

דוח מכין מעבדה 7

מגיש:

דור בן לביא 301315115

בר פורטנוי 308244185

A חלק תיאורטי – דוח מכין

1. הסבר את המושג analog signal chain ואיך הבקר קשור לכך.

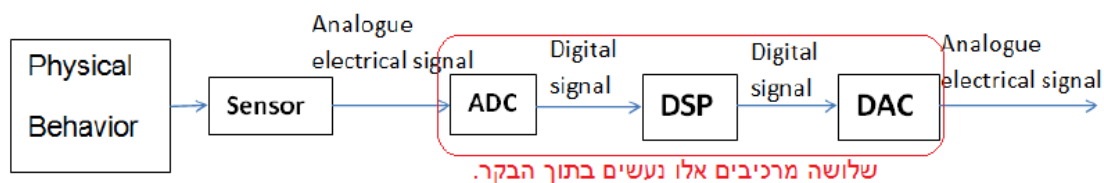
פתרון:

המושג Analog Signal Chain מתאר את המסלול המקשר בין העולם האנלוגי מחוץ לבקר (מתח רציף) לעולם הדיגיטלי (עיבוד נתונים בתוך הבקר). כדי לעבוד בעולם האמיתי יש צורך בממשק חומרתי הממיר ערכים רציפים של מתח למספר בינארי בעל m סיביות.

- בעזרת החיישנים מתבצעת מדידה של תופעה פיסיקלית והמרתה למתח רציף (המהווה כניסה למודול ADC12 שבתוך הבקר).

- המתח הרציף מומר לבקר דרך מודול ADC12 למספר בינארי בגודל 12 ביט שעליו ניתן לבצע עיבוד נתונים.

- המידע המעובד הינו דיגיטלי והוא מועבר למודול חומרתי בשם DAC12 הממיר אותו למתח אנלוגי (המהווה מוצא של הבקר).



2. הסבר במילים את המושגים הבאים וההבדל ביניהם: Quantized, Sampled Signal, Analog Signal, Digital Signal, Signal.

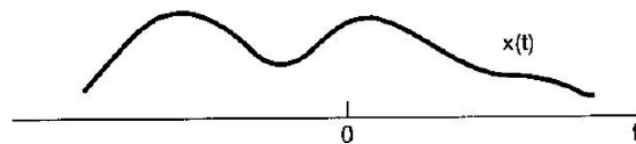
פתרון:

Signal

אות הוא תנודה, גודל כלשהו התלוי בזמן או במרחב, או הפרעה בעלי משמעות כלשהי.

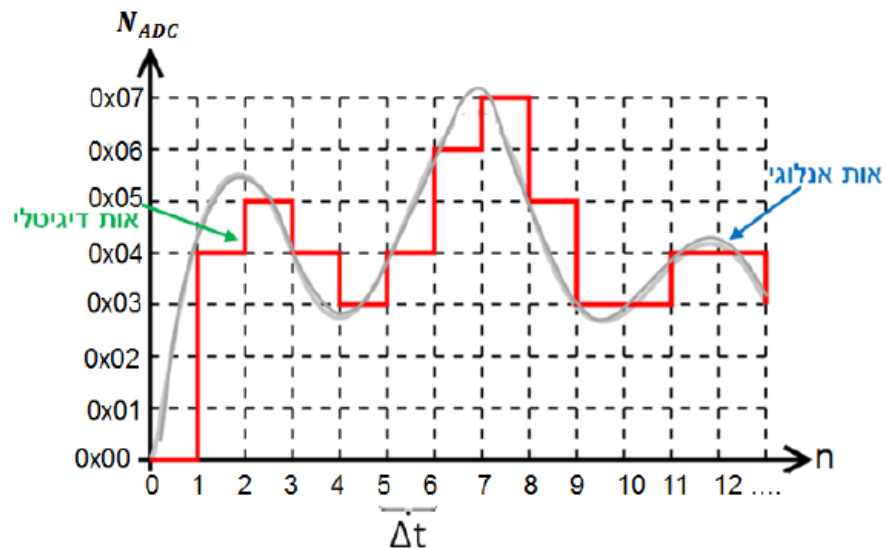
Analog Signal

אות אנלוגי $x(t)$ מורכב מערכים רציפים ולכן אות זה רציף בציר הזמן ובציר המתח. כל מקטע שנבחר מורכב מאינסוף נקודות.



