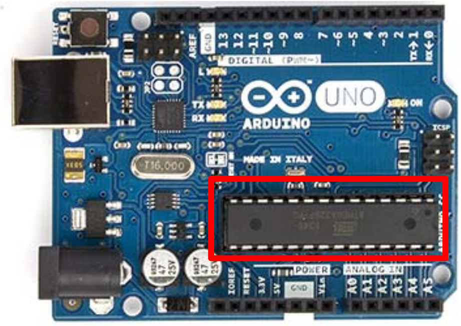
Arduino UNO

SAPIENZA
UNIVERSITÀ DI ROMA

Laboratorio Multidisciplinare di Elettronica II

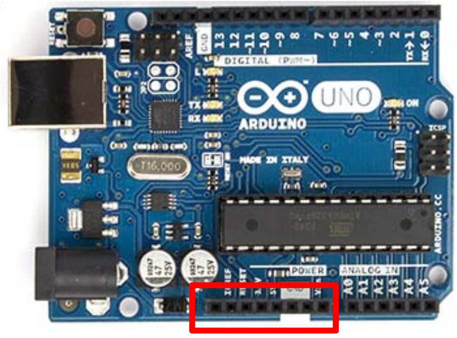
HARDWARE

Il nome della scheda deriva da quello di un bar di Ivrea (che richiama a sua volta il nome di Arduino d'Ivrea, Re d'Italia nel 1002) frequentato da alcuni dei fondatori del progetto



- Microcontroller(ATmega328)
- Operating Voltage 5V
- Input Voltage (limits) 7-12V (6 20V)
- Digital I/O Pins 14 (of which 6 PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 32 KB
- 0.5 KB us bootloader
- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- Clock Speed 16 MHz
- Length 68.6 mm
- Width 53.4 mm
- Weight 25 g

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.



- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of

Description

Reads the value from the specified analog pin. The Arduino board contains a 6 channel (8 channels on the Mini and Nano, 16 on the Mega), 10-bit analog to digital converter. This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023. This yields a resolution between readings of: 5 volts / 1024 units or, .0049 volts (4.9 mV) per unit. The input range and resolution can be changed using analogReference().

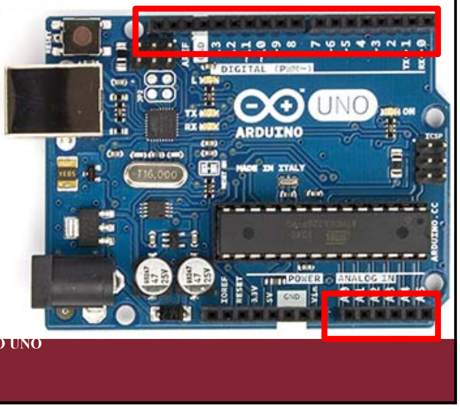
It takes about 100 microseconds (0.0001 s) to read an analog input, so the maximum reading rate is about 10,000 times a second.

using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

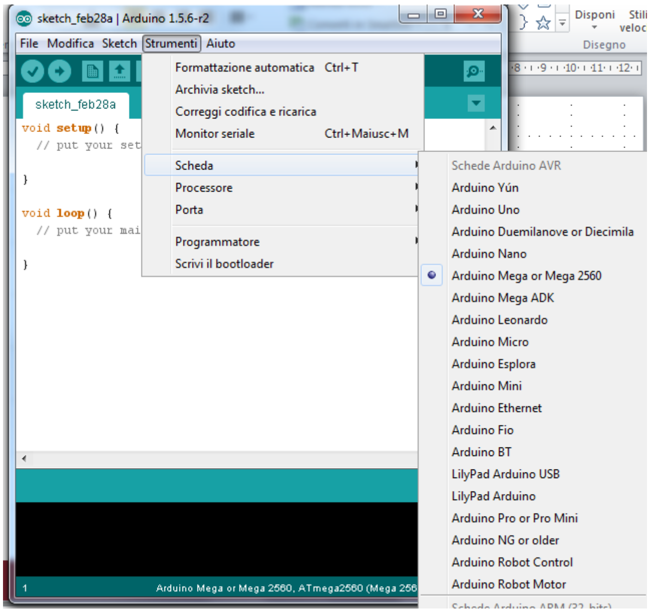
- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analogReferen
- Reset. Bring this line LOW to reset the microcontroller. Typically used to board.

