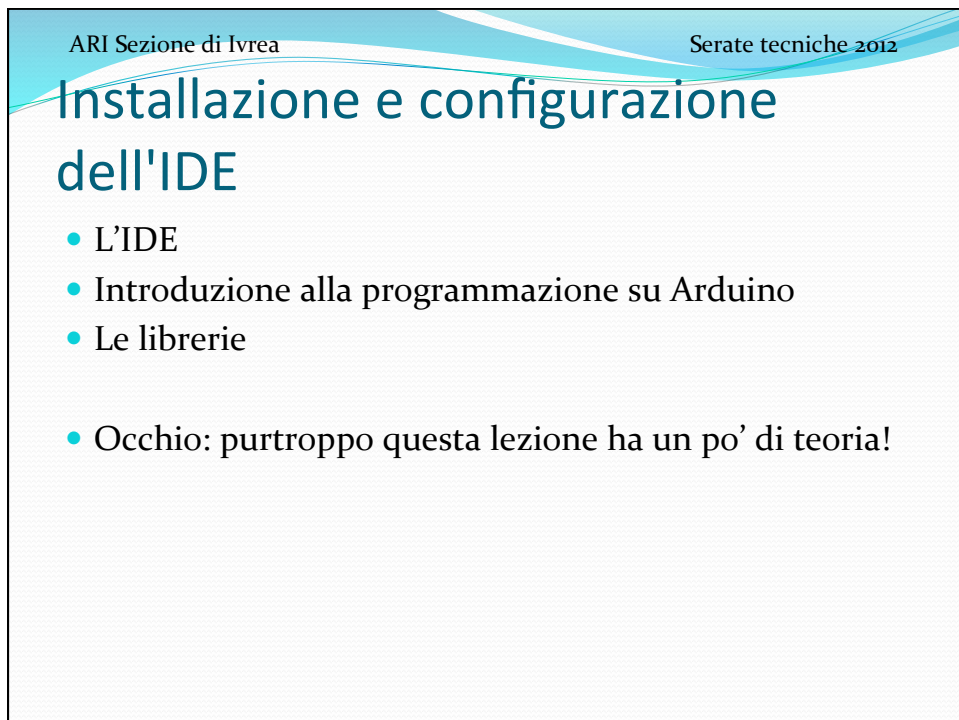
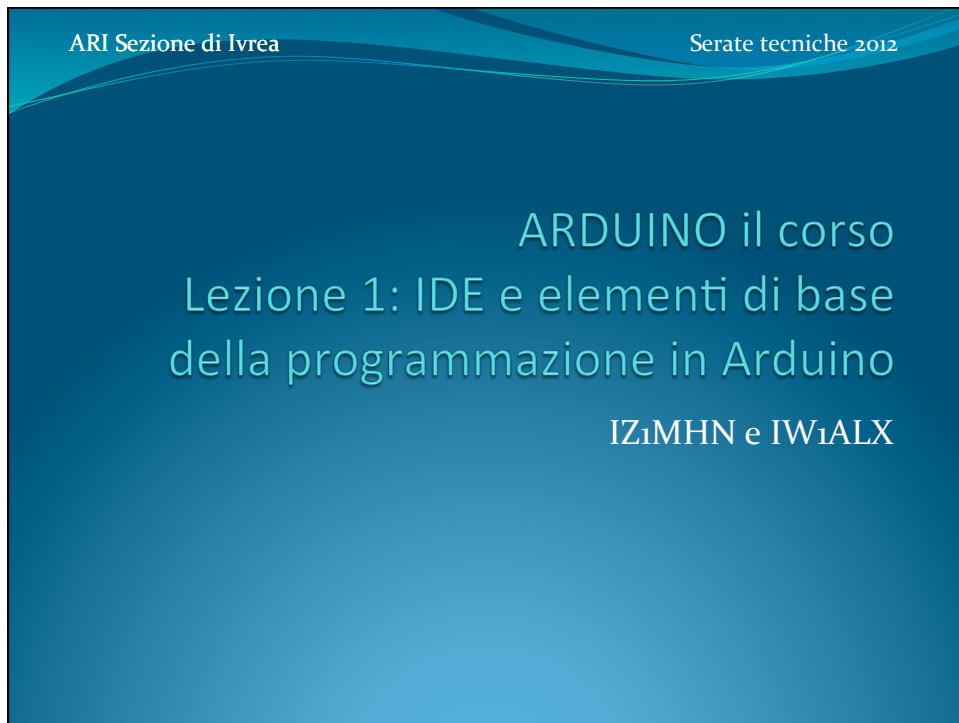




