



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**UNIDAD DE GESTIÓN DE  TECNOLOGÍAS**

**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN**

**CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN  
Y AVIÓNICA.**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE TECNÓLOGO EN ELECTRÓNICA MENCIÓN  
INSTRUMENTACIÓN Y AVIÓNICA.**

**TEMA: “IMPLEMENTACIÓN DE UNA MAQUETA DIDÁCTICA  
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 FIDEL QUINAHUANO JITALA**

**DIRECTOR: ING. NEL AUGUSTO VACA FLORES, MSc.**

**LATACUNGA**

**2017**



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN**  
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y  
AVIÓNICA.

## CERTIFICACIÓN

Certifico que el trabajo titulado "**IMPLEMENTACIÓN DE UNA MAQUETA DIDÁCTICA 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**", realizado por el Sr. **QUINAHUANO JITALA HÉCTOR FIDEL**, ha sido revisado en su totalidad y analizado por el software anti-plagio, mismo que cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **HÉCTOR FIDEL QUINAHUANO JITALA**, para que lo sustente públicamente.

Latacunga, 17 de marzo 2017.

---

**DIRECTOR**

**ING. NEL AUGUSTO VACA FLORES, MSc.**

C.I. 1713086476



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN  
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y  
AVIÓNICA.

## AUTORÍA DE RESPONSABILIDAD

Yo, **QUINAHUANO JITALA HÉCTOR FIDEL**, con cedula de identidad 1717936585 declaro que este trabajo de titulación **"IMPLEMENTACIÓN DE UNA MAQUETA DIDÁCTICA 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"**, ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del trabajo de grado en mención

Latacunga, 17 de marzo 2017.

---

**QUINAHUANO JITALA HÉCTOR FIDEL**

**C.C.: 1717936585**



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN  
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y  
AVIÓNICA.

## AUTORIZACIÓN

Yo, **QUINAHUANO JITALA HÉCTOR FIDEL**, autorizo a la Universidad de las Fuerzas Armadas-ESPE, Publicar en la biblioteca virtual de la Institución el presente trabajo de titulación **"IMPLEMENTACIÓN DE UNA MAQUETA DIDÁCTICA 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"**, cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Latacunga, 17 de marzo 2017.

---

**QUINAHUANO JITALA HÉCTOR FIDEL**  
C.C.: 1717936585

## **DEDICATORIA**

Una nueva etapa de mi vida está por terminar y es por eso que quiero dedicar con mucho amor y cariño este trabajo de graduación a Dios, por brindarme salud y vida para poder culminar con éxito mi carrera.

A mis Padres, quienes con su apoyo incondicional y sabios consejos me han guiado por el camino del bien mediante un semblante ejemplarizado, mismo que me ha perimido ser una persona íntegra, tratando siempre de alcanzar una excelencia profesional e incentivándome a ser cada día mejor.

A mi esposa Nataly, quien con sus ocurrencias y apoyo incondicional siempre supo sacarme una sonrisa aún en los peores momentos en mi carrera profesional.

**QUINAHUANO JITALA HÉCTOR FIDEL**

**C.C.: 1717936585**

## **AGRADECIMIENTO**

A Dios y la Virgen del Quinche, por darme fuerza y valor en cada momento difícil de la vida y por mostrarme una luz donde algún día hubo oscuridad.

A la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas por abrirme sus puertas para poder formarme como Tecnólogo en Electrónica Mención Instrumentación y Aviónica.

A los docentes de la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas, en especial a mi Director de Tesis por la guía técnica y científica para la elaboración del presente trabajo.

**QUINAHUANO JITALA HÉCTOR FIDEL**

**C.C.: 1717936585**

## ÍNDICE DE CONTENIDOS

<b>CERTIFICACIÓN.....</b>	<b>ii</b>
<b>AUTORÍA DE RESPONSABILIDAD .....</b>	<b>iii</b>
<b>AUTORIZACIÓN .....</b>	<b>iv</b>
<b>DEDICATORIA.....</b>	<b>v</b>
<b>AGRADECIMIENTO.....</b>	<b>vi</b>
<b>ÍNDICE DE CONTENIDOS.....</b>	<b>vii</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>x</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>xi</b>
<b>RESUMEN.....</b>	<b>xiii</b>
<b>ABSTRACT.....</b>	<b>xiv</b>
<b>CAPITULO I.....</b>	<b>1</b>
<b>PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>1</b>
1.1. Antecedentes. ....	1
1.2. Planteamiento del problema. ....	1
1.3. Justificación.....	2
1.4. Objetivos. ....	3
1.4.1. Objetivo General. ....	3
1.4.2. Objetivos Específicos. ....	3
1.5. Alcance. ....	3
<b>CAPITULO II .....</b>	<b>4</b>
<b>MARCO TEÓRICO.....</b>	<b>4</b>
2.1 Descripción general.....	4
2.2 Sistema de arranque de una aeronave.....	5
2.3 Sistema eléctrico. ....	6
2.3.1 Generación de corriente continua.....	6
2.3.2 Regulador de voltaje. ....	8
2.3.3 Bobina de alta tensión. ....	8
2.3.4 Antorcha de encendido. ....	9
2.3.5 Contactador disyuntor. ....	9

2.4	Controles e instrumentos.....	10
2.4.1	Voltímetro.....	10
2.4.2	Interruptores.....	10
2.4.3	Luces de precaución. ....	10
2.5	Sistema de encendido.....	11
2.6	Interfaces de comunicación HMI. ....	12
2.6.1	LabVIEW.....	12
2.7	Microcontroladores.....	15
2.7.1	Características internas de un microcontrolador.....	15
2.7.2	Innovadores de tarjetas electrónicas. ....	16
2.7.3	Microcontroladores vs Microprocesadores ....	17
2.7.4	Plataformas de Desarrollo.....	18
2.7.5	Plataforma desarrollo de Arduino ....	18
2.7.6	Arduino Mega 2560.....	19
2.8	Motor de pasos 28BYJ-48.....	25
2.8.1	Características técnicas del 28BYJ-48.....	26
2.8.2	El controlador del motor de pasos ULN2003.....	27
2.8.3	Códigos de programación para motor 28BYJ-48.....	29
2.9	Comunicación serial RS232 I2C.....	29

### **CAPÍTULO III**

<b>DESARROLLO DEL TEMA.</b> ....	<b>31</b>
3.1 Preliminares.....	31
3.2 Estudio de factibilidad. ....	32
3.2.1 Factor técnico.....	32
3.2.2 Factor económico.....	32
3.3 Selección de información técnica. ....	32
3.4 Implementación de la maqueta didáctica.....	33
3.4.1 Comprobación de materiales a utilizarse.....	33
3.4.2 Elaboración de la maqueta didáctica.....	33
3.5 Prueba de funcionalidad de Arduino Mega.....	35
3.6 Programación de la tarjeta Arduino ....	37
3.6.1 Declaración de librerías.....	37
3.6.2 Declaración de variables. ....	38



3.6.3	Inicialización del programa.....	38
3.6.4	Configuración de entradas y salidas de los pines de la tarjeta Arduino.....	38
3.6.5	Estructura del control.....	39
3.6.6	Envío de datos por el puerto serial.....	39
3.6.7	Control del interruptor de batería.....	40
3.6.8	Control del interruptor triple posición OFF.....	41
3.6.9	Control del interruptor triple posición RUN.....	43
3.6.10	Control del interruptor triple posición INJ.....	44
3.6.11	Configuración de la comunicación serial.....	44
3.7	Programación en el software LabVIEW.....	45
3.7.1	Diseño del HMI en el panel frontal LabVIEW.....	45
3.7.2	Programación en el diagrama de bloques de LabVIEW.....	48
3.8	Pruebas de funcionamiento de la maqueta didáctica.....	51
<b>CAPITULO IV.....</b>		<b>55</b>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>55</b>
4.1.	Conclusiones.....	55
4.2.	Recomendaciones.....	56
<b>GLOSARIO DE TÉRMINOS.....</b>		<b>57</b>
<b>ABREVIATURAS.....</b>		<b>60</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>		<b>61</b>
<b>ANEXOS.....</b>		<b>63</b>

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Características Técnicas Arduino Mega 2560 .....	20
<b>Tabla 2.</b> Operadores aritméticos usados en arduino .....	23
<b>Tabla 3.</b> Operadores booleanos usados en arduino .....	24
<b>Tabla 4.</b> Operadores compuestos usados en arduino. ....	24
<b>Tabla 5.</b> Operadores comparación usados en arduino. ....	24
<b>Tabla 6.</b> Tipos de Datos de la plataforma de Arduino .....	25
<b>Tabla 7.</b> Especificaciones técnicas del motor unipolar 28BYJ-48 .....	27

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> Helicóptero Gazelle SA 342 L .....	4
<b>Figura 2:</b> Unidad de energía en tierra GPU.....	7
<b>Figura 3:</b> Batería de NiCd. ....	7
<b>Figura 4:</b> Bobina de alta tensión .....	8
<b>Figura 5:</b> Panel frontal del entorno grafico de LabVIEW. ....	13
<b>Figura 6:</b> Diagrama de bloques del entorno grafico de LabVIEW.....	14
<b>Figura 7:</b> Representación gráfica interna de un microcontrolador. ....	16
<b>Figura 8:</b> Microcontrolador 16f877A de fabricación norteamericana. ....	17
<b>Figura 9:</b> Tarjeta de AUTOMASIS (prototipo-guía del tipo ee-02). ....	18
<b>Figura 10:</b> Tarjeta Arduino Mega del tipo 2560. ....	19
<b>Figura 11:</b> Principales componentes Arduino Mega 2560. ....	20
<b>Figura 12:</b> Barra de menú de un sketch. ....	21
<b>Figura 13:</b> Componentes de un Sketch de arduino. ....	22
<b>Figura 14:</b> Motor de pasos 28BYJ-48 con driver ULN 2003. ....	26
<b>Figura 15:</b> Tarjeta de control del motor 28by-48. ....	28
<b>Figura 16:</b> Conexión de la placa Arduino con el driver ULN2003 y el motor 28BYJ-48.....	28
<b>Figura 17:</b> Diagrama interno de la comunicación I2C.....	30
<b>Figura 18:</b> Remoción de pintura de la carcasa del panel frontal de instrumentos. ....	34
<b>Figura 19:</b> Elaboración del panel de instrumentos y colocación de señalética y caratulas de instrumentos. ....	34
<b>Figura 20:</b> Instalación y verificación de elementos electromecánicos (interruptores simples, dobles y triple). ....	35
<b>Figura 21:</b> Comunicación entre el arduino y la PC. ....	36
<b>Figura 22:</b> Ejemplo de encendido y apagado de un led.....	36
<b>Figura 23:</b> Diagrama de conexiones de elementos electromecánicos.....	37
<b>Figura 24:</b> Indicadores de; Voltaje, Presión, T4, Temperatura y RPM.....	45
<b>Figura 25:</b> Controles de desplazamiento ON/OFF. ....	46
<b>Figura 26:</b> Indicadores de ALARM, START, STOP e INJ.....	46
<b>Figura 27:</b> Paro del proceso y selector de puerto COM. ....	47
<b>Figura 28:</b> Monitoreo el encendido y arranque de una aeronave. ....	47

<b>Figura 29:</b> Configuración del puerto serial y VISA.....	48
<b>Figura 30:</b> Configuración del puerto serial .....	49
<b>Figura 31:</b> Conversión de decimal String a carácter numérico. ....	49
<b>Figura 32:</b> Variables locales en la case structure .....	50
<b>Figura 33:</b> Programación en el diagrama de bloques de LabVIEW. ....	51
<b>Figura 34:</b> Pruebas de funcionalidad del interruptor batt, en la maqueta.....	52
<b>Figura 35:</b> Pruebas de funcionalidad del interruptor batt, en el panel frontal .....	52
<b>Figura 36:</b> Secuencia de encendido /arranque, interruptor triple en RUN en la maqueta. ....	53
<b>Figura 37:</b> Secuencia de encendido /arranque, interruptor triple en RUN en LabVIEW (panel frontal).....	53
<b>Figura 38:</b> Secuencia de encendido /arranque, interruptor triple en INJ en la maqueta. ....	54
<b>Figura 39:</b> Secuencia de encendido /arranque, interruptor triple en INJ en labwiev (panel frontal).....	54

## RESUMEN

El presente trabajo práctico tiene como finalidad de representar un modelo de proceso de Arranque de una aeronave, en este caso con el manejo de motores paso a paso para simular el movimiento asíncrono de los instrumentos; indicadores de **presión y temperatura** del aceite del motor, temperatura del motor, indicador de **RPM** y el indicador de **voltaje**, del cual serán controlados por dos interruptores; interruptor de batería e interruptor triple.

Para esto se ocupa un arduino mega y el software LabVIEW quienes controlarán 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, en base al software de **LabVIEW** y la programación desarrollada en el diagrama de bloques así como la interfaz desarrollada en el panel frontal permitirán monitorear el proceso en tiempo real. El arduino mega quien actúa como controlador es programado utilizando el IDE de Arduino, para que adquiera y envíe señales provenientes los interruptores de batería y el interruptor colocado en la maqueta. De esta manera mediante la comunicación entre ambos dispositivos Arduino y PC permitirán realizar un monitoreo de los cinco instrumentos antes mencionados, mediante el envío y recepción de datos por parte del controlador (Arduino) mismo que procesa esta información para interactuar entre los siguientes elementos; Arduino, elementos electromecánicos y el software LabVIEW.

### PALABRAS CLAVE

- **PRESIÓN**
- **RPM**
- **TEMPERATURA**
- **VOLTAJE**
- **LabVIEW**

## ABSTRACT

The present practical work aims to represent a model of process of start of an aircraft, in this case with the handling of motors step by step to simulate the asynchronous movement of the instruments; engine oil **pressure** and **temperature** indicators, engine temperature, **RPM** indicator and **voltage** indicator, which will be controlled by two switches; battery switch and triple switch.

This is done by a mega arduino and **LabVIEW** software that will control sending signals from the arduino to the electromechanical elements to the PC through serial communication RS232, which will act as HMI, based on the software of LabVIEW and the developed programation that the block diagram as well as the interface developed in the front panel will allow to monitor the process in real time. The arduino mega that acts as controller is programmed using the Arduino IDE, so that it acquires and sends signals from the battery switches and the triple switch placed in the model. This way, through the communication between the two Arduino and PC devices, it will be possible to monitor the five before mentioned instruments by sending and receiving data by the controller (Arduino) that which processes this information to interact among the following elements; Arduino, electromechanical elements and software LabVIEW.

### KEYWORDS:

- **PRESSURE**
- **RPM**
- **TEMPERATURE**
- **VOLTAGE**
- **LabVIEW**

**Checked by: Lic. Wilson Villavicencio. F, MSc**  
**DOCENTE UGT**

# CAPITULO I

## PLANTEAMIENTO DEL PROBLEMA

### 1.1. Antecedentes.

Los adelantos de la ciencia y la tecnología exige a los institutos tecnológicos a mejorar la calidad de la educación superior y lograr así contribuciones relevantes en el desarrollo de las actividades relacionadas con la aviación militar, por ende he realizado este proyecto técnico que aparte de servir como una ayuda de instrucción para quienes se forman en la carrera aeronáutica, constituirá en un aporte para la 15 B.A.E “PAQUISHA”.

La Brigada de Aviación del Ejército N° 15 “PAQUISHA”, localizada en la Provincia de Pichincha, en la ciudad de Sangolquí, tiene su Escuela Técnica de Aviación del Ejército (ETAE) que nació progresivamente con la creación del Servicio Aéreo del Ejército (SAE) en el año 1954, posteriormente constituida en Brigada de Aviación del Ejército en el año 1978, durante este tiempo ha formado a varias promociones de personal técnico aeronáutico, es así que con el afán de mejorar día a día.

