ENT SENIOR DESIGN PROJECT REPORT

Two-Stroke Ignition Redesign for Yamaha KT100 Engine

Submitted to

Professor Robert Weissbach
Engineering Technology Department IUPUI
by
Corey Schoene & Weigang Wang

April 29, 2019

Ignition Redesign

ABSTRACT

This report details research and development put into the ignition controller in place for a two-stroke racing engine. The project involved reverse engineering and documenting a transistor-controlled ignition (TCI) module currently used on the Yamaha KT100 engine. A theory of operation as well as design details of the original unit are discussed. Using the knowledge of the engine, the unit is completely rebuilt to allow a programmable spark advance curve to be implemented. The new unit is a direct drop in replacement on the engine. The theory of operation and design details of the new unit are discussed extensively. The new unit adds value to the engine as it allows the user to tune the engine to gain performance under certain conditions.

TABLE OF CONTENTS

REVISION HISTORY

Version	Date	Revised by	Description
1.0	24 April 2019	Corey Schoene	Initial Draft
1.1	27 April 2019	Weigang Wang	First Revision
2.0	29 April 2019	Corey Schoene	Final Version

Ignition Redesign

1. INTRODUCTION

The purpose of the ignition system is to fire the spark plug at a very precise time. Depending on the speed of the engine, this event can occur up to 250 times per second. This project was an investigation into the stock transistor-controlled ignition (TCI) module to find how it worked and verify its output. After this was done, a new system using a microcontroller and custom hardware was prototyped. The new system allows the unit to be software controlled to deliver a spark at specified piston positions depending upon speed. The new system is a drop in replacement module meaning that that the micro controller power, engine speed, engine positon, and spark control must all be accomplished through a single wire and a grounded outer case.

1.1 Problem Statement The original TCI must be well understood as too provide an accurate depiction for its purpose. The limitations of the system can be circumvented by creating a new system that allows tuning of the spark advance curve. This system must not alter the engine in any way. This new system allows a customizable ignition map to be implemented leading to customizable engine performance in any desired portion of the RPM band.

1.2 System Overview This project was provided to us by Christopher Finch while the complete engine was provided to us by John Copeland. The engines intended use is strictly for racing go-karts. The new system would allow the user to set a timing curve that would give the best performance for each individual track allowing a single module to be reconfigured for each track as well as other factors such as air pressure and fuel grade. The original ignition system automatic advance is at 5,000 RPM (28 degree) and 10,000 RPM (22.5 degree) [1]. The new system at its essence allows a lookup table with engine RPM as its input and degrees before top dead center to as its output. The user can set any real function that has one and only one angle output for each speed input as long as the input range stays between 0-15,000 rpm and the output angle stays between 0-36 degrees before top dead center. A list of the required specifications to build to can be found in the referenced documents [2].

2. REFERENCED DOCUMENTS

Table 1: Referenced Documents

Title	Comment
Two-Stroke Engines by Harry Senn	Chapter 8 in book [1], [3]
System Test Results	In Draft [2]

3. SYSTEM-WIDE DESIGN DECISIONS

This section will discuss the design reasoning behind the high-level sections used on the final product. A simplified high-level diagram of the new system can be seen below in figure A in the appendices section. The system interfaces to the engine through a single wire. Hardware is used to condition the signal, gather power, and control the spark. The conditioned signal from hardware is given to the software in order to make the correct timing decision. Software then controls the hardware switch section in order to create a spark in turn controlling the engine.

3.1 Stock Hardware The original hardware must have been understood in order to continue. After disassembly and reverse engineering, the schematic as seen in figure B can be seen from the original TCI. A detailed theory of operation will be discussed below. The system as a whole works by shorting out the stator coil and releasing the short. Since the stator is inductive, the sudden disruption in current flow will cause the voltage to suddenly increase to the point that it causes a spark to jump across the electrodes of the spark plug. The stock engine hardware (stator and rotor) unloaded output can be seen below in figure C. This is same signal being used on the final system so understanding how to use it and how the spark is created is critical.

