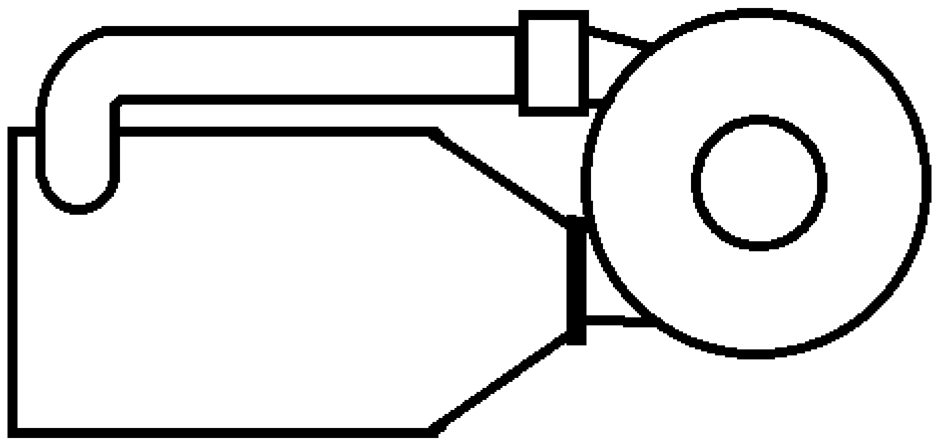
FRG Turbojet

Senior Project Report



By Tyler Vitti
Under the Advisement of Dr. John Oliver

Computer Engineering Department
California Polytechnic State University
San Luis Obispo
Spring 2012

Table of Contents

1 Introduction

The purpose of this project is to create a stable, functional, and self-sustaining turbojet engine. Digital sensors and computer control will be used to monitor the engine for steady operation and anomalous behavior.

1.1 Personnel

This project represents a significant undertaking and a large multi-disciplinary effort. Students contributed from various backgrounds and areas of expertise. The most significant contributors are:

- Tyler Vitti, 4^{th} year CPE. Project lead. Coordinated construction and testing and wrote the computer monitoring software.
- Jake Gardner, 4^{th} year CPE. Embedded hardware specialist. Assisted with construction and designed the embedded control software and sensor interfaces.
- Brian Spence, 4^{th} year CPE. Project consultant. Assisted with design, construction, and testing.
- Tim Lui, 4^{th} year ME. Mechanical consultant. Assisted with design and fabrication.
- Steve Janning, 4^{th} year ME. Design consultant. Assisted with design and mathematical modeling.
- Bryan Harris, 2^{nd} year Nursing. Machinists assistant. Assisted with fabrication and construction.
- Dr. John Oliver, Cal Poly Faculty. Project advisor. Channeled funding and provided project guidance and support.

Several other Cal Poly students and faculty, particularly from the ME department, have also participated in the project. They have provided valuable input and feedback throughout the design, construction, and testing cycles.

1.2 Inspiration

Special thanks must be extended to Don Giandomenico of RCDon. A well experienced and talented machinist, Don has designed several turbojet systems. He has carefully documented all of his efforts and findings on his website at www.rcdon.com. Don’s careful and in-depth documentation served both as the original inspiration and an invaluable reference for our project efforts. His designs and formulas are publicly available and he has also been kind enough to address a few of my design questions personally. Without the wealth of information exposed by his efforts, this project would not have been possible.

1.3 Theory of Operation

A turbojet engine is a heat machine which produces thrust in the form of expanding gasses. See Figure 1-1 for a simplified diagram. An air inlet harvests external air for a compressor, which compresses the air and forces it into the combustion chamber. The combustion chamber mixes fuel with the compressed air and sustains combustion of the mixture. The rapidly expanding exhaust gasses are forced out the exhaust of the engine, where some of their energy is harnessed by a turbine to power to compressor at the front of the engine. In this way, the exhaust from the fuel combustion provides fresh air for the flame, sustaining combustion and the operation of the engine.

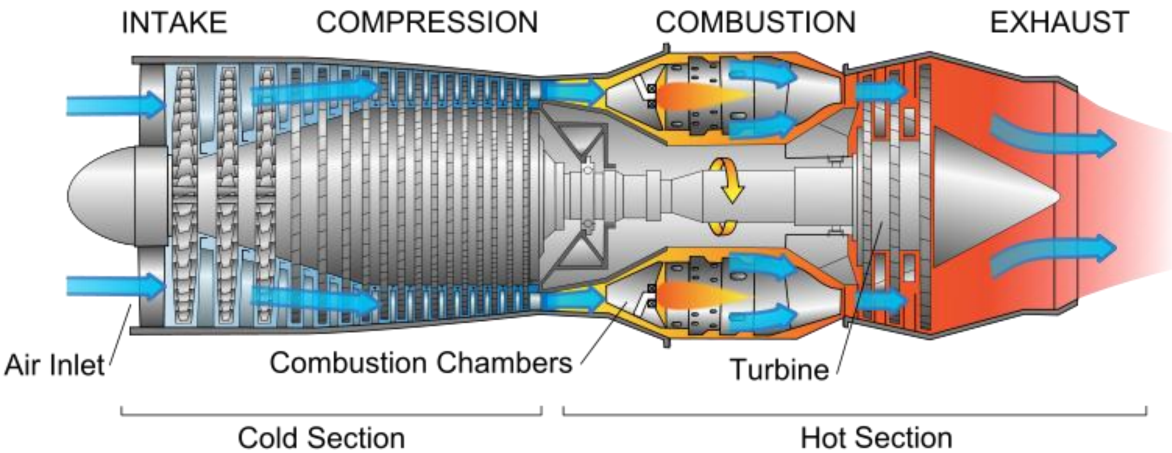


Figure 1-1: Fundamentals of a Turbojet Engine

The engine we have constructed uses a large automotive turbocharger as the compressor and turbine. This greatly simplifies the fabrication requirements of the project, as these parts would be extremely difficult to machine by hand. It also creates what is known as a “radial flow” engine, where the gasses are flowing radially from the rotating shaft, as opposed to the traditional “axial flow” engine, where the gasses are flowing axially along the shaft.

In our design, seen in Figure 1-2, air is harvested by the turbocharger compressor and forced down the compressor duct to the combustion chamber. Here, diesel fuel is burned with the incoming air to create hot gasses which are forced through the turbocharger turbine before being exhausted to the outside. Later sections will investigate the workings and components in more depth.

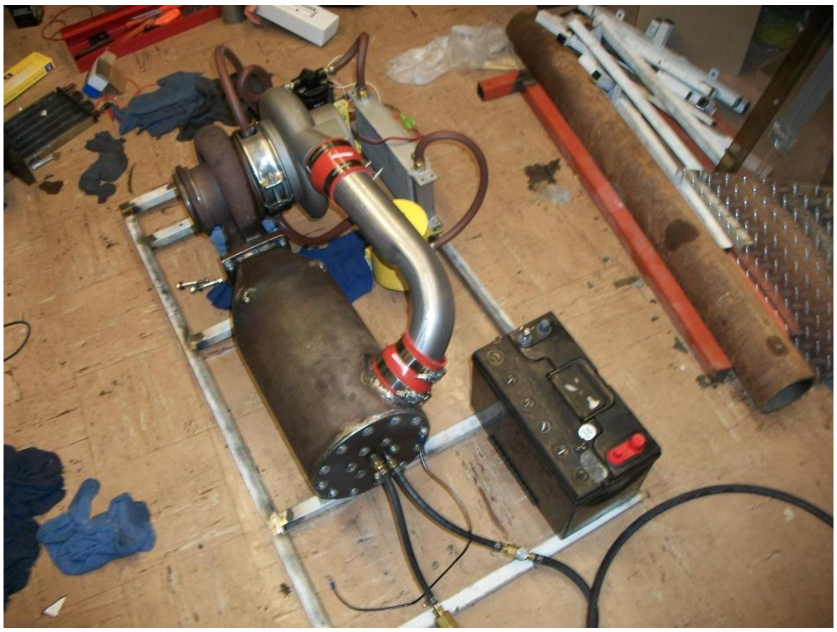


Figure 1-2: The FRG Turbojet

2 Mechanical Systems

2.1 Overview

The FRG turbojet project consists of the following components:

- Turbocharger; Houses the compressor and turbine essential to a turbojet engine
- Combustion Chamber; Mixes the fuel-air mixture and contains and controls combustion
- Oil System; Cools and circulates oil for the turbocharger bearings
- Fuel System; Stores, pressurizes, and delivers fuel to the combustion chamber
- Ignition System; Ignites raw fuel in the combustor to start the combustion process
- Embedded Control; Monitors the engine and regulates fuel, oil, and air flow
- Computer Control; Provides a remote interface for monitoring and controlling the engine

The relation of the individual components is shown here in Figure 2-1.

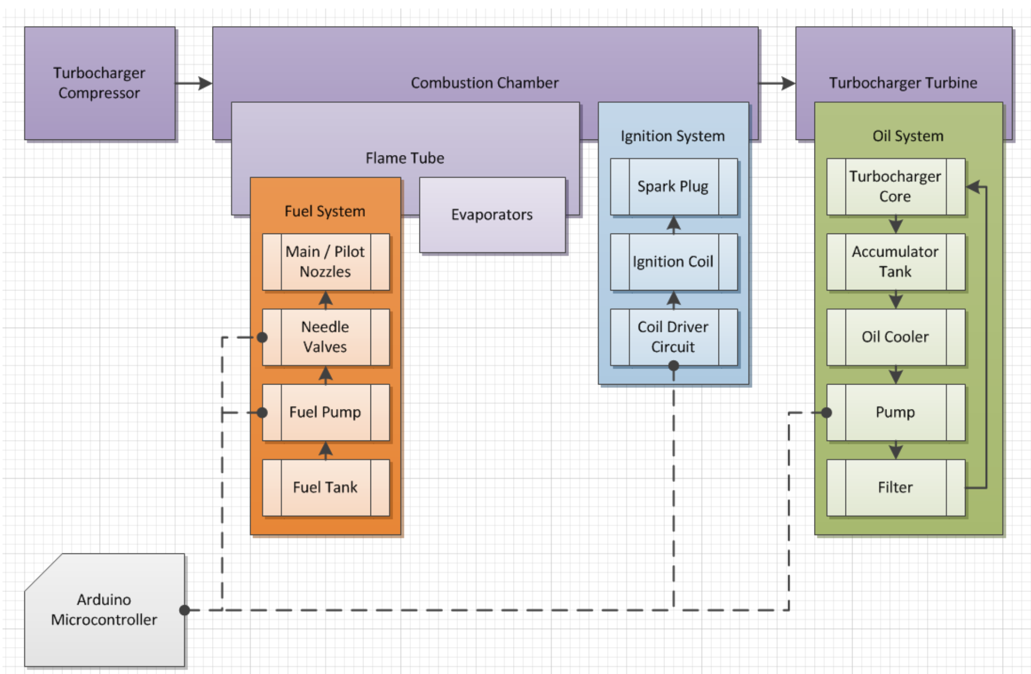


Figure 2-1: Engine Layout

2.2 Turbocharger

The turbocharger used on the engine, shown in Figure 2-2, is a T-46 equivalent. Designed originally for industrial engines of 400hp or larger, it is a unit of considerable size capable of large flow rates and high boost pressures. This makes it ideally suited for the high power sustained operations of a turbojet engine. The turbocharger's capabilities should also provide a sizeable envelope of safe operations, allowing the engine to generate a good deal of energy while remaining within safe limits.