Digital Signal

אות דיגיטלי הינו בדיד בציר הזמן ובדיד בציר המתח. הוא שילוב של אות דגום ואות בדיד ולכן לכל ערך דגימה בציר הזמן יש ערך מתאים בציר המתח.

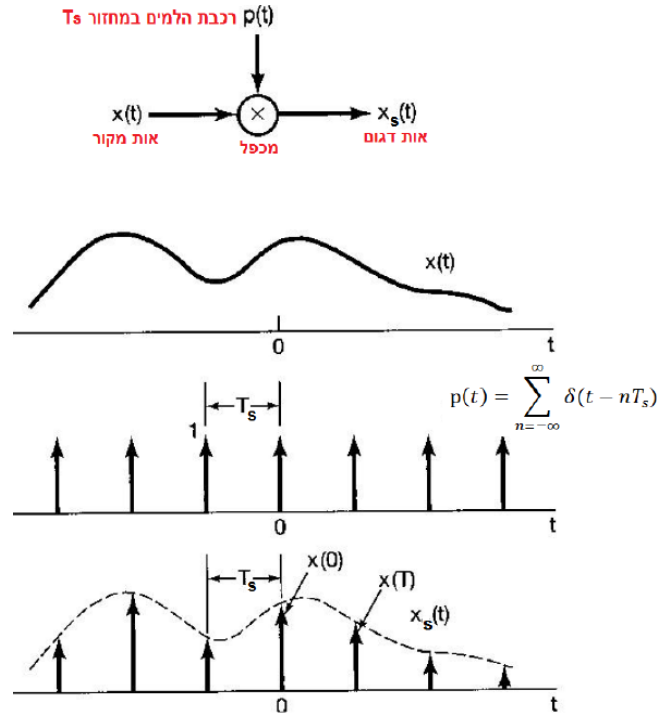


$$N_{ADC} = (2^m - 1) \times \frac{V_{in} - V_{R_min}}{V_{R_max} - V_{R_min}}$$

Sampled Signal

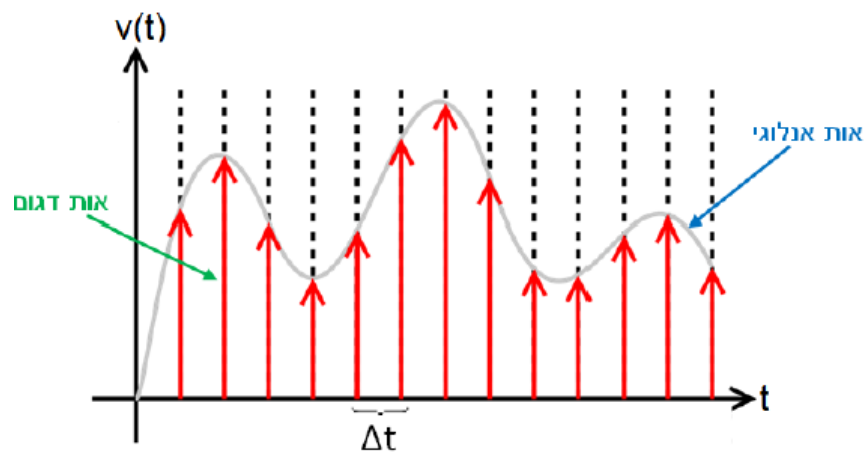
אות דגום $x_s(t)$ הינו אות מתמטי, אות זה בדיד בציר הזמן ורציף בציר המתח.

