# COMPANY PROFILE

## Hindustan Aeronautics Limited (HAL)

Hindustan Aeronautics Limited (HAL) came into existence on 1st October 1964. The Company was formed by the merger of Hindustan Aircraft Limited with Aeronautics India Limited and Aircraft Manufacturing Depot, Kanpur.

The Company traces its roots to the pioneering efforts of an industrialist with extraordinary vision, the late Seth Walchand Hirachand, who set up Hindustan Aircraft Limited at Bangalore in association with the erstwhile princely State of Mysore in December 1940. The Government of India became a shareholder in March 1941 and took over the Management in 1942.

Today, HAL has 19 Production Units and 9 Research and Design Centers in 7 locations in India. The Company has an impressive product track record - 12 types of aircraft manufactured with in-house R & D and 14 types produced under license. HAL has manufactured over 3550 aircraft, 3600 engines and overhauled over 8150 aircraft and 27300 engines.

HAL has been successful in numerous R & D programs developed for both Defense and Civil Aviation sectors. HAL has made substantial progress in its current projects:

* Dhruv, which is Advanced Light Helicopter (ALH)
* Tejas - Light Combat Aircraft (LCA)
* Intermediate Jet Trainer (IJT)
* Various military and civil upgrades.
* Dhruv was delivered to the Indian Army, Navy, Air Force and the Coast Guard in March 2002, in the very first year of its production, a unique achievement

.

HAL has played a significant role for India's space programs by participating in the manufacture of structures for Satellite Launch Vehicles like

* PSLV (Polar Satellite Launch Vehicle)
* GSLV (Geo-synchronous Satellite Launch Vehicle)
* IRS (Indian Remote Satellite)
* INSAT (Indian National Satellite)

HAL has formed the following Joint Ventures (JVs):

* BAeHAL Software Limited
* Indo-Russian Aviation Limited (IRAL)
* Snecma HAL Aerospace Pvt Ltd
* SAMTEL HAL Display System Limited
* HALBIT Avionics Pvt Ltd
* HAL-Edgewood Technologies Pvt Ltd
* INFOTECH HAL Ltd

Apart from these seven, other major diversification projects are Industrial Marine Gas Turbine and Airport Services. Several Co-production and Joint Ventures with international participation are under consideration.

HAL's supplies / services are mainly to Indian Defense Services, Coast Guards and Border Security Forces. Transport Aircraft and Helicopters have also been supplied to Airlines as well as State Governments of India. The Company has also achieved a foothold in export in more than 30 countries, having demonstrated its quality and price competitiveness.

HAL has won several International & National Awards for achievements in R&D, Technology, Managerial Performance, Exports, Energy Conservation, Quality and Fulfillment of Social Responsibilities.

* HAL was awarded the “INTERNATIONAL GOLD MEDAL AWARD” for Corporate Achievement in Quality and Efficiency at the International Summit (Global Rating Leaders 2003), London, UK by M/s Global Rating, UK in conjunction with the International Information and Marketing Centre (IIMC).
* HAL was presented the International - “ARCH OF EUROPE” Award in Gold Category in recognition for its commitment to Quality, Leadership, Technology and innovation.
* At the National level, HAL won the "GOLD TROPHY" for excellence in Public Sector Management, instituted by the Standing Conference of Public Enterprises (SCOPE).

# GENERAL DESCRIPTION OF HELICOPTER

## Configuration Description

The Advanced Light Helicopter is a twin engine, light weight, multirole

Helicopter of 5.5 ton class. It is powered by two TM 333-2B2 Turbo meca engines.

The Airframe consists of Cockpit, mid fuselage, rear fuselage and tail boom with empennage. The cockpit provides all round visibility, crashworthy seats for two pilots and crew entry/exit doors. The mid fuselage provides space for 14 troops and contains two sliding doors. Two main fuel tanks and two supply fuel tanks are located in the crashworthy bottom structure. The rear fuselage comprises of cargo compartment with two clamshell doors; rear fuel tank and avionics bay in the bottom structure; turbine deck, firewalls and electrical bay in the upper structure. The engine installation is covered by a cowling. The tail boom supports the tail rotor with controls, the vertical fin and horizontal stabilizer with end plates. The tail boom is protected by a tail bumper.

The helicopter is fitted with a skid landing gear consisting of a forward cross tube, a rear cross tube and two skid tubes.

Power from both the engines is fed to the Main Gear Box and transmitted to the main rotor. MGB is mounted above the cabin on ARIS to isolate rotor vibration being transmitted to fuselage. Power from Main Gear Box is transmitted to the tail rotor via flexibly-coupled shafts and auxiliary gear box. The main rotor is a four-bladed, hinge less type with flexible fiberglass blades and the tail rotor is a four-bladed, stiff-in-plane, bearing less type using flex beam concept.

The fuel is stored in three main tanks and two supply tanks. The supply tanks provide fuel independently to the engines. Single point fuel filling is provided. Fuel shut off valves operated from cockpit are installed to cut off fuel supply to the respective engines in case of emergency.

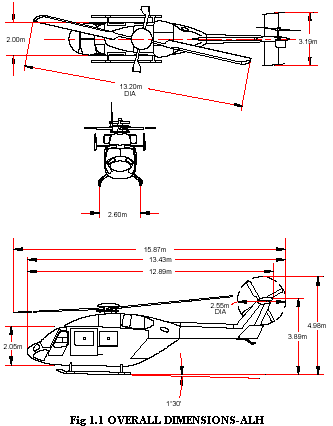
The Flight Control System is a dual, conventional, hydraulically powered, mechanical system. The upper control system is positioned inside the MGB housing and the main rotor drive shaft. The combination of MGB, Upper controls and Rotor head as a single unit is called Integrated Dynamic System (IDS). Duplex, tandem hydraulic actuators are used to control main and tail rotors in collective, and yaw channels respectively.

The hydraulic system comprises of a variable delivery axial piston pump, a hydraulic package to control and monitor the hydraulic power and the servo actuators. ALH is fitted with a four axes AFCS which provides, stability and control augmentation, auto trim functions and auto pilot modes. Forced air ventilation and heating system is also provided.

The electrical system consists of both AC and DC generating systems. The DC system has two independent sub-systems each consisting of a generator, a battery and associated control and protection systems. When both DC generators fail, emergency DC loads are supplied from batteries. The AC system has two independent sub systems each consisting of an alternator together with associated control and protection system. When both alternators fail, emergency AC loads are supplied from a static inverter.

AUTOMATIC FLIGHT CONTROL SYSTEM (AFCS) provides handling qualities, trim facilities and autopilot modes necessary for carrying out various missions. The computer units are dual, micro-processor based digital systems.

The helicopter is equipped with a standard instrument package, an intercom system, Radio communication system V/UHF (AM), Automatic Direction Finder (ADF), Doppler GPS, True Air Speed system (TAS), AHRS, HF (SSB), Altimeter, Weather radar, IFF and FDR/CVR.

****

## Leading Particulars

**General**

Description … ALH Army version

Type … Twin Engine

Seating Capacity … 2 Pilots + 14 Troops

Overall Dimensions … See Figure 1.1

**Engine**

Type … TM 333-2B2 (TURBOMECA)

Rated speed (NF) … 37562 RPM

Output shaft speed … 6000 RPM

Take off power … 801 KW at SL, ISA.

Fuel used … ATF K50

Oil used … MIL-L-23699

**Transmission**

a) Main Gear Box

Input speed … 6000 RPM

(Engine-MGB connection)

Main rotor drive speed … 314 RPM

Tail drive shaft speed … 4163 RPM

Oil … MIL-L-239663C

b) Tail Gear Box

Input speed … 4033 RPM

Tail rotor drive speed … 1564 RPM

Oil … MIL-L-23699

**Oil Cooling System**

Engine … Integrated oil cooling system

MGB … Separate system with its own cooler and fan driven by Auxiliary Gear Box.

**Main Rotor**

Number of blades … 4

Rotor Diameter … 13.2 m

Direction of rotation … Clockwise

Rotor speed … 314 RPM

**Tail Rotor**

Number of blades … 4

Rotor Diameter … 2.55 m

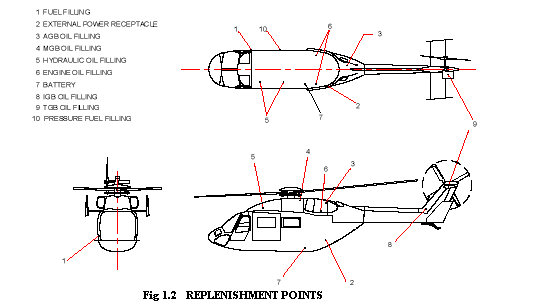
Direction of rotation … Bottom blade moving forward

Rotor speed … 1564 RPM

**Fuel System**

Total capacity of the five normal tanks - 1370 lt.

Capacity of ferry tank (optional) - 240 lt.



**Hydraulic System**

Hydraulic fluid … MIL-H-5606

Pump normal working … 206 ± 4 Bar Pressure

Max. flow rate … 25 Lt/mn

Pressure circuit filter … 15 microns absolute

Return filter … 5 microns absolute

**Electrical System**

a) DC Voltage

Rated voltage … 28 ± 0.5 V

Starter generator (2) … 6 KW (200 A) each

Nickel-cadmium battery (2) … 27 AH each

b) AC Voltage

Three phase 115V-26V/400Hz

Alternators … 2x5 KVA each

Static inverter … 1x350 VA

## Interior Arrangement

The cockpit can accommodate two pilots. It has dual controls and necessary flight instruments. The cabin has a volume of 7.33 cubic meters and a seating capacity for 14 troops. The cargo compartment has a volume of 2.16 cubic meters with clam shell doors at its rear. A schematic diagram giving the various exterior replenishment points is given in Figure 1.2.

