RX Family

Capacitive Liquid Level Indicator Demo System Sample Software

Introduction

This document describes the Capacitive Liquid Level Indicator Demo System sample software as an application example of the self-capacitance method used in Capacitive Touch Sensor Unit (CTSU) hardware. CTSU detects human touch by measuring the electrostatic capacitance generated between the touch electrode and the human body.

Target Device

RX130 Group

Related Documents

1. RX Family Using QE and FIT to Develop Capacitive Touch Applications (R01AN4516)
2. RX Family Firmware Integration Technology User's Manual (R01AN1833)
3. RX Family Board Support Package Module using Firmware Integration Technology (R01AN1685)
4. RX Family QE Touch Module Firmware Integration Technology (R01AN4470)
5. RX Family QE CTSU Module Using Firmware Integration Technology (R01AN4469)
6. RX Family CTSU Improved Accuracy Sample Software (R11AN0314)
7. RX130 Group Capacitive Liquid Level Indicator Demo System (R12AN0108)
8. RX Family Level Monitor User’s Manual (R11UZ0027)

Contents

1. Overview

This application note describes the sample software that operates the Capacitive Liquid Level Indicator Demo System based on the Capacitive Level Sensor. Please refer to Section 6.4 Sample Application Functions for more details on how to operate the Demo System.

1.1 Software Structure

Figure 1.1 shows the software structure diagram.

Capacitive measurement with the CTSU employs software generated by QE for Capacitive Touch, a development support tool for capacitive touch sensor applications, and Smart Configurator. The software is referred to as QE Touch Module and QE CTSU Module, respectively.

The CTSU Water Level Detection API detects the level of water from the CTSU measurement results of the QE Touch module and QE CTSU module.

The application informs the water level to the user via LEDs on the Capacitive Liquid Level Indicator Demo System and USB communications with the user PC.

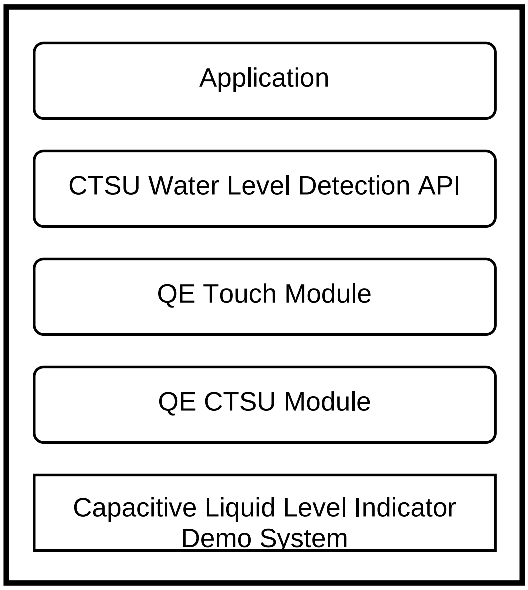

Figure 1.1 Software Structure Diagram

1.2 File Configuration

Table 1.1 shows the file configuration used in this sample software. Source files and header files generated by the Smart Configurator, such as the QE Touch module, have been omitted for brevity.

Table 1.1 File Configuration

Folder/File Name	Description
liquid_level_sample_project	Project folder
.cproject	C project file
.project	Project file
liquid_level_sample_project HardwareDebug.launch	Debug configuration file
liquid_level_sample_project.scfg	Smart configurator configuration file
src	Source/header file storage folder
qe_common.c	QE Touch common source file
qe_common.h	QE Touch common header file
qe_config01.c	QE Touch configuration definition source file
qe_config01.h	QE Touch configuration definition header file
r_caplevel_command.c	Capacitive Liquid Level Indicator Demo System communication command control source file
r_caplevel_command.h	Capacitive Liquid Level Indicator Demo System communication command control header file
r_caplevel_config.c	Capacitive Liquid Level Indicator Demo System water level detection configuration source file
r_caplevel_config.h	Capacitive Liquid Level Indicator Demo System water level detection configuration header file
r_caplevel_dataflash.c	Capacitive Liquid Level Indicator Demo System data flash control source file
r_caplevel_measure.c	Capacitive Liquid Level Indicator Demo System water level detection API source file
r_caplevel_measure.h	Capacitive Liquid Level Indicator Demo System water level detection API header file
r_liquid_level_sample_project.c	Application file
QE-Touch	QE for Capacitive Touch generated folder
liquid_level_sample_project.tifcfg	Touch I/F configuration file

2. Operating Environment

Table 2.1 shows the operating environment of the sample software.

Table 2.1 Operating Environment

Item	Description
Capacitive Liquid Level Indicator Demo System	Capacitive Liquid Level Indicator Demo System for RX130 (RTK5RX1300D02001BJ)
Capacitive Liquid Level Indicator Demo System Control Board	Capacitive Liquid Level Indicator Demo Control Board (RTK0EG0018B00001BJ)
Capacitive Liquid Level Indicator Demo System Ruler Board	Capacitive Liquid Level Indicator Demo Ruler Board (RTK0EG0019B00001BJ)
Capacitive Liquid Level Indicator Demo System Electrode Board	Capacitive Liquid Level Indicator Demo Electrode Board (RTK0EG001AB00001BJ)
MCU	RX130
Operating frequency	32 MHz
Operating voltage	5.0 V
Integrated Development Environment	e^2 studio V7.8.0
C compiler	CC-RX V3.01.00
CTSU development support tool	QE for Capacitive Touch V1.1.0

Figure 2.1 shows the device connection diagram.

