ISSN 0280-5316
ISRN LUTFD2/TFRT--5648--SE

SDL Programming of LEGO Robots

Tomi Ervasti
Torkel Niklasson

Department of Automatic Control
Lund Institute of Technology
October 2000

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
October 2000

Document Number
ISRN LUTFD2/TFRT--5648--SE

Author(s)
Tomi Ervasti, Torkel Niklasson

Supervisor
Erik Rahlén, Telelogic AB
Karl-Erik Årzén, LTH

Sponsoring organization

Title and subtitle
SDL Programming of LEGO Robots. (SLD programmering av LEGO robotar)

Abstract
This report describes the usage of the SDL development tool Telelogic Tau SDL Suite with the LEGO Mindstorms Robotics Invention System.
The purpose of the project was to produce an alternative to the custom built platform currently used by Telelogic for demonstration and education.
The report focuses on the adaption of the C code that is generated by Telelogic Tau SDL Suite to the robot environment using legOS, an open source operating system available for the robot plat-form.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
92

Recipient's notes

Security classification

Contents

1 Introduction

1.1 Purpose

The purpose of this master thesis is to investigate whether or not it is possible to use the tools of the Telelogic Tau SDL Suite to program the LEGO Mindstorms robot with SDL.

Telelogic wants a mass produced demonstration and teaching platform. Building especially for the purpose is expensive and inconvenient, and having a mass produced platform shows the versatility of the product. Having a fun platform attracts attention and makes it easier to learn.

1.2 Objectives

There are a few objectives that should be reached within the frame of this master thesis. The first is a development kit that should consist of packaged files and instructions to get up and running with the system, including some simple demo applications. A larger demo application should also be developed. This should show the advantages of using SDL in an embedded system.

1.3 Required knowledge

To understand this report, some previous knowledge is required.

- C programming
- SDL programming (a short introduction is available in *Appendix E*)

2 Technologies

This chapter examines the available methods and technology and their respective properties. Some are vital to the goals of the project and others are optional.

2.1 Overview

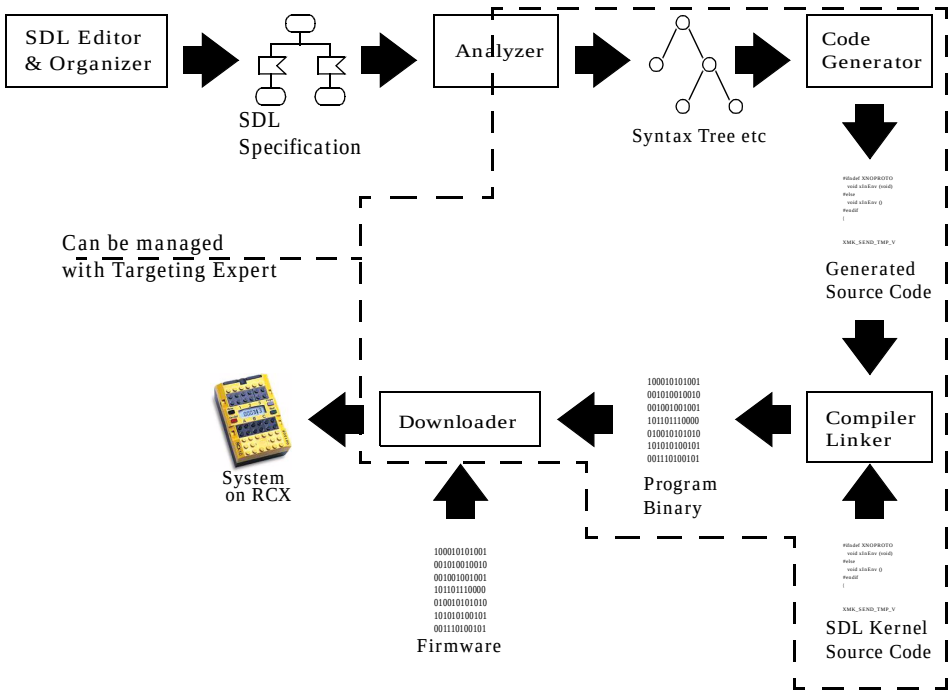


Figure 1 From SDL to running application with Telelogic Tau SDL Suite

In *Figure 1*, the relationship between the different parts of Telelogic Tau SDL Suite are explained. With *SDL Editor & Organizer* the SDL specification is generated. This is then analyzed by the *Analyzer*, and turned into code by the *Code Generator*. The generated source code and the kernel source code are then compiled with the *Compiler/Linker*. The binary from the compiler and linker are then downloaded to the platform with the *Downloader* after the

Firmware has been downloaded. The term *Firmware* is used when referring to the operating system on the RCX.

2.2 Telelogic Tau SDL Suite

The Telelogic Tau SDL Suite consists of three parts that are essential to the process of turning a design into a running system on the target platform.

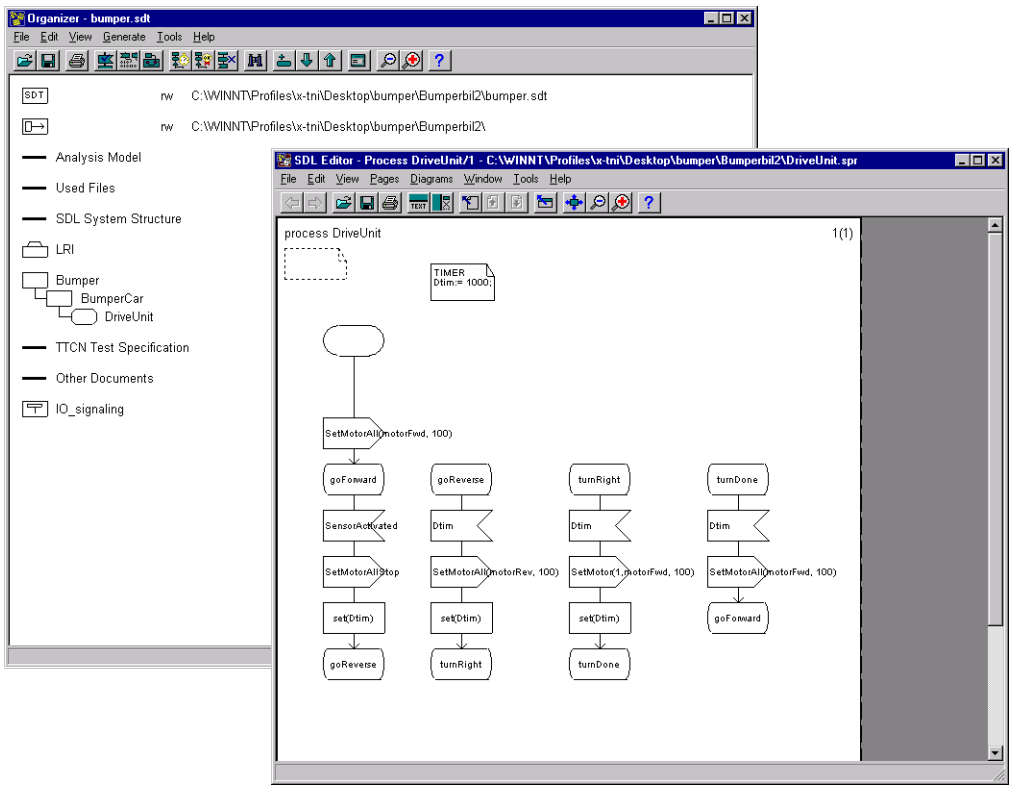


Figure 2 The Organizer and the SDL Editor

- The SDL Editor and Organizer

  The SDL Editor and Organizer is used to enter the specifics of your SDL system, and edit the relations between the different parts. Shown in *Figure 2*.

- The SDL Analyzer

  The SDL Analyzer is used to analyze the SDL system you have entered. Checks for syntactic and semantic errors.

- SDL to C Converter

  SDL to C Converter converts the analyzed SDL application into C code, that may then be compiled to an executable. The C compiler is specific to every platform.

Other advanced tools are available, such as the Simulator and the Validator.

2.2.1 Targeting Expert

To be able to easily adapt SDL system code produced with Telelogic Tau SDL Suite to a platform, a tool called Targeting Expert has been introduced into the SDL Suite. It is a graphical interface that allows you to specify the details of you compiler and scale your system.

This is a rather new feature, and one reason why this project was started, was to investigate if the Targeting Expert would be usable for adapting to a platform like LEGO Mindstorms RIS.

2.2.2 SDL Access

SDL Access is an interface between the SDL analyzer and the code generator. It can be used to write a new code generator, without having to develop a new analyzer. The data generated by the analyzer can be used as input to a different code generator.

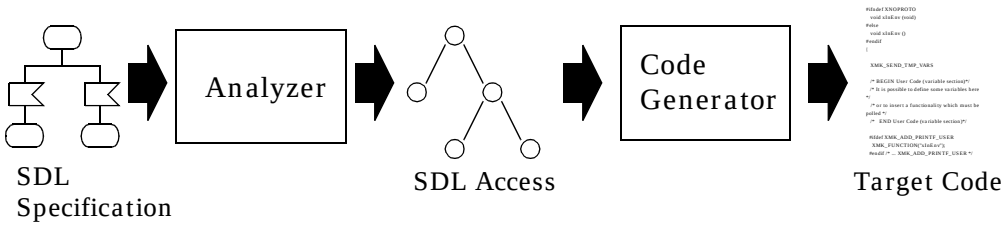


Figure 3 The SDL to target code process

2.2.3 SDL Kernel

The program downloaded to the target platform consists of two parts: The SDL kernel and the SDL program: the SDL kernel can be considered a virtual machine that runs the SDL program.

There were two different kernel choices available to us: CAdvanced and CMicro. CAdvanced supports full SDL, whereas CMicro has several limitations (see *Chapter 4.3.2*). CMicro on the other hand is highly scalable, which leads to the possibility of a very small memory footprint. An approximation of the CMicro kernel's minimal memory usage would be 4kb, and for CAdvanced 20kb.

2.3 The RCX

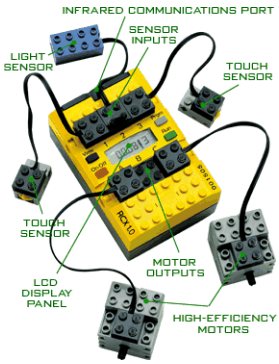


Figure 4 The RCX with motors and sensors

The RCX is the main component of the LEGO Mindstorms Robotics Invention System (RIS). At the core of the RCX is a Hitachi H8 micro controller with 32K of off-chip RAM. The micro controller is used to control three motor ports, three sensor ports, and an infrared serial communications port. A battery sensor is also built into the RCX.

Figure 5 Base System

The base system for using the RCX consists of the RCX itself, an infrared transceiver (the IR tower), and a PC. Additional components, such as motors, sensors, and other building elements, combine with the base system to allow the creation of functional autonomous robotic devices.

2.4 Firmware / Languages

The software that is associated with the RCX usually consists of two parts: The operating system or firmware, and the user programs. There are several firmwares available for the RCX. The most relevant ones are listed below.

2.4.1 LEGO

The firmware that is part of the original LEGO Mindstorms RIS package is a byte code interpreter. Not only LEGO's development environment can be used

to produce the byte code. Several other options are available, but the most relevant to our project is NQC (Not Quite C). NQC is an open source compiler for a C like language. The compiler does not support ANSI C.

2.4.2 legOS

legOS is an independent RCX operating system. It offers preemptive multitasking, energy saving, dynamic memory management, POSIX semaphores, native access to display, buttons, IR communication, motors and sensors.

legOS is an open source project, and is first and foremost available for Linux, but it has also been ported to other UNIX platforms and Windows 9x/NT.

Since the GNU C compiler (gcc) has been adapted to cross-compile to Hitachi H8 format, real ANSI C can be compiled and run on legOS. LEGO uses graphical building blocks instead of a text-based language.

As a replacement for the LEGO Mindstorms original firmware, the following advantages are gained with legOS (*[3]*).

- User tasks are executed as native code, not interpreted.
- Instead of the 32 variables that are offered in LEGO firmware, the whole 32kb of RAM are available.
- Priority-based preemptive multitasking.
- Real process synchronization with POSIX semaphores.
- Fine-grained hardware control: complete LCD control, raw mode IR.
- Floating point math.

2.4.3 Java firmware - TinyVM

Another interesting firmware replacement is the TinyVM. It runs a Java virtual machine on the RCX and is programmed in Java. This of course brings along all the advantages and disadvantages of regular Java programming, plus some specific to TinyVM.

- No garbage collection
- No floating point support
- No switch statements
- String constants are ignored
- No auto-power-off

Read *[8]* for further details.

2.5 Positioning

Since the robot can be mobile it is interesting to be able to estimate its position. This can be done either by the robot itself or by using external devices. Internal positioning, or dead-reckoning can be done by using information

about direction and speed, rotation sensors etc. External positioning can be done by using ultrasound, radio, laser etc. The internal estimations are less accurate and the error in location increases with time, while the external positioning is more accurate and the error is constant.

A research project at the Department of Automatic Control at LTH about positioning with ultrasound for RCX was examined.

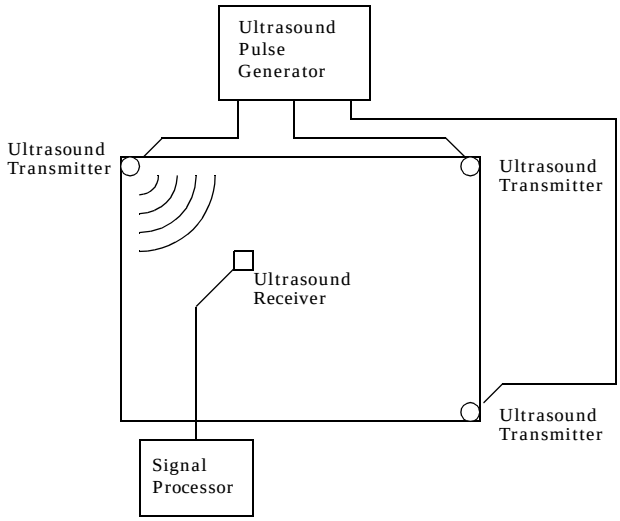


Figure 6 Ultrasound positioning

The ultrasound positioning in that project was done using 3 transmitters (speakers) on 3 corners of a table sending pulses at different times. A detector (microphone) was placed on the mobile unit that needed to be positioned. The detector was connected by wire to a computer that calculated the position based on the pulse time delay from transmitter to detector. In addition to calculating X and Y coordinates, a quality measurement of the positioning is given. This is based on the amplitude of the received pulses and makes it possible to discard positioning errors when pulses have not been properly received.

The data is passed to the RCX through a sensor port. Each value is transferred as 16 bits. They are encoded as follows.

Each value consists of 4 pulses and each pulse has 16 levels (4 bits per pulse). The values are sent consecutively, with a 3 ms low signal sent first for synchronization.

In theory this results in a maximum data transfer rate of 1.2 kbits/second. In practice, a transfer rate of 0.8 kbits/second is acheived, since the the values are sent every 60ms, and not as soon as the last set of values are sent.

The plans for the research project was to build a chip with the signal processing functionality in it. This chip would be attached to the RCX and that would make it possible for each RCX to independently determine its position.

Unfortunately this project was not completed in time to take advantage of its results.

2.6 Communication

Since SDL is a language foremost developed for telecommunication applications, the possibilities of communication between LEGO robots, and between the computer and the LEGO robots should be explored.

2.6.1 IR Communication

IR communication is a cheap way to transfer data, but it has many drawbacks. It has limited range, limited data rate and is sensitive to interference. It also requires that the parties communicating can “see” each other, since it is based on infra-red light.

The IR communication in this project is between the IR tower and the RCX as seen in *Figure 5*. The IR tower is connected to a serial port on the computer. The IR protocol associated with sending a “message” to the RCX is simple. The bit encoding is 2400 baud NRZ (Non-Return to Zero) with one start bit, 8 data bits, odd parity and one stop bit. A ‘0’ bit is encoded as a 417µs pulse of 38kHz IR and a ‘1’ bit as 417µs of nothing.

The selection of higher level protocols available are directly linked to the choice of firmware. Each firmware has its own set of protocols.

2.6.2 Radio Communication - Bluetooth

Another master thesis (conducted by Johan Brodin at Sigma Exallon, in cooperation with the Department of Communication Systems at LTH) was using Bluetooth for communication with the RCX. The result was a Bluetooth circuit that used the RCX as a motor controller. The Bluetooth chip on the robot receives a signal that is passed on to a sensor port of the RCX and the RCX transfers that value to a motor port.

For this project to be relevant to this master thesis, it should have had more advanced RCX to Bluetooth communication possibilities. The data transfer possible using the sensor ports have in practical tests been shown to take up too much processor time, although the data transfer capability is sufficient for message transfer. It is not, however, enough to warrant a replacement of the IR data transfer.

3 Solution

This chapter provides an overview of the path taken when we were examining and developing the project. More thorough descriptions are available in the appendices.

3.1 Decisions

During the course of the project several decisions had to be made on which direction to take.

3.1.1 Development Platform Selection

LegOS is foremost developed for Linux, but since an easily installable Windows port was found, and since Windows NT and Windows 95 are the most common teaching and demonstration platforms, Windows NT was chosen as our development platform.

3.1.2 Firmware and Language Selection

The choice of firmware was obvious, once the testing started. LegOS turned out to be the only option if the C code that was generated by Telelogic Tau SDL Suite was to be compiled.

If legOS had not been able to compile ANSI C, the next step would have been to write a code generator using SDL Access (see *Chapter 2.2.2*). This would have meant that the project would have consisted mostly of that task, and not even that task would have been finished.

3.1.3 SDL Kernel Selection

Two options were available: CAdvanced and CMicro. CAdvanced has a minimum memory requirement of approximately 20kb and CMicro can be scaled down to as little as 4kb (see *Chapter 2.2.3*). With only 21kb available for the system (see *Chapter 4.2*) CMicro was chosen.

3.1.4 IR Communication

The protocol for the IR communication was not really a choice. Each firmware had their assigned protocol. LNP, the legOS IR network protocol, is described further in *Appendix A.3*.

3.2 Implementation

This section describes the actual implementation, based on the previous decisions.

3.2.1 Adaption of legOS

To simplify the installation process on Windows NT, the distribution of legOS called WinlegOS was chosen. This came as a prepackaged installation file, and made installing on Windows much easier.

To get legOS to work with Tau, some adjustments of the original distribution had to be made. These are listed in *Chapter 4.4*.

3.2.2 Adaption of Telelogic Tau SDL Suite code

To get Tau to compile an SDL system with legOS, the Targeting Expert was used to select compiler, linker and all other legOS specific options. See *Appendix C.1* for details on this. The modifications of the CMicro kernel to work in the legOS environment had to be done by hand, but these were straightforward, since most of the things needed in the kernel were already implemented in legOS. See *Appendix G* for details on this.

Because the thread handling in legOS is still in its infancy, the SDL system is run as one thread on the RCX. The scheduling of the processes is handled by the CMicro kernel.

3.2.3 Environment file

The environment file is really a part of the CMicro kernel, but it deserves to be mentioned separately. This is the file where the actual signals between the SDL system and the legOS environment are handled.

The environment file, named env.c in the actual system, is automatically generated each time the system is compiled. Manual changes are kept, if they have been made within the user code marked areas.

The file consists of four functions.

- xInitEnv

  This function is called by the kernel during initialization of the SDL system. It is used to initialize the environment.

- xInEnv

  This function is called by the kernel continuously to poll the environment and generate signals to the SDL system.

- xOutEnv

  This function is called by the kernel if an SDL signal is to be sent to the environment.

- xCloseEnv

  This function is called by the kernel at termination. Used to shut down the environment.

The signals out from the SDL system are implemented in the xOutEnv function that is called when a signal is to be sent. Implementing these are relatively easy, and only requires that you fill in the code to be executed in the correct spot.

With signals from the environment there are two different ways of handling them: polling and interrupts.

The polling alternative means that the signal handling is implemented in the xInEnv function. Every time xInEnv is executed, an if-statement is checked. If true, the signal is put in the signal queue. This alternative is usually sufficient when used with the basic touch sensor, or other “slow” sensors. Other functionality that needs more real-time handling, such as the reception of IR transmissions, must be handled with interrupts.

When using interrupts, a signal is inserted in the signal queue when the interrupt is triggered. A function is executed that inserts this signal. This of course raises questions of thread safety and possible data corruption if the queue is accessed by the CMicro thread and the interrupt thread simultaneously. This is handled in CMicro by using critical paths, where the interrupts are turned off, and turned on again once the critical path ends. For further details on interrupts, see *Appendix A.2*.

3.2.4 Hardware Abstraction

A hardware abstraction is a way of hiding the inner workings of the hardware from the intended user. This makes it easier for the user, since he does not have to bother with hardware internals, and he can concentrate fully on the programming task at hand.

The hardware abstraction should have as little advanced and intelligent functionality as possible, because of the memory constraint.

