# SPICoder™ UR5HCSPI

## Extremely Low-Power Keyboard Encoder & Power Management IC for H/PCs

**HID & SYSTEM MANAGEMENT PRODUCTS, H/PC IC FAMILY**

## DESCRIPTION

The SPICoder™ UR5HCSPI keyboard encoder and power management IC is designed specifically for handheld PCs (H/PCs). The off-the-shelf SPICoder™ will readily work with CPUs designed for Windows® CE, saving OEMs significant development time and money as well as minimizing time-to-market for the new generations of handheld products.

Three main design features of the SPICoder™ make it the ideal companion for the new generation of Windows® CE-compatible, single-chip computers: low-power consumption; real estate-saving size; and special keyboard modes.

Extremely low power consumption (less than 2 µA at 3 Volts), a must for H/PCs, provides the host system with both power management and I/O flexibility, with almost no battery drainage.

Finally, special keyboard modes and built-in power management features allow the SPICoder™ to operate in harmony with the power management modes of Windows® CE, resulting in more user flexibility and longer battery life.

The SPICoder™ also offers programmable features for wake-up keys and general purpose I/O pins.

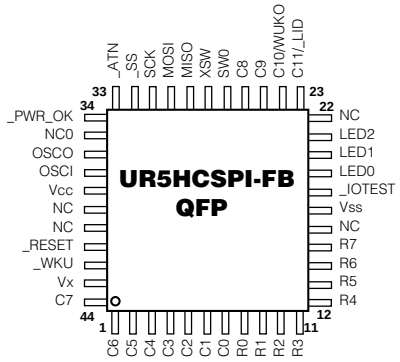
## FEATURES

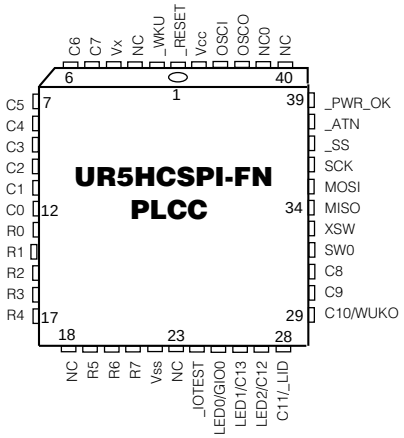
- SPI-compatible keyboard encoder and power management IC
- Compatible with Windows® CE keyboard specification
- Extremely low power consumption — typically consuming less than 2 µA between 3-5V
- Offers overall system power management capabilities
- Compatible with “system-on silicon” CPUs for H/PCs
- Special keyboard and power management modes for H/PCs, including programmable “wake-up” keys
- Scans, debounces, and encodes an 8 x 12 matrix and controls discrete switches and LED indicators

## APPLICATIONS

- StrongARM™ handheld PCs
- Windows® CE platforms
- Web phones
- Personal digital assistants (PDAs)
- Wearable computers
- Internet appliances

