

MENDOZA
FUTURA



MENDOZA GOBIERNO
Ministerio de Economía y Energía

GUÍA DE ACOMPAÑAMIENTO PARA FACILITADORES

EPISODIO 4

Habilidades Tecnológicas

MENDOZA, 2023

HABILIDADES ESPERABLES	3
ESTACIÓN 1	5
CONCEPTOS	5
Videos de referencia	6
NOTA:	10
Videos de referencia	11
ACTIVIDADES	12
1. ¿Qué es una variable byte?	12
2. ¿Qué es un motor de 4 pasos?	12
3. ¿Cuál es la diferencia entre un motor paso a paso y un servomotor?	12
4. Explique en forma coloquial el código propuesto.	12
5. Habría otra forma de mover el motor.	12
ESTACIÓN 2	13
BRAZOS ROBÓTICOS	13
Brazos robóticos Universal Robots	14
TIPOS DE BRAZOS MECÁNICOS ROBÓTICOS	14
Proyecto A: Brazo robótico casero	16
Videos de referencia	24
ACTIVIDADES	24
ESTACIÓN 3	25
DOMÓTICA	25
¿Domótica cableada o inalámbrica?	26
Videos de referencia	28
¿Qué es MQTT?	29
Datos de acceso	31
REFERENCIAS DE TRABAJO	32
WEBGRAFÍA	32
● ESP32 puesta en marcha y consideraciones sobre sus pines	32
○ Botones dentro TelegramBot en ESP32 y ESP8266:	
https://www.youtube.com/watch?v=5U-I_12QcQ0&list=PLWQQswW6kqpXuTWwCG79BZYfeZS-Kqp-i&index=4	32
ACTIVIDADES:	33

CONTENIDOS A DESARROLLAR

MOTORES

- Control de motores: corriente continua y paso a paso. (ver Tinkercad)
- Con Transistores
- Con módulos

BRAZOS ROBÓTICOS

- Componentes
- Grados de libertad
- Control con Joysticks

DOMÓTICA

- Control de corriente alterna con relé (ver Tinkercad)
- IOT

DESAFÍOS PYTHON

- Tratamiento de datos
- Librerías
- Resolución de situaciones problemáticas



HABILIDADES ESPERABLES

- Amplía la capacidad de abstracción mediante procesos de análisis y síntesis
- Desarrolla el pensamiento lógico a través de estructuras de programación
- Potencia el pensamiento crítico y las habilidades de liderazgo por la dinámica de trabajo grupal
- Despierta el espíritu colaborativo gracias a la estrategia de las competencias.
- Estimula la creatividad mediante el diseño y la resolución de problemas.
- Desarrolla capacidades de expresión oral y escrita
- Fomenta la integración y el respeto mediante desafíos grupales
- Estimula de forma lúdica el interés por las ciencias y las tecnologías

ESTACIÓN 1

CONCEPTOS

Uno de los objetivos principales en robótica es hacer que las cosas se muevan por sí mismas. Para dar movimiento a un robot, se utilizan motores eléctricos como los motores paso a paso, especialmente cuando se requiere precisión con un control de bucle abierto.

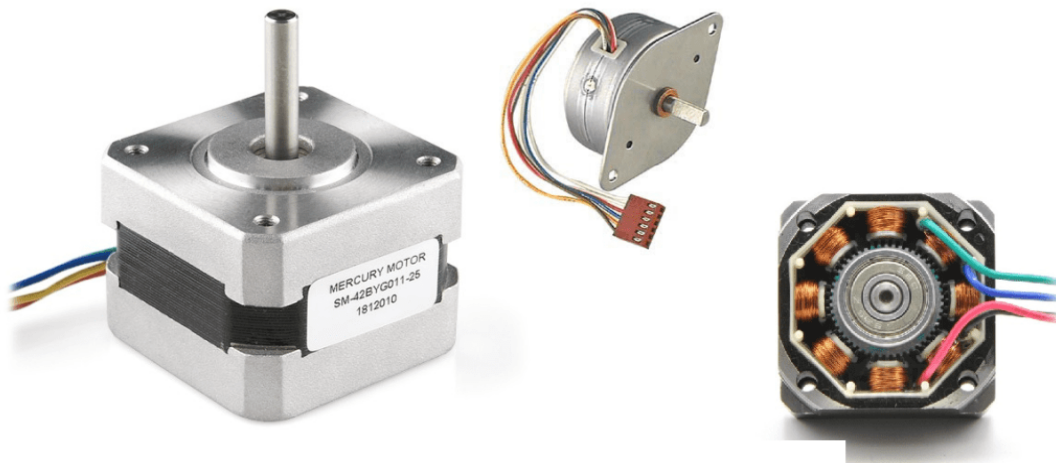
Un motor paso a paso o también conocidos como “stepper motor” es un dispositivo electrónico que permite efectuar un movimiento muy preciso en ángulos pequeños y por pasos en ambas direcciones, por lo tanto este dispositivo es usado en infinidad de procesos de precisión como CNC, impresoras 3D, robots, fotocopadoras, cortadoras láser entre otros.

Los motores paso a paso son motores de CC que se mueven en pasos discretos. Tienen múltiples bobinas que se organizan en grupos llamados «fases». Al energizar cada fase en secuencia, el motor rotará, un paso a la vez.

Los pasos del motor son controlados por una computadora o un microcontrolador, pudiendo lograr un posicionamiento muy preciso y / o controlar la velocidad del motor paso a paso. Por esta razón, los motores paso a paso son el motor elegido para muchas aplicaciones de control de movimiento de precisión.

Los motores paso a paso vienen en diferentes tamaños y estilos y características eléctricas. Esta guía detalla lo que necesita saber para elegir el motor adecuado para su trabajo.

Básicamente un motor paso a paso puede ser representado por la siguiente figura:



Como puede ser observado en la figura anterior y como ya fue mencionado, el motor paso a paso posee diferentes cables «fases» y es porque internamente dicho motor posee varias bobinas, las cuales deben ser energizadas en una secuencia ordenada para poder conseguir el movimiento del motor.

