# Virtual engine management simulator for educational purposes

**R Drosescu**

Technical University “Gh. Asachi” of Iasi, Romania

Radu.drosescu@tuiasi.ro

**Abstract**. *This simulator was conceived as a software program capable of generating complex control signals, identical to those in the electronic management systems of modern spark ignition or diesel engines. Speed in rpm and engine load percentage defined by throttle opening angle represent the input variables in the simulation program and are graphically entered by two-meter instruments from the simulator central block diagram. The output signals are divided into four categories: synchronization and position of each cylinder, spark pulses for spark ignition engines, injection pulses and, signals for generating the knock window for each cylinder in the case of a spark ignition engine. The simulation program runs in real-time so each signal evolution reflects the real behavior on a physically thermal engine. In this way, the generated signals (ignition or injection pulses) can be used with additionally drivers to control an engine on the test bench.*

## 1. Introduction

During the last decade, because of the increasingly restrictive normative requirements which establish gas emissions and fuel consumption, the electronic management systems of internal combustion engines have become increasingly complex, the number and diversity of sensors has increased considerably, and the control strategies and algorithms more and more sophisticated and difficult to test and to calibrate.
First goal of the simulator described in this paper is for teaching purposes, and it is therefore compact and easy to program and use, and it allows simulating, controlling and monitoring the main input and control parameters of modern internal combustion engines with spark ignition (with port and direct injection) or diesel engines with direct injection. Virtual engine management simulator (VEMS) is not a Hardware -in-the-Loop (HIL) equipment and makes the presence of the engine, sensors or electronic control units (ECU) unnecessary. The simulation process is going on until the control signals are generated for an engine operating mode, without considering the engine responses (torque, fuel consumption, emission, etc.) to these commands.

The main input parameters of the simulator are the rotational speed pulses and synchronization pulses with the crankshaft, which in real engines are generated by inductive sensors for crank wheels and by hall sensors for camshaft pulse wheels. Another input parameter simulates the throttle valve angle in accordance with the expected output torque.

The control outputs are generated in a synchronous manner with the rotational speed and position pulses of the crankshaft, just like in a real engine in static or transient operation (rotational speed, load), in which both the rotational speed and the load change according to circular motion kinematic relationships.

For a Spark Ignition Engines (SI) outputs are represented by ignition, injection and knock tracking windows pulses, while in the case of Compression Ignition Engines (CI) only Start Injection Angle and injection impulses are considered. If the simulator works properly, the injection and ignition pulses generated by the equipment are identical from the viewpoint of their form and duration with the ones from the electronic control unit of the engine in stable operation, and they may be in their turn used directly to actuate the engine. The basic condition for a good functioning of the simulator is given by a perfect synchronization between input values defining the operating cycle of the engine and output (controls) pulses generated throughout the entire engine range of rotational speeds, just as in case of real engine operation. Therefore, the simulator cannot operate on a PC running the Windows operating system, whereas SDK Android or IOS systems may can prove to be slow to simulate situations in which the rotational speed reaches 6000-9000 rpm.

The system described here is also useful for studying various control strategies applied to internal combustion engines, for instance the generation of multiple injection or ignition pulses per work cycle, of knock analysis algorithms or miss fires. Considering that any other equipment involved in engine management will work synchronously with each engine cycle, the simulator may be equipped with hardware devices and software programs capable to generating other output signals, such as electronic throttle control, fuel-supply control valve for a high-pressure pump in direct injection systems, control for wastegate or bypass valves in the turbocharging systems and controls for variable valve distribution actuating devices. Such complex hardware/software system are present in [2] and [4].

## 2. Simulator Input Signals

The main function of any internal combustion engine management system consists of the analysis of the signals from the rotational speed and crankshaft position sensors, to find synchronization between every engine cylinder and the crankshaft angular positions during each revolution, and finally to command the generation of injection or ignition pulses, to ensure a proper dosing fuel quantity on each injection pulse. Modern direct injection engines, in which multiple injection events per cycle are generated, require the extreme precision synchronization, below 0.1° crankshaft rotation accuracy. Considering that the duration of the engine cycle (720°) may reach 12 ms, and sometimes even less the simulator's program should loop at 600 kHz in order to attain a 0.1° resolution for a 10000 rpm rotational speed.

The signals generated by these two sensors are of extreme importance, since any malfunction of either of them makes it impossible for the engine management system to work properly. Therefore, modern electronic injection systems also generate an internal synchronization signal, which allows the engine to operate in the limp-home mode and allows the driver to drive his car to the nearest car service shop.

The synchronization signals (from the crankshaft and the camshaft) can come directly from the transducers placed on the test bench, but they may also be generated by the simulator. There is an additional option, in which the simulator generates analogical rotational speed signals like those emitted by a variable reluctance transducer used in almost all IC engines to measure rotational speed. Thus, the simulator is equipped with digital/analogical convertors (DAC) and the signal source for them are represented by Excel files containing samples acquired directly from the two sensors mounted on the engine or from the test bench. In this case, we used a NI-cRIO 9068 controller and a NI 9263 module as analogical output signal generator (4 channels 16-byte resolution, 100 ks/sec, 4 channels with simultaneous update). When only two DAC channels are used the user manual provides a maximum update rate of 200 kHz. As shown in Fig.1, about 50 samples are used for two alternations to generated signal having the same characteristics as the real one**.** Thus, signals with a minimum duration of 250 μs may be achieved. Supposing that the maximum rotational speed is 9000 rpm (150 Hz), it is possible to generate analog signals (AC) like those coming from wheels or ring with a maximum of 25 teeth using DAC hardware.

A series of acquisition equipment based on reconfigurable input/output FPGA technologies which offers high speed acquisition or update time, compact modules, and easy programming are described in [5] and [6].

Fig.2 shows the most commonly used pulse wheels, the former having 35 teeth with 10°RAM distance between them and a missing tooth in position 36, and the latter having 58 teeth with 6°RAM distance between them and 2 missing teeth in positions 59 and 60. Other configuration for crank teeth wheal and cam wheel as well as about synchronization process with crankshaft revolution are extensively described in [1] and [3].

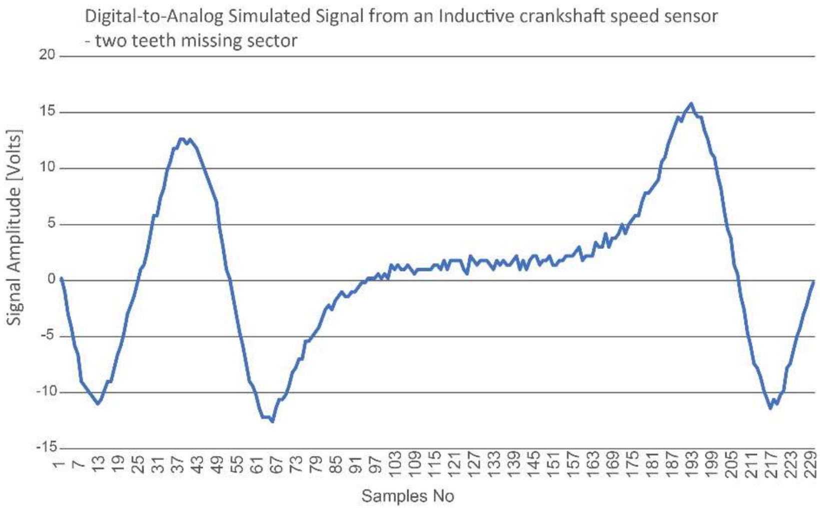


Fig.1 Analog pulse signal generating with a NI 9263 compact configurable DAC module from an Excel file

The missing teeth allow the microcontroller to extract information about the position of crankshaft in relation to engine Top Dead Center (TDC) and help to achieve the synchronization process. To simulate a real transient engine operating cycles that are more difficult to model, it is useful to acquire signals directly from the engine sensors, logging them to excel files and use as source files for DAQ card with high speed (500 000 updates/sec) or with a NI-9236 DAC module but at a resolution of only 25 points for a signal period.

