

# Minimal algorithm for running an internal combustion engine

V Stoica<sup>1</sup>, A Borborean<sup>1</sup>, A Ciocan<sup>1</sup> and C Manciu<sup>2</sup>

<sup>1</sup>Politehnica University of Timisoara, Department of Mechanical Machines, Equipment and Transportation, Mihai Viteazu 1, 300222 Timisoara, Romania

<sup>2</sup>Continental Automotive Romania SRL, Siemens Street No. 1, 300704, Timisoara, Romania

E-mail: [virgil.stoica@upt.ro](mailto:virgil.stoica@upt.ro)

**Abstract.** The internal combustion engine control is a well-known topic within automotive industry and is widely used. However, in research laboratories and universities the use of a control system trading is not the best solution because of predetermined operating algorithms, and calibrations (accessible only by the manufacturer) without allowing massive intervention from outside. Laboratory solutions on the market are very expensive. Consequently, in the paper we present a minimal algorithm required to start-up and run an internal combustion engine. The presented solution can be adapted to function on performance microcontrollers available on the market at the present time and at an affordable price. The presented algorithm was implemented in LabView and runs on a CompactRIO hardware platform.

## 1. Introduction

There are many developed algorithms for engine control in automotive industry but, in the literature, there is no description of them, they are only mentioned [1-3].

Anand [4] presents roughly the developed algorithm but the most important parts, the engine position tracking, injector and ignition coils control, are missing. Strapko [5] provide a complete overview of the entire developed system including the electronic modules for signal conditioning, but the module for crankshaft position tracking is not clearly described and also the frequency of the FPGA loops are not provided. Busuttill [6] is still using analogic electronics for controlling the water injection within the intake manifold injection that is triggered by the fuel injector signal without the possibility to control the water injection starting moment.

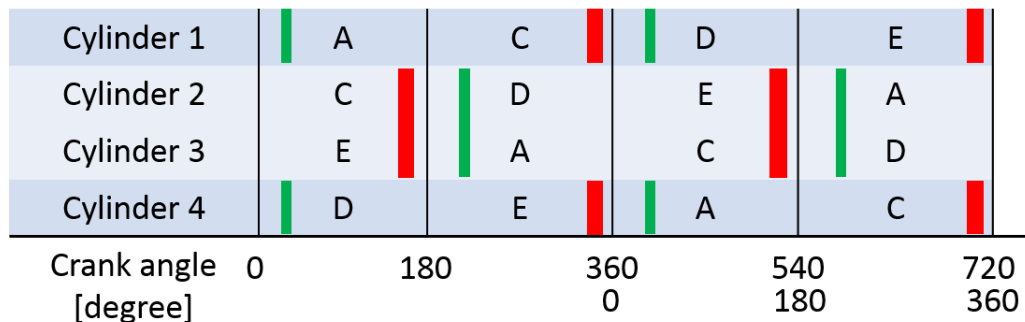
Within this context we developed a minimal algorithm for starting an internal combustion engine. The presented modular algorithm can be adapted to any needs and can be completed with some other modules for controlling other injectors for water or a secondary fuel like alcohol or biogas for example.

## 2. Algorithm description

A spark ignition 4 stroke engine with 4 cylinders can be managed by a double spark and injection as used to be in case of EURO 3 engine. The spark is provided in the same time at two cylinders, one at desired moment (the end of the compression stroke), and the second one is provided on the evacuation stroke, and it will be unused. The same solution is applied in case of fuel injection. One injection is provided during admission stroke and the second one during detention (each time is injected half of the desired fuel mass). A schema of the injection (green), and ignition (red) timing overlapped over the engine strokes is presented in **Error! Reference source not found.** This solution requires the crank