Only the basic signals to and from the environment are defined in the SDL package diagram LRI (see *Appendix D.1*) and the environment file env.c only contains the code for legOS function calls that are needed to implement those signals. It is fast, memory efficient and uncomplicated.

4 Integration of SDL and legOS

This chapter describes the interaction between the CMicro kernel and legOS. It addresses compiler issues, performance metrics, limitations etc.

4.1 Performance Metrics

The kernel loop metrics in *Table 1* were obtained with an SDL system containing one process, one timer and an out-signal as a response to every timeout or in-signal.

Table 1 Kernel loop metrics

	no signals to process	1 signal/loop to process
1000000 loops	20 s	60 s
1 loops	0.02 ms	0.06 ms
loop/ms	50	17

In every loop, a function called xInEnv in the environment file env.c is executed once to check whether or not any signals should be sent from the environment to the SDL system.

The “1 signal/loop” metrics were obtained by sending a signal to the SDL system every time the xInEnv function was executed. The “no signals to process” metrics were obtained by letting the system run through the xInEnv function without sending any signals to the SDL system.

To measure the number of loops that were executed, a beep was emitted every 1 000 000th time the xInEnv function was executed.

4.2 Memory Metrics

The RCX has access to 32 kb of memory in total. To measure how much memory was used by legOS, a small program was written to count the unallocated memory. The function used to measure memory can be found in *Appendix F.1*.

The memory left when running just legOS was 21 kb. This means that 11 kb of the 32 kb are being used by legOS.

The example system in *Appendix D* was compiled and download to the RCX. The memory left when running the system was 10 kb. The SDL system takes up approximately 11 kb. An illustration of the memory image is shown in *Figure 7*.

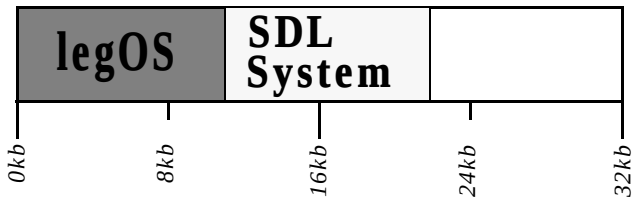


Figure 7 RCX Memory Image

Because both the legOS firmware and the CMicro kernel are highly scalable and very dependent on each other, these measurements are just examples. The results may vary greatly between different configurations of CMicro and legOS.

4.3 Limitations

The combination of legOS, Telelogic Tau and the CMicro kernel brings on a few limitations that may well cause serious trouble, unless one is aware of them. These limitations are listed below, grouped according to source of the limitation.

4.3.1 General Limitations

Reserved legOS names and definitions can not be used within the SDL system.

4.3.2 CMicro Limitations

The following are not allowed when using the Cmicro SDL to C Compiler:

- Inheritance of procedures
- Procedures with states
- Remote Procedure Calls
- Nested procedure call data scope
- Macros
- Export / Import

- View / Reveal
- Enabling condition / Continuous signal
- Service and priority input and output
- Channel substructure
- Declaring an infinite number of process instances (x,) or (,)
- FPARS when creating a process
- Omission of parameters in a signal input
- Output via all
- Timers duration values cannot be real
- Timers with more than one parameter
- Timers with another parameter than sort integer
- The any expression
- Names of processes within different blocks must be different

None of the limitations listed above significanlty impact the functionality of the systems created for use with legOS. The ones that have impact on the programming can be circumvented by using other SDL constructs.

4.4 legOS Modifications

To make CMicro and the SDL system C code generated from Tau work with legOS and the RCX some changes had to be made to the prepackaged WinLegOS distribution.

- strcat - string concatenating function. add second string to the end of the first
- abs - returns the absolute value of the input value

Abs was technically not implemented in the legOS code, but rather in the file that needed it. It is done in a macro fashion in sctpred.c.

4.5 CMicro Modifications

The following files have to be changed to fit the target platform.

- mk_cpu.c

  Character read / write on hardware / OS-level
  Functions for dynamic memory handling

- mk_stim.c

  Timer functions

- mk_user.c

  Functions for central errorhandling
  Functions for the Cmicro Kernel (optional in order to optimize)
  Functions to trigger a hardware-Watchdog
  Function for Call Stack

- ml_mem.c

  a dynamic memory management
  some string and memory functions.

- sctpred.c

  implementation of operators and the read and write functions for the predefined data types (except PId)

For the details on these changes see *Appendix G*.

5 Conclusion

5.1 Conclusion

The actual results from this master thesis are the hardware abstraction LRI, examples using LRI and a target adaptation usable with LEGO Mindstorms Robotics Invention System (see *Appendix D*).

Telelogic Tau SDL Suite is an adaptable tool, and the Targeting Expert is very useful in target adaptations.

LEGO Mindstorm Robotics Invention System is usable as a demo and teaching platform, even though it has its shortcomings, such as the limited debugging facilities.

5.2 Further work

Things we did not have time to finish or perhaps even start:

- Ultra sound positioning

  Ultra sound positioning enables the robots to know their current position. The applications can be made more interesting and the behaviour could be made more advanced. Robots would be able to interact with precision and solve tasks together.

- Bluetooth

  Bluetooth has a high hype-factor at the moment and can be used to create more media interest in demonstrations. It can be used for more reliable and faster communication and it is not as sensitive to obstacles in the surroundings as is IR-communication.

- Implementing the Target Tester

  Implementing the Target Tester would enable tracing over the IR link and other forms of debugging. The estimated time needed to implement it is 3 weeks.

- SDL System on a PC communicating with the RCX

  Using WinLNP, a port of the legOS IR communication protocol, an SDL system on a PC could communicate with the RCX. This would require the use of the C++-to-SDL converter available in the Telelogic Tau SDL Suite. The C++-to-SDL converter makes it possible to instantiate C++ objects and use them in the SDL system.

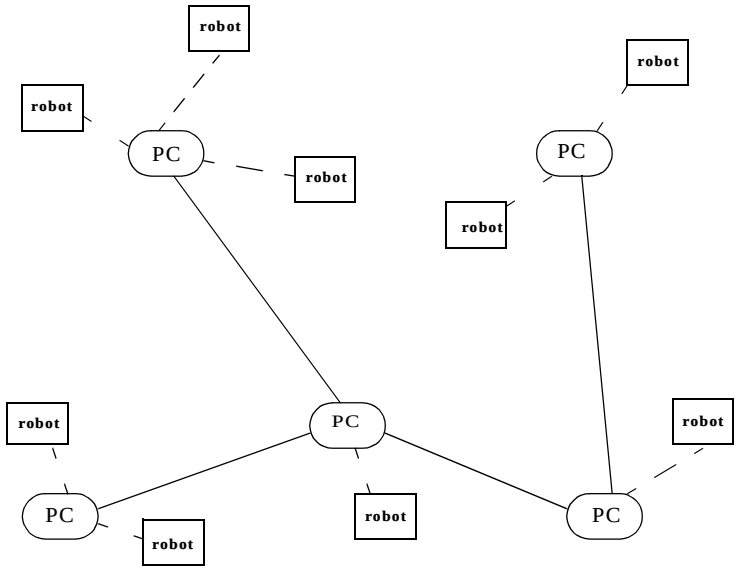


Figure 8 Model of large demo

- A larger demo

  A suggestion for a larger demo application could be to use mobile phone network as model for the communication. The robots communicate via the closest PC. Several PCs are in contact with each other and keep track of the robots. This way the robots connected to different PCs can communicate with each other. This would require the functionality of the previous paragraph. Once that has been implemented, this demo would require about 2 weeks for completion.

6 Glossary

RCX	The heart of the LEGO Mindstorm Robotics Invention System. It contains a Hitachi H8 processor, a display, an IR port, sensor ports and motor ports. See *Chapter 2.3*.
SDL	Specification and Description Language. Formal programming language. See *Appendix E*.
Telelogic Tau SDL Suite	A collection of tools for developing and testing SDL applications.
CMicro Kernel	A version of the SDL Kernel specifically designed for embedded systems. Has a smaller memory foot-print than the CBasic and CAdvanced kernels.
legOS	Open source freeware operating system for the RCX. The package contains an ANSI C compiler (gcc)
LNP	Layered Network Protocol. Used for IR communications between LEGO Mindstorm robots running legOS and the IR tower.
firmware	The term used in this report for the operating system on the RCX.
IR Tower	An IR transceiver that connects to a serial port of a PC. It is used to communicate between the PC and the RCX.
RIS	Abbreviation of Robotics Invention System
MSC	Message Sequence Chart

References

6.1 Books

[1] Ellsberger, Hogrefe & Sarma, (1997). *SDL Formal Object-oriented Language for Communicating Systems*. Prentice Hall Europe. ISBN 0-13-632886-5

[2] Knudsen, (1999). *The Unofficial Guide to LEGO MINDSTORMS Robots*. O'Reilly. ISBN 1-56592-692-7

6.2 Web Addresses

[3] Markus L. Noga (5 October 2000), *Official site of legOS*
http://www.noga.de/legOS

[4] (5 October 2000), *Windows based LNP COM library*
http://www.geocities.com/winlnp

[5] (5 October 2000), *LegOS discussion forum*
http://lugnet.robotics.rcs.legos

[6] (5 October 2000), *NQC official home page*
http://www.enteract.com/~dbaum/nqc

[7] (5 October 2000), *legOS development home page*
http://legos.sourceforge.com

[8] (5 October 2000), *Java firmware home page*
http://sourceforge.net/projects/tinyvm

Appendix A
LegOS Details

Selected details of the legOS operating system. See [3] and [5] for further details.

A.1 Threads

To use the threading functionality in legOS, the following function is used:

```
pid_t execi(&function_name, int argc,
       char argv, int prior, DEFAULT_STACK_SIZE)
```

It calls the function *function_name* and assigns it the priority *prior*. *execi()* immediately returns a process id (*pid_t*). The function *function_name* is run as a thread. The function *function_name* must take an *int* and a ***char* as parameters, even if they are not used. They will get assigned the values of argc and argv.

The function *kill(pid_t)* allows you to kill a thread, using its process id (*pid_t*).

An example of thread usage is shown below. Two threads *thread1* and *thread2* are executed, and display the strings *t1* and *t2* respectively, with a given interval.

```
void thread1(int argc, char** argv) {
   while(1) {
      cputs("t1"); /* Display "t1" */
      msleep(1200); /* sleep for 1200 ms */
   }
}

void thread2(int argc, char** argv) {
   while(1) {
      cputs("t2"); /* Display "t2" */
      msleep(800); /* sleep for 800 ms */
   }
}

int main() {
```

```
    /* Execute thread 1 */
    pid1 = execi(&thread1, 0, NULL, 1, DEFAULT_STACK_SIZE);
    /* Execute thread 2 */
    pid2 = execi(&thread2, 0, NULL, 2, DEFAULT_STACK_SIZE);
    return 0;
}
```

A.2 Interrupts and Thread Safety

When, for example, an IR packet is received, the current program flow is interrupted and a function is executed. This can lead to data corruption if the current program and the interrupting thread are manipulating the same data.

In legOS this has been addressed by the functions disable_irqs() and enable_irqs(). disable_irqs() disables interrupts and enable_irqs() enables them again. These functions can be used during critical phases in the main program, to make sure that no data corruption occurs.

A.3 LNP - the legOS IR network protocol

IR is a relatively insecure way to transfer data, and in the LNP protocol there is no way of knowing whether your package reached its destination safely or not. This can of course be solved by implementing a transport protocol on top of the LNP, where timeouts and acknowledgements would be involved. The following information has been gathered from *[5]*.

There is only one type of node in the LNP network, and it is referred to as a host. The RCX and the PC are peers.

The basic function of the LNP is this: The packets are sent with a simple function call, and are received by registering a listener, a function that is executed when a packet intended for that listener is received.

There are two available types of transmission: Broadcast and addressed.

A.3.1 Broadcast - integrity packet

This type of transmission is used to transmit data that should be received by everyone who is within range. There is no built-in way to find out from which host an integrity packet came. This can be solved by putting the source address in the data section.

F0: Identifies this as an integrity packet
LEN: length of IDATA section (1 to 255 bytes)
IDATA: Payload data
CHK: Checksum

Figure 9 The Integrity Packet Format

The integrity packet can be received by any host who has a integrity packet handler registered. Such a function is registered like this:

```
lnp_integrity_set_handler(my_integrity_handler);
```

For this to work, you have to declare the handler as a function. The name my_integrity_handler is just an example. It could be any name.

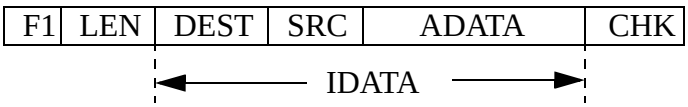
```
void my_integrity_handler(const unsigned char *data,
                          unsigned char length) {
     // Insert functionality here, as long as it is non-blocking
     // *data is the pointer to the incoming data
     // length is the length of the data
}
```

To send integrity packets use the function

```
lnp_integrity_write(data, length);
// const unsigned char * data - pointer to the data
// unsigned char length - the length of the data
// return non-zero if a collision was detected
```

A.3.2 Addressed - addressing packet

This type of packet is used for addressing a specific LNP host, and is received only by the host with the specific address, and only if the host has a handler registered to the specific port. The address is set in the firmware (see *Appendix A.3.3*).



F1: Identifies this as an addressing packet
LEN: length of IDATA section (1 to 255 bytes)
DEST: Destination address (2 bytes)
SRC: Source address (2 bytes)
ADATA: Payload data
CHK: Checksum

Figure 10 The Addressing Packet Format

The handler for addressing packets is set like this:

```
lnp_addressing_set_handler(port, my_addressing_handler);
// unsigned char port - the port number assigned to this handler
// lnp_addressing_handler_t handler - the addressing handler
```

The handler is declared like this

```
void my_addressing_handler(const unsigned char *data,
                           unsigned char length,
                           unsigned char src_address) {
// Insert functionality here, as long as it is non-blocking
// *data is the pointer to the incoming data
// length is the length of the data
// src_address is the address from where the packet came
}
```

The addressing packets are sent with a call of this function

```
int lnp_addressing_write(data, length, dest, srcport);
// const unsigned char * data - pointer to the data
// unsigned char length - the length of the data
// unsigned char dest - destination address
// unsigned char srcport - source port
```

A.3.3 Setting the LNP address

The LNP address of an RCX is set in the legOS firmware. This means that when the address has to be changed, the legOS has to be recompiled. The address is set in the file *<legOS dir>/Boot/config.h*.

`#define CONF_LNP_HOSTADDR 0x10 //!< LNP host address`
of this file determines the address. The first digit (hexadecimal) is the address, and in this case, the address is 1. To recompile, run the make command in the same directory. See *Appendix B.2* for details on this.

A.4 Sensors

Sensors can be read in two ways: Directly, like you would check the value of any variable, or via interrupts, where you assign a function that detects when a certain level has been reached.

A.4.1 Direct

The direct value of sensor port 1 can be read like this

```
if (SENSOR_1<0xf000) {
     // What needs to be done
}
```

A.4.2 Interrupts

To avoid busy wait, interrupts should be used. Interrupts are not very fast, so if millisecond resolution is required, busy wait is the only option. Interrupts are declared like this

```
wakeup_t sensor_press_wakeup(wakeup_t data) {
     return SENSOR_1<0xf000;
}
```

and used like this

```
wait_event(&sensor_press_wakeup,0);
// Holds here until sensor_press_wakeup(...) returns true
// 0 is the argument passed to sensor_press_wakeup(...)
```

Appendix B
LEGO SDL Development Kit

This appendix contains a description of the SDL development kit, and instructions for its use. The kit has been developed for Telelogic Tau SDL Suite 4.0.

B.1 Contents of the Development Kit

The files necessary to get the example system running are:

- **WinLegOS** 1.0 (legOS 0.2.3) installation file.
- **Perl installation**. Used for recompilation of the legOS kernel.
- **LFN** allows long file names to be used in Windows NT. It is needed to recompile the firmware, and if a legOS program is compiled manually. It is not needed when the Targeting Expert is used.
- **Precompiled firmware binaries** with the addresses 1 and 2 respectivly.
- **Example system files**. See *Appendix D.2*.
- **Adapted files**. The set of files that need to be changed for CMicro to work with legOS and vice versa.

Other files included:

- **WinLNP** with precompiled binary. A PC version of the legOS LNP IR protocol.

B.2 Installation and Usage

1. **Install Perl**. Perl is needed to recompile the legOS firmware.
2. **Install WinLegOS** to C:\legOS

3. **Copy the changed legOS files**. Copy the content of the *legos_mod_files* directory to their respective directories.
4. **Start a command prompt**
5. **Recompile the libraries** by changing directory to <legOS dir>/lib and typing *make*.
6. **Recompile the firmware** by changing directory to <legOS dir>/boot and typing *make*. This is not necessary if you are using the precompiled binaries.
7. **Download the firmware** to the RCX. This is done using the program *firmdl3.exe* that is a part of the legOS distribution. Make sure only one robot is switched on while downloading the firmware.

```
c:\legos>firmdl3
usage: firmdl3 [options] filename
      --debug      show debug output, mostly raw bytes
  -f, --fast       use fast 4x downloading (default)
  -s, --slow       use slow 1x downloading
      --tty=TTY    assume tower connected to TTY
  -h, --help       display this help and exit
```

8. **Unpack the example system** to a location of your choice
9. **Start Telelogic Tau SDL Suite** and open the example system
10. **Start the Targeting Expert**
11. **Make the system**
12. **Download the system**. If the example system is used, this should be done automatically by the Targeting Expert. It can be done manually by using the program *dll.exe*. The *dll.exe* program, as opposed to the *firmdl3.exe*, downloads the program to a specific address, so make sure that the correct address of the RCX is specified.

```
c:\legos>dll
usage: dll file.lx
[-rrcxaddress -pprogramnumber -ssrcport -v]
```

Appendix C
Targeting Expert Setup Instructions

This appendix contains the instruction to set up the Targeting Expert

C.1 Targeting Expert

No presets were used during the setup up targeting expert to the legOS platform. The legOS gcc compiler was given the name GCCLEGOS. Information from the make files included in the legOS distribution were used to figure out how to use the compiler with Targeting Expert. Since the legOS linker uses a three step process to generate the executable, a separate script was written for this (See *Appendix F.2*).

Besides providing means to adapt to a new compiler and platform, the targeting expert provides numerous possibilities to scale the kernel to suit the needs of the system. Functionality can be added and removed with the click of a button.

C.2 Creating a New System

1. Create a new directory <SystemName>
2. Start up SDL Suite
3. Select Add New.../System. Call it <SystemName>.
4. Create your system. Define the signals going into and out of the system. (for example In: SensorActivated Out: DisplayInt(Integer))
5. Start Targeting Expert. In the box Integration, select <user defined>. Then choose “Add new compiler section”. Call the compiler GCCLEGOS as mentioned above.
6. Then select your kernel. CMicro should be used with legOS. Let

sdl_cfg.h be created when asked.

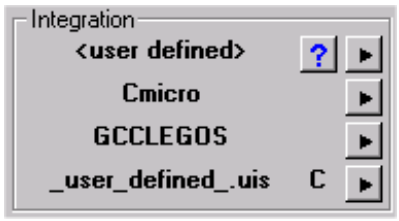


Figure 11 Integration Box

7. Add the compiler by Edit/Add Compiler. Compiler name and Compiler macro should be set to GCCLEGOS.
8. Select Edit/Edit Compiler Section, and alter the compiler section according to this
    - prototypes are handled correctly
    - const is handled correctly
    - critical path begin
      disable_irqs();
    - critical path end
      enable_irqs();
    - The following libraries should be added
      #include <stdlib.h>
      #include <string.h>
      #include <sys/irq.h>

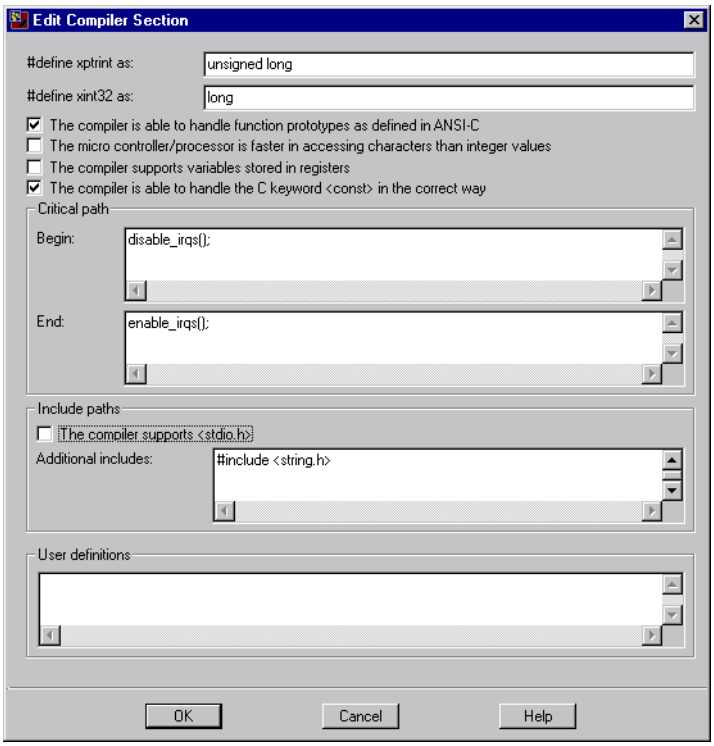


Figure 12 Edit Compiler Section Window

9. The compiler's switches and executables should be specified. Press the Compiler/Linker button in the Configuration group. Enter the path for the compiler. It should be C:\legOS\H8\bin\h8300-hms-gcc.exe, but it depends on where you have chosen to install legOS.

