
plasduino

Open source data acquisition framework

Utilizzare Arduino per l'acquisizione dati in semplici esperienze didattiche di laboratorio

Carmelo Sgrò² (carmelo.sgro@pi.infn.it)

Massimiliano Razzano^{1,2} (massimiliano.razzano@pi.infn.it)

¹ Dipartimento di Fisica E. Fermi, Università di Pisa

² INFN-Sezione di Pisa

Incontri di Fisica 2015

LNF - 8 Ottobre 2015

Ovvero cosa faremo oggi:

- ▶ Il progetto Plasduino: cos'è e qual'è il suo scopo
- ▶ L'acquisizione di segnali con Arduino: tipi di segnali e performance
- ▶ Un paio di casi pratici: vedremo insieme alcuni esempi di esperienze implementate nel corso di Laboratorio 1 di Fisica a Pisa
 - ▶ Per prendere confidenza con il sistema
 - ▶ Acquisire un po' di dati
 - ▶ Studiare un po' di analisi dati
 - ▶ La "misura" e la sua incertezza
 - ▶ Fluttuazioni statistiche
 - ▶ La verifica del fenomeno fisico
- ▶ Prova pratiche: siete liberi di provare da soli
- ▶ Feedback: siamo interessati alle vostre opinioni, suggerimenti, modifiche etc.
 - ▶ Palsduino è libero e aperto, potete usarlo, modificarlo e adattarlo alle vostre esigenze come preferite
 - ▶ Saremo felici di offrirvi supporto nel vostro sviluppo

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

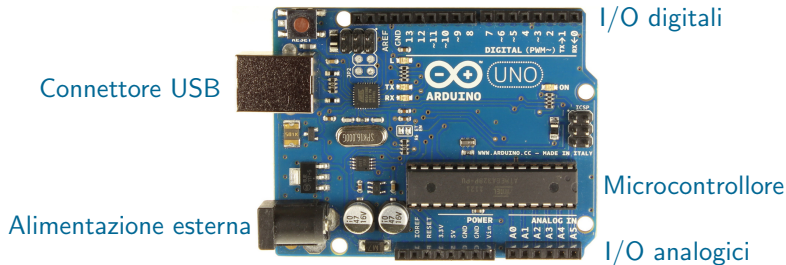
- ▶ Registrare quantità fisiche come tempi, temperature etc...
 - ▶ Con risoluzioni adeguate per i laboratori didattici
- ▶ Semplificare e automatizzare (quando possibile) la presa dati
- ▶ Lasciare la successiva analisi dati agli studenti

Mira ad essere:

- ▶ Semplice ed estendibile:
 - ▶ Semplice da installare e utilizzare;
 - ▶ Adattabile con poco sforzo a varie esperienze di laboratorio.
- ▶ Economico (~ 50 €):
 - ▶ Replicabile senza investimenti proibitivi sulla scala di un laboratorio didattico.
- ▶ Completamente libero (*Free and Open Source*):
 - ▶ Utilizza solo componenti (*hardware* e *software*) “open”;
 - ▶ Il codice sorgente è liberamente accessibile sotto licenza GPL;
 - ▶ Gli schemi elettronici e la documentazione sono disponibili sulla pagina web del progetto.

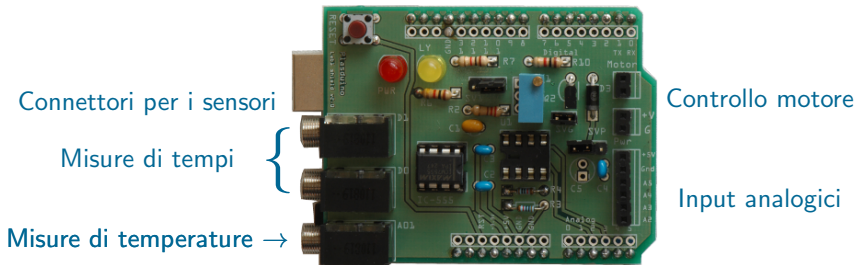
ARDUINO: IL CUORE DEL SISTEMA

[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 (per 23 €) offre:
 - ▶ Un microcontrollore;
 - ▶ 14+ ingressi/uscite digitali (con supporto per PWM e *interrupt*);
 - ▶ 6+ ingressi analogici con ADC a 10/12 bit;
 - ▶ Un'interfaccia seriale via USB.

