



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE  TECNOLOGÍAS

CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

TEMA:

IMPLEMENTACIÓN DE UNA MAQUETA DIDACTICA DEL SISTEMA DE ARRANQUE DE UNA AERONAVE, SIGUIENDO LAS NORMAS Y PROCEDIMIENTOS ESPECIFICADOS EN LOS RESPECTIVOS MANUALES, PARA LA CAPACITACIÓN TEÓRICA/PRÁCTICA DE LOS ALUMNOS DE LA ESCUELA TÉCNICA DE AVIACIÓN DEL EJÉRCITO.

AUTOR:

HÉCTOR QUINAHUANO

Objetivo general:

- Implementar una maqueta didáctica del sistema de arranque de una aeronave, siguiendo las normas y procedimientos especificados en los respectivos manuales, para la capacitación práctica de los alumnos de la especialidad de electrónica de la Escuela Técnica de Aviación del Ejército.

Objetivos específicos:

- Recopilar información técnica y realizar un estudio de las condiciones actuales del funcionamiento de los componentes de un sistema de arranque.
- Determinar los materiales que se van a utilizar en la implementación de este proyecto.
- Desarrollar un programa de control y automatización del sistema de arranque de una aeronave, para garantizar el correcto funcionamiento del sistema.

Alcance:

- Este proyecto está enfocado a la Implementación de una maqueta didáctica, para mejorar el proceso de enseñanza-aprendizaje sobre el arranque de una aeronave, utilizando microcontroladores y elementos electrónicos, el cual facilitara la capacitación técnica de los alumnos de la Escuela Técnica de Aviación del Ejército.

Sistema de arranque de una aeronave.

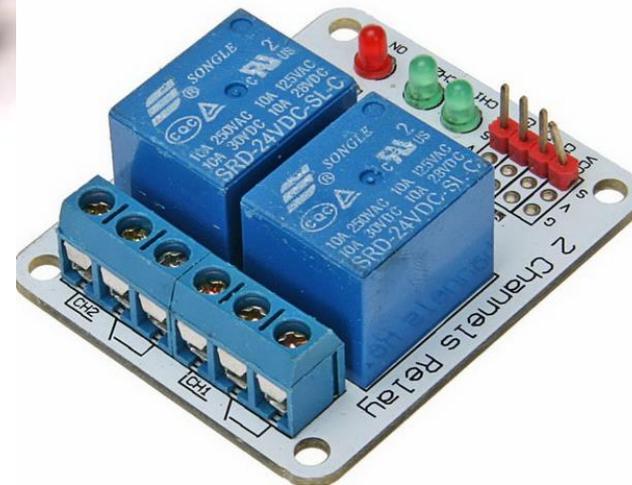
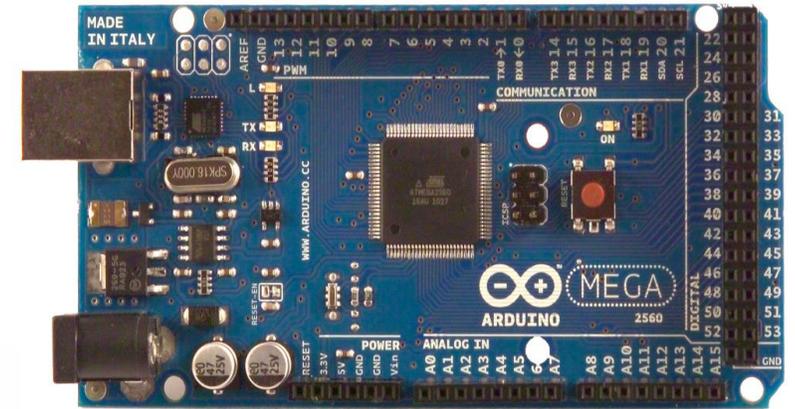
- La vanguardia de la aeronáutica han permitido que la aeronaves vayan mejorando su sistema de arranque, por lo cual muchas aeronaves utilizan el sistema de arranque eléctrico/electrónico, en este sistema van acoplados un conjuntos de equipos y elementos conformados por; fuente eléctrica, cableado eléctrico, interruptores simples, dobles, triples, solenoides , relays, contactores, bobinas y un motor de arranque.