הגדרה מתמטית תיאורטית:



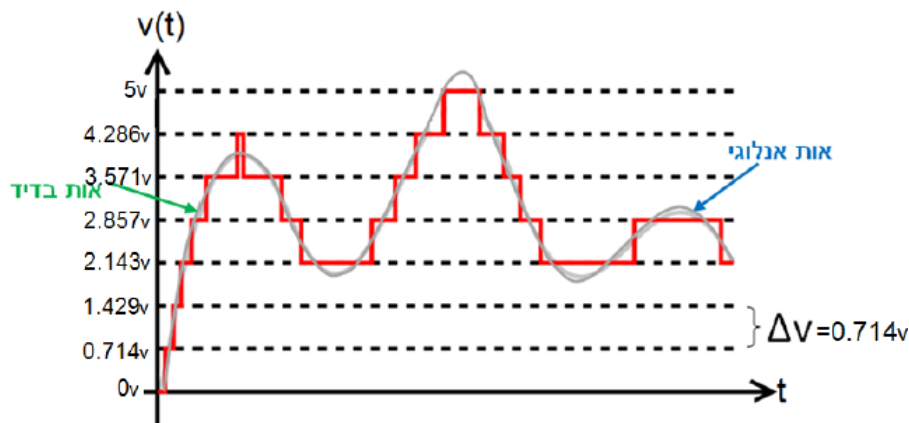
מרווח הדגימה $T_s = \Delta t \text{ [sec]}$ (נקרא גם מחזור הדגימה) קובע את מספר הנקודות לכל מקטע זמן

שנבחר. תדר הדגימה הוא $f_s = \frac{1}{T_s} \text{ [sps]}$



Quantized Signal

אות $V_{zo}(t)$ זה רציף בציר הזמן ובדיד בציר המתח. מחלקים את מרווח המתח הכולל הנמצא בין גבולות המתח $(V_{R_max} - V_{R_min})$ ל- N חלקים בגודל ΔV עבור 2^m רמות מתח (כאשר m מהווה את מספר הביטים). רמות מתח אלו הן בדידות וכל רמה משויכת למספר מתאים המיוצג בבסיס הבינארי ע"י m סיביות. רמת המתח V_{R_max} נקרא רמת ייחוס עליונה ורמת המתח V_{R_min} נקראת רמת ייחוס תחתונה. בקביעת הערכים: V_{R_max} , V_{R_min} , m , ניתן לשלוט על הרזולוציה של המתח (גודל ΔV).



3. מהם מקורות שעון ההמרה ADC12CLK ומה הצורך בכולם.

פתרון:

למודול ADC12 ישנו שעון ADC12CLK הקובע את משך זמן פעולת הדגימה ומשך פעולת ההמרה. שעון ADC12CLK מוזן דרך 4 מקורות שונים:

- ADC12OSC שעון פנימי בתדר של $5MHz$.
- ACLK שעון פנימי בתדר של $2^{15}Hz$.
- MCLK שעון פנימי בתדר של $2^{20}Hz$.
- SMCLK שעון פנימי בתדר של $2^{20}Hz$.

הצורך בתדרים השונים נובע מכך שעל מנת שתתבצע דגימה ושחזור איכותיים ומדויקים יש צורך בתדרים גבוהים (תדר נייקוויסט לדוגמה). לעיתים נדרשים גם תדרים נמוכים כדי למנוע overflow בתהליכי הדגימה וההמרה. לכן בנוסף ל- 4 מקורות השעון יש גם רכיב חומרתי שתפקידו לבצע השגחה.

4. פרט והסבר בקצרה את השיטות לקביעת מרחק בין הדגימות של מודול ADC12.

פתרון:

קביעת המרחק בין הדגימות של מודול ADC12 נעשה באופן הבא:

בחירת המקור:

ישנם ארבעה מקורות לשעון ADC12CLK והם:

- ADC12OSC שעון פנימי בתדר של $5MHz$.
- ACLK שעון פנימי בתדר של $2^{15}Hz$.
- MCLK שעון פנימי בתדר של $2^{20}Hz$.
- SMCLK שעון פנימי בתדר של $2^{20}Hz$.

בחירת המקור משנה את התדר הבסיסי אתו אנו עובדים. ובכך משפיעה על המרווח בין הדגימות.

