AN14934

Ultra-Low-Power Rotation Sensing using L-C Sensors and MCX L255

Rev. 2.0 — 13 July 2026 Application note

Document information

Information	Content
Keywords	AN14934, L-C sensing, MCX L255, FlexLC, rotation detection, flow meter.
Abstract	This application note describes an ultra-low-power rotational sensing solution using inductive L-C sensors integrated with the NXP MCX L255 microcontroller. The document outlines the operation principles of L-C sensing, the FlexLC hardware block, and the system architecture enabling accurate detection of slow and fast rotational movements in battery-powered flow-meter applications. It also details the software architecture, timing control, calibration procedures, and low-power operation strategies that enable continuous sensing while maintaining minimal energy consumption.

1 Introduction

The NXP MCXL25x devices integrate a dual-core microcontroller expanding to Arm Cortex-M33 (CM33) and Arm Cortex-M0+ (CM0+) into a single package. The controller supports dual power domains. The real-time main power domain, with ARM CM33 with clock speed of up to 96 MHz, high-speed peripherals, and advanced security blocks gives the efficiency to execute compute centric HMI, security and communication tasks with the help of real-time operating systems, or in bare-metal application software. The ultralow power AON domain with always-ON (AON) peripherals gives options to interface a range of sensors with limited computational requirement fulfilled by ARM CM0+ core, which can operate up to 10MHz clock speed. NXP supports MCXL25x controller device with tools and software that include hardware evaluation boards, software development IDE, example applications, and drivers. Use case-specific Application notes documents are available to complement chip Reference manual and datasheet documents. General and Application specific uses cases software are supported through Application Code Hub (ACH) and are extensions developed with NXPs MCUXpresso SDK bare-metal device drivers, middleware and so on.

Flow sensors can be interfaced through a range of modules in the MCXL25x MCU, such as QTMRs, LPTMR, LPADC, ACMP, LPCMP, FlexLC, GPIOs, or their combinations. These modules can remain active even when the device enters deep Power-down mode, significantly reducing overall power consumption to ultralow levels.

This application note demonstrates how to build a low-power encoder using inductive (LC) sensors. The objective is to design an ultra-low-power solution suitable for half-metal disc flow meters, enabling accurate measurement of disc rotation at low rotational speeds.

Using LC sensors for rotational movement detection is a common practice in hybrid flow meter designs. In these systems, the rotation of the paddle wheel is measured by an LC sensor that detects the presence of conductive material on a disc attached to the wheel. Electronic paddle wheel flow meters (hybrid flow meters) are typically battery-operated, making current consumption a critical aspect of the design.

2 Working principle of LC sensors

A key element of the encoder is contactless sensing of conductive material by using LC circuit. When the LC circuit is excited, it starts to oscillate. The oscillation decays at a rate that is determined by the attenuation given by the value of the inherent parasitic resistance of the circuit and the damping factor caused by eddy current generated in the nearby conductive material.

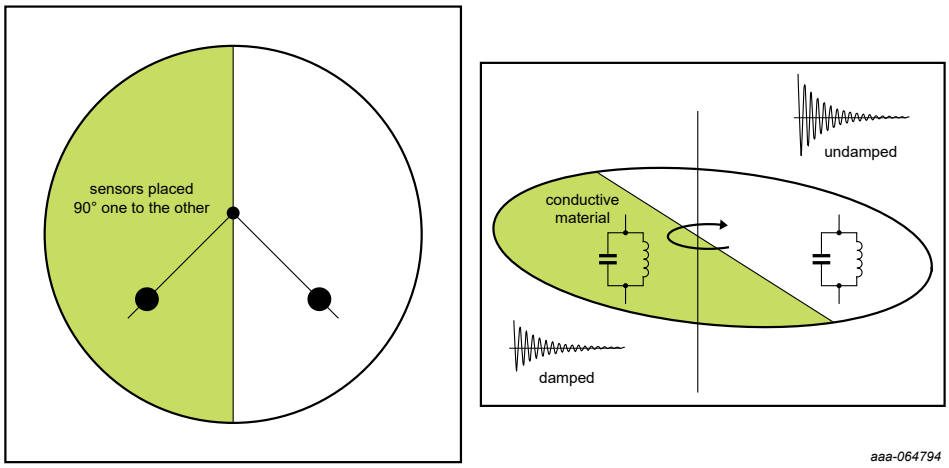


Figure 1. The two-LC sensor principle at work

Two (or three) coils are placed near the rotation disc. The rotation disc is partially covered with a conductive material and forms the quadrature encoder for 2 coils or 3-state encoder for 2 coils. The LC sensors are periodically excited (sampling rate) and the LC circuit oscillation dampening factor is measured to determine

whether the conductive material is near to the sensor or not. For two coils sensors, the coil sensors are placed at 90 degrees to each other, and near the rotation disc, with half of its surface covered with a conductive material, as shown in Figure 1, make up the quadrature encoder. The quadrature encoder produces the signal shown in Figure 2.

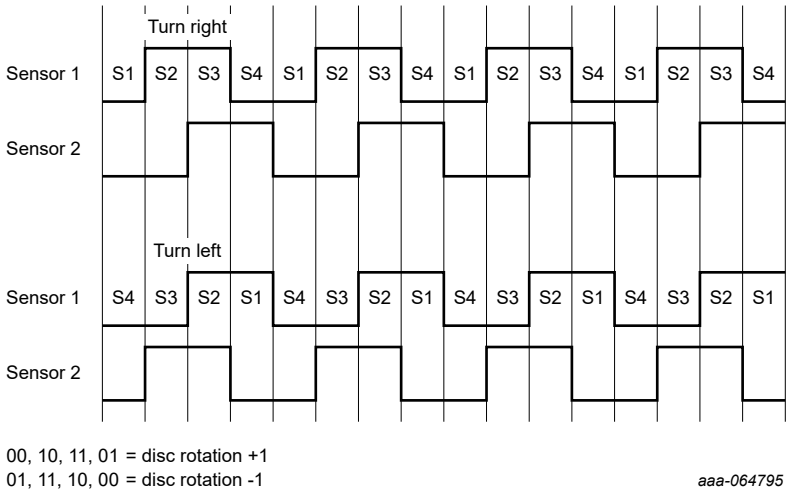


Figure 2. The quadrature encoder signals from the sensors when turned right/left in continuous mode

In Figure 3, there is a graph of signals produced by the encoder. The rotation and rotation direction may be determined. The signals in the picture are continuous, but the LC sensor allows only sampling of these signals by a given sampling rate. For three coils sensors, the coil sensors are placed at 120 degrees to each other, and near the rotation disc, with half of its surface covered with a conductive material, as shown in Figure 1, make up the 3-state encoder. The 3-state encoder produces the signal shown in Figure 3.

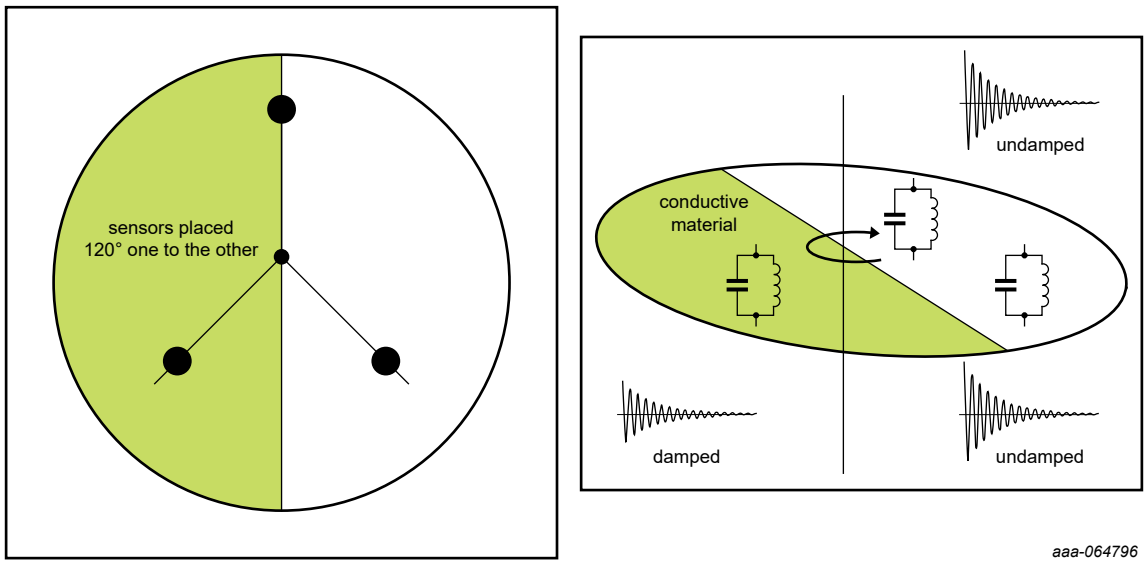


Figure 3. The three-LC sensor principle at work

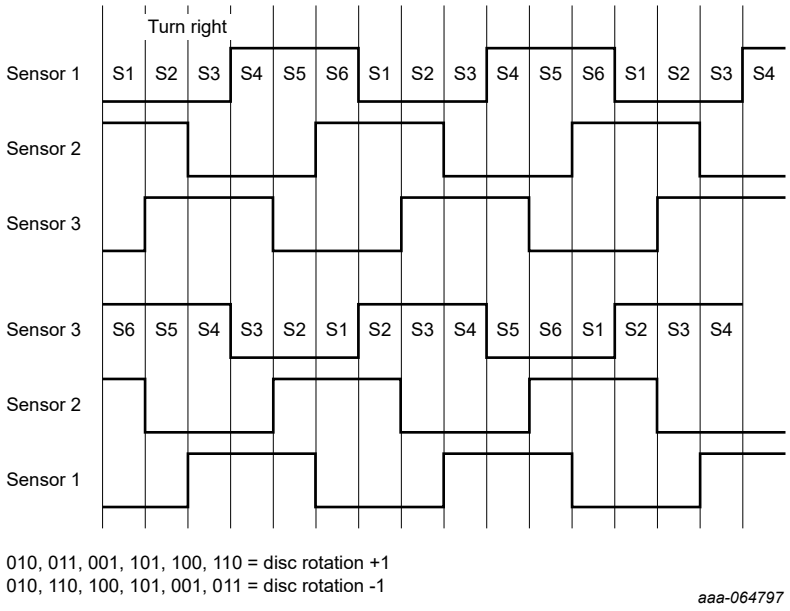


Figure 4. The 3-state encoder signals from the sensors when turned right/left in continuous mode

In Figure 4, there is a graph of signals produced by the encoder. The rotation and rotation direction may be determined. The signals in the picture are continuous, but the LC sensor allows only sampling of these signals by a given sampling rate. For accuracy and safe rotation detection, a correct sampling rate must be chosen. Use the following formula to calculate the minimal sampling rate, if two sensors spaced 90 degrees to each other are used.

$$\mathrm{fs} = 2(\mathrm{sensors}) \times 2(\mathrm{disc\ parts}) \times 2(\mathrm{nuy\,nquist}) \times \mathrm{rot}_{\mathrm{max}}$$

where rot $_{max}$ is rotation disc maximal rotation [turns/sec]

The samples of the signal produced by the coils are then processed to calculate the turns of the disc.

3 System block diagram

The evaluation demo setup of LC sensor application consists of a FRDM-MCXL255, FLO-SNSR-BRD, and a disc rotor motor assembly. The block scheme is shown in Figure 5.

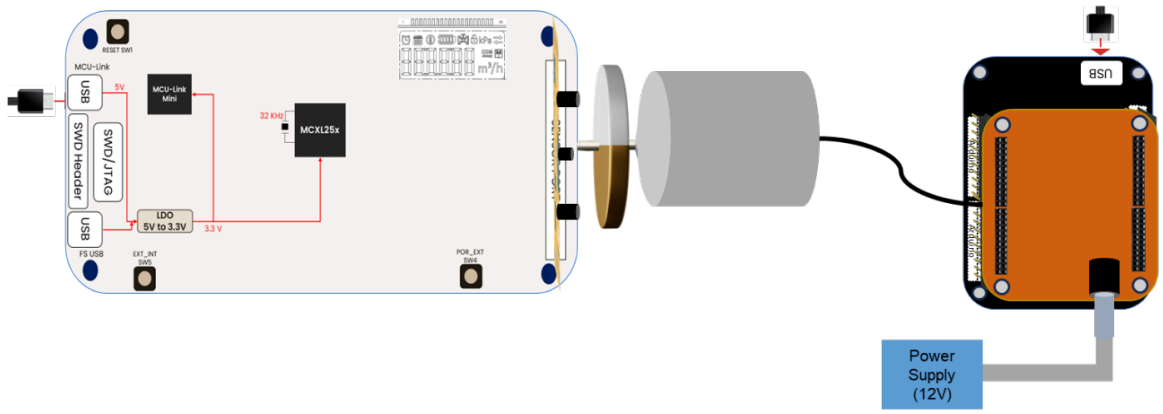


Figure 5. System block diagram

4 Application description

This section gives a brief overview of the hardware setup and operation of the LC-sensor–based rotation-sensing demo. It outlines the required boards, how to assemble and align them, and how the MCX L255 and FlexLC hardware detect disc rotation using two or three LC coils.

4.1 FRDM-MCXL255 board

The NXP FRDM-MCXL255 board is a low-cost design and evaluation board based on the MCX L255 device.

The FRDM-MCXL255 board consists of one MCX L255 device with a 4 Mbit external serial flash (provided by Winbond). The board also features an NMH1000 magnetic sensor, FXLS8974 Acceleration sensor,

RGB LED, segment LCD glass, push buttons, and MCU-Link debug probe circuit. The board is compatible with the Arduino shield modules, PMOD boards, and mikroBUS. The board also supports a dedicated GPIO interface to connect ultralow power sensor interfaces externally.

The details of the FRDM-MCXL255 board can be found in the Quick Start Guide (QSG) and User Manual (UM) documents of the board. Figure 6 shows an FRDM-MCXL255 board.

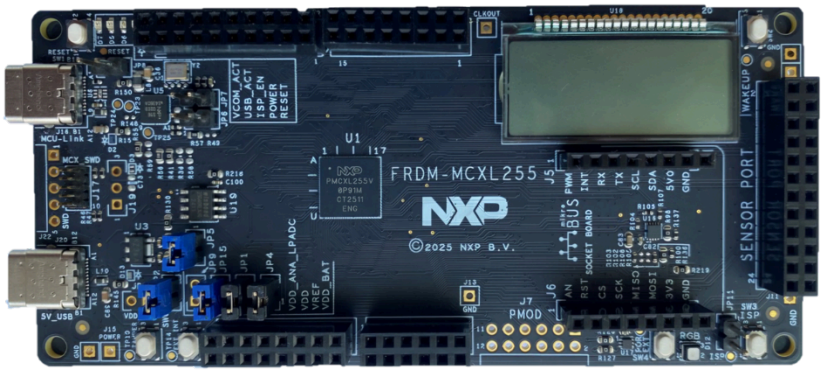


Figure 6. FRDM-MCXL255 board

4.2 FLO-SNSR-BRD for LC sense

FLO-SNSR-BRD board is a low-cost sensor board, used as an add-on board for multiple use cases:

- LC Sense,
- Ultrasonic,
- Magnetic Sensor.

Figure 7 shows the top view of FLO-SNSR-BRD. The right-angle male connector J1 at the edge is used to connect to FRDM-MCXL255 at J8. The two other Arduino-shield connectors J102, J103 are not used for LC sense operation.

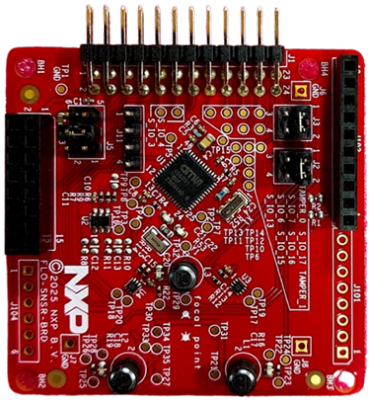


Figure 7. FLO-SNSR-BRD top view

Few jumpers are needed to be set up for an LC sense demo with this board. The settings on those jumpers are shown in Table 1.

Table 1. FLO-SNSR-BRD jumper settings

Connector	Positions
J5	1-2: Open, 3-4: Short, 5-6: Open
J3	1-2: Short, 3-4: Open
J2	1-2: Short, 3-4: Open

The connection of jumpers is shown in Figure 8.

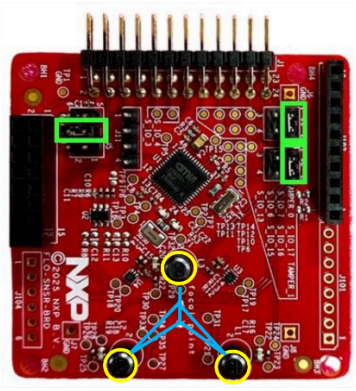


Figure 8. FLO-SNSR-BRD jumper settings for LC-sense operation

4.3 Preparing for the demo

Table 2 lists the boards required to prepare the demo.

Table 2. List of boards

Serial No.	Item	Description	Reference link
1	FRDM-MCXL255	Freedom board, USB-Type C cable	User Manual: TBD Reference Manual: TBD Datasheet: TBD
2	FLO-SNSR-BRD	Flow sensor board	This document
3	Motor Assembly	FRDM-MCXA153 + FRDM-MC-LVMTR	PMSM Sensorless FOC Using MCX A153 | NXP Semiconductors
4	Disc with half-metal		

1. Gather all required components as shown in Table 2.
2. Mount FLO-SNSR-BRD perpendicularly on the SENSOR PORT connector of FRDM-MCXL255. The J1 connector of FLO-SNSR-BRD must meet and align to the J5 connector of the FRDM-MCXL255 board as shown in Figure 9.
3. Connect a USB-Type C cable to connector J16 of FRDM-MCXL255. The power LED D4 should turn on.
4. Download and build LC sense demo program from Application Code Hub.
5. Build and program the `flowmeter_lcsense2_mcxl20` software in the FRDM-MCXL255 board using the on-board programmer accessible through the type-C cable. Power cycle the board and observe the welcome message NXP appearing on the segment LCD.

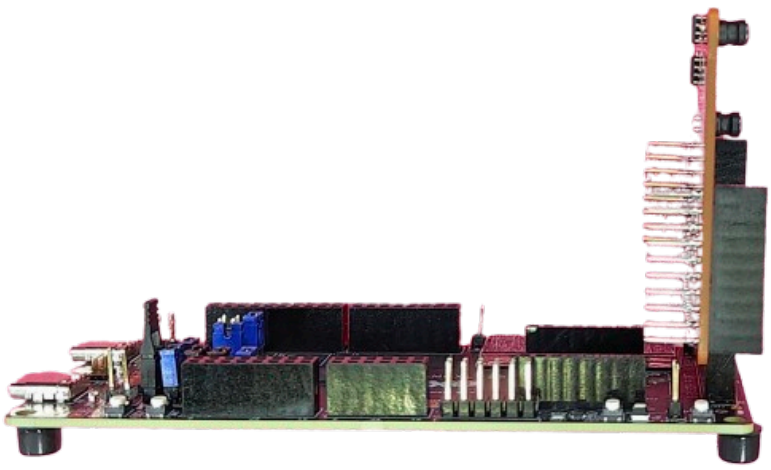


Figure 9. FRDM-MCXL255 and FLO-SNSR-BRD assembly

6. Prepare the disc rotor assembly:
   a. Place the FRDM-MC-LVMTR board onto the Arduino connectors (J1, J2, J3, J4) of the FRDM-MCXA153 connectors (J1, J2, J3, J4).
   b. Connect type-C USB cable to FRDM-MCXA153 (original - FRDM-MCXA154, confirm that my fix is correct) connector J15.

c. Connect a 24V DC adapter jack to connector J6 of this board. Connect the DC motor to connector J7 and J8 of the FRDM-MC-LVMTR board.
d. Mount the disc with a half-metal disc on the rotor shaft of the DC motor.

7. Download the SDK for FRDM-MCXA153 and build the `frdmmcxa153_mc_pmsm_enc` software. Program the FRDM-MCXA153 board with the on-board debugger accessible through a type-C USB cable used to power this board.
8. Download the FreeMASTER tool from https://www.nxp.com/design/design-center/software/development-software/freemaster-run-time-debugging-tool:FREEMASTER. Find file `pmsm_frac_enc.pmpx` at `frdmmcxa153_mc_pmsm_enc\motor_control\freemaster` and open with the FreeMASTER tool 3.2. The interface looks like Figure 10.

