Rockchip Developer Guide RT-Thread CAN&CANFD

文件标识：RK-KF-YF-162

发布版本：V0.1.0

日期：2024-09-09

文件密级：□绝密 □秘密 □内部资料 ■公开

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址：　　福建省福州市铜盘路软件园A区18号

网址：　　www.rock-chips.com

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：fae@rock-chips.com

前言

概述

产品版本

芯片名称	功能	版本
RK3568	CAN	RT-Thread&HAL
RK2118	CANFD	RT-Thread&HAL
RK3576	CANFD	RT-Thread&HAL
RK3506	CANFD	RT-Thread&HAL

读者对象

本文档（本指南）主要适用于以下工程师：

技术支持工程师

软件开发工程师

修订记录

版本号	作者	修改日期	修改说明
V0.1.0	张晴	2024-09-09	初始版本

目录

1. CAN&CANFD 配置

1.1 HAL CAN&CANFD

1.1.1 驱动

驱动文件所在位置：

`bsp/rockchip/common/hal/lib/hal/src/hal_canfd.c`

1.1.2 常用 API

```
HAL_Status HAL_CANFD_Config(struct CAN_REG *pReg, eCANFD_Bps nbps, eCANFD_Bps
dbps);
HAL_Status HAL_CANFD_Init(struct CAN_REG *pReg, struct CANFD_CONFIG *initStrust);
HAL_Status HAL_CANFD_Start(struct CAN_REG *pReg);
HAL_Status HAL_CANFD_Stop(struct CAN_REG *pReg);
HAL_Status HAL_CANFD_SetNBps(struct CAN_REG *pReg, eCANFD_Bps bsp);
HAL_Status HAL_CANFD_SetDBps(struct CAN_REG *pReg, eCANFD_Bps bps);
HAL_Status HAL_CANFD_Transmit(struct CAN_REG *pReg, struct CANFD_MSG *TxMsg);
HAL_Status HAL_CANFD_Receive(struct CAN_REG *pReg, struct CANFD_MSG *RxMsg);
uint32_t HAL_CANFD_GetInterrupt(struct CAN_REG *pReg);
uint32_t HAL_CANFD_GetErrInterruptMaskCombin(eCANFD_IntType type);
```

1.1.3 初始化

```
HAL_Status HAL_CANFD_Config(struct CAN_REG *pReg, eCANFD_Bps nbps, eCANFD_Bps
dbps);
HAL_Status HAL_CANFD_Init(struct CAN_REG *pReg, struct CANFD_CONFIG *initStrust);
HAL_Status HAL_CANFD_Start(struct CAN_REG *pReg);
```

1.1.4 TX和RX

```
HAL_Status HAL_CANFD_Transmit(struct CAN_REG *pReg, struct CANFD_MSG *TxMsg);
HAL_Status HAL_CANFD_Receive(struct CAN_REG *pReg, struct CANFD_MSG *RxMsg);
```

1.2 RT-Thread CAN 配置

1.2.1 RT-Thread CAN 接口

```
int rockchip_canfd_dev_init(void)
```

1.2.2 RT-Thread CAN 宏配置

使用示例：

```
diff --git a/bsp/rockchip/rk3506-32/hal_conf.h b/bsp/rockchip/rk3506-
32/hal_conf.h
index ceba993bc0f9..b9add69a24dc 100644
--- a/bsp/rockchip/rk3506-32/hal_conf.h
+++ b/bsp/rockchip/rk3506-32/hal_conf.h
@@ -43,6 +43,10 @@
 #define HAL_CRU_MODULE_ENABLED
 #endif

+#ifdef RT_USING_CAN
+#define HAL_CANFD_MODULE_ENABLED
+#endif
+
```

```
diff --git a/bsp/rockchip/rk3506-32/rtconfig.h b/bsp/rockchip/rk3506-
32/rtconfig.h
index aaf767598e7d..2ca4bab357bf 100644
--- a/bsp/rockchip/rk3506-32/rtconfig.h
+++ b/bsp/rockchip/rk3506-32/rtconfig.h
@@ -86,6 +89,7 @@
 #define RT_USING_SERIAL
 #define RT_USING_SERIAL_V1
 #define RT_SERIAL_RB_BUFSZ 512
+#define RT_USING_CAN
+#define RT_USING_CAN0
```