ARI Sezione di Ivrea Serate tecniche 2012

L'IDE o ADE

- Cos'è un IDE
- Installazione dell' Arduino Development Environment
- Principali funzioni

ARI Sezione di Ivrea Serate tecniche 2012

Cos'è un IDE

- Un IDE (Integrated Development Environment) è un software che offre un insieme di funzioni per sviluppare.
- Tipicamente contiene tre componenti:
 - Un editor.
 - Uno strumento di automazione (per interfacce o codice).
 - Un debugger.
 - In più un IDE per microcontrollori consente l'upload sulle schede.
- L'IDE di Arduino prende il nome di ADE (Arduino Development Environment):
 - Multiplatforma (Windows, MacOS e Linux).
 - Basato su Processing.
 - Open Source.

Installazione dell'IDE

- L'installazione è simile su tutti gli ambienti (anche se, ovviamente su Mac funziona meglio ;-).
- Si scarica da <http://arduino.cc/en/Main/Software>.
- Si basa su Java ☺, pertanto è necessario scaricare anche il JRE. NON serve il JDK!
- Al momento del corso la versione per Arduino UNO R3 più recente è la 1.0.2
- Sul sito trovate anche le versioni vecchie dell'IDE: possono servire con particolari librerie (molte librerie sono testate e collaudate sulla 1.0)

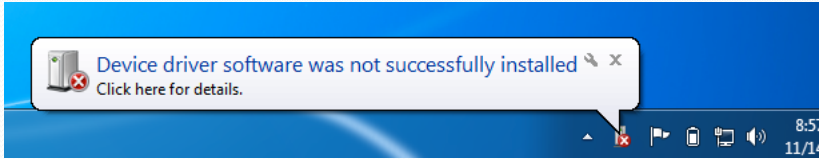
Installazione dell'IDE

- Una volta scaricato il file compresso è sufficiente scompattarlo.
- Viene creata una cartella arduino-[versione], ad esempio arduino-1.0.2.
- È possibile spostare la cartella dove si vuole!
- **NON LANCIATE ARDUINO ADESSO!**

ARI Sezione di Ivrea Serate tecniche 2012

Installazione dei driver

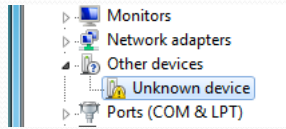
- Collegare la scheda al PC con un cavo USB A-B.
- L'alimentazione viene fornita direttamente dal PC.
- Quando collegate la scheda parte la procedura di installazione dei driver, che quasi di sicuro... fallisce!



ARI Sezione di Ivrea Serate tecniche 2012

Installazione dei driver

- Dal Control Panel (Pannello di Controllo) lanciamo (con i privilegi di amministratore) Device Manager (Gestione Dispositivi)
- Si trova una voce con un punto esclamativo su sfondo giallo (la posizione varia con il sistema)
 - Su XP finisce sotto Ports
 - Su Vista e 7 sotto Other device



ARI Sezione di Ivrea

Serate tecniche 2012

Installazione dei driver

- Selezionare la voce quindi tasto destro poi **Update Driver Software...**
- Selezionare **Browse my computer for driver software**
- Selezionare il pulsante **Browse** quindi navigare il disco fino ad arrivare a selezionare la cartella **Drivers**, ad esempio
c:\programmi\arduino-1.0.2\drivers.
- Potrebbe comparire una segnalazione sui driver che possiamo ignorare.
- Al termine un messaggio ci conferma che abbiamo correttamente installato una scheda Arduino UNO R3!

