

GUIA PARA EL DOCUMENTO DE DISEÑO DETALLADO  
TALLER DE SISTEMAS ELECTRONICAS  
FACULTAD DE INGENIERIA ELECTRONICA Y TELECOMUNICACIONES  
UNIVERSIDAD DEL CAUCA



PRESENTADO POR:  
JESÚS HUMBERTO DORADO DE LA CRUZ  
SERGIO ANDRES MUÑOZ HOYOS

PRESENTADO A:  
ING. VLADIMIR TRUJILLO  
ING. VICTOR MONCAYO

INGENIERIA EN AUTOMATICA INDUSTRIAL  
V SEMESTRE

POPAYAN, CAUCA  
3 DE DICIEMBRE 2014

## **GUIA PARA EL DOCUMENTO DE DISEÑO DETALLADO.**

### *I. INTRODUCCION.*

El brazo Robótico Ax-18 es manipulado a través del software Robo Plus y el controlador CM 700 de Robotis, en este proyecto se diseña un controlador con Arduino mega 2560 con interactor de mando un Joystick Genius F22, para lograr reemplazar su controlador por defecto. Los Smart Actuadores Dynamixel poseen características superiores a sus homólogos, cada motor cuenta con un micro controlador que se encarga de monitorear la posición, temperatura, velocidad y torque además de establecer alarmas en caso de cruzar algún límite parametrizado. Los motores se comunican serialmente cada uno responderá a un identificador (ID) y pueden crearse redes hasta 254 dispositivos, en un diseño completamente modular.

### *II. SELECCIÓN DEL CIRCUITO Y EXPLICACION DE FUNCIONAMIENTO.*

El circuito que se utiliza para el control de transmisión de datos es el integrado 74LS241N. El interactor humano – brazo se hace por medio del joystick F22 de Genius el cual se le adapta para el óptimo funcionamiento deseado para el controlador.

El brazo robótico AX-18 es alimentado por una fuente externa cuyas características necesarias son:

Voltaje: 9 Voltios dc → máx: 12 voltios

Corriente: 250 miliamperios → máx: 1 amperio dc

Es importante resaltar que tanto la fuente externa como la alimentación de la tarjeta arduino y demás componentes extras que se pudiesen utilizar deben compartir una tierra en común esto se debe cumplir para evitar la pérdida de datos.

También tener en cuenta que la resistencia limitadora de corriente de los datos debe ser proporcional al número de motores conectados en serie margarita, para el brazo robótico AX-18 que manejan 7 servomotores la resistencia a utilizar es de 1 K $\Omega$ .

### *III. CALCULO DE COMPONENTES.*

Con las ecuaciones que resumen el comportamiento de la aproximación circuital seleccionada, especificaciones técnicas de fabricantes de dispositivos, algoritmos SW depurados para una tarea específica y los CRITERIOS DE DISEÑO debidamente SUSTENTADOS, se debe realizar la selección de los componentes para la aplicación particular, sean software o hardware

### *IV. JUSTIFICACION DE LOS ELEMENTOS.*

- Se selecciona un joystick genérico Genius F22A de palanca de mando que en su estructura interna está compuesta por potenciómetros exponenciales, pulsadores y cableado secuencial. Esto es una muy buena herramienta debido a que cada

articulación es activada por un pulsador y el movimiento de la misma por el eje compuesto por 2 potenciómetros exponenciales a los cuales se les estudio su comportamiento.

- Se trabaja en la tarjeta Arduino debido a su fácil manejo y lenguaje de programación en C, además se investiga e adapta la librería Dynamixel Serial1 para arduino lo que nos representa avances a grandes pasos en el manejo del software hacia el hardware.
- Se selecciona el circuito 3 estados 74LS241N debido a su fácil manejo e interpretación de lógica.
- Se trabaja con la fuente externa debido a que abarca las necesidades de potencia que requiere el brazo y es la fuente con la que ha trabajado el brazo.

