Proyecto #2 microprocesadores

“Carro seguidor de línea”

Ricardo Andrés Saavedra B

Iván Darío Zuluaga A

Ingeniero Gerardo Andrés López

Universidad del Quindío

Facultad de ingeniería

Armenia, Quindío

Lunes 18 de abril de 2016

**Avance #1 para la implementación de carro seguidor de línea**

Para el proyecto numero 2 correspondiente a la materia de microprocesadores se empieza a estudiar la forma de la implementación de un carro seguidor de línea, para este proyecto se comienza la implementación la cual consta en dos partes, la parte física del carro (hardware) y la parte de la programación (software), las dos son muy importantes para el correcto funcionamiento del proyecto.

Para este primer avance que muestra el trabajo realizado hasta el momento se documenta específicamente el procedimiento llevado a cabo para dar forma al carro con los elementos que se consiguieron en el mercado con una previa búsqueda de la información para conocer que se debe tener en cuenta y que resultaría más óptimo para la práctica.

Para esto se consultó algo de teoría con lo cual se pudo conocer también el proceso de la programación la cual representará el uso del controlador PID

**Lista de elementos necesarios:**

* Sensores QTR A8
* Procesador ATMEGA 328
* Baby orangután
* Baquela
* Leds
* Pulsador
* Borneras de tornillos
* Switch
* Regleta
* Batería
* Motores
* Llantas de goma

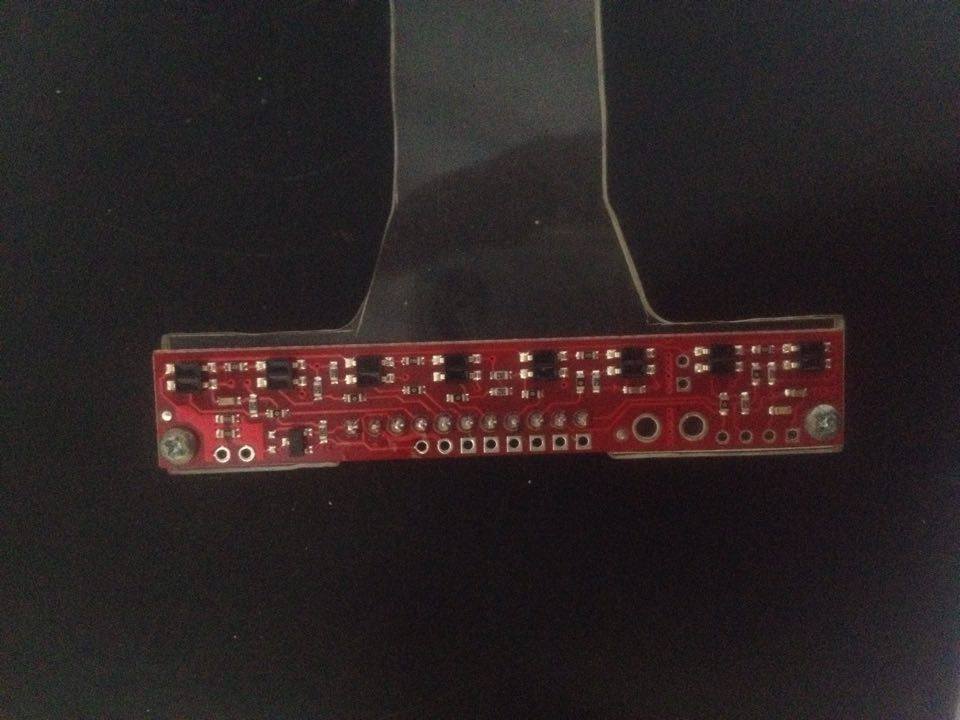
Inicialmente se construye el carro con los anteriores elementos, estos se compraron también pensando en la utilización del algoritmo que se va a realizar (PID).

En primera instancia se buscaron ruedas que tengan características que permitan un mejor movimiento y que no generen mucho rozamiento lo cual impediría que el carro se mueva o no viaje a una mayor velocidad por causa física.

****

**Fig.1** Ruedas de pololu y motores

A continuación se va a mostrar los sensores QTR cuya finalidad es la de percibir la línea la cual debe seguir, este le enviara la información al controlador el cual vendrá siendo el cerebro del carro. Estos sensores funcionan con un principio óptico el cual está determinado por la reflexión de luz, por ende será necesario que la pista cuente con características que le permitan al sensor percibir los cambios



**Fig.2** sensores QTR 8A de pololu

Se construye una base de acrílico la cual representará el esqueleto del carro, esta se construye de tal forma no genere complicaciones con el movimiento y que se adapte de una mejor manera a el seguimiento de una línea que tendrá un grosor no muy grande



**Fig.3** base de acrílico (esqueleto del carro)



**Fig.4** cables conectores para unir el sensor QTR con la tarjeta de control

Los conectores de la figura 4 son aquellos por los cuales existirá la comunicación del exterior con el controlador ATMEGA y así cumplir con la conexión de todo el circuito.

