**“SISTEMA AUTOMÁTICO DE CONTROL INALÁMBRICO PARA PERSIANAS MEDIANTE UNA APLICACIÓN (AUTOMATIC BLINDS)”**

**UNIVERSIDAD MEXIQUENSE DEL BICENTENARIO**



**MANUAL TÉCNICO**

**INTEGRANTES:**

MIRANDA SILVA SALMA IVONNE.

GONZÁLEZ ESCALERA SEBASTIAN MARTÍN.

**CARRERA:**

INFORMÁTICA.

**SEMESTRE:**

7MO SEMESTRE.

**GRUPO:**

LI-171.

**PROFESOR:**

ARACELI ROSALES BLAS.

**15/01/2022**

## 

**ÍNDICE**

[PRESENTACIÓN 1](#_Toc93057463)

[OBJETIVO 1](#_Toc93057464)

[REQUISITOS DEL SISTEMA 2](#_Toc93057465)

[HERRAMIENTAS PARA EL DESARROLLO 3](#_Toc93057466)

[DIAGRAMA DE CONEXIÓN DE LA PLACA ARDUINO CON EL MOTOR A PASOS 5](#_Toc93057467)

[CIRCUITO DEL SISTEMA 6](#_Toc93057468)

[CÓDIGO CARGADO EN ARDUINO 8](#_Toc93057469)

[CÓDIGO EN JAVA 16](#_Toc93057470)

[CÓDIGO OBJETOS DE INTERFAZ JAVA 21](#_Toc93057471)

[INTERFAZ DEL PANEL DE CONTROL DEL MOTOR 22](#_Toc93057472)

[PROTITIPO DEL SISTEMA 23](#_Toc93057473)

[PREGUNTAS FRECUENTES 26](#_Toc93057474)

[REFERENCIAS 27](#_Toc93057475)

# **PRESENTACIÓN**

El siguiente manual guiará a técnicos que ofrecerán soporte al sistema, el cual les dará a conocer los requerimientos y la estructura para la construcción del sistema, así como la aplicación móvil conectados mediante una placa Arduino, mostrando las herramientas necesarias para la construcción y la funcionalidad del sistema.

# **OBJETIVO**

Informar y especificar a los técnicos la estructura y conformación del sistema con el fin de poder otorgar las bases para el mantenimiento, soporte y posibles modificaciones para el sistema en general.

# **REQUISITOS DEL SISTEMA**

* Requerimientos de Hardware (para el desarrollo o mantenimiento de la aplicación)

-Equipo de Cómputo, Teclado, Mouse/ Laptop

-Procesador Dual Core a 2.4 GHz o superior.

-Memoria RAM 4 GB o superior

-3 GB de espacio libre en disco

-Tarjeta de red LAN y/o Wireless

-Placa Arduino UNO

-Jumpers macho-hembra

-Driver ULN2003

* Requerimientos de Software

-Sistema Operativo Windows 7 o posterior

-Java (JDK) 16.0.2.0 o superior

-Librería RXTX para java versión 2.1 o superior

-NetBeans 12.4 o superior

-Arduino IDE 1.8.15

-Librería Separador para Arduino (opcional)

-Sistema Operativo móvil emulado (Android 7.0 en adelante)

# **HERRAMIENTAS PARA EL DESARROLLO**

**JAVA**

Java es un lenguaje sencillo y orientado a objetos, que permite el desarrollo de aplicaciones en diversas áreas, como seguridad, animación, acceso a bases de datos, aplicaciones cliente-servidor, interfaces gráficas, páginas Web interactivas y desarrollo de aplicaciones móviles, entre otras.

Una de sus principales características es la creación de módulos reutilizables, que funcionan sin la necesidad de conocer su estructura interna. Esto permite al usuario añadir nuevos módulos, además de obtener programas independientes de la plataforma en la cual fueron desarrollados, gracias a la implementación de la llamada Máquina Virtual de Java (JVM) (UNAM, 2021).

**NETBEANS IDE**

Netbeans es un IDE (Integrated Development Environment) o entorno de desarrollo integrado, que es gratuito y de código abierto. Sirve para el desarrollo de aplicaciones web, corporativas, de escritorio y móviles que utilizan plataformas como Java y HTML5, entre otras.

Se trata de un editor de código fuente, junto con recursos de construcción automáticos y un depurador. Igualmente, brinda la función de autocompletado inteligente de código o IntelliSense. Y en el caso de la plataforma de Netbeans IDE, ofrece un compilador y un intérprete (Fantino, 2021).

**LIBRERÍA RXTX**

RXTX es una biblioteca de clase Java de código abierto que proporciona comunicación en serie y paralela. El proyecto RXTX proporciona la implementación de la API del paquete de comunicación serie javax.comm compatible en los sistemas operativos Windows, Linux, Mac os X, Solaris para que otros desarrolladores desarrollen aplicaciones en serie en dichos sistemas (Programador clic, 2020).

