LINUX PLUMBERS CONFERENCE September 20-24, 2021

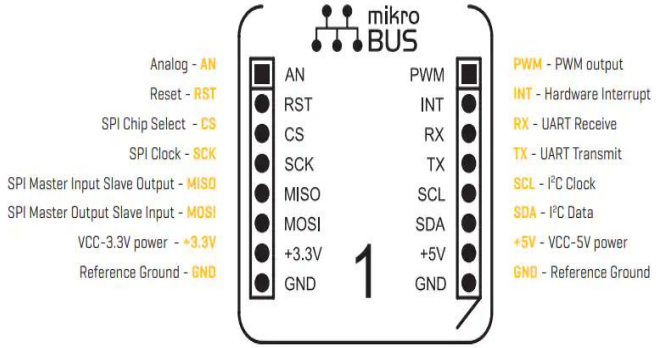
mikroBUS Driver for Add-on Boards

Vaishnav M A <vaishnav@beagleboard.org>

IoThree's Company MicroConference

mikroBUS™ add-on board socket standard

- Open Standard from MikroElektronika
  - mikroBUS™ sockets can be added to board designs
  - new add-on boards can be designed.
- 350+ development boards have adopted mikroBUS
- more than 1000 add-on boards available.
- 150+ Add-on board devices have existing device driver support in kernel.



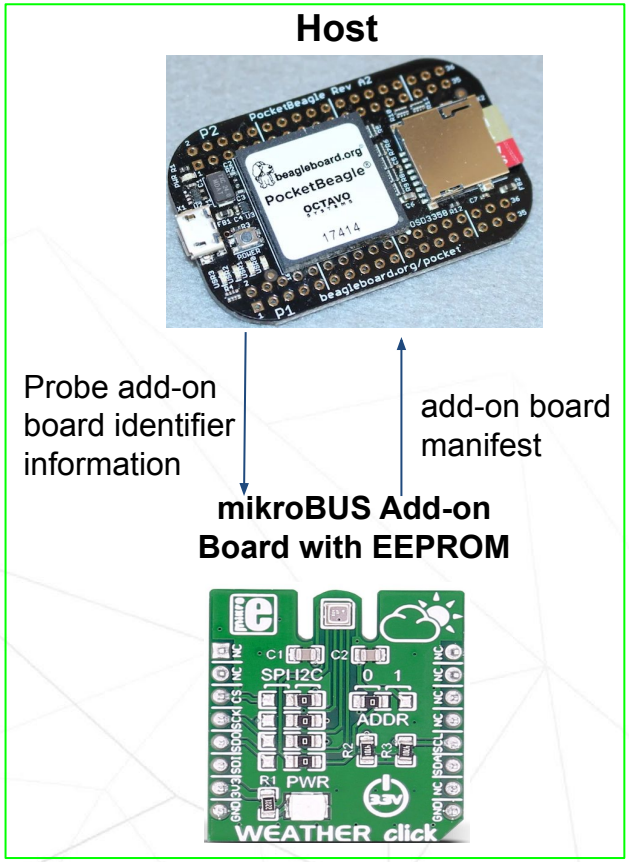
https://www.mikroe.com/mikrobus

- **Newer versions of add-on boards will have identifier EEPROMs.**

Problem Statement

- **Expose mikroBUS as a probe-able bus**
  - enabling host to identify the devices connected to mikroBUS socket and probe corresponding device drivers.

- **Create/re-use an add-on board identifier standard**
  - must be able to describe information required for add-on board device driver probe.
  - does not have any board(host) specific information.
  - needs to support devices on buses that doesn't have device-tree support today. Eg. greybus



