Journal of Information Technology and Engineering
Vol. 3 No. 1 (June, 2018)

Research on Traffic Data-Collecting System Based on MC9S12D64 Microcontroller

Jian-Wei LIU, Huai-Kun XIANG, Zhi-Gang ZHU, Quan CAO & Tian YU

ITS Research Center, Harbin Institute of Technology Shenzhen Graduate School
Shenzhen 518055, China, *E-mail: jwliu@szhittech.com*

Received: 03rd March 2017 Revised: 14th April 2017 Accepted: 10th September 2017

Abstract: *Traffic data-collecting is one of the basal contents of Intelligent Transportation System (ITS). We supply in the paper a design scheme for data-collecting systems based on 9S12D64 microcontroller. Furthermore, we discuss in detail the system's operating principle, hardware structure and software design. This system has been developed, and the main function of the given scheme has been realized. The idea of the design scheme supplied in the paper, which is also applied in other CAN (Controller Area Network) distributed-control system, is to resultfully combine the high-speed of MC9S12D64 microcontroller with the high reliability and real-time of CAN field-bus. The system was certified by the national Ministry of Communications, and has been applied successfully to the actual projects in some cities in China, and its results are satisfying.*

Key words: *Intelligent Transportation System, Traffic data-collecting, System structure, CAN field-bus, Software design*

1. INTRODUCTION

In recent years, performances of many integrated on-chip MCU systems are improved largely, at the same time, the relevant software and development tools are becoming perfect, with the price of MCU dropped. With microprocessor based on control appeared [1-10], in which integrated high-powered processor kernel and plenty of microcontroller peripherals, it rapidly became an ideal substitute for the traditional multi-microprocessor unit.

A small quantity of processing operations about traffic data were dealt by several distributed processing units, whose main problems are that system's operational stability and data-transmitting reliability are so poor and the ratio of performance and price (here its abbreviation is RPP) is not good. With the RPP and integrating degree of on-chip MCU increasing, the above existing problems were solved or improved at a different degree already, its more far-ranging application in ITS is foreseeable.

We design and develop in the paper a road traffic data-collecting system centered MCU with a general CAN-Bus. In whole system's structure design, we make use of 9S12D64 microcontroller with embedded CAN module produced by Motorola Company, and TJA1050T Bus transceiver produced by PHILIPS Company as a peripherals interface. Through R&D to the system, some shortcomings existing in the traditional data-collecting management system before, for example, system's stability and data-transmitting and so on, have been made up for. In addition, these functions, road-traffic data-collecting, data-communicating and intelligent control and such, simply come true by adding some high-arithmetic to program codes. In comparison with traditional road-traffic data-collecting systems, measuring precision of some parameters in system, vehicle detecting, vehicle flow-detecting and vehicle speed-measuring etc, is heightened, especially RPP of system. The design development and realization of system's software are mainly introduced in the paper.

2. DESIGN OF SYSTEM SCHEME

2.1 Introduction of 9S12D64 Characteristic

The MC9S12D64 microcontroller unit is a 16-bit device composed of standard on-chip peripherals including a 16-bit central processing unit (HCS12 CPU), 64K bytes of Flash EEPROM, 4K bytes of RAM, 1K bytes of EEPROM, two asynchronous serial communications interfaces(SCI), one serial peripheral interface(SPI), an 8-channel IC/OC enhanced capture timer, two 8-channel, 10-bit analog-to-digital converters (ADC), an 8-channel pulse-width modulator(PWM), a digital Byte Data Link Controller (BDLC), 29 discrete digital I/O channels (Port A, Port B, Port K and Port E), 20 discrete digital I/O lines with interrupt and wakeup capability, a CAN2.0 A, B software compatible modules (MSCAN12), and an Inter-IC Bus.

2.2 System's Whole Structure Design

Traditional industry field control system communication often appears hardware structure, which is difficult to equip with a system of wire and to extend further. In the paper CAN Bus acts as communication network to link each node in the whole system, each field intelligent measure equipment links with CAN Bus through the main controller. 9S12D64 is the hard-core in the data-collecting system, in which includes some peripheral circuits, signal inductive equipment, high-speed photo electricity coupler and CAN transceiver (TJA1050T), provides CAN-Bus interface circuits, reequip with rs232 serial interface to communicate with the host. The main node in system structure is shown as Fig.1.