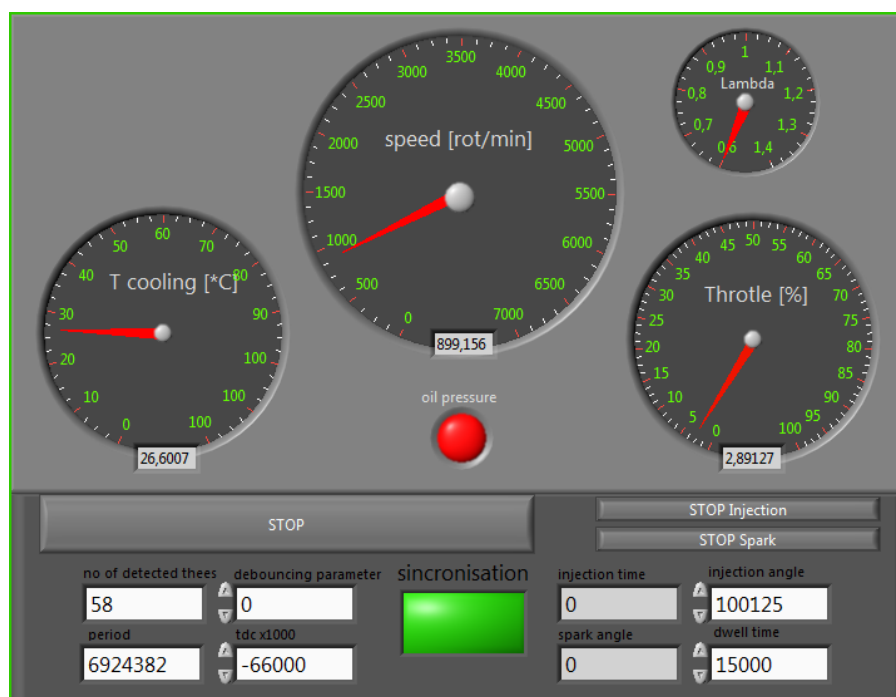


angle tracking only in the interval 0-360 degree and not necessarily a complete engine cycle of 0-720. Therefore a second sensor mounted on the camshaft is not required.



**Figure 1.** Ignition and fuel injection timing

The engine management program is divided in two main parts, one running on the real time module **Error! Reference source not found.**, and the second one, time critical, running on the FPGA.