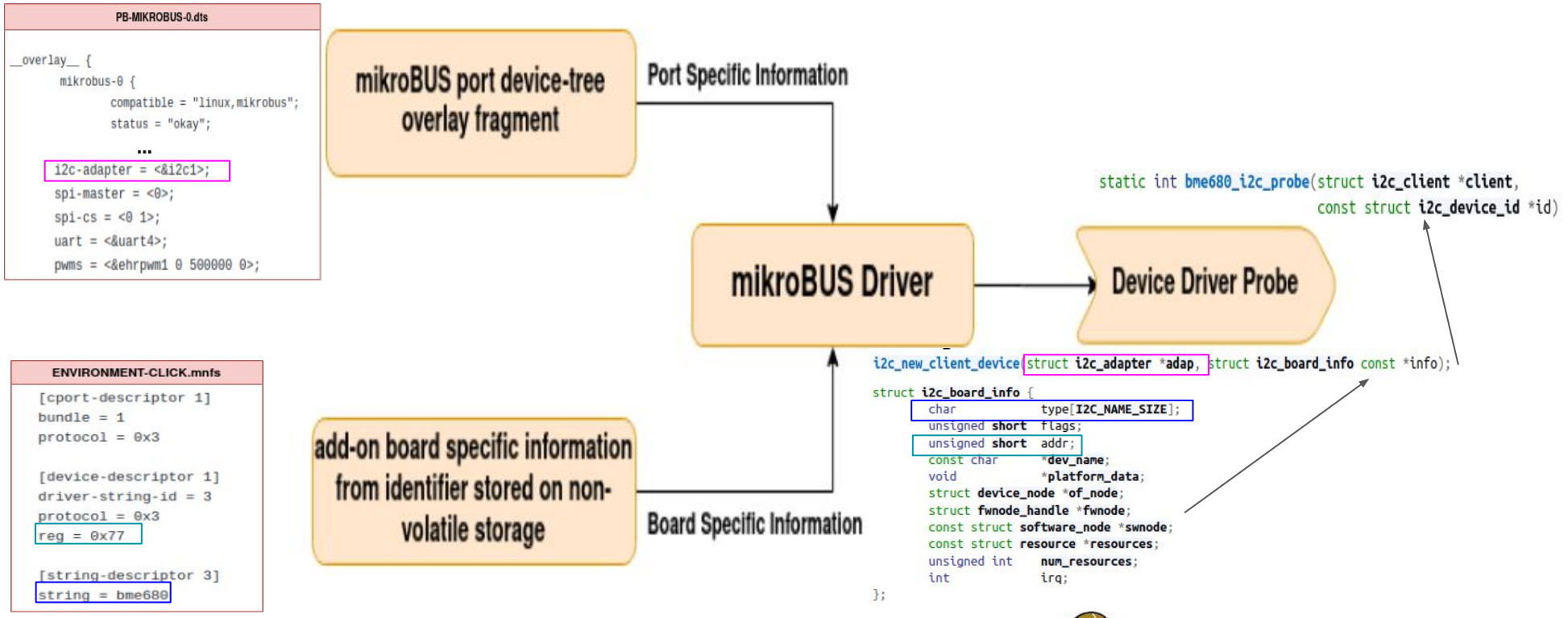
Recap from LPC2020

- mikroBUS driver overview
- draft version of identifier mechanism
- extension to greybus manifest, custom descriptors
- proposal for adding mikroBUS ports over greybus
- demo covering add-on boards connected directly to host mikroBUS sockets

Topics Covered Today

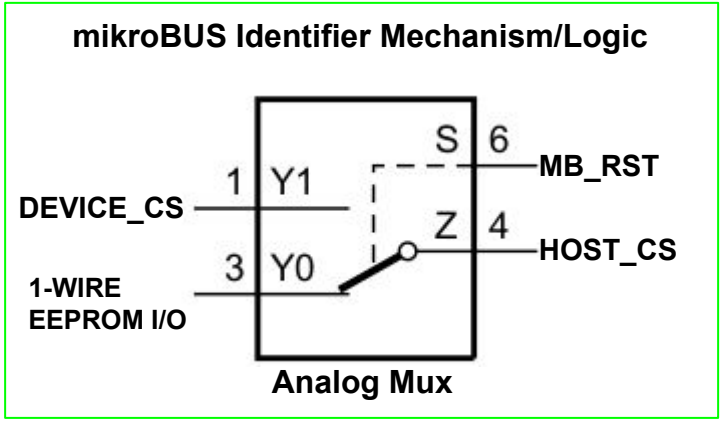
- mikroBUS Identifier Mechanism
- mikroBUS ports over greybus
- greybus manifest patching and changes made in Zephyr
- BeagleConnect™
- demo covering local & remote (over greybus) mikroBUS add-on boards.

