

SAMD21 xplained Pro - Stop Watch Timer Controller Tutorial - PhillyNJ

Overview

In this tutorial I will show you how to implement a Stop Watch Timer Controller. This tutorial is based off the Atmel's LWMesh Peer2Peer Application for the SAMR21 xplained pro. While working on the example project, there was a System Software Timer routine that I wanted to implement on other projects.

This code works like a stop watch. You can start and stop the timer as needed. This can be very useful measuring timeouts. I recently worked on a LWMesh project with 2 SAMR21 Xplained Pro boards. The SAMR21 is a Radio Tranceivers. The project consisted of sending large amounts of data from one board to the next via RF. I implemented a time out to know that that sender know that the receiver never received all the data.

In you are familiar with the Atmel Software Foundation (ASF), you will realize that ASF provides a lot of examples for working with their AVR and SAM 32-bit ARM® Cortex®-M0+ processors. However, implement these example projects in another project can be daunting and time consuming. This tutorial takes the leg work out of implementing a Stop Watch timer controller and adding it any SAMD21 project.

Atmel's AVR2130: Lightweight Mesh Developer Guide provides a detail of the LWMesh library. For reference, section **5.11.2 Software timers**, which covers the code we will be implementing in our project.

Hardware Prerequisites

- Atmel® | SMART SAM D21 Xplained Pro Evaluation kit
- Micro-USB standard device cable

Software Prerequisites

- Atmel Studio 7
- Atmel Software Framework (ASF) (3.19.0)
- Download the File here <https://github.com/PhillyNJ/SamD21SystemTimerExample>

Tutorial Time

Estimated completion time: 30min.

Download Files

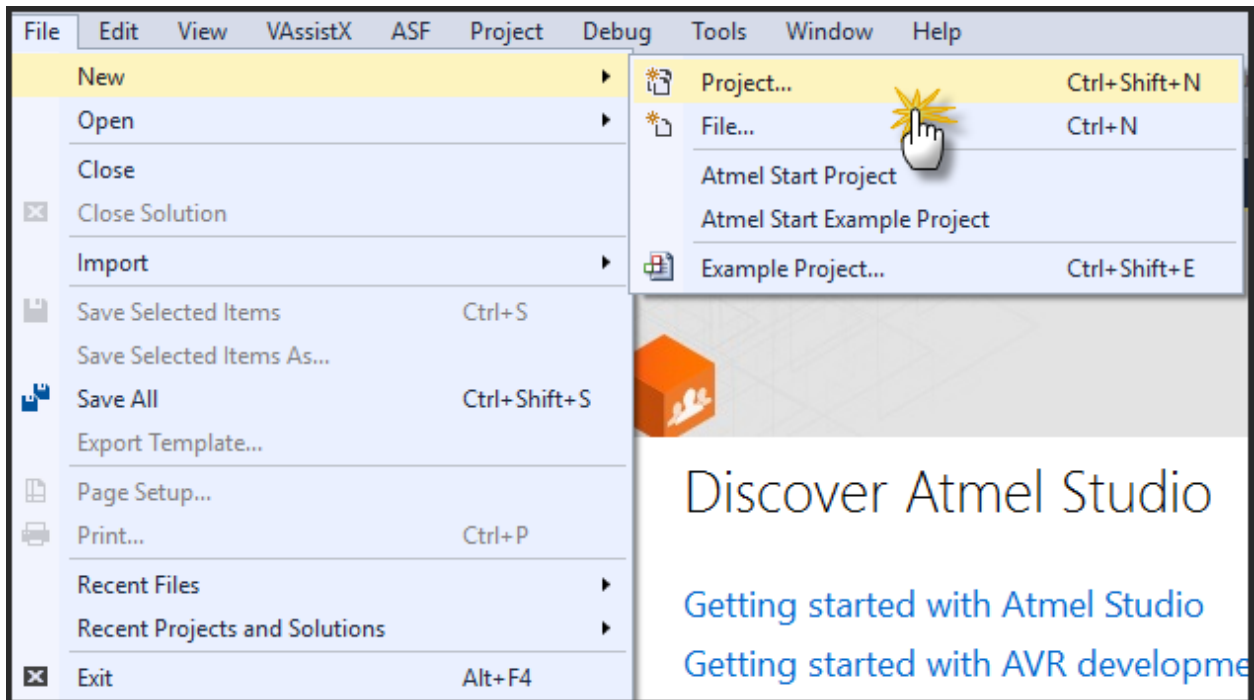
I have uploaded all the files you will need for this tutorial. You can download the files from GitHub here:

