Digital Electronics 101

An introductory course on Electronics, C++ and Arduino-like platforms

J. Rodrigues R. Antunes

HackerSchool

What are Electronics?

Fundamental Notions and Laws in Electronics

There are three main notions to be understood in electronics:

- **Current aka I** (SI: Ampere): The ordered flow of electrons, therefore electrical charge per time unit
- **Electrical Tension or Potential Difference aka U** (SI: Volt): The tension applied on said electrons, therefore energy per charge
- **Resistance aka R** (SI: Ohm): The resistance of a medium to the electron flow

Ohm's Law

We can describe the resistance as the tension we have to "apply" to push the electron flow establishing an equality-**Ohm's Law:**

$$R = \frac{U}{I}$$

Or in a funnier way:

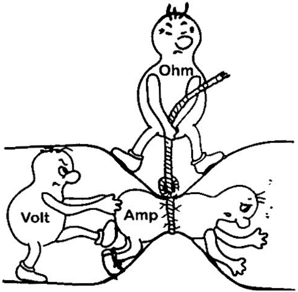


Figure 1: Ohm's Law

Kirchhoff's Voltage Law

The sum of all potential diferences in a loop is zero!

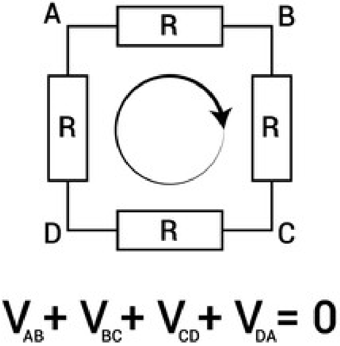


Figure 2: Kirchhoff's Voltage Law

Kirchhoff's Current Law

There can be no residual current in a node!

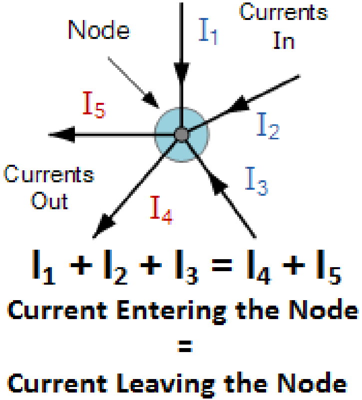


Figure 3: Kirchhoff's Current Law

Additional Notions and Fundamentals

Then we can also add some additional notions

- **Power**(SI: Watt): Rate of transference of electrical energy through a circuit, using the definition of Electrical Tension and current:

$$P = U \times I$$

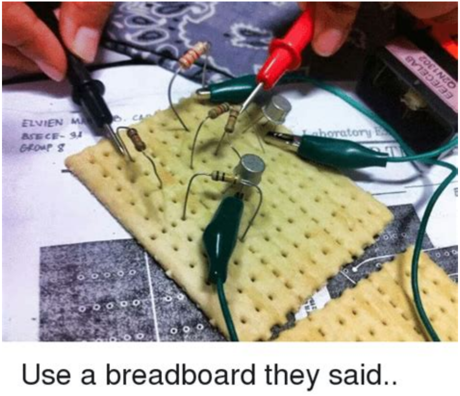


Figure 4: A BREADboard!

Breadboards and PCBs II

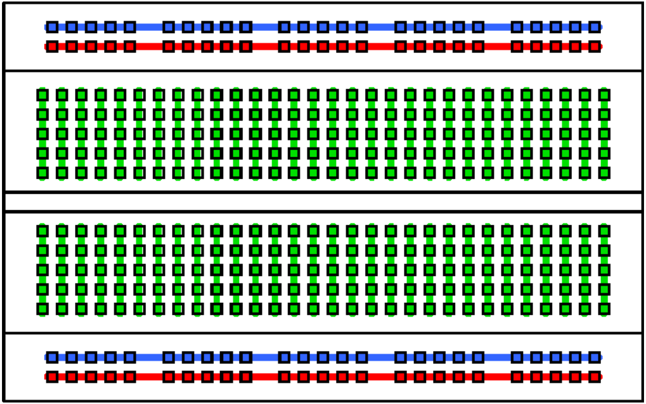


Figure 5: An actual breadboard

Breadboards and PCBs III

After testing our circuit in a breadboard, we might have a very complex and not portable weave of wires and components. . .

