Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8032
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	45
Program Memory Size	288KB (288K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/upsd3334d-40u6

Table 2. Pin definitions (continued)

Port pin	Signal name	80-Pin num.	52-Pin num.(1)	In/Out	Function		
					Basic	Alternate 1	Alternate 2
P4.6	SPITXD TCM5	19	15	I/O	General I/O port pin	PCA1-TCM5	SPI Transmit (SPITxD)
P4.7	SPISEL PCACLK1	18	14	I/O	General I/O port pin	PCACLK1	SPI Slave Select ($\overline{\text{SPISEL}}$)
AV_{REF}		70	N/A	I	Reference voltage input for ADC. Connect AV_{REF} to V_{CC} if the ADC is not used.		
$\overline{RD}$		65	N/A	O	READ Signal, external bus		
$\overline{WR}$		62	N/A	O	WRITE Signal, external bus		
$\overline{PSEN}$		63	N/A	O	$\overline{PSEN}$ Signal, external bus		
ALE		4	N/A	O	Address Latch signal, external bus		
$\overline{RESET\_IN}$		68	44	I	Active low reset input		
XTAL1		48	31	I	Oscillator input pin for system clock		
XTAL2		49	32	O	Oscillator output pin for system clock		
DEBUG		8	5	I/O	I/O to the MCU debug unit		

Standard 8032 MCU architecture cannot write to its own program memory space to prevent accidental corruption of firmware. However, this becomes an obstacle in typical 8032 systems when a remote update to firmware in Flash memory is required using IAP. The PSD module provides a solution for remote updates by allowing 8032 firmware to temporarily “reclassify” Flash memory to reside in data space during a remote update, then returning Flash memory back to program space when finished. See the VM register (*Table 115 on page 178*) in the PSD module section of this document for more details.

event signal. If not used, the Debug Event pin should be pulled up to V_{CC} as described in *Section 27.5.8: Debugging the 8032 MCU module on page 240*.

- The duration of a pulse, generated when the Event pin configured as an output, is one MCU clock cycle. This is an active-low signal, so the first edge when an event occurs is high-to-low.
- The clock to the Watchdog timer, ADC, and I^2C interface are not stopped by a breakpoint halt.
- The Watchdog timer should be disabled while debugging with JTAG, else a reset will be generated upon a watchdog timeout.

17 I/O ports of MCU module

The MCU module has three 8-bit I/O ports: Port 1, Port 3, and Port 4. The PSD module has four other I/O ports: Port A, B, C, and D. This section describes only the I/O ports on the MCU module.

I/O ports will function as bi-directional General Purpose I/O (GPIO), but the port pins can have alternate functions assigned at run-time by writing to specific SFRs. The default operating mode (during and after reset) for all three ports is GPIO input mode. Port pins that have no external connection will not float because each pin has an internal weak pull-up (~150 kOhms) to V_{CC}.

I/O ports 3 and 4 are 5 V tolerant, meaning they can be driven/pulled externally up to 5.5 V without damage. The pins on Port 4 have a higher current capability than the pins on Ports 1 and 3.

Three additional MCU ports (only on 80-pin UPSD33xx devices) are dedicated to bring out the 8032 MCU address, data, and control signals to external pins. One port, named MCUA[11:8], contains four MCU address signal outputs. Another port, named MCUAD[7:0], has eight multiplexed address/data bidirectional signals. The third port has MCU bus control outputs: read, write, program fetch, and address latch. These ports are typically used to connect external parallel peripherals and memory devices, but they may NOT be used as GPIO. Notice that only four of the eight upper address signals come out to pins on the port MCUA[11:8]. If additional high-order address signals are required on external pins (MCU addresses A[15:12]), then these address signals can be brought out as needed to PLD output pins or to the Address Out mode pins on PSD module ports. See *Section 27.4.39: Latched address output mode on page 214* for details.

Figure 15 on page 78 represents the flexibility of pin function routing controlled by the SFRs. Each of the 24 pins on three ports, P1, P3, and P4, may be individually routed on a pin-by-pin basis to a desired function.

17.1 MCU port operating modes

MCU port pins can operate as GPIO or as alternate functions (see *Figure 16* through *Figure 18 on page 80*).

Depending on the selected pin function, a particular pin operating mode will automatically be used:

- GPIO - quasi-bidirectional mode
- UART0, UART1 - quasi-bidirectional mode
- SPI - quasi-bidirectional mode
- I2C - open drain mode
- ADC - analog input mode
- PCA output - push-pull mode
- PCA input - input only (quasi-bidirectional)
- Timer 0,1,2 - input only (quasi-bidirectional)

18 MCU bus interface

The MCU module has a programmable bus interface. It is based on a standard 8032 bus, with eight data signals multiplexed with eight low-order address signals (AD[7:0]). It also has eight high-order non-multiplexed address signals (A[15:8]). Time multiplexing is controlled by the address latch signal, ALE.

This bus connects the MCU module to the PSD module, and also connects to external pins only on 80-pin devices. See the *Section 28: AC/DC parameters on page 242* at the end of this document for external bus timing on 80-pin devices.

