



S I L I C O N L A B S

C8051F310

מנחים: יואל כהן ואבי חיון

<http://site.2all.co.il/hayun/>

מבנה הרכיב

Digital I/O

- 29 PORT I/O המתואמים ל-5v
- תקשורת טורית – uart, spi, smb
- 4 טימרים
- PCA 5
- תכנות חיבור הפורטים-crossbar

Clock source

- פנימי של 24.5MHz דיוק 2%
- חיצוני –גביש, RC, C או מחולל

זכרון

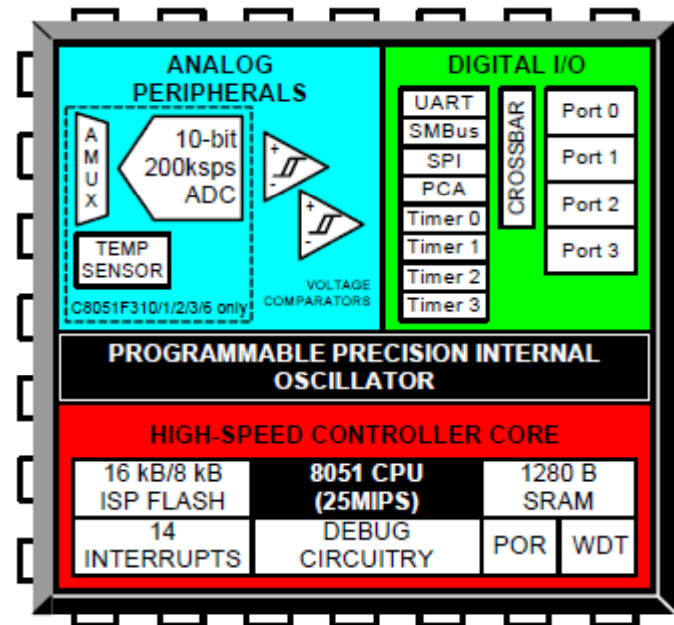
- Flash 16KB-תוכנית
- Ram –1024 +256

מתח הפעלה

- 2.7-3.6v

High speed 8051 μ c core

- 70% מההוראות מתבצעות במחזור אחד או שניים מחזורי מכונה
- עד 25 MIPS (Mega Instruction Per Second) בתדר 25MHz



Analog- Peripherals

Analog-to-Digital

- 10 סביבות ADC
- עד 200KSPS (Kilo Sample Per Second)
- עד 21 כניסות אנלוגיות
- בודדות או דיפרנציאליות
- Vref חיצוני או VDD
- חישן טמפ' build in

On-Chip Debug

- Break points
- Single stepping

משוויים

- 2 משוויים
- עם תכנות היסטריזיס וזמן תגובה
- יכולים להיות מתוכנתים כגורמי פסיקה

מאפיינים

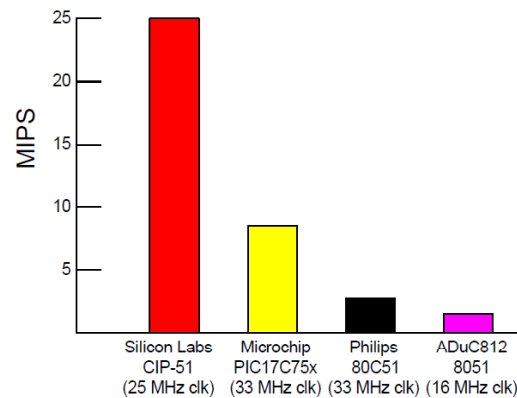
Smallest ♦

- מתח עבודה 2.7V עד 3.6V
- 32 פינים – 29 פיני פורט I/O (P0-8,P1-8,P2-8,P3-5)

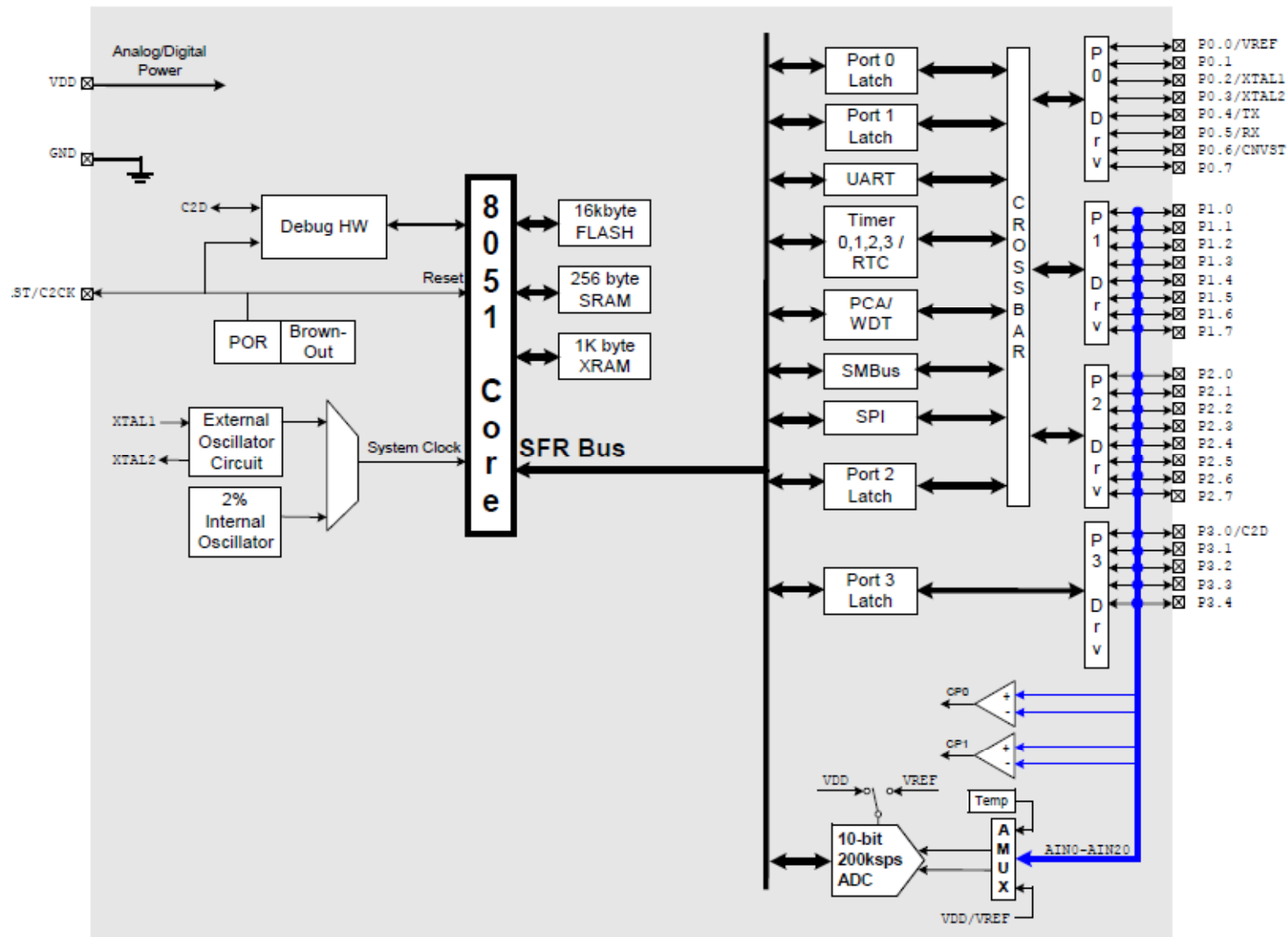


High speed 8051 μ core ♦

- 70% מההוראות מתבצעות במחזור אחד או שניים מחזורי מכונה
- עד 25 MIPS (Mega Instruction Per Second) בתדר 25MHz



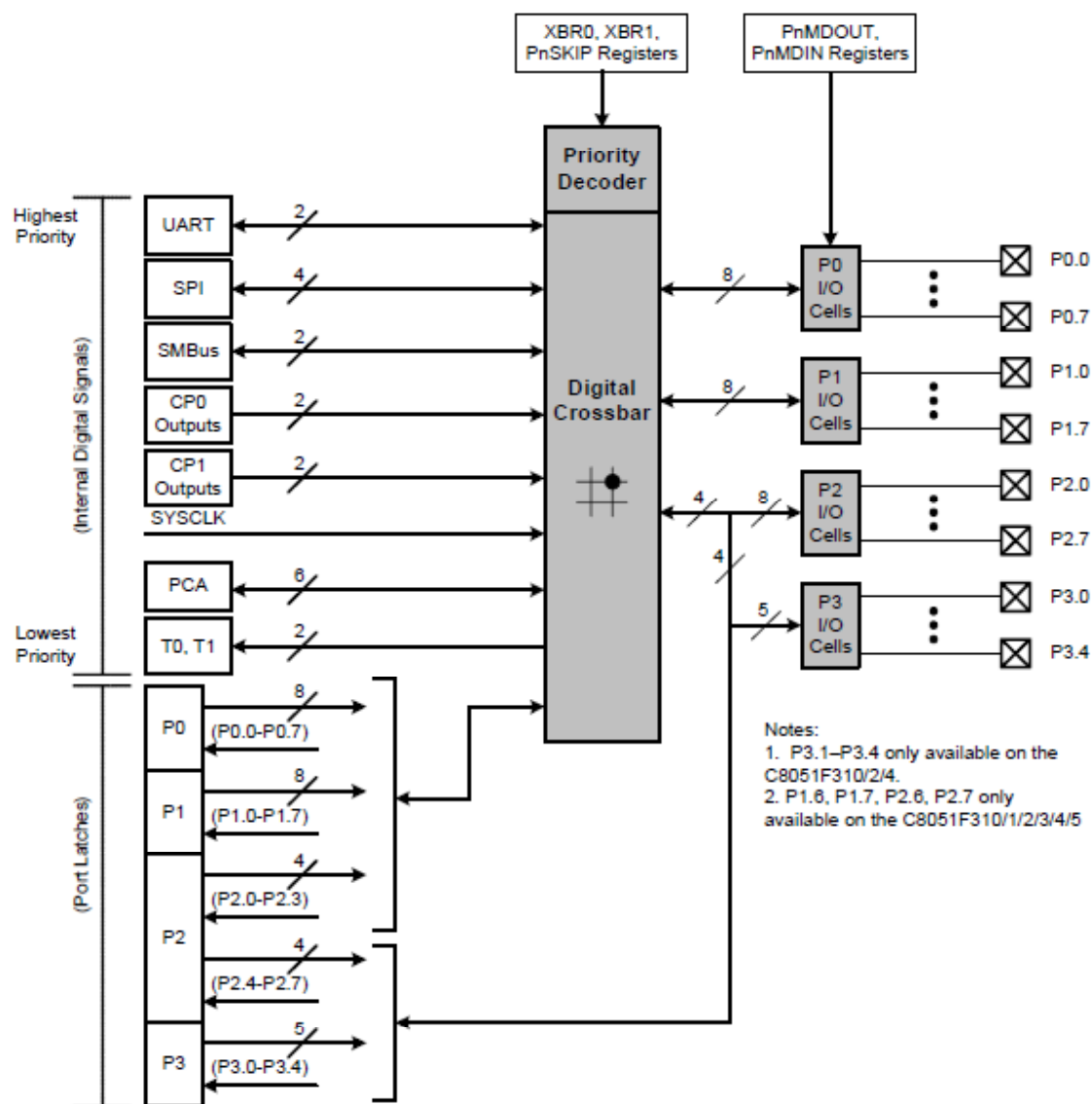
Configuration



<http://site.2all.co.il/hayun/> אבי חיון ויואל כהן

Crossbar

חלק מהפורטים
 עוברים דרך מטריצת
 חיבורים הניתנים
 לתכנות ע"י מילות
 בקרה



תכנות PORT I/O

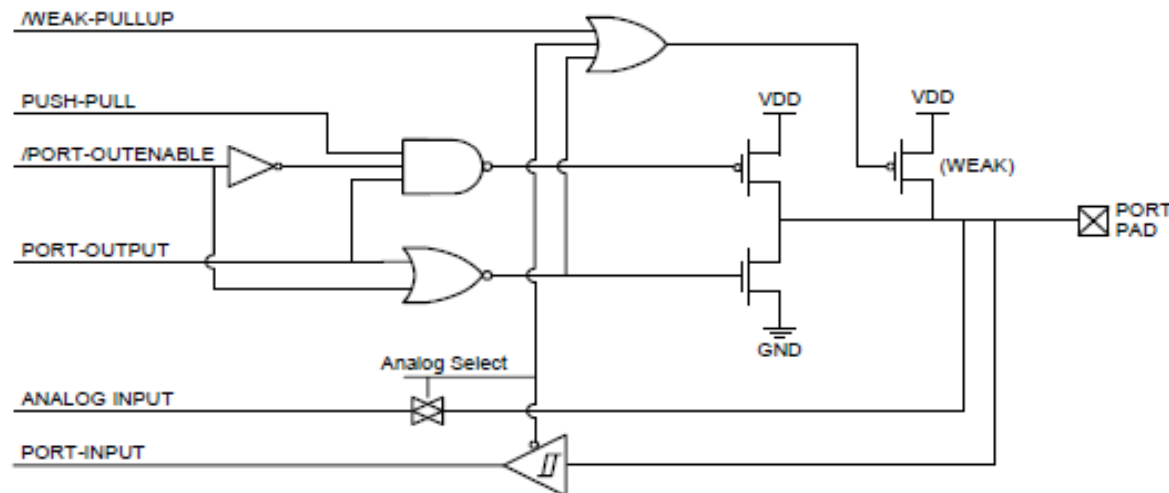


Figure 13.2. Port I/O Cell Block Diagram

◆ כל PORT I/O יכול להיות מתוכנת ל:

- Open collector
- Push- pull
- עם נגד pull-up פנימי (הערה: לכל הפורטים)
- כניסה אנלוגית

נתונים טכניים של PORT I/O

Table 13.1. Port I/O DC Electrical Characteristics

$V_{DD} = 2.7$ to 3.6 V, -40 to $+85$ °C unless otherwise specified

Parameters	Conditions	Min	Typ	Max	Units
Output High Voltage	$I_{OH} = -3$ mA, Port I/O push-pull	$V_{DD} - 0.7$	—	—	V
	$I_{OH} = -10$ μ A, Port I/O push-pull	$V_{DD} - 0.1$	—	—	
	$I_{OH} = -10$ mA, Port I/O push-pull	—	$V_{DD} - 0.8$	—	
Output Low Voltage	$I_{OL} = 8.5$ mA	—	—	0.6	V
	$I_{OL} = 10$ μ A	—	—	0.1	
	$I_{OL} = 25$ mA	—	1.0	—	
Input High Voltage		2.0	—	—	V
Input Low Voltage		—	—	0.8	V
Input Leakage Current	Weak Pullup Off	—	—	± 1	μ A
	Weak Pullup On, $V_{IN} = 0$ V	—	25	40	

♦ ניתן להגיע לזרם מוצא במצב 1 לוגי ל-10mA

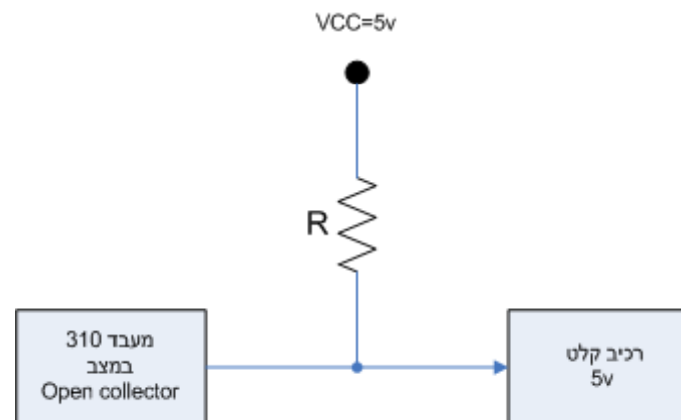
$$I_{OH}=10mA$$

♦ במצב 0 לוגי זרם מוצא עד 25mA

$$I_{OL}=25mA$$

חיבור לרכיבים הפועלים במתח 5v

- ♦ פורט של המעבד מסוגל לעמוד במתח של 5v , לכן פורט המשמש ככניסה מקבל את הרמות הלוגיות הנדרשות כאשר הוא עובד עם רכיבי 5V
- ♦ פורט המשמש כמוצא מספק קרוב ל-0v במצב 0 לוגי ו-3v במצב 1 לוגי, לכן צריך לבדוק אם הרכיבים של 5v עומדים בתנאים הלוגים ובחסינות לרעש.
- ♦ במקרה ולא ניתן לתאם בין המעבד של 3v לבין רכיבי 5v ע"י חיבור במצב open collector (קיימת אפשרות זו ע"י תכנות) עם נגד חיצוני



