

Pràctiques PID amb Arduino

Joan Martínez Domene

18 de març de 2013

Índex

1 Muntatge físic del circuit	1
1.1 Convertir 0..10V a 0..5V	1
1.2 Convertir el PWM a senyal 0..10V	1
1.2.1 Filtrat R-C	2
1.2.2 Amplificador X2	3
2 Instruccions per usar el programa	3

1 Muntatge físic del circuit

Per tal de connectar el Arduino amb la maqueta del levitador aerodinàmic, es requereix una adaptació entre les entrades i les sortides de Arduino. Volem connectar la sortida de la placa Arduino a l'entrada de control de la maqueta de levitació aerodinàmica (Connexió de PWM a 0..10 VDC), i connectar la sortida de la senyal del sensor del levitador a l'entrada analògica del Arduino (0..10VDC a 0..5VDC).

1.1 Convertir 0..10V a 0..5V

Aquesta és la part més simple de la adaptació entre la maqueta i la placa de Arduino. El cor de la placa Arduino és el microcontrolador Atmega328. Donat que la impedància de entrada del convertidor ADC en el Atmega328P té un valor elevat, de $100M\Omega$ segons les característiques, podem considerar que aquesta entrada no consumirà corrent. Siguen així podem definir un senzill divisor resistiu amb dues resistències de $100K\Omega$ en sèrie i prenent la mostra del punt mig, tal com queda il·lustrat en la Figura 1.

1.2 Convertir el PWM a senyal 0..10V

Per adaptar una senyal PWM i convertir-la en senyal de tensió continua de 0 a 10 V es poden utilitzar diferents mètodes. No obstant en el nostre cas tenim diferents

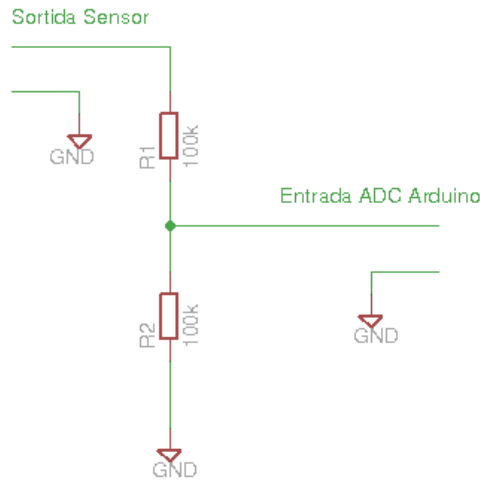


Figura 1: Divisor resistiu per adaptar la sortida de la maqueta a l'entrada de Arduino

limitacions amb les que s'ha de comptar per tal d'escollir la millor manera i més simple possible.

Cal tenir en compte algunes especificitats d'on volem connectar el nostre circuit adaptador.

- No hi ha fonts d'alimentació simètriques. Únicament podem aprofitar la sortida simple regulada entre $-10V$ i $+10V$.
- L'entrada de control de la maqueta té una impedància d'entrada no infinita. De fet té una impedància $\leq 20K\Omega$.

Aquests dos factors ens limiten la forma en que plantejarem el disseny de la etapa de adaptació.

De fet donat que la impedància d'entrada està acotada, sembla que la opció de posar una etapa amplificadora a la sortida és la més raonable. Conseqüentment, primerament utilitzarem una etapa de filtrat R-C i seguidament l'etapa amplificadora.

1.2.1 Filtrat R-C

Es connecta una etapa R-C a la sortida del terminal de sortida PWM de Arduino. El senyal PWM de l'Arduino té una freqüència (funcionant amb la configuració per defecte) d'uns 490 Hz. Amb aquest valor de freqüència tenim que el cicle té un període de 2 ms. Amb un petit filtrat hauria de ser suficient per poder mantenir la sortida PWM a un valor estable, amb un nivell de rissat mínim. Després de realitzar diferents experiments, s'ha comprovat que els valors de $R = 100k$ i $C = 470nF$ anulen pràcticament el nivell de rissat. La constant de temps, τ , del conjunt R-C és de 56ms. Aquest valor podria ocasionar problemes d'estabilitat quan es treballi en llaç tancat,

per tant caldrà ajustar els valors de R i C per obtenir els millors resultats, caldrà un compromís entre la velocitat de resposta i baix nivell de rissat.

1.2.2 Amplificador X2

L'etapa amplificadora la construïrem emprant un Amplificador Operacional Rail-to-Rail, de manera que la sortida de l'amplificador pugui aprofitar tot el marge de tensió que ens proporcioni la font d'alimentació disponible (teòricament de $0V$ a $10V$). L'amplificador escollit és el *LT1490ACN8*. Aquest es configura com a amplificador no inversor amb les dues resistències d'igual valor. Amb això s'ha aconseguit un guany de 2.

El circuit teòric complet és el que es mostra en la Figura 2.

