



קובץ הכנה ניסוי מעבדה מס' 7

Tutorial 7.2 – Module ADC12 + Code Example

מעבדת מיקרומחשבים – המחלקה להנדסת חשמל ומחשבים

מס' קורס - 361.1.3353

כתיבה ועריכה: חנן ריבוא

מהדורה 1 – שנה"ל תשע"ו

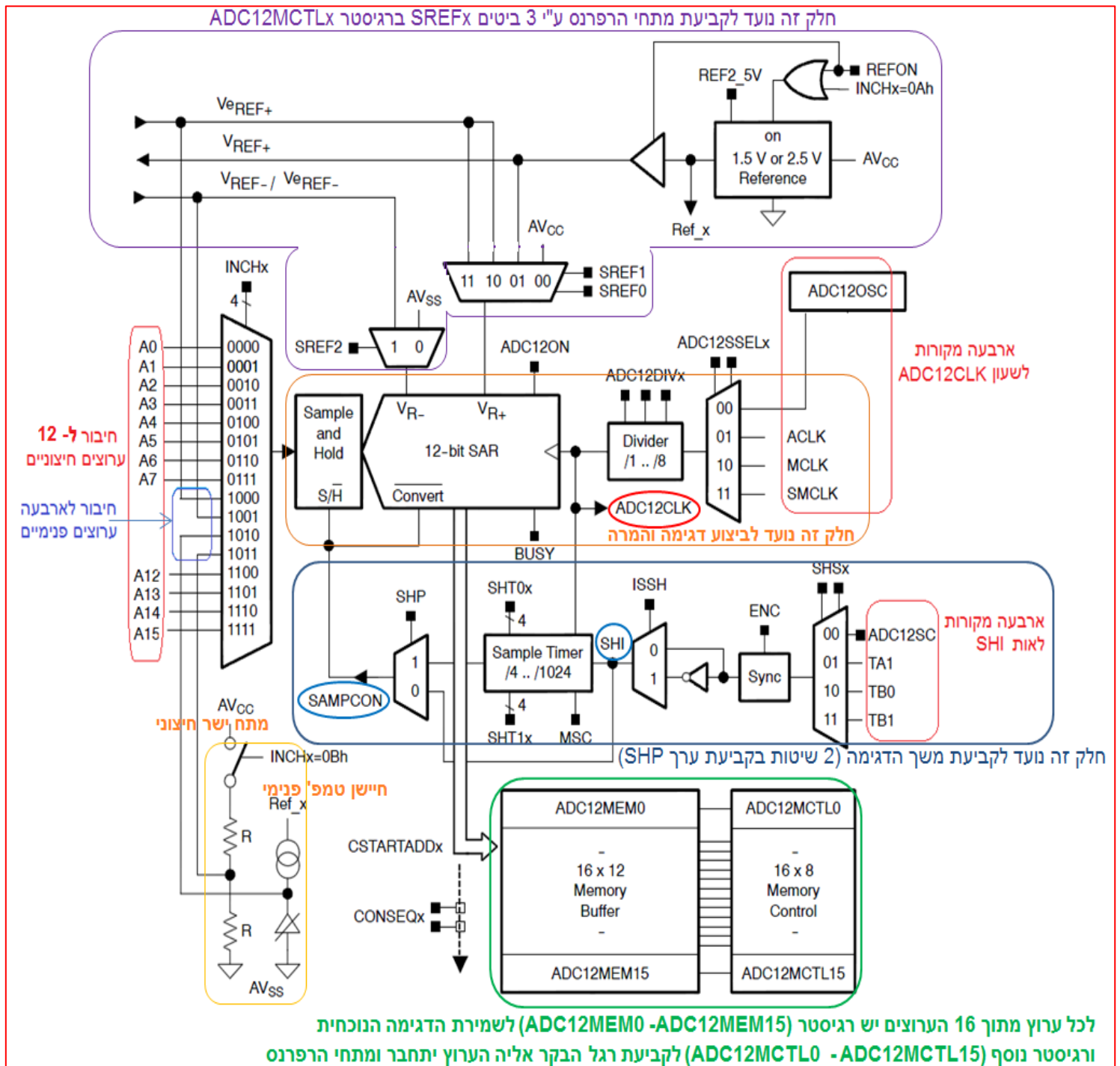
ADC12 (Analog-to-Digital Converter)