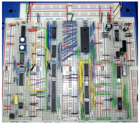


Figure 6: A very confusing weave of wires

We can then use some software tools (like KiCAD) to help us create a PCB schematic.After that we can send it to a manufacturer or do it ourselves!

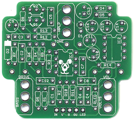


Figure 7: A printable circuit board

Power sources

The most fundamental devices are power sources. They supply (active component) a given tension to our circuit through the conversion of an x type of energy (chemical, mechanical, etc.) into electrical energy.

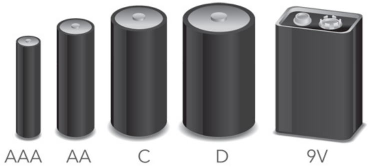


Figure 8: Batteries - A conversion device of chemical into electrical energy

Resistors

Another fundamental component. Depending on the resistor type they introduce a fixed or variable resistance (R) into our circuits. They are unpolarized.

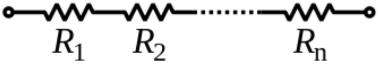


Figure 9: Series association: $R_{\text{eq}} = R_1 + R_2 + \cdots + R_n$

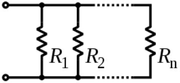
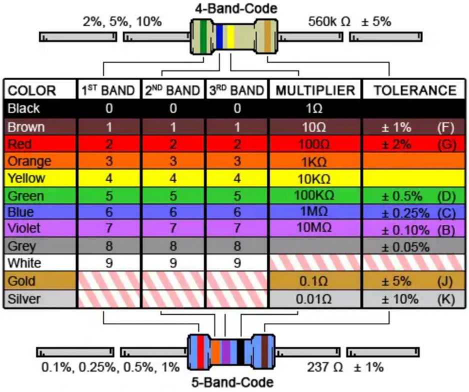


Figure 10: Paralel association: $\frac{1}{R_{\text{eq}}} = \frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_n}$

Resistors II



COLOR	1ST BAND	2ND BAND	3RD BAND	MULTIPLIER	TOLERANCE
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	± 1% (F)
Red	2	2	2	100Ω	± 2% (G)
Orange	3	3	3	1KΩ	
Yellow	4	4	4	10KΩ	
Green	5	5	5	100KΩ	± 0.5% (D)
Blue	6	6	6	1MΩ	± 0.25% (C)
Violet	7	7	7	10MΩ	± 0.10% (B)
Grey	8	8	8		± 0.05%
White	9	9	9		
Gold				0.1Ω	± 5% (J)
Silver				0.01Ω	± 10% (K)

Figure 11: How to read a resistor

Toggle Components

They toggle the loop where they are placed, opening or closing said loop.

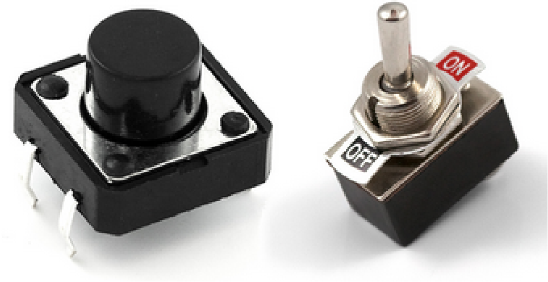


Figure 12: A press button and a toggle switch

Capacitors in a DC circuit

Capacitors are components with relevant capacitance.

- **Capacitance**(SI: Farad): The ability of a material to store electrical charge. In a DC circuit:

$$C = \frac{q}{U}$$

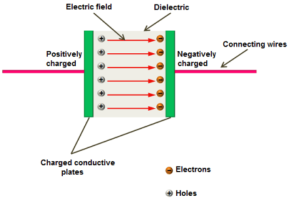


Figure 13: A capacitor schematic

Why is this relevant?. . . .