#### V. PLAN ELECTRICO PIN A PIN.

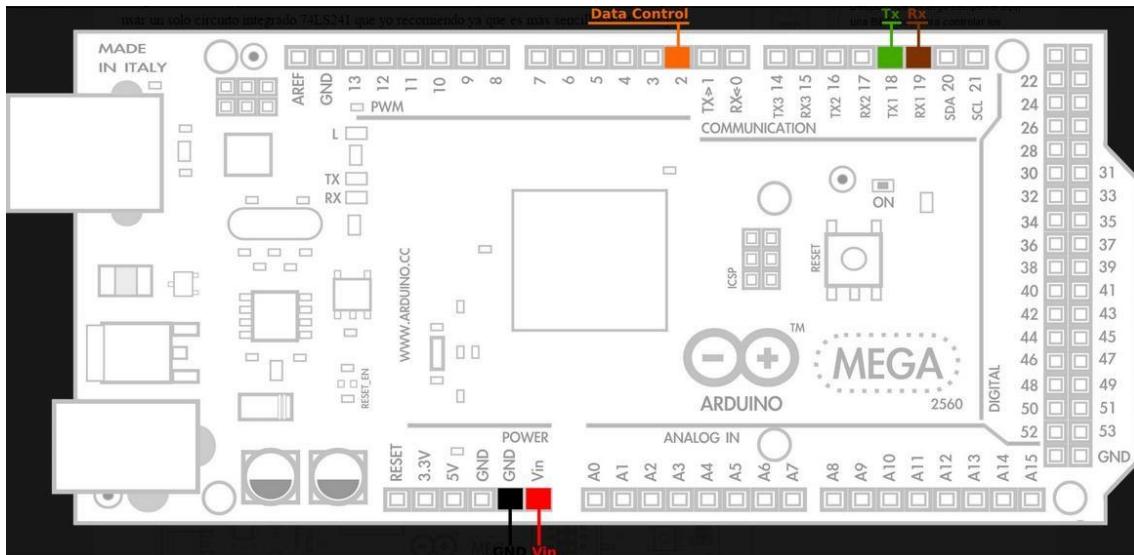


Figura 1 Conexión Arduino

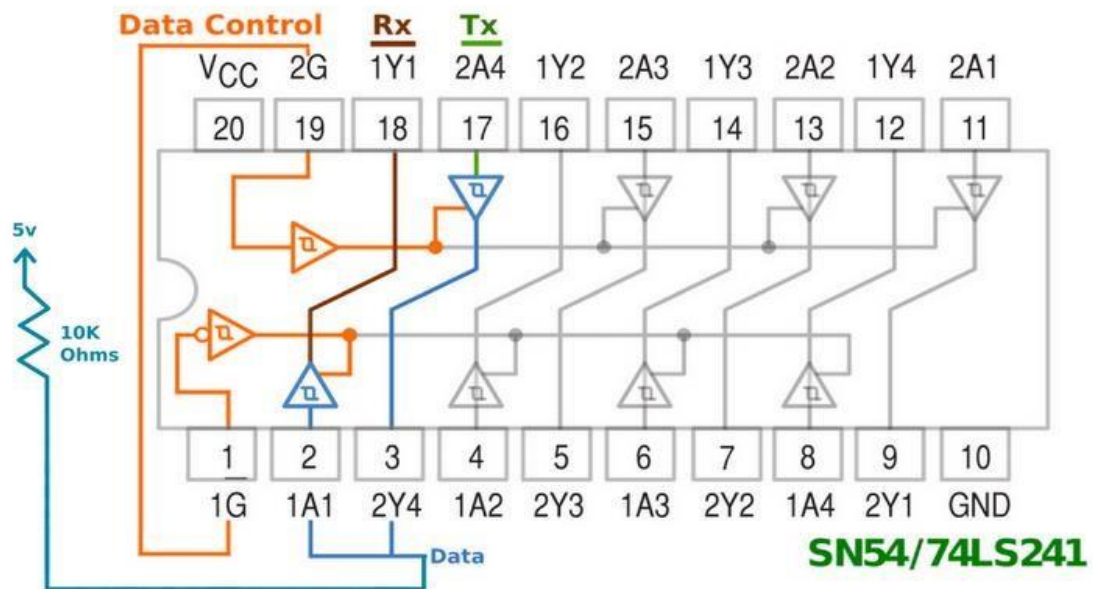


Figura 2 Conexión circuito integrado 74LS241N

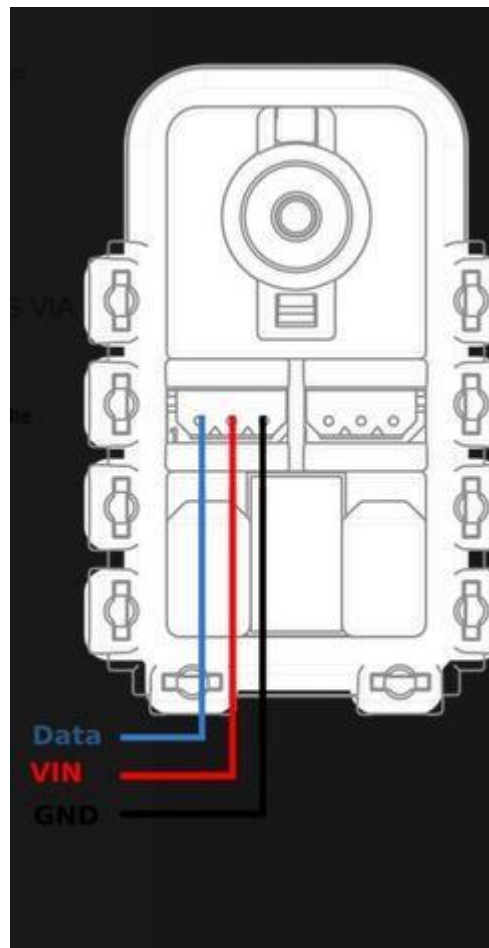


Figura 3. Conexión motor dynamixel AX-18<sup>a</sup>

En la (figura 1) vemos el circuito a utilizar donde la señal manipulada que emite la tarjeta arduino RX, TX y los datos del PIN2 llegan al circuito integrado (3 estados) 74LS241N (figura 2)

cuya función es unificar y posteriormente brindar una señal de salida (datos) bidimensional correspondiente a la conexión y protocolo que manejan los servomotores dynamixel AX-18.(figura 3).

## Codigo:

```
#include <DynamixelSerial1.h> // Libreria del manejo de servos
```

```
void setup(){  
  Serial.begin(9600); // inicia comunicación serial de arduino Dynamixel.begin(1000000);//  
  velocidad de transmision serial en bps Dynamixel.begin(1000000, 2);//velocidad y control de  
  transmision y recepción de datos PIN2
```

Una vez establecido el protocolo de comunicación y velocidad de transmisión de datos se procede activar las características rudimentarias del brazo.

```
Dynamixel.torqueStatus(1, ON); //Activa o desactiva el torque de servomotor  
Dynamixel.setMaxTorque(1,1023); //configura el torque maximo el el servomotor
```

Ahora se verifica el estado de respuesta de los motores para verificar su funcionamiento y comunicación efectiva, esto lo denominamos un feedback que es como una realimentación de estados.

```
// Articulación 1:  
if((bot1==1)&&(bot2==0)&&(bot3==0)&&(bot4==1)){  
  unsigned int p1,p2,pf1,pf2;  
  p1 = map(pot1,380,1023,640,0); // se mapea mediante tal función el valor del potenciómetro  
  leído a uno que nos permita la maxima inclinación en el joystick.  
