

## USB 2.0 to UART Protocol Converter with GPIO

### Features

#### Universal Serial Bus (USB)

- Supports full-speed USB (12 Mb/s)
- Implements USB protocol composite device CDC device for communications, configuration and I/O control
- 128 byte buffer to handle data throughput at any UART baud rate:
  - 64 byte transmit
  - 64 byte receive
- Fully configurable VID and PID assignments, and string descriptors
- Bus powered or self-powered
- USB 2.0 Compliant (certification #: TBD)

#### USB Driver and Software Support

- Uses standard Microsoft® Windows® drivers for Virtual Com Port (VCP):
  - Windows XP(SP2 and later)/Vista/7
- Configuration utility for initial configuration

#### Universal Asynchronous Receiver/Transmitter (UART)

- Responds to SET LINE CODING commands to dynamically change baud rates
- Supports baud rates: 300-1000k
- Hardware flow control
- UART signal polarity option

#### General Purpose Input/Output (GPIO) Pins

- Eight (8) general purpose I/O pins

#### EEPROM

- 256 bytes of user EEPROM

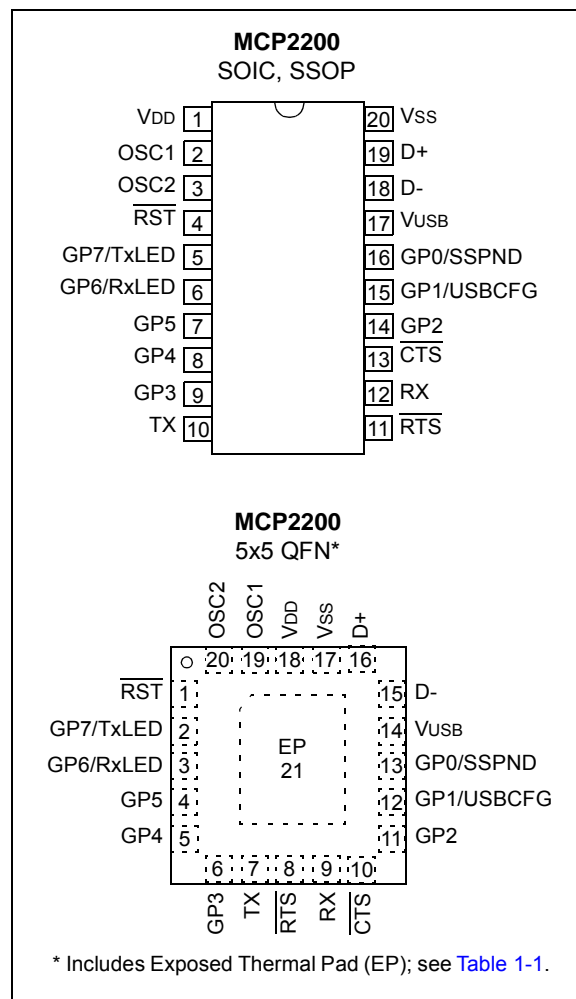
#### Other

- USB activity LED outputs (TxLED and RxLED)
- SSPND output pin
- USBCFG output pin (indicates if requested current is allowed)
- Operating voltage: 3.0-5.5V
- Oscillator input: 12 MHz
- ESD protection > 4 kV HBM
- Industrial (I) Operating Temperature: -40°C to +85°C

### Package Types

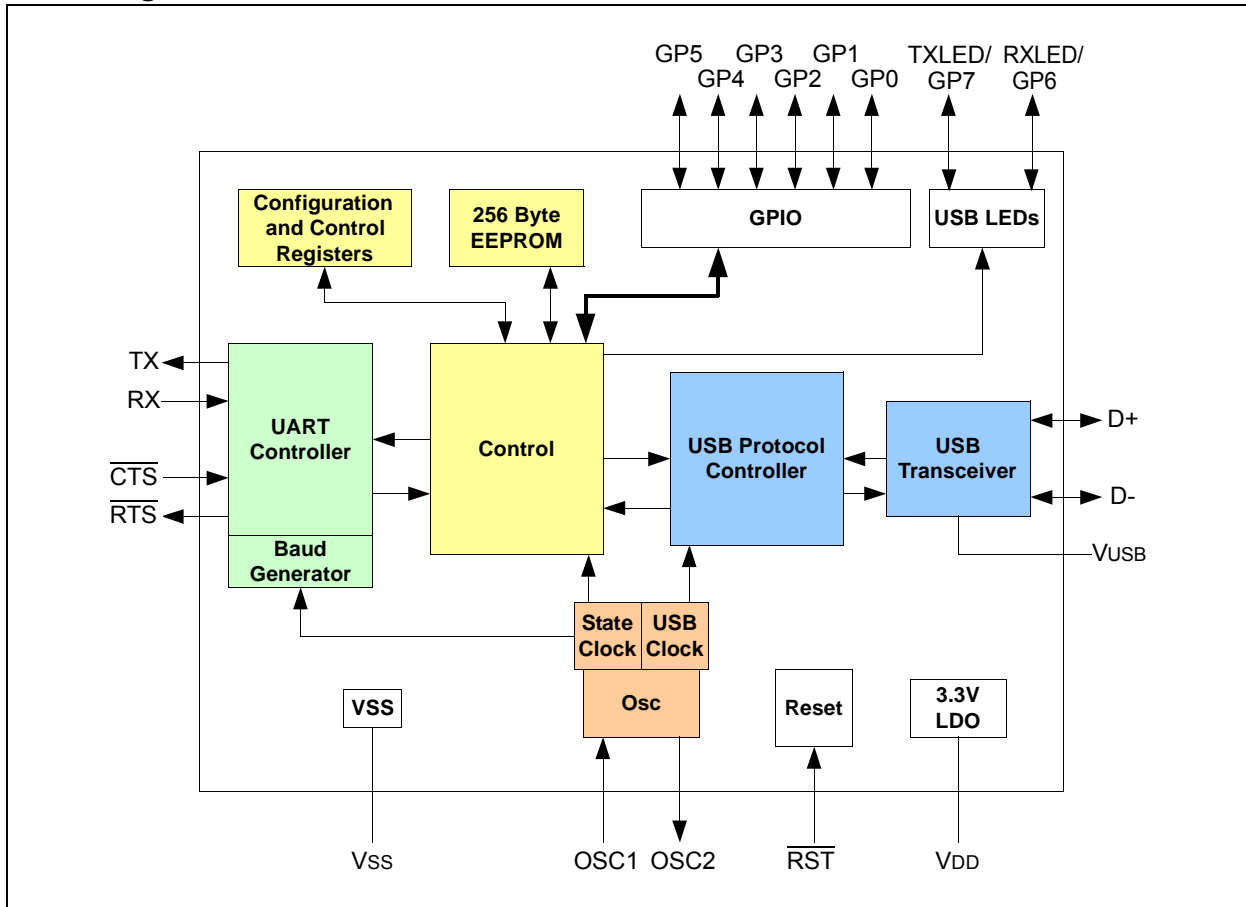
The device will be offered in the following packages:

- 20-lead QFN (5 x 5[mm])
- 20-lead SOIC
- 20-lead SSOP



# MCP2200

## Block Diagram



## 1.0 FUNCTIONAL DESCRIPTION

The MCP2200 is a USB-to-UART serial converter which enables USB connectivity in applications that have a UART interface. The device reduces external components by integrating the USB termination resistors. The MCP2200 also has 256 bytes of integrated user EEPROM.

The MCP2200 has eight general purpose input/output pins. Four pins have alternate functions to indicate USB and communication status. See [Table 1-1](#) and [Section 1.6 “GPIO Module”](#) for details about the pin functions.

**TABLE 1-1: PINOUT DESCRIPTION**

Pin Name	SSOP, SOIC	QFN	Pin Type	Standard Function	Alternate Function
GP0/SSPND	16	13	I/O	General purpose I/O	USB suspend status pin (refer to <a href="#">Section 1.6.1.1 “SSPND Pin Function”</a> )
GP1/USBCFG	15	12	I/O	General purpose I/O	USB configuration status pin (refer to <a href="#">Section 1.6.1.2 “USB-CFG Pin Function”</a> )
GP2	14	11	I/O	General purpose I/O	
GP3	9	6	I/O	General purpose I/O	
GP4	8	5	I/O	General purpose I/O	
GP5	7	4	I/O	General purpose I/O	
GP6/RxLED	6	3	I/O	General purpose I/O	USB receive activity LED output (refer to <a href="#">Section 1.6.1.3 “RxLED Pin Function (IN Message)”</a> )
GP7/TxLED	5	2	I/O	General purpose I/O	USB transmit activity LED output (refer to <a href="#">Section 1.6.1.4 “TxLED Pin Function (OUT Message)”</a> )
RST	4	1	I	Reset input must be externally biased	
CTS	13	10	I	Hardware flow control “Clear to Send” input signal	
RX	12	9	I	USART RX input	
RTS	11	8	O	Hardware flow control “Request to Send” output signal	
TX	10	7	O	USART TX output	
VDD	1	18	P	Power	
VSS	20	17	P	Ground	
OSC1	2	19	I	Oscillator input	
OSC2	3	20	O	Oscillator output	
D+	19	16	I/O	USB D+	
D-	18	15	I/O	USB D-	
VUSB	17	14	P	USB power pin (internally connected to 3.3V). Should be locally bypassed with a high quality ceramic capacitor.	
EP	—	21	—	Exposed Thermal Pad (EP). Do not electrically connect, or connect to VSS.	

# MCP2200

---

## 1.1 Supported Operating Systems

Microsoft Windows XP(SP2 and later)/Vista/7 operating systems are supported.

### 1.1.1 ENUMERATION

The MCP2200 will enumerate as a USB device after POR. The device enumerates as both a Human Interface Device (HID) for I/O control, and a VCP.