Durante estos años en que la ETAE cumple funciones de capacitar al personal técnico de aviación, se han presentado dificultades para realizar dichas tareas por la falta de material didáctico; por lo cual se realiza una investigación sistemática del problema presentado, determinándose como parte fundamental la implementación de una maqueta didáctica del sistema de encendido/arranque de una aeronave, permitiendo que el personal de estudiantes obtenga conocimientos significativos basados en la práctica.

### 1.2. Planteamiento del problema.

La ETAE, en lo correspondiente a material didáctico necesarios para efectuar demostraciones prácticas del funcionamiento de las aeronaves, ha conllevado a tener dificultades de enseñanza, ya que el instituto no cuenta con una maqueta que indique el principio del sistema de arranque de una

aeronave. La falta de este material didáctico ha originado dificultades e inconvenientes en el proceso de enseñanza, ya que dichos conocimientos son dictados de manera teórica y no reforzados con la práctica.

Consecuentemente, como una alternativa de solución, se ha planteado la implementación de una maqueta didáctica, del sistema de Encendido/Arranque de una aeronave, aspecto de vital importancia pues permitirá mejorar la enseñanza, basado en la práctica y brindar una educación de calidad acorde a las exigencias propias de la carrera aeronáutica, de la mano de la tecnología actual.

### **1.3. Justificación.**

El trabajo a realizar tiene como propósito que la Escuela Técnica de Aviación del Ejército cuente con una maqueta didáctica en el cual pueda indicar el Arranque de una aeronave, de tal manera que las clases sean teórico-práctico, que servirá como apoyo en el proceso formativo, necesario para que en un futuro el instituto pueda ser avalado por la DGAC.

Los principales beneficiarios con el desarrollo de este proyecto de graduación serán los alumnos de la especialidad de electrónica de la ETAE, ya que los conocimientos teóricos adquiridos en clase, se fortalecerán mediante su capacitación práctica, adquiriendo habilidades y destrezas de un técnico aeronáutico, lo que permitirá mejorar el aprendizaje de los estudiantes. Los instructores también se verán favorecidos ya que podrán exponer todos sus conocimientos, habilidades y experiencias adquiridos de una manera más detallada, permitiendo generar eficiencia y calidad en el personal técnico.

La implementación de este material didáctico necesario para la Escuela Técnica es factible ya que cuenta con el apoyo y aprobación de la dirección de la ETAE, además la información necesaria para realizar el proyecto se encuentran en los manuales respectivos de cada una de las aeronaves y existe la disponibilidad de docentes y técnicos para la asesoría técnica, con respecto al proyecto.



## **1.4. Objetivos.**

### **1.4.1. 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.

### **1.4.2. Objetivos Específicos.**

- Recopilar información técnica y realizar un estudio de las condiciones actuales del funcionamiento de los componentes del 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.

## **1.5. Alcance.**

Este proyecto está enfocado a la creación 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.

## CAPITULO II

### MARCO TEÓRICO

#### 2.1 Descripción general

En la década de los 80 la Aviación del Ejército es robustecida al incorporarse nuevas aeronaves a su flota. Durante este periodo la Aviación del Ejército recibe los SA 330 Puma, AS 332 y SA 315 Lama. Esto sería aún más fortificado con la llegada de los legendarios caza tanques de origen francés SA 342 Gazelle, helicópteros netamente de ataque y reconocimiento equipados con misiles Hot anti- tanque.



**Figura 1:** Helicóptero Gazelle SA 342 L

**Fuente:** (Eurocopter, 1967)

El primer vuelo de estos tipos de aeronaves fue en el año de 1967, su velocidad de crucero oscilan entre los 120 Kt/h o 255 Km/h. Favorecido por un estudio aerodinámico muy profundo. Sin embargo este tipo de aeronaves fueron los pioneros en el mundo en recibir un rotor de tipo semirrígido con palas en material compuesto y seguridad integrada, un rotor trasero (anti-torque). El Ecuador es el único país de Latinoamérica que cuenta con ellos.

Las aeronaves de ala rotativa como el helicóptero Gazelle SA-342 L, SA 330 Puma, AS 332 Súper Puma y SA 315 Lama son helicópteros ligeros, polivalentes, compuestos por paneles tipo "SANDWICH-METÁLICO" de construcción Franco Británico, propulsado por un Turbo Motor TURBOMECA ASTAZOU XIV H, algunos de tipo Turbina Ligada y otros de turbina libre en los cuales impulsa un rotor principal compuesto por 3 y 5 palas y un rotor trasero de 3 y 13 Alabes.

## **2.2 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, este sistema va enlazado en conjuntos de equipos y elementos conformados por; fuente eléctrica, cableado eléctrico, interruptores simples dobles triples, solenoides y un motor de arranque. En si muchas aeronaves se activa de forma y modo automático y se desactiva cuando el motor llegan a su punto máximo de operación, pero algunas aeronaves antiguas se activan mecánicamente controlados por una palanca accionada por el piloto.

El suministro de la energía eléctrica para el arranque suele ser entregado por una batería de a bordo o muchas veces puede ser suministrado por una fuente de energía externa denominada GPU, también a través de una unidad de potencia auxiliar denominada APU. Cuando el interruptor de la batería se pone en la posición ON, cierra el circuito de arranque y suministra la energía necesaria a la barra de alimentación principal a través del relay de la batería.

Al momento de realizar la energización del motor de arranque y el interruptor de arranque, el motor de arranque no funcionará hasta que el relay de encendido es energizado por el interruptor triple de arranque, secuencialmente moviéndolo hacia la posición RUN. Cuando el interruptor triple de arranque se libera de la posición RUN este se desplaza de forma manual hacia la posición INJ. Posteriormente el relay de arranque se corta y esta deshabilita la alimentación del motor de arranque. (VUELO, 2007)

## 2.3 Sistema eléctrico.

La electrónica dentro de una aeronave es uno de los sistemas más importantes ya que la mayor parte de sus componentes funcionan en base a la electricidad y electrónica. Cualquier tipo de aeronave cuenta con dos tipos de generación eléctrica:

- Circuito de Generación de Corriente Continua
- Circuito de Generación de Corriente Alterna.

### 2.3.1 Generación de corriente continua.

La generación de corriente continua está suministrado por tres elementos principales los cuales están en capacidad de generar Corriente Continua, estos elementos son:

- Batería de Níquel Cadmio.
  - Generador Arrancador.
  - Sistema de receptáculo de planta externa.
- 
- **Generador arrancador.-** El Generador Arrancador inicialmente es un motor de arranque que absorbe la potencia de la batería o de un Grupo de Corriente Externa, para poner en funcionamiento el GTM. La tensión del arrancador es de 20 VCC y la intensidad es de 40 A. con una velocidad de 1800 r.p.m. El Generador está en capacidad de producir Corriente Continua, este está ubicado en la Plancha de accesorios del GTM para así ser arrastrado por el movimiento del mismo.
  - **Sistema de receptáculo de planta externa.-** La instalación comprende de un receptáculo de planta externa que está asociado a un contactor y a un relé auxiliar. El receptáculo de planta externa está asegurado a un plato colocado bajo la puerta del copiloto (lado izquierdo) está conectado al helicóptero a un voltaje de 28V de CC por una fuente colectiva externa o

por una unidad de poder en tierra. El receptáculo consiste de dos pines de potencia positivo y negativo, un pin auxiliar (él más corto).



**Figura 2:** Unidad de energía en tierra GPU

**Fuente:** (Fadeasa, 2015).

- **Batería de Níquel Cadmio.-** La batería está conformada por 20 pilas de 1.2 VCC y 20 A las cuales están conectadas en Serie y nos permite obtener un total de carga de 24 a 26 VCC. Con una capacidad nominal de 400 A. Las pilas son construidas en material de Níquel Cadmio, la posición de la batería está dada en la parte delantera del helicóptero para así beneficiarse de una ventilación dinámica, la batería es acoplada a la aeronave por el contactor de la batería controlado por el interruptor BAT.



**Figura 3:** Batería de NiCd.

**Fuente:**(Eurocopter, 1967)

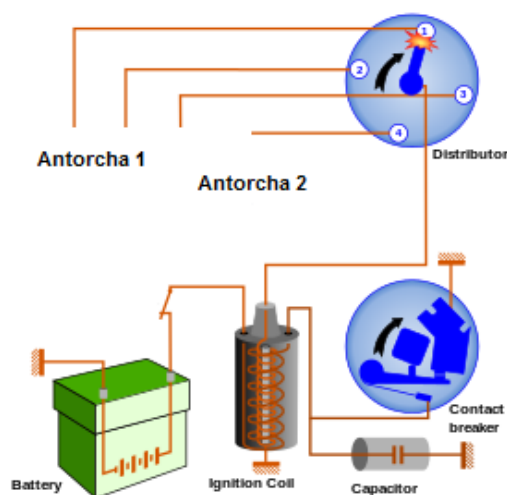
### 2.3.2 Regulador de voltaje.

El regulador de voltaje electrónico está asociado con el generador en clavado en la cabaña del hombre, posee un tornillo sobre salido que sirve para ajustar y regular el voltaje, básicamente es un elemento de protección que regula o ajusta el voltaje del generador, el instrumento de medida puede ir conectado a los dos plugs de prueba localizados en él regulador de voltaje para realizar la verificación. El regulador encierra tres diferentes secciones:

- El regulador de voltaje propiamente dicho.
- Una unidad de protección de sobre voltaje.
- Un relé de dos posiciones controlado por la unidad de protección de sobre voltaje.

### 2.3.3 Bobina de alta tensión.

La bobina de alta tensión o conocida también como bobina de encendido, es un elemento eléctrico de inducción electromagnética o inductor, que forma parte del encendido de un motor de combustión interna alternativo, que cumple con la función de mantener el voltaje normal de a bordo de 25 V a 30V según lo necesitado en la cabina, los mismos que gracias a un amplificador de alta potencia elevan su valor de 1000 veces.



**Figura 4: Bobina de alta tensión**

**Fuente:** (manualvuelo, 2014)

Esta bobina es un transformador elevador de tensión de características similar al de encendido por batería, la diferencia consiste en que tiene en serie con el primario un vibrador accionado magnéticamente por el mismo núcleo, de la bobina. La bobina es alimentada por la batería de abordó o tierra.

- **Constitución Bobina de alta tensión.-** Las bobinas de alta tensión cuentan con dos arrollamientos; primario y secundario fuertemente aislados entre sí, con una relación de N espiras de 1 a 1000 N espiras aproximadamente, con tamaños inversamente proporcionales a longitudes de cables de cobre y un núcleo ferromagnético de gran tamaño capaz de albergar a cientos y vueltas de cables de cobre. Cuenta con conexiones para las espiras del primario y para las espiras del secundario: alimentado desde el contactor de encendido del motor.

#### **2.3.4 Antorcha de encendido.**

La función es producir una llama que permita el encendido de la cámara de combustión en el arranque. Están alimentadas con combustible bajo presión por un micro bomba especial y con corriente de AT por la bobina de encendido.

#### **2.3.5 Contactor disyuntor.**

El contactor disyuntor es un elemento electrónico que acopla al generador a la batería cuando entra a funcionar el generador el contactor disyuntor pasa a la posición de activado permitiendo alimentar a todo el helicóptero o a través de la barra principal y recargar a la batería. Debajo del asiento del copiloto se encuentra el corazón eléctrico.

- **Principio de funcionamiento del contactor disyuntor.-** Asegura el acoplamiento del generador a la barra de distribución automáticamente cuando se cumplen las siguientes condiciones:
  - Interruptor "GEN" está en posición ON

- Relé auxiliar de planta externa no está energizado
- La unidad de protección del regulador de voltaje está en reposo
- La puesta en funcionamiento ocurre cuando el voltaje del generador es mayor  $0,5V \pm 0,15V$  de CC que el voltaje de la barra de distribución.

## **2.4 Controles e instrumentos.**

Los controles e instrumentos para el sistema de control y monitoreo se encuentran agrupados en el panel de control, detallados a continuación;

### **2.4.1 Voltímetro.**

Está conectado directamente a la BUS BAT tiene un rango de voltaje de 0 a 35V, provee una indicación visual producida por el generador cuando está activada la batería, cuando el generador esta desactivado y el receptáculo de planta externa no está energizado.

### **2.4.2 Interruptores.**

Un interruptor GEN que actúa sobre el sistema de control, de relé de corriente inversa permitiendo conectar o desconectar el generador. Un interruptor de BAT permite conectar la batería o la planta externa directamente a la BUS BAT.

### **2.4.3 Luces de precaución.**

Se debe de tener las siguientes consideraciones al momento de realizar las prácticas estándares de arranque/ encendido:

- Una luz GEN localizada en el panel del helicóptero, esta luz nos indica el funcionamiento normal o anormal del contactor disyuntor.
- Una luz BATT indica la abertura del contactor de batería o batería apagada.



- Una luz START indica que el encendido de dio en el momento indicado de la operación realizada.
- Una luz ALARM indica que se debe tener precaución al momento de realizar el encendido.
- Una luz STOP indica que esta en modo detenido en caso de emergencia.
- Una luz INJ indica que se está inyectando combustible necesario para continuar con el encendido.

**NOTA:** las luces de precaución "GEN y BAT" se encienden cuando está conectado la fuente de planta externa y está alimentada de 28 V.

## **2.5 Sistema de encendido.**

Un motor de combustión interna ciclo compuesto, el sistema de encendido proporciona una chispa de alto amperaje, de manera que enciende la mezcla aire/combustible, los motores de combustión interna ciclo compuesto se compone de:

- Magnetos,
- Bujías,
- Cables de alta tensión y
- El interruptor de encendido.

El magneto interno genera una tensión de voltaje competentemente alta, para generar la chispa en la bujía de cada cilindro interno. El sistema inicia al momento que el motor de arranque se activa y el eje principal empieza a girar. El funcionamiento de los magnetos es controlado en la cabina de vuelo por el interruptor de encendido. El interruptor tiene tres posiciones:

- OFF
- RUN
- IGNICIÓN.

Incluso con el interruptor triple de arranque desactivado, se encuentra en la posición OFF, si el cable de tierra está en o lógico este no presenta ninguna reacción no contrario si se encuentra en la posición RUN, ahí presenta un cambio de encender los magnetos de encendido. Mientras que si el interruptor triple se encuentran en la posición INJ, esta acción indica que se está inyectando combustible y si se presenta cambios en los instrumento de control del motor, si esto ocurre la única manera de parar el motor es mover el interruptor triple a la posición RUN y luego a la posición OFF (SISTEMAS DE ARRANQUE, 2011).

