ARDUINO FOR SOFTWARE HACKERS

Or How I stopped worring and Love the Hardware

GONZALO MALDONADO

WHO IS? GONZALO

twitter.COM/ELG0NZ

github.COM/ELG0NZ
SOCIAL CODING

TWIT THIS!

#BARCAMPMEXICO

#ARDUINOMX

HARDWARE FAIL!

HARDWARE FAIL!

ARDUINO
IS A PLATFORM

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino programming language (based on Wiring) + Arduino development environment (based on Processing).

ARDUINO
IS A MICROCONTROLLER

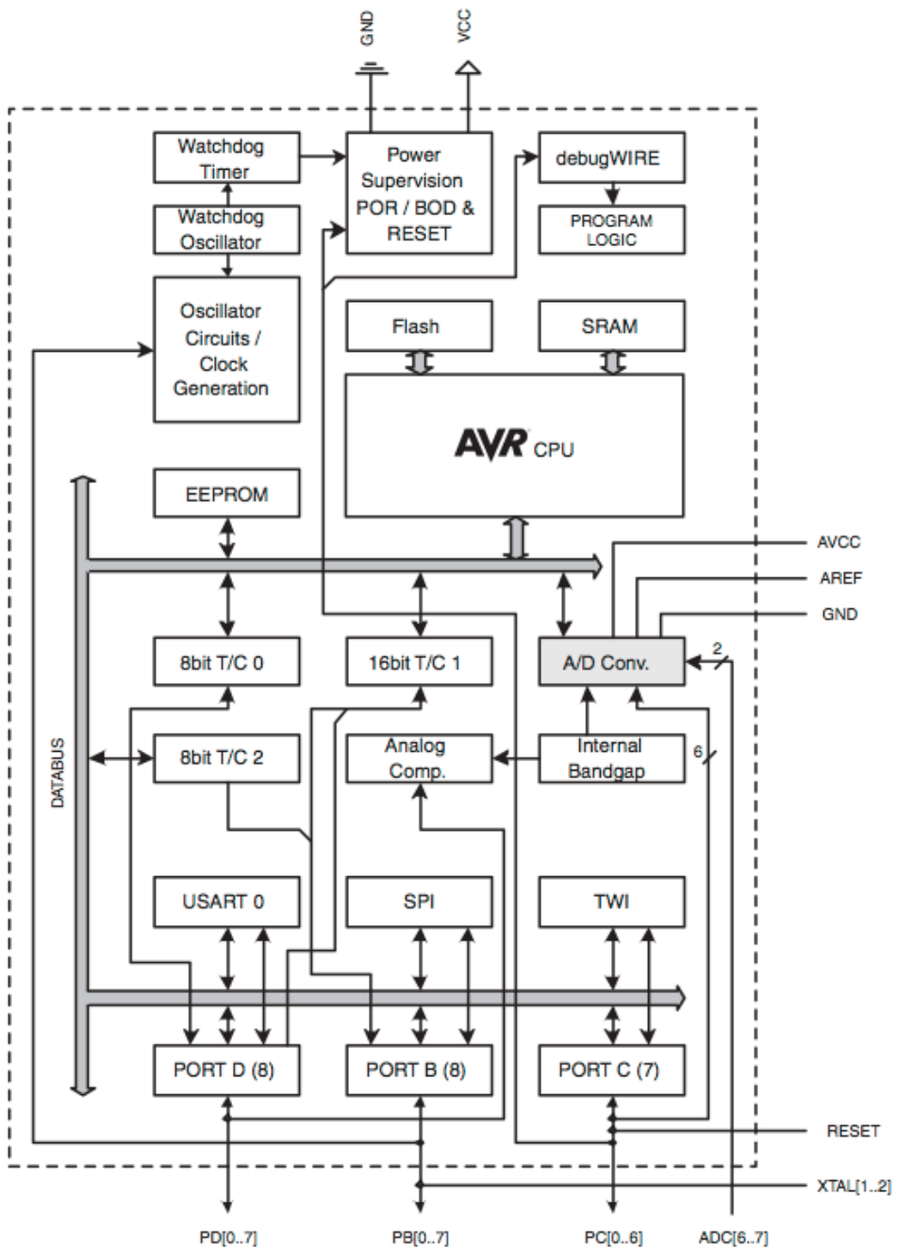
Arduino boards are based on the Atmel
ATmega168

Risc 8 bits
131 Instrucions
32 x 8 registers
512 Bytes EEPROM
1Kbyte SRAM
In Circuit
Programming

16 Kbytes Flash
2 8bit Counters
1 16 bit Counter
6 PWMs
6 10bit ADCs
USART + SPI + I2C

ARDUINO

WTF?



