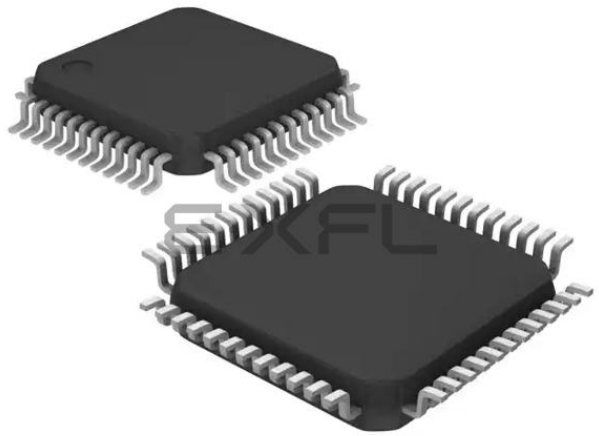


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	V850ES
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	CANbus, CSI, EBI/EMI, I²C, UART/USART, USB
Peripherals	DMA, LVD, PWM, WDT
Number of I/O	32
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	24K x 8
Voltage - Supply (Vcc/Vdd)	2.85V ~ 3.6V
Data Converters	A/D 6x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd70f3816ga-gam-ax

1.6.2 Internal units

(1) CPU

The CPU uses five-stage pipeline control to enable single-clock execution of address calculations, arithmetic logic operations, data transfers, and almost all other instruction processing.

Other dedicated on-chip hardware, such as a multiplier (16 bits × 16 bits → 32 bits) and a barrel shifter (32 bits) contribute to faster complex processing.

(2) Bus control unit (BCU)

The BCU starts a required external bus cycle based on the physical address obtained by the CPU. When an instruction is fetched from external memory space and the CPU does not send a bus cycle start request, the BCU generates a prefetch address and prefetches the instruction code. The prefetched instruction code is stored in an instruction queue.

(3) Flash memory (ROM)

This is a 256/128/64/32/16 KB flash memory mapped to addresses 0000000H to 003FFFFH/0000000H to 001FFFFH/0000000H to 000FFFFH/0000000H to 0007FFFH/0000000H to 0003FFFH/. It can be accessed from the CPU in one clock during instruction fetch.

(4) RAM

This is a 24/16/8 KB RAM mapped to addresses 3FF9000H to 3FFEFFFH/3FFB000H to 3FFEFFFH/ 3FFD000H to 3FFEFFFH. It can be accessed from the CPU in one clock during data access.

(5) Interrupt controller (INTC)

This controller handles hardware interrupt requests (NMI, INTP02, INTP05, INTP07 to INTP11, INTP14 to INTP16) from on-chip peripheral hardware and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiplexed servicing control can be performed.

(6) Clock generator (CG)

A main clock oscillator and subclock oscillator are provided and generate the main clock oscillation frequency (f_X) and subclock frequency (f_{XT}), respectively. There are two modes: In the clock-through mode, f_X is used as the main clock frequency (f_{XX}) as is. In the PLL mode, f_X is used multiplied by 8.

The CPU clock frequency (f_{CPU}) can be selected from among f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$, $f_{XX}/16$, $f_{XX}/32$, and f_{XT}.

(7) Internal oscillator

An internal oscillator is provided on chip. The oscillation frequency is 220 kHz (TYP). The internal oscillator supplies the clock for watchdog timer 2 and timer M.

(8) Timer/counter

Four-channel 16-bit timer/event counter AA (TAA), one-channel 16-bit timer/event counter AB (TAB)[Note], one-channel 16-bit timer/event counter T (TMT), and four-channel 16-bit interval timer M (TMM) are provided on chip. The motor control function can be realized using TAB1 and TAA4 in combination.

Note V850ES/JE3-H only

3.2.1 Program register set

The program registers include general-purpose registers and a program counter.

(1) General-purpose registers (r0 to r31)

Thirty-two general-purpose registers, r0 to r31, are available. Any of these registers can be used to store a data variable or an address variable.

However, r0 and r30 are implicitly used by instructions and care must be exercised when these registers are used. r0 always holds 0 and is used for an operation that uses 0 or addressing of offset 0. r30 is used by the SLD and SST instructions as a base pointer when these instructions access the memory. r1, r3 to r5, and r31 are implicitly used by the assembler and C compiler. When using these registers, save their contents for protection, and then restore the contents after using the registers. r2 is sometimes used by the real-time OS. If the real-time OS does not use r2, it can be used as a register for variables.

Table 3-1. Program Registers

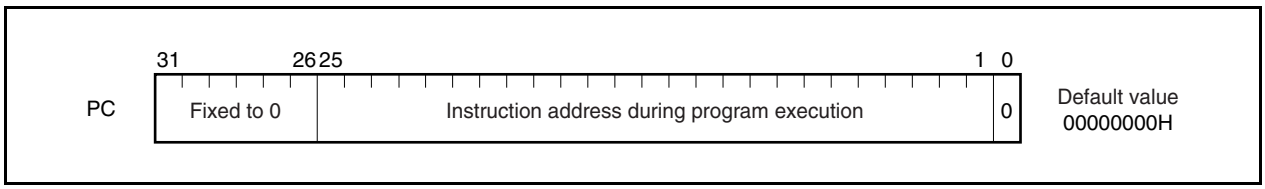
Name	Usage	Operation
r0	Zero register	Always holds 0.
r1	Assembler-reserved register	Used as working register to create 32-bit immediate data
r2	Register for address/data variable (if real-time OS does not use r2)	
r3	Stack pointer	Used to create a stack frame when a function is called
r4	Global pointer	Used to access a global variable in the data area
r5	Text pointer	Used as register that indicates the beginning of a text area (area where program codes are located)
r6 to r29	Register for address/data variable	
r30	Element pointer	Used as base pointer to access memory
r31	Link pointer	Used when the compiler calls a function
PC	Program counter	Holds the instruction address during program execution

Remark For further details on the r1, r3 to r5, and r31 that are used in the assembler and C compiler, refer to the **CA850 (C Compiler Package) Assembly Language User's Manual**.

