

A close-up photograph of an Arduino Uno R3 printed circuit board (PCB). The board is green and features various components including a USB Type-B port, a DC power jack, a reset button, and a microcontroller chip. The text "MADE IN ITALY" is visible in the top left corner, and "ARDUINO" is printed in the center. The main title "Introduction to ARDUINO" is overlaid in large yellow font.

# Introduction to ARDUINO

<http://arduino.cc>

# Variations



**UNO**  
with AtMega328

# Variations



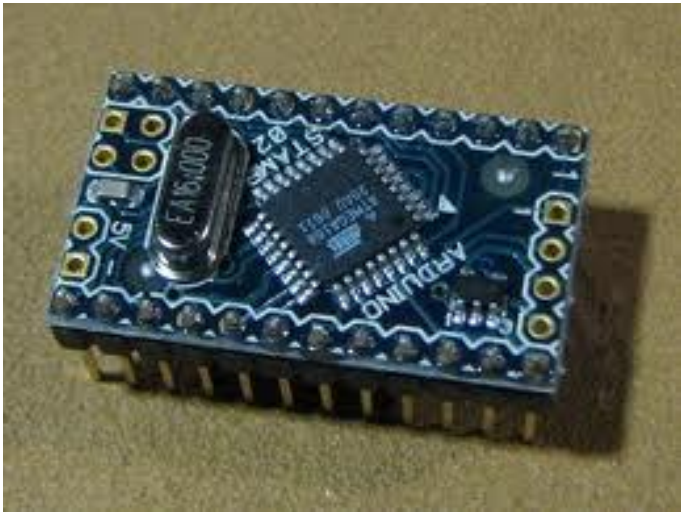
**Duemilanove** (dua-mel-a-nov)  
with AtMega328 or  
AtMega168

# Variations



**MEGA**  
with AtMega2560

# Variations



Older Mini

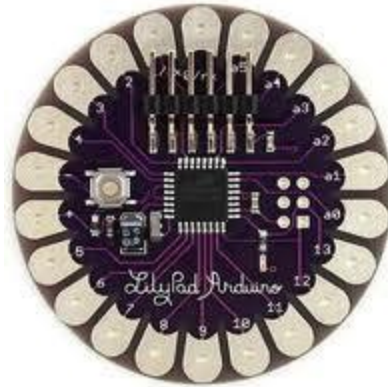


Mini



Nano

# Variations



LilyPad

# More Variations

[Arduino Website](#)