  p2 = map(pot2,380,1023,640,0);
```

```
  if(p1<570&&(p2>100)&&(p2<550)) // debido a que los potenciómetros son exponenciales se  
  hacen calculos que proporcionen una magnitud obtenida en tiempo real y se junta en una sola  
  variable
```

```
    pf1 = p1+(p2/10);  
    if((p1<570)&&(p2>550))  
      pf1 = 55 + p1;  
    if(p1>570&&(p2>100)&&(p2<750))  
      pf1 = p1+(50-(p2/10));  
    if((p1<650)&&(p1>200)&&(p2<150))
```

```

    pf1 = 780;
    if((p1<200)&&(p2<100)) // estos son los estados maximos que el joystick permite analogo al
    giro de la primera articulacion del brazo
        pf1 = 0;
        pf1=map(pf1,0,780,0,1023);
        Dynamixel.moveSpeed(1,pf1,80);} // instrucción que nos permite mover el motor (1),
        posición(pf1) y velocidad(80).

```

Debido a que los potenciómetros son exponenciales se comporta diferente y por lo cual se hacen diferentes condicionales en los que se establece una serie de funciones que logran describir el comportamiento del eje de joystick con una analogía muy próxima al primer motor correspondiente a la articulación 1.

```

// Articulacion 2
if((bot1==0)&&(bot2==0)&&(bot3==0)&&(bot4==1)){ // estado de los pulsadores
    unsigned int p3,p4,pf3,pf4;
    p3 = map(pot2,380,1023,878,92); // se lleva el valor de potenciómetro del eje x a un valor
    maximo especificado por la estructura del brazo( grados de inclinación)
    pf3 = constrain(p3,92,878); // se contrae dicho valor al permitido para asegurar que no se
    excede el valor del angulo.
    p4 = map(pot2,380,1023,930,146);
    pf4 = constrain(p4,146,930);
    Dynamixel.moveSpeed(2, pf3 ,80); // se mueve el motor 2 de forma inversa al 3 debido a la
    ubicación de cada motor que compone esta articulación
    Dynamixel.moveSpeed(3, 1023-pf4 ,80);}