## PIN ASSIGNMENTS



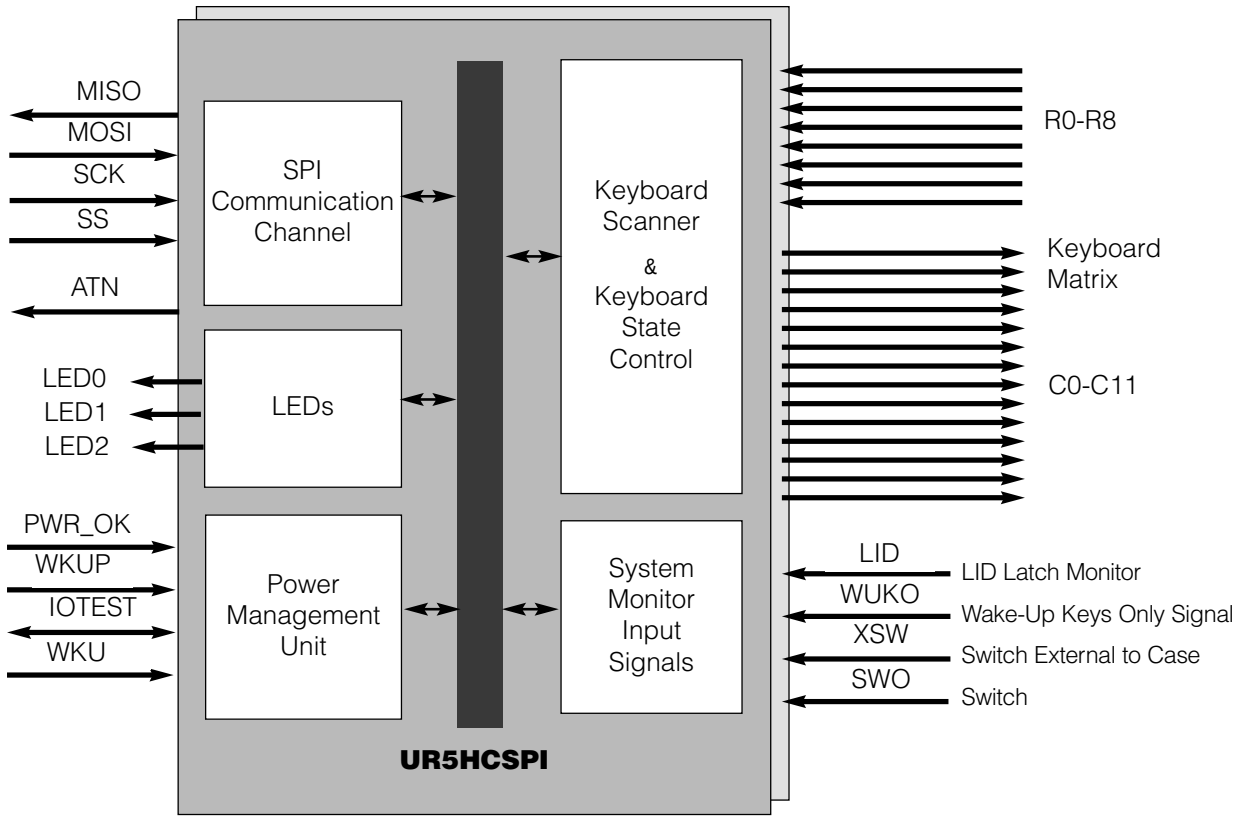
## ORDERING CODE

| Package Options | Pitch in mm | TA = -20° C to +85° C |
|---|---|---|
| 44-pin, Plastic PLCC | 1.27 mm | UR5HCSPI-XX-FN |
| 44-pin, Plastic QFP | 0.8 mm | UR5HCSPI-XX-FB |

| Other Materials | Type | Order number |
|---|---|---|
| SPICoder™ Eval. Board | Evaluation Board | ASY5-SPI-XXX |

**Note:** XX = different model/feature set, XXX = revision number

## BLOCK DIAGRAM



SEMTECH

## FUNCTIONAL DESCRIPTION

The SPICoder™ consists functionally of five major sections as shown in the the block diagram. These are the Keyboard Scanner and State control, the LEDs, the SPI Communication Channel, the System Monitor and the Power Management unit. All sections communicate with each other and operate concurrently.

## PIN DEFINITIONS

| Mnemonic | PLCC | QFP | | Type | Name and Function |
|---|---|---|---|---|---|
| VCC | 44 | 38 | | | **Power Supply:** 3-5V |
| VSS | 22 | 17 | | I | **Ground** |
| VX | 4 | 43 | | I | Tie to VCC |
| OSCI | 43 | 37 | | I | **Oscillator input** |
| OSCO | 42 | 36 | | O | **Oscillator output** |
| _RESET | 1 | 41 | | I | **Reset:** apply 0V for orderly start up |
| | | | | | **SPI Interface Signals** |
| MISO | 34 | 29 | | O | Master In, Slave Out |
| MOSI | 35 | 30 | | I | Master Out, Slave In |
| SCK | 36 | 31 | | I | SPI clock |
| _SS | 37 | 32 | | I | Slave Select: If not used tie to VSS |
| _IOTEST | 24 | 18 | | O | **Wake-Up Control Signals** |
| _WKU | 2 | 42 | | I | |
| R0-R4 | 13-17 | 8-12 | | I | **Row Data Inputs** |
| R5-R7 | 19-21 | 13-15 | | I | Port provides internal pull-up resistors |
| C0-C5 | 12-7 | 7-2 | | O | **Column Select Outputs** |
| C6-C7 | 6-5 | 1,44 | | O | |
| C8-C9 | 31-30 | 26-25 | | O | |
| | | | | | **Multi-function pins** |
| C10/WUKO | 29 | 24 | | I/O | C10 & “Wake-Up Keys Only” imput |
| C11/_LID | 28 | 23 | | I/O | C11 & lid close detect input |
| | | | | | **Miscellaneous functions** |
| LED2 | 27 | 21 | | I/O | LED2 output |
| LED1 | 26 | 20 | | I/O | LED1 output |
| LED0 | 25 | 19 | | I/O | LED0 output |
| XSW | 33 | 28 | | I | External discrete switch |
| SWO | 32 | 27 | | I | Discrete switch |
| | | | | | **Power Management Pins** |
| _ATN | 38 | 33 | | O | CPU Attention Output |
| _PWR_OK | 39 | 34 | | I | Power OK Input |
| NC | 3,18 | 39-40 | | | **No Connects**: these pins are unused |
| | 23,40 | 16,22 | | | |
| NC0 | 41 | 35 | | | NC0 should be tied to VSS or GND |

**Note:** An underscore before a pin mnemonic denotes an active low signal.

## PIN DESCRIPTIONS

### VCC and VSS

VCC and VSS are the power supply and ground pins. The SPICoder™ will operate from a 3-5 Volt power supply. To prevent noise problems, provide bypass capacitors and place them as close as possible to the IC with the power supply. VX, where available, should be tied to Vcc.

### OSCI and OSCO

OSCI and OSCO provide the input and output connections for the on-chip oscillator. The oscillator can be driven by any of the following circuits:

- Crystal
- Ceramic resonator
- External clock signal

The frequency of the on-chip oscillator is 2.00 MHz.

### _RESET

A logic zero on the _RESET pin will force the SPICoder™ into a known start-up state. The reset signal can be supplied by any of the following circuits:

- RC
- Voltage monitor
- Master system reset

### MOSI, MISO, SCK, _SS, _ATN

These five signals implement the SPI interface. The device acts as a slave on the SPI bus. The _SS (Slave Select) pin should be tied to ground if not used by the SPI master. The _ATN pin is asserted low each time the SPICoder™ has a packet ready for delivery. For a more detailed description, refer to the SPI Communication Channel section of this document.

### _IOTEST and _WKU

“Input Output Test” and “Wake Up” pins control the stop mode exit of the device. The designer can connect any number of active low signals to these two pins through a 17K resistor, in order to force the device to exit the stop mode. A sample circuit is shown on page 15 of this document.

All the signals are “wire-anded.” When any one of these signals is not active, it should be floating (i.e., these signals should be driven from “open-collector” or “open-drain” outputs). Other configurations are possible; contact Semtech.

### R0 - R7

The R0-R7 pins are connected to the rows of the scanned matrix. Each pin provides an internal pull-up resistor, eliminating the need for external components.

