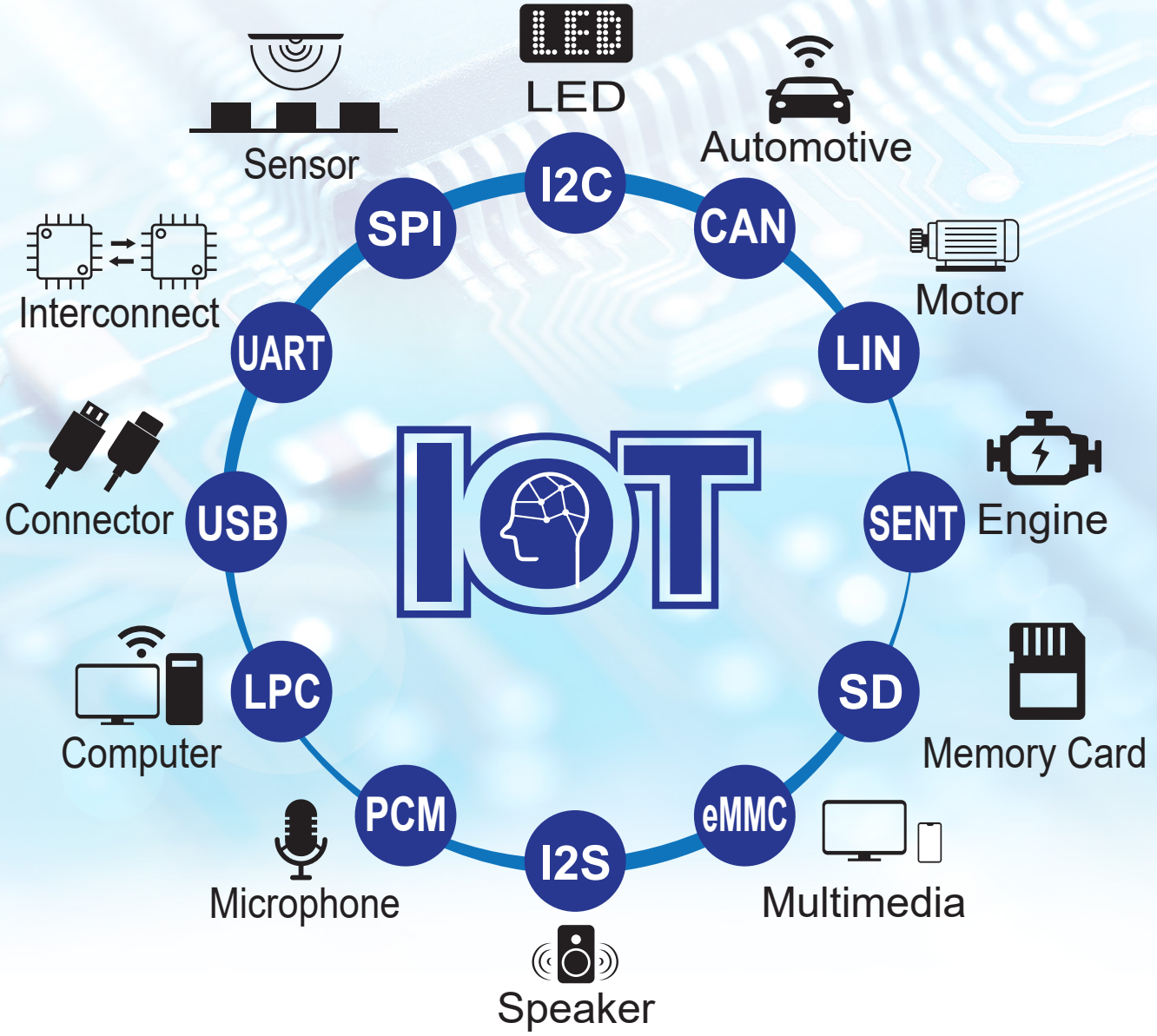


Debugging protocols
for your IoT Design

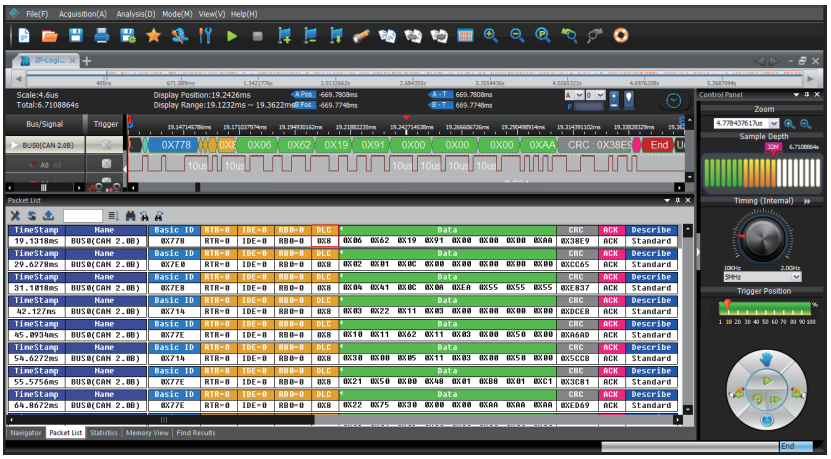
With the development and popularization of IoT technology, more and more devices and systems are becoming intelligent and interconnected, forming a seamlessly integrated IoT ecosystem. This trend has promoted the development of IoT protocols and lowered industry barriers for protocol applications. When IoT products gradually meet requirements such as scalability, interoperability, security, low power consumption, and real-time data transmission, it also brings unprecedented opportunities and challenges to IC design.

IoT devices often come from different manufacturers and use different protocols and technologies, so designers need to consider the compatibility and interoperability between different devices and protocols, which increases the complexity of design and development. Most of the time manufacturers would choose to place multiple communication protocols in the control system at the same time to guarantee the interoperability of different devices in the system, so that the new system can be better controlled and run more intelligently.

Replacing the bus is not always the best choice, most of the time manufacturers will choose to place multiple communication protocols in the control system at the same time to ensure the interoperability of different devices in the system, so that the new system can be better controlled and updated operate intelligently.

