Silicon Labs CP210x USB to UART Bridge (COM6)
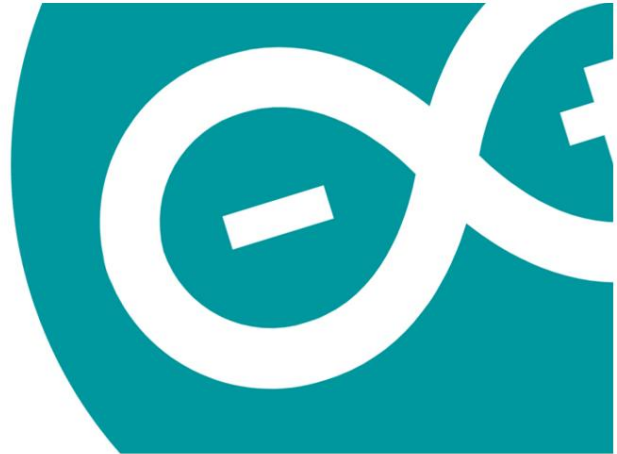
The brackets behind the device name is the port number you need, just select it in the Arduinio IDE.


In the toolbar under the menu bar of the Arduino IDE, the two buttons on the left are "compile only" and "compile+upload" **buttons** from left to right . **Usually we only need to click the second button, and the Arduino** IDE will automatically upload (program) the compiled binary file to the TT expansion module **after the compilation is completed .**

Usually, Arduino IDE takes a long time (three to five minutes, depending on computer performance) to compile the contents of the RMTT core library when compiling a new project, and then the compilation process will call the compilation cache for secondary compilation. If you think compilation is too slow, there is a solution mentioned in the next section.
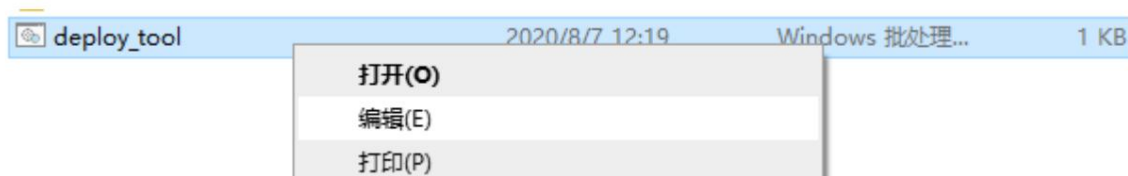
External compilation and upload mode without Arduino IDE (external deployment)

Inside the TT package you can see the "External Deployment Tools" folder, go into that folder and you will see

 That is, the Ministry
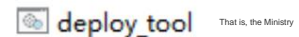
signing tool.

first use

You need to select "Edit" in the right-click menu, and change the environment path ("<arduino_loc>" **and** "<lib_loc>") in the first line of the deploy_tool.bat file to your Arduino program path and the installation of the Hardware library respectively **path (mentioned in the corresponding section above).**

```
@echo off call

tool_core\build.bat <arduino_loc> <lib_loc> %1 call tool_core\flash_firmonly.bat %2 %3
```

After the changes are completed, the contents of the entire deploy_tool.bat file should be similar to the following:

```
@echo off call

tool_core\build.bat D:\arduino-1.8.12 C:\Users\username\Documents\Arduino %1 call tool_core\flash_firmonly.bat %2 %3
```

How to use after the environment path is configured

Press SHIFT + right key in the blank area of the "External Deployment Tools" folder, and select "Open Powershell window here" or "Open command prompt (CMD) window here".

```
.\deploy_tool.bat <your_ino_file_location> <your_board_COM_NO>
<your_project_name>
```

Enter the above command in the Powershell (or CMD) window that appears.

Please remember to replace the three brackets ("<>") with the corresponding content before entering the command, which are "the address of the ino file that needs to be compiled", "the COM slogan that needs to be uploaded", "the name of your ino file "

For example:

```
.\deploy_tool.bat D:\MyProj\first.ino COM6 first
```

In this way, the external deployment tool can automatically use the previously compiled and cached core library content to speed up the compilation process.

Get to know the Tello Talent extension module

This module is an expansion module of Tello Talent. It is fixed with TT through the building block interface, and the module is powered and data exchanged through the onboard Micro USB interface.

This module is equipped with full-color LED lights, Tof sensors and 8*8 red and blue LED dot matrix screens. Through our example, you will learn how to program and control this expansion module in Arduino, and use it to create your own projects.

Top full color LED

Module composition

This module consists of three high-brightness LEDs, red, green and blue, and an external diffuser. We can control the color of the external perception by controlling the brightness of LEDs of different colors and supplementing it with an external diffuser.

module interface

```
class
    RMTT_RGB{ public:
        static void Init(); static
        void SetRed(uint32_t val, uint32_t valueMax = 255); static void SetBlue(uint32_t
        val, uint32_t valueMax = 255); static void SetGreen(uint32_t val, uint32_t
        valueMax = 255); static void SetRGB(uint32_t R, uint32_t G, uint32_t B, uint32_t
        valueMax
= 255); };
```

Common function name	pass parameters	parameter range	Parameter interpretation	function
Heat	/	/	/	**Necessary** - Initialize LEDs
SetRed	val	0-255	val LED brightness of this channel	Set the brightness of the red LED bead
SetBlue	val	0-255	val LED brightness of this channel	Set the brightness of the blue LED bead
SetGreen	val	0-255	val LED brightness of this channel	Set the brightness of the green lamp bead
SetRGB	R, G, B	Both are 0-255	R Red LED brightness G Green LED brightness B Blue LED brightness	Set the brightness of three channel lamp beads at the same time

Control Example - Breathing Light

The example of this module is to use RMTT_RGB to control the full-color LED light to make it appear breathing light effect.

```cpp
#include <RMTT_Libs.h>
#include <Wire.h>

#define BREATH_Hz (0.5)

#define BREATH_RED (0)
#define BREATH_BLUE (255) #define
BREATH_GREEN (255)

RMTT_RGB tt_rgb; // instantiate RMTT_RGB object

bool breath_toggle = false; // Brightness change direction
uint8_t breath_rate = 0; int period =         // Percentage of brightness brightness percentage
10;

void setup()
        { Wire.begin(27, 26); // Initialize the I2C bus
        Wire.setClock(400000);

        period = 1000 / BREATH_Hz; // Calculate the duration of a single breathing
        cycle in ms
        tt_rgb.Init(); tt_rgb.SetRGB(0, 0, 0);
}

void loop() {

    /* Brightness increase or decrease, pay attention to the direction
    of breath_toggle to avoid overflow */ breath_rate = breath_toggle ? breath_rate - 1 : breath_rate + 1;


    if ((breath_rate == 100) || (breath_rate == 0))
            breath_toggle = !breath_toggle; // Change the brightness increase or decrease direction

    tt_rgb.SetRGB(BREATH_RED * breath_rate / 100.0,
                        BREATH_GREEN   * breath_rate / 100.0, *
                        BREATH_BLUE    breath_rate / 100.0);


    delay(period / 100); // delay one hundredth of a breathing cycle
}
```

Red and blue 8*8 LED matrix

Module composition

This module consists of 64 red lamp beads and 64 blue lamp beads, a total of 128 lamp beads. Through the packaged interface, you can control the color and brightness of each light bead in the LED matrix.

module interface

```cpp
class RMTT_Matrix{ public:

        static void On(); static
        void Off();
```