beagleboard.org

mikroBUS Driver for Add-on Boards

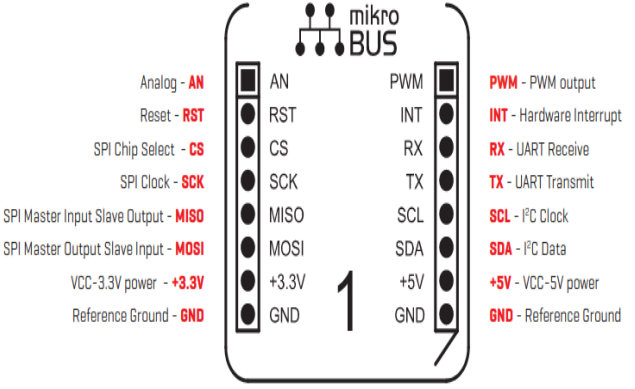
Vaishnav M A <vaishnav@beagleboard.org>

You, Me and IoT Two MicroConference

LINUX PLUMBERS CONFERENCE / August 24-28 2020

mikroBUS™ add-on board socket standard

- Open Standard from MikroElektronika
  - can add mikroBUS™ Sockets to board designs.
  - can design new add-on boards.
- 140+ Development Boards Supported.
- 850+ Add-on Boards.
- not excessively flexible and require abritrary device-tree overlays like Cape/Hats/Shields.
- 150+ add-on boards already have corresponding device drivers within the linux kernel.



https://www.mikroe.com/mikrobus

Usage of mikroBUS comaptible add-on boards today

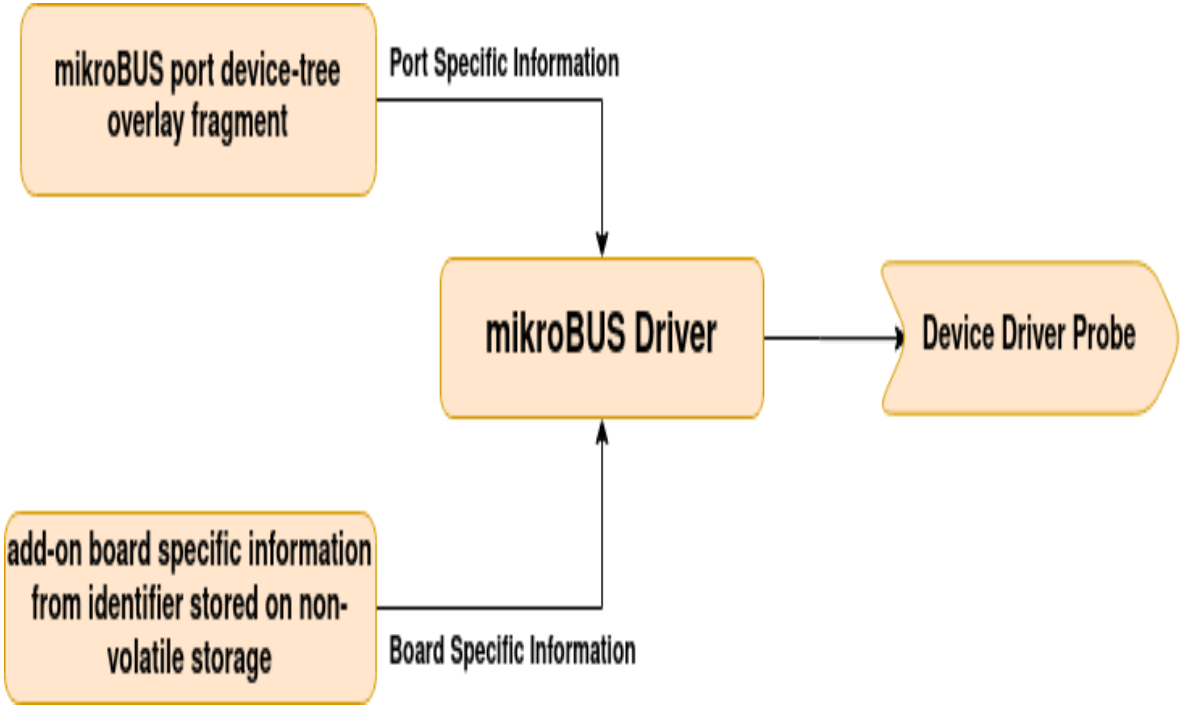
- 150+ add-on board has existing device drivers on the kernel.
- Device-Tree Overlays loaded at boot time.
- requires maintenance of overlay fragments specific to each add-on board for each socket.*
- Considering a BeagleBoard.org PocketBeagle target alone(2 mikroBUS socket) with the 150 add-on boards which has existing device drivers, 300 different device tree overlay fragment needs to be maintained.
- Time consuming and potentially error-prone.