<https://github.com/PhillyNJ/SamD21SystemTimerExample>

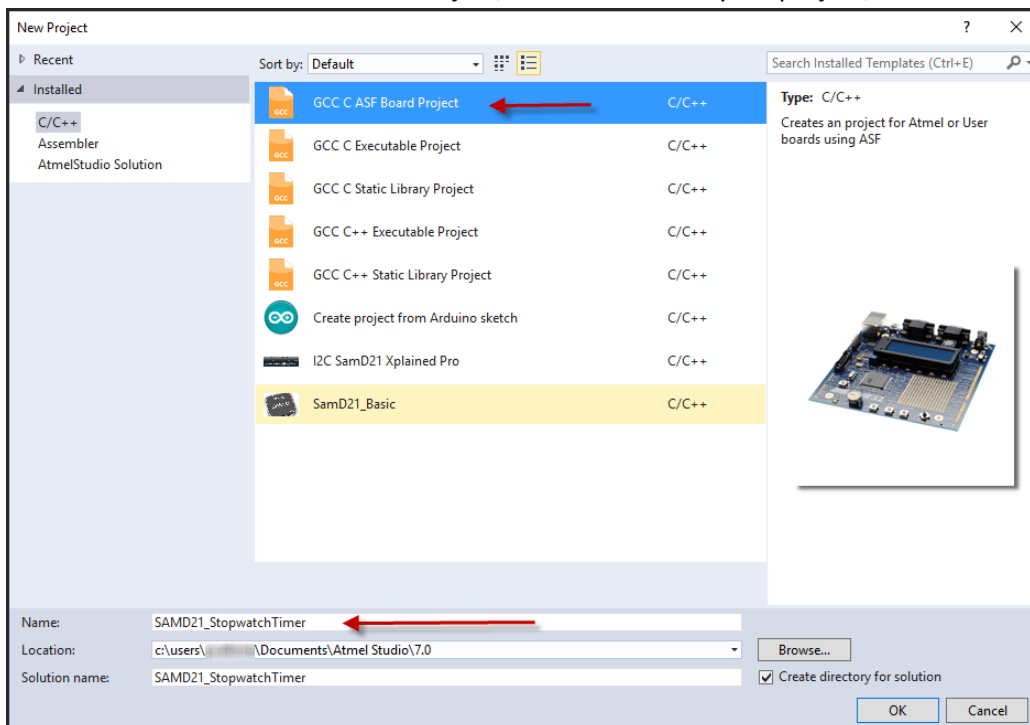
Unzip to a folder of your choice. You will need these files for the tutorial.

Create the Project

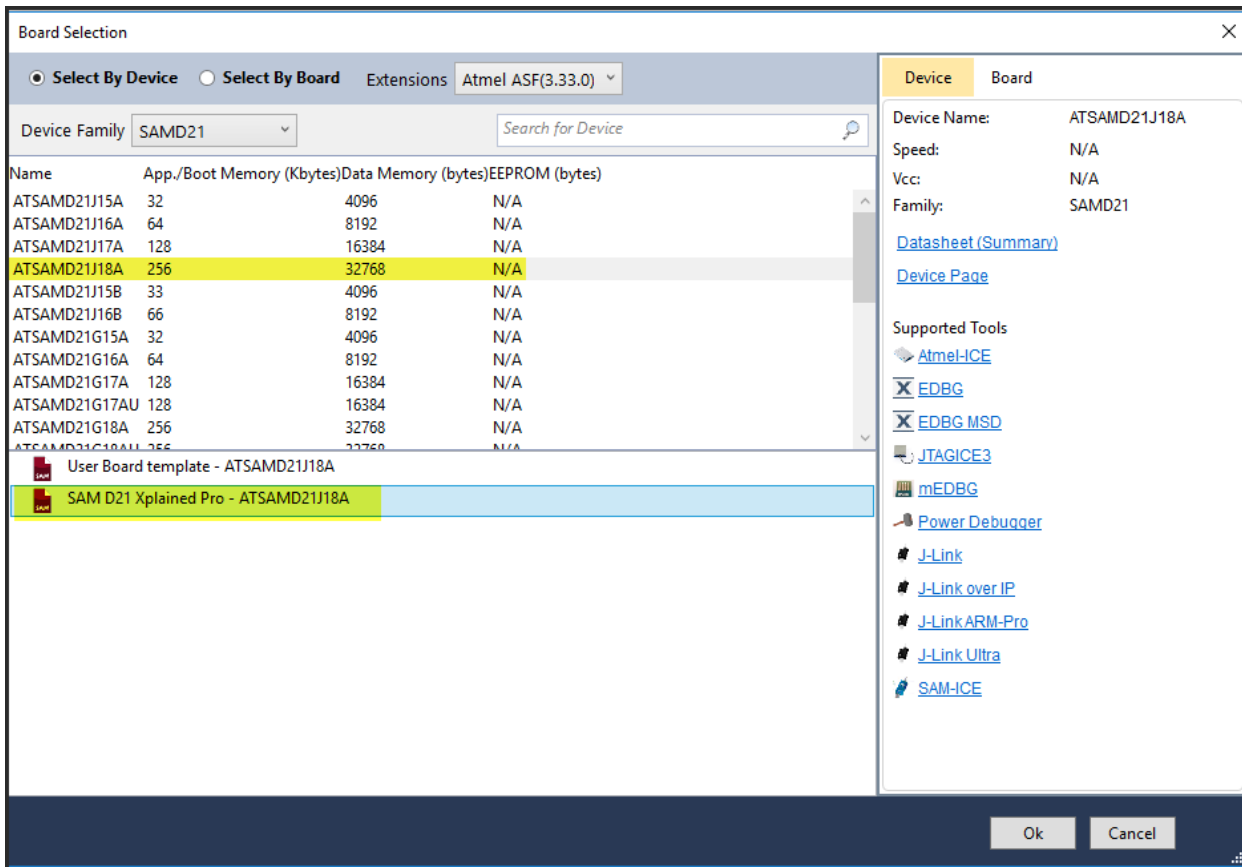
- Click File>New>Project



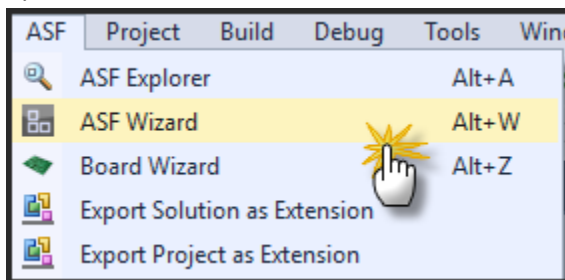
- Choose "GCC C ASF Board Project, Enter a name for your project, click OK



