

## Function Test Method of Anti-pinch Window Based on LIN Bus

Anyu Cheng, Yong Deng<sup>a</sup> and Shuang Ji

School of Automation, Chongqing University of Posts and Telecommunications, Chongqing  
400065, China

<sup>a</sup>dengyong@sumarte.com

### Abstract

**Due to the increasingly complicated functions of automobile windows, the human cost of functional testing of the ECU (Electronic Control Unit) of a window increases and the test period increases. This text has studied the automation function test method of the anti-pinch window based on LIN (Local Interconnect Network) bus. To use SMS (Simple Mechanics Simulator) as a hardware platform to replace the anti-pinch motor to achieve the complete motor mechanical movement process. Form a closed loop test of the test system and the ECU under test, The bottom driver module is written in Python language, It can realize batch test of complicated functions of ECU, greatly reduce manpower cost and shorten testing cycle.**

### Keywords

**Windows, control unit, anti-clip, functional test.**

### 1. INTRODUCTION

The window system is an important part of automobile body. However, the increase of a series of functions such as the anti-pinch of the window leads to the more complicated function test of ECU. References [1][2] are still based on the traditional motor Mechanical structure of the functional test, ECU itself a single function, the test for manual testing. References [3][4] in the traditional ECU function test, using the test panel to generate various signals required by the ECU to trigger its driving module to control the work, the whole process is also one by one manually testing, manual control of the results of the direct Observe and record. References [5][6] are based on VT System functional tests that partially test the functionality associated with I/O hard-wire connections, but it can not do testing against pinch-related functions. These kinds of test methods all increase the testing labor cost, complexity and testing time of the testing process, and it can not automatically test and automatically generate results, and it also can not test multiple ECUs at the same time, it greatly increased the ECU development process cycle. Therefore, a new functional test method is proposed. Based on the LIN bus ECU functional test method, the LIN bus transmission is taken as the key technology. The LIN master-slave scheduling mechanism is simulated by Canoe, and the ECU functional test cases are written in Python. And test control logic functions. Using Aptana Studio as a host computer software, it controls the execution of the device under test and the status feedback of the executing device. The closed-loop test mode for automating the functional testing of the ECU is implemented.

### 2. Control Unit Functional Test Introduction

Vehicle's ECU main task is to solve the control of body electrical equipment, ECU shared network information on the bus. Therefore, ECU testing should first focus on functional control, that is, according to the control protocol, functional requirements for the development of test programs to determine whether the ECU can meet the requirements. Different ECUs have different control functions, and the overall implementation of the vehicle-mounted ECU realizes overall control over the functions of the vehicle. Therefore, ECU for different functions, It is necessary to develop a personalized, automated and informational test scheme based on specific requirements. However, no matter how the test plan is formulated, the controller test process can not interfere with the normal bus communication [7]. In this paper, LIN bus anti-pinch window control unit function test research

and design, the test system should include: ECU function test and fault diagnosis test. Among them, ECU function test is mainly a LIN message, IO function, anti-pinch function and voltage level test. It is an important part of the ECU research and development test. Therefore, Design a customized automatic test platform based on the functional requirements of ECU functional testing and provide the most accurate feedback information for development. The whole system is divided into three parts: the upper computer software, the execution equipment and the measured equipment, shown in Figure 1.

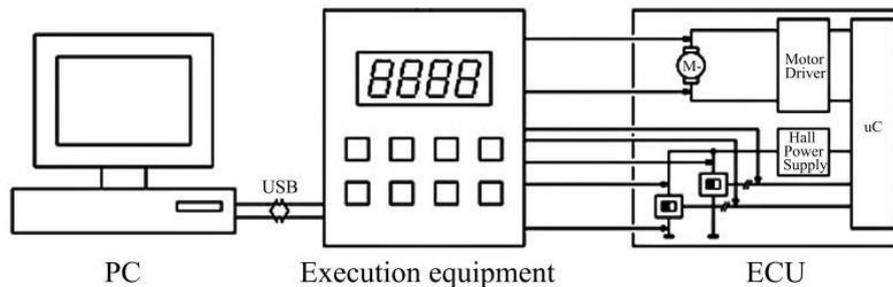


Figure 1 Test plane structure

The main work of the host computer is to generate simulation signals, test control strategies and the results of the judgment; The middle implementation of the device according to the control command, the signal output of the implementation and comparison, and feedback to the ECU and PC software. The device under test (ECU) is based on LIN and I/O trigger signal to control the movement and state changes, and then through the implementation of equipment status feedback to build the entire test system into a closed-loop system. At the same time, the function test of anti-pinch windows based on LIN bus presented in this paper can be based on different functional specifications of the implementation of different configurations of the device, and the host computer software to make the appropriate control strategy adjustment to meet the different ECU functional requirements of the test.

