S1C17 Family Application Note

S1C17M02/M03 Application Note

SEIKO EPSON CORPORATION

Rev.1.2

NOTICE: PLEASE READ THE FOLLOWING NOTICE CAREFULLY BEFORE USING THIS DOCUMENT

The contents of this document are subject to change without notice.

1. This document may not be copied, reproduced, or used for any other purpose, in whole or in part, without the consent of the Seiko Epson Corporation ("Epson").
2. Before purchasing or using Epson products, please contact our sales representative for the latest information and always be sure to check the latest information published on Epson's official web sites and other sources.
3. Information provided in this document such as application circuits, programs, usage, etc., are for reference purposes only. Using the application circuits, programs, usage, etc. in the design of your equipment or systems is your own responsibility. Epson makes no guarantees against any infringements or damages to any third parties' intellectual property rights or any other rights resulting from the information. This document does not grant you any licenses, intellectual property rights or any other rights with respect to Epson products owned by Epson or any third parties.
4. Epson is committed to constantly improving quality and reliability, but semiconductor products in general are subject to malfunction and failure. By using Epson products, you shall be responsible for your hardware. Software and systems must be designed well enough to prevent death or injury as well as any property damage even if any of the malfunctions or failures might be caused by Epson products. When designing your products using Epson products, please be sure to check and comply with the latest information regarding Epson products (this document, specifications, data sheets, manuals, Epson's web site, etc.). When using the information included above materials such as product data, charts, technical contents, programs, algorithms and application circuit examples, you shall evaluate your products both on a stand-alone basis as well as within your overall systems. You shall be solely responsible for deciding whether or not to adopt and use Epson products.
5. Epson has prepared this document and programs provided in this document carefully to be accurate and dependable, but Epson does not guarantee that the information and the programs are always accurate and complete. Epson assumes no responsibility for any damages which you incur due to misinformation in this document and the programs.
6. No dismantling, analysis, reverse engineering, modification, alteration, adaptation, reproduction, etc., of Epson products is allowed.
7. Epson products have been designed, developed and manufactured to be used in general electronic applications (office equipment, communications equipment, measuring instruments, home electronics, etc.) ("General Purpose") and applications which is individually listed in this document or designated by Epson ("Designated Purpose"). Epson products are NOT intended for any use beyond the General Purpose and Designated Purpose uses that requires particular/higher quality or reliability in order to refrain from causing any malfunction or failure leading to death, injury, serious property damage or severe impact on society, including, but not limited to those listed below ("Particular Purpose"). Therefore, you are advised to use Epson products only for General Purpose and Designated Purpose uses. Should you desire to buy and use Epson products for a Particular Purpose, Epson makes no warranty and disclaims with respect to Epson products, whether express or implied, including without limitation any implied warranty of merchantability or fitness for any Particular Purpose. Please be sure to contact our sales representative and obtain approval in advance.

   [Examples of Particular Purpose]

   Space equipment (artificial satellites, rockets, etc.) /
   Transportation vehicles and their control equipment (automobiles, aircraft, trains, ships, etc.) /
   Medical equipment / Relay equipment to be placed on ocean floor /
   Power station control equipment / Disaster or crime prevention equipment / Traffic control equipment / Financial equipment

   Other applications requiring similar levels of reliability as those listed above. Please be sure to contact our sales representative for details of the other applications.
8. Epson products listed in this document and our associated technologies shall not be used in any equipment or systems that laws and regulations in Japan or any other countries prohibit to manufacture, use or sell. Furthermore, Epson products and our associated technologies shall not be used for developing weapons of mass destruction, or any other military purposes or applications. If exporting Epson products or our associated technologies, you shall comply with the Foreign Exchange and Foreign Trade Control Act in Japan, Export Administration Regulations in the U.S.A. (EAR) and other export-related laws and regulations in Japan and any other countries and follow the required procedures as provided by the relevant laws and regulations.
9. Epson assumes no responsibility for any damages (whether direct or indirect) caused by or in relation with your non-compliance with the terms and conditions in this document.
10. Epson assumes no responsibility for any damages (whether direct or indirect) incurred by any third party that you assign, transfer, loan, etc., Epson products to.
11. For more details or other concerns about this document, please contact our sales representative.
12. Company names and product names listed in this document are trademarks or registered trademarks of their respective companies.

Evaluation board/kit and Development tool important notice

1. Epson evaluation board/kit or development tool is designed for use for engineering evaluation, demonstration, or development purposes only. Do not use it for other purposes. It is not intended to meet the requirements of design for finished products.
2. Epson evaluation board/kit or development tool is intended for use by an electronic engineer and is not a consumer product. The user should use it properly and in a safe manner. Epson does not assume any responsibility or liability of any kind of damage and/or fire caused by the use of it. The user should cease to use it when any abnormal issue occurs even during proper and safe use.
3. The part used for Epson evaluation board/kit or development tool may be changed without any notice.

Rev. e1.5, 2025. 7

Table of Contents

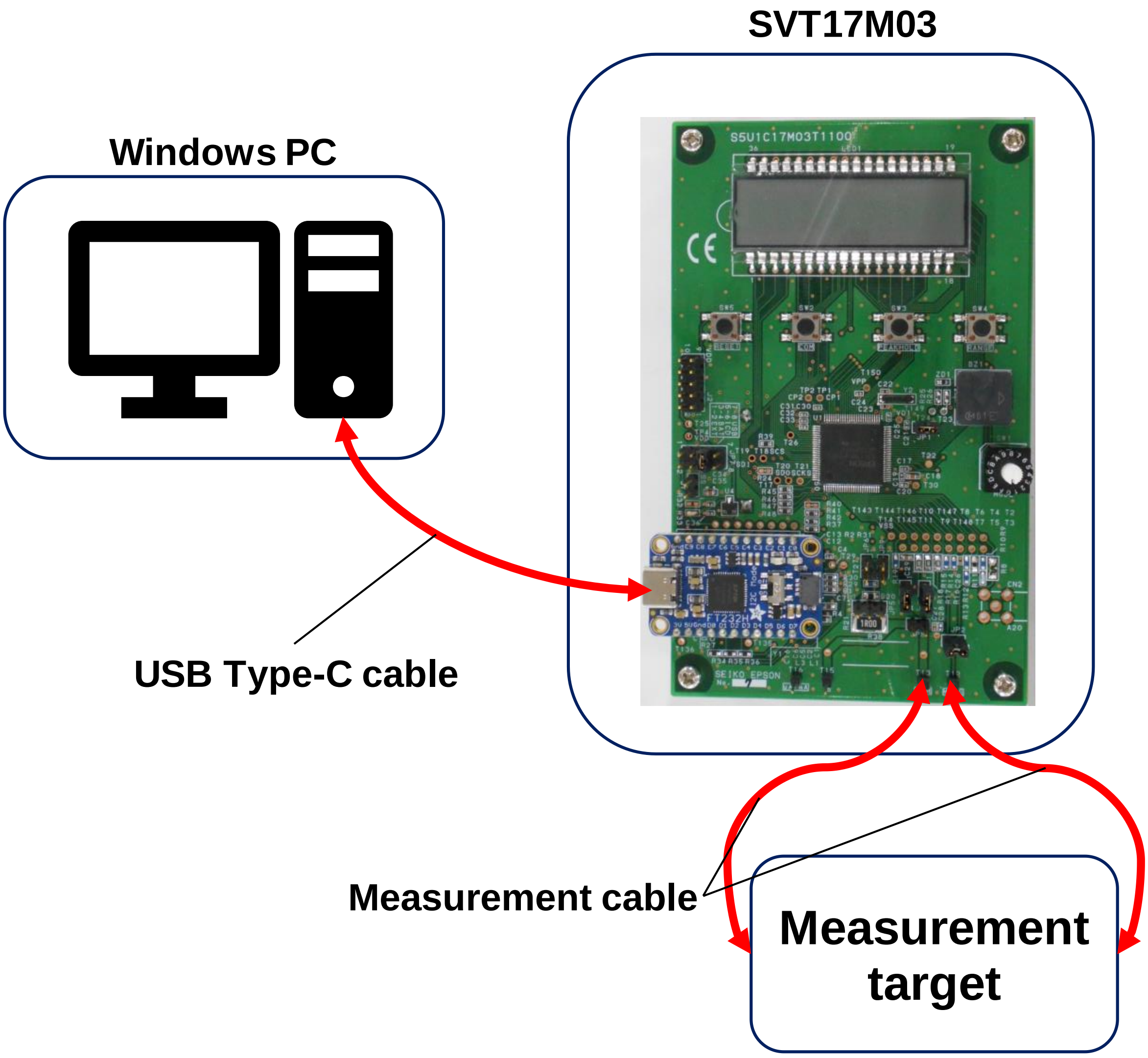
1. Overview

The Reference Solution for Seiko Epson Digital Multimeters (DMMs) is a solution package for designing DMMs using Seiko Epson's microcontroller S1C17M03.

This package consists of the reference board S5U1C17M03T1100(SVT17M03), reference software, and application note (this document), and is provided to enable the efficient design of DMMs with a small number of person-hours.

This application note is intended for use with the software package S1C17M02_M03_demo_software_v2.X.

1.1. System Configuration



Note: Please provide your own USB cable, and Measurement cable.

Figure 1.1-1 Hardware Configuration

Table 1.1-1 List of Offerings

	Items	Contents	Descriptions
1	Hardware	S5U1C17M03T(SVT17M03)	Evaluation board
2	Software	S1C17M02_M03_demo_software_vX.X.zip*	Software package
3	Document	This document	Application note

*X: Software version

Table 1.1-2 Contents of Software Package

	Items	Descriptions
1	s1c17m02_m03_application_verX.X* └ s1c17m02_m03_application_gnu17v3	Project folder
2	DmmTools_X.XX* ├ DmmEvalTool.exe └ dll files	Communication software for PC
3	Dmm_readme_en.txt Dmm_readme_jp.txt	Readme file in English Readme file in Japanese
4	License_e.txt	Software License Agreement
5	S5U1C17M03Txxxx_jumper connections_en.xlsx* S5U1C17M03Txxxx_ jumper connections_jp.xlsx*	Jumper configuration table in English Jumper configuration table in Japanese
6	DMM_Calibration_Sheet_en.xlsx DMM_Calibration_Sheet_jp.xlsx	Header file creation sheet for calibration in English Header file creation sheet for calibration in Japanese

*X: Software version

2. System Resources

2.1. Microcontroller Resources

Table 2.1-1 Microcontroller Resources

Items	Contents
CPU	S1C17M03
System clock	3.2MHz (Internal OSC3)

2.2. Functional Specification Description

PPORT	P07 of PPORT is used for output as a timing signal for SPI communication. P40, P41, P42, and P43 are used as inputs for rotary switches. P44, P45, and P46 are used as tactile switch inputs.
DMM controller	Used for measurement as a DMM.
LCD4B	Controls the LCD for display.
SNDA	Used to play beep sound by a buzzer.
SPIA Ch.0	Assign P00 of PPORT to #SPISSn, P01 to SDO, and P03 to SPICLK with using UPMUX. Used for communication with PC.
T16 Ch.0	Used for mode name display and DMM controller timeout.
T16 Ch.1	Used for polling scan of various switches and LCD display switching.
T16B_DMM Ch.0	Frequency calculation: Used for the frequency measurement period.
T16B_DMM Ch.1	Frequency calculation: Used to capture the measurement waveform signal during the frequency measurement period. Assign the tout output of T16B_DMM Ch.0 to CAP.
T16B_DMM Ch.2	Frequency calculation: Used to count during one cycle of the measurement waveform. Assign the measurement waveform pulse signal to CAP.
SVD4	Used to detect the battery level.
UART3 Ch.0	Assign P03 of PPORT to USIN and P02 to USOUT with using UPMUX. Used for communication with PC.
Clock	OSC3 (internal oscillation: 3.2MHz) is used as the system clock. OSC1 is used to drive LCD4B, SNDA, T16 Ch.0, T16B_DMM Ch.0, T16B_DMM Ch.2, and SVD4. OSC3 is used to drive the DMM controller, and T16 Ch.1.

3. Operation Description

3.1. Operation Overview

This system uses the DMM controller and 16-bit PWM timer for DMM in the S1C17M03 on the SVT17M03 to perform measurements in the corresponding measurement range. Although this software can use all measurement settings available in the S1C17M03, the actual measurement is limited to the measurement modes available due to the limitations of the SVT17M03. Please refer to the separately published manual “S5U1C17M03T Manual” for details.

Table 3.1-1 Measurement Modes and Ranges Available

Measurement modes	Measurement ranges
DC voltage	600 mV/ 6 V / 60 V / 600 V / 1,000 V
AC voltage	600 mV / 6 V / 60 V / 600 V / 1,000 V
DC current	600 uA / 6 mA / 60 mA / 600 mA / 6 A / 10 A
AC current	600 uA / 6 mA / 60 mA / 600 mA / 6 A / 10 A
Resistance (CC-method)	600 Ω / 6kΩ / 60 kΩ / 600 kΩ / 6 MΩ / 60 MΩ
Resistance (CV-method)	600 Ω / 6 kΩ / 60 kΩ
Continuity check	CV / CC
Capacitance (CC-method)	1 uF / 10 uF / 100 uF / 1,000 uF
Capacitance (CV-method)	10 nF / 100 nF
Diode VF	-
AC voltage and frequency	6 V
AC current and frequency	6 mA
Internal temperature	-

Among the measurement modes listed above, the values obtained for DC voltage, AC voltage, resistance (CC method), resistance (CV method), capacitance (CC method), and capacitance (CV method) are the corrected values obtained by calibration using linear interpolation. For details on the calibration process, please refer to section "5.5.4 Calculation by Calibration."

The measured results will be output through the following methods.

- Displayed on the LCD panel on the SVT17M03
- Transmission to a PC via UART or SPI communication*
- Play the continuity check buzzer on the SVT17M03

*The internal temperature measurement does not display the result on the LCD panel but outputs the measurement register value through serial communication.

3.2. Measurement Configuration Switching

When performing measurement operations as a DMM, the user can change the measurement mode and other settings and start transmitting measurement results via serial communication by operating the tactile switches (SW2 to SW5) and rotary switch (SW1) mounted on the SVT17M03.