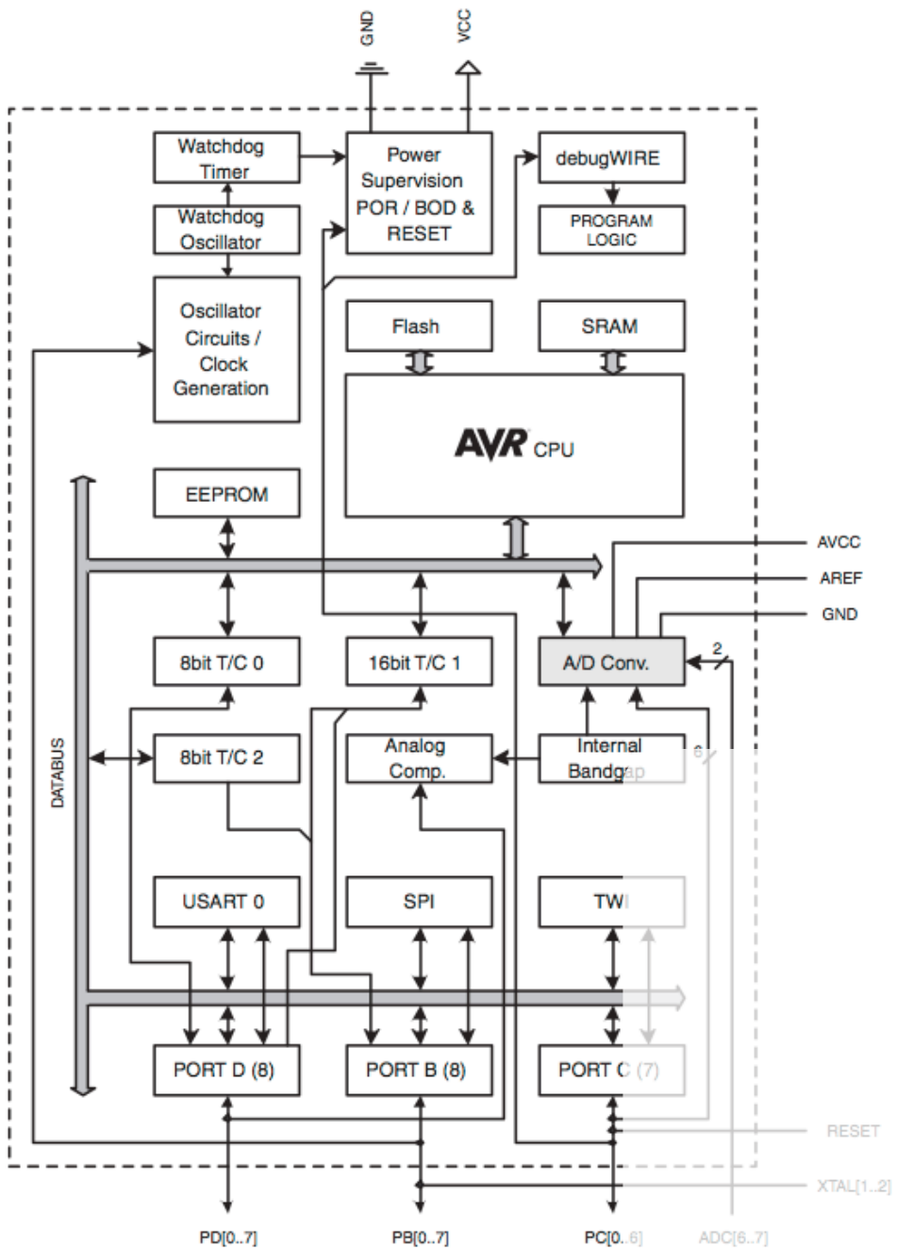
=

Intel 286

http://www.slideshare.net/eoinbrazil/imediaarduino08

ARDUINO

WTF?



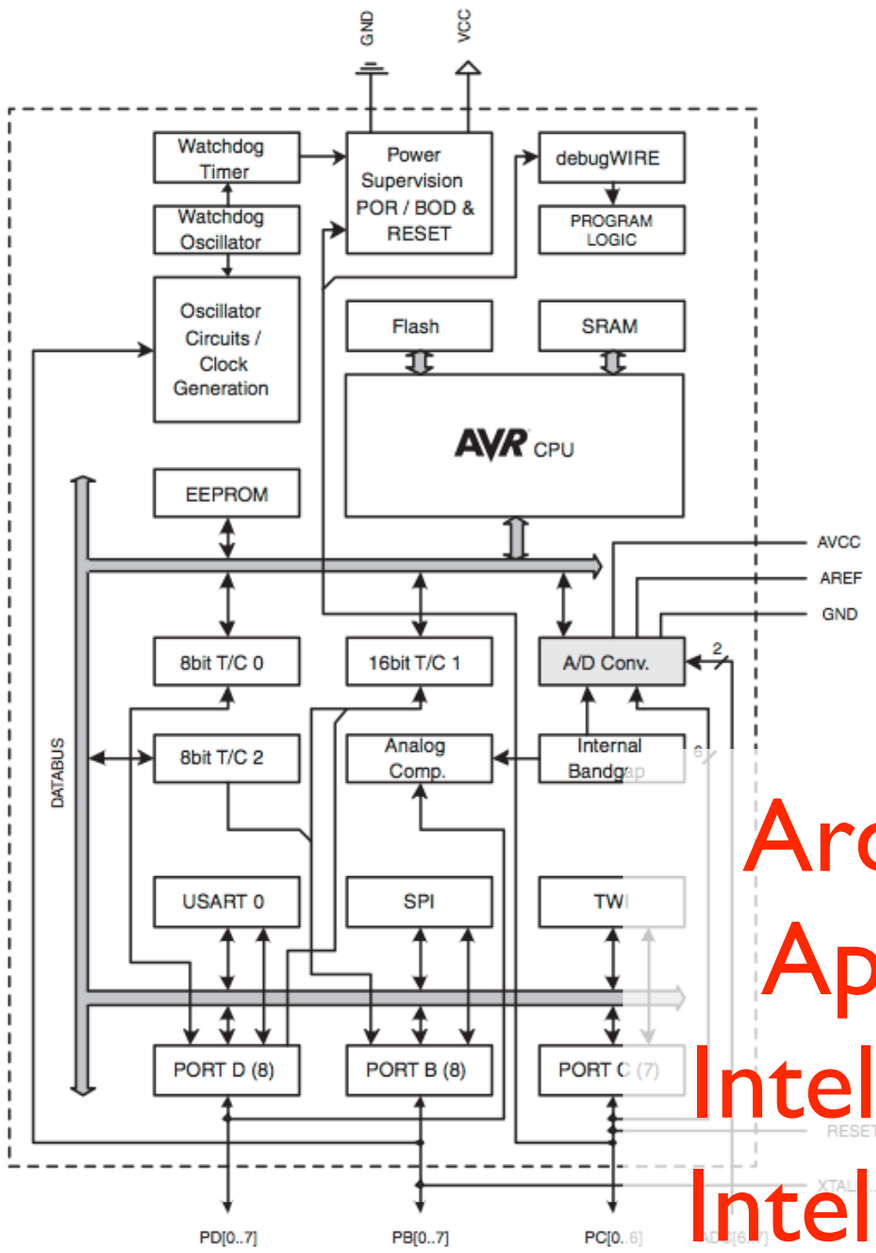
=

Intel 286

http://www.slideshare.net/eoinbrazil/imediaarduino08

ARDUINO

WTF?



Intel 286

Arduino 16 Mhz.

