



Università degli Studi dell'Aquila  
Dipartimento di ingegneria e scienze dell'informazione e matematica



Corso di Laurea in  
Ingegneria Informatica

# Oscilloscopio digitale con Arduino e Android

**Relatore**  
Vincenzo Stornelli

**Studente**  
Luca Di Vita

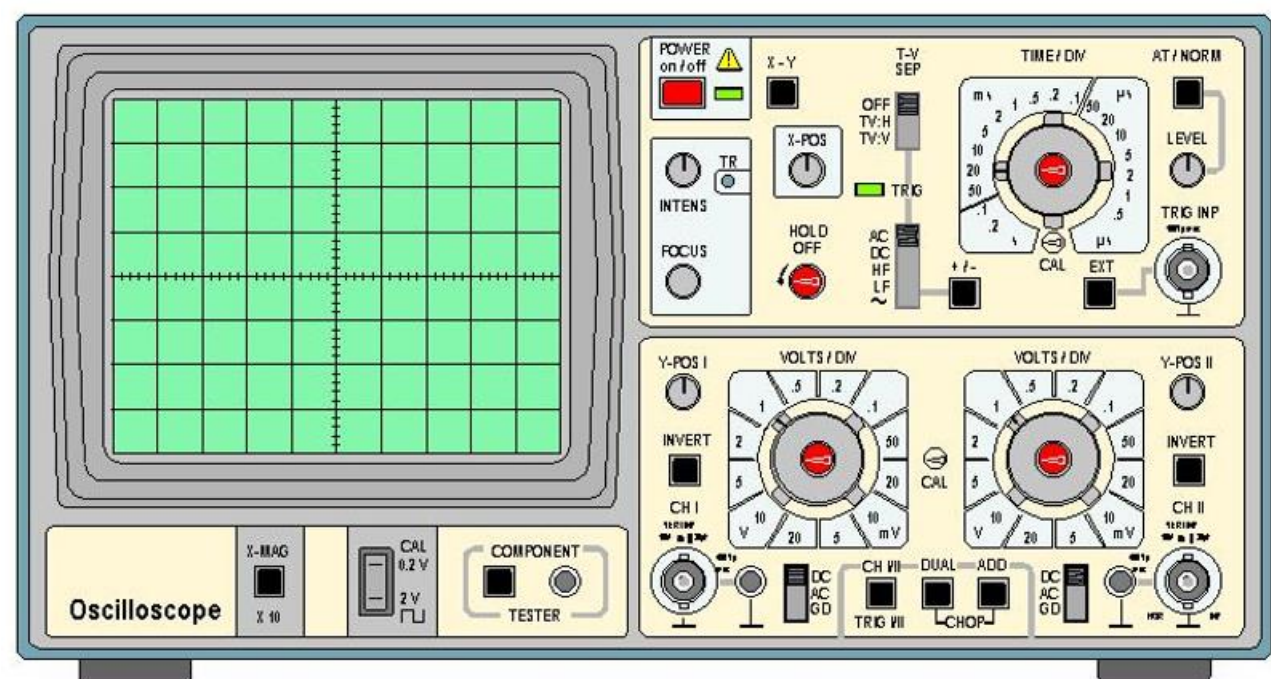
A.A 2015/2016

# Sommario

- Generalità
- Il Software
- L'Hardware
- L'App
- Conclusioni

# Generalità - 1

L'Oscilloscopio è uno strumento di misura elettronico che consente di visualizzare, su un riferimento cartesiano bidimensionale, l'andamento nel tempo di segnali di tensione consentendo di eseguire misure nel tempo quali: periodo, frequenza ecc. e in ampiezza come: valor medio, valore efficace ecc. La versatilità dello strumento permette, attraverso l'uso di opportuni trasduttori, di visualizzare l'andamento temporale di qualsiasi grandezza fisica. Il suo utilizzo è, generalmente, rivolto all'analisi di segnali periodici. L'evoluzione tecnologica ha portato alla definizione di due tipi di oscilloscopio: Digitale e Analogico.



# Generalità - 2

Esistono in commercio esemplari di oscilloscopi portatili che sono l'argomento centrale del lavoro.

In generale questi strumenti sono caratterizzati da alta portabilità ma anche da costi elevati come si può vedere dalla pagina sulla sinistra, presa da Google Shopping.



Oscilloscopio Portatile Lcd 12mhz Con Usb - Hps50

**290,00 €** da eBay - a.r.elettronica

Cerca Oscilloscopio Portatile Lcd 12mhz Con Usb - Hps50 su eBay nella categoria TV, Au



Oscilloscopio Palmare 1x25 MHz, UNI-T, UTD1025CL.

