

Time interval controller (TIC)

Таблица активности временных слотов. (timeSlotStateTable)

Состояние временного слота 0 timeSlotState	Состояние временного слота 1 timeSlotState	Состояние временного слота N timeSlotState	Состояние временного слота 49 timeSlotState
Флаг приема	Флаг приема	Флаг приема	Флаг приема
Флаг передачи	Флаг передачи	Флаг передачи	Флаг передачи

Аппаратный таймер 32.768K

setTimer(uint16_t time)
setInterruptTime(uint16_t time, timeslotNum)
setInterruptHandler((*handlerIRQ)(*timeslot))
adjustTimer (int adjustTime)

Обработчик аппаратного таймера hwTimerHandler(timeSlot)

Проверяем поле "Флаг передачи" таблицы timeSlotStateTable
Если поле TRUE
 Вызываем обработчик передачи пакета(timeSlot)
 Вызываем TDMA Scheduler(timeSlot)
 Вызываем обработчик slotEnd(timeSlot)
 ВЫХОД
Проверяем поле "Флаг приема" таблицы timeSlotStateTable
Если поле TRUE
 Вызываем обработчик приема пакета(timeSlot)
 Вызываем TDMA Scheduler(timeSlot)
 Вызываем обработчик slotEnd(timeSlot)
 ВЫХОД
Вызываем TDMA Scheduler(timeSlot)
Вызываем обработчик slotEnd(timeSlot)
ВЫХОД

Планировщик TDMA. TDMA Scheduler(timeSlot)

Перебираем следующие timeSlot из timeSlotStateTable
Если "Обработчик прием пакета" не равен NULL
ИЛИ "Обработчик передачи пакета" не равен NULL
 Устанавливаем аппаратный таймер на этот таймслот
 ВЫХОД
Тут мы окажемся если обработчиков нет.
Устанавливаем аппаратный таймер на таймслот 0

Time interval controller (TIC)

init()	Инициализация
--------	---------------

setReceiveCallback(ticCallback_f handler)	Устанавливает обработчик интервала приема пакета
setSendCallback(ticCallback_f handler)	Устанавливает обработчик интервала передачи пакета
setSlotEndCallback(ticCallback_f handler)	Устанавливает обработчик по завершению слота
clearCallbacks()	Удаляет все обработчики

bool setReceptionState(timeSlot_t n, bool state)	Установить таймслот в режим приема пакетов
bool setTransmissionState(timeSlot_t n, bool state)	Установить таймслот в режим передачи пакета
bool getReceptionState(timeSlot_t n)	Прочитать активность состояния приема пакета
bool getTransmissionState(timeSlot_t n)	Прочитать активность состояния передачи пакета

bool adjustTimer(usec_t usec)	Подстроить значение таймера
bool setTimer(usec_t usec)	Установить значение таймера
usec_t getTimer()	Прочитать состояние таймера

ticCallback_Type

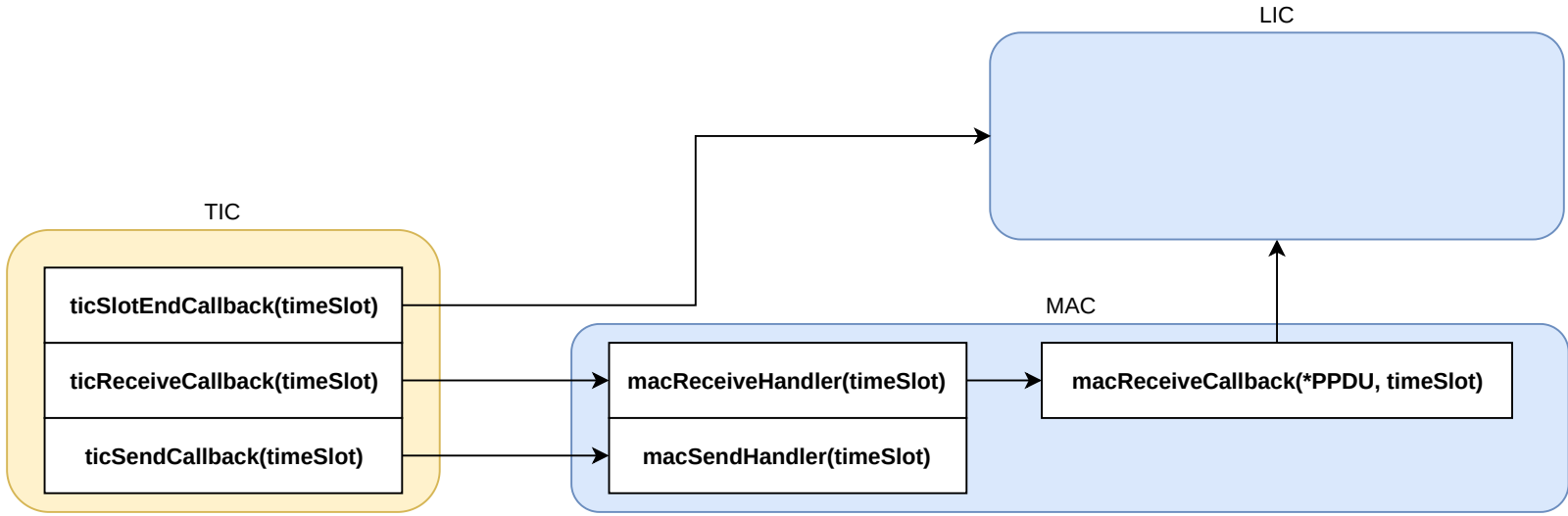
```
typedef void (*ticCallback_f)(timeSlot_t n)
```