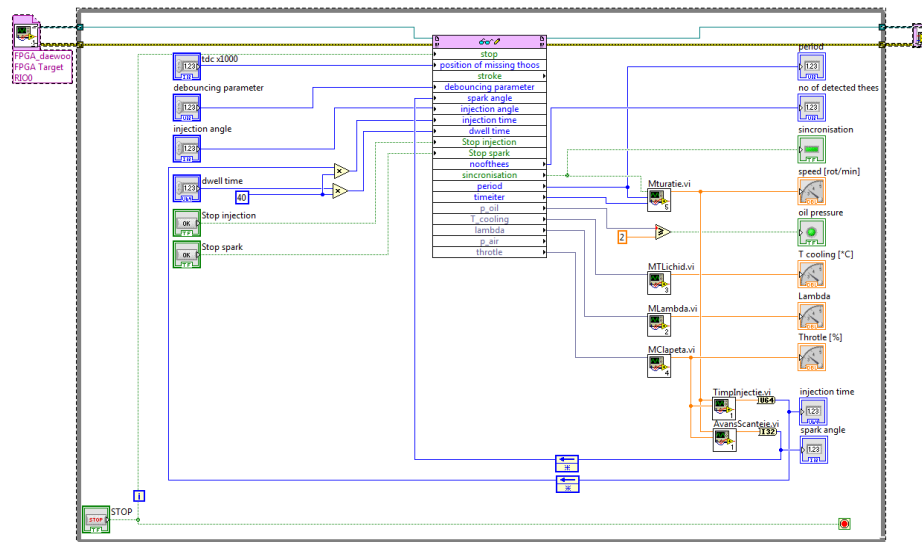


**Figure 2.** Program graphical user interface

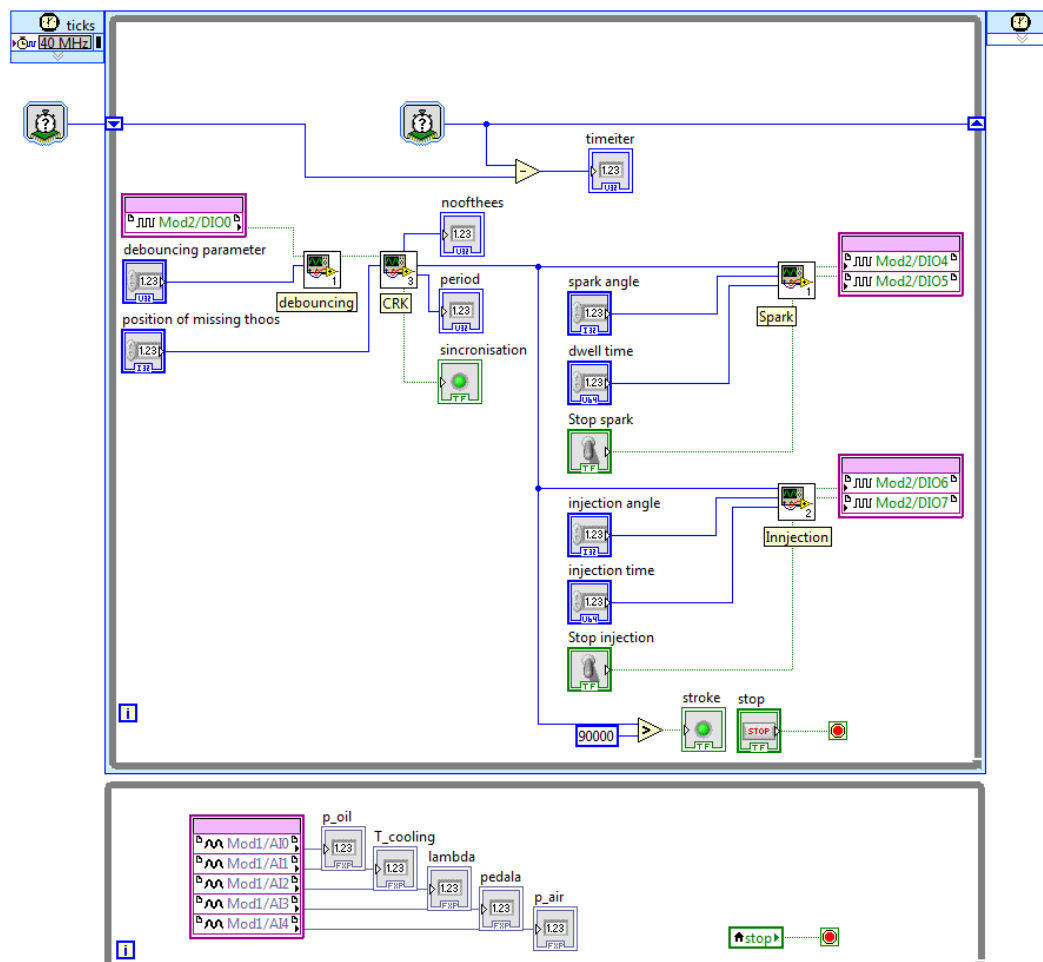
An overview of the program which is running on the FPGA is presented in Figure 4. It consists of 4 parts:

1. Monitoring and establishing the crankshaft position (CRK module);
2. Commanding the injectors (Injection module);
3. Commanding the induction coil (Spark module);
4. Acquiring data from the sensors.

The first 3 parts are running in a clock cycle loop (40MHZ) and the last one, less important, running in a separate loop cycle.