**406,26 €** da Distrelec Italia

Oscilloscopio. Base temporale: 10 ns/50 s/div. Dimensioni L x LA x A: 199 x 118 x 49 mm.



Oscilloscopio Palmare 2x25 MHz, UNI-T, UTD1025DL.

**588,04 €** da Distrelec Italia

Oscilloscopio. Base temporale: 10 ns/50 s/div. Dimensioni L x LA x A: 199 x 118 x 49 mm.



Oscilloscopio Palmare 1x25 MHz, UNI-T, UTD1025CL

**398,94 €** da Distrelec Italia

Peso: 0.9 kg. - Impedenza di ingresso: 1 MOhm. La dotazione include: 1 sonda canale - Cu



# Generalità - 3

L'obiettivo preposto è quello di creare uno strumento che sia adatto ad essere utilizzato tramite dispositivi mobili (quindi con grande portabilità), e conveniente sotto l'aspetto economico. Questo è stato reso possibile grazie a tecnologie attuali come:

- Android OS
- Arduino
- Protocollo Bluetooth

E numerosi framework e librerie quali:

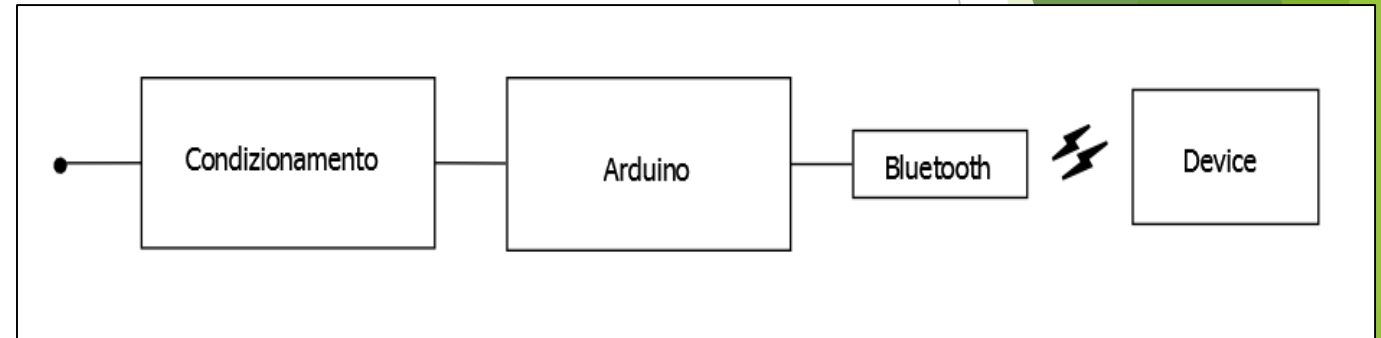
- JQueryMobile
- Cordova
- Highcharts.js



# Generalità - 4

La realizzazione dello strumento è stata semplificata fino ad essere ridotta al seguente schema a blocchi:

- Il Condizionamento adatta il segnale per lo stadio successivo.
- Arduino è il microcontrollore utilizzato per: campionare e convertire il segnale tramite l'ADC che lo stesso offre, permettere una comunicazione seriale con l'ultimo blocco.
- Il Bluetooth è il protocollo di comunicazione utilizzato per inviare i dati su un dispositivo mobile. E' stato scelto per il ridotto consumo di potenza, e per la disponibilità di API.

