



Contribution ID: 189

Type: **not specified**

A case for a generic Linux driver for connecting MCUs to MPUs

Friday, 20 September 2024 18:10 (20 minutes)

In Linux based IOT embedded applications there has always been this ongoing desire to attach MCUs (Micro-Controller Unit) to MPUs (Micro-Processor Unit) running Linux. The usual reason is that the MCU is able to handle low latency data processing more efficiently than the higher-level functioning MPU. The MCU might also add a missing peripheral on the MPU that is more system cost efficient. The data passed between the two processors can be as simple as a couple of register values to something more complex such as streaming low-level protocol traffic. Typically, a specific driver for a specific MCU has to be developed and up streamed before a connection can be made. This can restrict choices for a project that can impact overall development time. This presentation is about proposing a generic device driver to be up streamed for connecting MCUs from an application perspective to an MPU that would allow the Linux side to provide a common framework for connecting MCUs. The MCU side would have a framework to develop code to pass data with.

As part of the presentation an example application will be discussed that shows connecting an MCU collecting data that is then passed to a Linux application. As part of the discussion the complexity of the data being passed will be considered and whether the data type and amount is not suitable for a generic approach. The overall goal would be to have a generic kernel driver that would attach over a peripheral interface such as SPI and be able to communicate with the MCU without having a dedicated driver.

A key capability of this driver would enable application developers to start their project without having to wait for a specific up streamed kernel driver.

Primary author: PATTON, Schuyler (Texas Instruments - Embedded Processing)

Presenter: PATTON, Schuyler (Texas Instruments - Embedded Processing)

Session Classification: Internet of Things & Embedded MC

Track Classification: Internet of Things & Embedded MC