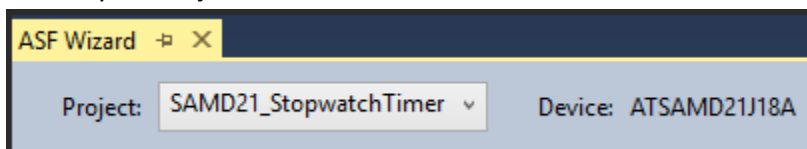
- Select the SAMD21 Device ATSAMD21J18A & SAM D21 Xplained Pro - ATSAMD21J18A board and click Ok



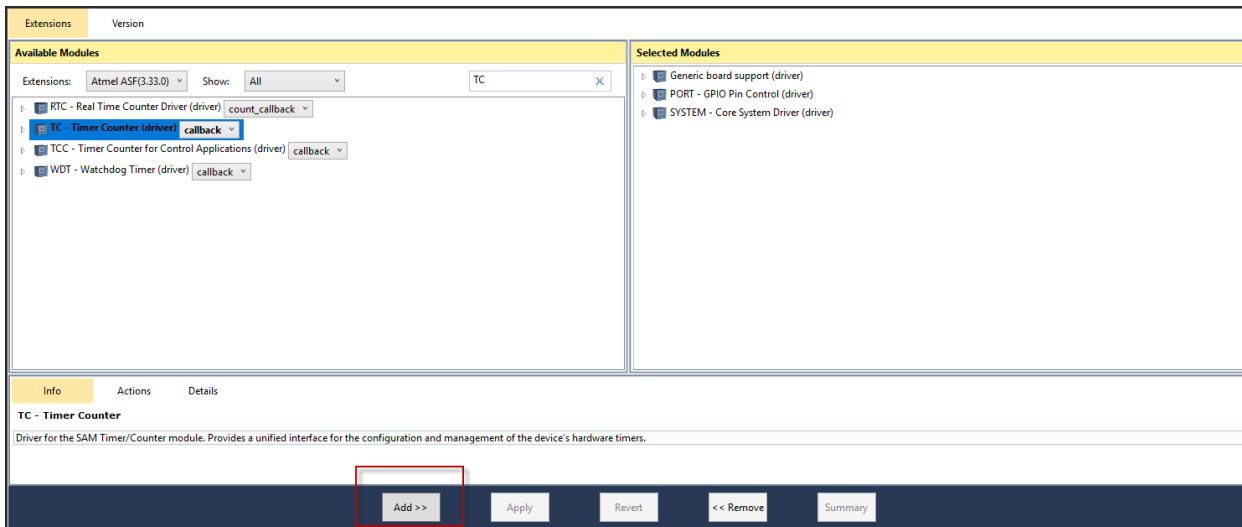
- Open the ASF Wizard



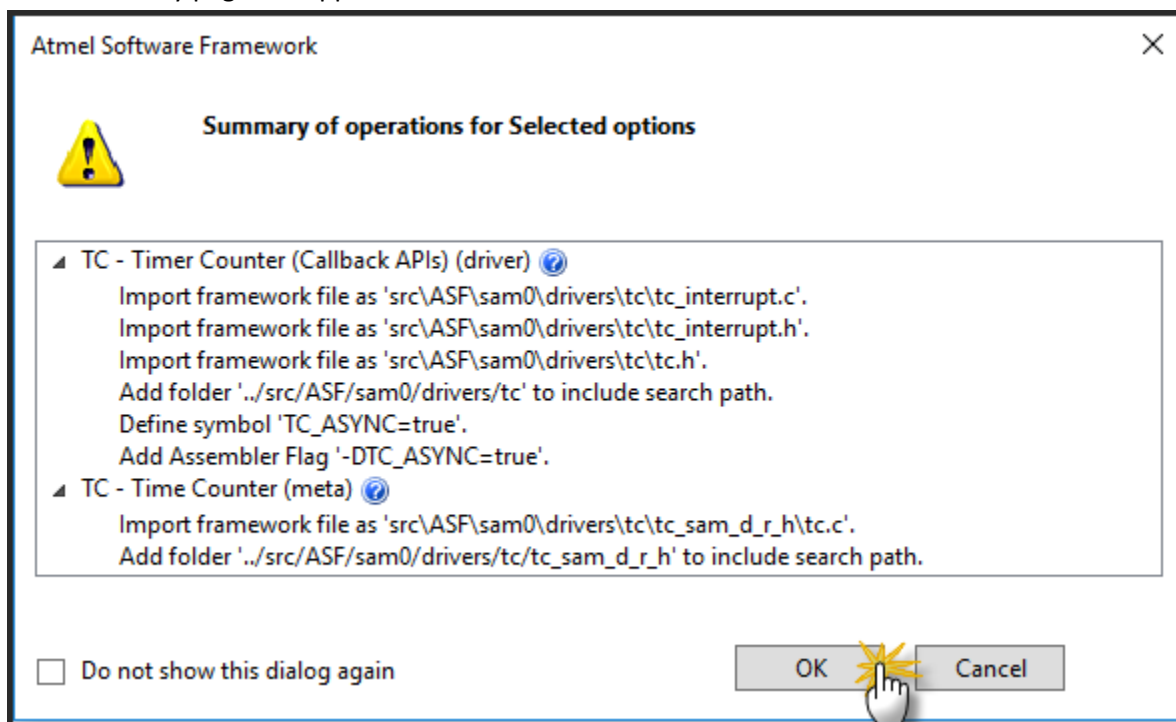
- Choose your Project:



- Add the following modules:
- Choose the TC – Timer Counter Control (driver) Callback and click Add>>

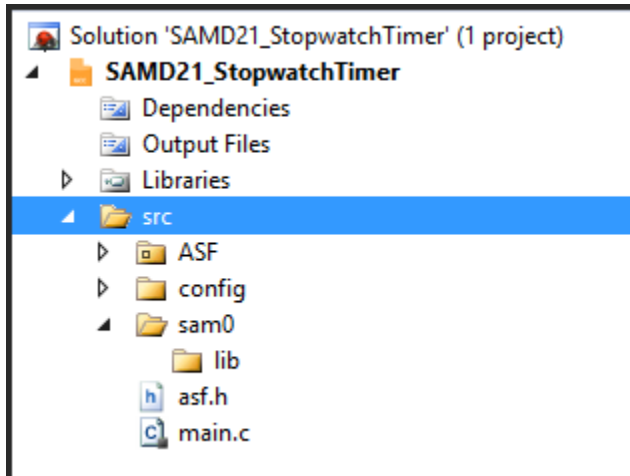


- Click Apply
- A summary page will appear. Click OK

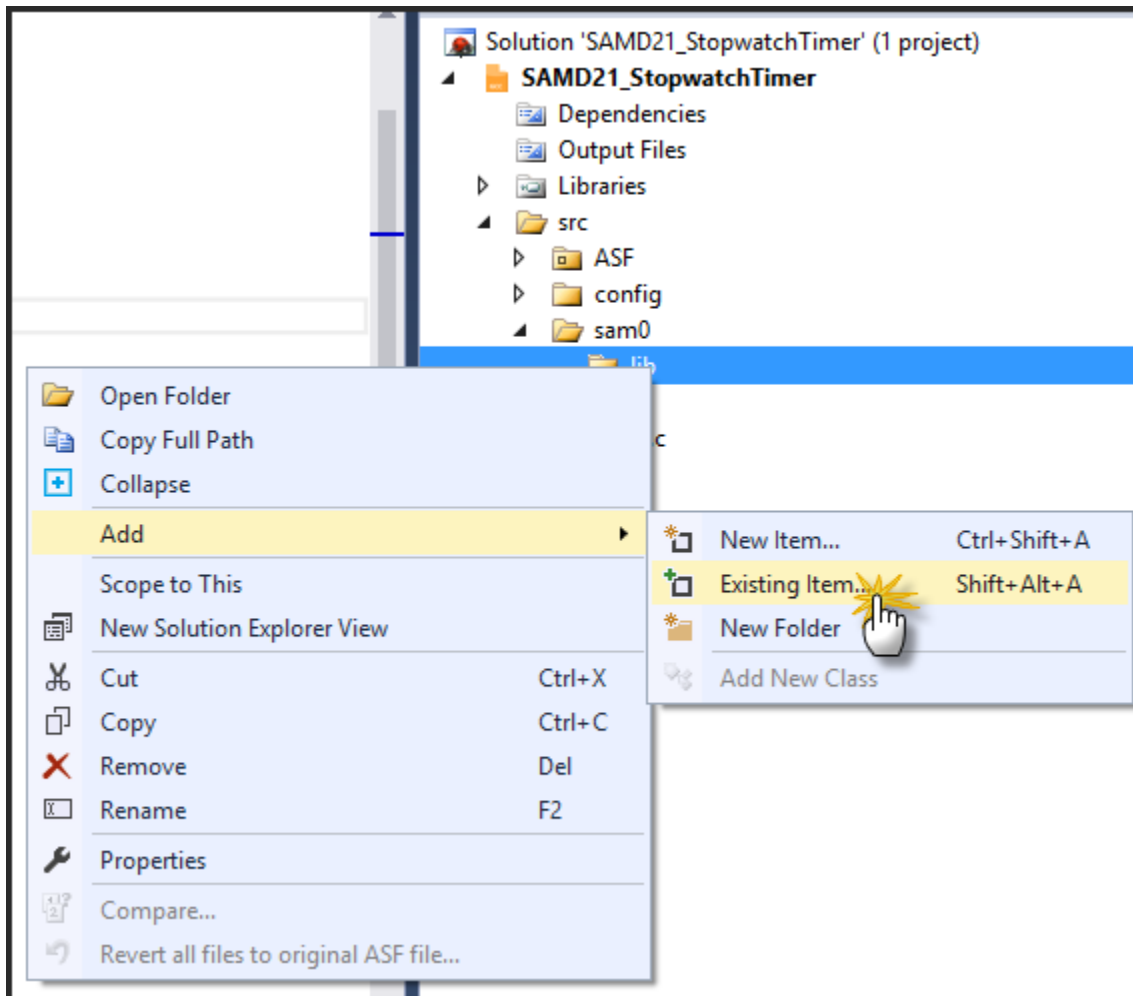


Add the Project Files

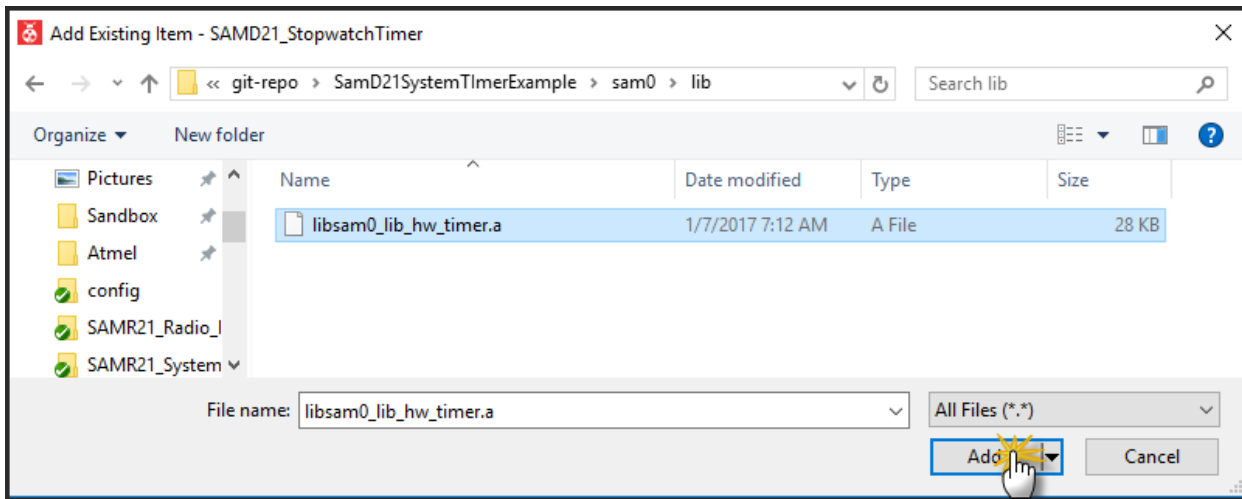
