

Alcune esperienze di Laboratorio I utilizzano un sistema di acquisizione dati basato su Arduino

- ▶ Il framework Plasduino è concepito per automatizzare e semplificare il più possibile la fase di acquisizione
- ▶ Lasciare spazio e tempo per la successiva analisi dati agli studenti

Plasduino è un progetto per un ambiente *hardware* e *software* di acquisizione dati concepito per esperienze didattiche:

- ▶ Utilizzare sensori “semplici”
- ▶ Registrare quantità fisiche, come tempi, temperature etc...
 - ▶ Con risoluzioni adeguate per i laboratori didattici

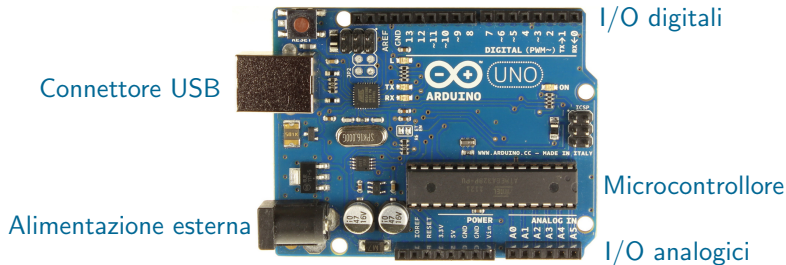
I dettagli di come programmare e costruire sistemi simili li studierete nei prossimi anni.

Non vi preoccupate troppo se non capite bene tutta la parte elettrica.

Oggi vogliamo dare una visione generale di quello che usate in Laboratorio

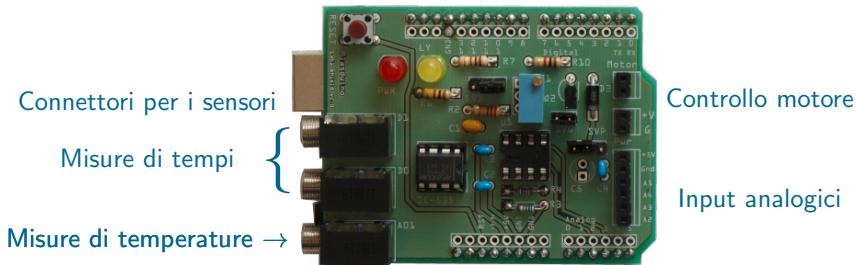
ARDUINO: IL CUORE DEL SISTEMA

[HTTP://WWW.ARDUINO.CC/](http://www.arduino.cc/)

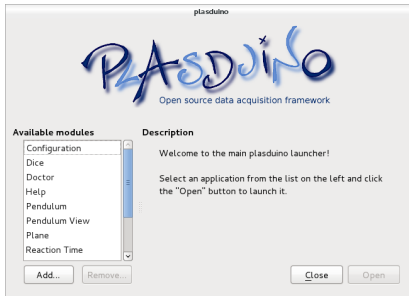


- ▶ Arduino: una piattaforma *open* di prototipizzazione elettronica:
 - ▶ Flessibile, facile da utilizzare, realizzata in Italia;
 - ▶ Programmabile ad alto livello (C/C++), ampia libreria di funzioni;
 - ▶ Utilizzato e supportato da una comunità numerosa ed attiva.
- ▶ Una tipica scheda di Arduino offre:
 - ▶ Un microcontrollore
 - ▶ I pin per le varie periferiche
 - ▶ Un'interfaccia seriale via USB
 - ▶ Alimentazione tramite USB o indipendente

GLI “SHIELD”



- ▶ Uno *shield* è un circuito stampato da connettere *sopra* la scheda di Arduino
 - ▶ Agisce da interfaccia con il mondo esterno: connettori per i sensori, condizionamento dei segnali, led, calibrazione interna
 - ▶ Nel nostro caso, abbiamo un unico shield per 6 esperienze di meccanica e termodinamica

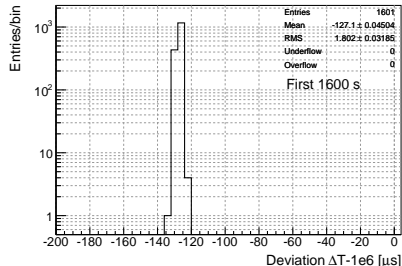
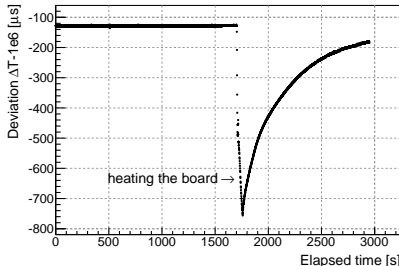