בחירת גודל השהיה:

בחירת ADC12DIVx מחלקת את התדר הבסיסי (כלומר מכפילה את זמן המחזור) לפי הערך המוכל בביטים. (ניתן לראות את ערך החלוקה באיור הבא)

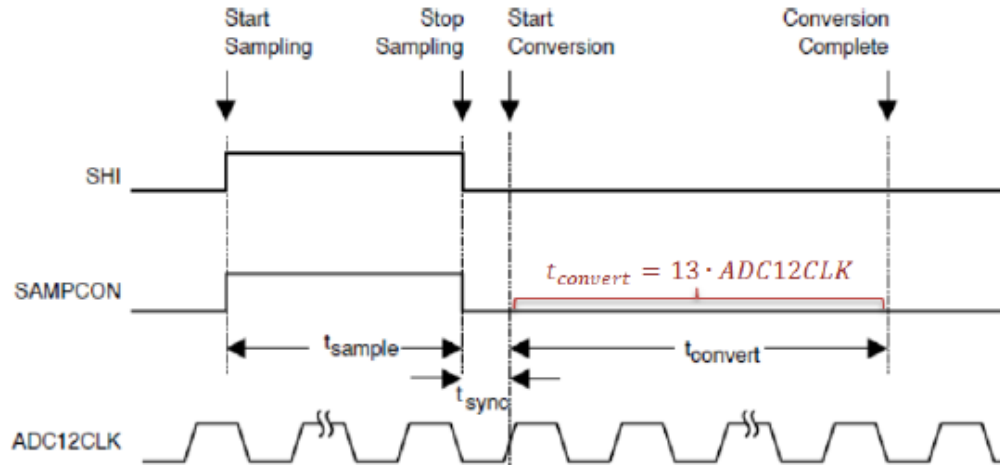
ADC12DIVx	Bits	ADC12 clock divider
	7-5	
		000 /1
		001 /2
		010 /3
		011 /4
		100 /5
		101 /6
		110 /7
		111 /8

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 ACD12CLK). כאשר SAMPCON ב – '1' לוגי מתבצעת דגימה וכאשר SAMPCON יורד ל – '0' מתחילה ההמרה של אותה דגימה. למעשה ישנן 2 שיטות שונות לדגימה המוגדרת ע"י ביט SHP, קביעה זו הינה עבור כל 16 הערוצים.

1. כאשר $SHP = 0$:

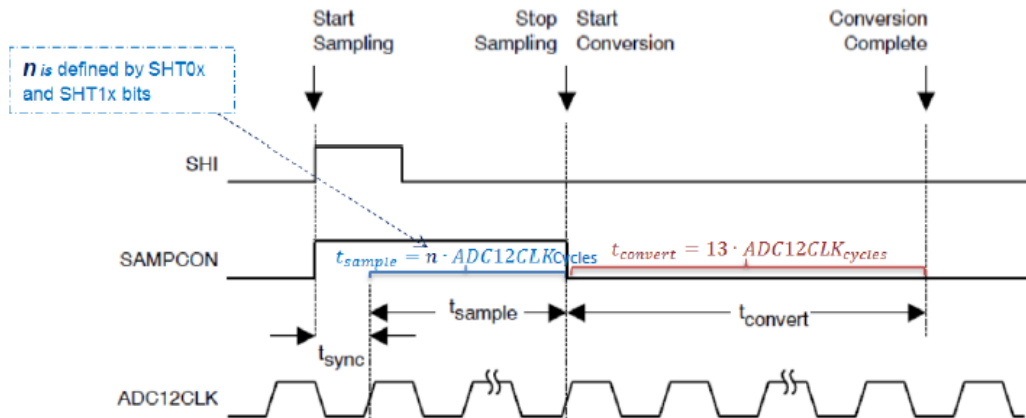
אות SHI שולט באופן ישיר על אות SAMPCON ומגדיר את משך זמן הדגימה t_{sample} .



