

Arduino

A free development system based on
Atmel AVR microcontrollers.

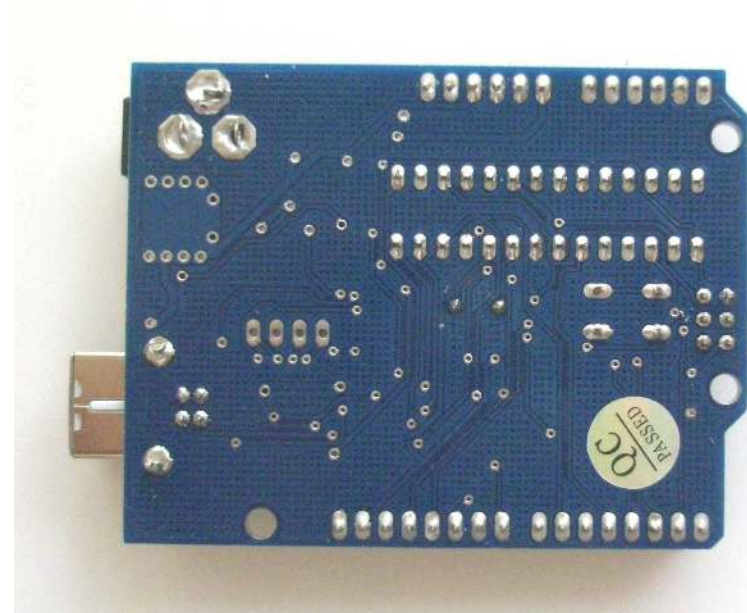
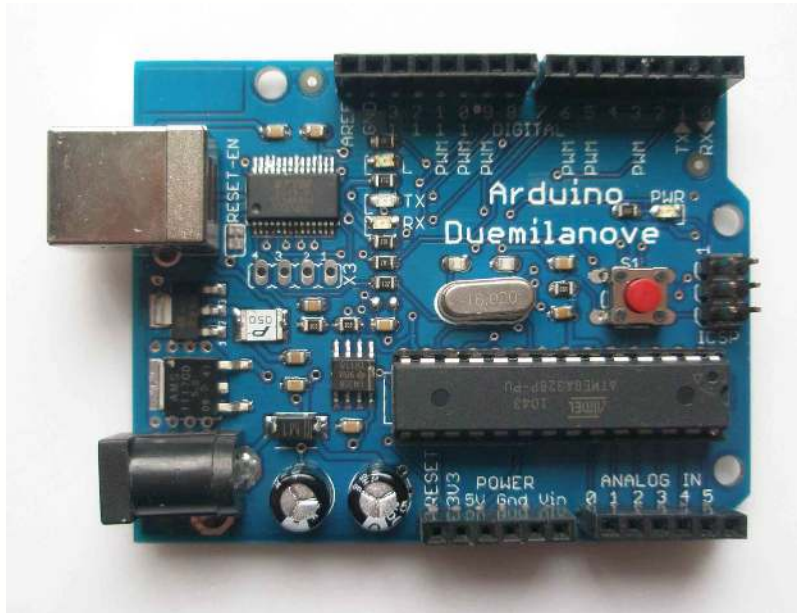
AVR Microcontroller

