INITIAL DESIGN REPORT

BY TITANSOFT

Tolga Özgün
Serhan Kars
Pınar Sönmezocak
Taylan Onur Utku

1 INTRODUCTION

1.1 Problem Definition:

Computer market is the most dynamic market of the contemporary World. Leader sectors of this market changes time to time. Today, embedded electronics drive the computer market.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product This feature makes embedded products the shining star of the market. Moreover, as it is a developing technology, there is not a fully robust development environment for it so embedded electronic environments are seen as a forsaken palace of software engineering.

A misleading view to embedded software is that, this technology is mostly concerned by electronic engineers however rapid development in embedded technologies calls for special software engineering techniques. As a result of these technique and obvious difference between high level programming techniques, a special development environment is required.

To provide personal for this special software techniques most of the computer engineering departments teach corresponding courses. Middle East University Computer Engineering department (METU-CENG) is one of these departments. In 2006- spring term, METU-CENG staff decided to use PIC microcontrollers instead of INTEL microcontrollers and a TA of CEng336 course, Alper KILIC, developed a special board for the course CENG-336.However, using such board and developing software projects brings some problems as below:

- Frequent upload of code to board is time consuming,

- Microprocessor and their components are so sensitive devices that small lack of care causes malfunctions, most of the time these malfunctions are not recoverable and they cost considerable amount of money and time ,
- Even there are emulators for PIC and its components , most of the time they require technical background which is generally not related at corresponding courses,
- METU-CENG staff and students prefer UNIX environment for courses however current emulators and development environments developed for Windows environment,
- Software engineers prefer vi as a text editor however such a development environment does not exist,
- Current Integrated development environments(IDE) provides a emulators for the track of code through registers and memory however they do not provide such features for components like LCD, Led Display , Keyboard etc
- Platform independency is much desired feature for every kind of applications, especially students have different operating system choices; this makes platform independency more valuable.

1.2 Project Scope and Goals

Main purpose of our project is to develop an IDE built on eclipse with the feature of emulating the board for CENG-336 course. However as an expansion point we will aim a general use IDE plug-in that can be used for PIC programming. The scope and features of the projects are stated below:

- Compiler: IDE will compile given PIC-assembly code in to hexadecimal code and will generate compile-time errors for syntax and semantics errors.
- To avoid hardware malfunctions, emulator is needed for the PIC microprocessor and components. Emulator will specially configured to PIC board used in CENG-336.

- IDE will check paging errors , if code exceeds page limits IDE will give warning,
- Run Time error: checking in simulator phase will be implemented. For instance stack overflows will be detected.
- Project Templates: To reduce time consumption for common projects templates will be provided.
- Workspace: Working environment of user will be saved therefore user setting such as opened files, emulator settings will be remembered.
- Debugging: Programmer will be able run code in emulator, will be able to put breakpoints, watch registers, and track memory-stack. Programmer can modify registers, memory during runtime.
- Code Completion, Interactive Help: During coding stage IDE will give assistance for the completion of commands and give warning in the case of syntax errors. Programmer will be able to get instance help about commands, for instance user will highlight the command or register, call help and help screen will display corresponding information. Calling help could be by a hot-key etc
- Data Explorer & Types: One of the challenging parts of low level programming is that they do not have data structures like high level programming languages. Our solution for the matters is user-defined data structures as collection of register, memory locations. Moreover user can tag certain parts of code to define data structure. These structures can be followed by a class-view like window.
- Favorites Code Block: Programmer can tag certain parts of the code as favorite and quickly jump to that location.
- Hot-Key Feature: Programmer can reach certain features of the IDE and simulator via hot-keys. Hot-keys can be defined by users and IDE will provide some common templates such as Visual Studio key settings or MPLab key settings.

- Component Feature: All components in pic-board (CENG-336 board) will be attachable and unattachable. User will be able to attach or unattach these components according to project requirements. Moreover as an open issue user will be able to create new components through a wizard step-by-step and attach this to simulator.
- Some common libraries will be provided to user like LCD, Clock-Timer commands.
- Clock-Frequency calculator: Programmer will be able to measure performance of the program or code segments. Clock cycles will be provided as a measure unit.
- Plug-in Feature: Programmers who wants to add new feature to IDE be able to write their own plug-ins and import them.
- Search/Replace: Programmer will be able Search/Replace keywords through project or current document or in selection.
- Higher Level to PIC : This feature is an open issue, we are thinking about designing a higher level language or optimizing a common used higher level language, possibly C, to write PIC programs. IDE compilation feature will compile this higher level language and produce PIC-assembly and then hex file.

1.3 Usage Areas:

Titansoft development environment will be a flexible development environment by plug-in mechanism.

UNIVERSITES: The main purpose of this project is to establish a development environment for CEng336 course. But every university who demands a robust development environment for corresponding course can demand and use this development environment. As it visualizes the codes of the student, it can be helpful in learning process. Moreover, this can reduce the cost of universities and they can save both money and time.

COMPANIES: This development environment can also be used in industrial area if the company is developing an embedded system which is supported by Titansoft plug-in mechanism. Companies using this development environment can shrink the costs.

1.4 Design Constraints:

- **TIME:** Time constraints are given in the syllabus. According to syllabus, we have approximately 6 months to finish the project and deliver it to end user. We are at the design phase and we have to produce a prototype within 6 weeks time. Detailed time constraints are stated in the Schedule part of this report.
- **LANGUAGE:** We decided to use JAVA language as implementation language. It allows us reaching many documents and easy programming. There are many plug-in and development environment example coded with Java and we can get benefit of these developers' experiences. However, we are worried about the execution time of java codes compared to C++ codes
- **MAINTENANCE:** Titansoft development environment will require minimal maintenance. Moreover, if users want to update the development environment, user should save the newly distributed plug in to predefined location and then s/he can use the upgraded version of the development environment
- **PERFORMANCE:** As java is used as a programming language, efficiency care in programming should be taken. Otherwise the latency problem will not be handled.
- **USER INTERFACE:** Designing a user friendly interface is our one of the most important aim. Some students believe that developing an embedded system is boring. This development environment should help student and other users by reducing the latency of programming, and quick accessing some highly used tools vice versa.

1.5 Design Objectives:

- **Extendibility:** As stated above, embedded market is a developing market. If this development environment is intended to be used also in the future, it should be updated regularly. The plug-in support of the Titansoft development environment realizes this purpose and makes it flexible and extendible.
- **Portability:** One of the main drawbacks of the currently used development environment is that it could not be operated under UNIX machines. Owing to Java, this development environment is platform independent.
- **Usability:** As this programs` candidate users are students, we should put extra effort to design user friendly development environment. This will ease the task of students.