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

2. כאשר $SHP = 1$:

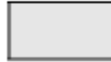
אות SHI משמש כטריגר, בעליית אות SHI נכנס לפעולה רכיב Sample Timer הקובע את משך הדגימה (משך אות SAMPCON) השווה ל- $n \cdot ADC12CLK_{cycles}$. כאשר n הוא מספר בטווח של 4 עד 1024 (נקבע ע"י הביטים SHT0x ו-SHT1x). כאשר $MSC = 0$ אות SHI מהווה טריגר לדגימה בודדת ולכן מתאים עבור אופן עבודה של דגימה בודדת, בלבד. כאשר $MSC = 1$ אות SHI מהווה טריגר לדגימה ראשונה ושאר הדגימות נעשות באופן אוטומטי מיד לאחר סיום המרת הדגימה הקודמת.



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

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

5. בדוגמה 2 בקובץ Tutorial_7.2 חשב על סמך הקוד וידע תיאורטי את המרווח בין כל 2 דגימות של המודול?

פתרון:

הקוד בדוגמה 2 הינו:

```
#include <msp430G46x.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
```

ניתן לראות בקוד ש $SHT0 = 8$, כלומר מתבצעים 256 מחזורי שעון $ADC12CLK$. הדגימה תהיה מחזורית מאחר ו- $MSC = 1$.

בנוסף, $SHP = 1$ ולכן משך הדגימה שווה ל- $n \cdot ADC12CLK_{cycles}$.

כיוון שלא נבחר שעון ספציפי, ברירת המחדל תהיה ADC12OSC כלומר, תדירות השעון הינה $ADC12CLK = 5MHz$. ולא נבחרה חלוקה בביטים ADC12DIVx ולכן המרווח בין כל 2 דגימות עוקבות יהיה:

$$5MHz = 5 \cdot 10^6 Hz$$

ישנם 256 מחזורי שעון שבהם מתבצעת הדגימה +13 מחזורי שעון שבהם מתבצעת ההמרה. סה"כ 269 מחזורי שעון בתדר 5 מגה-הרץ. כלומר, מרווח הזמן הכולל בין כל 2 דגימות יהיה:

$$t_{sample} + t_{convert} + t_{sync} = t_{total}$$

זמן סנכרון זניח יחסית וקבוע ולכן מרווח הדגימה נקבע ע"י הסכום של זמן הדגימה וזמן ההמרה. את מרווח הדגימה נקבל ע"י הנוסחה:

זמן מחזור*מספר מחזורים = סה"כ מרווח הזמן בין כל 2 דגימות.

$$\frac{269}{5MHz} = 53.8 \mu sec$$

6. הסבר בקצרה את 4 אופני העבודה של מודול ADC12, רשום דוגמה על הצורך בשימוש בכל אחד מהם.

פתרון:

1. דגימה בודדת של הערוץ:
 - כאשר ביט ADC12SC מהווה טריגר להמרת דגימה, אם ברצוננו לבצע דגימה נוספת נעלה שוב את ביט ADC12SC ל- '1' (הוא יורד אוטומטית לאחר הדגימה).
 - כאשר מקורות אחרים מהווים טריגר לדגימה נדרש לכבות ולהדליק את ביט ENC בין המרה להמרה.
 - לעצירת פעולת המרת הדגימה נוריד את ENC ל- '0'.

דוגמה:

ניתן להשתמש באופן עבודה זה על מנת לבצע דגימות מהירות מאוד ולמנוע overflow ע"כ שבכל סיום דגימה "נעיר" את ה- PC ונתחיל באופן מלאכותי דגימה חדשה. כך ניתן גם לבדוק את קצב הדגימה המקסימלי (כפי שעשינו בשיעור).

2. דגימה מחזורית של הערוץ:

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.

דוגמה:

נשתמש באופן עבודה זה לצורך דגימה מחזורית של ערוץ יחיד. במצב זה, בסוף כל דגימה והמרה, תתבצע באופן מידי דגימה נוספת. באופן עבודה זה, ישנם זמני עבודה קבועים עבור דגימה והמרה.

3. דגימה בודדת של מספר ערוצים באופן טורי:

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).

