A way to synchronize clocks with the FlexRay bus

Fengying Pang\textsuperscript{1a*}, Min Huang\textsuperscript{1b*}, Zhaoyang Mi\textsuperscript{1c}, Huimin Zhang\textsuperscript{1d}

\textsuperscript{1}North China Vehicle Research Institute, BeiJing 100072, China

\textsuperscript{a*}pangfengying1983@163.com, \textsuperscript{b*}keenh@126.com, \textsuperscript{c}miziaoyang1989@126.com, \textsuperscript{d}zhanghuimin1989@126.com

Abstract—In order to ensure that different systems work together under the same clock, it is necessary to propose a unified time management system and a unified time benchmark, this paper studies the method of clock synchronization based on FlexRay bus, which has been verified by the laboratory to achieve clock synchronization by FlexRay bus, with synchronization accuracy of less than 10us.

1. Introduction
With the development of information and intelligence, information systems need to complete many different functions at the same time, many different functions are usually performed by different nodes, and some backup nodes also need to complete the active fault tolerance function. In order to ensure the consistency of all the behavior of the whole information system, it is necessary to ensure that the information system is a real-time system and that all nodes work under the unified clock. Information system belongs to the typical distributed network system, each node's clock has drift, with the growth of time, the clock gap between the nodes is getting bigger and bigger, will eventually lose reference significance, and between nodes in the data transmission process, regardless of the use of star or bus topology, network data transmission has a delay, therefore, it is necessary to establish a unified time benchmark within the entire information system, each node and the time base synchronization, and verify synchronization accuracy.

FlexRay bus is a real-time bus based on time trigger mechanism, high speed, fault tolerance. Communication on the FlexRay bus takes place in a periodic loop that always includes static segments (ST) and network idle time (NIT), as well as dynamic segments (DYNs), symbolic windows (SWs). Static and dynamic segments consist of time slot slots, which transmit frame information through time slots, which are repeated over fixed periods. In static segments, time-triggered TDMA counting is implemented with time-sharing, and multiple time slots are fixed to each node, within which only that node is allowed to transmit data. This access method can be used to ensure certainty when a static segment accesses a bus. In order to reduce network transmission delay, the static segment of the FlexRay bus is used to synchronize the clock\cite{1}.

2. The Principle by Which The FlexRay Bus Implements Clock Synchronization
The FlexRay bus is a real-time bus based on time certainty, which is the basis for clock synchronization\cite{2}\cite{3}. The FlexRay bus node is hardware-based on Freescale microcontroller implementations and maintains local clock operation through the timers.

The FlexRay bus transmits a fixed time per data frame in a static segment, and the FlexRay node master clock and slave clock read the local clock at the beginning of the fixed time T0 (cycle0),
respectively, as Tm and Ts, and at the next cycle, the FlexRay node master clock sends the local clock time stamp to the FlexRay bus, after two cycle intervals. The FlexRay node receives the timestamp information sent by the FlexRay node master clock from the clock and compares Ts and Tm to make time-to-time judgments based on the time difference between Ts and Tm, and then to the time according to actual needs.

The FlexRay bus cycle length is configured at 5ms, with a large period of 320ms, and every 320ms the FlexRay node master clock and slave clock are compared and timed.

3. The Local Clock of FlexRay Bus Node

3.1. The data structure of local clock
The local clock consists of seconds, 50ms, us, three-stage time units, and the data structure is as follows:

```c
struct LocalClock {
    uint16 microsecondsField;
    uint8 fiftyMillisecondsField;
    uint32 secondsField[2];
}
```

3.2 The Timer of local clock
The EPIT timer consists of two levels of decreasing timers\(^4\). The first stage is a micro timer, with a timeout time of 1us, and the second stage timer sets a timeout of 50ms. After the first-stage timer times out, the value in the second-stage timer decreases by 1, and the second-stage timer enters the timer interrupt handler when the count is 0, and the interrupt handler flowchart is shown in Fig.1.

![Fig.1 Interrupt handler flowchart](image)
3.3 **The record of local clock**

The local clock's secondsField[2], fiftyMillisecondsField two-stage are saved in local memory for two levels of time, read directly, and the microsecondsField time is stored in the COUNTER register PITCNT of the EPIC timer, which is a decreasing counter, which is converted to microsecondsField after reading, and converted to a microsecondsField 49999-PITCNT.

3.4 **The record of local clock**

The local clock's secondsField[2], the fifty MillisecondsField two-stage time is saved in local memory, write directly, and the microsecondsField time is stored in the EPIC timer's counter register PITCNT, which is read-only register, cannot be rewritten, and needs to be temporarily adjusted by modifying the initial EPIT value.