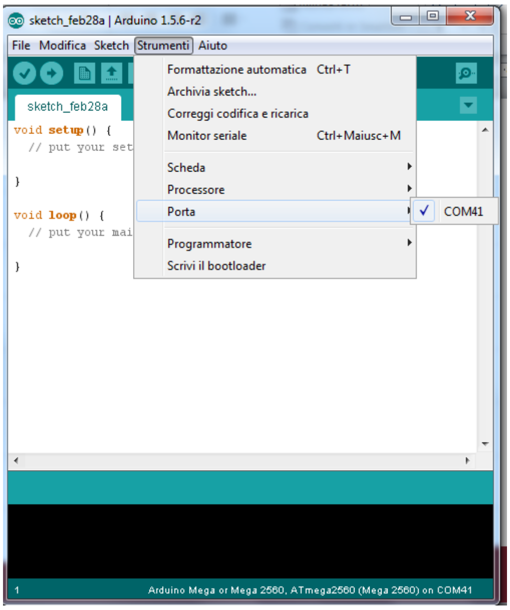


IDE Programmazione



```
void setup() {
  // put your set

}

void loop() {
  // put your mai

}
```

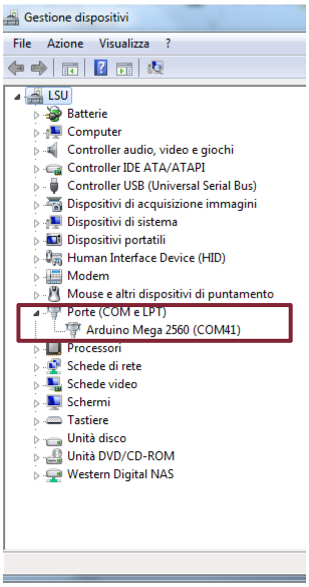
1 Arduino Mega or Mega 2560, ATmega2560 (Mega 256



sketch_feb28a | Arduino 1.5.6-r2
File Modifica Sketch Strumenti Aiuto
sketch_feb28a
Formattazione automatica Ctrl+T
Archivia sketch...
Correggi codifica e ricarica
Monitor seriale Ctrl+Maiusc+M
Scheda
Processore
Porta
✓ COM41
Programmatore
Scrivi il bootloader

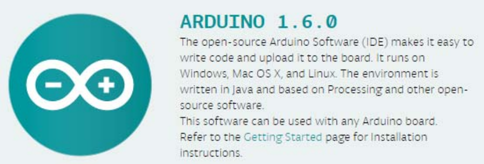
```
void setup() {
  // put your set

}

void loop() {
  // put your mai

}
```

1 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM41



Wiring (C/C++ library) is an open-source programming framework for microcontrollers. Wiring allows writing cross-platform software to control devices attached to a wide range of microcontroller boards to create all kinds of creative coding, interactive objects, spaces or physical experiences.

- Roadmap include support for multiple hardware architectures "Cores"
- The current AVR8 Core supports *any hardware* based on the AVR Atmega processors
- Simple third party Atmel hardware support integration
- For Linux, Mac OS X, and Windows

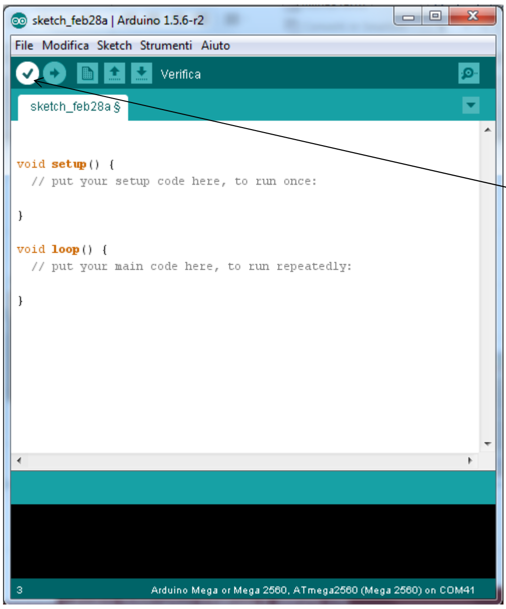


ARDUINO 1.6.0

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board.