10. Add the necessary flags for the compiler

-O2	optimizer level
-fno-builtin	
-Wall	all warnings on
-fomit-frame-pointer	
%I	include files. A targeting expert variable
-c %s	source files. %s is a targeting expert variable
-o %o	object files. %o is a targeting expert variable

The full line should look like this
-O2 -fno-builtin -Wall -fomit-frame-pointer %I -c %s -o %o

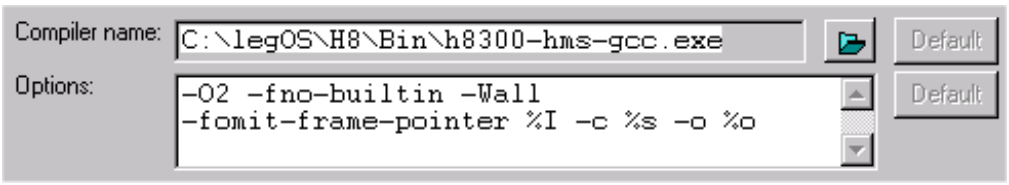


Figure 13 Compiler Name and Options

11. Include files should be added. During compilation these will replace %I in 10. Add these:

-I.	local includes
-I$(sdtdir)/cmicro/include	cmicro includes
-I$(sctkerneldir)	kernel includes
-Ic:/legos/boot	legos kernel includes
-Ic:/legos/include	legos includes

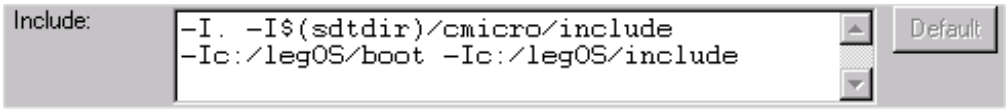


Figure 14 Include files

12. The following files have to be copied into the target directory and modified to fit our installation: mk_user.c, mk_stim.c, mk_cpu.c, ml_mem.c, sctpred.c. These files can be found in the LEGO SDL Development Kit (see *Appendix B*).

13. They should then be added to “User kernel and library files:” together with these files: $(sctkerneldir)\mk_main.c, $(sctkerneldir)\mk_sche.c, $(sctkerneldir)\mk_outp.c, $(sctkerneldir)\mk_tim1.c, $(sctkerneldir)\ml_mon.c, $(sctkerneldir)\mk_queu.c.

14. Select the Compiler Flag tab, and select GCCLEGOS

15. Define the linker. The linking script is usually located at C:\legOS\sdtutil\lnlegos.bat. Options should be set to “%O %e”. Executable extention should be “.lx”.

Figure 15 Linker Definition

16. Select the Make tab, and select microsoft NMake as your make tool. Add a name for your make file (makefile is good).

17. When starting from scratch in the targeting expert, several scaling selections are not yet defined. These have to be defined before a compilation is possible. These should be set as follows:
Kernel/Kernel Inititialization - Initialize Variables.
Kernel/Signal Structure - Receiver Process ID.
Kernel/Signal Queue - Combined static and dynamic queue.
Kernel/Error Handler - No error checks wanted.
Kernel/Reaction on warnings - End up in an endless loop
Kernel/Reaction on errors - End up in an endless loop.
Kernel/Timer queue - Use generated amount of timers.
Support of SDL/Character Strings - Limited Support
Environment/Environment - Initialize environment, Signals from environment, Signals to environment and Close environment.

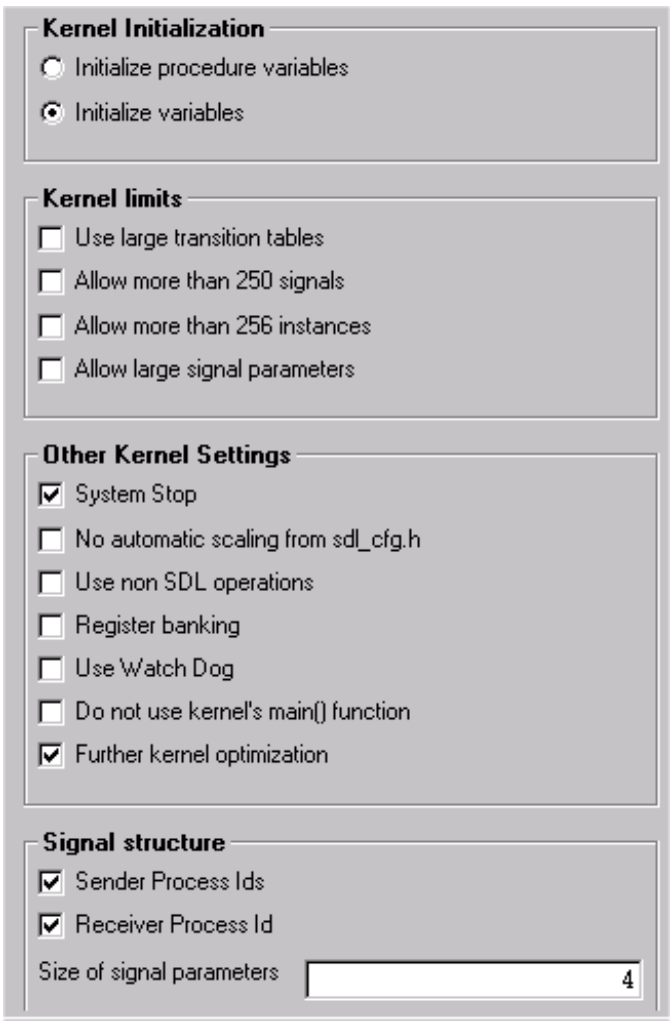


Figure 16 Kernel Part 1

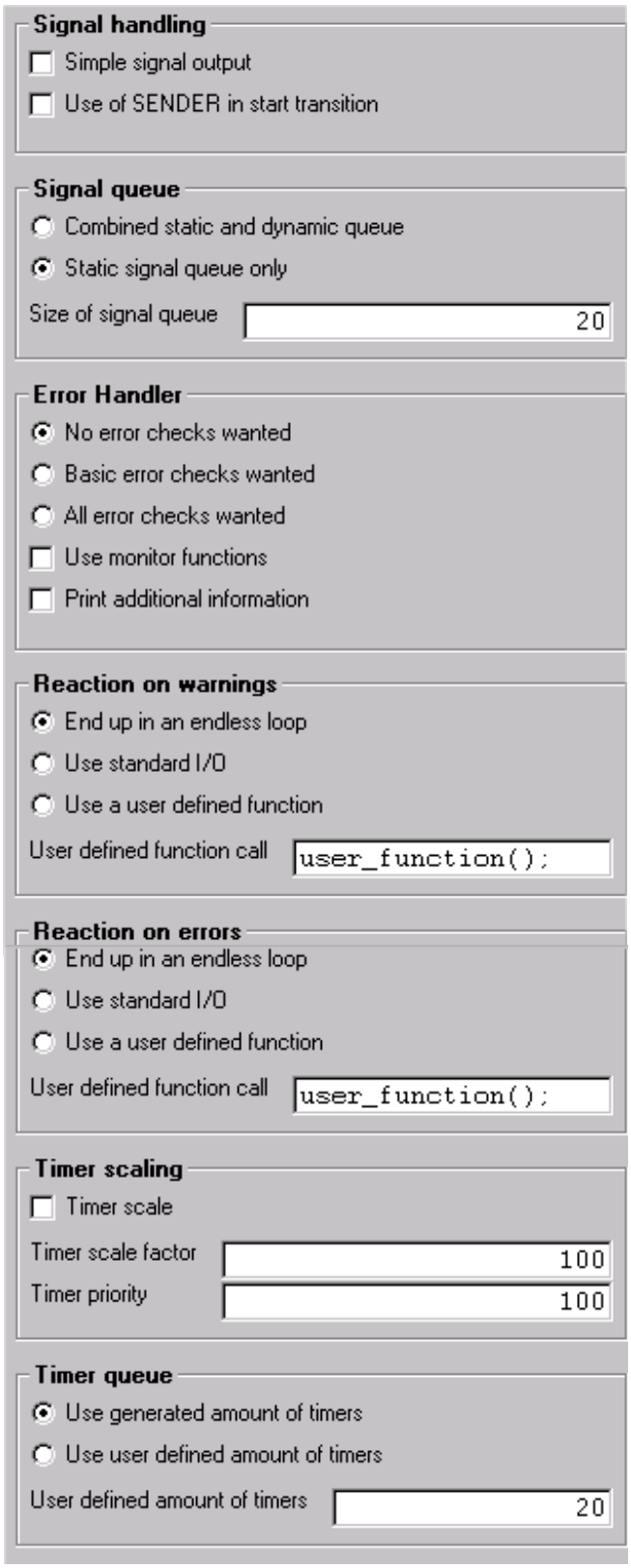


Figure 17 Kernel Part 2

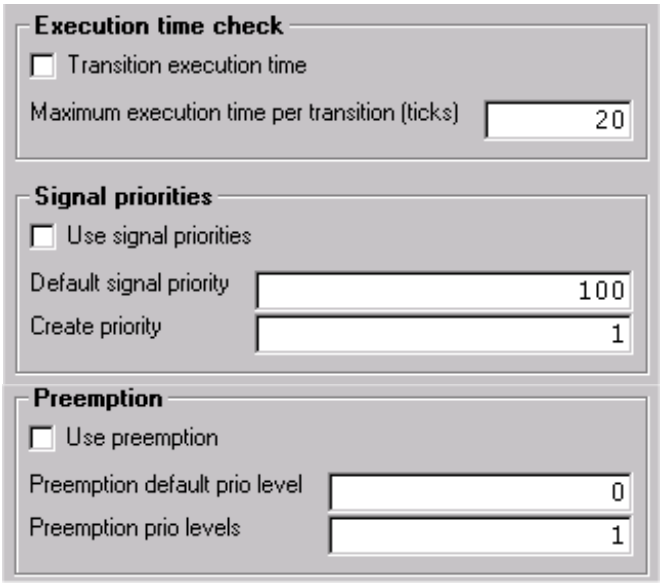


Figure 18 Kernel Part 3

18. At this time, a compilation of the system should be possible. During the compilation, the environment file env.c is generated. It should be modified to provide the correct functionality for each of the input/output signals.

19. An application can be defined for downloading to the RCX. Under Settings/Download Application, enter C:\legos\bin\dll %t, and check the box to download the executable after a successful compilation.

Appendix D
Hardware Abstraction

Contains the SDL package LRI, MSC diagram of the signals in the package LRI, the LRI environment file and a simple SDL example.

D.1 LRI - The SDL Package

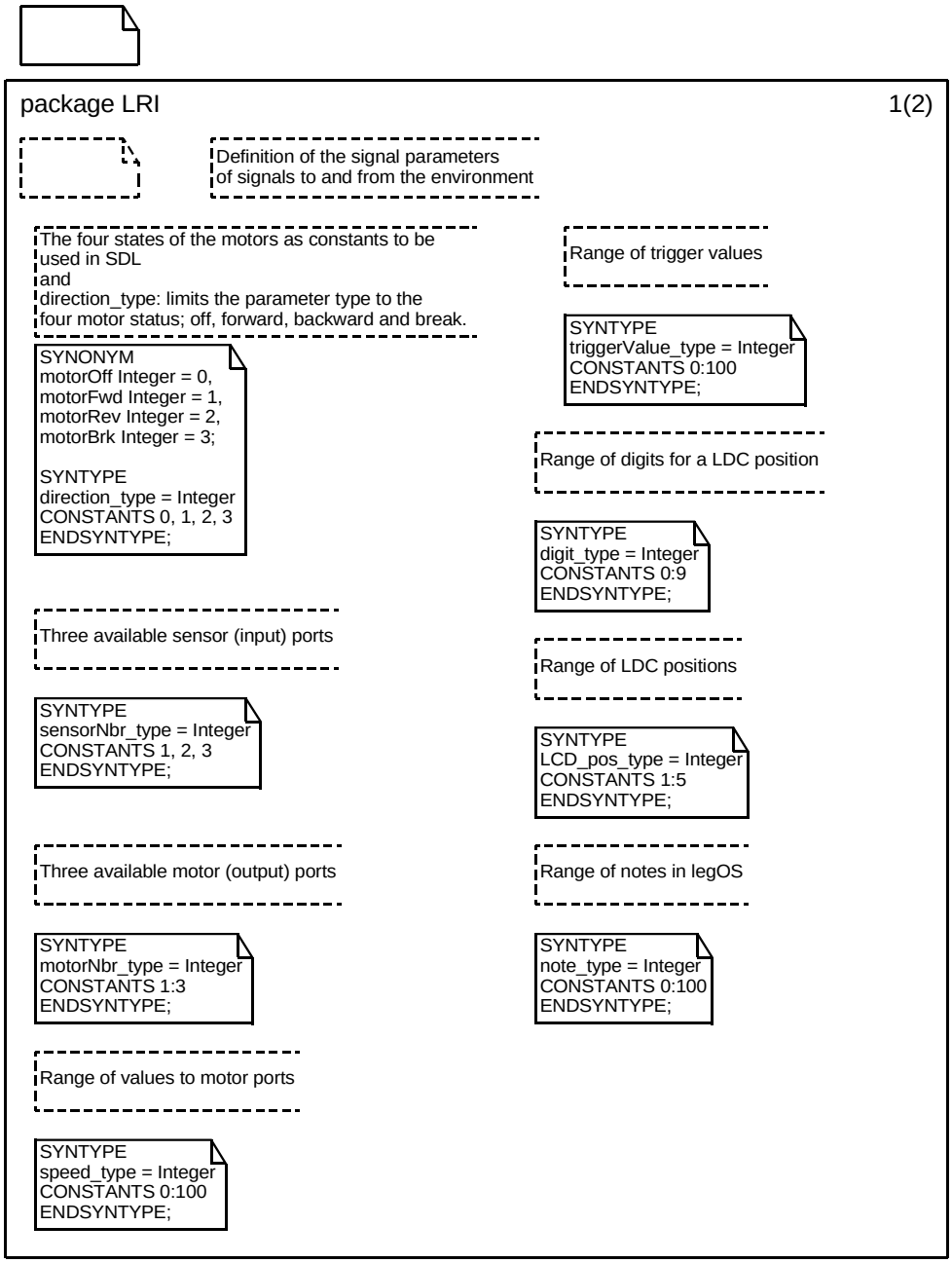


Figure 19 The LRI Package (page 1)

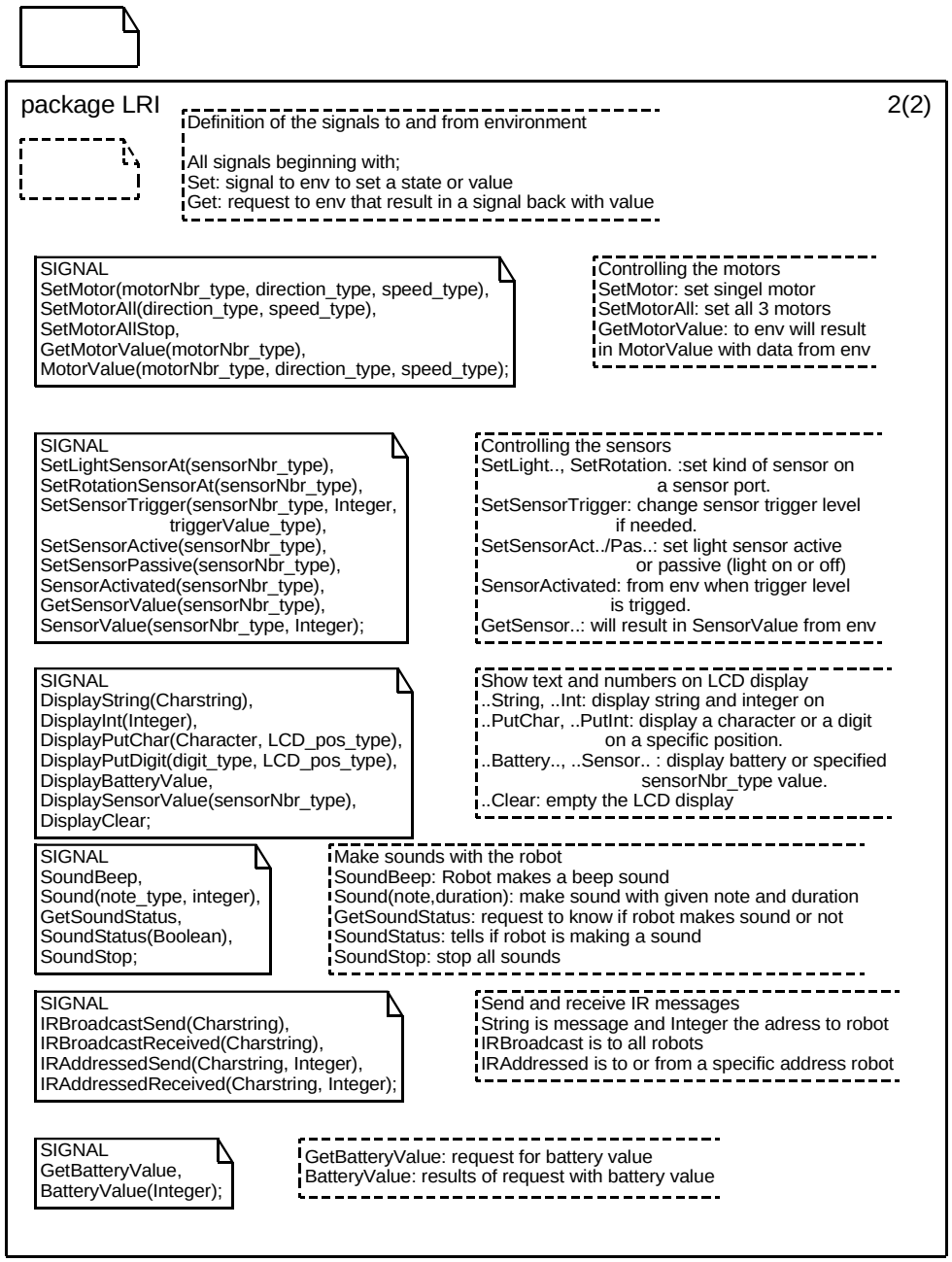


Figure 20 The LRI Package (page 2)

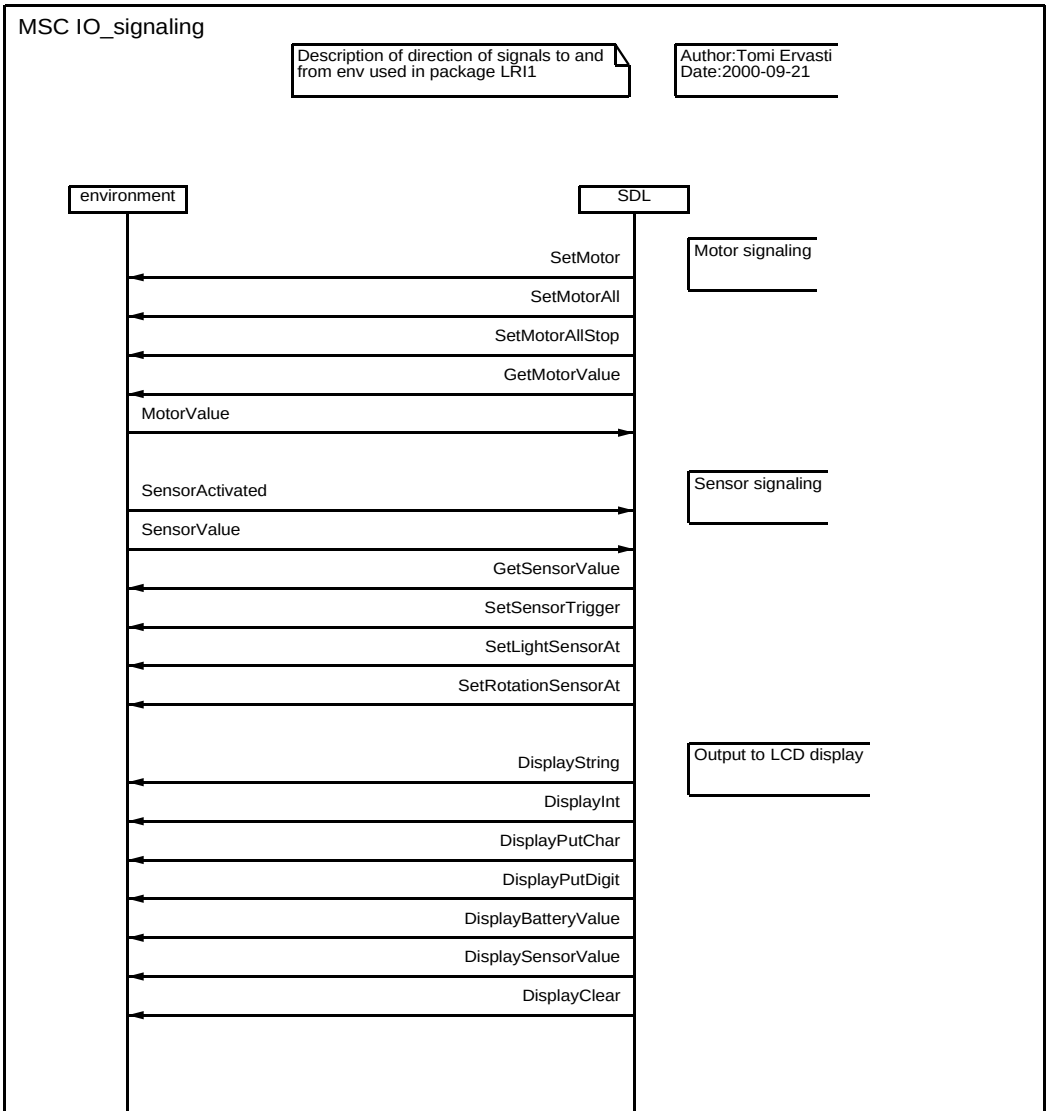


Figure 21 Message sequence chart (MSC) for signals in the LRI package (page 1)