Oscillators

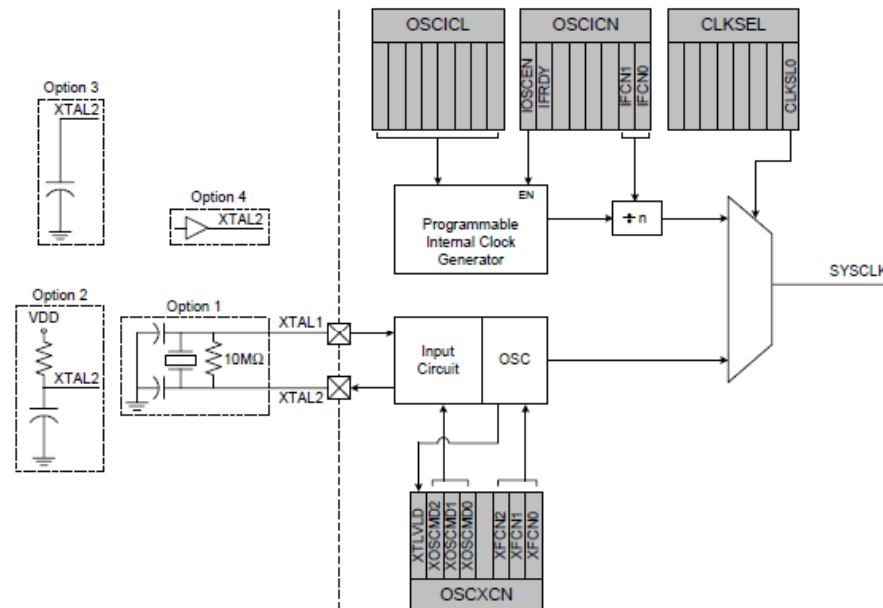


Figure 12.1. Oscillator Diagram

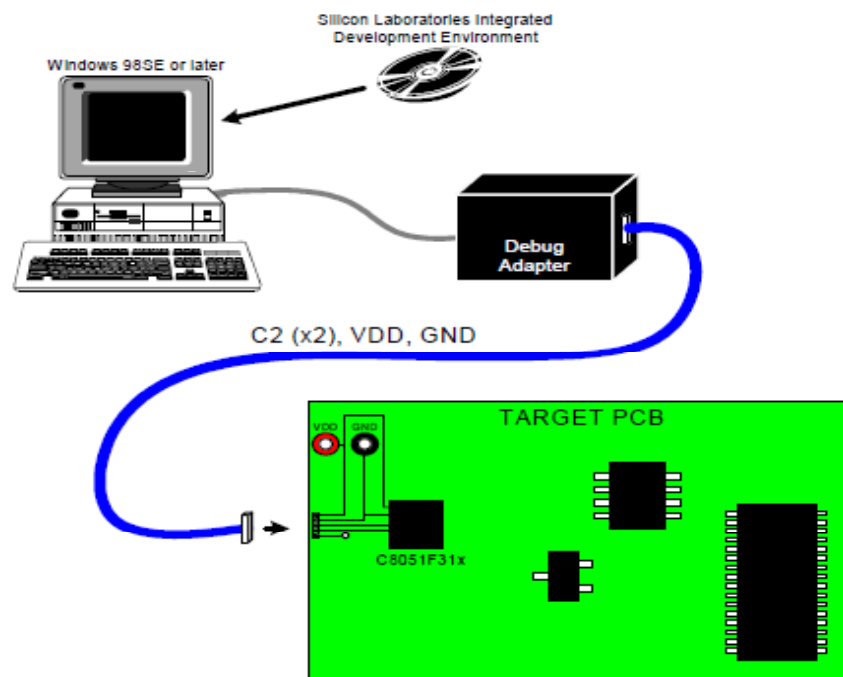
♦ ניתן לתכנות ע"י אוגרי בקרה

➤ CK פנימי או חיצוני

➤ חבור חיצוני – גביש, נגד-קבל, קבל או מחולל חיצוני

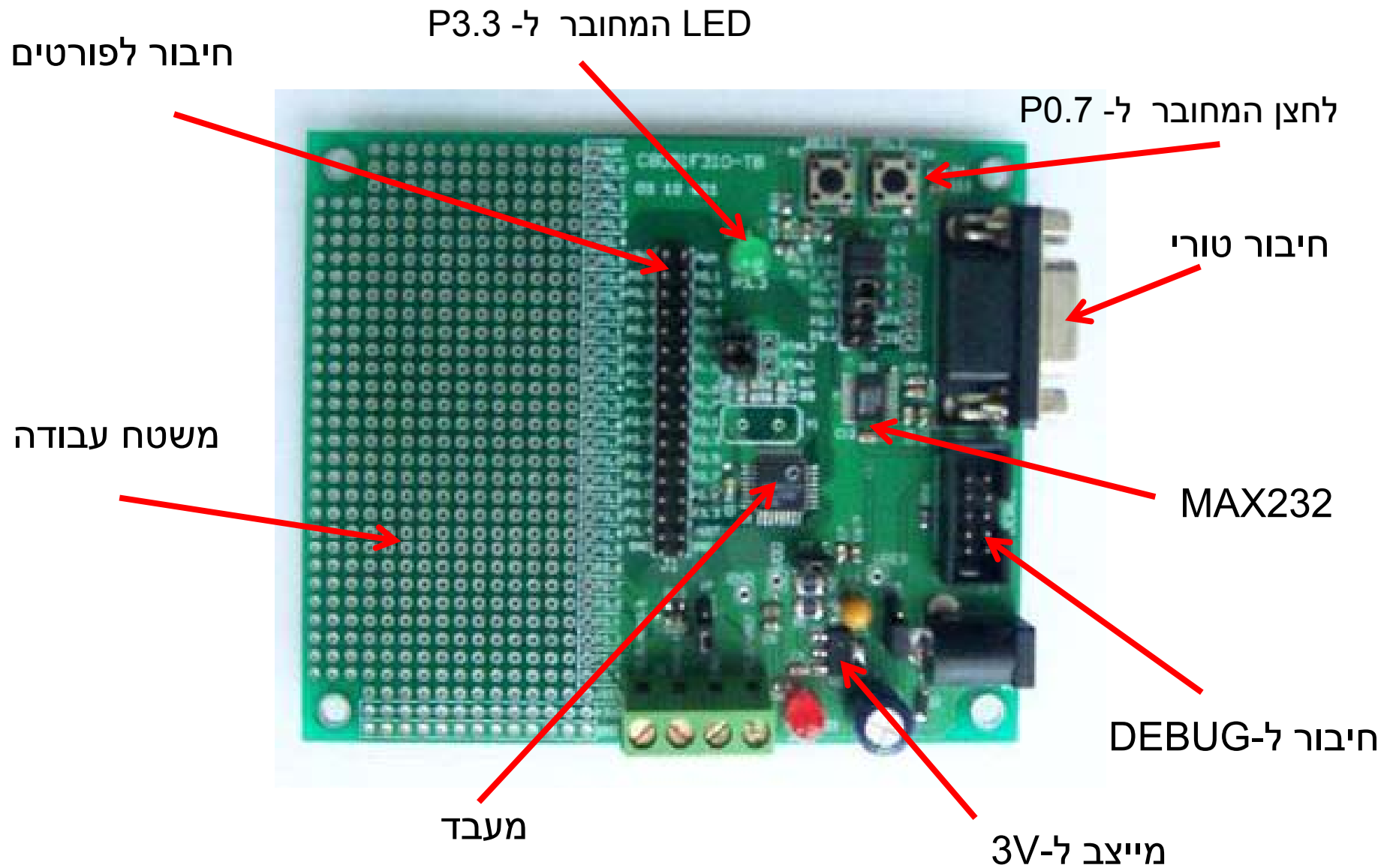
➤ CK פנימי ניתן לחלוקה ע"י מילת בקרה

On – Chip Debug



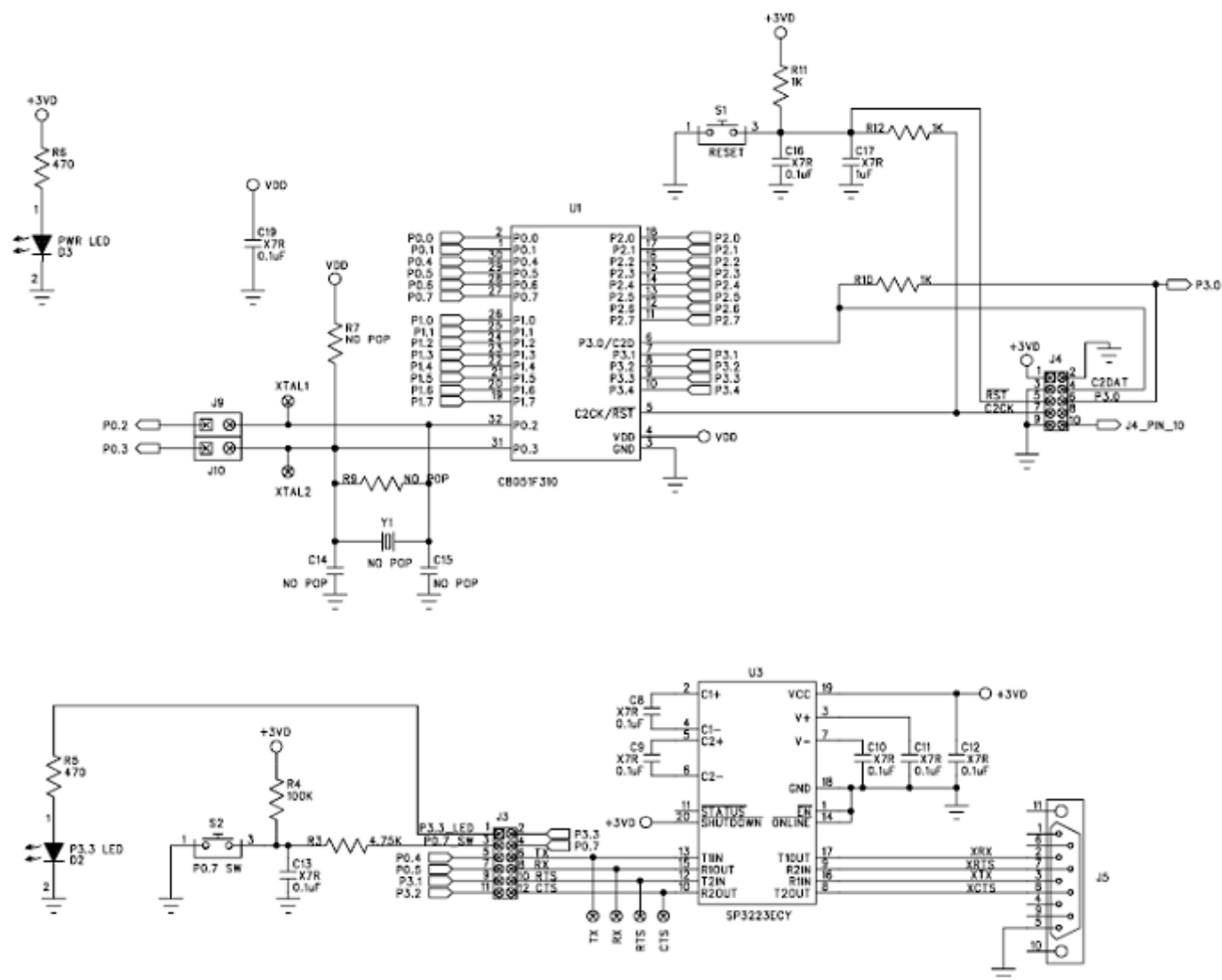
- ◆ תקשורת עם הרכיב ע"י 2 חוטים (C2) להורדה והרצת התוכנית
- ◆ אפשרות להרצה בודדת – Single Step
- ◆ נקודות עצירה – Break Points

כרטיס הפיתוח - C8051F310-DK



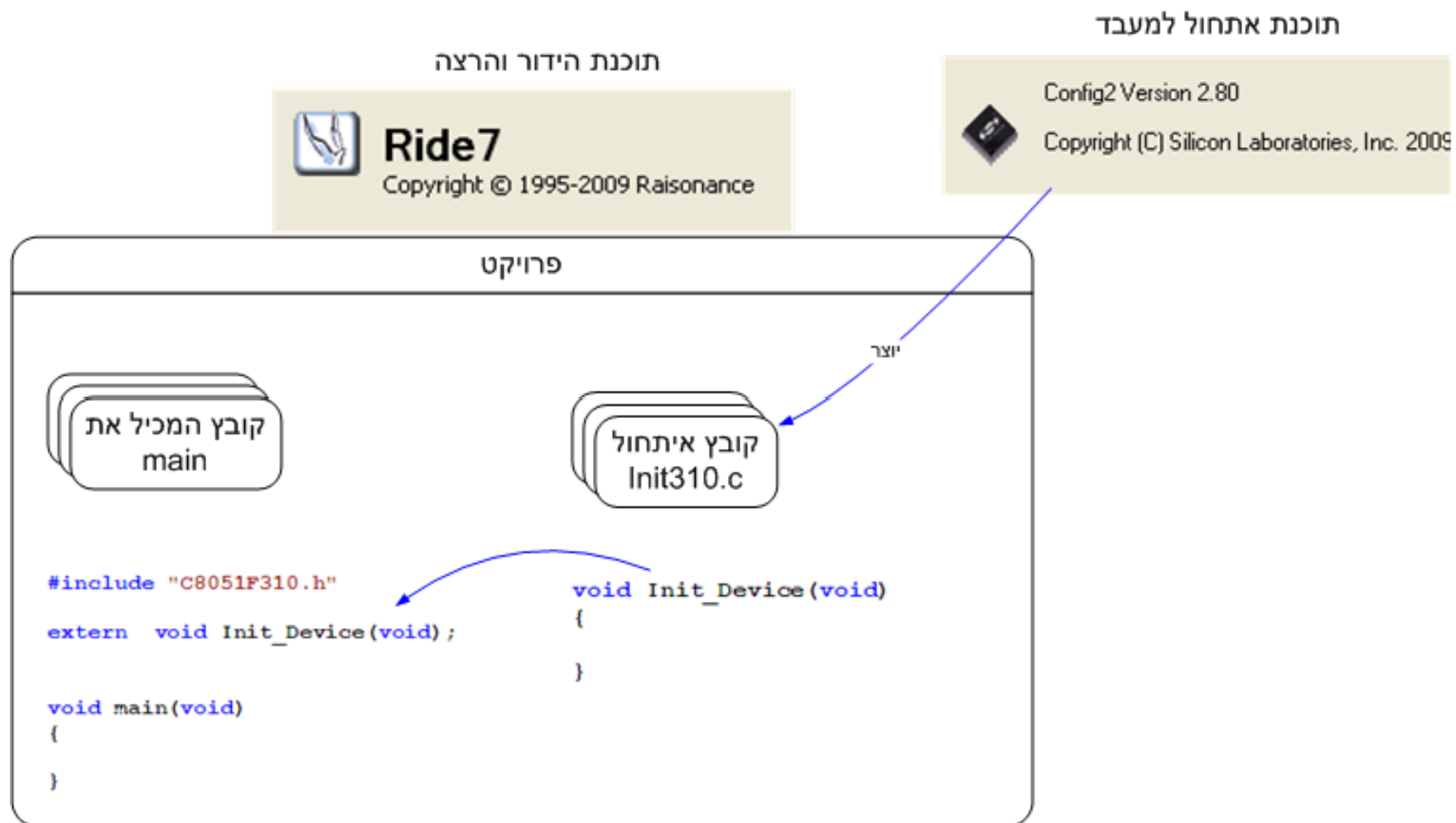
<http://site.2all.co.il/hayun/> אבי חיון ויואל כהן

