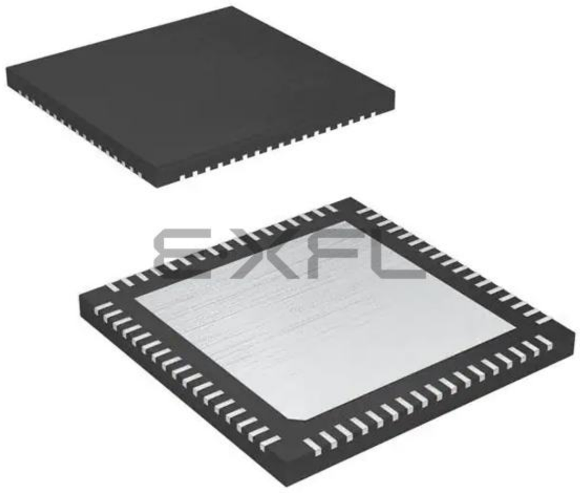


Welcome to E-XFL.COM

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	Obsolete
Core Processor	80515
Core Size	8-Bit
Speed	24MHz
Connectivity	I²C, SmartCard, UART/USART, USB
Peripherals	LED, POR, WDT
Number of I/O	8
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 6.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	68-VFQFN Exposed Pad
Supplier Device Package	68-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/73s1217f-68m-f-pe

Program Memory

Address	Use
0xFFFF	Flash Program Memory 64K Bytes
0x0000	

External Data Memory

Address	Use
0xFFFF	Peripheral Control Registers (128b)
0XFF80	
0xFF7F	Smart Card Control (384b)
0XFE00	
0xFDFF	USB Registers (512b)
0XFC00	
0xFBFF	–
0x0800	
0x07FF	XRAM
0x0000	

Internal Data Memory

Address	Use: Indirect Access	Use: Direct Access
0xFF	Byte RAM	SFRs
0x80		
0x7F	Byte RAM	
0x48		
0x47	Bit/Byte RAM	
0x20		
0x1F	Register bank 3	
0x18		
0x17	Register bank 2	
0x10		
0x0F	Register bank 1	
0x08		
0x07	Register bank 0	
0x00		

Figure 2: Memory Map

Dual Data Pointer: The Dual Data Pointer accelerates the block moves of data. The standard DPTR is a 16-bit register that is used to address external memory. In the 80515 core, the standard data pointer is called DPTR, the second data pointer is called DPTR1. The data pointer select bit chooses the active pointer. The data pointer select bit is located at the LSB of the DPS IRAM special function register (DPS.0). DPTR is selected when DPS.0 = 0 and DPTR1 is selected when DPS.0 = 1.

The user switches between pointers by toggling the LSB of the DPS register. All DPTR-related instructions use the currently selected DPTR for any activity.

Note: The second data pointer may not be supported by certain compilers.

Name	Location	Reset Value	Description
KROW	0XD2	0x3F	Keypad Row
KSCAN	0XD3	0x00	Keypad Scan Time
KSTAT	0XD4	0x00	Keypad Control/Status
KSIZE	0XD5	0x00	Keypad Size
KORDERL	0XD6	0x00	Keypad Column LS Scan Order
KORDERH	0XD7	0x00	Keypad Column MS Scan Order
BRCON	0xD8	0x00	Baud Rate Control Register (only BRCON.7 bit used)
A	0xE0	0x00	Accumulator
B	0xF0	0x00	B Register

1.5.3 External Data Special Function Registers (SFRs)

A map of the XRAM Special Function Registers is shown in Table 8. The smart card registers are listed separately in Table 114.

Table 8: XRAM Special Function Registers Reset Values

Name	Location	Reset Value	Description
DAR	0x FF80	0x00	Device Address Register (I^2C)
WDR	0x FF81	0x00	Write Data Register (I^2C)
SWDR	0x FF82	0x00	Secondary Write Data Register (I^2C)
RDR	0x FF83	0x00	Read Data Register (I^2C)
SRDR	0x FF84	0x00	Secondary Read Data Register (I^2C)
CSR	0x FF85	0x00	Control and Status Register (I^2C)
USRIntCtl1	0x FF90	0x00	External Interrupt Control 1
USRIntCtl2	0x FF91	0x00	External Interrupt Control 2
USRIntCtl3	0x FF92	0x00	External Interrupt Control 3
USRIntCtl4	0x FF93	0x00	External Interrupt Control 4
INT5Ctl	0x FF94	0x00	External Interrupt Control 5
INT6Ctl	0x FF95	0x00	External Interrupt Control 6
MPUCKCtl	0x FFA1	0x0C	MPU Clock Control
RTCCtl	0x FFB0	0x00	Real Time Clock Control
RTCCnt3	0x FFB1	0x00	RTC Count 3
RTCCnt2	0x FFB2	0x00	RTC Count 2
RTCCnt1	0x FFB3	0x00	RTC Count 1
RTCCnt0	0x FFB4	0x00	RTC Count 0
RTCACC2	0x FFB5	0x00	RTC Accumulator 2
RTCACC1	0x FFB6	0x00	RTC Accumulator 1
RTCACC0	0x FFB7	0x00	RTC Accumulator 0
RTCTrim2	0x FFB8	0x00	RTC TRIM 2
RTCTrim1	0x FFB9	0x00	RTC TRIM 1
RTCTrim0	0x FFBA	0x00	RTC TRIM 0
ACOMP	0x FFD0	0x00	Analog Compare Register
TRIMPCtl	0x FFD1	0x00	TRIM Pulse Control

Name	Location	Reset Value	Description
FUSECtl	0x FFD2	0x00	FUSE Control
VDDFCtl	0x FFD4	0x00	VDDFault Control
SECReg	0x FFD7	0x00	Security Register
MISCtl0	0x FFF1	0x00	Miscellaneous Control Register 0
MISCtl1	0x FFF2	0x10	Miscellaneous Control Register 1
LEDCtl	0x FFF3	0xFF	LED Control Register

Accumulator (ACC, A): ACC is the accumulator register. Most instructions use the accumulator to hold the operand. The mnemonics for accumulator-specific instructions refer to accumulator as “A”, not ACC.

B Register: The B register is used during multiply and divide instructions. It can also be used as a scratch-pad register to hold temporary data.