Recap: mikroBUS driver

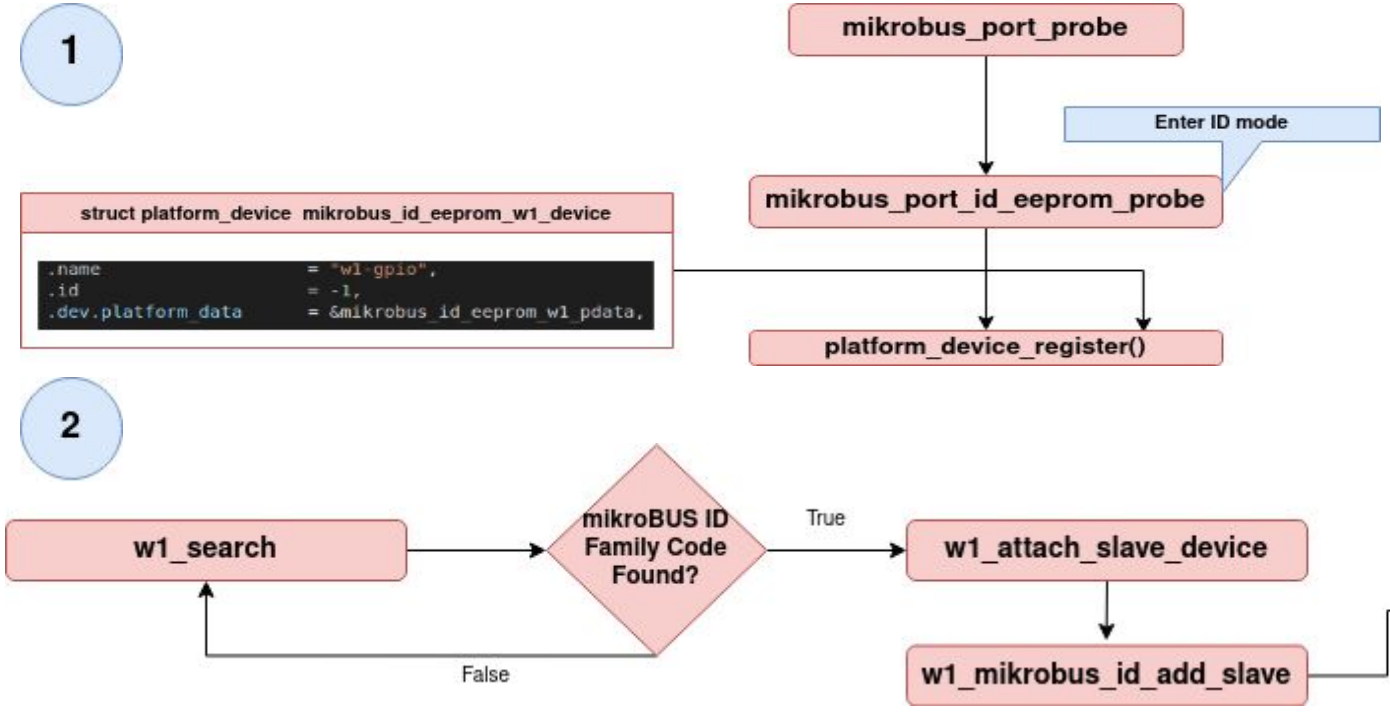


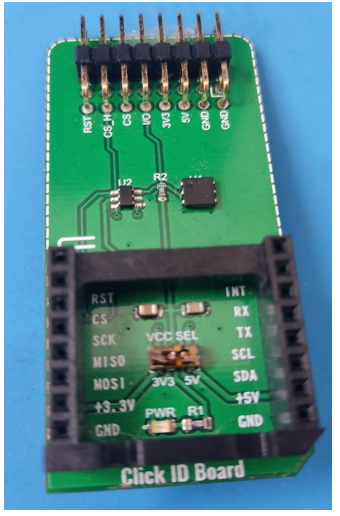
mikroBUS Identifier Mechanism

- Solution from MikroElektronika.
- uses a 1-Wire EEPROM to store add-on board identifier information.
- same socket standard is maintained and a mux is used for sharing the CS GPIO line between the EEPROM and device.
- At boot time host can enter ID mode, read EEPROM contents and probe corresponding device drivers from the manifest information.



