Mastering Microcontroller and Embedded Driver Development

About the Work

```
This work is about learning the concepts of various hardware components working
principles, architecture and programming of various Microcontrollers for utilizing
on my needs.

so the concepts and architecture I will be learning are
    - Embedded Code Debugging
    - Understanding MCU Memory Map
    - MCU Bus Interface
    - Understanding MCU Clocks and Details
    - Understanding MCU Clock tree
    - Understanding MCU Vector table
    - Understanding MCU interrupt Design, NVIC, Interrupt handling
    - Importance of "Volatile" Keyword
    - GPIO concepts
    - GPIO Programming structure and Registers
    - GPIO Registers : SPEED, PULLUP/DOWN, IDR and ODR
    - GPIO Alternate functionality register and usage
    - GPIO peripheral clock control
    - GPIO driver development and MCU specific header file with bus domain and
peripheral Details
    - Structuring peripheral registers
    - Writing Clock enable and disable macros
    - GPIO driver API requirements and handle structure
    - GPIO driver API Implementation : Clock control, GPIO init and de-init, GPIO
data read and write
    - GPIO pin Interrupt configuration
```

KEY WORDS USED IN EMBEDDED SYSTEM

```
- opcodes
- debugging
- breakpoints
```

Phase - 1

```
Understanding Basic Working of Microcontrollers
```

Things about to Learn

```
    - Embedded Code Debugging
    - Understanding MCU Memory Map
    - MCU Bus Interface
    - Understanding MCU Clocks and Details
    - Understanding MCU Clock tree
    - Understanding MCU Vector table
    - Understanding MCU interrupt Design, NVIC, Interrupt handling
    - Importance of "Volatile" Keyword
```

Information and Knowledge about the topics

Embedded Code Debugging

```
Embedded Code Debugging Options:
    - Serial Wire Viewer and data tracing(printf style debugging)
    - Single stepping, Stepping over and Stepping out
    - Breakpoint / Hardware Breakpoint (Inserting, Deleting and Skipping
Breakpoints)
    - Disassembly (Converting Higher Level language to Assembly level language)
    - Call stack
    - Expression and Variable windows
    - Memory browser (Used to examine various contents of memories of the
microcontroller, flash, SRAM, etc)
    - Data watch-points

1. SWV and ITM based prinf style debugging
    -
2. Single stepping, Stepping over and Stepping out
    - single stepping / stepping into ( stepping into the function or subroutine)
    - stepping over ()
    - stepping return / stepping over (stepping out of the function or subroutine)
3. Disassembly and Register windows

4. Breakpoints / Hardware Breakpoint
    - Breakpoints are the ways to tell the processor to halt or stop exeecution at
certain instruction address

5. Expression and Variable windows

6. Memory browser windows

7. Call stack and fault analyzers

8. Data watch-points
```

```
9. SFR windows (Special function register window)
```

Summary or General Understanding

MCU Bus Interfaces