

A General Framework for Developing Embedded Controller Applications

Bharati Bhangale¹, Pramod Patil²

Associate Professor, Department of Physics, A.S.C College Ozar Mig, Dist. Nasik India¹

Managing Director, AOH Consultant, Pune , India²

ABSTRACT : A General Framework for development of embedded Controller (8/16/32bit) application for Industrial Product design is presented including Hardware ,software , firmware development . A focus is given on sensor interfaces , flash Memory for boot block and application storage ,power electronics interfacing. Also PCB design discussed in brief. In software section various compilers and bootblock programming explained for illustrating applications such as settop box and motor control application is taken as reference. Throughout the paper, a special attention is given to the use of framework in the electronics/software engineering education process. Besides giving some suggestions for the use at different educational levels, the emphasis is on the implication of contemporary design methods, and the hardware's robust design. Describing the software part of the framework, programming and code development tools explained. The stress is given to the use of high-level tools, where the algorithms are described in the form of different graphical notations, i.e. block diagrams, instead in C or assembly code.

KEYWORDS: 80C51 ,PIC ,8/16/32 bit ,memory , Development system, sensor, compiler.

I. INTRODUCTION

A range of gadgets (smart phones) , equipment , appliances ,cars , medical devices like pace maker we use in our daily life are embedded with microcontrollers and there is no place where microcontrollers not have application. The central part of any embedded system [3] is based on microcontroller (μ C) or Digital Signal processor (DSP).Microcontrollers [1] are special microcomputers integrated on a single chip which consist of central processing unit , memory (RAM and flash type) and peripheral devices for connecting analogue world (sensor , interfaces to mechanical devices) such as Analog to Digital Converters (ADC) and Digital to Analog converters (DAC) , Timers and counters , PWM processing units , Serial interfaces like RS232, CAN , MOD Bus ,I²C , eathernets etc. The software plays crucial part in microcontrollers design and updates requires time to time without changing hardware and updation process can be performed from central place without visiting any gadgets , thanks to internet technology . If gadget is connected to Internet it can be updated with latest software without wire connection for. Eg. Wi fi , GSM link etc.[5]

The popular 8/16/32 microcontrollers are from ATMEL,PIC,INFINEON, INTEL ,TI ,Silicon Logics [27] , ST microelectronics etc.[15] To popularize their product for development each manufacture had provided design references , application notes and sample designed proto board . Software is developed in C/C++ compilers for custom built applications , but RTOS [11] are now popular and Java-Eclipse base software with J-Tag (It is cable to download firmware in to system flash memory on chip) are very popular. There is rapid changes and developments in Semiconductor world¹⁰ and application development cycle is reducing and to keep educational system at par with Industrial need a Framework is needed to ease and quicken development process. Hardware is developed in already proven circuitry to avoid additional inventory of components , PSPICE modelling and finally Printed Circuit board which is the house of application is developed with PCB design software's like Eagle , Orcad , Design spark etc. Some software are free for limited size of PCB / period of time. The main hardware parts are the main board with the attached microcontroller and the programmer used to download the firmware from a personal computer to a microcontroller. The framework's software tools can be divided into those for program code development, and those for programming the microcontrollers. In the latter group only freely available programs were considered, while in the

case of development tools we focus on those offering high-level [23] design, with the algorithms described in the form of different graphical representation (e.g. block diagrams, flowcharts) rather than written in C or assembly code. In this paper we will discuss the step by step process of developing hardware and software.

II. EMBEDDED SYSTEM DESIGN CYCLE

In current scenario of very fast development in semiconductor technologies, microcontroller system designers are facing ever-increasing challenges for their product to be in market competitive and time to reach in market and getting obsoleted very soon. In this context there is need to use systematic design approach, methods rather than trade off technical, cost and time-to-market feasibility factors. The design process models[10] are

- The Waterfall model is the most rigid one, suggesting to move to a phase only when its preceding phase is completed and perfected. Phases of development in the waterfall model are kept very separated, and there is no room for iteration or overlap.
- The V model has the same strict serial structure as the waterfall model, but it suggests that, before going to a more detailed design level, one should already test all the system features and properties that can be tested at the current level of design abstraction.
- The Incremental Model allows multiple iterations in some of the design phases, resulting in a *multi-waterfall* process.
- The Spiral Model is similar to the incremental model, with more emphases placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model).
- Model-Driven Engineering is an emerging design process, that improves on the V-Model by supporting the test phases at each design level by software models that simulate the system before real implementations exist already.