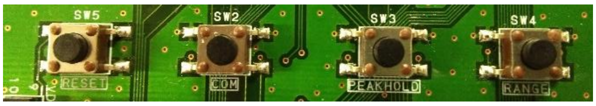


Figure 3.2-1 Tactile Switches (SW2 to SW5)

Figure 3.2-2 Rotary Switch (SW1)

Table 3.2-1 Operation When Each Switch is pressed

Switches		Operations
SW1		Measurement mode switching
SW2		Start/Stop communication mode Data-hold mode switching (from ON to OFF) Recovery from sleep mode after auto power off
SW3		Peak-hold configuration switching Recovery from sleep mode after auto power off
SW4		Measurement range switching Auto-range mode switching (from ON to OFF) Relative mode switching (from ON to OFF) Recovery from sleep mode after auto power off
SW5		Perform a hardware reset
SW3 Long press	SW2	Data-hold mode switching (from OFF to ON)
	SW4	Relative mode switching (from OFF to ON)
SW4 Long press		Auto-range mode switching (from OFF to ON)

3.2.1. Measurement Mode Switching

At initial startup and during measurement or communication mode, rotating the rotary switch (SW1) on the SVT17M03 sets the DMM controller's measurement mode setting to the value corresponding to the number on the switch. When switching, the name of the target measurement mode is displayed on the LCD on the SVT17M03 for 1 second. Also, the measurement range is set to the default range.

Figure 3.2.1-1 LCD Screen When Switched to DC Voltage Measurement

SW1 #	Measurement modes	Default ranges	Mode names
0	DC voltage	6V	DCV
1	AC voltage	6V	ACV
2	DC current	6mA	DCI
3	AC current	6mA	ACI
4	Resistance (CC-method)	600Ω	OHM CC
5	Resistance (CV-method)	600Ω	OHM CV
6	Continuity check	CV	CONT
7	Capacitance (CC-method)	1uF	CAP CC
8	Capacitance (CV-method)	10nF	CAP CV
9	Diode VF	-	DIODE
A	AC voltage and frequency	6V	FREQ ACV
B	AC current and frequency	6mA	FREQ ACI
C	Internal temperature	-	TEMP

Table 3.2.1-1 SW1 Number and Corresponding Measurement Mode

Note: Unused number is selected, no measurement is taken and "NOFUNC" is displayed on the LCD.

After the measurement mode is displayed, the measurement operation as a DMM is performed. The measured value is displayed on the LCD on the board as shown below. In the case of continuity check mode, the buzzer on the board plays when continuity is detected.

Figure 3.2.1-2 LCD Screen Displaying Measured Value

Figure 3.2.1-3 Buzzer

3.2.2. Peak-Hold Configuration Switching

Pressed each time the tactile switch (SW3) in the measurement mode, the peak hold function can be toggled one level at a time, and the enabled setting can be checked from the “ ’ ” indicator in the upper left corner of the LCD screen.

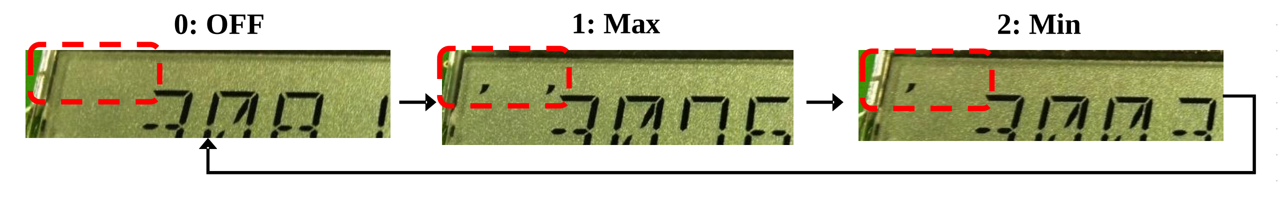


Figure 3.2.2-1 LCD Display for Peak-Hold Switching

3.2.3. Measurement Range Switching

Pressed each time the tactile switch (SW4) during measurement mode, the measuring range can be switched one step at a time within the settable range.

6V → 60V → 600V → 1,000V → 600mV → (back to 6V)

Figure 3.2.3-1 Example) Measurement Range Switching for DC Voltage Measurement

3.2.4. Auto-range Switching

Long-pressing the tactile switch (SW4) during measurement mode activates the auto-range mode, and the “ ’ ” indicator appears at the upper right corner of the LCD screen. Pressing SW4 again while the auto-range mode is active deactivates the mode.

Figure 3.2.4-1 LCD Display for Auto-Range Switching

3.2.5. Data-Hold Switching

By long-pressing the tactile switch (SW4) while simultaneously pressing the tactile switch (SW2), the data-hold mode is activated and three “ ’ ” indicators appear at the upper left corner of the LCD screen. Pressing SW2 again while the data hold mode is active will deactivate it.

Figure 3.2.5-1 LCD display for Data-Hold Switching

3.2.6. Relative Mode Switching

When the data hold mode is active, long-pressing the tactile switch (SW3) while pressing the tactile switch (SW4) activates the relative mode. Four “ ’ ” indicators appear at the upper left corner of the LCD screen when relative mode is ON. Pressing SW4 again while the relative mode is active will deactivate it.

Figure 3.2.6-1 LCD Display for Relative Switching

3.3. Communication Mode

Pressing the tactile switch (SW2) on the SVT17M03 while executing the measurement mode corresponding to the communication mode initiates the communication mode and places the device on standby for transmission of measurement results via serial communication. The communication mode is terminated when the specified number of measurement data is completed to be sent during the communication in progress mode or when the tactile switch (SW2) or rotary switch (SW1) on the board is rotated. During the communication mode, "COM" is displayed on the LCD.

Figure 3.3-1 LCD Display During Communication Mode

3.3.1. Communication Method

This software can use UART or SPI as the communication method. To change the communication method, the following description of the "s1c17m02_m03_application_gnu17v3" source code must be changed.

- When using UART communication
  Comment out the following description in “src/mid_communication.h”
  #define COMMUNICATION_MODE_SPI

- When using SPI communication
  If the following description in “src/mid_communication.h” is commented out, uncomment it.
  #define COMMUNICATION_MODE_SPI

3.3.1.1. UART communication

When the communication mode is started with UART communication enabled, the transmission of measurement results via UART begins.

(1) Communications protocol

Table 3.3.1.1-1 UART Communication Protocol

Items	Setup values
Baud rate	230,400 bps
Data length	8 bit
Stop bit	1 bit
Parity	none
Flow control	none

(2) Operation check procedure
1. Set the measurement settings for data communication using the rotary switch or tactile switches on the SVT17M03
2. Connect the PC to the board with a USB Type-C cable
3. Start terminal software (e.g., Tera Term) on PC and connect to SVT17M03
4. Press the tactile switch (SW2) on the board to start communication mode and send data
5. The terminal software displays the received data.

3.3.1.2. SPI communication

To communicate measurement data via SPI communication, it is necessary to operate “DmmEvalTool”, the PC communication software, at the same time as operating the SVT17M03. In this state, clicking the START button on the “DmmEvalTool” executes communication.

(1) DmmEvalTool
DmmEvalTool is included in the software package. Double-click “DmmEvalTool.exe” under the “DmmTools_X.XX” (X: version) folder in the package to start it.
When DmmEvalTool is started, the following window will appear on the PC screen.

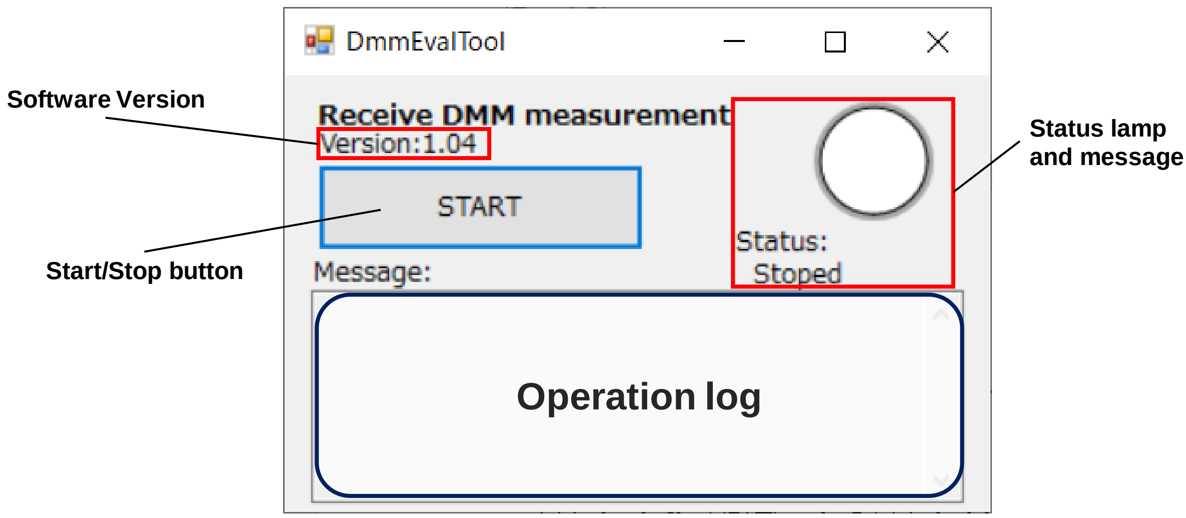


Figure 3.3.1.2-1 DmmEvalTool Window

When communication is completed according to the method described below, DmmEvalTool creates a "Logs" folder in the same directory as the execution file and saves the measurement data inside the folder under the name "Output.csv.
The data format recorded in "Output.csv" is as follows. If you perform a measurement multiple times, the second and subsequent measurements are recorded and updated in the second and subsequent rows of "Output.csv".

Table 3.3.1.2-1 Measurement Data Recording Format

Rows	Contents
1	Number of measurements
2	Measurement mode
3	Measurement range
4	Peak-hold configuration
5~	Measurement data

(2) Communications protocol

Table 3.3.1.2-2 SPI Communications Protocol

Items	Setup values
Control	Slave
Data sequence	MSB first
SPI mode	Mode 3
Data length	8 bit

(3) Operation check procedure

1. Set the measurement settings for data communication using the rotary switch or tactile switched on the SVT17M03
2. Connect the PC to the board with a USB Type-C cable
3. Start DmmEvalTool and connect to SVT17M03
4. Press the tactile switch (SW2) on the board to start the communication mode and put the board on standby for data transmission
5. Click the START button on DmmEvalTool to start communication
6. When communication is completed, DmmEvalTool generates a CSV file of the received measurement data

3.3.1.3. SPI data transmission

In data communication using DmmEvalTool, this software modifies a part of the SPI communication protocol and communicates using a method unique to this software.

This software does not use the SDI signal in SPI communication, but only the SDO signal for one-way communication. Therefore, one port of GPIO output is used as a timing signal to inform the communication partner of the start and end of data communication. P07 is used as the GPIO output.

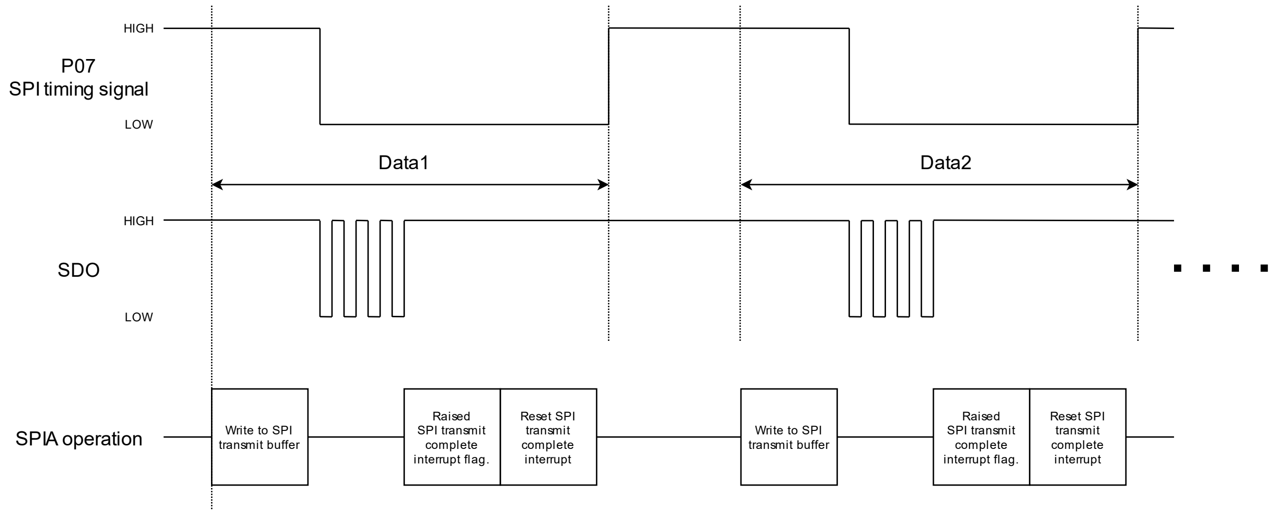


Figure 3.3.1.3-1 Data Communication via SPI

3.3.2. Transmission Data Format

The data transmitted from SVT17M03 in the communication mode consists of three consecutive pieces of data: measurement mode data, measurement data, and communication termination data, and is transmitted using the following data format.

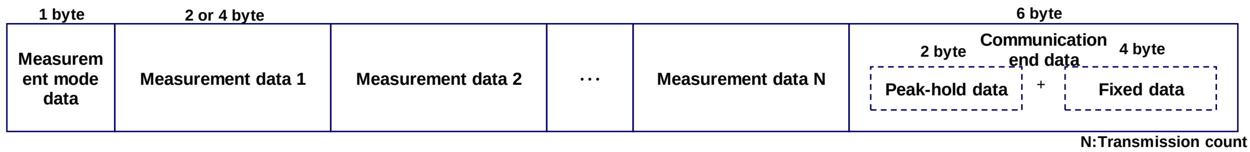


Figure 3.3.2-1 Communication Data Format

3.3.2.1. Measurement Mode Data

When the communication mode is initiated, one byte of measurement mode data is sent. The measurement mode data consists of the following data to identify measurement information. Each of the measurement range and measurement mode information that make up the measurement mode data corresponds to each measurement setting register value of the DMM controller.