DIY DRONES



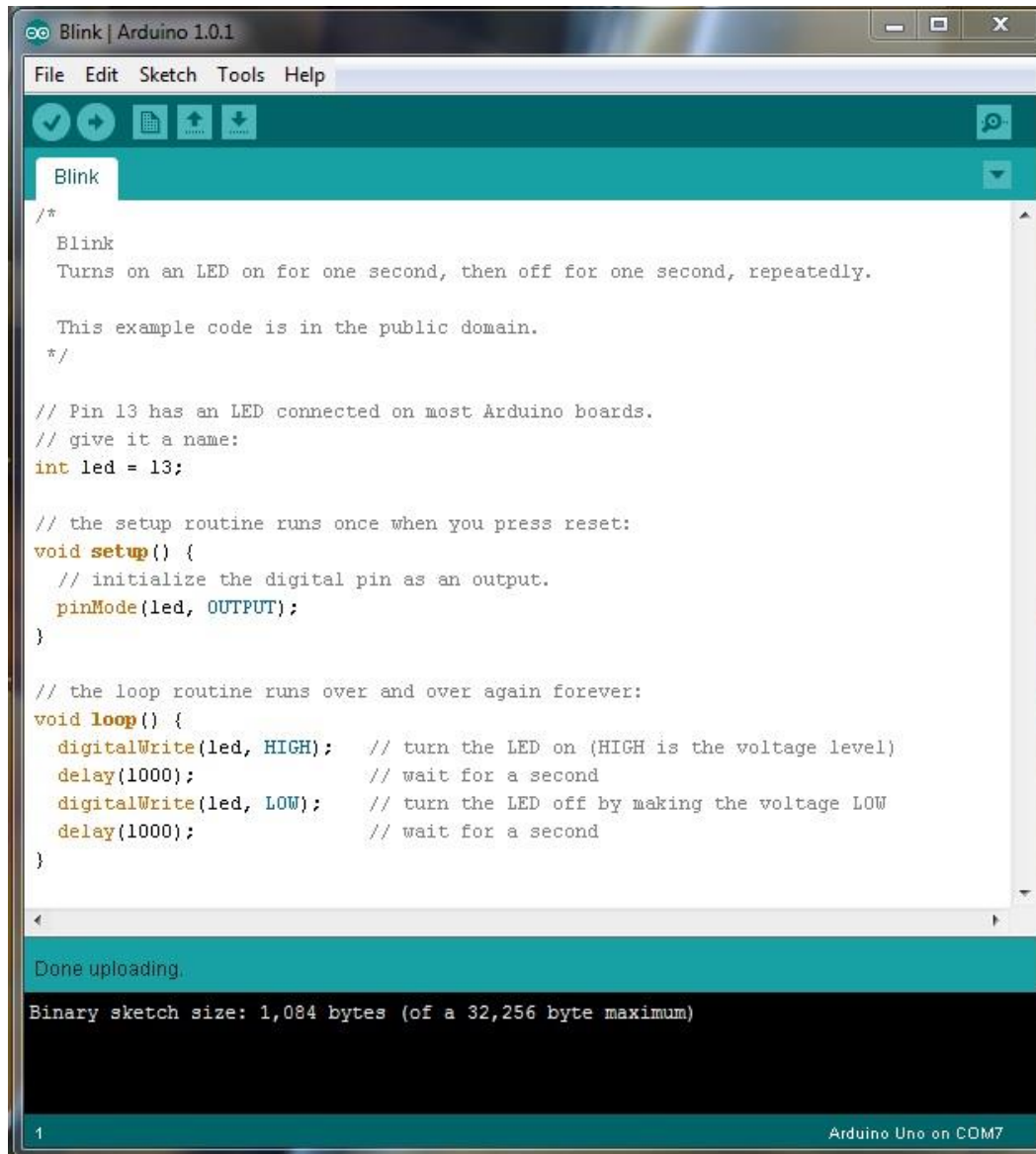
ArduPilot

# More Variations

Board Name	Processor								
	Family	SRAM	FLASH	EEPROM	Clock	UART	PWM	Digital	Analog
<i>Seeeduino Film</i>	ATmega168	1k	16k	512	8MHz	1	6	14	6
<i>LilyPad Simple Board</i>	ATmega168	1k	16k	512	8MHz	1	5	9	4
<i>Arduino Mega 2560</i>	ATmega2560	8k	256k	4k	16MHz	4	14	54	16
<i>Arduino Mega ADK</i>	ATmega2560	8k	256k	4k	16MHz	4	14	50	16
<i>Seeeduino Stalker V2.0</i>	ATmega328	2k	32k	1k	16MHz	1	6	14	6
<i>Uno (R3)</i>	ATmega328	2k	32k	1k	16MHz	1	6	14	6
<i>Duemilanove (328)</i>	ATmega328	2K	32k	1k	16MHz	1	6	14	6
<i>Arduino Ethernet</i>	ATmega328	2k	32k	1k	16MHz	1	4	9	6
<i>Arduino BT</i>	ATmega328	2k	32k	1k	16MHz	1	6	14	6
<i>Arduino Pro Mini 328 5V</i>	ATmega328	2k	32k	1k	16MHz	1	6	14	6
<i>Arduino Nano 3.0</i>	ATmega328	2k	32k	1k	16MHz	1	6	14	8
<i>Arduino Mini</i>	ATmega328	2k	32k	1k	16MHz	1	6	14	8
<i>Arduino Pro 3.3V</i>	ATmega328	2k	32k	1k	8MHz	1	6	14	6
<i>Arduino Pro 5V</i>	ATmega328	2k	32k	1k	16MHz	1	6	14	6
<i>Arduino Fio</i>	ATmega328	2k	32k	1k	8MHz	1	6	14	8
<i>LilyPad 328 Main Board</i>	ATmega328	2k	32k	1k	8MHz	1	6	14	6
<i>Seeeduino</i>	ATmega328	2k	32k	1k	16MHz	1	6	14	6
<i>Seeeduino Ethernet</i>	ATmega328	2k	32k	1k	16MHz	1	6	14	6
<i>Teensyduino</i>	ATmega32U4	2.5k	32k	1k	16MHz	1	7	25	12
<i>Leonardo</i>	ATmega32U4	2.5k	32k	1k	16MHz	1	7	25	12