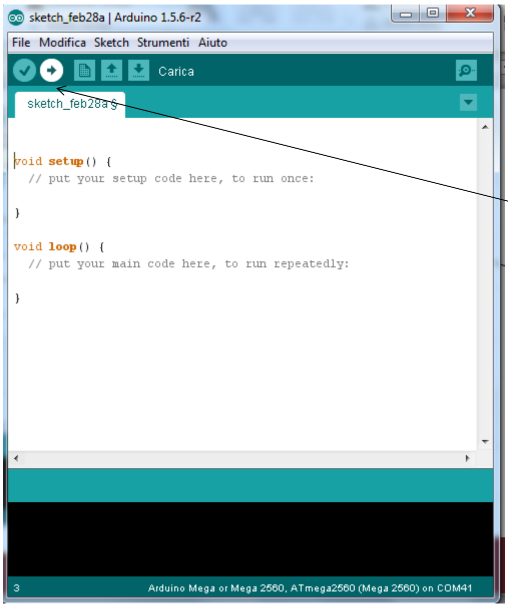
Refer to the Getting Started page for Installation instructions.


sketch_feb28a | Arduino 1.5.6-r2
File Modifica Sketch Strumenti Aiuto
Verifica
sketch_feb28a §

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

3 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM41

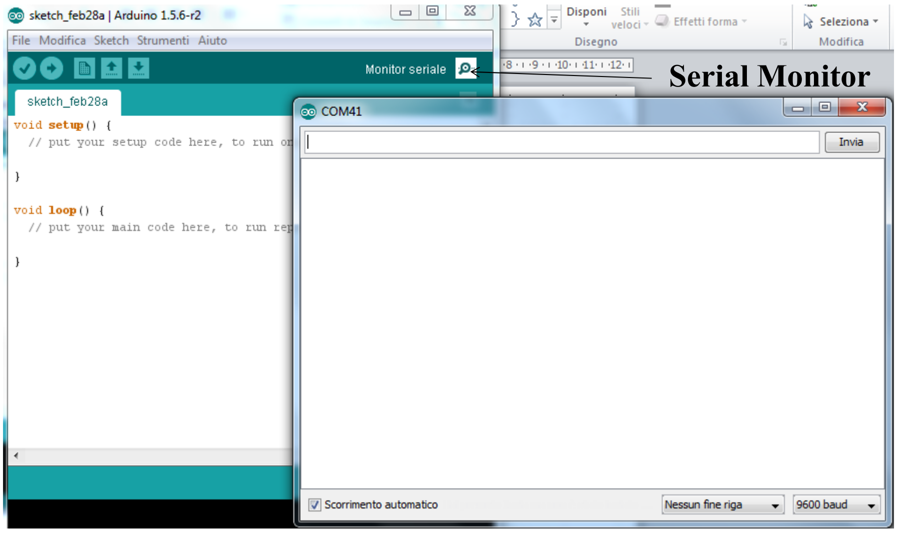
Verifica


sketch_feb28a | Arduino 1.5.6-r2
File Modifica Sketch Strumenti Aiuto
Carica
sketch_feb28a §

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

```
3        Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM41
```

