UNIVERSITÀ DEL PIEMONTE ORIENTALE

Arduino: introduzione e primi progetti

Giorgio Leonardi, Paola Giannini

giorgio.leonardi@uniupo.it

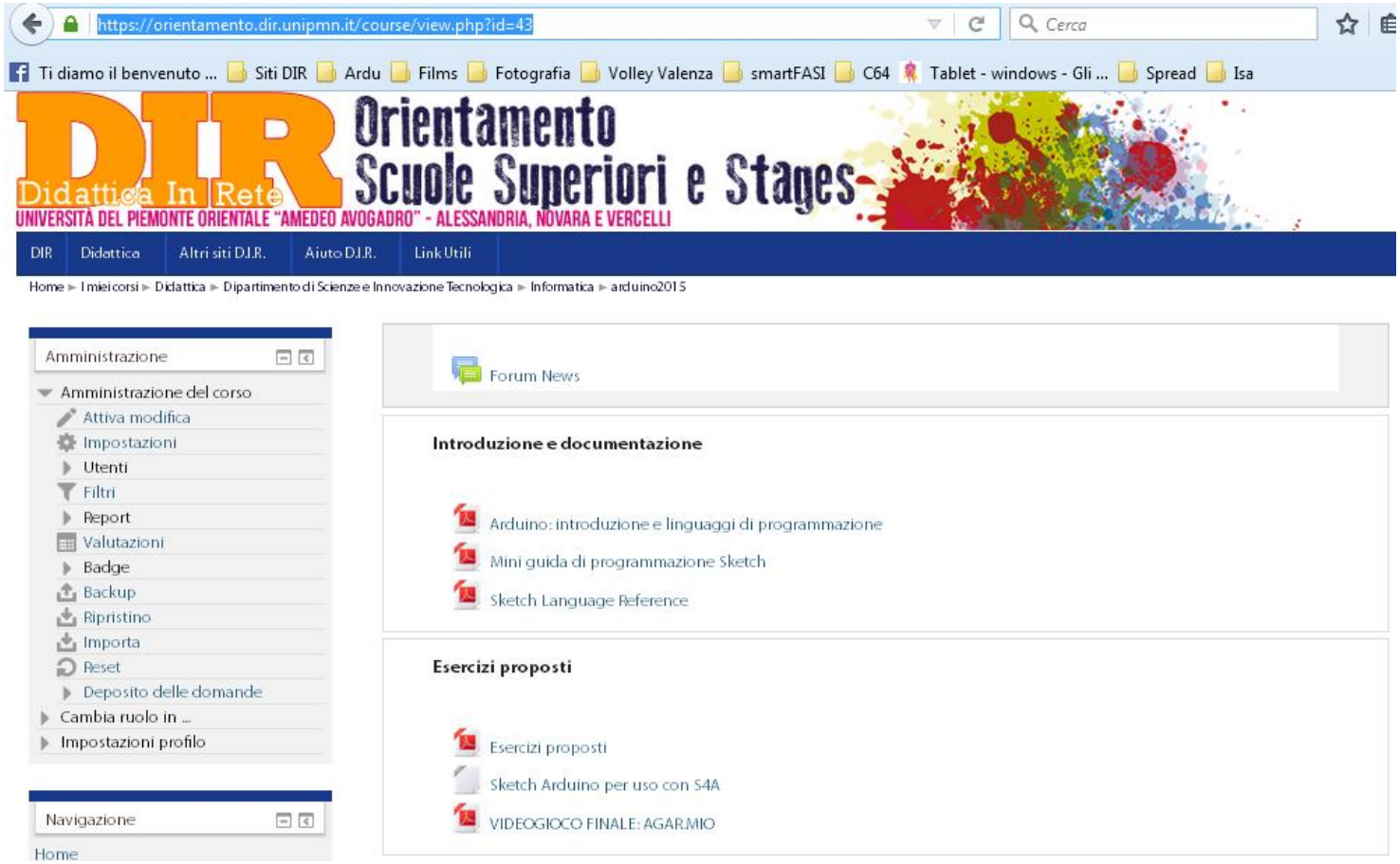
paola.giannini@uniupo.it

Collegatevi al sito:

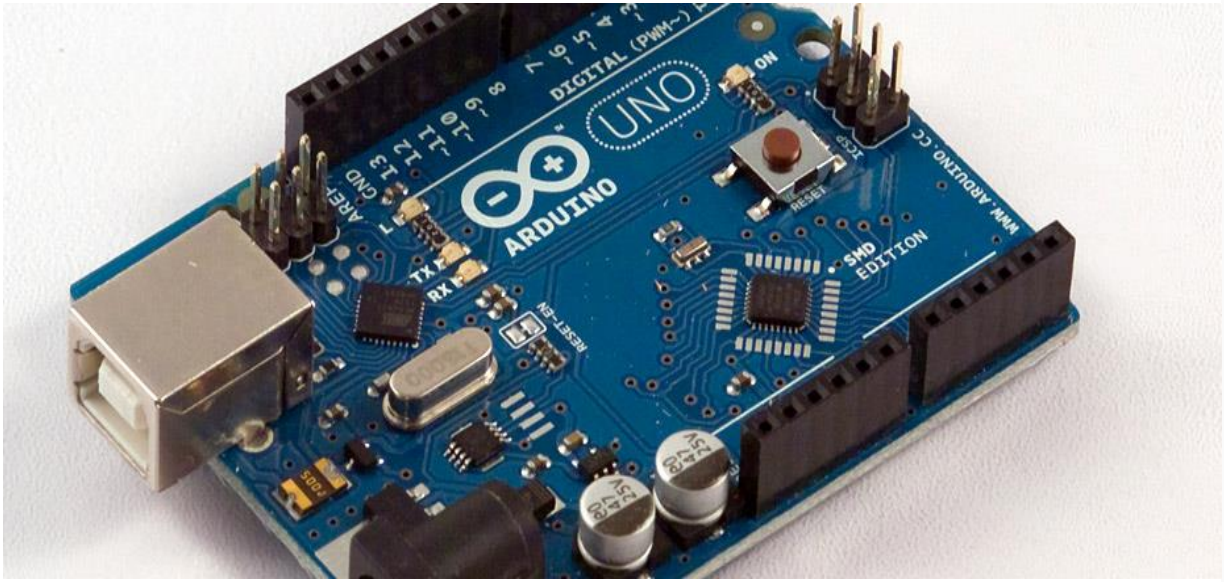
https://orientamento.dir.unipmn.it/course/view.php?id=43

oppure: https://orientamento.dir.unipmn.it → informatica → arduino2022

Chiave di iscrizione: arduino2022



Cos’è Arduino



«Arduino è una piattaforma di prototipazione elettronica open-source basata su hardware e software flessibili e di facile utilizzo.

E’ pensato per artisti, designer, hobbisti e chiunque sia interessato a creare oggetti o ambienti interattivi. »

Cos'è Arduino

- **Hardware a basso costo e bassa potenza:**
  - **Micro-controllore: versione semplificata di un microprocessore**
  - **Connessioni per i dispositivi controllati**
  - **Connessioni (USB/seriale) per la sua programmazione**

- **Esegue continuamente il firmware che viene caricato:**
  - **Riceve i dati da elaborare da sensori collegati in input e letti a intervalli regolari**
  - **Sulla base dei valori dell'input, attiva i dispositivi di output (attuatori) per eseguire le azioni opportune**

