LABORATORY MANUAL

CE2107
Microprocessor System Design and Development

Lab Experiment #4

Analog-to-Digital Conversion and Timer Capture Operations

SESSION 2022/2023
SEMESTER 1

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

This lab session will be done in groups of 2. The group will work together to complete the lab exercises.

1. OBJECTIVES

1.1 Understand the process of Analog-to-Digital Conversion, operation of Sample-and-Hold and Successive Approximation Register (SAR) ADC

1.2 Interface to IR Sensor via ADC and perform Calibration

1.3 Using Timer hardware's Capture Mode to measure period of periodic signal

2. LABORATORY

This experiment is conducted at the **Hardware Lab 2** at **N4-01b-05 (Tel: 67905036)**.

3. HARDWARE EQUIPMENT

- A Windows-based computer (PC) with a Universal Serial Bus (USB) port.
- Texas Instruments Robotic System Learning Kit (RSLK-MAX)
- A USB A-to-MicroB cable.
- Oscilloscope.

4. ACKNOWLEDGEMENT

This lab reference and leverage from the work done by Dr Jonathan Valvano on the RSLK-MAX. The original source (sans solution) can be downloaded online under filename slac799a.zip. Students can also download the original lab notes (slay052a.pdf) for reference but note that adaptation had been made so there are differences in information, instructions and tasks. The original RSLK Max workshop is also available online at https://university.ti.com/en/faculty/ti-robotics-system-learning-kit/ti-rslk-max-edition-curriculum

5. REFERENCES (Can be found in the Doc sub-folder)

[1] Wk1-6 Lecture Notes

[2] RSLK-MAX Construction Guide (sekp164.pdf)

[3] MSP432 Launchpad UG (slau597f.pdf)

[4] MSP432 TRM (slau356h.pdf)

[5] MSP432 Datasheets (msp432p401r.pdf)

[6] ARM Optimizing Compiler UG (spnu151r.pdf)

[7] ARM Assembly Language Tools UG (spnu118u.pdf)

[8] Cortex M3/M4F Instruction Set (spmu159a.pdf)

6. ANALOG TO DIGITAL CONVERSION

An analog signal is one that is continuous in both amplitude and time. Most signals in the world exist as continuous functions of time in an analog fashion (e.g., voltage, current, position, angle, speed, force, pressure, temperature, and flow etc.). In other words, the signal has an amplitude that can vary over time, but the value does not instantaneously change.

To represent a signal in the digital domain we must approximate it in two ways:

- amplitude quantizing and
- time quantizing.

From an amplitude perspective, we will first place limits on the signal restricting it to exist between a minimum and maximum value (e.g., 0 to +3.3V), and second, we will divide this amplitude range into a finite set of discrete values. The range has units, such as volts or cm. The precision of the system is given in by the number of bits used to represent the data. For example, an 8-bit system can uniquely identify 256 different values. The resolution is the smallest change in value that can be represented. The resolution is given in the same units as the range.

range = resolution*2^n, where n is the precision in bits

The second approximation occurs in the time domain. Time quantizing is caused by the finite sampling interval. In practice we will use a periodic timer to trigger an analog to digital converter (ADC) to digitize information, converting from the analog to the digital domain. The Nyquist Theorem states that if the signal is sampled with a frequency of fs, then the digital samples only contain frequency components from 0 to ½ fs. Conversely, if the analog signal does contain frequency components larger than ½ fs, then there will be an aliasing error during the sampling process. Aliasing is when the digital signal appears to have a different frequency than the original analog signal.

In this lab, we will interface three IR Sensors (See Figure 1) to the microcontroller using ADC inputs. You will use periodic interrupts to sample the distance to the wall from three positions on the robot and perform calibration of the sensors. Refer to notes in Lab3 for exception handling.