Carica

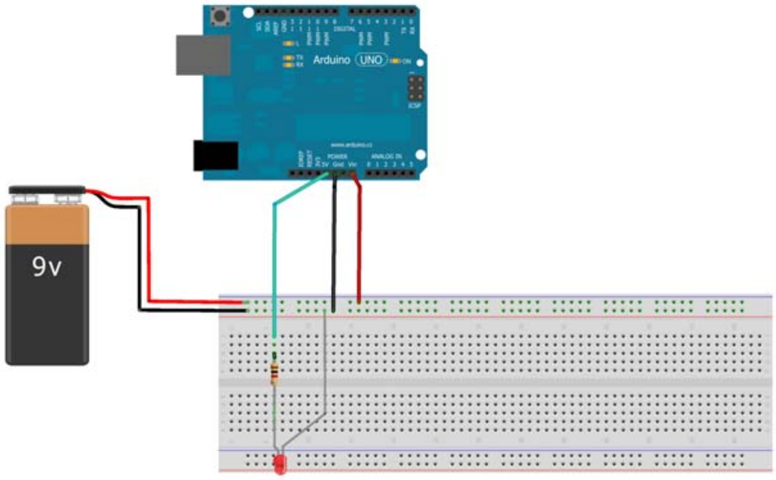


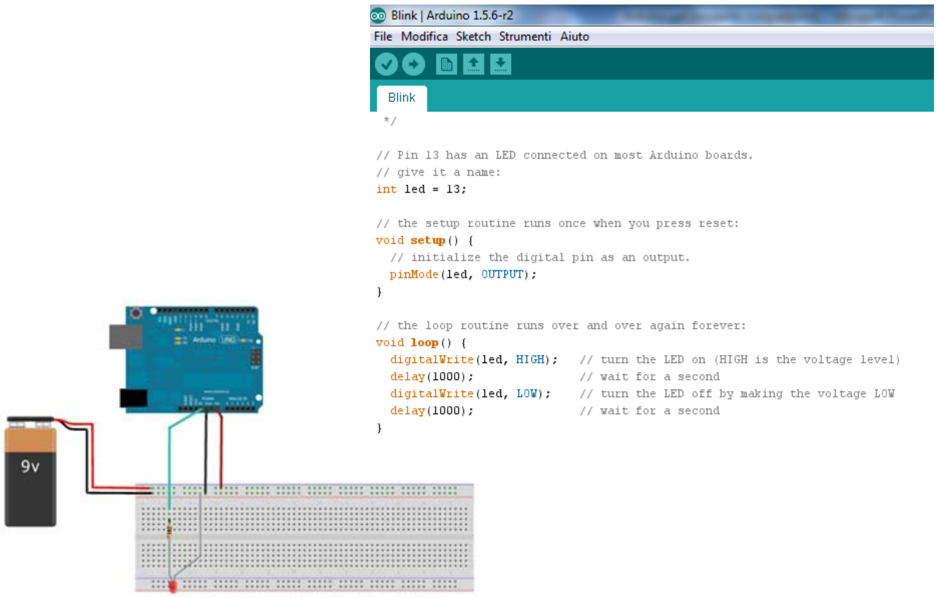
```
void setup() {
  // put your setup code here, to run o

}

void loop() {
  // put your main code here, to run rep

}
```

☑ Scorrimento automatico   Nessun fine riga   9600 baud

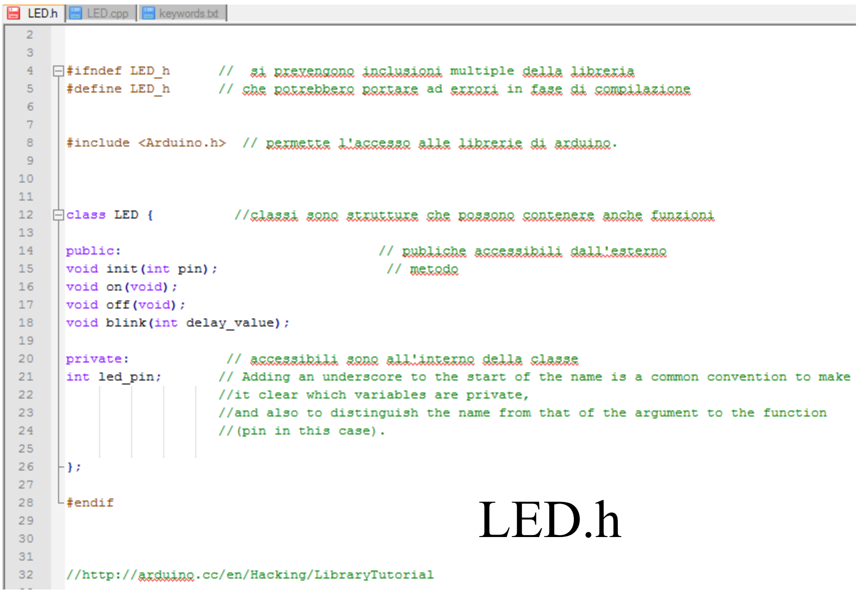
Serial Monitor

SCRIVERE LIBRERIE
(DRIVER di Controllo)



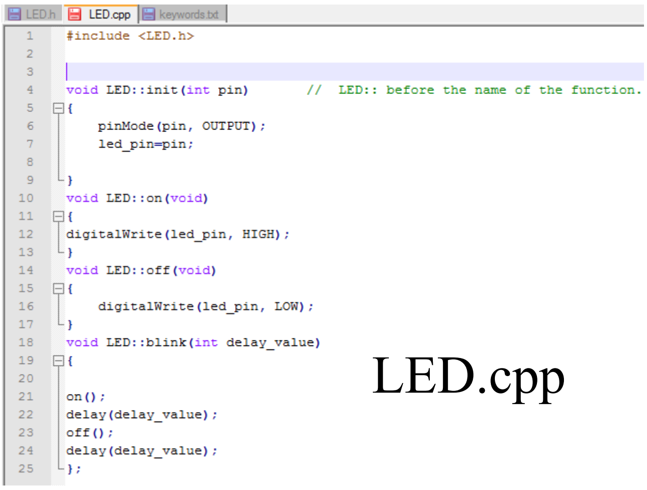


```
 */

// 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
}
```



```


#ifndef LED_h      //  si prevengono inclusioni multiple della libreria
#define LED_h      // che potrebbero portare ad errori in fase di compilazione


#include <Arduino.h>  // permette l'accesso alle librerie di arduino.



class LED {           //classi sono strutture che possono contenere anche funzioni

public:                                   // publiche accessibili dall'esterno
void init(int pin);                        // metodo
void on(void);
void off(void);
void blink(int delay_value);

private:            // accessibili sono all'interno della classe
int led_pin;       // Adding an underscore to the start of the name is a common convention to make
                   //it clear which variables are private,
                   //and also to distinguish the name from that of the argument to the function
                   //(pin in this case).

};

#endif



//http://arduino.cc/en/Hacking/LibraryTutorial
```