En la siguiente figura se podrá visualizar la tarjeta en la cual ira conectado el procesador, se puede apreciar también algunas resistencias que evitaran una sobrecarga en el circuito, se pueden apreciar unos pines de entrada y salida en donde se harán las conexiones con el sensor y el procesador, unos leds que tendrán como principal funcionalidad la detección de los errores (cambios con referencia al set point)

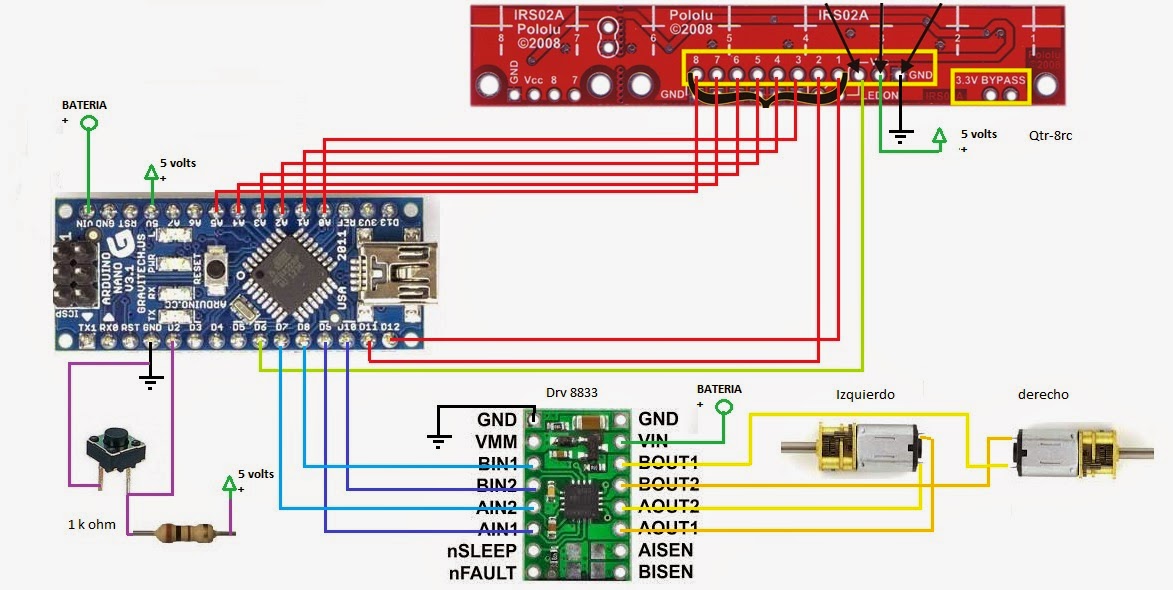


**Fig.5** placa de control



**Fig.6** montaje final del carro seguidor de línea

Todo se construyó teniendo en cuenta el siguiente esquema:

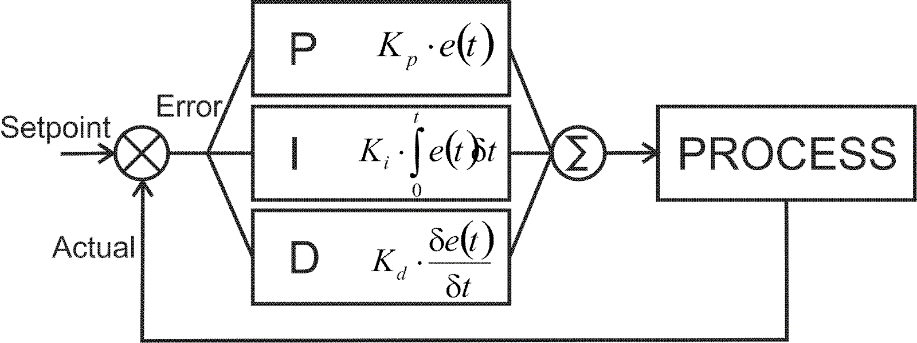
[](http://4.bp.blogspot.com/-_AygRKHF9x4/VR5bisAuGQI/AAAAAAAAAdM/SMe3nobDkOg/s1600/basico%2Bvel.jpg)

**Fig.7** esquema general del carro seguidor de línea

A continuación se muestran algunas características a tener en cuenta para el funcionamiento PID

**PID:**

El PID (control proporcional, integral y derivativo) es un mecanismo de control por realimentación que calcula la desviación o error entre un valor medido y el valor que se quiere obtener (set point, target position  o punto de consigna), para aplicar una acción correctora que ajuste el proceso.

