

**UNIVERSIDAD DE GUADALAJARA  
CENTRO UNIVERSITARIO DE CIENCIAS  
EXACTAS E INGENIERIA**



**PROYECTO MODULAR**

**TITULO:**

**BRAZO ROBOTICO CLASIFICADOR DE PAQUETES**

**PRESENTADO POR  
Carlos Adolfo Aguilar Rodríguez**

**ASESOR DEL PROYECTO  
Dr. David Israel Rodríguez Castellanos**



## UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTA E INGENIERÍAS  
DIVISIÓN DE TECNOLOGÍAS PARA LA INTEGRACIÓN CIBER-HUMANA  
COORDINACION DE COMUNICACIONES Y ELECTRONICA

### TITULO PROYECTO MODULAR

BRAZO ROBOTICO CLASIFICADOR DE PAQUETES

### AREA DE PARTICIPACIÓN

DESARROLLO TECNOLÓGICO: AUTOMATIZACION

### PROYECTO MODULAR:

**MÓDULO I.** ELECTRONICA ANALOGICA

**MODULO II.** ELECTRONICA DIGITAL

**MODULO III.** COMUNICACIONES

**MODULO IV.** INSTRUMENTACION Y CONTROL

---

### INTEGRANTES DEL EQUIPO

Nombre	Código	Correo
Carlos Adolfo Aguilar Rodríguez	215860049	<a href="mailto:Carlos.aguilar@alumnos.udg.mx">Carlos.aguilar@alumnos.udg.mx</a>

### ASESOR DEL PROYECTO

Nombre	Código	Correo
David Israel Rodríguez Castellanos	2959101	<a href="mailto:david.rcastellanos@academicos.udg.mx">david.rcastellanos@academicos.udg.mx</a>

<b>MISION .....</b>	<b>4</b>
<b>VISION .....</b>	<b>4</b>
<b>DESCRIPCIÓN DEL PROYECTO .....</b>	<b>4</b>
<b>OBJETIVO GENERAL .....</b>	<b>4</b>
<b>OBJETIVOS ESPECIFICOS.....</b>	<b>4</b>
<b>INTRODUCCIÓN Y JUSTIFICACIÓN.....</b>	<b>4</b>
<b>MARCO TEORICO .....</b>	<b>6</b>
FUENTE DE ALIMENTACION .....	6
SEMICONDUCTORES.....	6
ENLACE COVALENTE .....	7
MATERIALES TIPO P Y N.....	8
DIODO SEMICONDUCTOR.....	10
PUENTE H L293D.....	13
MODULACION POR ANCHO DE PULSO.....	14
CODIGO QR.....	15
CREACION DE INTERFACE.....	16
<b>DIAGRAMA DE FLUJO DEL SISTEMA.....</b>	<b>17</b>
<b>DIAGRAMA ELECTRONICO.....</b>	<b>18</b>
<b>ARMADO DE BRAZO ROBOTICO.....</b>	<b>19</b>
<b>ARMADO BANDA TRANSPORTADORA.....</b>	<b>31</b>
<b>DESARROLLO DE LOS MÓDULOS:.....</b>	<b>34</b>
JUSTIFICACIÓN ELECTRÓNICA ANALÓGICA .....	34
DESARROLLO ELECTRÓNICA ANALÓGICA .....	34
JUSTIFICACIÓN ELECTRÓNICA DIGITAL .....	35
DESARROLLO ELECTRÓNICA DIGITAL .....	35
MODULO COMUNICACIONES.....	35
DESARROLLO COMUNICACIONES .....	35
MODULO INSTRUMENTACIÓN Y CONTROL .....	37
DESARROLLO INSTRUMENTACIÓN Y CONTROL.....	37
<b>SOFTWARE .....</b>	<b>37</b>
CODIGO C++ ESP32.....	37
CODIGO QT CREATOR.....	49
CODIGO ESCANER QR QML .....	54
<b>HARDWARE.....</b>	<b>60</b>
CARACTERISTICAS ELECTRICAS DE LOS COMPONENTES.....	60
ESP 32.....	60
SERVOMOTOR MG996R .....	61
MOTOR REDUCTOR .....	61
TRANSFORMADOR .....	61
ENCAPSULADO DIP PUENTE H L293D.....	62
TRANSISTOR PNP TIP 42 C .....	63
DIODO 6A8 .....	64
REGULADOR DE VOLTAJE CV7805 Y CV7806.....	65
IMPLEMENTACION .....	65
<b>CONCLUSIONES .....</b>	<b>66</b>
<b>TRABAJOS FUTUROS .....</b>	<b>66</b>
<b>BIBLIOGRAFÍA .....</b>	<b>67</b>
<b>PÁGINAS WEB CONSULTADAS .....</b>	<b>67</b>

## MISION

Brindar una clasificación automatizada de paquetes para la compra y venta en internet y ofrecer una mayor velocidad de entrega para las empresas de logística.

## VISION

Proveer de servicios automatizados con visión e inteligencia artificial para empresas de paquetería, que permita una mejora en la velocidad y el control de entrega.

## DESCRIPCIÓN DEL PROYECTO

El sistema *Brazo robótico clasificador de paquetes* es un sistema destinado a la clasificación de paquetes y optimización de la entrega de paquetería mediante la automatización e inclusión de nuevas tecnologías en la logística y distribución de dichos paquetes. Teniendo como entrada al sistema los paquetes que no fueron entregados en una jornada laboral, mediante el reconocimiento de códigos QR a lo largo de una cinta transportadora el sistema escanea el paquete en turno y se encargara de ordenar, clasificar y ponderar este para su entrega el siguiente día. Con esto se pretende optimizar física y computacionalmente su salida con respecto a los tiempos de entrega.

Suponiendo un escenario en el cual, al término de una jornada laboral de un transportista, ciertos paquetes que no pudieron ser entregados. Entonces al descargar en almacén, el transportista solo deberá poner los paquetes en una banda transportadora donde se escanearán los códigos QR con información ligada a una página específica del paquete en cuestión y el sistema “*Brazo robótico clasificador de paquetes*” se encargará de dar las comandos y señales adecuadas a un brazo robótico que ordenará dichos paquetes por día de entrega estimada, es decir si el escaner QR detecta que el paquete es del lunes, se debe de dar la orden al brazo robótico de colocarlo en el día lunes, si es martes en el día martes y así para todos los días de la semana. El sistema dará prioridad de entrega a los paquetes que tengan más días sin poder ser entregados facilitando así el trabajo de la siguiente jornada. Esta aplicación busca darle mayor agilidad a la entrega de paquetes que tengan más días con entregas fallidas.

## OBJETIVO GENERAL

El objetivo principal del proyecto es el diseño e implementación de un sistema completo de reconocimiento, clasificación y optimización en la entrega de paquetes haciendo uso de la tecnología de código QR, un algoritmo de ordenamiento, una banda transportadora, un brazo robótico, con sus respectivas etapas de electrónica analógica, digital, comunicaciones instrumentación y control.

## OBJETIVOS ESPECIFICOS

- implementación del módulo de electrónica analógica
- implementación del módulo de electrónica digital
- implementación del módulo de comunicaciones
- implementación del sistema de instrumentación y control

## INTRODUCCIÓN Y JUSTIFICACIÓN

La automatización industrial mediante el uso de sistemas computarizados, electromecánicos, electroneumáticos y electrohidráulicos tiene ya mucho tiempo en desarrollo, pero es a partir del siglo XIX con la revolución industrial donde la sistematización toma impulso. Las ventajas más citadas de la automatización son repetitividad, un control de calidad más estrecho, una mayor eficiencia, mayor coordinación sistemas administrativos empresariales, un incremento de productividad, y reducción de trabajo humano y accidentes. Esto es, a través de los procesos automatizados se busca maximizar el

nivel de producción, mantener y superar los estándares de calidad y asegurar la integridad de las personas.

Empresas como HONEYWELL, ROCKWELL, EPSON, HP, FESTO, SIEMENS, CONTINENTAL, etc. entre otras han puesto sus objetivos e inversiones en proyectos que van encaminados a prestar servicios totalmente automatizados.

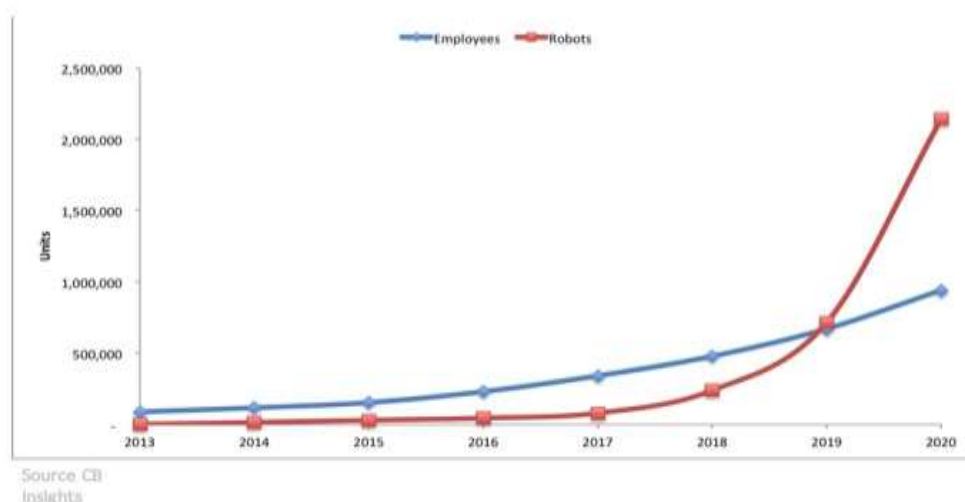


Las 10 principales compañías de automatización en el mundo  
( <https://forumautomation.com/t/top-10-automation-companies-in-the-world/9894>)

La robótica ha sido pieza clave en el desarrollo moderno de la automatización electrónica, ejemplos dentro de este rubro son la multitud de brazos robóticos utilizados en líneas de ensamblaje y producción de las industrias.

Por otra parte, en épocas recientes se ha visto un incremento del comercio en línea debido en parte a efectos de la pandemia del Covid-19, pero además debido también a la tendencia en crecimiento mundial de la virtualización de mercados. Mas aun, esta industria de comercio en línea se ha ido automatizando cada vez más llegando a casos de plantas industriales de distribución totalmente automatizadas. El ejemplo icónico en este rubro es la empresa AMAZON que maneja un promedio de 608 millones de paquetes cada año lo que equivale a un estimado de 1,600,000 de paquetes diariamente de manera automatizada o semi automatizada, además brinda servicios muy diversos desde el llamado hosting para empresas o servicios de nube. Con lo anterior se vuelve cada vez más y más necesario tanto el personal capaz de crear de este tipo de tecnologías como el desarrollo de base y patente para la evolución y aplicación en la industria tanto a nivel local como internacional.

## Amazon Robots



Inversionistas cada vez se convencer de que invertir en este tipo de desarrollos reditúa a corto plazo y que además aporta competitividad a la empresa que los implementa

(<https://www.computerweekly.com/es/cronica/Por-que-invertir-en-procesos-de-automatizacion>).

Un sistema tan complejo como el descrito tiene forzosamente módulos que trabajan en conjunto para llegar a los objetivos planteados. La solución aquí propuesta se plantea ser uno de estos módulos que contribuirán a el mejor desempeño del sistema de entregas.

## MARCO TEORICO

### FUENTE DE ALIMENTACION

Para realizar la fuente de alimentación es importante tener una base de conocimientos básicos esenciales. Lo principal es conocer los semiconductores y saber que la energía eléctrica que se proporciona en casa habitación y en la industria es una energía llamada “Corriente Alterna” y la mayoría de los dispositivos de uso cotidiano funcionan con energía llamada “Corriente continua”.

*A partir de aquí se abreviará CA como corriente alterna y CC para corriente continua.*

La CA tiene características importantes a tomar en cuenta, el sistema más básico de CA se llama monofásico, un sistema monofásico se compone de dos hilos conductores (*entiéndase por hilo conductor como cable, alambre, conductor de energía, etc*) 1 de los 2 hilos conductores es un medio de transporte para la energía llamada “fase” y el otro hilo conductor está destinado para cerrar el circuito con el llamado “neutro”. La fase tiene por lo regular una forma senoidal con un voltaje de 127V y una frecuencia de 60Hertz, los Hertz indican cuantas veces cambia de polaridad la fase, es decir cuántos ciclos realiza por segundo nos permite saber cuántas veces cambia de positivo a negativo, de allí su nombre “corriente alterna”. La señal de la fase tiene características más específicas como: voltaje pico a pico, cruce por cero, voltaje eficaz, voltaje promedio, periodo, amplitud, dominio en frecuencia, dominio en el tiempo, identidades trigonométricas, análisis en números reales y complejos. Existen más sistemas semejantes al monofásico únicamente se incrementan el número de fases que proporciona el generador de energía, llamándose bifásico los sistemas de dos fases, trifásico los sistemas con tres fases por mencionar algunos, la diferencia con el sistema monofásico es que se puede incrementar el voltaje y corriente máximo que se puede suministrar, si se trabaja con un sistema de bifásico o trifásico entre línea y línea se puede tener un voltaje aproximado de 220Volts. Normalmente la energía que se suministra en las calles de casa habitación son sistemas trifásicos y para proporcionar energía a los hogares únicamente se toma de 1 línea de las 3 líneas y se toma el hilo neutro, así podemos proporcionar únicamente el volteje de una fase 127V 60Hz.

Y por qué no suministrar energía de CC a los hogares?

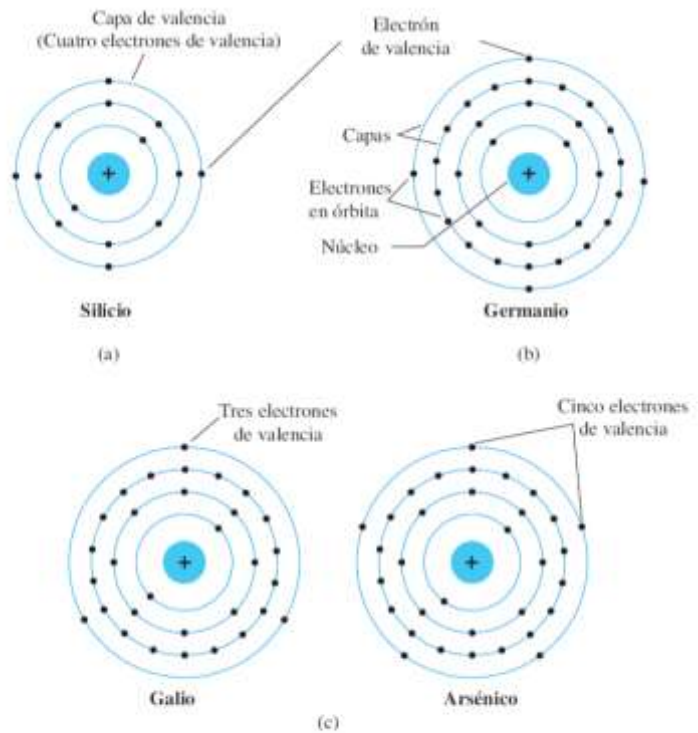
Por el simple hecho de que la CA es mucho más fácil de generar y transportar, las pérdidas de potencia son muchísimo menores a la perdida por transporte por CC, existe una gran historia referente a la batalla de CA y CC, entre Nikola Tesla y Thomas Edison.

### SEMICONDUCTORES

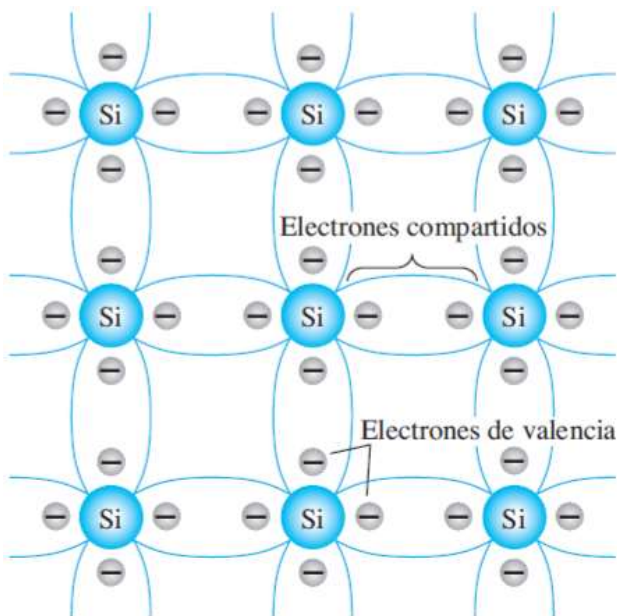
Para transformar la energía de CA a CC es importante conocer los semiconductores, un semiconductor es un componente electrónico que permite tener dos posibles configuraciones una donde el dispositivo puede conducir energía eléctrica y se comporta como conductor y otra donde no se puede conducir energía eléctrica y se comporta como aislante, de allí su nombre “semiconductor”. Dicho esto, cualquier dispositivo electrónico discreto de estado sólido está construido por un material semiconductor. Los semiconductores se clasifican dos clases, de un solo cristal como el germanio (Ge), silicio (Si) que tienen una estructura cristalina repetitiva y la clase de semiconductor compuesto como el Arseniuro de galio (GaAs). El sulfuro de cadmio (CdS), el fosfuro de galio y arsénico (GaAsP) y el nitrato de galio (GaN) son compuestos de 2 o más materiales semiconductores de varias estructuras atómicas. Los 3 tipos de semiconductor más usados en la industria electrónica son Arseniuro de galio (GaAs) Germanio (Ge) y Silicio (Si). Los primeros dispositivos electrónicos diseñados fueron el diodo en 1939 y el transistor en 1949 eran construidos con Germanio una característica del germanio es que son muy susceptibles a los cambios de temperatura, en 1954 se presentó el primer transistor de silicio. Una característica del transistor de silicio era que no podía operar a grandes frecuencias, fue en 1970 que se presentó el primer transistor de GaAs que era capaz de operar a velocidades de hasta 5 veces más rápido que los de silicio.

## ENLACE COVALENTE

Todo átomo está compuesto de tres partículas básicas: neutrón, protón y electrón. Los neutrones y los protones forman el núcleo, los electrones aparecen en las orbitas fijas alrededor del núcleo. Este modelo se conoce como “modelo de Bohr”. Como se muestra en la figura el germanio tiene 32 electrones en su órbita, silicio tiene 14 electrones en su órbita, el arsénico 33 y el galio 31. El germanio y el silicio tienen 4 electrones en su última orbita, llamándose electrones de valencia. El arsénico tiene 5 electrones de valencia y el galio 3. Los átomos que tienen 4 electrones de valencia se llaman tetravalentes, los que tienen 5 se llaman pentavalentes y los que tienen 3 se llaman trivalentes. El termino valencia es utilizado para indicar que el potencial requerido para mover cualquiera de estos electrones de la estructura atómica es significativamente más bajo que el requerido para cualquier otro electrón en la estructura.



**FIG. 1.3**  
Estructura atómica del (a) silicio; (b) germanio, y (c) galio y arsénico.



**FIG. 1.4**

*Enlace covalente del átomo de silicio.*

En la figura de la izquierda, en un cristal de silicio o germanio puros, los 4 electrones de valencia de un átomo formado por un arreglo de enlace con 4 átomos adyacentes. Este enlace de átomos, reforzado por compartir electrones se llama enlace covalente. El enlace covalente produce un enlace más fuerte entre los electrones de valencia y su átomo padre, aún es posible que los electrones de valencia absorban suficiente energía cinética proveniente de causas externas para romper el enlace covalente y asumir el estado “libre”.-

El termino libre es aplicado a cualquier electrón que se haya separado de la estructura entrelazada fija y es muy sensible a cualquier campo eléctrico aplicado como el establecido por fuentes de voltaje.