ILD (Inductive Loop Detector) in system includes signal optimized circuits and signal processing equipment. Under the control of 9S12D64 single-chip, traffic data from 4 forward channels of ILD in the system can be displayed then and there, and can be saved in MSF—FLASH (in addition

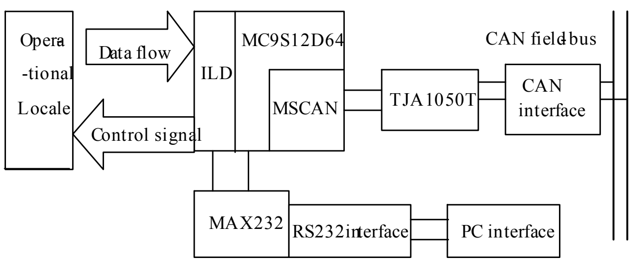

Figure 1: System Structure Master Node Sketch Map.

512K) to be propitious to use or examine timely. Moreover, we are able to control field detecting equipment through ILD control module, and are able to transmit the processed data gathered to network by CAN interface—transceiver, or through rs232 interface to the host. Since the single-chip is provided with on-line programmable characteristic, we expediently through rs232 interface realize on-line the reconfiguration or modification of system's operation parameters and sampling time and so on.

The traffic data-collecting system supplied here, which centered to 9S12D64 single-chip, is provided with high efficient data-collecting function, including signal processing, high speed real-time data-collecting, field-signal modulating and so on. Expediently, linking CAN interface with field-bus to realize the gathering and control to teledata. The single-chip has flexible on-line programmable function, through which to realize the on-line reconfiguration to data-collecting system's parameters is so easy. Furthermore, 9S12D64 provides COP watchdog and power supply monitor, so even if the system during the normal operation encounters abominable conditions, such as strong electromagnetism interference, etc, it still performs exactly as usual.

2.3 CAN Module Interface Circuit

CAN module is able to fulfill almost all Physical Layer's and Data Link Layer's functions required by CAN communication protocol. In system CAN control module is integrated in MCU 9S12D64, which can make circuit designs simple, compact, efficient, under some given conditions. CAN control module, under the control of 9S12D64, has charge of receiving and transmitting data, through internal interruption control. Each node in system makes use of Motorola single-chip with CAN module, which design simplifies the node circuits in whole hog. MCU and CAN transceiver—TJA1050T and some accessorial parts make up of CAN node together. As Fig.2 shows, a CAN transceiver, which supplies differential transmitting power to Bus and differential receiving power to CAN module, is used to act as an interface between CAN module and physical Bus.

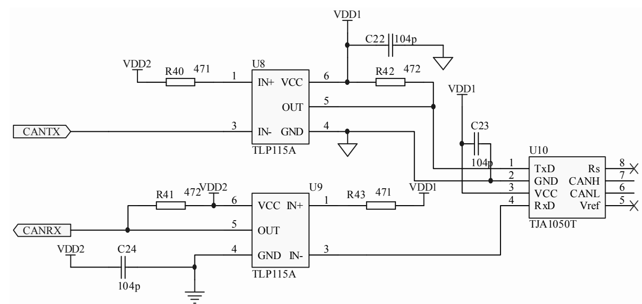

Figure 2: CAN Module Interface Circuit Sketch.

Pin CANH and Pin CANL of TJA1050T connect to Bus through a 5 ohm resistance, which restricts the electrical current to increase, and protects TJA1050T from over current impact. Between the two pins and the earth we make use of parallel connection to two 30P capacitance, which can filter high frequency interference in the Bus and shield electromagnetism radiation. The part and MCU (connecting with CANRX and CANTX) aren't displayed in Figure 2. Generally, between CAN transceiver and CAN module a photoelectricity coupler is installed to improve the ability of withstanding all kinds of transient background interference from the road environment, moreover, to ensure that traffic data would be transmitted exactly. The communication speed of CAN depends on one of photoelectricity couplers. Here to use TLP115A high speed photoelectricity coupler in CAN network, the speed of network communication, almost is equivalent with to use resistance network to drive. In addition, to match a condign terminal resistance is very important to CAN protocol Physical Layer (elided in Figure 2), otherwise, operations and network performance of CAN will be impacted directly.

