Calvin Nelson

Ryan Kachline

Team Flux Capacitors

EE403W Spring 2016

Project Statement

Device Description

Our group has decided to create an Automobile Engine Ignition Timing Delay Box. The purpose of this box will be to adjust the ignition advance as requested by the user. As a proof of concept, the device will delay the ignition timing by a set amount, when a push button is pressed. Expansion goals will be added as needed if time permits (see section on expansion goals for more details).

The device will follow the following algorithm:

1.      Check push button state

2.      Perform calculation for requested delay.

3.      Wait for ignition event.

4.      Delay for requested number of degrees.

5.      Send ignition event.

6.      Repeat.

A strategy that will be employed is using the down time between ignition events to calculate the following ignition delay. By doing this, the speed of the processor will not be a huge priority (See calculations section for more details).

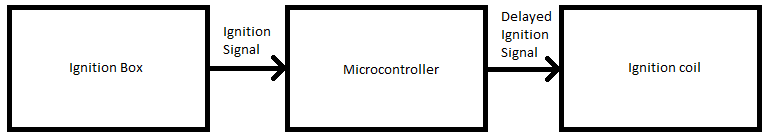


Figure 1: Block Diagram

Our group will be using the Arduino Uno for the following reasons:

1. It is fast enough for our application, and can provide an accurate delay on the order of microseconds.
2. It is relatively cheap.
3. We already own one.
4. Has required number of digital I/O ports for up to an 8 cylinder 4 stroke engine.

We will use the Arduino Software IDE for all development purposes. We chose this because we know that it is compatible with the microcontroller. The language that the code is written in is C, thus it uses a variant of a C compiler to execute and load the code.

Our current plan is to use the PCB for filtering noise. A car has a sizable amount of noise, so this task is necessary for a stable power supply.

Justification

This device will be used on older cars that do not have the ability to electronically adjust their ignition timing. On an older car, all of the ignition timing alterations are done mechanically. They mainly operate in response to intake plenum pressure changes, or engine speed changes. This device would allow the ignition timing to be further modified in response to up to 6 analog electrical sensors.

This is useful, in cases when you install power adders, such as a supercharger(s), turbocharger(s), and/or nitrous to a car. In these cases, the thermal expansion rate of the combustion charge is increased. At low RPM when the engine does not feel the effects of a power adder, you want the ignition timing to act like a stock car. Alternatively, at higher RPM and/or when the power adder becomes active, you want the ignition timing to take into account this increased rate of thermal expansion.

This can be accomplished a number of ways, one example being factoring in the incoming air temperature. As the incoming air temperature increases, so does the rate of thermal expansion. Also, as a safeguard to the engine, devices called knock sensors can be added to the equation as well. Knock is when the ignition timing happens early enough that the maximum force applied by the combustion charge happens before the piston reaches the top of its compression stroke. This can be extremely harmful to an engine, and prevention of it is paramount. Clearly this device is useful in the cases explored above.

Preliminary Design Requirements

The device will:

* Apply an ignition delay with a minimum step value of 0.5 degrees for a maximum operating speed of 6000 revolutions per minute.
* Use an Arduino Uno, and involve the design and implementation of a printed circuit board.
* Be able to function in an automobile using automobile power supply.

Calculations

Assuming that the engine will have a maximum operating speed of 6000 revolution per minute, and the engine will be a 4 stroke, 8 cylinder engine, our worst case time between ignition events will be:

This means that the microcontroller in the Arduino will have the following number of cycles to calculate the following ignition delay:

Assuming that the minimum step amount of angular ignition delay is 0.5 degrees, the minimum step length at 6000 revolution per minute, will be:

The Arduino is accurate for any delay above 3 microseconds. Therefore it will suit our needs.

Expansion Goals

Some expansion goals we will have if time permits are:

* Increase the maximum revolution per minute limit.
* Add analog sensor input for linear transfer curve capabilities.
* Add engine speed limiting capabilities.
* Add knock sensor timing retard capabilities.