Ahora, se deben conocer dos términos muy importantes “intrínseco” y “extrínseco”.

El termino intrínseco se aplica a cualquier material semiconductor que haya sido refinado para reducir el número de impurezas a un nivel muy bajo, lo más puro posible que se pueda fabricar.

Los electrones libres presentes en un material debido a solo causas externas se llaman portadores “intrínsecos”. La capacidad de cambiar las características de un material mediante este proceso se conoce como “dopado”. Los materiales conductores tienen una resistencia de coeficiente positivo, esto quiere decir que a medida que el calor aumenta la resistencia aumenta. Los materiales semiconductores



presentan un incremento de conductividad conforme el calor aumenta, los materiales semiconductores tienen un coeficiente de temperatura negativo. Dentro de la estructura atómica existen niveles específicos de energía asociados a cada capa, cada capa es diferente dependiendo del material con el que se trabaja. En general, cuanto más alejado se encuentra un electrón de un núcleo, mayor es su estado de energía y cualquier electrón que haya abandonado a su átomo padre tiene un estado de energía mayor que todo electrón que permanezca en la estructura atómica. Los electrones de valencia de un material de silicio pueden tener diversos niveles de energía hay un nivel de energía mínimo asociado con electrones que se encuentran en la banda de conducción y un nivel de energía máximo de electrones enlazados a la capa de valencia del átomo. Entre los dos hay una brecha de energía que el electrón en la banda de valencia debe salvar para convertirse en portador libre. Esa brecha de energía es diferente para Ge, Si y GaAs; el Ge tiene la brecha mínima y el GaAs la máxima.

Un electrón en la banda de valencia de silicio debe absorber más energía que uno en la banda de valencia de germanio para convertirse en portador libre. Asimismo, un electrón en la banda de valencia de arseniuro de galio debe absorber más energía que uno en la de silicio o germanio para entrar a la banda de conducción.

Esta diferencia en los requerimientos de las brechas de energía revela la sensibilidad de cada tipo de semiconductor a los cambios de temperatura, al elevarse la temperatura de una muestra de Ge, el número de electrones que pueden absorber energía térmica y entrar a la banda de conducción se incrementa con rapidez porque la brecha de energía es mínima. Sin embargo, el número de electrones que entran a la banda de conducción en Si o GaAs es mucho menor.

Esta sensibilidad a los cambios de nivel de energía puede tener efectos positivos y negativos. El diseño de fotodetectores sensibles a la luz y los sistemas de seguridad sensibles al calor, parecen ser una excelente área de aplicación de los dispositivos de Ge. No obstante, en el caso de redes de transistores, en las que la estabilidad es de alta prioridad, esta sensibilidad a la temperatura o a la luz puede ser un factor perjudicial

### MATERIALES TIPO P Y N

El silicio es el material más utilizado como material base (sustrato) en la construcción de dispositivos de estado sólido las características de un material semiconductor se pueden modificar de manera significativa con la adición de átomos de impureza específicos al material semiconductor relativamente puro. Estas impurezas, aunque sólo se agregan en 1 parte en 10 millones, pueden alterar la estructura de las bandas lo suficiente para cambiar del todo las propiedades eléctricas del material. Un material semiconductor que ha sido sometido al proceso de dopado se conoce como material extrínseco.

Hay dos materiales extrínsecos de inmensurable importancia en la fabricación de dispositivos semiconductores: materiales tipo n y tipo p. Cada uno se describe con algún detalle en las siguientes subsecciones.

### MATERIAL TIPO P

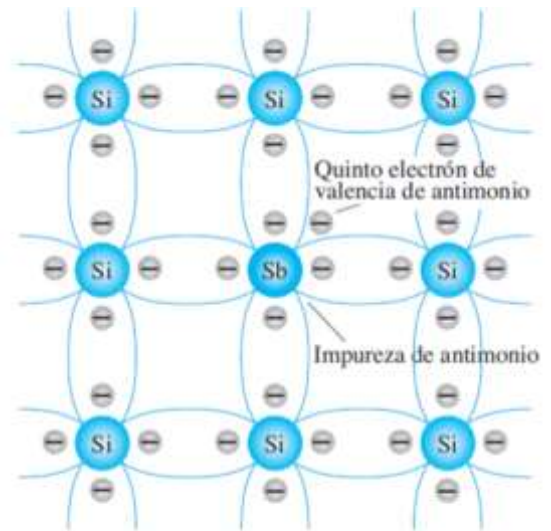
Tanto los materiales tipo n como los de tipo p se forman agregando un número predeterminado de átomos de impureza a una base de silicio. Un material tipo n se crea introduciendo elementos de impureza pentavalentes, como el antimonio, el arsénico y el fósforo como la impureza en una base de silicio. En la figura 1.7 puede Observar que los cuatros enlaces covalentes permanecen.

Y contienen un quinto electrón adicional debido al átomo de impureza, el cual no está asociado con cualquier enlace covalente particular. Este electrón restante, enlazado de manera poco firme a su átomo padre (antimonio), está en cierto modo libre para moverse dentro del material tipo n recién formado, puesto que el átomo de impureza insertado ha donado un electrón



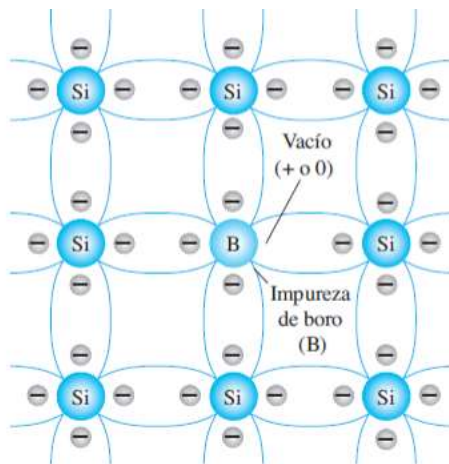
relativamente “libre” a la estructura. Las impurezas difundidas con cinco electrones de valencia se conocen como átomos donadores. Es importante tener en cuenta que aun cuando un gran número de portadores libres se ha establecido en el material tipo n, sigue siendo eléctricamente neutro puesto que de manera ideal el número de protones de carga positiva en los núcleos sigue siendo igual al de los electrones de carga negativa libres y en órbita en la estructura.

Los electrones libres creados por la impureza agregada se establecen en este nivel de energía y absorben con menos dificultad una cantidad suficiente de energía térmica para moverse en la banda de conducción a temperatura ambiente. El resultado es que, a temperatura ambiente, hay un gran número de portadores (electrones) en el nivel de conducción y la conductividad del material se incrementa de manera significativa.



**FIG. 1.7**

*Impureza de antimonio en un material tipo n.*



**FIG. 1.9**

*Impureza de boro en un material tipo n.*

### MATERIAL TIPO N

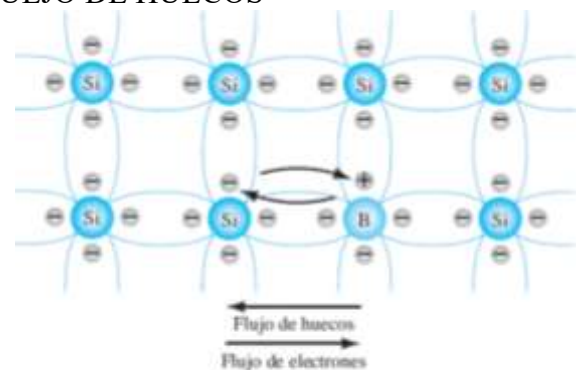
El material tipo p se forma dopando un cristal de germanio o silicio puro con átomos de impureza que tienen tres electrones de valencia. Los elementos más utilizados para este propósito son boro, galio e indio en la figura 1.9 que ahora el número de electrones es insuficiente para completar las bandas covalentes de la estructura recién formada. El vacío resultante se llama hueco y se denota con un pequeño círculo o un signo más, para indicar la ausencia de una carga positiva. Por lo tanto, el vacío resultante aceptará con facilidad un electrón libre: Las impurezas difundidas con tres electrones de valencia se llaman átomos aceptores.

El material tipo p es eléctricamente neutro por las mismas razones descritas para el material tipo n.

### FLUJO DE ELECTRONES VS FLUEJO DE HUECOS

El efecto del hueco en la conducción se aprecia en la figura 1.10. Si un electrón de valencia adquiere suficiente energía cinética para romper su enlace covalente y llenar el vacío creado por un hueco, entonces se creará un vacío o hueco en la banda covalente que cedió el electrón. En consecuencia, existe una transferencia de huecos hacia la izquierda y de electrones hacia la derecha. La dirección es la del flujo convencional, la cual está indicada por la dirección del flujo de huecos.

En el estado intrínseco, el número de electrones libres en Ge o Si se debe sólo a los electrones en la banda de valencia que adquirieron suficiente energía de fuentes térmicas o luminosas para romper la banda covalente o a las impurezas que no pudieron ser eliminadas. Los vacíos que quedan en la estructura de enlace covalente representan una fuente muy limitada de huecos. En un material tipo n, el número de



**FIG. 1.10**

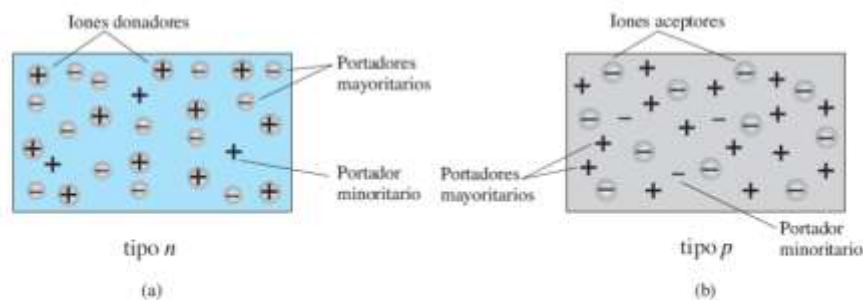
*Flujo de electrones contra flujo de huecos.*

huecos no cambia significativamente con respecto a este nivel intrínseco. El resultado neto, por consiguiente, es que el número de electrones sobrepasa por mucho al de huecos. Por eso:

En un material tipo n (Fig. 1.11a) el electrón se llama portador mayoritario y el hueco portador minoritario.

### PORTADORES MAYORITARIOS Y MINORITARIOS

En el estado intrínseco, el número de electrones libres en Ge o Si se debe sólo a los electrones en la banda de valencia que adquirieron suficiente energía de fuentes térmicas o luminosas para romper la banda covalente o a las impurezas que no pudieron ser eliminadas. Los vacíos que quedan en la estructura de enlace covalente representan una fuente muy limitada de huecos. En un material tipo n, el número de huecos no cambia significativamente con respecto a este nivel intrínseco. El resultado es que el número de electrones sobrepasa por mucho al de huecos. En un material tipo p el electrón se llama portador mayoritario y el hueco portador minoritario.



**FIG. 1.11**

(a) material tipo n; (b) material tipo p.

En el material tipo p el número de huecos excede por mucho al de electrones, en un material tipo p, el hueco es el portador mayoritario y el electrón el minoritario.

Cuando el quinto electrón de un átomo donador abandona el átomo padre, el átomo que queda adquiere una carga positiva neta: de ahí el signo más en la representación de ión donador. Por las mismas razones, el signo menos aparece en el ión aceptor.

Los materiales tipo n y p representan los bloques de construcción básicos de los dispositivos semiconductores. la “unión” de un material tipo n con uno tipo p producirá un elemento semiconductor de considerable importancia en sistemas electrónicos.

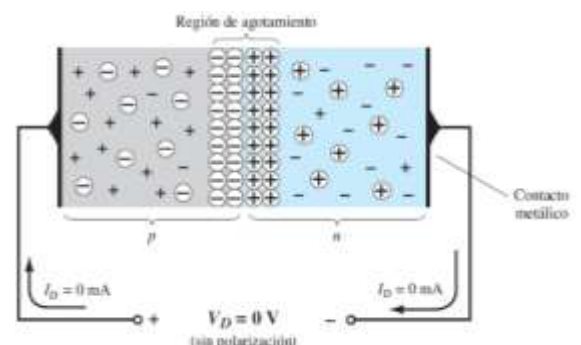
### DIODO SEMICONDUCTOR

El diodo semiconductor se crea uniendo un material tipo n a un material tipo p es la unión de un material con un portador mayoritario de electrones a uno con un portador mayoritario de huecos. La simplicidad básica de su construcción refuerza la importancia del desarrollo de esta área de estado sólido.

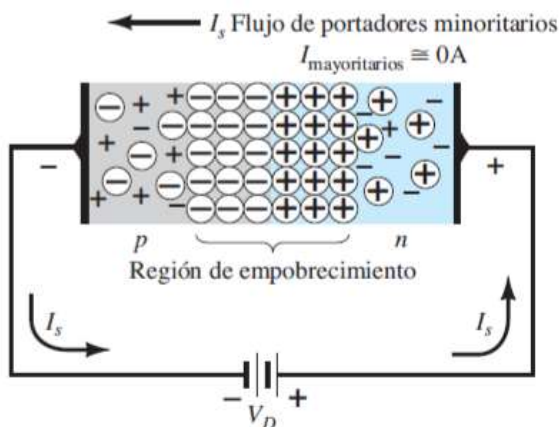
El diodo semiconductor cuenta con varias configuraciones, se llaman: sin polarización, polarización inversa y polarización directa.

### SIN POLARIZACION

Es el momento en que los dos materiales se “unen”, los electrones y los huecos en la región de la unión se combinan y provocan una carencia de portadores libres en la región próxima a la unión, las únicas partículas mostradas en esta región son los iones positivos y negativos que quedan una vez que los portadores libres han sido absorbidos. Esta región de iones positivos y negativos revelados se llama región de “empobrecimiento”, debido a la disminución de portadores libres en la región.



## POLARIZACION EN INVERSA



Si se aplica un potencial externo de  $V$  volts a través de la unión p-n con la terminal positiva conectada al material tipo n y la negativa conectada al material tipo p como se muestra en la figura, el número de iones positivos revelados en la región de empobrecimiento del material tipo n se incrementará por la gran cantidad de electrones libres atraídos por el potencial positivo del voltaje aplicado.

Por las mismas razones, el número de iones negativos no revelados se incrementará en el material tipo p. El efecto es una mayor apertura de la región de empobrecimiento, la cual crea una barrera

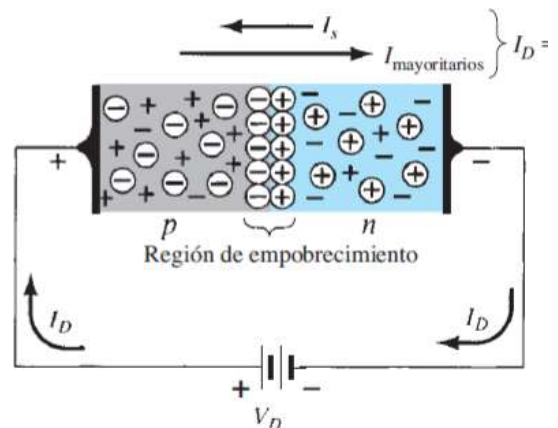
demasiado grande para que los portadores mayoritarios la puedan superar, por lo que el flujo de portadores mayoritarios se reduce efectivamente a cero, el número de portadores minoritarios que entran a la región de empobrecimiento no cambia, y se producen vectores de flujo de portadores minoritarios de la misma magnitud sin voltaje aplicado.

La corriente en condiciones de polarización en inversa se llama corriente de saturación en inversa y está representada por  $I_s$ . La corriente de saturación en inversa rara vez es de más de algunos microamperes, excepto en el caso de dispositivos de alta potencia. El término saturación se deriva del hecho de que alcanza su nivel máximo con rapidez y que no cambia de manera significativa con los incrementos en el potencial de polarización en inversa el lado negativo del voltaje aplicado está conectado al material tipo p y el lado positivo al material tipo n, y la diferencia indicada con las letras subrayadas por cada región revela una condición de polarización en inversa.

## POLARIZACION EN DIRECTA

La condición de polarización en directa se establece aplicando el potencial positivo al material tipo p y el potencial negativo al tipo n. La aplicación de un potencial de polarización en directa oprime a los electrones en el material tipo n y a los huecos en el material tipo p para que se recombinen con los iones próximos al límite y reducirá el ancho de la región de empobrecimiento el flujo de portadores minoritarios de electrones resultante del material tipo p al material tipo n no cambia de magnitud. Aunque la reducción del ancho de la región de empobrecimiento produjo un intenso flujo de portadores mayoritarios a través de la unión. Un electrón del material

tipo p ahora afronta una barrera reducida en la unión debido a la región de empobrecimiento reducida y a una fuerte atracción del potencial positivo aplicado al material tipo p. En cuanto se incrementa la magnitud de la polarización aplicada, el ancho de la región de empobrecimiento continuará reduciéndose hasta que un flujo de electrones pueda atravesar la unión, lo que produce un crecimiento exponencial de la corriente y la escala horizontal en la región de polarización en directa tiene un máximo de 1 V. Por consiguiente, en general el voltaje a través de un diodo polarizado en directa será menor de 1 V.



Contando con las bases de que es la CA, CC y los diodos semiconductores podemos comprender mas el funcionamiento de la fuente de voltaje y cada una de sus etapas.

En la siguiente imagen presento un diagrama a bloques de las etapas de la transformación de la corriente alterna a corriente continua.



Para empezar en la entrada tenemos un voltaje de corriente alterna a 127Volts a 60Hz ideales no siempre será el mismo voltaje, el voltaje varía dependiendo de la estabilidad de los generadores y de las cargas del sistema. Este voltaje de CA pasa por el transformador que cuenta con tres hilos, dos hilos que transportan el voltaje aun en CA y un hilo para derivación central (Tierra), entre estas tres terminales podemos proporcionar 24 Volts con 5 Amperes o 12Volts con 5 Amperes, para este proyecto se usara únicamente una de las terminales de voltaje y la terminal de derivación central, para proveer de energía con 12Volts 5 amperes al sistema, esta misma fuente tiene la capacidad de suministrar hasta 24Volts tomando las terminales de voltaje positivo y voltaje negativo, esto proporciona mayor amplitud a la señal de salida.

Una de las terminales de voltaje del transformador contiene la fase de la corriente alterna, pero reducida a 12 Volts 5 Amperes, esta señal pasara por un puente rectificador, el puente rectificador consta de 4 diodos conectados en antiparalelo para esta fuente se usa 4 diodos modelo 6A8. La configuración de este puente rectificador se encarga de recortar los ciclos negativos de la fase para lograr pasar únicamente la señal en los ciclos positivos, el arreglo de diodos permite una corriente de polarización directa máxima de 6 A y tienen una capacidad de voltaje en inversa de máximo 800Volts, como nosotros estamos usando solo 24 volts del transformador, el voltaje inverso máximo de ruptura nos garantiza que el puente de diodos no se verá afectado ni dañado, además el circuito cuenta con un fusible que brinda protección adicional a los componentes de cualquier corto circuito.