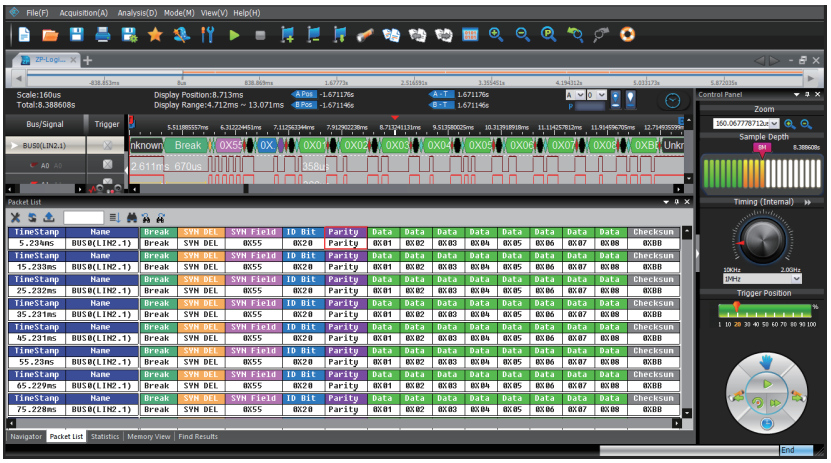
The evolution of bus in automotive electrical systems in recent years is a prime example. In order to meet the real-time requirements of each electronic system, the public data of the car (such as engine speed, wheel speed, accelerator pedal position, etc.) must be shared, and each control unit has different real-time requirements. Many vehicle designs use CAN, LIN or FlexRay to communicate between electronic control units (ECUs) and between ECUs and sensors, actuators and displays. These buses are critical to providing real-time communication within and between important subsystems, from braking systems to infotainment systems.

CAN is a multi-master serial bus standard for connecting electronic control units (ECUs), also known as nodes (automotive electronics is a major application area). Two or more nodes are required on a CAN network to communicate. Nodes range from simple digital logic to devices, such as PLDs, to FPGAs, to embedded computers that run large amounts of software. Such a computer can also be a gateway, allowing a general-purpose computer (such as a laptop) to communicate with devices on the CAN network via a USB or Ethernet port.