Figure 2-2: Turbocharger

2.3 Combustion Chamber

The combustion chamber is a deceptively complicated piece of equipment. Figure 2-3 shows the various components. The combustion chamber itself serves to channel expanding gasses and contain flame. It is fabricated from 8 inch diameter steel pipe with ¼ inch outer diameter. This heavy construction serves to safely contain the high pressure and temperature generated by the combustion process. At the front of the chamber, the neck tapers down to a flange used to bolt directly to the exhaust volute of the turbocharger. This taper was fabricated by using a plasma torch to cut triangular strips out of the pipe, then using an oxyacetylene torch and repetitive force to bend the pipe together to the desired taper. Spray MIG was then used to fill in the gaps and attach the flange. On the rear end of the chamber, a small ring provides a bolt surface for the combustor backplate.

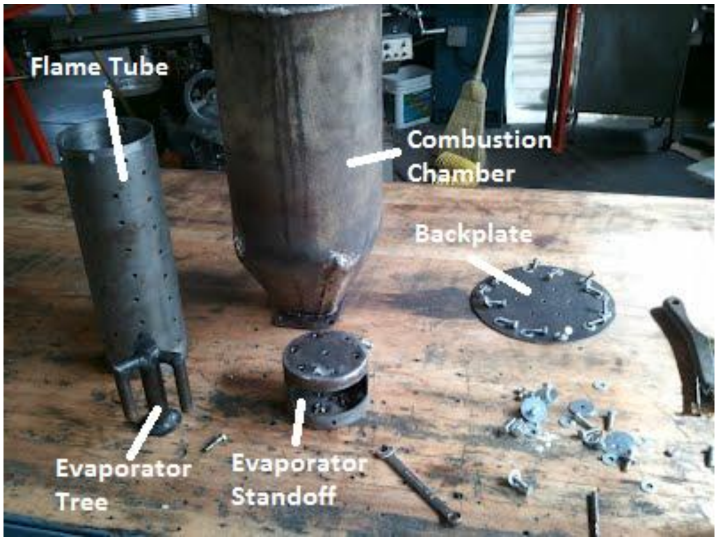


Figure 2-3: Combustion Chamber Components

The backplate is likewise made of ¼ inch steel to withstand the temperature and pressure. Due to technical difficulties with the plasma cutter at the time of construction, it had to be machined on a vertical end mill before being roughed to shape on a grinding wheel. The bolt pattern in the backplate and combustor mounting plate was made using an indexed rotary table and a drill press to make the bolt holes at equal locations.

The flametube is a 4" steel pipe with a thin wall thickness that serves to contain, shape, and cool the flame. Fuel is injected and burned at the base of the flametube, and cold air coming into the combustor is siphoned through the holes along the tube to shape and cool the flame. By the time the exhaust reaches the end of the flame tube, it should be free of all un-combusted fuel and already beginning to cool down. This design provides for a more consistent and reliable flame, as well as cooler operation for the turbocharger turbine and oil system. The flametube was particularly challenging to machine as it started with ¼ inch wall thickness. This made it too large and far too heavy. A large bore bar was used to significantly reduce the thickness of the pipe before it was placed on a rotary table under a drill press to drill the air holes.

The evaporator tree is the heart of the engine. This is where raw diesel fuel is introduced to the system through a small nozzle in the base of the evaporator. The purpose served by the evaporator tree is twofold. Because diesel's flash point is around 150 degrees Fahrenheit (200 degrees higher than conventional gasoline), it must be brought to a higher temperature before it may be readily burned. It also burns more efficiently when fully atomized. The fuel air mixture in the evaporator tree must pass down the main trunk and back one of the three arms before it is injected into the engine. During this time, the mixture is brought to temperature and the diesel atoms evaporate and atomize into the mixture. We had this assembly welded by Kevin Williams, the welding professor at Cal Poly. Due to the thin material, small components, difficult angles, and extreme sensitivity and importance of the part, we preferred to utilize Mr. Williams' expertise rather than make an attempt ourselves.

The evaporator standoff, shown in greater detail in Figure 2-4, is where incoming air is mixed with the diesel fuel and introduced to the evaporator assembly. The large opening on the side is aligned with the compressor duct as it enters the combustion chamber, creating a region of higher pressure inside the standoff. This helps accelerate air out the small opening at the top and past the primary fuel nozzle, creating the fuel air mixture and accelerating it down the evaporators and into the engine. The second nozzle protruding on the right is the ignition nozzle. It is a low-flow atomizing nozzle that creates a fine mist of diesel fuel. This mist is ignited by a wide-spark automotive spark plug and used to pre-heat the evaporators and warm the engine prior to startup. Once the evaporators reach a temperature sufficient to get the main fuel mixture above diesel's flash point (about 150 degrees Fahrenheit), the main fuel valve is opened and the engine begins normal combustion.

Figure 2-4: Evaporator Standoff

2.4 Oil System

Due to the extreme heat, pressure, and speeds present in a turbocharger core, the turbo must have a constant supply of oil to cool and lubricate the bearings. As such, it was necessary for us to build an entire oil system for our turbocharger. Figure 2-5 shows the flow of oil through the system. Once oil passes through the turbocharger bearings, it becomes heavily agitated. It must drain into the accumulator tank to allow the air bubbles to separate before being pulled through the rest of the system; otherwise air in the lines would severely impede operation. The accumulator also serves to replace oil lost from the system during routine operation.

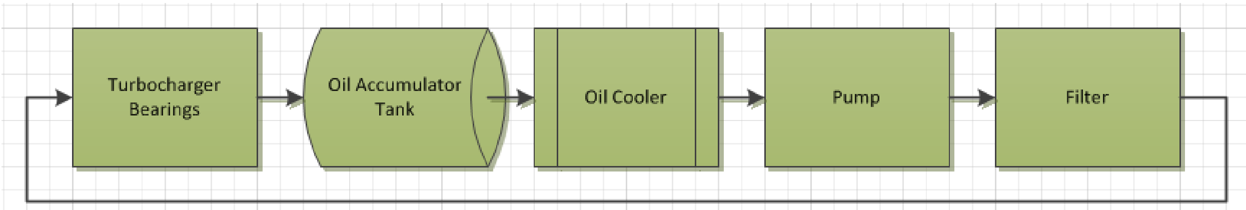


Figure 2-5: Oil System Overview

The cooler, pump, and filter are all commercial products purchased from automotive retailers. It is simply not practical to fabricate our own parts for these purposes; the pressures, tolerances, and complexity is well outside of the capabilities of the tools available. Figure 2-6 shows the completed oil system in its removable subframe. This modular design allows the entire system to easily be removed from the engine frame for service and modifications, and proved handy multiple times during construction.

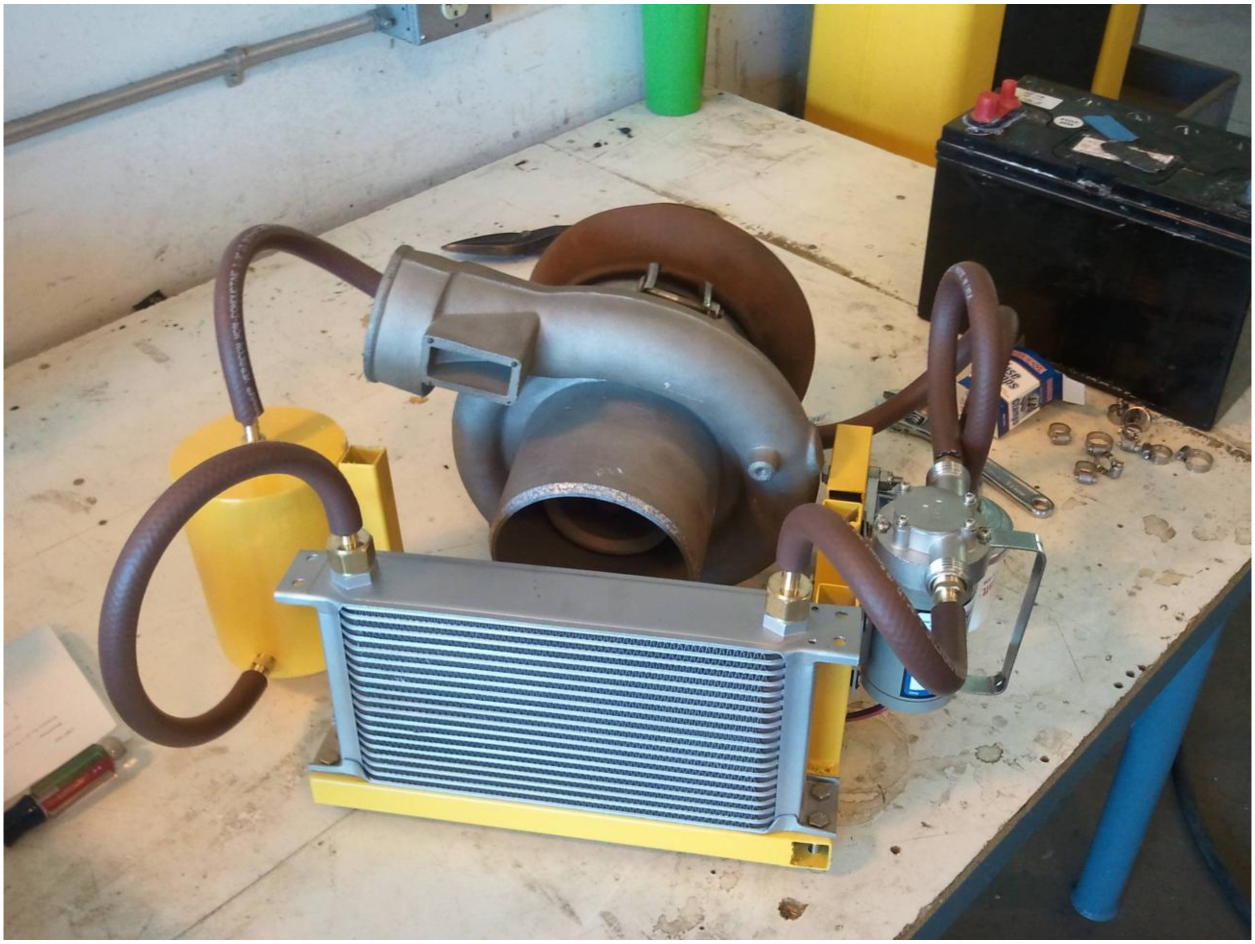


Figure 2-6: Oil System Subframe

The subsystem frame is made of welded scrap square stock. The various components are bolted onto the frame, with the exception of the accumulator. This is made from a chunk of pipe with two disks plasma cut to size and welded onto each end. A drill press and tap set was used to make the openings for the brass NPT fittings.

