EMBEDDED SYSTEM DESIGN
PROJECT 3: THIS SIDE UP
ENERGY OPTIMIZATION (V1.0)

OVERVIEW

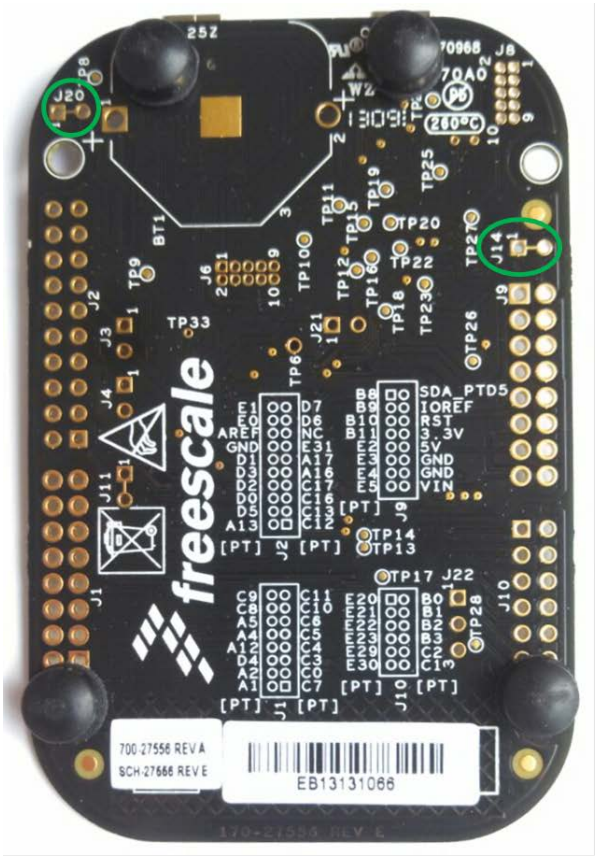
For this project you will create a device which monitors its orientation. If it tilts too far from horizontal it will flash the LED red or yellow to warn the user. Otherwise the LED flashes green. Your goal is to minimize its energy use, maximizing battery life.

DETAILS

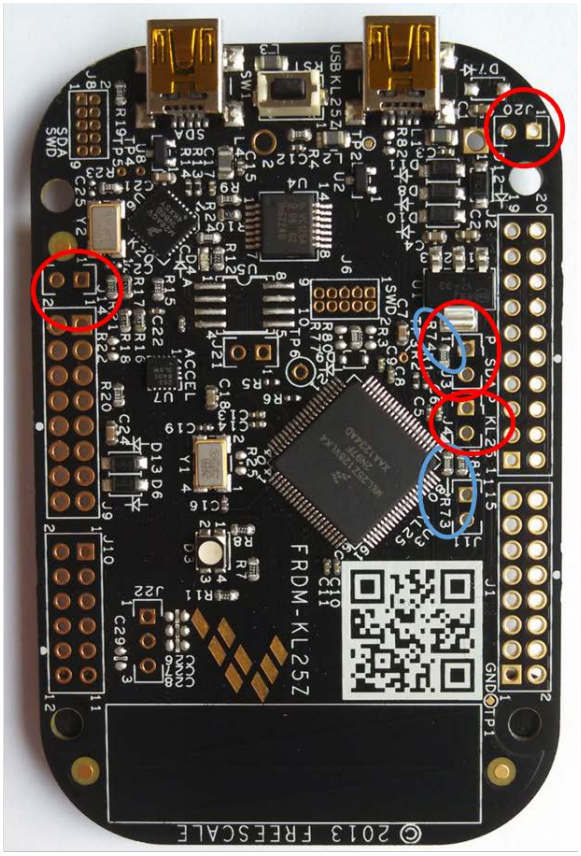
SYSTEM REQUIREMENTS

- Start with the KL25Z Freedom board and make modifications as needed.
- Determine the orientation every 100 ms using the MMA8451 accelerometer. The X, Y and Z accelerations can be converted to roll and tilt angles (in degrees) using these equations:
  - roll = atan2(a_y, a_z)*180/π
  - pitch = atan2(a_x, (a_y^2 + a_z^2)$^{1/2}$)*180/ π
- Flash the RGB LED at a frequency of 2 Hz to provide visual status information on the orientation of the system.
  - Color
    - Green: OK, <15 degrees
    - Yellow: Warning, >= 15 degrees
    - Red: Failure, >= 30 degrees
  - Timing
    - The LED must be on for 10 ms every 500 ms.