3. DESIGN OF SYSTEM SOFTWARE

3.1 Frequency Detecting and Data Storage

Frequency detecting is to make use of real-time pulse capture and counting function of 9S12D64 enhanced capture timer. Frequency value is a time difference calculated from the pulse numbers and free counter. Its codes are written in compile language, and its sampled values are obtained through interruption function. Frequency storage here uses a defined buffer, that is, a two-dimension array structure, which can store 16 sampled frequency values from one of channels:

```
typedef struct{
    unsigned char bFreqWtPos;
    unsigned char bFreqRdPos;
    unsigned int wFreqDat[16][4];
}measure_freq_stru;
```

3.2 Processing Method for Traffic Data

Refer to the frequency data processing function in the interruption function, every time to calculate frequency value will lead to quote the frequency data processing function once, thereinto, still including the calculating background frequency function: void CalNocarFreq() and the calculating time difference function: void cal_timeval(), which are implemented parallelly.

Requirement: Suppose four inductive loops, which frequency figure is f_i(t) i = 1, 2, 3, 4. Sensitive degree for one of inductive loops is supposed a percentage in advance, according to frequency values some values are needed to calculate as follows:

(1) According to given sensitive degree, to distinguish whether a vehicle comes into inductive loops laid;
(2) Enforce one dimension 5 dots Median filtering waves to frequency from one of inductive loops;
(3) Carry out differential coefficient calculation to get differential time value.

Method Introduction: Dynamic storage to sequential detecting frequencies from each inductive loop, supposed current time is letter t, so frequency values stored in Buffer are f_i(t), f_i(t–1),…, f_i(t–15). As soon as a new sampled frequency value arrives in Buffer, the oldest sampled frequency should be thrown away.

(1) One dimension 5 dots Median filtering waves to sampled frequency

$$\bar{f}_i(t) = middle(f_i(t-2), f_i(t-1), f_i(t), f_i(t+1), f_i(t+2)) \quad (1)$$

For the sake of convenience to express, after that, all the frequency values used to calculate are frequency values filtered as above.

(2) Search inductive loops' no-vehicle frequency value $f_i(0)$

If the uninterrupted number of fulfilling the inequality f_i(t)-f_i(t-15)<-60 exceeds 10□Cit indicates frequency descending edge arrives. After that, if the uninterrupted number of fulfilling the inequality | f_i(t)–f_i(t–15)|<10 exceeds 375, the frequency average here is considered to the background frequency value f_i(0).

(3) Calculate differential coefficient value

$$f_i'(t) = (f_i(t) - f_i(t-15)) \quad (2)$$

(4) Calculate interval

If the uninterrupted number of fulfilling the inequality f_i(t)-f_i(0)>given threshold value reaches 3, it indicates that a vehicle goes into given inductive loops, we note the time T_i(1); when the uninterrupted number of fulfilling the inequality |f_i(t)-f_i(0)|< given threshold value reaches 3, we note the time T_i(2). In turn, we will get another two times T_{i+1}(1) and T_{i+1}(2) from the second inductive loop laid at the same roadway. We here suppose that letter L_i stands for the inductive loop's length here.

(5) Calculate the length and mean speed of vehicle

Mean speed of vehicle:

$$V_i = 540 * l_i / (T_{i+1}(1) - T_i(1)) \quad (3)$$

where l_i stands for center distance of the two inductive loops at the same roadway;

Length of vehicle = $(T_i(2)-T_i(1))*l/(T_{i+1}(1)-T_i(1))-L_i$ (4)

3.3 Data Transmission

Between the host and the nodes we make use of data transparent transmission, that is, the message block transmitted from the host to rs232 serial interface, without any changes, carries out to repackage (data block includes SOF, source address, destination address, message block type, length of message block, message block, CRC, EOF), according to its destination address, the message block is transmitted to the master node (or the master card node) or through CAN to one of the slave card nodes in network. For the bigger message blocks (that need several frames to transmit), they need to be received, one by one with their transmitted order, by the slave card node, then need to be disconnected. After the slave card node receives the complete message from the host, it will respond correctly according to command words in the message received.