We are using V model for our Embedded controller system, in this model general progress of design steps in time domain is shown from left to right and therefore horizontal axis can be taken as time. The vertical axis represents system level components from High level abstraction to very very low level actualisation process.

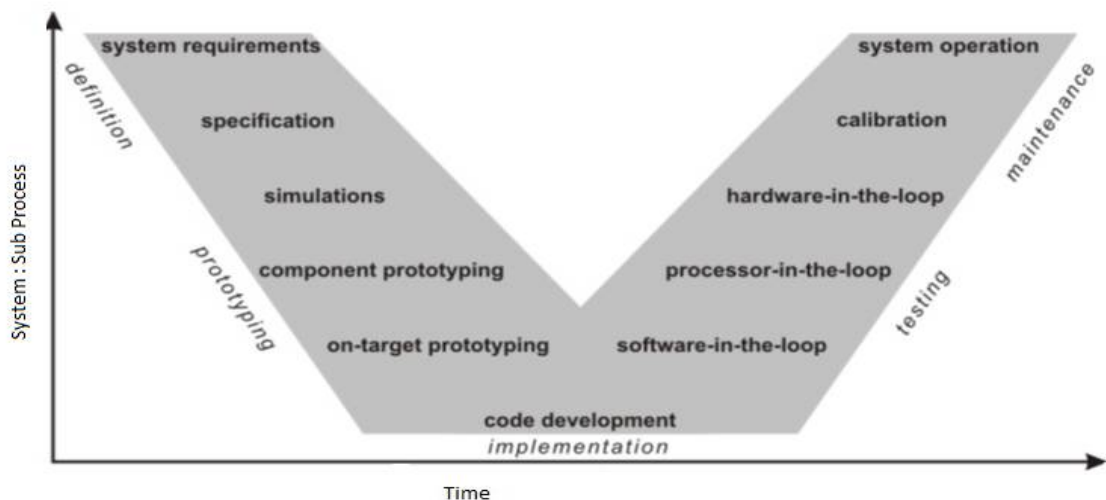


Fig: 1 V Model

The design starts with specifying the system requirements. These are then processed, around down ramp of V shape in to sub system components (split into smaller pieces). At the bottom of the V-diagram is the implementation stage, representing the transition from decomposition back to re-composition proceeding up along the right leg (ramp up) of the V-diagram. Here the system’s entities are combined back into larger pieces, resulting in a finally assembled system. The framework presented in this paper aids narrowing the embedded system design V-diagram by providing a common set of tools that are used through the prototyping, implementation and testing stages. In prototyping stage first a simulation model is designed that behaves according to the requirements. Then for the validation of feasibility, the design components are prototyped on general prototyping hardware. The timing characteristics are not modelled in this step, and may change in the actual design. Once the components behave appropriately, the on-target prototyping is employed on the envisaged processing device, in our case a microcontroller. Characteristics of code compiler, fixed-point precision effects and interactions among the system’s components are validated in this step. When all simulations and prototyping shows the correct operation of a design, a highly optimized code is produced in the implementation stage. Finally, in the testing stage the design is evaluated while moving up along the right leg of the V-diagram. Configurations such as software-in-the-loop, processor-in-the-loop, and hardware-in-the-loop are used in the proceeding steps[8].

III. EMBEDDED CONTROL DEVELOPMENT SYSTEM-PLATFORM

There are lots of commercially available platforms for development for Industrial products such as proto development boards (MSP 430-MSP Launch Pad , Silicon Logics- Thunder boardetc.) along with C compilers and reference design application notes software for basic interfacing. Engineering colleges and University and Industrial R & D centres use Micro controller kits for shortening time for development training. There are simulators and debuggers from Keil , IAR and eclipse base software is now available at affordable prices. The development board has flash memory which had divided in 3 parts [a] Boot Block : In which BIOS is written for serially downloading program from Development platform (IDE, Compilers /RTOS). [b] Parameter Block : Here fixed parameters used in application software are stored. [c] Application software Block : Here resides the software written for specific application or Product.