### 3. Anti-pinch windows functional test program design

In this paper, Vector CANoe, Continental SMS and T\_BOX are connected with the tested ECU to build an automated test platform. The entire test is built into a closed-loop structure, which greatly reduces the labor cost and shortens the product development cycle.

#### 3.1 The overall framework of the program

Aptana Studio is the software implementation platform for ECU functional testing. The overall framework of the test system consists of three parts: PC software (LIN network and Python script driver library), the implementation of equipment (SMS) and the device under test (ECU). as shown in Figure 2.

PC software simulation and control of various LIN message output to the ECU to trigger the ECU to control the output. Use Aptana Studio to write Python scripts to implement control strategies for test cases and to collaborate on individual peripherals. Through the collection of the motor state information to compare and analyze to determine whether the software meets the functional requirements of the definition, and the final output test report.

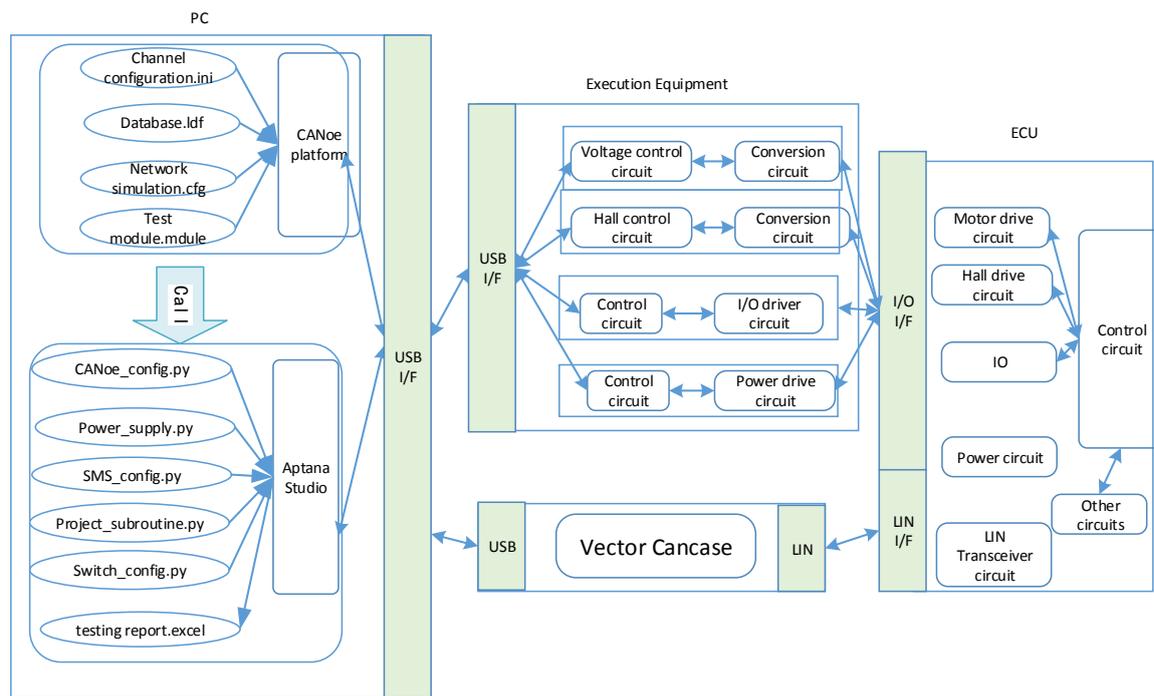


Figure 2 Anti-pinch window function test overall framework

**3.2 hardware configuration**

Customize the T\_BOX device and configure the driver files to load the configuration files into the Aptana Studio platform according to your requirements. During the test, Only need to test the relay operation to replace the window up and down switch.

