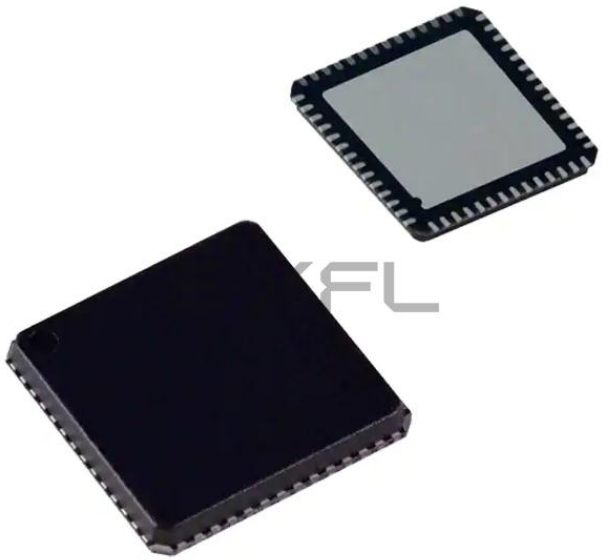


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8052
Core Size	8-Bit
Speed	16.78MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, PSM, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data Converters	A/D 8x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	56-VFQFN Exposed Pad, CSP
Supplier Device Package	56-LFCSP-VQ (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc842bcpz32-5

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
DAC AC CHARACTERISTICS				
Voltage Output Settling Time	15	15	µs typ	Full-scale settling time to within ½ LSB of final value
Digital-to-Analog Glitch Energy	10	10	nV-sec typ	1 LSB change at major carry
DAC CHANNEL SPECIFICATIONS[12, 13]				
Internal Buffer Disabled ADuC841/ADuC842 Only				
DC ACCURACY[10]				
Resolution	12	12	Bits	
Relative Accuracy	±3	±3	LSB typ	
Differential Nonlinearity[11]	–1	–1	LSB max	Guaranteed 12-bit monotonic
	±1/2	±1/2	LSB typ	
Offset Error	±5	±5	mV max	V_{REF} range
Gain Error	±0.5	±0.5	% typ	V_{REF} range
Gain Error Mismatch[4]	0.5	0.5	% typ	% of full-scale on DAC1
ANALOG OUTPUTS				
Voltage Range_0	0 to V_{REF}	0 to V_{REF}	V typ	DAC V_{REF} = 2.5 V
REFERENCE INPUT/OUTPUT REFERENCE OUTPUT[14]				
Output Voltage (V_{REF})	2.5	2.5	V	
Accuracy	±10	±10	mV Max	Of V_{REF} measured at the C_{REF} pin T_A = 25°C
Power Supply Rejection	65	67	dB typ	
Reference Temperature Coefficient	±15	±15	ppm/°C typ	
Internal V_{REF} Power-On Time	2	2	ms typ	
EXTERNAL REFERENCE INPUT[15]				
Voltage Range (V_{REF}) [4]	1	1	V min	
	V_{DD}	V_{DD}	V max	
Input Impedance	20	20	kΩ typ	
Input Leakage	1	1	µA max	Internal band gap deselected via ADCCON1.6
POWER SUPPLY MONITOR (PSM)				
DV_{DD} Trip Point Selection Range		2.93	V min	Two trip points selectable in this range programmed via TPD1–0 in PSMCON, 3 V part only
		3.08	V max	
DV_{DD} Power Supply Trip Point Accuracy		±2.5	% max	
WATCHDOG TIMER (WDT) [4]				
Timeout Period	0	0	ms min	Nine timeout periods selectable in this range
	2000	2000	ms max	
FLASH/EE MEMORY RELIABILITY CHARACTERISTICS[16]				
Endurance[17]	100,000	100,000	Cycles min	
Data Retention[18]	100	100	Years min	
DIGITAL INPUTS				
Input Leakage Current (Port 0, $\overline{EA}$)	±10	±10	µA max	V_{IN} = 0 V or V_{DD}
	±1	±1	µA typ	V_{IN} = 0 V or V_{DD}
Logic 1 Input Current				
(All Digital Inputs), SDATA, SCLOCK	±10	±10	µA max	V_{IN} = V_{DD}
	±1	±1	µA typ	V_{IN} = V_{DD}
Logic 0 Input Current (Ports 1, 2, 3) SDATA, SCLOCK	–75	–25	µA max	
	–40	–15	µA typ	V_{IL} = 450 mV
Logic 1 to Logic 0 Transition Current (Ports 2 and 3)	–660	–250	µA max	V_{IL} = 2 V
	–400	–140	µA typ	V_{IL} = 2 V
RESET	±10	±10	µA max	V_{IN} = 0 V
	10	5	µA min	V_{IN} = 5 V, 3 V Internal Pull Down
	105	35	µA max	V_{IN} = 5 V, 3 V Internal Pull Down

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
POWER REQUIREMENTS[19, 20]				
Power Supply Voltages				
AV_{DD}/DV_{DD} – AGND		2.7	V min	AV_{DD}/DV_{DD} = 3 V nom
		3.6	V max	
	4.75		V min	AV_{DD}/DV_{DD} = 5 V nom
	5.25		V max	
Power Supply Currents **Normal** Mode[21]				
DV_{DD} Current[4]	10	4.5	mA typ	Core CLK = 2.097 MHz
AV_{DD} Current	1.7	1.7	mA max	Core CLK = 2.097 MHz
DV_{DD} Current	38	12	mA max	Core CLK = 16.78MHz/8.38 MHz 5 V/3 V
	33	10	mA typ	Core CLK = 16.78MHz/8.38 MHz 5 V/3 V
AV_{DD} Current	1.7	1.7	mA max	Core CLK = 16.78MHz/8.38 MHz 5 V/3 V
DV_{DD} Current[4]	45	N/A	mA max	Core CLK = 20MHz **ADuC841 Only**
Power Supply Currents **Idle** Mode[21]				
DV_{DD} Current	4.5	2.2	mA typ	Core CLK = 2.097 MHz
AV_{DD} Current	3	2	µA typ	Core CLK = 2.097 MHz
DV_{DD} Current[4]	12	5	mA max	Core CLK = 16.78 MHz/8.38 MHz 5 V/3 V
	10	3.5	mA typ	Core CLK = 16.78 MHz/8.38 MHz 5 V/3 V
AV_{DD} Current	3	2	µA typ	Core CLK = 16.78 MHz/8.38 MHz 5 V/3 V
Power Supply Currents **Power-Down** Mode[21]				Core CLK = any frequency
DV_{DD} Current	28	18	µA max	Oscillator Off / TIMECON.1 = 0
	20	10	µA typ	
AV_{DD} Current	2	1	µA typ	Core CLK = any frequency, **ADuC841 Only**
DV_{DD} Current[4]	3	1	mA max	TIMECON.1 = 1
DV_{DD} Current[4]	50	22	µA max	Core CLK = any frequency
	40	15	µA typ	**ADuC842/ADuC843 Only,** oscillator on
Typical Additional Power Supply Currents				
PSM Peripheral	15	10	µA typ	AV_{DD} = DV_{DD}
ADC[4]	1.0	1.0	mA min	MCLK Divider = 32
	2.8	1.8	mA max	MCLK Divider = 2
DAC	150	130	µA typ	

See footnotes on the next page.

Pin No.	Mnemonic	Type[1]	Description
11	P1.4/ADC4		Input Port 1 (P1.4). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input 4 (ADC4). Channel selection is via ADCCON2 SFR.
12	P1.5/ADC5/$\overline{SS}$	I	Input Port 1 (P1.5). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input 5 (ADC5). Channel selection is via ADCCON2 SFR.
			Slave Select Input for the SPI Interface ($\overline{SS}$).
13	P1.6/ADC6	I	Input Port 1 (P1.6). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input 6 (ADC6). Channel selection is via ADCCON2 SFR.
14	P1.7/ADC7	I	Input Port 1(P1.7). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input 7 (ADC7). Channel selection is via ADCCON2 SFR.
15	RESET	I	Reset. Digital Input. A high level on this pin for 24 master clock cycles while the oscillator is running resets the device.
16	P3.0/RxD	I/O	Input/Output Port 3 (P3.0). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-up resistors.
			Receiver Data Input (Asynchronous) or Data Input/Output (Synchronous) of the Serial (UART) Port (RxD).
17	P3.1/TxD	I/O	Input/Output Port 3 (P3.1). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-up resistors.
			Transmitter Data Output (Asynchronous) or Clock Output (Synchronous) of the Serial (UART) Port (TxD).
18	P3.2/$\overline{INT0}$	I/O	Input/Output Port 3 (P3.2). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-up resistors.
			Interrupt 0 ($\overline{INT0}$). Programmable edge or level triggered interrupt input; can be programmed to one of two priority levels. This pin can also be used as a gate control input to Timer 0.
19	P3.3/$\overline{INT1}$/MISO/PWM1	I/O	Input/Output Port 3 (P3.3). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-up resistors.
			Interrupt 1 ($\overline{INT1}$). Programmable edge or level triggered interrupt input; can be programmed to one of two priority levels. This pin can also be used as a gate control input to Timer 1.
			SPI Master Input/Slave Output Data I/O Pin for SPI Serial Interface (MISO).
			PWM 1 Voltage Output (PWM1). See the CFG841/CFG842 register for further information.
20, 34, 48	DV_{DD}	P	Digital Positive Supply Voltage. 3 V or 5 V nominal.
21, 35, 47	DGND	G	Digital Ground. DGND is the ground reference point for the digital circuitry.