Media access control (MAC)

Состояние слотов. (slotStateTable)

Состояние слота 0 slotState	Состояние слота N slotState	Состояние слота 49 slotState
Состояние передачи	Состояние передачи	Состояние передачи
Флаг "Есть данные"	Флаг "Есть данные"	Флаг "Есть данные"
Количество попыток передачи	Количество попыток передачи	Количество попыток передачи
Канал радиопередатчика	Канал радиопередатчика	Канал радиопередатчика
Указатель на пакет PPDU	Указатель на пакет PPDU	Указатель на пакет PPDU
Состояние приема	Состояние приема	Состояние приема
Флаг "Прием разрешен"	Флаг "Прием разрешен"	Флаг "Прием разрешен"
Канал радиопередатчика	Канал радиопередатчика	Канал радиопередатчика

Принятый пакет receivedPacket
Указатель на пакет PPDU
timeSlot принятого пакета



Обработчик ticCallback_f sendHandler(timeSlot)
Процедура обработки события передачи пакета TIC Если флаг "Есть данные" = FALSE Запрещаем TIC обработку. setTransmissionState(timeSlot, FALSE) ВЫХОД Включить радиопередатчик, выставить частоту канала передачи Если PPDU TYPE = WP Начать передачу Если передача не удалась (контроль CCA) Количество попыток - 1 Если количество попыток = 0 Уничтожаем пакет, флаг "Есть данные" = FALSE Запрещаем TIC обработку. setTransmissionState(timeSlot, FALSE) Выключаем радио, ВЫХОД Прием пакета в течении 1 мс(?) Если приняли ACK Уничтожаем пакет, флаг "Есть данные" = FALSE Запрещаем TIC обработку. setTransmissionState(timeSlot, FALSE) Выключаем радио, ВЫХОД Если не приняли ACK Количество попыток - 1 Если количество попыток = 0 Уничтожаем пакет, флаг "Есть данные" = FALSE Запрещаем TIC обработку. setTransmissionState(timeSlot, FALSE) Выключаем радио, ВЫХОД ЕСЛИ PPDU TYPE = SYNC Заполнить поля HOUR, MIN, SEC, USEC = 0x80000000 Начать передачу специальной функцией Если передача не удалась Количество попыток - 1 Если количество попыток = 0 Уничтожаем пакет, флаг "Есть данные" = FALSE Запрещаем TIC обработку. setTransmissionState(timeSlot, FALSE) Если передача удалась Уничтожаем пакет, флаг "Есть данные" = FALSE Запрещаем TIC обработку. setTransmissionState(timeSlot, FALSE) Выключаем радио, ВЫХОД

Обработчик ticCallback_f receiveHandler(timeSlot)
Процедура обработки события приема пакета TIC Если флаг "Прием разрешен" равен FALSE Запрещаем TIC обработку. setTransmissionState(timeSlot,FALSE) ВЫХОД Включить радиопередатчик, выставить частоту канала приема Ждать приема не более 2-3 мс(?) Если пакета не пришло Выключить радио, ВЫХОД Если проверка NETID разрешена Если NETID не равен нашей сети Уничтожаем, выключаем радио, ВЫХОД Если версия протокола не равна нашей версии Уничтожаем, выключаем радио, ВЫХОД Если PPDU TYPE = WP Если размер пакета меньше 26 байт Уничтожаем, выключаем радио, ВЫХОД Если поле DST не соответствует адресу узла Уничтожаем, выключаем радио, ВЫХОД Если поле DLEN не равно размеру PPDU - 26 (размер DATA неверен) Уничтожаем, выключаем радио, ВЫХОД Передаем пакет ACK Выключаем радио Вызов обработчика приема пакета. macReceiveCallBack(*PPDU, timeSlot) ВЫХОД

Media access control (MAC)

init()	Инициализация
setReceiveCallback(ticCallback_f handler)	Устанавливает обработчик интервала приема пакета
setSendCallback(ticCallback_f handler)	Устанавливает обработчик интервала передачи пакета
clearCallbacks()	Удаляет все обработчики
bool setReceptionState(timeSlot_t n, bool state)	Установить таймслот в режим приема пакетов
bool setTransmissionState(timeSlot_t n, bool state)	Установить таймслот в режим передачи пакета
bool getReceptionState(timeSlot_t n)	Прочитать активность состояния приема пакета
bool getTransmissionState(timeSlot_t n)	Прочитать активность состояния передачи пакета
bool adjustTimer(usec_t usec)	Подстроить значение таймера
bool setTimer(usec_t usec)	Установить значение таймера
usec_t getTimer()	Прочитать состояние таймера

ticCallback:Type

```
typedef void (*ticCallback_f)(timeSlot_t n)
```

