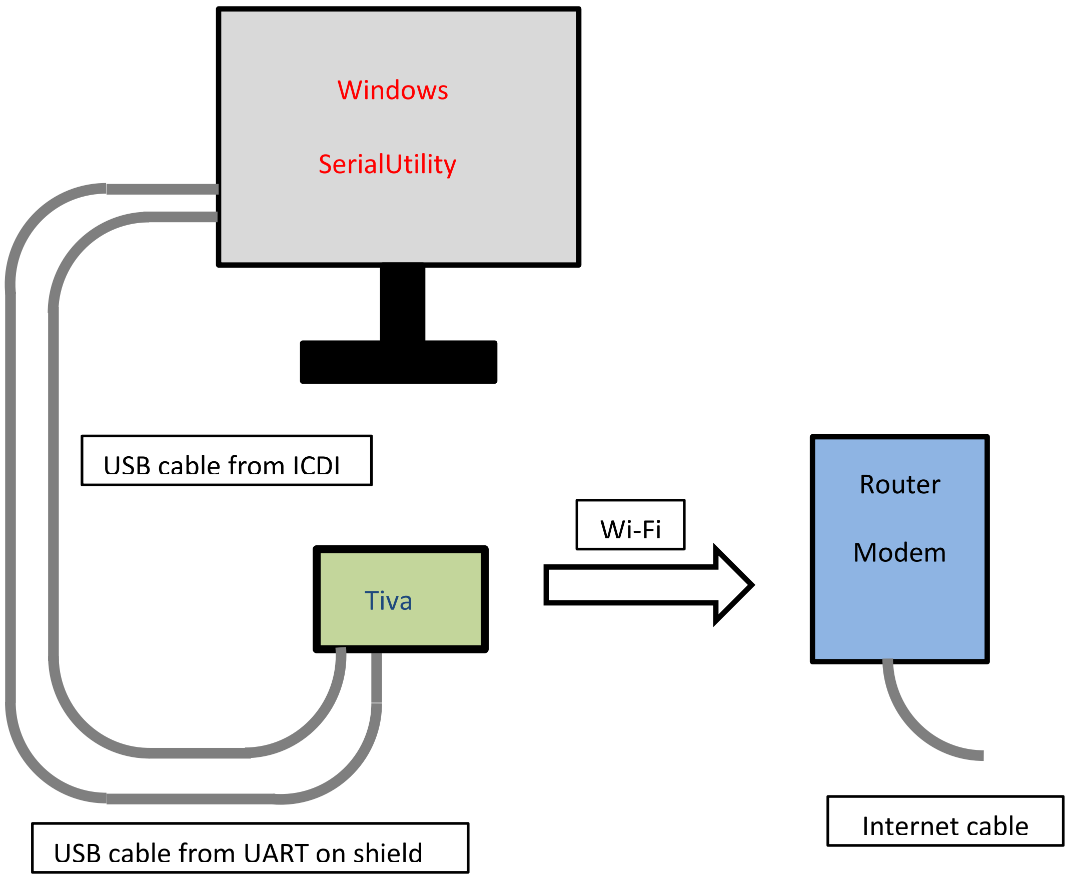
When I bought the Tiva board, I set out to write code for the Wi-Fi module included in the design. I could not make it work even with the MikroE example provided for the Tiva peripherals. I finally found the Wifi CC3000 Camera Client project from the MikroElektronika Team and was able to see how to implement a TCP client using the MikroE library functions.

This simple program makes a connection to my router, then establishes a TCP client connection, and finally makes a request to a time server for the current date and time. This value is then set into the real-time-clock (RTC) registers of the TI microcontroller.

The hardware is set up as follows:



The real-time-clock in this microcontroller can operate in two modes. In counter mode, a 32-bit register is incremented each second. The value in that register must be interpreted by software to derive the date and time. The code for this mode is included but is not used in this example.

In calendar mode, the RTC registers offer day-of-week, year, month, and day fields for the date and hour, minute second fields for the time. The hardware takes care of incrementing each field especially the variable month days and leap year compensation.

The code implements two state machines, one for managing the external CC3000 chip, and one for managing the internal RTC circuits. The benefit of using a state machine is to make the code non-blocking so that other tasks can run concurrently. If the state machine tasks are waiting for some event to happen, the UART can still get cycles to put out diagnostic messages. This makes for easier debugging if a task stalls.

UART1 is used to evoke action in the code through simple commands. There a quite a few diagnostic commands implemented but the important ones that demonstrate what the project is about are ‘o’ and ‘p’.

Command ‘o’ starts the RTC state machine and partly through the sequence, it starts the CC3000 state machine. RTC SM waits on a flag indicating that valid date/time data has been received from the time server and then proceeds to store the data into the calendar registers. Since the time from obtaining date/time from the server to setting the RTC is very small, the accuracy of the clock is extremely good.

Command ‘p’ can be used any time to read out the calendar registers.