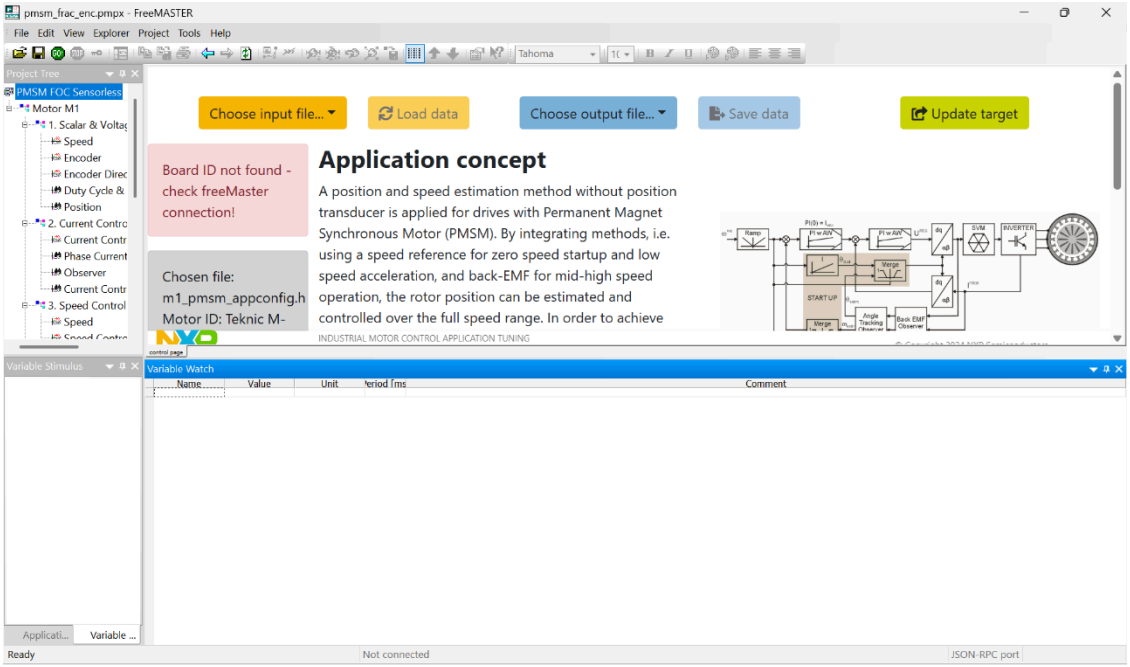


Figure 10. FreeMASTER interface

9. Mount the DC motor on a stable platform for operation during the demo. A fitting with FRDM-MCXA153 and FRDM-MC-LVMTR looks like Figure 12.
10. Place the FRDM-MCXL255+FLO-SNSR-BRD assembly close and aligned to the half-metal disc of the DC motor. Align the LC circuit focal point on the FLO-SNSR-BRD with the center of the half-metal disc. Refer to Figure 11 to identify the focal point on this board.

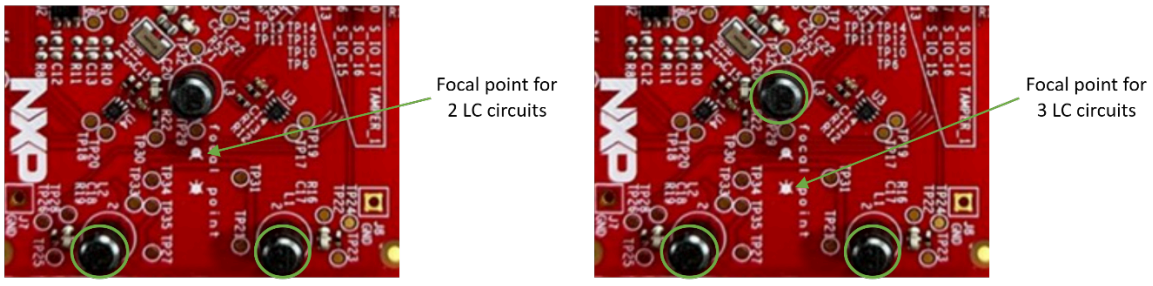


Figure 11. Two and three LC circuit focal points

After the all the arrangements, the setup looks like Figure 12.

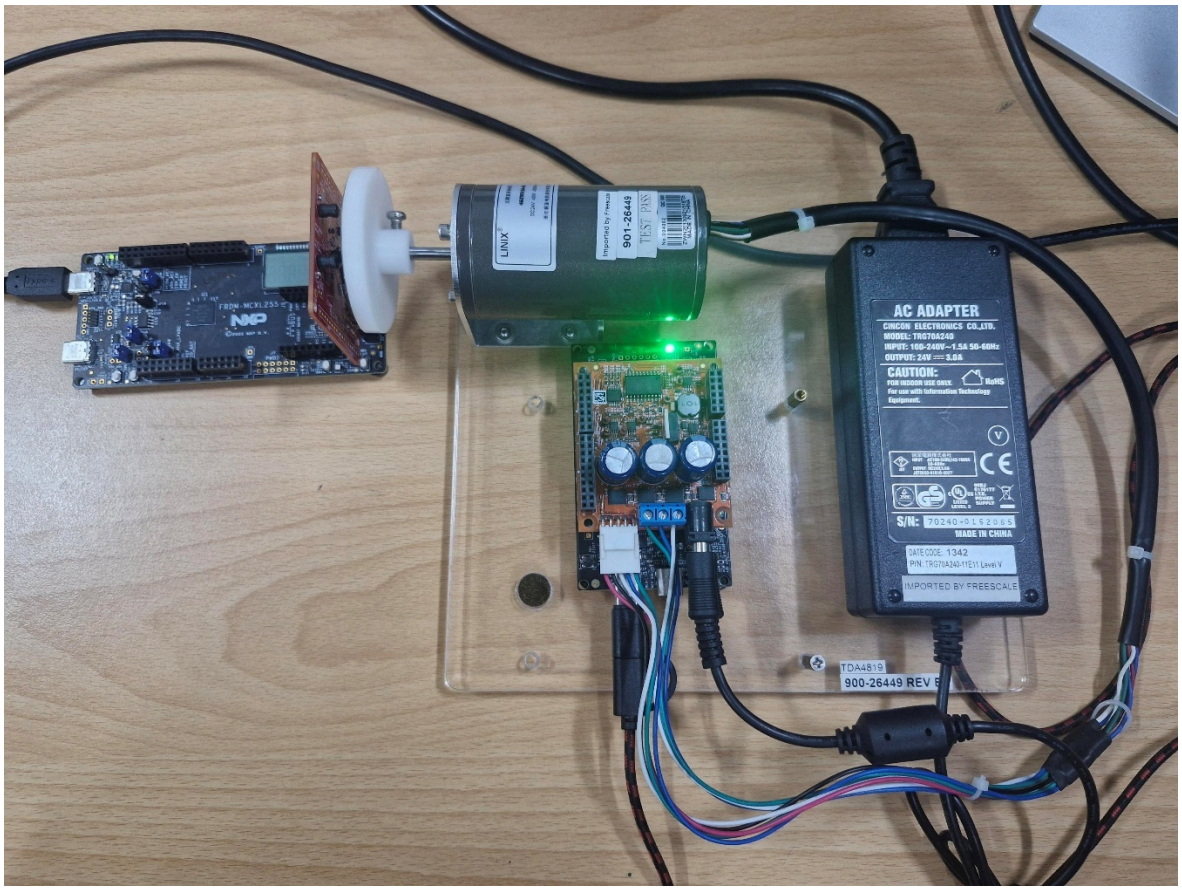


Figure 12. LC sense demo setup

4.4 Running Two-LC sense demo

The *flowmeter_lcsense2_mcxl20* application must be preprogrammed onto the board to run the two-LC demonstration.

After completing the setup described in Section 4.3, follow the steps below to run the 2-LC demonstration.

1. After opening `pmsm_frac_enc.pmpx` with FreeMASTER tool, the connection of the tool to FRDM-MCXL255 USB serial port can be opened as shown in Figure 13.

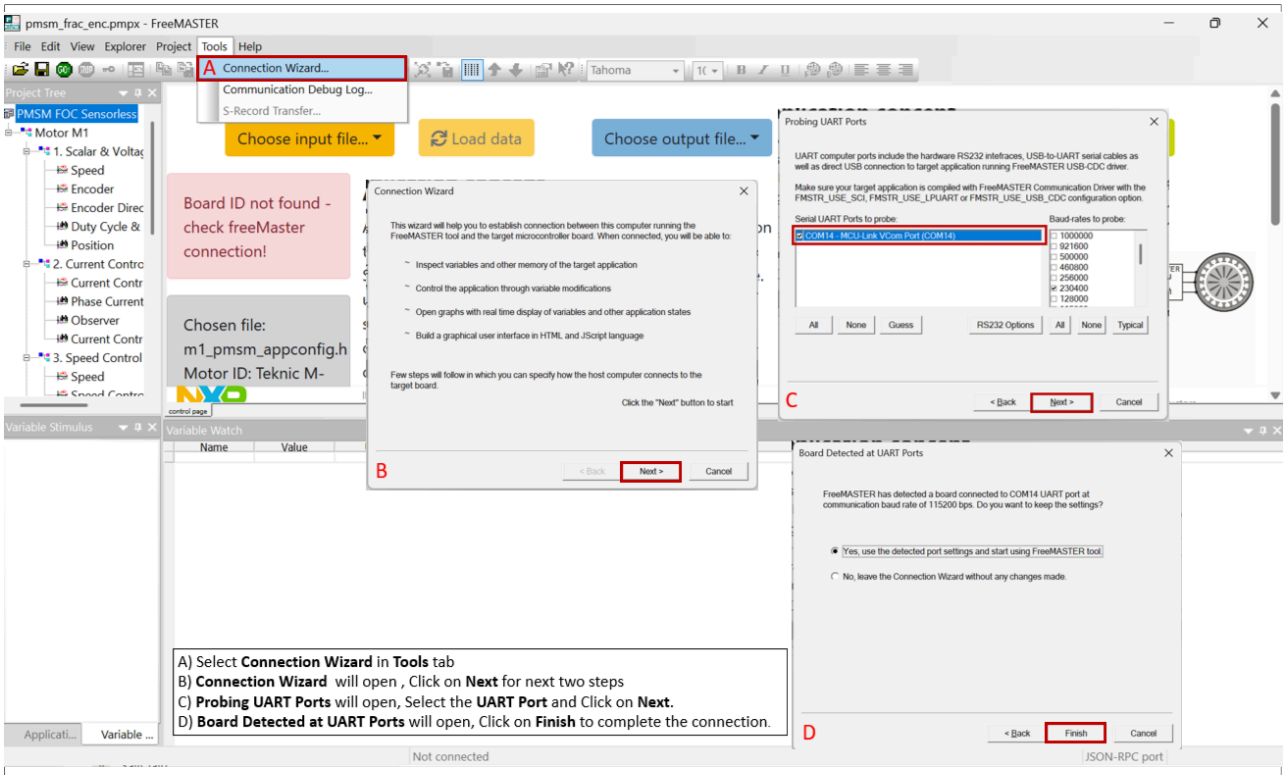


Figure 13. Setting up with FreeMASTER

After a successful connection, the work project appears in the left-hand pane of FreeMASTER. In **Motor M1 → Scaler and Voltage Control**, select the Speed parameter.

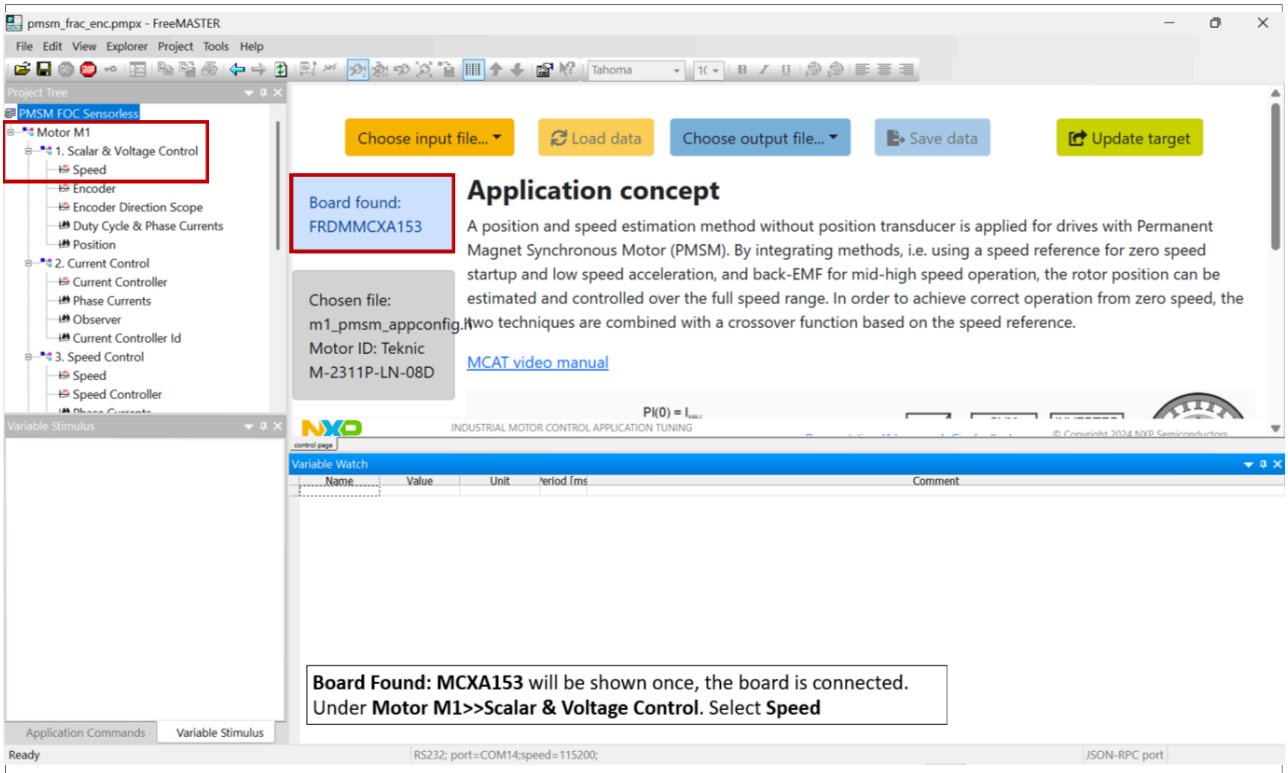


Figure 14. Status after successful connection

Set the Speed to -60 rpm. This makes the motor and installed half-metal disc to rotate at 60 rotations/ minute. Rotation in the -ve direction corresponds to the bottom-side placement of LC tank circuits on FLO-SNSR-BRD.

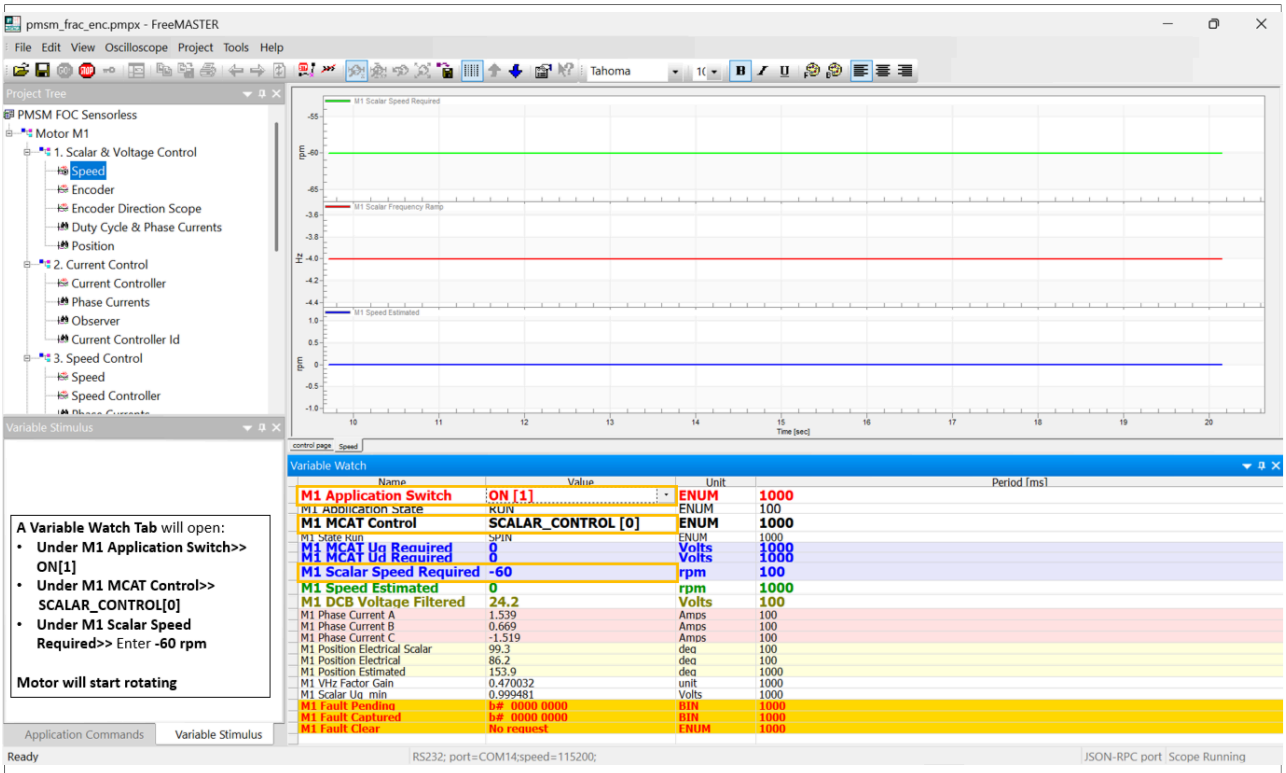


Figure 15. Controlling disc rotation speed

2. Power cycle FRDM-MCXL255. The NXP welcome message appears on the LCD. The LCD automatically cycles through the messages every two seconds in the following sequence:
   - NXP
   - Program build date
   - Program build time
   - Cumulative disc rotation count
   - Average disc rotation count (/hr)
3. The RTC date and time are updated. After every 1 second, the time message will update. After a date is changed, the next date will update on the LCD message. The cumulative flow count is updated after every second. Therefore, after power ON of the FRDM-MCXL255 and disc in rotation, the flow count on LCD will show values like 000001, 000002, 000003.

Figure 16. DISPLAY DISC ROTATION COUNT ON LCD

4. The messages on the LCD disappear after 15 seconds, as the MCU goes to deep-sleep mode. To wake up the MCU to active mode, press switch SW5. The sequence of messages on LCD shows up as mentioned before. The switch SW5 can also be used to manually scroll the messages. A press on SW5 resets the sleep-mode entry timeout and retains the MCU in active mode for another 15 seconds. The MCU goes to deep-sleep mode since the last time SW5 had been pressed.
5. While the DC motor with a half-metal disc keeps rotating, the flow volume is continued to be sensed by the LC sense circuits of FLO-SNSR-BRD prealigned to the disc. The entry-exit of the deep-sleep and active modes does not interrupt the flow volume accumulation as the sensor interface keeps working in the deep-sleep and active modes. Ensure that the FLO-SNSR-BRD remains correctly aligned with the motor disc throughout the demonstration.
6. Adjust the speed of the DC motor from FreeMASTER GUI from initial -60 rpm to higher rates: -120 rpm, -240 rpm. For -120 rpm rotation speed of DC motor, the flow count message on FRDM-MCXL255 will update faster than before. For -120 rpm speed, the flow count updates at the defined rate per seconds and so on. The maximum speed of the DC motor tested with the demo is -12000 rpm, that is, 200 rotations per second.

4.5 Running Three-LC sense demo

The flowmeter_lcsense3_mcxl20 application should be preprogrammed to run a three-LC demo.

After completing the setup described in Section 4.3, use the steps to run 3 x LC demo similar as in 2 x LC demo, except the alignment requirement of the center of a half-metal disc with the focal point as shown in Figure 11 of FLO-SNSR-BRD for 3 x LC sensors.

Figure 17. DISPLAY DISC ROTATION COUNT ON LCD

4.6 The LC Sensor schematic and interface to MCX L255 periphery

Flow meters are battery-operated devices; therefore, low power consumption is critical. To achieve low power consumption while sensing the LC sensors, only the required peripherals remain active, while the CPUs and other peripherals are either in the switch-off state or in ultralow power mode. Figure 18 shows a high-level block diagram of MCX L255. The real-time (main) domain with CM33 CPU and main domain peripherals are shut down in deep-power-down mode while the required peripherals on ultralow power (AON) domain are kept operational. The CM0+ processor and other unused peripherals on the AON domain are kept in OFF state to reduce power consumption.

