Journal of Scientific Research & Reports

25(4): 1-10, 2019; Article no.JSRR.52953
ISSN: 2320-0227

Design of Automobile Digital Tire Pressure Detector

Yunyu Cao[1*], Jinjin Dang[1] and Chenxu Cao[1]

[1]School of Electrical and Information Engineering, Anhui University of Technology, Anhui, China.

Authors' contributions

This work was carried out in collaboration among all authors. Author YC designed the study, performed the statistical analysis, wrote the protocol and wrote the first draft of the manuscript. Authors JD and CC managed the analyses of the study. Author YC managed the literature searches. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JSRR/2019/v25i430190
Editor(s):
(1) Assoc. Prof. Dr. Suleyman Korkut, Department of Forest Industry Engineering, Division of Wood Mechanic and Technology, Duzce University, 81620 Beciyorukler Campus Duzce, Turkey.
Reviewers:
(1) Raheel Muzzammel, The University of Lahore, Pakistan.
(2) Jurandyr Santos Nogueira, Federal University of Bahia Ondina Campus, Brazil.
(3) Xingting Wang, Howard University, USA.
Complete Peer review History: http://www.sdiarticle4.com/review-history/52953

Original Research Article

Received 19 September 2019
Accepted 25 November 2019
Published 02 December 2019

ABSTRACT

According to vast statistics, more than half of traffic accidents are related to the failure of car tire pressure. This paper investigated the design scheme of the existing tire pressure monitoring system, then selected the appropriate sensor, converted the tire pressure signal into the corresponding voltage signal and outputed it. Finally displayed the measurement result by the digital display tube. The main problem to be solved is that selecting a suitable pressure sensor requires accurate measurement of the required pressure parameters; completing the conversion of the analog signal and the digital signal; setting an appropriate threshold value. When the value is exceeded, the audible and visual alarm signal is emitted. This design adopted the design scheme of direct tire pressure monitoring system. Firstly, the circuit diagram is designed in the Altium desigener software environment and the corresponding simulation experiment is completed. Then, according to a successful experimental scheme, the PCB circuit board is processed to complete the welding of the physical object. At the same time, writing the running program in the C language environment and use Keil software to burn the program into the single-chip microcomputer. Finally, the debugging and modification of the physical object of the single-chip microcomputer completed the expected function.

**Corresponding author: E-mail: 787056057@qq.com;*

Keywords: Car tire pressure; single chip; sensor.

1. INTRODUCTION

Shima, Takashi Provided is a tire air pressure transmission device configured to determine the rotational position of the tire air pressure transmission device based on a gravitational acceleration component of a centrifugal acceleration at the time of transmission of the tire air pressure information and to transmit in a wireless signal and at a prescribed cycle, tire air pressure information and tire air pressure transmission device rotational position information [1]. The Tire Pressure Monitoring System (TPM), referred to as TPMS for the first time in July 2001, appeared on the statement issued by the US Department of Transportation and the Highway Traffic Safety Administration (NHTSA) on the automobile safety legislation promulgated by the US Congress [2,3]. From then on, the automobile tire pressure monitoring system entered the automobile consumer market and gradually developed and improved officially. The car tire pressure monitoring system plays an important role in the safe driving of the car, the maintenance of the car performance, the extension of the service life and the increase of the vehicle shock absorption to improve the driving comfort of the car. Since China's automobile tire pressure monitoring system started relatively late, the automobile tire pressure detection system has only begun to appear in the Chinese market since 2004 [4].

According to the working principle of the automobile tire pressure monitoring system, it is mainly divided into two types [5,6]. The first one is the direct tire pressure monitoring device, the second is the indirect tire pressure monitoring system. The working principle of the direct tire pressure monitoring device is to display the air pressure of each tire by receiving a signal from a pressure sensor in each tire directly, it alarming when the tire air pressure is too low or leaks. The working principle of the indirect tire pressure monitoring system mainly relies on the function of measuring the tire speed of the brake system. Measuring the air pressure of each tire by measuring the rotation speed of each tire at the distance of the driving unit indirectly. This article is designed according to the direct tire pressure monitoring system as the main design.