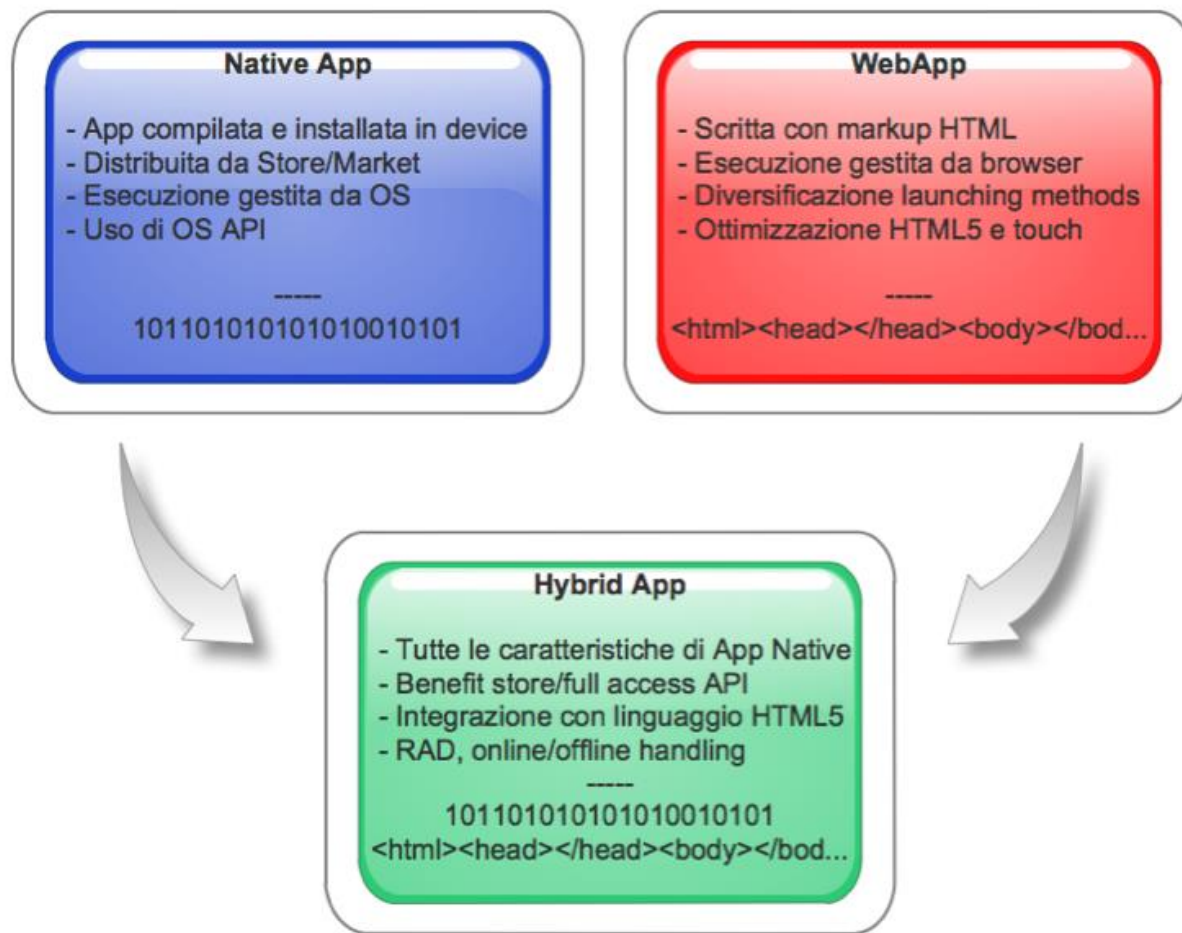


- Infine il Device è il dispositivo sul quale l'applicazione è installata. Ha il compito di avviare la connessione con il modulo bluetooth, riceve i campioni, ed esegue un plot della forma d'onda.



# Il Software - 1

Lo sviluppo dell'app è avvenuto con un approccio ibrido cioè che sfrutta le caratteristiche delle «Web App», che sono pagine web ottimizzate per dispositivi mobili e che fanno uso di tecnologie come HTML5, JavaScript, ecc, e delle «App Native» che sono app compilate per una specifica piattaforma.



# Il Software - 2

Per conseguire tale approccio si è usato il framework «Apache Cordova». La sua architettura si presenta come un contenitore di un'applicazione Web eseguita localmente. Infatti l'interfaccia dell'applicazione è costituita da una Web View che occupa l'intero schermo del dispositivo all'interno del quale viene visualizzato l'HTML, il CSS, ed il codice Javascript tramite il quale si accede ad un insieme di API che interfacciano l'applicazione con le funzionalità della piattaforma ospite.



APACHE  
CORDOVA™



# Il Software - 3

Pertanto Web view e API sono le componenti dell'applicazione, e sono queste componenti che il framework mette a disposizione dello sviluppatore, consentendogli di concentrarsi su un codice indipendente dalla piattaforma. Cordova infatti implementa lo stesso insieme di API su diverse piattaforme mobili creando un livello software standard a cui si possono interfacciare le nostre applicazioni, che saranno scritte con codice HTML, CSS e JavaScript. Una volta scritte avremo una applicazione per ciascuna piattaforma. Allo stato attuale *Apache Cordova* supporta diverse piattaforme mobili: Android, iOS, Blackberry, Windows Phone, ecc.



