Transmission control unit drive based on the AUTOSAR standard

Xiucai Guo, Zhen Qin
School of Electrical and Control Engineering, Xi’an University of Science and Technology, Xi’an 710054, China
qingzhen52099@163.com

Abstract. It is a trend of automotive electronics industry in the future that automotive electronics embedded system development based on the AUTOSAR standard. AUTOSAR automotive architecture standard has proposed the transmission control unit (TCU) development architecture and designed its interfaces and configurations in detail. This essay has discussed that how to drive the TCU based on AUTOSAR standard architecture. The results show that driving the TCU with the AUTOSAR system improves reliability and shortens development cycles.

1. AUTOSAR Architecture
AUTOSAR (Automotive Open System Architecture) is an open automotive systems architecture standard jointly developed by the world's major automotive manufacturers, component suppliers, automotive electronics companies, semiconductor and software companies. The goal is to establish an open automotive electronics industry standard to provides a fundamental function management solution for applications and software modules.

Basic philosophy of AUTOSAR:
1) Control the complexity of electronic equipment even functional requirements of automotive electronics is constantly growing.
2) Increase the flexibility of automotive electronics in porting, upgrading, and updating.
3) Provide electronic device scalability crossing product lines
4) Improve the quality and reliability of automotive electronics equipment.

1.1. AUTOSAR Standard Architecture

Figure 1 AUTOSAR Basic Architecture
From Figure 1 we can see, AUTOSAR basic structure is divided into four layers, namely: Application Layer, Runtime Environment (RTE), Basic Software and Microcontroller. AUTOSAR can make software architecture completely separated of layer and layer through hierarchical structure and specific software interface. The bottom is the underlying hardware—-the hardware platform MCU. The upper layer is the underlying software layer, mainly to interact directly with the MCU, provide MCU peripherals and board-level peripherals drive, and also provide software interface for upper layer. The underlying software layer consists of the operating system, communication layer, service layer, ECU abstraction layer, microcontroller abstraction layer, and complex drivers. It is not visible to the user. The upper layer of the underlying software layer is runtime environment, it exists in each ECU, and different ECU runtime environment has different ways to achieve, The purpose is to provide a bridge for communication between layers, and provide an environment to isolate the bottom for the upper layer software, making it completely independent of the underlying hardware of the system. The layer upper RTE is application layer, contains interfaces directly interact with users, at the same time, all software algorithms, strategies are packaged and implemented through application layer. In the above software architecture, layers can only interact with each other through a standard software interface to achieve relative isolation, then it can be ported on different hardware platforms.

1.2. AUTOSAR Development Process
AUTOSAR standardizes the development of automotive electronics software at the system, ECU and software component levels. System-level design includes communication matrix design, the design of the interfaces between ECUs in the system; ECU-level design is carried out in AUTOSAR standard development tools (such as Studio Tresos), including the interface of application software components, configuration of basic software; Software component-level design can build the component framework by importing software outputted by ECU-level. When the application software design is completed, Integrate and compile the .c/.h file and generate the target file, download it into the TCU.

2. Transmission Control Unit
Transmission Control Unit (TCU) is made up of 16-bit or 32-bit processor, signal processing circuits, power-driven modules and other components. Its operating temperature depends on the installation location, usually installed in the cockpit. So, the required temperature level is lower (~40 ~ 90 degrees). The terminal equipment --TCU-- can transmit the information about location, speed, vehicle or other information back to the system platform through wireless transmission. Users monitor or manage their vehicles easily by computer and mobile phone.

2.1. The Basic Composition of TCU
The components of TCU mainly are microcontroller, detection circuits, driving circuits, power circuits and communication circuits and other components, as shown in Figure 2.
The microcontroller of TCU circuit is MPC5744P produced by NXP company, it is based on Freescale's Power Architecture. This microcontroller requires highly integrity of automotive safety for chassis and safety application or other applications. In order to meet the highest safety standards for automotive and industrial safety applications, the MPC5744P’s operating frequency up to 200MHz. It owns embedded floating-point units, supports safety standards ISO 26262 / ASILD, and its device junction temperature up to 150 °C or 165 °C. This chip is suitable for automotive chassis and safety and industrial application.

