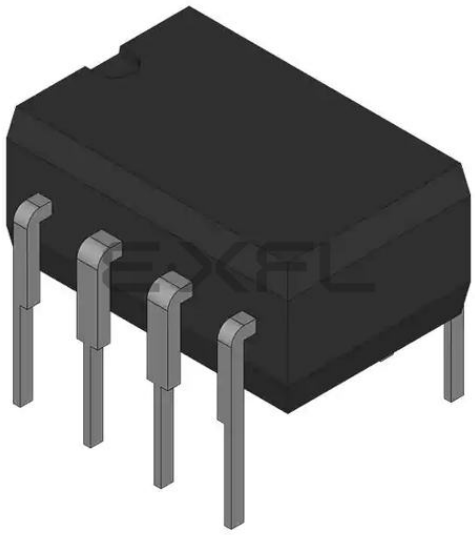


Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

Details	
Product Status	Obsolete
Applications	Digital Power Controller
Core Processor	8-Bit
Program Memory Type	EEPROM (1kB)
Controller Series	-
RAM Size	64 x 8
Interface	-
Number of I/O	6
Voltage - Supply	2.7V ~ 3.6V
Operating Temperature	-40°C ~ 85°C
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-DIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=fms7401len

List of Tables

1 Reset Circuit

The reset circuit in the FMS7401L contains four input conditions that trigger a main system reset. When the main system reset is triggered, a sequence of events occur defaulting all memory mapped registers (including the initialization registers) and I/Os to their initial states (see Table 1). During the system reset sequence, the instruction core execution is halted allowing time for the internal oscillator and other analog circuits to stabilize. Once the system reset sequence completes, the device will begin with its normal operation executing the instruction program residing in the code EEPROM memory. The time required for the system reset sequence to complete (T_{RESET}) is dependent on the individual trigger condition and is defined in the Electrical Characteristics section of the datasheet. The four reset trigger conditions are as follows:

- Power-on Reset (POR)
- External Reset[1]
- Brown-out Reset (BOR)
- Watchdog Reset[2]

Table 1. Default Register States

Peripheral/Register	External Reset	POR
G1, G2, G3, G4, G6, G7	High-impedance input (tri-state input)	
G0, G5	Defined by Init Reg. 4 (see Table 28)	
SRAM Memory	No change	Unspecified
Stack Pointer	0xF	0xF
Status Register	0x80	0x80
T1CMPA, T1CMPB and T1RA Registers	0xFFF	0xFFF
DTIME Register	0x1F	0x1F
All other memory mapped register not listed above.[3]	0x00	0x00

1.1 Power-on Reset Circuit

The Power-on Reset (POR) circuit maintains the device in a reset state until Vcc reaches a voltage level high enough to guarantee proper device operation. The POR circuit is sensitive to the different Vcc ramp rates and must be within S_{Vcc} as specified in the Electrical Characteristics section of the datasheet.

The POR circuit does not generate a system reset when Vcc is falling. This feature is performed by the Brown-out Reset (BOR) circuit and must be enabled by the BOREN bit of the Initialization Register 1.[4] In the case where Vcc does not drop to 0V before the next power-up sequence, it is necessary to enable the BOR circuit and/or reset the device externally through the RESET pin.[1]

1.2 External Reset[1]

The device may be externally reset through the RESET input pin if the POR/BOR circuits cannot be used to properly reset the device in the application. The RESET input pin contains an internal pull-up resistor making it an active low signal. Therefore, to issue a device system reset the RESET input should be held low for at least 10µS before being released (i.e. returned to a high state). While the RESET input is held low, the internal oscillator and other analog circuits are kept in a low power state reducing the current consumption of the device (a state resembling Halt Mode). In addition, the I/O pins are all initialized to an input tri-state configuration unless defaulted otherwise.[5] At the rising edge of the RESET input signal, the main system reset sequence is triggered releasing the internal oscillator and other analog circuits so that they may be initialized and begin their normal operation.

1.3 Brown-out Reset Circuit

The Brown-out Reset (BOR) circuit is one of the on-chip analog comparator peripherals and must be enabled through the BOREN bit of the Initialization Registers 1.[4] The BOR circuit is used to hold the device in a reset state when Vcc drops below a fixed threshold defined in the Electrical Characteristics section of the datasheet. While in reset, the device is held in its initial condition until Vcc rises above the fixed/power-on threshold. Shortly after Vcc rises above the fixed/power-on threshold, the internal system reset sequence is started. Once the system reset sequence completes, the device will begin with its normal operation executing the instruction program residing in the code EEPROM memory.

3 Power Saving Modes

The FMS7401L has both Halt and Idle power saving modes. Each mode is controlled by software and offers the advantage of reducing the total current consumption of the device in an application. For all current consumption details, please refer to the Electrical Characteristics section of the datasheet.

3.1 Halt Mode

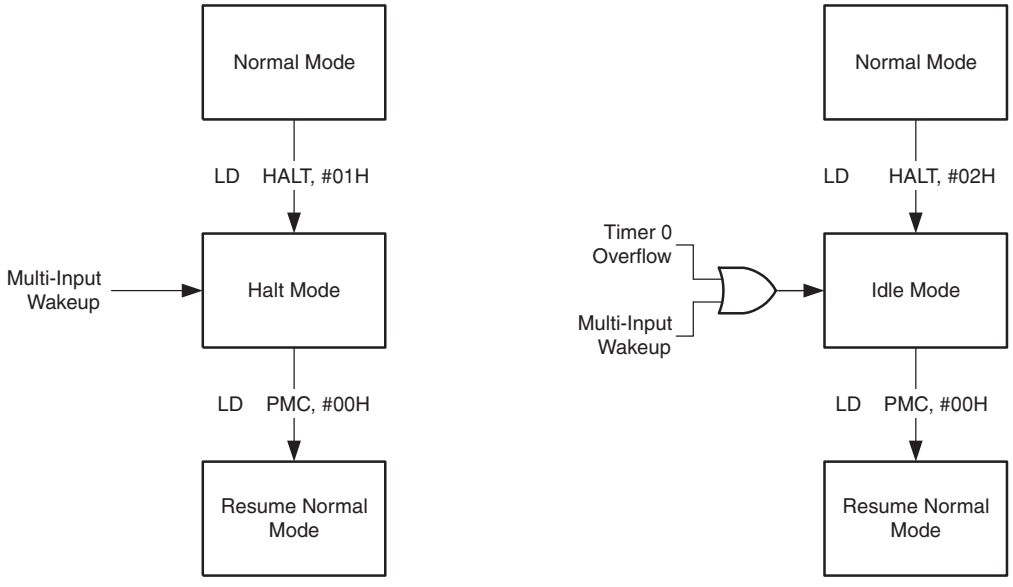
Halt Mode is a power saving feature that almost completely shuts down the device for current conservation. The device is placed into Halt Mode by setting the Halt enable bit (EHALT) of the HALT register using either the "LD M, #" or the "SBIT #, M" instructions in the software. EHALT is a write only bit and is automatically cleared upon exiting Halt Mode. When entering Halt Mode, the internal oscillator and all other on-chip systems including the Programmable Comparator (COMP) and Brown-out Reset (BOR) circuits are shut down.

The device can exit Halt Mode only by the Multi-input Wakeup (MIW) circuit.[1] Therefore, prior to entering Halt Mode, software must first configure the MIW circuit. After a wakeup from Halt Mode, a T_{HALT_REC}[2] start-up delay is initiated to allow the internal oscillator and other analog circuits to stabilize before normal device execution resumes. Immediately after exiting Halt Mode, software must clear the Power Mode Clear (PMC) register by using only the "LD M, #" instruction (see Figure 5).

