Design of Vehicle Lamp Control System based on LIN bus

Wen Jian-yue[1, a], Luo Feng[1, b]

[1]Clean Energy Automotive Engineering Center, Tongji University,

Cao'an Road, Shanghai 201804, China

[a]13teddy@tongji.edu.cn, [b]luo_feng@tongji.edu.cn

Keywords: LIN bus, Light control

Abstract. LIN bus is well adapted in vehicle network for its easier development, and through a LIN-CAN gateway, a LIN network can be mounted to CAN network as a distributed control system. To create the vehicle lamp control system, LIN bus is selected as communication bus, and to control lamp a high side switch is introduced. With a master node capturing signals from switch board, transmitting signals to slave nodes through LIN bus, and slave nodes control high side switch to accomplish lamp control. The system works well based on LIN bus and e-switch, which shows a simple but accurate, effective and reliable lamp control system.

1. Introduction

With the development of automotive electronics and network, the way to communicate between ECUs (Electronic Control Unit) has evolved from point to point to point to points. In nowadays, CAN bus has been used in vehicle network to transfer messages between ECUs, all ECUs are connected to the two-wire bus. However, using CAN (Controller Area Network) bus is a waste in some cases, for it needs a more expensive transceiver compared to LIN (Local Interconnect Network) bus transceiver, so LIN bus is more suitable for some situations with low data transmission speed. LIN bus has become an international standard in low-speed data transmission protocols and can be used as a subnet of CAN bus. In terms of vehicle light control methods, intelligent automotive light control systems has been developed and applied to modern cars, and using LIN bus in these systems can offer a good reliability with lower cost [1].

This paper is mainly about vehicle lamp control system, utilizing a high-side e-switch (10XS3412) as a switch to control vehicle lamp. The E-Switch offers four protected 10mOhm and 12mOhm high side switches, and it works by giving it instructions through SPI. In order to control e-switch, a 32-bit MCU (Kinetis EA series of Freescale production) is used, which also provides LIN/CAN module and is mainly targeted for a wide range of applications in automotive. The whole system consists of five parts, four light control parts each consists of a KEA MCU and an e-witch, and one human interface part consists of a KEA MCU and some switches.

2. System structure

This vehicle lamp control system (Fig. 1) simulates real vehicle lights, so it consists of four light nodes with KEA MCU as LIN slave nodes, and a human interface node also as a LIN master node, all the five node communicates through LIN bus. In real application, the master node will also provide a CAN interface to communicate with high level ECUs, which makes the vehicle lamp control system an atom unit from the view of other ECUs, reducing complexity

The master node captures switch signals from switch board either through GPIO (General Purpose Input/Output) input or ADC (Analog to Digital Converter) input continuously. GPIO input is connected to on/off switch to capture the on/off signals for each lamp, while ADC input is connected to a knob to capture the luminance signal for front lamps. The captured signals are periodically sent through LIN bus to slave nodes, and slave nodes receive lamp status signals and control lamps by sending commands to E-Switch.

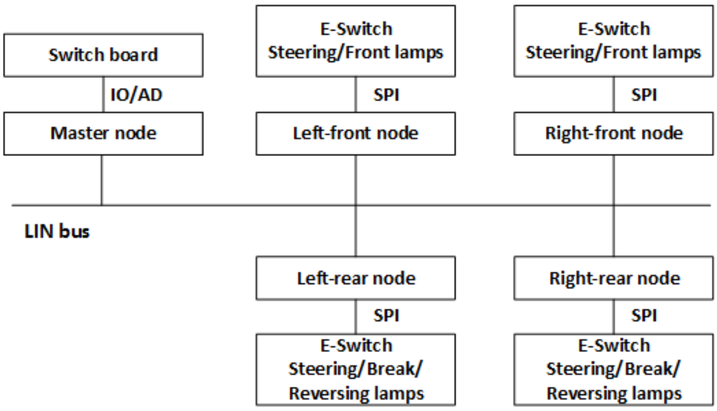

Fig. 1. System architecture

3. Hardware design

Hardware design is shown in Figure 2, mainly consists of master node and slave node, they both uses KEAZN64 as MCU, MC33662 as LIN transceiver. The master node is connected to a switch board which provides switch signals for vehicle lamps, and slave node is connected to e-switch board which is a high side switch for vehicle lamp control.