Four types of data transfers are supported, each transfer is to/from a memory location external to the MCU module:

- Code Fetch cycle using the $\overline{\text{PSEN}}$ signal: fetch a code byte for execution
- Code Read cycle using $\overline{\text{PSEN}}$: read a code byte using the MOVC (Move Constant) instruction
- XDATA Read cycle using the $\overline{\text{RD}}$ signal: read a data byte using the MOVX (Move eXternal) instruction
- XDATA Write cycle using the $\overline{\text{WR}}$ signal: write a data byte using the MOVX instruction

The number of MCU_CLK periods for these transfer types can be specified at runtime by firmware writing to the SFR register named BUSCON (*Table 47 on page 87*). Here, the number of MCU_CLK clock pulses per bus cycle are specified to maximize performance.

Note: ***Important:*** *By default, the BUSCON register is loaded with long bus cycle times (6 MCU_CLK periods) after a reset condition. It is important that the post-reset initialization firmware sets the bus cycle times appropriately to get the most performance, according to Table 49 on page 88. Keep in mind that the PSD module has a faster Turbo mode (default) and a slower but less power consuming Non-Turbo mode. The bus cycle times must be programmed in BUSCON to optimize for each mode as shown in Table 49 on page 88. See Section 27.4.55: PLD non-turbo mode on page 230 for more details.*

18.1 Bus read cycles ($\overline{\text{PSEN}}$ or $\overline{\text{RD}}$)

When the $\overline{\text{PSEN}}$ signal is used to fetch a byte of code, the byte is read from the PSD module or external device and it enters the MCU Pre-Fetch Queue (PFQ). When $\overline{\text{PSEN}}$ is used during a MOVC instruction, or when the $\overline{\text{RD}}$ signal is used to read a byte of data, the byte is routed directly to the MCU, bypassing the PFQ.

Bits in the BUSCON register determine the number of MCU_CLK periods per bus cycle for each of these kinds of transfers to all address ranges.

It is not possible to specify in the BUSCON register a different number of MCU_CLK periods for various address ranges. For example, the user cannot specify 4 MCU_CLK periods for $\overline{\text{RD}}$ read cycles to one address range on the PSD module, and 5 MCU_CLK periods for $\overline{\text{RD}}$ read cycles to a different address range on an external device. However, the user can specify one number of clock periods for $\overline{\text{PSEN}}$ read cycles and a different number of clock periods for $\overline{\text{RD}}$ read cycles.

Note: 1 A $\overline{PSEN}$ bus cycle in progress may be aborted before completion if the PFQ and Branch Cache (BC) determines the current code fetch cycle is not needed.

2 Whenever the same number of MCU_CLK periods is specified in BUSCON for both $\overline{PSEN}$ and $\overline{RD}$ cycles, the bus cycle timing is typically identical for each of these types of bus cycles. In this case, the only time $\overline{PSEN}$ read cycles are longer than $\overline{RD}$ read cycles is when the PFQ issues a stall while reloading. PFQ stalls do not affect $\overline{RD}$ read cycles. By comparison, in many traditional 8051 architectures, $\overline{RD}$ bus cycles are always longer than $\overline{PSEN}$ bus cycles.

18.2 Bus write cycles ($\overline{WR}$)

When the $\overline{WR}$ signal is used, a byte of data is written directly to the PSD module or external device, no PFQ or caching is involved. Bits in the BUSCON register determine the number of MCU_CLK periods for bus write cycles to all addresses. It is not possible to specify in BUSCON a different number of MCU_CLK periods for writes to various address ranges.

18.3 Controlling the PFQ and BC

The BUSCON register allows firmware to enable and disable the PFQ and BC at run-time. Sometimes it may be desired to disable the PFQ and BC to ensure deterministic execution. The dynamic action of the PFQ and BC may cause varying program execution times depending on the events that happen prior to a particular section of code of interest. For this reason, it is not recommended to implement timing loops in firmware, but instead use one of the many hardware timers in the UPSD33xx.

By default, the PFQ and BC are enabled after a reset condition.

*Note: **Important:** Disabling the PFQ or BC will seriously reduce MCU performance.*

20.6.2 Auto-reload mode

In the Auto-reload mode, there are again two options, which are selected by the bit EXEN2 in T2CON. *Figure 25 on page 104* shows Auto-reload mode.

If EXEN2 = 0, then when Timer 2 counts up and rolls over from FFFFh it not only sets the interrupt flag TF2, but also causes the Timer 2 registers to be reloaded with the 16-bit value contained in registers RCAP2L and RCAP2H, which are preset with firmware.

If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2X will also trigger the 16-bit reload and set the interrupt flag EXF2. Again, firmware servicing the interrupt must read both TF2 and EXF2 to determine the cause, and clear the flag(s) upon exit.

Note: The UPSD33xx does not support selectable up/down counting in Auto-reload mode (this feature was an extension to the original 8032 architecture).

Table 58. T2CON: Timer 2 Control register (SFR C8h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Table 59. T2CON register bit definition

Bit	Symbol	R/W	Definition
7	TF2	R,W	**Timer 2 flag,** causes interrupt if enabled. TF2 is set by hardware upon overflow. Must be cleared by firmware. TF2 will not be set when either RCLK or TCLK =1.
6	EXF2	R,W	**Timer 2 flag,** causes interrupt if enabled. EXF2 is set when a capture or reload is caused by a negative transition on T2X pin and EXEN2 = 1. EXF2 must be cleared by firmware.
5	RCLK(1)	R,W	**UART0 Receive Clock control.** When RCLK = 1, UART0 uses Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK=0, Timer 1 overflow is used for its receive clock
4	TCLK(1)	R,W	**UART0 Transmit Clock control.** When TCLK = 1, UART0 uses Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK=0, Timer 1 overflow is used for transmit clock
3	EXEN2	R,W	**Timer 2 External Enable.** When EXEN2 = 1, capture or reload results when negative edge on pin T2X occurs. EXEN2 = 0 causes Timer 2 to ignore events at pin T2X.
2	TR2	R,W	**Timer 2 run control.** 1 = Timer/Counter 2 is on, 0 = Timer Counter 2 is off.