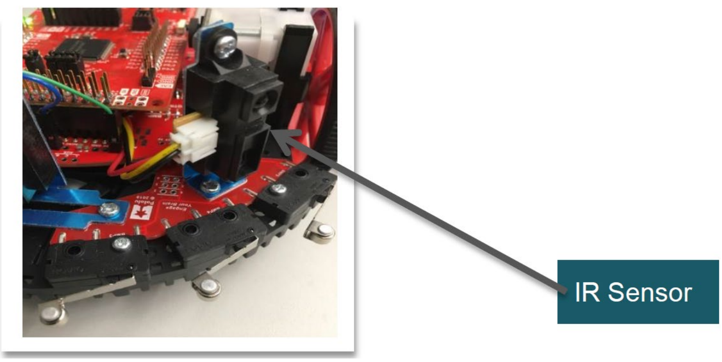


Figure 1: IR Sensors

6.1 IR Sensors

The IR Sensors used is GP2Y0A21YK0F from SHARP.

Figure 2 shows the measured relationship between output of the IR sensor and the distance to the wall (use a block wood and a ruler). Notice the non-monotonic behavior of the sensor. For example, if the system records a sensor value of 2V, it could mean 33 mm or 130 mm. In this lab, you can ignore the distance from 0 to 50 mm, or the distance corresponding to the maximum turning point of the graph, and any distance beyond 500 mm.

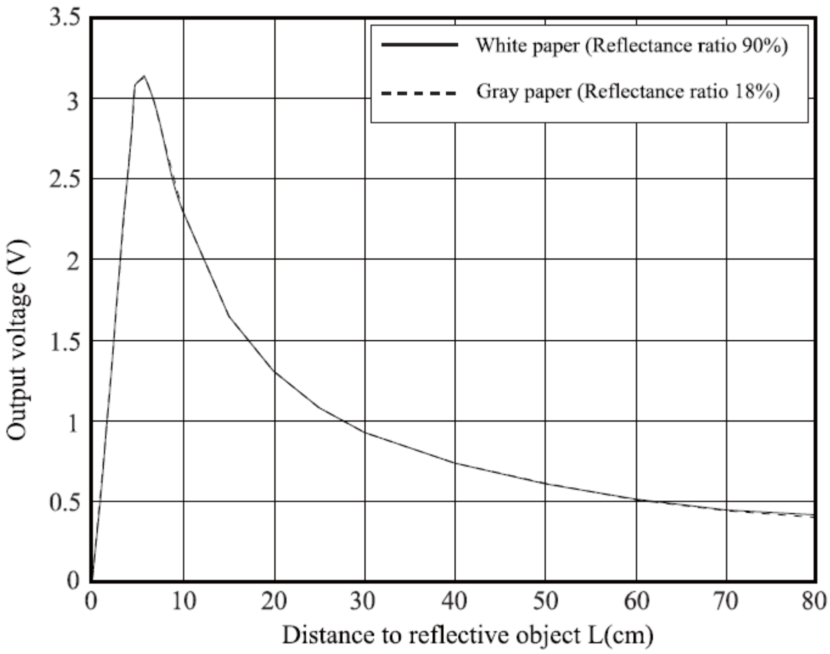


Figure 2: Voltage-Distance Characteristics of GP2Y0A21YK0F

There are a few finer points to note when doing placement of the IR Sensors, check the appendix for details.

The schematic diagram in Figure 5 shows the connection between the IR sensors and the MSP432. It is connected to Pin 9.0 (Right Sensor), 9.1(Left Sensor) and 4.1(Center Sensor), from the MSP432 datasheets, this corresponds to ADC17, ADC16 and ADC12 of MSP432. The sensor is very noisy so a decoupling capacitor on its power supply is required, we used a 10μF decoupling cap in the hardware.