GLI “SHIELD”

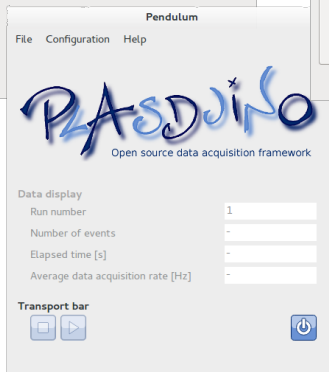
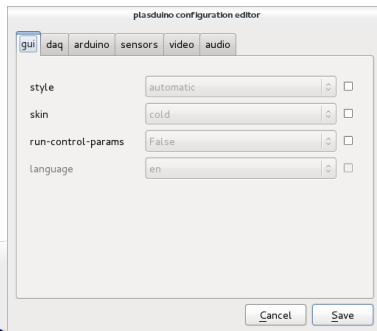
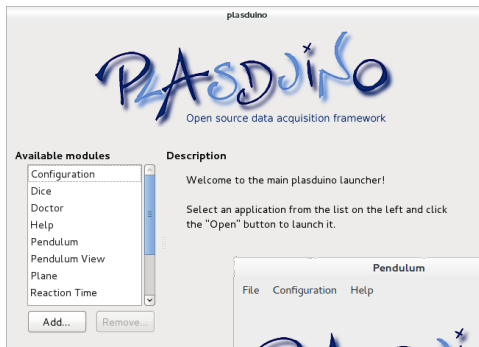


- ▶ Uno *shield* è un circuito stampato da connettere *sopra* la scheda di Arduino
 - ▶ Agisce da interfaccia con il mondo esterno: connettori per i sensori, condizionamento dei segnali, led, calibrazione interna
 - ▶ Nel nostro caso, abbiamo un unico shield per 6 esperienze di meccanica e termodinamica (replicabile a ~ 25 €)
- ▶ Disegnati ed assemblati “in casa”:
 - ▶ Descrizione, schemi elettronici, maschere e lista dei componenti disponibili sulla pagina web del progetto.

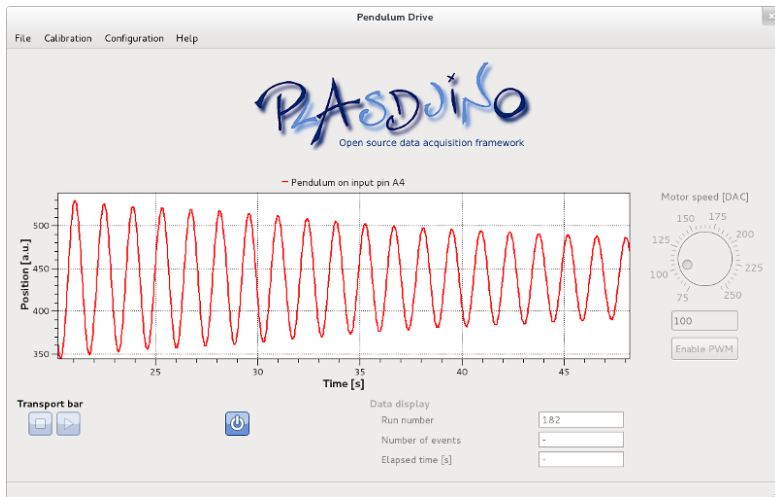
- ▶ plasduino offre un ambiente completo di acquisizione dati da eseguire su un PC connesso ad Arduino
 - ▶ Implementato nel linguaggio di programmazione Python¹
 - ▶ Multiplatforma (GNU/Linux, Windows, Mac OS)
- ▶ Un'applicazione per il controllo dell'acquisizione:
 - ▶ Il più automatizzato possibile (riconoscimento porta USB, caricamento del *firmware*, etc.)
 - ▶ Raccolta, prima elaborazione ed archiviazione dei dati
- ▶ Una vasta gamma di *widget* per l'interfaccia grafica
 - ▶ Organizzati in moduli da assemblare a piacimento
- ▶ Librerie per gestire sensori e dispositivi
- ▶ Un insieme di moduli per esperienze specifiche
 - ▶ Disegnate per le esperienze di Laboratorio 1

¹È il linguaggio con la crescita di popolarità più grande nello scorso decennio.