```

Como los motores 2 y 3 estan de forma inversa el máximo del motor 2 es el minimo de 3 y asi para cada proporción de angulo que requiera la articulación e igual velocidad.

```

// Articulacion 3
if((bot1==0)&&(bot2==0)&&(bot3==1)&&(bot4==0)){
    unsigned int p5,p6,pf5,pf6;
    p5 = map(pot2,380,1023,900,92);
    pf5 = constrain(p5,92,900);
    p6 = map(pot2,380,1023,932,124);
    pf6 = constrain(p6,124,932);

    Dynamixel.moveSpeed(4, pf5 ,80); // al igual que la anterior articulación pero diferentes
    motores y maximos angulos de inclinación

```

```
Dynamixel.moveSpeed(5, 1023 - pf6 ,80);}
```

También funciona en el eje x solo que con diferentes motores y ángulos de posición máximos.

```
// Articulacion 4
if((bot1==0)&&(bot2==1)&&(bot3==0)&&(bot4==0)){
    unsigned int p7,pf7;
    p7= map(pot1,255,1023,1023,0); // esta articulación no tiene restriccion de angulo por lo que
    solo se mapea al valor maximo de pot y se podria activar giro continuo
    pf7 = constrain(p7,0,1023);
    Dynamixel.moveSpeed(6,pf7,100); }
```

En la 4 articulación se mueve con el potenciómetro del eje x y no tiene restricción de ángulo por lo que podría ser de giro continuo.

```
// Articulacion 5 ( Pinzas )
if((bot1==1)&&(bot2==0)&&(bot3==0)&&(bot4==0)){
    unsigned int p8,pf8;
    p8= map(pot2,380,1023,750,500); // en esta articulación la pinza solo puede cerrar y
    presionar hasta cierta proporción por lo que se lleva la señal recibida a una deseada
    pf8 = constrain(p8,500,750);
    Dynamixel.moveSpeed(7,pf8,80);}
}
```

En la ultima articulación la pinza mediante cálculos se determina que puede cerrar a un minimo y abrir a un máximo para que la pinza no esfuerce demasiado y se trave el motor con los piñones de la pinza.