Table 4. HALT Register Definition

HALT Register (addr. 0xB7)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	EIDLE	EHALT

Figure 5. Recommended Halt/Idle Flow



3.1.1 PLL Steps for Halt Mode

When using Halt Mode and the PLL in an application, software must take the appropriate steps in order to keep the integrity of the clock structure before entering and after exiting Halt since the PLL must be disabled. While in Halt Mode, all other device circuits except for the MIW are disabled. Once the PLL is disabled, all output frequencies are turned off. If the PLL is re-

4 ADC Circuit

The Analog-to-Digital Converter (ADC) Circuit extends the features of the FMS7401L by offering a 5-channel 8-bit ADC. The ADC may be programmed to convert voltages on any of the eight inputs of the analog mux, where five are multifunction input channels (ACH1-ACH5) and three are used for system calibration. The integrated ADC function offers a single cost-effective solution for applications requiring voltage, current and temperature sensing. The multifunction input channels may be configured to perform standard conversions on any of the analog input pins (G4/AIN0, G3/AIN1, G2/AIN2, G3/AIN3 or G7/AIN4). Three of the multifunction input channels may be programmed to perform ADC conversions through the internal Autozero Amplifier, Uncommitted Amplifier, and Current Source Generator for special control system and battery management applications (see Figure 6).

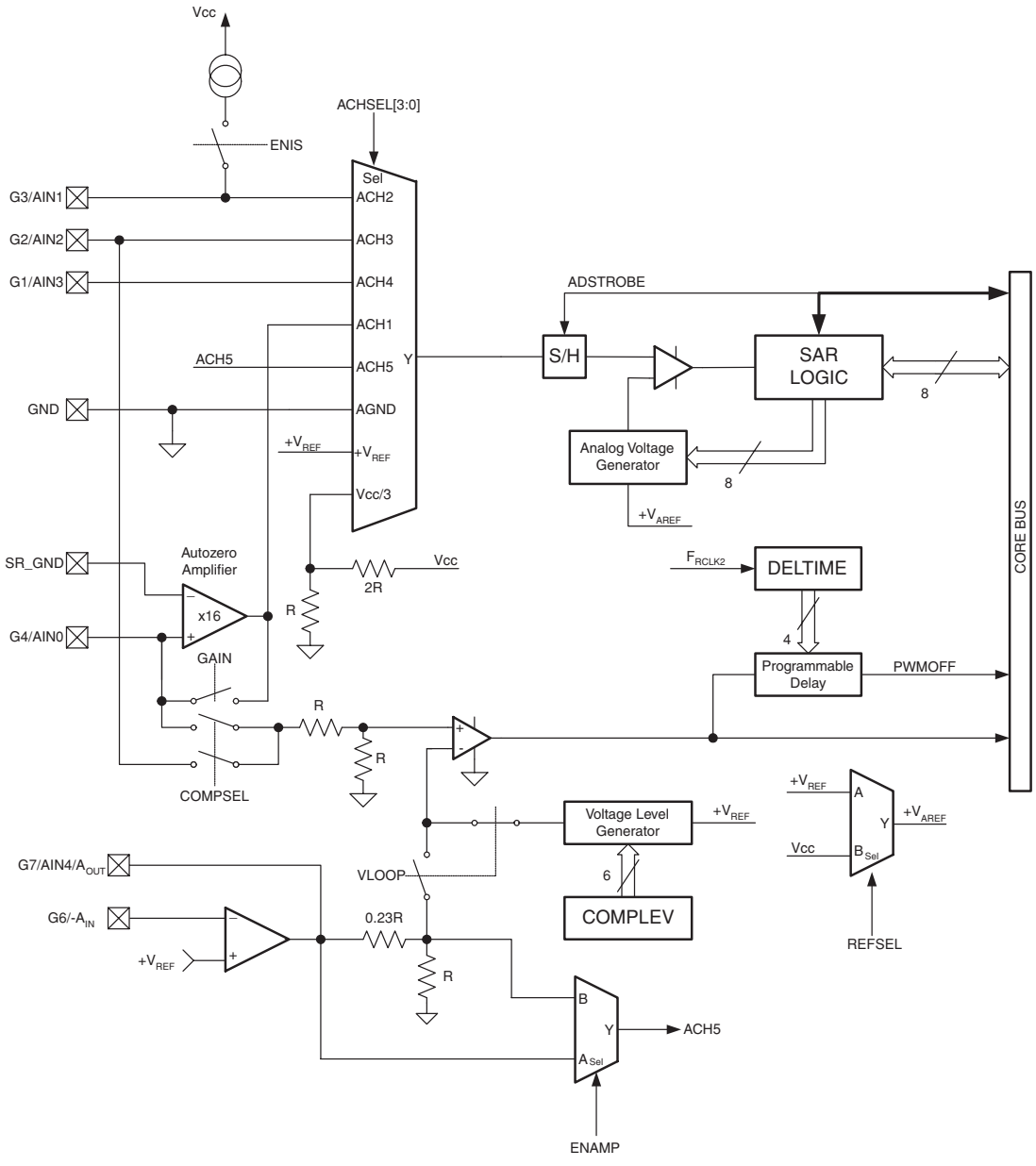
The ADC Circuit's eight analog inputs are software selectable where their analog input voltage is converted with respect to the internal ADC reference voltage (V_{AREF}). V_{AREF} may be programmed to use the internal bandgap reference voltage (V_{REF}) or Vcc as its source. By default, the ADC circuit's V_{AREF} is configured to use the internal V_{REF} as its source.[1]

The ADC performs conversions of 8-bit resolution with accuracy as defined in the Electrical Characteristics section of the datasheet. For a standard ADC conversion, the ADC circuit converts the analog input voltage in a total of 13 conversion clock cycles, and a total of 20 conversion clock cycles when performing an autozero ADC conversion. To yield a better ADC conversion accuracy, the ADC circuit may configure the ADC clock (F_{ADCLK}) to a slower frequency, lengthening the total conversion time while improving its accuracy. As part of the total conversion time, the ADC circuit completes a sample and hold phase to measure fast changing analog signals before converting the voltage. An ADC conversion can be initiated by a software command or automatically (using the gated auto-sampling mode) by the active (on) edge transition of the ADSTROBE PWM Timer 1 output.[2] If enabled, the ADC circuit offers the use of its microcontroller hardware interrupt (ADCI) triggered after each completed ADC conversion so that the microcontroller core is freed to perform other tasks.

4.1 ADC Circuit Configuration

Software must access the three memory mapped ADC registers to configure and control the ADC circuit.[3] The ADC Control 1 (ADCNTRL1) register is used to select the analog input channel and ADC reference voltage (V_{AREF}) for the conversion. In addition, it is used to initiate a conversion through software, monitor the ADC pending flag, and enable the ADC circuit's microcontroller hardware interrupt (ADCI). The ADC Control 2 (ADCNTRL2) register is used to enable the internal Autozero Amplifier, Uncommitted Amplifier, Current Source Generator, and/or ADC Auto-sampling Mode. The ADCNTRL2 register is also used to divide the ADC F_{ADCLK} clock to improve the conversion accuracy. Lastly, the ADC Data (ADATA) register is used by software to read the final converted 8-bit digital value. ADATA is a read only register and is updated automatically at the end of each ADC conversion.

Figure 6. ADC Block Diagram[4]



4.1.1 ADCNTRL1 Register

The ADCNTRL1 is an 8-bit memory map register used to configure and control the ADC circuits. Software has both read and write access to all bits of the register.

Bit 7 of the ADCNTRL1 register is the ADC pending (APND) flag and is triggered after the 8-bit converted digital value is latched to the ADATA register towards the end of the ADC conversion cycle. The APND bit may be used by software to monitor when to access ADATA or to issue microcontroller hardware interrupts (if enabled). In order for software to monitor APND, it must be cleared before the next converted value is latched in ADATA where the APND flag is set to 1.

options must be prepared prior to setting the ENDAS bit. Refer to the following ADC Gated Auto-sampling Mode section for additional details. The ADSTROBE signal is generated by the PWM Timer 1 circuit and is configured using its T1CMPB and T1RA registers. Refer to the PWM Timer 1 Circuit section of the datasheet for details regarding its operation. If ENDAS=0, the ADC circuit is configured to accept only ADC start commands issued by software when setting the ASTART bit of the ADCNTRL1 register to 1. Refer to the following ADC Conversion Modes section for additional details.