Videos de referencia

Motores paso a paso:

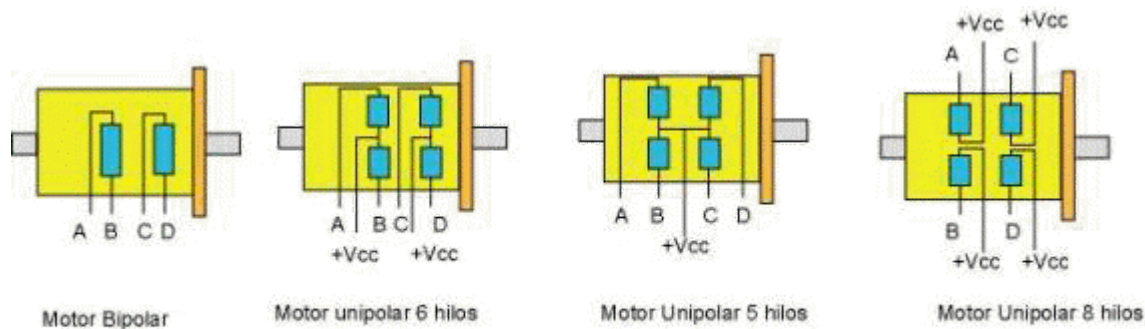
- Parte 1: <https://www.youtube.com/watch?v=MmSbAcKlyCQ>
- Parte 2: <https://www.youtube.com/watch?v=pbRIACowwYk>

Tipos de Motores Paso a Paso

Pero antes es importante entender que en el mercado existen dos tipos de motores paso a paso los cuales podremos controlar con Arduino.

Bobinas del motor paso a paso

A continuación podemos ver un esquema que muestra la configuración de las diferentes bobinas dentro de un motor paso a paso unipolar y bipolar.



Estos motores de paso pueden ser encontrados en diferentes dispositivos comunes tales como las impresoras y muchas veces puede resultar especialmente útil poder reconocer cada uno de los terminales del motor, por lo tanto prepararé el siguiente video explicándote cómo reconocer fácilmente las bobinas y los cables de un motor paso a paso para controlarlo posteriormente con un Arduino.

Motor Paso a Paso Unipolar con Arduino

Este es el motor de paso más común, el cual posee dos bobinas en cada uno de los estatores donde cada una de esas bobinas posee un punto común, por lo tanto este motor típicamente posee 5 o 6 cables. Y viendo esta entrada estarás en la capacidad de controlar un motor paso a paso de 6 cables con Arduino.

Los motores unipolares, siempre energizan sus fases de la misma forma. La derivación «común», siempre será negativa o positiva según nuestro criterio. La otra punta de la derivación contendrá la polarización contraria. Los motores unipolares se pueden implementar con un simple circuito de transistor. La desventaja es que hay menos torque disponible porque solo la mitad de las bobinas se pueden energizar a la vez.

Motor Paso a Paso Bipolar con Arduino

También posee dos bobinas, con la diferencia que no tienen un punto intermedio común, por lo tanto, es fácil intuir que posee 4 cables. Sin embargo, para conseguir el control de este motor bipolar será necesario efectuar una configuración de 2 puentes H usando 8 transistores, o bien usando algún driver o integrado especial para poder mandar la secuencia de pasos.

Los motores bipolares necesitan los circuitos de puente H porque es necesario invertir el flujo de corriente a través de las fases. Al energizar las fases alternando la polaridad, todas las bobinas pueden ponerse a trabajar girando el motor.

Movimiento del Motor paso a Paso con Arduino o cualquier Microcontrolador

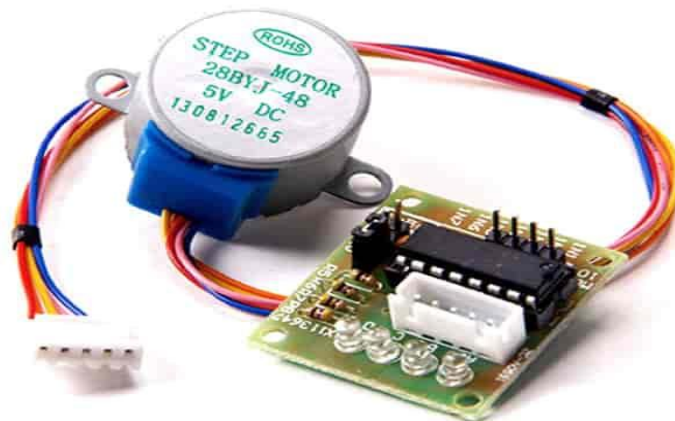
Para conseguir el movimiento de un motor paso a paso con Arduino o cualquier otro microcontrolador, necesariamente vamos a tener que recurrir a la ayuda de un circuito externo, debido a que un sistema microcontrolado no cuenta con la corriente suficiente para mover este dispositivo con carga.

En el mercado existen varios drivers diseñados para este fin tales como el ULN2003, L298, L296 entre otros, o también podríamos pensar en controlar un motor paso a paso con Arduino sin Driver a través de 4 transistores (unipolar) para activar cada bobina o 8 transistores (bipolar) para hacer una configuración de 2 Puente H que dirija la corriente dentro del motor.

A través de estos dispositivos mandaremos la secuencia en orden para poder mover el motor paso a paso.

Usando un Motor Paso a Paso 28BYJ-48 con ULN2003 y Arduino

Para esta práctica del Motor paso a paso con Arduino vamos a usar el popular motor 28BYJ-48 (datasheet) con el driver ULN2003 controlado con Arduino. Por lo tanto esta práctica consistirá en un motor paso a paso unipolar con Arduino.



El motor 28BYJ-48 es un motor paso a paso Unipolar, por lo tanto no necesita de un complejo sistema de driver para ser controlado por eso haremos uso del ULN2003 el cual es una agrupación de 7 darlingtons que sirven para proporcionarle la corriente necesaria al motor para moverse.

Según el datasheet, cuando el motor 28BYJ-48 funciona con una secuencia a medio paso, cada medio paso corresponde a una rotación de 5.625° . Eso significa que hay 64 impulsos por revolución ($360^\circ / 5.625^\circ = 64$).

Además, el motor posee una reducción de $1/64$. (En realidad es $1 / 63.68395$ pero a hacer un redondeo para $1/64$ que es una buena aproximación)

Esto significa que en realidad hay que dar $64 * 63.68395$ pasos por revolución = $4,075.7728 \sim 4076$ pasos por vuelta.

El motor 28BYJ-48 es más un motor de precisión que de velocidad. La velocidad máxima para un motor paso a paso 28BYJ-48 es aproximadamente 10-15 rpm a 5 V.