2. OVERALL DESIGN OF THE SYSTEM

The TPMS system realized by this design is mainly divided into two modules: a tire pressure detection module that is installed on the automobile. It is responsible for monitoring the target data and a main control module for analyzing and outputting the result on the automobile dashboard. The tire pressure monitoring module consists of a sensor, a signal generator and a microcontroller, as long as the function is to collect the pressure signal and send them to the host module. The mainframe module is mainly composed of a signal receiver, a microprocessor, a liquid crystal display, an alarm and a button. Its main function is to receive the information from the tire pressure detecting module, analyze and judge the message, and make the detected signal appear abnormal. The alarm will be triggered to alert the user. When the system fails, the system can be restarted by pressing the reset button. The hardware diagram is shown in Fig. 1.

When designing the overall design of the software in the tire pressure system of the car, it is necessary to consider the requirements in the actual application in order to design a suitable software configuration. The block diagram is shown in Fig. 2.

As can be seen from the above figure, the tire detection module has three functional modules, namely, data measurement program, data processing, and transmission program.

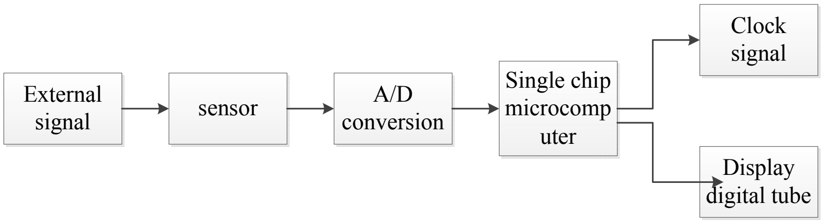


Fig. 1. Hardware design plan

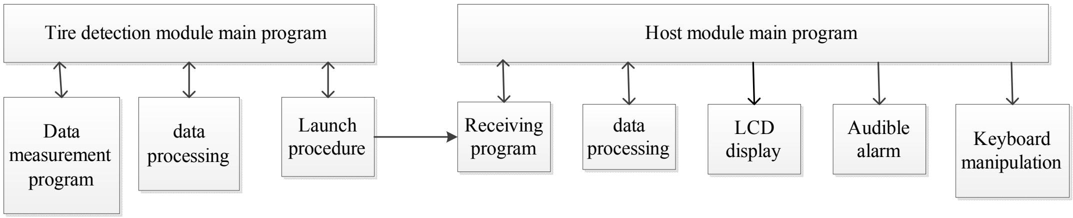


Fig. 2. Software design plan

3. SYSTEM HARDWARE DESIGN

3.1 Design of the Main Control Circuit

STC89C52 MCU is a new type of MCU introduced by STC to the market. It has strong anti-interference capability, high speed and low power consumption. Its instruction code is fully compatible with the traditional 8051 microcontroller, with 12 clocks, machine cycle and 6 clock, machine cycle 3 ways can be arbitrarily selected [7]. The MAX810 dedicated reset circuit integrated into the microcontroller is 6MHz and the clock frequency is 12MHz. The circuit diagram of the single chip microcomputer is shown in Fig. 3.

3.2 Design of Air Pressure Detection Circuit

The ADC0832 is an 8-channel resolution and dual-channel A/D analog-to-digital conversion developed by National Semiconductor Corporation [8]. Because of its small size, strong compatibility, high cost performance, etc., it is well received by the majority of MCU users and has a high penetration rate in the market.

The interface description of the ADC0832 chip is shown in Table 1.

