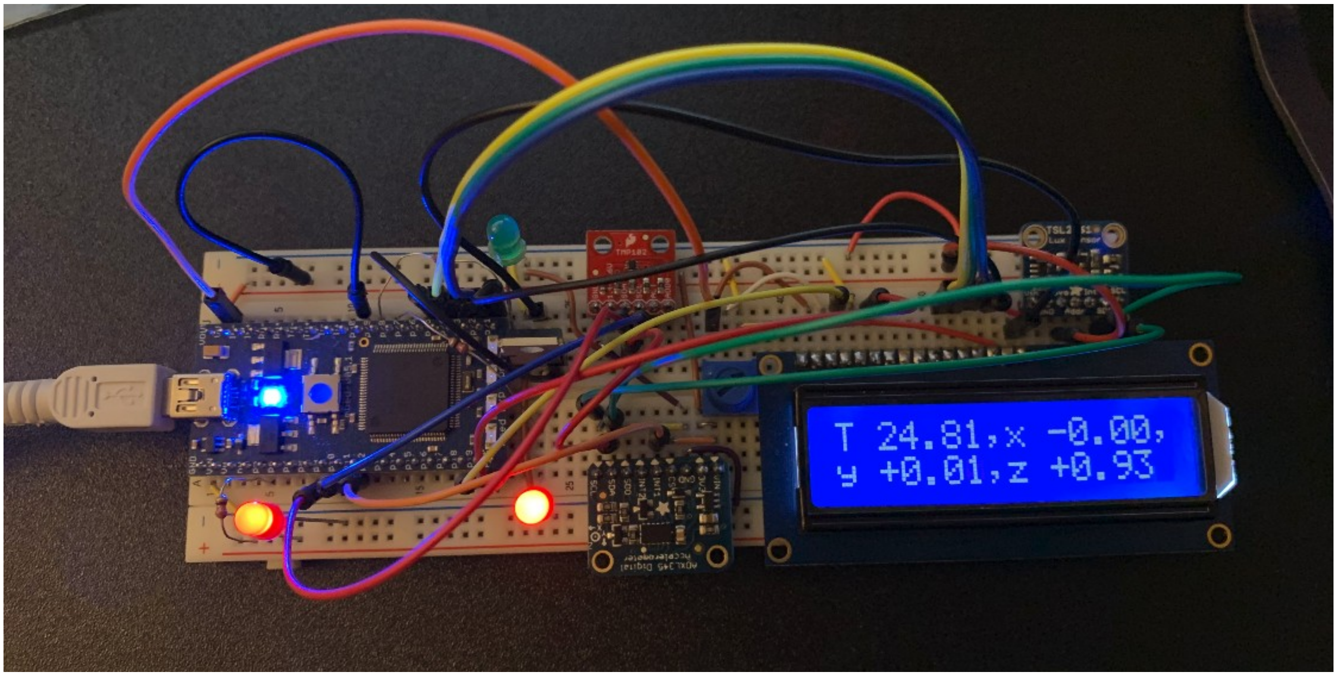
Embedded Device Power Management Project

Cheng Fu

High Level Overview:

I deigned a consumer electronic device to measure living condition.
The system measures living conditions and display temperature on Text LCD display.
Significant power saving features are added to allow device to last a longer time on battery.
Sensor devices in use are accelerometer ADXL345, temperature sensor TMP102, ALS (Ambient Light Sensor) TSL2561.

Project code: https://github.com/cheng-fu/EmbeddedFirmwareEssential_Final_Project



Input Output:

Use I2C to connect multiple sensor peripherals (ALS, TMP102 and accel).
Text LCD display with parallel port.
Use one GPIO to drive PWM through a transistor (AOT2618L N-channel MOSFET) to control display back light brightness.
One GPIO is used to capture button press interrupt, one LED is toggled per interrupt.
Accel event interrupt is configured to detect shock and tap, another LED is toggled per interrupt.
One LED to indicate system power status, on vs sleep.