(2) Program counter (PC)

The program counter holds the instruction address during program execution. The lower 32 bits of this register are valid. Bits 31 to 26 are fixed to 0. A carry from bit 25 to 26 is ignored even if it occurs.

Bit 0 is fixed to 0. This means that execution cannot branch to an odd address.



4.3 Port Configuration

Table 4-4. Port Configuration (V850ES/JC3-H (40 pin))

Item	Configuration
Control register	Port n mode register (PMn: n = 0, 3 to 5, 7, 9, DL) Port n mode control register (PMCn: n = 0, 3 to 5, 9, DL) Port n function control register (PFCn: n = 0, 3 to 5, 9) Port n function control expansion register (PFCEn: n = 0, 3 to 5, 9) Port n function register (PFn: n = 0, 3 to 5)
Ports	I/O: 25

Table 4-5. Port Configuration (V850ES/JC3-H (48 pin))

Item	Configuration
Control register	Port n mode register (PMn: n = 0, 1, 3 to 5, 7, 9, DL) Port n mode control register (PMCn: n = 0, 3 to 5, 9, DL) Port n function control register (PFCn: n = 0, 3 to 5, 9) Port n function control expansion register (PFCEn: n = 0, 3 to 5, 9) Port n function register (PFn: n = 0, 3 to 5, 9)
Ports	I/O: 32

Table 4-6. Port Configuration (V850ES/JE3-H)

Item	Configuration
Control register	Port n mode register (PMn: n = 0, 1, 3 to 7, 9, DL) Port n mode control register (PMCn: n = 0, 3 to 6, 9, DL) Port n function control register (PFCn: n = 0, 2 to 6, 9) Port n function control expansion register (PFCEn: n = 0, 3 to 6, 9) Port n function register (PFn: n = 0, 3 to 6, 9)
Ports	I/O: 45

(6) Port 0 alternate function specifications

PFCE03	PFC03	Specification of P03 pin alternate function
0	0	INTP02 input
0	1	ADTRG input
1	0	UCLK input
1	1	Setting prohibited

(7) Port 0 function register (PF0)

(a) V850ES/JC3-H

After reset: 00H R/W Address: FFFFFC60H

	7	6	5	4	3	2	1	0
PF0	0	0	0	0	PF03	0	0	0

PF03	Control of normal output or N-ch open-drain output
0	Normal output
1	N-ch open-drain output

