

Last updated: April 1, 2020

```

graph TD
    PowerOn([Power On]) --> Init[Initialize Motor controller, sensor, LCD communication etc.]
    Init --> ReadPot[Read Potentiometers  
Update:  
TV_temp  
BPM_temp  
I6_max_temp  
AT_temp]
    ReadPot --> SetButton{Set button pressed}
    SetButton -- Yes --> PreHome[Pre-home Move outwards at Vhome]
    PreHome --> LimitSwitch{Limit Switch is ON}
    LimitSwitch -- Yes --> HomeMove[Home Move inwards at Vzero]
    HomeMove --> HomingMsg[/Homing Message/]
    HomingMsg --> DisplayValues[/Display Values/]
    DisplayValues --> StartupMsg[/Startup Message/]
    StartupMsg --> ReadPot
    SetButton -- No --> DisplayValues
    LimitSwitch -- No --> DisplayValues
    
```

The flowchart illustrates the startup sequence for a CNC machine. It begins with a 'Power On' event, leading to an initialization step for the motor controller, sensors, and LCD communication. The main loop starts with reading potentiometers and updating temperature variables. A decision is made on whether a button has been pressed. If yes, the machine moves outwards at a specific velocity, checks for a limit switch, and if present, moves inwards to home. If no button is pressed or no limit switch is detected, the system proceeds to display startup information, including version, parameters, and a confirmation prompt. The sequence then loops back to reading potentiometers or displays a homing message before returning to the display values step.

Name	Equation	Description
T	$T = 60/BPM$	Time for one complete inhale/exhale cycle
T_{in}	$T_{in} = T/(1 + I\epsilon_{2000})$	Inhalation time
T_{ex}	$T_{ex} = T - T_{in}$	Exhalation time
T_{ac}	$T_{ac} = T$	Acceptable time without breathing during AC mode
V_{in}	$T V_{in}$	Inhalation velocity/rotation rate
V_{ex}	$T V_{ex}$	Exhalation velocity/rotation rate

The diagram illustrates the process of a parameter change. On the left, a light blue parallelogram labeled "Parameter Change Interrupt" has an arrow pointing to a light blue rectangle on the right. The rectangle contains the text: "PARAM CHANGE", "Set ParamChange to", and "In ParamChange".

```
graph TD
    Start([Start]) --> ParamChange{Is ParamChange = TRUE}
    ParamChange -- Yes --> ReadPot[Read Potentiometers]
    ParamChange -- No --> ModeSelect{Mode Selection}
    ModeSelect -- = HIGH --> ParamChange
    ModeSelect -- ModeSelect = LOW --> End([End])
    ReadPot --> Update[Update:  
TV_temp  
BPM_temp  
IE_extemp_temp  
AT_temp]
    Update --> Display[/Display Temp Values/]
    Update --> ReadPot
```

The flowchart illustrates the logic for the temperature control system. It begins with a start node leading to a decision diamond: "Is ParamChange = TRUE". If the answer is "Yes", the flow proceeds to a process rectangle: "Read Potentiometers". This rectangle contains a list of variables to be updated: "Update: TV_temp, BPM_temp, IE_extemp_temp, AT_temp". From this process, the flow goes to an output parallelogram: "Display Temp Values". There is a feedback loop from the "Display Temp Values" output back to the "Read Potentiometers" process. If the initial decision is "No", the flow goes to another decision diamond: "Mode Selection". From "Mode Selection", if the mode is "= HIGH", the flow loops back to the "Is ParamChange = TRUE" decision. If the mode is "ModeSelect = LOW", the flow proceeds to an end node.

ALARM SILENCE: In progress...

ALARM LOGIC: In Progress...

```

graph TD
    Start([Start]) --> ACWait[AC Wait  
Motor position to ZeroPoint  
Set T_elapsed to zero]
    ACWait --> P_mech_check{Is P_mech < P_threshold}
    P_mech_check -- Yes --> ACInHale[AC INHALE  
Set T_elapsed to zero  
Motor position to TV at speed V_in]
    P_mech_check -- No --> T_elapsed_Ac{Is T_elapsed > T_Ac}
    T_elapsed_Ac -- Yes --> AlarmAcuteCycleSkip((Alarm Interrupt  
Acute - Cycle Skip))
    T_elapsed_Ac -- No --> P_mech_check
    ACInHale --> T_elapsed_in{Is T_elapsed >= T_in}
    T_elapsed_in -- Yes --> Pause[PAUSE  
- Wait for T_pause  
- Update P_peak]
    T_elapsed_in -- No --> P_mech_max{Is P_mech >= P_max}
    P_mech_max -- Yes --> AlarmAcutePressure((Alarm Interrupt  
Acute - Pressure))
    P_mech_max -- No --> P_mech_min{Is P_mech <= P_min}
    P_mech_min -- Yes --> VCExhale[VC EXHALE  
Motor position to ZeroPoint at speed V_ex  
Set T_elapsed to zero  
Update P_plat]
    P_mech_min -- No --> UpdateP_peak1[/Update P_peak/]
    VCExhale --> UpdateP_plat[/Update P_plat/]
    UpdateP_plat --> T_elapsed_ex{Is T_elapsed >= T_ex}
    T_elapsed_ex -- Yes --> UpdateP_peak1
    T_elapsed_ex -- No --> UpdateP_mech[/Update P_mech/]
    UpdateP_peak1 --> UpdateP_mech
    UpdateP_mech --> P_perp_max{Is P_mech >= P_perp_max}
    P_perp_max -- Yes --> AlarmApnoeaPEEP((Alarm Interrupt  
Apnoea - PEEP))
    P_perp_max -- No --> P_perp_min{Is P_mech <= P_perp_min}
    P_perp_min -- Yes --> UpdateP_peak1
    P_perp_min -- No --> P_mech_check
  
```

[illegible]