

Proyecto Primer Parcial

Introducción

El proyecto a detallar a continuación es un prototipo de **"Shadecraft Sunflower"** o **"Sombrilla automática con sensores de luz"**, cabe mencionar es un prototipo a escala, así como su elaboración es una simulación de un proyecto, en el cual la sombrilla efectúa un movimiento que va acorde a la luz de incidencia en esta, con el objetivo de exponer ideas innovadoras en avances tecnológicos; Por otro lado es un sistema de control a lazo cerrado donde la señal emitida es proporcionada por fotoresistores, los cuales se comunican en un algoritmo que se manda al arduino y a servomotores generando un movimiento entre 0 y 180 grados en eje "Z" y "Y".

También los sensores cuentan con una calibración automática con el propósito que se ejecute sin problema con cualquier tipo de luz o en cualquier lugar, con esta función si los sensores reciben una luz proporcionada por el usuario más directa, no se afectan por la luz del medio externa; igualmente cuenta con un push button el cual bloquea la recepción de luz mientras se encuentra presionado esto en simulación de un switch on/off.

Los objetivos principales a lograr durante el proyecto es la implementación de sensores integrados en código de arduino en conjunto de un circuito para un proceso de control. Para la ejecución del proyecto se emplean los conocimientos adquiridos en prácticas previas así como en cursos como sistemas digitales/automatismos lógicos, por lo que el proyecto en mención es la interacción e integración de práctica, conocimiento y experiencia.

Marco Teórico

Un fotorresistor está formado por un semiconductor; su funcionamiento ocurre al incidir la luz sobre él algunos de los fotones son absorbidos, provocando que electrones pasen a la banda de conducción, ocasionando la disminución de la resistencia del componente. Por otro lado, la variación de la resistencia es relativamente lenta, por lo que hace difícil la posibilidad de registrar variaciones rápidas, como las producidas en

fuentes de luz artificiales alimentadas por corriente alterna, igualmente esto es favorable ya que el sensor posee una gran estabilidad.

También, los fotoresistores no resultan adecuados para proporcionar una medición de la iluminancia, es decir, para servir como luxómetro, esto es debido a su baja precisión, así como su fuerte dependencia con la temperatura y especialmente por su distribución espectral ya que no resulta adecuada para la medición de iluminancia, esto se muestra de forma más clara en la Imagen 1.

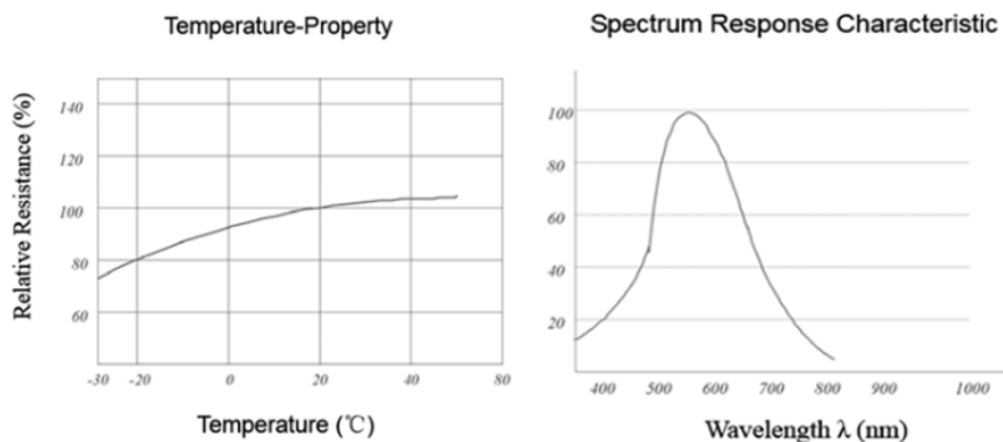


Imagen 1 .-Gráficos del comportamiento de fotoresistencias

Los sensores electrónicos han ayudado no solo a medir con mayor exactitud las magnitudes, sino a poder operar con dichas medidas. Pero no se puede hablar de los sensores sin sus acondicionadores de señal, ya normalmente los sensores ofrecen una variación de señal muy pequeña y es muy importante equilibrar las características del sensor con las del circuito que le permite medir, acondicionar, procesar y actuar con dichas medidas, por ello es necesario la calibración previa para ser precisos.

