Computer Applications in Electrical Engineering
Vol. 14 *2016*
DOI 10.21008/j.1508-4248.2016.0043

Driver's interface of the racing car

Jarosław Jajczyk, Mateusz Słomczyński
Poznań University of Technology
60–965 Poznań, ul. Piotrowo 3A, e–mail: Jaroslaw.Jajczyk@put.poznan.pl, slomczynski.mateusz@o2.pl

The paper presents the design of the driver's desktop for the racing car. The first part of the article reviews the available interfaces of the driver used in the racing cars. The examples of steering wheels with the implemented systems that enable the driver to control the selected parameters of the vehicle were presented. The necessary components of the driver's desktop were set out, which should be applied in the designed system. The second part of the paper presented the idea schemes of the electrical circuits of the applied modules. The focus was mainly on the elements that enable the information to the driver on the current parameters of the car. The support for the LED and LCD displays and the UART module designed for communication were implemented. The interface presented in the paper was developed for the purpose of using it in the racing car of the Formula Student class.

KEYWORDS: driver's interface, dashboard, racing car, electronic circuit design

1. Introduction

In motor sports, what is very important is the driver's awareness on the current operating parameters of the racing vehicle. It also plays an important role in the safety of the driver and people in the vicinity of the vehicle. The person controlling the car should be informed in the case of, e.g., the exceeded allowable temperature of the oil or coolant, so that he can adequately react.

The driver's interface is constructed by each team individually. The solution is adapted to the needs of the entire team, and especially to the driver. This is most often done through electronic systems implemented in the steering wheel of the racing car. The automotive industry offers the available finished systems used as the driver's desktop. They offer many functions and are flexible in terms of the displayed parameters, however, the commercial solutions do not always meet the requirements of the racing teams. The ready solutions rarely have all required modules or systems, which are needed by the team. That is why many teams opt for the individual development and performance of the driver's desktop. This solutions enables the complete adaptation of the project to own needs and assumptions. Thanks to the independent performance of the driver's interface, the team is able to adjust the shape and dimensions to the space available in the vehicle. This allows for the selection of only the necessary modules and electrical devices for the implementation of the assumed functions, limiting the amount of the additional, unnecessary systems.

The paper presents the design of the driver's desktop for the needs of the team representing the Technical University of Poznan in the racing class of Formula Student.

2. Characteristics of the driver's interface

The driver's interface in the racing car is to enable the control and change of the settings of the vehicle. This is accomplished by collecting the necessary data and its presentation in different ways on the steering wheel. The electronic part of the driver's desktop consists of the plate of insulating material with electrical connections. It has a corresponding number of buttons, switches, diodes and displays. The driver during the race must focus the entire attention on running the car and does not have the time to analyse whether, e.g., the current gear is adequate. For this purpose the driver's desktop was created, usually implemented in the steering wheel. The data from all sensors mounted in the vehicle and from the engine's driver come to it. The main control computer, which is usually the microcontroller, analyses the data and based on the required programs controls the elements attached to it [1].

The racing car has got a lot of parameters, which can be changed in order for a better adaptation of the car to the drive. This can be changed each time during the stay in the box, but due to the time and convenience, it was decided to be enabled to the driver without the need to take off the trail, and even during the race. It is a solution which allows the driver to change and test different setting and characteristics of the car without the unnecessary need to take the eyes off the track – provided that he knows the steering wheel well as well as the location of knobs and buttons. The steering wheel also has got additional buttons installed, which activate special functions during the race, such as, e.g., the KERS system (Kinetic Energy Recovery System) or DRS (Drag Reduction System) [2].

The driver's desktop is designed primarily to provide the driver – in the simplest and the most transparent manner possible – the necessary information, based on which the driver can make a quick decision. In addition, in the professional Formula 1 racing cars, the driver can also control various parameters in the car and change the individual settings [1, 10].

