



BEACON™

BEACON is a powerful software engineering tool suite focused on the design, implementation, test, and maintenance of high-integrity embedded systems, including those developed per RTCA/DO-178B. BEACON's graphical designer and multiple code generators work together to prevent many types of software errors from reaching the resultant code.

BEACON has been proven in the commercial aerospace industry with over a decade of use. Once every five seconds a plane takes off somewhere in the world with BEACON generated code running in the engine control systems. BEACON's automatic testing tool halts the migration of common software and design defects at the earliest point possible within the test phase, unit test. BEACON provides an overall lifecycle solution that is based on the design, not the code language or embedded processor, simplifying development and facilitating reuse.

Design

BEACON has several different types of diagrams that share a common data pool. The scope and visibility of each data element and data type is explicitly controlled by the designer.

Signal Flow Diagrams

- Dynamic filters - lead-lag, PID, notch, etc.
- Linear and nonlinear math blocks
- Multi-dimensional tables
- Boolean logic

Control Flow Diagrams

- Statements - pseudo-code or actual code
- Decisions - multiway branches, and loops
- Truth Tables

Data Windows

- Visibility and scope - local, body, external
- Data qualifiers - const, static, extern, etc.
- Fixed point scaling
- Minimum and maximum values
- User specified comments

Data Type Editors

- Visibility and scope - local, body, external
- Scalar types - unsigned int, #define, boolean, etc.
- Aggregate types - enum, array, record/struct
- User defined types and subtypes

Librarian

- Interfaces to legacy code or APIs
- Links to existing data and data types
- Generates exact function signatures
- Supports object-based designs

Diagram Parameter Forms

- Parent and child relationships
- Diagram update rates and offsets
- Parameter pass - by-val, by-ref, by global
- Parameter mode - in, out, in-out or state

BEACON is used to create a clear, traceable, and easy to review software specification using graphical highlights, code comment fields, and various print mechanisms.

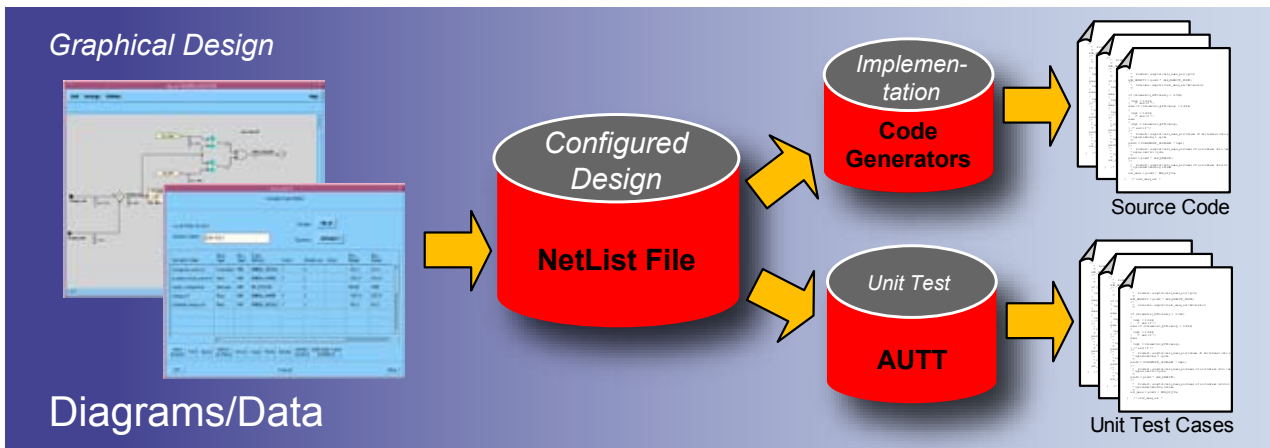
Annotation Options

- Diagram title block with code revision log
- Drawing palette - geometric objects and text
- Customizable print template - grids, logos, etc.
- Block, wire, and diagram comments

BEACON analyzes the design before generating code to ensure that it is explicit, consistent, secure, and that it satisfies safe programming practices.

Safety Checks

- Structured code - no gotos, single entry & exit
- Static memory - no recursion, no heap
- Strong typing
- Unambiguous and consistent data usage
- User specified loop breakers
- All control flow decisions have default paths
- Reduced aliasing of function parameters
- No reassignment of signal flow inputs



BEACON provides a highly-integrated software framework capable of generating and unit testing embedded software from design diagrams. The code executes on many platforms, including production ECUs.

Code Generation

BEACON generates efficient software that complies with programming best practices and standards. The code is compiler-independent and can execute on host platforms, real-time stations, or embedded microcontrollers.

Languages Supported

- ANSI-C
- Ada-83/95
- SPARK-83
- FORTRAN-77

Automatic Unit Test Tool (AUTT)

BEACON optionally creates unit test vectors, enhancing both the quality and reusability of the design and generated software. The test vectors target common error sources such as incorrect boolean logic, one-off comparison defects, and numerical overflows. AUTT provides coverages that are required by the strict RTCA/DO-178B guidelines used by the FAA.

Structural Coverages

- Statement coverage
- Decision coverage
- Modified condition/decision coverage
- Structured path coverage

Numerical Coverages

- Boundary value analysis
- Variable range stressing coverage
- Numerical value stressing coverage
- Numerical table data access coverage

Additional Coverages

- Hierarchical interface coverage
- Operand identity coverage
- Absolute value domain coverage
- Input value robustness testing

AUTT also performs additional analysis and computations, yielding a variety of metrics, warnings, and errors. This output can be generated prior to code generation because AUTT is based on the design, not the code.

Analysis Outputs

- Calculation of design ranges
- Notification of numerical overflow and underflow
- Identification of dead or deactivated code
- Generation of expected results
- Computation of various complexity measures

Platform Support

BEACON executes on Sun workstations running Solaris.