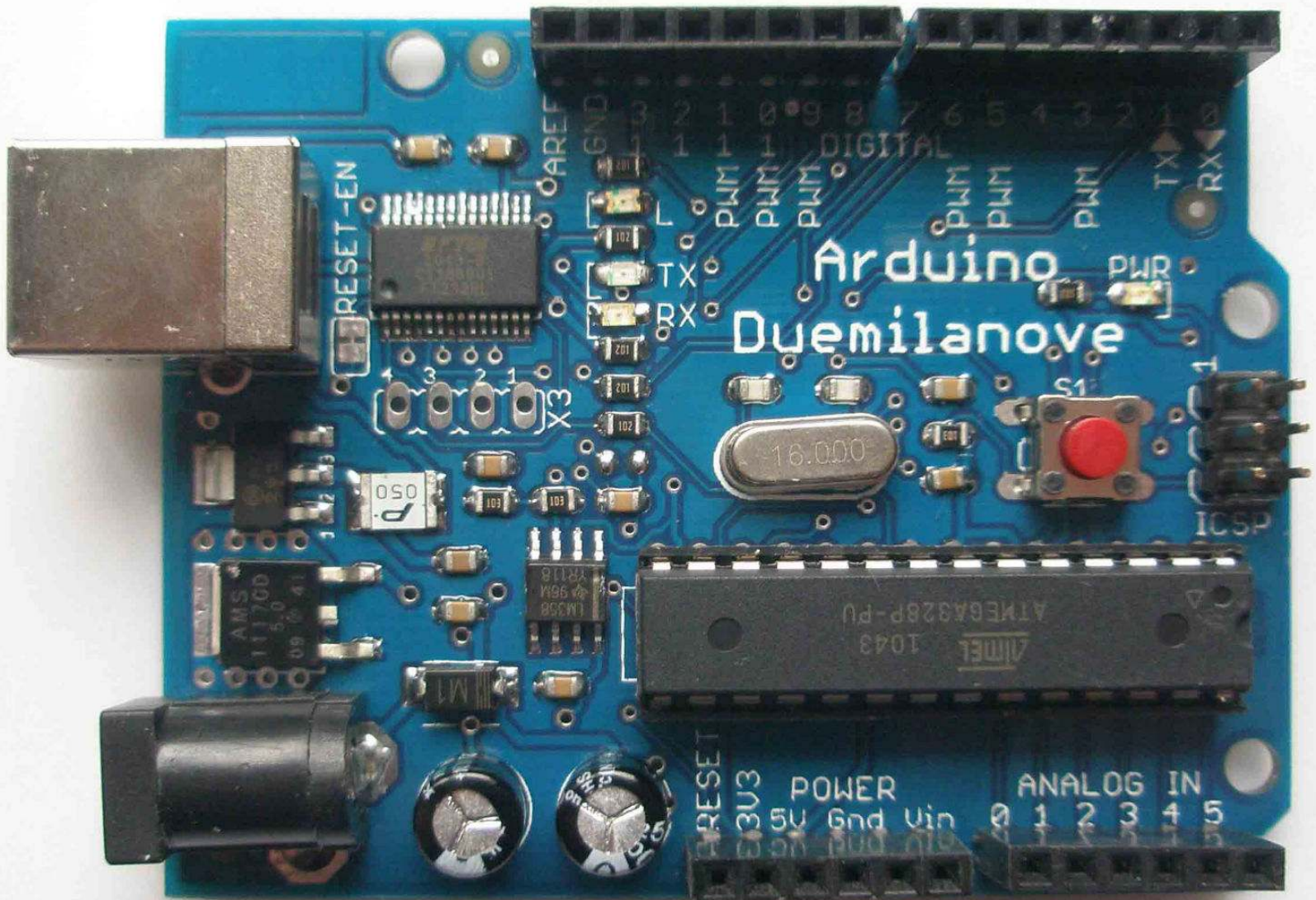
- RISC architecture microcontroller
- Designed for high level languages, developed in Trondheim, Norway in 1996
- Classic: AT90S1200, AT90S2343, AT90S2313, AT90S4433, AT90S8515, AT90S8535
- ATtiny22, ATtiny25-85, ATtiny2313 ...
- ATmega8, ATmega16, ATmega 48-328 ...
- Flash programmable memory
- ATmega self programming
- RAM, EEPROM and peripherals

What is Arduino

- Open Source Hardware, you can make your own board, or buy one.
- Cheap, easily available.
- Open Source Software.
- Very widespread, many projects openly available.
- Extra HW (shields) available.

Arduino Duemilanove (2009)





