

# MultiStopwatch

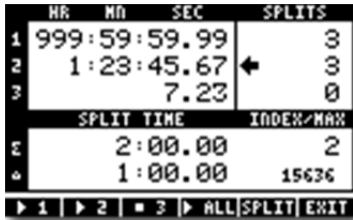
|                              | HR      | MM | SEC       | SPLITS |
|------------------------------|---------|----|-----------|--------|
| 1                            | 999     | 59 | 59.99     | 3      |
| 2                            | 1       | 23 | 45.67     | 3      |
| 3                            |         |    | 7.23      | 0      |
| SPLIT TIME                   |         |    | INDEX/MAX |        |
| Σ                            | 2:00.00 |    |           | 2      |
| Δ                            | 1:00.00 |    |           | 15636  |
| ▶ 1 ▶ 2 ■ 3 ▶ ALL SPLIT EXIT |         |    |           |        |

Current Version: 1.0.0  
Released: August 4, 2013

**MultiStopwatch** is a free library for the **HP 49g+/50g** that provides the basic functionality of a standard stopwatch, but with three independent timers instead of one. The timers are manipulated through the use of a console application (pictured above), or alternatively through the use of programmable functions that can be executed from UserRPL code as needed. **This library is only compatible with the HP 49g+ and 50g calculators.** If you have any feedback, please send email to [msw@apmats.com](mailto:msw@apmats.com).

**If English isn't your preferred language**, I'd be happy to compile a language-specific version for you. All I ask is that you agree to freely share your translation. All of the language-specific strings are in a single file that I can send if you're interested. Inquire at the above email address.

## Features



A screenshot of the calculator's stopwatch and split screen. The top section shows a list of splits with columns for lap number (1-3), time (999:59:59.99, 1:23:45.67, 7.23), and splits (3, 3, 0). The bottom section shows split time (2:00.00, 1:00.00) and index/max (2, 15636). Navigation buttons at the bottom include 1, 2, 3, ALL, SPLIT, and EXIT.

| WR | NO           | SEC | SPLITS |
|----|--------------|-----|--------|
| 1  | 999:59:59.99 |     | 3      |
| 2  | 1:23:45.67   |     | 3      |
| 3  | 7.23         |     | 0      |

| SPLIT TIME |  | INDEX/MAX |
|------------|--|-----------|
| 2:00.00    |  | 2         |
| 1:00.00    |  | 15636     |

1 2 3 ALL SPLIT EXIT

A console application which shows all timer and split information, with real-time running status and easy browsing capability for splits



A screenshot of the calculator's stopwatch and split screen. The top section shows a list of splits with columns for lap number (1-3), time (1:05:27.98, 1:01:37.00, .00, .00), and splits (0, 0, 0, 0). The bottom section shows split time (.00, .00) and index/max (0, 15636). Navigation buttons at the bottom include 1, 2, 3, ALL, SPLIT, and EXIT.

| WR | NO         | SEC | SPLITS |
|----|------------|-----|--------|
| 1  | 1:05:27.98 |     | 0      |
| 2  | 1:01:37.00 |     | 0      |
| 3  | .00        |     | 0      |
| 4  | .00        |     | 0      |

| SPLIT TIME |  | INDEX/MAX |
|------------|--|-----------|
| .00        |  | 0         |
| .00        |  | 15636     |

1 2 3 ALL SPLIT EXIT

All timer data is saved between invocations of library commands, so the timers continue to "run" while other operations are performed



A screenshot of the calculator's stopwatch and split screen. The top section shows a list of splits with columns for lap number (1-4), time (TEAM1, TEAM2, TEAM3), and splits (0, 0, 0, 0). The bottom section shows split time (TEAM1, TEAM2, TEAM3) and index/max (0, 15636). Navigation buttons at the bottom include 1, 2, 3, ALL, SPLIT, and EXIT.