2.5 Fuel System

The fuel system, illustrated in Figure 2-7, is fairly straightforward. An automotive fuel pump is used to pressurize the system from the fuel tank. A built in screen protects the fuel pump from large contaminants, and an in-line fuel filter protects the valves and nozzles from finer debris. The pilot fuel nozzle produces a finely atomized fuel mist, ready for detonation by the ignition system spark plug. The main fuel nozzle is a high-flow nozzle, not capable of atomization. This is the reason for the evaporator assembly inside the combustion chamber. It also necessitates a larger, higher flow needle valve than the pilot valve.

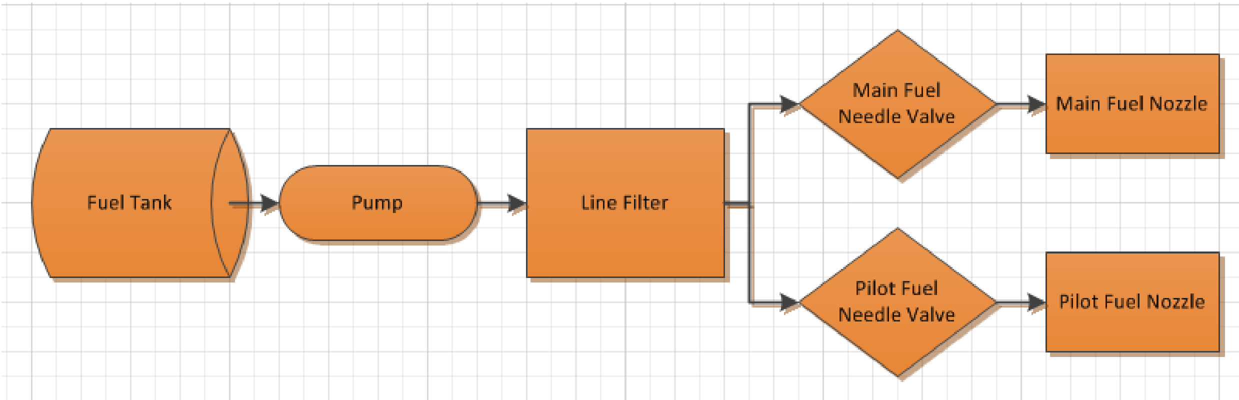


Figure 2-7: Fuel System Overview

2.6 Ignition System

Before main combustion can be initiated, the pilot nozzle must burn for a period of time to bring the engine up to heat. The pilot nozzle is ignited by the ignition system, outlined in Figure 2-8. The embedded control unit triggers the system by transmitting a 10Hz square wave. This signal is amplified by the power transistor in order to excite the ignition coil. When the wave is high, the power transistor allows the engine's battery to charge the primary winding of the ignition coil. When the wave drops low, the energy built up in the coil is released into the secondary winding. As the secondary winding contains considerably more windings than the primary, the voltage delivered is considerably higher, allowing the spark plug to create a healthy spark. The high heat inside the spark is sufficient to ignite the atomized diesel injected by the pilot nozzle, starting the pilot flame. A schematic of the ignition driver is shown in Figure 2-9.

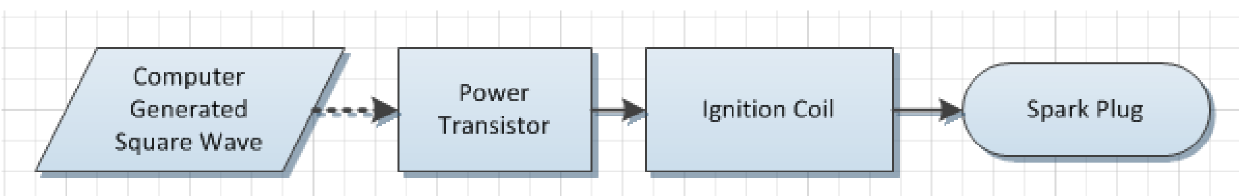


Figure 2-8: Ignition System Overview

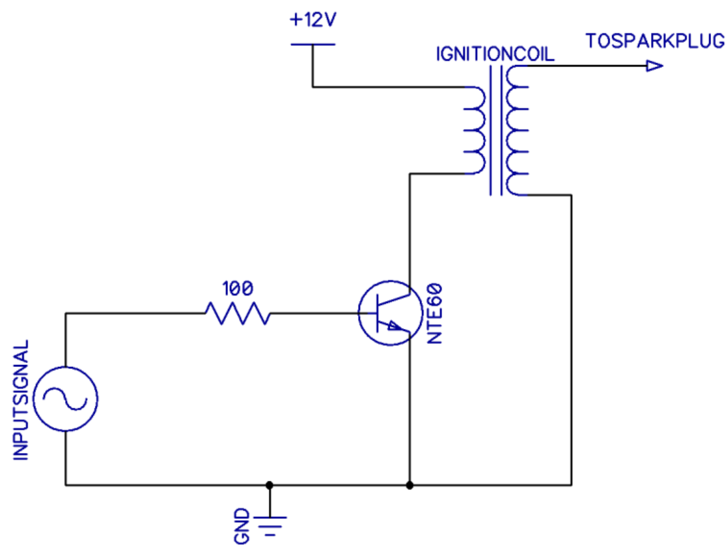


Figure 2-9: Ignition Driver Circuit

2.7 Embedded Control

The embedded control system consists of the following hardware components:

- Arduino microcontroller
- Optical RPM sensor
- Exhaust gas thermocouple
- Boost pressure sensor
- Oil temperature sensor
- Optical flame sensor
- Ignition driver circuit
- Primary and secondary fuel controls
- Primary and secondary power relays

The controls system is summarized in Figure 2-10. Datasheets for all sensors may be found in xxx.

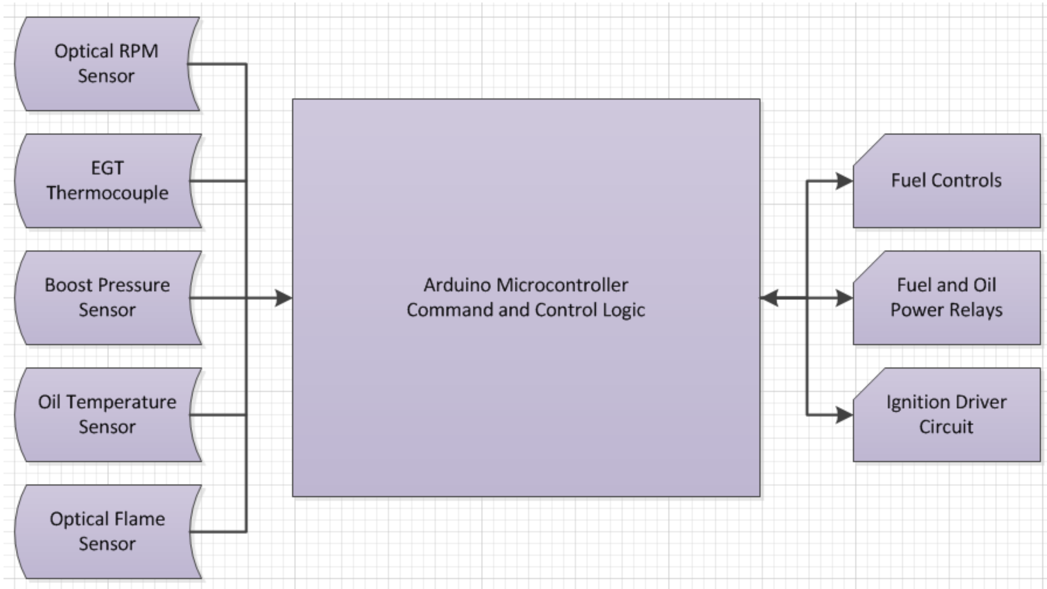


Figure 2-10: Control Systems Components

These components provide the Arduino's command and control software with all of the sensing and control capabilities necessary to properly monitor and regulate the engine's operation. This section looks only at the physical installation of the sensors. For information on the embedded command and control software itself, refer to Chapter 3: Embedded Control Software.

2.7.i Arduino Microcontroller

The Arduino Microcontroller is an Arduino Uno module, based on the ATmega328P microcontroller. This 8-bit Atmel AVR microcontroller contains 32 kilobytes of flash memory and operates at 16 Megahertz. The Arduino board exposes 6 analog input pins and 14 digital I/O pins. Of these 14 digital pins, 6 are capable of pulse width modulation. The analog inputs are used for reading the analog sensors while the digital outputs are used to run the control electronics.

Integration of the Arduino consists of a simple mounting to the engine frame as well as a proto-board for sensor electronics. The proto-board contains most of the project electronics. This includes the power relays, ignition circuit, and various pull-up and pull-down resistors.

Communication with the monitoring software is done through an Ethernet shield attached to the Arduino. The shield's onboard controller manages a functional TCP stack, enabling the embedded controller and monitoring software to communicate via TCP packets. The structure and function of these communication packets is discussed in Section 4.2.

2.7.ii Optical RPM Sensor

The Optical RPM sensor is an Omron model E3T-FD13 photoelectric sensor. It uses a red LED to detect the transition between white and black. We colored the turbocharger wheel with a white and black striping, such that the sensor can detect the revolutions of the wheel. These transitions are used by the Arduino to approximate the number of revolutions that occur each second.

