Arduino for the Arts

Fred Cahill, Shawn Nock

Unlondon Digital Media Assoc.

What's in your kit?

Kit Contents

- Arduino Uno R3 Clone
- Solderless Breadboard
- Connecting wires
- LEDs
- Resistors, Potentiometer
- Buzzer
- IR Remote
- IR Receiver

What is Arduino?

μC + reset button + led + USB communication

It's a kit (on a board) with the bare minimum components to easily use the μC hardware. They do the basic, boring design needed for any board, so users only need to add the neat stuff.

Arduino UNO

The Arduino variety that we are using is the Arduino UNO.

- Processor: Atmel Atmega328p
- Memory: 2K RAM + 32K Flash
- FT232RL Logic-level Serial↔USB Chip

Arduino Software

The Arduino folks also adapted an *Integrated Development Environment* (IDE) to their boards. This IDE allows users to easily write programs for their boards and then write the programs to the μC.

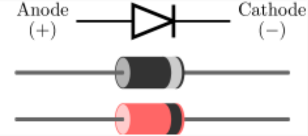
Get the Arduino IDE:

`https://www.arduino.cc/en/Main/Software`

Circuit Basics

Diode

- One way value for current[1]
- LED ≡ Light Emitting Diode
- Band marks (-)[2]
- Longer leg marks (+)



[1] https://learn.sparkfun.com/tutorials/diodes

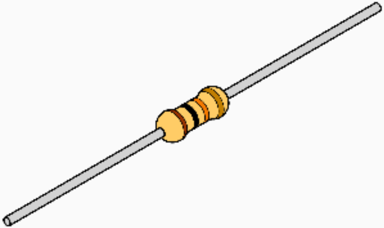
[2] https://learn.sparkfun.com/tutorials/polarity/diode-and-led-polarity

Diode Problems

- Diodes don't limit current
- Diodes aren't perfect (some current turned to heat)
- Too much current = Too much heat = **BANG**
- How do we limit current?

Resistor

- *Resist* the flow of current
- Needed for LEDs: $\approx 400\,\Omega$ (safe for $\leq 6\,\mathrm{V}$)
- Button Pull-up/down: $\geq 10\,\mathrm{k}\Omega$
- Color coded, Google it



Ohm's Law

Ohm's Law relates current to potential and resistance.

$$V = IR$$
$$I = \frac{V}{R}$$
$$R = \frac{V}{I}$$

- V = Potential in Volts (V)
- I = Current in Amperes (A)
- R = Resistance in Ohms (Ω)

Ohm's Law: Example

The datasheet for an LED says that the maximum continuous current is 15 mA. Your circuit operates at 5 V[1]. How big should your resistor be?

$$\Omega = \frac{5\,\mathrm{V}}{0.015\,\mathrm{A}} = 333.\overline{3}\Omega$$

How much current for our *cheet sheet* value?

$$\mathrm{A} = \frac{5\,\mathrm{V}}{400\,\Omega} = 12.5\,\mathrm{mA}$$

[1] Actually, this calculation is inaccurate. LEDs will have a *forward voltage drop* of between 300 mV and 700 mV this should be subtracted from V above... but it's not critical.

Buttons

- Buttons connect *or* disconnect two wires/parts
- Momentary Switch: Normally Closed (NC), Normally Open (NO)
- Toggle Switch

Digital Signals

- Vcc: The power supply of the digital circuit elements
- GND: The reference voltage (usually 0 V)
- Connecting a part to Vcc = Logical 1
- Connecting to GND = Logical 0

Transducers

Transducers turn electrical energy into another sort of energy:

Speaker	Electrical → Sound
Microphone	Sound → Electrical
LED	Electrical → Light
LED	Light → Electrical
Piezoelectric	Electrical → Motion

Piezo Buzzer

- Piezoelectric elements change shape when voltage is applied
- Thin discs can be made to oscillate and create sound.
- Contains oscillator circuit
- Two connections: Vcc, GND
- Use a switch; connected = annoying tone, disconnected = glorious silence

Power

The power supply provides the energy to drive the system *and* defines logical 1.

Can be a:

- Voltage Regulator (converts one potential to another)
- Batteries (Lemon, NiMH, LiPo)
- Solar Panel

In our circuits, your laptop is converting it's power source to 5V and delivering power to our circuit via USB. You also have a battery pack for computer-free shenanigans.

μController

Microcontroller (μC) is a *processor*, *memory* and a few *peripherals* on a standalone chip.

Processor is a group of transistors that understands a few dozen commands (ADD, SUB, JUMP..)

Memory a circuit that can hold values.

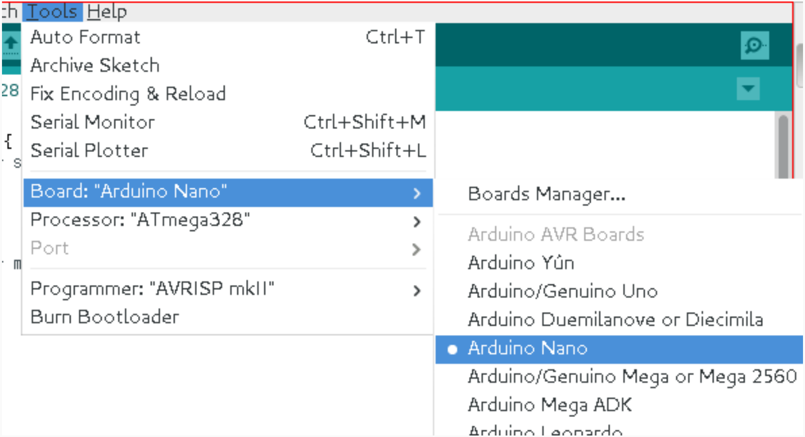
Peripherals Vary chip to chip, but often include timers, radios, communication interfaces

Seems complicated, but really simple. They literally read a command (and data) from memory, then execute the command. At the end of the command, the next command is read from the next memory cell and the process is repeated[1]

[1] some commands change the next command memory address

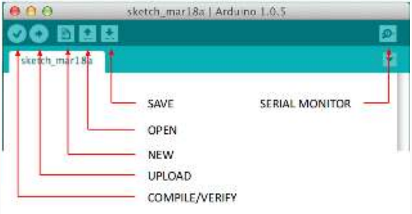
Let’s start programming

Configure Arduino



- Board: Arduino UNO
- Processor: ATmega328
- Port: . . .