**Figure 3.** The main program running on the real time module

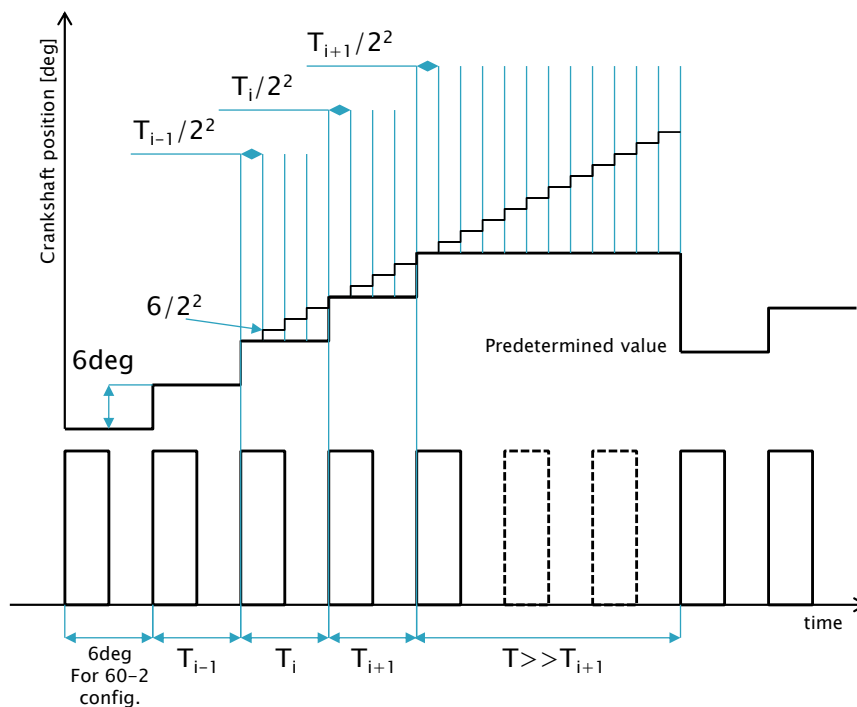


**Figure 4.** The overview of the module running on the FPGA

The most important parameter, that has to be monitored for an exact ignition timing, is the engine position, which in our case is done using an inductive sensor together with a special tooth wheel in a 60-2 configuration. The LabView program (the algorithm) is presented in Figure **Error! Reference source not found.6**. The program consists of:

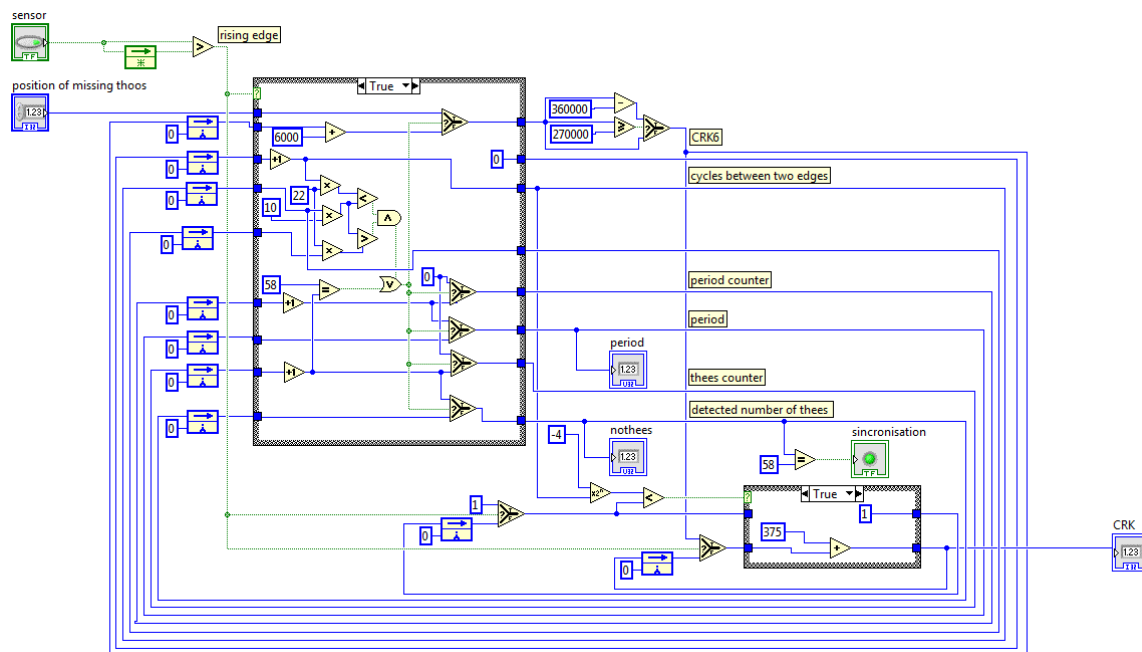
- Signal debouncing for rejecting de undesired noise;
- Determine the angle with a precision of 6 degrees by adding 6 degrees each time a rising edge is detected. The angle varies between -90 to 270;
- Estimating the angle in between the edges with a step of 0.375 degree by linear interpolation. For the interpolation the time between the previous consecutive edges is divided in 16 ( $2^4$ ) intervals, and with a help of a timer (counter) the angle is incremented with 0.375 ( $6/2^4$ ) at the end of each interval. As mentioned in literature other interval numbers are possible [2];
- Detecting the missing teeth based on the fact that the time period between 2 consecutive teeth is much smaller than the time period detected in case of the two missing ones;
- Counting the rising edges (number of teeth);
- Counting the period of a complete revolution.

An intuitive picture of the steps b, c and d is presented in Figure 5. For a better graphical description the angle is incremented only with 1.5 degrees ( $6/2^2$ ).



