



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	22
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f258-e-so

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

## EXAMPLE 4-2: INSTRUCTION PIPELINE FLOW

	TCY0	Tcy1	TCY2	TCY3	TCY4	TCY5			
1. MOVLW 55h	Fetch 1	Execute 1		_					
2. MOVWF PORTB		Fetch 2	Execute 2		_				
3. BRA SUB_1			Fetch 3	Execute 3					
4. BSF PORTA, BIT3 (	Forced NOP)			Fetch 4	Flush				
5. Instruction @ addre	ss SUB_1				Fetch SUB_1	Execute SUB_1			
Note:   All instructions are single cycle, except for any program branches. These take two cycles, since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.									

#### EXAMPLE 4-3: INSTRUCTIONS IN PROGRAM MEMORY

Instruction	Opcode	Memory	Address
—			000007h
MOVLW 055h	0E55h	55h	000008h
		0Eh	000009h
GOTO 000006h	0EF03h, 0F000h	03h	00000Ah
		0EFh	00000Bh
		00h	00000Ch
		0F0h	00000Dh
MOVFF 123h, 456h	0C123h, 0F456h	23h	00000Eh
		0C1h	00000Fh
		56h	000010h
		0F4h	000011h
			000012h





# 6.2 Control Registers

Several control registers are used in conjunction with the TBLRD and TBLWT instructions. These include the:

- EECON1 register
- EECON2 register
- TABLAT register
- TBLPTR registers

#### 6.2.1 EECON1 AND EECON2 REGISTERS

EECON1 is the control register for memory accesses.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the memory write and erase sequences.

Control bit EEPGD determines if the access will be a program or data EEPROM memory access. When clear, any subsequent operations will operate on the data EEPROM memory. When set, any subsequent operations will operate on the program memory.

Control bit CFGS determines if the access will be to the Configuration/Calibration registers or to program memory/data EEPROM memory. When set, subsequent operations will operate on Configuration registers regardless of EEPGD (see Section 24.0 "Special Features of the CPU"). When clear, memory selection access is determined by EEPGD. The FREE bit, when set, will allow a program memory erase operation. When the FREE bit is set, the erase operation is initiated on the next WR command. When FREE is clear, only writes are enabled.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a MCLR Reset or a WDT Time-out Reset during normal operation. In these situations, the user can check the WRERR bit and rewrite the location. It is necessary to reload the data and address registers (EEDATA and EEADR) due to Reset values of zero.

Control bits,  $\overline{\text{RD}}$  and  $\overline{\text{WR}}$ , initiate read and write operations, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at the completion of the read or write operation. The inability to clear the  $\overline{\text{WR}}$  bit in software prevents the accidental or premature termination of a write operation. The  $\overline{\text{RD}}$ bit cannot be set when accessing program memory (EEPGD = 1).



#### REGISTER 17-2: SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE)

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| WCOL  | SSPOV | SSPEN | CKP   | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit 7 |       |       |       |       |       |       | bit 0 |

bit 7 WCOL: Write Collision Detect bit (Transmit mode only)

 1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)

0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit

#### SPI Slave mode:

- 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF even if only transmitting data to avoid setting overflow (must be cleared in software).
- 0 = No overflow
  - **Note:** In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

#### bit 5 SSPEN: Synchronous Serial Port Enable bit

- 1 = Enables serial port and configures SCK, SDO, SDI and  $\overline{SS}$  as serial port pins
- 0 = Disables serial port and configures these pins as I/O port pins

Note: When enabled, these pins must be properly configured as input or output.

- bit 4 CKP: Clock Polarity Select bit
  - 1 = Idle state for clock is a high level
  - 0 = Idle state for clock is a low level
- bit 3-0 SSPM3:SSPM0: Synchronous Serial Port Mode Select bits
  - 0101 = SPI Slave mode, clock = SCK pin,  $\overline{SS}$  pin control disabled,  $\overline{SS}$  can be used as I/O pin 0100 = SPI Slave mode, clock = SCK pin,  $\overline{SS}$  pin control enabled
  - 0011 = SPI Master mode, clock = TMR2 output/2
  - 0010 = SPI Master mode, clock = Fosc/64
  - 0001 = SPI Master mode, clock = Fosc/16
  - 0000 = SPI Master mode, clock = Fosc/4
    - **Note:** Bit combinations not specifically listed here are either reserved or implemented in  $I^2C$  mode only.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bi	t, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

#### 17.3.8 SLEEP OPERATION

In Master mode, all module clocks are halted and the transmission/reception will remain in that state until the device wakes from Sleep. After the device returns to normal mode, the module will continue to transmit/ receive data.

In Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device from Sleep.

#### 17.3.9 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

#### 17.3.10 BUS MODE COMPATIBILITY

Table 17-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

TABLE 17-1. SPIT BUS MODES	TA	BLE	17-1:	SPI™	BUS	MODES
----------------------------	----	-----	-------	------	-----	-------

Standard SPI Mode	Control Bits State				
Terminology	СКР	СКЕ			
0, 0	0	1			
0, 1	0	0			
1, 0	1	1			
1, 1	1	0			

There is also an SMP bit which controls when the data is sampled.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	<b>INTOIE</b>	RBIE	TMR0IF	<b>INT0IF</b>	RBIF	0000 000x	0000 000u
PIR1	PSPIF <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
TRISC	PORTC Da	ta Direction F	Register						1111 1111	1111 1111
TRISA	—	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	-111 1111	-111 1111
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register									uuuu uuuu
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
SSPSTAT	SMP	CKE	D/A	Р	S	R/W	UA	BF	0000 0000	0000 0000

#### TABLE 17-2: REGISTERS ASSOCIATED WITH SPI™ OPERATION

**Legend:** x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the MSSP in SPI<sup>TM</sup> mode.**Note 1:**These registers or register bits are not implemented on the PIC18F248 and PIC18F258 and read as '0's.