- קריאה מקדימה, בקובץ מעבדה MSP430x4xx user guide עמודים 799 - 773

A. מודול ADC12 - הקדמה:

- המודול ADC12 תומך ב-16 ערוצים אנלוגיים בלתי תלויים. בזמן נתון, המודול מבצע דגימת ערוץ אחד בלבד בשלבי עבודה של **דגימה המרה ואחסון**. לכל ערוץ ישנו רגיסטר לאחסון ערך של דגימה בודדת, סה"כ ישנם 16 רגיסטרי אחסון בשם **ADC12MEM0 - ADC12MEM15**. פעולת המודול נעשית במקביל לעבודת CPU (פעולת חומרה).
- שליטה על ליבת מודול ADC12 מתבצעת ע"י כתיבה לרגיסטרי הבקרה **ADC12CTL0, ADC12CTL1**
- ישנם 12 ערוצים חיצוניים (דרך רגלי הבקר) ו-4 ערוצים פנימיים בתוך הבקר (חיישן טמפ' פנימי, ומתחים פנימיים נוספים)
- גבולות המתח (מתחי הרפרנס) V_{R+} , V_{R-} הם **חיוביים בלבד**, כאשר האפשרויות הן:
 $V_{R-} = 0V$ ו- $V_{R-} = 1.5V, 2.5V, 3.3V$ ובחירתם נעשית בהתאם לנדרש באפליקציה.
- למודול ADC12 ישנו שעון ADC12CLK הקובע את משך זמן פעולת הדגימה ומשך זמן פעולת ההמרה. שעון **ADC12CLK** מוזן דרך מקורות שונים (מקור של 32768Hz בשם **ACLK**, שעון פנימי של 5MHz בשם **ADC12OSC**, שני מקורות נוספים **MCLK, SMCLK** שערך ברירת המחדל שלהם הוא $1048576Hz = 32 \cdot 32768Hz$). מרווח הדגימה Δt , שווה לסכום של משך הדגימה ומשך ההמרה.
- זמני דגימה והמרה (Sample and Conversion Timing):
 תזמון של התחלת כל דגימה מתבצע בעליית האות SHI (איור 1 - דיאגרמת בלוקים) הנשלט בעזרת אחד מארבעה המקורות הבאים:
 - בתוכנה, ע"י ביט ADC12SC שברגיסטר ADC12CTL1
 - אות מוצא מערוץ TA1 של Timer_A
 - אות מוצא מערוץ TB0 של Timer_B
 - אות מוצא מערוץ TB1 של Timer_B

דיאגרמת בלוקים בהדגשת הביטים השולטים על תפקוד מודול ADC12:



B. פירוט הרגיסטרים של מודול ADC12:

- (a) **רגיסטרי בקרה:** ישנם 2 רגיסטרי בקרה השולטים על תפקוד המודול ADC12 הנקראים **ADC12CTL0** ו- **ADC12CTL1**. ישנו רגיסטר בקרה נוסף לכל ערוץ (ישנם 16 ערוצים) בשם **ADC12MCTLx** ($x=1, \dots, 15$) השולט על תפקוד של רגיסטר המכיל את ערך הדגימה מאותו ערוץ בשם **ADC12MEMx** ($x=1, \dots, 15$). סה"כ ישנם 18 רגיסטרי בקרה.

ADC12CTL0 (1st control register):

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SHT1x				SHT0x				MSC	REF2_5V	REFON	ADC12ON	ADC12OVIE	ADC12TOVIE	ENC	ADC12SC
rw-(0)				rw-(0)				rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Modifiable only when ENC = 0

SHTx Bits	ADC12CLK cycles
0000	4
0001	8
0010	16
0011	32
0100	64
0101	96
0110	128
0111	192
1000	256
1001	384
1010	512
1011	768
1100	1024
1101	1024
1110	1024
1111	1024

SHT0x Bits 11-8

Sample-and-hold time. These bits define the number of ADC12CLK cycles in the sampling period for registers **ADC12MEM0** to **ADC12MEM7**.

SHT1x Bits 15-12

Sample-and-hold time. These bits define the number of ADC12CLK cycles in the sampling period for registers **ADC12MEM8** to **ADC12MEM15**.

MSC	Bit 7	Multiple sample and conversion. Valid only for sequence or repeated modes.
0		The sampling timer requires a rising edge of the SHI signal to trigger each sample-and-conversion.
1		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.

REF2_5V	Bit 6	Reference generator voltage. REFON must also be set.
0		1.5 V
1		2.5 V

REFON	Bit 5	Reference generator on
0		Reference off
1		Reference on

ADC12ON	Bit 4	ADC12 on
0		ADC12 off
1		ADC12 on

ADC12OVIE	Bit 3	ADC12MEMx overflow-interrupt enable. The GIE bit must also be set to enable the interrupt.
0		Overflow interrupt disabled
1		Overflow interrupt enabled

ADC12TOVIE	Bit 2	ADC12 conversion-time-overflow interrupt enable. The GIE bit must also be set to enable the interrupt.
0		Conversion time overflow interrupt disabled
1		Conversion time overflow interrupt enabled

ENC	Bit 1	Enable conversion
0		ADC12 disabled
1		ADC12 enabled

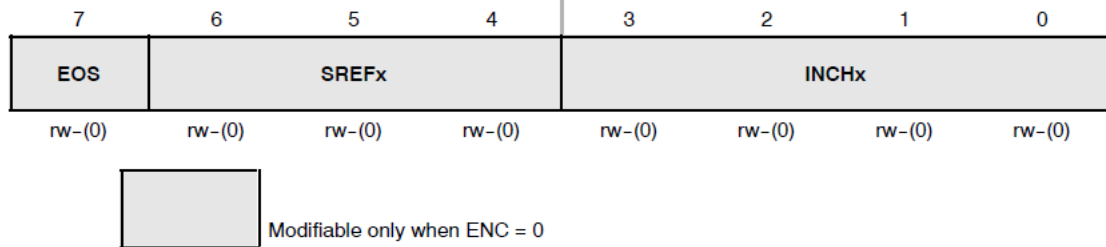
ADC12SC	Bit 0	Start conversion. Software-controlled sample-and-conversion start. ADC12SC and ENC may be set together with one instruction. ADC12SC is reset automatically.
0		No sample-and-conversion-start
1		Start sample-and-conversion

ADC12CTL1 (2nd Control Register):

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
CSTARTADDx				SHSx		SHP	ISSH	ADC12DIVx			ADC12SSELx		CONSEQx		ADC12 BUSY	
rw-(0)				rw-(0)		rw-(0)	rw-(0)	rw-(0)			rw-(0)		rw-(0)		r-(0)	
<div></div> Modifiable only when ENC = 0																