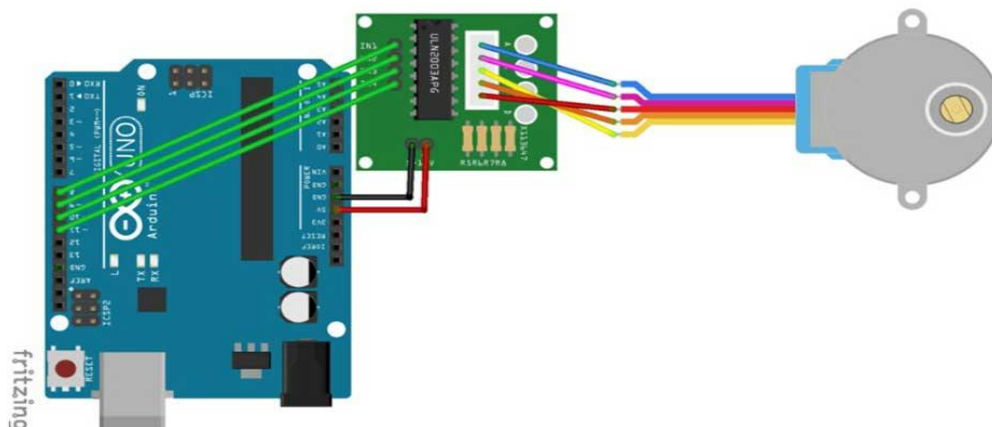
Ejemplo Motor PaP con Arduino UNL2003

Hagamos un ejemplo de un motor paso a paso unipolar arduino.

Crear un programa en arduino usando el driver UNL2003 y el Motor paso a paso 28BYJ-48 que sea capaz de detectar el cambio en 4 pulsadores. Un pulsador servirá para hacer el giro sentido horario, otro pulsador hará el giro antihorario, otro pulsador cambiará la secuencia de pasos para 1, 2 o medio paso y el último pulsador modificará la velocidad del motor.

El esquema es representado a continuación, donde se muestra cómo conectar un motor

paso a paso en Arduino:



NOTA:

Motor paso a paso arduino L293D y L298 Unipolar

Note que el ejemplo anterior del Motor Paso a Paso con Arduino Unipolar también puede ser implementado con drivers comerciales como el L293D y el Driver L298, que internamente poseen configuraciones en Puente H. Por lo que puedes usar exactamente el mismo código, cambiando el driver que tengas en tu poder.

También puede implementarse el control del motor paso a paso con arduino sin driver utilizando para ello un arreglo de 4 transistores que activen cada una de las bobinas del motor UNIPOLAR.

Motor Paso a Paso Bipolar con Arduino

El ejemplo anterior también puede ser utilizado con Motores Paso a Paso Bipolares, sin embargo el control de este tipo de motor puede resultar un poco más complejo, en el caso que se quiera implementar con un arreglo de 8 transistores para hacer 2 puente H.

A diferencia del motor paso a paso unipolar, el motor paso a paso bipolar tiene dos conductores por fase, ninguno de los cuales es común.

Motor Paso a Paso UNIPOLAR y BIPOLAR con Arduino

A continuación se muestra el esquema electrónico del ejemplo, que muestra la conexión para motores paso a paso unipolares y bipolares usando cualquiera de los drivers anteriormente mencionados o inclusive vamos a conectar un motor paso a

paso a Arduino sin usar DRIVER y lo más importante es que todos los esquemas usan el MISMO CÓDIGO DE PROGRAMACIÓN DEL ARDUINO.

PASO	A	B	C	D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Videos de referencia

- Paso a paso con Arduino y controlador L293D::
<https://www.youtube.com/watch?v=wXMIwPAliBY>
- Paso a paso con L298N y otros:
https://www.youtube.com/watch?v=WhsosF52K_I

CÓDIGO

```
/*
  Creado: Mendoza Futura
  Movimiento por ola motor 28BYJ-48
*/
// Definimos los pines donde tenemos conectadas las bobinas
#define IN1 8
#define IN2 9
#define IN3 10
#define IN4 11
// Secuencia de pasos (par máximo)
int paso [4][4] =
{
  {1, 0, 0, 0},
  {0, 1, 0, 0},
  {0, 0, 1, 0},
  {0, 0, 0, 1}
};
void setup()
{
  // Todos los pines en modo salida
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}
void loop()
{
  for (int i = 0; i < 4; i++)
  {
    digitalWrite(IN1, paso[i][0]);
    digitalWrite(IN2, paso[i][1]);
    digitalWrite(IN3, paso[i][2]);
    digitalWrite(IN4, paso[i][3]);
    delay(10);
  }
}
```

ACTIVIDADES

1. ¿Qué es una variable byte?
2. ¿Qué es un motor de 4 pasos?
3. ¿Cuál es la diferencia entre un motor paso a paso y un servomotor?
4. Explique en forma coloquial el código propuesto.
5. ¿Habría otra forma de mover el motor?
6. Programar en Tinkercad el movimiento de un motor paso a paso

ESTACIÓN 2

BRAZOS ROBÓTICOS

La Robótica Industrial y sus desarrollos a lo largo del siglo XX y XXI, han ido creando y conformando un conjunto de instrumental que ha permitido lograr importantes avances. Los brazos robóticos industriales sin duda se constituyen en uno de los hitos fundamentales en comprender los aportes que ha logrado realizar la Robótica Industrial en distintas áreas.

¿QUÉ ES UN BRAZO ROBÓTICO?

En esencia la definición corta de un brazo robótico son «robots articulados» pues un brazo mecánico es construido y elaborado con estructuras de carácter flexible y conformadas por articulaciones adaptables que permiten ejecutar un amplio rango de movimientos y funciones simulando a un brazo humano.

Tienen como principal característica el cumplir la funciones de agilizar actividades, tareas y funciones que requieren una actividad de repetitividad y precisión con entornos humanos seguros; durante los procesos de automatización de líneas de producción y manipulación de máquinas en diferentes industrias: Mecánica, Minería, Automovilística, Farmacéutica, Alimenticia etc.

Los materiales con los que son fabricados los brazos mecánicos para el sector industrial principalmente son: aluminio, fibra de carbono, acero y titanio.