ARI Sezione di Ivrea

Serate tecniche 2012

The screenshot shows the Windows Device Manager window with the 'Other devices' section expanded. A right-click context menu is open over an 'Unknown device', with 'Update Driver Software...' selected. Below this, the 'Browse my computer for driver software' wizard is displayed. The wizard shows the path 'C:\Program Files\arduino-1.0.2\drivers' entered in the 'Search for driver software in this location:' field. At the bottom, a confirmation message states: 'Windows has successfully updated your driver software' and 'Windows has finished installing the driver software for this device: Arduino UNO R3'.

Bloccare la seriale

- Per semplificarci la vita conviene controllare la seriale cui viene associato la scheda Arduino
 - Tasto destro sulla scheda: Arduino UNO R3 (Com X)
 - Properties
 - Port Settings
 - Advanced...
 - Quindi selezionare il Port Number più basso disponibile!

Configurazione di base dell'IDE

- Per prima cosa dal menù Tools selezionare la voce Board quindi scegliere il nostro modello.
- Poi sempre da Tools selezionare la voce Serial Port quindi scegliere la porta che abbiamo precedentemente impostato.



ARI Sezione di Ivrea Serate tecniche 2012

Altre opzioni

- Dal menù File scegliere Preferences
 - Impostare la lingua che si vuole
 - Opzionalmente scegliere di avere l'Output "verbose" per la verifica e/o per l'upload

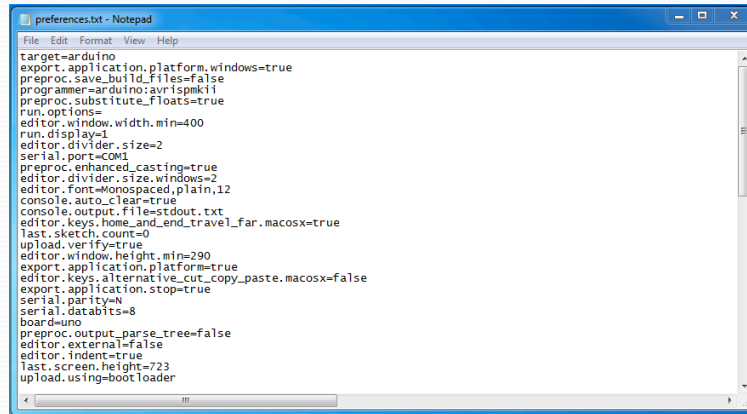
The image shows the 'Preferences' dialog box in the Arduino IDE. It contains the following settings:

- Sketchbook location: C:\Users\Admin\Documents\Arduino
- Editor language: System Default (requires restart of Arduino)
- Editor font size: 12 (requires restart of Arduino)
- Show verbose output during: ☐ compilation ☐ upload
- ☒ Verify code after upload
- ☐ Use external editor
- ☒ Check for updates on startup
- ☒ Update sketch files to new extension on save (.pde -> .ino)
- ☒ Automatically associate .ino files with Arduino

 At the bottom, it states: 'More preferences can be edited directly in the file C:\Users\Admin\AppData\Roaming\Arduino\preferences.txt (edit only when Arduino is not running)'. There are 'OK' and 'Cancel' buttons at the bottom right.