Master node and four slave nodes are using 32-bit MCU, KEAZN64. The Kinetis KEA series MCU is aimed for cost-sensitive applications in automotive markets offering very low power consumption. KEAZN64 has 64KB flash and 4KB RAM, and can work at a maximum 40MHz core clock. And it provides 57 GPIOs, one 16-channel 12-bit ADC, two SPI modules and three UART modules [2]. Based on these features, the design of master and slave nodes is shown in Fig. 2.

As is shown in Figure 2, the nodes works with 3.3V power supply, and Freescale MC33662 is used as LIN transceiver. MC33662 provides excellent EMC (Electromagnetic Compatibility) and Radiated Emission performance, ESD (Electrostatic Discharge) robustness, and safe behavior when LIN bus is shorted to ground. This transceiver is compatible with LIN protocol specification 1.3, 2.0, 2.1, and SAEJ2602-2, and is compatible with 5.0V and 3.3V digital inputs without any external components [3].

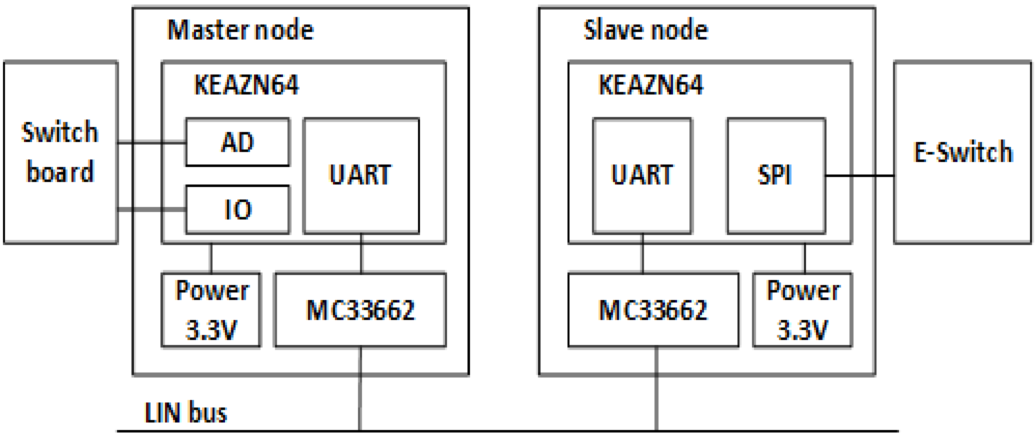

Fig 2. Hardware design

The high side switch, 10XS3412, is designed for low-voltage automotive lighting applications. It has four low $R_{DS(ON)}$ MOSFETs (dual 10mΩ/dual 12mΩ). It provides a 16-bit SPI interface for programming, control and diagnostics usage [4]. And it has a good electromagnetic compatibility (EMC) behavior with selectable slew rate output. Additionally, each output has its own parallel input, or SPI control for pulse-width modulation (PWM) control. 10XS3412 offers features like smart over-current shutdown, severe short-circuit, over-temperature protections, and fail-safe mode in case of MCU damage. It also provides output OFF or ON open-load detection, short to battery detection, analog current feedback and board temperature feedback to monitor lamp status.

4. Software design

4.1 Master and slave software. The task of master node is to capture switch signals from switch board and transmit switch signals to slave nodes through LIN bus. Besides, master node also requires lamp status from slave nodes. Two data buffers are used, one is switch buffer buffering switch states and will be used by LIN module, the other one is lamp buffer buffering all the lamp status received from slave nodes. The master node transmits LIN frames in LIN schedule table repeatedly with given timing. There are two kinds of frames in LIN schedule table, control frame and status frame. The control frame contains switch signals for all four slave nodes, and status frame contains lamp status of one specific slave node. LIN frame contains break field, sync field, protected identifier field, data field, and checksum field [5]. Once the system is started, the master node starts cycling LIN schedule table, transmitting frames to LIN bus. LIN frame starts with LIN header field, i.e. break field, sync field, and protected identifier (PID) field. And then master node checks the frame type to determine whether to continue transmitting or start receiving.