סכימה חשמלית של הכרטיס



C8051F310 Target Board Schematic

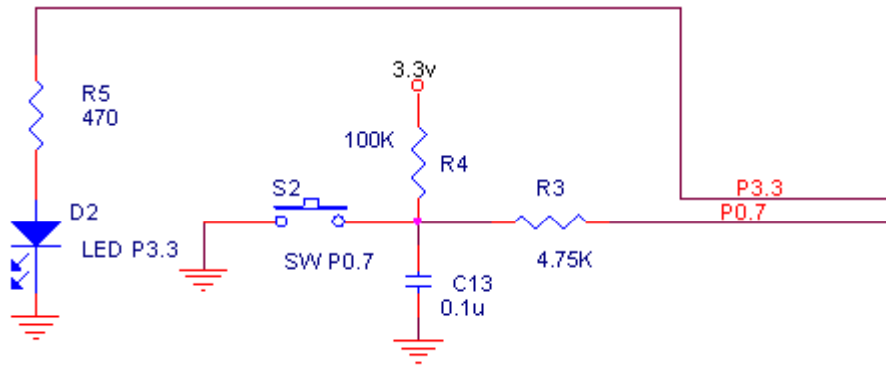
בניית פרויקט בעזרת תוכנות Ride7 ו- Config



הרצת תוכנית ראשונה

◆ ניסוי הדגמה ראשון עם הכרטיס

◆ קלט- לחצן ופלט – LED



◆ הלחצן וה- LED מחוברים בכרטיס

◆ נכתוב תוכנית ב- C להבהוב ה-LED כל עוד הלחצן לחוץ

שלבי בניית קובץ project

1. פתיחת תוכנות הידור RIDE7 ואתחול הרכיב config2
2. ע"י תוכנת config2 נגדיר את מילות הבקרה לרכיב, שייכתב לקובץ unit310.c
3. נכתוב קובץ ב-C הכולל את ה-main
4. נצרף את שני הקבצים ל-project
5. נבצע הידור
6. נריץ את התוכנית ע"י ה- DEBUGER ע"י RIDE7

תוכנית בשפת C

```
#include "C8051F310.h"

sbit LED=P3^3;
sbit SW=P0^7;

extern void Init_Device(void); // external function

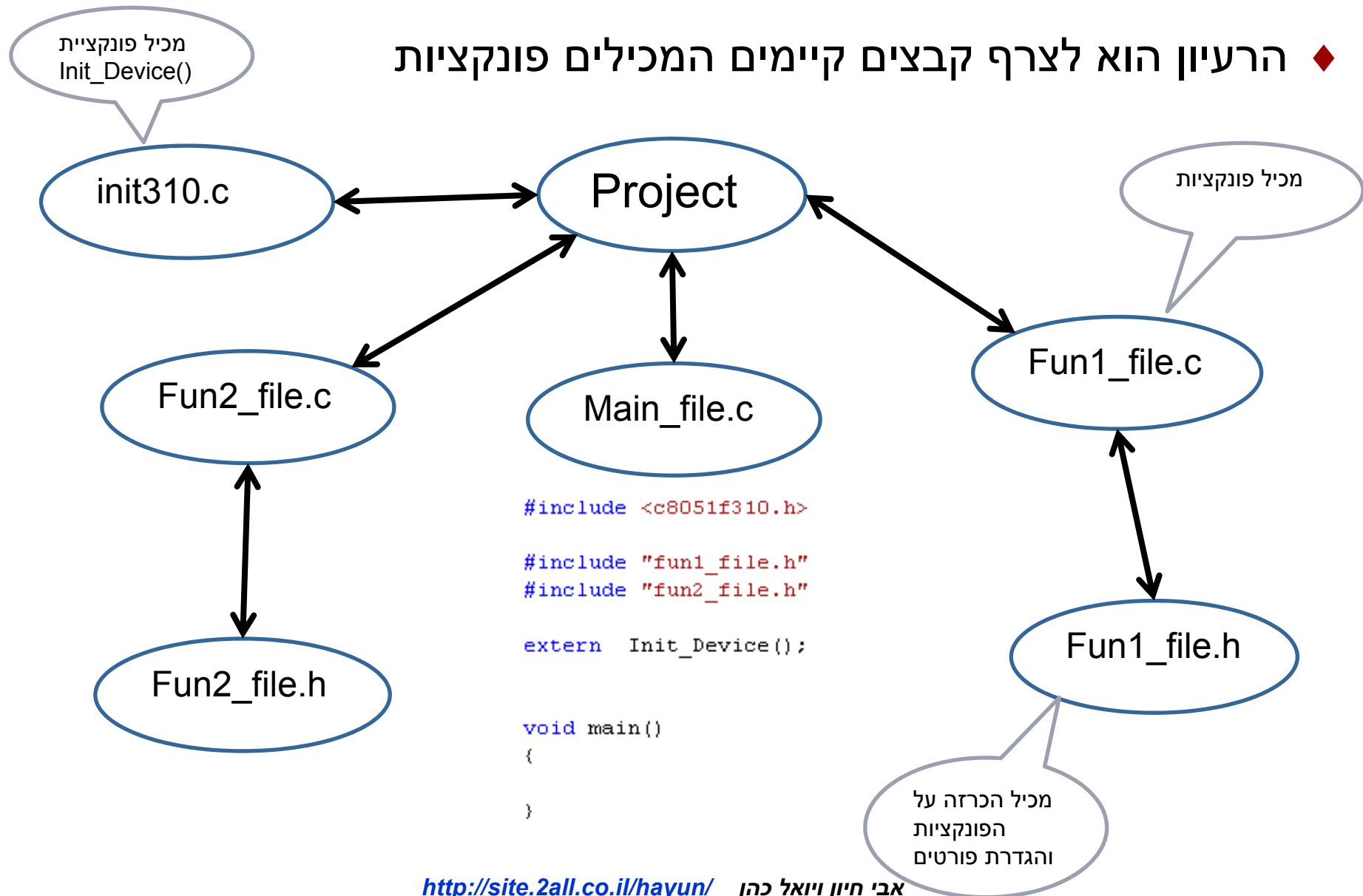
delay()
{
    unsigned int i=60000;
    while(i--);
}

void main(void)
{
    Init_Device();
    while(1)
    {
        if(SW==0)
        {
            delay();
            LED=!LED;
        }
        else
            LED=0;
    }
}
```

<http://site.2all.co.il/hayun/> אבי חיון ויואל כהן

כתיבת פרויקט הכולל מספר קבצים

♦ הרעיון הוא לצרף קבצים קיימים המכילים פונקציות



פונקציות שונות

◆ קובץ lcd_p.h מכיל:

```
lcd_config();           // אתחול LCD
lcd_line(1,10);         // הצבת סמן בשורה 1 במיקום 10
lcd_string("abcd1234"); // הצגת מחרוזת במסך
cursor(0);              //cursor off
cursor(1);              //cursor on
lcd_clr();              // נקה מסך
lcd_display('x');       // הצג תו
```

◆ קובץ wait_msec.c מכיל השהיה לפי msec

```
void T0_Wait_ms (unsigned int ms)
```

◆ קובץ ultrasonic.c מכיל מדידת מרחק ב- cm

```
int read_distance()
```

◆ קובץ motor_pwm.c מכיל ויסות מהירות מנוע DC לפי ערך PWM (255-עד 255)

```
motor_left(int pwm)
motor_right(int pwm)
```

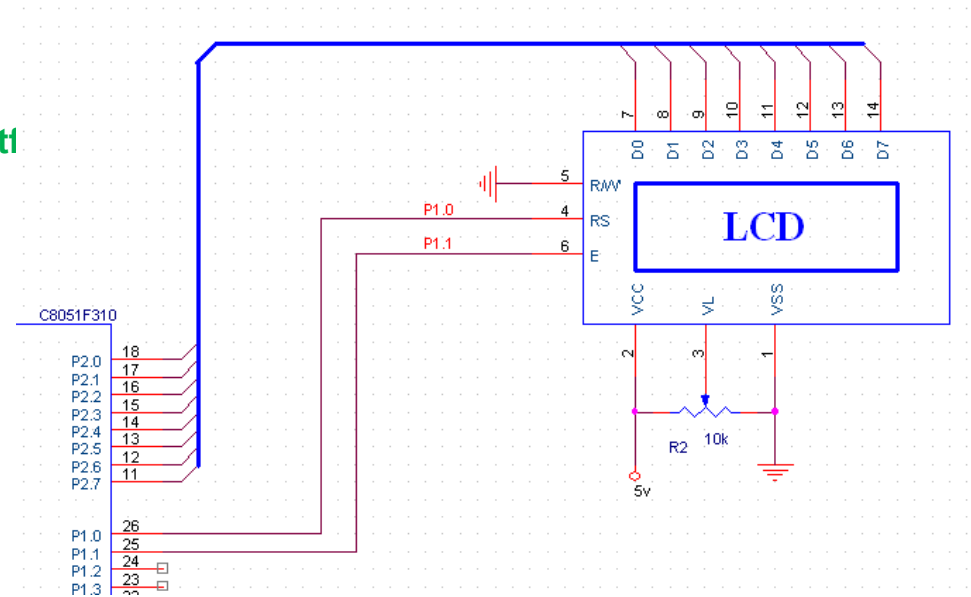
תוכנית להפעלת LCD

```
#include <c8051f310.h>
#include <stdio.h>      // for sprintf
#include "lcd_p.h"      // for lcd function

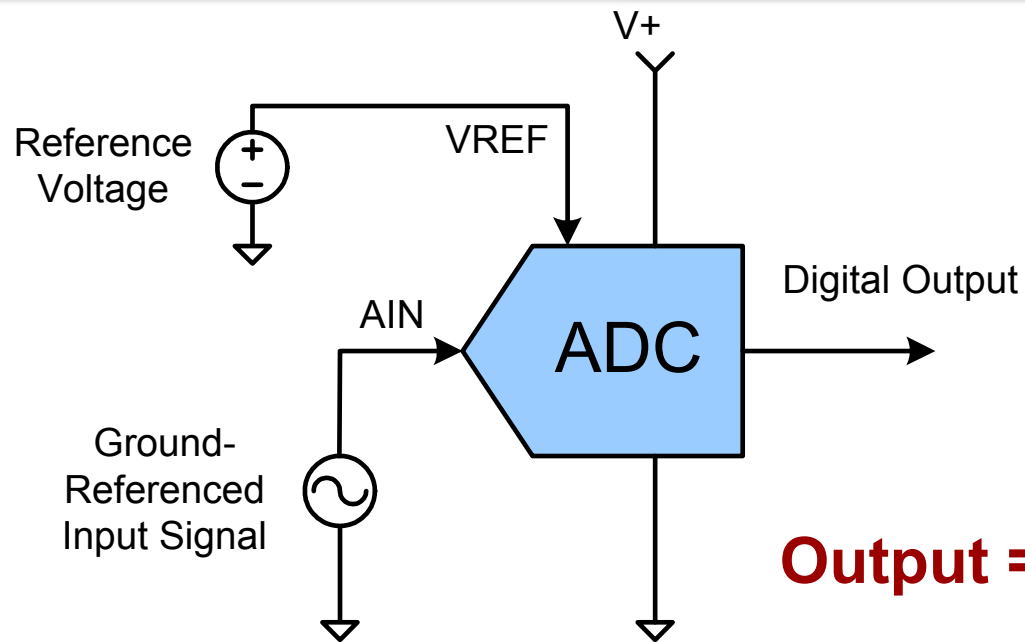
extern Init_Device();   // init 310
char str[30];           // use for LCD & sprintf

main()
{
    float x=17.5;
    Init_Device();
    lcd_config();        // init LCD
    lcd_line(1,0);       // line 1 place 0
    lcd_string("ava nagila ");
    lcd_line(2,0);
    sprintf(str,"x=%2.2f x/2=%2.2f ",x,x/2);
    lcd_string(str);

    while(1);
}
```



ממיר אנלוגי לדיגיטלי - ADC 10 סיביות

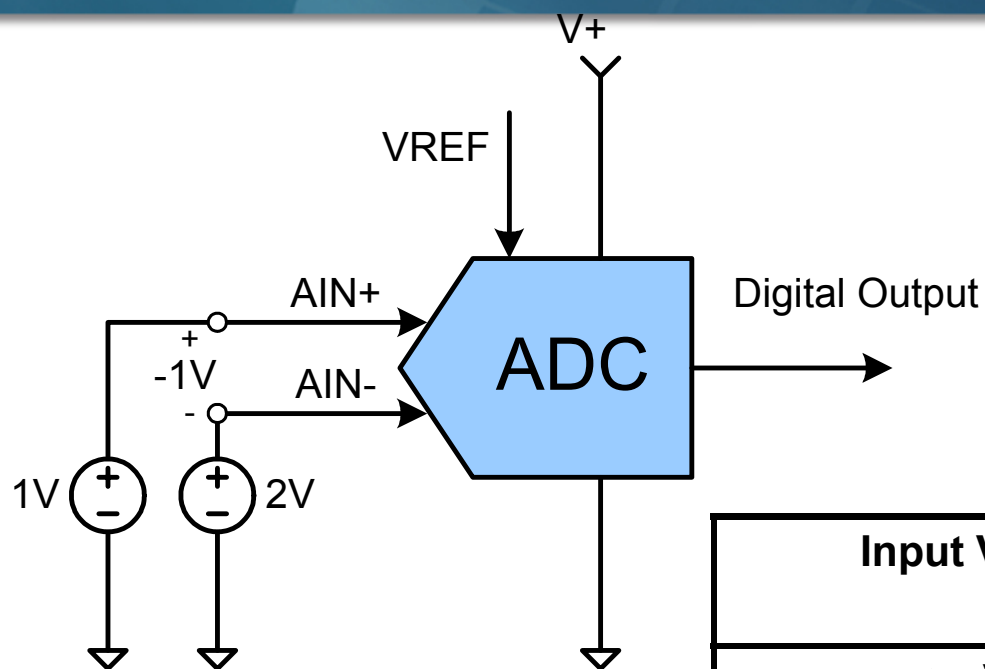


חיבור מתח כניסה אחד
המיוחס לאדמה

$$\text{Output} = (V_{IN} / V_{REF}) \times 2^N$$

Input Voltage	Output Code (10-bit)
$\geq V_{REF}$	1023 (0x03FF)*
$V_{REF} - 1 \text{ LSB}$	1023 (0x03FF)
$\frac{1}{2} V_{REF}$	512 (0x0200)
$\frac{1}{4} V_{REF}$	256 (0x0100)
0 V	0 (0x0000)