דוגמה:

באופן עבודה זה ניתן למדוד כמה ערוצים באופן נקודתי או לבחור לחזור על הפעולה ככל שנרצה על שיתקיים תנאי כלשהו, כגון: מתח גבוה מ- 1.5v. ועם התנאי הזה נוכל לעצור את הלולאה ע"י איפוס CONSEQx ולאחר מכן את ENC.

4. דגימה מחזורית של מספר ערוצים באופן טורי:

- לעצירת פעולת המרת הדגימות נוריד את ENC ל – '0' ולאחר שהמרת הערוץ בסדרה יסתיים תתבצע עצירה. (לעצירה מידית נאפס את CONSEQx ולאחר מכן את ENC).
- בחירת ערוץ (מתוך ה – 16) לביצוע דגימה נוכחית נקבע ע"י מצביע CSTARTADDx. (לפי ערך ביטים של CSTARTADDx ברגיסטר ADC12CTL1 שערכו נע בין 0 ל – 0x0f). כאשר מדובר בדגימת ערוץ אחד נכתוב בביטים CSTARTADDx את מספר הרגיסטר ADC12MEMx המקושר לערוץ.
- כאשר מדובר בדגימה של סדרת ערוצים נכתוב את מספר הרגיסטר ADC12MEMx המקושר לערוץ הראשון בסדרה, בביטים CSTARTADDx ולרגיסטר ADC12MEMx המקושר לערוץ האחרון בסדרה. נעלה את הביט EOS ל – '1' הנמצא ברגיסטר ADC12MCTLx השייך לו. במקרה זה המצביע CSTARTADDx יתקדם באופן אוטומטי ומוסתר לפי סדר הרגיסטרים וכשיגיע לערוץ שאצלו ערך EOS הוא '1' יתחיל את הסדרה מההתחלה.

דוגמה:

שימוש במצב זה מתואר בקובץ Tutorial_7.2 בדוגמה 2. בה אנו מבצעים דגימה מחזורית של מספר ערוצים ואחסון הערכים המומרים בזיכרון ה – RAM. בדוגמה זאת דוגמים גל משולש בתדר 1KHz.

הערה:

בכל 4 אופני העבודה, לסיום עבודה עם מודול ADC12 או כדי לחסוך באנרגיה כשהוא לא בשימוש, נוכל לאפס את ביט ADC12ON לאחר עצירת פעולת ההמרה.

7. הסבר את העיקרון והיתרון של שימוש ברגיסטר ADC12IV ומה התשלום בזמן ריצה ללא השימוש בו.

פתרון:

השימוש ב- ADC12IV נועד לשם ידיעת המקור לפסיקה בעת הכניסה לרוטינה. הרגיסטר יכול לקבל רק ערכים זוגיים, ובטבלה הבאה נפרט את הערכים השונים ומה כל ערך מייצג:

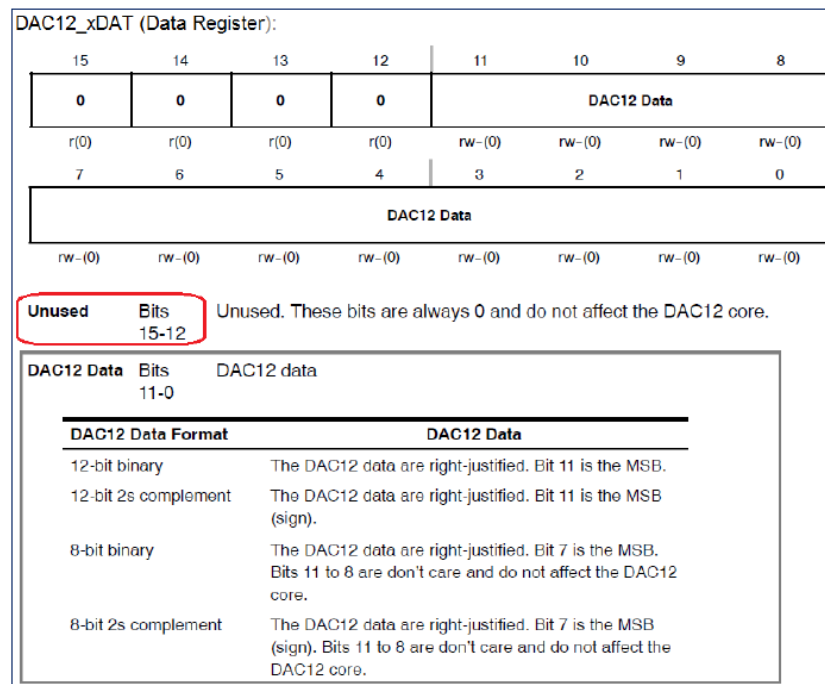
ADC12IV (Interrupt Vector Register):							
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
0	0	ADC12IVx					0
r0	r0	r-(0)	r-(0)	r-(0)	r-(0)	r-(0)	r0
ADC12IVx Bits 15-0 ADC12 interrupt vector value							
ADC12IV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority				
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				