CSTART ADDx	Bits 15-12	Conversion start address. These bits select which ADC12 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.
SHSx	Bits 11-10	Sample-and-hold source select 00 ADC12SC bit 01 Timer_A.OUT1 10 Timer_B.OUT0 11 Timer_B.OUT1
SHP	Bit 9	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. 0 SAMPCON signal is sourced from the sample-input signal. 1 SAMPCON signal is sourced from the sampling timer.
ISSH	Bit 8	Invert signal sample-and-hold 0 The sample-input signal is not inverted. 1 The sample-input signal is inverted.
ADC12DIVx	Bits 7-5	ADC12 clock divider 000 /1 001 /2 010 /3 011 /4 100 /5 101 /6 110 /7 111 /8
ADC12 SSELx	Bits 4-3	ADC12 clock source select 00 ADC12OSC 01 ACLK 10 MCLK 11 SMCLK
CONSEQx	Bits 2-1	Conversion sequence mode select 00 Single-channel, single-conversion 01 Sequence-of-channels 10 Repeat-single-channel 11 Repeat-sequence-of-channels
ADC12 BUSY	Bit 0	ADC12 busy. This bit indicates an active sample or conversion operation. 0 No operation is active. 1 A sequence, sample, or conversion is active.

ADC12MCTLx (Conversion Memory Control Registers) - x = 1,...,15



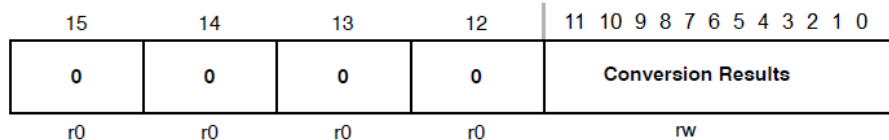
EOS	Bit 7	End of sequence. Indicates the last conversion in a sequence.	
		0	Not end of sequence
		1	End of sequence

SREFx	Bits	Select reference	
	6-4	000	$V_{R+} = AV_{CC}$ and $V_{R-} = AV_{SS}$
		001	$V_{R+} = V_{REF+}$ and $V_{R-} = AV_{SS}$
		010	$V_{R+} = V_{REF+}$ and $V_{R-} = AV_{SS}$
		011	$V_{R+} = V_{REF+}$ and $V_{R-} = AV_{SS}$
		100	$V_{R+} = AV_{CC}$ and $V_{R-} = V_{REF-} / V_{REF-}$
		101	$V_{R+} = V_{REF+}$ and $V_{R-} = V_{REF-} / V_{REF-}$
		110	$V_{R+} = V_{REF+}$ and $V_{R-} = V_{REF-} / V_{REF-}$
		111	$V_{R+} = V_{REF+}$ and $V_{R-} = V_{REF-} / V_{REF-}$

INCHx	Bits	Input channel select	
	3-0	0000	A0
		0001	A1
		0010	A2
		0011	A3
		0100	A4
		0101	A5
		0110	A6
		0111	A7
		1000	V_{REF+}
		1001	V_{REF-} / V_{REF-}
		1010	Temperature sensor
		1011	$(AV_{CC} - AV_{SS}) / 2$
		1100	A12
		1101	A13
		1110	A14
		1111	A15

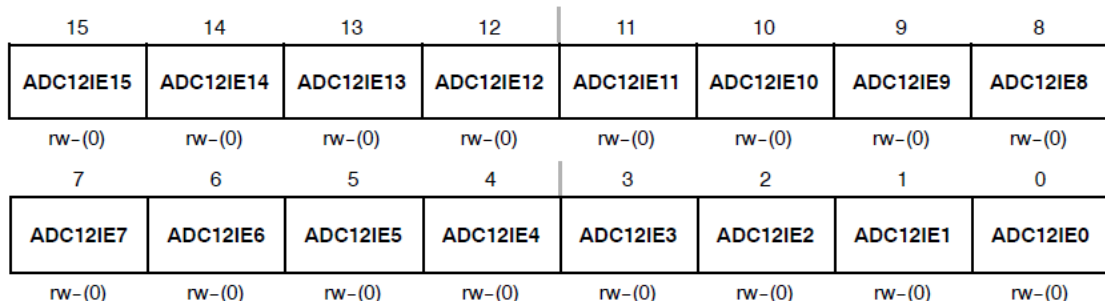
(b) רגיסטרי מידע, אפשר פסיקות ודגלי פסיקות:

ADC12MEMx (Conversion Memory Registers) - x = 1,...,15



Conversion Results	Bits	The 12-bit conversion results are right-justified. Bit 11 is the MSB. Bits 15-12 are always 0. Writing to the conversion memory registers will corrupt the results.	
	15-0		

ADC12IE (Interrupt Enable Register):



ADC12IEx	Bits	Interrupt enable. These bits enable or disable the interrupt request for the ADC12IFGx bits.	
	15-0	0	Interrupt disabled
		1	Interrupt enabled

ADC12IFG (Interrupt Flag 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)

ADC12IFGx	Bits	ADC12MEMx Interrupt flag. These bits are set when corresponding ADC12MEMx is loaded with a conversion result. The ADC12IFGx bits are reset if the corresponding ADC12MEMx is accessed, or may be reset with software.
	15-0	
	0	No interrupt pending
	1	Interrupt pending

ADC12IV (Interrupt Vector Register):

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
0	0	ADC12IVx					0
r0	r0	r-(0)	r-(0)	r-(0)	r-(0)	r-(0)	r0