- ▶ Un'applicazione per il controllo dell'acquisizione:
 - ▶ Il più automatizzato possibile
 - ▶ Raccolta dati
 - ▶ Prima elaborazione
 - ▶ Archiviazione dei dati
- ▶ Un interfaccia grafica per la gestione delle acquisizioni
- ▶ Librerie per gestire sensori e dispositivi
- ▶ Un insieme di moduli per esperienze specifiche
 - ▶ Disegnate per le esperienze di Laboratorio 1

MISURE DI TEMPO: TIMER E INTERRUPT

- ▶ Tutte le operazioni interne del micro sono sincronizzate da un oscillatore interno: il *clock*
 - ▶ Con un periodo nominale di 16 MHz
- ▶ Un *timer* è un contatore che viene incrementato dal clock e può essere utilizzato per misurare i tempi
 - ▶ La libreria di Arduino fornisce una funzione “`micros()`” per ottenere il tempo dall'accensione del dispositivo:
<https://www.arduino.cc/en/Reference/Micros>
 - ▶ Usa un *timer* incrementato dal clock interno e prescalato di un fattore 64
 - ▶ i.e., con una granularità nominale di 4 μ s
- ▶ Un “interrupt” è una funzione che viene eseguita interrompendo qualsiasi altra attività e dandosi la priorità
 - ▶ Si può attivare, ad esempio, quando un pin cambia stato
 - ▶ 2 pin disponibili per questa funzione
 - ▶ Si può programmare con la funzione “`attachInterrupt()`”:
<https://www.arduino.cc/en/Reference/AttachInterrupt>
- ▶ Per usare queste funzioni non è necessario conoscere i dettagli dell'architettura del micro

MISURE DI TEMPO: PERFORMANCE

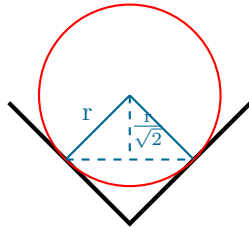


- ▶ Testato in laboratorio il 1PPS (1 pulse-per-second) di un GPS:
 - ▶ RMS dell'intervallo misurato tra due 1PPS successivi di $1.8 \mu\text{s}$, non lontano da $4/\sqrt{12} \mu\text{s}$.
 - ▶ Deviazione media dal valore nominale di $\sim 100 \mu\text{s}$ (su 1 s) a temperatura ambiente.
- ▶ La granularità di $4 \mu\text{s}$ è vera
- ▶ Qual'è l'incertezza sulla singola misure di tempo?
 - ▶ Incertezza strumentale più piccola delle fluttuazioni statistiche tipiche
 - ▶ Fate tante misure; studiate media e RMS...

PRIMO ESEMPIO PRATICO: IL PIANO INCLINATO

Moto di una sfera su di un piano inclinato con profilo “a V” (ad angolo retto)

- ▶ Il centro di rotazione a $\frac{r}{\sqrt{2}}$ dal centro della sfera
- ▶ Due traguardi ottici collegati a plasduino
 - ▶ Cambiano stato quando il link ottico viene interrotto
 - ▶ Fanno partire un interrupt in Arduino
- ▶ Si misurano le differenze di tempo tra due interrupt successivi
 - ▶ Non c'è distinzione tra i 2 traguardi ottici



IL PIANO INCLINATO: MISURE DI SPAZIO-TEMPO

Accelerazione del centro di massa della sfera lungo il profilo inclinato:

$$a = \frac{5}{9}g \sin(\alpha) \quad (1)$$

La legge oraria (con velocità iniziale nulla) sarà allora

$$l(t) = \frac{1}{2}at^2 = \frac{5}{18}g \sin(\alpha)t^2 \quad (2)$$

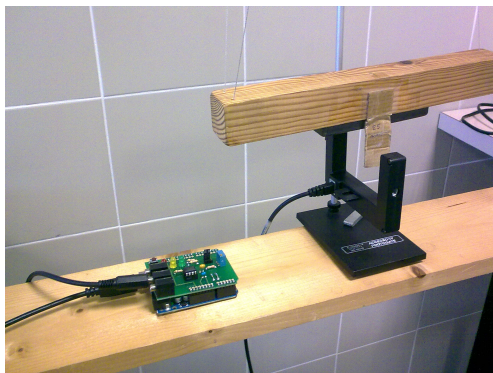
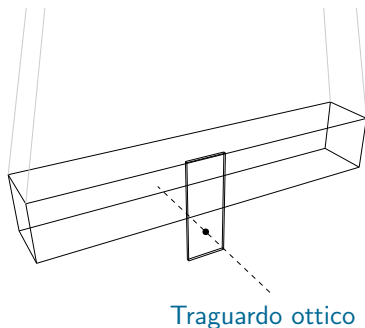
Procedura:

- ▶ Si pone una fotocella in fondo alla guida; l'altra fotocella verrà di volta in volta posta in punti diversi della guida
- ▶ Per ogni configurazione si misura la distanza dalle due fotocellule ed il tempo che occorre alla sferetta per compiere il percorso (con plasduino)
- ▶ Si costruisce una tabella di coppie spazio-tempo (con le incertezze associate): $(l_i \pm \Delta l_i, t_i \pm \Delta t_i)$

L'analisi dei dati:

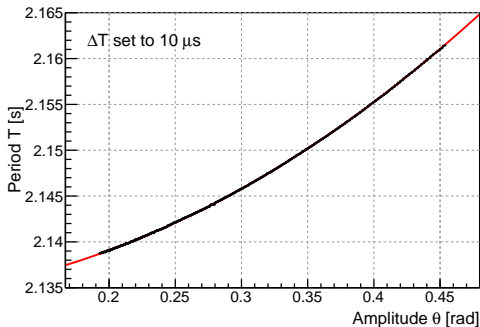
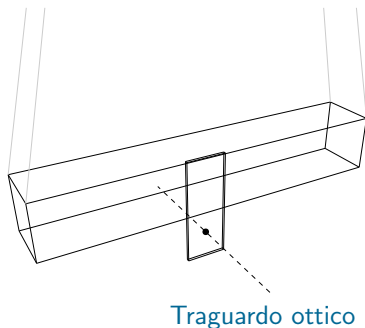
- ▶ Media e lo scarto quadratico medio
- ▶ Grafici in scala lineare o logaritmica
- ▶ Metodi di fit

UN ESEMPIO: IL PENDOLO “DIGITALE”



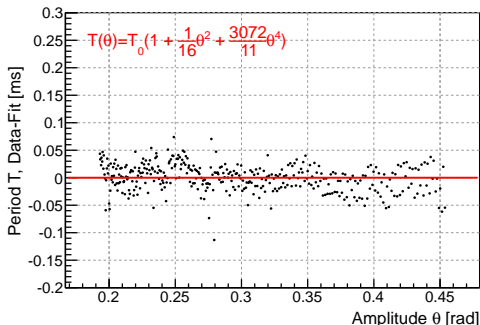
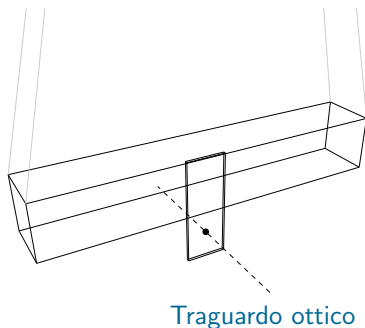
- ▶ Misura del periodo T e del tempo di transito di una bandierina nel punto più basso.
 - ▶ Misura dello smorzamento esponenziale (?)
- ▶ Trascurando le perdite di energia in una oscillazione possiamo stimare l'ampiezza θ_{\max} .
 - ▶ Misura dell'anarmonicità del pendolo.
 - ▶ Si apprezza chiaramente il termine in θ^4 !

UN ESEMPIO: IL PENDOLO “DIGITALE”



- ▶ Misura del periodo T e del tempo di transito di una bandierina nel punto più basso.
 - ▶ Misura dello smorzamento esponenziale (?)
- ▶ Trascurando le perdite di energia in una oscillazione possiamo stimare l'ampiezza θ_{\max} .
 - ▶ Misura dell'anarmonicit  del pendolo.
 - ▶ Si apprezza chiaramente il termine in θ^4 !

