GEDAE™ Technical Document #1

GEDAE™ Board Support Package Development Kit

June 28, 2000

By Michael Both

GEDAE Board Support Package

This figure shows the software components used when running an embedded GEDAE application. The components with a white background are the ones that differ from vendor to vendor and must be written by the developer of the GEDAE BSP.

The software is constructed so a single host processor can communicate with all the embedded processors in an embedded system and the embedded processors can communicate directly with each other as long as the underlying software supports that communication.

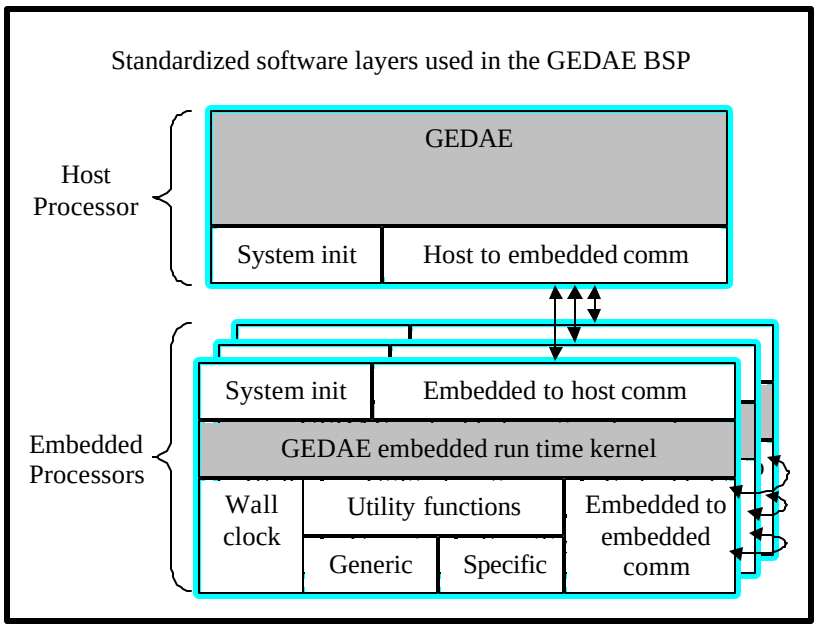


Figure 1

Role of the GEDAE Board Support Package (GEDAE BSP)

The main purpose of GEDAE™ is allowing users to easily build applications that are largely independent of target architecture. The GEDAE BSP hides vendor dependent details of an embedded system behind a uniform interface. Users need only a minimum understanding of the embedded hardware, but they can easily optimize the algorithm for that hardware. They do not need to know the details of how to handle the host to embedded communications, embedded to embedded communications, timing, and other components that make hand coding difficult.

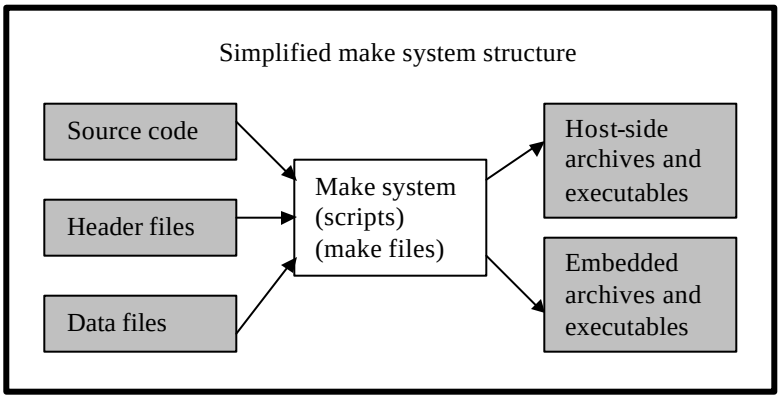
Role of the GEDAE BSP

This is a guide for how supported embedded systems are constructed to work with GEDAE, or a guide on how to build a GEDAE™ BSP to support a new embedded system.
The DSP Kit makes it easy to develop the required software components to allow the use of a particular embedded system. There are 3 parts to developing a GEDAE BSP.