## **2.6 Interfaces de comunicación HMI.**

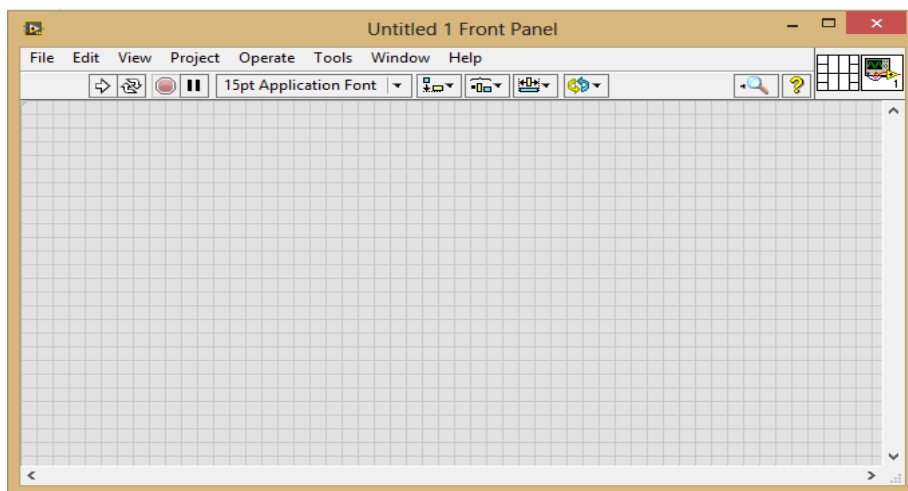
Existen varios tipos de interfaces de comunicación los cuales permiten la comunicación virtual entre el usuario, elementos electromecánicos y elementos mecánicos entre los más importantes tenemos a continuación:

### **2.6.1 LabVIEW.**

LabVIEW es un entorno de desarrollo gráfico, que tiene como objetivo ayudarnos a programar gráficamente de una manera muy sencilla. Además, tienes bibliotecas integradas y específicas, con de funciones de software e interface de hardware y características para análisis, visualización y compartir datos. Los programas realizados en LabVIEW se denominan Instrumentos Virtuales (VIs), porque su forma y funcionamiento imitan los de un instrumento real. Sin embargo son análogos a las funciones creadas con los lenguajes de programación convencionales. Los VIs tienen una parte interactiva con el usuario y otra parte de código fuente, y aceptan parámetros procedentes de otros Vis, todos los VIs tienen un panel frontal y un diagrama de bloques. (National Instruments, 2014)

- **Panel Frontal.-** Se trata de una interfaz gráfica del VI con el usuario. Esta interfaz recoge las entradas procedentes del usuario y representa las salidas proporcionadas por el programa. Un panel frontal está formado por una serie de botones, pulsadores, potenciómetros, gráficos. Cada uno de

ellos puede estar definido como un control o un indicador. Los controles sirven para introducir parámetros al VI, mientras que los indicadores se emplean para mostrar los resultados producidos, ya sean datos adquiridos o resultados de alguna operación. A continuación se presenta un ejemplo del panel frontal. (National Instruments, 2014)

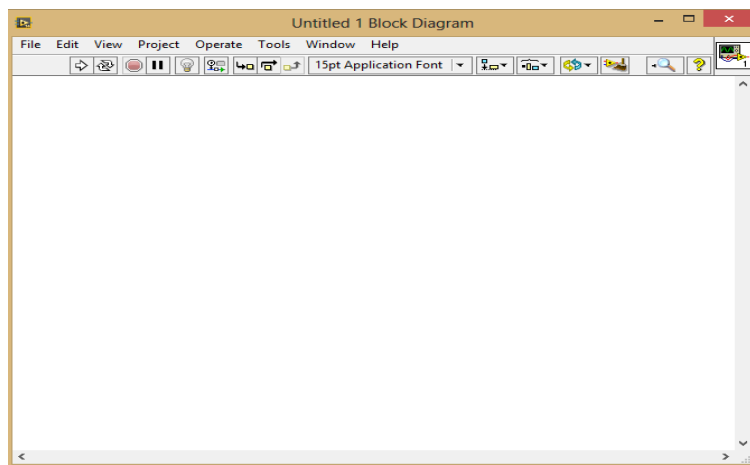


**Figura 5:** Panel frontal del entorno grafico de LabVIEW.

**Fuente:** (National Instruments, 2014).

- **Diagrama de bloques.-** El diagrama de bloques compone la compilación del VI. En el diagrama de bloques es donde se realiza la ejecución del programa del VI, para controlar o realizar cualquier proceso de las entradas y salidas que se crearon en el panel frontal. El diagrama de bloques se incluye funciones y estructuras integradas en las librerías que incorpora LabVIEW.

Las funciones y las estructuras son nodos elementales. Son similares a los operadores o librerías de funciones de los lenguajes convencionales, los controles e indicadores que se colocaron previamente en el Panel Frontal, se materializan en el diagrama de bloques mediante los terminales. A continuación se presenta un ejemplo del panel frontal:



**Figura 6:** Diagrama de bloques del entorno grafico de LabVIEW.

**Fuente:** (National Instruments, 2014).

- **Ventajas de LabVIEW.-** El empleo del software LabVIEW nos proporciona las siguientes características y ventajas:
  - Se reduce el tiempo de desarrollo de las aplicaciones, ya que es muy intuitivo y fácil de aprender.
  - Tiene una gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
  - Da la posibilidad a los usuarios de crear soluciones completas y complejas.
  - Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
  - El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.
  - Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

Con LabVIEW se puede alcanzar más objetivos propuestos en menor tiempo, ya que es el único entorno de programación gráfica; integradas de: bibliotecas constituidas por funciones de software e interfaces de hardware, contando con características para el análisis, visualización y compartimientos datos.

- **Programación en LabVIEW.-** Con el entorno gráfico de programación de LabVIEW se comienza a programar a partir del panel frontal. En primer lugar se definirán y seleccionarán de la paleta de controles todos los controles (entradas que dará el usuario) e indicadores (salidas que presentará en pantalla el VI) que se emplearán para introducir los datos por parte del usuario y presentar en pantalla los resultados.

Una vez colocados en la ventana correspondiente al panel frontal todos los objetos necesarios, debe pasarse a la ventana (diagrama de bloques). En donde se realiza la programación propiamente dicha. Al abrir esta ventana, en ella se encuentran los terminales correspondientes a los objetos situados en el panel frontal, dispuestos automáticamente por LabVIEW, se deben ir situando las funciones, estructuras, etc. que se requieran para el desarrollo del programa, las cuales se unen a los terminales mediante cables. (National Instruments, 2014).

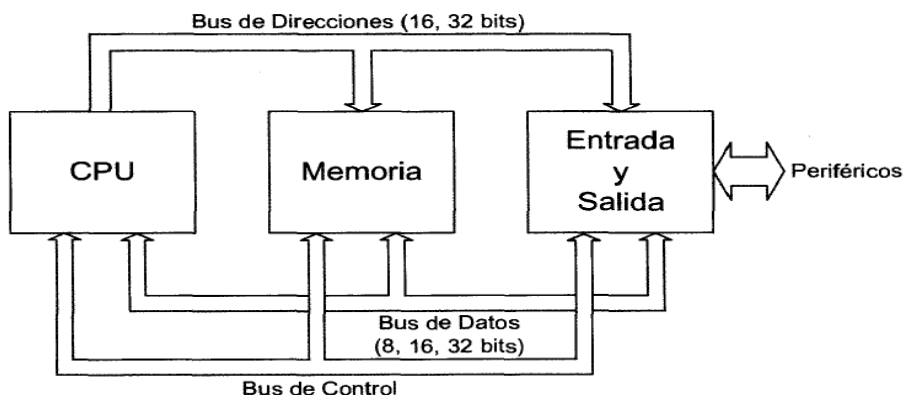
## **2.7 Microcontroladores**

Es la composición de varios circuitos integrados, donde estos están conformados por una arquitectura interna de un computador, esto suelen estar compuestos por; CPU, memorias RAM, EEPROM, SRAM, DRAM y interfaces de entradas y salida. Un microcontrolador nuevo no puede realizar ninguna tarea, ya estos integrados o tarjetas deben ser programado para que realice actividad desde un simple programa hasta el control y monitoreo de una secuencia de encendido. Estos circuitos impresos son capaz de reemplazar las tareas de los circuitos lógicos; como las compuertas lógicas, conversores de A/D, D/A, temporizadores midiendo así los tiempos de inicialización del programa, simplificando así un espacio en las placas de baquelita diseñada.

### **2.7.1 Características internas de un microcontrolador.**

Al igual que una computadora un microcontrolador se forma de bloques internos que a continuación lo describe; Unidad de Procesamiento Central (CPU), la memoria propiamente dicha e interfaces de entrada y salida. Las

unidades internas se enlazan entre sí mediante grupos de líneas de transmisión llamados buses. Los buses que transportan direcciones de memorias de entradas y salidas son conocidos como buses de dirección y si transportan datos o instrucciones son llamados buses de datos, o de control (Pérez, 2007)



**Figura 7:** Representación gráfica interna de un microcontrolador.

**Fuente:** (Pérez, 2007).

### 2.7.2 Innovadores de tarjetas electrónicas.

Los avances de la tecnología han permitido que la rama de la electrónica vaya en vanguardia con los procesos de fabricación de chip y tarjetas sumamente inteligentes y eficaces. Para lo cual detallaremos a continuación:

- Microchip:
- Atmel Corporation
- Texas Instruments
- Motorola
- Intel

La mayoría de estas empresas manufactureras están dedicadas a la fabricación de microchip para la modernización de sus tarjetas entrenadoras, a las cuales se han hecho referencias a sus industrias y a su participación en los avances de la tecnología. Además crean procesadores, microprocesadores y microcontroladores con capacidades hasta de 64kb de memoria con 75 instrucciones de direcciones, datos y bloqueo, sus mayores

logros en la industria han sido poner en el mercado microprocesadores de 8 bits, llamado 6800JH.

La mayoría de estas empresas son norteamericana, los cuales desarrollan y comercializa semiconductores con una gama de tecnología alta para los sistemas de cómputo. Igualmente, son los mayores productores de procesadores digitales de señal y de semiconductores analógicos/ digital. A continuación un ejemplo de un microcontrolador fabricado por microchip.



**Figura 8:** Microcontrolador 16f877A de fabricación norteamericana.

**Fuente:** (Sánchez., 2013)

### 2.7.3 Microcontroladores vs Microprocesadores

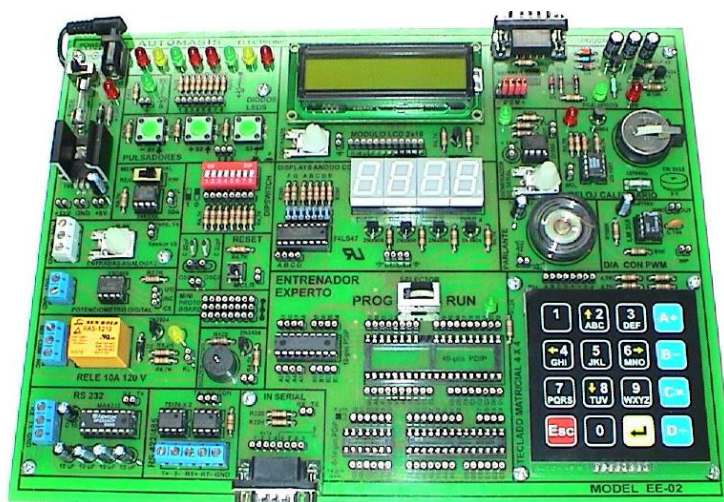
Según la historia de la tecnología los microcontroladores fueron tomando lugar luego de los microprocesadores, y estos han venido formando parte de los avances de la tecnología, evolucionándose así cada vez que sean necesarios de modificar sus estatus. Los microprocesadores al igual que los microcontroladores han desarrollado ordenadores de trabajo de mucha eficiencia, además están orientados al mercado para ilustrar a la tecnología.

Los microcontroladores y los microprocesadores han desarrollado sistemas de masivas aplicaciones y estas han cubierto diversas aplicaciones. Además se han usado en robótica, automoción, equipos de telecomunicaciones y de telefonía satelital, en instrumentos eléctricos/electrónicos, en equipos médicos e industriales de todo tipo, en electrodomésticos, en juguetes, etc.

## 2.7.4 Plataformas de Desarrollo.

Los entornos gráficos representados en plataformas de entrenamiento y desarrollo, desempeñan un rol importante dentro de la industria de la electricidad y la electrónica, permitiendo así disminuir los tiempos de respuesta en un proceso bastante complejo, ya que su diseño nos permite tener una rápida respuesta, aumentando la confiabilidad y velocidad de fabricación de cualquier prototipo.

En general la mayoría de las plataformas de entorno y desarrollo son compuestas de placas que integran; pequeños microcontroladores, circuitos y componentes electrónicos que permiten realizar diversas aplicaciones, algunas con alimentación eléctrica y otras solamente necesitan una entrada de audio. (Microbyte, 2005)



**Figura 9:** Tarjeta de Automasis (prototipo-guía del tipo ee-02).

**Fuente:** (Reyes, 2008).

## 2.7.5 Plataforma desarrollo de Arduino

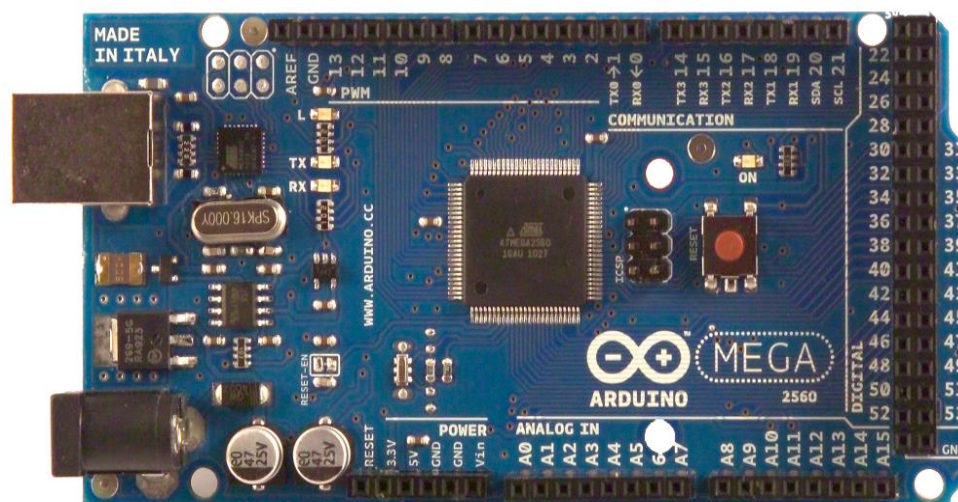
Básicamente la plataforma de desarrollo de Arduino es una herramienta sofisticada que permite controlar de manera física a través de un ordenador, conocida también como desarrollo de computación física que nos muestra códigos de programación en un lazo abierto en algunos casos y códigos de programación en lazo cerrado en otros casos más complejos. (Cádiz, 2011).



### 2.7.6 Arduino Mega 2560.

Arduino Mega del tipo 2560, es una placa electrónica con todo lo necesario para interactuar entre la tarjeta propiamente dicha y la PC, generalmente está basada en un microcontrolador del tipo Atmega 2560. Además cuenta con; 54 pines de entrada / salida digital, 16 entradas analógicas, 4 UARTs), un oscilador de cristal interno de 16 MHz, una conexión USB para la conexión a la PC, un conector de alimentación de 9 Vcc, y un botón de reset.

Contiene todo lo necesario para apoyar al microcontrolador, solamente es necesario conectar el cable USB a un ordenador, para que este tenga una comunicación serial de tipo RS232, adaptándose secuencialmente al ordenador de la PC, al igual que todos los dispositivos de la familia Arduino son compatibles con la mayoría de otros dispositivos electrónicos. (Arduino CC, 2010)



**Figura 10:** Tarjeta Arduino Mega del tipo 2560.

**Fuente:** (Arduino CC, 2010).

- **Especificaciones técnicas de la tarjeta Arduino.-** Existen varias especificaciones técnicas que nos permiten conocer la tarjeta Arduino Mega 2560, entre las cuales tenemos las siguientes:

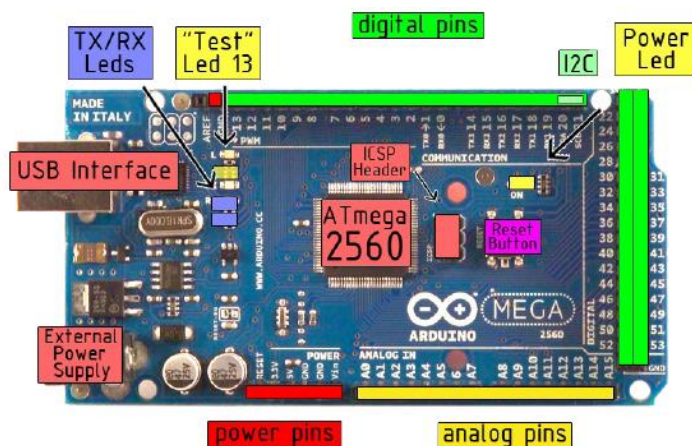
**Tabla 1**

Características Técnicas Arduino Mega 2560.