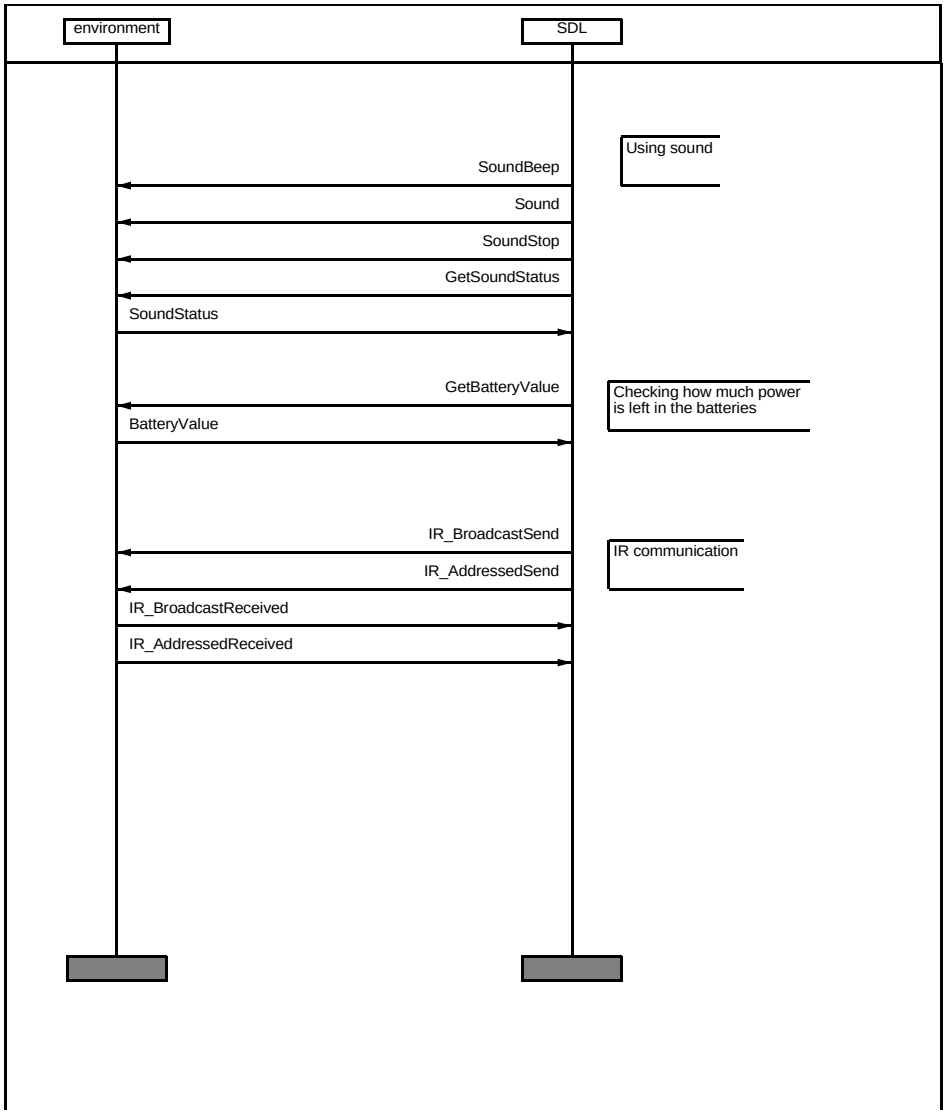


Figure 22 Message sequence chart (MSC) for signals in the LRI package (page 2)

D.1.1 Simple Example of Motor Control

To change the settings of motor 1, the signal *SetMotorValue(<motor number>, <direction>, <speed>)* is sent to the environment. The signal is taken care of in the *xOutEnv* function in the environment file. After checking the first parameter of the signal to determine which functions to use (in this example, the functions for motor A), two function calls are made where the second and third parameter of the signal are used:

```
motor_a_dir(dir);
motor_a_speed(speed);
```

Most signals to the environment are handled as easily.

D.1.2 Example of a Request

To find out the status or value of something in the environment a request signal (begins with *Get*) is used.

For example, to find out how much battery power is left, the signal *GetBatteryValue* is sent to the environment. The signal is taken care of in the environment file env.c by the function xOutEnv. The global legOS variable "BATTERY" is checked, and the response signal *BatteryValue(<parameter1>)* is created where *parameter1* is assigned the battery value.

Usually xInEnv is supposed to send signals into the SDL system, but to keep the environment file env.c as small as possible, the sending is done in xOutEnv when the requests can be solved simply by reading a global variable.

D.2 Example of Usage

The LEGO Mindstorms RIS configuration for this robot is the following:

- One touch sensor connected to sensor port 1.
- Two motors connected to motor ports A and C respectibly.

The robot moves forward until it hits an object, stops, backs up, turns right and continues to move forward.

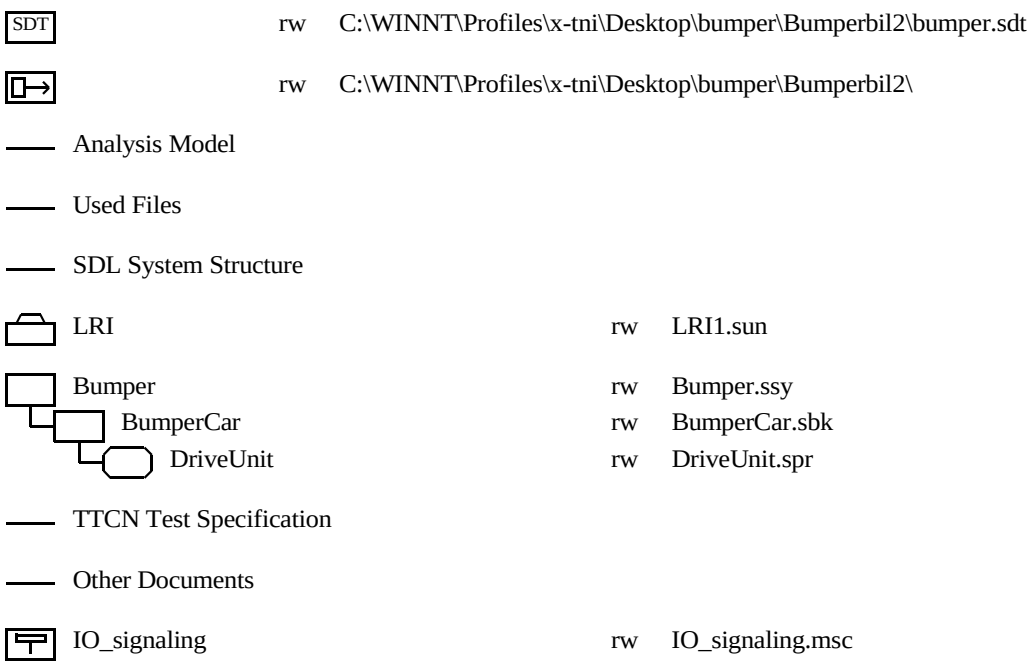


Figure 23 Organizer view

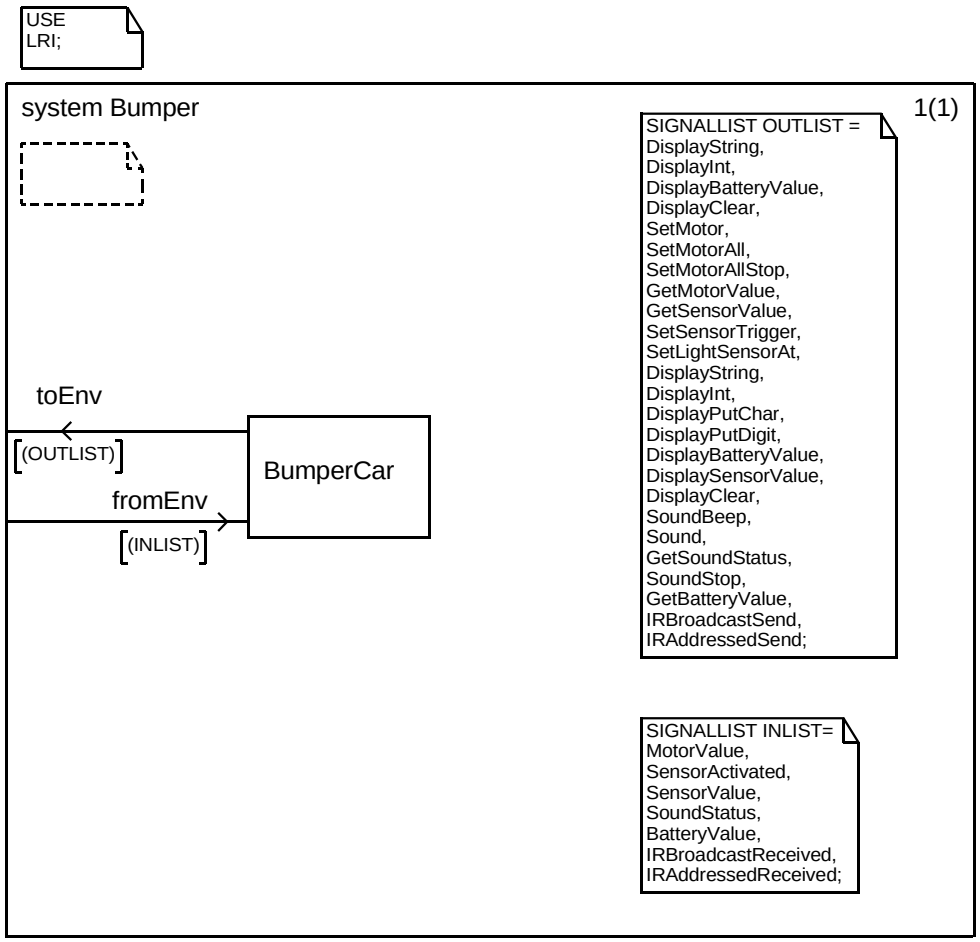


Figure 24 System Bumper

It is important to include the LRI in the system diagram. This is done in the top left corner box with “USE LRI”.

The signal lists in the system diagram contains all signals from the LRI even if just a few of them are actually used by the example. This is to make it easier and faster to edit the SDL system when more functionality is needed.

D.2.1 Block Diagram

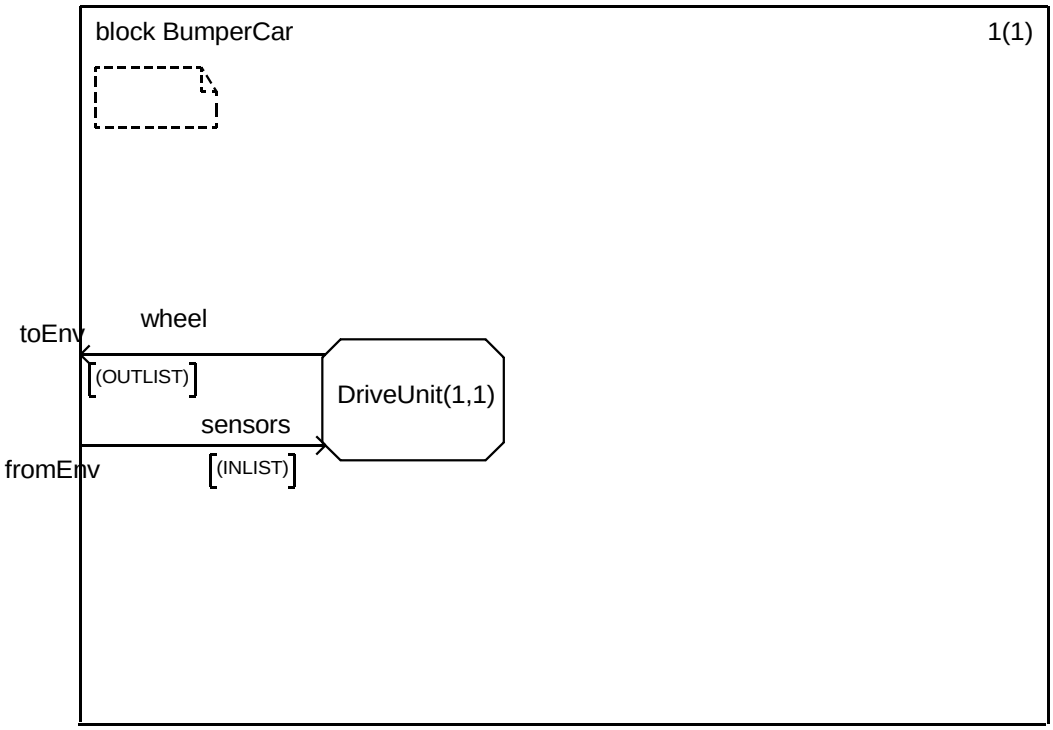


Figure 25 Block BumperCar

It is important to define the initial and maximum number of instances of the process when using CMicro. This is done with *DriveUnit(1,1)*.

D.2.2 Process Diagram

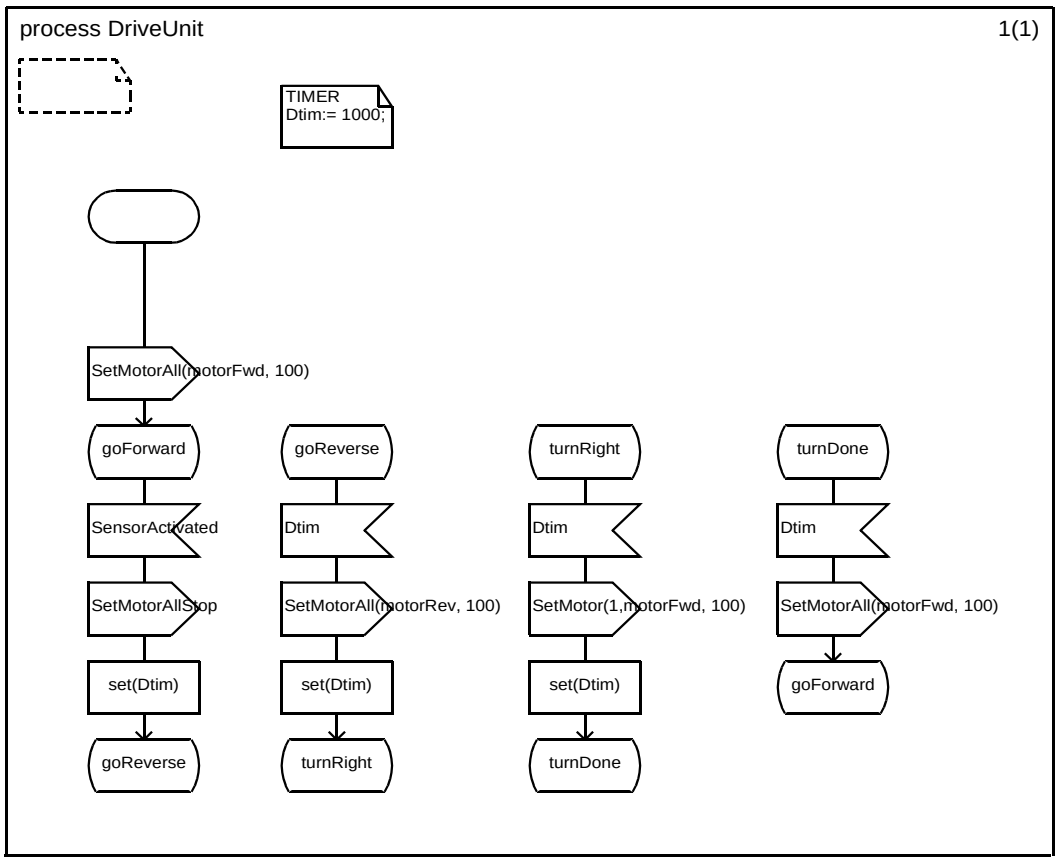


Figure 26 process DriveUnit

Process DriveUnit starts by setting all motors (A,B,C) to go forward with maximum speed (100). In state *goForward*, it waits for the robot to hit an obstacle. It will then receive the signal *SensorActivated* from the environment. All motors are then stopped and the timer *Dtim* is set. In the state *goReverse*, it will wait for the timeout (1000 milliseconds). When the timeout is received, all motors are set to reverse. After one second, motor A, located on left side will change direction for 100 ms and robot will turn right. All motors are set to move forward again and the first state goForward is entered and the behaviour is repeated indefinitely.