2 REQUIREMENTS

The details of hardware and software requirements were stated in the requirements analysis report. But IDE programming mentality has been changed and updated hardware and software requirements are given briefly below.

2.1 Hardware Requirements:

- **CEng336 Pic Board**
- **Development PC with**
  512 MB RAM

30 GB Free hard disk space

Internet Connection

Intel Pentium IV 1.6 GHz or equivalent AMD

Parallel Port

- **For End User**

256 MB RAM

150 MB free hard disk space

Graphic card

Mouse

Intel Pentium IV 1.2 GHz or equivalent AMD

Parallel Port

2.2 Software Requirements:

- **Development Phase:**

Eclipse software development kit

MS Windows

UBUNTU 5.x

MS Office Word

MS Visio

Smartdraw

- **End User Needs:**

Eclipse SDK

Any platform having a compatible Java run time environment is enough for end user.

2.3 Functional Requirements:

Main functional requirements are listed below.

- **Emulation:**
  Sometimes programming the processor will not be enough, user may want to see the simulation of board components via software. Development environment will emulate the codes written by user in another window.
- **Enable/Disable Transfer:**
  In some cases, code transfer from computer to processor should be suspended in order not to damage the board. After solving problems, user can enable the data transfer.
- **Save/ Load:**
  User should be able to save his/her projects under requested files. Then users should be able to load it and continue his/her projects. Save/load functionality can prevent data and time loss.
- **Add/Remove Component:**
  User may want to add or remove a new board component in emulation part. These components will be found in development environments library.

  In some cases, users want to remove components in order to see emulation clearly and increase the execution speed of the program.
- **Error Handling:**

User should see his/her mistakes and debug the code. Efficient error handling is required. Paging error handling and run time error recovery should be considered specially.

- **Hex Conversion:**
  Codes should be translated into .hex file in order to be interpreted by the development environment.
- **Breakpoint Setting:**
  User can set the breakpoints and trace his/her code efficiently.
- **Register Watching:**
  Sometimes users may want to see the content of the registers in order to understand the working mentality of assembly language.

2.4 Non-functional Requirements:

Non functional requirements is given below briefly

- **User Interface:**
  User interface is the interface of the eclipse which is enriched with plug in features. The interface should be robust and ease the workload of students. Pretty printing may help students debug their code easily.

- **Maintenance:**
  After delivery, the program should be maintainable. As stated before, embedded technology is growing rapidly and new hardware and software products are released to market. Thanks to plug in feature, development environment can be updated.

- **Platform Independency:**
  This development environment should be platform independent in order to meet teaching staff of METU-CENG and some students who desire to program embedded systems in UNIX.

- **Search and Replace:**
  User can search and replace code blocks so s/he can correct large block of mistakes.
- **Reliability:**
  The development environment should be as robust as possible. All modules should make their work without interrupting the normal routine of another module.

3 TESTING

As this is an initial design report, only basic testing strategies are explained.

3,1 Unit Testing

We are going to test units before integration. Some fundamental units are:

- **LCD**

  Some characters should be handled specially as they are troublesome while showing in liquid crystal display.
- **Processor**

  Connection and communication of processor-PIC- with PC and development environment is another issue that we should put more effort.
- **Button-keyboard- Set**

Synchronization of buttons is another concern. If two buttons are pressed at the same time, development environment should not get confused. The environment should get the difference of pressing the button, releasing the button and pressing to button for a long time.

- **A/D Converter**

  Conversion of analog input into digital input is the key point while designing A/D converter. Information loss will make the development environment useless bunch of program.

- **LED Display**

  The testing strategy of LCD display should also be applied to LED display testing.

- **Sample Plug in**

  We should put extra effort in accordance of plug-in with eclipse. Library conflict can also be troublesome.

- **Syntax Checker**

  The performance of error detection and error recovery of development environment is affecting the performance of students and embedded system developers. Helpful syntax checker should be developed and it should be bug free.

- **UART**

  Encryption of characters is very important as there can be data loss due to wrong encryption and transmission.

- **Emulator**

  We should put extra effort in emulation performance and emulation reality and these should be tested seriously. This unit's performance will distinguish our candidate development environment from the others.