Most racing teams have got their own individual driver's desktop. The number of buttons and knobs depends on the requirements of the team. The location and functions of the individual interface components are most often consulted with the driver, because the desktop is designed for him. In cases, when the team has got several drivers, different configurations of steering wheels with the implemented interfaces may occur. The accurate number and functions of the buttons, knobs, switches and displays on the desktop should be selected specifically for the given driver, because a universal solution is inefficient [9]. Figure 1 shows an exemplary driver's interface implemented in the steering wheel.

Fig. 1. View Racing Wheel integrated with the dashboard [3]: 1 – BOOST, 2 – LAP TIME, 3 – HARVEST, 4 – DOWNSHIFT PADDLE, 5 – MIX. 6 – BITE POINT, 7 – BPF, 8 – CLUTCH PADDLE, 9 – BBAL, 10 – REVERSE GEAR, 11 – SHIFT LIGHTS, 12 – LIMITER, 13 – ENGINE PARAMETERS, 14 – UPSHIFT PADDLE, 15 – TORQUE, 16 – TYRE, 17 – CLUTCH PADDLE, 18 – DIFFERENTIAL

The ready–made solutions are also available, which are offered by various manufacturers. One of them is the desktop MDU230 of the MagnetiMarelli company, presented in Figure 2.

Fig. 2. The device MDU230 company Magneti Marelli [4]

It is a combination of the dashboard with the modular inputs for controlling both the information display and the parts of the complete system for obtaining and monitoring data to be used in the motorsport. This device is relatively small, so it is ideal for use in the racing car, where the space is limited. MDU230 is equipped with a wide range of analogue and digital inputs. The bar graph is usually used to visualise the rotation of the engine. On the display, some fields are pre–assigned to display certain parameters, e.g.: the current run or time and the number of laps. It also has configurable fields, in which you can define the data to be displayed on your own. This device is equipped with 11 different screens, wherein different parameters can be shown on each screen. If the system was to work as a part of the system for obtaining and monitoring data, it can communicate with other devices using the CAN network [4].

To sum up, the type of a driver's interface is to be found in each racing car depends on the team – primarily on the common idea, technical skills and financial abilities of the given racing team.

3. Design of the driver's interface

3.1. Design assumptions

The aim of the project is to implement the electronic system used as the driver's interface for the racing car. The electronic system in the racing car will serve as the dashboard, which should contain the following information:

- current value of the vehicle's speed,
- current value of the rotation engine speed,
- light signalling of the engine speed,
- light signalling of the states of emergency.

The designed system should be universal, so that it can be easily reprogrammed and adapted to the current needs. For the driver during the race what is important is the speed and rotations, while during the tests he will focus on the operating parameters of the engine. Therefore, it should be possible to reprogram and reconfigure the system.

3.2. Computer controlling the dashboard

In the project used, all elements of the driver's desktop are integrated together using a single device. This enables the control of all elements of the driver's interface with one overarching system. The main unit of the driver's interface is the control computer (microcontroller) equipped with all required modules, which are required for the implementation of the assumed functions. It is also equipped with the appropriate interfaces used for communication. The use of a microcontroller with the CAN controller in one system reduces the amount of the systems and allows the limitation of the dimensions of the entire driver's desktop. Given the above assumptions, the AT90CAN32 microcontroller of the Atmel company was selected as the control computer (Fig. 3).

