Atmel Microcontrollers

AT09338: USB Device Interface (UDI) for Mass Storage Class (MSC)

APPLICATION NOTE

Introduction

USB Device Interface (UDI) for Mass Storage Class (MSC) provides an interface for the configuration and management of USB MSC storage device.

The outline of this documentation is as follows:

- API Overview
- Quick Start Guide for USB Device Mass Storage Module (UDI MSC)
- Configuration File Examples

For more details for Atmel® Software Framework (ASF) USB Device Stack, refer to following application notes:

- AVR4900: ASF - USB Device Stack
- AVR4920: ASF - USB Device Stack - Compliance and Performance Figures
- AVR4921: ASF - USB Device Stack Differences between ASF V1 and V2

Table of Contents

1. Software License

2. API Overview

2.1. Variable and Type Definitions

2.1.1. UDI MSC Interface for UDC

2.1.1.1. Variable udi_api_msc

```
UDC_DESC_STORAGE udi_api_t udi_api_msc
```

Global structure which contains standard UDI interface for UDC.

2.2. Structure Definitions

2.2.1. Struct udi_msc_desc_t

Interface descriptor structure for MSC.

Table 2-1. Members

Type	Name	Description
usb_ep_desc_t	ep_in	Data IN endpoint descriptors
usb_ep_desc_t	ep_out	Data OUT endpoint descriptors
usb_iface_desc_t	iface	Standard USB interface descriptor structure

2.3. Macro Definitions

2.3.1. USB Interface Descriptors

The following structures provide predefined USB interface descriptors. It must be used to define the final USB descriptors.

2.3.1.1. Macro UDI_MSC_STRING_ID

```
#define UDI_MSC_STRING_ID
```

By default no string is associated to this interface.

2.3.1.2. Macro UDI_MSC_EPS_SIZE_FS

```
#define UDI_MSC_EPS_SIZE_FS
```

MSC endpoints size for full speed.

2.3.1.3. Macro UDI_MSC_EPS_SIZE_HS

```
#define UDI_MSC_EPS_SIZE_HS
```

MSC endpoints size for high speed.

2.3.1.4. Macro UDI_MSC_DESC

```
#define UDI_MSC_DESC
```

Content of MSC interface descriptor for all speeds.

2.3.1.5. Macro UDI_MSC_DESC_FS

```
#define UDI_MSC_DESC_FS
```

Content of MSC interface descriptor for full speed only.

2.3.1.6. Macro UDI_MSC_DESC_HS

```
#define UDI_MSC_DESC_HS
```

Content of MSC interface descriptor for high speed only.

2.4. Function Definitions

2.4.1. Function udi_msc_process_trans()

Process the background read/write commands.

```
bool udi_msc_process_trans( void )
```

Routine called by the main loop.

2.4.2. Function udi_msc_trans_block()

Transfers data to/from USB MSC endpoints.

```
bool udi_msc_trans_block(
        bool b_read,
        uint8_t * block,
        iram_size_t block_size,
        void(*)(udd_ep_status_t status, iram_size_t n, udd_ep_id_t ep) callback)
```

Table 2-2. Parameters

Data direction	Parameter name	Description
[in]	b_read	Memory to USB, if true
[in, out]	block	Buffer on Internal RAM to send or fill
[in]	block_size	Buffer size to send or fill
[in]	callback	Function to call at the end of transfer. If NULL then the routine exit when transfer is finish.

Returns

`1` if function was successfully done, otherwise `0`.

3. Quick Start Guide for USB Device Mass Storage Module (UDI MSC)

This is the quick start guide for the USB Device Interface MSC Module (UDI MSC) with step-by-step instructions on how to configure and use the modules in a selection of use cases.

The use cases contain several code fragments. The code fragments in the steps for setup can be copied into a custom initialization function, while the steps for usage can be copied into, e.g., the main application function.

3.1. Basic Use Case

In this basic use case, the "USB MSC (Single Interface Device)" module is used. The "USB MSC (Composite Device)" module usage is described in Advanced Use Cases.

3.2. Setup Steps

As a USB device, it follows common USB device setup steps. Refer to USB Device Basic Setup.

The USB MSC interface accesses Memory through Common Abstraction Layer (ctrl_access) in ASF. See Common Abstraction Layer for Memory Interfaces.

3.2.1. Common Abstraction Layer for Memory Interfaces

Common abstraction layer (ctrl_access) can provide interfaces between Memory and USB. In USB MSC UDI the read/write invokes following ctrl_access functions:

```
extern Ctrl_status memory_2_usb(U8 lun, U32 addr, U16 nb_sector);
extern Ctrl_status usb_2_memory(U8 lun, U32 addr, U16 nb_sector);
```

Then the ctrl_access dispatch the read/write operation to different Logic Unit Numbers (LUNs).

The memory access in ctrl_access is configured through conf_access.h. E.g., to use LUN0 to access virtual memory disk, the configuration should include:

```
#define LUN_0                   ENABLE // Enable LUN0 access
//...

#define VIRTUAL_MEM             LUN_0
#define LUN_ID_VIRTUAL_MEM      LUN_ID_0
#define LUN_0_INCLUDE           "virtual_mem.h"         // APIs (complied to
ctrl_access)
#define Lun_0_test_unit_ready virtual_test_unit_ready // check disk ready
#define Lun_0_read_capacity   virtual_read_capacity   // get disk size
#define Lun_0_wr_protect      virtual_wr_protect      // check protection
#define Lun_0_removal         virtual_removal         // check if disk is
removable
#define Lun_0_usb_read_10     virtual_usb_read_10     // Disk to USB
transfer
#define Lun_0_usb_write_10    virtual_usb_write_10    // USB to Disk
transfer
#define LUN_0_NAME              "\"On-Chip Virtual Memory\""
//...

#define ACCESS_USB                  true    // USB interface.
//...
```

```
#define GLOBAL_WR_PROTECT         false
```

Since LUN_0 is defined as a "Virtual Memory", the module to encapsulate the internal or on-board memory to access as a disk is included. The configuration of such a virtual memory disk is in conf_virtual_mem.h. E.g., to use internal RAM to build such a memory disk, the configuration should include:

```
#define VMEM_NB_SECTOR 48 //Internal RAM 24KB (should > 20KB or PC can not
format it)
```

For more examples of the control access or disk configuration, refer to conf_access.h and conf_virtual_mem.h.

For more Information about Memory Control Access, refer to the online document:

- Atmel Software Framework - Memory Control Access

3.3. Usage Steps

3.3.1. Example Code

Content of conf_usb.h:

```
#define  USB_DEVICE_SERIAL_NAME  "12...EF" // Disk SN for MSC
#define UDI_MSC_GLOBAL_VENDOR_ID \
   'A', 'T', 'M', 'E', 'L', ' ', ' ', ' '
#define UDI_MSC_GLOBAL_PRODUCT_VERSION \
   '1', '.', '0', '0'
#define UDI_MSC_ENABLE_EXT() my_callback_msc_enable()
extern bool my_callback_msc_enable(void);
#define UDI_MSC_DISABLE_EXT() my_callback_msc_disable()
extern void my_callback_msc_disable(void);
#include "udi_msc_conf.h" // At the end of conf_usb.h file
```

Add to application C-file:

```
static bool my_flag_autorize_msc_transfert = false;
bool my_callback_msc_enable(void)
{
   my_flag_autorize_msc_transfert = true;
   return true;
}
void my_callback_msc_disable(void)
{
   my_flag_autorize_msc_transfert = false;
}

void task(void)
{
   udi_msc_process_trans();
}
```

3.3.2. Workflow

1. Ensure that conf_usb.h is available and contains the following configuration, which is the USB device MSC configuration:

   ```
   #define USB_DEVICE_SERIAL_NAME  "12...EF" // Disk SN for MSC
   ```

Note: The USB serial number is mandatory when a MSC interface is used.

```
#define UDI_MSC_GLOBAL_VENDOR_ID \
   'A', 'T', 'M', 'E', 'L', ' ', ' ', ' '
#define UDI_MSC_GLOBAL_PRODUCT_VERSION \
   '1', '.', '0', '0'
```

Note: The USB MSC interface requires a vendor ID (eight ASCII characters) and a product version (four ASCII characters).

```
#define UDI_MSC_ENABLE_EXT() my_callback_msc_enable()
extern bool my_callback_msc_enable(void);
```

Note: After the device enumeration (detecting and identifying USB devices), the USB host starts the device configuration. When the USB MSC interface from the device is accepted by the host, the USB host enables this interface and the UDI_MSC_ENABLE_EXT() callback function is called and return true. Thus, when this event is received, the tasks which call udi_msc_process_trans() must be enabled.

```
#define UDI_MSC_DISABLE_EXT() my_callback_msc_disable()
extern void my_callback_msc_disable(void);
```

Note: When the USB device is unplugged or is reset by the USB host, the USB interface is disabled and the UDI_MSC_DISABLE_EXT() callback function is called. Thus, it is recommended to disable the task which is called udi_msc_process_trans().

2. The MSC is automatically linked with memory control access component which provides the memories interfaces. However, the memory data transfers must be done outside USB interrupt routine. This is done in the MSC process ("udi_msc_process_trans()") called by main loop:

```
void task(void) {
  udi_msc_process_trans();
}
```

3. The MSC speed depends on task periodicity. To get the best speed the notification callback "UDI_MSC_NOTIFY_TRANS_EXT" can be used to wakeup this task (Example, through a mutex):

```
#define  UDI_MSC_NOTIFY_TRANS_EXT()     msc_notify_trans()
void msc_notify_trans(void) {
  wakeup_my_task();
}
```

3.4. Advanced Use Cases

For more advanced use of the UDI MSC module, see the following use cases:

- MSC in a Composite Device
- Change USB Speed
- Use USB Strings
- Use USB Remote Wakeup Feature
- Bus Power Application Recommendations
- USB Dynamic Serial Number
- Custom Configuration
- VBUS Monitoring

3.5. MSC in a Composite Device

A USB Composite Device is a USB Device which uses more than one USB class. In this use case, the "USB MSC (Composite Device)" module is used to create a USB composite device. Thus, this USB module can be associated with another "Composite Device" module, like "USB HID Mouse (Composite Device)".

Also, you can refer to application note AVR4902 ASF - USB Composite Device.

3.5.1. Setup Steps

For the setup code of this use case to work, the Basic Use Case must be followed.

3.5.2. Usage Steps

3.5.2.1. Example Code

Content of conf_usb.h:

```
#define USB_DEVICE_EP_CTRL_SIZE  64
#define USB_DEVICE_NB_INTERFACE (X+1)
#define USB_DEVICE_MAX_EP (X+2)

#define UDI_MSC_EP_IN  (X | USB_EP_DIR_IN)
#define UDI_MSC_EP_OUT (Y | USB_EP_DIR_OUT)
#define UDI_MSC_IFACE_NUMBER  X

#define UDI_COMPOSITE_DESC_T \
   udi_msc_desc_t udi_msc; \
   ...
#define UDI_COMPOSITE_DESC_FS \
   .udi_msc = UDI_MSC_DESC, \
   ...
#define UDI_COMPOSITE_DESC_HS \
   .udi_msc = UDI_MSC_DESC, \
   ...
#define UDI_COMPOSITE_API \
   &udi_api_msc, \
   ...
```

3.5.2.2. Workflow

1. Ensure that conf_usb.h is available and contains the following parameters required for a USB composite device configuration:

```
// Endpoint control size, This must be:
// - 8, 16, 32 or 64 for full speed device (8 is recommended to save RAM)
// - 64 for a high speed device
#define USB_DEVICE_EP_CTRL_SIZE  64
// Total Number of interfaces on this USB device.
// Add 1 for MSC.
#define USB_DEVICE_NB_INTERFACE (X+1)
// Total number of endpoints on this USB device.
// This must include each endpoint for each interface.
// Add 2 for MSC.
#define USB_DEVICE_MAX_EP (X+2)
```

2. Ensure that conf_usb.h contains the description of composite device:

```
// The endpoint numbers chosen by you for the MSC.
// The endpoint numbers starting from 1.
```

```
#define UDI_MSC_EP_IN  (X | USB_EP_DIR_IN)
#define UDI_MSC_EP_OUT (Y | USB_EP_DIR_OUT)
// The interface index of an interface starting from 0
#define UDI_MSC_IFACE_NUMBER  X
```

3. Ensure that conf_usb.h contains the following parameters required for a USB composite device configuration:

```
// USB Interfaces descriptor structure
#define UDI_COMPOSITE_DESC_T \
   ...
   udi_msc_desc_t udi_msc; \
   ...
// USB Interfaces descriptor value for Full Speed
#define UDI_COMPOSITE_DESC_FS \
   ...
   .udi_msc = UDI_MSC_DESC_FS, \
   ...
// USB Interfaces descriptor value for High Speed
#define UDI_COMPOSITE_DESC_HS \
   ...
   .udi_msc = UDI_MSC_DESC_HS, \
   ...
// USB Interface APIs
#define UDI_COMPOSITE_API \
   ...
   &udi_api_msc, \
   ...
```

Note: The descriptors order given in the four lists above must be the same as the order defined by all interface indexes. The interface index orders are defined through UDI_X_IFACE_NUMBER defines.