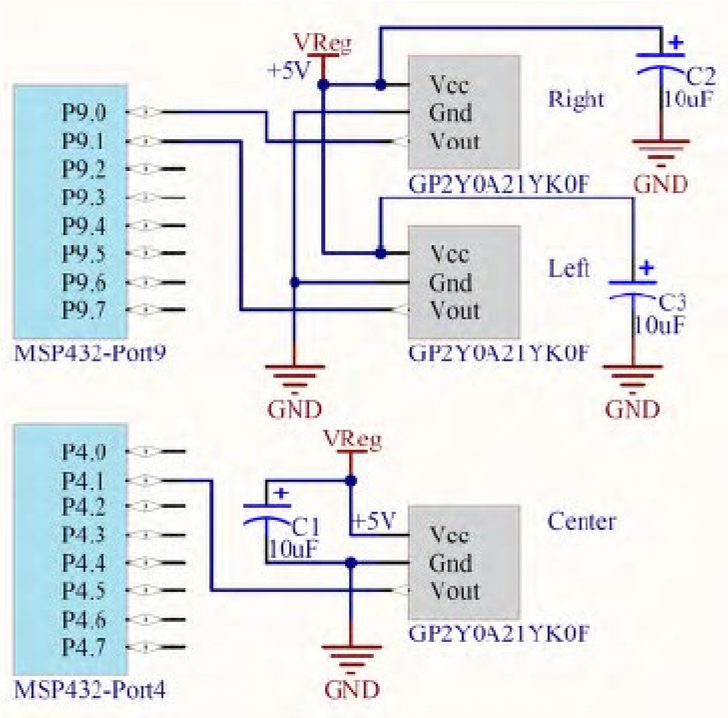


Figure 5: IR Sensors to MSP432 connection

6.2 ADC

The Precision ADC module is a native 14-bit SAR analog-to-digital conversions. The conversion-and-control buffer allows up to 32 independent analog-to-digital converter (ADC) samples to be converted and stored without any CPU intervention. Due to time complexity of configuring an ADC peripheral from scratch, the code is supplied in the project code. Details of the ADC peripheral can be found in Chapter 22 of slau356h. Some of the key points are listed below but you need not know the details of the ADC peripheral registers. You only need to know the setup procedure and what each function in the ADC14.c does. And note that other SAR ADC you will encounter in future is going to have a different set of control registers although the concept for a SAR ADC will be the same.

Port Pin mode Selection

The Precision ADC inputs are multiplexed with digital port pins. Selection of the pins are done in PSEL0 and PSEL1 registers. Corresponding value can be found in MSP432 datasheets, just search for the appropriate GPIO pins. The table for P9.0 is shown in Figure 6 below. Setting P9SEL0 register bit0 = 1, P9SEL1 register bit0 = 1 will configure P9.0 to function as Ch17 input of the ADC.

PIN NAME (P9.x)	x	FUNCTION	CONTROL BITS OR SIGNALS[1]		
			P9DIR.x	P9SEL1.x	P9SEL0.x
P9.0/A17[2]	0	P9.0 (I/O)	I: 0; O: 1	0	0
		N/A	0	0	1
		DVSS	1		
		N/A	0	1	0
		DVSS	1		
		A17[3]	X	1	1

Figure 6: Port pin mode selection for P9.0 (from Pg 156 of MSP432 datasheets)

Conversion Clock

The ADC14CLK is used both as the conversion clock and to generate the sampling period when the pulse sampling mode is selected. The Precision ADC clock source is selected using the ADC14SSELx bit. In this lab, we are using SMCLK to as the clock source for ADC14CLK. Figure 7 show the entire process of the conversion, the red box represent sample and hold period while the blue box represents the actual SAR ADC conversion.