# 17.4 I<sup>2</sup>C Mode

The MSSP module in I<sup>2</sup>C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer:

- Serial clock (SCL) RC3/SCK/SCL
- Serial data (SDA) RC4/SDI/SDA

The user must configure these pins as inputs or outputs through the TRISC<4:3> bits.

#### FIGURE 17-7: MSSP BLOCK DIAGRAM (I<sup>2</sup>C™ MODE)



#### 17.4.1 REGISTERS

The MSSP module has six registers for  $\mathsf{I}^2\mathsf{C}$  operation. These are:

- MSSP Control Register 1 (SSPCON1)
- MSSP Control Register 2 (SSPCON2)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- MSSP Shift Register (SSPSR) Not directly accessible
- MSSP Address Register (SSPADD)

SSPCON1, SSPCON2 and SSPSTAT are the control and status registers in I<sup>2</sup>C mode operation. The SSPCON1 and SSPCON2 registers are readable and writable. The lower 6 bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

SSPADD register holds the slave device address when the SSP is configured in  $I^2C$  Slave mode. When the SSP is configured in Master mode, the lower seven bits of SSPADD act as the Baud Rate Generator reload value.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not doublebuffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

# PIC18FXX8

NOTES:

# 19.0 CAN MODULE

#### 19.1 Overview

The Controller Area Network (CAN) module is a serial interface, useful for communicating with other peripherals or microcontroller devices. This interface/protocol was designed to allow communications within noisy environments.

The CAN module is a communication controller, implementing the CAN 2.0 A/B protocol as defined in the BOSCH specification. The module will support CAN 1.2, CAN 2.0A, CAN 2.0B Passive and CAN 2.0B Active versions of the protocol. The module implementation is a full CAN system. The CAN specification is not covered within this data sheet. The reader may refer to the BOSCH CAN specification for further details.

The module features are as follows:

- Complies with ISO CAN Conformance Test
- Implementation of the CAN protocol CAN 1.2, CAN 2.0A and CAN 2.0B
- · Standard and extended data frames
- 0-8 bytes data length
- Programmable bit rate up to 1 Mbit/sec
- Support for remote frames
- Double-buffered receiver with two prioritized received message storage buffers
- 6 full (standard/extended identifier) acceptance filters, 2 associated with the high priority receive buffer and 4 associated with the low priority receive buffer
- 2 full acceptance filter masks, one each associated with the high and low priority receive buffers
- Three transmit buffers with application specified prioritization and abort capability
- Programmable wake-up functionality with integrated low-pass filter
- Programmable Loopback mode supports self-test operation
- Signaling via interrupt capabilities for all CAN receiver and transmitter error states
- Programmable clock source
- Programmable link to timer module for time-stamping and network synchronization
- Low-power Sleep mode

## 19.1.1 OVERVIEW OF THE MODULE

The CAN bus module consists of a protocol engine and message buffering and control. The CAN protocol engine handles all functions for receiving and transmitting messages on the CAN bus. Messages are transmitted by first loading the appropriate data registers. Status and errors can be checked by reading the appropriate registers. Any message detected on the CAN bus is checked for errors and then matched against filters to see if it should be received and stored in one of the 2 receive registers.

The CAN module supports the following frame types:

- Standard Data Frame
- Extended Data Frame
- Remote Frame
- Error Frame
- Overload Frame Reception
- Interframe Space

CAN module uses RB3/CANRX and RB2/CANTX/INT2 pins to interface with CAN bus. In order to configure CANRX and CANTX as CAN interface:

- bit TRISB<3> must be set;
- bit TRISB<2> must be cleared.

#### 19.1.2 TRANSMIT/RECEIVE BUFFERS

The PIC18FXX8 has three transmit and two receive buffers, two acceptance masks (one for each receive buffer) and a total of six acceptance filters. Figure 19-1 is a block diagram of these buffers and their connection to the protocol engine.

#### 19.3.5 LOOPBACK MODE

This mode will allow internal transmission of messages from the transmit buffers to the receive buffers without actually transmitting messages on the CAN bus. This mode can be used in system development and testing. In this mode, the ACK bit is ignored and the device will allow incoming messages from itself, just as if they were coming from another node. The Loopback mode is a silent mode, meaning no messages will be transmitted while in this state, including error flags or Acknowledge signals. The TXCAN pin will revert to port I/O while the device is in this mode. The filters and masks can be used to allow only particular messages to be loaded into the receive registers. The masks can be set to all zeros to provide a mode that accepts all messages. The Loopback mode is activated by setting the mode request bits in the CANCON register.

#### 19.3.6 ERROR RECOGNITION MODE

The module can be set to ignore all errors and receive all message. The Error Recognition mode is activated by setting the RXM<1:0> bits in the RXBnCON registers to '11'. In this mode, all messages, valid or invalid, are received and copied to the receive buffer.