Power Management:

Display Brightness control:

Display usually consumes 40% of the system power, so auto-brightness feature was added to

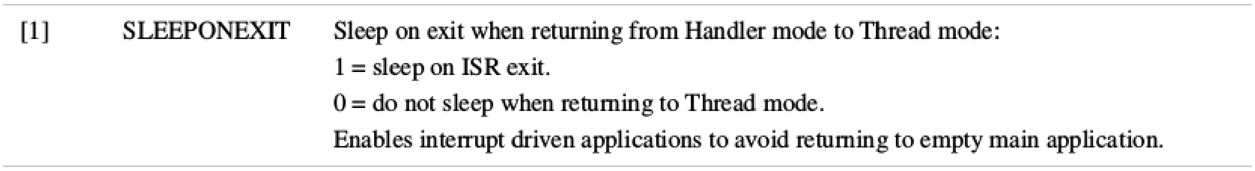
reserve power.
The project uses ALS sample to determine screen brightness level. 3 piece linear auto brightness curve is used to map 15 Lux ~ 1500 Lux input to 0.1 – 1.0 PWM duty cycle output. This is an approximation to the logarithmic human eye response to a broad range of lighting conditions.
If a desire dutycyle change is beyond 0.03, the project uses a series of time out callbacks to smooth the brightness change (0.04 second * 8 or 16 steps) to make user feel the display gradually becomes brighter/dimmer.

System Power Management policy:
Whenever LCD display is on, poll temperature and ALS sensor value every 400 ms.
All sensors are turned off except accel, which uses very low power.
Whenever system is up, read accelerometer sample and determine device orientation, face up or down.

Transition: Mode 1 (Normal mode) → Mode 2 (display off, system on) → Mode 3 (System sleep SLEEP ON EXIT)
After 20 seconds inactivity (no interrupt events, no change of orientation) the system turns off (Mode 1 → Mode 2)
After another 25 seconds inactivity, automatically enter sleep mode (set SLEEP ON EXIT bit) to reserve the most power. (Mode 2 → Mode 3)

Timeouts are used to perform mode transitions, button debounce and accel back to back interrupt suppression.

The system enter sleep from main loop, when idle condition is met.
In sleep mode, the CPU does not run any user code, only ISRs are executed.



[1]	SLEEPONEXIT	Sleep on exit when returning from Handler mode to Thread mode: 1 = sleep on ISR exit. 0 = do not sleep when returning to Thread mode. Enables interrupt driven applications to avoid returning to empty main application.

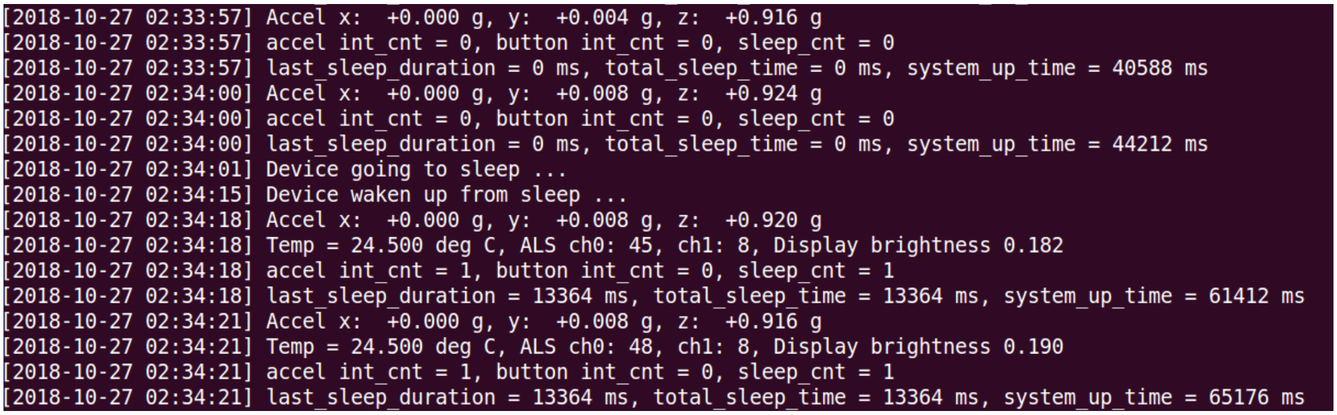
SLEEP ON EXIT explanation on Cortex M3 Technical Reference Manual