3.2 Hardware The new system fits within the original TCI case that measures approximately ½” × ½” × ¾” internal volume. The system is required to be in a metal case as the case acts as a conductor to the chassis that it is bolted onto. Since the engine must not be modified in any way other than a drop-in replacement for the TCI, the new system interfaces with the original stator through a single wire coming from a tap off the primary of the internal coil. To assist in understanding of the next portion, figure C in appendices may need to be referred too. This figure depicts the output from the stator while under no load. The hardware inside of the new TCI box is used to gather and regulator power for the micro controller, gather frequency of zero crossings for speed detection, use the time after zero crossings to determine position, and control a switch to create the spark in the negative going section. It should be noted that both the amplitude and frequency linearly scale with engine speed making the system predictable. The final version of the hardware board can be seen in figure D below. The board is shown next to TO-220 package transistor of which some are used in the original TCI.

3.3 Software The software must be very fast as an accuracy angle of +/- 1 degree was desired. This means for example that if the target was set to 20 degrees, the actual angle of spark must be no more or less than 21 and 19 degrees respectively. To ensure this, simple math can be used to show how often sampling of position should occur. Assuming highest speed at 15,000 RPM or 250 rotations per second and a desired accuracy of 1 degree in the 360-degree rotation, a sample speed of at least 360*250 or 90 KHz is needed. The software cannot directly interface with the engine but must use the hardware as an intermediary. The software receives a pulse from hardware during the negative going voltages from the stator. The software also outputs a pulse to the hardware ignition switch based after a specific amount of time after every pulse received.

This delay between these two is determined by a user set curve that works off of the engine speed.

3.4 Interface The user interfaces with the device in two possible ways. The first way is in the initial setup where it is simply plugged into the stator on the engine and bolted down. As it shares the same case, it is a drop-in replacement. The second interfacing with the system is in the form of using the desired curve. A combination of dip switch settings can be selected to select the best curve for the desired application. The system is fully capable of having a complete custom curve uploaded into it but a more user-friendly way of doing so is currently being explored and is unavailable at this time.

4. SYSTEM ARCHITECTURAL DESIGN

This section will explain the low-level design of the new system in detail. The schematics and other figures will be heavily referenced below.

4.1 System Components As stated above, the system is based around a completely stock engine, custom hardware to interface with it, and custom software in order to control the hardware and in turn control the engine.

4.1.1 Stock Hardware Low-Level The original system was reverse engineered and the schematic that part numbers will be referred to is in figure B. Q4 is the main current carrying transistor. It is initially turned on by SCR Q5 by a voltage reaching a threshold set by R6 and R5. Once Q4 has been turned on, current will start to flow through R7. Once the current through R7 is sufficient enough that the voltage drop across it can turn on Q6, this will force off Q4. Since the SCR must be "power cycled" ensuring that retriggering within the same cycle does not occur. It can be seen from this that there is no setting of the advance curve other than the natural characteristic of the circuit. The only thing that can alter the timing is the field strength provided to the stator. This only offset the curve and does not allow reshaping. This field strength is determined by the rotor to stator gap. Generally, a gap of 0.020 inches between rotor and stator gives the best results [3]. The curve from the stock TCI found by experimentation can be seen in figure E.

4.1.2 Hardware Low-Level The schematic for the hardware used in the new system can be seen in figure F below. All following part values called out for the remainder of this section will refer to this figure. The hardware can be split up into 3 main sections to help with readability. The first section is the power section which can be seen as components D3, R2, D4, C1, C4, U1, C5, and C3. The signal conditioning section which be components R11, R12, C7, U3, R10, D1, R9, and R4. The final section is the switching section, which is made up of components C10, U6, U4, U2, U5, C6, C8, C9, R1, R3, Q1, Q2, and D6. All parts either provide data into or receive data from the microcontroller. The main microcontroller that runs the system is based around a TI C2000 micro. Specifically, the TMS320F28027F. This micro controller is in a LQFP-48 package running at 60MHz. Only 2 GPIO are used.

The power supply sections by gathering power off the positive going peaks from figure C. This is done with D3 only allowing current to flow in the positive peaks. This charges C1 up to a voltage limited to the Zener voltage of D4. The current flow to this clamping diode is limited by R2. The values have been calculated to allow enough current to flow to the system without at all speed ranges without excessive heating. The regulator U1 then regulates the voltage down to a stable 3.3V that is required by the microcontroller. The capacitors C4 and C5 are used to prevent high frequencies of the regulator. Capacitor C3 is used for bulk storage on the low voltage side.