## 19.4 CAN Message Transmission

#### 19.4.1 TRANSMIT BUFFERS

The PIC18FXX8 implements three transmit buffers (Figure 19-2). Each of these buffers occupies 14 bytes of SRAM and are mapped into the device memory map.

For the MCU to have write access to the message buffer, the TXREQ bit must be clear, indicating that the message buffer is clear of any pending message to be transmitted. At a minimum, the TXBnSIDH, TXBnSIDL and TXBnDLC registers must be loaded. If data bytes are present in the message, the TXBnDm registers must also be loaded. If the message is to use extended identifiers, the TXBnEIDm registers must also be loaded and the EXIDE bit set.

Prior to sending the message, the MCU must initialize the TXInE bit to enable or disable the generation of an interrupt when the message is sent. The MCU must also initialize the TXP priority bits (see **Section 19.4.2 "Transmit Priority"**).

#### 19.4.2 TRANSMIT PRIORITY

Transmit priority is a prioritization within the PIC18FXX8 of the pending transmittable messages. This is independent from and not related to any prioritization implicit in the message arbitration scheme built into the CAN protocol. Prior to sending the SOF, the priority of all buffers that are queued for transmission is compared. The transmit buffer with the highest priority will be sent first. If two buffers have the same priority setting, the buffer with the highest buffer number will be sent first. There are four levels of transmit priority. If TXP bits for a particular message buffer are set to '11', that buffer has the highest possible priority. If TXP bits for a particular message buffer are '00', that buffer has the lowest possible priority.

#### FIGURE 19-2: TRANSMIT BUFFER BLOCK DIAGRAM



## 19.5 Message Reception

#### 19.5.1 RECEIVE MESSAGE BUFFERING

The PIC18FXX8 includes two full receive buffers with multiple acceptance filters for each. There is also a separate Message Assembly Buffer (MAB) which acts as a third receive buffer (see Figure 19-4).

#### 19.5.2 RECEIVE BUFFERS

Of the three receive buffers, the MAB is always committed to receiving the next message from the bus. The remaining two receive buffers are called RXB0 and RXB1 and can receive a complete message from the protocol engine. The MCU can access one buffer while the other buffer is available for message reception or holding a previously received message.

The MAB assembles all messages received. These messages will be transferred to the RXBn buffers only if the acceptance filter criteria are met.

Note: The entire contents of the MAB are moved into the receive buffer once a message is accepted. This means that regardless of the type of identifier (standard or extended) and the number of data bytes received, the entire receive buffer is overwritten with the MAB contents. Therefore, the contents of all registers in the buffer must be assumed to have been modified when any message is received.

When a message is moved into either of the receive buffers, the appropriate RXBnIF bit is set. This bit must be cleared by the MCU when it has completed processing the message in the buffer in order to allow a new message to be received into the buffer. This bit provides a positive lockout to ensure that the MCU has finished with the message before the PIC18FXX8 attempts to load a new message into the receive buffer. If the RXBnIE bit is set, an interrupt will be generated to indicate that a valid message has been received.

#### 19.5.3 RECEIVE PRIORITY

RXB0 is the higher priority buffer and has two message acceptance filters associated with it. RXB1 is the lower priority buffer and has four acceptance filters associated with it. The lower number of acceptance filters makes the match on RXB0 more restrictive and implies a higher priority for that buffer. Additionally, the RXB0CON register can be configured such if RXB0 contains a valid message and another valid message is received, an overflow error will not occur and the new message will be moved into RXB1 regardless of the acceptance criteria of RXB1. There are also two programmable acceptance filter masks available, one for each receive buffer (see Section 19.6 "Message Acceptance Filters and Masks").

When a message is received, bits <3:0> of the RXBnCON register will indicate the acceptance filter number that enabled reception and whether the received message is a remote transfer request.

The RXM bits set special Receive modes. Normally, these bits are set to '00' to enable reception of all valid messages as determined by the appropriate acceptance filters. In this case, the determination of whether or not to receive standard or extended messages is determined by the EXIDE bit in the Acceptance Filter register. If the RXM bits are set to '01' or '10', the receiver will accept only messages with standard or extended identifiers, respectively. If an acceptance filter has the EXIDE bit set, such that it does not correspond with the RXM mode, that acceptance filter is rendered useless. These two modes of RXM bits can be used in systems where it is known that only standard or extended messages will be on the bus. If the RXM bits are set to '11', the buffer will receive all messages regardless of the values of the acceptance filters. Also, if a message has an error before the end of frame, that portion of the message assembled in the MAB before the error frame will be loaded into the buffer. This mode has some value in debugging a CAN system and would not be used in an actual system environment.

#### 19.5.4 TIME-STAMPING

The CAN module can be programmed to generate a time-stamp for every message that is received. When enabled, the module generates a capture signal for CCP1 which in turns captures the value of either Timer1 or Timer3. This value can be used as the message time-stamp.

To use the time-stamp capability, the CANCAP bit (CIOCAN<4>) must be set. This replaces the capture input for CCP1 with the signal generated from the CAN module. In addition, CCP1CON<3:0> must be set to '0011' to enable the CCP special event trigger for CAN events.



