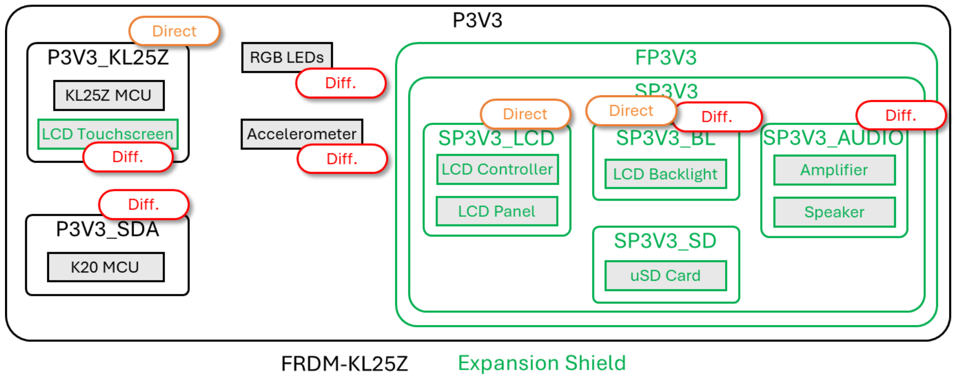
ECE 461/561
TAKE-HOME PORTION OF FINAL EXAM
(WAS HOMEWORK 4)
ANALYZING POWER AND ENERGY (V1.1)

CONTENTS

OVERVIEW

Portions of this assignment which are highlighted in cyan will be finalized in V1.1. Do not start working on them until you have V1.1. Only * starred sections have questions to answer.

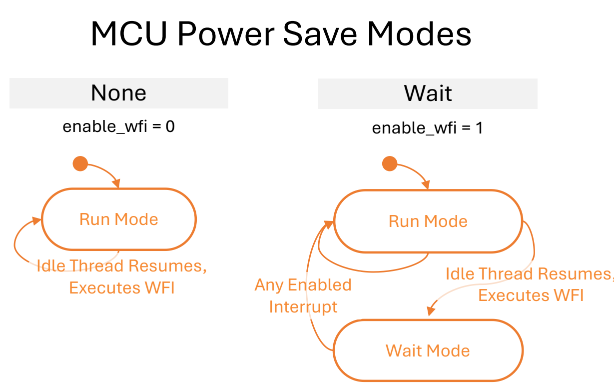
The goal of this assignment is to evaluate the power consumption of a demonstration program on the FRDM-KL25Z and expansion shield. The demonstration program uses various peripherals on the MCU, FRDM board and expansion shield. It has features to simplify power and energy measurement and testing.

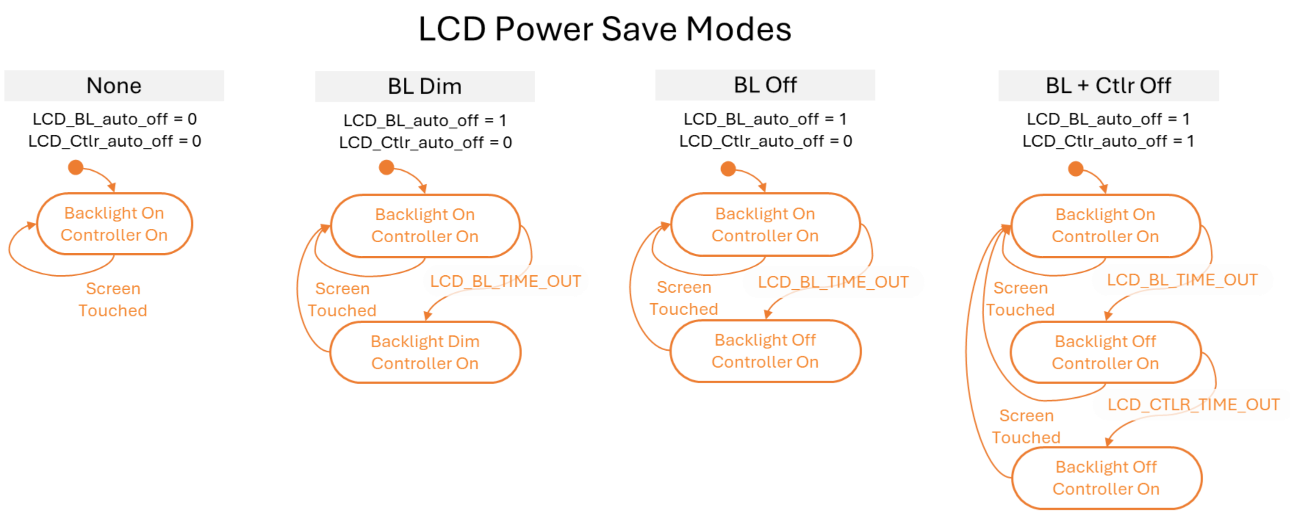


The power system architecture for the FRDM-KL25Z and expansion shield is shown in this figure. Some components are grouped into subsystems with separate power domains. Some of these domains are powered through a removable jumper, allowing them to be disconnected completely. Other domains are powered through a shunt resistor so we can determine its current by measuring the voltage drop across its shunt resistor. You will measure current use with a combination of **direct measurement** (of voltages across shunt resistors) and **differential analysis** (comparing measurements with different configurations of enabled components).

The test program's user interface lets you select different power modes for parts of the system. Some parts have a simple behavior where this **mode** directly defines its **operating state**: audio generation is on or off, or the accelerometer is active or standby.

Other parts of the system may have more than one state per mode to allow automatic power control. For example, the MCU power save mode of "Wait" will configure the idle thread's loop to include code to put the MCU to sleep (the wfi instruction). If the scheduler has no application tasks ready to run, it will resume the idle thread, which will execute the wfi instruction, putting the MCU into one of its wait modes. Any enabled interrupt will wake up the processor and put it back in run mode.





Similarly, the LCD's "BL Dim" mode has two states, as shown in the diagram above. Initially the backlight is on at full brightness, but if it hasn't been touched before a time-out expires, the backlight will dim (lower duty cycle) to save power.

SUBMISSION INFORMATION

You are to work **individually** on this assignment. Submit your work and responses through Moodle as follows **BEFORE THE END OF THE FINAL EXAM TIME SLOT FOR THE CLASS**:

- Spreadsheet file:
  - Open [this spreadsheet](https://docs.google.com/spreadsheets/d/1KSSVv2Qa5N0al338ITCfj2cOtopaJipC1TisSbU4x1g/edit?usp=sharing): https://docs.google.com/spreadsheets/d/1KSSVv2Qa5N0al338ITCfj2cOtopaJipC1TisSbU4x1g/edit?usp=sharing
  - Copy the spreadsheet: **File -> Make a Copy**
  - Use the green and yellow cells in your spreadsheet copy to enter your responses to this document's numbered questions:
    - Green cells are for numerical measurements. Formatting information is in the spreadsheet.
    - Yellow cells are for free-form text (which may include numbers but no tabs).
  - Download your completed spreadsheet as a tsv (tab-separated values) text file: **File -> Download -> Tab-Separated Values (.tsv).**
  - Rename the .tsv file with your email username (part before @ncsu.edu): e.g. agdean-HW4.tsv
  - Upload this renamed .tsv file to Moodle.