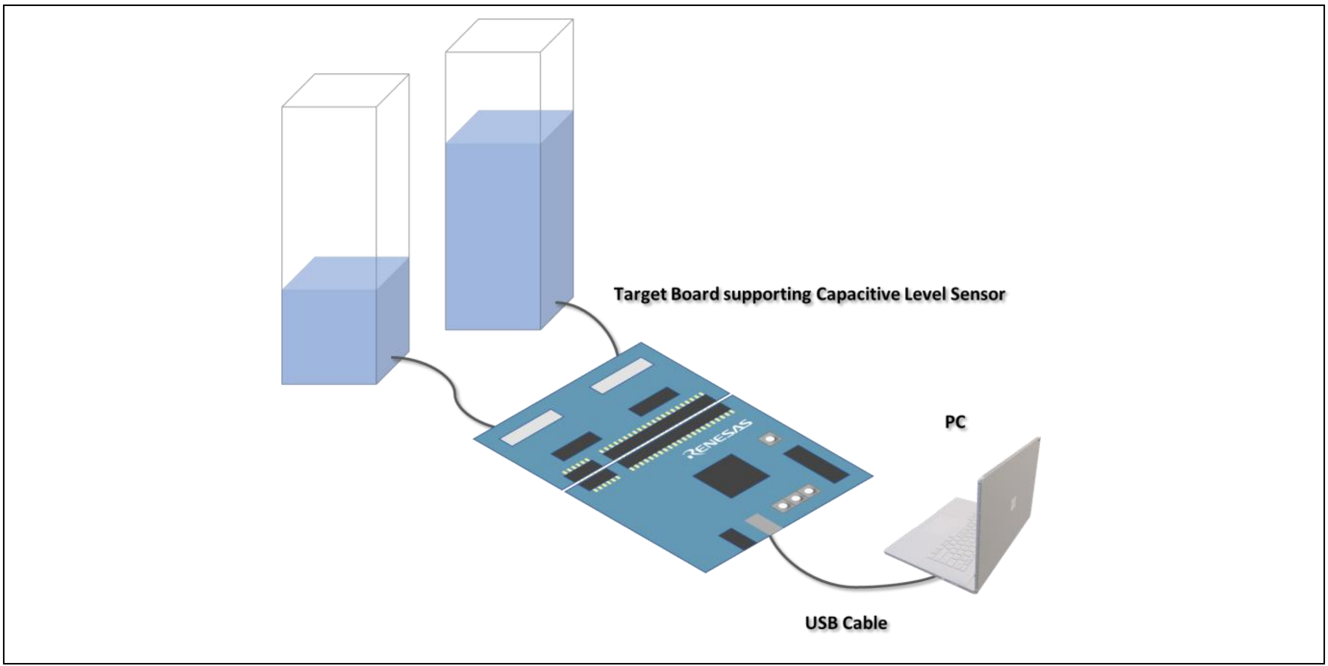

Figure 2.1 Device Connection Diagram

3. QE Touch and QE CTSU Settings

The QE Touch and QE CTSU modules are automatically generated by the QE for Capacitive Touch and the Smart Configurator, respectively. This chapter explains the modifications required in these modules for operating the Capacitive Liquid Level Indicator Demo System.

The software used to operate the Capacitive Liquid Level Indicator Demo System supports the updated ICO correction function, referred to as the “7-point correction function.” In order to support the 7-point correction function, we have made modifications to the conventional correction function process implemented in the QE CTSU and QE Touch modules, enabling selection of either the conventional or 7-point correction function as the ICO correction function.

Modifications made to r_ctsu_qe_if.h to enable the 7-point correction function:

```
#define  ENABLE_NEW_CORRECTION
#define  ENABLE_RUNNING_CALIBRATION
#define  ENABLE_VOLTAGE_CORRECTION
```

Modifications made to r_ctsu_qe_if.h to disable initial calibration (offset tuning).

```
#undef   ENABLE_INITIAL_OFFSET_TUNING
```

For more details regarding 7-point correction, please refer to “CTSU Improved Accuracy Sample Software” (R11AN0314).

4. Water Level Detection

4.1 Water Level Detection Principle

As shown in Figure 4.1, the Capacitive Liquid Level Indicator Demo System employs a double-layered water tank to detect the water level. The touch electrode is attached to the surface of the inner tank. The water level is calculated from the detected count value while fluctuations in the count value due to water adhesion are suppressed. As the electrode strip is large, employing a shield is recommended to reduce any influence of human touch on the sensitivity of the electrodes. The shield should be attached to the back of the outer tank to cover the area of the touch electrodes. Doing so will suppress fluctuation due to parasitic capacitance of the touch electrodes caused by a person approaching the tank, enabling stable sensitivity and an accurate detection of the water level.