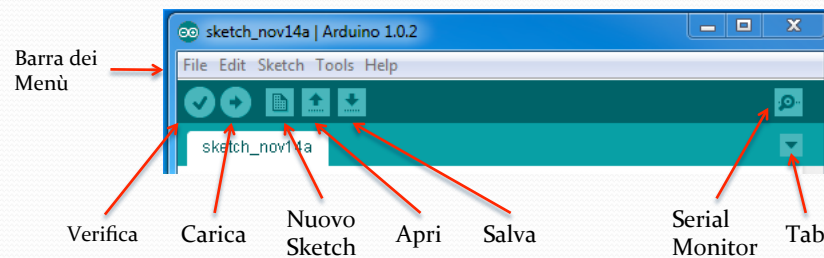
Opzioni avanzate

C:\Users\XXX\AppData\Roaming\Arduino\preferences.txt



```
preferences.txt - Notepad
File Edit Format View Help
target=arduino
export.application.platform.windows=true
preproc.save_build_files=false
programmer=arduino:avrispmkii
preproc.substitute_floats=true
run.options=
editor.window.width.min=400
run.display=1
editor.divider.size=2
serial.port=COM1
preproc.enhanced_casting=true
editor.divider.size.windows=2
editor.font=Monospaced,plain,12
console.auto_clear=true
console.output.file=stdout.txt
editor.keys.home_and_end_travel_far.macosx=true
last.sketch.count=0
upload.verify=true
editor.window.height.min=290
export.application.platform=true
editor.keys.alternative_cut_copy_paste.macosx=false
export.application.stop=true
serial.parity=N
serial.databits=8
board=uno
preproc.output_parse_tree=false
editor.external=false
editor.indent=true
last.screen.height=723
upload.using=bootloader
```

Principali funzioni dell'IDE



ARI Sezione di Ivrea Serate tecniche 2012

Principali funzioni dell'IDE

- Menù Edit
 - Copy for Forum
 - Copy as HTML
 - Indent
- Menù Sketch
 - Verify/Compile
 - Show Sketch Folder
 - Add File...
 - Import Library
- Menù Tools
 - Auto Format
 - Archive Sketch
 - Programmer
 - Burn Bootloader

ARI Sezione di Ivrea Serate tecniche 2012

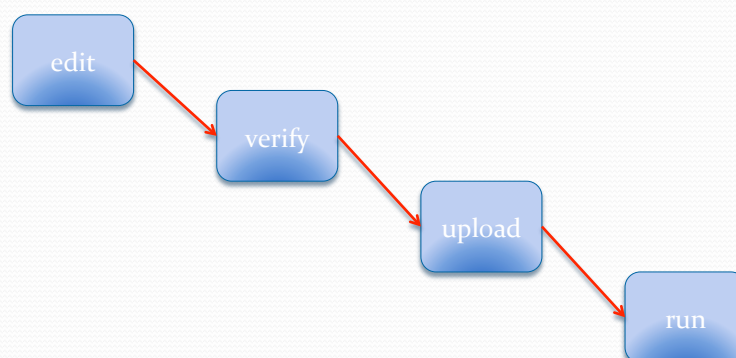
LAB

- A voi la palla:
 - Installate l'ambiente sul vostro PC!

Introduzione alla programmazione su Arduino

- Il primo sketch: accendere un led
- Variabili e procedure
- Il Serial Monitor

Ciclo di sviluppo su ADA



Gli Sketch

- Uno Sketch è un programma nel gergo di Arduino.
In pratica è ciò che viene caricato nel micro.
- Si compone di due aree principali
 - La dichiarazione delle variabili
 - Le funzioni
- È buona norma aggiungere una terza parte all'inizio: i commenti

Il linguaggio

- Simile al C e al C++
- <http://arduino.cc/en/Reference/HomePage>
- Si compone di tre “parti”:
 - Strutture
 - Setup, loop, strutture logiche e gli operatori
 - Valori
 - Costanti e variabili
 - Funzioni
 - I/O digitale e analogico, matematiche, gestione del “tempo”, interrupt e analogiche

Accendere un led

```
/*  
  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  
}
```

Sintassi

- ; il punto e virgola
 - Il punto e virgola (;) va inserito al termine di ogni linea.
 - Il ; va anche usato per separare gli elementi in un For.
- {} le parentesi graffe
 - Le parentesi graffe { } definiscono l'inizio e la fine di un blocco di istruzioni o di una funzione.
 - Ad una parentesi che si apre deve corrisponderne una che si chiude.

I commenti

- Tutto il testo (anche multilinea) compreso tra /* e */ viene considerato un commento
- Se una linea inizia con // è considerata un commento
- Usate sempre i commenti e scriveteli in modo che vi servano tra un po' di tempo!