(b) V850ES/JE3-H

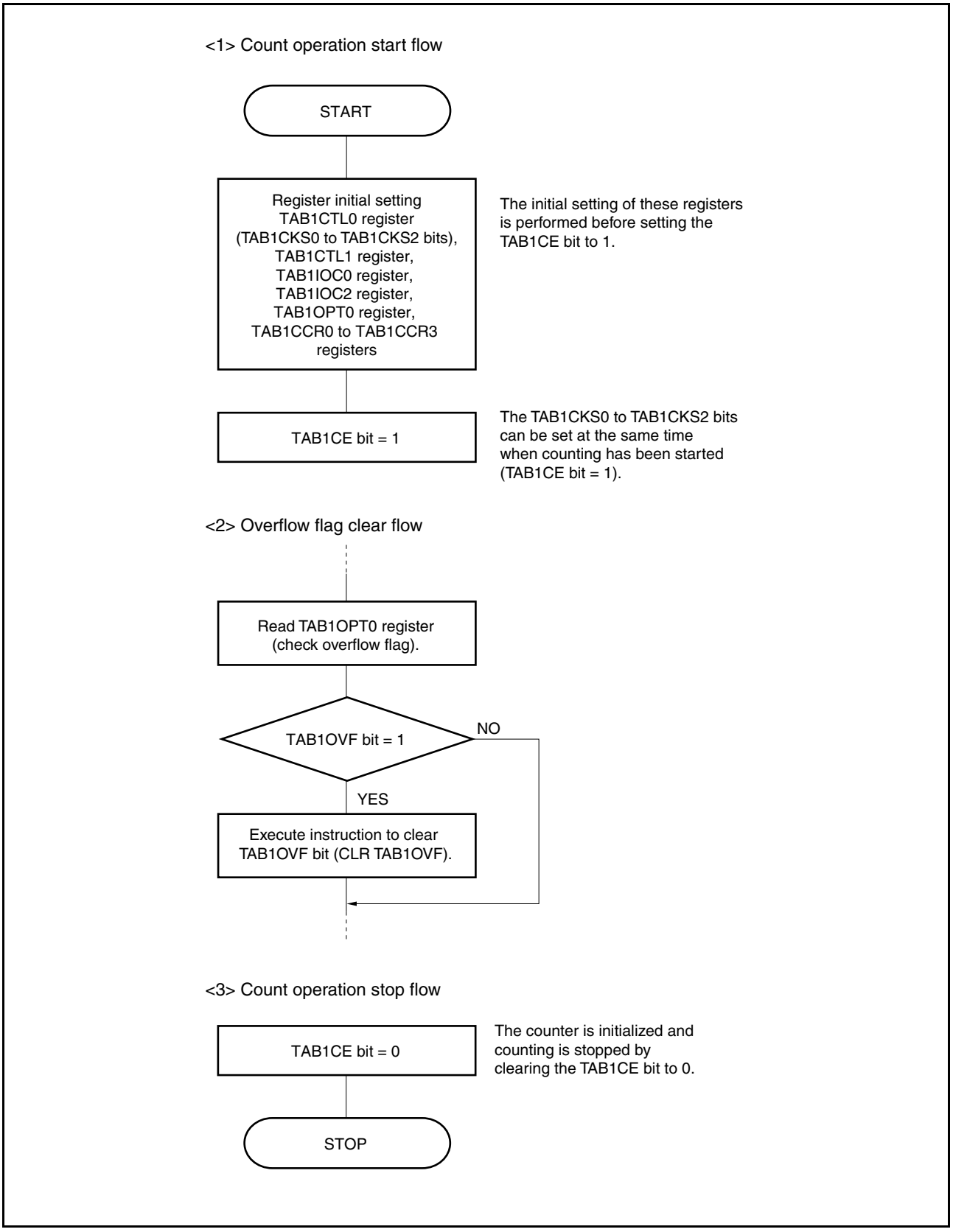
After reset: 00H R/W Address: FFFFFC60H

	7	6	5	4	3	2	1	0
PF0	0	0	0	0	PF03	PF02	0	0

PF0n	Control of normal output or N-ch open-drain output (n = 2, 3)
0	Normal output
1	N-ch open-drain output

Caution When an output pin is pulled up to EV_{DD} or higher, be sure to set the PF0n bit to 1.

Figure 7-32. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)



(5) TMT0 I/O control register 1 (TT0IOC1)

The TT0IOC1 register is an 8-bit register that controls the valid edge for the capture trigger input signals (TIT00, TIT01 pins).

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF604H

	7	6	5	4	3	2	1	0
TT0IOC1	0	0	0	0	TT0IS3	TT0IS2	TT0IS1	TT0IS0

TT0IS3	TT0IS2	Capture trigger input signal (TIT01 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TT0IS1	TT0IS0	Capture trigger input signal (TIT00 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions 1. Rewrite the TT0IS3 to TT0IS0 bits when the TT0CTL0.TT0CE bit = 0. (The same value can be written when the TT0CE bit = 1.) If rewriting was mistakenly performed, clear the TT0CE bit to 0 and then set the bits again.

2. The TT0IS3 and TT0IS2 bits are valid only in the free-running timer mode (only when the TT0OPT0.TT0CCS1 bit = 1) and the pulse width measurement mode. In all other modes, a capture operation is not performed.

The TT0IS1 and TT0IS0 bits are valid only in the free-running timer mode (only when the TT0OPT0. TT0CCS0 bit = 1) and the pulse width measurement mode. In all other modes, a capture operation is not performed.

Figure 8-41. Register Setting in Pulse Width Measurement Mode (2/2)

(e) TMT0 option register 0 (TT0OPT0)

			TT0CCS1	TT0CCS0				TT0OVF
TT0OPT0	0	0	0	0	0	0	0	0/1

Overflow flag

(f) TMT0 counter read buffer register (TT0CNT)

The value of the 16-bit counter can be read by reading the TT0CNT register.

(g) TMT0 capture/compare registers 0 and 1 (TT0CCR0 and TT0CCR1)

These registers store the count value of the 16-bit counter when the valid edge input to the TIT00 and TIT01 pins is detected.

Remark TMT0 control register 2 (TT0CTL2), TMT0 I/O control register 0 (TT0IOC0), TMT0 I/O control register 3 (TT0IOC3), TMT0 option register 1 (TT0OPT1), and TMT0 counter write register (TT0TCW) are not used in the pulse width measurement mode.

(12) Compare voltage generation DAC

This compare voltage generation DAC is connected between AV_{REF0} and AV_{SS} and generates a voltage for comparison with the analog input signal.

(13) ANI0 to ANI9 pins

These are analog input pins for the 10 A/D converter channels and are used to input analog signals to be converted into digital signals. Pins other than the one selected as the analog input by the ADA0S register can be used as input port pins.

Caution Make sure that the voltages input to the ANI0 to ANI9 pins do not exceed the rated values. In particular if a voltage of AV_{REF0} or higher is input to a channel, the conversion value of that channel becomes undefined, and the conversion values of the other channels may also be affected.

(14) AV_{REF0} pin

This is the pin used to input the reference voltage of the A/D converter. Always make the potential at this pin the same as that at the V_{DD} pin even when the A/D converter is not used.

The signals input to the ANI0 to ANI9 pins are converted to digital signals based on the voltage applied between the AV_{REF0} and AV_{SS} pins.

(15) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always make the potential at this pin the same as that at the V_{SS} pin even when the A/D converter is not used.

- Reception error causes

Error Flag	Reception Error	Cause
UCnPE	Parity error	Received parity bit does not match the setting
UCnFE	Framing error	Stop bit not detected
UCnOVE	Overrun error	Reception of next data completed before data was read from receive buffer

When reception errors occur, perform the following procedures depending upon the kind of error.

- Parity error
  If false data is received due to problems such as noise in the reception line, discard the received data and retransmit.

