



Bluetooth device tree configuration



Contents

1. Bluetooth device tree configuration	3
2. Device tree	9
3. STM32CubeMX	15
4. Serial TTY device tree configuration	21
5. WLAN and Bluetooth hardware component	27



A quality version of this page, approved on *15 October 2019*, was based off this revision.

Contents

1 Article purpose	4
2 Bluetooth DT bindings documentation	5
3 Bluetooth DT configuration	6
3.1 Bluetooth DT configuration (STM32 level)	6
3.2 Bluetooth DT configuration (board level)	6
4 How to configure Bluetooth using CubeMX	8
5 References	9



1 Article purpose

This article explains how to configure *Bluetooth* ^[1] **when the peripheral (or peripheral associated to the framework) is assigned to the Linux[®]OS.**

The configuration is performed using the **device tree mechanism** ^[2].

The Bluetooth companion chip chosen on our platform is a Cypress chip ^[3]



2 Bluetooth DT bindings documentation

The *Bluetooth*^[4] tree bindings are composed of:

- STM32 USART device tree bindings ^[5]
- The Cypress device, used as child node ^[6] of the host USART device to which the slave device is attached.



3 Bluetooth DT configuration

This hardware description is a combination of the STM32 microprocessor device tree files (.dtsi extension) and board device tree files (.dts extension). See the device tree for an explanation of the device tree file split.

3.1 Bluetooth DT configuration (STM32 level)

The USART peripheral node is located in *stm32mp157c.dtsi*

- This is a set of properties that may not vary for given STM32 device, such as: registers address, clock, reset...

The USART DT configuration is explained in [Serial TTY device tree configuration](#)

3.2 Bluetooth DT configuration (board level)

For ecosystem release v1.1.0 :

For ecosystem release v1.1.0 :

[Description for the STM32MP15-Ecosystem-v1.1.0 revision]

```
&usart2 {
    ...
    st,hw-flow-ctrl;                /* enable hardware flow
control */
    ...
    bluetooth {                    /* node of Bluetooth
companion chip */
        shutdown-gpios = <&gpioz 6 GPIO_ACTIVE_HIGH>; /* GPIO specifier, used to
enable the BT module */
        compatible = "brcm,bcm43438-bt";
        max-speed = <3000000>;
    };
};
```

Specific properties for USART:

- st,hw-flow-ctrl: bool flag to enable hardware flow control

For ecosystem release v1.0.0 :

[Description for the STM32MP15-Ecosystem-v1.0.0 and previous revisions]

```
&usart2 {
    ...
    st,hw-flow-ctrl;                /* enable hardware flow control */
    ...
    bluetooth {                    /* node of Bluetooth companion chip */
        pinctrl-names = "default";
    };
};
```



```
pinctrl-0 = <&btreg>;                               /* GPIO to power up or down the
internal companion chip regulators */
compatible = "brcm,bcm43438-bt";
max-speed = <3000000>;
};
};
```

Specific properties for USART:

- btreg: GPIO to power up or down the internal CYW4343W regulators used by the Bluetooth section
- st,hw-flow-ctrl: bool flag to enable hardware flow control



4 How to configure Bluetooth using CubeMX

The `STM32CubeMX` tool can be used to configure the STM32MPU device and get the corresponding platform configuration device tree files.

The `STM32CubeMX` may not support all the properties described in the above `DT bindings` documentation paragraph. If so, the tool inserts **user sections** in the generated device tree. These sections can then be edited to add some properties and they are preserved from one generation to another. Refer to `STM32CubeMX` user manual for further information.



5 References

- Bluetooth
- Device tree
- MURATA CYW4343W datasheet
- WLAN_and_Bluetooth_hardware_component
- Serial TTY device tree configuration
- Documentation/devicetree/bindings/net/broadcom-bluetooth.txt

Linux® is a registered trademark of Linus Torvalds.