Es necesario hacer hincapié en conceptos básicos/definiciones para una mejor comprensión de la práctica tal es el caso del arduino, es una plataforma de prototipos electrónica de código abierto, basada en hardware y software la cual es flexible y fáciles de usar por su estructura así como su código.


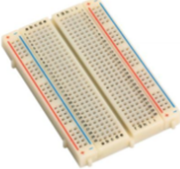




Material	Descripción	Imagen
Servomotor	Es un tipo especial de motor que permite controlar la posición del eje en un momento dado, está diseñado para moverse determinada cantidad de grados y luego mantenerse fijo en una posición; hace referencia a un sistema compuesto por componentes electromecánicos y electrónicos, ya que cuenta con un juego de engranes y una tarjeta de control, todo dentro de una carcasa de plástico. (PH, 2017)	
Protoboard	Es un tablero con orificios, en la cual se pueden insertar componentes electrónicos y cables para armar circuitos. Se divide en tres secciones en canal central: es la región localizada en el medio, se utiliza para colocar los circuitos integrados; buses: se localizan en ambos extremos, se representan por las líneas rojas (o de voltaje) y azules (o de tierra) y conducen de acuerdo a estas, no existe conexión física entre ellas; pistas: se localizan en la parte central, se representan y conducen según las líneas rosas. (CE, 2018)	
Fotoresistor	También se le conoce como LDR (light-dependent resistor), es un dispositivo cuya resistencia varía en función de la luz recibida; se usa esta variación para medir, a través de las entradas analógicas, una estimación del nivel del luz, funciona disminuyendo su resistencia a medida que aumenta la luz sobre él. (CDefinicion,	
Resistencias	Es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado; frenando el flujo de circulación en las cargas eléctricas o de electrones. En un consumidor conectado a un circuito eléctrico representa en sí una carga, resistencia u obstáculo para la circulación de la corriente eléctrica. (AF,2018)	
Multímetro	Es un instrumento de medida, que puede medir distintos parámetros eléctricos y magnitudes en el mismo aparato. Las mas comunes son las de voltímetro, amperímetro y óhmetro.(FT,2018)	
Arduino (UNO)	Es una plataforma de hardware libre, consiste en una placa con un microcontrolador, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios; se usa para desarrollar elementos autónomos, conectándose a dispositivos e interactuar tanto con el hardware como con el software. Existen diversos tipos y varían entre sí, esta el nano,uno ,etc.(Arduino, 2018)	

Tabla 1.- Lista de materiales

Funciones de programación utilizadas

Función	Descripción
void setup()	El setup es la primera función en ejecutarse dentro de un programa en Arduino. Es, básicamente, donde se "setean" las funciones que llevará a cabo el microcontrolador.

<code>void loop()</code>	Loop en inglés significa lazo o bucle. La función loop en Arduino es la que se ejecuta un número infinito de veces. Al encenderse el Arduino se ejecuta el código del setup y luego se entra al loop, el cual se repite de forma indefinida hasta que se apague o se reinicie el microcontrolador.
<code><Servo.h></code>	Es la librería para poder hacer uso de todas las opciones y funciones para poder hacer funcionar un servo. Se necesita tener esta librería para habilitar sus funciones
<code>myservo.attach()</code>	Adjunta la variable del Servo a un pin. Para poderle dar uso a nuestra salida.
<code>Serial.begin(9600)</code>	Establece la velocidad de los datos en bits por segundo.
<code>analogRead()</code>	Sirve para poder hacer una lectura en una de nuestras entradas análogas de nuestro Arduino y poder diferenciar en una escala, no solo en 1 y 0
<code>servo.write()</code>	Escribe un valor para el servo, controlando el movimiento del mismo. (0-180 grados)
<code>delay()</code>	Sirve para poder tener un tiempo de espera. De esta forma le damos oportunidad al programa de hacer sus funciones de una manera más despacio.
<code>Serial.println()</code>	Imprime datos en el puerto serie como texto ASCII legible.

Tabla 2.- Funciones de ARDUINO utilizadas en el programa

Desarrollo experimental

Descripción paso a paso de como se realizo la práctica

Como parte de nuestro primer proyecto desarrollamos una sombrilla con la capacidad de seguir la luz recibida por resistencias. Como primer paso realizamos un "sketch " de nuestro producto y pensamos el cómo se comportaría la sombrilla.