- **Sensori e attuatori sono detti «trasduttori»**

Una parte del mondo di Arduino

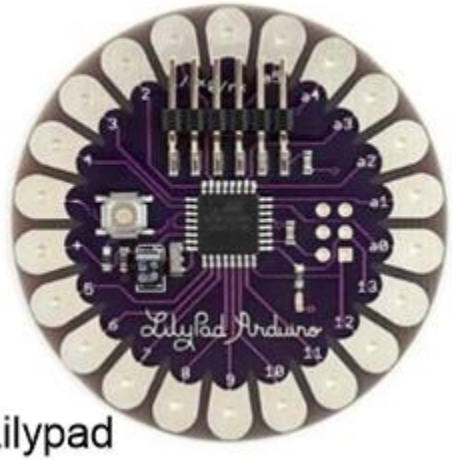
Arduino Uno

ARDUINO

Arduino Pro Mini

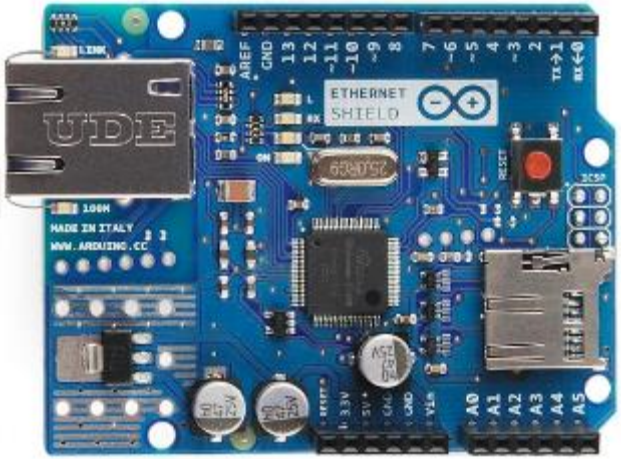
Arduino Nano

Arduino BT (Bluetooth)



Arduino Lilypad

Connessione con il mondo!

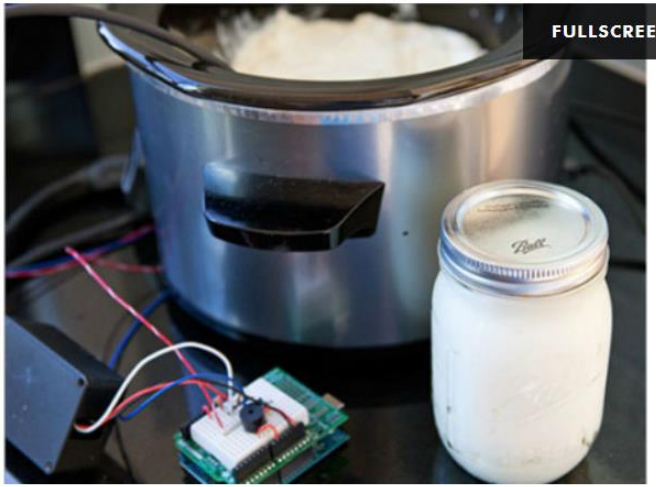


Net shield con estensione di memoria MicroSD

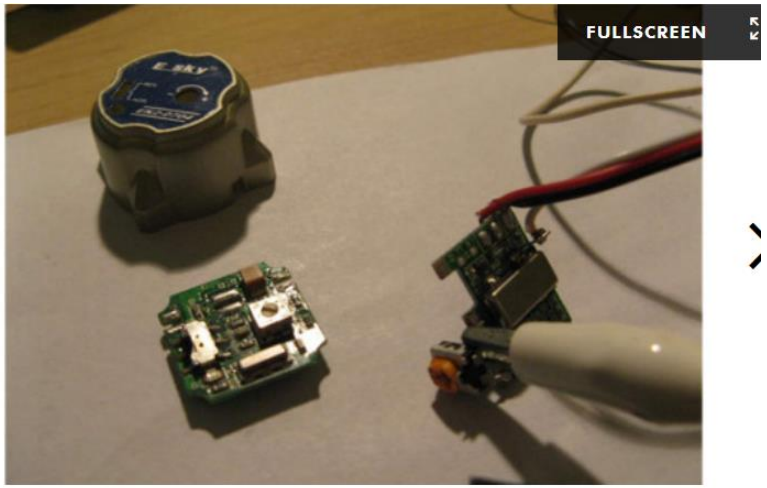
Modulo wireless – Portata di 100 metri

Modulo wireless pro – Portata di 1 miglio = 1.6 Km

Progetti con cuore Arduino



07/49 Yobot Arduino può anche fare lo yogurt. O meglio, controllare la temperatura durante il lungo processo di fermentazione

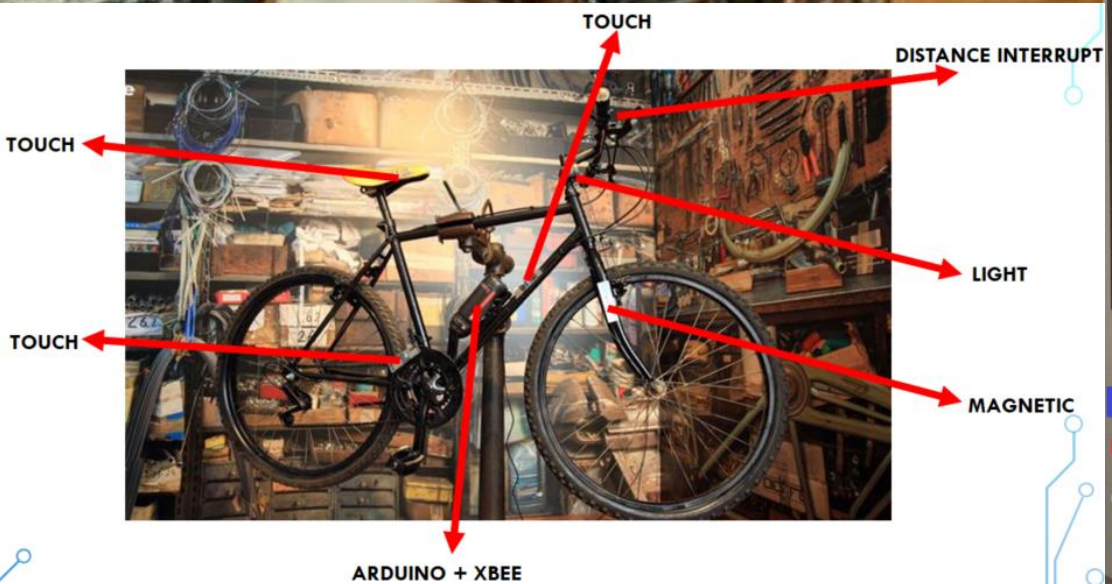


35/49 Arduino 3DOF Head Tracker Utilizzando Arduino è possibile anche tracciare il movimento della testa di un giocatore per proiettarlo in divertenti sparatutto in prima persona

08/49 DIYdrones ArduPilot è l'autopilota Arduino in grado di controllare il volo di droni fatti a mano in collaborazione con ArduPlane e AeduCopter

