

# ARDUINO

Brought to you by: The Transistor

# About us, and why we're here.

Zack



Aaron

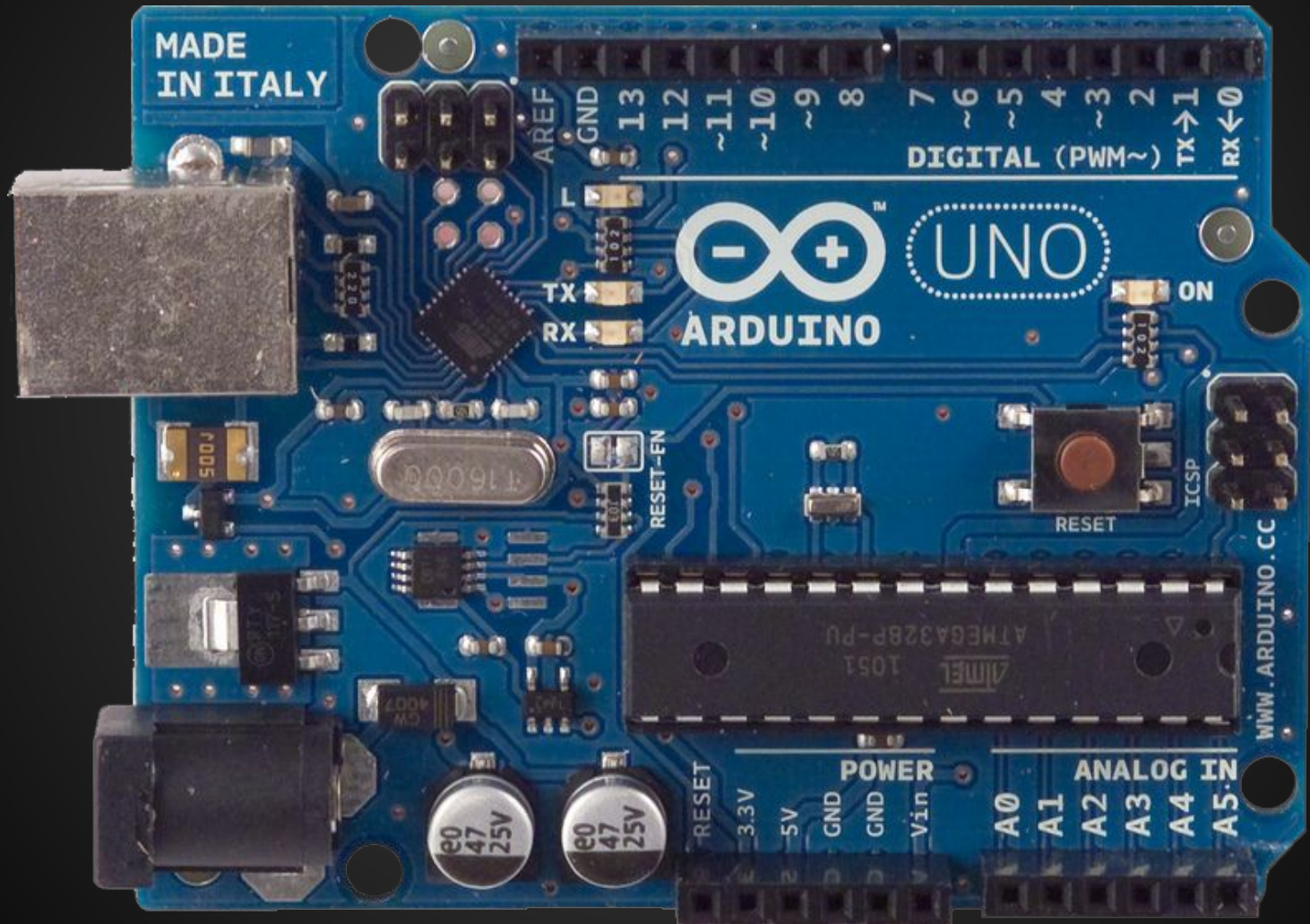


We're both active members a hackerspace  
in provo, The Transistor.