- Framing error
  A baud rate error may have occurred between the reception side and transmission side or the start bit may have been erroneously detected. Since this is a fatal error for the communication format, check the operation stop in the transmission side, perform initialization processing each other, and then start the communication again.

- Overrun error
  Since the next reception is completed before reading receive data, 1 frame of data is discarded. If this data was needed, do a retransmission.

Caution If a receive error interrupt occurs during continuous reception, read the contents of the UCnSTR register must be read before the next reception is completed, then perform error processing.

CSIFn includes the following hardware.

Table 17-1. Configuration of CSIFn

Item	Configuration
Registers	CSIFn receive data register (CFnRX) CSIFn transmit data register (CFnTX) CSIFn control register 0 (CFnCTL0) CSIFn control register 1 (CFnCTL1) CSIFn control register 2 (CFnCTL2) CSIFn status register (CFnSTR)

(1) CSIFn receive data register (CFnRX)

The CFnRX register is a 16-bit buffer register that holds receive data.

This register is read-only, in 16-bit units.

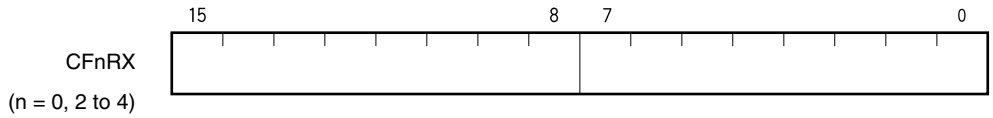
The receive operation is started by reading the CFnRX register in the reception enabled status.

If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CFnRXL register.

Reset sets this register to 0000H.

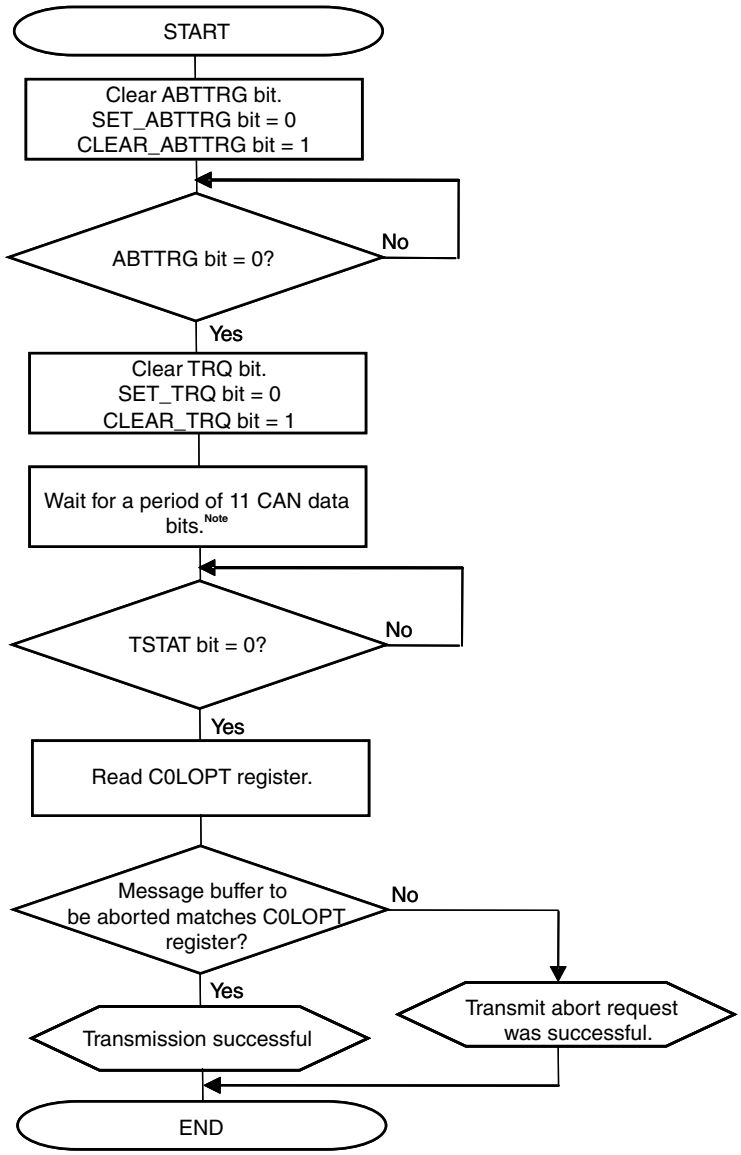
In addition to reset input, the CFnRX register can be initialized by clearing (to 0) the CFnPWR bit of the CFnCTL0 register.

After reset: 0000H R Address: CF0RX FFFFFD04H, CF2RX FFFFFD24H,
CF3RX FFFFFD34H[Note], CF4RX FFFFFD44H



Note V850ES/JC3-H (48 pin), V850ES/JE3-H only

Figure 19-47. Transmission Abort Processing Except for ABT Transmission (Normal Operation Mode with ABT)



Note During a period of a total of 11 bits, 3 bits of interframe space and 8 bits of suspend transmission, the transmission request may have already been acknowledged by the protocol layer. Consequently, transmission may not be aborted but started even if the TRQ bit is cleared.

Cautions 1. Execute transmission abort processing by clearing the TRQ bit, not the RDY bit.

2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.