Partes para el funcionamiento de un brazo robótico

Las partes o componentes que conforman un brazo robótico según sus principales actividades de funcionamiento son:

- **Controlador:** Se constituye por un microordenador con unidad central encargado de calcular los procesos, movimientos y comandos a ejecutar.
- **Actuadores:** Corresponden a los motores encargados de generar fuerza para los movimientos.
- **Manipulador:** Parte mecánica que cumple la función de realizar los movimientos de los brazos mecánicos.
- **Articulaciones y Muñecas:** Subpartes de los manipuladores, y permiten generar los movimientos lineales y angulares.
- **Muñeca:** Es una parte de los robots industriales encargada de realizar los movimientos de elevación, desviación y giro, según el tipo de funciones.

- Mano robótica o Gripper: Es la extensión final de la muñeca y cumple la función de la tarea de movimiento mecánico final, según sea el componente que corresponde a una pinza o alguna herramienta que cumpla las funciones de las condiciones de trabajo.

Brazos robóticos Universal Robots



TIPOS DE BRAZOS MECÁNICOS ROBÓTICOS

Actualmente, existe un crecimiento bastante importante de diferentes modelos de brazos robóticos que han sido incorporados a diversas aplicaciones en distintas industrias.

Su uso a distintos niveles de actividades siempre han estado identificados en sus diversas capacidades según el número de articulaciones que lo conforman, las configuraciones que implican los niveles de movimiento, sus capacidades de posicionamiento, el grado de exactitud al realizar diversas tareas programadas, las capacidades de carga, sin dejar a un lado los aspectos asociados a la cantidad de peso que pueden movilizar o transportar, y finalmente la velocidad, relacionado al desplazamiento en que pueden ejecutar sus actividades.

Al mencionar este aspecto podemos indicar que son varias las industrias donde pueden incorporarse. Las aplicaciones de los brazos robóticos tienen mayor relevancia en el sector manufactura, minero y alimenticio. A continuación enlistamos los de mayor relevancia:

Brazo robótico automatizado (Automotriz).

Se usan los robots de tipo cartesiano, los cuales tiene 3 ejes y cumplen con las funciones de soldadura, ensamblaje, pintura y atornillado garantizando repetitividad y bajo coste económico.

Ejemplo de este tipo de brazos robóticos son los de la compañía Universal Robots, con su modelos UR3e.

Brazo mecánico eléctrico.

Tienen un amplio dominio de los brazos robóticos tipo “Scara”, los cuales proporcionan un eje final en el plano Z, que permiten hacer movimientos con cualquier o determinada herramienta, permiten el traslado y levantamiento de objetos.

Robot para la Industria Minera.

Tiene predominancia de los robots articulados de 6 ejes, pues permiten realizar numerosas actividades de traslado de materiales pesados, laminación y ensamble de algunos procesos de fundición de metales.

Robot para la Industria de Alimentos.

Existe una notable variedad de brazos mecánicos útiles para este tipo de industria, destacan los robots paralelos que cumplen diversas funciones de descarga, paletizado y distribución de objetos en una línea de producción.

Brazo Robótico Soldador.

Son brazos robóticos diseñados y optimizados para garantizar eficiencia y rentabilidad en los procesos de soldaduras con láser, gas o de otro tipo. Son compactos y trabajan a velocidad elevada, se caracterizan por su flexibilidad y muñeca central final para trabajar en lugares precisos al soldar.

Brazo Robótico para Armar.

Son un tipo de brazos robóticos industriales que permiten generar soluciones viables y polivalentes, pues generan lograr una automatización mediante tareas continuas que impliquen movilizar objetos de distintos pesos combinados con precisión y realizar montajes en diversos ángulos y posiciones.

Estos robots de ensamblaje son resistentes al calor, polvo, agua y generalmente son una solución bastante económica para muchas empresas.

Brazo Robótico para aplicación de pintura.

Están contruidos para brindar soluciones en procesos de automatización que impliquen equilibrar velocidad y rendimiento en aplicaciones de pintura en distintas industrias, son muy útiles para optimizar los márgenes de rentabilidad, tienen un balance entre capacidad de carga y alcance con ciclos de aceleración que ofrecen mejor desempeño.

Brazos Robóticos para Transporte de Material.

Son un tipo de brazo robótico ideal para lograr traslado de materiales pesados superiores a los 100 Kg, tienen un montaje flexible, pueden trabajar a distintas temperaturas, garantizan una actividad sostenida y rentable, pues se dedican a una tarea repetitiva con costos relativamente reducidos.

Un ejemplo de este tipo de robots son los de la marca MIR (Mobile Industrial Robots) con sus modelos MIR100 y MIRHook.

Proyecto A: Brazo robótico casero

Circuito: Armando un brazo robótico casero

Con el propósito de demostrar que podemos realizar muchísimos proyectos utilizando Arduino sin la necesidad de utilizar materiales caros o difíciles de conseguir, en este tutorial diseñaremos y construiremos un brazo robótico utilizando materiales reciclables y para controlarlo utilizaremos 3 potenciómetros que funcionaran como joysticks para controlar el movimiento de cada una de las articulaciones del brazo.

Para la serie de proyectos DIY desde casa, estamos utilizando este kit. Podrás contar con todos los componentes necesarios para todas las prácticas de esta serie si adquieres el kit. De lo contrario, podrás realizar las prácticas con la lista de materiales a continuación:

Materiales:

- (1) Arduino Uno R3
- (1) Cable USB para programar
- (3) Potenciómetro de 10k ohms
- (1) Cartón o material reciclado resistente.
- (3) Servomotor (utilizando el SG90)
- (1) Pegamento
- (1) Batería 9V con portapilas
- (15) Jumpers (cables) macho-macho

Paso 1:

Para construir nuestro brazo lo primero es armar la base, para ello necesitaremos cuadrados de aproximadamente 15 cm por lado. Necesitaremos cuatro a los cuales en el centro les marcaremos y recortaremos un rectángulo del tamaño de uno de los servomotores que estaremos utilizando.



Fig. 1



Fig. 2

Ahora insertamos el servomotor por todos los cuadrados y pegaremos todos estos entre sí hasta tener el servo motor completamente firme dentro de la base.

NOTA: Para mantener la estabilidad de la base en caso de ser necesario por el peso del brazo podemos agregar peso en las esquinas con algún objeto. El caso ideal sería utilizar una base de un material más resistente y pesado como la madera.