# Arduino Integrated Development Environment (IDE)



The screenshot shows the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for opening files, saving, uploading, and downloading. The main text area displays the following code:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// 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
}
```

Below the code editor, a status bar indicates "Done uploading." and "Binary sketch size: 1,084 bytes (of a 32,256 byte maximum)". The bottom right corner shows "Arduino Uno on COM7".



# Language Reference

## Language Reference

### Structure

- + `setup()`
- + `loop()`

### Control Structures

- + `if`
- + `if...else`
- + `for`
- + `switch case`
- + `while`
- + `do... while`
- + `break`
- + `continue`
- + `return`
- + `goto`

### Further Syntax

- + `;` (semicolon)
- + `{}` (curly braces)
- + `//` (single line comment)
- + `/** */` (multi-line comment)
- + `#define`
- + `#include`

### Arithmetic Operators

- + `=` (assignment operator)
- + `+` (addition)
- + `-` (subtraction)

### Variables

#### Constants

- + `HIGH` | `LOW`
- + `INPUT` | `OUTPUT` | `INPUT_PULLUP`
- + `true` | `false`
- + `integer constants`
- + `floating point constants`

#### Data Types

- + `void`
- + `boolean`
- + `char`
- + `unsigned char`
- + `byte`
- + `int`
- + `unsigned int`
- + `word`
- + `long`
- + `unsigned long`
- + `float`
- + `double`
- + `string` - char array
- + `String` - object
- + `array`

#### Conversion

- + `char()`

### Functions

#### Digital I/O

- + `pinMode()`
- + `digitalWrite()`
- + `digitalRead()`

#### Analog I/O

- + `analogReference()`
- + `analogRead()`
- + `analogWrite()` - PWM

#### Advanced I/O

- + `tone()`
- + `noTone()`
- + `shiftOut()`
- + `shiftIn()`