D.3 The environment file

This is the environment file used with the LRI package. It should be included when writing new SDL programs using LRI.

```
/*
*****************************************************************************
**      env.c corresponding to file component.ifc
**
** This file has been generated by the Targeting Expert - Version 4.0.1
** Date : Sat Sep 23 12:09:23 2000
```

```
**
** ATTENTION!
** The user must modify this file in the appropriate sections surrounded by
** “BEGIN User Code” and “END User Code”
** All modifications done within these sections will be saved, all the other
** ones will be lost when creating this file again.
** Do NOT delete, add or modify the lines “BEGIN User Code” and
** “END User Code”
*******************************************************************************
*/

#define XMK_ENV_TIME 969707363

#ifndef __ENV_C_
#define __ENV_C_


#include “ml_typ.h”

#include “component.ifc”

#define XENV_ENC( x ) x
#define XENV_DEC( x ) x

/* BEGIN User Code (global section)*/
/* It is possible to define some global variables here */
/* or to include other header files.                   */

#include “conio.h”
#include “dbutton.h”
#include “dmotor.h”
#include “dsensor.h”
#include “dsound.h”
#include “lnp.h”
#include “sys/lnp.h”
#include “lnp-logical.h”


int memcnt = 0;

/* When reading sensor values need to know what sensortype is used:
0 for normal, 1 for lightsensor */
int defSens1=0, defSens2=0, defSens3=0;
int const lightsensor_type = 1;

/* Need to know what trigger values to use for sensors */
int sensor1trigger, sensor2trigger, sensor3trigger;

void my_integrity_handler(const unsigned char *data, unsigned char length) {
XMK_SEND_TMP_VARS

yPDef_IRBroadcastReceived var;

//memcpy(var.Param1, data, length); /* Overkill? */
var.Param1 = data;
    XMK_SEND_ENV( ENV,
                  IRBroadcastReceived,
                  xDefaultPrioSignal,
                  sizeof( yPDef_IRBroadcastReceived ),
                  &var,
                  GLOBALPID(xRunPID,0));
}

void  my_addressing_handler(const  unsigned  char  *data,  unsigned  char  length,
unsigned char src_address) {
XMK_SEND_TMP_VARS
```

```
yPDef_IRAddressedReceived var;

//memcpy(var.Param1, data, length); /* Overkill? */
var.Param1 = data;
var.Param2 = src_address;
    XMK_SEND_ENV( ENV,
                  IRAddressedReceived,
                  xDefaultPrioSignal,
                  sizeof( yPDef_IRAddressedReceived ),
                  &var,
                  GLOBALPID(xRunPID,0));
}



/* The following defines should be removed after the   */
/* xInEnv() has been adapted to the real hardware      */
#define i_have_to_send_signal_MotorValue  0
#define Who_should_receive_signal_MotorValue 0
#define i_have_to_send_signal_SensorActivated  0
#define Who_should_receive_signal_SensorActivated 0
#define i_have_to_send_signal_SensorValue  0
#define Who_should_receive_signal_SensorValue 0
#define i_have_to_send_signal_SoundStatus  0
#define Who_should_receive_signal_SoundStatus 0
#define i_have_to_send_signal_BatteryValue  0
#define Who_should_receive_signal_BatteryValue 0
#define i_have_to_send_signal_IRBroadcastReceived  0
#define Who_should_receive_signal_IRBroadcastReceived 0
#define i_have_to_send_signal_IRAddressedReceived  0
#define Who_should_receive_signal_IRAddressedReceived 0
/*   END User Code (global section)*/

#ifdef XMK_USE_xInitEnv
/*
+--------------------------------------------------------------------------------+
|  Functionname : xInitEnv                                                       |
+--------------------------------------------------------------------------------+
|                                                                                |
|  Description :                                                                 |
|  This function is called by the Cmicro Kernel during initialization of the     |
|  SDL-System.                                                                   |
|                                                                                |
|  Parameter    : -                                                              |
|  Return       : -                                                              |
|                                                                                |
+--------------------------------------------------------------------------------+
*/
#ifndef XNOPROTO
  void xInitEnv (void)
#else
  void xInitEnv ()
#endif
{

  /* BEGIN User Code (variable section)*/
  /* It is possible to define some variables here */
  /*   END User Code (variable section)*/

  #ifdef XMK_ADD_PRINTF_USER
    XMK_FUNCTION(“xInitEnv”);
  #endif

  /* BEGIN User Code (init section) */
  /* Do the actions here to initialize your environment */

lnp_logical_range(1); // decides range of IR-communication
```

```
lnp_integrity_set_handler(my_integrity_handler);
lnp_addressing_set_handler(0, my_addressing_handler);

  /*   END User Code (init section) */

  #ifdef XMK_ADD_PRINTF_USER
    XMK_TRACE_EXIT("xInitEnv");
  #endif
} /* END OF FUNCTION */
#endif /* ... XMK_USE_xInitEnv */

#ifdef XMK_USE_xInEnv
/*
+--------------------------------------------------------------------------------+
|  Functionname : xInEnv                                                         |
+--------------------------------------------------------------------------------+
|                                                                                |
|  Description :                                                                 |
|  This function is called by the Cmicro Kernel continuously to retrieve         |
|  signals polled from the Environment.                                          |
|                                                                                |
|  The use of xInEnv() is not absolutely necessary in the case, where            |
|  the Cmicro Kernel is scaled to preemption, and all external Events are        |
|  put into the SDL-System via an Interrupt Service Routine, which is to be      |
|  written by the user.                                                          |
|                                                                                |
|  Parameter    : -                                                              |
|  Return       : -                                                              |
+--------------------------------------------------------------------------------+
*/
#ifndef XNOPROTO
   void xInEnv (void)
#else
   void xInEnv ()
#endif
{

   XMK_SEND_TMP_VARS

   /* BEGIN User Code (variable section)*/
   /* It is possible to define some variables here      */
   /* or to insert a functionality which must be polled */
   /*   END User Code (variable section)*/

  #ifdef XMK_ADD_PRINTF_USER
    XMK_FUNCTION("xInEnv");
  #endif /* ... XMK_ADD_PRINTF_USER */

  /* BEGIN User Code */
  if (i_have_to_send_signal_MotorValue)
  /*   END User Code */
  {
    yPDef_MotorValue var;

  /* BEGIN User Code */
    /* It is possible to insert any user code here. */
  /*   END User Code */
    XMK_SEND_ENV( ENV,
                  MotorValue,
                  xDefaultPrioSignal,
                  sizeof( yPDef_MotorValue ),
                  &var,
  /* BEGIN User Code */
                  GLOBALPID(Who_should_receive_signal_MotorValue,0));
    return;
  /*   END User Code */
  }
```

```
/* BEGIN User Code */
if (SENSOR_1 < 0xf000)
/*   END User Code */
{
  yPDef_SensorActivated var;

/* BEGIN User Code */
  /* It is possible to insert any user code here. */
/*   END User Code */
  XMK_SEND_ENV( ENV,
                SensorActivated,
                xDefaultPrioSignal,
                sizeof( yPDef_SensorActivated ),
                &var,
/* BEGIN User Code */
                GLOBALPID(XPTID_DriveUnit,0));
  return;
/*   END User Code */
}

/* BEGIN User Code */
if (i_have_to_send_signal_SensorValue)
/*   END User Code */
{
  yPDef_SensorValue var;

/* BEGIN User Code */
  /* It is possible to insert any user code here. */
/*   END User Code */
  XMK_SEND_ENV( ENV,
                SensorValue,
                xDefaultPrioSignal,
                sizeof( yPDef_SensorValue ),
                &var,
/* BEGIN User Code */
                GLOBALPID(Who_should_receive_signal_SensorValue,0));
  return;
/*   END User Code */
}

/* BEGIN User Code */
if (i_have_to_send_signal_SoundStatus)
/*   END User Code */
{
  yPDef_SoundStatus var;

/* BEGIN User Code */
  /* It is possible to insert any user code here. */
/*   END User Code */
  XMK_SEND_ENV( ENV,
                SoundStatus,
                xDefaultPrioSignal,
                sizeof( yPDef_SoundStatus ),
                &var,
/* BEGIN User Code */
                GLOBALPID(Who_should_receive_signal_SoundStatus,0));
  return;
/*   END User Code */
}

/* BEGIN User Code */
if (i_have_to_send_signal_BatteryValue)
/*   END User Code */
{
  yPDef_BatteryValue var;
```

```
  /* BEGIN User Code */
    /* It is possible to insert any user code here. */
  /*   END User Code */
    XMK_SEND_ENV( ENV,
                  BatteryValue,
                  xDefaultPrioSignal,
                  sizeof( yPDef_BatteryValue ),
                  &var,
  /* BEGIN User Code */
                  GLOBALPID(Who_should_receive_signal_BatteryValue,0));
    return;
  /*   END User Code */
  }

  /* BEGIN User Code */
  if (i_have_to_send_signal_IRBroadcastReceived)
  /*   END User Code */
  {
    yPDef_IRBroadcastReceived var;

  /* BEGIN User Code */
    /* It is possible to insert any user code here. */
  /*   END User Code */
    XMK_SEND_ENV( ENV,
                  IRBroadcastReceived,
                  xDefaultPrioSignal,
                  sizeof( yPDef_IRBroadcastReceived ),
                  &var,
  /* BEGIN User Code */
                  GLOBALPID(Who_should_receive_signal_IRBroadcastReceived,0));
    return;
  /*   END User Code */
  }

  /* BEGIN User Code */
  if (i_have_to_send_signal_IRAddressedReceived)
  /*   END User Code */
  {
    yPDef_IRAddressedReceived var;

  /* BEGIN User Code */
    /* It is possible to insert any user code here. */
  /*   END User Code */
    XMK_SEND_ENV( ENV,
                  IRAddressedReceived,
                  xDefaultPrioSignal,
                  sizeof( yPDef_IRAddressedReceived ),
                  &var,
  /* BEGIN User Code */
                  GLOBALPID(Who_should_receive_signal_IRAddressedReceived,0));
    return;
  /*   END User Code */
  }

  #ifdef XMK_ADD_PRINTF_USER
    XMK_TRACE_EXIT(“xInEnv”);
  #endif

  return ;

} /* END OF FUNCTION */
#endif /* ... XMK_USE_xInEnv */

#ifdef XMK_USE_xOutEnv
/*
+--------------------------------------------------------------------------------+
|  Functionname : xOutEnv                                                        |
```

```
+-----------------------------------------------------------------------------+
|                                                                             |
|  Description :                                                              |
|  This function is called by the Cmicro Kernel, if an SDL signal is to be    |
|  sent to the environment.                                                   |
|  The user must take care for that for each possible signal there is an      |
|  appropriate handling. If a signal was handled, the function should return  |
|  with XMK_TRUE, otherwise XMK_FALSE                                         |
|                                                                             |
|  Note:                                                                      |
|  The user has several possibilities to send signals to the environment by   |
|  using #EXTSIG,#ALT or #CODE, please view the reference manual.             |
|                                                                             |
|  Parameter    :  xmk_TmpSignalID        - ID of the signal (see .ifc file)  |
|                  xmk_TmpPrio            - Priority of signal                |
|                  xmk_TmpDataLength      - Amount of bytes in signal's       |
|                                           parameters                        |
|                 *xmk_TmpDataPtr         - Pointer to signal's parameters    |
|                  xmk_TmpReceiverPID     - PID of the receiver.              |
|  Return       : Function must return with XMK_TRUE, if Signal was sent      |
|                 to the environment, otherwise with XMK_FALSE.               |
|                                                                             |
+-----------------------------------------------------------------------------+
*/
#ifndef XNOPROTO
  xmk_OPT_INT   xOutEnv (xmk_T_SIGNAL xmk_TmpSignalID
                         #ifdef XMK_USE_SIGNAL_PRIORITIES
                          , xmk_T_PRIO xmk_TmpPrio
                         #endif
                         #ifdef XMK_USED_SIGNAL_WITH_PARAMS
                          , xmk_T_MESS_LENGTH xmk_TmpDataLength,
                           void xmk_RAM_ptr  xmk_TmpDataPtr
                         #endif
                         #ifdef XMK_USE_RECEIVER_PID_IN_SIGNAL
                          , xPID xmk_TmpReceiverPID
                         #endif
                         )
#else
  xmk_OPT_INT   xOutEnv  (xmk_TmpSignalID
                         #ifdef XMK_USE_SIGNAL_PRIORITIES
                         ,  xmk_TmpPrio
                         #endif
                         #ifdef XMK_USED_SIGNAL_WITH_PARAMS
                         , xmk_TmpDataLength,
                         xmk_TmpDataPtr
                         #endif
                         #ifdef XMK_USE_RECEIVER_PID_IN_SIGNAL
                         , xmk_TmpReceiverPID
                         #endif
                         )
                         xmk_T_SIGNAL       xmk_TmpSignalID;
                         #ifdef XMK_USE_SIGNAL_PRIORITIES
                         xmk_T_PRIO         xmk_TmpPrio;
                         #endif
                         #ifdef XMK_USED_SIGNAL_WITH_PARAMS
                         xmk_T_MESS_LENGTH  xmk_TmpDataLength;
                         void xmk_RAM_ptr  xmk_TmpDataPtr;
                         #endif
                         #ifdef XMK_USE_RECEIVER_PID_IN_SIGNAL
                         xPID               xmk_TmpReceiverPID;
                         #endif
#endif
{
  xmk_OPT_INT xmk_result = XMK_FALSE;

  /* BEGIN User Code (variable section)*/
```

```
  /*   END User Code (variable section)*/

  #ifdef XMK_ADD_MICRO_ENVIRONMENT
    xmk_T_CMD_EXT_OUTPUT_TO_ENV_IND cmd_data;
  #endif

  #ifdef XMK_ADD_PRINTF_USER
    XMK_FUNCTION("xOutEnv");
  #endif

  #ifdef XMK_ADD_PRINTF_ADDITIONAL
    PRINTF (("signal_id = %d\n", xmk_TmpSignalID));
  #ifdef XMK_USE_SIGNAL_PRIORITIES
    PRINTF (("xmk_TmpPrio      = %d\n", xmk_TmpPrio));
  #endif
  #ifdef XMK_USED_SIGNAL_WITH_PARAMS
    PRINTF (("xmk_TmpDataLength  = %d\n", xmk_TmpDataLength));
  #endif
  #endif /* ... XMK_ADD_PRINTF_USER */

  #ifdef XMK_ADD_MICRO_ENVIRONMENT
    /* BEGIN User Code (external env section) */
    /*
    ** This part is used only, if an external environment is
    ** used on host side.
    */
    switch (xmk_TmpSignalID)
    {
     case DisplayString :
     case DisplayInt :
     case DisplayBatteryValue :
     case DisplayClear :
     case SetMotor :
     case SetMotorAll :
     case SetMotorAllStop :
     case GetMotorValue :
     case GetSensorValue :
     case SetSensorTrigger :
     case SetLightSensorAt :
     case DisplayPutChar :
     case DisplayPutDigit :
     case DisplaySensorValue :
     case SoundBeep :
     case Sound :
     case GetSoundStatus :
     case SoundStop :
     case GetBatteryValue :
     case IRBroadcastSend :
     case IRAddressedSend :
          cmd_data.signal = xmk_TmpSignalID;

          #ifdef XMK_USED_SIGNAL_WITH_PARAMS
            /*
            ** It is the user's task to ensure that 'xmk_TmpDataLength'
            ** is smaller than or equal to XMK_MAX_TSDL_PARAM.
              ** Please see the section 'Cmicro Tester/Trace Buffers/Length of one
transmitter buffer'
            ** in the SDT Targeting Expert.
            ** XMK_MAX_TSDL_PARAM = XMK_MAX_SEND_ONE_ENTRY -12
            */
            cmd_data.mess_length = xmk_TmpDataLength;
            memcpy(cmd_data.parametercopy, xmk_TmpDataPtr, xmk_TmpDataLength);
            xmk_Cod_Encode (XMK_MICRO_ENVIRONMENT,
                            CMD_EXT_OUTPUT_TO_ENV_IND,
                            (char*)&cmd_data,
                                          sizeof(xmk_T_CMD_EXT_OUTPUT_TO_ENV_IND)-
sizeof(cmd_data.parametercopy)+cmd_data.mess_length);
```

```
          #else
            cmd_data.mess_length = 0;
            xmk_Cod_Encode (XMK_MICRO_ENVIRONMENT,
                            CMD_EXT_OUTPUT_TO_ENV_IND,
                            NULL,
                            0);
          #endif

          xmk_result = XMK_TRUE;
          break;

       default :
         xmk_result = XMK_FALSE;/* to tell the caller that */
                                /* signal is NOT consumed  */
                                /* and to be handled by    */
                                /* the Cmicro Kernel ...    */
         break ;
     }
     /*   END User Code (external env section) */
  #endif /* ... XMK_ADD_MICRO_ENVIRONMENT */

  switch (xmk_TmpSignalID)
  {
     case DisplayString :
          {
            /* BEGIN User Code */
            /* Use (yPDP_DisplayString)xmk_TmpDataPtr to access the signal's param-
eters */
            /* ATTENTION: the data needs to be copied. Otherwise it */
            /*            will be lost when leaving xOutEnv */
            /* Do your environment actions here. */

cputs( (((yPDP_DisplayString)xmk_TmpDataPtr) -> Param1) );

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case DisplayInt :
          {
            /* BEGIN User Code */
           /* Use (yPDP_DisplayInt)xmk_TmpDataPtr to access the signal's parameters
*/
            /* ATTENTION: the data needs to be copied. Otherwise it */
            /*            will be lost when leaving xOutEnv */
            /* Do your environment actions here. */

int intval;
intval = ((yPDef_DisplayInt*) xmk_TmpDataPtr)->Param1 ;
cputw( intval );

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case DisplayBatteryValue :
          {
            /* BEGIN User Code */
            /* Do your environment actions here. */

cputw(BATTERY);

xmk_result = XMK_TRUE; /* to tell the caller that */
```

```
                                    /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case DisplayClear :
          {
            /* BEGIN User Code */
            /* Do your environment actions here. */

cls();

xmk_result = XMK_TRUE; /* to tell the caller that */
                                    /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case SetMotor :
          {
            /* BEGIN User Code */
             /* Use (yPDP_SetMotor)xmk_TmpDataPtr to access the signal's parameters
*/
            /* ATTENTION: the data needs to be copied. Otherwise it */
            /*            will be lost when leaving xOutEnv */
            /* Do your environment actions here. */

int dir, speed;
dir = ((yPDef_SetMotor*) xmk_TmpDataPtr)->Param2 ;
speed = ((yPDef_SetMotor*) xmk_TmpDataPtr)->Param3 ;

switch ( ((yPDef_SetMotor*) xmk_TmpDataPtr)->Param1 )
{
  case 1 :motor_a_dir(dir);
motor_a_speed(speed);
break;
  case 2 :motor_b_dir(dir);
motor_b_speed(speed);
break;
  case 3 :motor_c_dir(dir);
motor_c_speed(speed);
break;
           default: break;
}

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                    /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case SetMotorAll :
          {
            /* BEGIN User Code */
             /* Use (yPDP_SetMotorAll)xmk_TmpDataPtr to access the signal's parame-
ters */
            /* ATTENTION: the data needs to be copied. Otherwise it */
            /*            will be lost when leaving xOutEnv */
            /* Do your environment actions here. */

int dir, speed;
dir = ((yPDef_SetMotor*) xmk_TmpDataPtr)->Param1 ;
speed = ((yPDef_SetMotor*) xmk_TmpDataPtr)->Param2 ;

motor_a_dir(dir);
motor_a_speed(speed);
motor_b_dir(dir);
```

```
motor_b_speed(speed);
motor_c_dir(dir);
motor_c_speed(speed);


            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case SetMotorAllStop :
          {
            /* BEGIN User Code */
            /* Do your environment actions here. */

motor_a_dir(0);
motor_a_speed(0);
motor_b_dir(0);
motor_b_speed(0);
motor_c_dir(0);
motor_c_speed(0);

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case GetMotorValue :
          {
            /* BEGIN User Code */
            /* Use (yPDP_GetMotorValue)xmk_TmpDataPtr to access the signal's param-
eters */
            /* ATTENTION: the data needs to be copied. Otherwise it */
            /*            will be lost when leaving xOutEnv */
            /* Do your environment actions here. */

XMK_SEND_TMP_VARS

yPDef_MotorValue var;

var.Param1 = 1;
var.Param2 = dm_a.dir;
var.Param3 = dm_a.access.c.delta ;


    XMK_SEND_ENV( ENV,
                  MotorValue,
                  xDefaultPrioSignal,
                  sizeof( yPDef_MotorValue ),
                  &var,
                  GLOBALPID(xRunPID,0));



            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case GetSensorValue :
          {
            /* BEGIN User Code */
           /* Use (yPDP_GetSensorValue)xmk_TmpDataPtr to access the signal's param-
eters */
```

```
            /* ATTENTION: the data needs to be copied. Otherwise it */
            /*            will be lost when leaving xOutEnv */
            /* Do your environment actions here. */

XMK_SEND_TMP_VARS

yPDef_SensorValue var;

switch ( ((yPDP_GetSensorValue)xmk_TmpDataPtr)->Param1 )
{
  case 1 :if (defSens1==lightsensor_type)
var.Param1 = LIGHT_1 ;
           else
               var.Param1 = SENSOR_1 ;
break;
  case 2 : if (defSens2==lightsensor_type)
var.Param1 = LIGHT_2 ;
           else
                  var.Param1 = SENSOR_2 ;
break;
  case 3 : if (defSens3==lightsensor_type)
var.Param1 = LIGHT_3 ;
           else
           var.Param1 = SENSOR_3 ;
break;
         default: break;
}

XMK_SEND_ENV( ENV,
                  SensorValue,
                  xDefaultPrioSignal,
                  sizeof( yPDef_SensorValue ),
                  &var,
                  GLOBALPID(xRunPID,0));

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case SetSensorTrigger :
          {
            /* BEGIN User Code */
              /* Use (yPDP_SetSensorTrigger)xmk_TmpDataPtr to access the signal's
parameters */
            /* ATTENTION: the data needs to be copied. Otherwise it */
            /*            will be lost when leaving xOutEnv */
            /* Do your environment actions here. */

            int trigvalue;
            trigvalue= ((yPDP_SetSensorTrigger)xmk_TmpDataPtr)->Param2;

            switch ( ((yPDP_SetSensorTrigger)xmk_TmpDataPtr)->Param1)
            {
             case 1: sensor1trigger = trigvalue ;
             break;
             case 2: sensor2trigger = trigvalue ;
             break;
             case 3: sensor3trigger = trigvalue ;
             break;
             default: break;
            }

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
```

```
          }
          break ;

     case SetLightSensorAt :
          {
            /* BEGIN User Code */
               /* Use (yPDP_SetLightSensorAt)xmk_TmpDataPtr to access the signal's
parameters */
            /* ATTENTION: the data needs to be copied. Otherwise it */
            /*            will be lost when leaving xOutEnv */
            /* Do your environment actions here. */


switch ( ((yPDP_SetLightSensorAt)xmk_TmpDataPtr)->Param1)
{
case 1:defSens1 = lightsensor_type;
         ds_active(&SENSOR_1);
break;
case 2:defSens2 = lightsensor_type;
         ds_active(&SENSOR_2);
break;
case 3: defSens3 = lightsensor_type;
         ds_active(&SENSOR_3);
break;
         default: break;
}

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case DisplayPutChar :
          {
            /* BEGIN User Code */
           /* Use (yPDP_DisplayPutChar)xmk_TmpDataPtr to access the signal's param-
eters */
            /* ATTENTION: the data needs to be copied. Otherwise it */
            /*            will be lost when leaving xOutEnv */
            /* Do your environment actions here. */

     /* put one character on a position on the LCD display */
cputc( ((yPDP_DisplayPutChar)xmk_TmpDataPtr)->Param1 ,
  ((yPDP_DisplayPutChar)xmk_TmpDataPtr)->Param2 );

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case DisplayPutDigit :
          {
            /* BEGIN User Code */
               /* Use (yPDP_DisplayPutDigit)xmk_TmpDataPtr to access the signal's
parameters */
            /* ATTENTION: the data needs to be copied. Otherwise it */
            /*            will be lost when leaving xOutEnv */
            /* Do your environment actions here. */

/* put a digit on one position on the LCD display */
cputc_hex( (((yPDP_DisplayPutDigit)xmk_TmpDataPtr)->Param1) ,
      (((yPDP_DisplayPutDigit)xmk_TmpDataPtr)->Param2) );

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
```

```
            /*   END User Code */
          }
          break ;

     case DisplaySensorValue :
          {
            /* BEGIN User Code */
             /* Use (yPDP_DisplaySensorValue)xmk_TmpDataPtr to access the signal's
parameters */
            /* ATTENTION: the data needs to be copied. Otherwise it */
            /*            will be lost when leaving xOutEnv */
            /* Do your environment actions here. */

char sensVal;

switch ( ((yPDP_DisplaySensorValue)xmk_TmpDataPtr)->Param1 )
{
  case 1 : if (defSens1==lightsensor_type)
           sensVal = LIGHT_1;
                    else
                    sensVal = SENSOR_1 ;
break;
  case 2 :if (defSens2==lightsensor_type)
           sensVal = LIGHT_2;
                    else
                    sensVal = SENSOR_2 ;
break;
  case 3 :if (defSens3==lightsensor_type)
           sensVal = LIGHT_3;
                    else
                    sensVal = SENSOR_3 ;
break;
  default : sensVal = 0;
break;
}

cputw( sensVal );

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case SoundBeep :
          {
            /* BEGIN User Code */
            /* Do your environment actions here. */

dsound_system(DSOUND_BEEP);

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case Sound :
          {
            /* BEGIN User Code */
            /* Use (yPDP_Sound)xmk_TmpDataPtr to access the signal's parameters */
            /* ATTENTION: the data needs to be copied. Otherwise it */
            /*            will be lost when leaving xOutEnv */
            /* Do your environment actions here. */

note_t my_note[] = {
{ (((yPDef_Sound*) xmk_TmpDataPtr)->Param1)} ,
```

```
{ (((yPDef_Sound*) xmk_TmpDataPtr)->Param2)}
};

dsound_play( my_note );

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case GetSoundStatus :
          {
            /* BEGIN User Code */
            /* Do your environment actions here. */

XMK_SEND_TMP_VARS

yPDef_SoundStatus var;

if ( dsound_playing() == 0 )
var.Param1 = 0;
else
var.Param1 = 1;

    XMK_SEND_ENV( ENV,
                  SoundStatus,
                  xDefaultPrioSignal,
                  sizeof( yPDef_SoundStatus ),
                  &var,
                  GLOBALPID(xRunPID,0));



            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case SoundStop :
          {
            /* BEGIN User Code */
            /* Do your environment actions here. */

dsound_stop();

            xmk_result = XMK_TRUE; /* to tell the caller that */
                                   /* signal is consumed      */
            /*   END User Code */
          }
          break ;

     case GetBatteryValue :
          {
            /* BEGIN User Code */
            /* Do your environment actions here. */


XMK_SEND_TMP_VARS// to use XMK_SEND_ENV in OutEnv and xRunPID

yPDef_BatteryValue var;

var.Param1 = BATTERY ;// legOS gobal value

XMK_SEND_ENV( ENV,
                  BatteryValue,
```

```
                    xDefaultPrioSignal,
                    sizeof( yPDef_BatteryValue ),
                    &var,
                    GLOBALPID(xRunPID , 0));

              xmk_result = XMK_TRUE;/* to tell the caller that */
/* signal is consumed*/
              /*   END User Code */
            }
            break ;

       case IRBroadcastSend :
            {
              /* BEGIN User Code */
                 /* Use (yPDP_IRBroadcastSend)xmk_TmpDataPtr to access the signal's
parameters */
              /* ATTENTION: the data needs to be copied. Otherwise it */
              /*            will be lost when leaving xOutEnv */
              /* Do your environment actions here. */

lnp_integrity_write(((yPDP_IRBroadcastSend) xmk_TmpDataPtr)->Param1,
 (strlen(((yPDP_IRBroadcastSend) xmk_TmpDataPtr)->Param1))+1);
free(((yPDP_IRBroadcastSend) xmk_TmpDataPtr)->Param1);

              xmk_result = XMK_TRUE; /* to tell the caller that */
                                     /* signal is consumed      */
              /*   END User Code */
            }
            break ;

       case IRAddressedSend :
            {
              /* BEGIN User Code */
                 /* Use (yPDP_IRAddressedSend)xmk_TmpDataPtr to access the signal's
parameters */
              /* ATTENTION: the data needs to be copied. Otherwise it */
              /*            will be lost when leaving xOutEnv */
              /* Do your environment actions here. */

lnp_addressing_write(((yPDP_IRAddressedSend) xmk_TmpDataPtr)->Param1,
 (strlen(((yPDP_IRAddressedSend) xmk_TmpDataPtr)->Param1))+1,
 ((yPDP_IRAddressedSend) xmk_TmpDataPtr)->Param2,
 0);
free(((yPDP_IRAddressedSend) xmk_TmpDataPtr)->Param1);

              xmk_result = XMK_TRUE; /* to tell the caller that */
                                     /* signal is consumed      */
              /*   END User Code */
            }
            break ;

       default :
            xmk_result = XMK_FALSE;/* to tell the caller that */
                                   /* signal is NOT consumed  */
                                   /* and to be handled by    */
                                   /* the Cmicro Kernel ...   */
            break ;
    }

    #ifdef XMK_USE_SIGNAL_PRIORITIES
      XMK_SUPPRESS(xmk_TmpPrio);
    #endif
    #ifdef XMK_USED_SIGNAL_WITH_PARAMS
      XMK_SUPPRESS(xmk_TmpDataLength);
      XMK_SUPPRESS(xmk_TmpDataPtr);
    #endif
    #ifdef XMK_USE_RECEIVER_PID_IN_SIGNAL
```

```
    XMK_SUPPRESS(xmk_TmpReceiverPID);
  #endif

  #ifdef XMK_ADD_PRINTF_USER
    XMK_TRACE_EXIT("xOutEnv");
  #endif

  return (xmk_result);
} /* END OF FUNCTION */
#endif /* ... XMK_USE_xOutEnv */

#ifdef XMK_USE_xCloseEnv
/*
+--------------------------------------------------------------------------------+
|  Functionname : xCloseEnv                                                      |
+--------------------------------------------------------------------------------+
|                                                                                |
|  Description :                                                                 |
|  This function is called by the Cmicro Kernel after termination.               |
|  It is not that useful in a Cmicro controler application, but may be so in     |
|  a simulation with the Cmicro Package.                                         |
|                                                                                |
|  Parameter    : -                                                              |
|  Return       : -                                                              |
|                                                                                |
+--------------------------------------------------------------------------------+
*/
#ifndef XNOPROTO
  void xCloseEnv (void)
#else
  void xCloseEnv ()
#endif
{
  /* BEGIN User Code (variable section)*/
  /* It is possible to define some variables here */
  /*   END User Code (variable section)*/

  #ifdef XMK_ADD_PRINTF_USER
    XMK_FUNCTION("xCloseEnv");
  #endif

  /* BEGIN User Code (close section) */
  /* Do the actions here to close your environment */
  /*   END User Code (close section) */

  #ifdef XMK_ADD_PRINTF_USER
    XMK_TRACE_EXIT("xCloseEnv");
  #endif
} /* END OF FUNCTION */
#endif /* ... XMK_USE_xCloseEnv */

#endif /* ... __ENV_C_ */
```