## Centralized Warning System (CWS)

This system provides centralized light warnings of status and malfunctions.

System details

The system comprises

* 8 CBs (on overhead panel)
* A Centralized Warning Panel (CWP) with 66 light indicators
* Two master warning lights - one for each pilot (on instrument panel)
* A “FUEL LOW’ test switch (on overhead panel)
* An ‘AUDIO WARN MUTE’ push button on the collective grip
* Built - In - Test facility (BIT)

**System Description**

The system has three levels of warning/caution indication with dimming facility:

* First level: Red Warning light + Flashing Master Warnings + Distinct Audio tone on headset. This warning is for the most critical functions
* Second level: Red Warning light + Flashing Master Warnings. This indicates the next level of criticality.
* Third level: Yellow Caution light + Flashing Master Warnings. This indicates non critical cautionary conditions.

Audio Warnings (Associated with CWS)

* Three Audio tone generators (only 2 are used) built-in to Centralized Warning Panel
* Audio tones are distinct from one another for fast recognition
* Audio tones are sent to the headsets through the Intercom System

Audio tone 1 for Rotor RPM high: (Audio tone with an ON/OFF rate of 50 ms.)

* Frequency : 2400 Hz
* Condition : NR ≥106.5%
* Tone : Can be muted by pressing “AUDIO

WARN MUTE” push button.

Audio tone 2 for Rotor RPM Low: (Audio tone with an ON/OFF rate of 100 ms.)

* Frequency : 800 Hz
* Condition : NR ≤95%
* Tone : Can be muted by pressing

“AUDIO WARN MUTE” push button.

Audio tone 3: - Modulated audio tone

* Frequency sweeps from 700 Hz to 1700 Hz in 0.85 s with an interruption interval of 0.12 sec in between two sweeps
* Tone : Resettable by depressing master warnings

THIS AUDIO TONE IS NOT USED

Master Warnings: any faults in the parameters are highlighted.

* Color : Red
* Legend : MASTER
* Indication : Flashing

# LINE REPLACEABLE UNITS (LRU’S):

A line-replaceable unit is a complex component of an aircraft that is designed to be replaced quickly at the flight line or airport ramp area. An LRU is a blackbox (sealed units), such as a radio or other auxiliary equipment. LRUs speed up repair, because they can be stocked and replaced quickly from inventory, restoring the larger system to services. They also reduce the cost of system, and increase the quality, by spreading development costs of systems, and increase the quality, by spreading development costs. Although an LRU is called a “black box” because its internal design is immaterial to the end user, the box itself is often painted gray in color.

LRUs are designed to common specification, with cannol plug input/output interfaces. Specifications also define the tools necessary to remove and replace the unit and the bulk and wait. There are also requirements for testing the LRUs flammability, unwanted radio emissions, resistance to damage from fungus, static electricity, heat, pressure, humidity, vibration, radiation and other environmental measurements.

Many LRUs for commercial aircrafts are designed according to ARINC specifications, such as ARINC 404 and ARINC 600. LRUs are also defined by manufactures like airbus and Boeing and by various military organizations. In the military, electronics LRUs are typically designed to interface according to data bus standards such as MIL-STD-1553. There are about 340 LRUs used in ALH (DHRUV). Among them 135 LRUs are serviced and tested in HAL.

# HELICOPTER FUEL SYSTEM

The fuel system provides a continuous supply of fuel to both the engines of ALH under all conditions.

## General Description

ALH helicopter fuel system is a suction feed system with a cross fed value connecting both LH and RH engine feed lines. The aircraft fuel system has three main tanks and two Supply tanks. There are

• Forward main tank (FMT)

• Mid-Main tank (MMT)

• Rear main tank (RMT)

• Supply tank-1(ST-1)

• Supply tank-2(ST-2)

• Axillary tank (AT)

**Filling and Transfer Sequence**

Front and middle tanks get filled up simultaneously supply tanks and rear main tank are filled by transfer pumps from forward and middle tanks During operation after approximately 160kg of fuel has been consumed from FMT and MMT combined, fuel from RMT needs to be transferred to FMT by operation of RMT a transfer pump by the pilot .The location of switches for transfer pumps i.e; FMT, MMT and RMT to MAIN /MAIN to RMT.

**Auxiliary Fuel Tank**

Auxiliary fuel tank is an optional external fuel tank for long range .ferry mission .It is located on the cabin floor board as shown in figure .It has a capacity of 250lts .This fuel is transferred to the main fuel system by gravity by the operation of shut off valve in the transfer line .A Fuel gauge is provided in auxiliary fuel tank to record the fuel quantity inside the tank .A separate vent line from the auxiliary fuel tank is connected to the vent system .A Drain valve is also provided on the auxiliary tank bottom plate .

**External tanks**

External tanks are used to extend the range of an aircraft .Drop tanks are used by combat aircraft that need to discard them after use for performance reasons .To transfer fuel from the tip tank to main tank on each side ,there must be a fuel pump in the tip tank.

**Fuel quantity /Flow rate indication**

The fuel quantity/flow rate indicator consists of three analog and three digital displays indicating fuel quantity in kgs of the various fuel tanks. Two digital displays indicating fuel rate in Kg/hr of two engines .Two out of the three displays(analog and digital ) will indicate the individual supply tank quantity and the third (analog and digital ) display in conjunction with a rotary tank selection switch gives indication of the selected fuel tank viz, total, main ,rear and auxiliary 1/auxiliary 2.

**Fuel dumping system**

Fuel dump valve is provided in the bottom of forward main tank and connected to an outlet so that fuel can be dumped clearing the helicopter. Fuel from all main tanks (except supply tanks) could be dumped by gravity by opening the dump valve using a switch in the cockpit. Dump valve is provided with twin motors for redundancy and manual operation of dump valve is not provisioned .Dumping action can be stopped by putting the switch to off position. The same system could be used for defueling the tanks on the ground.

# FUEL PROBES

It is an instrument to indicate the level of fuel contained in the tank. It consists of

* Sensing unit.
* Indictor



The fuel quantity probes from LMS are lightweight, durable, and cost effective. They have been developed to withstand any flight conditions with outstanding reliability.

The sensing unit typically uses a float connected to a potentiometer typically printed ink design in a modern automobile. As the tank empty, the float drops and slides a moving contact along the resistor, increasing its resistance. In addition, when the resistance is at a certain point, it will also turn on a “low fuel” light on some vehicles.

Meanwhile, the indicator unit (usually mounted on the dashboard) is measuring and displaying the amount of electric current flowing through the sending unit. When the tank level is high and maximum current is flowing, the needle points to "F" indicating a full tank. When the tank is empty and the least current is flowing, the needle points to "E" indicating an empty tank.

The system can be fail-safe. If an electrical fault opens, the electrical circuit causes the indicator to show the tank as being empty (theoretically provoking the driver to refill the tank) rather than full (which would allow the driver to run out of fuel with no prior notification). Corrosion or wear of the potentiometer will provide erroneous readings of fuel level. However, this system has a potential risk associated with it. An electric current is sent through the variable resistor to which a float is connected, so that the value of resistance depends on the fuel level. In most automotive fuel gauges such resistors are on the inward side of the gauge, i.e., inside the fuel tank. Sending current through such a resistor has a fire hazard and an explosion risk associated with it. These resistance sensors are also showing an increased failure rate with the incremental additions of alcohol in automotive gasoline fuel. Alcohol increases the corrosion rate at the potentiometer, as it is capable of carrying current like water. Potentiometer applications for alcohol fuel use a pulse-and-hold methodology, with a periodic signal being sent to determine fuel level decreasing the corrosion potential. Therefore, demand for another safer, non-contact method for fuel level is desired.

Magnetoresistance type fuel level sensors, now becoming common in small aircraft applications, offer a potential alternative for automotive use. These fuel level sensors work similar to the potentiometer example, however a sealed detector at the float pivot determines the angular position of a magnet pair at the pivot end of the float arm. These are highly accurate, and the electronics are completely outside the fuel. The non-contact nature of these sensors address the fire and explosion hazard, and also the issues related to any fuel combinations or additives to gasoline or to any alcohol fuel mixtures. Magneto resistive sensors are suitable for all fuel or fluid combinations, including LPG and LNG. The fuel level output for these senders can be ratiometric voltage or preferable CAN bus digital. These sensors also fail-safe in that they either provide a level output or nothing.

Systems that measure large fuel tanks (including underground storage tanks) may use the same electro-mechanical principle or may make use of a pressure sensor, sometimes connected to a mercury manometer.

Many large transport aircraft use a different fuel gauge design principle. An aircraft may use a number (around 30 on an A320) of low voltage tubular capacitor probes where the fuel becomes the dielectric. At different fuel levels, different values of capacitance are measured and therefore the level of fuel can be determined. In early designs, the profiles and values of individual probes were chosen to compensate for fuel tank shape and aircraft pitch and roll attitudes. In more modern aircraft, the probes tend to be linear (capacitance proportional to fuel height) and the fuel computer works out how much fuel there is (slightly different on different manufacturers). This has the advantage that a faulty probe may be identified and eliminated from the fuel calculations. In total this system can be more than 99% accurate. Since most commercial aircraft only take on board fuel necessary for the intended flight (with appropriate safety margins), the system allows the fuel load to be preselected, causing the fuel delivery to be shut off when the intended load has been taken on board.