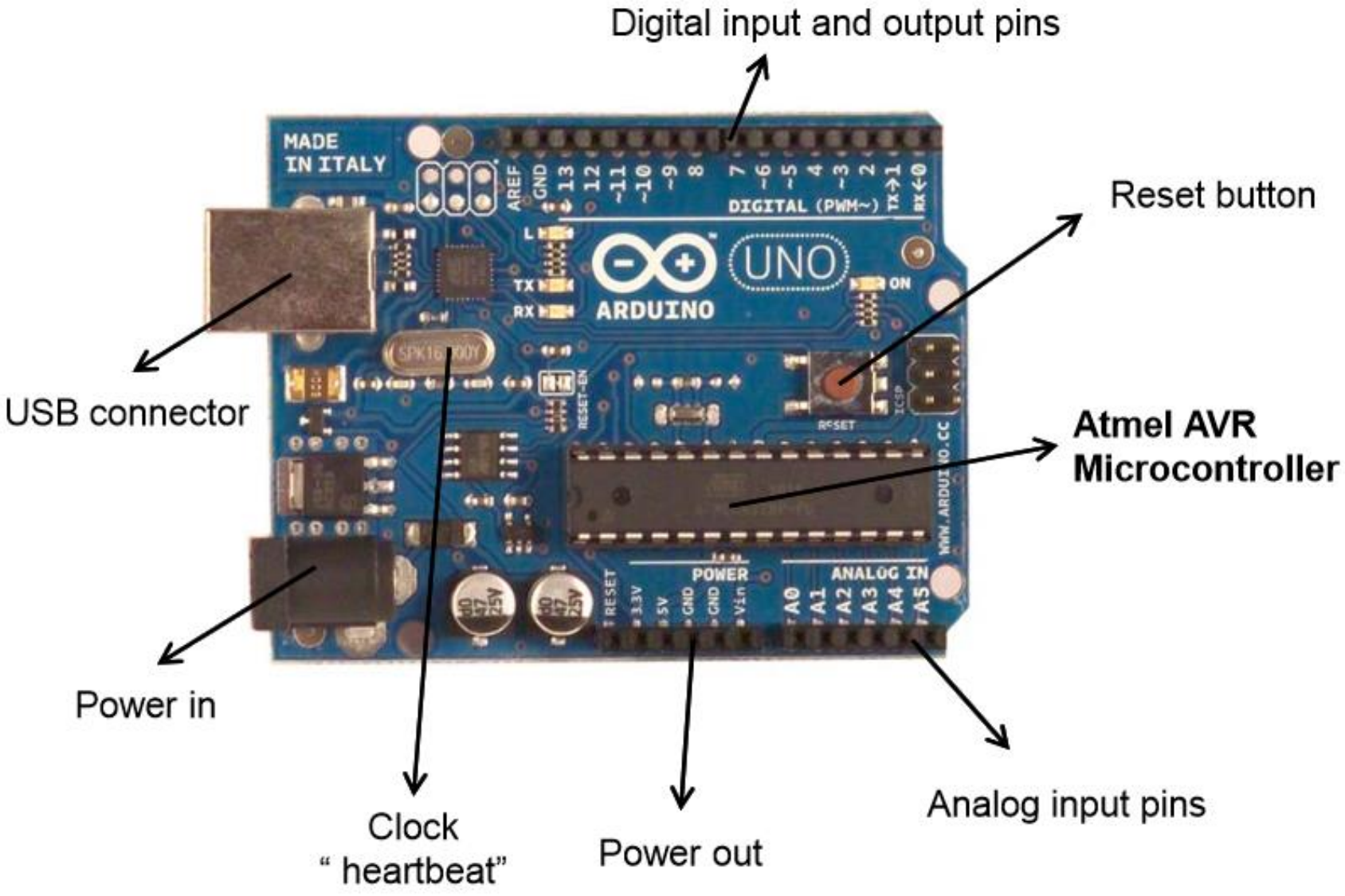
01/49 Makerbot L'azienda che distribuisce le più famose macchine da stampa 3D utilizza Arduino quale componente elettronico


Camera di Commercio Alessandria
Città di Alessandria
ALESSANDRIA
CITTA' DELLE BICICLETTE
una storia mai raccontata sulle prime pedalate in italia
La mostra "Alessandria città delle biciclette" e Università del Piemonte Orientale
Tecnologie multimediali a favore dei nonvedenti
11 Maggio alle ore 17,00
Alessandria, Camera di Commercio, Via Vochieri 58
Presentazione dei progetti multimediali ideati e progettati da Abilitando con il Dipartimento di Scienze e Innovazione Tecnologica
www.palazzomonferrato.it
www.facebook.com/cittadellebiciclette
UPO
UNIVERSITÀ DEL PIEMONTE ORIENTALE
Abilitando
Unione Italiana dei Ciechi e degli Ipovedenti ONLUS
irifor
RICERCA, FORMAZIONE RIABILITAZIONE per la disabilità visiva

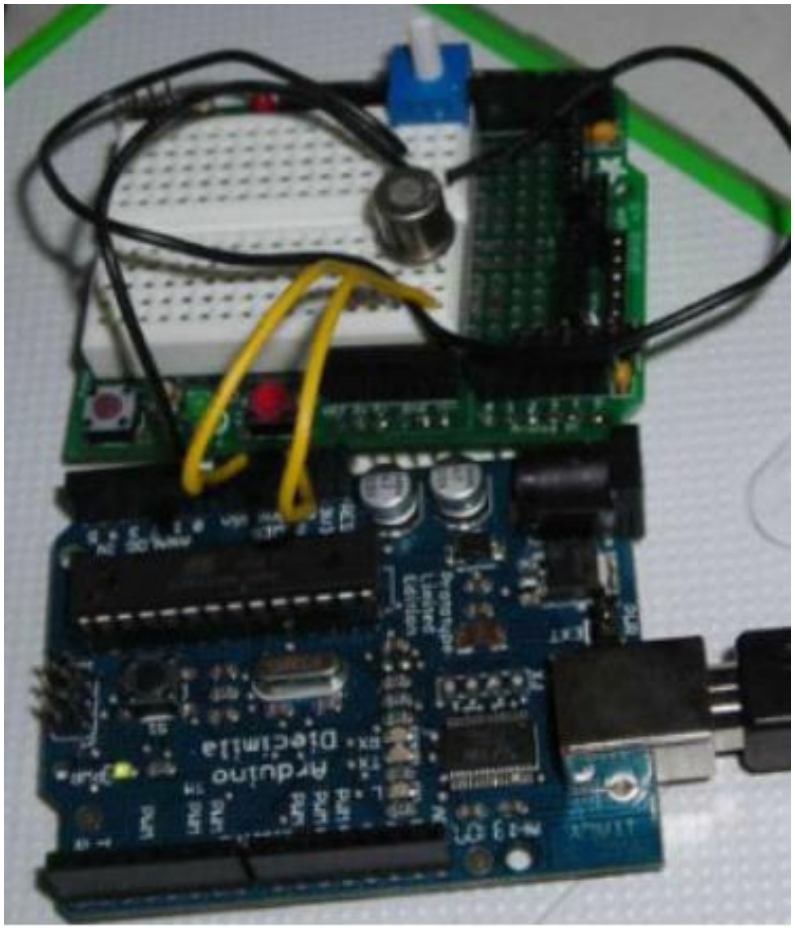

TOUCH
DISTANCE INTERRUPT
TOUCH
LIGHT
TOUCH
MAGNETIC
ARDUINO + XBEE


Il giornalismo sport
Eliso Rivera
ARDUINO
MODULO XBEE
LED RGB
TOUCH

I/O di Arduino uno



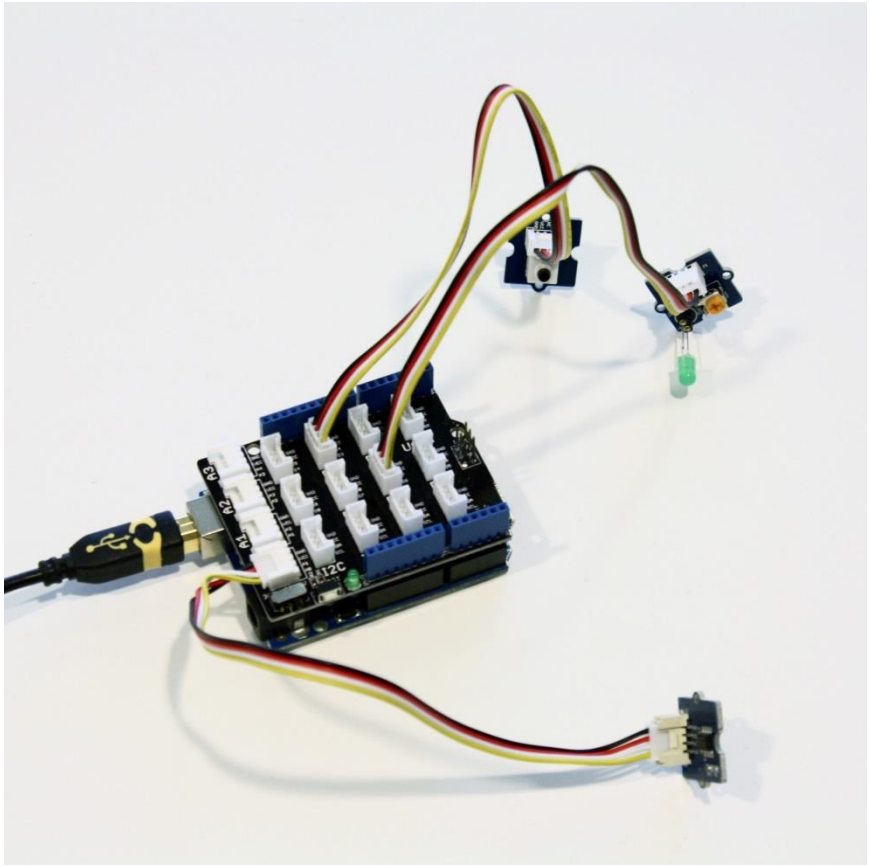
Collegamento con breadboard



Sfruttando i pin di ingresso e uscita, è possibile collegarsi a trasduttori montati su una breadboard