Figure 2-11 shows the output circuit from the datasheet used to read the transitions. This circuit is not completely accurate, as we have replaced the indicated load with the Arduino's analog input. This resulted in unstable behavior until we discovered that the ATmega328P prefers to see resistance on its analog inputs. We added a 10k resistor in series with the Arduino input and the circuit now works as desired.

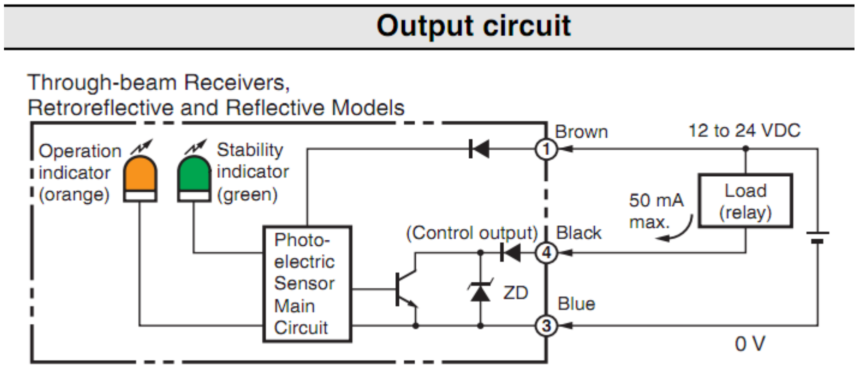


Figure 2-11: Optical Sensor Circuit Diagram

2.7.iii Exhaust Gas Thermocouple

The EGT thermocouple is a Harold G. Shaevitz Industries model EGT-DP type-K thermocouple. It provides an accuracy of +/- 2 degrees Fahrenheit and is rated for temperatures as high as 2300 degrees Fahrenheit, well above any temperature encountered in the engine. The sensor is mounted to the output of the combustion chamber using a NPT compression fitting. This location allows the control software to monitor the exhaust gas temperature just as it enters the turbocharger turbine. This way, the software can be sure that the turbine is never submitted to temperatures higher than its rated value.

The thermocouple output is very small and must be read by a specialized amplifier. This is done by an Analog Devices AD8495 IC. Figure 2-12 shows the basic operation of this IC. The amplifiers boost the minute thermocouple signal high enough to be read by the Arduino ADC, while the cold junction compensator adjusts the thermocouple reading according to the ambient temperature, providing an accurate temperature reading.

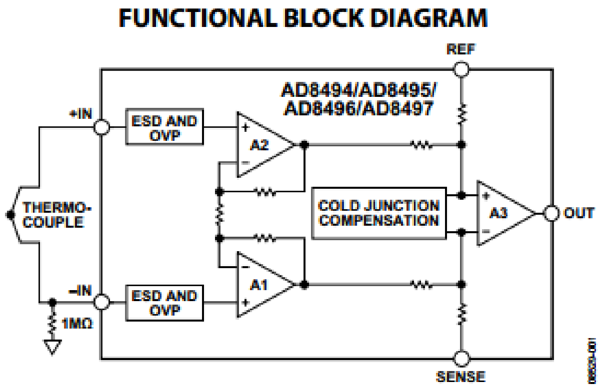


Figure 2-12: Thermocouple Amplifier

2.7.iv Boost Pressure Sensor

The boost pressure sensor is a Honeywell MLH pressure sensor. Rated for boost pressures up to 50 psi, the sensor is more than capable of functioning in our system. The sensor provides a ratiometric output ranging from 0.5 Vdc to 4.5 Vdc across its operating range. This means that each volt above 0.5 Vdc represents an additional 12.5 psi of pressure.

Integrating this sensor proved extremely simple. It attaches directly to the combustor backplate via a standard NPT fitting and connects straight to the Arduino's ADC.

2.7.v Oil Temperature Sensor

The oil temperature sensor is an AEM 30-2012 temperature sensor. It is rated up to 390 degrees Fahrenheit with an accuracy of +/- 4 degrees Fahrenheit. As seen in Figure 2-13, it is a remarkably simple circuit. As the temperature increases, the resistance of the sensor decreases, causing a voltage divider effect at the Arduino's analog input. Sensor integration was also very simple, requiring only a standard NPT fitting in the bottom of the oil reservoir.

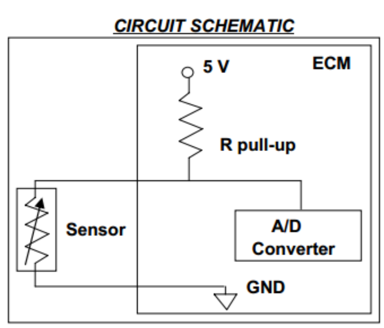


Figure 2-13: Oil Temperature Circuit Diagram

2.7.vi Optical Flame Sensor

The optical flame sensor is a Honeywell C55 optical flame detector. It is a standard photo-resistor designed explicitly for burners. When the combustor flame lights it causes the sensor to decrease its resistance. This drop is detected by the Arduino's analog input, indicating successful ignition.

Unfortunately, the shops closed for the year shortly after our receipt of the sensor. Despite its simple operation, it was not possible for use to integrate it into the combustion chamber. This will be an immediate priority for the next project owners, as the software is already designed for the sensor.

2.7.vii Ignition Driver Circuit

The ignition driver circuit is documented in Section 2.6: Ignition System. Please refer there for a detailed explanation.

2.7.viii Fuel Servos and Power Relays

The fuel servos are basic Sparkfun model 43R continuous rotation servos. They use pulse width modulation from the Arduino's digital outputs to drive the motors forward or backward at varying speeds. These servos are to be connected to the fuel needle valves. Unfortunately, the project member to whom this task was delegated neglected to perform the integration. As such, the engine currently relies on manual activation of the fuel valves. This will be an immediate priority for the next project owners, as the software is already designed for the fuel servos.

The power relays are Omron G4A power relays. Their contacts are rated for 20 amps of current flow. This is ample capacity to provide the fuel pump and oil pump with power. Integration of the power relays required only that they be soldered to the proto-board and have their coils attached to the Arduino's digital outputs for activation.

3 Embedded Control Software

3.1 Overview

The design of the embedded control system went through a number of iterations due to time constraints and problems that arose with the connected hardware. The overall flow of the control system is shown in Figure 3-1. All control software was written with the Arduino IDE in the C/C++ Wiring

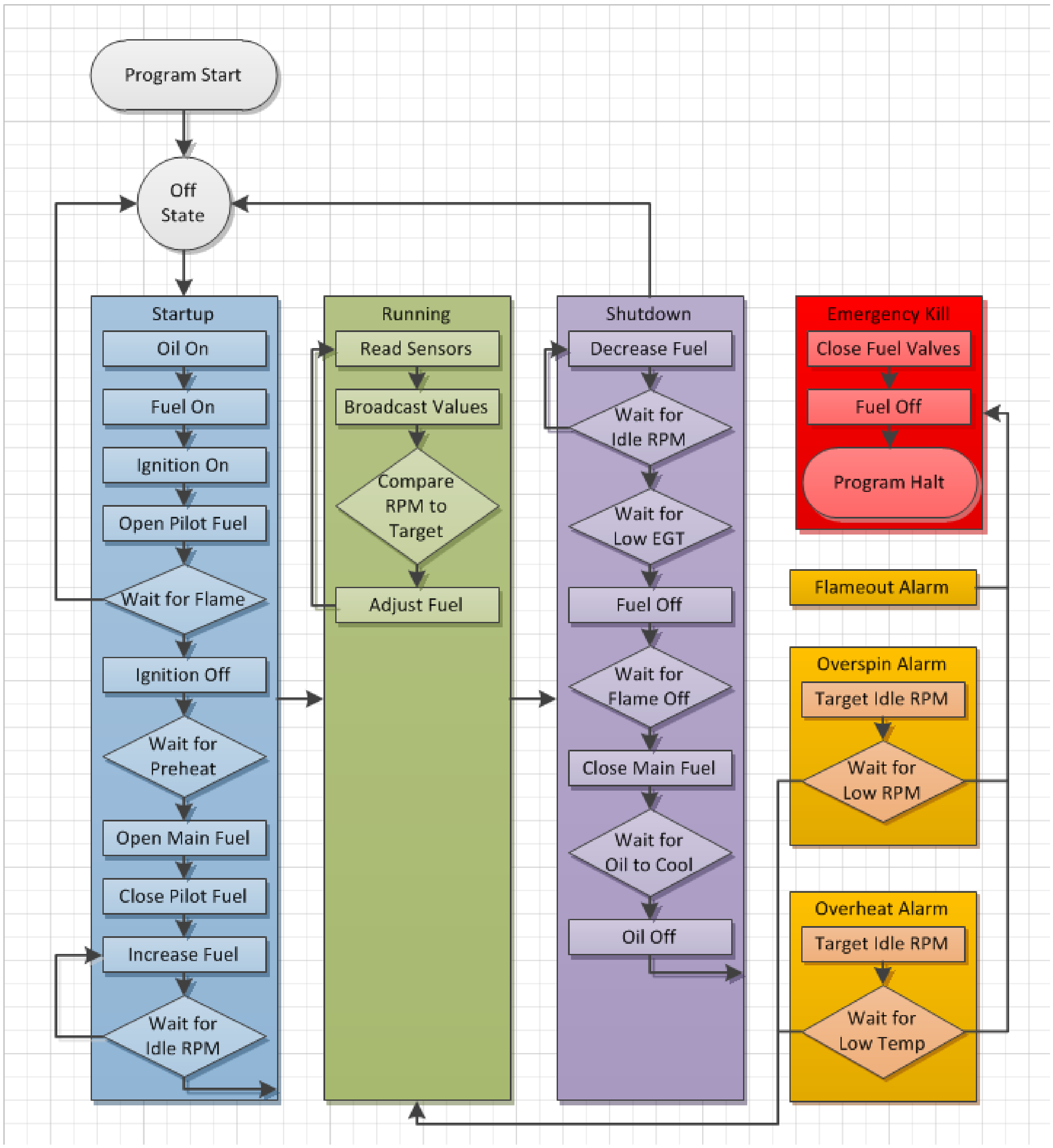


Figure 3-1: Embedded Control Flow

library. The final iteration of the embedded control software was designed to have full sensor capability and networking function, but little to no automated control of the turbojet system. This would allow the embedded system to communicate with the computer monitoring software, while the turbojet itself would still have to be started and run manually. Implemented remote start capability and automated control will be a future goal for this project.