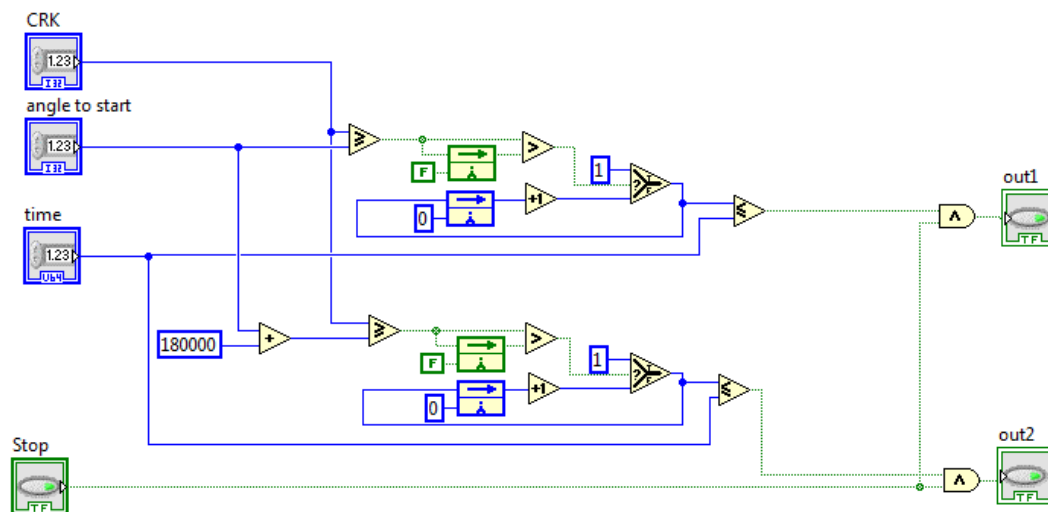
**Figure 5.** Crankshaft position incrementing and estimating procedure

(**Notification:** because on the FPGA is recommended to work with integer numbers and to use simple math, the angle values are multiplied by 1000).



**Figure 6.** The module used for crankshaft position tracking

For controlling the ignition and the fuel injection mainly counters are used. The subroutine is presented in Figure 7.



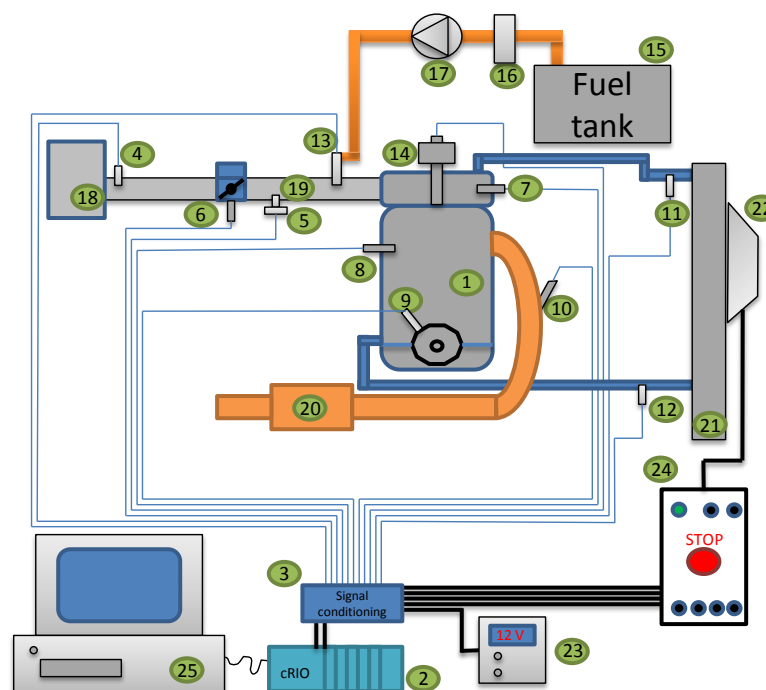
**Figure 7.** The module used for commanding the fuel injection and the ignition system

The counter is started at the desired crank angle and the output is maintained on 1 the prescribed time (number of cycles) which mean the injection time or the dwell angle.

### 3. Experimental setup

The experimental set up (Figure 8) consist of the Daewoo 1.6 liter engine 1 which is controlled by the cRIO system 2. The signals from and to the engine are passing through the signal conditioning box 3.