#### RECEIVE BUFFER BLOCK DIAGRAM



#### 19.7.1 TIME QUANTA

As already mentioned, the time quanta is a fixed unit derived from the oscillator period and baud rate prescaler. Its relationship to TBIT and the nominal bit rate is shown in Example 19-2.

#### EXAMPLE 19-2: CALCULATING TQ, NOMINAL BIT RATE AND NOMINAL BIT TIME

 $TQ (\mu s) = (2 * (BRP + 1))/FOSC (MHz)$ 

TBIT ( $\mu$ s) = TQ ( $\mu$ s) \* number of TQ per bit interval Nominal Bit Rate (bits/s) = 1/TBIT

## CASE 1:

For FOSC = 16 MHz, BRP<5:0> = 00h and Nominal Bit Time = 8 TQ:

 $TQ = (2 * 1)/16 = 0.125 \,\mu s \,(125 \,ns)$ 

TBIT =  $8 * 0.125 = 1 \ \mu s \ (10^{-6} s)$ 

Nominal Bit Rate =  $1/10^{-6} = 10^{6}$  bits/s (1 Mb/s)

## CASE 2:

For Fosc = 20 MHz, BRP<5:0> = 01h and Nominal Bit Time = 8 Tq:

 $Tq = (2 * 2)/20 = 0.2 \ \mu s \ (200 \ ns)$ 

TBIT =  $8 * 0.2 = 1.6 \,\mu s \,(1.6 * 10^{-6} s)$ 

Nominal Bit Rate =  $1/1.6 * 10^{-6}$ s = 625,000 bits/s (625 Kb/s)

#### CASE 3:

For FOSC = 25 MHz, BRP<5:0> = 3Fh and Nominal Bit Time = 25 TQ:  $TQ = (2 * 64)/25 = 5.12 \ \mu s$ TBIT = 25 \* 5.12 = 128 \ \mu s (1.28 \* 10<sup>-4</sup>s) Nominal Bit Rate = 1/1.28 \* 10<sup>-4</sup> = 7813 bits/s (7.8 Kb/s)

The frequencies of the oscillators in the different nodes must be coordinated in order to provide a system wide specified nominal bit time. This means that all oscillators must have a Tosc that is an integral divisor of TQ. It should also be noted that although the number of TQ is programmable from 4 to 25, the usable minimum is 8 TQ. A bit time of less than 8 TQ in length is not ensured to operate correctly.

## 19.7.2 SYNCHRONIZATION SEGMENT

This part of the bit time is used to synchronize the various CAN nodes on the bus. The edge of the input signal is expected to occur during the sync segment. The duration is 1 Tq.

## 19.7.3 PROPAGATION SEGMENT

This part of the bit time is used to compensate for physical delay times within the network. These delay times consist of the signal propagation time on the bus line and the internal delay time of the nodes. The length of the Propagation Segment can be programmed from 1 TQ to 8 TQ by setting the PRSEG2:PRSEG0 bits.

## 19.7.4 PHASE BUFFER SEGMENTS

The phase buffer segments are used to optimally locate the sampling point of the received bit within the nominal bit time. The sampling point occurs between Phase Segment 1 and Phase Segment 2. These segments can be lengthened or shortened by the resynchronization process. The end of Phase Segment 1 determines the sampling point within a bit time. Phase Segment 1 is programmable from 1 Tq to 8 Tq in duration. Phase Segment 2 provides delay before the next transmitted data transition and is also programmable from 1 Tq to 8 Tq in duration. However, due to IPT requirements, the actual minimum length of Phase Segment 2 is 2 Tq or it may be defined to be equal to the greater of Phase Segment 1 or the Information Processing Time (IPT).

#### 19.7.5 SAMPLE POINT

The sample point is the point of time at which the bus level is read and the value of the received bit is determined. The sampling point occurs at the end of Phase Segment 1. If the bit timing is slow and contains many TQ, it is possible to specify multiple sampling of the bus line at the sample point. The value of the received bit is determined to be the value of the majority decision of three values. The three samples are taken at the sample point and twice before, with a time of TQ/2 between each sample.

## 19.7.6 INFORMATION PROCESSING TIME

The Information Processing Time (IPT) is the time segment, starting at the sample point, that is reserved for calculation of the subsequent bit level. The CAN specification defines this time to be less than or equal to 2 Tq. The PIC18FXX8 defines this time to be 2 Tq. Thus, Phase Segment 2 must be at least 2 Tq long.

## 19.8 Synchronization

To compensate for phase shifts between the oscillator frequencies of each of the nodes on the bus, each CAN controller must be able to synchronize to the relevant signal edge of the incoming signal. When an edge in the transmitted data is detected, the logic will compare the location of the edge to the expected time (Sync\_Seg). The circuit will then adjust the values of Phase Segment 1 and Phase Segment 2, as necessary. There are two mechanisms used for synchronization.

#### 19.8.1 HARD SYNCHRONIZATION

Hard synchronization is only done when there is a recessive to dominant edge during a bus Idle condition, indicating the start of a message. After hard synchronization, the bit time counters are restarted with Sync\_Seg. Hard synchronization forces the edge which has occurred to lie within the synchronization segment of the restarted bit time. Due to the rules of synchronization, if a hard synchronization occurs, there will not be a resynchronization within that bit time.