3.6. Change USB Speed

In this use case, the USB device is used with different USB speeds.

3.6.1. Setup Steps

Prior to implement this use case, be sure to have already applied the UDI module "basic use case".

3.6.2. Usage Steps

3.6.2.1. Example Code

Content of conf_usb.h:

```
#if // Low speed
#define USB_DEVICE_LOW_SPEED
// #define USB_DEVICE_HS_SUPPORT

#elif // Full speed
// #define USB_DEVICE_LOW_SPEED
// #define USB_DEVICE_HS_SUPPORT
#elif // High speed
// #define USB_DEVICE_LOW_SPEED
#define USB_DEVICE_HS_SUPPORT

#endif
```

3.6.2.2. Workflow

1. Ensure that conf_usb.h is available and contains the following parameters required for a USB device low speed (1.5Mbit/s):

```
#define USB_DEVICE_LOW_SPEED
//#define  USB_DEVICE_HS_SUPPORT
```

2. Ensure that conf_usb.h contains the following parameters required for a USB device full speed (12Mbit/s):

```
//#define USB_DEVICE_LOW_SPEED
//#define  USB_DEVICE_HS_SUPPORT
```

3. Ensure that conf_usb.h contains the following parameters required for a USB device high speed (480Mbit/s):

```
//#define USB_DEVICE_LOW_SPEED
#define  USB_DEVICE_HS_SUPPORT
```

3.7. Use USB Strings

In this use case, the usual USB strings are added in the USB device.

3.7.1. Setup Steps

Prior to implement this use case, be sure to have already applied the UDI module "basic use case".

3.7.2. Usage Steps

3.7.2.1. Example Code

Content of conf_usb.h:

```
#define  USB_DEVICE_MANUFACTURE_NAME      "Manufacture name"
#define  USB_DEVICE_PRODUCT_NAME          "Product name"
#define  USB_DEVICE_SERIAL_NAME           "12...EF"
```

3.7.2.2. Workflow

1. Ensure that conf_usb.h is available and contains the following parameters required to enable different USB strings:

```
// Static ASCII name for the manufacture
#define  USB_DEVICE_MANUFACTURE_NAME "Manufacture name"
```

```
// Static ASCII name for the product
#define  USB_DEVICE_PRODUCT_NAME "Product name"
```

```
// Static ASCII name to enable and set a serial number
#define  USB_DEVICE_SERIAL_NAME "12...EF"
```

3.8. Use USB Remote Wakeup Feature

In this use case, the USB remote wakeup feature is enabled.

3.8.1. Setup Steps

Prior to implement this use case, be sure to have already applied the UDI module "basic use case".

3.8.2. Usage Steps

3.8.2.1. Example Code

Content of conf_usb.h:

```
#define  USB_DEVICE_ATTR \
(USB_CONFIG_ATTR_REMOTE_WAKEUP | USB_CONFIG_ATTR_..._POWERED)
#define UDC_REMOTEWAKEUP_ENABLE() my_callback_remotewakeup_enable()
extern void my_callback_remotewakeup_enable(void);
#define UDC_REMOTEWAKEUP_DISABLE() my_callback_remotewakeup_disable()
extern void my_callback_remotewakeup_disable(void);
```

Add to application C-file:

```
void my_callback_remotewakeup_enable(void)
{
// Enable application wakeup events (e.g. enable GPIO interrupt)
}
void my_callback_remotewakeup_disable(void)
{
// Disable application wakeup events (e.g. disable GPIO interrupt)
}

void my_interrupt_event(void)
{
   udc_remotewakeup();
}
```

3.8.2.2. Workflow

1. Ensure that conf_usb.h is available and contains the following parameters required to enable the remote wakeup feature:

   ```
   // Authorizes the remote wakeup feature
   #define  USB_DEVICE_ATTR (USB_CONFIG_ATTR_REMOTE_WAKEUP |
   USB_CONFIG_ATTR_..._POWERED)
   ```

   ```
   // Define callback called when the host enables the remotewakeup feature
   #define UDC_REMOTEWAKEUP_ENABLE() my_callback_remotewakeup_enable()
   extern void my_callback_remotewakeup_enable(void);
   ```

   ```
   // Define callback called when the host disables the remotewakeup feature
   #define UDC_REMOTEWAKEUP_DISABLE() my_callback_remotewakeup_disable()
   extern void my_callback_remotewakeup_disable(void);
   ```

2. Send a remote wakeup (USB upstream):

   ```
   udc_remotewakeup();
   ```

3.9. Bus Power Application Recommendations

In this use case, the USB device bus power feature is enabled. This feature requires a correct power consumption management.

3.9.1. Setup Steps

Prior to implement this use case, be sure to have already applied the UDI module "basic use case".

3.9.2. Usage Steps

3.9.2.1. Example Code

Content of conf_usb.h:

```
#define  USB_DEVICE_ATTR (USB_CONFIG_ATTR_BUS_POWERED)
#define  UDC_SUSPEND_EVENT()         user_callback_suspend_action()
extern void user_callback_suspend_action(void)
#define  UDC_RESUME_EVENT()          user_callback_resume_action()
extern void user_callback_resume_action(void)
```

Add to application C-file:

```
void user_callback_suspend_action(void)
{
   // Disable hardware component to reduce power consumption
}
void user_callback_resume_action(void)
{
   // Re-enable hardware component
}
```

3.9.2.2. Workflow

1. Ensure that conf_usb.h is available and contains the following parameters:

```
// Authorizes the BUS power feature
#define  USB_DEVICE_ATTR (USB_CONFIG_ATTR_BUS_POWERED)
```

```
// Define callback called when the host suspend the USB line
#define UDC_SUSPEND_EVENT() user_callback_suspend_action()
extern void user_callback_suspend_action(void);
```

```
// Define callback called when the host or device resume the USB line
#define UDC_RESUME_EVENT() user_callback_resume_action()
extern void user_callback_resume_action(void);
```

2. Reduce power consumption in suspend mode (max. 2.5mA on VBUS):

```
void user_callback_suspend_action(void)
{
   turn_off_components();
}
```

3.10. USB Dynamic Serial Number

In this use case, the USB serial strings are dynamic. For a static serial string refer to Use USB Strings.

3.10.1. Setup Steps

Prior to implement this use case, be sure to have already applied the UDI module "basic use case".

3.10.2. Usage Steps

3.10.2.1. Example Code

Content of conf_usb.h:

```
#define  USB_DEVICE_SERIAL_NAME
#define  USB_DEVICE_GET_SERIAL_NAME_POINTER serial_number
#define  USB_DEVICE_GET_SERIAL_NAME_LENGTH  12
extern uint8_t serial_number[];
```

Add to application C-file:

```
uint8_t serial_number[USB_DEVICE_GET_SERIAL_NAME_LENGTH];
void init_build_usb_serial_number(void)
{
serial_number[0] = 'A';
serial_number[1] = 'B';
...
serial_number[USB_DEVICE_GET_SERIAL_NAME_LENGTH-1] = 'C';
}
```

3.10.2.2. Workflow

1. Ensure that conf_usb.h is available and contains the following parameters required to enable a USB serial number string dynamically:

```
#define  USB_DEVICE_SERIAL_NAME // Define this empty
#define  USB_DEVICE_GET_SERIAL_NAME_POINTER serial_number // Give
serial array pointer
#define  USB_DEVICE_GET_SERIAL_NAME_LENGTH  12 // Give size of serial
array
extern uint8_t serial_number[]; // Declare external serial array
```

2. Before starting USB stack, initialize the serial array:

```
uint8_t serial_number[USB_DEVICE_GET_SERIAL_NAME_LENGTH];
void init_build_usb_serial_number(void)
{
serial_number[0] = 'A';
serial_number[1] = 'B';
...
serial_number[USB_DEVICE_GET_SERIAL_NAME_LENGTH-1] = 'C';
}
```

4. Configuration File Examples

4.1. conf_usb.h

4.1.1. UDI MSC Single

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */

#ifndef _CONF_USB_H_
#define _CONF_USB_H_

#include "compiler.h"

#warning You must refill the following definitions with a correct values

#define  USB_DEVICE_VENDOR_ID             USB_VID_ATMEL
#define  USB_DEVICE_PRODUCT_ID            USB_PID_ATMEL_ASF_MSC
#define  USB_DEVICE_MAJOR_VERSION         1
#define  USB_DEVICE_MINOR_VERSION         0
#define  USB_DEVICE_POWER                 100 // Consumption on VBUS line
(mA)
#define  USB_DEVICE_ATTR                  \
    (USB_CONFIG_ATTR_SELF_POWERED)
// (USB_CONFIG_ATTR_BUS_POWERED)
//  (USB_CONFIG_ATTR_REMOTE_WAKEUP|USB_CONFIG_ATTR_SELF_POWERED)
//  (USB_CONFIG_ATTR_REMOTE_WAKEUP|USB_CONFIG_ATTR_BUS_POWERED)

// #define  USB_DEVICE_MANUFACTURE_NAME      "Manufacture name"
// #define  USB_DEVICE_PRODUCT_NAME          "Product name"
#define  USB_DEVICE_SERIAL_NAME            "12...EF" // Disk SN for MSC

#if (UC3A3||UC3A4)
//#define  USB_DEVICE_HS_SUPPORT
#endif



// #define  UDC_VBUS_EVENT(b_vbus_high)      user_callback_vbus_action(b_vbus_high)
// extern void user_callback_vbus_action(bool b_vbus_high);
// #define  UDC_SOF_EVENT()                  user_callback_sof_action()
// extern void user_callback_sof_action(void);
// #define  UDC_SUSPEND_EVENT()              user_callback_suspend_action()
// extern void user_callback_suspend_action(void);
// #define  UDC_RESUME_EVENT()               user_callback_resume_action()
// extern void user_callback_resume_action(void);
// #define  UDC_REMOTEWAKEUP_ENABLE()        user_callback_remotewakeup_enable()
// extern void user_callback_remotewakeup_enable(void);
// #define  UDC_REMOTEWAKEUP_DISABLE()       user_callback_remotewakeup_disable()
// extern void user_callback_remotewakeup_disable(void);
// #define  UDC_GET_EXTRA_STRING()



#define UDI_MSC_GLOBAL_VENDOR_ID            \
   'A', 'T', 'M', 'E', 'L', ' ', ' ', ' '
```

```
#define UDI_MSC_GLOBAL_PRODUCT_VERSION            \
   '1', '.', '0', '0'

#define  UDI_MSC_ENABLE_EXT()          true
#define  UDI_MSC_DISABLE_EXT()
#define  UDI_MSC_NOTIFY_TRANS_EXT()
/*
 * #define UDI_MSC_ENABLE_EXT() my_callback_msc_enable()
 * extern bool my_callback_msc_enable(void);
 * #define UDI_MSC_DISABLE_EXT() my_callback_msc_disable()
 * extern void my_callback_msc_disable(void);
 * #define  UDI_MSC_NOTIFY_TRANS_EXT()    msc_notify_trans()
 * extern void msc_notify_trans(void) {
 */



#include "udi_msc_conf.h"