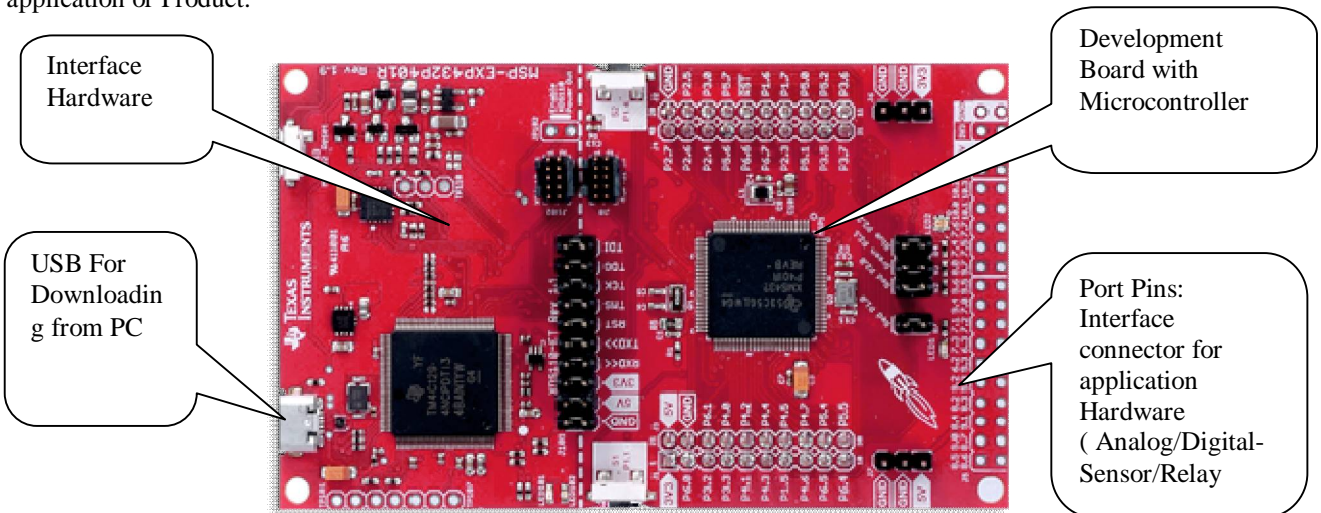


Fig.2 Launch Pad – MSP-EXP432P401R

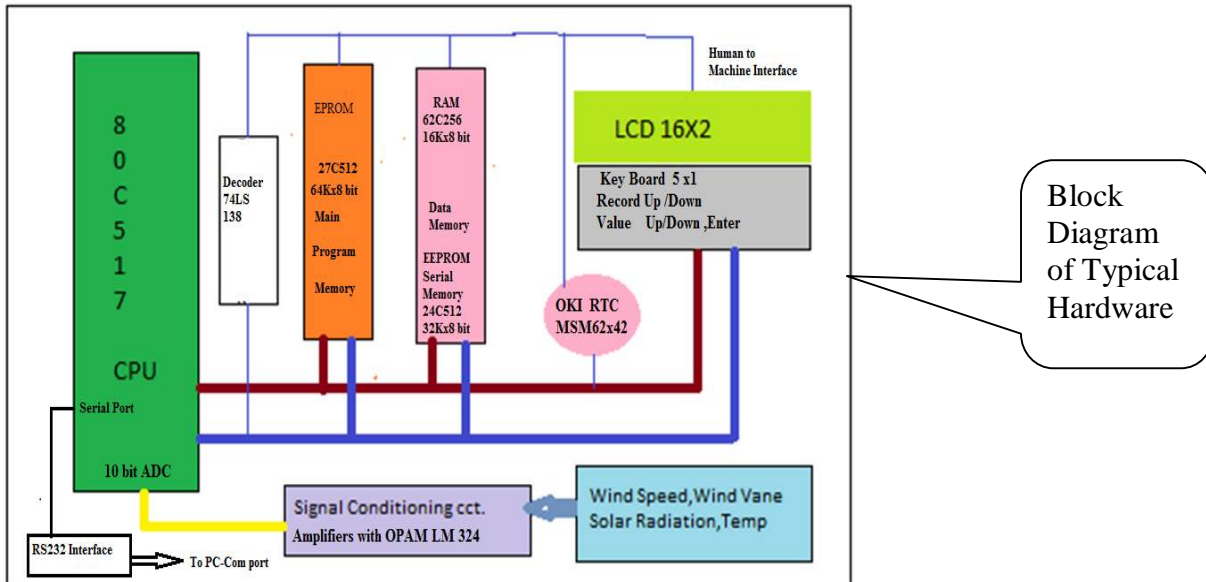


Fig.3 Hardware Design

Circuit Schematic Developed in Eagle PCB design Software

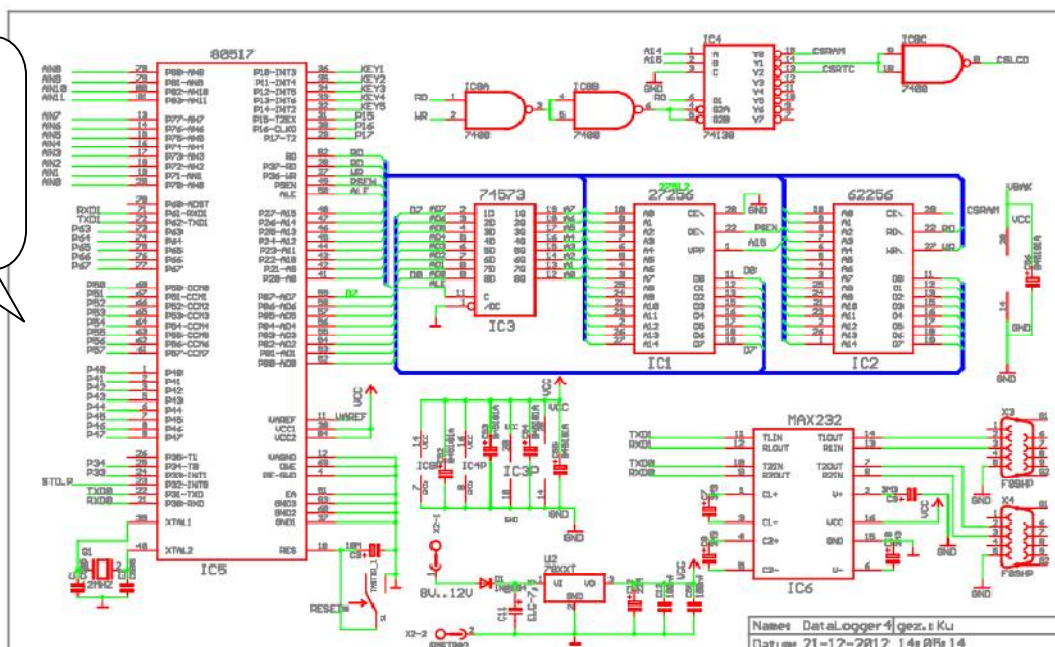


Fig 4: Schematic [Circuit diagram] Development in Eagle PCB design software

The documents generated for production process are

- Circuit diagrams
- Assembly procedure and Test procedure for hardware
- Part list or Bill of material

IV. EMBEDDED SOFTWARE DEVELOPMENT

The Software is developed using “C/C++ Cross Compilers” (Vendors-GNU , Keil , Tasking ,IAR etc).These cross compilers runs in operating system such as windows or Linux and produces object files. The C program is developed using IDE and compiled using Cross C Compiler utilities whose tool chain planned inside the IDE. The output of the compiled file is object file which is linked using Linker program to get Hex file for Burning inside the EPROM/Eprom simulator or downloading this application program in flash program memory using J-Link or Bootblock use. For writing modular programs function wise program modules are developed in separate C files and compiled in OBJ files. These becomes tested program files for specific hardware for eg. Keyboard , Display ,Menu ,CAN communication functions etc.