Bits 3 and 2 (ASPEED[1:0]) of the ADCNTRL2 register selects the divide factor (1, 2, 4, or 8) to slow the F_{ADCLK} clock extending the ADC conversion cycle time. In most cases, the F_{ADCLK} clock division is performed to improve the ADC conversion accuracy. Refer to the following ADC Conversion Clock Configuration section for addition details.

Bit 1 of the ADCNTRL2 register is the Current Source Generator Enable (ENIS) bit. If ENIS=0, the Current Source Generator circuit is disabled and its G3/AIN1 pin may be used as a normal I/O port or as a standard ADC conversion input through the analog ACH2 channel. If ENIS=1, the Current Source Generator circuit is enabled and its pin connection must be configured as a tri-state input bypassing the I/O circuitry.[9] If the ADC circuit is performing a conversion on the analog ACH2 input when driven by the Current Source Generator, software must avoid clearing the ENIS bit. Refer to the following Current Source Generator section for additional details.

Bit 0 (GAIN) of the ADCNTRL2 register is the autozero amplifier enable bit. If GAIN=0, the autozero amplifier with its gain 16 circuitry is disabled where its G4/AIN0 pin connections may be used as a normal I/O port. The G4/AIN0 pin may still be used as a standard ADC conversion input through the analog ACH1 channel. If GAIN =1, the autozero amplifier with its gain 16 circuitry is enabled and its G4/AIN0 pin connection must be configured as a tri-state input where G4/AIN0 is the non-inverting and SR_GND is the inverting input of the amplifier.[9] Software may write to the GAIN bit at any time; however, the actual GAIN enable signal will not change while an ADC conversion is in progress. If a read command is issued while a conversion is in progress, the current value of the GAIN bit may not necessarily reflect the actual state of the GAIN enable signal. The last value of the GAIN bit written by software at the time of the ADC conversion trigger, dictates the state of the GAIN enable signal for the triggered ADC conversion cycle. Refer to the following Autozero Amplifier section for additional details.

5 Programmable Comparator Circuit

The Programmable Comparator circuit is an analog comparator whose outputs may be monitored by software or fed into a digital delay filter used to disable the PWM Timer 1 circuit or its PWM cycle. The comparator's non-inverting input is software selectable by the COMPSEL bit of the ADCNTRL2 register.[1] If COMPSEL=0, the non-inverting input of the Programmable Comparator is the G4/AIN0 device pin. If COMPSEL=1, the non-inverting input of the Programmable Comparator is the G2/AIN2 device pin. Before enabling the Programmable Comparator circuit, the selected analog input port pin must be configured as a tri-state input bypassing the I/O circuitry.[2] The inverting input of the comparator is controlled by the Voltage Loop (VLOOP) enable bit of the comparator control (COMP) register. If VLOOP=0, the voltage loop is disabled and the inverting input of the analog comparator is configured as one of the 63 programmable voltage levels (V_{THL}, V_{THU}). If VLOOP=1, the analog comparator is set in a voltage loop configuration with the Uncommitted (Error) Amplifier output (A_{OUT}) connected to the comparator's inverting input (see Figure 9).

The Programmable Comparator circuit may be configured and controlled by software through the two 8-bit Comparator Control (COMP) and Digital Delay (DDELAY) registers. Both the Programmable Comparator and the digital delay filter must be enabled by software by setting the Comparator Enable (COMPEN) and clearing the EPWM bits of the Digital Delay (DDELAY) register. Upon a system reset, the Programmable Comparator is disabled and the digital delay filter is enabled. The COMP circuit is automatically disabled during Halt Mode. After exiting the Halt Mode, software must wait at least 10 instruction clock cycles before reading the COUT bit to ensure that the internal circuit has stabilized.

Table 8. Programmable Comparator (COMP) Control Register Bit Definitions

COMP Register (addr. 0xA0)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
CL[5:0]						VLOOP	COUT

Bit	Description
CL[5:0]	Programmable Comparator Voltage Reference Level bits. Refer to Table 9 and Table 10 for details.
VLOOP	(0) Configures the inverting input of the analog comparator as one of the 63 programmable voltage levels (V_{THL}, V_{THU}). (1) Configures the analog comparator in a voltage loop configuration with the Uncommitted Amplifier output (A_{OUT}) connected to the inverting input.
COUT	(0) G2/AIN2 or G4/AIN0 non-inverting input is less than inverting input configured by VLOOP. (1) G2/AIN2 or G4/AIN0 non-inverting input is greater than inverting input configured by VLOOP.

5.1 Programmable Comparator's Voltage Threshold Levels (VLOOP=0)

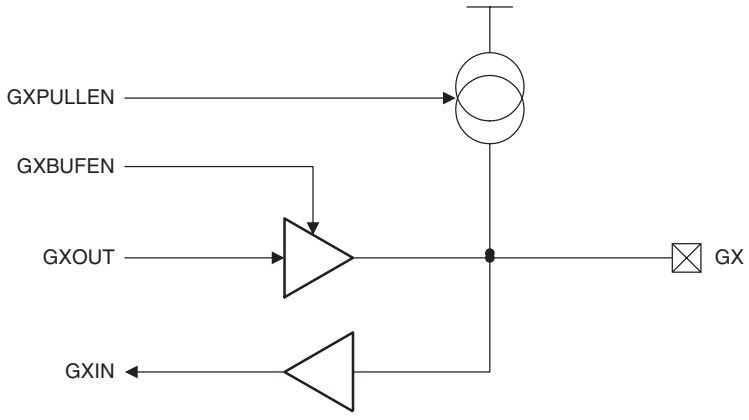
The Programmable Comparator circuit is configured to compare the G4/AIN0 or G2/AIN2 non-inverting input against the programmable voltage threshold levels on its inverting input (see Table 9 and Table 10). The comparator output (C_{OUT}) is 1 when the G4/AIN0 or G2/AIN2 input pin rises above the selected voltage threshold. As long as the input stays above the selected voltage threshold, the C_{OUT} signal will hold its state. The C_{OUT} signal will equal zero if the G4/AIN0 or G2/AIN2 input voltage falls below the programmed threshold voltage or if the Programmable Comparator circuit is disabled. Software may change the programmed threshold voltage on-the-fly as needed in the application. If the digital delay filter circuit is enabled (EPWM=0), the C_{OUT} signal is monitored for its rising edge to generate the PWMOFF signal. Refer to Figure 8 and the following Digital Delay Filter with PWMOFF Output section for addition details.

Bit 6 of the ADCNTRL2 register is the Programmable Comparator non-inverting input selection (COMPSEL) bit.[1] If COMPSEL=0, the non-inverting input of the Programmable Comparator is the G4/AIN0 device pin. If COMPSEL=1, the non-inverting input of the Programmable Comparator is the G2/AIN2 device pin. Before enabling the Programmable Comparator circuit, the selected analog input port pin must be configured as a tri-state input bypassing the I/O circuitry.[2]

8 I/O Ports

The eight I/O pins (six on the 8-pin package option) are bi-directional (see Figure 14). The bi-directional I/O pins can be individually configured by software to operate as high-impedance inputs, as inputs with weak pull-up, or as push-pull outputs. The operating state is determined by the contents of the corresponding bits in the data and configuration registers. Each bi-directional I/O pin can be used for general purpose I/O, or in some cases, for a specific alternate function determined by the on-chip hardware.

