EEE 204
Analog to Digital Converters

Asst. Prof. Dr. Mahmut AYKAÇ

CHAPTER 9

Analog to Digital Converters

ADCs allow embedded computers to convert analog voltages into digital values so that they can be monitored in real-time and trigger responses. The MSP430F5529 contains a 12-bit ADC core that can perform conversions on up to 16 user-selected inputs.

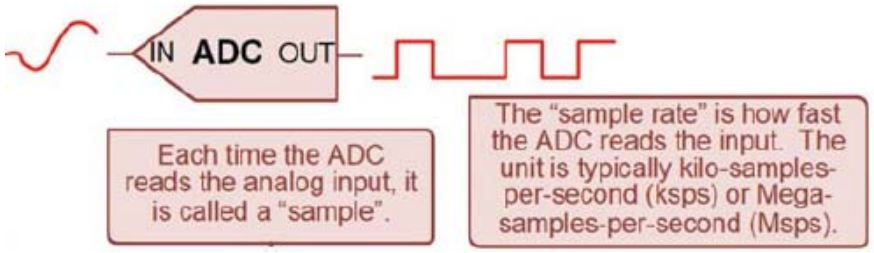
An analog-to-digital converter (ADC or A2D) is a circuit that takes in an analog voltage and produces a digital representation of its value. ADCs consist of two stages, a **sample-and-hold stage** and a **conversion stage**.

When the sample-and-hold stage is activated, it makes momentary contact with the input signal and allows it to charge a capacitor within the sample circuit. The goal of this momentary contact is to charge the capacitor to the same voltage as the input. The sample-and-hold circuitry is designed so that this can be accomplished very quickly so that it can disconnect from the input signal as soon as possible to avoid altering its signal integrity. The action of duplicating the voltage value of the input signal is called a **sample.** A capacitor is able to hold this voltage for a brief amount of time, thus providing the **hold** functionality of the sample-and-hold circuit. The conversion stage then produces a digital value that represents the analog value held on the capacitor.

Analog to Digital Converters

The conversion of the analog sample into a digital number is called **digitizing, quantizing, or discretizing** the value. All of these terms refer to how a continuous analog signal is converted into a set of discrete digital numbers. An ADC has an analog input range that it digitizes across. The range is provided to the ADC using two inputs, the voltage reference high (V_{R+}) and the voltage reference low (V_{R-}).

ADCs can be configured to sample periodically. The speed at which samples are collected is called the **sample rate** and typically has units of **kilo-samples-per-second (ksps)** or **Mega-samples-per-second (Msps)**. If the sample rate is sufficiently faster than the frequency of the input signal, then an accurate representation of the input signal can be reconstructed using the samples.



Analog to Digital Converters

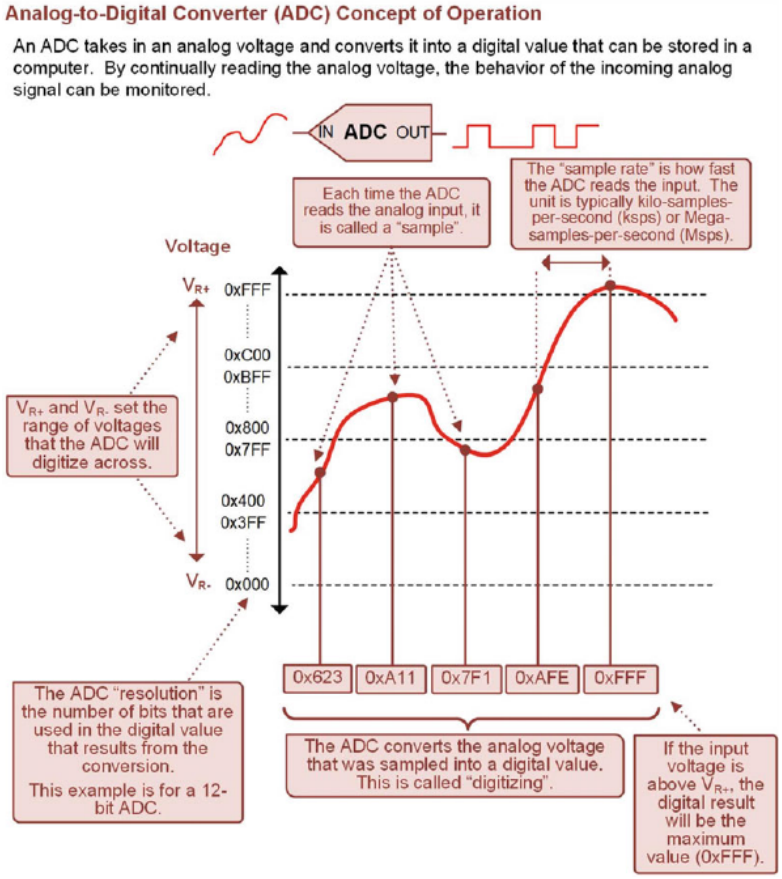
Figure shows the concept of operation of a basic ADC.