* Device Tree maintainer Frank Rowand has an active proposal to solve the problem of maintaining overlay fragments for each socket, but this might not apply for mikroBUS sockets added dynamically.(discussed later)

mikroBUS driver in the Linux Kernel

- mikroBUS as a probeable bus, kernel can discover the device(s) on the mikroBUS at boot time.
- Add-on board specific identifier information stored on a non-volatile storage on the add-on board.
- Currently the identifier is stored on an EEPROM on the I2C bus on the mikroBUS socket.
- Identifier format used is an extension to Greybus Manifest <linux/greybus/greybus_manifest.h>.
- Support tested for 110 mikroBUS add-on boards.
- MikroBUS v3 standard under development.

mikroBUS driver in the Linux Kernel



Add-on board identifier : Extension to Greybus Manifest

- data structure which contains the manifest header along with a set of descriptors, used within greybus to describe the capabilities of a module.
- Source similar to INI style configuration.
- existing descriptors cannot accurately describe a device with enough information to perform the device-driver probe.
- majority of the add-on boards require interrupt information for sucessful device driver probe, few require additional named properties, few require named GPIOs.
- Addition of new descriptors : mikrobus-descriptor, device-descriptor, property-descriptor, uses same manifesto tool to generate the manifest binary.

mikrobus-descriptor

- to describe the prior setup(pinmux, GPIO states) required on the mikroBUS for the add-on board.
- Each pin can take up its own default state or a GPIO state.

```
[mikrobus-descriptor]
pwm-state = 4
int-state = 1
rx-state = 7
tx-state = 7
scl-state = 6
sda-state = 6
mosi-state = 5
miso-state = 5
sck-state = 5
cs-state = 5
rst-state = 2
an-state = 1
```

device-descriptor

- Used to describe a single device on an add-on board, contains fields from spi_board_info and i2c_board_info, can also fairly describe other devices relevant to the mikroBUS(serdev, platform devices).
- Fixed length descriptor, can have multiple instances of in a single add-on board manifest.

```
[device-descriptor 1]
driver-string-id = 3
protocol = 0x0b
mode = 0x0
reg = 0
max-speed-hz = 16000000
irq = 1
irq-type = 0x2

[string-descriptor 3]
string = enc28j60
```

property-descriptor

- Used to pass additional named properties/GPIOs to device drivers, in case of properties the information is mapped to a struct property_entry under <linux/property.h> , in case of named GPIOs, the mikroBUS driver will map the values to the actual GPIOs and add a GPIOD lookup table for the device.

```
[property-descriptor 2]
name-string-id = 5
type = 0x05
value = <262144>

[string-descriptor 5]
string = size
```

MikroBUS ports over Greybus : Proposal

- Dynamically adding mikrobus ports over greybus has the advantage of using different transports.
- Addition of new bundle class.
- The mikroBUS bundle will have Cport with protocols corresponding to each of the Busses(SPI,I2C,UART) and other peripherals(PWM, GPIOs) on the mikroBUS socket.

Demo

- BeagleBoard.org PocketBeagle mikroBUS ports.
- PocketBeagle with Techlab Cape.
- BeagleBone Black Wireless with mikroBUS Cape.
- Writing manifests for mikroBUS add-on boards.

Open Problems/Challenges

- Device Drivers not using Unified Device Properties API to fetch properties :

  drivers which uses device_property_read_* calls work but of_propert_read_u32 will not work , similarly devm_gpiod_get will work but of_get_named_gpio will not work.

- Device Drivers for which the corresponding device tree entries have child nodes cannot be described through property descriptors.

- Pinmux handling when adding mikroBUS ports through Greybus

TODO

- SERDEV Devices Support (GNSS Add-on Boards)
- Update the identifier fetching mechanism according to mikroBUS v3 standard (add-on boards with on-board identifier mechanism).
- Greybus Bundle Class
- Documentation

beagleboard.org

THANK YOU

You, Me and IoT Two MicroConference

LINUX
PLUMBERS CONFERENCE / August 24-28 2020