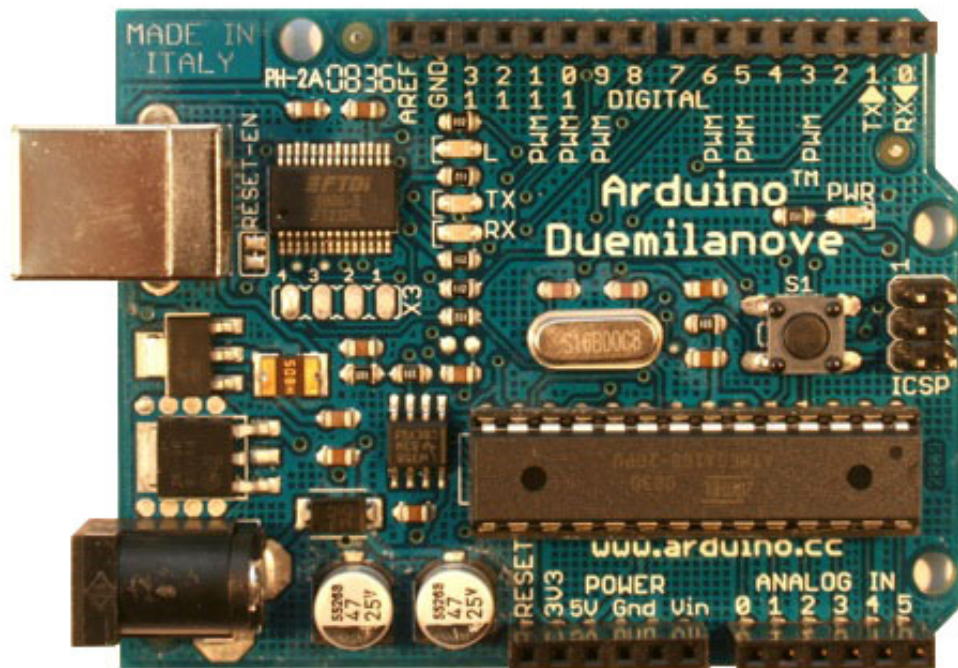


## Arduino Duemilanove com ATmega328

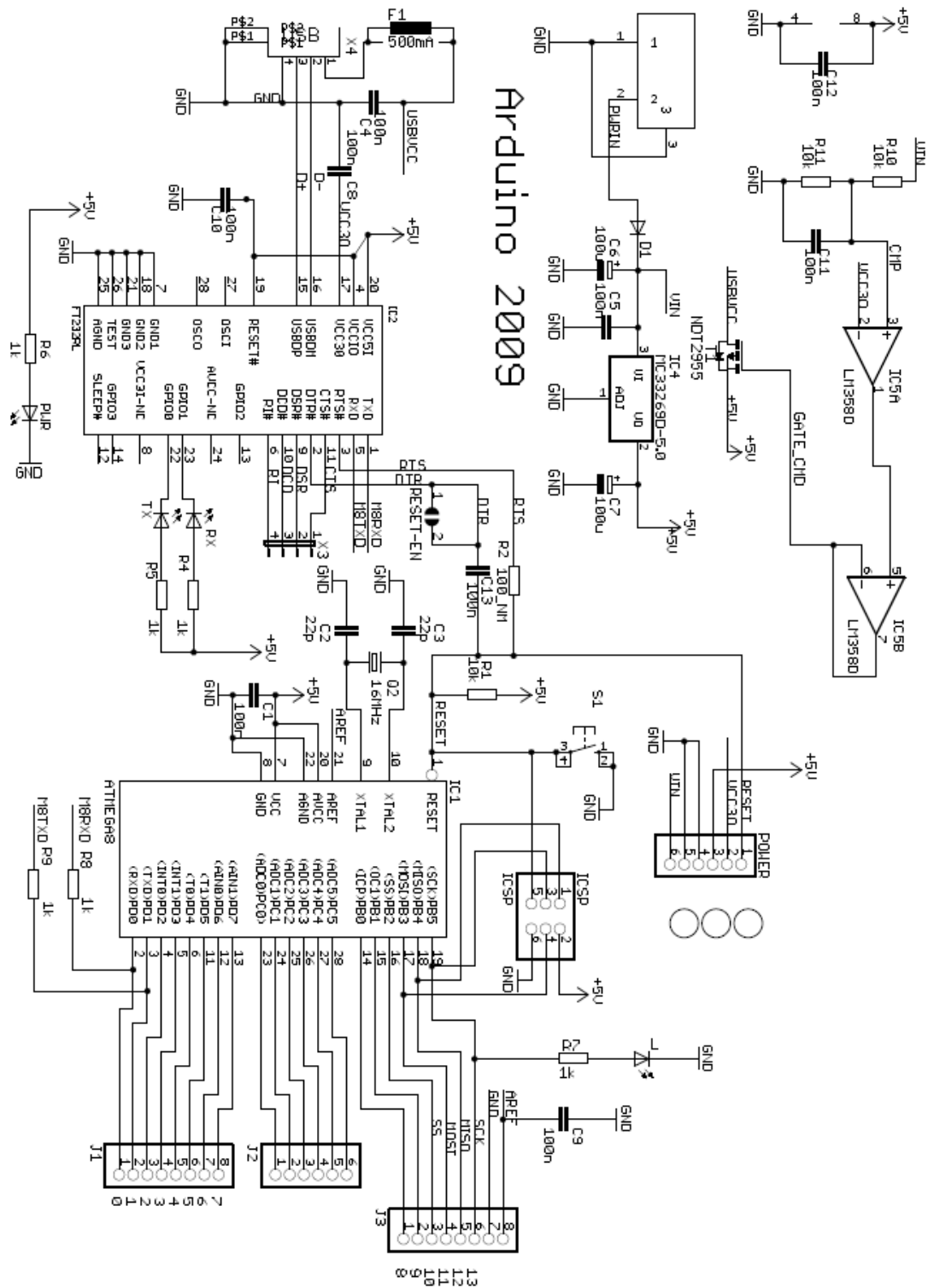
O Arduino Duemilanove (“2009”) é uma placa de microcontrolador baseada no ATmega328. Ele possui 14 pinos de entrada/saída digital (dos quais 6 podem ser usados como saídas analógicas PWM), 6 entradas analógicas, um cristal oscilador de 16 MHz, uma conexão USB, uma entrada para alimentação e um botão de reset.