Apple II 1 Mhz

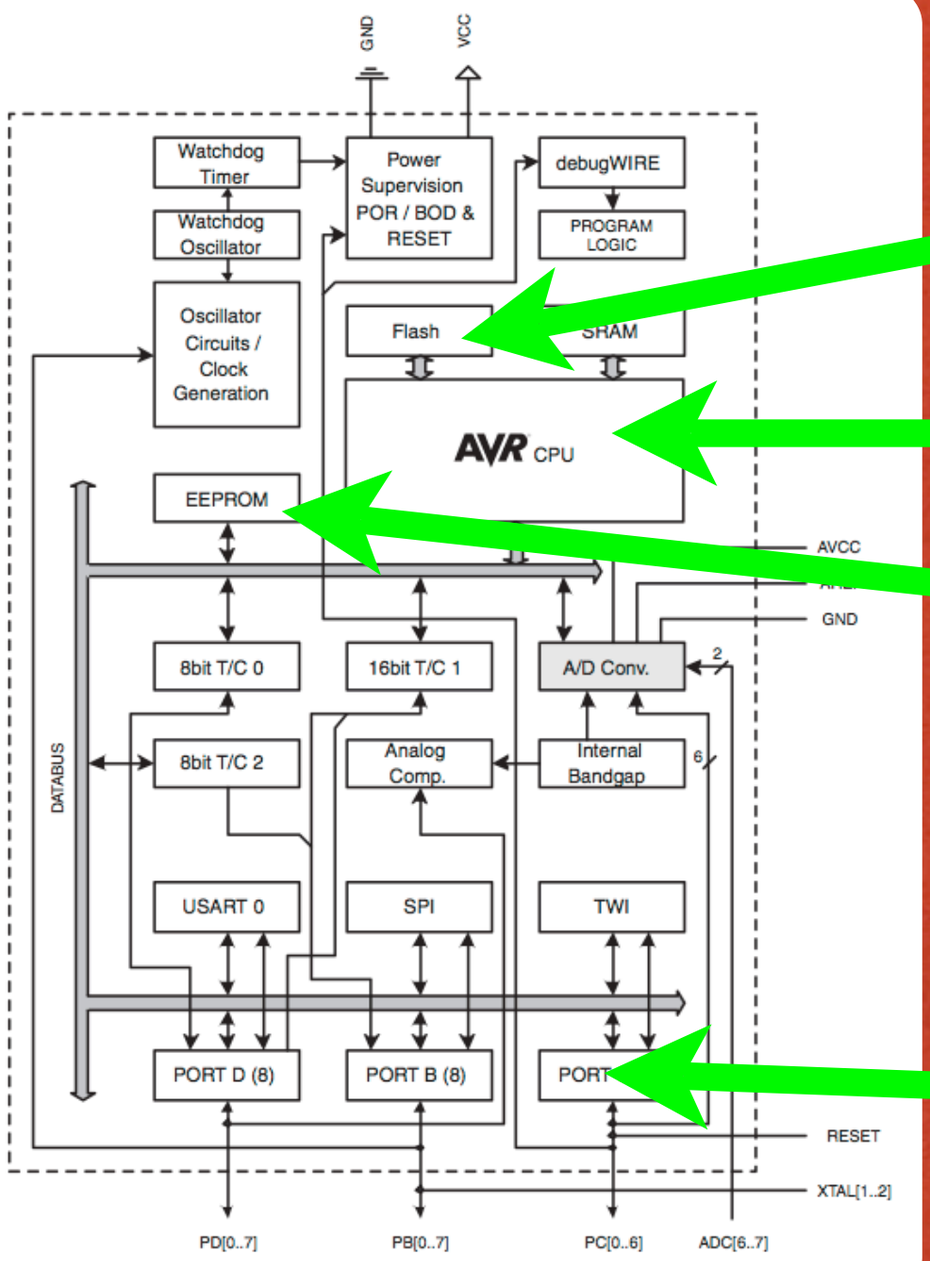
Intel 286 12.5 Mhz

Intel Core 2.3 Ghz

http://www.slideshare.net/eoinbrazil/imediaarduino08

ARDUINO

WTF?



RAM

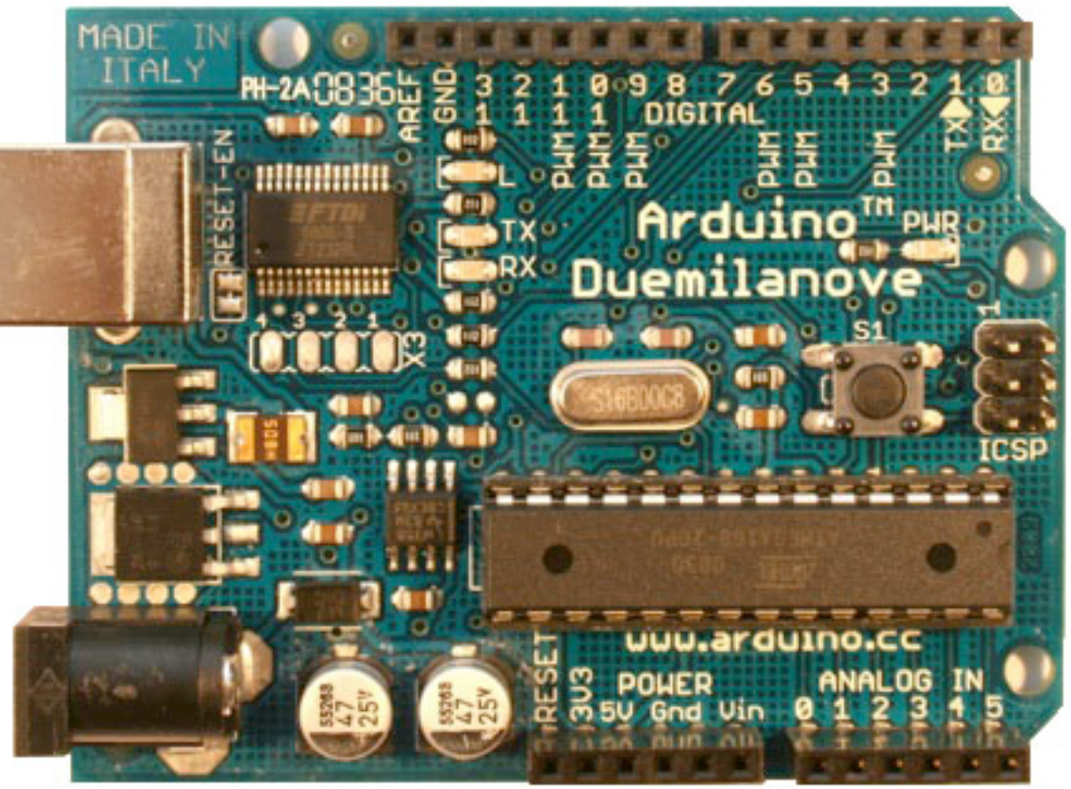
CPU

EEPROM

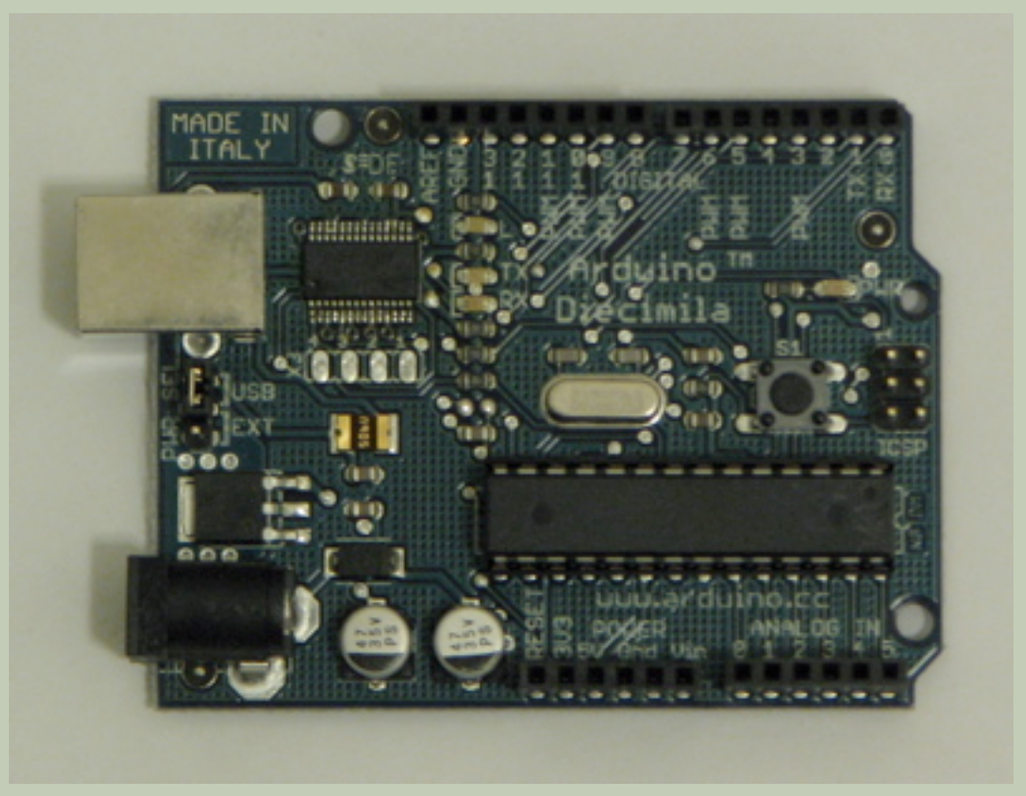
I/O

ARDUINO

COMES IN FLAVORS!