The signal conditioning section functions by allowing a positive going pulse at an acceptable voltage level for the microcontroller. This stays high while the analog input voltage as seen in figure C is negative. The circuit works by grabbing the input voltage and feeding into the positive input of the comparator U3. The signal is passed through R10 to limit current to a low value. This input is clamped by D1 to protect the comparator. R9 and R10 work together to add some hysteresis to the circuit so that noise at the output is reduced. R4 acts as a pull up resistor for the open collector output of the comparator. R11 and R12 acts a voltage divider to give a voltage for comparison. The voltage is barely above ground so that it the reference is lifted above the noise floor. C7 is simply a decoupling capacitor. This area could use improvement which is placed below in the recommendation section.

The switch section is the most complex due to the need for an extremely fast switching edge. This portion works by receiving a signal for switching and passing it to U6 which inverts the logic signal. Both the inverted and un-inverted signals are then passed to the gate driver U2. The gate driver allows a high current for both MOSFETs to flow while having both share their own independent reference while also being separate from the drive signal ground. Each gate needs its own supply at a different reference level therefore U4 and U5 are needed. They have a wide input range with a 12V output. C8 and C9 act as local capacitance for the gate driver. Q1 and Q2 are an N-channel and P-channel MOSFET respectively. These MOSFETs being back to back like this make sure that none of the internal freewheeling diodes are able to conduct so long as the complimentary MOSFET is off regardless of the sign of the voltage. R1 and R3 reduce the current flowing into the gates firing switching but also insure that the gate as seen by the driver is most resistive rather than inductive due to the traces ensuring no ringing. When the gate driver's input is high. The corresponding channels output ties to its VDD. When the input goes low, the corresponding VSS gets tied to the output. By the way the circuit is configured, bringing the P-channel MOSFET driver input low, turns it on while bringing the N-channel MOSFET driver high turns it on. This explains why the inverter is needed as complimentary states are needed for each MOSFET. For example: if AB as seen by the gate driver is 10, the switch is off, if AB is 01, the switch is on. D6 is a TVS diode used to protect the MOSFETs during inductive kickbacks that could overshoot the rated voltage of the switch.

4.1.3 Software Low-Level The software portion as seen in figure G below shows a flow of the software. The software functions by gathering the pulse waveform coming from the hardware through a GPIO that must be sampled at a minimum of 90 KHz as found above. The number of rising edges that occur is counted by a counter. This counter is reset and its data is passed every gating period. In Figure G this gating period is shown to be 1/10 of a second. The number of pulses found increases with engine speed linearly and can be used to find engine speed. A longer gating period allows more speed resolution but decreases dynamic engine responsiveness. 1/10 was found to be a good compromise. The speed is passed to a lookup table that then translate to a delay of switch turn off. When the pulse just starts (analog voltage just goes negative) the SR latch is set which turns on the switch and current starts to flow. After the delay has been reached, the SR latch is reset and the switch is shut off. The lookup table can be set to customize this release time. The blue section in the code is simply an engine output simulation block to allow for simulation testing.

4.2 Concept of Execution The complete system must work together at a relatively high speed to ensure smooth running of the system. The system begins with the original rotor and stator of the engine which deliver the waveform as seen in figure C below. As the engine rotates, it moves a rotor with embedded magnets past a stator inducing a voltage. The hardware in the new board uses the positive portions of this wave to capture power all hardware requiring power to operate. The frequency of zero crossings from the hardware conditioned signal is used in software to determine the speed of the engine and the timing to currently use. This timing signal is determined from a delay after the zero crossing. When software determines the proper time to open and close the switch based off of the lookup table, it is passed to a gate driver in the hardware. The gate driver will short out the negative voltage immediately after the zero crossing. The current will rise up until the switch is turned off. When this switch goes off, the stator will want to maintain the current flow due it being a large inductor. The voltage will rapidly rise up to the point that the spark plug will have an arc inside of the engine causing the ignition for that stroke to take place. This developed power will spin the engine and cause the process to repeat. Depending on the fuel, air, and load placed on the engine, the system will either speed up or slow down as needed. This will in turn change the speed and angle as detected in software.

4.3 Interface design The system was designed to directly interface with the existing engine without altering it in any way. Therefore, the system was designed to gather all needed information, power, and control through the one original wire without the need to add any additional sensors or power supplies. In order to operate correctly, control the spark on the negative going portion of voltage. The main reason for this is because this is where the highest power capability is as well as this portion being in the desired engine positon for a spark. The system was also set up to be this way as the rotor ad stator are keyed so using zero crossings for timing will be accurate regardless of rotor gap or field strength. No user interfacing is needed during run time.