Arduino
Duemilanove

ATMEL 1043
ATMEGA328P-PU

RESET 3V3 5V Gnd Vin 0 1 2 3 4 5
ANALOG IN

DIGITAL 0 1 2 3 4 5 6 7 8 9 10 11 12 13
PWR TX RX

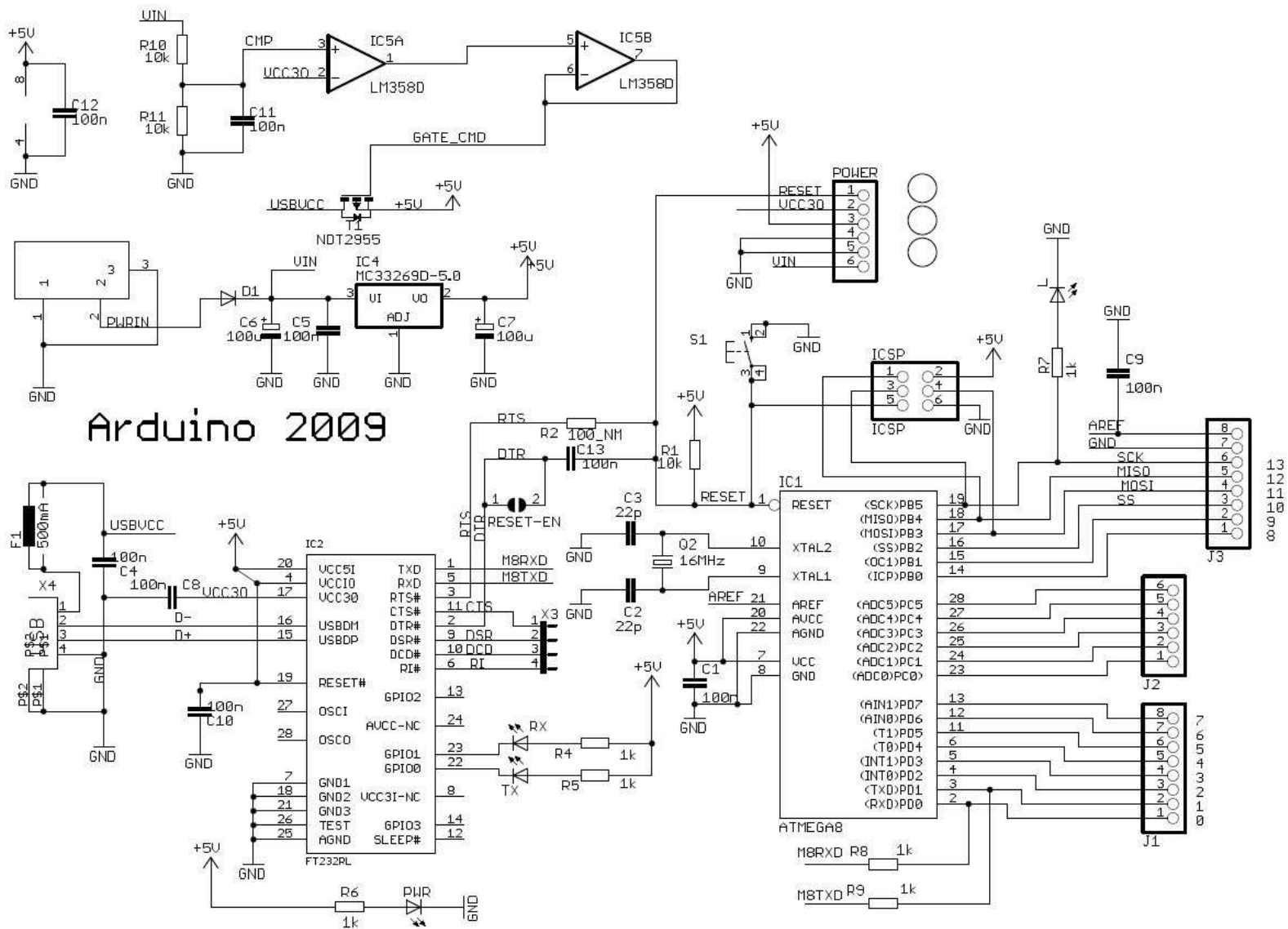
RESET-EN

X3

16.000

AMS 1117 5.0

LM358
YR118



Arduino 2009

ATMEGA8

FT232RL

POWER
1
2
3
4
5
6

ICSP
1
2
3
4
5
6

J3
8
7
6
5
4
3
2
1
13
12
11
10
9
8

J2
6
5
4
3
2
1

J1
8
7
6
5
4
3
2
1
7
6
5
4
3
2
1
0

RESET
XTAL2
XTAL1
AREF
AUC
AGND
UCC
GND

<SCK>PB5
<MISO>PB4
<MOSI>PB3
<SS>PB2
<OC1>PB1
<ICP>PB0

19
18
17
16
15
14
28
27
26
25
24
23

<AIN1>PD7
<AIN0>PD6
<T1>PD5
<T0>PD4
<INT1>PD3
<INT0>PD2
<TXD>PD1
<RXD>PD0

13
12
11
6
5
4
3
2
1

M8RXD R8
M8TXD R9

1k
1k

Structure
void **setup()** void **loop()**

Control Structures
if (x<5) { } else { }
switch (myvar) {
 case 1:
 break;
 case 2:
 break;
 default:
 }
for (int i=0; i<= 255; i++){ }
while (x<5) { }
do { } while (x<5);
continue; //Go to next in do/for/while loop
return x; // Or 'return;' for voids.
goto // considered harmful :-)

Further Syntax
// (single line comment)
/* (multi-line comment) */
#define DOZEN 12 //Not baker's!
#include <avr/pgmspace.h>

General Operators
= (assignment operator)
+ (addition) - (subtraction)
* (multiplication) / (division)
% (modulo)
== (equal to) != (not equal to)
< (less than) > (greater than)
<= (less than or equal to)
>= (greater than or equal to)
&& (and) || (or) ! (not)

Pointer Access
& reference operator
* dereference operator

Bitwise Operators
& (bitwise and) | (bitwise or)
^ (bitwise xor) ~ (bitwise not)
<< (bitshift left) >> (bitshift right)

Compound Operators
++ (increment) -- (decrement)
+= (compound addition)
-= (compound subtraction)
*= (compound multiplication)
/= (compound division)
&= (compound bitwise and)
|= (compound bitwise or)

Constants
HIGH | LOW
INPUT | OUTPUT
true | false
143 // **Decimal** number
0173 // **Octal** number
0b11011111 // **Binary**
0x7B // **Hex** number
7U // Force unsigned
10L // Force long
15UL // Force long unsigned
10.0 // Forces floating point
2.4e5 // 240000

Data Types
void
boolean (0, 1, false, true)
char (e.g. 'a' -128 to 127)
unsigned char (0 to 255)
byte (0 to 255)
int (-32,768 to 32,767)
unsigned int (0 to 65535)
word (0 to 65535)
long (-2,147,483,648 to 2,147,483,647)
unsigned long (0 to 4,294,967,295)
float (-3.4028235E+38 to 3.4028235E+38)
double (currently same as float)
sizeof(myint) // returns 2 bytes