The task of slave node is to receive switch signals from master node, control e-switch according to received switch signals, and transmit lamp status to master node. Slave node receives LIN frame header and checks it PID, if PID is slave response PID, slave node will transmit response field, i.e. data field and checksum field, otherwise, slave node will receive response field. And slave node will send commands to e-switch to accomplish light control when it receives LIN frame.

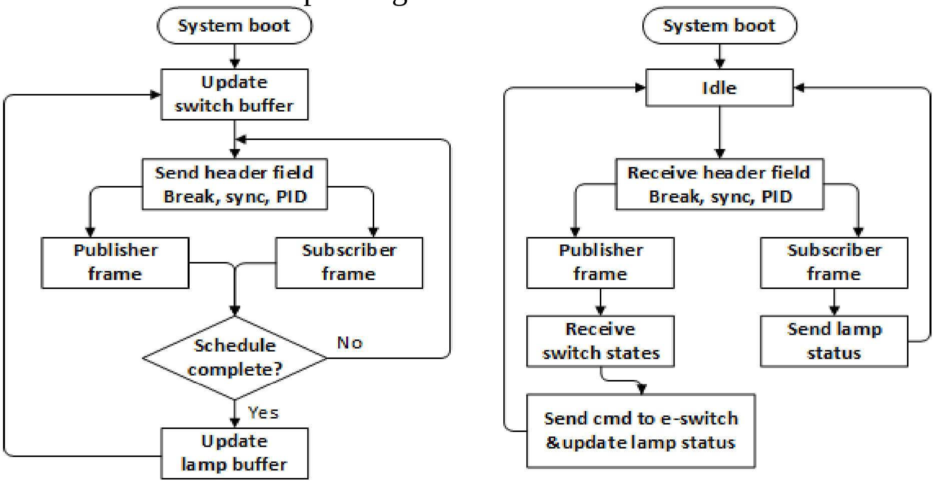

Fig. 3. Master & slave flow charts

Fig. 3 shows the flow charts of master node and slave node. Besides these functions, master node also has some LEDs, showing lamp status received from slave node, such as over current, severe short-circuit, over voltage, under voltage, etc.

4.2 LIN frame and e-switch command. One control frame is needed to transmit switch states to slave nodes, and four status frames are needed to require lamp status from slave nodes, so five frames are defined in this system. The control frame is a publisher frame, and the ID assigned to control frame is 0x10. The status frames are subscriber frames, and the IDs assigned to status frames are 0x20 (left front node), 0x21 (right front node), 0x22 (left rear node), 0x23 (right rear node). The LIN schedule table contains the five frames described above, and each frame occupies 20ms including transmission time and idle time after transmission complete. Table 1 shows the bit assignment of control frame, and Table 2 shows the bit assignment of status frame.

Table 1. Control frame bit definition

bit		7-4	3	2	1	0
Control frame ID: 0x10	Byte1	reserved	Break	Reversing	R-steering	L-steering
	Byte2	Front lamp luminance (PWM duty cycle)				Front

Control frame contains 2 bytes data field. Bit3-0 of byte1 represent on/off status of break/reversing/right-steering/left-steering lamp, and logic 0 denotes lamp off and logic 1 denotes lamp on. Bit0 of byte2 represents on/off status of front lamp, and bit7-1 represents lamp luminance value, the value is in the range of 0-127, representing 128 luminance level. If lamp luminance is not used, this value should be 127 which means full duty of PWM output.

Table 2. Status frame bit assignment

bit		7-6	5	4	3	2	1	0
Status frame	Byte1-3	reserved	UV	OV	OLON	OLOFF	SC	OC

Status frame contains 2 (front node) or 3 (rear node) bytes data field. The bit assignment of three bytes are the same except that each byte represents different lamp. Byte1 denotes status of steering lamp, byte2 denotes status of front (front node) or reversing (rear node) lamp, byte3 denotes status of break lamp. A logic 1 of the status bit means the condition occurred, and a logic 0 means the condition has not occurred. The given conditions are over current (OC), short-circuit (SC), open load in off mode (OLOFF), open load in on mode (OLON), over voltage (OV), under voltage (UV).