21.3.2 Using Timer/Counter 2 to generate baud rates

See *Section 20.6.3: Baud rate generator mode on page 102*.

Table 67. Commonly used baud rates generated from Timer 1

UART mode	f_{OSC} (MHz)	Desired baud rate	Resultant baud rate	Baud rate deviation	SMOD bit in PCON	Timer 1		
						C/$\overline{T}$ Bit in TMOD	Timer mode in TMOD	TH1 Reload value (hex)
Mode 0 Max	40.0	3.33 MHz	3.33MHz	0	X	X	X	X
Mode 2 Max	40.0	1250 kHz	1250 kHz	0	1	X	X	X
Mode 2 Max	40.0	625 kHz	625 kHz	0	0	X	X	X
Modes 1 or 3	40.0	19200 Hz	18939 Hz	-1.36%	1	0	2	F5
Modes 1 or 3	40.0	9600 Hz	9470 Hz	-1.36%	1	0	2	EA
Modes 1 or 3	36.0	19200 Hz	18570 Hz	-2.34%	1	0	2	F6
Modes 1 or 3	33.333	57600 Hz	57870 Hz	0.47%	1	0	2	FD
Modes 1 or 3	33.333	28800 Hz	28934 Hz	0.47%	1	0	2	FA
Modes 1 or 3	33.333	19200 Hz	19290 Hz	0.47%	1	0	2	F7
Modes 1 or 3	33.333	9600 Hz	9645 Hz	0.47%	1	0	2	EE
Modes 1 or 3	24.0	9600 Hz	9615 Hz	0.16%	1	0	2	F3
Modes 1 or 3	12.0	4800 Hz	4808 Hz	0.16%	1	0	2	F3
Modes 1 or 3	11.0592	57600 Hz	57600 Hz	0	1	0	2	FF
Modes 1 or 3	11.0592	28800 Hz	28800 Hz	0	1	0	2	FE
Modes 1 or 3	11.0592	19200 Hz	19200 Hz	0	1	0	2	FD
Modes 1 or 3	11.0592	9600 Hz	9600 Hz	0	1	0	2	FA
Modes 1 or 3	3.6864	19200 Hz	19200 Hz	0	1	0	2	FF

21.6 More about UART modes 2 and 3

For mode 2, refer to the block diagram in *Figure 31 on page 117*, and timing diagram in *Figure 32 on page 117*. For mode 3, refer to the block diagram in *Figure 33 on page 118*, and timing diagram in *Figure 34 on page 118*.

Keep in mind that the baud rate is programmable to either 1/32 or 1/64 of f_{OSC} in mode 2, but mode 3 uses a variable baud rate generated from Timer 1 or Timer 2 rollovers.

The receive portion is exactly the same as in mode 1. The transmit portion differs from mode 1 only in the 9th bit of the transmit shift register.

Transmission is initiated by any instruction which writes to SBUF. At the end of a write operation to SBUF, the TB8 Bit is loaded into the 9th position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually starts at the end of the MCU the machine cycle following the next rollover in the divide-by-16 counter. Thus, the bit times are synchronized to the divide-by-16 counter, not to the writing of SBUF. Transmission begins with activation of SEND which puts the start bit at pin TxD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to pin TxD. The first shift pulse occurs one bit time after that. The first shift clocks a '1' (the stop bit) into the 9th bit position of the shift register. There-after, only zeros are clocked in. Thus, as data bits shift out to the right, zeros are clocked in from the left. When bit TB8 is at the output position of the shift register, then the stop bit is just to the left of TB8, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift and then deactivate SEND, and set the interrupt flag, TI. This occurs at the 11th divide-by 16 rollover after writing to SBUF.

Reception is initiated by a detected 1-to-0 transition at pin RxD. For this purpose RxD is sampled at a rate of 16 times whatever baud rate has been established. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register. At the 7th, 8th, and 9th counter states of each bit time, the bit detector samples the value of RxD. The value accepted is the value that was seen in at least 2 of the 3 samples. If the value accepted during the first bit time is not '0,' the receive circuits are reset and the unit goes back to looking for another '1'-to-'0' transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame will proceed. As data bits come in from the right, '1s' shift out to the left. When the start bit arrives at the left-most position in the shift register (which in modes 2 and 3 is a 9-bit register), it flags the RX Control unit to do one last shift, load SBUF and RB8, and set the interrupt flag RI. The signal to load SBUF and RB8, and to set RI, will be generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

- RI = 0, and
- Either SM2 = 0, or the received 9th data bit = 1. If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received 9th data bit goes into RB8, and the first 8 data bits go into SBUF. One bit time later, whether the above conditions were met or not, the unit goes back to looking for a '1'-to-'0' transition on pin RxD.