- Create the following folder structure in your project:



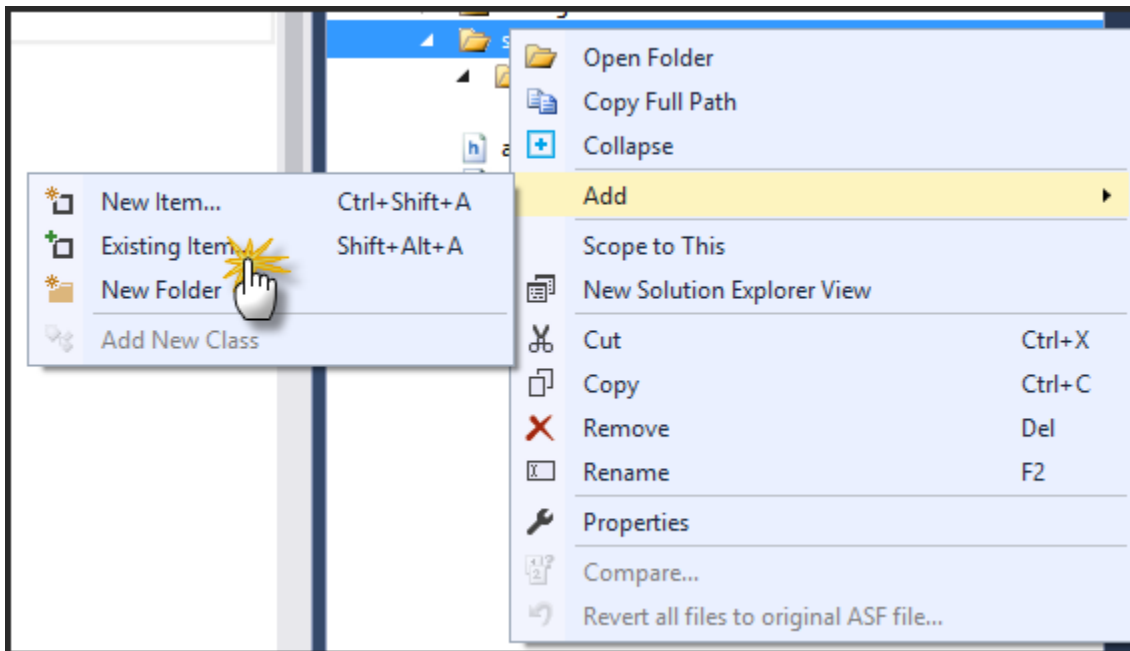
- We are going to add the files I provided from my Git Hub project
- First we need to add the compiled `libsam0_lib_hw_timer.a` library to our project
- Click on the lib folder, and select Add> Existing Item