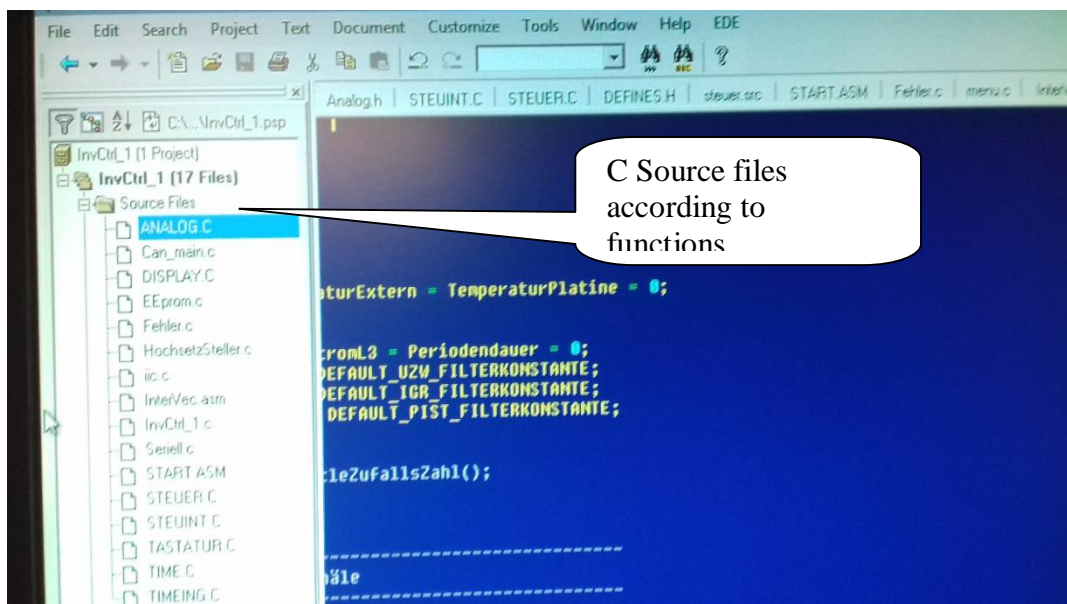


Fig 5 : Typical IDE environment (Tasking Compiler)

V. RESULT AND DISCUSSION: FRAMEWORK AS EDUCATIONAL TOOL

Embedded System : As the name suggest microcontroller is not the monopoly of Electrical / Electronics engineers , it can be used by other branches of Science / Technology field . A brief basic needs were discussed in this paper above for selection of hardware , software as per targeted application. This framework can be used by B.Sc/B.CS students to acquire Industrial skill development program required for Make in India. The pictorial representation given here illustrate the subject and process of designing. The circuit schematic are used to represent Hardware (which is developed in EDA CAD softwares like Eagle/P CAD/ORCAD etc).Software development process is illustrated in execution flow steps. Our framework can aid the education process by offering student to build there hardware models by interconnecting the microcontrollers with Interface design for the specific application and enabling them to easily convert from Conceptual High Level models to efficient Product implementations using contemporary software tools. This framework can be used in courses dealing with basics of microprocessor and microcontrollers ,their programming is assembly/ c language) and further to System / Product design process. The most motivated students in group can also fabricate system themselves. At University Level the framework can be used as practical tool to create Educational Equipment for the courses dealing with microprocessor and microcontrollers or embedded system, covering both Hardware pheripheral design as per specific application and programming modules for Hardware interfacing. Due to

the growing need of interdisciplinary education the system is also attractive solution for students of other science / technology streams e.g. Chemical , Biology , agriculture , mathematics , this framework can help to understand which include computer architecture , digital signal processing , mechatronics etc.

VI. CONCLUSION

- 1] In this paper the A general Framework for developing embedded Microcontroller based applications including Hardware system design and Software System design and required developments tools are discussed with some proposals to be used in educational courses.
- 2] With proper guidance University professors can develop Product development / Hardware Development/ Software development /Application engineering skill in student with this framework and easily available (even free development Hardware / software samples) from Manufacturers and build Industry and Education Institute partnerships.
- 3] The Framework is sufficiently general in nature and not focused to particular microcontroller device, but it enables to select the most appropriate device for each application design. As such A framework is particularly appropriate for education purpose , trying to support the students entering to the great world of Microcontrollers.