27.1.15 JTAG port

In-system programming (ISP) can be performed through the JTAG signals on Port C. This serial interface allows programming of the entire PSD module device or subsections of the PSD module (for example, only Flash memory but not the PLDs) without the participation of the 8032. A blank UPSD33xx device soldered to a circuit board can be completely programmed in 10 to 25 seconds. The four basic JTAG signals on Port C; TMS, TCK, TDI, and TDO form the IEEE-1149.1 interface. The PSD module does not implement the IEEE-1149.1 Boundary Scan functions, but uses the JTAG interface for ISP an 8032 debug. The PSD module can reside in a standard JTAG chain with other JTAG devices and it will remain in BYPASS mode when other devices perform JTAG functions. ISP programming time can be reduced as much as 30% by using two optional JTAG signals on Port C, TSTAT and $\overline{\text{TERR}}$, in addition to TMS, TCK, TDI and TDO, and this is referred to as “6-pin JTAG”. The FlashLINK JTAG programming cable is available from STMicroelectronics and PSDsoft Express software is available at no charge from *www.st.com/mcu*. More JTAG ISP information maybe found in the section titled “JTAG ISP and debug” on page 137.

The MCU module is also included in the JTAG chain within the UPSD33xx device for 8032 debugging and emulation. While debugging, the PSD module is in BYPASS mode. Conversely, during ISP, the MCU module is in BYPASS mode.

27.1.16 Power management

The PSD module has bits in csiop registers that are configured at run-time by the 8032 to reduce power consumption of the GPLD. The Turbo Bit in the PMMR0 register can be set to logic ’1’ and both PLDs will go to Non-Turbo mode, meaning it will latch its outputs and go to sleep until the next transition on its inputs. There is a slight penalty in PLD performance (longer propagation delay), but significant power savings are realized. Going to Non-Turbo mode may require an additional wait state in the 8032 SFR, BUSCON, because memory decode signals are also delayed. The default state of the Turbo Bit is logic '0,' meaning by default, the GPLD is in fast Turbo mode until the Turbo mode is turned off.

Additionally, bits in csiop registers PMMR0 and PMMR2 can be set by the 8032 to selectively block signals from entering both PLDs which further reduces power consumption. There is also an Automatic Power-down counter that detects lack of 8032 activity and reduces power consumption on the PSD module to its lowest level (see *Section 27.1.16: Power management on page 170*).

27.1.17 Security and NVM sector protection

A programmable security bit in the PSD module protects its contents from unauthorized viewing and copying. The security bit is specified in PSDsoft Express and programmed into the UPSD33xx with JTAG. Once set, the security bit will block access of JTAG programming equipment to the PSD module Flash memory and PLD configuration, and also blocks JTAG debugging access to the MCU module. The only way to defeat the security bit is to erase the entire PSD module using JTAG (the erase command is the only JTAG command allowed after the security bit has been set), after which the device is blank and may be used again.

Additionally and independently, the contents of each individual Flash memory sector can be write protected (sector protection) by configuration with PSDsoft Express. This is typically used to protect 8032 boot code from being corrupted by inadvertent WRITEs to Flash memory from the 8032.

Status of sector protection bits may be read (but not written) using two registers in csiop space.

Each Flash memory requires the 8032 to send an instruction sequence to program a byte or to erase sectors (see *Table 117 on page 184*).

If the byte to be programmed is in a protected Flash memory sector, the instruction sequence is ignored.

Note: ***IMPORTANT:*** *It is mandatory that a chip-select signal is active for the Flash sector where a programming instruction sequence is targeted. Make sure that the correct chip-select equation, FSx, or CSBOOTx specified in PSDsoft Express matches the address range that the 8032 firmware is accessing, otherwise the instruction sequence will not be recognized by the Flash array. If memory paging is used, be sure that the 8032 firmware sets the page register to the correct page number before issuing an instruction sequence to the Flash memory segment on a particular memory page, otherwise the correct sector select signal will not become active.*

Once the 8032 issues a Flash memory program or erase instruction sequence, it must check the status bits for completion. The embedded algorithms that are invoked inside a Flash memory array provide several ways to give status to the 8032. Status may be checked using any of three methods: Data Polling, Data Toggle, or Ready/$\overline{\text{Busy}}$ (pin PC3).

Table 118. Flash Memory Status bit definition(1)(2)

Functional Block	FSx, or CSBOOTx	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0
Flash memory	Active (the desired segment is selected)	Data Polling	Toggle Flag	Error Flag	X	Erase Timeout	X	X	X

1. X = Not guaranteed value, can be read either '1' or '0.'
2. DQ7-DQ0 represent the 8032 Data Bus Bits, D7-D0.

27.4.10 Data polling

Polling on the Data Polling Flag Bit (DQ7) is a method of checking whether a program or erase operation is in progress or has completed. *Figure 60 on page 188* shows the Data Polling algorithm.

When the 8032 issues a program instruction sequence, the embedded algorithm within the Flash memory array begins. The 8032 then reads the location of the byte to be programmed in Flash memory to check status. The Data Polling Flag Bit (DQ7) of this location becomes the compliment of Bit D7 of the original data byte to be programmed. The 8032 continues to poll this location, comparing the Data Polling Flag Bit (DQ7) and monitoring the Error Flag Bit (DQ5). When the Data Polling Flag Bit (DQ7) matches Bit D7 of the original data, then the embedded algorithm is complete. If the Error Flag Bit (DQ5) is '1,' the 8032 should test the Data Polling Flag Bit (DQ7) again since the Data Polling Flag Bit (DQ7) may have changed simultaneously with the Error Flag Bit (DQ5) (see *Figure 60 on page 188*).