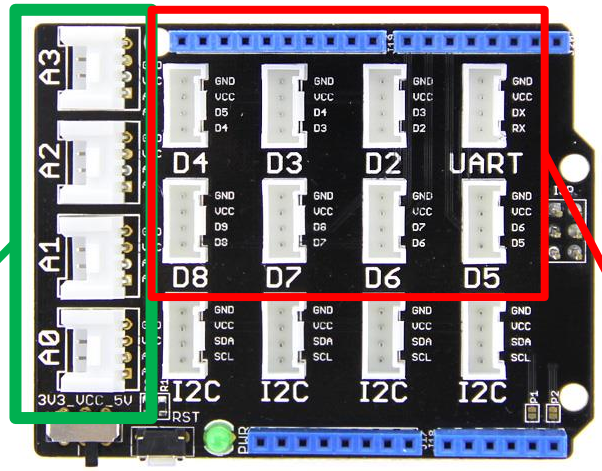
Posso usare tutti i trasduttori acquistabili in un negozio di elettronica, ma ho necessità di nozioni di elettronica per il dimensionamento dei componenti necessari al collegamento, come regolazione dell'alimentazione e calcolo delle resistenze

Grove kit



Il kit Grove utilizza uno «shield» che, installato sui pin di Arduino, permette il collegamento diretto con trasduttori compatibili, senza bisogno di conoscenze di elettronica per creare sistemi di controllo anche complessi.

Pin utilizzabili con Grove



- **Ingressi**
  - **A0 → A3: sono SOLO input ANALOGICI!**
  - **D2 → D8: sono SOLO input DIGITALI!**
- **Uscite**
  - **D2 → D8: TUTTE uscite DIGITALI**
  - **D3, D5 e D6: sono anche uscite ANALOGIGHE**

- **I2C: ingressi/uscite complesse, non utilizzabili nativamente con Snap4Arduino (per ora)**

Grove kit per Arduino

https://www.seeedstudio.com/category/Kits-for-Grove-c-28.html

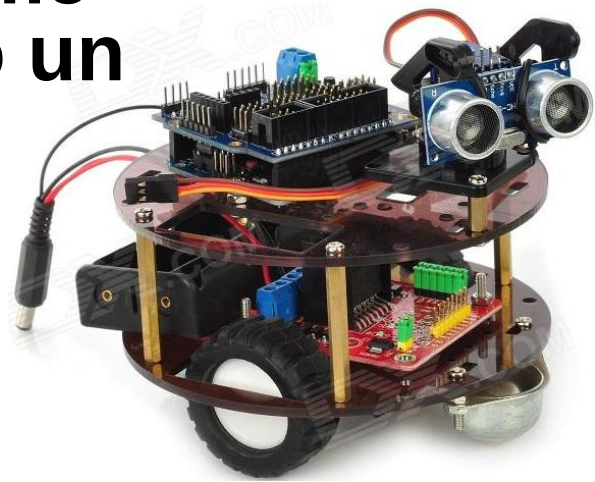
- **Nuovo kit con shield e più di 200 sensori/attuatori preconfigurati, tra cui:**
  - **Sensori, come lettura co2, polvere, infrarossi, umidità, codici RF-ID, riconoscitori vocali e di gesti, ecc.**
  - **Attuatori, come motori, speaker, monitor LCD, ventole, riproduttori MP3, ecc.**

Modi di utilizzo di Arduino

- **Arduino può essere utilizzato principalmente in due modi:**

1. **Come micro-controllore indipendente: Una volta caricato il firmware, Arduino può essere scollegato dal PC e alimentato a batteria o dalla rete elettrica.**

Arduino svolgerà il suo lavoro come dispositivo autonomo, eseguendo un ciclo infinito per il pilotaggio continuo dei suoi trasduttori.



Modi di utilizzo di Arduino

- **Arduino può essere utilizzato principalmente in due modi:**

2. **Come dispositivo per PC. In questa configurazione, Arduino si offre come «ponte» per l'uso dei trasduttori da parte del PC.**

La logica del programma è eseguita dal PC, mentre Arduino offre il supporto per l'accesso ai trasduttori

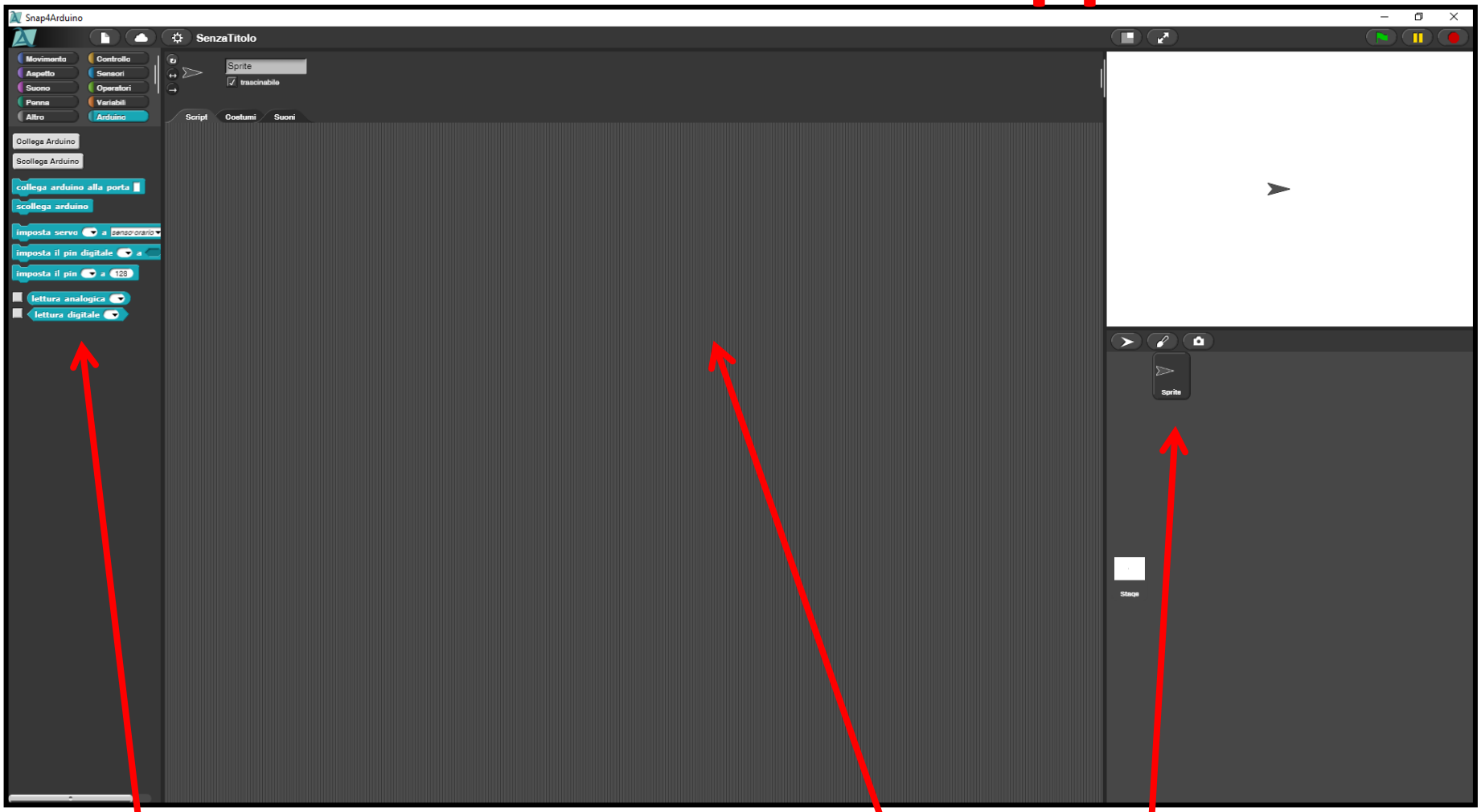
Snap!4Arduino

http://snap4arduino.rocks/

- **Ambiente di sviluppo implementato dall'Edutec research group from Citilab (Cornellà, Barcelona),**