#### 19.8.2 RESYNCHRONIZATION

As a result of resynchronization, Phase Segment 1 may be lengthened or Phase Segment 2 may be shortened. The amount of lengthening or shortening of the phase buffer segments has an upper bound given by the Synchronization Jump Width (SJW). The value of the SJW will be added to Phase Segment 1 (see Figure 19-8) or subtracted from Phase Segment 2 (see Figure 19-9). The SJW is programmable between 1 TQ and 4 TQ.

Clocking information will only be derived from recessive to dominant transitions. The property, that only a fixed maximum number of successive bits have the same value, ensures resynchronization to the bit stream during a frame. The phase error of an edge is given by the position of the edge relative to Sync\_Seg, measured in Tq. The phase error is defined in magnitude of Tq as follows:

- e = 0 if the edge lies within Sync\_Seg.
- e > 0 if the edge lies before the sample point.
- e < 0 if the edge lies after the sample point of the previous bit.

If the magnitude of the phase error is less than or equal to the programmed value of the synchronization jump width, the effect of a resynchronization is the same as that of a hard synchronization.

If the magnitude of the phase error is larger than the synchronization jump width and if the phase error is positive, then Phase Segment 1 is lengthened by an amount equal to the synchronization jump width.

If the magnitude of the phase error is larger than the resynchronization jump width and if the phase error is negative, then Phase Segment 2 is shortened by an amount equal to the synchronization jump width.

#### 19.8.3 SYNCHRONIZATION RULES

- Only one synchronization within one bit time is allowed.
- An edge will be used for synchronization only if the value detected at the previous sample point (previously read bus value) differs from the bus value immediately after the edge.
- All other recessive to dominant edges, fulfilling rules 1 and 2, will be used for resynchronization with the exception that a node transmitting a dominant bit will not perform a resynchronization as a result of a recessive to dominant edge with a positive phase error.

# FIGURE 19-8: LENGTHENING A BIT PERIOD (ADDING SJW TO PHASE SEGMENT 1)



# PIC18FXX8

#### REGISTER 20-2: ADCON1: A/D CONTROL REGISTER 1

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 ADFM: A/D Result Format Select bit

1 = Right justified. Six (6) Most Significant bits of ADRESH are read as '0'.
0 = Left justified. Six (6) Least Significant bits of ADRESL are read as '0'.

#### bit 6 ADCS2: A/D Conversion Clock Select bit (ADCON1 bits in **bold**)

ADCON1 <adcs2></adcs2>	ADCON0 <adcs1:adcs0></adcs1:adcs0>	Clock Conversion				
0	00	Fosc/2				
0	01	Fosc/8				
0	10	Fosc/32				
0	11	FRC (clock derived from the internal A/D RC oscillator)				
1	00	Fosc/4				
1	01	Fosc/16				
1	10	Fosc/64				
1	11	FRC (clock derived from the internal A/D RC oscillator)				

#### bit 5-4 Unimplemented: Read as '0'

bit 3-0 PCFG3:PCFG0: A/D Port Configuration Control bits

PCFG	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	Α	Α	Α	А	Α	Α	А	А	Vdd	Vss	8/0
0001	А	А	А	А	VREF+	А	А	А	AN3	Vss	7/1
0010	D	D	D	Α	Α	А	Α	Α	Vdd	Vss	5/0
0011	D	D	D	А	VREF+	А	А	А	AN3	Vss	4/1
0100	D	D	D	D	Α	D	А	А	Vdd	Vss	3/0
0101	D	D	D	D	VREF+	D	А	А	AN3	Vss	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	Α	Α	Α	А	VREF+	VREF-	А	А	AN3	AN2	6/2
1001	D	D	Α	Α	Α	А	Α	Α	Vdd	Vss	6/0
1010	D	D	Α	Α	VREF+	А	Α	Α	AN3	Vss	5/1
1011	D	D	Α	Α	VREF+	VREF-	Α	Α	AN3	AN2	4/2
1100	D	D	D	Α	VREF+	VREF-	Α	Α	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	А	А	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	А	Vdd	Vss	1/0
1111	D	D	D	D	VREF+	VREF-	D	А	AN3	AN2	1/2

A = Analog input D = Digital I/O

C/R = # of analog input channels/# of A/D voltage references

Note: Shaded cells indicate channels available only on PIC18F4X8 devices.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented	bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

**Note:** On any device Reset, the port pins that are multiplexed with analog functions (ANx) are forced to be analog inputs.

#### 20.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 12 TAD per 10-bit conversion. The source of the A/D conversion clock is software selectable. The seven possible options for TAD are:

- 2 Tosc
- 4 Tosc
- 8 Tosc
- 16 Tosc
- 32 Tosc
- 64 Tosc
- Internal RC oscillator.

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6  $\mu s.$ 

Table 20-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

## 20.3 Configuring Analog Port Pins

The ADCON1, TRISA and TRISE registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the TRIS bits.

- Note 1: When reading the port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.
  - 2: Analog levels on any pin that is defined as a digital input (including the AN4:AN0 pins) may cause the input buffer to consume current that is out of the device's specification.

