Simplify code portability and maximize return on investment

WHITE PAPER

VERSION 1.0

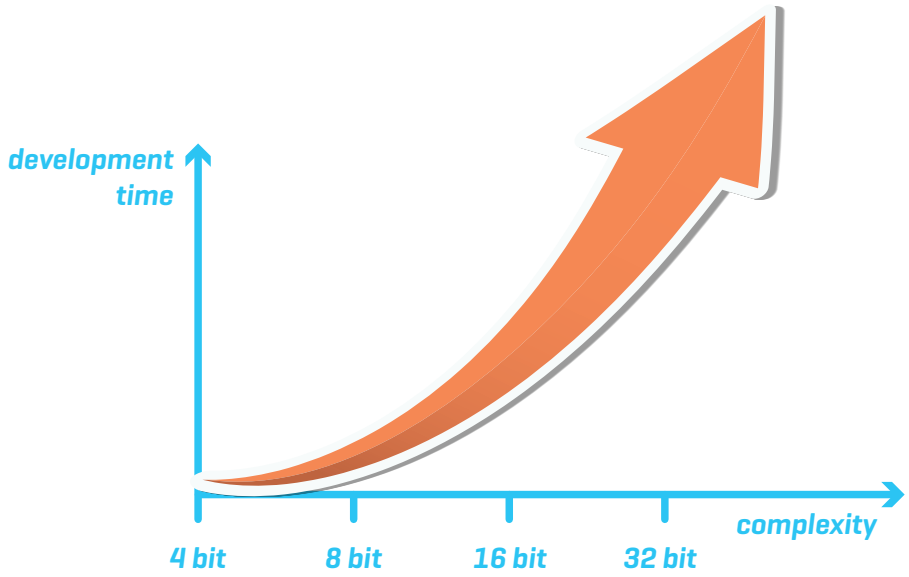
December, 2017.

MikroElektronika
DEVELOPMENT TOOLS | COMPILERS | BOOKS

THE TECHNOLOGICAL ADVANCEMENT AND THE CODE

Technological advancement brings new challenges for the software developers

The technology of making integrated circuits is advancing very fast. Day by day, new integrated devices and technologies are developed and produced, exponentially more complex than previous series. Accordingly, the application code becomes so complex that it is nearly impossible to add new features, track changes and generally – have a control over the code itself. Once written, the application code can work only on a specific system and with the specific MCU it was written for. Adding a new feature requires complete redevelopment of the code and as a result – increases the time needed and the cost of the whole manufacturing process.



Software tools that deliver solutions, are in demand

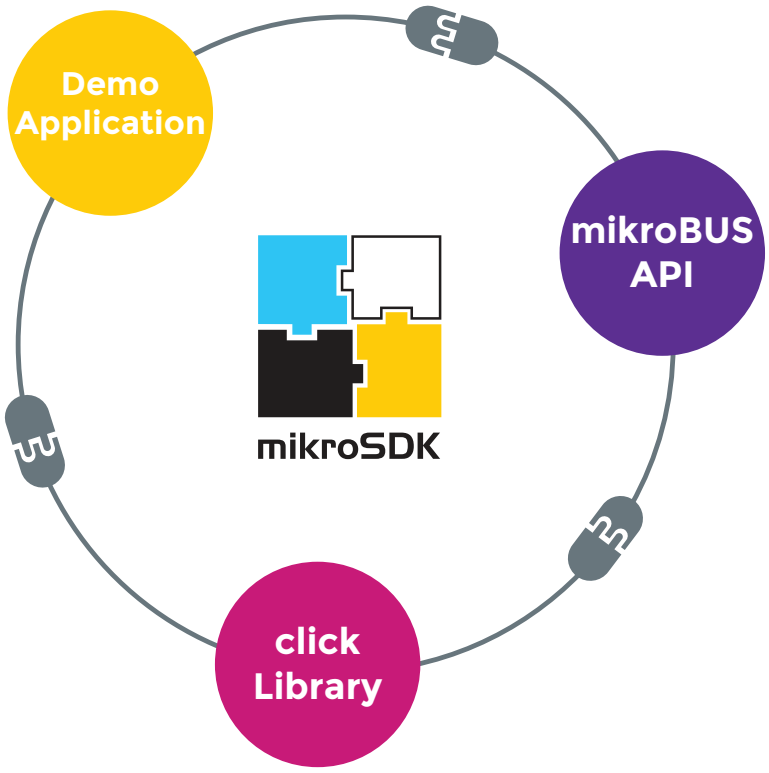
There is an increased demand for the software tools that can cope with the rapid hardware development. The need to unleash the full potential of the embedded hardware is bigger than ever before. The frontier of what is technologically possible is always pushed forward and the software development tools need not restrain or look for compromises – they have to offer solutions and possibilities, instead.