ללא השימוש ברגיסטר זה, ניהל לבצע שמירה מיוחדת של משתנה חדש ולוודא את מקור הפסיקה ע"י שימוש בתנאים כמו במעבדות 1-4 וכתלות ביעילות הקוד שלנו, נשקיע מספר שורות כדי לבצע בדיקה של מקור הפסיקה, השמת הערך שנבחר למשתנה או רגיסטר שאותו נוכל לבדוק. זמן הריצה שנבזז יהיה גדול מאוד כיוון שהפעולות האלו חוזרות על עצמן שוב ושוב כל פעם שנכנסים לרוטינה הפסיקה.

8. הסבר את המושג data format במודול DAC12 והצורך בשימוש בו.

פתרון:

במודול DAC12 קיים רגיסטר מידע הנקרא DAC12_xDAT. כאשר $x = 0$, $x = 1$ הם 2 מודולים בלתי תלויים של DAC12. רגיסטר זה מכיל את הערך הבינארי הנדרש להמרת מתח אנלוגי.



Data format הוא מושג המתאר את האופן שבו מתייחסים לערכים המספריים המאוכסנים ברגיסטרים. ניתן לראות כי ישנם 4 מצבים אפשריים:

עבור $m = 8$:

ניתן להשתמש בפורמט הבינארי **Unsigned** או בפורמט הבינארי 'משלים ל-2'.

עבור $m = 12$:

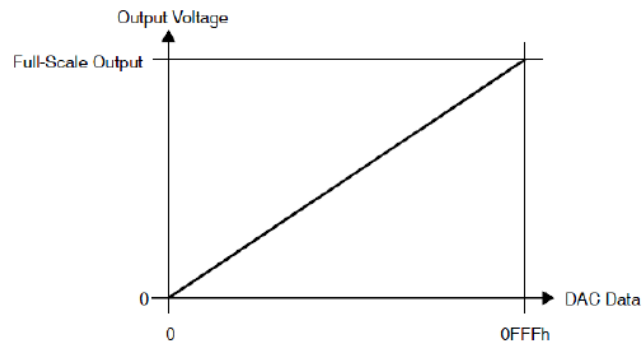
ניתן להשתמש בפורמט הבינארי **Unsigned** או בפורמט הבינארי 'משלים ל-2'.

צורך ושימוש:

- ייצוג חיובי (Unsigned)

DAC12_Xdat הוא ערך בין 0 ל- 0xFFF (רזולוציה של 12 ביט)

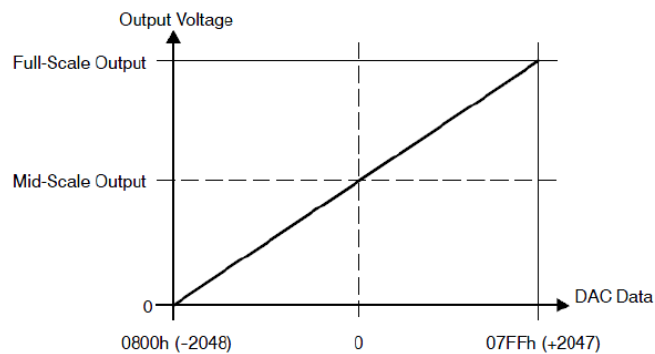
DAC12_Xdat הוא ערך בין 0 ל- 0xFF (רזולוציה של 8 ביט)



- ייצוג שלילי (2's complement)

DAC12_Xdat הוא ערך בין 0x800 ל- 0xFFF (רזולוציה של 12 ביט)

DAC12_Xdat הוא ערך בין 0x80 ל- 0xFF (רזולוציה של 8 ביט)



9. הסבר מהי רזולוציה עבור מודול DAC12 ואיך קובעים אותה.