A pratical example - the RC circuit

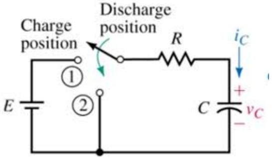


Figure 14: A resistor-capacitor circuit

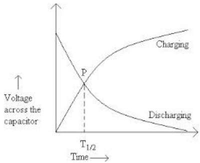


Figure 15: The capacitor's charge and discharge graphs with $\tau = RC$

And capacitors have many more appliances such as filtering, shielding, etc.

Diodes

Diodes are components that limit the flow of electrical current in a single orientation. Nowadays most diodes are composed of semiconductors in a p-n junction, which is also the material basis for another important component, the **transistor**.

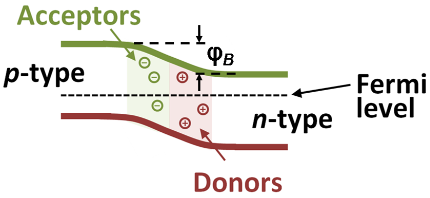


Figure 16: P-N junction

Diodes have a limited use in DC circuits, however we can use them to protect other components from inverted polarity, to construct diode logic gates, etc., or. . .

LEDs

Arguably the most used diodes in DC circuits.

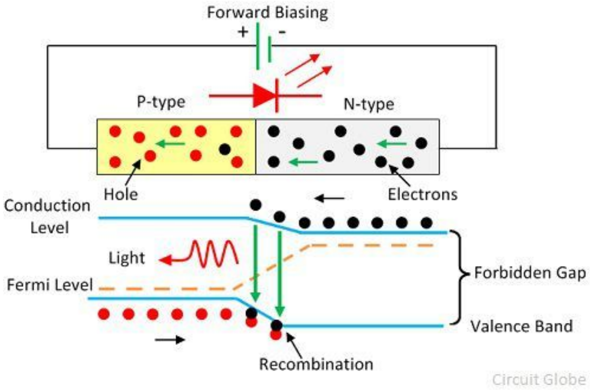


Figure 17: The inner workings of a LED

Integrated Circuits

Integrated circuits have a miriad of functions from amp-ops, to integrated diodes, to timers and even logic gates. They are composed of tiny MOSFE transistors.

Figure 18: The Signetic's 555 timer IC

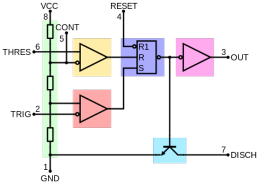


Figure 19: Logic diagram of the 555 timer

Integrated Circuits II

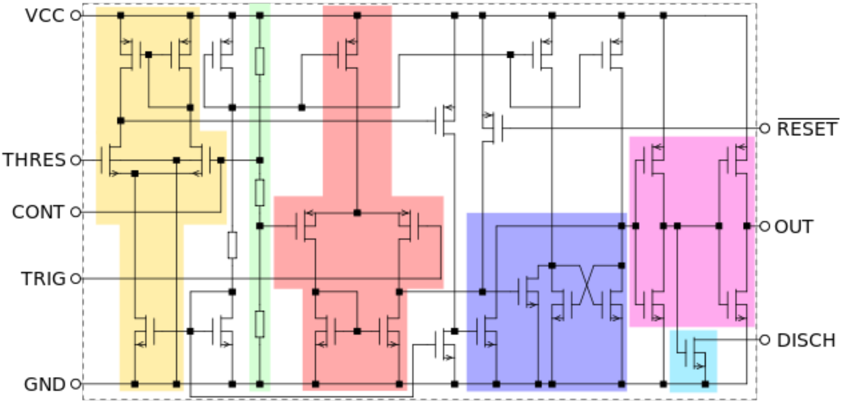


Figure 20: CMOS diagram of the 555 timer

Arduino

Arduino boards are microcontrollers under a CC-BY-SA license.

They allow the integration of digital and analogic control, UART bus communication, etc. into electronic circuits, expanding their potential.

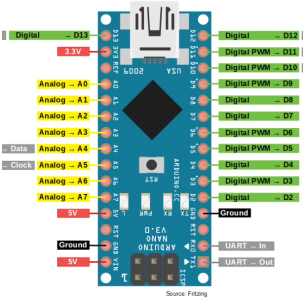


Figure 21: Arduino Nano pinout

Arduino IDE

The Arduino IDE, is an integrated development enviroment under LGPL.