(CAN)

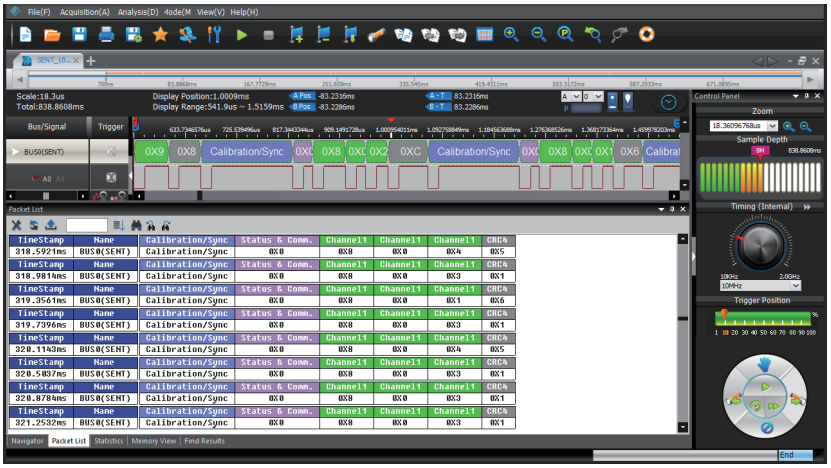
The LIN-Master uses one or more predefined schedules to initiate transmission and reception on the LIN bus. These schedules contain at least the relative times at which message transmissions are initiated. A LIN frame consists of a header and a response. The header is always sent by the LIN master; while the response is sent by a dedicated LIN slave or the LIN master itself.



(LIN)

While emerging technologies continue to evolve, that doesn't mean automakers will replace all legacy auto parts or adopt newer methods of serializing parts in one new car. How to create the greatest economic benefits with the least modification is often the most concerned issue for auto parts designers. This is why the SENT (Single Edge Nibble Transfer) bus is becoming more and more popular in the automotive industry.

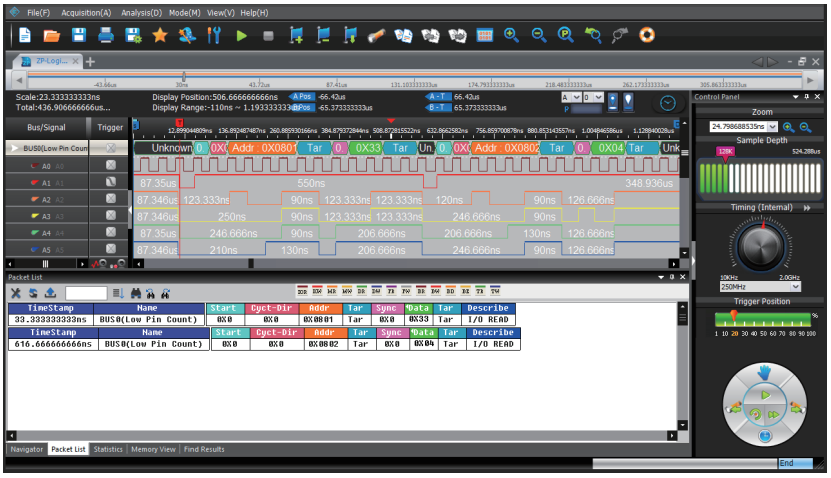
The SENT bus is used to transfer high-resolution sensor data from the sensor to the ECU. It can replace lower resolution methods and is a simpler and lower cost alternative to CAN or LIN. SENT is mainly used for the one-way data transmission protocol from the peripheral sensors of the car to the engine control unit, including sensors such as pressure, airflow, position, and temperature. It is intended to allow for transmission of high resolution data with a low system cost. After the emergence of the SENT protocol, equipment manufacturers can meet a variety of needs by modifying the design with a small amount, reducing the cost of the engine control unit.



