

# mikroBUS Driver for Add-on Boards

*Thursday, 27 August 2020 07:05 (40 minutes)*

mikroBUS is an add-on board socket standard by MikroElektronika that can be freely used by anyone following the guidelines. The mikroBUS standard includes SPI, I2C, UART, PWM, ADC, GPIO and power (3.3V and 5V) connections to interface common embedded peripherals, there are more than 750 add-on boards ranging from wireless connectivity boards to human-machine interface sensors which conform to the mikroBUS standard, out of which more than 140 boards already have device driver support in the Linux kernel. Today, the most straight forward method for loading these device drivers is to provide device-tree overlay fragments at boot time, this method suffers from the need to maintain a large out-of-tree database for which there is a need to maintain a separate overlay for every mikroBUS add-on board for every mikroBUS socket and also for targets that do not support dynamic loading of overlays, this method requires at-least a single reboot to enable the support in a potentially error-prone way.

The mikroBUS driver tries to solve the problem by introducing a new pseudo-bus driver(pseudo-bus since there is no actual bus controller involved) which enables the mikroBUS as a probeable bus such that the kernel can discover the device(s) on the bus at boot time, this is done by storing the add-on board device driver-specific information on a non-volatile storage accessible over one of the buses(currently on the mikroBUS I2C bus, subject to change) on the mikroBUS port. The format for describing the device driver-specific information is an extension to the Greybus manifest, the choice of using the Greybus manifest for the purpose is not entirely coincidental; there is ongoing work to evaluate ways to add mikroBUS sockets and devices via Greybus expansion and the manifest format can describe the device driver-specific data in a fairly good manner. With more than 100 clicks with tested support now, the mikroBUS driver makes use of the Unified Properties API and GPIO lookup tables for passing named properties and named GPIOs to device drivers. There are already several Linux platforms with mikroBUS sockets and the mikroBUS driver helps to reduce the time to develop and debug support for various mikroBUS add-on boards. Further, it opens up the possibility for support under dynamically instantiated busses such as Greybus.

## I agree to abide by the anti-harassment policy

I agree

**Primary author:** Mr M A, Vaishnav (Beagleboard.org)

**Session Classification:** You, Me, and IoT Two MC

**Track Classification:** You, Me, and IoT Two MC