Interrupt Enable 1 Register (IEN1): 0xB8 ← 0x00

Table 20: The IEN1 Register

MSB LSB

–	SWDT	EX6	EX5	EX4	EX3	EX2	–

Bit	Symbol	Function
IEN1.7	–	
IEN1.6	SWDT	Not used for interrupt control.
IEN1.5	EX6	EX6 = 0 – disable external interrupt 6.
IEN1.4	EX5	EX5 = 0 – disable external interrupt 5.
IEN1.3	EX4	EX4 = 0 – disable external interrupt 4.
IEN1.2	EX3	EX3 = 0 – disable external interrupt 3.
IEN1.1	EX2	EX2 = 0 – disable external interrupt 2.
IEN1.0	–	

Interrupt Enable 2 Register (IEN2): 0x9A ← 0x00

Table 21: The IEN2 Register

MSB LSB

–	–	–	–	–	–	–	ES1

Bit	Symbol	Function
IEN2.0	ES1	ES1 = 0 – disable serial channel interrupt.

Interrupt Request Register (IRCON): 0xC0 ← 0x00

Table 24: The IRCON Register

MSB							LSB
–	–	EX6	IEX5	IEX4	IEX3	IEX2	–

Bit	Symbol	Function
IRCON.7	–	
IRCON.6	–	
IRCON.5	IEX6	External interrupt 6 flag.
IRCON.4	IEX5	External interrupt 5 flag.
IRCON.3	IEX4	External interrupt 4 flag.
IRCON.2	IEX3	External interrupt 3 flag.
IRCON.1	IEX2	External interrupt 2 flag.
IRCON.0	–	

1.7.5.3 External Interrupts

The external interrupts (external to the CPU core) are connected as shown in Table 25. Interrupts with multiple sources are OR'ed together and individual interrupt source control is provided in XRAM SFRs to mask the individual interrupt sources and provide the corresponding interrupt flags. Multifunction USR [7:0] pins control Interrupts 0 and 1. Dedicated external interrupt pins INT2 and INT3 control interrupts 2 and 3. The polarity of interrupts 2 and 3 is programmable in the MPU. Interrupts 4, 5 and 6 have multiple peripheral sources and are multiplexed to one of these three interrupts. The peripheral functions will be described in subsequent sections. Generic 80515 MPU literature states that interrupts 4 through 6 are defined as rising edge sensitive. Thus, the hardware signals attached to interrupts 4, 5 and 6 are converted to rising edge level by the hardware.

SFR (special function register) enable bits must be set to permit any of these interrupts to occur. Likewise, each interrupt has its own flag bit that is set by the interrupt hardware and is reset automatically by the MPU interrupt handler.

Table 25: External MPU Interrupts

External Interrupt	Connection	Polarity	Flag Reset
0	USR I/O High Priority	see USRIntCtlx	Automatic
1	USR I/O Low Priority	see USRIntCtlx	Automatic
2	External Interrupt Pin INT2	Edge selectable	Automatic
3	External Interrupt Pin INT3	Edge selectable	Automatic
4	Smart Card Interrupts	N/A	Automatic
5	USB, RTC and Keypad	N/A	Automatic
6	I^2C, V_{DD}_Fault, Analog Comp	N/A	Automatic

Note: Interrupts 4, 5 and 6 have multiple interrupt sources and the flag bits are cleared upon reading of the corresponding register. To prevent any interrupts from being ignored, the register containing multiple interrupt flags should be stored temporary to allow each interrupt flag to be tested separately to see which interrupt(s) is/are pending.

Interrupt Priority 1 Register (IP1): 0xB9 ← 0x00

Table 29: The IP1 Register

MSB							LSB
–	–	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0

Table 30: Priority Levels

IP1.x	IP0.x	Priority Level
0	0	Level0 (lowest)
0	1	Level1
1	0	Level2
1	1	Level3 (highest)

Table 31: Interrupt Polling Sequence

	Polling sequence ↓
External interrupt 0	
Serial channel 1 interrupt	
Timer 0 interrupt	
External interrupt 2	
External interrupt 1	
External interrupt 3	
Timer 1 interrupt	
Serial channel 0 interrupt	
External interrupt 4	
External interrupt 5	
External interrupt 6	

1.7.5.6 Interrupt Sources and Vectors

Table 32 shows the interrupts with their associated flags and vector addresses.

Table 32: Interrupt Vectors

Interrupt Request Flag	Description	Interrupt Vector Address
N/A	Chip Reset	0x0000
IE0	External interrupt 0	0x0003
TF0	Timer 0 interrupt	0x000B
IE1	External interrupt 1	0x0013
TF1	Timer 1 interrupt	0x001B
RI0/TI0	Serial channel 0 interrupt	0x0023
RI1/TI1	Serial channel 1 interrupt	0x0083
IEX2	External interrupt 2	0x004B
IEX3	External interrupt 3	0x0053
IEX4	External interrupt 4	0x005B
IEX5	External interrupt 5	0x0063
IEX6	External interrupt 6	0x006B

Miscellaneous Control Register 0 (MISCtl0): 0xFFF1 ← 0x00

Transmit and receive (TX and RX) pin selection and loop back test configuration are set up via this register.

Table 37: The MISCtl0 Register

MSB							LSB
PWRDN	–	–	–	–	–	SLPBK	SSEL

Bit	Symbol	Function
MISCtl0.7	PWRDN	This bit places the 73S1217F into a power down state.
MISCtl0.6	–	
MISCtl0.5	–	
MISCtl0.4	–	
MISCtl0.3	–	
MISCtl0.2	–	
MISCtl0.1	SLPBK	1 = UART loop back testing mode. The pins TXD and RXD are to be connected together externally (with SLPBK =1) and therefore: SLPBK SSEL Mode 0 0 normal using Serial_0 0 1 normal using Serial_1 1 0 Serial_0 TX feeds Serial_1 RX 1 1 Serial_1 TX feeds Serial_0 RX
MISCtl0.0	SSEL	Selects either Serial_1 if set =1 or Serial_0 if set = 0 to be connected to RXD and TXD pins.