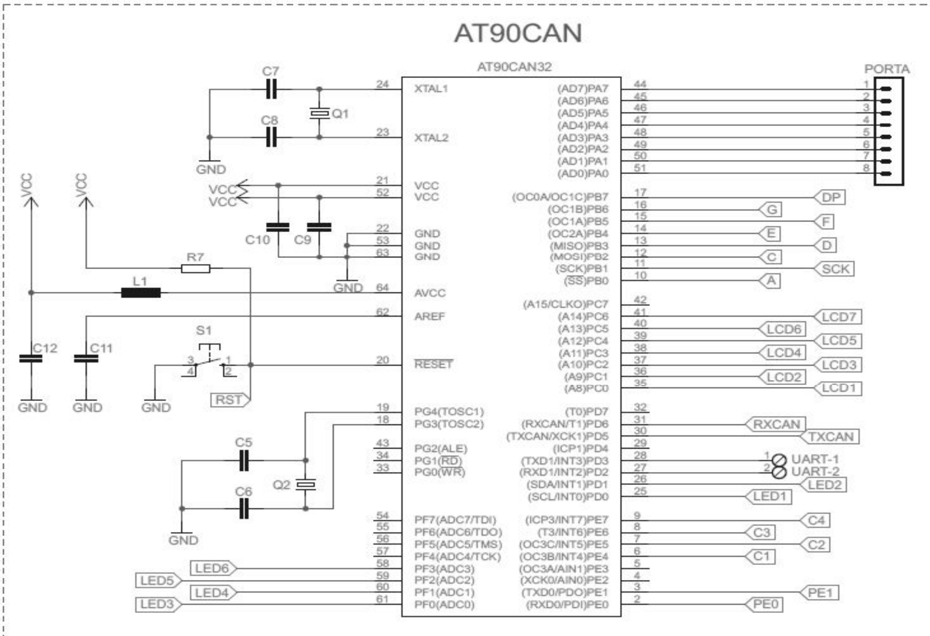

Fig. 3. Schematic diagram of connecting the microcontroller AT90CAN32

For the proper operation of the microcontroller, each VCC leg received the power of +5 V additionally filtered with ceramic capacitors. The manufacturer of the microcontroller recommends to connect the power of the AVCCC leg through the low–pass filter LC. GND pins were connected to the system mass. Because the system will not use the external reference voltage, the AREF leg was connected through the 100 nF capacitor to the system mass. The XTAL1 and XTAL2 pins of the microcontroller are used for the connection of the external quartz resonator Q1. The system connected the external resonator with the frequency of 16 MHz. The outputs of the resonator were connected to the mass through a pair of capacitators C7 and C8 with the value specified in the catalogue card of the microcontroller and it amounts to 22 pF. The microcontroller used has got its own external oscillator with the frequency up to 8 MHz, but due to the need to perform a large number of operations in the shortest time possible, the external resonator will be used in the system. The quartz generator Q2 is also connected to the system, as a watch quartz, used mainly as the source for timing for the timer – it is used for the precise timing. The S1 button is used for resetting the microcontroller and it is connected to the

RESET pin. It is required to connect the RESET leg, through the resistor of 10 kΩ, to the VCC supply to force the high state on this pin. This connection is necessary for the proper operation of the microcontroller [8, 11].

Table 1. Value of the components of the computer control

Description	Symbol	Value
Ceramic capacitor	C5–C6	15 pF
Ceramic capacitor	C7–C8	22 pF
Ceramic capacitor	C9–C12	100 nF
Quartz resonator	Q1	16 MHz
Quartz resonator	Q2	32,768 kHz
Resistor	R7	10 kΩ
Choke	L1	10 μH

The suitable microcontroller ports were assigned with the following functions:

- PORTA (PA): a module of free terminals used for the potential extension or connection of additional systems,
- PORTB (PB): it is used to operate the 7–segment LED display, namely to control the display cathodes, that is lighting the respective segments. The PB1 pin also has got an alternative function used during the microcontroller programming.
- PORTC (PC): it is used to operate the LCD display – power supply, control lines and data lines connected to this port.
- PORTD (PD): two youngest port bits are used as the outputs of the diodes signalling the alarm states occurring in the racing car. The PD0 and PD1 pins are the outputs of the CAN controller, so they are connected to the trans–receiver in order to enable the communication with the CAN interface. The PD2 and PD3 pins are the free outputs and their alternative functions include the possibility to generate the external interruptions or the additional serial interface UART.
- PORTE (PE): two youngest bits perform the function of the serial interface UART and are connected to the integrated system MAX232 in order to communicate with the motor controller. They are used during the update of the microcontroller program (while programming). Four oldest bits of the port are used to operate the 7–segment LED display and are responsible for the control of the display anodes (they control the displayed numbers).
- PORTF (PF): it is designed for the light signalling of the engine speed using the connected LED diodes.