- **Incorpora il linguaggio Snap! Sviluppato dall'Università di Berkley, estendendola con blocchi specifici per la gestione dell'I/O da e verso i trasduttori collegati**

Ambiente di sviluppo



Nuovi blocchi per gestione dei sensori (sezione «Arduino»)

Area degli script

Area degli Sprites

Ambiente di sviluppo

- **Grazie alla nuova sezione «Arduino», UNO sprite (ad eccezione dello Stage), può essere collegato ai sensori e agli attuatori.**
  - **SENSORI: la lettura dei valori in input può essere «analogica», oppure «digitale»:**
    - **Lettura «analogica» (A0-A3): il sensore restituirà un valore compreso tra 0 (minimo) e 1023 (massimo)**
    - **Lettura «digitale» (D2-D8): il sensore restituirà un valore TRUE (input attivato) o FALSE (input disattivato)**
  - **Attuatori: si può attivare un'azione in due modi:**
    - **Attivazione «analogica» (D3,D5 e D6): l'attuatore può essere spento (valore 0); acceso alla massima potenza (valore 255) o acceso gradualmente a qualsiasi valore intermedio.**
    - **Attivazione «digitale» (D2-D8): l'attuatore può essere solo spento (valore «false») o acceso alla max potenza (valore «true»)**

Connessione di Arduino

- **Prima di poter usare i sensori di Arduino, la nostra board deve essere «collegata» all'ambiente di sviluppo.**

- **Solo dopo il collegamento avremo accesso a tutti i sensori e attuatori.**

- **Al termine del lavoro (opzionale), o se dovessimo cambiare la board attualmente inserita, possiamo scollegarla con il pulsante «scollega»**

- **Collegamento e scollegamento possono essere fatte «al volo» da un programma in esecuzione, per «switchare» tra due o più board collegate contemporaneamente**

Blocchi per i sensori

- **Questi blocchi permettono al nostro programma Snap! di accedere alle letture dei sensori e alle scritture (comandi) agli attuatori.**

- **Lettura digitale:** Lettura del sensore digitale collegato al pin indicato.

- **Lettura analogica:** Lettura del sensore analogico collegato al pin indicato.

- **Scrittura digitale:** attiva o disattiva l'attuatore indicato con impostazione «true» o «false».

- **Scrittura analogica:** attiva o disattiva l'attuatore indicato con intensità da 0 a 255.

- **Controllo servo-motori:** imposta direzione e velocità di rotazione di un servo-motore.

Progetto 1: Cicli e LED analogico/digitale

a) Accensione/spegimento di led in digitale
b) Gestire gradazioni di luminosità in analogico
c) Blink di un led con effetto on/off
d) Blink con effetto fade

Progetto 1 c) e d): Algoritmo

1 c)

- **All'avvio del programma:**
  - **Per sempre, eseguo:**
    - **Imposto lo stato del LED ad ACCESO**
    - **Attendo 1 secondo**
    - **Imposto lo stato del LED a SPENTO**
    - **Attendo 1 secondo**

Come posso cambiare l'effetto on/off?

1 d)

- **All'avvio del programma:**
  - **Per sempre, eseguo:**
    - **Per I da 0 a 255:**
      - **Imposto la luminosità del LED al valore I**
    - **Per I da 255 a 0:**
      - **Imposto la luminosità del LED al valore I**

Come posso variare la velocità?

Progetto 2: concorrenza

- **Collegare ad Arduino due LED: uno su un'uscita digitale (ad esempio D2) e uno su un'uscita analogica (ad esempio D3)**
- **Creare, nello stesso sprite collegato ad Arduino, due script che partono contemporaneamente:**
  - **Il primo fa eseguire, in un ciclo «per sempre», il blink del led collegato a D2 (esercizio 1c)**
  - **Il secondo fa eseguire, in un ciclo «per sempre», l'effetto fade al led collegato a D3 (esercizio 1d)**

- **Alla partenza del programma, i due led sono controllati in maniera indipendente da due script eseguiti in concorrenza**

Progetto 3: Pulsante+LED, flags e comunicazione tra script

a) **Accendere il LED alla pressione del pulsante; spegnerlo al rilascio del pulsante**

b) **Accendere o spegnere alternativamente il LED ad ogni pressione del pulsante (acceso – spento – acceso – spento)**

Passo a): Algoritmo

- All’avvio del programma:
  - Spengo fisicamente il LED
  - Per sempre, esegui:
    - Se il pulsante è premuto:
      - Accendo fisicamente il LED
    - Altrimenti:
      - Spengo fisicamente il LED

Passo b): Algoritmo e comunicazione tra script

- **Uso 2 script: uno per gestire la pressione del pulsante, e uno per gestire lo stato del led**
- **Uso un «Flag» per ricordarmi lo stato del LED (è acceso o spento, in questo momento?)**

Script 1: inizializzazione LED e gestione del pulsante

- All'avvio del programma:
  - Imposto lo stato del LED a SPENTO
  - Spengo fisicamente il LED
  - Per sempre, esegui:
    - Se il pulsante è premuto:
      - Invio un messaggio «Pulsante premuto»

Script 2: gestione dello stato del LED

- Ogni volta in cui ricevo il messaggio: «Pulsante premuto», eseguo:
  - Se lo stato del LED è ACCESO:
    - Spengo fisicamente il LED
    - Imposto stato del LED a SPENTO
  - Altrimenti:
    - Accendo fisicamente il LED
    - Imposto stato del LED a ACCESO

Passo b): Problema pressione pulsante

- **Il ciclo «Per sempre» è talmente veloce che, quando premiamo il pulsante, il ciclo continua a vederlo premuto più volte, prima che noi riusciamo a togliere il dito**

Script 1: inizializzazione LED e gestione del pulsante

- All'avvio del programma:
  - Imposto lo stato del LED a SPENTO
  - Spengo fisicamente il LED
  - Per sempre, esegui:
    - Se il pulsante è premuto:
      - **Attendo fino a quando il pulsante è rilasciato**
      - Invio un messaggio «Pulsante premuto»

Script 2: gestione dello stato del LED

- Ogni volta in cui ricevo il messaggio: «Pulsante premuto», eseguo:
  - Se lo stato del LED è ACCESO:
    - Spengo fisicamente il LED
    - Imposto stato del LED a SPENTO
  - Altrimenti:
    - Accendo fisicamente il LED
    - Imposto stato del LED a ACCESO

Progetto 4: LED+Potenziometro, lettura analogica e nuovo blocco

- **Dato un potenziometro, illuminare un led proporzionalmente al valore del potenziometro**

- **Attenzione: i led hanno un valore di luminosità impostabile da 0 a 255, mentre i potenziometri hanno un range di valori da 0 a 1023. E' necessario adottare una proporzione tra i due range**

Progetto 4: Algoritmo

- **All'avvio del programma:**
  - **Per sempre, esegui:**
    - **Leggo la posizione del potenziometro P**
    - **Calcolo la luminosità relativa L del LED:**

      L : 255 = P : 1023 → L = ⌊(P * 255) / 1023⌋
    - **Imposto fisicamente la luminosità del LED al valore L**