Fig.3

Para la construcción del brazo y articulaciones agregaremos rectángulos de cartón de las medidas que queramos (hay que tomar en cuenta el peso del cartón que estemos utilizando ya que un brazo muy largo no tendrá la estabilidad suficiente como para realizar movimientos precisos). El brazo puede tener la estructura que queramos, lo que nosotros sugerimos es utilizar un rectángulo como altura pegado el primer servo de la base el cual nos dará rotaciones desde la base, el siguiente servomotor que pegaremos será el del movimiento del antebrazo, al igual que el anterior hay que asegurarnos de la estabilidad de todo el brazo, para ello la recomendación es utilizar un rectángulo de más chico al de la parte del brazo que está pegada a la base, y por último, pegaremos la mano. Al final debería quedarnos una estructura como la que se muestra en la figura 5.

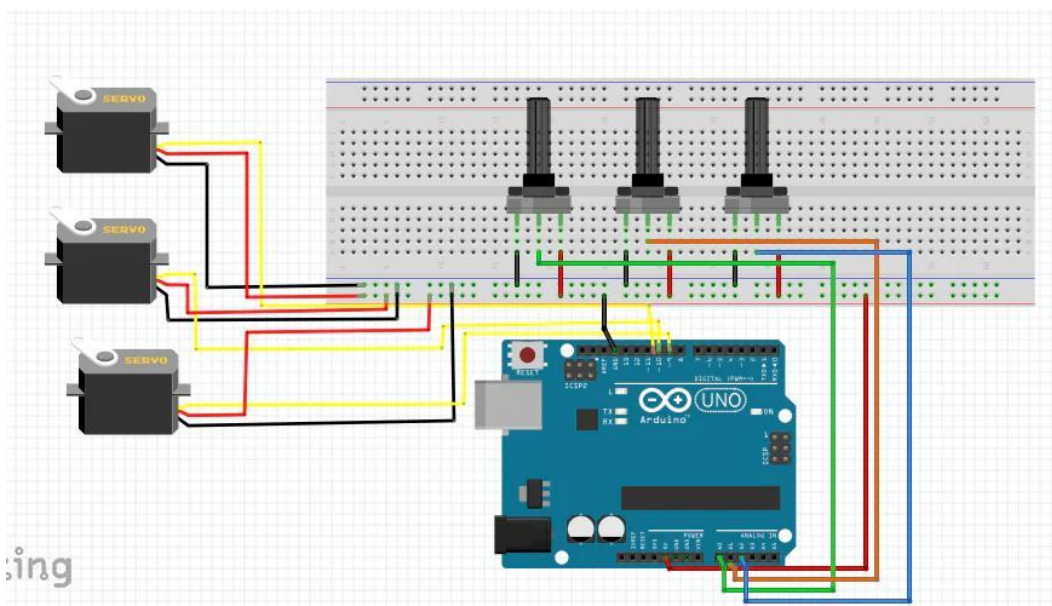
NOTA: Hay que asegurarse que los servomotores están muy bien pegados y que cada articulación es lo suficientemente resistente para mover la siguiente articulación u objeto.



Figura 5: Brazo robótico armado.

Paso 2:

Ya que tenemos nuestro brazo listo necesitamos realizar las conexiones mostradas en la figura 6.



10A Arduino 1.8.13 (Windows Store 1.8.42.0)

Archivo Editar Programa Herramientas Ayuda



10A

```
#include <Servo.h>

Servo myservol;
Servo myservo2;
Servo myservo3;

int pot1 = 0; //Potenciometro en el pin analogico 0
int pot2 = 1; //Potenciometro en el pin analogico 1
int pot3 = 2; //Potenciometro en el pin analogico 2

int val1 = 0;
int val2 = 0;
int val3 = 0;

void setup(){
myservol.attach(9); //servol en el pin 9
myservo2.attach(10); //servo2 en el pin 10
myservo3.attach(11); //servo3 en el pin 11
}

void loop(){

    val1 = analogRead(pot1);
    val1 = map(val1, 3, 1023, 0, 176);

    myservol.write(val1);

    delay(25);

    val2 = analogRead(pot2);
    val2 = map(val2, 3, 1023, 0, 176);

    myservo2.write(val2);

    delay(25);

    val3 = analogRead(pot3);
    val3 = map(val3, 3, 1023, 0, 175);

    myservo3.write(val3);

    delay(25);

}
```

ACTIVIDADES

1. ¿Qué significa en el código la declaración: `#include <Servo.h>`?
2. ¿Que implica la sentencia: `va11(map, 3, 1023, 0, 176)`?
3. ¿Qué es un potenciómetro en Arduino?
4. ¿Qué hacen los 3 pines de un potenciómetro?
5. ¿Qué es y para qué sirve el servomotor?
6. ¿Qué tipo de motor es un servomotor?
7. Programar en Tinkercad el movimiento conjunto de dos servos con potenciómetro.

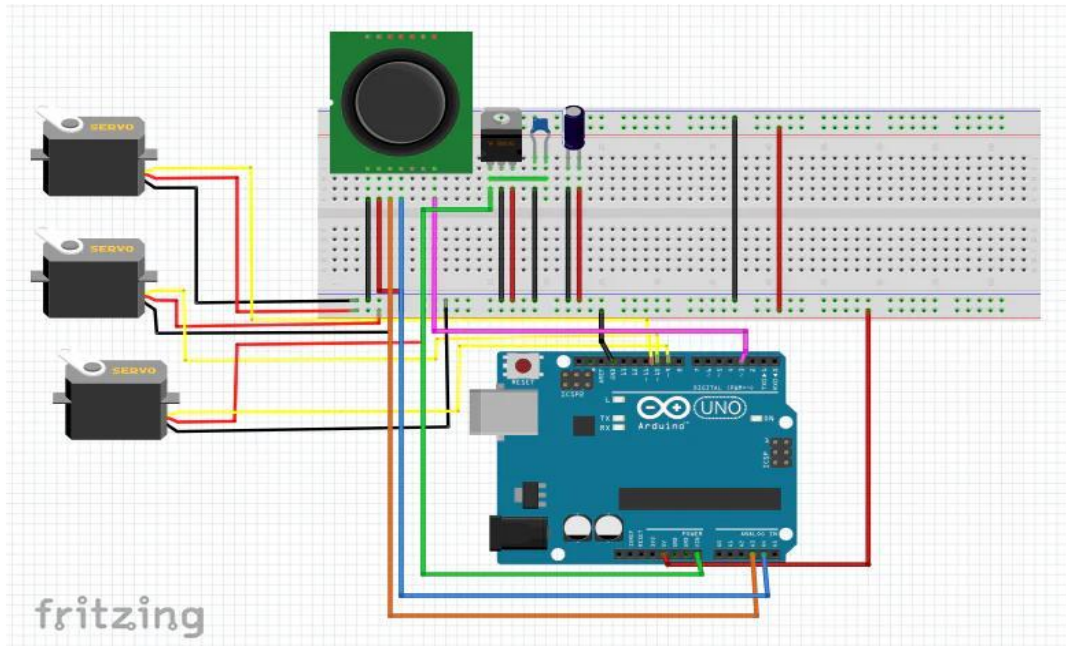
Circuito B: Brazo robótico controlado con joystick

Continuando con la serie del brazo robótico con materiales reciclables, realizaremos la programación de un brazo de cartón con un joystick para Arduino.