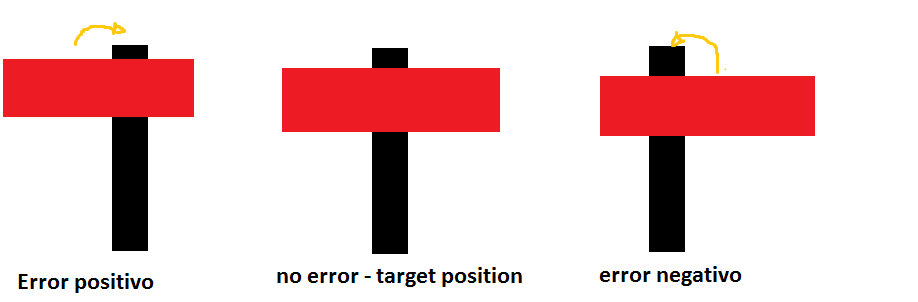
[](http://3.bp.blogspot.com/-6w08_oEbwVE/VIFmmdGrzaI/AAAAAAAAATs/Qt8ig581Fag/s1600/PIDController%2B(1).png)

**fig.8** Esquema del controlador PID

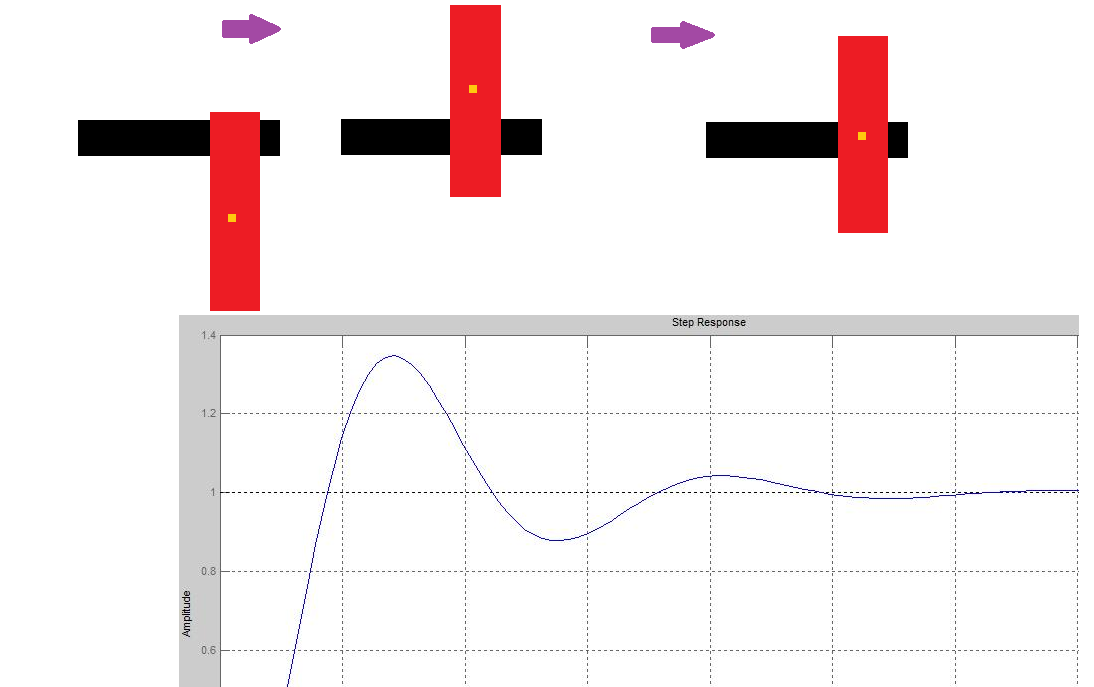
En el caso del robot velocista, el controlador PID, (que es una rutina basada matemáticamente), procesara los datos del sensor, y lo utiliza para controlar la dirección (velocidad de cada motor), para de esta forma  mantenerlo  en curso.