L'INTERFACCIA GRAFICA

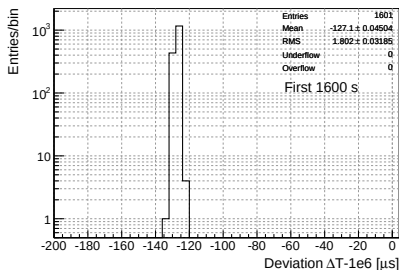
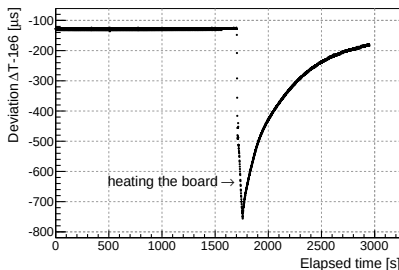


L'INTERFACCIA GRAFICA



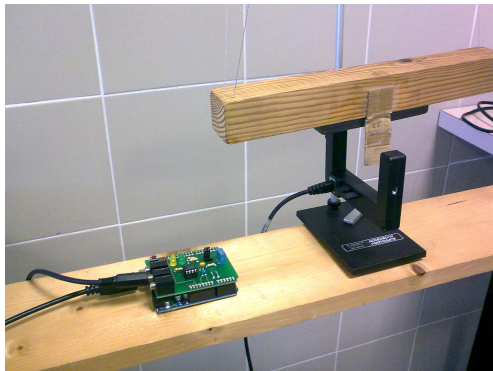
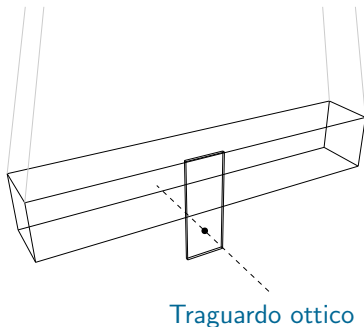
<http://pythonhosted.org/plasduino/screenshots.html>

MISURE DI TEMPO: PERFORMANCE



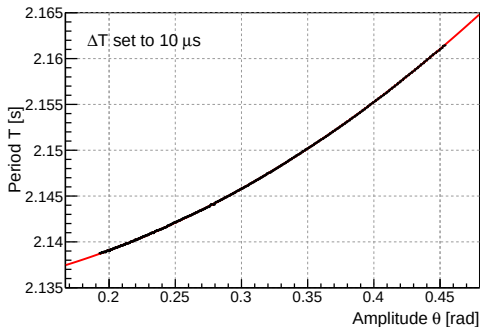
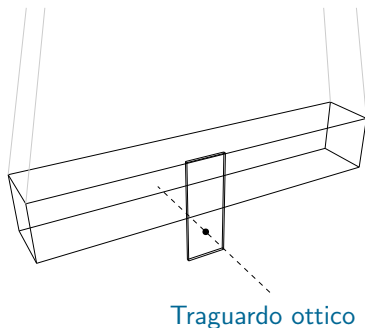
- ▶ La libreria di Arduino fornisce un *timer* incrementato dal clock a 16 MHz prescalato di un fattore 64:
 - ▶ i.e., con una granularità nominale di $4 \mu\text{s}$.
- ▶ Testato in laboratorio il 1PPS di un GPS (e con un impulsatore):
 - ▶ RMS dell'intervallo misurato tra due 1PPS successivi di $1.8 \mu\text{s}$, 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.
- ▶ Errore sistematico di 10^{-4} sulle misure di tempo.