mikroSDK

TOOLS FOR THE PORTABLE CODE DEVELOPMENT

The software engineers at MikroElektronika came up with the solution for this problem in a form of the mikroSDK ***standard*** which prescribes a set of coding rules, and the mikroSDK - the ***software development kit*** as a set of libraries and standardized function calls used to write the portable application code itself.

Making the application code mikroSDK standard ***compliant*** will allow for the application code portability among different platforms, as well as the code reusability.



The mikroSDK uses a modular approach to the portability problem – all the code that is architecture dependent is placed in separate layers. All the code that is specific to the controlled device is also placed in separate layers. Finally, all the hardware specific configurations and functions are located in their separate layers.

Modularity, to cope with the complexity

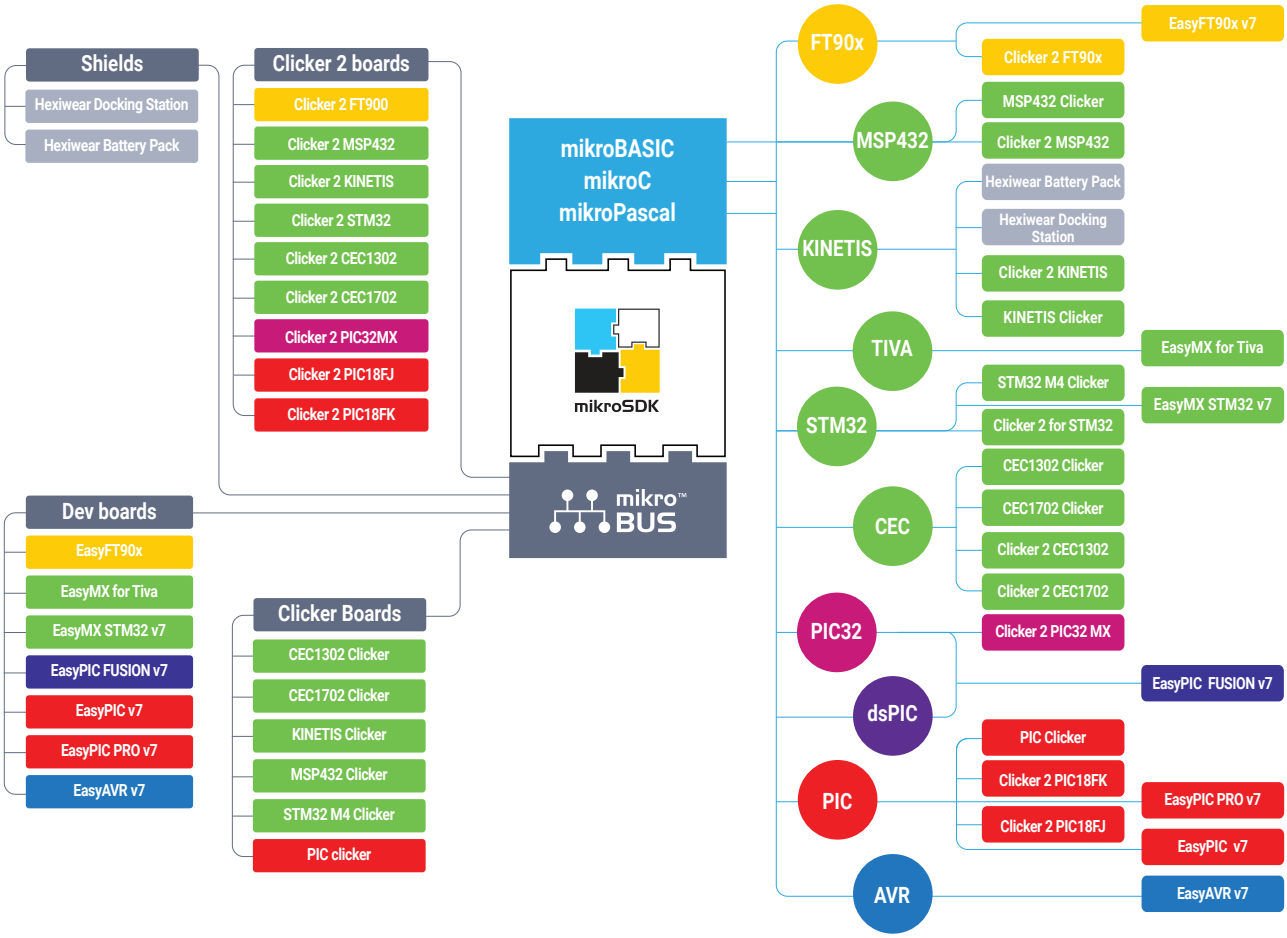
This allows the hardware platform to be switched, along with the corresponding layers – the application code will continue to run unmodified since it will get interfaced with the new hardware layer after a single recompiling for the selected architecture.