4.3.1 User Setup and Operation The unit is designed to be a drop-in replacement for a stock TCI box. The first step to using the new system is to remove the old TCI box. This can be done by removing the two bolts holding it on. Once unbolted simply pull on the plug to unplug it. The new box should be tightly bolted back down where the old one once sat. Once tight, plug new wire into the stator wire. The new system is operated the exact same as the old system. When the user places a starter up to the engine and starts spinning it, the system will need approximately 10 complete rotations at a minimum speed of 500 rpm in order to boot up and start creating sparks. This delay is usually completely unnoticeable as it takes many more cycles to get the engine primed and proper flow through the fuel lines. The way to change the timing curve is currently under development. Once implemented, the unit will have small dip switches to select from multiple pre-defined curves very quickly and easily. In order to create a custom curve, it must be plotted into Excel. Excel will then give numbers that need to be flashed into the micro controller through USB. Once finalized, a more detail walkthrough is to be provided.

5. CONCLUSIONS AND RECOMMENDATIONS

The final outcome of the project and added value will be discussed. Further building on the project and revisions are recommended to ensure ruggedness and allow easier tuning of the system.

5.1 Conclusion The final system adds value to the engine as it leaves it unaltered in every way but allows it to be tuned as needed. This can allow a user to set the engine up to perform better on specific types of courses, tune the engine for better power, better fuel efficiency, or compensate for environmental or fuel grade changes. The original system must have been understood in the first place to effectively use the original components. The reverse engineering and plotting of the new curve were provided in order to aid understanding of the engine. The goals of the project are mostly met except the accuracy which is not in control within the scope of the project. These can be seen on the referenced engineering requirements specs document.

5.1 Recommendations If to continue with the project there are some changes and additions that should be made in order to increase ruggedness and user friendliness of the system. The first change to be made should be on the signal conditioning portion of the hardware. To increases accuracy, the comparator should be configured to compare to an actual negative value. In order for this to occur, a negative rail must be generated in the hardware. This can either be accomplished by a small charge pump or a reverse biased diode and regulator (Zener and capacitor). Note that the comparator must be powered from this negative rail as well as to ensure the inputs do not exceed the rails. This will ensure that the comparator is only given a pulse out during a true negative peak.

Ignition Redesign

The next improvement to be made is in the custom timing graph entering mode. To improve user friendliness, the desired graph should be automatically converted into a file to upload into the microcontroller. This will allow a user to graph the exact response that want in terms of RPM vs degrees BTDC. This will allow users with mostly engine knowledge and little programming knowledge to quickly and easily place new response curves into the system.

The final issue to improve may not be applicable depending on the final engine application. Through testing, it was found that the portion of the shaft that hold the rotor wobbles due to engine vibration and weight not being balanced. The changing clearance caused anomalies on the stock timing curve due to the changing field strength. The changing curve does not affect the final system as much but there is still a small effect due to a torsion applied to the shaft changing the zero crossing points. The only solution to this would be to harden the shaft to reduce flexing. Predicting the wobble may be possible but not obvious resonant patterns were observed and no other reference inputs are available with using external sensors.

Ignition Redesign

NOTES

Two stroke engines work by pulling in the fuel and air mixture into the crankcase through intake port which is a reed valve. As the piston comes down, this area becomes slightly compressed. Once the piston goes far enough a port on the sides are exposed allowing the exhaust to exit while the slightly pressurized fuel and air mixture can come in and replace it. Once the piston goes back up and closes the ports off, it will reach the top of its stroke and the plug will fire pushing the piston down and repeating the cycle. The time at which the spark plug fires is critical as it determines the developed power. Firing too earlier will cause the engine to ping and knock by trying to create power in the opposing direction of engine rotation while firing too late will not allow the power stroke to last as long before the exhaust port bleeds out the pressure. The timing must be dynamic as the flame front before full combustion will move at the same speed regardless of engine speed while piston speed will increase as engine speed increases. This requires the spark to happening earlier and earlier as the engine speeds up. Other factors such as air temp, fuel octane, and engine load can alter the optimal angle of firing.