An aircraft fuel system allows the crew to pump, manage, and deliver aviation- or jet fuel to the propulsion system and Auxiliary Power Unit (APU) of an aircraft. Fuel systems differ greatly due to different performance of the aircraft in which they are installed. A single-engine piston aircraft has a simple fuel system; a tanker (such as the KC-135), in addition to managing its own fuel, can also provide fuel to other aircraft.[1]

Fuel is piped through fuel lines to a fuel control valve (usually known as the fuel selector). This valve serves several functions. The first function is to act as a fuel shut-off valve. This is required to provide the crew with a means to prevent fuel reaching the engine in case of an engine fire. The second function is to allow the pilot to choose which tank feeds the engine. Many aircraft have the left tank and right tank selections available to the pilot. Some Cessna airplanes feed only from both tanks; and many have the option to feed from left, right, or both tanks. The reason to have left only and right only options is to allow pilots to balance fuel load to reduce the banking moment. In some aircraft, the shut-off function is a different valve located after the fuel selector valve.

After the selector valve there usually is a gascolator - a fuel filter that can be drained. Other drainage points are in each tank (often more than one contaminant collection sump per tank) and at the injection pump.

Each tank is vented (or pressurised) to allow air into the tank to take the place of burned fuel; otherwise, the tank would be in negative pressure which would result in engine fuel starvation. A vent also allows for changes in atmospheric pressure and temperature.

# PROBLEM STATEMENT AND FORMULATION

## PROBLEM STATEMENT

Presently an analog setup is used in Hindustan Aeronautics Limited. It consists of fuel reserve, FCG electronic test kit, a fixed power supply unit and CRO.

This setup involves some problems as follows:

* Parallax error.
* Particles exceeding ½ inch diameter can be problematic.
* Changes in temperature of liquid results in shift of di electric constant.
* Time consuming.
* Requires more man power.
* Equipment used requires takes more space for installation.

## PROBLEM FORMULATION

The changes in the temperature of liquid results in shift of the dielectric constant which results in lots of problems as it lead to change of the readings for every fraction change of temperature. Due to so many short comings analog setup is time consuming. Thus to overcome all these shortcoming we switch to completely automated digitalized setup.

* Two microcontrollers (Arduino Uno) are used. One is to read the analog frequency and the same is converted to digital frequency, height of using suitable codes. Second controller is used to implement PID control of height of fuel in the tank using three phase AC motor.
* Time period, height are displayed on a 16 x 2 LCD. Final metrology of the probe is printed on a graphical user interface (GUI) using serial communication ports.
* A buzzer and low level warning on display have been introduced along with LED.
* Two relays are also used to control the motor.

## Discussion of the problem

Presently an analog setup is being used in Hindustan Aeronautics Limited whose setup consists of FCG tank, fuel reserve, FCG electronic test kit, a fixed power supply unit (28v and 5v) and CRO. The probe from the tank is connected to the test kit.

The supply is given to the FCG electronic test rig. The FCG test tank can test all the five main tanks and the connection is given to the FCG electronic test rig. Upon the selection of tank the content of ATF K-50 i:e; the fuel reserve height is read from the capillary tube attached to the FCG test tank. For the purpose of calculation of the frequency, time-period waveforms are read from the CRO. In this setup operator has to turn on the motor manually and stop it when desired height is reached by checking the capillary tube. Then he should note down the frequency, time-period manually from the CRO. This should be repeated nearly 35 times for metrology of single probe and requires three persons to do the job. One person should operate motor, second person should check the height in capillary tube and one more person should check the CRO for noting down time period and frequency.

In reading the waveform lot of parallax error can be encountered. Particles which are more than half inch diameter can be problematic as they interfere with the di electric constant and density of the liquid. The changes in the temperature of the liquid results in shift of the dielectric constant which results in lots of problems as it lead to change of the readings for every fraction change of temperature. This also requires more man power. Due to so many shortcomings this analog setup is time consuming. Thus to overcome all these we switch to a completely automated digitalized setup.

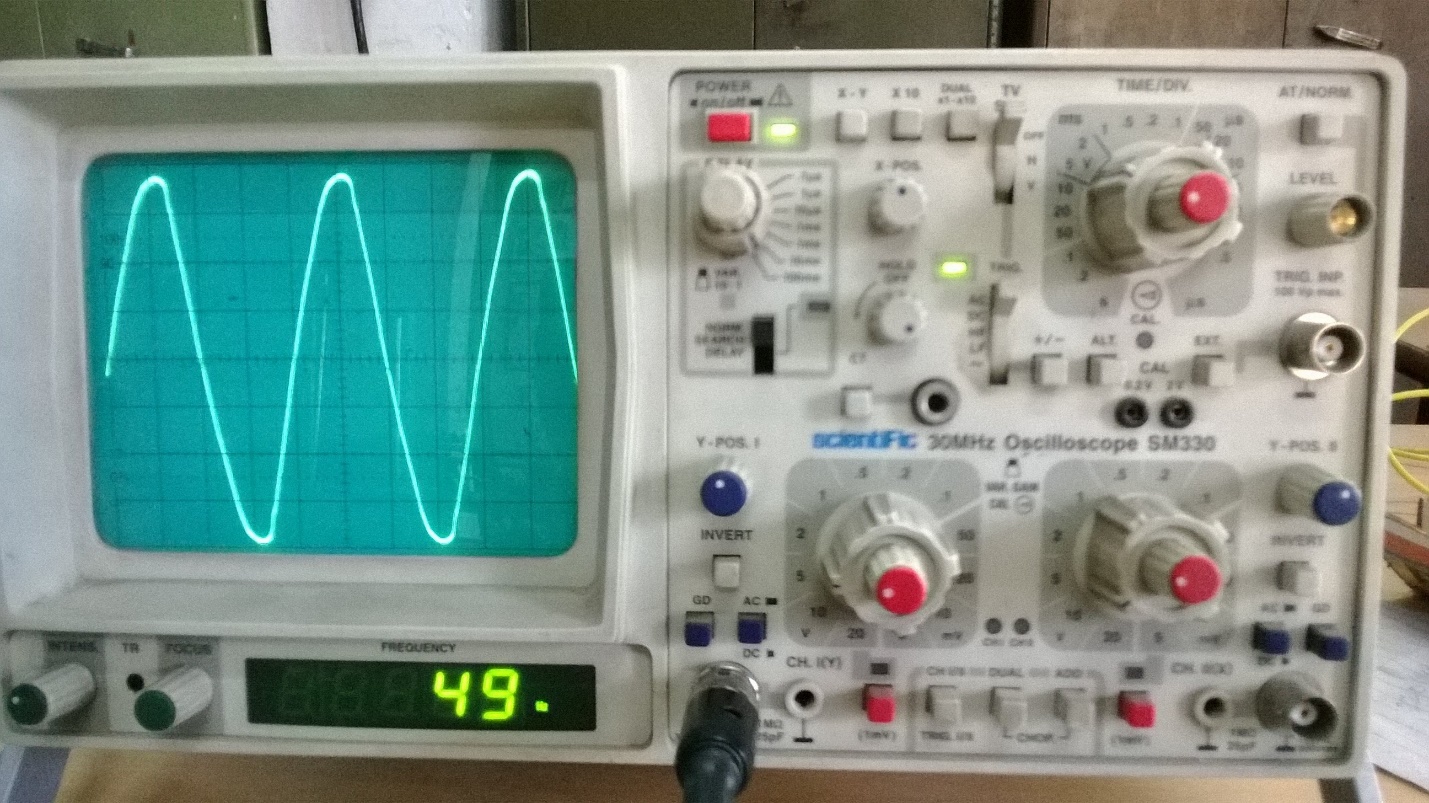
Most aircraft fuel tanks have numerous fuel quantity probes. Number of probes used can vary depending on tank size and system design. Most common configuration have 6-12 probes. This probes are capacitive transducers. The higher the fuel, the higher the capacitance. Fuel capacitance is read in Pico Farads. Tanks have one probe that measures fuel density. This is a compensator probe. The “comp” probe has a unique function, but its outputs are still read in pf.

# The Fuel Probe Test Box and its Electrical Schematic



## Cathode-Ray Oscilloscope (CRO)

Oscilloscopes are used in the sciences, medicine, engineering, automotive and the telecommunications industry. General-purpose instruments are used for maintenance of electronic equipment and laboratory work. It is commonly used to display and analyze the waveform of electronic signals. The basic oscilloscope, as shown in the illustration, is typically divided into four sections: the display, vertical controls, horizontal controls and trigger controls. The display is usually a CRT or LCD panel which is laid out with both horizontal and vertical reference lines referred to as the graticule. In addition to the screen, most display sections are equipped with three basic controls: a focus knob, an intensity knob and a beam finder button. The vertical section controls the amplitude of the displayed signal. This section carries a Volts-per-Division (Volts/Div.) selector knob, an AC/DC/Ground selector switch and the vertical (primary) input for the instrument. Additionally, this section is typically equipped with the vertical beam position knob. The horizontal section controls the time base or "sweep" of the instrument. The primary control is the Seconds-per-Division (Sec/Div) selector switch. Also included is a horizontal input for plotting dual X-Y axis signals. The horizontal beam position knob is generally located in this section.