# Il Software - 4

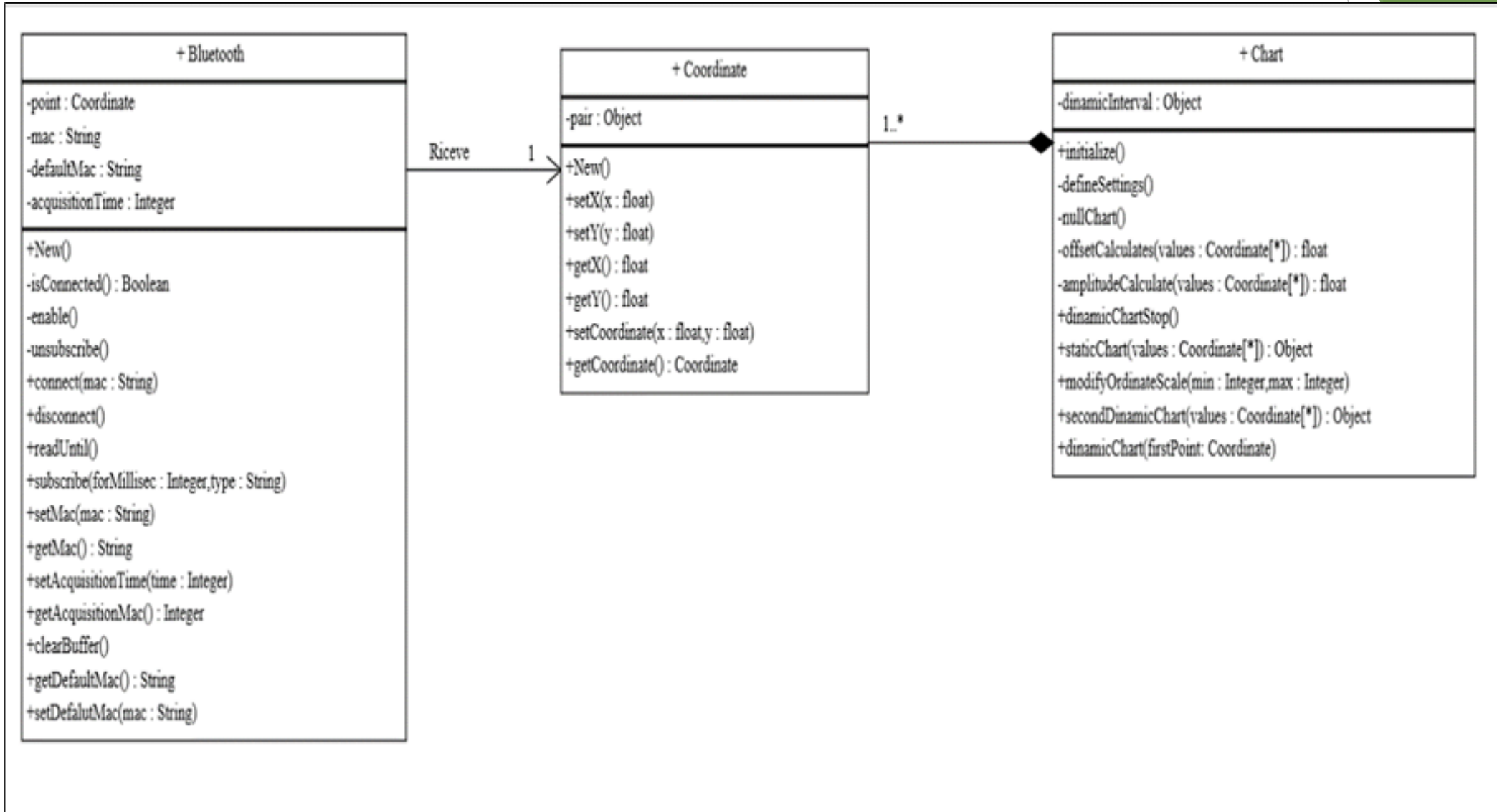
Per la programmazione JavaScript si è usato un approccio di tipo OOP con la progettazione di tre classi:

- Coordinate
- Chart
- Bluetooth

Si è simulato l'incapsulamento della classe utilizzando un approccio di tipo «module pattern», di cui si riporta un esempio (classe Coordinate)