UN ESEMPIO: IL PENDOLO “DIGITALE”



- ▶ Misura del periodo T e del tempo di transito di una bandierina nel punto più basso.
 - ▶ Misura dello smorzamento esponenziale (?)
- ▶ Trascurando le perdite di energia in una oscillazione possiamo stimare l'ampiezza θ_{\max} .
 - ▶ Misura dell'anarmonicit  del pendolo.
 - ▶ Si apprezza chiaramente il termine in θ^4 !

- ▶ I termistori possono essere visti come resistenze il cui valore cambia con la temperatura:
<https://it.wikipedia.org/wiki/Termistore>
- ▶ NTC: Negative Temperature Coefficient, resistenza che decresce con l'aumentare della temperatura
- ▶ La relazione che lega temperatura e resistenza è parametrizzata dall'equazione di "Steinhart-Hart":

$$\frac{1}{T} = A + B \cdot \ln(R_T) + C \cdot (\ln(R_T))^3 \quad (3)$$

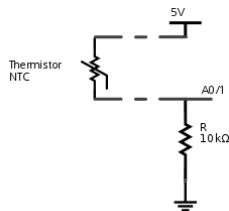
- ▶ I parametri A, B e C sono le costanti di calibrazione e dipendono dal modello di sensore che avete a disposizione
 - ▶ Il produttore vi dà i parametri nominali
 - ▶ Ma ve li potete ricalibrare da soli: quanti punti vi servono?

MISURE DI TEMPERATURA: L'ADC

- Per ricavare la resistenza utilizziamo lo schema elettrico di un partitore resistivo

- Dobbiamo leggere la tensione V e ricavare R_T

$$R_T = R(V_{CC}/V - 1) \quad (4)$$



- Arduino utilizza un convertitore Analogico Digitale a 10 bit ad approssimazioni successive:
 - 10 bit: $2^{10} = 1024$ passi da 0 a 5 V
 - approssimazioni successive: una tensione di riferimento viene confrontata con il segnale da digitalizzare, un bit alla volta a partire dal più significativo
- Ci sono 6 canali analogici, ma in realta' c'e' solo un ADC
 - Si legge un canale alla volta
- L'ADC si legge con la funzione `analogRead()`:
<https://www.arduino.cc/en/Reference/AnalogRead>
 - Anche in questo caso non è necessario conoscere i dettagli

- ▶ Calibrazione dello “zero” ($0\text{ }^{\circ}\text{C}$), con acqua distillata in equilibrio tra fase liquida e solida
 - ▶ e.g. <https://www.youtube.com/watch?v=KY0JayWqB3g>
 - ▶ c'è un “offset” sistematico nelle misure?
- ▶ Quali altri riferimenti “sicuri” si possono prendere per la calibrazione?
- ▶ Tempo di rilassamento: il sensore deve andare all'equilibrio termico con il mezzo, $T(t) \sim T_0 e^{t/\tau} + T_{fin}$ dove τ dipende dalla sua capacità termica
 - ▶ Siete in grado di misurare τ dai dati delle vostre esperienze?
 - ▶ In che scala conviene rappresentare $T(t) - T_{fin}$ vs t ?
- ▶ Contributi alla risoluzione:
 - ▶ Precisione dei componenti – effetti sistematici
 - ▶ Granularità dell'ADC – risoluzione strumentale
 - ▶ Se fate i conti bene dovreste trovare una risoluzione $\Delta T \sim 0.1 - 0.2\text{ }^{\circ}\text{C}$.

ESERCIZIO (PER I VOLENTEROSI)

RICAVARE IL CONTRIBUTO ALLA RISOLUZIONE IN T LEGATO ALL'ADC

Sappiamo che:

- ▶ L'ADC legge una tensione da 0 a V_{cc} (5V) in 1024 passi
- ▶ La resistenza del termistore si ricava quindi come:

$$R_T = R(1023/x - 1) \quad (5)$$

con $R = 10000 \Omega$ e $\Delta x = 1$

- ▶ La temperatura (in gradi kelvin) si ottiene con:

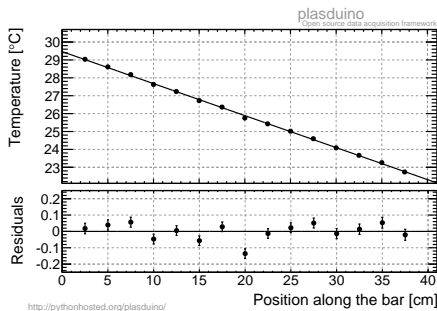
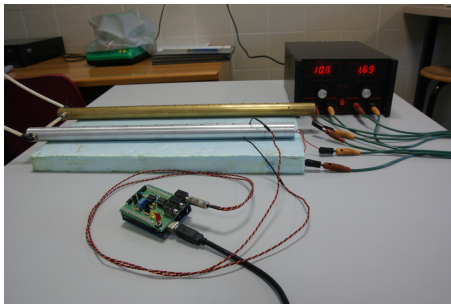
$$T = \frac{1}{A + B \cdot \ln(R_T)} \quad (6)$$

con $A = 8.65 \cdot 10^{-4}$ e $B = 3.55 \cdot 10^{-4}$ (trascurando C in eq. 3)

Siamo in grado di propagare l'incertezza ΔT al variare di T ?

- ▶ Usando le formule sacre
- ▶ Studiando $T(x+1) - T(x)$ in maniera numerica

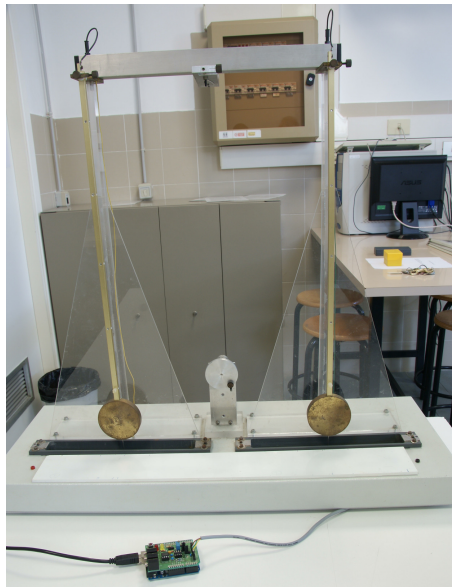
CONDUCIBILITÀ TERMICA DI UNA BARRA METALLICA



- ▶ In una barra metallica, riscaldata da un lato tramite una resistenza elettrica e raffreddata dall'altro con un flusso continuo d'acqua
- ▶ ci si aspetta un andamento lineare della temperatura:
$$T(x) = T_0 - \frac{W}{\lambda S}x$$
 - ▶ Con W potenza erogata e S sezione della barra
- ▶ Si misura la temperatura in diversi punti della barra
 - ▶ Ci sono dei fori per facilitare la misura al centro della barra stessa
- ▶ Dal coefficiente angolare si ricava la conducibilità termica λ

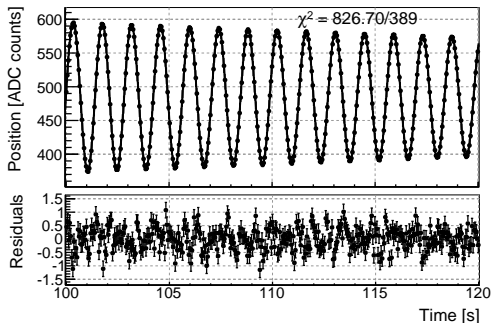
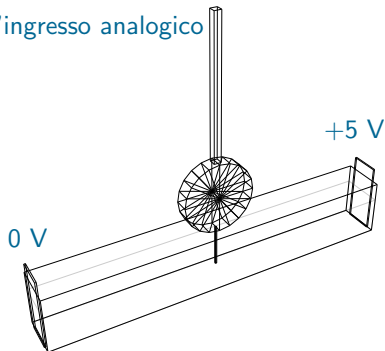
IL PENDOLO “ANALOGICO”: APPARATO SPERIMENTALE

- ▶ Altro esempio di utilizzo dell'ADC di Arduino nel Laboratorio 1
- ▶ Questo apparato consente di studiare, in dettaglio, il moto di un pendolo
 - ▶ Periodo di oscillazione
 - ▶ Smorzamento
- ▶ Si possono accoppiare i pendoli tramite una molla e studiare il comportamento
 - ▶ Modi normali
 - ▶ Battimenti
- ▶ Questo è l'apparato che userete nelle prossime esperienze



IL PENDOLO “ANALOGICO”: PRINCIPIO

All'ingresso analogico



- ▶ Utilizziamo una punta immersa in acqua come partitore resistivo.
 - ▶ Direttamente ad uno degli ingressi analogici.
- ▶ 10 bit di ADC su ~ 20 cm corrisponde ad una risoluzione spaziale di $\sim 200/\sqrt{12} \sim 60 \mu\text{m}$.
 - ▶ Il χ^2 ed i residui indicano che, almeno vicino al centro, non siamo lontani dalla risoluzione “teorica”
 - ▶ Non-linearità residua al livello di qualche % ai bordi