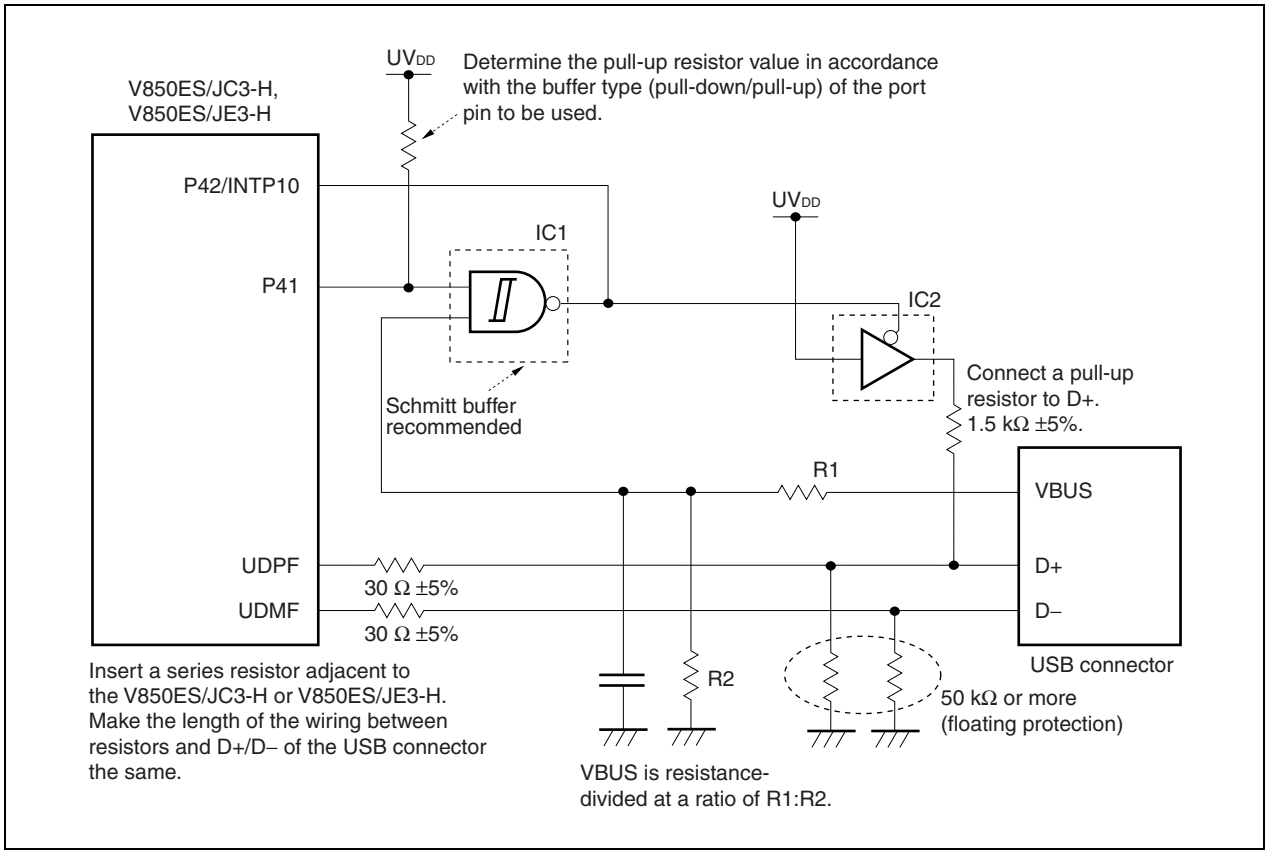
3. The TSTAT bit can be periodically checked by a user application or can be checked after the transmit completion interrupt.

4. Do not execute a new transmission request including in the other message buffers while transmission abort processing is in progress.

5. If data of the same message buffer are successively transmitted or if only one message buffer is used, judgments whether transmission has been successfully executed or failed may contradict. In such a case, make a judgment by using the history information of the C0TGPT register.

20.3.2 Connection configuration

Figure 20-3. Example of USB Function Controller Connection



(1) Series resistor connection to D+/D–

Connect series resistors of 30 Ω ±5% to the D+/D– pins (UFDP, UFDM) of the USB function controller in the V850ES/JC3-H and V850ES/JE3-H. If they are not connected, the impedance rating cannot be satisfied and the output waveform may be disturbed.

Allocate the series resistors adjacent to the V850ES/JC3-H or V850ES/JE3-H, and make the length of the wiring between the series resistors and the USB connectors the same, to make the impedance of D+ and D– equal (a differential with 90 Ω ±5% is recommended).

(2) Pull-up control of D+

Because the function controller of the V850ES/JC3-H and V850ES/JE3-H is fixed to full speed (FS), be sure to pull up the D+ pin (UFDP) by 1.5 kΩ ±5% to UV_{DD}.

To disable a connection report (D+ pull up) to the USB host/HUB (such as during high priority servicing or initialization), control the pull-up resistor of D+ via a general-purpose port in the system. For a circuit such as the one shown in Figure 20-3, control the pull-up control signal and the VBUS input signal of the D+ pin by using a general-purpose port and the USB cable VBUS (AND circuit). In Figure 20-3, if the general-purpose port is low level, pulling up of D+ is prohibited.

For the IC2 in Figure 20-3, use an IC to which voltage can be applied when the system power is off.

20.6.2 External bus control registers

(1) Data wait control register 0 (DWC0)

To realize interfacing with a low-speed memory or I/O, up to seven data wait states can be inserted in the bus cycle that is executed for each USB space.