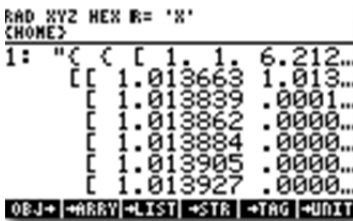
| WR | NO    | SEC | SPLITS |
|----|-------|-----|--------|
| 1  | TEAM1 |     | 0      |
| 2  | TEAM2 |     | 0      |
| 3  | TEAM3 |     | 0      |
| 4  |       |     | 0      |

| SPLIT TIME |  | INDEX/MAX |
|------------|--|-----------|
| TEAM1      |  | 0         |
| TEAM2      |  | 0         |
| TEAM3      |  | 0         |

1 2 3 ALL SPLIT EXIT

Multiple stopwatch data files can be stored, allowing many active sets to be simultaneously maintained



A screenshot of the calculator's stopwatch and split screen. The top section shows a list of splits with columns for lap number (1-10), time (1.013663, 1.013839, 1.013862, 1.013884, 1.013905, 1.013927), and splits (6.212, .0001, .0000, .0000, .0000, .0000). The bottom section shows split time (1.013663, 1.013839, 1.013862, 1.013884, 1.013905, 1.013927) and index/max (6.212, .0001, .0000, .0000, .0000, .0000). Navigation buttons at the bottom include 1, 2, 3, ALL, SPLIT, and EXIT.

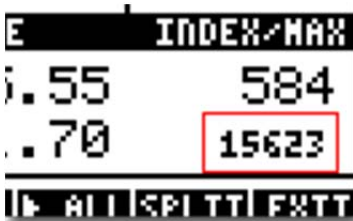
| WR | NO       | SEC | SPLITS |
|----|----------|-----|--------|
| 1  | 1.013663 |     | 6.212  |
| 2  | 1.013839 |     | .0001  |
| 3  | 1.013862 |     | .0000  |
| 4  | 1.013884 |     | .0000  |
| 5  | 1.013905 |     | .0000  |
| 6  | 1.013927 |     | .0000  |

| SPLIT TIME |  | INDEX/MAX |
|------------|--|-----------|
| 1.013663   |  | 6.212     |
| 1.013839   |  | .0001     |
| 1.013862   |  | .0000     |
| 1.013884   |  | .0000     |
| 1.013905   |  | .0000     |
| 1.013927   |  | .0000     |

1 2 3 ALL SPLIT EXIT

Data export is available for moving timer and split data to other applications



A screenshot of the calculator's stopwatch and split screen. The top section shows a list of splits with columns for lap number (1-2), time (1.55, .70), and splits (584, 15623). The bottom section shows split time (1.55, .70) and index/max (584, 15623). Navigation buttons at the bottom include 1, 2, 3, ALL, SPLIT, and EXIT.

| WR | NO   | SEC | SPLITS |
|----|------|-----|--------|
| 1  | 1.55 |     | 584    |
| 2  | .70  |     | 15623  |

| SPLIT TIME |  | INDEX/MAX |
|------------|--|-----------|
| 1.55       |  | 584       |
| .70        |  | 15623     |

1 2 3 ALL SPLIT EXIT

Split information is limited only by available memory in the calculator



A screenshot of the calculator's stopwatch and split screen. The top section shows a list of splits with columns for lap number (1-10), time (1.013663, 1.013839, 1.013862, 1.013884, 1.013905, 1.013927), and splits (6.212, .0001, .0000, .0000, .0000, .0000). The bottom section shows split time (1.013663, 1.013839, 1.013862, 1.013884, 1.013905, 1.013927) and index/max (6.212, .0001, .0000, .0000, .0000, .0000). Navigation buttons at the bottom include 1, 2, 3, ALL, SPLIT, and EXIT.