- **La proporzione può essere calcolata creandoci un nuovo blocco!**

Progetto 5: LED+Pusante, uso del cronometro di sistema

- **Dato un LED e un pulsante (o un sensore «touch»), simulare il comportamento della luce temporizzata delle scale di un palazzo:**
  - **Il LED parte spento;**
  - **Alla pressione del pulsante, il LED si accende, e resta acceso fino a che:**
    - **Non si preme nuovamente il pulsante, oppure**
    - **Passano 10 secondi e non è stato ancora premuto il pulsante**

Progetto 5: Algoritmo

- **All'avvio del programma:**
  - **Spengo fisicamente il LED**
  - **Per sempre, esegui:**
    - **Se il pulsante è premuto e rilasciato:**
      - **Azzero il cronometro**
      - **Accendo fisicamente il LED**
      - **Attendo fino a quando (il pulsante è premuto e rilasciato), oppure (il cronometro ha contato 10 secondi)**
      - **Spengo fisicamente il LED**

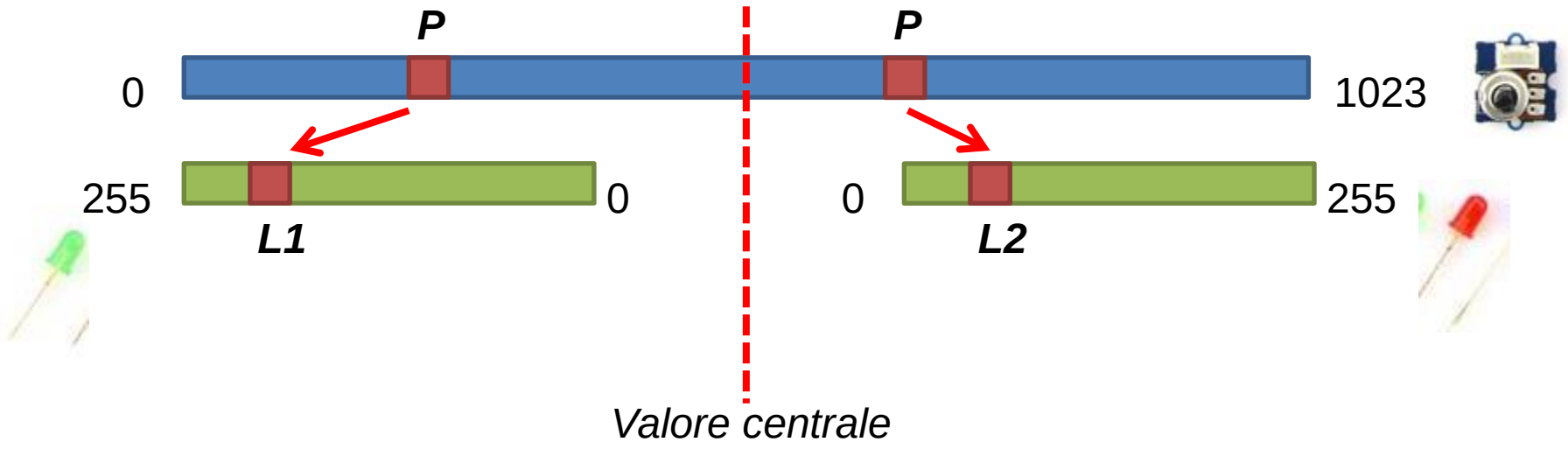
ATTENZIONE: Per la capire se il pulsante è premuto, posso crearmi un nuovo blocco che restituisce FALSE se il pulsante non è premuto, e TRUE se il pulsante è stato premuto e rilasciato!

Progetto 6

- **Dati due led e un potenziometro (o un joystick):**
- **Stabilite il valore «centrale» del potenziometro o del joystick: il valore, cioè, che divide il suo range a metà**

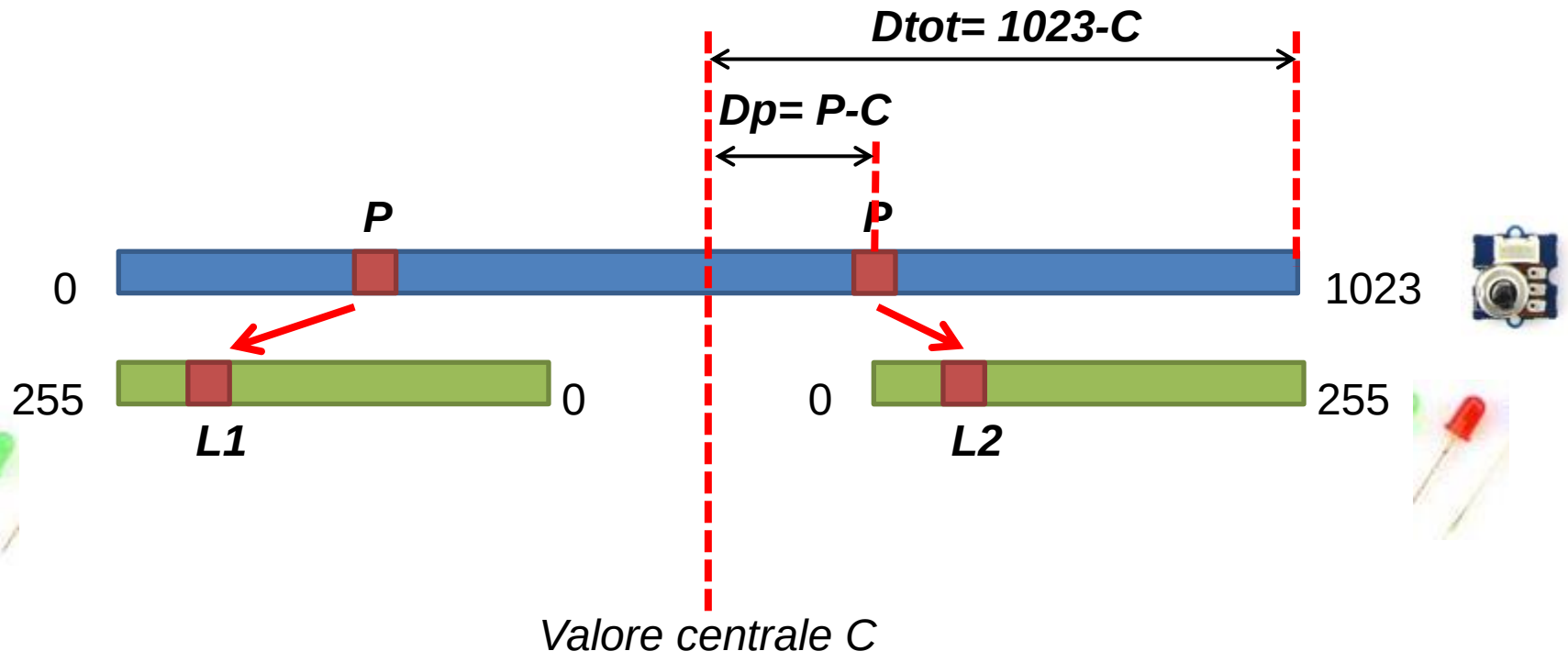
- **Illuminate uno solo dei due led in questo modo:**
  - **se il potenziometro è al centro, i due led sono spenti;**
  - **Se si muove il potenziometro a «sinistra», il primo led si illumina proporzionalmente alla sua posizione; il secondo led rimane spento**
  - **Se si muove il potenziometro a «destra», il primo led resta spento, mentre il secondo si illumina proporzionalmente alla posizione del potenziometro**