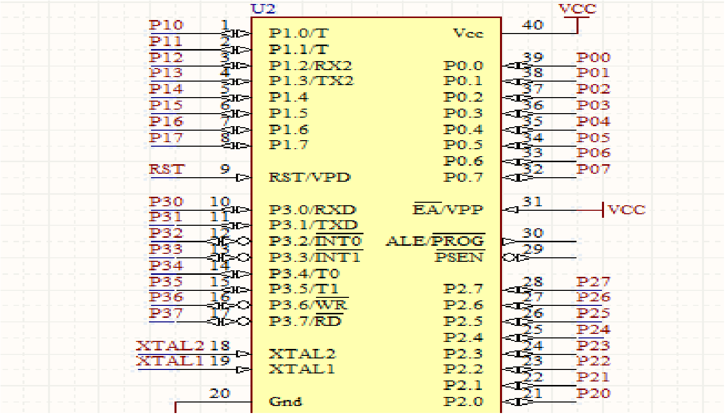


Fig. 3. Single chip circuit

Table 1. ADC0832 pin description

CS	Chip select enable, low level chip use.
CH0	Analog input channel 0, or used as Ｉ N+/.
CH1	Analog input channel 1, or used as Ｉ N+/.
GND	Chip reference 0 potential (ground).
D1	Data signal input, select channel control.
D0	Data signal output, conversion data output.
CLK	Chip clock input.
VCC/REF	Power input and reference voltage input (multiplexed).

二XGZP type piezoresistive pressure sensitive components

The tire pressure range of the car is generally not lower than 2.1bar and not higher than 3.0bar. This design selects XGZP type piezoresistive pressure sensitive component with a pressure measuring range of 0.1bar~7bar to fully meet the testing requirements of ordinary automobiles. XGZP piezoresistive pressure sensitive component is a pressure sensor widely used in automotive electronics research [9], which is mainly based on a MEMS processed silicon piezoresistive pressure sensitive. Its working principle is an elastic aperture and four varistor on the diaphragm to share the Wheatstone bridge structure, so that once the pressure acts on the elastic aperture, a voltage proportional to the pressure will be generated on the bridge.

The XGZO pressure sensor is packaged in standard SOP6 and DIP6 styles. It is installed in a dual in-line style with good linearity, repeatability and stability [10].

The main function of the air pressure detection module is to measure the air pressure value of the research object. After processing by the microcontroller, the signal is sent to the digital-to-analog converter, converted into a digital signal and then sent to the single-chip microcomputer for processing. Therefore, the signal detection circuit is mainly composed of an XGZP pressure sensor and an ADC0832 digital-to-analog converter. The detection circuit is shown in Fig. 4.

3.3 Display Circuit Design

There are two ways to illuminate the display: dynamic lighting and static lighting. The so-called dynamic lighting display is to control a specific display to continue to be in a constant on or off state to display a certain character. Because this method requires an 8-bit output port for each bit, there are too many I/O ports required for this mode when there are many bits. In this case, dynamic display can be used. The so-called static lighting display is to cycle through the various bits of the display (cycle for each bit cycle), so that for each bit there is a lighting cycle and the brightness is proportional to the current passing through. This design adopts the dynamic point mode. The MCU P00~P07 ports connect a~f7 LEDs and P24~P27 connect 4 TTL circuits to control the current through the diode respectively. The display circuit is shown in Fig. 5.

