uRT51 Real-Time Suite

User's Manual

Ricardo Cayssials

June 2005

Contents

1. Overview

uRT51 Real-Time Suite: An Integrated Real-Time System Tool

uRT51 Real-Time Suite is a high performance tool that supports programming, debugging and analysis of real-time systems implemented on uRT51 microprocessors.

uRT51 Real-Time Suite describes a real-time system in terms of tasks, real-time properties and priority discipline. You may include as many tasks as you like within the memory constraints of your uRT51 microprocessor system. These tasks are schedule accordingly to the priority discipline you select. Arbitrary priority disciplines can be included in the uRT51 Real-Time Suite under request.

uRT51 Real-Time Suite is full featured, high-performance, interactive and easy to use. It generates real-time applications quickly and easily, supports code debug and allows trusty analysis of runtime execution on the uRT51 microprocessor. It includes all facilities for runtime analysis as well as supports data logging of all real-time data values.

Real-Time Tasks and Properties: The Basic Objects

The basic object of a real-time system in the uRT51 Real-Time Suite is a task. A task carries out a certain function of the application. A task is defined by:

- its code.
- its real-time properties: it sets the temporal properties of the task such as periodical invocation, priority an so on. Some real-time properties depend on the priority discipline selected.

The task code defines the way that the task performs its function. The real-time properties establish the runtime behaviour of the task such as periodical invocation, priority and so. Some real-time properties depend on the priority discipline selected.

Priority Discipline

uRT51 Real-Time Suite supports a predefined set of priorities disciplines. A priority disciplines defines the way in which real-time tasks are schedule during runtime and consequently the behaviour of the system.

uRT51 Real-Time Suite simplifies the process of programming the priority discipline on the uRT51 microprocessor. Because the high flexibility of uRT51 microprocessor a wide range of priority disciplines can be implemented on it and additional ones can be integrated into the uRT51 Real-Time Suite under request.

There is no present limit to the priority disciplines that can be included. Further priority disciplines implementations will be available as updates.

Runtime Analysis

uRT51 Real-Time Suite includes fully featured tools for runtime analysis. Data is retrieved from uRT51 microprocessor during runtime with no interference on the task execution. Runtime behaviour can be monitored and analysed on-line.

uRT51 Real-Time Suite allows logging of the runtime data for further analysis. Coma Separated Value (CSV) format is used for easy integration with data processing tools.

uRT51 Real-Time Suite Modules

- Task Editor
- Real-Time Properties dialog box
- Control Console
- CPU Register viewer
- Runtime code execution viewer
- Scheduling Analyser
- Variable Viewer
- Message Viewer.

The Task Editor allows user to specify the task code and the real time properties of each task of the application. Each task is hold in an individual and resizable window for easy, quick and interactive task programming.

The Real-time property dialog box allows user to define the priority discipline that the uRT51 microprocessor will implement. Additional priority disciplines can be included under request.

The control console allows user to interact with the uRT51 microprocessor board to download, trace and supervise its runtime behaviour.

The CPU registers viewer shows the data holds in each register of the CPU. The data can be displayed either in hexadecimal, decimal or binary format.

The runtime code execution viewer shows the current instruction that the processor is executing. It changes from task to task according to the current executing task.

The scheduling analyser shows graphically the execution of each task of the application. Zooms and measurement rules are included.

The variable viewer shows the data stored in each variable. The set of variables to be displayed is selected from a list and can be put on view in different formats. Logging of the runtime behaviour can be done for further analysis.

The message viewer shows the errors that happened in the last compilation. A double-click on the error will switch to the line code that caused the error.

Documentation Conventions

This section contains information about the typographical and stylistic conventions used in this user guide.

Special Fonts

The following inline references are represented by a bold font:

- Menu commands (For example: **File** menu).
- Literal user input (For example: Enter **100**).

- Program output (For example: uRT51 Real-Time Suite record data into file **name.rtd**)
- All dialog elements – fields, checkboxes, drop-down menus, etc. (For example: Click **Accept**).

Freestanding quotations of input examples, file listings, and output messages are represented by a constant-width for – for example:

```
loop1: mov a, 01H
```

Variables for which context-specific substitutions should be made are represented by bold italics – for example, ***myfile*.asm**.

Sequential steps in a tutorial are set off with a triangle ( ) in the margin.

References to mouse buttons are given in all capitals – for example, MOVE/EDIT. When a key is to be pressed and held while a mouse button is used, the key and button are adjoined by a plus sing (+). For example, **Shift**+SELECT means that the **Shift** key is pressed and held while the SELECT mouse button is used.

The terms “left-click”, “right-click”, and “middle-click” all assume default mappings for mouse buttons.

Text omitted for clarity or brevity is indicated by an ellipsis (...).

Menu Commands and Dialog Titles

Elements in hierarchical menu paths are separated by a > sign. For example, **File > Open** means the **Open** command in the **File** menu.

Tabs in dialog boxes are set off from the command name or dialog title by a dash. For example, **Properties – Real-Time** refers to the **Real-Time** tab of the **Properties** dialog.

Special Keys

Special keys are represented by the following abbreviations:

Key	***Abbreviation***
Shift	**Shift**
Enter	**Enter**
Control	**Ctrl**
Alternate	**Alt**
Backspace	**Back**
Delete	**Del**
Escape	**Esc**
Insert	**Ins**
Tab	**Tab**

Home	**Home**
End	**End**
Page Up	**PgUp**
Page Down	**PgDn**
Functions Keys	**F1 F2 F3 ...**
Arrow Keys	**🡻, 🡸, 🡺, 🡹**

When certain keys are to be pressed simultaneously, their abbreviations are adjoined by a plus sign (+). For example, **Ctrl**+**S** means that the **Ctrl** and **R** keys are pressed at the same time.