SINGOLO PENDOLO

Il moto di un singolo pendolo è descrivibile come un oscillatore armonico smorzato, la cui soluzione è del tipo:

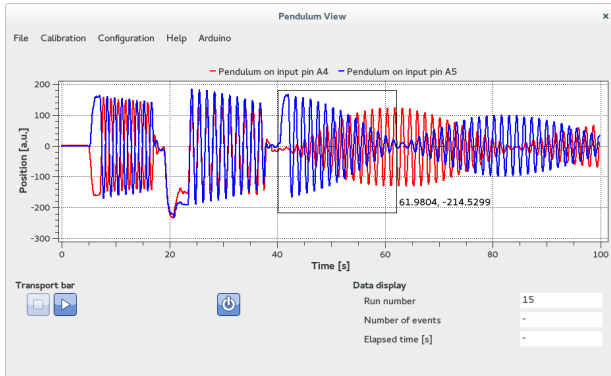
$$x(t) = c_0 + c_1 e^{-\lambda t} \sin(\omega_0 t + \phi_0) \quad (7)$$

Dove ω_0^{th} previsto dalla teoria vale:

$$\omega_0^{th} = \sqrt{\frac{mgl}{I}} \quad (8)$$

- ▶ La prima misura consiste quindi nel mettere in oscillazione un pendolo da solo e confrontare ω_0 con quello atteso
- ▶ Primo esempio di fit numerico serio (attenzione ai parametri iniziali)
- ▶ Si può studiare lo smorzamento: è davvero esponenziale?
- ▶ Si possono studiare entrambi i pendoli e controllare che siano uguali (stesso periodo e stesso smorzamento)
- ▶ Si può aumentare l'attrito (con uno "smorzatore", e.g. un galleggiante da pesca) e studiare in nuovo smorzamento.

COME APPARE IL MODULO “PANDULUM VIEW”



- ▶ Tasti di Start e Stop per inizio e fine dell'acquisizione
- ▶ La scala delle y è in unità arbitrarie, si può centrare sullo zero tramite il menù “Calibration”
- ▶ Passando il mouse nella finestra grafica vengono visualizzate le coordinate
 - ▶ Per una prima analisi veloce dei dati

PENDOLI ACCOPPIATI: INTRODUZIONE

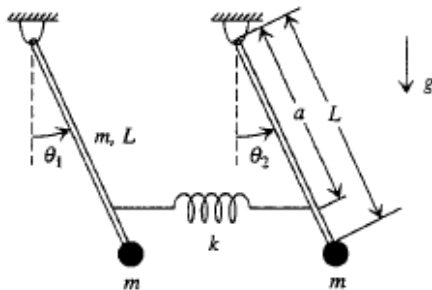
Due pendoli uguali realizzati con due aste rigide collegate tra loro da una molla. La forza che accoppia i due pendoli è del tipo:

$$F_k = -k(a \sin(\theta_2) - a \sin(\theta_1) - l_0)$$

Con un po' di conti si risolve (diagonalizza) il sistema e si trova la soluzione.

Ci sono però due configurazioni iniziali (corrispondenti ai cosiddetti *modi normali* di oscillazione) per cui il moto di entrambi i pendoli è armonico:

- ▶ quella in cui essi si muovono in fase
- ▶ quella in cui essi si muovono in controfase



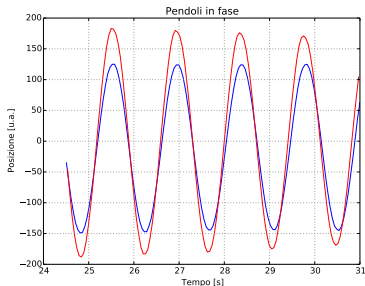
PENDOLI ACCOPPIATI: MODI NORMALI

OSCILLAZIONE IN FASE

Spostare i pendoli nello *stesso verso*, di uguali ampiezze e lasciarli andare contemporaneamente.