Figure 14. PORTGD Logic Diagram



8.1 I/O Registers

The I/O pins (G0–G7) have three memory mapped port registers associated with the I/O circuitry: a Port Configuration (PORTGC), Port Data (PORTGD) and Port Input (PORTGP) register.[1] PORTGC is used to configure the pins as inputs or outputs. A pin may be configured as an input by writing a 0 or as an output by writing a 1 to its corresponding PORTGC bit. If a pin is configured as an output, its PORTGD bit represents the state of the pin (1 = logic high, 0 = logic low). If the pin is configured as an input, its PORTGD bit selects whether the pin is a weak pull-up or a high-impedance input. Table 20 provides details of the port configuration options. The port configuration and data registers can both be read from or written to. Reading PORTGP returns the value of the port pins regardless of how the pins are configured. Since this device supports MIW, all input ports have Schmitt triggers.

Upon power-up, the PORTGC and PORTGD registers are initialized to 0x00. However, the G0/T1HS1 and G5/T1HS2 pins may be defaulted to the different I/O configurations defined by the default I/O configuration bits of the Initialization Register 4. Refer to Table 29 in the Device Memory section of the datasheet for details.

Table 20. I/O Register Bit Assignments

PORTGC, PORTGD, PORTGD Registers (addr. 0xB3, 0xB2, 0xB4)							
Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
G7[2]	G6[2]	G5	G4	G3	G2	G1	G0

Table 21. I/O Configuration Options

PORTGC Bit	PORTGD Bit	Port Pin Configuration
0	0	High-impedance input (tri-state input)
0	1	Input with pull-up (weak one input)
1	0	Push-pull zero output
1	1	Push-pull one output

1. Refer to Table 30 of the Device Memory section of the datasheet for the detailed memory map.
2. Available only on the 14-pin package option.

10.2 Addressing Modes

The microcontroller core has seven instruction addressing modes: inherent, immediate, direct, indirect, indexed, absolute jump and relative jump (see Table 24).

Inherent

The inherent addressing mode instructions either have no operand associated or the contents of the operand are already known to the microcontroller core. The microcontroller core then inherently knows how to execute the instruction without needing any additional information provided by additional operands.

Immediate

The immediate addressing mode instructions contain a 3-bit,[2] 8-bit or 12-bit[3] immediate field as an operand. Immediate addressing is so-named because the value needed to complete the instruction is provided immediately to the core within the instruction code. That is to say, the instruction itself dictates what the data value is to be e.g. stored in a register.

Direct

The direct addressing mode instructions contain an 8-bit address operand that directly points to a location within the data memory space. Direct addressing is so-named because the value needed to complete the instruction must be directly accessed by the core from the memory address provided by the instruction code.

Indirect

The indirect addressing mode instructions use the content in XLO, X[7:0], to address a specific location within the data memory space (0x00 – 0xFF).[4] Indirect addressing is so-named because the value needed to complete the instruction must be retrieved indirectly by the core from the address provided by the X-pointer.

Indexed

The indexed offset addressing mode instructions add an 8-bit unsigned offset value to the X-pointer yielding a new effective address to select a specific location anywhere within the memory map (both program and data memory space, 0x000-0xFFF). Indexed addressing expands the functions of indirect addressing by providing the only means to access the data stored within the program memory space.

Absolute

The absolute jump addressing mode instructions (e.g. JMP and JSR) replace the program counter with the value in the operand field. This allows jumping to any location within the program memory space.[5]

Relative

The opcode instruction field for the relative jump addressing mode instruction, JP, is calculated from the distance to the absolute program memory location in the operand addressing the next instruction to be executed. The base opcode for JP is 0xC0 where bit 5 indicates the direction within memory to jump. Bits 4 to 0 indicate the number of bytes to jump where the maximum distance is 31 bytes. If bit 5 is zero, the address for the next instruction executed is determined by subtracting the lower 5 bits of the opcode (0xC1-0xDF) from the program counter; otherwise, the lower 5 bits of the opcode (0xE0-0xFF) are added to the program counter.[6]

11 Device Memory

The FMS7401L has 64 bytes of SRAM and 64 bytes of EEPROM (data EEPROM) available for data storage. It also has 1K Byte of EEPROM (code EEPROM) memory for program storage. During the device's normal operation, software has both read and write access of SRAM and data EEPROM memories but has only read access of the code EEPROM.[1] That is, the code EEPROM is protected from unauthorized writes that can corrupt its contents during normal operating conditions. The code EEPROM can only be written to when the device is in programming mode[2] and if the write disable (WDIS) bit of Initialization Register 1 is set to 0

While in normal operating mode, the user can write to the data EEPROM array by polling the ready (R) flag of the status register then executing the appropriate instruction. If the R flag is 1, the data EEPROM block is ready to perform the next write. If the R flag is 0, the data EEPROM is busy performing a write operation. The data EEPROM array will set the R flag to 1 after completing the write operation. Attempts to read, write, or enter Halt/Idle Mode while the data EEPROM is busy (R=0) can affect the current data being written and cause the intruding read or write command to also fail.

The SRAM, data EEPROM, code EEPROM, and all other data register are memory mapped for easy access by software (see Table 30). The microcontroller core has an 11-bit X-pointer register that may be used to address data bytes within the memory map.[3] Bit 10 of the X-pointer (X[10] or XHI[1]) selects between the code and data memory space within the memory map. When X[10] is set to 1, the X-pointer selects the code memory space (addr. 0xC00 to 0xFFF) physically addresses a byte in the code EEPROM memory. Since the code EEPROM memory is 1K bytes, it requires only 10 address bits to physically address a byte of its memory. Bits 9-0 of the X-pointer (X[9:0] or {XHI[1:0],XLO[7:0]}) is the physical address of the code EEPROM used during a byte read instruction operation. When X[10] is set to 0, the X-pointer automatically addresses the data memory space (addr. 0x00 to 0xFF). Bits 9-0 of the X-pointer is the memory mapped (not physical) address for the entire data memory space (including the SRAM, data EEPROM, and all other data registers) used during a byte read/write instruction operation. In addition, when using X-pointer instructions with the "[X]" syntax, only the lower 8 bits of X are considered addressing the data memory space only. However, instructions with the "[#0,X]" syntax allow read access of the code memory space for look-up tables, etc. When using the X-pointer to address a byte in either the data or code memory space, software should load X with its 12-bit memory mapped address.

11.1 Initialization Registers

The FMS7401L has four 8-bit wide non-volatile initialization registers that are only accessible by the user in programming mode (if the memory security bits are not enabled). Each register has a corresponding shadow volatile register that is automatically updated during a reset and is used to initialize specific on-chip peripherals.

The Initialization Register 1 contains the three memory security bits, three feature enable bits, and the clock selection bit. Table 26 provides a detailed description of the Initialization Register 1. This register is defaulted to zero by the factory.

The Initialization Register 2 contains the internal oscillator frequency trim setting, F_{OSC}.[4] Prior to leaving the factory, the internal oscillator is trimmed to the appropriate frequency and the non-volatile register is pre-programmed. During a reset, the volatile shadow register (at address 0xBA) is updated with the factory programmed trim value. The shadow register associated with the Initialization Register 2 is accessibly by software during normal operation and may be written to in order to perform fine adjustments e.g. of the PWM timer outputs. If the software saved the original factory trim value, the software may restore the frequency to its original frequency.[5]

The Initialization Register 3 contains the factory calibration values for the two internal analog comparator circuits (Brown-out Reset and Programmable Comparator). The calibration is performed in order to configure the comparators to their proper levels (see Table 27). The non-volatile register is preprogrammed prior to leaving the factory.

The Initialization Register 4 contains the factory calibration value for the internal current source generator as well as the default G0/T1HS1 and G5/T1HS2 port configuration. The factory calibrates the current source generator to ensure that, if enabled, G3 can source I_{SRC}[4] of current. During the initial clock cycles of the reset sequence, the shadow register is updated configuring the G0/T1HS1 and G5/T1HS2 I/O ports to their pre-determined initial states. This offers the capability of driving G0/T1HS1 and G5/T1HS2 high within the first T_{DIO}[4] after the device is powered. The non-volatile register is pre-programmed

11.2 Memory Map

All I/O ports, peripheral registers, and core registers (except the accumulator and the program counter) are mapped into the memory space.