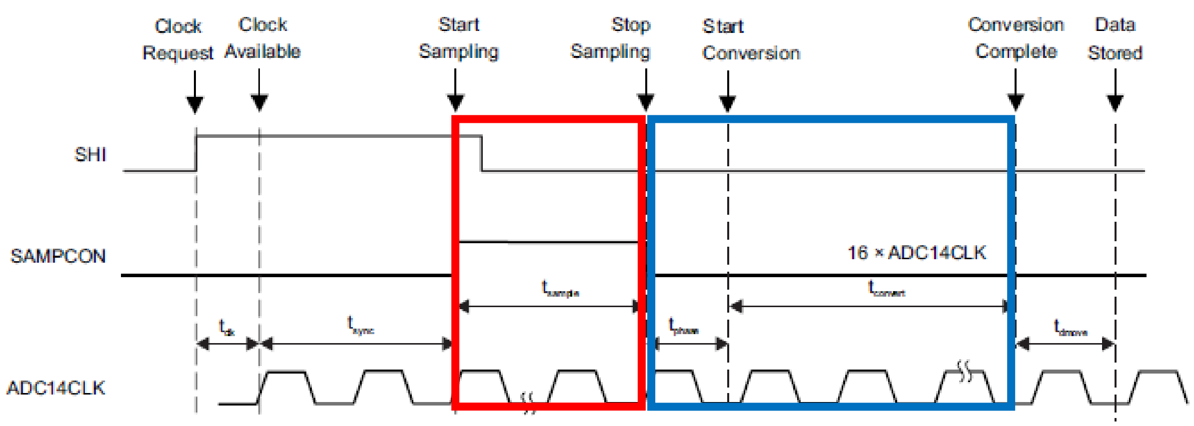


Figure 7: ADC Conversion process (Pulse Sample Mode)

Voltage Reference

This determines the range of inputs allowed. The Precision ADC module may use the following power rail as supply to ADC. In this lab, we are using the Analog Voltage Rail (AVcc), which is 3.3V. An input of 0V will yield a raw ADC output of 0x0, while an input of 3.3V will yield 0x3FFF (14 bits ADC).

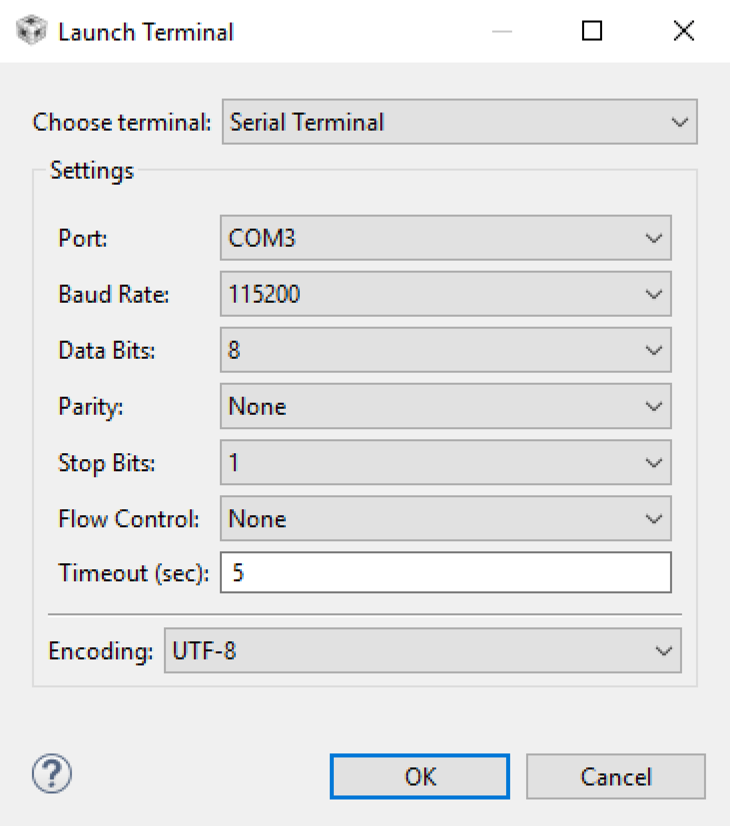
Sample and Conversion

As mentioned above, the whole ADC process consist of two parts, the ADC first sample the input signal voltage (Sample-and-Hold), the detected voltage level is then passed into the SAR ADC for digitization (Conversion). The process can be triggered by setting a software bit or by external signals. In this lab, we are using software trigger i.e. by setting the ADC14SC bit in ADC14CTL0 register. The complete description of the ADC operation and configuration can be found in the appendix. Check it out if you are interested.

6.3 Interfacing to IR Sensors

In this section, you will interface the IR sensors on the robot to the ADC14 peripheral on MSP432. The Lab4_ADC_IRSensors project uses systick timer interrupt to trigger the ADC periodically by setting the ADC14SC bit. The ADC raw outputs are processed using a Low Pass Filter to remove noise. The filtered ADC outputs are passed into the convert routine (that you need to code) to derive the obstacle distance. The distance information is output via UART to a Terminal Window.