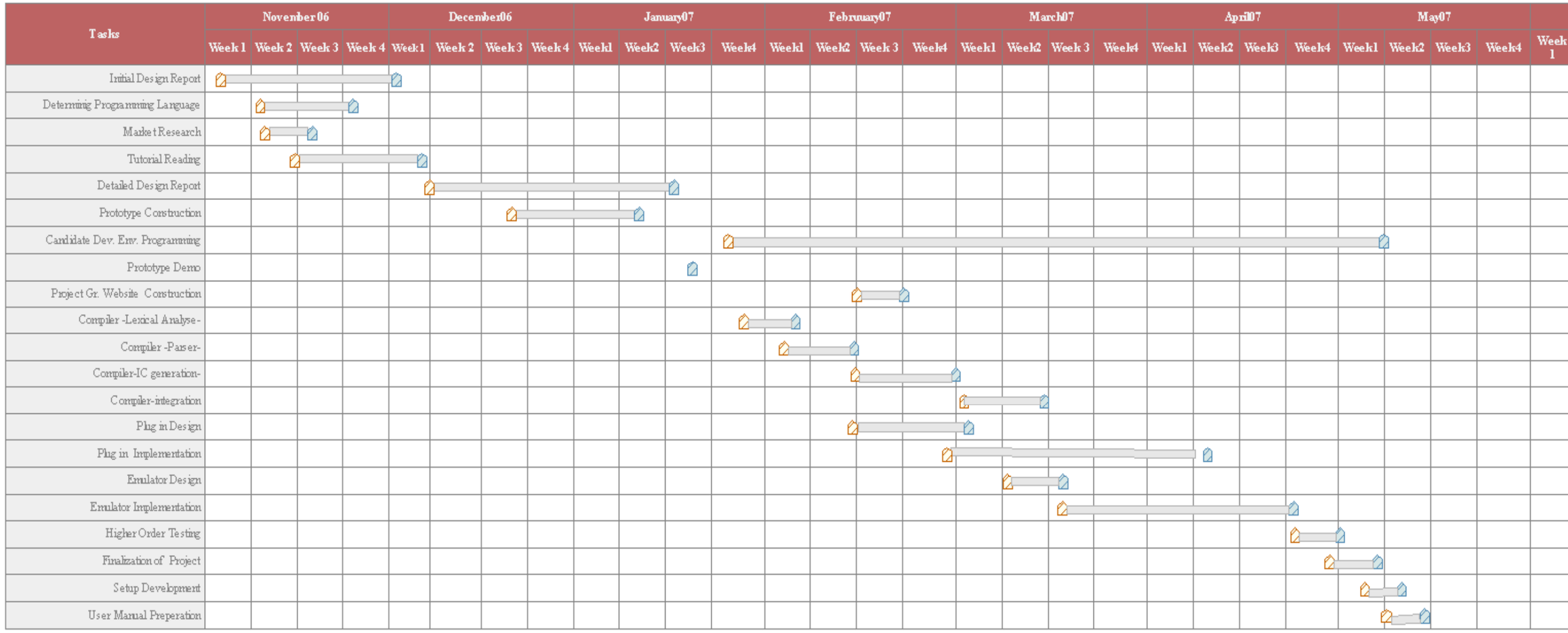
3,2 Integration Testing

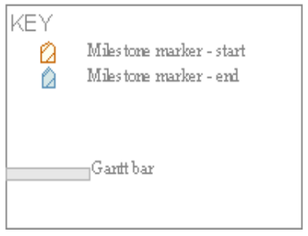
After completing all components programming and testing, units should be integrated. After integration phase, units should be tested under integrated development environment.

4 SCHEDULE

4,1 Gannt Chart

GANTT CHART - 8 MONTH TIME LINE





4.2 Mission Set

- **Finished Activities:**

  Customer Communication

  Specification of Requirements

  Initial Design Report

  Learning the language and tools

- **Undone Activities:**

  Plug in designing

  Design

  Establishing Prototype

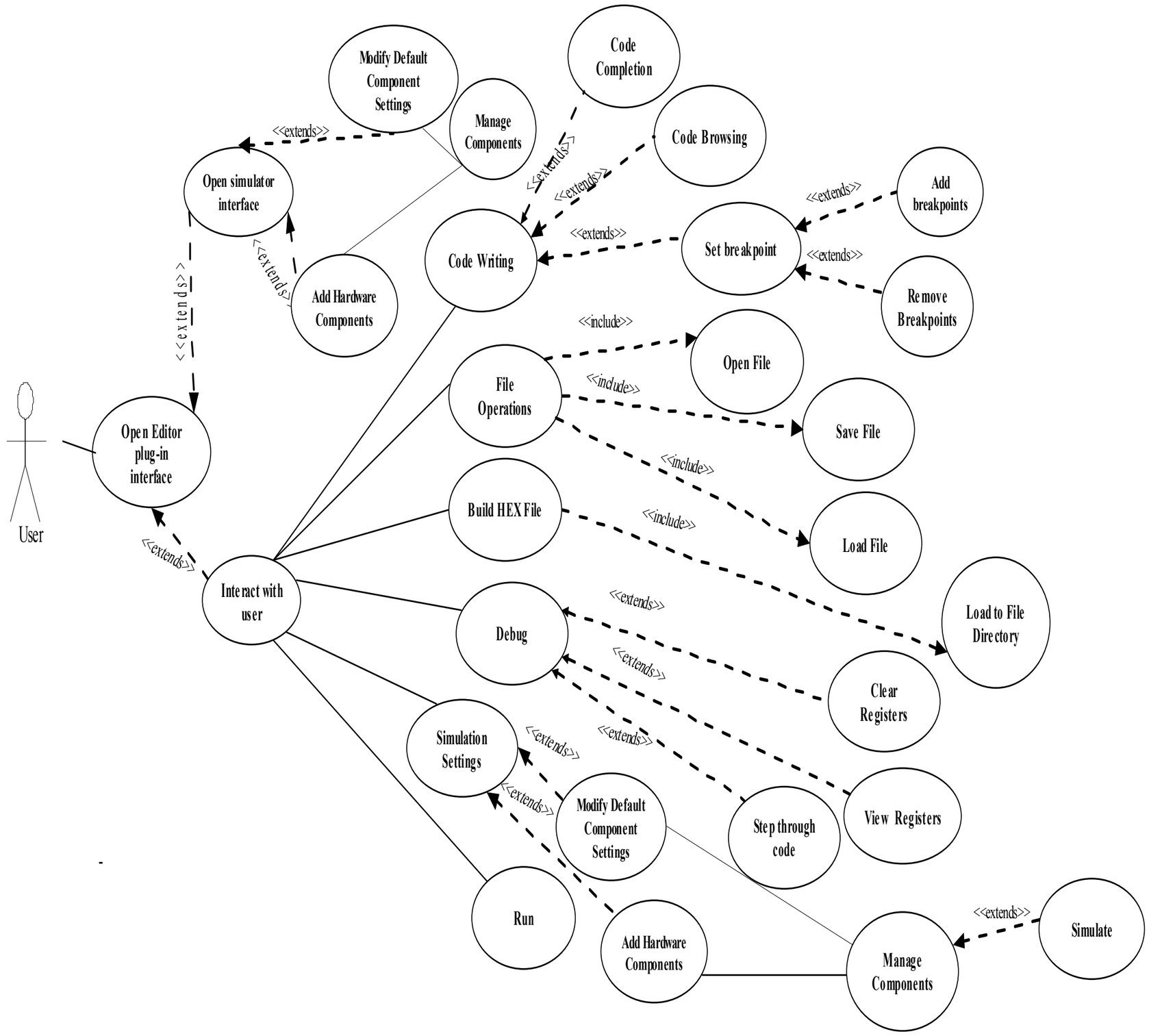
  Interface Construction

  Programming

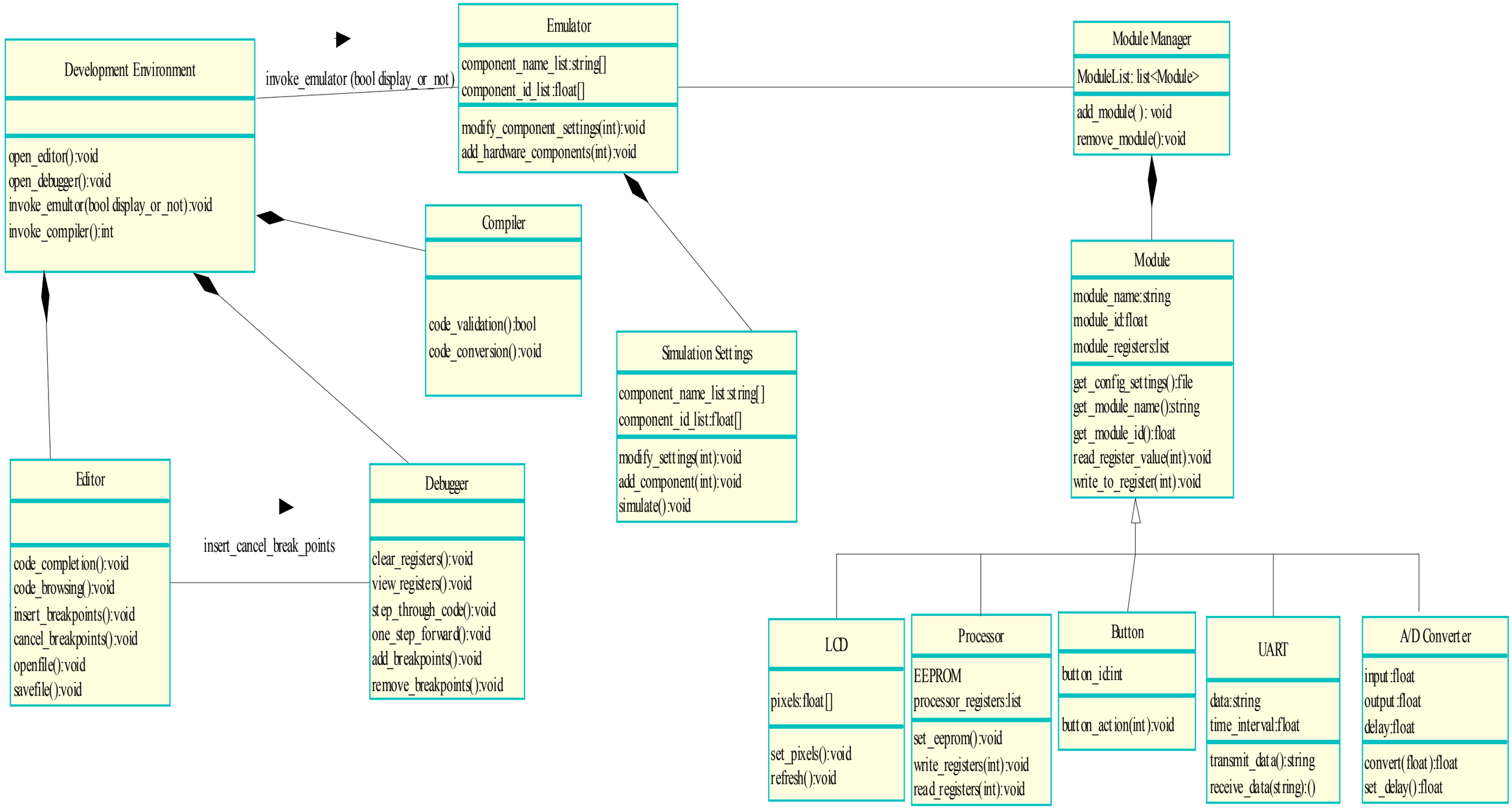
  Adaptation of units

  Testing

5.USE CASES

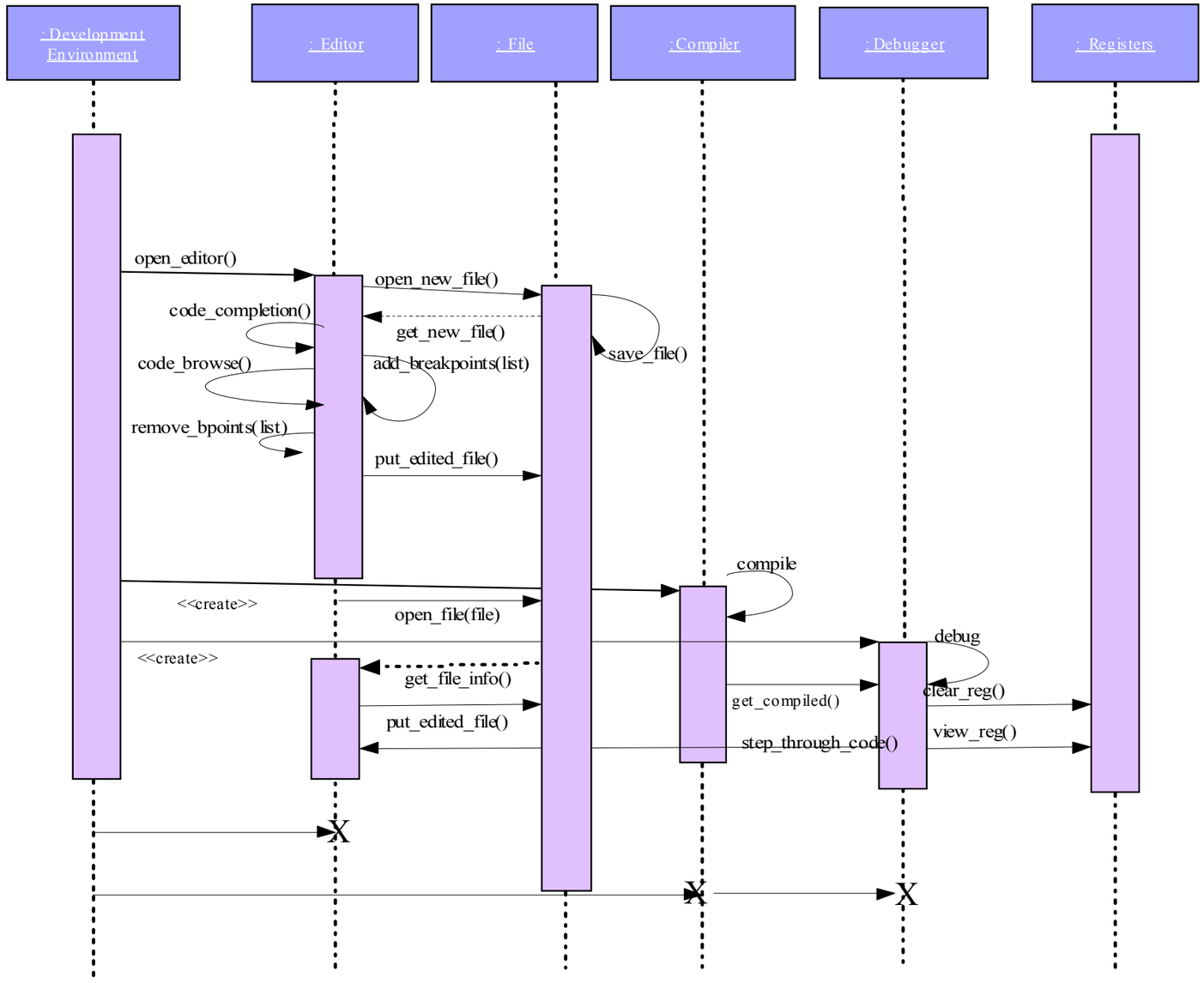

Modify Default Component Settings
Manage Components
Code Completion
Code Browsing
Open simulator interface
Add Hardware Components
Code Writing
Set breakpoint
Add breakpoints
Remove Breakpoints
Open Editor plug-in interface
User
File Operations
Open File
Save File
Build HEX File
Load File
Load to File Directory
Interact with user
Debug
Clear Registers
View Registers
Step through code
Simulation Settings
Modify Default Component Settings
Run
Add Hardware Components
Manage Components
Simulate
<<extends>>
<<include>>

6.CLASS DIAGRAM

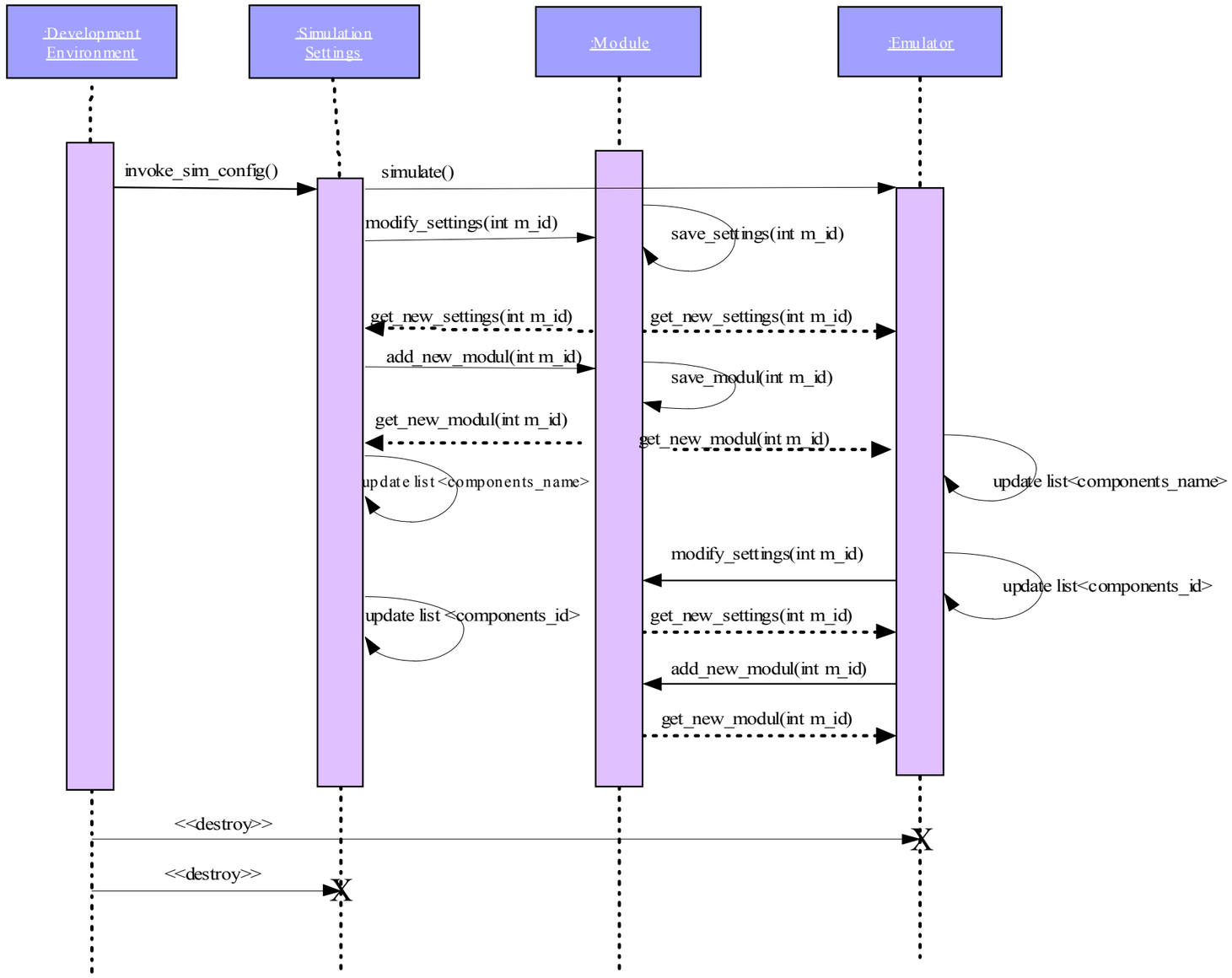


7.SEQUENCE DIAGRAMS

7.1 SEQUENCE DIAGRAM1_INTERACTION WITH USER

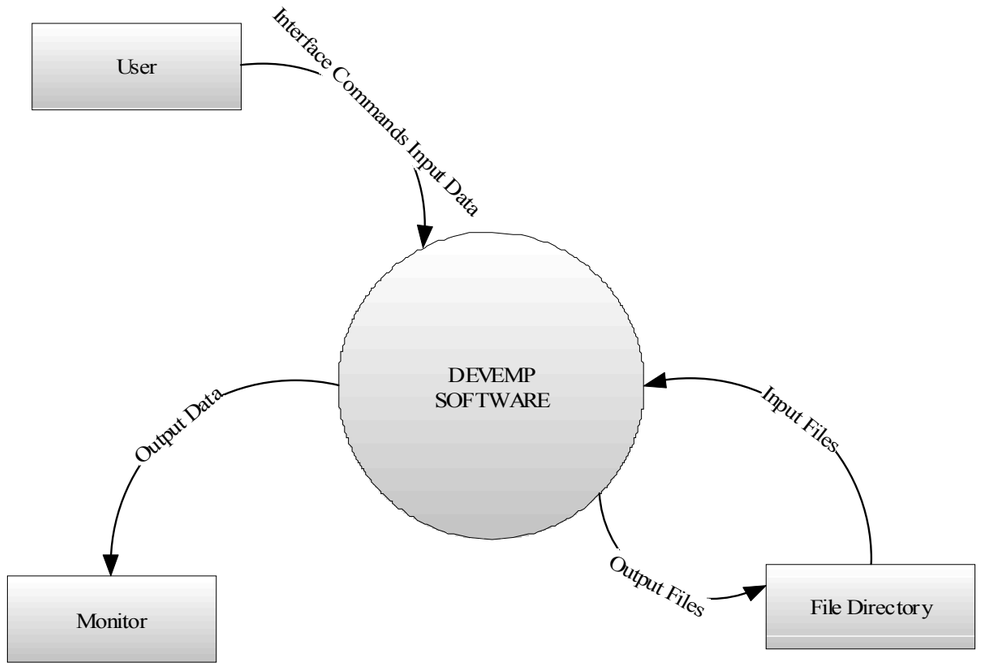


7.2 SEQUENCE DIAGRAM2_SIMULATION PART

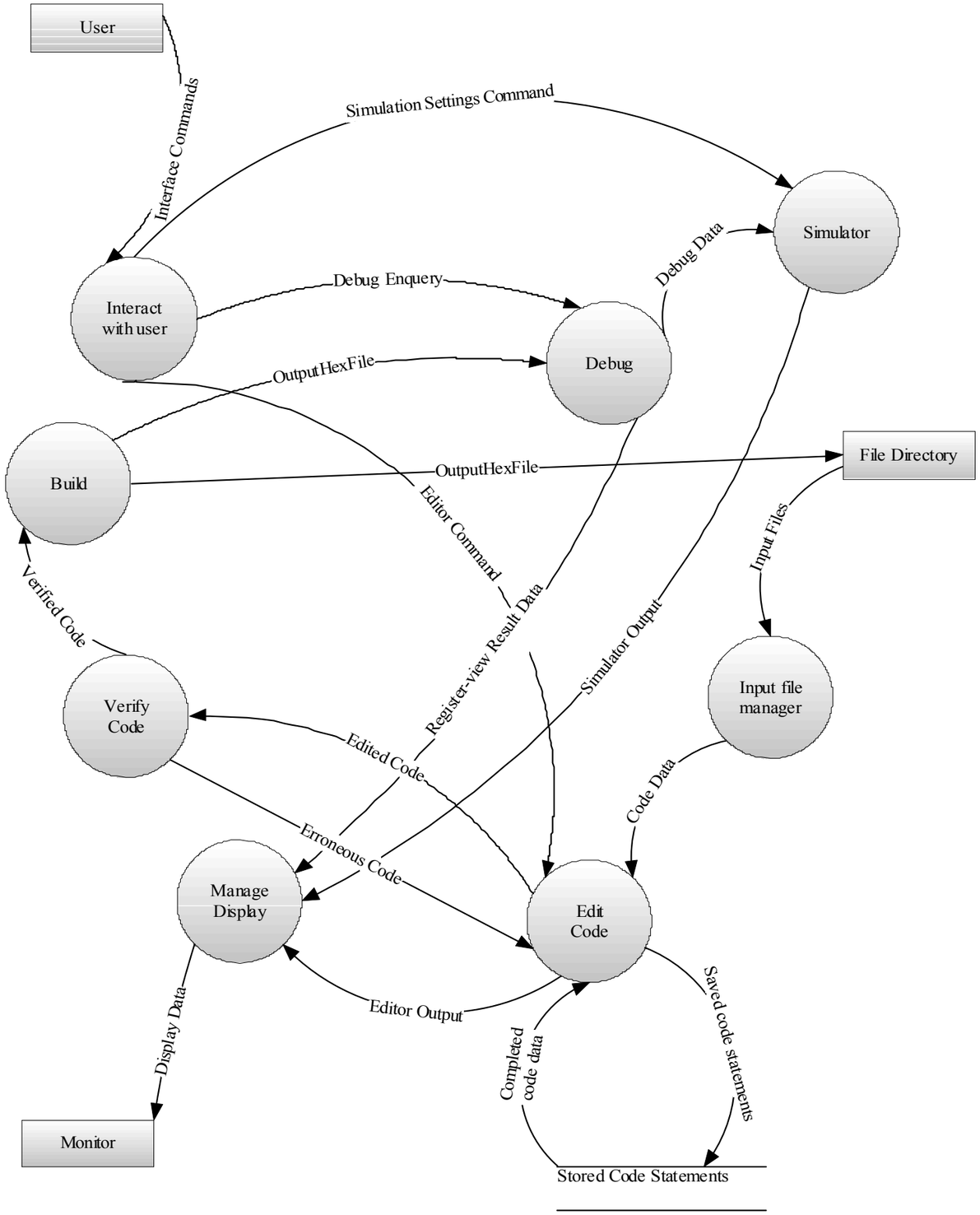


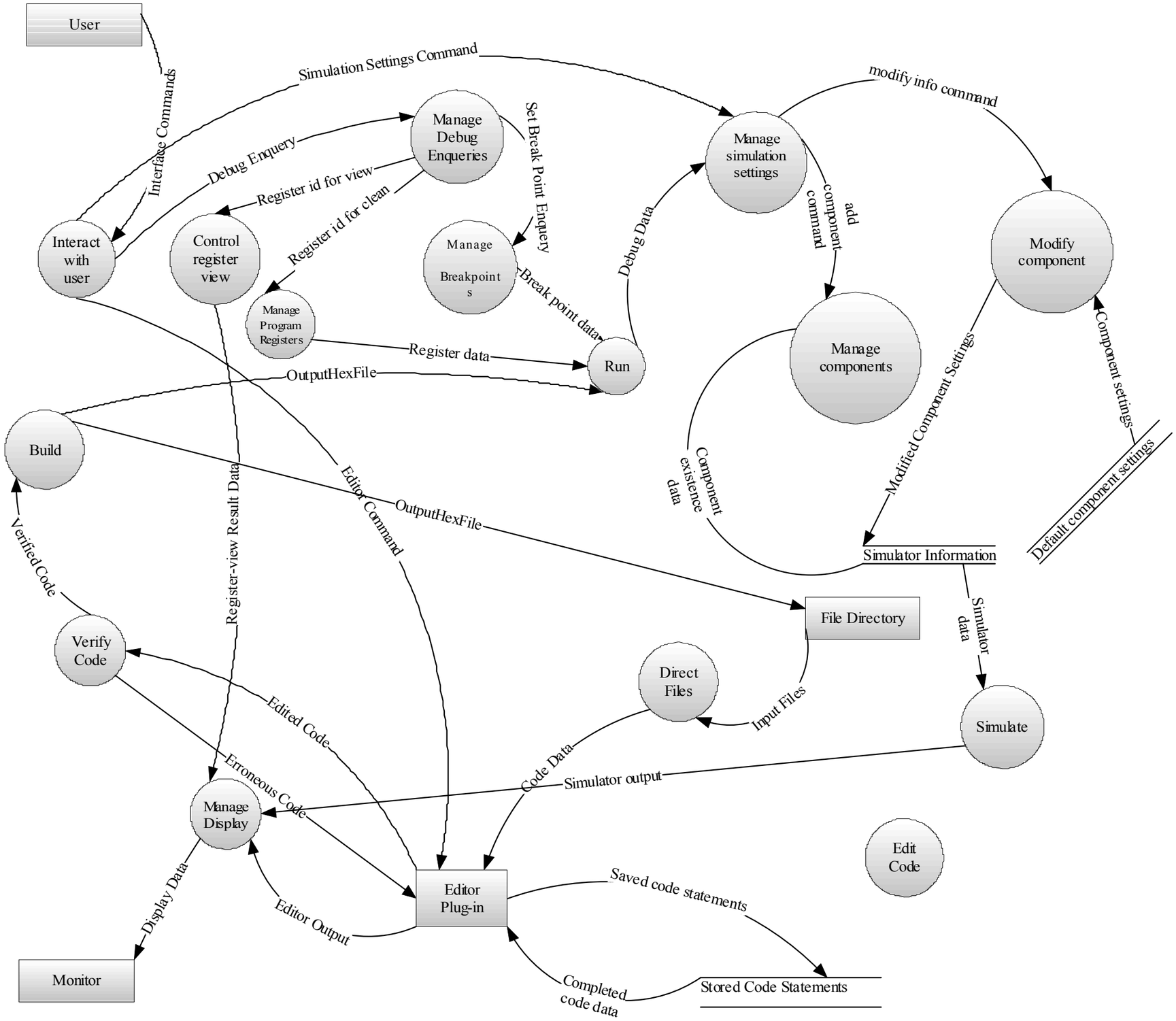
8.DATA FLOW DIAGRAMS

LEVEL 0 DFD



LEVEL1 & LEVEL2 DFDs


User
Interface Commands
Simulation Settings Command
Simulator
Debug Data
Interact with user
Debug Enquery
Debug
OutputHexFile
Build
OutputHexFile
File Directory
Editor Command
Input Files
Verified Code
Data
Register-view Result Data
Simulator Output
Verify Code
Input file manager
Edited Code
Code Data
Erroneous Code
Manage Display
Edit Code
Editor Output
Saved code statements
Display Data
Completed code data
Monitor
Stored Code Statements


User
Interface Commands
Simulation Settings Command
modify info command
Manage Debug Enqueries
Set Break Point Enquery
Debug Enquery
Register id for view
Register id for clean
Interact with user
Control register view
Manage Breakpoints
Manage Program Registers
Break point data
Debug Data
Manage simulation settings
add component command
Modify component
Manage components
Register data
Run
OutputHexFile
Build
Verified Code
Register-view Result Data
Editor Command
OutputHexFile
Component existence data
Modified Component Settings
Component settings
Default component settings
Simulator Information
File Directory
Simulator data
Verify Code
Direct Files
Input Files
Simulate
Edited Code
Code Data
Simulator output
Erroneous Code
Manage Display
Edit Code
Editor Plug-in
Saved code statements
Display Data
Editor Output
Monitor
Completed code data
Stored Code Statements

9.USER INTERFACE

For our initial design report general outlines of our user interface will be determined to give an intiution about how users are going to interact with features of the program, therefore do not forget any sample screenshot shown below is not the final interface. Screenshots just putted there to give the essence of user interactions with the program.

As stated earlier, mainly IDE composed of two parts : The part that assembly code is written which is the Editor-Compiler part namely Eclipse plugin part, and the other part is emulator part where user is going to be able run the code and watch how board is going to react the code written in Editor part.

Challenging part in Interface design is to make connection between emulator and eclipse part. The question is that where the setting corresponding emulator will be put. Putting related controls only in Emulator will not be very very useful since users will as for changing emulator behavior without running it and configure it while writing code. As a result we decide to put some of controls related to emulator into eclipse part so user can setup emulator and editor settings from the same environment.

Another Challengin part is to prepare interface of the Emulator part which is very complicated to prepare for the initial design phase. There for outlines of the emulator interface will discussed through the report and basic criterias will be stated however no screenshot will be demonstrated through report since we do not want to any wrong ideas to settle in minds about emulator interface since it is mainly a graphical part related to board parts.

Beyond this point of the interface part , interface parts will be explained in detail.

9.1Editor , Eclipse Plug-in

In this part of the interface design some interface features will be introduced to user , again do not for get these interfaces below are for only giving initial ideas.

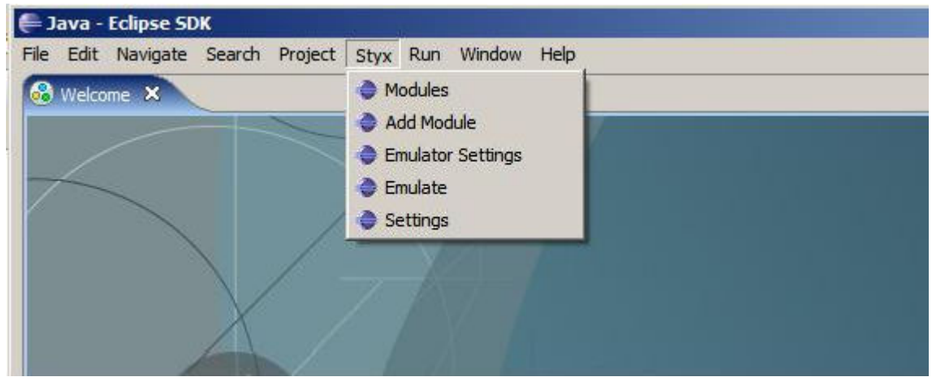


Figure 1

In figure 1 general menu in eclipse is shown. This menu and its derivatives will appear as a right click menu and as shown in figure as a main-bar menu. Buttons in the menu bar opens related menus that are described in detain below.