ADC12IVx Bits 15-0 ADC12 interrupt vector value

ADC12IV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority
000h	No interrupt pending	-	
002h	ADC12MEMx overflow	-	Highest
004h	Conversion time overflow	-	
006h	ADC12MEM0 interrupt flag	ADC12IFG0	
008h	ADC12MEM1 interrupt flag	ADC12IFG1	
00Ah	ADC12MEM2 interrupt flag	ADC12IFG2	
00Ch	ADC12MEM3 interrupt flag	ADC12IFG3	
00Eh	ADC12MEM4 interrupt flag	ADC12IFG4	
010h	ADC12MEM5 interrupt flag	ADC12IFG5	
012h	ADC12MEM6 interrupt flag	ADC12IFG6	
014h	ADC12MEM7 interrupt flag	ADC12IFG7	
016h	ADC12MEM8 interrupt flag	ADC12IFG8	
018h	ADC12MEM9 interrupt flag	ADC12IFG9	
01Ah	ADC12MEM10 interrupt flag	ADC12IFG10	
01Ch	ADC12MEM11 interrupt flag	ADC12IFG11	
01Eh	ADC12MEM12 interrupt flag	ADC12IFG12	
020h	ADC12MEM13 interrupt flag	ADC12IFG13	
022h	ADC12MEM14 interrupt flag	ADC12IFG14	
024h	ADC12MEM15 interrupt flag	ADC12IFG15	Lowest



ברצוני לחבר כניסה אנלוגית לרגל A3, מדובר על רגל 2 של הבקר ולה יש 3 אפשרויות פעולה:

- כיצד נבחר בתוכנה אחת מ-3 אפשרויות אלו?

בקובץ MSP430xG461x datasheet נלחץ ctrl+ F ונחפש את **A3** או **P6.3** ונגיע לטבלת קינפוג מתאימה

82 בעמוד

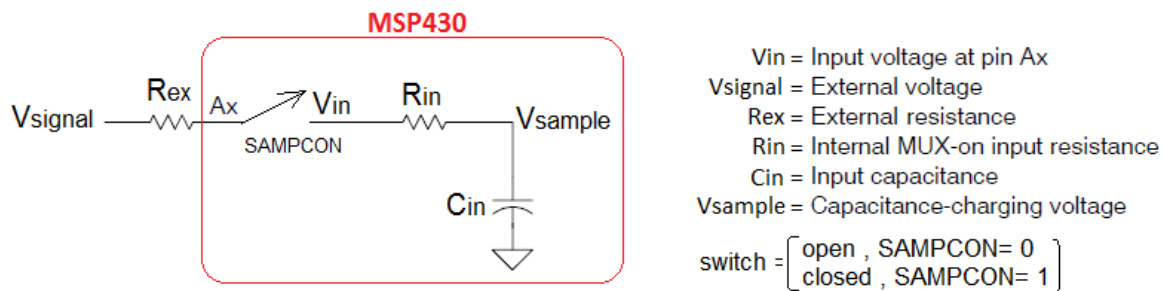
©Hanan Ribo

מהטבלה רואים שכדי לבחור רגל **P6.3** לשמש כניסה אנלוגית **A3** נדרש לבצע `bis.b #0x08.&P6SEL` ו-`mov.b #INCH 3.&ADC12MCTL3`

D. דגימה והמרה:

(a) דגימה:

- לצורך דגימה של אות חיצוני המשתנה בזמן ישנן מספר גישות אפשריות לטיפול חומרתי בכך.
- בתוך אינטרוול זמן של דגימה, לצורך החלטה על ערך הדגימה לפני המרתה. ניתן לדגום את האות מספר פעמים, בקצב מהיר יותר מקצב השתנות האות ואז לקחת את הערך של ה- Majority vote (ערך עם מספר מופעים הגדול ביותר). אפשרות חלופית, לקחת את הערך הממוצע של הדגימות.
 - מקרה שלנו:** בתוך אינטרוול זמן של דגימה, לצורך החלטה על ערך הדגימה לפני המרתה. ניתן לדגום פעם אחת ולהחזיק את ערך הדגימה בצורה יציבה. **מעגל זה נקרא sample and hold.** מעגל זה ממודל ע"י מעגל RC המחובר למתג. כאשר המתג סגור ערך הקבל נטען (sample) לערך הדגימה. משך זמן הטעינה, נקרא זמן הדגימה t_{sample} לצורך טעינת הקבל לערך ברזולוציה של 12 ביט. כאשר המתג פתוח, ערך הדגימה מוחזק (hold) בצורה יציבה לצורך המרה. בדיאגרמת מודול ADC הנ"ל, ניתן להבחין שדגימת האות החיצוני המחובר לרגל הבקר במצב של כניסה אנלוגית A_x מתבצעת רק כאשר אות $SAMPCON=1$, כמתואר בדיאגרמה הבאה.