Terms used in this paper:

BTDC – Before top dead center
MOSFET – Metal oxide semiconductor field effect transistor
RPM – Revolution per minute
SCR – Silicon-controlled rectifier
TCI – Transistor-controlled ignition
TO-220 – Transistor outline number two hundred and twenty
TVS – Transient voltage suppression

Ignition Redesign

REFERENCES

Material Type	In-Text Citation	Works Cited
Book in Print	[1] [3]	Senn, Harry. “The Ignition System,” in Two-Stroke Engines, 1st ed. Tinley Park, IL, The Goodheart-Willcox Company, Inc., 2018, Chapter 8, pp. 40–43.

Ignition Redesign

APPENDIXES

Qualification Testing

Engineering Requirements	Compliance Determined	Accept/Reject Criteria	Test Required?
The original TCI system must be reverse engineered.	**Completed** A Theory of operation was created and validated.	Pass the requirements of Senior Design Project	Yes
The original ignition system must be plotted out to find the spark curve.	**Completed** The Stock engine was running and timing was monitored throughout engine speed.	Pass the requirements of Senior Design Project	Yes
The new system must be powered by original magnetism source.	**Pass** The new system uses the original magneto system to gather power, speed, and control spark.	Pass the requirements of Senior Design Project	Yes
The new TCI box to be designed must fit in the same foot print as the old one.	**Pass** The new components selected as well as the new circuit board all fit within the original case.	Pass the requirements of Senior Design Project	Yes
The new TCI must maintain spark angle control accuracy to +/-1%.	**Fail** The spark angle cannot be maintained within this range due to the unhardened shaft holding the rotor.	Reject	Yes
Must endure harsh environments (high temp and vibration).	**Pass** All components are rated to at least 105 degrees C. MLCC have been avoided to improve vibration withstand	Pass the requirements of Senior Design Project	Yes