### C0 - C9

C0 to C9 are bi-directional pins connected to the columns of the scanned matrix. When a column is selected, the pin outputs an active low signal. When the column is de-selected, the pin turns into high-impedance.

### C10 / WUKO

The C10 / WUKO pin acts alternatively as column scan output and as an input. As an input, the pin detects the “Wake-Up Keys Only” signal, typically provided by the host CPU to indicate that the user has turned the unit off. When the device detects an active high state on this pin, it feeds this information into the “Keyboard State Control” unit, in order to disable the keyboard and enable the programmed wake-up keys.

### C11 / _LID

The C11 / _LID pin acts in a similar manner to the C10 / WUKO. This pin is typically connected to the LID latch through a 150K resistor, in order to detect physical closing of the device cover. When the pin detects an active low state in this input, it feeds this information into the “Keyboard State Control” unit, in order to disable keys inside the case and enable only switches located physically on the outer body of the H/PC unit.

### LED0, LED1 and LED2

These three pins provide an active low drive for LED indicators. The programming of these pins is explained in the LEDs section on page 8 of this document.

## PIN DESCRIPTIONS (CONT'D)

### XSW

The XSW pin is dedicated to an external switch. This pin is handled differently than the rest of the switch matrix and is intended to be connected to a switch physically located on the outside of the unit.

### SW0

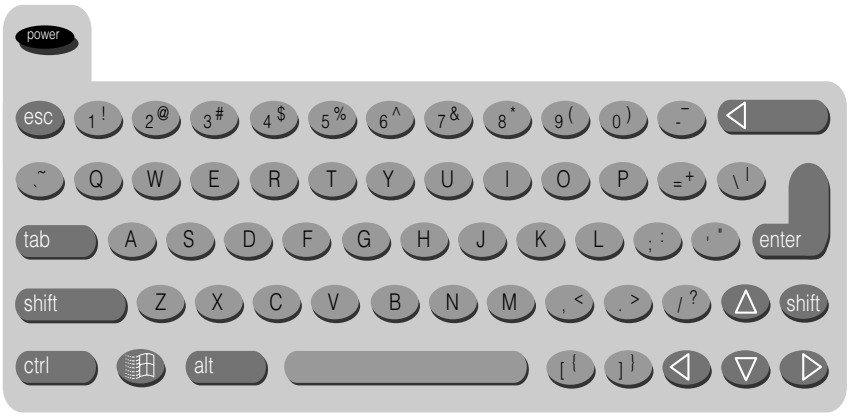
The SW0 pin is a dedicated input pin for a switch.

### PWR_OK

The PWR_OK is an active low pin that monitors the battery status of the unit. When the SPICoder™ detects a transition from high to low on this pin, it will immediately enter the STOP mode, turn the LED off and remain in this state until the batteries of the unit are replaced and the signal is deasserted.

## THE WINDOWS® CE KEYBOARD

The following illustration shows a typical implementation of a Windows® CE keyboard.




Windows® CE does not support the following keyboard keys typically found on desktop and laptop keyboards:

- INSERT
- SCROLL LOCK
- PAUSE
- NUM LOCK
- Function Keys (F1-F12)
- PRINT SCREEN

If the keyboard implements the Windows key, the following key combinations are supported in the Windows® CE environment:

| Key Combination | Result |
|---|---|
| Windows | Open Start Menu |
| Windows+K | Open Keyboard Tool |
| Windows+I | Open Stylus Tool |
| Windows+C | Open Control Panel |
| Windows+E | Explore the H/PC |
| Windows+R | Display the Run Dialog Box |
| Windows+H | Open Windows® CE Help |
| Ctrl+Windows+A | Select all on desktop |

## "GHOST" KEYS

In any scanned contact switch matrix, whenever three keys defining a rectangle on the switch matrix are pressed at the same time, a fourth key positioned on the fourth corner of the rectangle is sensed as being pressed. This is known as the "ghost" or "phantom" key problem.

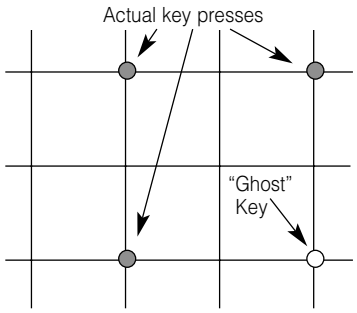



**Figure 1**: "Ghost" or "Phantom" Key Problem

Although the problem cannot be totally eliminated without using external hardware, there are methods to neutralize its negative effects for most practical applications. Keys that are intended to be used in combinations should be placed in the same row or column of the matrix, whenever possible. Shift keys (Shift, Alt, Ctrl, Window) should not reside in the same row (or column) as any other keys. The SPICoder™ has built-in mechanisms to detect the presence of "ghost" keys.

## KEYBOARD SCANNER

The encoder scans a keyboard organized as an 8 row by 12 column matrix for a maximum of 96 keys. Smaller size matrixes can also be accommodated by simply leaving unused pins open. The SPICoder™ provides internal pull-ups for the Row input pins. When active, the encoder selects one of the column lines (C0-C11) every 512 µS and then reads the row data lines (R0-R7). A key closure is detected as a zero in the corresponding position of the matrix.

A complete scan cycle for the entire keyboard takes approximately 9.2 ms. Each key found pressed is debounced for a period of 20 ms. Once the key is verified, the corresponding key code(s) are loaded into the transmit buffer of the SPI communication channel.

## N-KEY ROLLOVER

In this mode, the code(s) corresponding to each key press are transmitted to the host system as soon as that key is debounced, independent of the release of other keys.