Table 1 window ECU part of the signal table

Control signal	Explanation	Types	category	Remark
Authorization	Authority	IO	input	Highly
Manual_Up	up	IO	input	Highly
Manual_Down	down	IO	input	Highly
AUTO	auto	IO	input	Highly
VBATT-P	Voltage	IO	input	6V-18V
ECU Initialization	Initialization	LIN	1 bit	0-1
Hall Sensor Error	Hall error	LIN	1 bit	0-1
Motor relay Error	Motor error	LIN	1 bit	0-1
Window under voltage	Undervoltage	LIN	1 bit	0-1
Window over voltage	Overvoltage	LIN	1 bit	0-1
Window switch stuck	Blocked	LIN	1 bit	0-1

According to Table 1 window ECU control signal table, set up automated test system. Executive equipment includes IO, LIN, Power and other equipment. Hardware configuration steps are as follows

- 1 Aptana Studio configuration IO signal, SMS driver, Power driver.
- 2 CANoe build simulation network. Cfg engineering, simulation LIN network scheduling and analysis.
- 3 Aptana Studio add a test module to write test case logic and judgment rules.
- 4 Aptana Studio executes the test module, executes the test strategy, and generates the test report.

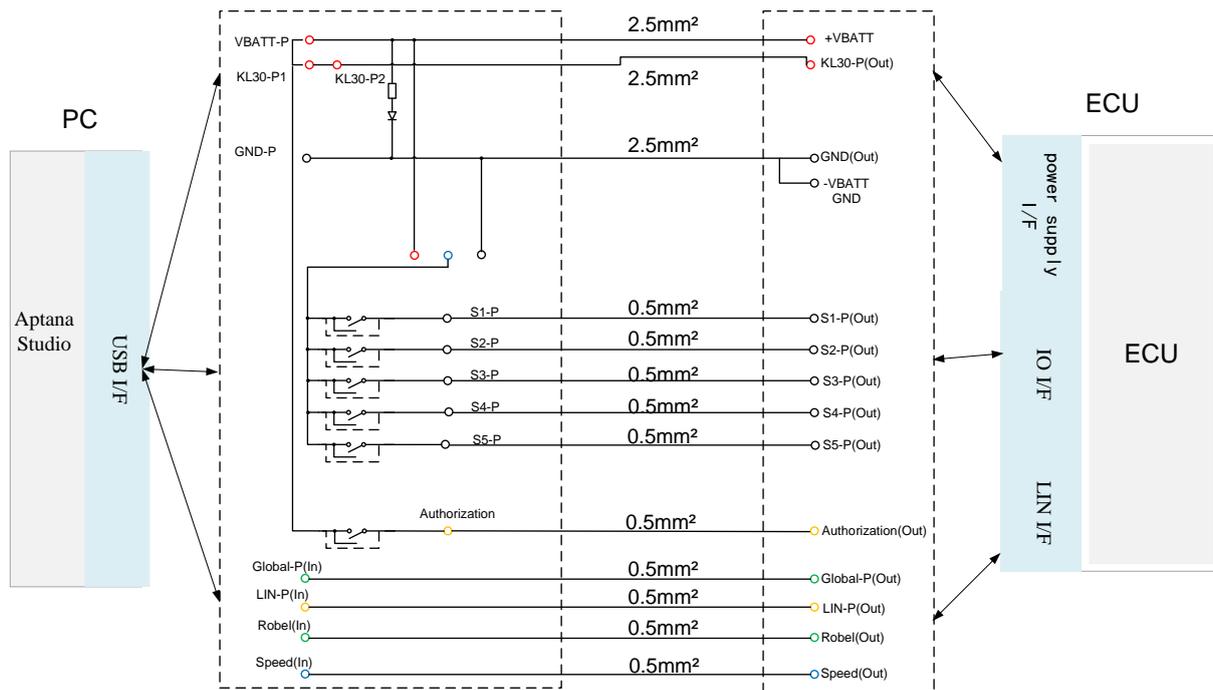


Figure 3 Tbox signal wiring diagram

As shown in Figure 3, according to the working principle of the window ECU, at the same time in order to ensure that the system test function test system to achieve automated testing, The controllable Tbox of the host computer software is used to test the ECU. The external switch signal, power signal, and LIN signal can be configured by the PC software to trigger the internal logic of the ECU. PC software at the same time the ECU output feedback signal comparison analysis to trigger and detect the ECU, the formation of a closed-loop purpose, to achieve the system fully automated test analysis process, and generate test reports.