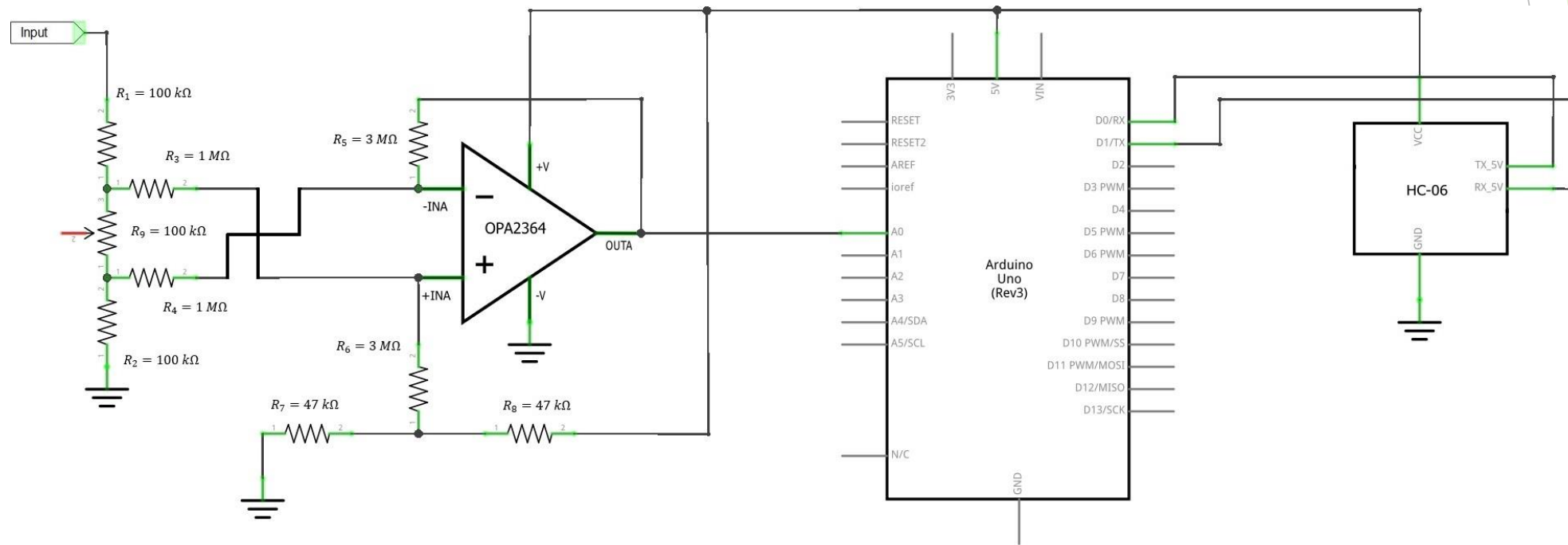
```
var Coordinate = (function() {  
  var pair = {  
    x: null,  
    y: null  
  };  
  return {  
    setX: function(coordinateX) {  
      pair.x = coordinateX;  
    },  
    setY: function(coordinateY) {  
      pair.y = coordinateY;  
    },  
    getX: function(){  
      return pair.x;  
    },  
    getY: function(){  
      return pair.y;  
    },  
    setCoordinate: function(coordinateX,coordinateY) {  
      pair.x = coordinateX;  
      pair.y = coordinateY;  
    },  
    getCoordinate: function() {  
      return pair;  
    }  
  }  
}());
```

# Il Software - 5



# L'Hardware - 1

Il circuito di seguito, costituisce l'hardware del dispositivo.



# L'Hardware - 2

Come si vede è composto da 3 stadi:

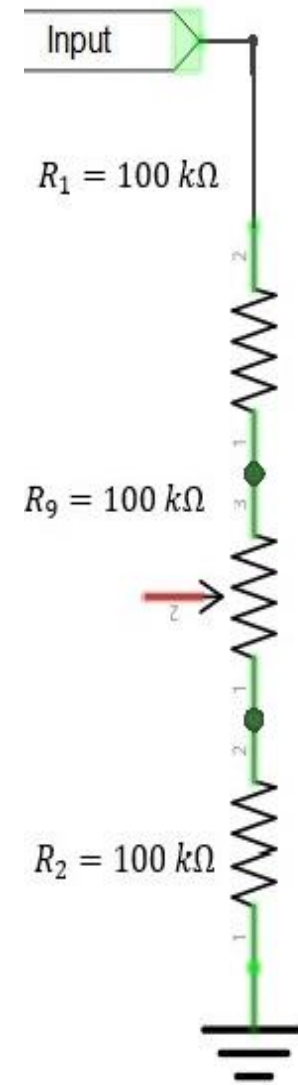
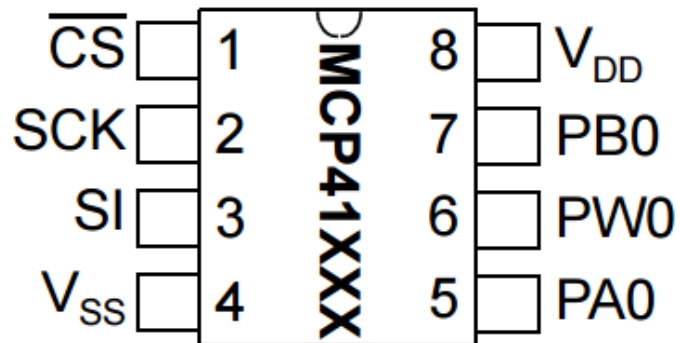
1. Un semplice partitore di tensione,
2. Un amplificatore operazionale differenziale,
3. Arduino e modulo Bluetooth.