El puente rectificador nos proporciona a su salida un voltaje positivo y un voltaje negativo, si tomamos la señal de estos dos voltajes obtenemos 24 Volts ideales, para los requerimientos del proyecto se tomara únicamente el voltaje positivo y la terminal de derivación central del transformador, el voltaje positivo ya está rectificado y únicamente pasan los ciclos positivos de la CA ahora es necesario realizar un filtrado de señal para ello se van a utilizar capacitores que permitan mantener más constante el voltaje, los capacitores permiten almacenar la carga de la señal y mantener una CC aquí hay varias características a tomar en cuenta, cuanto mayor sea la capacidad del capacitor en faradios mayor será el voltaje que entregue el capacitor a su salida, pero tardara mas tiempo en cargarse y proveer un voltaje estable, además el capacitor cuenta con un voltaje máximo de operación, para este proyecto se utilizan capacitores de 4700mf a 50 V y 2200mf a 50V, no es recomendable exceder el 60% del voltaje máximo del capacitor para garantizar un mayor tiempo de vida útil al capacitor. Después de la etapa del filtrado y obtener un voltaje mas estable necesitamos regular el voltaje para energizar los componentes electrónicos, la señal ya medida de la fuente nos proporciona un voltaje de 18VCD sin carga. El voltaje de 18VCD pasara por dos reguladores de voltaje, el primero es un regulador modelo 7805CV que proporciona a su salida un voltaje de 5Volts a 1 Amper, este regulador alimenta al motor reductor de la banda transportadora y al puente H L293D para el control de giro del motor y suministro de potencia, ya que los pines del esp 32 no proporcionan suficiente energía a su salida. El otro regulador es un 7806CV proporciona un voltaje de salida de 6 Volts a 1 Amper, este regulador será el encargado de suministrar el voltaje regulado que alimente a los servomotores MG996R, pero como el 7806CV solo proporciona 1



Amper a su salida es necesario utilizar un transistor de paso de potencia que permita suministrar corriente suficiente a los motores, para el proyecto vamos a utilizar un transistor PNP TIP42C es capaz de suministrar hasta 6 Amperes a su salida. Ambos reguladores de voltaje cuentan con un arreglo de capacitores que reduzca señales de ruido y frecuencias parasito del sistema.

### PUENTE H L293D

El puente H L293D es un CI que se compone por cuatro canales monolíticos integrados de alto voltaje y alta corriente diseñado para aceptar niveles lógicos TTL estándar y controlar cargas inductivas.

El puente H L293D puede controlar hasta dos motores a la vez para este proyecto solo usaremos un puente H, para activar el puente H de nuestro motor que será el motor 1 se debe activar el pin ENABLE 1 del puente H e ingresar en las terminales de control los niveles de tensión para controlar el sentido de giro de motor, estos pines funcionan como una compuerta XOR donde si los dos pines de control de giro son iguales las salidas del motor serán 0 y el motor no girara, si las salidas son diferentes es decir 1 y 0 el motor girara en un sentido y si los pines de control son 0 y 1 el motor girara en el sentido opuesto. El puente H puede considerarse además como control de giro, como una protección para las corrientes inductivas parasitas que genere el motor cuando está girando en un sentido u otro.

MOTOR	S1	S2	S3	S4
LIBRE	ABIERTO	ABIERTO	ABIERTO	ABIERTO
GIRA A LA DERECHA	ABIERTO	CERRADO	ABIERTO	CERRADO
FRENADO	ABIERTO	ABIERTO	CERRADO	CERRADO
GIRA A LA IZQUIERDA	CERRADO	ABIERTO	CERRADO	ABIERTO
FRENADO	CERRADO	CERRADO	ABIERTO	ABIERTO
NO VALIDO	CERRADO	X	CERRADO	X
NO VALIDO	X	CERRADO	X	CERRADO

### ESPECIFICACIONES TÉCNICAS Y CARACTERÍSTICAS

L293D DIP 16-pin IC Stepper Motor Drive Controller

Capacidad del canal: corriente / canal de salida de 600 mA

Corriente de salida máxima: 1.2 A / canal (no repetitiva)

Habilitar instalación

Protección contra sobre temperatura

Voltaje lógico / 0 / entrada de hasta 1.5V (alta inmunidad al ruido)

Diodos de pinza internos

Peso: 1.5 g

Vs Tensión de alimentación maximo: 36V

Vss Voltaje de alimentación lógico maximo: 36V

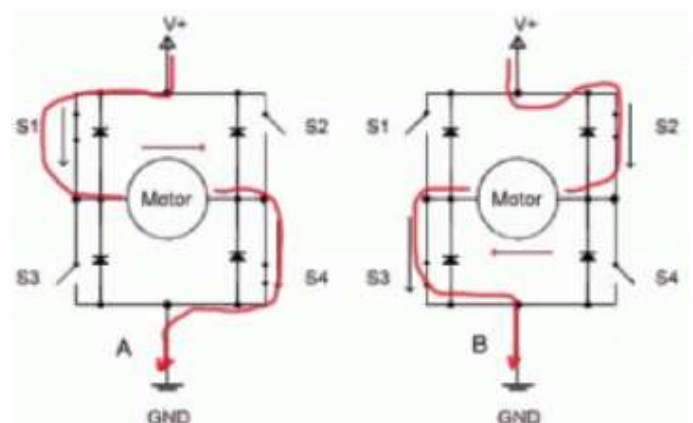
Vi voltaje de las señales de entrada: 5V

Voltaje maximo de habilitación Venable: 5V

Corriente de salida máxima de Io: 1.2<sup>a</sup>

Potenciatotal de Disipación 4 Watts

Rangos de temperatura de operacion: -40 gradosC a 150 gradosC



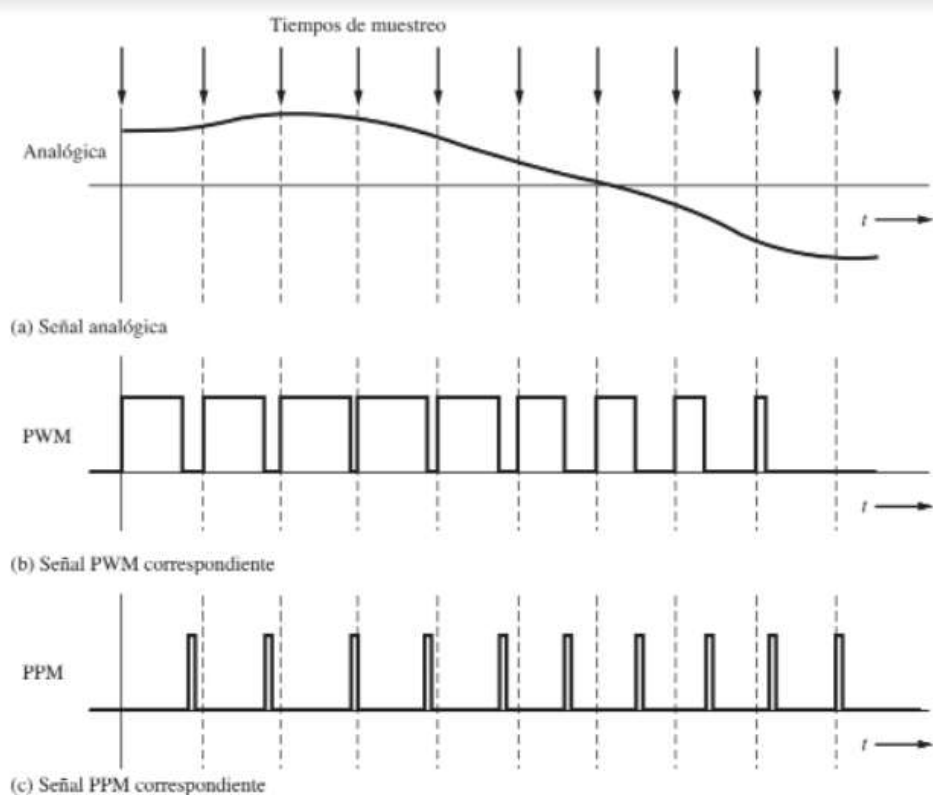
## MODULACION POR ANCHO DE PULSO

La modulación por tiempo de pulsos es una técnica de señalización que codifica las muestras de una señal analógica dentro del eje de tiempo de una digital. La PTM es semejante a la modulación de ángulo. Como las técnicas PAM, PCM y DM codifican las muestras dentro de las características de amplitud de la señal digital.

Los dos tipos principales de PTM son la modulación por ancho de pulso (PWM, por sus siglas en inglés) y la modulación por posición de pulso (PPM, por sus siglas en inglés).

En PWM las muestras de la forma de onda analógica se utilizan para determinar el ancho de la señal de pulso. Se puede emplear tanto el muestreo instantáneo como el natural. La siguiente imagen presenta una técnica para la generación de señales PWM con muestreo instantáneo y muestra la PWM con muestreo natural. En PPM, la muestra analógica determina la posición de un pulso estrecho relativa al tiempo de sincronización. Las técnicas para generar PPM se muestran también en la imagen y se puede observar que ésta se obtiene fácilmente a partir de la PWM mediante un circuito multivibrador monoestable. En la literatura acerca de las señales PTM, el nivel del comparador  $V_r$  se conoce como nivel de corte.

En resumen, la modulación por ancho de pulso es una técnica utilizada para transmitir señales analógicas donde la señal portadora es digital, con esta modulación se modifica el ciclo de trabajo de una señal periódica, para este proyecto se usará la modulación por ancho de pulso para controlar la cantidad de energía que será enviada a la carga en este caso los servomotres MG996R. una de las características de la modulación por ancho de pulso, es el duty cycle (ciclo de trabajo). El duty cycle de una señal periódica es el ancho del pulso de su parte positiva, en relación con el periodo de la misma señal. El duty ciclo se mide en porcentaje. Dependiendo el duty cycle será la posición de los servomotores MG996R.



La señalización por modulación por tiempo de pulso no se utiliza ampliamente para la comunicación a través de canales ya que se requiere de un ancho de banda del canal relativamente grande, especialmente para PPM.

## CODIGO QR

Un código QR es en esencia un código de barras más complejo. Un clásico código de barras permite almacenar hasta 20 caracteres alfanuméricos de forma horizontal. El problema es que no sirve para los caracteres kanji o kana que se usan en Japón. La información no cabe.

Los kanji son los sinogramas utilizados en la escritura del idioma japonés, es decir, los caracteres tradicionales, mientras que los kana, son los dos silabarios del idioma japonés (para los términos o nombres de origen extranjero que no poseen representaciones en kanji). Y es que en un idioma que se lee en vertical y posee tres escrituras (kanji, kana y romaji, es decir, el abecedario latino), el código de barras clásico se quedaba muy corto.

El código QR fue inventado por un ingeniero de Denso, Hara Masahiro, hace 25 años. Según explica Hara en nippon.com, Denso usaba códigos de barras para realizar un seguimiento de las piezas de coches que se enviaban, sobre todo a Toyota, pero el sistema era ineficiente. "Había más de diez códigos de barras en una caja", relata Hara.

"Los empleados se cansaron de tener que escanear varias veces una misma caja, y esto nos llevó a buscar un código que permitiera transmitir un gran volumen de información con una sola pasada de escaner". De la necesidad de hacer un mejor seguimiento de las piezas de automóvil surgió el código QR.

Los códigos QR pueden contener un gran volumen de información, desde una simple url hasta un amplio volumen de información compuesto por hasta 7.000 caracteres alfanuméricos, incluyendo los kanji y los kana. Además, se puede escanear boca abajo, por ejemplo, ya que el QR incluye tres marcadores de posición, y se puede leer incluso si está dañado o manchado, gracias a una función de corrección de errores.

El código QR se inspira en el tablero del go

A Hara, la inspiración le vino jugando al go. A la hora de comer, Hara solía jugar al go y un día al ordenar de nuevo las piezas blancas y negras se le ocurrió que sería una manera muy directa de presentar la información. "Fue un momento eureka", recuerda.

Si bien Hara es reconocido como el inventor por su inspiración, fue el equipo de desarrollo de Denso el responsable de crear los códigos que no conocemos hoy. "Tener una idea es una cosa", afirma Hara, "pero también se necesita un sistema que admita su uso". Denso carecía de los recursos para desarrollar la tecnología por sí mismo, y en su lugar decidió liberar las patentes con la esperanza de que otras compañías usaran códigos QR.

Y funcionó. Pronto, otras compañías empezaron a usar los códigos QR. La que le dio el primer gran empujón fue Toyota. En el gigante japonés están obsesionados con no desperdiciar ni tiempo ni recursos. Esto ha llevado a lo que se conoce en la industria como el "just-in-time" y que todos los fabricantes han adoptado.

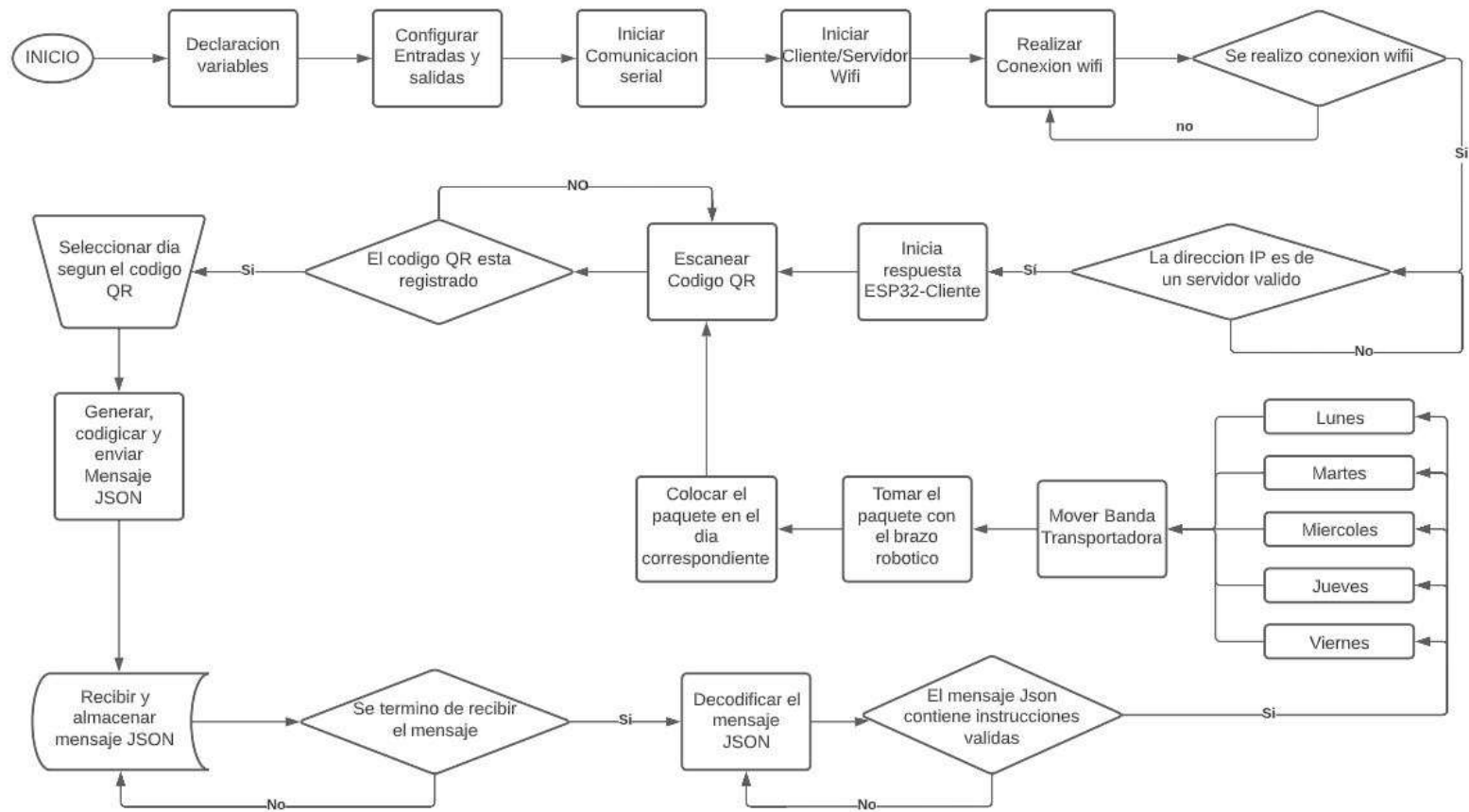
Es un sistema en el que la fábrica no tiene stock de recambios ni piezas (en todo caso, no más allá de 24 horas como máximo en algunos casos puntuales). Es un proceso que requiere una gran coordinación entre los proveedores y la gestión de stock, así como una gran precisión a la hora de saber dónde están las piezas necesarias, qué cantidad hay, para qué modelo son, cuanto tiempo van a tardar en llegar, etc. En el caso de Toyota, el stock se reduce a su más mínima expresión. Y más aún. Cuentan en el libro *The Machine That Changed the World*, que los autores pasaron un cuestionario a la dirección de Toyota en el que, entre otras cosas, se preguntaba para cuántos días de stock tenían. Uno de los directivos preguntó humildemente si había un error de traducción. "Seguramente, habrán querido decir para cuántos minutos de inventario", explicó el directivo.

El código QR no es el único código de barras que puede almacenar mucha información, sin embargo, es el que más se utiliza en el mundo. Su habilidad para corregir por sí mismo los errores de lectura y los tres marcadores de posición lo hacen muy versátil, pero al mismo tiempo fue la decisión de Denso de compartir las patentes con el mundo para que fuese adoptado masivamente lo que permitió su éxito.