Table 30. Memory Mapped Registers

Address	Memory Space	Block	Contents
0x00 – 0x3F	Data	SRAM	Data RAM
0x40 – 0x7F	Data	EEPROM	Data EEPROM
0x9D	Data	ADC	ADATA register[7]
0x9F	Data	ADC	ADCNTRL1 register
0xA0	Data	ADC	ADCNTRL2 register
0xA2	Data	Prog. Comparator	DDELAY register
0xA4	Data	PWM Timer 1	PSCALE register
0xA5	Data	PWM Timer 1	DTIME register
0xA6	Data	PWM Timer 1	T1CMPALO register
0xA7	Data	PWM Timer 1	T1CMPAHI register
0xA8	Data	PWM Timer 1	T1CMPBLO register
0xA9	Data	PWM Timer 1	T1CMPBHI register
0xAA	Data	PWM Timer 1	T1RALO register
0xAB	Data	PWM Timer 1	T1RAHI register
0xAC	Data	PWM Timer 1	TMR1LO register[7]
0xAD	Data	PWM Timer 1	TMR1HI register[7]
0xAE	Data	PWM Timer 1	T1CNTRL register
0xAF	Data	MIW	WKEDG register
0xB0	Data	MIW	WKPND register
0xB1	Data	MIW	WKEN register
0xB2	Data	I/O	PORTGD register
0xB3	Data	I/O	PORTGC register
0xB4	Data	I/O	PORTGP register[7]
0xB5	Data	Timer 0	WDSVR
0xB6	Data	Timer 0	T0CNTRL register
0xB7	Data	Clock	Halt Mode register
0xB9	Data	Init Register	Initialization Register 1 (volatile)[8]
0xBA	Data	ACE Core	Internal Clock trimming register (volatile)
0xBB	Data	Init Register	Initialization Register 1[8]
0xBC	Data	Init Register	Initialization Register 2[8]
0xBD	Data	Prog. Comparator	COMP register
0xBE	Data	ACE Core	XHI register
0xBF	Data	ACE Core	XLO register
0xC0	Data	ACE Core	Power mode clear (PMC)
0xCE	Data	ACE Core	SP register
0xCF	Data	ACE Core	Status Register (SR)[9]
0xD0	Data	ACE Core	Initialization Register 3 (volatile)[8]
0xD1	Data	Init Register	Initialization Register 3[8]
0xD2	Data	Signature	Device_ID register[7]
0xD3	Data	Init Register	Initialization Register 4 (volatile)[8]
0xD4	Data	Init Register	Initialization Register 4[8]
0xC00 – 0xFF5	Program	EEPROM	Code EEPROM
0xFF6 – 0xFF7	Program	ACE Core	Timer0 Interrupt vector
0xFF8 – 0xFF9	Program	ACE Core	PWM Timer1 Interrupt vector
0xFFA – 0xFFB	Program	ACE Core	MIW Interrupt vector
0xFFC – 0xFFD	Program	ACE Core	Software Interrupt vector
0xFFE – 0xFFF	Program	ACE Core	ADC Interrupt vector

The opcode must be shifted in after Vcc settles to its nominal voltage level and before the system reset sequence (T_{RESET}) completes. Otherwise, the device will begin with its normal operation executing the instruction program residing in the code EEPROM memory. If an external reset is applied by bringing the RESET pin low, the 10-bit opcode may be shifted once RESET is released and before the system reset sequence completes.

12.2 Programming Protocol

Once the device is in programming mode, the programming protocol and commands may be issued. An externally controlled 4-wire interface consisting of a LOAD (G3) control, serial data SHIFT_IN (G4) input, serial data SHIFT_OUT (G2) output, and CLOCK (G1) pins are used to access the internal memory and registers. Communication between the external programmer and the FMS7401L is performed through a 32-bit command and response word, as described in Table 23. The serial data timing for the 4-wire interface is shown in Figure 20 and the programming protocol is shown in Figure 19. In order to exit programming mode, the device must be powered down or an external reset must be applied.

12.2.1 Byte Write Sequence

After the external programmer puts the FMS7401L into programming mode, the LOAD pin must be set to Vcc before serially shifting the first 32-bit command word using the SHIFT_IN and CLOCK signals. By definition, bit 31 of the command word must be shifted first followed by all other bits. With each bit of the 32-bit write command word shifted, the device shifts out a bit of the 32-bit response word from the previous command through the SHIFT_OUT pin. The external programmer may sample SHIFT_OUT after T_{ACCESS} from the rising edge of CLOCK. The serial response word sent immediately after entering programming mode may contain indeterminate data.

After all 32 bits of the command word are shifted, the external programmer must set the LOAD signal to 0V and apply two clock pulses to the CLOCK signal, as shown in Figure 19, to complete the program cycle. Once the LOAD signal is brought low, the SHIFT_OUT pin acts as the handshaking signal between the device and external programmer hardware. When executing the write command, the device sets SHIFT_OUT low by the time the external programmer has issued the second rising edge of CLOCK informing the external programmer that the memory write is in progress. The external programmer must wait T_{READY} for SHIFT_OUT to return high before returning the LOAD signal to Vcc to initiate the next command cycle.

12.2.2 Page Write Sequence

Page mode is a convenient and fast way to program the code EEPROM memory. In this mode, 16 bytes of data are written using a single write command followed by a stream of data bytes. Only full pages can be written in page mode where the address in the command word points to the beginning of a page.[4] After all 16 bytes of data has been shifted, the data will be written at once speeding up the total write time by a factor of 16 compared to byte mode programming. Figure 21 shows the page mode programming protocol.

Page mode's 32-bit write command word is similar to a byte write command except that bit 31 must be set to 1 in order to enable page mode. The address in the page-write command word (bits 17 to 8) must select the page to program the 16 bytes of data (the page address is a multiple of the page size: 0x000, 0x010, 0x020, etc.). The first byte of the page to program must be placed in the last 8 bits of the page-write command word (bits 7 to 0). All other bytes in the page must immediately follow after the initial page-write command has been entered.

The LOAD pin must be set to Vcc before serially shifting in the 32-bit page-write command word using the SHIFT_IN and CLOCK signals. By definition, bit 31 of the command word must be shifted first followed by all other bits. After all 32 bits of the command word are shifted, the external programmer must set the LOAD signal to 0V and apply two clock pulses to the CLOCK signal to latch the first byte of the page in its temporary data buffer. The LOAD signal must be returned to Vcc in order for the external programmer to shift the second byte of the page into the device (without repeating the command word). Once all 8 bits of the byte are shifted, the LOAD signal must again be set to 0V followed by two clocks pulses of the CLOCK signal in order to latch byte into its temporary data buffer. This process must be repeated until all 16 bytes are loaded into their data buffers.

While the 16^{th} byte of data is being latched, the actual write to the code EEPROM page, selected by the address in the 32-bit page-write command word, occurs. Once the LOAD signal is brought low, the SHIFT_OUT pin acts as the handshaking signal

between the device and external programmer hardware. The device sets SHIFT_OUT low during a page-write command by the time the external programmer has issued the second rising edge of CLOCK informing the programmer that the memory write is in progress. The external programmer must wait T_{READY} for SHIFT_OUT to return high before returning the LOAD signal to Vcc to initiate the next command cycle.

12.2.3 Byte Read Sequence

The external programmer can only perform memory reads a byte at a time. Before shifting each new command, the external programmer must set the LOAD signal to Vcc. By definition, bit 31 of the command word must be shifted first followed by all other bits. With each bit of the 32-bit read command word shifted, the device shifts out a bit of the 32-bit response word from the previous command through the SHIFT_OUT pin. The external programmer must sample SHIFT_OUT after T_{ACCESS} from the rising edge of CLOCK. The serial response word sent immediately after entering programming mode may contain indeterminate data.