- Una vez definidos los requerimientos de nuestro proyecto procedimos a realizar el programa de arduino para observar el comportamiento de los servos al iluminar las fotorresistencias. Desafortunadamente el programa no funcionó como se tenía pensado en un inicio y decidimos realizar uno nuevo.
- Para el nuevo programa se consideraron 4 fotorresistencias que satisficieran las condiciones y ciclos necesarios para mover nuestros servos de manera horizontal y vertical. (Más adelante se muestra y explica el programa)
- Una vez que obtuvimos un programa funcional decidimos comenzar con la construcción del prototipo.



Imagen 2.- Prototipo de la sombrilla

- Se probaron distintos acomodos para las resistencias y nos dimos cuenta de que la mejor recepción de luz de manera uniforme era alcanzada al colocar las fotorresistencias en la circunferencia de una botella.

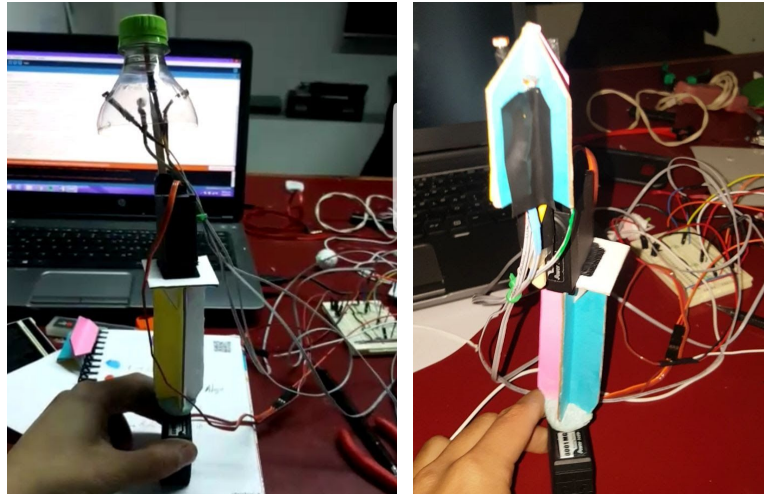


Imagen 3.- Pruebas de recepción de luz

- A pesar de que la botella funcionó muy bien decidimos utilizar el modelo tipo cruz debido a que era mucho más estético. Además dicho modelo nos permitía obtener mejores resultados para el cabeceo del servo vertical, sin embargo en cuanto a la rotación se vio más limitado en comparación al modelo con la botella.
- Ya que funcionó el prototipo se decidió mejorar la apariencia del proyecto, por lo que el arduino y el protoboard fueron colocados en una pequeña caja.

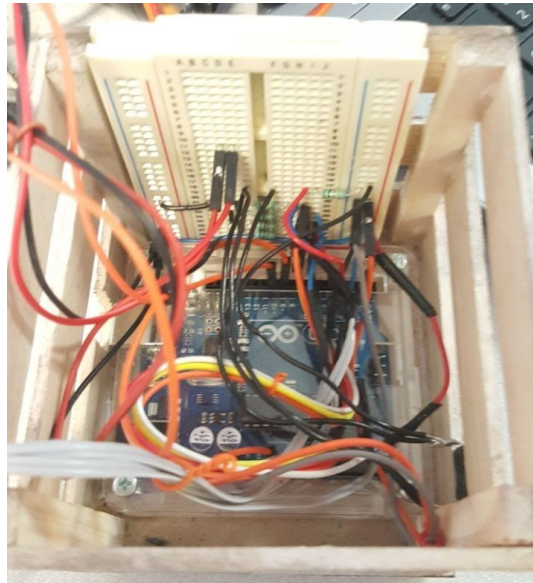


Imagen 4.- Cableado del diagrama de conexión

- Con el circuito cableado dentro de la caja se montó el proyecto sobre una pequeña base.

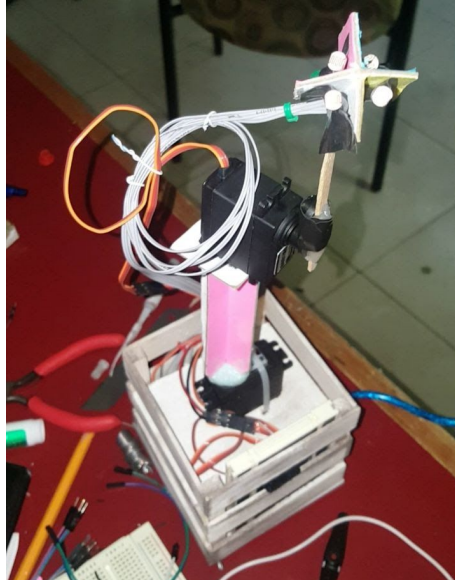


Imagen 5.- Prototipo montado en base

- Como último paso se cambiaron los ejes de rotación por unos más rígidos y estéticos, además, se colocó una sombrilla en el eje del servo vertical.