Strings
char S1[15];
char S2[8]={'a','y','d','u','t','n','o'};
char S3[8]={'a','y','d','u','t','n','o','0'};
//Included '0' null termination
char S4[] = "arduino";
char S5[8] = "arduino";
char S6[15] = "arduino";

Arrays
int myInts[6];
int myPins[] = {2, 4, 8, 3, 6};
int mySensVals[6] = {2, 4, -8, 3, 2};

Conversion
char() byte()
int() word()
long() float()

ARDUINO CHEAT SHEET V.02c

Mostly taken from the extended reference:
<http://arduino.cc/en/Reference/Extended>
Gavin Smith – Robots and Dinosaurs, The Sydney Hackspace



Qualifiers
static // persists between calls
volatile // use RAM (nice for ISR)
const // make read-only
PROGMEM // use flash

Digital I/O
pinMode(pin, [INPUT,OUTPUT])
digitalWrite(pin, value)
int digitalRead(pin)
//Write High to inputs to use pull-up res

Analog I/O
analogReference([DEFAULT,INTERNAL,EXTERNAL])
int analogRead(pin) //Call twice if switching pins from high Z source.
analogWrite(pin, value) // PWM

Advanced I/O
tone(pin, freqhz)
tone(pin, freqhz, duration_ms)
noTone(pin)
shiftOut(dataPin, clockPin, [MSBFIRST,LSBFIRST], value)
unsigned long pulseIn(pin, [HIGH,LOW])

Time
unsigned long millis() // 50 days overflow.
unsigned long micros() // 70 min overflow
delay(ms)
delayMicroseconds(us)

Math
min(x, y) max(x, y) abs(x)
constrain(x, minval, maxval)
map(val, fromL, fromH, toL, toH)
pow(base, exponent) sqrt(x)
sin(rad) cos(rad) tan(rad)

Random Numbers
randomSeed(seed) // Long or int
long random(max)
long random(min, max)

Bits and Bytes
lowByte() highByte()
bitRead(x,bit) bitWrite(x,bit,bit)
bitSet(x,bit) bitClear(x,bit)
bit(bit) //bit: 0-LSB 7-MSB

External Interrupts
attachInterrupt(interrupt, function, [LOW,CHANGE,ISING,FALLING])
detachInterrupt(interrupt)
interrupts()
noInterrupts()

Libraries:

Serial.
begin([300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200])
end()
int available()
int read()
flush()
print()
println()
write()

EEPROM (#include <EEPROM.h>)
byte read(intAddr)
write(intAddr,myByte)

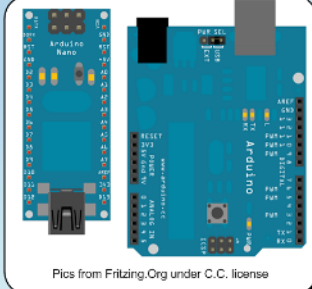
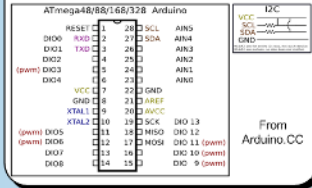
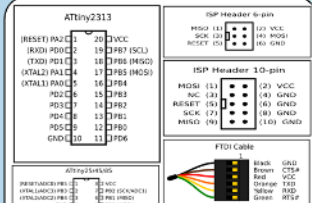
Servo (#include <Servo.h>)
attach(pin, [min_uS, max_uS])
write(angle) // 0-180
writeMicroseconds(uS) //1000-2000, 1500 is midpoint
read() // 0-180
attached() //Returns boolean
detach()

SoftwareSerial(RxPin,TxPin)
// #include<SoftwareSerial.h>
begin(longSpeed) // up to 9600
char read() // blocks till data
print(myData) or println(myData)

Wire (#include <Wire.h>) // For I2C
begin() // Join as master
begin(addr) // Join as slave @ addr
requestFrom(address, count)
beginTransmission(addr) // Step 1
send(mybyte) // Step 2
send(char * mystring)
send(byte * data, size)
endTransmission() // Step 3
byte available() // Num of bytes
byte receive() //Return next byte
onReceive(handler)
onRequest(handler)