The **resolution** of an ADC refers to how many bits wide (n) the digital output value is. Common resolution values in MCUs are 8-bit, 10-bit, and 12-bit. The larger the resolution, the more accurate the conversion of the input signal is.

The **precision** of an ADC is the smallest voltage that the LSB of the digitized number can represent. The precision is found by dividing the input voltage range by 2^n (i.e., $(V_{R+} - V_{R-})/2^n$).

The actual analog voltage that the ADC result represents can be found by multiplying the result, the number that is read from the ADC, (N_{ADC}) by the resolution.

$$\boldsymbol{V_{analog} = N_{ADC} \cdot \text{precision}}$$



The **accuracy** of an ADC states how close its digital output is to the actual input signal voltage. An ADC will only ever be able to get within ±½LSB of the original input signal due to the way that the input range is divided into discrete values that are 1 LSB apart from each other. The accuracy of a sample gives a range of voltages that the final digital output lies within

$$\boldsymbol{V_{analog} = N_{ADC} \cdot \text{precision} \pm ½\text{LSB}}$$

$$\boldsymbol{Accuracy = \pm ½\text{LSB}}$$

Analog to Digital Converters (Examples)

Analog-to-Digital Converter (ADC) Parameters

The **resolution** of an ADC represents how many bits (n) are in the digital output.

The **precision** of an ADC represents how much voltage the LSB of the digitized value is worth. This depends on the number of bits in the ADC (n) and the range of voltages that the ADC digitizes across. The range of inputs is dictated by the maximum (V_{R+}) and minimum voltage references (V_{R-}) of the ADC.

$$\text{Precision} = \frac{V_{R+} - V_{R-}}{2^n}$$

Example: What is the precision of a 12-bit ADC digitizing between V_{R+} = +3.4v and V_{R-} = 0v?

$$\text{Precision} = \frac{3.4 - 0}{2^{12}} = 830\ \mu V$$

Example: For the ADC configuration in the above example, a conversion produced a result of 0x08A5. What is the analog value that was read?

$$V_{analog} = 0x08A5 \cdot 830\ \mu V$$

$$V_{analog} = 2213_{10} \cdot 830\ \mu V$$

$$V_{analog} = +1.836962\ V$$

The **accuracy** of an ADC can only ever be +/- ½ LSB of the result. The accuracy represents the range of voltages that any N_{ADC} lies within.

Accuracy = +/- ½ LSB

Example: For the V_{analog} voltage read in the above example, what is the accuracy of the result?

$$V_{analog} = +1.836962 +/- 415\ \mu V$$

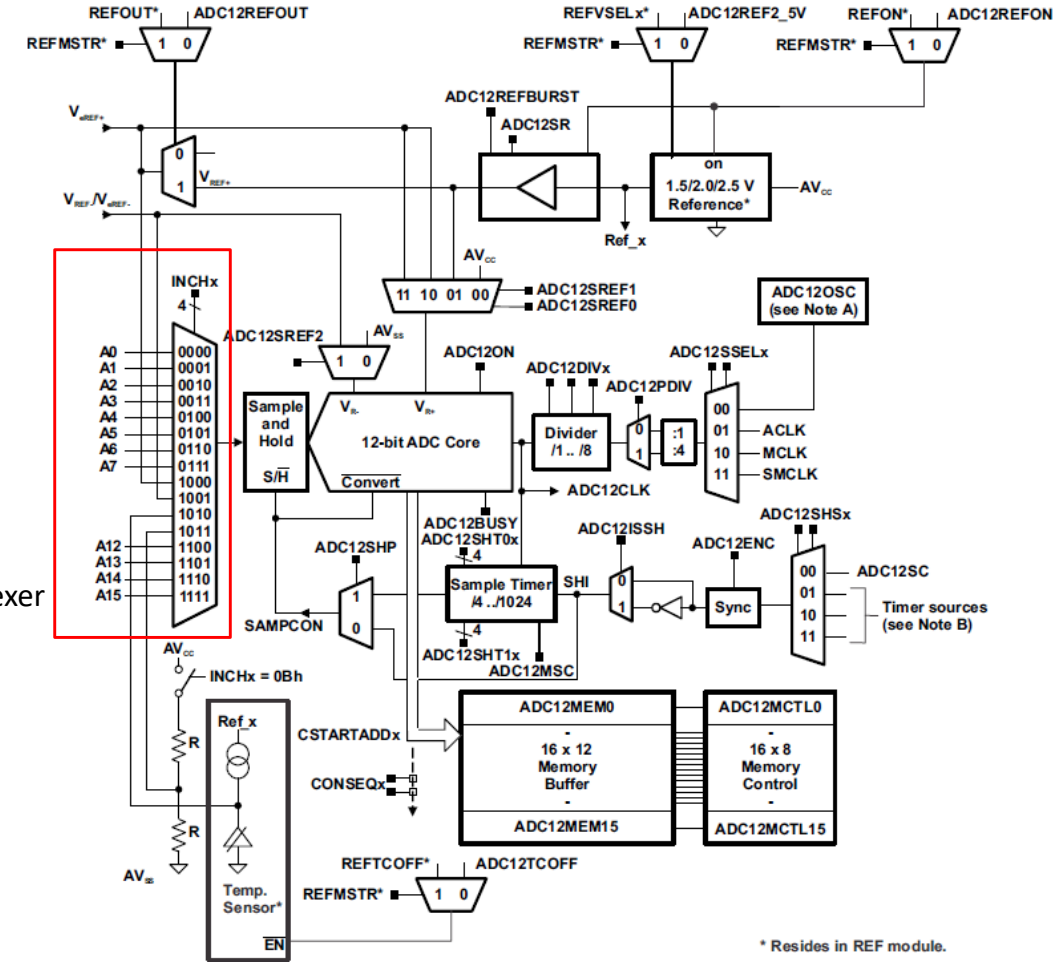
This means that the result 0x08A5 could be any voltage between **+1.836548** and **+1.837378**.