Imagen 6.- Proyecto terminado

Cálculos y simulación

Como parte de los cálculos realizados se encuentra el mismo procedimiento de la práctica anterior para la obtención de las resistencias. Se observó mediante divisor de voltaje que la resistencia de $10K\Omega$ nos daba los valores óptimos para los voltajes mínimos y máximos requeridos. A continuación se demuestra como se realizó el divisor para una de las fotorresistencias. Las otras 3 no fueron calculadas debido a que se infirió que sus valores iban a ser muy cercanos entre si.

$$V_{min} \rightarrow \frac{(5V)(3200\Omega)}{(13200\Omega)} = 1.2121 V$$

$$V_{max} \rightarrow \frac{(5V)(13.27M\Omega)}{(13.28M\Omega)} = 4.996 V$$

Diagramas eléctricos y de conexión.

Para la realización del diagrama se utilizó el software *Tinkercad*. El resultado obtenido fue el siguiente:

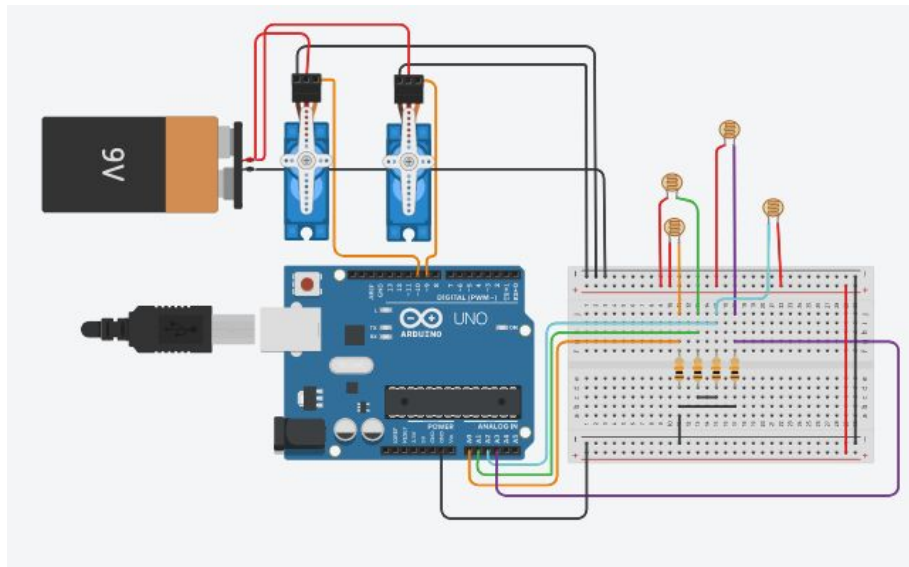


Imagen 7.- Diagrama eléctrico

NOTA: Se utilizó una fuente de voltaje de 6 Volts en lugar de la de 9Volts, sin embargo no se encontró dicha fuente en el simulador

Algoritmo

1. Apagar o encender el programa dependiendo del valor del botón presionado
2. Definir librería para el servo
3. Declarar las variables a utilizar
4. Establecer los valores máximos y mínimos de operación para el servo
5. Definir pines de salida para cada servo
6. Leer los pines de entrada de manera análoga (fotorresistencias)
7. Realizar la autocalibración
8. Calcular los valores máximos y mínimos para el movimiento vertical y horizontal
9. Comparar el valor de la fotorresistencia con el límite definido analíticamente de 50
10. Comparar valor máximo superior y máximo inferior
11. Aumentar una unidad el servo si el valor superior máximo es mayor al inferior máximo hasta que estén en el mismo nivel
12. Si el valor superior máximo es menor al máximo inferior reducir una unidad hasta que estén en el mismo nivel.
13. Enviar modificaciones al servo
14. Realizar lo anterior para el servo horizontal y enviar las modificaciones al servo

