Timer Programming

Last updated 6/10/21

Timer Programming

- Timer Operation
  - Nucleo-L476RG has 12 timers
  - It’s not clear how many of these can be used in our implementation

Timer Programming

- Timer Connections
  - There are no connections enabled in the Mbed system

Timer Programming

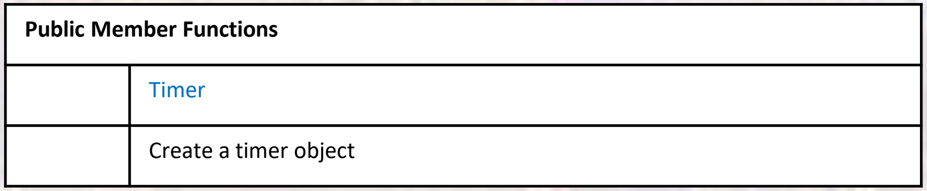
- Timer Class
  - The documentation is missing for this class

Public Member Functions	
	Timer
	Create a timer object
void	start()
	Start the timer
void	stop()
	Stop the timer
void	reset()
	Reset the timer

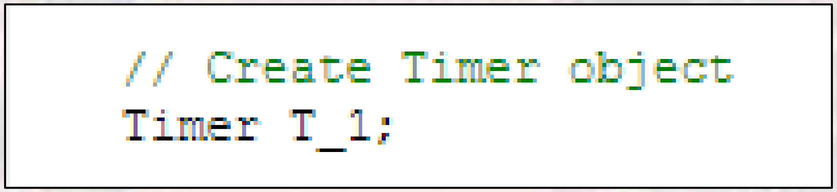
Public Member Functions – From Chrono Library	
int	elapsed_time().count()
	Returns the time on the counter in us (note the return type is int and may truncate)
int	chrono::duration_cast<chrono::milliseconds>
	Cast the time on the counter to ms (note the return type is int and may truncate)
int	chrono::duration_cast<chrono::seconds>
	Cast the time on the counter to s (note the return type is int and may truncate)

Timer Programming

- Constructors



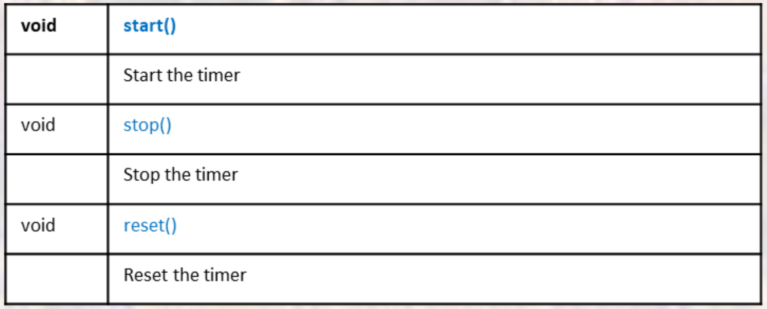
Public Member Functions	
	Timer
	Create a timer object



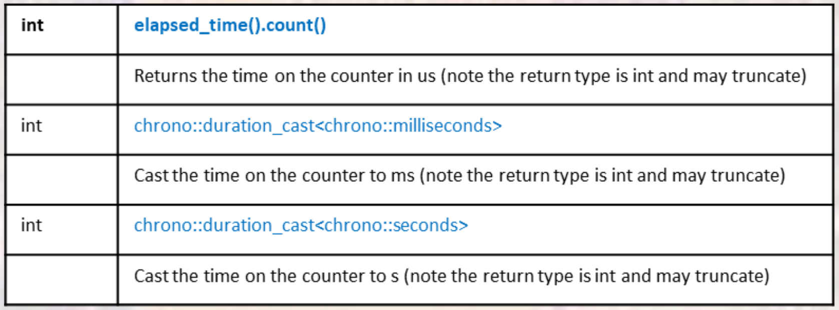
```
// Create Timer object
Timer T_1;
```