On interrupt (accel event detection or button GPIO state change), display turned back on, system exits from sleep (clear SLEEP ON EXIT bit).

Special mode: Mode 4 (Device facing down, system up)
If anytime during (Mode 1), accelerometer detects device facing down, power off the display (enter Mode 2 but with was_facing_down flag set).
If device stay in this Mode 4 for 25 seconds, enter system sleep mode (Mode 3).
If within 25 seconds, device turned back facing up, enter Mode 1.
If within 25 seconds, interrupts occurs, if device is still facing down, igore, otherwise enter Mode 1.

Profiling system power:

Profiling statistics collected: number of sleep cycles the system has gone through, time spent when system is off.

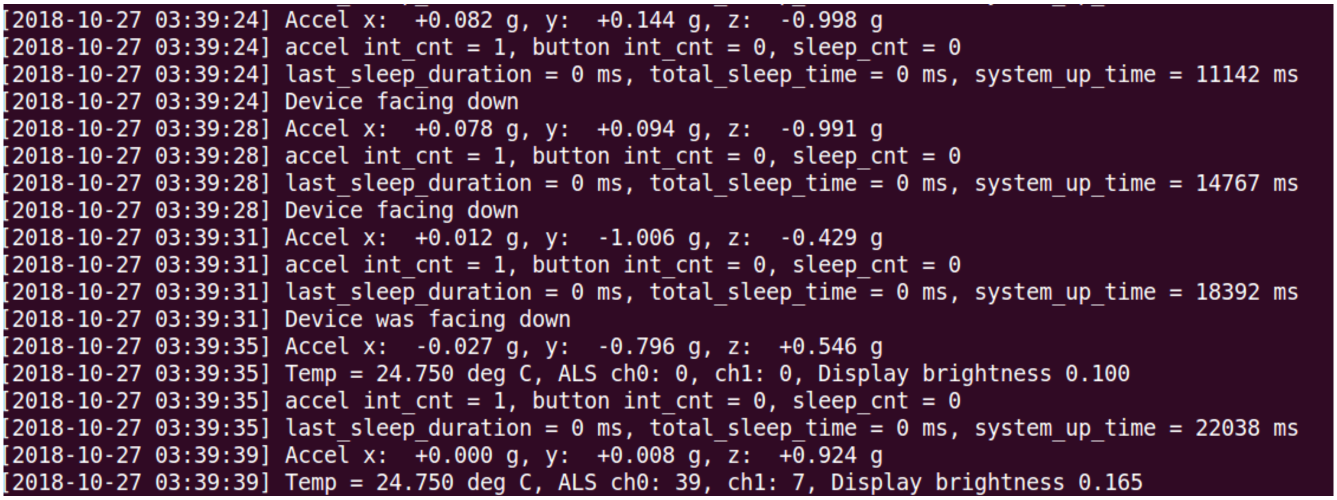


```
[2018-10-27 02:33:57] Accel x:  +0.000 g, y:  +0.004 g, z:  +0.916 g
[2018-10-27 02:33:57] accel int_cnt = 0, button int_cnt = 0, sleep_cnt = 0
[2018-10-27 02:33:57] last_sleep_duration = 0 ms, total_sleep_time = 0 ms, system_up_time = 40588 ms
[2018-10-27 02:34:00] Accel x:  +0.000 g, y:  +0.008 g, z:  +0.924 g
[2018-10-27 02:34:00] accel int_cnt = 0, button int_cnt = 0, sleep_cnt = 0
[2018-10-27 02:34:00] last_sleep_duration = 0 ms, total_sleep_time = 0 ms, system_up_time = 44212 ms
[2018-10-27 02:34:01] Device going to sleep ...
[2018-10-27 02:34:15] Device waken up from sleep ...
[2018-10-27 02:34:18] Accel x:  +0.000 g, y:  +0.008 g, z:  +0.920 g
[2018-10-27 02:34:18] Temp = 24.500 deg C, ALS ch0: 45, ch1: 8, Display brightness 0.182
[2018-10-27 02:34:18] accel int_cnt = 1, button int_cnt = 0, sleep_cnt = 1
[2018-10-27 02:34:18] last_sleep_duration = 13364 ms, total_sleep_time = 13364 ms, system_up_time = 61412 ms
[2018-10-27 02:34:21] Accel x:  +0.000 g, y:  +0.008 g, z:  +0.916 g
[2018-10-27 02:34:21] Temp = 24.500 deg C, ALS ch0: 48, ch1: 8, Display brightness 0.190
[2018-10-27 02:34:21] accel int_cnt = 1, button int_cnt = 0, sleep_cnt = 1
[2018-10-27 02:34:21] last_sleep_duration = 13364 ms, total_sleep_time = 13364 ms, system_up_time = 65176 ms
```