The number of wait states can be programmed by using the DWC0 register. Immediately after system reset, 7 data wait states are inserted for all the blocks.

The DWC0 register can be read or written in 16-bit units.

Reset sets this register to 7777H.

Cautions 1. The internal ROM and internal RAM areas are not subject to programmable wait, and are always accessed without a wait state. The on-chip peripheral I/O area is also not subject to programmable wait, and only wait control from each peripheral function is performed.

2. Write to the DWC0 register after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the DWC0 register are complete.

After reset: 7777H R/W Address: FFFFF484H

	15	14	13	12	11	10	9	8
DWC0	0	1	1	1	0	1	1	1

7	6	5	4	3	2	1	0
0	DW12Note	DW11Note	DW10Note	0	1	1	1

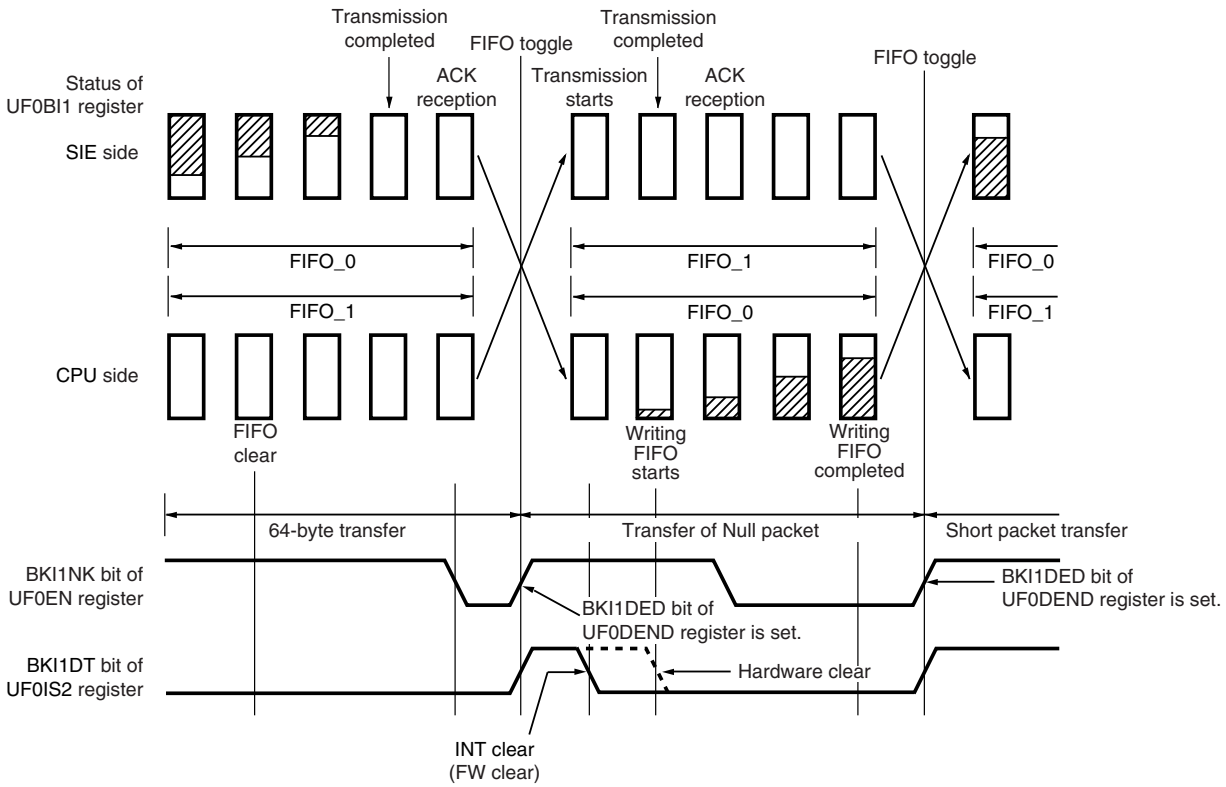
DW12	DW11	DW10	Number of wait states inserted in USBn space
0	0	0	None
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Note The DW12 to DW10 bits set wait of access to the USB function area.
It is recommended to set the DW12 to DW10 bits to 001B (1 wait).

Caution Be sure to clear bits 15, 11, 7, and 3 to “0”.

Figure 20-9. Operation of UF0BI1 Register (3/3)

(c) Operation example 3



(10) UF0 bulk-in 2 register (UF0BI2)

The UF0BI2 register is a 64-byte × 2 FIFO that stores data for Endpoint3. This register consists of two banks of 64-byte FIFOs each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when no data is in the FIFO on the SIE side (counter value = 0) and when the FIFO on the CPU side is correctly written (FIFO full or BKI2DED bit = 1).

This register is write-only, in 8-bit units. When this register is read, 00H is read.

The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint3 only when the BKI2NK bit of the UF0EN register is set to 1 (when NAK is not transmitted). The address at which data is to be written or read is managed by the hardware. Therefore, FW can transmit data to the host only by writing the data to the UF0BI2 register sequentially. A short packet is transmitted when data is written to the UF0BI2 register and the BKI2DED bit of the UF0DEND register is set to 1 (BKIN2 bit of UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0BI2 register is cleared and the BKI2DED bit of the UF0DEND register is set to 1 (BKIN2 bit of the UF0EPS0 register = 1 (data exists)). When the data is transmitted correctly, a FIFO toggle operation occurs. The BKI2DT bit of the UF0IS2 register is set to 1, and an interrupt request is generated for the CPU. An interrupt request or DMA request can be selected by using the DQBI2MS bit of the UF0IDR register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BI2	BKI27	BKI26	BKI25	BKI24	BKI23	BKI22	BKI21	BKI20	00200112H	Undefined

Bit position	Bit name	Function
7 to 0	BKI27 to BKI20	These bits store data for Endpoint3.