Remember that the precision is equal to LSB!

ADC Operation on the MSP430F5529

TheMSP430F5529 contains an ADC core with selectable resolution (8-bit, 10-bit, or 12-bit). The ADC can be driven with 1 of 16 inputs that are selected using an analog multiplexer that sits in front of the ADC core. The ADC clock source is selectable with two stages of programmable dividers/prescalers. The voltage range of the ADC is also programmable with options of using the power supply (Vcc) and GND (Vss), input pins, or a variety of internally generated voltages.

Analog Multiplexer



ADC Operation on the MSP430F5529

The basic use model for the ADC peripheral is that it is first configured using a set of registers, then the conversion is started by the user (or by the successful completion of a prior conversion), and then the result of the conversion can be read from the ADC's conversion memory register. Flags can be used to track the status of the conversion and trigger interrupts to react to various states of the conversion (i.e., conversion complete). The complete list of ADC registers on the MSP430F5529 MCU is as follows:

- ADC Control 0 (ADC12CTL0) Register
- ADC Control 1 (ADC12CTL1) Register
- ADC Control 2 (ADC12CTL2) Register
- ADC Memory Control (ADC12MCTL0) Register
- ADC Conversion Memory (ADC12MEM0) Register
- ADC Interrupt Enable (ADC12IE) Register
- ADC Interrupt Flag (ADC12IFG) Register
- ADC Interrupt Vector (ADC12IV) Register

- **The ADC is configured using four main registers: ADC12CTL0, ADC12CTL1, ADC12CTL2 and ADC12MCTL0.** We will go through the settings in each of these registers that are needed to get a basic ADC program working. **The ADC12CTL0 register contains the settings for the number of ADC12CLK cycles to use during the conversion (ADC12SHTxx), how the ADC is triggered (ADC12MSC), turning the ADC on (ADC12ON), enabling the conversion (ADC12ENC), and starting a conversion (ADC12SC).** The ADC12ENC and ADC12SC bits are used to start a conversion by asserting them simultaneously. All other settings are done in the initialization portion of the program.

ADC Control Register (ADC12CTL0)