# About us, and why we're here.

I have no experience using Arduinos. So I've enlisted the help of Aaron, a fellow member at The Transistor that knows what he's doing with arduinos.

# Basic overview of the Arduino platform



# The Arduino platform

- IDE (programming environment)
- C/C++ microcontroller libraries
- Dev board built around Atmel ATmega series microcontrollers.

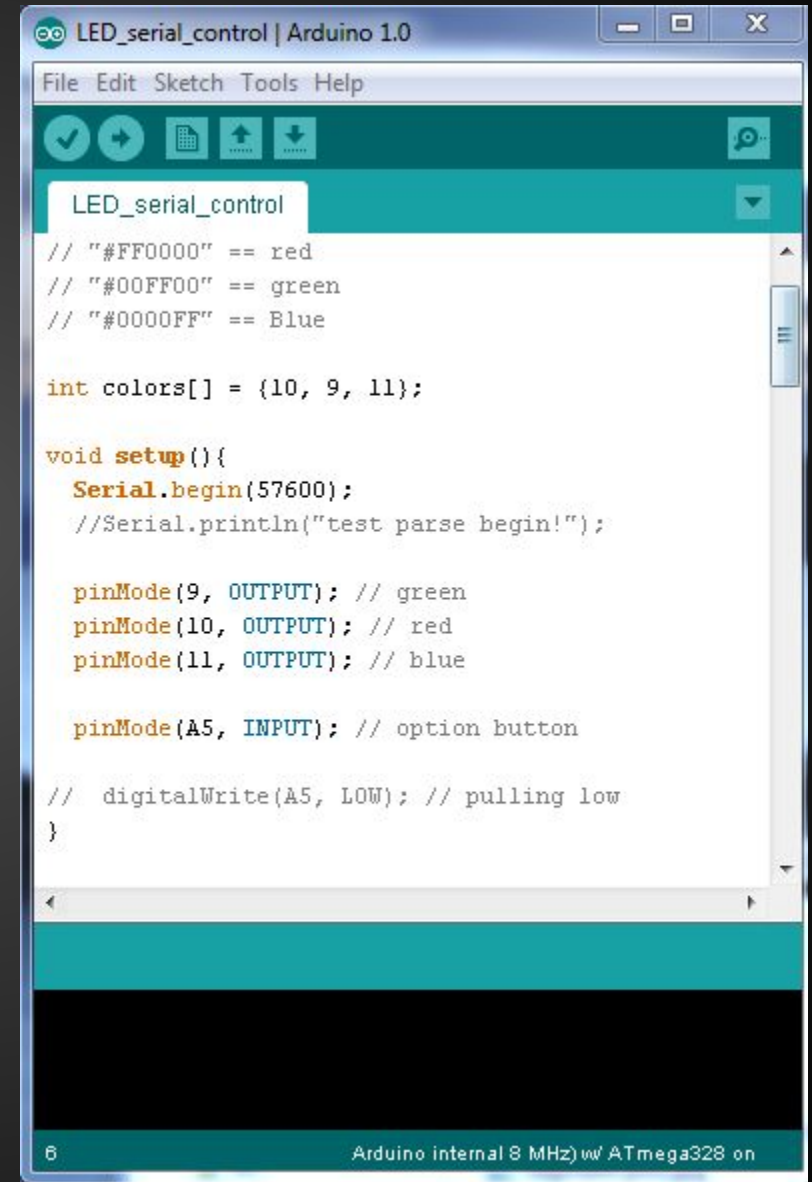
# Who should use Arduino?

Everybody!