1.7.6.1 Serial Interface 0

The Serial Interface 0 can operate in four modes:

- **Mode 0**
  Pin RX serves as input and output. TX outputs the shift clock. Eight bits are transmitted with the LSB first. The baud rate is fixed at 1/12 of the crystal frequency. Reception is initialized in Mode 0 by setting the flags in S0CON as follows: RI0 = 0 and REN0 = 1. In other modes, a start bit when REN0 = 1 starts receiving serial data.
- **Mode 1**
  Pin RX serves as input, and TX serves as serial output. No external shift clock is used, 10 bits are transmitted: a start bit (always 0), 8 data bits (LSB first), and a stop bit (always 1). On receive, a start bit synchronizes the transmission, 8 data bits are available by reading S0BUF, and stop bit sets the flag RB80 in the Special Function Register S0CON. In mode 1 either internal baud rate generator or timer 1 can be use to specify baud rate.
- **Mode 2**
  This mode is similar to Mode 1, with two differences. The baud rate is fixed at 1/32 or 1/64 of oscillator frequency and 11 bits are transmitted or received: a start bit (0), 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). The 9th bit can be used to control the parity of the serial interface: at transmission, bit TB80 in S0CON is output as the 9th bit, and at receive, the 9th bit affects RB80 in Special Function Register S0CON.
- **Mode 3**
  The only difference between Mode 2 and Mode 3 is that in Mode 3 either internal baud rate generator or timer 1 can be use to specify baud rate.

The S0BUF register is used to read/write data to/from the serial 0 interface.

Serial Interface 0 Control Register (S0CON): 0x9B ← 0x00

Serial Interface Control Register (S1CON): 0x9B ← 0x00

The function of the serial port depends on the setting of the Serial Port Control Register S1CON.

Table 39: The S1CON Register

MSB							LSB
SM	–	SM21	REN1	TB81	RB81	TI1	RI1

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th></tr>
<tr><td>S1CON.7</td><td>SM</td><td>Sets the UART operation mode.
<table><tr><th>SM</th><th>Mode</th><th>Description</th><th>Baud Rate</th></tr><tr><td>0</td><td>A</td><td>9-bit UART</td><td>variable</td></tr><tr><td>1</td><td>B</td><td>8-bit UART</td><td>variable</td></tr></table></td></tr>
<tr><td>S1CON.6</td><td>–</td><td></td></tr>
<tr><td>S1CON.5</td><td>SM21</td><td>Enables the inter-processor communication feature.</td></tr>
<tr><td>S1CON.4</td><td>REN1</td><td>If set, enables serial reception. Cleared by software to disable reception.</td></tr>
<tr><td>S1CON.3</td><td>TB81</td><td>The 9th transmitted data bit in Mode A. Set or cleared by the MPU, depending on the function it performs (parity check, multiprocessor communication etc.).</td></tr>
<tr><td>S1CON.2</td><td>RB81</td><td>In Mode B, if sm21 is 0, rb81 is the stop bit. Must be cleared by software.</td></tr>
<tr><td>S1CON.1</td><td>TI1</td><td>Transmit interrupt flag, set by hardware after completion of a serial transfer. Must be cleared by software.</td></tr>
<tr><td>S1CON.0</td><td>RI1</td><td>Receive interrupt flag, set by hardware after completion of a serial reception. Must be cleared by software.</td></tr>
</table>

Multiprocessor operation mode: The feature of receiving 9 bits in Modes 2 and 3 of Serial Interface 0 or in Mode A of Serial Interface 1 can be used for multiprocessor communication. In this case, the slave processors have bit SM20 in S0CON or SM21 in S1CON set to 1. When the master processor outputs slave's address, it sets the 9th bit to 1, causing a serial port receive interrupt in all the slaves. The slave processors compare the received byte with their network address. If there is a match, the addressed slave will clear SM20 or SM21 and receive the rest of the message, while other slaves will leave the SM20 or SM21 bit unaffected and ignore this message. After addressing the slave, the host will output the rest of the message with the 9th bit set to 0, so no serial port receive interrupt will be generated in unselected slaves.

External Interrupt Control Register (USRIntCtl1) : 0xFF90 ← 0x00

Table 50: The USRIntCtl1 Register

MSB LSB

–	U1IS.6	U1IS.5	U1IS.4	–	U0IS.2	U0IS.1	U0IS.0

External Interrupt Control Register (USRIntCtl2) : 0xFF91 ← 0x00

Table 51: The USRIntCtl2 Register

MSB LSB

–	U3IS.6	U3IS.5	U3IS.4	–	U2IS.2	U2IS.1	U2IS.0

External Interrupt Control Register (USRIntCtl3) : 0xFF92 ← 0x00

Table 52: The USRIntCtl3 Register

MSB LSB

–	U5IS.6	U5IS.5	U5IS.4	–	U4IS.2	U4IS.1	U4IS.0

External Interrupt Control Register (USRIntCtl4) : 0xFF93 ← 0x00

Table 53: The USRIntCtl4 Register

MSB LSB

–	U7IS.6	U7IS.5	U7IS.4	–	U6IS.2	U6IS.1	U6IS.0

Figure 11 shows the timing of the I²C write mode.