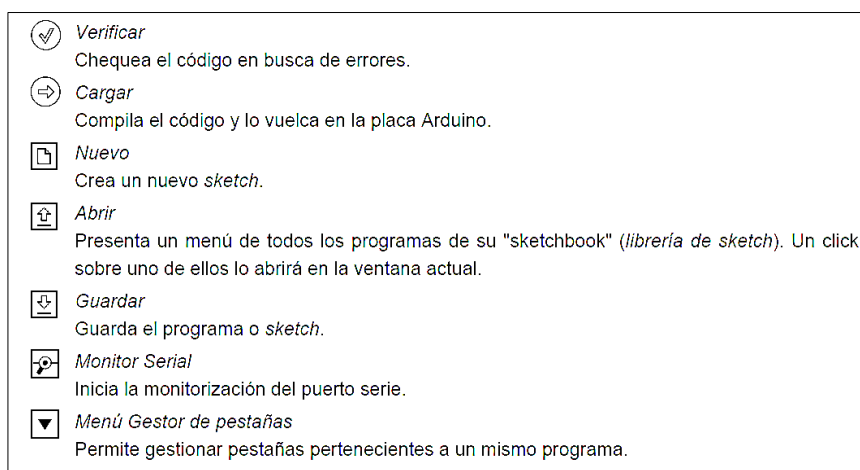
Especificaciones	Características
Microcontrolador	ATmega2560
Voltaje de operación	5V
Voltaje de entrada	7-12V
Voltaje de entrada limite	6-20V
Pins Digitales I/O	54
Pins Analógicos Input	16
Corriente DC I/O	20 mA
Memoria flash	256 KB - 8 KB
SRAM	8 KB
EEPROM	4 KB
Oscilador interno	16 MHz

**Fuente:** (Arduino CC, 2010)

- **Componentes principales de Arduino Mega.-** Dentro de la tarjeta arduino mega existen varios componentes detallados a continuación; 54 pines de entrada / salida digital, 16 entradas analógicas, 4 UARTs), un oscilador de cristal interno de 16 MHz, una conexión USB para la conexión a la PC, un conector de alimentación de 9 Vcc, y un botón de reset, a continuación la ilustración grafica de la tarjeta arduino mega 2560.

**Figura 11:** Principales componentes Arduino Mega 2560.**Fuente:** (Arduino CC, 2010)

- **Entorno de desarrollo para Arduino mega 2560.-** Las plataformas de desarrollo de Arduino nos permiten, mejorar la conexión desde la tarjeta arduino hacia el ordenador, además nos permiten cargar programas en código hexadecimal desde la PC hacia la tarjeta arduino, consecuentemente el despliegue de ventana sketch, nos permiten escribir el programa deseado al cual se obtiene algunos editores de texto como son; cortar, copiar, pegar, etc. En el área de mensajes de información se muestra leyendas mientras se cargan los programas y también muestra los errores.



**Figura 12:** Barra de menú de un sketch.

**Fuente:** (Technology, 2013)

Para la exploración de Arduino existen menús importantes que nos permiten: Archivar, Editar y una barra de Herramientas, al cual se puede ir agregando librerías según sean los casos. Además insertar tareas simultáneamente para que se ejecuten al momento de cargar a la tarjeta arduino.

- **Estructura Sketch.-** En el entorno de programación de arduino, la estructura sketch nos permite tener un esquema de programación más amplio en relación a otros programas de compilación, ya que tiene 3 partes principales, que se detallan a continuación.
  - La zona global
  - La función **void setup()**

– La función **void loop()**

En estas tres partes del entorno de programación encontramos un sinnúmero de opciones como las estructuras de control y los operadores de artificios matemáticos.

```

Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}

```

1 Arduino Uno on /dev/tty.usbmodemfa131

Zona Global      void Setup  
void Loop

**Figura 13:** Componentes de un Sketch de arduino.

**Fuente:** (MEDIALAB, 2010).

- **Zona global.-** En la zona global de arduino nos indica; los nombres de los pines y donde vamos a crear variables que queremos que existan en todo el programa, para que no exista errores de compilación. (TOVAR, 2014).
- **La función void setup().-** La función void setup () nos permite ejecutar un programa cada vez que inicie el programa o también cada vez que se utilice el pulsador reset, además realiza operaciones cuánticas u operaciones aritméticas cada vez que configuramos de los pines que vamos a utilizar. (TOVAR, 2014).

- **La función void loop().**- La función void loop es la parte central e inteligente del programa creado en arduino ya que esta función permanece en ejecución en forma de bucle infinito en un lazo cerrado. Esto quiere decir que se ejecuta de comienzo a fin, de forma repetida, siempre. (MEDIALAB, 2010).
- **Estructuras de Control de arduino.**- Para el buen manejo de operaciones aritméticas, constantes y herramientas de menús, en el entorno de desarrollo de arduino es necesario conocer las estructuras de control, que nos permitirán conseguir un control de cualquier programación, las mismas que se detallan a continuación:(TOVAR, 2014).
  - if...else (comparador sí o no).
  - For.... (contador).
  - While... (comparador).
- **Operadores.**- El uso de operadores aritméticos, booleanos, compuestos y comparadores nos permite tener un lenguaje de programación C y C++, los mismos que se clasifican en; aritméticos (Tabla 2), booleanos (Tabla 3), compuestos (Tabla 4) y de comparación (Tabla 5). (TOVAR, 2014).

**Tabla 2**

Operadores aritméticos usados en arduino.

OPERADOR	ACCIÓN REALIZADA
=	Operador de la asignación.
+	Operador de suma.
-	Operador de resta.
*	Operador de multiplicación.
/	Operador de división.

**Fuente:** (TOVAR, 2014)

**Tabla 3**

Operadores booleanos usados en arduino.

OPERADOR	ACCIÓN REALIZADA
&&	AND Lógico.
	OR Lógico.
!	NOT Lógico.
&	Operador de bits AND.
	Operador de bits OR.
~	Operador de bits NOT.

Fuente: (TOVAR, 2014)

**Tabla 4**

Operadores compuestos usados en arduino.

OPERADOR	ACCIÓN REALIZADA
X++	Incrementa a X en 1 y devuelve el valor antiguo de X.
++X	Incrementa a X en 1 y devuelve el nuevo valor de X.
X--	Decrementa en 1 a X y devuelve el valor antiguo de X.
--X	Decrementa X en 1 y devuelve el nuevo valor de X.

Fuente: (TOVAR, 2014)

Los operadores compuestos usados en arduino son muy similar los usados en lenguaje C y C++.

**Tabla 5**

Operadores comparación usados en arduino.

OPERADOR	ACCIÓN REALIZADA
==	Igual.
!=	Diferente.
<	Menor que.
>	Mayor que.
<=	Menor o igual que.
>=	Mayor o igual que.

Fuente: (TOVAR, 2014)

Los operadores de comparación son usados en arduino son muy similar los usados en lenguaje C y C++ ya que tienen realizan las mismas acciones.

- **Variables, constantes y tipos de datos.-** Para el manejo del entorno de Arduino es necesario conocer las variables, constantes y los tipos de datos existen, además conocer su tamaño y lugar de ubicación dentro de un Sketch, esto nos permitirá tener un control de los mismos, para asignar una variable o almacenar una constante se debe declarar en la zona global, ya que estas estarán en uso continuo cada vez que se ejecute el programa, además podrán realizar un cálculo matemático o aritmético dependiendo las variables y constantes declaradas. (Technology, 2013).

**Tabla 6**

Tipos de Datos de la plataforma de Arduino

TIPO	TAMAÑO	RANGO	DESCRIPCIÓN
<b>Boolean</b>	1 bit	0 a 1	Entero de 1 bit
<b>Byte</b>	8 bit	0 a 255	Entero
<b>Unsigned char</b>	8 bit	0 a 255	Entero sin signo
<b>Char</b>	8 bit	-128 a 127	Entero con signo
<b>Int</b>	8 bit	-32,768 a 32,757	Entero de 8 bit con signo
<b>Word unsigned int</b>	16 bit	0 – 65,535	Entero de 16 bit
<b>Long</b>	16 bit	-2,147'483,64 a 2,147'483,647	Entero de 16 bit con signo
<b>Unsigned long</b>	4 bit	0 a 4,294'967,295	Entero sin signo de 4 bits
<b>Float/double</b>	4 bit	-3.4028235e37 a 3.4028235e38	Decimal con signos de 4 bytes

**Fuente:** (Alazte, 2016)

## 2.8 Motor de pasos 28 BYJ-48.

El principio de funcionamiento de un motor paso a paso se define en que se mueve en incrementos angulares precisos, conocidos como pasos, un motor paso a paso es muy parecido a un servo: se mueve a una posición predefinida. Los servos, sin embargo, los servos se limitan en movimientos generalmente a un rango de 0-180 grado, mientras que un motor paso a paso



puede girar de forma continua en un grado de 0 – 360 grado, similar a un motor ordinario de corriente continua.

Además tiene las características de un motor servo que constituye en uno de los motores más confiables en el área de robótica ya que nos permite obtener una mayor eficacia de trabajo de los robos al momento de realizar una incursión en aérea de explotación minera, la ventaja de un motor paso a paso con respecto a los motores comunes de corriente continua, es lograr una mayor precisión y control sobre el movimiento. La desventaja de usar motores paso a paso es su difícil control.



**Figura 14:** Motor de pasos 28 BYJ-48 con driver ULN 2003.

**Fuente:** (Ilamas, 2016)

El 28 BYJ-48 es un pequeño motor paso a paso de 5 o 12 voltios, bastante económicos. Estos pequeños steppers (motores paso a paso) se pueden obtener junto con un pequeño controlador del motor de pasos ULN 2003 compatible con Arduino. (arduino, 2015).

### 2.8.1 Características técnicas del 28 BYJ-48

Características y especificaciones que se debe tomar en cuenta al momento de trabajar con el motor paso a paso unipolar 28BYJ-48



**Tabla 7**

Especificaciones técnicas del motor unipolar 28BYJ-48

<b>Tipo de motor</b>	<b>Motor paso a paso unipolar 28BYJ-48</b>
<b>Tipo de conexión</b>	5 Conexión del cable (al controlador del motor)
<b>voltaje</b>	5-12 voltios DC
<b>Frecuencia</b>	100 Hz
<b>Modo de Paso</b>	Modo de medio paso recomendado (8 paso de la señal de control de secuencia)
<b>Ángulo Paso</b>	<b>Modo medio paso:</b> La secuencia de señal de control 8 paso (recomendada) 5.625 grados. <b>Modo paso completo:</b> La secuencia de señal de control de 4 pasos 11.25 grados.
<b>Relación de transmisión</b>	El desmontaje de los engranajes de los motores indica la relación de transmisión exacta es de <b>63,68395: 1</b> . Por lo cual su movimientos giratorio es de 64 pasos por rotación del motor x 63.684 relación de transmisión = 4076 pasos por vuelta completa (aproximadamente).
<b>Cableado al controlador ULN2003</b>	A azul, B rosa, C Amarillo, D naranja, E rojo, común
<b>Peso</b>	30g

Fuente: (arduino, 2015).

### 2.8.2 El controlador del motor de pasos ULN2003.

El controlador del motor 28BYJ-48, es el ULN2003 el cual nos permite controlar fácilmente el motor paso a paso 28BYJ-48 desde un micro controlador, como el Arduino Uno, Mega.



**Figura 15:** Tarjeta de control del motor 28by-48.

**Fuente:** (llamas, 2016)



**Figura 16:** Conexión de la placa Arduino con el driver ULN2003 y el motor 28BYJ-48.

**Fuente:** (llamas, 2016)

**NOTA:** Tenga en cuenta que no se recomienda alimentar el motor con la conexión de 5 V de la placa Arduino, ya que el motor puede drenar más corriente que el micro controlador de la que puede manejar y podría dañarlo. Conecte el conductor ULN2003 IN1, IN2, IN3 e IN4 al pin digital del Arduino. Conecte el cable positivo de una batería de 5-12V a la patilla “+” del conductor ULN2003 y el pin “-” a GND. Asegúrese de que el puente “on / off” esté

conectado. Si conecta la placa Arduino de una batería diferente, conecte los GND juntos.

### 2.8.3 Códigos de programación para motor 28BYJ-48.

La librería AccelStepper ya viene pre-instalado con el Arduino IDE, pero sólo admite el método de paso completo limitando así algunas características del motor 28BYJ-48, actuando de manera muy eficiente, Para mejorar el rendimiento se dispone de la siguiente biblioteca AccelStepper. En donde se puede ejecutar los movimientos varios motores a la vez, de manera muy eficiente y también es compatible con la aceleración (que permite el paso a paso para llegar a una velocidad más alta). (Illamas, 2016)

## 2.9 Comunicación serial RS232 I2C.

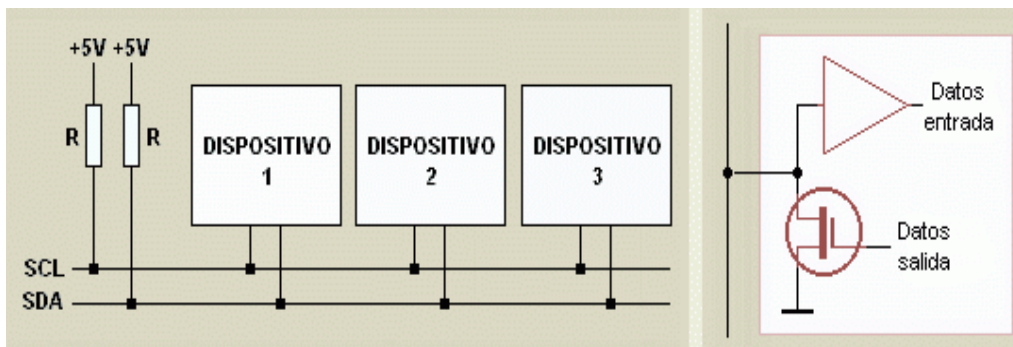
El bus de comunicación I2C, un estándar de comunicación que facilita la comunicación entre microcontroladores, memorias y otros dispositivos con cierto nivel de inteligencia, además solo se requiere de dos líneas de señal y un común o masa. Fue diseñado en los años de los 80', a este efecto se le conoce como efecto de Philips y permite el intercambio de información entre muchos dispositivos a una velocidad de 100 Kbits por segundo, aunque hay casos especiales en los que el reloj llega hasta los 3 MHz.

La metodología de comunicación de datos del bus de comunicación I2C es en serie y sincrónica. Una de las señales del bus marca el tiempo utilizado (pulsos de reloj) y la otra se utiliza para intercambiar datos entre sí.

- **Descripción de las señales.**

- **SCL** (System Clock), indica las señales de pulsos de reloj que sincronizan el sistema.
- **SDA** (System Data), indica los movimientos y señales que existen entre los datos y los otros los dispositivos conectados al sistema.
- **GND** (Masa) indica que es el común de la interconexión entre todos los dispositivos sujetos al bus.

Las señales SDA y SCL son del tipo colector abierto, es decir, un estado similar al de dren abierto, pero asociadas a un transistor de efecto de campo (FET). Obligatoriamente se deben polarizar en estado alto es decir 1 lógico (conectando a la alimentación resistencia en "pull-up") lo que indicara la conexión de los buses en sentido paralelo de múltiples entradas y salidas. (Carletti, 2006)



**Figura 17:** Diagrama interno de la comunicación I2C.

**Fuente:** (Carletti, 2006)

## CAPÍTULO III

### DESARROLLO DEL TEMA.

En el siguiente proyecto se detallan los procedimientos realizados para el desarrollo del tema, el mismo que es de gran utilidad para el personal de alumnos e instructores de la ETAE.