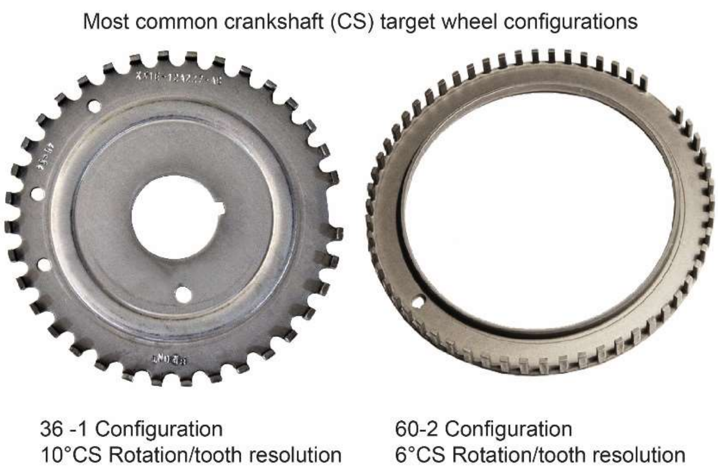


Fig.2 Most common crank wheel configurations

In any engine control ECU, the analogical signal from the crankshaft inductive transducer is converted into a digital signal by means of a comparator, whereas the signal from the camshaft position transducer is sent directly as digital signal due to the use of a Hall-type sensor. Thus, the microcontroller will receive both Crank and Cam signals in digital format. VEMS can receive synchronization signals in digital format in input mode, but it is able to generate its own testing synchronization signals from a multitude of patterns, thus covering all the configurations existing in real engines.

When it is generated by the simulator, the crankshaft synchronization signal allows up to 256 pulses for a rotation, whereas 1, 2 or 3 teeth may be missing. The rarer case in which the missing tooth is replaced by an additional one is also accepted [2]. The position pulse generating gear placed on the camshaft has a multitude of configurations, differing from manufacturer to manufacturer, sometimes even from one model to another of the same manufacturer.

The simulator was tested using several (crankshaft) synchronization and (camshaft) position configurations. In the example below shown in Fig.3 two classical configurations are considered: the firs have 35 synchronization pulses and one missing pulse (36 -1) and the second with 58 synchronization pulses and two missing pulses (60-2). The came wheel have 3, respectively 4-lobes. In the first case, the simulator generates three pulses as Cam positioning signal which correspond to 24, 6, and 6 crankshaft width teeth and three gapes sectors with width of 9, 12, respectively 15 teeth (one tooth corresponds to 10° crankshaft rotation angle). These signals generated by the simulator were recorded by a logical analyser and can be seen in Fig.3 by also with an oscilloscope (see Fig.4).

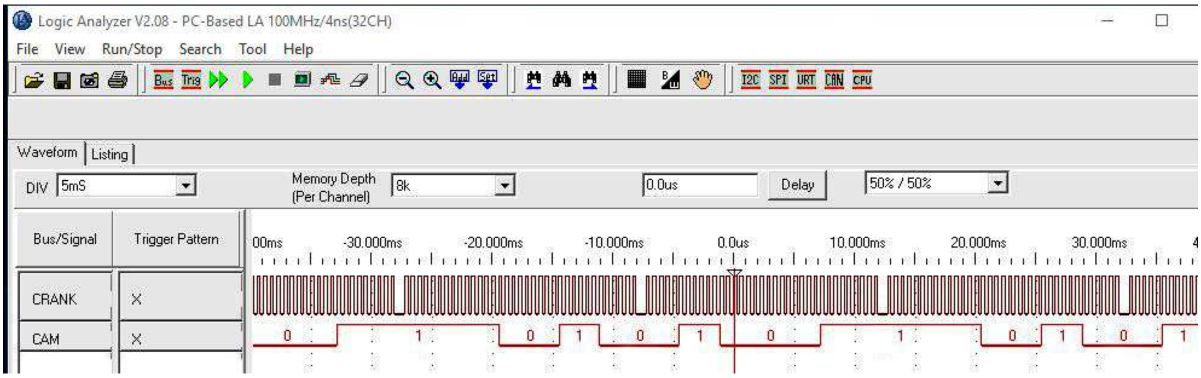


Fig.3 Crank and cam signals generated by simulator similar to those comming from a 36-1 crankshaft wheel (10° resolution) and a 3 lobs cam whel-view from Logic Analizer PLA 2532

In the case of a crankshaft wheel with 58 pulses (60 - 2 configuration) a four-lobe cam has been selected with three equal pulses having a duration corresponding to 6 teeth (36°RAM) and a pulse with a duration corresponding to 24 teeth (144°RAM). The distances between the three equal pulses are 39, 15, 9 and 15 periods corresponding to 6°RAM. The signals generated by the simulator and viewed on the logical analyser are shown in Fig.5.

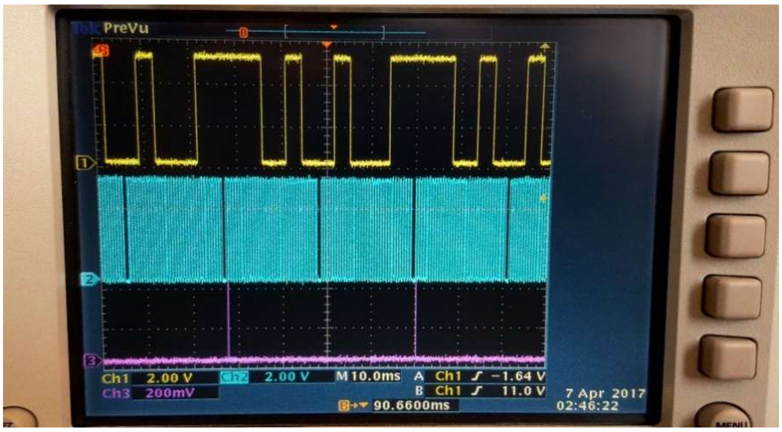


Fig.4 The synchronization signals from the above figure viewed on an oscilloscope

The programming of the pattern of the pulses generating from camshaft wheel especially lob and gape number and width configuration are done in a window of the simulator program starting from a base clock base with programmable frequency. The missing teeth are programmed in a secondary window (removes base pulses), whereas the third window is used for adding teeth programming (adds base pulses). This technique allows to generate any pulses patterns including the case when missing tooth is replaced with an additional tooth, or when a single crank wheel is provided with two sectors with missing tooth/teeth. The graphical programming windows is created using LabView and Veristand development tools.