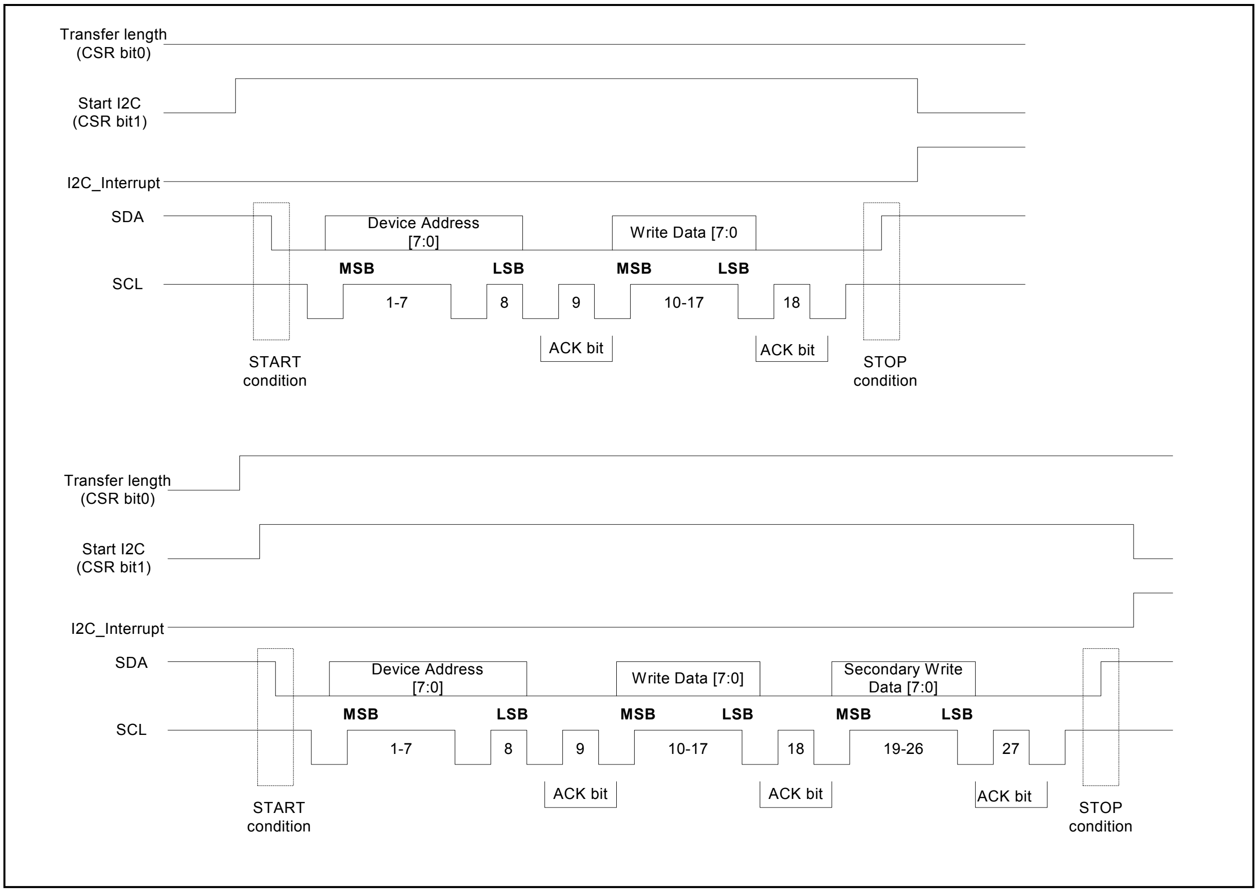


Figure 11: I²C Write Mode Operation

1.7.13.2 I²C Read Sequence

To read data on the I²C Master Bus from a slave device, the 80515 has to program the following registers in this sequence:

1. Write slave device address to the Device Address register (DAR). The data contains 7 bits device address and 1 bit of op-code. The op-code bit should be written with a 1.
2. Write control data to the Control and Status register (CSR). Write a 1 to bit 1 to start I²C Master Bus. Also write a 1 to bit 0 if the Secondary Read Data register (SRDR) is to be captured from the I²C Slave device.
3. Wait for I²C interrupt to be asserted. It indicates that the read operation on the I²C bus is done. Refer to information about the INT6Ctl, IEN1 and IRCON registers for masking and flag operation.
4. Read data from the Read Data register (RDR).
5. Read data from Secondary Read Data register (SRDR) if bit 0 of Control and Status register (CSR) is written with a 1.

Keypad Column Register (KCOL): 0xD1 ← 0x1F

This register contains the value of the column of a key detected as valid by the hardware. In bypass mode, this register firmware writes directly this register to carry out manual scanning.

Table 69: The KCOL Register

MSB LSB

–	–	–	COL.4	COL.3	COL.2	COL.1	COL.0

Bit	Symbol	Function
KCOL.7	–	
KCOL.6	–	
KCOL.5	–	
KCOL.4	COL.4	Drive lines bit mapped to corresponding with pins COL(4:0). When a key is detected, firmware reads this register to determine column. In bypass (S/W keyscan) mode, Firmware writes this register directly. 0x1E = COL(0) low, all others high. 0x0F = COL(4) low, all others high. 0x1F = COL(4:0) all high.
KCOL.3	COL.3	
KCOL.2	COL.2	
KCOL.1	COL.1	
KCOL.0	COL.0	

Keypad Row Register (KROW): 0xD2 ← 0x3F

This register contains the value of the row of a key detected as valid by the hardware. In bypass mode, this register firmware reads directly this register to carry out manual detection.

Table 70: The KROW Register

MSB LSB

–	–	ROW.5	ROW.4	ROW.3	ROW.2	ROW.1	ROW.0

Bit	Symbol	Function
KROW.7	–	
KROW.6	–	
KROW.5	ROW.6	Sense lines bit mapped to correspond with pins ROW(5:0). When key detected, firmware reads this register to determine row. In bypass mode, firmware reads rows and has to determine if there was a key press or not. 0x3E = ROW(0) low, all others high. 0x1F = ROW(5) low, all others high. 0x3F = ROW(5:0) all high.
KROW.4	ROW.4	
KROW.3	ROW.3	
KROW.2	ROW.2	
KROW.1	ROW.1	
KROW.0	ROW.0	

Keypad Scan Time Register (KSIZE): 0xD5 ← 0x00

This register is not applicable when HWSCEN is not set. Unused row inputs should be connected to VDD.

Table 73: The KSIZE Register

MSB							LSB
–	–	ROWSIZ.2	ROWSIZ.1	ROWSIZ.0	COLSIZ.2	COLSIZ.1	COLSIZ.0

Bit	Symbol	Function
KSIZE.7	–	
KSIZE.6	–	
KSIZE.5	ROWSIZ.2	Defines the number of rows in the keypad. Maximum number is 6 given the number of row pins on the package. Allows for a reduced keypad size for scanning.
KSIZE.4	ROWSIZ.1	
KSIZE.3	ROWSIZ.0	
KSIZE.2	COLSIZ.2	Defines the number of columns in the keypad. Maximum number is 5 given the number of column pins on the package. Allows for a reduced keypad size for scanning.
KSIZE.1	COLSIZ.1	
KSIZE.0	COLSIZ.0	

Keypad Column LS Scan Order Register (KORDERL): 0xD6 ← 0x00

In the KORDERL and KORDERH registers, Column Scan Order(14:0) is grouped into 5 sets of 3 bits each. Each set determines which column (COL(4:0) pin) to activate by loading the column number into the 3 bits. When in HW_Scan_Enable mode, the hardware will step through the sets from 1Col to 5Col (up to the number of columns in Colsize) and scan the column defined in the 3 bits. To scan in sequential order, set a counting pattern with 0 in set 0, and 1 in set 1,and 2 in set 2, and 3 in set 3, and 4 in set 4. The firmware should update this as part of the interrupt service routine so that the new scan order is loaded prior to the next key being pressed. For example, to scan COL(0) first, 1Col(2:0) should be loaded with 000'b. To scan COL(4) fifth, 5Col(2:0) should be loaded with 100'b.