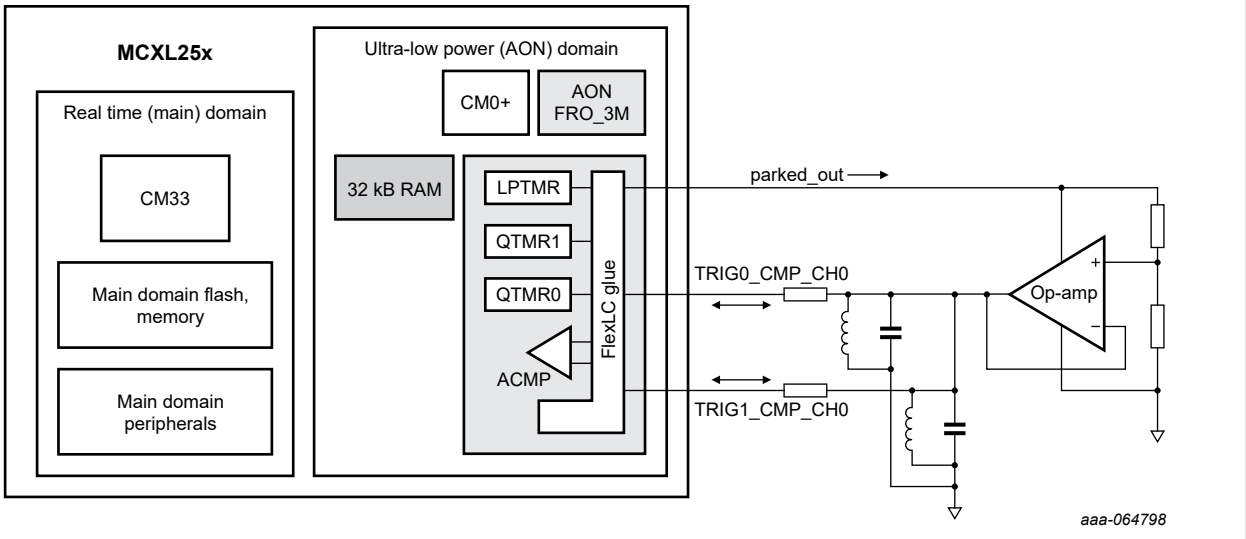


Figure 18. The LC sensor interfacing block diagram

The AON clock is set at 3 MHz with an internal RC oscillator (FRO). The real-time-clock (RTC) runs at 16.384 kHz rate with the help of an external 32.768 kHz crystal. The 3 MHz AON FRO clock is used as the functional clock for the timers and comparator used for LC sensors, after division by module-specific clock dividers. The RTC clock is not used for LC sensor but is used to run the RTC module calendar and raises a periodic alarm (ALARM0) used to calculate the flow rate of the sensor.

Figure 19 shows the schematic of the LC tank circuit with the single-channel operational amplifier present on FLO-SNSR-BRD.

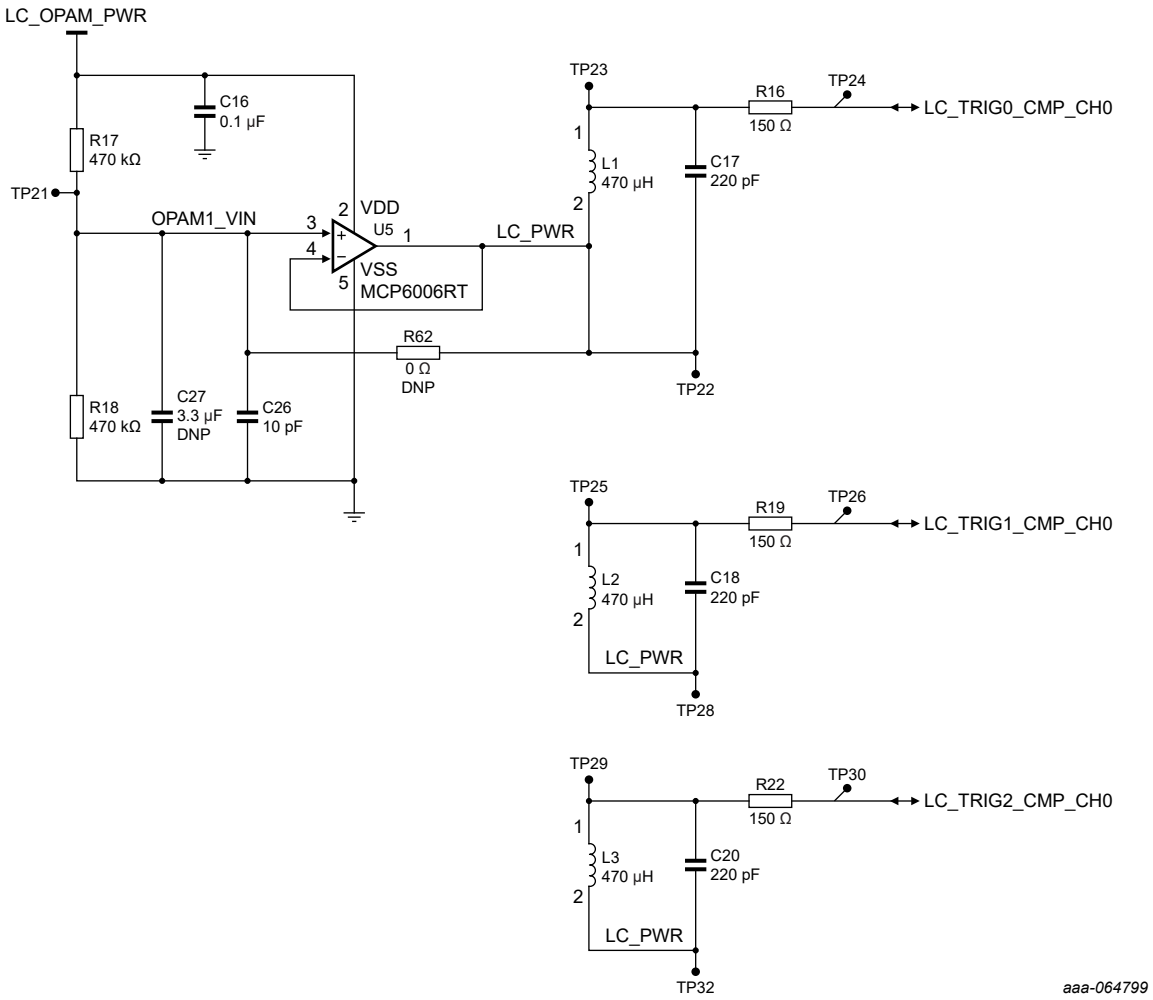


Figure 19. Schematic diagram of LC sensors

The L1, C17 resonator circuit is driven and charged through the U5 operational amplifier. The single instance operational amplifier is powered by the MCX L255 port pin signal LC_OPAM_PWR from AON port. The resistive divider R17, R18 biases the op-amp with *VDD/2 level configured in unity-gain buffer configuration. The output of the buffer is held at VDD/2 level. The L1, C17 resonator/tank circuit can be charged when the voltage level at `LC_TRIGn_CH0` is held low for a short duration. Once capacitor C17 is charged to VDD/2, the LC_TRIGn_CH0 switches back to comparator (input) mode and the L-C circuit starts to oscillate.

Note: *VDD is typically 3.3 V, sourced from an AON port pin of the MCX L255 MCU.*

4.7 FlexLC block diagram

Figure 20 shows the block diagram of the FlexLC module in the MCX L255 MCU. It is a glue-logic implementation that coordinates various timers and comparators to emulate LC sense operation.

The Flex LC (or LC Sense) logic block is designed to create a low-power encoder based on inductive-capacitive (LC) sensors. The objective is to develop a low-power design suitable for paddle wheel flow meters that assess the rotation of the paddle wheel. Employing LC sensors to measure rotational movement is a standard

approach in hybrid flow meter designs. The paddle-wheel rotation is monitored by multiple LC sensors that detect the presence of conductive material on the disc attached to the wheel.

The Flex LC logic block is composed of multiple IPs, which are integrated using custom chip glue logic to optimize it for LC Sensing applications. Although this glue logic is designed for LC Sensing, it can also be applied to other scenarios if its functionality aligns with the necessary specifications.

These are the components of the Flex LC logic:

- Peripheral input multiplexing mappings
- Channel Sequencer
- Comparator Pad Logic
- Control Pad Logic
- Channel State Capture
- LPTMR and QTMR counters
- Control and Status registers in SYSCON_AON

Each of the components listed above is described in detail in the MCXL25x Reference Manual. End users can evaluate the best ways to apply the available features. NXP has supplied the guidance through this application notes, software, and demonstrations. End customer can use this application note as a reference to implement and customize as per their needs.

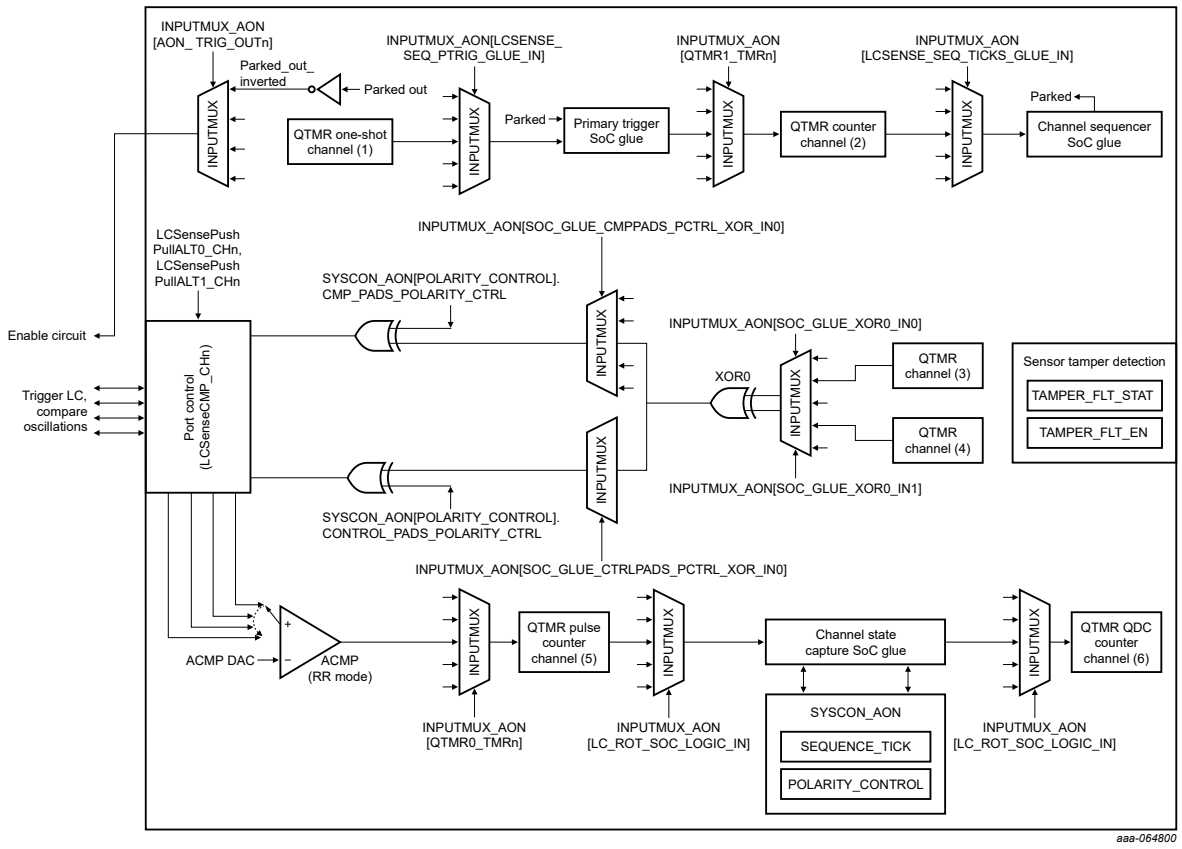


Figure 20. Block diagram of FlexLC Logic

4.8 Two-LC sense operation

The two-LC sensing circuitry operates autonomously for the lifetime of the flow meter by using the mechanism described here. The FlexLC and associated modules must be configured only once, after which they operate continuously without further intervention. The rotation counter stores disc rotation count automatically until an overflow occurs in the 32-bit counter, and the count rate can be measured periodically using an RTC ALARM0 wake-up of CM0+ and/or CM33 CPU, to do metering application, and then again going back to sleep.

4.8.1 Working principle

- The primary trigger used to activate the FlexLC block is generated by the QTMR1.CH3 channel. To produce this trigger periodically, the LPTMR module is used as a pretrigger source. LPTMR initiates the QTMR1.CH3 one-shot pulse of approximately 300 usec. This one-shot duration is sufficient to sample the two LC circuits at regular intervals. For a sampling rate of 200 samples per second across two LC circuits, the LPTMR frequency is configured to 200 Hz, corresponding to a 5 ms period. As a result, QTMR1.CH3 generates a 300 usec one-shot pulse every 5 ms.
- The QTMR1.CH0 channel serves as the input to the channel sequencer, which generates the periodic sequence for LC1 and LC2. QTMR1.CH1, QTMR1.CH2, and their XOR-combined output generate the trigger pulses for the LC1 and LC2 circuits. The pulse periods and widths of QTMR1.CH1 and QTMR1.CH2 are precisely configured so that the XOR output produces a narrow trigger pulse of approximately 3 usec.
- During each 300 usec QTMR1.CH3 one-shot window, two independent 3 usec trigger pulses are generated sequentially for LC1 and LC2, separated by approximately 140 usec.
- When the trigger pulse is output on an MCU port pin, the external LC tank circuit is charged to VDD/2 using the output of an operational amplifier. This amplifier is driven by an internally generated envelope signal ("parked_out") from the MCU. The “parked_out” signal remains asserted for approximately the same duration as the QTMR1.CH3 one-shot pulse, that is, around 300 usec.
- Immediately after the LC1 trigger pulse is driven low for 3 usec, the port pin is switched back to comparator-input mode. This removes the drive from the LC tank circuit, allowing the stored charge to discharge and form a damped oscillation. These oscillations are monitored by the ACMP_AON module through the same port pin. The oscillation frequency depends on the selected L and C values of the tank circuit, and is defined as:

$$f_{osc} = \frac{1}{2\pi\sqrt{L \times C}}$$

  Where:

  $L = 470\mu H = 470 \times 10^{-6} H$

  $C = 220pF = 220 \times 10^{-12} F$

  The resulting f_{osc} is 496 kHz. The oscillation duration is approximately 2 usec.
- The primary sensing element is the L coil, which forms a resonant LC pair with the capacitor. The parameters of both components directly influence sensitivity. The coil’s series resistance and the oscillation frequency determine the damping factor and decay time. The coil geometry influences the current magnitude and overall power consumption.
- For the L and C values used on the FLO-SNSR-BRD, the damping oscillation generated by the 3 usec trigger decays within approximately 70 usec. Therefore, the LC2 trigger is delayed by about 140 usec relative to the LC1 trigger.
- The LC2 circuit is triggered by a second MCU port pin. Once the trigger pulse is withdrawn, the port pin switches back to comparator-input mode, allowing observation of the new oscillation sequence.
- When the LC circuit is close to the metallic portion of the rotating disc, the magnetic characteristics of the tank circuit change, causing a faster decay and shorter oscillation time. When the LC circuit is aligned with the nonmetallic part of the disc, the oscillation count reaches its maximum, depending on coil Q-factor and other component characteristics.

- When the port pins return to comparator-input mode, both are routed to two separate inputs of the ACMP_AON comparator block. The comparator's positive input is alternately connected to LC1 and LC2 in a round-robin manner, while the negative input is tied to the DAC output.
- The ACMP_AON compares the oscillation waveform against its internal DAC output. The LC oscillation is centered around VDD/2 (1.65 V when VDDA = 3.3 V). To ensure accurate detection, the DAC output is set above VDD/2 to suppress noise near this offset. With the DAC connected to the negative input, incoming oscillations at the positive input generate a sequence of pulses. A higher DAC level results in fewer pulse outputs, while a lower DAC level produces more pulses.
- A counter on QTMR0.CH0 counts the comparator-generated pulses. The counter must be reset between LC1 and LC2 measurements; otherwise, pulses from both channels would accumulate. After the sequencer completes the LC1 measurement window, the counter is automatically reset so that LC2 pulses are counted starting from zero.
- QTMR0.CH0 compares the pulse count against a predefined threshold. When the count exceeds the compare value stored in QTMR0.CH0.CMPLD1, the channel output toggles high (default is low), although counting continues. The compare value is selected so that pulse counts for the nonmetal (air) region exceed the threshold, while counts for the metal region fall below it. For example, if the comparator produces 28 pulses in the air region and 16 pulses near metal, the midpoint (28 + 16) / 2 = 22 is used as the compare value.
- QTMR0.CH0 is reset on the rising edge of QTMR1.CH0, ensuring that LC1 pulses are cleared before LC2 pulses begin accumulating.
- QTMR0.CH1 is configured in Quadrature Counter mode to count the internal signals LC_ROT_SOC_LOGIC_OUT1 and LC_ROT_SOC_LOGIC_OUT2, as described in the timing diagram subsection.

4.8.2 Sensor measurement timing diagram

The timing of trigger generation for LC1, LC2, and the switching back to comparator inputs is controlled by the internal channel-sequencer logic of FlexLC and by the pad-control logic implemented through the ALTn function for LC-sense support. The sequencer logic is activated by the QTMR1.CH3 one-shot timer output and remains active as long as the one-shot timer output is high. The timing diagram and relation between QTMR1.CH3 output, the outputs of the channel sequencer, the `parked_out` signal, QTMR1.CH0, QTMR1.CH1, QTMR1.CH2, the XOR trigger output, the LC tank-circuit drive, and the comparator input are shown in Figure 21.

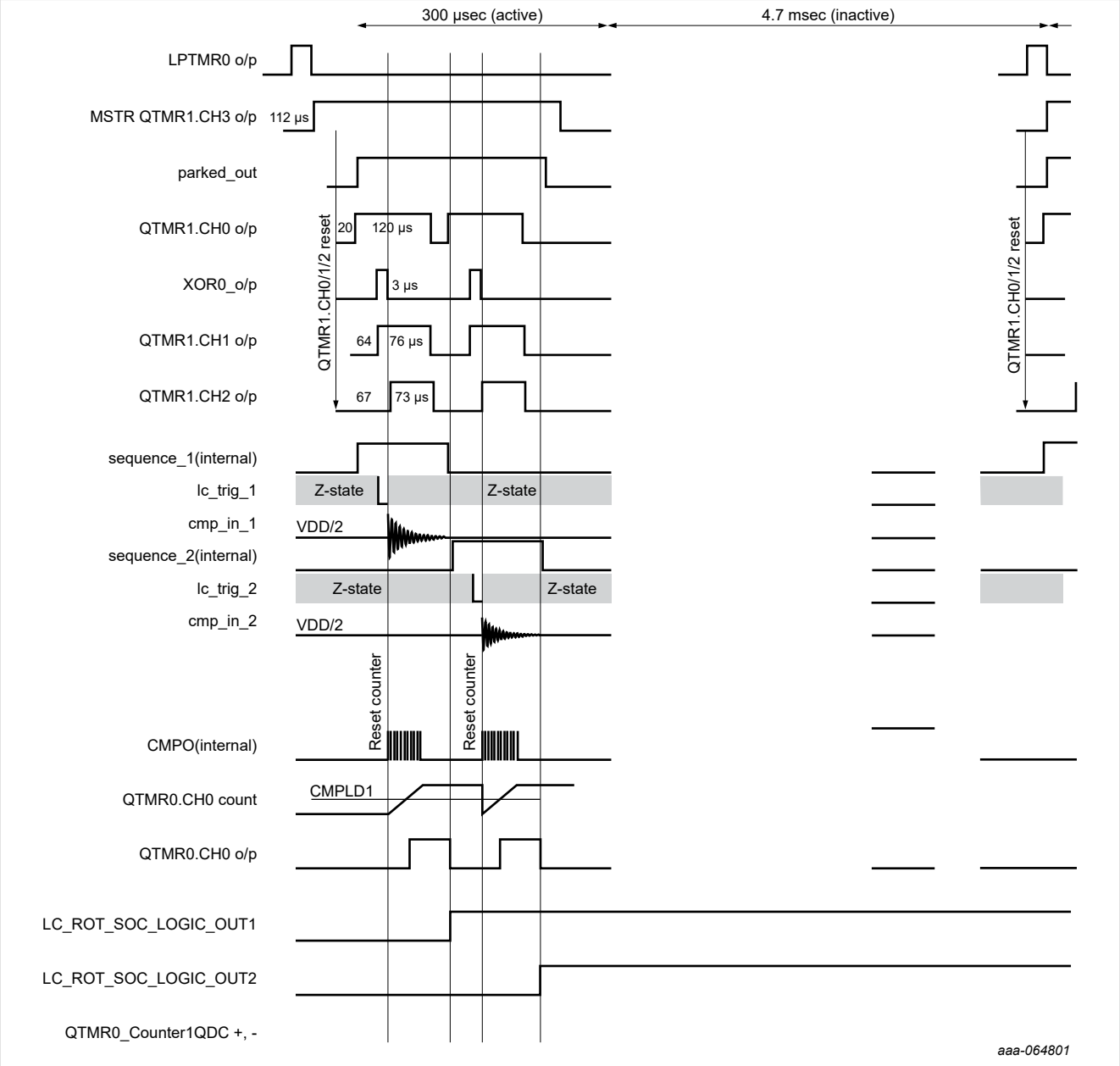


Figure 21. Two-LC sensor timing diagram

Understanding the timing diagram:

- LPTMR starts the sampling and periodicity. In this demo, the compare frequency of LPTMR is set to 200 Hz so that it generates a trigger pulse every 5 msec before resetting and resuming itself. With the trigger rate of 200 Hz, which is also the sampling rate, the sampling will happen at the rate of 200 x no. of LC circuits per second. It can support sampling of 200/(no. of disc x Nyquist) = 200/(2 x 2) = 50 rotations per second, at maximum. If the maximum rotation of the end-user application is fewer than 50 rotations per second, the frequency of LPTMR can be reduced so that the sense activity time is reduced, thereby reducing power consumption.
- LPTMR starts QTMR1.CH3 in one-shot mode, which generates a high pulse of about 300 usec.
- Along with QTMR1.CH3, QTMR1.CH0, QTMR1.CH1, QTMR1.CH2 are also started together in synchronous start method. Therefore, all four channels of QTMR1 are started together and work synchronously.