If some data blocks need to be transmitted from the slave node to the host, they will be packaged in the same format. Then, the packaged data (8 bytes a frame) are transmitted to the master node one by one through CAN Bus, and stored in the given serial interface buffers, according to destination address in the message block received, to be uploaded to the host.

4. SYSTEM'S DEVELOPMENT AND REALIZATION

Modularization design ideas are used in course of system's software development. System's software development mainly includes four sections: traffic data-collecting module, data-processing module, rs232 communication module and CAN Bus communication module.

4.1 Data Transmission

Traffic data-collecting modularization is to make use of group of loop sensors distributed on roadway to sample the real-time traffic data in the field, then to use median filtering waves to filters the sampled frequency values, and stores the processed frequency values into the given buffers, to prepare for the next data-processing modularization.

……

```
void save_freq_dat(unsigned char bConst_row)
{
    sFreqDat.wFreqDat[sFreqDat.bFreqWtPos][0]
=wCh1Freq;
    sFreqDat.wFreqDat[sFreqDat.bFreqWtPos][1]
=wCh2Freq;
    sFreqDat.wFreqDat[sFreqDat.bFreqWtPos][2]
=wCh3Freq;
    sFreqDat.wFreqDat[sFreqDat.bFreqWtPos][3]
=wCh4Freq;      /*Endow internal given array*/
sFreqDat.bFreqWtPos++;
if (sFreqDat.bFreqWtPos >bConst_row){
    sFreqDat.bFreqWtPos =0;
mem_copy(&sFreqDat.wFreqDat,& testbuf ,bNum);
/*Cite internal function to store sample value
filtered into the given buffers*/
        ……
    }
}
```

4.2 Traffic Data-processing Software Design

Traffic data-processing module is mainly to make use of some control arithmetic to process the sampled frequencies stored in given buffers, here includes data filtering waves and signal compensation; users here need write some proper program codes on the basis of users' requirement and custom. After that, data will be transmitted to rs232 sending buffers or CAN controller sending buffers under the control of MCU. Here we need to pay especial attention to that in a ADC sampling period the following course must be finished: frequencies-sampling, data-processing and transmitting, then to gather the next channel's frequency signal, herein repeat the foregoing course to finish the multi-channel frequency-sampling. Therefore, this section is associated with data-colleting modularization (here write two software modularizations to make program's frame clear).

Here we give an example to calculate vehicle's speed and length, judge overspeed and overlength, as follows:

```
void dataprocess(void){
    if(sCfgFile.wParafileExist!=1) return; /*Judge whether
to download configuration parameters*/
        ……
Divdata = ((unsigned int)sCfgFile.bLoopDistance[0])
*calculate_sRoadway[0].dwFirstLoopTime;
sVehicle.Road[0].wSpeed = sCfgFile.wUnitTime
/(calculate_sRoadway[0].dwLoopDiffTime);
sVehicle.Road[0].wLength = Divdata
/(calculate_sRoadway[0].dwLoopDiffTime)
- (unsigned int)sCfgFile.bLoopLenght[0];
    /*Calculate vehicle's speed and length*/
if(sVehicle.Road[0].wSpeed > sCfgFile.bUpSpeedLimit)
/* Judge whether to overspeed*/
{ ……
    sVehicle.Roadstatus[0].mask_overspeed=1;
}
if(sVehicle.Road[1].wLength>sCfgFile.LargLengthUpLi
    mit)
    /*Judge whether to overlength*/
    { ……
        sVehicle.Roadstatus[1].mask_overlength=1;
    }
    ……
}
```

4.3 RS232 Communication Module

Through using rs232 interface and internal SCI module of 9S12D64 MCU we realize rs232 communication. Rs232 interface module is protected with ESD and discharging diode. The section includes ports' initialization, referring to the receiving and sending function. Ports' initialization is mainly to set the baud rate register, control register 1 and control register 2, which need to be carried out in whole system's initialization function. Here we need to pay especial attention to match proper communication baud rate between serial communication interface and CAN Bus. In SCI interruption function data's receiving and transmitting functions are quoted. Writing the receiving and transmitting function, which is so simple, is on the basis of current requirement, so that we leave out the two functions here. For example:

```
interrupt void sci0_isr(void){
/*Judge SCI0 status, Cite receiving/transmitting function*/
if (SCI0SR1_RDRF)
    sci0_recv(); /* Quote user's receiving function*/
if (SCI0SR1_TC ||SCI0SR1_TDRE)
    sci0_tran(); /* Quote user's sending function*/
}
```

4.4 CAN Bus Communication Module

CAN Bus communication is based on CAN2.0B protocol. CAN communication protocol, including all kinds of data frames' receiving /transmitting and checking errors function and so on, is achieved by integrated circuit embedded in CAN control module. We use 9S12D64 MCU with CAN module is convenient to design intelligent communication modules, such as module self-check, communication baud rate detection and communication module errors processing. The realization method of initializing CAN control module and receiving and sending data frames is described as follows.

