Ariadna Fernández Martínez Ignacio Anguita Espadaler

Laboratori de circuits electrònics

No quitar esto por si hay que añadir algo más en la portada

Projecte de Circuits Electrònics

Control de la inclinació d'un braç giratori accionat per un conjunt hèlix-motor

# Índex

[Introducció 2](#_Toc501897953)

[Realització 3](#_Toc501897954)

[Primera fase 3](#_Toc501897955)

[Segona fase 6](#_Toc501897956)

[Tercera fase 9](#_Toc501897957)

[Millores teòriques proposades 14](#_Toc501897958)

[Millora 1: Lectura automàtica del valors màxim i mínim de la posició 14](#_Toc501897959)

[Millora 2: Càlcul automàtic del controlador PID 16](#_Toc501897960)

[Conclusions i comentaris 18](#_Toc501897961)

# Introducció

L'objectiu del projecte és el control, usant una targeta Arduino UNO, de la inclinació d'un braç giratori accionat per un conjunt hèlix-motor (figura 1).

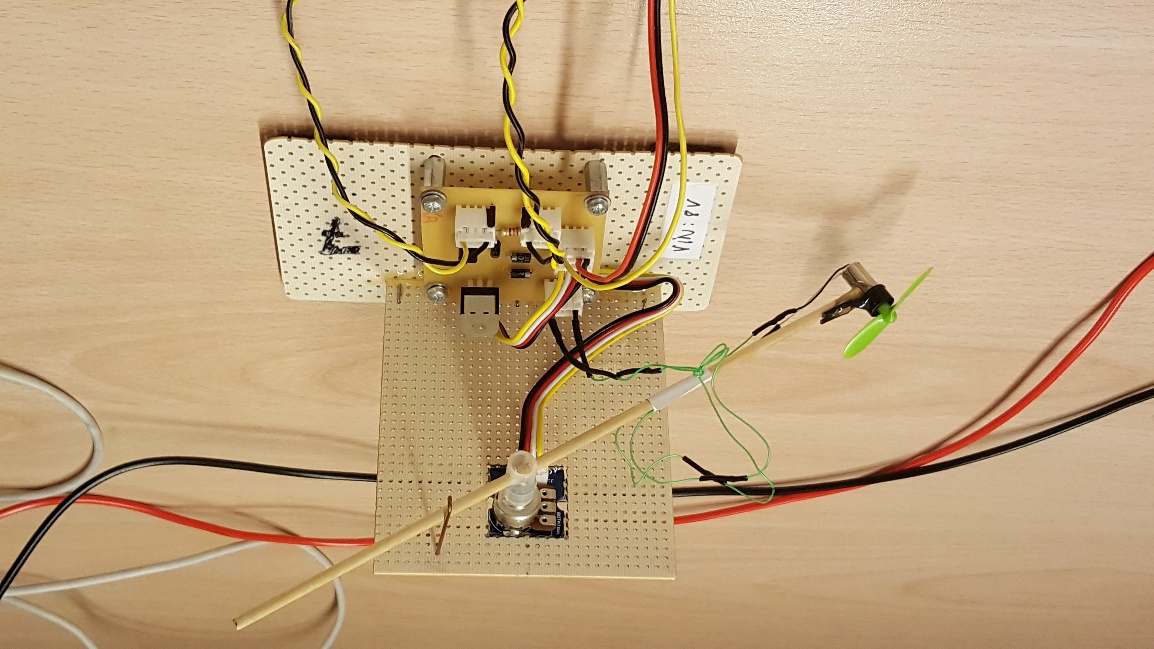


Figura 1: Conjunt hélix-motor.

El braç esta sotmès a l’acció de dos parells horitzontals i de signe oposat (el generat per l'empenta causada per la rotació de l’hèlix i el degut a l’acció de la forca de la gravetat sobre el motor) i girarà en un sentit o en un altre segons quin d’ells sigui major.

A banda del braç i del conjunt hèlix-motor, el sistema inclou un sensor d’inclinació i una targeta d'accionament i sensat.

El sensor d'inclinació del braç és un potenciòmetre connectat de forma que en aplicar 5 V entre els seus terminals fixes, es genera en el terminal variable una tensió que canvia linealment amb la inclinació. La targeta d'accionament i sensat permet excitar el sensor i governar el motor.

La velocitat de rotació del motor (i, per tant, de l'hèlix) es controlarà amb una tensió PWM (Pulse Width Modulation) de freqüència ≈ 500 Hz, valor mínim 0 V i valor màxim 5 V.

En la figura 2, aquesta tensió és Vcont i correspon a la tensió amb el mateix nom de les figures 3 i 4 . La tensió Vcont controla l’estat del transistor de la targeta d’adquisició i control (vegeu les figures 3 i 4), el qual actua com un interruptor: el transistor condueix (interruptor tancat) quan Vcont=5V i no ho fa (interruptor obert) quan Vcont=0V . En cada cicle (durada T=1/500 ), la tensió Vcont val 5 V al llarg de D·T unitats de temps consecutives, on D , 0≤D≤1 , és l’anomenat cicle de treball, i val 0 V al llarg de (1−D) ·T unitats de temps consecutives . En conseqüència, la tensió mitjana d’alimentació del motor val D·VCC.

Llavors, donat que la velocitat de rotació d’aquest creix -de forma no lineal- amb la seva tensió mitjana d’alimentació, modificant el cicle de treball D entre 0 i 1 pot fer-se variar la velocitat de rotació del motor entre els seus valors mínim (motor aturat) i màxim.



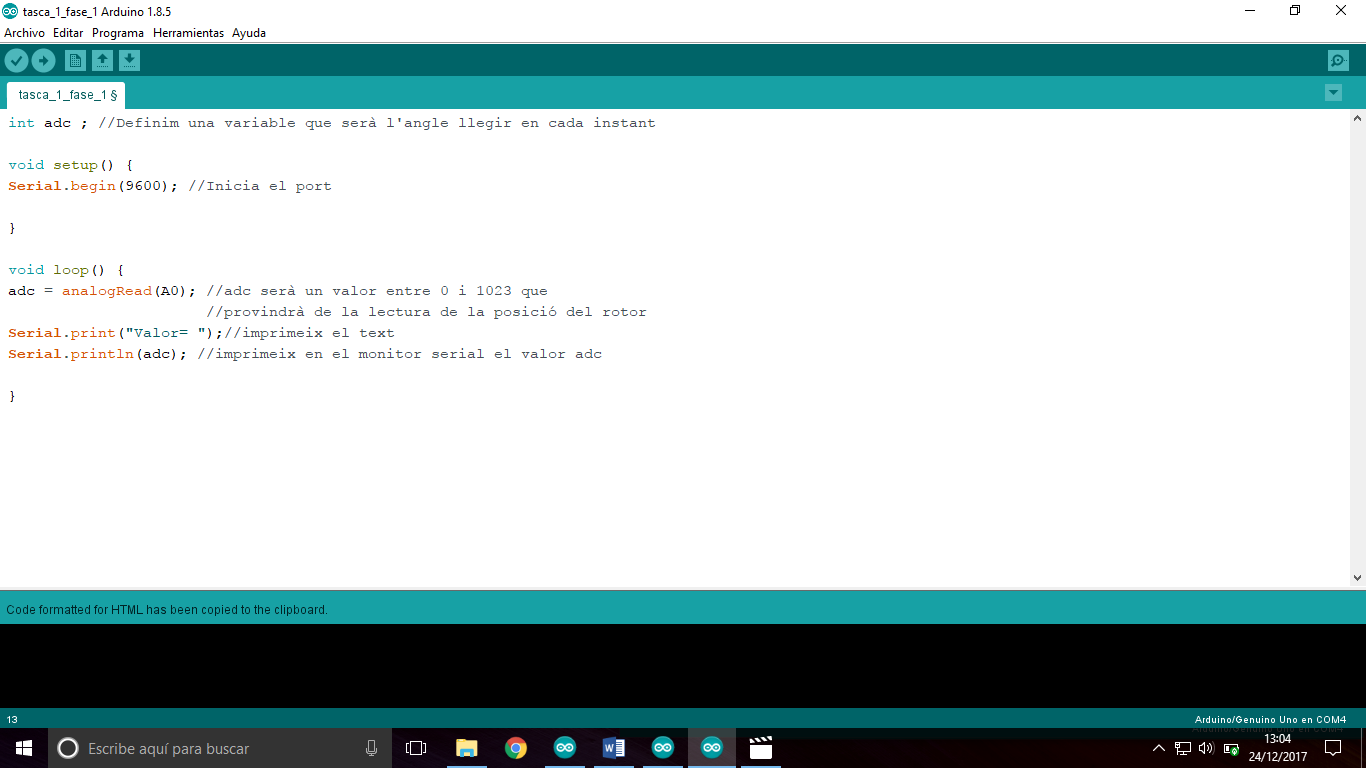
Figura 2: Senyal PWM de control de la velocitat de rotació del motor.

# Realització

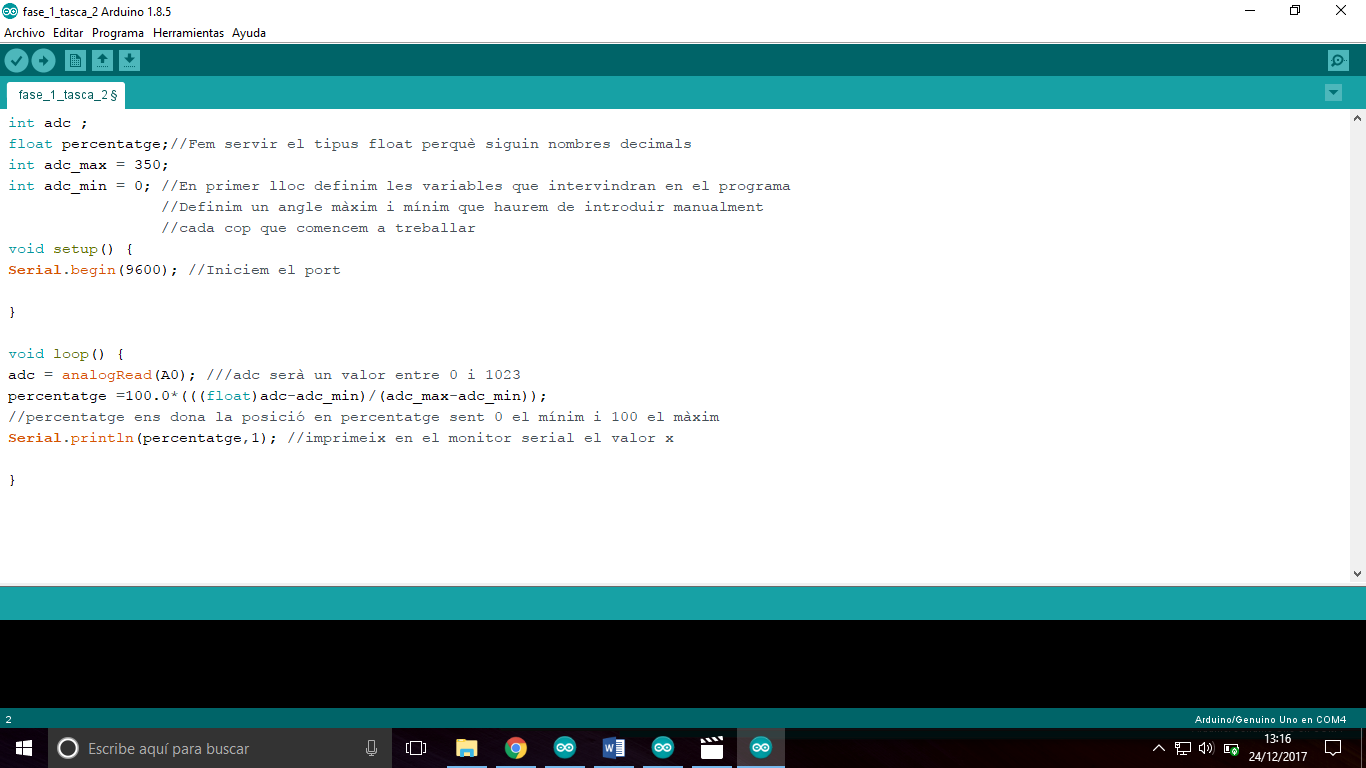
Es procedirà per fases. Tot seguit es detallen les tasques a realitzar en cada fase amb els corresponents codis.