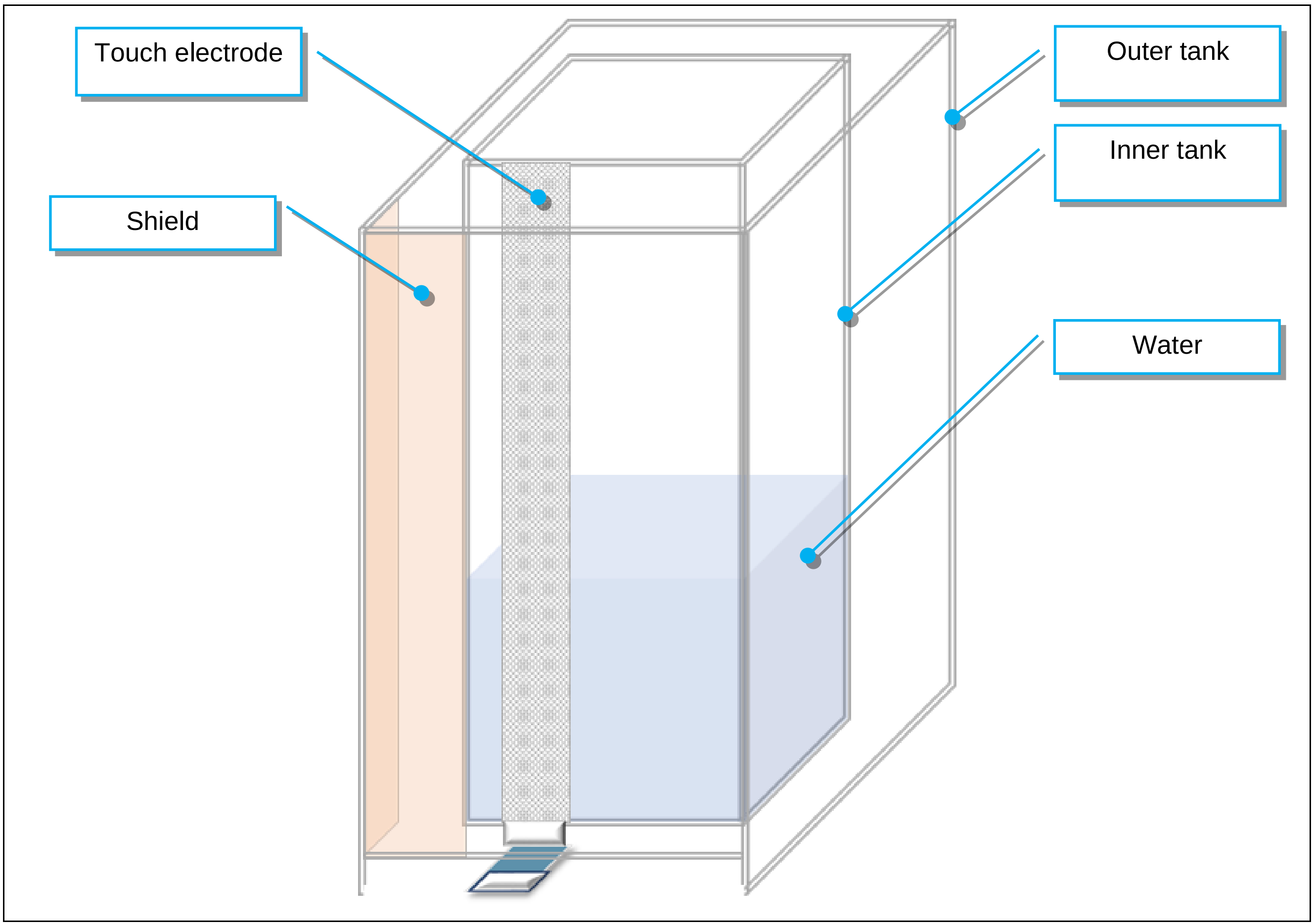

Figure 4.1 Water Level Detection Principle

4.2 Water Level Calculation Method

The count value of the touch electrode attached to the water tank surface is used to detect the level of the water.

The count values are measured in advance when the tank is empty (water level = 0%) and when it is full (water level = 100%). These values are defined as the lower threshold and upper threshold, respectively. The water level is calculated from the count value obtained by CTSU measurement using the lower and upper thresholds based on the following formulas.

Water level = ((count value - lower threshold) * 100) / (upper threshold - lower threshold)

The water level is calculated from the water level [%] using the following formula.

Water level = water level [%] * max. water level [%] / 100

The water volume is calculated from the water level [%] using the following formula.

Water volume = water level [%] * max. water value / 100

4.3 Preset Calibration

When the QE Touch module is started, it executes standard CTSU measurement correction processing and offset adjustment (referred to as “initial calibration” herein). An issue that comes up when designing the CTSU Water Level Detection API into a product, is that there must be a certain amount of water in the tank to be measured at start up. As a resolution, we provide a mechanism that converts the results of the initial calibration into data that is incorporated into the software. This mechanism is called “preset calibration.” For this preset calibration to work, the initial calibration must be disabled. Refer to section 3. QE Touch and QE CTSU Settings for instructions on how to disable the initial calibration.

5. Water Level Detection Software Specification

5.1 File Configuration

Table 5.1 shows the source files used in this sample software.

Table 5.1 Source Files

File Name	Description	Notes
r_caplevel_measure.c	CTSU Water Detection API	-
r_caplevel_config.c	CTSU Water Level Detection Configuration	-
r_caplevel_command.c	CTSU Water Level Detection Communication Command Control	-
r_caplevel_dataflash.c	Data Flash Control	-

Table 5.2 shows the header files.

Table 5.2 Header Files

File Name	Description	Notes
r_caplevel_measure.h	CTSU Water Detection API	-
r_caplevel_config.h	CTSU Water Level Detection Configuration	-
r_caplevel_command.h	CTSU Water Level Detection Communication Command Control	-

5.2 Constants

Table 5.3 shows the r_caplevel_measure.h constants.

Table 5.3 Constants (r_caplevel_measure.h)

Constant Name	Setting Value	Description
CAPLEVEL_SUCCESS	(0)	CTSU Water Detection API successful completion
CAPLEVEL_ERROR	(1)	CTSU Water Detection API abnormal termination

Table 5.4 shows the r_caplevel_config.h constants.