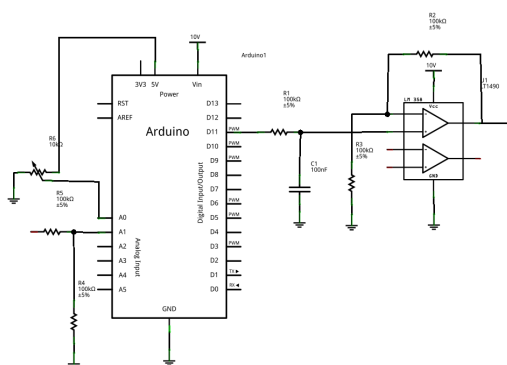


Figura 2: Esquema teòric del adaptador

El circuit que cal muntar en la protoboard es mostra en la Figura 3.

2 Instruccions per usar el programa

La llibreria que implementa el PID s'ha fet amb llenguatge C . Per facilitar-ne el seu ús s'utilitzarà l'Entorn de programació de Arduino (IDE).

Primerament cal tenir l'entorn de programació instal·lat en el ordinador (*ArduinoIDE*). Aquest, es pot trobar en el repositori d'Ubuntu. Al instal·lar-se es crea un directori de treball, on es deixen els projectes. Aquest directori està normalment a $/home/USUARI/sketchbook$.

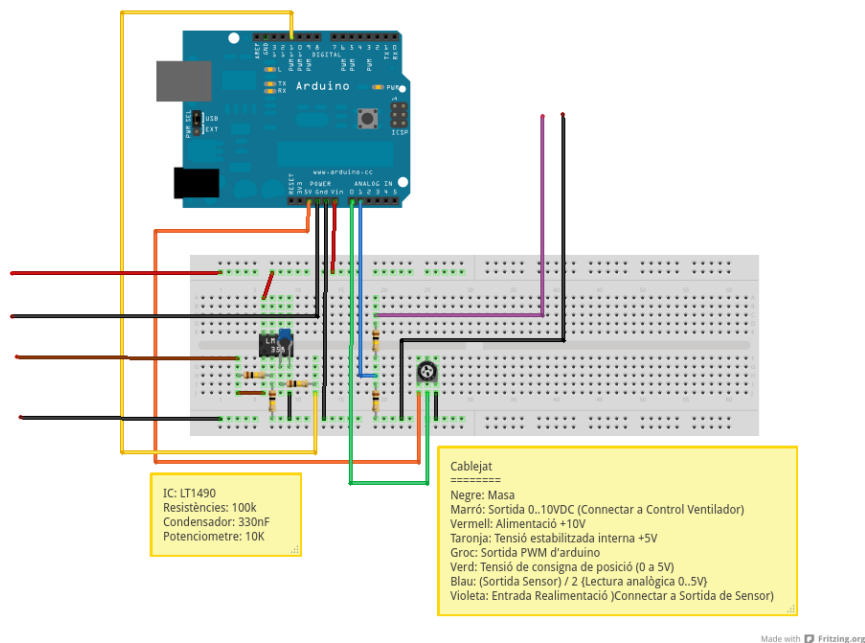


Figura 3: circuit complet per adaptar els senyals entre Arduino i la maqueta

Després d'instal·lar l'entorn de programació cal descarregar els mòduls de control del PID. Aquests mòduls es poden trobar a: <http://escriny3.epsem.upc.edu/projects/engeeknyer/repository>. Els arxius a descarregar són *PID.c*, *PID.h* i *DIPSE_PID.ino*.

- *PID.c*: És el mòdul que implementa les funcionalitats de control del PID. Aquestes funcionalitats implementades actuen sobre un tipus de estructura de dades que simbolitza un sistema PID de control.
- *PID.h*: Incorpora les capçaleres i definicions necessàries per el mòdul del PID.
- *DIPSE_PID.ino*: Aquest és el programa principal, amb el qual s'utilitza el mòdul de control i s'implementa físicament a partir de la lectura de una consigna d'entrada, la entrada de realimentació i la sortida de control analògic.

L'arxiu *DIPSE_PID.ino* el copiarem directament en el directori de treball (sketch-book). Els altres dos arxius cal copiar-los en el directori que crearem expresament dins

de sketchbook: *libraries/PID/*. Una vegada s'han copiat aquests dos fitxers (*PID.c* i *PID.h*) dins del directori creat s'ha de renombrar el arxiu *PID.c* a *PID.cpp*. És a dir, es renombra per tal que el fitxer sigui reconegut com a *C++*.

Una vegada completats els passos anteriors, podem posar en marxa l'Arduino IDE i carregar-hi el programa (*Fitxer > sketchbook > DIPSE_PID*).

La primera línia de codi és:

```
Set_Constants_PID(&PID1, 0.75, 8.5 , 0.0001);
```

El primer paràmetre de la funció és la variable del tipus PID que utilitzarem. Aquesta és una estructura de dades que accepta les funcions típiques d'un sistema PID, concretament el PID implementat és de la forma:

$$output(t) = K_P \Delta e(t) + K_I \int_0^t e(p) dp + K_D \frac{de(t)}{dt} \quad (1)$$

Els tres següents són els paràmetres K_P , K_I i K_D , els quals podrem mirar d'ajustar per jugar amb l'estabilitat del sistema.

Índex alfabètic

tableofcontents, 1