- **Elementos electrónicos utilizados.**

MOTOR 28BY-J-48 Y CONTROLADOR ULN 2003



ARDUINO MEGA 2560



MODULO DE RELAYS

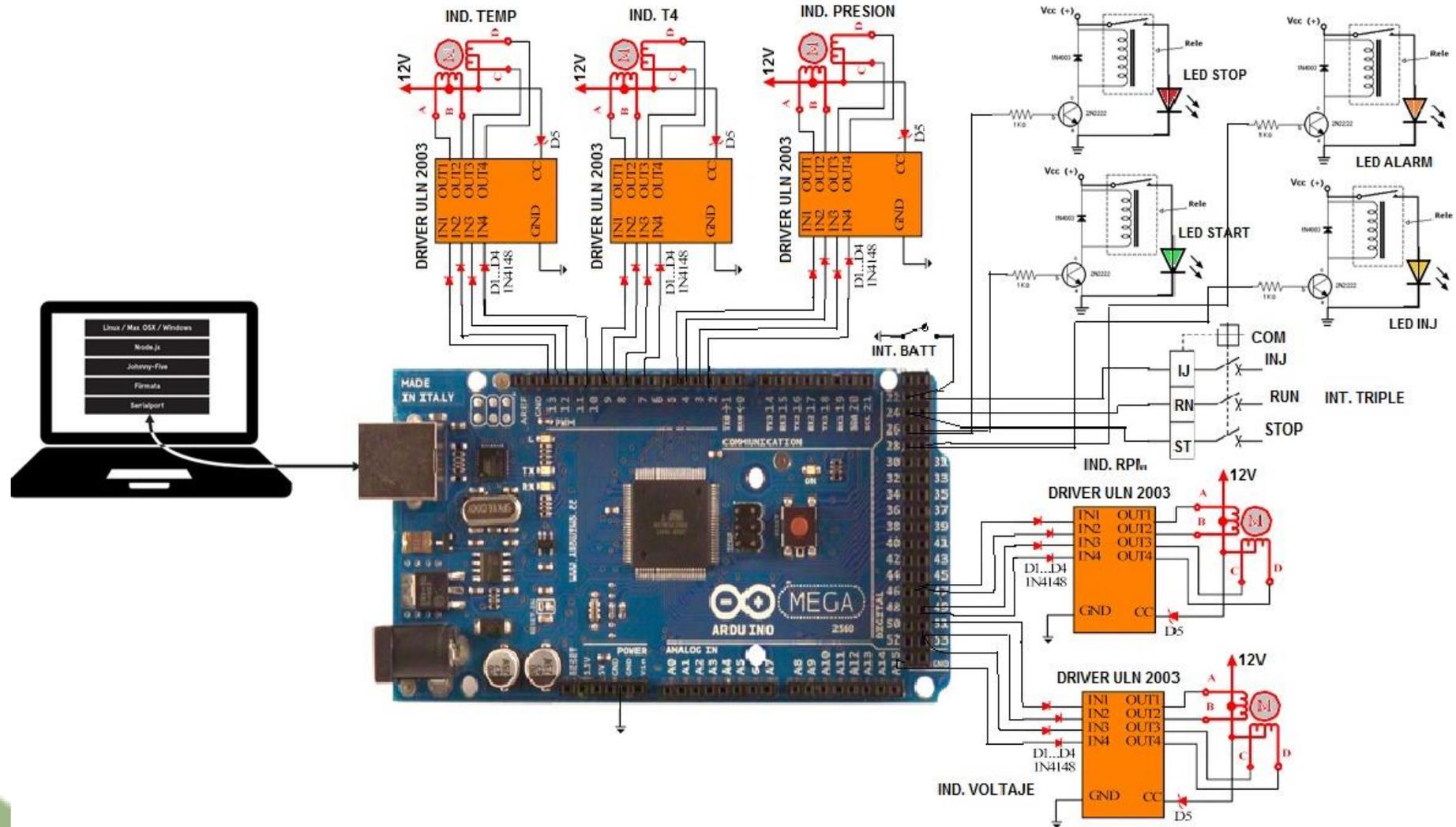
Características técnicas del motor 28BY J-48

| Tipo de motor | Motor paso a paso unipolar 28BY J-48 |
|---------------------------------|--|
| Tipo de conexión | 5 Conexión del cable (al controlador del motor) ULN 2003 |
| Voltaje | 5-12 voltios DC |
| Modo de Paso | Modo de medio paso recomendado (8 paso de la señal de control) |
| Ángulo Paso | <p>Modo medio paso: La secuencia de señal de control 8 paso.</p> <p>Modo paso completo: La secuencia de señal de control de 4 pasos .</p> |
| Relación de transmisión | <p>La relación de cada uno de los engranajes de los motores indica la relación de transmisión, que es exactamente de 63,684: 1. Por lo cual su movimientos giratorio es de 64 pasos y esto multiplicado por rotación del motor (63.684) el resultado = 4076 pasos por vuelta completa (aproximadamente).</p> |
| Cableado al controlador ULN2003 | A azul, B rosa, C Amarillo, D naranja, E rojo, común |
| Peso | 30g |