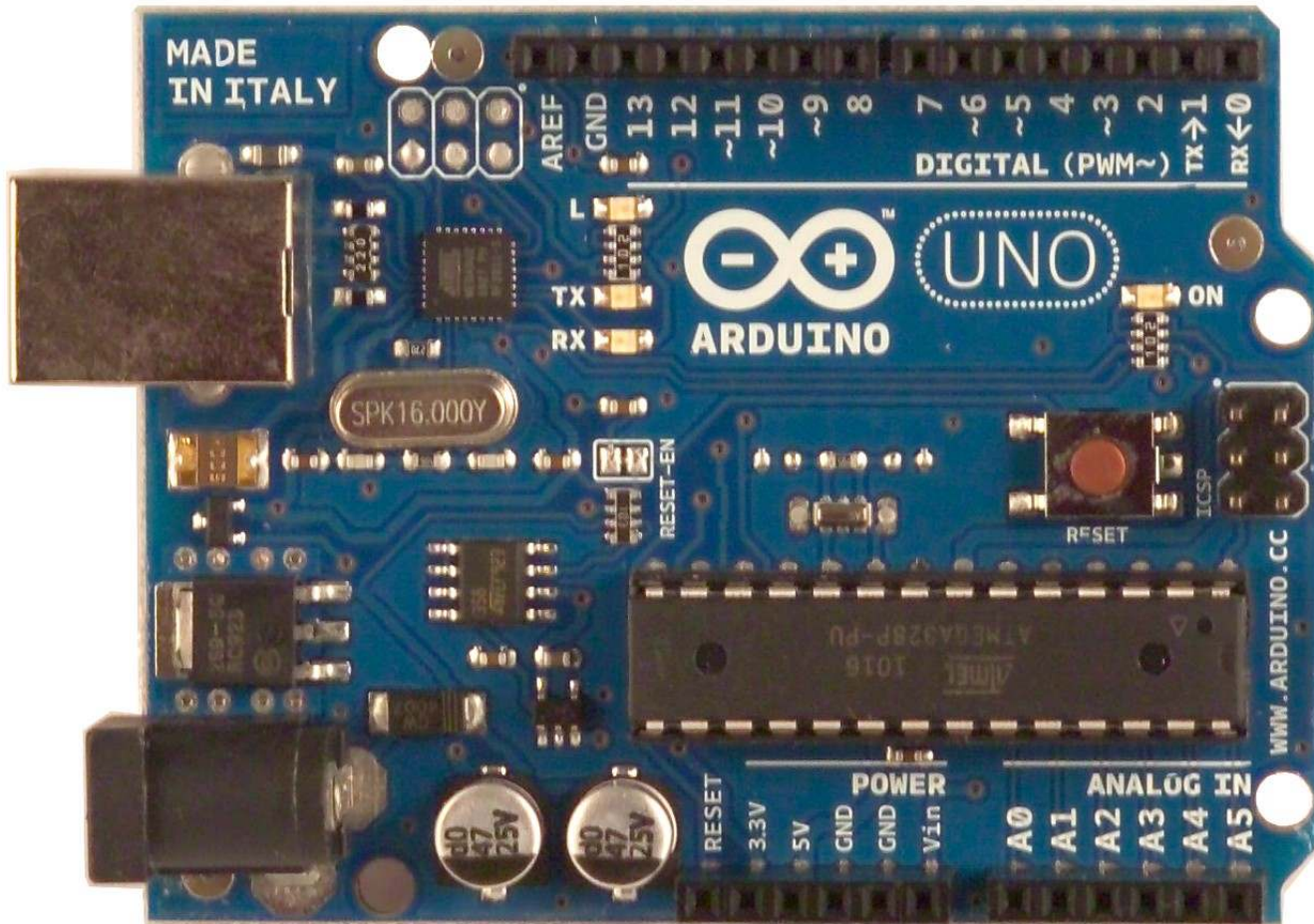
	ATmega168	ATMega328	ATmega1280
= 12k for bootloader	16KB	32KB	128KB
SRAM	1KB	2KB	8KB
EEPROM	512B	1KB	4KB

	Duemilanove/ Nano/ Pro Mini	Mega
V of I/O	14 + 8 Analog (Nano has 14 I/O)	54 + 16 Analog
Serial Pins	0 - RX 1 - TX	0 - RX1 - TX1 16 - RX2 - TX2 17 - RX3 - TX3 18 - RX4 - TX4
Ext Interrupts	2 - int0; 3 - int1;	0,3,21,20,19,18 (RISC ICs)
PWM pins	5, 6 - Timer 0 9, 10 - Timer 1 11, 12 - Timer 2	0-13
SPI	10 - SS 11 - MISO 12 - MISC 13 - SCK	00 - SS 51 - MISO 50 - MISO 52 - SCK
I2C	Analog9 - SDA Analog8 - SCL	21 - SDA 20 - SCL