I primi due si occupano di condizionare il segnale per permettere di inviarlo ad uno degli ingressi analogici di Arduino che ha un range di funzionamento da 0 a 5 Volt senza che lo stesso si guasti. Il segnale sarà ricostruito a livello software, e inviato tramite Bluetooth al dispositivo.

# L'Hardware - 3

Idealmente, si suppone che l'operazionale, che ha in ingresso resistore di ordine grande, non assorbi corrente, che le resistenze non introducano distorsioni, e che il segnale in ingresso non abbia offset.

L'idea alla base, è quella di attenuare a priori un segnale, per poi compensarlo tramite software. Il primo stadio è quindi costituito da un semplice partitore di tensione. Il resistore al centro, è un potenziometro digitale (MCP41100 da 100 k $\Omega$ ) pilotato da Arduino tramite librerie SPI.





# L'Hardware - 4

L'equazione del partitore è, dati  $R_1 = R_2 = R$  e  $R_9 = \frac{R}{n}$ :  $V_2 = \frac{R}{\frac{R}{n} + 2R} V_{in} =$

$$\frac{n}{2n+1} V_{in}, V_9 = \frac{\frac{R}{n} + R}{\frac{R}{n} + 2R} V_{in} = \frac{n+1}{2n+1} V_{in}.$$

Quello che si fa, è impostare inizialmente la resistenza digitale al suo valore minimo, in modo che, come si vedrà dallo stadio successivo, il rapporto di attenuazione è massimo. Contrariamente, con il massimo valore del componente, si ha un rapporto di attenuazione pari a 1.

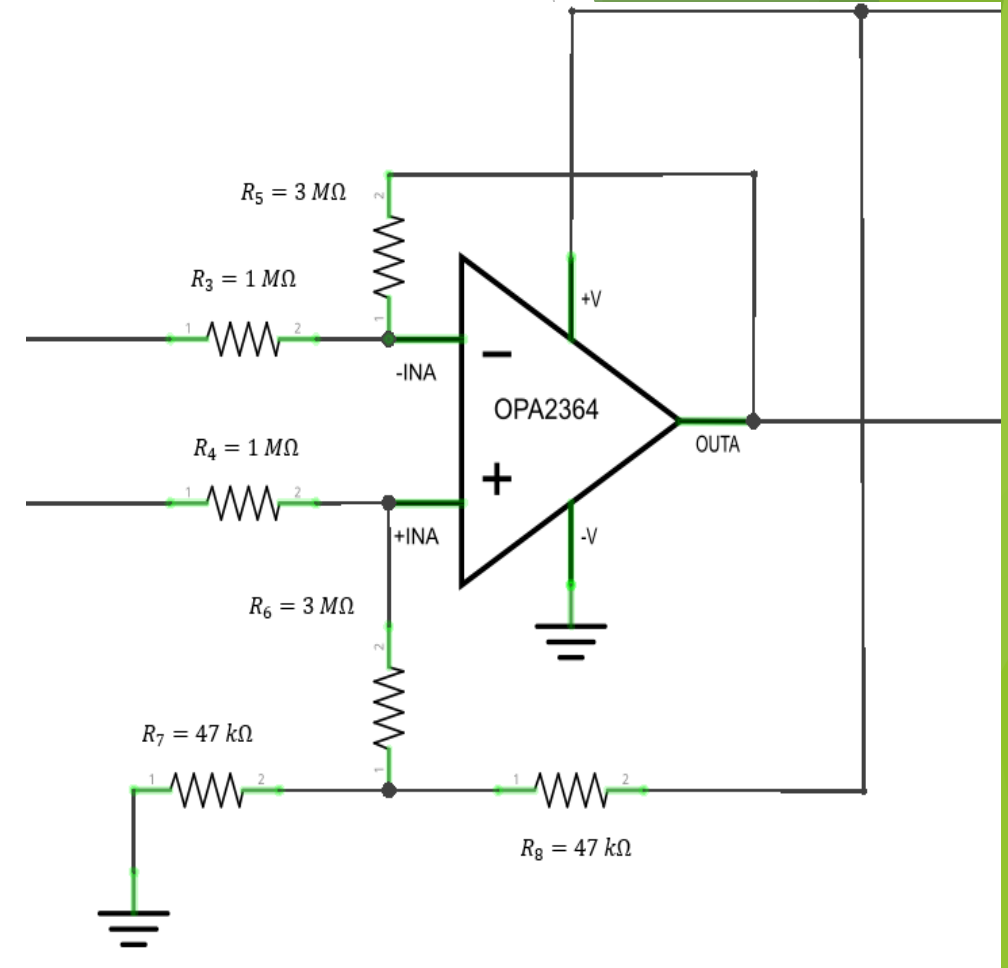
Se Arduino non rileva alcun segnale, vuol dire che è stato attenuato troppo e bisogna aumentare il valore del resistore. Quando un segnale sarà rilevato esso verrà compensato a livello software in quanto la resistenza è nota a priori da Arduino stesso che potrà calcolare il rapporto di attenuazione applicato.