The operation of the UF0BI2 register is illustrated below.

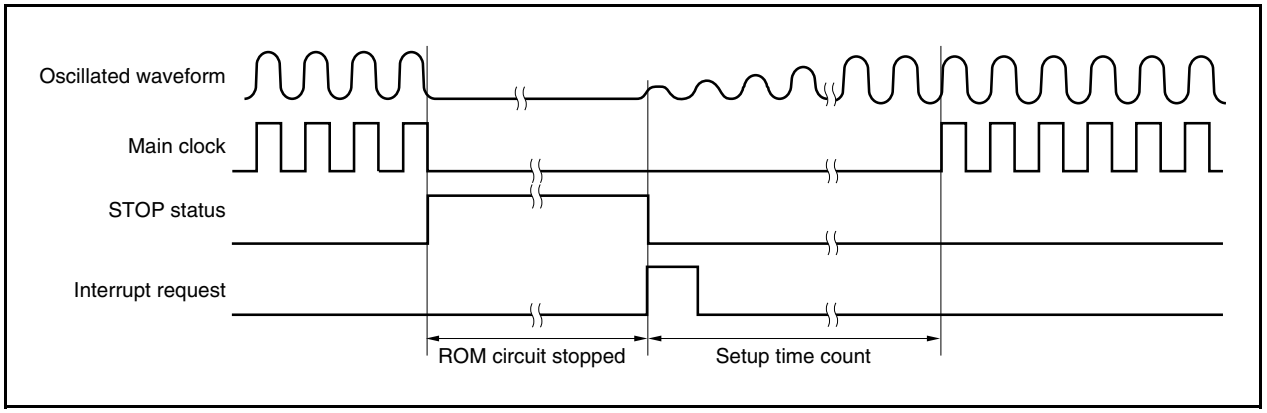
24.6.3 Securing oscillation stabilization time when releasing STOP mode

Secure the oscillation stabilization time for the main clock oscillator after releasing the STOP mode because the operation of the main clock oscillator stops after STOP mode is set.

(1) Releasing STOP mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

Secure the oscillation stabilization time by setting the OSTS register.

When the releasing source is generated, the dedicated internal timer starts counting according to the OSTS register setting. When it overflows, the normal operation mode is restored.



(2) Release by reset

This operation is the same as that of a normal reset.

The oscillation stabilization time is the initial value of the OSTS register, 2^{16}/fx.

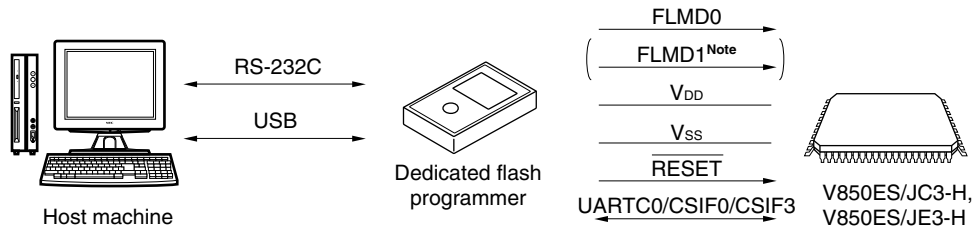
30.4 Rewriting by Dedicated Flash Programmer

The flash memory can be rewritten by using a dedicated flash programmer after the V850ES/JC3-H and V850ES/JE3-H are mounted on the target system (on-board programming). The flash memory can also be rewritten before the device is mounted on the target system (off-board programming) by using a dedicated program adapter (FA series).

30.4.1 Programming environment

The following shows the environment required for writing programs to the flash memory of the V850ES/JC3-H and V850ES/JE3-H.

Figure 30-2. Environment Required for Writing Programs to Flash Memory



Note Connect the FLMD1 pin to the flash programmer or connect to GND via a pull-down resistor on the board.

A host machine is required for controlling the dedicated flash programmer.

UARTC0, CSIF0, or CSIF3 is used for the interface between the dedicated flash programmer and the V850ES/JC3-H and V850ES/JE3-H to perform writing, erasing, etc. A dedicated program adapter (FA series) required for off-board writing.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 30-8. Wiring of V850ES/JE3-H Flash Writing Adapters (1/3)