Operating System

Device Tree

Universal Synchronous/Asynchronous Receiver/Transmitter

General-Purpose Input/Output (A realization of open ended transmission between devices on an embedded level. These pins available on a processor can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.)

BlueTooth

Stable: 05.11.2021 - 11:08 / Revision: 05.11.2021 - 11:05

Contents

1 Article purpose	10
2 Bluetooth DT bindings documentation	11
3 Bluetooth DT configuration	12
3.1 Bluetooth DT configuration (STM32 level)	12
3.2 Bluetooth DT configuration (board level)	12
4 How to configure Bluetooth using CubeMX	14
5 References	15



1 Article purpose

This article explains how to configure *Bluetooth*^[1] **when the peripheral (or peripheral associated to the framework) is assigned to the Linux[®]OS.**

The configuration is performed using the **device tree mechanism**^[2].

The Bluetooth companion chip chosen on our platform is a Cypress chip^[3]



2 Bluetooth DT bindings documentation

The *Bluetooth*^[4] tree bindings are composed of:

- STM32 USART device tree bindings ^[5]
- The Cypress device, used as child node ^[6] of the host USART device to which the slave device is attached.



3 Bluetooth DT configuration

This hardware description is a combination of the STM32 microprocessor device tree files (.dtsi extension) and board device tree files (.dts extension). See the device tree for an explanation of the device tree file split.

3.1 Bluetooth DT configuration (STM32 level)

The USART peripheral node is located in *stm32mp157c.dtsi*

- This is a set of properties that may not vary for given STM32 device, such as: registers address, clock, reset...

The USART DT configuration is explained in [Serial TTY device tree configuration](#)

3.2 Bluetooth DT configuration (board level)

For ecosystem release v1.1.0 :

For ecosystem release v1.1.0 :

[Description for the STM32MP15-Ecosystem-v1.1.0 revision]

```
&usart2 {
    ...
    st,hw-flow-ctrl;                /* enable hardware flow
control */
    ...
    bluetooth {                    /* node of Bluetooth
companion chip */
        shutdown-gpios = <&gpioz 6 GPIO_ACTIVE_HIGH>; /* GPIO specifier, used to
enable the BT module */
        compatible = "brcm,bcm43438-bt";
        max-speed = <3000000>;
    };
};
```

Specific properties for USART:

- st,hw-flow-ctrl: bool flag to enable hardware flow control

For ecosystem release v1.0.0 :

[Description for the STM32MP15-Ecosystem-v1.0.0 and previous revisions]

```
&usart2 {
    ...
    st,hw-flow-ctrl;                /* enable hardware flow control */
    ...
    bluetooth {                    /* node of Bluetooth companion chip */
        pinctrl-names = "default";
    };
};
```



```
pinctrl-0 = <&btreg>;                               /* GPIO to power up or down the
internal companion chip regulators */
compatible = "brcm,bcm43438-bt";
max-speed = <3000000>;
};
};
```

Specific properties for USART:

- btreg: GPIO to power up or down the internal CYW4343W regulators used by the Bluetooth section
- st,hw-flow-ctrl: bool flag to enable hardware flow control



4 How to configure Bluetooth using CubeMX

The `STM32CubeMX` tool can be used to configure the STM32MPU device and get the corresponding platform configuration device tree files.

The `STM32CubeMX` may not support all the properties described in the above `DT bindings` documentation paragraph. If so, the tool inserts **user sections** in the generated device tree. These sections can then be edited to add some properties and they are preserved from one generation to another. Refer to `STM32CubeMX` user manual for further information.



5 References

- Bluetooth
- Device tree
- MURATA CYW4343W datasheet
- WLAN_and_Bluetooth_hardware_component
- Serial TTY device tree configuration
- Documentation/devicetree/bindings/net/broadcom-bluetooth.txt

Linux® is a registered trademark of Linus Torvalds.