#### 1.1.1.1 HID

The MCP2200 enumerates as a HID so the device can be configured and the I/O can be controlled. A DLL is supplied by Microchip that allows I/O control using a custom interface.

#### 1.1.1.2 VCP

The VCP enumeration implements the USB-to-UART data translation.

## 1.2 Control Module

The control module is the heart of the MCP2200. All other modules are tied together and controlled via the control module. The control module manages the data transfers between the USB and the UART, as well as command requests generated by the USB host controller, and commands for controlling the function of the UART and I/O.

### 1.2.1 SERIAL INTERFACE

The control module interfaces to the UART and USB modules.

### 1.2.2 INTERFACING TO THE DEVICE

The MCP2200 can be accessed for reading and writing via USB host commands. The device cannot be accessed and controlled via the UART interface.

## 1.3 UART Interface

The MCP2200 UART interface consists of the TX and RX data signals and the RTS/CTS flow control pins.

The UART is configurable for several baud rates. The available data formats and baud rates are listed in [Table 1-3](#).

### 1.3.1 INITIAL CONFIGURATION

The default UART configuration is 19200, 8, N, 1. The default start up baud rate can be changed using the Microchip supplied configuration PC tool. The primary baud rates are shown in [Table 1-3](#).

Alternatively, a custom configuration tool can be created using the Microchip supplied DLL to set the baud rate, as well as other parameters. See [Section 2.0 “Configuration”](#) for details.

### 1.3.2 GET/SET LINE CODING

The `GET_LINE_CODING` and `SET_LINE_CODING` commands are used to read and set the UART parameters while in operation. For example, Hyperterminal sends the `SET_LINE_COMMAND` when connecting to the port. The MCP2200 responds by setting the baud rate only. The other parameters (Data Bits, Parity, Stop Bits) remain unchanged.

#### 1.3.2.1 Rounding Errors

The primary baud rate setting (with the rounding errors) are shown in [Table 1-3](#). If baud rates other than the ones shown in the table are used, the error percentage can be calculated using [Equation 1-1](#) to find the actual baud rate.

**TABLE 1-2: UART CONFIGURATIONS**

Parameter	Configuration
Primary Baud Rates	See <a href="#">Table 1-3</a>
Data Bits	8
Parity	N
Stop Bits	1

### 1.3.3 CUSTOM BAUD RATES

Custom baud rates are configured by sending the `SET_LINE_CODING` USB command, or by using the DLL. See [Section 2.0 “Configuration”](#) for more information.

## 1.3.4 HARDWARE FLOW CONTROL

Hardware flow control uses the  $\overline{\text{RTS}}$  and  $\overline{\text{CTS}}$  pins as a handshake between two devices. The  $\overline{\text{RTS}}$  pin of one device is typically connected to the CTS of the other device.

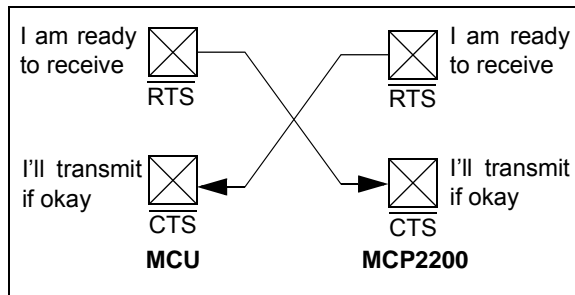
$\overline{\text{RTS}}$  is an active low output which notifies the other device when it can receive data by driving the pin low. The MCP2200 trip point for de-asserting  $\overline{\text{RTS}}$  (high) is 63 characters. This is one character short of “buffer full”.

$\overline{\text{CTS}}$  is an active low input which is used to notify the MCP2200 when it can send data. The MCP2200 will check  $\overline{\text{CTS}}$  just before loading and sending UART data. If the pin is asserted during a transfer, the transfer will continue. Refer to [Figure 1-1](#).

### 1.3.4.1 Flow Control Disabled

The buffer pointer does not increment (or reset to zero) if the buffer is full. Therefore, if hardware flow control is not enabled and an overflow occurs (i.e., 65 unprocessed characters received), the new data overwrites the last position in the buffer.

**FIGURE 1-1:  $\overline{\text{RTS}}/\overline{\text{CTS}}$  CONNECTIONS EXAMPLE**



**EQUATION 1-1: SOLVING FOR ACTUAL BAUD RATE**

$$ActualRate = \frac{12MHz}{int(x)}$$

Where:

$$x = \frac{12MHz}{DesiredBaud}$$

**TABLE 1-3: UART PRIMARY BAUD RATES**

Desired Rate	Actual rate	% Error
300	300	0.00%
1200	1200	0.00%
2400	2400	0.00%
4800	4800	0.00%
9600	9600	0.00%
19200	19200	0.00%
38400	38339	0.16%
57600	57692	0.16%
115200	115385	0.16%
230400	230769	0.16%
460800	461538	0.16%
921600	923077	0.16%

## 1.4 USB Protocol Controller

The USB controller in the MCP2200 is full-speed USB 2.0 compliant.

- Composite device (CDC + HID):
  - CDC: USB-to-UART communications
  - HID: I/O control, EEPROM access and initial configuration
- 128 byte buffer to handle data throughput at any UART baud rate:
  - 64 byte transmit
  - 64 byte receive
- Fully configurable VID and PID assignments, and descriptors (stored on-chip)
- Bus powered or self-powered

### 1.4.1 DESCRIPTORS

The descriptors are stored in the MCP2200 during configuration using the supplied PC interface.

### 1.4.2 SUSPEND AND RESUME

The USB Suspend and Resume signals are supported for power management of the MCP2200. The device will enter Suspend mode when suspend signaling is detected on the bus.

The MCP2200 exits the Suspend mode when any of the following occur:

1. Resume signaling is detected or generated
2. A USB Reset signal is detected
3. A device Reset occurs

# MCP2200

## 1.5 USB Transceiver

The MCP2200 has a built-in, USB 2.0, full-speed transceiver internally connected to the USB module.

The USB transceiver obtains power from the V<sub>USB</sub> pin, which is internally connected to the 3.3V regulator. The best electrical signal quality is obtained when V<sub>USB</sub> is locally bypassed with a high quality ceramic capacitor.

### 1.5.1 INTERNAL PULL-UP RESISTORS

The MCP2200 devices have built-in pull-up resistors designed to meet the requirements for full-speed USB.

#### 1.5.1.1 Bus Power Only

In Bus Power Only mode, all power for the application is drawn from the USB (Figure 1-2). This is effectively the simplest power method for the device.

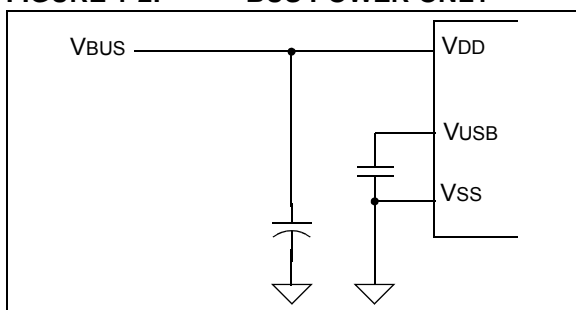
In order to meet the inrush current requirements of the USB 2.0 specifications, the total effective capacitance appearing across V<sub>BUS</sub> and ground must be no more than 10  $\mu$ F. If not, some kind of inrush limiting is required. For more details, see Section 7.2.4 of the “Universal Serial Bus Specification”.

According to the USB 2.0 specification, all USB devices must also support a Low-Power Suspend mode. In the USB Suspend mode, devices must consume no more than 500  $\mu$ A (or 2.5 mA for high powered devices that are remote wake-up capable) from the 5V V<sub>BUS</sub> line of the USB cable.

The host signals the USB device to enter Suspend mode by stopping all USB traffic to that device for more than 3 ms.

During USB Suspend mode, the D+ or D- pull-up resistor must remain active, which will consume some of the allowed suspend current budget (500  $\mu$ A/ 2.5 mA).

**FIGURE 1-2: BUS POWER ONLY**



## 1.6 GPIO Module

The GPIO Module is a standard 8-bit I/O port.

### 1.6.1 CONFIGURABLE PIN FUNCTIONS

The pins can be configured as:

- GPIO – Individually configurable general purpose input or output
- SSPND – USB Suspend state
- USBCFG – Indicates USB configuration status
- RxLED – Indicates USB receive traffic
- TxLED – Indicates USB transmit traffic

#### 1.6.1.1 SSPND Pin Function

The SSPND pin (if enabled) reflects the USB state (Suspend/Resume). The pin is active 'low' when the Suspend state has been issued by the USB host. Likewise, the pin drives 'high' after the Resume state is achieved.

This pin allows the application to go into Low Power mode when USB communication is suspended, and switches to a full active state when USB activity is resumed.

#### 1.6.1.2 USBCFG Pin Function

The USBCFG pin (if enabled) starts out 'low' during power-up or after Reset, and goes 'high' after the device successfully configures to the USB. The pin will go 'low' when in Suspend mode and 'high' when the USB resumes.

#### 1.6.1.3 RxLED Pin Function (IN Message)

The 'Rx' in the pin name is in respect to the USB host. The RxLED pin is an indicator for USB 'IN' messages.

This pin will either pulse low for a period of time (configurable for ~100 ms or ~200 ms), or toggle to the opposite state for every message received (IN message) by the USB host. This allows the application to count messages or provide a visual indication of USB traffic.

#### 1.6.1.4 TxLED Pin Function (OUT Message)

The 'Tx' in the pin name is in respect to the USB host. The TxLED pin is an indicator for USB 'OUT' messages.

This pin will either pulse low for a period of time (configurable for ~100 ms or ~200 ms), or toggle to the opposite state for every message transmitted (OUT message) by the USB host. This allows the application to count messages or provide a visual indication of USB traffic.

