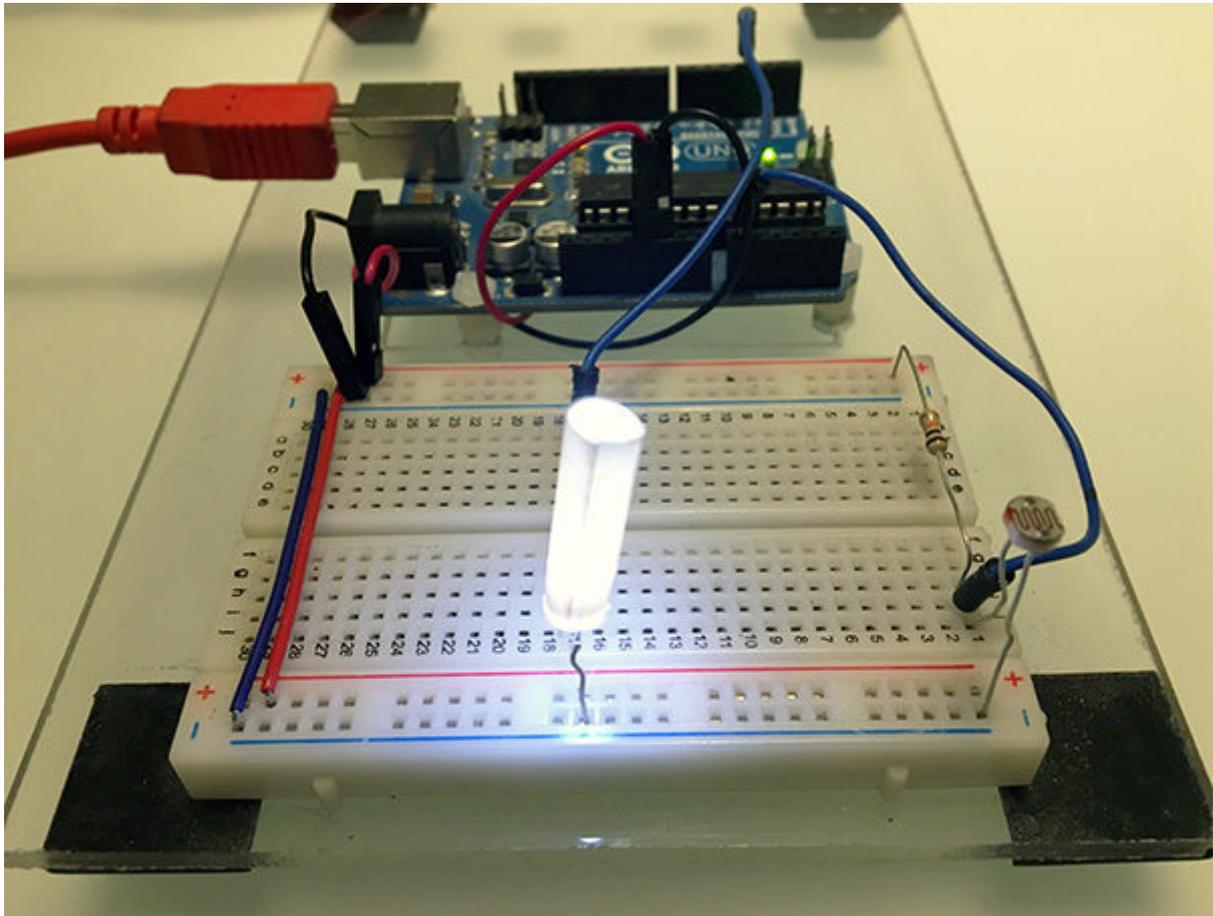


# PID library arduino

---

[mauroalfieri.it/elettronica/pid-library-arduino.html](http://mauroalfieri.it/elettronica/pid-library-arduino.html)

PID Library arduino è uno degli argomenti che spesso mi viene richiesto nei corsi o nella realizzazione di progetti.