Código fuente comentado

```
#include <Servo.h> // libreria para los servo motores

//VARIABLES PROGRAMA GENERAL

int pin0 = 0, pin1 = 1, pin2 = 2, pin3 = 3; // conexión de las fotorresistencias
int valsi, vals, valii, valid; //valores superiores e inferiores
int supmax, infmax, izqmax, dermax; //variables para inferiores y superiores máximos
int dif1, dif2; //variables para almacenar las diferencias entre el máximo superior e inferior/derecha e izquierda

//VARIABLES BOTÓN ENCENDIDO Y APAGADO

const int reset = 3; // Pin del botón de apagado/encendido
const int led = 13; // Pin del LED de apagado
int estado=0; //Variable que recibirá valores
```

```
//VARIABLES AUTOCALIBRACIÓN
```

```
int sensor1=0, sensor2=0, sensor3=0, sensor4=0;  
int sensormin1=1023, sensormin2=1023, sensormin3=1023, sensormin4=1023;  
int sensormax1=0, sensormax2=0, sensormax3=0, sensormax4=0;
```

```
Servo horizontal; // Servo horizontal  
int servoh = 180; //Variable con valor de 180 definido para el servo horizontal  
int servohLimitHigh = 180; //limite en posición derecha máxima  
int servohLimitLow = 0; //limite para posición izquierda máxima
```

```
Servo vertical; // Servo vertical Servo  
int servov = 45; //Variable con valor de 180 definido para el servo horizontal  
int servovLimitHigh = 180; //limite en posicion superior  
int servovLimitLow = 15; //limite para posicion inferior
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600); // Inicializa función para imprimir valores más adelante  
  horizontal.attach(9); //servo horizontal pin digital 9  
  vertical.attach(10); // servo vertical pin digital 10  
  while (millis() < 10000)  
  {  
    sensor1=analogRead(pin0);  
    sensor2=analogRead(pin1);  
    sensor3=analogRead(pin2);  
    sensor4=analogRead(pin3);  
    horizontal.attach(9); //servo horizontal pin digital 9  
    vertical.attach(10); // servo vertical pin digital 10
```

```
  // CALIBRACIÓN DURANTE 5 SEGUNDOS
```

```
  if (sensor1 > sensormax1) // Registrar valor máximo del sensor 1  
  {  
    sensormax1 = sensor1;  
  }  
  if (sensor1 < sensormin1) // Registrar valor minimo del sensor 1  
  {  
    sensormin1 = sensor1;  
  }
```

```
if (sensor2 > sensormax2) // Registrar valor máximo del sensor 2
{
    sensormax2 = sensor2;
}
if (sensor2 < sensormin2) // Registrar valor minimo del sensor 2
{
    sensormin2 = sensor2;
}
if (sensor3 > sensormax3) // Registrar valor máximo del sensor 3
{
    sensormax3 = sensor3;
}
if (sensor3 < sensormin3) // Registrar valor minimo del sensor 3
{
    sensormin3 = sensor3;
}
if (sensor4 > sensormax4) // Registrar valor máximo del sensor 4
{
    sensormax4 = sensor4;
}
if (sensor4 < sensormin4) // Registrar valor minimo del sensor 4
{
    sensormin4 = sensor4;
}
}

}

void loop()
{
    valsi = analogRead(pin0); // lee el valor del fotorresistor (0-510) (superior izquierdo)
    valsd = analogRead(pin1); // lee el valor del fotorresistor (511-1023) (superior derecho)
    valii = analogRead(pin2); // lee el valor del fotorresistor (0-510) (inferior izquierdo)
    valid = analogRead(pin3); // lee el valor del fotorresistor (511-1023) (inferior derecho)

    supmax = (valsi + valsd) / 2; // valor superior maximo
    infmax = (valii + valid) / 2; // valor inferior maximo
    izqmax = (valsi + valii) / 2; // valor izquierdo máximo
    dermax = (valsd + valid) / 2; // valor derecho máximo
```

```
dif1 = supmax - infmax; // diferencia entre superior e inferior
dif2 = izqmax - dermax; // diferencia entre izquierdo y derecho
```

```
Serial.print(supmax); // Imprime valor de supmax para determinar la tolerancia
Serial.print(" ");
```

```
estado=digitalRead(reset); // Lee el valor del botón de apagado
(Encendido/Apagado)
```

```
//CONTROL VERTICAL
```

```
if (estado==LOW)
{
  if (-50 > dif1 || dif1 > 50) // revisa la diferencia para cambiar al ángulo vertical
  {
    if (supmax > infmax) // revisa la diferencia para cambiar al ángulo vertical
    {
      servov = ++servov; // Aumenta una unidad cada vez que se cumpla el ciclo
      if (servov > servovLimitHigh)
      {
        servov = servovLimitHigh; // Establece un valor nuevo para servov
      }
    }
  }
}
```

```
else if (supmax < infmax) // Si no se cumple lo anterior se ejecuta la siguiente sentencia
{
  servov= --servov; // Disminuye una unidad cada vez que se cumpla el ciclo
  if (servov < servovLimitLow)
  {
    servov = servovLimitLow; // Establece un valor nuevo para servov
  }
}
vertical.write(servov); // Manda la señal al servo para ejecutar el movimiento
}
```

```
//CONTROL HORIZONTAL
```

```
if (-50 > dif2 || dif2 > 50) // revisa la diferencia para cambiar el ángulo horizontal
{
  if (izqmax > dermax) // revisa la diferencia para cambiar el ángulo horizontal
  {
    servoh = --servoh; // Disminuye una unidad cada vez que se cumpla el ciclo
```

```
    if (servoh < servohLimitLow)
    {
        servoh = servohLimitLow; // Establece un valor nuevo para servoh
    }
}

else if (izqmax < dermax) // Si no se cumple lo anterior se ejecuta la siguiente sentencia
{
    servoh = ++servoh; // Aumenta una unidad cada vez que se cumpla el ciclo

    if (servoh > servohLimitHigh)
    {
        servoh = servohLimitHigh; // Establece un valor nuevo para servoh
    }
}

horizontal.write(servoh); // Manda la señal al servo para ejecutar el movimiento
}
delay(50); // Tiempo de reacción del servo
}

Else // Si no se cumple la sentencia del botón de encendido, se ejecuta la siguiente
sentencia
{
    digitalWrite(led, HIGH); // Encender LED
}
}
```