#### TABLE 20-1: TAD vs. DEVICE OPERATING FREQUENCIES

AD Clock S	Source (TAD)	Device Frequency					
Operation	ADCS2:ADCS0	20 MHz	5 MHz	1.25 MHz	333.33 kHz		
2 Tosc	000	100 ns <sup>(2)</sup>	400 ns <sup>(2)</sup>	1.6 μs	6 µs		
4 Tosc	100	200 ns <sup>(2)</sup>	800 ns <sup>(2)</sup>	3.2 μs	12 µs		
8 Tosc	001	400 ns <sup>(2)</sup>	1.6 μs	6.4 μs	24 μs <sup>(3)</sup>		
16 Tosc	101	800 ns <sup>(2)</sup>	3.2 μs	12.8 μs	48 μs <sup>(3)</sup>		
32 Tosc	010	1.6 μs	6.4 μs	25.6 μs <sup>(3)</sup>	96 μs <sup>(3)</sup>		
64 Tosc	110	3.2 μs	12.8 μs	51.2 μs <sup>(3)</sup>	192 μs <sup>(3)</sup>		
RC	011	2-6 μs <sup>(1)</sup>	2-6 μs <sup>(1)</sup>	2-6 μs <sup>(1)</sup>	2-6 μs <sup>(1)</sup>		

Legend: Shaded cells are outside of recommended range.

**2:** These values violate the minimum required TAD time.

3: For faster conversion times, the selection of another clock source is recommended.

#### TABLE 20-2: TAD vs. DEVICE OPERATING FREQUENCIES (FOR EXTENDED, LF DEVICES)

AD Clock S	Source (TAD)	Device Frequency					
Operation	ADCS2:ADCS0	4 MHz	2 MHz	1.25 MHz	333.33 kHz		
2 Tosc	000	500 ns <sup>(2)</sup>	1.0 μs <sup>(2)</sup>	1.6 μs <sup>(2)</sup>	6 µs		
4 Tosc	100	1.0 μs <sup>(2)</sup>	2.0 μs <sup>(2)</sup>	3.2 μs <sup>(2)</sup>	12 µs		
8 Tosc	001	2.0 μs <sup>(2)</sup>	4.0 μs	6.4 μs	24 μs <sup>(3)</sup>		
16 Tosc	101	4.0 μs <sup>(2)</sup>	8.0 μs	12.8 μs	48 μs <sup>(3)</sup>		
32 Tosc	010	8.0 μs	16.0 μs	25.6 μs <sup>(3)</sup>	96 μs <sup>(3)</sup>		
64 Tosc	110	16.0 μs	32.0 μs	51.2 μs <sup>(3)</sup>	192 μs <sup>(3)</sup>		
RC	011	3-9 μs <sup>(1)</sup>	3-9 μs <sup>(1)</sup>	3-9 μs <sup>(1)</sup>	3-9 μs <sup>(1)</sup>		

Legend: Shaded cells are outside of recommended range.

Note 1: The RC source has a typical TAD time of 6 µs.

- **2:** These values violate the minimum required TAD time.
- 3: For faster conversion times, the selection of another clock source is recommended.

Note 1: The RC source has a typical TAD time of 4  $\mu$ s.





## 22.2 Voltage Reference Accuracy/Error

The full range of voltage reference cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 22-1) keep VREF from approaching the reference source rails. The voltage reference is derived from the reference source; therefore, the VREF output changes with fluctuations in that source. The absolute accuracy of the voltage reference can be found in **Section 27.0 "Electrical Characteristics"**.

## 22.3 Operation During Sleep

When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the CVRCON register are not affected. To minimize current consumption in Sleep mode, the voltage reference should be disabled.

#### 22.4 Effects of a Reset

A device Reset disables the voltage reference by clearing bit CVREN (CVRCON register). This Reset also disconnects the reference from the RA2 pin by clearing bit CVROE (CVRCON register) and selects the high-voltage range by clearing bit CVRR (CVRCON register). The CVRSS value select bits, CVRCON<3:0>, are also cleared.

#### 22.5 Connection Considerations

The voltage reference module operates independently of the comparator module. The output of the reference generator may be connected to the RA0/AN0 pin if the TRISA<0> bit is set and the CVROE bit (CVRCON<6>) is set. Enabling the voltage reference output onto the RA0/AN0 pin, with an input signal present, will increase current consumption. Connecting RA0/AN0 as a digital output, with CVRSS enabled, will also increase current consumption.

The RA0/AN0 pin can be used as a simple D/A output with limited drive capability. Due to the limited current drive capability, a buffer must be used on the voltage reference output for external connections to VREF. Figure 22-2 shows an example buffering technique.

# PIC18FXX8

BNO	v	Branch if N	Branch if Not Overflow							
Synta	ax:	[ <i>label</i> ] BN	[ <i>label</i> ] BNOV n							
Operands:		-128 ≤ n ≤ 1	-128 ≤ n ≤ 127							
Oper	ation:	if Overflow I (PC) + 2 + 2	if Overflow bit is '0' (PC) + 2 + 2n $\rightarrow$ PC							
Statu	is Affected:	None	None							
Enco	oding:	1110	0101 :	nnnn	nnnn					
Desc	ription:	If the Overfl program wil The 2's con added to the incremented tion, the new PC + 2 + 2r two-cycle in	If the Overflow bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruc- tion, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.							
Word	ls:	1	1							
Cvcles:		1(2)	1(2)							
Q Cycle Activity: If Jump:										
	Q1	Q2	Q3		Q4					
	Decode	Read literal 'n'	Process Data	Wri	te to PC					
	No operation	No operation	No operatior	n op	No eration					
lf No	o Jump:									
	Q1	Q2	Q3		Q4					
	Decode	Read literal 'n'	Process Data	ор	No eration					
Example: HERE BNOV Jump										
	Before Instruct PC	tion = ad	on = address (HERE)							
After Instruction If Overflow = 0; PC = address (Jump) If Overflow = 1; PC = address (HERE + 2)										