mikroBUS Identifier Mechanism:
Sequence of Events in Kernel

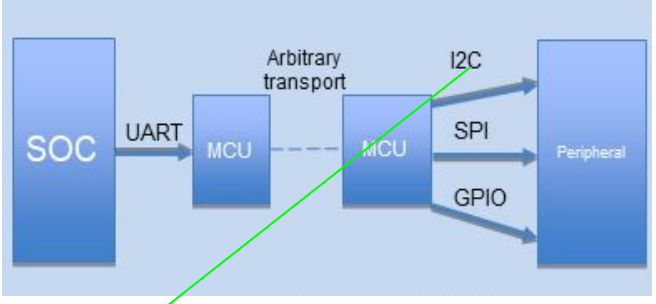




MikroElektronika Click ID Board Prototype

Overview of Greybus

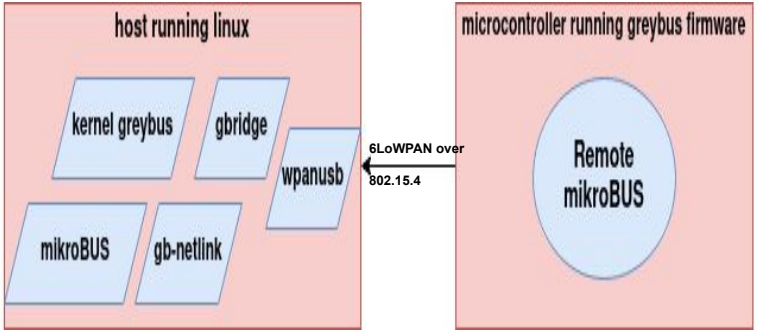
- initially designed for Google's Project Ara smartphone.
- RPC protocol to manage and control modules.
- greybus allows peripherals on a remote device to appear as if they were on the host through a transport.
- transport could be a wired or wireless network.
- greybus supports module discovery and hot-plug/unplug during run-time.
- for a wireless transport, connecting to the network can be considered equivalent to hot-plug event.
- allows to keep the intelligence in the host.
- remote device requires very generic firmware.
- In 2019, most of greybus core drivers were moved out of staging.



greybus allows peripherals on a remote device to appear as if they were on the host

mikroBUS ports over Greybus

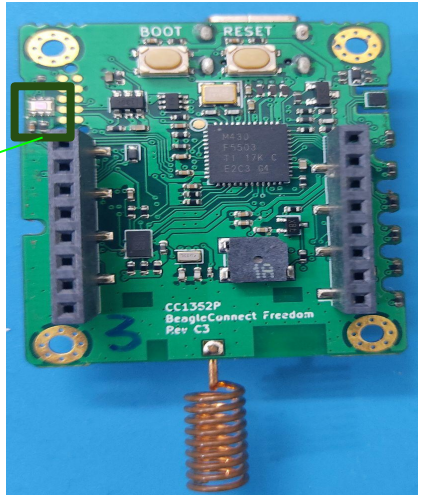
- mikroBUS from kernel perspective is just an association of SPI, I2C, UART, PWM and GPIOs.
- these peripherals need not always be directly on the SoC but can be on a remote microcontroller.
- greybus allows peripherals on a remote microcontroller to appear as if it was on the host over a suitable transport.
- mikroBUS driver can associate these greybus virtual peripherals and probe device drivers on the host for add-on boards connected on the remote mikroBUS port on the microcontroller.




```
vaishnav@ubuntu:~/$ ls /sys/bus/gbphy/devices/gbphy
gbphy1/
vaishnav@ubuntu:~/$ ls /sys/bus/gbphy/devices/gbphy1/
driver/      i2c-8/       mikrobus-0/  power/       protocol_id  subsystem/   uevent
vaishnav@ubuntu:~/$ ls /sys/bus/gbphy/devices/gbphy1/mikrobus-0/
delete_device  device/        name           new_device     power/         rescan         subsystem/     uevent
```

mikroBUS ports over Greybus:
User Experience

```
vaishnav@ubuntu:~/$ dmesg
greybus 1-2.2: GMP VID=0x00000126, PID=0x00000126
greybus 1-2.2: DDBL1 Manufacturer=0x00000126, Product=0x00000126
greybus 1-2.2: excess descriptors in interface manifest
mikrobus:mikrobus_port_gb_register: mikrobus gb_probe , num cports= 3, manifest_size 252
mikrobus:mikrobus_port_register: registering port mikrobus-0
mikrobus_manifest:mikrobus_manifest_attach_device: parsed device 1, driver=bme280, protocol=3, reg=76
mikrobus_manifest:mikrobus_manifest_attach_device: parsed device 2, driver=opt3001, protocol=3, reg=44
mikrobus_manifest:mikrobus_manifest_attach_device: parsed device 3, driver=ams-iaq-core, protocol=3, reg=5a
mikrobus_manifest:mikrobus_manifest_parse:  Greybus Service Sample Application manifest parsed with 3 devices
mikrobus mikrobus-0: registering device : bme280
mikrobus mikrobus-0: registering device : opt3001
mikrobus mikrobus-0: registering device : ams-iaq-core
vaishnav@ubuntu:/manifesto$ ls /sys/bus/iio/devices/iio:device1/
current_timestamp_clock events/ in_illuminance_integration_time name subsystem/
dev in_illuminance_input integration_time_available power/ uevent
vaishnav@ubuntu:~/$ cat /sys/bus/iio/devices/iio:device0/in_illuminance_input
170.560000
```