UN ESEMPIO: IL PENDOLO “DIGITALE”



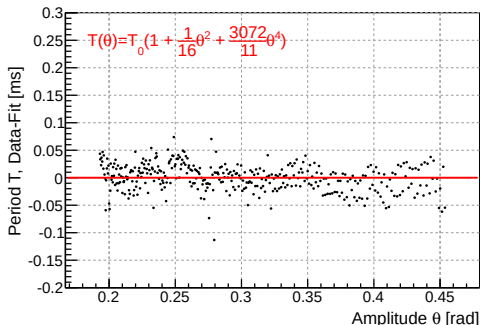
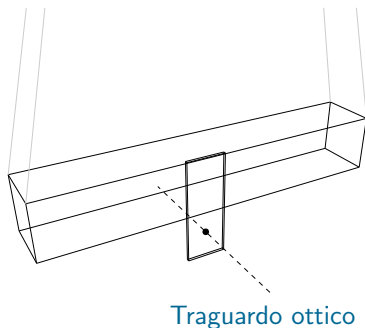
- ▶ Misura del periodo T e del tempo di transito di una bandierina nel punto più basso.
 - ▶ Misura dello smorzamento esponenziale (?)
- ▶ Trascurando le perdite di energia in una oscillazione possiamo stimare l'ampiezza $\theta_{\max}$.
 - ▶ Misura dell'anarmonicità del pendolo.
 - ▶ Si apprezza chiaramente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

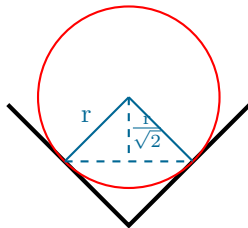
PRIMO ESEMPIO PRATICO: IL PIANO INCLINATO

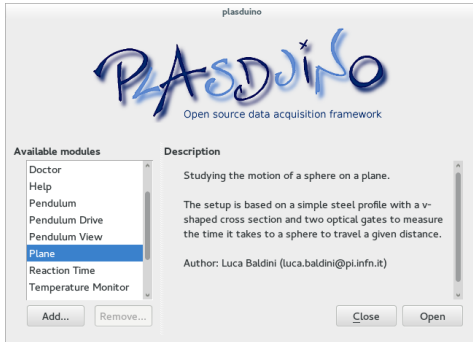
Lo scopo è quello di studiare il moto di una sfera su di un piano inclinato
A disposizione:

- ▶ Un profilo metallico “a V” (ad angolo retto)
- ▶ Una sfera metallica
 - ▶ Il centro di rotazione a $\frac{r}{\sqrt{2}}$ dal centro della sfera
- ▶ Due traguardi ottici collegati a plasduino
 - ▶ Per misurare del differenze di tempo
- ▶ Metro a nastro

Esperienze che abbiamo implementato:

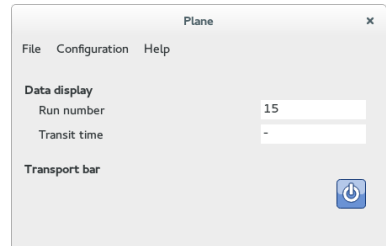
- ▶ Legge Spazio–Tempo
- ▶ Conservazione dell'Energia





- ▶ Dopo aver lanciato plasduino, compare la finestra principale
- ▶ L'elenco di tutti i moduli, per le varie esperienze è visibile sulla sinistra
- ▶ Evidenziandolo, compare una breve descrizione
- ▶ Per il piano inclinato selezionare "Plane"

- ▶ Plane è estremamente semplice →:
 - ▶ Numero di "run", che si incrementa ad ogni lancio di plasduino
 - ▶ Ultimo tempo di transito
 - ▶ Tra due segnali successivi, non c'è distinzione tra i 2 traguardi ottici.
- ▶ Il tempo di transito va trascritto a mano sul proprio quaderno ad ogni misura