The default setting value is the value indicated by the Capacitive Liquid Level Indicator Demo System; please modify as necessary to fit the usage environment.

Table 5.4 Constants (r_caplevel_config.h)

Constant Name	Setting Value	Description
CAPLEVEL_METHOD_CH0	(0)	Channel 0 – method number
CAPLEVEL_METHOD_CH1	(0)	Channel 1 – method number
CAPLEVEL_METHOD_CH2	(0xff)	Channel 2 – method number
CAPLEVEL_METHOD_CH3	(0xff)	Channel 3 – method number
CAPLEVEL_SENSOR_CH0	(17)	Channel 0 – TS number
CAPLEVEL_SENSOR_CH1	(1)	Channel 1 – TS number
CAPLEVEL_SENSOR_CH2	(0xff)	Channel 2 – TS number
CAPLEVEL_SENSOR_CH3	(0xff)	Channel 3 – TS number
CHANNEL0_LOWER_THRESHOLD	(15360)	Channel 0 – lower threshold
CHANNEL0_UPPER_THRESHOLD	(24800)	Channel 0 – upper threshold
CHANNEL0_MAX_HEIGHT	(175)	Channel 0 – tank height
CHANNEL0_MAX_VOLUME	(438)	Channel 0 – tank volume
CHANNEL1_LOWER_THRESHOLD	(15360)	Channel 1 – lower threshold
CHANNEL1_UPPER_THRESHOLD	(25800)	Channel 1 – upper threshold
CHANNEL1_MAX_HEIGHT	(175)	Channel 1 – tank height
CHANNEL1_MAX_VOLUME	(438)	Channel 1 – tank volume
LED1_LEVEL_RIGHT	(11)	Water level when bottom LED ruler board (R) is on
LED2_LEVEL_RIGHT	(40)	Water level when lower-middle LED ruler board (R) is on
LED3_LEVEL_RIGHT	(69)	Water level when upper-middle LED ruler board (R) is on
LED4_LEVEL_RIGHT	(94)	Water level when top LED ruler board (R) is on
LED1_LEVEL_LEFT	(11)	Water level when bottom LED ruler board (L) is on
LED2_LEVEL_LEFT	(40)	Water level when lower-middle LED ruler board (L) is on
LED3_LEVEL_LEFT	(69)	Water level when upper-middle LED ruler board (L) is on
LED4_LEVEL_LEFT	(94)	Water level when top LED ruler board (L) is on
AUTO_CHANGE_RIGHT	(100)	Water level at completion of auto injection to tank (R)
AUTO_CHANGE_LEFT	(100)	Water level at completion of auto injection to tank (L)
CAPLEVEL_MAX_CHANNEL	(4)	Maximum number of channels

Table 5.5 shows the r_caplevel_command.c constants.

Table 5.5 Constants (r_caplevel_command.c)

Constant Name	Setting Value	Description
CAPLVLCMD_CMD_START	(0x55)	Communication command start ID
CAPLVLCMD_CMD_STOP	(0x0A)	Communication command stop ID
CAPLVLCMD_CMD_MONITOR_START	(0x01)	Communication command ID (monitor start request)
CAPLVLCMD_CMD_MONITOR_STOP	(0x02)	Communication command ID (monitor stop request)
CAPLVLCMD_CMD_MONITOR_DATA	(0x03)	Communication command ID (send measured value)
CAPLVLCMD_CMD_READ_PARM	(0x04)	Communication command ID (read parameters request)
CAPLVLCMD_CMD_WRITE_PARAM	(0x05)	Communication command ID (write parameters request)
CAPLVLCMD_CMD_INIT_PARAM	(0x06)	Communication command ID (parameter initialization request)
CAPLVLCMD_CMD_UTILITY	(0x07)	Communication command ID (data flash update, offset adjustment execution request)
CAPLVLCMD_CMD_REPLY	(0x80)	Communication command reply ID
CAPLVLCMD_CMD_ERROR	(0x40)	Communication command error ID
CAPLVLCMD_CMD_MAX_CHANNEL	(4)	Maximum number of channels

5.3 Structures

Table 5.6 shows the captouch_level_sensor_config_t structures.

Table 5.6 captouch_level_sensor_config_t Structures

Type	Member	Description
uint8_t	method_number	Method number
uint8_t	sensor_number	TS number
uint16_t	lower_threshold	Lower threshold
uint16_t	upper_threshold	Upper threshold
uint16_t	max_height	Tank height
uint16_t	max_volume	Tank volume
uint16_t	count_value	Count value
uint16_t	raw_value	CTSU sensor counter
uint16_t	sensor_offset0	CTSU sensor offset register 0 setting value
uint16_t	sensor_offset1	CTSU sensor offset register 1 setting value
uint16_t	ctsu_clock	CTSU measurement frequency
uint16_t	ctsu_status	CTSU status

Table 5.7 shows the captouch_level_config_t structures.

Table 5.7 captouch_level_config_t Structures

Type	Member	Description
captouch_level_sensor_config_t	sensor_config[CAPLEVEL_MAX_CHANNEL]	CTSU sensor configuration
uint8_t	max_channel	Number of enabled channels
uint16_t	filter_value	Moving average filter value

5.4 API Functions

5.4.1 R_CAPLEVEL_Open

Initialization API

Format

void R_CAPLEVEL_Open (captouch_level_config_t * p_config, uint8_t max_channel)

Argument

p_config	Pointer to water level detection structure
max_channel	Number of enabled channels

Return value

None

Description

This API function initializes the CTSU Water Detection API.

