



# Getting Started with the Timex USB Datalink SDK

---

## Introducing the Timex USB Datalink Wristwatch

The Timex USB Datalink Wristwatch, known internally by Timex as the model M851, features a watch platform that can be configured by a PC and can upload and download through a standard USB port. The watch has 14 slots available for applications which you can choose according to your requirements. Want a business watch for the week? Configure the watch with appointment and contact applications. Running this weekend? A choice of a 200-lap chronograph and an interval timer can make the event worthwhile. No need to write down the stored laps and manually enter the data in the computer. The upload features of the PIM can do that for you.

The watch provides built-in applications that address the needs of general watch users for business and sports. Examples would be : Chronograph, Countdown Timer, Interval Timer, Synchro Timer, Alarm, Appointments, Occasion, Contacts (phonebook), Schedule and Notes. WristApps (custom applications you can download to the watch) are also available: digital counter, pulse, demo, games, and world time viewer.

## But, I would like the watch to do this...

Don't like the standard PIM? Do you want to customize the configuration software to grab data from other sources like the internet? Do you need a specific watch based application to serve the needs of your clientele? Do you want to create a stand alone application that configures the watch for a specific purpose?

The Timex USB Datalink Software Development Kit (SDK) provides four solutions to customizing the Datalink system.

- **WristApps.** Design custom watch applications using assembly language.
- **WristApp Configuration Plug-In.** Use the built-in watch application and provide a custom PC user interface to grab data from any source and convert it to the watch database format.
- **Data Import.** Store the data into a comma-delimited format and import it using the standard configuration interface of the built-in applications.
- **PIM.** Create a custom PC application interface between the user and the watch. Program with Visual C++ or Visual Basic.

## Tell me more about designing WristApps...

Writing custom WristApps gives you the most flexible solution for your specific watch-based requirements. You need to have the following:

- Timex USB Datalink Watch;
- PC with Windows OS and a USB port;
- Text file editor;
- Experience in writing assembly language and debugging code;
- Patience to read through some thick documents.

## Let's build the sample Counter WristApp...

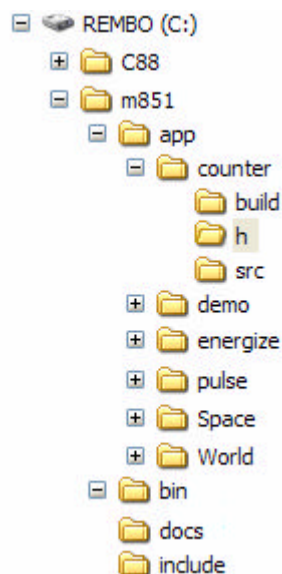
This will guide you to building the Counter WristApp source files provided in the WristApp SDK directory.

1. **Installation.** You need to do this only once. Download the WristApp SDK Installer and execute it. This will setup the required directories and utilities required to build the WristApp. This also installs the sample application that we will be building in this guide.

The WristApp SDK files are located in the root directory at C:\M851.

2. **Coding the WristApp.** Actually, the sample Counter WristApp is already coded, debugged and tested. The design process from start to finish is detailed in the document "M851 Counter WristApp Design Guide.pdf".

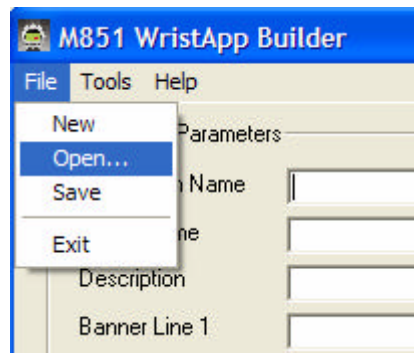
The source files are located in the directory: C:\M851\APP\COUNTER. Open up file explorer to the directory and you will see three directories: header directory (H), source directory (SRC) and the build directory (BUILD). Take a look at the files using your preferred text editor. Do not make any modifications to the files.



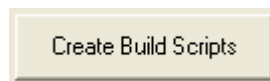
3. **Run the WristApp Builder.** Now, we will convert the source files that can be downloaded to the watch. Using file explorer, proceed to the directory C:\M851\BIN. Double-click on the file M851 WristApp Builder.exe.



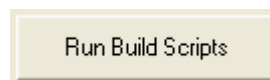
The information required for the WristApp builder to create the Counter WristApp is located in the file C:\m851\app\counter\build\counter.scr. Populating the entries in this utility is described in the document "M851 Counter WristApp Design Rev 1\_0.pdf". So let's open the builder file and start compiling. Click on the menu item File\Open... and browse through the directories until you find COUNTER.SCR, select the file and then click OK.