Table 3.3.2.1-1 Measurement Mode Data Format

Bit #	Contents
7	0
6:4	DSADC16CTL.RANGESEL[3:0] (Measurement range)
3:0	DSADC16CTL.FUNCSEL[3:0] (Measurement mode)

3.3.2.2. Measurement Data

After measurement mode data transmission is complete, measurement data is transmitted next. The measurement data consists of the measurement result register value or the calculated measurement value as shown below. For details on the measurement result register values, please refer to the “S1C17M02/M03 Technical Manual” published separately.
As mentioned above, measurement data is transmitted until the number of data specific to each measurement mode has been transmitted, the tactile switch (SW2) is pressed, or the rotary switch (SW1) is turned.

Table 3.3.2.2-1 Measurement Data Format

Measurement mode	Data length	Data type	Transmit data counts
DC voltage	2 byte	Signed integer	500
AC voltage	4 byte	Unsigned integer	250
DC current	2 byte	Signed integer	500
AC current	4 byte	Unsigned integer	250
Resistance (CC-method)	2 byte	Signed integer	500
Resistance (CV-method)	2 byte	Signed integer	500
Continuity check		Communication unsupported	
Capacitance (CC-method)	4 byte	Single precision floating point	100
Capacitance (CV-method)	4 byte	Single precision floating point	100
Diode VF	2 byte	Signed integer	500
AC voltage and frequency	4 byte	Single precision floating point	100
AC current and frequency	4 byte	Single precision floating point	100
Internal temperature	2 byte	Signed integer	500

3.3.2.3. Communication end data

When the specified number of measurement data is sent, or when the tactile switch (SW2) is pressed or the rotary switch (SW1) is turned, communication end data is sent and the communication mode is terminated.
The communication end data consists of 2 bytes of data related to the peak hold function and 4 bytes of fixed data, for a total of 6 bytes of data.

Table 3.3.2.3-1 Communication End Data Format

Peak-hold functions	Peak-hold data	Fixed data
Minimum value peak-hold	FF 0D	5A 96 96 5A
Maximum value peak-hold	FF 0E	5A 96 96 5A
OFF	FF 0F	5A 96 96 5A

3.4. Low battery Mode

During the operation of this software, the low battery mode is initiated when the supply voltage supplied to the SVT17M03 falls below 2.2V and is detected by the Supply Voltage Detector (SVD4). In this mode, the supply of clock pulses to all other peripheral circuits except for the circuits necessary for the LCD is stopped, and the LCDs "LOW BATT" as shown in the figure below prompts the user to replace the batteries.

Figure 3.4-1 LCD Display During Low Battery Mode

3.5. Auto Power Off Mode

In this software, if no tactile switch operation is detected for a certain period (Default: 60 s, configurable via software), the LCD automatically turns off and the microcontroller enters sleep mode. Pressing any of the tactile switches (SW2, SW3, SW4) will exit sleep mode.

3.6. Auto-Range Mode

This software includes an auto-range mode that automatically switches the measurement range according to the measured value. Pressing and holding SW4 during measurement mode activates the auto-range mode. Pressing SW4 again while the auto-range mode is active deactivates the mode. The availability of the auto-range mode for each measurement mode is as follows.

Table 3.6-1 Measurement Modes Supporting Auto-Range Mode

Measurement mode	Auto-range mode support
DC voltage	Supported (*1)
AC voltage	Supported (*1)
DC current	Supported (*2)
AC current	Supported (*2)
Resistance (CC-method)	Supported
Resistance (CV-method)	Supported
Continuity check	Not supported
Capacitance (CC-method)	Supported
Capacitance (CV-method)	Supported
Diode VF	Not supported
AC voltage and frequency	Not supported
AC current and frequency	Not supported
Internal temperature	Not supported

*1 Due to the need to change jumper settings depending on the measurement range, the auto-range mode may not operate correctly in some cases.
*2 Due to the need to change the jumper settings and measurement terminals depending on the measurement range, the auto-range mode may not operate correctly in some cases.

3.7. Data-Hold Mode

In this software, during measurement, pressing and holding SW3 while pressing SW2 activates the data hold mode. When the data hold mode is turned ON, the measured value at that moment is held, and the held value continues to be displayed on the LCD screen.

3.8. Relative Mode

In this software, when the data hold mode is ON, pressing SW4 while holding down SW3 activates the relative mode, which displays the difference between the current measured value and the held value. While the relative mode is active, the auto-range mode and manual range switching are disabled.

4. Software Description

4.1. Function Blocks

This software consists of a driver layer that directly operates peripheral circuits, an application layer that realizes DMM operation, and a middle layer that serves as an interface between them.

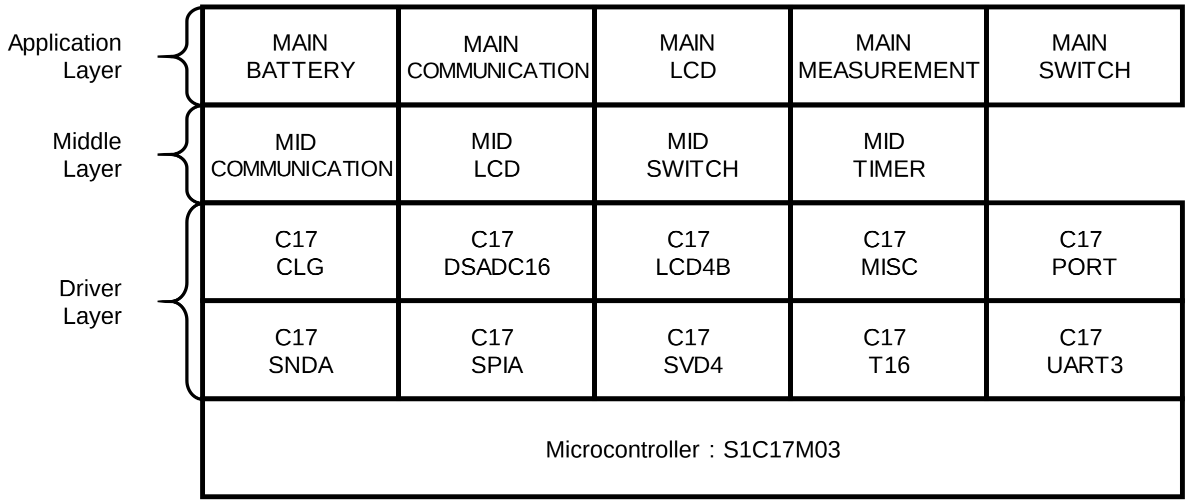


Figure 4.1-1 Function Blocks

Table 4.1-1 Function Block Overview

Function blocks	Description
MAIN BATTERY	Low battery mode
MAIN COMMUNICATION	Communication mode
MAIN LCD	LCD display for various modes
MAIN MEASUREMENT	Measurement mode
MAIN SWITCH	Mode switching
MID COMMUNICATION	Communication interface
MID LCD	LCD interface
MID SWITCH	Switch interface
MID TIMER	Timer interface
C17 CLG	CLG driver
C17 DSADC16	DMM controller driver
C17 LCD4B	LCD4B driver
C17 MISC	MISC driver
C17 PORT	PPORT driver
C17 SNDA	SNDA driver
C17 SPIA	SPIA driver
C17 SVD4	SVD4 driver
C17 T16	T16 driver
C17 UART3	UART3 driver

4.2. Operation Conceptual Diagram

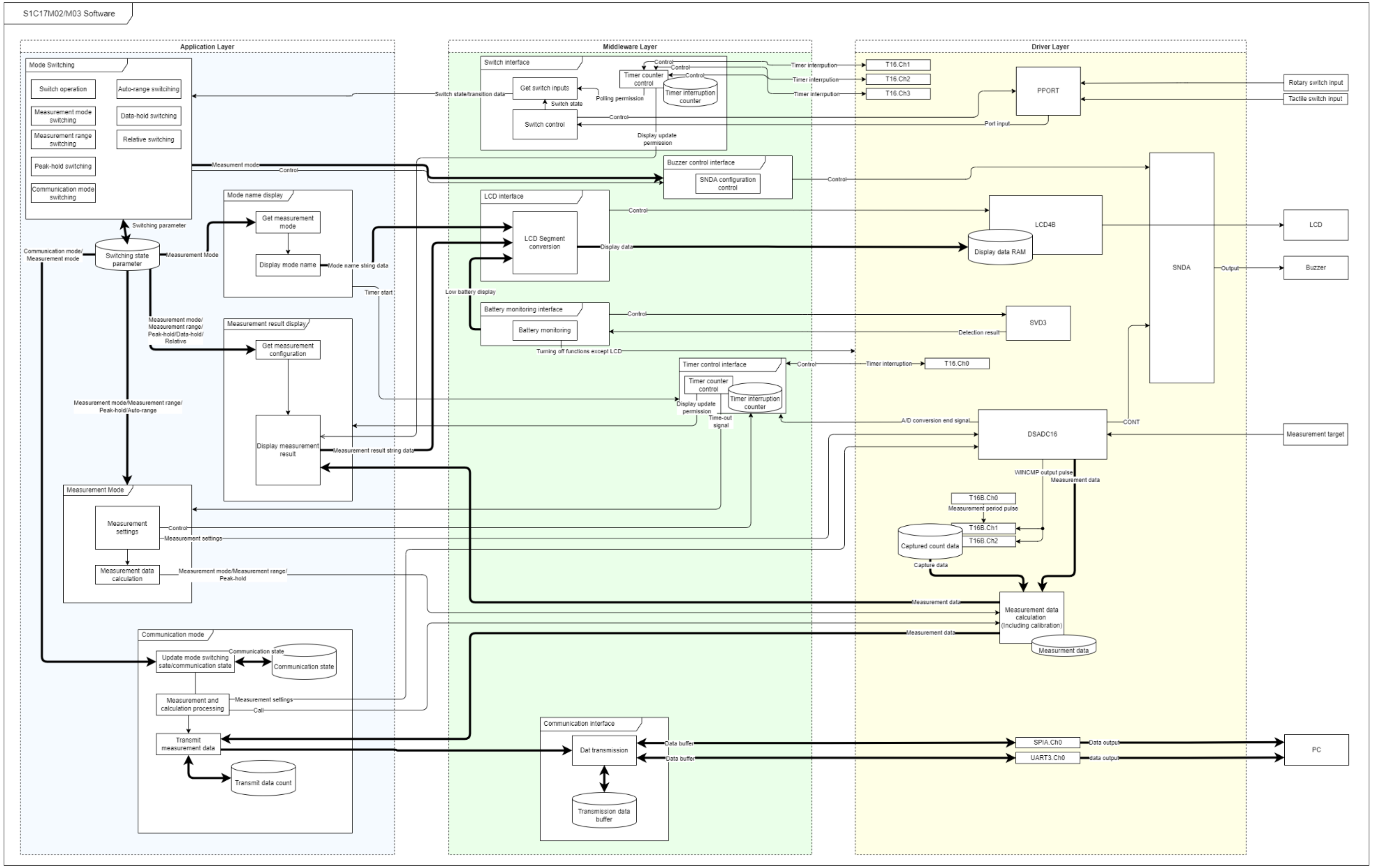


Figure 4.2-1 Software Operation Conceptual Diagram

4.3. About s1c17m02_m03_application_gnu17v3

This section describes the "s1c17m02_m03_application_gnu17v3" software that enables operation as a DMM.

4.3.1. File Structure(src/)

The prefix at the beginning of the file name indicates the application layer (main_), middle layer (mid_), and driver layer (c17_).

Table 4.3.1-1 Application Layer Files

File name	Functions
main_battery.c/.h	Low battery mode program file
main_communication.c/.h	Communication mode program file
main_lcd.c/.h	LCD display for various modes program file
main_measurement.c/.h	Measurement mode program file
main_switch.c/.h	Mode switching program file
main.c	Main routine program file

Table 4.3.1-2 Middle Layer Files

File name	Functions
lcd_font.h	LCD display font definition file
mid_communication.c/.h	Communication interface program file
mid_lcd.c/.h	LCD interface program file
mid_switch.c/.h	Switch interface program file
mid_timer.c/.h	Timer interface program file

4.3.2. File Structure(src/driver/)

The driver folder contains the S1C17M02/M03 peripheral circuit driver files.

Table 4.3.2-1 Driver Layer Files

File name	Functions
boot.c	Device startup initialization program file
c17_clg.c/.h	CLG driver program file
c17_dsadc16.c/.h	DMM controller driver program file
c17_eerpomc.c/.h	EEPROMC driver program file
c17_i2c.c/.h	I2C driver program file
c17_init_config.h	C17 driver configuration file
c17_lcd4b.c/.h	LCD4B driver program file
c17_misc.c/.h	MISC driver program file
c17_port.c/.h	PPORT driver program file
c17_pwg.c/.h	PWG driver program file
c17_snda.c/.h	SNDA driver program file
c17_spia_asm.s	SPI data Tx/Rx Assembly file
c17_spia.c/.h	SPI driver program file
c17_svd4.c/.h	SVD4 driver program file
c17_t16.c/.h	T16 driver program file
c17_uart3.c/.h	UART3 driver program file
c17_wdt2.c/.h	WDT2 driver program file
crt0.h	C runtime file
libfloat.c/.h	Numeric to string conversion program file
strnum.c/.h	String creation and concatenation program file

4.3.3. File Structure(inc/)

The inc folder contains the S1C17M02/M03 peripheral circuit register definition files.

Table 4.3.3-1 Files List in inc Folder

File name	Functions
reg/	S1C17M02/M03 peripheral circuit register definition file folder
c17_mcu_select.h	MCU selection header file
c17m02_reg.h	S1C17M02 peripheral circuit register definition file
c17m03_reg.h	S1C17M03 peripheral circuit register definition file

4.3.4. Configuration of MCU to be Used

In this software, the MCU to be operated can be specified by enabling either one of the following macro descriptions in "inc/c17_mcu_select.h".

```
#define C17_MCUSEL_M02          Enable if operate with S1C17M02
#define C17_MCUSEL_M03          Enable if operate with S1C17M03
```

If the MCU to be used is designated as S1C17M02, some functions will not be available because S1C17M03 uses peripheral functions that are not available in S1C17M02. The following is a list of software functions and Flash resources that can be used with each MCU selects.