Características técnicas de la tarjeta Arduino Mega 2560.

| Especificaciones | Características |
|---------------------------|-----------------|
| Microcontrolador | ATmega2560 |
| Voltaje de operación | 5V |
| Voltaje de entrada | 7-12V |
| Voltaje de entrada limite | 6-20V |
| Pins Digitales I/O | 54 |
| Pins Analógicos Input | 16 |
| Corriente DC I/O | 20 mA |
| Memoria flash | 8 KB -256 KB |
| SRAM Y EEPROM | 8 KB y 4 KB |
| Oscilador interno | 16 MHz |

Diagrama eléctrico de la maqueta



Softwares utilizados.

- IDE Arduino



- LabVIEW



Elementos del Panel de Instrumentos utilizados para el Arranque.



- Interruptor de batería



- Interruptor de la bomba de cebado



- Selector "START"

Elementos del Panel de Instrumentos utilizados para el Arranque.



- Indicador de la presión



- Indicador de voltaje



- Indicador de la temperatura T4

- **Elementos del Panel de Instrumentos utilizados para el Arranque.**



- **Indicador de la temperatura del aceite**



- **Indicador Taquimétrico**

Programación en Arduino

- **Librerías:**

```
#include <AccelStepper.h>
```

```
#define HALFSTEP 8
```

- **Declaración de Variable:**

-
- | | |
|----------------------------|----------------------------|
| • int a,b,c,d,e; | • byte ALARM=25; |
| • byte sw1=0; | • byte INJ=26; |
| • byte sw2=0,sw3=0; | • byte ledALARM=33; |
| • byte bateria=22; | • byte ledSTART=35; |
| • byte STOP=23; | • byte ledINJ=37; |
| • byte START=24; | • byte ledSTOP=39; |
-

- **Función principal**

```
void setup()
```

- **Configuración de la comunicación serial RS 232**

```
Serial.begin (9600);
```

- **Configuración de entradas y salidas de los pines de la tarjeta**

Arduino.

- AccelStepper stepper1(HALFSTEP, 2, 3, 4, 5); // INDI. PRESIÓN//
- AccelStepper stepper2(HALFSTEP, 6, 7, 8, 9); // INDI. TEMP//
- AccelStepper stepper3(HALFSTEP, 10, 11, 12, 13); // INDI. T4//
- AccelStepper stepper4(HALFSTEP, 46, 47, 48, 49); // INDI. RPM//
- AccelStepper stepper5(HALFSTEP, 50, 51, 52,53); // INDI. V //

Configuración de entradas y salidas de los pines de la tarjeta Arduino.

```
pinMode(bateria,INPUT_PULLUP);  
pinMode(STOP,INPUT_PULLUP);  
pinMode(START,INPUT_PULLUP);  
pinMode(ALARM,INPUT_PULLUP);  
pinMode(INJ,INPUT_PULLUP);
```

```
pinMode(ledSTOP,OUTPUT);  
pinMode(ledSTART,OUTPUT);  
pinMode(ledALARM,OUTPUT);  
pinMode(ledINJ,OUTPUT);  
pinMode(ledBOMBA,LOW);
```



Lee el valor de un pin digital especificado, ya sea alto o en bajo.

- **Función cíclica.**

void loop()

- **Lectura de las variables del int de batería.**

ARDUINO

```
f(digitalRead(bateria)==LOW && sw1==0)
```

```
{
```

```
if(digitalRead(STOP)==LOW &&
```

```
digitalRead(START)==HIGH &&
```

```
digitalRead(ALARM)==LOW &&
```

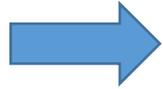
```
digitalRead(INJ)==HIGH)
```

```
{
```



MAQUETA DIDÁCTICA





```
digitalWrite(ledSTOP,LOW);  
digitalWrite(ledSTART,HIGH);  
digitalWrite(ledALARM,LOW);  
digitalWrite(ledINJ,HIGH);  
digitalWrite(ledBOMBA,LOW);  
Serial.println("b");  
}  
stepper5.moveTo(1000);  
a=stepper5.distanceToGo();  
while(a!=0)  
{  
sw1=1;  
stepper5.run();  
a=stepper5.distanceToGo();  
}  
Serial.println("c");  
}
```

MAQUETA DIDÁCTICA Y LABVIEW



Lectura de las variables del int de START.