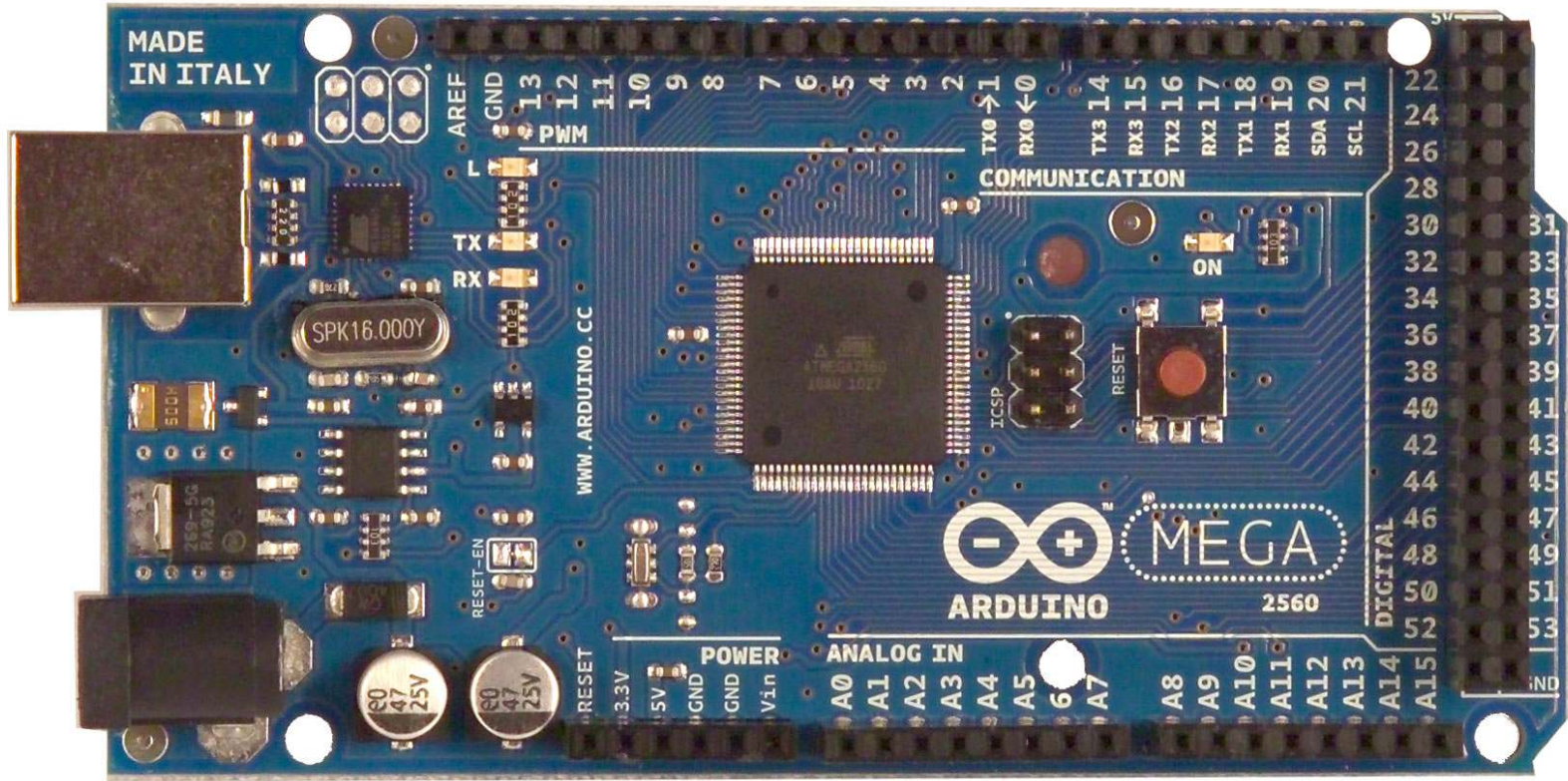


Pics from Fritzing.Org under C.C. license

Arduino Uno



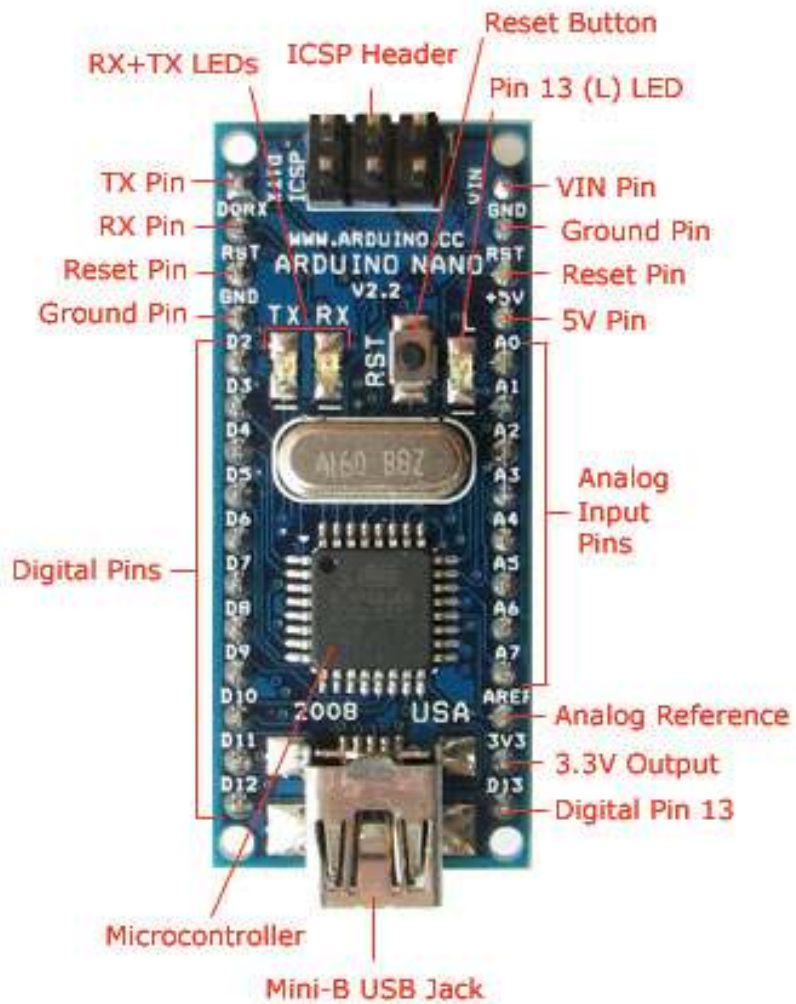
Arduino Mega 2560



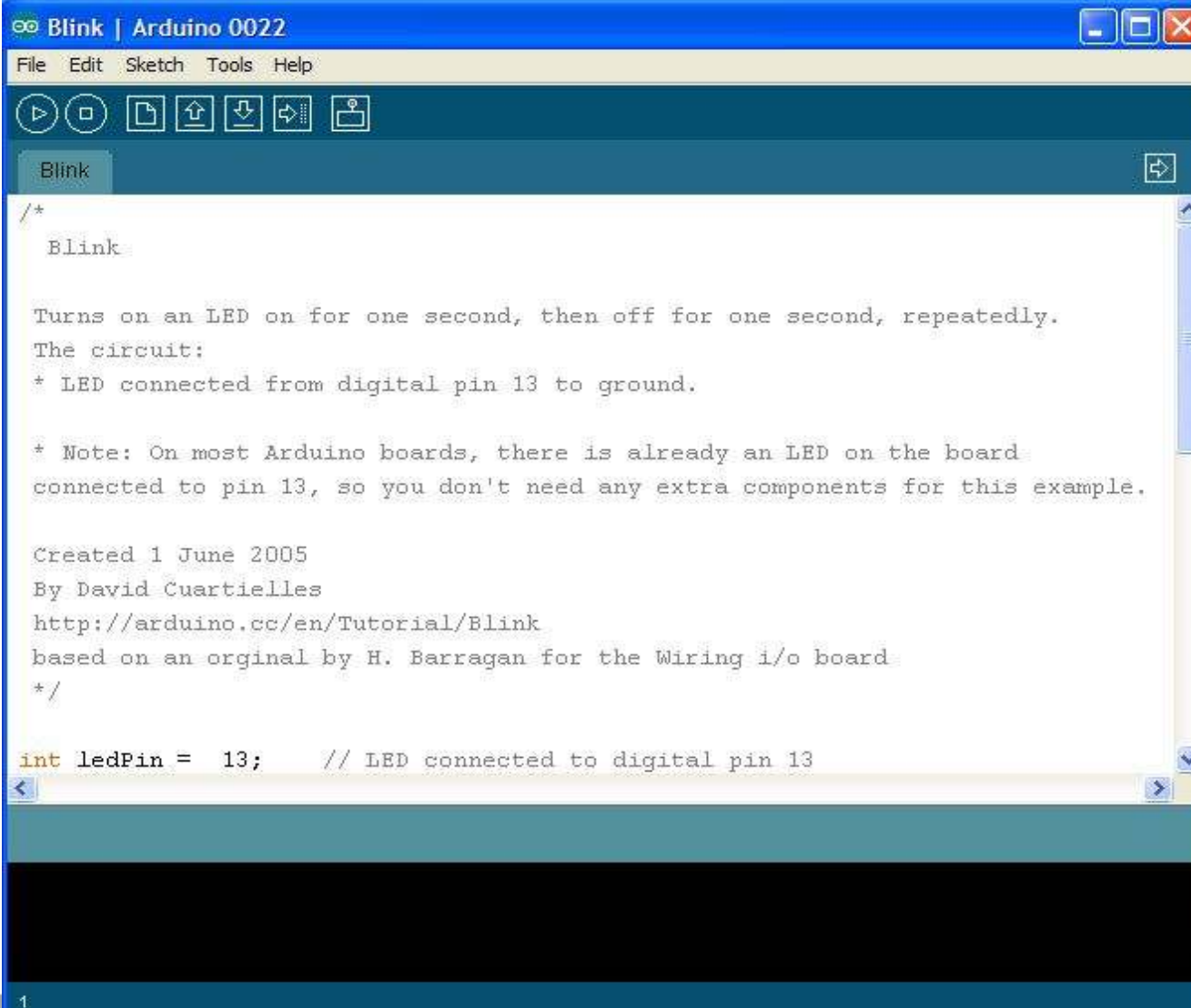
Original Arduino with RS-232



Arduino Nano



Arduino IDE



The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 0022". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, stopping, saving, and other functions. The main editor area shows the "Blink" sketch with the following content:

```
/*  
  Blink  
  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  The circuit:  
  * LED connected from digital pin 13 to ground.  
  
  * Note: On most Arduino boards, there is already an LED on the board  
  connected to pin 13, so you don't need any extra components for this example.  
  
  Created 1 June 2005  
  By David Cuartielles  
  http://arduino.cc/en/Tutorial/Blink  
  based on an original by H. Barragan for the Wiring i/o board  
*/  
  
int ledPin = 13;    // LED connected to digital pin 13
```