## Regulated DC power supply.

A regulated power supply is an embedded circuit; it converts unregulated AC into a constant DC. With the help of a rectifier it converts AC supply into DC. Its function is to supply a stable voltage (or less often current), to a circuit or device that must be operated within certain power supply limits. The output from the regulated power supply may be alternating or unidirectional, but is nearly always DC. Digital power has come out of the realm of R&D lab and into the main stream. Note that the term “digital power” is often loosely defined with respect to power supply which is digital, otherwise known as switching power supply. This type of supply may utilize either digital or analog methods to control the on/off time of the power MOSFET. Also, analog power supply controllers may make use of digital circuitry for power supply sequencing and tracking, voltage margining, as well as I2C implementations of reading back or fault conditions , such as input/output under voltage and overvoltage, or output short circuit.

Now-a-days designing power supply plays a vital role in any electronic equipment. Power supply must be as compact as possible. It should be free from harmonics and should be reliable in operation. Size of power supply decides the overall size of the electronic device. So one must give vital importance for designing and implementing power supply for any electronic equipment.



# LITERATURE SURVEY

## Data sheets of Various Tank Probes by HAL Lucknow.

These datasheets indicate the capacity of each tanks and also the calibration table which consists of frequency, height of the probe and height of the fuel. The document also indicated the power rating, tolerance values of different probes. The document are of standard revision 2013.

## Arduino user Manual-Hands on Research in complex system Shanghai Jiao Tong University 17-29, 2012

This manual helps a beginner to get to know about the Arduino boards and start-up with simple programs. This manual also gives the internal structure of the Atmel microcontroller and its specifications in detail.

## FDU user manual Mfd. By Flight Test Centre, Aircraft Research and design center, Hindustan Aeronautics Limited, Bengaluru.

This manual provides the functional description, operation and maintenance procedures for the fuel display unit (FDU) installed in the instrument panel of Advanced Light Helicopter (ALH) cockpit as part of fuel gauging system.

## Proportional Integral Derivative (PID) Control

PID control stands for proportional plus derivative plus integral control. PID control is a feedback mechanism which is used in control system. This type of control is also termed as three term control. By controlling the three parameters - proportional, integral and derivative we can achieve different control actions for specific work. PID is considered to be the best controller in the control system family. For PID control the actuating signal consists of proportional error signal added with derivative and integral of the error signal. Therefore, the actuating signal for PID control is –



There are some control actions which can be achieved by using any of the two parameters of the PID controller. Two parameters can work while keeping the third one to zero. So PID controller becomes sometimes PI (Proportion integral), PD (proportional derivative) or even P or I. The derivative term D is responsible for noise measurement while integral term is meant for reaching the targeted value of the system. In early days PID controller was used as mechanical device. These were pneumatic controllers as they were compressed by air. Mechanical controllers include spring, lever or mass. Many complex electronic systems are provided with PID control loop. In modern days PID controllers are used in PLC (programmable logic controllers) in the industry. The proportional, derivative and integral parameters can be expressed as - Kp, Kd and Ki. All these three parameters have effect on the closed loop control system. It affects rise time, settling time and overshoot and also the steady state error.

PID control combines advantages of proportional, derivative and integral control actions. Let us discuss these control actions in brief.

Proportional or P- controller gives output which is proportional to current error e (t). It compares desired or set point with actual value or feedback process value. The resulting error is multiplied with proportional constant to get the output. If the error value is zero, then this controller output is zero. This controller requires biasing or manual reset when used alone. This is because it never reaches the steady state condition. It provides stable operation but always maintains the steady state error. Speed of the response is increased when the proportional constant Kc increases.

I-Controller: Due to limitation of p-controller where there always exists an offset between the process variable and set point, I-controller is needed, which provides necessary action to eliminate the steady state error. It integrates the error over a period of time until error value reaches to zero. It holds the value to final control device at which error becomes zero.

Integral control decreases its output when negative error takes place. It limits the speed of response and affects stability of the system. Speed of the response is increased by decreasing integral gain Ki. As the gain of the I-controller decreases, steady state error also goes on decreasing. For most of the cases, PI controller is used particularly where high speed response is not required. While using the PI controller, I-controller output is limited to somewhat range to overcome the integral wind up conditions where integral output goes on increasing even at zero error state, due to nonlinearities in the plant.

D-Controller: I-controller doesn’t have the capability to predict the future behavior of error. So it reacts normally once the set point is changed. D-controller overcomes this problem by anticipating future behavior of the error. Its output depends on rate of change of error with respect to time, multiplied by derivative constant. It gives the kick start for the output thereby increasing system response. Response of D controller is more, compared to PI controller and also settling time of output is decreased. It improves the stability of system by compensating phase lag caused by I-controller. Increasing the derivative gain increases speed of response.

So finally we observed that by combining these three controllers, we can get the desired response for the system. Different manufactures designs different PID algorithms.

## Tuning methods of PID Controller

Before the working of PID controller takes place, it must be tuned to suit with dynamics of the process to be controlled. Designers give the default values for P, I and D terms and these values couldn’t give the desired performance and sometimes leads to instability and slow control performances. Different types of tuning methods are developed to tune the PID controllers and require much attention from the operator to select best values of proportional, integral and derivative gains. Some of these are given below.

Trial and Error Method: It is a simple method of PID controller tuning. While system or controller is working, we can tune the controller. In this method, first we have to set Ki and Kd values to zero and increase proportional term (Kp) until system reaches to oscillating behavior. Once it is oscillating, adjust Ki (Integral term) so that oscillations stops and finally adjust D to get fast response.

Process reaction curve technique: It is an open loop tuning technique. It produces response when a step input is applied to the system. Initially, we have to apply some control output to the system manually and have to record response curve. After that we need to calculate slope, dead time, rise time of the curve and finally substitute these values in P, I and D equations to get the gain values

Zeigler-Nichols method: Zeigler-Nichols proposed closed loop methods for tuning the PID controller. Those are continuous cycling method and damped oscillation method. Procedures for both methods are same but oscillation behavior is different. In this, first we have to set the p-controller constant, Kp to a particular value while Ki and Kd values are zero. Proportional gain is increased till system oscillates at constant amplitude.

Gain at which system produces constant oscillations is called ultimate gain (Ku) and period of oscillations is called ultimate period (Pc). Once it is reached, we can enter the values of P, I and D in PID controller by Zeigler-Nichols table depends on the controller used like P, PI or PID, as shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| Controller | Kc | Ti | Td |
| P | Ku/2 |  |  |
| PI | Ku/2.2 | Pu/1.2 |  |
| PID | Ku/2 | Pu/2 | Pu/8 |

# IMPLEMENTATION METHODOLOGY

## Hardware Implementation

* Hardware Implementation describes the process by which the Analog Probe Test Kit is replaced by Automatic Fuel Probe Test Kit.
* Analog Probe Test Kit, which has a power supply for both 28volts and 5volts, CRO for measuring the time period and a test kit for giving 28volts and 5volts to the fuel probe and taking Frequency signal from fuel probe.
* The Automatic Fuel Probe Test Kit is powered by 220V AC. Test kit internal circuit generates 5Vand 28V for powering microcontrollers, LCD, relays and for probe excitation. It also has an LCD display for displaying the height of fuel and corresponding time period. It is connected to motor via two 230V/5V relays. This relay is controlled by the control signals from microcontroller thus implementing PID control of height of fuel in the tank by suitable program.
* Microcontroller is connected to a computer through a ‘COM’ port. An application is made for sending calibration chart in to the controller and saving height – timeperiod metrology of the probe that is being tested.
* The microcontroller used is Arduino Uno (ATMEGA 328P)

## Automated Fuel Probe Test Kit

The kit consists of

* Fuel tank with Fuel probe
* Three phase Motor
* Controller kit with LCD display
* Computer

## List of Components

* Two Arduino Uno Board
* One 28V/5V Relay
* Two Channel 230V/5V Relay Module
* LCD Display (16x2)
* Rotary Switch
* Buzzer
* Five DPDT Switches
* D-type Male connector
* Different resistors

## POWER SUPPLY

Automatic Fuel Probe Test Kit has a Power cable socket where you plug the power cable from 220v AC. The Test Kit has two AC to DC converters. One is 230V/5V convertor and the other one is 230V/28V convertor. 5V supply is used to power Arduinos (microcontrollers), LCD, relays and for probe excitation.28V is for excitation of Low-Level Sensor in fuel probe.

## LCD DISPLAY

LCD, an acronym for Liquid Crystal Display revolutionized the modern display technology with its compactness and versatility. Today it is seen embedded in various electronic gadgets and devices like T.V., Computers, Laptops, Watches, etc. A Liquid crystal coating is the heart of the display which is sandwiched between two polarized glasses.

LCD’s are available in various shapes and sizes depending on the configurations. A [16x2 LCD](http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet) shown in the image below can display 32 characters with 16 characters in each row. It is capable of displaying any character with ASCII values ranging from 0 to 255.

****

****

The Liquid Crystal Library allows you to control LCD Displays that are compatible with the **JHD204A** Driver.There are many of them out there , and you can usually tell them by the 16pin interface. The LCDs have a parallel interface, meaning that the microcontroller has to manipulate several interface pins at once to control the display. The interface consists of the following pins.

****

Character Lcd's have a controller built into them named HD44780. We actually talk with this controller in order to display a character on the LCD screen. HD44780 must be properly handled and initialized before sending any data to it. HD44780 has some registers which are initialized and manipulated for character displaying on the LCD. These registers are selected by the pins of c Rs(Register select)

Register select selects the HD44780 controller registers. It switches between Command and data register.

• Command Register

• Data Register

**Command Register**:

When we send commands to LCD these commands go to Command register and are process them. Commands with their full description are given in the picture below. When Rs=0 command register is selected.