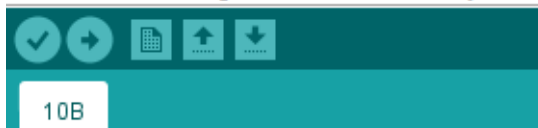
Para la serie de proyectos desde casa, estamos utilizando este kit. Podrás contar con todos los componentes necesarios para todas las prácticas de esta serie si adquieres el kit. De lo contrario, podrás realizar las prácticas con la lista de materiales a continuación:

Materiales:

- (1) Arduino Uno R3
- (1) Cable USB para programar
- (1) Cartón o material reciclado resistente.
- (3) Servomotor (utilizando el SG90)
- (1) Módulo joystick para Arduino.
- (1) Regulador de voltaje 5V.
- (1) Capacitor electrolítico 1uF.
- (1) Capacitor cerámico 100nF
- (1) Pegamento
- (1) Batería 9V con portapilas
- (15) Jumpers (cables) macho-macho



10B Arduino 1.8.13 (Windows Store 1.8.42.0)
 Archivo Editar Programa Herramientas Ayuda



```
#include "Arduino.h"
#include "Joystick.h"
#include "Servo.h"

#define JOYSTICK_PIN_VRX  A3
#define JOYSTICK_PIN_VRY  A4
#define JOYSTICK_PIN_SW  5

const int baseMinPosition  = 20;
const int baseMaxPosition = 90;

const int armMinPosition  = 20;
const int armMaxPosition = 60;

const int gripperMinPosition  = 0;
const int gripperMaxPosition = 90;

int joystickX = 55;
int joystickY = 60;

int joystickX_Cal = 0;
int joystickY_Cal = 0;

bool gripperState = 0;
```

```
Servo base;
Servo arm;
Servo gripper;
Joystick joystick(JOYSTICK_PIN_VRX, JOYSTICK_PIN_VRY, JOYSTICK_PIN_SW);

void setup() {
    Serial.begin(9600);
    base.attach(2);
    base.write(55);
    arm.attach(3);
    arm.write(60);
    gripper.attach(4);
    gripper.write(90);
    joystickX_Cal = joystick.getX();
    joystickY_Cal = joystick.getY();
}

void loop() {
    joystickX += (joystick.getX() - joystickX_Cal)/50;
    joystickY += (joystick.getY() - joystickY_Cal)/20;
    bool joystickSW = joystick.getSW();

    joystickX = constrain(joystickX, 0, 1023);
    joystickY = constrain(joystickY, 0, 1023);

    int basePos = map(joystickX, 0, 1023, baseMinPosition, baseMaxPosition);
    int armPos = map(joystickY, 0, 1023, armMaxPosition, armMinPosition);

    base.write(basePos);
    arm.write(armPos);

    if (joystickSW)
    {
        if(gripperState == 0)
            gripper.write(gripperMaxPosition);
        else
            gripper.write(gripperMinPosition);

        gripperState = !gripperState;
    }

    delay(20);
}
```

Videos de referencia

Brazos robóticos:

- Parte 1: <https://www.youtube.com/watch?v=JBI7gwf7ORU>
- Parte 2: <https://www.youtube.com/watch?v=B2lwaLmHDEI>

ACTIVIDADES

1. ¿Qué es un joystick en Arduino?
2. ¿Qué es y cómo funciona un joystick?
3. ¿Qué función cumple el regulador de voltaje?
4. ¿Qué función cumplen los capacitores?
5. ¿Cómo se alimentará de energía el sistema?
6. Enumere tres aplicaciones en práctica industrial de este sistema.
7. Diseñar un sistema de brazo robótico con por lo menos tres servos en Tinkercad.

Proyecto de estación

- Construcción de un sistema brazo robótico comandado por botones o joysticks, previo presentación de la funcionalidad en Tinkercad.

ESTACIÓN 3

DOMÓTICA

La domótica se refiere a las tecnologías que se orientan al control y la automatización inteligente de la vivienda, utilizando en forma eficiente la energía pero también optimizando los tiempos y la calidad de vida de sus habitantes. Por ejemplo: control de la iluminación, alarmas de intrusión, riego del jardín.

¿Cuáles son los beneficios de tener una casa inteligente?

- Seguridad. Un hogar inteligente es un hogar más seguro. Puedes automatizar tu alumbrado para que parezca que estás en casa, puedes comprobar a distancia si tus puertas están cerradas, si las ventanas están cerradas, qué dispositivos están activos, etc.
- Eficiencia energética. ¿Sabes cuánta energía consumen tus aparatos? ¿Tus luces necesitan estar encendidas todo el tiempo? Automatizarlos, para encenderlos sólo cuando estés cerca. Una casa inteligente es una casa sobre la que tienes control total.
- Confort. ¿Cuántas veces te preguntas si has cerrado la puerta, cerrado las ventanas o apagado la estufa? Con un sinfín de combinaciones y posibilidades de automatización, existen infinitas maneras de hacer tu vida más fácil y cómoda.

Empieza siempre por la función que quieres automatizar

Antes de que empieces a pensar cómo hacerlo, necesitas una idea de qué hacer. ¿Cuál es tu objetivo? ¿Saber cuando alguien se conecta a tu Wi-Fi, tener control sobre el consumo de energía, recibir un email cuando la alarma de humo se apaga?

Necesitas inspiración para encontrar tu idea «conductora». Piensa en las cosas repetitivas con las que luchas actualmente. Piensa en un problema y luego busca una solución inteligente.

¿Domótica cableada o inalámbrica?

La primera es decidir si va a hacer domótica cableada o inalámbrica.