Pin No.	Mnemonic	Type[1]	Description
24	P3.4/T0/PWMC/PWM0/EXTCLK	I/O	Input/Output Port 3 (P3.4). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-up resistors. Timer/Counter 0 Input (T0). PWM Clock Input (PWMC). PWM 0 Voltage Output (PWM0). PWM outputs can be configured to use Port 2.6 and Port 2.7 or Port 3.4 and Port 3.3. Input for External Clock Signal (EXTCLK). This pin function must be enabled via the CFG842 register.
25	P3.5/T1/$\overline{\text{CONVST}}$	I/O	Input/Output Port 3 (P3.5). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-up resistors. Timer/Counter 1 Input (T1). Active Low Convert Start Logic Input for the ADC Block when the External Convert Start Function is Enabled ($\overline{\text{CONVST}}$). A low to high transition on this input puts the track-and-hold into hold mode and starts the conversion.
26	P3.6/$\overline{\text{WR}}$	I/O	Input/Output Port 3 (P3.6). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-up resistors. Write Control Signal, Logic Output ($\overline{\text{WR}}$). Latches the data byte from Port 0 into the external data memory.
27	P3.7/$\overline{\text{RD}}$	I/O	Input/Output Port 3 (P3.7). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-up resistors. Read Control Signal, Logic Output ($\overline{\text{RD}}$). Enables the external data memory to Port 0.
28	SCLOCK	I/O	Serial Clock Pin for I^2C-Compatible Clock or for SPI Serial Interface Clock.
29	SDATA/MOSI	I/O	User Selectable, I^2C Compatible, or SPI Data Input/Output Pin (SDATA). SPI Master Output/Slave Input Data I/O Pin for SPI Interface (MOSI).
30	P2.0/A8/A16	I/O	Input/Output Port 2 (P2.0). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A8). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A16). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
31	P2.1/A9/A17	I/O	Input/Output Port 2 (P2.1). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A9). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A17). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
32	P2.2/A10/A18	I/O	Input/Output Port 2 (P2.2). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A10). Port 2 emits the middle address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A18). Port 2 emits the high-order address byte during accesses to the external 24-bit external data memory space.

SPECIAL FUNCTION REGISTER BANKS

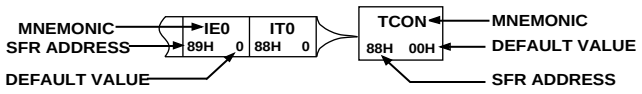
All registers except the program counter and the four general-purpose register banks reside in the special function register (SFR) area. The SFR registers include control, configuration, and data registers, which provide an interface between the CPU and other on-chip peripherals. Figure 27 shows a full SFR memory map and SFR contents on reset. Unoccupied SFR locations are shown dark-shaded in the figure (NOT USED). Unoccupied locations in the SFR address space are not implemented, that is, no register exists at this location. If an unoccupied location is read, an unspecified value is returned. SFR locations reserved for on-chip testing are shown lighter shaded (RESERVED) and should not be accessed by user software. Sixteen of the SFR locations are also bit addressable and denoted by 1 in Figure 27, that is, the bit addressable SFRs are those whose address ends in 0H or 8H.

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON[1] F8H 04H
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B[1] F0H 00H
I2CSI/MDO EFH 0	I2CGC/MDE EEH 0	I2C1O1MCO EDH 0	I2C1O0/MDI ECH 0	I2CM EBH 0	I2CRS EAH 0	I2CTX E9H 0	I2CI E8H 0	BITS	I2CCON[1] E8H 00H
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC[1] E0H 00H
ADCI DFH 0	DMA DEH 0	CCONV DDH 0	SCONV DCH 0	CS3 DBH 0	CS2 DAH 0	CS1 D9H 0	CS0 D8H 0	BITS	ADCCON2[1] D8H 00H
CY D7H 0	AC D6H 0	F0 D5H 0	RS1 D4H 0	RS0 D3H 0	OV D2H 0	FI D1H 0	P D0H 0	BITS	PSW[1] D0H 00H
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS	T2CON[1] C8H 00H
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 1	WDIR C3H 0	WDS C2H 0	WDE C1H 0	WDWR C0H 0	BITS	WDCON[1] C0H 10H
PSI BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS	IP[1] B8H 00H
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS	P3[1] B0H FFH
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS	IE[1] A8H 00H
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2[1] A0H FFH
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	TI 99H 0	RI 98H 0	BITS	SCON[1] 98H 00H
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1[1, 2] 90H FFH
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS	TCON[1] 88H 00H
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0[1] 80H FFH

DAC0L F9H 00H	DAC0H FAH 00H	DAC1L FBH 00H	DAC1H FCH 00H	DACCON FDH 04H	RESERVED	RESERVED	
ADCOFSL[3] F1H 00H	ADCOFSH[3] F2H 20H	ADCGAINL[3] F3H 00H	ADCGAINH[3] F4H 00H	ADCCON3 F5H 00H	RESERVED	SPIDAT F7H 00H	
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCON1 EFH 40H	
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	
ADCDATAL D9H 00H	ADCDATAH DAH 00H	RESERVED	RESERVED	RESERVED	RESERVED	PSMCON DFH DEH	
RESERVED	DMAL D2H 00H	DMAH D3H 00H	DMAP D4H 00H	RESERVED	RESERVED	PLLCON D7H 53H	
RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED	
RESERVED	CHIPID C2H XXH	RESERVED	RESERVED	RESERVED	EDARL C6H 00H	EDARH C7H 00H	
ECON B9H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H	
PWM0L B1H 00H	PWM0H B2H 00H	PWM1L B3H 00H	PWM1H B4H 00H	NOT USED	NOT USED	SPH B7H 00H	
IEIP2 A9H A0H	RESERVED	RESERVED	RESERVED	RESERVED	PWMCON AEH 00H	CFG841/ CFG842 AFH 00H	
TIMECON A1H 00H	HTHSEC A2H 00H	SEC A3H 00H	MIN A4H 00H	HOUR A5H 00H	INTVAL A6H 00H	DPCON A7H 00H	
SBUF 99H 00H	I2CDAT 9AH 00H	I2CADD 9BH 55H	NOT USED	T3FD 9DH 00H	T3CON 9EH 00H	NOT USED	
I2CADD1 91H 7FH	I2CADD2 92H 7FH	I2CADD3 93H 7FH	NOT USED	NOT USED	NOT USED	NOT USED	
TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED	
SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H	

SFR MAP KEY:

THESE BITS ARE CONTAINED IN THIS BYTE.



NOTES

[1]SFRs WHOSE ADDRESS ENDS IN 0H OR 8H ARE BIT ADDRESSABLE.

[2]THE PRIMARY FUNCTION OF PORT1 IS AS AN ANALOG INPUT PORT; THEREFORE, TO ENABLE THE DIGITAL SECONDARY FUNCTIONS ON THESE PORT PINS, WRITE A 0 TO THE CORRESPONDING PORT 1 SFR BIT.

[3]CALIBRATION COEFFICIENTS ARE PRECONFIGURED ON POWER-UP TO FACTORY CALIBRATED VALUES.

03260-0-025

Figure 27. Special Function Register Locations and Reset Values

ADCCON3—(ADC Control SFR 3)

The ADCCON3 register controls the operation of various calibration modes and also indicates the ADC busy status.

SFR Address	F5H
SFR Power-On Default	00H
Bit Addressable	No

Table 10. ADCCON3 SFR Bit Designations

Bit No.	Name	Description
7	BUSY	ADC Busy Status Bit. A read-only status bit that is set during a valid ADC conversion or during a calibration cycle. Busy is automatically cleared by the core at the end of conversion or calibration.
6	RSVD	Reserved. This bit should always be written as 0.
5	AVGS1	Number of Average Selection Bits.
4	AVGS0	This bit selects the number of ADC readings that are averaged during a calibration cycle. AVGS1 AVGS0 Number of Averages 0 0 15 0 1 1 1 0 31 1 1 63
3	RSVD	Reserved. This bit should always be written as 0.
2	RSVD	This bit should always be written as 1 by the user when performing calibration.
1	TYPICAL	Calibration Type Select Bit. This bit selects between offset (zero-scale) and gain (full-scale) calibration. Set to 0 for offset calibration. Set to 1 for gain calibration.
0	SCAL	Start Calibration Cycle Bit. When set, this bit starts the selected calibration cycle. It is automatically cleared when the calibration cycle is completed.

