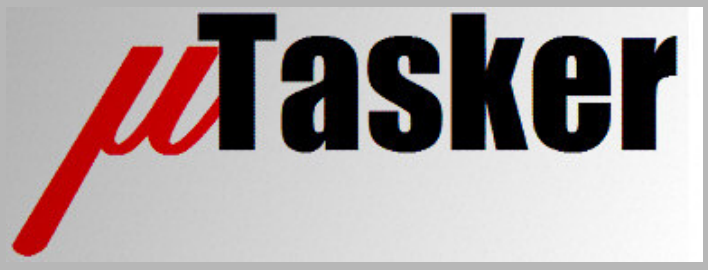
Embedding it better...



µTasker Document

AVR32 USB Device

Table of Contents

1. Introduction

AVR32 USB controller supports 2.0 low and full speed device and low and full speed On-The-Go (OTG) modes of operation. A flexible end-point configuration and management is possible with dedicated DMA channels. The on-chip transceivers include integrated pull-ups.

7 pipes/endpoints and 960 bytes embedded dual-port RAM are available for their use.

Dual-buffer operation can be performed on non-control endpoints in order to achieve isochronous transfer speeds.

2. Configuration

The USB controller needs an accurate 48MHz clock +/- 0.25% to derive its 12MHz USB clock. This is generated by a dedicated generic clock from the power manager. This 48MHz clock is enabled as first activity before using the USB.

- AT32UC3B devices contain 4 generic clocks. Generic clocks 3 is for use by the USBB
- AT32UC3A devices contain 6 generic clocks. Generic clocks 4 is for use by the USBB

The generic clocks takes their inputs from either oscillators 0 or 1, or PLL 0 or 1. Since the oscillators have a maximum speed of 16MHz it follows that the USB operation requires a PLL (0 or 1) to be set to operate at 48MHz in order to be able to work.

When the HW has only one oscillator (either due to the fact that only one is mounted or because a smaller package doesn't have inputs for oscillator 1) it means that the main PLL must be set to 48MHz (rather than maximum possible of 66MHz).

> Note: The ATMEL EVK1101 development board doesn't have oscillator 1 mounted as standard and so the development for this board was performed with PLL0 at 48MHz to satisfy the USB requirement.

Before the USB controller registers can be controlled the USB PBB and HSB clocks need to be enable (*note that all clocks are disabled by default in the µTasker project in order to optimise power consumption when peripherals are not actively used*).

The mode of operation (Device or Host) is determined by the `USB_ID` pin. '1' sets device mode and '0' sets host mode. In order to force device mode of operation when `USB_ID` is not used the flag `UIMOD` (USB Macro Mode) is set to the `USBCON` register and the `UIDE` bit (`USB_ID` pin enable) is cleared. When the USB controller is enabled it will now automatically select device mode.

An internal pull-up resistor to `VBUS` can be internally connected to either `D+` or `D-`. This is controlled by `UDCON_LS` in the `UDCON` register [`UDCON_LS = 0` for full speed – *pull-up on D+* or `UDCON_LS = 1` for low speed – *pull-up on D-*]. In order for this to be physically connected, the OTG pad must also be enabled by setting `USBCON_OTGPADE` in `USBCON`.

After this simple configuration and enabling end of reset and suspend state change interrupts the USB driver can detect suspend states (when the cable has been removed) and resets (when the USB host commands resets, which is the first action after connecting the USB cable).

3. Control Endpoint 0

The control endpoint 0 is the only one that is enabled during the enumeration phase. This endpoint can only operate in single buffer control mode and has no DMA support. It can be allocated 6, 16, 32 or 64 bytes of buffer space in the dual-port RAM.

To do..

4. Configuration Endpoints

When enumeration completes certain endpoints are enabled depending on the specific configuration. The AVR32 supports up to 6 additional endpoints which all support dual-buffered operation and can be allocated a share of the dual-port RAM.

To do..

5. Conclusion

This document is in progress and has not been officially released.

Modifications:
- V0.00 5.6.2010 – provisional version for the documentation page for development monitoring