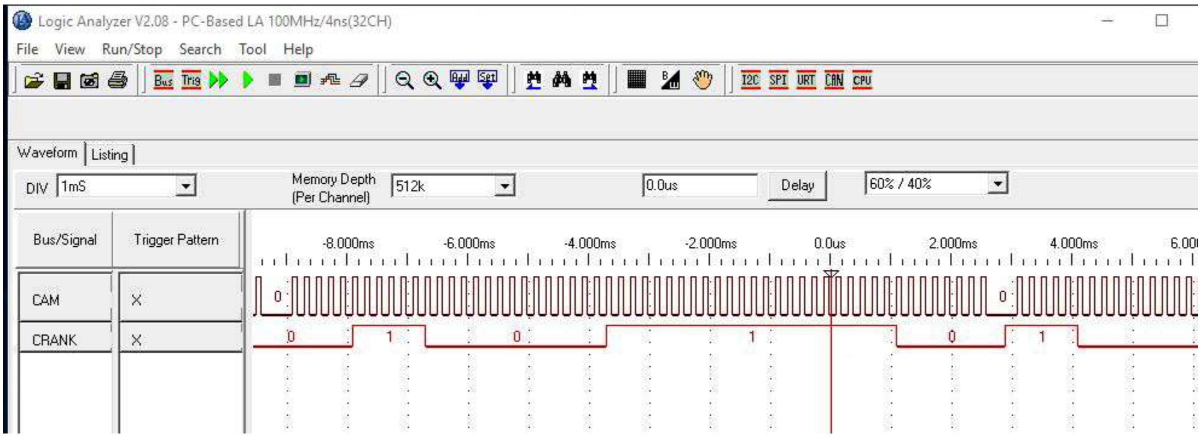


Fig.5 Crank and cam signals generated by simulator similar to those comming from a 60-2 crank wheel (6° crankshaft rotation resolution) and a 3 lobs camshaft whel

Thus, an additional tooth between two teeth of basic pulses is equivalent to a tooth three times thicker, whereas a missing tooth is the equivalent to a gap three times thicker. Fig.6 shows the programming window for the generation of a 36-1 crankshaft wheel pulses, while the same windows is used as in Fig.7 to generate the three lobe camshaft position pulses. A similar configuration program using LabView and Veristand development tools is given in [4] and [5].

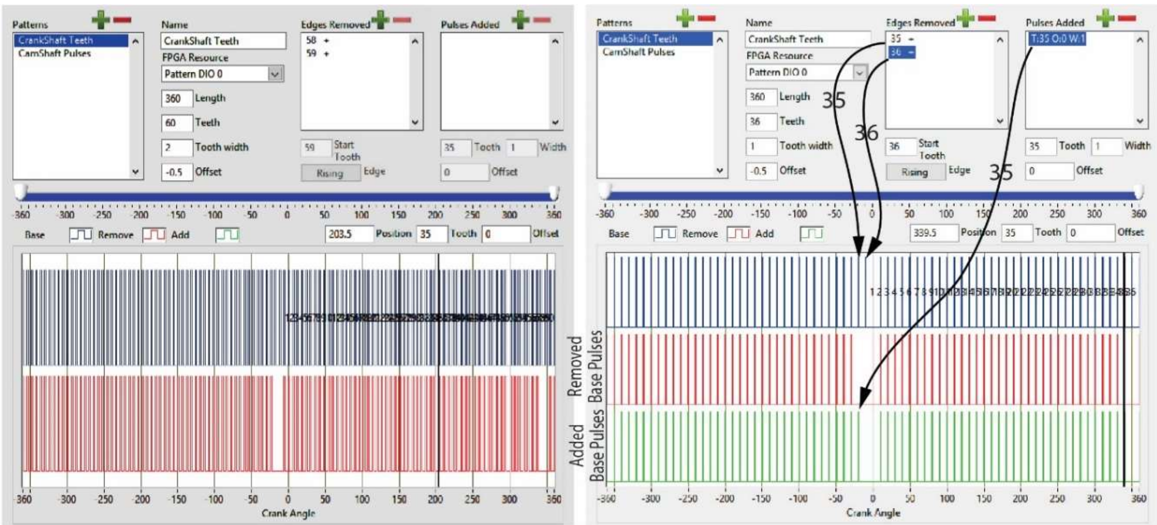


Fig.6 Programming windows for cranckshaft pulse pattern generation

Digital patterns for the two signals can be obtained using a NI-9401 cRIO module or similar PXI DAQ cards. The analog input signals coming from inductive or Hall crankshaft and camshaft sensors can be acquired with simultaneous sample NI-9222 or similar DAQ cards. Any NI-cRIO multiplexed analog input module having a sample rate greater than 250 ks/sec can also be used for crankshaft synchronizing and position signals acquisition. As a real-time controller, a NI 9068 chassis with 8 slots was used.

### 3. VEMS Synchronization process

The most important and complex function of the simulator consists of the internal synchronization of its central kernel both in time and angularly, with the rotation movement of the crankshaft quantified by the input signals from the two sensors: rotational speed (crankshaft) and position (camshaft). All the simulated output (pulses for manifold port or direct cylinder injection, ignition pulses, knock validation windows) must be perfectly synchronized with the angular position of the crankshaft during the whole 720° cycle, i.e. when the signals are sent by simulator to a real engine, it should "consider" them as coming from its own ECU. In addition to the crankshaft speed and position signals, there are other engine input signals the variation of which is synchronous with the rotation of the engine, namely: the mass air flow, the manifold air pressure, the knock vibrations, the cylinder pressure, and the engine torque. All of these signals simulation should be synchronized with the crankshaft rotation angle (event based) and not in time (time-based).

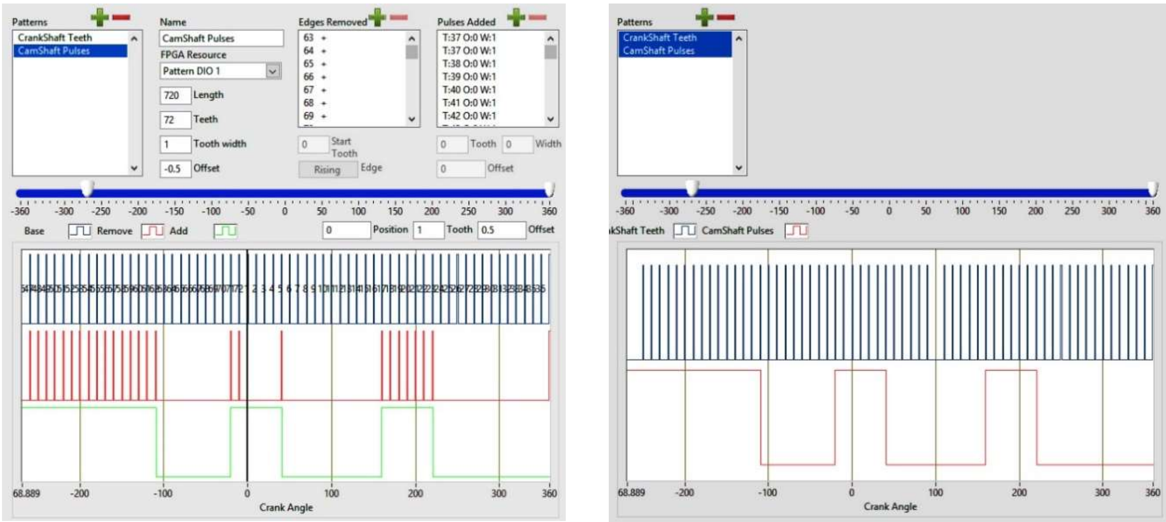

Fig.7 Programming windows for camshaft pulse pattern generation

There are also values which having an asynchronous evolution with the crankshaft revolutions, such as the throttle position, the ambient pressure, the intake air temperature, the engine coolant temperature, the signals from the switches actuated by the car driver, etc. The same characteristic may be extrapolated to the output signals generated by the simulator for all these signals. In certain cases, the control signal should be emitted synchronously with the angular position of crankshaft, but it should last a very clearly defined time, regardless of rotational speed or angle (injection pulses, diesel start of injection duration, dwell time for ignition pulses).