The e-switch (10XS3412) operates by receiving commands from slave nodes. There are mainly 2 commands used, name it control command and status command. Command definition is shown in Table 3. The control command controls the switch channels, while status command requires status of switch channels. Control command can be sent through SPI in 16-bit time, but status command should be sent twice to obtain real status. The first status command is to tell e-switch what is needed, and the second status command is to receive status data.

Table 3. E-switch commands

bit	15	14-13	12-8	7	6-5	4-3	2-0
Control command	WDIN	Switch channel	0x04	ON/OFF	PWM duty cycle value		
Status command	WDIN	Reserved				Switch channel	0x0

Bit15 of both commands feeds watch dog of e-switch, this bit should be flipped each time. In control command, bit14-13 represents which switch channel (0-3) is selected, bit7 represents the selected channel is on or off, bit6-0 represents the PWM output duty cycle value. In status command, bit4-3 represents which switch channel is selected. By sending status command twice, slave node will receive the lamp status of selected switch channel, and the received status data is the same as Table 2 shows.

5. System test

Left picture of Fig. 4 shows the system hardware controller modules, and the right picture shows the whole system components. As is described in previous parts, hardware modules consist of switch board, master node, and four slave & e-switch nodes shown in Fig. 4. Switch board provides switches for power, steering lamp, front lamp, break lamp, reversing lamp and parking lamp. By pressing these switches, the master node will capture these signals and transmit to slave nodes and then control e-switch board to turn lamp on or off. The whole system also have a 12V power source, and four vehicle lamps.

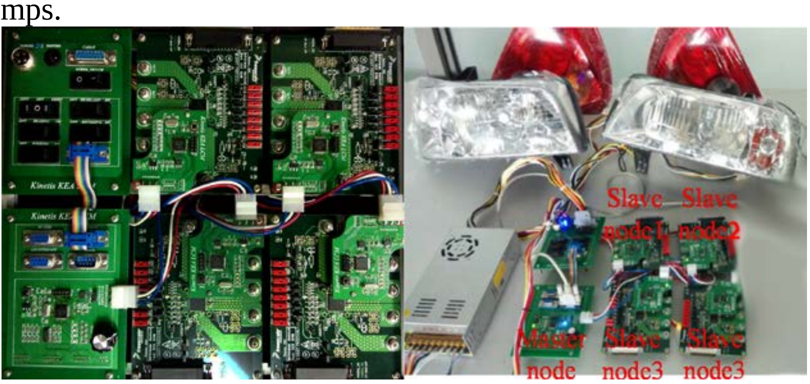

Fig. 4. System hardware components

With power on, all the nodes are ready, and by pressing switches on switch board, the corresponding lamps will light on or light off. The time delay between pressing switches and the action of lamps are ignorable from the view of human, and the exact delay is maximum at 100ms. So the LIN bus provides a very steady and reliable communication way for control systems like this one. And disconnect front lamp from the whole system intentionally, the led on the master node indicating open load is on, and reconnect front lamp to the system that led is off. So the e-switch works well and have the ability of detecting lamp faults.

6. Conclusion

This paper presents a vehicle lamp control system based on LIN bus and e-switch MCU. The system is has mainly two parts, master node and slave nodes, which uses Freescale 32-bit KEAZN64 as MCU, MC33662 as LIN transceiver, and MC10XS3412 as high side switch for lamp control. LIN bus is used to communicate between master and slave nodes, and in this vehicle lamp control system, LIN bus works well. And it shows that LIN bus is capable of cost-sensitive and low speed data transmission situations in vehicle applications. E-switch plays a very important role in this system, for the control of vehicle lamps is accomplished by it. It provides good protection when something bad happens like short circuit, and also offers detection of lamp faults which is convenient for diagnostic and monitoring. The system is an application of e-switch, and based on LIN bus the whole system is built, aiming at offering a simple architecture of vehicle lamp control system and showing the feature of automotive distributed control system.

References

[1] Liu Jingjing. The automotive Light Control System Based on LIN Bus [J]. International Electronic Elements, 2004(10): 4-10.

[2] Freescale Semiconductor. KEA64 Sub-Family Reference Manual [Z]. 2014.

[3] Freescale Semiconductor. MC33662 Datasheet [Z]. 2004.

[4] Freescale Semiconductor. 10XS3412 Datasheet [Z]. 2008.

[5] ISO 17987 Part 1-7. www.iso.org