If using the temperature sensor as the ADC input, the ADC should be configured to use an ADCCLK of MCLK/32 and four acquisition clocks.

Increasing the conversion time on the temperature monitor channel improves the accuracy of the reading. To further improve the accuracy, an external reference with low temperature drift should also be used.

ADC DMA Mode

The on-chip ADC has been designed to run at a maximum conversion speed of 2.38 µs (420 kHz sampling rate). When converting at this rate, the ADuC841/ADuC842/ADuC843 MicroConverter® has 2 µs to read the ADC result and to store the result in memory for further postprocessing; otherwise the next ADC sample could be lost. In an interrupt driven routine, the MicroConverter would also have to jump to the ADC interrupt service routine, which also increases the time required to store the ADC results. In applications where the parts cannot sustain the interrupt rate, an ADC DMA mode is provided.

To enable DMA mode, Bit 6 in ADCCON2 (DMA) must be set, which allows the ADC results to be written directly to a 16 MByte external static memory SRAM (mapped into data memory space) without any interaction from the core of the part. This mode allows the part to capture a contiguous sample stream at full ADC update rates (420 kHz).

Typical DMA Mode Configuration Example

Setting the parts to DMA mode consists of the following steps:

1. The ADC must be powered down. This is done by ensuring that MD1 and MD0 are both set to 0 in ADCCON1.
2. The DMA address pointer must be set to the start address of where the ADC results are to be written. This is done by writing to the DMA mode address pointers DMAL, DMAH, and DMAP. DMAL must be written to first, followed by DMAH, and then by DMAP.
3. The external memory must be preconfigured. This consists of writing the required ADC channel IDs into the top four bits of every second memory location in the external SRAM, starting at the first address specified by the DMA address pointer. Because the ADC DMA mode operates independently from the ADuC841/ADuC842/ADuC843 core, it is necessary to provide it with a stop command. This is done by duplicating the last channel ID to be converted followed by 1111 into the next channel selection field. A typical preconfiguration of external memory is shown in Figure 34.

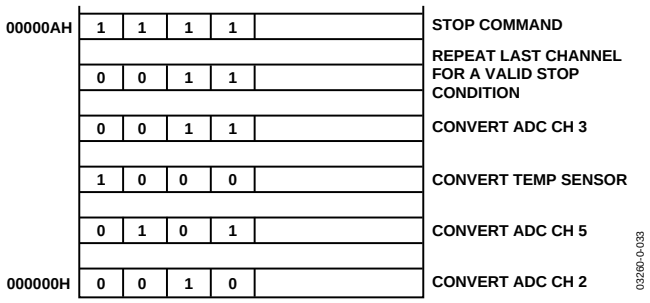


Figure 34. Typical DMA External Memory Preconfiguration

4. The DMA is initiated by writing to the ADC SFRs in the following sequence:
   a. ADCCON2 is written to enable the DMA mode, that is, MOV ADCCON2, #40H; DMA mode enabled.
   b. ADCCON1 is written to configure the conversion time and power-up of the ADC. It can also enable Timer 2 driven conversions or external triggered conversions if required.
   c. ADC conversions are initiated. This is done by starting single conversions, starting Timer 2, running for Timer 2 conversions, or receiving an external trigger.

When the DMA conversions are complete, the ADC interrupt bit, ADCI, is set by hardware, and the external SRAM contains the new ADC conversion results as shown in Figure 35. Note that no result is written to the last two memory locations.

When the DMA mode logic is active, it takes the responsibility of storing the ADC results away from both the user and the core logic of the part. As the DMA interface writes the results of the ADC conversions to external memory, it takes over the external memory interface from the core. Thus, any core instructions that access the external memory while DMA mode is enabled does not get access to the external memory. The core executes the instructions, and they take the same time to execute, but they cannot access the external memory.

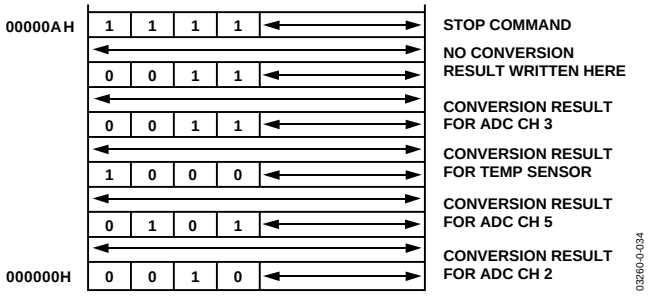


Figure 35. Typical External Memory Configuration Post ADC DMA Operation

Initiating the Calibration in Code

When calibrating the ADC using ADCCON1, the ADC must be set up into the configuration in which it is used. The ADCCON3 register can then be used to set up the device and to calibrate the ADC offset and gain.

```
MOV ADCCON1,#08CH ;   ADC on; ADCCLK set
                      ;to divide by 32,4
                      ;acquisition clock
```

To calibrate device offset:

```
MOV ADCCON2,#0BH      ;select internal AGND
MOV ADCCON3,#25H      ;select offset calibration,
                      ;31 averages per bit,
                      ;offset calibration
```

To calibrate device gain:

```
MOV ADCCON2,#0CH      ;select internal VREF
MOV ADCCON3,#27H      ;select offset calibration,
                      ;31 averages per bit,
                      ;offset calibration
```

To calibrate system offset, connect system AGND to an ADC channel input (0).

```
MOV ADCCON2,#00H      ;select external AGND
MOV ADCCON3,#25H      ;select offset calibration,
                      ;31 averages per bit
```

To calibrate system gain, connect system V_{REF} to an ADC channel input (1).

```
MOV ADCCON2,#01H      ;select external VREF
MOV ADCCON3,#27H      ;select offset calibration,
                      ;31 averages per bit,
                      ;offset calibration
```

The calibration cycle time T_{CAL} is calculated by the following equation:

$$T_{CAL} = 14 \times ADCCLK \times NUMAV \times (16 + T_{ACQ})$$

For an ADCCLK/FCORE divide ratio of 32, T_{ACQ} = 4 ADCCLK, and NUMAV = 15, the calibration cycle time is

$$T_{CAL} = 14 \times (1/524288) \times 15 \times (16 + 4)$$

$$T_{CAL} = 8\ ms$$

In a calibration cycle, the ADC busy flag (Bit 7), instead of framing an individual ADC conversion as in normal mode, goes high at the start of calibration and returns to zero only at the end of the calibration cycle. It can therefore be monitored in code to indicate when the calibration cycle is completed. The following code can be used to monitor the BUSY signal during a calibration cycle:

```
WAIT:
MOV A, ADCCON3          ;move ADCCON3 to A
JB ACC.7, WAIT          ;If Bit 7 is set jump to
                        WAIT else continue
```

NONVOLATILE FLASH/EE MEMORY

The ADuC841/ADuC842/ADuC843 incorporate Flash/EE memory technology on-chip to provide the user with nonvolatile, in-circuit, reprogrammable code and data memory space. Flash/EE memory is a relatively recent type of nonvolatile memory technology, which is based on a single transistor cell architecture. Flash/EE memory combines the flexible in-circuit reprogrammable features of EEPROM with the space efficient/density features of EPROM as shown in Figure 37.

Because Flash/EE technology is based on a single transistor cell architecture, a flash memory array, such as EPROM, can be implemented to achieve the space efficiencies or memory densities required by a given design. Like EEPROM, flash memory can be programmed in-system at a byte level; it must first be erased, the erase being performed in page blocks. Thus, flash memory is often and more correctly referred to as Flash/EE memory.

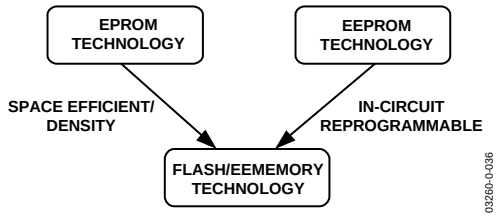


Figure 37. Flash/EE Memory Development

Overall, Flash/EE memory represents a step closer to the ideal memory device that includes nonvolatility, in-circuit programmability, high density, and low cost. Incorporated in the parts, Flash/EE memory technology allows the user to update program code space in-circuit, without the need to replace one-time programmable (OTP) devices at remote operating nodes.

Flash/EE Memory and the ADuC841/ADuC842/ADuC843

The parts provide two arrays of Flash/EE memory for user applications. Up to 62 kBytes of Flash/EE program space are provided on-chip to facilitate code execution without any external discrete ROM device requirements. The program memory can be programmed in-circuit by using the serial download mode provided, by using conventional third party memory programmers, or via a user defined protocol that can configure it as data if required.