When certain keys are to be pressed in sequence, their abbreviations are separated by a space ( ). For example, **Alt**+**E R** means that the **Alt** and **E** keys are pressed at the same time and then released, immediately after which the **R** key is pressed.

Abbreviations for alternative key-presses are separated by a slash (/). For example, **Shift**+🡹 / 🡻 means that the **Shift** key can be pressed together with either the up (🡹) arrow key or the down (🡻) arrow key.

Contact Information

uRT51 Real-Time Solutions

info@uRT51.com.ar

2. Getting Started

Launching uRT51 Real-Time Suite

To launch uRT51 Real-Time Suite, click the **Start** button on the windows taskbar and navigate to the uRT51 Real-Time Suite installation directory (in default installation, in **Programs** > **uRT51 Suite** > **uRT51 Suite**).

You can also double-click the uRT51 Real-Time Suite icon on your desktop, which looks like this:

Setup Files

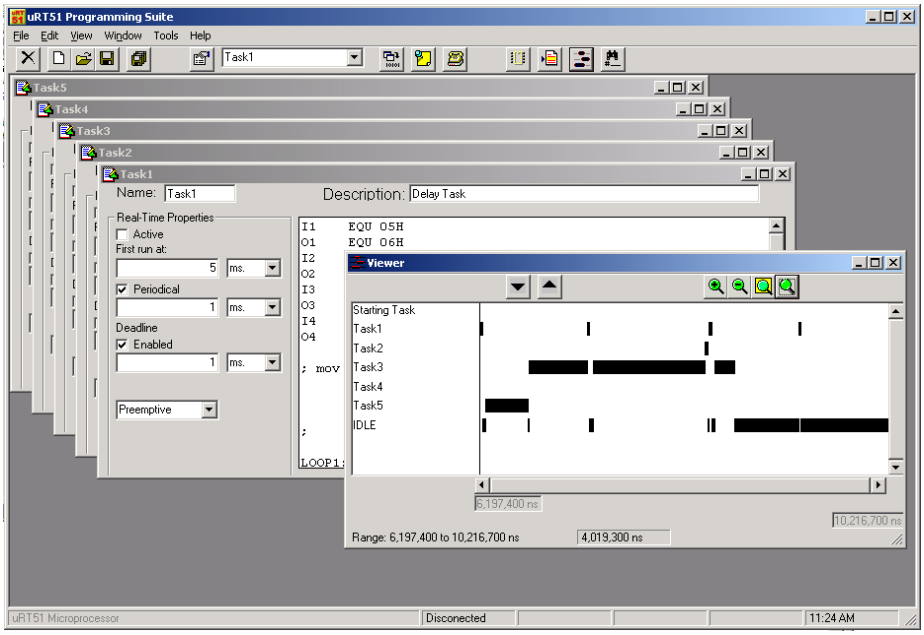
uRT51 Real-Time Suite contains basic information such as a current project, compilation directory, environment settings, and module-specific options for viewers. Collectively, this information is known as the “setup”.

When you launch uRT51 Real-Time Suite, the program attempts to locate the file **suite.ini** and read it for setup information. If it finds the file in the installation directory, then the last project is loaded. In this way, program starts in the same state as it was close last time. If **suite.ini** is not found, an empty project with no tasks is loaded.

User Interface

The uRT51 Real-Time Suite has the following three major components:

- Menu bar (adjoined to the title bar).
- Standard toolbar.
- Working area.



Title Bar and Menu Bar

The *title bar* indicates the current project. The *menu bar*, the horizontal space at the top of the screen, contains the titles of the uRT51 Real-Time Suite menus.

File	Commands for creating, opening, saving, and printing tasks codes.
Edit	Commands for copying, deleting, pasting, and system configuration.
View	Command for status bar viewing.
Window	Commands for displaying windows.
Tools	Commands for accessing uRT51 Real-Time Suite modules.
Help	Commands for accessing online user guides and general information about uRT51 Real-Time Suite.

Arranging Windows

The **Window** menu contains commands for manipulating tasks editing windows and modules windows.

If you have multiple windows open, you can bring one of them to the front by selecting it from the list at the botton of this menu.

Window > Cascade arranges windows in overlapping fashion, starting from the top left corner of the display area, so that the title bars are visible. The active window remains active (in front).

Window > Horizontally Tile arranges windows from top to bottom in non-overlapping fashion, resizing them to fill the display area.

Window > Vertically Tile arranges windows from left to right in non-overlapping fashion, resizing them to fill the display area.

Window > Arrange Icons arranges icons of minimized windows in rows starting at the botton left of the display area.

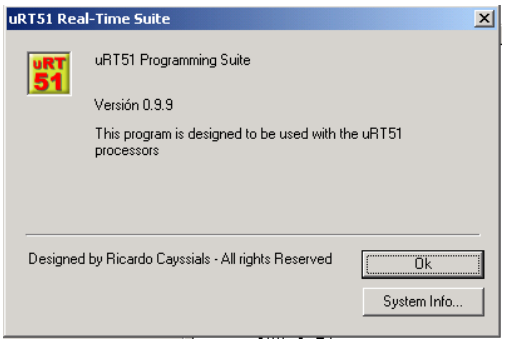
Window > (open window list) lists all open files in the order they were opened. The active file is indicated with a check (√).

Getting Help

To access the documentation select one of the following **Help** menu commands:

- Contents
- Search ...

To determine what version of uRT51 Real-Time Suite you are using, choose **Help > About uRT51 Real-Time Suite**. uRT51 Real-Time Suite displays the following dialog:



Standard Toolbar

Standard toolbar contains buttons for quick access to commands.