**ARDUINO IDE**

Arduino IDE es un entorno de desarrollo y en él se realiza la programación de cada una de las placas de Arduino. Tiene como base el entorno de Processing al igual que un lenguaje de programación fundamentado en Wiring. El software Arduino de código abierto facilita la escritura de código y su carga en la placa. Pudiendo utilizarlo con cualquier placa Arduino (Tuelectronica, s.f.).

**LIBRERÍA SEPARADOR**

Librería desarrollada por el usuario “IoTicos” para Arduino que permite simplificar el proceso y el código para extraer caracteres de una cadena.

# **DIAGRAMA DE CONEXIÓN DE LA PLACA ARDUINO CON EL MOTOR A PASOS**

La conexión se compone de los siguientes elementos:

-Equipo de computo con el software especificado en los requisitos

-Conector Plug 2.1 Mm

-Batería cuadrada de 9V

-Placa Arduino UNO

-Jumpers macho-hembra

-Driver ULN2003

-Motor a pasos unipolar 28BYJ-48 de 4 bobinas

La inclusión del driver no es obligatoria, pero es recomendable para proteger el equipo de variaciones en el voltaje de entrada y facilitar la conexión de los componentes.

Diagrama, Esquemático

Descripción generada automáticamente

# **CIRCUITO DEL SISTEMA**

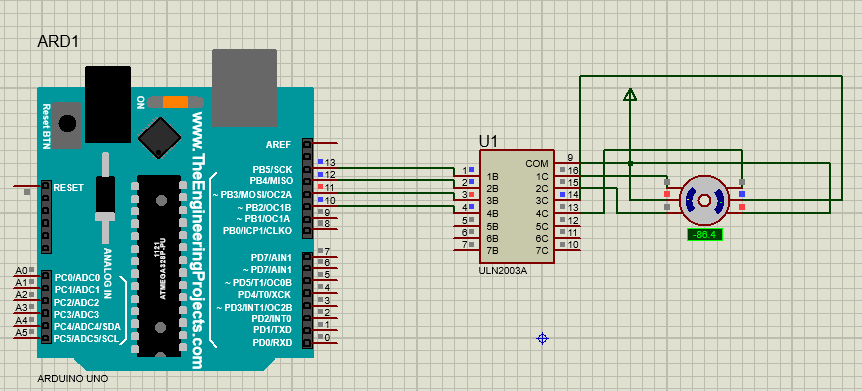
Las conexiones de los jumpers deben realizarse como se especifica en el siguiente diagrama, el orden de los colores no influye en la funcionalidad del circuito, sin embargo, se recomienda utilizarlo para seguir el modelo estándar y facilitar la comprensión de las conexiones.

Diagrama

Descripción generada automáticamente

La placa nos ofrece una mayor cantidad de entradas a las especificadas, es importante recalcar que deben conectarse los puertos especificados en el circuito para que el sistema pueda funcionar.

Puede hacer uso del software proteus para realizar el circuito en una simulación y probar la funcionalidad del código, siendo necesario incluir en el software las librerías correspondientes a los componentes de Arduino:



# **CÓDIGO CARGADO EN ARDUINO**

El código se encuentra comentado para facilitar su comprensión, de manera general se incluye la librería Separador para su posterior uso, se crean las variables de las cuales algunas almacenarán datos que se recibirán mediante el código de la aplicación desarrollada con Java y el método Void setup en el cual se nos permitirá configurar los puertos del Arduino en modo salida.

Texto

Descripción generada automáticamente

A continuación, trabajamos con el método Void loop, el código de dicho método estará ejecutándose todo el tiempo en la placa, dentro de este bucle se verifican los datos que se reciban mediante el puerto serial (los que se enviaran mediante la aplicación) y se procesaran estos datos de tal manera que, los valores numéricos se interpreten como grados y la selección de un tipo de movimiento especifico.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Después, los datos ingresados se procesan para poder dar una vuelta en la cantidad de grados especificada, con el tipo de movimiento especificado y en base a si el valor en grados es positivo o negativo, se dará una vuelta a la izquierda o a la derecha, todo ello apoyado en el uso de métodos para simplificar el código.