The synchronization pulses generated by the rotational speed sensor are spaced at 10° or 6°, or in other words the simulator receives one pulse for every 10 or 6 degree of crankshaft rotation. The control signals of modern direct injection engines require much finer angular intervals, usually 0.1° RAM, or even smaller. Thus, between 2 consecutive teeth, the engine ECU and also the simulator requires 60 or 100 perfectly spaced pulses, which follow the speed and angular acceleration of the flywheel as a PLL multiplier.

In order to be able to meet these requirements, the simulator is equipped with two counter-timer, an angle marks generator and a state machine which are designed to generate appropriate pulses for a T/C1 reference time base, and a T/C2 programmable generator/counter angular marks. The angular marks generator can generate a configurable 1 to 1024 equally spaced pulses" inserted" between edge transitions coming from two successive teeth of the crankshaft speed transducer. The period of the angle marks counted by T/C2 is rated in time marks generated by T/C1.

For instance, if the toothed wheel on the crankshaft has 36 teeth (35-1 configuration) and a resolution of 0.1°crankshaft rotation is required, on each rotation T/C2 generates 3600 mark pulses for a whole revolution, respectively 7200 pulses for a complete engine cycle. Supposing a 9000 rpm rotational speed, the minimum frequency of the counter clock should be 540 kHz. For the 60 teeth configuration (58-2) and the same 0.1°cranckshaft resolution, we will need 60 angle marks for each tooth, which means also a 540-kHz frequency (see Fig.8). As one may notice, although the configuration and distance between the teeth are different in the two cases, the angle marks with a 0.1°RAM resolution have the same duration, namely 1.85 μs.

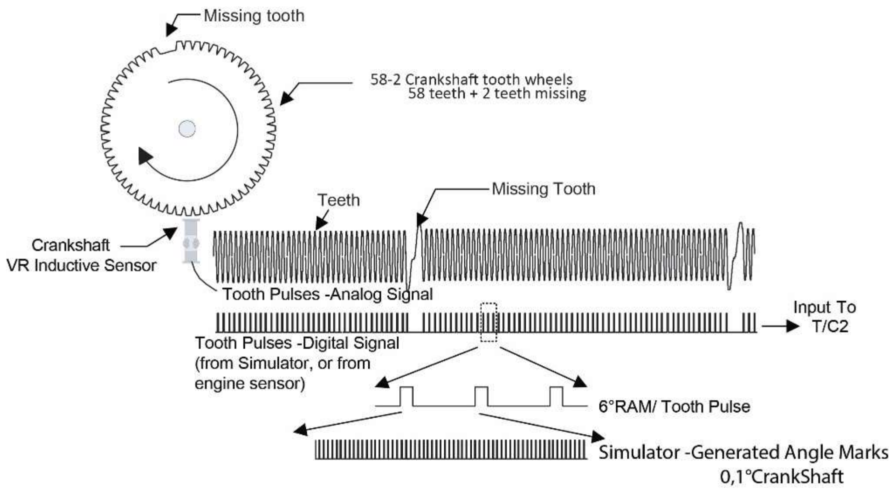


Fig.8 Angular marks pulse generator

Although two successive teeth always have a constant angular deviation, the time elapsed differs with the angular acceleration and speed of the crankshaft. Thus, the frequency of the angle marks will have to change accordingly between two successive teeth in order to achieve an automatic compensation in case of crankshaft acceleration or deceleration. At the same time, when one or two teeth are missing, VEMS will generate its own angular marks signals like those coming from real teeth, so that the injection or ignition pulses should not have interruptions generating lack of ignition or combustion.

The angular marks generator thus becomes the main part of the simulator, its operation being implemented with a state machine which was tested in Simulink Stateflow and then embedded into a configurable Xilinx Zynq 7000 FPGA. For the generation of pulses representing angle marks, the generator has an internal clock, the most commonly used frequency being 100 MHz (10 ns period).

The NI-9361 compact module is equipped with 8 timer/counter with 32 bite resolution, a 100 MHz internal base which allows down prescaller, and a maximum sampling rate (capture of transitions coming on the counting line) of 102.4 kHz. For example, from a rotational speed of the crankshaft of 9000 rpm, the 35-1 pulses have a frequency of 5.4 kHz, whereas the maximum frequency of the 58-2 configuration of the generating gear is 9 kHz; both options are easily covered by the above-mentioned module.

The synchronization process starts by searching the area where one, two or even three teeth are missing. For simplicity reasons and for explanations, the 36-1 configuration is considered. The basic function of any timer/counter consists of the edge detecting (rising, falling or both fronts) of a train of rectangular pulses and the measuring of the period (frequency) by using a large internal frequency base. Thus, it is possible to measure the period between two fronts of the teeth signal and to set a validation window where the next tooth is expected to appear. If the next tooth fails to appear in the validation window, the missing tooth sector may be next, or the crankshaft may suffer a deceleration. Conversely, if the next tooth appears before the expected time calculated in the estimation window, the crankshaft is supposed to have accelerated.

The simulator should detect whether the lack of a pulse in the validation window is due to a missing tooth or to a very strong deceleration, and whether the lack of two successive pulses represents the two missing teeth in a pulse wheel with 60-2 configuration or is the consequence of a missing tooth in the 36-1 configuration and of a sharp deceleration.

As a rule, maximum accelerations are obtained at low crankshaft speed. For example, from the cold starting speed of about 50 rpm, the crankshaft can achieve in a very brief time 5000-7000 rpm. Such a case is presented in Fig. 9 which illustrates the time evolution of pulses coming from a 60-2 crank wheel where a ratio between the teeth period $t_A$ immediately preceding the area with two missing teeth (B sector) and the period $t_C$ between the first two teeth after missing teeth area can reach a 75 value. High crankshaft speed variations result from changing high gear to low gear in a single maneuver. Smaller ratios are obtained in very strong braking maneuvers. At higher crankshaft speeds, much lower variation of the ratios between successive teeth periods are obtained.