Modules :

In this Window User can interact with available modules . User will be able to add , remove ,configure modules. As a remainder, modules corresponds to components of the PIC hardware for instance for our board lcd screen, keyboard is a module. In this window as user selects module , corresponding settings will appear in this interface, user will be able to change this settings. To demonstrate this interaction below some screenshots are placed :

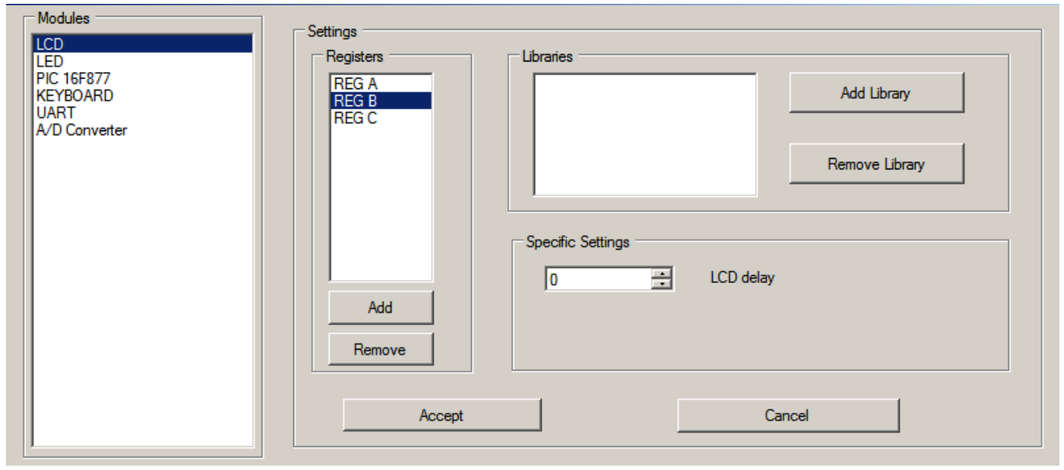


Figure 2

In Fact Figure 2 gives a general idea for all property. In “Modules” List box user selects the Module that will be configured . At the right side of the panel user can make corresponding manipulations . Can add registers that is related to corresponding Module. In figure 2 For the sake of giving a simple understanding of configurations a basic register selection option is added , however in final package user will be able to do further manipulations on registers like connecting specific ports of the LCD to selected registers. Another option is “Libraries”. In this area user will be able add pre-defined libraries to selected module. When coding if user adds the module than related libraries will automatically included. “Specific Settings” area another area that user will add alter hardware specific options. In this example user can alter LCD delay which determines the LCD refresh rate.

Add Module

Via this button user will be able add wizard by the help of a wizard. Wizard will guide user through the module adding processing such as adding libraries, settings,registers. In other words user will be guided to fill settings in figure2 but in a much detail way.

Emulator Settings

User will be able configure settings of the emulator via this button. The configurable properties by this option are which and how emulator screens will be displayed. A possible scenario is that user selects just to see EPROM of processor that user must specify corresponding option from emulator settings.

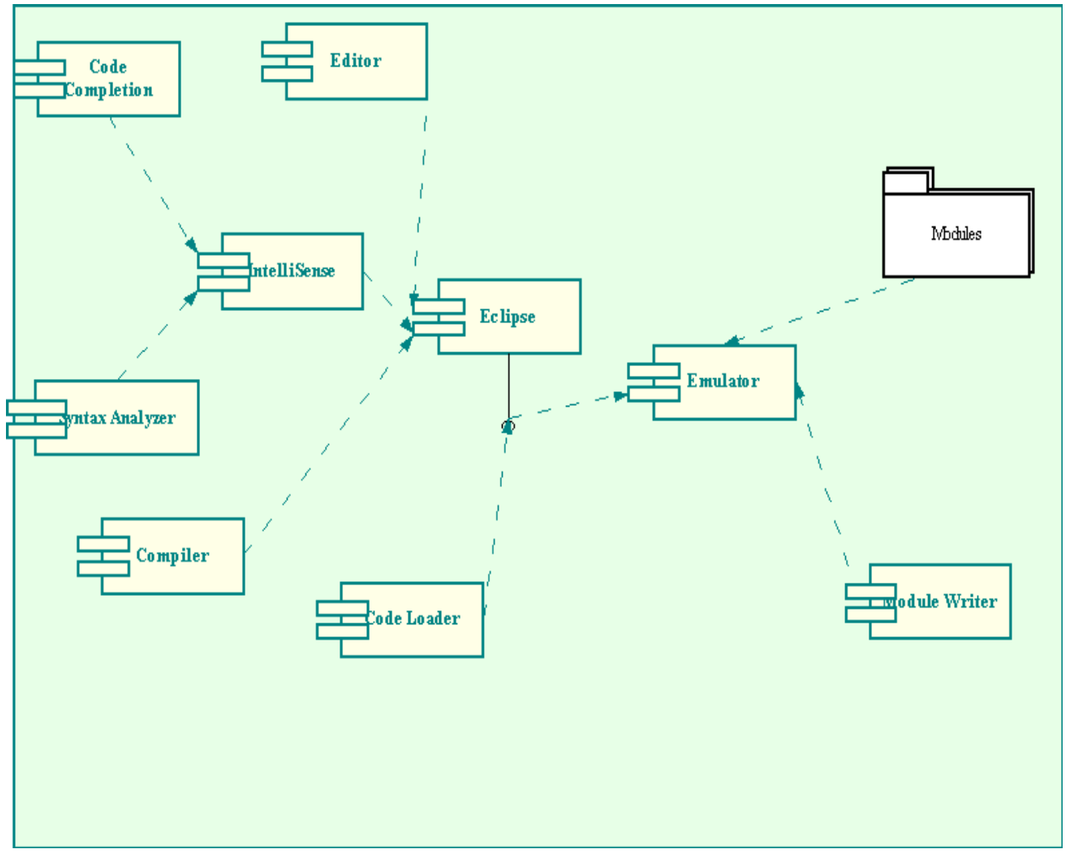
Emulate

User starts the emulation phase. Program opens emulator and emulator starts to emulate the compiled code.

Settings

User changes editor options like themes, syntax coloring , intellisense options.

10.Components



Components can be classified in two main groups : Eclipse side Components these are Editor,Intellisense,Code Completion,Syntax Analyzer,Compiler ; other main group is Emulator side which is composed of Modules and Module writer component. Below descrtiptions of the components are given :

Eclipse :

In fact for whole project eclipse is used like an UI framework that assists other components like editor, syntax analyzer. Eclipse component will organize windows, save load functions and etc.

Editor :

The component that code is written and most of the interface features of other components are place . for instance after code segments passed from syntax analyzer , coloring and errors will be given in this component.

Intellisense , Syntax Analyzer, Code Completion :

These components are very related to each other. Intellisense takes outputs of Syntax analyzer and Code Completion and call corresponding methods in eclipse component.

Syntax Analyzer interprets code segments and checks syntax of the segments. Makes defined color changes in text for readability. Code Completion is very similar to syntax analyzer , when user calls Code completion code completion takes code segment and gives back the alternative completion options.

Compiler :

Compiles PIC code into hexadecimal .

Code Loader :

Loads code to PIC board from parallel port.

Emulator :

Emulator takes code from eclipse part and emulates the running of the code in defined emulator environment. This component composed of litter components like memory manager, processor etc however this components will be detailed in final design report.

Modules :

Abstract the behavior of PIC hardware devices like LCD, analog to digital converter.

Module Writer:

Encapsulates the API for module writing.