When a key is released, the corresponding break code is transmitted to the host system. There is no limitation to the number of keys that can be held pressed at the same time. However, two or more key closures, occurring within a time interval of less than 5 ms, will set an error flag and will not be processed. This feature is to protect against the effects of accidental key presses.

## KEYBOARD STATES

These states of operation refer only to the keyboard functionality and, although they are related to power states, they are also independent of them.

### "Send All Keys"

Entry Conditions: Power on reset, soft reset, PWR_OK =1, {(LID=1) AND (WUKO=0)}

Exit Conditions: PWR_OK = 0 -> "Send No Keys"(WUKO=1) AND (Key Press) -> "Send Wake-Up Keys Only"(LID = 0) AND (WUKO=0) AND (Key Press) -> "Send XSW Key Only"

Description: This is the SPICoder™'s normal state of operation, accepting and transmitting every key press to the system. This state is entered after the power-on and is sustained while the unit is being used.

### "Send Wake-Up Keys Only"

Entry Conditions: (WUKO=1) AND (Key or Switch press)

Exit Conditions: Soft Reset -> "Send All Keys"PWR_OK = 0 -> "Send No Keys"

Description: This state is entered when the user turns the unit off. A signal line driven by the host will notify the UR5HCSPI about this state transition. While in this state, the UR5HCSPI will transmit only keys programmed to be wake-up keys to the system. It is not necessary for the UR5HCSPI to detect this transition in real time, since it does not effect any operation besides buffering keystrokes.

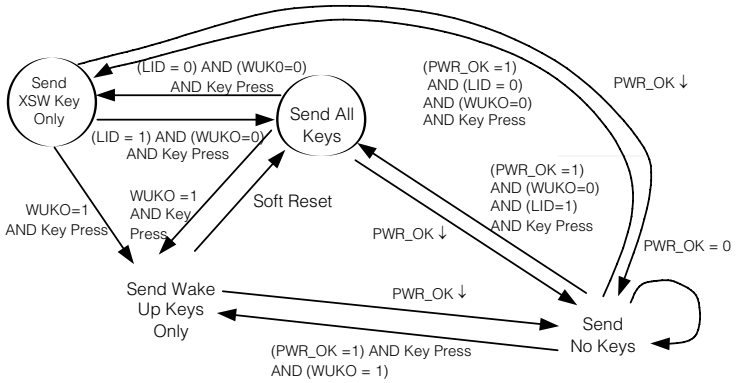



**Figure 2:** The UR5HCSPI implements four modes of keyboard and switch operation.

### "Send No Keys"

Entry Conditions: PWR_OK transition from high to low

Exit Conditions: (PWR_OK = 1) AND (Matrix key pressed OR Switch OR _WKUP)

Description: This state is entered when a PWR_OK signal is asserted (transition high to low), indicating a critically low level of battery voltage. The PWR_OK signal causes an interrupt to the SPICoder™, which guarantees that the transition is performed in real time. In this state, the SPICoder™ will perform as follows:

1. The LED is turned off. Nevertheless, its state is saved and restored after exiting the disabled state (change of batteries).

2. The SPICoder™ enters the STOP mode for maximum energy conservation.

3. Stop mode time-out entry is shortened to conserve energy further.

4. While in this state all interrupts are disabled. The SPICoder™ exits this state on the next interrupt event that detects the PWR_OK line is de-asserted.

### "Send XSW Key Only"

Entry Condition: (LID=0) AND (WUKO=0) AND (Key Press)

Exit Condition: (LID=1) AND (WUKO=0) AND (Key Press) -> "Send All Keys"PWR_OK = 0 -> "Send No Keys"
(WUKO = 1) AND (Key Press) -> "Send Wake Up Keys Only"

Description: This state is entered upon closing the lid of the device. While in this state, the encoder transmits only the XSW key, which is located outside the unit. This feature is designed to accommodate buttons on the outside of the box, such as a microphone button, that need to be used while the lid is closed.

## KEY CODES

Key codes range from 01H to 73H and are arranged as follows:

Make code = column_number * 8 + row_number + 1

Break code = Make code OR 80H

Discrete Switches transmit the following codes:

XSW = 71H

SW0 = 72H

## LED MODES

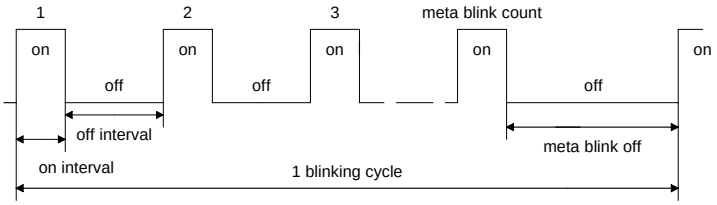



**Figure 3**: The behavior of an LED using the settings 1: LED on; 0: LED off.

The SPICoder™ provides three LED pins. There are three LED modes: off, on, and blinking. The LED can be individually set to one of these modes. In the blinking mode, both the on-interval and the off-interval can be individually set. Additionally, a meta blink count and meta blink interval may be specified. This describes an interval of a different length which may be inserted after each specified number of blinks. All the intervals are based on a 1/16th of a second duration. When the LED is on or blinking, the SPICoder™ does not enter the STOP Mode unless the PWR_OK signal is asserted low. In this case, the device saves the status of the LED and turns it off. The default LED mode is off.

The above timing chart describes the behavior of an LED using these settings,1: LED on; 0: LED off.

## SPI COMMUNICATION CHANNEL

SPI data transfers can be performed at a maximum clock rate of 500 KHz. When the SPICoder™ asserts the _ATN signal to the host master, the data is already loaded into the data register waiting for the clocks from the master. The Slave Select (SS) line can be tied permanently to ground if the SPICoder™ is the only slave device in the SPI network. One _ATN signal is used per each byte transfer. If the host fails to provide clock signals for successive bytes in the data packet within 120 ms, the transmission is aborted and a new session is initiated by asserting a new ATN signal. In this case, the whole packet is re-transmitted.