The AVR microcontroller is programmed using the ISP interface (In–system Programming Interface), which allows the reprogramming of the driver's desktop after implementing it in the vehicle [6].

3.3. Communication module

The driver's interface displays the current operating parameters of the racing car, which are supplied by the control unit. Communication between the engine controller and the interface is implemented through the appropriate communication module enabling the data transmission. The system was equipped with the ability to communicate through the RS232 interface and the CAN interface. This way, the greater versatility of the implemented project was provided. For this purpose, two integrated systems were selected, used for communication:

– MAX232 – RS232 interface,
– L9616 – CAN interface.

The schematic diagram of the system used for communication using the RS232 standard is show in Figure 4.

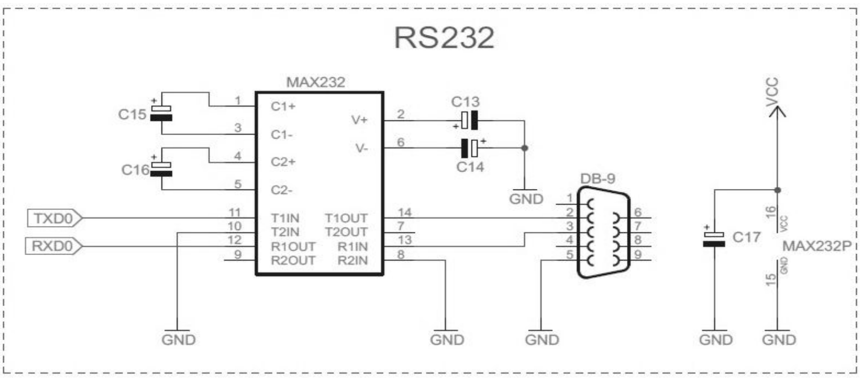

Fig. 4. Schematic diagram of the RS232 communication module

The desktop layout is connected to the engine controller through the RS232 connector. The 2 and 3 pin of the DB–9 connector are the TxD signals (responsible for the data transmission) and RxD (responsible for the data reception). The TxD and RxD signals cannot directly go to the UART controller in the microcontroller, because the standard of RS232 voltage is different from the level of voltage in the controller [5]. Therefore, we must use the proxy system, which is the integrated system MAX232 enabling the mutual data transmission. Its task is to convert the voltage levels from the RS232 standard to the TTL standard, which is used by the UART controller in the microcontroller. The T1IN and R1OUT pins of the MAX232 system are connected to the UART controller in the microcontroller.

The second alternative communication module is the integrated system L9616 that allows communication through the CAN network. Figure 5 presents the schematic diagram of the communication module.

