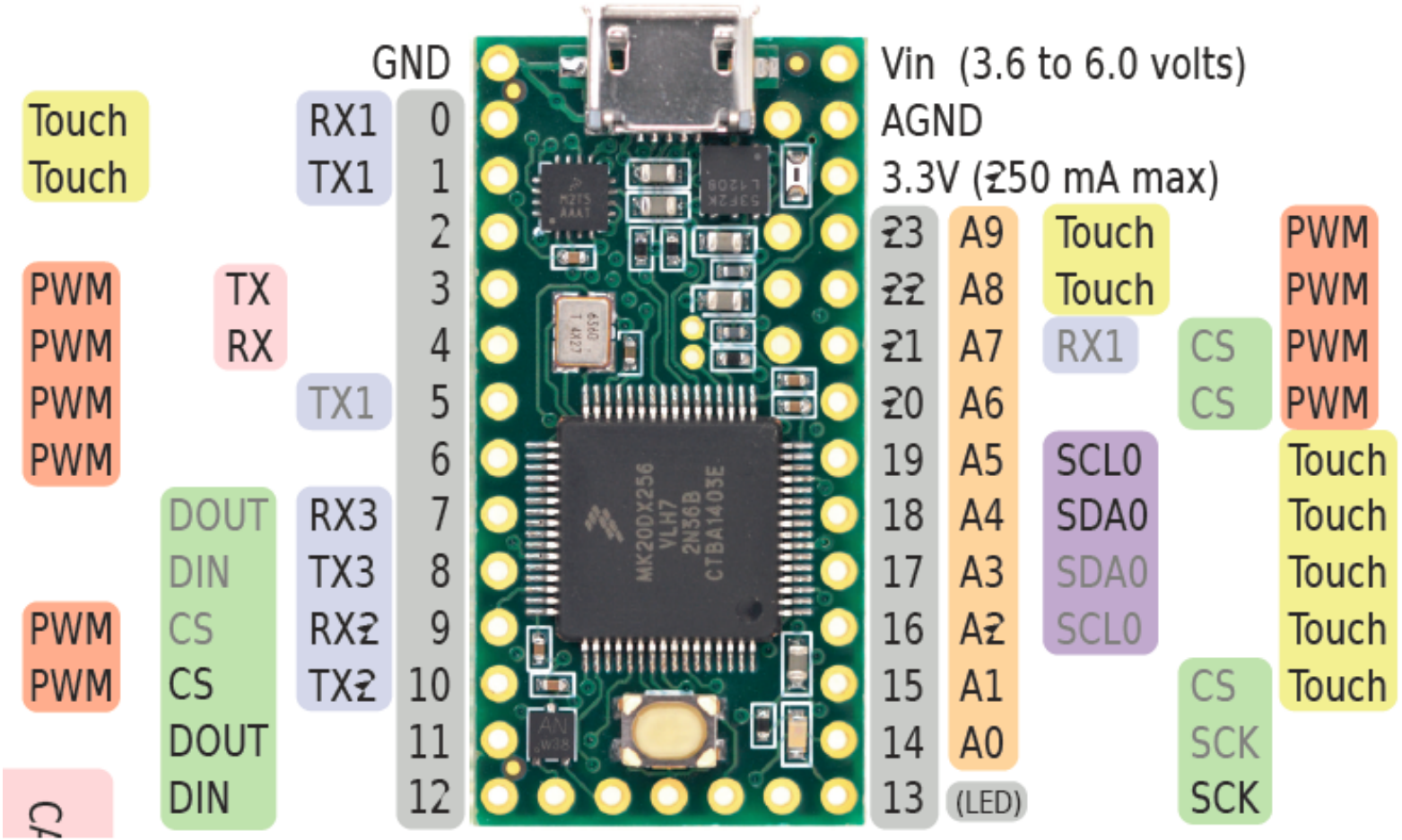
Hmmmm…is there a more convenient way to write code available???

Arduino Development Language

- The complete specification for the Arduino language is available at:http://arduino.cc/en/Reference/HomePage
- Since this is a subset of the C language, it should hopefully all be familiar to you.
- We will spend just a short time reviewing the elements of the language.
- While reviewing the language specification, did you notice that all the reserved words and functions started with a lower case letter (other than 4 constants)?
- These are all part of the C language.
- There are only a few exceptions: Serial, Stream, String (plus Keyboard and Mouse) which are all C++ classes.
- They are unique in Arduino in having class methods eg. Serial.begin(), Serial.read(), Serial.write(), Serial.print(), and Serial.println().
- You should be familiar with the behavior and features of the String class from your experience in Java and C++.
- As a consequence, the Arduino language is not particularly *deep*: there isn't much in the way of class hierarchies to learn.

Embedded Systems and [no significant] operating system, and the infinite loop

- Embedded systems have a finite amount of memory and no significant operating system (i.e. windows, etc..).
- In the case of Teensy, the bootloader, after it finishes it's uploading responsibilities, will also load the address of the first instruction of the sketch into the stack.
- Furthermore, in order to keep the app running (presumably the device has been deployed in the field), an infinite loop is necessary.
- There is underlying primitive code behind the loop() function


GND
Vin (3.6 to 6.0 volts)
AGND
3.3V (250 mA max)
Touch
Touch
RX1
TX1
0
1
2
3
4
5
6
7
8
9
10
11
12
23
22
21
20
19
18
17
16
15
14
13
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
(LED)
Touch
Touch
RX1
SCL0
SDA0
SDA0
SCL0
PWM
PWM
PWM
PWM
PWM
PWM
PWM
PWM
TX
RX
TX1
DOUT
DIN
CS
CS
DOUT
DIN
RX3
TX3
RX2
TX2
CS
CS
SCK
SCK
Touch
Touch
Touch
Touch
Touch
MK20DX256
VLH7
2N36B
CTBA1403E

Semiconductor Sales Leaders, 2016-2017

2017 Rank	2016 Rank	Vendor	2017 Revenue	2017 Market Share (%)	2016 Revenue	2016-2017 Growth (%)
1	2	Samsung Electronics	61,215	14.6	40,104	52.6
2	1	Intel	57,712	13.8	54,091	6.7
3	4	SK Hynix	26,309	6.3	14,700	79.0
4	6	Micron Technology	23,062	5.5	12,950	78.1
5	3	Qualcomm	17,063	4.1	15,415	10.7
6	5	Broadcom	15,490	3.7	13,223	17.1
7	7	Texas Instruments	13,806	3.3	11,901	16.0
8	8	Toshiba	12,813	3.1	9,918	29.2
9	17	Western Digital	9,181	2.2	4,170	120.2
10	9	NXP	8,651	2.1	9,306	-7.0
		Others	**174,418**	**41.6**	**157,736**	**10.6**
		Total Market	**419,720**	**100.0**	**343,514**	**22.2**

Source: Gartner (January 2018)

Source: https://www.gartner.com/newsroom/id/3842666

Identify the Parts:

1. USB Port
2. Processor – 32 bit ARM Cortex-M4
3. Reset Button
4. I/O Pins
5. Onboard LED
6. GND Pin
7. 5 volt input
8. 16 KB Flash ROM Memory
9. Various surface mount resistors and capacitors
10. 16 MHz Oscillator
11. 3.3 volt output
12. Voltage Regulator