(SENT)

The rise of the LPC bus in the PC industry is similar to that of SENT. Not only is LPC capable of connecting low-bandwidth and "legacy" devices to the CPU, but it has completely replaced the Industry Standard Architecture (ISA) bus as it is more suitable for battery-operated devices.

The LPC bus was introduced by Intel in 1998 as a software-compatible replacement for the Industry Standard Architecture (ISA) bus. It is similar to what the ISA is to software, although physically very different. The ISA bus has a 16-bit data bus and a 24-bit address bus, which can be used for either 16-bit I/O port addresses or 24-bit memory addresses; both run at speeds up to 8.33 MHz. The LPC bus uses a highly multiplexed four-bit wide bus running at four times the clock speed (33.3 MHz) to transfer address and data with similar performance. In addition, the individual interrupt request (Interrupt Request, IRQ) signal of the peripheral device is also replaced by a SERIRQ signal, and some related control signals for power saving are added, including CLKRUN#, PME# and LPCPD#, etc. Therefore, the LPC bus interface not only runs faster, but also reduces the number of signal lines by about 30 signal lines compared with the ISA bus interface, and the system power consumption is also more suitable for mobile devices that use batteries.



(LPC)

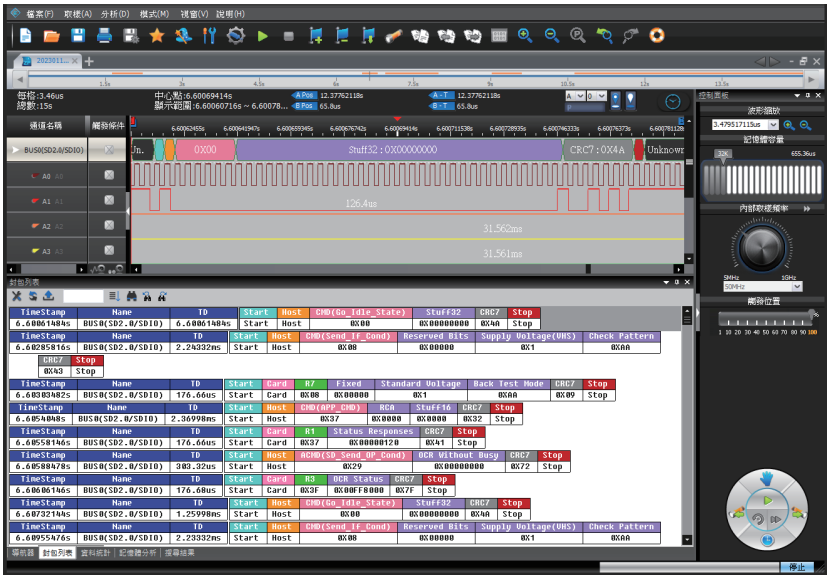
Creating a digital circuit or system that meets the specific requirements and specifications of an intended end use means ensuring that the design is reliable, efficient, and meets the performance, functionality, and other criteria required for the intended application. For example, if the target application is a digital temperature control system for a greenhouse, the design needs to be reliable, efficient, and meet the performance and functional criteria required by the application; if the target application is medical equipment, the design must meet safety, reliability, and performance aspects regulatory requirements. Delivering a high-quality design equals producing a design that is reliable, efficient, and meets the performance and functional requirements of the target application. This requires a focus on design quality throughout the design process, including verifying and validating the design, testing and debugging to ensure the design works as expected.

For IC designers, behind every groundbreaking functional design is the possibility of designing with unfamiliar buses in mind. Integrating a well-established bus standard into a schematic design is less risky and cost-effective than customizing a bus from scratch, although unfamiliarity with bus bars can cause various situations to arise during verification of the design.

Taking eMMC as an example, this embedded memory standard specification originally defined by MMC (Multimedia Card Association) for mobile phones or tablets can improve data security and simplify memory design due to its high-capacity flexibility——eMCP Embedded Multi-memory card chip packaging, NAND flash memory chip and control chip are packaged on one chip through MCP technology - not only can effectively manage large-capacity flash memory, save the circuit board area occupied by components, but also reduce the calculation amount of the main chip. Therefore, in recent years, eMMC has begun to be applied to new products in industries such as automobiles, displays, and smart homes.