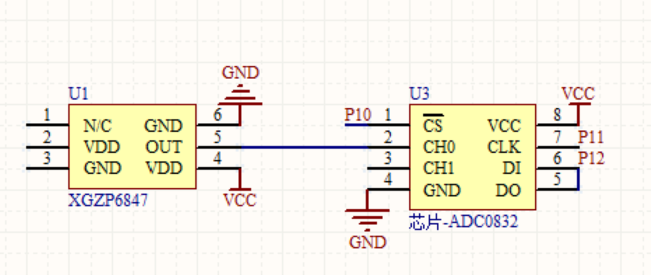


Fig. 4. Air pressure detection circuit

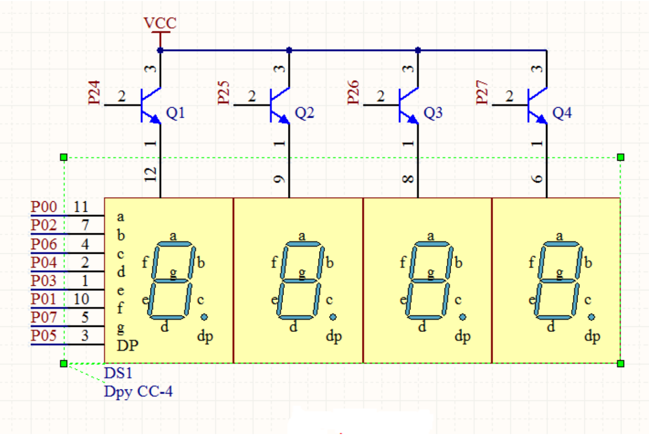


Fig. 5. Digital tube display circuit

3.4 Hardware Production Process and Results Display

After selecting the appropriate components, you need to draw the circuit diagram in the Altium designener software environment. Using Altium Desigener's function of performing circuit simulation, the circuit simulation was carried out. After some debugging and improvement, the printed circuit board (PCB) file was started, and the fabricated PCB file was sent to the manufacturer who made the printed circuit board to produce the finished circuit board. Finally, the selected components are soldered to the circuit board in turn according to the design scheme of the circuit diagram to complete the physical production.

3.4.1 The specific design and results of the circuit schematic

First open the DXP program and enter the Altium designener software environment.

(1) Design concept/Sspecification determines the design concept.

(2) Create PCB project to create a PCB project mouse left click on "File" Click "New" in the pop-up menu and then find "Project" in the pop-up selection board and select PCB Project. After creating a new project, click Save project and select the appropriate hard disk location in the computer named "Bi she PCB". Click "Save" to save the project.

(3) Add sheets/Sheet symbols to bulid design hier airclry to add a workbook and set the level of work, click "File" and select "New" to continue to select "Schematic" to create a new schematic work plane. After creating the work plane, click “Desigen” and select Options. In the pop-up interface, click “Sheet Options” in the upper left corner and select the paper size in “Standard styles”. The design selects “A4” and clicks “OK”. Complete the drawing settings.

(4) Fied and plce components from libraries are components. Clicking on “libraries” in the lower left corner of the SCH design interface will bring up the component library dialog box. Then enter the target component code into the space on the third line and click “Search” to start searching for the target in the component library. When there is no target component in the component library, you need to manually add a new component library. Click “libraries” to find the Orderde List of installed libraries. Find the downloaded component library. Click “Move up” and click “Add Library” at the bottom. "Complete the addition of the component library.

(5) Wire design is wiring. Auto-wiring click “Auto Route” in the PCB operation interface. In the pop-up dialog box, select “Setup” to adjust the setting properties in the automatic routing settings (this design can be set with the default settings). Click “OK” to complete the automatic routing. You can also choose manual routing, click “View” and select “Toolbars”. In the pop-up dialog box, click “Placement” to open the component placement toolbar, and select the wiring shape you need in the toolbar to complete the wiring.

(6) Annofate design is annotated. The comment is for the user to be able to clearly calculate the properties of the components in the circuit or the required wire or material for the printed circuit board manufacturer. If you click "Place", a dialog box will pop up. These include Coordinate (coordinate attribute), Dimension (distance label), Polygin Plane (copper) and so on.

(7) Compile and verifydesign is compiled and verified. Click “Tools”. Clicking Design Rule Check in the displayed column will pop up the design planning check option, click “Run Design Rule Check” to check, and the system will automatically give the Messages message box will be marked with green font. An error occurred and the generated inspection report file was suffixed with DRC.

3.4.2 PCB circuit specific design and results

The design of the PCB circuit needs to rely on the schematic of the previous step, encapsulating the schematic diagram and generating the netlist and loading the netlist to complete the step by step.

(1) Establish a SCH document. In the main menu, click "File" and select "New". Click "PCB Project" and click "Schematic" under the project file to create the SCH schematic.

(2) Define the wiring board layer and the non-electric layer. In the main command menu, click “Design” to select Layer Stack Manager. In the pop-up dialog box, set the “Signal layer” and “Internal Planes” on the board to the signal layer and inner layer.