For monitoring the engine state, the following sensors are used: 4- intake air temperature, 5 intake air pressure, 6 throttle position, 7 cooling liquid temperature, 8- oil pressure, 9 – crank angle position, 10 – oxygen level within the exhaust gases (binary type) and cooling liquid temperature at the input 11 and output 12 of the heat exchanger. The control of the engine fueling and ignition is done with the injectors 13 and ignition coils and spark plugs 14. The fuel is brought to the common rail from the tank 15 through the filter 16 by the pump 17. The intake air passes the filter 18, the intake manifold 19 and riches the engine. The exhaust gases leave the engine and pass the catalyst 20. The engine cooling is assured with the heat exchanger 21 and the cooling fan 22. The 12 Volt power supply 23 ensure the powering of the entire system. The switches board 24 is for emergency situations and for activating or deactivating some subsystems like injection, ignition and cooling. The computer 25 is used for programing the cRIO system, for visualization of the monitored parameter and for prescribing the injection and spark timing.



**Figure 8.** The experimental setup



**Figure 9.** The experimental test bench

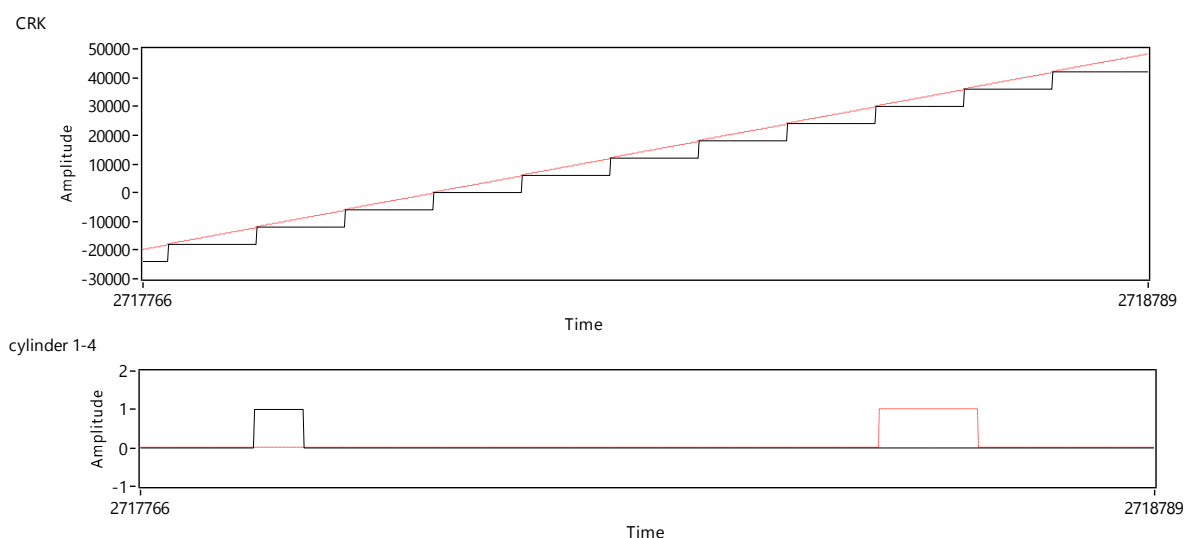
The controlling system consist of a real time power PC embedded controller for Compact RIO of 533MHz, with 2Gb storage and 256 Mb DRAM, a TTL digital module NI 9401 with 8-Channels, with a response time of 100 ns, and a 16-Bit analogic input module NI 9205 with 32-Channels having the input range of  $\pm 10$  V, and an acquiring frequency of 250 kS/s.

#### 4. Results

Before implementing the algorithm, on the cRIO system, it was tested and the results are presented in.

##### Error! Reference source not found.10

As mentioned before, the crankshaft angle is multiplied by 1000 therefore in figure 10 on the upside the angle is in the range -30 to 50 degree. The spark is produce at 10 degree before top dead center at the end of the compression stroke and the injection starts at 40 degree after the top dead center during the detention stroke.