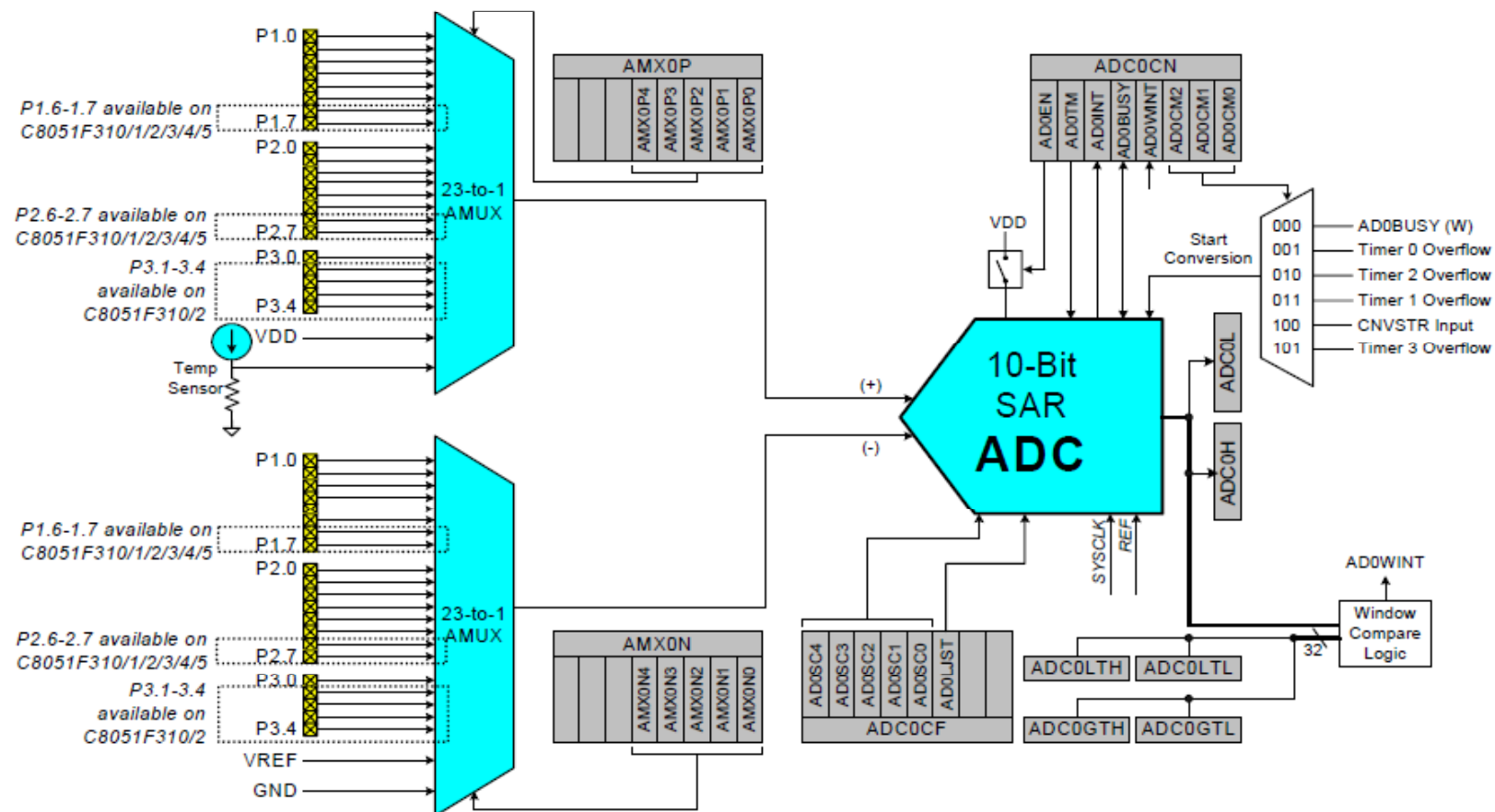
כניסה דיפרנציאלית



ערך המידע הדיגיטלי מקבל
ערכים חיוביים ושלילים לפי הפרש
המתחים

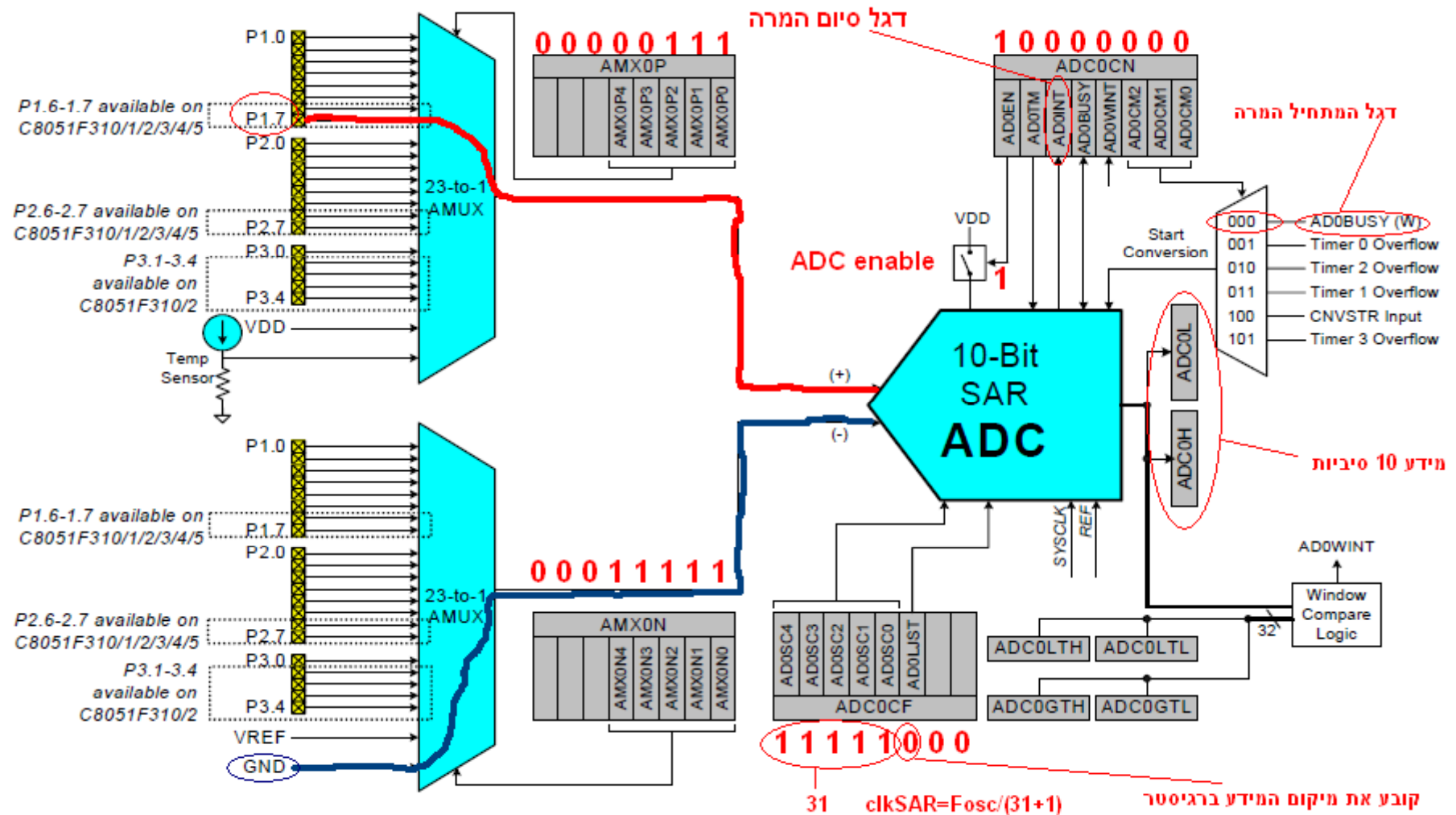
Input Voltage	Output Code (10-bit, sign extended)
$\geq V_{REF}$	511 (0x01FF)*
$V_{REF} - 1 \text{ LSB}$	511 (0x01FF)
$\frac{1}{2} V_{REF}$	256 (0x0100)
0 V	0 (0x0000)
$-\frac{1}{2} V_{REF}$	-256 (0xFF00)
$-V_{REF}$	-512 (0xFE00)
$< -V_{REF}$	-512 (0xFE00)*

דיאגראמת הממיר



- ◆ המידע האנלוגי עובר דרך 2 MUX אנלוגיים הניתנים לתכנות ע"י מילות בקרה
- ◆ הכניסות של ה-MUX מתחברות לפורטים שונים, חישן טמפ' פנימי, אדמה, VREF, VDD

דוגמא לתכנות ה-ADC

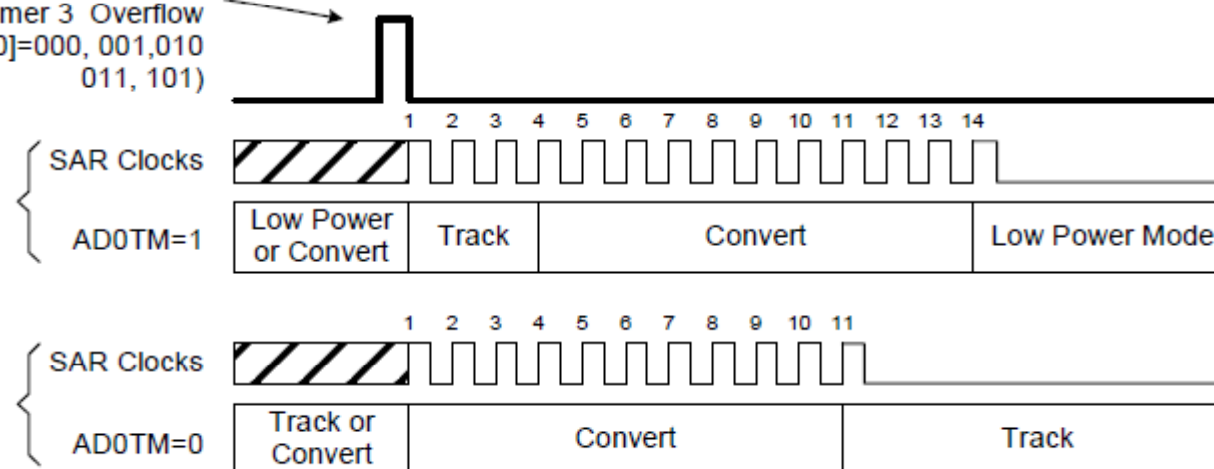


זמנים ב- ADC

Conversion Rate					
SAR Conversion Clock		—	—	3	MHz
Conversion Time in SAR Clocks		10	—	—	clocks
Track/Hold Acquisition Time		300	—	—	ns
Throughput Rate		—	—	200	ksps

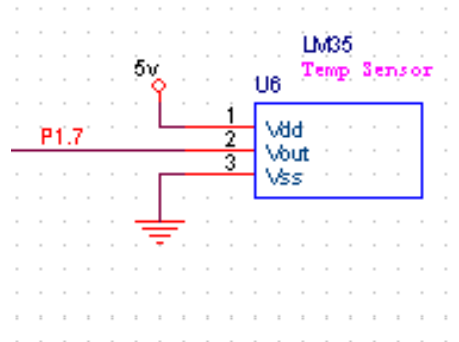
B. ADC0 Timing for Internal Trigger Source

Write '1' to AD0BUSY,
Timer 0, Timer 2,
Timer 1, Timer 3 Overflow
(AD0CM[2:0]=000, 001, 010
011, 101)



- ♦ התחלת המרה ע"י דגל או אחד הטיימרים
- ♦ זמן המרה שווה ל-10 פולסי שעון + זמן עקיבה

הרצת תוכנית עם ה- ADC0 לחיישן טמפ' LM35



♦ בעזרת תוכנת Config נקבע:

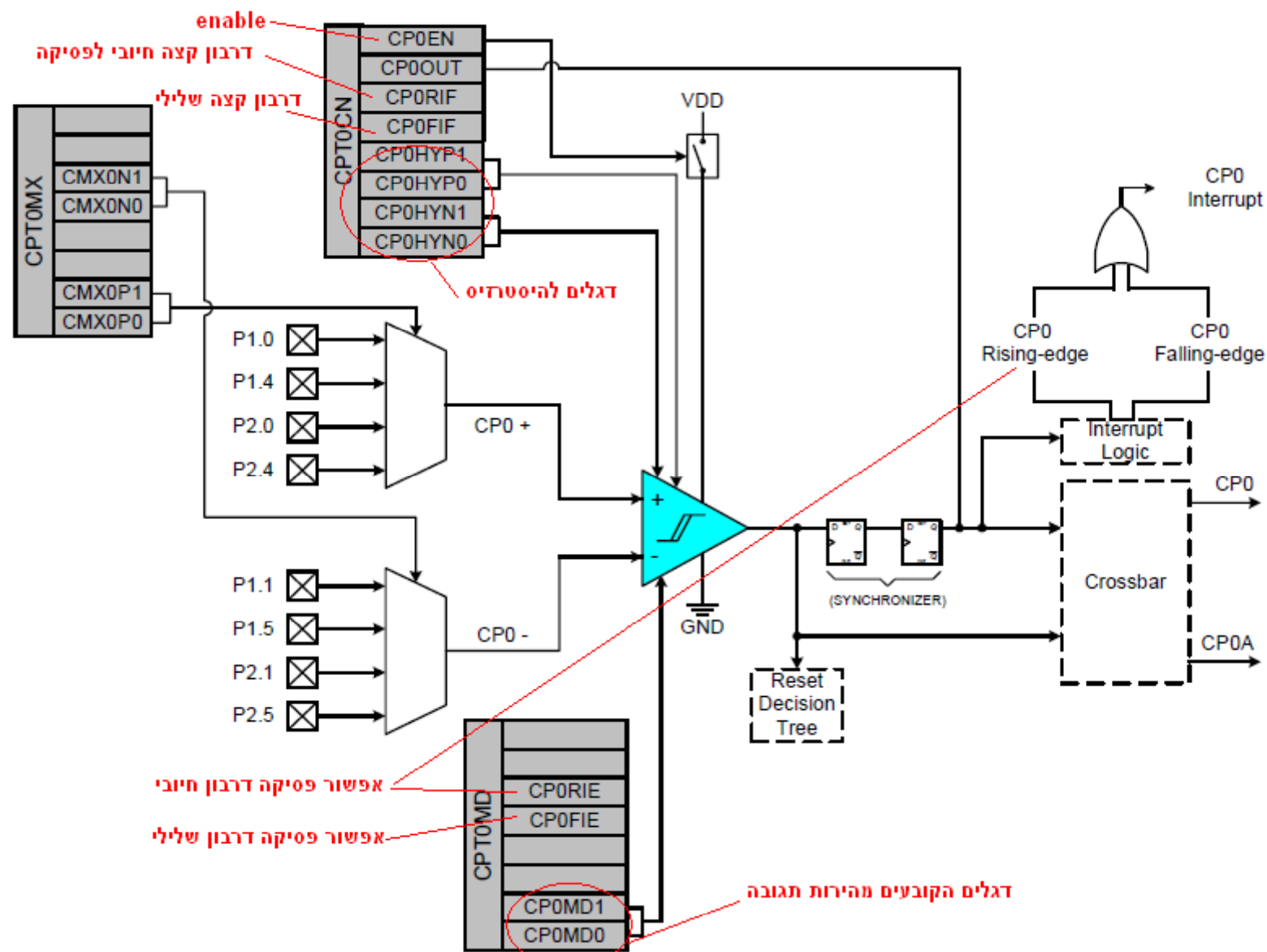
1. אפשרור הממיר – Enable ADC0
2. התחלת המרה ע"י דגל AD0BUSY
3. קביעת כניסות ה-ADC, כניסה חיובית ל-P1.7 וכניסה שלילית לאדמה
4. קובעים תדר שעון של ה-ADC (אסור לעבור תדר של 3MHz)
5. קובעים את Vref ל-VDD (3.3v)
6. קובעים את הכניסה P1.7 אנלוגית A

פונקציה ה-ADC0

♦ פונקציה להפעלת הממיר

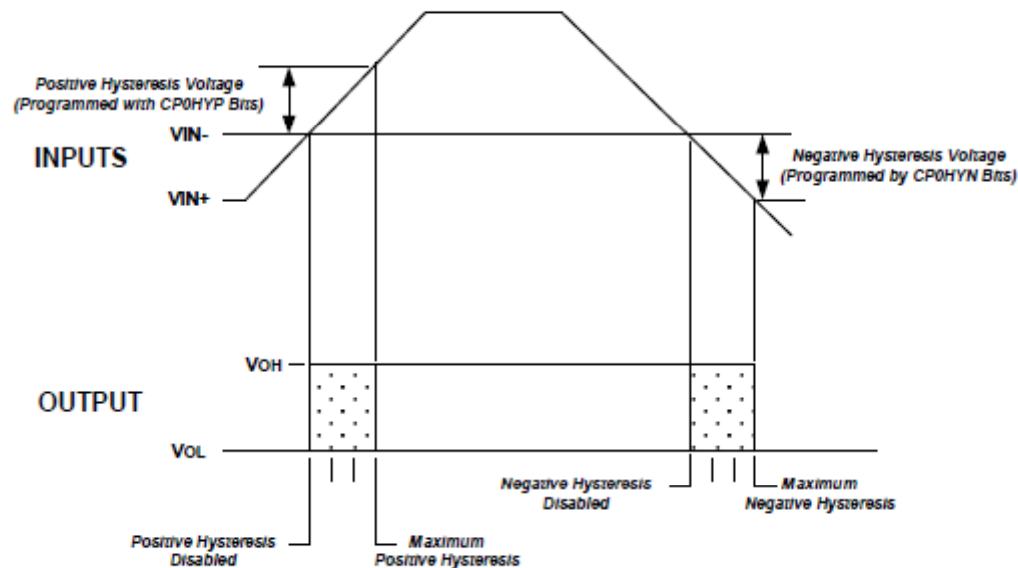
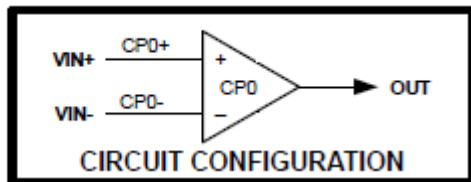
```
int ADC_IN()  
{  
    ADOBUSY=1; // start conversion  
  
    while(!ADOINT); // wait end conversion  
  
    ADOINT = 0; // clear ADC0 conversion complete flag  
    return (ADCOH*256+ADC0L); //ADC0;  
}
```

משוויים



<http://site.2all.co.il/hayun/> אבי חיון ויואל כהן

היטרזיס (חשל)



Bits3–2: CP0HYP1-0: Comparator0 Positive Hysteresis Control Bits.