ECE 461 students can receive extra credit for completing sections marked **561 Only**.

*MCU POWER MODELING

Assume that the KL25Z MCU we use has these characteristics. Note that the MCU current **does not** depend on the supply voltage V_{DD}.

- In normal run mode, I_{MCU} = 1.86 mA + f_{Clock} * 82 µA/MHz, for 1.8 V ≤ V_{DD} ≤ 3.6 V and 0 MHz ≤ f_{Clock} ≤ 48 MHz
- In VLP (very low power) run mode, I_{MCU} = 0.16 mA + f_{Clock} * 33 µA/MHz, for 1.8 V ≤ V_{DD} ≤ 3.6 V and 0 MHz ≤ f_{Clock} ≤ 4 MHz

Our test program has these characteristics:

- The CPU runs the function f_work() once every 20 ms, and the function takes 10000 clock cycles to run each time.
- The CPU runs an idle loop the rest of the time.

*FREQUENCY SCALING

Consider the MCU operating at 30 MHz in Run mode and V_{DD} = 3.0 V. Under these conditions,

1. What is the MCU's current consumption (in mA)?
2. What is the MCU's power consumption (in mW)?
3. How much time does it take for the CPU to execute function f_work() once (in ms)?
4. How much energy is used by the MCU to execute function f_work() once (in nJ)?

Consider the MCU operating at 3 MHz in VLP Run mode and V_{DD} = 3.0 V.

5. What is the MCU's current consumption (in mA)?
6. What is the MCU's power consumption (in mW)?
7. How much time does it take for the CPU to execute function f_work() once (in ms)?
8. How much energy is used by the MCU to execute function f_work() once (in nJ)?

*FREQUENCY AND VOLTAGE SCALING

Consider the same conditions as in the Frequency Scaling section, but with the operating voltage V_{DD} now reduced to 2.0 V.

With the MCU operating at 30 MHz in Run mode and V_{DD} = 2.0 V,

9. What is the MCU's current consumption (in mA)?
10. What is the MCU's power consumption (in mW)?
11. How much time does it take for the CPU to execute function f_work() once (in ms)?
12. How much energy is used by the MCU to execute function f_work() once (in nJ)?

With the MCU operating at 3 MHz in VLP Run mode and V_{DD} = 2.0 V,

13. What is the MCU's current consumption (in mA)?
14. What is the MCU's power consumption (in mW)?
15. How much time does it take for the CPU to execute function f_work() once (in ms)?
16. How much energy is used by the MCU to execute function f_work() once (in nJ)?

*LETTING THE CPU SLEEP

Next we want to change the design to use the CPU's low power and sleep features. To simplify using very low power (**VLP**) run and wait modes, we change the clock and oscillator configurations to use internal oscillators. As a result, the **current consumption changes** to the values shown below.

The supply voltage V_{DD} is **2.0 V**. Again, the MCU current **does not** depend on the supply voltage V_{DD} (1.8 V ≤ V_{DD} ≤ 3.6 V).

- **Normal** modes:
  - Configuration
    - FLL Engaged Internal (FEI) uses Slow Internal Reference Clock -> FLL -> OUTDIV1
    - f_{Clock_High} = (21, 24, 42, 48 MHz) / [1, 2, …, 15, 16]

- Current consumption
  - In normal **run** mode, I_{MCU} = 2636.58 µA + f_{Clock_High} * 96.95 µA/MHz
  - In normal **wait** mode, I_{MCU} = 2627.13 µA + f_{Clock_High} * 44.82 µA/MHz
- **VLP** (very low power) modes:
  - Configuration
    - Bypassed Low-Power Internal uses Fast Internal Reference Clock -> FCRCDIV -> OUTDIV1
    - f_{Clock_Low} = 4 MHz / ([1, 2, 4, 8, … 64, 128] * [1, 2, 3, …, 15, 16])
  - Current consumption
    - In **VLP run** mode, I_{MCU} = 244.87 µA + f_{Clock_Low} * 72.07 µA/MHz
    - In **VLP wait** mode, I_{MCU} = 234.78 µA + f_{Clock_Low} * 30.38 µA/MHz

Order	**1**	**2**	**3**	**4**
Phase	**Work**	**Enter Sleep**	**Sleep**	**Exit Sleep**
CPU Execution Mode	Run	Run	Wait (no code execution)	Run
Code Executing	f_work()	Sleep entry code	None	Sleep exit code
CPU Frequency	f_{Clock_High}	f_{Clock_Low}	f_{Clock_Low}	f_{Clock_Low}
CPU Power Mode	Normal	VLP	VLP	VLP

The program has four phases (shown above) which execute in order and then repeat. They have these characteristics:

- **Work**: The CPU runs the function f_work() once, taking 10000 clock cycles. All of this code runs at the f_{Clock_High}.
- **Enter sleep**: After f_work() completes, the CPU executes sleep entry code, taking 21 µs + 496 CPU cycles. Assume that all of this code runs at the f_{Clock_Low}. This code:
  - reconfigures clocks and switches the CPU to f_{Clock_Low},
  - switches to the VLP (very-low-power) run mode,
  - enters the wait/sleep mode using a WFI instruction.
- **Sleep:** The CPU is in wait mode, so it will not execute any instructions until an enabled interrupt occurs.
- **Exit Sleep:** A timer interrupt wakes up the MCU every 20 ms, upon which the CPU executes sleep exit code, taking 42 µs + 332 CPU cycles. Assume that all of this code runs at the f_{Clock_Low}. This code:
  - switches to the normal run mode,
  - reconfigures clocks and switches the CPU to f_{Clock_High}.

