R32C/100 Series

Remote Control Signal Reception Using the Intelligent I/O

R01AN0954EJ0100
Rev. 1.00
July 31, 2013

Abstract

This document describes receiving signals from a remote control using the time measurement function of the intelligent I/O in the R32C/118 Group.

The time measurement function of the intelligent I/O in the R32C/118 Group can use up to 16 channels - channel 0 to channel 7 in group 0, and channel 0 to channel 7 in group 1. The document uses channel 0 in group 1. When using a channel other than channel 0 in group 1, refer to the User's Manual: Hardware and modify the registers associated with the channel and group used.

Products

R32C/116 Group
R32C/117 Group
R32C/118 Group

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Specifications

An infrared signal is transmitted from a remote control, and the remote control receiver converts the infrared signal to an electrical signal (remote control waveform). The waveform received from the remote control is recognized as transmit data after the remote control waveform pulse width is measured by the time measurement function of the intelligent I/O.

Table 1.1 lists the Peripheral Functions and Their Applications. Figure 1.1 shows the Outline Block Diagram of Remote Control Signal Reception.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
Time measurement function of the intelligent I/O (channel 0 in group 1)	Remote control waveform pulse width measurement
Timer A0	1 frame timer measurement for a timer A1 event count
Timer A1	1 frame timer measurement

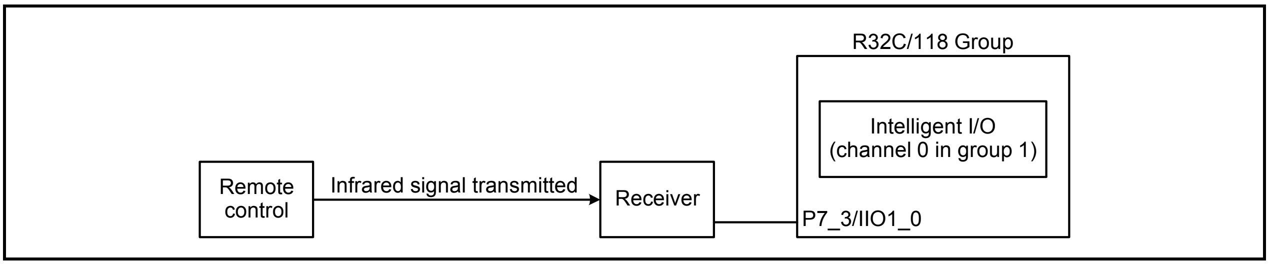


Figure 1.1 Outline Block Diagram of Remote Control Signal Reception

2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item	Contents
MCU used	R5F64189DFD (R32C/118 Group)
Operating frequencies	• XIN clock: 16 MHz • PLL clock: 100 MHz • Base clock: 50 MHz • CPU clock: 50 MHz • Peripheral bus clock: 25 MHz • Peripheral clock: 25 MHz
Operating voltage	5 V
Integrated development environment	Renesas Electronics Corporation High-performance Embedded Workshop Version 4.09
C compiler	Renesas Electronics Corporation R32C/100 Series C Compiler V.1.02 Release 01
	Compile options -D__STACKSIZE__=0X300 -D__ISTACKSIZE__=0X300 -DVECTOR_ADR=0x0FFFFFBDC -c -finfo -dir "$(CONFIGDIR)" The default setting is used in the integrated development environment.
Operating mode	Single-chip mode
Sample code version	1.00
Board used	Renesas Starter Kit for R32C/118 (device part no.: R0K564189S000BE)

3. Reference Application Notes

Application notes associated with this application note are listed below. Refer to these application notes for additional information.

- R32C/100 Series Configuring PLL Mode (REJ05B1221)
- R32C/100 Series Pulse-Width Measurement Using the Time Measurement Function of Intelligent I/O Groups 0 and 1 (R01AN0096EJ)

4. Peripheral Functions

4.1 Overview of the Time Measurement Function

The time measurement function of the intelligent I/O synchronizes with the external trigger input and stores the base timer value to the GiTMj register (i = 0, 1; j = 0 to 7). Figure 4.1 shows an Example of Measuring the Pulse Width of a Remote Control Waveform Using the Time Measurement Function.

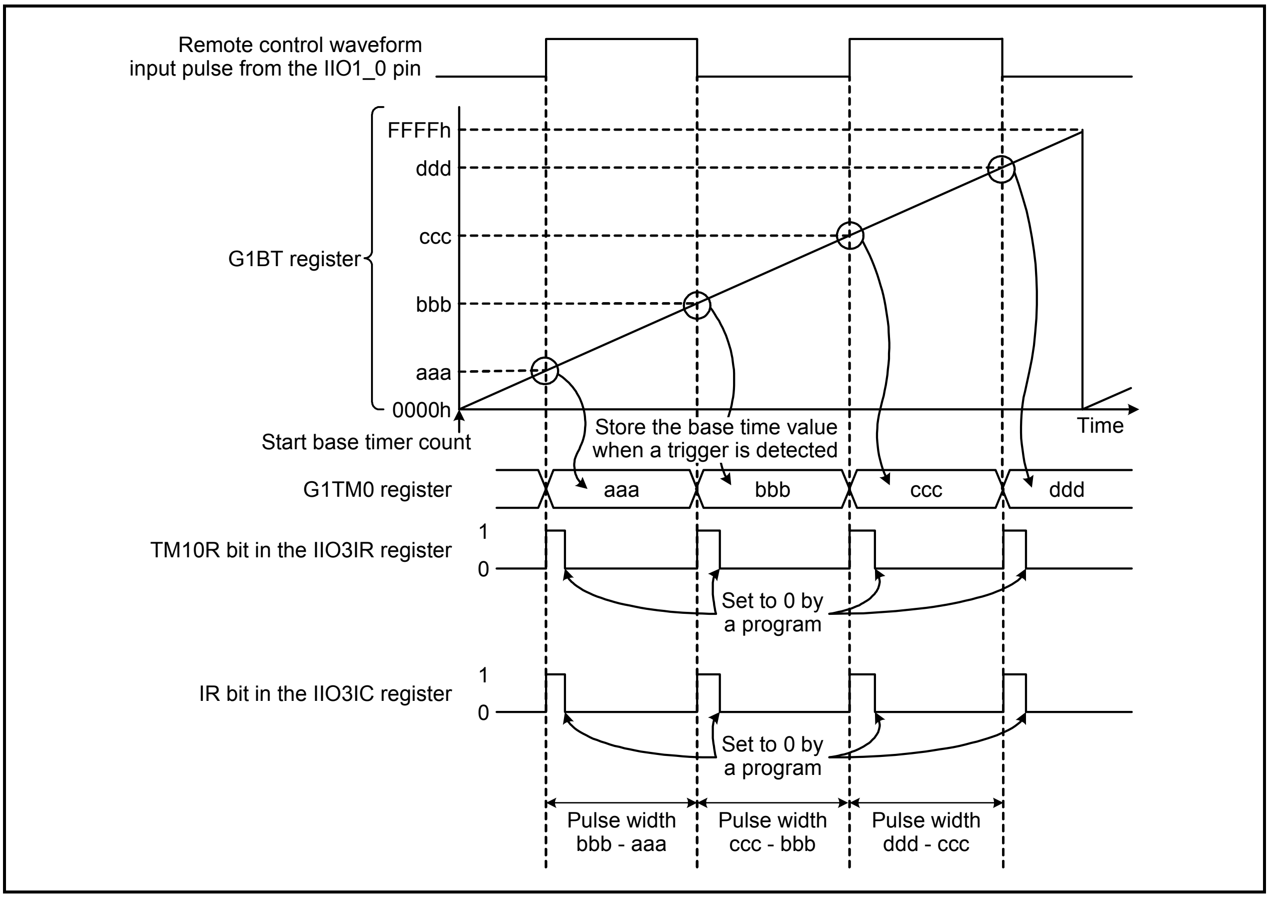


Figure 4.1 Example of Measuring the Pulse Width of a Remote Control Waveform Using the Time Measurement Function

5. Hardware

5.1 Pins Used

Table 5.1 lists the Pin Used and Its Function.

Table 5.1 Pin Used and Its Function

Pin Name	I/O	Function
P7_3/IIO1_0	Input	Input remote control waveforms

6. Software

6.1 Operation Overview

The sample code accompanying this document uses the time measurement function (channel 0 in group 1) of the intelligent I/O to measure the pulse width of a remote control waveform input from the remote control receiver. Also, timer A0 and timer A1 are used to measure one frame from the leader code of the input remote control waveform.

6.1.1 Intelligent I/O

Table 6.1 lists the Settings for the Time Measurement Function (Channel 0 in Group 1) of the Intelligent I/O.

Table 6.1 Settings for the Time Measurement Function (Channel 0 in Group 1) of the Intelligent I/O

Item	Setting
Count source	f1
Count source divide ratio	Divided by 50
Base timer reset source	Not used
Increment/decrement control	Increment mode
Time measurement trigger	Both edges
Digital filter	Not used
IIO1_0 input pin	P7_3 used
Base timer interrupt	Not used
Intelligent I/O group 1 time measurement function channel 0 interrupt	Not used

6.1.2 Timers

Set timer A0 to timer mode, and set timer A1 to event counter mode (count the number of timer A0 underflows). Table 6.2 lists the Timer A0 Settings and Table 6.3 lists Timer A1 Settings.

Table 6.2 Timer A0 Settings

Item	Setting
Operating mode	Timer mode
Count source	f1
Gate function	Gate function not used
Count setting value	25000 - 1 (1 ms)
Timer A0 interrupt	Not used

Table 6.3 Timer A1 Settings

Item	Setting
Operating mode	Event counter mode
Count operation type	Reloading
Increment/decrement select	Decrement
Timer A1 event/trigger select	Overflow or underflow of timer A0
Count setting value	140 - 1 (140 × 1 ms [timer A0 underflow] = 140 ms)
Timer A1 interrupt	Not used

6.1.3 Remote Control Data Detection Specifications

Remote control data transmitted to the receiver is processed according to the following specifications:

- For the first data, reception is determined to be complete when one frame (i.e. the interval from the leader code through the frame space) is received within 108 ms. One frame here is the leader code, custom code (8 bits), inverted custom code (8 bits), data code (8 bits), inverted data code (8 bits), stop bit (1 bit), and the frame space (interval where there is no infrared transmission).
- For the second and subsequent data, reception is determined to be complete when one frame is received within 108 ms. Here, one frame of the frame space is the leader code and a stop bit (1 bit).
- For each code, code recognition is determined to be complete if the error is within ±30% of the remote control data format value. The same applies to one frame is within 108 ms +30%.
- When the leader code is detected, detection takes place in the order of custom code, data code, stop bit, and frame space.
- If a receive error occurs on each code, the next rising or falling edge is determined as the leader code (first data) and reception starts.
- When one frame (including the +30% error) or more has elapsed after the leader code, if the frame space is being recognized in the received data, reception is determined to be complete.
- When the leader code after the frame space is detected within one frame (including the +30% error), the detected received data is recognized as the second or subsequent data. (The first leader code may be received within one frame after the frame space as the +30% error is included.)

Figure 6.1 shows the Remote Control Data Format.

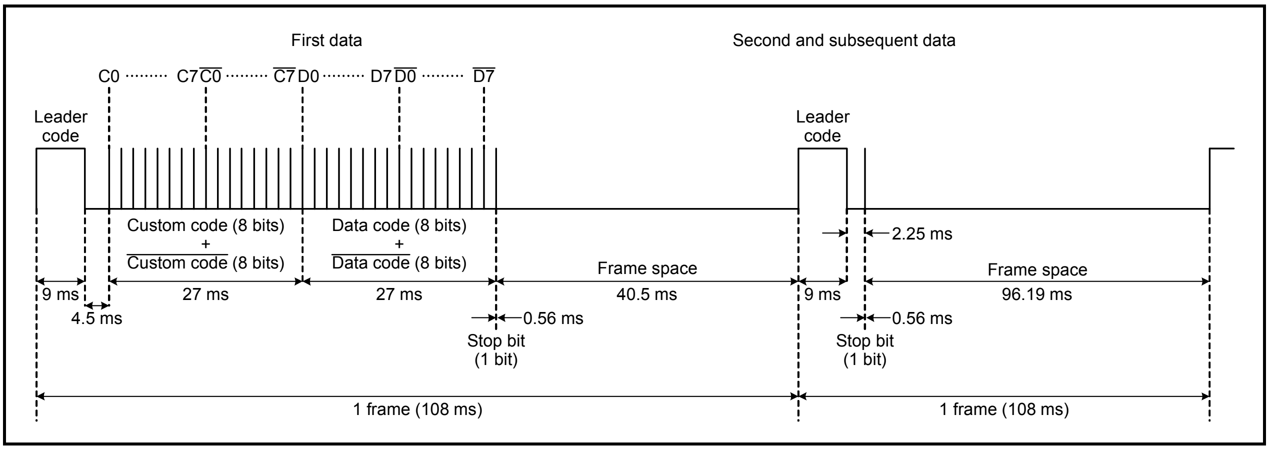


Figure 6.1 Remote Control Data Format

Figure 6.2 shows Enlargements of Each Code.