Table 4.3.4-1 Available Software Functions

Function blocks	S1C17M02	S1C17M03
Application Layer	Not supported*	✓
Middle Layer	Not supported	✓
Driver Layer	✓	✓

*It is generally not available for use, but main_measurement.c can be utilized with some modifications.

Table 4.3.4-2 Flash Resources

Function blocks	S1C17M02 [byte]	S1C17M03 [byte]
Application Layer	-	9,714
Middle Layer	-	2,669
Driver Layer	12,532	16,936
Standard library	-	12,890

4.3.5. Configuration of Peripheral Circuit to be Used

The software allows configuring the peripheral circuits to be used by writing macros in "src/c17_init_config.h." By enabling the macros defined in the form of C17_PE_XX, the peripheral circuit functions corresponding to the macro names can be used.

Table 4.3.5-1 Peripheral Circuit Function Enable Macro

Macro names	Functions
C17_PE_DSADC16	DMM controller driver
C17_PE_EEPROMC	EEPROMC driver
C17_PE_I2C_MASTER	Master mode of I2C driver
C17_PE_I2C_SLAVE	Slave mode of I2C driver
C17_PE_LCD4B	LCD4B driver
C17_PE_PORT	PPORT driver
C17_PE_PWG	PWG driver
C17_PE_SNDA	SNDA driver
C17_PE_SPIA_SLAVE	Master mode of SPIA driver
C17_PE_SPIA_MASTER	Slave mode of SPIA driver
C17_PE_SVD4	SVD4 driver
C17_PE_T16	T16 driver
C17_PE_UART3	UART3 driver
C17_PE_WDT2	WDT2 driver

4.3.6. Software Modules Description

For each module in the file, the function name and its function are described, focusing on the functions that make up the DMM.

4.3.6.1. main_battery.c

Function name	Functions
initBattery()	Initializes functions related to battery monitoring.
SvdIntHandler()	Interrupt handler for the supply voltage detection circuit. Stops all circuits other than the peripheral circuits required for LCD and displays a low battery on the LCD.
stopPeripheralExceptDispOnLcd()	Stops all circuits other than the peripheral circuits required for LCD.
dispLowBattOnLcd()	Displays a low battery on the LCD.

4.3.6.2. main_communication.c

Function/global variable name	Functions
runComMode()	The main routine that performs the functions of the communication mode
updateComState()	Manages communication status.
setComStartDaraToBuf()	Stores the start data for communication in the transmit data buffer.
setMeasDataToBuf()	Stores the measurement data in the transmit data buffer.
setComEndDataToBuf()	Stores the end data for communication in the transmit data buffer.
sentCount	Transmitted data counter

4.3.6.3. main_lcd.c

Function name	Functions
dispModeNameOnLcd()	Displays mode name and measurement mode name on LCD.
dispResultOnLcd()	Displays measurement result on LCD.

4.3.6.4. main_measurement.c

Function/global variable name	Functions
runMeasMode()	The main routine that performs the function of the measurement mode
stopMeasurement()	Stops measurement
startMeasurement()	Starts measurement
result	Measured value variable after calculation
switchRange()	Switch measurement range by auto range function

4.3.6.5. main_switch.c

Function name	Functions
switchMeasMode()	Performs measurement mode transitions.
switchMeasRange()	Performs measurement range transitions.
switchPeakHold()	Performs peak-hold configuration transitions.
switchAutoRange()	Performs auto-range mode transitions.
switchDataHold()	Performs data-hold mode transitions.
switchRelativeValue()	Performs relative mode transitions.

4.3.6.6. main.c

Function name	Functions
main()	The main routine that realizes the DMM function
initPeripheral()	Initializes all peripheral circuits and interfaces to be used.
setClg()	Sets the clock generator settings.
checkLcdDisp()	Turns on all segments of the LCD.
runAutoPowerOff()	Put the microcontroller into sleep mode if no tactile switch operation is detected for a certain period.
runAutoRange()	Refresh the LCD display if measurement range is changed by the auto-range mode.
dispDataHold()	Refresh the LCD display if the data-hold mode is activated.
dispRelativeValue()	Refresh the LCD display if the relative mode is activated.

4.3.6.7. mid_communication.c

Function/global variable name	Functions
initCommunication()	Initializes the communication interface.
startCommunication()	Starts operation of the communication interface.
stopCommunication()	Stops operation of the communication interface.
initBuf()	Initializes the transmit data buffer.
sendTxDataToBuf()	Stores data in the transmit data buffer.
sendTxBuf()	Transmits the contents of the transmit data buffer.
c17sendSpiaData()	Transmits data via SPI communication.
comMode	Current communication mode state variable

4.3.6.8. mid_lcd.c

Function name	Functions
putStrOnLcd()	Displays any string on the LCD.
putChrOnLcd()	Displays any character on the LCD.
putQuoteOnLcd()	Controls the display of “ ’ ” at any position on the LCD.
getLcdVramAddr()	Gets the VRAM address of any LCD segment.

4.3.6.9. mid_switch.c

Function/global variable name	Functions
initSwitch()	Initializes the switch interface.
updateSwitch()	Manages switch state transitions.
getTactileSwitchInput()	Gets all tactile switch inputs.
getRotarySwitchInput()	Gets rotary switch input.
transition	Switch state transition structure variable
checkLongPress	Detect a long-press on a tactile switch.

4.3.6.10. mid_timer.c

Function/global variable name	Functions
initTimer()	Initializes the timer interface
startTimer()	Starts counting for the specified timer.
stopTimer()	Stops and resets counting for the specified timer.

T16Ch0IntHandler()	T16Ch0 interrupt handler that generates time
T16Ch1IntHandler()	T16Ch1 interrupt handler that generates time
T16Ch2IntHandler()	T16Ch2 interrupt handler that generates time
T16Ch3IntHandler()	T16Ch3 interrupt handler that generates time
modeNameDispFlg	Timer flag variable for mode name display
timeOutFlg	Timeout timer flag variable
swPollingFlg	Timer flag variable for switch polling
lcdDispRefreshFlg	Timer flag variable for LCD display update
powerOffTimeOutFlg	Timer flag variable for auto power off
longPressFlg	Timer flag variable for detection of a long-press
rangeUpFlg	Timer flag variable for range up

4.3.6.11. c17_dsadc16.c

Function/global variable name	Functions
c17calcMeasValue()	Calculates the measured value
c17calcCapasitance()	Calculates the capacitance value
c17intT16bCh1()	Interrupt handler for T16B_DMM Ch1. Counts the values and calculates the frequency required for frequency calculation.
c17intT16bCh2()	Interrupt handler for T16B_DMM Ch2. Counts the values required for frequency calculation.
c17dsadc16AdcEndFlg	A/D conversion end flag variable
c17dsadc16OverWriteFlg	Over write flag variable
t16bCh1IntCapFlg	T16B_DMM Ch1 capture flag variable
frequency	Calculated frequency value variable
meas	Measurement setting structure variable
acPeakMax	Maximum value peak-hold variable
acPeakMin	Minimum value peak-hold variable
t16bint_occurred_Flg	T16B interrupt occurred flag variable
noMeasurementFlg	Unmeasured flag variable

5. Software Operation Description

This software includes functions to control SVT17M03 to realize the DMM function.
The following table shows the number of the Refer to "Figure 4.2-1: Software Operation Conceptual Diagram" for a description of the operations included in the software.

Main software functions:
- Control function for DMM mode operation
- Measurement calculation function
- Serial communication function of measured values

5.1. System Initialization

This process collectively initializes the peripheral circuit functions and interfaces required for DMM operation, including the clock settings for operating the MCU. The following are the setting values for various peripheral circuits in this software, which can be changed to any desired value from c17_init_config.h in the driver. Only the peripheral circuits used in this software are listed.

Table 5.1-1 Peripheral Circuit Setting Values (1)

Peripheral circuit	Setting items	Configurations
CLG	System clock	OSC3
	Frequency	3.2 MHz
DSADC16	Interrupt level	3
	Clock source	OSC3
	Clock frequency division ratio	1:4
LCD4B	Interrupt level	0
	Clock source	OSC1
	Clock frequency division ratio	1:1
PPORT(GPIO)	Interrupt level	0
P07	Input / Output	Output
	Chattering prevention filter	Disable
	Pull-up/down resistor	Disable
P40, P41, P42, P43, P44, P45, P46	Input / Output	Input
	Chattering prevention filter	Disable
	Pull-up/down resistor	Pull-up enable
SNDA	Interrupt level	0
	Clock source	OSC1
	Clock frequency division ratio	1:1
SPI	Interrupt level	0
	Master/Slave	Slave
	Input pin pull-up/down	Disable
	Clock phase/polarity	Mode 3
	Data sequence	MSB first
	Data length	8bit
SVD4	Interrupt level	3
T16 Ch.0	Clock source	OSC1
	Clock frequency division ratio	1:256
	Mode	Repeat mode
	Reload value	127
T16 Ch.1	Clock source	OSC3
	Clock frequency division ratio	1:128
	Mode	Repeat mode
	Reload value	1249
T16 Ch.2	Clock source	OSC1
	Clock frequency division ratio	1:256
	Mode	Repeat mode
	Reload value	127
T16 Ch.3	Clock source	OSC3
	Clock frequency division ratio	1:128
	Mode	Repeat mode
	Reload value	1249

Table 5.1-2 Peripheral Circuit Setting Values (2)

Peripheral circuit	Setting items	Configurations
T16B_DMM Ch.0	Interrupt level	0
	Clock source	OSC1
	Clock frequency division ratio	1:1
	Mode	Repeat mode
T16B_DMM Ch.1	Interrupt level	6
	Clock source	EXCL0
	Clock frequency division ratio	1:1
	Mode	Repeat mode
T16B_DMM Ch.2	Interrupt level	0
	Clock source	OSC1
	Clock frequency division ratio	1:1
	Mode	Repeat mode
UART	Interrupt level	0
	Clock source	OSC3
	Clock frequency division ratio	1:1
	Parity	None
	Data length	8 bit
	Baud rate	230400
	Flow control	None

5.2. Timer Control Interface

This software uses timers throughout the process to generate time, so T16 Ch.0 and T16 Ch.1 are used to generate four timers: "mode name display timer," "timeout timer," "switch polling scan timer," and "LCD display switching timer.

Table 5.2-1 Timer Settings

Name	Peripheral circuit	Period
Mode name display timer	T16 Ch.0	1s
Timeout timer	T16 Ch.0	3s
Switch polling scan timer	T16 Ch.1	50 ms
LCD display switching timer	T16 Ch.1	500 ms
Auto power off switching timer	T16 Ch.2	60s (*) Note: configurable via software
Long-press detection timer	T16 Ch.3	1 s

5.3. Measurement Mode

The DMM controller is activated and the measurement process is performed. The operation in this mode depends on the measurement mode currently set in the DMM controller. The operation flow in this mode is as follows.

Table 5.3-1 Measurement Mode Operations

Measurement mode	Operation types
DC voltage	(1)
AC voltage	(1)
DC current	(1)
AC current	(1)
Resistance (CC-method)	(1)
Resistance (CV-method)	(1)
Continuity check	(1)
Capacitance (CC-method)	(2)
Capacitance (CV-method)	(2)
Diode VF	(1)
AC voltage and frequency	(2)
AC current and frequency	(2)
Internal temperature	(1)

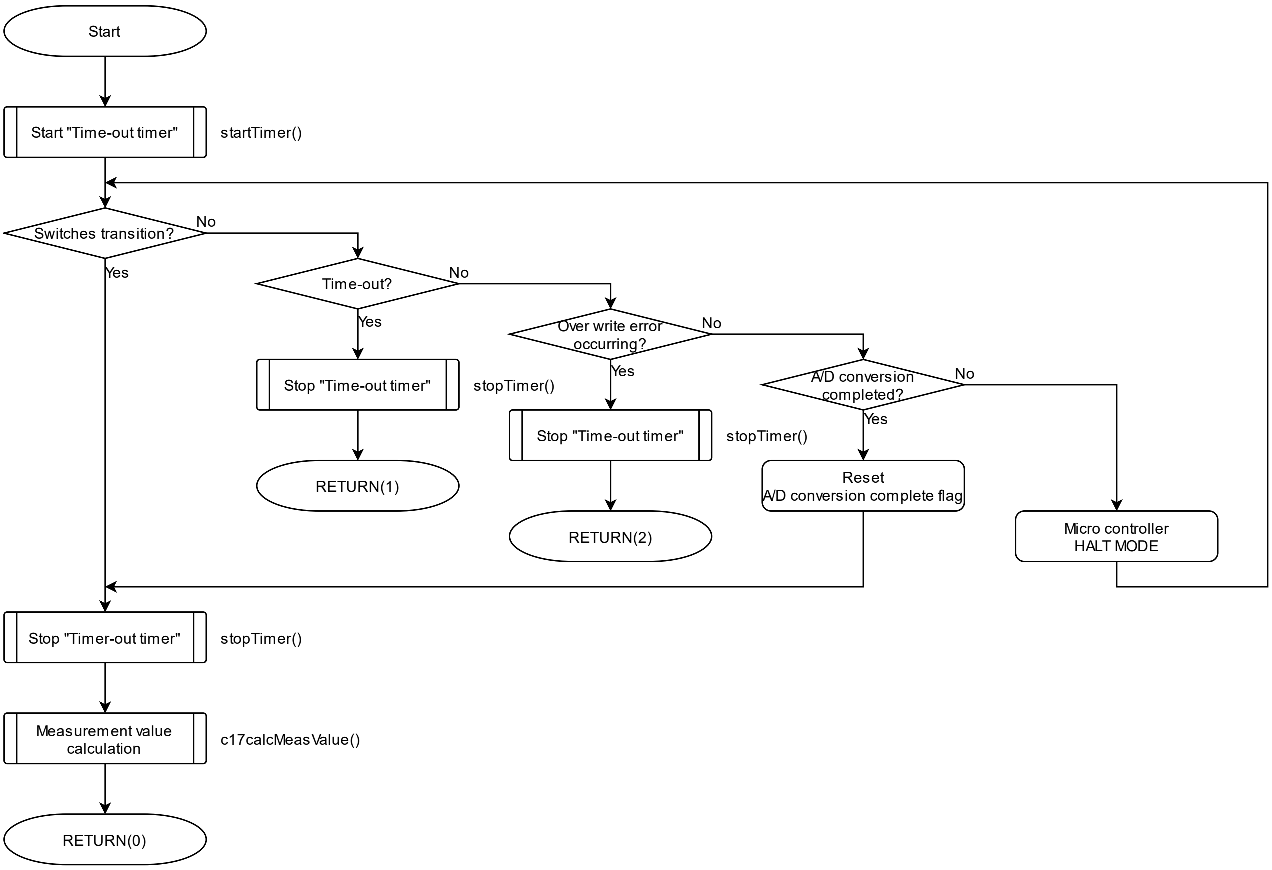


Figure 5.3-1 Measurement Mode Operation (1)

Measurement mode operation, which is classified as (1), mainly processes waiting for the completion of A/D conversion by the DMM controller. Normally, the microcontroller waits in HALT mode and returns to operation when it receives an interrupt for the completion of A/D conversion. A timeout period of 3s is provided, at which time the operation is terminated if the A/D conversion operation is not completed or an overwrite error occurs. In addition, the A/D conversion wait operation is interrupted when a setting change due to a switch transition is detected. However, the switch to be monitored differs for each measurement mode because the response to communication and the response to measurement range switching and peak hold switching differs depending on the measurement mode.