BNZ		Branch if N	Branch if Not Zero						
Synta	ax:	[label] BN	Zn						
Oper	ands:	-128 ≤ n ≤ 1	27						
Oper	ation:	if Zero bit is (PC) + 2 + 2	$_{0}^{'0}$ 2n $\rightarrow$ PC	;					
Statu	s Affected:	None							
Enco	ding:	1110	0001	nnni	n nnnn				
Description: If the Zero bit is '0', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will ha incremented to fetch the next instruc- tion, the new address will be PC + 2 + 2n. This instruction is then a two evolutions instruction.					e program er '2n' is PC will have ext instruc- be on is then a				
Word	s:	1	1						
Cvcle	es:	1(2)							
Q C If Ju	ycle Activity: mp:								
	Q1	Q2	Q	3	Q4				
	Decode	Read literal 'n'	Proce Data	ess a	Write to PC				
	No operation	No operation	No operation		No operation				
If No	Jump:								
	Q1	Q2	Q	3	Q4				
	Decode	Read literal 'n'	Process Data		No operation				
		L							
<u>Exan</u>	<u>ıple:</u>	HERE	BNZ	Jump					
	Before Instruc PC	tion = ad	dress (I	HERE)					
After Instruction If Zero = 0; PC = address (Jump) If Zero = 1; PC = address (HERE + 2)									

# PIC18FXX8

BBNCE	Botate Bight f (no carry)	SETE	Set f			
Syntax:	[/abe/] BBNCE f[d[a]]	Syntax:		TF f[a]		
Dperands: $0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$		Operands:	ds: $0 \le f \le 255$ $a \in [0,1]$ on: FFh $\rightarrow f$ Affected: None ng: 0110 100a ffff fff tion: The contents of the specified register are set to FFh. If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value (default). 1 1 e Activity:			
Operation:	peration: $(f) \rightarrow dest, (f<0>) \rightarrow dest<7>$ atus Affected: N, Z ncoding: 0100 00da fffff ffff escription: The contents of register 'f' are rotated one bit to the right. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value (default).					
Status Affected: Encoding: Description:						
	register f	Q1 Decode	Q2 Read	Q3 Process	Q4 Write	
Words:	1		register T	Data	register T	
Q Cycle Activity: Q1 Decode	Q2 Q3 Q4 Read Process Write to register 'f' Data destination	Example: Before Instruc REG After Instructio REG	SETF 1 ction = 0x on = 0x	REG 5 <b>A</b> FF		
Example 1: Before Instruc REG After Instructio BEG	RRNCF REG, 1, 0 ction = 1101 0111 on = 1110 1011					
Example 2: Before Instruc W REG After Instructio	= 1110 1011 RRNCF REG, W ction = ? = 1101 0111 on = 1110 1011					
REG	= 1101 0111					

## 27.1 DC Characteristics (Continued)

PIC18LFXX8 (Industrial)				Standard Operating Conditions (unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for industrial							
PIC18FXX8 (Industrial, Extended)			$\begin{array}{llllllllllllllllllllllllllllllllllll$								
Param No.	Symbol	Characteristic/ Device	Min	Min Typ Max Units Conditions							
	ΔIWDT Module Differential Current										
D022		Watchdog Timer PIC18LFXX8		0.75 0.8 7	1.5 8 25	μΑ μΑ μΑ	VDD = 2.5V, +25°C VDD = 2.0V, -40°C to +85°C VDD = 4.2V, -40°C to +85°C				
D022		Watchdog Timer PIC18FXX8		7 7 7	25 25 45	μΑ μΑ μΑ	VDD = 4.2V, +25°C VDD = 4.2V, -40°C to +85°C VDD = 4.2V, -40°C to +125°C				
D022A	ΔIBOR	Brown-out Reset <sup>(5)</sup> PIC18LFXX8		38 42 49	50 55 65	μΑ μΑ μΑ	VDD = 2.0V, +25°C VDD = 2.0V, -40°C to +85°C VDD = 4.2V, -40°C to +85°C				
D022A		Brown-out Reset <sup>(5)</sup> PIC18FXX8		46 49 50	65 65 75	μΑ μΑ μΑ	VDD = 4.2V, +25°C VDD = 4.2V, -40°C to +85°C VDD = 4.2V, -40°C to +125°C				
D022B	ΔILVD	Low-Voltage Detect <sup>(5)</sup> PIC18LFXX8	   	36 40 47	50 55 65	μΑ μΑ μΑ	VDD = 2.0V, +25°C VDD = 2.0V, -40°C to +85°C VDD = 4.2V, -40°C to +85°C				
D022B		Low-Voltage Detect <sup>(5)</sup> PIC18FXX8		44 47 47	65 65 75	μΑ μΑ μΑ	VDD = 4.2V, +25°C VDD = 4.2V, -40°C to +85°C VDD = 4.2V, -40°C to +125°C				
D025	ΔITMR1	Timer1 Oscillator PIC18LFXX8	-  -  -	6.2 6.2 7.5	40 45 55	μΑ μΑ μΑ	VDD = 2.0V, +25°C VDD = 2.0V, -40°C to +85°C VDD = 4.2V, -40°C to +85°C				
D025		Timer1 Oscillator PIC18FXX8		7.5 7.5 7.5	55 55 65	μΑ μΑ μΑ	VDD = 4.2V, +25°C VDD = 4.2V, -40°C to +85°C VDD = 4.2V, -40°C to +125°C				

Legend: Rows are shaded for improved readability.