Como se puede observar el código que utilizamos nos funcionó para la parte 1 y 2. Únicamente agregamos algunas instrucciones extra para que la lectura de datos en lugar de venir del potenciómetro fuera realizada por las fotorresistencias. Por su parte se crearon nuevas variables (val, val2 y val3) y mediante las funciones de Serial.println calibramos los parámetros para cada una de las 2 variables.

Como se mencionó anteriormente en un inicio nuestro programa no funcionó, sin embargo solucionamos el problema al asignar un mínimo y un máximo de 5° y 95° a una de las variables, mientras que la otra su mínimo fue el máximo previamente

obtenido de 95° y su máximo respectivo fue de 175° . El circuito funcionó correctamente después de realizar las respectivas modificaciones.

Análisis de resultado

Nuestros resultados fueron variando pues en un inicio se fue trabajando con prueba y error. Hicimos una relación para las salidas para que fueran funcionales y poder tener un código de programación que hiciera exactamente lo que le estábamos pidiendo.

Una vez teniendo el código tuvimos que tener cuidado en dónde ponemos los fotoresistores pues cada uno cumple con una función distinta y se colocaron de forma precisa para su buen funcionamiento. Lo probamos pero en una maqueta provisional para asegurarnos que todo el funcionamiento era el adecuado. Decidimos pasarlo en una maqueta de calidad a la que se le invirtió tiempo para que el aspecto estético fuera sobresaliente y tener un trabajo de mejor calidad pero vimos que no era lo mismo que tenerlo en la base de prueba pues se ajustó todo para hacerlo en el prototipo.

Tuvimos que ajustar nuestra sombrilla estética con un híbrido de nuestro prototipo para poder hacerlo funcionar adecuadamente pero la calidad estética fue un poco menor. Fue un poco complejo pues debíamos juntar el protoboard en una cajita junto al arduino y un servomotor pues queríamos que la maquinaria estuviera cubierta.

Al preparar todo el proyecto para poder presentarlo se nos fueron presentando bastantes problemas que fueron muy tardados en resolverse. El proyecto empezó a vibrar mucho pues la base no era lo suficientemente rígida y afectaba bastante con la posición de la sombrilla. También nos fuimos dando cuenta que poco a poco se iba moviendo más lento y no sabíamos por qué. Tuvimos un problema bastante fuerte pues ya no respondía el proyecto. Creíamos que había sido por daño en los fotoresistores y los cambiamos. Al momento de cambiarlos vimos que el proyecto hacía lo mismo y no funcionaba.