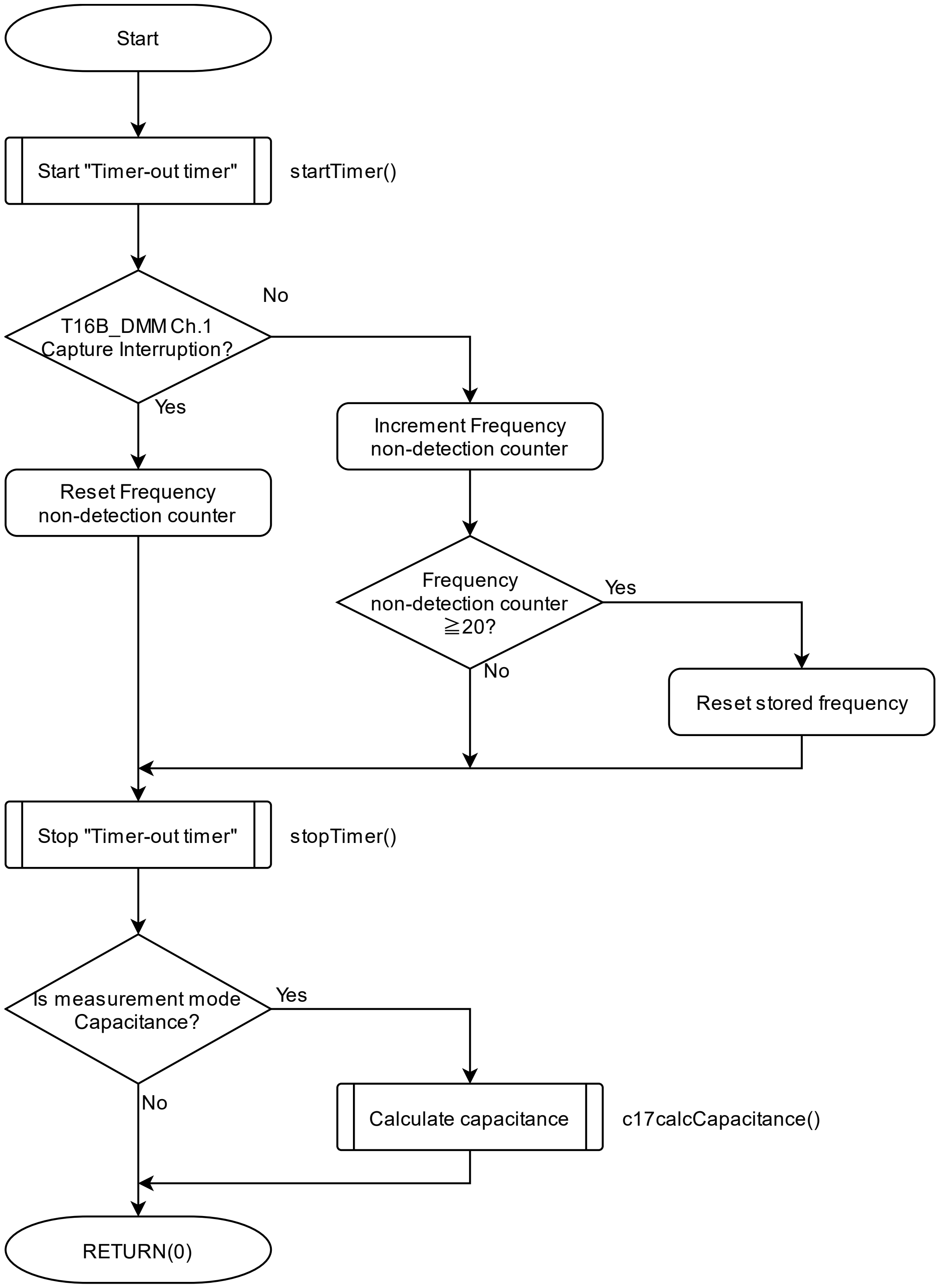


Figure 5.3-2 Measurement Mode Operation (2)

The measurement mode operation, which is classified as (2), mainly performs operations related to the frequency measurement process. The T16B_DMM Ch.1 capture interrupt is linked to the measurement signal input. Therefore, the case where the interrupt does not occur is the case where there is no input signal to be measured. When this state of no input signal occurs 20 times in a row, the calculation frequency stored so far is reset to 0. Finally, if the measurement mode is capacitance, the capacitance value is calculated.

5.3.1. Measurement Mode Switching

When a rotary switch transition is detected at initial startup and during measurement or communication mode, measurement is interrupted if measurement is in progress and the DMM controller is set to the measurement mode corresponding to the rotary switch value.

5.3.2. Measurement Range Switching

When the tactile switch (SW4) is pressed during the measurement mode, the measurement range switching process is performed if the measurement mode being executed corresponds to the measurement range switching. When the measurement mode is capacitance (CC method), the T16B_DMM Ch.0 comparator counter value is different only for the 1000uF range, so the setting is changed here. The measurement range switching process flow in this software is as follows.

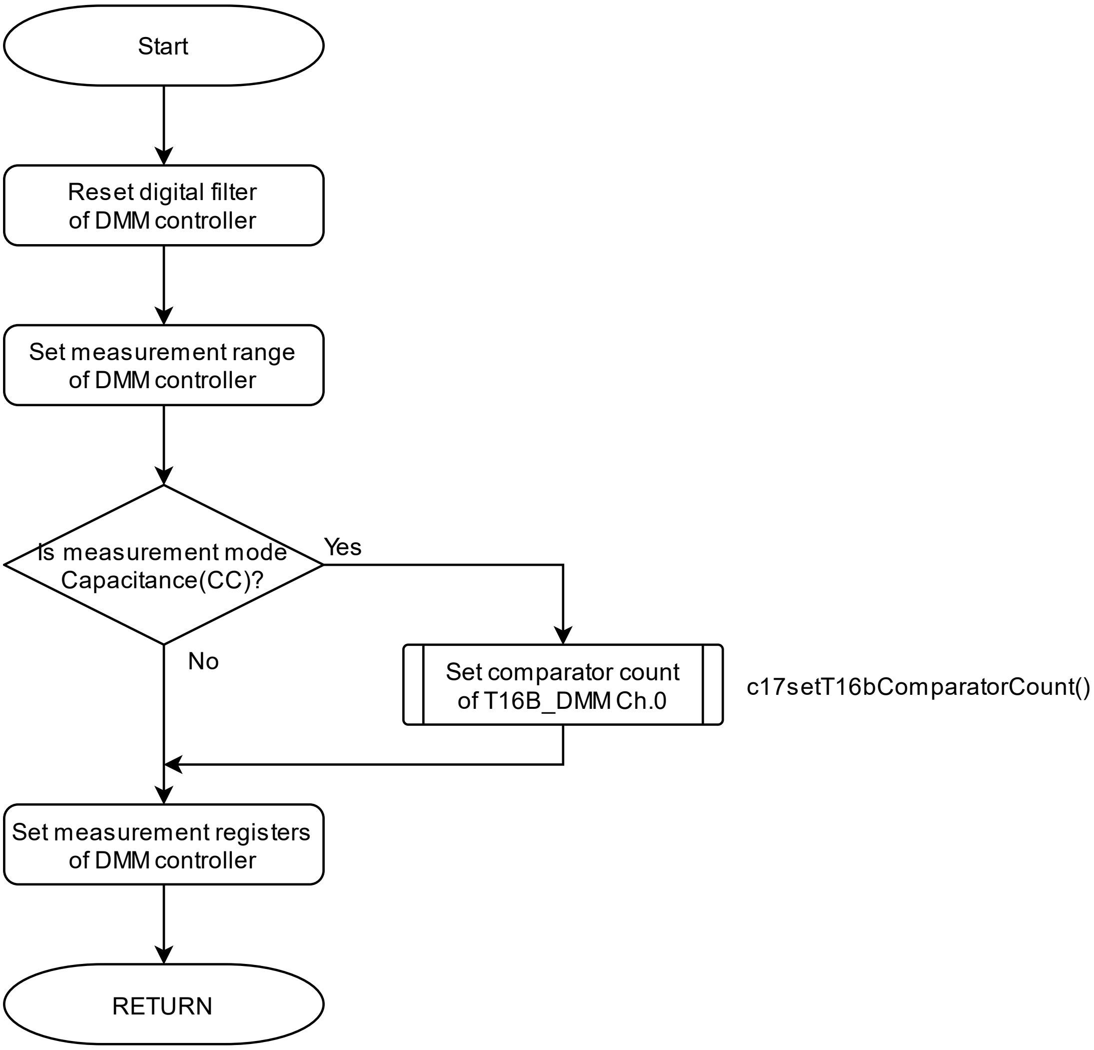


Figure 5.3.2-1 Measurement Range Switching Operation

5.3.3. Peak-Hold Switching

When the tactile switch (SW3) is pressed during the measurement mode, the peak hold switching process is performed if the measurement mode being executed supports the peak hold function. In addition to the peak hold setting of the DMM controller, this section also initializes variables necessary when peak hold is realized by software processing, as described later.

5.3.4. Auto-range Mode Switching

When the tactile switch (SW4) is long-pressed during measurement mode and the active measurement mode supports the auto-range function, the system switches the auto-range function from OFF to ON.
Additionally, if the auto-range function is ON and SW4 is pressed, the system switches the auto-range setting from ON to OFF.

5.3.5. Data-Hold Mode Switching

While in measurement mode, pressing the tactile switch (SW2) while long-pressing SW3 activates the data-hold mode, provided it is currently disabled. When the data-hold mode is enabled, pressing SW2 alone toggles the mode OFF.

5.3.6. Relative Mode Switching

When the data-hold mode is ON, pressing SW4 while long-pressing SW3 activates the relative mode. When the relative mode is already ON, pressing SW4 toggles it OFF.

5.4. Communication Mode

When the tactile switch (SW2) is pressed during the measurement mode, the communication mode is processed if the measurement mode being executed supports data transmission via communication. Once the communication mode is started, the processing is performed according to the following processing flow. In the communication status update process in the flow, the communication status is updated according to the following state transition diagram. At the same time, processing of communication termination by pressing the tactile switch (SW2) during the communication mode is also performed here.