New! Standard Flavor

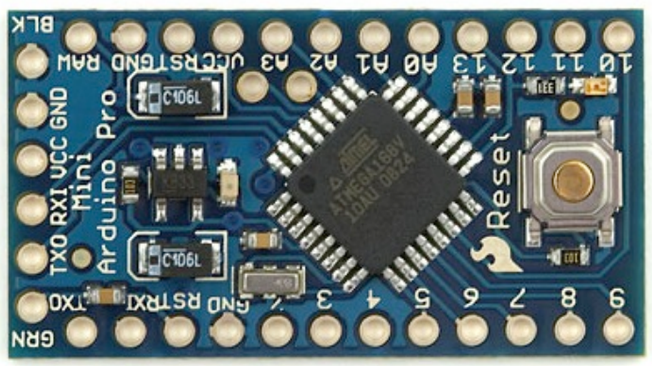


Classic Flavor

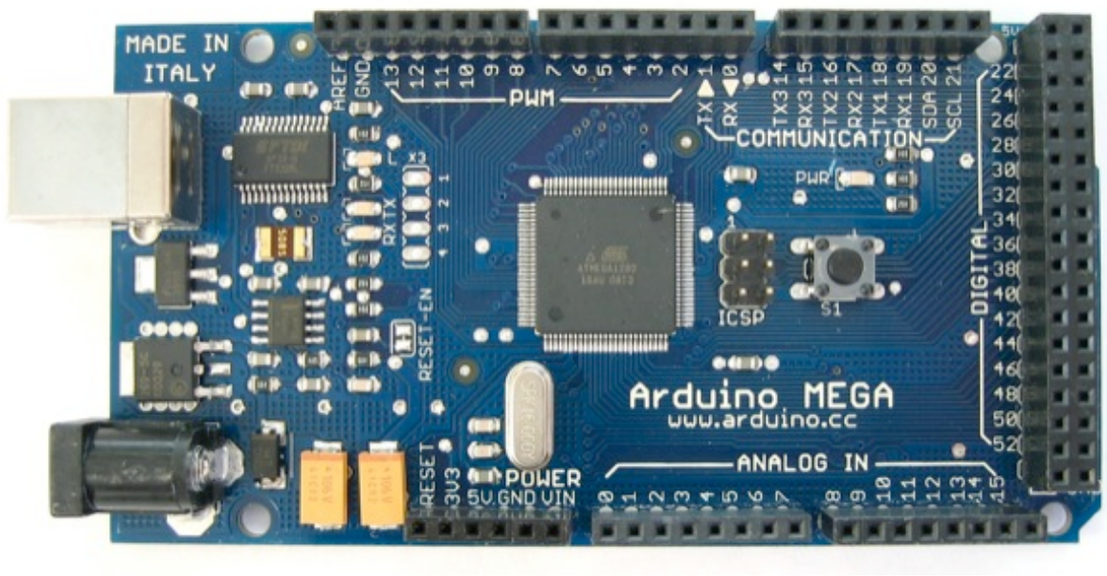
http://arduino.cc/en/Main/Hardware

ARDUINO

COMES IN FLAVORS!



New and Tiny!

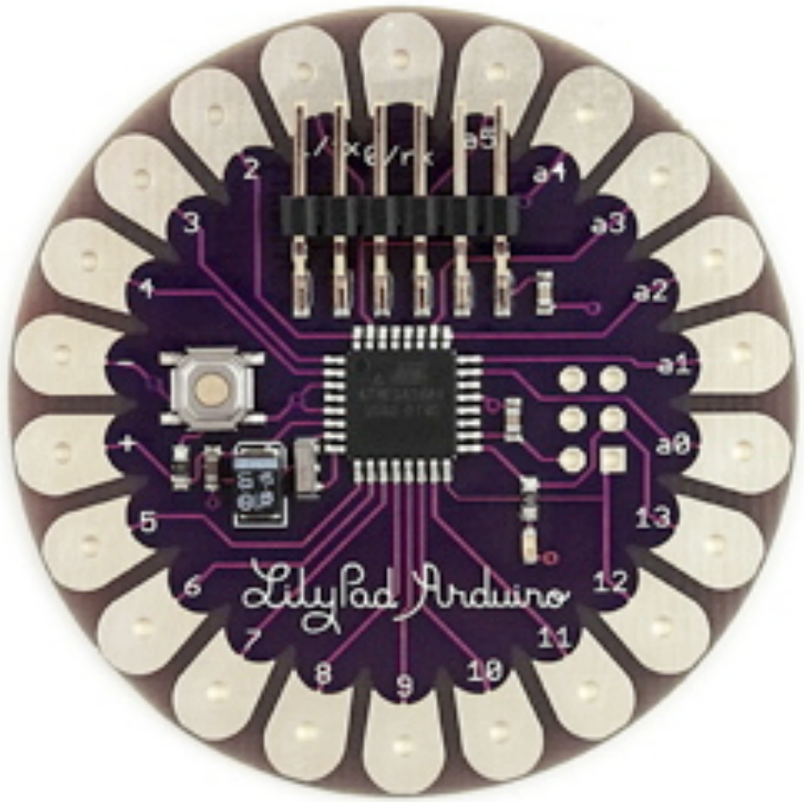


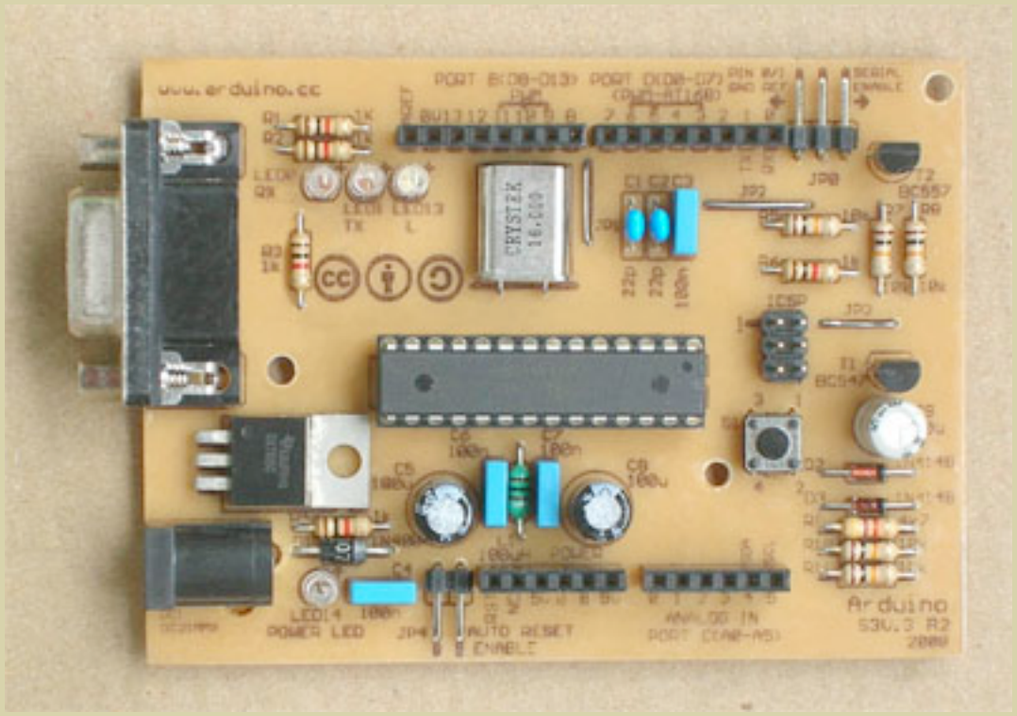
Mega Size me!

http://arduino.cc/en/Main/Hardware

ARDUINO

COMES IN FLAVORS!