פתרון:

כפי שנלמד בכיתה, אות דיגיטלי הוא אות רציף שעבר ברכיב ADC12 והפך לערכים בינאריים בדידים. אותם יש להחזיר לאות קוונטי (דמוי רציף). גודל המספר הבינארי אותו ממיר DAC12 למתח אנלוגי הוא באורך 12 ביט (או 8 ביט) ונמצא ברגיסטר DAC12_xDAT, זו למעשה קביעת הרזולוציה של DAC12 (מרחק בין רמות מתח סמוכות) ע"י הביטים DAC12IR, DAC12RES ברגיסטר DAC12_CTL.

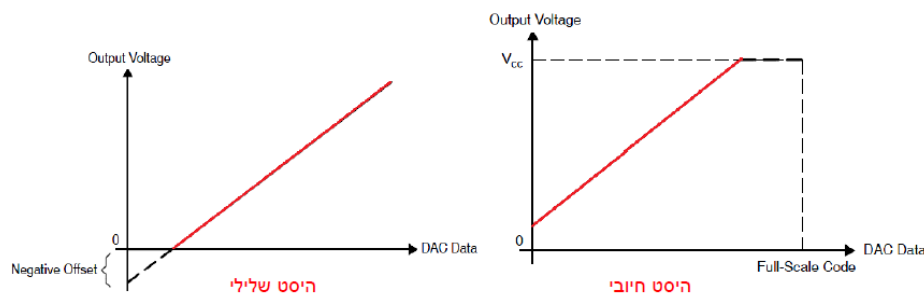
Resolution	DAC12RES	DAC12IR	Output Voltage Formula
12 bit	0	0	$V_{out} = V_{ref} \times 3 \times \frac{DAC12_xDAT}{4096}$
12 bit	0	1	$V_{out} = V_{ref} \times \frac{DAC12_xDAT}{4096}$
8 bit	1	0	$V_{out} = V_{ref} \times 3 \times \frac{DAC12_xDAT}{256}$
8 bit	1	1	$V_{out} = V_{ref} \times \frac{DAC12_xDAT}{256}$

הרזולוציה של המודול נקבעת ע"י צפיפות הערכים שאותם ניתן לייצג ע"י מספרים בינאריים במחשב. לכן, ככל שיש לנו m גדול יותר (מספר ביטים גדול יותר למספרים המייצגים את המתחים השונים) נקבל רזולוציה גבוהה יותר.

10. הסבר את המושג Self-Calibration ומתי נרצה להשתמש בו.

פתרון:

המושג Self-Calibration משמעותו "כיוול עצמי". הסיבה שבגינה יש צורך במנגנון זה היא שבמצב המעשי יש סטייה (חיובית או שלילית) כך שמתח מוצא DAC12 לא מתנהג כפי שנדרש. לשם כך יש צורך בכיוול חומרה.



במודול DAC12 ישנו מנגנון כיול עצמי לכיול היסט המתח כך שגרף המוצא ביחס לערך הרגיסטר DAC12_0DAT יהיה בשיפוע 1 וללא היסט. כאשר נעלה ל – '1' את ביט DAC12CALON יתחיל הכיול העצמי שבסיומו ערך הביט DAC12CALON מתאפס (ניתן לוודא סיום כיול באמצעות **palling**) כיול המודול יתבצע **אחרי** קינפוג המודול **ולפני** השימוש בו. בזמן הכיול ולצורך דיוק מירבי, כדאי לצמצם את פעולת הבקרה.