| WR | NO       | SEC | SPLITS |
|----|----------|-----|--------|
| 1  | 1.013663 |     | 6.212  |
| 2  | 1.013839 |     | .0001  |
| 3  | 1.013862 |     | .0000  |
| 4  | 1.013884 |     | .0000  |
| 5  | 1.013905 |     | .0000  |
| 6  | 1.013927 |     | .0000  |

| SPLIT TIME |  | INDEX/MAX |
|------------|--|-----------|
| 1.013663   |  | 6.212     |
| 1.013839   |  | .0001     |
| 1.013862   |  | .0000     |
| 1.013884   |  | .0000     |
| 1.013905   |  | .0000     |
| 1.013927   |  | .0000     |

1 2 3 ALL SPLIT EXIT

Library can be installed in any of ports 0-2



Individual keys are used to start/stop and split each of the three timers



Most stopwatch features are available as UserRPL programmable functions

```

4:1
1:00:00
1:00.00

***** STOPWATCH 2 *****
Start Date: 07/10/13
Start Time: 12:04:45P
Status: Stopped
Total Time: 4:30.42
Total Splits:
Split Range: 1-4

Split      HH:MM:SS.ss
1:         1:01.03
           1:01.03

2:         1:57.88
           56.85

3:         3:14.94
           1:17.06

4:         4:30.42
           1:15.48

***** STOPWATCH 3 *****
Start Date: 07/10/13
Start Time: 12:04:45P
Status: Stopped
Total Time: 4:30.42
Total Splits:
Split Range: 1-4




```



Lists of timer and split info can be printed for those with access to a printer

## Installation/Removal

Installing the MultiStopwatch library follows the same process as most other libraries for the hp 49-series systems. After downloading the library, unzip and obtain the library specific to your preferred language (eg. EN=English, DE=German, etc.). Copy the resulting file to your calculator using the [HP Calculator PC Connectivity Kit \(Conn4x\)](#) or a standard SD card in the normal manner. Then use the following steps to install.

To *install* (steps assume RPN mode):

1. Load the library onto stack level one (use the Filer if needed)
2. Enter the port number you wish to store the library in (0|1|2)
3. Press the  key
4. Press and hold the  key, then press  and release both keys at the same time

The library will attach itself, and you will then be able to see the library commands in the **LIB** list (   ). The library commands are listed in the **MULTISH** library directory.

To *remove*:

1. Enter **1417** onto the stack
2. Execute **DETACH**
3. Enter **:x:1417 PURGE** where *x* is the port number in which you originally installed the library.

Note that any stopwatch data files you created will still exist in the same places you left them; you will need to delete those manually.

## MSW Console

|  | HR | MM | SEC       | SPLITS |
|--|----|----|-----------|--------|
| 1                                      |    |    | .00       | ← 0    |
| 2                                      |    |    | .00       | 0      |
| 3                                      |    |    | .00       | 0      |
| SPLIT TIME                             |    |    | INDEX/MAX |        |
| Σ                                      |    |    | .00       | 0      |
| Δ                                      |    |    | .00       | 15644  |
| ▶ 1   ▶ 2   ▶ 3   ▶ ALL   SPLIT   EXIT |    |    |           |        |

Pressing the **MSW** menu button activates the **console**, which is the primary way in which the stopwatch timers and splits are manipulated and viewed. There are three rows for the timers, each showing the current run time as well as the number of splits that have already been stored for each timer. An arrow points to the timer which is considered to be *active*. The split information shown below the timers pertains to the active timer only. You can change the active timer by using the up and down arrow keys at any time. Likewise, you can navigate through the list of splits for the currently active timer by pressing the left and right arrow keys. The active timer is also the one which will be altered if you press the default **SPLIT** menu button (or **ENTER**).

In addition to the menu buttons and arrow keys, the **APPS** **G**, **MODE** **H**, **TOOL** **I**, and **ON** keys have a special meaning while the console is running. They will directly add a split to timer 1, 2, 3, or *all* running timers (respectively) regardless of the currently active timer. Using these keys gives you the ability to easily add splits without having to change the active timer.


