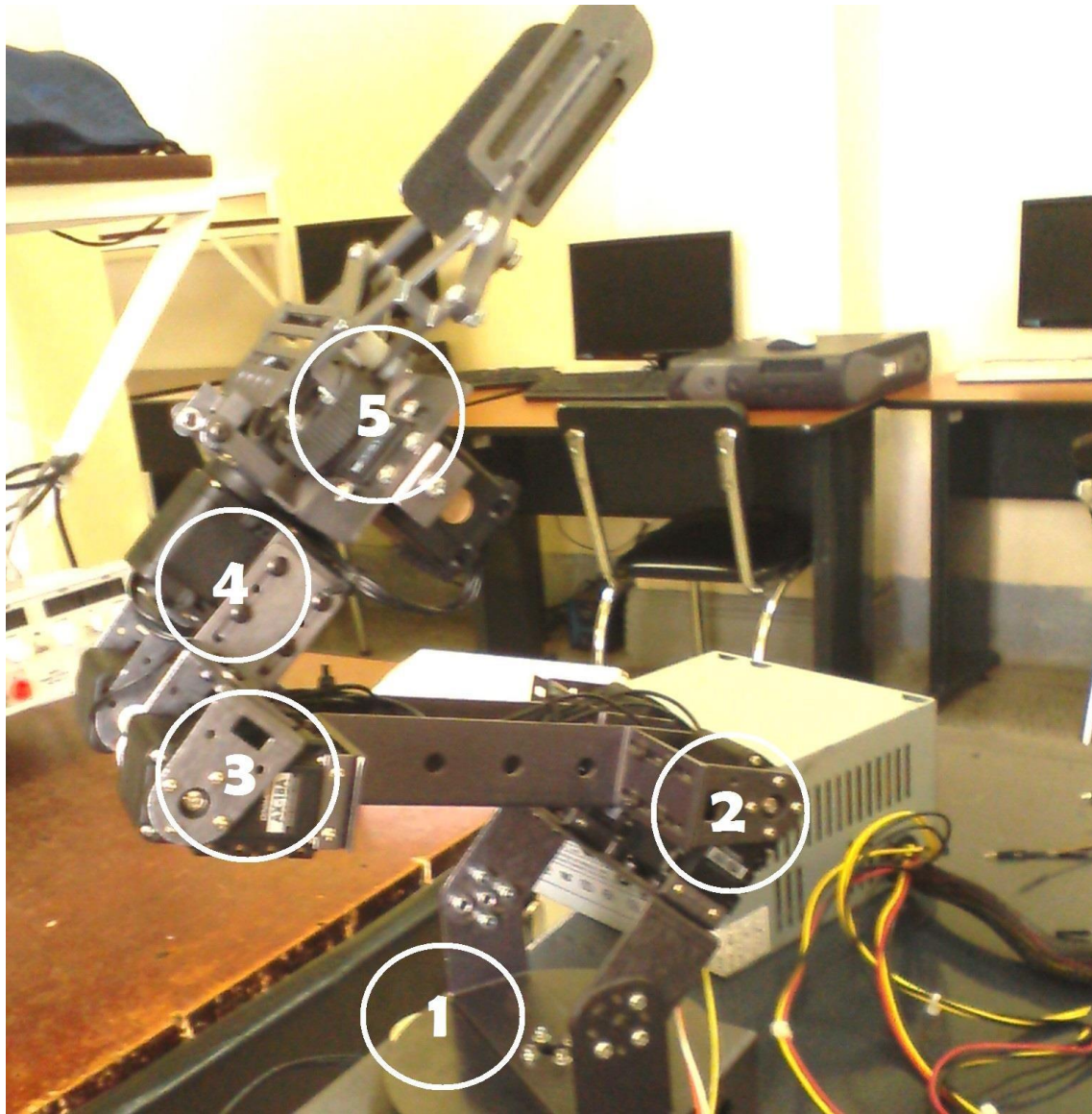
## VI. PRUEBAS Y RESPUESTAS ESPERADAS.

### ▪ Objetivo de la Prueba.

Realizadas las conexiones se produjo a escribir el código por partes, probando solamente así las articulaciones por individual. Primero ensayando la base con giros constantes, teniendo en cuenta que se debe aislar la conexión entre la articulación 1 y 2 del brazo, también pruebas de posición por línea de código y por hardware.

De esta primera prueba se conectó nuevamente todas las articulaciones para probar lo que es la número 2. En este ensayo no se tuvo éxito como se esperaba ya que los motores no recibían la señal necesaria para poder funcionar, entonces se realizó más prueba de código solamente con la primera y segunda articulación.

En la siguiente imagen se observan el número de articulaciones.



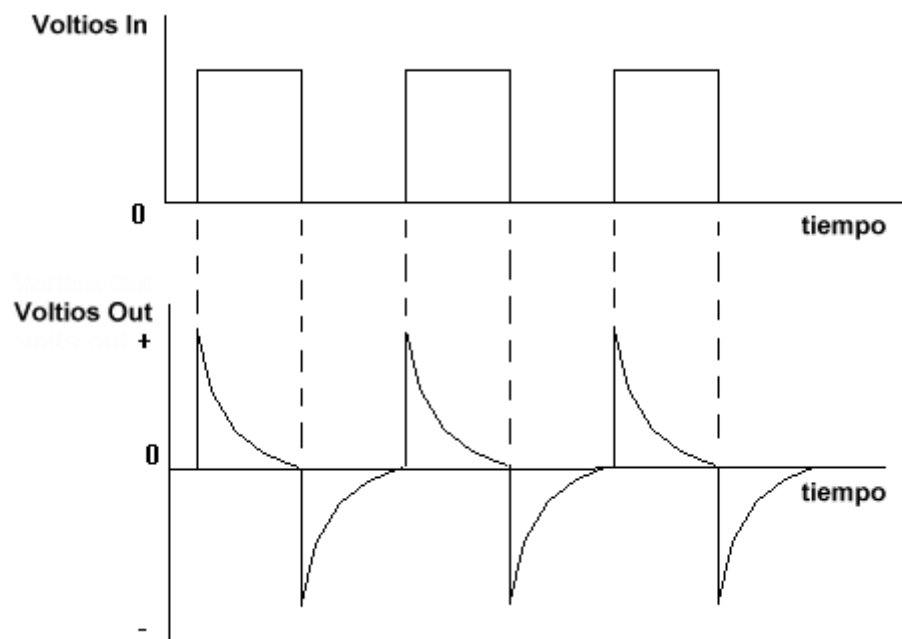
*Brazo Robótico con Dynamixel AX 18.*

Para continuar con las demás articulaciones se realizaron pruebas conectando la placa base desde el motor 4 (articulación 3) teniendo en cuenta que son 7 motores o actuadores Dynamixel AX 18a, mientras se encontraba una solución al problema de señal para todo el brazo en conjunto.