The Error Flag Bit (DQ5) is set if either an internal timeout occurred while the embedded algorithm attempted to program the byte (indicating a bad Flash cell) or if the 8032 attempted to program bit to logic ’1’ when that bit was already programmed to logic ’0’ (must erase to achieve logic ’1’).

It is suggested (as with all Flash memories) to read the location again after the embedded programming algorithm has completed, to compare the byte that was written to the Flash memory with the byte that was intended to be written.

on the PSD module as an input to both PLDs (without routing a signal externally on PC board) and it's signal name is "rd_bsy". The Ready/Busy output can be probed during lab development to check the timing of Flash memory programming in the system at run-time.

27.4.13 Bypassed Unlock sequence

The Bypass Unlock mode allows the 8032 to program bytes in the Flash memories faster than using the standard Flash program instruction sequences because the typical AAh, 55h unlock bus cycles are bypassed for each byte that is programmed. Bypassing the unlock sequence is typically used when the 8032 is intentionally programming a large number of bytes (such as during IAP). After intentional programming is complete, typically the Bypass mode would be disabled, and full protection is back in place to prevent unwanted WRITEs to Flash memory.

The Bypass Unlock mode is entered by first initiating two Unlock bus cycles. This is followed by a third WRITE operation containing the Bypass Unlock command, 20h (as shown in *Table 117 on page 184*). The Flash memory array that received that sequence then enters the Bypass Unlock mode. After this, a two bus cycle program operation is all that is required to program a byte in this mode. The first bus cycle in this shortened program instruction sequence contains the Bypassed Unlocked Program command, A0h, to any valid address within the unlocked Flash array. The second bus cycle contains the address and data of the byte to be programmed. Programming status is checked using toggle, polling, or Ready/Busy just as before. Additional data bytes are programmed the same way until this Bypass Unlock mode is exited.

To exit Bypass Unlock mode, the system must issue the Reset Bypass Unlock instruction sequence. The first bus cycle of this instruction must write 90h to any valid address within the unlocked Flash Array; the second bus cycle must write 00h to any valid address within the unlocked Flash Array. After this sequence the Flash returns to Read Array mode.

During Bypass Unlock mode, only the Bypassed Unlock Program instruction, or the Reset Bypass Unlock instruction is valid, other instruction will be ignored.

27.4.14 Erasing Flash memory

Flash memory may be erased sector-by-sector, or an entire Flash memory array may be erased with one command (bulk).

27.4.15 Flash bulk Erase

The Flash Bulk Erase instruction sequence uses six WRITE operations followed by a READ operation of the status register, as described in *Table 117 on page 184*. If any byte of the Bulk Erase instruction sequence is wrong, the Bulk Erase instruction sequence aborts and the device is reset to the Read Array mode. The address provided by the 8032 during the Flash Bulk Erase command sequence may select any one of the eight Flash memory sector select signals FSx or one of the four signals CSBOOTx. An erase of the entire Flash memory array will occur in a particular array even though a command was sent to just one of the individual Flash memory sectors within that array.

During a Bulk Erase, the memory status may be checked by reading the Error Flag Bit (DQ5), the Toggle Flag Bit (DQ6), and the Data Polling Flag Bit (DQ7). The Error Flag Bit (DQ5) returns a '1' if there has been an erase failure. Details of acquiring the status of the Bulk Erase operation are detailed in *Section 27.4.9: Programming Flash memory on page 186*.

used, it is referred to as “6-pin JTAG”. PC3 and PC4 cannot be used for other functions if they are used for 6-pin JTAG. See *Section 27.5.1: JTAG ISP and JTAG debug on page 233* for details.

- PC3 can be used as an output to indicate when a Flash memory program or erase operation has completed. This is specified in PSDsoft Express as *Section 27.4.12: Ready/Busy (PC3) on page 189*.

The remaining four pins (TDI, TDO, TCK, TMS) on Port C are dedicated to the JTAG function and cannot be used for any other function. See *Section 27.5.1: JTAG ISP and JTAG debug on page 233*.

Port C also supports the open drain output drive type options on pins PC2, PC3, PC4, and PC7 using the csiop Drive Select registers.

no PLD inputs are changing, and the PLDs will even use less AC current when inputs do change compared to Turbo mode.

When the Turbo mode is enabled, there is a significant DC current component AND the AC current component is higher than non-Turbo mode, as shown in *Figure 84 on page 242* (5 V) and *Figure 85 on page 243* (3.3 V).

Blocking bits

Significant power savings can be achieved by blocking 8032 bus control signals ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE) from reaching PLD inputs, if these signals are not used in any PLD equations. Blocking is achieved by the 8032 writing to the "blocking bits" in csiop PMMR registers. Current consumption of the PLDs is directly related to the composite frequency of all transitions on PLD inputs, so blocking certain PLD inputs can significantly lower PLD operating frequency and power consumption (resulting in a lower frequency on the graphs of *Figure 84 on page 242* and *Figure 85 on page 243*).

Note: It is recommended to prevent unused inputs from floating on Ports A, B, C, and D by pulling them up to V_{DD} with a weak external resistor (100 KΩ), or by setting the csiop Direction register to "output" at run-time for all unused inputs. This will prevent the CMOS input buffers of unused input pins from drawing excessive current.

