

Preface

To be able to plan and write this type of book, you need a good work environment. In my case, I was able to benefit from the best working conditions for this enterprise. In terms of infrastructure and material, the *Institut National de Sciences Appliquées de Toulouse*, France (Toulouse National Institute of Applied Sciences), and in particular their Electrical and Computer Engineering Department, has never hesitated to invest in computer systems engineering, so that the training of our future engineers will always be able to keep up with rapid technological change. I express my profound gratitude to this institution. These systems would not have amounted to much unless, over the years, there was an educational and technical team bringing their enthusiasm and dynamism to implement them. The following pages also contain the hard work of Pascal Acco, Guillaume Auriol, Pierre-Emmanuel Hladik, Didier Le Botlan, José Martin, Sébastien Di Mercurio and Thierry Rocacher. I thank them sincerely. Two final respectful and friendly nods go to François Pompignac and Bernard Fauré who, before retirement, did much work to fertilize this now thriving land.

When writing a book on the assembly language of a μ processor, we know in advance that it will not register in posterity. By its very nature, an assembly language has the same life expectancy as the processor it supports – perhaps 20 years at best. What's more, this type of programming is obviously not used for the development of software projects and so is of little consequence.

Assembly language programming is, however, an indispensable step in understanding the internal functioning of a μ processor. This is why it is still widely taught in industrial computer training, and particularly in training engineers. It is clear that a good theoretical knowledge of a particular assembly language, combined with a practical training phase, allows for easier learning of other programming languages, whether they are the assembly languages of other processors or high-level languages.

Thus, this book intends to *dissect* programming in the assembly language of a μ controller constructed around an ARM Cortex-M3 core. The choice of this μ controller rests on the desire to explain:

- a 32-bit processor: the choice of the ARM designer is essential in the 32-bit world. This type of processor occupies, for example, 95% of the market in the domain of mobile telephony;

- a processor of recent conception and architecture: the first licenses for Cortex-M3 are dated October 2004 and those for STMicroelectronics' 32-bit flash microcontrollers (STM32) were given in June 2007;

- a processor adapted to the embedded world, based on the observation that 80% of software development activity involves embedded systems.

This book had been written to be as generic as possible. It is certainly based on the architecture and instruction set of Cortex-M3, but with the intention of explaining the basic mechanisms of assembly language programming. In this way we can use systematically modular programming to show how basic algorithmic structures can be programmed in assembly language. This book also presents many illustrative examples, meaning it is also practical.