After all 32 bits of the read command word are shifted, the external programmer must set the LOAD signal to 0V and apply two clock pulses to the CLOCK signal, as shown in Figure 19, to complete the read cycle. At the rising edge of the second clock pulse, the data read from the address provided in the read command word is latched into the lower 8-bits of its response word. Once LOAD is returned to Vcc, the next 32-bit command word may be shifted while the response word to the previous read command is shifted out with the data read from memory. If the last read command has been shifted, a dummy read command must be shifted to collect the last response word containing the last data byte read.

Table 33. 32-Bit Command and Response Word

Bit Number	Input Command Word	Output Response Word
Bit 31	Set to 1 to enable page mode memory access otherwise 0 for byte mode.	Same as the input command word.
Bit 30	Must be set to 0.	Same as the input command word.
Bit 29	Set to 1 to access the data memory space (data EEPROM or initialization registers) otherwise 0.	Same as the input command word.
Bit 28	Set to 1 to access the code EEPROM otherwise 0.	Same as the input command word.
Bits 27 – 25	Must be set to 0.	Same as the input command word.
Bit 24	Set to 1 to perform a read or 0 to perform a write.	Same as the input command word.
Bit 22	Set to 1 to perform a program memory erase otherwise 0.	Same as the input command word.
Bits 23, 21 – 18	Must be set to 0.	Same as the input command word.
Bits 17 – 8	Lower 10-bits of the memory mapped address byte to read/write or first byte of the page to write.	Same as the input command word.
Bits 7 – 0	Data to be programmed if a write command or all zeros if a read command.	Same as the previous input write command word or the data read after an input read command word.

AC Electrical Characteristics

All measurements are valid for T_A=+25°C unless otherwise stated.

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
F_{OSC}[5]	Internal oscillator frequency (factory trim set-point)	Vcc=3.3V	1.96	2.00	2.04	MHz
Δckv	Internal oscillator frequency voltage variation		-0.5		+0.5	%
Δckt	Internal oscillator frequency temperature variation	Vcc=3.3V -40°C to 85°C	-3		+3	%
		Vcc=3.3V -40°C to 125°C	-4		+4	%
F_{PLL}	PLL input reference frequency			2.00		MHz
T_{PLL_LOCK}[3]	PLL Lock time	-40°C to 125°C			60	µS
T_{EEW}	EEPROM Writing time			3.7	5	mS
T_{RESET}[3]	System reset time	-40°C to 125°C	2.5	3.7	4.7	mS
T_{DIO}[3]	T1HS1 and T1HS2 default I/O configuration settling time			40		µS
T_{HALT_REC}[3]	Internal device start time after exiting from Halt where $F_{ICLK} = F_{OSC}$[2]	-40°C to 125°C		5	7	µS

Brown-out Reset (BOR) Electrical Characteristics

All measurements are valid for T_A=+25°C unless otherwise stated.

Parameter	Conditions	Min.	Typ.	Max.	Units
BOR Trigger Vcc Threshold Level		2.64	2.71	2.78	V
	-40°C to +85°C	2.60		2.83	V
	-40°C to 125°C	2.59		2.83	V

Programmable Comparator Electrical Characteristics

All measurements are valid for T_A=+25°C unless otherwise stated.

Parameter	Conditions	Min.	Typ.	Max.	Units
All 32 thresholds (V_{THU})		-6%	V_{THU}	+6%	V
Upper Range (0.45V to 2.0V)	-40°C to 125°C	-8%	V_{THU}	+8%	V
All 31 thresholds (V_{THL})		V_{THU} – 30mV	V_{THL}	V_{THU} + 30mV	V
Lower Range (0.03V to 0.43V)	-40°C to 125°C	V_{THU} – 35mV	V_{THL}	V_{THU} + 35mV	V
Comparator Response Time[3]	2mV overdrive		359		nS
	5mV overdrive		173		nS
	10mV overdrive		95		nS

Figure 22. Internal Oscillator Frequency (F_{OSC}) vs. Temperature

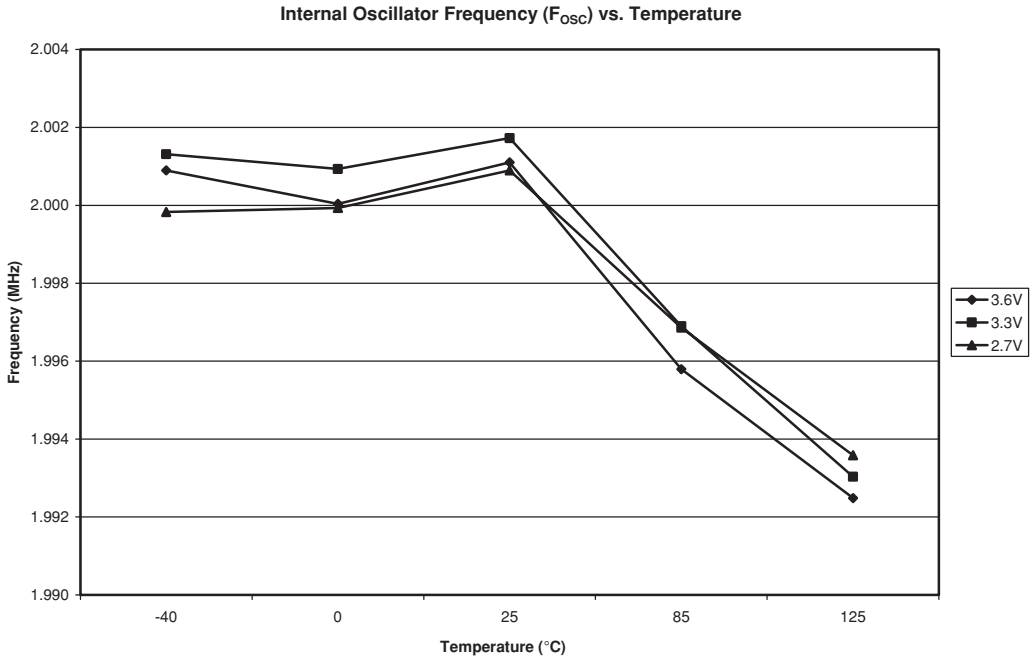


Figure 23. Icc Active vs. Temperature (no PLL or data EEPROM writes)

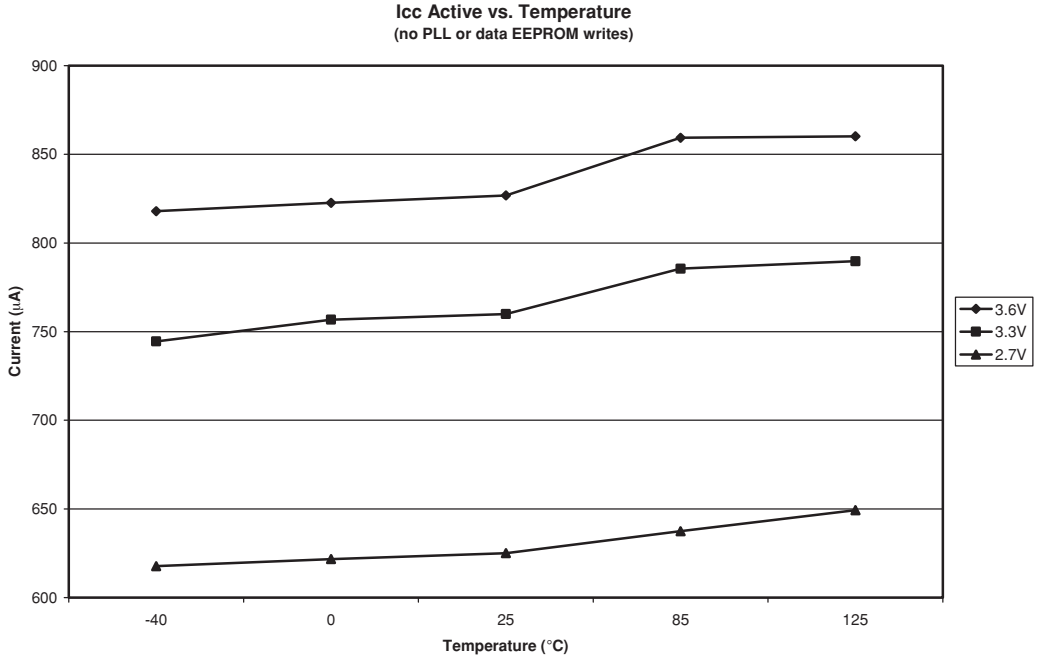


Figure 32. V_{OH} vs. I_{OH} @ 25°C (G0, G5)