Table 74: The KORDERL Register

MSB							LSB
3COL.1	3COL.0	2COL.2	2COL.1	2COL.0	1COL.2	1COL.1	1COL.0

Bit	Symbol	Function
KORDERL.7	3COL.1	Column to scan 3rd (lsb's).
KORDERL.6	3COL.0	
KORDERL.5	2COL.2	Column to scan 2nd.
KORDERL.4	2COL.1	
KORDERL.3	2COL.0	
KORDERL.2	1COL.2	Column to scan 1st.
KORDERL.1	1COL.1	
KORDERL.0	1COL.0	

1.7.16 USB Interface

The 73S1217F provides a single interface, full speed -12Mbps - USB device port as per the *Universal Serial Bus Specification, Revision 2.0* (backward compatible with USB 1.1). USB circuitry gathers the transceiver, the Serial Interface Engine (SIE), and the data buffers. An internal pull-up to V_{DD} on D+ indicates that the device is a full speed device attached to the USB bus (allows full speed recognition by the host without adding any external components). When using the USB interface, V_{DD} must be between 3.0V – 3.6V in order to meet the USB VOH requirement. The interface is highly configurable under firmware control. Control (Endpoint 0), Interrupt IN, Bulk IN and Bulk OUT transfers are supported. Four endpoints are supported and are configured by firmware:

- Endpoint 0, the default (Control) endpoint as required by the USB specification, is used to exchange control and status information between the 73S1217F and the USB host.
- Bulk IN Endpoint #1
- Bulk OUT Endpoint #1
- Interrupt IN Endpoint #2
- The USB block contains several FIFOs used for communication.
- There is a 128-byte RAM FIFO for each BULK endpoint. Maximum Bulk packet size is 64 bytes.
- There is a 32-byte RAM FIFO for the interrupt endpoint. Maximum Interrupt packet size is 16 bytes.
- There is a 16-byte RAM FIFO for the control endpoint. Maximum Control packet size is 16 bytes.

Figure 15 shows the simplified block diagram of the USB interface.

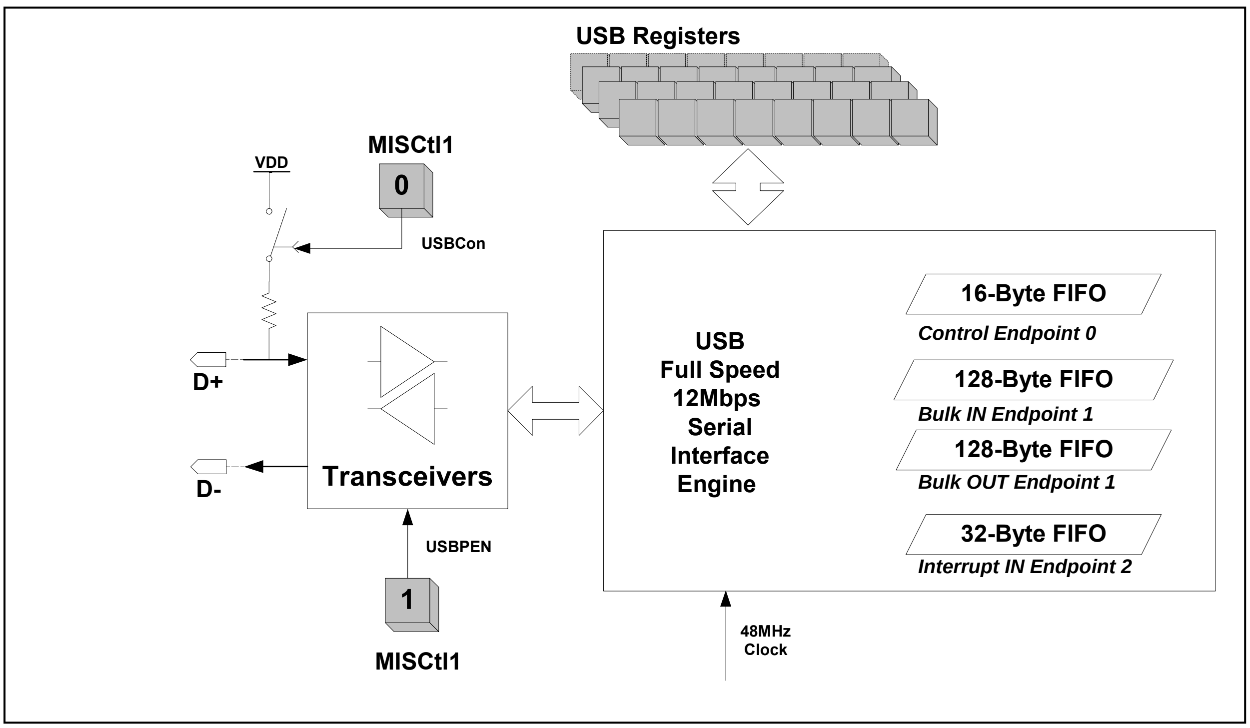
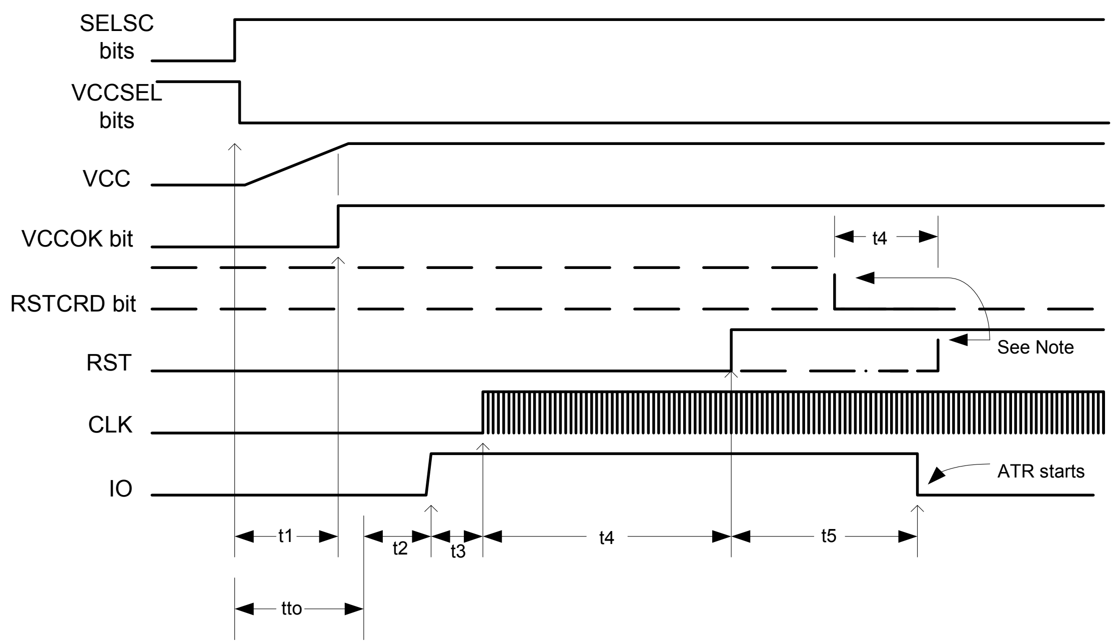