**CAMPO:** Electrónica & Aviónica

**ÁREA:** Aviónica

**TEMA:** Implementación de una maqueta didáctica 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.

**BENEFICIARIOS:** Escuela Técnica de Aviación del Ejército

**UBICACIÓN:** Quito, La Balbina

**INSTITUCIÓN EJECUTORA:** Unidad de Gestión de Tecnologías

**COSTO:** 1010 USD.

#### 3.1 Preliminares.

En el presente capítulo se detallará los procedimientos realizados, sobre la realización de la implementación de una maqueta didáctica 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.

### **3.2 Estudio de factibilidad.**

Para el estudio de factibilidad del presente proyecto técnico se consideró los siguientes factores; factor técnico y el factor socioeconómico.

#### **3.2.1 Factor técnico.**

Se refiere al estudio detallado de cada uno de los elementos que conforman el sistema de arranque/encendido de una aeronave, considerando las normas establecidas en los respectivos manuales, determinando el grado de dificultad del trabajo a realizarse, así como la operación del sistema. En tal virtud se considera factible, ya que se puede aprovechar la estructura frontal del panel de instrumentos, accesorio disponibles en stock en la sección de abastecimiento aéreo, del Centro de Mantenimiento de Aviación del Ejército (CEMAE-15), así como también la disponibilidad de equipos, herramientas especiales, localizados en los laboratorios de electrónica y lo más importante que se cuenta con asesoría de personal técnico capacitado, para la realización de la maqueta.

#### **3.2.2 Factor económico.**

Es necesario analizar los costos que generó los materiales utilizados para la implementación de la maqueta didáctica, que simula el sistema de arranque/ encendido de una aeronave, así también se consideró gastos personales necesarios durante el desarrollo del proyecto técnico.

### **3.3 Selección de información técnica.**

Para realizar la implementación del sistema de arranque/ encendido de una aeronave se empleó información de manuales y referencias bibliográficas citadas al final del proyecto técnico. Además se investigó en las páginas de internet sobre la secuencia de encendido/arranque de una aeronave.

### **3.4 Implementación de la maqueta didáctica.**

#### **3.4.1 Comprobación de materiales a utilizarse.**

Para realizar el funcionamiento de La maqueta didáctica se empezará inspeccionando cada uno de los elementos se encuentre en perfectas condiciones de funcionamiento y operación.

- Comprobar la continuidad de los cables de alimentación.
- Comprobar el voltaje de salida, en los pines que indica Vout, esta acción indica que los elementos electromecánicos se encuentra en perfecto estado de funcionamiento.

#### **3.4.2 Elaboración de la maqueta didáctica.**

Para la elaboración de la maqueta didáctica se debe tener las siguientes consideraciones:

- El uso adecuado de equipos de protección personal, tanto del sistema respiratorio, como de la piel al momento de realizar el pintado de la carcasa.
- El uso adecuado de equipos de comprobación eléctrica/ electrónica y de la manilla estática, los mismos que permitirán el buen desempeño de trabajo de los elementos electromecánicos.
- La utilización correcta de las plataformas virtuales (tales como de LabVIEW y arduino), el cual nos permitirán el correcto funcionamiento de las simulaciones y en si el funcionamiento de la maqueta didáctica.
- **Implementación de la estructura del panel de instrumentos.-** Para realizar la implementación de la maqueta se tuvo que canibalizar la estructura principal de instrumentos, luego realizar la remoción de pintura de la carcasa del panel frontal de instrumentos



**Figura 18:** Remoción de pintura de la carcasa del panel frontal de instrumentos.

Luego de la remoción de la pintura se procedió a pintar la estructura principal, posteriormente se elaboró del panel de instrumentos y de control, colocando de señalética y caratulas de instrumentos.



**Figura 19:** Elaboración del panel de instrumentos y colocación de señalética y caratulas de instrumentos.

Luego de la colocación de la señalética se procedió a la instalación y verificación de elementos electromecánicos (interruptores simples, dobles y triple). Posteriormente al chequeo de los elementos electromecánicos (electroválvula, switch de la bomba booster, switch de dos posiciones de ON, OFF, switch de tres posiciones de ON, RUN, OFF).





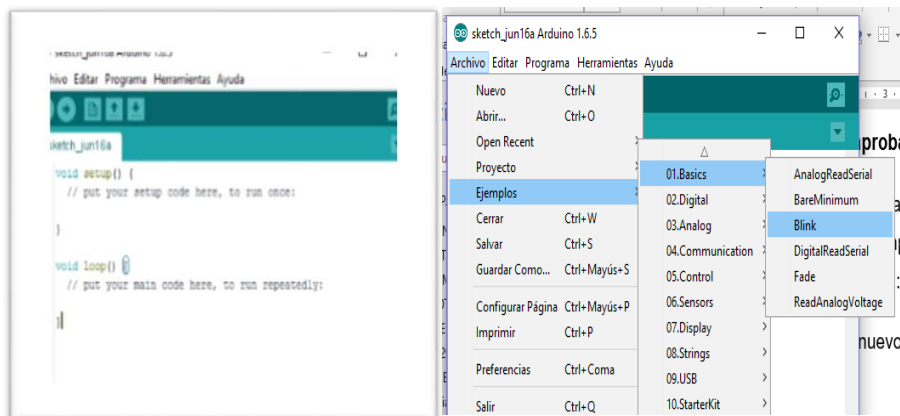
**Figura 20:** Instalación y verificación de elementos electromecánicos (interruptores simples, dobles y triple).

- **Comprobación de los elementos electromecánicos.-** Para conocer el estado de los elementos electromecánicos se debe realizar la comprobación funcional y la comprobación eléctrica poniendo operación dichos elementos para lo cual se realizara de la siguiente manera:
  - Comprobar el estado físico del elemento electromecánico.
  - Comprobar la continuidad de los cables de alimentación.
  - Alimentar los elementos electromecánicos (electroválvula, switch de la bomba bouster, switch de dos posiciones de ON, OFF, switch de tres posiciones de ON, INJ, OFF).
  - Chequeo y verificación de funcionalidad de los interruptores y switch.
  - Verificación de funcionamiento y control de los motores pasó a paso. (los motores estarán controlados con la tarjeta arduino mega )
  - Comprobar el voltaje de salida, en los pines que indica  $V_{out}$ .

### 3.5 Prueba de funcionalidad de Arduino Mega.

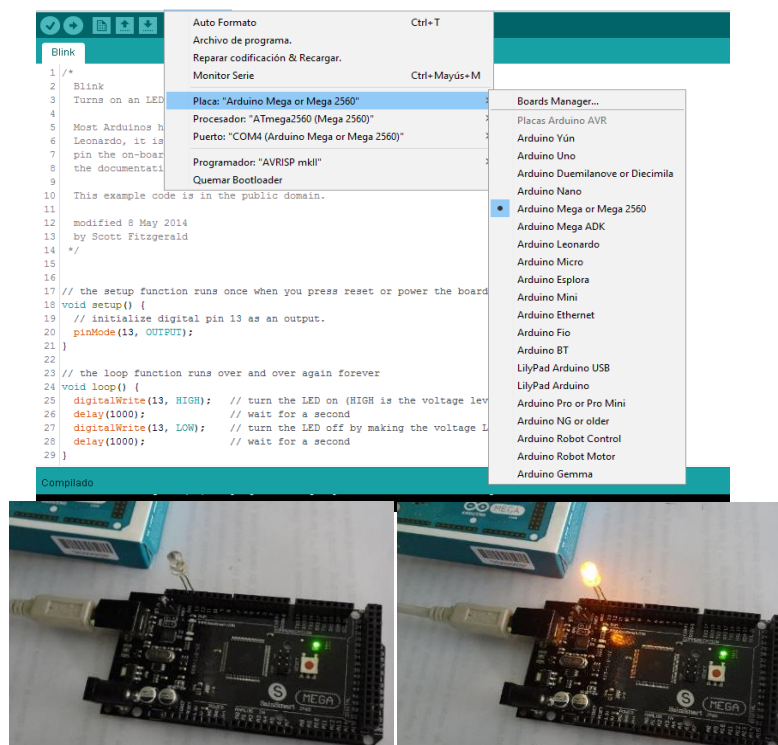
Para conocer el estado de funcionalidad de la tarjeta Arduino Mega, se debe cargar un ejemplo y así comprobar su funcionamiento, para lo cual se realiza de la siguiente manera:

Abrir un nuevo IDE en el software Arduino, luego nos dirigimos a Archivo, Ejemplos, Basic, donde se abrirá un Sketch, posterior a esto conectamos el arduino mega a la PC y seleccionamos la tarjeta y el puerto de comunicación en este caso COM23.



**Figura 21:** Comunicación entre el arduino y la PC.

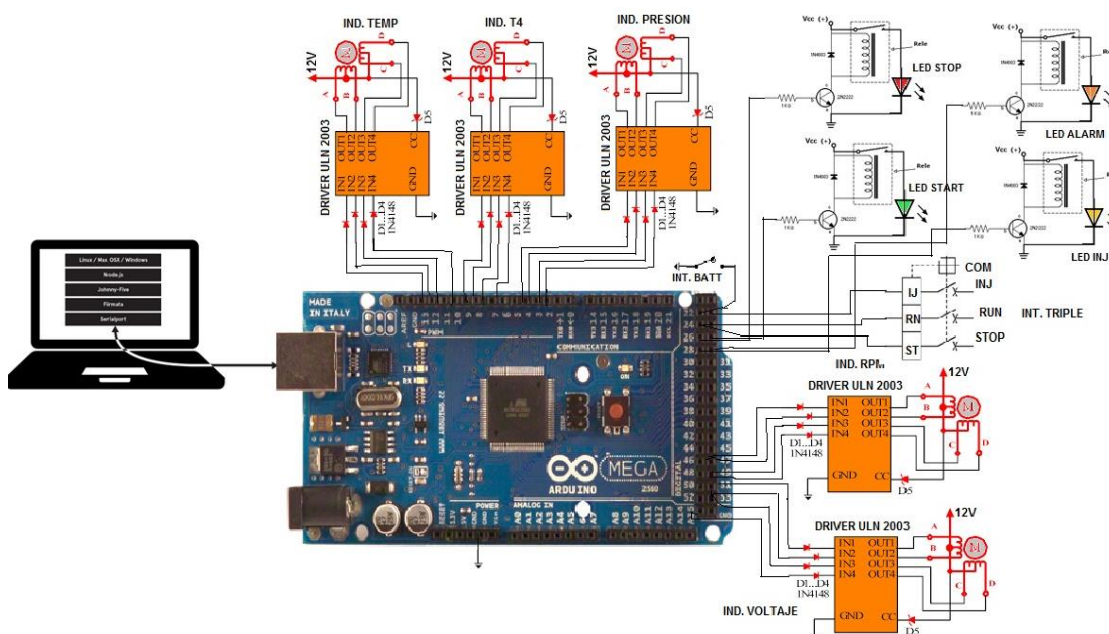
Posteriormente cargamos el Sketch a Arduino Mega y Conectamos un led al Pin de salida este deberá oscilar en un intervalo de tiempo, Esta acción indicara que la tarjeta Arduino Mega se encuentra en condiciones operables de uso, y se podrá compilar cualquier programa realizado.



**Figura 22:** Ejemplo de encendido y apagado de un led.

### 3.6 Programación de la tarjeta Arduino

Una vez incluidas las librerías en el software de Arduino así como también de la maqueta de forma física, se procede a desarrollar el código de programación para la tarjeta arduino, para esto se procede a realizar las conexiones de todos los elementos electromecánicos a la tarjeta arduino, para así realizar el control del proceso de encendido y arranque de una aeronave. A continuación se muestra la conexión:



**Figura 23:** Diagrama de conexiones de elementos electromecánicos

En primer lugar se conecta el arduino mega a la PC para la comprobación de funcionalidad del mismo y luego cargamos el programa anteriormente realizado en el IDE de arduino, luego que se termine el código de programación en el IDE, se procede a desarrollar una HMI en LabVIEW para crear una interfaz de comunicación entre el Arduino y la PC.

#### 3.6.1 Declaración de librerías.

Para la declaración de librerías se incluyen en el código de programación es la `#include <AccelStepper.h>` y declaramos el tipo de motor se va a utilizar

#define HALFSTEP 8, ambas son necesarias para programar la tarjeta arduino.

### 3.6.2 Declaración de variables.

En este caso se utilizara variables que permite controlar los motores paso a paso y a los elementos electromecánicos, y consecuentemente puedan tener una comunicación serial rs232 bidireccional (TX/RX datos).

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

### 3.6.3 Inicialización del programa.

Para empezar la programación en el entorno IDE, se empieza programando dentro de la estructura (void setup), en esta estructura se coloca el código de programación, que se ejecutara una sola vez dentro de un lazo abierto.

```
void setup()
{
```

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

En este caso se utiliza algunas variables para determinar las entradas que controlaran los interruptores simples y triples, consecuentemente las salidas que representaran en luces.

- Configuración de las entradas para control de motores paso a paso en la tarjeta arduino
- AccelStepper stepper1(HALFSTEP, 2, 3, 4, 5);  
AccelStepper stepper2(HALFSTEP, 6, 7, 8, 9);  
AccelStepper stepper3(HALFSTEP, 10, 11, 12, 13);

```
AccelStepper stepper4(HALFSTEP, 46, 47, 48, 49);
AccelStepper stepper5(HALFSTEP, 50, 51, 52,53);
```

- Configuración de entradas de la tarjeta arduino para realizar el control del interruptor triple

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

- Configuración de salidas de la tarjeta arduino para realizar el control de las luces

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

### 3.6.5 Estructura del control.

Para obtener un control automático de la programación se escribe en la estructura (void loop), dentro de esta estructura se coloca la programación que se va a ejecutar, así como las instrucciones de la plataforma que se van a estar ejecutándose de manera seguida, mientras el Arduino esté conectado a una fuente de alimentación o a una PC.

```
void loop()
{
{
```

### 3.6.6 Envío de datos por el puerto serial.

Debido a que se tiene que enviar y recibir datos de manera predeterminado en un mismo tiempo, se debe separar estos datos por

constantes que son identificados en el monitor serie, donde se envía cada valor junto con una letra y se los almacena en una memoria EEPROM, creando así una nueva variable tipo string.

**Serial.println("a");**

### 3.6.7 Control del interruptor de batería.

Las siguientes líneas de programación permiten controlar el indicador de voltaje y las luces STAR, STOP, INJ, ALARMA mediante la activación de relés que da paso al módulo ULN2003, el cual permite que el motor gire en sentido horario y en sentido antihorario según la señal recibida, simulando así el instrumento deseado y el control de dichas luces.

- Programación y control del interruptor de batería

```

if(digitalRead(bateria)==LOW && sw1==0)
{
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);
}
stepper5.moveTo(1000);
a=stepper5.distanceToGo();
while(a!=0)
{
sw1=1;
stepper5.run();
a=stepper5.distanceToGo();
}
}

```

```

//Bateria Apagada
if(digitalRead(bateria)==HIGH && sw1==1)
{
    digitalWrite(ledSTOP,HIGH);
    digitalWrite(ledSTART,HIGH);
    digitalWrite(ledALARM,HIGH);
    digitalWrite(ledINJ,HIGH);
    stepper5.moveTo(-stepper5.currentPosition());
    a=stepper5.distanceToGo();
    while(a!=0)
    {
        sw1=0;
        stepper5.moveTo(-stepper5.currentPosition());
        stepper5.run();
        a=stepper5.distanceToGo();
    }
}

if (sw1==1)

}

```

### 3.6.8 Control del interruptor triple posición OFF.

Las siguientes líneas de programación permiten controlar lo siguiente; indicador de presión del aceite del motor, indicador de temperatura del aceite del motor, el indicador de temperatura del motor (T4), indicador de RPM del motor y las luces STAR, STOP, INJ, ALARMA, en donde el arduino envía una señal para que los motores paso a paso no realicen ninguna acción, simulando así el apagado del sistema.