Con muchas pruebas y una de ellas verificar la señal de datos para comprobar si estaba o no débil.

Para ello conectamos un osciloscopio a la señal de datos y se observó algo muy similar a la imagen, entonces se decidió cambiar el valor de la resistencia de 10k conectada en el envío de datos del tres estado (circuito integrado 74LS241) por una de 330  $\Omega$ , con esto se dio solución y se pudo probar todo el brazo en conjunto.



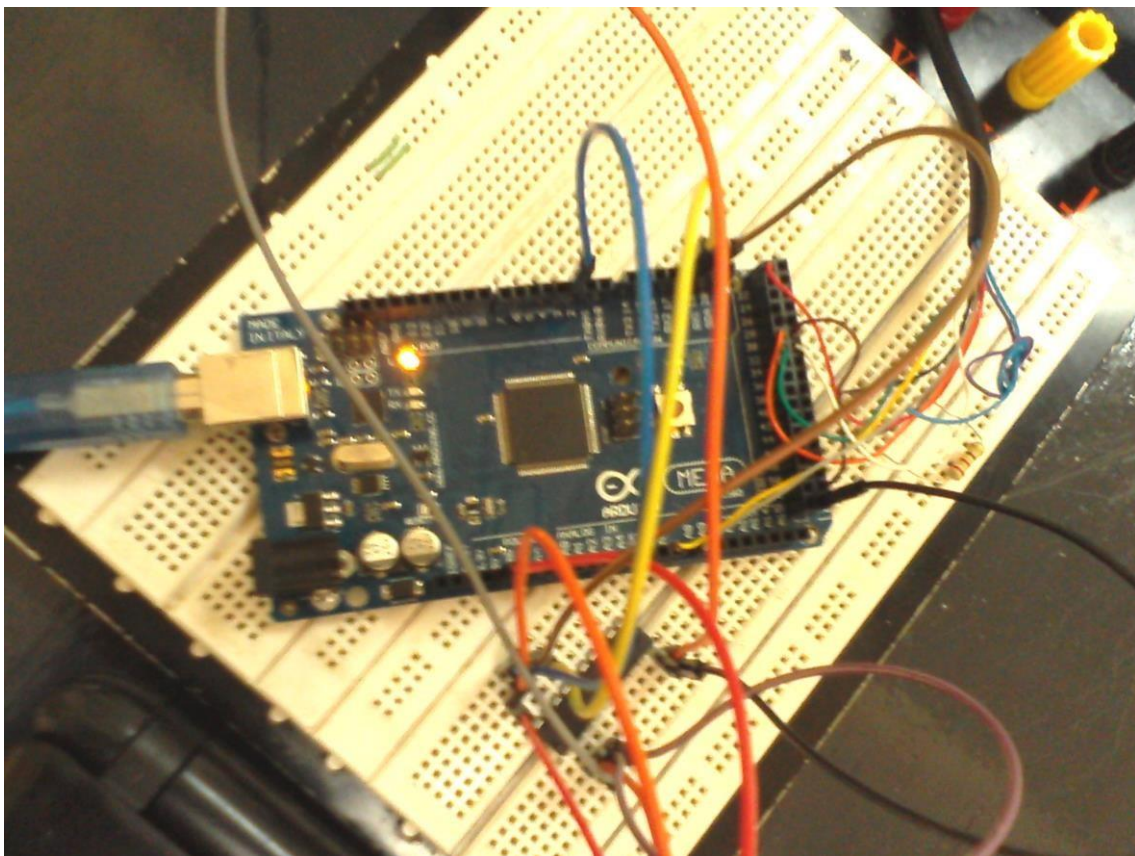


*Señal de pulsos a la salida del circuito 74LS241.*

## VII. APLICACIONES.

Evidenciar la utilidad del sistema diseñado y de sus módulos particulares.

