3.1.1 Operating Modes

The Smart Tree can operate in four modes: Normal, Error, Sleep, and Debug. The Normal mode represents daytime operation without errors. The Error mode represents when at least one cell has detected an error condition; in this mode an error is displayed on the screen. The Sleep mode is entered at night; the screen is turned off in this mode. The Debug mode can be activated by a technician to debug or change settings in the Smart Tree.

Instead of representing these 4 modes with a minimal 4 states, we use 3 flags: Debug, Sleep, and Error. Each of these flags is always kept updated, whether it has an immediate effect on the system or not. This way, after debug or sleep is turned off, the proper state can be restored immediately. Internally there are 8 states, but the end result is the original four operating modes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Flags** | | | **Mode** | **Indicators** | |
| **Debug** | **Sleep** | **Error** | **LED1** | **LED2** |
| 0 | 0 | 0 | Normal mode: displays energy and power | 0 | 0 |
| 0 | 0 | 1 | Error mode: displays error | 0 | 1 |
| 0 | 1 | X | Sleep mode; display turned off | 1 | 0 |
| 1 | X | X | Debug mode | 1 | 1 |

The don’t-care flags will continually be updated by the BMS and sleep timer interrupts during the sleep or debug modes, so when they exit, its respective flag can simply be cleared and operation can resume immediately in the proper state, without requiring additional measurements.

Rationale

The rationale for this slightly more complicated model is because of the decision making required when exiting the sleep and debug states. Exiting the sleep mode requires analysis of the error status of each cell to enter the correct mode; exiting the debug state requires analysis of both the sleep timer and the error status. To minimize bugs and maximize performance, it is preferable to perform this analysis in one location of the code, controlled by the interrupts as in normal operation. In addition, it is preferable to perform this analysis in constant intervals of time, to facilitate calculation of energy and power. We solve both of these issues by saving the sleep and error flags even when they do not affect the current mode, so that the proper mode to enter is already determined upon exit.

Error mode (requires a error message)

1. undervoltage for any output
2. overcurrent for any output
3. overtemperature for any output

**Error mode: still whenever there’s an error**

1. **undervoltage**
   1. **if not all of them are undervoltage, draw small box, clear with a rectangle**
   2. **if all of them are undervoltage draw large box, redraw screen**
2. **overcurrent**
   1. **large box warning**
   2. **redraw screen**
3. **overtemperature:**
   1. **large box warning**
   2. **redraw screen**