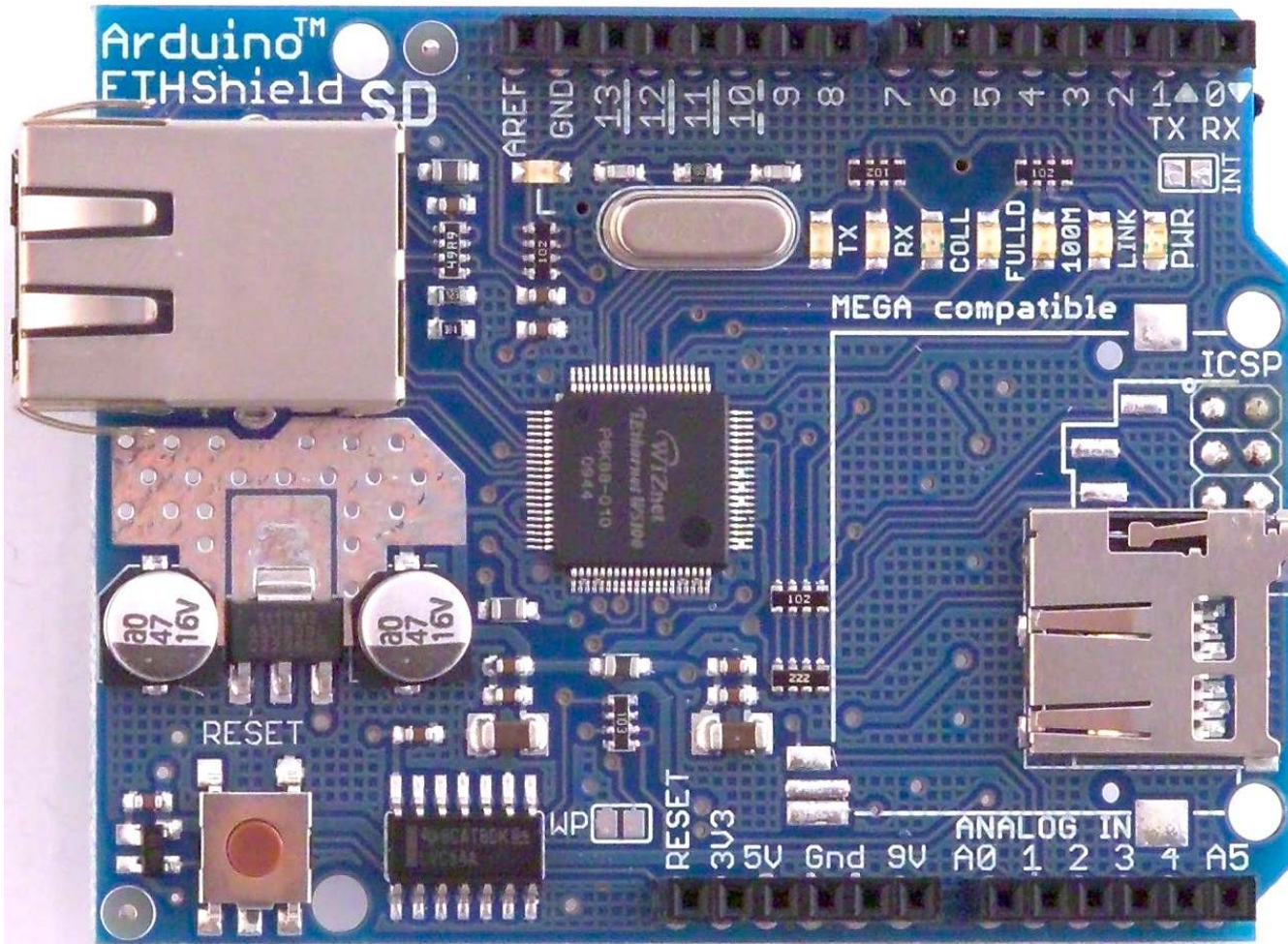

Arduino Language

- C like syntax, but simplified
- Abstracts the pin naming to numbers
- Trades efficiency for ease of use
- Easy to learn, yet powerful
- Lots of example code
- Easy to reuse C-code from other projects
- Libraries can be written in C++
- Lots of libraries available

Hallo World Example

```
int ledPin = 13; // LED connected to digital pin 13
// The setup() method runs once, when the sketch starts
void setup() {
    // initialize the digital pin as an output:
    pinMode(ledPin, OUTPUT);
}
// the loop() method runs over and over again, as long as the Arduino has power
void loop()
{
    digitalWrite(ledPin, HIGH); // set the LED on
    delay(500); // wait for half a second
    digitalWrite(ledPin, LOW); // set the LED off
    delay(500);
} // wait for half a second
```

Ethernet shield w/micro-SD reader



Resources

- www.atmel.com/avr
- www.avrfreaks.net
- www.arduino.cc
- en.wikipedia.org/wiki/Arduino
- www.mcselec.com (BascomAVR)
- www.argentdata.com
- www.ebay.com
- www.sparkfun.com