Appendix E
Introduction to SDL

This appendix presents a short introduction to SDL concepts. This introduction was strongly influenced by Telelogic SDL books and tutorials ([1]).

SDL (Specification and Description Language) is a standard language to specify and describe communicating systems. It was developed by CCITT (Comité Consultatif Internationel Télégraphique et Téléphonique), now ITU-T, and is the ITU-T Recommendation Z.100.

SDL is specifically concerned with the specification of behaviour, structure and data. It was developed for use in telecommunications systems, but it may be used in all kinds of real-time communicating systems.

The part used in this project is the graphical representation of SDL. This is the part of SDL that has made it user friendly and, because of that, more popular. Most other specification languages have only a textual representation.

E.1 Hierarchical Levels

SDL has the hierarchical levels System, Block and Process.

A block can contain other blocks, which can in turn contain other blocks. A hierarchical structure is created with blocks in blocks.

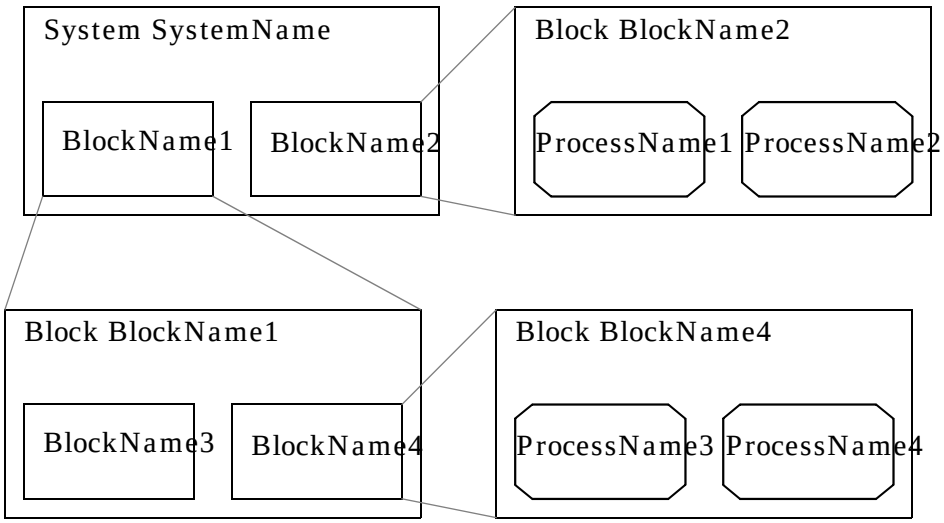


Figure 27 The Hierarchical Structure of SDL

A block can also contain processes. Processes describe behaviour. A block must contain at least one block or one process. Blocks and processes must not be mixed in one block.

E.2 Behaviour

Dynamic behaviour is described by processes. Processes execute in parallel and are independent of each other. This means that the status in one process in not known by the other process in the system.

Processes are defined in the static specification. During run-time when a system is executing, instances from that definition are created. More than one instance of a process can exist at the same time during run-time.

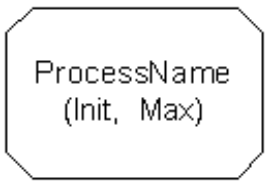


Figure 28 Process Declaration

Process instances can be created at system start up or created and terminated dynamically at run-time. Init = the initial number of instances at start up. Max = the maximum number of instances that can exist at the same time during execution.

The model used to describe behaviour in processes is the finite state machine.

E.2.1 Finite State Machine

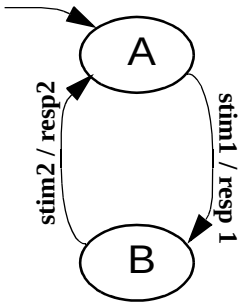


Figure 29 A Finite State Machine

A finite state machine (FSM) consists of states, in this case (*Figure 29*) state A and B. Going from one state to another is called a transition. A transition between two states is made only after a stimulus has been received. The state machine is waiting in state A and when stim1 is received the transition to B is made. Now the state machine is waiting in state B.

When a finite state machine is executed, its initial state must be known. Therefor one of the states is defined as the start state. A process is a finite state machine extended with data and communication.

E.2.2 Process Behaviour

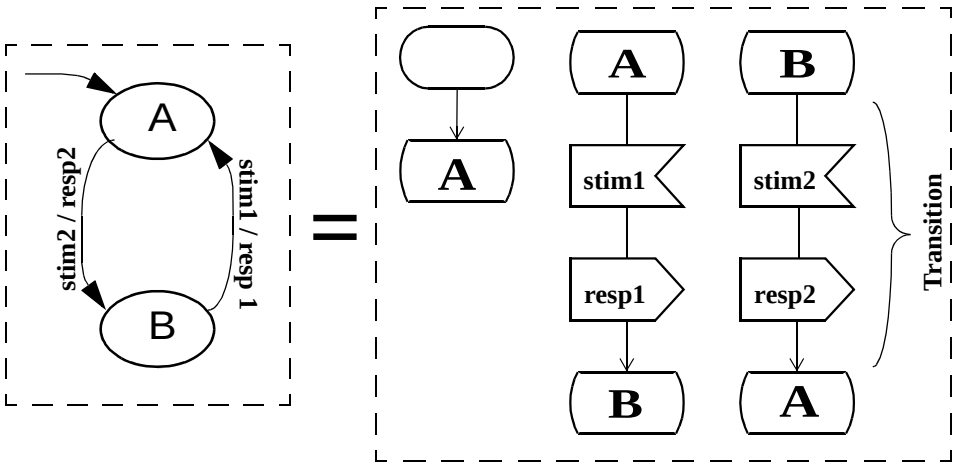


Figure 30 A Finite State Machine and its SDL Equivalent

This process (*Figure 30*) consists of two states A and B. State A is defined as the start state. The stimuli in SDL are called signals. The only way to leave a state and enter another state is to receive a signal, also called input of a signal. When the signal is received, the transition is initiated.

During the transition actions can be executed. In this example a signal is sent out. The next state defines the end of the transition and which state to enter next.

E.2.3 Process Symbols

Behaviour is described using a flow chart notation. The following symbols are the basic building blocks in the processes.

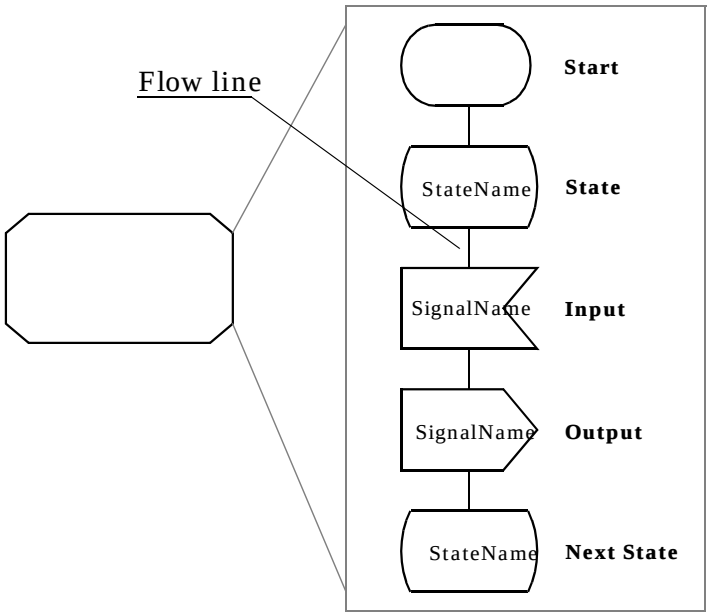


Figure 31 Process Symbols

E.3 Communication

Processes communicate with discrete signals. Communication in SDL is asynchronous, which means that the sending process continues executing without waiting for an acknowledgement from the receiving process. Signals are defined in a text symbol.

```
SIGNAL
stim1, stim2,
resp1, resp2,
SignalName;
```

Figure 32 Signal Declaration

E.3.1 Channels

Channels define the communication path through which blocks communicate with each other or with the environment. Communication with the environment takes place at frame level.

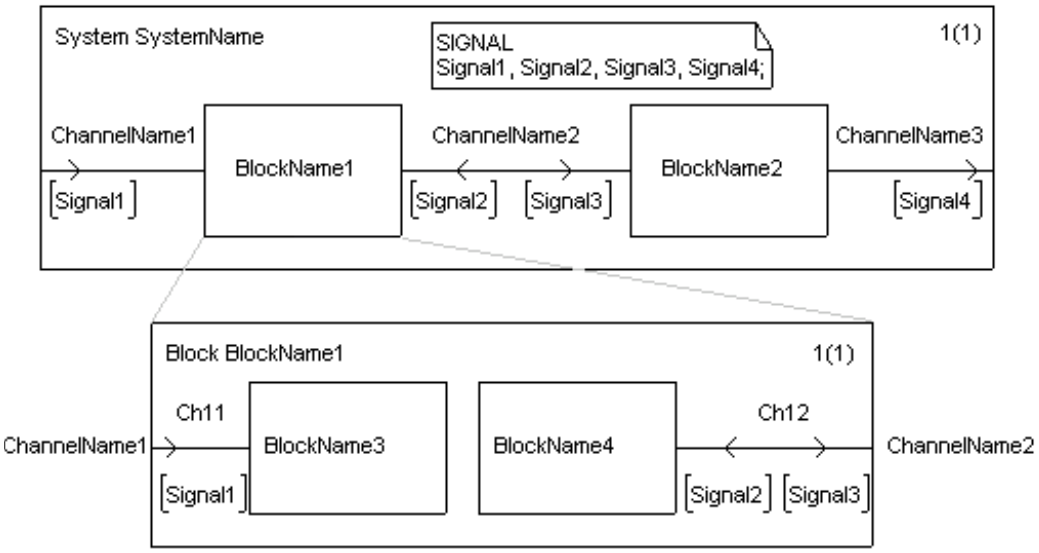


Figure 33 Channels

Adjacent to the channel arrow, signals which can travel on the channel in the arrow direction are stated within [].

E.3.2 Signal Routes

Signal routes define the communication path through which processes communicate with each other or with the block level above. Communication with the level above takes place at frame level and the name of the channel is stated outside the frame.

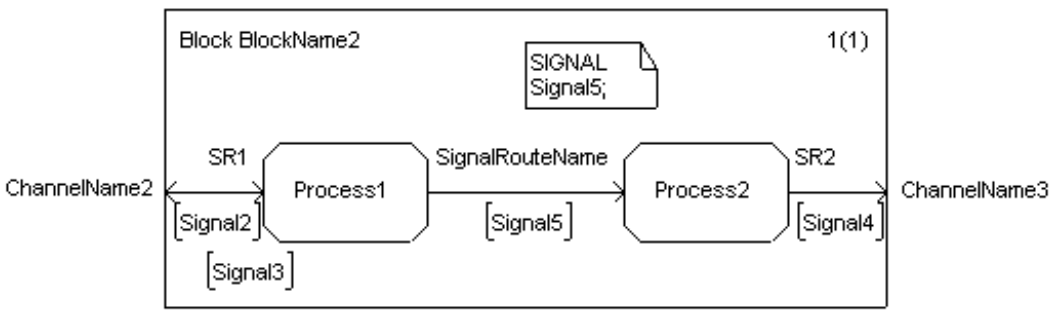


Figure 34 Signal Routes

Adjacent to the signal route arrow, the signals which can travel on the signal route in the arrow direction are stated within [].

Appendix F
Other Files

Contains other files written during the course of the project.

F.1 Memory Checker

Used to check for memory leaks. The function memcheck() returns a size_t type variable containing the amount of unallocated memory in kilobytes.