The csiop PMMR register definitions are shown in *Table 154* through *Table 156 on page 226*.

Table 154. Power Management Mode register PMMR0 (address = csiop + offset B0h)(1)

Bit num.	Bit name	Value	Description
Bit 0	X	0	Not used, and should be set to zero.
Bit 1	APD Enable	0	Automatic Power-down (APD) counter is disabled.
		1	APD counter is enabled
Bit 2	X	0	Not used, and should be set to zero.
Bit 3	PLD Turbo Disable	0 = on	PLD Turbo mode is on
		1 = off	PLD Turbo mode is off, saving power.
Bit 4	Blocking bit, CLKIN to PLDs(2)	0 = on	CLKIN (pin PD1) to the PLD Input Bus is not blocked. Every transition of CLKIN powers-up the PLDs.
		1 = off	CLKIN input to PLD Input Bus is blocked, saving power. But CLKIN still goes to APD counter.
Bit 5	Blocking bit, CLKIN to OMCs only(2)	0 = on	CLKIN input is not blocked from reaching all OMC's common clock inputs.
		1 = off	CLKIN input to common clock of all OMCs is blocked, saving power. But CLKIN still goes to APD counter and all PLD logic besides the common clock input on OMCs.
Bit 6	X	0	Not used, and should be set to zero.
Bit 7	X	0	Not used, and should be set to zero.

1. All the bits of this register are cleared to zero following Power-up. Subsequent Reset ($\overline{RST}$) pulses do not clear the registers.
2. Blocking bits should be set to logic '1' only if the signal is not needed in a DPLD or GPLD logic equation.

27.4.53 Forced Power-down (FDP)

An alternative to APD is FPD. The resulting power-savings is the same, but the PDN signal in *Figure 77 on page 229* is set and Power-down mode is entered immediately when firmware sets the FORCE_PD Bit to logic '1' in the csiop *register* PMMR3 (Bit 1). FPD will override APD counter activity when FORCE_PD is set. No external clock source for the APD counter is needed. The FORCE_PD Bit is cleared only by a reset condition.

Caution must be used when implementing FPD because code memory goes off-line as soon as PSD module Power-down mode is entered, leaving the MCU with no instruction stream to execute.

The MCU module must put itself into Power-down mode after it puts the PSD module into Power-down mode. How can it do this if code memory goes off-line? The answer is the Pre-Fetch Queue (PFQ) in the MCU module. By using the instruction scheme shown in the 8051 assembly code example in *Table 157 on page 228*, the PFQ will be loaded with the final instructions to command the MCU module to Power-down mode after the PDS module goes to Power-down mode. In this case, even though the code memory goes off-line in the PSD module, the last few MCU instruction are sourced from the PFQ.

Table 157. Forced Power-down example

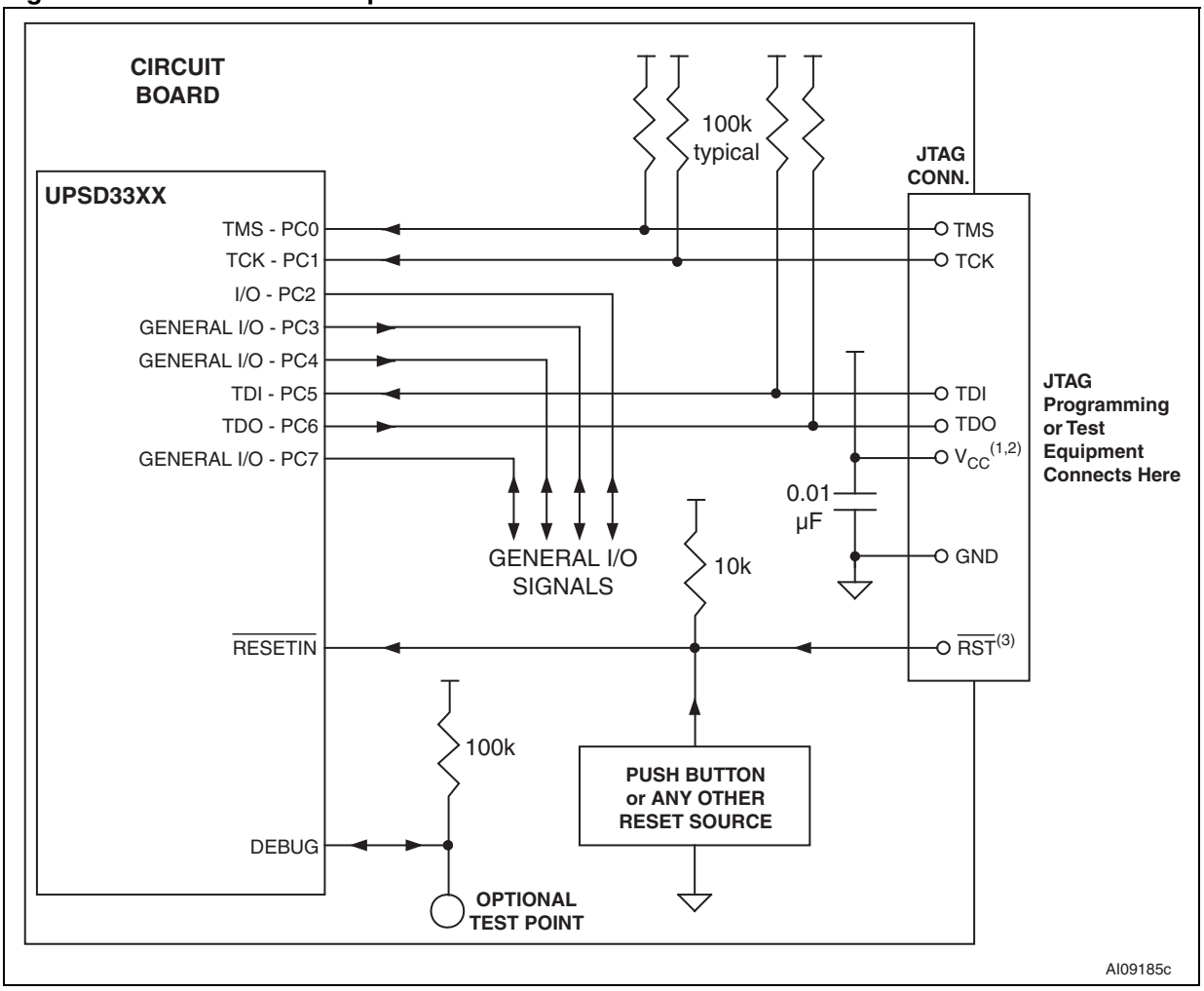
```
PDOWN:  ANL   A8h, #7Fh      ; disable all interrupts
        ORL   9Dh, #C0h      ; ensure PFQ and BC are enabled
        MOV   DPTR, #xxC7    ; load XDATA pointer to select PMMR3 register (xx = base
                             ; address of csiop registers)
        CLR   A              ; clear A
        JMP   LOOP           ; first loop - fill PFQ/BQ with Power-down instructions
        NOP                  ; second loop - fetch code from PFQ/BC and set Power-
                             ; Down bits for PSD module and then MCU module
LOOP:   MOVX  @DPTR, A       ; set FORCE_PD Bit in PMMR3 in PSD module in second
                             ; loop
        MOV   87h, A         ; set PD Bit in PCON register in MCU module in second
                             ; loop
        MOV   A, #02h        ; set power-down bit in the A register, but not in PMMR3 or
                             ; PCON yet in first loop
        JMP   LOOP           ; UPSD enters into Power-down mode in second loop
```