Note that the following sections use the 62 kByte program space as an example when referring to ULOAD mode. For the other memory models (32 kByte and 8 kByte), the ULOAD space moves to the top 8 kBytes of the on-chip program memory, that is, for 32 kBytes, the ULOAD space is from 24 kBytes to 32 kBytes, the kernel still resides in a protected space from 60 kBytes to 62 kBytes. There is no ULOAD space present on the 8 kBtye part.

ON-CHIP PLL

The ADuC842 and ADuC843 are intended for use with a 32.768 kHz watch crystal. A PLL locks onto a multiple (512) of this to provide a stable 16.78 MHz clock for the system. The ADuC841 operates directly from an external crystal. The core can operate at this frequency or at binary submultiples of it to allow power saving in cases where maximum core performance is not required. The default core clock is the PLL clock divided by 8 or 2.097152 MHz. The ADC clocks are also derived from the PLL clock, with the modulator rate being the same as the crystal oscillator frequency. The preceding choice of frequencies ensures that the modulators and the core are synchronous, regardless of the core clock rate. The PLL control register is PLLCON.

At 5 V the core clock can be set to a maximum of 16.78 MHz, while at 3 V the maximum core clock setting is 8.38 MHz. The CD bits should not be set to 0 on a 3 V part.

Note that on the ADuC841, changing the CD bits in PLLCON causes the core speed to change. The core speed is crystal freq/2^{CD}. The other bits in PLLCON are reserved in the case of the ADuC841 and should be written with 0.

PLLCON PLL	**Control Register**
SFR Address	D7H
Power-On Default	53H
Bit Addressable	No

Table 17. PLLCON SFR Bit Designations

Bit No.	Name	Description			
7	OSC_PD	Oscillator Power-Down Bit. Set by the user to halt the 32 kHz oscillator in power-down mode. Cleared by the user to enable the 32 kHz oscillator in power-down mode. This feature allows the TIC to continue counting even in power-down mode.			
6	LOCK	PLL Lock Bit. This is a read-only bit. Set automatically at power-on to indicate that the PLL loop is correctly tracking the crystal clock. If the external crystal subsequently becomes disconnected, the PLL rails. Cleared automatically at power-on to indicate that the PLL is not correctly tracking the crystal clock. This may be due to the absence of a crystal clock or an external crystal at power-on. In this mode, the PLL output can be 16.78 MHz ±20%.			
5	----	Reserved. Should be written with 0.			
4	----	Reserved. Should be written with 0.			
3	FINT	Fast Interrupt Response Bit. Set by the user enabling the response to any interrupt to be executed at the fastest core clock frequency, regardless of the configuration of the CD2–0 bits (see below). Once user code has returned from an interrupt, the core resumes code execution at the core clock selected by the CD2–0 bits. Cleared by the user to disable the fast interrupt response feature.			
2	CD2	CPU (Core Clock) Divider Bits.			
1	CD1	This number determines the frequency at which the microcontroller core operates.			
0	CD0	CD2	CD1	CD0	Core Clock Frequency (MHz)
		0	0	0	16.777216
		0	0	1	8.388608
		0	1	0	4.194304
		0	1	1	2.097152 (Default Core Clock Frequency)
		1	0	0	1.048576
		1	0	1	0.524288
		1	1	0	0.262144
		1	1	1	0.131072

PWM Modes of Operation

Mode 0: PWM Disabled

The PWM is disabled allowing P2.6 and P2.7 to be used as normal.

Mode 1: Single Variable Resolution PWM

In Mode 1, both the pulse length and the cycle time (period) are programmable in user code, allowing the resolution of the PWM to be variable.

PWM1H/L sets the period of the output waveform. Reducing PWM1H/L reduces the resolution of the PWM output but increases the maximum output rate of the PWM. For example, setting PWM1H/L to 65536 gives a 16-bit PWM with a maximum output rate of 266 Hz (16.777 MHz/65536). Setting PWM1H/L to 4096 gives a 12-bit PWM with a maximum output rate of 4096 Hz (16.777 MHz/4096).

PWM0H/L sets the duty cycle of the PWM output waveform, as shown in Figure 48.

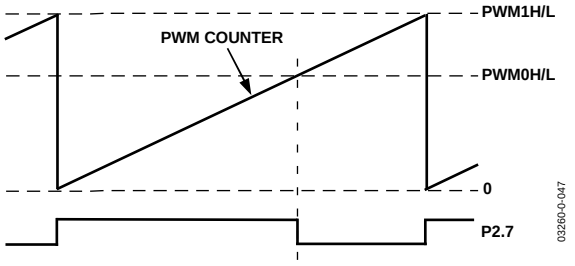


Figure 48. PWM in Mode 1

Mode 2: Twin 8-Bit PWM

In Mode 2, the duty cycle of the PWM outputs and the resolution of the PWM outputs are both programmable. The maximum resolution of the PWM output is 8 bits.

PWM1L sets the period for both PWM outputs. Typically, this is set to 255 (FFH) to give an 8-bit PWM, although it is possible to reduce this as necessary. A value of 100 could be loaded here to give a percentage PWM, that is, the PWM is accurate to 1%.

The outputs of the PWM at P2.6 and P2.7 are shown in Figure 49. As can be seen, the output of PWM0 (P2.6) goes low when the PWM counter equals PWM0L. The output of PWM1 (P2.7) goes high when the PWM counter equals PWM1H and goes low again when the PWM counter equals PWM0H. Setting PWM1H to 0 ensures that both PWM outputs start simultaneously.

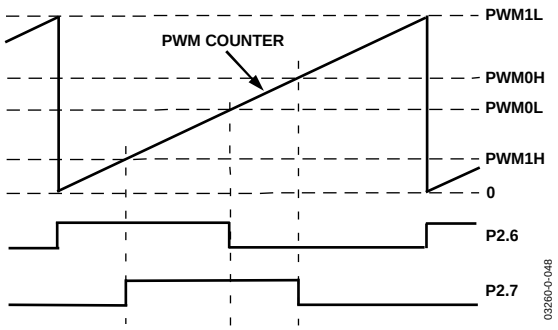


Figure 49. PWM Mode 2

Mode 3: Twin 16-Bit PWM

In Mode 3, the PWM counter is fixed to count from 0 to 65536, giving a fixed 16-bit PWM. Operating from the 16.777 MHz core clock results in a PWM output rate of 256 Hz. The duty cycle of the PWM outputs at P2.6 and P2.7 is independently programmable.

As shown in Figure 50, while the PWM counter is less than PWM0H/L, the output of PWM0 (P2.6) is high. Once the PWM counter equals PWM0H/L, PWM0 (P2.6) goes low and remains low until the PWM counter rolls over.

Similarly, while the PWM counter is less than PWM1H/L, the output of PWM1 (P2.7) is high. Once the PWM counter equals PWM1H/L, PWM1 (P2.7) goes low and remains low until the PWM counter rolls over.

In this mode, both PWM outputs are synchronized, that is, once the PWM counter rolls over to 0, both PWM0 (P2.6) and PWM1 go high.

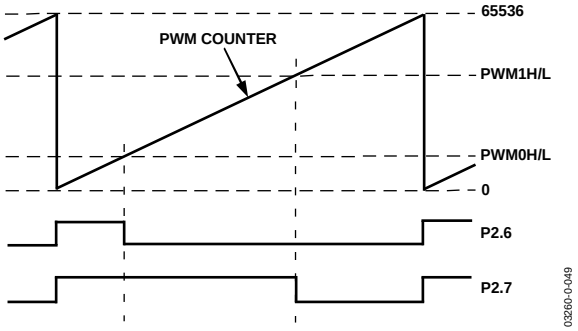


Figure 50. PWM Mode 3

Using the SPI Interface

Depending on the configuration of the bits in the SPICON SFR shown in Table 19, the ADuC841/ADuC842/ADuC843 SPI interface transmits or receives data in a number of possible modes. Figure 54 shows all possible SPI configurations for the parts, and the timing relationships and synchronization between the signals involved. Also shown in this figure is the SPI interrupt bit (ISPI) and how it is triggered at the end of each byte-wide communication.