# IDE

- Written in Java
- Simple
- Handles compiling and uploading sketches automatically



The screenshot shows the Arduino IDE interface with a sketch named "LED\_serial\_control" open. The code is written in C++ and includes comments for color definitions, pin configurations, and setup functions. The IDE window has a title bar "LED\_serial\_control | Arduino 1.0" and a menu bar with "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checking, running, saving, and uploading. The code editor shows the following code:

```
// "#FF0000" == red
// "#00FF00" == green
// "#0000FF" == Blue

int colors[] = {10, 9, 11};

void setup(){
  Serial.begin(57600);
  //Serial.println("test parse begin!");

  pinMode(9, OUTPUT); // green
  pinMode(10, OUTPUT); // red
  pinMode(11, OUTPUT); // blue

  pinMode(A5, INPUT); // option button

  // digitalWrite(A5, LOW); // pulling low
}
```

The status bar at the bottom indicates "6" and "Arduino internal 8 MHz w/ ATmega328 on".

# I/O

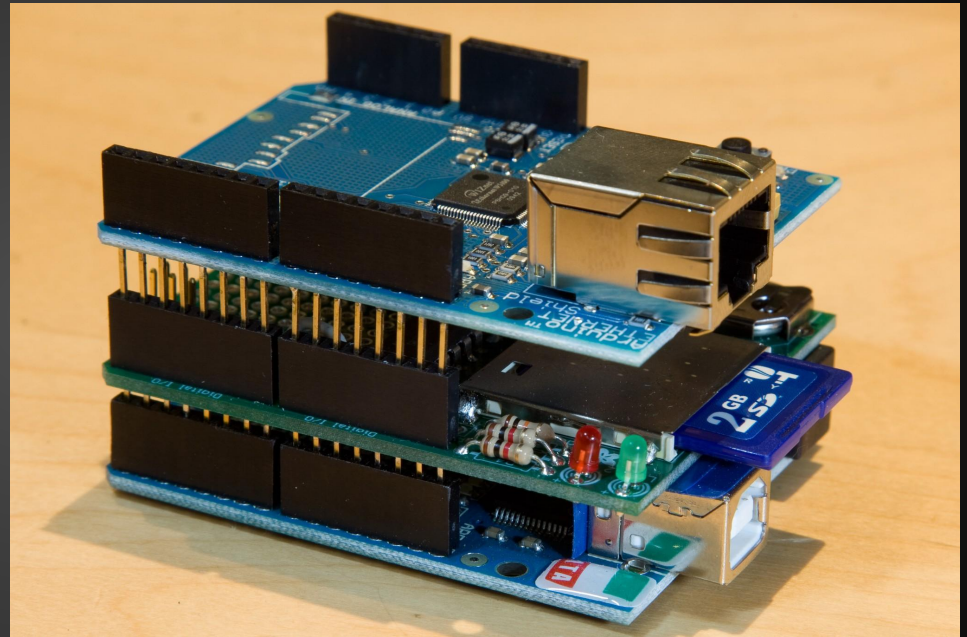
Digital I/O: 14 (of which 6 provide PWM output)

Analog Input: 6



# Shields

Shields are boards containing various components that allow you to easily extend the functionality of the Arduino



# Language

- Based on Wiring
- Syntactically similar to C++

// will run at the beginning of sketch, only once

```
void setup(){  
    // setup stuff here  
}
```

```
void loop(){  
    // infinite loop  
}
```

# Language - common functions

## Digital I/O:

- pinMode()
- digitalWrite()
- digitalRead()

## Analog I/O:

- analogReference()
- analogRead()
- analogWrite() - PWM

# Language - blink an LED

```
// Pin 13 has an LED connected on most Arduino boards.
```

```
// give it a name:
```

```
int led = 13;
```

```
// the setup routine runs once when you press reset:
```

```
void setup() {
```

```
    // initialize the digital pin as an output.
```

```
    pinMode(led, OUTPUT);
```

```
}
```

```
// the loop routine runs over and over again forever:
```

```
void loop() {
```

```
    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
```

```
    delay(1000);             // wait for a second
```

```
    digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
```

```
    delay(1000);             // wait for a second
```

```
}
```