It allows serial communication with a Arduino board, compilation of .ino files (based on C++) and flashing of said binaries into a Arduino Board. Recently a 2.0 version of the IDE has been released.

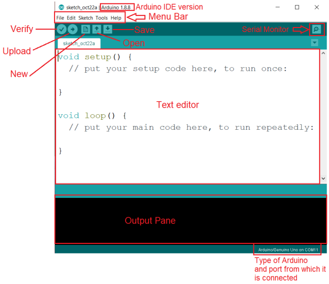


Figure 22: Arduino IDE

C++ vs Python

- C++ is compiled, while Python is interpreted.
- Statements end with a semicolon (;), brackets are used to define block of code, instead of indentation.
- Variables are statically typed, instead of dynamically typed.
- Lower level, closer to the hardware, allows for more control.

Variables

Python

```python
foo = 1
bar = 3.1415 # sample text
baz = True
```

C++

```cpp
int foo = 1;
float bar = 3.1415; // sample text
bool baz = true;
```

Functions

Python

```python
def add(a, b):
    return a + b
```

C++

```cpp
int add(int a, int b) {
    return a + b;
}
```

Conditional Statements

Python

```python
if bar > 0:
    return bar
elif bar < 0:
    return -bar
else:
    return 0
```

C++

```cpp
if (bar > 0) {
    return bar;
} else if (bar < 0) {
    return -bar;
} else {
    return 0;
}
```

While Loops

Python

```python
foo = 0
while foo < 10:
    foo += 1
    # do something
```

C++

```cpp
int foo = 0;
while (foo < 10) {
    foo += 1;
    // do something
}
```

For Loops

Python

```python
for foo in range(10):
    # do something
```

C++

```cpp
for (int foo = 0; foo < 10; foo += 1) {
    // do something
}
```

Controlling your Arduino with C++

Program Structure

```
void setup() {
    // code here runs once, when the board is powered on
}

void loop() {
    // code here runs repeatedly, forever
}
```

Serial

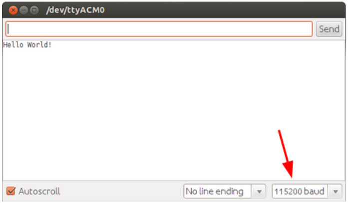


Figure 23: Serial Monitor

```
void setup() {
    Serial.begin(9600);
}

void loop() {
    Serial.println("Hello World!");
}
```

Delay

```
int secondsElapsed;

void setup() {
    Serial.begin(9600);
    secondsElapsed = 0; // start at 0
}

void loop() {
    Serial.print(secondsElapsed);
    Serial.println(" seconds have passed");
    secondsElapsed += 1;
    delay(1000); // wait for a second before repeating
}
```

Figure 24: It's always time for a JoJo reference

```
0 seconds have passed
1 seconds have passed
2 seconds have passed
3 seconds have passed
4 seconds have passed
5 seconds have passed
(...)
```

IO functions

- `pinMode(pin, mode)`: sets the mode of a pin to either `INPUT` or `OUTPUT`.
- `digitalWrite(pin, value)`: sets the value of a pin to either `HIGH` or `LOW`.
- `digitalRead(pin)`: returns the value of a pin, either `HIGH` or `LOW`.
- `analogWrite(pin, value)`: sets the value of a pin to a value between 0 and 255.
- `analogRead(pin)`: returns the value of a pin, between 0 and 1023.

LED Blink

```
int ledPin = 13;

void setup() {
    pinMode(ledPin, OUTPUT); // set LED pin as output
}

void loop() {
    digitalWrite(ledPin, HIGH); // turn on the LED
    delay(1000); // wait for a second
    digitalWrite(ledPin, LOW); // turn off the LED
    delay(1000); // wait for a second
}
```

Button

```
int buttonPin = 2;

void setup() {
    pinMode(buttonPin, INPUT); // set pin 2 as input
    Serial.begin(9600);
}

void loop() {
    if (digitalRead(buttonPin) == HIGH) {
        Serial.println("Button pressed");
    }
}
```

Project