Data Register:

When we send Data to LCD it goes to the data register and is processed there. When Rs=1 data register is selected.

**R/W(Read - Write):**

R/W pin is used to read and write data to HD44780 data and command registers. When R/W=1 we can read data from LCD. When R/W=0 we can write to LCD.

**En(Enable signal):**

When we select the register Rs(Command and Data) and set Rw(read - write) and placed the raw value on 8-data lines, now it’s time to execute the instruction. By instruction I mean the 8-bit data or 8-bit command present on Data lines of LCD. For sending the final data/command present on the data lines we use this enable pin. Usually, it remains en=0 and when we want to execute the instruction we make it high en=1 for some mills seconds. After this we again make it ground en=0.

**VE (Set Lcd contrast):**

To set LCD display sharpness use this pin. The best way is to use variable resistor such as potentiometer a variable current makes the character contrast sharp. Connect the output of the potentiometer to this pin. Rotate the potentiometer knob forward and backward to adjust the LCD contrast.

There are power supply pins(+5v and Gnd) and LCD Backlight(BKLt+ and BKLt-)that you can use to power the LCD, and turn on and off the LCD backlight.

The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register.The Liquid Crystal Library simplifies this for you so you don’t need to know the low-level instructions.The JHD204A LCDs can be controlled in two modes:4bit or 8bit.The 4bit mode requires seven I/O pins from the Arduino, while the 8bit mode requires 11 pins.For displaying text on the screen, you can do everything in 4bit mode.

**FEATURES**

• 5\*8 dots with cursor

• Built-in controller(HD44780)

• +5V power supply (Also available for +3V)

• 1/16 duty cycle

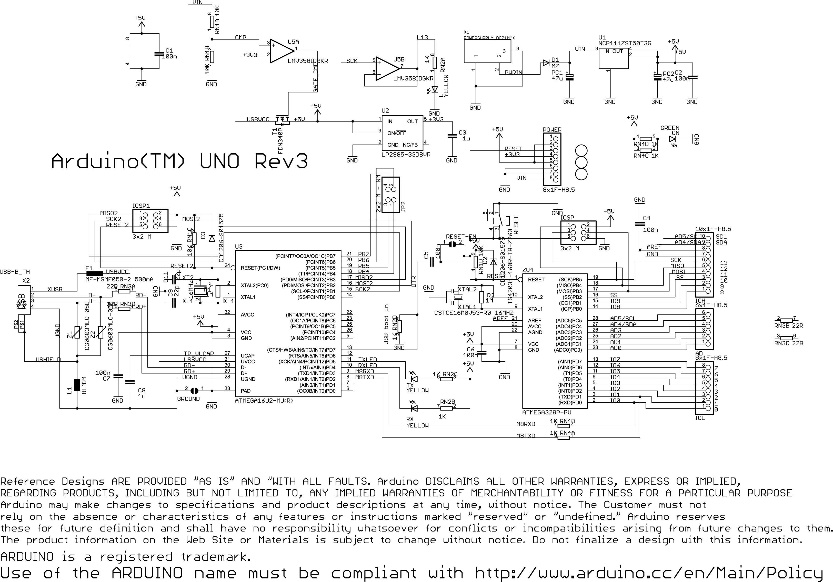
**LCD PINS AND ARDUINO CONNECTIONS**

|  |  |
| --- | --- |
| LCD PIN | ARDUINO PIN |
| 1. VSS | GND |
| 1. VDD | 5v |
| 1. Contrast | 3.3V |
| 1. RS | 3 |
| 1. R/W | GND |
| 1. Enable | 4 |
| 1. D7 | 9 |
| 1. D6 | 10 |
| 1. D5 | **11** |
| 1. D4 | 12 |
| 1. D3 | No connection |
| 1. D2 | No connection |
| 1. D1 | No connection |
| 1. D0 | No connection |
| 1. LCD +ve | 3.8V |
| 1. LCD-ve | **GND** |

**Interfacing Pin Configuration**

**Arduino Microcontroller**

Arduino is a single-board microcontroller to make using electronics in multidisciplinary projects more accessible. The hardware consists of a simple open source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM. The software consists of a standard programming language compiler and a boot loader that executes on the microcontroller.





**Hardware Specifications**

• Microcontroller: ATmega328

• Operating Voltage: 5V

• Input Voltage (recommended):7-12V

• Input Voltage (limits): 620V

• Digital I/O Pins: 14 (of which 6 provide PWM output)

• Analog Input Pins: 6

• DC Current per I/O Pin: 40 mA

• DC Current for 3.3V Pin: 50 mA

• Flash Memory: 32 KB (ATmega328)

• SRAM: 2 KB (ATmega328)

• EEPROM: 1 KB (ATmega328)

• Clock Speed: 16 MHz

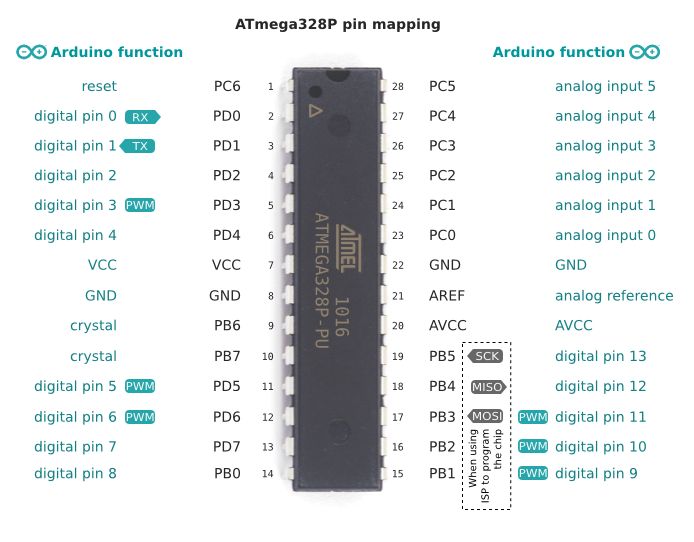
An Open source design:-The advantage of being an open source is that it has a large community of people using and troubleshooting it.This makes it easy to find someone to help you to debug your project .

An ICSP connector for bypassing the USB port and interfacing the arduino directly as a serial device .This port is necessary to re-boot load your chip if it corrupts &can no longer talk to your computer.

An on-board LED attached to digital pin 13 for the fast and easy debugging of code &last ,but not the least , a button to reset the program on the chip .

An Easy USB interface:-The chip on the board plugs straight into your USB port &registers on your computer as a virtual serial port .This allows you to interface with it were a serial device .The benefit of this setup is that serial communication is an extremely easily (&time tested ) protocol &USB makes connecting it to the modern computers really convinent.

Very convenient power management & built in voltage regulation, you can connect an external power source up to 12v and it will be regulate it to both 5v and 3.3 v.It also can be powered directly of USB port without external power.



**Digital Pins**

In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the pinMode(), digitalRead(), and digitalWrite() commands. Each pin has an internal pull-up resistor which can be turned on and off using digitalWrite() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40 mA.

* Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module. On the Arduino Mini and LilyPad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).
* External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
* PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.
* BT Reset: 7. (Arduino BT-only) Connected to the reset line of the bluetooth module.
* SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
* LED: 13. On the Diecimila and LilyPad, there is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

**Analog Pins**

In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the analogRead() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

* I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

**Power Pins**

* VIN (sometimes labelled "9V"). The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Note that different boards accept different input voltages ranges, please see the documentation for your board. Also note that the LilyPad has no VIN pin and accepts only a regulated input.
* 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply
* 3V3. (Diecimila-only) A 3.3 volt supply generated by the on-board FTDI chip.
* GND. Ground pins.

**Other Pins**

* AREF. Reference voltage for the analog inputs. Used with analogReference().
* Reset. (Diecimila-only) Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board

# SOFTWARE IMPLEMENTATION

## Procedure to execute the program

This document explains how to connect your Uno board to the computer and upload your first sketch. The Arduino Uno is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards and running both online and offline.

If you want to program your Arduino/Genuino Uno while offline you need to install the Arduino Desktop IDE. The Uno is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards. Before you can move on, you must have installed the Arduino Software (IDE) on your PC.

Connect your Uno board with an A B USB cable; sometimes this cable is called a USB printer cable.

The USB connection with the PC is necessary to program the board and not just to power it up. The Uno automatically draw power from either the USB or an external power supply. Connect the board to your computer using the USB cable. The green power LED (labelled PWR) should go on.

**Install the board drivers**

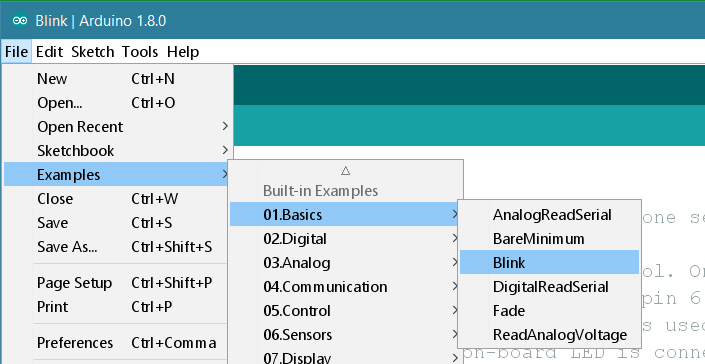
If you used the Installer, Windows - from XP up to 10 - will install drivers automatically as soon as you connect your board.

If you downloaded and expanded the Zip package or, for some reason, the board wasn't properly recognized, please follow the procedure below.

