Manuale porting RTEMS su Raspberry Pi

Clark Ezpeleta

May 12, 2020

1 Introduzione

Viene utilizzato il BSP (Board Support Package) di RTEMS per Raspberry Pi, arm-rtems5.
Per l'installazione abbiamo a disposizione l'architettura per raspberrypi collocata in ..arm/raspberrypi.
Viene utilizzata la v 5.1.0, versione stabile e con importanti drivers a disposizione.
La tool suite di RTEMS consiste in un cross tool chain(Binutils, GCC, GDB, Newlib, etc.) per l'architettura target, e dei RTEMS tools forniti dal progetto RTEMS.
La tool suite viene intallata e buildata con RTEMS Source Builder (RSB).
Da default RSB inizializzerà il prefix path con un path specifico del sistema operativo del host computer, più la dicitura "rtems" e la versione.
i.e. su Linux avrei /opt/rtems/5.
Avere il numero di versione nel path permette di gestire e migrare le versioni di RTEMS quando vengono rilasciate.
In questo manuale utilizzeremo come RTEMS tool suite prefix: $HOME/rtems-dev/rtems/5.

2 Installazione

2.1 Preparazione del host computer

Sistema operativo utilizzato - Ubuntu 18.04 LTS.E' preferibile utilizzare un sistema operativo Linux.
Requisiti del host computer: - Circa 1.5GB di spazio sull'harddisk per buildare tools, il kernel di RTEMS, network stack e package di terze parti per il BSP. - Ambiente di sviluppo C,C++,Python.
- Git.
Prima di tutto bisogna installare l'ambiente di sviluppo e Git, quindi apriamo il terminale e scriviamo i seguenti comandi: - sudo apt-get install build-essential
-sudo apt-get install git
-sudo apt-get install python-dev
-sudo apt-get build-dep binutils gcc g++ gdb unzip git
A questo punto si è pronti ad installare RTEMS.
Dobbiamo clonare 2 git repositories:
- il RSB (RTEMS Source Builder), che semplifica il processo di building per la toolchain di RTEMS

- il source code di RTMES, la tool suite e le sorgenti sono strettamente accoppiate.
Quindi i comandi sono i seguenti :
-mkdir -p $HOME/rtems-dev/src
-cd $HOME/rtems-dev/src
-git clone git://git.rtems.org/rtems-source-builder.git rsb
-git clone git://git.rtems.org/rtems.git
In $HOME/rtems-dev/src/rsb troviamo RSB e in
$HOME/rtems-dev/src/rtems abbimo il source code.
A questo punto possiamo buildare e installare la tool suite usando RSB.
-cd $HOME/rtems-dev/src/rsb/rtems
- ../source-builder/sb-set-builder –prefix=$HOME/rtems-dev/rtems/5
5/rtems-arm
Quando la build è completata possiamo controllare che il cross C compiler funzioni:
$HOME/rtems-dev/rtems/5/bin/arm-rtems5-gcc –version
Procediamo con il bootstrap dei kernel files, assicuriamoci prima che i comandi della toolchain siano nel variabile d'ambiente PATH:
export PATH=$HOME/quick-start/rtems/5/bin:"$PATH"
cd $HOME/quick-start/src/rtems
./rtems-bootstrap
Adesso possiamo configurare, buildare e installare le BSP di Raspberri Pi e Pi2:
- cd $HOME/rtems-dev/src/rsb/rtems
- $HOME/rtems-dev/src/rtems/configure
–prefix=$HOME/rtems-dev/rtems/5
–target=arm-rtems5
–enable-rtemsbsp="raspberrypi raspberrypi2"
–enable-tests=samples –enable-networking –enable-posix
- make
- make install

3 Test

RTEMS mette a disposizione degli esempi, il comando configure sopracitato ci permette di generare deei test programs. Li possiamo usare per controllare se l'ambiente di sviluppo è attivo e se possiamo runnare il nostro software su raspberry pi .
Generiamo le img per RPi1 e RPi2:
-arm-rtems5-objcopy -Obinary /Users/giorgio/quick-start/build/b-

raspberrypi/arm-rtems5/c/raspberrypi/testsuites/samples/ticker.exe
kernel-pi.img
- arm-rtems5-objcopy -Obinary /Users/giorgio/quick-start/build/b-
raspberrypi/arm-rtems5/c/raspberrypi2/testsuites/samples/ticker.exe
kernel-pi2.img
Adesso che abbiamo questi file , copiamo nel SD il firmware di Raspberry, rimuoviamo tutti i kernel*.img e facciamo un copy-paste del kernel crato con RTMES. Inoltre bisogna creare un file config.txt nella SD card che contiene:
enable_ uart=1
kernel_ address=0x200000
kernel=kernel-pi2.img/kernel-pi.img
si può non inserire l'opzione di kernel, e importare il kernel in RPi come "kernel.img" e in RPi2 come "kernel7.img".
Siamo pronti ad avviare il test, collegiamo il cavo UART e vediamo il risultato

4 Configurazione IDE per sviluppo di applicazioni per RTEMS

RTEMS ha a disposizione un plugin installabile su Eclipse C/C++ per sviluppare applicativi che utilizzano le librerie di RTMES, e per buildare gli eseguibili necessari per creare il kernel*.img della nostra applicazione.
Scarichiamo l'IDE di eclipse https://www.eclipse.org/downloads/.
Installiamo il plugin, andiamo su Help-¿Install New Softare e clicchiamo add. per inserire il link da dove prenderà il plugin
-ftp://ftp.rtems.org/pub/rtems/eclipse/updates/.
Selezioniamo RTEMS CDT support e installiamo.
Plug-in setup
andiamo su Window -¿ Preferences -¿ C/C++ -¿ RTEMS e settiamo i seguenti valori:
-Base path : il path dove è installata la toolchain di RTEMS
-BSP parth : il path dove è installato la BSP Makefile.inc
Creazione del primo progetto
Ci spostiamo sulla schemata C/C++ quindi Window -¿ Open Perspective -¿ Other -¿ C/C++ Perspective . Adesso creiamo un nuovo progetto C o C++ quindi File -¿ New -¿ C Project (o C++ Project) e seguire i passi.
Selezioniamo RTEMS Executablee e clicchiamo finish.
A questo punto abbiamo creato un progetto in cui la build creerà un file

.exe da cui sarà possibile ricavare il kernel file.
Fonti : Installazione di RTEMS
-https://docs.rtems.org/branches/master/user/start/index.html
-https://gist.github.com/giorgiobasile/1c1930a8a3ff8e36061cd7f4ef83da95
Installazione plugin
-https://devel.rtems.org/wiki/Developer/Eclipse/Plugin