

# Мікропроцесорна техніка

(лекція 8)  
Благітко Б.Я.  
2019 р.

**PSoC Creator 4.2**  
**Designing with PSoC 3/5**



# PSoC@3 /5 Stepper Motors

PSoC Creator 4.2  
Designing with PSoC 3/5



# PSoC@3 /5

Управління  
кутом повороту ротора  
крокових двигунів  
постійного струму

PSoC Creator 4.2  
Designing with PSoC 3/5



A DC motor uses a commutator to supply continuous direct current to a coil placed between two or more magnets.

A stepper motor receives pulses of electricity from a pulse generator and sends them through a similar coil-magnet setup.

In both cases, the Oersted effect causes the coils to turn the motor's shaft.

Read more: [DC Motor vs. Stepper Motor | eHow.com](http://www.ehow.com/facts_6777501_dc-motor-vs_-stepper-motor.html#ixzz2QKmmwrgy) [http://www.ehow.com/facts\\_6777501\\_dc-motor-vs\\_-stepper-motor.html#ixzz2QKmmwrgy](http://www.ehow.com/facts_6777501_dc-motor-vs_-stepper-motor.html#ixzz2QKmmwrgy)

## Stepper Motor

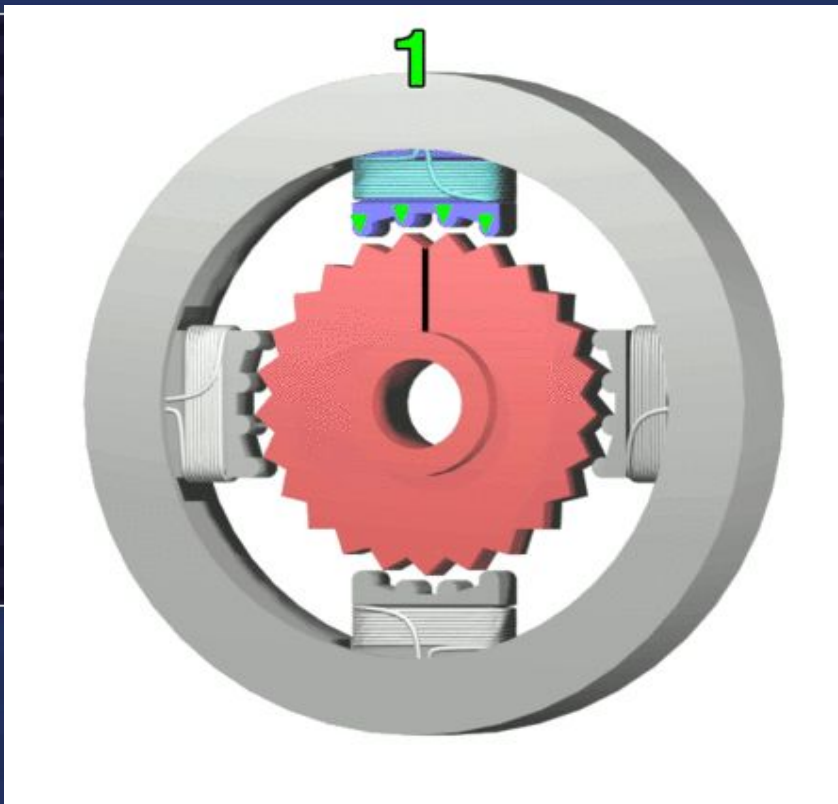




Figure 7-3. Component Catalog

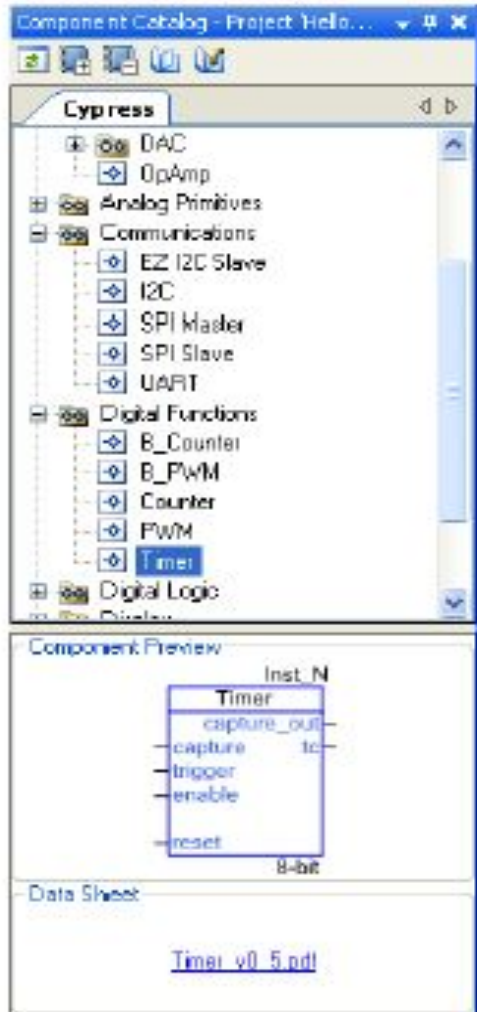
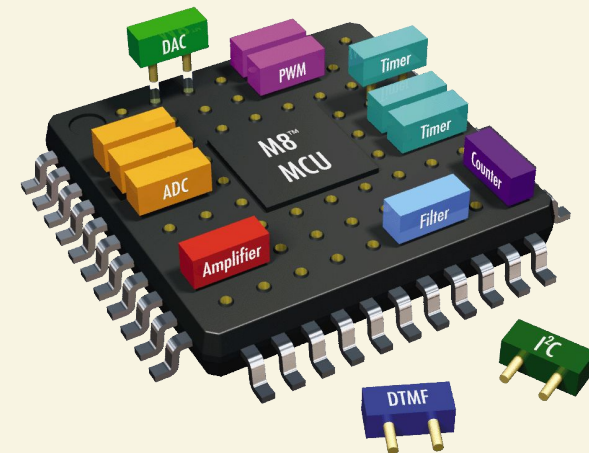
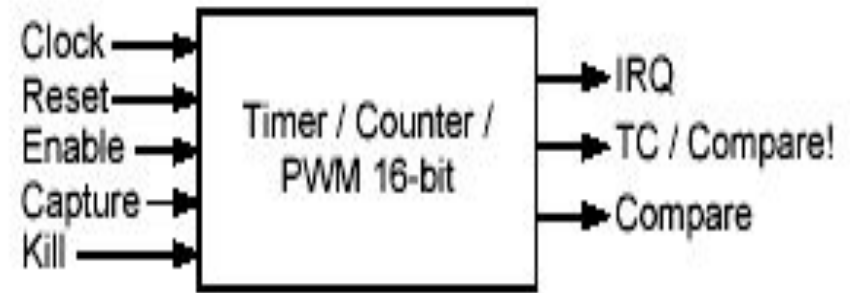


Figure 7-21. Timer/Counter/PWM





**PWMs** and **Counters** share many capabilities but each provides specific capabilities.

## **When to Use a PWM**

The most common use of the **PWM** is to generate periodic waveforms with adjustable duty cycles. The PWM also provides optimized features for power control, motor control, switching regulators and lighting control. The PWM can also be used as a clock divider by driving a clock into the clock input and using the terminal count or a PWM output as the divided clock output.



Output	May Be Hidden	Description
tc	N	The terminal count output is '1' when the period counter is equal to zero. In normal operation this output will be '1' for a single cycle where the counter is reloaded with period. If the PWM is stopped with the period counter equal to zero then this signal will remain high until the period counter is no longer zero. This output is synchronized to the block clock input of the component.
interrupt	Y	The interrupt output is the logical OR of the group of possible interrupt sources. This signal will go high while any of the enabled interrupt sources are true. The interrupt output shall remain asserted until the Status Register is read out by the software. In order to receive subsequent interrupts, the interrupt shall be cleared by reading the Status Register using the PWM_ReadStatusRegister() API. The interrupt output is not visible if the Use Interrupt parameter is not set. This allows the status register to be removed for resource optimization as necessary.
pwm/pwm1	Y	The pwm or pwm1 output is the first or only pulse width modulated output. This signal is defined by PWM Mode, compare modes(s), and compare value(s) as indicated in waveforms in the Configure dialog. When the instance is configured in one output, Dual Edged, Hardware Select, Center Aligned, or Dither PWM Modes, then the output "pwm" is visible. Otherwise the output "pwm1" is visible with "pwm2" the other pulse width signal. This output is synchronized to the block clock input of the component.