Texto

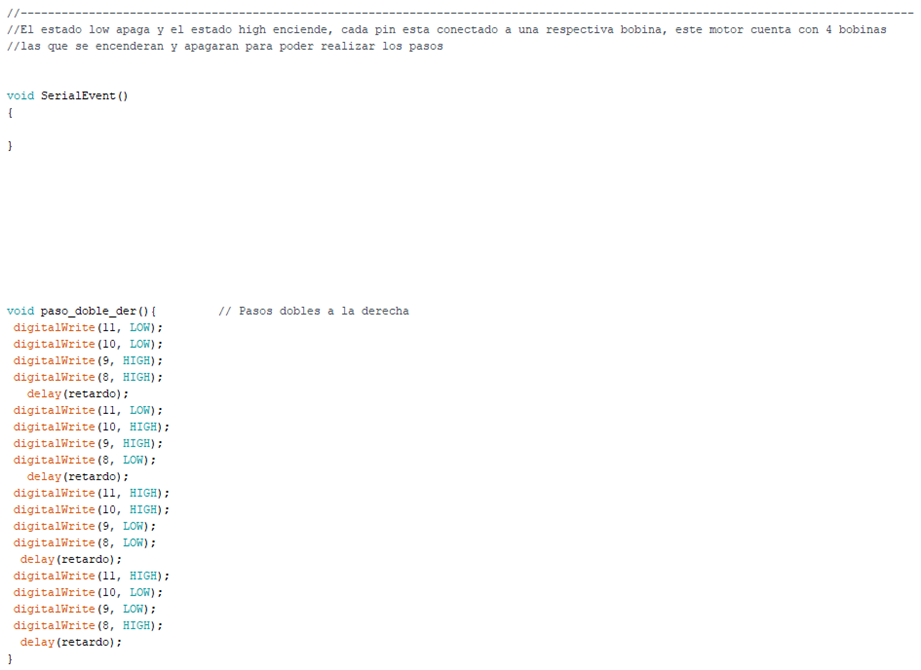
Descripción generada automáticamente

Una vez hecha la operación, se vacían los datos ingresados y se apaga el motor

Texto

Descripción generada automáticamente

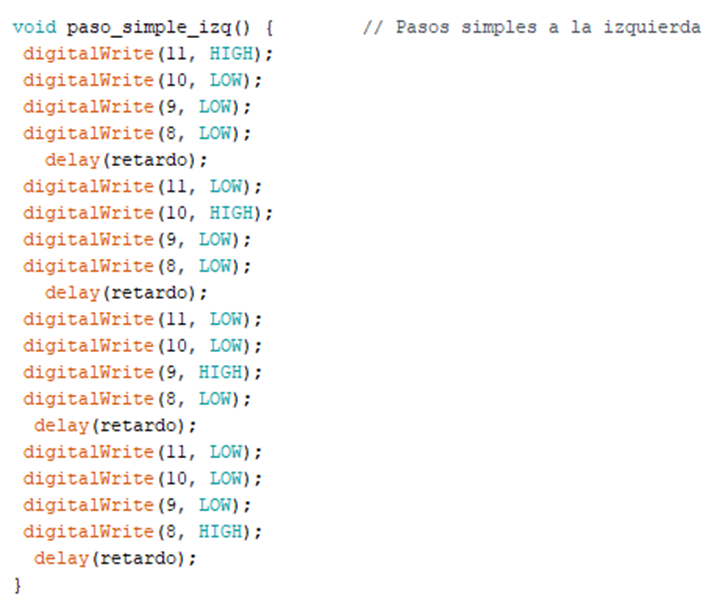
Las siguientes capturas pertenecen a los métodos que son utilizados en el código:

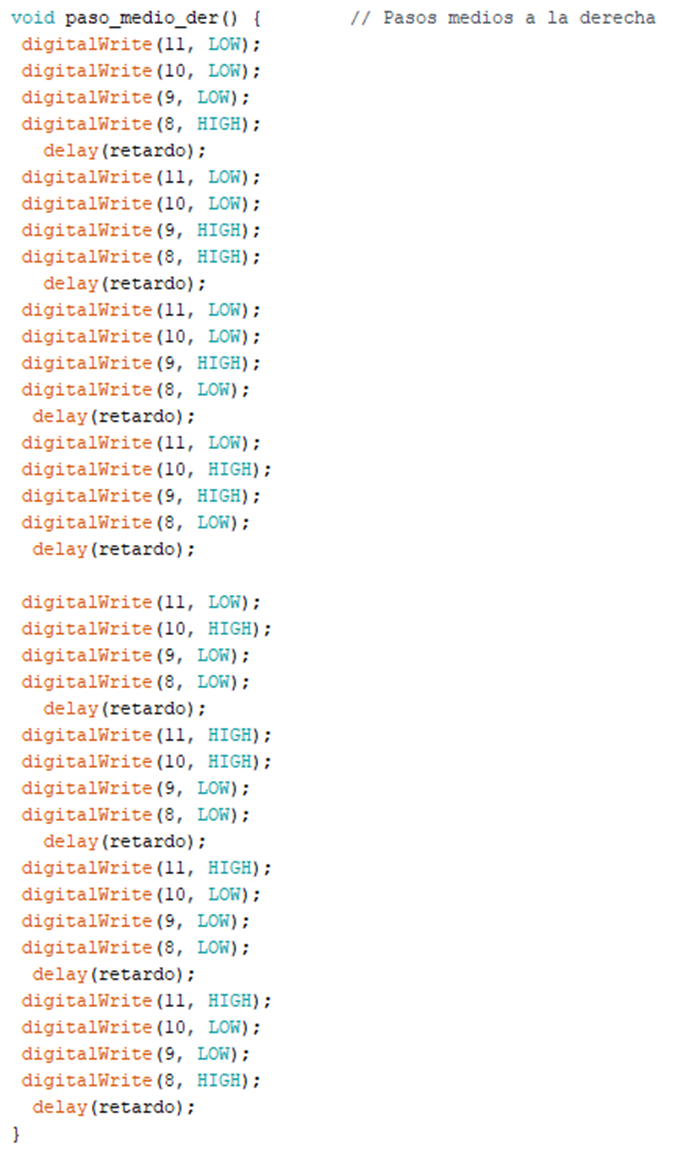
****

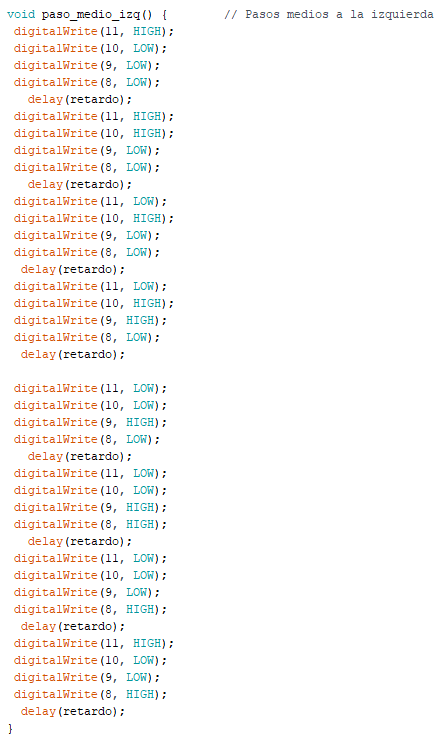


En ellos se especifican los puertos mediante los que se enviara la corriente para poder hacer el movimiento especificado considerando que el motor utilizado es de 4 bobinas:

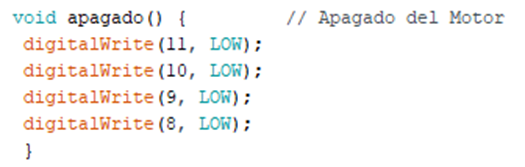








Y para finalizar, el método que dejara sin corriente todos los puertos de salida apagando el motor

****

# **CÓDIGO EN JAVA**

El código de Java se centra en la aplicación que utilizara el usuario para pasar los datos con los que operara el código cargado en Arduino:

Texto

Descripción generada automáticamente

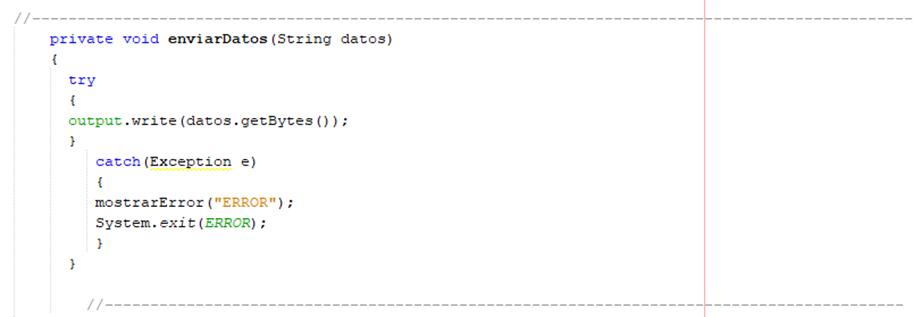
Primero se integran las librerías necesarias para poder comunicarse con el Arduino (componentes de RXTX), se declaran valores constantes que se ocuparan posteriormente para ofrecer movimientos predefinidos al usuario (valores en grados) y configura el puerto conectado con el cable serial para ser utilizado.



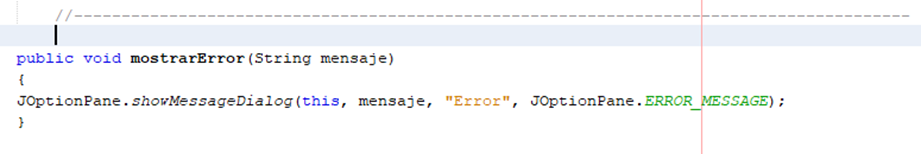
Estas líneas le dan aspecto al panel de control de la aplicación e inicializan la conexión serial cuando esta es cargada.