Floral and Purple!

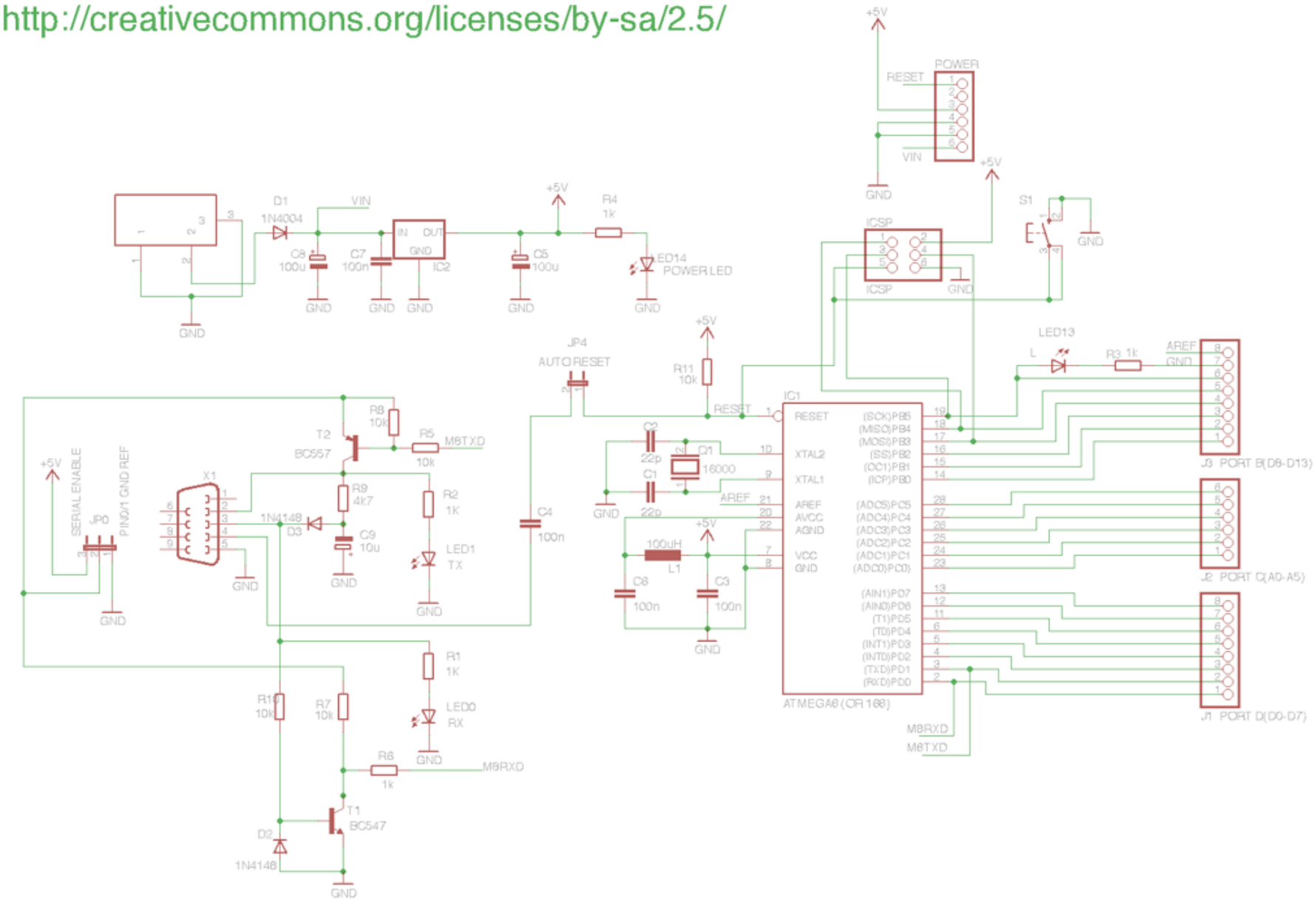
Manly DIY!