## Primera fase

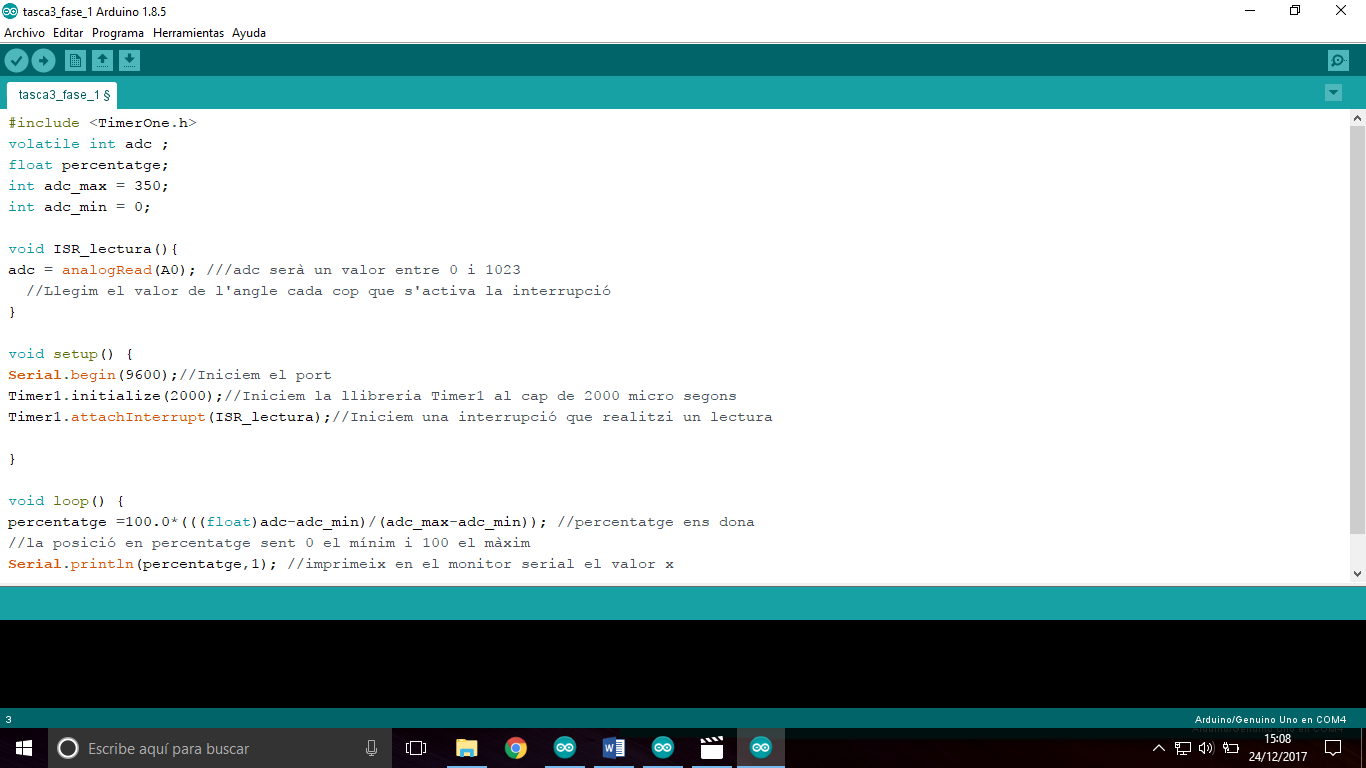
1. **Mostrar en el monitor sèrie de l'entorn de desenvolupament de la targeta Arduino UNO (d'ara en endavant, monitor sèrie) l’enter que resulta de digitalitzar, usant el convertidor A/D incorporat a la targeta, la tensió Vsens (vegeu les figures 3 i 4) quan la inclinació del braç giratori és mínima i quan és màxima.**

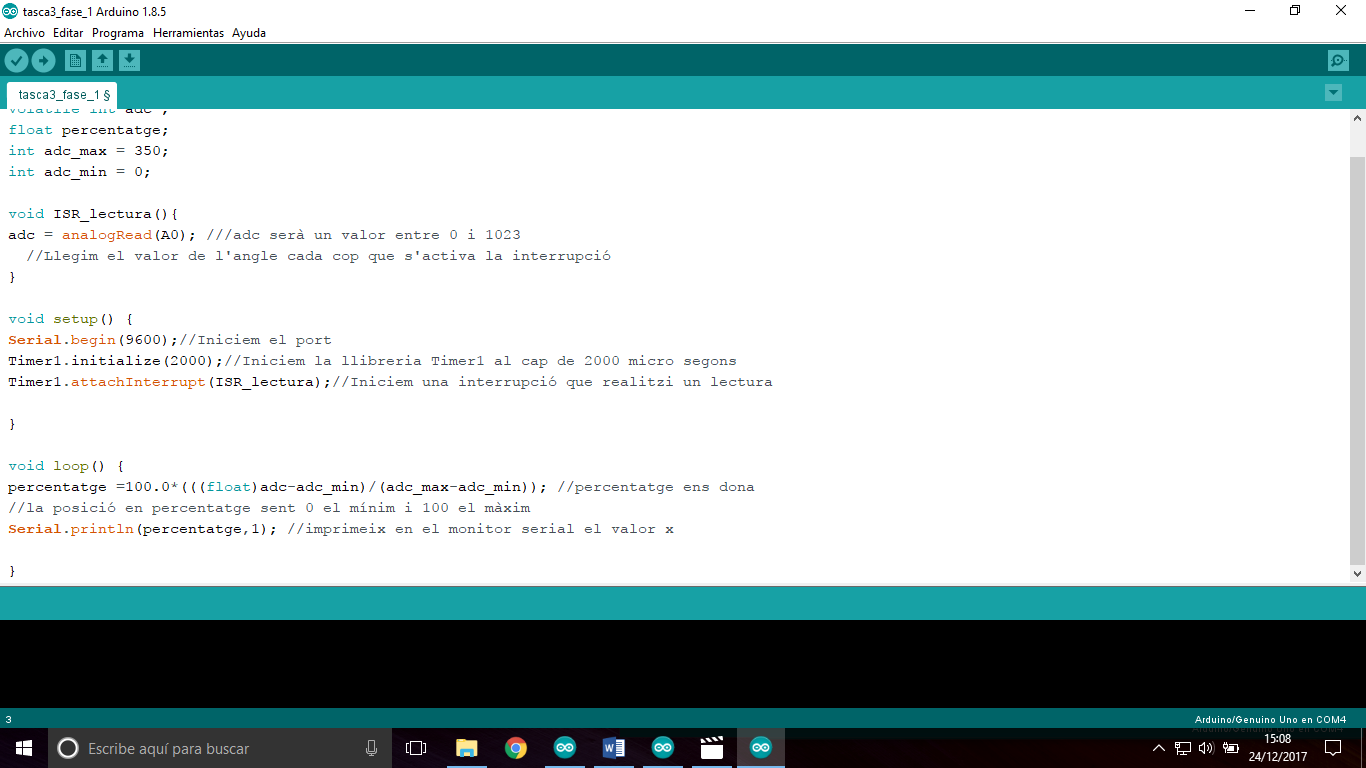


1. **Mostrar en el monitor sèrie la inclinació del braç giratori expressada en tant per cent del canvi d'inclinació màxim. No es controlarà la freqüència de mostreig.**

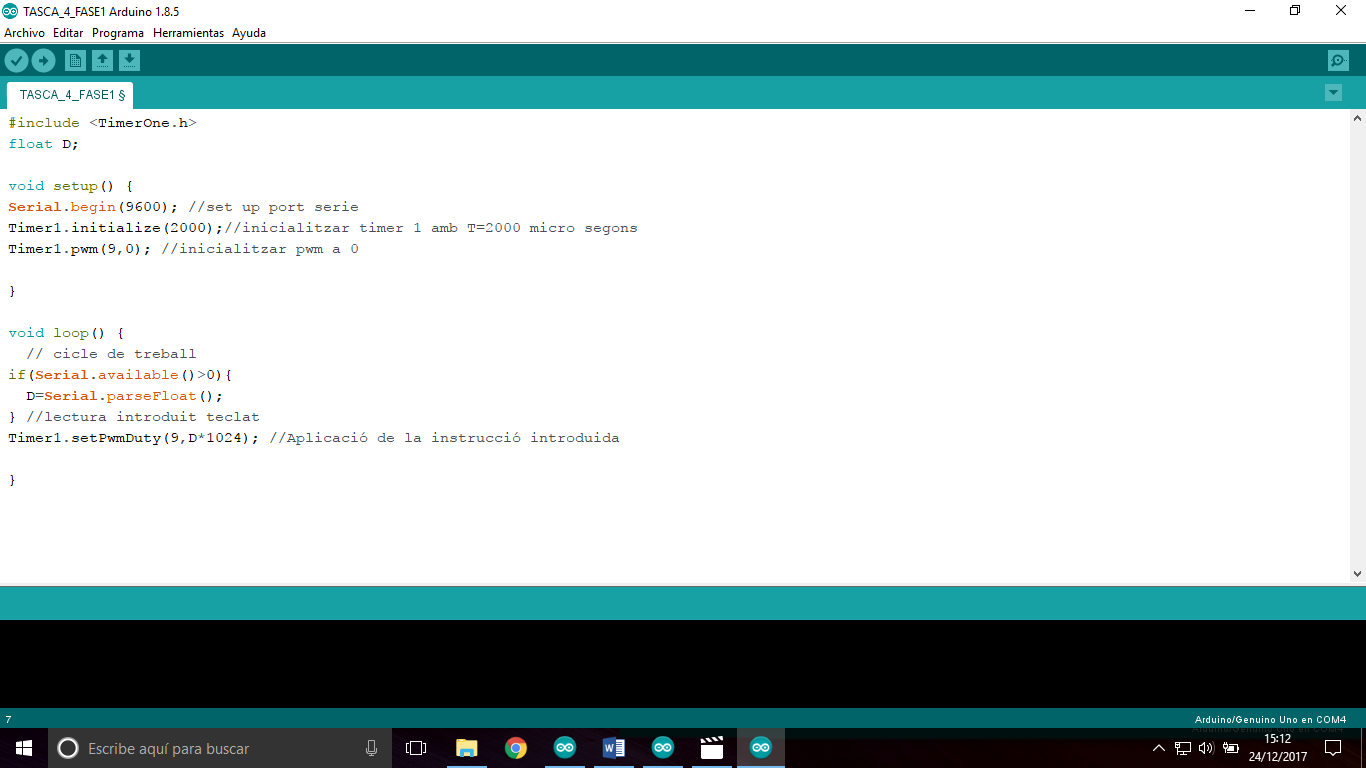


1. **Mostrar en el monitor sèrie la inclinació del braç giratori expressada en tant per cent del canvi d'inclinació màxim. La freqüència de mostreig haurà de ser ≈ 500 Hz.**