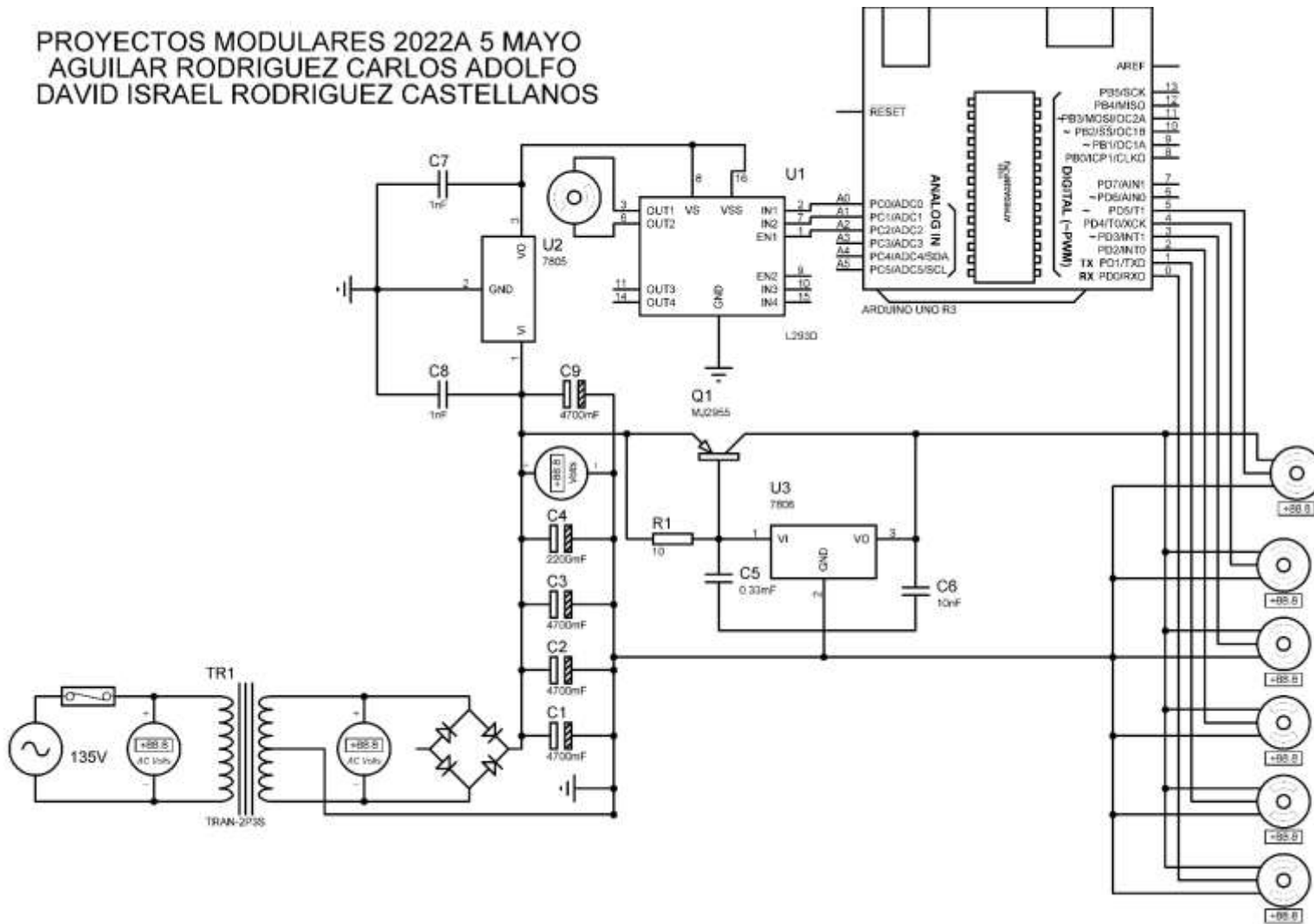
## CREACION DE INTERFACE

## DIAGRAMA DE FLUJO DEL SISTEMA



## DIAGRAMA ELECTRONICO

PROYECTOS MODULARES 2022A 5 MAYO  
AGUILAR RODRIGUEZ CARLOS ADOLFO  
DAVID ISRAEL RODRIGUEZ CASTELLANOS



## ARMADO DE BRAZO ROBOTICO

Primero juntamos la herramienta y las piezas necesarias para el armado del brazo robótico, se requiere un desarmador de varias puntas, hexagonales cruz y plano, pinzas, tornillerías y servomotores.



El primer paso es identificar las dos piezas de la base del brazo y unir las con tornillos, son las siguientes:



Deben quedar así:



Ahora tomamos una tercera pieza parecida a las de la base y la conectamos perpendicular a las otras bases  
De esta manera



El siguiente paso es identificar las siguientes dos piezas para conectar el primer servomotor a la base del brazo.



Deben de quedar de la siguiente manera



Ahora identificamos la siguientes piezas:



Debe quedar de la siguiente forma:



Ya teniendo esas dos piezas armadas se deben de ver asi





Ahora identificamos la base del brazo y el clip donde ira el motor encargado del giro del primer eje del brazo



Debemos de unirlo con tornillos a la base de la siguiente manera.





El siguiente paso es incorporar el primer servomotor MG996R en el disco metálico que se incorporo en el paso anterior



Debe quedar fijo en la base q del primer eje que rotara 180° en su mismo eje



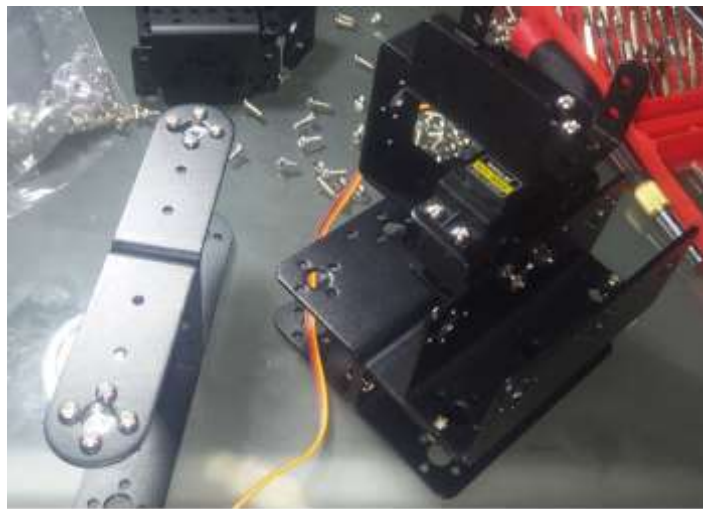
Ahora debemos identificar estas dos piezas en forma de U para unir las y agregar un disco metálico de los servomotores en cada uno de los extremos.



Deben quedar de la siguiente manera, por dentro de las piezas en forma de U de el mismo lado ambos discos



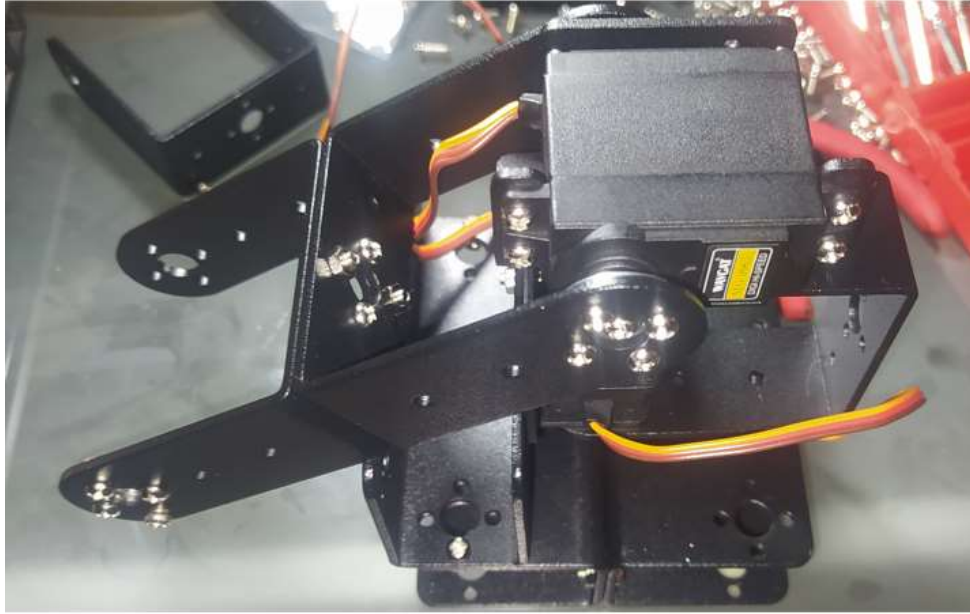
Estas dos piezas deben unirse con la base para colocar los siguientes dos motores



Deben quedar de esta forma



Ahora podemos colocar el segundo servomotor, debemos de calibrar los límites del giro del motor para poder usar todo el ángulo permitido por las piezas del brazo esto lo podemos realizar con las manos cuidadosamente para no dañar los engranes del servomotor, cuando este calibrado los movimientos del motor con los del brazo podemos colocar los tornillos de los discos metálicos que van hacia el servomotor, se sugiere apretar con fuerza moderada para no barrer la rosca del eje del servomotor pero lo suficiente para que no se caigan los tornillos con las vibraciones,



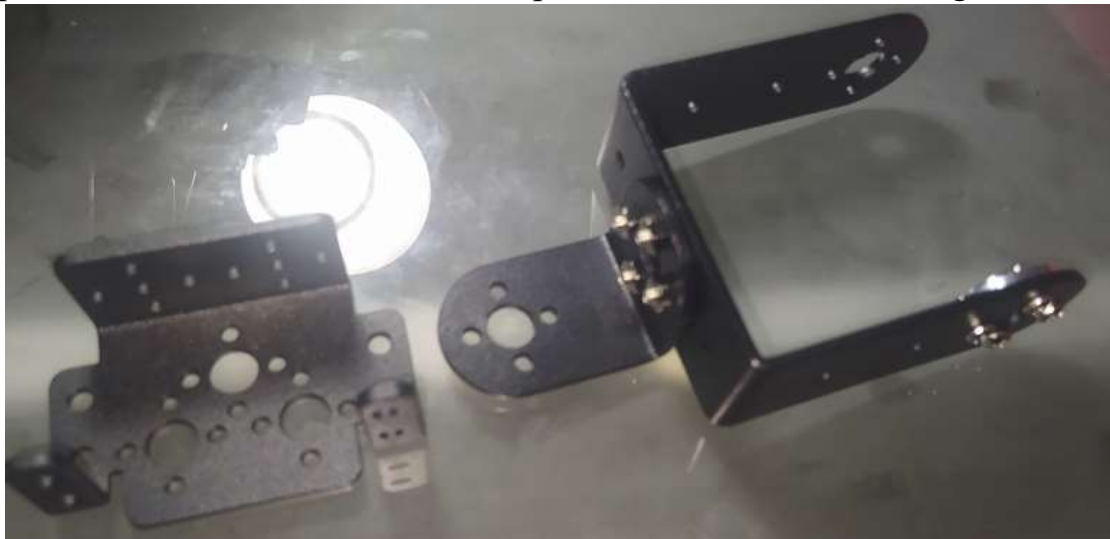
Identificamos las siguientes piezas para comenzar el armado de la pinza



De la misma manera que las otras piezas en forma de U, se coloca el disco metálico por dentro de la siguiente forma



La pieza en forma de L se coloca en la pieza en forma de U de la siguiente manera.



La pieza restante se debe colocar como se muestra en la siguiente imagen.





La pieza recién armada se va a unir a la estructura del brazo.



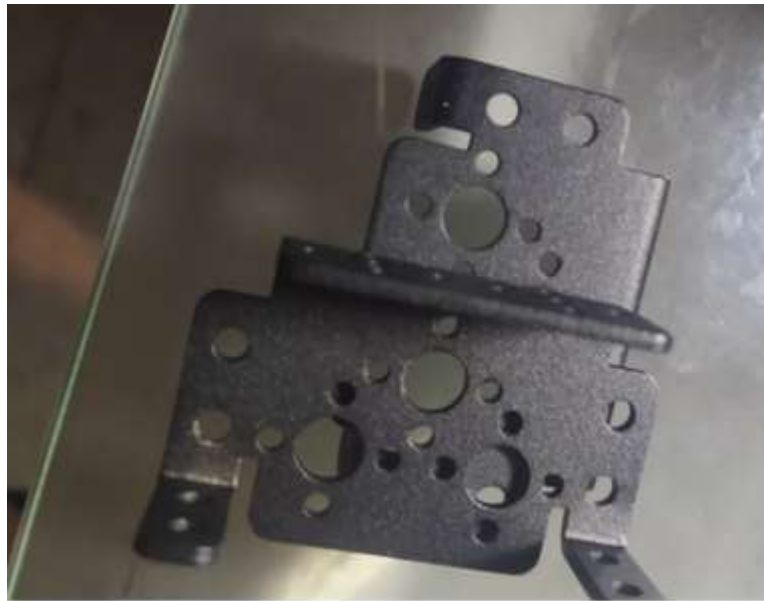
Se sugiere que quede de la siguiente manera:



Procedemos a identificar las siguientes dos piezas para el armado de la pinza



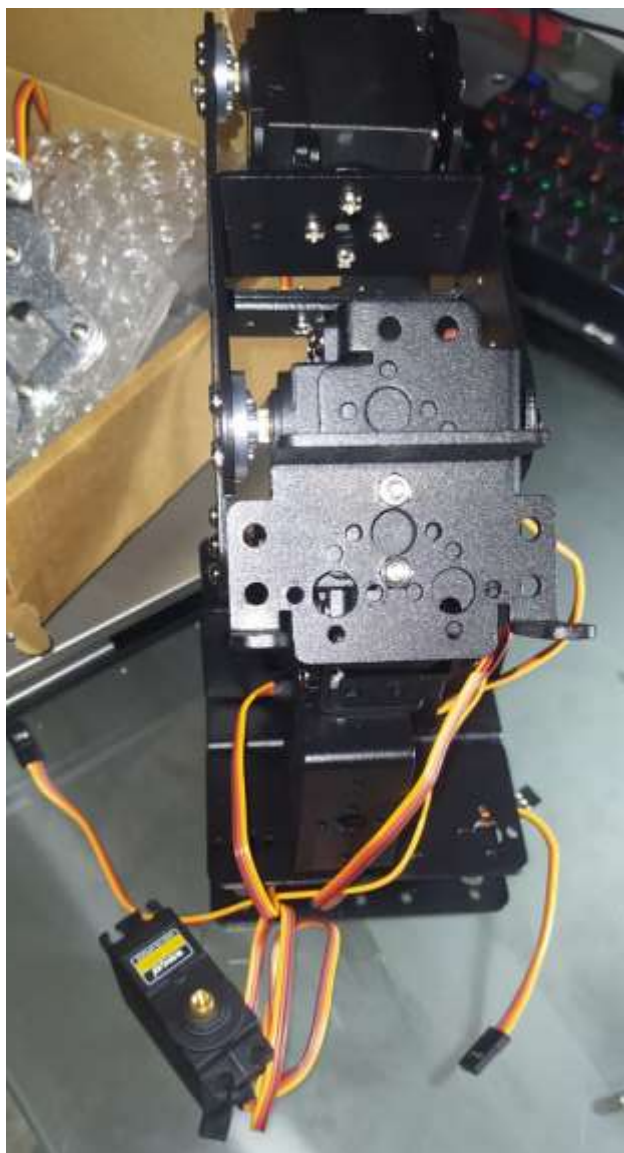
Debemos unir las de la siguiente manera.



Las ultimas dos piezas deben de unirse al extremo del brazo, a la pieza en forma de U. De la siguiente manera



Colocamos un servomotor en el extremo del brazo debe verse así:





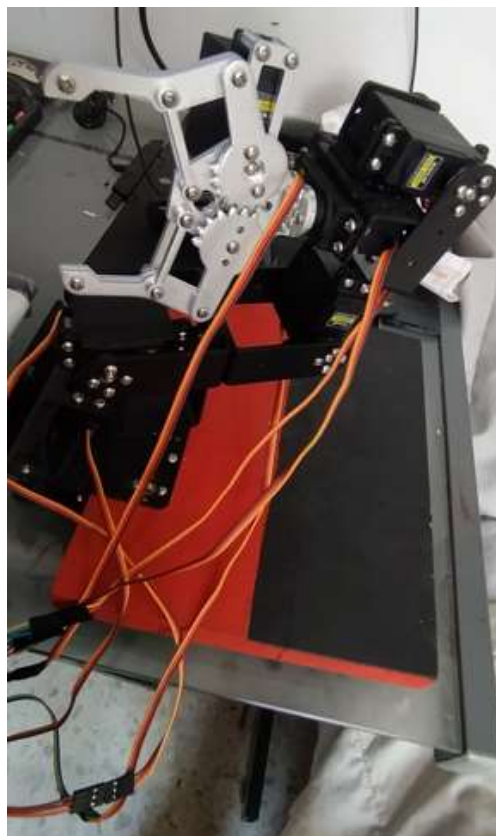
Ahora vamos a poner un disco metálico en la pinza para colocar un servomotor.



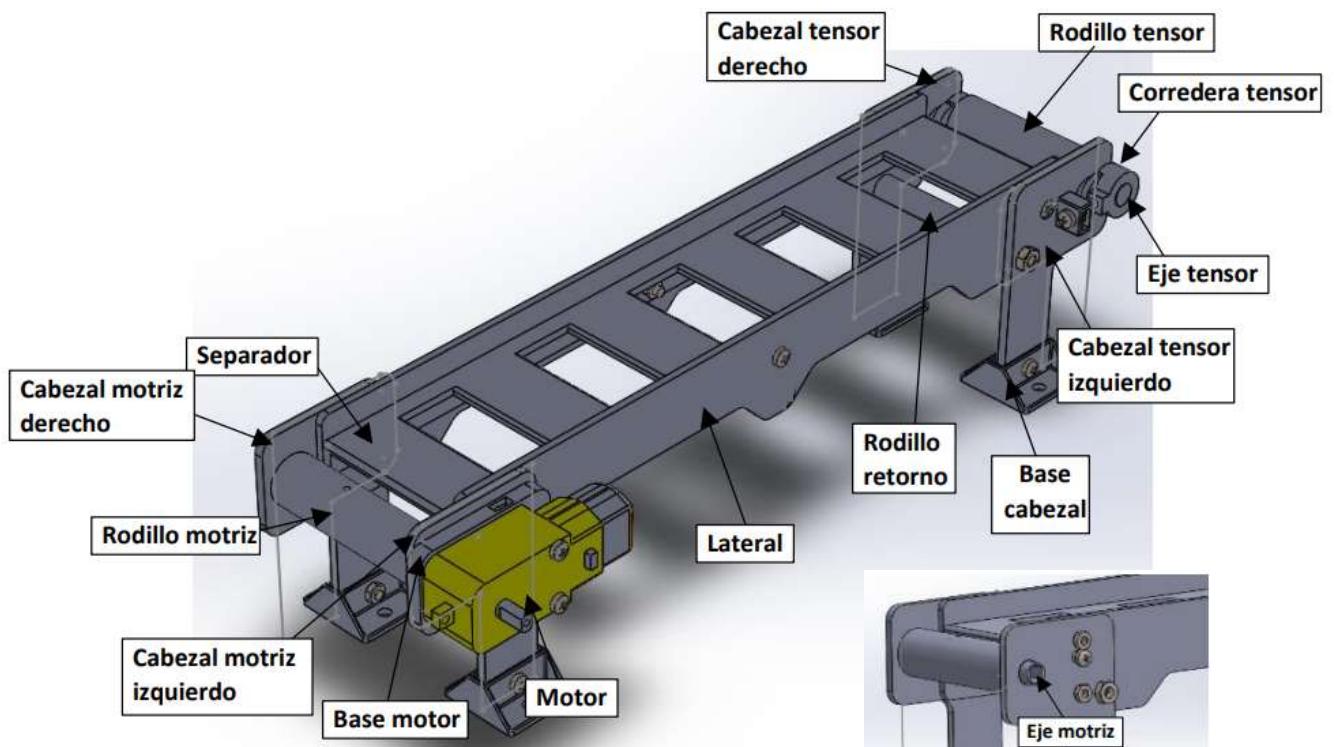
Vamos a unir la pinza con el extremo del brazo



El brazo robótico terminado luce así, se agrego una base de madera para dar un poco de soporte a los movimientos del brazo.