The Code Environment

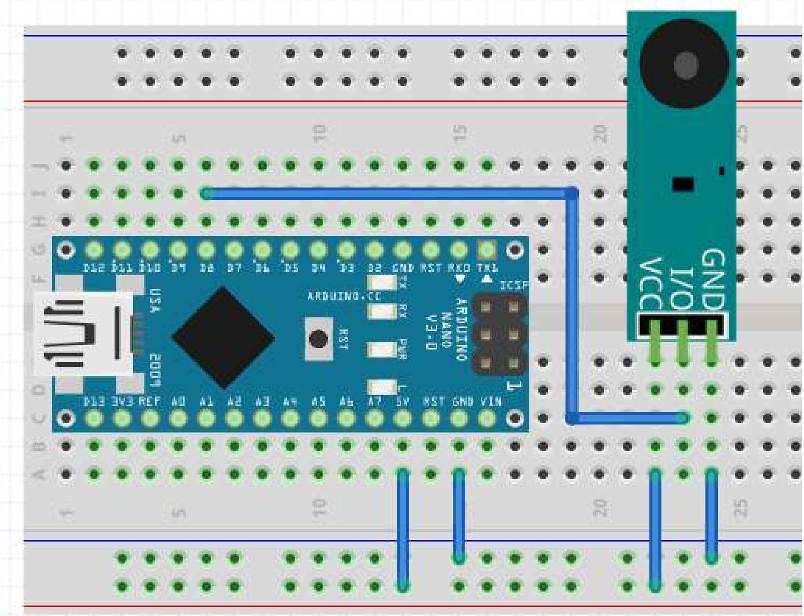


Your first Program

```
/* the setup function runs once on reset / power */
void setup() {
  /* set pin 13 as an output */
  pinMode(13, OUTPUT);
}

/* the loop function repeats forever */
void loop() {
  digitalWrite(13, HIGH);   // turn on LED
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the off LED
  delay(1000);              // wait for a second
}
```

Buzzer: Hardware



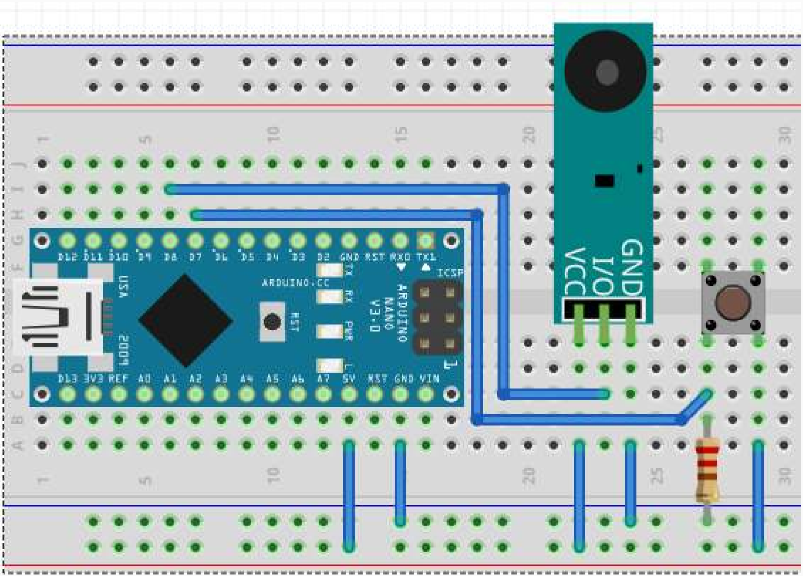
D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RXD TX1
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN
ARDUINO.CC
ARDUINO NANO V3.0
ICSP
RST
TX
RX
PWR
L
USA
2009
GND
I/O
VCC

Buzzer: Software

```
#define BUZZER 8       /* Make BUZZER same as pin 8 */

void setup() {
  pinMode(BUZZER, OUTPUT);
  digitalWrite(BUZZER, HIGH);    /* Turn off buzzer */
}
void loop() {
  digitalWrite(BUZZER, LOW);      /* Turn on buzzer */
  delay(100);                     /* wait for 100ms */
  digitalWrite(BUZZER, HIGH);    /* Turn off buzzer */
  delay(900);                         /* wait 900ms */
}
```

Push Button: Hardware


D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN
ARDUINO.CC
ARDUINO
NANO
V3.0
ICSP
USA
2009
RST
TX
RX
PWR
L
GND
I/O
VCC
1
5
10
15
20
25
30
J
I
H
G
F
E
D
C
B
A

Push Button: Software (Part 1)

```
#define BUTTON 7
#define BUZZER 8

int button_state = 0;

void setup() {
  pinMode(BUTTON, INPUT);
  pinMode(BUZZER, OUTPUT);
  digitalWrite(BUZZER, HIGH);
}
```

Push Button: Software (Part 2)

```
void loop() {
  button_state = digitalRead(BUTTON);
  if (button_state == HIGH) {
    digitalWrite(BUZZER, LOW);
  } else {
    digitalWrite(BUZZER, HIGH);
  }
}
```

The End?

Questions?