Consider the design points in this table. For each design point,

17. Calculate the time to execute f_work() (ms)
18. Calculate the time to enter sleep (ms)
19. Calculate the time to exit sleep (ms)
20. Calculate each phase's energy consumption (nJ)
21. Calculate the total energy consumption (nJ) for completing all four phases once
22. Identify which phase has the highest power consumption
23. Identify which phase has the highest energy consumption

Design Point	**f_{Clock_High} (MHz)**	**f_{Clock_Low} (MHz)**
A	**48**	**4**
B	**3**	**1**
C	**48**	**1**
D (561 Only)	**3**	**4**
E (561 Only)	**48**	**0.0625**

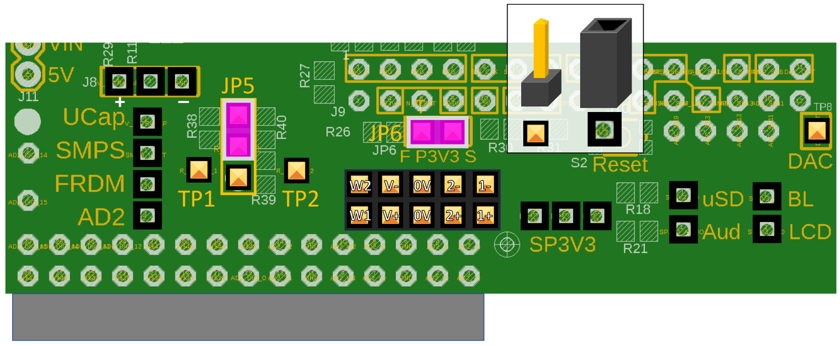
Which design point has:

24. The lowest total energy consumption?
25. The shortest time to exit sleep?

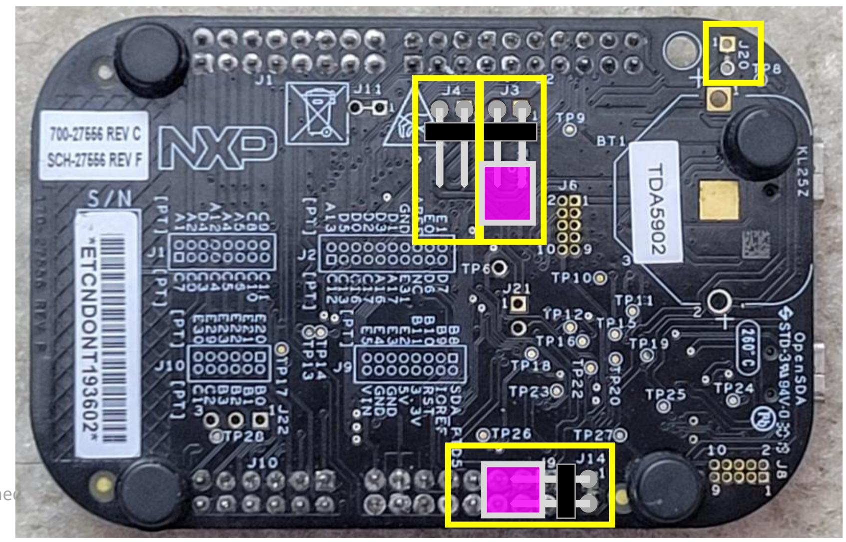
PLATFORM SET-UP

BASICS

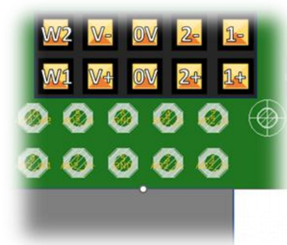
HARDWARE



- Confirm that the shield has these pins and sockets mounted:
  - Pins
    - Single: TP1, TP2, DAC
    - Headers: JP5, JP6 (“F P3V3 S”), with a shorting jumper block on each
  - Sockets
    - Single: uSD, Aud, BL, LCD
    - Headers: UCap/SMPS/FRDM/AD2, SP3V3 (TP3)
- Confirm the FRDM-KL25Z has these components and changes:
  - 2 pin headers at positions J3, J4 and J14. Some of these may be on the top of the FRDM PCB.
  - Shorting jumper blocks on headers J3 and J14
  - Trace cut at J20, or header at J20 with no shorting jumper block
- You will need several jumper wires:
  - 1 pin-pin
  - 5 pin-socket
  - 4 socket-socket



- Please refer to this pin-out information for additional AD2/3 conne



Set up the code as follows:

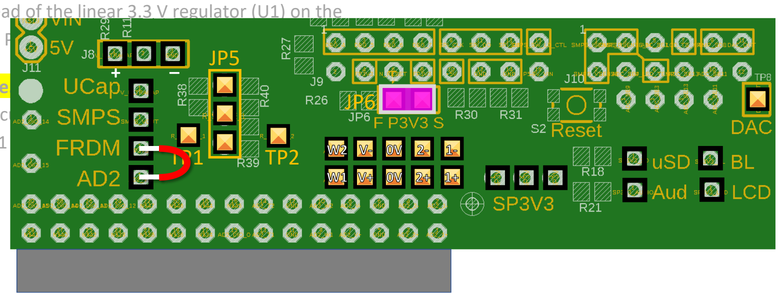
- Download the project code for the assignment from the class repository’s HW4 directory.
- Confirm that you can build the program without errors. Note that we are **not using the profiler** here, so you only need to build the program once after a code change, not three times.

SOFTWARE

- Download the project code for the assignment from the class repository’s HW4 directory.

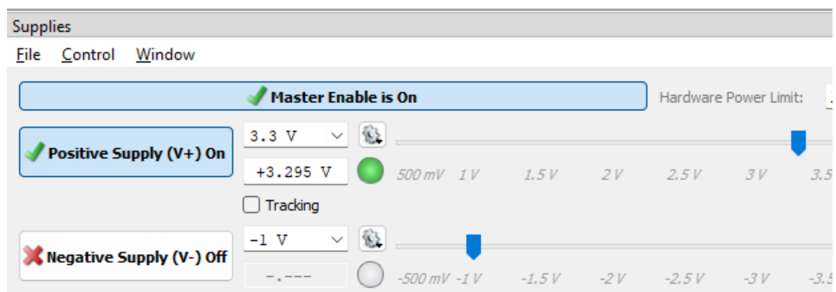
POWER SUPPLY CONFIGURATION

We will start by using the AD2's positive adjustable voltage supply V+ (instead of the linear 3.3 V regulator (U1) on the FRDM board) to power the boards. This will let us adjust the voltage on the F



- **Remove USB power from the FRDM-KL25Z by unplugging its USB cable**
- Confirm that the J20 shorting trace on the back of the FRDM has been c there is no shorting jumper mounted on it. This will confirm that the U1 through diode D12.
- Power the two boards from the AD2's V+ output (**AD2**). Use a jumper wire to connect **AD2** to **FRDM** (the FRDM P3V3 supply rail) as shown in the diagram.
- Make sure that a jumper is mounted on JP6 to power the shield (SP3V3) from the FRDM's P3V3 supply (FP3V3).