Progetto 6: suggerimenti



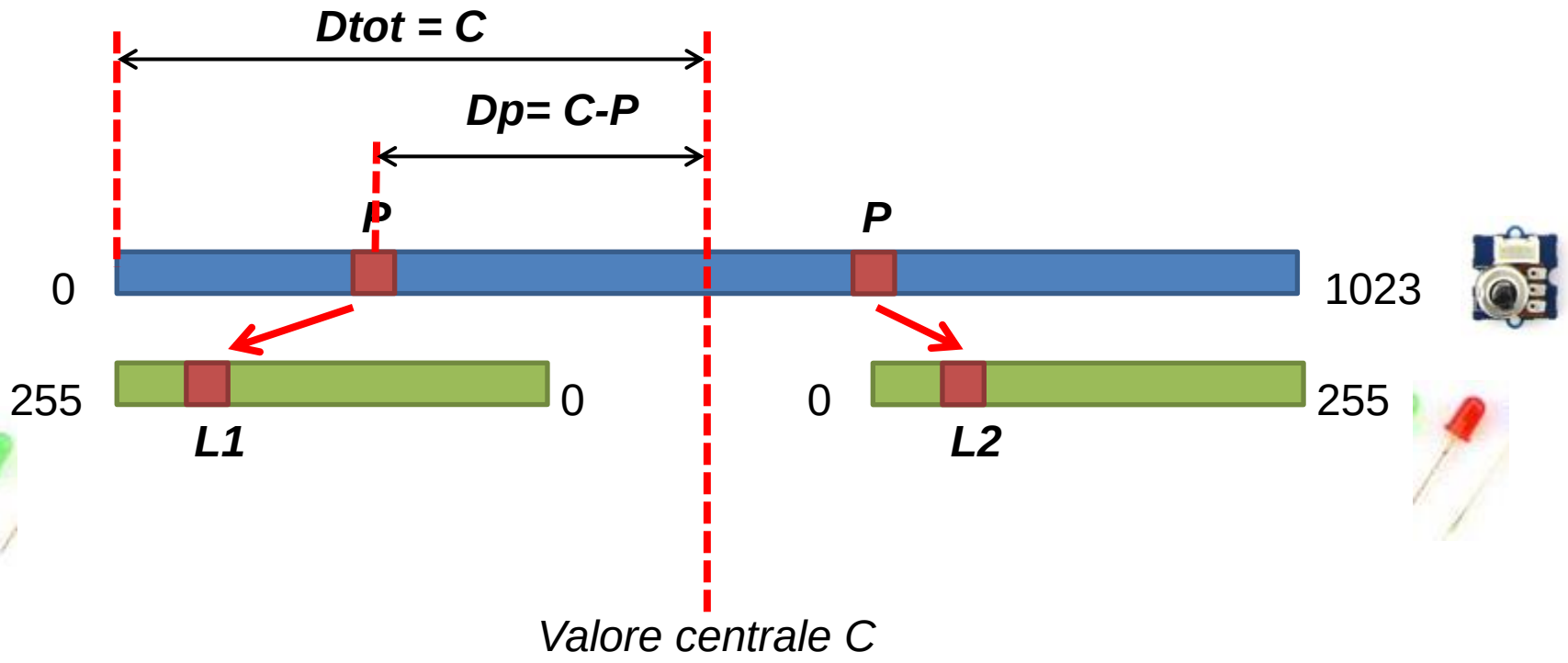
- **Se il joystick è inclinato a destra, L1 dev'essere a 0, mentre L2 è il risultato di una proporzione**
- **Se il joystick è inclinato a sinistra, L2 dev'essere a 0, mentre L1 è il risultato di una proporzione (inversa!)**

Progetto 6: soluzione



- **Se il joystick è inclinato a destra:**
  - **DP : DTOT = L2 : 255**
  - **L2 = (DP / DTOT) * 255 → [(P-C)/(1023-C)] * 255**

Progetto 6: soluzione



- **Se il joystick è inclinato a sinistra:**
  - **DP : DTOT = L1 : 255**
  - **L1 = (DP / DTOT) * 255 → [(C-P)/(C)] * 255**

Progetto 6: soluzione

- **Identificare il punto centrale C**
- **Per sempre:**
  - **Leggo il valore del potenziometro P**
  - **Se P > C:**
    - **L1= 0**
    - **L2 = 255 * (P-C)/(MAX-C)**
  - **Altrimenti:**
    - **L2= 0**
    - **L1= 255 * (C-P)/(C – MIN)**
  - **Assegno al led1 il livello di luminosità L1**
  - **Assegno al led2 il livello di luminosità L2**

Progetto 7: Monitoraggio ambientale, uso di più sprites, liste e penna

- **Dato un fotoresistore (o un termistore) e un led, effettuare il campionamento della luminosità (o della temperatura) una volta al secondo**

- **Ogni volta in cui si prende un campione:**
  - **Far lampeggiare il led**
  - **Memorizzare il valore campionato in una coda dei campionamenti, inizialmente vuota**
  - **Visualizzare il campione sullo schermo**

Progetto 7: Algoritmo

- **All'avvio del programma:**
  - **Creo una lista «Campionamenti» e la inizializzo come lista vuota**
  - **Per sempre, eseguo:**
    - **Assegno alla variabile «Valore corrente» la lettura del valore di luminosità (o di temperatura)**
    - **Aggiungo il valore appena letto in coda alla lista «Campionamenti»**
    - **Attendo un secondo, all'interno del quale uso 0.2 secondi per il lampeggio del led**

Progetto 7 (b)

- **Un passo in più: generare il grafico dei campionamenti**

- **Ogni volta in cui si prende e si memorizza un campione:**
  - **Tracciare una linea verticale sullo schermo, proporzionale al valore campionato**
  - **La sequenza di questi valori formerà un istogramma dei campionamenti**

Progetto 7 b): Algoritmo

Sprite di controllo:

- **All'avvio del programma:**
  - **Creo una lista «Campionamenti» e la inizializzo come lista vuota**
  - **Per sempre, eseguo:**
    - **[…]**
    - **Mando un messaggio: «Campione disponibile» allo sprite della penna**

- **Sprite «penna»:**

- **All'avvio del programma, mi posiziono nell'angolo in basso a sinistra dello Stage e pulisco lo schermo**
- **PEN= dimensione della penna (larghezza di un'asta dell'istogramma)**

- **Quando ricevo un messaggio: «Campione disponibile»:**
  - **Calcolo la lunghezza dell'asta corrente:**

    campione corrente : 1023 = asta : 360

  - **Punto lo sprite in direzione: «SU»**
  - **Aziono la penna per tracciare**
  - **Faccio un numero di passi pari ad «asta»**
  - **Disaziono la penna, per smettere di tracciare**
  - **Sposto lo sprite di PEN passi a destra, e poi in basso, nello stage**

Progetto 7 (c)

- **Un passo in più: generare il grafico dei campionamenti**

- **Al raggiungimento della fine dello schermo da parte della matita:**
  - **Cancellare lo schermo e riprendere i campionamenti**

Progetto 7 c): Algoritmo

Sprite di controllo:

- **All'avvio del programma:**
  - **Creo una lista «Campionamenti» e la inizializzo come lista vuota**
  - **Per sempre, eseguo:**
    - **[…]**
    - **Mando un messaggio: «Campione disponibile» allo sprite della penna**

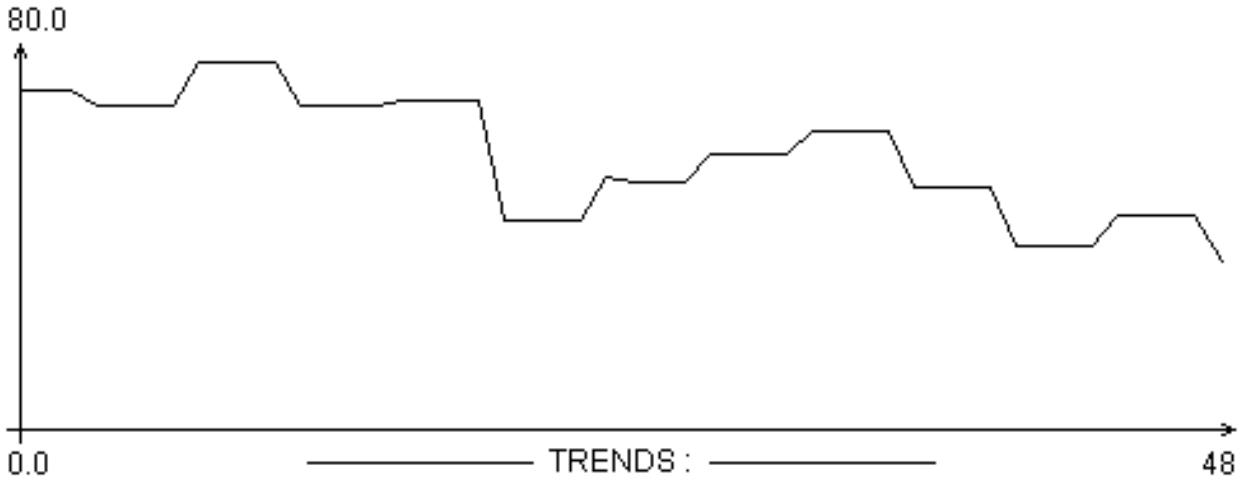
- **Sprite «penna»:**

- **All'avvio del programma, mi posiziono nell'angolo in basso a sinistra dello Stage e pulisco lo schermo**