Notes

Call this function only once after system startup.

5.4.2 R_CAPLEVEL_GetLevel

Get water level [%] API

Format

uint8_t R_CAPLEVEL_GetLevel (uint8_t channel, uint16_t * p_level)

Argument

channel	Channel number
p_level	Pointer to water level [%] storage variable

Return value

CAPLEVEL_SUCCESS	Successful completion
CAPLEVEL_ERROR	Abnormal termination

Description

This API stores the water level [%] held by the CTSU Water Detection API in argument *p_level.

When the argument is NULL, the function returns CAPLEVEL_ERROR.

5.4.3 R_CAPLEVEL_GetHeight

Get Water Level Height API

Format

uint8_t R_CAPLEVEL_GetHeight (uint8_t channel, uint16_t * p_height)

Argument

channel	Channel number
p_height	Pointer to water level storage variable

Return value

CAPLEVEL_SUCCESS	Successful completion
CAPLEVEL_ERROR	Abnormal termination

Description

This API function stores the water level held by the CTSU Water Detection API in argument *p_height.

When the argument is NULL, the function returns CAPLEVEL_ERROR.

5.4.4 R_CAPLEVEL_GetVolume

Get Water Volume API

Format

uint8_t R_CAPLEVEL_GetVolume (uint8_t channel, uint16_t * p_volume)

Argument

channel	Channel number
p_volume	Pointer to water volume storage variable

Return value

CAPLEVEL_SUCCESS	Successful completion
CAPLEVEL_ERROR	Abnormal termination

Description

This API function stores the water volume held in the CTSU Water Detection API in the argument *p_volume.

When the argument is NULL, the function returns CAPLEVEL_ERROR.

5.4.5 R_CAPLEVEL_Config_Init

Water Level Detection Configuration Initialization API

Format

void R_CAPLEVEL_Config_Init (void)

Argument

None

Return value

None

Description

This API function initializes the water level detection configuration.

Notes

Call this function only once after system startup.

5.4.6 R_CAPLVLCMD_Open

Communication Command Control Initialization API

Format

void R_CAPLVLCMD_Open (captouch_level_config_t * p_config, uint8_t * max_channel)

Argument

p_config	Pointer to water level detection configuration
max_channel	Number of enabled channels

Return value

None

Description

This API function initializes the communication command control.

Notes

Call this function only once after system startup.

5.4.7 R_CAPLVLCMD_Send

Monitor Data Send API

Format

void R_CAPLVLCMD_Send (void)

Argument

None

Return value

None

Description

When the level monitor is enabled, this API function sends each measurement value detected, including water level [%].

5.4.8 R_CAPLVLCMD_Receive

Communication Command Receive Control API

Format

void R_CAPLVLCMD_Receive (void)

Argument

None

Return value

None

Description

This API function analyzes the communication command received from the level monitor and carries out the corresponding process or response.

6. Sample Applications

The following sample application using the Capacitive Liquid Level Indicator Demo System has been added to this sample software, based on the QE Touch module application file r_liquid_level_sample_project.c.

6.1 Capacitive Liquid Level Indicator Demo System Initialization

The following initialization functions have been added before the main function loop

- R_CAPLEVEL_Config_Init() - Water level detection configuration initialization
- SampleInitialize() - CTSU measurement initialization
- R_CAPLEVEL_Open() - Water level detection initialization
- R_CAPLVLCMD_Open() - Communication command control initialization

6.2 Results Notification

To notify the user of water level detection measurement results, function R_CAPLVLCMD_Send () has been added after R_TOUCH_UpdateDataAndStartScan(), which is called in the main function loop.

This ensures that R_CAPLVLCMD_Send () is called for each touch measurement cycle.

Notification command receive control has also been added to the main function loop.

6.3 Switch Detection/LED Control

SampleApplication(), the function that detects when the Demo System switch has been pressed and controls the functions related to the switch, has also been added to the main function loop.

SampleApplication() also controls the LEDs that indicate the function status of each switch.

6.4 Sample Application Functions

This section describes the Demo System functions realized with the sample application.

6.4.1 Switches

The following are the switch functions used in the sample application. Pressing the switches transitions the mode from normal to auto and offset adjustment. Mode transitions are described below.