Button	***Menu Command***
	**File > Delete**
	**File > New**
	**File > Open Task**
	**File > Save Current Task**
	**File > Save Project**
	**Edit > System Configuration – Real-Time**
	**File > Compile**
	**Tools > Message Viewer**
	**Tools > Control Console**
	**Tools > CPU Registers**

Tools > Runtime Code Execution Viewer

Tools > Scheduling Analyzer

Tools > Variable Viewer

Status Bar

The *status bar*, located at the bottom of the uRT51 Real-Time Suite window, displays context-sensitive information about items in the interface.

The status bar includes the status of the link with the uRT51 microprocessor board.

Working Area

The area available for displaying windows of modules is called the *working area*. Windows in the working area can be arrange according to the option of the **Window** menu.

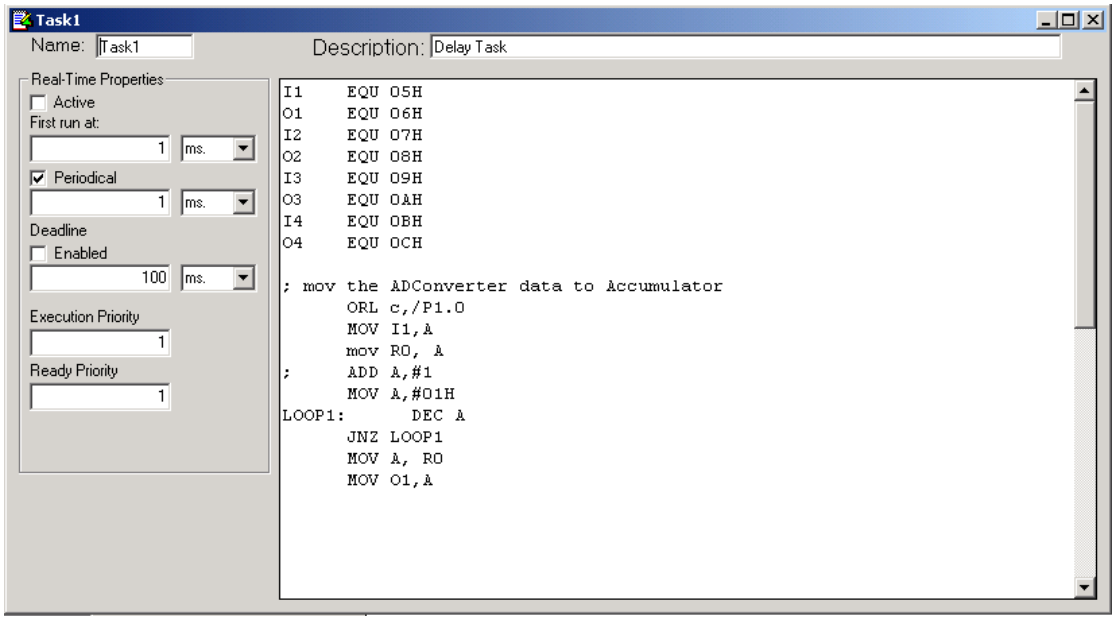
3. Working with Tasks

Tasks

A complete uRT51 Real-Time Suite project is composed of tasks. The code of each tasks is saved in a **.asm** file. Your project can include as many tasks as your hardware allows. The name of each task should be unique in the project.

Creating Tasks

Create a new task by choosing **File > New Task**, which creates a new Task Editor Window.



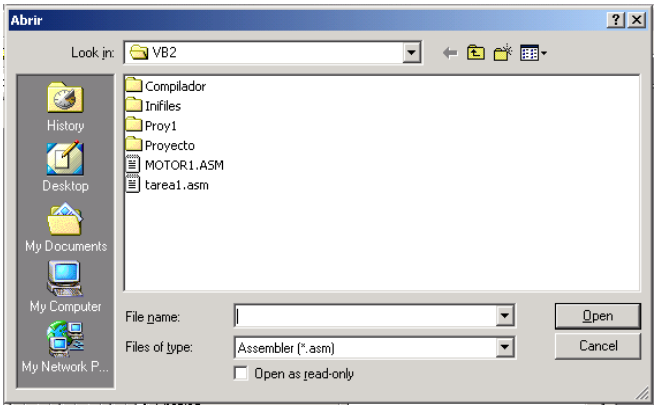
A Task Editor window contains the following sections:

- **Name**: is used as the identification of the task in the system. The name of the task must be unique in the project. When a new task is created, a sequential default name is assigned.
- **Description** (optional): is an information text that can be used for documentation purposes. When a new task is created, it is assigned the task's name.
- **Real-Time Properties** box: holds the real-time parameters of the task. These parameters change according to the priority discipline chosen. When a new task is created, default values are assigned.
- **Task Code:** contains the code of the task. The code is the 8051 subset supported by the uRT51 microprocessor. An empty code is set when a new task is created.

Opening Tasks

There are three ways to open a file:

- Select **File > Open Task** ....
- Click the open file button ( ).
- Press Ctrl + O.

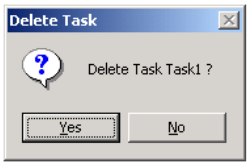


Options include:

Look in	The source directory.
File name	The name of the file to be opened. The wildcard character (*) can be used to narrow down the list of available files. (For example to list only ASM files whose names begin with the letter **a**, type **a*.asm** and press **Enter**.) Only one file can be opened at a time. All files of the specified type in the source directory are listed in the space above this field.
Files of type	The type of file listed. Predefined files are ***.asm**.

Closing (Deleting) Files

To close the current file, choose **File > Delete**. uRT51 Real-Time Suite will prompt you to confirm deletion.