3.2 Operation

The control system was originally designed to automatically run the startup procedure and keep the turbojet in a running state unless an emergency situation was encountered. However, due to the problem of implemented the fuel servos, and the time constraints of the project, this design was scaled down so that the embedded control would be used primarily for purely monitoring purposes.

3.2.i Startup Procedure

Upon powering up the Arduino microcontroller, the relay for the oil system would be switched on. However, it was discovered that the digital output pins on the Arduino were not able to supply the 5 volts required to switch on the relays. Due to this problem, the oil and fuel systems had to be switched on manually. The start button on the turbojet frame would cause the control system to switch on the fuel relay (though this did not occur due to the aforementioned relay problem), and fire the ignition system as long as the button was held down. If the embedded control detects a client via the ethernet connection, it sends an initial configuration packet, as well as notifications of startup, fuel, oil, and ignition.

3.2.ii Running State

Once ignition has been reached, the embedded system no longer plays an active part in control, purely monitoring sensors and sending state packets if a computer client is connected. In regards to the sensors, the optical flame sensor and EGT thermocouple were not implemented due to manufacturing problems. Because of this, the control system was not able to detect an overheat of exhaust gas or a flameout. However, it could detect overspin or oil overheat conditions, and would send a corresponding event packet to the computer client should such an emergency occur. All packet types are described in greater detail in Section 4.2. If not for the problem switching the relays, the fuel pump could be shut off with either the stop button on the turbojet frame, or by a stop command sent by the computer client.

3.3 Implementation

The final iteration of the embedded control system was designed to have control of the fuel and oil relays, start and stop button inputs, and sensor inputs for RPM, boost pressure, and oil temperature. The layout for the system is shown in figure 3-1.

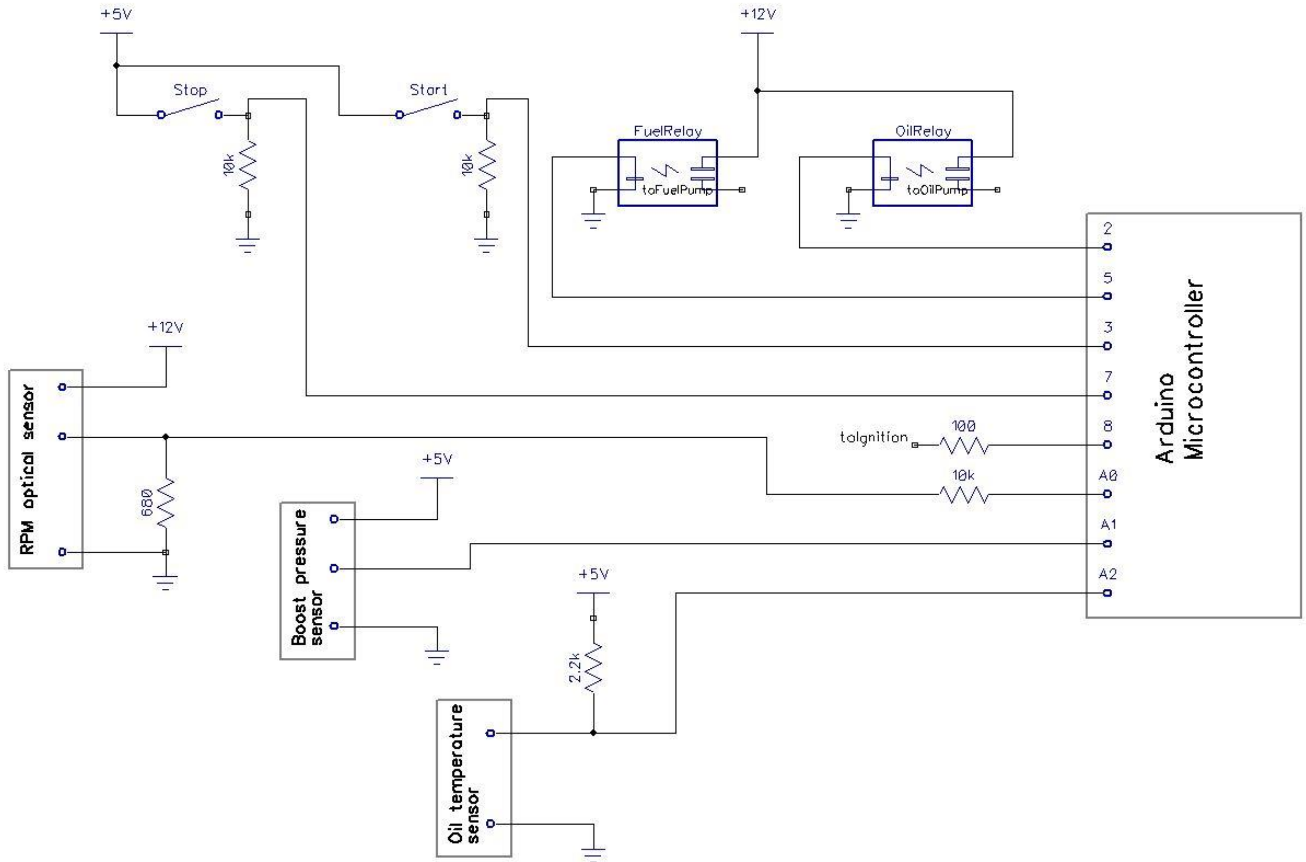


Figure 3-2: Embedded Control System Schematic

4 Computer Monitoring Software

4.1 Overview

The PC monitoring software is shown in Figure 4-1. The application is written in C# on the 3.5 dotNET framework and uses Windows Presentation Foundation for its GUI display. The software is therefore Microsoft Windows specific. The host computer must have a minimum resolution of 1024 x 700 to display the full GUI and an Ethernet port with TCP capabilities. Plans exist to convert the application to run on 802.11 WiFi in the future.

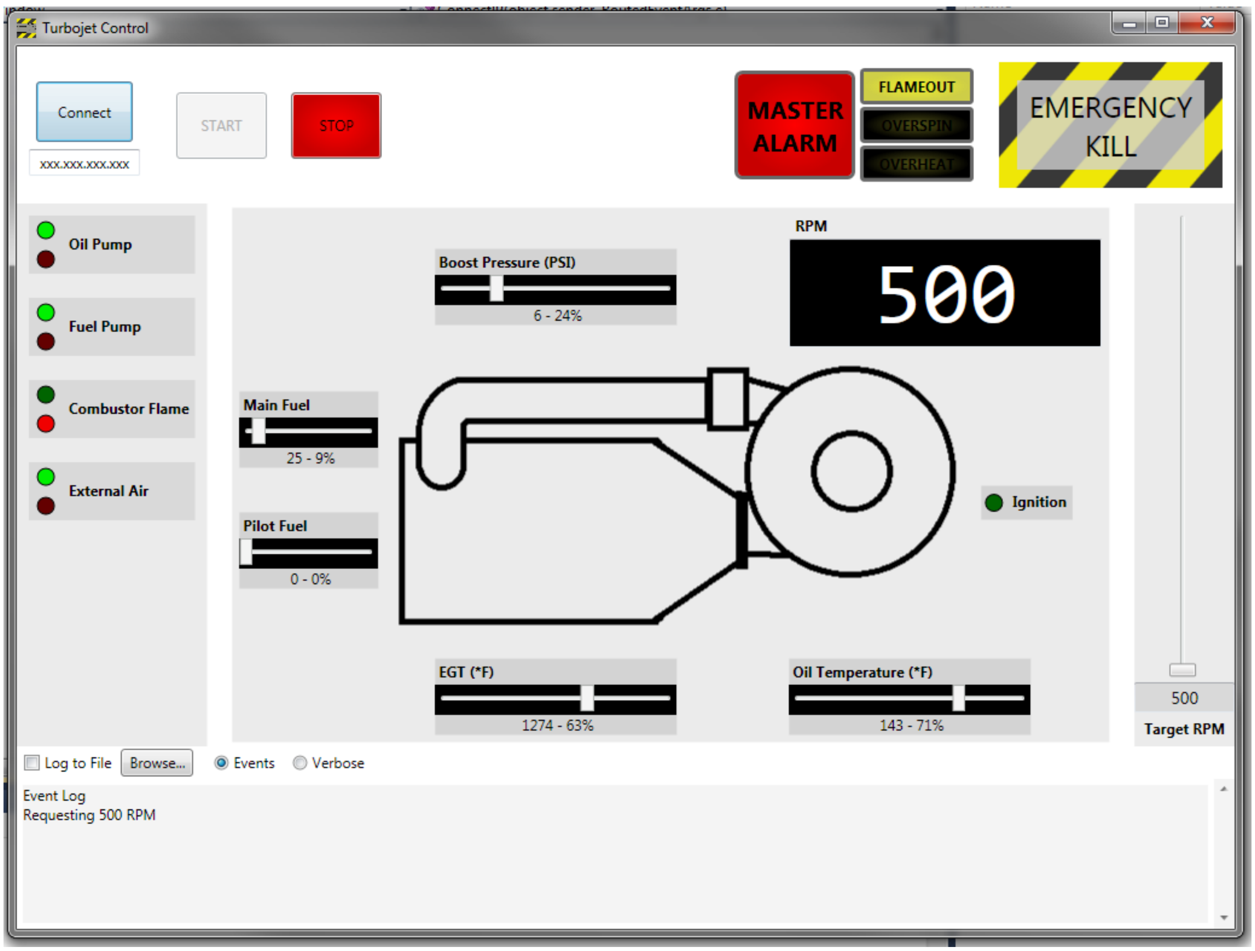


Figure 4-1: Computer Monitoring Software

4.2 Operation

The computer monitoring software contains no control logic. It serves only as a graphical interface with the embedded command unit. This way, any modifications to engine operation need be made only to the Arduino code. All information regarding engine operation is maintained exclusively on the Arduino

and relayed to the monitoring software when a connection is established. Thanks to this design approach, the engine may in fact be started and ran normally without connecting the monitoring software. However, the monitoring software is the only way to request a higher RPM from the engine control software; otherwise it will only idle until the user initiates shutdown.

The monitoring and control software communicate via the exchange of data packets over the Ethernet interface. Figure 4-2 illustrates the flow of data. Four types of packets are sent across the interface; Initial Configuration, Routine States, Event, and Command. The roll of each packet type is explained in detail below.

