UNIT-3

Embedded Firmware

By

veera

CONTENTS

3.1. Embedded Firmware

- ✓ Embedded Firmware refers to the control algorithm(Program instructions) and or the configuration settings that an embedded system developer dumps into the code(program) memory of the embedded system

- ✓ There are various methods available for developing the embedded firmware.

1) Write the program in high level languages like Embedded C/C++ using an IDE.

2) Write the program in assembly language using the instructions supported by your application's target processor/controller.

- ✓ The instruction set for each family of processor/controller is different and the program written in either of the methods given above should be converted into a processor understandable machine code before loading it into the program memory.
- ✓ The process of converting the program written in either a high level language or processor/controller specific assembly code to machine readable binary code is called <u>'HEX file creation'</u>

- ✓ The methods used for hex file creation is different depending on the programming techniques used.
- ✓ If the program is written in Embedded C/C++ using an IDE, the cross compiler included in the IDE converts it into corresponding processor/controller understandable ‘HEX file’.
- ✓ If the program is written in Assembly language, you can use the utilities supplied by the processor/controller vendors to convert source code into HEX file.

SYSTEM COMPONENTS

- The system components refer to the components/ICs which are necessary for the proper functioning of the embedded system.
- Some of these circuits may be essential for the proper functioning of the processor/controller and firmware execution.
- Watchdog timer, Reset IC, Brown-out Protection IC, etc are examples of circuits/ICs which are essential for the proper functioning of processor/controllers

3.2. Reset circuit

- ✓ The reset circuit is essential to ensure that the device is not operating at a voltage level where the device is not guaranteed to operate, during system power ON.
- ✓ The reset signal brings the internal registers and the different hardware systems of the processor/controller to known state and starts the firmware execution from reset vector.

- ✓ The reset signal can be either Active High or active Low.

- ✓ The reset signal to the processor can be applied at power ON through an external passive reset circuit comprising a capacitor and resistor or through a standard reset IC like MAX810 from Maxim Dallas.

- ✓ The processor behaviour may not be predictable if the power supply voltage falls below the recommended operating voltage.
- ✓ It may lead to situations like data corruption.
- ✓ A brown-out protection circuit holds the processor/controller in reset state, when the operating voltage falls below the threshold, until it rises above the threshold.

3.4. Oscillator Unit

- ✓ A microprocessor/Microcontroller is a digital device made up of digital combinational and sequential circuits.
- ✓ The instruction execution of a microprocessor/controller occurs in sync with a clock signal.
- ✓ Oscillator unit is responsible for generating the precise clock for the processor.

- ✓ Certain processors/controllers integrate a built-in oscillator unit and simply require an external ceramic resonator/quartz crystal for producing necessary clock signals.

- ✓ The speed of the operation of a processor is primarily dependant on the clock frequency.

- ✓ The total system power consumption is directly propotional to the clock frequency.

3.5. RTC

- ✓ RTC is a system component responsible for keeping track of time.
- ✓ It holds information like current time(In hours, minutes, and seconds) in 12 hour/24 hours format, date, month, year, day of the week, etc. and supply timing reference to the system.
- ✓ RTC is intended to function even in the absence of the power.