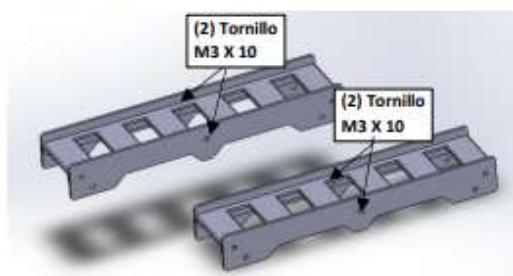


## ARMADO BANDA TRANSPORTADORA

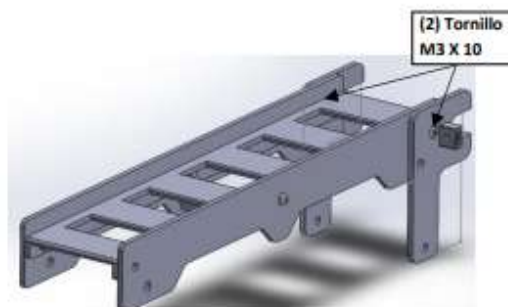


No. DE ELEMENTO	NOMBRE DE PIEZA	CANTIDAD
1	Separador	2
2	Cabezal motriz izquierdo	1
3	Cabezal motriz derecho	1
4	Cabezal tensor izquierdo	1
5	Cabezal tensor derecho	1
6	Lateral	4
7	Rodillo motriz	1
8	Rodillo tensor	1
9	Eje motriz	1
10	Eje tensor	1
11	Base motor	1
12	Corredera tensor	2
13	Base cabezal	6
14	Base sensor	2
15	Base unión	2
16	Rodillo retorno	3
17	Motor	1
18	Tornillo M3 X 10	12
19	Tornillo M3 X 12	7
20	Tornillo M3 X 16	1
21	Tornillo M3 X 25	2
22	Tuerca hexagonal M3	22
23	Tuerca de seguridad 3/16"	6
24	Varilla M4	3

**Paso 1. Montar los dos laterales sobre el separador.**



**Paso 2. Montar cabezal tensor izquierdo y derecho.**



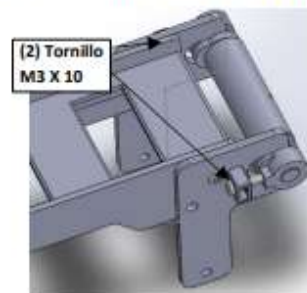
**Paso 3. Ensamblar eje tensor en rodillo tensor (49mm).**



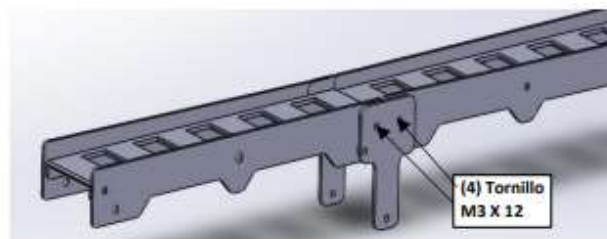
**Paso 4. Montar (2) corredera tensor a los extremos del eje tensor.**



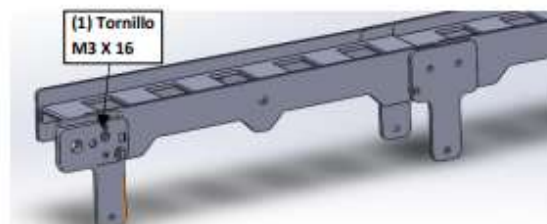
**Paso 5. Alinear (2) corredera tensor a la ranura cabezal tensor izquierdo y derecho, y montar.**



**Paso 6. Unir los dos separadores, acoplando base unión.**



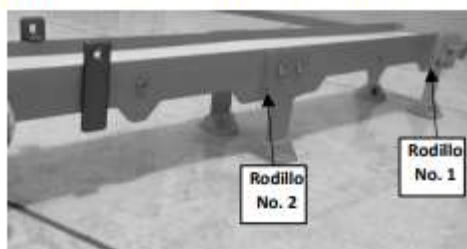
**Paso 7. Unir la base del motor con cabezal motriz izquierdo y montarlo en el lateral de la banda.**



**Paso 8. Montar banda de plástico sobre el separador.**



**Paso 9. Montar rodillo retorno (no. 1 y no.2) sobre varilla, ajustar con tuerca de seguridad 3/16".**



Rodillo retorno no.1 ----- 70mm

Rodillo retorno no.2 ----- 70 mm

Rodillo retorno no.3 -----71 mm

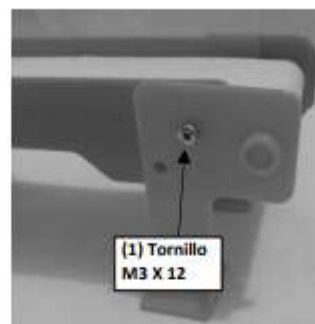
**Paso 10. Ensamblar eje motriz en rodillo motriz (55mm).**



**11. Introducir rodillo y eje motriz en la banda de plástico y alinear con barreno de cabezal motriz izquierdo.**



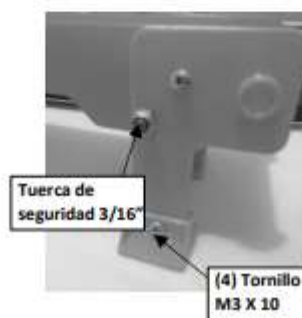
**12. Montar cabezal motriz derecho, alinear e introducir eje motriz.**



**Paso 13. Introducir tuerca de seguridad 3/16" en base motor.**



**Paso 14. Montar rodillo retorno sobre varilla de 71 mm, ajustar con tuerca de seguridad 3/16".**



**Paso 15. Montar motor sobre base de motor y ajustar el sistema tensor de la banda.**



## DESARROLLO DE LOS MÓDULOS: JUSTIFICACIÓN ELECTRÓNICA ANALÓGICA

El sistema cuenta con un sistema de alimentación de energía específico. Se propone el diseño de un circuito electrónico de potencia para el suministro de suficiente corriente y voltaje a los motores y sensores. Lo anterior debido a que la tarjeta de desarrollo no puede proporcionar suficiente energía a todos los motores y sensores de este sistema. El sistema elegido utiliza la energía suministrada por un transformador reductor de 127VAC/12VCC y 4 A mediante un puente rectificador de onda completa, además cuenta con reguladores de voltaje y transistores de potencia. Para el uso del motor reductor se implementa un puente H.

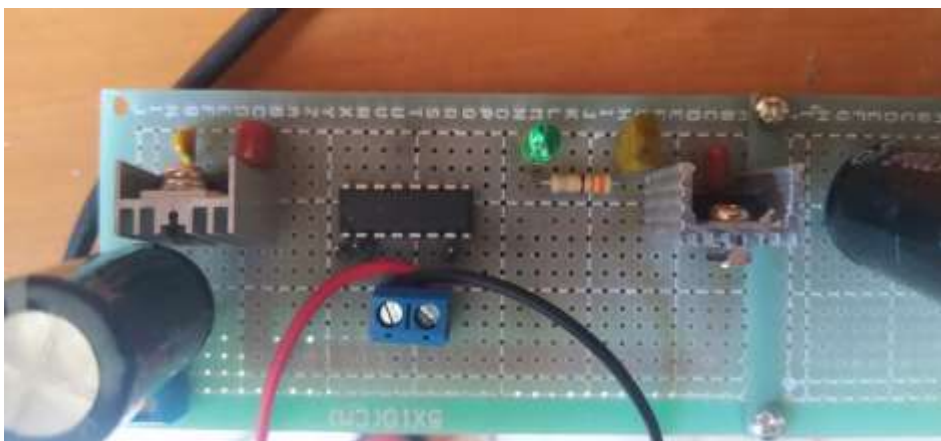
### DESARROLLO ELECTRÓNICA ANALÓGICA

El sistema tiene un voltaje de alimentación de entrada de 127VAC 60Hz, este voltaje es reducido con un transformador con derivación central de 24V 5A, se realiza la rectificación de onda completa de la señal con un puente de diodos modelo 6A8 con una capacidad de corriente máxima ideal rectificadora de 6A y un voltaje máximo en inversa de 800V imposible de romper aun con sobre tensiones de una



línea de 127V. Posterior al puente rectificador se implementan capacitores para el filtrado de señal, 3 de 4700µF a 50V y uno de 2200µF a 50V, los capacitores toman el voltaje positivo de la señal rectificada y la derivación central como tierra, esta configuración seleccionada proporciona 18.03Volts de corriente continua. De la mencionada fuente de voltaje se considera para alimentar un regulador de voltaje de 5V modelo 7805CV que suministra una corriente máxima de 1A, el voltaje de salida del regulador alimenta un encapsulado DIP puente H modelo L293D para suministrar el voltaje al motor reductor de la banda transportadora

Los servomotores MG996R requieren entre 4.8V 400mA y 7.2V 900mA para funcionar para suministrar la energía requerida se utiliza un regulador de voltaje de 6V modelo 7806CV y un transistor pnp modelo TIP42C con una corriente máxima ideal de 6A, la señal de entrada del regulador 7806CV es la misma que la señal de entrada del regulador 7805CV proveniente del transistor (18.08VDC).



## JUSTIFICACIÓN ELECTRÓNICA DIGITAL

Dentro de este módulo se crea un programa en lenguaje de programación C++, haciendo uso de declaración de variables funciones y sentencias para el control y ejecución del programa en el entorno de la tarjeta de desarrollo ESP 32. Diseño diagrama esquemático.

Se determina el posicionamiento adecuado de los servomotores aplicación de la modulación de ancho de pulso, y ciclo de trabajo.

Elaboración de un sistema de control de giro del motor de la banda transportadora en lenguaje c++

## DESARROLLO ELECTRÓNICA DIGITAL

Dentro de este módulo el programa del sistema se ejecuta en una tarjeta de desarrollo con el microcontrolador modelo ESP32. Para la programación de la tarjeta se utiliza el IDE de Arduino con lenguaje de programación en C++. El brazo robótico tiene 6 servomotores que requieren un periodo de modulación de ancho de pulso de 20 ms generado por el ESP32. El puente H L293D requiere un estado alto en el pin de activación y el mismo circuito integrado aplica una compuerta XOR en los pines 2 y 7 donde el orden de las entradas indica el sentido del giro del motor 1 y 0 gira en un sentido 0 y 1 gira en el otro sentido, si se aplican estados iguales el motor estará apagado.

La elaboración de la interface en QT creator implementa la decodificación de códigos QR para ello se utiliza zxing una herramienta para procesamiento de imágenes de códigos de barras, una vez decodificado el código qr el ESP32 determinara la zona correspondiente del día decodificado.

## MODULO COMUNICACIONES

Dentro de este módulo se da la implementación de protocolos de comunicación serial, y TCP/IP para la comunicación entre la tarjeta de desarrollo, la computadora y la interfaz gráfica.

Dentro de este módulo se crea un servidor WEB NGINX con base de datos MySQL para la consulta de los registros capturados y enviados mediante formato JSON por la tarjeta de desarrollo a la interfaz del programa.

## DESARROLLO COMUNICACIONES

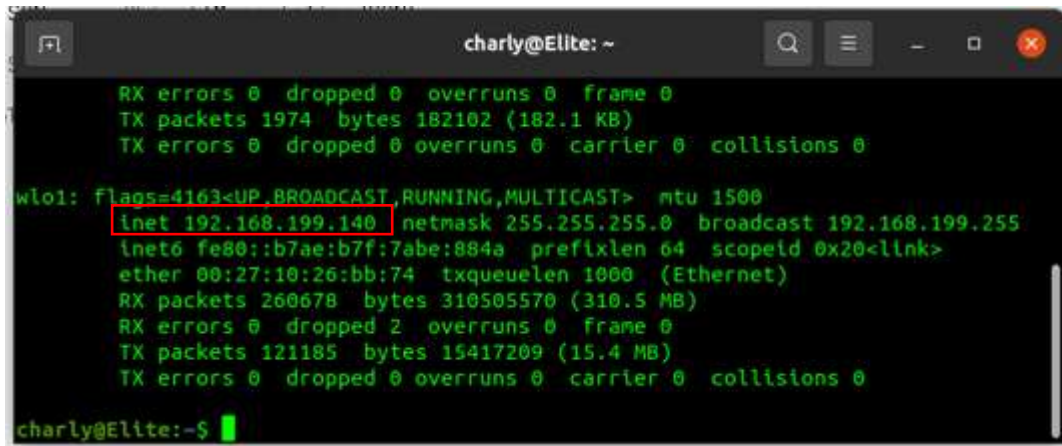
Se monta un servidor web en el ESP32 para ello se requiere una conexión a una red wifi con internet, al realizar la conexión con el punto de acceso se le asigna una dirección IP al ESP 32, esta dirección IP es la dirección IP para el servidor web, este servidor el medio de conexión y comunicación entre la interfaz gráfica que es el cliente y el ESP32 es el servidor.

Una vez realizada la conexión, se escanea el código QR para determinar el día de entrega del paquete, dependiendo del registro del paquete, será enviada la señal codificada por mensaje JSON desde la interfaz hacia el ESP32, si el ESP32 detecta un mensaje JSON lo comienza a recibir y almacenar, una vez que se termina de recibir el mensaje se codifica para comprobar si contiene alguna instrucción para ejecutar una función de secuencia de acomodo de paquetes.

Este módulo incluye la modificación de APN en teléfono celular que permita compartir red wifi en computadora y celular, la conexión a la red telefónica genera una dirección IP para los dispositivos estas direcciones serán el medio de comunicación entre la placa ESP32 y la computadora y el servidor WEB.

Para configurar la conexión via internet del esp32 con la interface de la computadora es necesario conocer la dirección IP de la computadora, para ello ejecutamos el siguiente comando en la terminal de Ubuntu:

if config



```
charly@Elite: ~  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 1974 bytes 182102 (182.1 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.199.149 netmask 255.255.255.0 broadcast 192.168.199.255  
    inet6 fe80::b7ae:b7f:7abe:884a prefixlen 64 scopeid 0x20<llnk>  
    ether 00:27:10:26:bb:74 txqueuelen 1000 (Ethernet)  
    RX packets 260678 bytes 310505570 (310.5 MB)  
    RX errors 0 dropped 2 overruns 0 frame 0  
    TX packets 121185 bytes 15417209 (15.4 MB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
charly@Elite:~$
```

La dirección IP asignada es 192.168.199.149 es la que esta encerrada en el rectángulo rojo inet. Esta dirección se debe de configurar en el código del esp32 para realizar la conexión via internet.

```
void conexionURL(){  
    bool ElServidorFunciona = true;  
    //Variable de tipo caracter para almacenar el mensaje JSON enviado al esp32  
    char MensajeTipoJSON[200];  
    //Buffer para no perder los caracteres que van llegando  
    char temporal[50];  
    //Variable de tipo caracter que contiene la pagina web que va a desplegar el esp32  
    char mensajeHTML[500];  
    char urlVar[10] = "/";  
    //Aqui va la direccion ip de la pc  
    char Servidor[25] = "192.168.199.140";  
    //Asignacion de usuario  
    char Usuario[10] = "Brazo";  
    int j = 0;  
    for(j = 0; j<=499; j++){  
        mensajeHTML[j] = NULL;  
    }  
    for(j = 0; j<=199; j++){  
        MensajeTipoJSON[j] = NULL;  
    }  
}
```

Se debe compilar el proyecto, subir a la tarjeta y realizar la conexión del esp32 con el punto de acceso wifi. Al realizar la conexión, se debe consultar vía comunicación serial la dirección IP asignada al esp32. Esta dirección IP será la dirección IP del servidor al cual se conectará la interface para indicar la secuencia del acomodo del paquete. En la ventana de la terminal, se presentan los datos de conexión de la tarjeta esp32, la dirección IP asignada al esp32 es "192.168.199.19". Esta dirección se debe colocar nuevamente en el código del esp32, compilar y volver a subir a la tarjeta. Si la configuración con el punto de acceso es la adecuada, el servidor en el esp32 estará listo para su comunicación con la interface y esperar los mensajes JSON enviados por la interface.



```
/dev/ttyUSB0  
.....  
Conectando a la red WifiRedmi Note 8  
.....  
Conexion Wifi con exito  
La direccion IP asignada para el esp32 es:  
192.168.199.19  
La direccion MAC asignada al esp32 es:  
30:AE:A4:6C:3D:E8  
  
Autoscroll [x] Mostrar marca temporal [x] Ambos NL
```



## MODULO INSTRUMENTACIÓN Y CONTROL

En este módulo se implementa el modelado matemático del motor de la banda transportadora para diseñar un controlador PID que permita una buena respuesta transitoria y obtener un estado estable. Lo anterior para tener un control de la posición y velocidad de movimiento de la banda transportadora y proporcionar tiempo suficiente para que el brazo robótico trabaje adecuadamente y posicionar el paquete en el rango de operación del brazo robótico.

## DESARROLLO INSTRUMENTACIÓN Y CONTROL

El sistema cuenta con retroalimentación que permite saber cuándo termina de recibir información durante la decodificación de la información del mensaje JSON. Si el mensaje decodificado contiene una secuencia de control conocida el ESP 32 ejecutara la función de la secuencia de clasificado de paquete para que el brazo robótico coloque el paquete en su lugar y el motor de la banda transportadora proporcione tiempo suficiente para que el brazo robótico pueda tomar con la pinza el paquete y dejarlo en el área correspondiente del día de entrega. Esto busca obtener agilizar y obtener un mayor control en la entrega de paquetes con mayor número de entregas fallidas.

Se desarrollo un sistema seguro y adecuado para la operación sin riesgos al usuario, separando las etapas de control donde se manejan bajos niveles de voltaje y corriente y la etapa de electrónica de potencia donde el paso de corriente es mucho mayor y se pueden ocasionar lesiones.