Setup ()

- È una funzione
- Viene eseguita all'avvio della scheda.
- Viene eseguita UNA sola volta
- Viene usata per
 - inizializzare le variabili
 - definire il modo dei pin
 - gestire le librerie...

```
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}
```

Loop ()

- È la funzione principale
- In pratica è il programma
- Viene eseguito continuamente fino al reset della scheda

```
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  
}
```

Costanti

- Sono “valori” che non cambiano nel tempo... (che fantasia!)
- Si dividono in tre categorie:
 - Predefinite
 - Intere
 - Floating

Costanti predefinite

- Di nuovo... tre tipi:
 - Livelli logici: TRUE e FALSE
 - Pin Level: HIGH e LOW
 - Digital Pin: INPUT, INPUT_PULLUP, e OUTPUT
 - NB: SONO CASE SENSITIVE!!!
 - <http://arduino.cc/en/Reference/Constants>

Costanti: livelli logici

- False: facile... vale... zero!
- True: sfatiamo un mito... non vale 1! Qualsiasi intero che non sia 0... è true!

Costanti: Pin Level

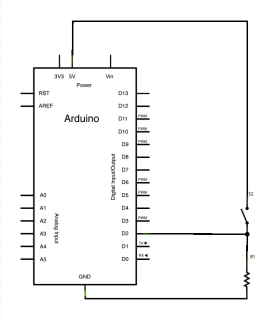
- HIGH
 - Varia se è un Pin di INPUT, di INPUT_PULLUP o di OUTPUT
 - Se un Pin è impostato a INPUT è HIGH se è presente una tensione maggiore o uguale a 3 Volt.
 - Se un Pin è impostato a INPUT_PULLUP, il pin viene messo ad alto e rimane alto (grazie alla resistenza interna da 20K) fino a quando viene mandato basso da un circuito esterno.
 - Se un PIN è impostato a OUTPUT, il pin viene messo a 5 Volt e può generare corrente per alimentare un carico
- LOW
 - Varia se è un Pin di INPUT o di OUTPUT
 - Se è un Pin impostato a INPUT è LOW se è presente una tensione inferiore a 2 Volt.
 - Se un PIN è impostato a OUTPUT, il pin viene messo a 0 Volt

Costanti: Digital Pin

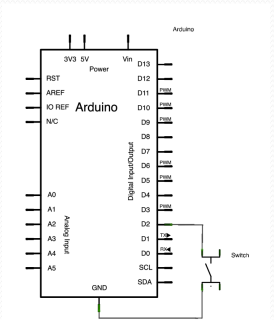
- Gestito dalla funzione PinMode()
- Tutto ciò che non viene specificato in output è di default in input.
- INPUT
 - I Pin vengono messi in uno stato di alta impedenza.
 - Hanno poche richieste nei confronti del circuito, come se ci fossero una serie di resistenze da 100M prima del pin.
 - Normalmente si mette a massa come riferimento (con una resistenza di pull-down).
- INPUT_PULLUP
 - Serve se si preferisce usare la resistenza interna invece che una resistenza esterna di pull-down.
 - Va "pensato" al contrario: HIGH sensore off, e LOW sensore on.
- OUTPUT
 - Sono in uno stato di bassa impedenza.
 - Possono gestire (in o out) fino a 40 mA di corrente.
 - Se messi a massa o a +5V... si distruggono

ARI Sezione di Ivrea Serate tecniche 2012

INPUT_PULLUP



INPUT



INPUT_PULLUP

ARI Sezione di Ivrea Serate tecniche 2012

Costanti Intere e Floating

- Si usano per migliorare la leggibilità del codice
 - Decimali: esempio = 17
 - Binarie: esempio = **B**10001
 - Ottali: esempio = **O**21
 - Esadecimale: esempio = **0x**11
 - Floating: esempio = 17.17
 - Si possono usare sia E sia e per la notazione scientifica (es: 2.34E5)

Variabili

- Una variabile, cioè un valore che cambia nel tempo, deve sempre essere dichiarata.
- Dichiarare una variabile significa
 - definire il tipo di valore
 - assegnarle un nome
 - assegnarle un valore iniziale (facoltativo)
- Ad esempio: `int pippo=10;`

Variabili: lo scope

- Lo scope di una variabile ne definisce la visibilità e dipende da dove viene dichiarata.
- Una variabile può essere definita in tre punti:
 - Fuori da una funzione (ad esempio `setup()` o `loop()`) . In questo caso è una variabile globale.
 - All'interno di funzioni. In questo caso è una variabile locale che vale solo nella funzione in cui è definita.
 - All'interno di un blocco di istruzioni ed è nuovamente locale e vale solo all'interno del blocco.

