Beacon Firmware Tool Usage

The tools are launched by batch files using one of the following commands.

```
  program_beacon [options] <firmware.hex> <parameters.csv>
  update_beacon  [options] <firmware.hex> <parameters.csv>
  update_beacon  [options] <firmware.hex>
  update_beacon  [options] <parameters.csv>

Options (defaults shown in parentheses):
    -r, --[no-]run                   Run firmware after programming (on)
    -i, --[no-]interaction           prompt for device insertion (on)

Common Options:
    -v, --[no-]verbose               Run verbosely (on)
    -d, --debug                      Turn on debug mode (off)
    -h, --help                       Display this help message
```

The `program_beacon` command can be used even with devices that are not programmed. It erases and overwrites the flash without considering what it might have previously contained.

The first form of the `update_beacon` command upgrades (or downgrades) the firmware of an existing Beacon device. It also lets you alter some or all of the parameter values.

The second form of `update_beacon` does a firmware upgrade/downgrade only. All original firmware parameter values are preserved in the updated beacon, if possible.

The third form of `update_beacon` modifies only the parameter values specified in the CSV file. The rest of the parameter values and the firmware itself are not altered.

Command-line Arguments

When both an Intel Hex file and CSV file are given, the order does not matter but they do have to use ".hex" or ".csv" file extensions. (This is not case sensitive.)

<firmware.hex>

this can be the name of any file in the current working directory or it may use a relative or absolute path to any location. The data must be in Intel Hex file format.

<parameters.csv>
: The CSV file can be any file in the current working directory or use a relative or absolute path to any locaton.

Command-line Options

-r, --[no-]run
: By default, each time a device is programmed the programmer will release it's TM line so the device can start running immediately. Use the --no-run option to prevent the device from running.

-i, --[no-]interaction
: By default, the beacon tools assume they are interacting with a human operator and will provide prompts for device insertion. In production-line environments, however, when these tools may be run by a scripts that do not recognize or respond to prompts, use of --no-interaction can be used to suppress prompting.

-v, --[no-]verbose
: Verbose mode is on by default. Use --no-verbose to reduce the number of messages displayed on the console.

-d, --debug
: Debug mode is off by default. Using --debug will generate quantities of console messages---mostly data structure dumps---that may be helpful in solving problems

Matching Up Firmware Parameters

There are up to three sources of parameter values: existing firmware in the device, default values in the new firmware image and any values specified in the CSV file. The following rules sort out what happens.

1. If the CSV file contains a value, that one is used; if not,
2. The value in the existing device firmware is used, if defined and, finally,
3. The value contained in the new firmware will be used as a default.

However, if the new firmware does not use a parameter that is given elsewhere, the tools will

- abort execution if it is specified in the CSV file *or*
- ignore it if it is found in the existing device firmware.

As implied in the earlier discussion, the rule 2 is ignored by the `program_beacon` command.

The parameter names used for matching are defined by the Beacon Firmware tools, not by the firmware source code. In almost all cases, however, the names are the same to avoid confusion. In at least one case, however, a parameter name was changed in the firmware, but the values were still used for essentially the same purpose. To preserve compatibility across different firmware versions, the tools use the original parameter name.

See Also

The README files and the ParameterDictionary files (both as .md and .pdf) in this directory.