IL PIANO INCLINATO: CONSERVAZIONE DELL'ENERGIA

Lasciando una sfera da un'altezza h , la legge di conservazione dell'energia si può scrivere come:

$$mgh = \frac{1}{2}mv^2 + \frac{1}{2}I\omega^2 \quad (1)$$

che fruttando

$$I = \frac{2}{5}mr^2 \quad (2)$$

$$v = \omega \frac{r}{\sqrt{2}} \quad (3)$$

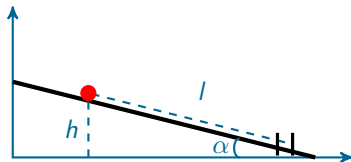
$$h = l \sin(\alpha) \quad (4)$$

diventa:

$$I = \frac{1}{g \sin(\alpha)} \left(\frac{1}{2} + \frac{2}{5} \right) v^2 = \frac{1}{2} \frac{1}{g \sin(\alpha)} \frac{9}{5} v^2 \quad (5)$$

Ovvero I è proporzionale a v^2 (vero anche se si trascura la parte rotazionale).

Notazioni come in figura:



Procedura:

- ▶ Se si pongono le due fotocellule in fondo alla guida, vicine tra loro, si può misurare la velocità finale come $v = s/t$, con s proprio la distanza tra le due fotocellule
- ▶ Si lascia andare la sfera da una distanza l dai due traguardi ottici e si registra il tempo di transito t , da cui si ricava v
- ▶ Si ripete per diverse distanze l_i e si costruisce una tabella di coppie spazio-velocità (con le incertezze associate)²: $(l_i \pm \Delta l_i, v_i \pm \Delta v_i)$
- ▶ Attenzione alle incertezze³:
 - ▶ Δl è facile ed è circa la precisione del metro
 - ▶ Δv va propagato: $\Delta v/v = \Delta s/s + \Delta t/t$

²Indichiamo con Δx l'incertezza associata alla misura di x

³Per gli studenti di Lab 1 “capire” le incertezze è fondamentale

Cosa ci si può imparare:

- ▶ Propagazione delle incertezze
- ▶ Fluttuazioni statistiche
 - ▶ L'incertezza Δt_i è maggiore della risoluzione del daq, va stimata dalla dispersione dei dati
 - ▶ Per ogni configurazione, ripetere le misure di tempo più volte: ogni volta la misura sarà leggermente diversa ed è possibile apprezzare le fluttuazioni statistiche
- ▶ Media e lo scarto quadratico medio
- ▶ Verifica del “modello” teorico (la formula 5)
 - ▶ Sia qualitativa che quantitativa:
 - ▶ Verificare che l è proporzionale a v^2
 - ▶ Misurare il coefficiente angolare e confrontarlo con quello atteso
- ▶ Fare e capire grafici:
 - ▶ l vs v^2 o l vs v o l vs t
 - ▶ Anche in carta bilogaritmica
 - ▶ Fit con una retta, con righello o al calcolatore
- ▶ ...

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 e spazio percorso all'istante iniziale anch'esso nullo) sarà allora

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

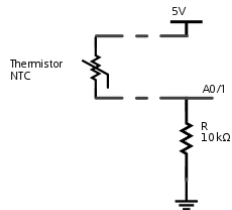
Procedura:

- ▶ Si pone una fotocella in fondo alla guida (dove rimarrà fissa per tutta la durata dell'esperienza); 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 è simile al caso precedente e valgono le stesse considerazioni sulle incertezze, grafici etc.

MISURE DI TEMPERATURA: SCHEMA

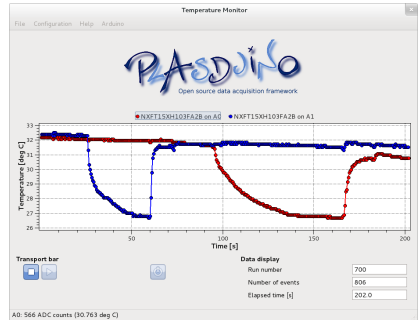
- ▶ Per la misura di temperatura usiamo termistori letti con l'ACD interno di Arduino
- ▶ I termistori possono essere visti come resistenze il cui valore cambia con la temperatura:
<https://it.wikipedia.org/wiki/Termistore>