Each split is represented with two distinct values: the *absolute* time (marked with the  $\Sigma$  symbol) and the *relative* time (marked with the  $\Delta$  symbol). Absolute represents the elapsed time **since the timer started**, and relative is the elapsed time **since the last split**. Splits are added automatically when the timer is stopped (or reaches its maximum value of 999:59:59.99).

When the MSW console is started, it looks for a MSW library data object ("MSWdata") in the current directory to load current timer data. If one isn't found in the current directory, a new one is created automatically. **Note that the console application does not follow the usual HP 49-series practice of looking up through the parent directories of the current one to seek the library data object.** This is done so that there is no confusion about which data object is current (and to make sure that you don't accidentally alter a different file than was intended). You must set the current directory *before* starting the console application (or using the programmable functions, see below).

The MSW Console can be exited at any time by pressing either the **EXIT** menu button or the **ON** key.

**NOTE:** The calculator is constantly running a Saturn code object at full speed while the console is active, even if none of the timers are running. The built-in auto-powerdown feature of the calculator is not enabled while the console is active, so it's possible to deplete the batteries if you leave it running indefinitely. If you don't need to keep the console active, it's best to exit it if battery consumption is a concern. As always, the timers will be maintained after you exit the console.

## Console Menu Items

| Menu Item  | Description   |
|--|---|
|  1  2  3<br> 1  2  3 | Toggles the run/stop status of each stopwatch timer   |
|  ALL  | Starts all timers simultaneously. If a timer is already running, it isn't altered by this command.  |
|  SPLIT  | Adds a split to the currently active timer  |
|  EXIT   | Exits the application and saves all changes made during the console session to the library data object in the current directory. The  key performs this same function. |
|  +SP 1+SP 2+SP 3  | Adds a split to the timer indicated on the menu button. Changes the active timer to the one specified.  |
|  = ALL  | Stops all timers simultaneously. If a timer is already stopped, it isn't altered by this command.   |
|  SPALL   | Adds a split to all timers simultaneously. Only applies to running timers.  |
|  ABORT  | All changes (starts, stops, splits, resets) since the console was activated are cancelled with this option  |
|  × 1 × 2 × 3  | The indicated timer is stopped, the time cleared, and all splits deleted. Also clears the initial start date/time (only available when exporting or printing stopwatch data).   |
|  × ALL  | All stopwatch timers are reset as described above   |
|  ABOUT  | Shows the current version/date/copyright information  |

## Programmable Functions

Most of the functionality available in the console is also available in stand-alone functions that can be called from a UserRPL program. **In every case, the current directory should be set appropriately before calling the function to make sure that the proper timer set is being manipulated.** Here's a list of the functions, the needed parameter(s), and how they affect the timers.

| Function      | Stack Parameters   | Description   |
|---------------|--|---|
| <b>RSTSW</b>  | L1: <b>Stopwatch ID</b><br>(1   2   3) [numeric]   | The indicated timer is stopped, the time cleared, and all splits deleted. Also clears the initial start date/time (only available when exporting or printing stopwatch data).   |
| <b>RSTAL</b>  | (none)   | All stopwatch timers are reset as described for <b>RSTSW</b> , and the current timer is set to Stopwatch 1.   |
| <b>STARTS</b> | L1: <b>Stopwatch ID</b><br>(1   2   3) [numeric]   | Starts the indicated timer if it is currently not running. If the timer is already running, no changes occur. If the timer has never been run before, the initial start date and time is set appropriately. A timer that has already reached the maximum timer value ( <b>999:59:59.99</b> ) cannot be started.           |
| <b>STARTA</b> | (none)   | Starts all timers simultaneously. See <b>STARTS</b> for a complete description.   |
| <b>SPLTS</b>  | L1: <b>Stopwatch ID</b><br>(1   2   3) [numeric]   | Adds a split to the stopwatch indicated in stack level one.   |
| <b>SPLTA</b>  | (none)   | Adds a split to all running stopwatches. Stopped stopwatches are not altered.   |
| <b>STOPS</b>  | L1: <b>Stopwatch ID</b><br>(1   2   3) [numeric]   | Stops the stopwatch indicated in stack level one, and adds a split at the same point in time the stop occurred.   |
| <b>STOPA</b>  | (none)   | Stops any running timers, adding splits for each one that was stopped. Timers that aren't running aren't altered in any way by this command.  |
| <b>RCLSP</b>  | L2: <b>Stopwatch ID</b><br>(1   2   3) [numeric]<br><br>L1: <b>Split Number</b><br>[numeric] | Recalls the split indicated by the given parameters to the stack. The split is an array with two values: the absolute time and relative time, expressed as real numbers in the form HHH.MMSSss. The absolute value is the elapsed time since the timer started, and the relative value is the elapsed time since the last |