F.1.1 memcheck.h

```
/*! \file   include/template.h
    \brief
    \author Markus L. Noga <markus@noga.de>
 */

/*
 *  The contents of this file are subject to the Mozilla Public License
 *  Version 1.0 (the "License"); you may not use this file except in
 *  compliance with the License. You may obtain a copy of the License at
 *  http://www.mozilla.org/MPL/
 *
 *  Software distributed under the License is distributed on an "AS IS"
 *  basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the
 *  License for the specific language governing rights and limitations
 *  under the License.
 *
 *  The Original Code is legOS code, released October 17, 1999.
 *
 *  The Initial Developer of the Original Code is Markus L. Noga.
 *  Portions created by Markus L. Noga are Copyright (C) 1999
 *  Markus L. Noga. All Rights Reserved.
 *
 *  Contributor(s): Markus L. Noga <markus@noga.de>
 */

#ifndef __memcheck_h__
#define __memcheck_h__

////////////////////////////////////////////////////////////////////////////
//
// Definitions
//
////////////////////////////////////////////////////////////////////////////


////////////////////////////////////////////////////////////////////////////
//
// Variables
//
////////////////////////////////////////////////////////////////////////////
extern size_t *mm_first_free;
```

```
extern size_t mm_start;


//////////////////////////////////////////////////////////////////////////////
//
// Functions
//
//////////////////////////////////////////////////////////////////////////////
extern size_t memcheck();

#endif // __template_h__
```

F.1.2 memcheck.c

```
/**
 *
 *
 **/
#include <unistd.h>
#include <stdlib.h>
#include <sys/tm.h>
#include <sys/mm.h>
#include <sys/irq.h>

size_t memcheck() {
size_t size, total;
size_t *ptr;

total = 0;

ptr = mm_first_free;
while (ptr >= &mm_start) {
if (*(ptr++) == MM_FREE) {
size = *ptr << 1;
total += size;
}
ptr += (*ptr) +1;
}

total = (total / 1024) * 100 + (total % 1024) / 10;
return total;
}
```

F.2 legOS Linking Script

Use to bring the whole linking procedure of legOS into one command. The procedure consists of 3 steps. Link twice to different addresses, and then combine the two to the legOS moveable binary format (lx).

```
@echo off
:LOOP
if “%2”==”” goto DONE
set objects=%objects% %1
shift
goto loop
:DONE
REM Linking #1
C:\legOS\H8\Bin\h8300-hms-ld   -T   c:/legos/boot/legos.lds   -relax   -Lc:/legos/lib
%objects% -lc -lmint -lfloat -o %1.ds1 -Ttext 0xb000
REM Linking #2
C:\legOS\H8\Bin\h8300-hms-ld   -T   c:/legos/boot/legos.lds   -relax   -Lc:/legos/lib
%objects% -lc -lmint -lfloat -o %1.ds2 -Ttext 0xb210
```

```
REM Makelx
c:\legOS\util\makelx %1.ds1 %1.ds2 %1

REM Cleanup
set objects=
```

F.3 strcat function

```
/*! \file   strcat.c
    \brief  strcat function
    \author Markus L. Noga <markus@noga.de>
*/

/*
 *  The contents of this file are subject to the Mozilla Public License
 *  Version 1.0 (the “License”); you may not use this file except in
 *  compliance with the License. You may obtain a copy of the License at
 *  http://www.mozilla.org/MPL/
 *
 *  Software distributed under the License is distributed on an “AS IS”
 *  basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the
 *  License for the specific language governing rights and limitations
 *  under the License.
 *
 *  The Original Code is legOS code, released October 17, 1999.
 *
 *  The Initial Developer of the Original Code is Markus L. Noga.
 *  Portions created by Markus L. Noga are Copyright (C) 1999
 *  Markus L. Noga. All Rights Reserved.
 *
 *  Contributor(s): Markus L. Noga <markus@noga.de>
 *                  Torkel Niklasson <tniklasson@onebox.com>
 */

//! Concatenate null-terminated string src to dest
char* strcat(char *dest,const char *src) {
char *d2 = dest;

while(*d2 != ‘\0’) d2++; /* you are null terminating right! */

while(*src != ‘\0’) {    /* loop *src until null terminator */
*d2 = *src;
d2++;
src++;
}

*d2 = ‘\0’;             /* Null termination is your friend */

return dest;
}
```

Appendix G
CMicro File Changes

Contains the change logs generated by the DLSupC32 compare tool. These document the changes that have been made to the template files of CMicro.

G.1 Summary

G.1.1 mk_stim.c

The following lines were added to add the time functions from legOS to the CMicro kernel:

```
#elif defined(GCCLEGOS)
#include “time.h”
```

There is no need to initialize, deinitialize or set the system clock, so the contents of these functions are removed (made to comments).

In the function to retreive the current system time (xmk_NOW) the following was added:

```
#elif defined(GCCLEGOS)
  #ifdef XMK_ADD_PRINTF_TIMER
    XMK_FUNCTION("xmk_NOW");
  #endif

  XMK_BEGIN_CRITICAL_PATH;
  _SystemTime = sys_time;
  XMK_END_CRITICAL_PATH;
#else
```

G.1.2 mk_user.c

Not much of mk_user.c needed to be altered in our implementation. Parts were made into comments to make compilation possible, as shown in *Appendix G.3*.

G.1.3 mk_cpu.c

Nothing was added to this file. Certain parts were made into comments, as shown in *Appendix G.4.*

G.1.4 ml_mem.c

Only one thing was changed in ml_mem.c.

`#if defined(ARM_THUMB) || defined(_GCC_) || defined(MCC68K)`
was changed to

`#if defined(ARM_THUMB) || defined(_GCC_) || defined(MCC68K) || defined(GCCLEGOS)`

G.1.5 sctpred.c

The line

`#define abs(j) ((j)<0?-(j):(j))`
was inserted to define the abs function. This is technically a modification of legOS. It was only used in sctpred.c, so we chose to implement it there.

G.2 mk_stim.c Change Log

```
(UnRegistered) DLSupC32 File Compare Report -  9/10/00  5:00:48 PM

Panel 1 Report -- New Source Matches + Inserts
New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\mk_stim.c  5/15/00 3:32:58 PM
Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\template\mk_stim.c  5/5/00 11:41:14 AM
Ref# ID  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8 N-REF# CHNG-TYPE   O-REF# LEN
   1    ************* 87 Unchanged New File Lines Not Shown ****************************
   2     #elif defined(MICROSOFT_C)                                                      00088              00088
   3       #ifdef XMK_WINDOWS                                                            00089              00089
   4         #include <sys\timeb.h>                                                     00090              00090
   5       #else                                                                         00091              00091
   6         #include "Error_in_mk_stim.c_NoTimerImplementationAvailable-02"           00092              00092
   7       #endif                                                                        00093              00093
   8 I - #elif defined(GCCLEGOS)                                                       00094 INSERT(NEW)          2
   9 I -   #include "time.h"                                                           00095 INSERT
  10     #endif                                                                        00096              00094   79
  11                                                                                   00097              00095
  12     /*============================================================================*/ 00098              00096
  13     /*-IMPORT*/                                                                   00099              00097
  14                                                                                   00100              00098
  15                                                                                   00101              00099
  16    ************* 67 Unchanged New File Lines Not Shown ****************************
  17     #endif                                                                        00169              00167
  18                                                                                   00170              00168
  19     #ifdef TCC80C196                                                              00171              00169
  20       static unsigned int LastTime;                                               00172              00170
  21     #endif                                                                        00173              00171
  22                                                                                   00174              00172
  23 I - #if defined(MICROSOFT_C) || defined(BORLAND_C) || \                          00175 INSERT(NEW)          2
  24 I -     defined(IARC51) || defined(KEIL_C51) || defined(GNU80166) || \           00176 INSERT
  25         defined(TCC80166) || defined(MSP58C80) || defined(IARC7700) || \         00177              00175   5
  26         defined(HYPERSTONE) || defined(KEIL_C166) || defined(TCC80C196)          00178              00176
  27       static xmk_T_TIME SystemTime;                                               00179              00177
  28     #endif                                                                        00180              00178
  29                                                                                   00181              00179
  30
  31     /*============================================================================*/ 00182              00181   180
  32     /*-MGG*/                                                                      00183              00182
  33                                                                                   00184              00183
  34     /*+FHDR E*/                                                                   00185              00184
  35     /*                                                                            00186              00185
  36     +----------------------------------------------------------------------------+ 00187              00186
  37    ************* 168 Unchanged New File Lines Not Shown ****************************
  38            XMK_BEGIN_CRITICAL_PATH;                                               00356              00355
  39            StartTime = UnixTime; /* OK for loosing precision in cast from long to u_ 00357            00356
  40            XMK_END_CRITICAL_PATH;                                                 00358              00357
  41         }                                                                         00359              00358
  42       #else                                                                       00360              00359
  43                                                                                   00361              00360
  44 I -     // #include "Fill_up_function_xmk_InitSystime_for_your_compiler"         00362 INSERT(NEW)          1
  45                                                                                   00363              00362   63
  46       #endif                                                                      00364              00363
  47                                                                                   00365              00364
```

```
 48       #ifdef XMK_ADD_PRINTF_TIMER                                                00366              00365
 49         XMK_TRACE_EXIT("xmk_InitSystime");                                       00367              00366
 50       #endif                                                                     00368              00367
 51    ************* 51 Unchanged New File Lines Not Shown ****************************
 52       #elif defined(_GCC_)                                                       00420              00419
 53         /*                                                                       00421              00420
 54         ** Nothing specifically to be done.                                     00422              00421
 55         */                                                                      00423              00422
 56       #else                                                                      00424              00423
 57                                                                                  00425              00424
 58 I -     // #include "Fill_up_function_xmk_DeinitSystime_for_your_compiler"      00426 INSERT(NEW)          1
 59                                                                                  00427              00426   73
 60       #endif                                                                     00428              00427
 61                                                                                  00429              00428
 62       #ifdef XMK_ADD_PRINTF_TIMER                                                00430              00429
 63         XMK_TRACE_EXIT("xmk_DeInitSystime");                                     00431              00430
 64       #endif                                                                     00432              00431
 65    ************* 61 Unchanged New File Lines Not Shown ****************************
 66         /*                                                                       00494              00493
 67         ** Nothing specifically to be done.                                     00495              00494
 68         */                                                                      00496              00495
 69         NewTime = NewTime; /* Satisfy compiler for unused variable warning */   00497              00496
 70       #else                                                                      00498              00497
 71                                                                                  00499              00498
 72 I -     // #include "Fill_up_function_xmk_SetTime_for_your_compiler"           00500 INSERT(NEW)          1
 73                                                                                  00501              00500   175
 74       #endif                                                                     00502              00501
 75                                                                                  00503              00502
 76       #ifdef XMK_ADD_PRINTF_TIMER                                                00504              00503
 77         XMK_TRACE_EXIT("xmk_SetTime");                                           00505              00504
 78       #endif                                                                     00506              00505
 79    ************* 163 Unchanged New File Lines Not Shown ****************************
 80         #ifdef XMK_ADD_PRINTF_TIMER                                              00670              00669
 81           XMK_FUNCTION("xmk_NOW");                                               00671              00670
 82         #endif                                                                   00672              00671
 83         XMK_BEGIN_CRITICAL_PATH;                                                 00673              00672
 84         _SystemTime = SystemTime ;                                               00674              00673
 85         XMK_END_CRITICAL_PATH;                                                   00675              00674
 86 I -   #elif defined(GCCLEGOS)                                                   00676 INSERT(NEW)          9
 87 I -     #ifdef XMK_ADD_PRINTF_TIMER                                            00677 INSERT
 88 I -       XMK_FUNCTION("xmk_NOW");                                             00678 INSERT
 89 I -     #endif                                                                 00679 INSERT
 90 IM-                                                                           00680 MOVE(N TO O) 00180
 91 I -     XMK_BEGIN_CRITICAL_PATH;                                               00681 INSERT
 92 I -     _SystemTime = sys_time;                                                00682 INSERT
 93 I -     XMK_END_CRITICAL_PATH;                                                 00683 INSERT
 94 I -                                                                            00684 INSERT
 95 RN-  #else                                                                     00685 REFORM(NEW)  00675   1
 96                                                                                  00686              00676   1
 97 I -    // #include "Fill_up_function_xmk_NOW_for_your_compiler"               00687 INSERT(NEW)          1
 98       #endif                                                                     00688              00678   96
 99                                                                                  00689              00679
100       #ifdef XMK_ADD_PRINTF_TIMER                                                00690              00680
101         XMK_TRACE_EXIT("xmk_NOW");                                               00691              00681
102       #endif                                                                     00692              00682
103                                                                                  00693              00683
104    ************* 90 Unchanged New File Lines Not Shown ****************************
```

```
Panel 2 Report -- Old Source Matches + Deletes

New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\mk_stim.c  5/15/00 3:32:58 PM
Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\template\mk_stim.c  5/5/00 11:41:14 AM
Ref# ID  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8 N-REF# CHNG-TYPE   O-REF# LEN
   1    ************* 87 Unchanged New File Lines Not Shown ****************************
   2     #elif defined(MICROSOFT_C)                                               00088              00088
   3       #ifdef XMK_WINDOWS                                                     00089              00089
   4         #include <sys\timeb.h>                                               00090              00090
   5       #else                                                                  00091              00091
   6         #include "Error_in_mk_stim.c_NoTimerImplementationAvailable-02"     00092              00092
   7       #endif                                                                 00093              00093
   8
   9
  10     #endif                                                                   00096              00094   79
  11                                                                              00097              00095
  12     /*=========================================================================*/ 00098              00096
  13     /*-IMPORT*/                                                              00099              00097
  14                                                                              00100              00098
  15                                                                              00101              00099
  16    ************* 67 Unchanged New File Lines Not Shown ****************************
  17     #endif                                                                   00169              00167
  18                                                                              00170              00168
  19     #ifdef TCC80C196                                                         00171              00169
  20       static unsigned int LastTime;                                          00172              00170
  21     #endif                                                                   00173              00171
  22                                                                              00174              00172
  23 D - #if defined(MICROSOFT_C) || defined(BORLAND_C) || defined(GNU80166) || \        DELETED(OLD) 00173   2
  24 D -     defined(IARC51) || defined(KEIL_C51) || defined(ARM_THUMB) || \            DELETED      00174
  25         defined(TCC80166) || defined(MSP58C80) || defined(IARC7700) || \     00177              00175   5
  26         defined(HYPERSTONE) || defined(KEIL_C166) || defined(TCC80C196)      00178              00176
  27       static xmk_T_TIME SystemTime;                                          00179              00177
  28     #endif                                                                   00180              00178
  29                                                                              00181              00179
  30 DM-                                                                          00680 MOVE(O TO N) 00180
  31     /*=========================================================================*/ 00182              00181   180
  32     /*-MGG*/                                                                 00183              00182
  33                                                                              00184              00183
  34     /*+FHDR E*/                                                              00185              00184
  35     /*                                                                       00186              00185
  36     +-------------------------------------------------------------------------+ 00187              00186
  37    ************* 168 Unchanged New File Lines Not Shown ****************************
  38             XMK_BEGIN_CRITICAL_PATH;                                         00356              00355
```

```
 39            StartTime = UnixTime; /* OK for loosing precision in cast from long to u_ 00357              00356
 40            XMK_END_CRITICAL_PATH;                                                    00358              00357
 41          }                                                                           00359              00358
 42        #else                                                                         00360              00359
 43                                                                                      00361              00360
 44 D -     #include "Fill_up_function_xmk_InitSystime_for_your_compiler"                       DELETED(OLD) 00361   1
 45                                                                                      00363              00362   63
 46        #endif                                                                        00364              00363
 47                                                                                      00365              00364
 48        #ifdef XMK_ADD_PRINTF_TIMER                                                   00366              00365
 49          XMK_TRACE_EXIT("xmk_InitSystime");                                          00367              00366
 50        #endif                                                                        00368              00367
 51    ************* 51 Unchanged New File Lines Not Shown ****************************
 52        #elif defined(_GCC_)                                                          00420              00419
 53          /*                                                                          00421              00420
 54          ** Nothing specifically to be done.                                        00422              00421
 55          */                                                                          00423              00422
 56        #else                                                                         00424              00423
 57                                                                                      00425              00424
 58 D -     #include "Fill_up_function_xmk_DeinitSystime_for_your_compiler"                   DELETED(OLD) 00425   1
 59                                                                                      00427              00426   73
 60        #endif                                                                        00428              00427
 61                                                                                      00429              00428
 62        #ifdef XMK_ADD_PRINTF_TIMER                                                   00430              00429
 63          XMK_TRACE_EXIT("xmk_DeInitSystime");                                        00431              00430
 64        #endif                                                                        00432              00431
 65    ************* 61 Unchanged New File Lines Not Shown ****************************
 66          /*                                                                        00494              00493
 67          ** Nothing specifically to be done.                                      00495              00494
 68          */                                                                        00496              00495
 69          NewTime = NewTime; /* Satisfy compiler for unused variable warning */    00497              00496
 70        #else                                                                       00498              00497
 71                                                                                    00499              00498
 72 D -     #include "Fill_up_function_xmk_SetTime_for_your_compiler"                       DELETED(OLD) 00499   1
 73                                                                                    00501              00500   175
 74        #endif                                                                      00502              00501
 75                                                                                    00503              00502
 76        #ifdef XMK_ADD_PRINTF_TIMER                                                 00504              00503
 77          XMK_TRACE_EXIT("xmk_SetTime");                                            00505              00504
 78        #endif                                                                      00506              00505
 79    ************* 163 Unchanged New File Lines Not Shown ****************************
 80          #ifdef XMK_ADD_PRINTF_TIMER                                               00670              00669
 81            XMK_FUNCTION("xmk_NOW");                                                00671              00670
 82          #endif                                                                    00672              00671
 83          XMK_BEGIN_CRITICAL_PATH;                                                  00673              00672
 84          _SystemTime = SystemTime ;                                                00674              00673
 85          XMK_END_CRITICAL_PATH;                                                    00675              00674
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95 RO-   #else                                                                       00675 REFORM(OLD)  00675
 96                                                                                  00686              00676   1
 97 D -     #include "Fill_up_function_xmk_NOW_for_your_compiler"                         DELETED(OLD) 00677   1
 98        #endif                                                                      00688              00678   96
 99                                                                                    00689              00679
100        #ifdef XMK_ADD_PRINTF_TIMER                                                 00690              00680
101          XMK_TRACE_EXIT("xmk_NOW");                                                00691              00681
102        #endif                                                                      00692              00682
103                                                                                    00693              00683
104    ************* 90 Unchanged New File Lines Not Shown ****************************

DLSupC32 V5.6f -  Compare Totals And Statistics Section
New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\mk_stim.c  5/15/00 3:32:58 PM
Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\template\mk_stim.c  5/5/00 11:41:14 AM

    765 Number Of Line Matches         19 Total Changes (Paired+NonPaired Chg)
      1 Reformatted Lines               7 Paired Changes (Refm+Paired Ins/Del)
     17 New File Line Insertions       11 Non-Paired Inserts (Ins Paired-Chg)
      7 Old File Line Deletions         1 Non-Paired Deletes (Del Paired-Chg)
    783 New File Lines Processed      773 Old File Lines Processed
      1 Number of Moved lines detected
  26467 New File Total Bytes        26240 Old File Total Bytes
LISTING-TYPE = Chng  OPTIONS = Sbsl Lorl Fmvl  LONGEST-LINE = (99/99) PASSES = 1
INFORM: The processed files have differences.
```

G.3 mk_user.c Change Log

```
(UnRegistered) DLSupC32 File Compare Report -  9/10/00  5:00:48 PM

Panel 1 Report -- New Source Matches + Inserts
New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\mk_user.c  5/15/00 3:39:06 PM
Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\template\mk_user.c  3/3/00 2:34:44 PM
Ref# ID  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8 N-REF# CHNG-TYPE   O-REF# LEN
   1    ************* 371 Unchanged New File Lines Not Shown ****************************
   2       */                                                                        00372              00372
```

```
  3        switch (xmk_TmpSignalID)                                                    00373              00373
  4        {                                                                           00374              00374
  5           /*                                                                       00375              00375
  6           **  S D L   T i m e r s ...                                              00376              00376
  7           */                                                                      00377              00377
  8 I -    /* case SDL_Timer1  : return (XPTID_ProcessName)        break;              00378 INSERT(NEW)         1
  9           case SDL_Timer2  : return (XPTID_ProcessName)        break;              00379              00379   3
 10           case SDL_Timer3  : return (XPTID_ProcessName)        break;              00380              00380
 11           case SDL_TimerN  : return (XPTID_ProcessName)        break;              00381              00381
 12 I -    */                                                                          00382 INSERT(NEW)         1
 13           /*                                                                      00383              00383   3
 14           **  O r d i n a r y  S D L   S i g n a l s ...                          00384              00384
 15           */                                                                      00385              00385
 16 I -   /*   case SDL_Signal1 : return (XPTID_ProcessName)         break;            00386 INSERT(NEW)         1
 17           case SDL_Signal2 : return (XPTID_ProcessName)        break;              00387              00387   4
 18           case SDL_Signal3 : return (XPTID_ProcessName)        break;              00388              00388
 19                 ......                                                           00389              00389
 20           case SDL_SignalN : return (XPTID_ProcessName)        break;              00390              00390
 21 I -   */                                                                          00391 INSERT(NEW)         2
 22 I -    /*   default          : ErrorHandler (ERR_N_NO_RCV);                       00392 INSERT
 23                              return (xNULLPID)                 break;              00393              00393   1
 24 I -    */  }                                                                     00394 INSERT(NEW)         1
 25                                                                                  00395              00395   330
 26                                                                                  00396              00396
 27     } /* END OF FUNCTION */                                                      00397              00397
 28                                                                                  00398              00398
 29     #endif /* ... XMK_USE_RECEIVER_PID_IN_SIGNAL */                             00399              00399
 30     #ifdef XMK_USE_KERNEL_WDTRIGGER                                              00400              00400
 31    ************* 324 Unchanged New File Lines Not Shown ****************************
```

```
Panel 2 Report -- Old Source Matches + Deletes

New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\mk_user.c  5/15/00 3:39:06 PM
Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\template\mk_user.c  3/3/00 2:34:44 PM
Ref# ID  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8 N-REF# CHNG-TYPE   O-REF# LEN
   1    ************* 371 Unchanged New File Lines Not Shown ****************************
   2        */                                                                      00372              00372
   3        switch (xmk_TmpSignalID)                                                00373              00373
   4        {                                                                       00374              00374
   5           /*                                                                   00375              00375
   6           **  S D L   T i m e r s ...                                          00376              00376
   7           */                                                                  00377              00377
   8 D -       case SDL_Timer1  : return (XPTID_ProcessName)        break;                DELETED(OLD) 00378   1
   9           case SDL_Timer2  : return (XPTID_ProcessName)        break;          00379              00379   3
  10           case SDL_Timer3  : return (XPTID_ProcessName)        break;          00380              00380
  11           case SDL_TimerN  : return (XPTID_ProcessName)        break;          00381              00381
  12 D -                                                                                DELETED(OLD) 00382   1
  13           /*                                                                   00383              00383   3
  14           **  O r d i n a r y  S D L   S i g n a l s ...                       00384              00384
  15           */                                                                  00385              00385
  16 D -       case SDL_Signal1 : return (XPTID_ProcessName)        break;                DELETED(OLD) 00386   1
  17           case SDL_Signal2 : return (XPTID_ProcessName)        break;          00387              00387   4
  18           case SDL_Signal3 : return (XPTID_ProcessName)        break;          00388              00388
  19                 ......                                                       00389              00389
  20           case SDL_SignalN : return (XPTID_ProcessName)        break;          00390              00390
  21 D -                                                                                DELETED(OLD) 00391   2
  22 D -       default          : ErrorHandler (ERR_N_NO_RCV);                           DELETED      00392
  23                              return (xNULLPID)                 break;          00393              00393   1
  24 D -    }                                                                           DELETED(OLD) 00394   1
  25                                                                                  00395              00395   330
  26                                                                                  00396              00396
  27     } /* END OF FUNCTION */                                                      00397              00397
  28                                                                                  00398              00398
  29     #endif /* ... XMK_USE_RECEIVER_PID_IN_SIGNAL */                             00399              00399
  30     #ifdef XMK_USE_KERNEL_WDTRIGGER                                              00400              00400
  31    ************* 324 Unchanged New File Lines Not Shown ****************************

 DLSupC32 V5.6f -  Compare Totals And Statistics Section
 New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\mk_user.c  5/15/00 3:39:06 PM
 Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\template\mk_user.c  3/3/00 2:34:44 PM

     718 Number Of Line Matches          6 Total Changes (Paired+NonPaired Chg)
       0 Reformatted Lines               6 Paired Changes (Refm+Paired Ins/Del)
       6 New File Line Insertions        0 Non-Paired Inserts (Ins Paired-Chg)
       6 Old File Line Deletions         0 Non-Paired Deletes (Del Paired-Chg)
     724 New File Lines Processed      724 Old File Lines Processed
       0 Number of Moved lines detected
   29619 New File Total Bytes        29603 Old File Total Bytes
 LISTING-TYPE = Chng  OPTIONS = Sbsl Lorl Fmvl  LONGEST-LINE = (99/99) PASSES = 1
 INFORM: The processed files have differences.
```