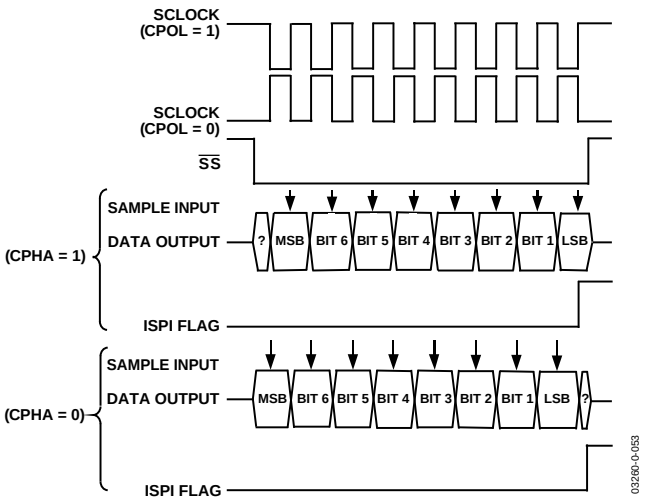


Figure 54. SPI Timing, All Modes

SPI Interface—Master Mode

In master mode, the SCLOCK pin is always an output and generates a burst of eight clocks whenever user code writes to the SPIDAT register. The SCLOCK bit rate is determined by SPR0 and SPR1 in SPICON. Also note that the $\overline{SS}$ pin is not used in master mode. If the parts need to assert the $\overline{SS}$ pin on an external slave device, a port digital output pin should be used.

In master mode, a byte transmission or reception is initiated by a write to SPIDAT. Eight clock periods are generated via the SCLOCK pin and the SPIDAT byte being transmitted via MOSI. With each SCLOCK period, a data bit is also sampled via MISO. After eight clocks, the transmitted byte is completely transmitted, and the input byte waits in the input shift register. The ISPI flag is set automatically, and an interrupt occurs if enabled. The value in the shift register is latched into SPIDAT.

SPI Interface—Slave Mode

In slave mode, SCLOCK is an input. The $\overline{SS}$ pin must also be driven low externally during the byte communication. Transmission is also initiated by a write to SPIDAT. In slave mode, a data bit is transmitted via MISO, and a data bit is received via MOSI through each input SCLOCK period. After eight clocks, the transmitted byte is completely transmitted, and the input byte waits in the input shift register. The ISPI flag is set automatically, and an interrupt occurs if enabled. The value in the shift register is latched into SPIDAT only when the transmission/reception of a byte has been completed. The end of transmission occurs after the eighth clock has been received if CPHA = 1, or when $\overline{SS}$ returns high if CPHA = 0.

DUAL DATA POINTER

The ADuC841/ADuC842/ADuC843 incorporate two data pointers. The second data pointer is a shadow data pointer and is selected via the data pointer control SFR (DPCON). DPCON also includes some useful features such as automatic hardware post-increment and post-decrement as well as automatic data pointer toggle. DPCON is described in Table 22.

DPCON	Data Pointer Control SFR
SFR Address	A7H
Power-On Default	00H
Bit Addressable	No

Table 22. DPCON SFR Bit Designations

Bit No.	Name	Description
7	----	Reserved.
6	DPT	Data Pointer Automatic Toggle Enable. Cleared by the user to disable autoswapping of the DPTR. Set in user software to enable automatic toggling of the DPTR after each each MOVX or MOVC instruction.
5	DP1m1	Shadow Data Pointer Mode.
4	DP1m0	These two bits enable extra modes of the shadow data pointer's operation, allowing for more compact and more efficient code size and execution. m1 m0 Behavior of the shadow data pointer. 0 0 8052 behavior. 0 1 DPTR is post-incremented after a MOVX or a MOVC instruction. 1 0 DPTR is post-decremented after a MOVX or MOVC instruction. 1 1 DPTR LSB is toggled after a MOVX or MOVC instruction. (This instruction can be useful for moving 8-bit blocks to/from 16-bit devices.)
3	DP0m1	Main Data Pointer Mode.
2	DP0m0	These two bits enable extra modes of the main data pointer operation, allowing for more compact and more efficient code size and execution. m1 m0 Behavior of the main data pointer. 0 0 8052 behavior. 0 1 DPTR is post-incremented after a MOVX or a MOVC instruction. 1 0 DPTR is post-decremented after a MOVX or MOVC instruction. 1 1 DPTR LSB is toggled after a MOVX or MOVC instruction. (This instruction can be useful for moving 8-bit blocks to/from 16-bit devices.)
1	----	This bit is not implemented to allow the INC DPCON instruction toggle the data pointer without incrementing the rest of the SFR.
0	DPSEL	Data Pointer Select. Cleared by the user to select the main data pointer. This means that the contents of this 24-bit register are placed into the three SFRs: DPL, DPH, and DPP. Set by the user to select the shadow data pointer. This means that the contents of a separate 24-bit register appears in the three SFRs: DPL, DPH, and DPP.

Note 1: This is the only place where the main and shadow data pointers are distinguished. Everywhere else in this data sheet wherever the DPTR is mentioned, operation on the active DPTR is implied.

Note 2: Only MOVC/MOVX @DPTR instructions are relevant above. MOVC/MOVX PC/@Ri instructions do not cause the DPTR to automatically post increment/decrement, and so on.

To illustrate the operation of DPCON, the following code copies 256 bytes of code memory at address D000H into XRAM starting from Address 0000H.

```
    MOV DPTR,#0          ; Main DPTR = 0
    MOV DPCON,#55H       ; Select shadow DPTR
                         ; DPTR1 increment mode,
                         ; DPTR0 increment mode
                         ; DPTR auto toggling ON
    MOV DPTR,#0D000H     ; Shadow DPTR = D000H
MOVELOOP:
    CLR A
    MOVC A,@A+DPTR       ; Get data
                         ; Post Inc DPTR
                         ; Swap to Main DPTR (Data)
    MOVX @DPTR,A         ; Put ACC in XRAM
                         ; Increment main DPTR
                         ; Swap Shadow DPTR (Code)
    MOV A, DPL
    JNZ MOVELOOP
```

MOSI is shared with P3.3 and, as such, has the same configuration as the one shown in Figure 61.

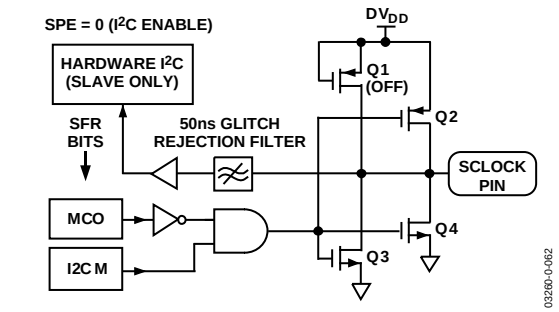


Figure 63. SCLOCK Pin I/O Functional Equivalent in I²C Mode

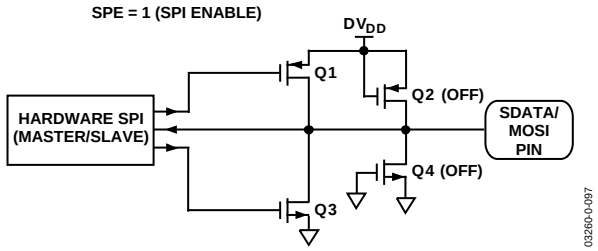


Figure 64. SDATA/MOSI Pin I/O Functional Equivalent in SPI Mode

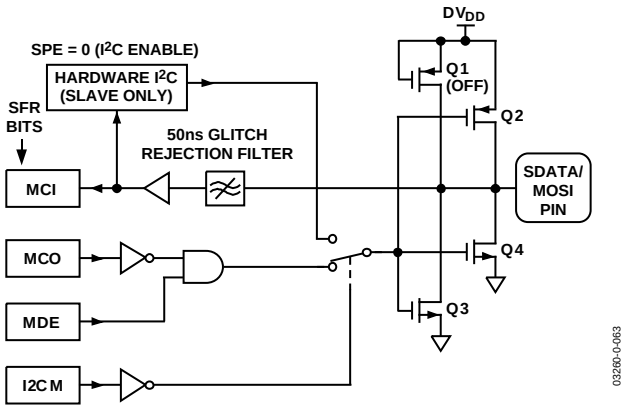


Figure 65. SDATA/MOSI Pin I/O Functional Equivalent in I²C Mode

Read-Modify-Write Instructions

Some 8051 instructions that read a port read the latch while others read the pin. The instructions that read the latch rather than the pins are the ones that read a value, possibly change it, and then rewrite it to the latch. These are called read-modify-write instructions, which are listed below. When the destination operand is a port or a port bit, these instructions read the latch rather than the pin.

Table 28. Read-Write-Modify Instructions

Instruction	Description
ANL	Logical AND, for example, ANL P1, A
ORL	(Logical OR, for example, ORL P2, A
XRL	(Logical EX-OR, for example, XRL P3, A
JBC	Jump if Bit = 1 and clear bit, for example, JBC P1.1, LABEL
CPL	Complement bit, for example, CPL P3.0
INC	Increment, for example, INC P2
DEC	Decrement, for example, DEC P2
DJNZ	Decrement and Jump if Not Zero, for example, DJNZ P3, LABEL
MOV PX.Y, C[1]	Move Carry to Bit Y of Port X
CLR PX.Y[1]	Clear Bit Y of Port X
SETB PX.Y[1]	Set Bit Y of Port X

[1] These instructions read the port byte (all 8 bits), modify the addressed bit, and then write the new byte back to the latch.