## 1.7 EEPROM Module

The EEPROM module is a 256 byte array of nonvolatile memory. The memory locations are accessed for read/write operations via USB host commands. Refer to **Section 2.0 “Configuration”** for details on accessing the EEPROM.

The host should wait for the write cycle to complete and then verify the write by reading the byte(s).

## 1.8 RESET/POR

### 1.8.1 RESET PIN

The  $\overline{\text{RST}}$  pin provides a method for triggering an external Reset of the device. A Reset is generated by holding the pin low. These devices have a noise filter in the reset path which detects and ignores small pulses.

### 1.8.2 POR

A Power-on Reset pulse is generated on-chip whenever VDD rises above a certain threshold. This allows the device to start in the initialized state when VDD is adequate for operation.

To take advantage of the POR circuitry, tie the  $\overline{\text{RST}}$  pin through a resistor (1 kOhm to 10 kOhm) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset delay.

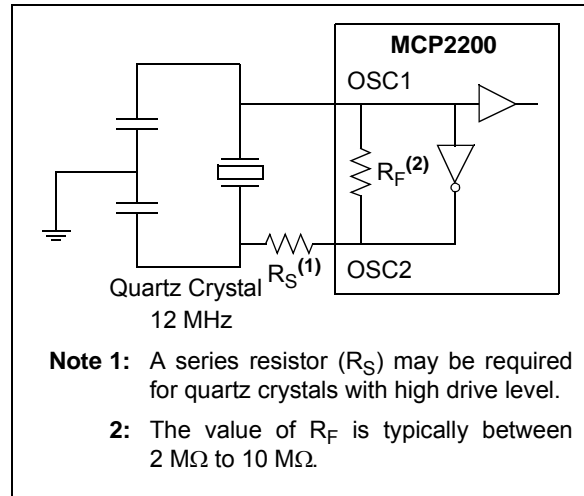
When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not achieved, the device must be held in Reset until the operating conditions are met.

## 1.9 Oscillator

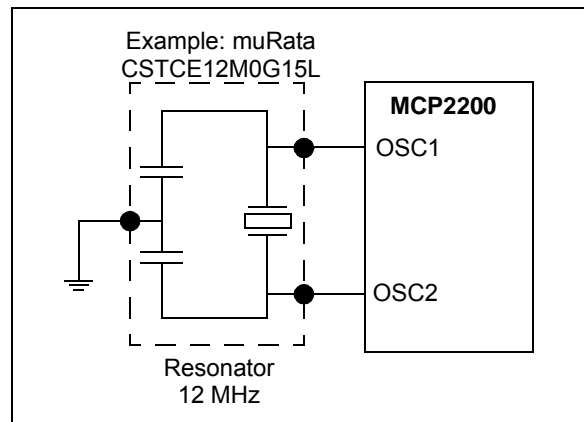
The input clock must be 12 MHz to provide the proper frequency for the USB module.

USB full-speed is nominally 12 Mb/s. The clock input accuracy is  $\pm 0.25\%$  (2,500 ppm maximum).

**FIGURE 1-3: QUARTZ CRYSTAL OPERATION**



**FIGURE 1-4: CERAMIC RESONATOR OPERATION**



# MCP2200

## 2.0 CONFIGURATION

The MCP2200 is configured by writing special commands using the HID interface. Configuration can be achieved using the configuration utility provided by Microchip. Alternatively, a custom utility can be developed by using the DLL available on the MCP2200 product page.

## 2.1 Configuration Utility

A configuration utility is provided by Microchip to allow the user to configure the MCP2200 to custom defaults. The Configuration Utility ([Figure 2-1](#)) connects to the device's HID interface where all of the configurable features can be set.

## 2.2 Serial String

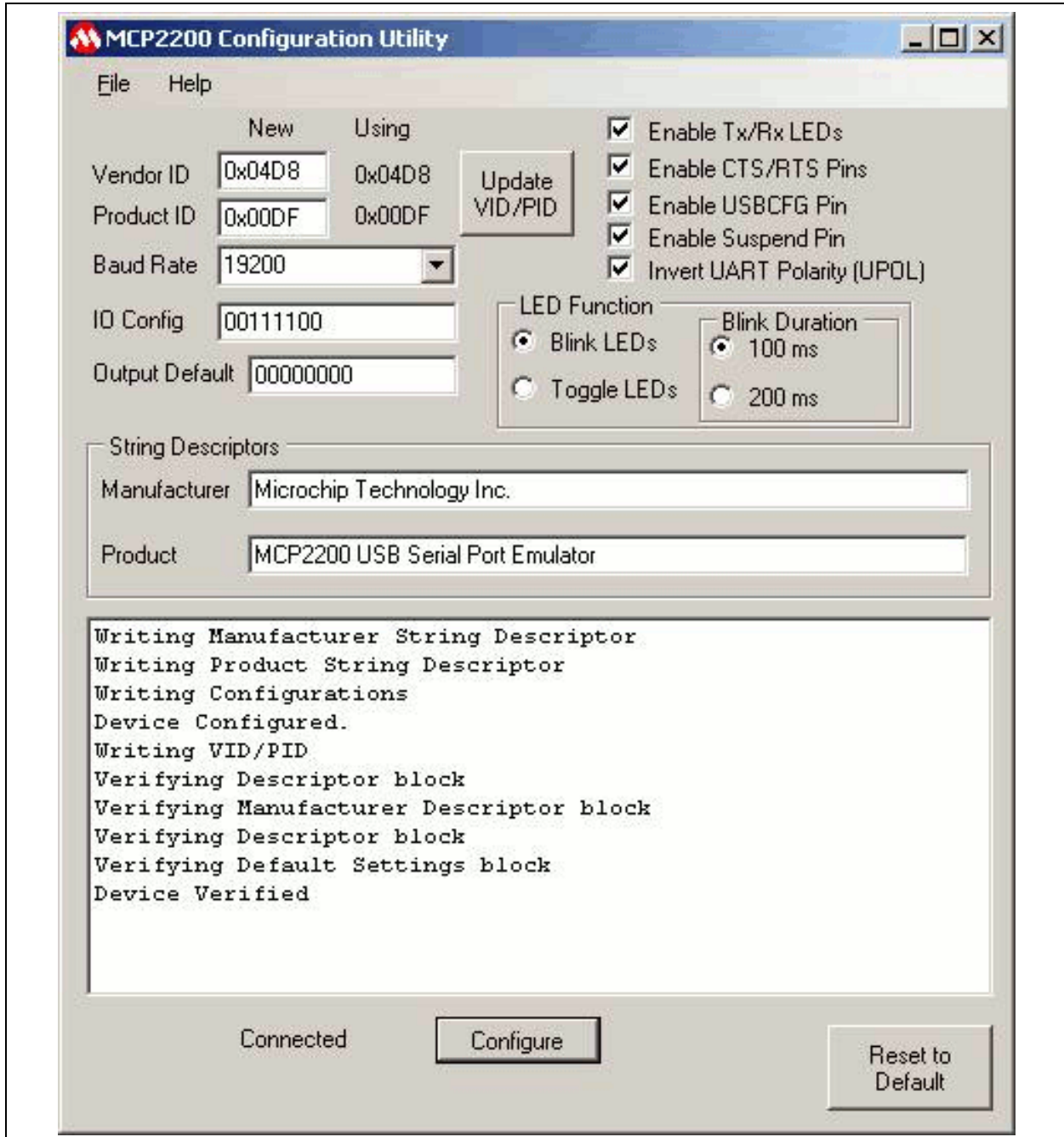
The MCP2200 is supplied from the factory with a serialized USB serial string.

**TABLE 2-1: CONFIGURATION DESCRIPTIONS**

Configuration Name	Description
Vendor ID (0x04D8)	The USB vendor identification assigned to Microchip by the USB consortium.
Product ID (0x00DF)	Device ID assigned by Microchip. The device can be used "as-is", or Microchip can assign a custom PID by request.
Baud Rate	Sets the UART baud rate using a list of primary baud rates. See the UART section for details on setting non-primary baud rates.
IO Config	Individually configures the I/O to inputs or outputs.
IO Default	Individually configures the output default state for pins configured as outputs.
Tx/Rx LEDs	Enables/disables the GP6 and GP7 pins to function as USB traffic indicators. Pins are active low when configured as traffic indicators.
Hardware Flow Control	Enables/disables $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ flow control.
USBCFG Pin	Enables/disables the GP1 pin as a USB configuration status indicator.
Suspend Pin	Enables/disables the GP0 pin as a USB suspend status pin.
Invert Sense	Enables/disables the UART lines states: <ul style="list-style-type: none"><li>- Normal – Tx/Rx idle high; <math>\overline{\text{CTS}}/\overline{\text{RTS}}</math> active low</li><li>- Inverted – Tx/Rx idle low; <math>\overline{\text{CTS}}/\overline{\text{RTS}}</math> active high</li></ul>
Manufacturer String	USB manufacturer string.
Product String	USB product string.



FIGURE 2-1: CONFIGURATION UTILITY



# MCP2200

## 2.3 Simple Configuration and I/O DLL

A DLL is provided by Microchip to help the user develop a custom configurator. See the documentation on the MCP2200 product page for details on associating the DLL with a Visual C++ project.

### 2.3.1 SIMPLE I/O DLL CALLS

The DLL provides the following functions to allow configuration of the device and control of the I/O.