1. ARDUINO MEGA R3 2560.

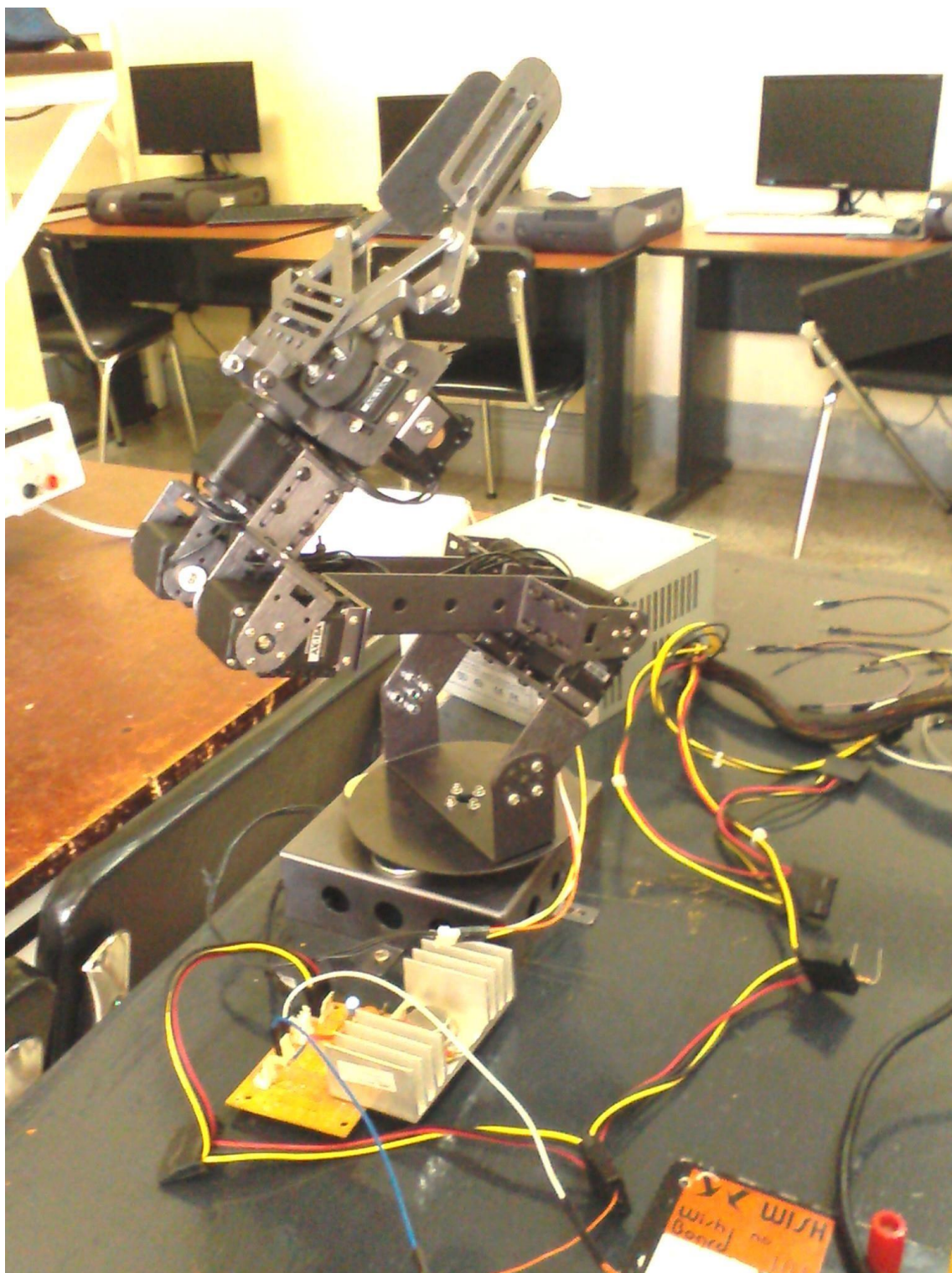


## 2. JOYSTICK GENIUS

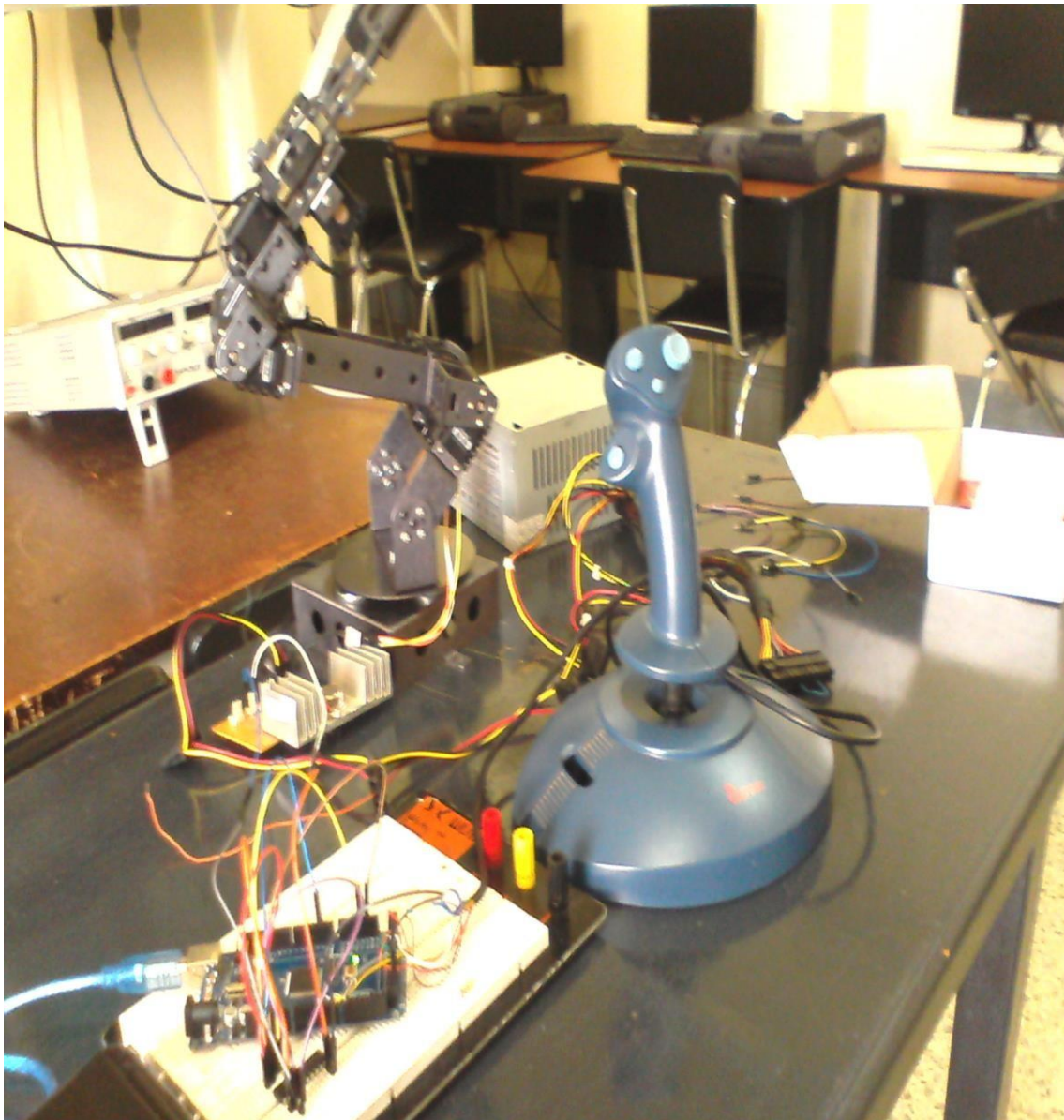




### 3. BRAZO ROBOTICO CON DYNAMIXEL AX 18º



#### 4. TODO EN CONJUNTO.



### *VIII. LISTADO DE ELEMENTOS.*

- 1) Brazo robótico compuesto de 7 motores Dynamixel AX 18A
- 2) Protoboard
- 3) Circuito integrado 74LS241N
- 4) Resistencia de  $330\Omega$
- 5) Arduino mega R3 2560
- 6) Fuente regulada de 9v – 12v, 3A
- 7) Joystick marca Genius
- 8) Cables macho-macho y hembra-macho
- 9) Un computador

### *IX. BIBLIOGRAFIA.*

<http://savageelectronics.blogspot.com/2011/01/arduino-y-dynamixel-ax-12.html>

<http://forum.arduino.cc/index.php?topic=22672.0>

<https://www.youtube.com/watch?v=yEpoMIM0fBI>

<https://www.youtube.com/watch?v=u8j-r8dsIP4>