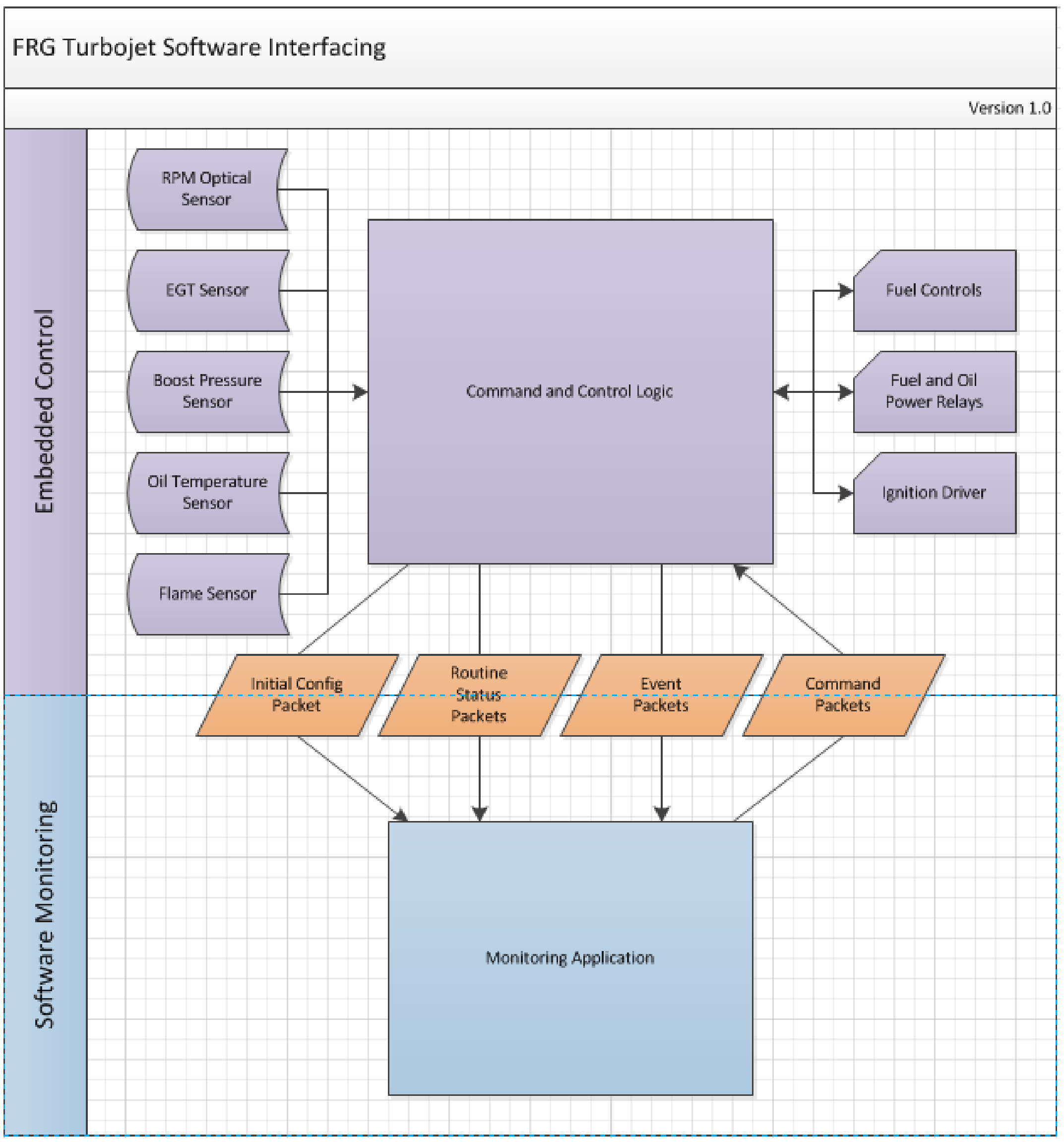


Figure 4-2: Software Interface

4.2.i Initial Configuration Packet

On startup, the monitoring software knows nothing about the engine. Because it may be connected at any time during the engine's operation, it is necessary for the embedded control unit to transmit the complete current state of the engine when the monitor connects. The initial configuration packet serves this purpose. It notifies the monitoring application of the current readings, settings, and minimum and maximum values of each component of the sensor array. This allows settings such as the maximum allowable RPM or minimum sustainable boost pressure to be updated and stored exclusively on the Arduino.

4.2.ii Routine Status Packets

Routine status packets are sent regularly by the embedded controller at a constant interval. They contain a stream of information from the constantly operating onboard sensors, including the following:

- RPM of the turbine shaft, as indicated by the optical RPM sensor. Values range from 0 to the maximum threshold of the sensor
- EGT, as indicated by the EGT thermocouple. Values range from 0 to 2000 degrees Fahrenheit
- Boost pressure, as indicated by the pressure sensor. Values range from 0 to 50 psi
- Oil temperature, as indicated by the oil thermistor. Values range from 0 to 390 degrees Fahrenheit
- Pilot fuel valve position, as approximated by the Arduino's servo controller
- Main fuel valve position, as approximated by the Arduino's servo controller

4.2.iii Event Packets

Event packets are used for asynchronous or one-time notifications. These packets often signal a change in system state. The supported event packets are:

- Fuel power relay on/off, as required by the control software
- Oil power relay on/off, normally active whenever the Arduino has power
- Flame sensor on/off, as indicated by the optical flame sensor
- Start or Stop signal requested from embedded controller
- Add / remove external air, as required by the control software to start the engine
- Ignition driver on/off, as required by the control software to initiate combustion
- Flameout event, as detected by the control software. Irrecoverable
- Overspin event, as detected by the control software. Recoverable by decreasing RPM
- Overheat event, as detected by the control software. Recoverable by decreasing engine heat
- Emergency Kill event, as requested by the user. Immediately stops fuel delivery

4.2.iv Command Packets

Command packets are used by the monitoring software to relay the user's requests to the embedded controller. These packets are the only communication from the monitor to the controller, and represent the only commands available to the end user. The available command packets are as follows:

- Monitoring software connected

- Start or Stop signal requested by the user
- Target RPM requested by the user for the controller to aim for
- Emergency Kill requested by the user
- Monitoring software disconnected

4.3 Interface

The application interface is broken into three main components; Annunciation and Control, Monitoring, and Logging. Annunciation and Control consists of the connection interface, the start / stop controls, the RPM selector slider, emergency kill switch, and alarm annunciator panel. Monitoring consists of the oil, fuel, flame, and external air indicators, as well as the central status view containing RPM, EGT, oil temperature, boost pressure, and fuel flow gauges. Logging consists of the bottom panel's event log and log settings. Refer above to Figure 4-1 to see where the different controls lie on the application.

4.3.i Annunciation and Control

The annunciation and control panel of the application exposes the engine's controls to the user. The user begins operation by entering the IP address of the controller. This value is hard-coded into the Arduino's Ethernet interface, and should be present somewhere on the engine frame. The 'Connect' / 'Disconnect' button initializes connection with the controller or disconnects the monitor, respectively. The 'Start' and 'Stop' buttons start the engine's startup and shutdown routines, respectively. The 'Emergency Kill' button initializes an emergency shutdown. This is not recommended, as it bypasses the normal shutdown procedure and risks permanent damage to the engine. Finally, the RPM slider and text box on the side allow the user to request a new target RPM, either via the slider or a direct key-in. The C# event handler for each function composes the appropriate command packet, sends it down the interface, and saves the action to the log.

The annunciator panel is used to notify the user of an alarm condition. Upon receiving an alarm packet, the C# handler toggles the appropriate alarm panel state variable. This triggers a separate event unique to the type of alarm being triggered which causes WPF to redraw the panel with the appropriate indicators illuminated. Once the alarm condition subsides, the same process occurs with a silence alarm message.

4.3.ii Monitoring

The monitor panel is the heart of the application. It visualizes all of the engines controls and gauges in one central display. Indicator lamps down the left side signal system states, such as proper flame or the need for external air. Gauges are displayed using horizontal sliders around the central engine illustration. Each display is approximately co-located with the location of the actual sensor or component being monitored. This makes for a simple and intuitive display from the user's point of view. The initial configuration packet gives minimum and maximum expected values for each gauge, allowing the slider positions to give appropriate at-a-glance indications. In addition to the visual representation, the reading's percentage of nominal values is also calculated and displayed. This allows out of bounds readings to display as a negative percent or value over one hundred percent. The actual gauge reading is always displayed in addition, so an abnormal reading is not clipped to the minimum / maximum nominal values.

Information regarding gauges and state indicators is received from routine status packets and event packets, respectively. The C# event handler sets the appropriate variable, which causes WPF to redraw with the appropriate indications.

4.3.iii Logging

The application's logging panel allows the user to see and diagnose system events. By default, only events of interest are logged. These include:

- User events, such as 'Start', 'Connect', or requesting a new RPM
- Engine events, such as 'Oil On' or 'Ignition Off'
- Alarm conditions

The user may change from normal event mode to verbose mode. This will cause the application to log the sending and receipt of every network packet, as well as the contents of the packet. The user may also chose to enable or disable the logging of timestamps associated with each message.

The application also has the capability of redirecting log output to a file. If desired, the user may specify a file on their computer to receive a copy of all log messages, exactly as seen in the log window. This is particularly useful when used with timestamps and verbose mode, as it allows for a closer examination of system behavior.

5 The Design and Construction Process

This chapter will explain the major milestones, timeline, and obstacles encountered along the way. Examples include flame and ignition tests, the order of major construction events, and issues encountered with ignition and sustained idle.

5.1 Timeline

The project officially began on September 21st, 2010. With the exception of two major hiatuses, work proceeded on a regular basis. All fabrication has been done in the Cal Poly Hangar. The Hangar is open Tuesday, Thursday, and Saturday of all regular school weeks. Originally, fabrication was done only on Saturdays. Once the project became accepted as a senior project and received proper funding, fabrication stepped up to all open days. Output increased dramatically during this time.