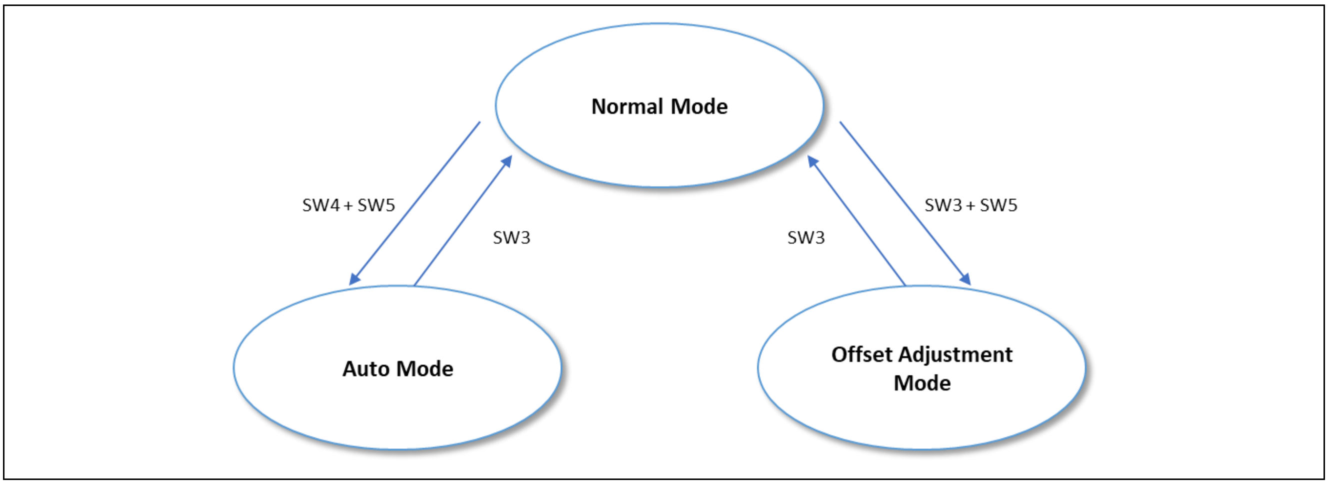

Figure 6.1 Mode Transition

Table 6.1 describes the switch functions in normal mode.

Table 6.1 Switch Functions (normal mode)

Switch	Function	Notes
SW3	Stops water injection.	
SW4	Starts water injection to left-side tank.	
SW5	Starts water injection to right-side tank.	
SW4 + SW5	Transitions to auto mode.	Injects water into both tanks repeatedly, alternating from left to right or right to left.
SW3 + SW5	Transitions to offset adjustment mode.	Adjusts the amount of current offset for TS1 and TS17.
SW3 + SW4 + SW5	Executes auto offset adjustment.	Auto offset adjustment results are stored in the data flash.

Table 6.2 describes the switch functions in auto mode.

Table 6.2 Switch Functions (auto mode)

Switch	Function	Notes
SW3	Stops auto mode.	

Table 6.3 describes the switch functions in offset mode.

Table 6.3 Switch Functions (offset mode)

Switch	Function	Notes
SW3	Changes adjustment target to TS17. Stops offset adjustment mode.	TS1 is the adjustment target when the mode is transitioned to offset adjustment. Pressing SW3 changes the target to TS17, pressing SW3 once more stops the offset adjustment mode.
SW4	Current offset amount +1	
SW5	Current offset amount -1	

6.4.2 LED Displays

Table 6.4 describes the meanings of the LED displays.

Table 6.4 LED Display

LED	Function	Notes
Electrode board (L)	Blinks when water is being injected into the left tank.	
Electrode board (R)	Blinks when water is being injected into the right tank.	
Ruler board (bottom)	Turns on when water level in tank reaches 20mm.	Same function for right and left tanks.
Ruler board (lower-middle)	Turns on when water level in tank reaches 70mm.	Same function for right and left tanks.
Ruler board (upper-middle)	Turns on when water level in tank reaches 120mm.	Same function for right and left tanks.
Ruler board (top)	Turns on when water level in tank reaches 165mm.	Same function for right and left tanks.

6.4.3 Level Monitor and USB Communication

Communication command control function R_CAPLVLCMD_Receive() has been added to the main function of the sample application.

The monitoring tool, Level Monitor, transmits information using the formats shown below. For more details on Level Monitor, see *Level Monitor User's Manual (R11UZ0027)*.

6.4.3.1 Communication Format for Monitoring

The following communication formats are used when monitoring.

- RX130 (Capacitive Liquid Level Indicator Demo System) --> PC (Level Monitor)

Bytes 0 – 2

Byte	0	1	2
Value	Start ID	Send ID	Command ID

Bytes 3 – 20

Byte	3 4	5 6	7 8	9 10	11 12	13 14	15 16	17 – 20
Value	Level 0	Height 0	Volume 0	Count Value 0	CTSU Sensor Counter 0	CTSU Status 0	Offset 0	Reserved

Bytes 21 – 38

Byte	21 22	23 24	25 26	27 28	29 30	31 32	33 34	35 – 38
Value	Level 1	Height 1	Volume 1	Count Value 1	CTSU Sensor Counter 1	CTSU Status 1	Offset 1	Reserved

Bytes 39 – 75

Byte	39 – 74	75
Value	Reserved	Stop ID

Table 6.5 Communication Format for Monitoring (RX130 -> PC)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00 – 0xff	Value incremented for each communication
Command ID	0x03	Communication command ID
Level 0	0x0000 – 0xffff	Water level [%] detected at TS17
Height 0	0x0000 – 0xffff	Height of water level detected at TS17
Volume 0	0x0000 – 0xffff	Volume of water level detected at TS17
Count Value 0	0x0000 – 0xffff	TS17 count level
CTSU Sensor Counter 0	0x0000 – 0xffff	TS17 CTSU sensor counter
CTSU Status 0	0x0000 – 0xffff	TS17 CTSU status
Offset 0	0x0000 – 0xffff	TS17 CTSUSO0 setting value
Level 1	0x0000 – 0xffff	Water level [%] detected at TS1
Height 1	0x0000 – 0xffff	Height of water level detected at TS1
Volume 1	0x0000 – 0xffff	Volume of water level detected at TS1
Count Value 1	0x0000 – 0xffff	TS1 count value
CTSU Sensor Counter 1	0x0000 – 0xffff	TS1 CTSU sensor count value
CTSU Status 1	0x0000 – 0xffff	TS1 CTSU status
Offset 1	0x0000 – 0xffff	TS1 CTSUSO0 setting value
Reserved	0x00	Reserved area
Stop ID	0x0a	Stop communication data

- PC (Level Monitor) -> RX130 (Capacitive Liquid Level Indicator Demo System)

Bytes 0 – 3

Byte	0	1	2	3
Value	Start ID	Send ID	Command ID	Stop ID

Table 6.6 Communication Format for Monitoring (PC -> RX130)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x01	Communication command ID (monitor start request)
	0x02	Communication command ID (monitor stop request)
Stop ID	0x0a	Stop communication data

6.4.3.2 Communication Format for Parameter Reading

The following communication formats are used when reading parameters.