- + `pulseIn()`

#### Time

- + `millis()`
- + `micros()`
- + `delay()`
- + `delayMicroseconds()`

#### Math

- + `min()`
- + `max()`
- + `abs()`
- + `constrain()`

- + `-` (subtraction)
- + `*` (multiplication)
- + `/` (division)
- + `%` (modulo)

#### Comparison Operators

- + `==` (equal to)
- + `!=` (not equal to)
- + `<` (less than)
- + `>` (greater than)
- + `<=` (less than or equal to)
- + `>=` (greater than or equal to)

#### Boolean Operators

- + `&&` (and)
- + `||` (or)
- + `!` (not)

#### Pointer Access Operators

- + `*` dereference operator
- + `&` reference operator

#### Bitwise Operators

- + `&` (bitwise and)
- + `|` (bitwise or)
- + `^` (bitwise xor)
- + `~` (bitwise not)
- + `<<` (bitshift left)
- + `>>` (bitshift right)

- + `char()`
- + `byte()`
- + `int()`
- + `word()`
- + `long()`
- + `float()`

#### Variable Scope & Qualifiers

- + `variable scope`
- + `static`
- + `volatile`
- + `const`

#### Utilities

- + `sizeof()`

- + `constrain()`
- + `map()`
- + `pow()`
- + `sqrt()`

#### Trigonometry

- + `sin()`
- + `cos()`
- + `tan()`

#### Random Numbers

- + `randomSeed()`
- + `random()`

#### Bits and Bytes

- + `lowByte()`
- + `highByte()`
- + `bitRead()`
- + `bitWrite()`
- + `bitSet()`
- + `bitClear()`
- + `bit()`

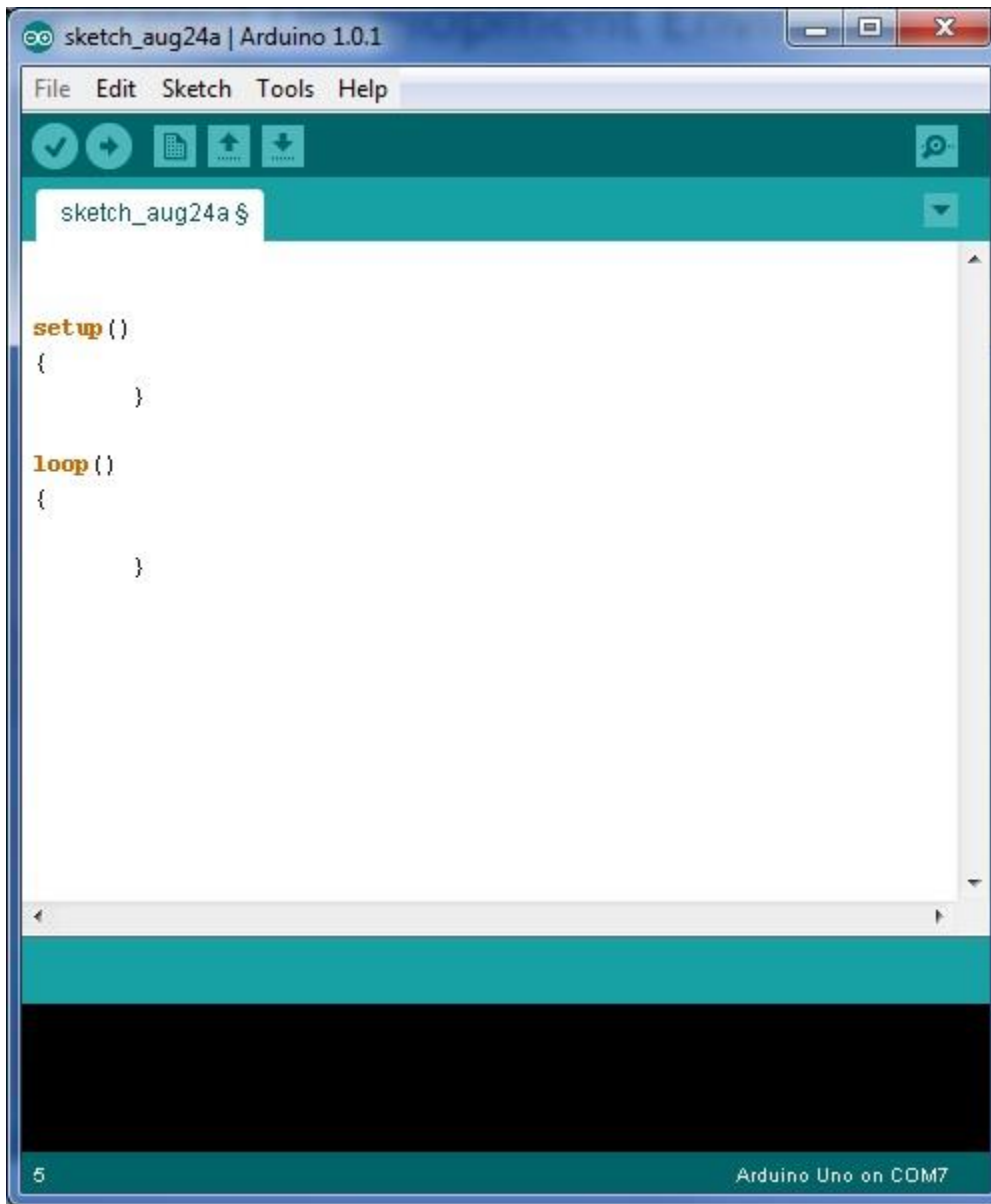
#### External Interrupts

- + `attachInterrupt()`
- + `detachInterrupt()`

#### Interrupts

# Program Structures

## Required Functions



The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_aug24a | Arduino 1.0.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checkmark, refresh, document, upload, download, and a gear. A status bar at the top shows "sketch\_aug24a \$". The main editor area contains the following code:

```
setup()
{
}

loop()
{
}
```

At the bottom of the IDE, a status bar shows "5" on the left and "Arduino Uno on COM7" on the right.

### setup()

- ❖ Use it to initialize
  - Variables
  - pin modes
  - start using libraries, etc.
- ❖ **The setup function will only run once**, after each power up or reset of the Arduino board.

### loop()

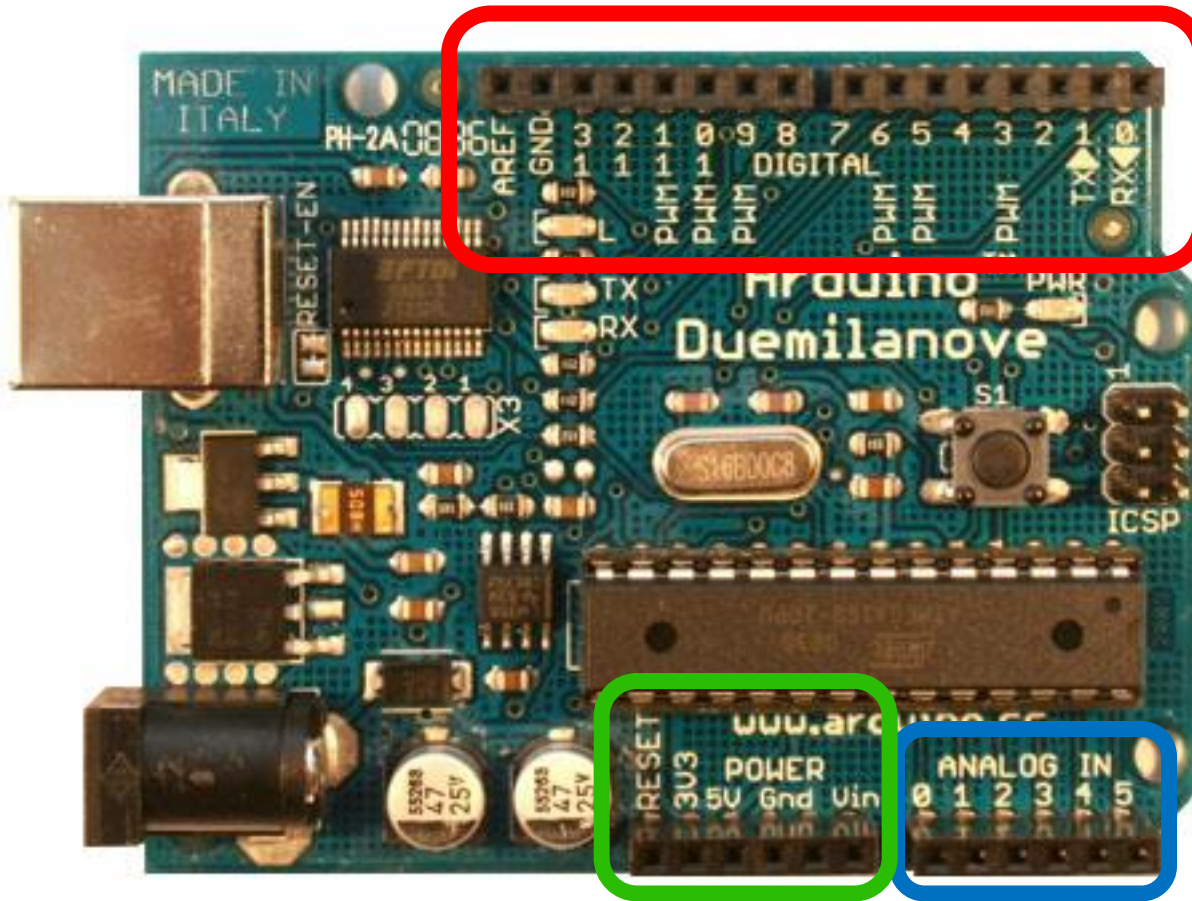
- ❖ The loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