Ele contém tudo que é necessário para que o microcontrolador funcione; para começar, apenas faça a conexão com um computador através de um cabo USB ou use uma fonte de alimentação de corrente contínua ou uma bateria.

“Duemilanove” significa 2009 em italiano e o nome foi escolhido pelo ano de lançamento. O Duemilanove é a mais recente de uma série de placas Arduino USB.



## Esquema



## Características

Microcontrolador	ATmega328
Voltagem operacional	5V
Voltagem de alimentação (recomendada)	7-12V
Voltagem de alimentação (limites)	6-20V
Pinos I/O digitais	14 (dos quais 6 podem ser saídas PWM)
Pinos de entrada analógica	6
Corrente contínua por pino I/O	40 mA
Corrente contínua para o pino 3.3V	50 mA
Memória flash	32 KB (2KB usados para o bootloader)
SRAM	2 KB
EEPROM	1 KB
Velocidade de clock	16 MHz

## Alimentação

O Arduino Duemilanove pode ser alimentado pela conexão USB ou por qualquer fonte de alimentação externa. A fonte de alimentação é selecionada automaticamente. Alimentação externa (não-USB) pode ser tanto de uma fonte ou de uma bateria. A fonte pode ser conectada com um plug de 2,1mm (centro positivo) no conector de alimentação. Cabos vindos de uma bateria podem ser inseridos nos pinos GND (terra) e  $V_{IN}$  (entrada de voltagem) do conector de alimentação.

A placa pode operar com uma alimentação externa de 6 a 20 volts. Entretanto, se a alimentação for inferior a 7 volts o pino 5V pode fornecer menos de 5 volts e a placa pode ficar instável. Se a alimentação for superior a 12 volts o regulador de voltagem pode superaquecer e avariar a placa. A alimentação recomendada é de 7 a 12 volts. Os pinos de alimentação são:

- **$V_{IN}$ .** Entrada de alimentação para a placa Arduino quando uma fonte externa for utilizada. Você pode fornecer alimentação por este pino ou, se usar o conector de alimentação, acessar a alimentação por este pino.
- **5V.** A fonte de alimentação utilizada para o microcontrolador e para outros componentes da placa. Pode ser proveniente do pino  $V_{IN}$  através de um regulador on-board ou ser fornecida pelo USB ou outra fonte de 5 volts.
- **3V3.** Alimentação de 3,3 volts fornecida pelo chip FTDI. A corrente máxima é de 50 mA.
- **GND.** Pino terra.