#endif // _CONF_USB_H_
```

4.1.2. UDI MSC Multiple (Composite)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */

#ifndef _CONF_USB_H_
#define _CONF_USB_H_

#include "compiler.h"

#warning You must refill the following definitions with a correct values

#define  USB_DEVICE_VENDOR_ID             USB_VID_ATMEL
#define  USB_DEVICE_PRODUCT_ID            0xFFFF
#define  USB_DEVICE_MAJOR_VERSION         1
#define  USB_DEVICE_MINOR_VERSION         0
#define  USB_DEVICE_POWER                 100 // Consumption on VBUS line
(mA)
#define  USB_DEVICE_ATTR                  \
        (USB_CONFIG_ATTR_SELF_POWERED)
// (USB_CONFIG_ATTR_BUS_POWERED)
// (USB_CONFIG_ATTR_REMOTE_WAKEUP|USB_CONFIG_ATTR_SELF_POWERED)
// (USB_CONFIG_ATTR_REMOTE_WAKEUP|USB_CONFIG_ATTR_BUS_POWERED)

// #define  USB_DEVICE_MANUFACTURE_NAME      "Manufacture name"
// #define  USB_DEVICE_PRODUCT_NAME          "Product name"
// #define  USB_DEVICE_SERIAL_NAME           "12...EF" // Disk SN for MSC

//#define  USB_DEVICE_LOW_SPEED

#if (UC3A3||UC3A4)
//#define  USB_DEVICE_HS_SUPPORT
#endif


// #define  UDC_VBUS_EVENT(b_vbus_high)      user_callback_vbus_action(b_vbus_high)
// extern void user_callback_vbus_action(bool b_vbus_high);
// #define  UDC_SOF_EVENT()                  user_callback_sof_action()
// extern void user_callback_sof_action(void);
```

```
// #define  UDC_SUSPEND_EVENT()              user_callback_suspend_action()
// extern void user_callback_suspend_action(void);
// #define  UDC_RESUME_EVENT()               user_callback_resume_action()
// extern void user_callback_resume_action(void);
// #define  UDC_REMOTEWAKEUP_ENABLE()        user_callback_remotewakeup_enable()
// extern void user_callback_remotewakeup_enable(void);
// #define  UDC_REMOTEWAKEUP_DISABLE()       user_callback_remotewakeup_disable()
// extern void user_callback_remotewakeup_disable(void);
// #define  UDC_GET_EXTRA_STRING()

#define  USB_DEVICE_EP_CTRL_SIZE       64

#define  USB_DEVICE_NB_INTERFACE       1 // 1 or more

#define  USB_DEVICE_MAX_EP             1 // 0 to max endpoint requested by
interfaces



#define  UDI_CDC_PORT_NB 1

#define  UDI_CDC_ENABLE_EXT(port)             true
#define  UDI_CDC_DISABLE_EXT(port)
#define  UDI_CDC_RX_NOTIFY(port)
#define  UDI_CDC_TX_EMPTY_NOTIFY(port)
#define  UDI_CDC_SET_CODING_EXT(port,cfg)
#define  UDI_CDC_SET_DTR_EXT(port,set)
#define  UDI_CDC_SET_RTS_EXT(port,set)
/*
 * #define UDI_CDC_ENABLE_EXT(port) my_callback_cdc_enable()
 * extern bool my_callback_cdc_enable(void);
 * #define UDI_CDC_DISABLE_EXT(port) my_callback_cdc_disable()
 * extern void my_callback_cdc_disable(void);
 * #define  UDI_CDC_RX_NOTIFY(port) my_callback_rx_notify(port)
 * extern void my_callback_rx_notify(uint8_t port);
 * #define  UDI_CDC_TX_EMPTY_NOTIFY(port) my_callback_tx_empty_notify(port)
 * extern void my_callback_tx_empty_notify(uint8_t port);
 * #define  UDI_CDC_SET_CODING_EXT(port,cfg) my_callback_config(port,cfg)
 * extern void my_callback_config(uint8_t port, usb_cdc_line_coding_t * cfg);
 * #define  UDI_CDC_SET_DTR_EXT(port,set) my_callback_cdc_set_dtr(port,set)
 * extern void my_callback_cdc_set_dtr(uint8_t port, bool b_enable);
 * #define  UDI_CDC_SET_RTS_EXT(port,set) my_callback_cdc_set_rts(port,set)
 * extern void my_callback_cdc_set_rts(uint8_t port, bool b_enable);
 */

#define  UDI_CDC_LOW_RATE

#define  UDI_CDC_DEFAULT_RATE             115200
#define  UDI_CDC_DEFAULT_STOPBITS         CDC_STOP_BITS_1
#define  UDI_CDC_DEFAULT_PARITY           CDC_PAR_NONE
#define  UDI_CDC_DEFAULT_DATABITS         8

#define  UDI_CDC_DATA_EP_IN_0          (1 | USB_EP_DIR_IN)  // TX
#define  UDI_CDC_DATA_EP_OUT_0         (2 | USB_EP_DIR_OUT) // RX
#define  UDI_CDC_COMM_EP_0             (3 | USB_EP_DIR_IN)  // Notify
endpoint
#define  UDI_CDC_DATA_EP_IN_2          (4 | USB_EP_DIR_IN)  // TX
#define  UDI_CDC_DATA_EP_OUT_2         (5 | USB_EP_DIR_OUT) // RX
#define  UDI_CDC_COMM_EP_2             (6 | USB_EP_DIR_IN)  // Notify
endpoint
#define  UDI_CDC_DATA_EP_IN_3          (7 | USB_EP_DIR_IN)  // TX
#define  UDI_CDC_DATA_EP_OUT_3         (8 | USB_EP_DIR_OUT) // RX
#define  UDI_CDC_COMM_EP_3             (9 | USB_EP_DIR_IN)  // Notify
```

```
endpoint

#define  UDI_CDC_COMM_IFACE_NUMBER_0   0
#define  UDI_CDC_DATA_IFACE_NUMBER_0   1
#define  UDI_CDC_COMM_IFACE_NUMBER_2   2
#define  UDI_CDC_DATA_IFACE_NUMBER_2   3
#define  UDI_CDC_COMM_IFACE_NUMBER_3   4
#define  UDI_CDC_DATA_IFACE_NUMBER_3   5



#define UDI_MSC_GLOBAL_VENDOR_ID            \
   'A', 'T', 'M', 'E', 'L', ' ', ' ', ' '
#define UDI_MSC_GLOBAL_PRODUCT_VERSION           \
   '1', '.', '0', '0'

#define  UDI_MSC_ENABLE_EXT()          true
#define  UDI_MSC_DISABLE_EXT()
#define  UDI_MSC_NOTIFY_TRANS_EXT()
/*
 * #define UDI_MSC_ENABLE_EXT() my_callback_msc_enable()
 * extern bool my_callback_msc_enable(void);
 * #define UDI_MSC_DISABLE_EXT() my_callback_msc_disable()
 * extern void my_callback_msc_disable(void);
 * #define  UDI_MSC_NOTIFY_TRANS_EXT()    msc_notify_trans()
 * extern void msc_notify_trans(void) {
 */

#define  UDI_MSC_EP_IN                 (1 | USB_EP_DIR_IN)
#define  UDI_MSC_EP_OUT                (2 | USB_EP_DIR_OUT)

#define  UDI_MSC_IFACE_NUMBER          0



#define  UDI_HID_MOUSE_ENABLE_EXT()        true
#define  UDI_HID_MOUSE_DISABLE_EXT()
// #define UDI_HID_MOUSE_ENABLE_EXT() my_callback_mouse_enable()
// extern bool my_callback_mouse_enable(void);
// #define UDI_HID_MOUSE_DISABLE_EXT() my_callback_mouse_disable()
// extern void my_callback_mouse_disable(void);

#define  UDI_HID_MOUSE_EP_IN           (1 | USB_EP_DIR_IN)

#define  UDI_HID_MOUSE_IFACE_NUMBER    0



#define  UDI_HID_KBD_ENABLE_EXT()        true
#define  UDI_HID_KBD_DISABLE_EXT()
// #define UDI_HID_KBD_ENABLE_EXT() my_callback_keyboard_enable()
// extern bool my_callback_keyboard_enable(void);
// #define UDI_HID_KBD_DISABLE_EXT() my_callback_keyboard_disable()
// extern void my_callback_keyboard_disable(void);
#define  UDI_HID_KBD_CHANGE_LED(value)
// #define  UDI_HID_KBD_CHANGE_LED(value) my_callback_keyboard_led(value)
// extern void my_callback_keyboard_led(uint8_t value)

#define  UDI_HID_KBD_EP_IN           (1 | USB_EP_DIR_IN)
```

```
#define  UDI_HID_KBD_IFACE_NUMBER    0



#define  UDI_HID_GENERIC_ENABLE_EXT()       true
#define  UDI_HID_GENERIC_DISABLE_EXT()
#define  UDI_HID_GENERIC_REPORT_OUT(ptr)
#define  UDI_HID_GENERIC_SET_FEATURE(f)
/*
 * #define UDI_HID_GENERIC_ENABLE_EXT() my_callback_generic_enable()
 * extern bool my_callback_generic_enable(void);
 * #define UDI_HID_GENERIC_DISABLE_EXT() my_callback_generic_disable()
 * extern void my_callback_generic_disable(void);
 * #define  UDI_HID_GENERIC_REPORT_OUT(ptr) my_callback_generic_report_out(ptr)
 * extern void my_callback_generic_report_out(uint8_t *report);
 * #define  UDI_HID_GENERIC_SET_FEATURE(f) my_callback_generic_set_feature(f)
 * extern void my_callback_generic_set_feature(uint8_t *report_feature);
 */
#define  UDI_HID_REPORT_IN_SIZE             64
#define  UDI_HID_REPORT_OUT_SIZE            64
#define  UDI_HID_REPORT_FEATURE_SIZE        4
#define  UDI_HID_GENERIC_EP_SIZE            64

#define  UDI_HID_GENERIC_EP_OUT   (2 | USB_EP_DIR_OUT)
#define  UDI_HID_GENERIC_EP_IN    (1 | USB_EP_DIR_IN)

#define  UDI_HID_GENERIC_IFACE_NUMBER    0



#define  UDI_PHDC_ENABLE_EXT()          true
#define  UDI_PHDC_DISABLE_EXT()

#define  UDI_PHDC_DATAMSG_FORMAT       USB_PHDC_DATAMSG_FORMAT_11073_20601
#define  UDI_PHDC_SPECIALIZATION       {0x2345} // Define in 11073_20601

#define  UDI_PHDC_QOS_OUT              \
        (USB_PHDC_QOS_MEDIUM_BETTER|USB_PHDC_QOS_HIGH_BEST)
#define  UDI_PHDC_QOS_IN               \
        (USB_PHDC_QOS_LOW_GOOD|USB_PHDC_QOS_MEDIUM_BETTER|
USB_PHDC_QOS_MEDIUM_BEST)

#define  UDI_PHDC_METADATA_DESC_BULK_IN   {0x01,0x02,0x03}
#define  UDI_PHDC_METADATA_DESC_BULK_OUT  {0x01,0x02,0x03}
#define  UDI_PHDC_METADATA_DESC_INT_IN    {0x01,0x02,0x03}

#define  UDI_PHDC_EP_BULK_OUT          (1 | USB_EP_DIR_OUT)
#define  UDI_PHDC_EP_BULK_IN           (2 | USB_EP_DIR_IN)
#if ((UDI_PHDC_QOS_IN&USB_PHDC_QOS_LOW_GOOD)==USB_PHDC_QOS_LOW_GOOD)
// Only if UDI_PHDC_QOS_IN include USB_PHDC_QOS_LOW_GOOD
#  define  UDI_PHDC_EP_INTERRUPT_IN      (3 | USB_EP_DIR_IN)
#endif

#define  UDI_PHDC_EP_SIZE_BULK_OUT     32
#define  UDI_PHDC_EP_SIZE_BULK_IN      32
#define  UDI_PHDC_EP_SIZE_INT_IN       8


#define  UDI_PHDC_IFACE_NUMBER          0
```

```
#define  UDI_VENDOR_ENABLE_EXT()           true
#define  UDI_VENDOR_DISABLE_EXT()
#define  UDI_VENDOR_SETUP_OUT_RECEIVED()   false
#define  UDI_VENDOR_SETUP_IN_RECEIVED()    false
/*
 * #define UDI_VENDOR_ENABLE_EXT() my_callback_vendor_enable()
 * extern bool my_callback_vendor_enable(void);
 * #define UDI_VENDOR_DISABLE_EXT() my_callback_vendor_disable()
 * extern void my_callback_vendor_disable(void);
 *
 * #define  UDI_VENDOR_SETUP_OUT_RECEIVED()  my_vendor_setup_out_received()
 * extern bool my_vendor_setup_out_received(void);
 * #define  UDI_VENDOR_SETUP_IN_RECEIVED()   my_vendor_setup_in_received()
 * extern bool my_vendor_setup_in_received(void);
 */

#define  UDI_VENDOR_EPS_SIZE_INT_FS    64
#define  UDI_VENDOR_EPS_SIZE_BULK_FS   64
#define  UDI_VENDOR_EPS_SIZE_ISO_FS   256

#define  UDI_VENDOR_EPS_SIZE_INT_HS    64
#define  UDI_VENDOR_EPS_SIZE_BULK_HS  512
#define  UDI_VENDOR_EPS_SIZE_ISO_HS    64

#define  UDI_VENDOR_EP_INTERRUPT_IN  (1 | USB_EP_DIR_IN)
#define  UDI_VENDOR_EP_INTERRUPT_OUT (2 | USB_EP_DIR_OUT)
#define  UDI_VENDOR_EP_BULK_IN       (3 | USB_EP_DIR_IN)
#define  UDI_VENDOR_EP_BULK_OUT      (4 | USB_EP_DIR_OUT)
#define  UDI_VENDOR_EP_ISO_IN        (5 | USB_EP_DIR_IN)
#define  UDI_VENDOR_EP_ISO_OUT       (6 | USB_EP_DIR_OUT)


#define  UDI_VENDOR_IFACE_NUMBER     0



//... Eventually add other Interface Configuration



#define UDI_COMPOSITE_DESC_T

#define UDI_COMPOSITE_DESC_FS

#define UDI_COMPOSITE_DESC_HS

#define UDI_COMPOSITE_API

/* Example for device with cdc, msc and hid mouse interface
#define UDI_COMPOSITE_DESC_T \
    usb_iad_desc_t udi_cdc_iad; \
    udi_cdc_comm_desc_t udi_cdc_comm; \
    udi_cdc_data_desc_t udi_cdc_data; \
    udi_msc_desc_t udi_msc; \
    udi_hid_mouse_desc_t udi_hid_mouse

#define UDI_COMPOSITE_DESC_FS \
    .udi_cdc_iad               = UDI_CDC_IAD_DESC_0, \
    .udi_cdc_comm              = UDI_CDC_COMM_DESC_0, \
    .udi_cdc_data              = UDI_CDC_DATA_DESC_0_FS, \
    .udi_msc                   = UDI_MSC_DESC_FS, \
    .udi_hid_mouse             = UDI_HID_MOUSE_DESC

#define UDI_COMPOSITE_DESC_HS \
    .udi_cdc_iad               = UDI_CDC_IAD_DESC_0, \
```

```
    .udi_cdc_comm               = UDI_CDC_COMM_DESC_0, \
    .udi_cdc_data               = UDI_CDC_DATA_DESC_0_HS, \
    .udi_msc                    = UDI_MSC_DESC_HS, \
    .udi_hid_mouse              = UDI_HID_MOUSE_DESC

#define UDI_COMPOSITE_API  \
    &udi_api_cdc_comm,       \
    &udi_api_cdc_data,       \
    &udi_api_msc,            \
    &udi_api_hid_mouse
*/




/* Example of include for interface
#include "udi_msc.h"
#include "udi_hid_kbd.h"
#include "udi_hid_mouse.h"
#include "udi_cdc.h"
#include "udi_phdc.h"
#include "udi_vendor.h"
*/
/* Declaration of callbacks used by USB
#include "callback_def.h"
*/

#endif // _CONF_USB_H_
```