The story goes the other way around, too. Once written for the specific hardware platform, the same hardware abstraction layer and the board definition files can be used with any application code, written for a specific device, or a click board. For example, whenever a new click board is used, the existing hardware layer will get interfaced with the driver layer of the new click board, and after a single compiling for the selected architecture, the click board will be ready to be used with no new code written at all.

Focus on the application code itself. Write it once, use it everywhere

MikroElektronika provides board definition files and hardware abstraction layer functions for own branded development systems, along with the demo application code – we have it all covered up for the end users and developers.

Since the mikroSDK is provided in a form of an open source, it can be modified, tweaked and expanded with new functionalities.



NOTE

This tree diagram represents the platforms currently supported by the mikroSDK. The tree will be expanded as the new platforms are added.

STANDARDIZED CODE AND LIBRARIES

To achieve the consistency and functionality of the application code, certain coding rules, defined by the mikroSDK standard, have to be followed. By following those rules, the application code will be microSDK standard compliant.

Clean and simple. Retain control over the code

The coding rules will ensure that the code will be portable and reusable, even with the future updates. Set of logical and easy to remember rules that make the code clutter-free, clean and readable, also allow for expandability on different systems and compilers - provided that the necessary low-level parts of the mikroSDK are written for those platforms. This way, the mikroSDK can find its way even among other companies and developers, expanding the ever–growing library of click boards™ to the variety of systems and architectures.

PERFORMANCE IMPROVEMENTS AND FUTURE UPGRADES

The applications which are written to be mikroSDK standard compliant will have a small memory footprint compared to the number of supported peripherals. The mikroSDK allows for the optimization of the code by removing the unused code sections from the compiling task. This results in smaller final code, further optimized by the smart SAA algorithms used inside the MikroElektronika compilers. The advantages of the smaller code are obvious – there is a room for even more features in the application code.