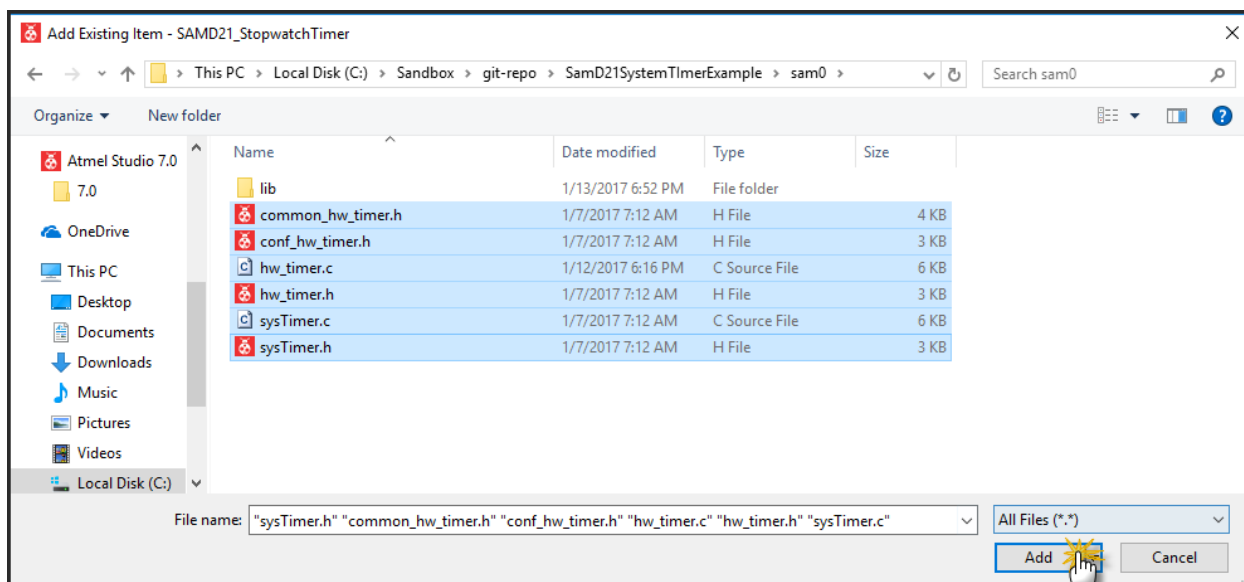
- Navigate to where you unzip the project files.
- Navigate to the `sam0/lib` and select the `libsam0_lib_hw_timer.a` file and click Add



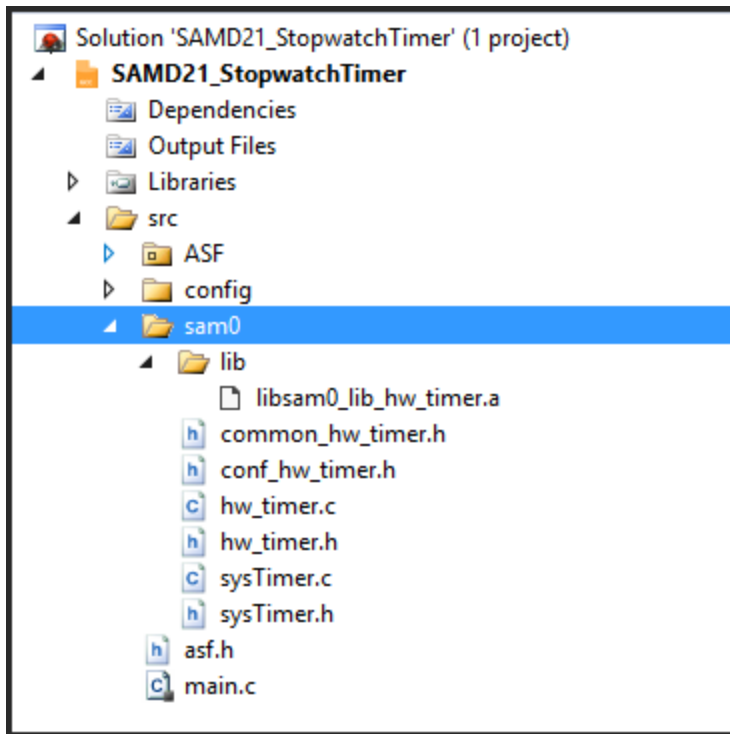
- Right Click on sam0>Add>Existing Item in the project solution



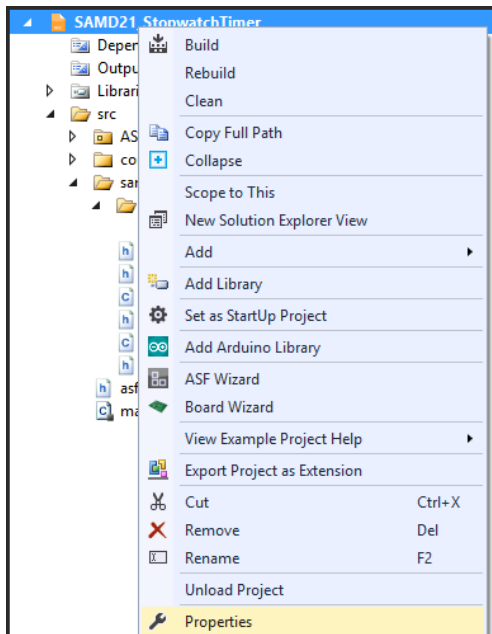
- Add the rest of the files:



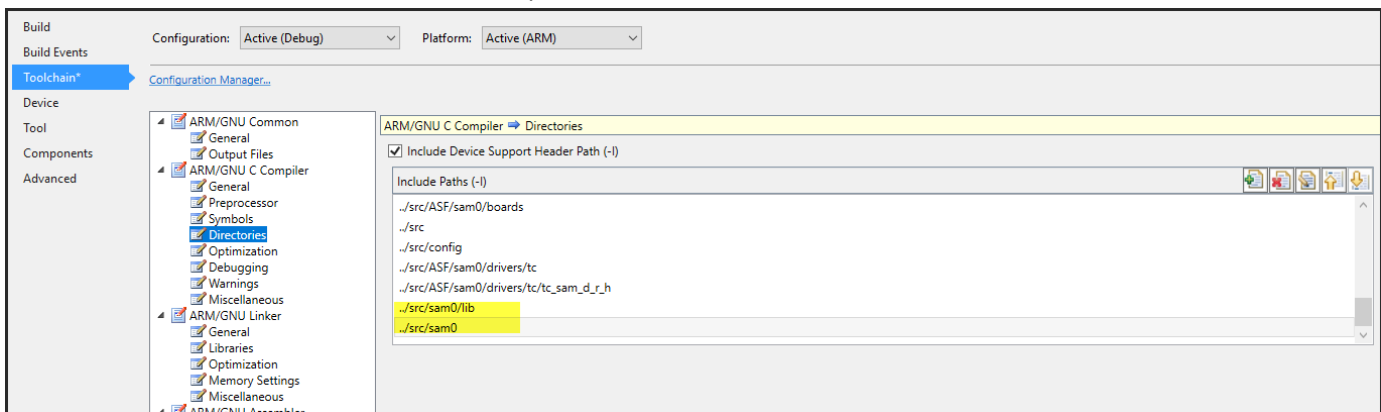
- Your Project should look like this:



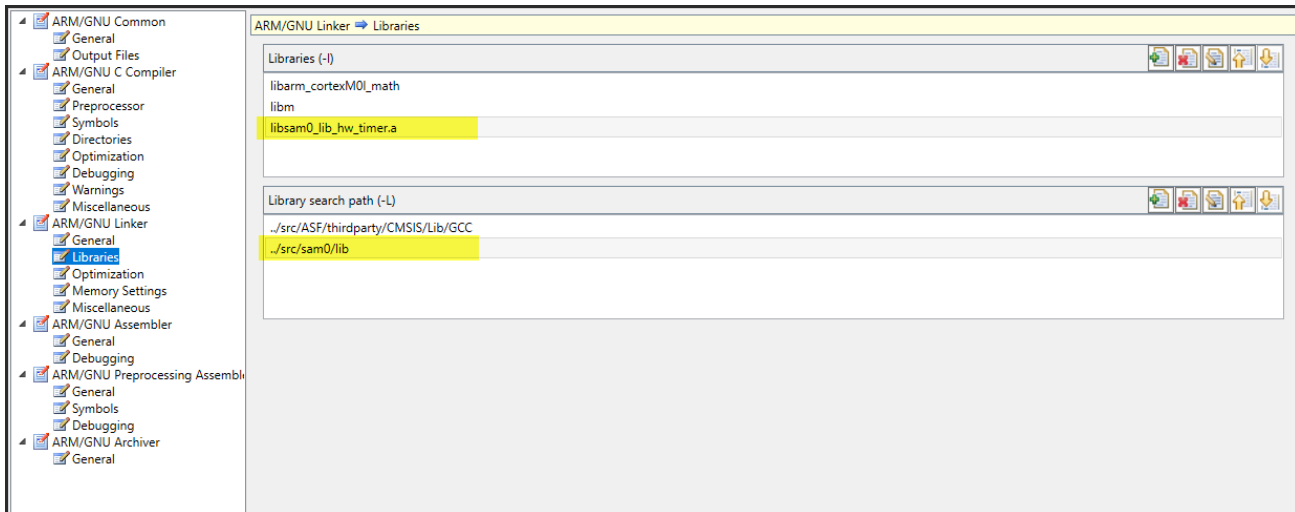
- Now, we need to tell the compiler where these files are located.
- Right click on the Project and click Properties



- Under Tool Chain>ARM/GNU C Compiler, add the 2 directories



- Under ARM/GNU Linker, click Libraries and add the compiled library and Path as shown



Add the Code

Open up the main.c and add the following code

```
#include <asf.h>
#include "sysTimer.h"
static SYS_Timer_t myTimer;
void appTimerTestHandler(SYS_Timer_t *timer);
void initMyTimer(void);
void startMyTimer(void);
void stopMyTimer(void);

int main (void)
{
    system_init();
    SYS_TimerInit();
    initMyTimer();
    startMyTimer();

    while (1) {
        SYS_TimerTaskHandler();
    }
}

void appTimerTestHandler(SYS_Timer_t *timer)
{
    port_pin_toggle_output_level(LED_0_PIN);

    (void)timer;
}

void initMyTimer(void)
{
    myTimer.interval = 1000; // milli seconds
    myTimer.mode = SYS_TIMER_PERIODIC_MODE;
    myTimer.handler = appTimerTestHandler;
}

void startMyTimer(void){

    SYS_TimerStart(&myTimer);
```



```

}

void stopMyTimer(void){

    SYS_TimerStop(&myTimer);

}