* Click on the Start Menu, and open up the Control Panel.
* While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager
* Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)". If there is no COM & LPT section, look under "Other Devices" for "Unknown Device".
* Right click on the "Arduino UNO (COmxx)" port and choose the "Update Driver Software" option.
* Next, choose the "Browse my computer for Driver software" option.
* Finally, navigate to and select the driver file named "arduino.inf", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory). If you are using an old version of the IDE (1.0.3 or older), choose the Uno driver file named "Arduino UNO.inf"
* Windows will finish up the driver installation from there.

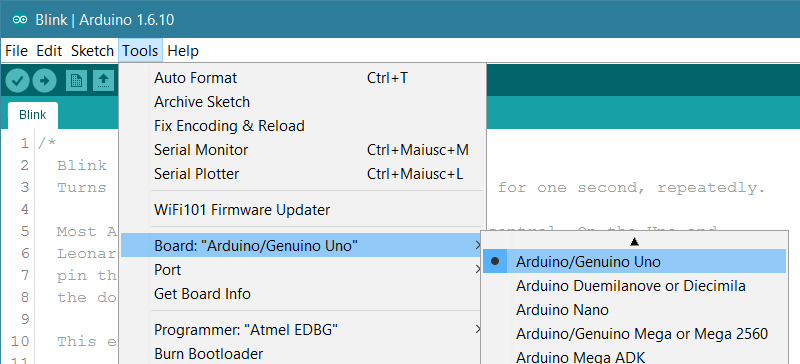
**Open your first sketch**

Open the LED blink example sketch: File > Examples >01.Basics > Blink.



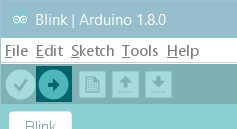
**Select your board type and port**

You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino or Genuino board.



**Upload the program**

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.



# PROGRAMME

## Code for reading probe frequency, calculating height of fuel:

#include <LiquidCrystal.h>

#include <FreqCounter.h>

// initialize the library with the numbers of the interface pins(apt)

LiquidCrystal lcd(3,4,9,10,11,12);

//LiquidCrystal lcd(12,11,10,9,4,3);

int pin1=8;

int Buz=7;

int state1;

float x1,x2,y1,y2;

float a[17];

int b[17];

int p,i;

float c[] = {101.4,102.8,106.8,110.8,115.0,119.0,123.1,129.3,135.3,141.4,147.4,153.9,160.0,165.8,171.2,177.2,179.3};//st patno:235900000

int d[] = {10,20,40,60,80,100,120,150,180,210,240,270,302,330,360,390,412};//st patno:235900000

int e[] = {};//FMT

int f[] = {};//MMT

int g[] = {};//RMT

int h[] = {};//AT

int sensorPin = A2;

float value=0;

float vo=0;

void setup()

{

Serial.begin(9600);

lcd.begin(16,2);

pinMode (pin1, INPUT);

pinMode (Buz, OUTPUT);

lcd.setCursor(0,0);

lcd.print("Select Probe");

delay(3000);

//lcd.clear();

lcd.setCursor(3,0);

lcd.print("ok");

lcd.clear();

value=0;

value = analogRead(sensorPin);

vo=((value\*5.1)/1025);

if ( vo >4.8 && vo<5.2) //FT PROBE

{

for(i=0;i<sizeof(e);i++)

{

b[i] = e[i];

}

}

if(vo >3.5 && vo<4.2) //MT PROBE

{

for(i=0;i<sizeof(f);i++)

{

b[i] = f[i];

}

}

if ( vo >2.5 && vo<3.2) //ST PROBE

{

for(i=0;i<sizeof(d);i++)

{

b[i] = d[i];

}

}

if(vo >1.5 && vo<2.2) //AT PROBE

{

for(i=0;i<sizeof(h);i++)

{

b[i] = h[i];

}

}

if(vo >0 && vo<1.2) //RT PROBE

{

for(i=0;i<sizeof(g);i++)

{

b[i] = g[i];

}

}

lcd.setCursor(0,0);

lcd.print("Rated timeperiod");

//lcd.clear();

lcd.setCursor(0,0);

int l=0;

while(l<17){

if(Serial.available()>0){

float p = Serial.parseFloat();

if(p>0)

{

a[l]=p;

lcd.setCursor(0,1);

lcd.print(l);

l++;

}

}

}

}

void loop()

{

//float a[] = {100.7,102.6,106.5,110.2,114.2,118.1,122.0,127.7,133.2,139.3,144.8,150.8,158.3,162.1,167.2,173.2,177.6};

//int b[] = {10,20,40,60,80,100,120,150,180,210,240,270,302,330,360,390,427};

float ht;

float frq;

FreqCounter::f\_comp=10;

FreqCounter::start(1000);

while (FreqCounter::f\_ready == 0)

frq=FreqCounter::f\_freq; //Sets the 'frq' variable as the measured frequency

frq=frq\*1;

lcd.setCursor(0,1);

lcd.print("T: us ");

lcd.setCursor(2, 1);

//lcd.print(frq);

double t=(1/frq)\*1000000;

lcd.print(t);

//delay(2000); //Prints this measured frequency

//delay(100);

///value=0;

//value = analogRead(sensorPin);

//vo=((value\*5.1)/1025);

//Serial.println(vo);

state1= digitalRead(pin1);

int j=0;

for (int i=0; i<17; i++)

{ j=i+1;

if ( t>=a[i] && t<=a[j])

{ y1=a[i];

y2=a[j];

x1=b[i];

x2=b[j];

}

}

ht1=(((x2-x1)/(y2-y1))\*(t-y1))+x1; //linear equation for value scaling

while(m<2 && ht1!=0 && ht1<500)

{

preht=ht1;

m++;

}

float p = preht-ht1;

if(p<0)

{

p = -p;

}

if(p<4)

{

ht = ht1;

}

Serial.print(ht);

Serial.print("&");

Serial.print(t);

Serial.println(":");

Serial.flush();

preht = ht;

if ( vo >4.8 && vo<5.2) //FT PROBE

{

//state1= digitalRead(pin1);

if (state1==HIGH)

{

lcd.setCursor(15,1);

lcd.print ('L');

digitalWrite(Buz, LOW);

}

else

{

lcd.setCursor(15,1);

lcd.print(' ');

//delay(100);

digitalWrite(Buz, HIGH);

}

lcd.setCursor(0,0);

lcd.print("FMT:");

lcd.setCursor(4,0);

lcd.print("HT= mm");

lcd.setCursor(7,0);

//ht=((t-98.80)/0.183);

lcd.print(ht);

}

if(vo >3.5 && vo<4.2) //MT PROBE

{

lcd.setCursor(0,0);

lcd.print("MMT:");

lcd.setCursor(4,0);

lcd.print("HT= mm");

lcd.setCursor(7,0);

//ht=((4.908\*t)-482.7)-3; //ht=((t-100.8)/0.201);

lcd.print(ht);

lcd.setCursor(15,1);

lcd.print(' ');

digitalWrite(Buz, LOW);

//delay(100);

}

if ( vo >2.5 && vo<3.2) //ST PROBE

{

if (state1==HIGH)

{

lcd.setCursor(15,1);

lcd.print ('L');

digitalWrite(Buz, LOW);

}

else

{

lcd.setCursor(15,1);

lcd.print(' ');

digitalWrite(Buz, HIGH);

}

lcd.setCursor(0,0);

lcd.print("ST1:");

lcd.setCursor(4,0);

lcd.print("HT= mm");

lcd.setCursor(7,0);

//ht=((5.292\*t)-531.1)+8;//ht=((t-100.0)/0.194);

lcd.print(ht);

}

if(vo >1.5 && vo<2.2) //AT PROBE

{

lcd.setCursor(0,0);

lcd.print("AT1:");

lcd.setCursor(4,0);

lcd.print("HT= mm");

lcd.setCursor(7,0);

//ht=((t-100.3)/0.128);

lcd.print(ht);

//delay(100);

lcd.setCursor(15,1);

lcd.print(' ');

digitalWrite(Buz, LOW);

}

if(vo >0 && vo<1.2) //RT PROBE

{

lcd.setCursor(0,0);

lcd.print("RMT:");

lcd.setCursor(4,0);

lcd.print("HT= mm");

lcd.setCursor(7,0);

//ht=((t-100.5)/0.255);

lcd.print(ht);

//delay(100);

lcd.setCursor(15,1);

lcd.print(' ');

digitalWrite(Buz, LOW);

}

}

## Code for PID control of induction motor using two relays:

int incoming;

float output1=0, output=0, iterm=0, iterm1=0, pout=0, pout1=0;

float x,y=0;

float k;

float val,val0,t,val2;

int val1;

int ht,ht1;

int preht=0;

int setpoint = 100;

int i =5;

float error = 0;

int kp = 10;

float ki = 5.5;

//char input[100];

//int b[] = {10,20,40,60,80,100,120,150,180,210,150,120,100,80,60};//st patno:235900000

int b[] = {10,20,40,60,80,100,120,150,180,210,240,270,302,330,360,390,412,390,360,330,302,270,240,210,180,150,120,100,80,60,40};//st probe aircraft no IA 1116

//int b[] = {10,20,40,60,80,100,120,150,180,210,240,270,315}

void setup() {

Serial.begin(9600);

pinMode(6,OUTPUT);

pinMode(4,OUTPUT);

/\*while(preht=0){

incoming = Serial.available();

if(incoming>0)

{

String s = Serial.readStringUntil('&');

if(s!=0)

{

int val1 = s.toInt();