BeagleConnect™ Freedom HW Prototype

Once remote device is connected to network, after module discovery, device drivers are probed and thus sensors on remote appear as if it was on host and user can communicate with the sensor directly using sysfs, without even knowing exact details of the command set of the sensor.

LINUX PLUMBERS CONFERENCE September 20-24, 2021

mikroBUS ports over Greybus:
what changes from a physical mikroBUS and why Greybus manifest as identifier format?

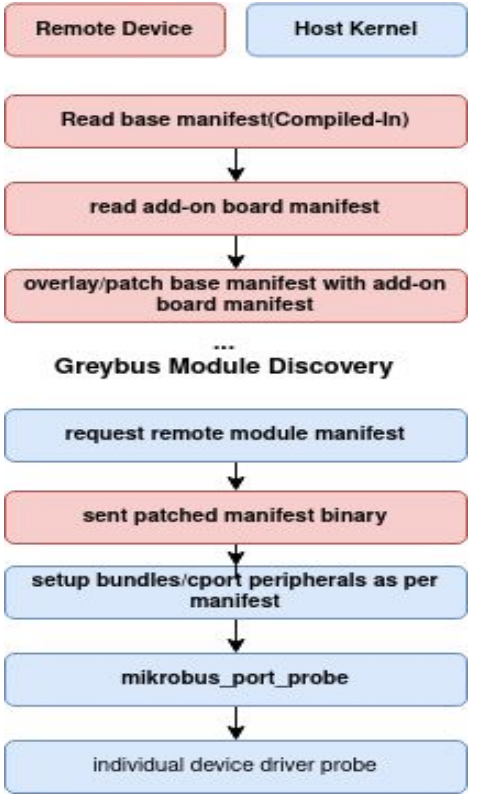
- on local mikroBUS direct 1-wire read can be performed for fetching the add-on board identifier manifest binary.
- greybus already has a module describing format(manifest)
- today greybus manifest allows to describe peripherals on a remote device but doesn't allow to describe devices connected to these peripherals
- a base greybus manifest(containing peripheral information) is overlayed with add-on board manifest(s) and the combined manifest is sent from Zephyr remote device to the host during greybus module discovery.
- base portion of manifest is used by greybus to map the peripherals and add-on board portion is used by mikroBUS driver to probe device drivers

Extending greybus manifest requires maintenance of only a single add-on board identifier information irrespective of how it is connected, locally or remotely over greybus

mikroBUS ports over Greybus:
sequence of events in remote, kernel

- mikroBUS is a variant of gbphy class; a subset of buses are supported and devices need to be instantiated on these virtual buses.
- greybus deletes manifest information once the bundle cport devices are created, for mikroBUS we need to retain this information until devices are instantiated.
- once the gbphy cport devices are created the gbphy device has valid SPI, I2C, GPIO .etc controllers according to the manifest and this along with the add-on board device information from manifest can be used to probe corresponding device drivers.