- Programación del Interruptor Start posición OFF.
 

```

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)
{
  stepper1.moveTo(-stepper1.currentPosition());
  b=stepper1.distanceToGo();
  stepper2.moveTo(-stepper2.currentPosition());
  c=stepper2.distanceToGo();
  stepper3.moveTo(-stepper3.currentPosition());
  d=stepper3.distanceToGo();
  stepper4.moveTo(-stepper4.currentPosition());
  e=stepper4.distanceToGo();
  while(b!=0)
  {
    stepper1.moveTo(-stepper1.currentPosition());
    b=stepper1.distanceToGo();
    stepper1.run();
    stepper2.moveTo(-stepper2.currentPosition());
    c=stepper2.distanceToGo();
    if (c!=0)
    stepper2.run();
    stepper3.moveTo(-stepper3.currentPosition());
    d=stepper3.distanceToGo();
    if(d!=0)
    stepper3.run();

    stepper4.moveTo(-stepper4.currentPosition());
    e=stepper4.distanceToGo();
    if(e!=0)
    stepper4.run();
```



```

    }
    sw2=0;
  }
}

```

### 3.6.9 Control del interruptor triple posición RUN.

Las siguientes líneas de programación permiten controlar el movimiento sincronizado de los siguientes instrumentos; indicador de presión del aceite del motor, indicador de temperatura del aceite del motor, el indicador de temperatura del motor (T4), indicador de RPM del motor y el control de las luces STAR, STOP, INJ, ALARMA, en donde el Arduino envía una señal para que los motores paso a paso continúen el movimiento, simulando así el encendido de una aeronave.

- Programación del Interruptor 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");
    sw3=0;
  }
  stepper1.moveTo(3000);
  stepper2.moveTo(1200);
  stepper3.moveTo(750);
  stepper4.moveTo(500);
  stepper1.run();
  stepper2.run();
  stepper3.run();

```

```

stepper4.run();
sw2=1;
}

```

### 3.6.10 Control del interruptor triple posición INJ.

En la siguiente programación permiten continuar con el control del movimiento sincronizado de los siguientes instrumentos; indicador de presión del aceite del motor, indicador de temperatura del aceite del motor, el indicador de temperatura del motor (T4), indicador de RPM del motor y el control de las luces STAR, STOP, INJ, ALARMA, en donde la tarjeta arduino envía una señal para que los motores paso a paso continúen el movimiento, simulando así el encendido de una aeronave.

- Programación del Interruptor 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");
sw3=1;
}
stepper1.run(); stepper2.run(); stepper3.run(); stepper4.run();
}
}

```

### 3.6.11 Configuración de la comunicación serial.

Para realizar una comunicación entre Arduino y PC, se utiliza la comunicación serial RS232, de esta manera se podrá enviar y recibir datos a

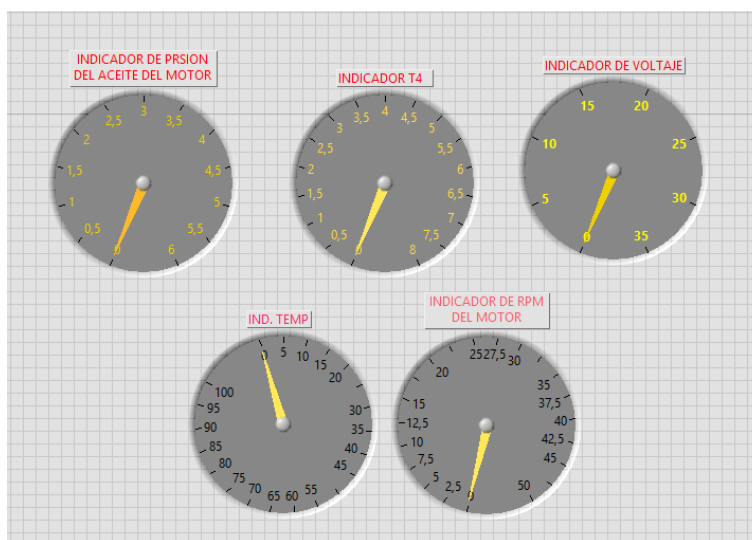
una velocidad determinada en la programación en la tarjeta arduino, en la cual se utiliza la velocidad transmisión de 9600 baudios.

### 3.7 Programación en el software LabVIEW.

Terminada la programación en el entorno de arduino IDE, se procede a cargar en la tarjeta Arduino, posteriormente se procede a crear el interface humano maquina en el panel frontal de LabVIEW, luego se programara en el diagrama de bloques LabVIEW.

#### 3.7.1 Diseño del HMI en el panel frontal LabVIEW.

Para iniciar con el interface humano maquina colocamos los elementos e indicadores necesarios en el panel frontal, los cuales representan los instrumentos de la maqueta didáctica en las cuales se podrá monitorear el proceso de encendido/arranque de una aeronave. Colocando así 5 indicadores quienes simularan instrumentos reales en el proceso de encendido y arranque, estos indicadores se pueden encontrar en la paleta de controles en el diagrama de bloques.



**Figura 24:** Indicadores de; Voltaje, Presión, T4, Temperatura y RPM.

Luego se ubicara dos controles de desplazamiento ON/OFF, OFF/RUN/INJ en el panel frontal, los mismos que se encuentran en la paleta de controles, los mismo que representaran los interruptores de batería y el interruptor triple.



**Figura 25:** Controles de desplazamiento ON/OFF.

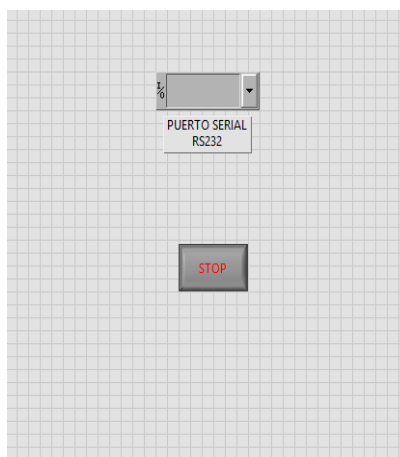
Luego colocar cuatro leds que se encuentran en la paleta de controles, para representar las luces START, STOP, ALARMA y INJ, que nos indicaran la simulación del encendido.



**Figura 26:** Indicadores de ALARM, START, STOP e INJ.

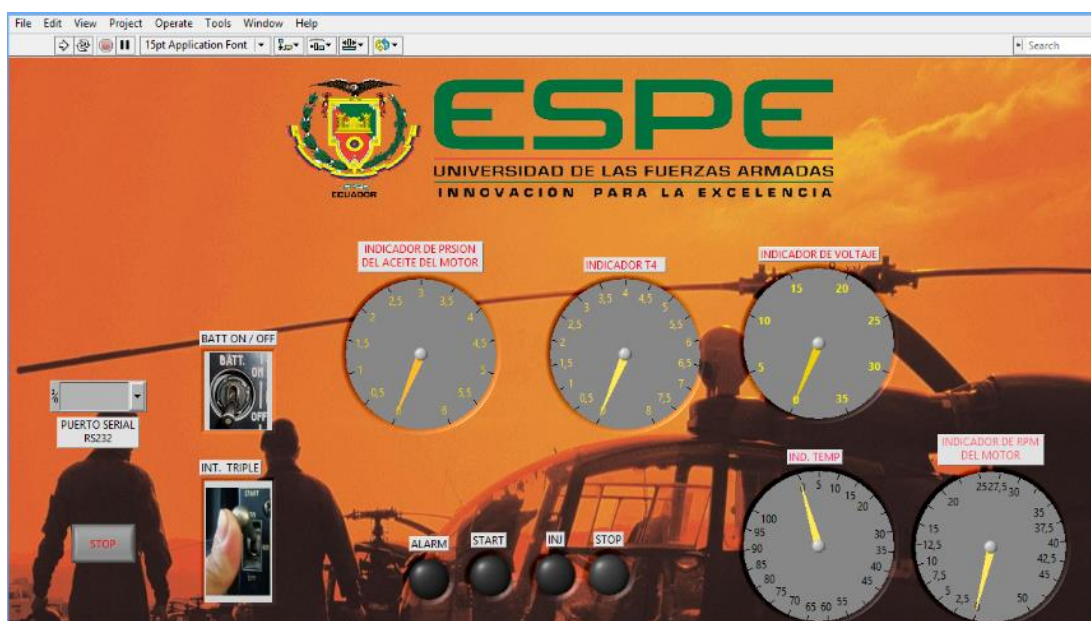
Finalmente se coloca un puertos COM, para seleccionar el puertos de comunicación RS 232 y a la vez se comunice serial mente con la PC, luego cuando se conecte el Arduino a la PC, el VI del puerto aparecerá automáticamente, asi quedaran en comunicación serial el arduino con la PC.

Además colocamos un botón de tipo booleano para utilizarlo como paro del proceso.



**Figura 27:** Paro del proceso y selector de puerto COM.

Una vez que se tiene todos los controles y los indicadores se proceden a colocar todo de una manera ordenada, para finalizar el interface humano máquina, se coloca advertencias del tipo control para manipular la secuencia de encendido/arranque.

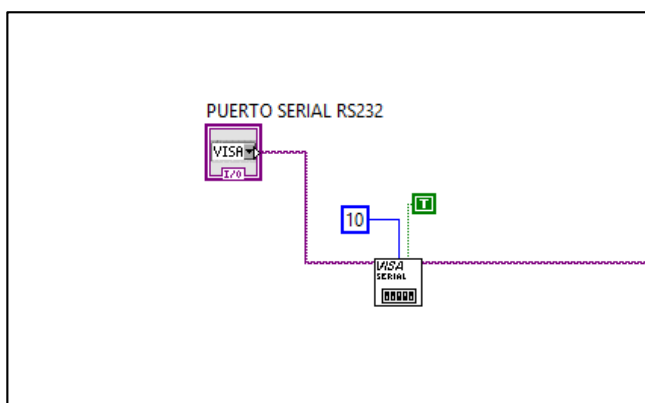


**Figura 28:** Monitoreo el encendido y arranque de una aeronave.

### 3.7.2 Programación en el diagrama de bloques de LabVIEW.

Para efectuar la programación en el diagrama de bloques, se realiza una interface humano máquina, entre los elementos instalados en el panel frontal y el Arduino mega, además se debe realizar las conexiones entre los controles y los indicadores en el diagrama de bloques.

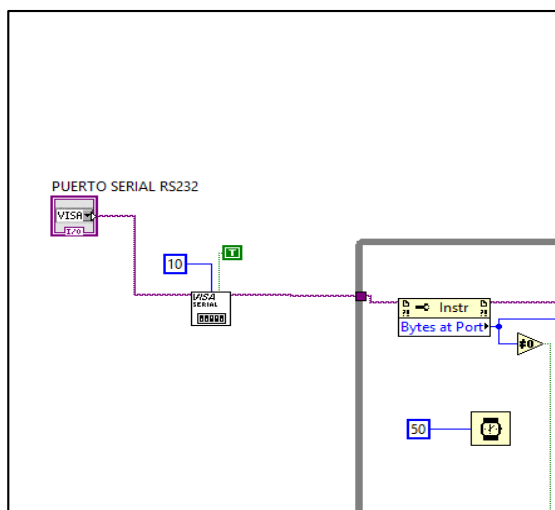
Para realizar la comunicación serial RS232 entre el software LabVIEW y el Arduino, se debe realizar la configuración del puerto serial, colocando iconos de VISA Configure y un Serial Port, ubicada en la sección Instruments de la paleta funciones. En estos iconos se coloca un control para registrar el puerto COM.



**Figura 29:** Configuración del puerto serial y VISA

Para examinar los datos que ingresan y salen a través del puerto serial VISA, se procede a configurar el puerto como lectura y escritura, para la lectura se utiliza la opción VISA Read y para la escritura se utiliza la opción VISA written

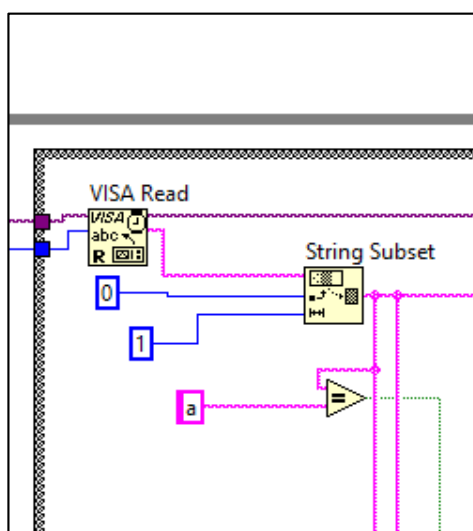
Para que los datos extraídos sean leídos en códigos de máquina se debe instalar el icono Property Node, el cual se conecta al count del VISA Read para determinar la cantidad y tamaño de bits que ingresan a través del puerto VISA, para registrar estos datos se debe dar clic sobre Serial Settings Number of Bytes at Serial Port.



**Figura 30:** Configuración del puerto serial

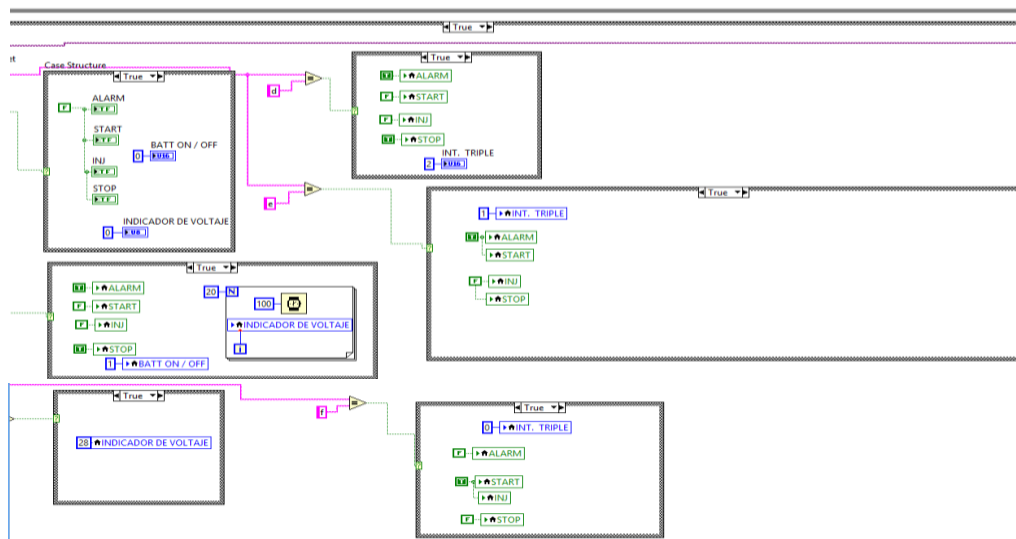
Los datos recibidos se deben procesar utilizando el icono String subset, el cual toma los datos de tipo String para luego separar el valor numérico de cada constante recibida (letras; a, b, c, d, e, f) y presentar a su salida únicamente el valor numérico.

Una vez recibido los datos correctos procedemos a insertar cada uno de las variables que se va utilizar en la programación, como son las de batería, interruptor triple, indicador de voltaje, indicador de presión del motor, indicador de temperatura del aceite del motor, la temperatura del motor, indicador de RPM, luces START, STOP, ALARM y INJ.



**Figura 31:** Conversión de decimal String a carácter Numérico

Luego creamos variables locales para insertar en cada uno de los case structure, el cual preguntara si es verdadero o falso; si es verdadero realiza la acción programada en un lazo cerrado, si es falso retorna a verificar los datos recibidos.