```
int ledPin = 13;           // LED connected to digital pin 13

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

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

# Arduino Pin Function



## Digital I/O Pins:

- 0 through 13
- PWM pins: 3,5,6,9,10,11
- Serial pins: 0 (Rx), 1 (Tx)

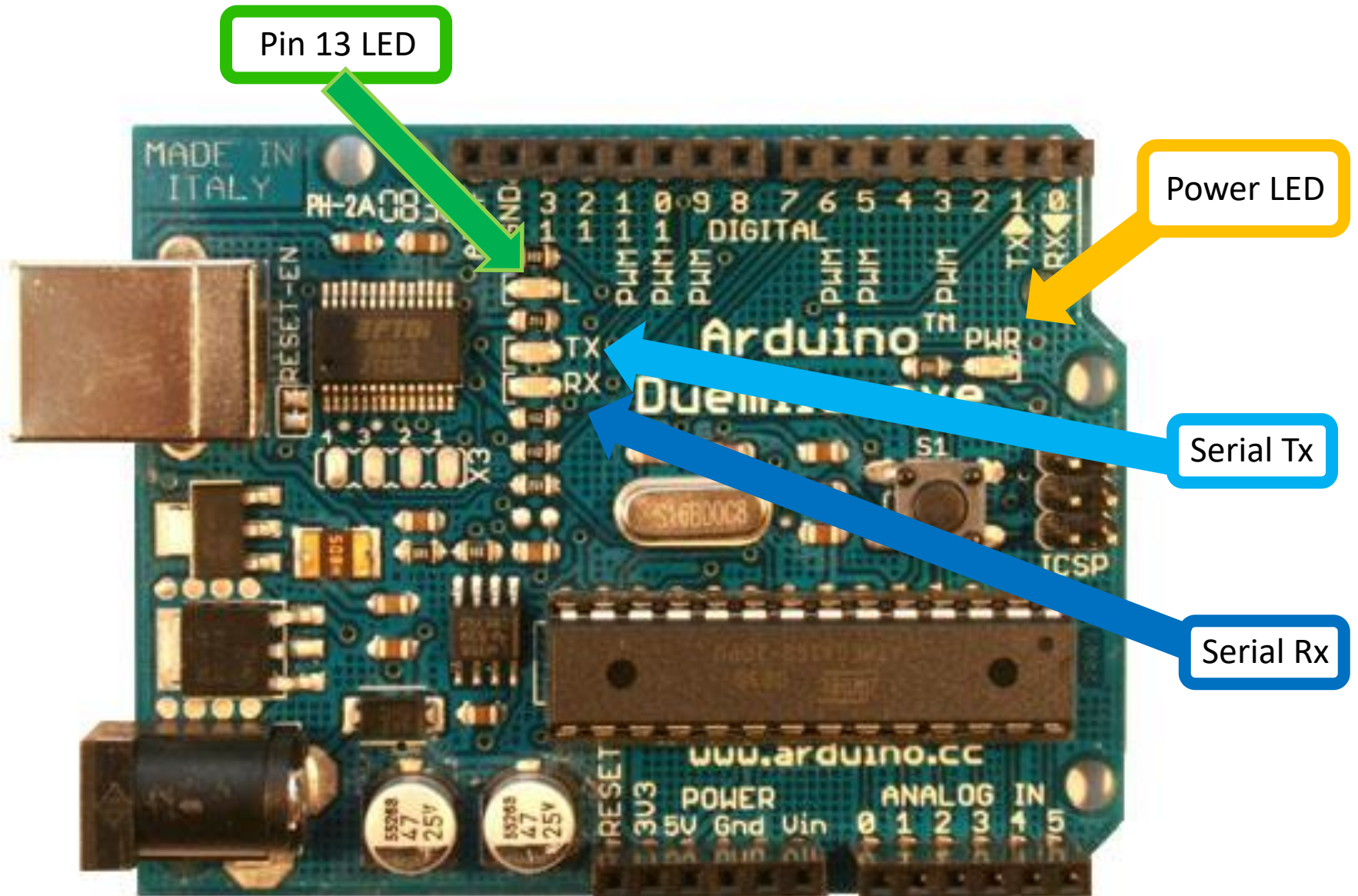
## Analog I/O Pins:

