2.3.2 URB Functions

There are four functions of the USB core that handle URBs.

**•** **purb t usb alloc urb(int iso packets);**

Whenever a URB structure is needed this function has to be called. The argument iso packets is used to specify the number of iso frame desc structures at the end of the URB structure when setting up isochronous transfers. If successful the return value is a pointer to a URB structure preset to zero otherwise a NULL pointer is returned.

Всякий раз, когда требуется структура URB, эта функция должна вызываться. Аргумент iso-пакетов используется для указания количества структур iso-кадра в конце структуры URB при настройке изохронных передач. В случае успеха возвращаемое значение является указателем на структуру URB, предварительно установленную в ноль, в противном случае возвращается указатель NULL.

**• void usb free urb (purb t purb);**

To free memory allocated by usb alloc urb simply call this function.

Чтобы освободить память, выделенную usb alloc urb, просто вызовите эту функцию.

**• int usb submit urb(purb t purb);**

This function sends a transfer request asynchronously to the USB core. The argument purb is a pointer to a previously allocated and initialized URB structure. If successful the return value is 0 otherwise an appropriate error code is returned (see section 3.1). The function returns always non-blocking and it is possible to schedule several URBs for different endpoints without waiting. On isochronous endpoints it is even possible to schedule more URBs for one endpoint. This limitation is caused due to error handling and retry mechanisms of the USB protocol (see [4] section 8.5)

Эта функция асинхронно отправляет запрос на передачу в ядро USB. Аргумент purb является указателем на ранее выделенную и инициализированную структуру URB. В случае успеха возвращаемое значение равно 0, в противном случае возвращается соответствующий код ошибки (см. Раздел 3.1). Функция всегда возвращает неблокирование, и можно запланировать несколько URB для разных конечных точек без ожидания. На изохронных конечных точках даже можно запланировать больше URB для одной конечной точки. Это ограничение вызвано механизмами обработки ошибок и повторных попыток протокола USB (см. [4] раздел 8.5)

**• int usb unlink urb(purb t purb);**

This function cancels a scheduled request before it is completed. The argument purb is a pointer to a previously submitted URB structure. The function can be called synchronously or asynchronously depending on the transfer flag USB ASYNC UNLINK (see 2.3.1). Synchronously called the function waits for 1ms and must not be called from an interrupt or completion handler. The return value is 0 if the function succeeds. Asynchronously called the function returns immediately. The return value is -EINPROGRESS if the function was successfully started. When calling usb unlink urb the completion handler is called after the function completed. The URB status is marked with -ENOENT (synchronously called) or -ECONNRESET (asynchronously called).

Эта функция отменяет запланированный запрос до его завершения. Аргумент purb является указателем на ранее переданную структуру URB. Функцию можно вызывать синхронно или асинхронно в зависимости от флага передачи USB ASYNC UNLINK (см. 2.3.1). Синхронно вызываемая функция ожидает 1 мс и не должна вызываться из обработчика прерываний или завершения. Возвращаемое значение равно 0, если функция завершается успешно. Асинхронно вызванная функция сразу возвращается. Возвращаемое значение равно -EINPROGRESS, если функция была успешно запущена. При вызове usb unlink urb обработчик завершения вызывается после завершения функции. Состояние URB помечается как -ENOENT (синхронно вызывается) или -ECONNRESET (асинхронно вызывается).

usb unlink urb is also used to stop an interrupt transfer URB. As documented in sections 1.2.2, 2.3.1 interrupt transfers are automatically rescheduled. Call usb unlink urb even for “one shot interrupts”.

usb unlink urb также используется для остановки передачи прерываний URB. Как указано в разделах 1.2.2, 2.3.1, передача прерываний автоматически переносится. Вызовите usb unlink urb даже для «прерываний одним выстрелом».

2.3.3 URB Macros

To initialize URB structures for different transfer types there exist some macros:

Для инициализации структур URB для разных типов передачи существуют несколько макросов:

• FILL CONTROL URB(purb, dev, pipe, setup packet, transfer buffer, transfer buffer length, complete, context); • FILL BULK URB(purb, dev, pipe, transfer buffer, transfer buffer length, complete, context);

• FILL INT URB(purb, dev, pipe, transfer buffer, transfer buffer length, complete, context, interval);

• FILL CONTROL URB TO();

• FILL BULK URB TO(); FILL BULK URB TO The macros are self explaining - more documentation can be found in the include file usb.h.