(eMMC)

Secure Digital is also a memory card standard, referred to as SD. Based on the multimedia card (MMC) format, not only the random performance (Random Performance) or continuous performance (Sequential Performance) is better than eMMC, but also the storage data can be set to use permissions to prevent stolen data from being copied. Therefore, in the development of IoT devices with higher privacy and security requirements, such as digital cameras, drones, and multimedia players, it is more suitable to apply SD storage standards to product design.

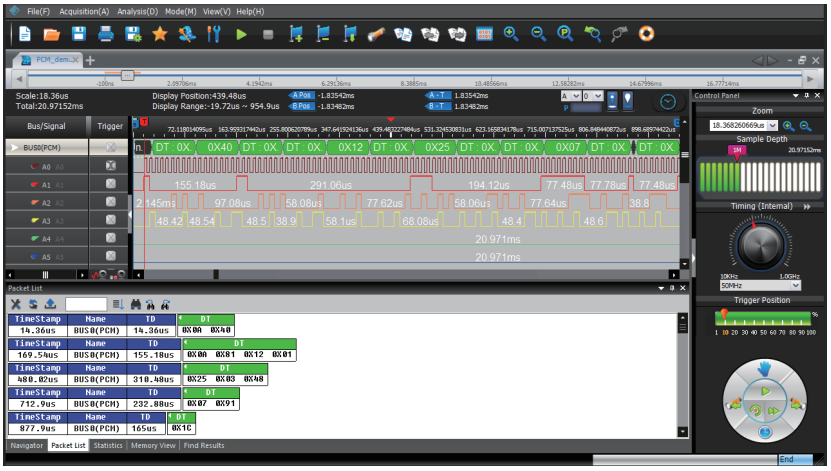


(SD)

Compared with general electronic products, IoT devices take the audio system into product design considerations more. Audio systems come in a variety of configurations depending on the intended use of the product. Consumers using portable communication devices such as cell phones, wireless headsets, and laptops with integrated webcam microphones and Voice over Internet Protocol (VoIP) accessories demand the ability to hear each other in noisy environments without speaking aloud, and of not be affected by echo. Some systems preserve analog audio for as long as possible, using analog microphone and speaker amplifiers with minimal digital processing in between. Some systems choose to take digital audio data without any analog audio signal, process it, and output it to another system. Audio compression processing, audio data communication and conversion between audio signals can be difficult for designers unfamiliar with codec ICs to verify designs.

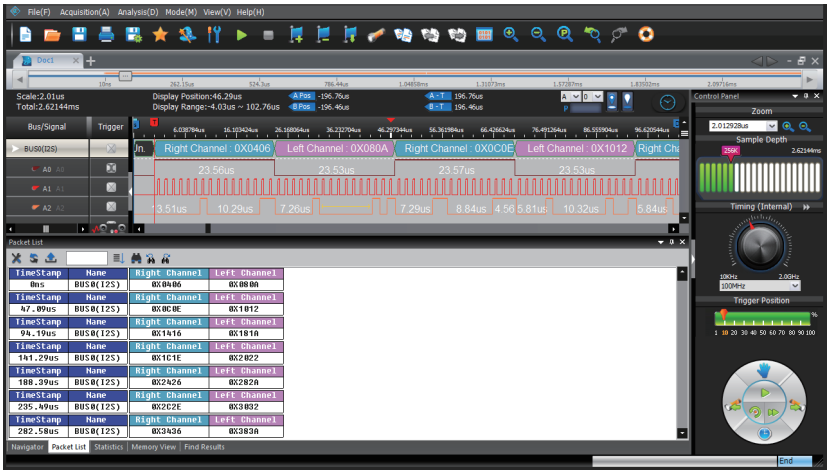
Audio codec IC, which refers to an IC capable of audio exchange and conversion, usually supports additional protocols such as PCM, I2S, and MIPI, providing developers with greater flexibility. Some audio devices choose to use the TDM and PCM interface options for audio data communication, and some choose to use the I2S bus. The I2S bus is simple and effective, can effectively improve the quality of the output data, and is widely used in various embedded audio systems. However, in embedded audio system design, not all MCUs support the I2S bus format, and I2S does not have a unified interface standard, so the inter-integrated circuit (I2C) and serial peripheral interface (SPI) are often used as Communication protocol with I2S.