#### 2.3.1.1 Initialization

```
void InitMCP2200(VID, PID)
bool IsConnected()
```

#### 2.3.1.2 Configuration (only needs to be set once; stored in NVM)

```
bool ConfigureIOPins(mask)
bool fnRxLED (On/Off, Toggle/Blink, 100/200mS)
bool fnTxLED (On/Off, Toggle/Blink, 100/200mS)
bool fnHardwareFlowControl (On/Off)
bool fnUSBcfg(on/off)
bool fnSuspend (on/off)
bool SetBaudRate (baudrate)
bool ConfigureAll (TRISmask, RxLED, TxLED, RxTGL, TxTGL, LEDX, FLOW, USBCFG, SSPND, BaudRate)
```

#### 2.3.1.3 I/O Control

```
bool ClearPin(pinnumber)
bool ReadPin(pinnumber)
bool SetPort (portValue)
bool ReadPort(*portValue)
bool WriteEEPROM (Address, Data)
bool ReadEEPROM (Address)
```

#### 2.3.1.4 Summary

```
Summary:
SimpleIOClass::InitMCP2200 (unsigned int VendorID, unsigned int ProductID)
bool SimpleIOClass::ConfigureMCP2200 (unsigned char IOMap, unsigned long BaudRate, unsigned int RxLED, unsigned int TxLED, bool Hardware Flow Control, bool USBCFG pin function, bool Suspend pin function)
bool SimpleIOClass::SetPin(unsigned int pin)
bool SimpleIOClass::ClearPin(unsigned int pin)
bool SimpleIOClass::ReadPin(unsigned int pin, unsigned int *returnvalue)
bool SimpleIOClass::WritePort(unsigned int value)
bool SimpleIOClass::ReadPort(unsigned int *returnvalue)
```

While ConfigureMCP2200 configures the device with one call, it may also be configured one parameter at a time:

```
bool SimpleIOClass::fnRxLED (unsigned int mode)
bool SimpleIOClass::fnTxLED (unsigned int mode)
bool SimpleIOClass::fnHardwareFlowControl (unsigned int onOff)
bool SimpleIOClass::fnUSBCFG (unsigned int onOff)
bool SimpleIOClass::fnSuspend (unsigned int onOff)
bool SimpleIOClass::fnSetBaudRate (unsigned long BaudRateParam)
bool SimpleIOClass::ConfigureIO (unsigned char IOMap)
```

Constants:

```
const unsigned int OFF = 0;
const unsigned int ON = 1;
const unsigned int TOGGLE = 3;
const unsigned int BLINKSLOW = 4;
const unsigned int BLINKFAST = 5;
```

#### 2.3.1.5 InitMCP2200

```
SimpleIOClass::InitMCP2200 (unsigned int VendorID, unsigned int ProductID)
```

Configures the Simple IO class for a specific Vendor and product ID.

Parameters:

Vendor ID - Assigned by USB IF ([www.usb.org](http://www.usb.org))  
Product ID - Assigned by the Vendor ID  
Holder

Returns:

none

Example:

```
InitMCP2200 (0x04D8, 0x00DF);
```

#### 2.3.1.6 ConfigureMCP2200

```
bool SimpleIOClass::ConfigureMCP2200 (unsigned char IOMap, unsigned long BaudRate, unsigned int RxLED, unsigned int TxLED, bool Hardware Flow Control, bool USBCFG pin function, bool Suspend pin function)
```

Configures the device's default baudrate, GPIO configuration and pin functions. Other functions set each parameter one at a time. This configures the part in one call.

Precondition:

The Vendor and Product ID must have been specified by SimpleIOInit.

## Parameters:

IOMap - A byte which represents the input/output state of the pins (each bit may be either a 1 for input, and 0 for output).  
Baudrate -  
RxLED & TxLED - constant OFF, TOGGLE, BLINKSLOW, BLINKFAST  
Boolean - HWFlow control, USBCFG, Suspend

## Returns:

Function returns true if the transmission is successful, returns False if there the transmission fails.

## Example:

<code>

```
    if (SimpleIOClass::ConfigureMCP2200(0x43,
9600, BLINKSLOW, BLINKFAST, false, false,
false) == SUCCESS)
        lblStatusBar->Text = "Success";
    else
        lblStatusBar->Text = "Invalid command
"
</code>
```

### 2.3.1.7 fnRxLED

bool SimpleIOClass::fnRxLED (unsigned int mode)

## Summary:

Configures the RxLED pin function.

## Description:

Sets the

## Precondition:

The Vendor and Product ID must have been specified by SimpleIOInit.

## Parameters:

mode (constant): OFF, TOGGLE, BLINKSLOW, BLINKFAST

## Returns:

Function returns true if the transmission is successful, returns False if there the transmission fails.

Error code logged in LastError.

## Example:

<code>

```
    if (SimpleIOClass::fnRxLED(BLINKFAST) ==
SUCCESS)
        lblStatusBar->Text = "Success";
    else
        lblStatusBar->Text = "Invalid command
" + SimpleIOClass::LastError;
</code>
```

### 2.3.1.8 fnTxLED

bool SimpleIOClass::fnTxLED (unsigned int mode)

## Summary:

Configures the TxLED pin function.

## Description:

Sets the

## Precondition:

The Vendor and Product ID must have been specified by SimpleIOInit.

## Parameters:

mode (constant): OFF, TOGGLE, BLINKSLOW, BLINKFAST

## Returns:

Function returns true if the transmission is successful, returns False if there the transmission fails.

Error code logged in LastError.

## Example:

<code>

```
    if (SimpleIOClass::fnTxLED(BLINKFAST) ==
SUCCESS)
        lblStatusBar->Text = "Success";
    else
        lblStatusBar->Text = "Invalid command
" + SimpleIOClass::LastError;
</code>
```

### 2.3.1.9 fnHardwareFlowControl

bool SimpleIOClass::fnHardwareFlowControl (unsigned int onOff)

## Summary:

Configures the device.

## Description:

Sets the

## Precondition:

The Vendor and Product ID must have been specified by SimpleIOInit.

## Parameters:

IOMap - An array of configuration parameters, one byte per IO Line

## Returns:

Function returns true if the transmission is successful, returns False if there the transmission fails.

# MCP2200

---

Example:

```
<code>
if
(SimpleIOClass::ConfigureMCP2200(OutputPacket
Buffer) == SUCCESS)
lblStatusBar->Text = "Success";
else
    lblStatusBar->Text = "Invalid command
" + SimpleIOClass::LastError;
</code>
```

## 2.3.1.10 fnUSBCFG

bool SimpleIOClass::fnUSBCFG (unsigned int onOff)

Summary:

Configures the USBCFG pin function.

Precondition:

The Vendor and Product ID must have been specified by SimpleIOInit.

Parameters:

onOff (constant): ON or OFF.

Returns:

Function returns true if the transmission is successful, returns False if there the transmission fails.

Example:

```
<code>
if
(SimpleIOClass::ConfigureMCP2200(OutputPacket
Buffer) == SUCCESS)
    lblStatusBar->Text = "Success";
else
    lblStatusBar->Text = "Invalid command
" + SimpleIOClass::LastError;
</code>
```

Remarks:

Error code is returned in LastError

## 2.3.1.11 fnSuspend

bool SimpleIOClass::fnSuspend (unsigned int onOff)

Summary:

Configures the SSPND function

Precondition:

The Vendor and Product ID must have been specified by SimpleIOInit.

Parameters:

onOff (constant): ON or OFF.

Returns:

Function returns true if the transmission is successful, returns False if there the transmission fails.

Example:

```
<code>
if
(SimpleIOClass::ConfigureMCP2200(OutputPacket
Buffer) == SUCCESS)
    lblStatusBar->Text = "Success";
else
    lblStatusBar->Text = "Invalid command
" + SimpleIOClass::LastError;
</code>
```

Remarks:

Error code is returned in LastError

## 2.3.1.12 fnSetBaudRate

bool SimpleIOClass::fnSetBaudRate (unsigned long BaudRateParam)

Summary:

Configures the device.

Precondition:

The Vendor and Product ID must have been specified by SimpleIOInit.

Parameters:

IOMap - An array of configuration parameters, one byte per IO Line

Returns:

Function returns true if the transmission is successful, returns False if there the transmission fails.

Example:

```
<code>
if
(SimpleIOClass::ConfigureMCP2200(OutputPacket
Buffer) == SUCCESS)
    lblStatusBar->Text = "Success";
else
    lblStatusBar->Text = "Invalid command
" + SimpleIOClass::LastError;
</code>
```

Remarks:

Error code is returned in LastError

## 2.3.1.13 ConfigureIO

bool SimpleIOClass::ConfigureIO (unsigned char IOMap)

Summary:

Configures the IO pins for Digital Input, Digital Output

Description:

IO Pins can be configured as Digital Input, Digital Output

**Precondition:**

The Vendor and Product ID must have been specified by SimpleIOInit.

**Parameters:**

IOMap - one byte, with each bit corresponding to each GP pin. 0 for output, 1 for input.

**Returns:**

Function returns true if the transmission is successful, returns False if there the transmission fails.

**Example:**

```
<code>
    if
    (SimpleIOClass::ConfigureGPIO(OutputPacketBuf
fer) == SUCCESS)
        lblStatusBar->Text = "Success";
    else
        lblStatusBar->Text = "Invalid command
" + SimpleIOClass::LastError;
</code>
```

### 2.3.1.14 SetPin

```
bool SimpleIOClass::SetPin(unsigned int pin)
```

**Summary:**

Sets the specified pin.

**Description:**

Sets the specified pin to one.

**Precondition:**

Must have previously been configured as an output via a ConfigureGPIO call.

**Parameters:**

pin - The pin number to set (0-7)

**Returns:**

Function returns true if the transmission is successful, returns False if there the transmission fails.

**Example:**

```
<code>
    if (SimpleIOClass::SetPin (2))
    {
        lblStatusBar->Text = "Success";
    }
    else
        lblStatusBar->Text = "Invalid command
" + SimpleIOClass::LastError;
</code>
```

**Remarks:**

none

### 2.3.1.15 ClearPin

```
bool SimpleIOClass::ClearPin(unsigned int
pin)
```

**Summary:**

Clears the specified pin.

**Description:**

Sets the specified pin to zero.