4.2. conf_clock.h

4.2.1. XMEGA (USB)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */
#ifndef CONF_CLOCK_H_INCLUDED
#define CONF_CLOCK_H_INCLUDED

#define CONFIG_USBCLK_SOURCE     USBCLK_SRC_RCOSC
#define CONFIG_OSC_RC32_CAL      48000000UL

#define CONFIG_OSC_AUTOCAL_RC32MHZ_REF_OSC  OSC_ID_USBSOF

#define CONFIG_SYSCLK_SOURCE     SYSCLK_SRC_RC32MHZ
#define CONFIG_SYSCLK_PSADIV     SYSCLK_PSADIV_2
#define CONFIG_SYSCLK_PSBCDIV    SYSCLK_PSBCDIV_1_1

/*

#define CONFIG_PLL0_SOURCE       PLL_SRC_XOSC
#define CONFIG_PLL0_MUL          6
#define CONFIG_PLL0_DIV          1

#define CONFIG_USBCLK_SOURCE     USBCLK_SRC_PLL

#define CONFIG_SYSCLK_SOURCE     SYSCLK_SRC_PLL
#define CONFIG_SYSCLK_PSADIV     SYSCLK_PSADIV_2
#define CONFIG_SYSCLK_PSBCDIV    SYSCLK_PSBCDIV_1_1
*/
```

```
#endif /* CONF_CLOCK_H_INCLUDED */
```

4.2.2. AT32UC3A0, AT32UC3A1, and AT32UC3B Devices (USBB)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */
#ifndef CONF_CLOCK_H_INCLUDED
#define CONF_CLOCK_H_INCLUDED

// ===== System Clock Source Options
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_RCSYS
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_OSC0
#define CONFIG_SYSCLK_SOURCE          SYSCLK_SRC_PLL0

// ===== PLL0 Options
#define CONFIG_PLL0_SOURCE            PLL_SRC_OSC0
//#define CONFIG_PLL0_SOURCE          PLL_SRC_OSC1
#define CONFIG_PLL0_MUL               8 /* Fpll = (Fclk * PLL_mul) / PLL_div */
#define CONFIG_PLL0_DIV               2 /* Fpll = (Fclk * PLL_mul) / PLL_div */

// ===== PLL1 Options
//#define CONFIG_PLL1_SOURCE          PLL_SRC_OSC0
//#define CONFIG_PLL1_SOURCE          PLL_SRC_OSC1
//#define CONFIG_PLL1_MUL             8 /* Fpll = (Fclk * PLL_mul) / PLL_div */
//#define CONFIG_PLL1_DIV             2 /* Fpll = (Fclk * PLL_mul) / PLL_div */

// ===== System Clock Bus Division Options
//#define CONFIG_SYSCLK_CPU_DIV         0 /* Fcpu = Fsys/(2 ^ CPU_div) */
//#define CONFIG_SYSCLK_PBA_DIV         0 /* Fpba = Fsys/(2 ^ PBA_div) */
//#define CONFIG_SYSCLK_PBB_DIV         0 /* Fpbb = Fsys/(2 ^ PBB_div) */

// ===== Peripheral Clock Management Options
//#define CONFIG_SYSCLK_INIT_CPUMASK  ((1 << SYSCLK_SYSTIMER) | (1 << SYSCLK_OCD))
//#define CONFIG_SYSCLK_INIT_PBAMASK  (1 << SYSCLK_USART0)
//#define CONFIG_SYSCLK_INIT_PBBMASK  (1 << SYSCLK_HMATRIX)
//#define CONFIG_SYSCLK_INIT_HSBMASK  (1 << SYSCLK_MDMA_HSB)

// ===== USB Clock Source Options
//#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_OSC0
#define CONFIG_USBCLK_SOURCE          USBCLK_SRC_PLL0
//#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_PLL1
#define CONFIG_USBCLK_DIV             1 /* Fusb = Fsys/(2 ^ USB_div) */

#endif /* CONF_CLOCK_H_INCLUDED */
```

4.2.3. AT32UC3A3 and AT32UC3A4 Devices (USBB with High Speed Support)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */
#ifndef CONF_CLOCK_H_INCLUDED
#define CONF_CLOCK_H_INCLUDED

// ===== System Clock Source Options
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_RCSYS
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_OSC0
#define CONFIG_SYSCLK_SOURCE          SYSCLK_SRC_PLL0
```

```
// ===== PLL0 Options
#define CONFIG_PLL0_SOURCE            PLL_SRC_OSC0
//#define CONFIG_PLL0_SOURCE          PLL_SRC_OSC1
#define CONFIG_PLL0_MUL               11 /* Fpll = (Fclk * PLL_mul) / PLL_div
*/
#define CONFIG_PLL0_DIV               2  /* Fpll = (Fclk * PLL_mul) / PLL_div
*/

// ===== PLL1 Options
//#define CONFIG_PLL1_SOURCE          PLL_SRC_OSC0
//#define CONFIG_PLL1_SOURCE          PLL_SRC_OSC1
//#define CONFIG_PLL1_MUL             8 /* Fpll = (Fclk * PLL_mul) / PLL_div */
//#define CONFIG_PLL1_DIV             2 /* Fpll = (Fclk * PLL_mul) / PLL_div */

// ===== System Clock Bus Division Options
#define CONFIG_SYSCLK_CPU_DIV         0 /* Fcpu = Fsys/(2 ^ CPU_div) */
#define CONFIG_SYSCLK_PBA_DIV         0 /* Fpba = Fsys/(2 ^ PBA_div) */
//#define CONFIG_SYSCLK_PBB_DIV       0 /* Fpbb = Fsys/(2 ^ PBB_div) */

// ===== Peripheral Clock Management Options
//#define CONFIG_SYSCLK_INIT_CPUMASK  ((1 << SYSCLK_SYSTIMER) | (1 << SYSCLK_OCD))
//#define CONFIG_SYSCLK_INIT_PBAMASK  (1 << SYSCLK_USART0)
//#define CONFIG_SYSCLK_INIT_PBBMASK  (1 << SYSCLK_HMATRIX)
//#define CONFIG_SYSCLK_INIT_HSBMASK  (1 << SYSCLK_MDMA_HSB)

// ===== USB Clock Source Options
#define CONFIG_USBCLK_SOURCE          USBCLK_SRC_OSC0
//#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_PLL0
//#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_PLL1
#define CONFIG_USBCLK_DIV             1 /* Fusb = Fsys/(2 ^ USB_div) */

#endif /* CONF_CLOCK_H_INCLUDED */
```

4.2.4. AT32UC3C, ATUCXXD, ATUCXXL3U, and ATUCXXL4U Devices (USBC)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */
#ifndef CONF_CLOCK_H_INCLUDED
#define CONF_CLOCK_H_INCLUDED

// ===== System Clock Source Options
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_RCSYS
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_OSC0
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_OSC1
#define CONFIG_SYSCLK_SOURCE          SYSCLK_SRC_PLL1
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_PLL1
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_RC8M

// ===== PLL0 Options
#define CONFIG_PLL0_SOURCE            PLL_SRC_OSC0
//#define CONFIG_PLL0_SOURCE          PLL_SRC_OSC1
//#define CONFIG_PLL0_SOURCE          PLL_SRC_RC8M
#define CONFIG_PLL0_MUL               3 /* Fpll = (Fclk * PLL_mul) / PLL_div */
#define CONFIG_PLL0_DIV               1 /* Fpll = (Fclk * PLL_mul) / PLL_div */

// ===== PLL1 Options
#define CONFIG_PLL1_SOURCE            PLL_SRC_OSC0
//#define CONFIG_PLL1_SOURCE          PLL_SRC_OSC1
//#define CONFIG_PLL1_SOURCE          PLL_SRC_RC8M
```

```
#define CONFIG_PLL1_MUL                 3 /* Fpll = (Fclk * PLL_mul) / PLL_div */
#define CONFIG_PLL1_DIV                 1 /* Fpll = (Fclk * PLL_mul) / PLL_div */

// ===== System Clock Bus Division Options
//#define CONFIG_SYSCLK_CPU_DIV       0 /* Fcpu = Fsys/(2 ^ CPU_div) */
//#define CONFIG_SYSCLK_PBA_DIV       0 /* Fpba = Fsys/(2 ^ PBA_div) */
//#define CONFIG_SYSCLK_PBB_DIV       0 /* Fpbb = Fsys/(2 ^ PBB_div) */
//#define CONFIG_SYSCLK_PBC_DIV       0 /* Fpbc = Fsys/(2 ^ PBC_div) */

// ===== Peripheral Clock Management Options
//#define CONFIG_SYSCLK_INIT_CPUMASK  ((1 << SYSCLK_SYSTIMER) | (1 << SYSCLK_OCD))
//#define CONFIG_SYSCLK_INIT_PBAMASK  (1 << SYSCLK_USART0)
//#define CONFIG_SYSCLK_INIT_PBBMASK  (1 << SYSCLK_HMATRIX)
//#define CONFIG_SYSCLK_INIT_HSBMASK  (1 << SYSCLK_MDMA_HSB)

// ===== USB Clock Source Options
//#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_OSC0
//#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_OSC1
#define CONFIG_USBCLK_SOURCE          USBCLK_SRC_PLL0
//#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_PLL1
#define CONFIG_USBCLK_DIV             1 /* Fusb = Fsys/(2 ^ USB_div) */

#endif /* CONF_CLOCK_H_INCLUDED */
```

4.2.5. SAM3S, SAM3SD, and SAM4S Devices (UPD: USB Peripheral Device)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */

#ifndef CONF_CLOCK_H_INCLUDED
#define CONF_CLOCK_H_INCLUDED

// ===== System Clock (MCK) Source Options
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_SLCK_RC
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_SLCK_XTAL
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_SLCK_BYPASS
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_4M_RC
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_8M_RC
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_12M_RC
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_XTAL
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_BYPASS
#define CONFIG_SYSCLK_SOURCE          SYSCLK_SRC_PLLACK
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_PLLBCK

// ===== System Clock (MCK) Prescaler Options   (Fmck = Fsys / (SYSCLK_PRES))
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_1
#define CONFIG_SYSCLK_PRES            SYSCLK_PRES_2
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_4
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_8
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_16
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_32
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_64
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_3

// ===== PLL0 (A) Options   (Fpll = (Fclk * PLL_mul) / PLL_div)
// Use mul and div effective values here.
#define CONFIG_PLL0_SOURCE            PLL_SRC_MAINCK_XTAL
#define CONFIG_PLL0_MUL               32
#define CONFIG_PLL0_DIV               3
```

```
// ===== PLL1 (B) Options   (Fpll = (Fclk * PLL_mul) / PLL_div)
// Use mul and div effective values here.
#define CONFIG_PLL1_SOURCE          PLL_SRC_MAINCK_XTAL
#define CONFIG_PLL1_MUL             16
#define CONFIG_PLL1_DIV             2

// ===== USB Clock Source Options   (Fusb = FpllX / USB_div)
// Use div effective value here.
//#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_PLL0
#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_PLL1
#define CONFIG_USBCLK_DIV           2

// ===== Target frequency (System clock)
// - XTAL frequency: 12MHz
// - System clock source: PLLA
// - System clock prescaler: 2 (divided by 2)
// - PLLA source: XTAL
// - PLLA output: XTAL * 32 / 3
// - System clock is: 12 * 32 / 3 / 2 = 64MHz
// ===== Target frequency (USB Clock)
// - USB clock source: PLLB
// - USB clock divider: 2 (divided by 2)
// - PLLB output: XTAL * 16 / 2
// - USB clock: 12 * 16 / 2 / 2 = 48MHz