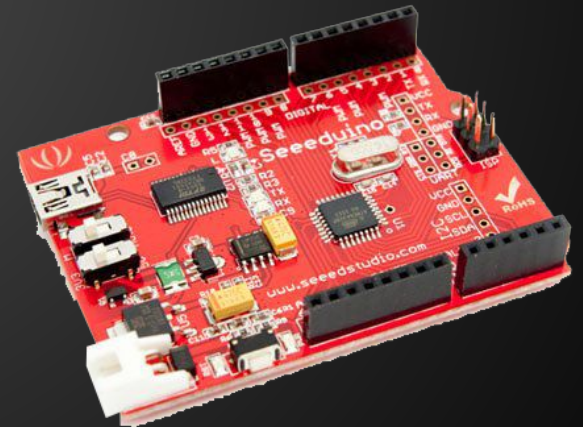
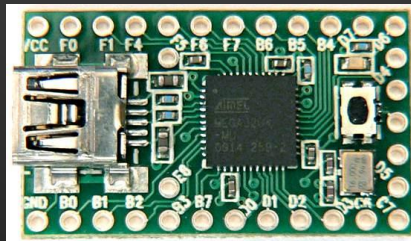
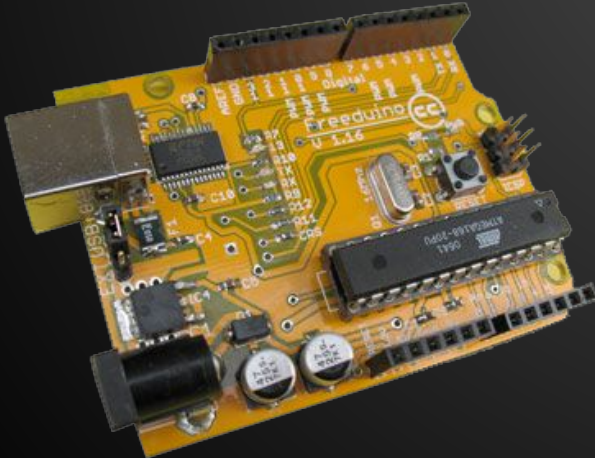
# Android ADK

- Android Accessory Development kit
- Uses Arduino core libraries
- Supports Android 2.3.4+



# Arduino compatible dev boards

- Freeduino
- Teensy
- Seeeduino





# Open Source projects utilizing Arduino

- Amblone (ambient display lighting system)
- Grbl (CNC project)
- Ardupilot
- arducopter
- WSDL (Weather Station Data Logger)
- RepRap (3D printer)

**Let's take a look at one of those  
projects~!!!!!!**

# Ambclone

>> Philips Ambilight.



Photo by Stephan Legachev.