ביטוי כללי לזמן הדגימה t_{sample} המינימאלי לרמת דגימה ברזולוציה של 12 ביט:

$$t_{sample} > (R_{ex} + R_{in}) \cdot \ln(2^{13}) \cdot C_{in} + 800ns$$

ביטוי ל- t_{sample} המינימאלי עבור מודול ADC12, כאשר $R_{ex} = 10k\Omega$, $R_{in} = 2k\Omega$, $C_{in} = 40pF$

$$t_{sample} > (10k + 2k) \cdot 9.011 \cdot 40p + 800n > 5.13\mu sec$$

משך זמן הדגימה t_{sample} ניתן לתכנות (ראה סעיף c עמוד 10), אולם עלינו לדאוג לקיום התנאי:

$$t_{sample} > 5.13\mu se$$

(b) המרה:

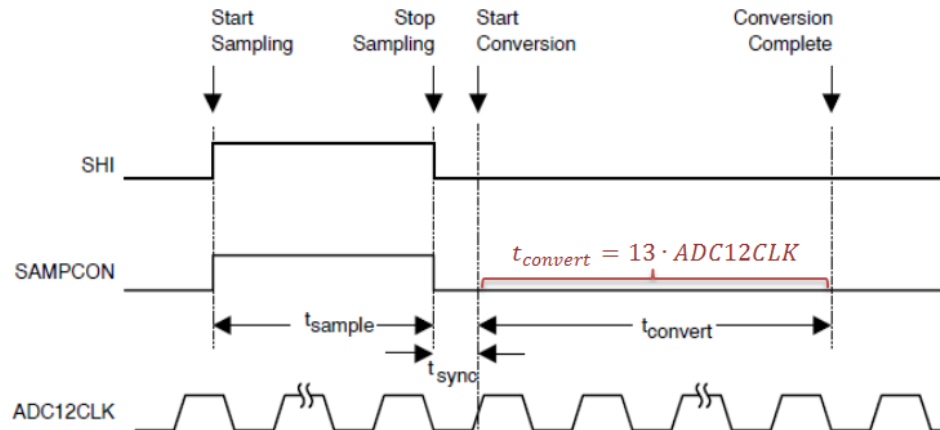
בתוך אינטרוול זמן של דגימה נדרש להמיר אותה לערך הנדרש באורך 12 ביט. ההמרה מתבצעת בחומרה בשיטת Successive approximation ולוקחת זמן קבוע t_{conv} של $m+1$ (במקרה שלנו $m=12$) מחזורי שעון של המודול ADC12.

(c) קביעת משך זמן הדגימה:

אות SHI הופך לאות SAMPCON באחת משתי אפשרויות (באופן ישיר או כטריגר – הבחירה בין השניים ע"י ביט SHP) ואות SAMPCON שולט באופן ישיר על משך מחזור הדגימה ועל תזמון תחילת ההמרה (משך זמן ההמרה הוא קבוע, 13 מחזורים של ADC12CLK). כאשר SAMPCON ב-1' לוגי מתבצעת דגימה וכאשר SAMPCON יורד ל-0' מתחילה ההמרה של אותה הדגימה. למעשה ישנן 2 שיטות שונות לדגימה המוגדרות ע"י ביט SHP, קביעה זו הינה עבור כל 16 הערוצים.

(1) כאשר ביט SHP=0:

אות SHI שולט באופן ישיר על אות SAMPCON ומגדיר את משך זמן הדגימה t_{sample} .

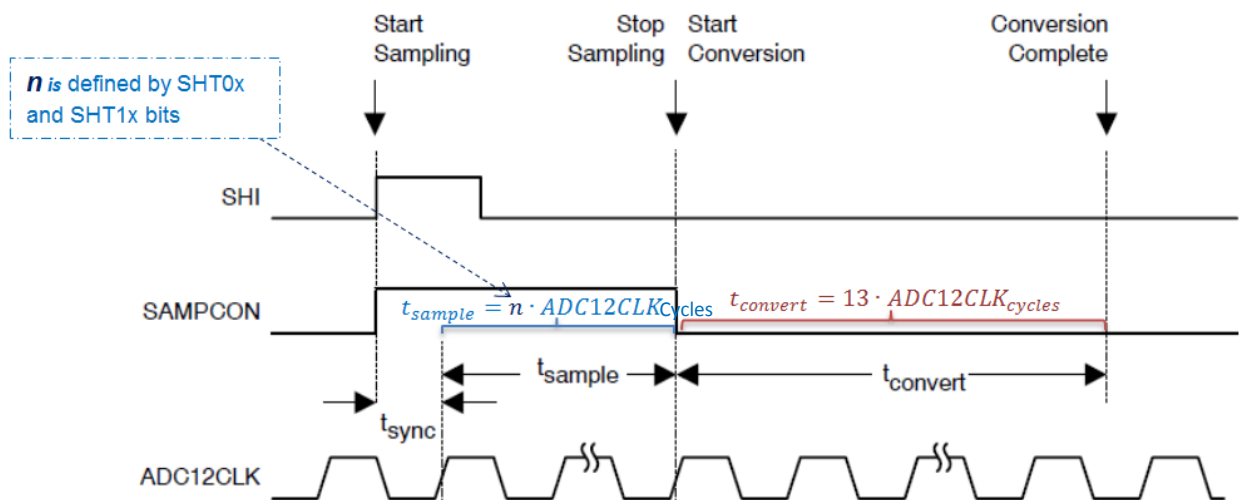


נדרש זמן נוסף t_{sync} לסנכרון בין SAMPCON ו-ADC12CLK.

(2) כאשר ביט SHP=1:

אות SHI משמש כטריגר, בעליית אות SHI נכנס לפעולה רכיב בשם Sampling Timer הקובע את משך הדגימה (משך אות SAMPCON) השווה ל- $n \cdot \text{ADC12CLK}_{\text{cycles}}$ כאשר n הוא מספר בטווח של $4 \div 1024$ (נקבע ע"י ביטים SHT0x ו-SHT1x). כאשר ביט MSC = 0 אות SHI מהווה טריגר לדגימה בודדת ולכן מתאים עבור אופן עבודה של דגימה בודדת, בלבד. כאשר ביט MSC = 1 אות SHI מהווה טריגר לדגימה ראשונה (מתאים לשאר אופני העבודה חוץ מאופן של דגימה בודדת) ושאר

הדגימות נעשות באופן אוטומטי מיד לאחר סיום המרת הדגימה הקודמת.



