Wood gas generator management system for vehicles

A A Popov, E M Gritsenko and S A Nosyrev
Reshetnev Siberian State University of Science and Technology, prospect “Krasnoyarsky rabochy“ 31, Krasnoyarsk, Russia

E-mail: tolnbms@yandex.ru

Abstract. This article presents the design and development aspects of a software package and hardware configuration which are allows for the preparation and supply of an air-fuel mixture to a car’s internal combustion engines which using charcoal or firewood as fuel. The software package and hardware should be implemented as a fuel control system. The microcomputer processes the signals from the sensors and allows you to receive data using the semi-automatic control of the fuel system, in addition collects statistical data and provides telemetry output to the dashboard or on the remote display and can be controlled via wireless interface. The following information is recorded in the database: state of sensors, current system settings and actions performed by the user. A remote device running on the Android operating system is connected via the widespread wireless Bluetooth communication channel, real time information about the system and the fuel system control interface are displayed in graphical form.

1. Introduction
Heating boilers and internal combustion engines use carbon and hydrogen compounds extracted from the bowels of the earth - methane, propane and gasoline, which are becoming more expensive every year, engineers are looking for alternative ways using existing inventions. One of them is a wood gas generator used during World War II on civilian vehicles. [1].

At present time, gas-generating technologies have found their application in stationary boiler equipment and have occupied their niche here as factory heating systems for rooms and water heating [2].

In an attempt to find factory gas-generating equipment for automobiles, it was found that in this area there are no companies mass-producing plants of this type, it is possible to order such equipment only from a private master. [3]. The only mass alternative to a standard gasoline fuel system is autogas (LPG) - a fuel system that operates using combustible gas stored in a gas bottle.

The usage of wood gas generating equipment in transport was previously carried out in conditions of a shortage of liquid fuel, such systems were manually controlled [3,4]. Over time, this technology has ceased to be used, since then in this area there have been few innovations. [5].

The increase in liquid fuel prices makes us think about alternatives. As an alternative, wood fuel was considered. Wood fuel in large quantities is produced as a by-product at forest processing enterprises and is not used efficiently in most cases, every year deadwood disappears in the forests, the harvesting and collection of which, according to federal law, is free from January 1, 2019 for citizens of the Russian Federation.
At present, wood fuel gas generating units are not universally used in automobile systems, digital control systems have not yet been developed for them. To popularize alternative fuel resources and revive the technology, it was decided to create such a system.

2. Methods

The gas generator allows gasification of solid fuel, which makes its use more convenient and efficient in conjunction with a heating boiler, internal combustion engine, gas turbine or in the chemical industry. When burning with a small amount of oxygen (pyrolysis), oxidation reactions of coal and hydrocarbons occur in the gas generator [5].

For the operation of the internal combustion engine, fuel is necessary, which must be supplied to the cylinders at certain points in time - this is the task of the fuel systems (fuel supply systems).

A car’s fuel supply system is a system designed to supply fuel (gasoline or diesel fuel) from a fuel tank to an engine (more precisely, to a carburetor or nozzle). Also, this system provides storage of fuel and its cleaning before feeding into the engine. Typically, heavy vehicles with a powerful engine are equipped with gas generating units, although passenger transport can be equipped with such a device if needed.

The system will consist of the following nodes:

- Gas generator;
- Cyclone type filter;
- Fine filter;
- Cooler;
- Centrifugal Fan;
- Gas mixer;
- Single board computer;
- Microcontroller;
- Telemetry sensors;
- Solenoid valves;
- System management software.

After analyzing various characteristics, the Raspberry Pi 3B microcomputer was selected due to lower market value in relation to competitors and a more developed community of developers of this particular hardware platform [6,7,8]. The Arduino Nano device was chosen as the microcontroller most suitable for the prototype project. [8].

After selecting the hardware, a visual layout of the devices was made in the form of a scheme (figure 1).

Automotive electrical system. The battery and its power wiring voltage of 12 or 24 volts. System devices, the power of which cannot be provided by means of a microcomputer or microcontroller, are connected to the power wiring (figure 1, block 1).

Control node. The microcomputer is powered by a step-down voltage converter, which is connected to the on-board electrical system of the car. The microcontroller and the Bluetooth wireless module are connected to the microcomputer via the USB bus. The microcomputer keeps statistics and an event log, and also serves as a bridge between the microcontroller and the remote control device. The microcontroller receives instructions from the microcomputer via the USB serial interface, transmits a control signal to the peripherals connected to it, and takes readings from the sensors (figure 1, block 2).
Figure 1. Hardware layout scheme.

The area located directly on the gas generator. The Hall sensor is located under the cover for loading fuel, a permanent magnet is installed on the cover opposite the sensor. When the fuel cap is opened, the sensor lets the system know when to stop the air supply, which is used to maintain the pyrolysis process. The thermocouple takes temperature readings directly in the pyrolysis zone, this information makes it clear when the system enters the operating mode and the gas generation process has begun. A centrifugal fan pumps air into the boiler during system startup to quickly enter the operating mode (figure 1, block 3).

Cyclone filter tank for large particles. After calibrating the system, it will be possible to determine what period of time is needed to fill the tank and promptly notify the user that it is necessary to release it from the contents (figure 1, block 4).

Cooling radiator. In the cooler, the generated gas cools and begins to occupy a smaller volume, and water vapor, which was previously the water contained in the fuel, began to condense in the cooling system. In order to prevent the radiator from filling up with water, the solenoid valve opens in a timely manner and water flows into a special tank. A float sensor is installed in the water tank, which changes electric resistance depending on the water level, on the basis of which the system promptly notifies that the tank should be vacated (figure 1, block 5).
Remote control device. This device is any device with a wireless Bluetooth module that runs on the Android operating system. The device communicates wirelessly with a microcomputer, after which it has the opportunity to receive information about the system and manage it (figure 1, block 6).

