# FashionStar串行总线舵机库-使用说明 (Ubuntu)

#### FashionStar串行总线舵机库-使用说明(Ubuntu)

物理接线 安装依赖

**CMake** 

CSerialPort

安装FashionStar串口总线舵机库

如何运行示例代码

作者: 深圳市华馨京科技有限公司

网站: https://fashionrobo.com/

更新时间: 2021/06/12

### 物理接线

1. 安装USB转TTL模块的驱动程序。

2. 将串口舵机转接板跟USB转TTL模块通过杜邦线相连。

串口舵机转接板	USB转TTL模块
RX (串口舵机转接板RX接收端)	Tx (USB转TTL模块的接收端)
TX (串口舵机转接板TX发送端)	Rx (USB转TTL模块的发送端)
GND	GND

## 安装依赖

#### **CMake**

sudo apt install cmake

### **CSerialPort**

CSerialPort 是基于C++的轻量级开源跨平台串口类库,可以轻松实现跨平台多操作系统的串口读写。

FashionStar串口总线舵机,通信部分使用的是CSerialPort库。使用前需要编译安装。

#### CSerialPort - Github代码仓库地址

详情可以看这个库的WIKI,因为原仓库文件有些冗余,因此我做了删减版本的,放置到 fashionstaruart-servo-cpp/dependency/CSerialPort 文件夹下。

在 CSerial Port 文件夹下创建build文件夹,依次执行指令。

```
mkdir build
cd build
cmake ..
make
sudo make install
```

注: 如果build为非空文件夹,每次cmake之前,需要清空build文件夹下的所有缓冲文件。

#### 安装日志

从日志文件里面可以看出,库文件最终被安装在了/usr/local 目录下。

```
kyle@turing:~/Project/fashionstar-uart-servo-cpp/dependency/CSerialPort/build$
sudo make install
[100%] Built target libcserialport
Install the project...
-- Install configuration: "Release"
-- Up-to-date: /usr/local/include/CSerialPort
-- Up-to-date: /usr/local/include/CSerialPort/SerialPortInfoUnixBase.h
-- Up-to-date: /usr/local/include/CSerialPort/SerialPort_global.h
-- Up-to-date: /usr/local/include/CSerialPort/SerialPortUnixBase.h
-- Up-to-date: /usr/local/include/CSerialPort/SerialPortBase.h
-- Up-to-date: /usr/local/include/CSerialPort/SerialPort.h
-- Up-to-date: /usr/local/include/CSerialPort/osplatformutil.h
-- Up-to-date: /usr/local/include/CSerialPort/SerialPortInfo.h
-- Up-to-date: /usr/local/include/CSerialPort/SerialPortInfoBase.h
-- Up-to-date: /usr/local/include/CSerialPort/sigslot.h
-- Up-to-date: /usr/local/include/CSerialPort/SerialPortInfoWinBase.h
-- Up-to-date: /usr/local/include/CSerialPort/SerialPortWinBase.h
-- Up-to-date: /usr/local/lib/libcserialport.so
-- Installing: /usr/local/lib/cmake/CSerialPort/cserialport-config.cmake
```

## 安装FashionStar串口总线舵机库

回退到 fashionstar-uart-servo-cpp 目录,创建 build 文件夹。

```
mkdir build
cd build
cmake ..
make
sudo make install
```

注:如果build为非空文件夹,每次cmake之前,需要清空build文件夹下的所有缓冲文件。

#### 安装日志

```
kyle@turing:~/Project/fashionstar-uart-servo-cpp/build$ make
Scanning dependencies of target fsuartservo

[ 33%] Building CXX object
src/CMakeFiles/fsuartservo.dir/FashionStar_UartServoProtocol.cpp.o

[ 66%] Building CXX object
src/CMakeFiles/fsuartservo.dir/FashionStar_UartServo.cpp.o

[100%] Linking CXX shared library ../lib/libfsuartservo.so

[100%] Built target fsuartservo
```

```
kyle@turing:~/Project/fashionstar-uart-servo-cpp/build$ sudo make install
[sudo] password for kyle:
[100%] Built target fsuartservo
Install the project...
-- Install configuration: ""
-- Installing: /usr/local/lib/libfsuartservo.so
-- Set runtime path of "/usr/local/lib/libfsuartservo.so" to ""
-- Installing:
/usr/local/include/FashionStar/UServo/FashionStar_UartServoProtocol.h
-- Installing: /usr/local/include/FashionStar/UServo/FashionStar_UartServo.h
```

### 如何运行示例代码

在 example 文件夹下, 存放了若干独立的示例工程, 每个例程都是独立的。

举例来讲, 舵机通信检测例程 servo\_ping, 进入到工程目录下。

修改源码里面的端口号名称。

```
mkdir build
cd build
cmake ..
make
```

执行编译出来的可执行文件:

```
./servo_ping
```