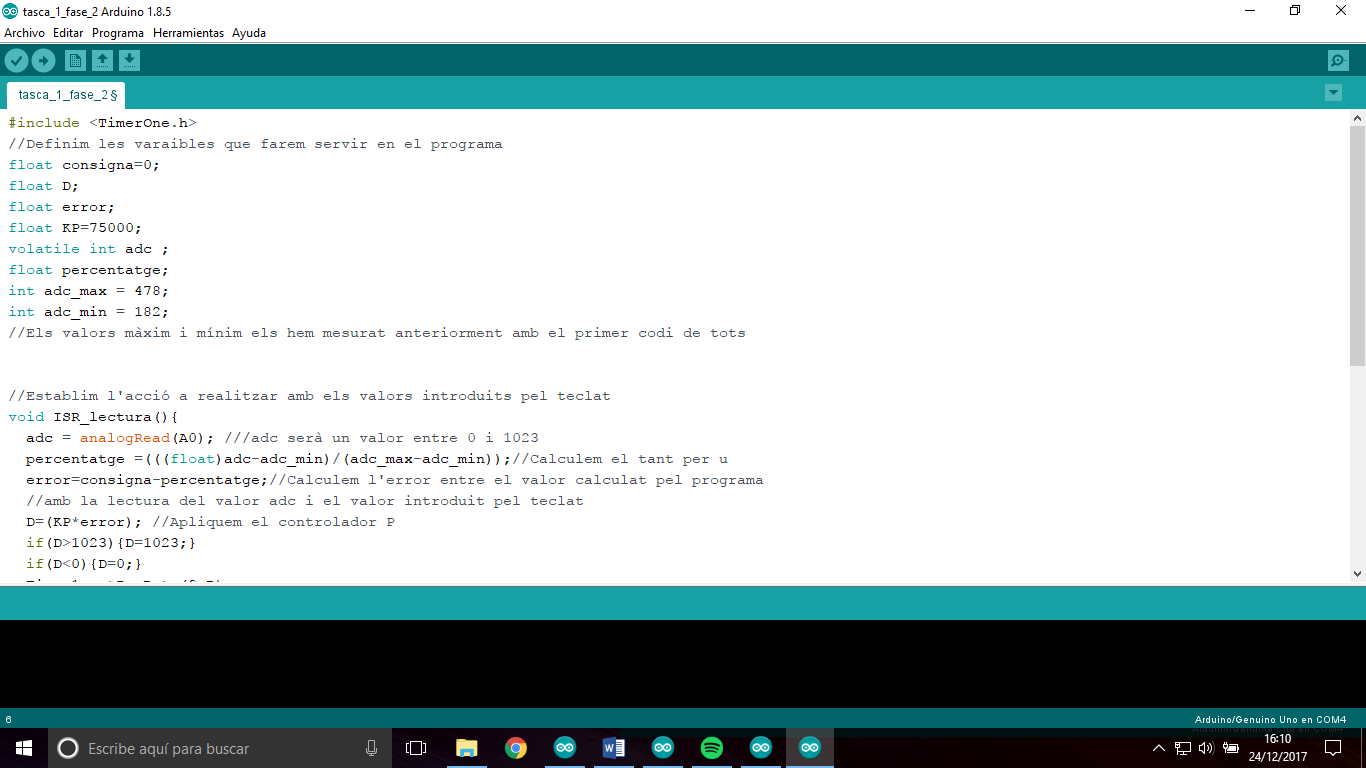
1. **Generar la tensió Vcont i accionar el motor amb velocitat variable. El cicle de treball s'introduirà mitjançant el monitor sèrie en forma d'enter comprès entre 0 (correspondrà a D=0 ) i 100 (correspondrà a D=1 ).**

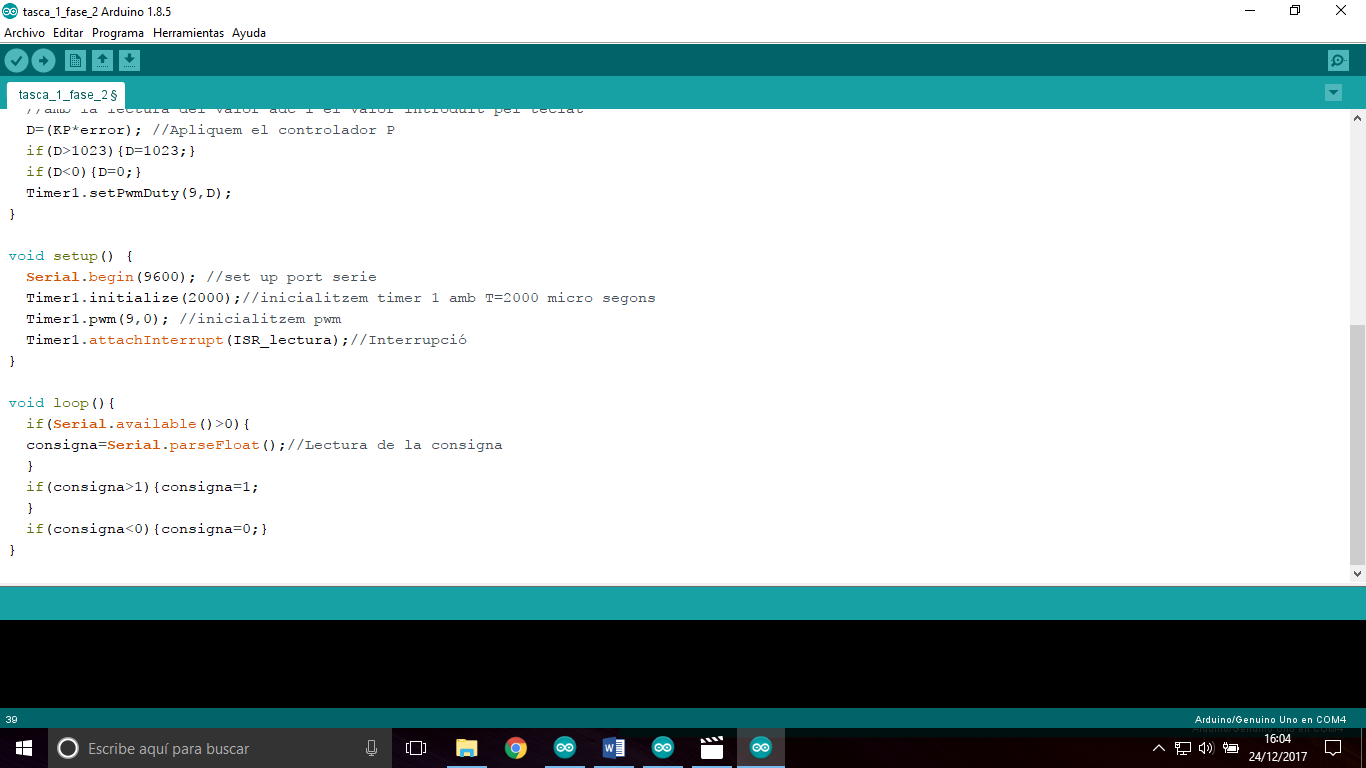


## Segona fase

1. **Implementar un control proporcional de la inclinació del braç. La freqüència amb què es mostrejarà la inclinació i s’actualitzarà el cicle de treball del senyal Vcont serà de ≈ 500 Hz. La inclinació de referència o consigna s'introduirà a través del monitor sèrie en forma d'enter comprès entre 0 (0% d’inclinació) i 100 (100% d’inclinació).**
2. **Ajustar el valor del guany proporcional usant el mètode de Zieger-Nichols (enllaç Wikipedia) i comprovar experimentalment el funcionament del control ajustat.**

Per a realitzar aquestes dues tasques fem servir un sol programa. Partim de Kp=0 i anem augmentant fins a obtenir una oscil·lació sostinguda en el temps. Buscant una oscil·lació sostinguda en el temps podrem trobar els paràmetres per a aconseguir que el sistema sigui ràpid i estable més endavant, quan implementem un controlador PID.





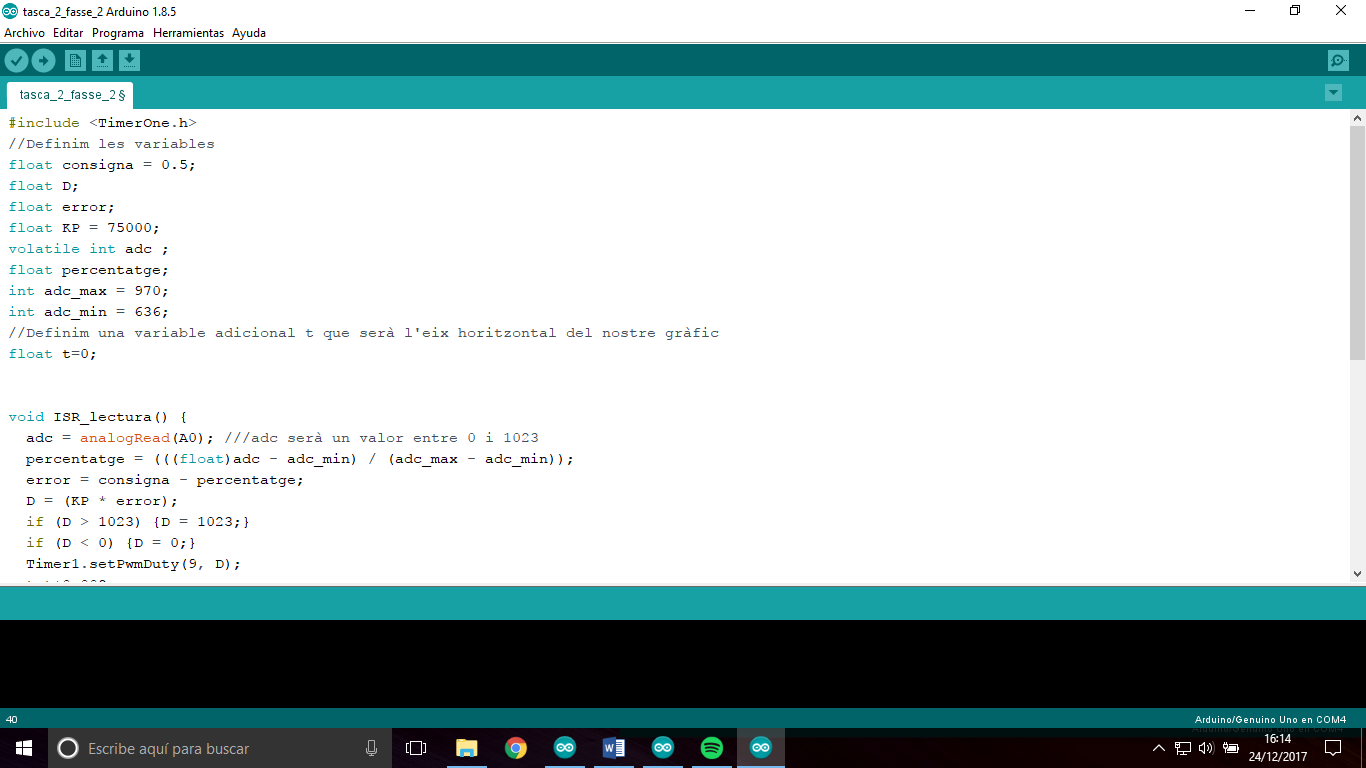
Si fixem el valor de la consigna a l’inici al 50%, obtenim la següent reacció del sistema. Podem veure que l’oscil·lació al principi s’atenua una mica, però es sosté en el temps.