27.5.4 4-pin JTAG ISP (default)

The four basic JTAG pins on Port C are enabled for JTAG operation at all times. These pins may not be used for other I/O functions. There is no action needed in PSDsoft Express to configure a device to use 4-pin JTAG, as this is the default condition. No 8032 firmware is needed to use 4-pin ISP because all ISP functions are controlled from the external JTAG program/test equipment. *Figure 80 on page 236* shows recommended connections on a circuit board to a JTAG program/test tool using 4-pin JTAG. It is required to connect the $\overline{\text{RST}}$ output signal from the JTAG program/test equipment to the $\overline{\text{RESET\_IN}}$ input on the UPSD33xx. The $\overline{\text{RST}}$ signal is driven by the equipment with an Open Drain driver, allowing other sources (like a push button) to drive $\overline{\text{RESET\_IN}}$ without conflict.

Note: The recommended pull-up resistors and decoupling capacitor are illustrated in Figure 80.

Figure 80. Recommended 4-pin JTAG connections



1. For 5 V UPSD33xx devices, pull-up resistors and V_{CC} pin on the JTAG connector should be connected to 5 V system V_{DD}.
2. For 3.3 V UPSD33xx devices, pull-up resistors and V_{CC} pin on the JTAG connector should be connected to 3.3 V system V_{CC}.
3. This signal is driven by an Open-Drain output in the JTAG equipment, allowing more than one source to activate $\overline{\text{RESETIN}}$.

Figure 85. PLD I_{CC} /frequency consumption (3 V range)