HARDWARE MODIFICATIONS



Cut 2 traces



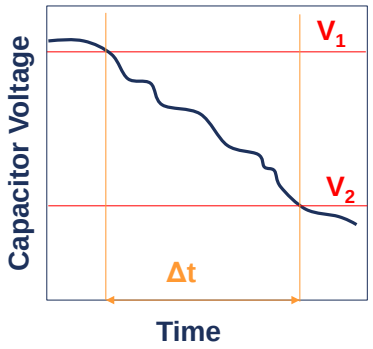
Remove 2 resistors
Add 4 headers

Modify your Freedom board as follows:

1. Remove resistor R73 to enable target MCU current measurement. The MCU's current will flow through a 10 ohm resistor (R81), producing a voltage drop of 10 times the current.
2. Add 2 pin header at location J4. Measure this voltage drop at location J4 (labeled P_KL25) and divide it by 10 to determine MCU current. *Note that putting an ammeter across J4 will give an inaccurate reading because the ammeter's internal shunt resistor will be in parallel with R81, reducing the effective resistance, voltage drop and current reading.*
3. Remove resistor R74 to enable debug circuit current measurement.
4. Populate J3 (labeled P_SDA) with a 2 pin header. Insert a shorting jumper on J3 when you wish to use the debug MCU (e.g. debugging, or downloading new code).
5. Cut the trace on the back of J14 to disconnect the target MCU's reset line from the debug MCU. Populate J14 with a 2 pin header. Insert a shorting jumper on J14 when you wish to use the debug MCU.
6. Cut the trace on the back of J20 to allow measurement of output current from the U1 linear voltage regulator. Optionally populate J20 with a 2 pin header. Do not place a shorting jumper on J20, as diode D12 is needed to prevent current from flowing from the ultracapacitor back to the voltage regulator.

ENERGY MEASUREMENT

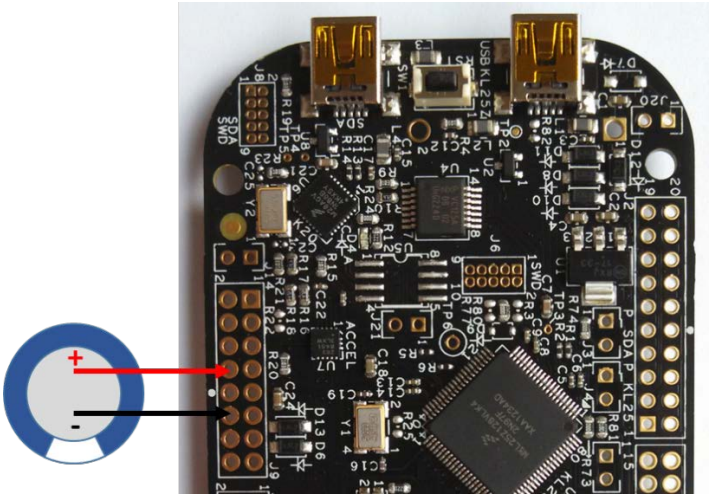
Mount a 0.1 F capacitor on the P3V3 rail, charge it and use it to power the system. The amount of time the system runs will indicate the total energy used, enabling you to calculate the average power used.



$$Energy = C\frac{V_1^2 - V_2^2}{2}$$

$$Average\ Power = C\frac{V_1^2 - V_2^2}{2t}$$

Insert the ultracapacitor on the P3V3 rail, connecting the negative pin to ground (J9 pin 12) and the positive pin to P3V3 (J9 pin 8).



RECOMMENDED APPROACH AND HINTS

- Planning
  - Work through the energy measurement lab exercise to understand the basic concepts, and use that code as a starting point.
  - Build a power and energy model in a spreadsheet to help guide your design and optimization. You should plan how to optimize the system's energy use before you starting coding, and use the model as you proceed with your optimization.
- Software structure
  - Use the low-power timer LPTMR to wake the processor periodically.
- Disabling unused components
  - Disconnect the debugger (power and reset line) when running your timing tests.
  - Disable the clock signals to unused MCU peripherals using the SIM_CG registers.

- Using low-power modes
  - Operate devices (peripherals and MCU) in their low power modes when possible.
- Minimizing active times
  - Use your optimized accelerometer roll/pitch calculation code as a starting point.
  - Eliminate extra computations.
  - You are allowed to use PWM to light the LEDs.
- Run the I^2C communication link as fast as possible.
  - Use the MMA8451Q's built-in configurable orientation detection hardware, detailed in Freescale application note AN4068.
- Debugging
  - When using the debugger (MDK), you may need to disable the __wfi() calls.
  - Use an oscilloscope to monitor active times for the processor and inertial sensor. Use one or more digital output bits as debug signals for visibility into the system.
  - Optimize the code to reduce execution cycles required.

DELIVERABLES

- Electronic Submission
  - Report with explanation of your optimization process, containing text and a power model for each optimization step.
  - Project archive with source code.
- Demonstration
  - Demonstrate system operation to a TA using a 0.1 F ultracapacitor to calculate average power use based on discharge time.
    - Connect the USB cable to the SDA interface
    - Incline the board so the LED flashes red
    - Plug the ultracapacitor into the P3V3 rail, allowing it to charge for two minutes.
    - Verify that headers J13 and J14 do not have shorting jumpers installed.
    - Disconnect the USB cable. The system is now powered by the ultracapacitor.
    - Measure how long it takes until the LED stops flashing. **You should be able to achieve a run time of at least 5 minutes. To receive full credit for the performance portion of the grade, you will need to achieve a run time of 8 minutes.**