Figure 15: USB Block Diagram

The USB interface consists of a Serial Interface Engine (SIE) that handles NRZI encoding/decoding, bit stuffing / unstuffing, and CRC generation/checking. It also generates headers for packets to be transmitted and decodes the headers of received packets. An analog transceiver interfaces with the external USB bus. The USB interface hardware performs error checking and removes the USB protocol fields from the incoming messages before passing the data to the firmware. The hardware also adds the USB protocol fields to the outgoing messages coming from the firmware. The hardware implements NRZI encoding/decoding, CRC checking/generation (both on data and token packets), device address

1.7.17.2 Answer to Reset Processing

A card insertion event generates an interrupt to the firmware, which is then responsible for the configuration of the electrical interface, the UART and activation of the card. The activation sequencer goes through the power up sequence as defined in the ISO 7816-3 specification. An asynchronous activation timing diagram is shown in Figure 18. After the card RST is de-asserted, the firmware instructs the hardware to look for a TS byte that begins the ATR response. If a response is not provided within the pre-programmed timeout period, an interrupt is generated and the firmware can then take appropriate action, including instructing the 73S1217F to begin a deactivation sequence. Once commanded, the deactivation sequencer goes through the power down sequence as defined in the ISO 7816-3 specification. If an ATR response is received, the hardware looks for a TS byte that determines direct/inverse convention. The hardware handles the indirect convention conversion such that the embedded firmware only receives direct convention. This feature can be disabled by firmware within the SByteCtl register. Parity checking and break generation is performed on the TS byte unless disabled by firmware. If during the card session, a card removal, over-current or other error event is detected, the hardware will automatically perform the deactivation sequence and then generate an interrupt to the firmware. The firmware can then perform any other error handling required for proper system operation. Smart card RST, I/O and CLK, C4, C8 shall be low before the end of the deactivation sequence. Figure 19 shows the timing for a deactivation sequence.



t1: SELSC.1 bit set (selects internal ICC interface) and a non-zero value in VCCSEL bits (calling for a value of Vcc of 1.8, 3.0, or 5.0 volts) will begin the activation sequence. t1 is the time for Vcc to rise to acceptable level, declared as Vcc OK (bit VCCOK gets set). This time depends on filter capacitor value and card Icc load.

tto: The time allowed for Vcc to rise to Vcc OK status after setting of the VCCSEL bits. This time is generated by the VCCTMR counter. If Vcc OK is not set, (bit VCCOK) at this time, a deactivation will be initiated. VCCSEL bits are not automatically cleared. The firmware must clear the VCCSEL bits before starting a new activation.

t2: Time from VCCTmr timeout and VCC OK to IO reception (high), typically 2-3 CLK cycles if RDYST = 0. If RDYST = 1, t2 starts when VCCOK = 1.

t3: Time from IO = high to CLK start, typically 2-3 CLK cycles.

t4: Time allowed for start of CLK to de-assertion of RST. Programmable by the RLength register.

t5: Time allowed for ATR timeout, set by the STSTO register.

Note: If the RSTCRD bit is set, RST is asserted (low). Upon clearing RSTCRD bit, RST will be de-asserted after t4.

Figure 18: Asynchronous Activation Sequence Timing

When the SCISYN or SCESNC bits (SPrtcol, bit 7, bit 5, respectively) are set, the selected smart card interface operates in synchronous mode and there are changes in the definition and behavior of pertinent register bits and associated circuitry. The following requirements are to be noted:

1. The source for the smart card clock (CLK or SCLK) is the ETU counter. Only the actively selected interface can have a running synchronous clock. In contrast, an unselected interface may have a running clock in the asynchronous mode of operation.

2. The control bits CLKLVL, SCLKLVL, CLKOFF, and SCLKOFF are functional in synchronous mode. When the CLKOFF bit is set, it will not truncate either the logic low or logic high period when the (stop at) level is of opposite polarity. The CLK/SCLK signal will complete a correct logic low or logic high duty cycle before stopping at the selected level. The CLK "start" is a result of the falling edge of the CLKOFF bit. Setting clock to run when it is stopped low will result in a half period of low before going high. Setting clock to run when it is stopped high will result in the clock going low immediately and then running at the selected rate with 50% duty cycle (within the limitations of the ETU divisor value).