http://arduino.cc/en/Main/Hardware

ARDUINO

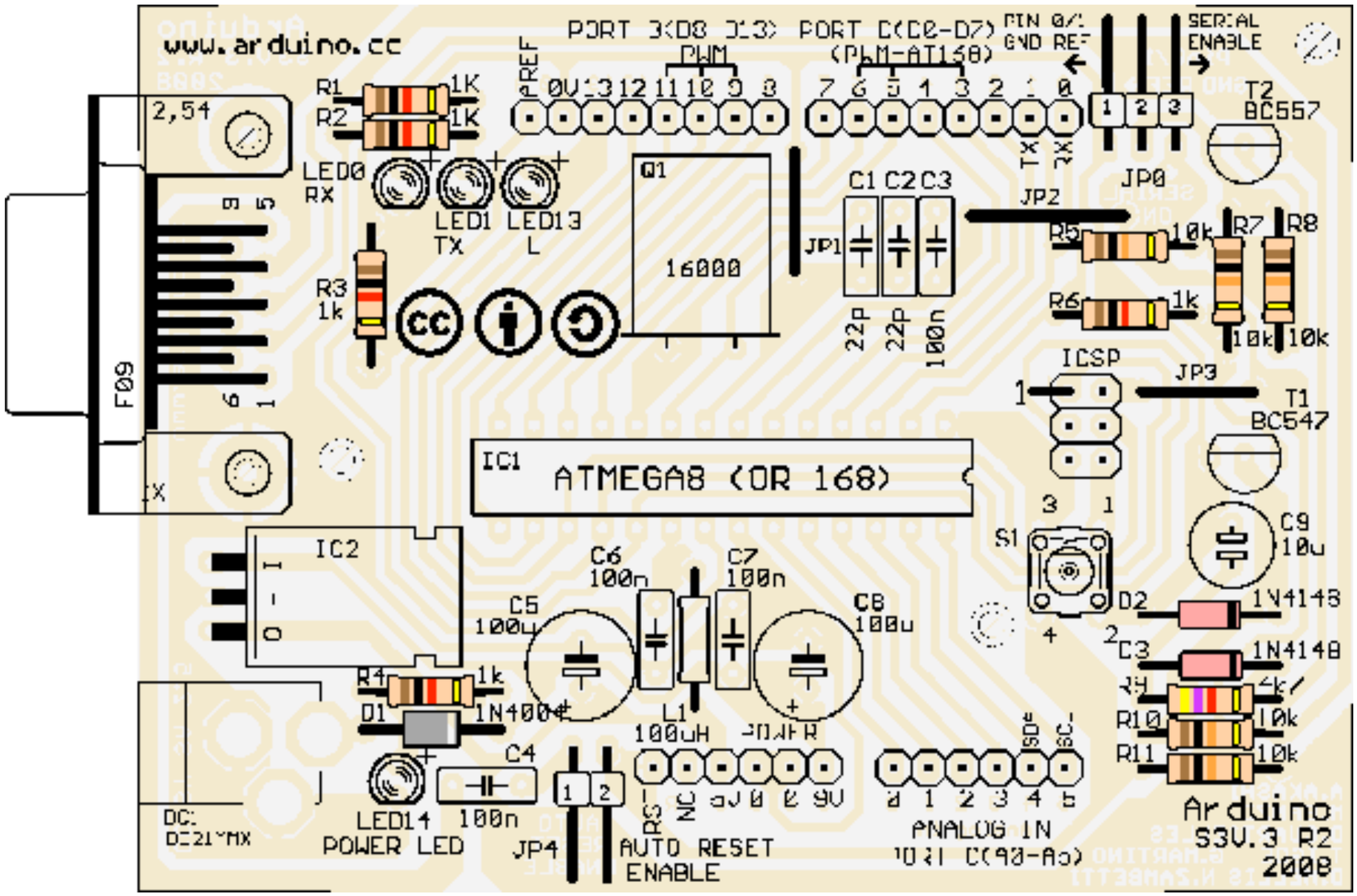
DIY

Arduino S3v3 Revision 2



ARDUINO

DIY



http://arduino.cc/en/Main/Hardware

ARDUINO DIY

BILL OF MATERIAL FOR ARDUINO SERIAL SINGLE SIDED VERSION 3 (S3V3) - REVISION 2					
QTY	POSITION	DESCRIPTION	VALUE	DETAIL	
2	C1, C2	ceramic disc capacitor	22pF (22 pico Farad)		
4	C3, C4, C6, C7	ceramic or polyester capacitor	100nF (100 nano Farad - or 0.1 micro Farad)		
2	C5, C8	electrolytic capacitor	100µF (100 micro Farad)	16volts (or more: 25v)	radial-lead
1	C9	non-polarized electrolytic capacitor	10µF (10 micro Farad)	16volts (or more: 25v, 50v)	radial-lead
1	D1	diode	1N4004	DO41-10	
2	D2, D3	diode	1N4148	DO35-10	
1	DC1	2.1mm. DC power jack			
1	IC1	ATMEGA8 (or ATMEGA168)		28P3 package	
1	IC2	Tension Regulator	7805C		
1	ICSP	male pin header	2x3		
2	J1, J3	female pin header	1x8	0.1" (or 2.54 mm.)	
2	J2, POWER	female pin header	1x6	0.1" (or 2.54 mm.)	
1	JP0	right angle pin header	1x3	0.1" (or 2.54 mm.)	
1	JP4	right angle pin header	1x2	0.1" (or 2.54 mm.)	
1	L1	leaded inductor	100µH (100 micro Henry)	axial leaded	(silver)brown, black, brown, golden
4	LED0, LED1, LED13, LED14	LED	3 mm.	choose colors	
1	Q1	16 MHz crystal			
5	R1, R2, R3, R4, R6	Resistor	1kohm (1.0 kilo ohms)	1/4 Watt, ±5%	brown, black, red, gold
1	R9	Resistor	4k7ohms (4.7 kilo ohms)	1/4 Watt, ±5%	yellow, violet, red, gold
5	R5, R7, R8, R10, R11	Resistor	10kohms (10.0 kilo ohms)	1/4 Watt, ±5%	brown, black, orange, gold
1	S1	Switch Tactile	6x6 mm., 4 terminals		B3F-10XX
1	T1	Transistor	BC547	NPN general purpose transistor	TO92
1	T2	Transistor	BC557	PNP general purpose transistor	TO92
1	X1	D-SUB CONNECTOR	9 PIN FEMALE RIGHT ANGLE PC MOUNT	DE-9 CONNECTOR	
2	Jumpers	jumper for 0.1" header		0.1" (or 2.54 mm.)	

ARDUINO DIY

PART LIST FOR ARDUINO SERIAL SINGLE SIDED VERSION 3 (S3V3) - REVISION 2				
POSITION	VALUE	DESCRIPTION	DETAIL	
C1	22pF (22 pico Farad)	ceramic disc capacitor		
C2	22pF (22 pico Farad)	ceramic disc capacitor		
C3	100nF (100 nano Farad - or 0.1 micro Farad)	ceramic or polyester capacitor		
C4	100nF (100 nano Farad - or 0.1 micro Farad)	ceramic or polyester capacitor		
C5	100µF (100 micro Farad)	electrolytic capacitor	16volts (or more: 25v)	radial-lead
C6	100nF (100 nano Farad - or 0.1 micro Farad)	ceramic or polyester capacitor		
C7	100nF (100 nano Farad - or 0.1 micro Farad)	ceramic or polyester capacitor		radial-lead
C8	100µF (100 micro Farad)	electrolytic capacitor	16volts (or more: 25v)	radial-lead
C9	10µF (10 micro Farad)	non-polarized electrolytic capacitor	16volts (or more: 25v, 50v)	radial-lead
D1	1N4004	diode	DO41-10	
D2	1N4148	diode	DO35-10	
D3	1N4148	diode	DO35-10	
DC1		2.1mm. DC power jack		
IC1		ATMEGA8 (or ATMEGA168)	28P3 package	
IC2	7805C	Tension Regulator		
ICSP	2x3	male pin header		ICSP
J1	1x8	female pin header	0.1" (or 2.54 mm.)	PORT D(D0-D7)
J2	1x6	female pin header	0.1" (or 2.54 mm.)	PORT C(A0-A5)
J3	1x8	female pin header	0.1" (or 2.54 mm.)	PORT B(D8-D13)
JP0	1x3	right angle pin header	0.1" (or 2.54 mm.)	
JP4	1x2	right angle pin header	0.1" (or 2.54 mm.)	AUTO RESET
L1	100µH	leaded inductor	axial leaded	(silver)brown, black, brown, golden
LED0	3 mm.	LED	choose a color	Rx Led
LED1	3 mm.	LED	choose a color	Tx Led
LED13	3 mm.	LED	choose a color	Pin13 Led
LED14	3 mm.	LED	choose a color	Power Led
POWER	1x6	female pin header		
Q1		16 MHz crystal		
R1	1kohm (1.0 kilo ohm)	Resistor	1/4 Watt, ±5%	brown, black, red, gold
R2	1kohm (1.0 kilo ohm)	Resistor	1/4 Watt, ±5%	brown, black, red, gold
R3	1kohm (1.0 kilo ohm)	Resistor	1/4 Watt, ±5%	brown, black, red, gold
R4	1kohm (1.0 kilo ohm)	Resistor	1/4 Watt, ±5%	brown, black, red, gold
R5	10kohms (10.0 kilo ohms)	Resistor	1/4 Watt, ±5%	brown, black, orange, gold
R6	1kohm (1.0 kilo ohm)	Resistor	1/4 Watt, ±5%	brown, black, red, gold
R7	10kohms (10.0 kilo ohms)	Resistor	1/4 Watt, ±5%	brown, black, orange, gold
R8	10kohms (10.0 kilo ohms)	Resistor	1/4 Watt, ±5%	brown, black, orange, gold
R9	4k7ohms (4.7 kilo ohms)	Resistor	1/4 Watt, ±5%	yellow, violet, red, gold
R10	10kohms (10.0 kilo ohms)	Resistor	1/4 Watt, ±5%	brown, black, orange, gold
R11	10kohms (10.0 kilo ohms)	Resistor	1/4 Watt, ±5%	brown, black, orange, gold
S1	6x6 mm., 4 terminals	Switch Tactile		B3F-10XX
T1	BC547	Transistor	NPN general purpose transistor	TO92
T2	BC557	Transistor	PNP general purpose transistor	TO92
X1	9 PIN FEMALE RIGHT ANGLE PC MOUNT	D-SUB CONNECTOR	DE-9 CONNECTOR	
jumper		0.1" (or 2.54 mm.)		
jumper		0.1" (or 2.54 mm.)		