|              |  |  |
|--------------|--|--|
|              |  | split.   |
| <b>RSPCT</b> | L1: <b>Stopwatch ID</b><br>(1   2   3) [numeric] | Recalls the split count for the given stopwatch to the stack as a real number.   |
| <b>RCLTM</b> | L1: <b>Stopwatch ID</b><br>(1   2   3) [numeric] | Recalls the current timer value of the given stopwatch to the stack as a real number in HHH.MMSSss format.   |
| <b>RXPRT</b> | L1: <b>Export Parameters</b> [list]              | Recalls a list containing the specified timer data to the stack. See the <a href="#">Exporting Data</a> documentation for details of the list object required as input as well as the output obtained. |
| <b>RPRSW</b> | L1: <b>Print Parameters</b> [list]               | Prints timer and split information in a special format (see <a href="#">Printing</a> ). See the <a href="#">Exporting Data</a> documentation for details of the list object required as input.         |



## Exporting Data

All of the data that MultiStopwatch uses to keep track of the various timer and split information is contained in a "library data" object. If you recall one of these objects to the stack, the calculator simply treats the entire contents as a single object since it doesn't know what the internal format of the data is:

```
RAD XYZ HEX R= 'X'
<HOME>
7:
6:
5:
4:
3:
2:
1: Library Data
MSWdata C6501
```

This is beneficial from a couple of standpoints. It simplifies (and speeds up) the process of storing and moving the data object around to other locations (eg. subdirectories and the SD card). It also strengthens the integrity of the data contained within the object by making it's contents unalterable by any standard means.

It's possible that you'll want to write your own programs to take advantage of the stopwatch data, and there are several programmable functions that can retrieve specific fields within a given **MSWdata** file. **RCLSP** (Recall Split), **RSPCT** (Recall Split Count), and **RCLTM** (Recall Time) can all be used in this way. These functions provide a convenient way to obtain these specific items from the MSWdata objects, and they're simple to use within a program when you need just a few items.

When you want to obtain *all* of the data (or any fields other than splits, split counts, and timer values), the best and fastest approach is to use the export features of MultiStopwatch. Two options are available to export data: **EXPRT** and **PXPRT**. **EXPRT** is an interactive command that first establishes which data you wish to obtain, then creates a list object containing the specified data and leaves it in stack level one upon completion. **PXPRT** works in a similar fashion, but instead of prompting the user to provide the details, it looks in stack level one for a list that identifies the set of data you want.

### EXPRT Command

Pressing the **EXPRT** button loads a form that allows you to select which timers you wish to include in the exported data and the range of splits to include with them:

```
Select Stopwatch Data
Stopwatch Include Split Range
1      [X]      1      93
2      [X]      1      56
3      [X]      1     174

Check to include data from SW1
EDIT [X]CHK [X]CANCEL OK
```

This is a standard HP 49-series form which uses the the usual "INFORM" controls and options. The fields default to the complete set of data from the current MSWdata object. Checking or un-checking the mark in the "Include" column will determine if that timer's data is included in the export. The range of splits can also be edited, and the values you provide will be checked for consistency with the actual data in the MSWdata object. If you only wish to export the timer information with no split data, set both fields in the Split Range to zero (they will be shown with a dash). The export will always contain at least one split for every timer, even if none exist. In the

case where none exist (or you exclude them in the range), an array with one entry of zeros ( `[[ 0. 0. ]]` ) is still included to ensure consistency of the format of the exported data.