Variabili: lo scope

```

int led 13;                // variabile globale visibile ad ogni funzione
void loop( ){
  int k=0;                 // variabile locale visibile in tutta la funzione loop
                           // compreso nel ciclo for
  for(int i=0;i<100;i++){
    k++;
    float f=analogRead(2); // variabile locale visibile al solo nel ciclo for
  }
}

```

Variabili: tipi di dati

Tipi di dichiarazione	Rappresentazione	N. di byte	Intervallo
Boolean			True – False / On –Off / High - Low
Char	Carattere	1 (8 bit)	- 127 + 127
Byte	Carattere	1 (8 bit)	0 + 255
Int	Numero intero	2 (16 bit)	-32.768 + 32.767
Unsigned int	Numero intero	2 (16 bit)	0 + 65.535
Short	Numero intero "corto"	2 (16 bit)	
Long	Numero intero "lungo"	4 (32 bit)	-2.147.483.648 + 2.147.483.647
Float	Numero reale	4 (32 bit)	-3.4028235E+38 a + 3.4028235E+38
Double	Numero reale "lungo"	8 (64 bit)	1.7976931348623157 x 10 ³⁰⁸

Variabili: String

- String: occhio alla maiuscola
- Sostituisce il vecchio accesso basato su char*.
- Consente operazioni evolute: concatenazione, append, ricerca e sostituzione...
- Utilizza maggior memoria rispetto all'array di char, ma è molto più flessibile.

Variabili: Array

- È una collezione di variabili che vengono accedute con un indice numerico.
- Gli indici partono da 0.
- Ad esempio:

```
int numeri[6]={1,2,3,4,5,6}; //dichiaro e carico un'array da 6
                             //elementi
int valore;
valore = numeri[4];           //accedo al valore in 5 posizione, quindi
                             //valore=5
```

ARI Sezione di Ivrea Serate tecniche 2012

LAB

- A voi.
 - Caricate il programma blink e giocateci!
 - Provate a fare due lampeggi uno breve e uno lungo in sequenza.

ARI Sezione di Ivrea Serate tecniche 2012

Le librerie

- Il concetto di libreria
- Gestione delle librerie

Il concetto di libreria

- Le librerie estendono il linguaggio base aggiungendo funzionalità extra ad esempio per gestire hardware o dati.
- Le librerie non sono altro che “brandelli” di codice che non dobbiamo scrivere noi!
- Le librerie si possono dividere in tre categorie:
 - Incluse: arrivano insieme all’IDE
 - Terze parti: si scaricano da Internet (tipicamente accompagnano un “componente”)
 - Personali: se dobbiamo ripetere un compito... nessuno ci vieta di scriverci una libreria

Librerie standard

- Alcune librerie sono “dedicate” ad alcune schede (ad esempio Leonardo o Due)
- Le librerie standard sono quelle raccolte sul sito:
 - <http://arduino.cc/en/Reference/Libraries>

ARI Sezione di Ivrea

Serate tecniche 2012

Librerie standard

Standard Libraries

- ✦ **EEPROM** - reading and writing to "permanent" storage
- ✦ **Ethernet** - for connecting to the internet using the Arduino Ethernet Shield
- ✦ **Firmata** - for communicating with applications on the computer using a standard serial protocol.
- ✦ **LiquidCrystal** - for controlling liquid crystal displays (LCDs)
- ✦ **SD** - for reading and writing SD cards
- ✦ **Servo** - for controlling servo motors
- ✦ **SPI** - for communicating with devices using the Serial Peripheral Interface (SPI) Bus
- ✦ **SoftwareSerial** - for serial communication on any digital pins. Version 1.0 and later of Arduino incorporate Mikal Hart's NewSoftSerial library as SoftwareSerial.
- ✦ **Stepper** - for controlling stepper motors
- ✦ **WiFi** - for connecting to the internet using the Arduino WiFi shield
- ✦ **Wire** - Two Wire Interface (TWI/I2C) for sending and receiving data over a net of devices or sensors.