Operating System

Device Tree

Universal Synchronous/Asynchronous Receiver/Transmitter

General-Purpose Input/Output (A realization of open ended transmission between devices on an embedded level. These pins available on a processor can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.)

BlueTooth

Stable: 23.09.2020 - 13:22 / Revision: 12.06.2020 - 13:25

Contents

1 Article purpose	16
2 Bluetooth DT bindings documentation	17
3 Bluetooth DT configuration	18
3.1 Bluetooth DT configuration (STM32 level)	18
3.2 Bluetooth DT configuration (board level)	18
4 How to configure Bluetooth using CubeMX	20
5 References	21



1 Article purpose

This article explains how to configure *Bluetooth*^[1] **when the peripheral (or peripheral associated to the framework) is assigned to the Linux[®]OS.**

The configuration is performed using the **device tree mechanism**^[2].

The Bluetooth companion chip chosen on our platform is a Cypress chip^[3]



2 Bluetooth DT bindings documentation

The *Bluetooth*^[4] tree bindings are composed of:

- STM32 USART device tree bindings ^[5]
- The Cypress device, used as child node ^[6] of the host USART device to which the slave device is attached.



3 Bluetooth DT configuration

This hardware description is a combination of the STM32 microprocessor device tree files (.dtsi extension) and board device tree files (.dts extension). See the device tree for an explanation of the device tree file split.

3.1 Bluetooth DT configuration (STM32 level)

The USART peripheral node is located in *stm32mp157c.dtsi*

- This is a set of properties that may not vary for given STM32 device, such as: registers address, clock, reset...

The USART DT configuration is explained in [Serial TTY device tree configuration](#)

3.2 Bluetooth DT configuration (board level)

For ecosystem release v1.1.0 :

For ecosystem release v1.1.0 :

[Description for the STM32MP15-Ecosystem-v1.1.0 revision]

```
&usart2 {
    ...
    st,hw-flow-ctrl;                /* enable hardware flow
control */
    ...
    bluetooth {                    /* node of Bluetooth
companion chip */
        shutdown-gpios = <&gpioz 6 GPIO_ACTIVE_HIGH>; /* GPIO specifier, used to
enable the BT module */
        compatible = "brcm,bcm43438-bt";
        max-speed = <3000000>;
    };
};
```

Specific properties for USART:

- st,hw-flow-ctrl: bool flag to enable hardware flow control

For ecosystem release v1.0.0 :

[Description for the STM32MP15-Ecosystem-v1.0.0 and previous revisions]

```
&usart2 {
    ...
    st,hw-flow-ctrl;                /* enable hardware flow control */
    ...
    bluetooth {                    /* node of Bluetooth companion chip */
        pinctrl-names = "default";
    };
};
```



```
pinctrl-0 = <&btreg>;                               /* GPIO to power up or down the
internal companion chip regulators */
compatible = "brcm,bcm43438-bt";
max-speed = <3000000>;
};
};
```

Specific properties for USART:

- btreg: GPIO to power up or down the internal CYW4343W regulators used by the Bluetooth section
- st,hw-flow-ctrl: bool flag to enable hardware flow control



4 How to configure Bluetooth using CubeMX

The `STM32CubeMX` tool can be used to configure the STM32MPU device and get the corresponding platform configuration device tree files.

The `STM32CubeMX` may not support all the properties described in the above `DT bindings` documentation paragraph. If so, the tool inserts **user sections** in the generated device tree. These sections can then be edited to add some properties and they are preserved from one generation to another. Refer to `STM32CubeMX` user manual for further information.



5 References

- Bluetooth
- Device tree
- MURATA CYW4343W datasheet
- WLAN_and_Bluetooth_hardware_component
- Serial TTY device tree configuration
- Documentation/devicetree/bindings/net/broadcom-bluetooth.txt

Linux[®] is a registered trademark of Linus Torvalds.

Operating System

Device Tree

Universal Synchronous/Asynchronous Receiver/Transmitter

General-Purpose Input/Output (A realization of open ended transmission between devices on an embedded level. These pins available on a processor can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.)