Follow these steps with the Waveforms program:

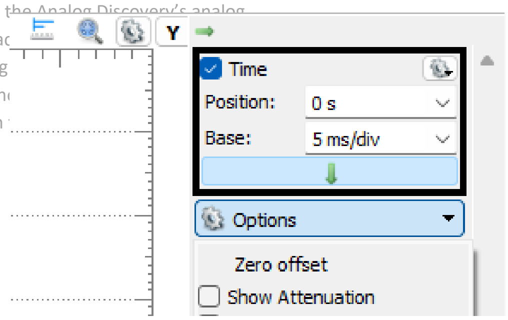


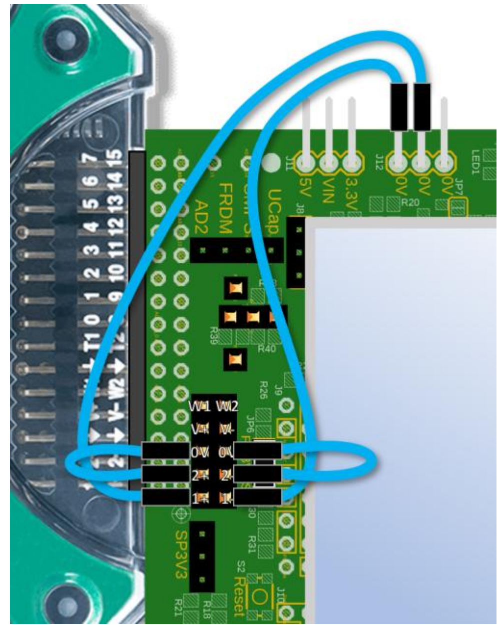
- Select the power supplies through Welcome + -> Supplies. Set the positive power supply to to **3.3V** as shown below. **NEVER SET THIS TO MORE THAN 3.6 V – THE MAXIMUM OPERATING VOLTAGE RATING FOR THE MCU.**
- Confirm that Positive Supply (V+) is either On or Rdy. For safety, switch the Negative Supply (V-) to Off.
- Enable the power supply by pressing the Master Enable button, changing it from Off to On. This should cause the FRDM and shield to power up and run normally. If it doesn't, please check your work.

OPTIONAL: ZERO OUT ANALOG INPUT OFFSETS

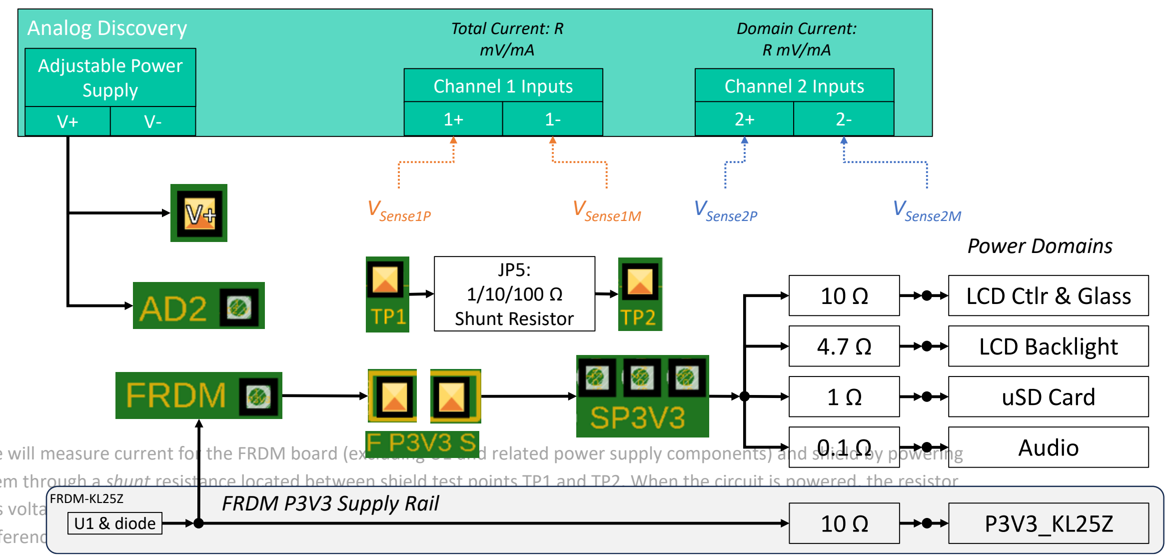
There may be a slight voltage offset error for the Analog Discovery's analog input channels, and this can affect the accurac remove these offsets, connect all four analog ground (0V) using the jumper wires and/or sh are multiple ground available on

Then follow the instructions in Waveforms Options-> **Zero offset**.





POWER MONITOR CONFIGURATION



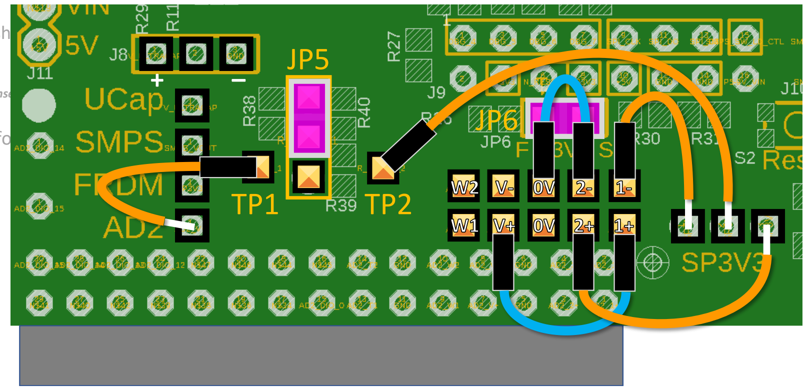
We will measure current for the FRDM board (ex related power supply components) and shield by powering them through a *shunt* resistance located between shield test points TP1 and TP2. When the circuit is powered, the resistor has volta differenc I_{Sense}*$V_{Sense1M}$.

We will measure the voltages at these test points using AD2's analog inputs 1+ (for TP1) and 1- (for TP2).

- We measure the voltage $V_{SenseDiff}$ directly across TP1 ($V_{Sense1P}$) and TP2 ($V_{Sense1M}$): $V_{SenseDiff} = V_{Sense1P} - V_{Sense1M}$.

- The resistance R_{Sense} will be set to 1 Ω with the JP5 jumper mounted as sh
- Dividing $V_{SenseDiff}$ by R_{Sense} gives the current I_{Sense}.
- We can compute power drawn by the boards (excluding R_{Sense}) as $P = I_{Sens}$

Make the following changes and additions to the power supply circuit and Wavefo



- ➢ Turn off the power supply by changing Master Enable to Off.
- ➢ Remove the wire between **AD2** and **FRDM** to eliminate the original power connection.
- ➢ Insert the sense resistance in the power path (**AD2** -> **TP1**, **TP2** -> **SP3V3**)
  - Connect **AD2** to **TP1** with a wire.
  - Connect **TP2** to **SP3V3** with a wire.
  - Place a shorting jumper on **JP5** in the 1 Ω position (closer to center of board).
- ➢ Although Channel 1 is measuring V_{Sense2} as the lower part of its differential input, we cannot get its value directly. Instead, we will measure it with Channel 2+, by connecting **2+** to **SP3V3** with a wire. JP6 connects FP3V3 to SP3V3, so Channel 2+ now measures the supply voltage V_{SP3V3} which is the same as V_{FP3V3}.
- ➢ Connect Channel 2- to ground (0V).
- ➢ Open the Voltmeter tool.
- ➢ Click on the Gear icon next to the Voltmeter's Stop button and reduce the update period from 1 s to 100 ms.
- ➢ Turn on the power supply by changing Master Enable to On.