HEY BC!

HELLO WORLD

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

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as
output
}

void loop()                     // run over and over again
{
  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
}
```

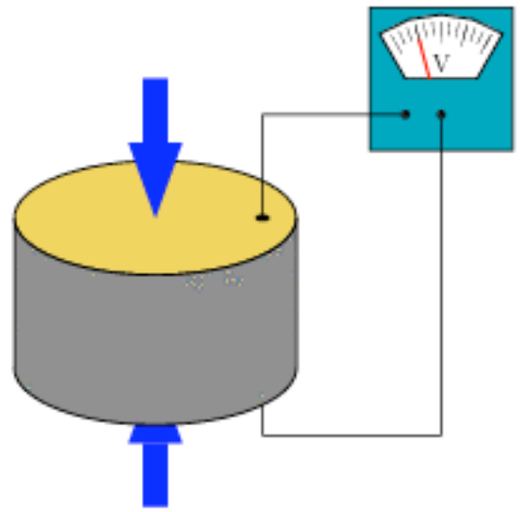
V=RI

OHMM

GRAPH.C

```
void setup()                    // run once, when the sketch starts
{
  Serial.begin(9600);
}

void loop()                     // run over and over again
{
  Serial.println(analogRead(0));
  delay(100);
}
```



GO RETRO

NES

NESTURTLES.C

```
#include <NESpad.h>
NESpad nintendo = NESpad();
.
.
.
void loop() {

  state = nintendo.buttons();

  if (state & NES_A)  Serial.print('a');
  if (state & NES_B)  Serial.print('b');
  if (state & NES_UP)  Serial.print('u');
  if (state & NES_DOWN)  Serial.print('d');
  if (state & NES_LEFT)  Serial.print('l');
  if (state & NES_RIGHT)  Serial.print('r');
  if (state & NES_START)  Serial.print('s');
  //Serial.println(~state, BIN);
  delay(250);
}
```

GO RETRO

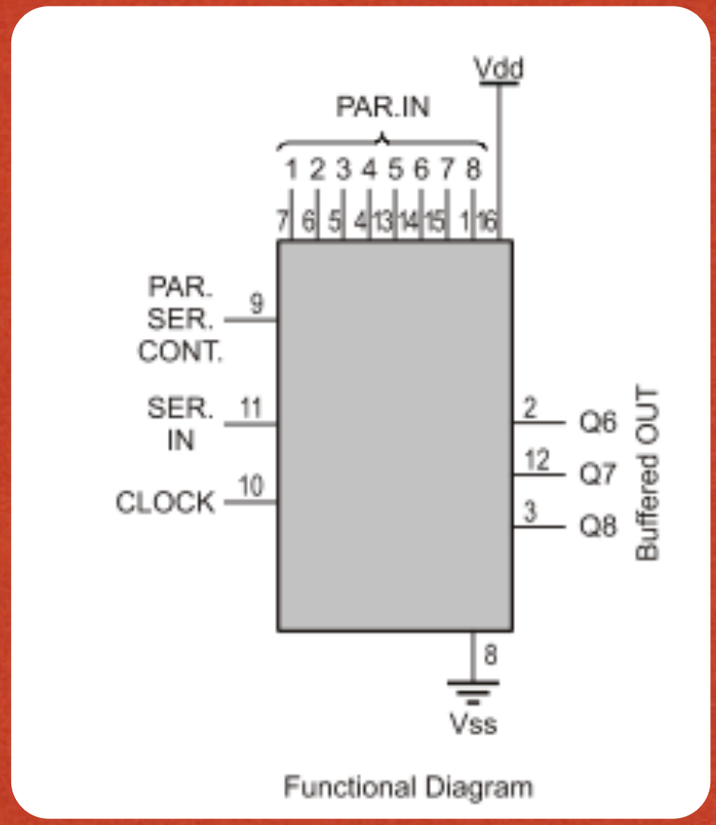
NES

NESTURTLES.C

```
#include <NESpad.h>
NESpad nintendo = NESpad();
.
.
.
void loop() {

  state = nintendo.buttons();

  if (state & NES_A)  Serial.print('a');
  if (state & NES_B)  Serial.print('b');
  if (state & NES_UP)  Serial.print('u');
  if (state & NES_DOWN)  Serial.print('d');
  if (state & NES_LEFT)  Serial.print('l');
  if (state & NES_RIGHT)  Serial.print('r');
  if (state & NES_START)  Serial.print('s');
  //Serial.println(~state, BIN);
  delay(250);
}
```



Functional Diagram

GO RETRO

NES

NESTURTLES.C

GO RETRO

NES

NESTURTLES.C

Operation	Latch	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK
	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 8	bit 9	bit 10	bit 11	bit 12	bit 13	bit 14	bit 15
Data	B	Y	Select	Start	Up	Down	Left	Right	A	X	L	R	1	1	1	1

▲SNES Data Format

Figure 6:7

GO RETRO

NES

NESTURTLES.C

GO RETRO

NES

NESTURTLES.C

```
#define SNES_B          0x01
#define SNES_Y          0x02
#define SNES_SELECT     0x04
#define SNES_START      0x08
#define SNES_UP         0x10
#define SNES_DOWN       0x20
#define SNES_LEFT       0x40
#define SNES_RIGHT      0x80
#define SNES_A          0x100
#define SNES_X          0x200
#define SNES_L          0x400
#define SNES_R          0x800
```

GO RETRO

NES

NESTURTLES.C

```
#define SNES_B          0x01
#define SNES_Y          0x02
#define SNES_SELECT     0x04
#define SNES_START      0x08
#define SNES_UP         0x10
#define SNES_DOWN       0x20
#define SNES_LEFT       0x40
#define SNES_RIGHT      0x80
#define SNES_A          0x100
#define SNES_X          0x200
#define SNES_L          0x400
#define SNES_R          0x800
```