1.2.3 RT-Thread CAN收发示例

使用示例：

默认代码目前只有can_sample,只发送一帧。

使用示例：

```
can_sample rk_can0
```

如果需要多帧发送，或者不同帧格式发送，可以使用如下补丁：

使用示例：

```
diff --git a/bsp/rockchip/common/drivers/drv_canfd.c
b/bsp/rockchip/common/drivers/drv_canfd.c
index 3438e2c3990c..38888bcaeac9 100644
--- a/bsp/rockchip/common/drivers/drv_canfd.c
+++ b/bsp/rockchip/common/drivers/drv_canfd.c
@@ -565,6 +565,90 @@ int can_sample(int argc, char *argv[])
     return res;
 }
 MSH_CMD_EXPORT(can_sample, can device sample);
+
+int can_open(int argc, char *argv[])
+{
+    rt_err_t res = 0;
+    rt_thread_t thread;
+    char can_name[RT_NAME_MAX];
+
+    if (argc == 2)
+    {
+        rt_strncpy(can_name, argv[1], RT_NAME_MAX);
+    }
+    else
+    {
+        rt_strncpy(can_name, CAN_DEV_NAME, RT_NAME_MAX);
+    }
+    can_dev = rt_device_find(can_name);
+    if (!can_dev)
+    {
+        rt_kprintf("find %s failed!\n", can_name);
+        return RT_ERROR;
+    }
+
+    rt_sem_init(&rx_sem, "rx_sem", 0, RT_IPC_FLAG_FIFO);
+    res = rt_device_open(can_dev, RT_DEVICE_FLAG_INT_TX |
RT_DEVICE_FLAG_INT_RX);
+    RT_ASSERT(res == RT_EOK);
+    rt_device_control(can_dev, RT_CAN_CMD_SET_MODE, (void *)RT_CAN_MODE_NORMAL);
+    thread = rt_thread_create("can_rx", can_rx_thread, RT_NULL, 2048, 25, 10);
+    if (thread != RT_NULL)
+    {
+        rt_thread_startup(thread);
+    }
+    else
+    {
+        rt_kprintf("create can_rx thread failed!\n");
+    }
+    return res;
+}
+MSH_CMD_EXPORT(can_open, can device open);
+
+int can_tx(int argc, char *argv[])
+{
+    struct rt_can_msg msg = {0};
+    char can_name[RT_NAME_MAX];
+    rt_err_t res = 0;
+    rt_size_t  size;
+    int i = 0, j = 0;
+    int ide, rtr, len, cnt, ms;
+    rt_uint32_t id;
+
+    rt_strncpy(can_name, argv[1], RT_NAME_MAX);
+    rt_strncpy(can_name, CAN_DEV_NAME, RT_NAME_MAX);
+    ide = strtol(argv[2], NULL, 10);
+    rtr = strtol(argv[3], NULL, 10);
+    len = strtol(argv[4], NULL, 10);
+    id = strtol(argv[5], NULL, 16);
+    cnt = strtol(argv[6], NULL, 10);
+    ms = strtol(argv[7], NULL, 10);
+
+    for (i = 0; i < cnt; i++) {
+        msg.ide = ide;
+        if (msg.ide)
+            msg.id = (unsigned int)rand() & 0x1fffffff;
+        else
+            msg.id = (unsigned int)rand() & 0x7ff;
+
+        if (id)
+           msg.id = id;
+        msg.rtr = rtr;
+        msg.len = len;
+        for (j = 0; j < msg.len; j++) {
+            msg.data[j] = (unsigned int)rand();
+        }
+        size = rt_device_write(can_dev, 0, &msg, sizeof(msg));
+        if (size == 0)
+        {
+            rt_kprintf("can dev write data failed!\n");
+        }
+        if (ms)
+            rt_thread_mdelay(ms);
+
+    }
+    return res;
+}
+MSH_CMD_EXPORT(can_tx, can device tx);
```

接收：

```
can_open rk_can0
```

接收并发送：

```
can_open rk_can0
can_tx rk_can0 0 0 8 0 10 1 //间隔1ms 连续发10帧 标准帧 数据帧
can_tx rk_can0 1 0 8 0 10 1 //间隔1ms 连续发10帧 扩展帧 数据帧
can_tx rk_can0 0 1 8 0 10 1 //间隔1ms 连续发10帧 标准帧 远程帧
can_tx rk_can0 1 1 8 0 10 1 //间隔1ms 连续发10帧 扩展帧 远程帧
```

1.2.4 RT-Thread CAN 比特率配置

```
rockchip_canfd0.config.baud_rate = CAN500kBaud;
```

1.3 RT-Thread CANFD 配置

RT-Thread目前还没有标准的CANFD变速，如果需要支持CANFD变速，目前只能强制修改hal_canfd.c去配置变速段的比特率。其他的配置按照CAN配置即可。

使用示例：

```
HAL_CANFD_Config(pReg, initStrust->bps, CANFD_BPS_2MBAUD);
```

接收：

```
can_open rk_can0
```

接收并发送：

```
can_open rk_can0
can_tx rk_can0 0 0 64 0 10 1 //间隔1ms 连续发10帧 标准帧 数据帧 64byte
can_tx rk_can0 1 0 64 0 10 1 //间隔1ms 连续发10帧 扩展帧 数据帧 64byte
```

1.4 RT-Thread CAN&CANFD IOMUX配置

使用示例：

在iomux.c中增加下：

```
diff --git a/bsp/rockchip/rk3506-32/board/evb1/iomux.c b/bsp/rockchip/rk3506-
32/board/evb1/iomux.c
index 941e1aaa342a..bd5440082cc1 100644
--- a/bsp/rockchip/rk3506-32/board/evb1/iomux.c
+++ b/bsp/rockchip/rk3506-32/board/evb1/iomux.c
@@ -20,6 +20,22 @@ void rt_hw_iodomain_config(void)
 {
 }

+#ifdef RT_USING_CAN0
+/**
+ * @brief  Config iomux for CAN0
+ */
+void can0_iomux_config(void)
+{
+    HAL_PINCTRL_SetRMIO(GPIO_BANK1,
+                        GPIO_PIN_D2,
+                        RMIO_CAN0_TX);
+
+    HAL_PINCTRL_SetRMIO(GPIO_BANK1,
+                         GPIO_PIN_D3,
+                         RMIO_CAN0_RX);
+}
+#endif
+
 void sai1_iomux_config(void)
 {
     HAL_PINCTRL_SetIOMUX(GPIO_BANK0,
@@ -48,6 +64,10 @@ void rt_hw_iomux_config(void)
     i2c0_iomux_config();
 #endif

+#if defined(RT_USING_CAN0)
+    can0_iomux_config();
+#endif
+
 #if defined(RT_USING_SDIO0)
     emmc_iomux_config();
 #endif
```