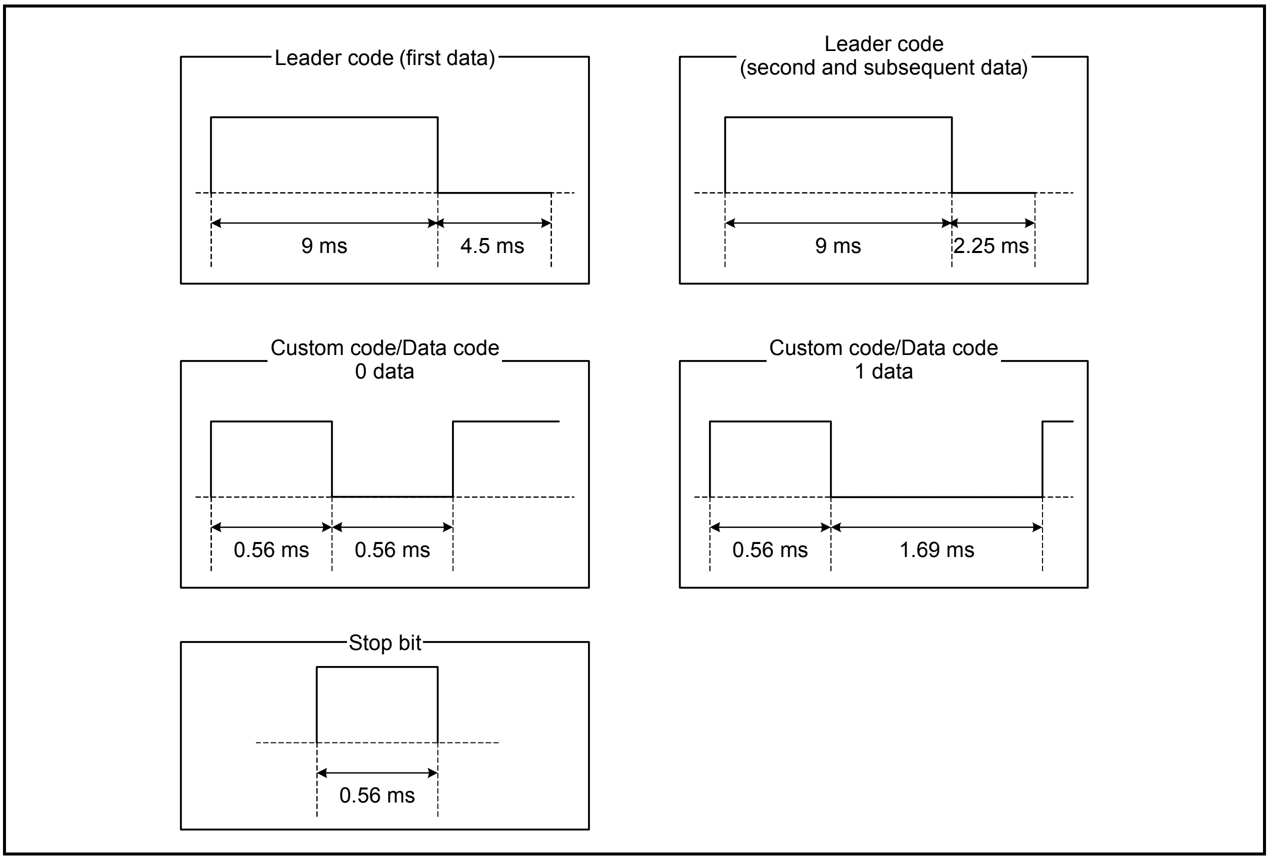


Figure 6.2 Enlargements of Each Code

Table 6.4 lists the Recognition Range by Code.

Table 6.4 Recognition Range by Code

Code Name	Code Recognition Range (1)
Leader code high	6.3 to 11.70 ms
Leader code low	3.15 to 5.85 ms
Custom code high	0.392 to 0.728 ms
Custom code (0 data) low	0.392 to 0.728 ms
Custom code (1 data) low	1.184 to 2.196 ms
Data code high	0.392 to 0.728 ms
Data code (0 data) low	0.392 to 0.728 ms
Data code (1 data) low	1.184 to 2.196 ms
Stop bit	0.392 to 0.728 ms
Frame space	28.35 to 52.65 ms
Leader code high (repeat)	6.3 to 11.70 ms
Leader code low (repeat)	1.576 to 2.924 ms
Stop bit (repeat)	0.392 to 0.728 ms
Frame space (repeat)	67.334 to 125.046 ms

Note:
1. fBT1 = count source f1 (25 MHz) divided by 50: The code can be recognized if the value is within a ±30% error range of the remote control data format value.

6.2 Constants

Table 6.5 lists the Constants Used in the Sample Code.

Table 6.5 Constants Used in the Sample Code

Constant Name	Setting Value	Contents
OK	0	OK (function return value)
NG	1	NG (function return value)
IDLE	0	Measurement standby
LEADER_CODE_H	1	Leader code high interval measured
LEADER_CODE_L	2	Leader code low interval measured
CUSTM_CODE_H	3	Custom code high interval measured
CUSTM_CODE_L	4	Custom code low interval measured
DATA_CODE_H	5	Data code high interval measured
DATA_CODE_L	6	Data code low interval measured
STOP_BIT	7	Stop bit interval measurement
FRAMESPACE	8	Frame space interval measurement
RE_LEADER_CODE_H	9	Leader code high interval (repeat) measured
RE_LEADER_CODE_L	10	Leader code low interval (repeat) measured
RE_STOP_BIT	11	Stop bit interval (repeat) measurement
RE_FRAMESPACE	12	Frame space interval (repeat) measurement
LEADER_CODE_H_POS	0	Sequence position of the leader code high interval recognition range value
LEADER_CODE_L_POS	1	Sequence position of the leader code low interval recognition range value
CUSTM_H_POS	2	Sequence position of the custom code high interval recognition range value
CUSTM_0_L_POS	3	Sequence position of the custom code (0 data) low interval recognition range value
CUSTM_1_L_POS	4	Sequence position of the custom code (1 data) low interval recognition range value
DATA_H_POS	5	Sequence position of the data code high interval recognition range value
DATA_0_L_POS	6	Sequence position of the data code (0 data) low interval recognition range value
DATA_1_L_POS	7	Sequence position of the data code (1 data) low interval recognition range value
STOP_BIT_POS	8	Sequence position of the stop bit interval recognition range value
FRAMESPACE_POS	9	Sequence position of the frame space interval recognition range value
RE_LEADER_CODE_H_POS	10	Sequence position of the leader code high interval (repeat) recognition range value
RE_LEADER_CODE_L_POS	11	Sequence position of the leader code low interval (repeat) recognition range value
RE_STOP_BIT_POS	12	Sequence position of the stop bit interval (repeat) recognition range value
RE_FRAMESPACE_POS	13	Sequence position of the frame space interval (repeat) recognition range value
PULSE_MAX	100	Maximum position of the base timer value storage buffer
REV_PULSE_MAX	10	Maximum position of the inverted data code verification buffer
CUSTM_MAX_BIT_CNT	16	Maximum number of bits received from the custom code
DATA_MAX_8_BIT_CNT	8	Maximum number of bits received from the data code
DATA_MAX_LOW_BIT_CNT	16	Maximum number of data code low intervals received
RCV_COMP_BIT_CNT	32	Number of bits received

6.3 Variables

Table 6.6 lists the Global Variables, Table 6.7 lists the static Variable, and Table 6.8 lists the const Variable.

Table 6.6 Global Variables

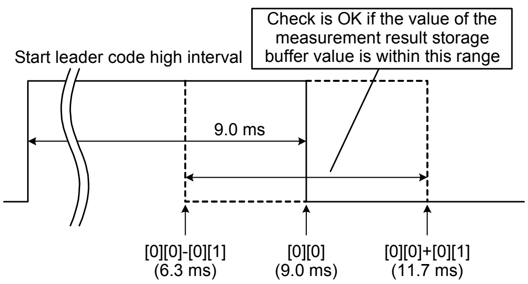
Type	Variable Name	Contents	Function Used
unsigned char	rcv_mode	Processing mode	main, rcv_data, time_over, check_code
unsigned char	pulse_cnt	Measurement result storage buffer counter	main, rcv_data, time_over, set_pulse_value, check_code
unsigned short	pulse[]	Measurement result storage buffer	time_over, set_pulse_value, check_code
unsigned char	rcv_data_cnt	Number of data received	main, rcv_data, time_over, check_code
unsigned char	rcv_bit_cnt	Number of custom data and data code bits received	main, rcv_data, check_code
unsigned char	rev_pulse[]	Buffer for verifying inverted data code	set_reversing_code, cmp_reversing_code
unsigned char	rev_cnt	Buffer counter for verifying inverted data code	main, rcv_data, judge_reversing_code, set_reversing_code, cmp_reversing_code
unsigned char	code_low_cnt	Counter for counting receive data code lows	main, rcv_data, judge_reversing_code

Table 6.7 static Variable

Type	Variable Name	Contents	Function Used
static unsigned short	old_tr	Compared value	set_pulse_value

Table 6.8 const Variable

<table>
<tr><th>Type</th><th>Variable Name</th><th>Contents</th><th>Function Used</th></tr>
<tr><td>const unsigned short</td><td>cmp_tbl[][]</td><td>Received code compare table
• [*][0]: Format value for each interval,
[*][1]: Format value ±30% error range value

Example of a leader code high interval:
cmp_tbl[0][0] = 4500 (9.0 ms)
cmp_tbl[0][1] = 1350 (2.7 ms)

</td><td>check_code</td></tr>
</table>

6.4 Functions

Table 6.9 lists the Functions.

Table 6.9 Functions

Function Name	Outline
main	Main processing
iio_init	Intelligent I/O initialization
timer_a0_init	Timer A0 initialization
timer_a1_init	Timer A1 initialization
rcv_data	Receive data settings
time_over	Time over settings
set_pulse_value	Pulse value settings
check_code	Receive data determination processing
cmp_pulse	Receive data range check
judge_reversing_code	Inverted data code determination
set_reversing_code	Inverted data code buffer setting
cmp_reversing_code	Inverted data code comparison

6.5 Function Specifications

The following tables list the sample code function specifications.