Ignition Redesign

FIGURE

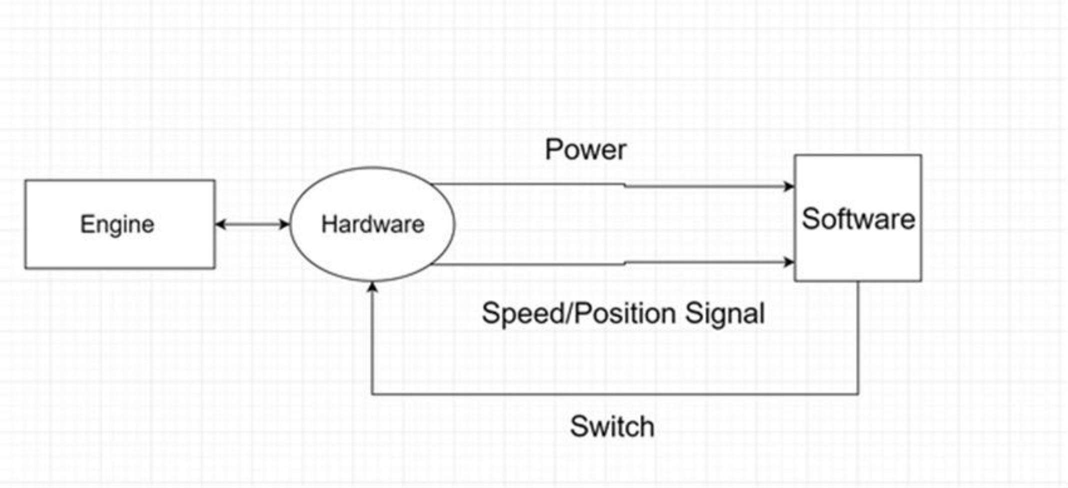

Figure A: High-Level System Overview

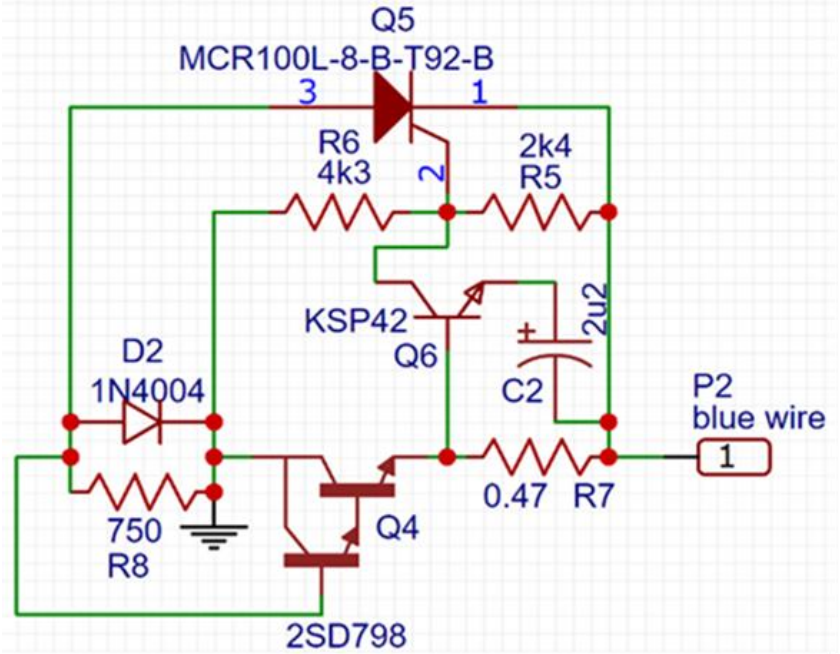

Figure B: Original TCI Schematic

Ignition Redesign

Figure C: Unloaded Stator

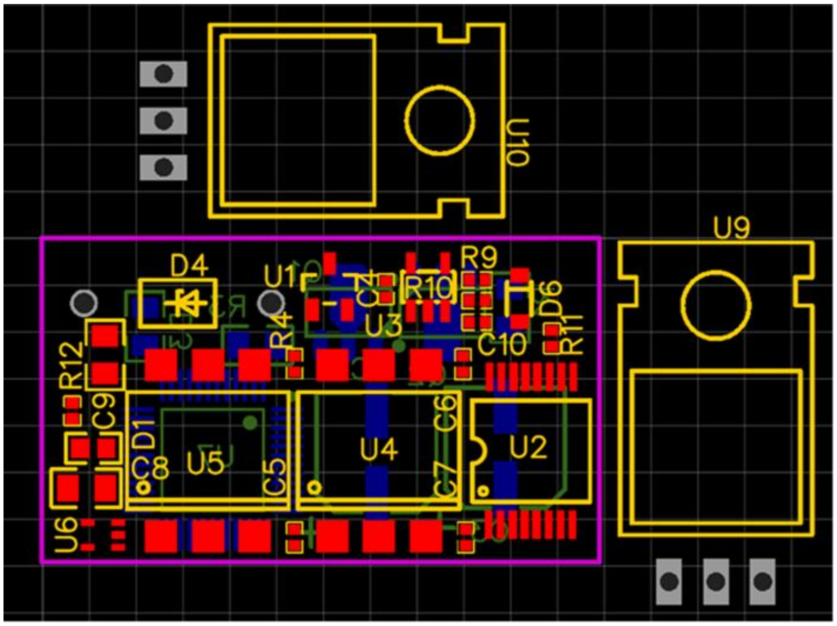

Figure D: New System Circuit Board (only components shown)