נדרש זמן נוסף t_{sync} לסנכרון בין SAMPCON ו-ADC12CLK.

E. ארבעת אופני העבודה של מודול ADC12:**(1) דגימה בודדת של הערוץ (Single-channel) $CONSEQx = 00$:**

(a) כאשר ביט **ADC12SC** מהווה טריגר להמרת דגימה ($SHSx = 0$), אם ברצוננו לבצע דגימה נוספת

נעלה שוב את ביט **ADC12SC** ל-1 (**הוא יורד ל-0 באופן אוטומטי לאחר הדגימה**).

(b) כאשר מקורות אחרים מהווים טריגר לדגימה ($SHSx > 0$) נדרש לכבות ולהדליק את ביט **ENC** בין המרה להמרה.

(c) לעצירת פעולת המרת הדגימה (לאחר שהמרת הדגימה האחרונה הושלמה, אפשר לבדוק זאת ע"י תשאול ביט **BUSY**, אחרת איבדנו אותה) נוריד את **ENC** ל-0.

(2) דגימה מחזורית של הערוץ (repeat-single-channel) $CONSEQx = 10$:

הכרחי לקרוא את ערך הדגימה מיד לאחר שהושלמה מרגיסטר **ADC12MEMx** המתאים ולפני שתושלם המרת הדגימה הבאה. לעצירת פעולת המרת הדגימות נוריד את **ENC** ל-0 ולאחר שההמרה האחרונה תסתיים תתבצע עצירה.

(3) דגימה בודדת של מספר ערוצים באופן טורי (sequence) $CONSEQx = 01$:

לעצירת פעולת המרת הדגימות נוריד את **ENC** ל-0 ולאחר שהמרת הערוץ האחרון בסדרה יסתיים תתבצע עצירה (לעצירה מידית נאפס את **CONSEQx** ולאחר מכן את **ENC**).

(4) דגימה מחזורית של מספר ערוצים באופן טורי (repeat-sequence) $CONSEQx = 11$:

לעצירת פעולת המרת הדגימות נוריד את **ENC** ל-0 ולאחר שהמרת הערוץ האחרון בסדרה יסתיים תתבצע עצירה. (לעצירה מידית נאפס את **CONSEQx** ולאחר מכן את **ENC**).

(a) בחירת ערוץ (מתוך ה-16) לביצוע דגימה נוכחית נקבע ע"י מצביע (לפי ערך ביטים של

CSTARTADDx ברגיסטר **ADC12CTL1** שערכו נע בין $0-0x0F$ **CSTARTADDx**. כאשר מדובר בדגימת ערוץ אחד (דגימה בודדת / מחזורית) נכתוב בביטים **CSTARTADDx** את מספר הרגיסטר **ADC12MEMx** המקושר לערוץ.

(b) כאשר מדובר בדגימה של סדרת ערוצים (דגימה בודדת / מחזורית) נכתוב את מספר הרגיסטר **ADC12MEMx** המקושר לערוץ הראשון בסדרה, בביטים **CSTARTADDx** ולרגיסטר **ADC12MEMx** המקושר לערוץ **האחרון בסדרה** (מספר **ADC12MEMx** הוא הגבוה מבין רגיסטרי אחסון הדגימות בסדרה) נעלה את ביט **EOS** ל-'1' הנמצא ברגיסטר **ADC12MCTLx** השייך לו. במקרה זה המצביע **CSTARTADDx** יתקדם באופן אוטומטי ומוסתר (לא ניתן לראותו בתוכנה) לפי סדר הרגיסטרים **ADC12MEMx** וכשיגיע לערוץ שאצלו ערך **EOS=1** יתחיל את הסדרה מההתחלה (מתאים לאופן **(repeat-sequence)**).

הערה: בכל 4 אופני העבודה לסיום עבודה עם מודול **ADC12** או כדי לחסוך באנרגיה כשהוא לא בשימוש.

לאחר עצירת פעולת ההמרה נוכל לאפס את ביט **ADC12ON**.

F. פסיקות מודול ADC12:

(a) כאשר המרת הדגימה הושלמה ונכתבת לרגיסטר מתאים מבין **ADC12MEM0 - ADC12MEM15**, הדגל המתאים עבורו **ADC12IFG0 - ADC12IFG15** ברגיסטר **ADC12IFG** עולה ל-'1'. דגל זה ירד ל-'0' רק לאחר שניגש בתוכנה (פקודה כלשהי המשתמשת בו) לרגיסטר **ADC12MEM** המתאים שלו. ניתן לאפס את דגל הפסיקה באופן ידני. בקשת פסיקה תאפשר (מעבר של PC לביצוע ה-ISR שלה) בתנאי שבצענו אפשר מקומי לפסיקה ($ADC12IE = 1$) ואפשר גלובלי לפסיקה ($GIE = 1$). סה"כ 16 מקורות פסיקה הקשורים ל-16 ערוצי דגימה של **ADC12**.

