Programming a small micro-controller is different from programming a desktop or other large system: the programmer must constantly be aware of the limited resources (especially ROM and RAM) he has available. Traditional C++ programming tools and practices are tuned to systems that have (compared to a small micro-controller) practically unlimited memory. This has led to a widespread view that C++ is not suitable to programming small micro-controllers, and to the status-quo of C being the dominant programming language in this field, despite the advantages C++ has to offer, when used appropriately.

In this talk and the accompanying workshop we will explore both aspects: what has C++ to offer in a field where C dominates (and is often regarded as totally adequate: "We don't need the no bells and whistles of a more complex language."), and which practices and tools from the traditional C++ world use are useable for small micro-controllers.

An important task of many micro-controller applications is the manipulation of hardware external to the micro-controller via its GPIO (General Purpose IO) pins. We will explore this aspect using a target board, a solderless breadboard, and some simple hardware components, focusing on how the abstraction mechanisms provided by C++ can help writing such software in a way that supports better re-use than the traditional C approach.

The participant must bring his or her laptop, which must run a (preferably modern) windows version.

The software tools and example code will be provided in advance, and participants are strongly encouraged to install and test these. The hardware set will be provided at the workshop day.