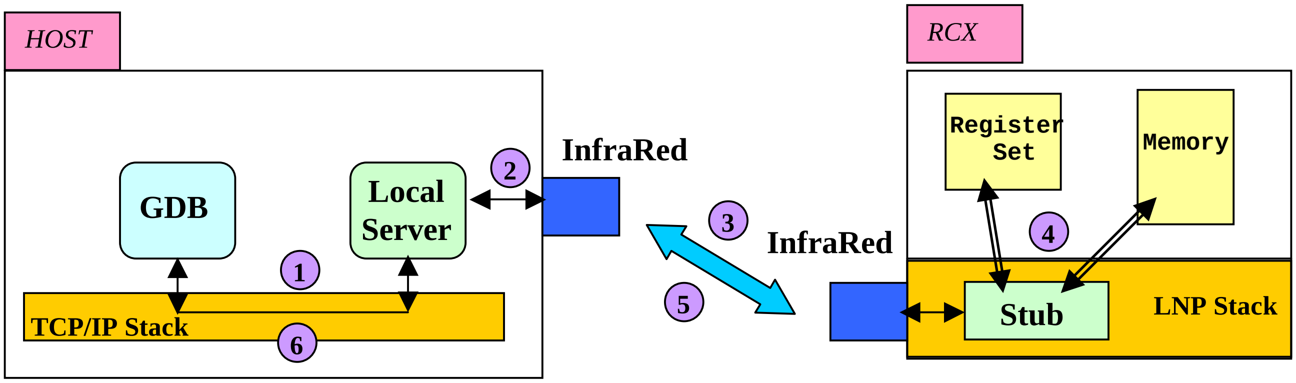
Remote Debugging of RCX (Hitachi H8/300 processor) with GDB

Project Members :
Gautam Gopinadhan (gvgopina@unity.ncsu.edu)
Palash Kasodhan (pmkasodh@unity.ncsu.edu)
Jaydeep Marathe (jpmarath@unity.ncsu.edu)

Project Website : http://www4.ncsu.edu/unity/users/g/gvgopina/www/rcx.html

Project Statement : To design and implement a remote debugging framework running on a Lego Mindstorm RCX brick running LegOS. The framework will enable the GNU debugger (gdb) running on a host machine to communicate with the remote RCX brick through IR communication and run debugging commands on the RCX.

Project Detailed Overview



1. GDB Connects to local server application through localhost tcp connection.
2. GDB sends command request to localserver, which relays it through IR to Stub on RCX.
3. Stub reads and decodes incoming request from server application on host.
4. Stub executes requested operation (register read, breakpoint insertion, etc.)
5. Stub returns completion code (failure/success or requested data) to server on
6. Server on Host returns completion code to GDB.

Current Status

- Preliminary framework designed by Les Smithson (http://www.hare.demon.co.uk/lego/gdb.html)
- Primitive local server (written in Python) and RCX stub applications exist.
- Support for programmed breakpoints, read and write to registers and memory exist.
- No support for online breakpoint insertion, instruction stepping etc.
- No support for freezing operating environment state machine (scheduler, other threads, motors, etc. when target is being debugged).

Tentative Project Goals

- Examine support for online breakpoint insertion.
- Involves understanding H 8/300 instruction set, to calculate effective address of next instruction (for inserting breakpoint).
- Convert local server to C/C++ to remove Python dependency.
- Freezing operating environment state (turning motors off, halting other threads temporarily, etc.) , for easier debugging.

Note : This is an open-ended project, due to the lack of documentation, as also due to possible deficiencies in the target processor instruction set (lack of explicit trap instruction, etc.). We hope to achieve as many of the tentative goals as possible, given the tight time constraints.

Relevant Resources

Project WebSite : http://www4.ncsu.edu/unity/users/g/gvgopina/www/rcx.html

- Previous work on remote debugging in RCX using gdb : http://www.hare.demon.co.uk/lego/gdb.html
- Description of Lego Network Protocol Daemon on Linux : http://legos.sourceforge.net/files/linux/LNPD/
- Description of GDB internals : http://www.cs.utah.edu/dept/old/texinfo/gdb/gdbint_toc.html
- Hitachi 8/300 Processor architecture and instruction set : http://www.hitachi-eu.com/hel/ecg/products/micro/pdf/h8_8bit.pdf