Following is a summary of all work days logged on the project:

- 10/21/2010 – Purchased the turbocharger to be used for the project. It took considerable time to find a turbo of appropriate size, construction, and price.
- 11/06/2010 – Lathed the first half of the flame tube from ¼" down to a more appropriate thickness
- 11/13/2010 – Finished lathing the flame tube and began the combustion chamber taper
- 11/20/2010 – Finished the combustion chamber taper and cut out the turbo mounting flange
- 12/05/2010 – Welded the combustion chamber taper and finished the flame tube
- 12/11/2010 – Welded the turbo mounting flange and started on the evaporator base

- 01/08/2011 – Worked on the evaporator base and made first attempt at combustor backplate
- 01/22/2011 – Cut the combustion chamber and flame tube to length. Worked on the evaporator base
- 02/12/2011 – Second attempt at combustor backplate
- 02/26/2011 – Bolt patterns for backplate and evaporator base. Worked on evaporator base. Started evaporator tubes
- 10/01/2011 – Welded evaporator tubes and assembled flame tube
- 10/06/2011 – Assembled combustion chamber
- 10/15/2011 – Started fuel standoffs in the evaporator base
- 11/10/2011 – Worked on fuel standoffs
- 11/17/2011 – Finished fuel standoffs
- 11/19/2011 – First flame test. Unsuccessful
- 12/03/2011 – Second flame test. Success!
- 04/03/2012 – Acquired new components. Started on oil subsystem
- 04/04/2012 – Worked on oil subsystem
- 04/05/2012 – Finished oil subsystem
- 04/07/2012 – First idle test. Unsuccessful
- 04/10/2012 – Started on engine frame. Repaired oil subsystem
- 04/12/2012 – Worked on engine frame. Mounted combustion chamber to frame
- 04/14/2012 – Second idle test. Unsuccessful
- 04/17/2012 – Examine evaporators and make gaskets for combustion chamber and turbo flange
- 04/21/2012 – Third idle test. Unsuccessful
- 04/24/2012 – Started on ignition subsystem
- 04/26/2012 – Worked on ignition subsystem
- 04/28/2012 – Finished ignition subsystem. First ignition test. Success!
- 05/01/2012 – Started compressor tube
- 05/08/2012 – Finished compressor tube. Fourth idle test. Success!
- 05/17/2012 – Improve oil subsystem and start battery pan
- 05/19/2012 – Finished oil subsystem and ignition subsystem mount
- 05/24/2012 – Began sensor integration. Added oil cooler fans.
- 05/29/2012 – Painted engine frame
- 05/30/2012 – Continued sensor integration
- 05/31/2012 – Worked on sensor integration and wiring. Senior project expo

5.1.i Milestones

The following major components are critical to the project. The completion of each one proved an important milestone along the fabrication path.

- Combustion Chamber – Consists of the evaporators and flame tube assembly
- Oil Subsystem – The accumulator, cooler, and pump used to cool the turbocharger bearings
- Ignition Subsystem – The driver, coil, and sparkplug used to ignite the fuel during startup

- Engine Frame – Packages all of the components into one complete transportable system
- Sensor Integration – Interfaces the embedded controller with the engine

Each major physical component was completed on the following dates:

- Combustion Chamber – 11/17/2011
- Oil Subsystem – 04/05/2012
- Ignition Subsystem – 04/28/2012
- Engine Frame – 05/23/2012
- Sensor Integration – 06/03/2012

In addition to the physical components, the following are the major functional milestones critical to operational completion. Each milestone represents a distinct capability bringing the engine closer to full operation.

- Flame Test – Completed combustion chamber mixes fuel and air and burns properly
- Ignition Test – Ignition subsystem ignites fuel air mixture without external assistance
- Idle Test – Engine maintains combustion without external assistance
- Sensors Test – Embedded controller can properly obtain and relay all sensor readings
- Power Test – Embedded controller brings engine to full power in a safe and controlled manner

Each major physical component was completed on the following dates:

- Flame Test – 12/03/2011
- Ignition Test – 04/28/2012
- Idle Test – 05/08/2012
- Sensors Test – 06/04/12
- Power Test – 06/04/12

5.1.ii Major delays

Three major delays hindered progress on the project. For some time the engine was a club project subject to the free time of the project members. As such, the first two delays were simple hiatuses due to mundane factors outside the project. Fabrication first stopped from 02/26/2011 to 10/01/2011 due to sudden personal events in the lives of several project members. Fabrication stopped again from 12/03/2011 to 04/03/2012 due to an extremely heavy winter quarter for several project members.

Once the engine was accepted as an official senior project, commitment and progress have both increased considerably. Only one other issue caused a considerable delay. The idle test milestone proved extremely difficult to accomplish. Originally targeted for 04/07/2012, the engine was incapable of sustained combustion until 05/08/2012, a full month later. Several issues plagued successful engine idle. The predominant issues were insufficient air flow for start and low boost pressure.

The airflow issues proved an easy fix. The considerable size and mass of the turbocharger core combined with the appreciable volume of the combustion chamber require a large amount of air volume to start

the ignition process. The weed blower originally purchased for the project was sufficient for a simple flame test but could not move enough air for sustained idle. The Cal Poly groundskeepers, conveniently collocated with the hangar shop, were kind enough to loan us a large gas weed blower that proved sufficient for engine ignition.

The boost pressure issues were more prevalent. Sustained combustion requires the turbocharger compressor to provide both airflow and boost pressure to the combustion chamber. Poor welds, leaky seals, and unsealed ducting made it impossible to develop any boost pressure within the combustion chamber. Fixing this problem required re-doing several welds, fabricating gaskets for all joints, and fabricating an airtight compressor duct. Once the boost pressure issues were addressed, the engine started up and idled successfully, costing us a full month of development time. This lost time has emerged during sensor and computer integration, leaving only two weeks to integrate, develop, and test the computer control components.

5.2 Major Obstacles to Development

The first obstacle to development came during the fabrication of the combustion chamber. One of the other students in the shop had damaged the optical plasma cutter. This tool is invaluable for precisely cutting the thick steel plating used throughout the project. Without the cutter, progress slowed considerably as we were forced to adopt new methods for cutting our designs. Oxyacetylene cutting worked well enough, until someone broke that unit as well. After that, we were forced to use the mill to perform cutting. This was a slow, inaccurate, and error-prone process. The mills and jigs are not designed for parts of our size and weight. We ended up destroying several of our parts before getting them close enough to specification to be used.

The next obstacle occurred on 11/19/2011 during the first flame test. During this test, we discovered that the combustion chamber did not have sufficient airflow to burn properly. Fortunately, this proved to be an easy fix. During our next flame test, we used a considerably larger air mover. The combustion chamber performed as expected, and we were free to continue development.

Once we had a successful flame test under our belt, development proceeded smoothly until the completion of the oil system and the first sustained idle test. During this test, two problems with the oil system became apparent. The subframe was originally designed to have the turbocharger inlet directly behind the oil cooler, pulling cool air over the fins. However, this requires the turbocharger bearing ports to be mounted horizontally. This causes large amounts of oil to spill into the turbo volutes, resulting in extreme amounts of oil smoke. Second, the oil pump selected for the oil system could not perform at the pressures asked of it, and it failed partway through the first idle test. These problems necessitated a rebuild of the oil system, using a newer much more powerful pump and positioning the turbocharger compressor several inches higher.

The largest single obstacle to development occurred during the sustained idle tests. We simply could not get the engine to start. Originally this was due to insufficient airflow. We borrowed a full size weed blower for Cal Poly Grounds, but this was still not getting the engine started. We later determined that the problem was also due to insufficient boost pressure in the combustion chamber. We added gaskets

to the combustion chamber junctions and added an airtight combustor duct. Finally, we were able to start the engine, 30 days after our scheduled milestone.

Due to the turbocharger's elevated position above the oil cooler, the oil temperature rose rapidly without cooling airflow. The idle test had to be cut short due to these rising temperatures. While not a considerable delay, this did force us to step back and address the oil system before proceeding with development. The solution was to simply zip tie two 120mm computer cooling fans to the cooler. This has proven successful in moving enough air over the fins to maintain a more stable oil temperature.

The delayed idle test also posed a new problem. While addressing the problems with sustained idle, we neglected to purchase the electronics for the engine. When we finally did purchase the sensors, the shipping delay meant we only had two weeks to test and integrate them. This massive time delay (from nearly two months to less than three weeks) meant that there simply wasn't sufficient time to integrate all of the control electronics. The EGT thermocouple, optical flame sensor, and fuel servos had to be omitted from the end product due to tight time constraints. Even though these parts have been tested and programmed, there simply wasn't time to integrate them with the engine systems.

5.3 Tools and Techniques Used

While this project utilized a huge variety of tools and practices throughout its fabrication, a few were used repetitively enough to earn particular interest.

5.3.i MIG Welder

Metal Inert Gas welding, or MIG welding, is a process in which a consumable electrode wire is fed continuously into a pool of molten metal to form a mechanical bond. A shielding gas is used to protect the pool from contaminants while it solidifies. This form of welding is very fast, but also very messy. MIG tends to produce lower quality welds due the phenomena of dross and porosity.

We used MIG welding for many of our welds due to its speed and extreme ease of use. It was also especially useful when sealing the combustion chamber tapers and filling in the spark plug mount hole. These parts had extremely large air gaps, up to ¼" in some places. MIG's fast wire deposit rate makes it the ideal process for filling in these large gaps.

5.3.ii TIG Welder

Tungsten Inert Gas welding, or TIG welding, is a process which uses a non-consumable electrode to produce a pool of molten metal. Filler rod is added to the pool to create a weld bead. As in MIG, a shielding gas is used to protect the pool during solidification. TIG welding provides the operator much greater control over the weld bead because the electrode remains constant throughout the process. By controlling the distance of the welding torch, speed, amount of filler rod added, and current through the torch, a skilled operator can produce extremely high quality welds.

TIG welding proved very useful for the majority of our welds. Its fine control and quality of bead made it ideal for the more precise welds on components such as the frames and subframes. Smaller, cleaner beads were much easier to grind down and were generally much less likely to burn through the metal.

5.3.iii Lathe
The rotary lathe also proved useful to many stages of combustion chamber manufacturing. The ability to easily remove and shape material on round components meant it was the perfect tool for parts such as the flame tube and evaporator base. The high precision afforded by the machine made it easy for us to introduce close tolerances and tight fits in the combustion chamber components.

5.3.iv Vertical End Mill
The vertical end mill is a specialized form of drill press that allows movement of the bit in three dimensional space. The bits used are also specialized in that they can plunge like a normal drill bit or cut sideways like a router. This flexible platform proved useful for all sorts of material removal processes, from opening the evaporator base to rounding out endplates. The operator may also add a rotary table to the mill table, giving the part rotation in one axis in addition to the existing three axes of movement. This precise rotation capability allowed us to perform operations such as the even bolt pattern on the back of the combustion chamber endplates.

5.3.v Optical Plasma Cutter
Arguably the most useful tool, the optical plasma cutter is a conventional plasma cutter with optical line following. Plasma cutters use a high power electric arc combined with a shielding gas to create a high temperature plasma blast. This plasma beam is hot enough to burn through most materials, and focused enough to make neat clean cuts. The hand plasma cutter is ideal for quick, rough cuts on practically any conductive material.