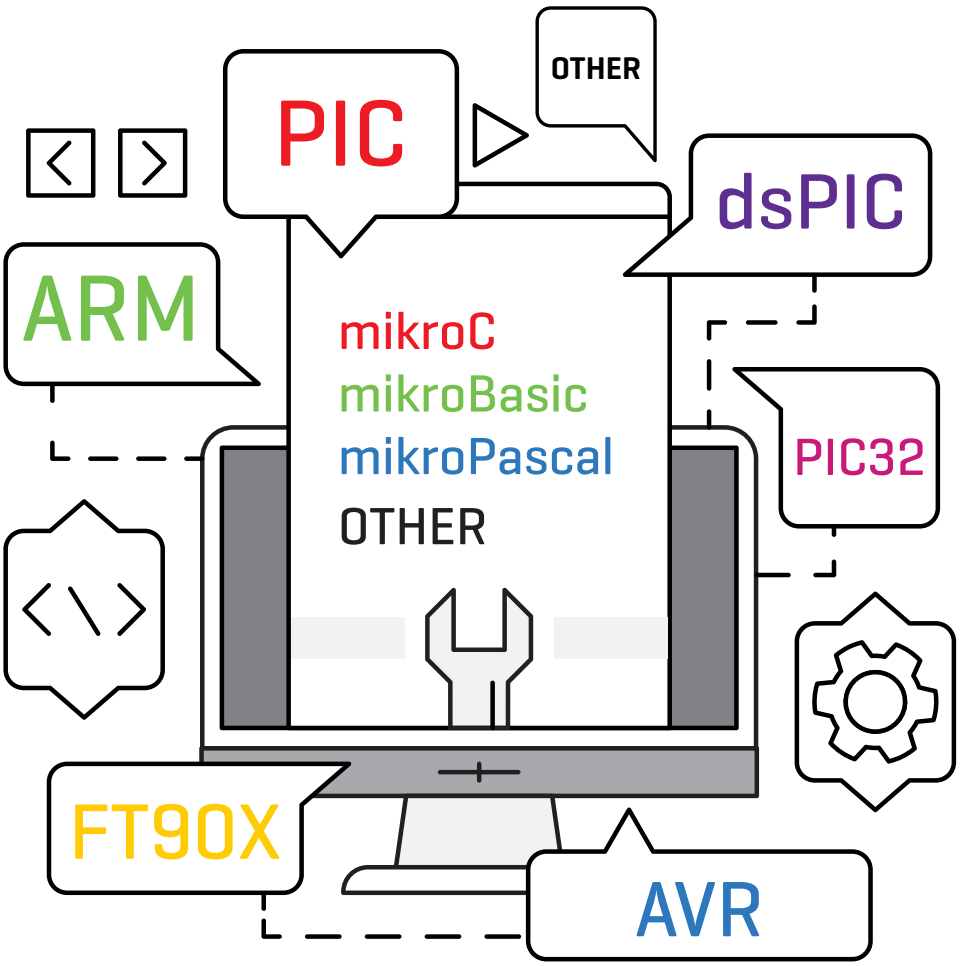
mikroSDK is created with care

The mikroSDK is designed with care. It allows for a rapid development, it is easy to understand and use, and it is very well documented. The standard strictly defines folder structures, documentation, and the application code format, preventing any ambiguous situations. The mikroSDK is easily expanded and leaves room for future improvements – the backward compatibility is always maintained because the mikroSDK is highly modular.

CONCLUSION

The mikroSDK allows for the rapid software development. The developers do not need to worry about the low-level part of the code, they are able to focus on the application code itself. This means that changing the MCU or even the whole platform will not force developers to redevelop their code for the new MCU or the platform. They will be able

to just switch to the desired platform, apply the correct board definition file - and the application code will continue to run after a single compiling. The saved time can now finally be invested in the next big project, allowing for faster product delivery and faster return on the investment. Be it PIC or PIC32 or STM - for the user application code - it all just the same.



mikroSDK

mikroe.com/mikrosdk

mikroe.com/mikrobus

libstock.mikroe.com