(1) Initialization of CAN control module

The course is very simple, only need to set initial values to the following registers: CAN0CTR0, CAN0CTR1, CAN0BTR0, CAN0BTR1, CAN0IDAC, CAN0IDAR, CAN0IDMR and CAN0RIER, according to their datasheet. Mainly it includes some communication parameters, the baud rate, the length of bit time segment, the sample point and such.

(2) Sending and receiving data through CAN Bus

Sample values, processed by data-processing module, are stored into given buffers under the control of 9S12D64 MCU to be prepared for sending to networks or to the host. Sending data frames and receiving data frames are controlled by CAN module embedded in 9S12D64. Here there are two operational modes to choose: interruption mode and state inquiry mode. Because the second mode will take plenty of CPU's time, moreover, system here needs to process high real-time data from the given roadway, the result is that we chose interruption mode. The interruption function flows of receiving and transmitting data frames are shown as Fig.3, Fig.4.

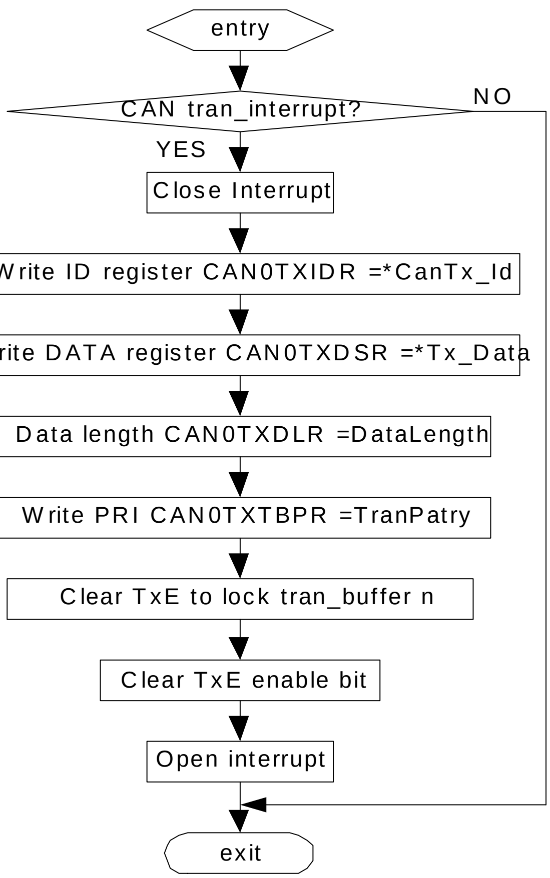

Figure 3: Flow Diagram of Transmission Interruption.

The CAN module sends data frames to Bus through the transmitting interruption function. We must write proper program to put data into transmitting buffers of CAN module, then launch the sending command of data frames. Also, the CAN module receives data frames from Bus depending on the receiving interruption function, we must write proper program to read data from receiving buffers of CAN module. We are familiar with the two functions already: the receiving function and sending function, so no examples to be shown here.

4.5 System Testing

The system introduced, so far as being, has examined by quality examination department belonging to the national Minister of communication, and its performance indexes answer the special demands completely. Its parameters examined mainly include: the vehicle's speed, the vehicle's length (they are mentioned in preceding text), the motorcade's length, the traffic flow and time occupancy ratio (the algorithm is simple, so there are no descriptions in the article) etc. The examination results of the system in some city of Shandong province, by the end of March, 2006 are as follows: the vehicle detecting error ratio is less than 1%; the speed average error is less than 3%, at low speed to 1%; the traffic flow error is less than 5%, which is greater for some large vehicles or trailers to 15%; the vehicle length error is less than 10%; the motorcade length error is less than 4 meters; the time occupancy ratio error is less than 5%. Some main parameters are showed in Table 1. (Note: the valid speed range is 10-250km/h, the usual range is 20-130km/h. The valid vehicle length scope is 3-30m.)