15	14	13	12	11	10	9	8
ADC12SHT1x				ADC12SHT0x			
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADC12MSC	ADC12REF2_5V	ADC12REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ADC12ENC	ADC12SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Bit	Field	Type	Reset	Description
15-12	ADC12SHT1x	RW	0h	ADC12_A sample-and-hold time. These bits define the number of ADC12CLK cycles in the sampling period for registers ADC12MEM8 to ADC12MEM15.
11-8	ADC12SHT0x	RW	0h	ADC12_A sample-and-hold time. These bits define the number of ADC12CLK cycles in the sampling period for registers ADC12MEM0 to ADC12MEM7. 0000b = 4 ADC12CLK cycles 0001b = 8 ADC12CLK cycles 0010b = 16 ADC12CLK cycles 0011b = 32 ADC12CLK cycles 0100b = 64 ADC12CLK cycles 0101b = 96 ADC12CLK cycles 0110b = 128 ADC12CLK cycles 0111b = 192 ADC12CLK cycles 1000b = 256 ADC12CLK cycles 1001b = 384 ADC12CLK cycles 1010b = 512 ADC12CLK cycles 1011b = 768 ADC12CLK cycles 1100b = 1024 ADC12CLK cycles 1101b = 1024 ADC12CLK cycles 1110b = 1024 ADC12CLK cycles 1111b = 1024 ADC12CLK cycles
7	ADC12MSC	RW	0h	ADC12_A multiple sample and conversion. Valid only for sequence or repeated modes. 0b = The sampling timer requires a rising edge of the SHI signal to trigger each sample-and-convert. 1b = The first rising edge of the SHI signal triggers the sampling timer, but further sample-and-conversions are performed automatically as soon as the prior conversion is completed.
6	ADC12REF2_5V	RW	0h	ADC12_A reference generator voltage. ADC12REFON must also be set. In devices with the REF module, this bit is only valid if the REFMSTR bit of the REF module is set to 0. In the F54xx devices (non-A), the REF module is not available. 0b = 1.5 V 1b = 2.5 V
5	ADC12REFON	RW	0h	ADC12_A reference generator on. In devices with the REF module, this bit is only valid if the REFMSTR bit of the REF module is set to 0. In the F54xx devices (non-A), the REF module is not available. 0b = Reference off 1b = Reference on
4	ADC12ON	RW	0h	ADC12_A on 0b = ADC12_A off 1b = ADC12_A on
3	ADC12OVIE	RW	0h	ADC12MEMx overflow-interrupt enable. The GIE bit must also be set to enable the interrupt. 0b = Overflow interrupt disabled 1b = Overflow interrupt enabled
2	ADC12TOVIE	RW	0h	ADC12_A conversion-time-overflow interrupt enable. The GIE bit must also be set to enable the interrupt. 0b = Conversion time overflow interrupt disabled 1b = Conversion time overflow interrupt enabled
1	ADC12ENC	RW	0h	ADC12_A enable conversion 0b = ADC12_A disabled 1b = ADC12_A enabled
0	ADC12SC	RW	0h	ADC12_A start conversion. Software-controlled sample-and-conversion start. ADC12SC and ADC12ENC may be set together with one instruction. ADC12SC is reset automatically. 0b = No sample-and-conversion-start 1b = Start sample-and-conversion

Mostly required for; ADC clock cycles (ADC12SHTxx)
ADC Turning On (ADC12ON)
ADC Conversion Enable (ADC12ENC)
ADC Sampling and Conversion start (ADC12SC)

ADC Control Register (ADC12CTL1)

15	14	13	12	11	10	9	8
ADC12CSTARTADDx				ADC12SHSx		ADC12SHP	ADC12ISSH
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ADC12DIVx			ADC12SSELx		ADC12CONSEQx		ADC12BUSY
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-(0)

Bit	Field	Type	Reset	Description
15-12	ADC12CSTARTADDx	RW	0h	ADC12_A conversion start address. These bits select which ADC12_A conversion-memory register is used for a single conversion or for the first conversion in a sequence. The value of CSTARTADDx is 0 to 0Fh, corresponding to ADC12MEM0 to ADC12MEM15.
11-10	ADC12SHSx	RW	0h	ADC12_A sample-and-hold source select 00b = ADC12SC bit 01b = Timer source (see device-specific data sheet for exact timer and locations) 10b = Timer source (see device-specific data sheet for exact timer and locations) 11b = Timer source (see device-specific data sheet for exact timer and locations)
9	ADC12SHP	RW	0h	ADC12_A sample-and-hold pulse-mode select. This bit selects the source of the sampling signal (SAMPCON) to be either the output of the sampling timer or the sample-input signal directly. 0b = SAMPCON signal is sourced from the sample-input signal. 1b = SAMPCON signal is sourced from the sampling timer.
8	ADC12ISSH	RW	0h	ADC12_A invert signal sample-and-hold 0b = The sample-input signal is not inverted. 1b = The sample-input signal is inverted.
7-5	ADC12DIVx	RW	0h	ADC12_A clock divider 000b = Divide by 1 001b = Divide by 2 010b = Divide by 3 011b = Divide by 4 100b = Divide by 5 101b = Divide by 6 110b = Divide by 7 111b = Divide by 8
4-3	ADC12SSELx	RW	0h	ADC12_A clock source select 00b = ADC12OSC (MODCLK) 01b = ACLK 10b = MCLK 11b = SMCLK **In our board, MCLK=SMCLK=1MHz**
2-1	ADC12CONSEQx	RW	0h	ADC12_A conversion sequence mode select 00b = Single-channel, single-conversion 01b = Sequence-of-channels 10b = Repeat-single-channel 11b = Repeat-sequence-of-channels
0	ADC12BUSY	R	0h	ADC12_A busy. This bit indicates an active sample or conversion operation. 0b = No operation is active. 1b = A sequence, sample, or conversion is active.