If the SPI transmission fails 20 times consecutively, the synchronization between the master and slave may be lost. In this case, the SPICoder™ enters the reset state.

The SPICoder™ implements the SPI communication protocol according to the following diagram:

CPOL = 0 ---------- SCK line idles in low state

CPHA = 1 ---------- SS line is an output enable control

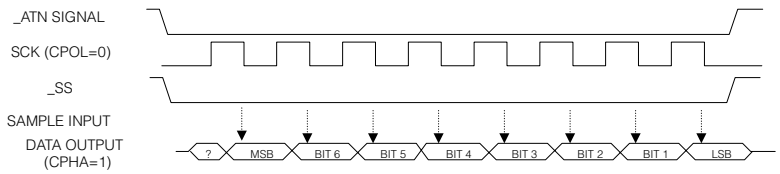



**Figure 4:** SPI Communication Protocol

When the host sends commands to the keyboard, the SPICoder™ requires that the minimum and maximum intervals between two successive bytes be 200 µs and 5 ms respectively.

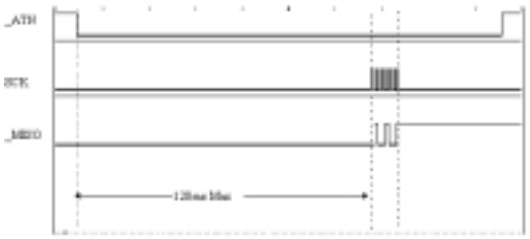



**Figure 5:** Transmitting Data Waveforms:

**Figure 6:** Receiving Data Waveforms

## DATA / COMMAND BUFFER

The SPICoder™ implements a data buffer that contains the key code/command bytes waiting to be transmitted to the host. If the data buffer is full, the whole buffer is cleared and an "Initialize" command is sent to the host. At the same time, the keyboard is disabled until the "Initialize" or "Initialize Complete" command from the host is received.

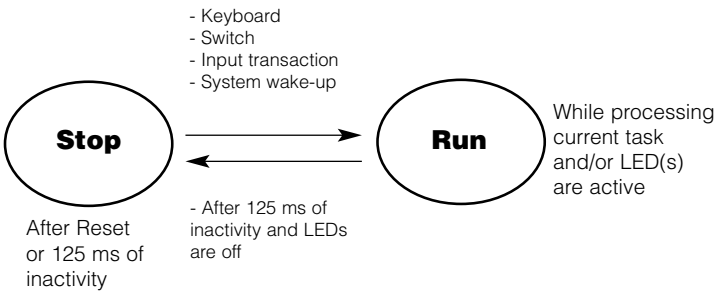
## POWER MANAGEMENT UNIT

The SPICoder™ supports two modes of operation. The following table lists the typical and maximum supply current (no DC loads) for each mode at 3.3 Volts (+/- 10%).

| Current | Typical | Max | Unit | Description |
|---|---|---|---|---|
| RUN | 1.5 1 | 3.0 | mA | Entered only while data/commands are in process and if the LEDs are blinking |
| STOP | 2.0 | 20 | µA | Entered after 125 ms of inactivity if LEDs is low |

Power consumption of the keyboard sub-system is determined primarily by the use of the LEDs. While the SPICoder™ is in the STOP mode, an active low Wake-Up output from the master must be connected to the edge-sensitive _WKU pin of the SPICoder™. This signal wakes up the SPICoder™ in order to receive data from the master host. The master host needs to wait a minimum of 5 ms prior to providing clocks to the SPICoder™. The SPICoder™ enters the STOP mode after a 125 ms period of keypad and/or host communications inactivity, or anytime the PWR_OK line is asserted low by the host. Note that while one or more keys are held pressed, the SPICoder™ does not enter the STOP mode until every key is released.

**Figure 7:** The power states of the SPICoder™



## COMMUNICATION PROTOCOL

There are eight commands that may be sent from the SPICoder™ to the host, and ten commands that may be sent from the host to the SPICoder™.

Each command from SPICoder™ to the host is composed of a sequence of codes. All commands start with <CONTROL> code (80H) and end with LRC code (see the description of the LRC calculation on page 12). Command details are listed below.

**Commands to the Host - Summary**

| Command Name | Code | Description |
|---|---|---|
| Initialize Request | AOH | Sent to the host when the data buffer is full |
| Initialize Complete | A1H | Issued upon completion of the “Initialize” command issued by the host |
| Heartbeat Response | A2H | Response to “Heartbeat Request” issued by the host |
| Identification Response | F2H | Response to “Identification Request” issued by the host |
| LED Status Report | A3H | Response to “LED Status Request” |
| Resend Request | A5H | Issued upon error during the reception of a packet |

## LRC CALCULATION

The LRC is calculated for the whole packet, including the command code and the command prefix. The LRC is calculated by first taking the bitwise exclusive OR of all bytes from the message. If the most significant bit (MSB) of the LRC is set, the LRC is modified by clearing the MSB and changing the state of the next most significant bit. Thus, the packet check byte will never consist of a valid LRC with the most significant bit set.

## COMMANDS TO THE HOST ANALYTICALLY

### Initialize Request

| | |
|---|---|
| <CONTROL> | 80H |
| <INIT> | A0H |
| <LRC> | 20H |

The SPICoder™ sends the initialize request command to the host when its data buffer is full.

### Initialization Complete

| | |
|---|---|
| <CONTROL> | 80H |
| <INIT COMPLETE> | A1H |
| <LRC> | 21H |

The SPICoder™ sends the initialize complete report to the host when it finishes the initialization caused by initialize command from the host.

