1) **What is Linux IIO subsystem? Please explain IIO in terms of its purpose and user interface provision.**

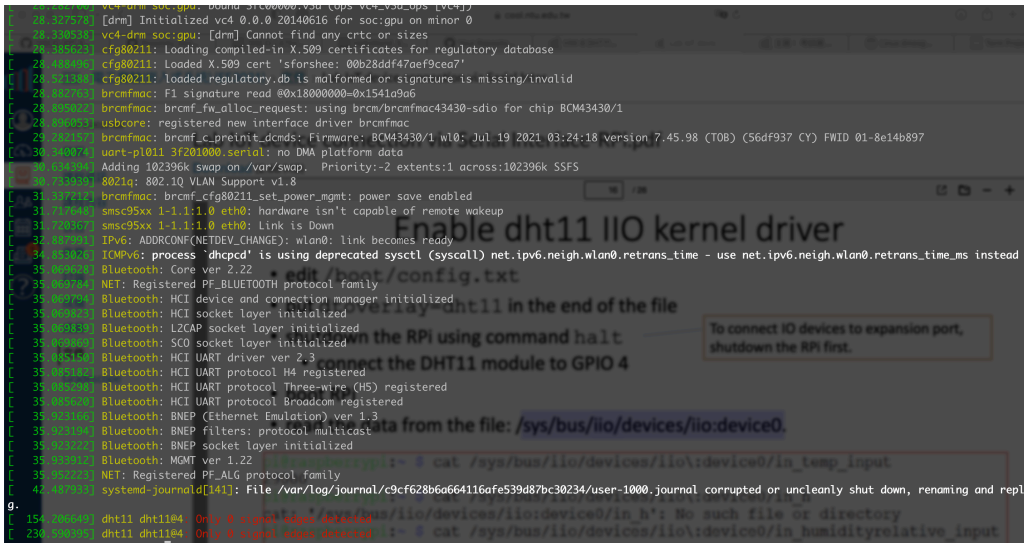
The Linux IIO subsystem is a part of the Linux kernel that provides support for devices that perform I/O operations in an industrial context. This includes a wide range of devices like sensors, ADCs (Analog to Digital Converters), DACs (Digital to Analog Converters), and other similar types of hardware.

The user interface provision in the IIO subsystem is likely through the sysfs (system file system) interface, as indicated by references to "IIO buffer sysfs interface" and "IIO trigger sysfs interface". Sysfs in Linux is a virtual file system that provides a window into the kernel, allowing user-space programs to interact with kernel components.

This subsystem is crucial for applications that require precise and reliable data acquisition and control in industrial settings, making it a key component for developers working with such hardware on Linux systems.

2) **How is the efficiency difference when compared between interrupt-driven I/O and programming I/O (polling I/O)?**

In programmed I/O, the CPU actively checks (polls) the status of the peripheral device to determine if it's ready for an I/O operation. This involves repeatedly reading the device's status register. On the other hand, In interrupt-driven I/O, the CPU is freed from the task of continuously checking the device status. Instead, the I/O device sends an interrupt signal to the CPU when it's ready for an I/O operation. Therefore, the interrupt-driven I/O allows for better CPU utilization and typically results in more responsive systems.



We thought something went wrong with either Rpi or DHT11. However, by tracing the source codes, we get to see the difference between interrupt-driven I/O and programming I/O. For interrupt-driven method, when an edge (falling and rising) is detected, a ISR is triggered to store the time stamp and the value. The data is later decoded to obtain the sensor value. On the other hand, the polling method continuously checks for the updates from pin, rendering CPU idle and busy waiting.