- ▶ Il più semplice schema per leggerli è tramite un partitore resistivo
- ▶ L'ADC di Arduino legge la tensione A0/1
- ▶ Il software converte la tensione in temperatura tramite la relazione R-T del sensore
 - ▶ Servono le costanti di calibrazione (di solito fornite dal produttore)
 - ▶ I termistori sono fortemente non lineari, la resistenza R dipende dalla scelta del sensore (ma cambiarla è facile)
- ▶ La risoluzione ΔT è circa $0.1 - 0.2 \text{ }^{\circ}\text{C}$, sufficiente per i nostri scopi

MISURE DI TEMPERATURA: ESEMPI PRATICI

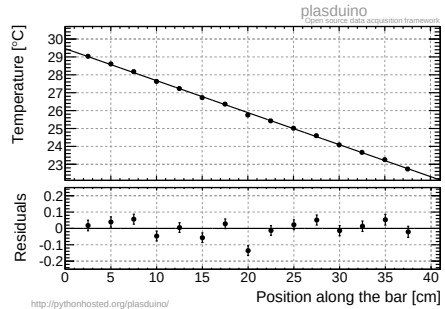
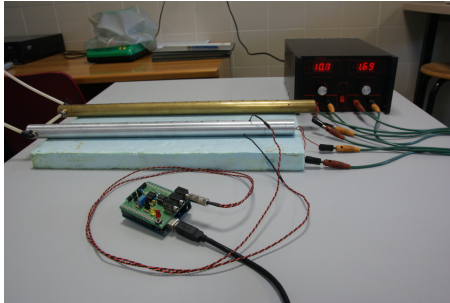
- ▶ Abbiamo implementato la lettura di due termistori, connessi ad un singolo connettore miniJack
- ▶ La temperatura viene letta ad intervalli regolari
- ▶ Plasuino mostra i dati in un grafico e li salva per una successiva elaborazione



Questo è un setup abbastanza generale e si presta a varie esperienze:

- ▶ Calibrazione dello “zero” (0 °C), con acqua distillata in equilibrio tra fase liquida e solida
- ▶ Tempo di rilassamento: il sensore deve andare all'equilibrio termico con il mezzo, $T(t) \sim T_a e^{t/\tau} + T_b$ dove τ dipende dalla capacità termica
- ▶ Conducibilità termica di un metallo (vedi dopo)
- ▶ ...

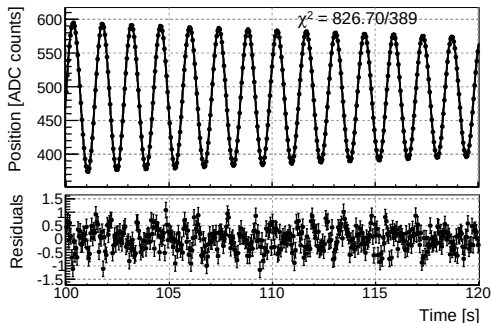
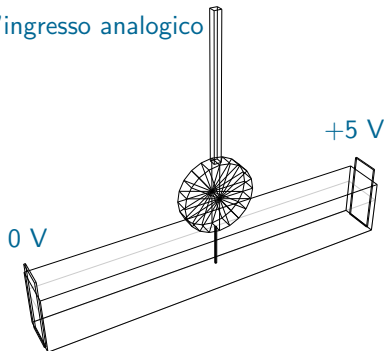
CONDUCIBILITÀ TERMICA DI UNA BARRA METALLICA



- ▶ 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”

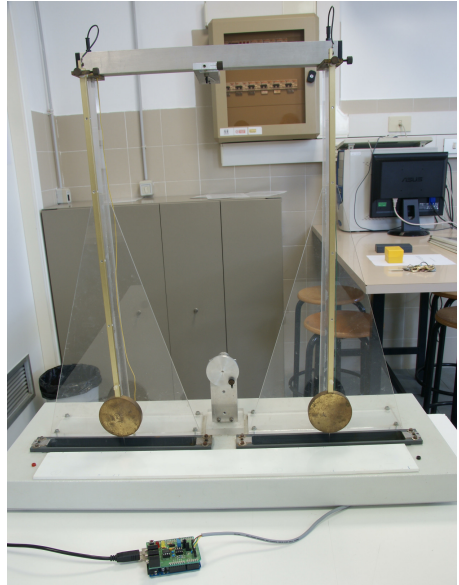
All'ingresso analogico



- ▶ Utilizziamo una punta immersa in acqua come partitore resistivo.
 - ▶ Direttamente ad uno degli ingressi analogici.
- ▶ 10 bit ($2^{10} = 1024$) 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