- The periods of QTMR1.CH0, QTMR1.CH1 and QTMR1.CH2 are set to 140 usec. The low-level period of QTMR1.CH1, QTMR1.CH2 are set to be 64 usec and 67 usec so that their duty cycle difference helps to generate the required trigger for LC sense when passed through the XOR gate. The XOR gate is part of FlexLC glue logic. The high period of XOR gate = (67 – 64) = 3 usec. Refer to Figure 28
- The QTMR1.CH0 is used as a source to the channel sequencer. Two sequence signals “sequence_1”, “sequence_2” are internally used to separate XOR output triggers to separate channels. The push-pull pad control of port pins used to drive external LC circuits with these separated trigger signals is also enabled by the “sequence_1”, “sequence_2” signals internally.
- After the trigger period is over, the pad control logic (by using LCSensePushPullALT0_CHn, LCSensePushPullALT1_CHn functions of Port pins) switches the port pins to comparator input direction by FlexLC glue logic hardware. Only after this can the oscillations on LC circuits start.
- The timing and period of QTMR1.CHn are carefully selected to allow the damping of oscillations to decay for about 70 usec. The trigger for LC2 channel is done after QTMR1.CH0/1/2 period of 140 usec, therefore allowing more time than 70 usec oscillations for LC1 circuit.
- The comparator output converts the oscillations of both LC1and LC2 into pulses. These pulses follow the LC1, LC2 oscillations above the comparator negative input threshold. The frequency of pulses is about 496 kHz varying duty cycle.
- QTMR0.CH0 counts the pulses from comparator for both LC1and LC2 oscillations. The counter output toggles if the count is more than a preset compare-threshold value. Counter output does not toggle if the pulses are less than the preset compare-threshold value, which happens when there is a metal part of the half-metal disc in the vicinity.
- The serialized QTMR0.CH0 toggling output is sampled and separated into LC_ROT_SOC_LOGIC_OUT1, LC_ROT_SOC_LOGIC_OUT2 signals. These two signals are phase-shifted by 90 degrees to each other, only within the overall sampling duration of 300 usec, and not the overall LPTMR per-trigger sampling duration of 5 msec. This usually is not an issue for the QTMR0.CH0 in quadrature decoder (QDC) mode counter, as the nonactivity of the ACMP_AON round-robin comparator for the rest of (5 msec – 300 usec =) 4.7 msec helps to reduce current consumption.
- The quadrature signals LC_ROT_SOC_LOGIC_OUT1, LC_ROT_SOC_LOGIC_OUT2 are connected to QTMR0.CH0 phase A, B inputs and counts up/down based on the phase states.

Figure 22 shows the oscilloscope trace of two LC circuits. “parked” out signal is used to power the op-amp in FLO-SNSR-BRD. cmp_in_1, cmp_in_2 are the trigger (XOR o/p) cum comparator inputs generating oscillations. These three signals are only active for two LC circuits external to the MCX L255 MCU.

Figure 22. Two-LC sensor oscilloscope traces

Figure 23 shows the oscilloscope trace of two LC circuits, with an additional debug signal to see the comparator output generated internally. The CMPO (internal) shows the oscillations from cmp_in_1, cmp_in_2 are converted into pulses with varying amount, depending on the oscillation inputs. This signal can be brought out of MCX L255 for diagnostics purposes during development (and few more such internal signals) but must be disabled after the tune-up or debug is done, to save port power.

Figure 23. Two-LC sensor oscilloscope traces with comparator output

From the oscilloscope traces, the proximity and traces of LC circuits to each other can influence each other due to mutual induction, which can result in one signal to generate mild oscillations into another. Care must be taken during the placement and routing of these signals during PCB placement and layout to avoid this intersymbol crosstalk. Due to the round-robin feature of the AON.ACMP comparator, the influence in one LC channel signal does not directly impact the other channel during the comparison channel sequencer window. But due to too much proximity, the signals crosstalk can be higher to disturb the oscillation signal itself, resulting in difficulty for AON.ACMP to isolate the signals from each other.

4.9 Three-LC sense operation

The three-LC sense circuit drive part works autonomously for the lifetime of the flow meter employing this mechanism. However, the sense part can be employed either autonomously (as described in Section 4.8) or with the help of CM0+ software measurement with the CPU waking up periodically. The FlexLC and modules must be configured only once for forever operation. The rotation count and rate can be measured periodically using an RTC ALARM0 wake-up of the CM0+ and/or CM33 CPU to do the metering application, and then again going back to sleep.

In the case of three-LC sensors, the circuits are driven by the timers and glue logic of FlexLC. The disc rotation sense can be achieved with only two LC circuits. In this case, the QTMR0.CH0 can run autonomously in QDC counter mode, as described in Section 4.8. Typically, this is the case when the third LC circuit is used to detect a magnetic or cover-open tamper while operating alongside the first and second circuits.

Alternatively, in the case where all three LC circuits are used to detect the disc rotation, the working principle and timing are described in Section 4.9.1 and Section 4.9.2.

4.9.1 Working principle

- The QTMR1.CH3 channel generates the primary trigger for activating the FlexLC block. To generate this primary trigger periodically, LPTMR is used. LPTMR periodically starts the pretriggering process of three LC tanks circuits. LPTMR is used as a pretrigger to start QTMR1.CH3 to run a one shot for about 450 usec. The QTMR1.CH3 one-shot time is effective to sample three LC circuits periodically. For a sampling rate of 200 samples/sec, and three LC circuits, the frequency of LPTMR is set to 200 Hz, with a period of 5 msec. Therefore, the one shot QTMR1.CH3 is generated for 450 usec at every 5 msec periodically.
- Channel QTMR1.CH0 is used as an input to the channel sequencer to generate a periodic sequence for LC1, LC2, and LC3. QTMR1.CH1, QTMR1.CH2 and their outputs, and an XOR gate are used to generate the trigger pulses for the LC1, LC2, LC3 channels. The pulse period and pulse widths of QTMR1.CH1, QTMR1.CH2 are carefully set to produce the trigger small enough: about 3 usec only at the output of the XOR gate.
- During each 450usec, QTMR1.CH3 one-shot period, the 3 usec trigger pulses for LC1, LC2, and LC3 are generated sequentially, separated by approximately 140 usec.
- With the trigger output on a port pin of MCXL25x, the external LC tank circuit is charged to the VDD/2 level generated by an operational amplifier. The operational amplifier is powered with an internally generated envelope signal from the MCU called “parked_out”. The `parked_out` signal stays high as long as the one-shot QTMR1.CH3 output is high.
- After the 3 usec trigger for LC1 is applied, the port pin is switched back to comparator-input signal mode, thereby withdrawing the drive from the LC tank circuit to allow the developed charge to discharge gradually as a damping oscillation. The oscillations are allowed to be compared by the `ACMP_AON` module from this port pin. The oscillation frequency depends on the selected L and C component values in the LC tank circuit. The oscillation frequency is defined as

$$f_{osc} = \frac{1}{2\pi\sqrt{L \times C}}$$

Where:

$$L = 470\mu H = 470 \times 10^{-6} H$$
$$C = 220pF = 220 \times 10^{-12} F$$

The value of f_{osc} becomes 496 kHz. The duration of oscillations is approximately 2 usec.

- With the L and C components of FLO-SNSR-BRD, the damping oscillations generated with a 3 usec trigger damp within approximately 70 usec. Therefore, the trigger for LC2 is spaced about 140 usec from the trigger for LC1. Similarly, the trigger for LC3 is spaced about 140 usec from the trigger for LC2.
- The trigger to the LC2 circuit is applied by a second port pin of the MCU. Withdrawing the trigger drive generates another sequence of oscillations while the same port pin is switched back to comparator input mode.
- Similarly, the trigger to the LC3 circuit is applied by a third port pin of the MCU. Withdrawing the trigger drive generates another sequence of oscillations while the same port pin is switched back to comparator input mode.
- The main component of the sensor is the L coil, which is a sensing element. The coil L resonates with the C capacitor. The coil and capacitor parameters are important for proper sensitivity. The equivalent series resistance of the coil and the oscillation frequency influence the damping factor and damping time. Also, the physical dimensions of the coil influence the current flowing into the coil and affect the overall current consumption.
- When the LC circuit is closer to the metal half of the rotating disc, it affects the magnetic properties of the LC tank and reduces the oscillation time as it decays more quickly. If an LC circuit is close to the nonmetal part of the disc, the maximum number of oscillations can occur, depending on the Q-factor of the L component, and so on.
- Both triggering port pins, while switched back to comparator input, are connected to three different input signals of the single comparator block ACMP_AON of MCX L255. The positive input of the comparator is connected to LC1, LC2, and LC3 oscillation inputs in a round-robin mode, while its negative input is fixed to the output of its DAC.
- The comparator AON.ACMP compares the oscillations with its internal DAC output. LC oscillations occur with a DC offset of VDD/2 (= 1.65 V for VDDA = 3.3 V of the MCU). Therefore, to compare effectively, the internal DAC output of ACMP_AON is set at a higher level than VDD/2 so that the oscillations or noise near VDD/2 are excluded from comparison. The internal DAC output of ACMP_AON is connected to the negative input, so the incoming oscillations at the positive input are converted to pulses. Based on the DAC output level, the number of pulses produced at the comparator output varies. If the DAC level is set too high, it produces fewer pulse outputs. If the DAC level is low, it produces more pulse outputs. The DAC level should not be too small and close to the VDD/2 level.
- A counter channel in QTMR0.CH0 counts the pulses generated by the comparator. Unless the counter is reset after a channel oscillation, it will count pulses for all three LC channels. Therefore, after counting pulses for the dedicated time interval for LC1 timing for the sequencer channel (which is generated by the channel sequencer), the counter is reset. Then it counts the LC2 channel oscillations afresh, and it is reset again to count oscillations for LC3.
- The counter QTMR0.CH0 is set up to compare the pulse count with a predefined compare value. Whenever the count crosses the compare value stored in QTMR0.CH0.CMPLD1, its output toggles to high (the default state is ‘low’), although it continues to count the remaining available pulses. The compare value is carefully selected so that the number of pulses with a nonmetal (air) disc position is higher than this value, but lower when the metal part of the disc is close by. For example, with a given comparator DAC setting, if the comparator produces 28 pulses with the air part of the disc and 16 pulses with the metal part, then (28 + 16) / 2 = 22 is a suitable midpoint to be set as the compare value of the QTMR0.CH0 channel.
- The counter QTMR0.CH0 is reset at the rising edge of QTMR1.CH0. This helps reset the count for LC1 oscillation pulses before counting LC2 oscillation pulses again from a zero value.
- QTMR0.CH1 is used in one-shot mode and is triggered by the LPTMR output pretrigger to help the CM0+ CPU wake up periodically from deep-power-down (2) mode. The duration of this timer is set to surpass the three LC oscillations and counting, allowing their states to be reflected in the SYSCON_AON register

LC_ROT_LOGIC[CH_COMP_OUT] bits so that the CM0+ CPU can read the states before going back to sleep and wait for the next one-shot wake-up. This process wakes the CM0+ CPU more often and therefore causes higher power consumption than the two-LC operations described in Section 4.8.

4.9.2 Sensor measurement timing diagram

The timing of trigger generation for LC1, LC2, LC3, and the switching back to comparator inputs is controlled by the internal channel-sequencer logic of FlexLC and the pad-control logic implemented through the ALTn function for LC-sense support. The sequencer logic is activated by the QTMR1.CH3 one-shot timer output and remains active as long as the one-shot timer output is high. The timing diagram and relation between QTMR1.CH3 output, the outputs of the channel sequencer, the `parked_out` signal, QTMR1.CH0, QTMR1.CH1, QTMR1.CH2, the XOR trigger output, the LC tank circuit drive, and the comparator input are shown in Figure 24.

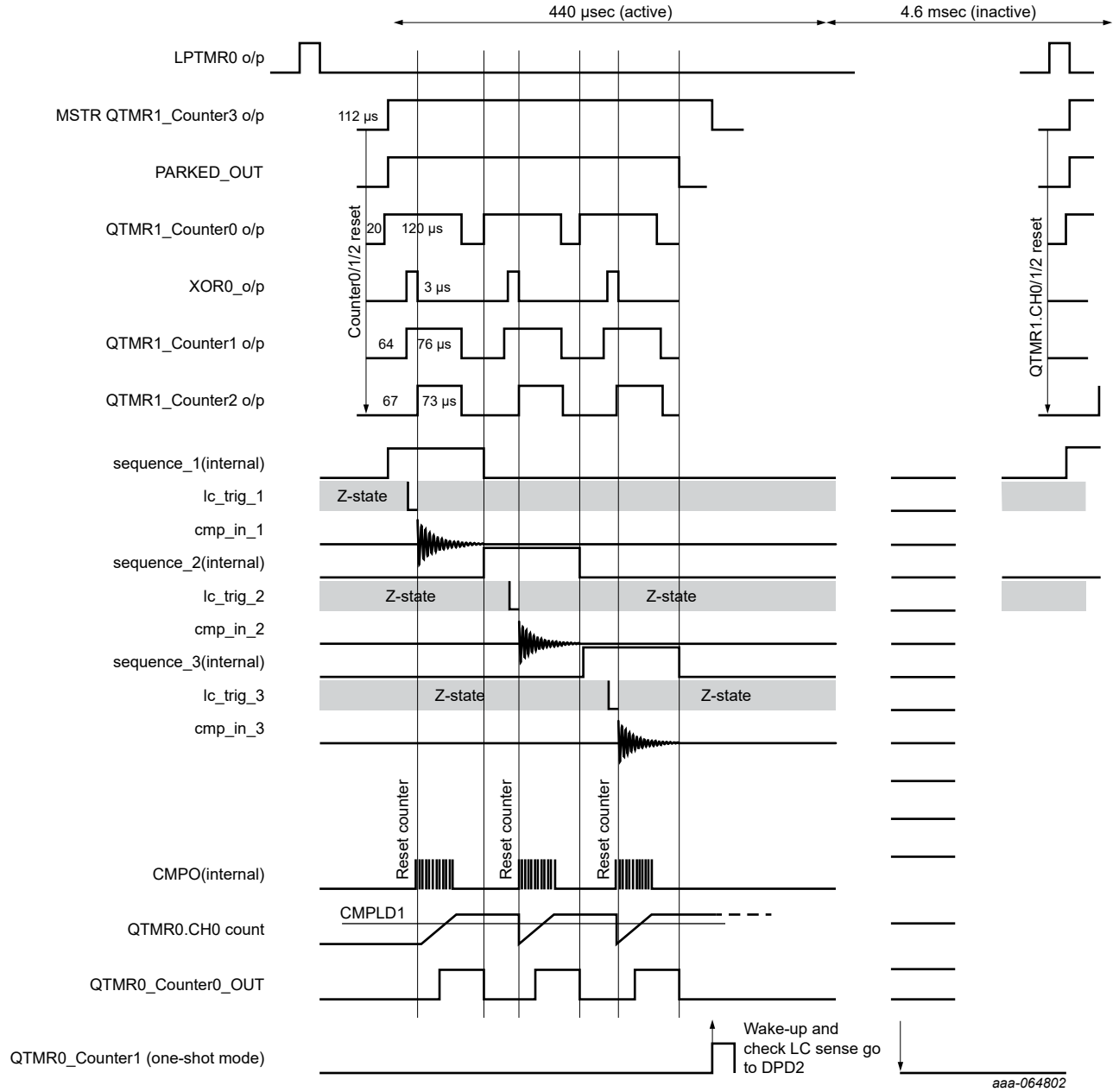


Figure 24. Three-LC sensor timing diagram

Understanding the timing diagram:

- LPTMR starts the sampling and periodicity. If its frequency is 200 Hz, it samples at the rate of 200 x no. of LC circuits. It supports sampling of 200/(no. of disc x Nyquist) = 200/(2 x 2) = 50 rotations per seconds, the maximum. When the rotation is less than 50 rotations, the frequency of LPTRM can be reduced so that the sense activity time is reduced, It reduces the power consumption.
- LPTMR starts QTMR1.CH3 in one-shot mode, which generates a high pulse of about 440 usec.
- Along with QTMR1.CH3, QTMR1.CH0, QTMR1.CH1, QTMR1.CH2 are also started together in synchronous start method. Therefore, all four channels of QTMR1 are started together and work synchronously.
- The period of QTMR1.CH0, QTMR1.CH1 and QTMR1.CH2 is set to 140 usec. The low-level period of QTMR1.CH1, QTMR1.CH2 is set to 64 usec and 67 usec so that their duty cycle difference helps to generate

the required trigger for LC sense when passed through the XOR gate. The XOR gate is part of the FlexLC glue logic. The high period of XOR gate = (67 – 64) = 3 usec. Refer to Figure 28

- QTMR1.CH0 is used as a source for the channel sequencer. Three sequence signals “sequence_1”, “sequence_2” and “sequence_3” are internally used to separate XOR output triggers into separate channels. The push-pull pad control of port pins used to drive external LC circuits by these separated trigger signals are also enabled by the “sequence_1”, “sequence_2” and “sequence_3” signals internally.
- After the trigger period is over, the pad control logic (by using LCSensePushPullALT0_CHn, LCSensePushPullALT1_CHn functions of Port pins) helps the port pins to switch to comparator input direction by FlexLC glue logic hardware. After this only, the oscillations on LC circuits can start.
- The timing and period of QTMR1.CHn are carefully selected to allow the damping of oscillations to decay for about 70 seconds. The trigger for LC2 channel is done after QTMR1.CH0/1/2 period of 140 usec, therefore allowing more time than 70 usec oscillations for LC1 circuit. Similarly, the trigger for LC3 channel is done after another 140 usec.
- The comparator output is seen to convert oscillations of LC1, LC2, LC3 into pulses. These pulses follow the timing of LC1, LC2, and LC3 oscillations. The frequency of pulses is 496 kHz varying duty cycle.
- QTMR0.CH0 counts the pulses from comparator for LC1, LC2, and LC3 oscillations. The counter output toggles if the count is more than a preset compare-threshold value. Counter output does not toggle if the pulses are less than the preset compare-threshold value. This happens when there is metal part of the half-metal disc in the vicinity.
- The serialized QTMR0.CH0 toggling output is sampled and separated into LC_ROT_SOC_LOGIC_OUT1, LC_ROT_SOC_LOGIC_OUT2 and LC_ROT_SOC_LOGIC_OUT3 signals. These signals are phase shifted by 120 degrees to each other, only within the overall sampling duration of 450 usec, and not the overall LPTMR per-trigger sampling duration of 5 msec. This usually is not an issue for the QTMR0.CH0 in quadrature decoder (QDC) mode counter, as the nonactivity of the ACMP_AON round-robin comparator for the rest of (5 msec – 450 usec =) 4.55 msec helps to reduce current consumption.
- The phase shifted signals LC_ROT_SOC_LOGIC_OUT1, LC_ROT_SOC_LOGIC_OUT2 and LC_ROT_SOC_LOGIC_OUT3 are connected to memory mapped to SYSCON_AON register LC_ROT_LOGIC[CH_COMP_OUT], which read by software executed by CM0+ CPU by periodic wake-up. The content of LC_ROT_LOGIC[CH_COMP_OUT] is not sticky; therefore, it must be read by CM0+ CPU before being overwritten by another fresh LC sense sequence within another 5 msec.

Figure 25 shows the oscilloscope trace of three LC circuits. “parked” out signal is used to power the op-amp in FLO-SNSR-BRD. cmp_in_1, cmp_in_2, and cmp_in_3 are the triggers (XOR o/p) cum comparator inputs generating oscillations. These four signals are only active for three LC circuits external to the MCX L255 MCU.