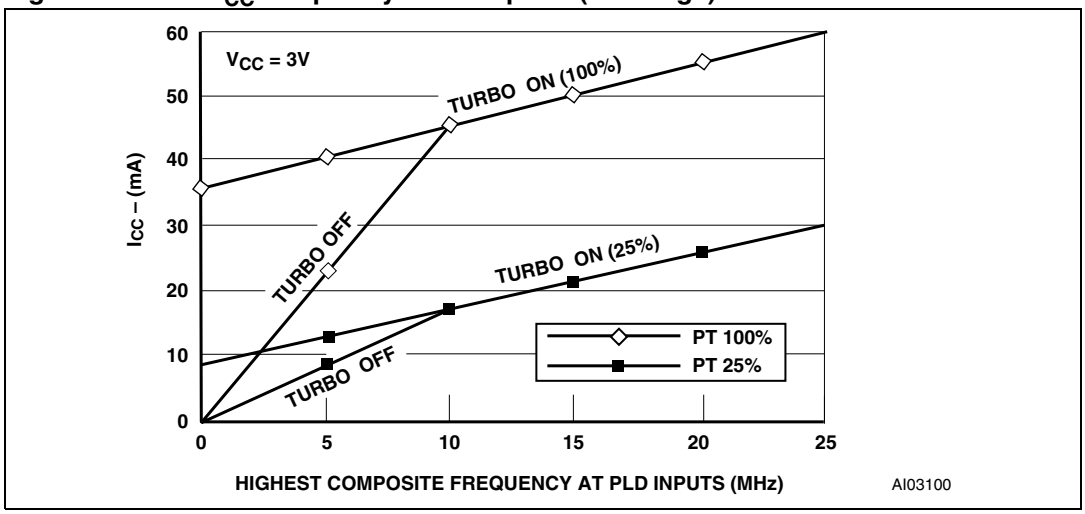
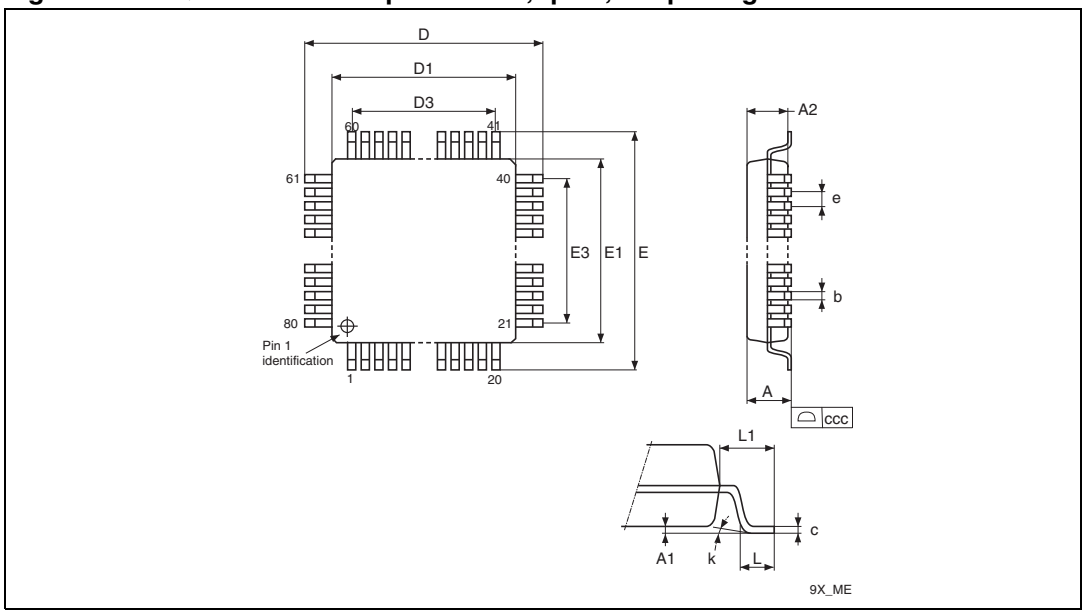


Table 159. PSD module example, typ. power calculation at V_{CC} = 5.0 V (Turbo mode Off)

Conditions		
	MCU Clock Frequency	= 12 MHz
Highest Composite PLD input frequency		
	(Freq PLD)	= 8 MHz
MCU ALE frequency (Freq ALE)		= 2 MHz
	% Flash memory Access	= 80%
	% SRAM access	= 15%
	% I/O access	= 5% (no additional power above base)
Operational modes		
	% Normal	= 40%
	% Power-down mode	= 60%
Number of product terms used		
	(from fitter report)	= 45 PT
	% of total product terms	= 45/182 = 24.7%
	Turbo mode	= Off
Calculation (using typical values)		
I_{CC} total	= I_{CC}(MCUactive) x %MCUactive + I_{CC}(PSDactive) x %PSDactive + I_{PD}(pwrdown) x %pwrdown	

Figure 103. LQFP80 – 80-lead plastic thin, quad, flat package outline



1. Drawing not to scale.

Table 192. LQFP80 – 80-lead plastic thin, quad, flat package mechanical data

Symbol	millimeters			inches[1]		
	Typ	Min	Max	Typ	Min	Max
A			1.600			0.0630
A1		0.050	0.150		0.0020	0.0059
A2	1.400	1.350	1.450	0.0551	0.0531	0.0571
b	0.220	0.170	0.270	0.0087	0.0067	0.0106
C		0.090	0.200		0.0035	0.0079
D	14.000			0.5512		
D1	12.000			0.4724		
D3	9.500			0.3740		
E	14.000			0.5512		
E1	12.000			0.4724		
E3	9.500			0.3740		
e	0.500			0.0197		
L	0.600	0.450	0.750	0.0236	0.0177	0.0295
L1	1.000			0.0394		
k		0°	7°		0°	7°
ccc	0.080			0.0031		

1. Values in inches are converted from mm and rounded to 4 decimal digits.

32 Part numbering

Table 193. Ordering information scheme

Example: UPSD 33 3 4 D V – 40 U 6 T

Device Type

UPSD = Microcontroller PSD

Family

33 = Turbo core

SRAM Size

1 = 2 Kbyte

3 = 8 Kbyte

5 = 32 Kbyte

Main Flash memory Size

2 = 64 Kbyte

3 = 128 Kbyte

4 = 256 Kbyte

IP Mix

D = IP Mix: I^2C, SPI, UART(2), IrDA, ADC, Supervisor, PCA

Operating voltage

blank = V_{CC} = 4.5 to 5.5 V

V = V_{CC} = 3.0 to 3.6V

Speed

–40 = 40 MHz

Package

T = 52-pin LQFP ECOPACK-compliant package

U = 80-pin LQFP ECOPACK-compliant package

Temperature Range

6 = –40 to 85 °C

Shipping Option

Tape & Reel Packing = T

For a list of available options (e.g., Speed, Package) or for further information on any aspect of this device, please contact the ST Sales Office nearest to you.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com