Read-modify-write instructions are directed to the latch rather than to the pin to avoid a possible misinterpretation of the voltage level of a pin. For example, a port pin might be used to drive the base of a transistor. When 1 is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it reads the base voltage of the transistor and interprets it as a Logic 0. Reading the latch rather than the pin returns the correct value of 1.

TIMER/COUNTER 0 AND 1 OPERATING MODES

The following sections describe the operating modes for Timer/Counters 0 and 1. Unless otherwise noted, assume that these modes of operation are the same for both Timer 0 and Timer 1.

Mode 0 (13-Bit Timer/Counter)

Mode 0 configures an 8-bit timer/counter. Figure 66 shows Mode 0 operation. Note that the divide-by-12 prescaler is not present on the single-cycle core.

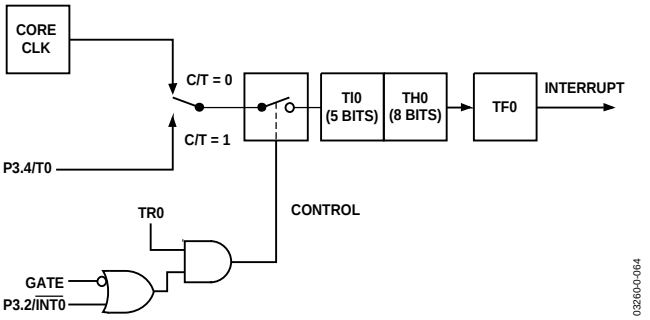


Figure 66. Timer/Counter 0, Mode 0

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the timer overflow flag, TF0. TF0 can then be used to request an interrupt. The counted input is enabled to the timer when TR0 = 1 and either Gate = 0 or $\overline{INT0}$ = 1. Setting Gate = 1 allows the timer to be controlled by external input $\overline{INT0}$ to facilitate pulse-width measurements. TR0 is a control bit in the special function register TCON; Gate is in TMOD. The 13-bit register consists of all 8 bits of TH0 and the lower five bits of TL0. The upper 3 bits of TL0 are indeterminate and should be ignored. Setting the run flag (TR0) does not clear the registers.

Mode 1 (16-Bit Timer/Counter)

Mode 1 is the same as Mode 0, except that the Mode 1 timer register is running with all 16 bits. Mode 1 is shown in Figure 67.

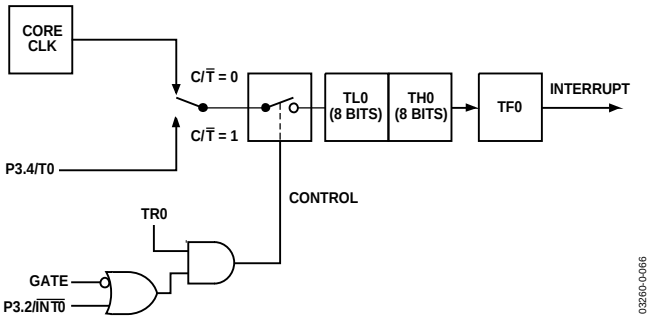


Figure 67. Timer/Counter 0, Mode 1

Mode 2 (8-Bit Timer/Counter with Autoreload)

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reload, as shown in Figure 68. Overflow from TL0 not only sets TF0, but also reloads TL0 with the contents of TH0, which is preset by software. The reload leaves TH0 unchanged.

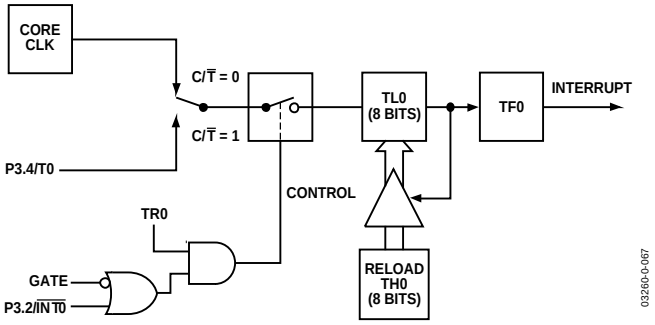


Figure 68. Timer/Counter 0, Mode 2

Mode 3 (Two 8-Bit Timer/Counters)

Mode 3 has different effects on Timer 0 and Timer 1. Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. This configuration is shown in Figure 69. TL0 uses the Timer 0 control bits: C/$\overline{T}$, Gate, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the Timer 1 interrupt. Mode 3 is provided for applications requiring an extra 8-bit timer or counter.

When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3, or it can still be used by the serial interface as a baud rate generator. In fact, it can be used in any application not requiring an interrupt from Timer 1 itself.

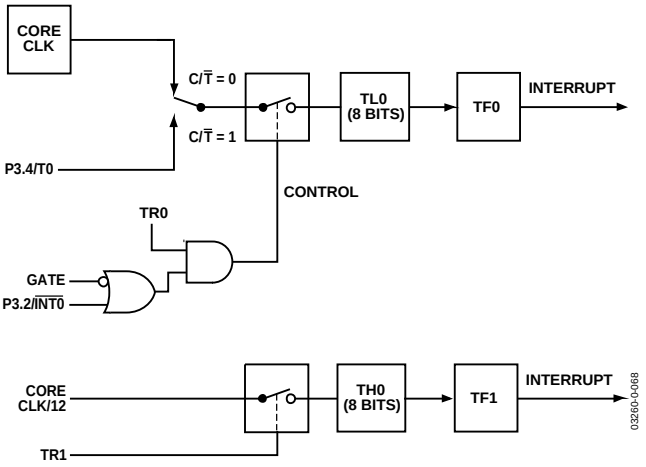


Figure 69. Timer/Counter 0, Mode 3

UART SERIAL INTERFACE

The serial port is full-duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can begin receiving a second byte before a previously received byte has been read from the receive register. However, if the first byte still has not been read by the time reception of the second byte is complete, the first byte is lost. The physical interface to the serial data network is via Pins RxD(P3.0) and TxD(P3.1), while the SFR interface to the UART is comprised of SBUF and SCON, as described below.

SBUF

Both the serial port receive and transmit registers are accessed through the SBUF SFR (SFR address = 99H). Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

SCON UART	Serial Port Control Register
SFR Address	98H
Power-On Default	00H
Bit Addressable	Yes

Table 33. SCON SFR Bit Designations

Bit No.	Name	Description		
7	SM0	UART Serial Mode Select Bits.		
6	SM1	These bits select the serial port operating mode as follows:		
		SM0	SM1	Selected Operating Mode.
		0	0	Mode 0: Shift Register, fixed baud rate (Core_Clk/2).
		0	1	Mode 1: 8-bit UART, variable baud rate.
		1	0	Mode 2: 9-bit UART, fixed baud rate (Core_Clk/32) or (Core_Clk/16).
		1	1	Mode 3: 9-bit UART, variable baud rate.
5	SM2	Multiprocessor Communication Enable Bit. Enables multiprocessor communication in Modes 2 and 3. In Mode 0, SM2 must be cleared. In Mode 1, if SM2 is set, RI is not activated if a valid stop bit was not received. If SM2 is cleared, RI is set as soon as the byte of data has been received. In Modes 2 or 3, if SM2 is set, RI is not activated if the received 9th data bit in RB8 is 0. If SM2 is cleared, RI is set as soon as the byte of data has been received.		
4	REN	Serial Port Receive Enable Bit. Set by user software to enable serial port reception. Cleared by user software to disable serial port reception.		
3	TB8	Serial Port Transmit (Bit 9). The data loaded into TB8 is the 9th data bit transmitted in Modes 2 and 3.		
2	RB8	Serial Port Receiver Bit 9. The 9th data bit received in Modes 2 and 3 is latched into RB8. For Mode 1, the stop bit is latched into RB8.		
1	TI	Serial Port Transmit Interrupt Flag. Set by hardware at the end of the 8th bit in Mode 0, or at the beginning of the stop bit in Modes 1, 2, and 3. TI must be cleared by user software.		
0	RI	Serial Port Receive Interrupt Flag. Set by hardware at the end of the 8th bit in Mode 0, or halfway through the stop bit in Modes 1, 2, and 3. RI must be cleared by software.		

