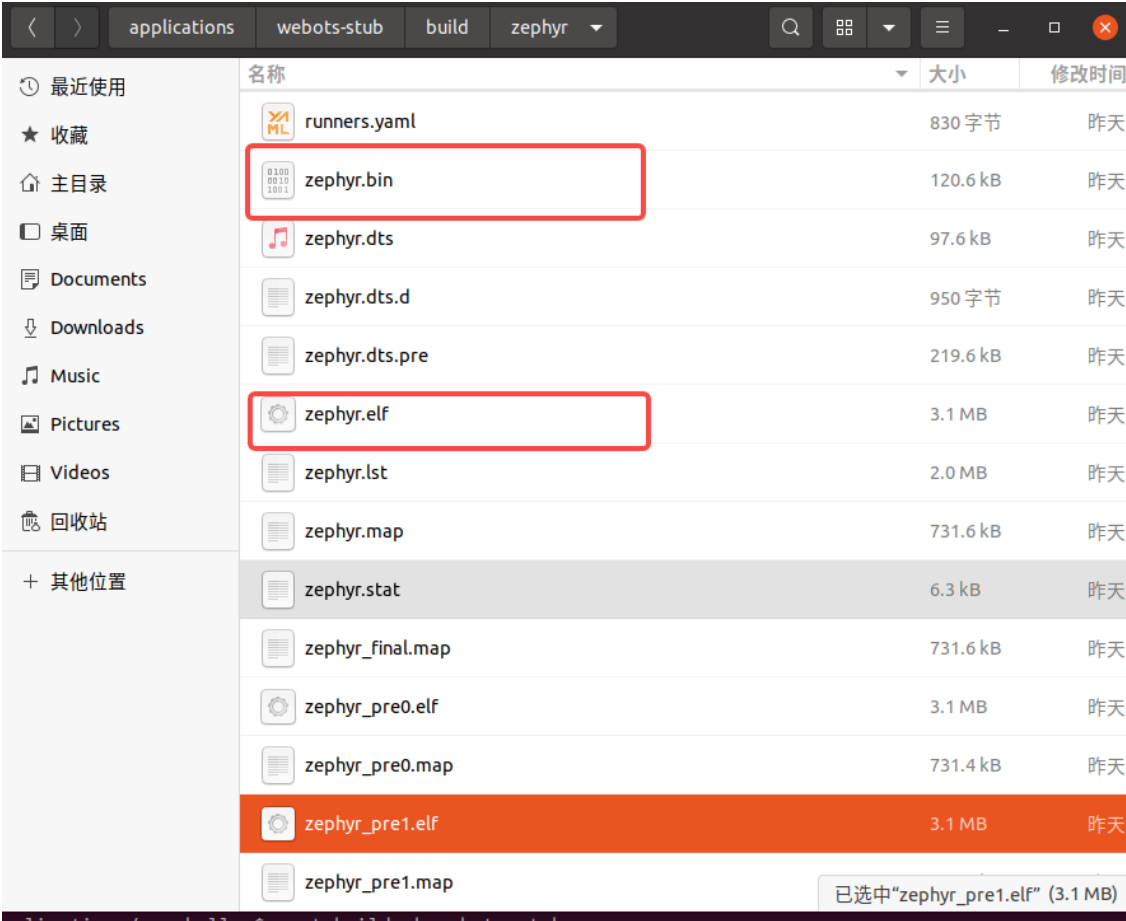


# zephyr运控使用说明

1. 参考zephyr官方文档  
([https://docs.zephyrproject.org/latest/develop/getting\\_started/index.html](https://docs.zephyrproject.org/latest/develop/getting_started/index.html))安装好编译环境。下面路径以Linux/macOS为例，Windows的同学修改一下路径。
2. 把官方文档的“Get the Zephyr source code:” 这个一步里面的  
west init ~/zephyrproject  
替换为公司的代码库，如下  
west init -m git@git.ddt.dev:rbt/mcu/diablo\_cerebellum.git --mr dev  
~/zephyrproject
3. 拉取代码后更新一下  
west update
4. 先激活虚拟环境  
source ~/zephyrproject/.venv/bin/activate
5. 运控板在新的设计里叫webots-stub，进入webots-stub目录就可以编译固件  
cd ~/zephyrproject/directdrivetech/applications/webots-stub  
west build -b diablo\_apollo\_evk


6. 编译好的文件在里面



## 补充说明

1. 注意有一个出现硬件中断的bug，若出现硬件中断，将下面的左侧语句改成右侧语句。

| cd ~/zephyrproject/directdrivetech/ap...

 刘可心 2023年5月15日  
cd  
~/zephyrproject/directdrivetec  
h/applications/cerebellum

```
zephyr > drivers > serial > C:\zephyr\project\4.1\uart_mcu_lpuart.c
858 | data->tx_poll_stream on = false;
859 | mcux_lpuart_pm_policy_state_lock_put(dev);
860 | }
861 | assert(LPUART_ClearStatusFlags(config->base,
862 |   KLUART_TransmissionCompleteFlag) == 0U);
863 | }
864 | #endif /* CONFIG_PM */
865 |
866 | #if CONFIG_UART_INTERRUPT_DRIVEN
867 | if (data->callback) {
868 |   data->callback(dev, data->cb_data);
869 | }
870 |
871 | if (status & KLUART_RxOverrunFlag) {
872 |   LPUART_ClearStatusFlags(config->base, KLUART_RxOverrunFlag);
873 | }
874 | #endif
875 |
876 | #if CONFIG_UART_ASYNC_API
877 | if (status & KLUART_IdleLineFlag) {
878 |   async_timer_start(&data->async_rx_dma_params.timeout_work,
879 |     data->async_rx_dma_params.timeout_us,
880 |     assert(LPUART_ClearStatusFlags(config->base, KLUART_IdleLineFlag) == 0U);
881 |   );
882 | }
883 | #endif /* CONFIG_UART_ASYNC_API */
884 | #endif /* CONFIG_UART_MCUX_LPUART_ISR_SUPPORT */
885 |
886 | static int mcux_lpuart_configure_init(const struct device *dev, const struct uart_config
887 | {
888 |   const struct mcux_lpuart_config *config = dev->config;
889 |   struct mcux_lpuart_data *data = dev->data;
890 |   uint32_t clock_freq;
891 |
892 |   if (!device_is_ready(config->clock_dev)) {
893 |     return -ENODEV;
894 |   }
895 |
896 |   if (clock_control_get_rate(config->clock_dev, config->clock_subsys,
897 |     &clock_freq)) {
898 |     return -EINVAL;
899 |   }
900 |
901 |   lpuart_config_t uart_config;
```

- 2. 在新的设计里，A1/A1.1使用STM32F407运控板在代码里的名字叫diablo\_athena，A2现在的测试板叫diablo\_apollo\_evk，如果要生成A2的固件把上面的-b diablo\_athena替换为-b cd 就可以了
- 3. 目前代码兼容A1.0的结构和电机，兼容A1.0/A1.1（F407）和A2 EVK（NXP1061）两款运控板，A1.1的电机兼容在适配中
- 4. 编译A2 EVK的固件需要手工修改modules/hal/nxp下面的两个文件，这个问题在后续的版本中将会修正。