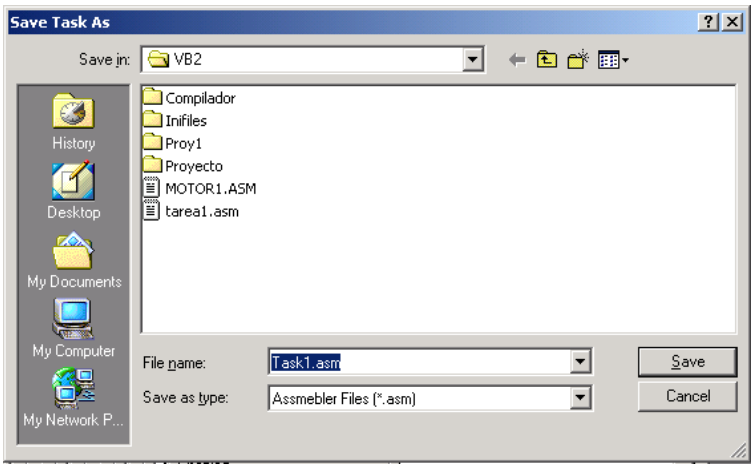


- Clicking **Yes** deletes the task from the project.
- Clicking **No** cancels the deleting operation.

Saving Tasks

To save the current task, choose **File > Save Current Task**. uRT51 Real-Time Suite saves the tasks using its current filename and path.

To save a task using a different name or location, choose **File > Save As**. uRT51 Real-Time Suite displays the **Save As** dialog box.



Options includes

Save in The target directory.

File name The name under which the file corresponding to the active window is to be saved. The space above this field lists of the specified type in the target directory. If you choose a name that already belongs to an existing file, uRT51 Real-Time Suite prompts you for permission to overwrite the existing file.

Save as type The type of file listed. By default, the active file is saved in its current type – e.g. .asm.

Printing

You can print tasks in uRT51 Real-Time Suite by choosing **File > Print**. The Print dialog box appears.

Exiting uRT51 Real-Time Suit

Choose **File > Exit** to exit uRT51 Real-Time Suite. A warning prompts you to save changes in each unsaved file.

4. Building a Project

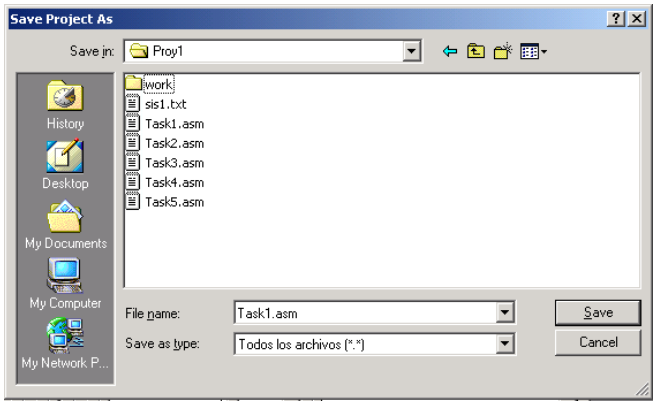
Project

A uRT51 Real-Time Suite project defines a real-time system. The project is composed of tasks, real-time properties and the priority discipline utilised. Your project can include as many tasks as your hardware allows. The name of each task should be unique in the project.

Creating a Project

A project is created when a set of tasks is saved as a project by choosing **File > Save Project** or **File > Save Project As**.

uRT51 Real-Time Suite displays the **Save As** dialog box.



Options includes

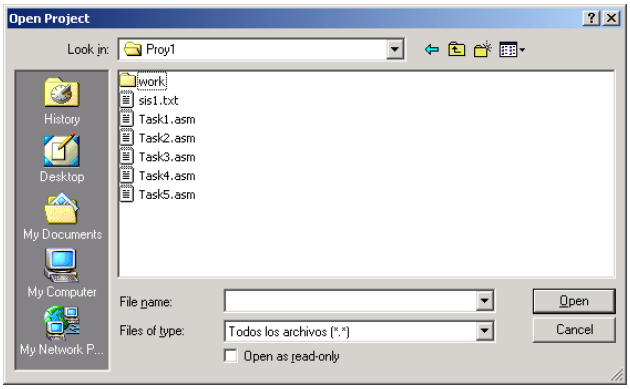
Save in — The target directory.

File name — The name under which the current project is to be saved. The space above this field lists of the specified type in the target directory. If you choose a name that already belongs to an existing file, uRT51 Real-Time Suite prompts you for permission to overwrite the existing file.

Save as type — The type of file listed. By default, the project is saved in its current type – e.g. **.txt**.

Opening a Project

To open a project, choose **File > Open Project**. uRT51 Real-Time Suite will prompt you to save the current project. uRT51 Real-Time Suite displays the **Open Project** dialog.



Options include:

Look in — The source directory.

File name — The name of the file to be opened. The wildcard character (*) can be used to narrow down the list of available project files. (For example to list only ASM files whose names begin with the letter **a**, type **a*.asm** and press **Enter**.) Only one file can be opened at a time. All files of the specified type in the source directory are listed in the space above this field.

Files of type — The type of file listed. Predefined files are ***.asm**.

Saving the Project

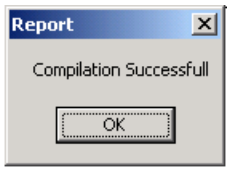
To save the current project, choose **File > Save Project**. uRT51 Real-Time Suite saves the project using its current filename and path.

Compiling the Project

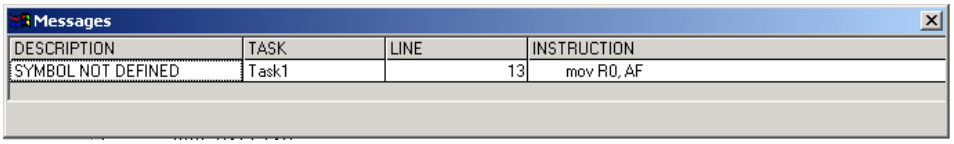
When a project is compiled, a code for the uRT51 microprocessor is built from the tasks' code and real-time properties of the project.