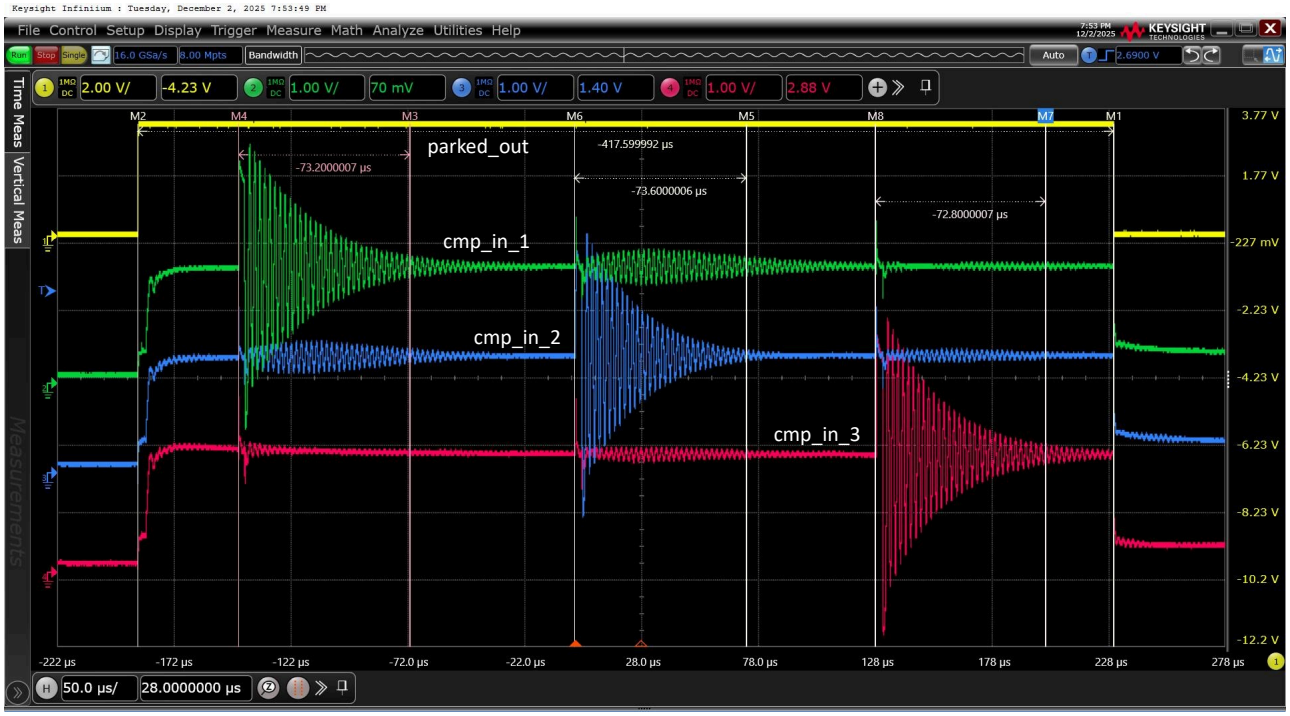


Figure 25. Three LC sensor oscilloscope traces

Figure 26 shows the oscilloscope trace of three LC circuits, with an additional debug signal to see the comparator output generated internally. The CMPO (internal) shows the oscillations from cmp_in_1, cmp_in_2 and cmp_in_3 are converted into pulses with varying amount, depending on the oscillation inputs. This signal can be brought out of MCX L255 for diagnostics purposes during development (and few more such internal signals) but must be disabled after the tune-up or debug is done, to save port power.

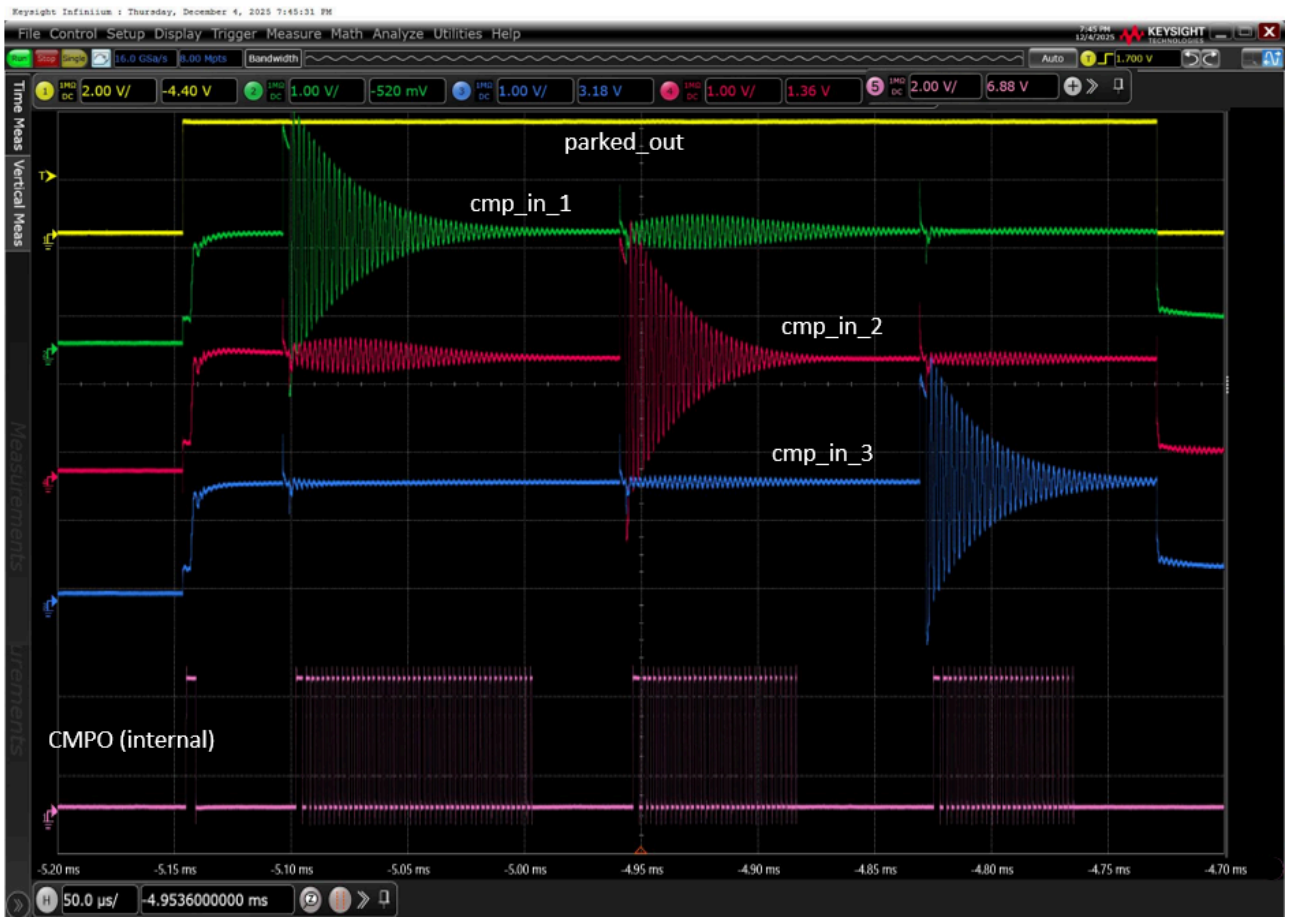


Figure 26. Three LC sensor oscilloscope traces with comparator outputs on air

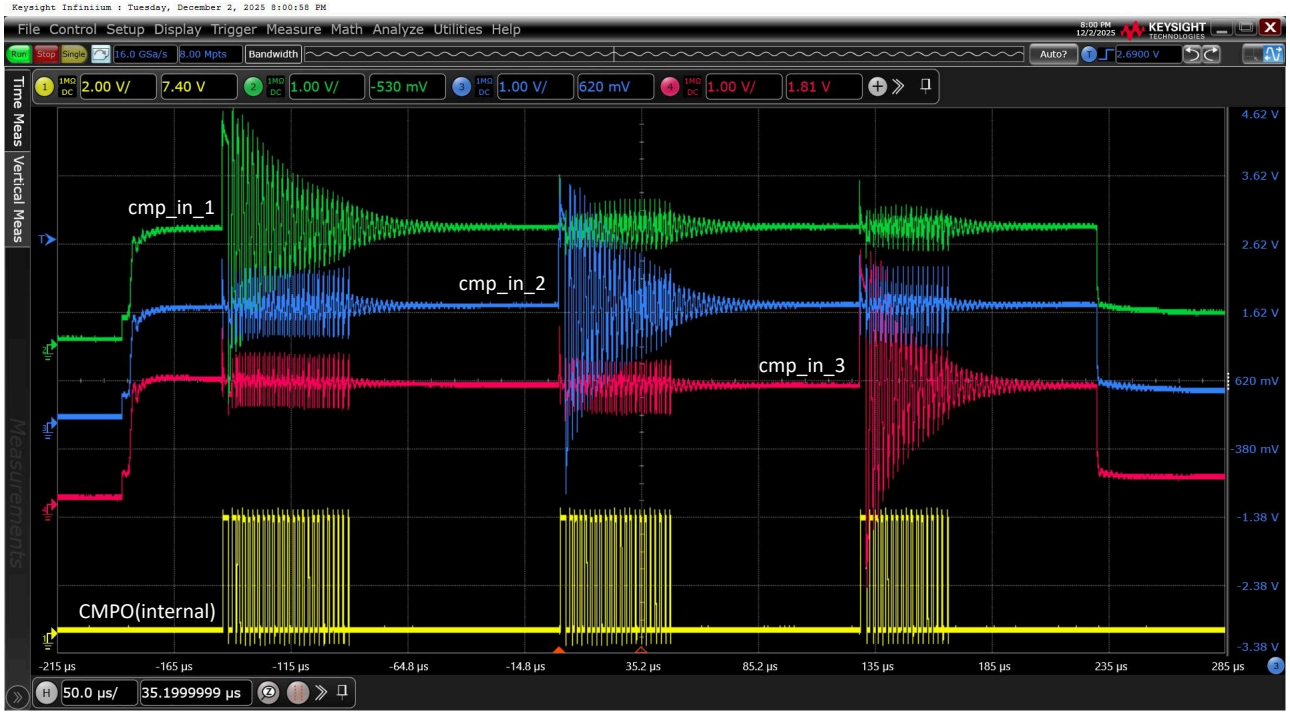


Figure 27. Three LC sensor oscilloscope traces with comparator outputs with air, metal combination

From the oscilloscope traces, the proximity and traces of LC circuits to each other can influence each other due to mutual induction, which can result in one signal to generate mild oscillations into another. Care must be taken during the placement and routing of these signals during PCB placement and layout to avoid this intersymbol crosstalk. Due to the round-robin feature of the ACMP_AON comparator, the influence in one LC channel signal does not directly impact the other channel during the comparison channel sequencer window. But due to too much proximity, the signals crosstalk can be higher to disturb the oscillation signal itself, resulting in difficulty for ACMP_AON to isolate the signals from each other.

5 Software description

This section describes the software implementation of the MCX L255 LC sensor demo. The software application consists of configuration of the needed peripherals, the measurement, and calculation of the half-metal disc rotation, LCD display, and user interactions. The demo software also demonstrates how the calibration is done for LC sense circuits. The application software has been written in C-language and compiled using the IAR Embedded Workbench for ARM (version 9.70.1), with high optimization for execution speed. The software application is based on the NXP's MCX L255 bare-metal MCUXpresso SDK software drivers. The software running in MCX L255 is divided into two parts. The CM33 software runs out of Flash memory with SRAM in the main domain. It also loads the software for CM0+ in the AON domain once after power-on reset is done.

The functions of main domain software are as listed below:

- Initialization of clock for CM33 and main domain peripherals
- Initialization of Port pins for GPIO or other IP-specific alternative port functions
- Initialization of an LCD driver and glass parameters
- Initialization of timer to assist the demo application time tick
- Initialization of a push button switch for user interaction
- Configuration of RTC oscillator (32.768 kHz) and starting calendar for real-time timekeeping

- Copy AON CM0+ software from Flash to SRAM in AON domain, and then start CM0+ to execute the same from AON SRAM
- Initialization of FlexLC modules (details below) for the sensor part
- Waiting for AON CM0+ to finish initialization and send an interprocessor message for sync up with CM33 CPU
- Waiting in the main application for the asynchronous interrupts/events like timer interrupts and push button interrupts, LCD message display, serial communication for calibration request and services for the same
- Going to sleep mode if a timeout occurs

The functions of AON domain software are as listed below:

- Initialization of clock for CM0+ (Initialization of all AON domain peripherals are already done by main domain CM33 software)
- Waiting for interprocessor message for sync-up with CM33 CPU
- Enable an RTC alarm to wake up CM0+ CPU after every 1 second (configurable)
- Read disc rotation count and store to AON SRAM
- Calculate disc rotation rate
- Wait for CM33 to send a request to go to deep-sleep-mode (2).

5.1 Main domain software

The modules used in the main domain software are initialized in the sequence described in this section. Post initialization, the main domain software runs in an endless loop until a predefined timeout, while servicing interrupts due to asynchronous events.

5.1.1 Clock setting

The CM33 CPU is run with the FRO96 internal Free running oscillator. In this demo application, the clock setting for CM33 is set to highest 96 MHz, but can be done with lower frequencies like 48 MHz, 24 MHz or 12MHz.

The peripherals in main and AON domain are clocked and functional with the clock rates as listed in Table 3.

Table 3. CPU and peripheral clock settings

Module	Clock rate	Clock source
CM33 CPU	96 MHz	FRO96
CM0+ CPU	3 MHz	FRO3M AON
PORT0,1,2,3	24 MHz	FRO96 ÷ 4
GPIO0,1,2,3	24 MHz	FRO96 ÷ 4
SYSTICK timer	96 MHz	FRO96
SLCD	16 kHz	16.384 kHz from ROSC
ROSC	32.768 kHz	External crystal
LPUART0	96 MHz	FRO_HF_DIV (from FRO96)
QTMR0,1	1.5 MHz	FRO3M AON ÷ 2
ACMP.AON	1.5 MHz	FRO3M AON ÷ 2
LPTMR0.AON	1.5 MHz with internal divider 16	FRO3M AON ÷ 2

5.1.2 FlexLC module initialization

This section describes how the FlexLC components are configured for the demo application. Refer to the MCXL25x Reference Manual document for a detailed understanding of:

- QTMR
- LPTMR
- AON ACMP
- AON INPUTMUX
- Clock module AON SCG
- AON SYSCON
- FlexLC

The initialization is done only once by CM33 after POR reset.

5.1.2.1 Clock configuration

The AON clock is set at 3 MHz with an internal FRO3M AON oscillator. The RTC runs at a 16.384 kHz rate with the external 32.768 kHz clock. The 1.5 MHz clock divider from FRO3M AON is used as the functional clock for the timers and comparator used for LC sensors, with internal dividers for operation.

5.1.2.2 Pretrigger configuration

The AON. LPTMR0 is clocked by a prescaler divider of value ‘8’ on the incoming clock of 1.5 MHz from the FRO3M AON RC oscillator. This results in a clock frequency of 93.75 kHz clock for LPTMR operation. The compare value is set to ‘468’ to reach the compare value at about 5 msec.

Select INPUTMUX_AON:

1. LCSENSE_SEQ_PTRIG_GLUE_IN input to QTMR1.CH3 output.

5.1.2.3 Primary trigger configuration

The QTMR1.CH3 one-shot time is effective to sample LC circuits periodically. This timer is to simulate the AON.ACMP0 initialization delay (112 usec). This counter uses SCS as LPTMR output to receive shots to start every time.

- Set QTMR1.CH3 in one-shot mode for 112 usec. This counter also reinitializes QTMR1_Coutner0/1/2.
- Set CTRL0 = 0xD084
  1. CM = 110b (one-shot)
  2. PCS = 1000b (1.5 MHz clock)
  3. SCS = 1b (Counter input 1)
  4. LENGTH = 0b
  5. COINIT = 0b
  6. OUTMODE = 100b
- Set COMPARISON_1, COMPARATOR_LOAD_1 for 112usec to generate the initial delay to sequencer logic to enable QTMR1.CH0/ QTMR1.CH1/ QTMR1.CH2.
- Set COMPARISON_2, COMPARATOR_LOAD_2 for 496 usec to allow QTMR1.CH0/ QTMR1.CH1/ QTMR1.CH2 to operate and then force these timers to be disabled, which are operating in gated-count modes (configuration mentioned in later subsection).
- CSCTRL0 = 0
- Set SCTRL0 = 0x24
  1. MSTR = 1b (Master mode)
  2. FORCE = 1b (Force other channels in QTMR.CHn)

QTMR1.CH3 is set as “Master” to all QTMR1 channels to re-initialize/reset once in every 5 msec.

Select INPUTMUX_AON:

1. QTMR1_TMR[0] input to LC Sense primary trigger glue output.

QTMR1_TMR[1] = AON.LPTMR0 output.

5.1.2.4 Sequencer configuration

QTMR1.CH0 (Gated count mode) with 140 usec period is used to run the Sequence logic.

- Set QTMR1.CH0 period = 140 usec.
- Set COMPARISON_1, COMPARATOR_LOAD_1 for 20 usec @ 1.5 MHz clock period
- Set COMPARISON_2, COMPARATOR_LOAD_2 for 120 usec @ 1.5 MHz clock period
- CSCTRL0 = 0b
- SCTRL0 =0x10

1. EEOF = 1b This bit enables the compare from another counter/timer within the same module to force the state of this counter's OFLAG output signal.

- Set CTRL0 = 0x702C // set mode and start
  1. CM = 011b (gated count)
  2. PCS = 1000b (1.5 MHz clock)
  3. SCS = 0b (Counter input 0)
  4. LENGTH = 1b
  5. COINIT = 1b
  6. OUTMODE = 100b

The other part of sequencer configuration is done through FlexLC glue logic using SYSCON_AON register SEQUENCE_TICK[SW_CHANNEL_EN] for the number of channels that is, two or three channels.

Set SYSCON_AON register SEQUENCE_TICK[SW_CHANNEL_EN] = 001b for two channels, 011b for three channels.

Select INPUTMUX_AON:

1. LCSENSE_SEQ_TICKS_GLUE_IN input to QTMR1.CH0 output.

5.1.2.5 Trigger pulse generation

Set the QTMR1.CH1 period to 140 usec. 64 usec low-state time, 76 usec high-state time.

- Set COMPARISON_1, COMPARATOR_LOAD_1 for 64 usec
- Set COMPARISON_2, COMPARATOR_LOAD_2 for 76usec
- CSCTRL1 = 0
- SCTRL1 =0x10

1. EEOF = 1b This bit enables the compare from another counter/timer within the same module to force the state of this counter's OFLAG output signal.

- Set CTRL1 = 0x702C
  1. CM = 011b (gated count)
  2. PCS = 1000b (1.5 MHz clock)
  3. SCS = 0b (Counter input 0)
  4. LENGTH = 1b
  5. COINIT = 1b
  6. OUTMODE = 100b

Set QTMR1.CH2 period for 140 usec. 67 usec low-state time, 73 usec high-state time.

- Set COMPARISON_1, COMPARATOR_LOAD_1 for 67 usec
- Set COMPARISON_2, COMPARATOR_LOAD_2 for 73usec
- CSCTRL1 = 0
- SCTRL1 = 0x10

1. EEOF = 1b This bit enables the compare from another counter/timer within the same module to force the state of this counter's OFLAG output signal.

- Set CTRL1 = 0x702C
  1. CM = 011b (gated count)
  2. PCS = 1000b (1.5 MHz clock)
  3. SCS = 0b (Counter input 0)
  4. LENGTH = 1b
  5. COINIT = 1b
  6. OUTMODE = 100b

After configuring QTMR1.CH1 and QTMR1.CH2, the FlexLC XOR gate inputs are needed to configure via Input-mux. Select INPUTMUX_AON:

1. soc_glue_XOR0_in0 input to QTMR1_tmr1_output,
2. soc_glue_XOR0_in1 input to QTMR1_tmr2_output

Figure 28 shows both QTMR1.CH1, QTMR1.CH2 and XOR output.