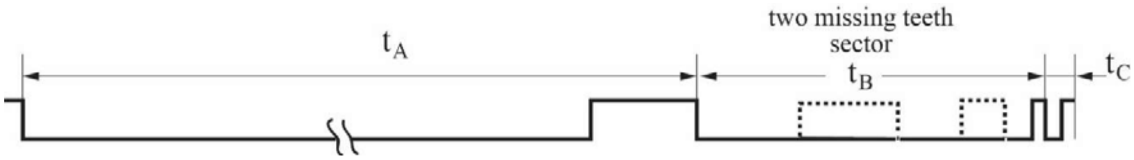


Fig.9 Time evolution for pulses comming from a crank wheel with 60-2 teeth configuration wich is accelerated from 50 rpm to 5000 rpm in a very brief time

The VEMS was designed to accommodate constant crankshaft angular accelerations or decelerations, this assumption being accepted in all calculations regarding crankshaft kinematic and dynamics.

We assume a crankshaft acceleration from initial speed of $n_0$=900 rpm to the final value of $n_f$=6000 rpm in one second. Internal timing base of the simulator is driven by a clock frequency of 100 MHz. Crank wheel has 35 teeth and one missing tooth and generates one pulse for every 10° crankshaft rotation. VEMS can generate a maximum of 1024 angle marks for each tooth, but for simplicity of this example a value of 100 is considered resulting a 0,1° crankshaft rotation accuracy.

As in the real case of engine acceleration, where the transition from 900 rpm to 6000 rpm is made up gradually in a specified number of crankshaft successive revolutions, the simulator must have a similar behavior and decrease gradually teeth period from $t_0$ =1,85 ms to $t_f$=0,277 ms (277 µs) as resulting from relations (1) and (2).

$$t_0 = \frac{60}{n_0 \cdot z_{teeth}} = \frac{60}{900 \cdot 36} = 1{,}85\ ms \tag{1}$$

$$t_f = \frac{60}{n_f \cdot z_{teeth}} = \frac{60}{6000 \cdot 36} = 0{,}27\ ms \tag{2}$$

Angular marks will gradually change their frequency from initial value of 54 kHz (i.e. 900/60x36x100) to the final value of 360 kHz. Crankshaft and camshaft pulses generated by VEMS for this acceleration are partially shown in Fig.10.

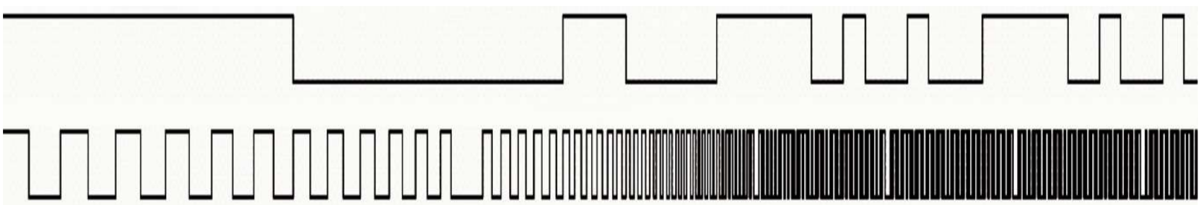

Fig.10 Partial view of a crank and cam pulses evolution for a simulated acceleration of the crankshaft between 900 and 6000 rpm in 1 sec.

From the kinematic equations of circular movement between angular velocity $\omega = \frac{\pi \cdot n[rpm]}{30}$, rotation angle $\theta[rad]$ and angular acceleration $\varepsilon[\frac{rad}{sec^2}] = ct$ it results the change in angular velocity between two succesive teeth separated by a 10° angle:

$$\omega_1^2 - \omega_0^2 = 2 \cdot \varepsilon \cdot \Delta\theta = 2 \cdot \left(\frac{\pi \cdot n_f}{30} - \frac{\pi \cdot n_0}{30}\right) \cdot \frac{1}{\Delta t = 1\ sec} \cdot \frac{10° \cdot \pi}{180°} = 186{,}425\ \frac{rad}{sec^2} \tag{3}$$

Before acceleration, all teeth spin with a constant angular velocity $\omega_0 = 94{,}241\frac{rad}{sec}$, but after acceleration the new angular speed reflected on the first tooth is $\omega_1 = 95{,}231\frac{rad}{sec}$, according to relationship (3). The frequency of the angular marks is changing according to the new velocity but it is maintained to the new value for the rest of whole 10° tooth period.

Thus, the speed difference counted at the first tooth that accelerate is about $\frac{30}{\pi} \cdot (\omega_1 - \omega_0) = 9{,}45$ rpm. Relation (3) can be generalized for any successive teeth in the form given by equation (4).

$$\omega_{i+1}^2 - \omega_i^2 = 186{,}425\ \frac{rad}{sec^2} \quad i=0\div 35 \tag{4}$$

Again, using relation (4) to the second tooth gives an angular velocity $\omega_2 = 96{,}205\frac{rad}{sec}$, which means an increasing speed with 9,3 rpm compared with the first tooth, and with 18.75 rpm compared to static angular speed at 900 rpm. As can be seen from this example, the ratio between tooth period at 900 rpm and 6000 rpm is only 6,8, also the faster the crankshaft speed, the less is the tooth period modification.

The basic function of the synchronization process is to generate angle ticks with well-defined frequency (usually for achieving 0,1° or 0,01° crankshaft rotation accuracy) used to clock a timer/counter T/C2 which defines at any time the position of crankshaft and the evolution of each cylinder during on a whole engine cycle (720°) with reference to the TDC of first cylinder. The synchronization process is controlled by a state-machine that can be implemented either in Simulink Stateflow or using LabView StateChart and Control Design Toolboxes.

Synchronization process start with teeth pulses edges (rising edge, falling edge, or both). First edge captured with a Timer/Counter from NI-9361 cRIO timing module denotes that the crankshaft rotates without providing any further information. The detection of the second tooth allows the calculation of the first period between two teeth and the estimation of the validation window for the next coming tooth.

As can be seen from Fig.11 this period can be either between two normally spaced successive teeth or between two teeth separated by a missing tooth (missing teeth) sector.

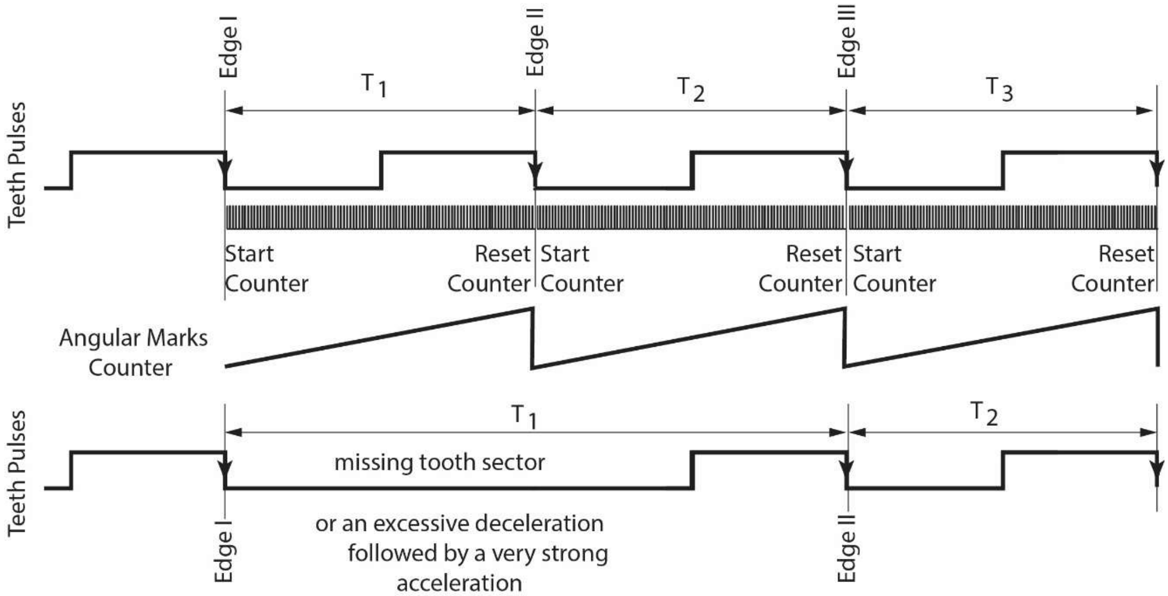


Fig.11 Teeth pulses fronts and missing tooth sector are the key elements in synchronization process