main	
Outline	Main processing
Header	None
Declaration	void main(void)
Description	Maskable interrupts are disabled; the system clock, intelligent I/O, timer A0, and timer A1 are initialized; after the intelligent I/O base timer starts counting, maskable interrupts are enabled; and then the following processes are performed. (1) The time measurement function TM10R interrupt request bit is monitored, and the remote control waveform input pulse width is measured. (2) Monitor the timer A1 interrupt request flag, and time manage one frame.
Argument	None
Returned value	None

iio_init	
Outline	Intelligent I/O initialization
Header	None
Declaration	void iio_init(void)
Description	Set the time measurement function (channel 0 in group 1) of the intelligent I/O.
Argument	None
Returned value	None

timer_a0_init	
Outline	Timer A0 initialization
Header	None
Declaration	void timer_a0_init(void)
Description	Set the timer A0 operating mode to timer mode.
Argument	None
Returned value	None

timer_a1_init	
Outline	Timer A1 initialization
Header	None
Declaration	void timer_a1_init(void)
Description	Set the timer A1 operating mode to event counter mode, and set the event trigger to the underflow of timer A0.
Argument	None
Returned value	None

rcv_data	
Outline	Receive data settings
Header	None
Declaration	void rcv_data(void)
Description	The following processes are performed depending on the mode. (1) When the processing mode is “measurement standby”, “frame space interval measurement”, or “frame space interval (repeat) measurement: Set the count value for timer A0 and timer A1, and start the timer. (2) When the processing mode is “leader code high interval measurement”: • Initialize the variable for the number of custom data and data code bits received • Initialize the variable for the buffer counter for verifying inverted data code • Initialize the variable for the receive data code low counter and • Initialize the variable for the number of received data Then pulse setting and receive data determination are performed.
Argument	None
Returned value	None

time_over	
Outline	Time over settings
Header	None
Declaration	void time_over(void)
Description	After one frame of time has elapsed, 0 is set to the measurement result storage buffer, and the processing mode is set to “measurement standby”.
Argument	None
Returned value	None

set_pulse_value	
Outline	Pulse value setting
Header	None
Declaration	void set_pulse_value(void)
Description	The difference between the base timer value read from the G1TM0 register and the previous base timer value are calculated, and the value of the difference is stored in the measurement result storage buffer as the pulse width. Then, store the base timer value read from the G1TM0 register to the old_tr variable.
Argument	None
Returned value	None

check_code	
Outline	Receive data determination
Header	None
Declaration	void check_code(void)
Description	Perform the receive data range check according to the processing mode, and a determination is made to see if the processing mode conforms to the pulse width stored in the measurement result storage buffer. If the values correspond, set the following processing mode to the processing mode; if the values do not correspond, set the processing mode to “measurement standby”. If the processing mode is “data code low interval measurement”, then the inverted data code determination processing is performed.
Argument	None
Returned value	None

cmp_pulse	
Outline	Receive data range check
Header	None
Declaration	unsigned char cmp_pulse(unsigned short d_pulse, unsigned short hi, unsigned short low)
Description	Determine the pulse width stored in the measurement result storage buffer is within ±30% error range of the pulse width for the remote control data format.
Argument	unsigned short d_pulse: Pulse width for the remote control data format unsigned short hi: +30% of the pulse width for the remote control data format unsigned short low: -30% of the pulse width for the remote control data format
Returned value	Results OK: Within the error range NG: Outside the error range

judge_reversing_code	
Outline	Inverted data code determination
Header	None
Declaration	unsigned char judge_reversing_code(unsigned char rtn0, unsigned char rtn1)
Description	When using the “non-inverted data code low interval measurement”, data for inverted data code comparison is set to the inverted data code verification buffer. When using the “inverted data code low interval measurement”, the inverted data code is determined based on the data set to the inverted data code verification buffer.
Argument	unsigned char rtn0: Data code (0 data) low interval determination result OK: Data code (0 data) low interval is within the error range NG: Data code (0 data) low interval is outside the error range unsigned char rtn1: Data code (1 data) low interval determination result OK: Data code (1 data) low interval is within the error range NG: Data code (1 data) low interval is outside the error range
Returned value	Results OK: Inverted data present NG: No inverted data present

set_reversing_code	
Outline	Inverted data code buffer setting
Header	None
Declaration	void set_reversing_code(unsigned char rtn0, unsigned char rtn1)
Description	When OK is the determination result for the data code (0 data) low interval, F1h is set to the inverted data code verification buffer. When OK is the determination result for the data code (1 data) low interval, F0h is set to the inverted data code verification buffer.
Argument	unsigned char rtn0: Data code (0 data) low interval determination result OK: Data code (0 data) low interval is within the error range NG: Data code (0 data) low interval is outside the error range unsigned char rtn1: Data code (1 data) low interval determination result OK: Data code (1 data) low interval is within the error range NG: Data code (1 data) low interval is outside the error range
Returned value	None

cmp_reversing_code	
Outline	Comparison of inverted data code
Header	None
Declaration	unsigned char cmp_reversing_code(unsigned char rtn0, unsigned char rtn1)
Description	When OK is the determination result for the data code (0 data) low interval, if F0h is set to the inverted data code verification buffer, then inverted data is determined to be preset. When OK is the determination result for the data code (1 data) low interval, F1h is set to the inverted data code verification buffer, then inverted data is determined to be preset. Other than those above, inverted data is determined to be not preset.
Argument	unsigned char rtn0: Data code (0 data) low interval determination result OK: Data code (0 data) low interval is within the error range NG: Data code (0 data) low interval is outside the error range unsigned char rtn1: Data code (1 data) low interval determination result OK: Data code (1 data) low interval is within the error range NG: Data code (1 data) low interval is outside the error range
Returned value	Results OK: Inverted data OK NG: Inverted data NG

6.6 Flowcharts

6.6.1 Main Processing

Figure 6.3 shows the Main Processing.

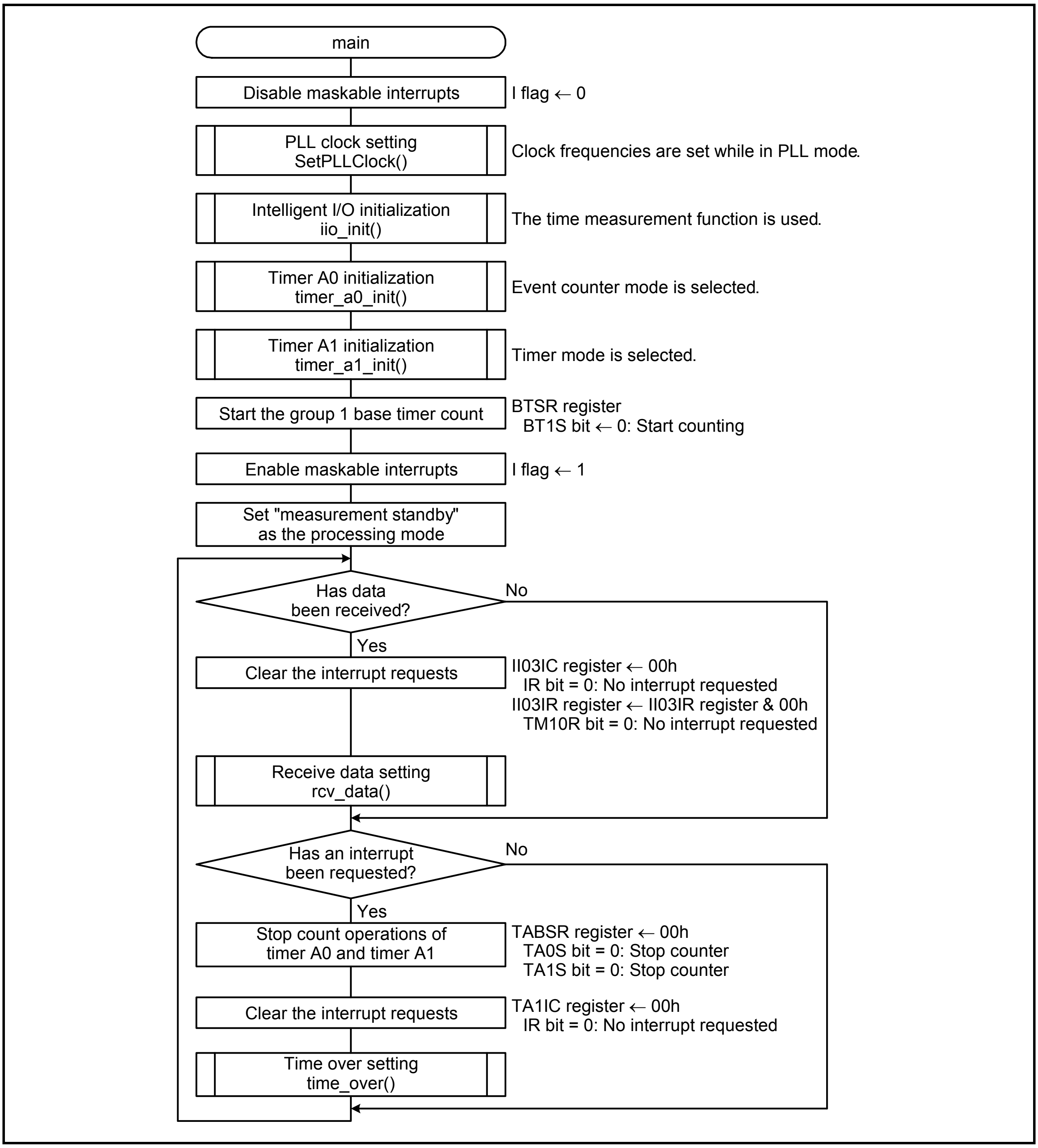


Figure 6.3 Main Processing

6.6.2 Initial Settings of the Intelligent I/O

Figure 6.4 shows Initial Settings of the Intelligent I/O.

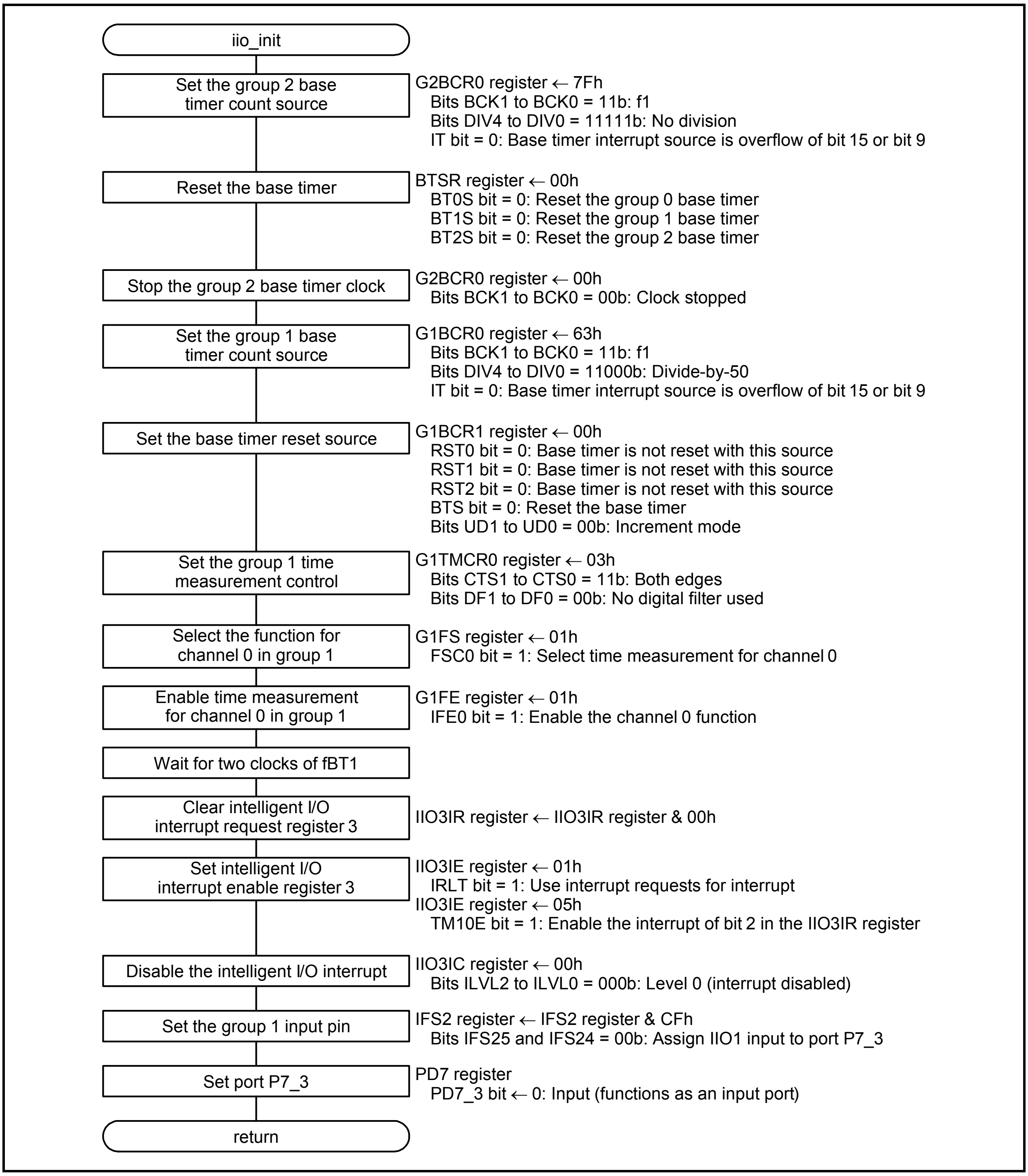


Figure 6.4 Initial Settings of the Intelligent I/O

6.6.3 Timer A0 Initialization

Figure 6.5 shows Timer A0 Initialization.