00: Positive Hysteresis Disabled.

01: Positive Hysteresis = 5 mV.

10: Positive Hysteresis = 10 mV.

11: Positive Hysteresis = 20 mV.

Bits1–0: CP0HYN1-0: Comparator0 Negative Hysteresis Control Bits.

00: Negative Hysteresis Disabled.

01: Negative Hysteresis = 5 mV.

10: Negative Hysteresis = 10 mV.

11: Negative Hysteresis = 20 mV.

4 דגלים של רגיסטרים

CPT0CN,CPT1CN

קובעים את ההיסטרזיס

החיובי והשלילי בערכים:

0

5mV

10mV

20mV

זמני תגובה של המשווים

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	CP0RIE	CP0FIE	-	-	CP0MD1	CP0MD0	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9D

Bits7–6: UNUSED. Read = 00b. Write = don't care.

Bit5: CP0RIE: Comparator Rising-Edge Interrupt Enable.

0: Comparator rising-edge interrupt disabled.

1: Comparator rising-edge interrupt enabled.

Bit4: CP0FIE: Comparator Falling-Edge Interrupt Enable.

0: Comparator falling-edge interrupt disabled.

1: Comparator falling-edge interrupt enabled.

Bits1–0: CP0MD1–CP0MD0: Comparator0 Mode Select

These bits select the response time for Comparator0.

Mode	CP0MD1	CP0MD0	CP0 Response Time (TYP)
0	0	0	Fastest Response Time
1	0	1	—
2	1	0	—
3	1	1	Lowest Power Consumption

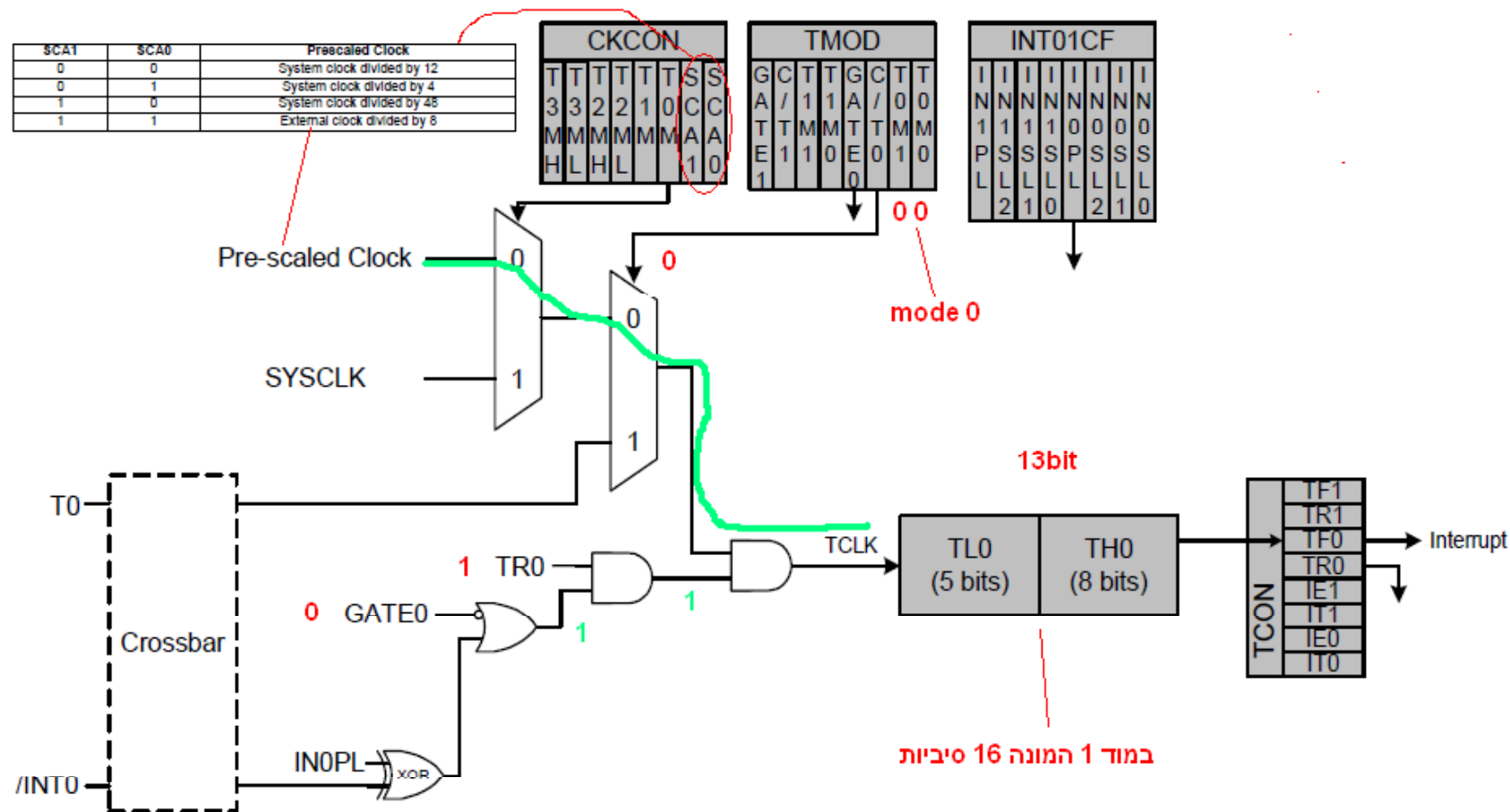
זמני תגובה נעים בין 100nsec ועד 5200nsec תלוי במוד העבודה

טיימרים

♦ 4 טיימרים במצבי עבודה שונים

Timer 0 and Timer 1 Modes:	Timer 2 Modes:	Timer 3 Modes:
13-bit counter/timer	16-bit timer with auto-reload	16-bit timer with auto-reload
16-bit counter/timer		
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)		

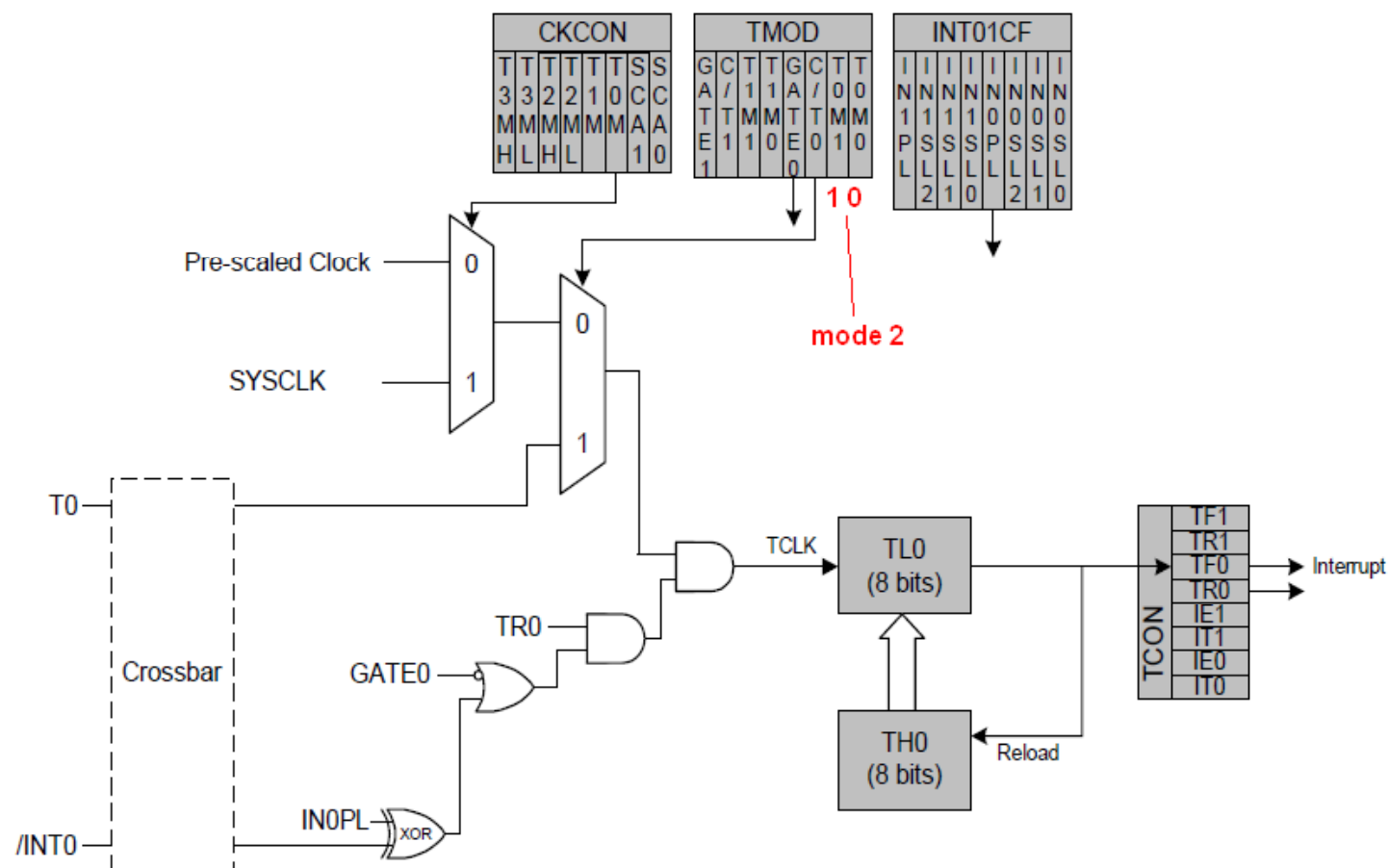
Timer0 and Timer1 mode0



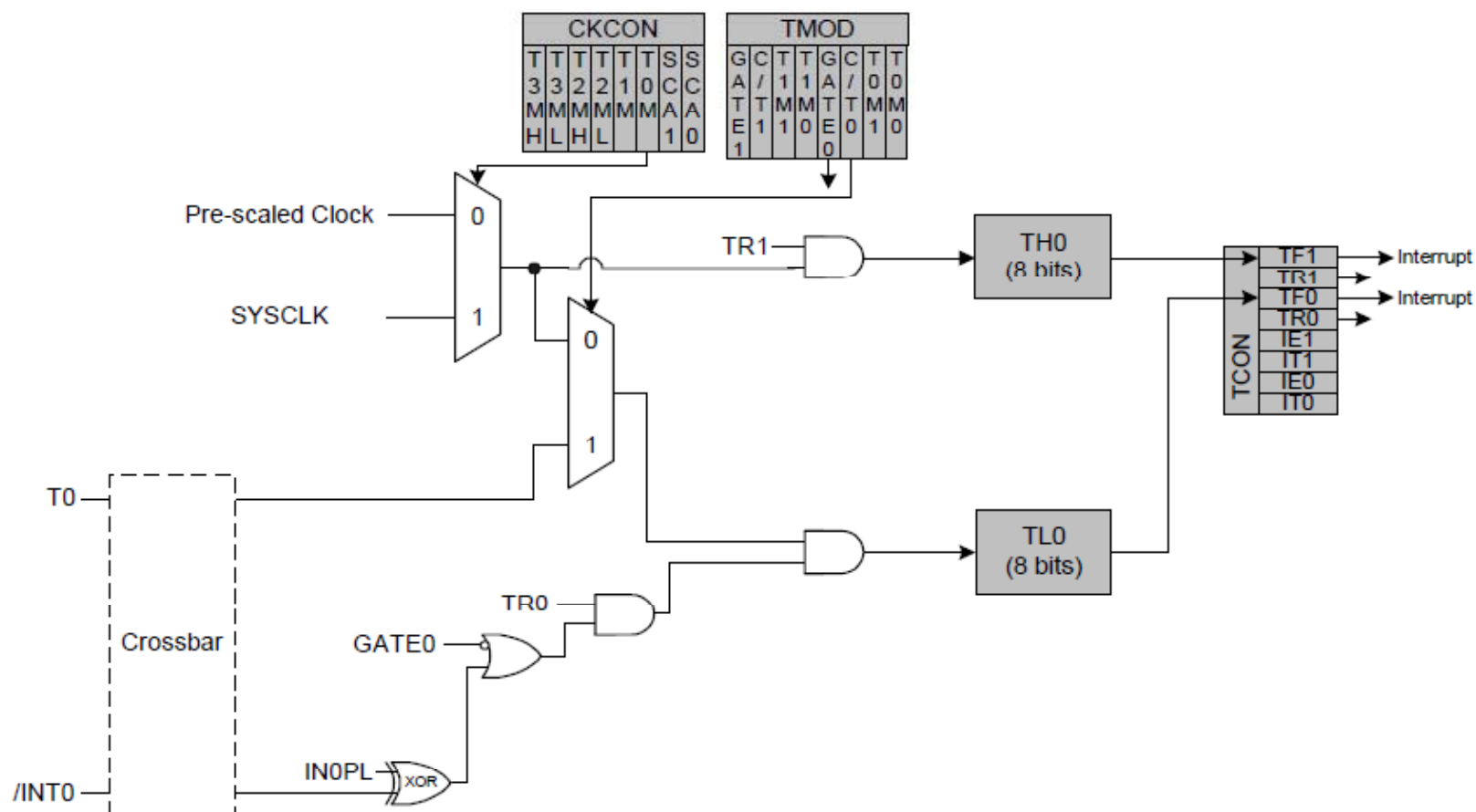
♦ ב- mode 1 המונה עובד באותה צורה, אבל עם 16 סיביות

<http://site.2all.co.il/hayun/> אבי חיון ויואל כהן

Timer0 and Timer1 mode2 - autoreload



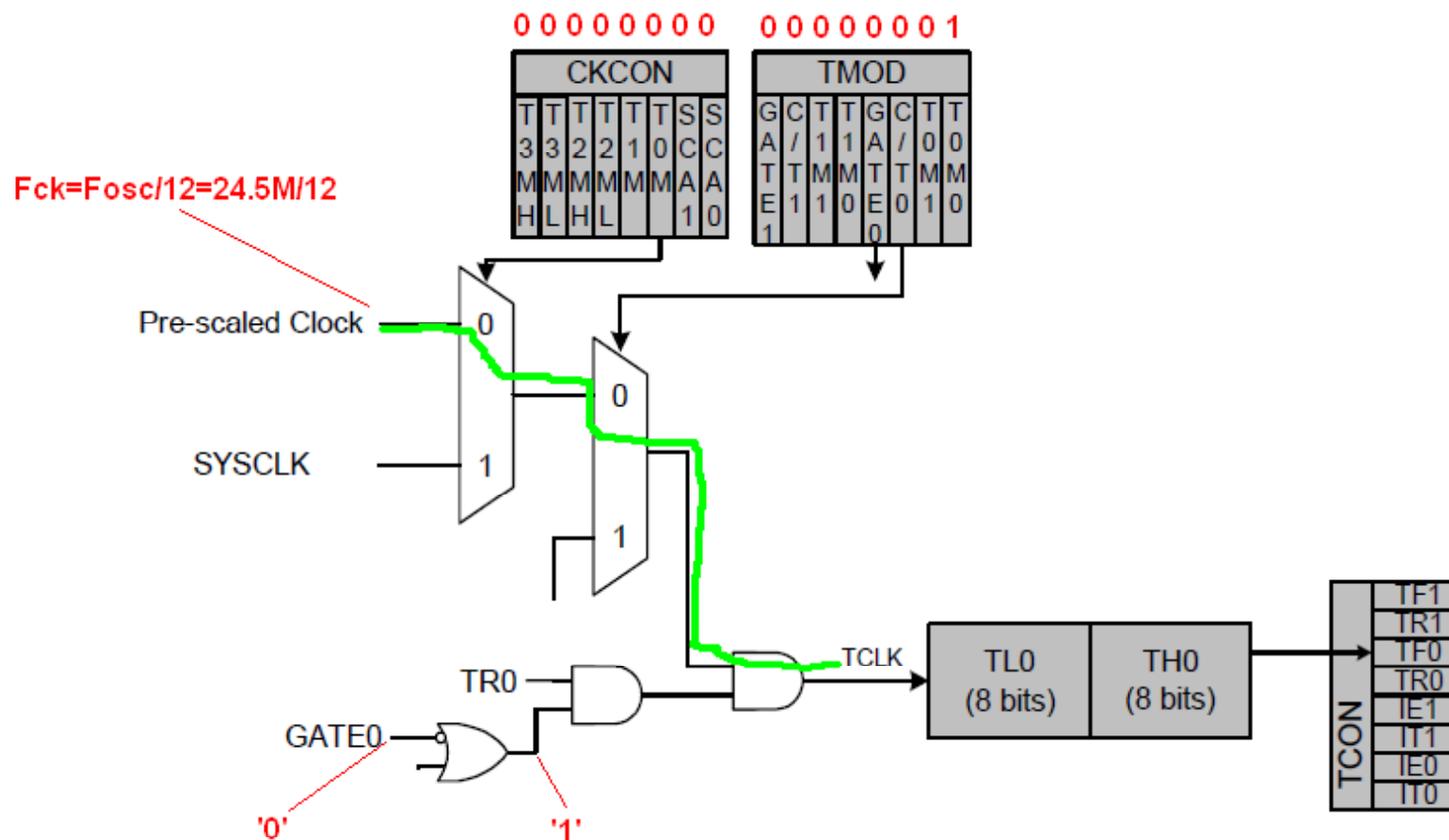
Only Timer0 mode3



◆ קיים בטיימר 0 בלבד, 2 מונים של 8 סיביות המופעלים ע"י TR0,TR1

תוכנית – שימוש בטיימר להשהיה לפי msec

- ◆ נעבוד עם טיימר 0 במוד עבודה של 16bit TMODE=0X01
- ◆ תדר השעון של המונה = תדר השעון מחולק ב-12