Lo conectamos a una fuente de voltaje y empezó a reaccionar pues lo teníamos alimentado los servomotores del arduino. Tuvimos que reajustar los fotoresistores pues al cambiarlos se movieron de lugar. Al final pudimos lograr de nueva cuenta que funcionara correctamente nuestro proyecto pero se nos generaron bastantes problemas por la alimentación de nuestros servomotores. Ahora sabemos cómo actuar ante una situación similar pues al no hacerlo de la manera adecuada hicimos un gran cambio que nos costó mucho tiempo.

No tenemos otro resultado más que el funcionamiento adecuado del proyecto pues no se hizo ninguna medición para el proyecto, eso porque no era necesario. El único cálculo que pudimos haber utilizado sería el de las resistencias que debíamos usar para las fotoresistencias haciendo que su voltaje máximo fuera de 5V. Este cálculo ya lo teníamos por uso y aplicación de prácticas pasadas.

Conclusiones

Como conclusiones tenemos que fue un trabajo bastante pesado y un poco complicado pues tan solo desarrollar la lógica fue un verdadero reto. Tuvimos mucho que pensar y analizar para poder lograr el objetivo de nuestro proyecto. Aparte de que en esta ocasión la presentación era bastante importante y también fue complicado acomodar todo adecuadamente sin que los cables se salieran del protoboard ni del arduino.

De alguna forma fue bastante entretenido pues nos sentimos en la industria generando un nuevo producto el cual era importante que funcionara como se esperaba. Sabemos que en la industria nos vamos a enfrentar a situaciones similares a esta y vamos a necesitar lograr el objetivo de los proyectos si es que en verdad queremos ser grandes ingenieros y sobretodo lograr conseguir puestos importantes.

Nos quedamos con nuevas experiencias y conocimientos del funcionamiento de algunas herramientas y del adecuado funcionamiento. Pues al usar los servomotores y alimentarlos con el arduino vimos que poco a poco iban funcionando deficientemente. Aún es una duda para nosotros el porque funcionaban perfectamente en un inicio y después fueron fallando pero sabemos como debemos

usarlo en adelante para no tener problema con su movilidad. También aprendimos bastante de los fotoresistores pues creíamos que trabajar con estos iba a ser complicado porque se ven frágiles por la función que pueden tener pero su rendimiento fue sorprendente y vimos que no es tan sencillo dañarlos.

En general fue sorprendente ver la unión de todo el trabajo funcionando al final. Estamos muy contentos con los resultados y esperamos seguir pasando y enfrentando de forma adecuada las siguientes pruebas y prácticas que prosigan.

Fuentes o Referencias

5HzE. (2018). 5 Hertz. Obtenido de Electronica:
https://www.5hertz.com/index.php?route=tutoriales/tutorial&category_id=1&tutorial_id=11#2

AF. (01 de 2018). AsFunciona. Obtenido de
http://www.asifunciona.com/electrotecnia/ke_resistencia/ke_resistencia_1.htm

Agustín J. Frascino . (2018). Tecnoeducacion. Obtenido de
<https://tecnoedu.com/F1000/ModuloV.php>

CDefinicion. (2014). Concepto definicion. Obtenido de
<http://conceptodefinicion.de/fuente-poder/>

CE. (01 de 2018). CircuitosElectronicos. Obtenido de
<http://www.circuitoselectronicos.org/2007/10/el-protoboard-tableta-de-experimentacin.html>

Equipos y laboratorio. (2015). Obtenido de
http://www.equiposylaboratorio.com/sitio/contenidos_mo.php?it=1484

FT. (01 de 2018). FinalTest. Obtenido de
<https://www.finaltest.com.mx/product-p/art-8.htm>

Hayt, W. H., Kemmerly, J. E., & Durbin, S. M. (2012). Análisis de circuitos en ingeniería. México: McGraw Hill.

Ing.F. (2018). Ingeniería mecafenix . Obtenido de
<http://www.ingmecafenix.com/electronica/potenciometro/>

Luis Llamas. (2018). Electrónica, Informatica y Diseño. Obtenido de
<https://www.luisllamas.es/medir-nivel-luz-con-arduino-y-fotoreistencia-ldr/>

PH. (2017). Panama Hitek. Obtenido de
<http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>

SEE. (01 de 2018). Simbología Electricos y Electronicos. Obtenido de
<http://www.simbologia-electronica.com/simbologia-electrica-electronica/codigos-resistencias.htm>