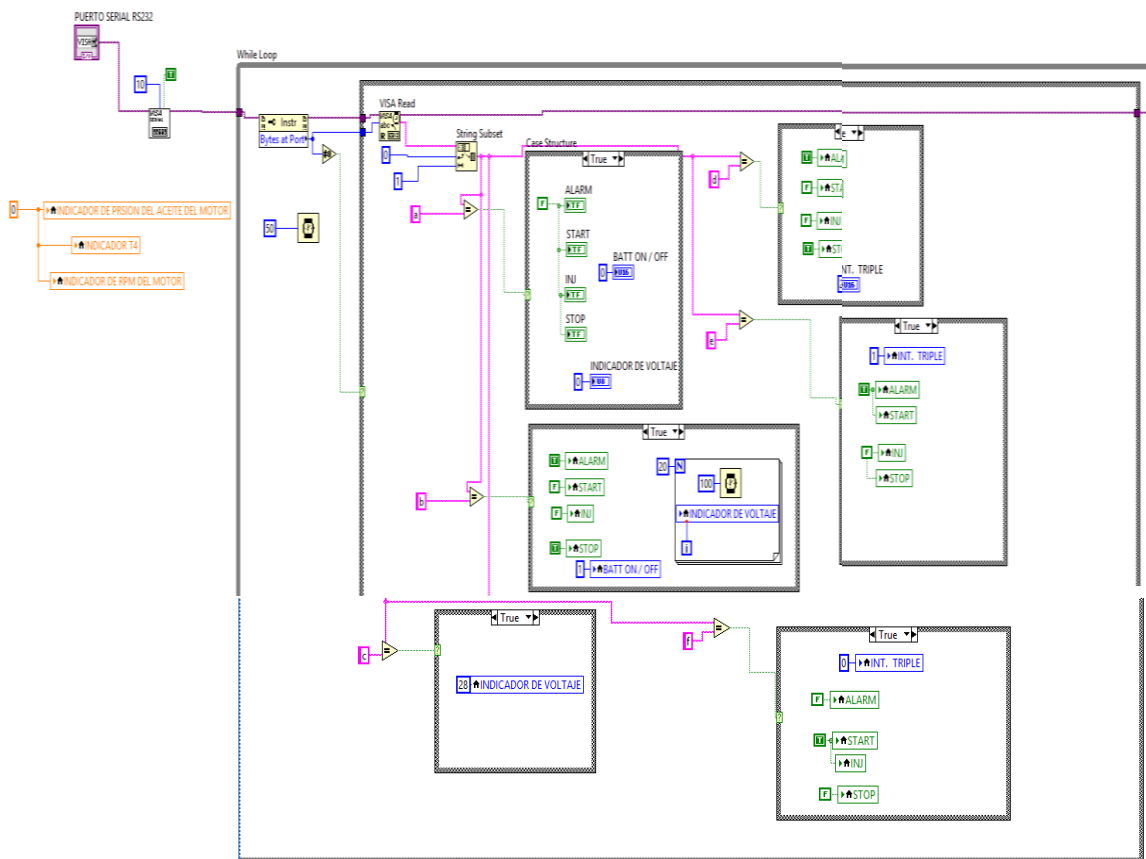
**Figura 32:** Variables locales en la case structure

Para enviar/recibir datos se utiliza una estructura de casos en el diagrama de bloques, para comparar el estado lógico de los indicadores según los datos enviados desde el Arduino, para esto se utilizan estructuras de casos, para el indicador del voltaje, indicadores de presión y temperatura del aceite del motor, el indicador de temperatura del motor y el indicador de RPM del motor, además se utiliza una estructura de caso de tipo booleano para enviar una señal la cual se utiliza para detener todo el proceso.

Para enviar/recibir los datos del tipo String (letras), se debe definir que esos datos recibidos/enviados sean los correctos en la estructura de programación en el IDE de arduino, ya que este nos permitirá realizar la comunicación serial, además para la salida de datos se utiliza estructura de casos controlada por un while loop, quien permite que el proceso continúe hasta que no se envíe otra orden al lazo abierto de while loop.



Para finalizar los procesos se debe incluir los iconos Port y de los VISA Close, dentro de la estructura While Loop que repite la programación cíclicamente.



**Figura 33:** Programación en el diagrama de bloques de LabVIEW.

### 3.8 Pruebas de funcionamiento de la maqueta didáctica.

Se realizaron las pruebas de funcionalidad de la maqueta siguiendo las normas de procedimiento establecidos en los manuales de operación y de los manuales vuelo, los mismos que establecen una lista de chequeo antes durante y después del encendido/ arranque de una aeronave.

El encendido/arranque se va a dar en una sola situación con batería incorporada y se verifica el monitoreo en el panel frontal de labwiev.



**Figura 34:** Pruebas de funcionalidad del interruptor batt, en la maqueta



**Figura 35:** Pruebas de funcionalidad del interruptor batt, en el panel Frontal.

Luego procedemos a iniciar la simulación el encendido/arranque, tanto en la maqueta como el monitoreo del proceso de encendido en labwiev (panel frontal). Colocamos el interruptor triple en posición RUN y se enciende las siguientes; luces; start, alarma y bomba.



**Figura 36:** Secuencia de encendido /arranque, interruptor triple en RUN en la maqueta



**Figura 37:** Secuencia de encendido /arranque, interruptor triple en RUN en LabVIEW (panel frontal).

Continuando con la secuencia de encendido procedemos a colocar el interruptor triple en INJ que indica que se va inyectar combustible y que los instrumentos del panel continúan su movimiento hasta llegar a su punto máximo como el T4 llegue hasta los 450°C o que el indicador de RPM llegue hasta los 9000 RPM.



**Figura 38:** Secuencia de encendido /arranque, interruptor triple en INJ en la maqueta.



**Figura 39:** Secuencia de encendido /arranque, interruptor triple en INJ en labwiev (panel frontal).

## CAPITULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1. 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 brindará la maqueta didáctica en un entorno aeronáutico, de esta manera se desarrolló el proceso automatizado del sistema de 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 máquina 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.

#### **4.2. 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 ya que la inobservancia de estas pueden ocasionar incidentes o inhabilitación de la maqueta didáctica.
- Mantener la conexión del Arduino del puerto COM de la computadora, ya que este puerto se utilizó como enlace entre el entorno de desarrollo IDE de arduino y el software LabVIEW, además este puerto establece la comunicación entre los dos elementos; arduino y LabVIEW.
- 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.



## GLOSARIO DE TÉRMINOS

### A

**AC.** Corriente Alterna.

**ACTUADOR.** Dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado.

**ALGORITMO.** Conjunto ordenado de operaciones sistemáticas que permite hacer un cálculo y hallar la solución de un tipo de problema.

**ASÍNCRONO.** No tiene un intervalo de tiempo constante entre cada evento.

**AUTÓNOMO.** Trabaja de forma independiente.

### B

**BUCLE.** Sentencia que se realiza repetidas veces a un trozo aislado de código.

### C

**CC.** Corriente Continua.

**CODIFICADOR.** Circuito combinacional con  $2^N$  entradas y N salidas, cuya misión es presentar en la salida el código binario correspondiente a la entrada activada.

**CPU.** Unidad Central de Procesamiento.

### E

**EEPROM.** Siglas de Electrically Erasable Programmable Read-Only Memory (ROM programable y borrable eléctricamente).

**ENCAPSULADO.** Resultado de la etapa final del proceso de fabricación de dispositivos con semiconductores.

### G

**GANANCIA.** Magnitud que expresa la relación entre la amplitud de una señal de salida respecto a la señal de entrada.

**GND (Masa)** común de la interconexión entre todos los dispositivos "enganchados" al bus.

### H

**HZ.** Unidad de medida de frecuencia.

**HMI.** Interfaz humano maquina

**I**

**I2C.** Inter-Integrated Circuit, es un bus de datos serial desarrollado en 1982 por Philips Semiconductor.

**IDE.** Entorno de Desarrollo Integrado.

**IMU.** Unidad de Medida Inercial.

**INTERFAZ.** Conexión funcional entre dos sistemas o dispositivos de cualquier tipo dando una comunicación entre distintos niveles.

**L**

**LIBRERÍAS.** Conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

**M**

**MEMS.** Sistemas Micro electromecánicos.

**N**

**NIVEL LÓGICO.** Circuitos en los que solo existen dos estados posibles, alto y bajo.

**O**

**OSCILADOR.** Sistema capaz de crear perturbaciones o cambios periódicos en un medio.

**P**

**POLARIDAD.** Propiedad de los terminales (polos) de una batería o de una pila, que pueden ser positivos o negativos.

**PWM.** Modulación por ancho de pulsos (pulse-width modulation).

**R**

**RESOLUCIÓN.** Longitud de la palabra digital.

**S**

**SENSOR.** Objeto capaz de detectar magnitudes físicas o químicas.

**SÍNCRONO.** Sistema de comunicación, el transmisor debe coordinarse con el receptor antes del envío de datos.

**SCL** (System Clock) es la línea de los pulsos de reloj que sincronizan el sistema.

**SDA** (System Data) es la línea por la que se mueven los datos entre los dispositivos.



U

**UARTs.-** (Universal Asynchronous Receiver-Transmitter), en español: Transmisor-Receptor Asíncrono Universal, es el dispositivo que controla los puertos y dispositivos serie.

V

**VALOR ABSOLUTO.-** Es el número natural que resulta al suprimir su signo.

**INTERFAZ ES.-** Conexión, física o lógica, entre una computadora y el usuario, un dispositivo periférico o un enlace de comunicaciones.

**VELOCIDAD DE CRUCERO.-** velocidad que mantiene un vehículo terrestre, embarcación o aeronave durante la mayor parte de un determinado recorrido.

## ABREVIATURAS

**ETAE:** Escuela Técnica de Aviación del Ejército.

**CEMAE:** Centro de mantenimiento de Aviación del Ejército.

**G.A.E:** Grupo Aéreo del Ejército I.P.C: Catálogo Ilustrado de Partes.

**M.O:** Manual de Operación.

**M.M:** Manual de Mantenimiento.

**VIS.-** Instrumentos Virtuales.

**M.S:** Manual de Seguridad.

**E.P.P:** Equipo de Protección Personal.

**R.P.M.:** Revoluciones por minuto

**HMI:** Interface humano maquina

**MEMS.** Sistemas Micro electromecánicos

**SCL:** System Clock

**SDA:** System Data

**UARTs:** Universal Asynchronous Receiver-Transmitter

**AC:** Corriente Alterna.

## REFERENCIAS BIBLIOGRÁFICAS

- amazon. (s.f.). *www.amazon.com*. Obtenido de *www.amazon.com*:  
<http://www.amazon.com/Set-piece-IC-AD620-AD620AN/dp/B00CKNLC3O>
- ARDUINO. (23 de Julio de 2016). *arduino.cc*. Obtenido de *arduino.cc*:  
<http://arduino.cc/en/Main/ArduinoBoardMega2560>
- arduino, E. c. (05 de 2015). *elcajondeardu.blogspot.com*. Obtenido de *elcajondeardu.blogspot.com*:  
<http://elcajondeardu.blogspot.com/2015/05/tutorial-motor-paso-paso-con-modulo.html>
- ARDUINO.CL. (08 de Julio de 2016). *Arduino-cl*. Obtenido de *Arduino-cl*:  
<http://arduino.cl/arduino-mega-2560/>
- ares.cnice. (19 de 05 de 2009). *ares.cnice.mec.es*. Recuperado el 26 de 04 de 2015, de *ares.cnice.mec.es*:  
<http://ares.cnice.mec.es/informes/17/contenido/31.htm>
- chapingo. (23 de 0 de 2013). *www.chapingo.mx*. Recuperado el 13 de 08 de 2015, de *www.chapingo.mx*:  
<http://www.chapingo.mx/dima/contenido/tesis2011/tesisEstrada.pdf>
- culturacion. (2014). *www.culturacion.com*. Obtenido de *www.culturacion.com*: <http://www.culturacion.com/como-limpiar-una-fuente-de-poder/>
- ele.uva. (12 de 4 de 2013). *www.ele.uva.es*. Obtenido de *www.ele.uva.es*:  
[http://www.ele.uva.es/~lourdes/docencia/Master\\_IE/Acondicionamiento.pdf](http://www.ele.uva.es/~lourdes/docencia/Master_IE/Acondicionamiento.pdf)
- Electronica, D. d. (s.f.). *HOJAS DE CARACTERÍSTICAS* . Obtenido de *Circuito Funcionalidad*:  
[http://www.uhu.es/adoracion.hermoso/sist\\_digit/documentos/Hoja\\_caract\\_Cl.pdf](http://www.uhu.es/adoracion.hermoso/sist_digit/documentos/Hoja_caract_Cl.pdf)
- electrotecnia. (2013). *ocw.usal.es*. Obtenido de *ocw.usal.es*:  
[http://ocw.usal.es/eduCommons/enseñanzas-tecnicas/electronica/contenido/electronica/Tema4\\_Falimentac.pdf](http://ocw.usal.es/eduCommons/enseñanzas-tecnicas/electronica/contenido/electronica/Tema4_Falimentac.pdf)
- FADEASE. (2015). Argentina: Fadeasa.
- google. (s.f.). *conectoresfisicoscomputadora*. Obtenido de *conectoresfisicoscomputadora*:  
<https://www.google.com.ec/search?q=tipos+de+conectores+fisicos+computadora&safe>
- gruposip. (19 de 01 de 2012). *www.gruposip.com/pdfs*. Recuperado el 26 de 04 de 2015, de *www.gruposip.com/pdfs*:

<http://www.gruposip.com/pdfs/5%20Centro%20de%20carga%20Norma%20Nema%20Bticino%20Tiven.pdf>

industry.siemens. (06 de 04 de 2013). *cache.industry.siemens.com*. Recuperado el 09 de 06 de 2015, de *cache.industry.siemens.com*: [https://cache.industry.siemens.com/dl/files/461/16527461/att\\_82567/v1/Logo\\_s.pdf](https://cache.industry.siemens.com/dl/files/461/16527461/att_82567/v1/Logo_s.pdf)

infopl. (23 de 07 de 2012). *www.infopl.net*. Recuperado el 28 de 07 de 2015, de *www.infopl.net*: <http://www.infopl.net/descargas/105-siemens/automatas/s7-1500/2316-simatic-s7-1500-tia-portal>

ingenieriaymantenimiento. (2013). *ingenieriaymantenimiento.blogspot.com*. Recuperado el 2015, de *ingenieriaymantenimiento.blogspot.com*: [ingenieriaymantenimiento.blogspot.com/.../niveles-de-mantenimiento](http://ingenieriaymantenimiento.blogspot.com/.../niveles-de-mantenimiento)

llamas, I. (13 de Agosto de 2016). *www.luisllamas.es*. Obtenido de *www.luisllamas.es*: <https://www.luisllamas.es/motor-paso-paso-28byj-48-arduino-driver-uln2003/>

*manualvuelo*. (04 de Diciembre de 2014). Obtenido de *manualvuelo*: <http://www.manualvuelo.com/SIF/SIF35.html>

National Instruments. (2014). *www.LabVIEW Measurement Manua.es*. Obtenido de *www.LabVIEW Measurement Manua.es*: <http://www.LabVIEW Measurement Manual, National Instruments Corporation, edición de Julio de 2000>.

Rodrigo Nevado Anton, S. G. (s.f.). *SUMADORES*. Obtenido de Sumador completo de 4 bits: [http://personales.unican.es/manzanom/Planantiguo/EDigital/Sum\\_G5\\_08.pdf](http://personales.unican.es/manzanom/Planantiguo/EDigital/Sum_G5_08.pdf)

Santos, C. (s.f.). *METAMANTENIMIENTO ELECTRÓNICO E INSTRUMENTAL INDUSTRIAL PRODUCCIÓN Y TRANSFORMACIÓN*. Obtenido de Sumador completo de 4 bits: <http://es.slideshare.net/Cesarsantoss/sumador-completo-de-4-bits-9695438>

SISTEMAS DE ARRANQUE. (07 de febrero de 2011). *sistemas de la aeronave*. Argentina: aeromasther. Obtenido de *sistemasdelaaeronave*: <http://sistemasdelaaeronave.blogspot.com/2011/02/sistema-de-encendido.html>

# ANEXOS

## ÍNDICE DE ANEXOS

**ANEXO A.** Lista de chequeos de los procedimientos antes y durante del arranque.

**ANEXO B.** Descripción de las especificaciones técnicas del motor 28BYJ-48.

**ANEXO C.** Programación del sistema de arranque de una aeronave en la plataforma de arduino (IDE).

## ANEXO A

Lista de chequeos de los procedimientos antes y durante del arranque.