APPARATO SPERIMENTALE

- ▶ 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
- ▶ L'apparato mostrato in figura è quello presente nei laboratori didattici dell'Università di Pisa



APPARATO SPERIMENTALE

- ▶ 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
- ▶ Oggi vogliamo mostrare come il principio sia molto semplice e la sua realizzazione pratica può essere fatta con materiali facilmente reperibili



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

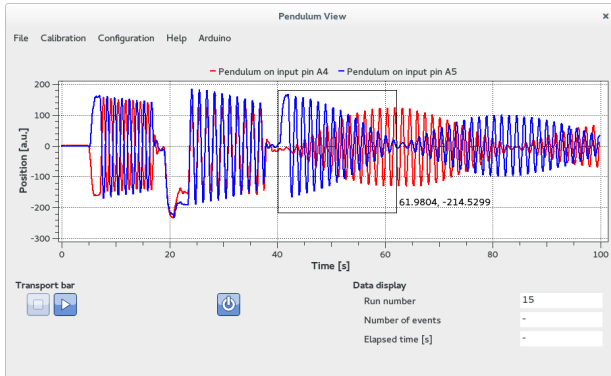
$$x(t) = c_0 + c_1 e^{-\lambda t} \sin(\omega_0 t + \phi_0) \quad (3)$$

Dove ω_0^{th} previsto dalla teoria vale:

$$\omega_0^{th} = \sqrt{\frac{mg/l}{I}} \quad (4)$$

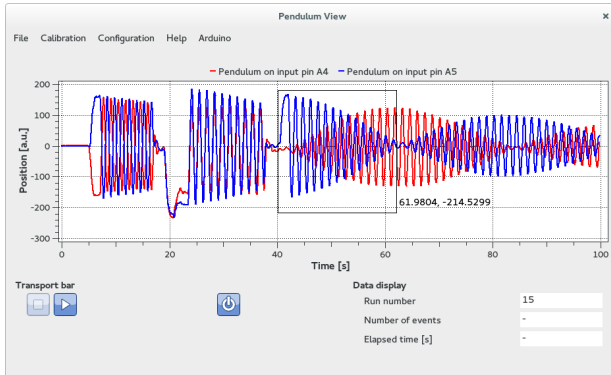
- ▶ La prima misura consiste quindi nel mettere in oscillazione un pendolo da solo e confrontare ω_0 con quello atteso
- ▶ Si può studiare lo smorzamento: è davvero esponenziale?
- ▶ Si possono studiare entrambi i pendoli e controllare che siano uguali (stesso periodo e stesso smorzamento)

COSIDERAZIONI PRATICHE I



- ▶ Il modulo plasduino da utilizzare si chiama “Pendulum View”
- ▶ Questa volta sono presenti i tasti di Start e Stop dell’acquisizione
- ▶ La scala delle y è in unità arbitrarie, si può centrare sullo zero tramite il menù “Calibration”

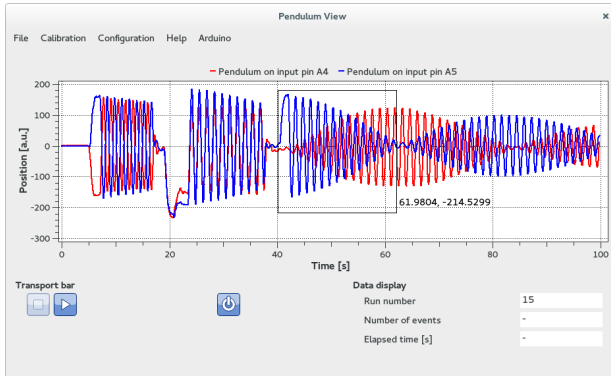
COSIDERAZIONI PRATICHE II



- Quando si preme Stop, i dati vengono salvati in un file di testo contenente quattro colonne che rappresentano, rispettivamente:
 1. il tempo e la posizione del primo pendolo;
 2. il tempo e la posizione del secondo pendolo;

```
#Time A Pos A Time B Pos B
#[s] [au] [s] [au]
0.024 0.92 0.050 -1.00
0.075 0.92 0.101 0.00
...
```