Mostly required for; ADC sampling source (ADC12SHP)
ADC clock divider (ADC12DIVx, x=0 as default)
ADC clock source select (ADC12SSELx)

ADC Control Register (ADC12CTL2)

15	14	13	12	11	10	9	8
Reserved							ADC12PDIV
r-0	r-0	r-0	r-0	r-0	r-0	r-0	rw-0
7	6	5	4	3	2	1	0
ADC12TCOFF	Reserved	ADC12RES		ADC12DF	ADC12SR	ADC12REFOUT	ADC12REFBURST
rw-(0)	r-0	rw-(1)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Bit	Field	Type	Reset	Description
15-9	Reserved	R	0h	Reserved. Always reads as 0.
8	ADC12PDIV	RW	0h	ADC12_A predivider. This bit predivides the selected ADC12_A clock source. 0b = Predivide by 1 1b = Predivide by 4
7	ADC12TCOFF	RW	0h	ADC12_A temperature sensor off. If the bit is set, the temperature sensor turned off. This is used to save power. In devices with the REF module, this bit is only valid if the REFMSTR bit of the REF module is set to 0. In the F54xx devices (non-A), the REF module is not available. 0b = Temperature sensor on 1b = Temperature sensor off
6	Reserved	R	0h	Reserved. Always reads as 0.
5-4	ADC12RES	RW	2h	ADC12_A resolution. This bit defines the conversion result resolution. 00b = 8 bit (9 clock cycle conversion time) 01b = 10 bit (11 clock cycle conversion time) 10b = 12 bit (13 clock cycle conversion time) 11b = Reserved
3	ADC12DF	RW	0h	ADC12_A data read-back format. Data is always stored in the binary unsigned format. 0b = Binary unsigned. Theoretically, the analog input voltage -VREF results in 0000h, the analog input voltage +VREF results in 0FFFh. 1b = Signed binary (twos complement), left aligned. Theoretically, the analog input voltage -VREF results in 8000h, the analog input voltage +VREF results in 7FF0h.
2	ADC12SR	RW	0h	ADC12_A sampling rate. This bit selects the reference buffer drive capability for the maximum sampling rate. Setting ADC12SR reduces the current consumption of the reference buffer. 0b = Reference buffer supports up to approximately 200 ksps. 1b = Reference buffer supports up to approximately 50 ksps.
1	ADC12REFOUT	RW	0h	Reference output. In devices with the REF module, this bit is only valid if the REFMSTR bit of the REF module is set to 0. In the F54xx devices (non-A), the REF module is not available. 0b = Reference output off 1b = Reference output on
0	ADC12REFBURST	RW	0h	Reference burst 0b = Reference buffer on continuously 1b = Reference buffer on only during sample-and-conversion

- The ADC12CTL2 register contains settings for the first clock divider stage (ADC12PDIV), the resolution of the ADC (ADC12RES), the data format of the result (ADC12DF), and the range for the anticipated sample rate (ADC12SR). **Note that the default setting for ADC12RES is 01 (10-bit).**

ADC Conversion Memory Control Register (ADC12MCTL0)

7	6	5	4	3	2	1	0
ADC12EOS	ADC12SREFx			ADC12INCHx			
rw	rw	rw	rw	rw	rw	rw	rw

where x=0

Bit	Field	Type	Reset	Description
7	ADC12EOS	RW	0h	End of sequence. Indicates the last conversion in a sequence. 0b = Not end of sequence 1b = End of sequence
6-4	ADC12SREFx	RW	0h	Select reference 000b = V_{R+} = AVCC and V_{R-} = AVSS 001b = V_{R+} = VREF+ and V_{R-} = AVSS 010b = V_{R+} = VeREF+ and V_{R-} = AVSS 011b = V_{R+} = VeREF+ and V_{R-} = AVSS 100b = V_{R+} = AVCC and V_{R-} = VREF-/VeREF- 101b = V_{R+} = VREF+ and V_{R-} = VREF-/VeREF- 110b = V_{R+} = VeREF+ and V_{R-} = VREF-/VeREF- 111b = V_{R+} = VeREF+ and V_{R-} = VREF-/VeREF-
3-0	ADC12INCHx	RW	0h	Input channel select 0000b = A0 0001b = A1 0010b = A2 0011b = A3 0100b = A4 0101b = A5 0110b = A6 0111b = A7 1000b = VeREF+ 1001b = VREF-/VeREF- 1010b = Temperature diode 1011b = (AVCC – AVSS) / 2 1100b = A12. On devices with the Battery Backup System, VBAT can be measured internally by the ADC. 1101b = A13 1110b = A14 1111b = A15

Mostly required part!