BlueTooth

Stable: 26.03.2021 - 15:50 / Revision: 26.03.2021 - 15:49

Contents

1 Article purpose	22
2 Bluetooth DT bindings documentation	23
3 Bluetooth DT configuration	24
3.1 Bluetooth DT configuration (STM32 level)	24
3.2 Bluetooth DT configuration (board level)	24
4 How to configure Bluetooth using CubeMX	26
5 References	27



1 Article purpose

This article explains how to configure *Bluetooth* ^[1] **when the peripheral (or peripheral associated to the framework) is assigned to the Linux[®]OS.**

The configuration is performed using the **device tree mechanism** ^[2].

The Bluetooth companion chip chosen on our platform is a Cypress chip ^[3]



2 Bluetooth DT bindings documentation

The *Bluetooth*^[4] tree bindings are composed of:

- STM32 USART device tree bindings ^[5]
- The Cypress device, used as child node ^[6] of the host USART device to which the slave device is attached.



3 Bluetooth DT configuration

This hardware description is a combination of the STM32 microprocessor device tree files (.dtsi extension) and board device tree files (.dts extension). See the device tree for an explanation of the device tree file split.

3.1 Bluetooth DT configuration (STM32 level)

The USART peripheral node is located in *stm32mp157c.dtsi*

- This is a set of properties that may not vary for given STM32 device, such as: registers address, clock, reset...

The USART DT configuration is explained in [Serial TTY device tree configuration](#)

3.2 Bluetooth DT configuration (board level)

For ecosystem release v1.1.0 :

For ecosystem release v1.1.0 :

[Description for the STM32MP15-Ecosystem-v1.1.0 revision]

```
&usart2 {
    ...
    st,hw-flow-ctrl;                /* enable hardware flow
control */
    ...
    bluetooth {                    /* node of Bluetooth
companion chip */
        shutdown-gpios = <&gpioz 6 GPIO_ACTIVE_HIGH>; /* GPIO specifier, used to
enable the BT module */
        compatible = "brcm,bcm43438-bt";
        max-speed = <3000000>;
    };
};
```

Specific properties for USART:

- st,hw-flow-ctrl: bool flag to enable hardware flow control

For ecosystem release v1.0.0 :

[Description for the STM32MP15-Ecosystem-v1.0.0 and previous revisions]

```
&usart2 {
    ...
    st,hw-flow-ctrl;                /* enable hardware flow control */
    ...
    bluetooth {                    /* node of Bluetooth companion chip */
        pinctrl-names = "default";
    };
};
```




```
pinctrl-0 = <&btreg>;                               /* GPIO to power up or down the
internal companion chip regulators */
compatible = "brcm,bcm43438-bt";
max-speed = <3000000>;
};
};
```

Specific properties for USART:

- btreg: GPIO to power up or down the internal CYW4343W regulators used by the Bluetooth section
- st,hw-flow-ctrl: bool flag to enable hardware flow control



4 How to configure Bluetooth using CubeMX

The `STM32CubeMX` tool can be used to configure the STM32MPU device and get the corresponding platform configuration device tree files.

The `STM32CubeMX` may not support all the properties described in the above `DT bindings` documentation paragraph. If so, the tool inserts **user sections** in the generated device tree. These sections can then be edited to add some properties and they are preserved from one generation to another. Refer to `STM32CubeMX` user manual for further information.



5 References

- Bluetooth
- Device tree
- MURATA CYW4343W datasheet
- WLAN_and_Bluetooth_hardware_component
- Serial TTY device tree configuration
- Documentation/devicetree/bindings/net/broadcom-bluetooth.txt

Linux® is a registered trademark of Linus Torvalds.

Operating System

Device Tree

Universal Synchronous/Asynchronous Receiver/Transmitter

General-Purpose Input/Output (A realization of open ended transmission between devices on an embedded level. These pins available on a processor can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.)