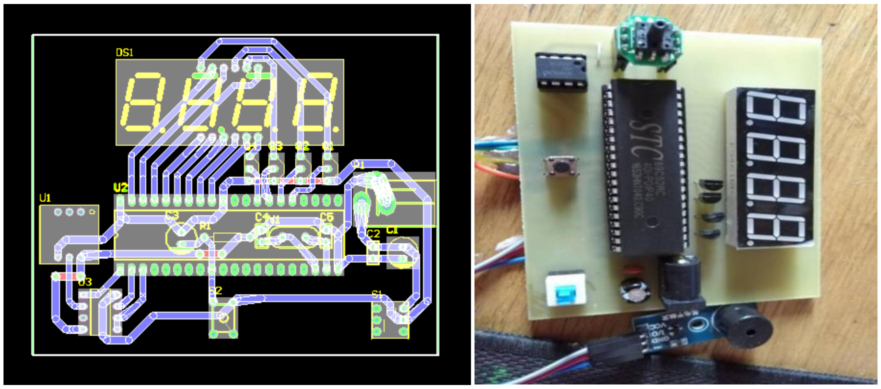


Fig. 6. PCB circuit diagram and physical map

(3) The package of components. Click "Footprints" in the lower right corner of the dialog box in the schematic design to define the package information for the component.

(4) Generate a netlist. In the main menu, click “Design” and select “Netlist” to continue to select Protel to generate a file with the same name as the source file and suffix .net, and define the netlist format for the component.

(5) Update the PCB board. Open the CLK.PCBDOC file in the SCH compilation environment. Click "Design" to select Update PCB CLK.PCBDOC in the pop-up dialog box to modify the corresponding state (component, netlist, space) to complete the modification and click Execute Changes, the system will perform the modifications. Finally, click Export in the lower left corner of the dialog to save the changes and generate a file in Excel format.

Finally, check and compile according to the schematic design to complete the final steps. After obtaining the PCB circuit board, the components are connected together according to the simulated circuit by means of a welding torch, a wire, a package glue, etc., to complete communication. The PCB circuit diagram and physical results are shown in Fig. 6.

4 SYSTEM SOFTWARE DESIGN

In this design, Altium Desigeners is used to assist in the hardware circuit fabrication and simulation. Finally, using the Keil programming software, the compiled C language program is burned into the MCU, and then the above operations are repeated step by step to debug and improve the control program to complete the entire task.

4.1 Software Environment Introduction

Altium Deigeners, also known as "Protel DXP", Protel DXP has powerful and complete design features that meet the auxiliary requirements of today's designers. These include circuit principle design, schematic editing, PCB circuit devise, routing rules setup, report file generation, hierarchical circuit diagram design, and EPGA program design. PCB circuit design flow chart is shown in Fig. 7.

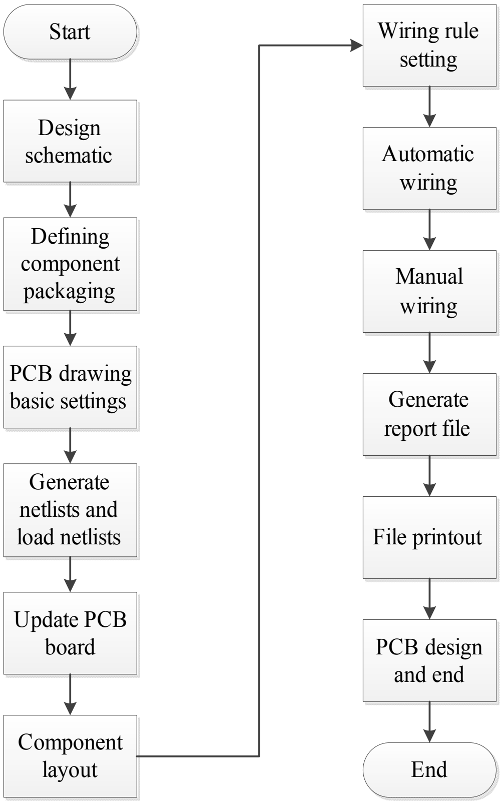


Fig. 7. Design flow chart of PCB circuit board

4.2 System Program Flow Chart

This design is based on the TPMS system of STC89C52 single-chip microcomputer. Firstly, it must initialize the single-chip microcomputer, A/D conversion, alarm system and pressure sensor. Then input the tire pressure analog signal to the XGZP type pressure sensor. Then under the control of the short set, the ADC0832 performs A/D conversion to output the digital signal to the LED digital tube, and judges whether the alarm system alarms according to the preset upper and lower limits. In summary, the system control program includes: a main control program, an A/D digital-to-analog conversion program, an LED display program, and an audible and visual alarm program.