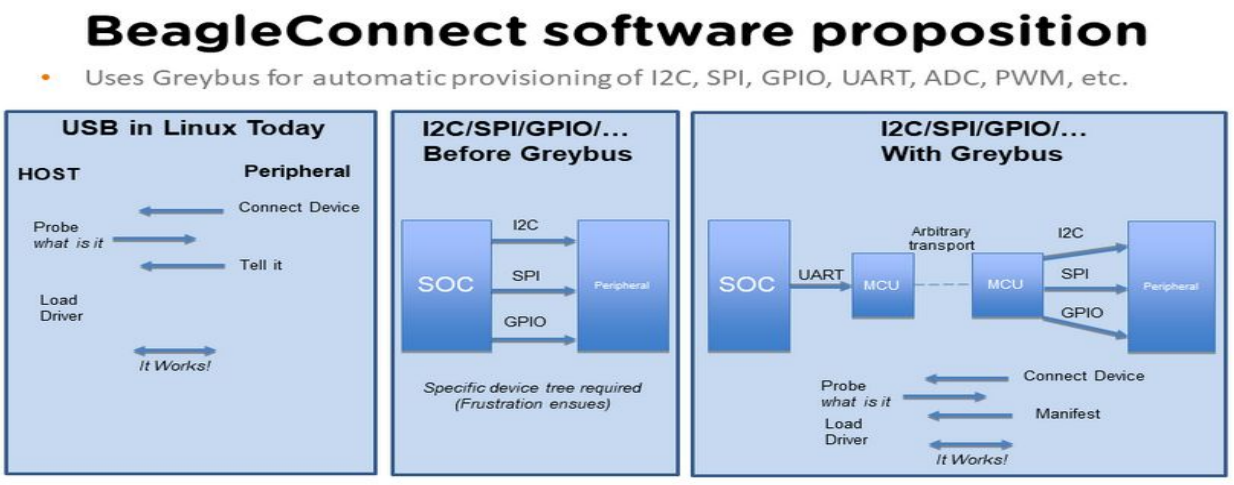
```
vaishnav@ubuntu:~/mygreybus$ ls /sys/bus/mikrobus/devices/mikrobus-0/
delete_device  device/        name           new_device     power/         rescan         subsystem/     uevent
vaishnav@ubuntu:~/mygreybus$ ls /sys/bus/gbphy/devices/gbphy
gbphy1/ gbphy2/
vaishnav@ubuntu:~/mygreybus$ ls /sys/bus/gbphy/devices/gbphy1/
driver/      i2c-8/       mikrobus-0/  power/       protocol_id  subsystem/   uevent
vaishnav@ubuntu:~/mygreybus$ ls /sys/bus/gbphy/devices/gbphy1/mikrobus-0/
delete_device  device/        name           new_device     power/         rescan         subsystem/     uevent
```

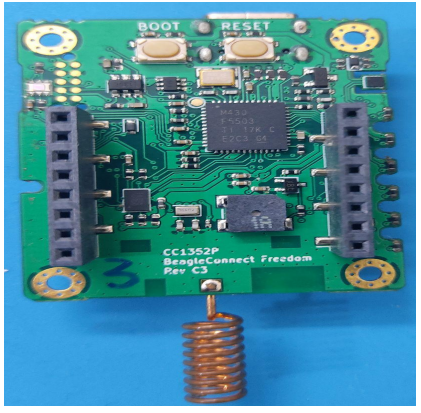


LINUX PLUMBERS CONFERENCE September 20-24, 2021

BeagleConnect™

BeagleConnect™ is a revolutionary technology virtually eliminating low-level software development for IoT and IIoT applications, such as building automation, factory automation, and home automation.BeagleConnect simply eliminates the need for low-level software development by shifting the burden into the most massive and collaborative software project of all time, the Linux Kernel.





BeagleConnect™ Freedom
HW Prototype

Contact Jason Kridner(BeagleBoard.org) to join BeagleConnect Freedom Beta Program : bbb.io/j

IoThree's Company MicroConference

beagleboard.org

Demo：BeagleConnect Freedom & PocketBeagle

Status of the work, TODO, Open Problems

- mikroBUS driver works with latest definition of add-on board identifier mechanism and also with mikroBUS over greybus
- Zephyr side greybus manifest changes and extension for mikroBUS
- draft patches without greybus changes sent for RFC
- 140 Add-on boards manifests available and tested
- latest changes needs to be cleaned up and patches to be sent
- Zephyr side needs updates to support add-on boards with UART, PWM and interrupts
- Pinmux handling while adding mikroBUS ports through greybus?

Additional Information

- BeagleConnect™ : https://github.com/jadonk/beagleconnect
- Using Linux, Zephyr and Greybus for IoT : https://www.zephyrproject.org/using-linux-zephyr-greybus-for-iot/
- Greybus: https://lwn.net/Articles/715955/
- List of supported add-on boards and their manifests: https://github.com/vaishnav98/manifesto/wiki/click_info
- mikroBUS driver patches: https://github.com/vaishnav98/bb-kernel/tree/clickid/patches/drivers/mikrobus
- BeagleConnect Freedom Beta Program: ***bbb.io/j***

LINUX PLUMBERS CONFERENCE September 20-24, 2021

Q & A

LINUX PLUMBERS CONFERENCE September 20-24, 2021

THANK YOU

IoThree's Company MicroConference

beagleboard.org