

Code selection for placing in the dongle

This is the most responsible and nontrivial stage of loadable code development. It essentially depends on the fact that the developer needs to decide on what code needs to be executed in the dongle.

The code needs to be designed so that a certain task will be executed in the following fashion:

- Receiving buffer with input data
- Calculations using data from this buffer
- Return of buffer with output data

There is a range of requirements to this code imposing rather serious limitations on the choice. The requirements can be divided into several types:

Security requirements

The loadable code needs to be rather complex, so that brute force or other (more effective and advanced) methods of black box analysis would not make the creation of an emulator in a short time possible. This code needs to be missing in the earlier versions of the application. Not following this condition makes it possible to compare the application versions and finding the transferred code for embedding it into the emulator.

Performance requirements

The dongle's controller features rather high calculation capacity, however it is still lower than that of the modern processors. Therefore, it is important to avoid making the code too resource-intensive, otherwise the time of its execution may be increased to unacceptable level. Besides, the code loadable into the dongle should not be called very often, for instance, per cycle. If the delay may seem to be insignificant with one call, it may become substantial in a cyclic call.

Implementation requirements

The code placed into the dongle must not:

- Contain API calls, which cannot be relocated into a dongle or have other external dependencies
- Use input-output streams
- Use console output
- Use dynamic memory distribution

The code executed inside the dongle has only standard C libraries and dongle functions available to it.

Guardant Code dongles allow executing algorithms of up to 20 thousand lines of C code (up to 60 thousand lines in the models with expanded memory). Consequently, the code size needs to be within these limitations.