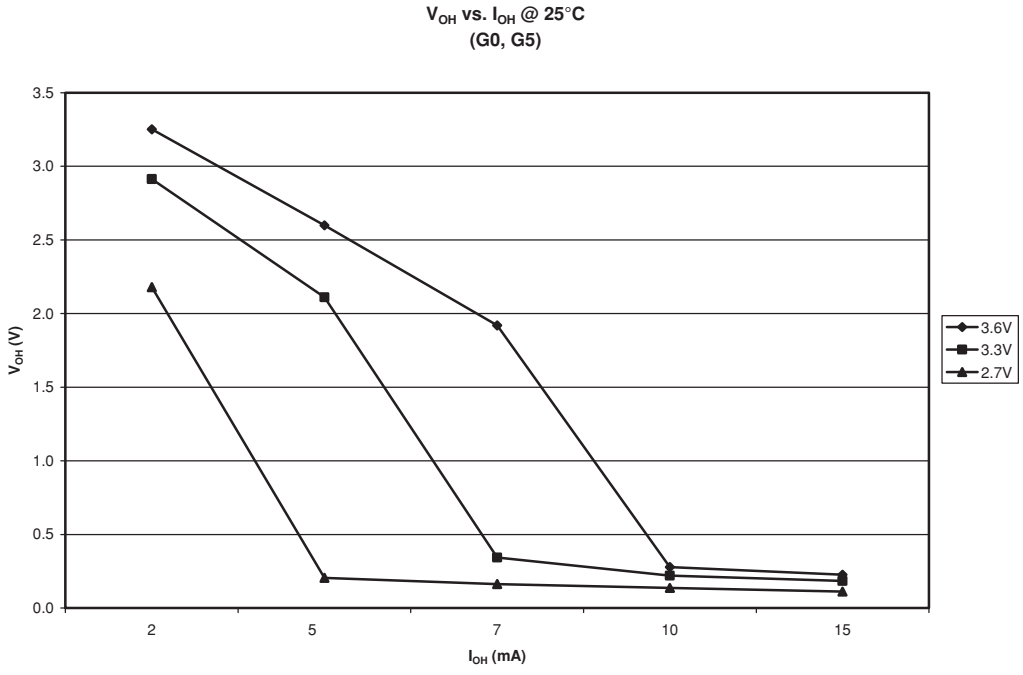


Figure 33. BOR Level vs. Temperature

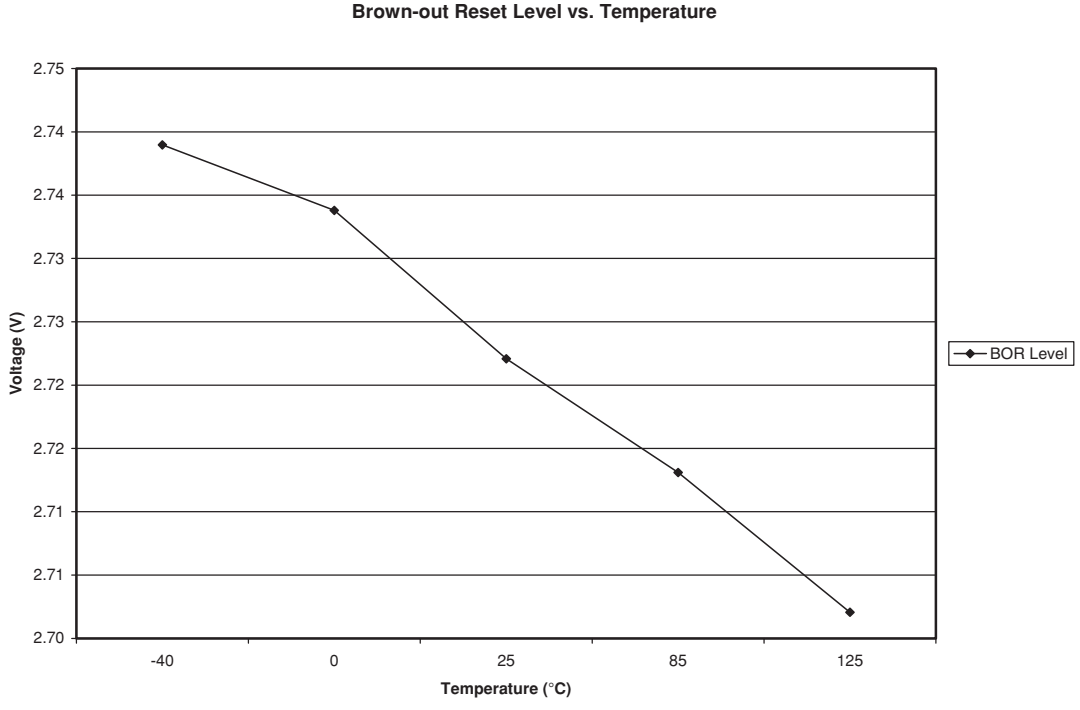


Figure 34. Programmable Comparator Voltage Level vs. Temperature

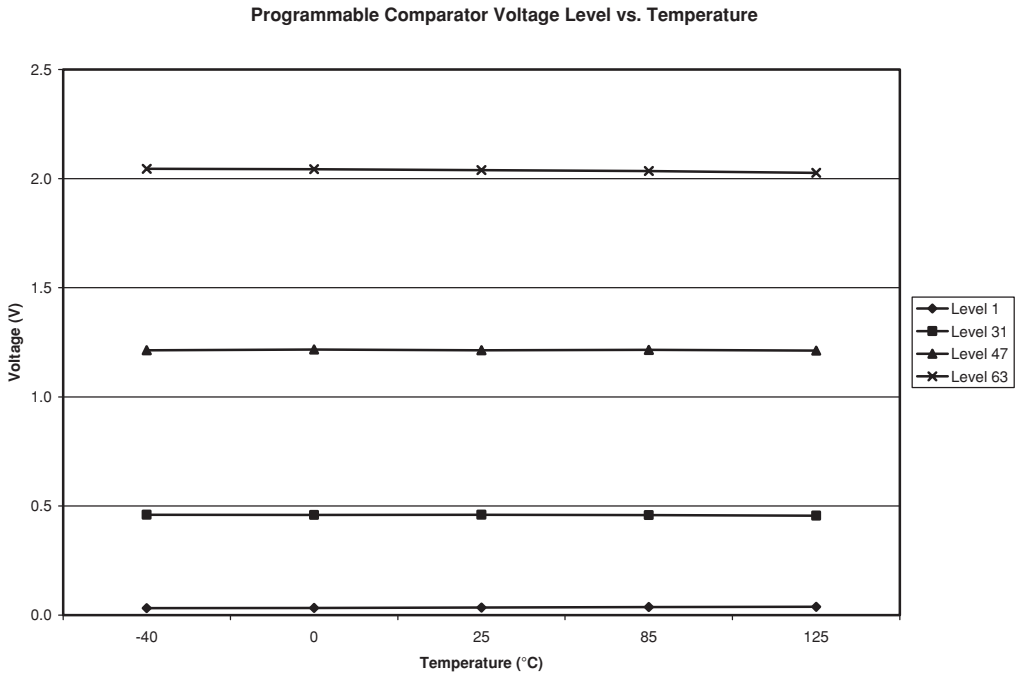
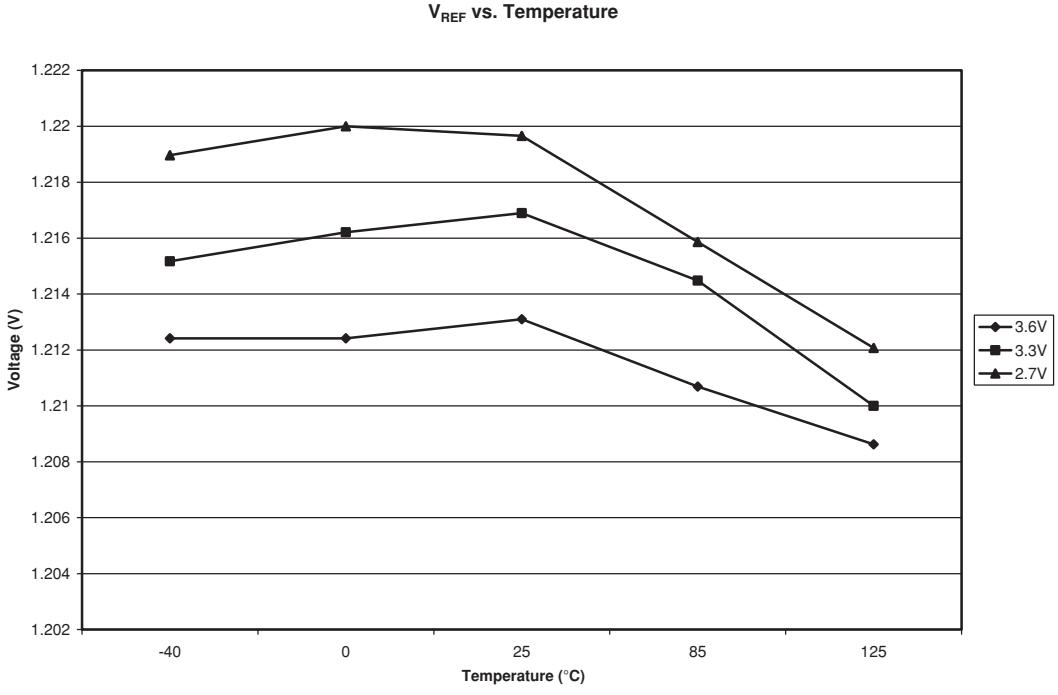


Figure 35. V_{REF} vs. Temperature

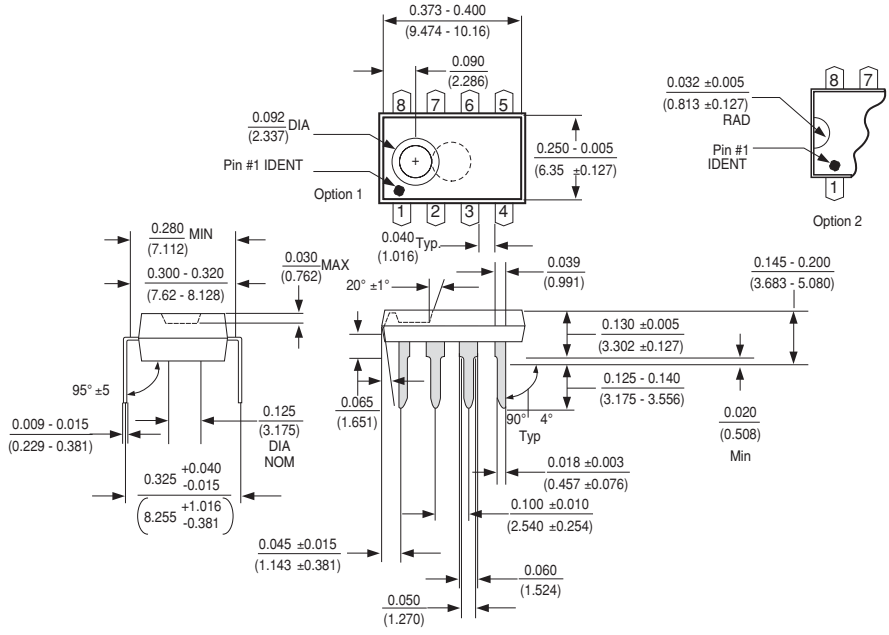


Ordering Information

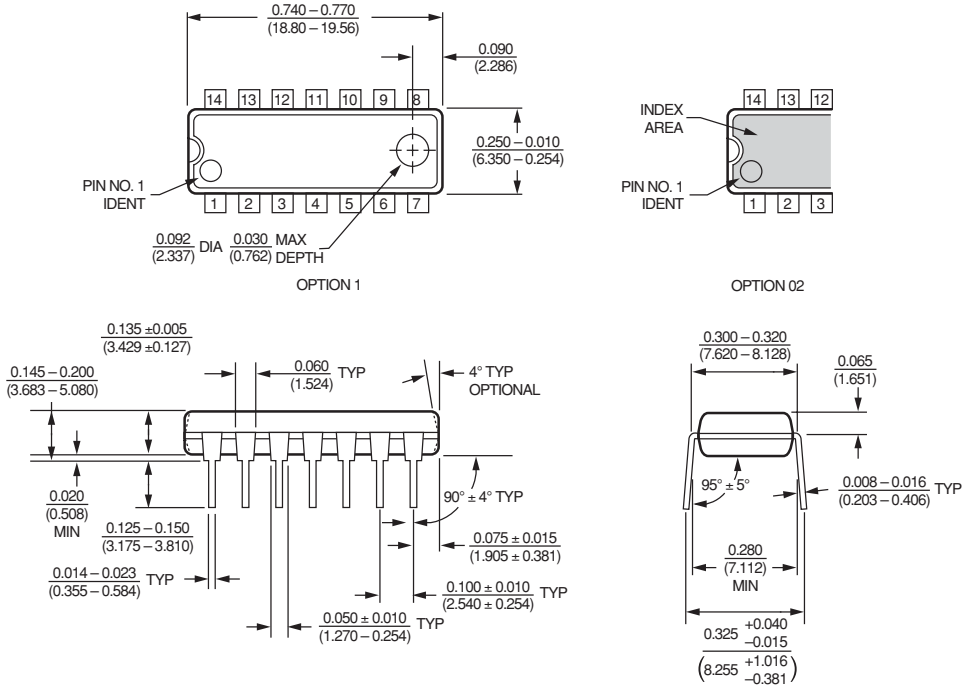
FSID	Package	Supply Voltage	Temperature Range	Packaging Option	
				Method	Qty
FMS7401LEN	PDIP8	2.7V to 3.6V	-40°C to 85°C	Rail	40
FMS7401LVN	PDIP8	2.7V to 3.6V	-40°C to 125°C	Rail	40
FMS7401LEN14	PDIP14	2.7V to 3.6V	-40°C to 85°C	Rail	25
FMS7401LVN14	PDIP14	2.7V to 3.6V	-40°C to 125°C	Rail	25
FMS7401LEM8X	SOIC8	2.7V to 3.6V	-40°C to 85°C	Tape Reel	2500
FMS7401LEMX	SOIC14	2.7V to 3.6V	-40°C to 85°C	Tape Reel	2500
FMS7401LEMT8X	TSSOP8	2.7V to 3.6V	-40°C to 85°C	Tape Reel	2500
FMS7401LEMTX	TSSOP14	2.7V to 3.6V	-40°C to 85°C	Tape Reel	2500

Physical Dimensions††

8-Pin PDIP



14-Pin PDIP



††Dimensions are in inches (millimeters) unless otherwise noted.