### Heartbeat Response

| | |
|---|---|
| <CONTROL> | 80H |
| <ONLINE> | A2H |
| <LRC> | 22H |

The SPICoder™ sends the heartbeat response to the host when it receives the heartbeat request command from the host.

### Identification Response

| | | |
|---|---|---|
| <CONTROL> | 80H | |
| <ID> | F2H | |
| <Vendor> | 02H | ---Semtech |
| <Revision> | 08H | --- Rev 0.8A |
| <Switch> | 00H | . |
| <LRC> | 7EH | |

The SPICoder™ sends the identification response to the host when it receives the identification request command from the host.

## LRC CALCULATION (CONT'D)

The following C language function is an example of an LRC calculation program. It accepts two arguments: a pointer to a buffer and a buffer length. Its return value is the LRC value for the specified buffer.

```
char Calculate LRC (char  buffer,
size  buffer)
{
char LRC;
size_t index;
/*
* Init the LRC using the first two
message bytes.
*/
LRC = buffer [0] ^ buffer [1];
/*
* Update the LRC using the
remainder of the buffer.
*/
for (index = 2; index < buffer; index
++)
  LRC ^ = buffer[index];
/*
* If the MSB is set then clear the
MSB and change the next most
significant bit
*/
if (LRC & 0x80)
LRC ^ = 0xC0;
/* * Return the LRC value for the
buffer.*/}
```

## COMMANDS FROM THE SPICODER™ TO THE HOST (CONT'D)

### LED Status Report

| | | |
|---|---|---|
| <CONTROL> | 80H | |
| <LED> | A3H | |
| <Status 0> | xxH | LED0 status:( 0=OFF; 1=ON; 2=BLINKING; 3=NO LED MODE ) |
| <Status 1> | xxH | LED1 status:( 0=OFF; 1=ON; 2=BLINKING; 3=NO LED MODE ) |
| <Status 2> | xxH | LED2 status:( 0=OFF; 1=ON; 2=BLINKING; 3=NO LED MODE) |
| <LRC> | xxH | |

The SPICoder™ will send the LED status report to the host when it receives the LED status request command from the host.

### Resend Request

| | |
|---|---|
| <CONTROL> | 80H |
| <RESEND> | A5H |
| <LRC> | 25H |

The SPICoder™ will send this resend request command to the host when its command buffer is full, or if it detects either a parity error or an unknown command during a system command transmission.

## COMMANDS FROM THE HOST TO THE SPICODER™

**Commands from the Host - Summary**

| Command Name | Code | Description |
|---|---|---|
| Initialize | AOH | Causes the SPICoder™ to enter the power-on state |
| Initialization Complete | A1H | Issued as a response to the “Initialize Request” |
| Heartbeat Request | A2H | The SPICoder™ will respond with “Heartbeat Response” |
| Identification Request | F2H | The SPICoder™ will respond with “Identification Response” |
| LED Status Request | A3H | The SPICoder™ will respond with “LED Status Response” |
| LED Modify | A6H | The SPICoder™ will change the LED accordingly |
| Resend Request | A5H | Issued upon error during the reception of a packet |
| Input/Output Mode Modify | A7H | The SPICoder™ will modify or report the status of the GIO0 pin |
| Output Data to I/O pin | A8H | The SPICoder™ will output a signal to the GIO0 pin |
| Set Wake-Up Keys | A9H | Defines which keys are “wake-up” keys |

Each command to SPICoder™ is composed of a sequence of codes. All commands start with <ESC> code (1BH) and end with the LRC code (bitwise exclusive OR of all bytes).

## COMMANDS FROM THE HOST TO THE SPICODER™ ANALYTICALLY

### Initialize

| | |
|---|---|
| <ESC> | 1BH |
| <INIT> | A0H |
| <LRC> | 7BH |

When the SPICoder™ receives this command, it clears all buffers and returns to the power-on state.

### Initialization Complete

| | |
|---|---|
| <ESC> | 1BH |
| <INIT COMPLETE> | A1H |
| <LRC> | 7AH |

When the SPICoder™ receives this command, it enables transmission of keyboard data. Keyboard data transmission is disabled if the TX output buffer is full (32 bytes). Note that if the transmit data buffer gets full, the encoder issues an "Initialize Request" to the host.

### Heartbeat Request

| | |
|---|---|
| <ESC> | 1BH |
| <ONLINE> | A2H |
| <LRC> | 79H |

When the SPICoder™ receives this command, it replies with the heartbeat response report.

### Identification Request

| | |
|---|---|
| <ESC> | 1BH |
| <ID> | F2H |
| <LRC> | 29H |

The SPICoder™ replies to this command with the identification response report.

## COMMANDS FROM THE HOST TO THE SPICODER™ (CONT'D)

### Set Wake-Up Keys

| | | |
|---|---|---|
| <ESC> | 1BH | |
| <SETMATRIX> | A9H | |
| <COL0> | xxH | (xxH = bitmap for R7 R6 R5 R4 R3 R2 R1 R0: 0-enabled, 1-disabled) |
| <COL1> | xxH | |
| <COL2> | xxH | |
| <COL3> | xxH | |
| <COL4> | xxH | |
| <COL5> | xxH | |
| <COL6> | xxH | |
| <COL7> | xxH | |
| <COL8> | xxH | |
| <COL9> | xxH | |
| <COL10> | xxH | |
| <COL11> | xxH | |
| <SWITCHES> | xxH | |
| <LRC> | xxH | |

The "Set Wake-Up Keys" command is used to disable specific keys from waking up the host. Using this command, the host can set only a group of keys.