4.2.1 Main program flow chart

The main control program description: STC89C52 as the main control core of the system, need to participate in the implementation of a lot of data processing and analysis, including control A/D digital-to-analog conversion, when the pressure exceeds the upper and lower limits, the control alarm system sends out an alarm signal. The main control program flow is shown in Fig. 8.

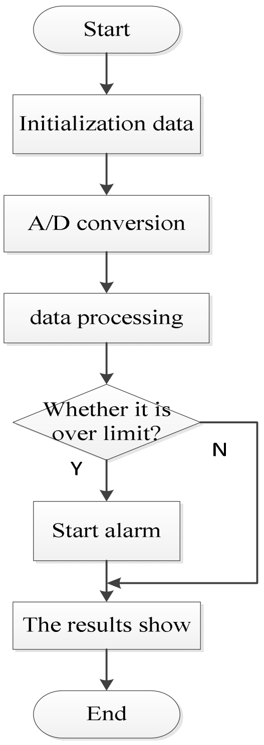


Fig. 8. Master program flow chart

4.2.2 A/D conversion program flow chart

A/D conversion flow chart description: After the main program initialization is completed, we need to define and initialize the chip signal, then open the read signal, set the microcontroller read amount, finally resume reading the input signal and close the chip select signal to the next cycle.

4.3 Calculation Process of Tire Pressure

Firstly, the XGZP type pressure sensor measures the target air pressure and converts it into a voltage signal output. After consulting the data of the XGZP type pressure chip, the relationship between the output voltage and the measured air pressure can be obtained:

$$V=VCC(0.004P-0.04)\pm error \qquad (1)$$

Where VCC=+5.1V voltage, error is the error factor, so

$$V=5.10\ (0.004P-0.04) \qquad (2)$$

Note: P is the pressure value unit in KPa.

In addition, the voltage outputted by the pressure sensor is converted into a corresponding digital quantity by A/D, wherein the relationship between the input voltage V1 and the output number two D1 is as follows:

$$V1/5=D1/255 \qquad (3)$$

Among them, "5" and "255" are calculation factors obtained by referring to the conversion factor given by the ADC0832 digital-to-analog conversion chip.

4.4 Program Compilation and Debugging Results

4.4.1 Compiling and debugging of the main program

First call the 52 MCU, ADC0832 digital-to-analog converter, digital picture tube and timer related functions at the beginning of the main program.

```
#include <reg52.h>
  #include "ADC0832.h"
#include "DpyX4.h"
#include "Timer0.h"
```

Then define the data to be read in

```
sbit Speaker=P3^7;
unsigned char MS100_Con=0;
unsigned char MS100_Change=0;
unsigned char ADC_Value=0;
unsigned int Press_Data=0;
```

Initialize the timer

```
Timer0_Init();
LED_View_Data[0]=22;
LED_View_Data[1]=22;
LED_View_Data[2]=22;
LED_View_Data[3]=22;
```

When the timer starts, the pressure detection starts. When the changed pressure value is “0”, the voltage signal is collected. If the electrical signal is less than 0.5V, indicating that no pressure acts on the pressure sensor, the output pressure value is 0.1 bar or the standard atmospheric pressure value.

```
MS100_Change=0;
ADC_Value=Read_ADC0832(0);
ADC_Value=ADC_Value*1.961;
if(ADC_Value<50
Press_Data=100;
```

When it is detected that the pressure change signal is not "0", the pressure signal is redefined as the value obtained by subtracting 50 plus 100 from the voltage signal.

```
Press_Data=ADC_Value-50;
Press_Data=Press_Data+100;
```

Since the unit used to detect the tire pressure of the car is "bar" and 1 bar = 100 KPa, the calculated pressure is divided by 100 to obtain the correct tire pressure unit, and the display data is updated to output the result.

```
LED_View_Data[0]=23;
LED_View_Data[1]=Press_Data/100;
LED_View_Data[2]=Press_Data%100/10;
LED_View_Data[3]=Press_Data%10;
```

The main program ends. Copy the compiled program to the newly created project plane in the keil software and perform error checking as shown in Fig. 9.