## PXPRT Function

The **PXPRT** function doesn't prompt the user for the timer parameters, but instead takes a list of nine numbers from stack level one (ex. `{ 1 1 93 1 1 56 1 1 174 }`). The nine numbers can be thought of as three groups of three, with each group representing the three options for each timer in the "Select Stopwatch Data" form shown above.

| Fields (in order) | Type    | Description  |
|-------------------|---------|--|
| IncludeSW1        | numeric | 0=Do not include; any non-zero number specifies that SW1 data will be included in the export. If IncludeSW1=0, SplitStartSW1 and SplitEndSW1 are ignored (but still required). |
| SplitStartSW1     | numeric | The first SW1 split to include in the export list.   |
| SplitEndSW1       | numeric | The last SW1 split to include in the export list.  |
| IncludeSW2        | numeric | 0=Do not include; any non-zero number specifies that SW2 data will be included in the export. If IncludeSW2=0, SplitStartSW2 and SplitEndSW2 are ignored (but still required). |
| SplitStartSW2     | numeric | The first SW2 split to include in the export list.   |
| SplitEndSW2       | numeric | The last SW2 split to include in the export list.  |
| IncludeSW3        | numeric | 0=Do not include; any non-zero number specifies that SW3 data will be included in the export. If IncludeSW3=0, SplitStartSW3 and SplitEndSW3 are ignored (but still required). |
| SplitStartSW3     | numeric | The first SW3 split to include in the export list.   |
| SplitEndSW3       | numeric | The last SW3 split to include in the export list.  |

## Export List

The exported data shows up in the stack as a list of lists in the following format:

`{ { SW data n } { SW data n+1 } { SW data n+2 } }`

The actual number of sublists in the export depends on which timers were selected for exporting. If no timers were selected, the export list will simply be empty ( `{ }` ). Otherwise, the total number of sublists will be equal to the number of timers that were selected for export.

The sublists each contain two arrays, representing *timer* and *split* data for the selected stopwatch. So a slightly more detailed view of the export list would be as follows:

```

{
{ [ array of timer info n ] [ array of splits n ] }
{ [ array of timer info n+1 ] [ array of splits n+1 ] }
{ [ array of timer info n+2 ] [ array of splits n+2 ] }
}

```

The timer info array has the following elements:

| Element          | Type        | Description  |
|------------------|-------------|--|
| StopwatchID      | real number | (1   2   3) The stopwatch ID Number of the stopwatch for which this data applies |
| RunStatus        | real number | 0=currently stopped, 1=currently running   |
| InitialStartDate | real number | The initial start date of the timer, using the current calculator settings       |
| InitialStartTime | real number | The initial start time of the timer, in HH.MMSSsss format                        |
| CurrentTime      | real number | The current elapsed time of the timer, in HHH.MMSSss format                      |
| SplitCount       | real number | The total number of splits defined for the timer                                 |
| FirstSplit       | real number | The index of the first split included in the split list                          |
| LastSplit        | real number | The index of the last split included in the split list                           |

Each element in the split array is an array containing two elements: the absolute and relative times of each split (see the [RCLSP documentation](#) for a description of these values).

**Note:** If no splits were selected for exporting, there will still be an entry for the array of splits in the export list, and it simply contains [[ 0. 0. ]].

