



Ana Vazquez <ana.vazquez.alonso@gmail.com>

Weekly report

9 messages

Anita <ana.vazquez.alonso@gmail.com>

Wed, Aug 10, 2011 at 10:24 PM

To: sylvain.joyeux@dfki.de

Hi!

Here it is a summary of the things I've been working this week:

I made a cross compiler (with support for c and c++) for x86 RTEMS. This was needed since the gcc used to generate binaries with RTEMS, even for x86 targets, can not make use of libraries like glibc, instead newlib (a small libc for embedded systems) needs to be used.

Then, I started to add RTEMS support in Rock/RTT. I compiled Xerces and adapted the compilation systems of Boost and OmniORB to make used of the cross compiler. I still need to do it in a clean and automatic way but, for now it's working and generating all the needed libraries using the x86 cross compiler.

The previous steps have been sucesfully implemented and at this moment I am already translating orocos system calls to RTEMS. For this, I'm using the existing GNU Linux RTT implementation as a reference since RTEMS provides nearly all the POSIX services. Not all of them though, so the translation is not always straight forward and some adaptation is needed.

So far, I'm optimistic with the initial RTEMS x86 port deadline.

Kind regards,
Ana

Sylvain Joyeux <sylvain.joyeux@dfki.de>

Fri, Aug 12, 2011 at 7:19 PM

To: Anita <ana.vazquez.alonso@gmail.com>

Cc: "rock-dev@dfki.de" <rock-dev@dfki.de>

On 08/10/2011 10:24 PM, Anita wrote:

Hi!

Here it is a summary of the things I've been working this week:

I made a cross compiler (with support for c and c++) for x86 RTEMS. This was needed since the gcc used to generate binaries with RTEMS, even for x86 targets, can not make use of libraries like glibc, instead newlib (a small libc for embedded systems) needs to be used.

Then, I started to add RTEMS support in Rock/RTT. I compiled Xerces and adapted the compilation systems of Boost and OmniORB to make used of the cross compiler. I still need to do it in a clean and automatic way but, for now it's working and generating all the needed libraries using the x86 cross compiler.

The previous steps have been sucesfully implemented and at this moment I am already translating orocos system calls to RTEMS. For this, I'm using the existing GNU Linux RTT implementation as a reference since RTEMS provides nearly all the POSIX services. Not all of them though, so the

translation is not always straight forward and some adaptation is needed.

So far, I'm optimistic with the initial RTEMS x86 port deadline.

Thanks for the status update. This sounds promising !

If you don't mind, I would suggest that you send these updates to rock-dev@dfki.de. I'm pretty sure that other members of the rock team would be glad to know about your progress ;-)

Sylvain

Anita <ana.vazquez.alonso@gmail.com>
To: Sylvain Joyeux <sylvain.joyeux@dfki.de>

Fri, Aug 19, 2011 at 11:24 AM

Hi Sylvain!

Here it is a summary of the things I've been working this week:

This week I have progressed less than expected. The dynamic libraries compiled using the cross compiler for OmniORB were not generated correctly and due to this I was getting duplicate symbols on the final link phase. I lost several days trying to find and fix this problem but it is now solved.

Now all the libraries (except `librtt-transport-mqueue-tems.so.2.4.0`) are being generated correctly, both statically and dynamically using the cross compiler.

About `librtt-transport-mqueue-rtems.so` error. For some reason, compiling `ParsedStateMachine.cpp`, `i386-rtems-as` does not like the output generated by `cc1plus`. I'm still working on this and I think I will find the error soon.

In parallel, I've been adding RTEMS support to Orogen and Autoproj.

So far, I specify the compilation flags for `rtt` cmake in `override.rb` but I think it would be a better idea to move them to other location based on the selected target (I need think more about this)

Anyway, lot of work in front of me, but I'm quite optimistic with the progress achieved until now. And since everybody love screenshots (even of a compilation process!!), I attached a png with the `autoproj rtems` building process :)

Greetings, Ana



autoprojRttBuild.png
66K

Sylvain Joyeux <sylvain.joyeux@dfki.de>
To: "rock-dev@dfki.de" <rock-dev@dfki.de>

Fri, Aug 19, 2011 at 11:39 AM

For info, everyone !

[Quoted text hidden]

Rock-dev mailing list
Rock-dev@dfki.de
<http://www.dfki.de/mailman/cgi-bin/listinfo/rock-dev>

autoprojRttBuild.png



66K

Sylvain Joyeux <sylvain.joyeux@dfki.de>
To: "rock-dev@dfki.de" <rock-dev@dfki.de>

Fri, Aug 19, 2011 at 11:43 AM

And now, some of my own comments ...

Summary: cool !

On 08/19/2011 11:39 AM, Sylvain Joyeux wrote:

> About librtt-transport-mqueue-rtems.so error. For some reason,
> compiling ParsedStateMachine.cpp , i386-rtems-as does not like the
> output generated by cc1plus. I'm still working on this and I think I
> will find the error soon.

I would not spend too much time fixing this. The internal state machine parser is not used at all in rock, and even in RTT it is being (slowly) superseded by the Lua scripting engine.

> In parallel, I've been adding RTEMS support to Orogen and Autoproj.
> So far, I specify the compilation flags for rtt cmake in override.rb
> but I think it would be a better idea to move them to other location
> based on the selected target (I need think more about this)

That would be a nice addition. In principle, the flags are set in the rock.toolchain package set (look into the autobuild files) based on the selected target.