To compile the project, choose **File > Compile** or click the compilation button ().

URT51 Real-Time Suite will give you a report of the compilation. If there were no errors, a message of "Compilation Successful" appears. Click **Ok** to continue.



If there exist compilation errors, a message shows how many errors were found during compilation. Clicking **Ok** a window reporting the errors shows up.



The **Message** window shows:

Description	The description of the compilation error.
Task	The name of the task which produced the error.
Line	The line which produced the error.
Instruction	A copy of the line that contains the error.

The uRT51 Real-Time Suite redirects to the line that produced the error when a line of the **Message** window is double clicked.

The **Message** windows can be opened by selecting **Tools > Message Viewer** or clicking the message button ().

When project is successfully compiled, it can be downloaded to the uRT51 microprocessor.

5. Running a Real-Time Application

Running a project

When a project is successfully compiled it is ready to be executed by the uRT51 microprocessor. A project defines a real-time application and it includes all the real-time information required during runtime. uRT51 Real-Time Suite includes a set of tools that communicates with the uRT51 microprocessor in order to download, to debug and to analyse a project.

Communicating with the uRT51 microprocessor

URT51 Real-Time Suite allows you to link to the uRT51 microprocessor in order to download your project or to control the execution of your application. The commands you can send to the uRT51 microprocessor are contained in the control console of the uRT51 Real-Time Suite.

Control Console

The **Control Console** windows can be opened by selecting **Tools** > **Control Console** or clicking the console button ().

The **Control Console** window contains all the command to download and to debug your application

Connecting to the uRT51 microprocessor

uRT51 Real-Time Suite connects to the uRT51 microprocessor by clicking the **Connect** button of the control console.

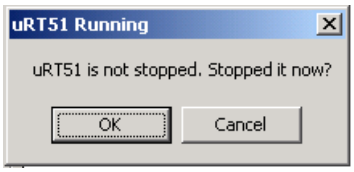
uRT51 tries to connect to the uRT51 microprocessor connect to the serial port of the PC. If no uRT51 microprocessor is connected or a communication error happens, then a error message saying “Communication not established” is shown.

The status of the connection is shown in the status bar of the uRT51 Real-Time Suite.

Downloading the Project

The first time you connect, you should download your project to the uRT51 microprocessor. You can doing that by clicking the download button of the control console. The application to download is the one gotten from the last successful compilation.

Download has to be performed with the uRT51 microprocessor stopped. If the uRT51 microprocessor is running, a warning windows is pop up asking whether you want to continue the downloading process or not.



Selecting **Ok** the uRT51 microprocessor is stopping and the download process is started. Selecting **Cancel**, the uRT51 microprocessor is not stopped and the download process is cancelled.

When the download process is in progress, the downloading window is shown. The downloading window shows the remaining time to complete the download.

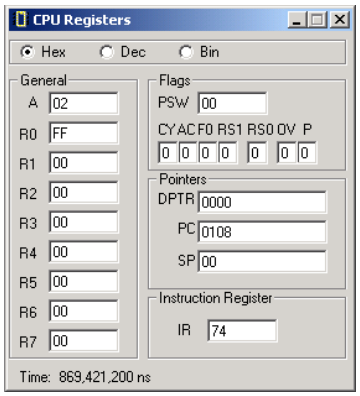
Resetting the uRT51 microprocessor

When the project was downloaded into the uRT51 microprocessor, a reset should be produced in order to start the real-time application from the beginning.

The reset of the uRT51 microprocessor has to be performed by clicking the **Reset System** button of the control console. A uRT51 microprocessor reset forces to initialise the registers of the uRT51 microprocessor and to set the time of the system to cero.

Watching the uRT51 microprocessor registers

The uRT51 microprocessors registers can be watched trough the **CPU Registers** window. The **CPU Registers** window can be opened by choosing **Tools** > **CPU Registers** or by clicking the CPU Registers button ().



The CPU Registers window shows the internal registers of the uRT51 microprocessor. At the bottom of the window the current time is shown.

The value of the registers can be watch in Hexadecimal, Decimal and Binary representations. This can be selected through the options at the top of the CPU Registers window.

Running the Real-Time Application

When project was downloaded and the uRT51 microprocessor reset, your can execute your application. To do so, you have to click the **Run** button in the control console. The uRT51 microprocessor will start to execute the application code without any debugging interference.

You can stop the execution of your application any time by clicking the **Stop** button in the control console. When the uRT51 microprocessor is stopped, all the information in the uRT51 Real-Time Suite is updated. You can re-run and stop the execution of the application by clicking the **Run** and **Stop** buttons, respectively.

The status of the uRT51 microprocessor is shown in the status bar.

The first code executed after a download is the initialisation code that is included in a special task named **Starting Task**. The uRT51 is initialised when the starting task is completed. The starting task is automatically created by the uRT51 Real-Time Suite.

The first run of each task should be set at a time that leaves enough time to the starting task to complete. In this way, the initialisation of the system finishes prior the execution of any real-time task.

6. Debugging a Project

Debugging a project

The uRT51 Real-Time Suite includes a set of tools that lets you to debug your project. The debug execution is based on events that stopped the execution of the uRT51 microprocessor. These events can come from both the complexion of a microprocessor instruction or the changed on the real-time task that is being executed.

Event configuration

When you debug an application, you want to watch the status of your application at certain inspection points. The uRT51 microprocessor should be stopped at these inspection points and the status of uRT51 microprocessor should be shown through the uRT51 Real-Time Suite.