ARI Sezione di Ivrea

Serate tecniche 2012

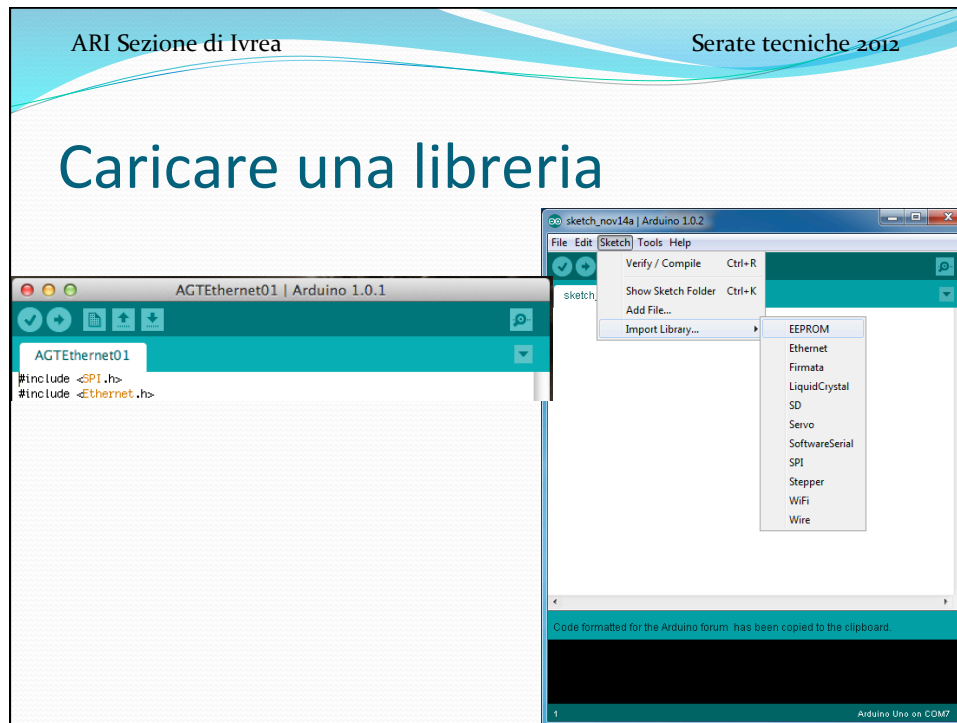
Librerie standard

USB Libraries (Leonardo and Due only)

- ✦ **Keyboard** - Send keystrokes to an attached computer.
- ✦ **Mouse** - Control cursor movement on a connected computer.

Due Only Libraries

- ✦ **Audio** - Play audio files from a SD card.
- ✦ **Scheduler** - Manage multiple non-blocking tasks.
- ✦ **USBHost** - Communicate with USB peripherals like mice and keyboards.



ARI Sezione di Ivrea Serate tecniche 2012

Librerie di terze parti

- Le librerie sono tipicamente distribuite come cartelle compresse.
- Il nome della cartella è il nome della libreria
- Contengono due “gruppi” di file:
 - Obbligatori
 - .cpp e .h
 - Facoltativi
 - Keywords.txt, esempi e file aggiuntivi necessari alla libreria