Il PID ( Proporzionale Integrale Derivato ) rappresenta un argomento abbastanza ostico sia per la complessità dei calcoli sia per i molti modi in cui può essere applicato ad uno sketch.

Cercando on-line delle soluzioni da impiegare in u progetto mi sono imbattuto nella PID library arduino di Brett Beauregard.

## PID library arduino di Brett Beauregard

---

La PID library arduino la puoi trovare nel repository dell'autore: [github di Brett Beauregard](#)

br3ttb / Arduino-PID-Library

Watch 139 Star 692 Fork 420

Code Issues (23) Pull requests (0) Projects (0) Insights

**Join GitHub today**  
 GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.  
[Sign up](#)

*No description, website, or topics provided.*

16 commits 1 branch 2 releases 3 contributors

Branch: master New pull request Find file Clone or download

br3ttb version	Latest commit sb4cate on 20 Jun 2017
examples	Version 1.2 7 months ago
PID_v1.cpp	version / license 7 months ago
PID_v1.h	version / license 7 months ago
README.txt	version 7 months ago
keywords.txt	Version 1.2 7 months ago
library.json	moved library code back to root from sub-folder 3 years ago
library.properties	version 7 months ago

```

*****
* Arduino PID Library - Version 1.2.1
* by Brett Beauregard <br3ttb@gmail.com> brettbeauregard.com
*
* This Library is licensed under the MIT License
*****

```

che oltre alla libreria vera e propria contiene degli esempi di utilizzo.

## Documentazione della PID Library arduino sul playground

Sul [playground arduino](#) trovi la documentazione relativa alla PID library arduino:

Nella pagina di documentazione leggi che il PID è un algoritmo basato sulla correzione dell'errore dato un **INPUT** al fine di raggiungere il **setpoint** desiderato controllando l'**OUTPUT**.

Un esempio interessante descritto nella stessa pagina chiarisce bene l'idea:

*For reference, in a car, the Input, Setpoint, and Output would be the speed, desired speed, and gas pedal angle respectively.*

Ossia se ti riferissi ad un'automobile in cui l'INPUT è la velocità ed il setpoint è la velocità a cui desideri andare l'OUTPUT l'angolo da imporre al pedale dell'acceleratore.

Modificando quest'ultimo valore ( output ) varia la velocità e di conseguenza si riduce o aumenta l'errore che fa discostare la velocità rilevata ( input ) e quella desiderata ( setpoint ).

## Il costruttore PID library arduino

Il costruttore della libreria si chiama `PID()`, ed accetta i parametri descritti nel paragrafo precedente oltre ad altri che ti servono per raggiungere il risultato desiderato:

- **Input:** di tipo double, è la variabile contenente la rilevazione eseguita dal sistema;
- **Output:** di tipo double, è la variabile con cui imponi un comando al sistema;
- **Setpoint:** di tipo double, è il punto di arrivo desiderato;
- **Kp, Ki, Kd:** di tipo double  $\geq 0$ , rappresentano le tre costanti mediante cui l'algoritmo modifica il proprio comportamento per raggiungere il punto desiderato, solitamente sono calcolati empiricamente;

- **Direction:** puoi definirla DIRECT o REVERSE e determina la direzione in cui si sposterà l'output in funzione dell'errore rilevato per raggiungere il setpoint;
- **POn:** Abilita la P\_ON\_E di default o P\_ON\_M ( Proportional on Measurement ).

Negli esempi che leggerai della PID Library arduino lasceremo il parametro POn sempre impostato a default.