BlueTooth

Stable: 03.02.2020 - 08:42 / Revision: 03.02.2020 - 08:30

Contents

1 Article purpose	28
2 Bluetooth DT bindings documentation	29
3 Bluetooth DT configuration	30
3.1 Bluetooth DT configuration (STM32 level)	30
3.2 Bluetooth DT configuration (board level)	30
4 How to configure Bluetooth using CubeMX	32
5 References	33



1 Article purpose

This article explains how to configure *Bluetooth*^[1] **when the peripheral (or peripheral associated to the framework) is assigned to the Linux[®]OS.**

The configuration is performed using the **device tree mechanism**^[2].

The Bluetooth companion chip chosen on our platform is a Cypress chip^[3]



2 Bluetooth DT bindings documentation

The *Bluetooth*^[4] tree bindings are composed of:

- STM32 USART device tree bindings ^[5]
- The Cypress device, used as child node ^[6] of the host USART device to which the slave device is attached.



3 Bluetooth DT configuration

This hardware description is a combination of the STM32 microprocessor device tree files (.dtsi extension) and board device tree files (.dts extension). See the device tree for an explanation of the device tree file split.

3.1 Bluetooth DT configuration (STM32 level)

The USART peripheral node is located in *stm32mp157c.dtsi*

- This is a set of properties that may not vary for given STM32 device, such as: registers address, clock, reset...

The USART DT configuration is explained in [Serial TTY device tree configuration](#)

3.2 Bluetooth DT configuration (board level)

For ecosystem release v1.1.0 :

For ecosystem release v1.1.0 :

[Description for the STM32MP15-Ecosystem-v1.1.0 revision]

```
&usart2 {
    ...
    st,hw-flow-ctrl;                /* enable hardware flow
control */
    ...
    bluetooth {                    /* node of Bluetooth
companion chip */
        shutdown-gpios = <&gpioz 6 GPIO_ACTIVE_HIGH>; /* GPIO specifier, used to
enable the BT module */
        compatible = "brcm,bcm43438-bt";
        max-speed = <3000000>;
    };
};
```

Specific properties for USART:

- st,hw-flow-ctrl: bool flag to enable hardware flow control

For ecosystem release v1.0.0 :

[Description for the STM32MP15-Ecosystem-v1.0.0 and previous revisions]

```
&usart2 {
    ...
    st,hw-flow-ctrl;                /* enable hardware flow control */
    ...
    bluetooth {                    /* node of Bluetooth companion chip */
        pinctrl-names = "default";
    };
};
```



```
pinctrl-0 = <&btreg>;                               /* GPIO to power up or down the
internal companion chip regulators */
compatible = "brcm,bcm43438-bt";
max-speed = <3000000>;
};
};
```

Specific properties for USART:

- btreg: GPIO to power up or down the internal CYW4343W regulators used by the Bluetooth section
- st,hw-flow-ctrl: bool flag to enable hardware flow control



4 How to configure Bluetooth using CubeMX

The `STM32CubeMX` tool can be used to configure the `STM32MPU` device and get the corresponding platform configuration device tree files.

The `STM32CubeMX` may not support all the properties described in the above `DT bindings` documentation paragraph. If so, the tool inserts **user sections** in the generated device tree. These sections can then be edited to add some properties and they are preserved from one generation to another. Refer to `STM32CubeMX` user manual for further information.



5 References

- Bluetooth
- Device tree
- MURATA CYW4343W datasheet
- WLAN_and_Bluetooth_hardware_component
- Serial TTY device tree configuration
- Documentation/devicetree/bindings/net/broadcom-bluetooth.txt

Linux® is a registered trademark of Linus Torvalds.

Operating System

Device Tree

Universal Synchronous/Asynchronous Receiver/Transmitter

General-Purpose Input/Output (A realization of open ended transmission between devices on an embedded level. These pins available on a processor can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.)

BlueTooth