Synchronization can be done only when a missing tooth (teeth) sector is recognized. A simple comparison test between the current and previous tooth periods can identify a possible missing tooth area, but also a sharp deceleration. So, another test is necessary to compare the current period (possible missing tooth) with the next tooth period. In the worst case, 35 fronts must be captured and 34 periods must be measured before the transducer can reach the missing tooth sector. On each front, the angular marks counter start counting until the next front is captured, when it is reset and restarted again over and over till the missing area is detected.

A valid test to detect the missing teeth sector is based on the observation that that an acceleration followed by a sharp deceleration and immediately succeeded by another crankshaft acceleration is physically impossible to meet in practice because of the crankshaft inertia. The above example can also prove this supposition. A good test for missing tooth sector is based on two comparisons between three periods corresponding to three successive teeth: $T_{i-1} < \kappa \cdot T_i > T_{i+1}$, where k is an engine related factor which can be set on test bed, common values being in range 0,3-0,8.

Once the sector with a missing tooth (teeth) has been detected, the VEMS control core start the angular marks timer counter T/C2 without resetting it at the next tooth pulse. In this situation there are three options, but only the first two have been implemented on this simulator.

The first variant is useful for simulating simultaneous and/or grouped fuel injection. The angular marks counter starts after the missing tooth (teeth) sector is detected and stop counting after one revolution of the crankshaft. Position of the TDC for each cylinder is well defined but not the phases from an engine cycle (720°).

In the second mode of operation, the T/C2 will count for two successive rotations before being reset and restarted on each engine cycle (720° crankshaft rotation). The both modes of operating are depicted in Fig.12.

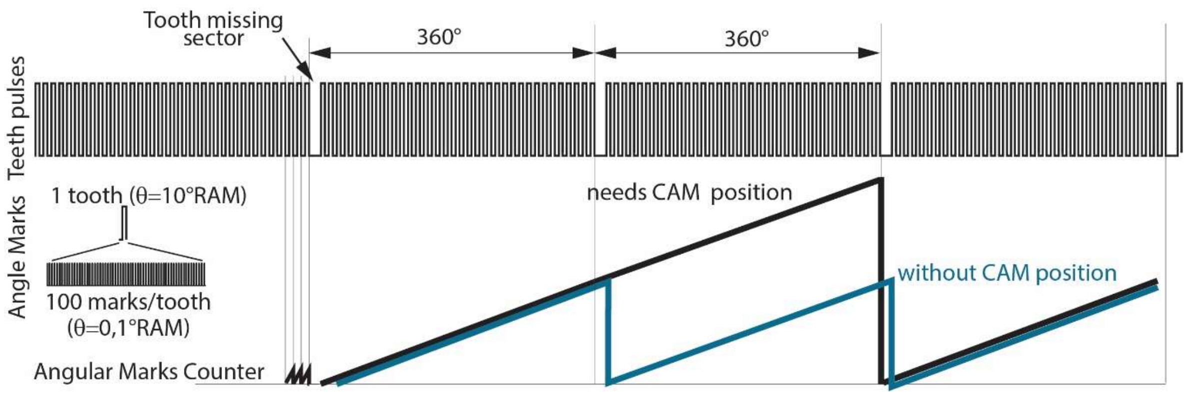


Fig.12 Timer/Counter counts angular marks over a full crankshaft revolution or over a full engine cycle

The third operating mode not yet implemented make possible that angular marks counter will increments on each tick generated by T/C1 with precision of up to 0.01 ° for several consecutive motor cycles thus allowing advance programming of synchronous events related to the crankshaft over multiple cycles. Considering that NI-9361 Compact Module features a 32-bit 8-bit timer/counter, at 9000 rpm, with a crank wheel in the 60-2 tooth configuration and 1024 angular marks for each tooth, it would result that a single counter can count 233 successive engine cycles until overflow and will be automatically reset.

## 4. VEMS Output Signals

VEMS can generate two types of signals: speed, synchronization and position of the crankshaft, or engine control pulses. In the first category, we find the signals that usually come from engine crank and cam sensors and allow the quantification of the engine crankshaft position to an accuracy of up to 0,01° crankshaft rotations. These signals can be also generated by the simulator and routed to the internal counter inputs, as if coming from external sensors. The simulator also allows the generation of analog signals like those collected from inductive sensors on a real engine. All these signals have been extensively analyzed in Chapter 2 dedicated to simulator input signals and will not be resumed any further. The second category of VEMS output signals comprises injection, ignition and pulses train that generates windows for knock signals detection and analysis.

The simulator can generate pulses for manifold injection or to generate and control current pulses for direct injection.

Pulses for simultaneous and/or grouped fuel injection can be generates on every crankshaft revolution because the fuel quantity required for combustion is injected into two stages: half in one revolution of the crankshaft (360 ° crankshaft rotation) and the rest in the next revolution. For group fuel injection, one injector group (half oh cylinders) injects the total fuel quantity in the first revolution of the crankshaft, and for the next revolution the second group injects. In both cases, there is no need to distinguish between the first rotation and the next one in a motor cycle, and the angular quanta counter is restarted at each detection of the first tooth after the missing teeth sector.

Fig.13 shows an oscilloscope capture with port injection pulses (on the third channel) in synchronization with crankshaft pulses also generated by simulator.

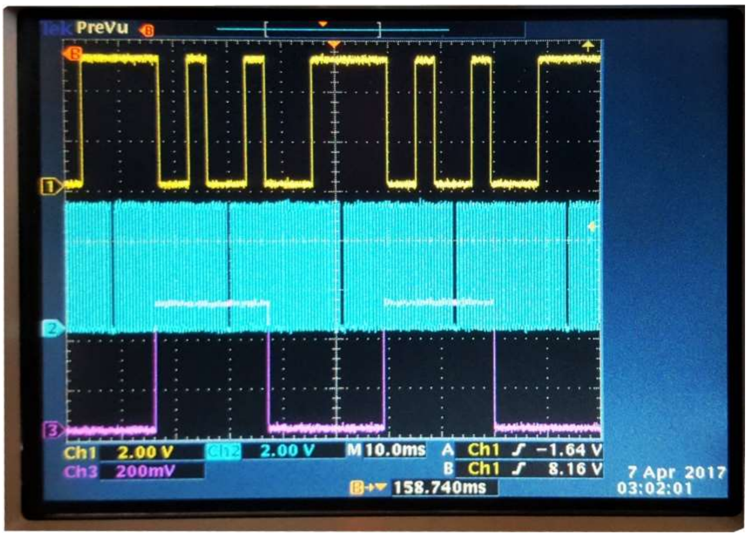


Fig.13 Cam, crankshaft and group injection pulses generated by simulator and viewed on the oscilloscope

Injection pulse width can be programmed in milliseconds with T/C1 but the start of injection pulse and the end of injection pulse positions in relation to crankshaft teeth pulses can be determined as an angular marks values of T/C2.