#endif /* CONF_CLOCK_H_INCLUDED */
```

4.2.6. SAM3U Device (UPDHS: USB Peripheral Device High Speed)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */

#ifndef CONF_CLOCK_H_INCLUDED
#define CONF_CLOCK_H_INCLUDED

// ===== System Clock (MCK) Source Options
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_SLCK_RC
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_SLCK_XTAL
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_SLCK_BYPASS
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_4M_RC
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_8M_RC
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_12M_RC
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_XTAL
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_BYPASS
#define CONFIG_SYSCLK_SOURCE          SYSCLK_SRC_PLLACK
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_UPLLCK

// ===== System Clock (MCK) Prescaler Options   (Fmck = Fsys / (SYSCLK_PRES))
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_1
#define CONFIG_SYSCLK_PRES            SYSCLK_PRES_2
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_4
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_8
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_16
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_32
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_64
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_3

// ===== PLL0 (A) Options   (Fpll = (Fclk * PLL_mul) / PLL_div)
// Use mul and div effective values here.
```

```
#define CONFIG_PLL0_SOURCE          PLL_SRC_MAINCK_XTAL
#define CONFIG_PLL0_MUL             16
#define CONFIG_PLL0_DIV             1

// ===== UPLL (UTMI) Hardware fixed at 480MHz.

// ===== USB Clock Source fixed at UPLL.

// ===== Target frequency (System clock)
// - XTAL frequency: 12MHz
// - System clock source: PLLA
// - System clock prescaler: 2 (divided by 2)
// - PLLA source: XTAL
// - PLLA output: XTAL * 16 / 1
// - System clock is: 12 * 16 / 1 / 2 = 96MHz
// ===== Target frequency (USB Clock)
// - USB clock source: UPLL
// - UPLL frequency: 480MHz
// - USB clock: 480MHz


#endif /* CONF_CLOCK_H_INCLUDED */
```

4.2.7. SAM3X and SAM3A Devices (UOTGHS: USB OTG High Speed)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */

#ifndef CONF_CLOCK_H_INCLUDED
#define CONF_CLOCK_H_INCLUDED

// ===== System Clock (MCK) Source Options
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_SLCK_RC
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_SLCK_XTAL
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_SLCK_BYPASS
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_4M_RC
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_8M_RC
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_12M_RC
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_XTAL
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_MAINCK_BYPASS
#define CONFIG_SYSCLK_SOURCE          SYSCLK_SRC_PLLACK
//#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_UPLLCK

// ===== System Clock (MCK) Prescaler Options   (Fmck = Fsys / (SYSCLK_PRES))
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_1
#define CONFIG_SYSCLK_PRES            SYSCLK_PRES_2
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_4
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_8
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_16
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_32
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_64
//#define CONFIG_SYSCLK_PRES          SYSCLK_PRES_3

// ===== PLL0 (A) Options   (Fpll = (Fclk * PLL_mul) / PLL_div)
// Use mul and div effective values here.
#define CONFIG_PLL0_SOURCE          PLL_SRC_MAINCK_XTAL
#define CONFIG_PLL0_MUL             14
#define CONFIG_PLL0_DIV             1

// ===== UPLL (UTMI) Hardware fixed at 480MHz.
```

```
// ===== USB Clock Source Options    (Fusb = FpllX / USB_div)
// Use div effective value here.
//#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_PLL0
#define CONFIG_USBCLK_SOURCE          USBCLK_SRC_UPLL
#define CONFIG_USBCLK_DIV             1

// ===== Target frequency (System clock)
// - XTAL frequency: 12MHz
// - System clock source: PLLA
// - System clock prescaler: 2 (divided by 2)
// - PLLA source: XTAL
// - PLLA output: XTAL * 14 / 1
// - System clock is: 12 * 14 / 1 /2 = 84MHz
// ===== Target frequency (USB Clock)
// - USB clock source: UPLL
// - USB clock divider: 1 (not divided)
// - UPLL frequency: 480MHz
// - USB clock: 480 / 1 = 480MHz


#endif /* CONF_CLOCK_H_INCLUDED */
```

4.3. conf_clocks.h

4.3.1. SAM D21 Device (USB)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */
#include <clock.h>

#ifndef CONF_CLOCKS_H_INCLUDED
#  define CONF_CLOCKS_H_INCLUDED

/* System clock bus configuration */
#  define CONF_CLOCK_CPU_CLOCK_FAILURE_DETECT     false
#  define CONF_CLOCK_FLASH_WAIT_STATES            2
#  define CONF_CLOCK_CPU_DIVIDER                  SYSTEM_MAIN_CLOCK_DIV_1
#  define CONF_CLOCK_APBA_DIVIDER                 SYSTEM_MAIN_CLOCK_DIV_1
#  define CONF_CLOCK_APBB_DIVIDER                 SYSTEM_MAIN_CLOCK_DIV_1
#  define CONF_CLOCK_APBC_DIVIDER                 SYSTEM_MAIN_CLOCK_DIV_1

/* SYSTEM_CLOCK_SOURCE_OSC8M configuration - Internal 8MHz oscillator */
#  define CONF_CLOCK_OSC8M_PRESCALER              SYSTEM_OSC8M_DIV_1
#  define CONF_CLOCK_OSC8M_ON_DEMAND              true
#  define CONF_CLOCK_OSC8M_RUN_IN_STANDBY         false

/* SYSTEM_CLOCK_SOURCE_XOSC configuration - External clock/oscillator */
#  define CONF_CLOCK_XOSC_ENABLE                  false
#  define CONF_CLOCK_XOSC_EXTERNAL_CRYSTAL
SYSTEM_CLOCK_EXTERNAL_CRYSTAL
#  define CONF_CLOCK_XOSC_EXTERNAL_FREQUENCY      12000000UL
#  define CONF_CLOCK_XOSC_STARTUP_TIME            SYSTEM_XOSC_STARTUP_32768
#  define CONF_CLOCK_XOSC_AUTO_GAIN_CONTROL       true
#  define CONF_CLOCK_XOSC_ON_DEMAND               true
#  define CONF_CLOCK_XOSC_RUN_IN_STANDBY          false

/* SYSTEM_CLOCK_SOURCE_XOSC32K configuration - External 32KHz crystal/clock
```

```
oscillator */
#  define CONF_CLOCK_XOSC32K_ENABLE               false
#  define CONF_CLOCK_XOSC32K_EXTERNAL_CRYSTAL
SYSTEM_CLOCK_EXTERNAL_CRYSTAL
#  define CONF_CLOCK_XOSC32K_STARTUP_TIME
SYSTEM_XOSC32K_STARTUP_65536
#  define CONF_CLOCK_XOSC32K_AUTO_AMPLITUDE_CONTROL  false
#  define CONF_CLOCK_XOSC32K_ENABLE_1KHZ_OUPUT    false
#  define CONF_CLOCK_XOSC32K_ENABLE_32KHZ_OUTPUT  true
#  define CONF_CLOCK_XOSC32K_ON_DEMAND            true
#  define CONF_CLOCK_XOSC32K_RUN_IN_STANDBY       false

/* SYSTEM_CLOCK_SOURCE_OSC32K configuration - Internal 32KHz oscillator */
#  define CONF_CLOCK_OSC32K_ENABLE                false
#  define CONF_CLOCK_OSC32K_STARTUP_TIME          SYSTEM_OSC32K_STARTUP_130
#  define CONF_CLOCK_OSC32K_ENABLE_1KHZ_OUTPUT    true
#  define CONF_CLOCK_OSC32K_ENABLE_32KHZ_OUTPUT   true
#  define CONF_CLOCK_OSC32K_ON_DEMAND             true
#  define CONF_CLOCK_OSC32K_RUN_IN_STANDBY        false

/* SYSTEM_CLOCK_SOURCE_DFLL configuration - Digital Frequency Locked Loop */
#  define CONF_CLOCK_DFLL_ENABLE                  true
#  define CONF_CLOCK_DFLL_LOOP_MODE
SYSTEM_CLOCK_DFLL_LOOP_MODE_USB_RECOVERY
#  define CONF_CLOCK_DFLL_ON_DEMAND               true

/* DFLL open loop mode configuration */
#  define CONF_CLOCK_DFLL_FINE_VALUE              (512)

/* DFLL closed loop mode configuration */
#  define CONF_CLOCK_DFLL_SOURCE_GCLK_GENERATOR   GCLK_GENERATOR_1
#  define CONF_CLOCK_DFLL_MULTIPLY_FACTOR         (48000000 / 32768)
#  define CONF_CLOCK_DFLL_QUICK_LOCK              true
#  define CONF_CLOCK_DFLL_TRACK_AFTER_FINE_LOCK   true
#  define CONF_CLOCK_DFLL_KEEP_LOCK_ON_WAKEUP     true
#  define CONF_CLOCK_DFLL_ENABLE_CHILL_CYCLE      true
#  define CONF_CLOCK_DFLL_MAX_COARSE_STEP_SIZE    (0x1f / 4)
#  define CONF_CLOCK_DFLL_MAX_FINE_STEP_SIZE      (0xff / 4)

/* SYSTEM_CLOCK_SOURCE_DPLL configuration - Digital Phase-Locked Loop */
#  define CONF_CLOCK_DPLL_ENABLE                  false
#  define CONF_CLOCK_DPLL_ON_DEMAND               true
#  define CONF_CLOCK_DPLL_RUN_IN_STANDBY          false
#  define CONF_CLOCK_DPLL_LOCK_BYPASS             false
#  define CONF_CLOCK_DPLL_WAKE_UP_FAST            false
#  define CONF_CLOCK_DPLL_LOW_POWER_ENABLE        false

#  define CONF_CLOCK_DPLL_LOCK_TIME
SYSTEM_CLOCK_SOURCE_DPLL_LOCK_TIME_DEFAULT
#  define CONF_CLOCK_DPLL_REFERENCE_CLOCK
SYSTEM_CLOCK_SOURCE_DPLL_REFERENCE_CLOCK_XOSC32K
#  define CONF_CLOCK_DPLL_FILTER
SYSTEM_CLOCK_SOURCE_DPLL_FILTER_DEFAULT

#  define CONF_CLOCK_DPLL_REFERENCE_FREQUENCY     32768
#  define CONF_CLOCK_DPLL_REFERENCE_DIVIDER       1
#  define CONF_CLOCK_DPLL_OUTPUT_FREQUENCY        48000000

/* DPLL GCLK reference configuration */
#  define CONF_CLOCK_DPLL_REFERENCE_GCLK_GENERATOR  GCLK_GENERATOR_1
/* DPLL GCLK lock timer configuration */
#  define CONF_CLOCK_DPLL_LOCK_GCLK_GENERATOR     GCLK_GENERATOR_1
```

```
/* Set this to true to configure the GCLK when running clocks_init. If set to
 * false, none of the GCLK generators will be configured in clocks_init(). */
#   define CONF_CLOCK_CONFIGURE_GCLK                true

/* Configure GCLK generator 0 (Main Clock) */
#   define CONF_CLOCK_GCLK_0_ENABLE                 true
#   define CONF_CLOCK_GCLK_0_RUN_IN_STANDBY         true
#   define CONF_CLOCK_GCLK_0_CLOCK_SOURCE           SYSTEM_CLOCK_SOURCE_DFLL
#   define CONF_CLOCK_GCLK_0_PRESCALER              1
#   define CONF_CLOCK_GCLK_0_OUTPUT_ENABLE          false

/* Configure GCLK generator 1 */
#   define CONF_CLOCK_GCLK_1_ENABLE                 false
#   define CONF_CLOCK_GCLK_1_RUN_IN_STANDBY         false
#   define CONF_CLOCK_GCLK_1_CLOCK_SOURCE
SYSTEM_CLOCK_SOURCE_XOSC32K
#   define CONF_CLOCK_GCLK_1_PRESCALER              1
#   define CONF_CLOCK_GCLK_1_OUTPUT_ENABLE          false

/* Configure GCLK generator 2 (RTC) */
#   define CONF_CLOCK_GCLK_2_ENABLE                 false
#   define CONF_CLOCK_GCLK_2_RUN_IN_STANDBY         false
#   define CONF_CLOCK_GCLK_2_CLOCK_SOURCE
SYSTEM_CLOCK_SOURCE_OSC32K
#   define CONF_CLOCK_GCLK_2_PRESCALER              32
#   define CONF_CLOCK_GCLK_2_OUTPUT_ENABLE          false

/* Configure GCLK generator 3 */
#   define CONF_CLOCK_GCLK_3_ENABLE                 false
#   define CONF_CLOCK_GCLK_3_RUN_IN_STANDBY         false
#   define CONF_CLOCK_GCLK_3_CLOCK_SOURCE           SYSTEM_CLOCK_SOURCE_OSC8M
#   define CONF_CLOCK_GCLK_3_PRESCALER              1
#   define CONF_CLOCK_GCLK_3_OUTPUT_ENABLE          false

/* Configure GCLK generator 4 */
#   define CONF_CLOCK_GCLK_4_ENABLE                 false
#   define CONF_CLOCK_GCLK_4_RUN_IN_STANDBY         false
#   define CONF_CLOCK_GCLK_4_CLOCK_SOURCE           SYSTEM_CLOCK_SOURCE_OSC8M
#   define CONF_CLOCK_GCLK_4_PRESCALER              1
#   define CONF_CLOCK_GCLK_4_OUTPUT_ENABLE          false

/* Configure GCLK generator 5 */
#   define CONF_CLOCK_GCLK_5_ENABLE                 false
#   define CONF_CLOCK_GCLK_5_RUN_IN_STANDBY         false
#   define CONF_CLOCK_GCLK_5_CLOCK_SOURCE           SYSTEM_CLOCK_SOURCE_OSC8M
#   define CONF_CLOCK_GCLK_5_PRESCALER              1
#   define CONF_CLOCK_GCLK_5_OUTPUT_ENABLE          false

/* Configure GCLK generator 6 */
#   define CONF_CLOCK_GCLK_6_ENABLE                 false
#   define CONF_CLOCK_GCLK_6_RUN_IN_STANDBY         false
#   define CONF_CLOCK_GCLK_6_CLOCK_SOURCE           SYSTEM_CLOCK_SOURCE_OSC8M
#   define CONF_CLOCK_GCLK_6_PRESCALER              1
#   define CONF_CLOCK_GCLK_6_OUTPUT_ENABLE          false

/* Configure GCLK generator 7 */
#   define CONF_CLOCK_GCLK_7_ENABLE                 false
#   define CONF_CLOCK_GCLK_7_RUN_IN_STANDBY         false
#   define CONF_CLOCK_GCLK_7_CLOCK_SOURCE           SYSTEM_CLOCK_SOURCE_OSC8M
#   define CONF_CLOCK_GCLK_7_PRESCALER              1
#   define CONF_CLOCK_GCLK_7_OUTPUT_ENABLE          false
```

```
/* Configure GCLK generator 8 */
#  define CONF_CLOCK_GCLK_8_ENABLE                 false
#  define CONF_CLOCK_GCLK_8_RUN_IN_STANDBY         false
#  define CONF_CLOCK_GCLK_8_CLOCK_SOURCE           SYSTEM_CLOCK_SOURCE_OSC8M
#  define CONF_CLOCK_GCLK_8_PRESCALER              1
#  define CONF_CLOCK_GCLK_8_OUTPUT_ENABLE          false

#endif /* CONF_CLOCKS_H_INCLUDED */
```