- RX130 (Capacitive Liquid Level Indicator Demo System) -> PC (Level Monitor)

Bytes 0 – 2

Byte	0	1	2
Value	Start ID	Send ID	Command ID

Bytes 3 – 20

Byte	3	4	5 6	7 8	9 10	11 12	13 14	15 16	17 – 20
Value	Method Number 0	TS Number 0	Lower Threshold 0	Upper Threshold 0	Max Height 0	Max Volume 0	Offset 0	CTSU Base Clock 0	Reserved

Bytes 21 – 38

Byte	21	22	23 24	25 26	27 28	29 30	31 32	33 34	35 – 38
Value	Method Number 1	TS Number 1	Lower Threshold 1	Upper Threshold 1	Max Height 1	Max Volume 1	Offset 1	CTSU Base Clock 1	Reserved

Bytes 39 – 79

Byte	39 – 74	75 76	77 78	79
Value	Reserved	Moving Average Filter	Reserved	Stop ID

Table 6.7 Communication Format for Reading Parameters (RX130 -> PC)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x84	Communication command ID
Method Number 0	0x00	TS17 method number
TS Number 0	0x11	TS17
Lower Threshold 0	0x0000 – 0xffff	TS17 lower threshold
Upper Threshold 0	0x0000 – 0xffff	TS17 upper threshold
Max Height 0	0x0000 – 0xffff	TS17 maximum height
Max Volume 0	0x0000 – 0xffff	TS17 maximum volume
Offset 0	0x0000 – 0xffff	TS17 CTSUSO0 setting value
CTSU Base Clock 0	0x0000 – 0xffff	TS17 CTSU base clock
Method Number 1	0x00	TS1 method number
TS Number 1	0x01	TS1
Lower Threshold 1	0x0000 – 0xffff	TS1 lower threshold
Upper Threshold 1	0x0000 – 0xffff	TS1 upper threshold
Max Height 1	0x0000 – 0xffff	TS1 maximum height
Max Volume 1	0x0000 – 0xffff	TS1 maximum volume
Offset 1	0x0000 – 0xffff	TS1 CTSUSO0 setting value
CTSU Base Clock 1	0x0000 – 0xffff	TS1 CTSU base clock
Reserved	0x00	Reserved area
Stop ID	0x0a	Stop communication data

- PC (Level Monitor) -> RX130 (Capacitive Liquid Level Indicator Demo System)

Bytes 0 – 3

Byte	0	1	2	3
Value	Start ID	Send ID	Command ID	Stop ID

Table 6.8 Communication Format for Reading Parameters (PC -> RX130)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x04	Communication command ID
Stop ID	0x0a	Stop communication data

6.4.3.3 Communication Format for Writing Parameters

The following communication formats are used when writing parameters.

- RX130 (Capacitive Liquid Level Indicator Demo System) -> PC (Level Monitor)

Bytes 0 – 4

Byte	0	1	2	3
Value	Start ID	Send ID	Command ID	Stop ID

Table 6.9 Communication Format for Writing Parameters (RX130 -> PC)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x85	Communication command ID
Stop ID	0x0a	Stop communication data

- PC (Level Monitor) -> RX130 (Capacitive Liquid Level Indicator Demo System)

Bytes 0 – 2

Byte	0	1	2
Value	Start ID	Send ID	Command ID

Bytes 3 – 14

Byte	3	4	5 6	7 8	9 10	11 – 14
Value	Method Number 0	TS Number 0	Lower Threshold 0	Upper Threshold 0	Offset 0	Reserved

Bytes 15 – 26

Byte	15	16	17 18	19 20	21 22	23 – 26
Value	Method Number 1	TS Number 1	Lower Threshold 1	Upper Threshold 1	Offset 1	Reserved

Bytes 27 – 51

Byte	27 – 50	51
Value	Reserved	Stop ID

Table 6.10 Communication Format for Writing Parameters (PC -> RX130)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x05	Communication command ID
Method Number 0	0x00	TS17 method number
TS Number 0	0x11	TS17
Lower Threshold 0	0x0000 – 0xffff	TS17 lower threshold
Upper Threshold 0	0x0000 – 0xffff	TS17 upper threshold
Offset 0	0x0000 – 0xffff	TS17 CTSUSO0 setting value
Method Number 1	0x00	TS1 method number
TS Number 1	0x01	TS1
Lower Threshold 1	0x0000 – 0xffff	TS1 lower threshold
Upper Threshold 1	0x0000 – 0xffff	TS1 upper threshold
Offset 1	0x0000 – 0xffff	TS1 CTSUSO0 setting value
Reserved	0x00	Reserved area
Stop ID	0x0a	Stop communication data

6.4.3.4 Communication Format for Initiating Parameters

The following communication formats are used when initiating parameters.

