The software structure used in building the LabGloves© is built using the following steps –

1. **Variable/Function Declarations -** We included all the required header files and associated C files including the ones for the LCD. This section also includes all the function declarations, variable declarations, LCD display buffers. We have defined each of the LabGloves© mode as a system state, by checking the system state variable the DMM function understands which case to switch ( ) on.
2. **Interrupt Service Routines –**

**ISR (TIMER0\_COMPA\_vect) compare vector interrupt** is used for scheduling the tasks. The main function ensures that the functions are timed based on the value of ‘T’ loaded into the ‘time’ and when it goes down to 0 the ISR triggers executing the functions in the main ( ).

**ISR (TIMER1\_CAPT\_vect)** – The ISR capture vector interrupt is to capture the value of ICR variable for making the frequency and capacitance measurements. It is linked internally with the analog comparator which executes the ISR when the comparator triggers. This ISR also consists of a logic that switches the ‘positive edge detect trigger’ to the ‘negative edge detect’ for calculating the Pulse Width and Duty Cycle without affecting the ‘time\_capture’ variable used in capacitance measurements.

1. **Key Detect Function & Debounce State Machine –** The capacitive touch breakout board is interfaced on the PIND.0 of the MCU. The output pin of the board goes HIGH when pressed and is otherwise LOW. Thus, by detecting the state of the PIND.0 we detect the switch press and pass it through the debounce state machine for avoiding the bouncing of the switch. We are using a time constant of 50 ms to check the debounce, since the average fastest human response is > 100 ms.
2. **Set and Reset PIN Logic –** In different modes of the DMM we are setting the PINS of the MCU to a logic level not required by other modes. Thus, it is required to set the PORT PINS based on the system\_state and reset it back once the mode is changed. This logic takes care that the PINS are restored back in the state they were by default so that the other modes are not affected. This gives us the flexibility of setting the port PINS the way we want in any given mode without bothering about the other modes.
3. **Main Function –** This is where the program enters at the beginning. It runs the initialize ( ) at the beginning to initialize the LCD, timers, PORT PINS and DDR for all the PORTS. After that the main ( ) simply acts as a scheduler to schedule the tasks at a given rate fixed initially. The main ( ) executes in a while (1) loop and runs continuously until the system is shut down by the user.
4. **DMM Compute Function –** This is the heart of the overall software development. Based on the existing system\_state these functions switches on different modes. It executes the logic written down in each of the code and generates the value - displays it on the LCD. It gets scheduled by the main () continuously after a certain time period. Each of the DMM case corresponds to a mode on the LabGLoves© and ensures that the display is wrapped after next three successive modes.
5. **Initialization Function –** As described earlier the initialize ( ) takes care to set all the PINS and TIMERS along with setting up the ADC and initializing the LCD display, setting up the global interrupts. It sets up different flags and system\_state in a way that they are used throughout the execution of the program. This function gets called at the beginning of the main ( ).