**Figure 10.** LabView simulation of the presented algorithm. up) The crankshaft position CRK with a 6 degree precision with black and 0.375 with red. down) The spark timing with black and injection timing with red

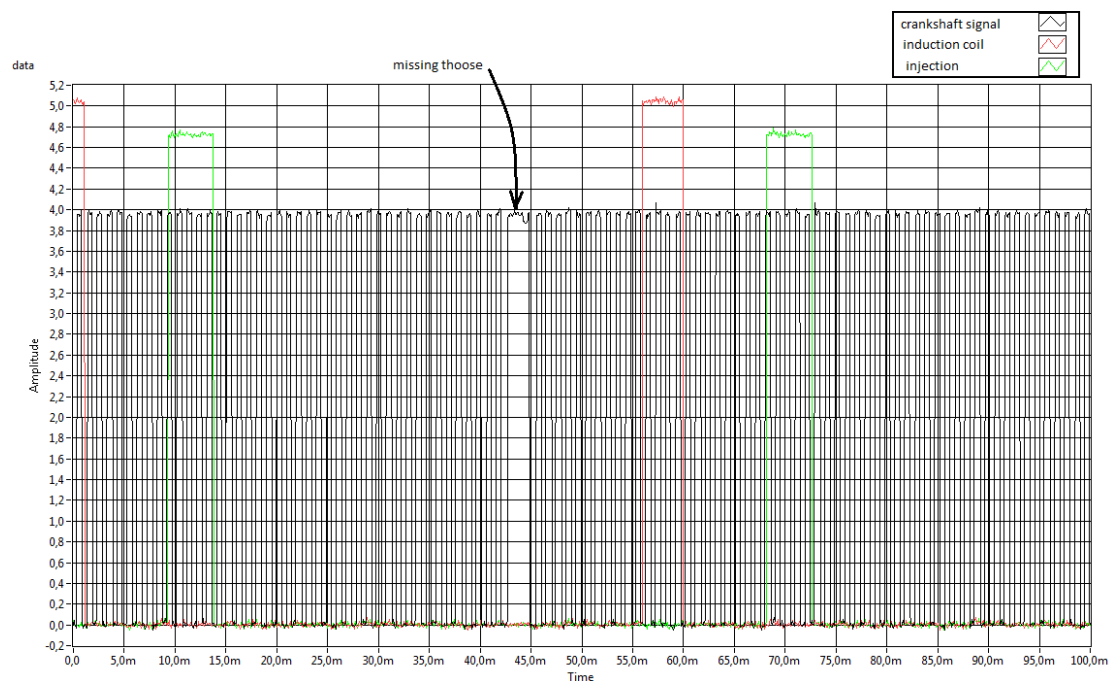
Similar results are obtained on the experimental bench test and they are presented in Figure 11.

Using the presented system it was possible to change, in real time, the injection timing and sparking and to monitor the most important parameters of the engine.

#### 5. Conclusions

The modular solution of the software presented here permits in future the rapidly development of different configurations like: dual injection gas and biogas, gas and hydrogen, programs for diesel engines with common rail systems and so on. With this hardware and software solution the implementation time of different algorithm is very short. Also this solution gives to mechanical engineers the possibility to implement and test their own algorithms in a more convenient way. The presented algorithms was simulated and then implemented on a Daewoo 1.6 liters spark ignition engine. The engine was tested without load in a speed domain of 800 to 2500 rpm. During the preliminary tests the injections timing was determined for different throttle position bringing the indication of lambda sensor to 1 (stoichiometric value).

The developed system will further be used for complete calibration of the engine, after mounting a hydraulic break, and will be tested with different fuel combinations and controlling strategies.



**Figure 11.** Measured data on the experimental test bench

## References

- [1] Dase C, Falcon J S and Maccleery B 2006 Motorcycle Control Prototyping Using an FPGA-Based Embedded Control System, *IEEE Control Systems Magazine*, October
- [2] Viele M 2012 *Applicatons of field programmable gate arrays for engine control*, Colorado State University, Fort Collins, Colorado, USA, Doctoral thesis
- [3] Irimescu A, Vasiu G and Trif-Tordai G 2014 Performance and emissions of small scale generator powered by a spark ignition engine with adaptive fuel injection control, *Applied Energy* **121** 196-206
- [4] Anand B P and Saravanan C G 2012 Development of research engine control unit using FPGA based embedded control system, *Journal of KONES Powertrain and Transport* **19**(3) 9-18
- [5] Strapko M and Tichánek R 2011 Engine Control Unit Based on the NI CompactRIO Platform, *Journal of Middle European Construction and Design of Cars* **9**(3)
- [6] Busuttil D and Farrugia M 2015 Experimental investigation on the effect of injecting water to the air to fuel mixture in a spark ignition engine, *MM Science Journal (March)* 585-590