UNIT III

EMBEDDED FIRMWARE DEVELOPMENT ENVIRONMENT

3

Programming in Assembly and HLL

• Processor and memory- sensitive instructions: Program codes may be written in assembly
• Most of codes :Written in a high level language (LL),_C‘,_C++‘or Java

Assembly Language Programming

Advantage

- Assembly codes sensitive to the processor, memory, ports and device hardware
- Gives a precise control of the processor internal devices
- Enablesfulluseofprocessorspecificfeaturesinitsinstructionsetanditsaddressing modes
- Machine codes are *compact,* processor and memory sensitive
- System needs a smaller memory.
- Memoryneededdoesnotdependontheprogrammerdatatypeselectionandruledeclarations
- Not the compiler specific and library functions specific
- Device driver codes may need only a few assembly instructions.
- Bottom-up-design approach

Advantage of using high level language (HLL) for Programming

Short Development Cycle

• Code reusability ─ A function or routine can be repeatedly used in a program
• Standardlibraryfunctions─Forexamples,themathematicalfunctionsanddelay(),wait(),sleep()functions
• Use of the modular building blocks
• Sub-modules are designed first for specific and distinct set of actions ,then the modules and finally integration into complete design.
• First code the basic functional modules and then build a bigger module and then integrate into The final system
• First design of main program (blueprint), then its modules and finally the sub-modules are designed for specific and distinct set of actions.
• Top-down design Most favour program design approach

Use of Data Type and Declarations

• Examples , *char* , *int* , *un signed short, long, float* , *double* ,*Boolean*.
• Each data type provides an abstraction of the (i) methods to use, manipulate and represent, and (ii)set of permissible operations.

Use of Type Checking

• Type checking during compilation makes the program less pro net errors.
• Example ─type checking on a char data type variable (a character) does not permit subtraction, multiplication and division.

Use of Control Structures, loops and Conditions

• Control Structures and loops
• Examples ─ *while, do- while* , *break* and *for*
• Conditional Statements examples
• *if, if- else, else-if* and *switch - case*)

- Makes tasks simple for the program flow Design

Use of Data structures

_Data structure

-A way of organizing large amounts of data.

_A data elements‘ collection

_Data element in a structure identified and accessed with the help of a few pointers and/or indices and /or functions.

Standard Data structure

- Queue
- Stack
- Array–one dimensional as a vector
- Multidimensional
- List
- Tree

Use of Objects

_Objects bind the data fields and methods to manipulate those fields

_ Objects reusability

_ Provide inheritance ,method overloading, over riding and interfacing

_Many other features for ease in programming

Advantage of using C for Programming C

•Procedure oriented language (No objects)

- Provision of inserting the assembly language codes in between (called in line assembly) to obtain a direct hardware control.
- Alargeprogramin_C‘splitsintothedeclarationsforvariables,functionsanddatastructure, simpler functional blocks and statements.

In-line assembly codes of C functions

- Processor and memory sensitive part of the program within the in line assembly, and the complex part in the HLL codes.
- Example function out port (q, p)
- Example ─*M oval, p; out q, al*