No error is displayed. No warning indicates that there is no error in the programming language. Finally, the program is burned into the MCU for final debugging.

4.4.2 Compilation of the alarm program

This design alarm program is not written separately, but the alarm program is added while the main program is being written. First, define the alarm data as sbit Speaker=P3^7. When the pressure value exceeds the preset pressure threshold, the alarm will be activated, otherwise the alarm will be in the sleep state.

```
If (Press_Data>120)
Speaker=0;
else
Speaker=1;
```

The debugging of the alarm program is carried out together with the debugging of the main program above, and the specific debugging is not repeated here.

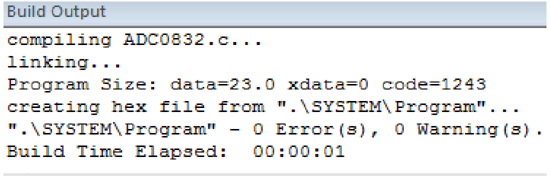
4.4.3 Preparation and debugging of common anode digital tube program

This design uses a common anode digital tube to show the measurement results, and the characters displayed by the digital tube need to be defined before the digital tube exhibit is called.

```
code unsigned char LED_Data[ ]=
    0xA0,  //"0"
    0xBB,  //"1"
    0x62,  //"2"
    0x2A,  //"3"
    0x39,  //"4"
    0x2C,  //"5"
    0x24,  //"6"
    0xBA,  //"7"
    0x20,  //"8"
    0x28,  //"9"
    0x30,  //"A"   10
    0x25,  //"B"   11
    0xE4,  //"C"   12
    0x23,  //"D"   13
    0x64,  //"E"   14
    0x74,  //"F"   15
    0x31,  //"H"   16
    0xE5,  //"L"   17
    0xB0,  //"n"   18
    0xA1,  //"u"   19
    0x70,  //"P"   20
    0x27,  //"o"   21
    0x7F,  //"-"   22
    0xFF  //Extinguished   23
```

Then set the initial values of the four digital tubes to zero.

```
    unsigned char DP1_Set=0;
    unsigned char DP2_Set=0;
    unsigned char DP3_Set=0;
    unsigned char DP4_Set=0;
```



```
compiling ADC0832.c...
linking...
Program Size: data=23.0 xdata=0 code=1243
creating hex file from ".\SYSTEM\Program"...
".\SYSTEM\Program" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:01
```

Fig. 9. Creating a target file map

Then use a switch statement to judge the input signals of the four digital tubes separately. Let's take the first display as an example:

```
switch(LED_Con)
 {
    case 0:
    LED_Pin=LED_Data[LED_View_Data[0]];
    if(DP1_Set)
    LED_DP=0;
    else
    LED_DP=1;
    LED_1=1;
    break;
 }
LED_Con++
```

When LED_Con=0, the first LED digital tube is detected. When the received signal is “0”, the loop ends. When the received signal is “1”, the value of the first digital tube is given “1”, and the process continues. The statement LED_Con++ continues to loop until the received DP1_Set=0 ends the loop.

After completing the compilation of all the programs, the error check is performed on all the programs. After the check is completed, the .HEX file is generated. Finally, the .HEX is burned into the MCU to complete the design of the software part.