Table 35. Commonly Used Baud Rates Using Timer 3 with the 16.777216 MHz PLL Clock

Ideal Baud	CD	DIV	T3CON	T3FD	% Error
230400	0	2	82H	09H	0.25
115200	0	3	83H	09H	0.25
115200	1	2	82H	09H	0.25
115200	2	1	81H	09H	0.25
57600	0	4	84H	09H	0.25
57600	1	3	83H	09H	0.25
57600	2	2	82H	09H	0.25
57600	3	1	81H	09H	0.25
38400	0	4	84H	2DH	0.2
38400	1	3	83H	2DH	0.2
38400	2	2	82H	2DH	0.2
38400	3	1	81H	2DH	0.2
19200	0	5	85H	2DH	0.2
19200	1	4	84H	2DH	0.2
19200	2	3	83H	2DH	0.2
19200	3	2	82H	2DH	0.2
19200	4	1	81H	2DH	0.2
9600	0	6	86H	2DH	0.2
9600	1	5	85H	2DH	0.2
9600	2	4	84H	2DH	0.2
9600	3	3	83H	2DH	0.2
9600	4	2	82H	2DH	0.2
9600	5	1	81H	2DH	0.2

IEIP2	Secondary Interrupt Enable Register
SFR Address	A9H
Power-On Default	A0H
Bit Addressable	No

Table 38. IEIP2 SFR Bit Designations

Bit No.	Name	Description
7	----	Reserved.
6	PTI	Priority for time interval interrupt.
5	PPSM	Priority for power supply monitor interrupt.
4	PSI	Priority for SPI/I²C interrupt.
3	----	This bit must contain zero.
2	ETI	Set by the user to enable, or cleared to disable time interval counter interrupts.
1	EPSMI	Set by the user to enable, or cleared to disable power supply monitor interrupts.
0	ESI	Set by the user to enable, or cleared to disable SPI or I²C serial port interrupts.

Interrupt Priority

The interrupt enable registers are written by the user to enable individual interrupt sources, while the interrupt priority registers allow the user to select one of two priority levels for each interrupt. An interrupt of a high priority may interrupt the service routine of a low priority interrupt, and if two interrupts of different priority occur at the same time, the higher level interrupt is serviced first. An interrupt cannot be interrupted by another interrupt of the same priority level. If two interrupts of the same priority level occur simultaneously, a polling sequence is observed as shown in Table 39.

Table 39. Priority within an Interrupt Level

Source	Priority	Description
PSMI	1 (Highest)	Power Supply Monitor Interrupt.
WDS	2	Watchdog Timer Interrupt.
IE0	2	External Interrupt 0.
ADCI	3	ADC Interrupt.
TF0	4	Timer/Counter 0 Interrupt.
IE1	5	External Interrupt 1.
TF1	6	Timer/Counter 1 Interrupt.
ISPI/I2CI	7	SPI Interrupt/I²C Interrupt.
RI + TI	8	Serial Interrupt.
TF2 + EXF2	9	Timer/Counter 2 Interrupt.
TII	11(Lowest)	Time Interval Counter Interrupt.

Interrupt Vectors

When an interrupt occurs, the program counter is pushed onto the stack, and the corresponding interrupt vector address is loaded into the program counter. The interrupt vector addresses are shown in Table 40.

Table 40. Interrupt Vector Addresses

Source	Vector Address
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI + TI	0023H
TF2 + EXF2	002BH
ADCI	0033H
ISPI/I2CI	003BH
PSMI	0043H
TII	0053H
WDS	005BH

If access to more than 64 kBytes of RAM is desired, a feature unique to the ADuC841/ADuC842/ADuC843 allows addressing up to 16 MBytes of external RAM simply by adding an additional latch as illustrated in Figure 79.

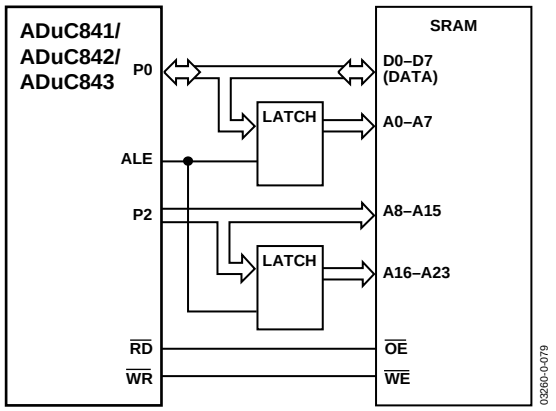


Figure 79. External Data Memory Interface (16 MBytes Address Space)

In either implementation, Port 0 (P0) serves as a multiplexed address/data bus. It emits the low byte of the data pointer (DPL) as an address, which is latched by a pulse of ALE prior to data being placed on the bus by the ADuC841/ADuC842/ADuC843 (write operation) or by the SRAM (read operation). Port 2 (P2) provides the data pointer page byte (DPP) to be latched by ALE, followed by the data pointer high byte (DPH). If no latch is connected to P2, DPP is ignored by the SRAM, and the 8051 standard of 64 kBytes external data memory access is maintained.

Power Supplies

The operational power supply voltage of the parts depends on whether the part is the 3 V version or the 5 V version. The specifications are given for power supplies within 2.7 V to 3.6 V or ±5% of the nominal 5 V level.

Note that Figure 80 and Figure 81 refer to the PQFP package. For the CSP package, connect the extra DV_{DD}, DGND, AV_{DD}, and AGND in the same manner. Also, the paddle on the bottom of the package should be soldered to a metal plate to provide mechanical stability. This metal plate should not be connected to ground.

Separate analog and digital power supply pins (AV_{DD} and DV_{DD}, respectively) allow AV_{DD} to be kept relatively free of the noisy digital signals that are often present on the system DV_{DD} line. However, though you can power AV_{DD} and DV_{DD} from two separate supplies if desired, you must ensure that they remain within ±0.3 V of one another at all times to avoid damaging the chip (as per the Absolute Maximum Ratings section). Therefore, it is recommended that unless AV_{DD} and DV_{DD} are connected directly together, back-to-back Schottky diodes should be connected between them, as shown in Figure 80.

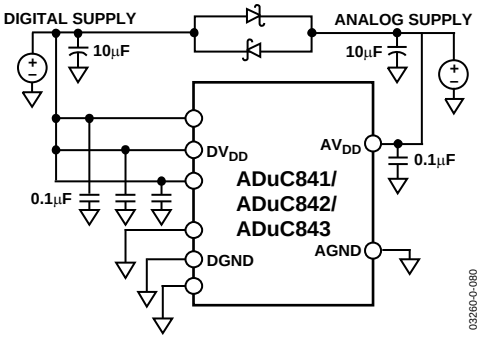


Figure 80. External Dual-Supply Connections

As an alternative to providing two separate power supplies, the user can help keep AV_{DD} quiet by placing a small series resistor and/or ferrite bead between it and DV_{DD}, and then decoupling AV_{DD} separately to ground. An example of this configuration is shown in Figure 81. With this configuration, other analog circuitry (such as op amps and voltage reference) can be powered from the AV_{DD} supply line as well. The user still needs to include back-to-back Schottky diodes between AV_{DD} and DV_{DD} to protect them from power-up and power-down transient conditions that could momentarily separate the two supply voltages.

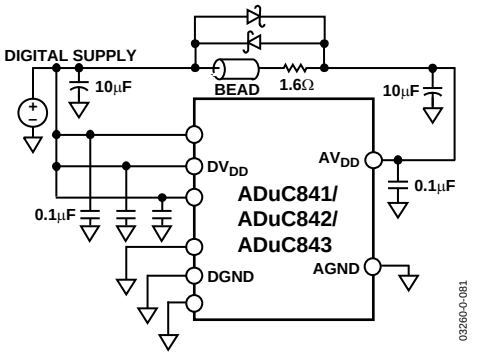


Figure 81. External Single-Supply Connections

Notice that in both Figure 80 and Figure 81, a large value (10 µF) reservoir capacitor sits on DV_{DD} and a separate 10 µF capacitor sits on AV_{DD}. Also, local small-value (0.1 µF) capacitors are located at each V_{DD} pin of the chip. As per standard design practice, be sure to include all of these capacitors, and ensure the smaller capacitors are close to each AV_{DD} pin with trace lengths as short as possible. Connect the ground terminal of each of these capacitors directly to the underlying ground plane. Finally, note that at all times, the analog and digital ground pins on the part must be referenced to the same system ground reference point.

5 V Part

For DV_{DD} below 4.5 V, the internal POR holds the part in reset. As DV_{DD} rises above 4.5 V, an internal timer times out for approximately 128 ms before the part is released from reset. The user must ensure that the power supply has reached a stable 4.75 V minimum level by this time. Likewise on power-down, the internal POR holds the part in reset until the power supply has dropped below 1 V. Figure 83 illustrates the operation of the internal POR in detail.