- The ADC12MCTL0 register contains settings for the reference voltage selection (ADC12SREFx) and the ADC input channel that will be routed to the sample-and-hold stage (ADC12INCHx). For the ADC12SREFx settings, “VREF” refers to the internal reference voltages that the MCU can produce while “VREF+/-” refers to external pins. Also, “AVCC” refers to the MCU power supply (+3.4 v) while “AVSS” refers to the MCU ground (0v). The ADC12INCHx setting allows 1 of 16 different input channels to be chosen. Of these 16 channels, 12 can come from MCU port pins. If any of these 12 port pins are to be used, **they must also be configured for ADC by using PxSEL. Making the corresponding bit 1 in PxSEL register makes that bit analog input.**
- Default value in ADC12SREFx is 0, which makes VREF+= VCC, VREF-= VSS

ADC Interrupt Registers (ADC12IE)

The ADC peripheral contains interrupt flag that can be used to monitor the status of the conversion. There is an interrupt flag that will trigger when a conversion is complete (ADC12IFGx). This interrupt is maskable and are locally enabled within the ADC12IEx register

15	14	13	12	11	10	9	8
ADC12IE15	ADC12IE14	ADC12IE13	ADC12IE12	ADC12IE11	ADC12IE10	ADC12IE9	ADC12IE8
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ADC12IE7	ADC12IE6	ADC12IE5	ADC12IE4	ADC12IE3	ADC12IE2	ADC12IE1	ADC12IE0
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Bit	Field	Type	Reset	Description
15	ADC12IE15	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG15 bit. 0b = Interrupt disabled 1b = Interrupt enabled
14	ADC12IE14	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG14 bit. 0b = Interrupt disabled 1b = Interrupt enabled
13	ADC12IE13	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG13 bit. 0b = Interrupt disabled 1b = Interrupt enabled
12	ADC12IE12	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG12 bit. 0b = Interrupt disabled 1b = Interrupt enabled
11	ADC12IE11	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG11 bit. 0b = Interrupt disabled 1b = Interrupt enabled
10	ADC12IE10	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG10 bit. 0b = Interrupt disabled 1b = Interrupt enabled
9	ADC12IE9	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG9 bit. 0b = Interrupt disabled 1b = Interrupt enabled
8	ADC12IE8	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG8 bit. 0b = Interrupt disabled 1b = Interrupt enabled
7	ADC12IE7	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG7 bit. 0b = Interrupt disabled 1b = Interrupt enabled
6	ADC12IE6	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG6 bit. 0b = Interrupt disabled 1b = Interrupt enabled
5	ADC12IE5	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG5 bit. 0b = Interrupt disabled 1b = Interrupt enabled
4	ADC12IE4	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG4 bit. 0b = Interrupt disabled 1b = Interrupt enabled
3	ADC12IE3	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG3 bit. 0b = Interrupt disabled 1b = Interrupt enabled
2	ADC12IE2	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG2 bit. 0b = Interrupt disabled 1b = Interrupt enabled
1	ADC12IE1	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG1 bit. 0b = Interrupt disabled 1b = Interrupt enabled
0	ADC12IE0	RW	0h	Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG0 bit. 0b = Interrupt disabled 1b = Interrupt enabled

ADC Interrupt Registers (ADC12IFG)

The ADC peripheral contains interrupt flag that can be used to monitor the status of the conversion. There is an interrupt flag that will trigger when a conversion is complete (ADC12IFGx). This interrupt is maskable and are locally enabled within the ADC12IEx register