In case of sequential or individual port injection, but especially for direct injection, it is mandatory to know the position of each cylinder in a complete engine cycle and to make the difference between the first rotation and the next rotation of the crankshaft over a full engine cycle. One NI-cRIO 9758 Port Fuel Injector Driven Module was used to generate pulses for up to eight low pressure injectors.

One of the most complicated task of the VEMS is to generate control signals for high pressure injectors used in direct injection. There are many new features compared to manifold injection:

Several injection pulses (up to five) can be generated each engine cycle for each injection valve;

Each injection must start at a predetermined angle and consists of many phases (up to 5 A-E) as can be seen from the Fig.14;

Each injection event may have different time for the same cylinder and/or from cylinder to cylinder;

Each phase uses a single pulse or a train pulse width modulated (PWM) to control a power transistor such that the current passing through the injector coil complies a certain shape and specific values (see. Fig.14);

During opening phase of the injector (phase A), the voltage has a higher value than battery (40-65V -VBoost). Then, VEMS switches to phase D and reduce the current through injector only to keep it open. At the end of injection, the current decay back to zero.

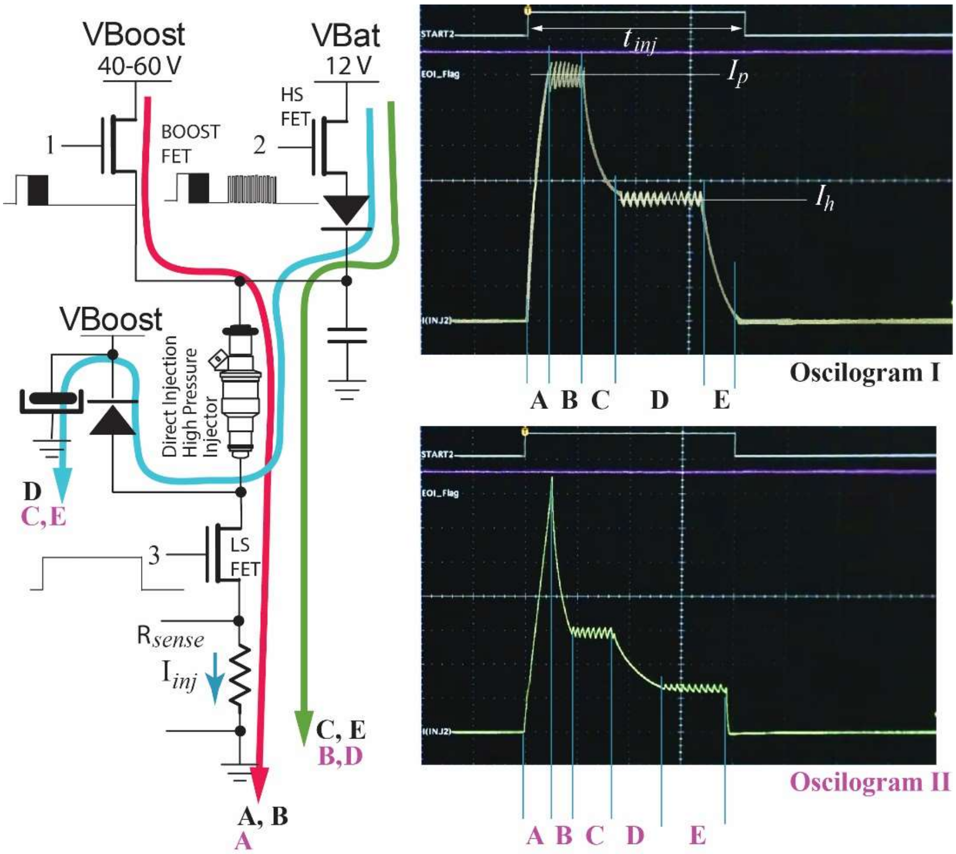


Fig.14 Current shapes used to drive solenoid valves for high pressure direct injection and the power stage driven by VEMS implemented with NI-9751 cRio module

VEMS use a compact RIO NI-9751 able to drive 3 solenoid type gasoline or diesel direct injectors, or piezo actuated injector. As can be seen in Fig.14 two current direct injection shapes (profiles) was generated and recorded on the digital oscilloscope.

Fig.15 is a screen captured from of a logic analyser showing a sequence of four ignition pulses and four injection pulses used to control a direct injection spark ignition engine. All these control signals are set with a high accuracy (0,01°) in relation to the teeth and cam lobes also generated by simulator.

For each cylinder, the simulator delivers one main dwell pulse for ignition coil with a programmed advance angle before TDC (measured in angular marks) followed by a programmable number of secondary ignition pulses. Multi spark ignition technique is used to enhance the combustion of very lean stratified mixture. Such a sequence of four ignition pulses is captured on the oscilloscope and can be visualised in Fig.16.

Spark pulses can be programmed to be active for at least a minimum amount of time to provide a sufficient sparking energy, or to not to exceed a maximum amount of time, which would cause the ignition coil to overheat. Thus, it is possible to provide an appropriate dwell angle, respectively, an optimum spark advance regardless of the accelerations or decelerations to which the crankshaft is subjected. It is also possible to generate more pulses for both ignition and injection events (pilot, main injection, post injection, etc.).

VEMS can generates multiple injection event per engine cycle such as pilot, main, secondary or delayed post- injections. For each injection pulse must provide the following parameters before running the simulation: injection pulse width $t_{inj}$, time for peak current $t_p$, time for hold current $t_h$, peak current $I_p$ and hold current $I_h$.

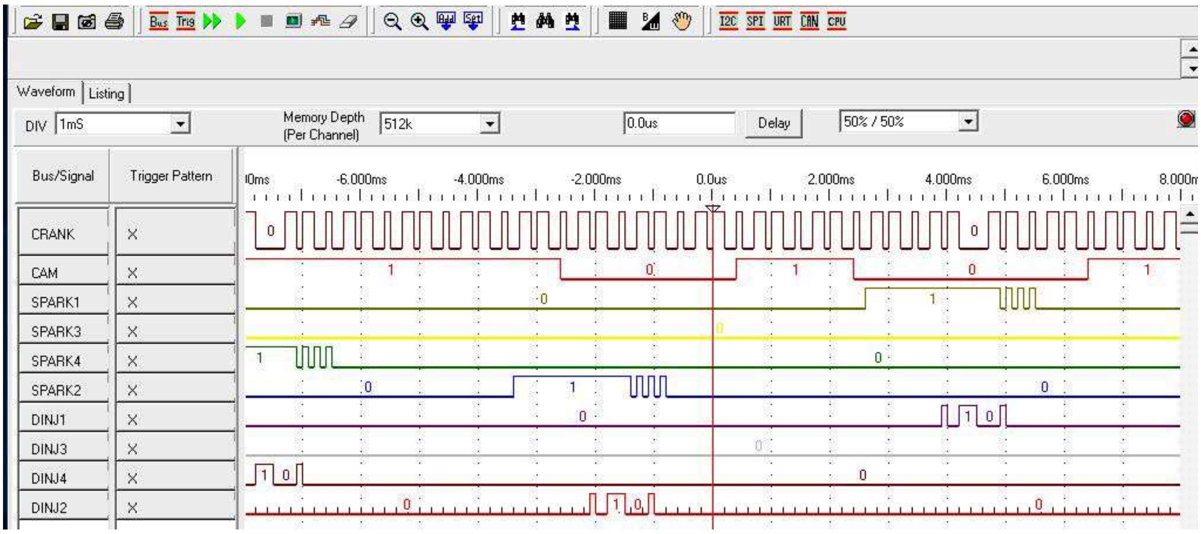


Fig.15 Spark ignition and direct injection pulses generated with VEMS and captured with a PC digital analyzer