PCM Pulse-code modulation, referred to as PCM, is mainly used for audio coding. It is an ADC encoding method that uses Pulse to convert an analog signal into a digital signal. PCM uses a special modulation and demodulation method to realize the multi-channel function, divides the signal strength into several segments according to the same interval, and then quantizes it with special digital symbols. Data obtained through analog-to-digital techniques such as PCM is often used to store the original recording.



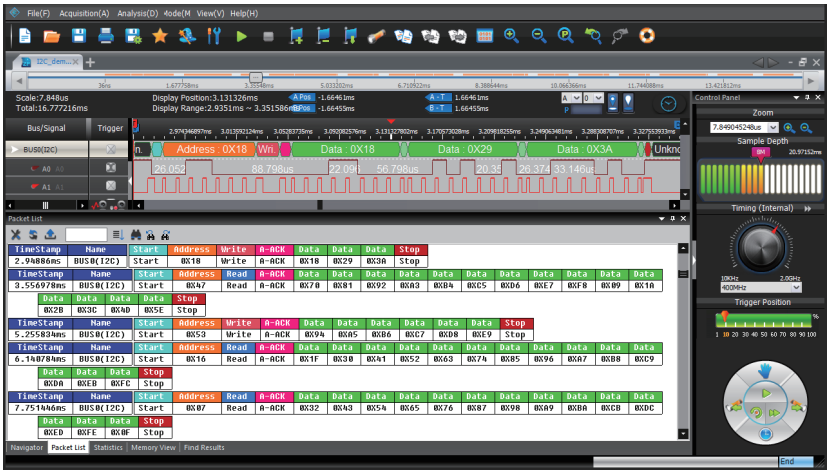
(PCM)

I2S is a serial bus interface designed to transfer digital audio data between integrated circuits (ICs). It is commonly used to transfer PCM audio data from a CD to a CD player's DAC. I2S transmits 2 sets of data (left and right channels) in sequence, consisting of 3 transmission lines: clock line (SCLK), selection line (LRCK), and data line (SDATA). Three different data formats have been developed, the Philips standard, the left-justified standard, and the right-justified standard. The I2S protocol sends pulse code modulation (PCM) audio data from the controller to the target. It has at least 3 lines: bit clock line, word select line and data line. The word select is used to specify which stereo channel the data should be sent to, the left or the right.



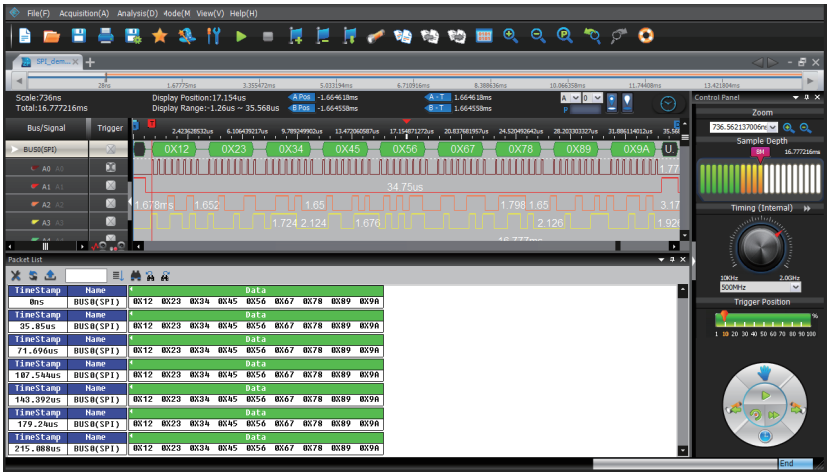
(I2S)

I2C (Inter-Integrated Circuit) is a field-programmable serial bus commonly used to communicate with other microcontrollers or peripherals. It is a simple and fast communication interface developed by NXP Semiconductors. Support multi-master and multi-slave applications. The transmission rate is generally below 400kbit/s. I2C consists of two wires: a serial data line (SDA) and a serial clock line (SCL). The main advantage of the I2C bus is its simplicity and efficiency. Because the interface is directly on the components, the space occupied by the I2C bus is very small, which reduces the space of the circuit board and the number of chip pins, and reduces the cost of interconnection, so it is widely used in the design of the IoT.



(I2C)

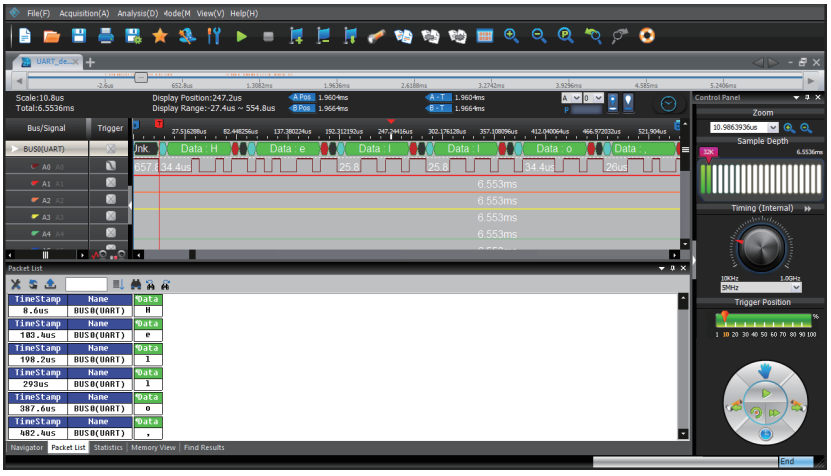
SPI (Serial Peripheral Interface) is a serial bus protocol commonly used for communication within microcontrollers and the connection between microcontrollers and other low-speed devices. The SPI consists of four lines: Master Out/Slave In (MOSI), Master In/Slave Out (MISO), Serial Clock (SCK), and Chip Select (SS) lines. The advantage of the SPI protocol is that the data transmission speed is fast, and data transmission can be realized in various ways.



(SPI)

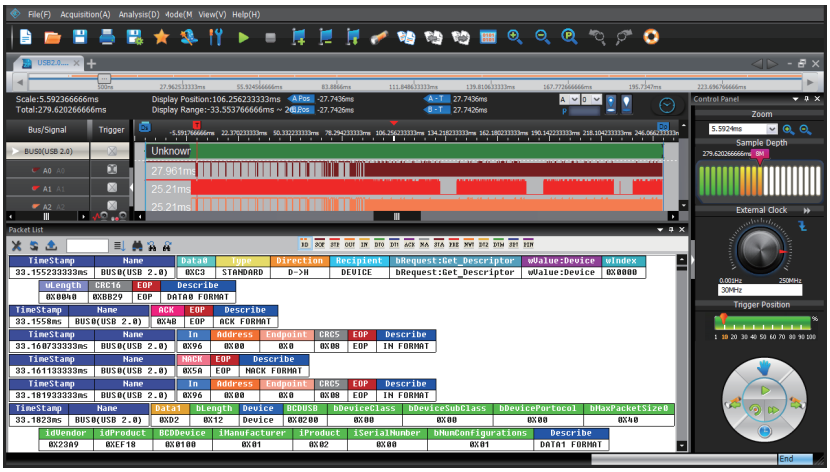
When IC designers design chips, they need to consider how to balance device functions and power consumption, compatibility and interoperability between devices, and issues such as data storage, data processing, and data analysis. The Internet of Things is designed so that various devices and sensors communicate with each other to enable real-time monitoring. As the communication network becomes more and more complex, the requirements for the data carrying capacity and data processing capacity of the chip will also become higher and higher. Among them, the three most commonly used communication protocols are I2C, SPI, and UART. Compared with i2c and spi, which are communication protocols, UART is biased towards the physical circuit or independent ic in the microcontroller.

UART (Universal Asynchronous Receiver/Transmitter) is an asynchronous communication protocol, often used for serial data transmission. A UART has only two wires, a transmit line and a receive line. It does not require a clock line and the data transfer rate is set by the baud rate. The UART is capable of converting data from serial to parallel transfers, requiring only a fixed serial transfer rate. The UART protocol is simple and easy to use. It can form a chip independently, or it can be used as a module embedded in other chips, such as RS232, RS422, RS485 and other interface standard specifications.



(UART)

USB (Universal Serial Bus) was originally initiated by Intel and Microsoft, and its biggest feature is to realize hot-swapping and plug-and-play as much as possible. When the device is inserted, the host enumerates the device and loads the required driver, so it is far more convenient to use than buses such as PCI and ISA. The design of USB is asymmetrical, and it consists of a host controller and several devices connected in a tree through a hub device. There can be up to 5 levels of Hubs under one controller. Including Hubs, up to 128 devices can be connected, because a 7-bit addressing field is used in design, and 2 to the 7th power is equal to 128. Most people say that there are 127 USB connections, which means that when connecting (a certain device), the USB interface connected to the host needs to be deducted, and a computer can have multiple handles at the same time. Peripherals that can be connected to USB include mice, keyboards, gamepads, joysticks, scanners, digital cameras, printers, hard drives, and network cards. The USB2.0 interface has been widely used in PC/NB peripheral devices, and even mobile devices for external device output/input applications, and has become the basic equipment of contemporary PCs.



(USB2.0)

During the verification and test design process, how to effectively collect, manage and analyze the large amount of data generated by IoT devices is also a challenge. To solve problems and challenges in IoT, designers need to select the appropriate design solutions and tools for each situation. Generally speaking, IoT debugging methods can be divided into software, hardware, and firmware.

(1) Software debugging method

Software debugging methods include debuggers, logs, trace tools, etc. Debuggers help developers monitor code execution, detect errors, and assist in correction. Logs can record information during software execution, allowing developers to better understand where the problem lies. Tracing tools can help developers locate problems and find bugs in software. This year a new tool - ChatGPT came out. More data and adjustments will be needed to confirm how helpful it can be

(2) Hardware debugging method

Hardware debugging methods include test instruments, analysis tools, simulators, etc. Test instruments can help engineers detect hardware problems, analysis tools can help developers gain insight into system operation, and simulators can help developers simulate different hardware environments to test system stability and reliability.

(3) Firmware Debugging Method

Firmware debugging methods include protocol analyzers, module solutions, transceivers, etc. Protocol analyzers can help engineers follow the adoption of the most forward-looking protocols, module solutions can help developers achieve higher-demand specification areas, and transceivers can help developers verify system compliance

Among them, the logic analyzer is a common tool that can help engineers solve many problems, speed up the development process, and improve the reliability and stability of the design. With the development of electronic technology, there are more and more serial signals around us. However, front-line R&D engineers also need to have corresponding R&D tools. Zeroplus logic analyzer supporting 130 protocols can be an invaluable tool for IC designers by providing debug, protocol analysis, performance optimization, verification and verification, and troubleshooting capabilities, helping them work more efficiently and deliver the high quality that meets their application needs Design goals.

ZEROPLUS Technology Logic Analyzer has developed a variety of serial communication analysis protocol modules, which can help engineers quickly analyze signal content, whether it is automotive electronics, multimedia audio, IC interface, computer system, memory, etc. Long-term recording function The logic analyzer uses Stream technology to capture the signal and transmit it to a computer or laptop for storage through the USB3.0 interface. The storage space can completely record signals lasting for hours or even days, and the recorded data can be taken away conveniently. Most importantly, there is no need to set complicated trigger conditions to use the long-time recording function. Development engineers can easily perform operations such as search, statistics, and analysis on data, so as to achieve greater success in their careers and gain greater recognition in the industry.

YouTube facebook ZEROPLUS

HeadQuarter：
2F., No. 123, Jian 8th Rd., Zhonghe Dist.,
New Taipei City, 23585, Taiwan (R.O.C.)
Contact Email：service@zeroplus.com.tw
ZEROPLUS
www.zeroplus.com.tw