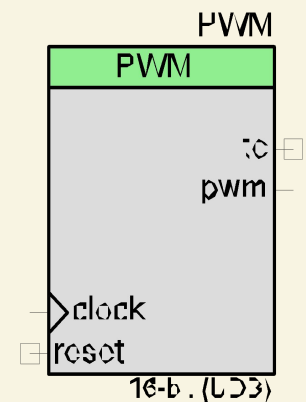
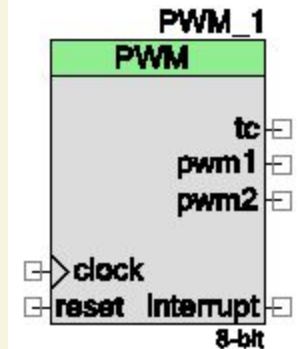
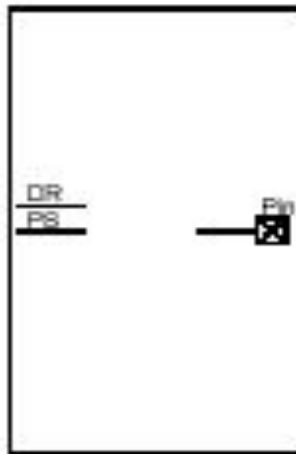
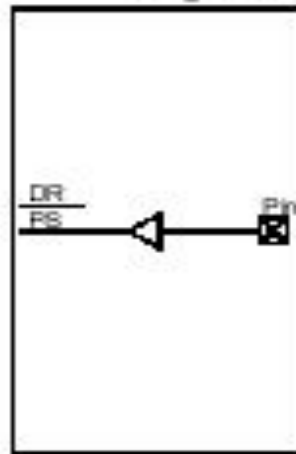