COSIDERAZIONI PRATICHE III



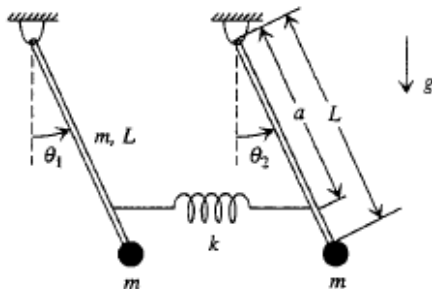
- ▶ Si può utilizzare il modulo Pendulum View per una prima analisi veloce dei dati
 - ▶ Passando il mouse nella finestra grafica vengono visualizzate le coordinate
 - ▶ Si può zoommare selezionando un rettangolo col tasto sinistro del mouse
 - ▶ Si torna alla visualizzazione intera premendo il tasto destro del mouse

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 fatica si risolve (diagonalizza) il sistema e si trova la soluzione. Ma alcune quantità sono difficili da misurare...



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

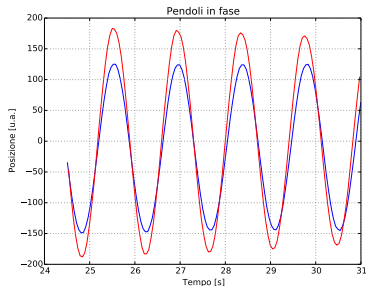
PENDOLI ACCOPPIATI: MODI NORMALI

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:

$$\omega_f \sim \omega_0$$

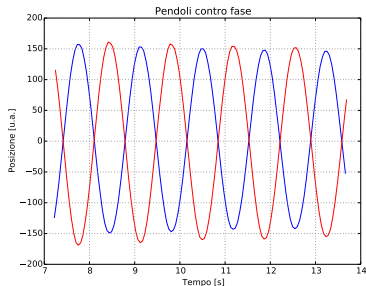


OSCILLAZIONE CONTROFASE

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

$$x(t) = A_0 [\cos(\omega_f t + \phi_1) + \cos(\omega_c t + \phi_2)] \quad (5)$$

o, per le formule di prostaferesi

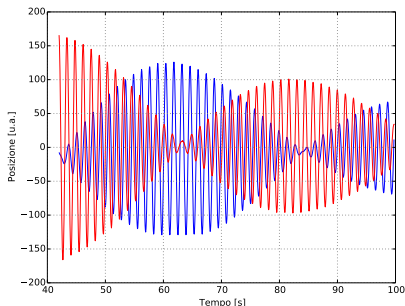
$$x(t) = 2A_0 \left\{ \cos \left[\frac{(\omega_c + \omega_f)t}{2} + \frac{(\phi_2 + \phi_1)}{2} \right] \cos \left[\frac{(\omega_c - \omega_f)t}{2} + \frac{(\phi_2 - \phi_1)}{2} \right] \right\}$$

L'oscillazione risultante, di pulsazione angolare portante

$$\omega_p = \frac{(\omega_c + \omega_f)}{2} \approx \omega_c, \omega_f \quad (6)$$

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

$$\omega_b = \frac{(\omega_c - \omega_f)}{2} \ll \omega_c, \omega_f \quad (7)$$



plasduino

Open source data acquisition framework

Home

Installation

Overview

Modules

Shields

Core development

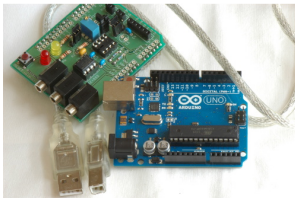
HOME

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/>
- Articolo di riferimento: <http://arxiv.org/abs/1312.1805>
- Download: <https://bitbucket.org/lbaldini/plasduino/downloads>
- Issue tracker: <https://bitbucket.org/lbaldini/plasduino/issues>

► Ogni manifestazione di interesse è benvenuta.