**Precondition:**

Must have previously been configured as an output via a ConfigureGPIO call.

**Parameters:**

pin - The pin number to set (0-7)

**Returns:**

Function returns true if the transmission is successful, returns False if there the transmission fails.

**Example:**

```
<code>
    if (SimpleIOClass::ClearPin (2))
    {
        lblStatusBar->Text = "Success";
    }
    else
        lblStatusBar->Text = "Invalid command
" + SimpleIOClass::LastError;
</code>
```

**Remarks:**

none

### 2.3.1.16 ReadPin

```
bool SimpleIOClass::ReadPin(unsigned int pin,
unsigned int *returnvalue)
```

**Summary:**

Reads the specified pin.

**Description:**

Reads the specified pin and returns the value in returnvalue. If the pin has been configured as Digital Input, the return value will be either 0 or 1. If the pin has been configured as Analog Input, the pin will be read by the ADC and return a 10 bit value, right justified.

**Precondition:**

Must have previously been configured as an input via a ConfigureGPIO call.

**Parameters:**

pin - The pin number to set (0-7)

returnvalue - the value read on the pin (0 or 1)

# MCP2200

---

## Returns:

true if the pin was successfully read.  
false if the pin was not read (not configured as an input).

## Example:

```
<code>
    unsigned int rv;
    if (SimpleIOClass::ReadGPIOOn (0, &rv))
    {
        lblStatusBar->Text = "Success";
    }
    else
        lblStatusBar->Text = "Invalid command
" + SimpleIOClass::LastError;
</code>
```

### 2.3.1.17 WritePort

bool SimpleIOClass::WritePort(unsigned int portValue)

## Summary:

Writes a value to the GPIO port.

## Description:

Writes the GPIO port. This provides a means to write all pins at once instead of one-at-a-time.

## Precondition:

Must have previously been configured as an output via a ConfigureGPIO call.

## Parameters:

portValue - Byte value to set on the port.

## Returns:

Function returns true if the transmission is successful  
returns False if there the transmission fails.

## Example:

```
<code>
    if (SimpleIOClass::WritePort (0x5A))
    {
        lblStatusBar->Text = "Success";
    }
    else
        lblStatusBar->Text = "Invalid command
" + SimpleIOClass::LastError;
</code>
```

## Remarks:

Pins configured for output returns the current state of the port.

Pins configured as input read as zero.

### 2.3.1.18 ReadPort

bool SimpleIOClass::ReadPort(unsigned int \*returnvalue)

## Summary:

Reads the GPIO port as digital input.

## Description:

Reads the GPIO port and returns the value in returnvalue. This provides a means to read all pins at once instead of one-at-a-time.

## Precondition:

Must have previously been configured as an input via a ConfigureGPIO call.

## Parameters:

returnvalue - the value read on the port.

## Returns:

Function returns true if the read is successful  
returns False if there the transmission fails.

## Example:

```
<code>
    unsigned int rv;
    if (SimpleIOClass::ReadGPIOPort (&rv))
    {
        lblStatusBar->Text = "Success";
    }
    else
        lblStatusBar->Text = "Invalid command
" + SimpleIOClass::LastError;
</code>
```

## Remarks:

Pins configured for output returns the current state of the port.

Pins configured as input read as zero.

## 3.0 ELECTRICAL CHARACTERISTICS

### Absolute Maximum Ratings <sup>(†)</sup>

Ambient temperature under bias .....	-40°C to +85°C
Storage temperature .....	-65°C to +150°C
Voltage on VDD with respect to Vss, PIC18F .....	-0.3V to +6.0V
Voltage on VDD with respect to Vss, PIC18LF .....	-0.3V to +4.0V
Voltage on $\overline{\text{MCLR}}$ with respect to Vss .....	-0.3V to +9.0V
Voltage on VUSB pin with respect to Vss .....	-0.3V to +4.0V
Voltage on D+ and D- pins with respect to Vss .....	-0.3V to (VUSB + 0.3V)
Voltage on all other pins with respect to Vss .....	-0.3V to (VDD + 0.3V)
Total power dissipation <sup>(1)</sup> .....	800 mW
Maximum current out of Vss pin .....	95 mA
Maximum current into VDD pin .....	95 mA
Clamp current, I <sub>K</sub> (V <sub>PIN</sub> < 0 or V <sub>PIN</sub> > VDD).....	± 20 mA
Maximum output current sunk by any I/O pin.....	25 mA
Maximum output current sourced by any I/O pin .....	25 mA
Maximum current sunk by all ports .....	90 mA
Maximum current sourced by all ports .....	90 mA

**Note 1:** Power dissipation is calculated as follows:  $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$ .

**2:** V<sub>USB</sub> must always be  $\leq V_{DD} + 0.3V$

† **NOTICE:** Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure above maximum rating conditions for extended periods may affect device reliability.

# MCP2200

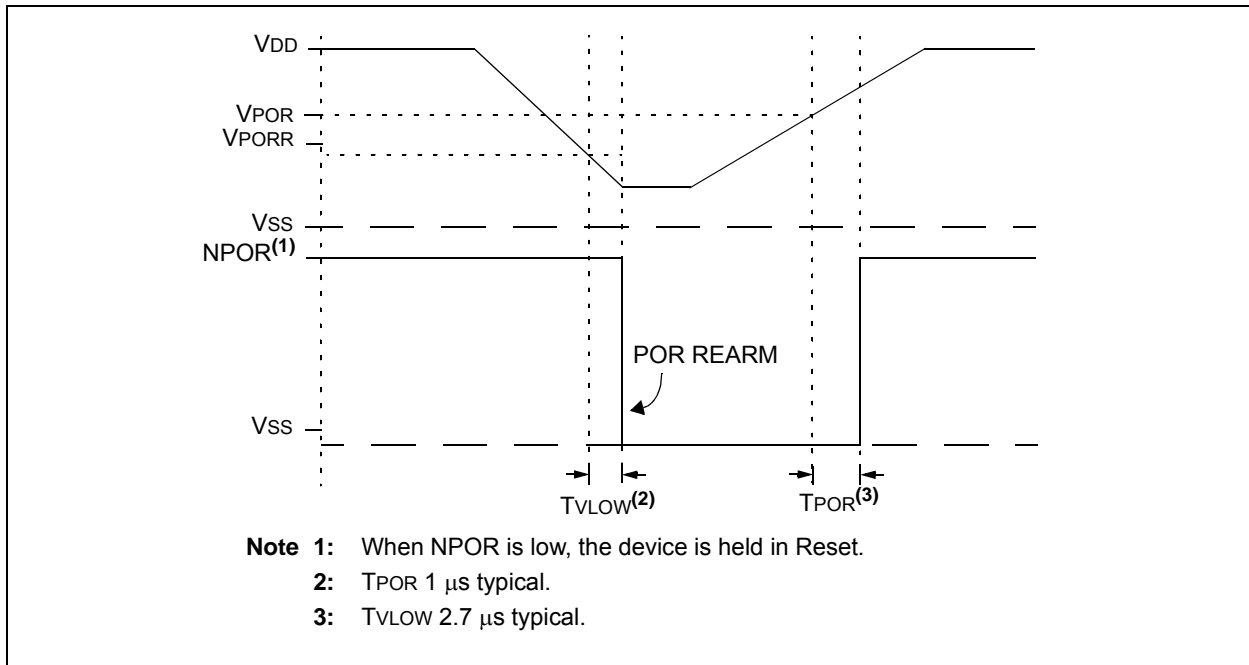
## 3.1 DC CHARACTERISTICS

DC Characteristics		Operating Conditions (unless otherwise indicated): 3.0V ≤ VDD ≤ 5.5V at -40°C ≤ TA ≤ +85°C (I-Temp)					
Param No.	Characteristic	Sym	Min	Typ	Max	Units	Conditions
D001	Supply Voltage	VDD	3.0	—	5.5	V	
	Power-on Reset Release Voltage	VPOR		1.6		V	
	Power-on Reset Rearm Voltage			0.8		V	
D003	VDD Rise Rate to Ensure Power-on Reset	SVDD	0.05	—	—	V/ms	Design guidance only Not tested
D004	Supply Current	IDD					
	VDD = 3.0V		—	10	12	mA	FOSC = 12 MHz, (330 nF on VUSB)
VDD = 5.0V		—	13	15	mA		
D005	Standby current	IDDS	—	9	—	μA	
<b>Input Low-Voltage</b>							
D031	Schmitt Trigger (GPIO)	VIL	—	—	0.2 VDD	V	3.0V ≤ VDD ≤ 5.5V
	TTL (CTS pin)		—	—	0.8		4.5V ≤ VDD ≤ 5.5V
<b>Input High-Voltage</b>							
D041	Schmitt Trigger (GPIO)	VIH	0.8 VDD	—	VDD	V	3.0V ≤ VDD ≤ 5.5V
	TTL (RTS pin)		2.0	—	VDD		4.5V ≤ VDD ≤ 5.5V
<b>Input Leakage Current</b>							
D060	GPIO, CTS	IIL	—	±50	±100	nA	VSS ≤ VPIN ≤ VDD, pin at Hi-Z
	RST			±50	±200		
	OSC1			±50	±100		
<b>Output Low-Voltage</b>							
D080	GPIO	VOL	—	—	0.6	V	IO_L = 8.0 mA, VDD = 5.0V
			—	—	0.6		IO_L = 6.0 mA, VDD = 3.3V
<b>Output High-Voltage</b>							
D090	GPIO	VOH	VDD - 0.7	—	—	V	IO_H = -3.5 mA, VDD = 5.0V
			VDD - 0.7	—	—		IO_H = -3.0 mA, VDD = 3.3V
<b>Capacitive Loading Specs on Output Pins</b>							
D101	OSC2	COSC2	—	—	15	pF	<b>Note 1</b>
D102	GPIO	CIO	—	—	50	pF	<b>Note 1</b>