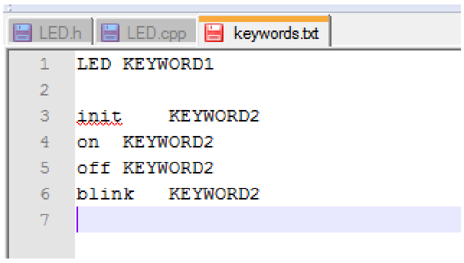
LED.h



```
#include <LED.h>


void LED::init(int pin)        //  LED:: before the name of the function.
{
    pinMode(pin, OUTPUT);
    led_pin=pin;

}
void LED::on(void)
{
digitalWrite(led_pin, HIGH);
}
void LED::off(void)
{
    digitalWrite(led_pin, LOW);
}
void LED::blink(int delay_value)
{

on();
delay(delay_value);
off();
delay(delay_value);
};
```

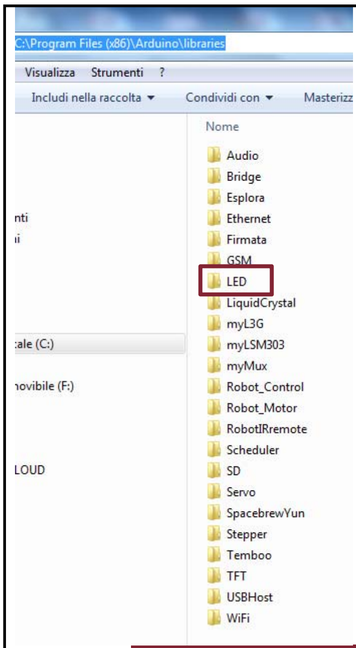
LED.cpp

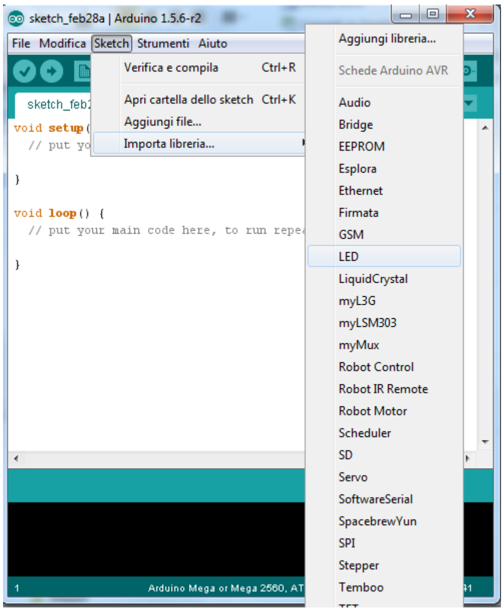
KEYWORD1 Classes, datatypes, and C++ keywords
KEYWORD2 Methods and functions
KEYWORD3 setup and loop functions,
LITERAL1 Constants
LITERAL2 Built-in variables (unused by default)
Each line has the name of the keyword, followed by a tab (not spaces), followed by the kind of keyword.
You'll have to restart the Arduino environment to get it to recognize the new keywords.