- RX130 (Capacitive Liquid Level Indicator Demo System) -> PC (Level Monitor)

Bytes 0 – 4

Byte	0	1	2	3
Value	Start ID	Send ID	Command ID	Stop ID

Table 6.11 Communication Format for Writing Parameters (RX130 -> PC)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x86	Communication command ID
Stop ID	0x0a	Stop communication data

- PC (Level Monitor) -> RX130 (Capacitive Liquid Level Indicator Demo System)

Bytes 0 – 7

Byte	0	1	2	3	4	5	6	7
Value	Start ID	Send ID	Command ID	TS Number 0	TS Number 1	TS Number 2	TS Number 3	Stop ID

Table 6.12 Communication Format for Parameter Initialization (PC -> RX130)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x86	Communication command ID
TS Number 0	0x11	TS17
TS Number 1	0x01	TS1
TS Number 2	0xff	Unused channel
TS Number 3	0xff	Unused channel
Stop ID	0x0a	Stop communication data

6.4.3.5 Communication Format for Data Flash Update

The following communication formats are used when writing the parameters to the data flash.

- RX130 (Capacitive Liquid Level Indicator Demo System) -> PC (Level Monitor)

Bytes 0 – 4

Byte	0	1	2	3	4
Value	Start ID	Send ID	Command ID	Sub Command ID	Stop ID

Table 6.13 Communication Format for Data Flash Update (RX130 -> PC)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x87	Communication command ID
Sub Command ID	0x00	Upper 4 bits: Execution results (0: success, 1: error) Lower 4 bits: sub communication command ID
Stop ID	0x0a	Stop communication data

- PC (Level Monitor) -> RX130 (Capacitive Liquid Level Indicator Demo System)

Bytes 0 – 4

Byte	0	1	2	3	4
Value	Start ID	Send ID	Command ID	Sub Command ID	Stop ID

Table 6.14 Communication Format for Data Flash Update (PC -> RX130)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x87	Communication command ID
Sub Command ID	0x00	Sub communication command ID
Stop ID	0x0a	Stop communication data

6.4.3.6 Communication Format for Confirmation of Data Flash Update Completion

The following formats are used when confirming completion of data flash update.

- RX130 (Capacitive Liquid Level Indicator Demo System) -> PC (Level Monitor)

Bytes 0 – 4

Byte	0	1	2	3	4
Value	Start ID	Send ID	Command ID	Sub Command ID	Stop ID

Table 6.15 Communication Format for Confirmation of Data Flash Update Completion (RX130 -> PC)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x87	Communication command ID
Sub Command ID	0x01	Upper 4 bits: Execution results (0: success, 1: error) Lower 4 bits: sub communication command ID
Stop ID	0x0a	Stop communication data

- PC (Level Monitor) -> RX130 (Capacitive Liquid Level Indicator Demo System)

Bytes 0 – 4

Byte	0	1	2	3	4
Value	Start ID	Send ID	Command ID	Sub Command ID	Stop ID

Table 6.16 Communication Format for Confirmation of Data Flash Update Completion (PC -> RX130)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x87	Communication command ID
Sub Command ID	0x01	Sub communication command ID
Stop ID	0x0a	Stop communication data

6.4.3.7 Communication Format for Auto Offset Adjustment

The following formats are used when executing auto offset adjustment.

- RX130 (Capacitive Liquid Level Indicator Demo System) -> PC (Level Monitor)

Bytes 0 – 4

Byte	0	1	2	3	4
Value	Start ID	Send ID	Command ID	Sub Command ID	Stop ID

Table 6.17 Communication Format for Auto Offset Adjustment (RX130 -> PC)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x87	Communication command ID
Sub Command ID	0x02	Upper 4 bits: Execution results (0: success, 1: error) Lower 4 bits: sub communication command ID
Stop ID	0x0a	Stop communication data

- PC (Level Monitor) -> RX130 (Capacitive Liquid Level Indicator Demo System)

Bytes 0 – 4

Byte	0	1	2	3	4
Value	Start ID	Send ID	Command ID	Sub Command ID	Stop ID

Table 6.18 Communication Format for Auto Offset Adjustment (PC -> RX130)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x87	Communication command ID
Sub Command ID	0x02	Sub communication command ID
Stop ID	0x0a	Stop communication data

6.4.3.8 Communication Format for Confirmation of Auto Offset Adjustment Completion

The following are used when confirming completion of auto offset adjustment.

- RX130 (Capacitive Liquid Level Indicator Demo System) -> PC (Level Monitor)

Bytes 0 – 4

Byte	0	1	2	3	4
Value	Start ID	Send ID	Command ID	Sub Command ID	Stop ID

Table 6.19 Communication Format for Confirmation of Auto Offset Adjustment Completion (RX130 -> PC)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x87	Communication command ID
Sub Command ID	0x03	Upper 4 bits: Execution results (0: success, 1: error) Lower 4 bits: sub communication command ID
Stop ID	0x0a	Stop communication data

- PC (Level Monitor) -> RX130 (Capacitive Liquid Level Indicator Demo System)

Bytes 0 – 4

Byte	0	1	2	3	4
Value	Start ID	Send ID	Command ID	Sub Command ID	Stop ID

Table 6.20 Communication Format for Confirmation of Auto Offset Adjustment Completion (PC -> RX130)

Item	Value	Notes
Start ID	0x55	Start communication data
Send ID	0x00	Always 0x00
Command ID	0x87	Communication command ID
Sub Command ID	0x03	Sub communication command ID
Stop ID	0x0a	Stop communication data

Revision History

Rev.	Issued	Description	
		Page	Summary
1.00	Aug.28.20	-	First edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.