1 Introduction

This book has been written to guide Electronic engineers in designing and developing their own custom embedded systems using STM32 Cortex-M4F series of Microcontrollers.

The book introduces concepts related to microcontrollers, Embedded Systems, EDA Tools, Firmware Development, Tool Chain etc.

Finally the book ends with complete details of a generic development board with most of the features of the STM32 microcontroller implemented along with a generic library and sample codes. This information can be used as reference by readers to design their own custom systems and contribute towards the progress of electronics and embedded systems.

2 Embedded System

Term Embedded system implies to a system (Electronics and Firmware) which is embedded into the larger system for achieving a specific functionality.

Modern embedded system market is overwhelmed with numerous microcontrollers and processors, each having it's own pros and cons. However STM32 Cortex-M4F series of microcontrollers dominate this market due to easily available free electronic and firmware development tools. All this helps in promoting, guiding and providing wings to the ideas of young engineers.

2.1 EDA Tools

Electronic Design Assistance (EDA) tools are a set of software applications which makes understanding, exploration, designing and testing of embedded systems possible.

The importance of EDA tools in the proliferation of embedded systems is quite visible from the market share of 32-bit micro controllers being sold world over by different companies. Free availability and comprehensive tool chain may motivate an engineer to select a specific microcontroller during his initial stage of engineering journey.

Therefore companies spend a lot in marketing their microcontroller through availability of Development Boards, Integrated Development Environment, programming-debugging tools and downloading tools.

2.2 Electronics Development

Electronic development implies designing and development board with required electronic components for achieving desired functionality. The activities involved in design and development of electronic hardware are listed below:

2.2.1 Electronic Design

The system to be designed needs to go through feasibility study to ensure success of the project. This includes selection of correct components, availability of components (lead time), cost of components, type of components (commercial, military, automotive) and qualification/certification requirements.

The information and artifacts during design and development process are all generated using EDA tools in common standard formats to ensure generation, storage, search, retrieval, update, modification and version control.

Following this the major activities involved in the process of Electronic hardware design & development:

2.2.1.1 Schematics

Schematic or Electrical Connection Diagram is the first stage of designing electronic hardware for embedded systems. This presents the interconnection of components to achieve the required functionality.

2.2.1.2 Gerber Files

Gerber file contains information related to placement of components on PCB, tracks on PCB for interconnections, number of PCB layers, dimensions and quality of PCB.

2.2.1.3 BoM

Bill of Material (BoM) presents information related to component part number, quantity, Manufacturer, cost etc.

2.2.2 3D PCB file

This file includes dimension of PCB, Mounting holes, height of components mounted and connectors. This file is required to be shared with mechanical team designing enclosure for the electronic hardware.

2.2.3 Power Supply Circuit

This is the most important circuit for any embedded system irrespective of it's complexity and usage. Power is what that makes electronics, do its magic.

The power requirements of the electronic system should be well calculated (after 50% de-rating) to ensure reliable system functioning in all the dynamic usage conditions.

2.2.4 Programming Circuit

Another important circuit for any electronic system based on microcontroller or processor is it's programming cum downloading circuit. This helps in burning the firmware logic inside the memory of the micro controller which is mounted on the PCB.

2.2.5 Reset Circuit

Reset Circuit helps the operator to restart the system. It may sometimes be used to bring the system out of hang situation (due to firmware bug).

2.3 Firmware Development

Firmware development involves designing, development, IV&V, deployment, version control and maintenance. The firmware development stages are explained below:

2.3.1 Firmware Design

Firmware design implies listing out functionality (functional requirements), development of flowchart (algorithm) to finalize what is to be done by firmware and in what sequence.

2.3.2 Firmware Development

Firmware development refers to writing and compilation of the code.

2.3.3 Firmware IV&V

Independent verification and validation (IV&V) implies testing of the code by third party for ensuring finding and removal of all firmware bugs before release.

2.3.4 Firmware Deployment

Firmware deployment refers to copying, downloading or installing of the firmware onto the electronic board. This also includes deployment of associated libraries.

2.3.5 Firmware Version Control

Version control is very important to ensure that changes in code are well documented and controlled to maintain the firmware at all stages of lifecycle.

2.3.6 Firmware Maintenance

Maintenance implies making modifications in code to remove bugs and enhance functionality.

3 SENSORTAG

Sensortag is a portable miniature matchstick size embedded system to measure, record and replay various environmental parameters as per user requirement.

The process of designing and development of an embedded system is explained next through by taking example of sensortag.

3.1 Sensortag

Sensortag_Board_Image

3.2 Schematic

Sensortag_Schematic

3.3 Gerber File

Sensortag_PCB_File

3.4 Code Examples

3.4.1 IO Application

3.4.2 UART Application

3.4.3 I2C Application

3.4.4 Timer Application

3.4.5 SPI Application

3.4.6 ADC Application

3.4.7 DMA Application

3.4.8 RTC Application