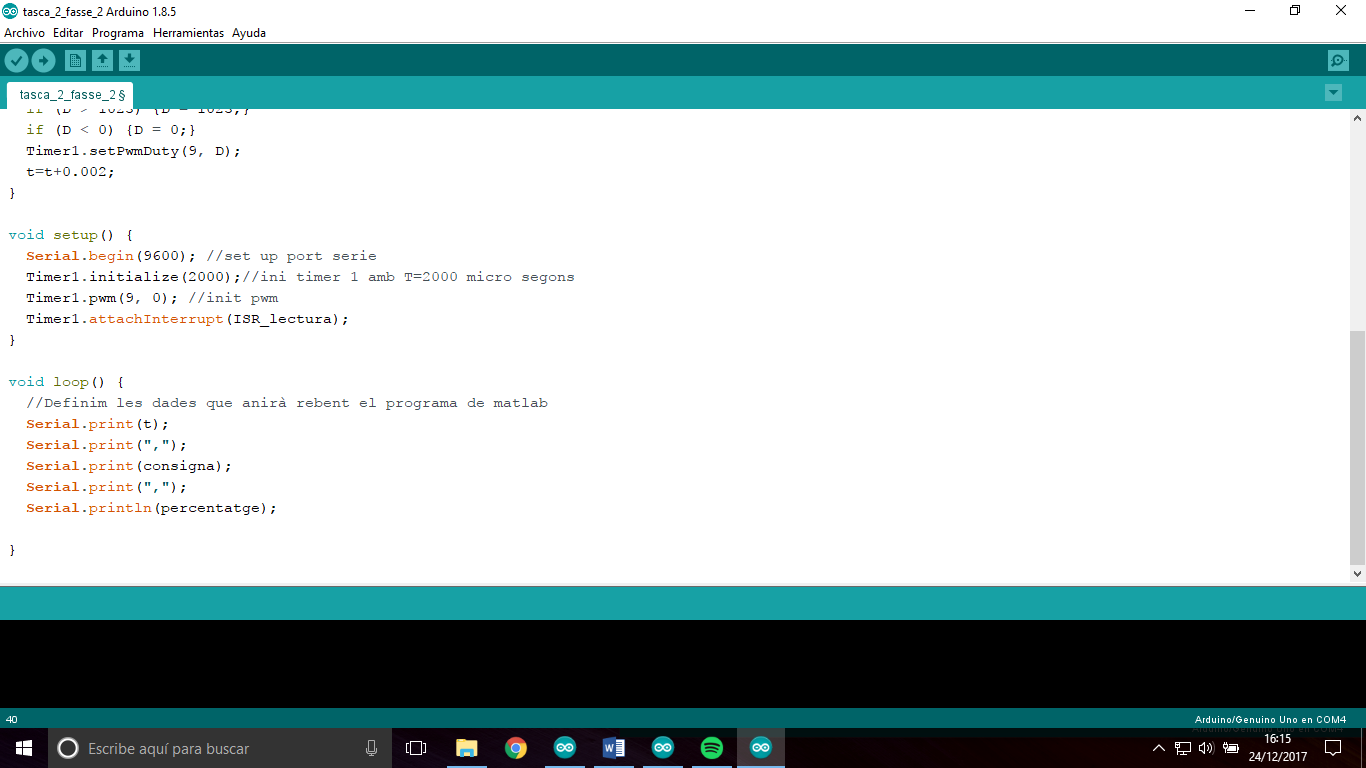


Figura 3: Representació temporal de la posició al iniciar el programa amb la consigna fixada al 50%.

Per a obtenir aquesta representació hem modificar lleugerament el codi anterior de forma que al carregar el programa en la placa, en lloc de controlar el sistema a través de la consigna, matlab el ‘segrestés’ i reproduís el seu funcionament. La posició l’obtenim en tant per u.

El codi és el següent:

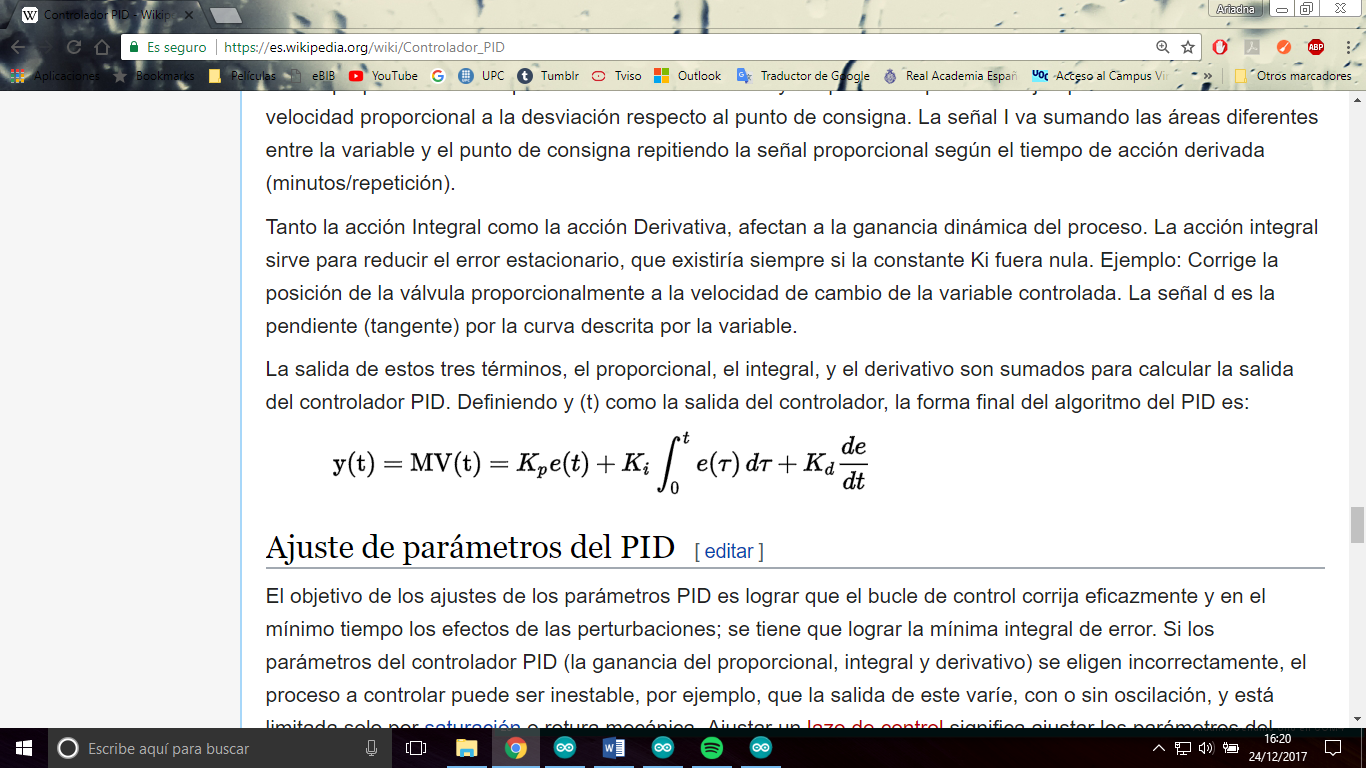




## Tercera fase

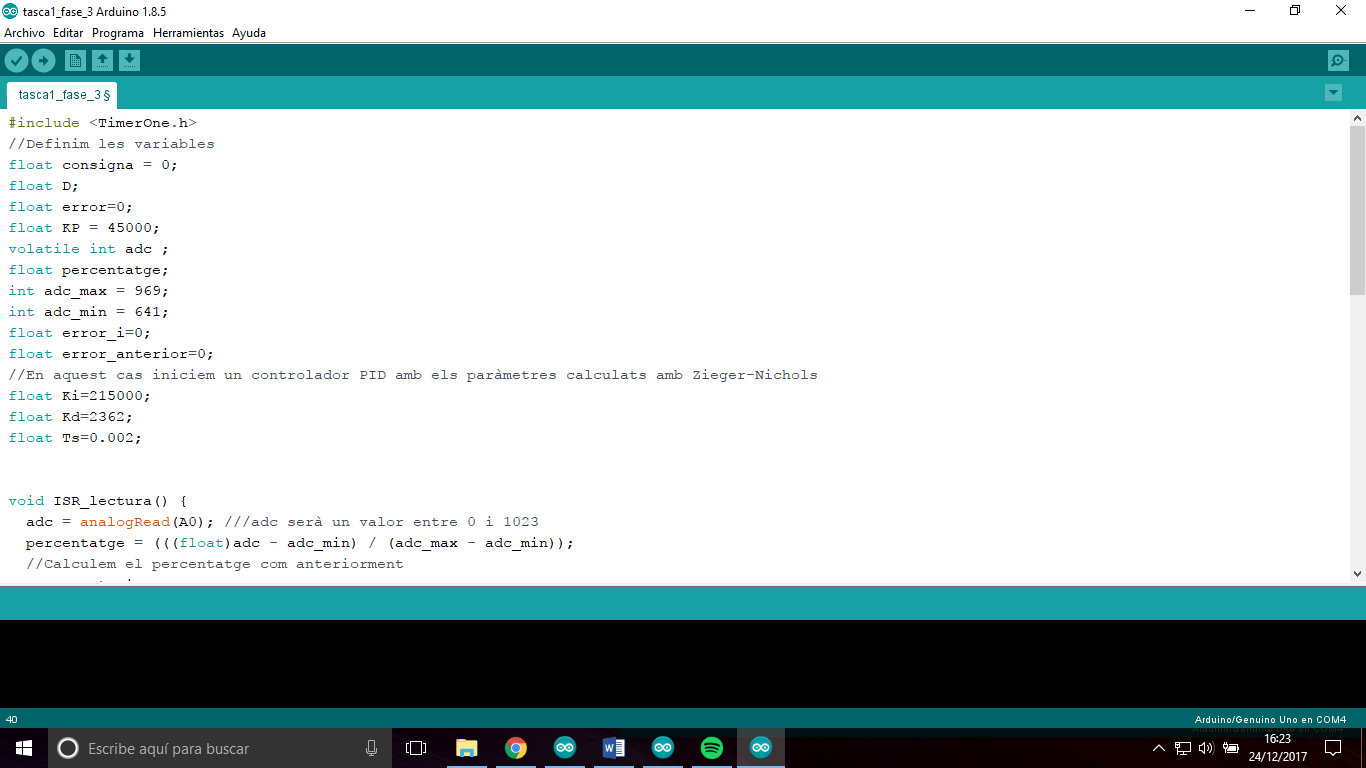
1. **Implementar un control proporcional-integral-derivatiu de la inclinació del braç. La freqüència amb què es mostrejarà la inclinació i s’actualitzarà el cicle de treball del senyal Vcont serà de ≈ 500 Hz. La inclinació de referència o consigna s'introduirà a través del monitor sèrie en forma d'enter comprès entre 0 (0% d’inclinació) i 100 (100% d’inclinació).**

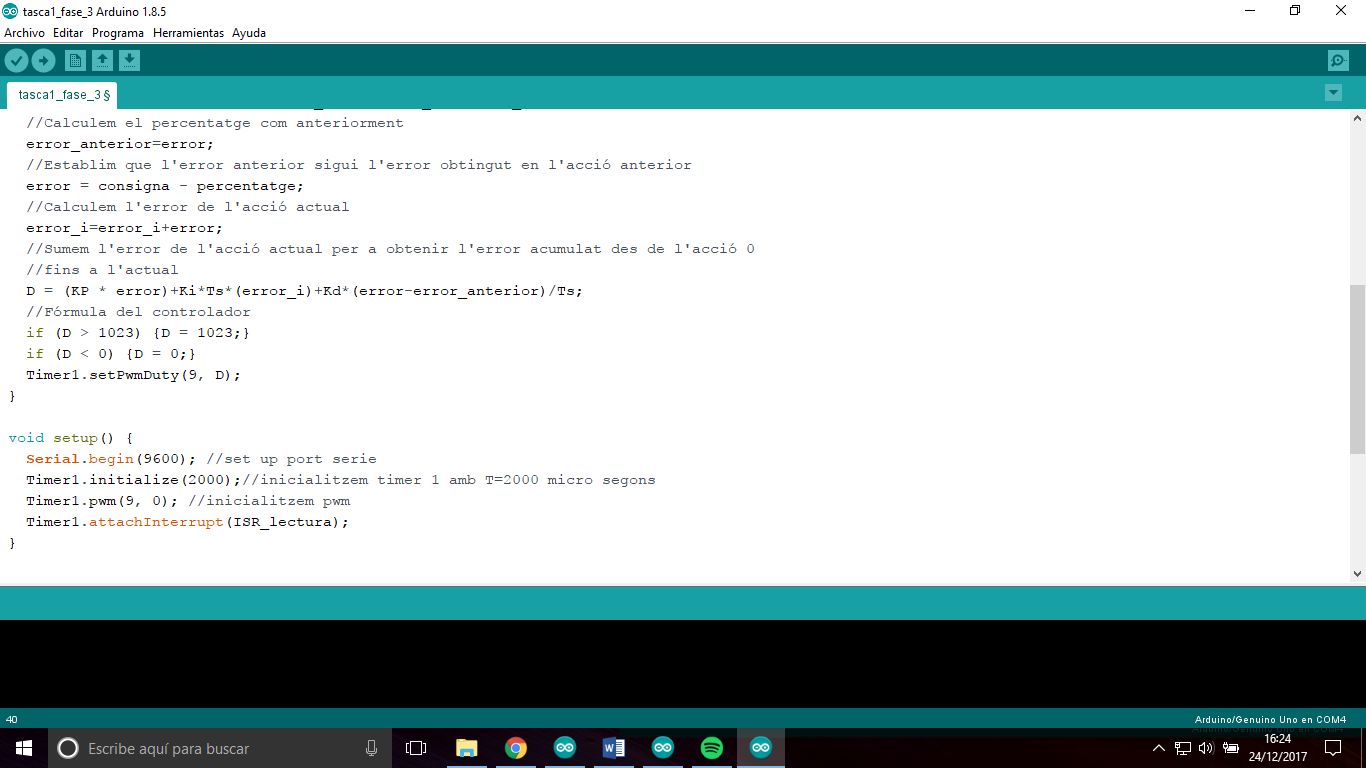
La forma del controlador que implementarem serà la següent:

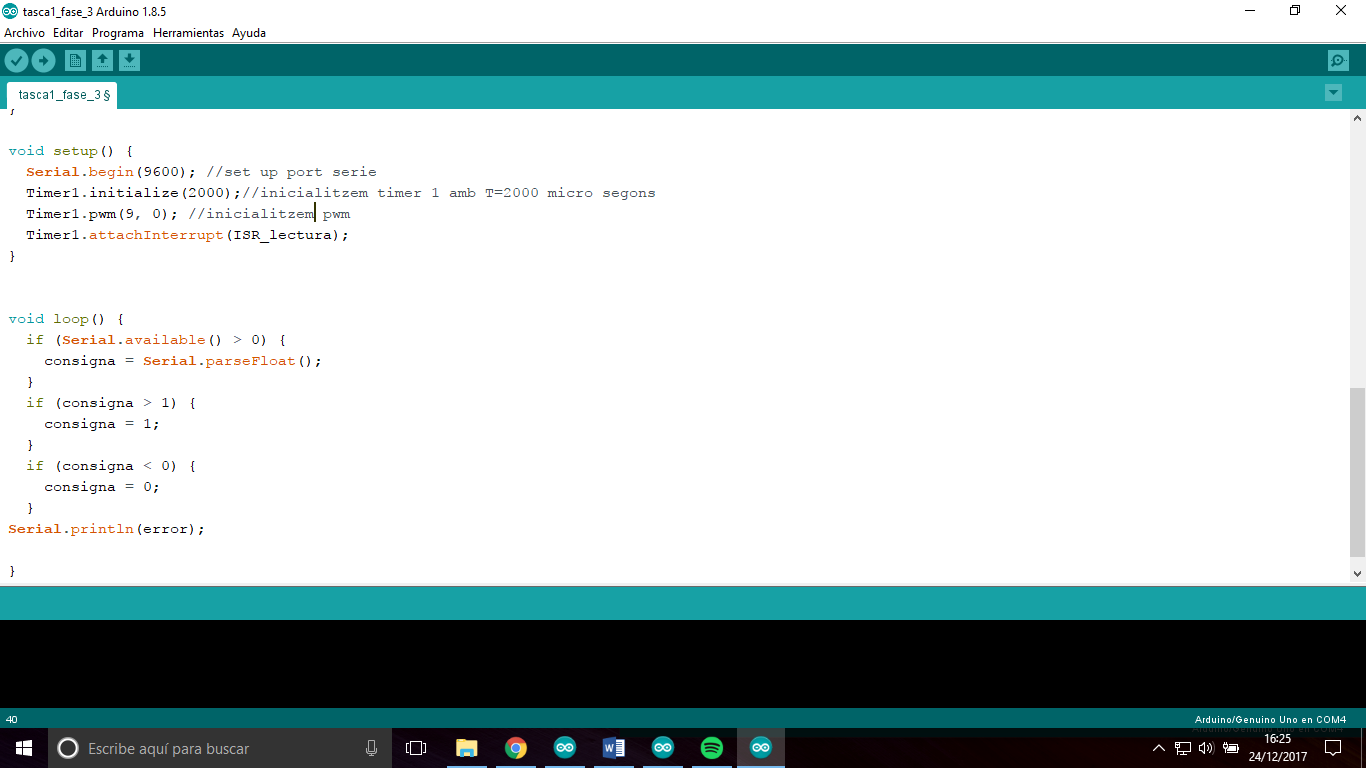


En nostre codi però la integral estarà aproximada a un sumatori de l’error acumulat i la derivada estarà aproximada a un diferencial de l’error.

El codi emprat és el següent:







Per a comprovar la validesa del nostre controlador vam estudiar la rapidesa i l’estabilitat de la resposta del motor a aquests valors de les constants de proporcionalitat.

Per a fer-ho vam fer com al cas anterior, vam fixar el valor de la consigna, vam eliminar l’acció d’introduir valors pel teclat i vam afegir la impressió de dades necessàries per a matlab.

Els resultats són els següents:



Figura 4:Resposta del sistema amb un valor de consigna del 25% i el controlador PID del codi anterior.



Figura 5:Resposta del sistema amb un valor de consigna del 50% i el controlador PID del codi anterior.

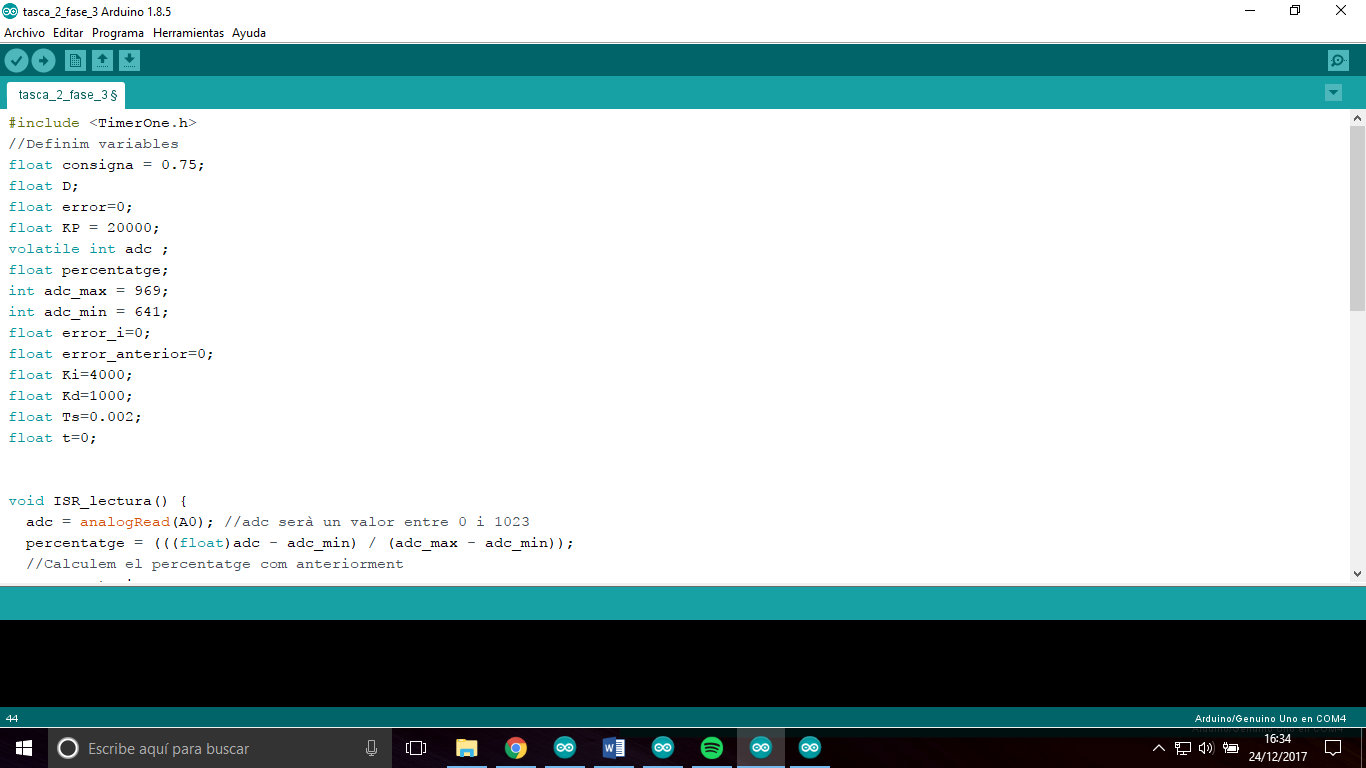


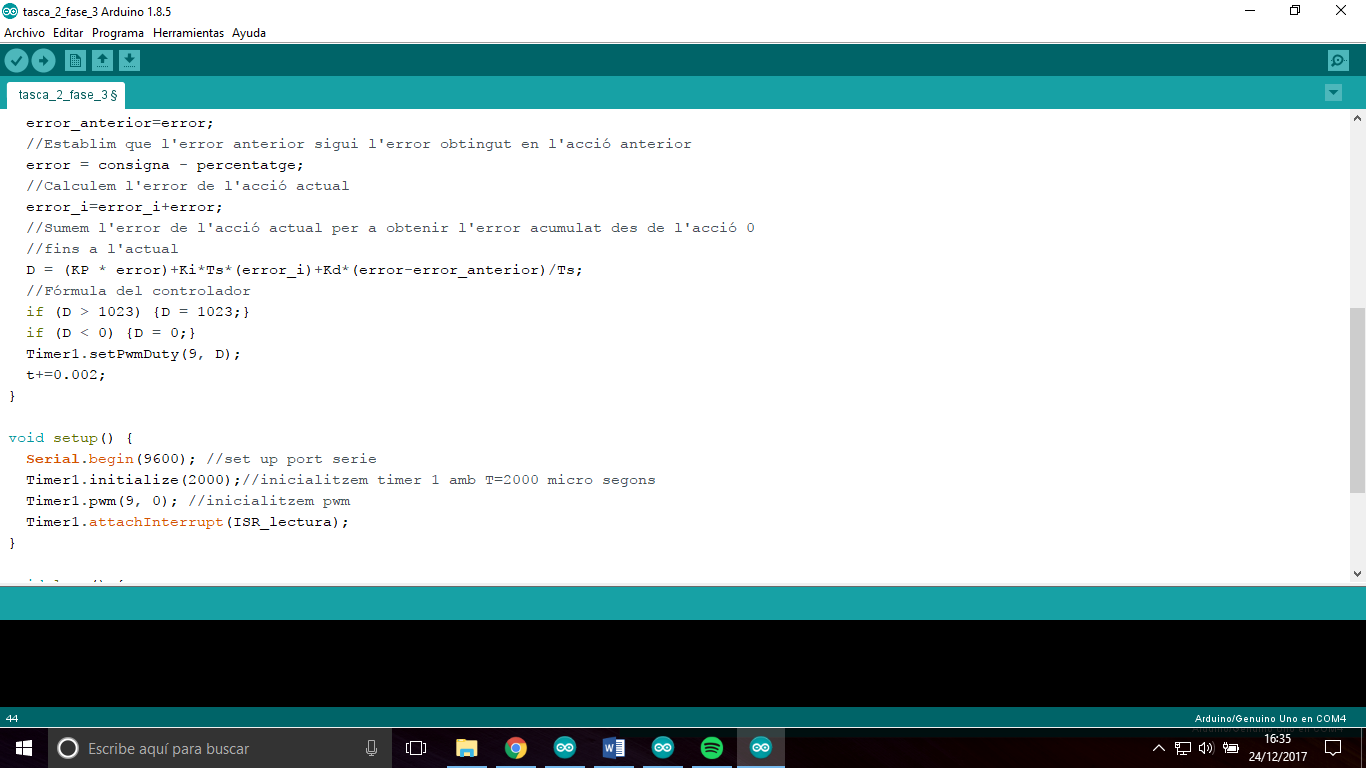
Figura 6:Resposta del sistema amb un valor de consigna del 75% i el controlador PID del codi anterior.