Ignition Redesign

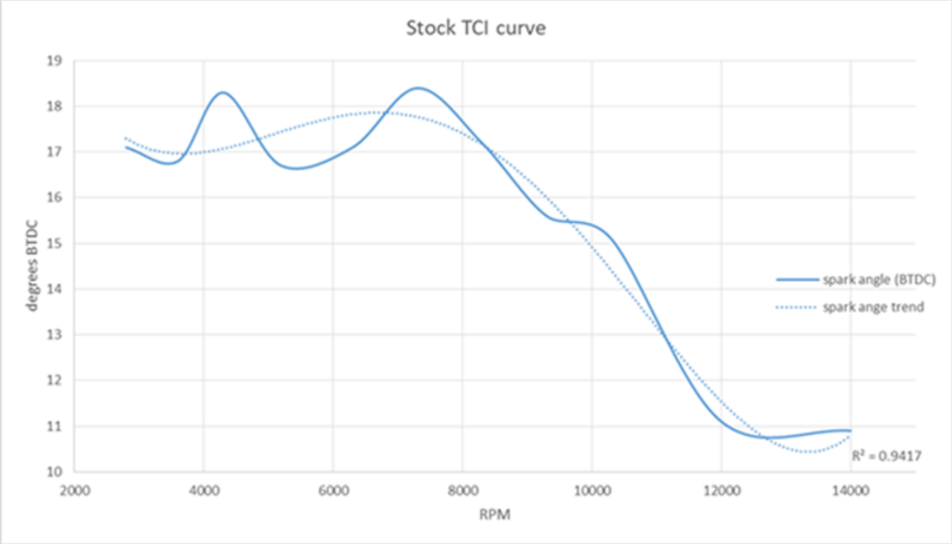

Figure E: Original TCI Advance Curve

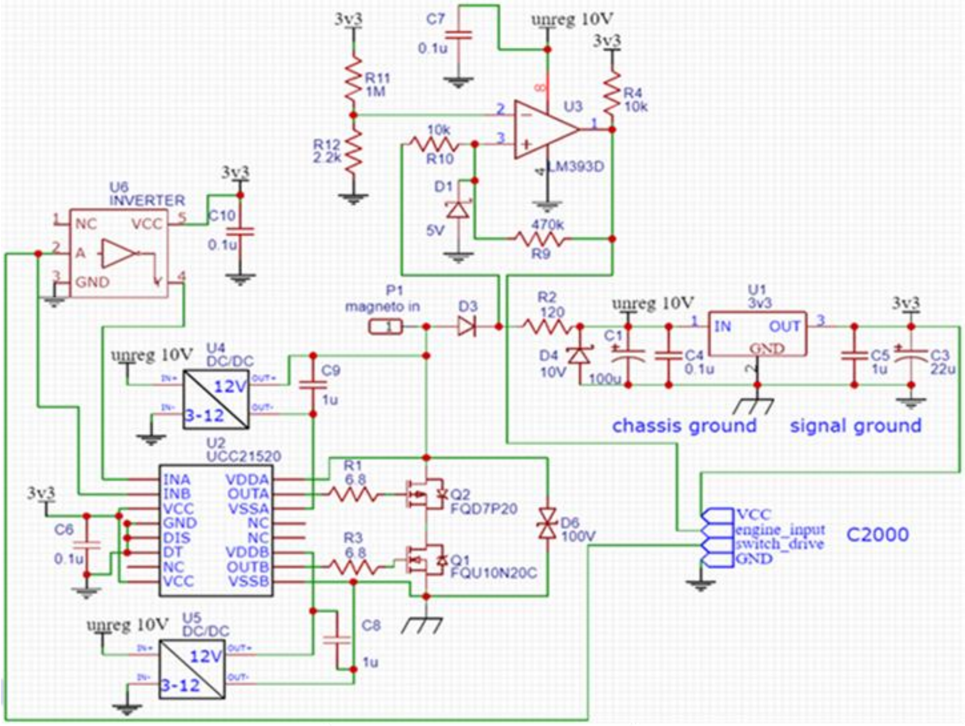

Figure F: New System

Ignition Redesign

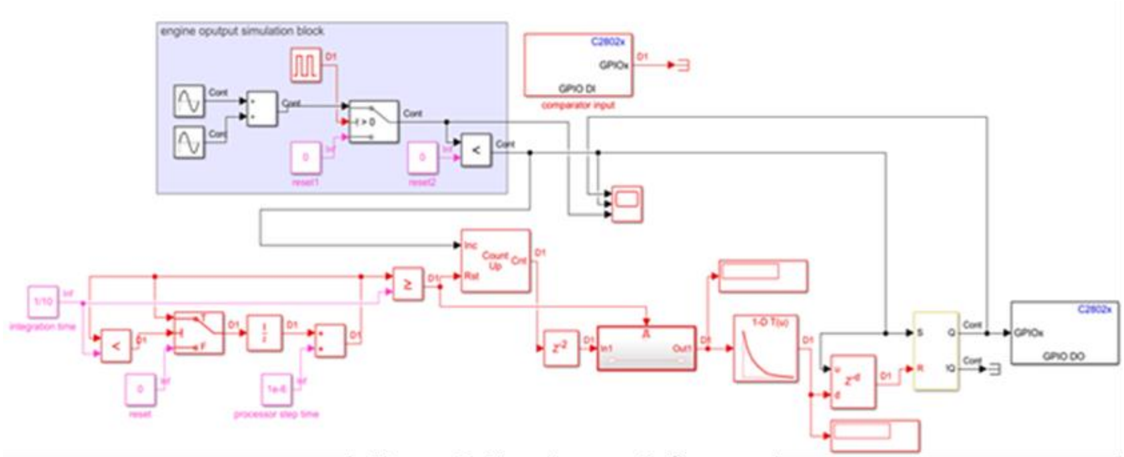

Figure G: New System Software