**Note 1:** This parameter is characterized, but not tested.



**FIGURE 3-1: POR AND POR REARM WITH SLOW RISING V<sub>DD</sub>**



**TABLE 3-1: USB MODULE SPECIFICATIONS**

DC Characteristics		Operating Conditions (unless otherwise indicated): 3.0V $\leq$ V <sub>DD</sub> $\leq$ 5.5V at -40°C $\leq$ T <sub>A</sub> $\leq$ +85°C (I-Temp)					
Param No.	Characteristic	Sym	Min	Typ	Max	Units	Conditions
D313	USB Voltage	V <sub>USB</sub>	3.0	—	3.6	V	Voltage on V <sub>USB</sub> pin must be in this range for proper USB operation
D314	Input Leakage on Pin	I <sub>IL</sub>	—	—	$\pm 1$	$\mu$ A	V <sub>SS</sub> $\leq$ V <sub>PIN</sub> $\leq$ V <sub>DD</sub> pin at high-impedance
D315	Input Low Voltage for USB Buffer	V <sub>ILUSB</sub>	—	—	0.8	V	For V <sub>USB</sub> range
D316	Input High Voltage for USB Buffer	V <sub>IHUSB</sub>	2.0	—	—	V	For V <sub>USB</sub> range
D318	Differential Input Sensitivity	V <sub>DIFS</sub>	—	—	0.2	V	The difference between D+ and D- must exceed this value while V <sub>CM</sub> is met
D319	Differential Common Mode Range	V <sub>CM</sub>	0.8	—	2.5	V	
D320	Driver Output Impedance <sup>(1)</sup>	Z <sub>OUT</sub>	28	—	44	$\Omega$	
D321	Voltage Output Low	V <sub>OL</sub>	0.0	—	0.3	V	1.5 k $\Omega$ load connected to 3.6V
D322	Voltage Output High	V <sub>OH</sub>	2.8	—	3.6	V	1.5 k $\Omega$ load connected to ground

**Note 1:** The D+ and D- signal lines have been built-in impedance matching resistors. No external resistors, capacitors or magnetic components are necessary on the D+/D- signal paths between the MCP2200 family device and the USB cable.

# MCP2200

**TABLE 3-2: THERMAL CONSIDERATIONS**

Standard Operating Conditions (unless otherwise stated)					
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (I-Temp)					
Param No.	Sym	Characteristic	Typ	Units	Conditions
TH01	$\theta_{JA}$	Thermal Resistance Junction to Ambient	85.2	$^{\circ}\text{C}/\text{W}$	20-pin SOIC package
			108.1	$^{\circ}\text{C}/\text{W}$	20-pin SSOP package
			TBD	$^{\circ}\text{C}/\text{W}$	20-pin QFN 5x5mm package
TH02	$\theta_{JC}$	Thermal Resistance Junction to Case	24	$^{\circ}\text{C}/\text{W}$	20-pin SOIC package
			24	$^{\circ}\text{C}/\text{W}$	20-pin SSOP package
			24	$^{\circ}\text{C}/\text{W}$	20-pin QFN 6x6mm package
TH03	$T_{JMAX}$	Maximum Junction Temperature	150	$^{\circ}\text{C}$	
TH04	PD	Power Dissipation	—	W	$PD = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	Internal Power Dissipation	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}^{(1)}$
TH06	$P_{I/O}$	I/O Power Dissipation	—	W	$P_{I/O} = \sum (I_{OL} * V_{OL}) + \sum (I_{OH} * (V_{DD} - V_{OH}))$
TH07	$P_{DER}$	Derated Power	—	W	$P_{DER} = P_{D_{MAX}} (T_J - T_A) / \theta_{JA}^{(2,3)}$

**Legend:** TBD = To Be Determined.

**Note 1:**  $I_{DD}$  is the current to run the chip alone without driving any load on the output pins.

**2:**  $T_A$  = Ambient Temperature.

**3:**  $T_J$  = Junction Temperature.

## 3.2 AC Characteristics

### 3.2.1 TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created in one of the following formats:

1. TppS2ppS

2. TppS

<p>T</p> <p>F      Frequency</p> <p>E      Error</p>	<p>T      Time</p>
--	--------------------

Lowercase letters (pp) and their meanings:

<p>pp</p> <p>io      Input or Output pin</p> <p>rx      Receive</p> <p>bitclk   RX/TX BITCLK</p> <p>drt      Device Reset Timer</p>	<p>osc      Oscillator</p> <p>tx      Transmit</p> <p>RST      Reset</p>
---	--

Uppercase letters and their meanings:

<p>S</p> <p>F      Fall</p> <p>H      High</p> <p>I      Invalid (high-impedance)</p> <p>L      Low</p>	<p>P      Period</p> <p>R      Rise</p> <p>V      Valid</p> <p>Z      High-impedance</p>
---	--

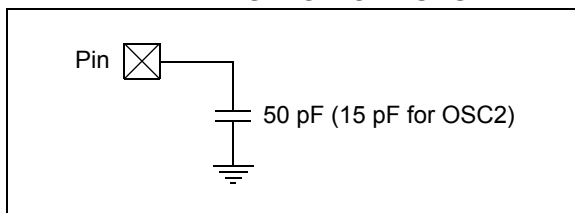
### 3.2.2 TIMING CONDITIONS

The operating temperature and voltage specified in [Table 3-3](#) apply to all timing specifications unless otherwise noted. [Figure 3-2](#) specifies the load conditions for the timing specifications.

**TABLE 3-3: TEMPERATURE AND VOLTAGE SPECIFICATIONS - AC**

<b>AC CHARACTERISTICS</b>	<p><b>Standard Operating Conditions (unless otherwise stated)</b></p> <p>Operating temperature <math>-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}</math></p> <p>Operating voltage VDD range as described in DC spec <a href="#">Section 3.1</a> "DC Characteristics".</p>
---------------------------	--

**FIGURE 3-2: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS**



# MCP2200

---

## 3.2.3 TIMING DIAGRAMS AND SPECIFICATIONS

**TABLE 3-4: RESET, OSCILLATOR START-UP TIMER AND POWER-UP TIMER PARAMETERS**

Standard Operating Conditions (unless otherwise stated)							
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$							
Param No.	Sym	Characteristic	Min	Typ <sup>†</sup>	Max	Units	Conditions
30	TRST	MCLR Pulse Width (low)	2	—	—	μs	
31	TPWRT	Power-up timer	40	65	140	ms	
32	TOST	Oscillator startup time	—	1024	—	TOST	

\* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

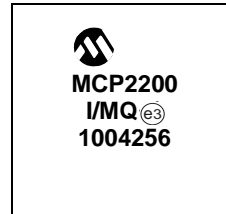
## 4.0 PACKAGING INFORMATION

### 4.1 Package Marking Information

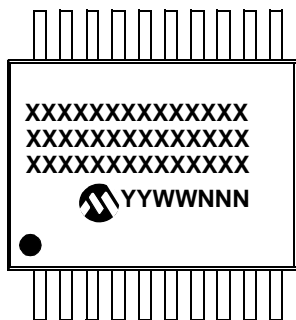
20-Lead QFN



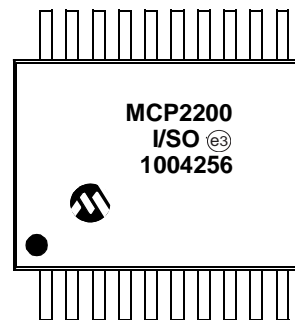
Example



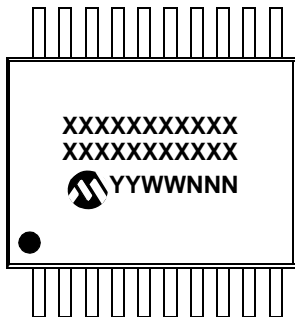
20-Lead SOIC



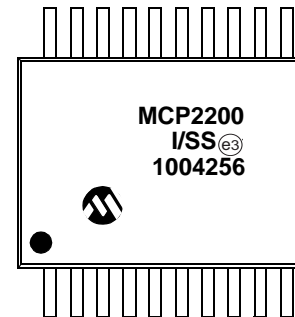
Example:



20-Lead SSOP



Example:



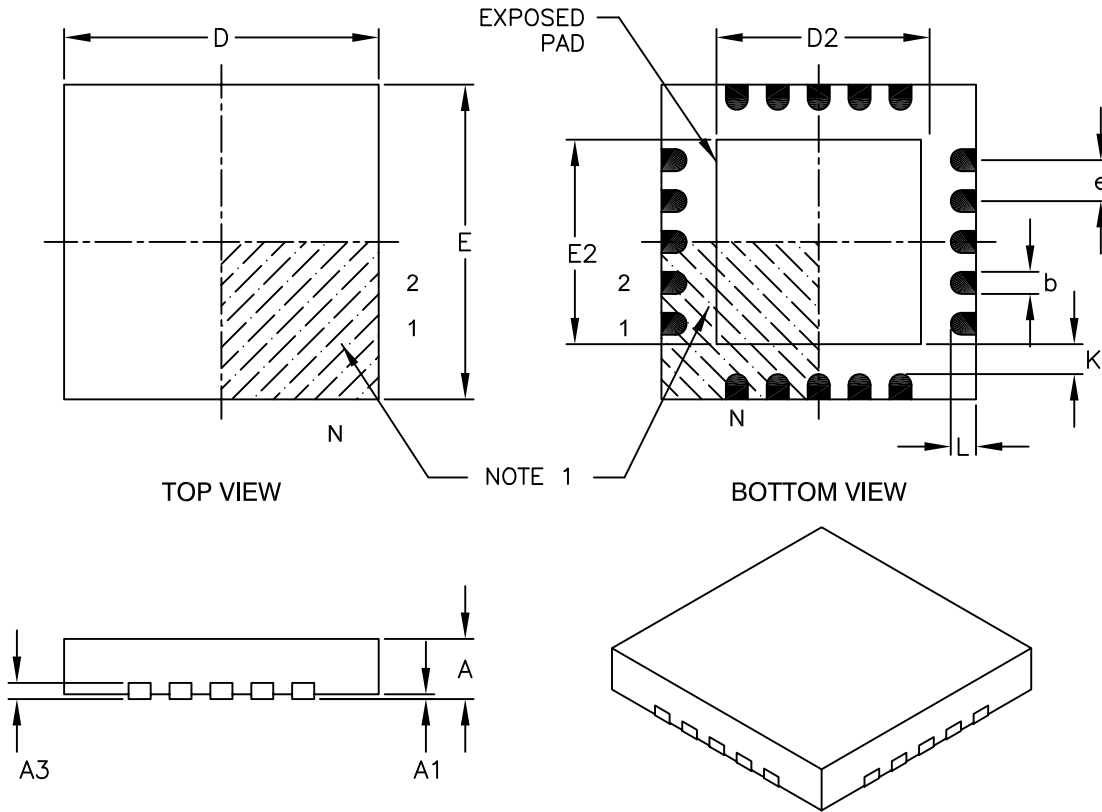
<b>Legend:</b>	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package.

**Note:** In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

# MCP2200

## 20-Lead Plastic Quad Flat, No Lead Package (MQ) – 5x5x0.9 mm Body [QFN]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Width	E	5.00 BSC		
Exposed Pad Width	E2	3.15	3.25	3.35
Overall Length	D	5.00 BSC		
Exposed Pad Length	D2	3.15	3.25	3.35
Contact Width	b	0.25	0.30	0.35
Contact Length	L	0.35	0.40	0.45
Contact-to-Exposed Pad	K	0.20	-	-

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

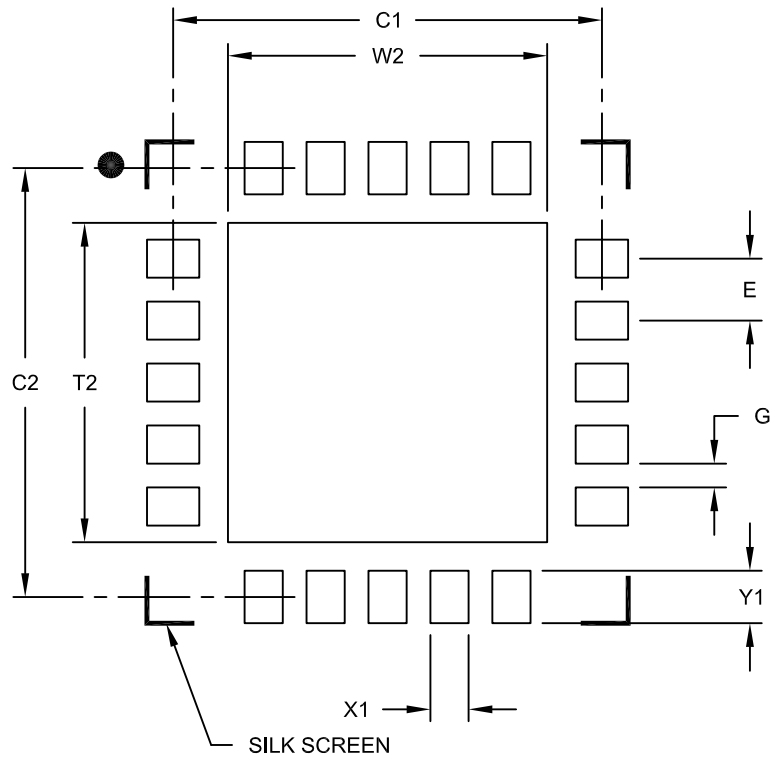
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-139B

## 20-Lead Plastic Quad Flat, No Lead Package (MQ) - 5x5 mm Body [QFN] With 0.40mm Contact Length

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			3.35
Optional Center Pad Length	T2			3.35
Contact Pad Spacing	C1		4.50	
Contact Pad Spacing	C2		4.50	
Contact Pad Width (X20)	X1			0.40
Contact Pad Length (X20)	Y1			0.55
Distance Between Pads	G	0.20		

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

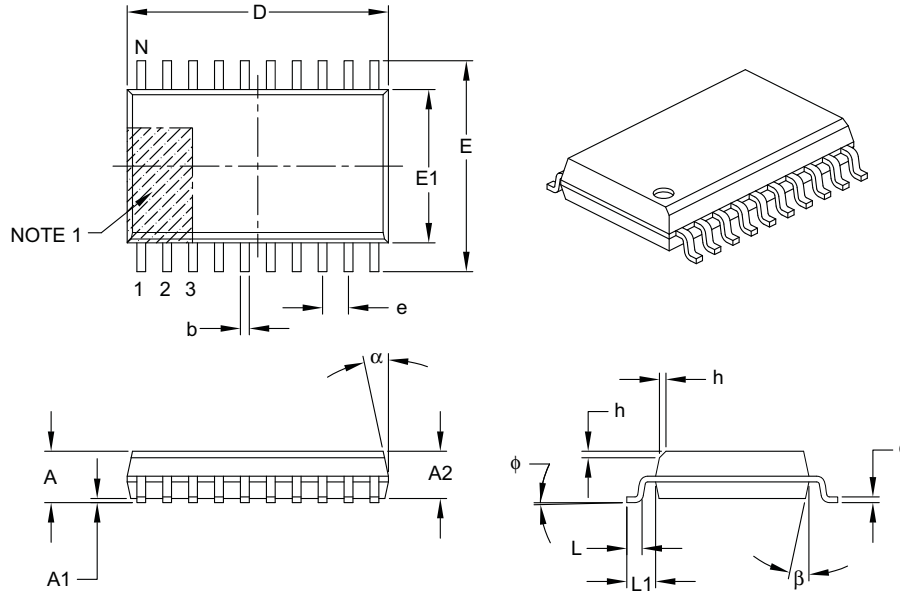
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2139A

# MCP2200

## 20-Lead Plastic Small Outline (SO) – Wide, 7.50 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	1.27 BSC		
Overall Height	A	–	–	2.65
Molded Package Thickness	A2	2.05	–	–
Standoff §	A1	0.10	–	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	12.80 BSC		
Chamfer (optional)	h	0.25	–	0.75
Foot Length	L	0.40	–	1.27
Footprint	L1	1.40 REF		
Foot Angle	φ	0°	–	8°
Lead Thickness	c	0.20	–	0.33
Lead Width	b	0.31	–	0.51
Mold Draft Angle Top	α	5°	–	15°
Mold Draft Angle Bottom	β	5°	–	15°

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

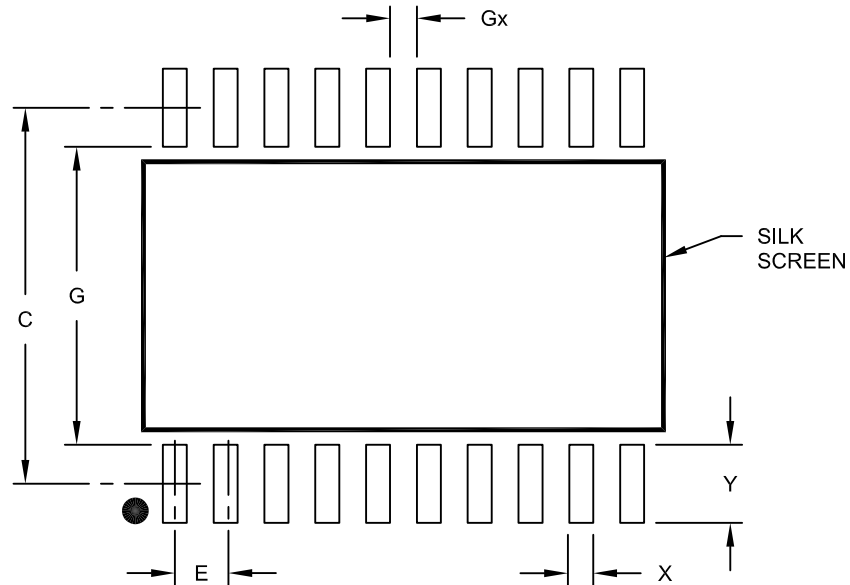
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-094B



## 20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		9.40	
Contact Pad Width (X20)	X			0.60
Contact Pad Length (X20)	Y			1.95
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.45		

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

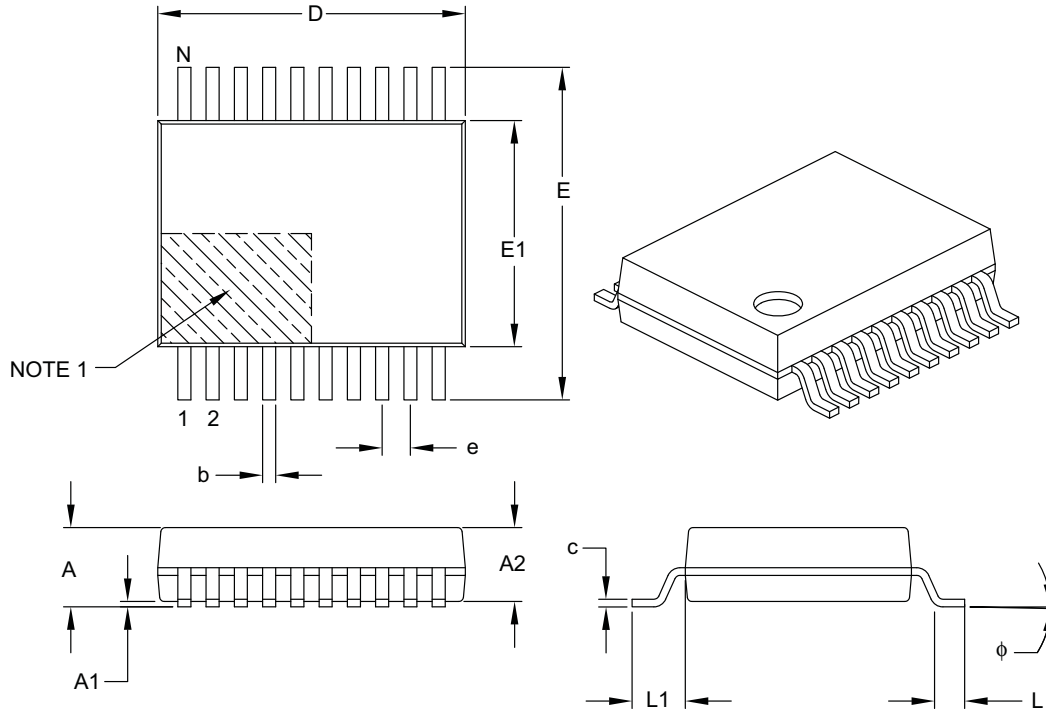
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A

# MCP2200

## 20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	0.65 BSC		
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	c	0.09	–	0.25
Foot Angle	$\phi$	0°	4°	8°
Lead Width	b	0.22	–	0.38

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

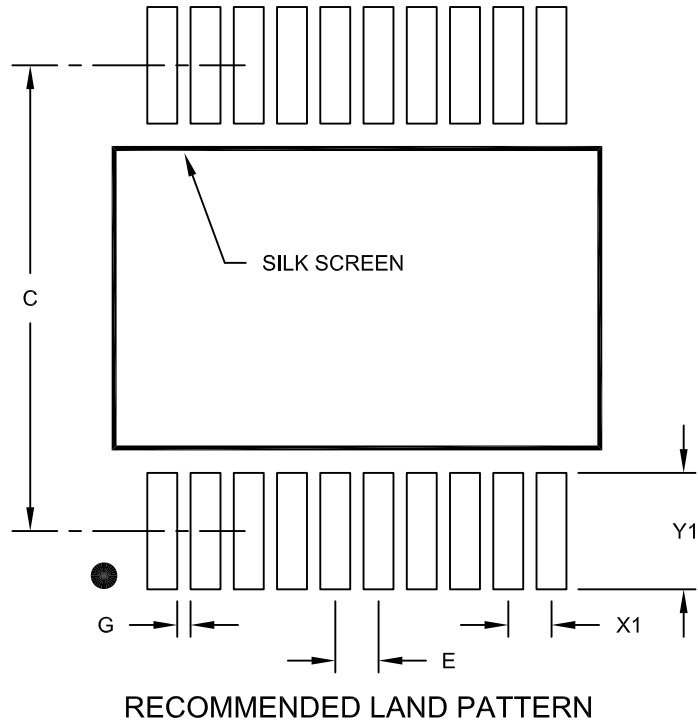
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

## 20-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



		Units	MILLIMETERS		
Dimension Limits			MIN	NOM	MAX
Contact Pitch	E		0.65 BSC		
Contact Pad Spacing	C			7.20	
Contact Pad Width (X20)	X1				0.45
Contact Pad Length (X20)	Y1				1.75
Distance Between Pads	G	0.20			

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072A

# MCP2200

---

NOTES:

## APPENDIX A: REVISION HISTORY

### Revision A (March 2010)

- Original Release of this Document.

# MCP2200

---

NOTES:

## PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

<u>PART NO.</u>	<u>X</u>	<u>/XX</u>
Device	Temperature Range	Package
Device	MCP2200: USB-to-UART serial converter MCP2200T: USB-to-UART serial converter (Tape and Reel)	
Temperature Range	I = -40°C to +85°C (Industrial)	
Package	MQ = Plastic Quad Flat, No Lead Package 5x5x0.9 mm Body (QFN), 20-Lead SO = Plastic Small Outline - Wide, 7.50 mm Body (SO), 20-Lead SS = Plastic Shrink Small Outline - 5.30 mm Body (SS) 20-Lead	

**Examples:**

- a) MCP2200- I/MQ: Industrial temperature, 20LD QFN Package.
- b) MCP2200T- I/MQ: Tape and Reel, Industrial temperature, 20LD QFN Package.
- a) MCP2200- I/SO: Industrial temperature, 20LD SOIC Package.
- b) MCP2200T- I/SO: Tape and Reel, Industrial temperature, 20LD SOIC Package.
- a) MCP2200- I/SS: Industrial temperature, 20LD SSOP Package.
- b) MCP2200T- I/SS: Tape and Reel, Industrial temperature, 20LD SSOP Package.

# MCP2200

---

NOTES:



---

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

**Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC<sup>32</sup> logo, rPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Octopus, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICTail, REAL ICE, rLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2010, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

ISBN: 978-1-60932-051-5

*Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC<sup>®</sup> MCUs and dsPIC<sup>®</sup> DSCs, KEELOQ<sup>®</sup> code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*

**QUALITY MANAGEMENT SYSTEM  
CERTIFIED BY DNV  
== ISO/TS 16949:2002 ==**



---

---

## WORLDWIDE SALES AND SERVICE

---

---

### AMERICAS

#### Corporate Office

2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7200  
Fax: 480-792-7277  
Technical Support:  
<http://support.microchip.com>  
Web Address:  
[www.microchip.com](http://www.microchip.com)

#### Atlanta

Duluth, GA  
Tel: 678-957-9614  
Fax: 678-957-1455

#### Boston

Westborough, MA  
Tel: 774-760-0087  
Fax: 774-760-0088

#### Chicago

Itasca, IL  
Tel: 630-285-0071  
Fax: 630-285-0075

#### Cleveland

Independence, OH  
Tel: 216-447-0464  
Fax: 216-447-0643

#### Dallas

Addison, TX  
Tel: 972-818-7423  
Fax: 972-818-2924

#### Detroit

Farmington Hills, MI  
Tel: 248-538-2250  
Fax: 248-538-2260

#### Kokomo

Kokomo, IN  
Tel: 765-864-8360  
Fax: 765-864-8387

#### Los Angeles

Mission Viejo, CA  
Tel: 949-462-9523  
Fax: 949-462-9608

#### Santa Clara

Santa Clara, CA  
Tel: 408-961-6444  
Fax: 408-961-6445

#### Toronto

Mississauga, Ontario,  
Canada  
Tel: 905-673-0699  
Fax: 905-673-6509

### ASIA/PACIFIC

#### Asia Pacific Office

Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2401-1200  
Fax: 852-2401-3431

#### Australia - Sydney

Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

#### China - Beijing

Tel: 86-10-8528-2100  
Fax: 86-10-8528-2104

#### China - Chengdu

Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

#### China - Chongqing

Tel: 86-23-8980-9588  
Fax: 86-23-8980-9500

#### China - Hong Kong SAR

Tel: 852-2401-1200  
Fax: 852-2401-3431

#### China - Nanjing

Tel: 86-25-8473-2460  
Fax: 86-25-8473-2470

#### China - Qingdao

Tel: 86-532-8502-7355  
Fax: 86-532-8502-7205

#### China - Shanghai

Tel: 86-21-5407-5533  
Fax: 86-21-5407-5066

#### China - Shenyang

Tel: 86-24-2334-2829  
Fax: 86-24-2334-2393

#### China - Shenzhen

Tel: 86-755-8203-2660  
Fax: 86-755-8203-1760

#### China - Wuhan

Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

#### China - Xian

Tel: 86-29-8833-7252  
Fax: 86-29-8833-7256

#### China - Xiamen

Tel: 86-592-2388138  
Fax: 86-592-2388130

#### China - Zhuhai

Tel: 86-756-3210040  
Fax: 86-756-3210049

### ASIA/PACIFIC

#### India - Bangalore

Tel: 91-80-3090-4444  
Fax: 91-80-3090-4123

#### India - New Delhi

Tel: 91-11-4160-8631  
Fax: 91-11-4160-8632

#### India - Pune

Tel: 91-20-2566-1512  
Fax: 91-20-2566-1513

#### Japan - Yokohama

Tel: 81-45-471-6166  
Fax: 81-45-471-6122

#### Korea - Daegu

Tel: 82-53-744-4301  
Fax: 82-53-744-4302

#### Korea - Seoul

Tel: 82-2-554-7200  
Fax: 82-2-558-5932 or  
82-2-558-5934

#### Malaysia - Kuala Lumpur

Tel: 60-3-6201-9857  
Fax: 60-3-6201-9859

#### Malaysia - Penang

Tel: 60-4-227-8870  
Fax: 60-4-227-4068

#### Philippines - Manila

Tel: 63-2-634-9065  
Fax: 63-2-634-9069

#### Singapore

Tel: 65-6334-8870  
Fax: 65-6334-8850

#### Taiwan - Hsin Chu

Tel: 886-3-6578-300  
Fax: 886-3-6578-370

#### Taiwan - Kaohsiung

Tel: 886-7-536-4818  
Fax: 886-7-536-4803

#### Taiwan - Taipei

Tel: 886-2-2500-6610  
Fax: 886-2-2508-0102

#### Thailand - Bangkok

Tel: 66-2-694-1351  
Fax: 66-2-694-1350

### EUROPE

#### Austria - Wels

Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

#### Denmark - Copenhagen

Tel: 45-4450-2828  
Fax: 45-4485-2829

#### France - Paris

Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

#### Germany - Munich

Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

#### Italy - Milan

Tel: 39-0331-742611  
Fax: 39-0331-466781

#### Netherlands - Drunen

Tel: 31-416-690399  
Fax: 31-416-690340

#### Spain - Madrid

Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

#### UK - Wokingham

Tel: 44-118-921-5869  
Fax: 44-118-921-5820