```

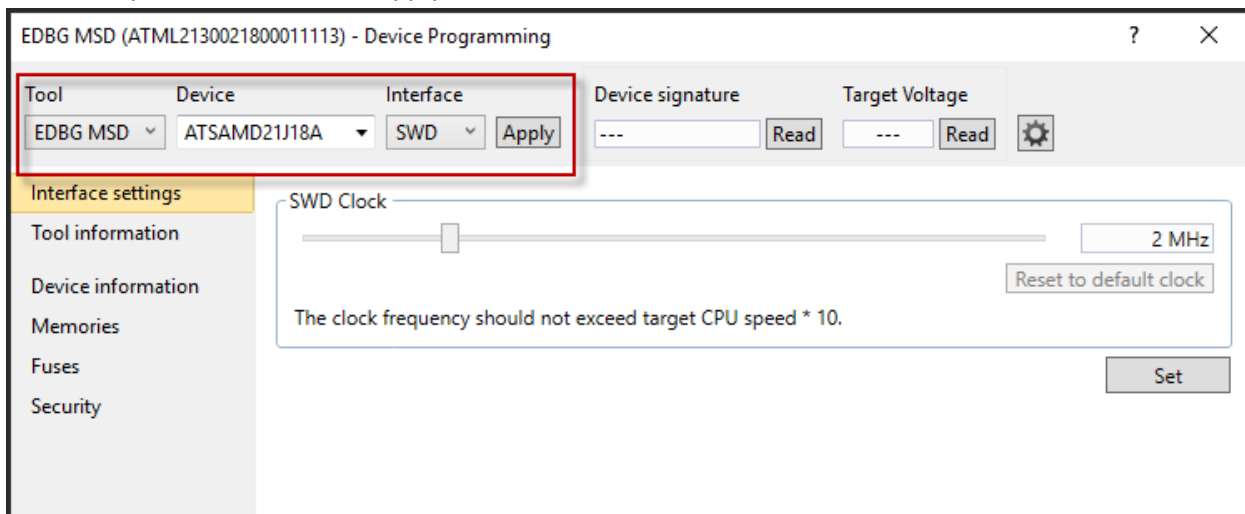
A copy of this code can be found in project files downloaded from github.

Compile and Upload

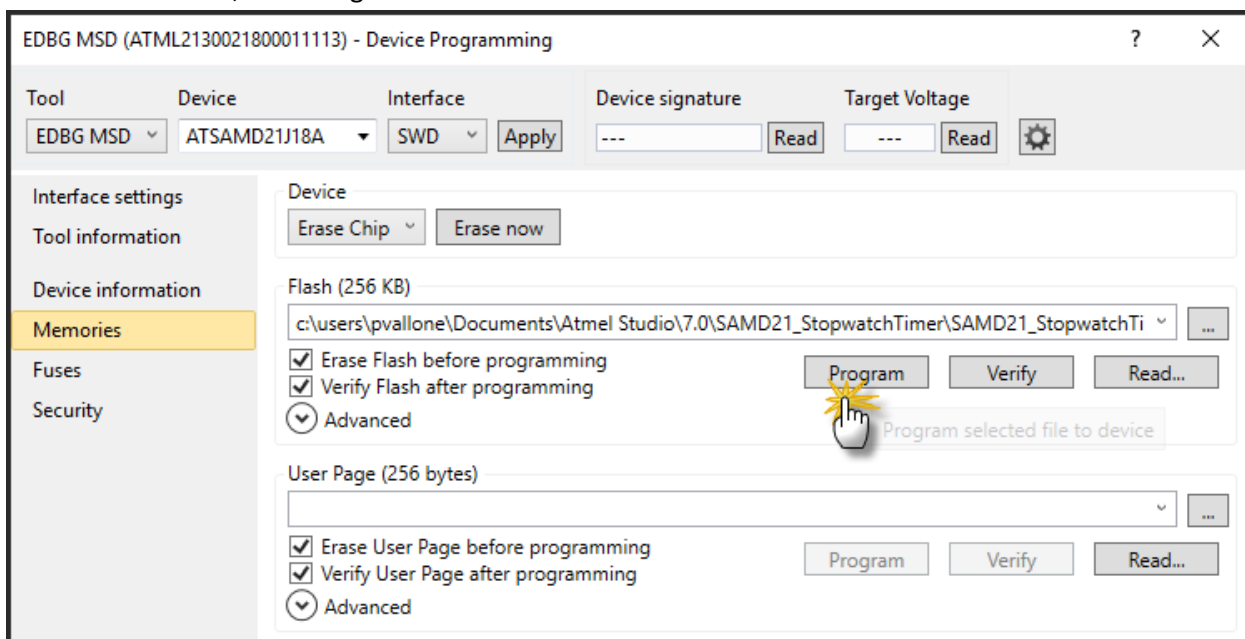
- Press F7 to compile the project
- Click the Device Programming Icon



- Select your Tool, then click Apply



- Click Memories, then Program



The LED on the Samd21 should blink every 1 second. You can change the interval by modifying the following line:

```
myTimer.interval = 1000;
```