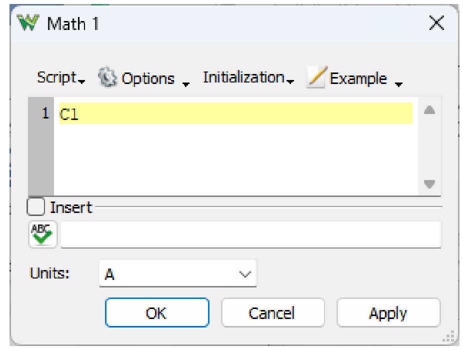
OVERALL SYSTEM ANALYSIS

BASIC ANALYSIS

Let's start off by measuring the current and power used by the entire system. Use the voltmeter to determine the following parameters for the entire system. Make sure Audio is Off, RGB Mode is Pause, Accel is On, and both LCD Psv and MCU Psv are None.

26. What is the average current drawn (Channel 1 DC value/R_{Sense}) in mA? It should be between 100 and 200 mA.
27. What is the average voltage V_{SP3V3} at V_{Sense2} (Channel 2 DC value)?
28. What is the average power used (Channel 2 DC value * Channel 1 DC value/ R_{Sense}) in mW?

*CURRENT OVER TIME



We will use the oscilloscope to evaluate current consumption over time at 3.3 V across a variety of program conditions.

- Adjust the Waveforms positive power supply so that V_{SP3V3} = 3.300 V.

Follow these steps in Waveforms to set up the scope and add a math channel to compute and display the current through R_{Sense}:

- Open the Scope tool.
- Turn on the power supply by changing Master Enable to On. The FRDM and shield should turn on.
- In the Scope window, select Add Channel -> Custom, then edit the code to return the value C1 as shown. Change the units to A (amps) and select OK.
- Press the gear icon for the math channel. Then change the channel's name to Current and label to I1.
- Enable the Measurements window and add average, minimum, and maximum currents for I1. Use the Measurements gear icon to enable measurements over multiple captures.
- Set the horizontal (time) sensitivity to 20 ms/division, so the screen shows 200 ms total.

*C1: BASELINE CONFIGURATION

Configuration Name	ID	Touchscreen State	RGB Mode	RGB State	Accel. Mode	LCD Powersave Mode	LCD BL State	LCD Ctlr State	Audio Mode	OpenSDA Mode	MCU Powersave Mode	MCU Powersave State
Overall System Analysis												
Baseline	C1	Untouched	Pause	All Off	Active	None	On	On	Off	Connected	None	Run

Use the touchscreen's user interface to stabilize the test conditions as configuration **C1** (**Baseline**) in the table above. For example, the RGB LEDs should all be off and not cycling.

- Adjust the Waveforms positive power supply so that V_{SP3V3} = 3.30 V (or as close as possible).

29. What is the voltage coming out of the AD2 supply (V_{Sense1})? Do not change this voltage for the rest of this assignment.

- In the Waveforms program, in Scope1's File menu, select Load Acquisition and load the file **C1.dwf3scopeacq**. Current 1 (magenta) is the total system current, and Current 2 (light blue) is the KL25Z MCU current (currently unused).

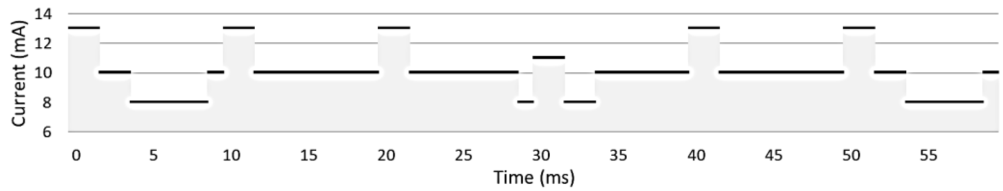
32. Take a screenshot showing the current for 200 ms (20 ms/division) and put it in your report. **This screenshot must include the acquisition's time stamp and device serial number.**

Use Waveforms to determine:

30. The minimum, average and maximum current in mA (with one digit after the decimal point).
31. What is the average power, given that the current is drawn at nominal 3.3 V?

*561 ONLY: MEASURING TEMPORAL CURRENT VARIATIONS

There should be some periodic behavior in the current consumption. For example, this diagram shows current consumption with a



baseline of 10 mA and two periodic components: an extra 3 mA are drawn for 1 ms every 10 ms (100 Hz), and current drops by 2 mA for 5 ms every 25 ms (40 Hz).

33. Determine the current baseline (mA). Assume that it is the average current from the baseline configuration above.

Determine the current change (+ or - mA), pulse width (ms) and period (ms) for the component with…

34. The largest current change.
35. The second largest current change

36. The th
➢ Keep t
time. '

ECE 561 ONLY: OPERATING VOLTAGE RANGE

Let's see what the minimum operating voltage is for different parts of the system. Use the Waveforms supply to lower the voltage on the P3V3 rail (V_{SP3V3}) a little at a time down to 1.8 V. Besides typing in the numerical voltage you want in upper drop-down box, you can also nudge the voltage up or down:

- … by 10 mV at a time by clicking left (-=10 mV) or right (+=10 mV)of the blue slider pointer.
- … by 1 mV at a time using keyboard arrow keys (← or ↓ -= 1 mV) (→ or ↑ += 1 mV) if slider control is selected.

Recall that that the system is running at a voltage ($V_{SP3V3} = V_{Sense2}$) which is lower than the output of the AD2 positive power supply (V_{Sense1}) because of the voltage drop across R_{Sense}.

Please perform these tests with consistent ambient lighting conditions. Report these voltages with two digits after the decimal point (e.g. round to the nearest 10 mV).

37. At what voltage V_{SP3V3} does the LCD become too dark to read?

Raise V_{SP3V3} so the LCD is visible again. Use screen control to set RGB mode to Cycle, so the LED cycles through different colors. Determine if each of the LEDs below is still visible at V_{SP3V3} = 1.8V. If it is not, report the largest value of V_{SP3V3} where the LED is not visible:

38. Blue LED.
39. Green LED.
40. Red LED.

The datasheet lists the MCU's minimum supply voltage (V_{DD}) for operation as 1.71 V. Will it run that low? One of the LEDs still works at this voltage, so while that LED flashes the MCU is running.

Note that if the MCU's supply voltage falls below V_{POR} (or upon power up), the MCU will stay in reset until it rises above V_{LV}

ECE 561 ONLY: CURRENT AND POWER VS. OPERATING VOLTAGE

to

of reset and running again. If the LCD is all white, reset the processor so it re-initializes the LCD controller.