4.4.4 Compilation and debugging of ADC0832 digital-to-analog conversion program

First, the read-in quantity "CH" of the ADC0832 digital-to-analog conversion and the first eight bits of the input data "dis0" are defined by the eight bits "dis1". Then pull down the clock signal, because the chip is disabled when the clock signal is in the high state, so before using the digital-to-analog conversion function of the chip, the clock signal needs to be pulled down, and this state is maintained until the end of the call.

```
    include "ADC0832.h"
    unsigned char Read_ADC0832(unsigned
    char CH)
    unsigned char i,dis0,dis1;
    unsigned int date;
    ADC_Clk=0;
    ADC_DI=1;
    _nop_();
    ADC_CS=0
```

The second step is the selection of the channel. The first two clocks are pulled down to the DI assignment to determine the channel selection. The third time the low clock DI is disabled, and D0 starts the data output.

```
if(CH==0)
  {
   ADC_Clk=0;
   ADC_DI=1;
   _nop_();
   ADC_Clk=1;
   _nop_();
   ADC_Clk=0;
   ADC_DI=0;
   _nop_();
   ADC_Clk=1;
   _nop_();
  }
  else
  {
   ADC_Clk=0;
   ADC_DI=1;
   _nop_();
   ADC_Clk=1;
   _nop_();
   ADC_Clk=0;
   ADC_DI=1;
   _nop_();
   ADC_Clk=1;
   _nop_();
  }
  ADC_Clk=0;
ADC_DI=1;
```

The third step reads the first 8 bits of the value, and if the last 8 bits are equal, the value is assigned to the digital signal and output. The output is pulled high to complete the conversion.

Software design should fully refer to the reference materials provided by the components used in the devise. This can reduce the cost of many incentives and practices. It only needs to call the component reference materials to successfully drive the components used. Some programs that work together after connecting can be used.

```
for(i= 0;i<8;i++)
   {
     _nop_();
     dis0<<= 1;
     ADC_Clk=1;
     _nop_();
     ADC_Clk=0;
     if (ADC_DO)
        dis0|=0x01;
    else
        dis0|=0x00;
   }
for (i=0;i<8;i++)
   {
     dis1>>= 1;
     if (ADC_DO)
      dis1|= 0x80;
     else
      dis1|= 0x00;
     _nop_();
     ADC_Clk=1;
     _nop_();
     ADC_Clk=0;
   }
           if(dis0==dis1)
             date=dis0;
   _nop_();
   ADC_CS=1;
   ADC_DO=1;
   ADC_Clk=1;
```

5. CONCLUSION

The tire pressure monitoring device designed to automatically monitor the tire pressure and provide a warning signal in real time when the tire air pressure is in an abnormal working state has important practical significance.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Shima T, Sakaguchi K, Terada S. Tire air pressure monitoring system [J]; 2017.
2. Regina Mathias, Ambresh P. Ambalgi, Anup M. Upadhyaya. Grating based pressure monitoring system for subaquatic application [J]. International Journal of Information Technology. 2018;10(8):551-557.
3. Moon CH. Implementation of a monitoring system for the measurement of temperature, flow rate, and fluid pressure of cooling systems [C]; 2018.
4. Waqas M, Khan A, Muhammad Rizwan, Lauri Sydanheimo. Effect of temperature variation on remote pressure readout in wirelessly powered intracranial pressure monitoring system [C]. 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE; 2017.
5. Banzhof Frank S, Demontoux Stephane D. Remote tire pressure monitoring system with plastic thin-walled valve cap and method of installing the monitoring system [J]; 2003.
6. Liuxi Tan, Sheng Liu, Honghai Zhang, et al. Numerical analysis of the reliability of tire pressure monitoring system installed on wheel hub with glue [C]; 2007.
7. Tianhu Wang, Tianyu Chen, Yue Hu. Design of intelligent LED lighting systems based on STC89C52 microcomputer [J]. Optik - International Journal for Light and Electron Optics. 2018;158.
8. Wang Hui-Feng, Liu Xiao-Hua. Design of liquid crystal digital voltmeter based on single chip microcomputer [C]. Consumer Electronics, Communications and Networks (CECNet), 3rd International Conference on. IEEE; 2013.
9. Weijie Liu, Nishuang Liu, Yang Yue. Piezoresistive pressure sensor based on synergistical innerconnect polyvinyl alcohol nanowires/wrinkled graphene film [J]. Small. 2018;14(15):1704149.
10. Kyoungwon Kim, Yong-Won Song, Seongpil Chang. Fabrication and characterization of Ga-doped ZnO nanowire gas sensor for the detection of et al. CO [J]. Micron. 2009;518(4):1190-1193.