- **Quando ricevo un messaggio: «Campione disponibile»:**
  - **[…]**
  - **Se la posizione dello sprite è superiore alla massima ordinata dello stage, allora:**
    - **Pulisco lo schermo**
    - **Riporto lo sprite penna nell'angolo in basso a sinistra dello stage**

Astrazioni temporali: stati e trend

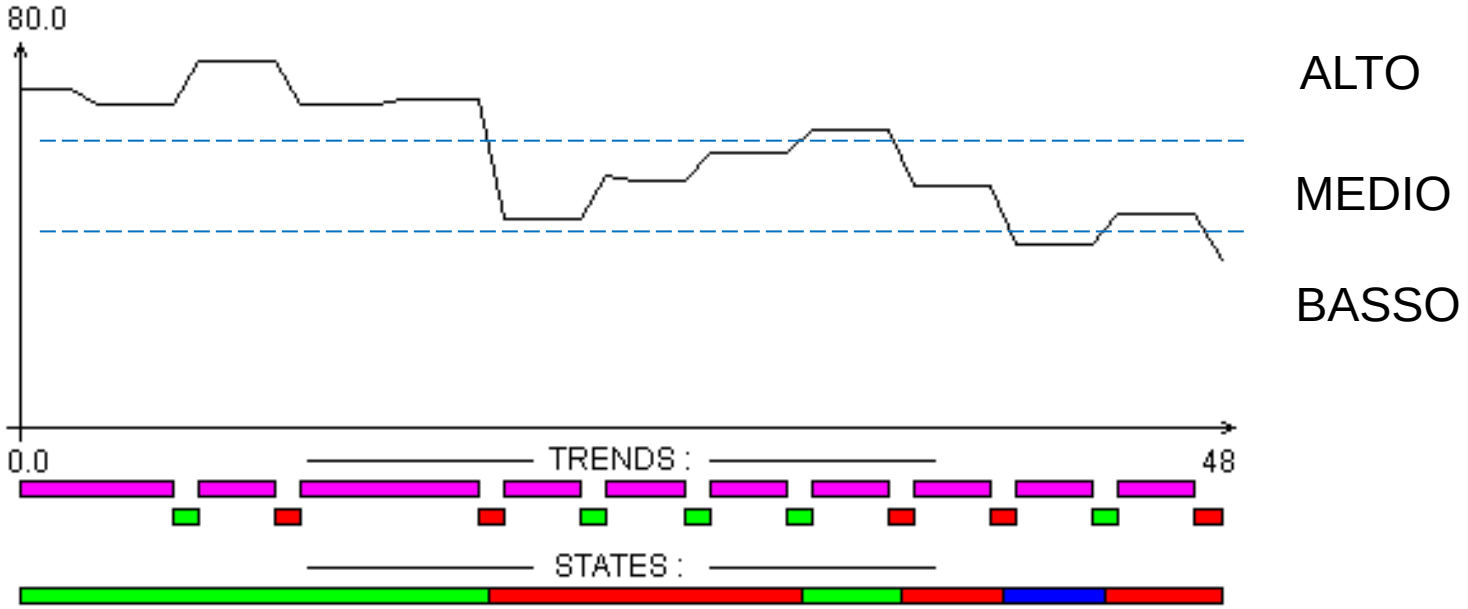
- **I campioni prelevati, e salvati nella lista, definiscono una serie temporale:**



- **Serie temporale: campionamento di un segnale a intervalli regolari**
  - **Nel nostro caso, un campione al secondo = 1 Hertz**
  - **La musica digitale è campionata 44.100 volte al secondo → 44.1 KHz**

Astrazioni temporali: stati e trend

- **Una serie può essere riassunta in sequenze di stati (livello del segnale, come ALTO o BASSO) e di trend (andamento: IN CRESCITA, IN DECRESCITA O STABILE):**



- **Stati: sono utili per lanciare allarmi (la temperatura del computer è troppo alta), o per far partire azioni (il terreno è troppo secco, quindi attivo l'irrigatore**
- **Trend: sono utili per verificare l'andamento del segnale: la febbre è in diminuzione, oppure la temperatura atmosferica si sta alzando**

Per il nostro progetto: trends

- **Calcoliamo il trend del segnale misurato, e lo mostriamo con uno sprite a forma di freccia, da piazzare un un angolo dello schermo:**
  - **Se il trend è stabile, la freccia punta a destra**
  - **Se il trend è in crescita, la freccia punta in alto (45°)**
  - **Se il trend è in diminuzione, la freccia punta in basso (135°)**

Calcolo del trend

- **La cosa più facile è quella di confrontare l'ultimo campione con il precedente, ma il sensore è instabile, quindi usiamo una strategia più furba:**
  - **Un campione, non è un valore «secco» («raw»), ma lo calcoliamo come media di esso e dei due valori precedenti**
  - **Questa media ce la calcola un blocco nuovo (campioneMedio), che prende in ingresso la lista e ne fa la media degli ultimi 3 campioni**

campioneMedio

- **Input: una lista L**
- **Output: un valore medio M**

- **Algoritmo:**
  - **LUN = lunghezza di L (numero di elementi in L)**
  - **Se LUN è 0, allora M = 0**
  - **Se LUN è 1, allora M = elemento 1 di L**
  - **Se LUN è 2, allora M = (elemento 1 di L + elemento 2 di L) /2**
  - **Se LUN è maggiore di 2, allora:**
    - **M = elemento LUN di L**
    - **M = M + elemento (LUN - 1) di L**
    - **M = M + elemento (LUN - 2) di L**
    - **M = M / 3**
  - **Risultato = M**

Rilevazione del trend: sprite freccia

- **Ogni volta in cui si prende e si memorizza un campione, mando due messaggi:**
  - **Il messaggio «Campione disponibile» allo sprite penna (e qui non cambia nulla, abbiamo già fatto tutto)**
  - **Un messaggio «Nuovo trend» allo sprite freccia**

- **Lo sprite freccia si attiva al ricevimento del messaggio «Nuovo trend», e:**
  - **Calcola il nuovo campione medio**
  - **Lo confronta con il campione medio precedente**
  - **A seconda della differenza, aggiusta l'angolo della freccia**

Progetto 7 d): Algoritmo

Sprite di controllo:

- **All'avvio del programma:**
  - **Creo una lista «Campionamenti» e la inizializzo come lista vuota**
  - **Per sempre, eseguo:**
    - **[…]**
    - **Mando un messaggio: «Campione disponibile» allo sprite della penna**
    - **Mando un messaggio: «Nuovo trend» allo sprite della freccia**

- **Sprite «freccia»:**

- **All'avvio del programma:**
  - **mi posiziono nell'angolo in alto a destra dello Stage**
  - **Pongo: mediaCorrente= 0, mediaPrecedente= 0**

- **Quando ricevo un messaggio: «Nuovo trend»:**
  - **mediaPrecedente= mediaCorrente**
  - **mediaCorrente= campioneMedio (Campionamenti)**
  - **Differenza= mediaCorrente – mediaPrecedente**
  - **Se (differenza > -3 e differenza < 3), allora punto in direzione 90° (trend stabile)**
  - **Se (Differenza > 2), allora punto in direzione 45° (trend crescita)**
  - **Se (Differenza < -2), allora punto in direzione 135° (trend decrescita)**