4.4. conf_board.h

4.4.1. AT32UC3A0, AT32UC3A1, and AT32UC3B Devices (USBB)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */
#ifndef CONF_BOARD_H_INCLUDED
#define CONF_BOARD_H_INCLUDED

// Enable AT45DBX Component.
#define CONF_BOARD_AT45DBX
// Enable SD and MMC card component.
#define CONF_BOARD_SD_MMC_SPI

#endif /* CONF_BOARD_H_INCLUDED */
```

4.4.2. AT32UC3A3 and AT32UC3A4 Devices (USBB with High Speed Support)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */
#ifndef CONF_BOARD_H_INCLUDED
#define CONF_BOARD_H_INCLUDED

// Enable AT45DBX component.
#define CONF_BOARD_AT45DBX
// Enable SD and MMC card component.
#define CONF_BOARD_SD_MMC_MCI

#endif /* CONF_BOARD_H_INCLUDED */
```

4.4.3. AT32UC3C, ATUCXXD, ATUCXXL3U, and ATUCXXL4U Devices (USBC)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */
#ifndef CONF_BOARD_H_INCLUDED
#define CONF_BOARD_H_INCLUDED

// Enable AT45DBX Component.
#define CONF_BOARD_AT45DBX
// Enable SD and MMC card component.
#define CONF_BOARD_SD_MMC_SPI
```

```
#endif /* CONF_BOARD_H_INCLUDED */
```

4.4.4. SAM3X and SAM3A Devices (UOTGHS: USB OTG High Speed)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */

#ifndef CONF_BOARD_H_INCLUDED
#define CONF_BOARD_H_INCLUDED

/* Nand flash is used */
#define CONF_BOARD_NAND

// Enable SD MMC interface pins through HSMCI
#define CONF_BOARD_SD_MMC_HSMCI

/* USB pins are used */
#define CONF_BOARD_USB_PORT

#endif /* CONF_BOARD_H_INCLUDED */
```

4.4.5. SAM D21 Device (USB)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */

#ifndef CONF_BOARD_H_INCLUDED
#define CONF_BOARD_H_INCLUDED

/* Enable USB VBUS detect */
#define CONF_BOARD_USB_VBUS_DETECT

#endif /* CONF_BOARD_H_INCLUDED */
```

4.5. conf_access.h

4.5.1. AT32UC3A0, AT32UC3A1, and AT32UC3B Devices (USBB)

On EVK1100, the AT45DBx and one SD/MMC are for MSC.

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */

#ifndef _CONF_ACCESS_H_
#define _CONF_ACCESS_H_

#include "compiler.h"
#include "board.h"


#define LUN_0                   DISABLE
#define LUN_1                   ENABLE
```

```
#define LUN_2                 ENABLE
#define LUN_3                 DISABLE
#define LUN_4                 DISABLE
#define LUN_5                 DISABLE
#define LUN_6                 DISABLE
#define LUN_7                 DISABLE
#define LUN_USB               DISABLE


#define VIRTUAL_MEM                           LUN_0
#define LUN_ID_VIRTUAL_MEM                    LUN_ID_0
#define LUN_0_INCLUDE                         "virtual_mem.h"
#define Lun_0_test_unit_ready                 virtual_test_unit_ready
#define Lun_0_read_capacity                   virtual_read_capacity
#define Lun_0_wr_protect                      virtual_wr_protect
#define Lun_0_removal                         virtual_removal
#define Lun_0_usb_read_10                     virtual_usb_read_10
#define Lun_0_usb_write_10                    virtual_usb_write_10
#define Lun_0_mem_2_ram                       virtual_mem_2_ram
#define Lun_0_ram_2_mem                       virtual_ram_2_mem
#define LUN_0_NAME                            "\"On-Chip Virtual Memory
\""

#define AT45DBX_MEM                           LUN_1
#define LUN_ID_AT45DBX_MEM                    LUN_ID_1
#define LUN_1_INCLUDE                         "at45dbx_mem.h"
#define Lun_1_test_unit_ready                 at45dbx_test_unit_ready
#define Lun_1_read_capacity                   at45dbx_read_capacity
#define Lun_1_wr_protect                      at45dbx_wr_protect
#define Lun_1_removal                         at45dbx_removal
#define Lun_1_usb_read_10                     at45dbx_usb_read_10
#define Lun_1_usb_write_10                    at45dbx_usb_write_10
#define Lun_1_mem_2_ram                       at45dbx_df_2_ram
#define Lun_1_ram_2_mem                       at45dbx_ram_2_df
#define LUN_1_NAME                            "\"AT45DBX Data Flash\""

#define SD_MMC_0_MEM                          LUN_2
#define LUN_ID_SD_MMC_0_MEM                   LUN_ID_2
#define LUN_2_INCLUDE                         "sd_mmc_mem.h"
#define Lun_2_test_unit_ready                 sd_mmc_test_unit_ready_0
#define Lun_2_read_capacity                   sd_mmc_read_capacity_0
#define Lun_2_wr_protect                      sd_mmc_wr_protect_0
#define Lun_2_removal                         sd_mmc_removal_0
#define Lun_2_usb_read_10                     sd_mmc_usb_read_10_0
#define Lun_2_usb_write_10                    sd_mmc_usb_write_10_0
#define Lun_2_mem_2_ram                       sd_mmc_mem_2_ram_0
#define Lun_2_ram_2_mem                       sd_mmc_ram_2_mem_0
#define LUN_2_NAME                            "\"SD/MMC Card Slot 0\""

#define MEM_USB                               LUN_USB
#define LUN_ID_MEM_USB                        LUN_ID_USB
#define LUN_USB_INCLUDE                       "host_mem.h"
#define Lun_usb_test_unit_ready(lun)          host_test_unit_ready(lun)
#define Lun_usb_read_capacity(lun, nb_sect)   host_read_capacity(lun,
nb_sect)
#define Lun_usb_read_sector_size(lun)         host_read_sector_size(lun)
#define Lun_usb_wr_protect(lun)               host_wr_protect(lun)
#define Lun_usb_removal()                     host_removal()
#define Lun_usb_mem_2_ram(addr, ram)          host_read_10_ram(addr, ram)
#define Lun_usb_ram_2_mem(addr, ram)          host_write_10_ram(addr,
ram)
#define LUN_USB_NAME                          "\"Host Mass-Storage Memory
```

```
\""

#define memory_start_read_action(nb_sectors)    ui_start_read()
#define memory_stop_read_action()               ui_stop_read()
#define memory_start_write_action(nb_sectors)   ui_start_write()
#define memory_stop_write_action()              ui_stop_write()
#include "ui.h"

#define ACCESS_USB           true
#define ACCESS_MEM_TO_RAM    false
#define ACCESS_STREAM        false
#define ACCESS_STREAM_RECORD false
#define ACCESS_MEM_TO_MEM    false
#define ACCESS_CODEC         false


#define GLOBAL_WR_PROTECT    false


#endif // _CONF_ACCESS_H_
```

4.5.2. AT32UC3A3 and AT32UC3A4 Devices (USBB with High Speed Support)

On EVK1104, the AT45DBx and two SD/MMC slots are for MSC.

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */

#ifndef _CONF_ACCESS_H_
#define _CONF_ACCESS_H_

#include "compiler.h"
#include "board.h"


#define LUN_0                DISABLE
#define LUN_1                ENABLE
#define LUN_2                ENABLE
#define LUN_3                ENABLE
#define LUN_4                DISABLE
#define LUN_5                DISABLE
#define LUN_6                DISABLE
#define LUN_7                DISABLE
#define LUN_USB              DISABLE


#define VIRTUAL_MEM                             LUN_0
#define LUN_ID_VIRTUAL_MEM                      LUN_ID_0
#define LUN_0_INCLUDE                           "virtual_mem.h"
#define Lun_0_test_unit_ready                   virtual_test_unit_ready
#define Lun_0_read_capacity                     virtual_read_capacity
#define Lun_0_wr_protect                        virtual_wr_protect
#define Lun_0_removal                           virtual_removal
#define Lun_0_usb_read_10                       virtual_usb_read_10
#define Lun_0_usb_write_10                      virtual_usb_write_10
#define Lun_0_mem_2_ram                         virtual_mem_2_ram
#define Lun_0_ram_2_mem                         virtual_ram_2_mem
#define LUN_0_NAME                              "\"On-Chip Virtual Memory
\""

#define AT45DBX_MEM                             LUN_1
```

```
#define LUN_ID_AT45DBX_MEM                      LUN_ID_1
#define LUN_1_INCLUDE                           "at45dbx_mem.h"
#define Lun_1_test_unit_ready                   at45dbx_test_unit_ready
#define Lun_1_read_capacity                     at45dbx_read_capacity
#define Lun_1_wr_protect                        at45dbx_wr_protect
#define Lun_1_removal                           at45dbx_removal
#define Lun_1_usb_read_10                       at45dbx_usb_read_10
#define Lun_1_usb_write_10                      at45dbx_usb_write_10
#define Lun_1_mem_2_ram                         at45dbx_df_2_ram
#define Lun_1_ram_2_mem                         at45dbx_ram_2_df
#define LUN_1_NAME                              "\"AT45DBX Data Flash\""

#define SD_MMC_0_MEM                            LUN_2
#define LUN_ID_SD_MMC_0_MEM                     LUN_ID_2
#define LUN_2_INCLUDE                           "sd_mmc_mem.h"
#define Lun_2_test_unit_ready                   sd_mmc_test_unit_ready_0
#define Lun_2_read_capacity                     sd_mmc_read_capacity_0
#define Lun_2_wr_protect                        sd_mmc_wr_protect_0
#define Lun_2_removal                           sd_mmc_removal_0
#define Lun_2_usb_read_10                       sd_mmc_usb_read_10_0
#define Lun_2_usb_write_10                      sd_mmc_usb_write_10_0
#define Lun_2_mem_2_ram                         sd_mmc_mem_2_ram_0
#define Lun_2_ram_2_mem                         sd_mmc_ram_2_mem_0
#define LUN_2_NAME                              "\"SD/MMC Card Slot 0\""

#define SD_MMC_1_MEM                            LUN_3
#define LUN_ID_SD_MMC_1_MEM                     LUN_ID_3
#define LUN_3_INCLUDE                           "sd_mmc_mem.h"
#define Lun_3_test_unit_ready                   sd_mmc_test_unit_ready_1
#define Lun_3_read_capacity                     sd_mmc_read_capacity_1
#define Lun_3_wr_protect                        sd_mmc_wr_protect_1
#define Lun_3_removal                           sd_mmc_removal_1
#define Lun_3_usb_read_10                       sd_mmc_usb_read_10_1
#define Lun_3_usb_write_10                      sd_mmc_usb_write_10_1
#define Lun_3_mem_2_ram                         sd_mmc_mem_2_ram_1
#define Lun_3_ram_2_mem                         sd_mmc_ram_2_mem_1
#define LUN_3_NAME                              "\"SD/MMC Card Slot 1\""

#define MEM_USB                                 LUN_USB
#define LUN_ID_MEM_USB                          LUN_ID_USB
#define LUN_USB_INCLUDE                         "host_mem.h"
#define Lun_usb_test_unit_ready(lun)            host_test_unit_ready(lun)
#define Lun_usb_read_capacity(lun, nb_sect)     host_read_capacity(lun,
nb_sect)
#define Lun_usb_read_sector_size(lun)           host_read_sector_size(lun)
#define Lun_usb_wr_protect(lun)                 host_wr_protect(lun)
#define Lun_usb_removal()                       host_removal()
#define Lun_usb_mem_2_ram(addr, ram)            host_read_10_ram(addr, ram)
#define Lun_usb_ram_2_mem(addr, ram)            host_write_10_ram(addr,
ram)
#define LUN_USB_NAME                            "\"Host Mass-Storage Memory
\""