# L'Hardware - 5

Il secondo stadio è un operazionale differenziale con una continua di 2,5 V sul morsetto non invertente, la quale uscita è data da  $V_{out} = A_d \Delta V + A_c \left( \frac{V^+ + V^-}{2} \right)$ , con  $A_d = \frac{R_5}{R_3} = 3$  e supponiamo  $A_c \rightarrow 0$ . In uscita si avrà:  $V_{out} = A_d A_t V_{in} + 2,5$  con  $A_d$  e  $A_t$  fattore di amplificazione e di attenuazione rispettivamente.

Ad esempio, senza considerare la continua: impostando valore di  $R_9$  massimo cioè  $R_9 = 100 \text{ k}\Omega$ ,  $V_9 = \frac{2}{3} V_{in}$  e  $V_2 = \frac{1}{3} V_{in}$  si ha  $\Delta V = V_9 - V_2 = A_t V_{in} = \frac{1}{3} V_{in}$  e quindi  $V_{out} = 3 \frac{1}{3} V_{in} = V_{in}$ , cioè si riporta in uscita lo stesso segnale in ingresso.

Impostandola invece al minimo, che è  $R_9 = 125 \Omega$ , si ha:  $V_9 = \frac{801}{1601} V_{in} \approx \frac{1}{2} V_{in}$  e  $V_2 = \frac{801}{1601} V_{in} \approx \frac{1}{2} V_{in}$ . E quindi  $V_{out} \approx 0$ .



# L'Hardware - 6

Il segnale in uscita dall'operazionale risulta così attenuato e con dinamica centrata a 2,5 V.

I morsetti di alimentazione dell'operazionale sono collegati a 5 V quello positivo, e a massa quello negativo. Sia la tensione da 2,5 sia l'alimentazione dell'operazionale sono forniti da Arduino. I segnali che superano in uscita il range  $[0,5]$  V, saranno quindi tagliati. Questa operazione è d'obbligo in quanto Arduino non può ricevere segnali negativi, e di superiori a 5 V di ampiezza.

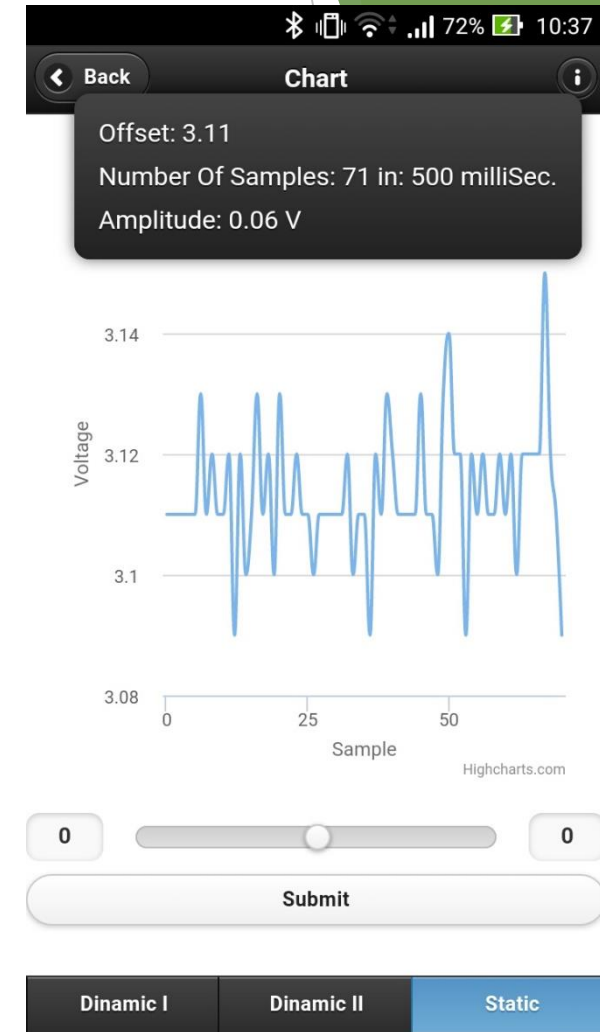
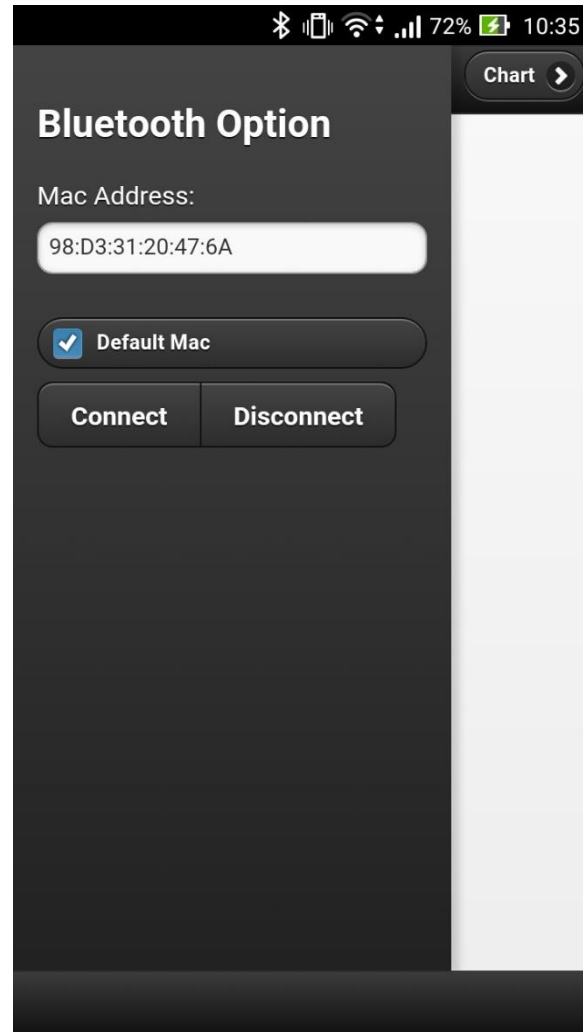
Arduino, ricevuto il segnale  $V_i = V_{out} = (A_d A_t V_{in} + 2.5) V$ , lo ricostruisce con le seguenti operazioni:

- ✓ Elimina la componente continua:  $V_a = V_i - 2.5 = A_d A_t V_{in} V$
- ✓ Calcola  $A_t = \frac{R_9 + R_1}{R_9 + R_1 + R_2} - \frac{R_1}{R_9 + R_1 + R_2} = \frac{R_9}{R_9 + R_1 + R_2} \Omega$  con  $R_9 = \frac{R_{AB} * D_n}{256} + R_w \Omega$  dove  $R_w = 125 \Omega$   $D_n \in [0,255]$  e  $D_n \in \mathbb{N}$ . Quindi divide  $V_a$  per  $A_t$ :  $V_b = V_a / A_t = A_d V_{in}$
- ✓ Divide per il fattore di amplificazione fisso  $A_d = 3$ :  $V_f = V_b / A_d = V_{in}$

Ricostruito, il segnale viene inviato al dispositivo.

# L'App - 1

Il risultato finale del lavoro di programmazione è un app composta da due schermate: Home e Chart, dove nella prima è possibile stabilire la connessione Bluetooth, e nella seconda è possibile visualizzare il grafico del segnale e delle informazioni relative ad esso quali offset, numero di campioni acquisiti e ampiezza.



# L'App - 2

Offre tre diverse modalità di stampa del segnale:

1. Static: Si acquisiscono il massimo numero di campioni acquisibili in mezzo secondo, e si stampano. Modalità normale.
2. Dinamic I: Si acquisisce un campione ogni mezzo secondo e lo si stampa, per cui permette di vedere l'evoluzione temporale del segnale. E' stato scelto mezzo secondo di tempo in quanto tempi minori causavano instabilità delle librerie grafiche. Modalità roll.
3. Dinamic II: Si acquisiscono il massimo numero di campioni acquisibili in mezzo secondo, e si stampa un campione ogni mezzo secondo. Modalità scan.

E' inoltre presente uno slider per il resizing che, se impostato al valore di default (0,0), farà sì che il riferimento cartesiano si centri automaticamente sul segnale, mentre se modificato assumerà, sull'asse delle ordinate, l'intervallo scelto.

# Un esempio



# Conclusioni

Da prove sperimentali si è visto che, lo strumento ha due tipi di limitazioni:

1. Inerenti alla frequenza del segnale in ingresso,
2. Inerenti l'ampiezza del segnale in ingresso.

Entrambe sono dovute alle limitate potenzialità di Arduino, e possono essere superate cambiando controllore. Inoltre:

- Il segnale deve essere centrato in 0.
- Si suppone l'idealità in cui l'operazionale non assorbe corrente.

Che possono essere facilmente superate apportando delle modifiche al circuito.

Eventuali utilizzi dello strumento sono a livello didattico, e se perfezionato, potrebbe essere commercializzato.

Grazie per l'attenzione