Timer Programming

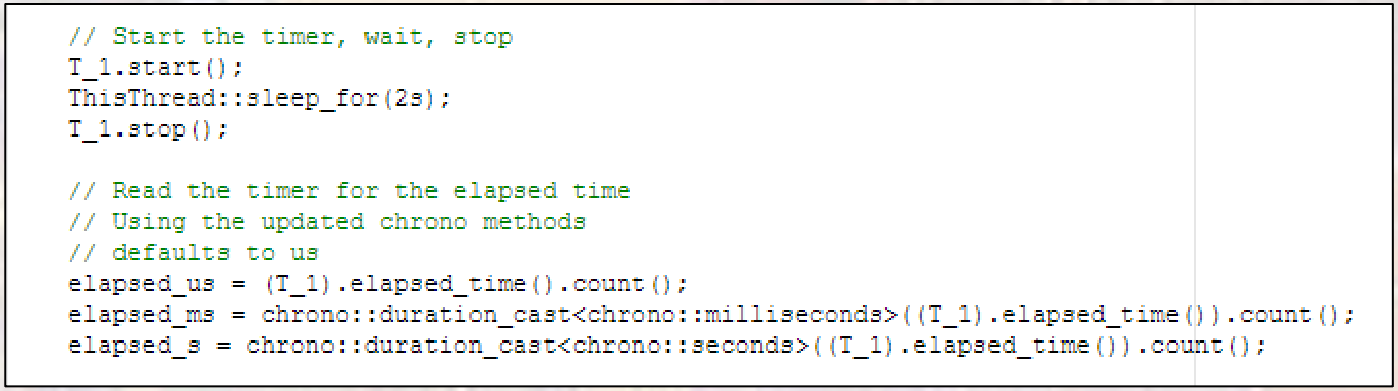
- Member Functions (Methods)



void	**start()**
	Start the timer
void	stop()
	Stop the timer
void	reset()
	Reset the timer



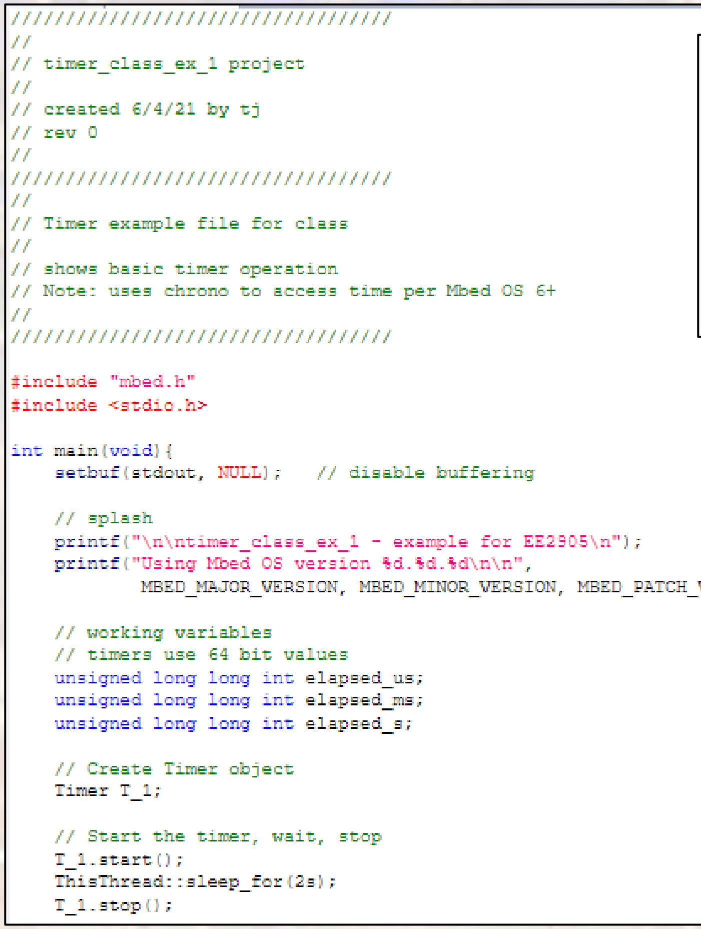
int	**elapsed_time().count()**
	Returns the time on the counter in us (note the return type is int and may truncate)
int	chrono::duration_cast<chrono::milliseconds>
	Cast the time on the counter to ms (note the return type is int and may truncate)
int	chrono::duration_cast<chrono::seconds>
	Cast the time on the counter to s (note the return type is int and may truncate)



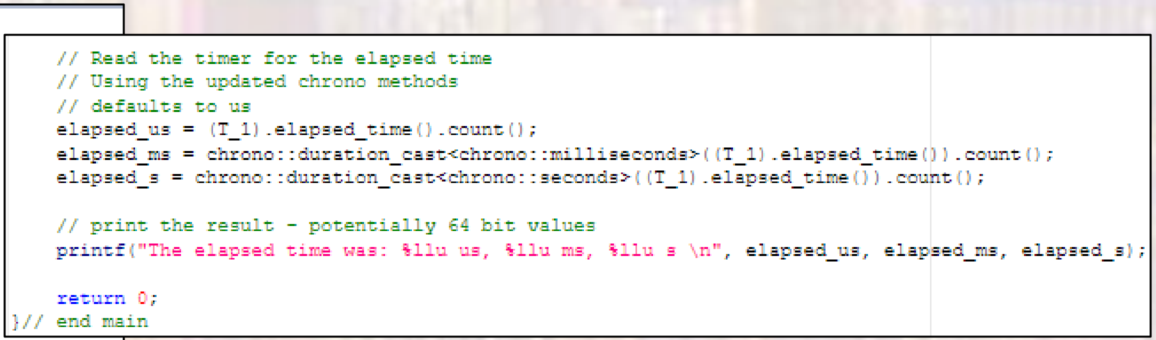
```
// Start the timer, wait, stop
T_1.start();
ThisThread::sleep_for(2s);
T_1.stop();

// Read the timer for the elapsed time
// Using the updated chrono methods
// defaults to us
elapsed_us = (T_1).elapsed_time().count();
elapsed_ms = chrono::duration_cast<chrono::milliseconds>((T_1).elapsed_time()).count();
elapsed_s = chrono::duration_cast<chrono::seconds>((T_1).elapsed_time()).count();
```