The UART configuration used by the program is as follow.



#Task:

- Open Lab4_ADC_IRSensors project
- Check out the codes to see how ADC16, ADC12 and ADC17 are connected to the Left, Center and Right Sensor respectively.
- Calibrate the Sensors to derive a ADCvalue-to-distance table for each sensor.
- Potential Files to modify.
    - IRDistance.c

The above task involves a few sub-task illustrated below

- Open the project **Lab4_ADC_IRSensors**

- The top-level file is provided to you in **Lab4_ADCmain.c**. Open the file to check what are the top-level functions required. Some of these functions are empty functions which requires you to fill in.
    - **LeftConvert(), CenterConvert() and RightConvert()** function. This would require you to make measurement of the ADC Values against the obstacle's distance.

The graph of the IR Sensor output vs distance is non-linear as seen in Figure 2. There are two options to calibrate the IR Sensors.

- Making an exact curve fit using the hyperbolic equation $\mathbf{X} = A/(\mathbf{n} + B)$. You could also use other polynomial or log functions for curve fitting.
- Doing a piecewise linear approximation, e.g. partition the curve into sectors of 100 mm and fit a straight line ($y = mx + c$) within this sector, repeat for other sectors.

For both methods, you will need to collect sufficient data points to form the required equation(s). You can ignore distance below that corresponding to the Maximum turning point.

7. TIMER CAPTURE MODE (Code Observation and understanding)

7.1 Introduction

Timer capture mode is selected when CAP = 1. Capture mode is used to record timing of events. It can be used for speed computations or time measurements. The capture inputs CCIxA and CCIxB are connected to external pins or internal signals and are selected with the CCIS bits. The CM bits select the capture edge of the input signal as rising, falling, or both. A capture occurs on the selected edge of the input signal. If a capture occurs:

- The timer value is copied into the TAxCCRn register.
- The interrupt flag CCIFG is set.

Devices may have different signals connected to CCIxA and CCIxB. See the device-specific data sheet for the connections of these signals. On MSP432, there is no CCIxB input pins, only CCIxA pins are available for external timer input signals. For example, TA3.CCI0A, the input pin for Timer_A3 CCR0 in capture mode, is multiplexed with P10.4 on MSP432. Below is an extract from Pg 166 of the MSP432 datasheets. The table also shows the required configuration to initialise P10.4 to Timer capture pin.

PIN NAME (P10.x)	x	FUNCTION	CONTROL BITS OR SIGNALS[1]		
			P10DIR.x	P10SEL1.x	P10SEL0.x
P10.4/TA3.0/C0.7[2]	4	P10.4 (I/O)	I: 0; O: 1	0	0
		TA3.CCI0A	0	0	1
		TA3.0	1		
		N/A	0	1	0
		DVSS	1		
		C0.7[3][4]	X	1	1

7.2 Timer Capture Implementation

In this section, we will use the Timer to measure the period of a periodic signal.

Open the project Lab4_TimerCapture_Tach. This project configures TimerA3 channel 0 and 1 input pins (P10.4 and P10.5) in capture mode. The square wave output from the Tachometer on the two wheels are connected to P10.4 and P10.5. Each rising edge detected on the pins will cause the timer count to be latched to the channel registers, the difference of timer count value between two successive rising edges is the period of the square pulses.

#Task:

- Open Lab4_TimerCapture_Tach project.
- Understand how Timer_A3 is configured to take input from TA3CCI0A (P10.4) and TA3CCI1A (P10.5).
- Understand how the ISRs are linked to the two interrupt handlers servicing interrupts from TA3CCI0A and TA3CCI1A. Recall that each timer can be serviced by two interrupts.

8. ASSIGNMENT

- Complete the Lab4 handout and submit before your next lab.