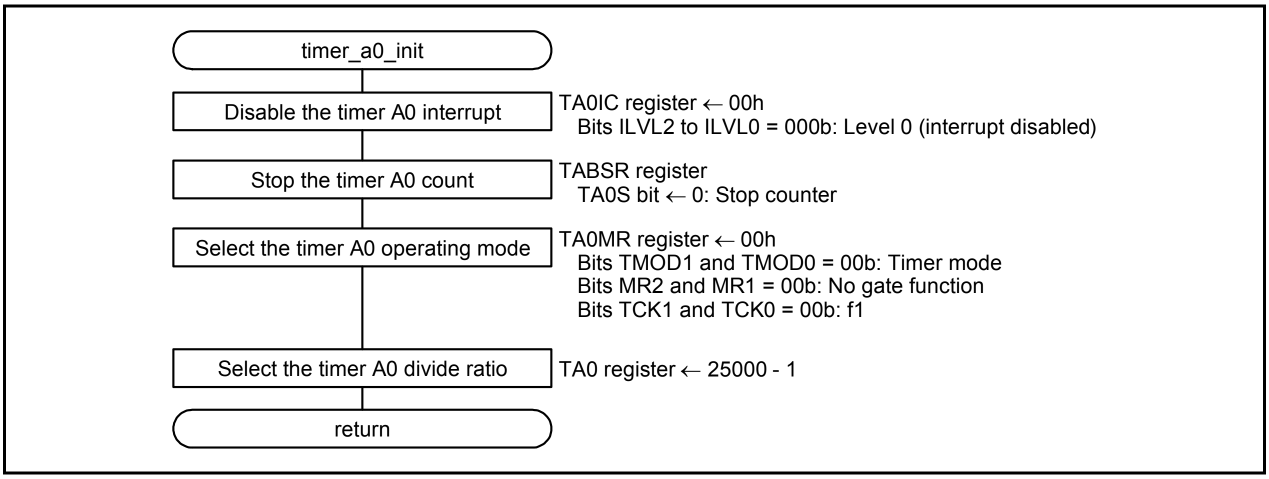


Figure 6.5 Timer A0 Initialization

6.6.4 Timer A1 Initialization

Figure 6.6 shows Timer A1 Initialization.

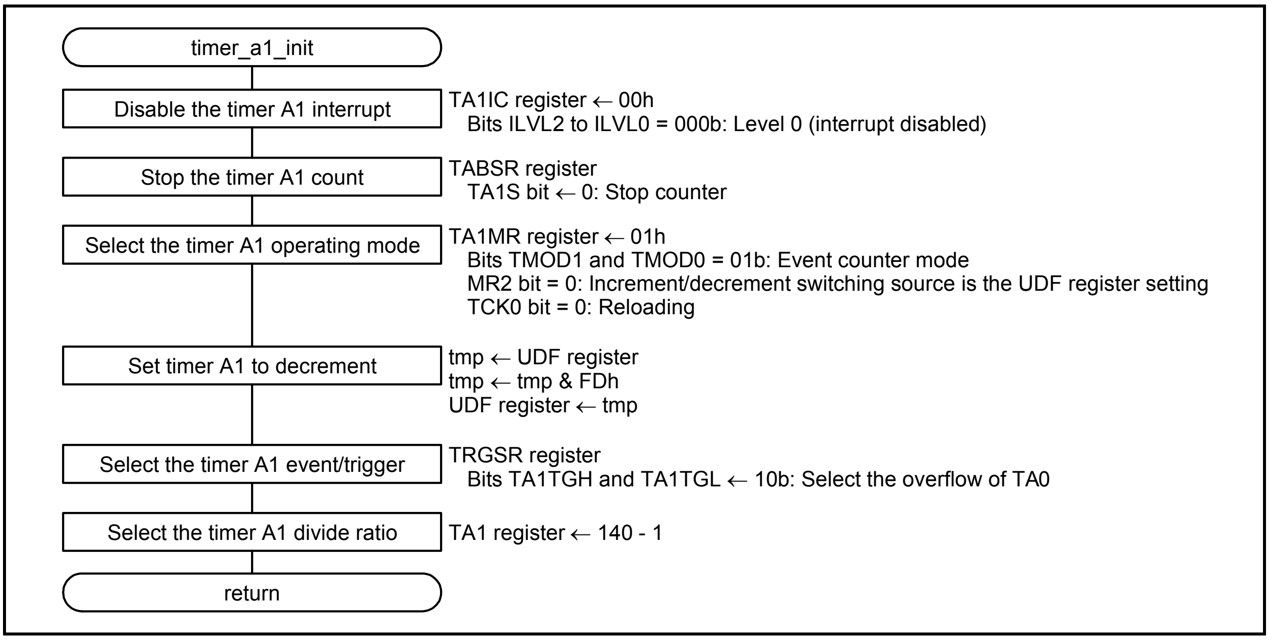


Figure 6.6 Timer A1 Initialization

6.6.5 Receive Data Settings

Figure 6.7 shows the Receive Data Settings.

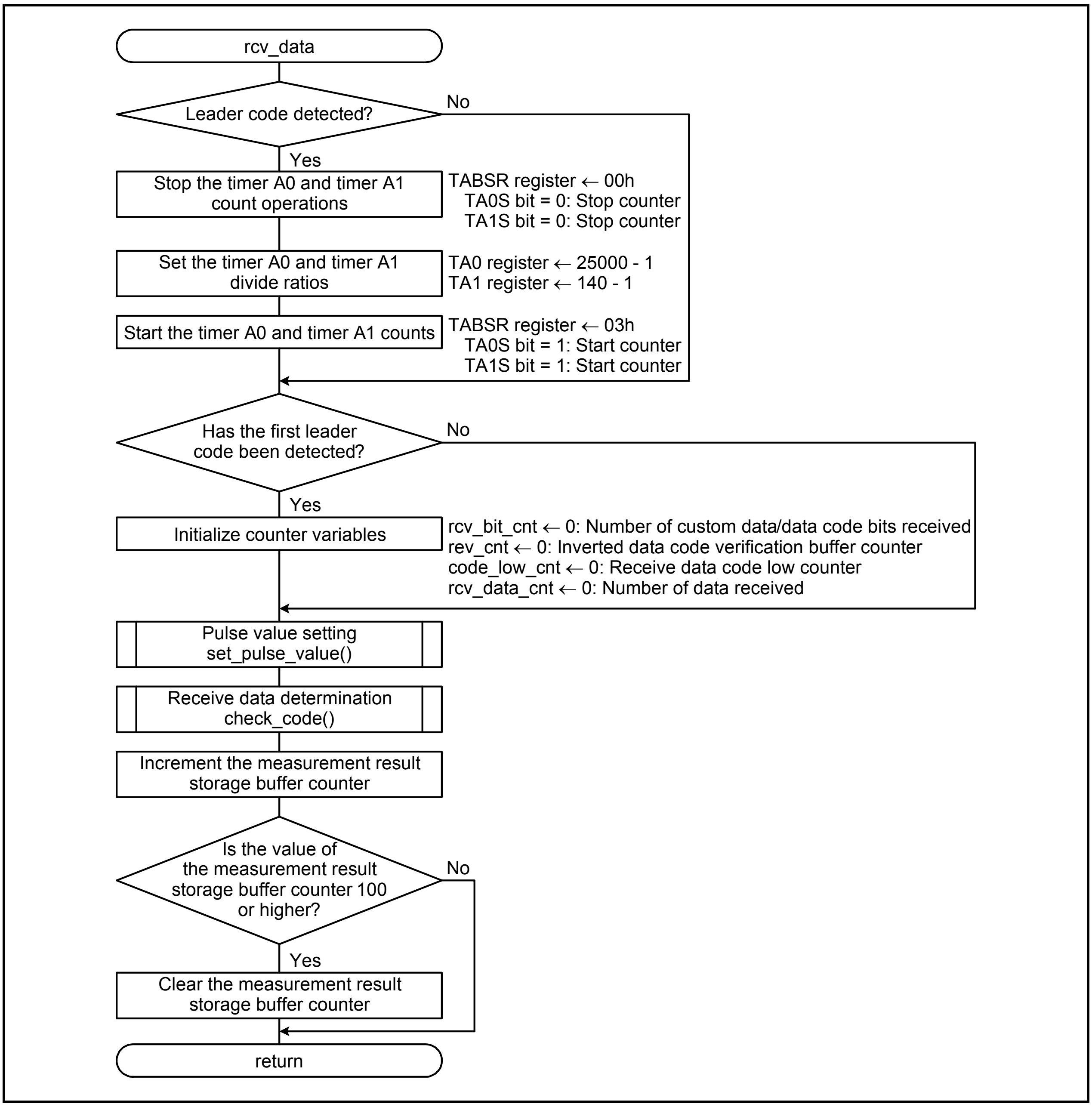


Figure 6.7 Receive Data Settings

6.6.6 Time Over Settings

Figure 6.8 shows the Time Over Settings.

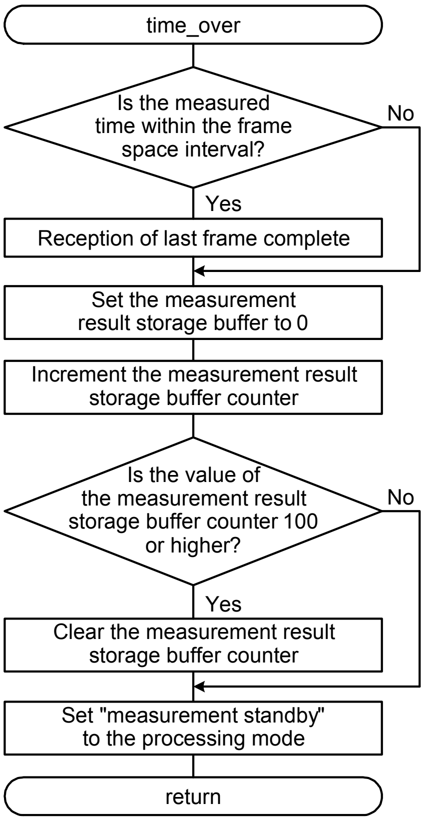


Figure 6.8 Time Over Settings

6.6.7 Pulse Value Settings

Figure 6.9 shows the Pulse Value Settings.

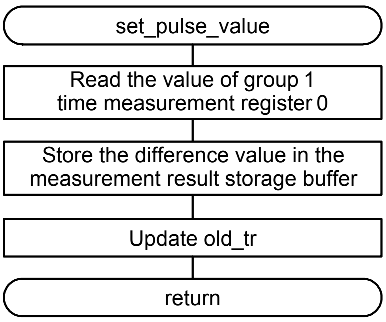


Figure 6.9 Pulse Value Settings

6.6.8 Receive Data Determination Processing

Figures 6.10 to 6.23 show the receive data determination processing.