The uRT51 Real-Time Suite lets you to define three kinds of events that will stop the execution of your application:

- when a fetch cycle is about to be started. The event is produced just before the uRT51 microprocessor starts to execute the next instruction. The status of the microprocessor is the one left by the execution of the previous instruction. Selecting the **Enable Single Step** checkbox of the control console sets this event.
- when there is a change on the current executed task. The event is produced when the uRT51 microprocessor changes the task that is being executed. Selecting the **Enable Task Trace** checkbox of the control console sets this events.
- when a **Break Execution** command is performed. When the application is being debugged, the uRT51 microprocessor can be stopped be performing a Break Execution command by clicking the **Break Execution** button of the control console.

Each one of these events stops the uRT51 microprocessor when it is in debug mode. If both **Enable Single Step** and **Enable Task Trace** are disable, the only way to stop the execution of the uRT51 microprocessor is through a **Break Execution** command.

Running a project in debug mode

In debug mode, the uRT51 microprocessor is stopped when it is running in trace mode and an event described above happens.

Starting the execution in debug mode

There exist two ways to execute an application in debug mode:

- Trace mode: the uRT51 microprocessor is stopped when an enabled event occurs, its status is updated in the uRT51 Real-Time Suite and it is automatically re-run until the next event in which the process is repeated. The trace mode allows tracing the evolution of the application. It is started by clicking the **Start Trace** button of the control console and you come out from this mode by clicking the **Stop Trace** button of the control console.

- Run 1 Step mode: the uRT51 microprocessor is stopped when an enabled event occurs and the status is updated in the uRT51 Real-Time Suite. Clicking the Run 1 Step button in the control console starts it.

Analysing the runtime behaviour of an application

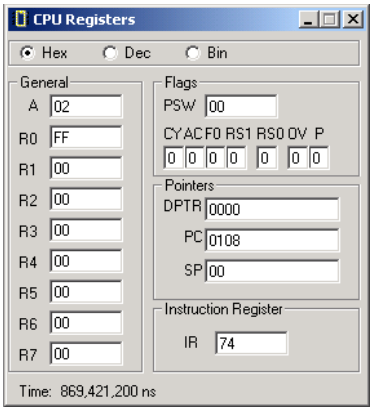
The uRT51 Real-Time Suite contains a set of tools to analyse the runtime behaviour of the real-time application. These tools let you to watch the status of the uRT51 microprocessor in order to analyse whether there exist some bugs or not.

CPU Registers viewer

The uRT51 microprocessors registers can be watched through the **CPU Registers** window. The **CPU Registers** window can be opened by choosing **Tools** > **CPU Registers** or by clicking the CPU Registers button ().

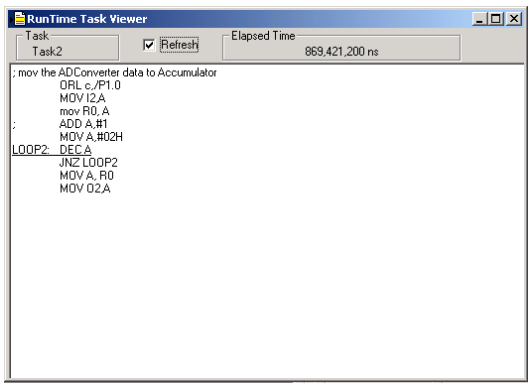
The CPU Registers window shows the internal registers of the uRT51 microprocessor. The current time is shown at the bottom of the window.

The value of the registers can be watch in Hexadecimal, Decimal and Binary representations. This can be selected through the options at the top of the CPU Registers window.

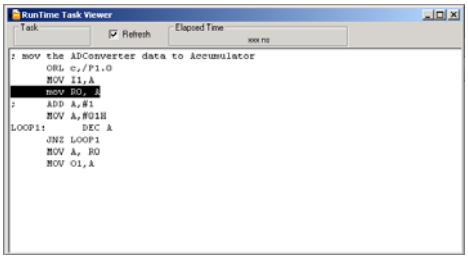


Runtime Task viewer

The Runtime Task Viewer shows the current instruction that the processor is executing. It changes from task to task according the current executing task. The current time is shown at the top of the window.



The **Runtime Task Viewer** window can be opened by choosing **Tools > Runtime Task Viewer** or by clicking the Runtime Task Viewer button ( ).



Scheduling Analyser

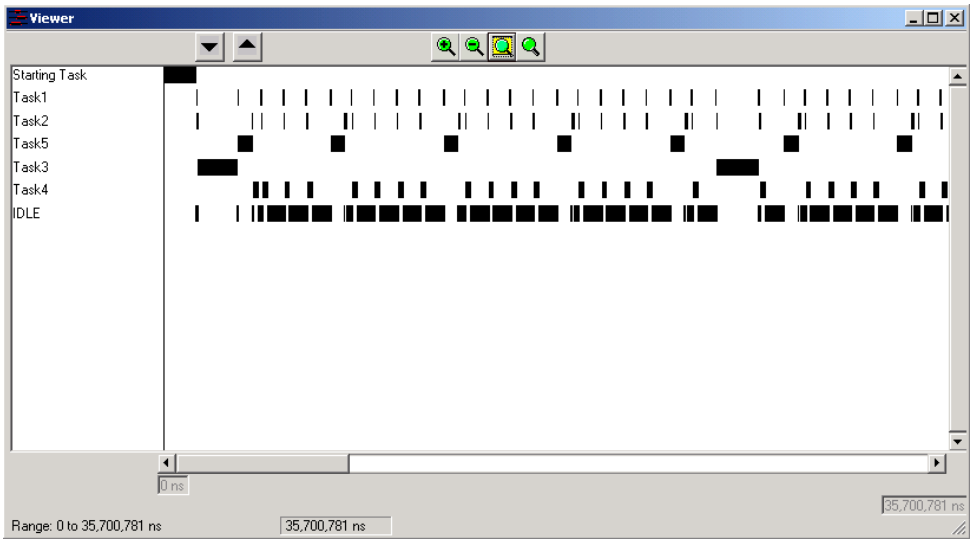
The scheduling analyser shows graphically the execution of each task of the application. The diagram showed is the last runtime recorded when execution was performed in Debug Mode.