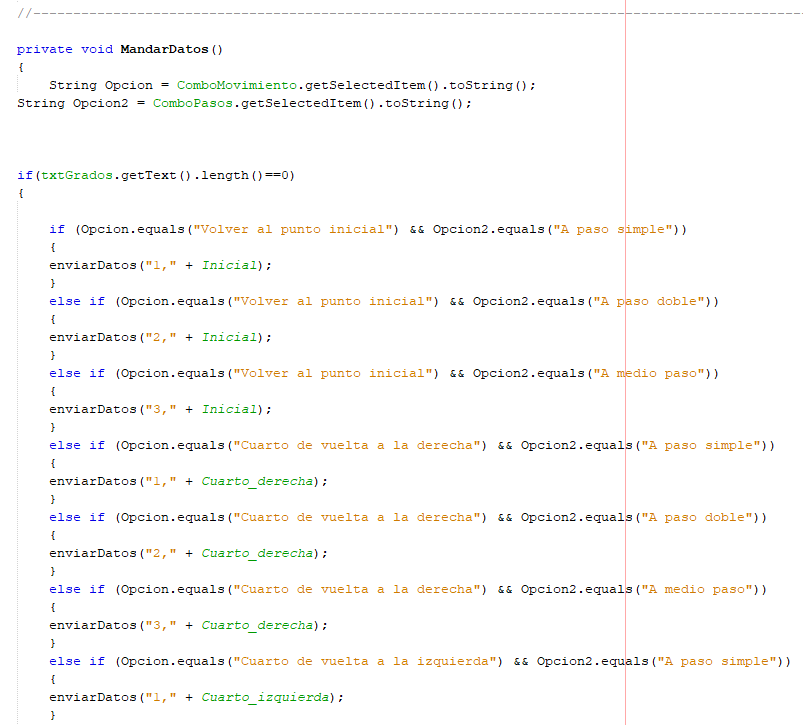
Se crea el método que inicializa la conexión y se verifica que este exista y se este utilizando, en caso de encontrar alguna anomalía en la conexión la aplicación notificara al usuario y se cerrara.



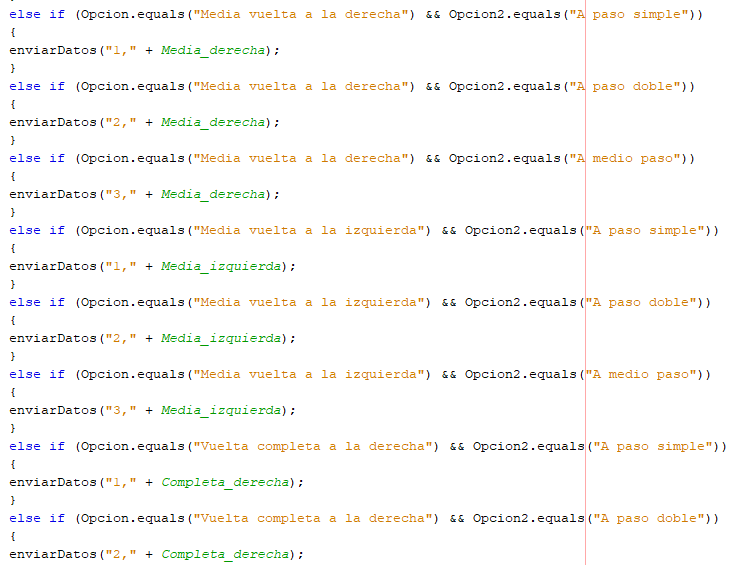
Esta verificación se realizará también previo el envió de datos, de esta manera se verifica que la conexión sea correcta antes de intentar hacer operaciones.

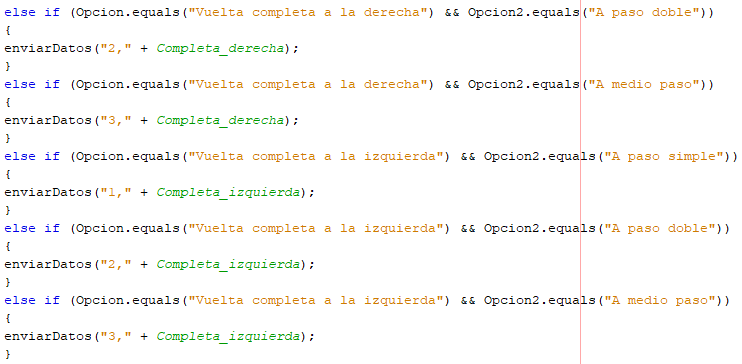


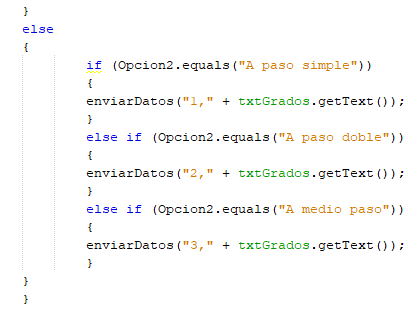
Mientras este método será el que permita generar los mensajes que notificaran al usuario de dichos errores.



Luego tenemos un método que permitirá extraer los valores seleccionados por el usuario de una lista predefinida de movimientos en un objeto combo box, esto condicionado a que en una caja de texto donde el usuario podrá ingresar valores propio este vacía, si por el contrario esta caja tiene algún valor, se ejecutara este en su lugar, de la misma manera se evaluaran las distintas opciones seleccionadas entre if anidados.







# **CÓDIGO OBJETOS DE INTERFAZ JAVA**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Este código aplicado a los objetos de la interfaz, permite la caja de texto donde el usuario puede ingresar valores propios acepte solo números y que, adicional a ello, el usuario pueda confirmar la inserción de los valores solo presionando una tecla (Enter).