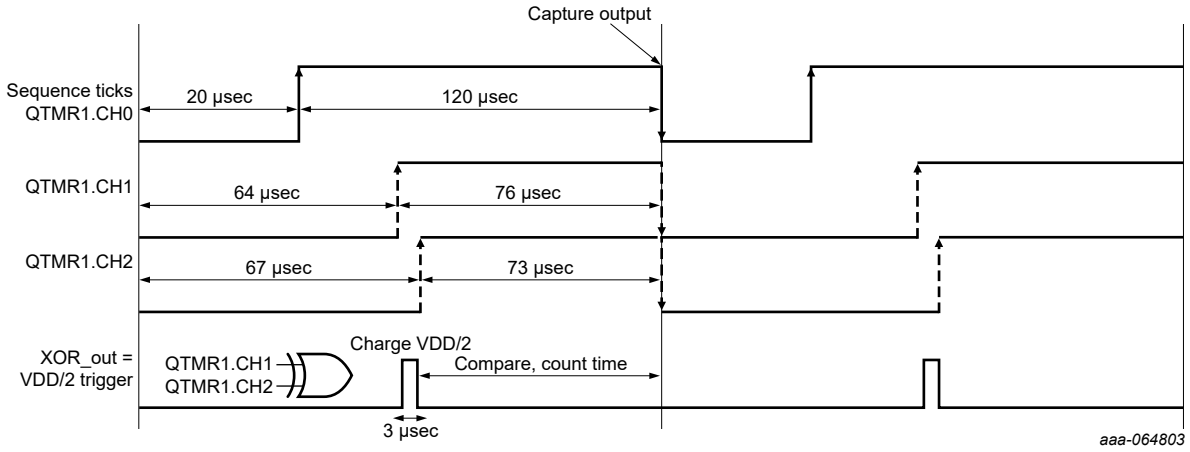


Figure 28. Trigger pulse generation

5.1.2.6 AON.ACMP configuration

AON.AMCP DAC reference selection: The DAC inside AON.ACMP used VDD (3.3V) as its reference with VREFH0 = VDD as selection. There is a possibility to use VREFH1 = VREFI available through U6 Op-amp on FLO-SNSR-BRD and can be connected to P0_0 pin of MCXL25x as VREFI ALTn port input.

Using ACMP for LC sense: The ACMP module has been used in round-robin mode for LC sense. In this case, the round-robin is triggered by the LPTMR output periodically. At each trigger, the comparator does a comparison of channel inputs 0, 1, and so on, in a round-robin fashion. After one round is over, the comparator is disabled to save power. The comparator is activated again with the next trigger from LPTMR output. Therefore, the round-robin comparison happens once in 5 msec. The duration of the round-robin is also approximately synchronized with the oscillation generation, which is also triggered by the same LPTMR.

Below is the defined configuration for LPCMP:

1. Select RR trigger input from LPTMR0 output using INPUTMUX_AON:

CMP0_RR_TRIG [INP] = 16 (LPTMR0 output)

2. Set ACMP DAC control DCR = 0x008A0001. This sets the DAC output to 1.78V.
3. Set CCR0 = 0x2
4. Set RRCR0 = 0xF290201
   a. Configures RRCR0[RR_NSAM] = 0b
   b. Configures CMP initialization delay RRCR0[RR_INITMOD] = 42
   c. Configures RRCR0[RR_SAMPLE_CNT] = 16
   d. Configures RRCR0[RR_TRG_SEL] = 0b (external trigger).
   e. Enables RR trigger mode by setting RRCR0[RR_EN] to 1b.
5. Configure RRCR1 = 0x00710001 (Channel 7 / DAC reference, Channel 5 only input for round-robin)
6. Configure channels for comparison by RRCR1[RR_CHnEN] = 101b (Channel 5 only)

Set CCR0 = 0x1 to enable ACMP.

Select INPUTMUX_AON:

1. SOC_GLUE_CMPPADS_PCTRL_XOR_IN0 input to FlexLC XOR output,
2. SOC_GLUE_CTRLPADS_PCTRL_XOR_IN0 input to FlexLC XOR output

5.1.2.7 LC Pulse counter configuration

Set QTMR0.CH_0 in triggered-count mode 2. The primary source of this counter is connected to the output of ACMP while the secondary (trigger input) is connected to the output of QTMR1.CH2.

1. Set QTMR0.CH0 as below:
   a. Set COMPARISON_1 = COMPARATOR_LOAD_1 = 21 as the threshold for pulse count
   b. Set SCTRL0 = 0
   c. Set CSCTRL[TCI] =1b TCI bit enables the counter to be reinitialized when a trigger occurs at secondary input. It resets the counter after every count period effective for an LC circuit operation(oscillations).
   d. Set CTRL0 = if set mode and start
      i. CM = 110b
      ii. PSC = 0b
      iii. SCS = 1b
      iv. OUTMODE = 101b (set on compare, clear on secondary input edge)

Figure 29 shows the input connection of QTMR0.CH0. The inputs are routed from INPUTMUX_AON:

1. QTMR0.TMR[0] input to AON. ACMP output,
2. QTMR0.TMR[1] input to QTMR1.CH2 output

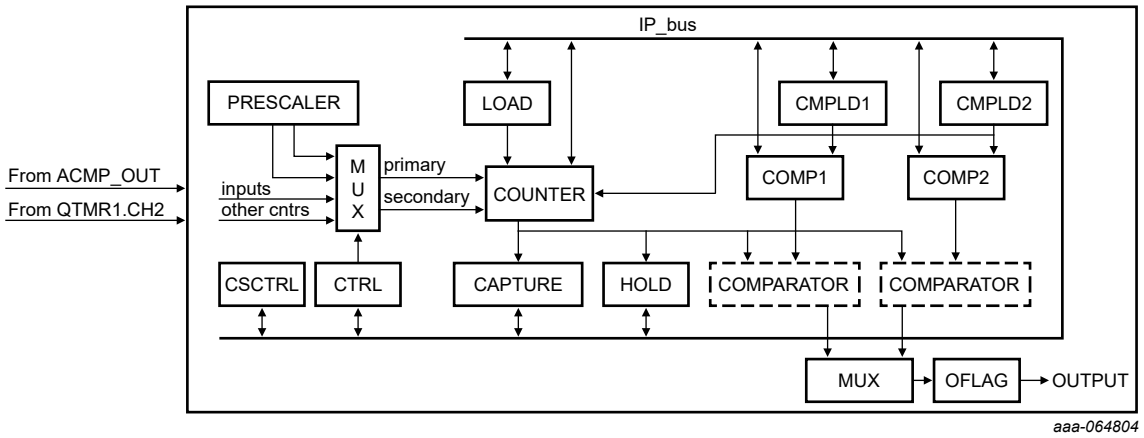


Figure 29. QTMR0.CH0 triggered mode-2 input connections

5.1.2.8 LC Pulse encoder configuration

Two inputs of QTMR0.CH1 are used as PHASEA, PHASEB inputs of the counter in Quadrature mode. This is done either for two LC configuration or for three LC configuration with the third channel not used for disc rotation counting, but for tamper detection.

Set QTMR0.CH1 in quadrature-count mode.

1. Set QTMR0.CH1 as
   a. SCTRL0 = 0b
   b. CSCTRL = 0b
   c. Set CTRL0 = 0x8580
      i. CM = 100b
      ii. PSC = 10b
      iii. SCS = 11b

Figure 30 shows the input connection of QTMR0.CH1. The inputs are routed from INPUTMUX_AON:

1. QTMR0.TMR[2] input to FlexLC LC_ROT_SOC_LOGIC_OUT1,
2. QTMR0.TMR[3] input to FlexLC LC_ROT_SOC_LOGIC_OUT2

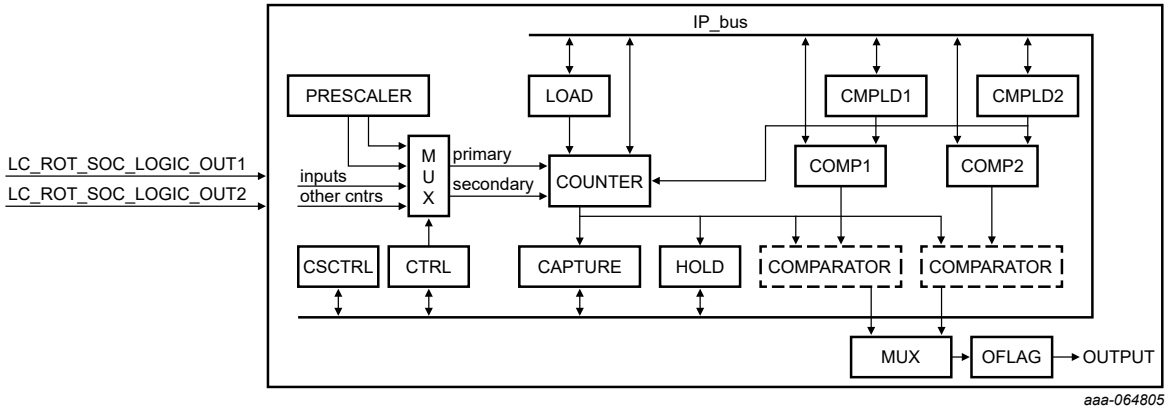


Figure 30. QTMR0.CH1 QDC mode inputs connection

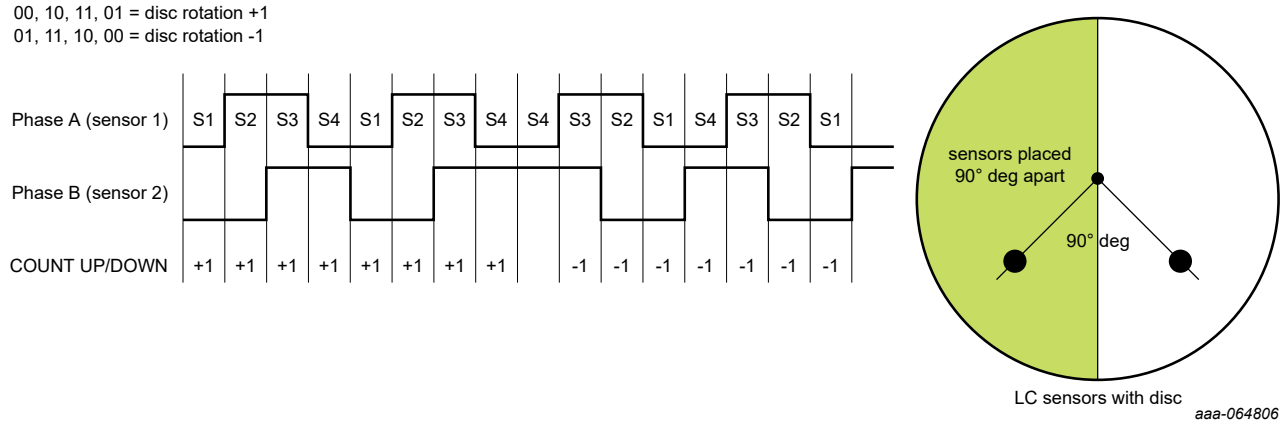


Figure 31. QTMR0.CH1 QDC encoding with LC disc

5.1.2.9 LC Pulse rotation reader configuration

When all three LC circuits are used to detect disc rotation, QTMR0.CH1 is used to periodically wake up the CM0+ CPU. The initialization of this timer channel is done from CM33 main domain software.

Set QTMR0.CH2 in one-shot mode.

Set QTMR0.CH1 as

1. SCTRL0 = 0
2. CSCTRL = 0x40 (TCF1EN set)
3. Set CTRL0 = 0xD084
   a. CM = 110b (one-shot)
   b. PCS = 1000b (1.5 MHz clock)
   c. SCS = 1b (Counter input 1)
   d. LENGTH = 0b
   e. COINIT = 0b
4. OUTMODE = 100b

Select INPUTMUX_AON:

1. LC_ROT_SOC_LOGIC_IN input to QTMR1.CH0 output.

5.1.2.10 Port pins configuration

The port pins used to drive LC circuits are configured in alternative functions as defined in the attachment Pinout spread sheet of the reference manual document of MCX L255. Table 4 provides a summary of pins that are used for LC sense demo operation. More such port pins alternative functions are listed in FlexLC chapter in the reference manual document, the **External Interfaces** section.

Table 4. Port pins used for LC sensor circuits

Pin	Type	Description
P0_7	Output	Pin alternate function ALT1(AONTRIG_OUT1) brings out the signal generated by FlexLC glue logic “parked_out” being routed by INPUTMUX_AON multiplexed configuration.
P0_1	Analog input / Output	The alternate digital function ALT7 = LCSenseCMP_CH0 from signal multiplexing, along with the analog function LCSense_CMP_CH0 has been used to trigger and compare an external LC Sense circuit.

Table 4. Port pins used for LC sensor circuits*...continued*

Pin	Type	Description
P0_2	Analog input / Output	The alternate digital function ALT7 = LCSenseCMP_CH1 from signal multiplexing, along with the analog function LCSense_CMP_CH0 has been used to trigger and compare an external LC Sense circuit.
P0_3	Analog input / Output	The alternate digital function ALT7 = LCSenseCMP_CH2 from signal multiplexing, along with the analog function LCSense_CMP_CH0 has been used to trigger and compare an external LC Sense circuit.

Among the port pins mentioned in Table 4, pins P0_7, P0_1, P0_2 are used both the cases of two-LC or three-LC circuits, and P0_3 is used only in three-LC circuit.

5.1.3 SLCD initialization

The SLCD driver initializes itself for 4 backplane (COM) and 16 frontplane to interface external glass FTP12557. This is done by using the SDK driver API SLCD_Init() and followed by SLCD_Engine_Init() of display specific SLCD engine.

The initialization is done every time CM33 is reset by POR or wake-up.

5.1.4 Push button initialization

The demo application supports one SW5 push button, which is used to scroll through messages on the LCD glass and to wake the MCX L255 from deep-power-down sleep mode. The push button is initialized with the SDK driver APIs according to the port pins on the FRDM-MCXL255 board. An interrupt is generated on the rising edge of the port pin assigned to SW5.

The initialization runs each time the CM33 is reset by a POR event or a wake-up.

5.1.5 SysTick initialization

The SYSTICK timer in the ARM CM33 core subsystem is used to generate a time base (tick) for the application. The SYSTICK period is 25 ms, and its interrupt service routine (ISR) is used to update the LCD messages.

The initialization runs each time the CM33 is reset by a POR event or a wake-up.

5.1.6 LED initialization

The demo software uses the on-board RGB LED to indicate calibration status and to signal normal operation during active mode. The three port pins that drive the R-G-B LED are initialized as GPIO outputs using the SDK driver APIs.

The initialization runs each time the CM33 is reset by a POR event or a wake-up.

5.1.7 FlexLC Tamper initialization

The FlexLC glue logic has inbuilt logic to detect any tampers occurred due to disturbances or the removal of LC circuits under operation. This logic works alongside the disc rotation count mechanism described in the document. The configuration of two, three or four LC circuit to be considered for tamper detection is done through SYSCON_AON register SEQUENCE_TICK[SW_CHANNEL_EN] for the number of channels with TAMPER_FLT_EN[LC_TAMPER_FAULT_ENABLE] bit set to enable the interrupt due to tamper condition. The tamper condition can be used to wake up the either or both of CM33 and CM0+ CPUs from deep power down modes or interrupt an ongoing process running through the processors. The wake-up configurations can be done by using SMM block register AON_CPU[WKUP_SRC_AON_CPU] value set to ‘14’.

The initialization is done only ONCE after by CM33 after POR reset.

The FlexLC glue logic includes built-in mechanisms to detect tamper events caused by disturbances or the removal of LC circuits during operation. This logic operates alongside the disc-rotation count mechanism described in this document. The configuration for using two, three, or four LC circuits for tamper detection is controlled through the **SYSCON_AON** register:

- **SEQUENCE_TICK[SW_CHANNEL_EN]** selects the number of active channels, and
- **TAMPER_FLT_EN[LC_TAMPER_FAULT_ENABLE]** enables tamper-fault interrupts.

A tamper condition can wake either CPU or both CPUs (CM33 and CM0+) from deep-power-down modes, or it can interrupt an ongoing process running on the processors. Wake-up sources are configured through the **SMM** block register **AON_CPU[WKUP_SRC_AON_CPU]**, which must be set to `14`.

Initialization is performed **only once** by the CM33 after a POR reset.

5.1.8 LPUART initialization

The onboard MCU-LINK serial port on the FRDM-MCXL255 board is used to communicate with the LC-Calib Monitor tool, as described in this document. The baud rate is set to 115200, with 8 data bits and no parity and hardware flow control.

The initialization is done every time CM33 is reset by POR or wake-up.

5.1.9 Wake-up initialization

The demo software has been configured to take the MCU in deep-power-down (2) mode periodically. The wakeup configuration included a push button and an RTC alarm as show below in Table 5.

Table 5. Wake-up sources

Wake-up source	Wake-up destination	Description
SW5 Push button on P0_9/EXT_INT	CM33 CPU	This button is configured as a wake-up source from AON peripheral to main domain CM33
RTC ALARM0	CM0+ CPU	This ALARM is set to wake up only CM0+ CPU periodically every 1 second (configurable by a C-macro)

The initialization is done before every time the MCX L255 goes into deep-power-down (2) mode.

5.1.10 MU_A messaging unit

The messaging unit has two ports to help two CPUs to communicate to each other asynchronously. The side A of the MU peripheral is configured here in the main domain software, with interrupts enabled to receive notification from CM0+ CPU as it is connected to side B of the MU block.

5.1.11 RTC timer

ROSC is part of AON domain peripherals but it is configured by the CM33 main domain software. The oscillation is configured and started once.

Initialization is done every time the CM33 is reset by POR or wake-up.

5.1.12 Main domain software RUN mode

The main domain software is run by CM33. CPU is described with the help of a flow diagram shown in Figure 32.

The starting points of the software can be the following:

1. After Power On Reset (POR)
   The CM33 CPU software does the initialization of main and AON domain peripherals. Then, it enables the EXT_INT (SW5) pin interrupt, Systick interrupt, and MU_A interrupt. After this, it waits for the push button and Systick asynchronous interrupts endlessly before going to deep-power-down (2) Sleep mode.
2. After wake-up from deep-power-down modes,
   The CPU clock and peripherals are reinitialized, except for peripherals that require one-time-only initialization. It then waits for the push button and Systick asynchronous interrupts endlessly before going to deep-power-down (2) Sleep mode.

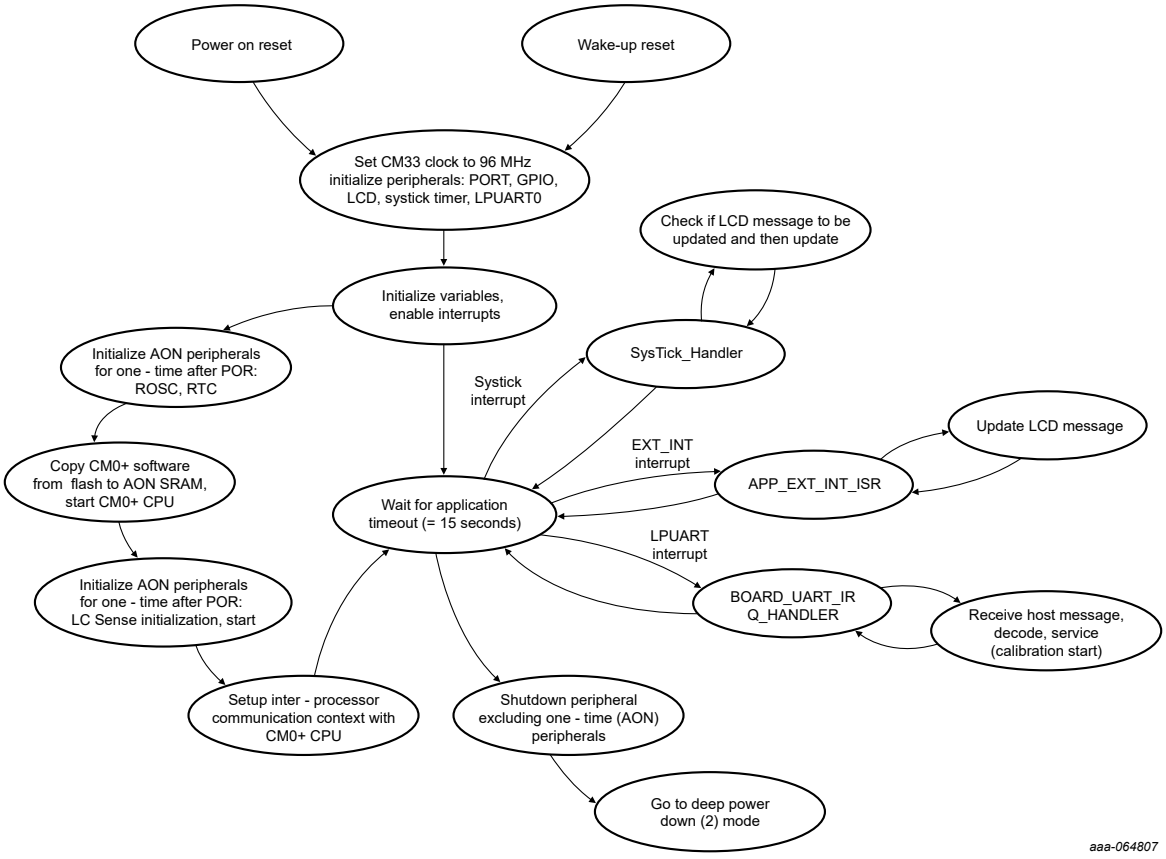


Figure 32. CM33 main domain software flow diagram

After initialization, the software waits for asynchronous events as mentioned below:

- A Systick interrupt is generated every 25 msec, thereby giving the required granularity to count 1 second and more. The 1 second resolution helps to update specific display messages like RTC time values with better resolution. In this ISR, the LCD message is updated periodically. Systick timer is also used to count up to PB_ACTIVE_TIMEOUT (the default C-macro value is '15') to track the active mode time of the main domain application.
- A push button interrupt is generated as the switch SW5 is pressed. It is used to scroll through the LCD messages quicker than the time period auto adjusted by the Systick ISR. This ISR also resets the active mode time to PB_ACTIVE_TIMEOUT value.