### 3.3 Anti-pinch window test software framework

The design of software test cases is divided into pre-conditions, testing process and test results in three parts. As shown in Figure 4

- 1) Preconditions. Initialization of peripheral control equipment, and initial configuration of operation signals.
- 2) testing process. According to the functional requirements of the test case corresponding test points, the operation of peripherals and output simulation control signal.
- 3) test results. According to the test step corresponding to the results of comparison, the data to determine whether the definition of the value.

Build the Python test software framework, the relationship between the modules nested between each other as follows, we can easily program flow clean test framework to facilitate the tracking of issues and configuration parameters, but also facilitate the transplantation of similar products.

1. The window ECU function test configuration contains all the variable usage information, type and bus type, entered in the devices\_config module.
2. Variable input to the abstraction layer module and constant and switch the instance mapping for all classes.
3. The variables used in the common routine are entered along with the address constant that contains the project parameters.
4. According to the common routine is in the subroutine PWL (Power Window Lifter) input configuration, the specific function of the window ECU increased.
5. The project subprogram module adds specific functions to the project (eg signal functions for specific projects, diagnostic commands, hardware inputs).

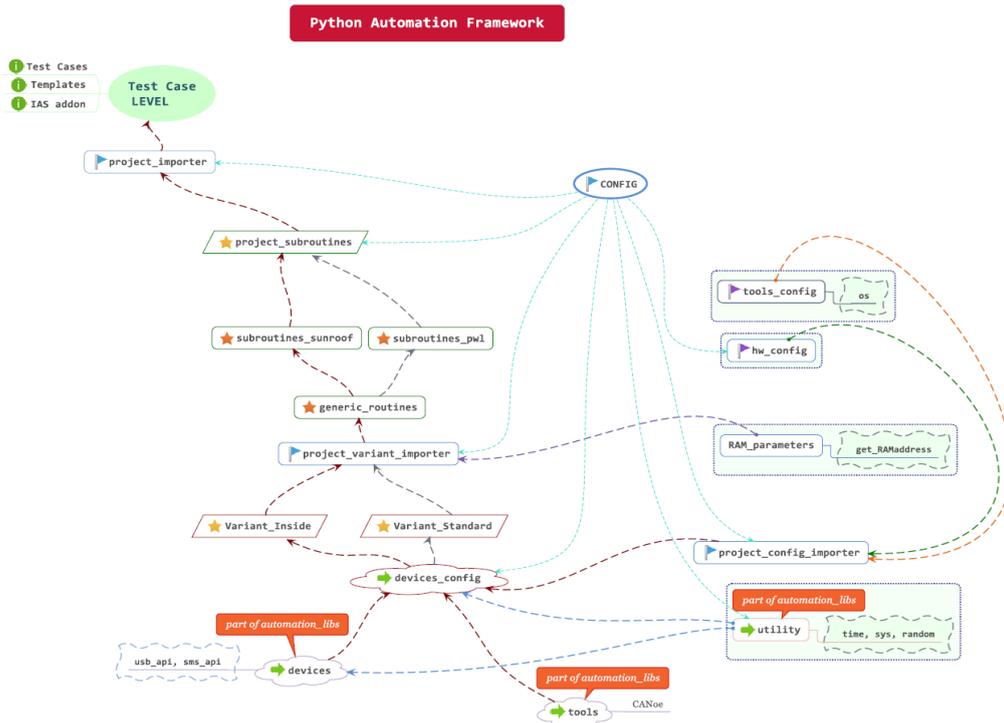


Figure 4 Python test software framework

### 4. Test implementation

The system mainly implements the automated testing of PWL (Power Window Lifter) to determine whether the given protocol, requirement and specification are met. At the same time many times after the cycle test to determine whether the control function has good stability. Test system interface shown in Figure 8.

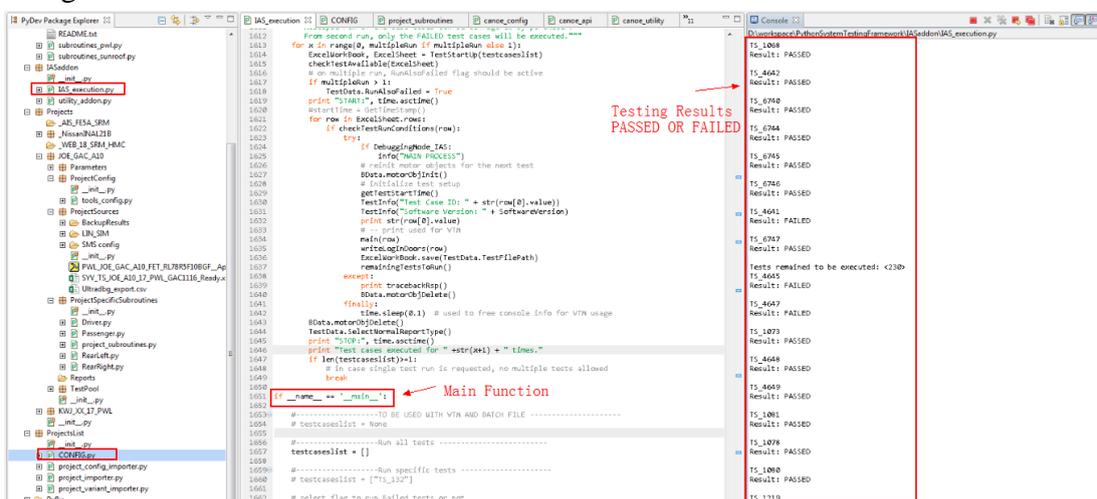


Figure 5 software test system interface

IAS\_execution in the figure is the main function module of the whole project, which is equivalent to the function of "main" function. It calls the external device to execute the test process collaboratively and calls the input excel test case as the test case list. IAS\_execution defines the implementation of the calling relationship excel, and the test results are output to the Console and print out to the test excel report, it is convenient for testers to analyze the testing process. The whole process completed the test and generated the report automatically.

In the generated test report, the details of the test are pointed out, such as the precondition procedure, the test procedure and the execution details of the test result. Combining with the test case, CANoe

Trace data and corresponding Time stamp, and the test report to analyze and compare, so as to verify the feasibility and correctness of this paper. Figure 7 below is a test record of one of the test cases.

```

0.000: -STEP-----PRE-SETTINGS-----
0.000: -STEP- LocalSwitch Release all is set to: [0]
0.209: -STEP- DriverSwitch Released - PWSM_1:PWSM_FRSwitchSt is set to: 0.0
0.270: -STEP- Set Authorisation signal to ON
0.270: -STEP- BCM_LIN2_1:BCM_APWLEnable = 1
0.272: -STEP- Set Ignition signal to ON
0.272: -STEP- BCM_LIN2_1:BCM_KeySt = 2
0.276: -STEP- Send Remote Control Close to Release
0.277: -STEP- BCM_LIN2_1:BCM_MasterWindsCloseCmd = 3 for 3 frames
0.278: -STEP- Send Remote Control Open to Release
0.279: -STEP- BCM_LIN2_1:BCM_MasterWindsOpenCmd = 3 for 3 frames
0.282: -STEP- Send the Comfort Enable to ON for 3 frames
0.282: -STEP- Release the Comfort Command for 3 frames
0.283: -STEP- Set Child Protection signal to OFF
0.283: -STEP- PWSM_1:PWSM_WindowsLockButtonS = 0
0.285: -STEP- Set Child Switch Stuck signal to Inactive
0.285: -STEP- PWSM_1:PWSM_WindowsLockButtonStuckS = 0
0.286: -STEP- Set Driver Switch Stuck signal to inactive
0.286: -STEP- PWSM_1:PWSM_FRSwitchStuckS = 0
0.960: -INFO- Power supply is set to: 13.790[V] in 0.351[s].
0.965: -STEP- Block/Pinch removed on motor WL
1.184: -INFO- Current motor position: 1267
1.184: -STEP- LocalSwitch Manual Close Pressed is set to: [1]
1.372: -PASSED- Motor active within activation time (0.006s < 0.120s). <WL> position: 1258.
7.267: -PASSED- Motor stopped from moving towards target. Motor monitored for: 5.893s. <WL> position: 32.
7.267: -STEP- LocalSwitch Manual Close Released is set to: [2]
7.738: -PASSED- Motor sent to position: 33. Actual <WL> position: 32.
8.429: -INFO- Power supply is set to: 9.310[V] in 0.352[s].
8.429: -STEP- Set Ignition signal to OFF
8.429: -STEP- BCM_LIN2_1:BCM_KeySt = 0
8.440: -
8.441: -INFO-----PROCEDURE-----
8.441: -INFO- LOG: motor: action, [positions], delta [M1-M2], position based, time based, [states]
9.890: -LOG- WL, inactive, [32], 33, 0, 0.0, [WL inactive].
9.896: -STEP- Send Remote Control Open to Global Open
9.899: -STEP- BCM_LIN2_1:BCM_MasterWindsOpenCmd = 2 for 20 frames
10.236: -LOG- WL, active_open, [33], 34, 1, 0.346, [WL active_open].
10.240: -INFO- Wait for 300 ms
11.213: -INFO- Power supply is set to: 8.200[V] in 0.352[s].
12.416: -LOG- WL, inactive, [178], 179, 145, 2.18, [WL inactive].
13.117: -INFO- Power supply is set to: 9.310[V] in 0.352[s].
13.126: -STEP- Send Remote Control Open to Release
13.131: -STEP- BCM_LIN2_1:BCM_MasterWindsOpenCmd = 3 for 10 frames
13.136: -STEP- Send Remote Control Open to Global Open
13.141: -STEP- BCM_LIN2_1:BCM_MasterWindsOpenCmd = 2 for 100 frames
13.491: -LOG- WL, active_open, [179], 180, 1, 1.075, [WL active_open].
25.588: -LOG- WL, inactive, [1269], 1270, 1090, 12.097, [WL inactive].
28.672: -INFO-----POST PROCESSING RESULTS-----
28.677: -PASSED- 2. WL active_open, within remoteActiveTime
28.679: -PASSED- 4. WL inactive, within UnderVolStoptime
28.681: -PASSED- 8. WL active_open, within remoteActiveTime
28.683: -PASSED- 9. WL inactive, when WLWLOpen
    
```

