



## Bluetooth device tree configuration



---

## Contents

---

1. Bluetooth device tree configuration .....	3
2. Device tree .....	8
3. STM32CubeMX .....	13
4. Serial TTY device tree configuration .....	18
5. WLAN and Bluetooth hardware component .....	23



CLASS: PROTECTED | OBJECT REVISION: PROTECTED | DATE:

A quality version of this page, approved on *11 June 2020*, 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 .....	7
5 References .....	8



---

## 1 Article purpose

---

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

The configuration is performed using the **device tree mechanism**<sup>[2]</sup>.

The Bluetooth companion chip chosen on our platform is a Cypress chip<sup>[3]</sup>



---

## 2 Bluetooth DT bindings documentation

---

The *Bluetooth*<sup>[4]</sup> tree bindings are composed of:

- STM32 USART device tree bindings <sup>[5]</sup>
- The Cypress device, used as child node <sup>[6]</sup> 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 *stm32mp151.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)

```
&usart2 {
    ...
    uart-has-rtscts;                               /* 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 = <30000000>;
    };
};
```

Specific properties for USART:

- `uart-has-rtscts`; 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: **Not stable** / Revision: 29.09.2021 - 15:37

### Contents

1 Article purpose .....	9
2 Bluetooth DT bindings documentation .....	10
3 Bluetooth DT configuration .....	11
3.1 Bluetooth DT configuration (STM32 level) .....	11
3.2 Bluetooth DT configuration (board level) .....	11
4 How to configure Bluetooth using CubeMX .....	12
5 References .....	13





---

## 1 Article purpose

---

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

The configuration is performed using the **device tree mechanism**<sup>[2]</sup>.

The Bluetooth companion chip chosen on our platform is a Cypress chip<sup>[3]</sup>



---

## 2 Bluetooth DT bindings documentation

---

The *Bluetooth*<sup>[4]</sup> tree bindings are composed of:

- STM32 USART device tree bindings <sup>[5]</sup>
- The Cypress device, used as child node <sup>[6]</sup> 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 *stm32mp151.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)

```
&usart2 {
    ...
    uart-has-rtscts;                               /* 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 = <30000000>;
    };
};
```

Specific properties for USART:

- `uart-has-rtscts`; 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 .....	14
2 Bluetooth DT bindings documentation .....	15
3 Bluetooth DT configuration .....	16
3.1 Bluetooth DT configuration (STM32 level) .....	16
3.2 Bluetooth DT configuration (board level) .....	16
4 How to configure Bluetooth using CubeMX .....	17
5 References .....	18



---

## 1 Article purpose

---

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

The configuration is performed using the **device tree mechanism** <sup>[2]</sup>.

The Bluetooth companion chip chosen on our platform is a Cypress chip <sup>[3]</sup>



---

## 2 Bluetooth DT bindings documentation

---

The *Bluetooth*<sup>[4]</sup> tree bindings are composed of:

- STM32 USART device tree bindings <sup>[5]</sup>
- The Cypress device, used as child node <sup>[6]</sup> 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 *stm32mp151.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)

```
&usart2 {
    ...
    uart-has-rtscts; /* 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 = <30000000>;
    };
};
```

Specific properties for USART:

- `uart-has-rtscts`; 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 .....	19
2 Bluetooth DT bindings documentation .....	20
3 Bluetooth DT configuration .....	21
3.1 Bluetooth DT configuration (STM32 level) .....	21
3.2 Bluetooth DT configuration (board level) .....	21
4 How to configure Bluetooth using CubeMX .....	22
5 References .....	23



---

## 1 Article purpose

---

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

The configuration is performed using the **device tree mechanism**<sup>[2]</sup>.

The Bluetooth companion chip chosen on our platform is a Cypress chip<sup>[3]</sup>



---

## 2 Bluetooth DT bindings documentation

---

The *Bluetooth*<sup>[4]</sup> tree bindings are composed of:

- STM32 USART device tree bindings <sup>[5]</sup>
- The Cypress device, used as child node <sup>[6]</sup> 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 *stm32mp151.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)

```
&usart2 {
    ...
    uart-has-rtscts;                               /* 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 = <30000000>;
    };
};
```

Specific properties for USART:

- `uart-has-rtscts`; 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<sup>®</sup> 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 .....	24
2 Bluetooth DT bindings documentation .....	25
3 Bluetooth DT configuration .....	26
3.1 Bluetooth DT configuration (STM32 level) .....	26
3.2 Bluetooth DT configuration (board level) .....	26
4 How to configure Bluetooth using CubeMX .....	27
5 References .....	28



---

## 1 Article purpose

---

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

The configuration is performed using the **device tree mechanism** <sup>[2]</sup>.

The Bluetooth companion chip chosen on our platform is a Cypress chip <sup>[3]</sup>





---

## 2 Bluetooth DT bindings documentation

---

The *Bluetooth*<sup>[4]</sup> tree bindings are composed of:

- STM32 USART device tree bindings <sup>[5]</sup>
- The Cypress device, used as child node <sup>[6]</sup> 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 *stm32mp151.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)

```
&usart2 {
    ...
    uart-has-rtscts;                               /* 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:

- `uart-has-rtscts`; 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<sup>®</sup> 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