41. At what voltage on V_{SP3V3} does the MCU stop running?

Let's see how much current and power the system uses over a range of supply voltages. How much current and power are used at different voltages? Some parts of the system will draw more current and power at higher voltages. However, the MCU has an internal voltage regulator, stabilizing the current across voltage levels.

LCD BACKLIGHT ON

- Make sure Audio is Off, RGB Mode is Pause, Accel is On, and both LCD Psv and MCU Psv are None. To make measurement easier, we'll stabilize the test conditions by disabling all RGB LEDs (no cycling).
- On the LCD, press RGB label as needed so the RGB LED is not lit.

42. Use the Waveforms Supplies tool to adjust the Positive Supply voltage output to the settings shown in this table. You will need to iteratively adjust the positive supply's output voltage to get the specified V_{SP3V3} value. Then measure the current and compute the power for each. The currents should range from roughly 200 mA down to under 40 mA.
43. Create a scatter plot of the current and power vs. V_{SP3V3} and include it in your report.

V_{SP3V3} (± 10 mV)	**Current (mA)**	**Power (mW)**
3.3 V		
3.0 V		
2.7 V		
2.4 V		
2.1 V		
1.8 V		

LCD BACKLIGHT OFF

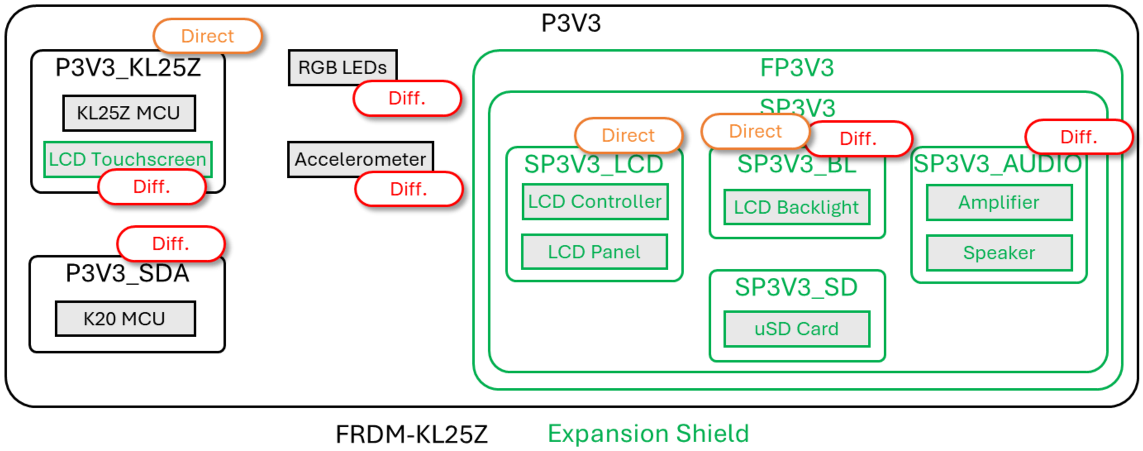
LEDs draw exponentially more current as voltage increases. The LCD backlight consists of several white LEDs. We will disable the backlight and repeat the current measurements and power calculations.

- Make sure Audio is Off, RGB Mode is Pause, Accel is On, and MCU Psv is None.
- Again, stabilize the test conditions by disabling all RGB LEDs (no cycling).
- Change LCD Psv to BL Off and wait until the LCD Backlight turns off before starting your measurements.
- On the LCD, press RGB label as needed so the RGB LED is not lit.

44. Use the Waveforms Supplies tool to adjust the Positive Supply voltage output to the settings shown in this table. You will need to iteratively adjust the positive supply's output voltage to get the specified V_{SP3V3} value. Then measure the current and compute the power for each.
45. Create a scatter plot of the current and power vs. V_{SP3V3} and include it in your report.

V_{SP3V3} (± 10 mV)	**Current (mA)**	**Power (mW)**
3.3 V		
3.0 V		
2.7 V		
2.4 V		
2.1 V		
1.8 V		

*BREAKING OUT SUBSYSTEM CONSUMPTION



The power domains of the FRDM-KL25Z and shield are summarized in this diagram. Now we can start breaking down the current and power into smaller pieces. We will use two approaches:

- Differential power analysis ("Diff." in diagram) of easily controlled devices: Some devices can be controlled by software (e.g. an LED connected to a port output). We can configure the device to operate in one state (e.g. all LEDs off), measure the total current and power, move to a different state (e.g. red LED on), measure the total current and power, and then determine the differences.
- Direct measurement of power domains: The FRDM-KL25Z and the shield have multiple power domains. We can measure the domain's current and voltage to determine the power it uses.

We will use the following table as a roadmap to guide our analysis. Key differences are emphasized with text in color.

Configuration Name	Configuration ID	Touchscreen State	RGB Mode	RGB State	Accel. Mode	LCD Powersave Mode	LCD Powersave BL State	LCD Powersave Ctlr State	Audio Mode	OpenSDA Mode	MCU Powersave Mode	MCU Powersave State	
Overall System Analysis													
Baseline	C1	Untouched	Pause	All Off	Active	None	On	On	Off	Connected	None	Run	1
Differential Analysis													
Touchscreen	C2	**Touched**	Pause	All Off	Active	None	On	On	Off	Connected	None	Run	2
LEDs	C3	Untouched	Pause	**Red**	Active	None	On	On	Off	Connected	None	Run	3
LEDs	C4	Untouched	Pause	**Green**	Active	None	On	On	Off	Connected	None	Run	4
LEDs	C5	Untouched	Pause	**Blue**	Active	None	On	On	Off	Connected	None	Run	5
LEDs	C6	Untouched	Pause	**All On**	Active	None	On	On	Off	Connected	None	Run	6
Accelerometer	C7	Untouched	Pause	All Off	**Standby**	None	On	On	Off	Connected	None	Run	7
LCD Backlight	C8	Untouched	Pause	All Off	Active	**BL Dim**	**Dim**	On	Off	Connected	None	Run	8
LCD Backlight	C9	Untouched	Pause	All Off	Active	**BL Off**	**Off**	On	Off	Connected	None	Run	9
LCD Backlight	C10	Untouched	Pause	All Off	Active	**BL + Ctlr Off**	**Off**	**Off**	Off	Connected	None	Run	10
Audio Generation	C11	Untouched	Pause	All Off	Active	None	On	On	**On**	Connected	None	Run	11
OpenSDA Debug MCU	C12	Untouched	Pause	All Off	Active	None	On	On	Off	**Disconnected**	None	Run	12
Direct Measurement													
LCD Backlight	C13	Untouched	Pause	All Off	Active	**BL Dim**	**Dim**	On	Off	Connected	None	Run	13
LCD Backlight	C14	Untouched	Pause	All Off	Active	**BL Off**	**Off**	On	Off	Connected	None	Run	14
LCD Controller and Driver	C15	Untouched	Pause	All Off	Active	**BL + Ctlr Off**	**Off**	On	Off	Connected	None	Run	15
LCD Controller and Driver	C16	Untouched	Pause	All Off	Active	**BL + Ctlr Off**	**Off**	**Off**	Off	Connected	None	Run	16
Audio Generation	C17	Untouched	Pause	All Off	Active	None	On	On	**On**	Connected	None	Run	17
KL25Z	C18	Untouched	Pause	All Off	Active	None	On	On	Off	Connected	**None**	**Run**	18
KL25Z	C19	Untouched	Pause	All Off	Active	None	On	On	Off	Connected	**Wait**	**Idle**	19
KL25Z	C20	Untouched	Pause	All Off	Active	None	On	On	Off	Connected	**VLPW**	**Idle**	20