### SOFTWARE CODIGO C++ ESP32

```
//-----  
//PROYECTOS MODULARES 5 DE MAYO DEL 2022-A AGUILAR RODRIGUEZ CARLOS ADOLFO 215860049  
//CODIGO ESP 32 LECTOR CODIGO QR CON BANDA TRANSPORTADORA Y 6 SERVOMOTORES  
//BRAZO ROBOTICO CLASIFICADOR DE PAQUETES CON CON CODIGO QR  
//-----Ultima Modificacion 04/Mayo/2022/23:29-----  
//-----INCLUIR LIBRERIAS NECESARIAS-----  
#include "BluetoothSerial.h"  
#include <Servo.h>  
#include <WiFi.h>  
#include <ESPmDNS.h>  
#include <WiFiClient.h>  
#include <ArduinoJson.h>  
#include <EEPROM.h>  
#include "esp_system.h"  
#include <Servo.h>  
//-----DECLARACIONES DE LAS VARIABLES GLOBALES-----  
String header;  
int GradosEje4;  
int GradosBase;  
int GradosEje2;  
int GradosRotacionPinza;  
int GradosEje1;  
int ServoPinza;  
static const int servoPin4 = 4;  
static const int servoPin16 = 16;  
static const int servoPin17 = 17;  
static const int servoPin23 = 23;  
static const int servoPin19 = 19;
```

```

static const int servoPin21 = 21;
String estadoSalida = "off";
const int salida = 5;
const int retardoservo = 1500;
int Pin1MotorPuentes = 27;
int Pin2MotorPuentes = 26;
int HabilitarPinPuentes = 14;
Servo Base;
Servo Eje1;
Servo Eje2;
Servo RotacionPinza;
Servo Eje4;
Servo Pinza;

//-----CONFIGURACION DEL PWM -----
const int freq = 30000;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 240;

/*String pagina = "<!DOCTYPE html>"
"<html>"
"<head>"
"<meta charset='utf-8' />"
"<title>Proyecto Modular Aguilar Rodriguez Carlos Adolfo</title>"
"</head>"
"<body>"
"<center>"
"<h1>Proyectos Modulares 5 MAYO 2022A</h1>"
"<p><a href='/on'><button style='height:30px;width:110px'>ENCENDER</button></a></p>"
"<p><a href='/off'><button style='height:30px;width:110px'>APAGAR</button></a></p>"
"<p><a href='/Lunes'><button style='height:30px;width:110px'>Lunes</button></a></p>"
"<p><a href='/Martes'><button style='height:30px;width:110px'>Martes</button></a></p>"
"<p><a href='/Miercoles'><button style='height:30px;width:110px'>Miercoles</button></a></p>"
"<p><a href='/Jueves'><button style='height:30px;width:110px'>Jueves</button></a></p>"
"<p><a href='/Viernes'><button style='height:30px;width:110px'>Viernes</button></a></p>"
"</center>"
"</body>"
"</html>";
*/

//-----INICIAR EL SERVIDOR WIFI-----
WiFiServer IniciarServidor(80);
//Este es el nombre de la red a la que se conectara el esp32
char ssid[100] = "Redmi Note 8";
//Esta es la contraseña de la red a donde se conectara el esp32
char password[100] = "00051000";
//Opcion para cambiar la modalidad del cliente o del servidor
bool Servidor = true;
bool FinalDelMensaje = false;
//Variable para cuantificar las veces que se han intentado realizar la conexion del esp 32 con la red wifi
long int TiempoInicio = 0;
long int TiempoFinal = 0;
long int TiempoConexionInicio = 0;
long int CompararTiempo = 0;
int Contador = 0;
//Borrar la cadena de caracteres que contiene la linea actual

```

```

String LineaActual = "";
//Variable para determinar cuando se termina de leer el mensaje

void conectaWiFi(){

    Serial.println();
    Serial.print("Conectando a la red Wifi");
    Serial.println(ssid);
    //Comando ordenar al esp32 conectarse a la red asignada en ssid y al password
    WiFi.begin(ssid,password);
    //iteracion para realizar la conexion wifi, se intentara conectar hasta que se conecte
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("Conexion Wifi con exito");
    Serial.println("La direccion IP asignada para el esp32 es: ");
    //Mostrar la direccion IP asignada al esp32
    Serial.println(WiFi.localIP());
    Serial.println("La direccion MAC asignada al esp32 es: ");
    //Mostrar la direccion MAC asignada al esp32
    Serial.println(WiFi.macAddress());
    //Asignar el nombre Modular5Mayo a la pagina web
    if (!MDNS.begin("Modula5Mayo")) {
    }
    else{
        //Iniciar el servidor del esp32
        IniciarServidor.begin();
        //Configurar los protocolos y el puerto de comunicacion del servidor
        MDNS.addService("http", "tcp", 80);
    }
}

//Funcion para detener la banda transportadora
void DetenerBanda() {
    Serial.println("Deteniendo Motor de la banda transportadora");
    digitalWrite(HabilitarPinPuentesH, LOW);
    digitalWrite(Pin1MotorPuentesH, LOW);
    digitalWrite(Pin2MotorPuentesH, LOW);
    delay(1500);
}

//Funcion para avanzar la banda transportadora
void AvanzarBanda() {
    Serial.println("Avanzando Motor de la banda transportadora");
    digitalWrite(HabilitarPinPuentesH, HIGH);
    digitalWrite(Pin1MotorPuentesH, LOW);
    digitalWrite(Pin2MotorPuentesH, HIGH);
    delay(1150);
}

//Funcion para realizar la conexion con el cliente y leer el mensaje JSON
void conexionURL(){
    bool EIServidorFunciona = true;
    //Variable de tipo caracter para almacenar el mensaje JSON enviado al esp32

```

```

char MensajeTipoJSON[200];
//Buffer para no perder los caracteres que van llegando
char temporal[50];
//Variable de tipo caracter que contiene la pagina web que va a desplegar el esp32
char mensajeHTML[500];
char urlVar[10] = "/";
//Aqui va la direccion ip de la pc
char Servidor[25] = "192.168.199.140";
//Asignacion de usuario
char Usuario[10] = "Brazo";
int j = 0;
for(j = 0; j<=499; j++){
    mensajeHTML[j] = NULL;
}
for(j = 0; j<=199; j++){
    MensajeTipoJSON[j] = NULL;
}

//Crear la estructura del mensaje tipo JSON
sprintf(MensajeTipoJSON, "{\"var\":\"%i\",\"key\":\"sdfsdf\"}", Contador);
    Contador++;
int cuantosBytes = strlen(MensajeTipoJSON);
//Serial.println(mensajeJSON);
//Serial.print("Bytes: ");
//Serial.println(cuantosBytes);
//Crear el mensaje HTML

sprintf(temporal, "POST %s HTTP/1.0\r\n", urlVar);
strcat(mensajeHTML, temporal);

sprintf(temporal, "Host: %s\r\n", Servidor);
strcat(mensajeHTML, temporal);

sprintf(temporal, "User-Agent: %s\r\n", Usuario);
strcat(mensajeHTML, temporal);

sprintf(temporal, "Content-Length: %i\r\n", cuantosBytes);
strcat(mensajeHTML, temporal);

strcat(mensajeHTML, "Content-Type: application/json\r\n");
strcat(mensajeHTML, "\r\n");
strcat(mensajeHTML, MensajeTipoJSON);

//Serial.println(mensajeHTML);
int Mensajes = strlen(mensajeHTML);
//Sentencia para realizar la conexion URL si el servidor funciona, realizar la conexion
if(ElServidorFunciona){
    WiFiClient client;
    //Aqui va la direccion IP asignada al ESP32
    if (client.connect("192.168.199.19", 80)) {
        Serial.print("Bytes para transmitir: ");
        Serial.println(Mensajes);
        for(j = 0; j<=Mensajes-1; j++){
            client.print(mensajeHTML[j]);
            Serial.print(mensajeHTML[j]);
        }
    }
}

```

```

Serial.println(" ");
delay(1000);
client.stop();

}
else{
  Serial.println("No se pudo conectar con el servidor error de conexion por favor verifique la configuracion con el
servidor");
}
}
else{
  Serial.print("Bytes ara transmitir: ");
  Serial.println(Mensajes);
  for(j = 0; j<=Mensajes-1; j++){
    Serial.print(mensajeHTML[j]);
  }
  Serial.println(" ");
}
}
}

```

//Funcion para realizar las configuracion globales del esp32

```

void setup() {
  //Iniciar la comunicacion serial a 115200 Baudios
  Serial.begin(115200);
  //Asignacion de funciones de servomotor a variables de los pines
  Base.attach(servoPin4);
  Eje1.attach(servoPin16);
  Eje2.attach(servoPin17);
  RotacionPinza.attach(servoPin23);
  Eje4.attach(servoPin19);
  Pinza.attach(servoPin21);
  //Asignacion de salidas a los pines del motor
  pinMode(Pin1MotorPuentesH, OUTPUT);
  pinMode(Pin2MotorPuentesH, OUTPUT);
  pinMode(HabilitarPinPuentesH, OUTPUT);
  pinMode(salida, OUTPUT);
  //Llamado a la funcion de conexion wifi
  conectaWiFi();
  TiempoInicio = millis();
}

```

//Funcion de secuencia para llevar el brazo a su posicion principal

```

void SecuenciaServoPrincipal(){
  Base.write(60);
  delay(20);
  Eje1.write(45);
  delay(20);
  Eje2.write(1);
  delay(20);
  RotacionPinza.write(99);
  delay(20);
  Eje4.write(30);
  delay(20);
}

```

//Funcion de secuencias para que el brazo este en posicion de agarrar el paquete



```

void SecuenciaServoAgarrar(){
    Base.write(60);
    Serial.println("La base se desplaza hacia 60 grados");
    delay(20);

    Eje1.write(98);
    Serial.println("El Eje1 se desplaza hacia 97 grados");
    delay(20);

    Eje2.write(1);
    Serial.println("El Eje2 se desplaza hacia 2 grados");
    delay(20);

    RotacionPinza.write(99);
    Serial.println("La rotacion de la pinza se desplaza 99 grados");
    delay(20);

    Eje4.write(95);
    Serial.println("El Eje4 se desplaza hacia 95 grados");
    delay(20);

    //ServoPinza = 5;
    Pinza.write(5);
    Serial.println(ServoPinza);
    delay(20);
}
//Funcion para cerrar la pinza
void CerrarPinza(){
    Pinza.write(145);
    Serial.println("La pinza se mueve hacia 130 grados");
    delay(20);
}
//Funcion para Abrir la pinza
void AbrirPinza(){
    //ServoPinza = 75;
    Pinza.write(5);
    Serial.println("La pinza se mueve hacia 5 grados");
    delay(20);
}
//Funcion de secuencias para que el brazo coloque el paquete en la posicion de el lunes
void SecuenciaServoLunes(){
    Eje1.write(45);
    Serial.println("El Eje1 se desplaza hacia 45 grados");
    delay(400);
    Base.write(130);
    Serial.println("La base se desplaza hacia 130 grados");
    delay(20);
    Eje2.write(45);
    Serial.println("El Eje2 se desplaza hacia 45 grados");
    delay(400);
    RotacionPinza.write(99);
    Serial.println("La rotacion de la pinza se desplaza hacia 99 grados");
    delay(20);
    Eje4.write(165);
    Serial.println("El Eje4 se desplaza hacia 165 grados");
    delay(20);
    Pinza.write(130);
}

```

```
Serial.println(ServoPinza);  
delay(20);  
}
```

//Funcion de secuencias para que el brazo coloque el paquete en la posicion de el Martes

```
void SecuenciaServoMartes(){  
    //GradosEje1 = 55;  
    Eje1.write(45);  
    Serial.println("El Eje! se desplaza hacia 45 grados");  
    delay(400);  
  
    //GradosBase=90;  
    Base.write(100);  
    Serial.println("Los grados de base se desplaza hacia 100 grados");  
    delay(20);  
  
    //GradosEje2 = 30;  
    Eje2.write(45);  
    Serial.println("El Eje2 se desplaza hacia 45 grados");  
    delay(400);  
  
    //GradosRotacionPinza = 99;  
    RotacionPinza.write(99);  
    Serial.println("Los grados de la pinza se desplazan hacia 165 grados");  
    delay(20);  
  
    //GradosEje4 = 165;  
    Eje4.write(165);  
    Serial.println("El Eje4 se desplaza hacia 165 grados");  
    delay(20);  
    Pinza.write(130);  
    Serial.println(ServoPinza);  
    delay(20);  
}
```

//Funcion de secuencias para que el brazo coloque el paquete en la posicion de el Miercoles

```
void SecuenciaServoMiercoles(){  
  
    //GradosEje1 = 55;  
    Eje1.write(45);  
    Serial.println("El Eje1 se desplaza hacia 45 grados");  
    delay(400);  
  
    //GradosBase=90;  
    Base.write(70);  
    Serial.println("La Base se desplaza hacia 70 grados");  
    delay(20);  
  
    //GradosEje2 = 30;  
    Eje2.write(45);  
    Serial.println("El Eje2 se desplaza hacia 45 grados");  
    delay(400);  
  
    //GradosRotacionPinza = 99;
```

```
RotacionPinza.write(99);
Serial.println("La rotacion se desplaza hacia 165 grados");
delay(20);
```

```
//GradosEje4 = 165;
Eje4.write(165);
Serial.println("El Eje4 se desplaza hacia 165 grados");
delay(20);
```

```
}
```

```
//Funcion de secuencias para que el brazo coloque el paquete en la posicion de el Jueves
void SecuenciaServoJueves(){
```

```
//GradosEje1 = 55;
Eje1.write(45);
Serial.println("El Eje1 se desplaza hacia 45 grados");
delay(400);
```

```
//GradosBase=90;
Base.write(40);
Serial.println("La base se desplaza hacia 40 grados");
delay(20);
```

```
//GradosEje2 = 30;
Eje2.write(45);
Serial.println("El Eje2 se desplaza hacia 45 grados");
delay(400);
```

```
//GradosRotacionPinza = 99;
RotacionPinza.write(99);
Serial.println("El Eje4 se desplaza hacia 99 grados");
delay(20);
```

```
//GradosEje4 = 165;
Eje4.write(165);
Serial.println("El Eje4 se desplaza hacia 165 grados");
delay(20);
```

```
}
```

```
//Funcion de secuencias para que el brazo coloque el paquete en la posicion de el Viernes
void SecuenciaServoViernes(){
```

```
//GradosEje1 = 55;
Eje1.write(45);
Serial.println("El Eje! se desplaza hacia 55 grados");
delay(400);
```

```
//GradosBase=90;
Base.write(10);
Serial.println("La base se desplaza hacia 10 grados");
delay(20);
```

```
//GradosEje2 = 30;
Eje2.write(45);
Serial.println("El Eje2 se desplaza hacia 45 grados");
```

```

delay(400);

//GradosRotacionPinza = 99;
RotacionPinza.write(99);
Serial.println("La rotacion de la pinza se desplaza hacia 99 grados");
delay(20);

//GradosEje4 = 165;
Eje4.write(165);
Serial.println("El Eje4 se desplaza hacia 165 grados");
delay(20);

}

```

//ciclo principal del esp 32 para ejecucion del programa

```

void loop() {
  if(Servidor){
    LineaActual = " ";
    WiFiClient clienteServidor = IniciarServidor.available();
    FinalDelMensaje = false;
    if (clienteServidor) {
      TiempoConexionInicio = xTaskGetTickCount();
      while(clienteServidor.connected()){
        if(clienteServidor.available() > 0){
          char c = clienteServidor.read();      // read a byte, then
          Serial.print((char)c);                // print it out the serial monitor
          //header += c;
          ///Comenzar a recibir el mensaje tipo JSON
          if(c == ' '){ FinalDelMensaje = true; }
          if(c == '\n') {      // if the byte is a newline character

            if (LineaActual.length() == 0) {
              //Inicia la respuesta
              /*
              clienteServidor.println("HTTP/1.1 200 OK");
              clienteServidor.println("Content-type:text/html");
              clienteServidor.println();

              if (header.indexOf("GET /on") >= 0) {
                Serial.println("GPIO on");
                estadoSalida = "on";
                digitalWrite(salida, LOW);
              } else if (header.indexOf("GET /off") >= 0) {
                Serial.println("GPIO off");
                estadoSalida = "off";
                digitalWrite(salida, HIGH);
              }
              else if (header.indexOf("GET /Lunes") >= 0) {
                Serial.println("GPIO off");
                estadoSalida = "off";
                digitalWrite(salida, HIGH);
              } */
              //clienteServidor.println(pagina);
              //clienteServidor.stop();
              //clienteServidor.println();
              //break;
            } else { // if you got a newline, then clear currentLine:

```

```

LineaActual = "";
}
} else if (c != '\r') { // if you got anything else but a carriage return character,
LineaActual += c;    // add it to the end of the currentLine
}
if(FinalDelMensaje){
String MensajeTipoJSON = LineaActual;
//Serial.println(mensajeJSON);
//Decodificacion mensaje JSON - inicio
StaticJsonBuffer<200> bufferJSON;
JsonObject& objetoJSON = bufferJSON.parseObject(MensajeTipoJSON);
if(objetoJSON.success()){
int opcionMotor = objetoJSON["Servomotor"];
if(opcionMotor == 1){
//Mover los motores para adelante 1 segundo
digitalWrite(salida, HIGH);
delay(1000);
Serial.println("Lunes");
AvanzarBanda();
DetenerBanda();
AbrirPinza();
delay(retardoservo);

SecuenciaServoAgarrar();
delay(retardoservo);
CerrarPinza();
delay(retardoservo);

SecuenciaServoLunes();
delay(retardoservo);
AbrirPinza();
delay(retardoservo);

//delay(retardoservo);
//SecuenciaServoPrincipal();
//delay(retardoservo);
digitalWrite(salida, LOW);
break;
}
else if(opcionMotor == 2){
//Mover los motores para atras 1 segundo
delay(1000);
Serial.println("Martes");
digitalWrite(salida, HIGH);
AvanzarBanda();
DetenerBanda();
AbrirPinza();
delay(retardoservo);

SecuenciaServoAgarrar();
delay(retardoservo);
CerrarPinza();
delay(retardoservo);

SecuenciaServoMartes();
delay(retardoservo);
AbrirPinza();
}
}

```



```

delay(retardoservo);

//delay(retardoservo);
//SecuenciaServoPrincipal();
//delay(retardoservo);
digitalWrite(salida, LOW);
}
else if(opcionMotor == 3){
    delay(1000);
    Serial.println("Miercoles");
    digitalWrite(salida, HIGH);
    AvanzarBanda();
    DetenerBanda();
    AbrirPinza();
    delay(retardoservo);
    SecuenciaServoAgarrar();
    delay(retardoservo);
    CerrarPinza();
    delay(retardoservo);

    SecuenciaServoMiercoles();
    delay(retardoservo);
    AbrirPinza();
    delay(retardoservo);

    //delay(retardoservo);
    //SecuenciaServoPrincipal();
    //delay(retardoservo);
    digitalWrite(salida, LOW);

}
else if(opcionMotor == 4){
    delay(1000);
    Serial.println("Jueves");
    digitalWrite(salida, HIGH);
    AvanzarBanda();
    DetenerBanda();
    AbrirPinza();
    delay(retardoservo);
    SecuenciaServoAgarrar();
    delay(retardoservo);
    CerrarPinza();
    delay(retardoservo);

    SecuenciaServoJueves();
    delay(retardoservo);
    AbrirPinza();
    delay(retardoservo);

    //delay(retardoservo);
    //SecuenciaServoPrincipal();
    //delay(retardoservo);
    digitalWrite(salida, LOW);
}
else if(opcionMotor == 5){
    delay(1000);
    Serial.println("Viernes");

```

```

digitalWrite(salida, HIGH);
AvanzarBanda();
DetenerBanda();
AbrirPinza();
delay(retardoservo);

SecuenciaServoAgarrar();
delay(retardoservo);
CerrarPinza();
delay(retardoservo);

SecuenciaServoViernes();
delay(retardoservo);
AbrirPinza();

delay(retardoservo);
//delay(retardoservo);
//SecuenciaServoPrincipal();
//delay(retardoservo);
digitalWrite(salida, LOW);

}
}

//Decodificacion del mensaje JSON
clienteServidor.println("HTTP/1.1 200 OK");
clienteServidor.println("Content-type:text/html");
clienteServidor.println();
clienteServidor.println("Configuracion Recibida");
clienteServidor.println();
//respuestaServidor();
break;
}
//respuestaServidor();

//Termina recibir mensaje JSON
CompararTiempo = xTaskGetTickCount();
if(CompararTiempo > (TiempoConexionInicio + 3000)){
    //clienteServidor.stop();
    Serial.println("Error, tiempo de espera excedido");
    break;
}
}
}
//header = "";
    clienteServidor.stop();
delay(500);
}
}
else{
    TiempoFinal = millis();
    if(TiempoFinal > (TiempoInicio+10000)){
        TiempoInicio = millis();
        //Este if, funcionara cada 2 segundos.
        conexionURL();
    }
}

```

```
}  
}
```

## CODIGO QT CREATOR

**El siguiente código pertenece al código fuente main.cpp**

```
#include "mainwindow.h"  
#include <QApplication>  
int main(int argc, char *argv[])  
{  
    QApplication a(argc, argv);  
    MainWindow w;  
    w.show();  
    return a.exec();  
}
```