Figure 6 test process records



Figure 7 test data record

In the test report to do the stability and accuracy of the analysis, the experiment is repeated. The same test case, hundreds of experiments, and compare the data records of the original data and test reports are the same. After repeated tests the same timestamp LIN message semaphore test report output and the actual data record file. Shown in Figure 7 below, intercepted one of the test case data recording screenshot. Therefore, the method adopted in this paper can guarantee the correctness and stability of the automated testing process.

## 5. Conclusion

This paper introduces the function test method of automobile anti-pinch window control unit. The function test method that is aimed at the traditional car window can not be qualified based on the test of the complicated anti-pinch car window. In this paper, LIN bus transmission is taken as the key technology, and LIN master-slave scheduling mechanism is simulated by Canoe. Aptana Studio is used as a script to write functional test cases. To control the implementation of the test ECU to the implementation of the process equipment and the status of the implementation of the feedback device analysis to achieve window ECU automated functional testing of closed-loop test methods. This paper's functional testing process is more simple and intelligent, reducing labor costs and shortening the development cycle for project development. In improving the simulation signal and the underlying test driver library, it can be extended to the same type of different functional requirements ECU functional test, it has better portability and stability.

## References

- [1] Su Hong, window bus anti-pinch control system based on LIN [D]. Harbin Institute of Technology, 2015
- [2] Du Yanhai, Double Hall sensor with anti-clip electric window design [D]. Donghua University, 2014
- [3] Wang Liping, Automotive CAN Network Control System Design and Research [J]. Industrial Instrumentation and Automation, 2009 (5): 77-79
- [4] WU WEI-BIN, HONG TS, LUO CAI RU, et. Hardware-in-loop of alternative fuel engine ECU [C] // Proceedings of the Second International Conference on Computer Modeling and Simulation. Washington, DC: IEEE Computer Society, 2010: 291-294.
- [5] Mao Honglin, Xu Yong, Han Guangguo. Validate System Implementation Based on VT System [J]. Automotive, 2016 (12): 48-52 (in Chinese).
- [6] Gao Shasha, Fu Dongxiang, Wang Yagang. Automatic Testing System of Automotive ECU Based on VTSystem [J]. Information Technology, 2015 (1): 129-132.
- [7] Zhang Yonggang, CAN bus-based automotive network and function automation test system development [J]. 2017.
- [8] Cao Zhong, Kalman filter-based anti-pinch control system for electric windows [D]. East China University of Science and Technology, 2015
- [9] Wang Feng, Fan Xiaobin. Overview of automotive power window anti-pinch technology programs [J]. Modern Manufacturing Engineering, 2015 (8): 150-156.
- [10] Fu Zhaohui. Discussion on electric anti-pinch window technology [J]. Automotive, 2012 (7): 15-18.