4. **Creating the Build Scripts.** Click on the button "Create Build Scripts". This will create the DOS batch files that will assemble and link the source files. Once the scripts are generated, this step is not necessary, unless there are modifications made to the source file setup section.



5. **Executing the Build Scripts.** Click on the button "Run Build Scripts". This will open up a DOS command window that executes the build scripts. This may take some time to complete. Be patient. Once completed, the DOS command window will close.



6. **Create the WristApp files.** Click on the button "Create WristApp". This will create all the WristApp files that will be used for downloading the application to the watch.



7. **Download the WristApp.** Click on the button "Download WristApp" to open a utility that will download the WristApp to the watch. Connect the watch to the USB port. When the watch display shows "COMM READY", click the "Download" button. When download is complete, disconnect

the USB cable. The watch will go back to Time-of-Day mode. Press the MODE switch to proceed to the next mode, which should be the Counter mode.

WristApp Downloader

## Recommended WristApp reading...

**WristApp Design Guide.** This guide provides an overview of the USB Datalink watch OS and how it processes system events to be passed to the applications for further processing. This document details how WristApps can make use of the system resources and API to perform custom operations. [m851 WristApp Design Guide Rev 1\_0.pdf]

**Counter WristApp Design Guide.** A step-by-step guide to build the Counter WristApp – from design specification, source code creation, compilation of the source files and creating the files required to download the WristApp to the watch. [M851 Counter WristApp Design.pdf]

**WristApp API Reference Guide.** A list of all the routines provided by the watch OS. [M851 WristApp API Reference Guide.pdf]

**WristApp GUI Plug-In Design Guide.** A guide that demonstrates how to create a DLL for editing/viewing WristApp data via a PIM running on a PC. This guide illustrates installing the DLL, creating the DLL, and interfacing with the PIM software.

**Periodic Task Design Guide.** A step-by-step guide to creating a periodic task – code modules that are attached to the Time-Of-Day application and executed at regular intervals. A sample periodic task is discussed that blanks out the display after a minute rollover. [M851 Periodic Task Design.pdf]

**Power Consumption Guide.** A document that shows some possible WristApp software design considerations that can minimize current drain on the watch battery. [M851 Power Consumption Consideration.pdf]

**Terminate and Stay Resident Code.** A guide to developing a special type of periodic task that plugs into the built-in Time-of-Day main processing code to enhance TOD functionality. A sample code shows how to tap into the update events passed to the TOD application to be manipulated during a minute update. [M851 Terminate And Stay Resident Code for TOD mode.pdf]

**Microcontroller Instruction Set.** Assembly language is required to write the WristApp code. So the following document is a must read to get to know the microcontroller used in the watch, the Epson EOC88349. [s1c88 core cpu manual.pdf] Other information on the microcontroller can be found in the following documents: s1c88349 technical manual.pdf, s1c88 tool package manual-1.pdf, and s1c88 tool package manual-2.pdf.

## Sample WristApps with source code...

**Counter.** Counts up (or down) the number of switch depressions. This serves as a general reference for WristApp design. This WristApp shows some basic operations of an application: changing the

foreground state handler, changing to the next mode, handling the crown, setting and blinking operations.

**TimeWarp.** A sample terminate-and-stay-resident code. A code plug-in for the built-in TOD application to enhance display updates during minute rollovers.

**World Time.** Displays the 24 time zones specified in a PC configuration application. This shows how the database access APIs are used. Scrolling APIs are also used in this application. The PC interface to configure the database for download is included.

**Invasion.** Battle alien ships before they reach the point of no return! This sample WristApp shows the use of the primitive pixel display APIs. This application can serve as a reference to building state handlers that require more code space than the overlay area.

**Blank.** A sample periodic task that functions like a screen saver. The routine is executed every minute rollover to clear the display. It will restore the display after any user switch input.

**Demo.** A Demo application that shows the feature set of the watch used for marketing purposes.

## And for the PC software developer...

The following documents demonstrate how to create your own PIM or specific PIM components using Visual C++ or Visual Basic to communicate and configure the watch.

**WristApp PIM Plug-in Design Guide.** A guide that demonstrates how to create a DLL for editing/viewing WristApp data via the PIM running on the PC. This guide illustrates installing the DLL, creating the DLL, and interfacing with the PIM software. [WristApp PIM Plugin Design Guide.pdf]

**Packager API Design & Usage Guide.** An instructional text that shows how to create Windows based programs that interface directly with the watch. PIM developers may use this document for help in integrating their software with the USB Data Link watch. [Packager API Design Guide.pdf]