### LED Status Request

| | | |
|---|---|---|
| <ESC> | 1BH | |
| <LED> | A3H | |
| <LRC> | 78H | |

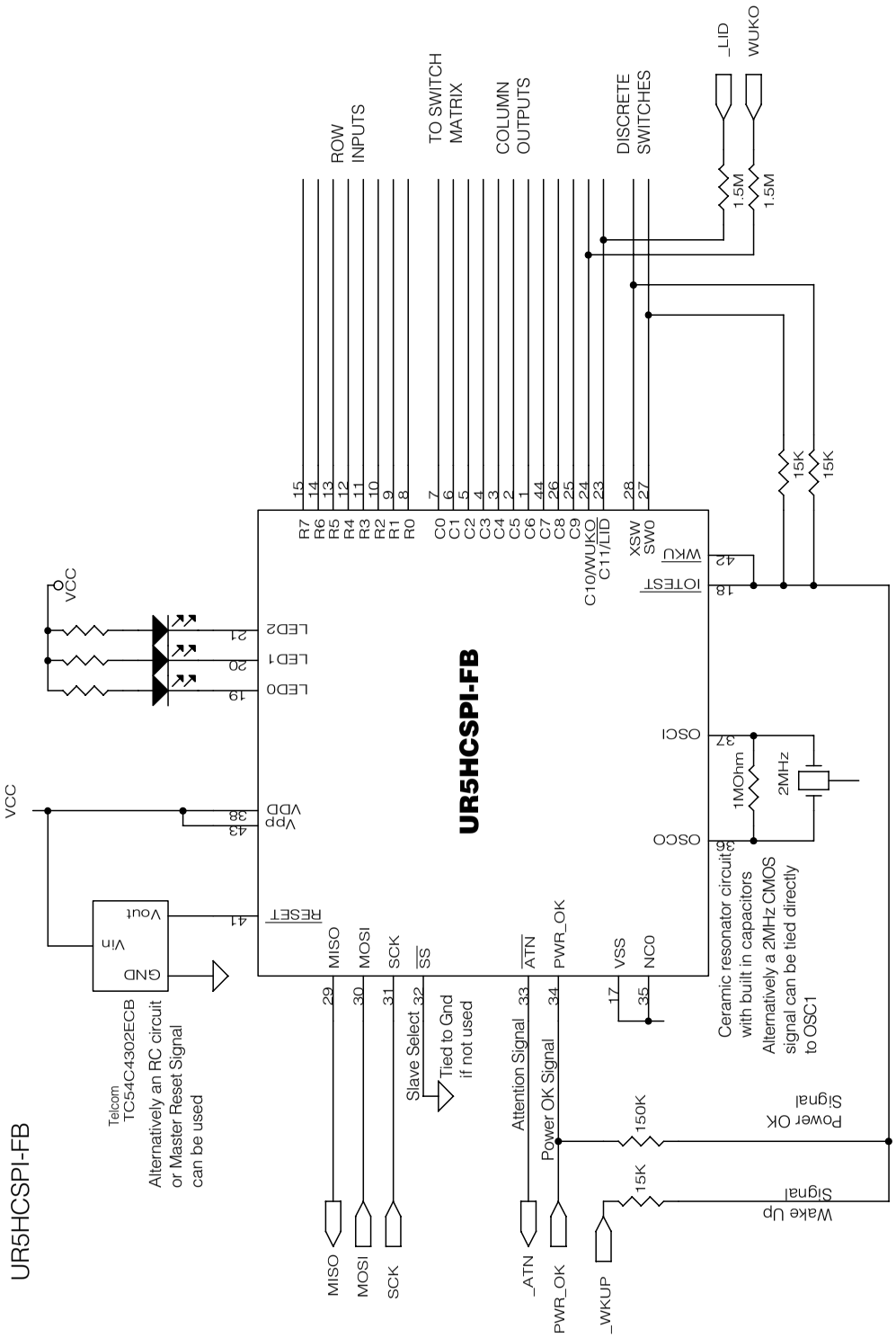
When the SPICoder™ receives this command, it replies with the LED Status Report.

### LED Modify

| | | |
|---|---|---|
| <ESC> | 1BH | |
| <MODLED> | A6H | |
| <LED NUMBER> | xxH | LED number (0) |
| <LED STATE> | xxH | (0=LED OFF; 1=LED ON; 2=LED BLINKING) |
| <ON INTERVAL> | xxH | Time in 1/16ths of a second for LED to be on |
| <OFF INTERVAL> | xxH | Time in 1/16ths of a second for LED to be off |
| <META COUNT> | xxH | Number of blinks after which to apply meta blink interval |
| <META INTERVAL> | xxH | Time in 1/16ths of a second for LED to be off after <META COUNT> blinks |
| <LRC> | xxH | |

When the SPICoder™ receives this command, it changes the LED mode accordingly.

## SUGGESTED SCHEMATIC FOR THE UR5HCSPI-FB



## IMPLEMENTATION NOTES FOR THE SPICODER™

The following notes pertain to the suggested schematic found on the previous page.

The built-in oscillator on the SPICoder™ requires the attachment of a 2.00 MHz ceramic resonator with built-in load capacitors. You can use either an AVX, part number PBRC-2.00 BR; or a Murata part number CSTCC2.00MG ceramic resonator.

It may also be possible to operate with the 2.00 MHz crystal, albeit with reduced performance. Due to their high Q, the crystal oscillator circuits start-up slowly. Since the SPICoder™ constantly switches the clock on and off, it is important that the ceramic resonator is used (it starts up much quicker than the crystal). Resonators are also less expensive than crystals.

Also, if crystal is attached, two load capacitors (33 pF to 47pF) should be added, a capacitor between each side of the crystal and ground.