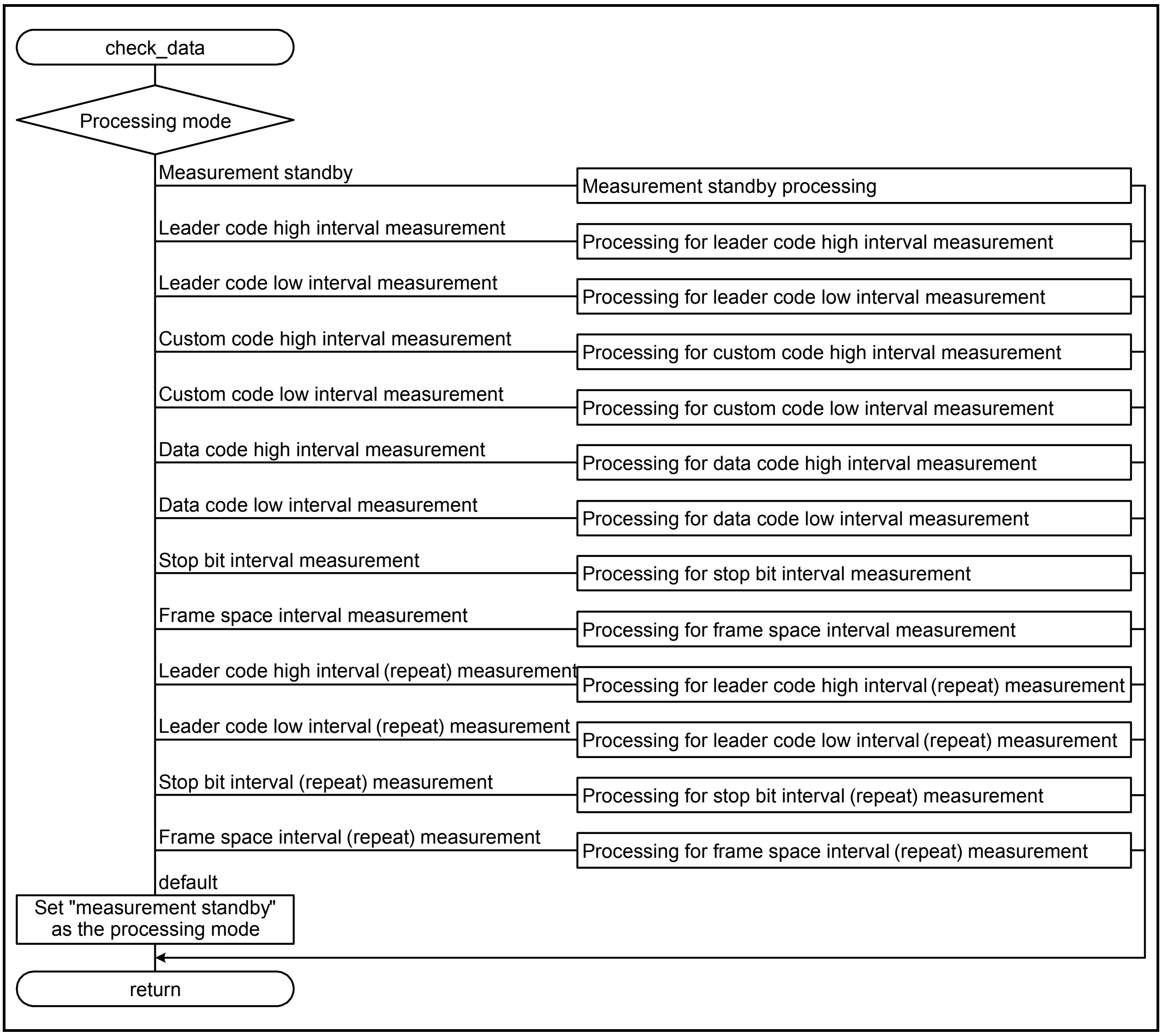


Figure 6.10 Receive Data Determination Processing (1/14)

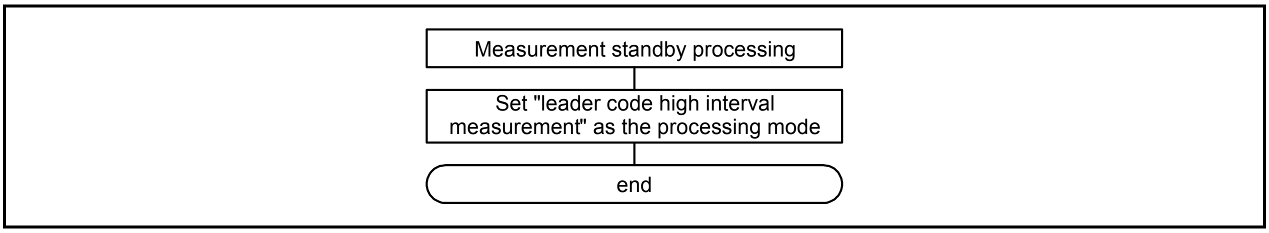


Figure 6.11 Receive Data Determination Processing (2/14)

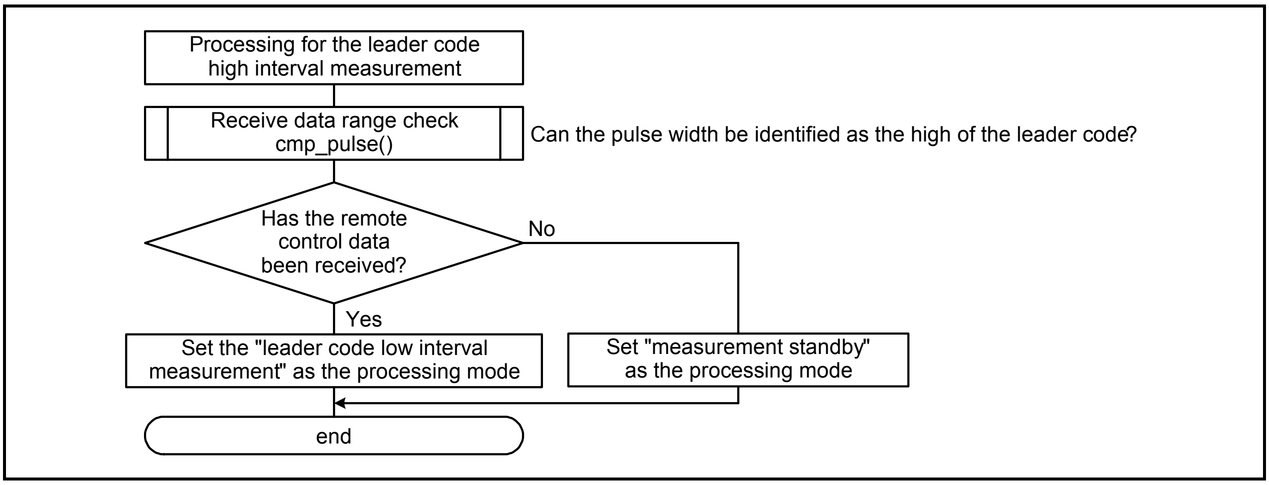


Figure 6.12 Receive Data Determination Processing (3/14)

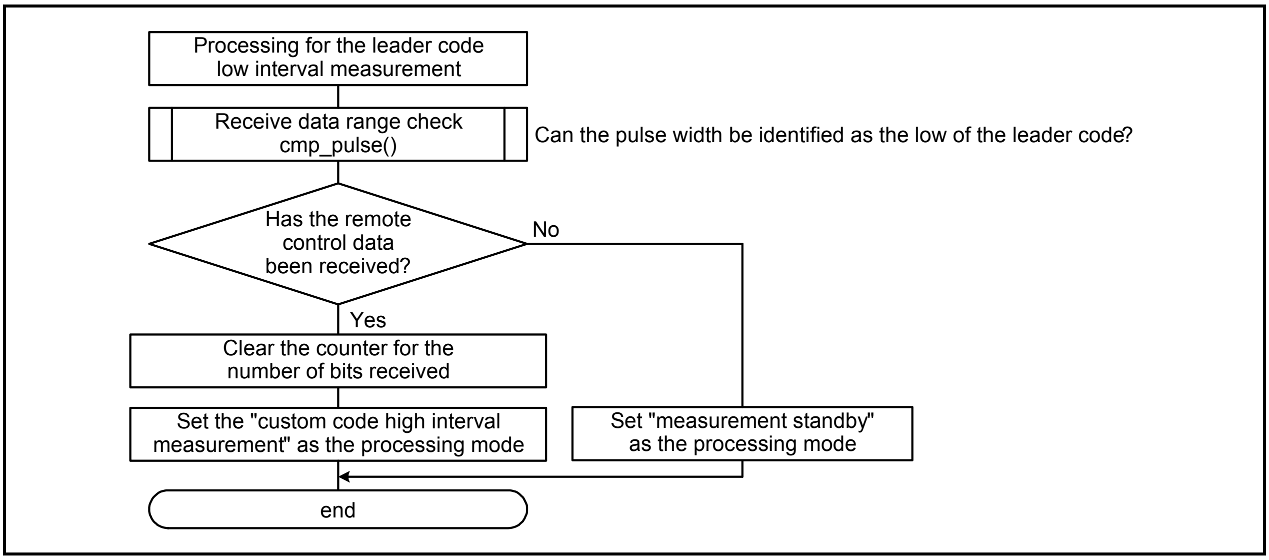


Figure 6.13 Receive Data Determination Processing (4/14)

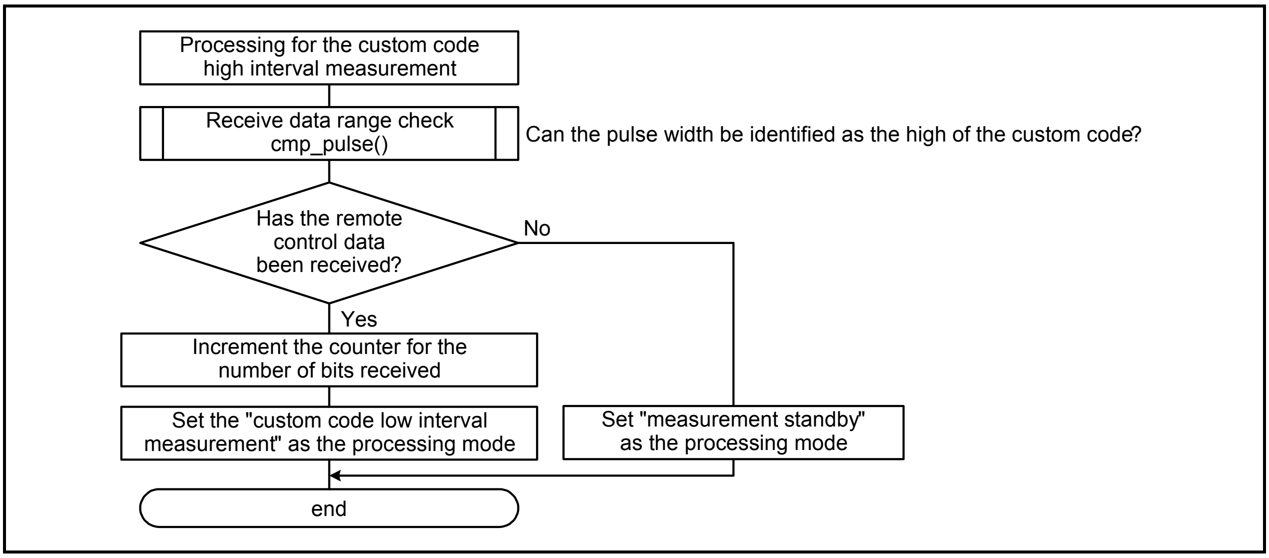


Figure 6.14 Receive Data Determination Processing (5/14)

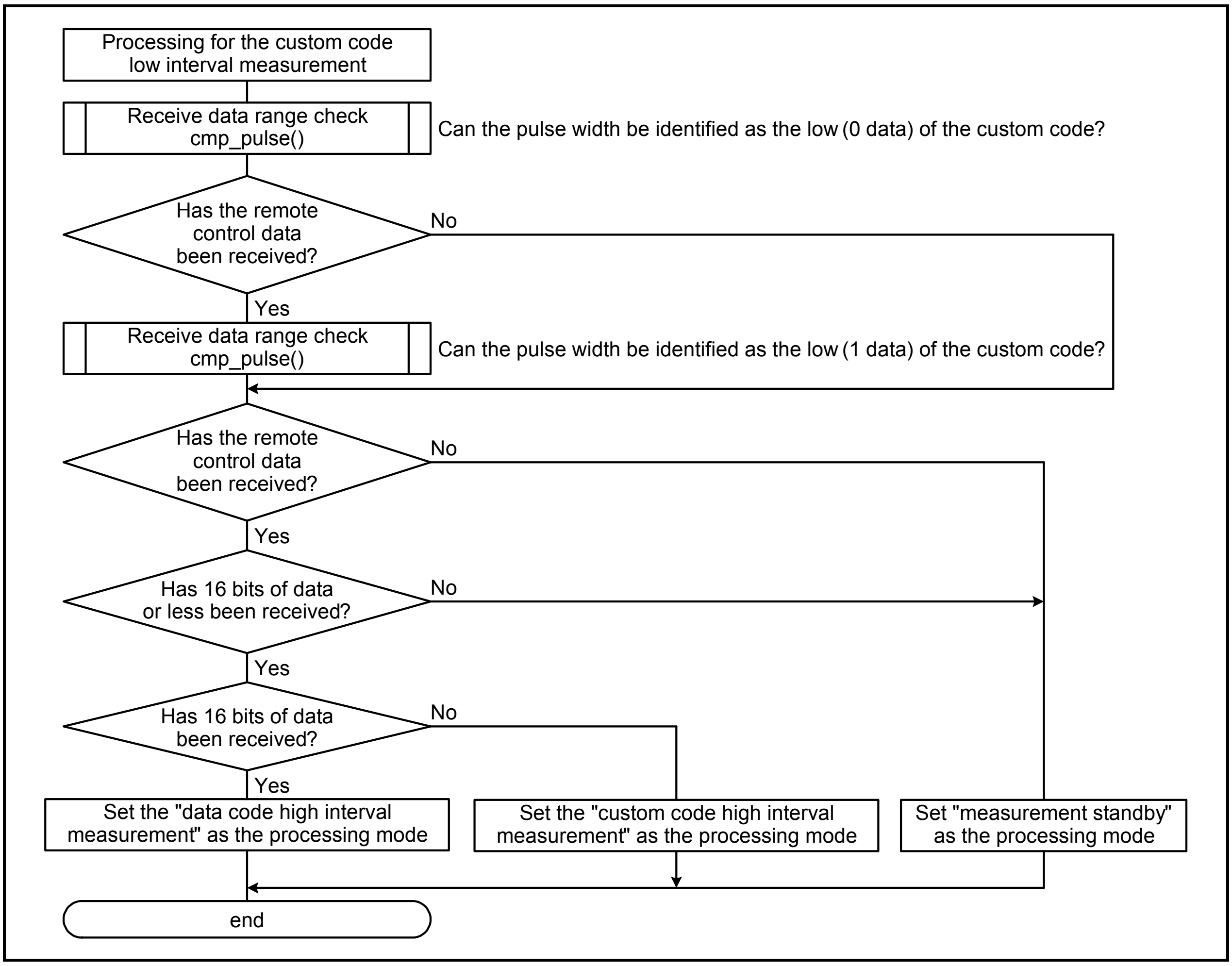


Figure 6.15 Receive Data Determination Processing (6/14)