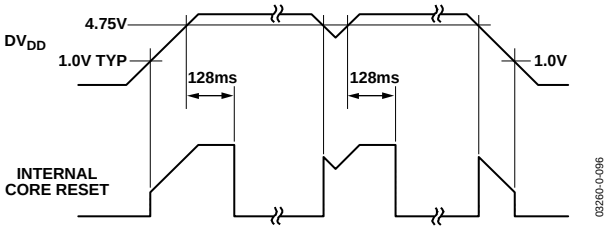


Figure 83. Internal POR Operation

Grounding and Board Layout Recommendations

As with all high resolution data converters, special attention must be paid to grounding and PC board layout of ADuC841/ADuC842/ADuC843 based designs to achieve optimum performance from the ADC and the DACs. Although the parts have separate pins for analog and digital ground (AGND and DGND), the user must not tie these to two separate ground planes unless the two ground planes are connected together very close to the part, as illustrated in the simplified example of Figure 84a. In systems where digital and analog ground planes are connected together somewhere else (for example, at the system's power supply), they cannot be connected again near the part since a ground loop would result. In these cases, tie all the part's AGND and DGND pins to the analog ground plane, as illustrated in Figure 84b. In systems with only one ground plane, ensure that the digital and analog components are physically separated onto separate halves of the board such that digital return currents do not flow near analog circuitry and vice versa. The part can then be placed between the digital and analog sections, as illustrated in Figure 84c.

In all of these scenarios, and in more complicated real-life applications, keep in mind the flow of current from the supplies and back to ground. Make sure the return paths for all currents are as close as possible to the paths that the currents took to reach their destinations. For example, do not power components on the analog side of Figure 84b with DV_{DD} since that would force return currents from DV_{DD} to flow through AGND. Also, try to avoid digital currents flowing under analog circuitry, which could happen if the user places a noisy digital chip on the left half of the board in Figure 84c. Whenever possible, avoid large discontinuities in the ground plane(s) (like those formed by a long trace on the same layer), since they force return signals to travel a longer path. And of course, make all connections to the ground plane directly, with little or no trace separating the pin from its via to ground.

If the user plans to connect fast logic signals (rise/fall time < 5 ns) to any of the part's digital inputs, a series resistor should be added to each relevant line to keep rise and fall times longer than 5 ns at the part's input pins. A value of 100 Ω or 200 Ω is usually sufficient to prevent high speed signals from coupling capacitively into the part and from affecting the accuracy of ADC conversions.

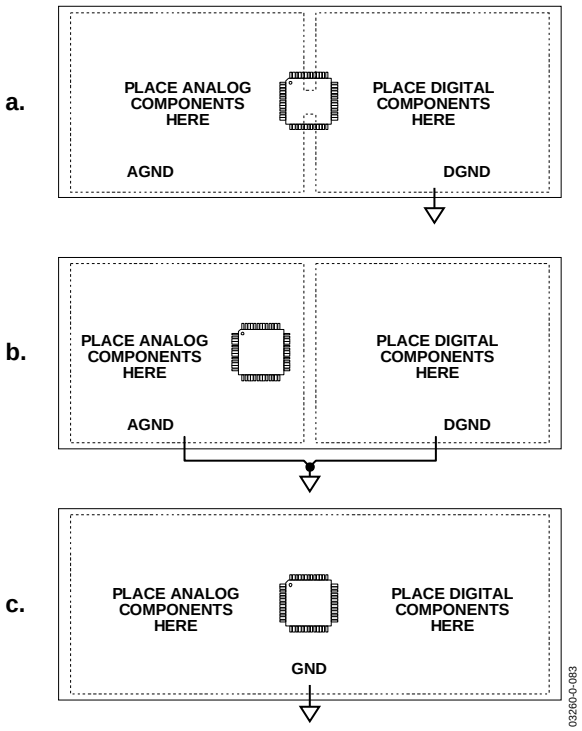


Figure 84. System Grounding Schemes

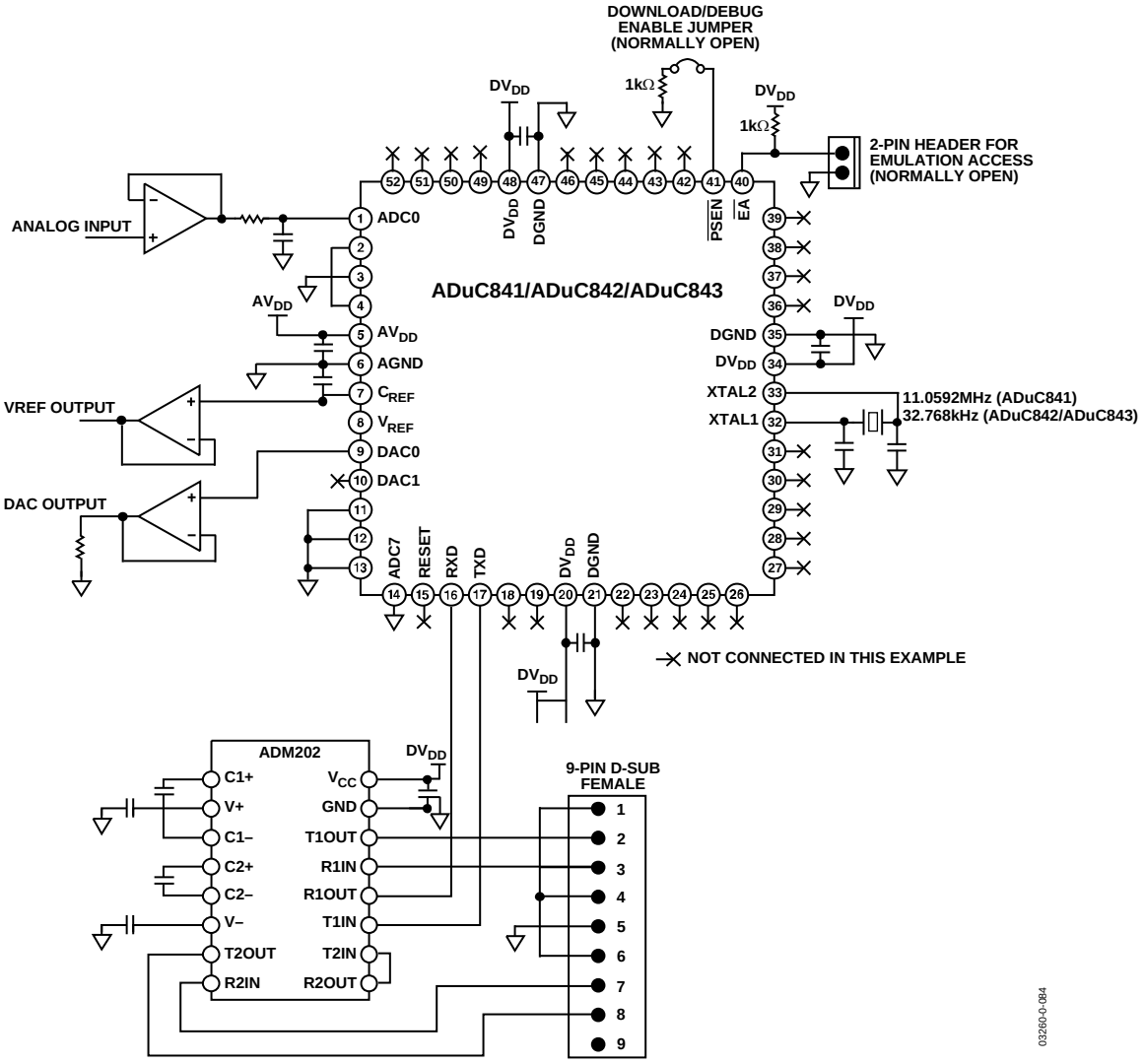


Figure 85. Example System (PQFP Package), DACs Not Present on ADuC843

OTHER HARDWARE CONSIDERATIONS

To facilitate in-circuit programming, plus in-circuit debug and emulation options, users need to implement some simple connection points in their hardware to allow easy access to download, debug, and emulation modes.

In-Circuit Serial Download Access

Nearly all ADuC841/ADuC842/ADuC843 designs want to take advantage of the in-circuit reprogrammability of the chip. This is accomplished by a connection to the ADuC841/ADuC842/ADuC843's UART, which requires an external RS-232 chip for level translation if downloading code from a PC. Basic configuration of an RS-232 connection is illustrated in Figure 85 with a simple ADM202 based circuit. If users would rather not design an RS-232 chip onto a board, refer to Application Note uC006, *A 4-Wire UART-to-PC Interface*, (at www.analog.com/microconverter) for a simple (and zero-cost-per-board) method of gaining in-circuit serial download access to the part.

In addition to the basic UART connections, users also need a way to trigger the chip into download mode. This is accomplished via a 1 kΩ pull-down resistor that can be jumpered onto the $\overline{\text{PSEN}}$ pin, as shown in Figure 85. To get the part into download mode, simply connect this jumper and power-cycle the device (or manually reset the device, if a manual reset button is available), and it is then ready to serially receive a new program. With the jumper removed, the device comes up in normal mode (and runs the program) whenever power is cycled or RESET is toggled.