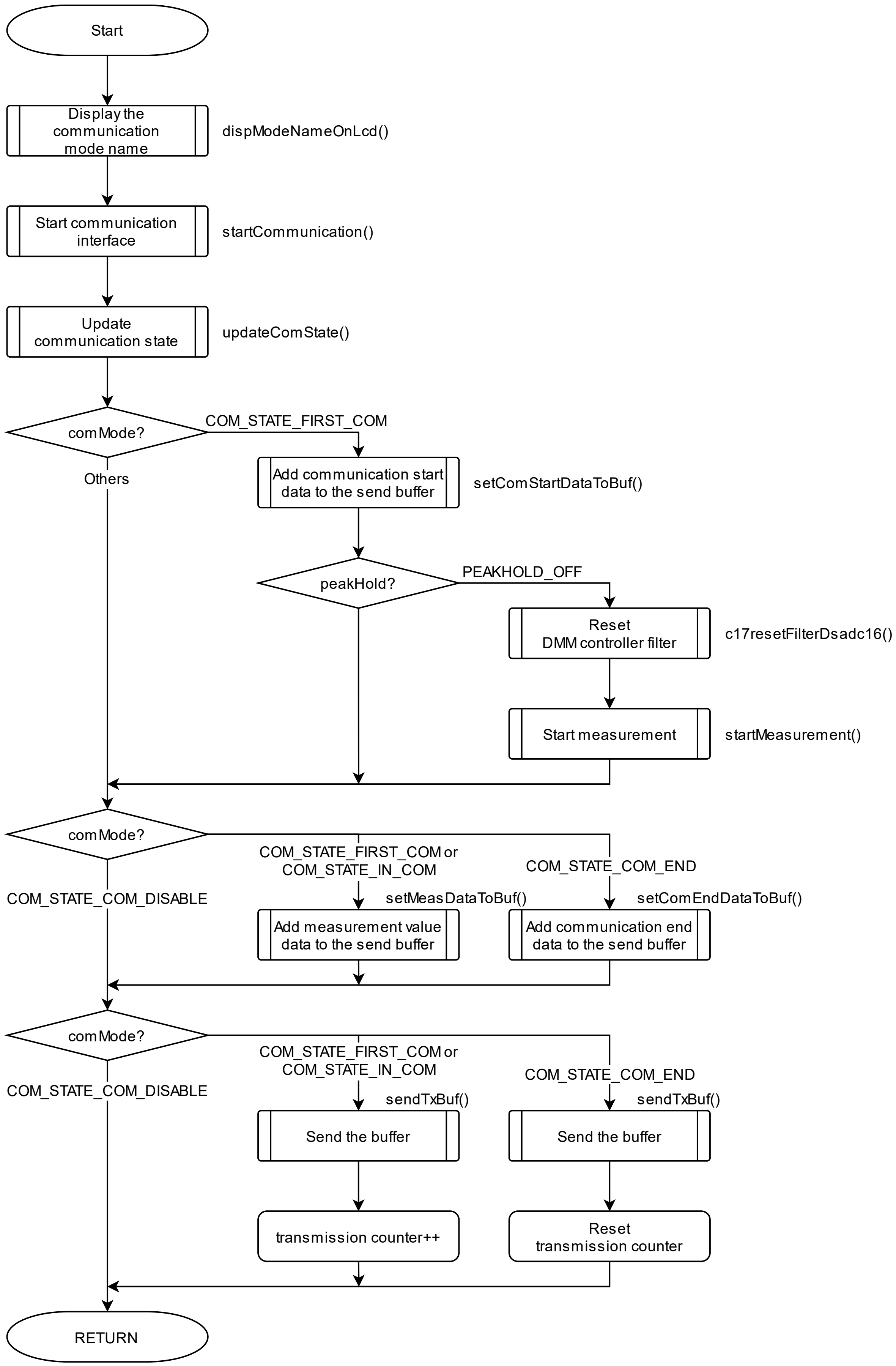


Figure 5.4-1 Communication Mode Operation

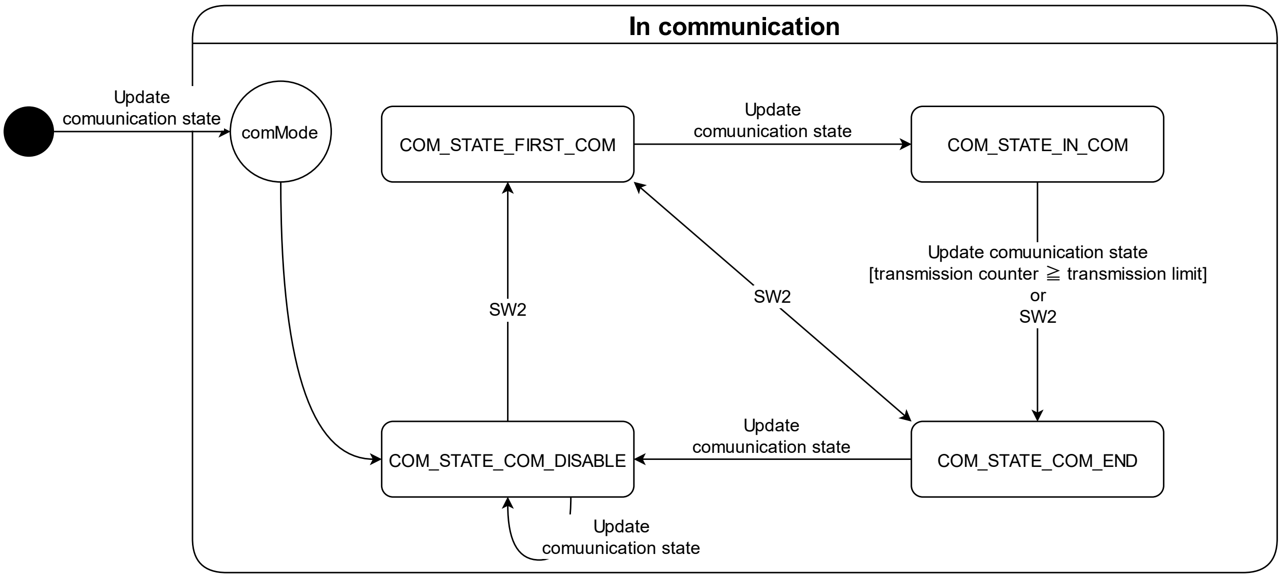


Figure 5.4-2 Communication Mode State Transition

5.5. Calculation Function of Measurement data

The S1C17M03 on SVT17M03 can measure various circuit parameters by properly configuring the DMM controller and 16-bit PWM timer for DMM. This software performs conversion operations to measurement units such as voltage [V], frequency [Hz], and capacitance [F] based on the A/D conversion values and timer count values output from the circuit.

In the measurement mode that can be measured by the DMM controller, the conversion of measurement units is processed in the following three ways. Note that no special calculation is performed in this software to use the linkage function between the buzzer output control of the MCU and the DMM controller in the continuity check mode. Also, no calculation is performed for internal temperature measurement because this software does not derive the measured temperature. The following is an explanation of each conversion method. However, the explanation of “(3) Calculation using frequency” is omitted because this software does not perform any special processing.

Table 5.5-1 Method of Measured Value Calculation

Measurement mode	Conversion method
DC voltage	(1) Calculation from A/D conversion value / (4) Calculation by calibration
AC voltage	(1) Calculation from A/D conversion value /
DC current	(1) Calculation from A/D conversion value
AC current	(1) Calculation from A/D conversion value
Resistance (CC-method)	(1) Calculation from A/D conversion value
Resistance (CV-method)	(1) Calculation from A/D conversion value
Continuity check	None
Capacitance (CC-method)	(3) Frequency-based calculation
Capacitance (CV-method)	(3) Frequency-based calculation
Diode VF	(1) Calculation from A/D conversion value
AC voltage and frequency	(2) Calculation using T16B_DMM PWM Timer
AC current and frequency	(2) Calculation using T16B_DMM PWM Timer
Internal temperature	None

5.5.1. Calculation From A/D Conversion Value

In this method, A/D conversion values acquired by the DMM controller can be converted to measurement units by multiplying the A/D conversion values by numerical values specific to each measurement mode and measurement range. (If calculation by calibration is not used.)
The operation is expressed by the following equation using the conversion coefficient K shown on the next page.

$$Measurement\ value = A/D\ conversion\ value\ \times K \qquad (Eq. 5.5.1.1)$$

Table 5.5.1-1 Method of Measured Value Calculation

Measurement mode	Measurement range	Coefficient K
DC voltage	600 mV	9.1552734e-5
	6 V	9.1635214e-4
	60 V	9.1561799e-3
	600 V	9.1644287e-2
	1000 V	8.4518667e-2
AC voltage	600 mV	9.1552734e-5
	6 V	9.1635214e-4
	60 V	9.1561799e-3
	600 V	9.1644287e-2
	1000 V	8.4518667e-2
DC current	600 uA	8.3665198e-8
	6m A	9.0637298e-7
	60 mA	8.3673482e-6
	600 mA	9.0646272e-5
	6 A	8.4510216e-4
	10 A	9.1552734e-3
AC current	600 uA	8.3665198e-8
	6 mA	9.0637298e-7
	60 mA	8.3673482e-6
	600 mA	9.0646272e-5
	6 A	8.4510216e-4
	10 A	9.1552734e-3
Resistance (CC-method)	600 Ω	5.493164e-2
	6 kΩ	4.577637e-1
	60 kΩ	4.577637
	600 kΩ	4.5771835e1
	6 MΩ	4.5735164e2
	60 MΩ	4.5776367e3
Resistance (CV-method)	600 Ω	5.493164e-2
	6 kΩ	4.577637e-1
	60 kΩ	4.577637
Diode VF	-	2.2888184e-4

5.5.2. Software Processing of Peak-Hold Function

In this method, software processing is used to replace the peak hold function of the DMM controller in some register settings. The register setting conditions and processing flow for software processing of the peak hold function are as follows. After processing, the peak hold function is realized by using the “previous measurement result register value” saved in the flow to calculate the measured value.

Table 5.5.2-1 Conditions for Software Processing of Peak-Hold Function

DSADC16CTL.FUNCSEL	DSADC16CONF.TRUERMS_ON
1（AC voltage）	1
3（AC current）	1

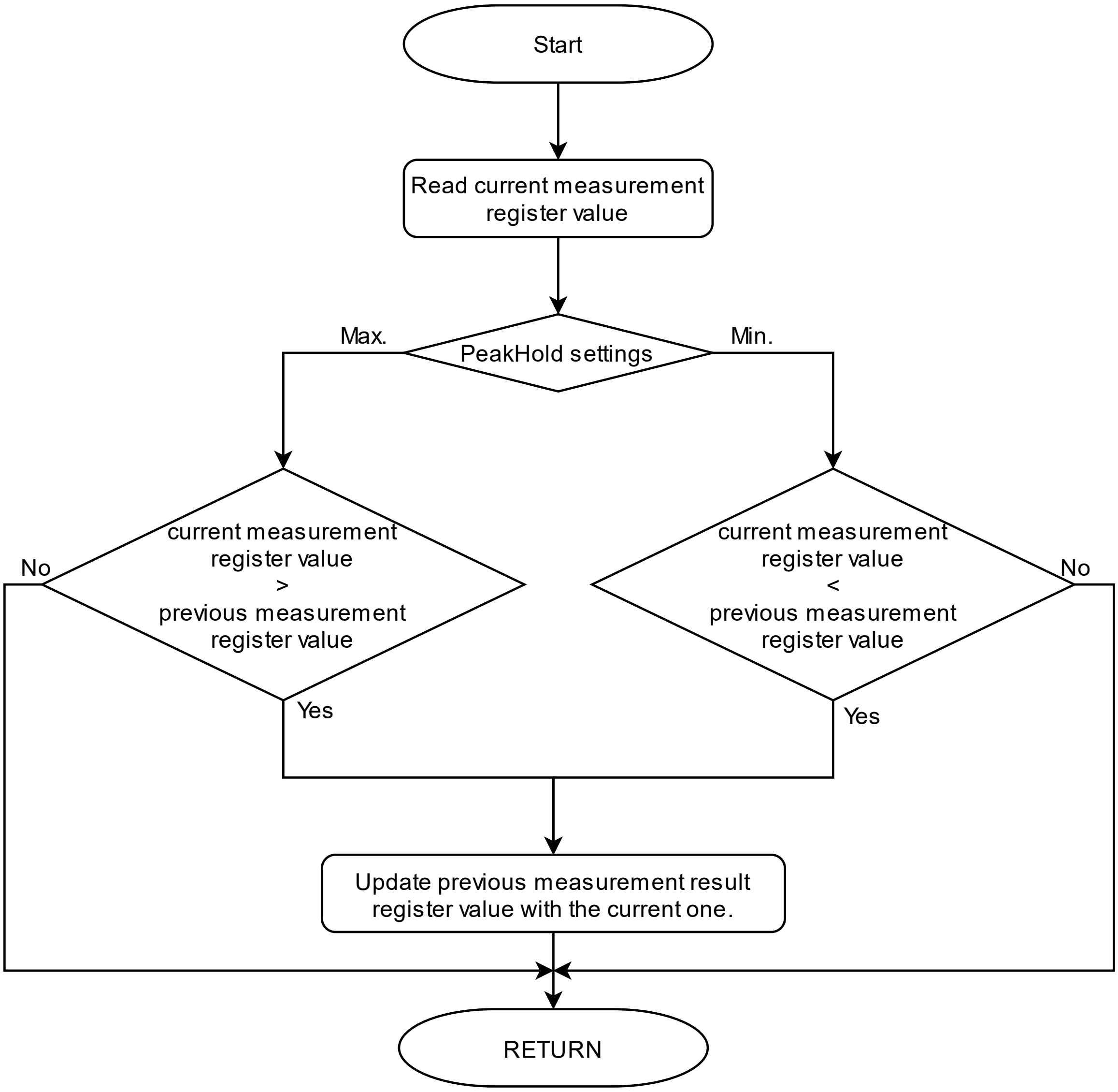


Figure 5.5.2-1 Software Processing of Peak-Hold Function

5.5.3. Calculation Using T16B_DMM PWM Timer

The frequency is measured by measuring the square wave (input pulse to be measured for frequency) output from the DMM controller's WINCMP using three channels of 16-bit PWM timers for DMM and converting it to a value equivalent to the frequency. For details on the role and settings of the PWM timers used for frequency measurement, please refer to the "S1C17M02/M03 Technical Manual".

When the T16B_DMM setting is made as described in the Technical Manual, the frequency can be measured in the configuration shown below. In this software, frequency calculation is handled in the CMPCAP0 interrupt handler of T16B_DMM Ch.1 to enable immediate frequency calculation.

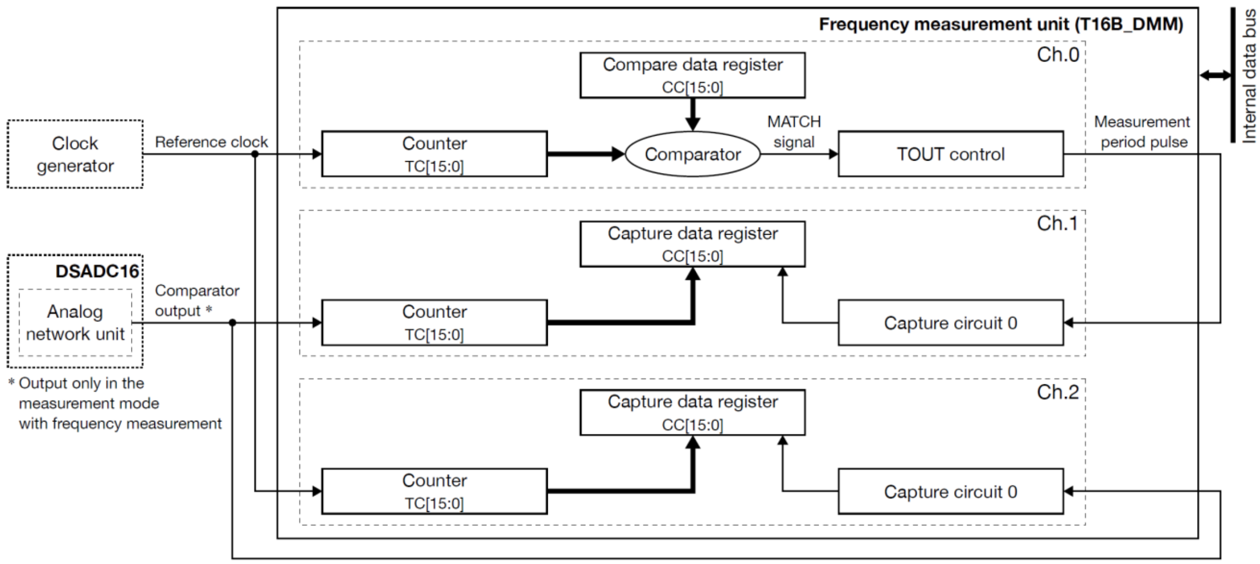


Figure 5.5.3-1 Frequency Measurement Unit Block Diagram (Excerpt from Technical Manual)

5.5.3.1. Frequency Calculation example

Frequency calculation operations are performed as described in the technical manual. As an example, this is the measurement of a 5 Hz frequency input signal.