Example Serial logging with tapping accel to wake (with power profiling info)

Logging:

Compiler flags were used to control debug loggings for each module.
Main program has one more flag to control verbosity of profling info, sensor data logging.



```
[2018-10-27 03:39:24] Accel x:  +0.082 g, y:  +0.144 g, z:  -0.998 g
[2018-10-27 03:39:24] accel int_cnt = 1, button int_cnt = 0, sleep_cnt = 0
[2018-10-27 03:39:24] last_sleep_duration = 0 ms, total_sleep_time = 0 ms, system_up_time = 11142 ms
[2018-10-27 03:39:24] Device facing down
[2018-10-27 03:39:28] Accel x:  +0.078 g, y:  +0.094 g, z:  -0.991 g
[2018-10-27 03:39:28] accel int_cnt = 1, button int_cnt = 0, sleep_cnt = 0
[2018-10-27 03:39:28] last_sleep_duration = 0 ms, total_sleep_time = 0 ms, system_up_time = 14767 ms
[2018-10-27 03:39:28] Device facing down
[2018-10-27 03:39:31] Accel x:  +0.012 g, y:  -1.006 g, z:  -0.429 g
[2018-10-27 03:39:31] accel int_cnt = 1, button int_cnt = 0, sleep_cnt = 0
[2018-10-27 03:39:31] last_sleep_duration = 0 ms, total_sleep_time = 0 ms, system_up_time = 18392 ms
[2018-10-27 03:39:31] Device was facing down
[2018-10-27 03:39:35] Accel x:  -0.027 g, y:  -0.796 g, z:  +0.546 g
[2018-10-27 03:39:35] Temp = 24.750 deg C, ALS ch0: 0, ch1: 0, Display brightness 0.100
[2018-10-27 03:39:35] accel int_cnt = 1, button int_cnt = 0, sleep_cnt = 0
[2018-10-27 03:39:35] last_sleep_duration = 0 ms, total_sleep_time = 0 ms, system_up_time = 22038 ms
[2018-10-27 03:39:39] Accel x:  +0.000 g, y:  +0.008 g, z:  +0.924 g
[2018-10-27 03:39:39] Temp = 24.750 deg C, ALS ch0: 39, ch1: 7, Display brightness 0.165
```

Example Serial logging with flipping device from facing down to facing up to wake display.
(ALS sample, display brightness are only printed when display on)

Future Work:

Hock up humidity sensor,
Calibrate accelerometer,
Analysis of motion patterns to do lift to wake, raise to wake,
More OOP, create sensor base class and do inheritance.
Handle system timer rollover.