# **INTERFAZ DEL PANEL DE CONTROL DEL MOTOR**

Permite 3 tipos de movimientos del motor a pasos Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Botón que ejecuta los movimientos seleccionados.

Permite ingresar el número de grados a girar para el motor a pasos

Botón que cierra la ventana del panel de control de movimiento.

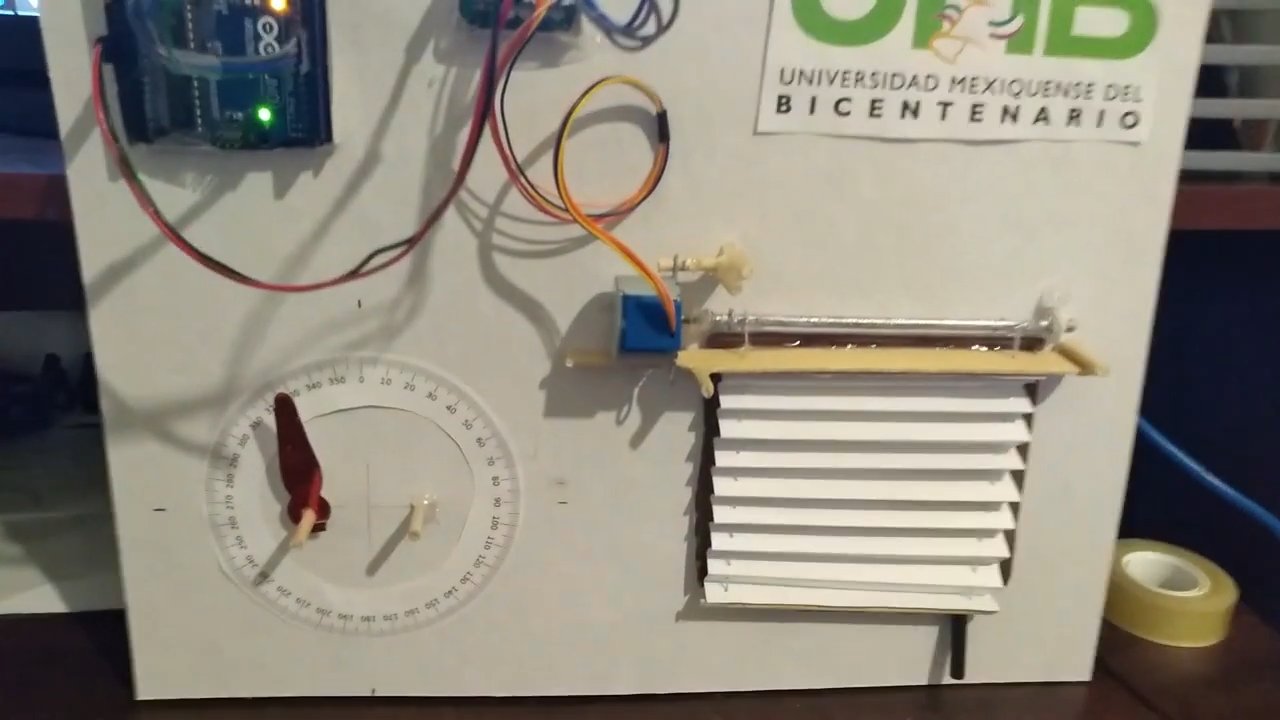
Permite seleccionar el movimiento que desea que de el motor a pasos

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

# **PROTITIPO DEL SISTEMA**

Para el prototipo se realizo una pequeña persiana a escala armada con el circuito, papel cascaron, madera, silicón, un pequeño tubo de plástico, hoja opalina delgada e hilo.



Esta persiana queda totalmente cerrada a los 0° grados y totalmente cerrada a los 1600° grados, los componentes utilizados fueron los especificados en los requerimientos:

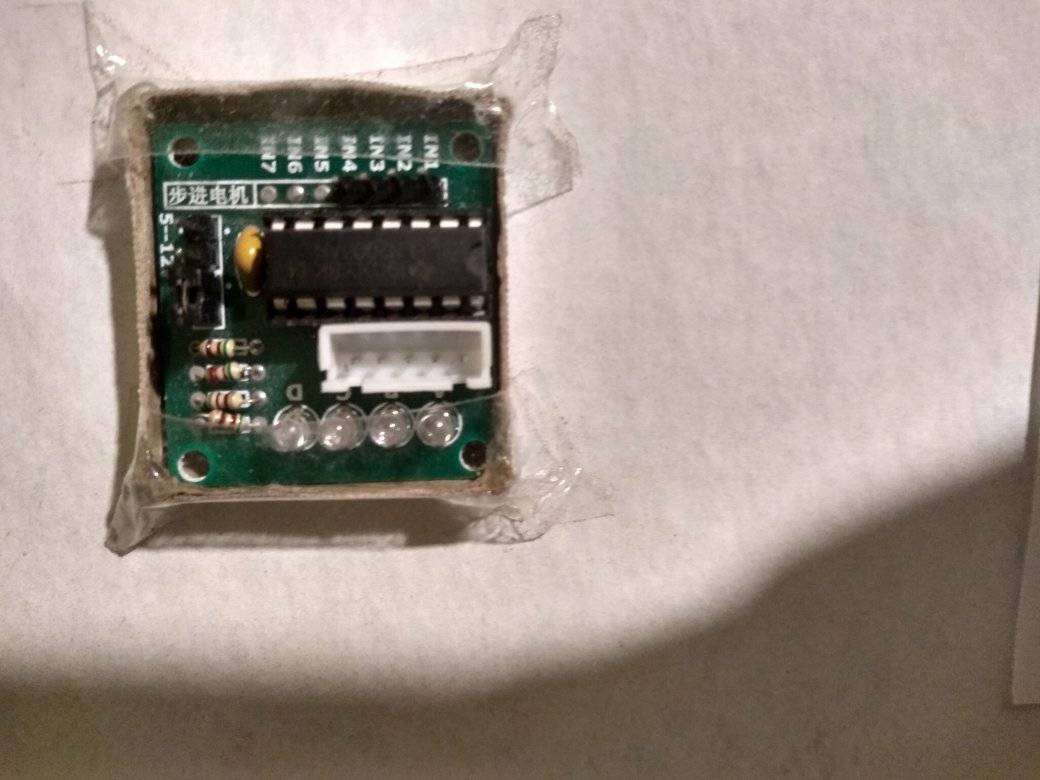
El motor a pasos unipolar:



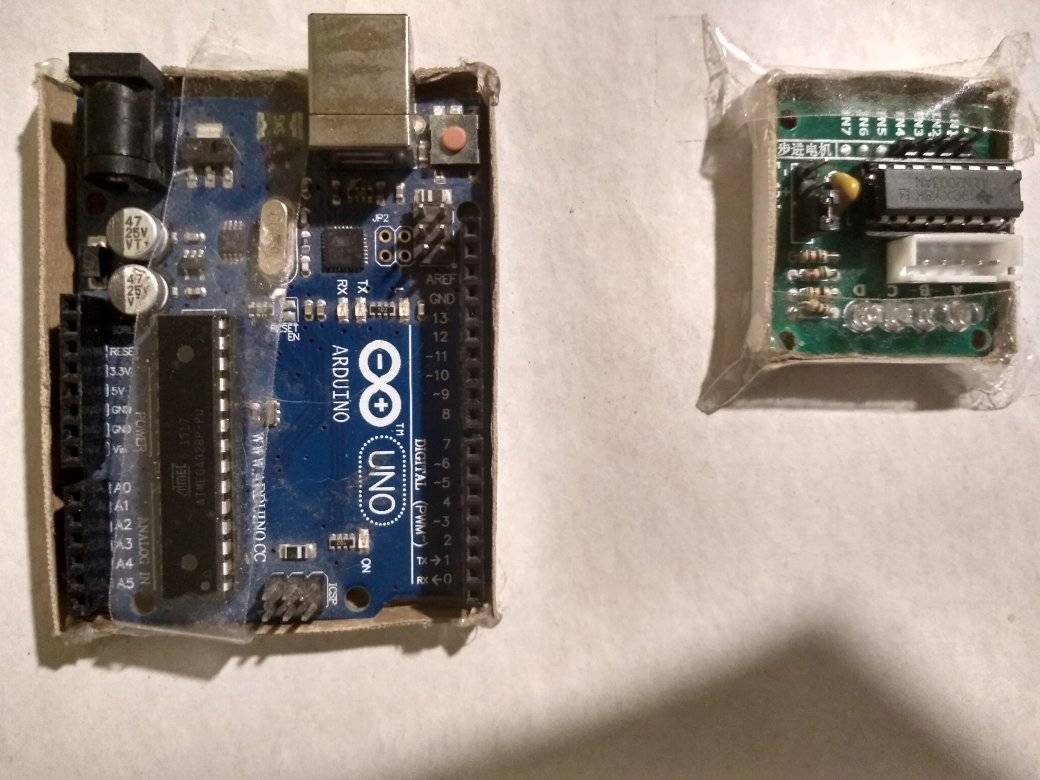
Los jumpers macho-hembra:



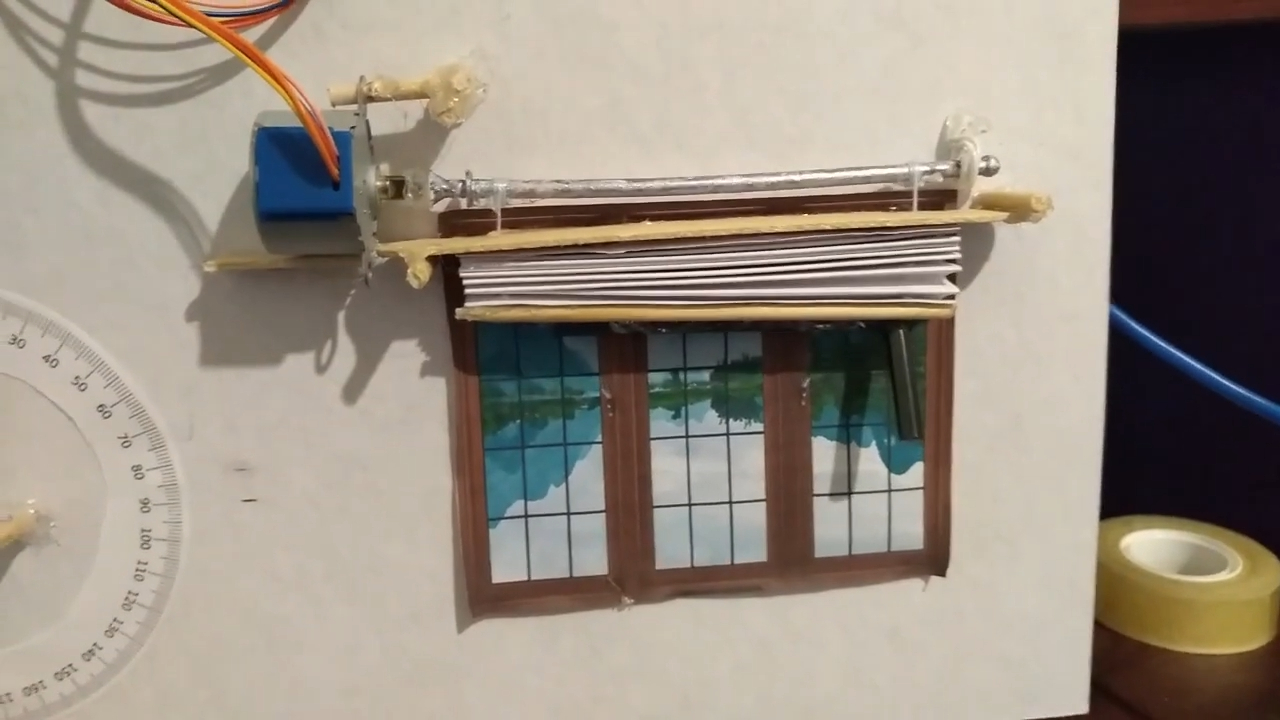
El driver:



Y la placa Arduino UNO



El prototipo funciona gracias al movimiento del motor, que permite dar vueltas al tubo de la persiana, al mismo tiempo este tubo enrolla el hilo que cruza la hoja doblada en zigzag, esto permite que la hoja suba y baje como lo haría la persiana.



# **PREGUNTAS FRECUENTES**

|  |  |
| --- | --- |
| **PREGUNTA** | **SOLUCIÓN** |
| Quiero conectar el sistema a una placa Arduino o driver distintos a los especificados, ¿Qué puedo hacer? | Puede realizar la conexión normalmente, pero debe asegurarse de que las conexiones especificadas en el circuito sean iguales revisando el PCB (circuito impreso) de sus componentes. |
| El código no compila, me muestra errores ¿Qué puedo hacer? | Asegúrese de que ha descargado he instalado las librerías RXTX y Separador en sus respectivos entornos de desarrollo correctamente, así como integrarlas en el proyecto. |
| La aplicación funciona, pero la persiana no se mueve en base a los valores especificados | Asegúrese de que ha conectado los jumpers en el orden especificado y colocado el sistema con la persiana cerrada y el motor en la posición inicial (grado 0). |
| La aplicación no detecta el sistema conectado mediante el puerto serial y se cierra ¿Qué puedo hacer? | Ubique en el código de Java el método de conexión y el puerto que se está inicializando (COM3 por defecto) y cámbielo por el puerto que esté utilizando su equipo, puede consultar este dato en la interfaz de desarrollo de Arduino en Menú->Herramientas->Puerto. |
| Quiero utilizar un modelo de motor unipolar diferente al especificado ¿Qué puedo hacer? | Debe consultar las especificaciones de su motor y verificar que el voltaje de salida aceptado sea de 9 a 12v, la conexión sea compatible con su driver y adaptar el código al nuevo motor.  Es necesario adaptar la regla de tres que adapta el número de pasos que ofrece el motor a grados, la cantidad de bobinas que utiliza el motor y los tipos de movimiento que ofrece. |

# **REFERENCIAS**

Fantino, J. (Noviembre de 2021). *Crehana*. Obtenido de https://www.crehana.com/mx/blog/desarrollo-web/que-es-netbeans/

*Programador clic*. (2020). Obtenido de https://programmerclick.com/article/8673739341/

*Tuelectronica*. (s.f.). Obtenido de https://tuelectronica.es/que-es-arduino-ide/

UNAM. (Febrero de 2021). Obtenido de https://docencia.tic.unam.mx/presenciales/Lenguaje-de-programacion-java.html