}

String s2 = Serial.readStringUntil(':');

if(s2!= 0)

{

val2 = s2.toFloat();

}

}

//val1 = Serial.parseFloat();

//Serial.print("453");

Serial.print(val1);

Serial.println(val2);

preht = val1;

}\*/

}

void loop() {

incoming = Serial.available();

if(incoming>0)

{

String s1= Serial.readStringUntil(':');

//Serial.println(s1);

if(s1!=0)

{

int a=s1.indexOf('&');

if(a>0)

{

// Serial.println(a);

int l =s1.length();

//Serial.println(l);

String s2=s1.substring(0,a);

if(s2!=0)

{

ht1=s2.toInt();

}

while(m<2)

{

preht = ht1;

m++;

}

if(ht1-preht<6)

{

ht = preht;

}

}

String s3= s1.substring(a+1);

t=s3.toFloat();

Serial.print(ht);

Serial.print(" ");

Serial.println(t);

}

}

}

error = setpoint-ht;

if(setpoint>ht)

{

iterm = iterm + ki\*error;

output = kp\*error + iterm;

y = 0;

x = output-pout;

if(x<0)

{

x= -x;

}

//Serial.println(x);

if(x<3.1)

{

digitalWrite(6,HIGH);

digitalWrite(4,HIGH);

//Serial.println(t);

}

else

{

digitalWrite(6,HIGH);

}

pout=output;

}

else

{

error = -error;

x=0;

iterm1 = iterm1 + ki\*error;

output1 = kp\*error + iterm1;

y = output1-pout1;

if(y<0)

{

y= -y;

}

if(y<3.1)

{

digitalWrite(4,HIGH);

digitalWrite(6,HIGH);

}

else

{

digitalWrite(4,HIGH);

digitalWrite(6,LOW);

}

pout1 = output1;

}

if(x<2.9&&y<2.9)

{

i = i+1;

setpoint = b[i];

x = 0;

y = 0;

delay(5000);

Serial.print(ht);

Serial.println(0);

Serial.print(" ");

Serial.println(t);

delay(5000);

}

preht = ht;

}

# GRAPHICAL USER INTERFACE

A Graphical User Interface (GUI) is designed using Microsoft Visual Studio, programmed with C# programming language. This interface is used to send calibrated fuel probe chart data to first microcontroller, log output height, time-period data from second microcontroller and save it as a txt file.

## Microsoft Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

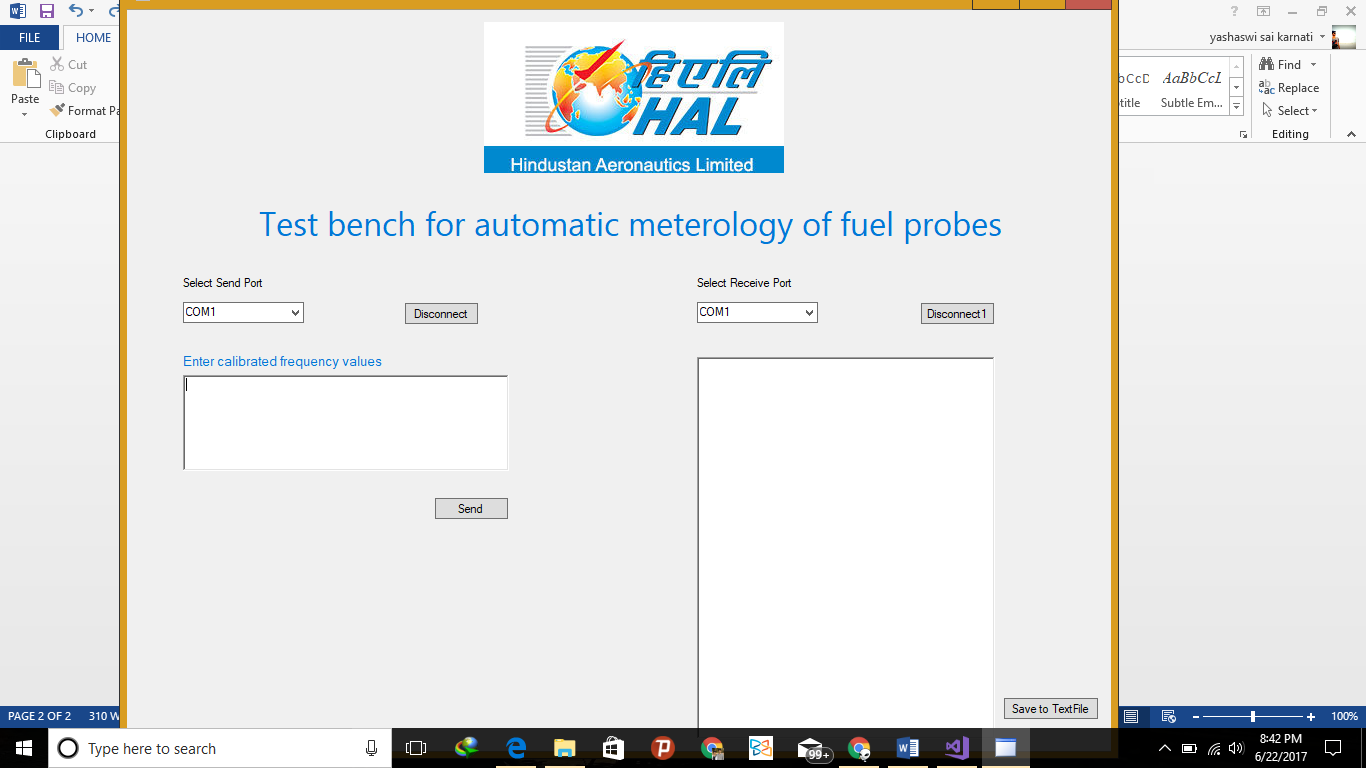
## Visual C#

C# (pronounced as see sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure.

C# is a general-purpose, object-oriented programming language.[12] Its development team is led by Anders Hejlsberg. The most recent version is C# 7.0 which was released in 2017 along with Visual Studio 2017.

## 

## Hands on GUI



On the left section of interface select ‘COM’ port to which first microcontroller is connected. Enter calibrated timeperiod values of fuelprobe seperated by comma and click send. On the right section select ‘COM’ port to which second microcontroller is connected. Expermental values of time period to the particular setpoint heights of different probes will be logged in the space provided. After completion of probe metrology click on the button “Save to TextFile”. The experimental calibration chart will be stored as.txt file in the location C:\testdata.

## Source code for GUI

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO.Ports;

using System.IO;

namespace TestPortsApp

{

public partial class TestForm : Form

{

private static SerialPort m\_serialPort;

private static SerialPort m\_serialPortReceive;

private const string Tx = "Tx >> ";

private const string Delimiter = "\r";

int m = 1;

public TestForm()

{

InitializeComponent();

LoadPorts();

}

public void LoadPorts()

{

string[] availablePorts = {"COM1","COM2","COM3", "COM4", "COM5", "COM6", "COM7", "COM8", "COM9", "COM10" };// SerialPort.GetPortNames();

string[] availableReceivePorts = { "COM1", "COM2","COM3", "COM4", "COM5", "COM6", "COM7", "COM8", "COM9", "COM10" };// SerialPort.GetPortNames();

if (availablePorts.Count() > 0)

{

ListPort.DataSource = availablePorts;

listReceivePort.DataSource = availableReceivePorts;

}

}

private void button1\_Click(object sender, EventArgs e)

{

string strInput = txtInput.Text;

string[] values = strInput.Split(',');

foreach (string input in values)

{

//label2.Text += input;

if (ConnectSendPort(ListPort.SelectedValue.ToString(), 9600))

{

SendData(Decimal.Parse(input));

//label2.Text += "Ports Available" + ListPort.SelectedValue.ToString() ;

}

else

{

//label2.Text += "No Ports Available";

}

}

// Receive Data

ConnectReceivePort();

}

// Send Data

void SendData(decimal strValue)

{

if ((m\_serialPort != null) && m\_serialPort.IsOpen)

{

m\_serialPort.DiscardOutBuffer();

m\_serialPort.Write(strValue + Delimiter);

//OnDataSend(Tx + strValue + Delimiter);

}

}

//Connect port

public bool ConnectSendPort(string portName, int baudRate)

{

if (m\_serialPort == null)

{

m\_serialPort = new SerialPort

{

PortName = portName,

BaudRate = baudRate,

Parity = Parity.None,

StopBits = StopBits.One,

DataBits = 8,

Handshake = Handshake.None,

ReadTimeout = 500, //500 ms

};

}

else

{

CloseSerialPort(m\_serialPort);

m\_serialPort.PortName = portName;

m\_serialPort.BaudRate = baudRate;

}

OpenSerialPort();

return IsPortOpen;

}

public void ConnectReceivePort()

{

m\_serialPortReceive = new SerialPort(listReceivePort.SelectedValue.ToString());

m\_serialPortReceive.BaudRate = 9600;

m\_serialPortReceive.Parity = Parity.None;

m\_serialPortReceive.StopBits = StopBits.One;

m\_serialPortReceive.DataBits = 8;

m\_serialPortReceive.Handshake = Handshake.None;

m\_serialPortReceive.RtsEnable = true;

m\_serialPortReceive.DataReceived += new SerialDataReceivedEventHandler(DataReceivedHandler);

m\_serialPortReceive.Open();

string indata = m\_serialPortReceive.ReadExisting();

txtDataReceive.Text += indata;

}

private void DataReceivedHandler(

object sender,

SerialDataReceivedEventArgs e)

