CUTE Details

*********** Changes are highlighted in RED ***********

CARETAKER Mode

CUTE must be on a horizontal surface when powered on for the first time. Upon powering on, CUTE must have the following CARETAKER mode behaviors:

- The following message to NAME is sent once each time CARETAKER mode is entered:

  Entering CARETAKER mode\r\n

- The SEGMENT_DISPLAY is off
- The GRAPHICS_DISPLAY is off
- BLINK_RED is off
- BLINK_BLUE is off
- The TEMPERATURE_SENSOR, LIGHT_SENSOR and ACCELEROMETER are idle and not reading any data
- There are no transmissions to NAME

If SW_MONITOR is pressed during the CARETAKER mode, CUTE goes into MONITOR mode.

MONITOR Mode

MONITOR mode for CUTE represents the situation when the latter is being deployed in a real environment for monitoring of the elderly person's environment. As soon as CUTE enters MONITOR mode, the TEMPERATURE_SENSOR, LIGHT_SENSOR and ACCELEROMETER are sampled. The following behavior occurs in MONITOR mode:

- The following message to NAME is sent once each time MONITOR mode is entered:

  Entering MONITOR mode\r\n

- The GRAPHICS_DISPLAY must include the following word throughout MONITOR mode: MONITOR
- The SEGMENT_DISPLAY hexadecimal value increases by 1 for every second that passes by. After the SEGMENT_DISPLAY shows the last value, it restarts with the first value of the hexadecimal system. The sequence is as indicated below:

  0 1 2 3 4 5 6 7 8 9 A B C D E F

- The TEMPERATURE_SENSOR, LIGHT_SENSOR, and ACCELEROMETER are sampled and the values are updated on the GRAPHICS_DISPLAY when the SEGMENT_DISPLAY shows:

  5 A F

- Transmission of the currently sampled sensor values from the TEMPERATURE_SENSOR, LIGHT_SENSOR, and ACCELEROMETER, to NAME, occurs once each time the SEGMENT_DISPLAY shows:

  F

- The format of the transmitted data should be as follows:

  NNN_-_T*****_L*****_AX*****_AY*****_AZ*****\r\n

  where the temperature value (accurate up to 1 decimal place), light value, and x-axis value, y-axis value, z-axis value of the accelerometer[#] are respectively preceded by T, L, AX, AY, and AZ. * is an optional numerical value, a decimal point, or a minus sign. The special characters \r\n indicates that the next data transmission has a carriage return and is on a new line. NNN represents a 3-digits value that starts from 000 and increments by 001 each time such set of sensor data is transmitted to NAME from CUTE. NNN never resets itself to 000, unless CUTE itself is powered on from a power off state. It is assumed that 999 will never be reached.

- Any time between two data transmissions to NAME, if any 'movement' is detected by the ACCELEROMETER[$] and the sample value from LIGHT_SENSOR is below LIGHT_LOW_WARNING, BLINK_BLUE should happen to indicate that the elderly person is moving in a dark environment. If the sample value from TEMPERATURE_SENSOR is above TEMP_HIGH_WARNING, BLINK_RED should happen to indicate that fire is detected.
- BLINK_BLUE and BLINK_RED can happen at the same time. Both must be synchronized (i.e., they should either alternate or be turned on and off simultaneously) if they occur at the same time. BLINK_BLUE and BLINK_RED should not be turned off, unless CUTE goes into CARETAKER mode.

- If BLINK_BLUE is happening at the instant a scheduled transmission to NAME occurs, the following message must also be sent before the usual sensor values from the TEMPERATURE_SENSOR, LIGHT_SENSOR, and ACCELEROMETER:

  Movement in darkness detected\r\n

- If BLINK_RED is happening at the instant a scheduled transmission to NAME is happening, the following message must also be sent before the usual sensor values from the TEMPERATURE_SENSOR, LIGHT_SENSOR, and ACCELEROMETER:

  Fire detected\r\n

- Messages due to BLINK_RED occurs before messages due to BLINK_BLUE.

If SW_CARETAKER is pressed **twice within 1 second** during MONITOR mode, CUTE returns to CARETAKER mode, simulating the presence of a caretaker. The double-press requirement will minimize accidental return to CARETAKER mode when the patient is alone. The following message is sent to NAME before returning to the CARETAKER mode.

Leaving MONITOR mode\r\n

Specifications

- LIGHT_LOW_WARNING: 50 Lux
- TEMP_HIGH_WARNING: 45 degrees Celsius
- $The way ACCELEROMETER is used to detect 'movement' is left to students' discretion
- #You can choose the unit for displaying accelerometer value (raw value or m/s^2 or g's). However, you should know how to convert from one unit to another if required.

Devices Used

- ACCELEROMETER: MMA7455L
- LIGHT_SENSOR: ISL29003 with LIGHT_RANGE_4000
- TEMPERATURE_SENSOR: MAX6576
- SEGMENT_DISPLAY: 7-segment LED display
- GRAPHICS_DISPLAY: 96x64 White OLED
- BLINK_BLUE: Blue Light of RGB LED, alternating between ON and OFF every 500 milliseconds
- BLINK_RED: Red Light of RGB LED, alternating between ON and OFF every 500 milliseconds
- SW_CARETAKER: SW3, read using interrupts
- SW_MONITOR: SW4
- NAME: UART terminal program (Tera Term) on a personal computer

General Clarifications and Hints

- For a project of this nature, it is impossible to have the specifications cover all possible scenarios. We leave it to your discretion to decide how the system should respond in scenarios not covered by the above specifications.
- The exact messages/format displayed on the GRAPHICS_DISPLAY/ UART isn't important. The format and messages are just suggestions. Feel free to make *reasonable* modifications.
- *You can also make *reasonable* adjustments to thresholds. The thresholds we have given are just indicative.
- Light sensor/accelerometer/temperature readings using interrupts is **optional** and will be evaluated only as an enhancement (see Enhancements page for more info); it is **not a basic requirement.**
- To blink the LEDs in a timely fashion described in BLINK_BLUE and BLINK_RED, the only practical option is to do it in a handler (either SysTick or an on-chip timer) **or** have a main program which is very fast such that it can handle the blinking in a precise manner (which is not easy to implement, and will certainly need a lot of optimizations in the main program which will include reading sensors using interrupts, among many other things).
- It is fine to use FreeRTOS if you wish to (no support will be provided).