3. The Rlen(7:0) is configured to count the falling edges of the ETU clock (CLK or SCLK) after it has been loaded with a value from 1 to 255. A value of 0 disables the counting function and RLen functions such as I/O source selection (I/O signal bypasses the FIFOs and is controlled by the SCCLK/SCECLK SFRs). When the RLen counter reaches the "max" (loaded) value, it sets the WAITTO interrupt (SCInt, bit 7), which is maskable via WTOIEN (SCIE, bit 7). I2CMODEIt must be reloaded in order to start the counting/clocking process again. This allows the processor to select the number of CLK cycles and hence, the number of bits to be read or written to/from the card.

4. The FIFO is not clocked by the first CLK (falling) edge resulting from a CLKOFF de-assertion (a clock start event) when the CLK was stopped in the high state and RLen has been loaded but not yet clocked.

5. The state of the pin IO or SIO is sampled on the rising edge of CLK/SCLK and stored in bit 5 of the SCCtl/SCECtl register.

6. When Rlen = max or 0 and I2CMODE = 1 (STXCtl, b7), the IO or SIO signal is directly controlled by the data and direction bits in the respective SCCtl and SCECtl register. The state of the data in the TX FIFO is bypassed.

7. In the SPrtcol register, bit 6 (MODE9/8B) becomes active. When set, the RXData FIFO will read nine-bit words with the state of the ninth bit being readable in SRXCtl, bit 7 (B9DAT). The RXDAV interrupt will occur when the ninth bit has been clocked in (rising edge of CLK or SCLK).

8. Care must be taken to clear the RX and TX FIFOs at the start of any transaction. The user shall read the RX FIFO until it indicates empty status. Reading the TX FIFO twice will reset the input byte pointer and the next write to the TX FIFO will load the byte to the "first out" position. Note that the bit pointer (serializer/deserializer) is reset to bit 0 on any change of the TX/RXD bit.

Special bits that are only active for sync mode include: SRXCtl, b7 "BIT9DAT", SPrtcol b6 "MODE9/8B", STXCtl, b7 "I2CMODE", and the definition of SCInt b7, which was "WAITTO", becomes RLenINT interrupt, and SCIE b7, which was "WTOIEN", becomes RLenIEN.

Smart Card Control Register (SCCtl): 0xFE0A ← 0x21

This register is used to monitor reception of data from the smart card.

Table 89: The SCCtl Register

MSB							LSB
RSTCRD	–	IO	IOD	C8	C4	CLKLVL	CLKOFF

Bit	Symbol	Function
SCCtl.7	RSTCRD	1 = Asserts the RST (set RST = 0) to the smart card interface, 0 = De-assert the RST (set RST = 1) to the smart card interface. Can be used to extend RST to the smart card. Refer to the RLength register description. This bit is operational in all modes and can be used to extend RST during activation or perform a "Warm Reset" as required. In auto-sequence mode, this bit should be set = 0 to allow the sequencer to de-assert RST per the RLength parameters. In sync mode (see the SPrtcol register) the sense of this bit is non-inverted, if set =1 , RST = 1, if set = 0, RST = 0. Rlen has no effect on Reset in sync mode.
SCCtl.6	–	
SCCtl.5	IO	Smart Card I/O. Read is state of I/O signal (Caution, this signal is not synchronized to the MPU clock). In Bypass mode, write value is state of signal on I/O. In sync mode, this bit will contain the value of I/O pin on the latest rising edge of CLK.
SCCtl.4	IOD	Smart Card I/O Direction control Bypass mode or sync mode. 1 = input (default), 0 = output.
SCCtl.3	C8	Smart Card C8. When C8 is an output, the value written to this bit will appear on the C8 line. The value read when C8 is an output is the value stored in the register. When C8 is an input, the value read is the value on the C8 pin (Caution, this signal is not synchronized to the MPU clock). When C8 is an input, the value written will be stored in the register but not presented to the C8 pin.
SCCtl.2	C4	Smart Card C4. When C4 is an output, the value written to this bit will appear on the C4 line. The value read when C4 is an output is the value stored in the register. When C4 is an input, the value read is the value on the C4 pin (Caution, this signal is not synchronized to the MPU clock). When C4 is an input, the value written will be stored in the register but not presented to the C4 pin.
SCCtl.1	CLKLVL	1 = High, 0 = Low. If CLKOFF is set = 1, the CLK to smart card will be at the logic level indicated by this bit. If in bypass mode, this bit directly controls the state of CLK.
SCCtl.0	CLKOFF	0 = CLK is enabled. 1 = CLK is not enabled. When asserted, the CLK will stop at the level selected by CLKLVL. This bit has no effect if in bypass mode.

Parity Control Register (SParCtl): 0xFE11 ← 0x00

This register provides the ability to configure the parity circuitry on the smart card interface. The settings apply to both integrated smart card interfaces.

Table 95: The SParCtl Register

MSB							LSB
–	DISPAR	BRKGEN	BRKDET	RETRAN	DISCRX	INSPE	FORCPE

Bit	Symbol	Function
SParCtl.7	–	
SParCtl.6	DISPAR	Disable Parity Check – 1 = disabled, 0 = enabled. If enabled, the UART will check for even parity (the number of 1's including the parity bit is even) on every character. This also applies to the TS during ATR.
SParCtl.5	BRKGEN	Break Generation Disable – 1 = disabled, 0 = enabled. If enabled, and T=0 protocol, the UART will generate a Break to the smart card if a parity error is detected on a receive character. No Break will be generated if parity checking is disabled. This also applies to TS during ATR.
SParCtl.4	BRKDET	Break Detection Disable – 1 = disabled, 0 = enabled. If enabled, and T=0 protocol, the UART will detect the generation of a Break by the smart card.
SParCtl.3	RETRAN	Retransmit Byte – 1 = enabled, 0 = disabled. If enabled and a Break is detected from the smart card (Break Detection must be enabled), the last character will be transmitted again. This bit applies to T=0 protocol.
SParCtl.2	DISCRX	Discard Received Byte – 1 = enabled, 0 = disabled. If enabled and a parity error is detected (Parity checking must be enabled), the last character received will be discarded. This bit applies to T=0 protocol.
SParCtl.1	INSPE	Insert Parity Error – 1 = enabled, 0 = disabled. Used for test purposes. If enabled, the UART will insert a parity error in every character transmitted by generating odd parity instead of even parity for the character.
SParCtl.0	FORCPE	Force Parity Error – 1 = enabled, 0 = disabled. Used for test purposes. If enabled, the UART will generate a parity error on a character received from the smart card.