1) **The make system**– This has to successfully create the host and embedded software components.
2) **The GEDAE BSP software components** – They add the functionality to allow:
   a) The host to start an embedded process.
   b) The host to communicate to the embedded process.
   c) Embedded processors to communicate with each other.
   d) The optimized vector library routines for the embedded processor.
3) **Testing of the GEDAE BSP** – The testing must verify that the make system and all the software components are working as defined by the GEDAE requirements.

The Make System

Figure 2



The GEDAE™ make system supplies the information for building executables that run on an embedded system.

First, the GEDAE™ host executable needs access to the embedded system. Board vendors usually provide a library with a set of C-callable functions that give access to the embedded system. This library must be linked into GEDAE™ along with the software layer that GEDAE™ places on top of the vendor's functions. This allows GEDAE™ to give the same look and feel for all embedded systems.

Second, GEDAE™ needs to be able to create executables that can run on the embedded system. If GEDAE™ is provided with the correct information to compile, archive, and link then this can be accomplished. The GEDAE™ embedded runtime kernel is provided to the BSP developer as source code, so it must be compiled and archived into a library. The vendor provides C-callable functions to access the boards specific capabilities (wallclock, communication, etc). The GEDAE™ make system must compile these into a software layer that is used by the GEDAE™ embedded kernel. As mentioned above, this allows GEDAE™ to give the same look and feel for all embedded systems.

Make System Requirements

The GEDAE make system must be expanded to compile, archive, and link files needed to allow GEDAE to be used with the embedded system.
The following is a list of items needed by the GEDAE make system.

a) The make utility (GNU make, or Sun make).
b) The embedded system compiler, archiver, and linker. These three utility functions are used to create object code, libraries, and executables.
c) The host system compiler, archiver, and linker.
d) Some simple make files and scripts that contain arguments and other information on how to use the host and embedded system compiler, archiver, and linker. Default versions of these files are provided as part of GEDAE™ BSP development kit. They are modified as necessary by the BSP developer.
e) The host and embedded system libraries, header files, and other utility files. The board vendor usually supplies these.

GEDAE™ provides the rules to compile, archive, and link code. These make files that are compatible with either Sun's make utility or GNU's make utility. The BSP developer uses Script files to specify the compiler, archiver, and linker options.

The BSP developer only needs to create or modify 5 files for describing the embedded make system, and 3 files for the host system. Templates for all of these files are provided and are small and simple.

The Software Components

Referring to figure 1, the following table describes the software modules in more detail.

COMPONENT	PROVIDE?	APPROXIMATE NUMBER OF FUNCTIONS	DESCRIPTION
System init (host)	No	6	Initialize system parameters, start an embedded process, kill an embedded process.
System init (embedded)	No	2	Start and exit a process.
Host to embedded communication (host)	No	10	Open and close control and data ports, send and receive data.
Host to embedded casting (host)	No	6	Only needed if the host and embedded formats are not the same (e.g. IEEE floats).
Embedded to host communication (embedded)	No	8	Opening and closing control and data ports, sending and receiving data.
GEDAE embedded run time kernel (embedded)	Yes, as C source code*	n/a	Download GEDAE application, execute the application, and collect and report tracing information to the host.
Wall clock (embedded)	No	1	Keep track of the elapsed time since the process started. It is used for execution tracing and helps with synchronizing some of the run time kernel features.
Generic utility functions (embedded)	No	7 system functions.	The system functions: processor id and memory allocation functions.
Vector library functions (embedded)	No	~200 functions	These functions provide a uniform interface to vendor specific optimized vector libraries. GEDAE™ provides a C code version of any function not in the vendor library.
Specific utility functions (embedded)	No	Decided by developer	These application utility functions are used to handle real-time devices, or any other system specific capability.
Embedded to embedded communications (host and embedded)	No	1 on the host. 8 for each embedded to embedded communications mechanism	5 of the embedded functions handle opening, closing, sending and receiving of data between embedded processors. 3 functions are used to synchronize transfer of data between processors.

* The GEDAE™ BSP Development Kit license limits modifications to correction of compiler warnings or errors.