La automatización cableada se recomienda para establecimientos comerciales, por ejemplo, hoteles grandes. Con la tecnología cableada dispondrá de una red algo más segura y fiable (en caso de largas distancias entre dispositivos). Mientras que la tecnología inalámbrica viene con un cierto nivel de incertidumbre cuando se trata de rendimiento a larga distancia. Con los sistemas cableados puedes estar más seguro de que la señal llegará en escenarios grandes y distantes con numerosas plantas, etc.

Cuando esté operando con una mayor cantidad de datos, el sistema cableado proporcionará una conexión más estable. Esto resulta especialmente útil cuando se tienen establecimientos comerciales y se utilizan muchos dispositivos, pantallas y sistemas de videovigilancia. La desventaja de los sistemas con cable es que no se pueden mover fácilmente los cables a una ubicación diferente cuando sea necesario. Tendrías que recablear completamente el espacio para poder mover un dispositivo.

La automatización inalámbrica es más adecuada para el uso doméstico y de pequeña oficina. La tecnología inalámbrica es mucho más fácil de instalar (no siempre se necesitan profesionales), y puedes mover libremente tus dispositivos. Cuando no se requiere una fiabilidad completa para grandes espacios, los sistemas inalámbricos son más fáciles de configurar y gestionar. Con la tecnología actual, las tecnologías inalámbricas, como Z-Wave, se acercan mucho a los sistemas cableados en términos de fiabilidad. Cuando se trata de hacer que tu «viejo» hogar sea inteligente, el sistema inalámbrico también es mucho más barato y sin obras. Si no has construido la casa con todas las características inteligentes en mente, lo más probable es que la instalación de un sistema de cableado será una costosa e inalcanzable aspiración.

Pero a veces, ni siquiera sabes si tienes un problema hasta que alguien te lo indica. Estás acostumbrado a hacer las cosas como siempre las has hecho. Eso no significa que no haya una mejor manera de hacer ciertas cosas.

Si estás dándole vueltas con ideas, aquí hay algunas que te ayudarán a pensar en lo que podrías convertir en «inteligente».

- Controla remotamente tus persianas y ventanas.
- Controla de forma remota el calentamiento del agua.

Videos de referencia

Placas ESP32

- *Conceptos y características:*

Parte 1: <https://www.youtube.com/watch?v=pdgFqPbw64g>

Parte 2: <https://www.youtube.com/watch?v=7lKTQXZgmts&t=397s>

- *ESP32 CAM - Una cámara que nos ofrece mucho por muy poco:*
<https://www.youtube.com/watch?v=AKG-rljqaDE>

¿Qué es MQTT?

El MQTT se trata de un protocolo de comunicaciones estandarizado, muy ligero y sencillo que nos permite comunicar de una forma rápida y sin gran consumo dispositivos de forma remota. Funciona sobre el TCP/IP, es decir, sobre las redes que habitualmente tenemos en casa y que reina en internet, por lo que podemos hacer el paso de dicha información, no solo a nivel de una red local sino dar el salto a través de internet.

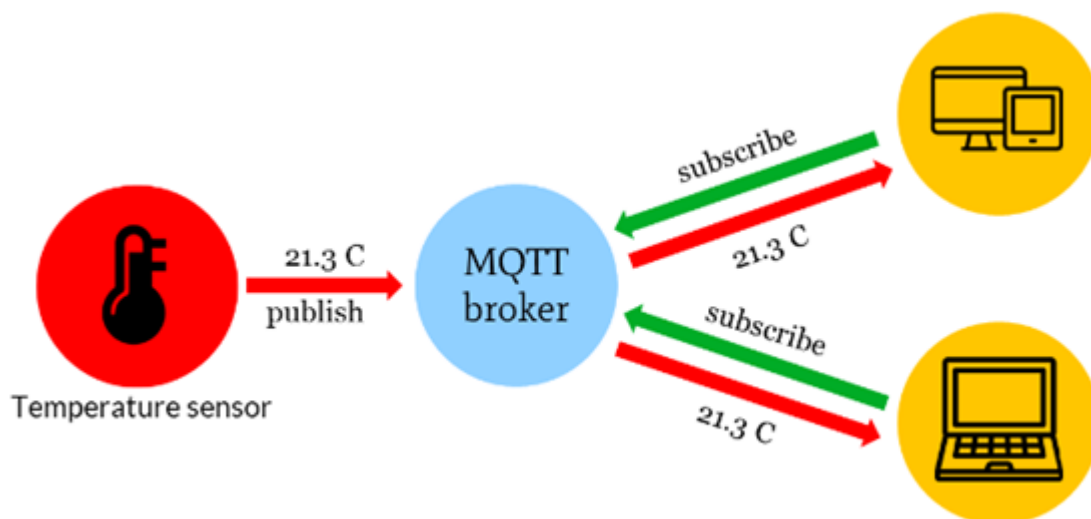
Se trata de un protocolo en el que se envían los mensajes y, por otro lado, otros dispositivos están “escuchando” por si alguno de los mensajes enviados son de su interés. A pesar de que pueda parecer algo lioso, con un ejemplo podremos verlo de una forma más sencilla. Digamos que tenemos un dispositivo que recoge la temperatura en una cámara frigorífica y, cada muestreo, lo manda (ahora veremos como lo hace) con MQTT en el que informa con un mensaje sencillo del estilo a “CAMARA/TEMPERATURA 5”, otro dispositivo, podría estar pendiente de los mensajes que vayan etiquetados con “CAMARA/TEMPERATURA” con el fin de recoger el dato que le manda, en nuestro ejemplo, 5, para si bajara o subiera de un nivel concreto realizar alguna acción (alarma, encender ventiladores, etc).

Para poder hacer el paso de mensajes, es necesario instalar un software llamado broker que es el encargado de recibir los mensajes, así como aceptar suscripciones de otros dispositivos (los que están “pendientes” de ciertos mensajes) para mostrarles los mensajes relacionados con su interés.

El mensaje enviado, puede tener lo que llaman QoS (Quality of Service, o calidad del servicio) en el que le decimos al broker si queremos que sea simplemente enviado (nivel 0), enviado con al menos 1 confirmación de entrega (nivel 1) o bien que garanticemos la entrega por completo (nivel 2).

¿En que se puede usar MQTT?

Gracias a este protocolo, los microcontroladores como el ESP8266, son capaces de funcionar de manera bidireccional (ya que mandan y se suscriben en el broker) mensajes para el control de diferentes dispositivos, desde sensores, hasta control tipo relé.