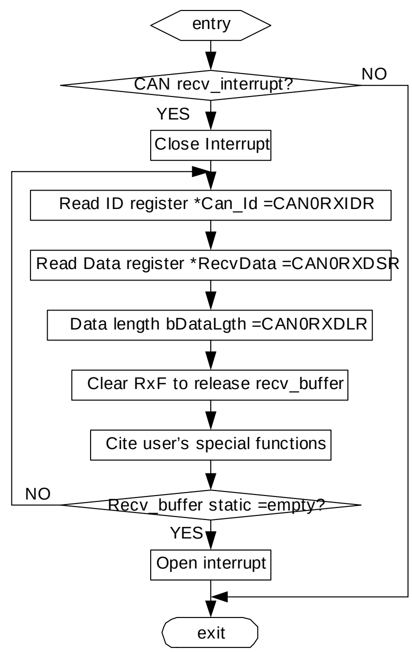

Figure 4: Flow Diagram of Receiving Interruption.

Table 1
System Testing Result

Detecting type	*Error*
Average speed	< 3%
Average speed (Low speed)	< 2%
Average length	<10%
Vehicle detecting	< 1%
Motorcade length	< 4m
Time occupancy ratio	< 5%

Note: error = | real value-detected value | or | real value-detected value | *100% / detected value.

In addition, we are also able to connect the large numbers of loop detecting cards (16 cards is testing number, that is 16 nodes) through CAN Bus to form a local area network, namely, a multi-cards traffic data-collecting system. It can meet the vehicle-detecting's demand and the other use's demand in multi-node network, in the multiple roadways road-traffic information gathering system in some large and middle scale cities. And its parameter types examined, the precision and the error of measure are equivalent with those of the single card system almost.

5. CONCLUSIONS

We supply in the paper a method to research and develop a road traffic data-collecting system centered 9S12D64 microcontroller. The system's RPP is so high, and in which is provided with CAN Bus. The system we introduce in the paper, whose design idea is that resultful combination the high-speed of MC9S12D64 microcontroller with the high reliability and real-time of CAN field-bus, can widely be used in the data-collecting projects for ITS. The system not only can realize the traffic data centralism processing and the long-distance transmission, but also can realize the on-line system's reconfiguration, which are convenient to realize the scene equipment plugging and playing and the modularization, distributed system structure transformation of data-collecting system. In the article, we lay an emphasis on the introduction of the key technique of system's software modules' development and realization. The practices demonstrate that system's operation is stable and data transmission is credible, it can meet the practical project application demands completely. The design idea about road traffic data-collecting system we supplied in the paper, also meets the other situations' demands, where the high reliability and high real-time are requested, and has a certain application value.

REFERENCES

[1] Motorola. *Motorola Semiconductor Technical Data*, 2000.

[2] TOSHIBA. *TLP115A Technical Data Sheet*, 2002,9.

[3] Z. Y. Liu. *Intelligent Transportation Control Theory and Application*. Beijing: Science Press, 2003.

[4] Computer and Communication. Editorial Department of House of Computer and Communications, 2004, (5).

[5] S. Liu, and Z. Zhu. *Design and Realization of Intelligent Measuring and Controlling Node Based on CAN Bus*. Modern Electronics Technique. 2003, (3).

[6] Y. T. Rao. *Field Bus CAN Principle and Application Technology*. Beijing: Beihang Press, 2003.

[7] B. Have, and A. Kirby. *Transparent Interconnection of Local Area Network with Bridge*. Telecommunication Network. 1984, 3(2).

[8] J. G. Shi. P. R. Zhang, and Z. Y. Chen. *CAN-bus System Design Technique*. National Defense Industry Press, 2004, 10.

[9] X. H. Yang. *Field Bus Technology and Application*. Beijing: Tsinghua University Press, 1999.

[10] J. W. Jiang, Y. Lin, and J. H. Han. *CAN Field-bus Communication Protocol's Analysis and Realization*. Computer Engineering, 2002, 28 (2).