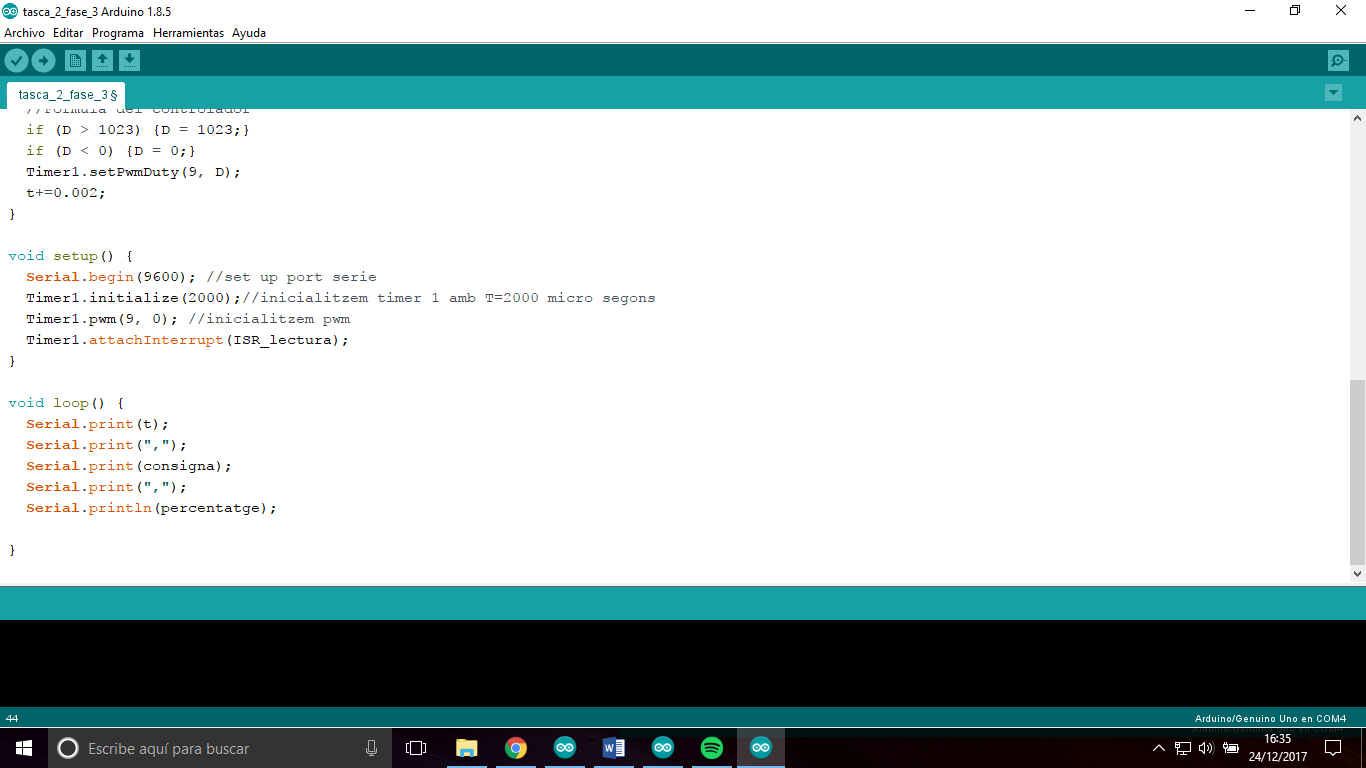
Resulta evident que el sistema és ràpid però gens estable. Presenta en els tres casos un important sobrepic a l’inici que triga força temps a estabilitzar-se. Per aquest motiu vam procedir a ajustar els valors del controlador al següent apartat.

1. **Ajustar el valor dels guanys usant el mètode de Zieger-Nichols (enllaç Wikipedia) i comprovar experimentalment el funcionament del control ajustat.**

En aquest punt únicament hem fet proves amb matlab i amb la consigna fixada perquè ja havíem comprovat que funcionava bé experimentalment.







Els resultats obtinguts són els següents:

****

Figura 7:Resposta del sistema amb un valor de consigna del 25% i el controlador PID ajustat del codi anterior.

****

Figura 8:Resposta del sistema amb un valor de consigna del 50% i el controlador PID ajustat del codi anterior.

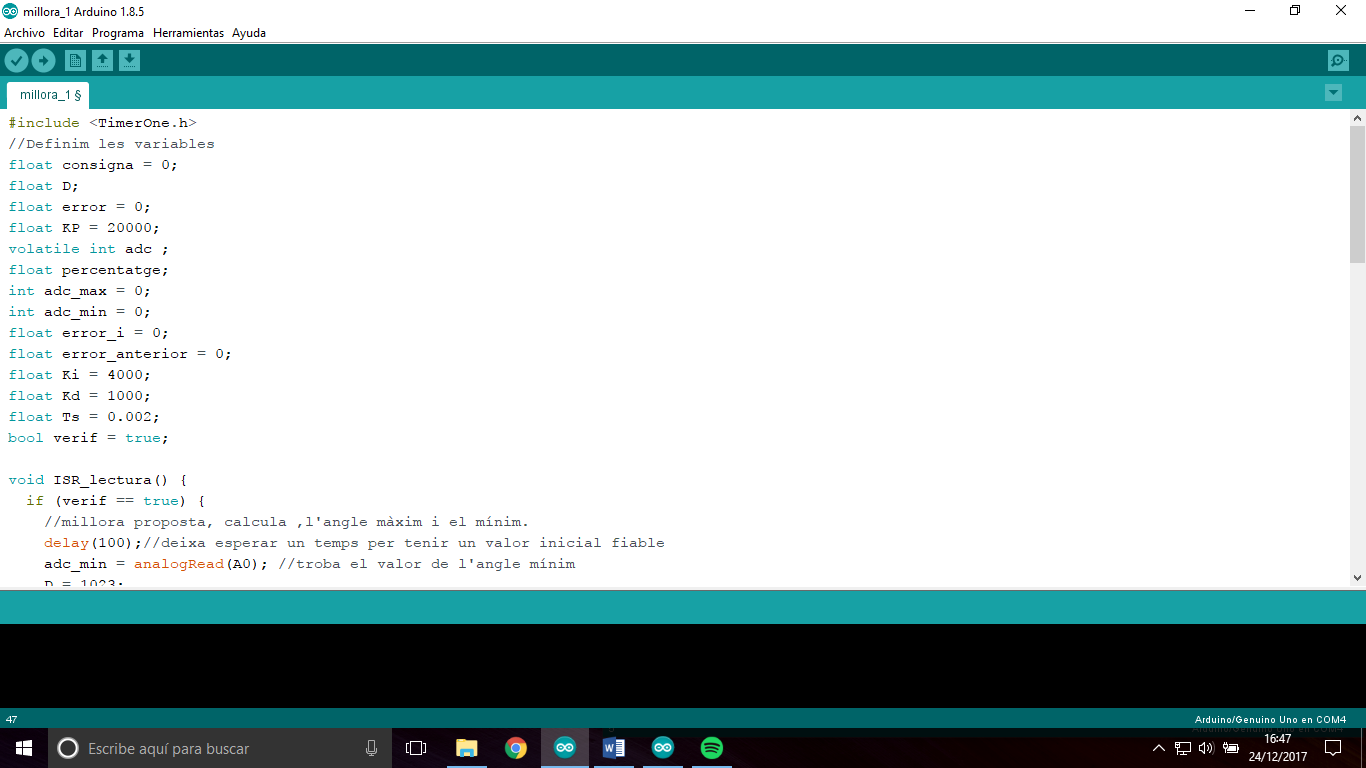
****

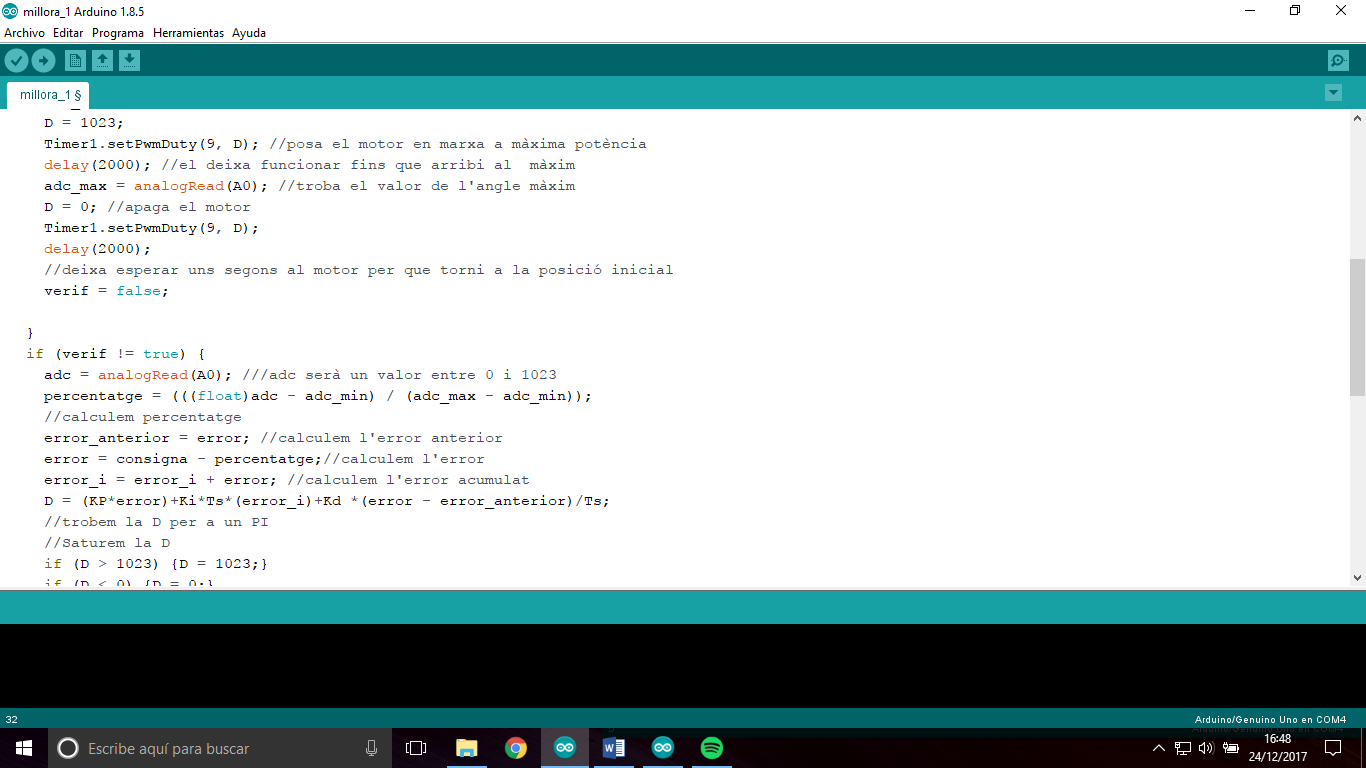
Figura 9:Resposta del sistema amb un valor de consigna del 75% i el controlador PID ajustat del codi anterior.