The optical follower uses a lens and carrier to trace dark lines across a light surface. This works optimally with sharpie lines on white paper. By tracing out parts and fittings with a sharpie, we can use the optical plasma cutter to make a part of exact dimensions in a matter of seconds. This tool did most of the metal cutting for the project; everything from the turbo flange to endplates to the electrical control panel.

It is worth noting that cutting operations without the plasma cutter are extremely difficult. During the fabrication of the combustion chamber one of the students damaged the plasma cutter. The oxyacetylene torch made for a poor stand-in until it too was damaged. Without these cutting tools, parts had to be slowly ground and hacked into shape. This was a laborious and time consuming process that often took multiple attempts.

5.3.vi Arduino Tool Set
The Arduino microcontroller is programmed in C. The process of compiling, uploading, and executing the code on the microcontroller is performed by the Arduino tool set software. A serial channel provides for immediate feedback from the Arduino during runtime, enabling debugging. The expansive library afforded to the Arduino community make for very easy programming and integration. In particular, the Ethernet library proved essential for easy transmission and receipt of TCP packets over IP.

5.3.vii Microsoft Visual Studio 2010 Ultimate
The software monitoring program was written using C# and XAML. While the code could have been written manually, the Visual Studio IDE proved invaluable in both simplifying and expediting code generation. Automatic code completion, object visualization, and integrated debugging saved a

considerable amount of development time. In addition, the XAML GUI tool made the creation of the visual interface extremely quick. No XAML code had to be written by hand; the visual designer and property managers allowed for full configuration of the interface, leaving only the functional code-behind to be written by hand.

6 Analysis of Design

6.1 Summary of Functional Requirements

Project completion includes the satisfaction of all original project goals. The turbojet engine is capable of idling under its own power, without any external assistance. The engine is stable and not prone to erratic behavior, such as sudden flameouts or dramatic changes in RPM. Onboard computer control makes it easy to start and stop engine operation with minimal effort from the user. Integrated sensors and controls allow for safe and reliable monitoring and control of the engine's operation. The remote software interface allows for even easier visualization and control of the engine. As per Section 1, the original functional requirements of the project have been fully and completely satisfied.

6.2 Primary Constraints

The main difficulties encountered throughout the project are documented in Section 5. Please refer there for full information.

6.3 Economic Impact

6.3.i Capital

The capital associated with this project is substantial. An abnormal number of man hours have been dedicated to the project by its various members. I personally have given over 170 man hours to fabrication time alone. Considering all of the efforts made by all of the team's members, total human capital likely numbers close to or above 1000 man hours.

Financial capital also proved essential to the project's success. The CPConnect foundation donated $805 and the Chevron Technical Foundation donated $800 to the project. Without this financial support, the necessary sensors, hardware, and support equipment would have been unobtainable. Despite the project being 25% over proposed budget, the costs were still covered by the available funding pool.

The manufactured capital consists of the physical project itself as well as its various support equipment. Fortunately, thanks to the high interest in the project and its extreme flexibility towards further applications, this capital will likely prove useful to future Cal Poly students. Already several students have expressed interest in expanding on the engine for their own senior project.

Due to the large size of the project, it also necessitated a large amount of natural capital. Natural resources were required for all of the raw materials, from steel to diesel to the embedded electronics. The project's limited fuel economy and extreme thirst for diesel and motor oil also require a continuous supply of natural capital to sustain its operation.

6.3.ii Financial Summary

Refer to Appendix A for the original budget and bill of materials.

6.3.iii Project Timeline

The project's complete timeline, including milestones and time spans, is documented in Section 5. Please refer there for full information.

6.4 Commercial Manufacturing

This project is a one-off prototype and is not intended for commercial manufacturing.

6.5 Environmental Impact

This project has a considerable environmental impact. The act of fabrication alone required a great amount of power, chemicals, and manufacturing consumables. Shielding gasses, electrodes, grinding wheels, cutting tools, and others were all required consistently throughout the fabrication process.

In addition to the costs of production, the engine also has a poor environmental impact during operation. The old seals and wide tolerances on a turbocharger of such size cause the turbo core to burn and spill oil at a surprising rate. This causes both oil slicks and thick oily smoke during normal operation. The engine also has very poor fuel efficiency and consumes diesel fuel oil at a very quick rate. This fuel is simply burnt and spent as exhaust. Without any form of harnessing the power of combustion, the engine is incapable of producing any physical work as output. This means that all inputs to the system are effectively wasted. Future efforts on the project should focus on harnessing the engine exhaust to perform some form of physical work in order to better justify the demands of operation.

6.6 Manufacturability

The milestones and issues encountered during fabrication are documented in Section 5. Please refer there for full information.

6.7 Sustainability

The project has appreciable maintenance requirements. The evaporator and flame tube assembly is subject to extreme conditions during normal operation. In commercial turbine engines, these components are designed to be consumable. In this project, they must be disassembled and inspected occasionally to watch for degradation. The process of manufacturing replacement components will prove to be very time intensive.

As mentioned in the environmental section, the engine maintains a strong appetite for diesel and oil. This is not sustainable, as no work is harnessed from the system. Replacing the turbocharger seals and adding equipment to harness engine output would greatly improve the sustainability of the engine's normal operation.

This upgrade path is nontrivial. Harnessing work from the system will prove to be a considerable engineering undertaking, requiring input from students of several different disciplines. The mechanical aspects of design, choice and fabrication of materials, and the modification of computer control will require a considerable amount of time and effort. Replacing the turbo seals is a far less complicated

procedure, but will still require considerable time and money. The turbo core is so old and unique that it will take some time to locate an acceptable replacement.

6.8 Ethical

This project poses few ethical implications. It should be noted, however, that it is a dangerous and messy system. The operator is responsible for the safety of bystanders and surrounding objects and equipment, as well as seeing after the mess created by the system. It would be unethical to operate the system against these considerations.

6.9 Health and Safety

This is an industrial project. As such, there are many concerns to health and safety. Diesel fuel and motor oil are hazardous substances and pose a health risk to the operator. The exhaust created by the engine is both dangerous and extremely hot, posing a risk to both the health and safety of the operator and bystanders. In addition, failures during operation pose an extreme safety hazard due to the tremendous amount of energy contained in the system. A sudden flameout or turbine fragmentation could cause serious injury or death to anyone near the engine.

The fabrication of the project also posed considerable risk to health and safety. The machine shop is a dangerous environment, with constant risk to life and limb. Large, powerful machines pose a risk of bodily harm. Various manufacturing chemicals present an array of threats to health. Most of the tools and machinery also produce considerable noise, creating a dangerous risk to hearing. Nearly all of the tools are also subject to the risk of sudden failure. For these reasons, protective clothing, ear and eye protection, and specialized training are required by all students working in the shop. Thanks to these safety measures, only two very minor injuries occurred throughout our fabrication effort.

6.10 Social and Political

This project poses no major social or political concerns.

6.11 Development

The development and manufacturing process is documented in Section 5. Please refer there for full information.

7 Appendix A: Budget and Materials

Combustor

Assembly	Part	Aprox Price	Act Price	Tolerance
Combustion Chamber	8"x4' Pipe	$0.00	$0.00	$0.00
	1/4" Plate Steel	$0.00	$0.00	$0.00
	Mounting Hardware	$25.00	$0.00	$25.00
	Turbocharger	$140.00	$140.00	$0.00
Flame Tube	4"x3' Pipe	$0.00	$0.00	$0.00
Evaporator	1"x6" Pipe	$0.00	$0.00	$0.00
	1/2"x16" Pipe	$0.00	$0.00	$0.00
Standoff	4"x4" Pipe	$0.00	$0.00	$0.00
	1/8" NPT Standoff x4	$9.24	$9.24	$0.00
	1/8" NPT Pipe x2	$3.36	$3.36	$0.00
	Hose barbs	$9.18	$9.18	$0.00
	1/8" 9.49gph Nozzle	$8.27	$8.27	$0.00
	1/8" 1.26gph Nozzle	$5.83	$5.83	$0.00
	1/4" Plate Steel	$0.00	$0.00	$0.00

Support Systems

Assembly	Part	Aprox Price	Act Price	Tolerance
Fuel	Pump	$0.00	$0.00	$0.00
	1/4" Fuel Line	$12.45	$0.00	$12.45
	Fuel Filter	$0.00	$0.00	$0.00
	Needle Valve x2	$39.86	$34.78	$5.08
	Tee Fitting	$7.50	$7.42	$0.08
Oil	Pump	$80.00	$181.53	($101.53)
	Drive Motor	$20.00	$0.00	$20.00
	Filter Adapter	$15.95	$18.90	($2.95)
	Filter	$5.06	$9.90	($4.84)
	Cooler	$0.00	$56.00	($56.00)
	Tubing	$20.00	$17.68	$2.32
	4"x6" Pipe	$0.00	$0.00	$0.00
	10W-20 Oil x5	$31.99	$29.08	$2.91
	Hose Clamps	$0.00	$11.40	($11.40)
Ignition	Coil	$22.79	$22.20	$0.59
	Spark Plug	$10.00	$1.82	$8.18
	Wire	$10.00	$3.85	$6.15
	Battery	$80.00	$0.00	$80.00
Frame	1" Square Stock	$0.00	$0.00	$0.00
	Diamond Plate	$0.00	$0.00	$0.00

Electronic

Assembly	Part	Aprox Price	Act Price	Tolerance
Computer	Arduino	$25.00	$0.00	$25.00
	WiFi	$55.00	$60.14	($5.14)
	LCD Display	$17.95	$0.00	$17.95
Sensors	Combustor Pressure	$109.42	$91.44	$17.98
	RPM	$105.00	$128.45	($23.45)
	EGT	$51.75	$51.75	$0.00
	Flame Sensor	$20.00	$12.95	$7.05
	Oil Temperature	$39.00	$37.76	$1.24
Controls	Fuel Relay	$5.61	$8.11	($2.50)
	Oil Relay	$5.61	$8.11	($2.50)
	Primary Servo	$13.95	$13.95	$0.00
	Ignition Servo	$13.95	$13.95	$0.00

Over Budget

Assembly	Part	Aprox Price	Act Price	Tolerance
Hardware	Various	$0.00	$178.73	($178.73)
Wiring	Various	$0.00	$57.92	($57.92)
Misc	Gasket Material	$0.00	$12.60	($12.60)
	Compressor Duct	$0.00	$10.00	($10.00)
	Oil Cooler Fans	$0.00	$15.20	($15.20)
	Thermocouple Amp	$0.00	$2.19	($2.19)

Proposed Budget	$1,018.72	$1,273.69
		$254.97