

main函数进入261行CreatPowerTask();

CreatPowerTask创建任务Task_Power, 并进行Power_Timer_Init

Power_Timer_Init创建10ms的定时器,回调nvdPowerTimer_10ms_Handle

nvdPowerTimer_10ms_Handle按10ms定时重复发送信号量(TASK_ID_POWER, TASK_ID_POWER, EV_POWER_MGR_10ms, NULL, 0, 0)

随后在Task_Power接收到匹配的信号量并开启对应的回调函数PowerCheck,并传递data (NULL)

PowerCheck开启并接收data,针对下方Set_KL15_Value函数获取的KL15做出处理

main函数进入252行 if (xTaskCreate(baseband_task, "baseband", 2048, (void *)0, TSK_PRIOR_HI, &bb_handle) 创建任务baseband_task baseband_task中获取KL15的值 Set_KL15_Value();

```
if(KL15 < 0.8)
{标志位kl15_0按10ms定时器加到30000走了5min
{

}
5min后
{
    gpio_write(CAN_TRANS_SB_IO, 0); //CAN0 STB to Low
    pwm_start(PWM2_ID, PWM2_FRE, 0, NULL); //MCU_HOLD_ON Close
    //睡眠
}

else if(KL15 > 1)
{
    KL15St = PowerON;
    pwm_start(PWM2_ID, PWM2_FRE, 100, NULL);
    //唤醒
}
}
```

随后用GetKL15St输出KL15St (bool) , 给这三个任务用作判断条件

- ✓ • `GetKL15St() : uint8_t`
- ✓ • `nvdCanTime100msTaskHandle(Type_stMSG *) : void`
 - `{init nstCanTxEvt_Tbl}() : const CanTxEvt_st []`
- ✓ • `nvdCanTime10msTaskHandle(Type_stMSG *) : void`
 - `{init nstCanTxEvt_Tbl}() : const CanTxEvt_st []`
- ✓ • `nvdLedLightModeHandle() : void`
 - > • `nvdCanTime10msTaskHandle(Type_stMSG *) : void`