When a 5-Hz square wave signal is input, the frequency measurement unit captures counts on each channel of the timer as shown below and calculates the frequency in the following steps. Since the count value of each count becomes 0 when it overflows, this software uses the CNTMAX interrupt in Ch.1 and Ch.2 to count the number of times it overflows to calculate the cumulative count capture value during the measurement period.

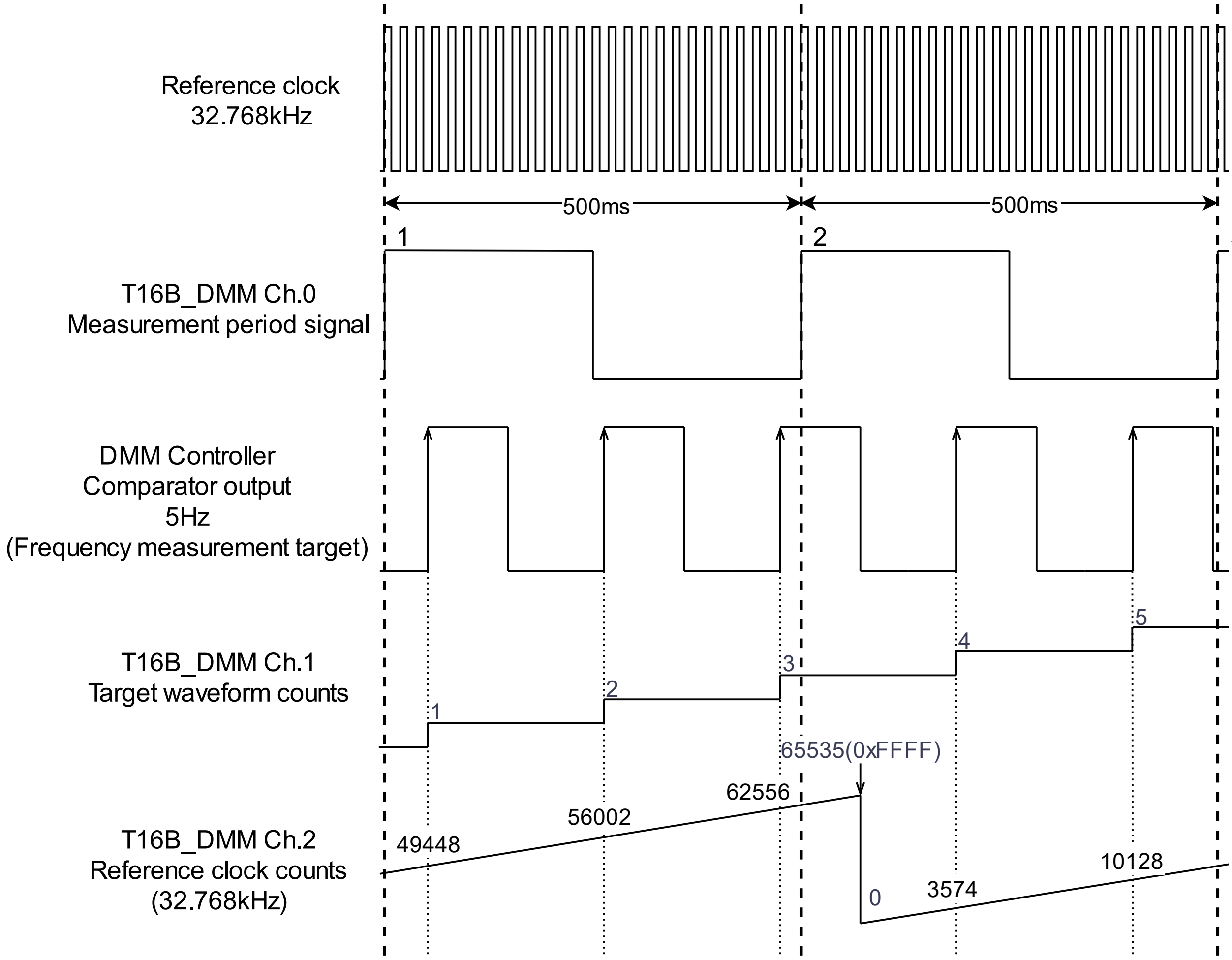


Figure 5.5.3.1-1 Capture Value of T16B_DMM Each Channel

- **Frequency calculation equation**

  T16B_DMM Ch.1 1st capture value: n_{IN1}
  T16B_DMM Ch.1 2nd capture value: n_{IN2}
  T16B_DMM Ch.2 1st capture value: n_{E1}
  T16B_DMM Ch.2 2nd capture value: n_{E2}
  Reference clock frequency (32,768Hz): f_R
  Measurement period ($2 \times T16B0CCR0$ register value): n_R

  When $n_{IN1} - n_{IN2} \geq 4000$:

  $$F[Hz] = \frac{f_R \times (n_{IN2} - n_{IN1})}{n_R} \qquad (Technical\ Manual\ Eq. 14.1)$$

  When $n_{IN1} - n_{IN2} < 4000$:

  $$F[Hz] = \frac{f_R \times (n_{IN2} - n_{IN1})}{n_{E2} - n_{E1}} \qquad (Technical\ Manual\ Eq. 14.2)$$

  1. At the second rise of the measurement period signal of T16B_DMM Ch.0, that is, when the T16B_DMM Ch.1 CMPCAP0 interrupt occurs, the T16B_DMM Ch.1 measurement waveform count capture value ($n_{IN1} = 3$) and T16B_DMM Ch.2 reference clock count capture value ($n_{E1} = 62556$) is saved.
  2. At the rise of the next T16B_DMM Ch.0 measurement period signal (the third time), similarly, save the T16B_DMM Ch.1 measurement waveform count capture value ($n_{IN1} = 5$) and the T16B_DMM Ch.2 reference clock count capture value n_{E2}. Now, the capture value of T16B_DMM Ch.2 is $n_{E2} = 10128 + 65536 \times 1 = 75664$ because it has gone through count overflow once.
  3. Calculate the difference between n_{IN}
     $n_{IN2} - n_{IN1} = 5 - 3 = 2$
  4. Since $n_{IN2} - n_{IN1} = 2 < 4000$, the frequency is calculated according to the Technical Manual Eq.14.2.
     $$F[Hz] = \frac{f_R \times (n_{IN2} - n_{IN1})}{n_{E2} - n_{E1}} = \frac{32768 \times (5 - 3)}{75664 - 62556} \approx 4.999695$$

5.5.4. Calculation by Calibration

In this software, calibration is performed using linear interpolation to correct the measured values obtained. The calibration function is always enabled, and all measurement values displayed on the LCD are corrected values.

The following table lists the measurement modes and measurement ranges that are subject to calibration.

Table 5.5.4-1 Measurement Modes and Ranges Subject to Calibration

Measurement mode	Measurement range
DC voltage	All ranges
AC voltage	All ranges
DC current	All ranges
AC current	All ranges
Resistance (CC-method)	All ranges
Resistance (CV-method)	All ranges
Capacitance (CC-method)	All ranges
Capacitance (CV-method)	All ranges

In this software, calibration requires prior measurement of two reference points (Reference Point_High and Reference Point_Low) for each measurement mode and range. The following outlines the preparatory steps necessary for performing calibration.

Items to Prepare:
- SVT17M03
- PC
- Calibrator
- DMM_Calibration_Sheet

Reference Value Measurement Procedure for Calibration

1. Connect the PC, SVT17M03, and the calibrator. (See the diagram below.)

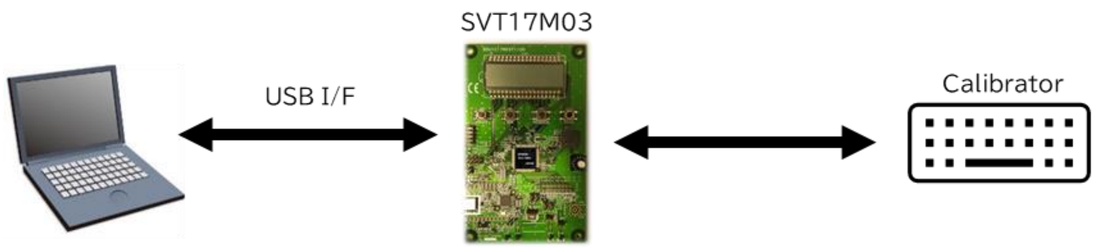


Figure 5.5.4-1 Connection Diagram for Reference Value Measurement for Calibration

2. Output an arbitrary reference value from the calibrator, and measure that value using the SVT17M03. For each measurement mode and range, perform measurements at two points: Reference Point_High and Reference Point_Low.
3. Run DmmEvalTool.exe on the PC to obtain the A/D conversion values of the reference points measured by the SVT17M03. (For capacitance measurements, obtain the capacitance values instead of A/D conversion values.)
4. Enter the input values from the calibrator and the obtained A/D conversion values into the “table” sheet of the DMM_Calibration_Sheet. (For capacitance measurements, enter the input values from the calibrator and the capacitance values obtained from DmmEvalTool.)

1	DC voltage				
2		Calibrator		SVT17M03	
3		Reference Input	Reference Value[V]	Reference ADC Value	Example Value
4					6640
5					-6704
6					6614
7					-6678
8					6630
9					-6696
10					6561
11					-6635
12					10323
13					-10563
14					

Figure 5.5.4-2 The “table”sheet in the DMM_Calibration_Sheet

5. Replace the #define statements in the relevant section (lines 159 to 447) of the reference software file c17dsadc.h with the #define statements generated in the “header” sheet of the DMM_Calibration_Sheet.

1	/// Reference values for calibration		
2	///DCV		
3	#define CALIB_MEAS_REF_DCV_600MV_HIGH	0	///< Reference Value of DCV_+600mV
4	#define CALIB_MEAS_REF_DCV_600MV_LOW	0	///< Reference Value of DCV_-600mV
5	#define CALIB_MEAS_REF_DCV_6V_HIGH	0	///< Reference Value of DCV_+6V
6	#define CALIB_MEAS_REF_DCV_6V_LOW	0	///< Reference Value of DCV_-6V
7	#define CALIB_MEAS_REF_DCV_60V_HIGH	0	///< Reference Value of DCV_+60V
8	#define CALIB_MEAS_REF_DCV_60V_LOW	0	///< Reference Value of DCV_-60V
9	#define CALIB_MEAS_REF_DCV_600V_HIGH	0	///< Reference Value of ACV_+6006V
10	#define CALIB_MEAS_REF_DCV_600V_LOW	0	///< Reference Value of DCV_-600V
11	#define CALIB_MEAS_REF_DCV_1000V_HIGH	0	///< Reference Value of ACV_+1000V
12	#define CALIB_MEAS_REF_DCV_1000V_LOW	0	///< Reference Value of DCV_-1000V
13			
14	///ACV		
15	#define CALIB_MEAS_REF_ACV_600MV_HIGH	0.424264069	///< Reference Value of ACV_600mV
16	#define CALIB_MEAS_REF_ACV_600MV_LOW	0	///< Reference Value of ACV_0Ohm(@600mV)
17	#define CALIB_MEAS_REF_ACV_6V_HIGH	4.242640687	///< Reference Value of ACV_6V
18	#define CALIB_MEAS_REF_ACV_6V_LOW	0	///< Reference Value of ACV_0Ohm(@6V)
19	#define CALIB_MEAS_REF_ACV_60V_HIGH	42.42640687	///< Reference Value of ACV_60V
20	#define CALIB_MEAS_REF_ACV_60V_LOW	0	///< Reference Value of ACV_0Ohm(@60V)

readme | table | header | calculation

Figure 5.5.4-3 The “header” Sheet in the DMM_Calibration_Sheet

6. Rebuild the project using GNU17, then flash the software to the SVT17M03.

When performing calibration using linear interpolation, use the following formula.

(Example) DCV 6V Range

Reference Point_High (Measured Value, A/D Conversion Value): (6.0V, 6640)
Reference Point_Low (Measured Value, A/D Conversion Value): (-6.0V, -6704)

$$\begin{aligned}
&Corrected\ Voltage\ Value\ =\ Measured\ Value\ at\ Reference\ Point_Low\ + \\
&(Measured\ Value\ at\ Reference\ Point_High\ -\ Measured\ Value\ at\ Reference\ Point_Low)\ \times \\
&(Actual\ A/D\ Conversion\ Value\ -\ A/D\ Conversion\ Value\ at\ Reference\ Point_Low)\ / \\
&(A/D\ Conversion\ Value\ at\ Reference\ Point_High\ -\ A/D\ Conversion\ Value\ at\ Reference\ Point_Low) \\
&\qquad = -6.0 + \{6640 - (-6704)\} * \{Actual\ A/D\ Conversion\ Value - (-6704)\}\ /\{6 - (-6)\}
\end{aligned}$$

Appendix A Switching Procedure from S1C17M03 to S1C17M02

The application and middleware software included in this software package are designed to operate the SVT17M03, which is an evaluation board equipped with the S1C17M03. Therefore, if the target is to operate on the S1C17M02, the software cannot be used as-is.This chapter outlines important considerations when reusing the measurement-related components of this software package in your own environment.

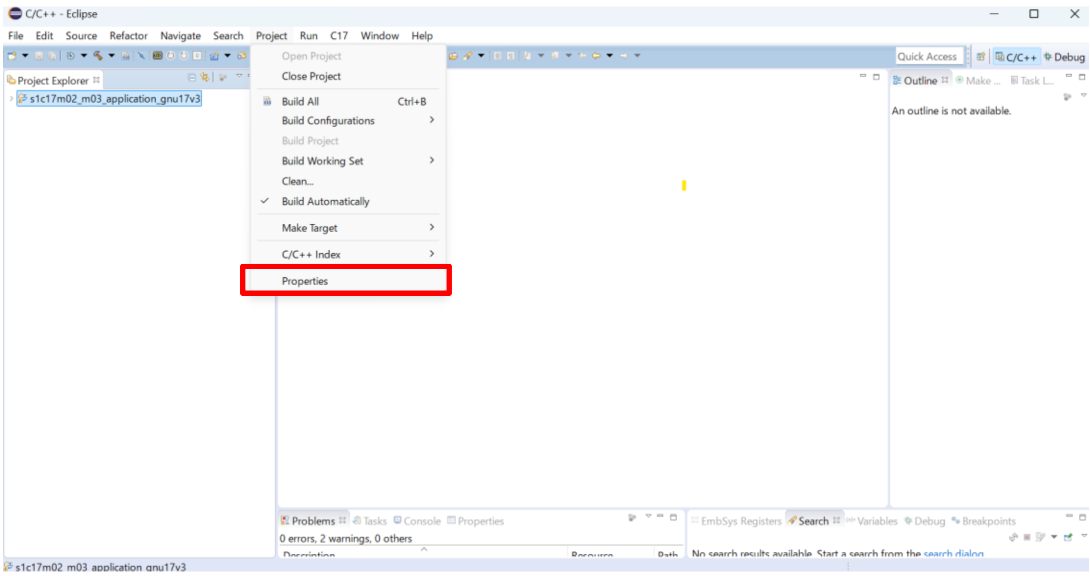
The following program files are assumed to be the software components intended for reuse. Please modify them, as necessary.