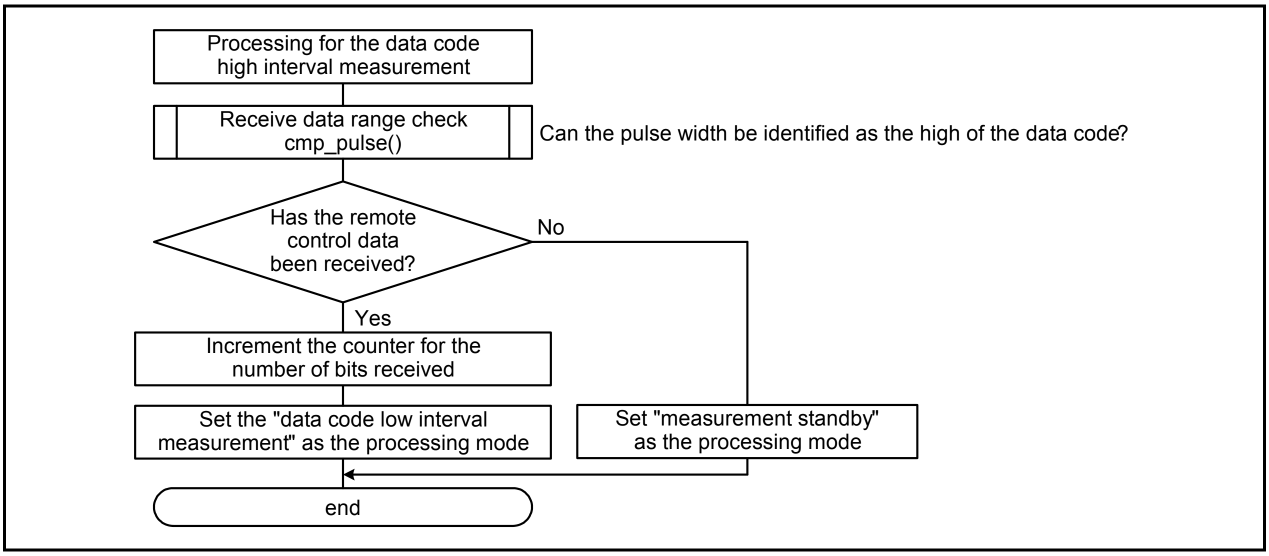


Figure 6.16 Receive Data Determination Processing (7/14)

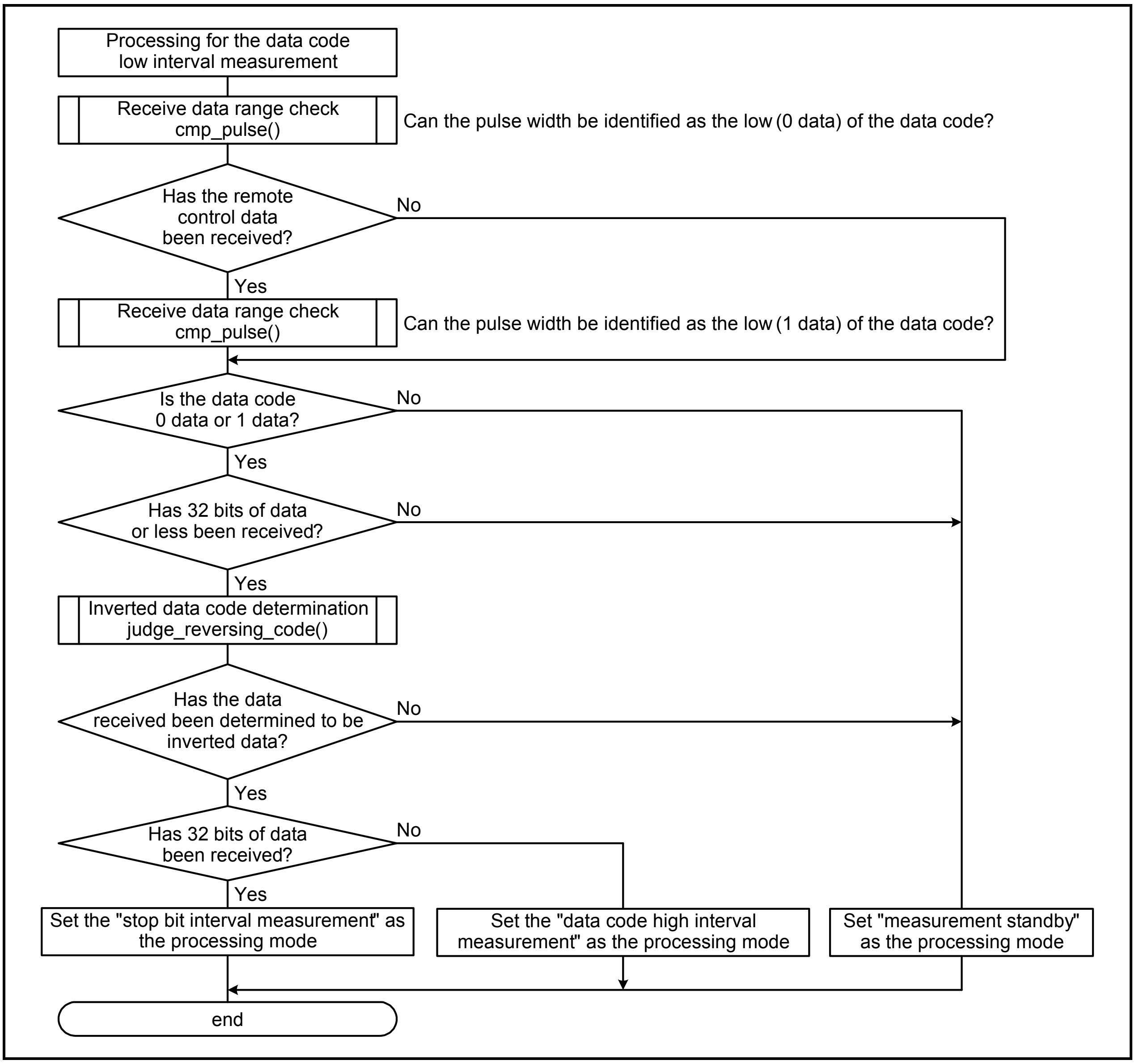


Figure 6.17 Receive Data Determination Processing (8/14)

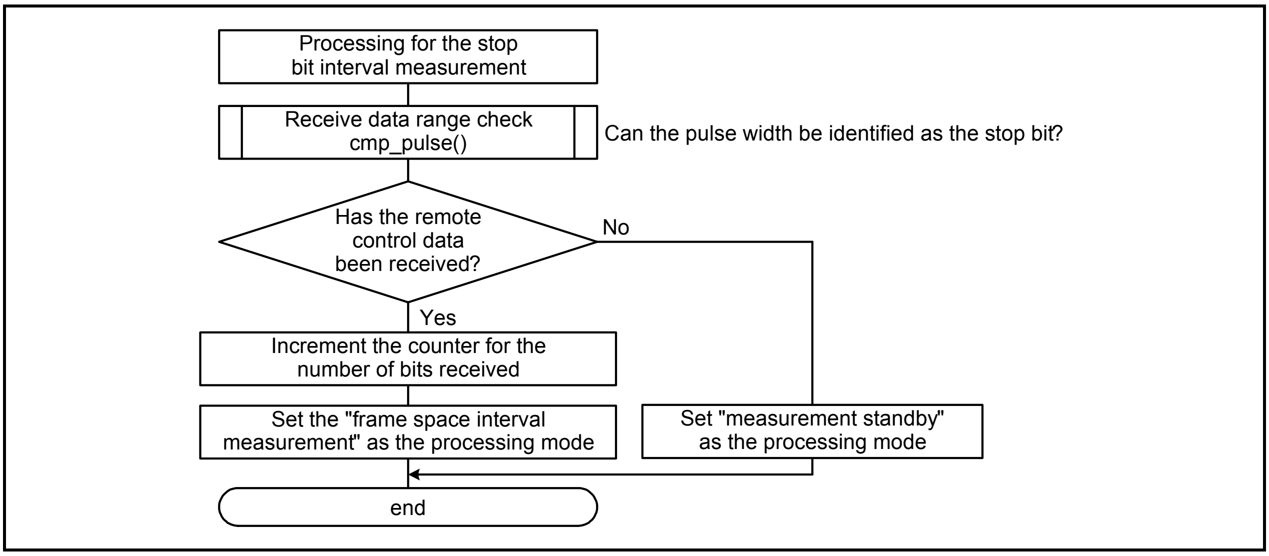


Figure 6.18 Receive Data Determination Processing (9/14)

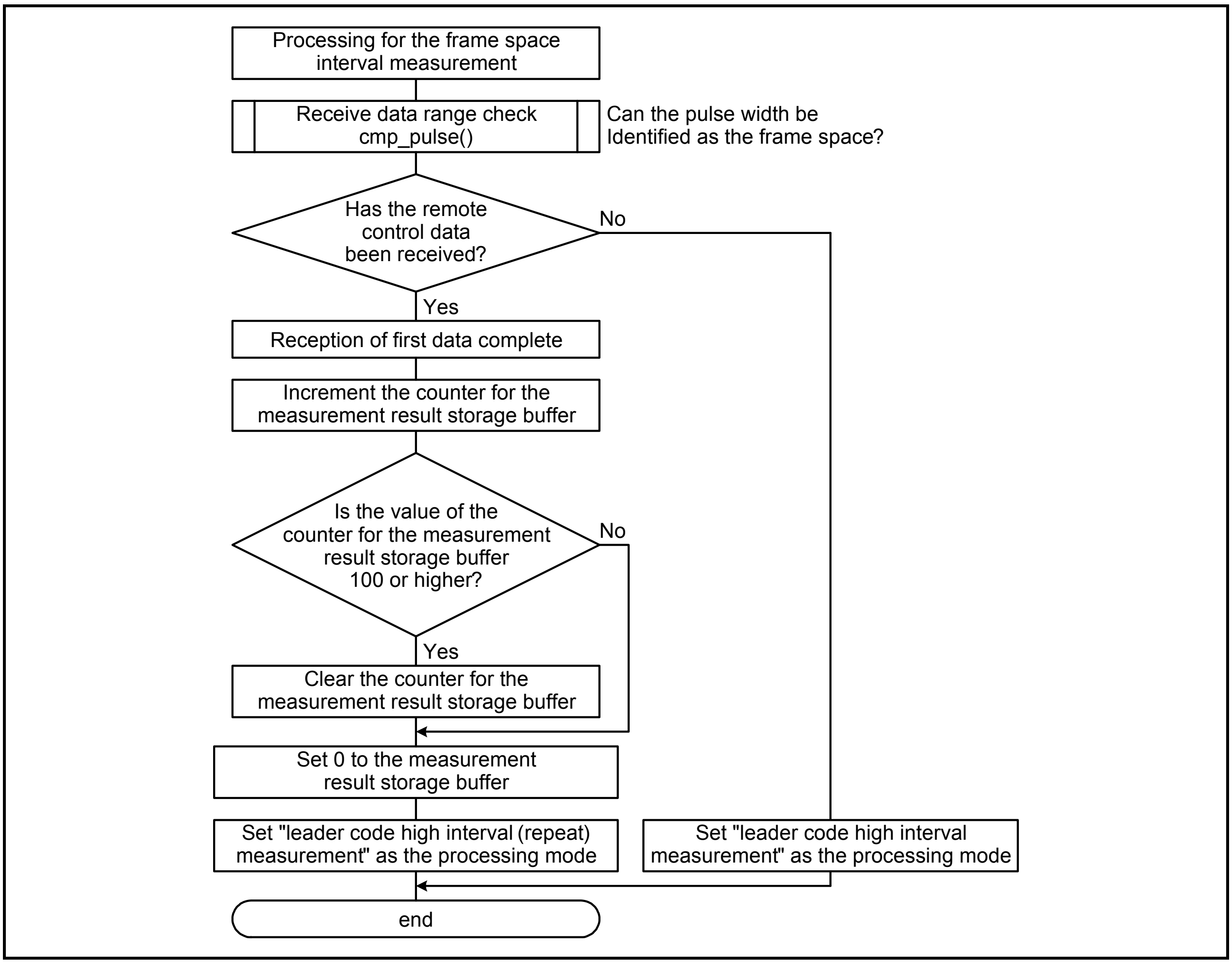


Figure 6.19 Receive Data Determination Processing (10/14)