A la hora de poner en marcha tus proyectos de IoT utilizando el protocolo MQTT necesitarás un broker que administre el flujo de información. Existen muchos en la red, algunos gratuitos y otros de pago.

En este Episodio mostramos como configurar y utilizar las dos opciones gratuitas que nos ofrece la empresa Hive MQ empleando el lenguaje Micropython.

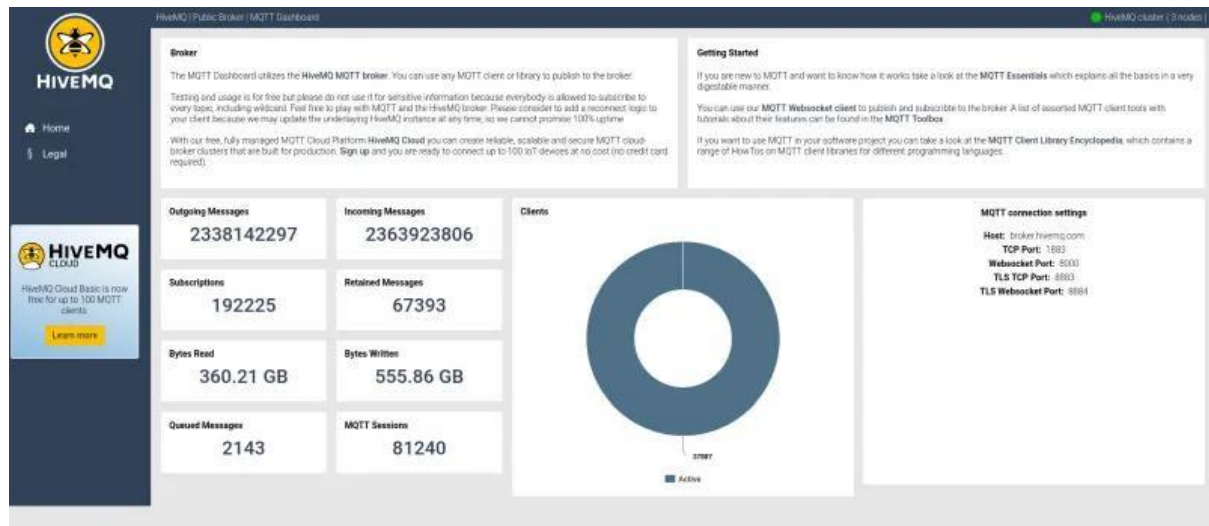
Hive MQ es una empresa que brinda distintos servicios de IoT, entre ellos los de Broker para el protocolo MQTT. Además de los planes pagos, ofrece dos opciones de servidores MQTT o broker sin costo que puedes utilizar para hacer pruebas y aprender el funcionamiento del protocolo y hasta poner en marcha pequeños proyectos de bajo tráfico.

La primera opción es un **servidor público o abierto** que se puede utilizar con una configuración mínima y permite empezar a probar el sistema en unos minutos. El otro es un **servidor privado** que cuenta con distintos planes de pago, pero que incluye también una opción totalmente gratuita.

El servidor público de Hive MQ es sumamente simple de utilizar, lo cual es una gran ventaja. Puedes publicar información y suscribirte a distintos tópicos para recibirla de una manera rápida y sencilla ya que no requiere utilizar ningún tipo de nombre de usuario o contraseña. De allí la denominación de servidor público.

Por lo tanto, este servidor sirve para hacer las primeras pruebas pero de ninguna manera se lo debería utilizar en un proyecto en que necesites mantener a salvo tu información.

Se puede acceder a un [dashboard](#) en tiempo real de este servidor que muestra el tráfico que está administrando.



Datos de acceso

Todo lo que necesitas para conectarte al servidor público son su dirección y número de puerto:

Dirección: broker.hivemq.com

Port: 1883

Si vas a utilizar una placa con ESP32, el firmware de Micropython ya tiene incluida la librería **mqtt.simple** y no necesitas realizar ninguna instalación (yo he utilizado la versión 1.19).

Este cliente está accesible en esta dirección:

<http://www.hivemq.com/demos/websocket-client/>

Connection disconnected

Host: Port: ClientID: Connect

Username: Password: Keep Alive: SSL: ☐ Clean Session: ☒

Last-Will Topic: Last-Will QoS: Last-Will Retain: ☐

Last-Will Message:

REFERENCIAS DE TRABAJO

WEBGRAFÍA

- **Protocolo MQTT**
 - <https://www.youtube.com/watch?v=0MbQ8xkS9Dk>
 - <https://www.youtube.com/watch?v=mWSkOmAwJ0s>
- **Placa ESP 32**
 - <https://www.youtube.com/watch?v=pdgFqPbw64g>
- **Broker, ESP32, MQTT y Clientes**
 - <https://www.youtube.com/watch?v=x5GML1FqcTQ>
- **ESP32 puesta en marcha y consideraciones sobre sus pines**
 - https://www.youtube.com/watch?v=xXIPs_iGv00
- **Prácticas domóticas**
 - Como encender luces con telegram y esp32 desde cualquier parte del mundo:
<https://www.youtube.com/watch?v=S3HxrIVCzGM&t=86s>
 - Botones dentro TelegramBot en ESP32 y ESP8266:
https://www.youtube.com/watch?v=5U-l_12QcQ0&list=PLWQQswW6kqpXuTWwCG79BZYfeZS-Kqp-i&index=4

ACTIVIDADES:

- ¿Qué es la tecnología Z-Wave?
- ¿Cuáles son los 3 dispositivos básicos que usa la domótica de que se encarga cada uno de ellos?
- ¿Qué significa DIY?
- Explique en qué consisten los protocolos Z-Wave y Zigbee?
- ¿En qué se basan las formación de redes mesh o de malla?
- ¿Cuál es más conveniente Z-Wave o Zigbee?
- ¿Cuál es la base de funcionamiento del protocolo MQTT?
- ¿Qué función cumple el broker?
- ¿Qué se entiende por suscripción en el protocolo MQTT?
- Crear un enlace de suscripción con un servidor MQTT gratuito.
- Un trabajo de estas características involucra conocimientos en distintos campos, lo que permite el trabajo en equipo. Justifique.
- Qué tipo de liderazgo conviene implementar. Justifique.

Propuesta de Desarrollo final

- Sistema integrado de sensores y actuadores con operación en forma remota, de aplicación funcional al proyecto del grupo de trabajo.