In both cases, using ceramic resonator with built-in load capacitors, or crystal with external load capacitors, a feedback resistor of 1 MegaOhm should be connected between OSCI and OSCO.

Troubleshoot the circuit by looking at the output pin of the oscillator. If the voltage is half-way between supply and ground (while the oscillator should be running) --- the problem is with the load caps / crystal. If the voltage is all the way at supply or ground (while the oscillator should be running) --- there are shorts on the PCB.

Note: When the oscillator is intentionally turned OFF, the voltage on the output pin of the oscillator is high (at the supply rail).

## ELECTRICAL SPECIFICATIONS

**Absolute Maximum Ratings**

| Ratings | Symbol | | Value | Unit |
|---|---|---|---|---|
| Supply Voltage | Vdd | | -0.3 to +7.0 | V |
| Input Voltage | Vin | | Vss -0.3 to Vdd +0.3 | V |
| Current Drain per Pin (not including Vss or Vdd) | I | | 25 | mA |
| Operating Temperature UR5HCSPI | Ta | | T low to T high -40 to +85 | ° C |
| Storage Temperature Range | Tstg - | | -65 to +150 | ° C |

**Thermal Characteristics**

| Characteristic | Symbol | Value | Unit |
|---|---|---|---|
| Thermal Resistance | Tja | | °C per W |
| ■ Plastic | | 60 | |
| ■ PLCC | | 70 | |

**DC Electrical Characteristics** (Vdd=3.3 Vdc +/-10%, Vss=0 Vdc, Temperature range=T low to T high unless otherwise noted)

| Characteristic | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Output Voltage (I load<10µA) | Vol | | | 0.1 | V |
| | Voh | Vdd–0.1 | | | |
| Output High Voltage (I load=0.8mA) | Voh | Vdd–0.8 | | | V |
| Output Low Voltage (I load=1.6mA) | Vol: | | | 0.4 | V |
| Input High Voltage | Vih | 0.7xVdd | | Vdd | V |
| Input Low Voltage | Vil | Vss | | 0.2xVdd | V |
| User Mode Current | Ipp | | 5 | 10 | mA |
| Data Retention Mode (0 to 70°C) | Vrm | 2.0 | | | V |
| Supply Current (Run) | Idd | | 1.53 | 3.0 | mA |
| (Wait) | | | 0.711 | 1.0 | mA |
| (Stop) | | | 2.0 | 20 | µA |
| I/O Ports Hi-Z Leakage Current | Iil | | | +/-10 | µA |
| Input Current | Iin | | | +/- 1 | µA |
| I/O Port Capacitance | Cio | | 8 | 12 | pF |

**Control Timing** (Vdd=3.3 Vdc +/-10%, Vss=0 Vdc, Temperature range=T low to T high unless otherwise noted)

| Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| Frequency of Operation | fosc | | | MHz |
| ■ Crystal Option | | | 2.0 | |
| ■ External Clock Option | | dc | 2.0 | |
| Cycle Time | tcyc | 1000 | | ns |
| Crystal Oscillator Startup Time | toxov | | 100 | ms |
| Stop Recovery Startup Time | tilch | | 100 | ms |
| RESET Pulse Width | trl | 8 | | tcyc |
| Interrupt Pulse Width Low | tlih | 250 | | ns |
| Interrupt Pulse Period | tilil | * | | tcyc |
| OSC1 Pulse Width | toh, tol | 200 | | ns |

*The minimum period tlil should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 tcyc.

## SPICODER™ BILL OF MATERIALS

### UR5HCSPI-FB

| Quantity | Manufacturer | Part# | Description |
|---|---|---|---|
| 3 | Generic | 330 Ohms | 330 ohm resistor |
| 3 | Generic | LED | LED used as LED0. LED1. LED2 |
| 3 | Generic | 15 K | 15 K resistors |
| 1 | Generic | 150 K | 150 K resistors |
| 1 | Generic | 1M | 1 M resistors |
| 2 | Generic | 1.5 K | 1.5 K resistors |
| 1 | TELCOM | TC54VC4302ECB713<br>TC54VC2702ECB713 | IC volt detector CMOS 4.3V SOT23, for 5V operation<br>IC volt detector CMOS 2.7V SOT23, for 3.3V operation |
| 1 | AVX | PBRC-2.00BR | 2.00 MHZ ceramic resonator with built in capacitors, SMT |

This Page Left Intentionally Blank

SEMTECH

**For sales information and product literature, contact:**


HID & System Mgmt Division
Semtech Corporation
652 Mitchell Road
Newbury Park, CA 91320
**hidinfo@semtech.com**
**http://www.semtech.com/**
805 498 2111 Telephone
805 498 3804 Telefax

Semtech Western Regional Sales
805-498-2111 Telephone
805-498-3804 Telefax

Semtech Central Regional Sales
972-437-0380 Telephone
972-437-0381 Telefax

Semtech Eastern Regional Sales
203-964-1766 Telephone
203-964-1755 Telefax

Semtech Asia-Pacific Sales Office
+886-2-2748-3380 Telephone
+886-2-2748-3390 Telefax

Semtech Japan Sales Office
+81-45-948-5925 Telephone
+81-45-948-5930 Telefax

Semtech Korea Sales Sales
+82-2-527-4377 Telephone
+82-2-527-4376 Telefax

Northern European Sales Office
+44 (0)2380-769008 Telephone
+44 (0)2380-768612 Telefax

Southern European Sales Office
+33 (0)1 69-28-22-00 Telephone
+33 (0)1 69-28-12-98 Telefax

Central European Sales Office
+49 (0)8161 140 123 Telephone
+49 (0)8161 140 124 Telefax