FashionStar串口总线舵机库-使用说明 (Windows)

作者:

：https://fashionrobo.com/

更新时间: 2021/06/12

C++开发环境配置

在Windows下配置C++的开发环境过程有些繁琐， 同时也有很多需要配置的地方， 以及技术常识。

配置过程详情见教程： cmake   C++

- Windows下配置C++的开发环境，安装mingw-w64, make, cmake
- 编译原理基础，动态链接库的作用原理。（在底部附录部分）
- 如何自己生成一个动态链接库，以及动态链接库的使用方法。

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

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 `fashionstar-uart-servo-cpp/dependency/CSerialPort` 文件夹下。

在 `CSerialPort` 文件夹下创建build文件夹，依次执行指令。

```
mkdir build
cd build
# 需要把`CMAKE_INSTALL_PREFIX` 里面的内容替换为你自己的C++库的安装路径.
cmake .. -D CMAKE_INSTALL_PREFIX="D:/KyleSoftware/CPP_LIBRARIES/"
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

```
[100%] Linking CXX shared library libcserialport.dll
[100%] Built target libcserialport
PS D:\KyleLab\FashionStar\fashionstar-uart-servo-cpp\dependency\CSerialPort\build> make install
Consolidate compiler generated dependencies of target libcserialport
[100%] Built target libcserialport
Install the project...
-- Install configuration: "Release"
-- Up-to-date: D:/KyleSoftware/CPP_LIBRARIES/include/CSerialPort
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/include/CSerialPort/osplatformutil.h
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/include/CSerialPort/SerialPort.h
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/include/CSerialPort/SerialPortBase.h
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/include/CSerialPort/SerialPortInfo.h
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/include/CSerialPort/SerialPortInfoBase.h
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/include/CSerialPort/SerialPortInfoUnixBase.h
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/include/CSerialPort/SerialPortInfoWinBase.h
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/include/CSerialPort/SerialPortUnixBase.h
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/include/CSerialPort/SerialPortWinBase.h
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/include/CSerialPort/SerialPort_global.h
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/include/CSerialPort/sigslot.h
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/lib/liblibcserialport.dll.a
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/bin/libcserialport.dll
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/lib/cmake/CSerialPort/cserialport-config.cmake
PS D:\KyleLab\FashionStar\fashionstar-uart-servo-cpp\dependency\CSerialPort\build>
```

安装FashionStar串口总线舵机库

修改CMAKE配置文件

`CMakeLists.txt`

```cmake
cmake_minimum_required(VERSION 3.00)
# 设置工程名称，设定版本
project(UART_DEMO VERSION 1.0)

# 动态链接库的安装路径
if(CMAKE_HOST_WIN32)
    # Windows
    # 注意事项：这里需要改成自己的Windows下的动态链接库的安装路径
    # <<<<<<<<<<<<<<<<<<<<<<<<<<<<< 改这里
    SET(CMAKE_INSTALL_PREFIX "D:/KyleSoftware/CPP_LIBRARIES/")
elseif(CMAKE_HOST_UNIX)
    # Linux
    # 注意事项：Linux下，CMake默认会将库安装到/usr/local，需要注意的是，安装的时候需要sudo权限
    # sudo make install
    SET(CMAKE_INSTALL_PREFIX "/usr/local/")
endif()

# 添加依赖的.h文件路径(即hello.h所在的文件夹)
include_directories(${CMAKE_INSTALL_PREFIX}/include  include)
# 添加link文件夹
link_directories(${CMAKE_INSTALL_PREFIX}/bin)

# 构建源码文件
add_subdirectory(src)
```

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

```
Windows PowerShell                                                    —  □  ×
PS D:\KyleLab\FashionStar\fashionstar-uart-servo-cpp\build> cmake ..
-- Building for: MinGW Makefiles
-- The C compiler identification is GNU 8.1.0
-- The CXX compiler identification is GNU 8.1.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: D:/KyleSoftware/mingw64/bin/gcc.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: D:/KyleSoftware/mingw64/bin/g++.exe - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: D:/KyleLab/FashionStar/fashionstar-uart-servo-cpp/build
PS D:\KyleLab\FashionStar\fashionstar-uart-servo-cpp\build>
```