- 0 through 5

## Power Pins:

- RESET
- 3.3vdc
- 5vdc
- Gnd (x2)
- Vin

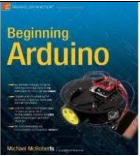

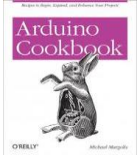
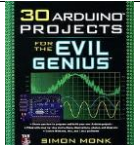

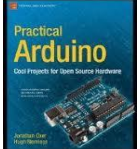

# Arduino LEDs



# Programming

## Recommended books

Tutoring assistance available

	<b>Beginning Arduino</b>	Intermediate level reading and experiments
	<b>Practical Arduino Cool Projects for Open Source Hardware</b>	(whew! Long title) Intermediate to advanced. More towards the Open Source hardware side.
	<b>Arduino Cookbook</b>	<b>**This is a must have for all Arduino fans.</b> Great reference! Each chapter is arranged like a recipe (cookbook). Before starting it lists the ingredients/items you will need to complete the session.
	<b>30 Arduino Projects for the "Evil Genius"</b>	Fun projects for all levels. Great intro to electronics.
	<b>Getting Started with Arduino</b>	<b>**A must-have for all beginners!</b> Search Amazon.com and you may find this book for free or as much as \$2.00
	<b>Practical Arduino</b>	A great reference for Intermediate readers. Explains the processor in detail.
	<b>Making Things Talk (second edition)</b>	This book introduced me to Arduino. Great interfacing projects and ideas to make your project talk!

- **Free!**
- Schedule ahead
- Weekends and evenings
- **Work YOUR project!**
- Structured to your build in module/segmented fashion
- Programming Help
- Email your code for help
- Quick phone answers
- Assist with hardware development and interfacing