## Memória

O ATmega328 tem 32 KB de memória flash para armazenar código (2 KB utilizados pelo bootloader), além de 2 KB de SRAM e 1 KB of EEPROM, que pode ser lida e escrita p/ **biblioteca EEPROM** - <http://www.arduino.cc/en/Reference/EEPROM>.

## Entrada e Saída

Cada um dos 14 pinos digitais do Duemilanove pode ser usado como entrada ou saída usando as funções de ***pinMode()***, ***digitalWrite()***, e ***digitalRead()***. Eles operam com 5 volts. Cada pino pode fornecer ou receber um máximo de 40 mA e tem um resistor pull-up interno (desconectado por padrão) de 20-50 KOhms. Além disso, alguns pinos têm funções especializadas:

- **Serial: 0 (RX) e 1 (TX).** Usados para receber (RX) e transmitir (TX) dados seriais TTL. Estes pinos são conectados aos pinos correspondentes do chip serial FTDI USB-to-TTL.
- **External Interrupts: 2 and 3.** Estes pinos podem ser configurados para disparar uma interrupção por um baixo valor, uma elevação ou falling edge ou uma mudança de valor. Veja a função ***attachInterrupt()*** para mais detalhes.
- **PWM: 3, 5, 6, 9, 10, e 11.** Fornecem uma saída analógica PWM de 8-bit com a função ***analogWrite()***.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** Estes pinos suportam comunicação SPI, que embora compatível com o hardware, não está incluída na linguagem do Arduino.
- **LED: 13.** Há um LED já montado e conectado ao pino digital 13. Quando o pino está no valor HIGH, o LED acende; quando o valor está em LOW, ele apaga.

O Duemilanove tem 6 entradas analógicas e cada uma delas tem uma resolução de 10 bits (i.e. 1024 valores diferentes). Por padrão, elas medem de 0 a 5 volts, embora seja possível mudar o limite superior usando o pino AREF e um pouco de código de baixo nível. Adicionalmente alguns pinos têm funcionalidades especializadas:

- **I<sup>2</sup>C: 4 (SDA) and 5 (SCL).** Suportam comunicação I<sup>2</sup>C (TWI) usando a **biblioteca Wire** (<http://wiring.org.co/reference/libraries/Wire/index.html>, documentação no site do WIRE).

Há ainda alguns outros pinos na placa:

- **AREF.** Referência de voltagem para entradas analógicas. Usados com ***analogReference()***.

- **Reset.** Envie o valor LOW para resetar o microcontrolador. Tipicamente utilizados para adicionar um botão de reset aos shields que bloqueiam o que há na placa.

## Mapeamento entre os pinos do Arduino e as portas do ATmega 168.

### Atmega168 Pin Mapping

Arduino function						Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)		analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)		analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)		analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)		analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)		analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)		analog input 0
VCC	VCC	7	22	GND		GND
GND	GND	8	21	AREF		analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC		VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)		digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)		digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)		digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)		digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)		digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

## Comunicação

Com o Arduino Duemilanove a comunicação com um computador, com outro Arduino ou com outros microcontroladores é muito simplificada. O ATmega328 permite comunicação serial no padrão UART TTL (5V), que está disponível nos pinos digitais 0 (RX) e 1 (TX). Um chip FTDI FT232RL na placa encaminha esta comunicação serial através do USB e os **drives FTDI** (<http://www.ftdichip.com/Drivers/VCP.htm>, incluído no software do Arduino) fornece uma porta COM virtual para o software no computador. O software Arduino inclui um monitor serial que permite que dados simples de texto sejam enviados à placa Arduino. Os LEDs RX e TX da placa piscam quando os dados estão sendo transferidos ao computador pelo chip FTDI e a conexão USB (mas não quando há comunicação serial pelos pinos 0 e 1).

A **biblioteca** **SoftwareSerial** (<http://www.arduino.cc/en/Reference/SoftwareSerial>) permite comunicação serial por quaisquer dos pinos digitais do Duemilanove.

O ATmega328 também oferece suporte aos padrões de comunicação I<sup>2</sup>C (TWI) e SPI (Serial Peripheral Interface Bus). O software do Arduino inclui uma **biblioteca Wire** para simplificar o uso do barramento I<sup>2</sup>C (Inter-Integrated Circuit Bus); veja a **documentação no site do Wiring** (<http://wiring.org.co/reference/libraries/Wire/index.html>) para mais detalhes. Para usar a comunicação SPI veja a folha de informações do ATmega328.