```
Windows PowerShell                                                    —  □  ×
-- Generating done
-- Build files have been written to: D:/KyleLab/FashionStar/fashionstar-uart-servo-cpp/build
PS D:\KyleLab\FashionStar\fashionstar-uart-servo-cpp\build> make
[ 33%] Building CXX object src/CMakeFiles/fsuartservo.dir/FashionStar_UartServoProtocol.cpp.obj
[ 66%] Building CXX object src/CMakeFiles/fsuartservo.dir/FashionStar_UartServo.cpp.obj
[100%] Linking CXX shared library ..\lib\libfsuartservo.dll
[100%] Built target fsuartservo
PS D:\KyleLab\FashionStar\fashionstar-uart-servo-cpp\build>
```

```
Windows PowerShell                                                    —  □  ×
[100%] Built target fsuartservo
PS D:\KyleLab\FashionStar\fashionstar-uart-servo-cpp\build> make install
Consolidate compiler generated dependencies of target fsuartservo
[100%] Built target fsuartservo
Install the project...
-- Install configuration: ""
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/lib/libfsuartservo.dll.a
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/bin/libfsuartservo.dll
-- Installing: D:/KyleSoftware/CPP_LIBRARIES/include/FashionStar/UServo/FashionStar_UartServoProtocol.h
-- Up-to-date: D:/KyleSoftware/CPP_LIBRARIES/include/FashionStar/UServo/FashionStar_UartServo.h
PS D:\KyleLab\FashionStar\fashionstar-uart-servo-cpp\build>
```

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。

修改例程的CMake配置文件

`CMakeLists.txt`

```cmake
cmake_minimum_required(VERSION 3.0)
# 设置工程名称，设定版本
project(SERVO_PING VERSION 1.0)

# 动态链接库的安装路径
if(CMAKE_HOST_WIN32)
    # Windows
    # 注意事项：这里需要改成自己的Windows下的动态链接库的安装路径
    # <<<<<<<<<<<<<<<<<<<<<<<<<<<< 改这里
    SET(CMAKE_INSTALL_PREFIX "D:/KyleSoftware/CPP_LIBRARIES/")
elseif(CMAKE_HOST_UNIX)
    # Linux
    # 注意事项：Linux下，CMake默认会将库安装到/usr/local，需要注意的是，安装的时候需要sudo
权限
    # sudo make install
    SET(CMAKE_INSTALL_PREFIX "/usr/local/")
endif()

# 添加依赖的.h文件路径(即hello.h所在的文件夹)
include_directories(${CMAKE_INSTALL_PREFIX}/include)
# # 添加link文件夹
link_directories(${CMAKE_INSTALL_PREFIX}/bin)

# 添加可执行程序hello_demo
add_executable(servo_ping servo_ping.cpp)
# 添加动态链接库的路径
target_link_libraries(servo_ping  fsuartservo cserialport)
```

通过Windows设备管理器查询USB转TTL设备的端口号

修改源码里面的端口号名称。

`servo_ping.cpp` 节选。

```cpp
// 串口总线舵机配置
// 设置串口总线Servo转接板的端口号
#ifdef _WIN32
// <<<<<<<<<<<<<<<<<<<< Windows 端口号改这里
#define SERVO_PORT_NAME "COM8"              // Windows下端口号名称 COM{}
#else
#define SERVO_PORT_NAME "/dev/ttyUSB0"  // Linux下端口号名称 /dev/ttyUSB{}
#endif
#define SERVO_ID 0                          // 舵机ID号
```

举例来讲，舵机通信检测例程 `servo_ping`，进入到工程目录下。

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

> 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。

执行编译出来的可执行文件：

```
.\servo_ping.exe
```