Embedded Programming and C++

This lesson will talk about author's perspective on Embedded Programming with Modern C++

WE'LL COVER THE FOLLOWING

The inventor of C++, Bjarne Stroustrup, FAQ makes it crystal clear that one of the design goals for C++11 was to make C++ even better for the embedded world.

According to Bjarne Stroustrup himself,

> "Improved performance and ability to work directly with hardware – make C++ even better for embedded systems programming and high-performance computation."

As a programmer, I resonate a lot with this statement, since I changed from software development in the middleware area into the embedded world. I began my search for the features in C++11 that are well suited for the embedded programming. I presented the results of my search at conferences in Berlin, Munich, or Sindelfingen.

My perception of the new features of Modern C++, such as user-defined literals, the type-traits library, or constant expressions has changed a lot. They are such great features that they belong in the toolbox of each professional C++ developer. This statement is even more true for the embedded world because they have to deal with stronger requirements. Therefore, this course is particularly valuable in embedded programming.

What's the Reason for my Perception?

What's the Reason for my Perception?

The answer is simple as there is no typical embedded system and the diversity is large. The diversity is large. There are RFID transponders about the size of a few millimeters, pacemakers that should work reliably for a longer period, defibrillators consisting of a few boards and cars having more than 100 electronic control units (ECUs). The borders between the embedded systems and other systems as game development or low-level systems are blurred.

What Makes Embedded Programming Special?

Embedded programming is more intensive than other forms of programming because it must handle specific requirements:

- High-performance requirements up to real-time requirements
- Safety-critical systems
- Reduced resources as memory and CPU power
- More Tasks that should be accomplished in parallel

I had a long fight on how to present the features in a structured way. In addition, the features are often dependent on the C++ standard. Here is my plan. The features are the answers to typical requirements in the embedded programming. These requirements are the key points of my structure.

- High safety requirements
- Performance matters
- Careful handling of resources
- Doing more jobs in parallel

Additionally, I will provide the information about the C++ standard which has these features.

Before we start with the lessons about embedded programming, we have to make a short detour. In my profession as a software developer, I, Rainer Grimm, heard a lot of myths about C++. I will present them in the next lesson and provide the facts in the lesson after the next.