Appendix

A. IR Sensors

There are a few finer points to note when doing placement of the IR Sensors, detail in Pg 6 of the GP2Y0A21YK0F datasheets. Briefly, when the obstacle has a huge difference in material or colour, the boundary line and the sensor's long edge should be in parallel (Figure 3). And in general, sensor's long ledge should be perpendicular to the sensor movement direction (Figure 4). But no worries, we won't get into these details for the lab.

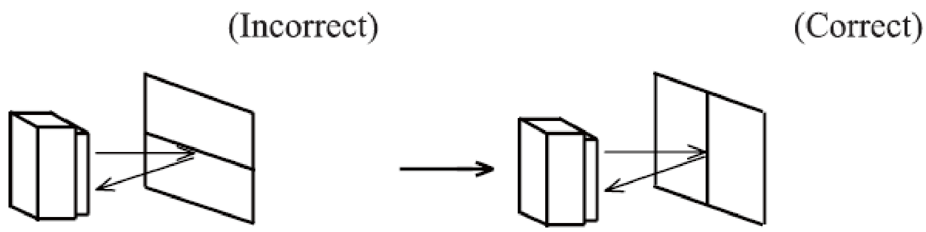


Figure 3: Huge difference in material characteristics

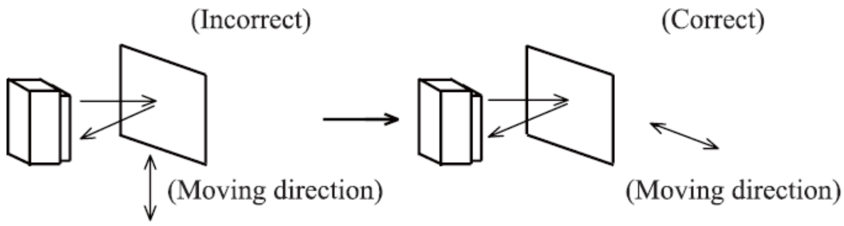


Figure 4: Sensor's long edge vs Movement direction

B. ADC

The Precision ADC module is a native 14-bit SAR analog-to-digital conversions. The conversion-and-control buffer allows up to 32 independent analog-to-digital converter (ADC) samples to be converted and stored without any CPU intervention. Due to time complexity of configuring an ADC peripheral from scratch, the code is supplied in the project code. Details of the ADC peripheral can be found in Chapter 22 of slau356h. Some of the key points are listed below but you need not know the details of the ADC peripheral registers, you only need to know the setup procedure and what each function in the ADC14.c does. And note that other SAR ADC you will encounter in future is going to have a different set of control registers although the concept for a SAR ADC will be the same.

Port Pin mode Selection

The Precision ADC inputs are multiplexed with digital port pins. Selection of the pins are done in PSEL0 and PSEL1 registers. Corresponding value can be found in MSP432 datasheets, just search for the appropriate GPIO pins. The table for P9.0 is shown in Figure 6 below. Setting P9SEL0 register bit0 = 1, P9SEL1 register bit0 = 1 will configure P9.0 to function as Ch17 input of the ADC.

PIN NAME (P9.x)	x	FUNCTION	CONTROL BITS OR SIGNALS[1]		
			P9DIR.x	P9SEL1.x	P9SEL0.x
P9.0/A17[2]	0	P9.0 (I/O)	I: 0; O: 1	0	0
		N/A	0	0	1
		DVSS	1		
		N/A	0	1	0
		DVSS	1		
		A17[3]	X	1	1

Figure 6: Port pin mode selection for P9.0 (from Pg 156 of MSP432 datasheets)

Conversion Clock

The ADC14CLK is used both as the conversion clock and to generate the sampling period when the pulse sampling mode is selected. The Precision ADC clock source is selected using the ADC14SSELx bit. Possible ADC14CLK clock sources are MODCLK, SYSCLK, ACLK, MCLK, SMCLK, and HSMCLK. The input clock can be divided by 1, 4, 32, or 64 using the ADC14PDIV bits and then subsequently divided by 1 to 8 using the ADC14DIV bits. Figure 7 show the entire process of the conversion, the red box represent sample and hold period while the blue box represents the actual SAR ADC conversion.