**El siguiente código debe de ir en el archivo .pro es la configuración general de el proyecto en QT creator**

```
QT      += core gui serialport network  
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets  
  
CONFIG += c++11  
DEFINES += QT_DEPRECATED_WARNINGS  
CONFIG(debug, debug|release) {  
    CONFIG+=qml_debug  
} else {  
    DEFINES += QT_NO_DEBUG  
    DEFINES += QT_NO_DEBUG_OUTPUT  
}  
SOURCES += \  
    main.cpp \  
    mainwindow.cpp \  
    mat2qimage.cpp \  
HEADERS += \  
    mainwindow.h \  
    mat2qimage.h \  
FORMS += \  
    mainwindow.ui  
INCLUDEPATH += /usr/local/include/opencv4  
LIBS += -L/usr/local/lib -lopencv_core -lopencv_imgcodecs -lopencv_highgui -lopencv_videoio  
CONFIG += link_pkgconfig  
PKGCONFIG += opencv4  
RESOURCES +=  
  
# Default rules for deployment.  
qnx: target.path = /tmp/${TARGET}/bin  
else: unix:!android: target.path = /opt/${TARGET}/bin  
!isEmpty(target.path): INSTALLS += target
```

**El siguiente código debe ir en archivo fuente de c++ del proyecto**

```
#include<opencv4/opencv2/cvconfig.h>
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include<opencv2/core/core.hpp>
#include<opencv2/ml/ml.hpp>
#include<opencv2/imgproc/imgproc.hpp>
#include<opencv2/highgui/highgui.hpp>
#include<opencv2/video/background_segm.hpp>
#include<opencv2/videoio.hpp>
#include<opencv2/imgcodecs.hpp>
#include"mat2qimage.h"
#include<QTimer>
#include<QDebug>
#include<QtSerialPort>
#include<QSerialPortInfo>
#include<QFile>
#include<QTextStream>
#include<QDateTime>
#include<QDate>
#include<QFileDialog>
#include<QDebug>
#include<QtNetwork>
#include<QMouseEvent>
#include <opencv2/objdetect.hpp>
#include <iostream>

using namespace std;
using namespace cv;

VideoCapture camara(0);
//VideoCapture camara("http://192.168.43.1:8080/video");
Mat salidaMux;
Mat imagenProcesada;
//Mat nombreMatriz(Y, X, TipoMatriz, Scalar(B,G,R));
Mat error33(600, 800, CV_8UC3, Scalar(237,82,170));
Mat imagenFija;
Mat imagenChica;

int contadorImagenes = 0;
bool temporizadorOn = true;
int contadorTiempo = 0;
```

```

QString MainWindow::conexionWeb(QString url, QString mensaje){
    QString respuesta = "error 1";

    QNetworkInterface redConectada = QNetworkInterface::interfaceFromName("wlo1");
    QList<QNetworkAddressEntry> lista = redConectada.addressEntries();
    //Comprobacion de que la red este activa
    if(!lista.empty()){
        QNetworkAddressEntry IP = lista.first();
        qDebug() << "La conexion de red esta activa: " << IP.ip() << endl;
        //Crear el mensaje HTML/HTTP
        QByteArray longitudMensaje = QByteArray::number(mensaje.size());
        QNetworkRequest solicitud;
        QNetworkAccessManager *clienteWeb = new QNetworkAccessManager();
        QUrl servidor(url.toUtf8().constData());
        //Comprobar que la direccion url sea valida
        if(servidor.isValid()){
            qDebug() << "El servidor se encuentra activo" << endl;
            //Formar el mensaje HTTP
            solicitud.setUrl(servidor);
            solicitud.setRawHeader(QByteArray("User-Agent"),QByteArray("bot"));
            solicitud.setRawHeader(QByteArray("Connection"),QByteArray("close"));
            solicitud.setHeader(QNetworkRequest::ContentTypeHeader,"application/json");
            solicitud.setHeader(QNetworkRequest::ContentLengthHeader, longitudMensaje);
            //Realizar la conexion
            QNetworkReply *conexionServidor = clienteWeb->post(solicitud, mensaje.toLatin1());
            //QNetworkReply *conexionServidor = clienteWeb->get(QNetworkRequest(servidor));
            //Esperar a que servidor responda
            QEventLoop funcionLoop;
            QObject::connect(clienteWeb, SIGNAL(finished(QNetworkReply*)), &funcionLoop,
            SLOT(quit()));
            funcionLoop.exec();
            //Leer la respuesta del servidor
            char datosWeb[300000];
            respuesta.clear();
            int sv=conexionServidor->read(datosWeb,300000);
            for(int i=0;i<sv;i++) {
                respuesta+= datosWeb[i];
            }
        }else{
            respuesta = "Error 3" ;
        }
    }else{
        respuesta = "Error 2";
    }
    return respuesta;
}

void MainWindow::Temporizador(){

```



```

if(temporizadorOn){
    contadorTiempo++;
    Mat imagen;
    contadorImagenes++;
    if(ui->radioButton_5->isChecked()){
        //Capturar la imagen original que viene de la camara
        camara >> imagen;
    }else{
        if(imagenFija.empty()){
            error33.copyTo(imagen);
        }else{
            imagenFija.copyTo(imagen);
        }
    }
    if(!imagen.empty()){
        cv::resize(imagen, imagenChica, Size(512,282),0,0,0);
    }

    if(ui->radioButton->isChecked()){
        imagenChica.copyTo(imagenProcesada);
    }

    else{
        error33.copyTo(imagenProcesada);
    }

    if(!imagenProcesada.empty()){
        //Paso · 2 - Cambiar el tamaño
        cv::resize(imagenProcesada, salidaMux, Size(512,288),0,0,INTER_LINEAR);
    }

    //Convertir la imagen Mat a una imagen de QT
    QImage imagenQT = Mat2QImage(salidaMux);

    //Convertir la imagen de QT, a un mapa de pixeles
    QPixmap mapaPixeles = QPixmap::fromImage(imagenQT);
    //Borra todo lo que este en la etiqueta "label" de la interfaz grafica
    ui->label->clear();
    //Muestra la Matriz de Pixeles en la etiqueta "label" de la interfaz grafica
    ui->label->setPixmap(mapaPixeles);
}
}

```

```

MainWindow::MainWindow(QWidget *parent):QMainWindow(parent),ui(new Ui::MainWindow){
    ui->setupUi(this);
    //Crea un cronometro
    QTimer *cronometro = new QTimer(this);
    //Configurar el cronometro
    connect(cronometro, SIGNAL(timeout()), this, SLOT(Temporizador()));
}

```

```

//Inicia el cronometro
cronometro->start(50);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_pushButton_6_clicked()
{
    conexionWeb("http://Modular.local", "{\"Servomotor\":\"1\"}");
}

void MainWindow::on_pushButton_7_clicked()
{
    conexionWeb("http://Modular.local", "{\"Servomotor\":\"2\"}");
}

void MainWindow::on_pushButton_9_clicked()
{
    conexionWeb("http://Modular.local", "{\"Servomotor\":\"3\"}");
}

void MainWindow::on_pushButton_8_clicked()
{
    conexionWeb("http://Modular.local", "{\"Servomotor\":\"4\"}");
}

void MainWindow::on_pushButton_10_clicked()
{
    conexionWeb("http://Modular.local", "{\"Servomotor\":\"5\"}");
}

```

## CODIGO ESCANER QR QML

El siguiente bloque de código pertenece a el archivo de configuración principal de QML, el archivo debe tener, por ejemplo: Nombredelarchivo.pro

```
QT      += core gui serialport network
TEMPLATE = app
CONFIG += qzxing_multimedia \
    enable_decoder_1d_barcodes \
    enable_decoder_qr_code \
    enable_decoder_data_matrix \
    enable_decoder_aztec \
    enable_decoder_pdf17

CONFIG(debug, debug|release) {
    CONFIG+=qml_debug
} else {
    DEFINES += QT_NO_DEBUG
    DEFINES += QT_NO_DEBUG_OUTPUT
}

HEADERS += \
    application.h

SOURCES += main.cpp \
    application.cpp \

RESOURCES += qml.qrc

QML_IMPORT_PATH =

include(../src/QZXing-components.pri)
include(deployment.pri)
```

El siguiente código pertenece a el archivo qml que contiene la interfaz del escáner de código QR

```
import QtQuick 2.5
import QtQuick.Window 2.0
import QtQuick.Controls 2.0
import QtQuick.Layouts 1.1
import QtMultimedia

import QZXing 3.3

ApplicationWindow
{
    id: window
    visible: true
    width: 640
    height: 480
    title: "Lector de Codigo QR MODULARES 2022A"
```

```
property int detectedTags: 0
property string lastTag: ""
```

Rectangle

```
{
    id: bgRect
    color: "white"
    anchors.fill: videoOutput
}
```

Text

```
{
    id: text1
    wrapMode: Text.Wrap
    font.pixelSize: 20
    anchors.top: parent.top
    anchors.left: parent.left
    z: 50
    text: "Codigos leidos " + detectedTags
}
```

Text

```
{
    id: fps
    font.pixelSize: 20
    anchors.top: parent.top
    anchors.right: parent.right
    z: 50
    text: (1000 / zxingFilter.timePerFrameDecode).toFixed(0) + "fps"
}
```

Camera

```
{
    id: camera
    active: true
    focusMode: Camera.FocusModeAutoNear
}
```

CaptureSession {

```
    camera: camera
    videoOutput: videoOutput
}
```

VideoOutput

```
{
    id: videoOutput
    anchors.top: text1.bottom
    anchors.bottom: text2.top
    anchors.left: parent.left
    anchors.right: parent.right
}
```

```

// fillMode: VideoOutput.Stretch

property double captureRectStartFactorX: 0.25
property double captureRectStartFactorY: 0.25
property double captureRectFactorWidth: 0.5
property double captureRectFactorHeight: 0.5

MouseArea {
    anchors.fill: parent
    onClicked: {
        camera.customFocusPoint = Qt.point(mouseX / width, mouseY / height);
        camera.focusMode = Camera.FocusModeManual;
    }
}

Rectangle {
    id: captureZone
    color: "blue"
    opacity: 0.2
    width: parent.width * parent.captureRectFactorWidth
    height: parent.height * parent.captureRectFactorHeight
    x: parent.width * parent.captureRectStartFactorX
    y: parent.height * parent.captureRectStartFactorY
}

Component.onCompleted: { camera.active = false; camera.active = true; }
}

QZXingFilter
{
    id: zxingFilter
    videoSink: videoOutput.videoSink
    orientation: videoOutput.orientation

    captureRect: {
        videoOutput.sourceRect;
        return Qt.rect(videoOutput.sourceRect.width * videoOutput.captureRectStartFactorX,
            videoOutput.sourceRect.height * videoOutput.captureRectStartFactorY,
            videoOutput.sourceRect.width * videoOutput.captureRectFactorWidth,
            videoOutput.sourceRect.height * videoOutput.captureRectFactorHeight)
    }

    decoder {
        enabledDecoders: QZXing.DecoderFormat_EAN_13 | QZXing.DecoderFormat_CODE_39 |
            QZXing.DecoderFormat_QR_CODE

        onTagFound: {
            console.log("El codigo QR es" + tag + " con formato " + decoder.foundedFormat() + " | " +
                decoder.charSet());
            window.detectedTags++;
        }
    }
}

```



```

        window.lastTag = tag;

        if(tag == "Lunes"){
            console.log(" Lunes ");
        }
    }

    tryHarder: false
}

onDecodingStarted:
{
    // console.log("started");
}

property int framesDecoded: 0
property real timePerFrameDecode: 0

onDecodingFinished:
{
    timePerFrameDecode = (decodeTime + framesDecoded * timePerFrameDecode) /
(framesDecoded + 1);
    framesDecoded++;
    if(succeeded)
        console.log("frame finished: " + succeeded, decodeTime, timePerFrameDecode,
framesDecoded);
}
}

```

Text

```

{
    id: text2
    wrapMode: Text.Wrap
    font.pixelSize: 20
    anchors.bottom: parent.bottom
    anchors.left: parent.left
    z: 50
    text: "UltimoCodigoLeido: " + lastTag
}

```

```

Switch {
    text: "AutoEnfoque"
    checked: camera.focusMode == Camera.FocusModeAutoNear
    anchors {
        right: parent.right
        bottom: parent.bottom
    }
    onCheckedChanged: {
        if (checked) {
            camera.focusMode = Camera.FocusModeAutoNear

```

```

    } else {
        camera.focusMode = Camera.FocusModeManual
        camera.customFocusPoint = Qt.point(0.5, 0.5)
    }
}
font.family: Qt.platform.os === 'android' ? 'Droid Sans Mono' : 'Monospace'
font.pixelSize: Screen.pixelDensity * 5
}
}

```

El siguiente bloque de código pertenece al código fuente de la aplicación escáner código QR

```

#include "application.h"
#include <QDebug>

#if QT_VERSION < 0x060000
#include "native.h"
#endif

#if defined(Q_OS_ANDROID)

#if QT_VERSION < 0x060100
#include <QAndroidJniObject>
#include <QtAndroid>
#else
#include <QJniObject>
#endif

#endif // Q_OS_ANDROID

Application::Application()
{
    //both signals will be connected to the same function for
    //simplicity

    connect(this, &Application::onPermissionsGranted,
            this, &Application::initializeQML);

    connect(this, &Application::onPermissionsDenied,
            this, &Application::initializeQML);

#if QT_VERSION < 0x060000
    NativeHelpers::registerApplicationInstance(this);
#endif
}

void Application::initializeQML()
{
#if QT_VERSION < 0x060200

```

```

    engine.load(QUrl(QStringLiteral("qrc:/main.qml")));
#else
    engine.load(QUrl(QStringLiteral("qrc:/main_qt6_2.qml")));
#endif // QT_VERSION < 0x060200
}

void Application::checkPermissions()
{
#ifdef Q_OS_ANDROID
    //intentionally called in the C++ thread since it is blocking and will continue after the check
    qDebug() << "About to request permissions";

    #if QT_VERSION < 0x060000
        QAndroidJniObject::callStaticMethod<void>("org/ftylitak/qzxing/Utilities",
            "requestQZXingPermissions",
            "(Landroid/app/Activity;)V",
            QtAndroid::androidActivity().object());
    #else
        emit onPermissionsGranted();
    #endif
    qDebug() << "Permissions granted";
#else
    emit onPermissionsGranted();
#endif //Q_OS_ANDROID
}

```

El siguiente bloque de código contiene configuraciones necesarias, pertenece a el archivo de código fuente principal

```

#include <QGuiApplication>
#include <QQmlApplicationEngine>
#include <QQmlContext>

#include <QDebug>

#include <Qt>
#include "QZXing.h"
#include "application.h"

int main(int argc, char *argv[])
{
    QCoreApplication::setAttribute(Qt::AA_UseHighDpiPixmaps);
    QGuiApplication app(argc, argv);

    QZXing::registerQMLTypes();

    Application customApp;
    customApp.checkPermissions();
    return app.exec();
}

```

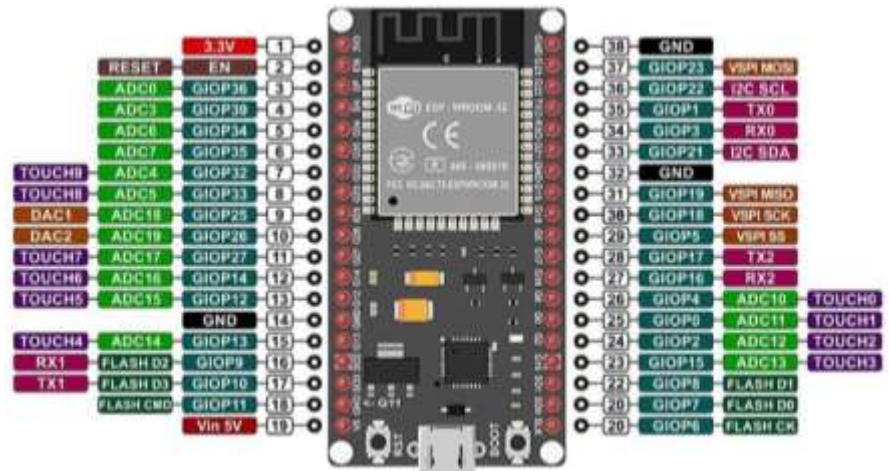
## HARDWARE

### CARACTERISTICAS ELECTRICAS DE LOS COMPONENTES

#### ESP 32

El Esp32 DevKit v1 es una de las placas de desarrollo ESP-WROOM-32. Se basa en el microcontrolador ESP32 que cuenta con compatibilidad con Wi-Fi, Bluetooth, Ethernet y bajo consumo de energía.

El ESP32 ya tiene integrada una antena, amplificador de potencia, amplificadores de bajo ruido, filtros y módulo de administración de energía. Toda la solución ocupa la menor cantidad de área de placa de



circuito impreso. Esta placa se utiliza con chips Bluetooth y Wi-Fi de modo dual de 2,4 GHz de TSMC 40nm tecnología de baja potencia, potencia y propiedades de RF mejores.

#### Memoria Flash

El flash interno del módulo ESP32 está organizado en un solo área de flash con páginas de 4096 bytes cada una. El flash comienza en la dirección 0x00000, pero muchas áreas están reservadas para Esp32 IDF SDK. Existen dos diseños diferentes basados en la presencia de soporte BLE.

#### Consumo

La alimentación del Esp32 DevKit v1 se suministra a través del conector USB Micro B incorporado o directamente a través del pin "VIN". La fuente de alimentación se selecciona automáticamente.

El dispositivo puede operar con un suministro externo de 6 a 20 voltios. Si usa más de 12 V, el regulador de voltaje puede sobrecalentarse y dañar el dispositivo. El rango recomendado es de 7 a 12 voltios.

**Cuenta con** un chip serial a usb integrado que permite programar y abrir la UART del módulo ESP32.

#### ESPECIFICACIONES Y CARACTERISITICAS TECNICAS

Microcontrolador: CPU de 32 bits dos núcleos Xtensa LX6

Voltaje de funcionamiento: 3,3 V.

Voltaje de entrada: 7-12V

Pines Digitales para entradas y salidas: 25

Pines de conversión analógica a digital (ADC): 6

Pines de conversión digital a analógica (DAC): 2

UART: 3

SPI: 2

I2C: 3

Memoria flash: 4 MB

SRAM: 520 KB

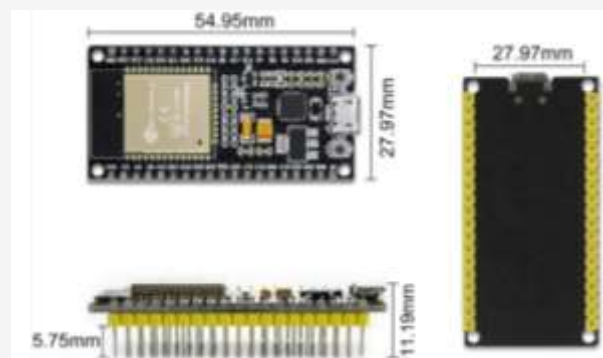
Velocidad máxima de reloj: 240MHz

Wifi con estándar IEEE 802.11 b/g/n/e/i:

Conmutador TR integrado, LNA, amplificador de potencia y red correspondiente

Tipos de encriptación para Autenticación: WEP o WPA/WPA2, o redes abiertas

Dimensiones: 51,5x29x5mm



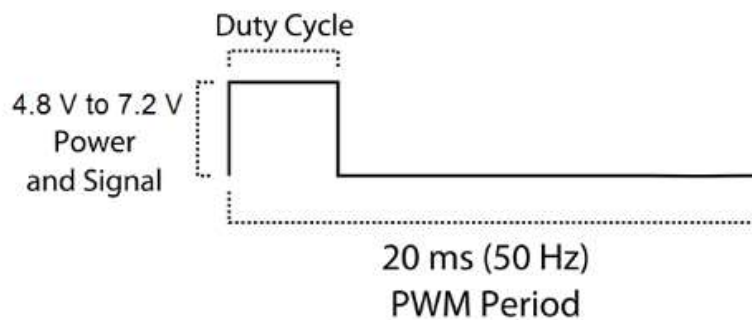
## SERVOMOTOR MG996R

### Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
- Operating speed: 0.17 s/60° (4.8 V), 0.14 s/60° (6 V)
- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA – 900 mA (6V)
- Stall Current 2.5 A (6V)
- Dead band width: 5  $\mu$ s
- Stable and shock proof double ball bearing design
- Temperature range: 0 °C – 55 °C



PWM=Orange (⏏)  
Vcc = Red (+)  
Ground=Brown (–)



### MOTOR REDUCTOR

- **Voltaje de operación: 3V~12VDC (recomendado 6V a 8V)**
- Torque máximo: 800gf cm/min (3V)
- Este motor cumple EMC, capacidad anti-interferencia. Puede operar con microcontrolador sin interferencias.
- Tamaño: 7 x 2.2 x 1.8cm (approx.)