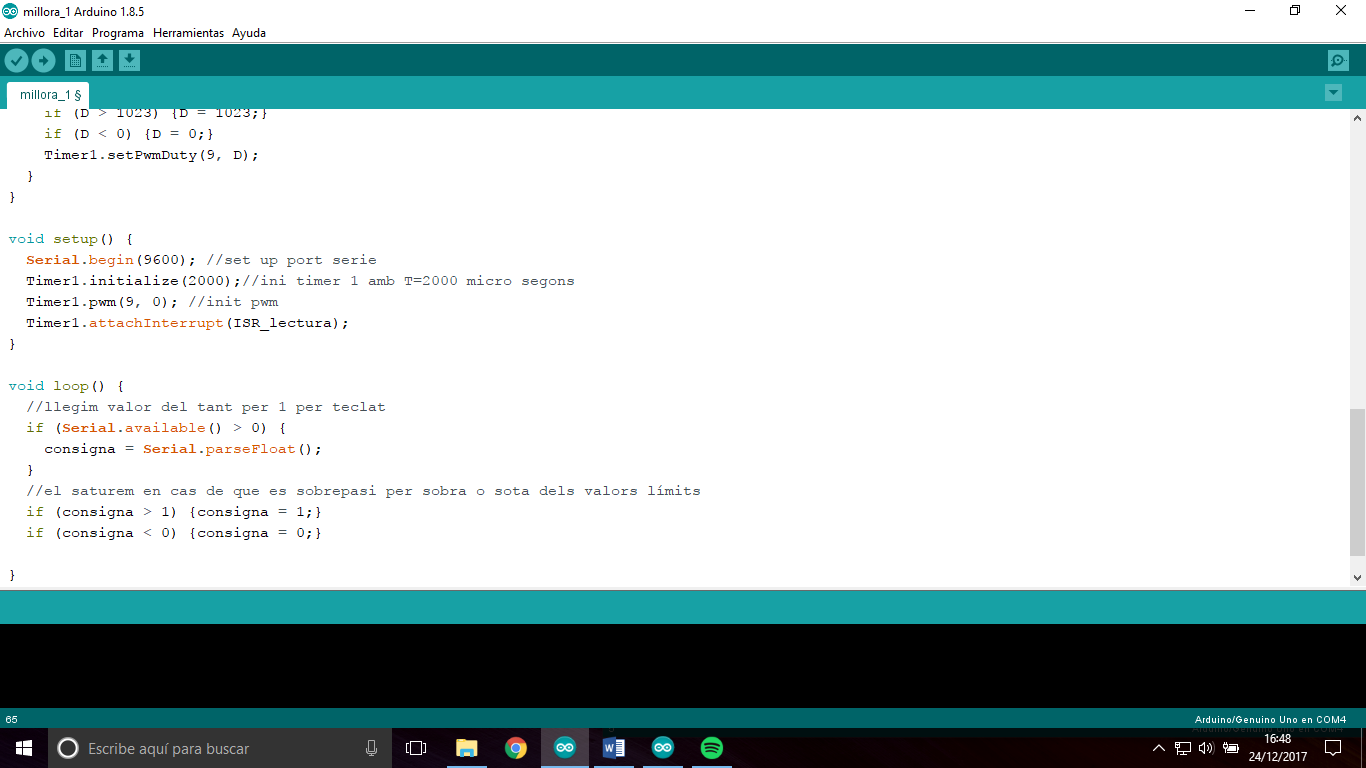
## Millores teòriques proposades

A banda de les tasques proposades hem intentar introduir al sistema dues millores a continuació proposades. Cal destacar que aquestes millores són **únicament teòriques,** ja que no vam disposar de temps suficient al laboratori per provar-les degut a diversos problemes que vam tenir durant la realització del projecte.

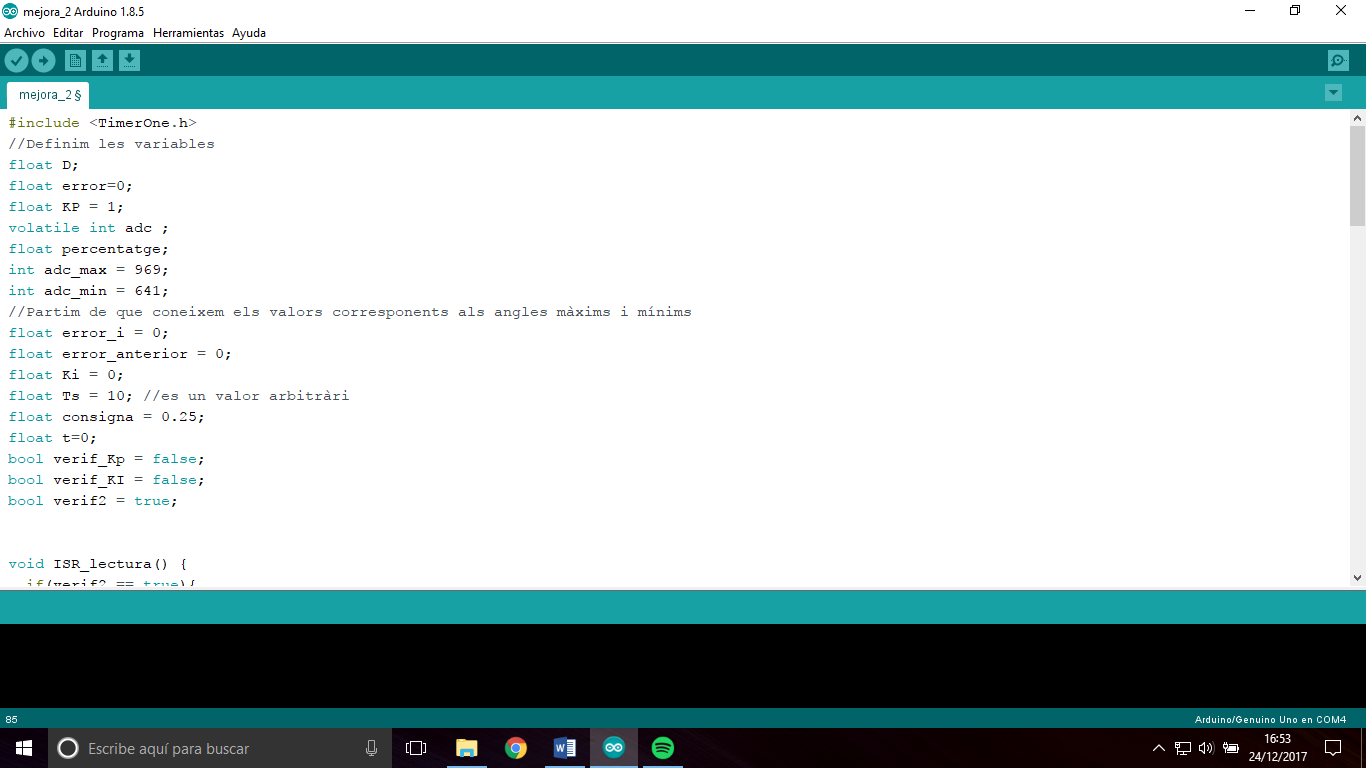
### Millora 1: Lectura automàtica del valors màxim i mínim de la posició