4. **The Synchronizes Process between Master Node and Slave Node of FlexRay Bus**

The FlexRay node and the standard clock source synchronization process consist of three states: out-of-sync, synchronous adjustment, and synchronization.

In the initial state, the slave node is in a non-synchronous state, in the non-synchronous state, the local clock state is set to invalid, the local clock output is turned off, 10 consecutive times to receive the master clock node adjacent to the two clock timescales difference of 320ms, think the master clock node is stable, into the synchronous adjustment state.

In the synchronous adjustment state, the local clock state output is set to invalid, the local clock output is turned off, the local clock and master clock time-to-time program is executed, the time difference between the local clock and the master clock is determined, if the local clock and the master clock time difference is less than or equal to 100us, into the synchronization hold state, and if the clock information sent by the primary node cannot be received three times in a row, into a non-synchronous state.

In the synchronous hold state, the local clock state is set to be valid, the local clock output is turned on, the local clock-to-master time program is executed, the local clock and the master clock time difference is determined, if the local clock and the master clock time difference is more than 200us or three consecutive times can not receive the clock information sent by the master clock, into a non-synchronous state.

The three state migration diagrams are shown in Fig.2.

![Fig.2 Clock synchronization status migration diagram](image-url)
4.1 The FlexRay bus node sends the master clock time information
The FlexRay bus node master clock records the local clock timestamp value (Tm) at each Cycle0 starting moment of FlexRay and sends T0 to the FlexRay bus on Cycle1[5][6].

4.2 The FlexRay bus node synchronizes the process from the clock to the master clock
The FlexRay bus node records the local clock timestamp value (Ts) from the clock at each Cycle0 starting moment in FlexRay, receives the timestamp Tm sent by the FlexRay bus node master clock at Cycle3, calculates the absolute value of the difference between Ts and Tm Tdiff, and then determines the state of the FlexRay bus node from the clock, and if it is in a synchronous adjustment state or synchronous hold state, performs the timing process.

The FlexRay bus node's slave clock vs. master clock timing process is shown in Fig.3.

![Fig.3 Process for synchronizing the master clock with the slave clock on the FlexRay bus node](image)

4.3 The timing process
Depending on the size of Tdiff, adjust the local clock as follows:
(1) If Tdiff is greater than or equal to 50ms, write the secondsField value in the time information of the master clock node directly to the local clock;
(2) If Tdiff is greater than or equal to 10ms and less than 50ms, gradually reduce Tdiff by adjusting the EPIT reset time, adjusting 10ms at a time;
(3) If Tdiff is greater than or equal to 1ms and less than 10ms, gradually reduce Tdiff by adjusting the EPIC reset time, adjusting 1ms at a time;
(4) If Tdiff is greater than or equal to 100us and less than 1ms, gradually reduce Tdiff by adjusting the EPIT reset time, adjusting 10us at a time;
(5) If Tdiff is greater than or equal to 10us and less than 100us, gradually reduce Tdiff by adjusting the EPIT reset time, adjusting 10us at a time;
(6) If Tdiff is greater than or equal to 2us and less than 10us, gradually reduce Tdiff by adjusting the EPIT reset time, adjusting 1us at a time;
(7) If Tdiff is less than 2us, do not adjust.
The specific process for the pair is shown in Fig.4.

Fig.4 Timing Process

5. Conclusion
Based on the results and discussions presented above, the conclusions are obtained as below:
(1) It is shown that Clock synchronization can be achieved through the FlexRay bus.
(2) Through the oscilloscope measurement, the time difference between the second pulse output by the master clock node and the slave clock node is 10us. Therefore, it is concluded that the synchronization accuracy of the clock synchronization realized by FlexRay is 10us.

References
[1] FlexRay Bus Communication Protocol Chinese version. https://max.book118.com/html/2017/0524/108870152.shtml.
[2] Zhirui Chen. The realization of FlexRay bus and its application in automobile[D].ShangHai:TongJi university,2009:31-32.
[3] Qitao Guo. FlexRay bus features and its application in vehicles and fault analysis[J]. Journal of Henan Science and Technology, 2014(11): 1-2.
[4] Freescale. MC9s12XF512 Reference Manual. 2010-5-18.
[5] Xiaochen Dai. Research and design based on FlexRay of brake-by-wire[J]. Electronic Measurement Technology, 2017, 40(12): 6-10.
[6] Biao Liu, Zhiqiang Hu, Guangchuan Zhen. Simulation research of Automobile Wire Control System based on FlexRay Bus[J]. DISTRIBUTION & UTILIZATION, 2018(1): 3-4.