## Programação

O Arduino Duemilanove pode ser programado com o **software Arduino** (<http://arduino.cc/en/Main/Software>).

O ATmega328 no Arduino Duemilanove vem pré-gravado com um **bootloader** (<http://arduino.cc/en/Tutorial/Bootloader>) que permite enviar novos programas sem o uso de um programador de hardware externo. Ele se comunica utilizando o **protocolo original STK500** (referência: [http://www.atmel.com/dyn/resources/prod\\_documents/doc2525.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2525.pdf) - Cheader files: [http://www.atmel.com/dyn/resources/prod\\_documents/avr061.zip](http://www.atmel.com/dyn/resources/prod_documents/avr061.zip)).

O ATmega328 também pode ser programado através do ICSP (In-Circuit Serial Programming) header; veja estas **instruções** (<http://arduino.cc/en/Hacking/Programmer>) para mais detalhes.

## Reset automático (Software)

Algumas versões anteriores do Arduino requerem um reset físico (pressionando o botão de reset na placa) antes de carregar um sketch. O Arduino Duemilanove é projetado de modo a permitir que isto seja feito através do software que esteja rodando no computador conectado. Uma das linhas de controle de hardware (DTR) do FT232RL está conectada ao reset do ATmega328 via um capacitor de 100 nanofarads. Quando esta linha é resetada (ativo baixo) o sinal cai por tempo suficiente para resetar o chip. O software Arduino usa esta característica para permitir carregar o programa simplesmente pressionando o botão “upload” no ambiente Arduino. Isto significa que o “bootloader” pode ter um “timeout” mais curto, já que a ativação do DTR (sinal baixo) pode ser bem coordenada com o início do “upload”.

Esta configuração tem outras implicações. Quando o Duemilanove está conectado a um computador rodando Mac OS X ou Linux, ele reseta toda vez que a conexão é feita por software (via USB). No próximo meio segundo aproximadamente, o bootloader estará rodando no Duemilanove. Considerando que é programado para ignorar dados espúreos (i.e. qualquer coisa a não ser um “upload” de um novo código), ele interceptará os primeiros bytes de dados sendo enviados para a placa depois que a conexão é aberta. Se um “sketch” rodando na placa recebe configuração de uma vez ou outros dados ao inicializar, assegure-se que o software que esteja comunicando espere um segundo depois de aberta a conexão antes de enviar estes dados.

O Duemilanove tem uma trilha que pode ser cortada para desabilitar o auto-reset e pode ser ressoldada para reativá-lo. É chamada de "RESET-EM". Você pode também desabilitar o auto-reset conectando um resistor de 110 ohms dos +5V até o sinal de reset; veja este **fórum** (<http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1213719666/all>) para detalhes.

## Proteção contra sobrecorrente USB

O Arduino Duemilanove tem um polifusível resetável que protege a porta USB do seu computador contra curto-circuito e sobrecorrente. Apesar da maioria dos computadores possuírem proteção interna própria, o fusível proporciona uma proteção extra. Se mais de 500mA foram aplicados na porta USB, o fusível irá automaticamente interromper a conexão até que o curto ou a sobrecarga seja removida.

## Características físicas

O comprimento e largura máximos do Duemilanove são 2,7" (68,50 mm) e 2,1" (53,34 mm) respectivamente, com o conector USB e o jack de alimentação indo um pouco além destas dimensões. Três furos de fixação permitem a montagem da placa numa superfície ou caixa. Note que a distância entre os pinos de entrada e saída digitais nº 7 e nº 8 é de 160 mil (milésimos de polegada), não é sequer múltiplo do espaçamento de 100 mil dos outros pinos.

## Referência da linguagem de programação Arduino

A descrição completa da linguagem Arduino está em <http://arduino.cc/en/Reference/HomePage>; a linguagem estendida está em <http://arduino.cc/en/Reference/Extended> e as bibliotecas estão em <http://arduino.cc/en/Reference/Libraries>.

No site brasileiro [Multilógica](http://multilogica.com) existe a tradução de todo este material de referência para a língua portuguesa: <http://multilogica-shop.com/Referencia>.

## Fontes principais:

<http://www.arduino.cc>

<http://www.atmel.com>

<http://multilogica-shop.com/>

<http://www.wiring.org>