3. Design and development of a control system
To create a control system, it is necessary to carry out design to determine the internal logic of programs, their interaction and presentation of data, as well as external properties and methods of interaction.

The class diagram occupies a major place in the design of information systems where various objects and components are used [9]. The class diagram of the developed control system is a set of static model elements (figure 2).

**Figure 2. Class diagram.**

Pairing Form – class of the pairing form object. The pairing form contains information about the Bluetooth devices available for connection, the connect and cancel buttons.

Info Form – class of the information form object. The information form contains information display objects of the textView type, which display the state of the system in text form.

Control Form – class of the control form object. A control form consists of controls and displays elements, such as textView, Switch and SeekBar.

Bluetooth Client and Server – classes of objects that provide buffering, encryption and data transfer.

Interaction Interface – class of the object of interaction between various system interfaces.

Serial Port – the serial port object class, which provides reception and transmission of telemetric information from objects of the Device class.

Device and Sensor – objects of these classes contain read and write methods for controlling and feedback from physical devices of the control system.
The sequence diagram shows on a single time axis the life cycle of any particular objects and the interaction of actors with them within a specific precedent. The sequence diagrams of the main scenarios of the user’s work with the system were designed (figure 3).

![Sequence diagram of the process of managing system devices.](image)

**Figure 3.** Sequence diagram of the process of managing system devices.

The control system has a physical distribution of software modules. The embedded DBMS (SQLite) works with a table that stores data on the state of the system for a certain period of time with a given recording step. Table 1 presents the structure of the stored data.

**Table 1.** Telemetry table structure.

| Property          | Data Type | Description                                 |
|-------------------|-----------|---------------------------------------------|
| Time              | Integer   | Table write time                            |
| Temperature       | Real      | Sensor temperature readings                 |
| Fan_Speed         | Integer   | Current centrifugal fan speed               |
| Cond              | Integer   | Condensate filter fill status               |
| Cond_Valve        | Integer   | Condensate drain valve status               |
| Cap               | Integer   | Fuel tank cap status                        |
| Fuel              | Integer   | The current fuel weight in the fuel tank    |

Layout of the user interface is carried out in the Android Studio IDE [10]. On each of the application screens, two LinearLayout objects with a vertical orientation were used, which allows us to place all the elements for display and control in two columns (figure 4,5).

The next step in creating a control system is firmware for the microcontroller using programming and debugging in the Arduino IDE development environment. The program is written in a procedural style and consists of the setup and loop functions. The setup function describes the initialization of the microcontroller: binding devices and sensors to specific data ports, as well as determining the purpose of each port used. The loop function describes a repeating sequence of commands that provide reading from sensors, writing to devices, and transmitting this data in two directions through a serial port.
The execution speed of the microcontroller firmware was tested. The Arduino Nano microcontroller has an eight-bit ATmega328 microprocessor with a clock frequency of 16 MHz. Table 2 shows the execution time of instructions in processor cycles and microseconds.

| Command name   | Runtime in processor cycles | Runtime in microseconds |
|----------------|-----------------------------|-------------------------|
| Write 1 byte   | 2                           | 0.125                   |
| Loop (int)     | 6                           | 0.375                   |
| pinMode        | 5                           | 0.3125                  |
| analogRead     | 1792                        | 112                     |
| analogWrite    | 144                         | 9                       |
| Serial.write   | 16000                       | 1000                    |
| Serial.read    | 24000                       | 1500                    |
| Serial.flush   | 80                          | 5                       |
| Serial.available | 80                      | 5                       |

The table is based on a sketch for testing the speed of command execution. As a result of the test, it was revealed that this time is enough to read and write microcontroller data with an interval of 5 seconds, it is with this interval that telemetry will be recorded and read from the database.

4. Conclusion

As a result of the work, an Android application was developed for remote control of the vehicle’s gas generating equipment control system, an interaction application that provides storage and transmission of control and telemetry information between the Android application and the microcontroller, and microcontroller firmware that provides direct control of the gas generating equipment.

Using an automated fuel system, it is planned to popularize environmentally friendly types of energy resources, develop a market for these resources and increase the share of their use in domestic needs.

The developed control system satisfies all the technical requirements set before it and has the potential for further development.
References
[1] Reed T and Das A 1998 Handbook of Biomass Downdraft Gasifier Engine Systems (Colorado: Solar Energy Research Institute) p 148
[2] Davidson P and Lucaw T 1978 The Andco-Torax High Temperature Slagging Pyrolysis (Washington: Systemin Solid Wastes and Residence Conversion by Advanced Thermal Processes) p 197
[3] Foley G and Barnard G Biomass 1983 Gasification in Developing Countries (London: Earthscan Techn. Rept) chapter 1 p 153
[4] Emrich W and Reidel D 1985 Handbook of Charcoal Making (Boston: of Solar Energy in the European Community) p 297
[5] Tokarev G Gas-generating cars 1955 (Moscow: State Scientific and Technical Publishing House of Engineering Literature) p 207
[6] Simon M 2019 Electronics. Recipebook: off-the-shelf solutions based on Arduino and Raspberry Pi (Saint-Petersburg: Dialectics) p 480
[7] Petin V 2016 Arduino and Raspberry Pi in Internet of Things (Saint-Petersburg) p 320
[8] Petin V 2017 The practical encyclopedia of Arduino (Saint-Petersburg) p 152
[9] Arlow D 2007 UML and the Unified Process. Practical Object-Oriented Analysis and Design (Saint-Petersburg: Symbol Plus) p 624
[10] Daytel P, Daytel H and Wald A 2016 Android for developers (Saint-Petersburg) p 512