```
  state = 0;
 strobe();
 for (i = 0; i < 16; i++) {
     ret = digitalRead(DATA);
     delayMicroseconds(12);
     digitalWrite(CLOCK,HIGH);
     delayMicroseconds(12);
     digitalWrite(CLOCK,LOW);
    shiftin();
  state |= ret << i;
 }
state =~state;
```

PING!

SONAR

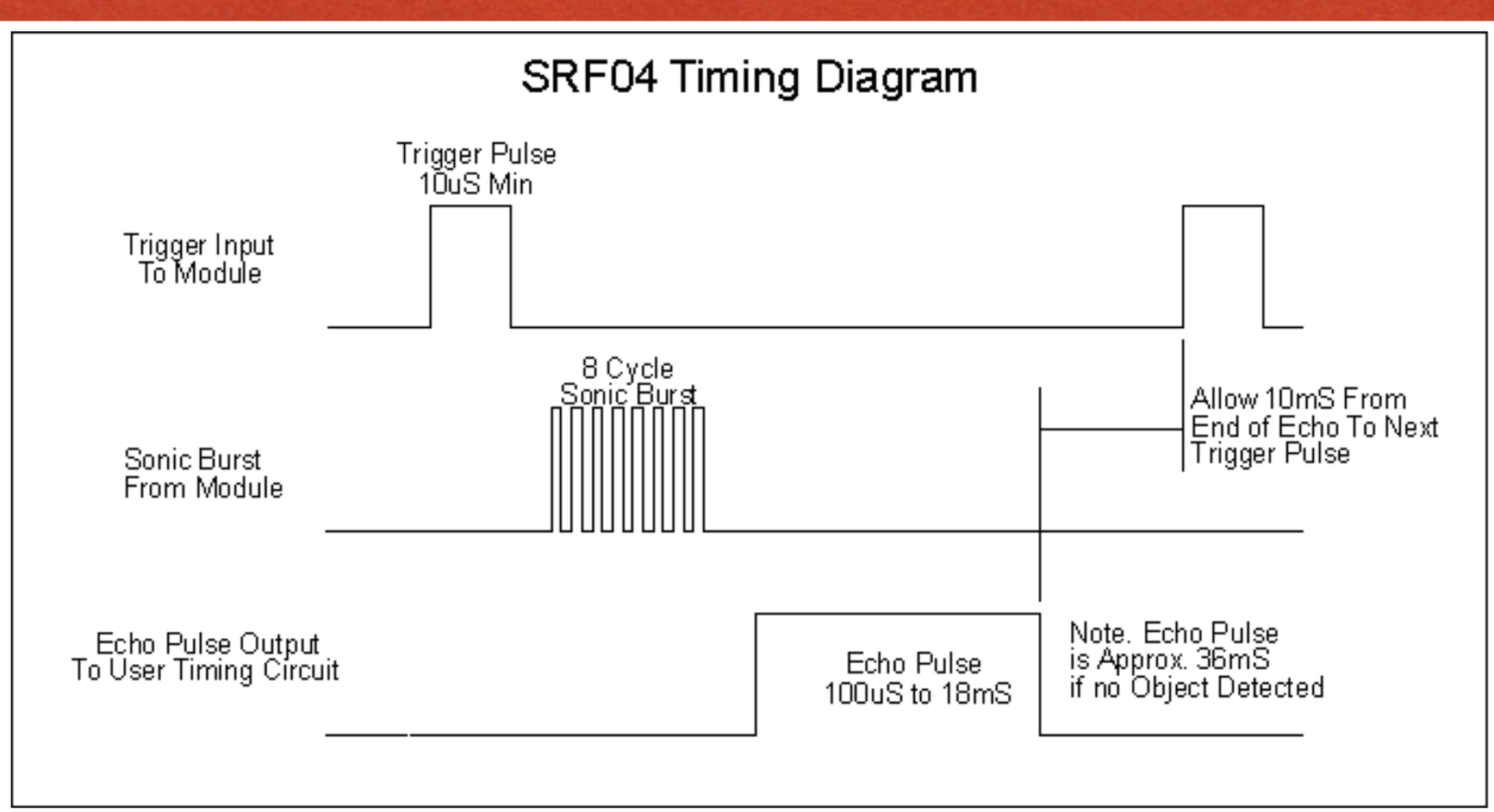
SONAR.C

```
void loop()
{
  for(pos = 0; pos < 180; pos += 1)  // goes from 0 degrees to 180 degrees
  {                                  // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    digitalWrite(initPin, HIGH);
    delay(15);                       // waits 15ms for the servo to reach the position
    digitalWrite(initPin, LOW);
    pulseTime = pulseIn(echoPin, HIGH);
    Serial.println(pulseTime, DEC);
  }
```

PING!

SONAR

SONAR.C

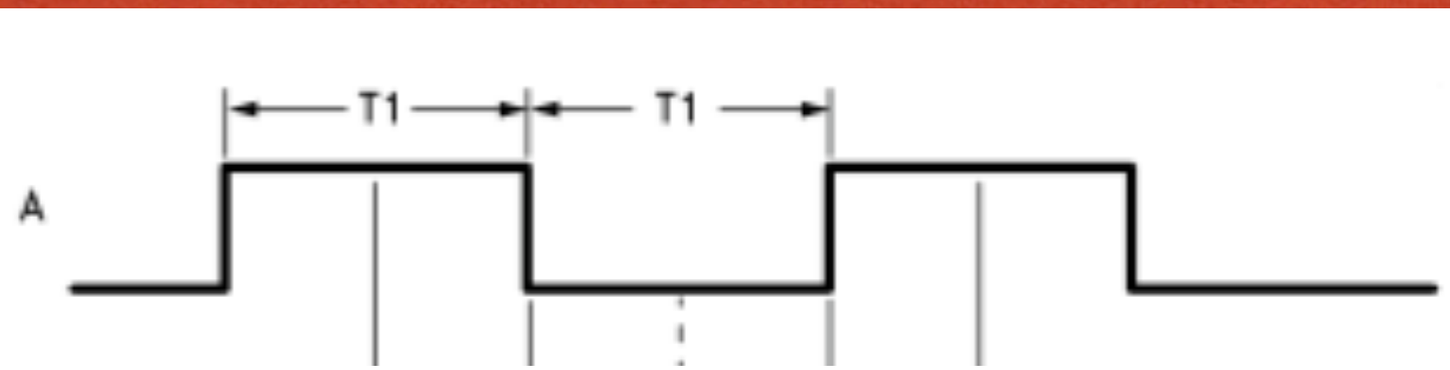


PING!

SONAR

SONAR.C

PWM



PING!

SERVO

```
void Servo::write(int angleArg)
{
  uint16_t p;

  if (angleArg < 0) angleArg = 0;
  if (angleArg > 180) angleArg = 180;
  angle = angleArg;

  // bleh, have to use longs to prevent overflow, could be tricky if always a
16MHz clock, but not true
  // That 8L on the end is the TCNT1 prescaler, it will need to change if the
clock's prescaler changes,
  // but then there will likely be an overflow problem, so it will have to be
handled by a human.
  p = (min16*16L*clockCyclesPerMicrosecond() + (max16-
min16)*(16L*clockCyclesPerMicrosecond())*angle/180L)/8L;
  if (pin == 9) OCR1A = p;
  if (pin == 10) OCR1B = p;
}
```

MORE?

ARDUINO.CC

SPARKFUN.COM

ELG0NZ@GMAIL

(0 ES UN CERO)