Timer Programming

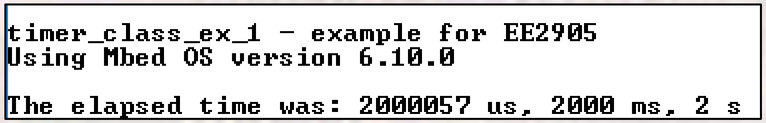
- Simple example 1
  - Measure a known delay



```
////////////////////////////////////
//
// timer_class_ex_1 project
//
// created 6/4/21 by tj
// rev 0
//
////////////////////////////////////
//
// Timer example file for class
//
// shows basic timer operation
// Note: uses chrono to access time per Mbed OS 6+
//
////////////////////////////////////

#include "mbed.h"
#include <stdio.h>

int main(void){
    setbuf(stdout, NULL);    // disable buffering

    // splash
    printf("\n\ntimer_class_ex_1 - example for EE2905\n");
    printf("Using Mbed OS version %d.%d.%d\n\n",
            MBED_MAJOR_VERSION, MBED_MINOR_VERSION, MBED_PATCH_VERSION);

    // working variables
    // timers use 64 bit values
    unsigned long long int elapsed_us;
    unsigned long long int elapsed_ms;
    unsigned long long int elapsed_s;

    // Create Timer object
    Timer T_1;

    // Start the timer, wait, stop
    T_1.start();
    ThisThread::sleep_for(2s);
    T_1.stop();
```



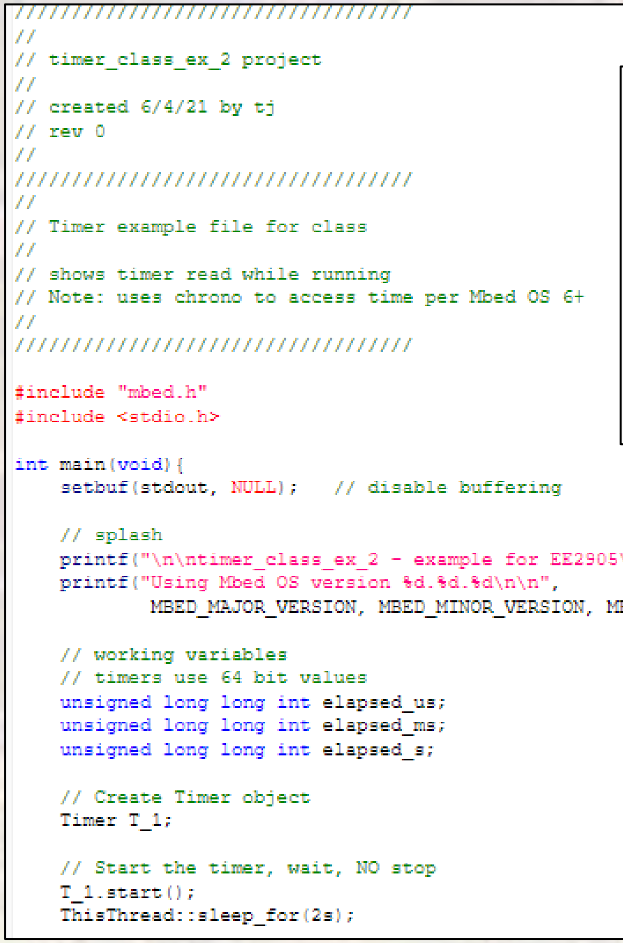
```
    // Read the timer for the elapsed time
    // Using the updated chrono methods
    // defaults to us
    elapsed_us = (T_1).elapsed_time().count();
    elapsed_ms = chrono::duration_cast<chrono::milliseconds>((T_1).elapsed_time()).count();
    elapsed_s = chrono::duration_cast<chrono::seconds>((T_1).elapsed_time()).count();

    // print the result - potentially 64 bit values
    printf("The elapsed time was: %llu us, %llu ms, %llu s \n", elapsed_us, elapsed_ms, elapsed_s);

    return 0;
}// end main
```