G.4 mk_cpu.c Change Log

```
(UnRegistered) DLSupC32 File Compare Report -  9/10/00  5:00:48 PM

Panel 1 Report -- New Source Matches + Inserts
New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\mk_cpu.c  5/15/00 3:30:12 PM
Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\template\mk_cpu.c  5/5/00 11:41:14 AM
Ref# ID  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8 N-REF# CHNG-TYPE   O-REF# LEN
   1    ************* 94 Unchanged New File Lines Not Shown ****************************
   2     #if defined(BORLAND_C)                                                     00095              00095
   3       #include <stdarg.h>                                                     00096              00096
```

```
 4     #endif                                                                    00097             00097
 5                                                                               00098             00098
 6     #if defined(MICROSOFT_C)                                                  00099             00099
 7       #include <stdarg.h>                                                    00100             00100
 8
 9
10
11
12
13
14     #endif                                                                    00101             00107   23
15                                                                              00102             00108
16                                                                              00103             00109
17     /*=========================================================================*/ 00104             00110
18     /*-IMPORT*/                                                              00105             00111
19                                                                              00106             00112
20    ************* 11 Unchanged New File Lines Not Shown ***************************
21       ** for this is that the functions of this module operate on global     00118             00124
22       ** variables and partitioned systems may preempt each other when       00119             00125
23       ** an Real Time Operating System is used.                             00120             00126
24       ** However, users can use this module as a basis for implementing     00121             00127
25       ** their own terminal and memory management functions.                00122             00128
26       */                                                                     00123             00129
27 I -   // #include "ERROR_in_mk_cpu_c_Module_cannot_be_used_for_Partitioning"  00124 INSERT(NEW)         1
28     #endif /* ... XSYSID */                                                  00125             00131   15
29                                                                              00126             00132
30                                                                              00127             00133
31     #ifdef P                                                                 00128             00134
32     #undef P                                                                 00129             00135
33     #endif                                                                   00130             00136
34    ************* 3 Unchanged New File Lines Not Shown ***************************
35     #define xmk_exit exit                                                    00134             00140
36     /*--------------------  Functions    ----------------------------------------*/ 00135             00141
37     #ifdef XMK_UNIX                                                          00136             00142
38       static void sig_catch(int);                                            00137             00143
39     #endif                                                                   00138             00144
40                                                                              00139             00145
41
42
43
44
45
46
47     /*--------------------  Variables    ----------------------------------------*/ 00140             00152   81
48                                                                              00141             00153
49     #ifdef XMK_UNIX                                                          00142             00154
50       struct termios tty_buf, xmk_OldTTYSettings;                            00143             00155
51     #endif                                                                   00144             00156
52                                                                              00145             00157
53    ************* 69 Unchanged New File Lines Not Shown ***************************
54           xmk_PutString_is_not_filled_yet = 0;                              00215             00227
55         #endif                                                               00216             00228
56       #else                                                                  00217             00229
57         /*                                                                   00218             00230
58         ** User must add his own C compiler here                            00219             00231
59         */                                                                   00220             00232
60 I -     // xmk_PutString_is_not_filled_yet = 0;                             00221 INSERT(NEW)         1
61                                                                              00222             00234   371
62       #endif                                                                 00223             00235
63                                                                              00224             00236
64       return ;                                                               00225             00237
65     }                                                                        00226             00238
66                                                                              00227             00239
67    ************* 359 Unchanged New File Lines Not Shown ***************************
68         ** User must insert his own printf function                          00587             00599
69         */                                                                   00588             00600
70       #else                                                                  00589             00601
71         /*                                                                   00590             00602
72         ** User must add his own C compiler here                            00591             00603
73         */                                                                   00592             00604
74 I -     // xmk_printf_not_filled_yet = 0;                                   00593 INSERT(NEW)         1
75       #endif                                                                 00594             00606   114
76       return (0);                                                            00595             00607
77                                                                              00596             00608
78     }                                                                        00597             00609
79                                                                              00598             00610
80     /*+FHDR E*/                                                            00599             00611
81    ************* 102 Unchanged New File Lines Not Shown ***************************
82           ** DO NOT USE xmk_printf because this uses malloc (512) !!!!      00702             00714
83           **                                                                00703             00715
84           ** An example for XMK_NO_MORE_MEMORY could look like :            00704             00716
85           ** #define XMK_NO_MORE_MEMORY \                                   00705             00717
86              { fprintf (stderr, "**ERROR: No more memory available.exit\n"); xCloseE 00706             00718
87           */                                                                00707             00719
88 I -       // #include "ERROR_in_mk_cpu_c_You_Must_Define_XMK_NO_MORE_MEMORY"  00708 INSERT(NEW)         1
89         #endif                                                               00709             00721   73
90       }                                                                      00710             00722
91       return (P);                                                            00711             00723
92                                                                              00712             00724
93     }                                                                        00713             00725
94                                                                              00714             00726
95    ************* 67 Unchanged New File Lines Not Shown ***************************

Panel 2 Report -- Old Source Matches + Deletes

New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\mk_cpu.c  5/15/00 3:30:12 PM
Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\template\mk_cpu.c  5/5/00 11:41:14 AM
Ref# ID  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8 N-REF# CHNG-TYPE   O-REF# LEN
   1    ************* 94 Unchanged New File Lines Not Shown ***************************
   2     #if defined(BORLAND_C)                                                 00095             00095
   3       #include <stdarg.h>                                                  00096             00096
```

```
 4     #endif                                                                    00097              00097
 5                                                                               00098              00098
 6     #if defined(MICROSOFT_C)                                                  00099              00099
 7       #include <stdarg.h>                                                    00100              00100
 8 D -   #ifdef __cplusplus                                                            DELETED(OLD) 00101   6
 9 D -     extern "C"                                                                  DELETED      00102
10 D -     {                                                                           DELETED      00103
11 D -       #include <conio.h>                                                       DELETED      00104
12 D -     }                                                                           DELETED      00105
13 D -   #endif                                                                       DELETED      00106
14     #endif                                                                    00101              00107   23
15                                                                              00102              00108
16                                                                              00103              00109
17     /*=========================================================================*/ 00104              00110
18     /*-IMPORT*/                                                               00105              00111
19                                                                              00106              00112
20    ************* 11 Unchanged New File Lines Not Shown ****************************
21       ** for this is that the functions of this module operate on global      00118              00124
22       ** variables and partitioned systems may preempt each other when        00119              00125
23       ** an Real Time Operating System is used.                               00120              00126
24       ** However, users can use this module as a basis for implementing      00121              00127
25       ** their own terminal and memory management functions.                 00122              00128
26       */                                                                     00123              00129
27 D -   #include "ERROR_in_mk_cpu_c_Module_cannot_be_used_for_Partitioning"            DELETED(OLD) 00130   1
28     #endif /* ... XSYSID */                                                  00125              00131   15
29                                                                              00126              00132
30                                                                              00127              00133
31     #ifdef P                                                                 00128              00134
32     #undef P                                                                 00129              00135
33     #endif                                                                   00130              00136
34    ************* 3 Unchanged New File Lines Not Shown ****************************
35     #define xmk_exit exit                                                    00134              00140
36     /*--------------------  Functions    ----------------------------------------*/ 00135              00141
37     #ifdef XMK_UNIX                                                          00136              00142
38       static void sig_catch(int);                                            00137              00143
39     #endif                                                                   00138              00144
40                                                                              00139              00145
41 D - #ifdef __cplusplus                                                               DELETED(OLD) 00146   6
42 D -   void * operator new (unsigned int pp, xSpecial vv)                             DELETED      00147
43 D -   {                                                                             DELETED      00148
44 D -     return (void *)vv;                                                          DELETED      00149
45 D -   }                                                                             DELETED      00150
46 D - #endif                                                                         DELETED      00151
47     /*--------------------  Variables    ----------------------------------------*/ 00140              00152   81
48                                                                              00141              00153
49     #ifdef XMK_UNIX                                                          00142              00154
50       struct termios tty_buf, xmk_OldTTYSettings;                            00143              00155
51     #endif                                                                   00144              00156
52                                                                              00145              00157
53    ************* 69 Unchanged New File Lines Not Shown ****************************
54           xmk_PutString_is_not_filled_yet = 0;                               00215              00227
55         #endif                                                               00216              00228
56       #else                                                                  00217              00229
57         /*                                                                   00218              00230
58         ** User must add his own C compiler here                            00219              00231
59         */                                                                   00220              00232
60 D -     xmk_PutString_is_not_filled_yet = 0;                                       DELETED(OLD) 00233   1
61                                                                              00222              00234   371
62       #endif                                                                 00223              00235
63                                                                              00224              00236
64       return ;                                                               00225              00237
65     }                                                                        00226              00238
66                                                                              00227              00239
67    ************* 359 Unchanged New File Lines Not Shown ****************************
68         ** User must insert his own printf function                          00587              00599
69         */                                                                   00588              00600
70       #else                                                                  00589              00601
71         /*                                                                   00590              00602
72         ** User must add his own C compiler here                            00591              00603
73         */                                                                   00592              00604
74 D -     xmk_printf_not_filled_yet = 0;                                             DELETED(OLD) 00605   1
75       #endif                                                                 00594              00606   114
76       return (0);                                                            00595              00607
77                                                                              00596              00608
78     }                                                                        00597              00609
79                                                                              00598              00610
80     /*+FHDR E*/                                                             00599              00611
81    ************* 102 Unchanged New File Lines Not Shown ****************************
82           ** DO NOT USE xmk_printf because this uses malloc (512) !!!!      00702              00714
83           **                                                                00703              00715
84           ** An example for XMK_NO_MORE_MEMORY could look like :            00704              00716
85           ** #define XMK_NO_MORE_MEMORY \                                   00705              00717
86              { fprintf (stderr, "**ERROR: No more memory available.exit\n"); xCloseE 00706              00718
87           */                                                                 00707              00719
88 D -       #include "ERROR_in_mk_cpu_c_You_Must_Define_XMK_NO_MORE_MEMORY"          DELETED(OLD) 00720   1
89         #endif                                                               00709              00721   73
90       }                                                                      00710              00722
91       return (P);                                                            00711              00723
92                                                                              00712              00724
93     }                                                                        00713              00725
94                                                                              00714              00726
95    ************* 67 Unchanged New File Lines Not Shown ****************************

DLSupC32 V5.6f -  Compare Totals And Statistics Section
New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\mk_cpu.c  5/15/00 3:30:12 PM
Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\template\mk_cpu.c  5/5/00 11:41:14 AM

    777 Number Of Line Matches         16 Total Changes (Paired+NonPaired Chg)
      0 Reformatted Lines               4 Paired Changes (Refm+Paired Ins/Del)
      4 New File Line Insertions        0 Non-Paired Inserts (Ins Paired-Chg)
     16 Old File Line Deletions        12 Non-Paired Deletes (Del Paired-Chg)
```

```
    781 New File Lines Processed      793 Old File Lines Processed
      0 Number of Moved lines detected
  28731 New File Total Bytes        28923 Old File Total Bytes
LISTING-TYPE = Chng  OPTIONS = Sbsl Lorl Fmvl  LONGEST-LINE = (112/112) PASSES = 1
INFORM: The processed files have differences.
```

G.5 ml_mem.c Change Log

```
(UnRegistered) DLSupC32 File Compare Report -  9/10/00  5:00:48 PM

Panel 1 Report -- New Source Matches + Inserts
New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\ml_mem.c  5/15/00 3:17:36 PM
Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\kernel\ml_mem.c  2/24/00 2:51:24 PM
Ref# ID  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8 N-REF# CHNG-TYPE   O-REF# LEN
   1    ************ 655 Unchanged New File Lines Not Shown ****************************
   2     #endif                                                                          00656              00656
   3                                                                                     00657              00657
   4     /*-FDEF E*/                                                                     00658              00658
   5                                                                                     00659              00659
   6     {                                                                               00660              00660
   7       register xmk_T_MBLOCK * xmk_MemPtr ;                                          00661              00661
   8 I -   #if defined(ARM_THUMB) || defined(_GCC_) || defined(MCC68K) || defined(GCCLEGO 00662 INSERT(NEW)          1
   9         size_t __TempSize;                                                          00663              00663   1180
  10       #endif                                                                        00664              00664
  11                                                                                     00665              00665
  12       #ifdef XMK_ADD_PRINTF_MEMORY                                                  00666              00666
  13         XMK_FUNCTION("xmk_Malloc");                                                 00667              00667
  14       #endif                                                                        00668              00668
  15    ************ 1174 Unchanged New File Lines Not Shown ****************************

Panel 2 Report -- Old Source Matches + Deletes

New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\ml_mem.c  5/15/00 3:17:36 PM
Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\kernel\ml_mem.c  2/24/00 2:51:24 PM
Ref# ID  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8 N-REF# CHNG-TYPE   O-REF# LEN
   1    ************ 655 Unchanged New File Lines Not Shown ****************************
   2     #endif                                                                          00656              00656
   3                                                                                     00657              00657
   4     /*-FDEF E*/                                                                     00658              00658
   5                                                                                     00659              00659
   6     {                                                                               00660              00660
   7       register xmk_T_MBLOCK * xmk_MemPtr ;                                          00661              00661
   8 D -   #if defined(ARM_THUMB) || defined(_GCC_) || defined(MCC68K)                         DELETED(OLD) 00662   1
   9         size_t __TempSize;                                                          00663              00663   1180
  10       #endif                                                                        00664              00664
  11                                                                                     00665              00665
  12       #ifdef XMK_ADD_PRINTF_MEMORY                                                  00666              00666
  13         XMK_FUNCTION("xmk_Malloc");                                                 00667              00667
  14       #endif                                                                        00668              00668
  15    ************ 1174 Unchanged New File Lines Not Shown ****************************

 DLSupC32 V5.6f -  Compare Totals And Statistics Section
 New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\ml_mem.c  5/15/00 3:17:36 PM
 Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\kernel\ml_mem.c  2/24/00 2:51:24 PM

    1841 Number Of Line Matches          1 Total Changes (Paired+NonPaired Chg)
       0 Reformatted Lines               1 Paired Changes (Refm+Paired Ins/Del)
       1 New File Line Insertions        0 Non-Paired Inserts (Ins Paired-Chg)
       1 Old File Line Deletions         0 Non-Paired Deletes (Del Paired-Chg)
    1842 New File Lines Processed     1842 Old File Lines Processed
       0 Number of Moved lines detected
   63207 New File Total Bytes        63186 Old File Total Bytes
 LISTING-TYPE = Chng  OPTIONS = Sbsl Lorl Fmvl  LONGEST-LINE = (103/103) PASSES = 1
 INFORM: The processed files have differences.
```

G.6 sctpred.c Change Log

```
(UnRegistered) DLSupC32 File Compare Report -  9/10/00  5:00:48 PM

Panel 1 Report -- New Source Matches + Inserts
New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\sctpred.c  6/15/00 4:34:30 PM
Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\kernel\sctpred.c  3/3/00 2:34:44 PM
Ref# ID  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8 N-REF# CHNG-TYPE   O-REF# LEN
   1    ************ 32 Unchanged New File Lines Not Shown ****************************
   2     03   Utility functions                                                          00033              00033
   3     04   Read and Write functions                                                   00034              00034
   4     */                                                                            00035              00035
   5                                                                                     00036              00036
   6     #include "scttypes.h"                                                          00037              00037
   7                                                                                     00038              00038
   8 I - #define abs(j) ((j)<0?-(j):(j)) /*?????*/                                     00039 INSERT(NEW)          2
   9 I -                                                                               00040 INSERT
```

```
  10     #ifndef XSYSD                                                          00041             00039   4347
  11     #define XSYSD                                                          00042             00040
  12     #endif                                                                 00043             00041
  13                                                                            00044             00042
  14     #ifdef XVALIDATOR_LIB                                                  00045             00043
  15     SDL_Time xMaxTime = {2147483647,0};                                    00046             00044
  16    ************ 4341 Unchanged New File Lines Not Shown ***************************

Panel 2 Report -- Old Source Matches + Deletes

New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\sctpred.c  6/15/00 4:34:30 PM
Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\kernel\sctpred.c  3/3/00 2:34:44 PM
Ref# ID  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8 N-REF# CHNG-TYPE   O-REF# LEN
   1    ************ 32 Unchanged New File Lines Not Shown ***************************
   2     03   Utility functions                                                00033             00033
   3     04   Read and Write functions                                         00034             00034
   4     */                                                                    00035             00035
   5                                                                           00036             00036
   6     #include "scttypes.h"                                                 00037             00037
   7                                                                           00038             00038
   8
   9
  10     #ifndef XSYSD                                                          00041             00039   4347
  11     #define XSYSD                                                          00042             00040
  12     #endif                                                                 00043             00041
  13                                                                            00044             00042
  14     #ifdef XVALIDATOR_LIB                                                  00045             00043
  15     SDL_Time xMaxTime = {2147483647,0};                                    00046             00044
  16    ************ 4341 Unchanged New File Lines Not Shown ***************************

 DLSupC32 V5.6f -  Compare Totals And Statistics Section
 New File = c:\winnt\profiles\x-tni\desktop\lri\lri\application._0\node._1\component._2\_user_defined_\sctpred.c  6/15/00 4:34:30 PM
 Old File = c:\telelogictau40\sdt\sdtdir\wini386\cmicro\kernel\sctpred.c  3/3/00 2:34:44 PM

    4385 Number Of Line Matches          2 Total Changes (Paired+NonPaired Chg)
       0 Reformatted Lines               0 Paired Changes (Refm+Paired Ins/Del)
       2 New File Line Insertions        2 Non-Paired Inserts (Ins Paired-Chg)
       0 Old File Line Deletions         0 Non-Paired Deletes (Del Paired-Chg)
    4387 New File Lines Processed     4385 Old File Lines Processed
       0 Number of Moved lines detected
  126085 New File Total Bytes       126040 Old File Total Bytes
 LISTING-TYPE = Chng  OPTIONS = Sbsl Lorl Fmvl  LONGEST-LINE = (92/92) PASSES = 1
 INFORM: The processed files have differences.
```