(b) ישנם 2 מקורות פסיקה נוספים:

- הראשון, כאשר ישנה בקשה ל-**ADC12** לביצוע המרה חדשה לפני שהושלמה קודמתה. בתנאי שבצענו אפשר מקומי לפסיקה ($ADC12TOVIE = 1$) ואפשר גלובלי לפסיקה ($GIE = 1$).
- השני, כאשר תוצאת המרה חדשה נכתבת לאחד מרגיסטרי **ADC12MEM** לפני שקראנו את קודמתה. בתנאי שבצענו אפשר מקומי לפסיקה ($ADC12OVIE = 1$) ואפשר גלובלי לפסיקה ($GIE = 1$).

(c) וקטור הפסיקה של מודול **ADC12** נקרא בשם **ADC12_VECTOR** והוא קשור ל-18 מקורות פסיקה של **ADC12** (פירוט בסעיף קודם). במצב זה לצורך ייעול כתיבת הקוד מבחינת זמן ריצה ונוחות (כאשר אנו עובדים עם 2 מקורות פסיקה ומעלה) ישנו רגיסטר בשם **ADC12IV** המכיל ערכים בתחום $0x000 - 0x024$ המתאימים ל-18 מקורות הפסיקה לפי הטבלה הבאה:

ADC12IV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority
000h	No interrupt pending	-	
002h	ADC12MEMx overflow	-	Highest
004h	Conversion time overflow	-	
006h	ADC12MEM0 interrupt flag	ADC12IFG0	
008h	ADC12MEM1 interrupt flag	ADC12IFG1	
00Ah	ADC12MEM2 interrupt flag	ADC12IFG2	
00Ch	ADC12MEM3 interrupt flag	ADC12IFG3	
00Eh	ADC12MEM4 interrupt flag	ADC12IFG4	
010h	ADC12MEM5 interrupt flag	ADC12IFG5	
012h	ADC12MEM6 interrupt flag	ADC12IFG6	
014h	ADC12MEM7 interrupt flag	ADC12IFG7	
016h	ADC12MEM8 interrupt flag	ADC12IFG8	
018h	ADC12MEM9 interrupt flag	ADC12IFG9	
01Ah	ADC12MEM10 interrupt flag	ADC12IFG10	
01Ch	ADC12MEM11 interrupt flag	ADC12IFG11	
01Eh	ADC12MEM12 interrupt flag	ADC12IFG12	
020h	ADC12MEM13 interrupt flag	ADC12IFG13	
022h	ADC12MEM14 interrupt flag	ADC12IFG14	
024h	ADC12MEM15 interrupt flag	ADC12IFG15	Lowest

מקור פסיקה בעל העדיפות הגבוהה ביותר מייצר מספר ברגיסטר ADC12IV בתחום 0x000 – 0x024 (גם כאשר מקור פסיקה ממוסך – אי-אפשר פסיקה באופן מקומי / גלובלי) שאותו נוכל להוסיף לרגיסטר PC בפקודה הראשונה של ISR (add &ADC12IV,PC) כך נוכל להגיע ישירות לשורות קוד המטפלות במקור הפסיקה הרלוונטי בלבד (ראה דוגמה הבאה של שימוש נכון ברגיסטר ADC12IV). כל גישה לרגיסטר (קריאה בלבד) ADC12IV באופן אוטומטי מאפסת אותו.

; Interrupt handler for ADC12.		Cycles
INT_ADC12	; Enter Interrupt Service Routine	6
ADD &ADC12IV,PC	; Add offset to PC	3
RETI	; Vector 0: No interrupt	5
JMP ADOV	; Vector 2: <u>ADC overflow</u>	2
JMP ADTOV	; Vector 4: <u>ADC timing overflow</u>	2
JMP ADM0	; Vector 6: ADC12IFG0	2
....	; Vectors 8-32	2
JMP ADM14	; Vector 34: ADC12IFG14	2
ADM15	; Handler for ADC12IFG15 starts here. <u>No JMP required.</u>	
MOV &ADC12MEM15,xxx	; Move result, flag is reset	
....	; Other instruction needed?	
JMP INT_ADC12	; <u>Check if other interrupts pending</u>	
; ADC12IFG14-ADC12IFG1 handlers go here		
ADM0		
MOV &ADC12MEM0,xxx	; Move result, flag is reset	
....	; Other instruction needed?	
RETI	; Return	5
ADTOV		
....	; Handle Conv. time overflow	
RETI	; Return	5
ADOV		
....	; Handle ADCMEMx overflow	
RETI	; Return	5