15	14	13	12	11	10	9	8
ADC12IFG15	ADC12IFG14	ADC12IFG13	ADC12IFG12	ADC12IFG11	ADC12IFG10	ADC12IFG9	ADC12IFG8
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ADC12IFG7	ADC12IFG6	ADC12IFG5	ADC12IFG4	ADC12IFG3	ADC12IFG2	ADC12IFG1	ADC12IFG0
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Bit	Field	Type	Reset	Description
15	ADC12IFG15	RW	0h	ADC12MEM15 interrupt flag. This bit is set when ADC12MEM15 is loaded with a conversion result. This bit is reset if the ADC12MEM15 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
14	ADC12IFG14	RW	0h	ADC12MEM14 interrupt flag. This bit is set when ADC12MEM14 is loaded with a conversion result. This bit is reset if the ADC12MEM14 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
13	ADC12IFG13	RW	0h	ADC12MEM13 interrupt flag. This bit is set when ADC12MEM13 is loaded with a conversion result. This bit is reset if the ADC12MEM13 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
12	ADC12IFG12	RW	0h	ADC12MEM12 interrupt flag. This bit is set when ADC12MEM12 is loaded with a conversion result. This bit is reset if the ADC12MEM12 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
11	ADC12IFG11	RW	0h	ADC12MEM11 interrupt flag. This bit is set when ADC12MEM11 is loaded with a conversion result. This bit is reset if the ADC12MEM11 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
10	ADC12IFG10	RW	0h	ADC12MEM10 interrupt flag. This bit is set when ADC12MEM10 is loaded with a conversion result. This bit is reset if the ADC12MEM10 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
9	ADC12IFG9	RW	0h	ADC12MEM9 interrupt flag. This bit is set when ADC12MEM9 is loaded with a conversion result. This bit is reset if the ADC12MEM9 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
8	ADC12IFG8	RW	0h	ADC12MEM8 interrupt flag. This bit is set when ADC12MEM8 is loaded with a conversion result. This bit is reset if the ADC12MEM8 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
7	ADC12IFG7	RW	0h	ADC12MEM7 interrupt flag. This bit is set when ADC12MEM7 is loaded with a conversion result. This bit is reset if the ADC12MEM7 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
6	ADC12IFG6	RW	0h	ADC12MEM6 interrupt flag. This bit is set when ADC12MEM6 is loaded with a conversion result. This bit is reset if the ADC12MEM6 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
5	ADC12IFG5	RW	0h	ADC12MEM5 interrupt flag. This bit is set when ADC12MEM5 is loaded with a conversion result. This bit is reset if the ADC12MEM5 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
4	ADC12IFG4	RW	0h	ADC12MEM4 interrupt flag. This bit is set when ADC12MEM4 is loaded with a conversion result. This bit is reset if the ADC12MEM4 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
3	ADC12IFG3	RW	0h	ADC12MEM3 interrupt flag. This bit is set when ADC12MEM3 is loaded with a conversion result. This bit is reset if the ADC12MEM3 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
2	ADC12IFG2	RW	0h	ADC12MEM2 interrupt flag. This bit is set when ADC12MEM2 is loaded with a conversion result. This bit is reset if the ADC12MEM2 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
1	ADC12IFG1	RW	0h	ADC12MEM1 interrupt flag. This bit is set when ADC12MEM1 is loaded with a conversion result. This bit is reset if the ADC12MEM1 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending
0	ADC12IFG0	RW	0h	ADC12MEM0 interrupt flag. This bit is set when ADC12MEM0 is loaded with a conversion result. This bit is reset if the ADC12MEM0 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending

Analog to Digital Converters (Example)

Ex: Design a program to use the ADC on the MSP430F5529 to digitize an analog voltage on P6.2. This pin can be driven on the LaunchPad board using a function generator. We will drive a sine wave voltage into P6.2 that goes from 0V to +3.4V. We will continually digitize this signal and assert the LEDs depending on its value. When P6.2 is below +3.0V, LED2 (P4.7) will be asserted(ON) and LED1 (P1.0) will be off. When P6.2 is above +3.0V, LED1 will be asserted(ON) and LED2 will be off.

Solution: The first step to set up the ADC for this design is to use the P6SEL registers to configure P6.2 to use its analog function (A2). Within the ADC12CTL0 register, we will set the number of conversion cycles to 16 using ADC12SHT and turn the ADC on using ADC12ON. Within ADC12CTL0 we do not need to use ADCMSC because we are not going to use repeated mode. Within ADC12CTL1 we will select SMCLK as the ADC clock source using ADSSELx and select the sample timer as the source for the sample trigger using (ADC12SHP).We will use the default values for ADC12SHS (use ADC12SC to trigger sample), ADC12ISSH (no inversion), and ADC12DIVx (no clock division, which means division by 1). We will also accept the default value for ADC12CONSEQx, which is to configure the ADC for a single-channel, single-conversion that is triggered by user. This means each conversion will need to be manually started. Within ADC12CTL2 we will set the resolution to 12-bits using ADC12RES. We will use the default values for ADC12PDIVx (no clock division which means division by 1), ADC12DF (data formatted as unsigned binary), and ADC12SR (support for sample rates up to ~200 ksps). Within ADC12MCTL0 we will set the ADC input channel to A2. We will accept the default values for ADC12SREFx, which uses V_{R+} = VCC and V_{R-} = 0V.