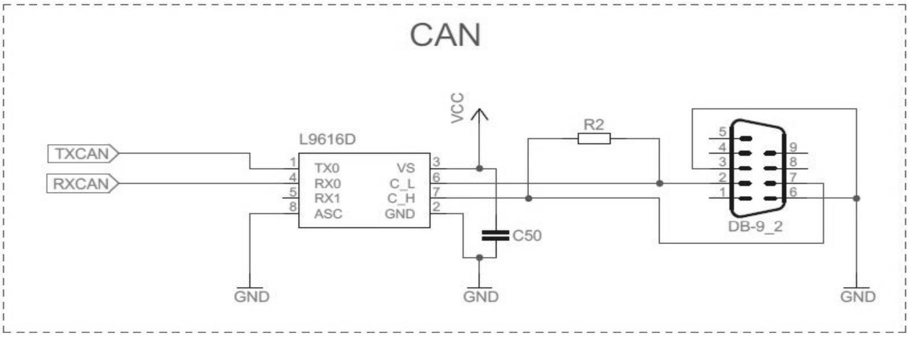

Fig. 5. Schematic diagram of the communication module CAN

The signals of the CAN system are received and sent in a similar manner to the RS232 interface, that is through the nine–pin connector for data transfer. The 3 and 6 pins are connected with the system mass. On the 2 and 7 terminals the signals are sent, respectively, CAN_Low and CAN_High. These signals pass from the connector to the CAN transceiver (that is the L9616 system) to the C_L and C_H pins. The transceiver converts the differential signal into the zero–one signal, and then, using the TX0 and RX0 pins it sends the signals to the CAN controller built–in to the microprocessor. In addition, the signal lines CAN_L and CAN_H are connected together with the terminating resistor R2 with the value of 120 Ω. The use of the terminating resistor protects the main line against the signal reflections [7].

3.4. Display of the current operating parameters of the racing car

The two–line alphanumeric LCD display is a device for displaying the vehicle speed and the engine rotation. This system has been used due to the simple construction and reliability. The display is sufficient to show the assumed information and it performs its function in a manner, which is transparent to the driver. One line can fit 16 characters, so there is quite a lot of space to display various operating parameters of the vehicle. Figure 6 presents the schematic diagram of the display.

The sample code of the program used for operating the display was presented below:

```
char tab1[] = {"Speed:"};
int a = Speed;                      // from UART
char tab2[] = {"Engine speed:"};
int b = Engine_speed;               // from UART

Int main(void)
{
   LCD_inicjalizacja();             // initialization LCD
   while(1)
```

```
{
   LCD_kursor(0,0);
   LCD_lancuch_znakow(tab1);
   LCD_kursor(0,10);
   LCD_liczba(a);
   LCD_kursor(1,0);
   LCD_lancuch_znakow(tab2);
   LCD_kursor(1,10);
   LCD_liczba(b);
}
```

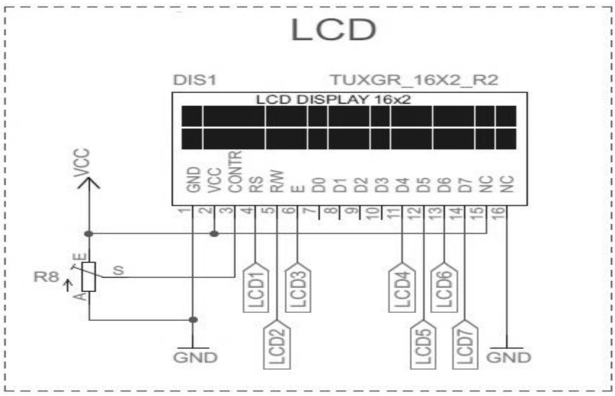

Fig. 6. Schematic diagram of the LCD display

The quad 7–segment LED display was designed for the indication of the current gear of the racing car. The information about the course is displayed on one segment of the display. In order to enhance the universality of the proposed solution the four–segment display was used. Figure 7 presents the scheme of the system.

Unexpected failures are frequent in the racing car. The driver should be kept informed, when the operating parameters of the racing car are exceeded, which can lead to the damage of the vehicle. One of the most important parameters is the engine speed. With the increase of the speed the number of the LEDs lit on the driver's desktop should increase. This solution clearly informs the driver about the need to change the gear. The light indication of the current engine speed and the failure states was implemented using the LED diodes. Below we can see an example of the program code used to operate the module in question:

```
DDRF|= (1<<PF0) | (1<<PF1) | (1<<PF2) | (1<<PF3);
PORTF |= (1<<PF0) | (1<<PF1) | (1<<PF2) | (1<<PF3);

if (Engine_speed >4000)
{
   PORTF &=~ (1<<PF0); // 1 LED
   if (Engine_speed > 5000)
   {
      PORTF &=~ ((1<<PF0) | (1<<PF1)); // 2 LEDs
```

```
        if (Engine_speed > 6000)
        {
           PORTF &=~ ((1<<PF0) | (1<<PF1) | (1<<PF2)); // 3 LEDs
           if (Engine_speed > 7000)
           {
           PORTF&=~ ((1<<PF0) | (1<<PF1) | (1<<PF2) | (1<<PF3));// 4 LEDs
           }
           else PORTF &=~ ((1<<PF0) | (1<<PF1) | (1<<PF2)); // 3 LEDs
        }
        else PORTF &=~ ((1<<PF0) | (1<<PF1)); // 2 LEDs
    }
    else PORTF &=~ (1<<PF0); //1 LED
}
    else PORTF |= (1<<PF0) | (1<<PF1) | (1<<PF2) | (1<<PF3);
```