שימוש בטיימר להשהיה של msec

♦ פונקצית השהייה ב-msec

טיימר 0 נטען בערך $T0 = -2041$

כלומר 2041 מחזורי שעון של $0.49\mu\text{sec}$ שנותן כמעט 1msec

```
#include "wait_ms.h"

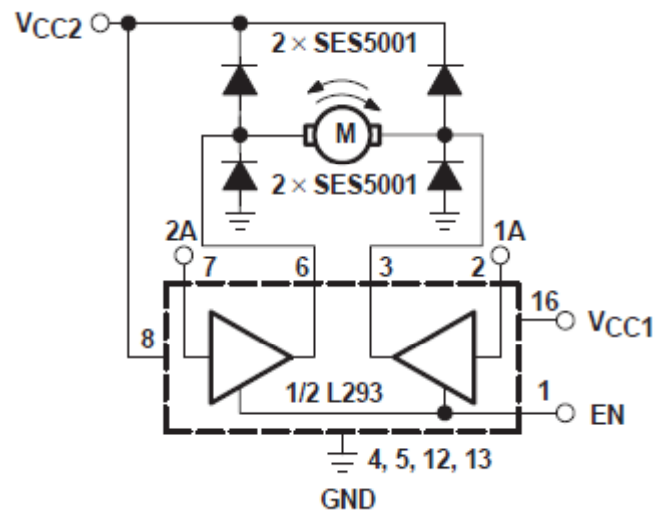
void T0_Wait_ms (unsigned int ms)
{
    TMOD |= 0x01;

    while (ms)
    {
        TR0 = 0; // Stop Timer0
        TL0 = (- (F_OSC/12000)) & 0X00FF;
        TH0 = ((- (F_OSC/12000)) & 0XFF00) >> 8; // Overflow in 1ms

        TF0 = 0; // Clear overflow indicator
        TR0 = 1; // Start Timer0
        while (!TF0); // Wait for overflow
        ms--; // Update ms counter
    }

    TR0 = 0; // Stop Timer0
}
```

דוחף מנוע L293D

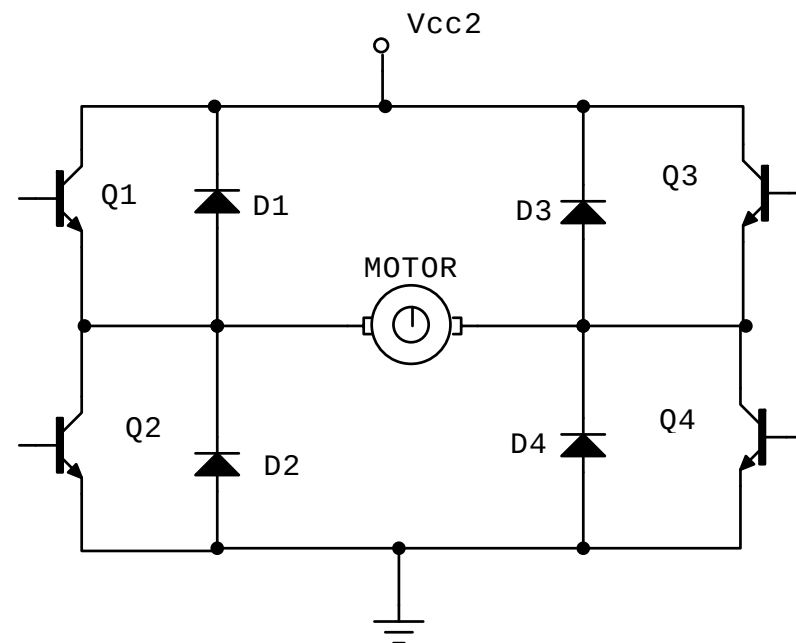


EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

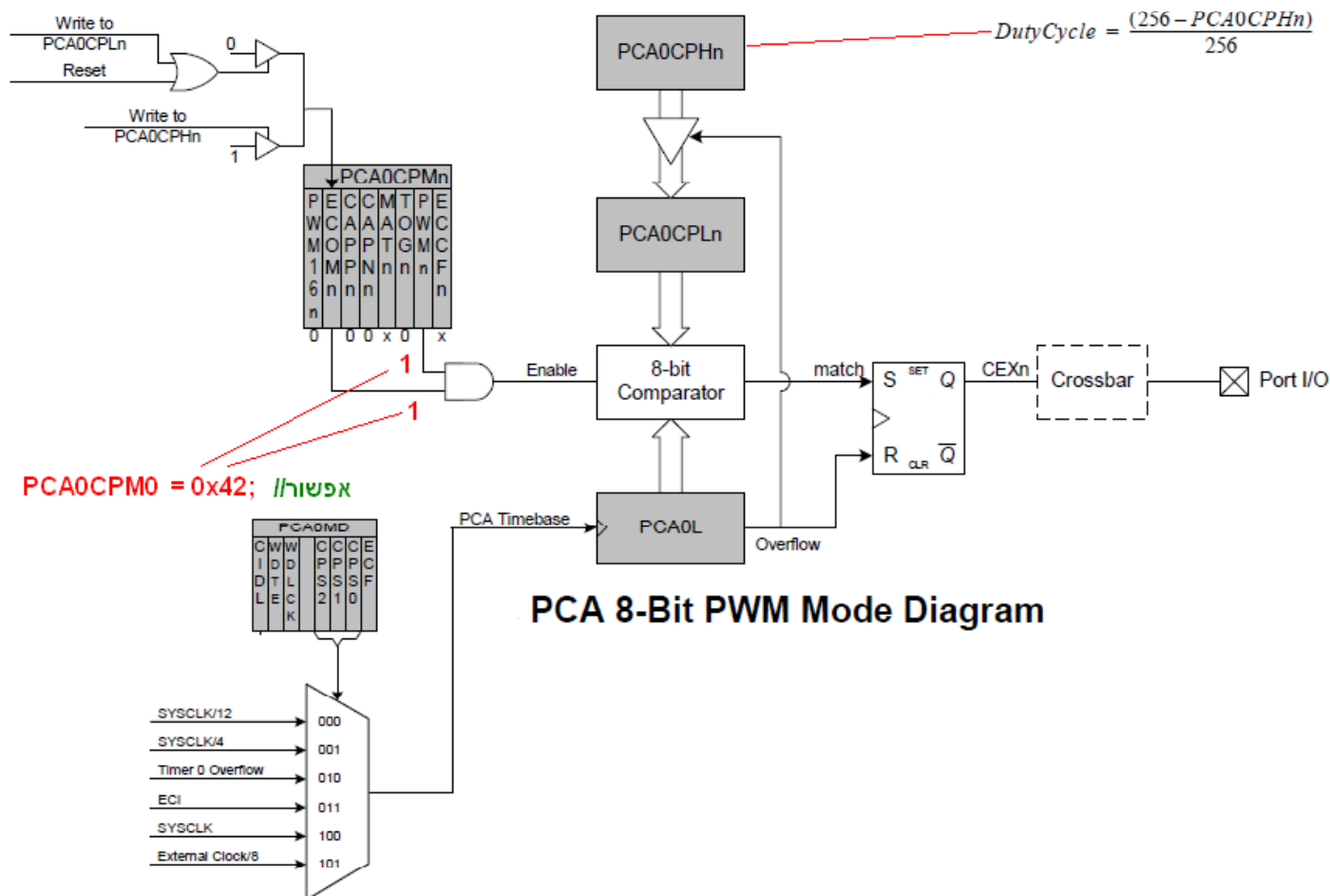
L = low, H = high, X = don't care

♦ מתח V_{CC2} 4.5v-36v

♦ זרם מוצא עד 600mA



PWM 8 Bit

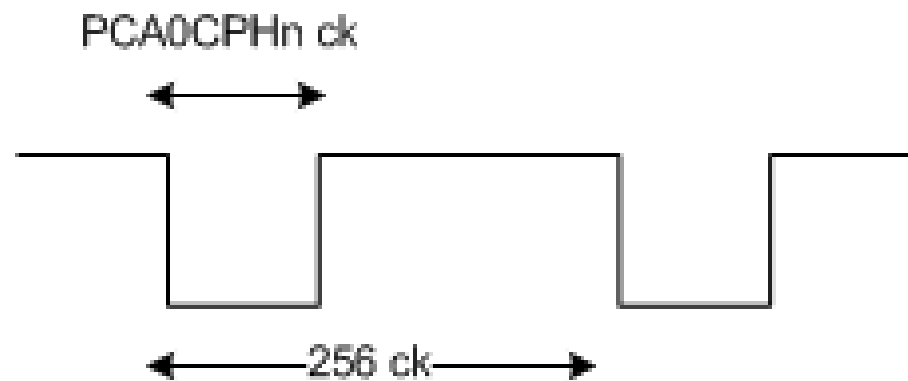


עבור מצב של חלוקה פי 12 של תדר השעון נקבל תדירות PWM

$$F_{pwm} = 24.5M / (12 * 256) = 7975Hz$$

קביעת PWM

$$DutyCycle = \frac{(256 - PCA0CPHn)}{256}$$



פונקציה PWM

דוגמא עבור $\text{pwm}=100$ ערך חיובי

$\text{PCA0CPH0}=100$

```
sbit PHASE_MOTOR=P0^3;
```

```
motor(int pwm)
```

```
{
```

```
    if(pwm>=0)
```

```
    {
```

```
        PCA0CPH0=pwm;
```

```
        PHASE_MOTOR=1;
```

```
    }
```

```
    else
```

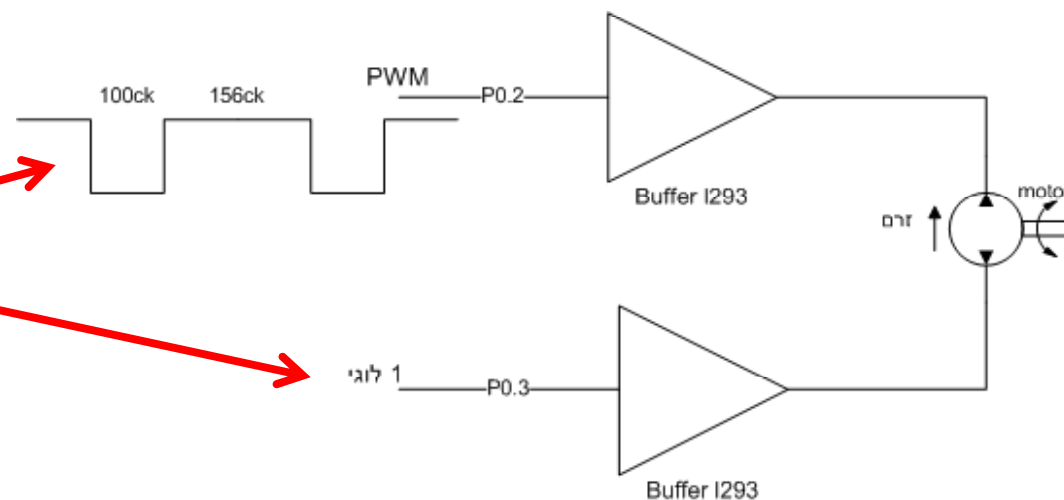
```
    {
```

```
        PCA0CPH0=pwm;
```

```
        PHASE_MOTOR=0;
```

```
    }
```

```
}
```



פונקציה PWM

דוגמא עבור $\text{pwm} = 100$ - ערך שלילי

$$\text{PCA0CPH0} = -100 = 256 - 100 = 156$$

```
sbit PHASE_MOTOR=P0^3;
```

```
motor(int pwm)
```

```
{
```

```
    if(pwm>=0)
```

```
    {
```

```
        PCA0CPH0=pwm;
```

```
        PHASE_MOTOR=1;
```

```
    }
```

```
    else
```

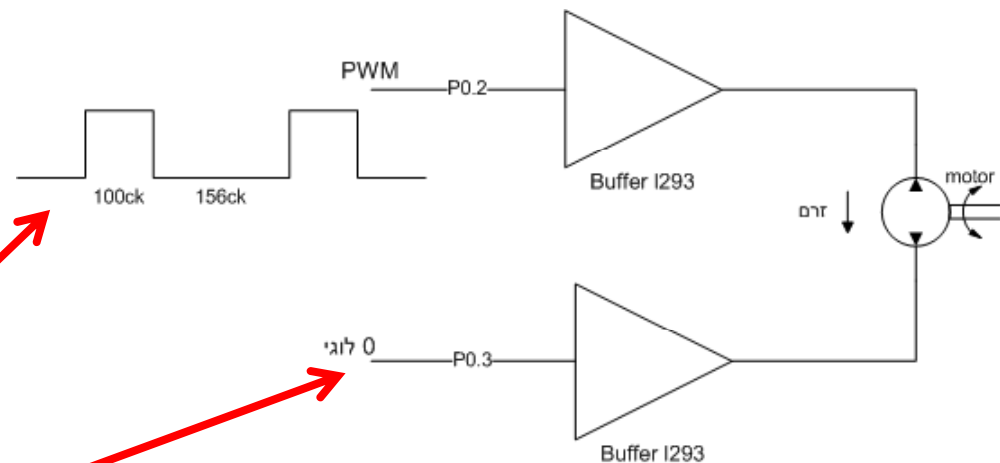
```
    {
```

```
        PCA0CPH0=pwm;
```

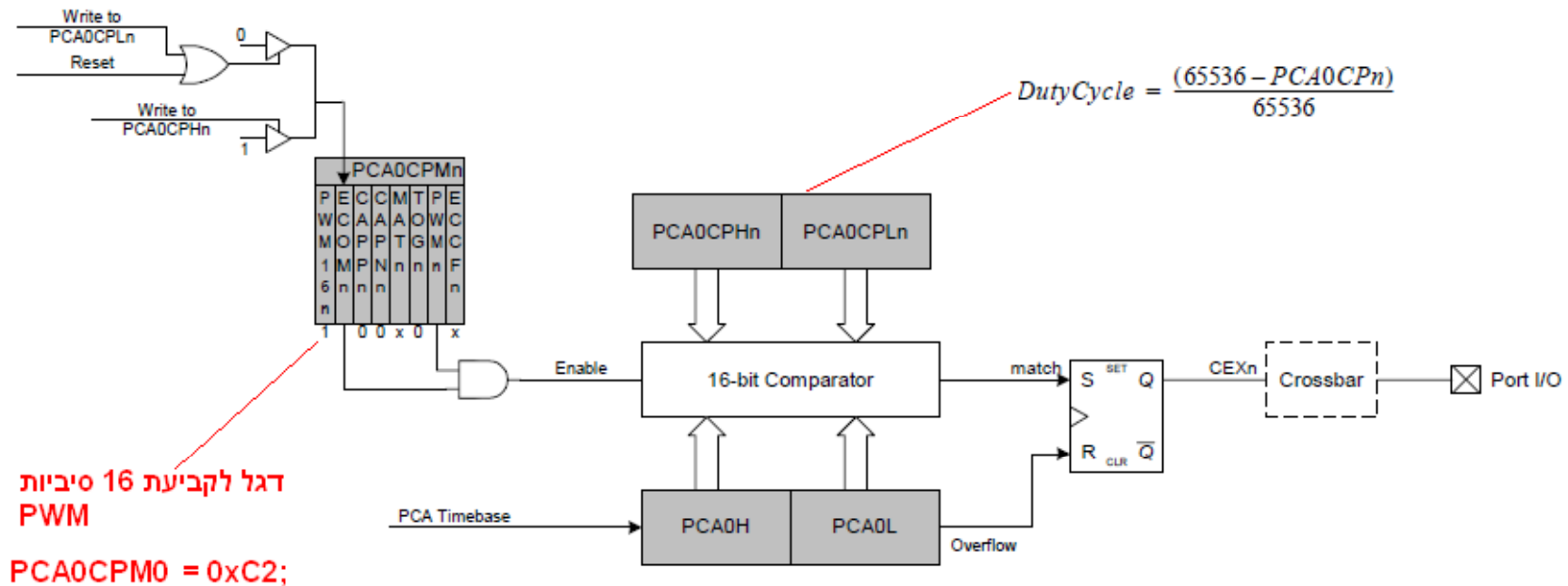
```
        PHASE_MOTOR=0;
```

```
    }
```

```
}
```