- LPUART Interrupt Service Routine (ISR) is invoked when a message is received from the onboard USB-Serial COM port, which can be connected to a PC host serial configuration tool. In the demo, this ISR is used to receive the calibration request from the PC host tool LC-calib Monitor. The receipt message is decoded, parsed, and handled by `comm_command_proc_serial.c/.h` source files. The response to the command is also sent back to the PC host. The detailed description of the command-response of LPUART protocol is out-of-scope.
- If no other asynchronous events/interrupts are pending to be handled and the application time out is reached, CM33 software shuts down peripherals that are not required to operate anymore. However, it does not stop the AON peripherals like ROSC, RTC, AON GPIOs, LC sense modules that are required to continue operation during deep-power-down (sleep) modes.
- Finally, CM33 software configures and starts the process to go to deep-power-down (2) mode, by configuring power management modules (CMC, SMM) and through communication with CM0+ software accordingly.

5.2 AON domain software

The AON domain software is stored in the flash of the main domain. After a POR reset, the main-domain CM33 CPU loads this software and enables the CM0+ CPU to run it from AON SRAM.

The primary function of the AON domain software is to periodically count the disc rotation, calculate the disc flow rate, and store them in AON SRAM. The flow volume rate calculation is not done here. It becomes the function of a flow/gas meter application, which this demo software does not focus on. Usually, the flow volume is calculated by flow/gas meter application as a multiplier unit to the disc rotation count, which can vary depending on the liquid type and the pipe dimension.

With the main domain software doing the AON peripheral initialization, the AON domain software is left with little requirement of initialization. The initialization and run mode software is described below.

5.2.1 Other peripherals initialization

After a POR reset or wake-up, the AON software that is executed by the CM0+ CPU must set up the CPU clock to 3 MHz. After this, the interrupt for the MU_B side of the messaging unit is enabled. It helps the main domain CM33 CPU to communicate to CM0+ CPU in the AON domain.

Only once after POR reset, an ALARM0 of the RTC module is enabled to raise an alarm interrupt to CM0+ CPU at every one second.

5.2.2 AON domain software RUN mode

After the first time of POR reset, and after initialization is done, the CM0+ CPU software waits for a sync-up message from the CM33 CPU software in the main domain. The synchronization happens with MU_B interrupt because of a message request from MU_A from the main domain. In turn, a callback **APP_SecondaryCoreCallback()** is registered by the AON application to store necessary backup data before entering deep-power-down (2) mode by the SDK power handler driver.

Entering deep-power-down (2) is done with the power management SDK driver functions executing concurrently in both the main domain and the AON domain software. The process begins with a message from the main domain side through the MU_A interrupt but ends with the power-down entry function executed by the wait-for-interrupt (WFI) of ARM CPU instruction. This process happens when an application timeout occurs in the main domain software, which triggers the deep-power-down (2) entry sequence.

5.2.2.1 Two-LC AON software RUN mode

The purpose of raising the RTC alarm every second is to read and save the disc rotation count from the QTMR0.CH1 QDC mode counter. The rotation count is saved in AON SRAM, which is always retained even in deep-power-down (2) mode.

The CM0+ software runs in active mode until the main domain application software is active. After this, the MCU enters in deep-power-down (2) mode.

Only the CM0+ CPU is wakened up every second due to alarm. During this time, the execution mode is called deep-power-down (1) mode as the main domain CPU does not work and is in shutdown mode. The CM0+ wake-up software helps to read the latest QTMR0.CH1 QDC counter for two LC circuits and store it in SRAM memory. Also, it calculates the disc rotation rate. After this, the CM0+ software configures MCX L255 in deep-power-down (2) again with the help of the SDK power handler driver.

Waking up back to Active mode can happen by a button press on SW5, as shown in Figure 33.

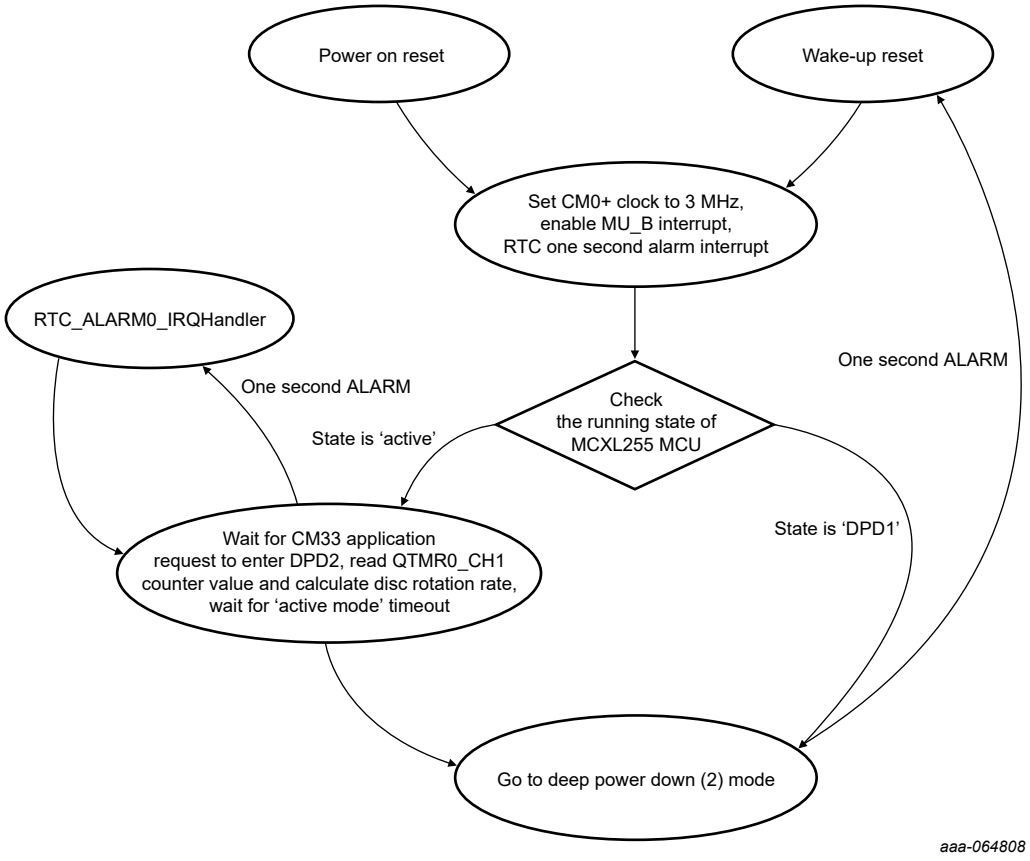
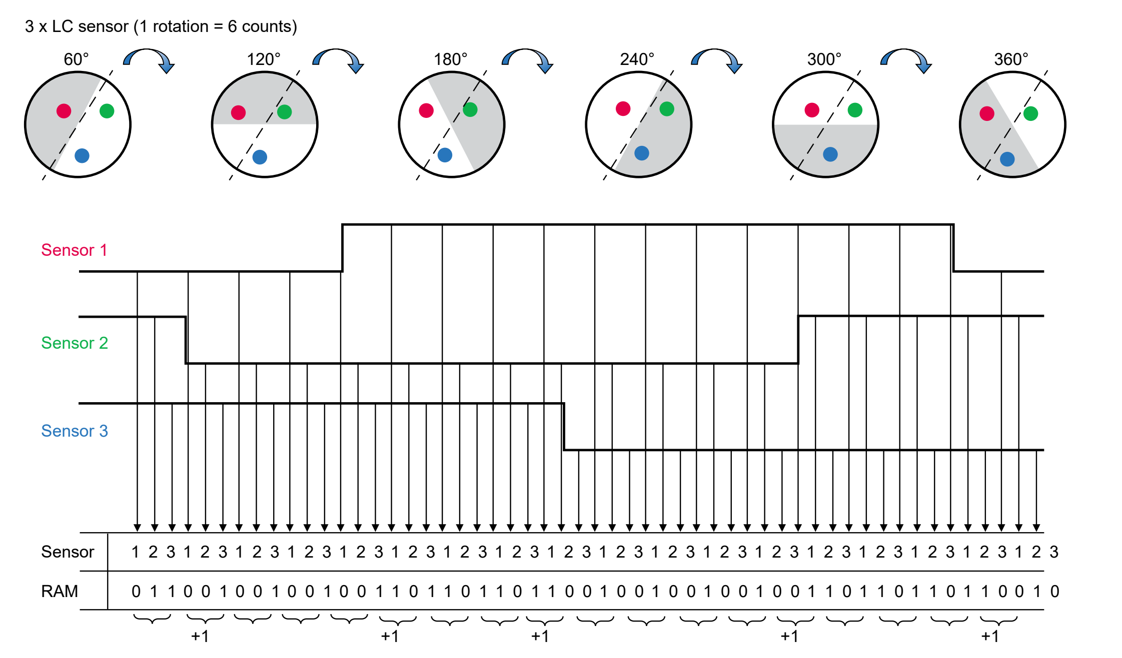


Figure 33. Two-LC CM0+ AON domain software flow diagram

5.2.2.2 Three LC AON software RUN mode

FlexLC hardware glue logic is not able to calculate disc rotation with three of four LC circuits. However, the rotation of the half-metal disc can be calculated with a little processing done in deep-power-down (1) mode of CM0+ CPU. For this, the QTMR0.CH1 is pre-configured for periodic wake-up.

After each sequence, of LC circuit oscillations are counted and rotation states are stored in the AON_SYSCON LC_ROT_LOGIC[CH_COMP_OUT] register bits. With the help of the periodic wake-up with the QTMR0.CH1 timer, CM0+ software can read the channel states and check for state transitions. With three LC channels, only the states as shown in Figure 34 are permissible, which is checked by the CM0+ software. Once 6 sequential state changes are identified, the disc rotation count is incremented by +1 or -1, depending on the direction of disc rotation.



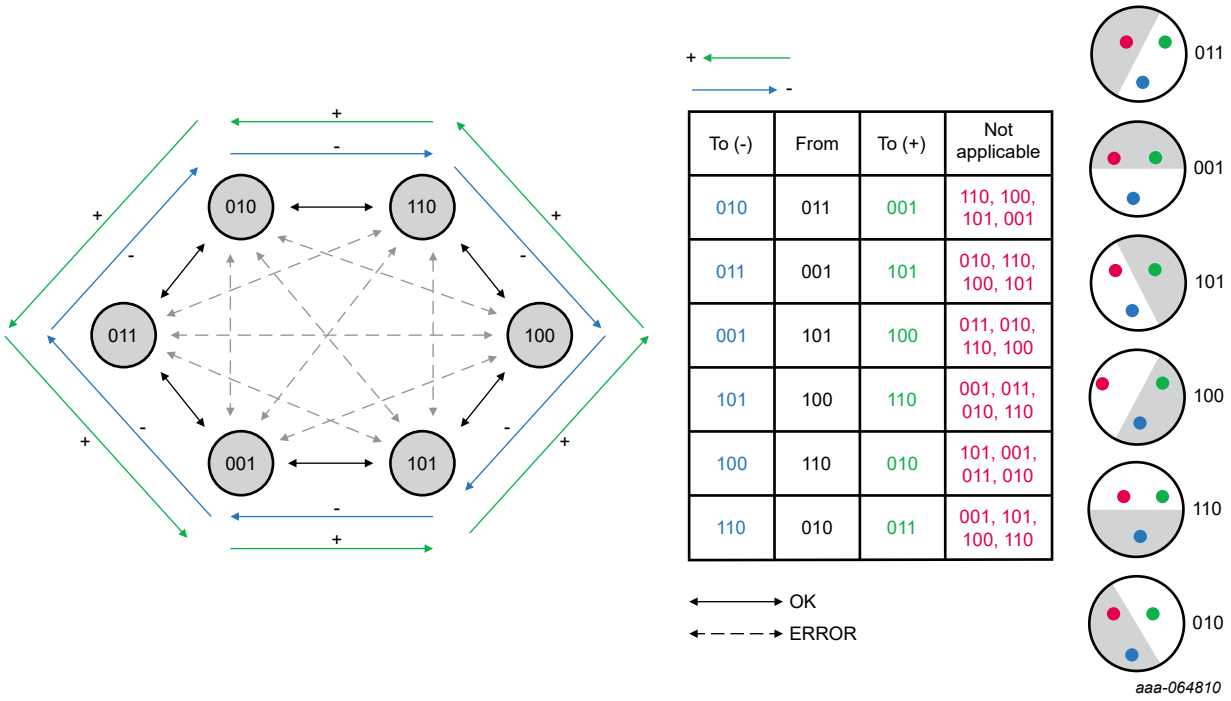
RAM	Count
011	0
001	1
001	1
001	1
001	1
101	2
101	2
101	2
100	3
100	3
100	3
100	3
100	3
110	4
110	4
110	4
110	4
010	5

Figure 34. Three-LC state capture into AON SRAM

In Figure 35, the rotation of the half-metal disc with three LC circuits is depicted with their states decoded by QTMR0.CH1 as described in sections Section 4.5. The state changes slower than the sampling frequency, but the interim count will be increased only when a state change is observed, for example, from initial 011b to 001b. As we can see, one complete half-metal disc rotation causes six state changes. Therefore, six state changes amount to one rotation, and is used to update the disc rotation counter.

Note: *The above logic, using two LC circuits results in four state changes.*

Figure 35 shows the permissible state changes with three LC circuits, which the CM0+ software checks for validation.



To (-)	From	To (+)	Not applicable
010	011	001	110, 100, 101, 001
011	001	101	010, 110, 100, 101
001	101	100	011, 010, 110, 100
101	100	110	001, 011, 010, 110
100	110	010	101, 001, 011, 010
110	010	011	001, 101, 100, 110

Figure 35. Three-LC state transition table

The purpose of raising the RTC one-second alarm is to read disc rotation count from the AON SRAM and help in the calculation of disc rotation rate. The rotation count and rates are saved in AON SRAM, which is always retained even in deep-power-down (2) mode. The one second RTC alarm interrupt is used to calculate rate of disc rotation.

The CM0+ software runs in active mode until the main domain application software is active. After this, the MCU enters in deep-power-down (2) mode. The QTMR0.CH1 periodic one-shot timer keeps CM0+ waking up to deep-power-down (1) mode and then goes back to deep-power-down (2) mode.

The software flow diagram of CM0 + CPU is shown in Figure 36.

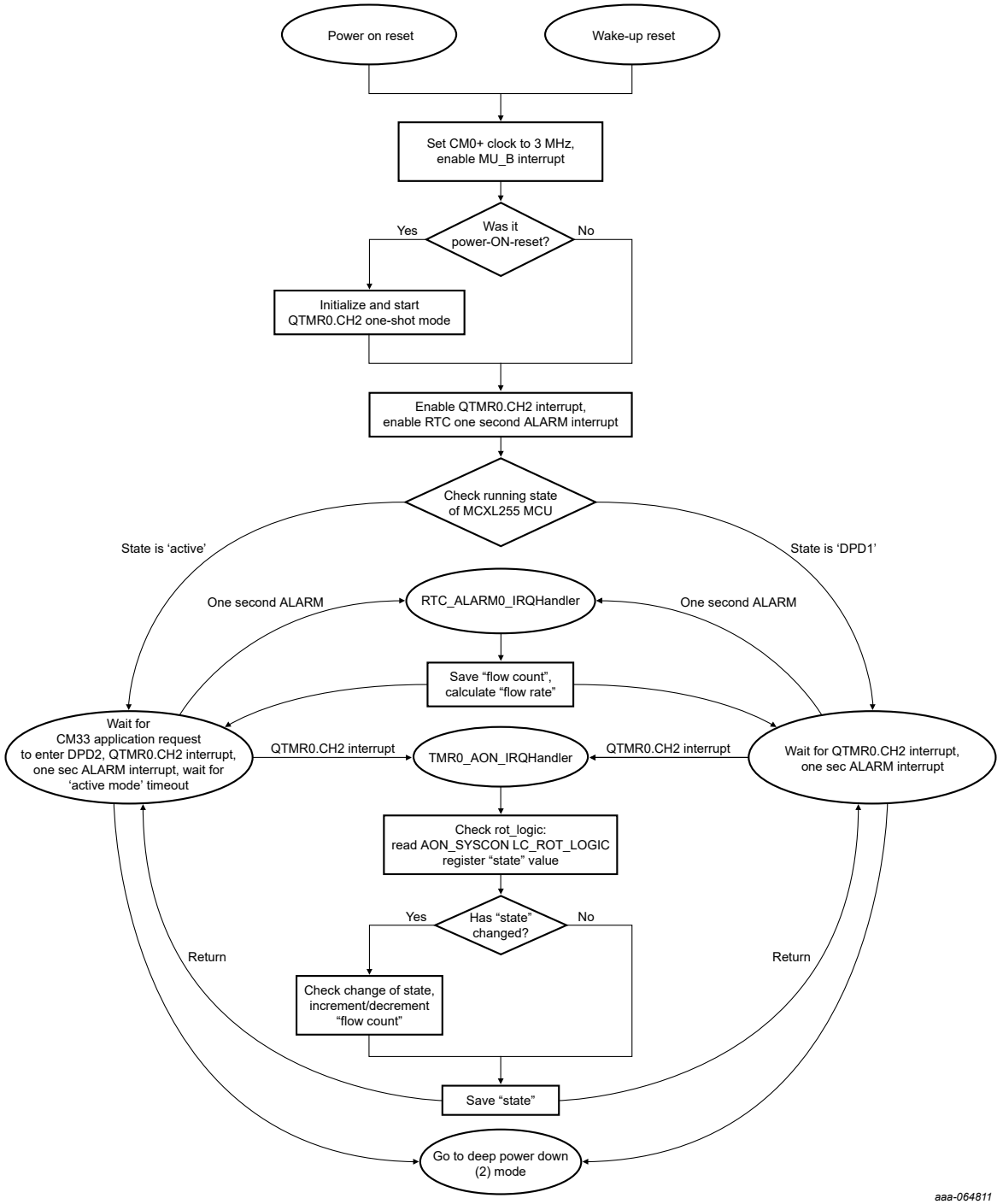


Figure 36. Three-LC CM0+ AON software flow diagram

5.2.2.3 Flow measurement

The demo software running in CM0+ CPU is responsible for (1) calculating flow count and (2) calculating flow count rate, apart from initialization, interrupt handling and deep-power-down mode switching operations.

Disc rotation count is stored continuously in the counter of QTMR1.CH1[CNTR] register and/or AON SRAM. For one rotation of the disc, the QTMR1.CH1[CNTR] counter increments or decrements by 4 counts in QDC mode. Therefore, to calculate disc rotation count, the channel counter value is divided by four.

Cumulative disc rotation is counted using the following formula:

$$\text{Cumulative disc rotation} = \frac{\text{QTMR1.CH1[CNTR]}}{4} \text{counts}$$

For the three-LC method with CM0+ deep-power-down (1) mode software, states are checked to update disc rotation counts stored in AON SRAM.

An averaging method can be used to calculate disc rotation rate. For example, after every one second RTC alarm wake-up, the disc rotation count value can be stored in AON SRAM memory, which is retained in low-power retention mode. At subsequent wakeups, the new count value can be subtracted from the old count value to calculate the counts during the last one second. The averaging method helps to reduce any random fluctuation of flow rate calculation.

$$\text{Rate of disc rotation} = \frac{\text{Disc counts in last N sec}}{N} \times \frac{3600 \text{ counts}}{\text{hour}}$$

5.3 Adaptive FlexLC Sampling

Adaptive LC sense mechanism attempts to readjust the sample rate to reduce average sense power consumption. As the sample rate depends on the maximum disc rotation rate, the sample rate can be reduced or increased to support slow- or fast-moving half-metal discs. If the sample rate can be reduced, it can save activity time of LC sensing and reduce power consumption.

We know we target a maximum of 50 times the disc rotation per second (rps), so the sample rate should be 50 rps x 2 (discs) x 2 (nyquist) = 200 Hz. Similarly, with a low disc rotation rate of, for example, 1 rps, only a 4 Hz sample rate suffices. The challenge is to realize the change in rotation speed and the ability to update the sample rate dynamically. This update should happen without disturbing the ongoing disc rotation calculations and therefore, without altering any forbidden configuration of the LC sense driver and sense circuits.

Usually, the rotation rate does not change frequently, and a mechanism implements a periodic check of the disc rotation rate; and if underrated, would adjust the sampling frequency only by adjusting the pretrigger frequency from default 200 Hz to lower. Similarly, in case of an increase in rotation rate, the pretrigger sample rate can be increased back to higher values.

