

Device driver is a software program which controls a specific embedded device. These drivers can be either common or separate for each device function. The important features of the device driver are,

1. **Software Interface**

Device driver acts as a software interface between the application and the physical device.

Devices are used during the execution of an application. Device driver helps in providing a 'routine', in order to make use of a device function in the application.

For instance, consider an application of transferring a stream of bytes. The transfer of data is accomplished using a network driver card, after packing the complete data (depending on the type of protocol used in the different layers). This use of network driver card is facilitated by the network drive routine, which provides a software layers between the application and network. Thus, device drivers helps in interfacing the application with a physical device.

2. The Device Drivers Help in Utilizing a Device by Simply Executing an ISR

Functions of a driver are written in such a way that it becomes easy for an application developer to use it. Thus, the device is then driven using a simple command from a task.

As soon as a driver function is made available for writing a software code, the application developer need not know any details regarding the device (such as mechanism, addresses, registers, bits and flags of the device).

For instance, consider a system clock which is to be set to tick 100 times in a second. The user developer accomplishes this task by making a 'call' to an operating system function such as `OS_TICKS(100)`. As soon as the `OS_TICKS(100)` is run, it interrupts the system and runs the SWI instruction. This instruction inturn calls the driver ISR (i.e., signalled routine) for the system ticking device. The driver ISR then assumes '100' as its input and configures the real-time clock such that it ticks for about 100 times per second.

3. The functions of a generic device driver (such as open, close, read, write, listen accept in high-level language) are used in high-level language program.
4. The programming of a device driver ISR requires the complete details of the processor, system, IO buses and register address in the particular hardware. It necessitates the extensive understanding of the following,
 - (i) The method of seeking the device data by software application program.
 - (ii) The way of writing device data using software application program.
 - (iii) The type of platform (i.e., operating system and hardware interfacing the system bus) used in the system.

SIA GROUP