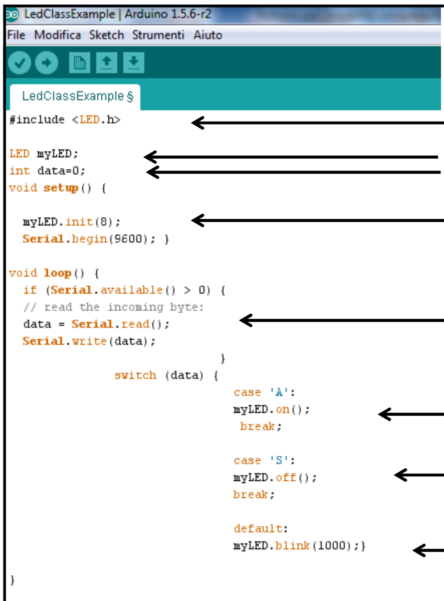
```
LED	KEYWORD1

init	KEYWORD2
on	KEYWORD2
off	KEYWORD2
blink	KEYWORD2

```

keywords.txt





```
void setup(
  // put yo

}

void loop() {
  // put your main code here, to run repe

}
```



```
#include <LED.h>

LED myLED;
int data=0;
void setup() {

  myLED.init(8);
  Serial.begin(9600); }

void loop() {
  if (Serial.available() > 0) {
  // read the incoming byte:
  data = Serial.read();
  Serial.write(data);
                                  }
             switch (data) {
                              case 'A':
                              myLED.on();
                               break;

                              case 'S':
                              myLED.off();
                              break;

                              default:
                              myLED.blink(1000);}

}
```

Includo la libreria

Oggetto tipo LED

Variabile Globale

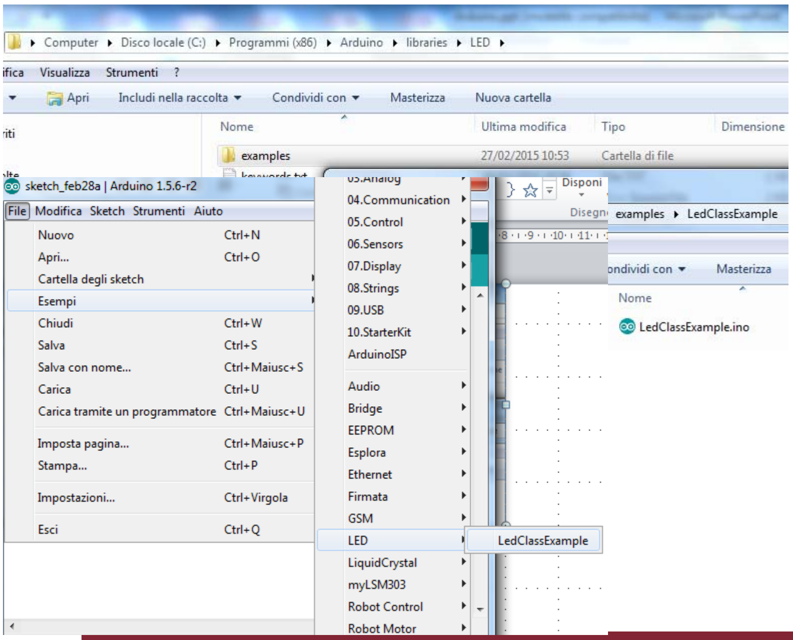
LED collegato al PIN 8

Leggi il dato dalla seriale

(A)ccendi il LED

(S)pegni il LED

Altrimenti Lampeggia 1 sec


Computer ▸ Disco locale (C:) ▸ Programmi (x86) ▸ Arduino ▸ libraries ▸ LED ▸
ifica Visualizza Strumenti ?
Apri Includi nella raccolta Condividi con Masterizza Nuova cartella
Nome Ultima modifica Tipo Dimensione
examples 27/02/2015 10:53 Cartella di file
sketch_feb28a | Arduino 1.5.6-r2
File Modifica Sketch Strumenti Aiuto
Nuovo Ctrl+N
Apri... Ctrl+O
Cartella degli sketch
Esempi
Chiudi Ctrl+W
Salva Ctrl+S
Salva con nome... Ctrl+Maiusc+S
Carica Ctrl+U
Carica tramite un programmatore Ctrl+Maiusc+U
Imposta pagina... Ctrl+Maiusc+P
Stampa... Ctrl+P
Impostazioni... Ctrl+Virgola
Esci Ctrl+Q
03.Analog
04.Communication
05.Control
06.Sensors
07.Display
08.Strings
09.USB
10.StarterKit
ArduinoISP
Audio
Bridge
EEPROM
Esplora
Ethernet
Firmata
GSM
LED
LiquidCrystal
myLSM303
Robot Control
Robot Motor
LedClassExample
Disponi
Disegno
examples ▸ LedClassExample
ondividi con Masterizza
Nome
LedClassExample.ino