PWM 16 Bit

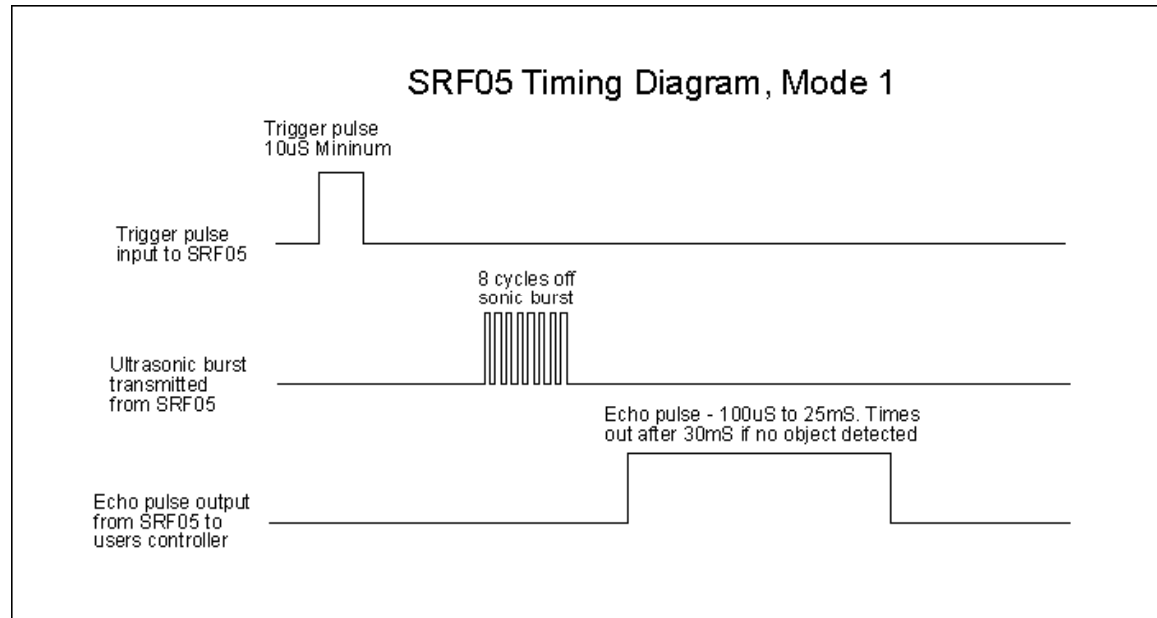
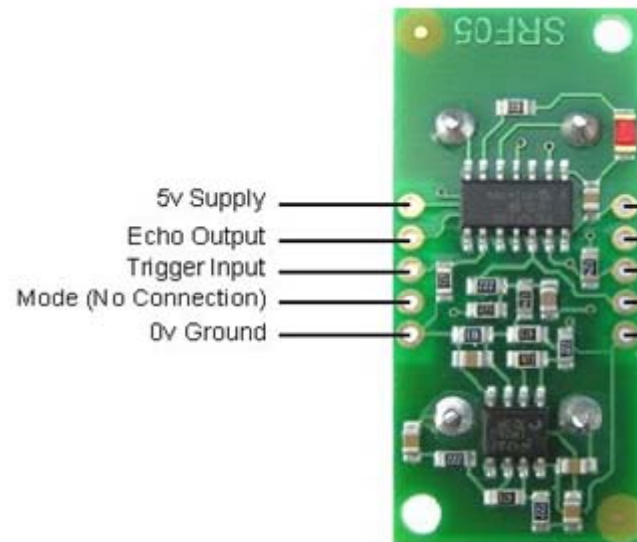


PCA 16-Bit PWM Mode

עבור מצב של תדר השעון ללא חלוקה נקבל תדירות PWM

$$F_{pwm} = 24.5M / (65536) = 373Hz$$

חישן אולטרסוניק SRF05



מהירות הקול באוויר היא: 344m/sec לכן הזמן שלוקח לגל הלוך וחזור
עבור 1cm הוא: $2x/v = 0.02/344 = 58\mu\text{sec}$

פונקצית חישן אולטרסוניק

```
#include "ultrasonic.h"

#define factor_ultra ((FOSC/12000000)*58)

delay1();

int read_distance()
{
    int distance;
    TMOD |= 0x01; // TIMER0 16bit

    TH0=0; // clear counter (8 bit MSB T0)
    TL0=0; // clear counter (8 bit LSB T0)
    trigger=1; // send trigger to ultrasonic
    delay1();
    trigger=0;
    while(!echo); //wait to start pulse from ultra
    TR0=1; // run counter T0
    while(echo); // wait to stop pulse from ultra
    TR0=0; // stop counter

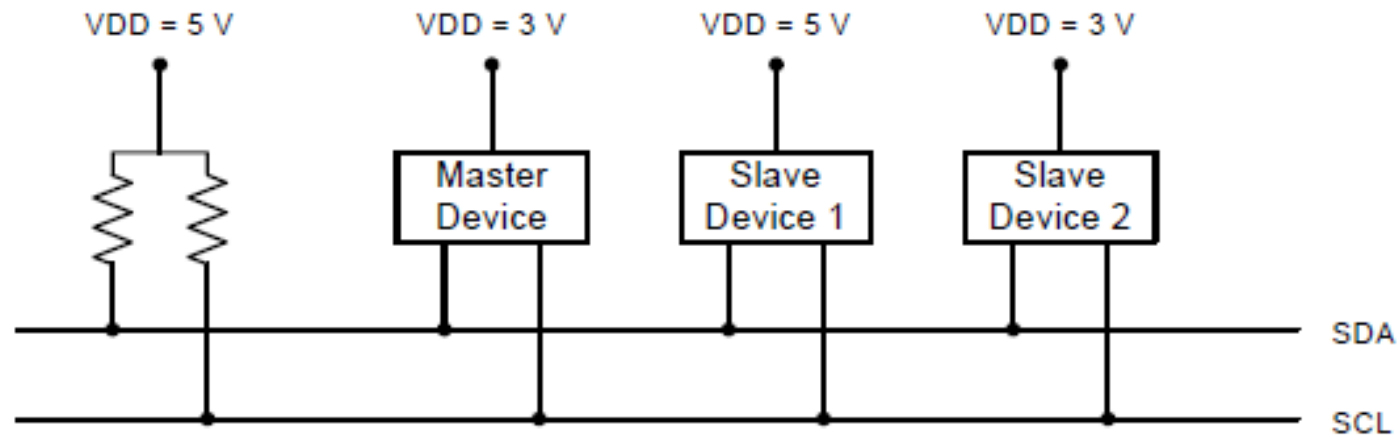
    distance=((TH0*256)+TL0)/factor_ultra;

    return distance;
}

delay1()
{
    int i=1000;
    while(i--);
}
```

<http://site.2all.co.il/hayun/> אבי חיון ויואל כהן

I2C BUS



I2C=Inter Integrated Circuit Bus

SDA - Serial Data line SCK - Serial Clock line קווים :

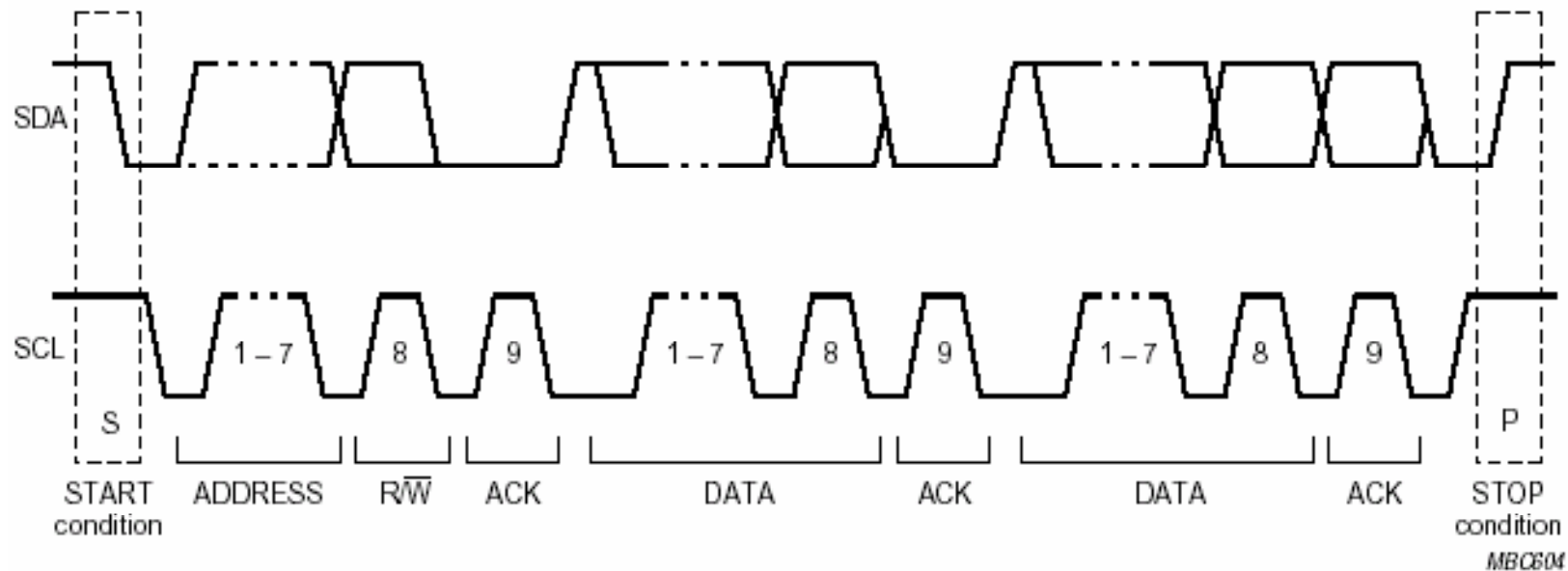
קצבים :

Standard Mode - 100Kbit/sec ➤

Fast Mode - 400Kbit/sec ➤

High-Speed Mode - 3.4Mbit/sec ➤

פרוטוקול I2C



- ♦ התחלת התקשורת – SDA START(S) יורד ולאחר מכן SCL
- ♦ MASTER שולח כתובת הרכיב בגודל 7 סיביות + ביט R/W
- ♦ ה-MASTER מקבל אישור ACK=0 מה-SLAVE
- ♦ ה-MASTER קורא או כותב ל-SLAVE לפי ביט שנשלח R/W
- ♦ סיום התקשורת – SCL STOP (P) עולה ולאחר מכן SDA
- ♦ במידה ויש המשך תקשורת ה-MASTER שולח Repeated START (Sr)

דוגמאות לרכיבי I2C

SYMBOL	PIN	DESCRIPTION
AIN0	1	analog inputs (A/D converter)
AIN1	2	
AIN2	3	
AIN3	4	
A0	5	hardware address
A1	6	
A2	7	
V _{SS}	8	negative supply voltage
SDA	9	I ² C-bus data input/output
SCL	10	I ² C-bus clock input
OSC	11	oscillator input/output
EXT	12	external/internal switch for oscillator input
AGND	13	analog ground
V _{REF}	14	voltage reference input
AOUT	15	analog output (D/A converter)
V _{DD}	16	positive supply voltage

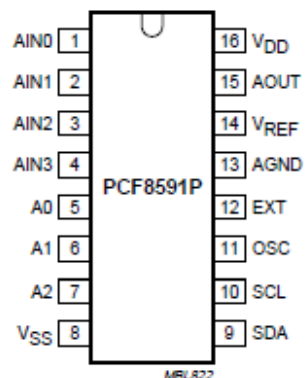
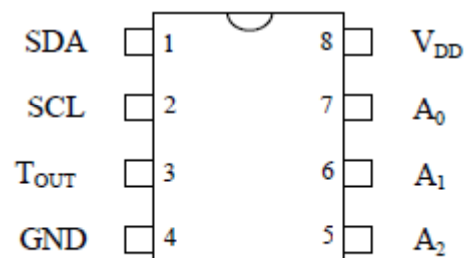


Fig.2 Pinning diagram (DIP16).

PCF8591 ♦

D/A + 4 כניסות A/D

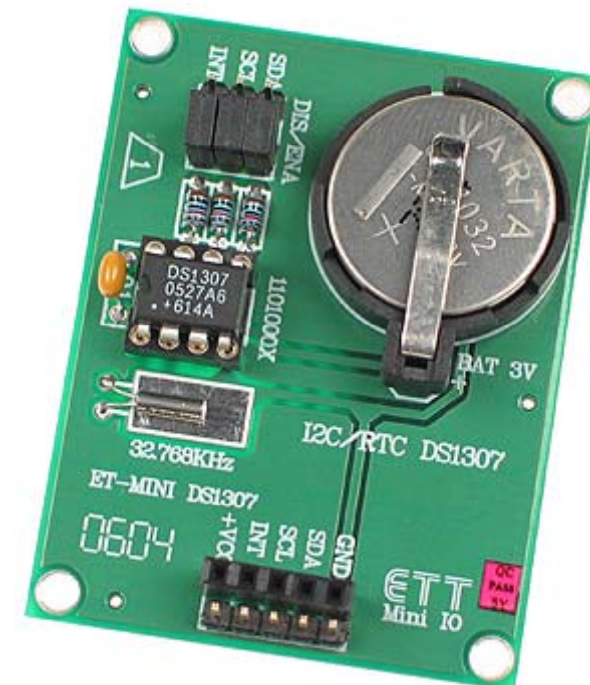
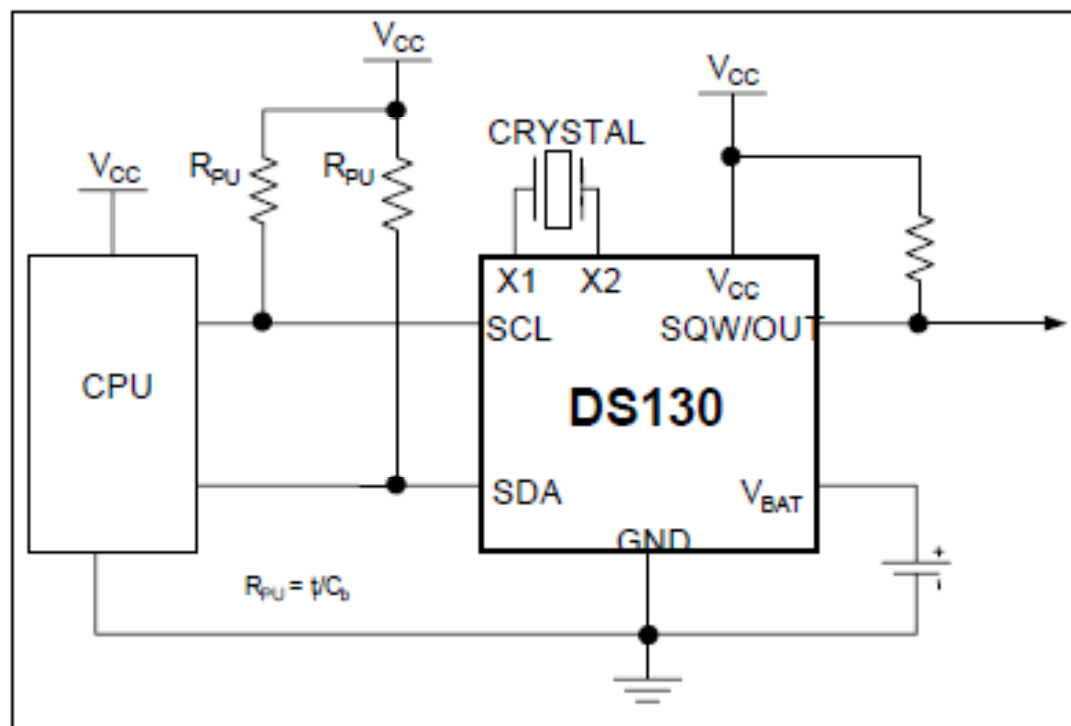