Note 1: This is the limit to which VDD can be lowered in Sleep mode, or during a device Reset, without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

The test conditions for all IDD measurements in active operation mode are:  $\frac{OSC1}{MCLR} = \text{external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD}$   $\frac{MCLR}{MCLR} = \text{VDD; WDT enabled/disabled as specified.}$ 

**3:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD and VSS and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, ...).

- 4: For RC oscillator configuration, current through REXT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2 REXT (mA) with REXT in kOhm.
- 5: The LVD and BOR modules share a large portion of circuitry. The △IBOR and △ILVD currents are not additive. Once one of these modules is enabled, the other may also be enabled without further penalty.

#### 27.3.3 TIMING DIAGRAMS AND SPECIFICATIONS



#### TABLE 27-6: EXTERNAL CLOCK TIMING REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Мах	Units	Conditions
1A	Fosc	External CLKI Frequency <sup>(1)</sup>	DC	40	MHz	EC, ECIO oscillator, -40°C to +85°C
		Oscillator Frequency <sup>(1)</sup>	DC	25	MHz	EC, ECIO oscillator, +85°C to +125°C
			DC	4	MHz	RC oscillator
			0.1	4	MHz	XT oscillator
			4	25	MHz	HS oscillator, -40°C to +85°C
			4	25	MHz	HS oscillator, +85°C to +125°C
			4	10	MHz	HS + PLL oscillator, -40°C to +85°C
			4	6.25	MHz	HS + PLL oscillator, +85°C to +125°C
			DC	200	kHz	LP oscillator
1	Tosc	External CLKI Period <sup>(1)</sup>	25	—	ns	EC, ECIO oscillator, -40°C to +85°C
		Oscillator Period("	40	—	ns	EC, ECIO oscillator, +85°C to +125°C
			250	—	ns	RC oscillator
			250	10,000	ns	XT oscillator
			40	—	ns	HS oscillator, -40°C to +85°C
			40	—	ns	HS oscillator, +85°C to +125°C
			100	250	ns	HS + PLL oscillator, -40°C to +85°C
			160	250	ns	HS + PLL oscillator, +85°C to +125°C
			5	200	μs	LP oscillator
2	Тсү	Instruction Cycle Time <sup>(1)</sup>	100	—	ns	Tcy = $4/Fosc$ , $-40^{\circ}C$ to $+85^{\circ}C$
			160	—	ns	TCY = 4/FOSC, +85°C to +125°C
3	TosL,	External Clock in (OSC1)	30	—	ns	XT oscillator
	IOSE		2.5	—	ns	LP oscillator
			10	—	μs	HS oscillator
4	TosR,	External Clock in (OSC1)	—	20	ns	XT oscillator
	IOSF	Rise of Fall Time	—	50	ns	LP oscillator
			—	7.5	ns	HS oscillator

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "Min." values with an external clock applied to the OSC1/CLKI pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

© 2006 Microchip Technology Inc.





Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
130	Tad	A/d Clock Period PIC18FXX8		1.6	20 <sup>(5)</sup>	μs	Tosc based, VREF $\geq$ 3.0V
			PIC18LFXX8	3.0	20 <sup>(5)</sup>	μs	Tosc based, VREF full range
			PIC18FXX8	2.0	6.0	μs	A/D RC mode
			PIC18LFXX8	3.0	9.0	μs	A/D RC mode
131	TCNV	Conversion Time (not including acquisitic	11	12	Tad		
132	TACQ	Acquisition Time (Note 3)		15 10		μs μs	-40°C ≤ Temp ≤ +125°C 0°C ≤ Temp ≤ +125°C
135	Tswc	Switching Time from Co	onvert $\rightarrow$ Sample		(Note 4)		
136	Тамр	Amplifier Settling Time (Note 2)		1	_	μs	This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 5 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).

## TABLE 27-24: A/D CONVERSION REQUIREMENTS

**Note 1:** ADRES register may be read on the following TCY cycle.

- 2: See Section 20.0 "Compatible 10-Bit Analog-to-Digital Converter (A/D) Module" for minimum conditions when input voltage has changed more than 1 LSb.
- **3:** The time for the holding capacitor to acquire the "New" input voltage when the voltage changes full scale after the conversion (AVDD to AVSS or AVSS to AVDD). The source impedance (*Rs*) on the input channels is  $50\Omega$ .
- 4: On the next Q4 cycle of the device clock.
- 5: The time of the A/D clock period is dependent on the device frequency and the TAD clock divider.



# FIGURE 28-27: MINIMUM AND MAXIMUM VIN vs. VDD (I<sup>2</sup>C™ INPUT, -40°C TO +125°C)