Table 0-1 List of Program Files Requiring Modification

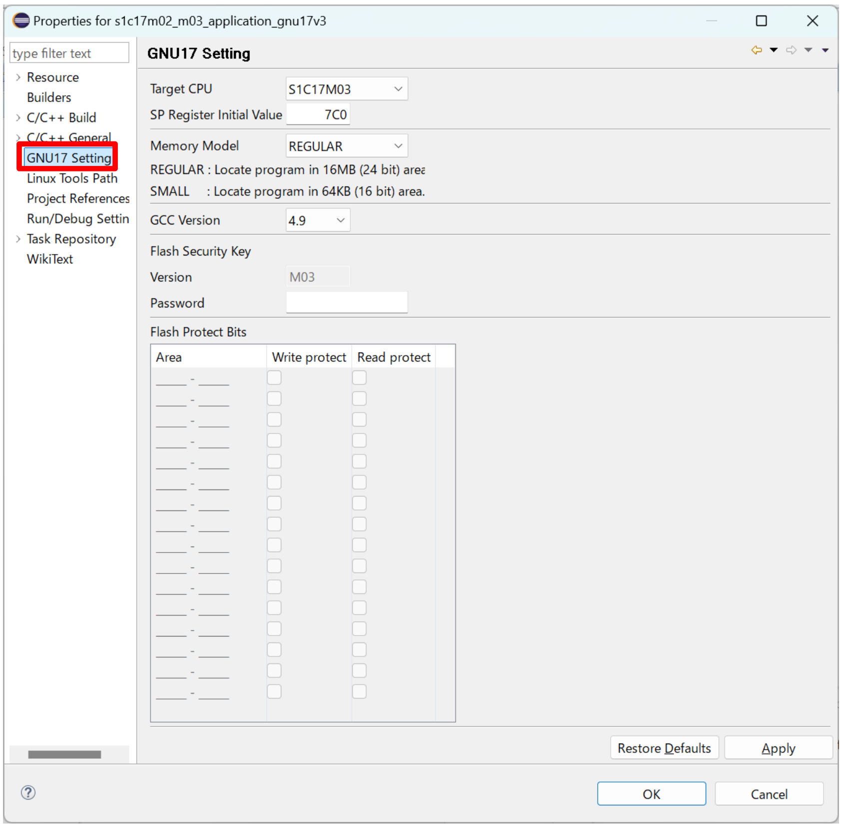
Program files	Description
inc/c17_mcu_select.h	Header file for MCU selection
main_measurement.c	Program file for measurement modes

Setting up the Development Environement (GNU17v3 IDE)
To develop S1C17M02 using GNU17v3 IDE, project configuration is required. Please follow the steps below to configure the project.

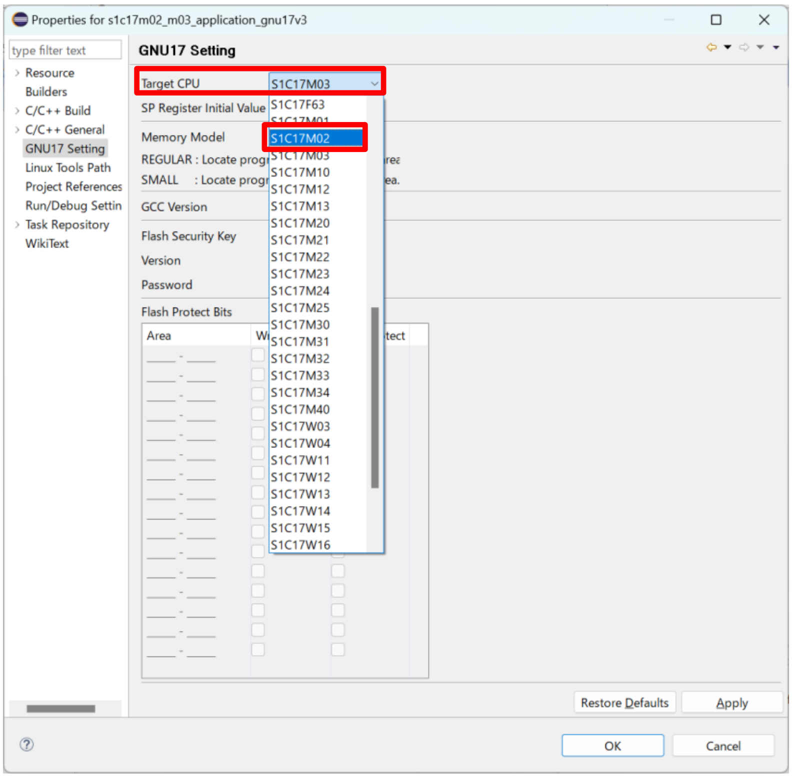
1. Open the project, then from the menu, select “Project”, followed by “Properties”.



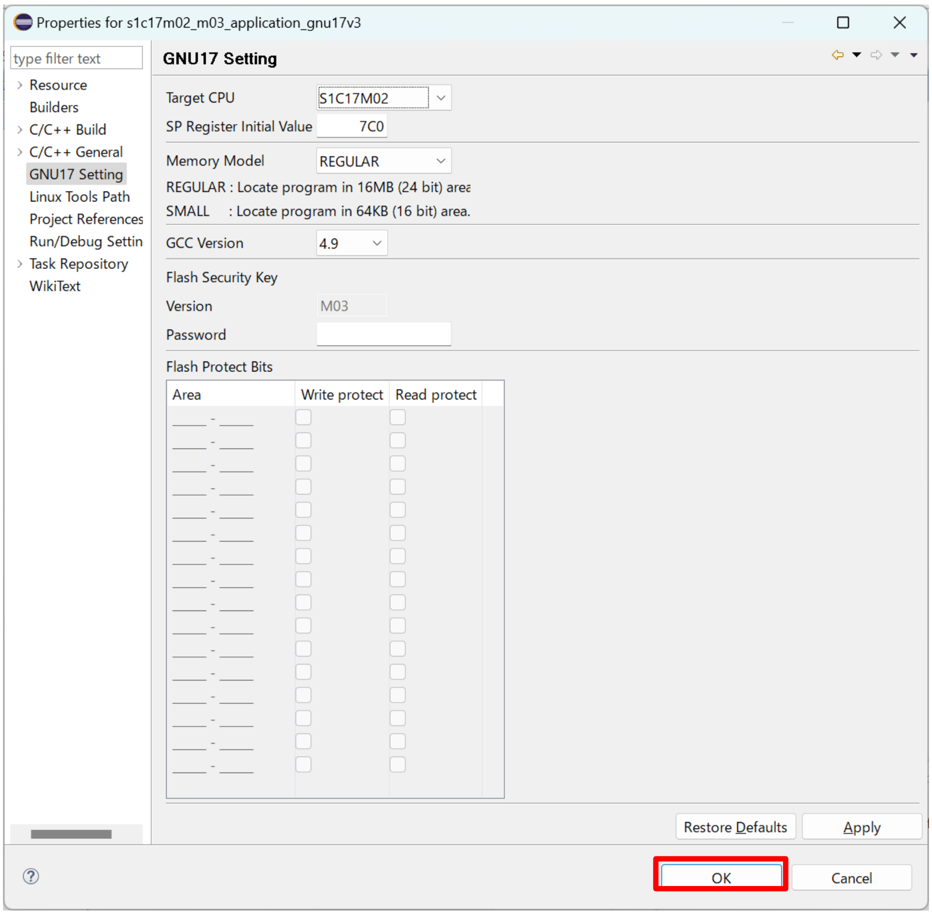
2. Select “GNU17 Setting”.



3. From the drop-down list under the “Target CPU” section, select “S1C17M02”.



4. Click the "OK" button.



c17_mcu_select.h

As previously mentioned, this software package is specifically designed to operate the SVT17M03. Therefore, the initial configuration is set up to run on the S1C17M03.

If you intend to use it with the S1C17M02, please disable the macro definition C17_MCUSEL_M03 and enable C17_MCUSEL_M02 in inc/c17_mcu_select.h, as described in section "4.3.4 Setting the Target MCU".

main_measurement.c

This file is an application program that enables DMM operation using various driver programs.
By utilizing the functions provided in this program, DMM operation can be implemented easily.

However, due to certain design constraints related to the programs used in this file, there are specific points that must be considered when reusing it in your own environment. These considerations are outlined below.

When C17_MCUSEL_M02 is enabled in inc/c17_mcu_select.h, the functions within main_measurement.c listed in "Table A-1: List of Program Files Requiring Modification" become unavailable by default.
This is an intentional measure to prevent errors caused by hardware differences between the S1C17M02 and S1C17M03.
Therefore, in order to use the functions in this file, several modifications are required.

Function: runMeasMode()

This function initiates measurement using DSADC16 and T16B_DMM, and processes the measurement results after completion (for detailed operation, refer to “Figure 5.3-1: Measurement Mode Operation (1))”.

Within the operation of this function, there are hardware-dependent sections, as illustrated in the following diagram.

```
switch(meas.mode)
{
    case MEASMODE_DCV:
    case MEASMODE_ACV:
    case MEASMODE_DCI:
    case MEASMODE_ACI:
    case MEASMODE_OHMCC:
    case MEASMODE_OHMCV:
    case MEASMODE_DIODE:
    case MEASMODE_TEMP:
        /// Wait until the AD conversion is complete.
        while(1) {
            /// Polling monitoring of switch inputs
            if(swPollingFlg == 1) {
                updateSwitchState();
                swPollingFlg = 0;                      (1)                    (2)
            }

            /// Break if there is a switch input transition
            if( (transition.swMeasModeFlg == 1) || (transition.swComModeFlg == 1) || (transition.swMeasRangeFlg ==
                break;
            }

            if(timeOutFlg == 1) {           ///< If timeout, exit this function
                stopTimer(TIMER_TIMEOUT);
                return EXIT_TIMEOUT;
            }

            if(c17dsadc16OverWriteFlg == 1) {   /// Exit this function if Over write error has occurred.
                stopTimer(TIMER_TIMEOUT);
                return EXIT_OVER_WRITE;
            }

            if(c17dsadc16AdcEndFlg == 1) {      ///< Break if the AD conversion is complete
                c17dsadc16AdcEndFlg = 0;
                break;
            }
            asm("halt");
        }
        stopTimer(TIMER_TIMEOUT);

        result.f = c17calcMeasValue(meas.mode, meas.range, meas.peakHold);
        break;
```

Figure A-1 Hardware-Dependent Sections in the runMeasMode Function

Section (1)

The process indicated by (1) in “Figure A-1” involves polling to update the input status of GPIOs connected to the tactile and rotary switches implemented on the SVT17M03.

In this software package, I/O ports P40 to P46 are used. Therefore, depending on your environment, it may be necessary to modify or remove this section.

For details, please refer to the functions getTactileSwitchInput and getRotarySwitchInput in mid_switch.c.

Section (2)

The section indicated by (2) in “Figure A-1” is where the loop is exited when an input operation is performed using a tactile switch or rotary switch.

The detection of whether an input operation has occurred uses a transition structure variable defined in mid_switch.c, so it may need to be edited (or removed) depending on your environment.

For details, please refer to the updateSwitchState function within the same program.

Revision History

Attachment-1

Rev. No.	Date	Page	Category	Contents
Rev 1.0	2022/3/17	All	new	new
Rev. 1.1	2022/6/27	18	revise	Revise Figure 4.3.4-1
		38-42	add	Add Appendix A.
Rev. 1.2	2025/6/30	16 20-22 25	revise	Revise Operation Conceptual Diagram Revise Software Modules Description Revise Timer Control Interface
		5-8 12-13 28 36-37	add	Add descriptions of additional functions in Operation Description Add Calibration by Calibration in Calculation Function of Measurement Data

EPSON

International Sales Operations

America

Epson America, Inc.

Headquarter:
3131 Katella Ave.
Los Alamitos, CA 90720, USA
Phone: +1-800-463-7766

San Jose Office:
2860 Zanker Road Suite 204
San Jose, CA 95134, USA
Phone: +1-800-463-7766

Europe

Epson Europe Electronics GmbH
Riesstrasse 15, 80992 Munich,
Germany
Phone: +49-89-14005-0

Asia

Epson (China) Co., Ltd.
4th Floor, Tower 1 of China Central Place, 81 Jianguo Road, Chaoyang District, Beijing 100025 China
Phone: +86-10-8522-1199

Shanghai Branch
Room 601-603, Building A One East, No.325 East Longhua Road, Shanghai 200023, China
Phone: +86-21-5330-4888

Shenzhen Branch
Room 603/604, 6 Floor, Tower 7, One Shenzhen Bay,
No.3008 Center Rd, Shenzhen 518054, China
Phone: +86-755-3299-0588

Epson Taiwan Technology & Trading Ltd.
15F, No.100, Songren Rd, Sinyi Dist, Taipei City 110. Taiwan
Phone: +886-2-8786-6688

Epson Singapore Pte., Ltd.
438B Alexandra Road,
Block B Alexandra TechnoPark, #04-01/04, Singapore 119968
Phone: +65-6586-5500

Epson Korea Co.,Ltd
10F Posco Tower Yeoksam, Teheranro 134 Gangnam-gu,
Seoul, 06235, Korea
Phone: +82-2-3420-6695

Seiko Epson Corp.
Sales & Marketing Division

MD Sales & Marketing Department
421-8 Hino, Hino-shi, Tokyo 191-8501, Japan

Document Code: 414282301
First Issue March 2022
Revised July 2025