*DIFFERENTIAL ANALYSIS

- Adjust the Waveforms positive power supply so that V_{SP3V3} = 3.30 V (or as close as possible). Then leave that supply voltage unchanged for the remainder of this assignment.

C2: TOUCHSCREEN

- Slowly press and release the lower portion of the LCD touchscreen where there are no controls.

46. How much average current and power does the **touchscreen** use when pressed?

C3-C6: LEDS

The program can light the RGB LED in the repeating sequence off, red, green, blue, white (R+G+B) using a periodic call to the function LED_Callback().

- Use the RGB screen control to control the LED. Pause at each color to measure current consumption.

47. How much average current and power does it take to **light the LED with the color**:
    a. Red?
    b. Green?
    c. Blue?
    d. White?

*C8-C10: LCD BACKLIGHT

- Use the screen control to select the appropriate LCD power save modes.

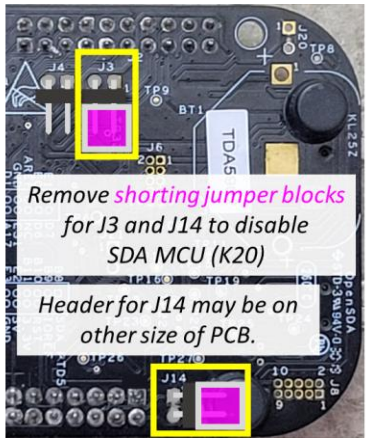
How much average current and power does the backlight use in each of these conditions, described as LCD power save mode|backlight state? Select the appropriate mode and then wait for the backlight time-out period (e.g. 3 seconds). **Use the Backlight Off case (BL Off, C9.dwf3scopeacq) as to determine current and power without the backlight. Then determine how much *additional* current and power are used the other cases (e.g. Dim vs. BL Off, None vs. BL Off).**

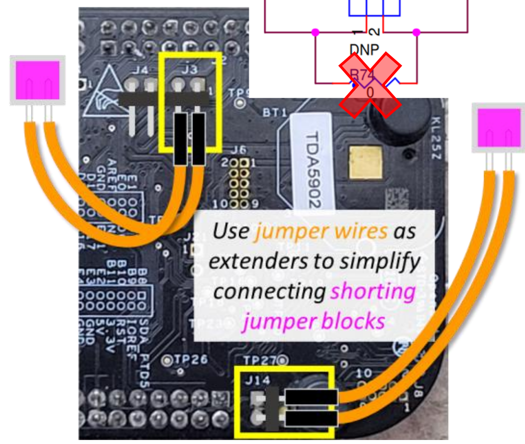
48. How much current and power does the system use when the backlight is off? BL Off| Backlight off. Use acquisition **C9.dwf3scopeacq**.
49. None|Backlight always on (duty cycle = LCD_BL_BR_FULL = 100%). Load acquisition **C1.dwf3scopeacq**, measure the current and calculate the power, and subtract your current and power numbers from question 48.
50. Dim|Backlight lit at reduced duty cycle (LCD_BL_BR_DIM). Load acquisition **C8.dwf3scopeacq**, measure the current and calculate the power, and subtract your current and power numbers from question 48.

C12: OPENSDA DEBUG MCU

The P3V3 supply rail is connected to P3V3_SDA (the OpenSDA debug MCU's supply rail) through J3 and R74, as shown in the diagram. Note that resistor R74 has been removed from the FRDM board so J3 alone controls the power to P3V3_SDA.

We will disable the debug MCU to measure the impact on the system's current. We also need to remove the jumper from J14, which lets the Debug MCU (K20) reset the Target MCU (KL25Z).





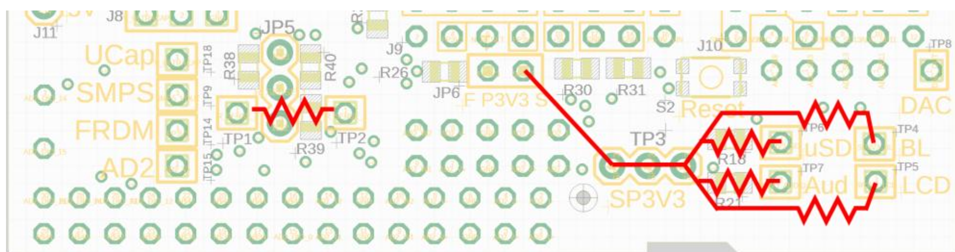
- Remove shorting jumpers from J3 and J14, disconnecting the debug MCU's power (J3) and reset output (J14) signal.
- Measure the total system current and calculate the difference from the previous reading.

51. How much additional current and power does the debug MCU use when enabled?

Remember to place the jumpers back on J3 and J14 when you need to reprogram or debug the KL25Z.

*DIRECT MEASUREMENT OF SHIELD SUBSYSTEMS

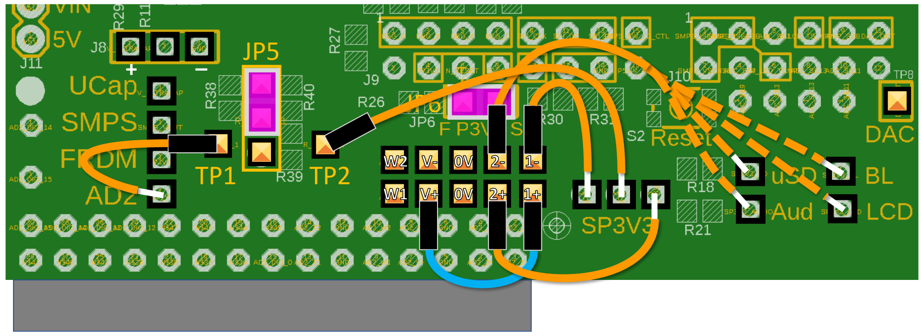
The FRDM-KL25Z and shield circuits group some components into subsystems with separate power domains. Some domains (e.g. P3V3_SDA above) are powered through a removable jumper rather than a shunt resistor. Other domains are powered through a shunt resistor so we can determine a domain's current by measuring the voltage drop across its shunt resistor. Each subsystem has a different sized shunt resistor due to different typical current draws, so the mA per mV factor will depend on the subsystem, as shown in this table.