To start the record of the runtime of the uRT51 microprocessor, the **Record Runtime** and the **Enable Task Trace** checkboxes of the control console should be enabled prior to start execution in debug mode.

> Note: Scheduling analyzer shows the runtime scheduling performed when **Record Runtime** and **Enable Task Trace** are enable and execution is performed using either **Start Trace** or **Run 1 Step.**

The **Scheduling Analyzer** window can be opened by choosing **Tools > Scheduling Analyzer** or by clicking the Scheduling Analyzer button ( ).

The Scheduling Analyzer shows a scheduling diagram of the runtime. The left panel shows the name of the tasks and the right one a timing diagram of the runtime. The order of the tasks can be modified using the buttons and .



Variable viewer

The variable viewer shows the data stored in each variable. The set of variables to be displayed is selected from a list and can be put on view in different formats.

The variables are updated when the uRT51 microprocessor is stopped. A log file can be defined to store the evolution of each variable during runtime.

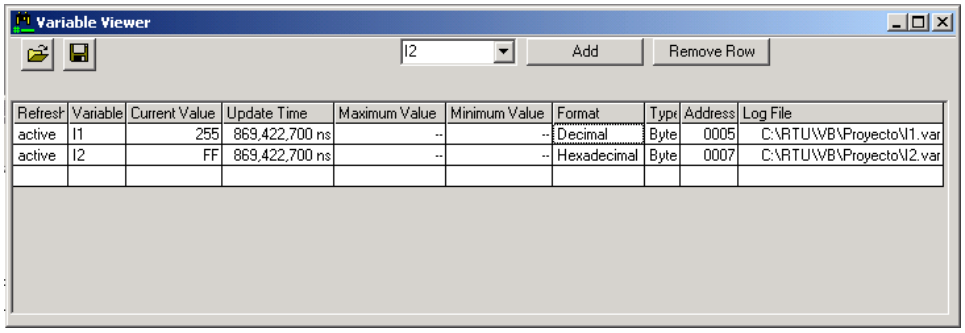
Note: Changes produced between events will not be logged.

The **Variable Viewer** window can be opened by choosing **Tools > Variable Viewer** or by clicking the Variable Viewer button ( ).

The configuration of the variable viewer can be save and load from files using the save ( ) and load ( ) button at the top of the variable viewer window, respectively.

The Variable Viewer shows the value of each variable, the last time it was modified, its maximum and minimum values and the log file.

The log file can be modified clicking the corresponding file cell.



7. uRT51 Real-Time Suite

Introduction

uRT51 Real-Time Suite is meant for the real-time programmers that want to program a real-time application in a very easy, fast and flexible way. It works with the uRT51 microprocessor board and it supports both Fixed Priorities and EDF priority disciplines.

uRT51 Real-Time Suite brings a very interactive environment that allows the user to analyse and to program real-time applications with a few key strokes.

uRT51 Real-Time Suite is intended to take much of the headache out of the art of making real-time applications. No timers, Interrupt Service Routines or handlers are needed to implement what you want. You just think what you want to do and let uRT51 Real-Time Suite program it for you.

uRT51 Real-Time Suite is a stand-alone application, that needs only the uRT51 microprocessor board.

What is the uRT51 Real-Time Suite

uRT51 Real-Time Suite is a programming environment that lets you built a real-time application with a few key strokes. You see the code of each real-time task in a resizable window where you can configure the real-time parameters as you desire. Both, Fixed Priority and EDF priority disciplines are available to schedule the real-time tasks. After setting all the parameters, you can compile and download the application to the uRT51 microprocessor board to run it.

When the uRT51 microprocessor board is configured, the uRT51 Real-Time Suite allows you to:

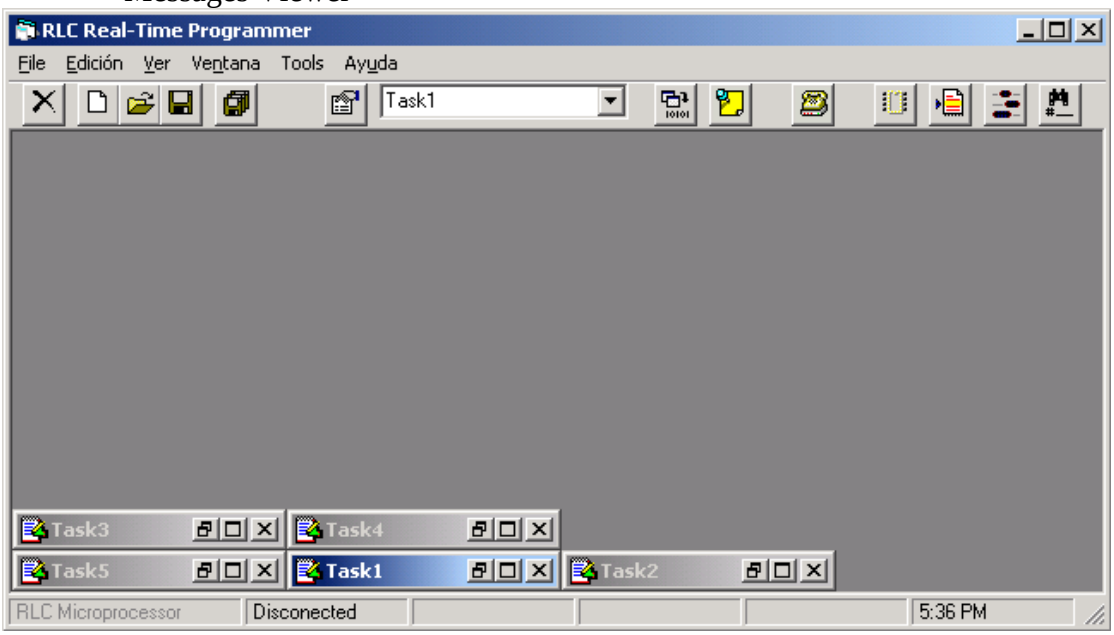
- run the application step by step.
- analyse the run time behaviour of each task.
- watch how tasks are scheduled.
- get statistics from the execution of the real time tasks.

uRT51 Real-Time Suite is a full featured, high-performance, interactive and easy to use.