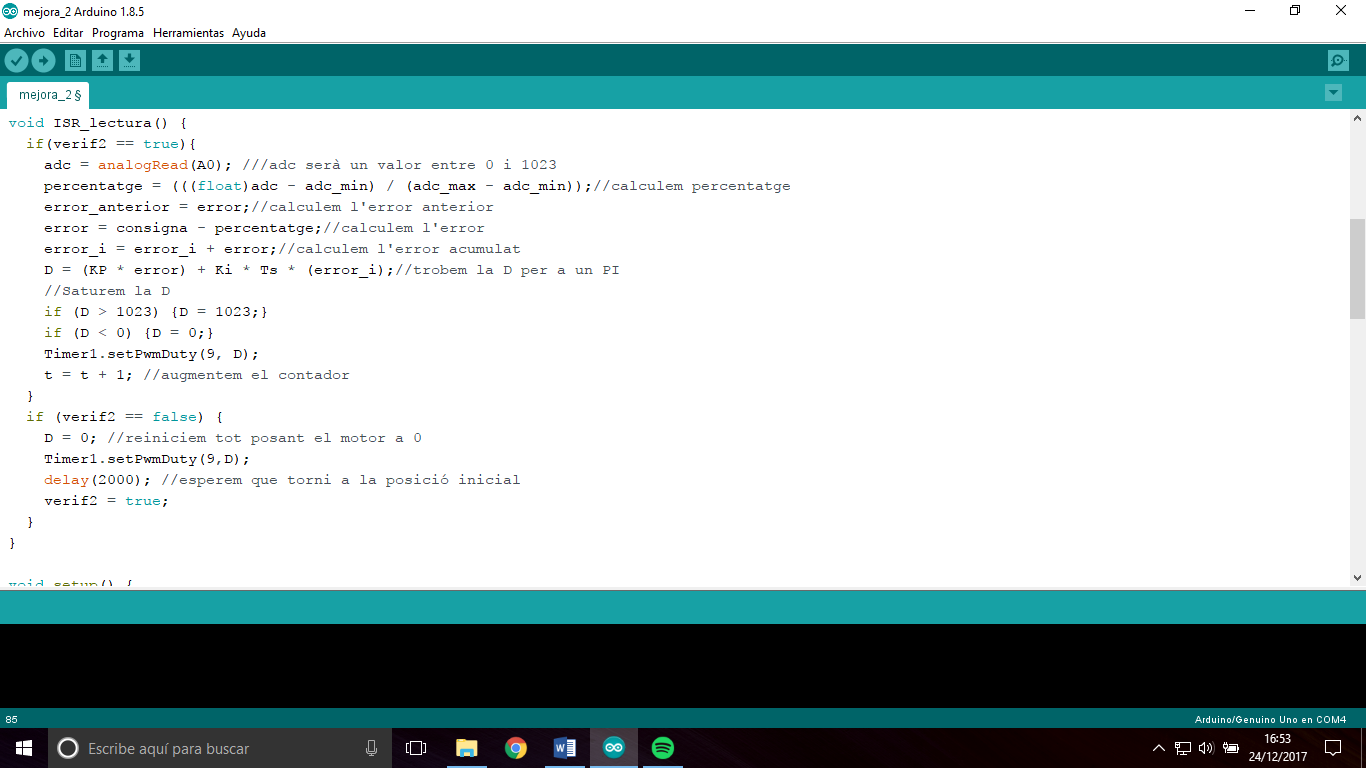


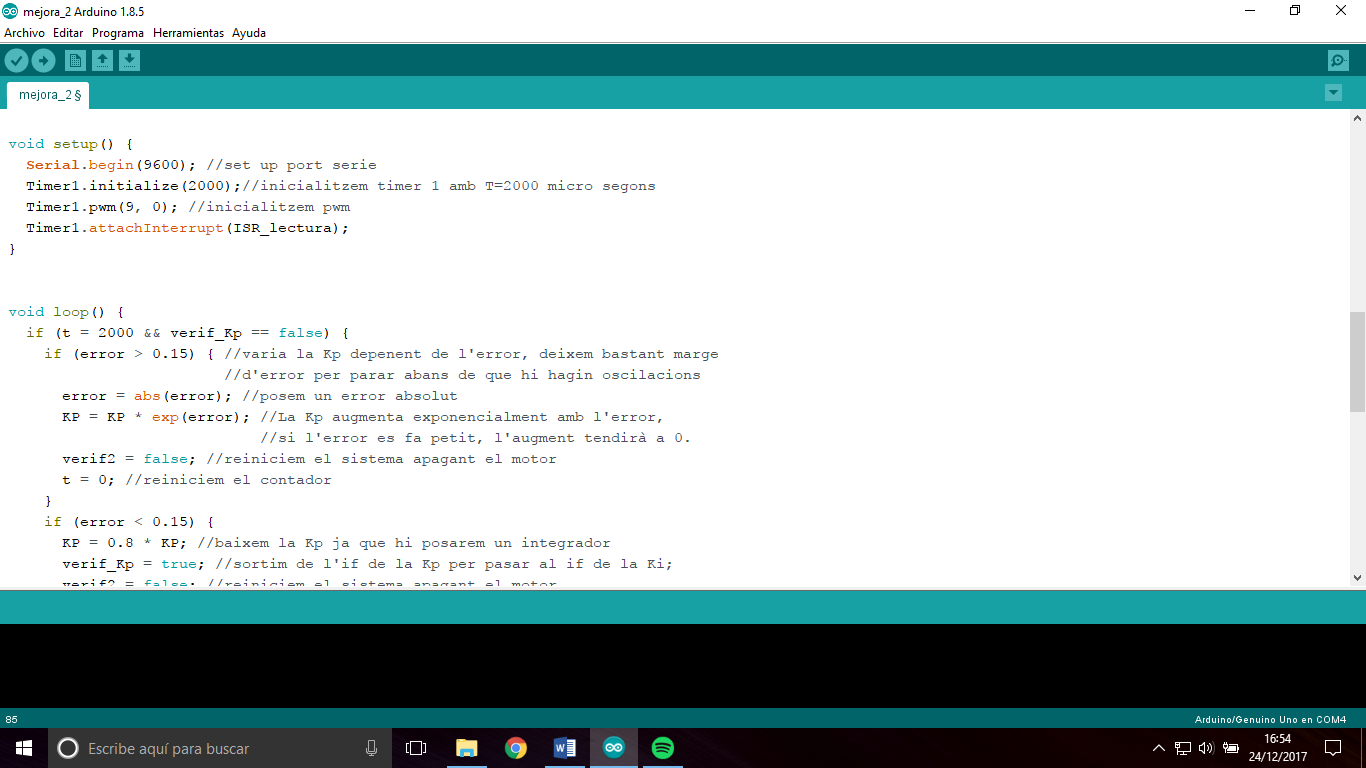


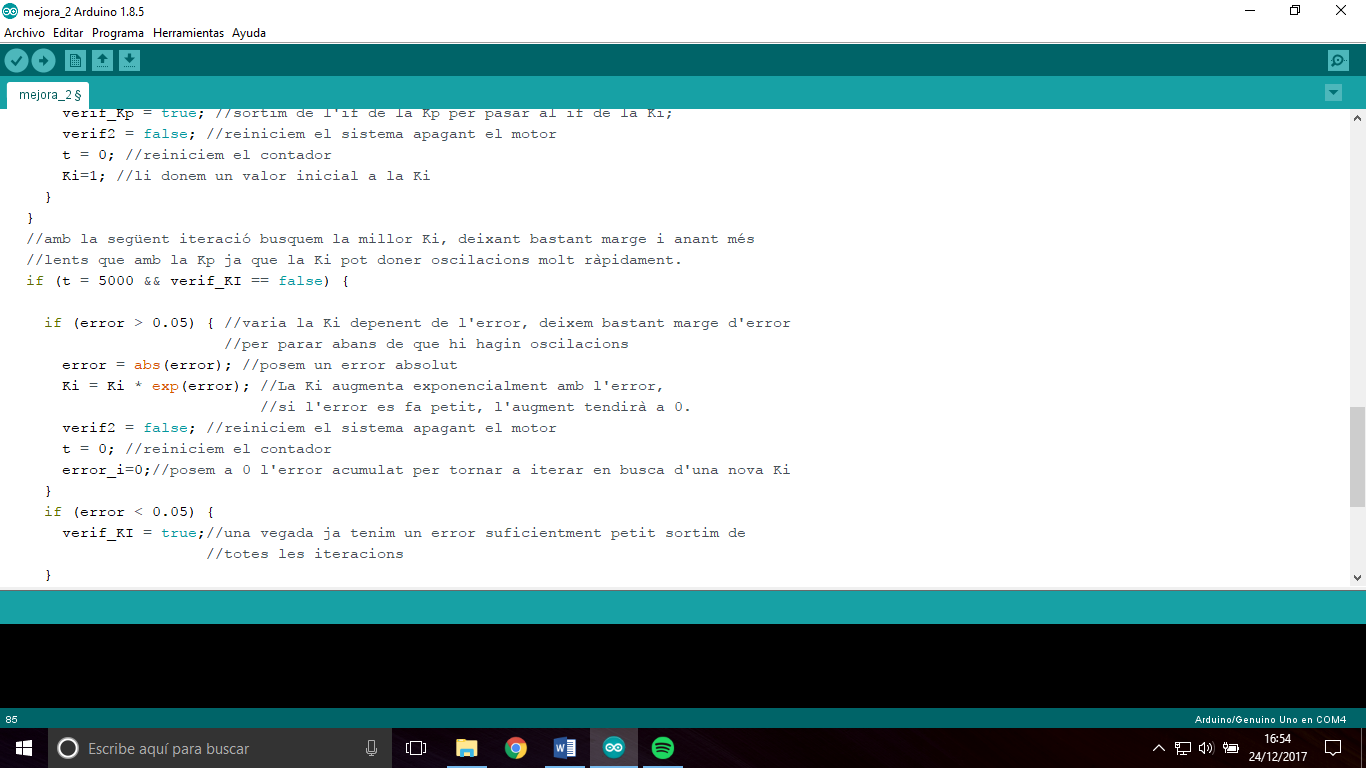


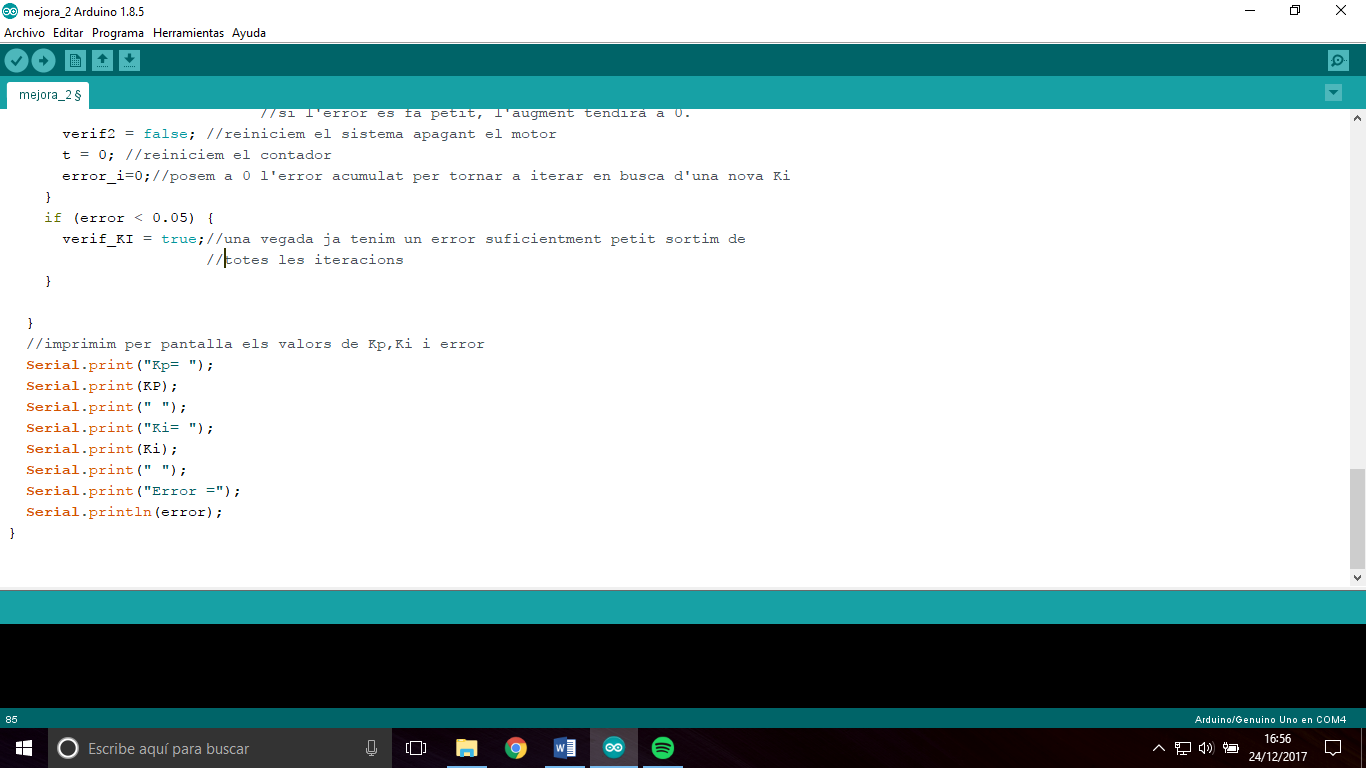
### Millora 2: Càlcul automàtic del controlador PI











# Conclusions i comentaris

Aquesta experiència treballant amb el microcontrolador Arduino ha sigut molt interessant per poder veure com un codi es traduïa en el comportament d’un sistema electromecànic. També ha sigut molt enriquidor poder estudiar els algoritmes de control automàtic d’una manera tan visual com la d’aquesta pràctica.

Com a comentaris cal destacar que la tècnica Zieger-Nichols per assignar constants al sistema PID no funciona correctament i dona uns resultats bastants dolents. Per tant com a metodologia de cara a regular el PID es fa patent la necessitat de fer-ho manualment en aquest cas. Els resultats obtinguts d’aquesta manera han sigut bastant bons.

Com a millores hem afegit una part extra de codi que ens serveix per determinar l’angle mínim i el màxim del sistema de cara a després aplicar el control. Aquest codi primerament pren nota del valor que correspon a l’angle mínim i després posa el motor a màxima potència (esperant una estona a que arribi a l’angle màxim) prenent el valor de l’angle màxim. Cal afegir que aquest codi no el vam probar axí que segurament els temps d’espera no siguin del tot correctes i li falti ajustar-ho una miqueta més.

També hem afegit una altre millora una mica diferent que busca les constants Kp i Ki per a un control PI. Cal esmenar que aquesta millora no s’ha probat i potser no funcionaria bé, pero compila de manera correcta. Aquest codi el que fa es anar iterant per trobar primer la Kp i després la Ki. Per trobar la Kp el que fa es un cicle on per a una determinada Kp i per a un angle objectiu intenta arribar i calcula l’error. Una vegada te l’error multiplica la Kp anterior per la funció exponencial elevada a l’error. La gràcia de multiplicar per la funció exponencial es que aquesta si l’error és molt gran augmentarà molt, però conforme l’error es va fent petit aquesta multiplicació es va reduint fins a arribar a 1. L’objectiu que fixem és que l’error sigui menor de 0.15 de manera que som molt conservadors per tal d’evitar oscilacions.

Aquest procediment es repeteix igual amb la Ki, (amb uns paràmetres diferentes) per tal d’arribar a uns valors de Kp i Ki amb un error raonablement alt. El codi en si es més complex ja que té diverses clausules per sortir, uns paràmetres que redueixen la Kp resultant... però en grans trets aquest és el funcionament del codi. Igual que l’altre, no s’ha pogut probar i hi ha moltes possibilitats de que no funcioni bé, segurament pel temps dels delays que potser no són correctes i s’hauria d’esperar més o menys entre cada cicle. Al no haber pogut depurar el codi ens hem quedat amb el dubte de si haguès funcionat.

Potser enlloc d’utilitzar la funció exponencial haguès sigut millor utilitzar algun tipus d’aproximació lineal com un desenvolupament de Taylor (tot i que l’exponencial d’Arduino en si també és una aproximació lineal a aquesta funció). No vam afegir un control derivatiu ja que avegades té un comportament bastant inestable i al buscar fer-ho de manera totalment automatitzada no sabiem com podriem haber modelitzat algun tipus de clausula que avisés d’un comportament oscilatori. Ens hauria agradat probar altres metodologies de control per a aquest projecte però desgraciadament no vam tenir temps d’implementar cap altre.