Subsystem	Test Point Label	Resistor	Resistance	Sensitivity
uSD – uSD Card	uSD	R18	1 Ω	1 mV/mA
Aud – Audio amplifier and speaker	Aud	R21	0.1 Ω	0.1 mV/mA
BL – Backlight for LCD	BL	R1	4.7 Ω	4.7 mV/mA
LCD – LCD controller and driver	LCD	R19	10 Ω	10 mV/mA

We will continue to use C1 to monitor the total current drawn by system, but now we will use C2 to measure one domain at a time. Change the wiring as shown in the diagram:

- Change the AD2 analog input C2- to measure different shunt resistors. C2+ is already connected to SP3V3, which is connected to the high side of the sense resistors (R18, R21, R1, R19). Connect C2- to one of the four subsystems at a time.



*C15-C16: LCD: LCD CONTROLLER AND DRIVER

In this mode, the code turns off the LCD backlight after a time-out (LCD_BL_TIME_OUT, typically two seconds), and then turns off the LCD controller after another time-out (LCD_CTLR_TIME_OUT, typically 2 more seconds).

- We are now measuring across a 10 Ω resistor, so you may add a math channel to display the current as C2/10.
- Use acquisition **C15-C16.dwf3scopeacq** to answer these questions. Channel Current 2 (light blue) shows the LCD Controller and driver current.

52. C15: What are the minimum, average and maximum current use for the LCD controller and driver use when the Powersave is set to BL + Ctlr Off, the LCD backlight is off, but the controller is still on?
53. C16: What are the minimum, average and maximum current use for the LCD controller and driver use when LCD Powersave is set to BL + Ctlr Off and both the backlight and controller are off?
54. ECE 561-Only: Compare the LCD from configuration C15 current changes over time with those from your Baseline screenshot. Which of the Baseline current changes (if any) did this subsystem cause?

OPTIONAL - C17: AUD: AUDIO

We have left the audio domain for last because it draws so much current that the 1 Ω resistor between TP1 and TP2 drops enough voltage to dim the display and affect audio quality.

- For this measurement only, bypass the 1 Ω resistor by removing the jumper from JP5 and connecting SP3V3 to the center terminal for JP5.
- Add a math channel to display the current as C2/0.1.

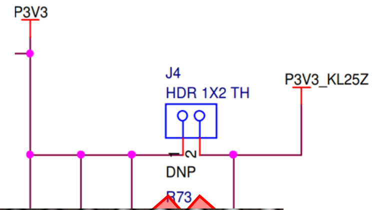
55. What are the minimum, average and maximum current use for the audio?

- Disable the audio.
- Restore the previous current monitoring connections.

*DIRECT MEASUREMENT OF FRDM-KL25Z SUBSYSTEMS

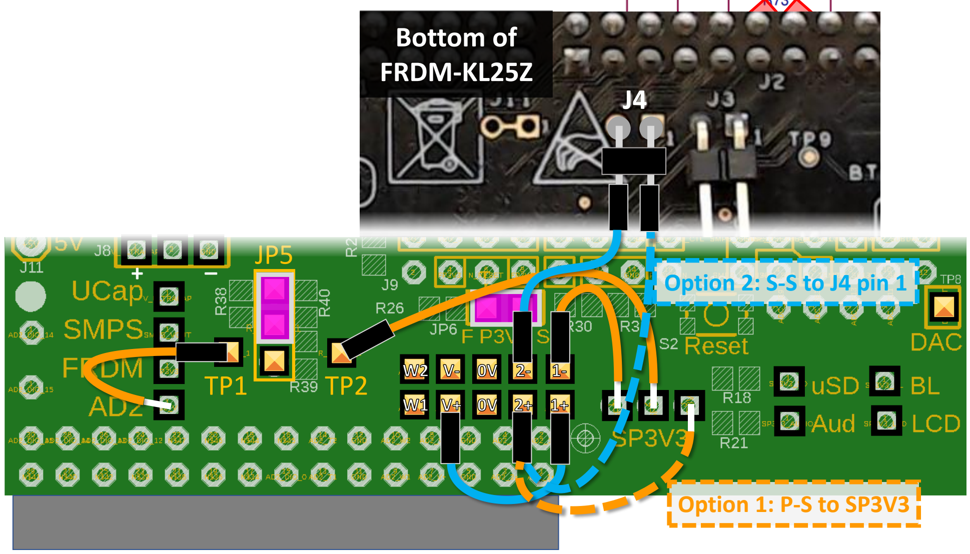
*C18-20: KL25Z MCU

The KL25Z MCU is powered by the P3V3_KL25Z supply rail, which is powered from the P3V3 rail through a 10 Ω shunt resistor R81 as shown in the diagram. Note that resistor R73 has been removed from the FRDM board so R81 is not shorted out. The voltage across R81 can be measured through header J4, which is accessible on the bottom of the FRDM-KL25Z board.



Because J4 pin 1 is connected to P3V3 (through SP3V3), there are two ways to monitor the voltage across R81. Either approach will work but they change the type of one jumper wire (option 1: pin-to-socket, option2: socket-to-socket). Connect the AD2 analog inputs 2+ to J4-1, and 2- to J4-2 as shown in the diagram.

Note that the LCD's touchscreen is powered by GPIO port bits, so pressing the touchscreen raises the MCU current.



- Remember that we are measuring across a 10 Ω resistor (R81), so the current is C2/(10 Ω).
- Set the MCU **Powersave** mode to **None**. In this mode, the idle thread loop keeps the CPU in the Run state.
- Use acquisition **C18.dwf3scopeacq**. Current 1 (magenta) is the total system current, and Current 2 (light blue) is the KL25Z MCU current.

56. C18: What are the minimum, average, and maximum current use for the KL25Z MCU?
57. C18: What are the minimum, average, and maximum power use for the KL25Z MCU?
58. Take a screenshot showing the current for 200 ms (20 ms/division) and put it in your report. **This screenshot must include the acquisition's time stamp and device serial number.**
59. ECE 561-Only: Compare the MCU current changes over time with those from the Baseline trace. Which of the Baseline current changes (if any) did the MCU cause?

- Set the MCU **Powersave** mode to **Wait**. In this mode, when the idle thread loop runs it puts the CPU into the Wait low-power state. The next interrupt puts the CPU back into the Run state.
- Use acquisition **C19.dwf3scopeacq**. Current 1 (magenta) is the total system current, and Current 2 (light blue) is the KL25Z MCU current.

60. C19: What are the minimum, average, and maximum current use for the KL25Z MCU?
61. C19: What are the minimum, average, and maximum power use for the KL25Z MCU?