Detection circuits are divided into pulse detections, switch detections and analog detections. Pulse detections are divided into pulse counting and pulse width detections. Such as engine speed, input or output shaft speed measurement uses the pulse counting. However, throttle valve opening uses the pulse width measurement.

The main measurement circuits of the analog are filter circuits and amplifier circuits. A / D conversion is built-in microcontroller with 12-bit A / D converter. Transmission ratio is driven by a DC motor. There are four H-type MOSFET circuits to achieve positive and negative PWM motor control. And, the solenoid valve driving current is also driven by the MOSFET. In addition to the mainly switching circuits and freewheeling diode in the driving circuits, there are protection circuits and current detection circuits.

The main functions of the communication interfaces are to observe the working status of the TCU, to analyse the failure of the detection sensors and to share the sensors resource.

2.2. TCU drive principle based on AUTOSAR standard

Since AUTOSAR has adopted standardized interfaces managements, users must use the appropriate softwares to set up each driving interface. Studio Tresos (EB) which is the AUTOSAR standard development tool is used in this study. EB is an Eclipse-based configuration and code generation tool for vehicle software module. It enables users to configure software modules, and users can validate configuration conformance. It can also generate codes for standard software modules such as the AUTOSAR standard software kernel. Due to extensibility and openness of the structure, it can integrate user-specific software modules or inherit software components in addition to the AUTOSAR basic software modules.

The TCU driving based on the AUTOSAR standard mainly develops components of the application layer software of the AUTOSAR architecture, that is definitions of ports, interfaces, running entities, and RTE events corresponding to triggering running entities.

2.2.1. Ports. Port defines the directions in which Software Components (SWC) communicating with the outside world. The definitions of the ports, that are, in the EB, the definitions of MPC5744P chip pins. The main definitions of the pins’ types and directions. The types of ports are GPIO, SPI, ADC, CAN, Etimer, etc.. The multiplexing functions of the chip pins have to be correctly chose in order to achieve the desired functions. Figure 3 shows the configurations of ports in the EB. This port is an output port of CAN communication and the internal pin ID is 11.
2.2.2. Software interface. After the definitions of the ports, users must define the software interfaces, because the first step only defines the input or output directions and types, the contents of these ports also need to be defined by the software interfaces. As shown in Figure 4, the software configured ADC0 sampling type that is interruption sampling, sampling prescaler is 2, the ground reference voltage is 3V and so on.

The functions to drive TCU are completely developed in advance based on Simulink, as long as the relevant parameters of the configuration are completed in the EB, the tool can automatically generate the corresponding C language functions. Therefore, developers only need to provide parameters. When access to external calibration parameters, these interfaces play key roles.

The above configurations completed single software component. If developers want to achieve the corresponding function, a combination of these software components is needed through the associated instance ports to create links. The links include the internal links and external links., External links use CAN bus communication.

2.2.3. CAN communication. Control Area Nerwork (CAN) was first introduced by the German company BOSCH for automotive internal measurement and implementation of data communication between components.
TCU own CAN communication serial ports, the external data link is also created mainly using CAN network communication. For CAN communication configurations, the most important things are the calculation of the baud rate and the calculation of the nominal bit time. According to the crystal clock and external serial clock to select the correct baud rate is the most critical step. The baud rate and the nominal bit time was the inverse relationship, that is,

\[ \text{baudrate} = \frac{1}{t_{NBT}} \] (1)

\[ t_{NBT} \] is the period of the nominal bit time, which consists of four sections, SYNC_SEG, PROP_SEG, PHASE_SEG1, and PHASE_SEG2, where the value of SYNC_SEG is a quantum of time. Therefore, the relationship between these four parts and time quantum can also be shown as follows,

\[ t_{NBT} = (1 + \text{PROP}_\_\text{SEG} + \text{PHASE}_\_\text{SEG1} + \text{PHASE}_\_\text{SEG2}) \times t_Q = \text{NBT} \times t_Q \] (2)