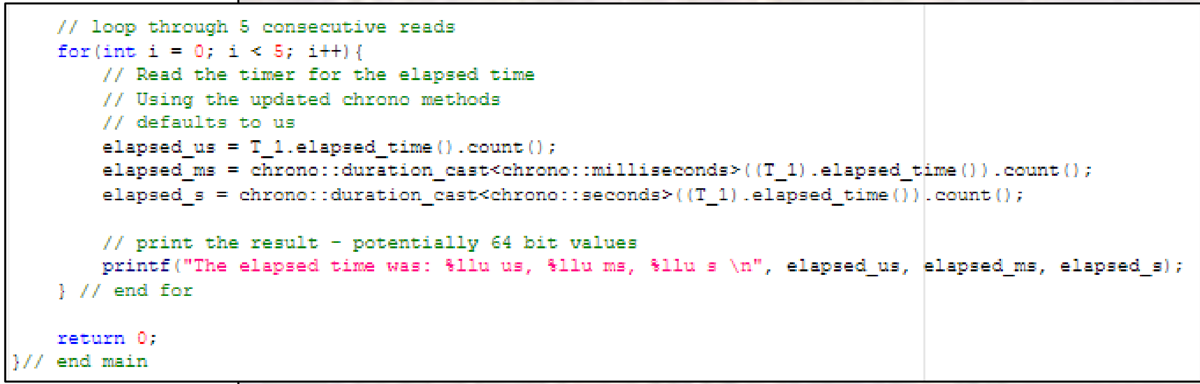
```
timer_class_ex_1 - example for EE2905
Using Mbed OS version 6.10.0

The elapsed time was: 2000057 us, 2000 ms, 2 s
```

Timer Programming

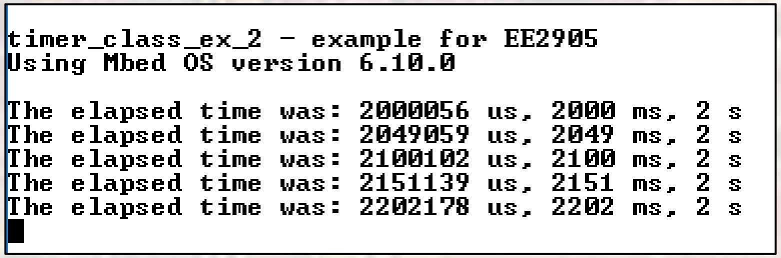
- Simple example 2
    - Measure delay on a running timer (5 measurements)



```
//////////////////////////////////////
//
// timer_class_ex_2 project
//
// created 6/4/21 by tj
// rev 0
//
//////////////////////////////////////
//
// Timer example file for class
//
// shows timer read while running
// Note: uses chrono to access time per Mbed OS 6+
//
//////////////////////////////////////

#include "mbed.h"
#include <stdio.h>

int main(void){
    setbuf(stdout, NULL);   // disable buffering

    // splash
    printf("\n\ntimer_class_ex_2 - example for EE2905\n");
    printf("Using Mbed OS version %d.%d.%d\n\n",
           MBED_MAJOR_VERSION, MBED_MINOR_VERSION, MBED_PATCH_VERSION);

    // working variables
    // timers use 64 bit values
    unsigned long long int elapsed_us;
    unsigned long long int elapsed_ms;
    unsigned long long int elapsed_s;

    // Create Timer object
    Timer T_1;

    // Start the timer, wait, NO stop
    T_1.start();
    ThisThread::sleep_for(2s);
```



```
    // loop through 5 consecutive reads
    for(int i = 0; i < 5; i++){
        // Read the timer for the elapsed time
        // Using the updated chrono methods
        // defaults to us
        elapsed_us = T_1.elapsed_time().count();
        elapsed_ms = chrono::duration_cast<chrono::milliseconds>((T_1).elapsed_time()).count();
        elapsed_s = chrono::duration_cast<chrono::seconds>((T_1).elapsed_time()).count();

        // print the result - potentially 64 bit values
        printf("The elapsed time was: %llu us, %llu ms, %llu s \n", elapsed_us, elapsed_ms, elapsed_s);
    } // end for

    return 0;
}// end main
```



```
timer_class_ex_2 - example for EE2905
Using Mbed OS version 6.10.0

The elapsed time was: 2000056 us, 2000 ms, 2 s
The elapsed time was: 2049059 us, 2049 ms, 2 s
The elapsed time was: 2100102 us, 2100 ms, 2 s
The elapsed time was: 2151139 us, 2151 ms, 2 s
The elapsed time was: 2202178 us, 2202 ms, 2 s
█
```

Timer Programming

- Limitations summary

  - Minimum measurable timer tick is 1us
    - Return from our elapsed_time().count() method is int