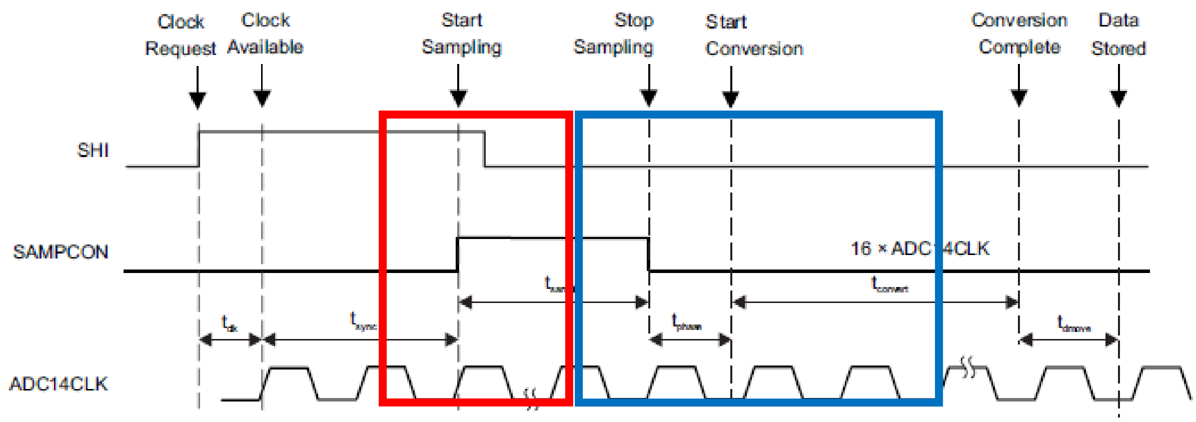


Figure 7: ADC Conversion process (Pulse Sample Mode)

Voltage Reference

This determines the range of inputs allowed. The Precision ADC module may use the following power rail as supply to ADC.

- On-chip shared reference module (VREF) that supplies three selectable voltage levels of 1.2 V, 1.45 V, and 2.5 V
- The chip's Analog Voltage (AVcc), 3.3V
- Through external pins VREF+/VeREF+ and VeREF-.

In this lab, we are using AVcc

Sample and Conversion

As mentioned above, the whole ADC process consist of two parts, the ADC first sample the input signal voltage (Sample-and-Hold), the detected voltage level is then passed into the SAR ADC for digitization (Conversion). The process can be triggered by setting a software bit or by external signals. This is configured by initializing the ADC14SHSx bit in the ADC14CTL0 register. In this lab, we are using software trigger i.e. by setting the ADC14SC bit in ADC14CTL0 register.

The conversion timing is dictated by the resolution used, the analog-to-digital conversion requires 9, 11, 14 and 16 ADC14CLK cycles for 8-bit, 10-bit, 12-bit, and 14-bit resolution respectively.

For sample timing, the ADC14SHT0x and ADC14SHT1x bits in ADC14CTL0 control the interval of the sampling timer. ADC14SHTx bits select the sampling time in 4x multiples of ADC14CLK. The programmable range of sampling timer is 4 to 192 ADC14CLK cycles. ADC14SHT0x selects the sampling time for ADC14MCTL8 to ADC14MCTL23, and ADC14SHT1x selects the sampling time for ADC14MCTL0 to ADC14MCTL7 and ADC14MCTL24 to ADC14MCTL31.

There are 32 ADC14MEMx conversion memory registers to store conversion results. Each ADC14MEMx is configured with an associated ADC14MCTLx control register.

Conversion Modes

The ADC can be configured to sample only one channel or sample a sequence of channels. This is selected by the CONSEQx bits.

ADC14CONSEQx	Mode	Operation	Section
00	Single-channel single-conversion	A single channel is converted once.	Section 22.2.8.1
01	Sequence-of-channels (autoscan)	A sequence of channels is converted once.	Section 22.2.8.2
10	Repeat-single-channel	A single channel is converted repeatedly.	Section 22.2.8.3
11	Repeat-sequence-of-channels (repeated autoscan)	A sequence of channels is converted repeatedly.	Section 22.2.8.4