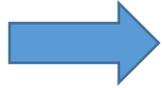
- **Posición STOP**

```
if(digitalRead(STOP)==LOW&&digitalRead(START)==HIGH&&  
digitalRead(ALARM)==LOW && digitalRead(INJ)==HIGH)
```

```
{  
  digitalWrite(ledSTOP,LOW);  
  digitalWrite(ledSTART,HIGH);  
  digitalWrite(ledALARM,LOW);  
  digitalWrite(ledINJ,HIGH);  
  if (sw3==0)  
  {  
    Serial.println("d");  
    sw3=1;  
  }  
  if(sw2==1)
```



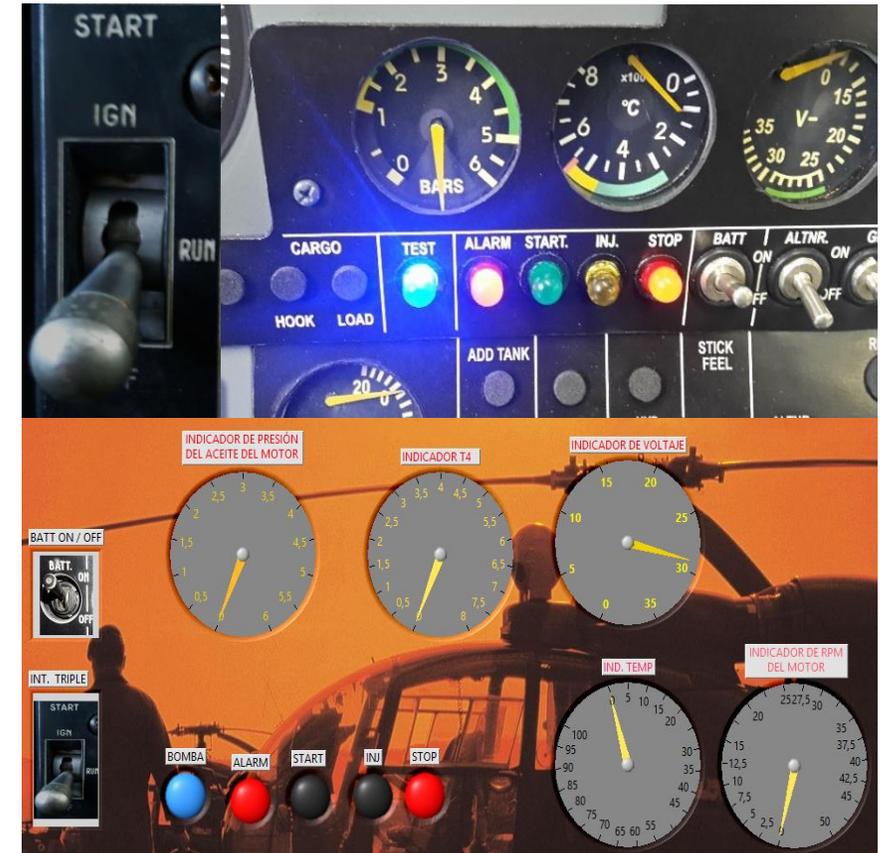
- **Lectura de las variables del int de START.**



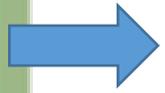
```
stepper1.moveTo(-stepper1.currentPosition());  
stepper2.moveTo(-stepper2.currentPosition());  
stepper3.moveTo(-stepper3.currentPosition());  
stepper4.moveTo(-stepper4.currentPosition());
```

```
while(b!=0)
```

```
{  
  stepper1.moveTo(-stepper1.currentPosition());  
  stepper1.run();  
  stepper2.moveTo(-stepper2.currentPosition());
```



- **Lectura de las variables del int de START.**



```
if (c!=0)
    stepper2.run();
if(d!=0)
    stepper3.run();
if(e!=0)
    stepper4.run();
}
sw2=0;
}
}
```



Lectura de las variables del int de START.

- **Posición RUN**

```
if(digitalRead(STOP)==HIGH && digitalRead(START)==LOW  
&& digitalRead(ALARM)==LOW && digitalRead(INJ)==HIGH)
```

```
{
```

```
digitalWrite(ledSTOP,HIGH);  
digitalWrite(ledSTART,LOW);  
digitalWrite(ledALARM,LOW);  
digitalWrite(ledINJ,HIGH);  
if(sw3==1)
```

```
{
```

```
Serial.println("e");
```



Lectura de las variables del int de START.



```
sw3=0;  
}  
stepper1.moveTo(3000);  
stepper2.moveTo(1200);  
stepper3.moveTo(750);  
stepper4.moveTo(500);  
stepper1.run();  
stepper2.run();  
stepper3.run();  
stepper4.run();  
sw2=1;  
}
```



Lectura de las variables del int de START.