<b>LISTA DE CHEQUEOS</b>	
<b>GAZELLE SA – 432 L</b>	
<b>3. PROCEDIMIENTOS ANTES Y DURANTE EL ARRANQUE</b>	
Interruptor de batería	Posición ON
Interruptor de generador	Posición ON
Luces encendidas arranque de batería	ALARM, STOP
Bomba Booster	Encendida ON
Después de 20 segundos	
Selector de arranque	Posición "RUN" (Marcha)
Luz verde "START"	Encendida
Selector de arranque "IGN"	De 1500 a 2000RPM NG
Luz "IGN" encendida	
Mantener hasta 14=450° C o 9000 rpm	
Soltar selector de arranque	Luz IGN se apaga
Motor estabilizado a 25500 rpm	
<b>OBSERVACIONES A CONSIDERAR</b>	
Si T4 alcanza 700°C	Cortar la bomba Booster
Selector de arranque en "OFF" si:	Luz roja "STOP" se enciende
	Luz "START" se enciende
	Luz "IGN no enciende en 10"
	T4 no sube 10 "después del encendido de luz IGN"
	T4 > 700°C durante 5", o alcanza 750°C (sobre temp.)
Si la turbina se apaga (caída T4) Nota: durante esta operación la T4 no debe sobrepasar los 550°C.	Selector de arranque en "IGN"
	Si re inyección comienza antes de 9 rpm parar a 9000 rpm
	Si comienza después de 9000rpm a 3" después
Verificar que la luz "START" se apague	Entre 18000 y 24000 rpm

Lista de chequeos de los procedimientos antes y durante del arranque con batería

**PRECAUCIÓN:**

Después de un arranque sin éxito, no intente arrancar o encender más de tres veces consecutivas, para evitar sobrecalentamiento del motor de arranque-generador. Después de tres intentos sin éxito espere 20 minutos antes de intentarlo nuevamente.

<b>ARRANCADO</b>		
<b>OPERACIÓN</b>	<b>CHEQUEOS</b>	<b>ACCIONES SON NECESARIAS</b>
Verifique la BOMBA BOOSTER "ON"	La BOMBA deberá ser escuchada operando. Si existe duda, verifique que el amperímetro indique un aumento de carga	Coloque el interruptor Selector de Arranque a la posición Desconectado, "OFF" en los siguientes casos:
Luego de 20 segundos suelte el Selector de Arranque CORRIDA "RUN"	Se enciende la luz indicadora de color Verde "START". La AGUJA indicadora (Ng.) del TACOMETRO inicia su movimiento.	1. Luz roja de "STOP" SE ENCIENDE. 2. La luz de Aviso Verde "START" permanece apagada "OFF" luego de 10".
DE 1500 A 2000 RPM suelte el Selector de Arranque "IGN" y mantenga en esta opción hasta que aumente la 450° C o la velocidad del motor Llegue a 9000 R.P.M.	<p><b><u>Algunos segundos después:</u></b></p> <p>La luz de Aviso Amarilla "INJ" "INYECCIÓN" se enciende (llave de combustible Eléctrica Abierta)</p> <p><b><u>Luego:</u></b></p> <ul style="list-style-type: none"> <li>• La luz de Aviso INYECCIÓN "INJ" se apaga (fin de inyección y secuencia de arranque).</li> <li>• Verificar el incremento de RPM (Ng) y controlar la temperatura T°4.</li> </ul>	<p>3. La temperatura T° 4 excede: - 700° C por 5 segundos - 750° C e instantánea</p> <p>4. Luz de aviso Amarilla "INJ" no se enciende luego de 10".</p> <p>5. Luz de Aviso Amarilla "INJ" permanece iluminada durante más de 10 segundos.</p> <p>6. Si T°4 no aumenta luego de 10" de encendida la luz "INJ".</p> <p>7. Si la luz "FIRE" se enciende (ver. Cap. 3.3.1.)</p>

**OBSERVACIONES PARTICULARES:**

- No efectuar más de 3 tentativas consecutivas de arranque o ventilación.
- Después de 3 tentativas infructuosas, dejar pasar 20´ antes de tratar nuevamente.
- No efectuar más de dos embragues en menos de 5´ de intervalo.
- No soltar el cíclico sin que el paso general este asegurado.

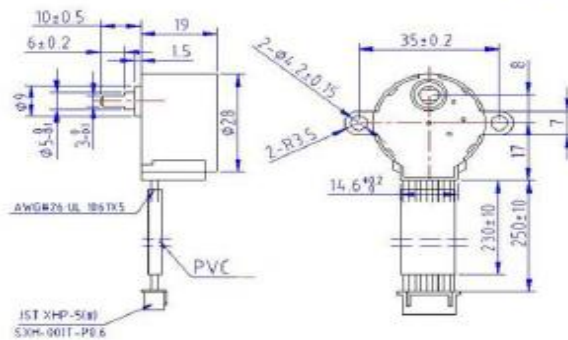
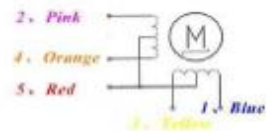


## ANEXO B

Descripción de las especificaciones técnicas del motor 28BYJ-48

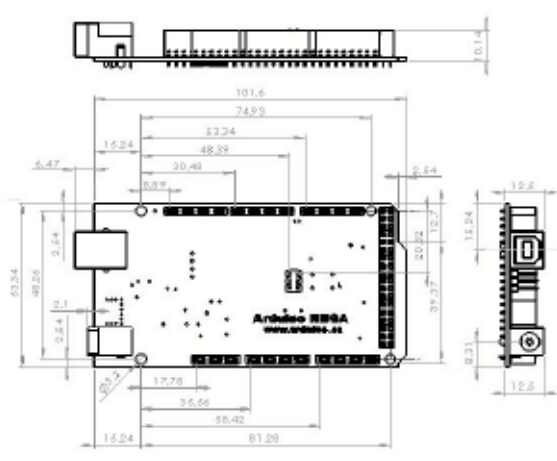
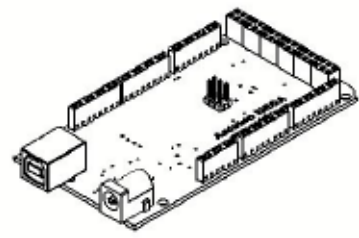
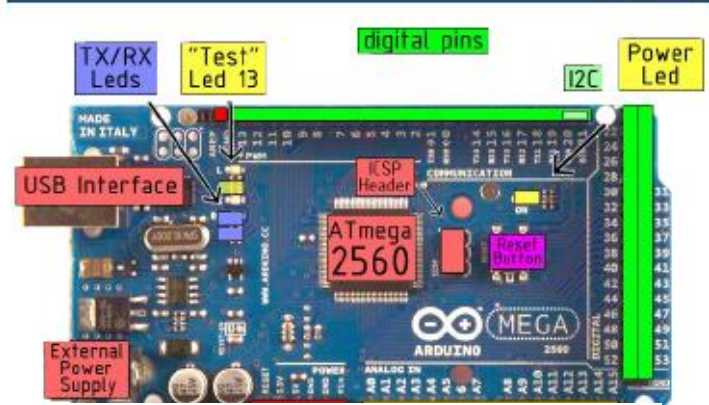


Rated voltage :	5VDC
Number of Phase	4
Speed Variation Ratio	1/64
Stroke Angle	5.625°/64
Frequency	100Hz
DC resistance	500±7% (25°C)
Idle In-traction Frequency	> 600Hz
Idle Out-traction Frequency	> 1000Hz
In-traction Torque	>34.3mN.m (120Hz)
Self-positioning Torque	>34.3mN.m
Friction torque	600-1300 gf.cm
Full In torque	300 gf.cm
Insulated resistance	>10MΩ (500V)
Insulated electricity power	600VAC/1mA/1s
Insulation grade	A
Rise in Temperature	<40K (120Hz)
Noise	<35dB (120Hz, No load, 10cm)
Model	28BYJ-48 - 5V



# Especificaciones técnicas del Arduino mega.

Technical Specification	
EAGLE files: <a href="#">_arduino-mega2560-reference-design.zip</a> Schematic: <a href="#">arduino-mega2560-schematic.pdf</a>	
Summary	
Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz



## ANEXO C

Programación del sistema de encendido/arranque de una aeronave en la plataforma de arduino.

```
#include <AccelStepper.h>
#define HALFSTEP 8
```

```
// Inicializar con la secuencia de encendido/arranque de una
aeronave en la cual se activa los siguientes pin IN1-IN3-IN2-IN4 para
usar el AccelStepper con 28BYJ-48.
```

```
//AccelStepper stepper5(HALFSTEP, 46, 48, 47,49);
```

```
AccelStepper stepper1(HALFSTEP, 2, 3, 4, 5);
AccelStepper stepper2(HALFSTEP, 6, 7, 8, 9);
AccelStepper stepper3(HALFSTEP, 10, 11, 12, 13);
AccelStepper stepper4(HALFSTEP, 46, 47, 48, 49);
AccelStepper stepper5(HALFSTEP, 50, 51, 52,53);
```

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

```
void setup() {
```

```
//Indicador de presion
```

```
stepper1.setMaxSpeed(1000.0);
stepper1.setAcceleration(10.0);
stepper1.setSpeed(200);
//stepper1.moveTo(3000);
```

```
//Indicador de temperatura
```

```
stepper2.setMaxSpeed(1000.0);
stepper2.setAcceleration(10.0);
stepper2.setSpeed(200);
```

```
//stepper2.moveTo(1200);
```

#### **//Indicador de temperatura T4**

```
stepper3.setMaxSpeed(100.0);  
stepper3.setAcceleration(5.0);  
stepper3.setSpeed(1000);  
//stepper3.moveTo(950);
```

#### **//Indicador de RPM**

```
stepper4.setMaxSpeed(100.0);  
stepper4.setAcceleration(5.0);  
stepper4.setSpeed(1000);  
//stepper4.moveTo(500);
```

#### **//Indicador de voltaje**

```
stepper5.setMaxSpeed(1000.0);  
stepper5.setAcceleration(1000.0);  
stepper5.setSpeed(1000);
```

```
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);
```

```
digitalWrite(ledSTOP,HIGH);  
digitalWrite(ledSTART,HIGH);  
digitalWrite(ledALARM,HIGH);  
digitalWrite(ledINJ,HIGH);
```

```
Serial.begin(9600);  
//--(end setup )---
```

#### **//EJECUCION DEL ENCENDIDO Y ARRANQUE**

```
void loop() {
```

#### **//Bateria encendida**

```
if(digitalRead(bateria)==LOW && sw1==0)  
{  
  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);
Serial.println("b");
}
```

```
stepper5.moveTo(1000);
a=stepper5.distanceToGo();
while(a!=0)
{
  sw1=1;
  stepper5.run();
  a=stepper5.distanceToGo();
}
Serial.println("c");
}
```

### **//Batería Apagada**

```
if(digitalRead(bateria)==HIGH && sw1==1)
{
  digitalWrite(ledSTOP,HIGH);
  digitalWrite(ledSTART,HIGH);
  digitalWrite(ledALARM,HIGH);
  digitalWrite(ledINJ,HIGH);
  stepper5.moveTo(-stepper5.currentPosition());
  a=stepper5.distanceToGo();

  Serial.println("a");
  while(a!=0)
  {
    sw1=0;
    stepper5.moveTo(-stepper5.currentPosition());
    stepper5.run();
    a=stepper5.distanceToGo();
  }
}
```

```
if (sw1==1)
{
```

### **//Interruptor Start posicion OFF**

```
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)
{
stepper1.moveTo(-stepper1.currentPosition());
b=stepper1.distanceToGo();
stepper2.moveTo(-stepper2.currentPosition());
c=stepper2.distanceToGo();
stepper3.moveTo(-stepper3.currentPosition());
d=stepper3.distanceToGo();
stepper4.moveTo(-stepper4.currentPosition());
e=stepper4.distanceToGo();

while(b!=0)
{
stepper1.moveTo(-stepper1.currentPosition());
b=stepper1.distanceToGo();
stepper1.run();

stepper2.moveTo(-stepper2.currentPosition());
c=stepper2.distanceToGo();
if (c!=0)
stepper2.run();

stepper3.moveTo(-stepper3.currentPosition());
d=stepper3.distanceToGo();
if(d!=0)
stepper3.run();

stepper4.moveTo(-stepper4.currentPosition());
e=stepper4.distanceToGo();
if(e!=0)
stepper4.run();
}
sw2=0;
}
}

```

### **//Interruptor Start posicion 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");
sw3=0;
}
stepper1.moveTo(3000);
stepper2.moveTo(1200);
stepper3.moveTo(750);
stepper4.moveTo(500);
stepper1.run();
stepper2.run();
stepper3.run();
stepper4.run();
sw2=1;
}

```

### **//Interruptor Start posicion 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");
sw3=1;
}
stepper1.run();
stepper2.run();
stepper3.run();
stepper4.run();
}
}
}

```

## HOJA DE VIDA



### DATOS PERSONALES:

---

**Nombres y apellidos:** Héctor Fidel Quinahuano Jitala  
**Documento de identidad:** 171793658 - 5  
**Fecha de nacimiento:** 3 de Diciembre de 1986  
**Lugar de nacimiento:** Parroquia Otón  
**Estado civil:** Soltero  
**Dirección:** Cayambe

**Teléfono:** 0996469660  
**E-mail:** ethofide@gmail.com

### FORMACIÓN ACADÉMICA:

---

**Estudios Primarios:** Unidad Educativa César Arroyo  
Dirección: Otón Cayambe  
**Estudios Secundarios:** Unidad Educativa Técnica Ascázubi  
Dirección: Ascázubi-Cayambe  
**Universitarios:** Escuela Politécnica del Ejercito  
Dirección: Ambato - Ecuador  
Año: 2008  
Título: Tecnólogo en Ciencias Militares  
**Idioma Extranjero:** Inglés  
Dominio del idioma hablado: Muy bueno  
Dominio del idioma escrito: Muy bueno

### CURSOS:

---

- Control industrial y redes eléctricas SECAP Año: 2004 200 horas

### FORMACIÓN MILITAR:

---

- Técnico de aviación del ejército Año: 2009
- Curso de Armamento Aéreo Año: 2009 Horas: 200
- Instalaciones eléctricas ETAE Año: 2009 Horas: 80
- Seguridad en redes ESPE MED Año: 2009 Horas: 120
- Sistema de combustibles Cal-15 Año: 2010 Horas: 40
- Control de corrosión y NDI ISTAC Año: 2010 Horas: 60

### FIRMA:

---

**HECTOR QUINAHUANO**

**C.C.: 171793658 - 5**



## **HOJA DE LEGALIZACIÓN DE FIRMAS**

**DEL CONTENIDO DE LA PRESENTE INVESTIGACIÓN SE  
RESPONSABILIZA EL AUTOR**

---

**HÉCTOR FIDEL QUINAHUANO JITALA**

CBOS. DE AE.

ID: L00338008

**DIRECTOR DEL PROYECTO TÉCNICO DE GRADUACIÓN**

---

**ING. NEL AUGUSTO VACA FLORES, MSc.**

MAYO. TEC DE AVC.

**DIRECTOR DE LA CARRERA DE ELECTRÓNICA MENCIÓN  
INSTRUMENTACIÓN Y AVIÓNICA**

---

**ING. PABLO PILATASIG PANCHI**

Latacunga, 17 de marzo 2017