The \( t_Q \) is time quantum. The ranges of SYNC_SEG, PROP_SEG, PHASE_SEG1 and PHASE_SEG2 are shown in Table 1.

| Segment       | Duration                    |
|---------------|-----------------------------|
| SYNC_SEG      | \( t_{\text{SYNC\_SEG}} = 1 \times t_Q \) |
| PROP_SEG      | \( t_{\text{PROP\_SEG}} = (1, 2, ..., 8) \times t_Q \) |
| PHASE_SEG1    | \( t_{\text{PHASE\_SEG}} = (1, 2, ..., 8) \times t_Q \) |
| PHASE_SEG2    | \( t_{\text{PHASE\_SEG2}} = \max(IPT, t_{\text{PHASE\_SEG2}}) \) |
| NBT           | \( 8, 9, 10, ..., 25 \) |

In general, peripheral clock and crystal clock can be obtained from the datasheet, so the required values can be calculated and configured according to the formula. After the configuration is completed, CANalyzer is used to debug. The debugging result is shown in Figure 5.

\[ \text{Figure 5 CAN Communication} \]

3. Test verification
Convert the above configurations to real C codes, embed the function codes of each component and the target codes file of BSW, then download it into the target TCU. In the end, connect the TCU with solenoid valves, sensors and other real loads.
In Figure 7, the red curve is the input shaft speed, the blue curve is the engine speed. When the vehicle speed accelerates to the experimental value, the input shaft speed rises to a certain value and remains unchanged, and the engine speed also rises to a certain value to keep gentle. When the vehicle speed declines, the two also will decline.

In fact, in theory, input shaft speed and engine speed will be a proportional relationship at a certain moment, this relationship is not only related to the speed, but solenoid valves, throttle opening, etc.

The test verifies that the TCU driving development method based on the AUTOSAR is correctly applied.

4. Summary
Automotive electronic embedded system development based on AUTOSAR standard is a trend in the future development of the automotive electronics industry. Based on the AUTOSAR standard, the modular design and real-time environment interface configuration of the TCU driving system can separate the system application layer software from the underlying basic software to facilitate the expansion, migration and reuse of the software. Compared with the traditional TCU system development model, this way will increase the reliability of TCU, and shorten the development cycle, improve the maintenance efficiency.

References
[1] Sun Shen, Song Ke, Zhang Tong, AUTOSAR Standard Development and Application Status, Mechatronics, 2014.12.007: 34-35
[2] Guo Xiwen, Automotive Electronic Communications and Applications Reference to AUTOSAR Standards, Zhejiang University(2004)
[3] VEVTOR, Control System Development Process Based on AUTOSAR Architecture
[4] Feng Chuan, Hu Jie, Yan Fuwu, Liu Chuanbao, CAN Underlying Communication Research Complies with AUTOSAR Standard, Journal of Wuhan University of Technology (Information and Management Engineering), (2013)35(6): 842-845
[5] Yu Jiangwei, Research on Control Method of Automobile Automatic Transmission and Development of Simulation System, Harbin Institute of Technology(2006)
[6] Huang Zongyi, Automatic Transmission Principle and Design of Modern Car, Shanghai: Tongji University Press(2006)
[7] Wu Guangqiang, He Lin, Fan Dapeng, Research on the Performance of Continuously Variable Transmission for Vehicle, Transmission technology(2007)(6):16-19
[8] Feng Jinxiang, Wu Jizhang, Automatic Transmission Structure Principle Atlas, Beijing: Mechanical Industry Press(2002)
[9] Zhang Liping, Research on Model Driven Design and Information Generation Based on AUTOSAR Standard, Zhejiang University, 2-3
[10] Liu Jizhou, Research and Application of Driverless Intelligent Vehicle System Platform, Zhejiang University(2017)