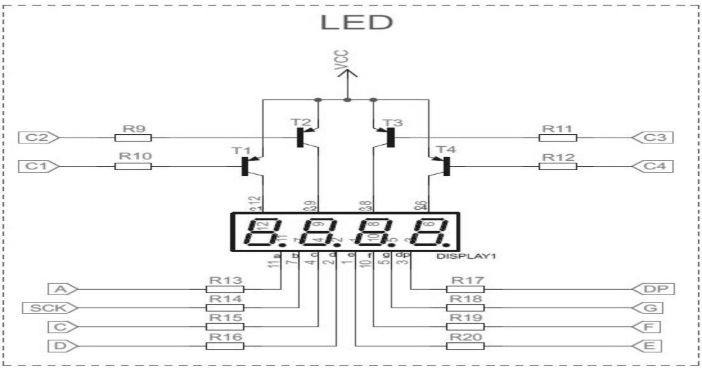

Fig. 7. Schematic diagram of the LCD display

The light signalling of the engine speed using LED diodes is an auxiliary module for the driver, which helps select the right moment to change gear, while signalling the faults plays an important role in the safety of the driver and his surroundings.

4. Conclusion

The presented design of the driver's interface is the part of the racing car of the Formula Student class. The system allows the present analysis of the current operating parameters of the racing vehicle during the competition and is the necessary equipment for the driver also during tests. The interface was equipped with all assumed elements used for displaying and signalling the operating parameters of the racing car. The applied components have been selected in a universal manner, so that they could be used for the performance of many tasks. The implemented project of the driver's interface is an essential element of the racing car. The experience gained during the design works will facilitate the construction of new driver's interfaces tailored to the needs of the racing teams.

References

[1] The Dashboard of an F1 car, http://www.f1technical.net/features/3492, 22.01.2016r.
[2] http://www.wired.com/2014/05/formula–1–steering–wheels/, 20.01.2016r.
[3] The Insanely Complicated F1 Steering Wheel, http://www.popularmechanics.com/cars/a6707/the–insanely–complicated–f1–steering–wheel/, 20.12.2015r.
[4] http://www.magnetimarelli.com/business_areas/motorsport/data–acquisition,–display–&–lap–trigger/data–display#tab—2, 14.01.2016r.
[5] Kardaś M., Mikrokontrolery AVR. Język C – podstawy programowania, Wydanie 2, Atnel, Sczecin 2013.
[6] Francuz T., Język C dla mikrokontrolerów AVR, wydanie 2, Helion, 2015.
[7] Herner A., Riehl H–J., Elektrotechnika i elektronika w pojazdach samochodowych, wydanie 9, WKŁ, Warszawa 2012.
[8] Atmel, 8–bit Microcontroller with 32K/64K/128K Bytes of ISP Flash and CAN Controller AT90CAN Datasheet
[9] Steering wheel, http://www.formula1–dictionary.net/steering_wheel.html, 21.01.2016r.
[10] The Dashboard of an F1 car, http://www.f1technical.net/features/3492, 22.11.2015r.
[12] Zasilanie mikrokontrolera, http://mikrokontrolery.blogspot.com/2011/04/zasilanie–mikrokontrolera.html, 18.12.2015r.

(Received: 2. 11. 2016, revised: 25. 11. 2016)