B. דוגמאות :

(1) בדוגמה זו תתבצע **דגימה בודדת** של ערוץ A3 (כשמתחי הרפרנס הם $V_{R-} = 0V$, $V_{R+} = AV_{CC} = 3.3V$) וביצוע בדיקה האם $A0 > 0.5 \cdot AV_{CC}$ אם כן, לד המחובר ל- P9.0 יידלק אחרת, ייכבה. מעלים בתוכנה את ADC12SC ל-1' לתחילת ביצוע דגימה, בסיום הדגימה ADC12SC מתאפס אוטומטית. משך הדגימה הוא 16 מחזורים של ADC12CLK. בין דגימה לדגימה בלולאה המרכזית הבקר נכנס למצב LPM0 לצורך שמירת אנרגיה. לפני יצאה מ- ISR הבקר יוצא ממצב שינה לצורך ביצוע דגימה בודדת פעם נוספת. לאחר שתטענו את הקוד לבקר במצב Active Application תוכלו להוציא מהמחולל גל ריבועי של $0V - 3.3V$ בתדר 2Hz ולחברו לרגל הבקר P6.3. הבחינו בלד המחובר ל- P9.0 מהבהב בתדר הגל הריבועי.

```
#include <msp430xG46x.h>

;-----
RSEG   CSTACK           ; Define stack segment
;-----
RSEG   CODE              ; Assemble to Flash memory
;-----
RESET      mov.w  #SFE(CSTACK),SP           ; Initialize stackpointer
StopWDT    mov.w  #WDTPW+WDTHOLD,&WDTCTL    ; Stop WDT
SetupADC12 mov.w  #SHT0_2+ADC12ON,&ADC12CTL0 ; Sampling time, ADC12 on
           mov.w  #SHP+CSTARTADD_3,&ADC12CTL1
           mov.b  #INCH_3,&ADC12MCTL3        ; Analog input is A3, VR+=3.3v VR-=0v
           mov.w  #0x08,&ADC12IE            ; Enable interrupt
           bis.w  #ENC,&ADC12CTL0
           bis.b  #0x08,&P6SEL               ; P6.3 ADC option select
SetupP9     bis.b  #0xff,&P9DIR              ; P9.0 output
           bic.b  #0xff,&P9OUT

Mainloop   bis.w  #ADC12SC,&ADC12CTL0        ; Start sampling/conversion
           bis.w  #CPUOFF+GIE,SR            ; LPM0, ADC12_ISR will force exit
           jmp    Mainloop                  ; Again
;-----
ADC12_ISR  ; Exit LPM0 on reti
;-----
           bic.b  #0x01,&P9OUT              ; P9.0 = 0
           cmp.w  #07FFh,&ADC12MEM3         ; ADC12MEM = A3 > 0.5AVcc?
           jlo    WakeUp
           bis.b  #0x01,&P9OUT              ; P9.0 = 1
WakeUp     bic.w  #CPUOFF,0(SP)             ; Exit LPM0 on reti
           reti
;-----
COMMON INTVEC           ; Interrupt Vectors
;-----
ORG  ADC12_VECTOR        ; ADC12 Vector
DW   ADC12_ISR
ORG  RESET_VECTOR        ; POR, ext. Reset
DW   RESET
END
```

(2) בדוגמה זו תתבצע דגימה מחזורית של סדרת ערוצים (3 ערוצים A5 – A3) והכנסת ערך הדגימות לזיכרון ה-RAM. לצורך בחינת הקוד, לאחר שתטענו את הקוד לבקר במצב Debugger תוכלו להוציא מהמחולל גל משולש של 0v – 3v בתדר 1kHz ולחברו לרגל הבקר P6.3. לאחר מכן חברו את כניסות P6.5, P6.4, לכניסת P6.3 לצורך חיבור 3 הערוצים במקביל, כך שלמעשה נדגום את הגל המשולש ב-3 ערוצים באופן טורי במטרה להבחין בשינוי הליניארי בין הדגימות (ערכי הדגימות נוכל לראות בחלון memory מתחילת כתובת 1100h). כניסה ל-ISR לצורך שמירת ערכי הדגימות ב-RAM תתבצע רק לאחר דגימת הערוץ האחרון בסדרה A5. עד לדגימת הערוץ האחרון ה-CPU במצב LPM0 ורק המודול ADC12 בפעולה.

```
#include <msp430xG46x.h>
A3result EQU 01100h ; Channel A3 results
A4result EQU 01110h ; Channel A4 results
A5result EQU 01120h ; Channel A5 results

;-----
RSEG CSTACK ; Define stack segment
;-----
RSEG CODE ; Assemble to Flash memory
;-----
RESET mov.w #SFE(CSTACK),SP ; Initialize stackpointer
StopWDT mov.w #WDTPW+WDTHOLD,&WDTCTL ; Stop watchdog
bis.b #0x38,&P6SEL ; Analog inputs A3,A4,A5

SetupADC12 mov.w #SHT0_8+MSC+ADC12ON,&ADC12CTL0
mov.w #SHP+CONSEQ_3+CSTARTADD_3,&ADC12CTL1
mov.w #0x20,&ADC12IE ; Enable ADC12IFG.5 for ADC12MEM5
mov.b #INCH_3,&ADC12MCTL3 ; A3 goes to MEM3
mov.b #INCH_4,&ADC12MCTL4 ; A4 goes to MEM4
mov.b #EOS+INCH_5,&ADC12MCTL5 ; A5 goes to MEM5, end of sequence
clr.w R5 ; Clear pointer

Mainloop bis.w #ENC,&ADC12CTL0 ; Enable conversions
bis.w #ADC12SC,&ADC12CTL0 ; Start conversions
bis.w #CPUOFF+GIE,SR ; Hold in LPM0, Enable interrupts
jmp Mainloop ; Again

;-----
ADC12_ISR ; Interrupt Service Routine for ADC12
;-----
mov.w &ADC12MEM3,A3result(R5) ; Move results to RAM,IFG reset
mov.w &ADC12MEM4,A4result(R5) ; Move results to RAM,IFG reset
mov.w &ADC12MEM5,A5result(R5) ; Move results to RAM,IFG reset
incd.w R5 ; Increment results table pointer
and.w #0Eh,R5 ; Modulo pointer
bic.w #CPUOFF,0(SP) ; Exit LPM0 on reti
reti

;-----
COMMON INTVEC ; Interrupt Vectors
;-----
ORG ADC12_VECTOR ; ADC12 Vector
DW ADC12_ISR
ORG RESET_VECTOR ; POR, ext. Reset
DW RESET
END
```