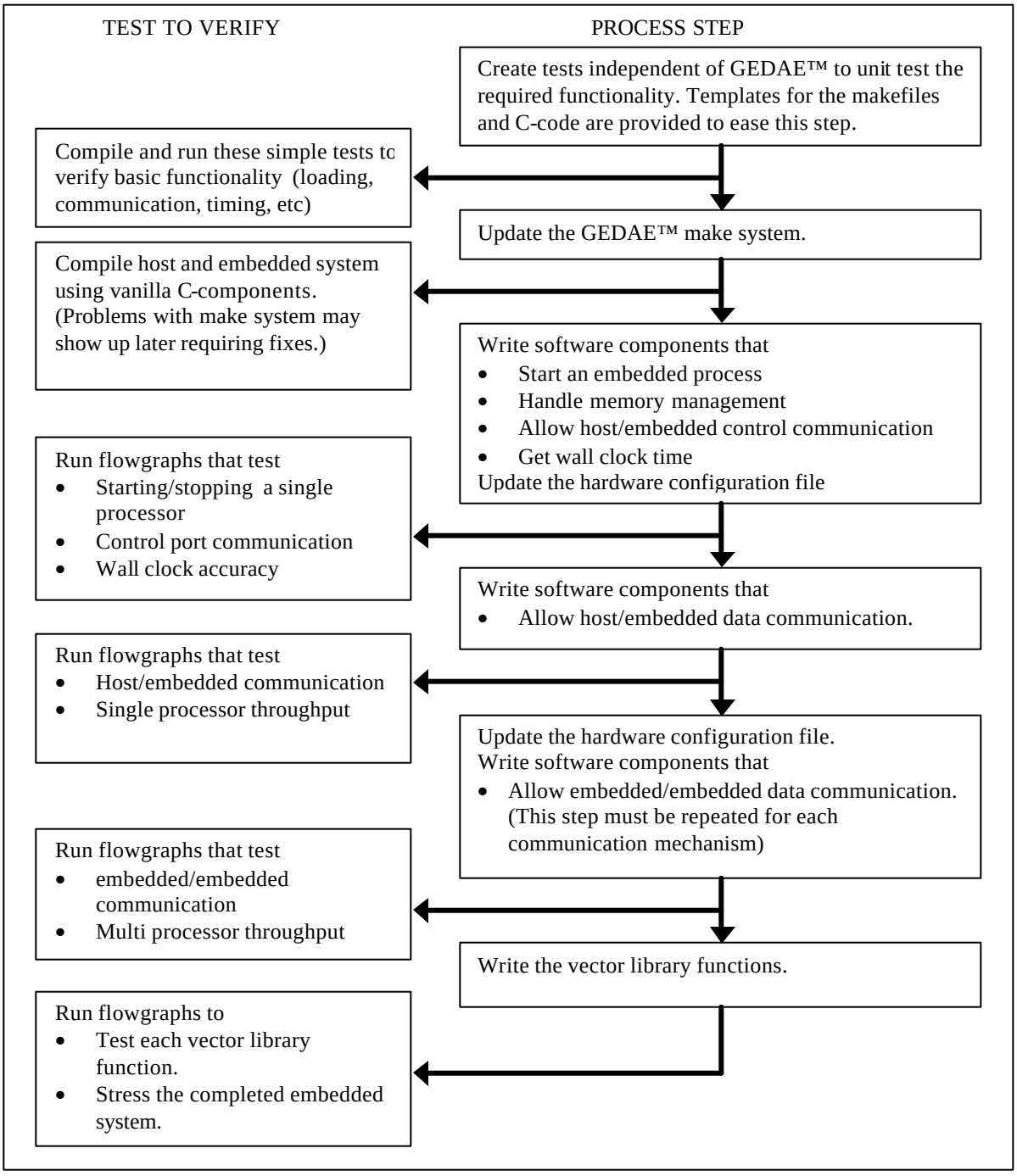
Purpose of the Software Components

The software components are created to link GEDAE and the embedded system together via a generic software layer. This allows GEDAE to communicate with all embedded system with a common look and feel

The utility functions should be optimized if possible to provide the most efficient execution possible. These are the functions that handle memory allocation, math functions (stream, vector, matrix, etc), DSP functions (fft, fir filter, etc), real-time device handlers, etc.

For single processor embedded systems, there is no need to write the ‘embedded to embedded communications’ component.

The Development Process



Except for the first step, all the software component testing will be done using GEDAE flowgraphs. The testing will be incremental, first by testing to see if a processor can start up, all the way to testing a multiprocessor system stressing it with as many features as possible at the same time.