Peak current is used for the injector fast opening and it must grow up from o to 20A in a very short time (few microseconds). This is achieved by means of a booster voltage of up to 65V. During pick-up phase $t_p$ battery voltage is applied to the solenoid valve and assist in opening quickly. In this time current $I_p$ is to approx. 20 A. During holding phase the current is dropped to approx. 13A and is entirely supplied from battery.

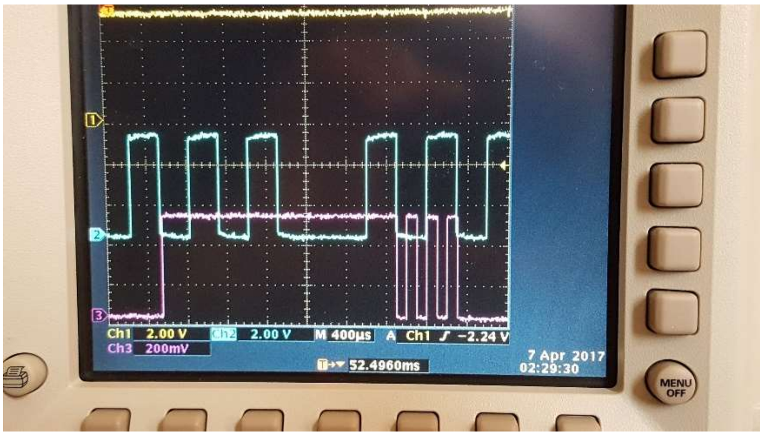


Fig.16 Spark ignition pulses simulated 2500 rpm speed of the crankshaft. VEMS generate one main ignition pulse with 30° crankshaft rotation and 2,4 ms width followed by three secondary pulses

The total time of the injection pulse is a critical parameter and is very well establish for each engine on the test bed and stored in ECU memory as three-dimension map. The simulator can use only a single set of values as a demonstrative tool. The angular distance between the start of injection, angular distance between the pilot and maim injection, or between the main and post-injection pulses can also to be programmed before the start of the simulator.

Fig 17 is another capture of the logic analyser in which are show all the simulator control signals for a gasoline direct injection engine including the knock windows and crankshaft and camshaft synchronization signals.

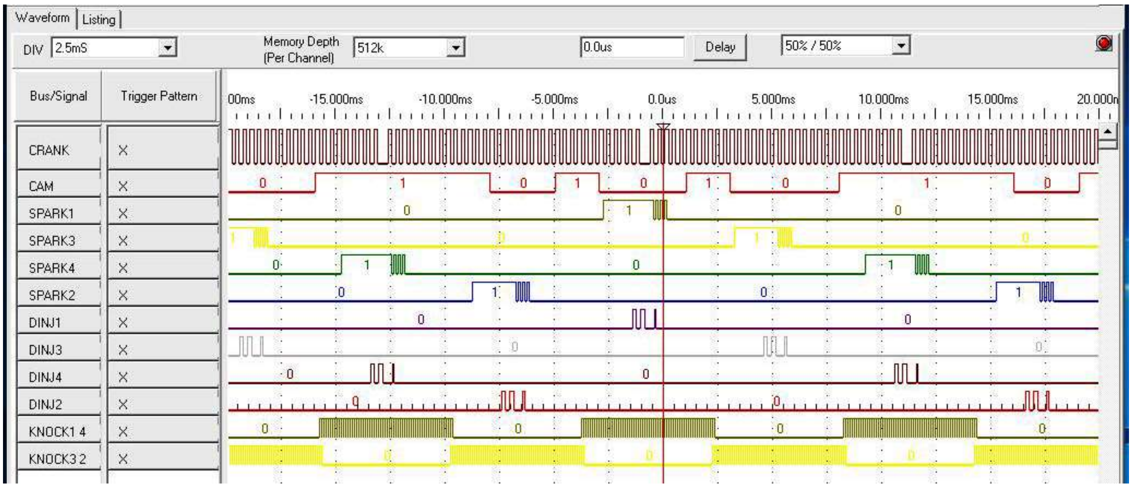


Fig.17 Spark ignition pulses simulated 2500 rpm speed of the crankshaft. VEMS generate one main ignition pulse with 30° crankshaft rotation and 2,4 ms width followed by three secondary pulses

## 5. Conclusions

Thus, the simulator described in this paper is a hardware/software system comprehensive enough to allow the development of complex projects on the control of any internal combustion engine, without the actual physical presence of the latter. This has several noteworthy advantages such as the simplicity and cost effectiveness of the project, which does not require the actual presence of the engine and of the test bench, at least not during the designing, modelling and simulation stages. The operating conditions that may be modelled here are far more diverse than the ones that may be implemented on a real engine, the control of which is achieved by means of an unchecked electronic control unit prototype, which may actuate the running of the device in dangerous conditions or, on the contrary, cannot cope with extreme working conditions. Moreover, the simulator has the advantage of being able to repeat the same simulated states and conditions without cyclic variations or deviations.

## References

[1] Boulat A, Genninasca Y, Charlet A and Higelin P 2001 *ECUTEST-A Real-time Engine Simulator for ECU Development and Testing*, SAE Technical Paper 2001-01-1911
[2] Viele M, Stein L, Gillespie M and Hoekstra G 2004 *A PC and FPGA Hybrid Approach to Hardware-in-the Loop Simulation*, SAE Technical Paper 2004-01-0904
[3] Wang J and Sarlashkar J 2007 Engine Crankshaft Position Tracking Algorithms Applicable for Given Arbitrary Cam- and Crank-Shaft Position Signal Patterns SAE Technical Paper 2007-01-1597
[4] Barrett S and Bouchez M 2015 *Addressing Engine ECU Testing Challenges with FPGA-Based Engine Simulation*, SAE Technical Paper 2015-01-0173
[5] Barrett S 2015 *NI Engine Simulation Toolkit for NI Veristand*, NI Press https://decibel.ni.com/content/docs/DOC-37304
[6] Anand P and Saravanan C G 2012 *Development of Research Engine Control Unit Using FPGA-Bassed Embedded Control System*, Journal of Kones Powertrain and Transport, Vol.19, No.3 2012
[7] Strapko M and Tichanek R *2012 Engine control unit based on NI compactrio platform*, Journal of Middle European Construction and Design of Cars