DS1621 8-PIN DIP (300mil)

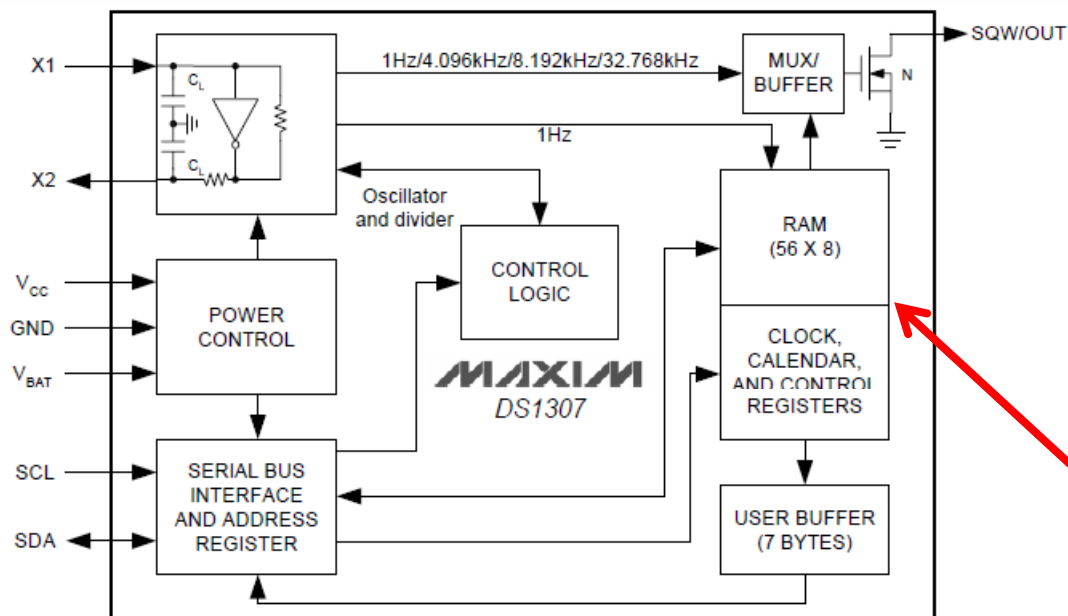
DS1621 ♦

רכיב טמפ' וטרמוסטט

DS1307 רכיב שעון RTC



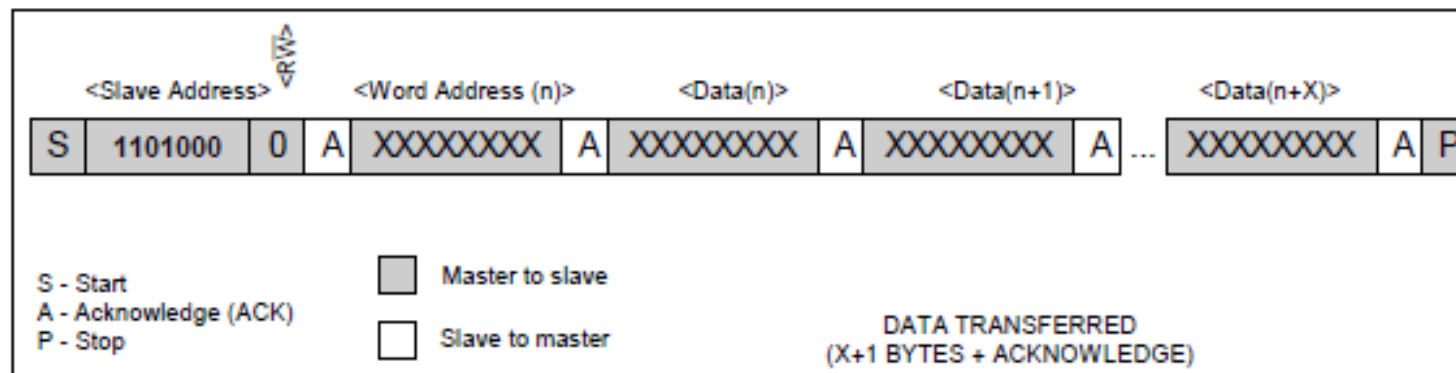
מבנה הרכיב



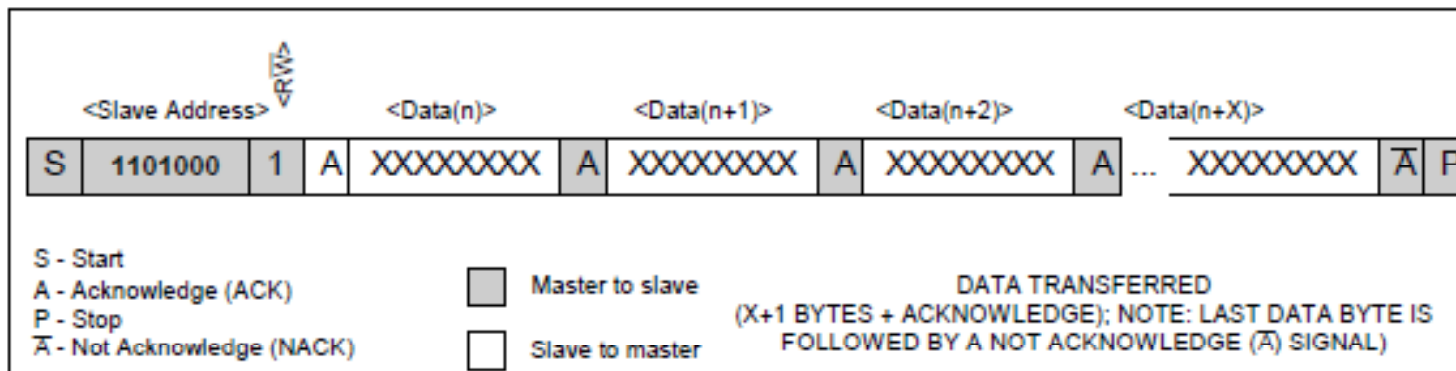
ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date			Date			Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

קריאה וכתובה מהרכיב DS1307

◆ Data Write—Slave Receiver Mode

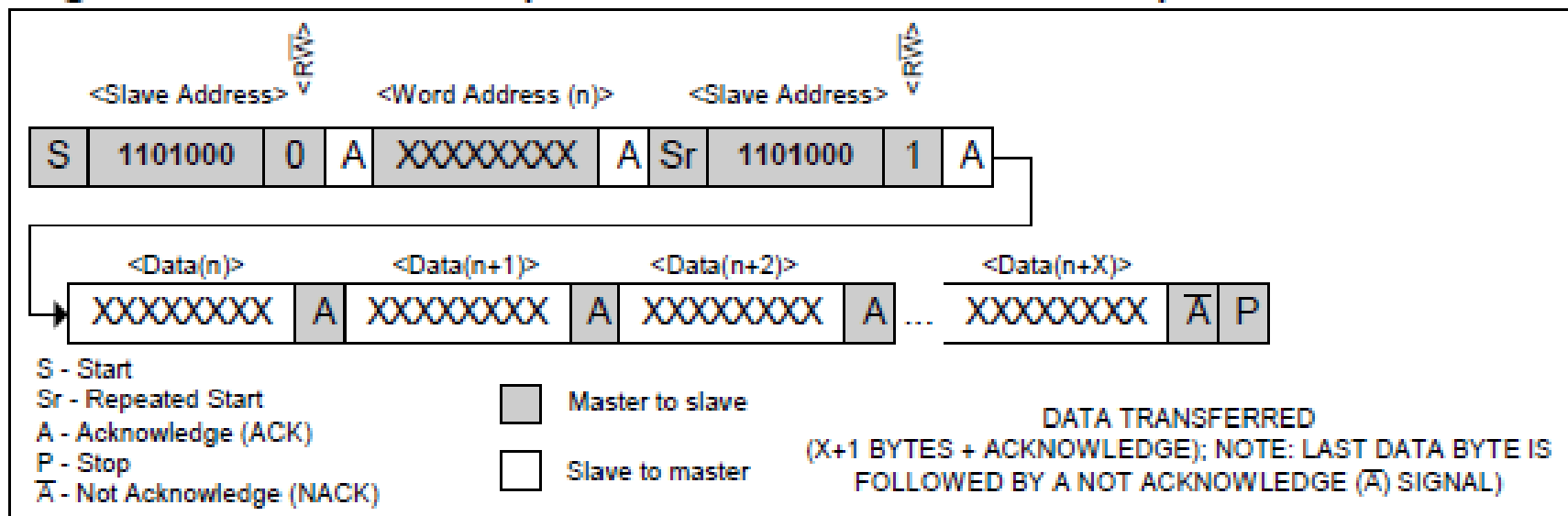


◆ Data Read—Slave Transmitter Mode



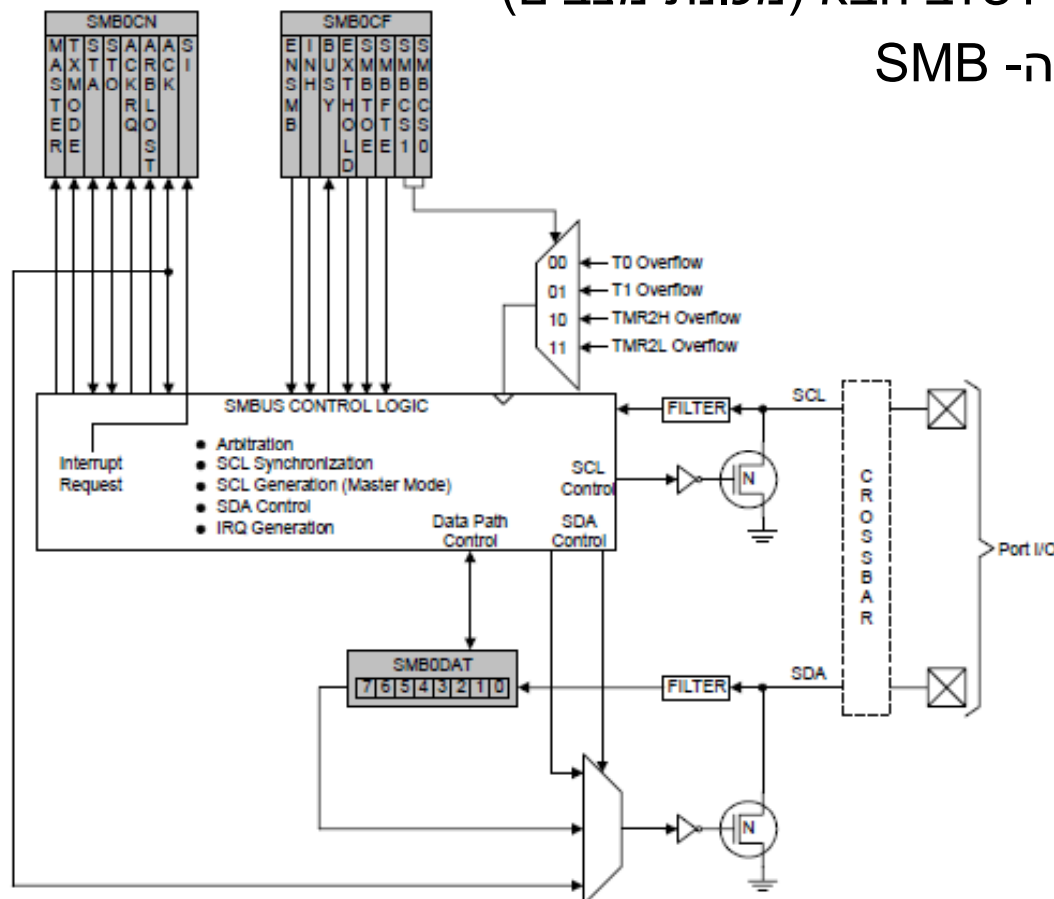
קריאה - SLAVE קולט ומשדר

◆ Data Read (Write Pointer, Then Read)—Slave Receive and Transmit



שיטת התקשורת של המעבד C8051F310

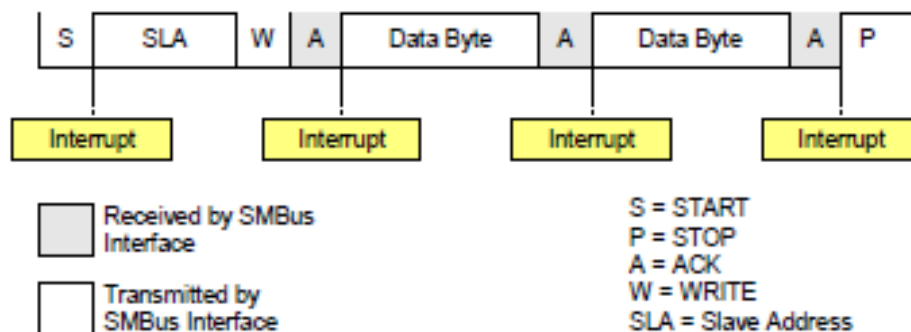
- ◆ טיימר 2 משמש כמבוא שעון של התקשורת (אפשרי גם טיימר 0 ו-1)
- ◆ טיימר 3 משמש לזיהוי מצב שבו SCL נמצא זמן רב במצב 0 לוגי (תקוע)
- ◆ התקשורת משתמשת בפסיקה (interrupt 7) המתקבלת אחרי כל שלב ולפי מלת סטאטוס (SMBOCN) מתקדמים לשלב הבא (מכונת מצבים)
- ◆ רגיסטר SMBOCF קובע פעולת ה-SMB



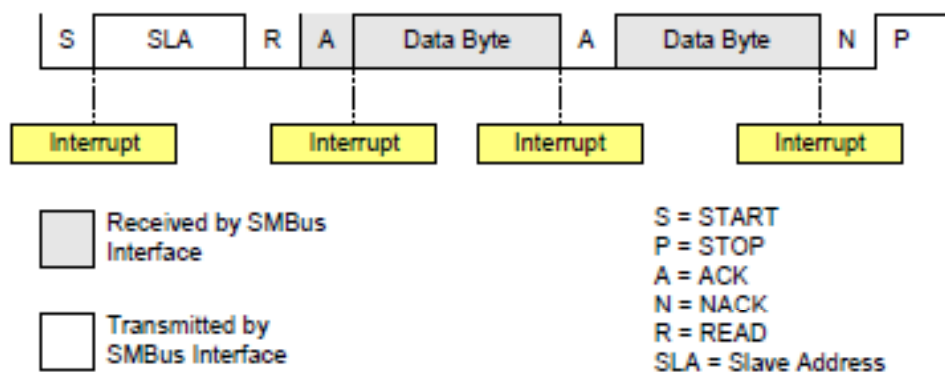
<http://site.2all.co.il/hayun/> אבי חיון ויואל כהן

פרוטוקול שליחה וקבלה של המעבד C8051F310 (MASTER)

♦ MASTER שולח נתונים ל-SLAVE



♦ MASTER מקבל נתונים מה-SLAVE



פונקציות לקריאה וכתובה DS1307

פונקצית שליחת מידע לשעון DS1307

void SMB_ByteWrite(unsigned char addr, unsigned char dat)

♦ addr - כתובת ה-RAM הפנימי

♦ dat - מידע הנשלח ל-RAM (תאריך, שעה, בקרה, נתונים כללים)

פונקצית קריאת מידע מהשעון DS1307

unsigned char SMB_ByteRead(unsigned char addr)

♦ addr - כתובת ה-RAM הפנימי

♦ הפונקציה מחזירה נתון מה-RAM כמו שעה ותאריך