Within the main program loop, we will start the conversion by simultaneously asserting ADC12ENC and ADC12SC in the ADC12CTL0 register. After the conversion starts, we will wait in a polling loop until the conversion is finished by monitoring ADC12IFG0. After the conversion completes, we will read the result from the ADC12MEM0. Reading ADC12MEM0 will clear ADC12IFG0. We will then have logic to set the LEDs according to the ADC result.

Analog to Digital Converters (Example)

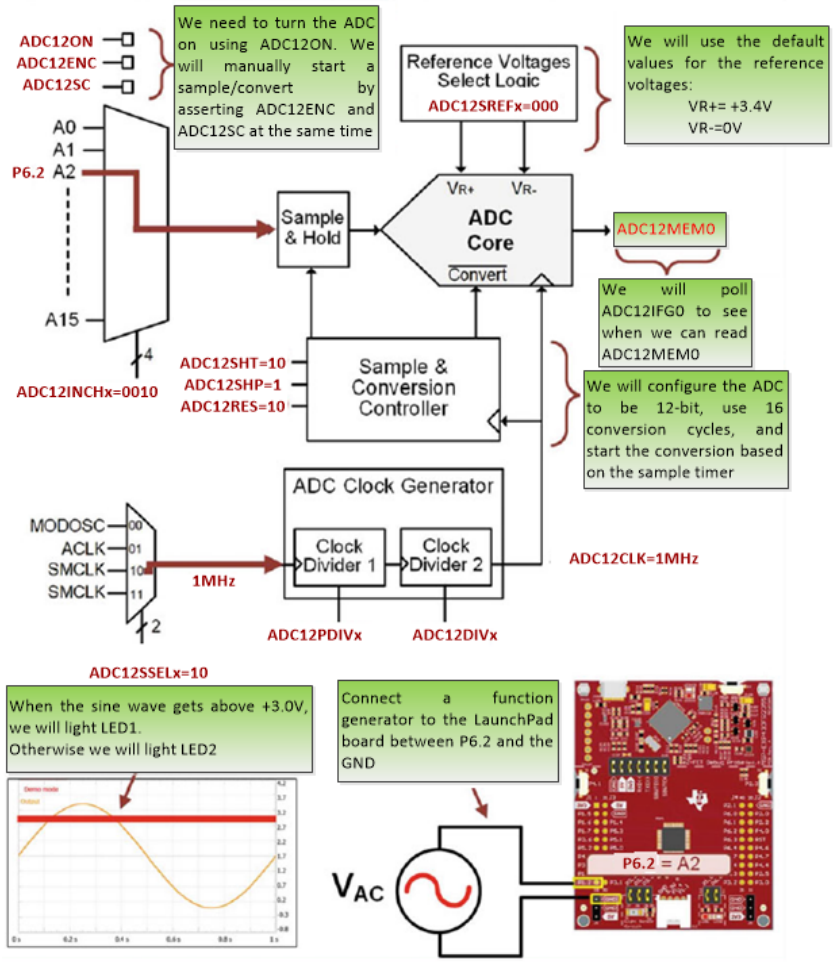


Fig. Graphical representation of the solution

Analog to Digital Converters (Example)

```
#include <msp430.h>
unsigned int ADC_value;// An unsigned integer variable to hold the ADC Value

int main(void)
{
WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
P1DIR |=BIT0; //P1.0 is output
P4DIR |=BIT7; //P4.7 is output
P6SEL |=BIT2; // Analog input as A2 on P6.2

ADC12CTL0 = ADC12SHT02 + ADC12ON; //16 ADC clock cycles and ADC is ON can also be ADC12CTL0 |= ADC12SHT02 | ADC12ON
ADC12CTL0 |= ADC12ENC; //Enable the conversion

ADC12CTL1 |= ADC12SHP; //Use sampling timer
ADC12CTL1 |= ADC12SSEL_2;  //ADC clock source is SMCLK, SMCLK=1MHz

ADC12CTL2 |= ADC12RES_2; //Resolution is 12-bit, n=12

ADC12MCTL0 |= ADC12INCH_2;// Use A2 as analog input, which corresponds to P6.2 on LaunchPad board
while(1)
{
    ADC12CTL0 |= ADC12SC; //Start sampling/ conversion
    while((ADC12IFG & ADC12IFG0)==0); //wait for conversion to complete
    ADC_value=ADC12MEM0; //Read the analog value and save in ADC_value variable

if (ADC_value>3614) //If A2>3v… 3.4/2^12=0.000830.  3/0.000830 = 3614
{
    P1OUT|=BIT0;    //LED1=ON
    P4OUT &= ~BIT7; //LED2=OFF
}
else // IfA2<3V
{
     P1OUT &= ~BIT0;  //LED1=OFF
     P4OUT |= BIT7;   //LED2=ON
}
}
return 0;
}
```