Other timers than LPTMR can also be used to facilitate a race-free and safe way to update the sample rate with pretriggering. LPTMR does not support an alternative/shadow register to change the compare value dynamically. LPTMR has only one CMR register to support the static configuration of the compare value. QTMR.CHn supports dynamic reload of the compare value support with register CSCTRLn[CL1], which can be used to load new compare values to CMPLD1 register, according to LC sampling rates.

6 Calibration

Calibration of the LC sensor includes the method to adjust the LC sensor operational circuit parameters for reliable and accurate operation, that is, to be able to count the disc rotation accurately. LC sense operation depends on several parameters including the accuracy and quality of production LC sense circuit components, mounting distance of the half-metal disc with alignment, operating temperature, voltage, humidity.

LC sense calibration is not the same as the calibration of a flow meter, which must ensure the accuracy of flow measurement with respect to another accurate reference flow meter.

6.1 Initial calibration

A one-time calibration can be performed under ambient temperature conditions using new LC components. This can be done during production using a factory calibration tool/mechanism.

As described in Section 4.8 and Section 4.9, QTMR1 channels are used to drive the LC sense circuit, while QTMR0 channels are used to sense the pulses generated by damped or nondamped oscillations.

During calibration, the compare interrupt for the LPTMR0 associated with the QTMR1.CH1 timer is enabled. The Nested Vector Interrupt Controller (NVIC) of the ARM CM33 core complex is configured to receive and service both the LPTMR and QTMR1.CH1 interrupts.

Note: *In MCX L255 MCU, both the main domain and AON peripheral interrupts are connected to the NVIC controller of the ARM CM33 core complex. The interrupt sources from AON peripherals are also connected to ARM CM0+ NVIC. The interrupt sources from main domain peripherals are not connected to ARM CM0+ NVIC. This gives much flexibility for sensor application development from the main domain as well.*

This is done without disrupting the operation as these timer channels are used in the ongoing LC sense operation. The interrupt generated by the LPTMR compare event is used to synchronize the calibration operation to a known starting point. The compare interrupt service routine (ISR) is used to start compare interrupt of QTMR1.CH1. The compare ISR of QTMR1.CH1 is used to count the oscillation pulse count values captured in the QTMR0.CH0 counter. After counting an LC channel pulses, the count value from QTMR0.CH0 must be read before this timer channel is reset by the QTMR1.CH2 rising edge, spaced apart far enough so that ISR can read the pulse counts for the current LC channels easily.

The LC Sense calibration software is placed in the `lc_cal_manager.c/.h` source files. The APIs that are used for calibration are:

`Start_Calib()`, `Read_CalibData()`, and `Get_CalibState()`.

The **`Start_Calib()`** function is used to start the calibration process. This is done after receiving a command from a PC host tool. After the calibration is done, it is stopped automatically.

The `Read_CalibData()` function returns the required set parameters after the successful calibration. It can return the maximum and minimum oscillation counts, the timer compare threshold value is set after a successful configuration.

`Get_CalibState()` can return one of the possible states of the calibration process:

- CALIB_IDLE = Calibration process is not active
- CALIB_PROGRESS = Calibration process is active
- CALIB_DONE = Calibration process is done successfully
- CALIB_FAIL = Calibration process is unsuccessful

The postproduction one-time calibration process is started after the LC sense software is preprogrammed and the flow meter is connected to a PC host tool used to assist calibration. NXP has developed a demo Microsoft Windows PC host GUI tool called “LC-calib Monitor”. The tool communicates to the FRDM-MCXL255 board with a serial COM port on either side. So, the USB cable, used to power FRDM-MCXL255, can be connected to a PC to power the FRDM board as well to get the serial port enumerated with the Microsoft Windows PC COM serial port.

1. Double-click `LCCalibMonitor.exe` and launch the tool main window.

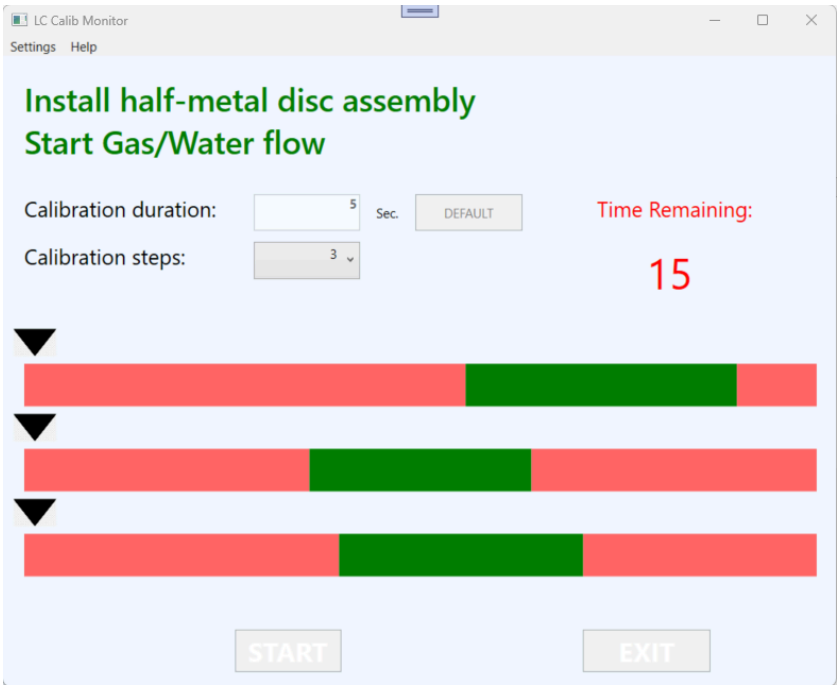


Figure 37. LC-calib Monitor tool launch view

The main window prompts the user to make sure the half-metal disc assembly is properly installed. The distance of LC circuits to the disc, its height, and alignment is critical for successful calibration. It usually needs a few trials to be able to set up the disc rotor motor setup shown in this demo application. The second condition is that the disc must be rotated at a minimum speed of 1 rotation per second.

1. The connection of the tool to the FRDM-MCXL255 USB serial port can be opened by accessing “Setting->Serial Port” as shown in Figure 38. Select the right COM port cautiously and click the **OK** button.

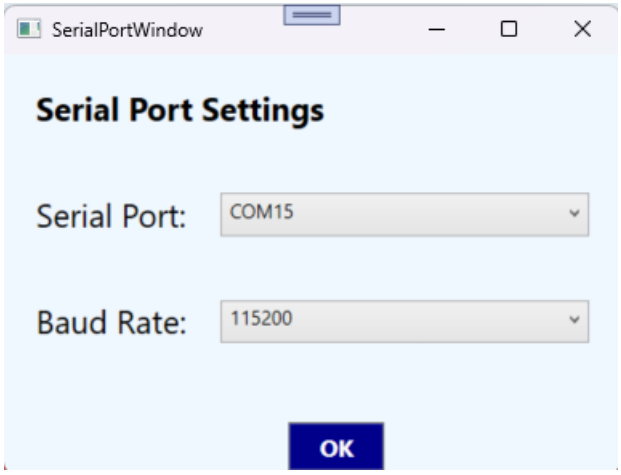


Figure 38. Settings for serial port

2. Select calibration steps from the drop-down combo-box in the main window, which allows 1, 2 or 3 steps only. Each step duration is set by entering the value in the “Calibration duration” textbox in the main window. The total duration of calibration is calculated by (step duration x steps). With default set values of duration (5 seconds) and steps (3), the total calibration duration is calculated to be 15 seconds.
3. Before starting the calibration process, the default settings of the calibration parameters can be viewed by opening the “Setting->LC Parameters”.

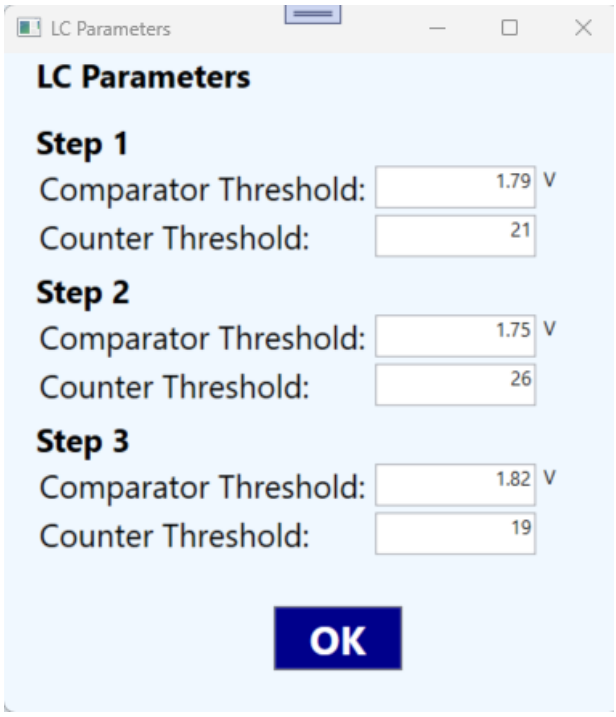


Figure 39. Settings for LC sensor configuration parameters

The parameters are good for the calibration of LC circuits in FLO-SNSR-BRD. End user application can set the parameters different from the default values. The stepwise setting parameters include a pair of (1) comparator threshold and (2) counter threshold values. The comparator threshold is the approximate value of the negative input to generate by the internal DAC of AON.ACMP. The counter threshold is used to set the pulse counter threshold value to determine counts are indicative to air or metal part of the disc.
To apply the settings, click the **OK** button.

4. Hit the START button from the main window. It starts the calibration process in steps. Every second, the “Time Remaining” is counted down, two live arrow pointers show the current average values of maximum and minimum counts for calibration duration per step. After the step duration, a counter threshold pointer is positioned at the average value of calculated maximum and minimum pulse count values. The steps repeat for a selected number of steps.

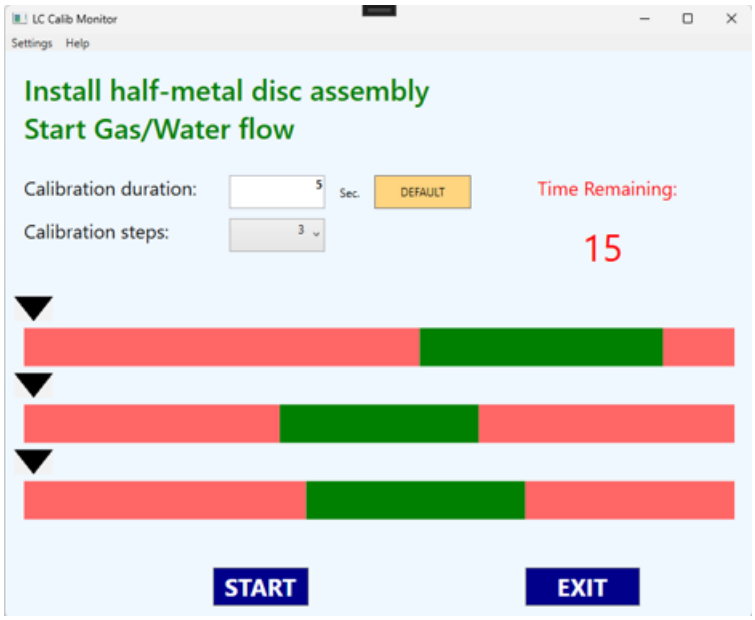


Figure 40. Calibration process started

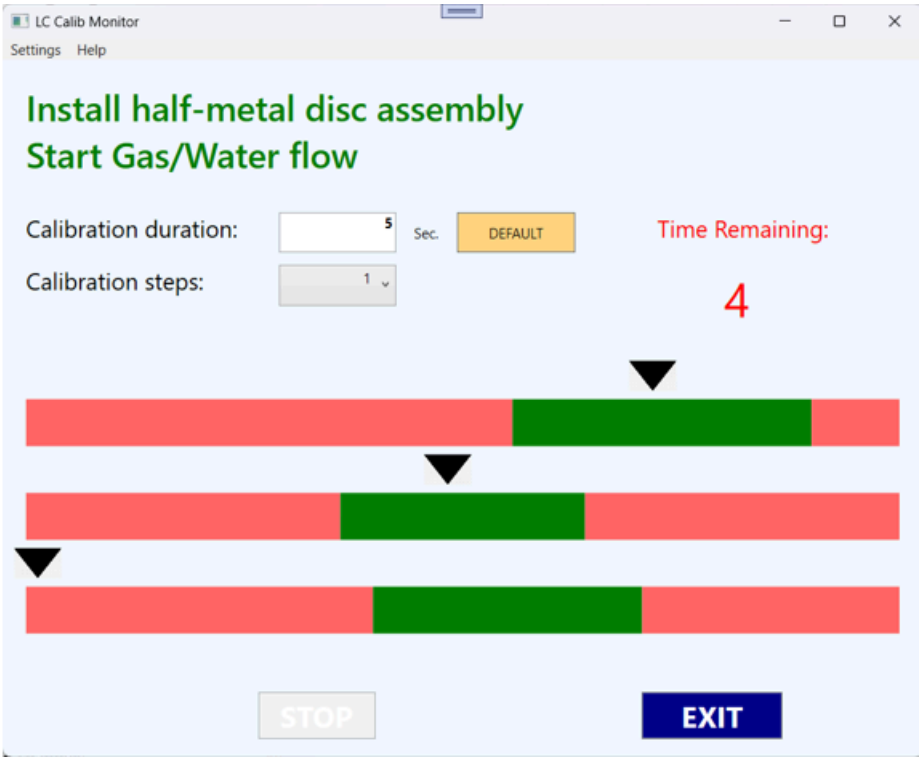


Figure 41. Ongoing calibration process

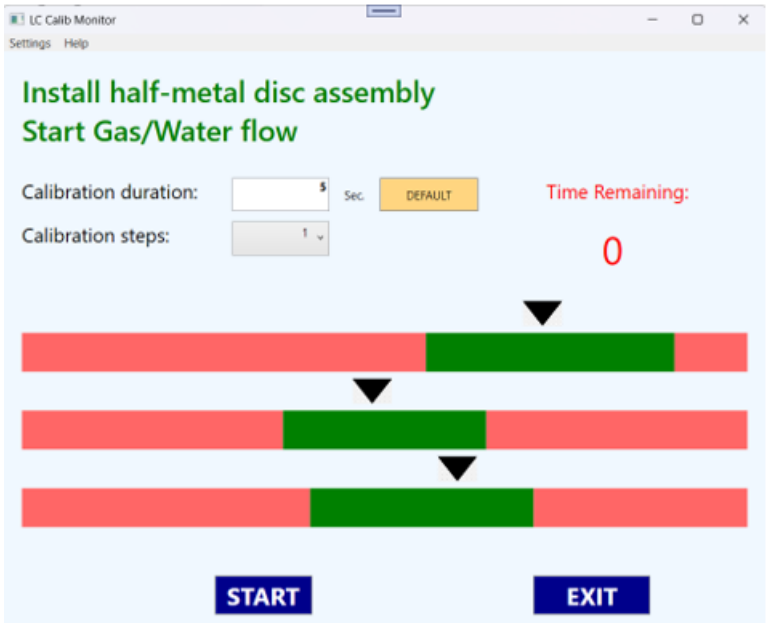


Figure 42. Calibration process done

5. After the elapsed calibration duration, the result is shown prompt “Success” or “Fail”. The best possible step value is automatically calculated in the MCX L255 calibration manager software and applied.
   A reinitialization of LC sense sensors is done to start from “zero” disc rotation counts, but with the configuration of FlexLC logic done with the calibrated (1) comparator threshold and (2) counter threshold values for the best possible calibrated operation.
   Note: *In case the installation of FLO-SNSR-BRD in incorrect, or due to the misalignment of disc, or stationary disc will result in the calibration process to fail. After fixing these issues, the calibration process can be restarted.*

6.2 Software implementation

Calibration software has been placed in part of the main domain CM33 software. It is not an active process for LC sense operation and therefore is not operational unless required. A trigger to this service starts calibration. Once the action is taken, the results are stored and applied to the LC software configuration before restarting the sensor.

Figure 43 shows the software state machine of one-time calibration process.

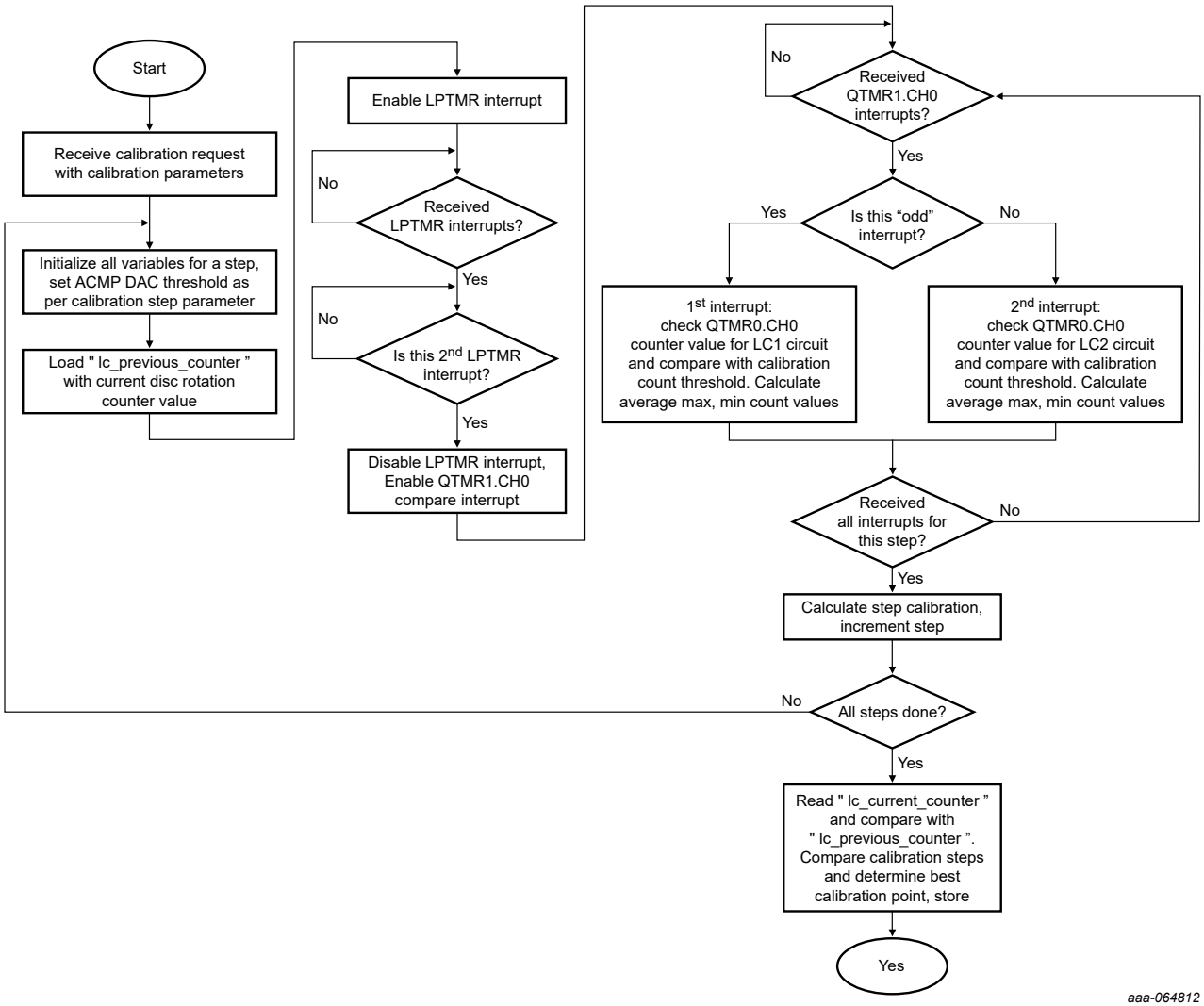


Figure 43. LC sensor static calibration flow diagram

6.3 Periodic calibration

While one-time calibration can be done during the product development/shipping stage, the operating condition and aging factor might cause performance degradation of LC sensor. LC sense circuit operation can deviate from initial calibrated condition due to change in operating temperature, voltage, humidity in the field, which might cause little deviation of sensor circuit performance, causing miscalculation of disc rotation count or permanent nonoperation.

Therefore, a periodic check and required recalibration method can be employed to tune the circuit operation over time, even after the flow meters using LC sense circuit is installed in the field. Usually this check can be done by running a method in flow meter software automatically. The calibration method described in Section 6.1 can be executed once a day, or week, or month to check if any the current calibration setting is optimal or deviating from working condition. Based on this, tuning the AON.ACMP DAC threshold and/or QTMR0.CH0 compare threshold values can be dynamically updated. Both of these parameters can be updated safely as supported by AON.ACMP and QTMR blocks. However, precaution must be taken not to overcompensate, which can impact the ongoing disc rotation count process, causing malfunction.

7 Revision history

Table 6 summarizes the revisions to this document

Table 6. Revision history

Document ID	Release date	Description
AN14934 v.2.0	13 July 2026	Initial public release
AN14934 v.1.0	27 April 2026	Initial NDA release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents