Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

MCP2221A

USB 2.0 to I²C/UART Protocol Converter with GPIO

Note: The MCP2221A is identical to the MCP2221 in all aspects except for the maximum supported baud rate of the UART, which has been increased from 115200 (MCP2221) to 460800 (MCP2221A).

Features

Universal Serial Bus (USB)

- Supports Full-Speed USB (12 Mb/s)
- Implements USB Protocol Composite Device:
  - Communication Device Class (CDC) for USB-to-UART conversion
  - Human Interface Device (HID) for I²C device control and configuration
- 448-Byte Buffer to Handle Data Throughput at Any Supported UART Baud Rate:
  - 64-byte transmit
  - 384-byte receive
- Human Interface Device (HID) for Both I²C Communication and Control:
  - 64-byte buffer to handle data throughput at any I²C baud rate
- Fully-Configurable VID and PID Assignments and String Descriptors
- Bus Powered or Self Powered
- USB 2.0-Compliant: TID# 40001594

USB Driver and Software Support

- Enumerates as a Composite USB Device (CDC and HID) Using Standard Drivers for Virtual Com Port (VCP) on the Following Windows® Operating Systems: XP® (SP3), Vista®, 7, 8, 8.1 and 10
- Configuration Utility for Establishing a Custom Boot-Up Configuration
- I²C/SMBus Terminal
- Windows DLL

CDC and Universal Asynchronous Receiver/Transmitter (UART) Options

- Communications Device Class (CDC) for the USB-to-UART Option
- Responds to `SET LINE CODING` Commands to Dynamically Change Baud Rates
- Supports Baud Rates: 300-460800
- UART Tx and Rx Pins Only
- Serial Number Used During the CDC Enumeration Can Be Enabled By Using the Microchip-provided Configuration Utility or By Calling the Proper API From the Support Libraries for this Device

I²C/SMBus

- The Device Runs as an I²C Master. The Data to Write/Read On the I²C Bus is Conveyed By the USB Interface.
- I²C Master:
  - Up to 400 kHz clock rate
  - Supports 7-bit or 10-bit addressable devices; 10-bit addressable devices are supported through the PC host library
  - Supports block reads/writes of up to 65,535 bytes long
- SMBus Master:
  - Supports all of the SMBus transfers
  - SMBus functionality is achieved through a combination of chip and support library processing
  - Up to 400 kHz clock rate

General Purpose Input/Output (GPIO) Pins

- Four General Purpose Input/Output Pins
- All GP Pins Can Be Assigned to Other Functionalities

Other Functionalities

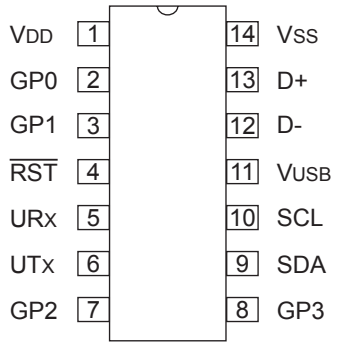
- UART Activity LED Outputs (UTx and URx)
- SSPND Output Pin
- USBCFG Output Pin (Indicates When the Enumeration Has Completed)
- Three ADC Inputs
- One DAC with Two Possible Output Options
- Clock Reference Output: 12 MHz or Other Configurable Values
- External Interrupt Edge Detection

Other

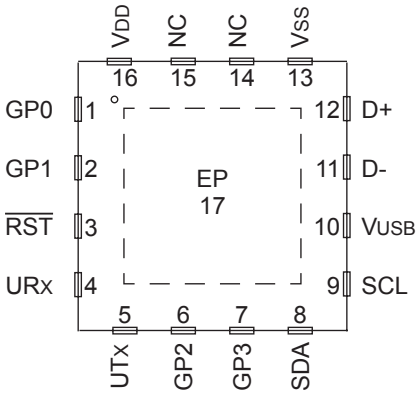
- Operating Voltage: 3.0 to 5.5V
- Electrostatic Discharge (ESD) Protection: > 4 kV Human Body Model (HBM)
- Industrial (I) Operating Temperature: -40°C to +85°C
- Automotive AEC-Q100 Qualified

Package Types

MCP2221A
PDIP/SOIC/TSSOP

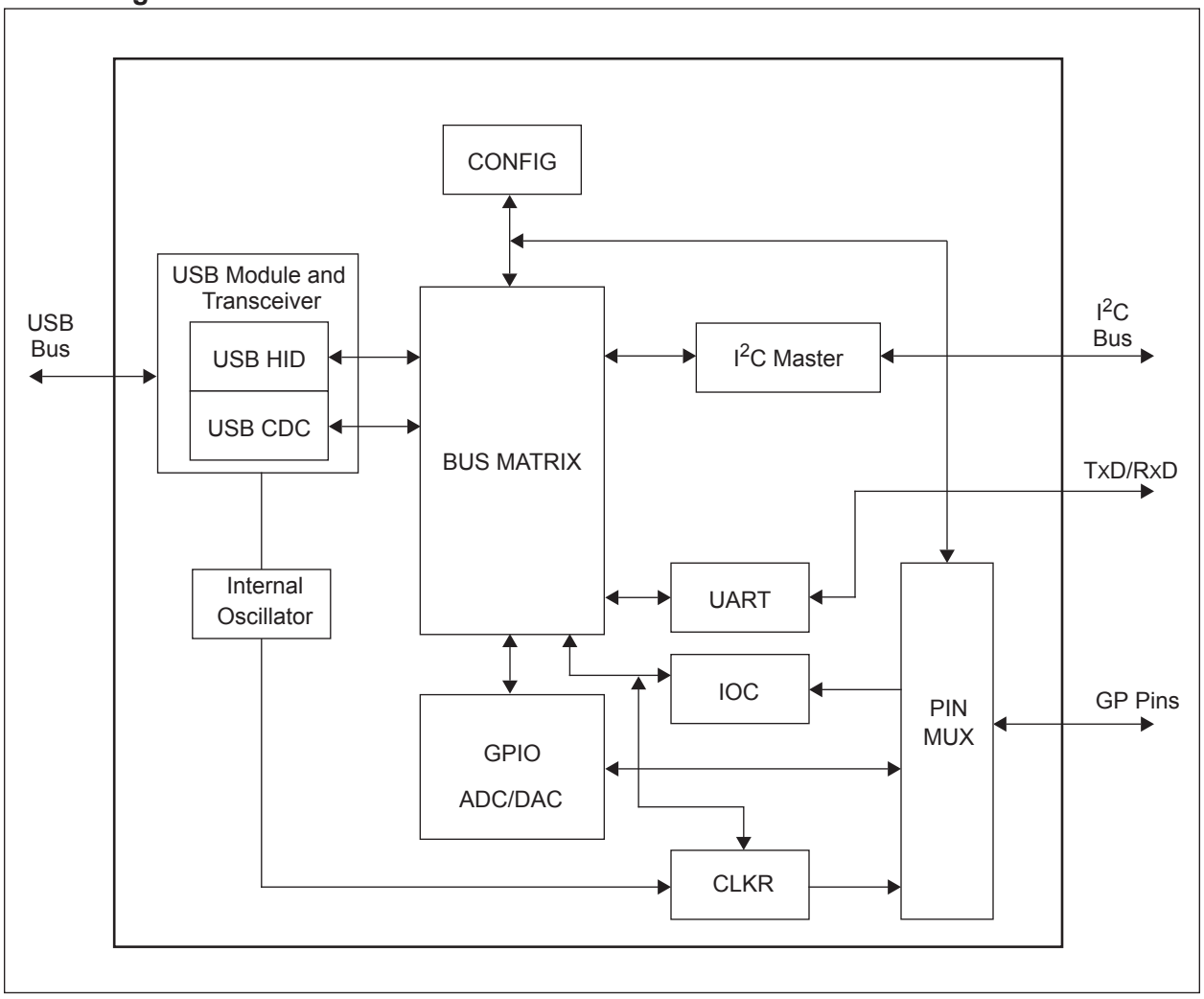


MCP2221A
4 x 4 QFN*



* Includes Exposed Thermal Pad (EP); see Table 1-1.

Block Diagram



1.0 FUNCTIONAL DESCRIPTION

The MCP2221A is a USB-to-UART serial converter that enables USB connectivity in applications that have UART and/or I^2C interfaces. The device reduces external components by integrating the USB termination resistors and the oscillator needed for USB operation.

The MCP2221A has four GP pins for miscellaneous functionalities (including GPIO, USBCFG, SSPND, Clock Output, ADC, DAC and interrupt detector).

See Table 1-1 and **Section 1.7 "Pin MUX Module"** for details about the pin functions.

TABLE 1-1: PINOUT DESCRIPTION

Pin Name	PDIP, SOIC, SSOP	QFN	Pin Type	Standard Function	Alternate Functions	
GP0	2	1	I/O	General purpose I/O or alternate function pin	SSPND (OUT) LED_URX (OUT)	Signals when the host has entered Suspend mode UART RX LED activity output (factory default)
GP1	3	2	I/O	General purpose I/O or alternate function pin	CLKR (OUT) ADC1 (IN) LED_UTX (OUT) IOC (IN)	Clock Reference Output ADC Channel 1 UART TX Led activity output (factory default) External interrupt edge detector
$\overline{RST}$	4	3	I	Reset input (with internal pull-up)	N/A	
URX	5	4	I	UART RX pin (input)	N/A	
UTX	6	5	O	UART TX pin (output)	N/A	
GP2	7	6	I/O	General purpose I/O or alternate function pin	USBCFG (OUT) ADC2 (IN) DAC1 (OUT)	USB device configured status (factory default) ADC Channel 2 DAC Output 1
GP3	8	7	I/O	General purpose I/O or alternate function pin	LED_I2C (OUT) ADC3 (IN) DAC2 (OUT)	USB-I^2C traffic indicator (factory default) ADC Channel 3 DAC Output 2
SDA	9	8	I/O	I^2C Data line	N/A	
SCL	10	9	I/O	I^2C Clock line	N/A	
VUSB	11	10	USB	USB Power pin (internally connected to 3.3V) Should be locally bypassed with a high-quality ceramic capacitor		
D-	12	11	USB	USB D-		
D+	13	12	USB	USB D+		
VSS	14	13	P	Ground		
NC	—	14 15	—	Not Connected		
VDD	1	16	P	Power		
EP	—	17	—	Exposed Thermal Pad (EP) Do not electrically connect.		

1.1 Supported Operating Systems

The following operating systems are supported:

- Windows – XP (SP3), Vista, 7, 8, 8.1 and 10
- Linux® – any distribution with support for CDC and HID classes
- Mac OS® – all versions – beginning with 10.7

1.1.1 ENUMERATION

The MCP2221A enumerates as a composite USB device after POR. The device enumerates as both a Human Interface Device (HID) for I^2C, GPIO control, and as CDC for the USB-to-UART converter.

1.1.1.1 USB HID

The MCP2221A enumerates as an HID, so the device can be configured, while the I^2C and GPIO can be controlled. A DLL package, with example applications and tools, is supplied by Microchip on the device web page, on the Microchip website www.microchip.com.

1.1.1.2 USB CDC

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

1.2 Bus Matrix Module

The Bus Matrix module is the heart of the MCP2221A. All other modules are tied together and controlled via the Bus Matrix module. This module manages the data transfers between the USB and the UART, the I^2C master module, as well as the command requests generated by the USB host controller and commands for controlling the function of the UART, GPIO, ADC, DAC and clock output.

1.2.1 UART

The control module interfaces to the UART and USB modules.

1.2.2 ACCESSING THE DEVICE

The MCP2221A can be accessed for reading and writing via USB host commands. The device cannot be accessed or controlled via the UART interface.

1.3 UART Interface

The MCP2221A UART interface consists of the TX and RX data signals.

The UART is configurable for several baud rates. The available baud rates are listed in Table 1-2.

1.3.1 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, terminal applications (e.g., Putty, RealTerm, Hyperterminal, etc.) send the `SET_LINE_COMMAND` when connecting to the port. The MCP2221A responds by setting the baud rate only.

The other parameters (Data bits, Parity, Stop bits) remain unchanged.

Note:	MCP2221A supports only eight Data bits, no Parity, and one Stop bit.

1.3.1.1 Rounding Errors

Primary baud rate settings (with associated rounding errors) are shown in Table 1-2.

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 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%

EQUATION 1-1: SOLVING FOR ACTUAL BAUD RATE

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

Where:

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

1.3.2 CUSTOM BAUD RATES

Custom baud rates are configured by sending the `SET_LINE_CODING` USB command. See **Section 2.0 "USB Enumeration Process"** for more information.

1.4 Device Configuration

The MCP2221A keeps all the essential device configuration settings stored in Flash memory.

Device configuration settings affect the way the MCP2221A behaves at run time.

The settings are stored into the Flash memory on the device. Some of the settings are also copied into SRAM at Power-Up/Reset.

These device configuration settings reside in the following two distinct areas of Flash memory:

- **Chip Settings**

  The Chip settings area stores the key MCP2221A parameters – USB parameters, ADC/DAC reference voltage choice, start-up DAC value, Clock Reference output (CLKR) frequency and duty cycle values.

- **GP Settings**

  The GP Settings area stores the GP designation settings. For GP settings that are assigned to GPIO output operation, output values (logic `1` or `0`) are also specified.

Even though the MCP2221A places a partial copy of the Chip settings in SRAM, the following Chip settings always reside in Flash:

- USB Manufacturer/Product and Serial Number descriptors
- USB VID and PID pair
- USB options (e.g., the requested amount of current that is presented to the USB host during the USB enumeration process)

1.4.1 POWER-UP/RESET DEVICE CONFIGURATION BEHAVIOR

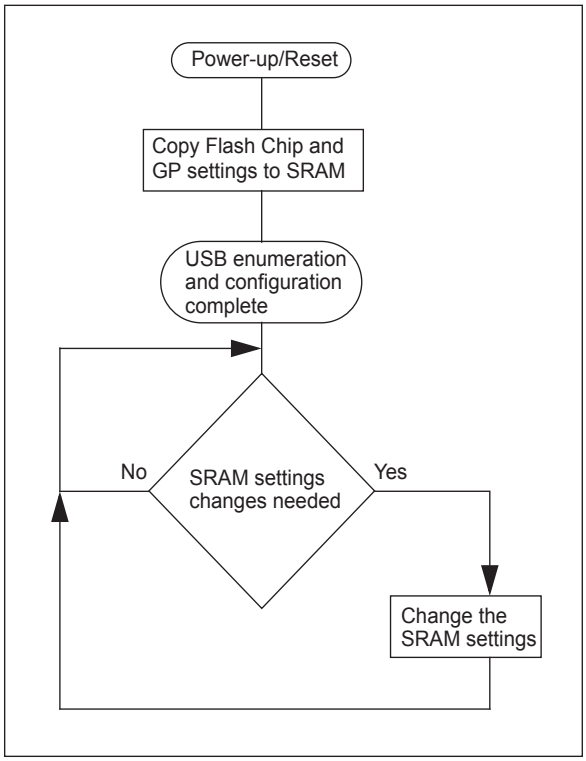
At Power-up/Reset, the MCP2221A configures the device options (GP designation, special function pins parameters and USB enumeration options) according to the Flash settings. Then, the Flash Chip settings and GP settings are loaded into SRAM to allow for their temporary modification at run time.

Chip settings of the device configuration Flash is copied partially into SRAM. Only the run-time-modifiable parameters are copied into SRAM.

GP Settings of the Device Configuration Flash (GP settings area) are copied entirely into the SRAM. By copying the GP settings completely into SRAM, the user is allowed to completely change the GP designation at run time.

The SRAM copy of the settings can be altered at run time in order to change certain device behavior, e.g., GP designation (the GPs can be re-assigned for a different type of operation than the one assigned at power-up) and special parameters (DAC value, ADC/DAC voltage references, clock output value).

FIGURE 1-1: CHIP SETTINGS RUN TIME MANAGEMENT



The SRAM settings (GP and partial Chip settings) can be modified through USB HID commands, and they will have an effect on the following device features:

- GP pin designation (switch between GPIO, dedicated or special functions modes)
- GPIO direction and output value (only for GPIO outputs) – for the GPs assigned to work in GPIO mode
- Clock Output duty cycle and value – if GP1 is assigned for CLKR mode (Clock Reference Output mode), by modifying the SRAM settings, the clock frequency and duty cycle can be changed at run time
- DAC value and voltage reference used – the DAC value setting as well as the voltage reference used for it are stored in SRAM settings and they can be changed at run time. Through this mechanism, at run time the user can change the DAC value, as well as the voltage reference.
- ADC voltage reference value – the voltage reference used for ADC conversions can be changed by altering its corresponding SRAM setting
- Interrupt-on-Change (IOC) detector settings – if GP1 is assigned for IOC mode, the SRAM settings are used for setting up the triggers used for external interrupt detection (positive, negative edge detection or both)

1.4.2 CHIP SETTINGS MAP

The Chip settings area resides in Flash memory and is copied into SRAM at run time. Not all of the device's settings can be altered at run time. All the fields in the Flash settings can be altered by the user.

TABLE 1-3: CHIP SETTINGS MAP

Byte Index	Register Name	Comments
0	CHIPSETTING0	Controls the USB CDC serial number enumeration, default state for the GP LED designation, default state for GP dedicated-function pins and Chip settings protection level
1	CHIPSETTING1	Default clock output divider and duty cycle
2	CHIPSETTING2	DAC reference options and default DAC value
3	CHIPSETTING3	ADC reference and interrupt detection settings
4	USBVIDL	USB VID lower byte
5	USBVIDH	USB VID higher byte
6	USBPIDL	USB PID lower value
7	USBPIDH	USB PID higher byte
8	USBPWRATTR	USB power attributes
9	USBREQCRT	USB required current
10	PASS0	Password byte 0
11	PASS1	Password byte 1
12	PASS2	Password byte 2
13	PASS3	Password byte 3
14	PASS4	Password byte 4
15	PASS5	Password byte 5
16	PASS6	Password byte 6
17	PASS7	Password byte 7

REGISTER 1-1: CHIPSETTING0 REGISTER

R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0
CDCSNEN	LEDURXINST	LEDUTXINST	LEDI2CINST	SSPNDINST	USBCFGINST	CHIPPROT1	CHIPPROT0
bit 7							bit 0

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

bit 7 **CDCSNEN:** USB CDC Serial Number Enable bit
- 1 = USB CDC Serial Number is enumerated
- 0 = No USB CDC Serial Number enumeration (factory default)

bit 6 **LEDURXINST:** LED UART Rx Inactive State bit
- 1 = LED UART Rx is inactive-high and active-low (factory default)
- 0 = LED UART Rx is inactive-low and active-high

bit 5 **LEDUTXINST:** LED UART Tx Inactive State bit
- 1 = LED UART Tx is inactive-high and active-low (factory default)
- 0 = LED UART Tx is inactive-low and active-high

bit 4 **LEDI2CINST:** LED I^2C Inactive State bit
- 1 = LED I^2C is inactive-high and active-low (factory default)
- 0 = LED I^2C is inactive-low and active-high

bit 3 **SSPNDINST:** SSPND Inactive State bit
- 1 = SSPND is inactive-high and active-low (factory default)
- 0 = SSPND is inactive-low and active-high

bit 2 **USBCFGINST:** USBCFG Inactive State bit
- 1 = USBCFG is inactive-high and active-low (factory default)
- 0 = USBCFG is inactive-low and active-high

bit 1-0 **CHIPPROT<1:0>:** Chip Settings Protection Level bits
- 11 =Reserved
- 10 =Permanently locked
- 01 =Password protection
- 00 =Chip settings unprotected (factory default)

REGISTER 1-2: CHIPSETTING1 REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1	R/W-0
—	—	—	CLKDC1	CLKDC0	CLKDIV2	CLKDIV1	CLKDIV0
bit 7							bit 0

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

bit 7-5 **Reserved:** Set to '0'

bit 4-3 **CLKDC<1:0>:** Clock-Out Duty-Cycle bits

11 =Duty cycle 75% (75% of 1 clock period is logic '1' and 25% of 1 clock period is logic '0')
10 =Duty cycle 50% (50% of 1 clock period is logic '1' and 50% of 1 clock period is logic '0') (factory default)
01 =Duty cycle 25% (25% of 1 clock period is logic '1' and 75% of 1 clock period is logic '0')
00 =Duty cycle 0% (100% of 1 clock period is logic '0')

bit 2-0 **CLKDIV<2:0>:** Clock-Out Divider Output bits

111 =375 kHz clock output
110 =750 kHz clock output
101 =1.5 MHz clock output
100 =3 MHz clock output
011 =6 MHz clock output
010 =12 MHz clock output (factory default)
001 =24 MHz clock output
000 =Reserved

REGISTER 1-3: CHIPSETTING2 REGISTER

R/W-1	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0
DACVRM1	DACVRM0	DACREF	DACVAL4	DACVAL3	DACVAL2	DACVAL1	DACVAL0
bit 7							bit 0

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

bit 7-6 **DACVRM<1:0>:** DAC Internal Voltage Reference (DAC VRM) Selection bits

11 =VRM voltage is 4.096V (only if V_{DD} is above this voltage)
10 =VRM voltage is 2.048V (factory default)
01 =VRM voltage is 1.024V
00 =VRM is off

bit 5 **DACREF:** DAC Reference Output Selection bit

1 = DAC reference output is DAC VRM voltage selection
0 = DAC reference output is V_{DD} (factory default)

bit 4-0 **DACVAL<4:0>:** Initial DAC Output Value bit

5-bit value for the DAC output (factory default is 8 decimal)

REGISTER 1-4: CHIPSETTING3 REGISTER

R/W-0	R/W-1	R/W-1	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
—	INTDETFEEN	INTDETREEN	ADCVRM1	ADCVRM0	ADCREF	—	—
bit 7							bit 0

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

bit 7 **Reserved:** Set to '0'

bit 6 **INTDETFEEN:** Interrupt Falling Edge Detect Enable bit
1 = Interrupt detector will trigger when a falling edge is detected (factory default)
0 = Falling edges will not trigger the detector

bit 5 **INTDETREEN:** Interrupt Rising Edge Detect Enable bit
1 = Interrupt detector will trigger when a rising edge is detected (factory default)
0 = Rising edges will not trigger the detector

bit 4-3 **ADCVRM<1:0>:** ADC Internal Voltage Reference (ADC VRM) Selection bits
11 =VRM voltage is 4.096V (only if V_{DD} is above this voltage)
10 =VRM voltage is 2.048V
01 =VRM voltage is 1.024V (factory default)
00 = VRM is off

bit 2 **ADCREF:** ADC Reference Output Selection bit
1 = ADC reference output is ADC VRM voltage selection (factory default)
0 = ADC reference output is V_{DD}

bit 1-0 **Reserved:** Set to '0'

REGISTER 1-5: USBVIDL REGISTER

R/W-1	R/W-1	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0
USBVIDL7	USBVIDL6	USBVIDL5	USBVIDL4	USBVIDL3	USBVIDL2	USBVIDL1	USBVIDL0
bit 7							bit 0

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

bit 7-0 **USBVIDL<7:0>:** USB VID Lower Byte (factory default: 0xD8(hex))

REGISTER 1-6: USBVIDH REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0
USBVIDH7	USBVIDH6	USBVIDH5	USBVIDH4	USBVIDH3	USBVIDH2	USBVIDH1	USBVIDH0
bit 7							bit 0

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

bit 7-0 **USBVIDH<7:0>:** USB VID Higher Byte (factory default: 0x04(hex))

REGISTER 1-7: USBPIDL REGISTER

R/W-1	R/W-1	R/W-0	R/W-1	R/W-1	R/W-1	R/W-0	R/W-1
USBPIDL7	USBPIDL6	USBPIDL5	USBPIDL4	USBPIDL3	USBPIDL2	USBPIDL1	USBPIDL0
bit 7							bit 0

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

bit 7-0 **USBPIDL<7:0>:** USB PID Lower Byte (factory default: 0xDD(hex))

REGISTER 1-8: USBPIDH REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
USBPIDH7	USBPIDH6	USBPIDH5	USBPIDH4	USBPIDH3	USBPIDH2	USBPIDH1	USBPIDH0
bit 7							bit 0

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

bit 7-0 **USBPIDH<7:0>:** USB PID Higher Byte (factory default: 0x00(hex))

REGISTER 1-9: USBPWRATTR REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SELFPWR	REMWKUP	—	—	—	—	—
bit 7							bit 0

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

bit 7 **Reserved:** Reserved – set to '1' (factory default)

bit 6 **SELFPWR:** USB Self-Powered Attribute bit

1 = Chip will enumerate on the USB bus as being self powered
0 = Chip will enumerate on the USB bus as being USB bus powered (factory default)

bit 5 **REMWKUP:** USB Remote Wake-Up Capability bit

1 = Chip will enumerate on the USB bus as being able to wake up the USB host
0 = Chip will enumerate as not being capable of remote wake-up of the USB host (factory default)

bit 4-0 **Reserved:** Set all bits to '0' (factory default)

REGISTER 1-10: USBREQCRT REGISTER

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-1	R/W-0
USBRE-QCRT7	USBRE-QCRT6	USBRE-QCRT5	USBRE-QCRT4	USBRE-QCRT3	USBRE-QCRT2	USBRE-QCRT1	USBRE-QCRT0
bit 7							bit 0

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

bit 7-0 **USBREQCRT<7:0>:** USB Bus-Powered Required Current Amount bits (in units of 2 mA)
Factory default is 50 (decimal); the USB enumeration interprets this value as a current requirement of 100 mA.

REGISTER 1-11: PASS0 – PASS7 REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PASSx7	PASSx6	PASSx5	PASSx4	PASSx3	PASSx2	PASSx1	PASSx0
bit 7							bit 0

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

bit 7-0 **PASSx<7:0>:** Password Byte X Value bits (factory default is 0)

1.4.3 GP SETTINGS MAP

The GP Settings area resides in Flash memory and is copied into SRAM at run time. The user can alter both the Flash and the SRAM GP settings. Any modification in the SRAM copy of the GP settings will have an immediate effect. The GP pins designation changes according to the new content of the SRAM settings. The Flash variant of the settings will affect the Power-Up behavior of the GP pins.

TABLE 1-4: GP SETTINGS MAP

Byte Index	Register Name	Comments
0	GPSETTING0	GP0 pin designation and GPIO default output value, when GP is set for GPIO output operation
1	GPSETTING1	GP1 pin designation and GPIO default output value, when GP is set for GPIO output operation
2	GPSETTING2	GP2 pin designation and GPIO default output value, when GP is set for GPIO output operation
3	GPSETTING3	GP3 pin designation and GPIO default output value, when GP is set for GPIO output operation

REGISTER 1-12: GPSETTING0 REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1	R/W-0
—	—	—	GPIOOUTVAL	GPIODIR	GPDES2	GPDES1	GPDES0
bit 7							bit 0

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

bit 7-5 **Reserved:** Set to ‘0’

bit 4 **GPIOOUTVAL:** GPIO Output Value bit (valid only when GP0 is set for GPIO output operation)
1 = Default output value is logic ‘1’ (factory default)
0 = Default output value is logic ‘0’

bit 3 **GPIODIR:** GPIO Direction bit (input or output; valid only when GP0 is set for GPIO operation)
1 = GPIO Input
0 = GPIO Output (factory default)

bit 2-0 **GPDES<2:0>:** GP0 Designation bits
111 =Reserved
110 =Reserved
101 =Reserved
100 =Reserved
011 =Reserved
010 =Alternate function 0 (LED UART Rx – LEDURX) (factory default)
001 =Dedicated function operation (SSPND)
000 =GPIO operation (GPIO0)

REGISTER 1-13: GPSETTING1 REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-1	R/W-1
—	—	—	GPIOOUTVAL	GPIODIR	GPDES2	GPDES1	GPDES0
bit 7							bit 0

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

bit 7-5 **Reserved:** Set to '0'

bit 4 **GPIOOUTVAL:** GPIO Output Value bit (valid only when GP1 is set for GPIO output operation)
1 = Default output value is logic '1' (factory default)
0 = Default output value is logic '0'

bit 3 **GPIODIR:** GPIO Direction bit (input or output; valid only when GP1 is set for GPIO operation)
1 = GPIO Input
0 = GPIO Output (factory default)

bit 2-0 **GPDES<2:0>:** GP1 Designation bits
111 =Reserved
110 =Reserved
101 =Reserved
100 =Alternate function 2 (Interrupt detector)
011 =Alternate function 1 (LED UART Tx – LEDUTX) (factory default)
010 =Alternate function 0 (ADC1)
001 =Dedicated function operation (Clock Output)
000 =GPIO operation (GPIO1)

REGISTER 1-14: GPSETTING2 REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1
—	—	—	GPIOOUTVAL	GPIODIR	GPDES2	GPDES1	GPDES0
bit 7							bit 0

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

bit 7-5 **Reserved:** Set to '0'

bit 4 **GPIOOUTVAL:** GPIO Output Value bit (valid only when GP2 is set for GPIO output operation)
1 = Default output value is logic '1' (factory default)
0 = Default output value is logic '0'

bit 3 **GPIODIR:** GPIO Direction bit (input or output; valid only when GP2 is set for GPIO operation)
1 = GPIO Input
0 = GPIO Output (factory default)

bit 2-0 **GPDES<2:0>:** GP2 Designation bits
111 =Reserved
110 =Reserved
101 =Reserved
100 =Reserved
011 =Alternate function 1 (DAC1)
010 =Alternate function 0 (ADC2)
001 =Dedicated function operation (USBCFG) (factory default)
000 =GPIO operation (GPIO2)

REGISTER 1-15: GPSETTING3 REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1
—	—	—	GPIOOUTVAL	GPIODIR	GPDES2	GPDES1	GPDES0
bit 7							bit 0

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

bit 7-5 **Reserved:** Set to '0'

bit 4 **GPIOOUTVAL:** GPIO Output Value bit (valid only when GP3 is set for GPIO output operation)
1 = Default output value is logic '1' (factory default)
0 = Default output value is logic '0'

bit 3 **GPIODIR:** GPIO Direction bit (input or output; valid only when GP3 is set for GPIO operation)
1 = GPIO Input
0 = GPIO Output (factory default)

bit 2-0 **GPDES<2:0>:** GP3 Designation bits
111 =Reserved
110 =Reserved
101 =Reserved
100 =Reserved
011 =Alternate function 1 (DAC2)
010 =Alternate function 0 (ADC3)
001 =Dedicated function operation (LEDI2C) (factory default)
000 =GPIO operation (GPIO3)

1.5 USB Module (HID, CDC and Transceiver Modules)

The USB HID and CDC modules in the MCP2221A are full-speed USB 2.0-compliant.

- Composite Device (CDC + HID):
  - CDC: USB-to-UART Communications
  - HID: I^2C Transactions, GPIO Control, Configuration and Miscellaneous Operations (ADC, DAC, Clock Output)
- 448-byte buffer to handle data throughput at any UART Baud Rate:
  - 64-Byte Transmit
  - 384-Byte Receive
- Fully configurable VID and PID assignments and descriptors (stored on-chip)
- Bus Powered or Self Powered

1.5.1 DESCRIPTORS

During configuration, the supplied PC interface stores the descriptors in the MCP2221A.

1.5.2 SUSPEND AND RESUME

The USB Suspend and Resume signals are supported for power management of the MCP2221A. The device enters Suspend mode when “Suspend Signaling” is detected on the bus.

The MCP2221A exits Suspend mode when any of the following events occur:

- “Resume Signaling” is detected or generated
- A USB “Reset” signal is detected
- A device Reset occurs

1.6 USB Transceiver

The MCP2221A has a built-in USB 2.0 full-speed transceiver internally connected to the USB module.

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

1.6.1 INTERNAL PULL-UP RESISTORS

The MCP2221A devices have built-in pull-up resistors designed to meet the requirements for full-speed USB.

1.6.2 MCP2221A POWER OPTIONS

The following are the main power options for the MCP2221A:

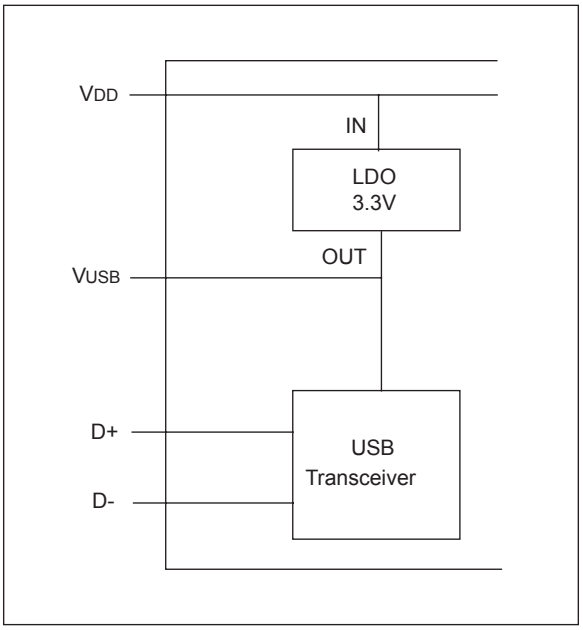
- USB Bus Powered (5V)
- 3.3V Self Powered

1.6.2.1 Internal Power Supply Details

MCP2221A offers various options for power supply. To meet the required USB signaling levels, the MCP2221A incorporates an internal LDO that is used solely by the USB transceiver to present the correct D+/D- voltage levels.

Figure 1-2 shows the internal connections of the USB transceiver LDO in relation to the VDD power supply rail. The output of the USB transceiver LDO is tied to the VUSB line. A capacitor connected to the VUSB pin is required if the USB transceiver LDO provides the 3.3V supply to the transceiver.

FIGURE 1-2: MCP2221A INTERNAL POWER SUPPLY DETAILS



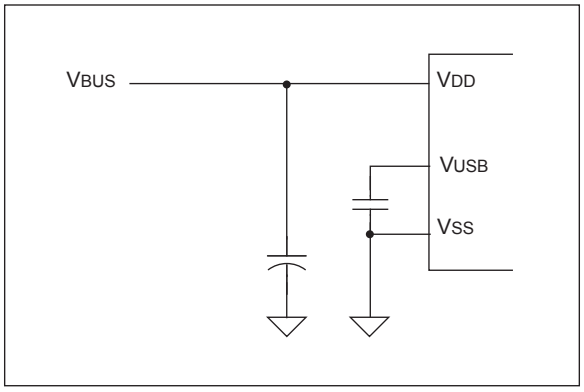
The provided VDD voltage has a direct influence on the voltage levels present on the GPIO and UART TX/RX pins. When VDD is 5V, all of these pins will have a logical '1' around 5V with the variations specified in **Section 4.1 "DC Characteristics"**.

For applications that require a 3.3V logical '1' level, VDD must be connected to a power supply providing the 3.3V voltage. In this case, the internal USB transceiver LDO cannot provide the required 3.3V power. It is necessary to also connect the VUSB pin of the MCP2221A to the 3.3V power supply rail. This way, the USB transceiver is powered up directly from the 3.3V power supply.

1.6.2.2 USB Bus-Powered (5V)

In Bus Power Only mode, all power for the application is drawn from the USB (Figure 1-3). This is effectively the simplest power method for the device.

FIGURE 1-3: BUS POWER ONLY



In order to meet the inrush current requirements of the USB 2.0 specifications, the total effective capacitance appearing across VBUS and ground must be no more than 10 µF. If it is more than 10 µF, some kind of inrush limiting is required. For more details on Inrush Current Limiting, search for that subject in the latest "*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 µA (or 2.5 mA for high-powered devices that are remote wake-up capable) from the 5V VBUS line of the USB cable.

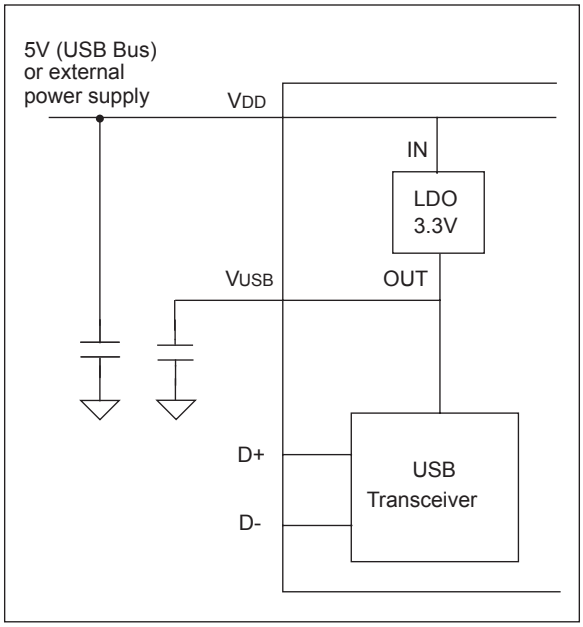
The host signals the USB device to enter Suspend mode by stopping all USB traffic to that device for more than 3 ms.

The USB bus provides a 5V voltage. However, the USB transceiver requires 3.3V for the signaling (on D+ and D- lines).

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 µA/2.5 mA). The VUSB pin requires an external bypass capacitor with a value between 0.22 and 0.47 µF (ceramic cap).

Figure 1-4 shows a circuit where MCP2221A's internal LDO is used to provide 3.3V to the USB transceiver. The voltage on the VDD affects the voltage levels onto the UART and GPIO pins. With VDD at 5V, these pins will have a logic '1' of 5V with the variations specified in **Section 4.1 "DC Characteristics"**.

FIGURE 1-4: TYPICAL POWER SUPPLY OPTION USING THE 5V PROVIDED BY THE USB



1.6.2.3 3.3V Self-Powered

Typically, many embedded applications are using 3.3V power supplies. When such option is available in the target system, MCP2221A can be powered up from the existing 3.3V power supply rail. The typical connections for MCP2221A are shown in Figure 1-5.

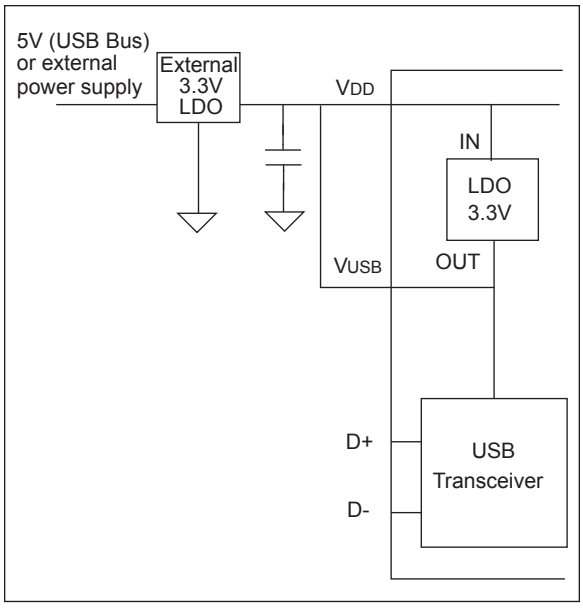
In this example, MCP2221A has both VDD and VUSB lines tied to the 3.3V rail. These tied connections disable the internal USB transceiver LDO of the MCP2221A to regulate the power supply on the VUSB pin. Another consequence is that the '1' logical level on the GPIO pins will be at the 3.3V level, in accordance with the variations specified in **Section 4.1 "DC Characteristics"**.

1.6.2.4 Remote Wake-Up Capability

The MCP2221A offers a mechanism for triggering a remote wake-up event for the USB host. The remote wake-up trigger works only with the external interrupt detector.

In order to use this capability, GP1 must be designated for interrupt detection operation. Before the USB host goes into Sleep/Standby, the interrupt detector must be set up for detecting positive edges, negative edges or both; also, the detector flag must be cleared. After these conditions are met, the USB host can go into Sleep/Standby mode and it will be awakened whenever an external signal on GP1 triggers the interrupt detector.

FIGURE 1-5: USING AN EXTERNALLY PROVIDED 3.3V POWER SUPPLY



1.7 Pin MUX Module

The pin MUX module offers multiple functionalities for the GP pins.

1.7.1 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
- LED_URX – indicates UART receive traffic (when seen from the MCP2221A)
- LED_UTX – indicates UART transmit traffic (when seen from the MCP2221A)
- LED_I2C – indicates I^2C traffic
- ADC1/2/3 – analog inputs connected to the internal 10-bit ADC
- DAC1/2 – analog outputs connected to the same 5-bit DAC
- CLKR – digital clock output (the nominal value is 12 MHz, but other values are possible)
- IOC – external interrupt detector

1.7.1.1 GPIO Pin Function

When the GPIO pin function is enabled for a given GP(n) pin, it will operate as a digital input or an output pin. When configured as a digital output, its value is controlled through the USB HID commands. When configured as a digital input, its logic value is read using USB HID commands.

1.7.1.2 SSPND Pin Function

The SSPND pin (if enabled) reflects the USB state (Suspend/Resume). The pin is active-low (factory-default setting; see the CHIPSETTING0 register for more details) 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.

1.7.1.3 USBCFG Pin Function

The USBCFG pin (if enabled) starts out low (factory default setting; see the CHIPSETTING0 register for more details) 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.

1.7.1.4 LED_URX

The “RX” in this pin name refers to the UART of the MCP2221A. The LED_URX pin is an indicator of UART RX characters being received.

This pin will pulse low or high (depending on the chip configuration settings; see the CHIPSETTING0 register for more details) for a period of time (a few milliseconds). This allows the application to provide a visual indication of the UART RX traffic.

1.7.1.5 LED_UTX

The “TX” in this pin name refers to the UART of the MCP2221A. The LED_UTX pin is an indicator of UART TX characters being transmitted.

This pin will pulse low or high (depending on the chip configuration settings; see the CHIPSETTING0 register for more details) for a period of time (a few milliseconds). This allows the application to provide a visual indication of the UART TX traffic.

1.7.1.6 LED_I2C

The “I2C” in this pin name refers to the I^2C module in the MCP2221A. The LED_I2C pin is an indicator of I^2C activity.

This pin will pulse low or high (depending on the chip configuration settings; see the CHIPSETTING0 register for details) for a period of time (a few milliseconds). This allows the application to provide a visual indication of the I^2C traffic.

1.7.1.7 ADC1/2/3

When GP1/2/3 are configured for ADC operation, they will work as analog input pins and they are tied to the first three channels of the 10-bit ADC in the MCP2221A.

1.7.1.8 DAC1/2

When GP2/3 are configured for DAC operation, they will work as analog output pins and they are tied to the output of the MCP2221A’s 5-bit DAC.

TABLE 1-5: GP DESIGNATION TABLE

GP Designation Bits <2:0>	Assignment	GP0	GP1	GP2	GP3
000	GPIO	GPIO	GPIO	GPIO	GPIO
001	DEDICATED_FUNC	SSPND	CLK OUT	USBCFG	LED_I2C
010	ALT_FUNC_0	LED_URX	ADC1	ADC2	ADC3
011	ALT_FUNC_1	—	LED_UTX	DAC1	DAC2
100	ALT_FUNC_2	—	IOC	—	—

1.8 GPIO/ADC/DAC Module

This module communicates with the USB HID sub-module through the Bus Matrix module. It allows the manipulation of GPIOs, retrieving the ADC data and setting the DAC value.

1.8.1 GPIO

When the GPs are configured for GPIO operation, those configured GPs can be used as digital inputs or outputs.

When working as outputs, the GPs output logic levels (logic `0` or `1`).

1.8.1.1 VRM

The ADC and DAC sub-modules each have a voltage reference module (VRM). Each VRM can be configured (at power-up and run time) to output one of the four voltage choices available.

The VRM can provide the following voltages as a reference:

- VDD – the VRM output is exactly the voltage present at the VDD pin of the MCP2221A. It can take any value from 3.3 to 5V.
- 1.024V – the VRM output of 1.024V is obtained from an internal voltage reference
- 2.048V – the VRM output of 2.048V is obtained from an internal voltage reference
- 4.096V – the VRM output of 4.096V is obtained from an internal voltage reference. If the VDD is lower than 4.096V, the VRM output will have the value of VDD.

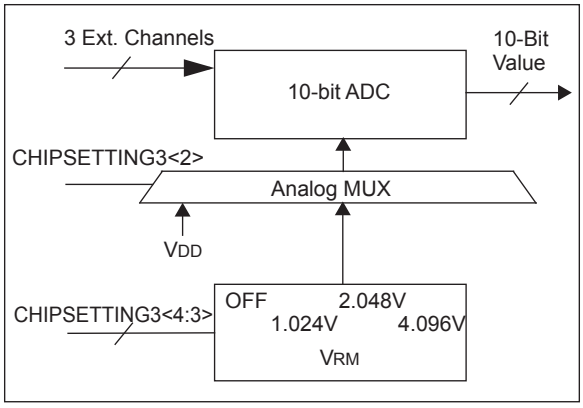
1.8.2 ADC CONVERTER

The ADC converter is producing 10-bit values and it uses its own VRM module.

It features three external channels (connected to GP1/2/3 if configured for ADC operation).

The sampling rate of the ADC is around 1000 sps.

FIGURE 1-6: ADC SUB-MODULE DETAILS



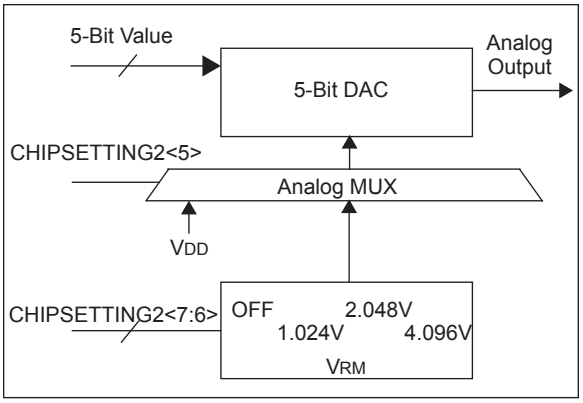
1.8.3 DAC CONVERTER

The DAC is 5-bit wide, has a single analog output and it uses its own VRM module.

The DAC output voltage can be routed to GP2/3 (if GP2/3 are configured for DAC operation).

If the GP2 and GP3 are configured for DAC operation, they will present the same analog voltage value because they are connected to the same DAC output.

FIGURE 1-7: DAC SUB-MODULE DETAILS



1.9 CLKR

When GP1 is configured for clock output operation, the GP1 pin will act as a digital output, providing a clock signal derived from the device's internal clock. The clock's nominal frequency is 12 MHz ± 0.25%. Other clock values and duty cycles are possible by setting different values that are associated with this mode of operation.

1.10 IOC

When GP1 is configured for Interrupt-on-Change (IOC) operation, GP1 acts as a digital input that is sensitive to positive and negative edges. Depending on the settings associated with this mode of operation, the GP1 can detect positive, negative or both edges.

1.11 RESET/POR

1.11.1 RESET PIN

The $\overline{RST}$ 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.

1.11.2 POR

A POR pulse is generated on-chip whenever V_{DD} rises above a certain threshold. This allows the device to start in the initialized state when V_{DD} is adequate for operation.

To take advantage of the POR circuitry, tie the $\overline{RST}$ pin to V_{DD} through a resistor (1 – 10 kΩ). This will eliminate external RC components usually needed to create a POR 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.

1.12 Internal Oscillator

The MCP2221A features an internal oscillator that provides a 12 MHz clock, which is needed for the USB modules (HID and CDC).

Full-speed USB is nominally 12 Mb/s. The clock signal's accuracy is over temp (2,500 ppm maximum).

The internal clock of the MCP2221A is fed into the CLKR module to provide a clock signal outside of the device.

GP1 can be configured as a clock output pin providing a 12 MHz clock to the rest of the system. Other clock and duty-cycle values are possible by using different settings for this module.

1.13 I^2C Master Module

The I^2C master module is responsible for the I^2C traffic generation. The module is controlled through the USB HID, through the Bus Matrix module.

The I^2C module only implements the functionality of an I^2C/SMBus master.

1.13.1 I^2C/SMBUS MASTER

The I^2C master initiates all the I^2C/SMBus transactions (being read or write operations) on the bus.

The I^2C/SMBus master module has the following capabilities:

- sending/receiving data at a multitude of bit rates, up to 400 kbps
- 7-bit Addressing mode
- single data transfers of up to 65,535 bytes
- clock-stretching (it allows the slower I^2C Slaves to communicate)

All the user data to be sent/transmitted over the I^2C bus is conveyed to the USB host only through the USB HID interface.

1.14 Bus Matrix Module

The Bus Matrix module manages the communication between various functional modules, such as: USB (HID and CDC), I^2C, UART, GPIO/ADC/DAC, Config, IOC, CLKR, pin MUX.

1.15 Config Module

The Config module is in charge of the storage of the device settings and also of their management (loading/modifying/access protection). The module uses nonvolatile memory for storing the power-up device settings.

At power-up, the module loads the settings from the nonvolatile storage area into an SRAM location (volatile settings). These settings represent the device's configuration, along with other key parameters (e.g., string descriptors, VID/PID, etc.). After the settings are loaded in SRAM (volatile settings), they can be changed through the USB HID interface.

The user can read/modify/change either settings (nonvolatile or volatile) through the same interface (USB HID).

The Config module contains the relevant power-up settings that are used by the MCP2221A. A few examples of settings are: USB descriptors, GP settings, ADC, DAC, CLKR.

2.0 USB ENUMERATION PROCESS

The MCP2221A implements the CDC class to support the USB-to-UART protocol converter functionality. Using USB-to-UART (CDC class) adapters with personal computers running the Windows operating system (OS) requires some consideration because of the way the Windows OS responds to their connection.

When a USB-to-UART (CDC class) adapter is connected to the USB port of the PC, Windows searches for a driver. After a suitable driver is found, the system creates an entry in the registry. The entry stores relevant information about the USB-to-UART adapter, its driver and the associated COM port.

The COM port and its number are legacy-type adapters, which are still supported by Windows OS. Historically, the COM ports in a computer are part of the computer's motherboard and are assigned a different index number. With the advent of USB-to-UART adapters, the Windows OS kept the COM port concept and extended it to support the USB adapters.

Whenever a USB-to-UART adapter is first connected to a PC, the system searches the registry for an entry that is suitable for the connected adapter. If one is not found, the system asks for a suitable driver. If this step is completed, it creates a registry entry, and assigns a COM port number as well. Then, whenever the USB-to-UART adapter is connected to that PC, the system checks the registry entry, loads the specified driver and assigns the given COM port number (as found in the registry entry).

During the enumeration process, the device can specify a serial number. If it does, this number is stored in the registry entry and it is used to assign the same COM port number to the adapter in question, no matter which USB port the adapter is connected to.

USB-to-UART adapters have the option to not present a serial number during USB enumeration. In this case, the operating system would not be able to differentiate between two identical devices, if neither is providing its serial number. Each time one of these two devices (with no serial number provided during enumeration) is connected to the same USB port, they will have the same COM port number assigned.

Both functionalities (with or without serial numbers) are very useful for different applications.

When the serial number is provided, an adapter using the MCP2221A solution receives the same COM port number from a Windows machine, no matter which USB port they are connected to.

The case with no serial number is useful for test/validation of products using the MCP2221A. The fact that all the tested boards are not supplying a serial number will force Windows to assign them the same COM port number (but only if connected to the same USB port).

The MCP2221A is factory-set to not use a serial number. Later in the process, if a customer wants the benefits provided by using a serial number, the Configuration Utility from Microchip can be used to enable the MCP2221A to enumerate its serial number as well.

The MCP2221A comes with a uniquely-provided serial number to be used during the USB enumeration process; however, this can be changed by the user in the Configuration Utility.

The serial number enumeration enable/disable can be changed, as well, using the Configuration Utility.

All the USB-related settings mentioned above are part of the Device Configuration (Chip settings area) and they reside only in Flash. When the Chip settings area (1st area) is being copied into the SRAM (at power-up), the USB settings are skipped (not copied into SRAM).

3.0 USB HID COMMUNICATION

Except for the USB CDC and UART modules, all the other modules in the MCP2221A use USB HID protocol for communication.

The USB HID protocol uses 64-byte reports.

A typical command exchange starts with a 64-byte packet that is written by the USB host (i.e., the PC). Afterward, the USB host reads the response from the device as a 64-byte packet.

3.1 USB HID Commands/Responses

3.1.1 STATUS/SET PARAMETERS

This command offers many options for this device. It is used to poll for the status of the device. It is also used to establish certain I^2C bus parameters/conditions.

TABLE 3-1: COMMAND STRUCTURE

Byte Index	Function Description	Value	Effect
0	—	0x10	`Status/Set Parameters` – command code
1	Don't care	Any value	
2	Cancel current I^2C/SMBus transfer (sub-command)	0x10	When this value is put in this field, the device will cancel the current I^2C/SMBus transfer and will attempt to free the I^2C bus. This command is very useful since it can cancel a transfer and free the bus. An example would be when trying to communicate with a device using a wrong address. This will cause a timeout to occur. This time-out situation can be read using the "Status/Set Parameter" and the cancellation of the I^2C/SMBus transfer can be achieved by this sub-command.
		Any other value	No effect.
3	Set I^2C/SMBus communication speed (sub-command)	0x20	When this value is put in this field, the device will take the next command field and interpret it as the system clock divider that will give the I^2C/SMBus communication clock.
		Any other value	No effect.
4	The I^2C/SMBus system clock divider that will be used to establish the communication speed	—	The value in this field is being taken into consideration only when the Byte Index 3 contains the code for establishing a new communication speed. In all the other cases, this field's value will not matter.
5-63	Don't care	Any value	

3.1.1.1 Responses

TABLE 3-2: RESPONSE 1 STRUCTURE

Byte Index	Function Description	Value	Effect
0	—	0x10	`Status/Set Parameters` – command code echo
1	—	0x00	Command completed successfully
2	Cancel transfer	0x00	No special operation (i.e., Cancel current I^2C/SMBus transfer)
		0x10	The current I^2C/SMBus transfer was marked for cancellation. The actual I^2C/SMBus transfer cancellation and bus release will need some time (a few hundreds of microseconds, depending on the communication speed initially chosen for the canceled transfer)
		0x11	The I^2C engine (inside MCP2221A) was already in Idle mode. The cancellation command had no effect.
3	—	0x00	No Set I^2C/SMBus communication speed was issued.
		0x20	The new I^2C/SMBus communication speed is now considered.
		0x21	The I^2C/SMBus communication speed was not set (e.g., I^2C transfer in progress).
4	—	The divider value given at the same index in the command field	Only in the case when the code for establishing a new communication speed is given at Byte Index 3.
		0x00	When the communication speed is not being set.
5-7	Don’t care	Any value	
8	Internal I^2C state machine state value	—	
9	Lower byte (16-bit value) of the requested I^2C transfer length	—	
10	Higher byte (16-bit value) of the requested I^2C transfer length	—	
11	Lower byte (16-bit value) of the already transferred (through I^2C) number of bytes	—	
12	Higher byte (16-bit value) of the already transferred (through I^2C) number of bytes	—	
13	Internal I^2C data buffer counter	—	
14	Current I^2C communication speed divider value	—	
15	Current I^2C time-out value	—	
16	Lower byte (16-bit value) of the I^2C address being used	—	
17	Higher byte (16-bit value) of the I^2C address being used	—	
18-21	Don’t care	Any value	
22	SCL line value – as read from the pin	—	

TABLE 3-2: RESPONSE 1 STRUCTURE (CONTINUED)

Byte Index	Function Description	Value	Effect
23	SDA line value – as read from the pin	—	
24	Interrupt edge detector state	0 or 1	
25	I^2C Read pending value	0, 1 or 2	This field is used by the USB host to know if the MCP2221A still has to read from a slave device.
26-45	Don't care	Any value	
46	MCP2221A Hardware Revision Major ('A')	—	
47	MCP2221A Hardware Revision Minor ('6')	—	
48	MCP2221A Firmware Revision Major ('1')	—	
49	MCP2221A Firmware Revision Minor ('1')	—	
50-55	ADC Data (16-bit) values.	—	3 x (16-bit) little-endian ADC channel values (CH0 LSB, CH0 MSB, CH1 LSB, CH1 MSB, CH2 LSB, CH2 MSB).
56-63	Don't care	Any value	