

# Atmel Software Framework Checklist for ARM Cortex ASF C Projects

Date: 30 May 2016

Version: 1.0  
Using Atmel Studio 7

Author: © 2016, W.A. Smith, <http://startingelectronics.org>

This checklist goes through the steps needed to make a new ASF project for Atmel ARM Cortex microcontrollers, which files to modify and how to add ASF modules to a project. The list is intended as a quick reference for starting new ASF projects. For a detailed tutorial on using ASF on ARM Cortex microcontrollers, see <http://startingelectronics.org/software/atmel/asf-arm-tutorial/>

Atmel Software Framework supported devices and latest documentation can be found at <http://asf.atmel.com/docs/latest/>

## Create a New ASF Project

1. Open Atmel Studio and start a new project using **File ► New ► Project...** from the top menu bar or **Ctrl + Shift + N** on the keyboard.
2. In the New Project dialog box, select **GCC ASF Board Project**, give the project a name and select the project location.
3. In the Board Selection dialog box enter the ARM microcontroller part number in the search field and then select User Board template for the desired microcontroller.

## Select Clock Sources and Frequencies

4. Open the ASF Wizard using **ASF ► ASF Wizard** from the top menu bar or **Alt + W** on the keyboard.
5. Search for **clock** under Available Modules in the ASF Wizard. Select **System Clock Control (service)** and then click the **Add >>** button. Click the **Apply** button to add the selected module.
6. Use Solution Explorer (figure 1, page 2) to open **src ► config ► conf\_board.h** and add definitions for the clock resonators and oscillator start-up time, e.g.

```
// clock resonators
#define BOARD_FREQ_SLCK_XTAL      (32768U)
#define BOARD_FREQ_SLCK_BYPASS   (32768U)
#define BOARD_FREQ_MAINCK_XTAL   (12000000U)
#define BOARD_FREQ_MAINCK_BYPASS (12000000U)
#define BOARD_MCK                  CHIP_FREQ_CPU_MAX
#define BOARD_OSC_STARTUP_US      15625
```

Without these definitions the compiler will issue several warnings.

7. Use Solution Explorer to open **src ► config ► conf\_clock.h** and comment out and uncomment the definitions to select the desired main clock source and settings.

8. To select an external 32.768kHz crystal for the slow clock source and for an example of changing the main clock to use an external crystal, see <http://startingelectronics.org/software/atmel/asf-arm-tutorial/external-crystals/>

## Add Hardware Definitions

9. Open **src ► config ► conf\_board.h** from Solution Explorer to add hardware definitions, e.g.:

```
// output pin for LED
#define LED0      IOPORT_CREATE_PIN(PIOB, 14)
// input pin for switch
#define SW0       IOPORT_CREATE_PIN(PIOA, 30)
```

## Add Board Specific Initialization Code and Initialize System Clock

10. Use Solution Explorer to open **src ► ASF ► common ► boards ► user\_board ► init.c** and add board specific initialization code in the **board\_init()** function in **init.c**, e.g. code to disable the watchdog timer, code to set the pin direction of GPIO pins defined in **conf\_board.h**, etc.

```
WDT->WDT_MR = WDT_MR_WDDIS;           // disable watchdog
ioport_init();                         // call before using IOPORT service
ioport_set_pin_dir(LED0, IOPORT_DIR_OUTPUT); // LED pin set as output
ioport_set_pin_dir(SW0, IOPORT_DIR_INPUT);  // switch pin set as input
```

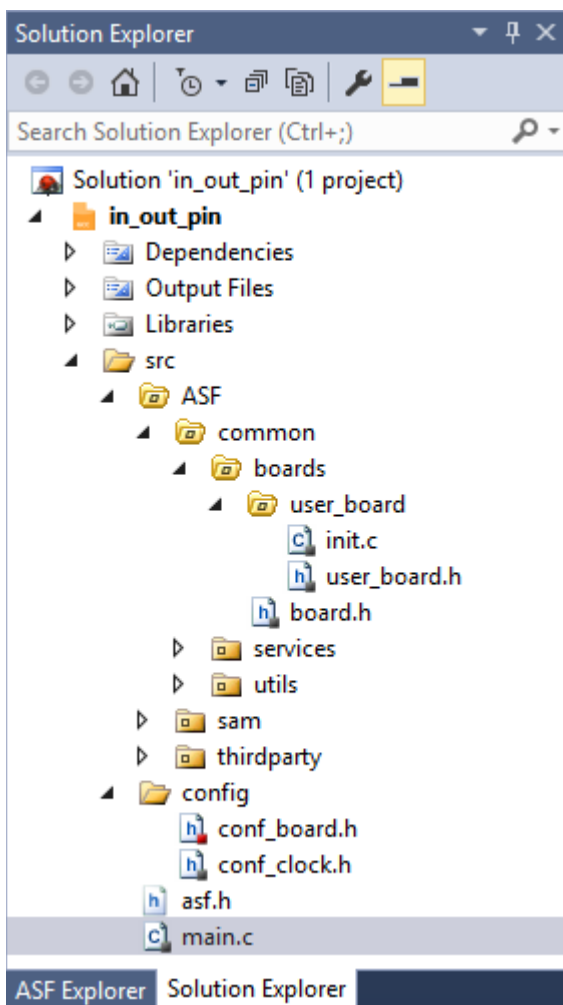


Figure 1: Atmel Studio Solution Explorer

In the above code example, the **IOPORT – General purpose I/O service (service)** ASF module must be added to the project to be able to call functions starting with **ioport\_** in the project.

11. Call **sysclk\_init()** in **main.c**

```
int main (void)
{
    sysclk_init();
    board_init();

    while (1) {
    }
}
```

Finally the application code for the project can be written.