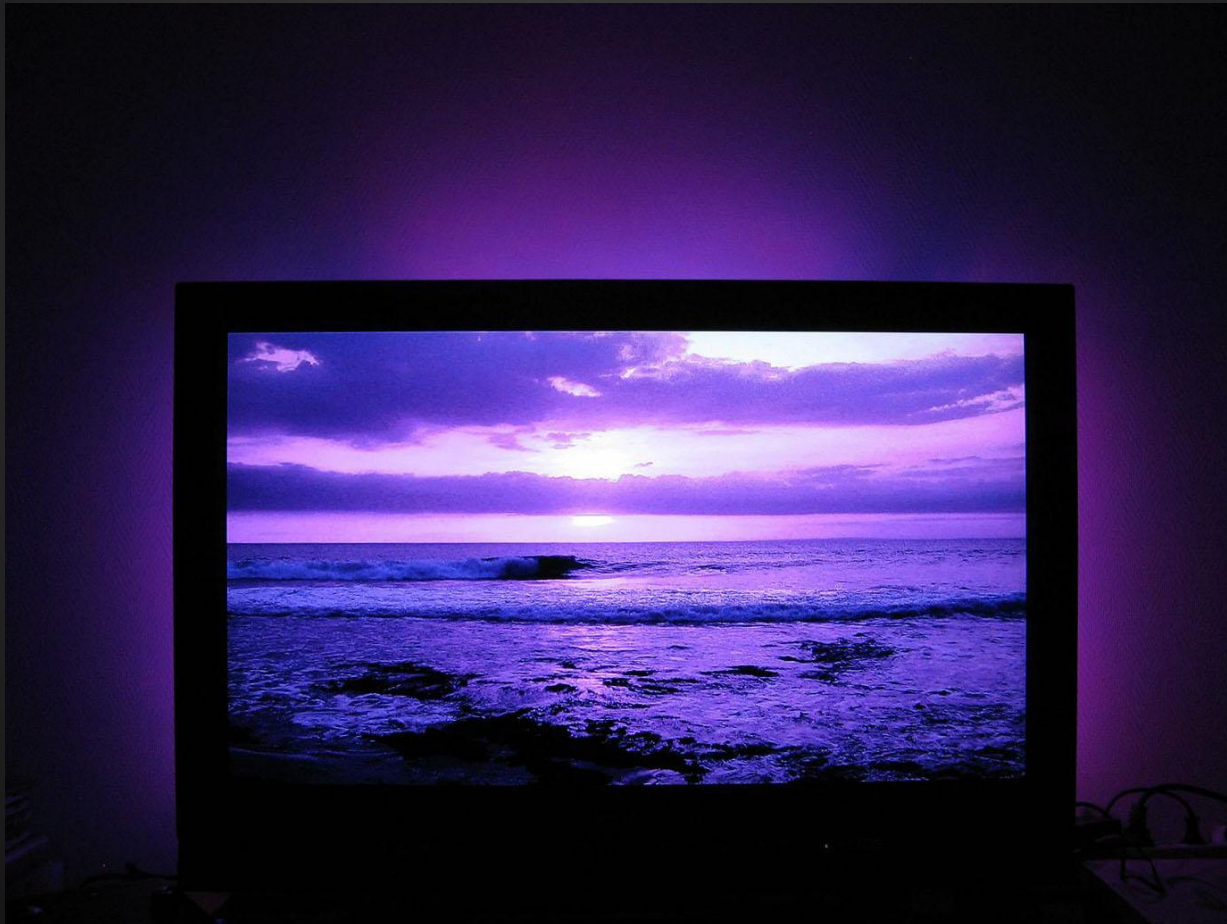
# Amblone

>> Philips Ambilight.

>> Amblone is an open source project that is a clone of Philips Ambilight.

# Amblone

>>And this is what Amblone looks like.



# Materials needed for Amblone

- >>Arduino Mega, Uno, Duemilanova, or other Arduino compatible development boards.
- >>RGB LEDs
- >>Resistors
- >>USB cable
- >>Amblone software
- >>Soldering Iron
- >>Transistors



# How Amblone works

>>Host side software takes the average of the colors on regions that you specify.

>>The Arduino takes the average screen color for the given regions and then takes care of the all the gritty PWM work.

# Practical Applications with Arduinos

>>Weather Station.

>>TF2 indicator lights.

>>Blinkin Lights :p (network traffic indicator)

>>Flight Computer.

>>Magic the Gathering Life Counter.

>>Entry Door control.

>>Spectrum Analyzer.

>>Whatever your imagination can come up with.

Now let's take a look at some alternative development/prototyping platforms.

# Alternative Developer/Prototyping Boards.

- >>Raspberry Pi \$25/35
- >>Beagle Board \$89/149
- >>terasic DE0-Nano \$79
- >>TI Launchpad \$6~15

# Questions