{

SerialPort sp = (SerialPort)sender;

string indata = sp.ReadExisting();

txtDataReceive.Text += indata;

//Console.WriteLine("Data Received:");

//Console.Write(indata);

}

private static void OpenSerialPort()

{

try

{

m\_serialPort.DataReceived += SerialPortDataReceived;

m\_serialPort.Open();

}

catch (Exception ex)

{

System.Diagnostics.Debug.WriteLine(ex.Message + ex.StackTrace);

}

}

private void CloseSerialPort(SerialPort portName)

{

if (portName.IsOpen)

{

portName.DataReceived -= SerialPortDataReceived;

portName.Close();

}

}

private static void SerialPortDataReceived(object sender, SerialDataReceivedEventArgs e)

{

var serialPort = (SerialPort)sender;

var data = serialPort.ReadExisting();

//ProcessData(data);

}

/// <summary>

/// check is port open

/// </summary>

public static bool IsPortOpen

{

get

{

return (m\_serialPort != null && m\_serialPort.IsOpen);

}

}

private void TestForm\_Load(object sender, EventArgs e)

{

}

private void listReceivePort\_SelectedIndexChanged(object sender, EventArgs e)

{

ConnectReceivePort();

}

private void button2\_Click(object sender, EventArgs e)

{

if (m\_serialPort != null)

{

CloseSerialPort(m\_serialPort);

}

}

private void button3\_Click(object sender, EventArgs e)

{

if (m\_serialPortReceive != null)

{

CloseSerialPort(m\_serialPortReceive);

}

}

private void button4\_Click(object sender, EventArgs e)

{

DateTime dt = DateTime.Now;

string strFileName = "OutPut\_" + dt.ToString("yyyyMMddHHmm");

TextWriter txt = new StreamWriter("C:\\testdata\\"+ strFileName + ".txt");

txt.Write(txtDataReceive.Text);

m++;

txt.Close();

}

}

}

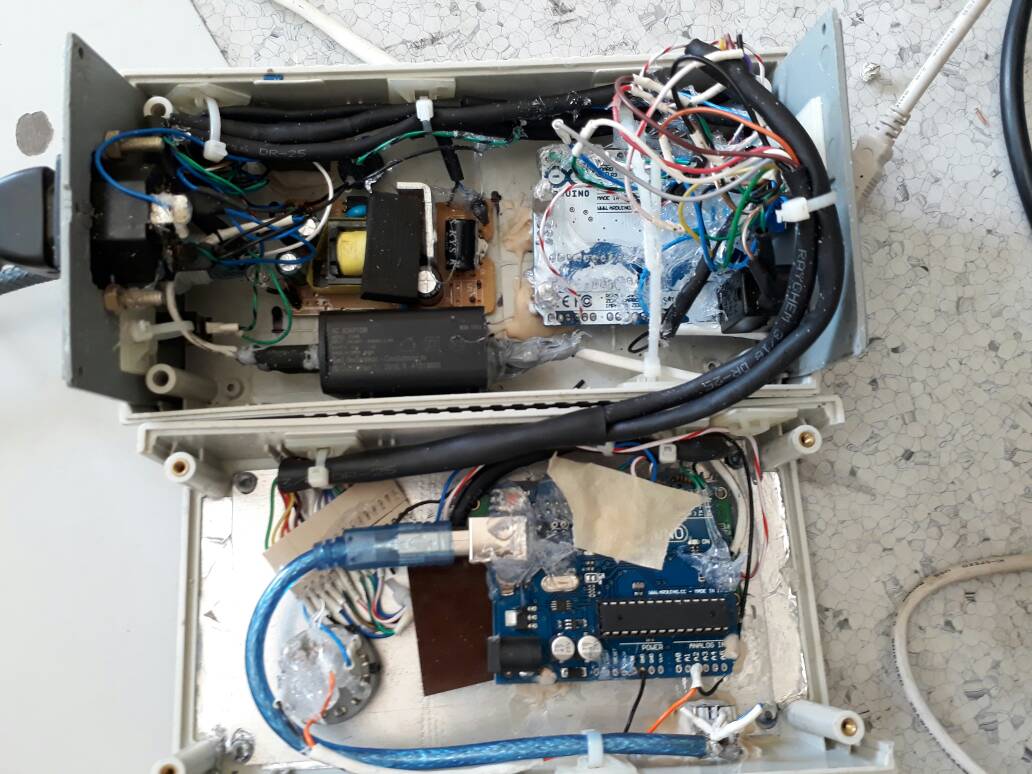
# TEST RESULTS

## Test results and discussions

The following results were obtained

* The related values of height of fuel in the tank, Frequency and Time period were verified with the standard values expected
* The low level warning which includes sound, glowing of LED and simultaneous display of low level on LCD are achieved at expected low levels of various tanks
* Provision of selection of tanks has been made so that single kit is made useful to test all kinds fuel probes
* Complete metrology of probe that is being tested is saved as a text file using GUI.
* PID control of motor is implemented so that the metrology is completely automated.
* Proportional constant (Kp), Integral constant (Ki) of PID controller are Kp = 10, Ki = 6.5.\
* In hardware section, the bulky DC power supply, the present test kit, CRO and man power required for controlling motor, taking experimental readings is completely replaced by this fully automated test kit.

## Project overview model





# OBSERVATION

Part No. 2359.00.000 SL No 152 /07

NAME: SUPPLY TANK PROBE (ALH)

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | Fuel height(mm) in Tank | Time Period in µsec | Frequency in Hz |
|  | 412 | T full=177.3 | 5640 |
|  | 390 | 175.9 | 5685 |
|  | 360 | 169.6 | 5896 |
|  | 330 | 164.3 | 6086 |
|  | 302 | 158.6 | 6305 |
|  | 270 | 152.4 | 6561 |
|  | 240 | 146.2 | 6839 |
|  | 210 | 140.1 | 7137 |
|  | 180 | 133.6 | 7485 |
|  | 150 | 128.0 | 7812 |
|  | 120 | 121.8 | 8210 |
|  | 100 | 117.8 | 8488 |
|  | 80 | 113.7 | 8795 |
|  | 60 | 109.6 | 9124 |
|  | 40 | 105.5 | 9478 |
|  | 20 | 101.4 | 9861 |
|  | 10 | T empty=100.2 | 9980 |

Part No. 2358.00.000 SL No 121 /08

NAME: REAR MAIN TANK PROBE (ALH)

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | Fuel height(mm) in Tank | Time Period in µsec | Frequency in Hz |
|  | 307 | 171.1 | 5844 |
|  | 270 | 165.8 | 6031 |
|  | 240 | 158.0 | 6329 |
|  | 210 | 150.1 | 6662 |
|  | 180 | 142.1 | 7037 |
|  | 150 | 134.4 | 7440 |
|  | 120 | 126.8 | 7886 |
|  | 100 | 122.1 | 8190 |
|  | 80 | 116.7 | 8568 |
|  | 60 | 111.5 | 8968 |
|  | 40 | 106.4 | 9398 |
|  | 20 | 101.2 | 9881 |
|  | E/10 | T empty=99.9 | 10010 |

Part No. 2357.00.000 SL No 113 /08

NAME: MIDDLE MAIN TANK PROBE (ALH)

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | Fuel height(mm) in Tank | Time Period in µsec | Frequency in Hz |
|  | 423 | T full=174 | 5747 |
|  | 390 | 170.6 | 6082 |
|  | 360 | 164.8 | 6276 |
|  | 330 | 158.8 | 6297 |
|  | 302 | 153.3 | 6523 |
|  | 270 | 146.8 | 6811 |
|  | 240 | 140.7 | 7107 |
|  | 210 | 134.9 | 7412 |
|  | 180 | 129.0 | 7751 |
|  | 150 | 123.0 | 8180 |
|  | 120 | 117.3 | 8225 |
|  | 100 | 113.4 | 8818 |
|  | 80 | 110.2 | 9074 |
|  | 60 | 106.5 | 9389 |
|  | 40 | 102.8 | 9727 |
|  | 20 | 98.9 | 10111 |
|  | 10 | T empty=98.1 | 10193 |

# CONCLUSION

* The readings of frequency, time and height obtained are verified with those obtained using current test setup.
* The analog method of taking readings using a CRO is replaced by 16x2 digital display and logging data into a computer.
* In hardware section, the bulky DC power supply, the present test kit, CRO and man power required for controlling motor, taking experimental readings is completely replaced by this fully automated test kit.
* A Graphical user interface is made for entering calibration data provided by the manufacturer and logging experimental calibration data.
* The height of fuel is measured using appropriate algorithm.
* The low level of the fuel in the tank is indicated by LED, buzzer and that corresponding height is logged on to a computer.
* PID control of height of fuel in the tank is implemented thereby completely automating the test kit for metrology of fuel probes.
* Experimental metrology data of the probe that is being tested is saved as a text file.

# References

1. Air Force Research and Development center (AFRDC) lab manual
2. Line Replacement Units (LRU) lab manual.
3. Arduino user manual : hands on research in complex system Shangai Jiao university june 17-29, 2012
4. Wikipedia <https://en.wikipedia.org/wiki/Aircraft_fuel_system>
5. Wikipedia <https://en.wikipedia.org/wiki/Fuel_gauge>
6. Understanding and Design of an Arduino-based PID Controller, Dinesh Bista, Virginia Commonwealth University, [bistad@vcu.edu](mailto:bistad@vcu.edu).
7. <http://blog.solutions-cubed.com/pid-motor-control-with-an-arduino/>