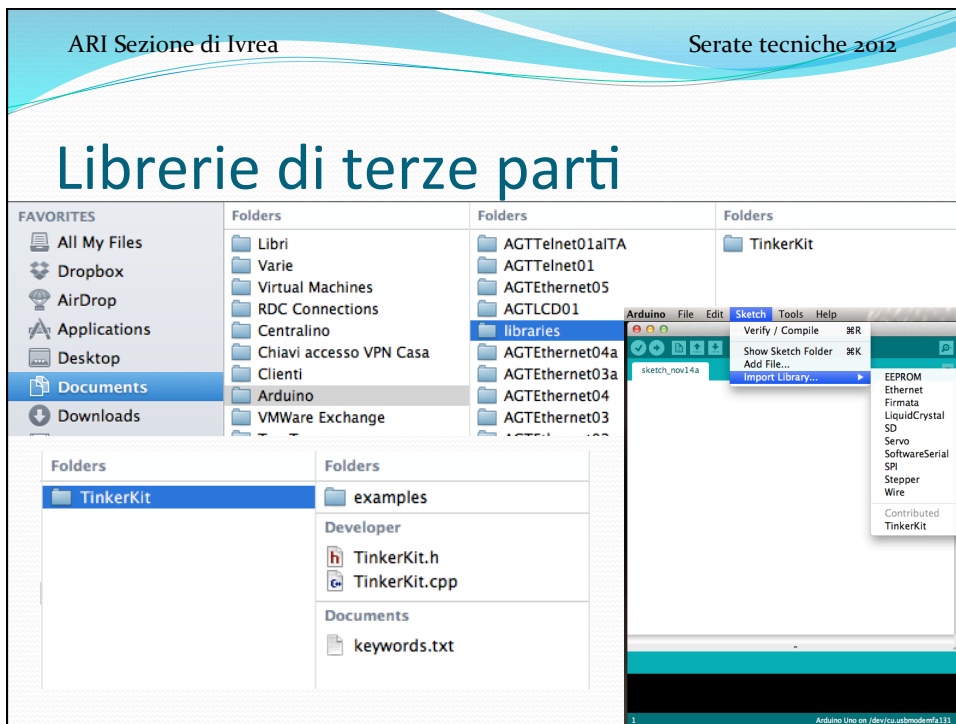
ARI Sezione di Ivrea Serate tecniche 2012

Installazione di librerie di terze parti

- Chiudere l'ADE
- Estrarre il file compresso. Se non viene generata una cartella mettere i file estratti in una cartella con il nome della libreria.
- Copiare la cartella sotto la cartella delle librerie
 - In Windows: "My Documents\Arduino\libraries"
 - In OSX: "Documents/Arduino/libraries"
 - Su Linux... dipende!
- Riavviare l'ADE!

ARI Sezione di Ivrea Serate tecniche 2012

Librerie di terze parti



The screenshot displays the Arduino IDE interface. In the background, a file explorer window shows the 'libraries' folder within the Arduino directory. The foreground shows the Arduino IDE's 'Sketch' menu with the 'Import Library...' option selected. A dropdown menu lists various libraries available for import, including EEPROM, Ethernet, Firmata, LiquidCrystal, SD, Servo, SoftwareSerial, SPI, Stepper, Wire, and TinkerKit. The TinkerKit library is highlighted at the bottom of the list.

Librerie personali

- Una libreria personale è composta da due file
 - Un file di header (con estensione.h): contiene le definizioni della libreria.
 - Un file sorgente (con estensione .cpp): contiene il codice sorgente dei metodi della libreria.
 - Se vogliamo che l'ADE identifichi le nostre "definizioni" dobbiamo creare anche il file keywords.txt.
- Fare riferimento alla documentazione ufficiale per il modo corretto di "costruirle":
<http://arduino.cc/en/Hacking/LibraryTutorial>

So...

- Primo: grazie 1000 per la pazienza
- Secondo: questa prima lezione è l'unica prettamente teorica. Scusate, ma era necessario!
- Terzo: vi è piaciuto? Cosa cambiereste?

Next Time: Semplici input

- Pulsanti
- Debounce
- Determinare la durata della pressione di un pulsante
- Usare un Keypad
- Leggere un valore analogico
 - Leggere più di 6 valori
 - Leggere più di 5 volt

Next Time: materiale

- BreadBoard
- 2 pulsanti n.a.
- 2 resistenze da 330 e 2 da 10K
- 1 potenziometro lineare da 10K
- Un tastierino 4x4
- fili rigidi da breadboard

Compiti a casa

- Pensavate davvero di scamparla...
- Leggete la versione “integrale” delle slide.
- Segnatevi qualsiasi cosa non sia chiara.
- Realizzare un piccolo sketch che faccia lampeggiare in CW il vostro nominativo!