User Interface

uRT51 Real-Time Suite brings an integrated user interface with the following built-in modules:

- Task Editor
- Real-Time Properties dialog box
- Control Console
- CPU Register viewer
- Runtime code execution viewer
- Scheduling Analyser
- Variable Viewer
- Messages Viewer



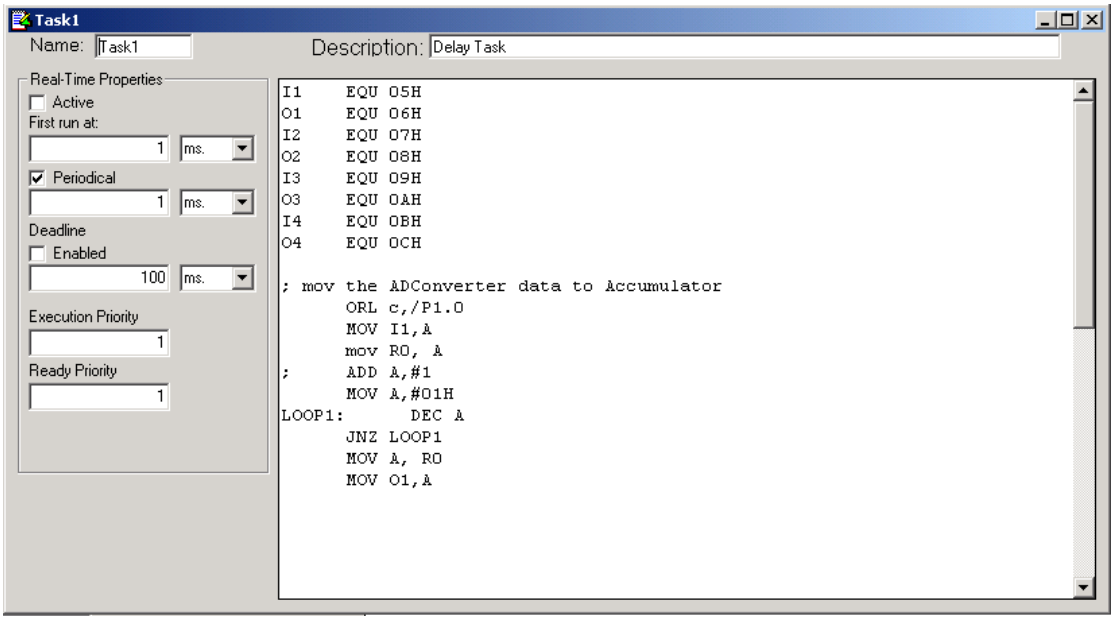
uRT51 Real-Time Suite User Interface

uRT51 Real-Time Suite Modules

Task Editor

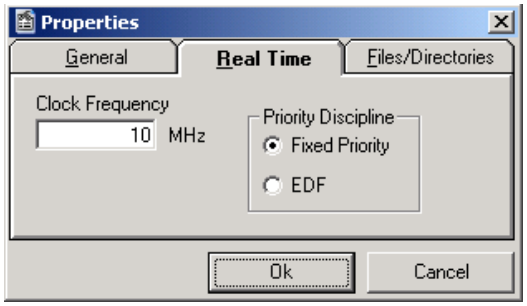
Task editor allows user to specify the code and the real time properties of each task of the application.
Each task is hold in an individual and resizable window that contains the following information:

- Name: is used as the identification of the task in the system. The name of the task must be unique in the application.
- Description (optional): is an information text that can be used for documentation purposes.
- Real-Time Properties box: holds the real-time parameters of the task. These parameters change according to the priority discipline chosen.
- Task Code contains the code of the task.



Real-Time Properties Dialog box

Real-time property dialog box allows user to define the priority discipline that the uRT51 microprocessor will implement. Additional priority disciplines can be included under request.
The Clock Frequency has to be set according to the clock frequency of the uRT51 microprocessor board.



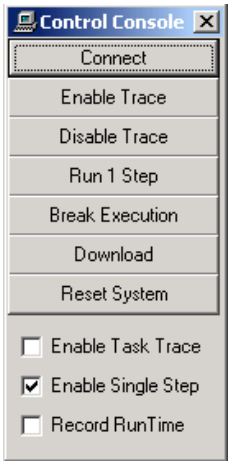
The Priority Discipline selector allows choosing the scheduling discipline to be applied during runtime. Additional priority disciplines may be included under request.

Control Console

The control console allows the user to interact with the uRT51 microprocessor board for the following commands:

- Connect with the uRT51 microprocessor board to establish communication
- Enable and Disable trace to follow the execution of each task of the system
- Download the real-time application to the board.
- Record the runtime scheduling for further analysis using the runtime analyser.

It must be noted that control console does not introduce any runtime overhead to the real-time tasks' execution.



Message Viewer

The message viewer shows the errors that happened in the last compilation. A double-click on the error will switch to the line code that caused the error.