### TRANSFORMADOR

Transformador de 24 Volts 5 Amperes con derivación central. Capacidad de potencia de 120 watts y un voltaje de 12 volts desde derivación central





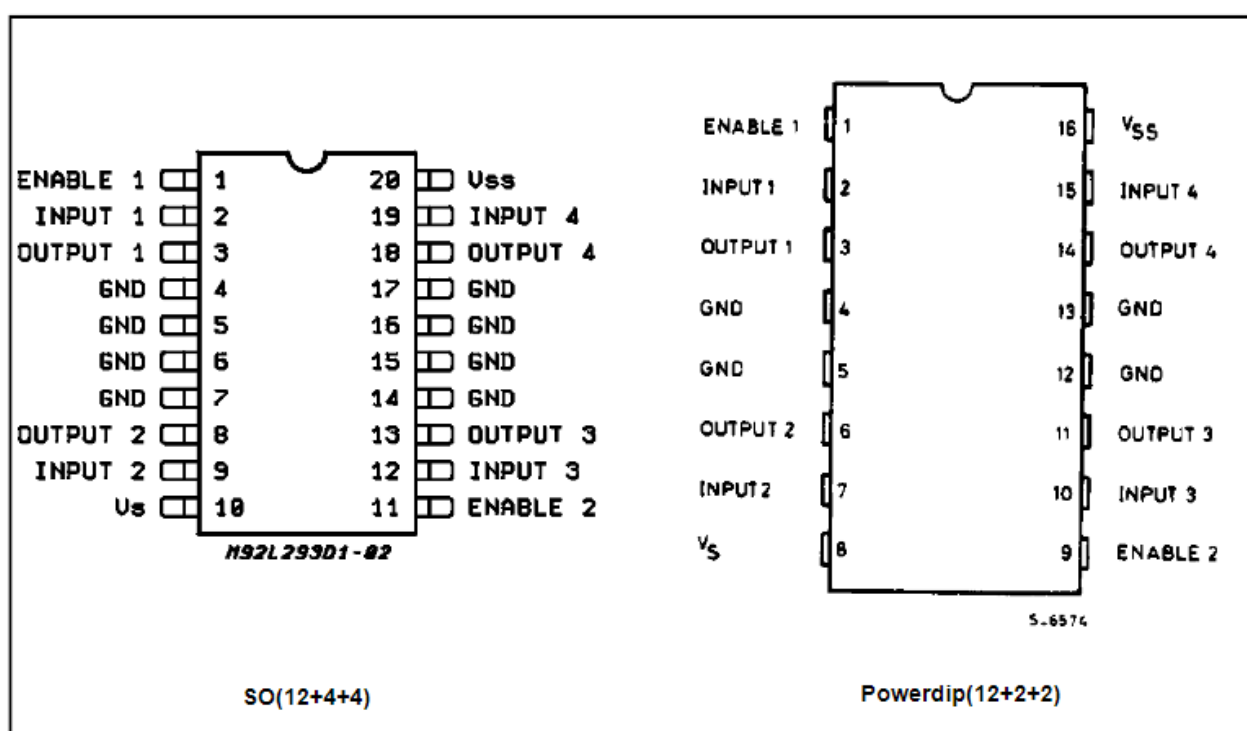
## ENCAPSULADO DIP PUENTE H L293D

### L293D - L293DD

#### ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
$V_S$	Supply Voltage	36	V
$V_{SS}$	Logic Supply Voltage	36	V
$V_i$	Input Voltage	7	V
$V_{en}$	Enable Voltage	7	V
$I_o$	Peak Output Current (100 $\mu$ s non repetitive)	1.2	A
$P_{tot}$	Total Power Dissipation at $T_{pins} = 90^\circ\text{C}$	4	W
$T_{stg}, T_j$	Storage and Junction Temperature	- 40 to 150	$^\circ\text{C}$

#### PIN CONNECTIONS (Top view)



## TRANSISTOR PNP TIP 42 C

### COMPLEMENTARY SILICON PLASTIC POWER TRANSISTORS

... designed for use in general purpose power amplifier and switching applications.

#### FEATURES:

- \* Collector-Emitter Sustaining Voltage -  
 $V_{CE(sus)}$  = 40V(Min)- TIP41, TIP42  
 60V(Min)- TIP41A, TIP42A  
 80V(Min)- TIP41B, TIP42B  
 100V(Min)-TIP41C, TIP42C
- \* Collector-Emitter Saturation Voltage-  $V_{CE(sat)}$  = 1.5V(Max) @  $I_C = 6.0A$
- \* Current Gain-Bandwidth Product  $f_T$  = 3.0 MHz (Min) @  $I_C = 500mA$

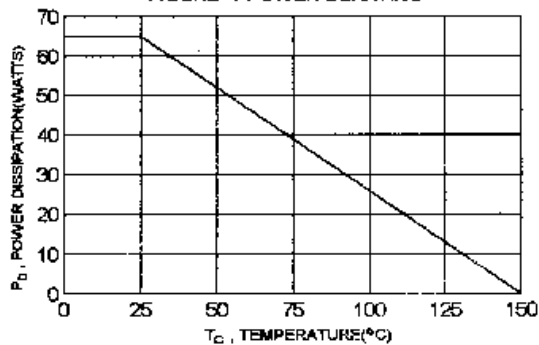
#### MAXIMUM RATINGS

Characteristic	Symbol	TIP41 TIP42	TIP41A TIP42A	TIP41B TIP42B	TIP41C TIP42C	Unit
Collector-Emitter Voltage	$V_{CEO}$	40	60	80	100	V
Collector-Base Voltage	$V_{CBO}$	40	60	80	100	V
Emitter-Base Voltage	$V_{EBO}$	5				V
Collector Current - Continuous - Peak	$I_C$	6 10				A
Base Current	$I_B$	2				A
Total Power Dissipation @ $T_C = 25^\circ C$ Derate above $25^\circ C$	$P_D$	65 0.52				W W/ $^\circ C$
Operating and Storage Junction Temperature Range	$T_J, T_{STG}$	-65 to +150				$^\circ C$

#### THERMAL CHARACTERISTICS

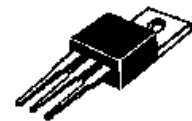
Characteristic	Symbol	Max	Unit
Thermal Resistance Junction to Case	$R_{\theta jc}$	1.92	$^\circ C/W$

FIGURE -1 POWER DERATING

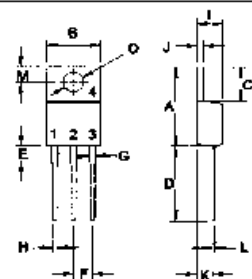


NPN	PNP
TIP41	TIP42
TIP41A	TIP42A
TIP41B	TIP42B
TIP41C	TIP42C

**6 AMPERE  
COMPLEMENTARY SILICON  
POWER TRANSISTORS  
40 -100 VOLTS  
65 WATTS**



TO-220



PIN 1, BASE  
2, COLLECTOR  
3, EMITTER  
4, COLLECTOR (CASE)

DIM	MILLIMETERS	
	MIN	MAX
A	14.68	15.31
B	9.78	10.42
C	5.01	6.52
D	13.06	14.82
E	3.57	4.07
F	2.42	3.66
G	1.12	1.36
H	0.72	0.96
I	4.22	4.88
J	1.14	1.38
K	2.20	2.97
L	0.33	0.55
M	2.48	2.98
O	3.70	3.90

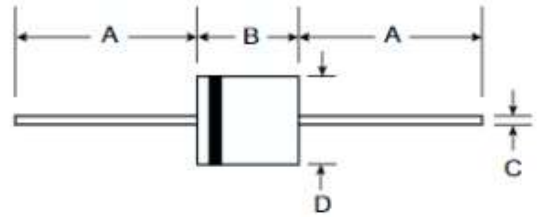
## DIODO 6A8

### Features

- High Surge Current Capability
- Low Leakage and Forward Voltage Drop

### Mechanical Data

- Case: R-6, Molded Plastic
- Terminals: Axial Leads, Solderable per MIL-STD-202, Method 208
- Polarity: Color Band Indicates Cathode
- Approx. Weight: 2.1 grams
- Plastic Material - UL Flammability Classification 94V-0



R-6		
Dim	Min	Max
A	25.40	—
B	8.60	9.10
C	1.20	1.30
D	8.60	9.10
All Dimensions in mm		

### Maximum Ratings and Electrical Characteristics @25°C unless otherwise specified

Ratings at 25°C ambient temperature unless otherwise specified.  
Single phase, halfwave, 60Hz, resistive or inductive load.

Characteristic	Symbol	6A05	6A1	6A2	6A4	6A6	6A8	6A10	Unit
Maximum Recurrent Peak Reverse Voltage	$V_{RRM}$	50	100	200	400	600	800	1000	V
Maximum RMS Voltage	$V_{RMS}$	35	70	140	280	420	560	700	V
Maximum DC Blocking Voltage	$V_{DC}$	50	100	200	400	600	800	1000	V
Maximum Average Forward Rectified Current 9.5mm lead length @ $T_A = 75^\circ\text{C}$ (See Fig. 1)	$I_{(AV)}$	6.0							A
Peak Forward Surge Current 8.3 ms single half sine-wave superimposed on rated load (JEDEC method)	$I_{FSM}$	400							A
Maximum Instantaneous Forward Current at 6.0A DC	$V_F$	0.90							V
Maximum DC Reverse Current @ $T_A = 25^\circ\text{C}$ at Rated Blocking Voltage @ $T_A = 100^\circ\text{C}$	$I_R$	10 100							$\mu\text{A}$
Operating and Storage Temperature Range	$T_J$ , $T_{STG}$	-65 to +175							$^\circ\text{C}$

## REGULADOR DE VOLTAJE CV7805 y CV7806

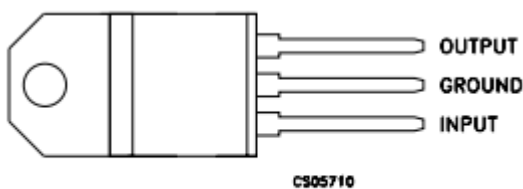
### ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter <sup>2</sup>		Value	Unit
V <sub>I</sub>	DC Input Voltage	for V <sub>O</sub> = 5 to 18V	35	V
		for V <sub>O</sub> = 20, 24V	40	
I <sub>O</sub>	Output Current		Internally Limited	
P <sub>tot</sub>	Power Dissipation		Internally Limited	
T <sub>stg</sub>	Storage Temperature Range		-65 to 150	°C
T <sub>op</sub>	Operating Junction Temperature Range	for L7800	-55 to 150	°C
		for L7800C	0 to 150	

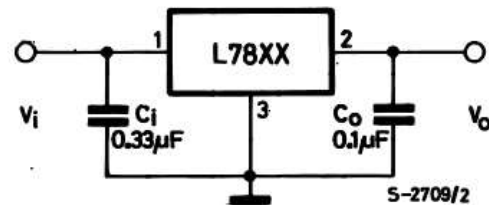
Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.

### THERMAL DATA

Symbol	Parameter	D <sup>2</sup> PAK	TO-220	TO-220FP	TO-3	Unit
$R_{thj-case}$	Thermal Resistance Junction-case Max	3	5	5	4	°C/W
$R_{thj-amb}$	Thermal Resistance Junction-ambient Max	62.5	50	60	35	°C/W



TO-220



### IMPLEMENTACION

Se realizo el diseño adecuado de una fuente de poder en una pcb, las pistas y las soldadura son realizadas a mano, para las pistas se utilizo una cinta conductora de cobre para reducir el uso de estaño para las pistas entre componentes la fuente cuenta con electronica de potencia para suministrar suficiente energia a los 7 motores, se realizaron las secuencias de acomodo de paquetes dependiendo el dia de registro que decodifique el lector de codigos QR, se comunican via serial y via wifi con conexión a internet por media del celular a la red de datos telefonica, la interface envia el mensaje JSON el cual contiene la informacion a decodificar por el ESP 32 para porsteriormete ejecutar la instruccion asignada.

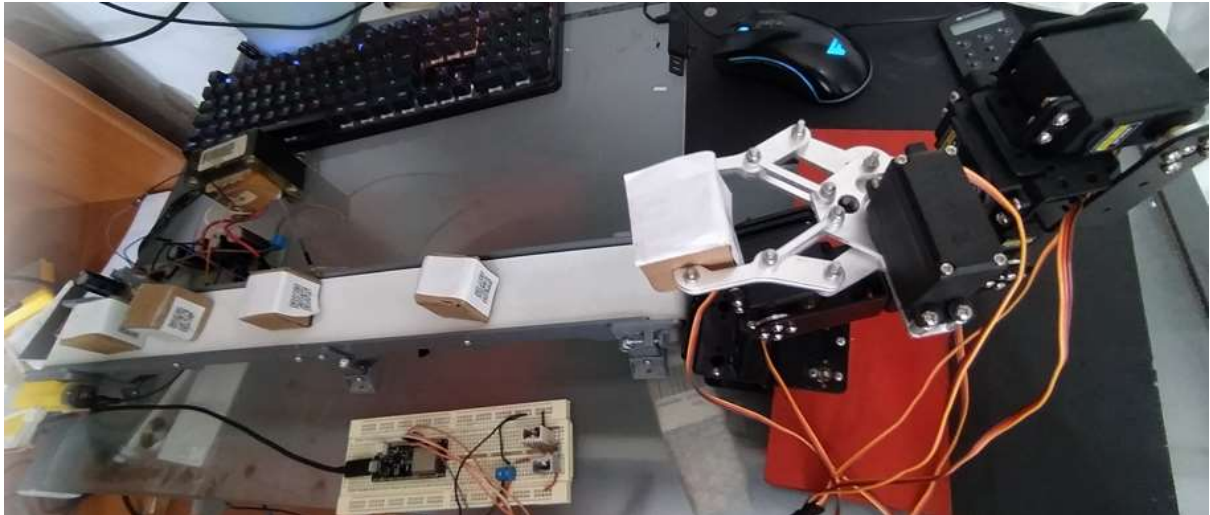
El dia 5 de mayo del año 2022 se presento el proyecto de forma adecuada a alumnos, profesores, evaludaores y publico en general interesado, en la exposicion de proyectos modulares en el laboratorio N de CUCEI por el alumno:

Carlos Adolfo Aguilar Rodriguez con el codigo de alumno 215860049

y el apoyo del asesor de proyecto el Doctor

David Israel Rodriguez Castellanos.con el codigo de profesor 2959101





## CONCLUSIONES

Se desarrollo de manera adecuada y con buenos resultados la primera aproximación del prototipo del proyecto *Brazo robótico clasificador de paquetes*”.

El sistema clasifico el paquete en la banda transportadora mediante el reconocimiento del código QR y genero las instrucciones para que el brazo robótico colocara dicho paquete de manera adecuada.

Es esencial utilizar pasta térmica y disipadores de calor en los reguladores de voltaje y en el transistor, al tener una alta demanda de corriente es importante tener buena capacidad de disipación.

Se sugiere tener una buena base que soporte los movimientos del brazo robótico, la base del brazo robótico cuenta con orificios para poder colocar pijas a una base de metal, concreto o madera. Esto proporcionara mayor firmeza y estabilidad a los movimientos del brazo.

## TRABAJOS FUTUROS

Implementar Detección de objetos con open cv para detectar la posición de la caja para determinar que instrucciones debe ejecutar la banda transportadora por ejemplo avanzar o retroceder para lograr entrar en el rango de operación del brazo robótico, se pretende mandar a fabricar el pcb para montar el circuito diseñado.



Crear la consulta de las bases de datos desde una página web, agregar y modificar la base de datos.  
Crear aplicación Android que permita el control de la interfaz desde un celular.  
Una vez implementada la visión artificial se pretende que si el paquete no contiene en la cara que viene de frente a la cámara el código QR el brazo robótico busque la cara que contiene el código QR, si no lo contiene generar un código QR donde se creara el registro del ingreso del producto con fecha y hora en una base de datos. De la misma forma con visión artificial es que si por alguna razón el paquete se encuentra afuera de la banda transportadora, el paquete mediante reconocimiento de objetos pueda agarrar el paquete independientemente si se encuentra dentro de la banda transportadora o no.

## BIBLIOGRAFÍA

Análisis básico de circuitos eléctricos y electrónicos

**Autor: Txelo Ruiz Vázquez**

Electrónica: Teoría de circuitos y dispositivos electrónicos

Robert L. Boylestad

Fundamentos de electrónica analógica

Gustavo Camps

Sistemas de comunicaciones electrónicas (4ª edición)

Tomasi

OGATA: INGENIERÍA DE CONTROL MODERNA, 3a. Ed.

Análisis de circuitos en ingeniería

Libro de William H. Hayt

## Páginas web consultadas

<https://morioh.com/p/6c308bf1be60>

<https://programmerclick.com/article/51391155104/>

<https://stackoverflow.com/questions/42979325/drawing-qr-code-with-qt-in-native-c-c>

<https://learnopencv.com/barcode-and-qr-code-scanner-using-zbar-and-opencv/>

<https://github.com/NaturalHistoryMuseum/pyzbar>

[https://www.hackster.io/Arnov\\_Sharma\\_makes/l293d-with-esp32-wemos-lolin-d32-v2-hacked-edition-ea2086](https://www.hackster.io/Arnov_Sharma_makes/l293d-with-esp32-wemos-lolin-d32-v2-hacked-edition-ea2086)

<https://morioh.com/p/6c308bf1be60>

<https://towardsdatascience.com/build-your-own-barcode-and-qr-code-scanner-using-python-8b46971e719e>

<https://www.dynamsoft.com/codepool/qt-desktop-barcode-reader.html>



<https://towardsdatascience.com/build-your-own-barcode-and-qr-code-scanner-using-python-8b46971e719e>

<https://esp32io.com/tutorials/esp32-servo-motor>

<https://www.the-qr-code-generator.com/es/>

[https://www.youtube.com/watch?v=SrZuwM705yE&ab\\_channel=Murtaza%27sWorkshop-RoboticsandAI](https://www.youtube.com/watch?v=SrZuwM705yE&ab_channel=Murtaza%27sWorkshop-RoboticsandAI)

<https://pdf1.alldatasheet.com/datasheet-pdf/view/1131873/ETC2/MG996R.html>

<https://ubuntu.com/server/docs/service-ftp>

<https://grobotronics.com/esp32-development-board-devkit-v1.html>

<https://scythe-studio.com/en/blog/how-to-scan-barcodes-in-qt-qml-application%22%20>