—

Sylvain Joyeux
Space& Security Robotics

!!! Achtung, neue Telefonnummer!!!

Standort Bremen:
DFKI GmbH
Robotics Innovation Center
Robert-Hooke-Straße 5
28359 Bremen, Germany

Phone: +49 (0)421 178-454136
Fax: +49 (0)421 218-454150
E-Mail: robotik@dfki.de

Weitere Informationen: <http://www.dfki.de/robotik>

Deutsches Forschungszentrum fuer Kuenstliche Intelligenz GmbH
Firmensitz: Trippstadter Straße 122, D-67663 Kaiserslautern
Geschaeftsfuehrung: Prof. Dr. Dr. h.c. mult. Wolfgang Wahlster
(Vorsitzender) Dr. Walter Olthoff
Vorsitzender des Aufsichtsrats: Prof. Dr. h.c. Hans A. Aukes
Amtsgericht Kaiserslautern, HRB 2313
Sitz der Gesellschaft: Kaiserslautern (HRB 2313)
USt-Id.Nr.: DE 148646973
Steuernummer: 19/673/0060/3

Anita <ana.vazquez.alonso@gmail.com>
To: "rock-dev@dfki.de" <rock-dev@dfki.de>

Sat, Sep 3, 2011 at 7:36 PM

Dear all,

This is a summary of what I've been doing this week in the ESA Summer of code:

I've started creating the cross compiler for ERC32, LEON2 and LEON3. I successfully compiled binutils, gcc, newlib and RTEMS with the SPARC cross compiler. I'm working right now in making boost and omniORB run in SPARC/RTEMS.

In parallel, I'm trying to compile a more complex tutorial on RTEMS x86 using rock-create-orogen. Due to this, initially I made the mistake to compile several libraries like utimm, sisl, base/types, etc using the cross compiler for RTEMS.

I managed to compile some of these packages, but after several days I crashed into utilrb and then, I realized that make ruby works on top of RTEMS will not be an easy job, and more important, it's probably not necessary at all :(

As far as I understand, these libraries are used only by oroGen in order to generate code from templates (but in the host computer, not in the target).

The only modification needed in oroGen would be in the generation of Makefiles, adding there support for RTEMS and the specified target. Am I right?

Cheers!
Ana

[Quoted text hidden]

Sylvain Joyeux <sylvain.joyeux@dfki.de>
To: Anita <ana.vazquez.alonso@gmail.com>
Cc: "rock-dev@dfki.de" <rock-dev@dfki.de>

Mon, Sep 5, 2011 at 10:59 AM

On 09/03/2011 07:36 PM, Anita wrote:

In parallel, I'm trying to compile a more complex tutorial on RTEMS x86 using rock-create-orogen. Due to this, initially I made the mistake to compile several libraries like utimm, sisl, base/types, etc using the cross compiler for RTEMS.

I managed to compile some of these packages, but after several days I crashed into utilrb and then, I realized that make ruby works on top of RTEMS will not be an easy job, and more important, it's probably not necessary at all :(

As far as I understand, these libraries are used only by oroGen in

order to generate code from templates (but in the host computer, not in the target).

The only modification needed in oroGen would be in the generation of Makefiles, adding there support for RTEMS and the specified target. Am I right?

You are right: oroGen and ruby are not needed on the RTEMS target.

However, base/types, sisl and so on are actual libraries (i.e. they are used by the functionality of Rock).

Also: I do not completely understand the "generation of Makefiles". Cannot you cross-compile for RTEMS using CMake ?

As a first target, I would suggest that you try to get the trajectory follower orogen package to cross-compile and run on RTEMS. It depends on the base libraries.

[Quoted text hidden]

Anita <ana.vazquez.alonso@gmail.com>
To: Sylvain Joyeux <sylvain.joyeux@dfki.de>

Mon, Sep 5, 2011 at 2:27 PM

Sylvain, thank you for your reply.

Compiling some tutorials, I have found out that oroGen needs for the target: libbase, rtt and typelib.

So I managed to compile with the cross compiler sisl, utilmm, libbase. But trying to compile typelib I found out that it depends on utilrb and therefore ruby, so that's why I was asking about dependencies for the target in my previous mail.

Regarding the autogenerated Makefiles, sorry I meant CMake. Modification in CMake in order to add RTEMS flags, and make use of static (and not dynamic) libraries.

Greetings

Ana.

[Quoted text hidden]

Sylvain Joyeux <sylvain.joyeux@dfki.de>
To: Anita <ana.vazquez.alonso@gmail.com>

Mon, Sep 5, 2011 at 2:54 PM

On 09/05/2011 02:27 PM, Anita wrote:

Sylvain, thank you for your reply.

Compiling some tutorials, I have found out that oroGen needs for the target: libbase, rtt and typelib.

libbase is needed on the target, but not because of oroGen, only because of the library / component dependencies.

typelib is needed on the target as well, but just the C++ library part (not the ruby part)

everything ruby-related should be left out.

Regarding the autogenerated Makefiles, sorry I meant CMake.

Modification in CMake in order to add RTEMS flags, and make use of static (and not dynamic) libraries.

OK. Thanks for the clarification

Sylvain