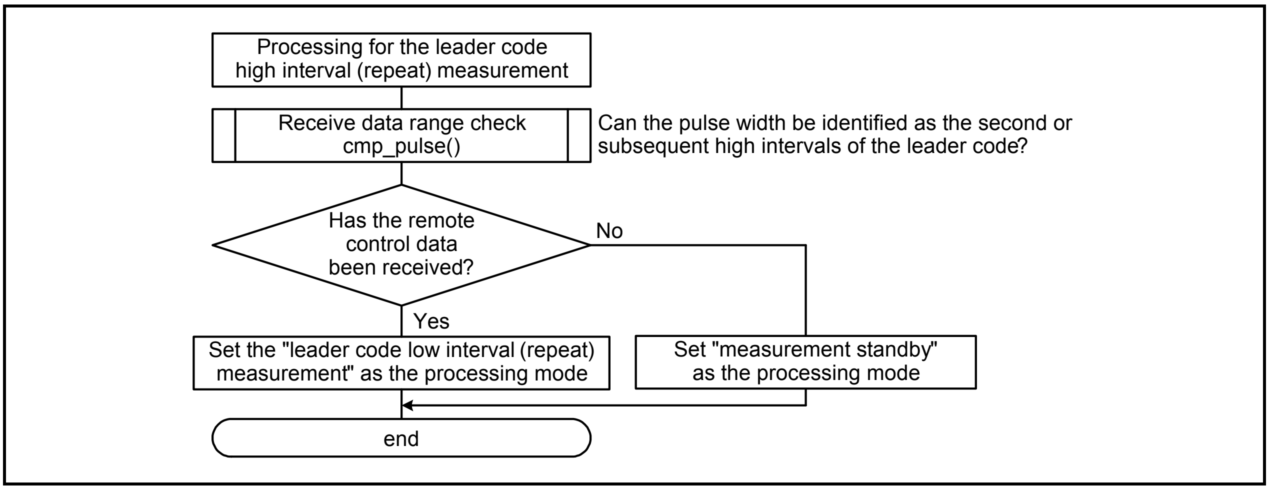


Figure 6.20 Receive Data Determination Processing (11/14)

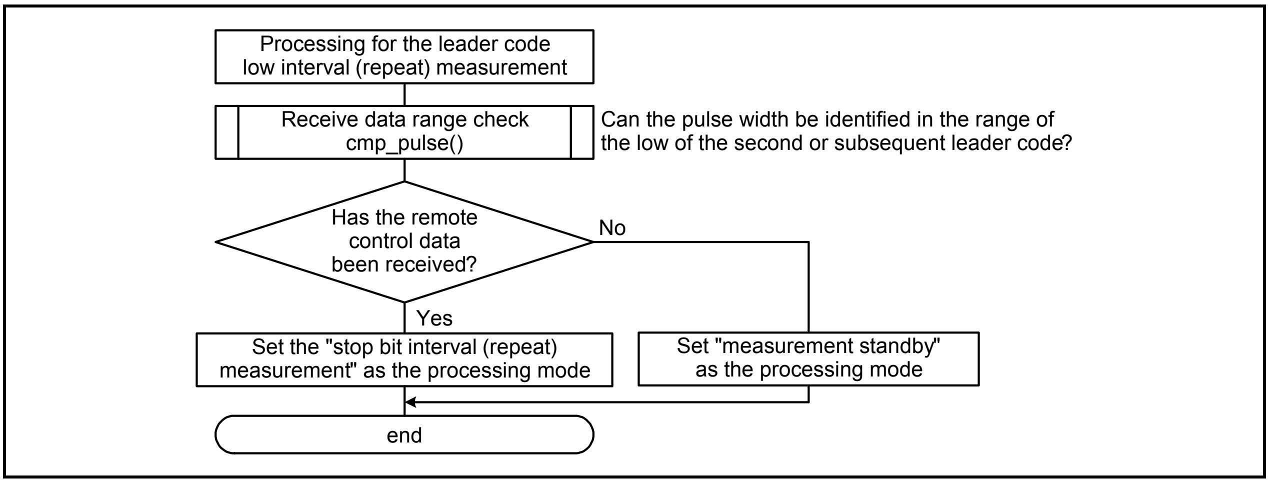


Figure 6.21 Receive Data Determination Processing (12/14)

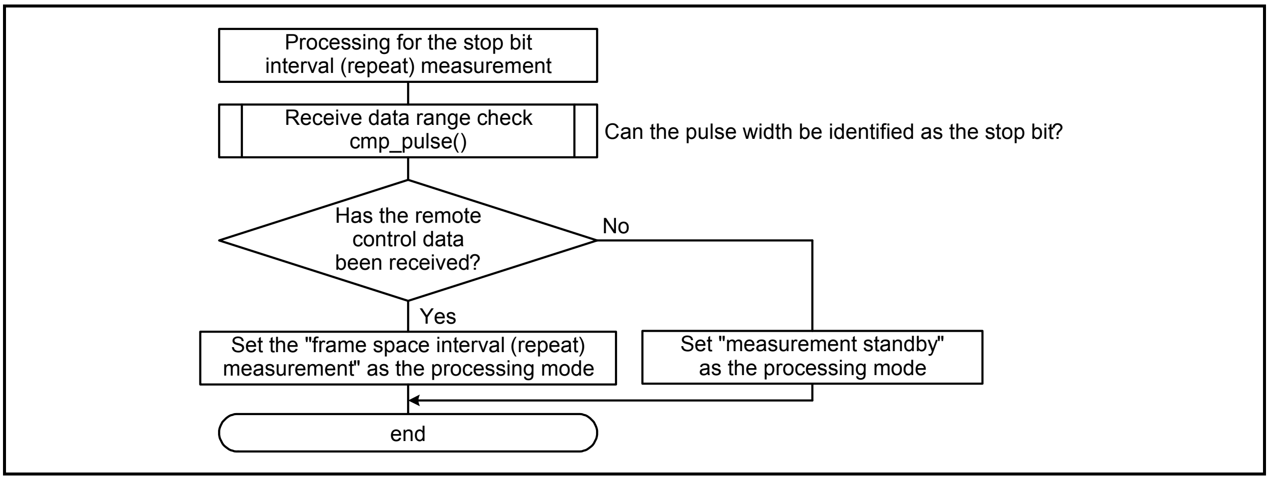


Figure 6.22 Receive Data Determination Processing (13/14)

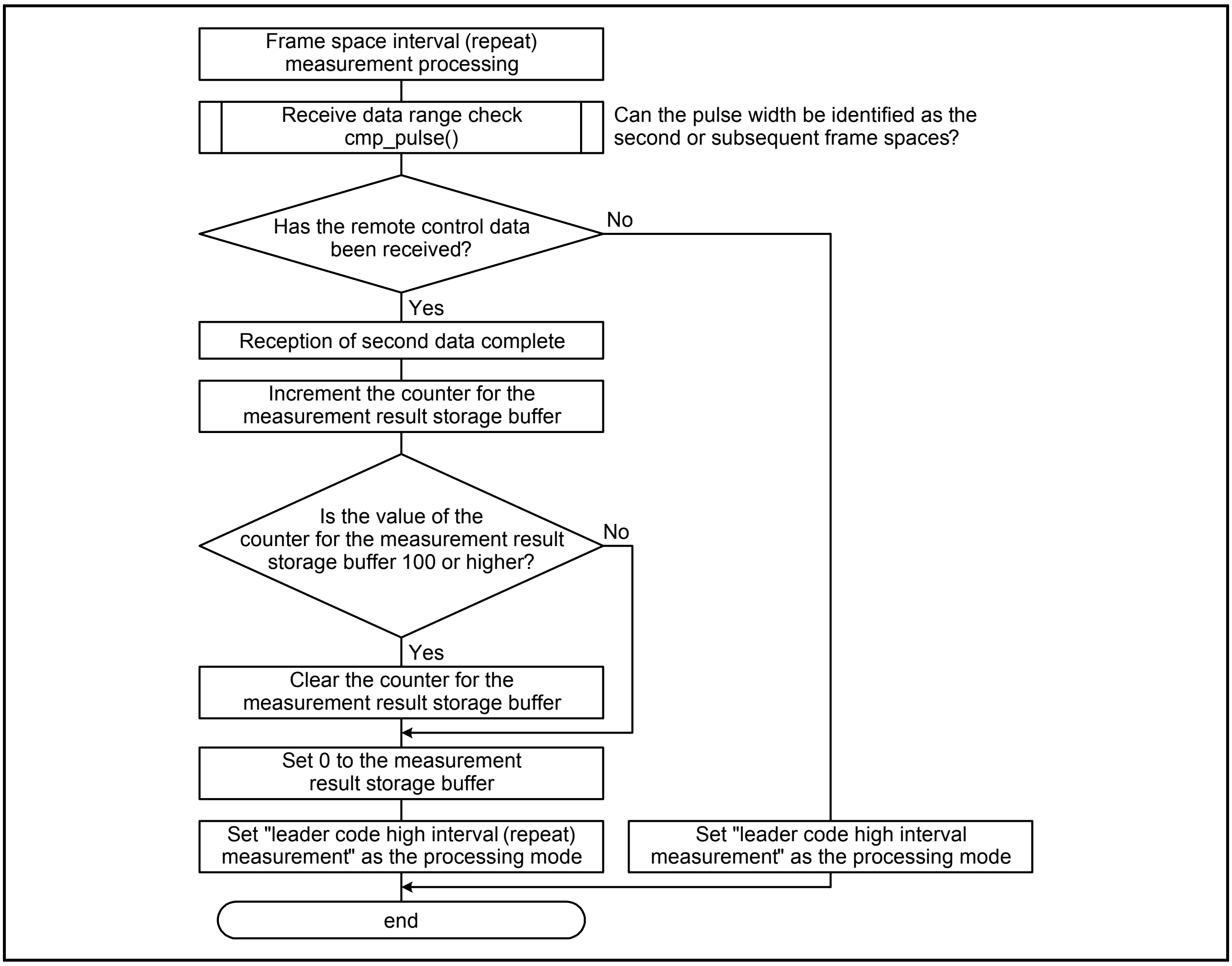


Figure 6.23 Receive Data Determination Processing (14/14)

6.6.9 Receive Data Range Check

Figure 6.24 shows the Receive Data Range Check.

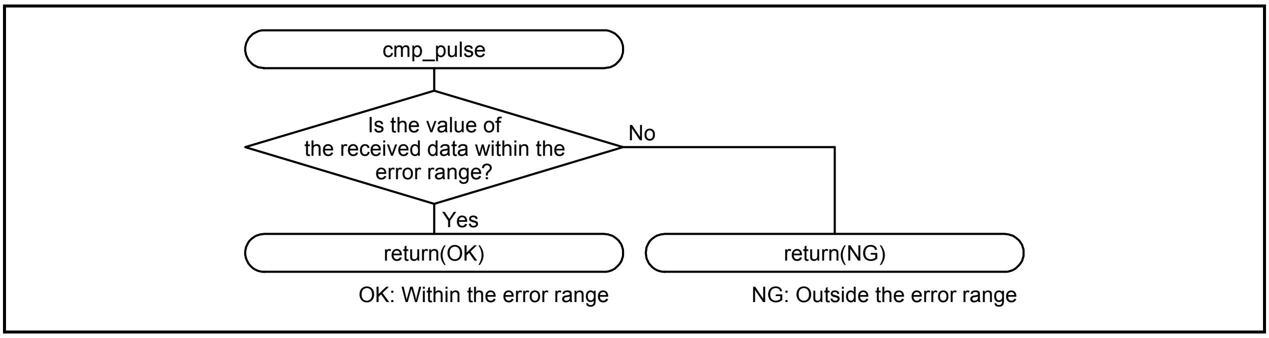


Figure 6.24 Receive Data Range Check

6.6.10 Inverted Data Code Determination

Figure 6.25 shows Inverted Data Code Determination.