- Posición INJ.

```
if(digitalRead(STOP)==HIGH &&  
digitalRead(START)==LOW &&  
digitalRead(ALARM)==HIGH &&  
digitalRead(INJ)==LOW)
```

```
{
```

```
digitalWrite(ledSTOP,HIGH);
```

```
digitalWrite(ledSTART,LOW);
```

```
digitalWrite(ledALARM,HIGH);
```

```
digitalWrite(ledINJ,LOW);
```

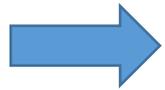
```
if(sw3==0)
```

```
{
```

```
Serial.println("f");
```



Lectura de las variables del int de START.



```
sw3=1;  
}  
stepper1.run();  
stepper2.run();  
stepper3.run();  
stepper4.run();  
}  
}  
}
```



HMI



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



The image displays a simulated HMI dashboard for a helicopter, overlaid on a background image of a helicopter cockpit. The dashboard includes several gauges and control elements:

- INDICADOR DE PRESIÓN DEL ACEITE DEL MOTOR:** A circular gauge with a scale from 0 to 6, showing a needle at approximately 1.5.
- INDICADOR T4:** A circular gauge with a scale from 0 to 8, showing a needle at approximately 1.5.
- INDICADOR DE VOLTAJE:** A circular gauge with a scale from 0 to 35, showing a needle at approximately 10.
- IND. TEMP:** A circular gauge with a scale from 0 to 100, showing a needle at approximately 10.
- INDICADOR DE RPM DEL MOTOR:** A circular gauge with a scale from 0 to 50, showing a needle at approximately 2.5.
- BATT. ON / OFF:** A control element with a rotary switch and a small inset image of the switch.
- INT. TRIPLE:** A control element with a rotary switch and a small inset image of the switch.
- PUERTO SERIAL RS232:** A control element with a dropdown menu showing '1/0'.
- STOP:** A rectangular button with the word 'STOP' in red text.
- ALARM, START, INJ, STOP:** Four circular buttons arranged horizontally at the bottom of the dashboard.

Proceso de control y monitoreo.

- La finalidad de representar un control y monitoreo del proceso de Arranque de una aeronave, es poder visualizar en tiempo real el proceso que realiza en la maqueta con la ayuda del arduino, puesto que el manejo de motores paso a paso nos permite simular el movimiento asíncrono de los instrumentos; indicadores de **presión y temperatura** del aceite del motor, temperatura del motor, indicador de **RPM** y de **voltaje**, controlados por dos interruptores; interruptor de batería e interruptor triple.
- Para esto se ocupó un arduino mega y el software LabVIEW quienes controlaran el sistema enviando señales desde el arduino hasta los elementos electromecánicos y conjuntamente hacia la PC mediante la comunicación serial RS 232, que hará las veces de HMI.

Conclusiones:

- Con la información técnica necesaria, se realizó el estudio de funcionalidad de todos los componentes del sistema de arranque, la cual permitió analizar las características técnicas de dichos elementos, además permitió conocer la prestación que brindara la maqueta didáctica en un entorno aeronáutico, de esta manera se desarrolló el proceso automatizado de encendido/arranque de una aeronave.
- Se estableció los materiales y equipos utilizados en el desarrollo de la maqueta didáctica, así como la comprobación técnica de funcionalidad de los mismos.

- En la elaboración del proyecto, se desarrolló el código de programación en la tarjeta Arduino mega para el control de elementos electromecánicos (motores paso a paso, interruptores y leds). Además se realizó una comunicación serial RS 232 entre el Arduino y la PC, mediante un cable micro USB, con lo cual se envió los datos necesarios, para realizar el monitoreo del sistema de arranque de una aeronave.
- Al diseñar el interfaz hombre maquina utilizando el software LabVIEW, se pudo monitorear el proceso automatizado del sistema de arranque de una aeronave, lo que permitirá apreciar en tiempo real los valores de los indicadores analógicos que fueron representados en la maqueta.

Recomendaciones:

- Evitar la manipulación de la maqueta didáctica si no se tiene el conocimiento previo del uso y manipulación del equipo.
- Revisar las conexiones eléctricas realizadas en la maqueta didáctica, puesto que la inobservancia de estas pueden ocasionar incidentes o inhabilitación de la maqueta didáctica.
- Considerar los avisos e indicaciones que se presentan en el interface humano máquina (HMI), con el fin de monitorear en tiempo real los instrumentos de la maqueta didáctica, de acuerdo a las instrucciones que ejecuta el Arduino mega.

¡GRACIAS POR SU ATENCIÓN!