Block Guard Time Register (BGT): 0xFE16 ← 0x10

This register contains the Extra Guard Time Value (EGT) most-significant bit. The Extra Guard Time indicates the minimum time between the leading edges of the start bit of consecutive characters. The delay is depends on the T=0/T=1 mode. Used in transmit mode. This register also contains the Block Guard Time (BGT) value. Block Guard Time is the minimum time between the leading edge of the start bit of the last character received and the leading edge of the start bit of the first character transmitted. This should not be set less than the character length. The transmission of the first character will be held off until BGT has elapsed regardless of the TX data and TX/RX control bit timing.

Table 102: The BGT Register

MSB LSB

EGT.8	–	–	BGT.4	BGT.3	BGT.1	BGT.2	BGT.0

Bit	Symbol	Function
BGT.7	EGT.8	Most-significant bit for 9-bit EGT timer. See EGT below.
BGT.6	–	
BGT.5	–	
BGT.4	BGT.4	Time in ETUs between the start bit of the last received character to start bit of the first character transmitted to the smart card. Default value is 22.
BGT.3	BGT.3	
BGT.2	BGT.2	
BGT.1	BGT.1	
BGT.0	BGT.0	

Extra Guard Time Register (EGT): 0xFE17 ← 0x0C

This register contains the Extra Guard Time Value (EGT) least-significant byte. The Extra Guard Time indicates the minimum time between the leading edges of the start bit of consecutive characters. The delay depends on the T=0/T=1 mode. Used in transmit mode.

Table 103: The EGT Register

MSB LSB

EGT.7	EGT.6	EGT.5	EGT.4	EGT.3	EGT.1	EGT.2	EGT.0

Bit	Function
EGT.7	Time in ETUs between start bits of consecutive characters. In T=0 mode, the minimum is 1. In T=0, the leading edge of the next start bit may be delayed if there is a break detected from the smart card. Default value is 12. In T=0 mode, regardless of the value loaded, the minimum value is 12, and for T=1 mode, the minimum value is 11.
EGT.6	
EGT.5	
EGT.4	
EGT.3	
EGT.2	
EGT.1	
EGT.0	

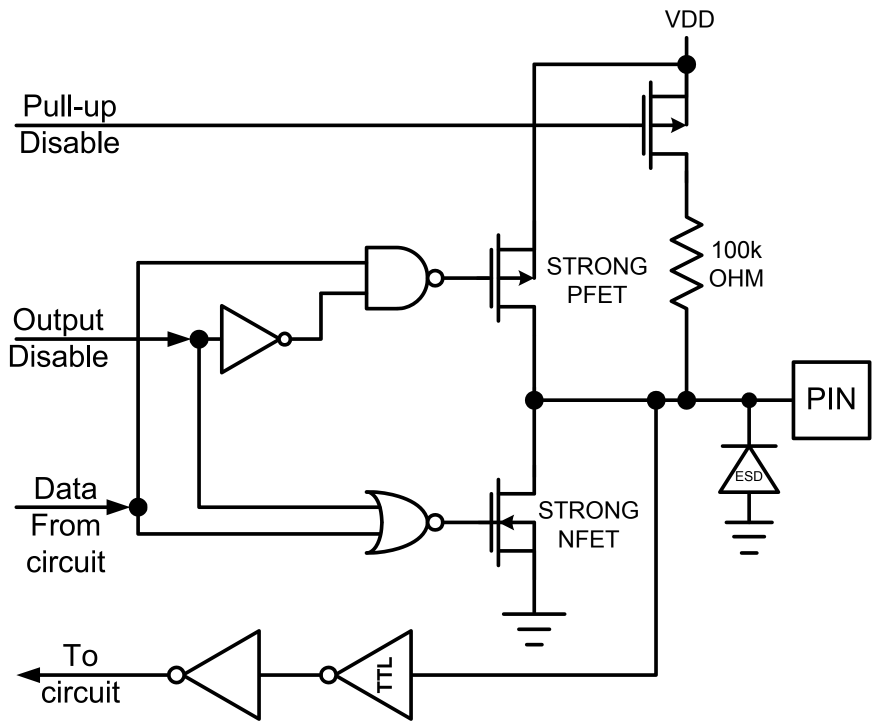


Figure 37: Keypad Row Circuit

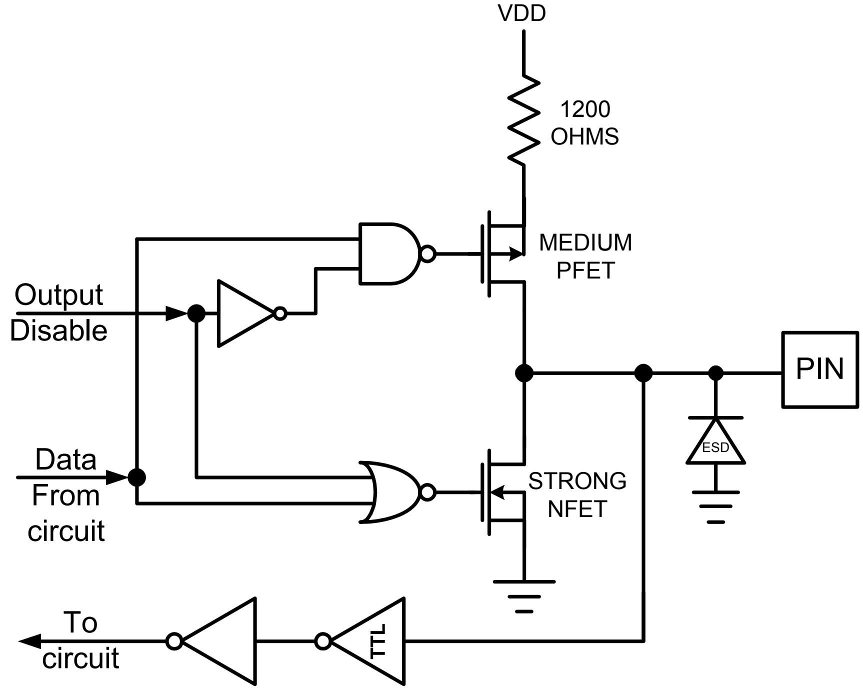


Figure 38: Keypad Column Circuit