## Esempio Basic PID

---

Il primo esempio proposto dall'autore della libreria è definito Basic per la semplicità dell'esempio e serve per introdurre l'utilizzo della libreria:

```

01  #include <PID_v1.h>
02  double Setpoint, Input, Output;
03  PID myPID(&Input, &Output, &Setpoint,2,5,1, DIRECT);
04  void setup() {
05  Serial.begin(9600);
06  Input = (analogRead(A0)/4);
07  Setpoint = 100;
08  myPID.SetMode(AUTOMATIC);
09  }
10  void loop() {
11  Input = (analogRead(A0)/4);
12  myPID.Compute();
13  analogWrite(3,Output);
14  Serial.print(Input); Serial.print( " " ); Serial.println(Output);
15  }
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

```

iniziando dalla linea 06: includi la libreria PID;

linea 09: definisci 3 variabili globali di tipo double;

linea 12: crea una istanza *myPid* a cui passi, per riferimento, i valori definiti alla linea 09 ed i valori iniziali di Kp,Ki,Kd oltre alla direzione;

linea 15: attiva la comunicazione seriale a 9600 baud, questa linea l'ho aggiunta allo sketch per visualizzare il dato sul monitor seriale;

linea 17: nella funzione di setup() leggi il valore di Input dal pin analogico A0 dividendolo per 4 in quanto come sai il convertitore AD di Arduino è a  $2^{10}$  ( 1024 valori ) e l'output è  $2^8$  ( 256 valori );

linea 18: imposta il setPoint a 100;

linea 21: metti il PID in modalità automatica usando il metodo setMode();

linea 25: ad ogni loop() leggi il valore di INPUT, sempre adattandolo da 1024 a 256 valori come alla linea 17;

linea 26: usa il metodo Compute() per calcolare l'output dato l'INPUT ed il setPoint;

linea 27: invia al pin 3 il valore di output appena calcolato;

linea 29: ho aggiunto la linea corrente per scrivere sul monitor seriale i valori di INPUT e OUTPUT;

## Il Video Demo

---

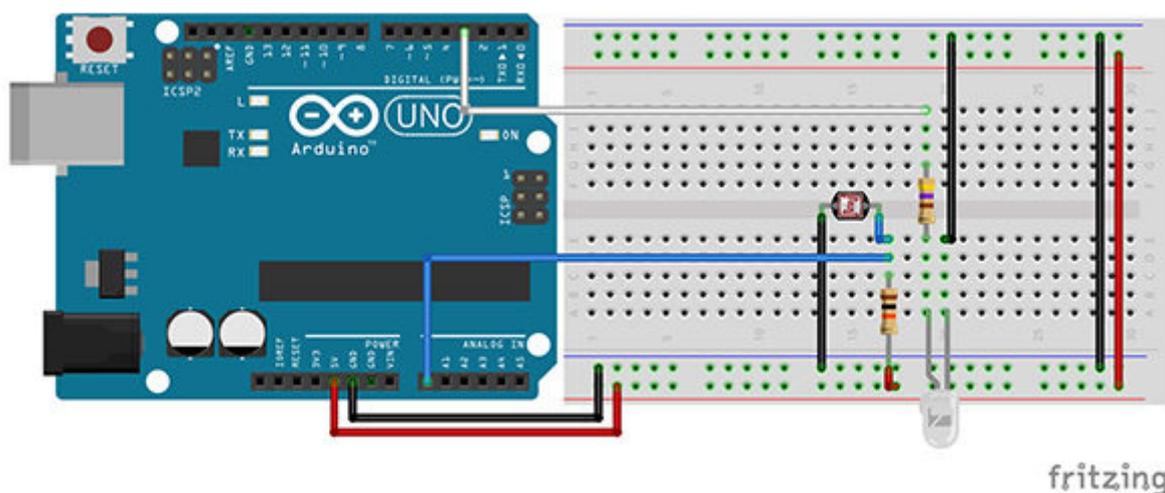
Per realizzare il video demo durante il test della libreria PID library arduino ho usato una fotoresistenza ed un led.

Coprendo con un foglio la fotoresistenza il valore rilevato si incrementa ed la PID library arduino entra in funzione per cercare di “fare luce” accendendo proporzionalmente il led:

## Il circuito del PID library Arduino usato nel test

---

Il circuito che ho utilizzato è molto semplice e fa solo uso di una fotoresistenza da 20Kohm e di un a resistenza di pull-up da 10Kohm per la parte di rilevazione della luminosità ambientale:



ed una resistenza da 470ohm per la protezione del led ad alta luminosità controllato dal pin 3.