[](http://4.bp.blogspot.com/-mHC2ZRTHbno/VIIIg_B065I/AAAAAAAAAUI/ZO1EgvhnNzc/s1600/ec_pid_basico.png) (1)

**- Error** - Llamamos a la diferencia entre la posición objetivo y la posición medida del error. (Que tan lejos del punto de consigna se encuentra el sensor, en nuestro caso el objetivo es tener  los sensores centrados)

[](http://2.bp.blogspot.com/-Oo43UyAGSzE/VIFdY3M-LII/AAAAAAAAATc/obvtVEPBc4Q/s1600/adferty.png)

**fig.9** Tipos de errores de posición

[](http://1.bp.blogspot.com/-VmoCQGjP-z0/VIFqmzjwAJI/AAAAAAAAAT4/gF4RNJOKolQ/s1600/lkjbu.png)

**fig.10** Representación del sistema PID gráficamente

**-Target Position**- Cuando el error es  0 (cero). En el caso del robot velocista, la idea es siempre mantenerlo en la línea, o lo que es el caso de los sensores,  mantenerlo centrado y así no  se llegue a salir de la línea.

**->PARAMETROS:**

**-Proporcional:** Es la respuesta  al error que se tiene que entregar de manera inmediata, es decir, si nos encontramos en el centro de la línea, los motores, tendrán en respuesta una velocidad de igual  valor, si nos alejamos del centro, uno de los motores reducirá su velocidad y el otro aumentará.

                                    **Proporcional= (posición) -punto\_consigna**

**-Integral:** La integral es la sumatoria de los errores acumulados, tiene como propósito el disminuir y eliminar el error en estado estacionario provocado por el modo proporcional, en otras palabras, si el robot velocista se encuentra mucho tiempo alejado del centro (ocurre muchas veces cuando se encuentra en curvas), la accion integral se ira acumulando e ira disminuyendo el error hasta llegar al punto de consigna,

                                **Integral=Integral + proporcional\_pasado**

-**Derivativo:**Es la derivada del error, su función es mantener el error al mínimo, corrigiéndolo proporcionalmente con la mismo velocidad que se produce, de esta manera evita que el error se incremente, en otra palabra, anticipara la acción evitando así las oscilaciones excesivas.

                    **Derivativo=proporcional-proporcional\_pasado**

**->CONSTANTES:**  
 **Factor (Kp)**- Es un valor constante utilizado para aumentar o reducir el impacto de Proporcional. Si el valor es excesivo, el robot tendera responder inestablemente, oscilando  excesivamente. Si el valor es muy pequeño, el robot respondera muy lentamente,  tendiendo a salirse de las curvas  
  
**Factor (Ki)**- Es un valor constante utilizado para aumentar o reducir el impacto de la Integral, El valor excesivo de este provocara oscilaciones excesivas, Un valor demasiado bajo no causara impacto alguno.  
  
**Factor (Kd)**- Es un valor constante utilizado para aumentar o reducir el impacto de la Derivada. Un valor excesivo provocara  una sobre amortiguación.  Provocando inestabilidad.

**Salida\_pwm = (proporcional \* Kp ) + ( derivativo \* Kd ) + (integral\*Ki);**

**->Sincronizacion PID:**

Aquí viene el reto, la sintonización pid, es aquí donde se tendrá que buscar las constantes que correspondan a las características físicas del robot, la forma más fácil de hacerlo es por ensayo y error, hasta obtener el valor deseado. Aquí hay unos pasos que ayudaran mucho a buscar esas constantes:

1. Comience con Kp, Ki y Kd
2. igualando 0 y trabajar con Kp primero. Pruebe  establecer Kp a un valor de 1 y observar el robot. El objetivo es conseguir que el robot siga la línea, incluso si es muy inestable. Si el robot llega más allá y pierde la línea, reducir el valor de Kp. Si el robot no puede navegar por una vez, o parece lenta, aumente el valor Kp.
3. Una vez que el robot es capaz de seguir un poco la línea, asignar un valor de 1 a Kd .Intente aumentar este valor hasta que vea menos oscilaciones.
4. Una vez que el robot es bastante estable en la línea siguiente, asigne un valor de 0,5 a 1,0 a Ki. Si el valor de Ki es demasiado alta, el robot se sacudirá izquierda y derecha rápidamente. Si es demasiado baja, no se verá ninguna diferencia perceptible. El Integral es acumulativo por lo tanto  el valor Ki tiene un impacto significativo.  puede terminar  ajustando por 0,01 incrementos.
5. Una vez que el robot está siguiendo la línea con una buena precisión, se puede aumentar la velocidad y ver si todavía es capaz de seguir la línea. La velocidad afecta el controlador PID y requerirá re sintonizar como los cambios de velocidad.