#define memory_start_read_action(nb_sectors)    ui_start_read()
#define memory_stop_read_action()               ui_stop_read()
#define memory_start_write_action(nb_sectors)   ui_start_write()
#define memory_stop_write_action()              ui_stop_write()
#include "ui.h"

#define ACCESS_USB           true
#define ACCESS_MEM_TO_RAM    false
#define ACCESS_STREAM        false
#define ACCESS_STREAM_RECORD false
```

```
#define ACCESS_MEM_TO_MEM    false
#define ACCESS_CODEC         false


#define GLOBAL_WR_PROTECT    false


#endif // _CONF_ACCESS_H_
```

4.5.3. AT32UC3C, ATUCXXD, ATUCXXL3U, and ATUCXXL4U Devices (USBC)

On EVK1100, the AT45DBx and one SD/MMC are for MSC.

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */

#ifndef _CONF_ACCESS_H_
#define _CONF_ACCESS_H_

#include "compiler.h"
#include "board.h"


#define LUN_0                DISABLE
#define LUN_1                ENABLE
#define LUN_2                ENABLE
#define LUN_3                DISABLE
#define LUN_4                DISABLE
#define LUN_5                DISABLE
#define LUN_6                DISABLE
#define LUN_7                DISABLE
#define LUN_USB              DISABLE


#define VIRTUAL_MEM                          LUN_0
#define LUN_ID_VIRTUAL_MEM                   LUN_ID_0
#define LUN_0_INCLUDE                        "virtual_mem.h"
#define Lun_0_test_unit_ready                virtual_test_unit_ready
#define Lun_0_read_capacity                  virtual_read_capacity
#define Lun_0_wr_protect                     virtual_wr_protect
#define Lun_0_removal                        virtual_removal
#define Lun_0_usb_read_10                    virtual_usb_read_10
#define Lun_0_usb_write_10                   virtual_usb_write_10
#define Lun_0_mem_2_ram                      virtual_mem_2_ram
#define Lun_0_ram_2_mem                      virtual_ram_2_mem
#define LUN_0_NAME                           "\"On-Chip Virtual Memory\""

#define AT45DBX_MEM                          LUN_1
#define LUN_ID_AT45DBX_MEM                   LUN_ID_1
#define LUN_1_INCLUDE                        "at45dbx_mem.h"
#define Lun_1_test_unit_ready                at45dbx_test_unit_ready
#define Lun_1_read_capacity                  at45dbx_read_capacity
#define Lun_1_wr_protect                     at45dbx_wr_protect
#define Lun_1_removal                        at45dbx_removal
#define Lun_1_usb_read_10                    at45dbx_usb_read_10
#define Lun_1_usb_write_10                   at45dbx_usb_write_10
#define Lun_1_mem_2_ram                      at45dbx_df_2_ram
#define Lun_1_ram_2_mem                      at45dbx_ram_2_df
#define LUN_1_NAME                           "\"AT45DBX Data Flash\""
```

```
#define SD_MMC_0_MEM                                LUN_2
#define LUN_ID_SD_MMC_0_MEM                         LUN_ID_2
#define LUN_2_INCLUDE                               "sd_mmc_mem.h"
#define Lun_2_test_unit_ready                       sd_mmc_test_unit_ready_0
#define Lun_2_read_capacity                         sd_mmc_read_capacity_0
#define Lun_2_wr_protect                            sd_mmc_wr_protect_0
#define Lun_2_removal                               sd_mmc_removal_0
#define Lun_2_usb_read_10                           sd_mmc_usb_read_10_0
#define Lun_2_usb_write_10                          sd_mmc_usb_write_10_0
#define Lun_2_mem_2_ram                             sd_mmc_mem_2_ram_0
#define Lun_2_ram_2_mem                             sd_mmc_ram_2_mem_0
#define LUN_2_NAME                                  "\"SD/MMC Card Slot 0\""

#define MEM_USB                                    LUN_USB
#define LUN_ID_MEM_USB                             LUN_ID_USB
#define LUN_USB_INCLUDE                            "host_mem.h"
#define Lun_usb_test_unit_ready(lun)               host_test_unit_ready(lun)
#define Lun_usb_read_capacity(lun, nb_sect)        host_read_capacity(lun,
nb_sect)
#define Lun_usb_read_sector_size(lun)              host_read_sector_size(lun)
#define Lun_usb_wr_protect(lun)                    host_wr_protect(lun)
#define Lun_usb_removal()                          host_removal()
#define Lun_usb_mem_2_ram(addr, ram)               host_read_10_ram(addr, ram)
#define Lun_usb_ram_2_mem(addr, ram)               host_write_10_ram(addr, ram)
#define LUN_USB_NAME                               "\"Host Mass-Storage Memory
\""

#define memory_start_read_action(nb_sectors)    ui_start_read()
#define memory_stop_read_action()               ui_stop_read()
#define memory_start_write_action(nb_sectors)   ui_start_write()
#define memory_stop_write_action()              ui_stop_write()
#include "ui.h"

#define ACCESS_USB           true
#define ACCESS_MEM_TO_RAM    false
#define ACCESS_STREAM        false
#define ACCESS_STREAM_RECORD false
#define ACCESS_MEM_TO_MEM    false
#define ACCESS_CODEC         false


#define GLOBAL_WR_PROTECT    false


#endif // _CONF_ACCESS_H_
```

4.5.4. SAM3X and SAM3A Devices (UOTGHS: USB OTG High Speed)

On SAM3X-EK, the SD/MMC and on-board nand are for MSC.

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */

#ifndef _CONF_ACCESS_H_
#define _CONF_ACCESS_H_

#include "compiler.h"
#include "board.h"


#define LUN_0                DISABLE
```

```
#define LUN_1                 DISABLE
#define LUN_2                 ENABLE
#define LUN_3                 DISABLE
#define LUN_4                 ENABLE
#define LUN_5                 DISABLE
#define LUN_6                 DISABLE
#define LUN_7                 DISABLE
#define LUN_USB               DISABLE


#define VIRTUAL_MEM                           LUN_0
#define LUN_ID_VIRTUAL_MEM                    LUN_ID_0
#define LUN_0_INCLUDE                         "virtual_mem.h"
#define Lun_0_test_unit_ready                 virtual_test_unit_ready
#define Lun_0_read_capacity                   virtual_read_capacity
#define Lun_0_unload                          NULL /* Can not be unloaded */
#define Lun_0_wr_protect                      virtual_wr_protect
#define Lun_0_removal                         virtual_removal
#define Lun_0_usb_read_10                     virtual_usb_read_10
#define Lun_0_usb_write_10                    virtual_usb_write_10
#define Lun_0_mem_2_ram                       virtual_mem_2_ram
#define Lun_0_ram_2_mem                       virtual_ram_2_mem
#define LUN_0_NAME                            "\"On-Chip Virtual Memory
\""

#define AT45DBX_MEM                           LUN_1
#define LUN_ID_AT45DBX_MEM                    LUN_ID_1
#define LUN_1_INCLUDE                         "at45dbx_mem.h"
#define Lun_1_test_unit_ready                 at45dbx_test_unit_ready
#define Lun_1_read_capacity                   at45dbx_read_capacity
#define Lun_1_unload                          NULL /* Can not be unloaded */
#define Lun_1_wr_protect                      at45dbx_wr_protect
#define Lun_1_removal                         at45dbx_removal
#define Lun_1_usb_read_10                     at45dbx_usb_read_10
#define Lun_1_usb_write_10                    at45dbx_usb_write_10
#define Lun_1_mem_2_ram                       at45dbx_df_2_ram
#define Lun_1_ram_2_mem                       at45dbx_ram_2_df
#define LUN_1_NAME                            "\"AT45DBX Data Flash\""

#define SD_MMC_0_MEM                          LUN_2
#define LUN_ID_SD_MMC_0_MEM                   LUN_ID_2
#define LUN_2_INCLUDE                         "sd_mmc_mem.h"
#define Lun_2_test_unit_ready                 sd_mmc_test_unit_ready_0
#define Lun_2_read_capacity                   sd_mmc_read_capacity_0
#define Lun_2_unload                          sd_mmc_unload_0
#define Lun_2_wr_protect                      sd_mmc_wr_protect_0
#define Lun_2_removal                         sd_mmc_removal_0
#define Lun_2_usb_read_10                     sd_mmc_usb_read_10_0
#define Lun_2_usb_write_10                    sd_mmc_usb_write_10_0
#define Lun_2_mem_2_ram                       sd_mmc_mem_2_ram_0
#define Lun_2_ram_2_mem                       sd_mmc_ram_2_mem_0
#define LUN_2_NAME                            "\"SD/MMC Card Slot 0\""

#define NAND_FLASH_MEM                        LUN_4
#define LUN_ID_NAND_FLASH_MEM                 LUN_ID_4
#define LUN_4_INCLUDE                         "nand_flash_mem.h"
#define Lun_4_test_unit_ready                 nand_flash_test_unit_ready
#define Lun_4_read_capacity                   nand_flash_read_capacity
#define Lun_4_unload                          NULL
#define Lun_4_wr_protect                      nand_flash_wr_protect
#define Lun_4_removal                         nand_flash_removal
#define Lun_4_usb_read_10                     nand_flash_usb_read_10
```

```
#define Lun_4_usb_write_10                          nand_flash_usb_write_10
#define Lun_4_mem_2_ram                             nand_flash_mem_2_ram
#define Lun_4_ram_2_mem                             nand_flash_ram_2_mem
#define LUN_4_NAME                                  "\"nand_flash on EBI\""

#define MEM_USB                                     LUN_USB
#define LUN_ID_MEM_USB                              LUN_ID_USB
#define LUN_USB_INCLUDE                             "uhi_msc_mem.h"
#define Lun_usb_get_lun()                           uhi_msc_mem_get_lun()
#define Lun_usb_test_unit_ready(lun)
uhi_msc_mem_test_unit_ready(lun)
#define Lun_usb_read_capacity(lun, nb_sect)
uhi_msc_mem_read_capacity(lun, nb_sect)
#define Lun_usb_read_sector_size(lun)
uhi_msc_mem_read_sector_size(lun)
#define Lun_usb_wr_protect(lun)                     uhi_msc_mem_wr_protect(lun)
#define Lun_usb_removal()                           uhi_msc_mem_removal()
#define Lun_usb_mem_2_ram(addr, ram)
uhi_msc_mem_read_10_ram(addr, ram)
#define Lun_usb_ram_2_mem(addr, ram)
uhi_msc_mem_write_10_ram(addr, ram)
#define LUN_USB_NAME                                "\"Host Mass-Storage Memory
\""

#define memory_start_read_action(nb_sectors)        ui_start_read()
#define memory_stop_read_action()                   ui_stop_read()
#define memory_start_write_action(nb_sectors)       ui_start_write()
#define memory_stop_write_action()                  ui_stop_write()
#include "ui.h"

#define ACCESS_USB           true
#define ACCESS_MEM_TO_RAM    false
#define ACCESS_STREAM        false
#define ACCESS_STREAM_RECORD false
#define ACCESS_MEM_TO_MEM    false
#define ACCESS_CODEC         false


#define GLOBAL_WR_PROTECT    false


#endif // _CONF_ACCESS_H_
```

4.5.5. SAM D21 Device (USB)

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */

#ifndef _CONF_ACCESS_H_
#define _CONF_ACCESS_H_

#include "compiler.h"
#include "board.h"


#define LUN_0                ENABLE
#define LUN_1                DISABLE
#define LUN_2                DISABLE
#define LUN_3                DISABLE
#define LUN_4                DISABLE
```

```
#define LUN_5                DISABLE
#define LUN_6                DISABLE
#define LUN_7                DISABLE
#define LUN_USB              DISABLE


#define VIRTUAL_MEM                              LUN_0
#define LUN_ID_VIRTUAL_MEM                       LUN_ID_0
#define LUN_0_INCLUDE                            "virtual_mem.h"
#define Lun_0_test_unit_ready                    virtual_test_unit_ready
#define Lun_0_read_capacity                      virtual_read_capacity
#define Lun_0_unload                             NULL /* Can not be unloaded */
#define Lun_0_wr_protect                         virtual_wr_protect
#define Lun_0_removal                            virtual_removal
#define Lun_0_usb_read_10                        virtual_usb_read_10
#define Lun_0_usb_write_10                       virtual_usb_write_10
#define Lun_0_mem_2_ram                          virtual_mem_2_ram
#define Lun_0_ram_2_mem                          virtual_ram_2_mem
#define LUN_0_NAME                               "\"On-Chip Virtual Memory
\""

#define MEM_USB                                  LUN_USB
#define LUN_ID_MEM_USB                           LUN_ID_USB
#define LUN_USB_INCLUDE                          "uhi_msc_mem.h"
#define Lun_usb_get_lun()                        uhi_msc_mem_get_lun()
#define Lun_usb_test_unit_ready(lun)
uhi_msc_mem_test_unit_ready(lun)
#define Lun_usb_read_capacity(lun, nb_sect)
uhi_msc_mem_read_capacity(lun, nb_sect)
#define Lun_usb_read_sector_size(lun)
uhi_msc_mem_read_sector_size(lun)
#define Lun_usb_wr_protect(lun)                  uhi_msc_mem_wr_protect(lun)
#define Lun_usb_removal()                        uhi_msc_mem_removal()
#define Lun_usb_mem_2_ram(addr, ram)
uhi_msc_mem_read_10_ram(addr, ram)
#define Lun_usb_ram_2_mem(addr, ram)
uhi_msc_mem_write_10_ram(addr, ram)
#define LUN_USB_NAME                             "\"Host Mass-Storage Memory
\""

#define memory_start_read_action(nb_sectors)     ui_start_read()
#define memory_stop_read_action()                ui_stop_read()
#define memory_start_write_action(nb_sectors)    ui_start_write()
#define memory_stop_write_action()               ui_stop_write()
#include "ui.h"

#define ACCESS_USB           true
#define ACCESS_MEM_TO_RAM    false
#define ACCESS_STREAM        false
#define ACCESS_STREAM_RECORD false
#define ACCESS_MEM_TO_MEM    false
#define ACCESS_CODEC         false


#define GLOBAL_WR_PROTECT    false


#endif // _CONF_ACCESS_H_
```