In queste condizioni la molla non è sollecitata dai due pendoli, e non influenzerà il movimento dei due pendoli:

$$\omega_f \sim \omega_0$$

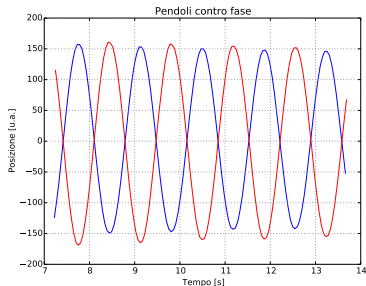


OSCILLAZIONE CONTROFASE

Spostare i pendoli nel *verso opposto*, di uguali ampiezze e lasciarli andare contemporaneamente.

In queste condizioni la molla è sollecitata in maniera simmetrica dai due pendoli, e ne cambierà la frequenza di oscillazione:

$$\omega_c > \omega_f$$



PENDOLI ACCOPPIATI: BATTIMENTI

Si sposta uno dei due pendoli tenendo l'altro fermo nella sua posizione di equilibrio e si lascia oscillare il sistema. Il moto risultante è dato dalla somma (con uguali ampiezze) dei due modi normali

$$x(t) = A_0 [\cos(\omega_f t + \phi_1) + \cos(\omega_c t + \phi_2)] \quad (9)$$

o, per le formule di prostaferesi

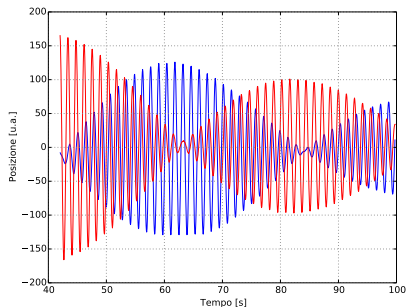
$$x(t) = 2A_0 \left\{ \cos \left[\frac{(\omega_c + \omega_f)t}{2} + \frac{(\phi_2 + \phi_1)}{2} \right] \cos \left[\frac{(\omega_c - \omega_f)t}{2} + \frac{(\phi_2 - \phi_1)}{2} \right] \right\}$$

L'oscillazione risultante, di pulsazione angolare portante

$$\omega_p = \frac{(\omega_c + \omega_f)}{2} \approx \omega_c, \omega_f \quad (10)$$

è *modulata* da un'onda di pulsazione angolare ω_b molto più piccola

$$\omega_b = \frac{(\omega_c - \omega_f)}{2} \ll \omega_c, \omega_f \quad (11)$$



plasduino

Open source data acquisition framework

[Home](#)[Installation](#)[Overview](#)[Modules](#)[Shields](#)[Core development](#)

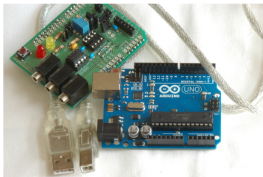
HOME

Plasduino is a data acquisition framework for the physics lab. It builds on top of an old, no longer maintained project for data acquisition through the parallel port ([plas](#)) and is based on the [arduino](#) platform (hence the name plasduino). If you are wondering what we are talking about, our [overview paper](#) is probably the single most useful reference, so take a look!

Plasduino is an open hardware/software project. The codebase is released under the GPL licence along with the shield schematics. The main goal is to provide all the necessary tools to assemble a flexible, easy-to-use, general-purpose data acquisition system, suitable for physics didactic experiments, for under 50\$.

Plasduino is written in [python](#) (relying on several third-party python modules) and uses [PyQt4](#) for the graphical user interface. A list of pre-requisites for the installation is available [here](#).

The plasduino code is hosted on [bitbucket](#). Among other things, you want to use bitbucket to [browse the source code](#) or [download plasduino](#) in your favourite flavour. In addition to that, we use the [issue tracker](#) provided by bitbucket for bug reports and feature request; using the issue tracker is by far the most effective way to complain or suggest improvements!



Copyright © 2013-2014 the plasduino team

We try and make sure that this page validates as [HTML 5](#) and [css level 3](#) (feel free to bug us if it doesn't).
Compiled from plasduino 2.0.2, last update Thursday, February 06 2014 at 14:45 (+0100).

► Alcuni link utili:

- Pagina web del progetto: <http://pythonhosted.org/plasduino/>
- Altra pagina web in italiano: <http://www.df.unipi.it/~plasduino/pfs/>
- Articolo di riferimento: <http://arxiv.org/abs/1312.1805>
- Download: <https://bitbucket.org/lbaldini/plasduino/downloads>
- Issue tracker: <https://bitbucket.org/lbaldini/plasduino/issues>