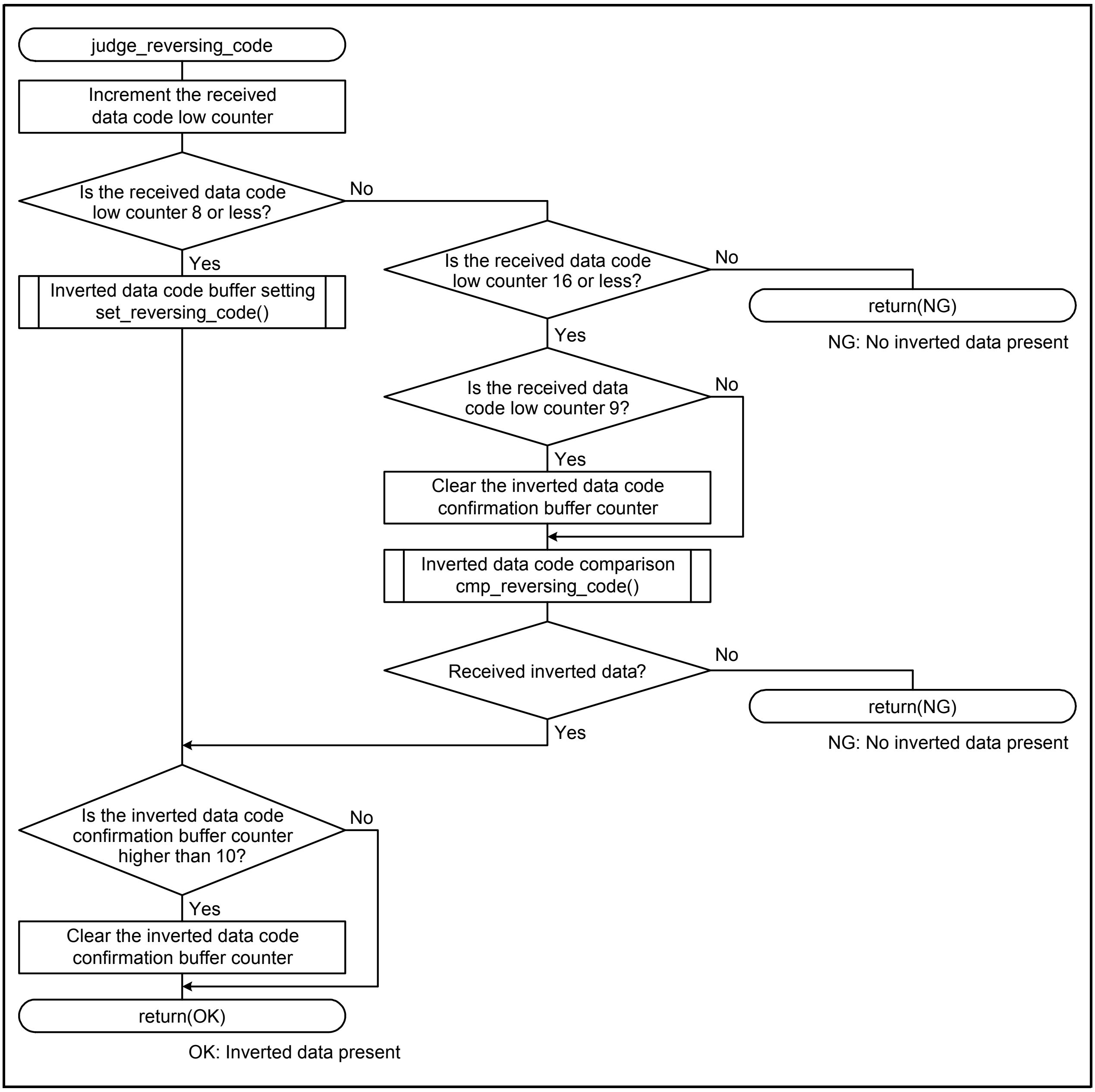


Figure 6.25 Inverted Data Code Determination

6.6.11 Inverted Data Code Buffer Settings

Figure 6.26 shows the Inverted Data Code Buffer Setting.

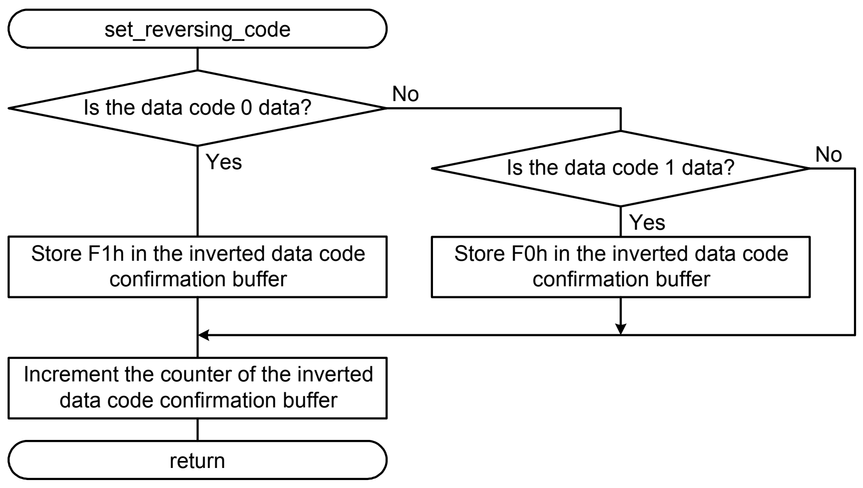


Figure 6.26 Inverted Data Code Buffer Setting

6.6.12 Inverted Data Code Comparison

Figure 6.27 shows the Inverted Data Code Comparison.

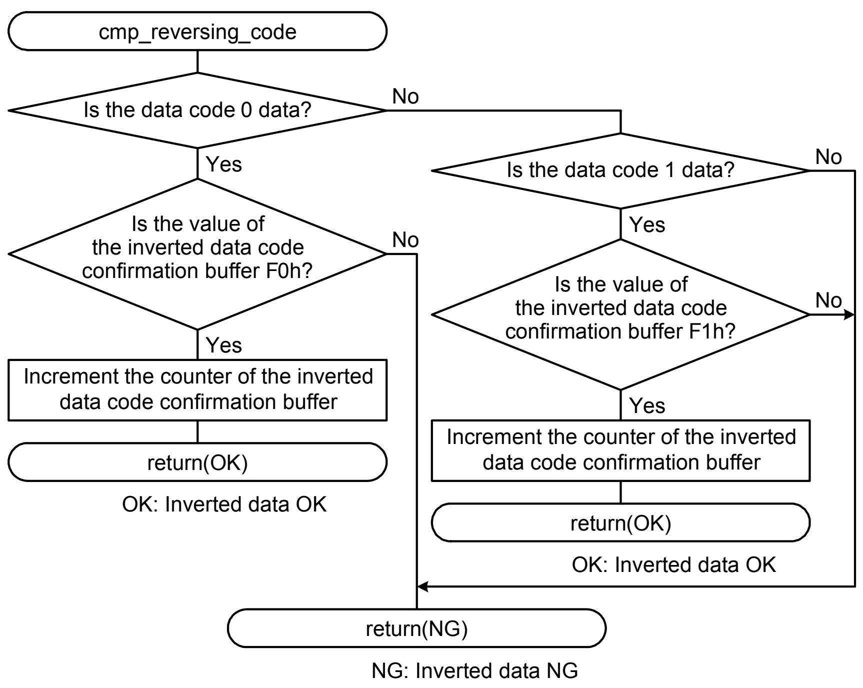


Figure 6.27 Inverted Data Code Comparison

7. Appendix

7.1 Overview of the Remote Control Signal Reception

An infrared signal transmitted from the remote control is transmitted to the receiver at a fixed frequency (carrier frequency). As the infrared signal is weakened through diffusion at the receiver, the output of the infrared receiving element must be amplified with a preamplifier. Also, passing through a bandpass filter (BPF) allows an accurate remote control signal to be obtained by extracting only the carrier waveform element and detecting and rectifying the waveform. Also, negative logic (inverted) data is output from the infrared signal remote control preamp. In this case, the carrier frequency is set to 38 kHz.

Figure 7.1 shows a Block Diagram of the Inside of the Receiving Module where infrared signals from the remote control are received, and Figure 7.2 shows a Carrier Waveform.

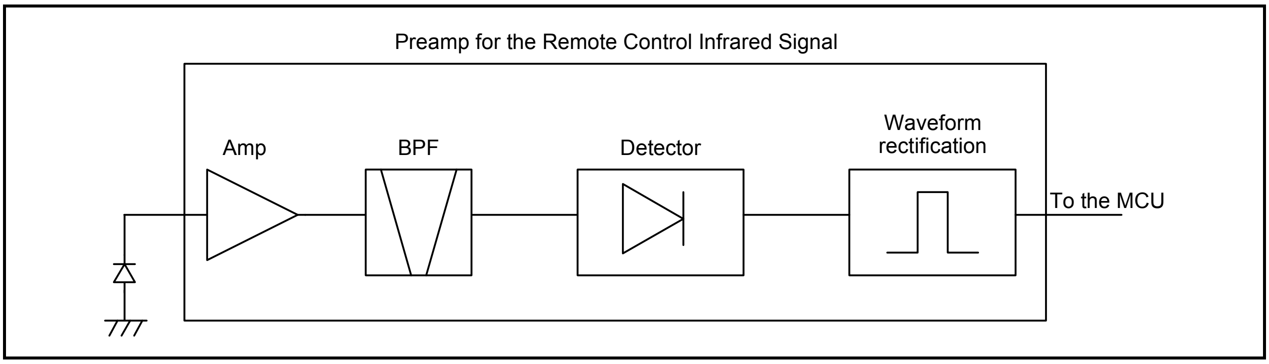
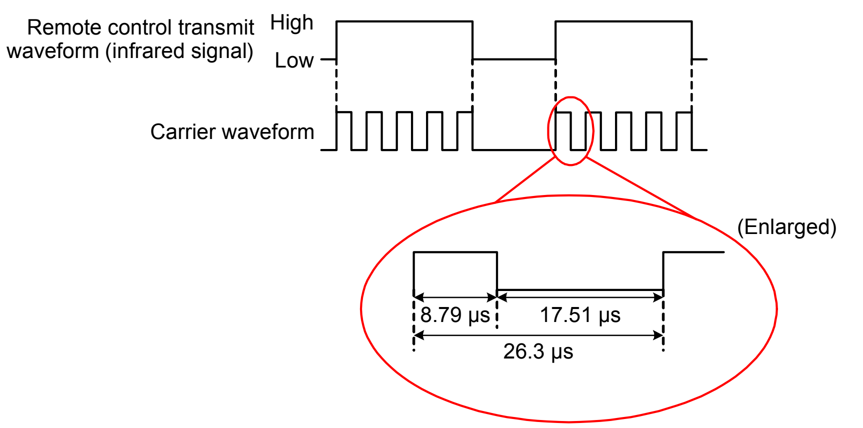


Figure 7.1 Block Diagram of the Inside of the Receiving Module



An infrared signal is not continuously transmitted throughout the entire high interval of the remote control transmit waveform. The interval of infrared signal transmission/no transmission is repeated at a fixed frequency (carrier frequency).

Figure 7.2 Carrier Waveform

8. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

9. Reference Documents

R32C/116 Group User's Manual: Hardware Rev.1.20
R32C/117 Group User's Manual: Hardware Rev.1.20
R32C/118 Group User's Manual: Hardware Rev.1.20
The latest versions can be downloaded from the Renesas Electronics website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Electronics website.

C Compiler Manual
R32C Series C Compiler Package V.1.02
C Compiler User's Manual Rev.2.00
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

Revision History	R32C/100 Series Remote Control Signal Reception Using the Intelligent I/O

Rev.	Date	Description	
		Page	Summary
1.00	July 31, 2013	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU 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. Handling of Unused Pins

   Handle unused pins in accord 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the 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.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 2.2