Pin No.	Pin Name	Recommended Connection
1	AV_{REF0}	Connect to VDD pin of the programmer
2	AV_{SS}	Connect to GND pin of the programmer
3	P10/ANO0	-
4	AV_{REF1}	Connect to VDD pin of the programmer
5	V_{DD}	Connect to VDD pin of the programmer
6	REGC	Connect the REGC pin to GND via 4.7 μF capacitor
7	V_{SS}	Connect to GND pin of the programmer
8	X1	Connect to 3 to 6 MHz Resonator
9	X2	Connect to 3 to 6 MHz Resonator
10	$\overline{\text{RESET}}$	Connect to RESET (output) pin of the programmer
11	XT1	Connect to GND pin of the programmer
12	XT2	-
13	P60/TOAB1T1/TOAB11/TIAB11	-
14	P61/TOAB1B1/TIAB10/TOAB10	-
15	P62/TOAB1T2/TOAB12/TIAB12	-
16	P63/TOAB1B2/TRGAB1	-
17	P64/TOAB1T3/TOAB13/TIAB13	-
18	P65/TOAB1B3/EVTAB1	Connect to GND pin of the programmer
19	P02/NMI	-
20	P03/INTP02/ADTRG/UCLK	-
21	UDMF	-
22	UDPF	-
23	UV_{DD}	Connect to VDD pin of the programmer
24	EV_{DD}	Connect to VDD pin of the programmer
25	V_{SS}	Connect to GND pin of the programmer
26	P34/TIAA10/TOAA10/TOAA1OFF/INTP09	-
27	P36/TXDC3/SCL00/CTXD0[Note2]	-
28	P37/RXDC3/SDA00/CRXD0[Note2]	-
29	P30/TXDC0/SOF4/INTP07	When UART (UARTC0) is used : connect to RxD (input) pin of the programmer When UART (UARTC0) is not used : pull-down[Note1]
30	P31/RXDC0/SIF4/INTP08	When UART (UARTC0) is used : connect to TxD (output) pin of the programmer When UART (UARTC0) is not used : pull-down[Note1]
31	P32/ASCKC0/$\overline{\text{SCKF4}}$/TIAA00/TOAA00	-
32	P40/SIF0/TXDC4/SDA01	When CSI (CSIF0) is used : connect to SO (output) pin of the programmer When CSI (CSIF0)is not used : pull-down[Note]

Notes1. Independently connect to V_{SS} or V_{DD} via a resistor.

2. μPD70F3825 only

(29/34)

Symbol	Name	Unit	Page
UF0CIE131	UF0 configuration/interface/endpoint descriptor register 131	USBF	1118
UF0CIE132	UF0 configuration/interface/endpoint descriptor register 132	USBF	1118
UF0CIE133	UF0 configuration/interface/endpoint descriptor register 133	USBF	1118
UF0CIE134	UF0 configuration/interface/endpoint descriptor register 134	USBF	1118
UF0CIE135	UF0 configuration/interface/endpoint descriptor register 135	USBF	1118
UF0CIE136	UF0 configuration/interface/endpoint descriptor register 136	USBF	1118
UF0CIE137	UF0 configuration/interface/endpoint descriptor register 137	USBF	1118
UF0CIE138	UF0 configuration/interface/endpoint descriptor register 138	USBF	1118
UF0CIE139	UF0 configuration/interface/endpoint descriptor register 139	USBF	1118
UF0CIE140	UF0 configuration/interface/endpoint descriptor register 140	USBF	1118
UF0CIE141	UF0 configuration/interface/endpoint descriptor register 141	USBF	1118
UF0CIE142	UF0 configuration/interface/endpoint descriptor register 142	USBF	1118
UF0CIE143	UF0 configuration/interface/endpoint descriptor register 143	USBF	1118
UF0CIE144	UF0 configuration/interface/endpoint descriptor register 144	USBF	1118
UF0CIE145	UF0 configuration/interface/endpoint descriptor register 145	USBF	1118
UF0CIE146	UF0 configuration/interface/endpoint descriptor register 146	USBF	1118
UF0CIE147	UF0 configuration/interface/endpoint descriptor register 147	USBF	1118
UF0CIE148	UF0 configuration/interface/endpoint descriptor register 148	USBF	1118
UF0CIE149	UF0 configuration/interface/endpoint descriptor register 149	USBF	1118
UF0CIE150	UF0 configuration/interface/endpoint descriptor register 150	USBF	1118
UF0CIE151	UF0 configuration/interface/endpoint descriptor register 151	USBF	1118
UF0CIE152	UF0 configuration/interface/endpoint descriptor register 152	USBF	1118
UF0CIE153	UF0 configuration/interface/endpoint descriptor register 153	USBF	1118
UF0CIE154	UF0 configuration/interface/endpoint descriptor register 154	USBF	1118
UF0CIE155	UF0 configuration/interface/endpoint descriptor register 155	USBF	1118
UF0CIE156	UF0 configuration/interface/endpoint descriptor register 156	USBF	1118
UF0CIE157	UF0 configuration/interface/endpoint descriptor register 157	USBF	1118
UF0CIE158	UF0 configuration/interface/endpoint descriptor register 158	USBF	1118
UF0CIE159	UF0 configuration/interface/endpoint descriptor register 159	USBF	1118
UF0CIE160	UF0 configuration/interface/endpoint descriptor register 160	USBF	1118
UF0CIE161	UF0 configuration/interface/endpoint descriptor register 161	USBF	1118
UF0CIE162	UF0 configuration/interface/endpoint descriptor register 162	USBF	1118
UF0CIE163	UF0 configuration/interface/endpoint descriptor register 163	USBF	1118
UF0CIE164	UF0 configuration/interface/endpoint descriptor register 164	USBF	1118
UF0CIE165	UF0 configuration/interface/endpoint descriptor register 165	USBF	1118
UF0CIE166	UF0 configuration/interface/endpoint descriptor register 166	USBF	1118
UF0CIE167	UF0 configuration/interface/endpoint descriptor register 167	USBF	1118
UF0CIE168	UF0 configuration/interface/endpoint descriptor register 168	USBF	1118
UF0CIE169	UF0 configuration/interface/endpoint descriptor register 169	USBF	1118
UF0CIE170	UF0 configuration/interface/endpoint descriptor register 170	USBF	1118