## An example

The following is an example of the exported data from a set of three timers, with all timer and split data included. The carriage returns and indentation were added to aid in readability:

```
{
  {
    [ 1. 1. 7.132013 21.414172644 2.22002929687E-2 2. 1. 2. ]
    [
      [.005545 .005545 ]
      [.020009 .010463 ]
    ]
  }
  {
    [ 2. 1. 7.132013 21.414172644 2.22002929687E-2 2. 1. 2. ]
    [
      [.010041 .010041 ]
      [.020009 .005967 ]
    ]
  }
  {
    [ 3. 0. 7.132013 21.414172644 2.09282348633E-2 3. 1. 3. ]
    [
      [.010637 .010637 ]
      [.020009 .005372 ]
      [.020928 .000919 ]
    ]
  }
}
```

If your ultimate goal with the export is to move the data to a computer or tablet device, you'll probably want to execute the ->STR function on the list before moving it.

## Printing

An interactive printing command (**PRMSW**) is provided with MultiStopwatch, and it can be used to print a pre-formatted listing of timer and split values. The utility uses the built-in commands for printing, so the current IO settings will apply (no flags are changed by the utility). Printing to a simulated printer using [Christoph Giesselink's excellent HP 82240b Simulator](#) also works, and this is another option for exporting data should you need to go that route.

When executing the **PRMSW** command, you'll first be prompted to specify the timers and splits to be included in the same fashion as for exporting (described above). Once selected, the timer and split information is sent to the printer formatted as in this example (the start date and start time fields will vary depending on your mode settings):

```
***** STOPWATCH 1 *****
start Date: 07/14/13
start Time: 01:38:48P
status: Running
Total Time: 1:50.20
Total Splits:
Split Range: 1-3

Split HHH:MM:SS.SS
1: 25.60
   25.60

2: 44.18
   18.50

3: 1:25.87
   41.69

***** STOPWATCH 2 *****
start Date: 07/14/13
start Time: 01:39:05P
status: Running
Total Time: 1:33.20
Total Splits:
Split Range: 1-2

Split HHH:MM:SS.SS
1: 5.13
   5.13

2: 42.91
   37.78

***** STOPWATCH 3 *****
start Date: 07/14/13
start Time: 01:39:26P
status: Stopped
Total Time: 37.23
Total Splits:
Split Range: 1-2

Split HHH:MM:SS.SS
1: 13.17
   13.17

2: 37.23
   24.06
```

Additionally, a programmable equivalent function is provided (**PPRSW**). The output sent to the printer is the same as shown above, and the input required is the same kind of list [described above](#) as input for the **PXPRT** function.

## General Usage Notes

### Simultaneous Functions

The main stopwatch functions (start/stop/split) all have commands to operate simultaneously on all three stopwatches. This can be useful, for example, when timing up to three different participants in a single event. Another use of this feature could be the timing of different intervals of a *single* participant -- for example marking splits for quarter laps in one timer, full laps in another. In each case, the functions will only perform their specific operation for a stopwatch under the appropriate conditions (eg. only running stopwatches will be split when executing **SPLTA**).

### System Clock is the Basis for Timer Operations

The built-in system clock on the 49g+/50g is the basis for all time measurements. The accuracy of the stopwatches is therefore tied to the accuracy of the system clock in your calculator, which is likely to drift 1-2 seconds/day. Please note that HP did not design these systems with precise timing accuracy in mind; it would be inappropriate to assume that the results of timer operations would be adequate for official timekeeping purposes.

Any adjustments made to your system clock between invocations of the MSW console or RPL commands will have a corresponding impact to all running timers. For example, if you were to adjust the date on your calculator by adding one day, all *running* timers would correspondingly advance by 24 hours. Stopped timers aren't affected by clock changes. It is important to keep this relationship in mind when making changes to your calculator's system clock.

### About Battery Consumption

The MSW Console (**MSW** in the main library menu) is a code object that is constantly running while it is active, and as such it uses more power when activated than an idle system does. As a result, leaving the console application open could potentially deplete your batteries if left unattended. It's best to close the console if you don't anticipate making changes in the near term. All of the RPL functions are only active for a very brief time (<1 sec), so their usage causes no more power drain than any other functions or commands you may execute. **Leaving a timer in a running state when exiting the console has *no* impact on battery consumption.** The timers aren't updated until the value is actually needed by an RPL command or the console.