4.6. conf_virtual_mem.h

4.6.1. On-chip Virtual Memory Disk

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */


#ifndef _CONF_VIRTUAL_MEM_H_
#define _CONF_VIRTUAL_MEM_H_

#define VMEM_NB_SECTOR 48 // Internal RAM 24KB (should > 20KB or PC can
not format it)

#endif // _CONF_VIRTUAL_MEM_H_
```

4.6.2. On-board Virtual Memory Disk

```
/*
 * Support and FAQ: visit <a href="http://www.atmel.com/design-support/">Atmel
Support</a>
 */


#ifndef _CONF_VIRTUAL_MEM_H_
#define _CONF_VIRTUAL_MEM_H_

// Start address of Virtual Memory if it's on external RAM (PSRAM)
#define VMEM_ADDRESS   0x60000000
#define VMEM_NB_SECTOR 2048 // External PSRAM 1MB

#endif // _CONF_VIRTUAL_MEM_H_
```

5. USB Device Basic Setup

5.1. Custom Configuration

The following USB Device configuration must be included in the conf_usb.h file of the application:

1. USB_DEVICE_VENDOR_ID (Word).

Vendor ID provided by USB org (Atmel 0x03EB).

2. USB_DEVICE_PRODUCT_ID (Word).

Product ID (Referenced in usb_atmel.h).

3. USB_DEVICE_MAJOR_VERSION (Byte).

Major version of the device.

4. USB_DEVICE_MINOR_VERSION (Byte).

Minor version of the device.

5. USB_DEVICE_MANUFACTURE_NAME (string).

ASCII name for the manufacture.

6. USB_DEVICE_PRODUCT_NAME (string).

ASCII name for the product.

7. USB_DEVICE_SERIAL_NAME (string).

ASCII name to enable and set a serial number.

8. USB_DEVICE_POWER (Numeric).

(unit mA) Maximum device power.

9. USB_DEVICE_ATTR (Byte).

USB attributes available:

- USB_CONFIG_ATTR_SELF_POWERED
- USB_CONFIG_ATTR_REMOTE_WAKEUP

Note: If remote wake is enabled, this defines remotewakeup callbacks.

10. USB_DEVICE_LOW_SPEED (Only defined).

Force the USB Device to run in low speed.

11. USB_DEVICE_HS_SUPPORT (Only defined).

Authorize the USB Device to run in high speed.

12. USB_DEVICE_MAX_EP (Byte).

Define the maximum endpoint number used by the USB Device.

This one is already defined in the UDI default configuration. E.g.:

- When endpoint control 0x00, endpoint 0x01, and endpoint 0x82 is used, then USB_DEVICE_MAX_EP=2
- When only endpoint control 0x00 is used, then USB_DEVICE_MAX_EP=0

- When endpoint 0x01 and endpoint 0x81 is used, then USB_DEVICE_MAX_EP=1 (configuration not possible on USBB interface)

5.2. VBUS Monitoring

The VBUS monitoring is used only for USB SELF Power application.

- By default the USB device is automatically attached when VBUS is high or when USB starts for devices without internal VBUS monitoring. conf_usb.h file does not contain definition USB_DEVICE_ATTACH_AUTO_DISABLE.

```
//#define USB_DEVICE_ATTACH_AUTO_DISABLE
```

- Add custom VBUS monitoring. conf_usb.h file contains define USB_DEVICE_ATTACH_AUTO_DISABLE:

```
#define USB_DEVICE_ATTACH_AUTO_DISABLE
```

User C-file contains:

```
// Authorize VBUS monitoring
if (!udc_include_vbus_monitoring()) {
  // Implement custom VBUS monitoring via GPIO or other
}
Event_VBUS_present() // VBUS interrupt or GPIO interrupt or other
{
  // Attach USB Device
  udc_attach();
}
```

- Case of battery charging. conf_usb.h file contains define USB_DEVICE_ATTACH_AUTO_DISABLE:

```
#define USB_DEVICE_ATTACH_AUTO_DISABLE
```

User C-file contains:

```
Event VBUS present() // VBUS interrupt or GPIO interrupt or ..
{
  // Authorize battery charging, but wait key press to start USB.
}
Event Key press()
{
  // Stop batteries charging
  // Start USB
  udc_attach();
}
```

5.3. USB Device Basic Setup

5.3.1. USB Device Controller (UDC) - Prerequisites

Common prerequisites for all USB devices.

This module is based on USB device stack full interrupt driven, and supporting sleepmgr. For AVR® and Atmel® | SMART ARM®-based SAM3/4 devices the clock services is supported. For SAM D21 devices the clock driver is supported.

The following procedure must be executed to set up the project correctly:

- Specify the clock configuration:
  - XMEGA® USB devices need 48MHz clock input. XMEGA USB devices need CPU frequency higher than 12MHz. You can use either an internal RC 48MHz auto calibrated by Start of Frames or an external OSC.
  - UC3 and SAM3/4 devices without USB high speed support need 48MHz clock input. You must use a PLL and an external OSC.
  - UC3 and SAM3/4 devices with USB high speed support need 12MHz clock input. You must use an external OSC.
  - UC3 devices with USBC hardware need CPU frequency higher than 25MHz
  - SAM D21 devices without USB high speed support need 48MHz clock input. You should use DFLL with USBCRM.
- In conf_board.h, the define CONF_BOARD_USB_PORT must be added to enable USB lines. (Not mandatory for all boards).
- Enable interrupts
- Initialize the clock service

The usage of sleep manager service is optional, but recommended to reduce power consumption:

- Initialize the sleep manager service
- Activate sleep mode when the application is in IDLE state

For AVR and SAM3/4 devices, add to the initialization code:

```
sysclk_init();
irq_initialize_vectors();
cpu_irq_enable();
board_init();
sleepmgr_init(); // Optional
```

For SAM D21 devices, add to the initialization code:

```
system_init();
irq_initialize_vectors();
cpu_irq_enable();
sleepmgr_init(); // Optional
```

Add to the main IDLE loop:

```
sleepmgr_enter_sleep(); // Optional
```

5.3.2. USB Device Controller (UDC) - Example Code

Common example code for all USB devices.

Content of conf_usb.h:

```
#define USB_DEVICE_VENDOR_ID 0x03EB
#define USB_DEVICE_PRODUCT_ID 0xXXXX
#define USB_DEVICE_MAJOR_VERSION 1
#define USB_DEVICE_MINOR_VERSION 0
#define USB_DEVICE_POWER 100
#define USB_DEVICE_ATTR USB_CONFIG_ATTR_BUS_POWERED
```

Add to application C-file:

```
void usb_init(void)
{
```

```
  udc_start();
}
```

5.3.3. USB Device Controller (UDC) - Workflow

Common workflow for all USB devices.

1. Ensure that conf_usb.h is available and contains the following configuration, which is the main USB device configuration:

```
// Vendor ID provided by USB org (Atmel 0x03EB)
#define USB_DEVICE_VENDOR_ID 0x03EB // Type Word
// Product ID (Atmel PID referenced in usb_atmel.h)
#define USB_DEVICE_PRODUCT_ID 0xXXXX // Type Word
// Major version of the device
#define USB_DEVICE_MAJOR_VERSION 1 // Type Byte
// Minor version of the device
#define USB_DEVICE_MINOR_VERSION 0 // Type Byte
// Maximum device power (mA)
#define USB_DEVICE_POWER 100 // Type 9-bits
// USB attributes to enable features
#define USB_DEVICE_ATTR USB_CONFIG_ATTR_BUS_POWERED // Flags
```

2. Call the USB device stack start function to enable stack and start USB:

```
udc_start();
```

Note: In case of USB dual roles (Device and Host) managed through USB OTG connector (USB ID pin), the call of `udc_start()` must be removed and replaced by uhc_start(). Refer to section "Dual roles" for further information in the application note: Atmel AVR4950: ASF - USB Host Stack

5.4. conf_clock.h Examples

Content of XMEGA conf_clock.h:

```
// Configuration based on internal RC:
// USB clock need of 48MHz
#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_RCOSC
#define CONFIG_OSC_RC32_CAL         48000000UL
#define CONFIG_OSC_AUTOCAL_RC32MHZ_REF_OSC  OSC_ID_USBSOF
// CPU clock need of clock > 12MHz to run with USB (Here 24MHz)
#define CONFIG_SYSCLK_SOURCE     SYSCLK_SRC_RC32MHZ
#define CONFIG_SYSCLK_PSADIV     SYSCLK_PSADIV_2
#define CONFIG_SYSCLK_PSBCDIV    SYSCLK_PSBCDIV_1_1
```

Content of conf_clock.h for AT32UC3A0, AT32UC3A1, and AT32UC3B devices (USBB):

```
// Configuration based on 12MHz external OSC:
#define CONFIG_PLL1_SOURCE          PLL_SRC_OSC0
#define CONFIG_PLL1_MUL             8
#define CONFIG_PLL1_DIV             2
#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_PLL1
#define CONFIG_USBCLK_DIV           1 // Fusb = Fsys/(2 ^ USB_div)
```

Content of conf_clock.h for AT32UC3A3 and AT32UC3A4 devices (USBB with high speed support):

```
// Configuration based on 12MHz external OSC:
#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_OSC0
#define CONFIG_USBCLK_DIV           1 // Fusb = Fsys/(2 ^ USB_div)
```

Content of conf_clock.h for AT32UC3C, ATUCXXD, ATUCXXL3U, and ATUCXXL4U devices (USBC):

```
// Configuration based on 12MHz external OSC:
#define CONFIG_PLL1_SOURCE          PLL_SRC_OSC0
#define CONFIG_PLL1_MUL             8
#define CONFIG_PLL1_DIV             2
#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_PLL1
#define CONFIG_USBCLK_DIV           1 // Fusb = Fsys/(2 ^ USB_div)
// CPU clock need of clock > 25MHz to run with USBC
#define CONFIG_SYSCLK_SOURCE        SYSCLK_SRC_PLL1
```

Content of conf_clock.h for SAM3S, SAM3SD, and SAM4S devices (UPD: USB Peripheral Device):

```
// PLL1 (B) Options   (Fpll = (Fclk * PLL_mul) / PLL_div)
#define CONFIG_PLL1_SOURCE          PLL_SRC_MAINCK_XTAL
#define CONFIG_PLL1_MUL             16
#define CONFIG_PLL1_DIV             2
// USB Clock Source Options   (Fusb = FpllX / USB_div)
#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_PLL1
#define CONFIG_USBCLK_DIV           2
```

Content of conf_clock.h for SAM3U device (UPDHS: USB Peripheral Device High Speed):

```
// USB Clock Source fixed at UPLL.
```

Content of conf_clock.h for SAM3X and SAM3A devices (UOTGHS: USB OTG High Speed):

```
// USB Clock Source fixed at UPLL.
#define CONFIG_USBCLK_SOURCE        USBCLK_SRC_UPLL
#define CONFIG_USBCLK_DIV           1
```

Content of conf_clocks.h for SAM D21 devices (USB):

```
// System clock bus configuration
#  define CONF_CLOCK_FLASH_WAIT_STATES            2

// USB Clock Source fixed at DFLL.
// SYSTEM_CLOCK_SOURCE_DFLL configuration - Digital Frequency Locked Loop
#  define CONF_CLOCK_DFLL_ENABLE                  true
#  define CONF_CLOCK_DFLL_LOOP_MODE
SYSTEM_CLOCK_DFLL_LOOP_MODE_USB_RECOVERY
#  define CONF_CLOCK_DFLL_ON_DEMAND               true

// Set this to true to configure the GCLK when running clocks_init.
// If set to false, none of the GCLK generators will be configured in clocks_init().
#  define CONF_CLOCK_CONFIGURE_GCLK               true

// Configure GCLK generator 0 (Main Clock)
#  define CONF_CLOCK_GCLK_0_ENABLE                true
#  define CONF_CLOCK_GCLK_0_RUN_IN_STANDBY        true
#  define CONF_CLOCK_GCLK_0_CLOCK_SOURCE          SYSTEM_CLOCK_SOURCE_DFLL
#  define CONF_CLOCK_GCLK_0_PRESCALER             1
#  define CONF_CLOCK_GCLK_0_OUTPUT_ENABLE         false
```

6. Document Revision History

Doc. Rev.	Date	Comments
42343B	12/2015	Fixed typos
42343A	12/2014	Initial release

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**