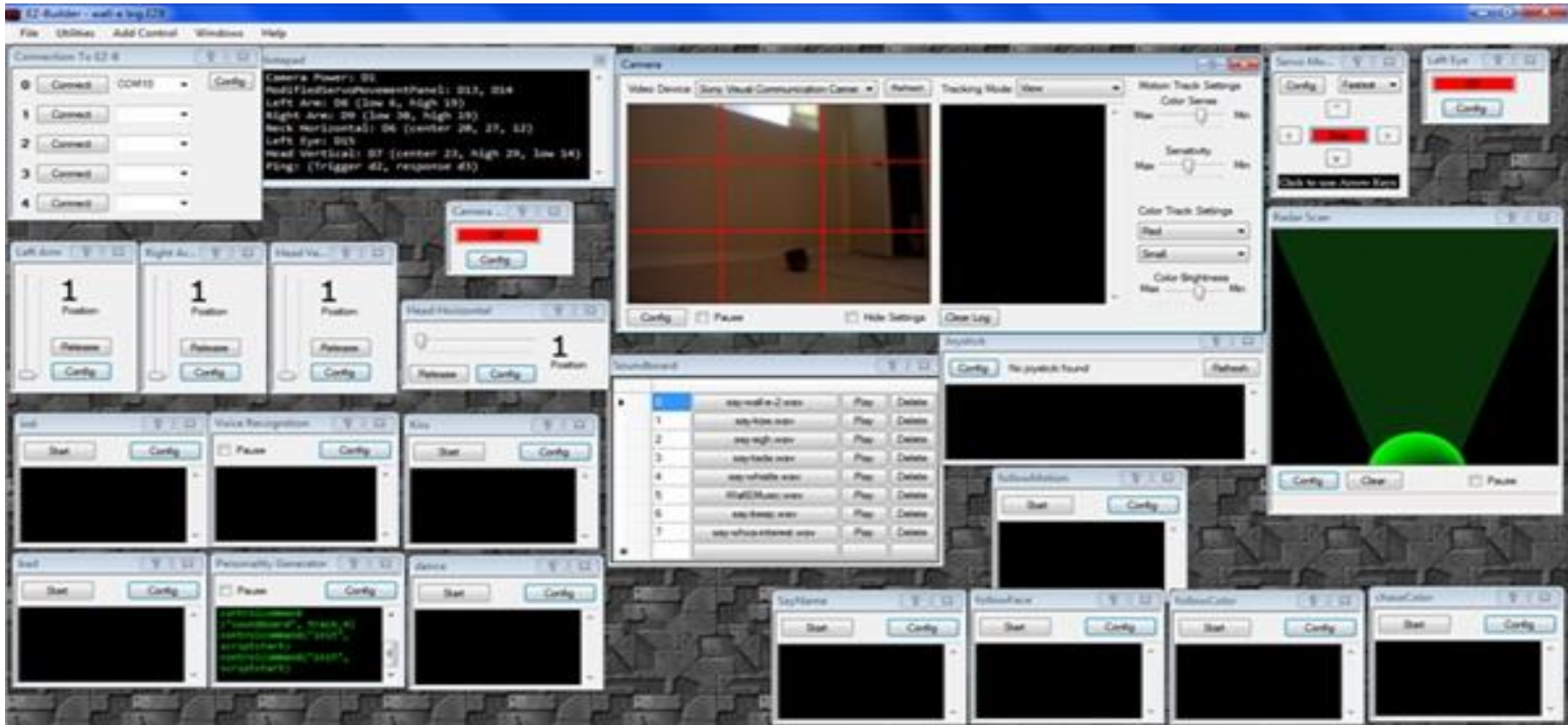
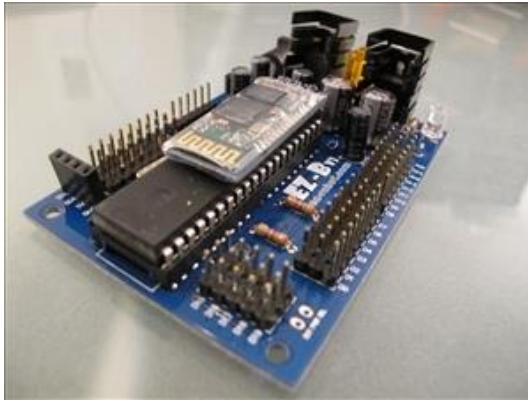
# New to Robotics!

## EZ-Robot Controller

by



# Maker SHED





Questions?