REFERENCES

- [1] A. H. G. Al-Dhaher, Integrating hardware and software for the development of micro-controller-based systems, *Microprocessors and Microsystems*, vol. 25, iss. 7, October 2001, pp. 317-328.
- [2] M. Predko, *Programming and customizing PICmicro microcontrollers*, McGraw-Hill, 2000.
- [3] P. Caspi et al., Guidelines for a graduate curriculum on embedded software and systems, *ACM Transactions on Embedded Computing Systems*, vol. 4, iss. 3, August 2005, pp 587-611.
- [4] W. Wolf and J. Madsen, Embedded systems education for the future, *Proceedings of the IEEE*, vol. 88, iss. 1, January 2000, pp. 23-30.
- [5] P. J. Mosterman, Automatic Code Generation: Facilitating New Teaching Opportunities in Engineering Education, *36th ASEE/IEEE Frontier in Education Conference*, October 2006.
- [6] D. J. Jackson and P. Caspi, Embedded Systems Education: Future Directions, Initiatives, and Cooperation, *SIGBED Review (Special Issue on the First Workshop on Embedded System Education)*, vol. 2, iss. 4, October 2005.
- [7] M. Smolnikar, MPICDs – MPIC development system, <http://MPICds.googlepages.com/>, December 2007.
- [8] P. J. Mosterman, S. Prabhhu and T. Erkinen, An Industrial Embedded Control System Design Process, *The Inaugural CDEN Design Conference (CDEN'04)*, July 2004.
- [9] M. Rupp, A. Burg and E. Beck, Rapid prototyping for wireless designs: the five-ones approach, *Signal Processing*, vol.83, iss. 7, July 2003.
- [10] –, Shortening the Embedded Design Cycle with Model-Based Design, National Instruments Corporation, <http://zone.ni.com/devzone/cda/tut/p/id/4074>, December 2007.
- [11] –, Flowcode and E-blocks, Matrix Multimedia Limited, <http://www.matrixmultimedia.com/>, December 2007.
- [12] L. F. Ferreira et al., MILES: A Microcontroller Learning System combining Hardware and Software tools, *35th ASEE/IEEE Frontiers in Education*, October 2005.
- [13] –, Technical Library CD-ROM 2004/2005 (DS00161Q), Microchip Technology Inc., 2004.
- [14] A. Clements, Selecting a processor for teaching computer architecture, *Microprocessors Microsystems*, vol. 23, iss. 5, October 1999, pp. 281-290.
- [15] –, Product Selector Guide 2005 (DS00148K2), Microchip Technology Inc., 2005.
- [16] –, PIC-Programmer, <http://www.jdm.homepage.dk/newpic.htm>, December 2007.
- [17] –, P16PRO and PICALL, <http://picallw.feniks-pro.com/>, December 2007.
- [18] –, PICmicro 18C Family Reference Manual – In-Circuit Serial Programming (DS39531A), Microchip Technology Inc., 2004.
- [19] –, MPLAB IDE User's Guide (DS51519B), Microchip Technology Inc., 2006.
- [20] –, PICC compiler, HI-TECH Software, <http://www.htsoft.com/products/compiler/picccompiler.php>, December 2007.
- [21] –, IAR Embedded Workbench, IAR Systems, <http://www.iar.com/>, December 2007.
- [22] –, CCS C Compiler, CCS Inc., <http://www.ccsinfo.com/>, December 2007.
- [23] –, mikroC compiler, mikroElektronika, <http://www.mikroe.com/en/compiler/mikroc/pic/>, December 2007.
- [24] –, Simulink, The MathWorks Inc., <http://www.mathworks.com/products/simulink/>, December 2007.
- [25] –, Embedded Target for 16 bits PIC, <http://www.kerhuel.eu/RTWdsPIC/>, December 2015.
- [26] –, FLEX: Microchip dsPIC evaluation board, Evidence, <http://www.evidence.eu.com/content/view/114/204/>, December 2007.
- [27] –, Scilab: The open source platform for numerical computation, INRIA, <http://www.scilab.org/>, December 2007.
- [28] –, Scicos: Scilab's block diagram modeler/simulator, INRIA, <http://www.scicos.org/>, December 2007.
- [29] –, IC-Prog Prototype Programmer, <http://www.ic-prog.com/>, December 2007.
- [30] –, WinPic - A PIC Programmer for Windows, <http://freene-thomepage.de/dl4yhfw/winpipr.html>, December 2007.
- [31] W.A. Stapleton, Microcomputer Fundamentals for Embedded Systems Education, *36th ASEE/IEEE Frontiers in Education Conference*, October 2014