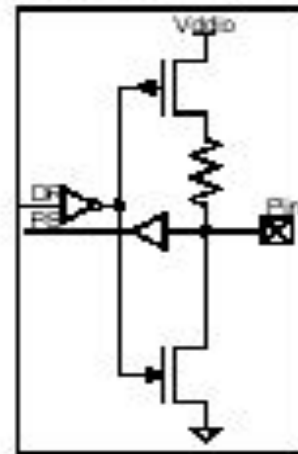
Figure 6-11. Drive Mode



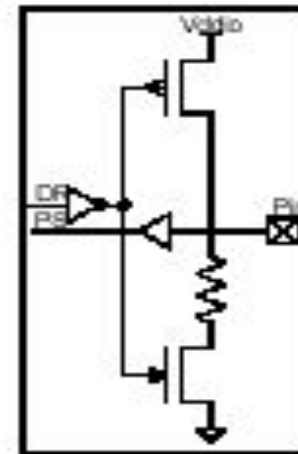
0. High Impedance Analog



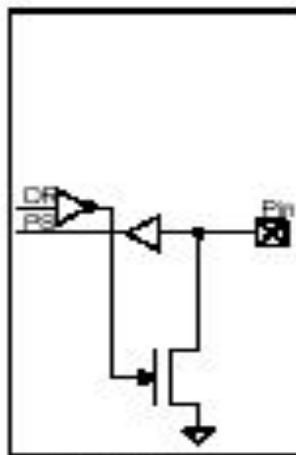
1. High Impedance Digital



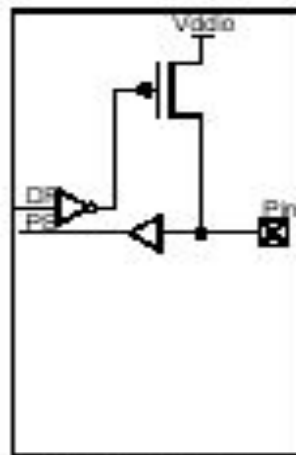
2. Resistive Pull-Up



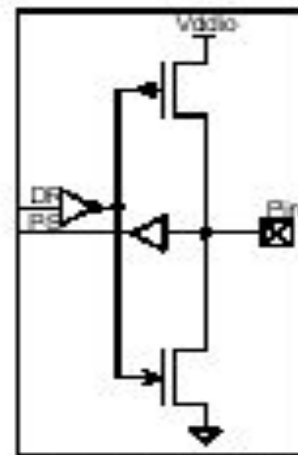
3. Resistive Pull-Down



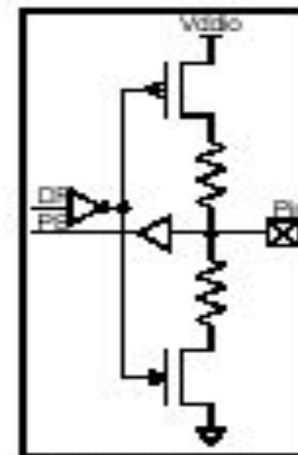
4. Open Drain Drives Low



5. Open Drain Drives High



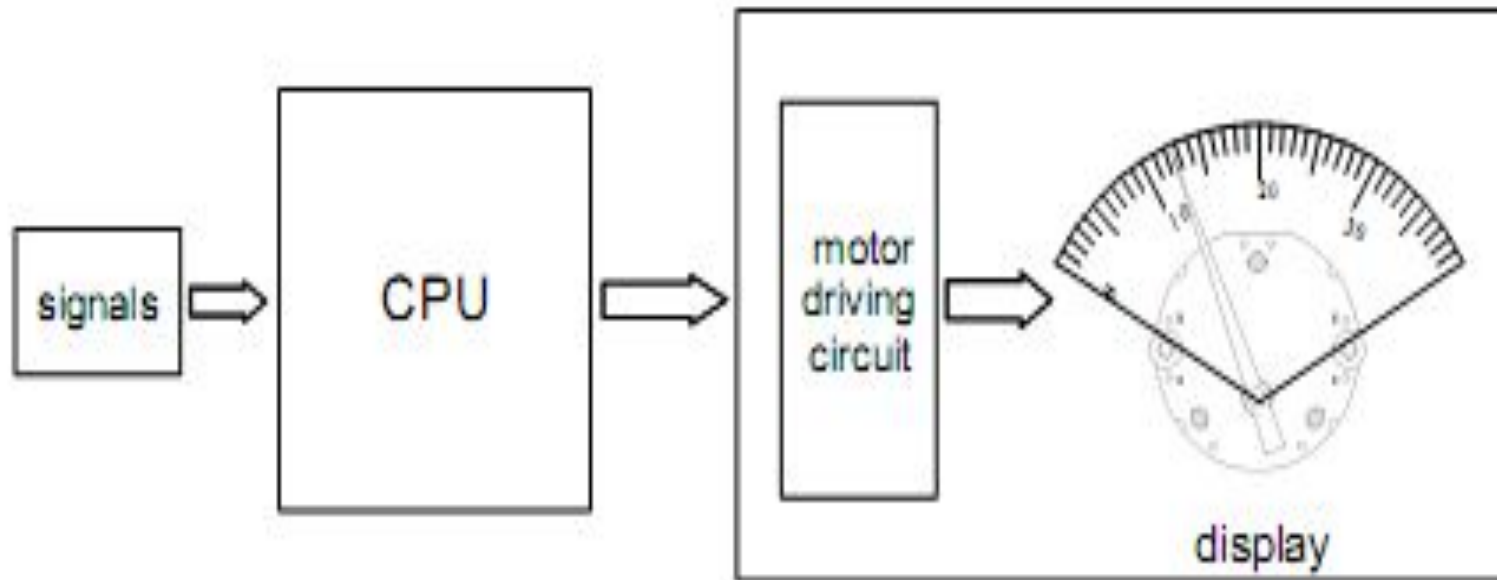
6. Strong Drive



7. Resistive Pull-Up and Pull-Down

## Stepper Motor

### TYPICAL APPLICATION



perfect combination of digital accuracy and analog facility



VID29 Series Cluster Stepper Motor



## MAIN FEATURES

- High speed rotation: 600°/sec.
- High  $\mu$ -step resolution: 1/12°.
- Wide working voltage: 3.5~10V.
- Wide working temperature: -40 ~ 105°C.
- Low current consumption.
- Extremely robust construction:  $\Phi$ 30mm X 7.6mm.

## FRAEN AMD's 6405-15xx stepper motors

6405-500 Stepper Motor

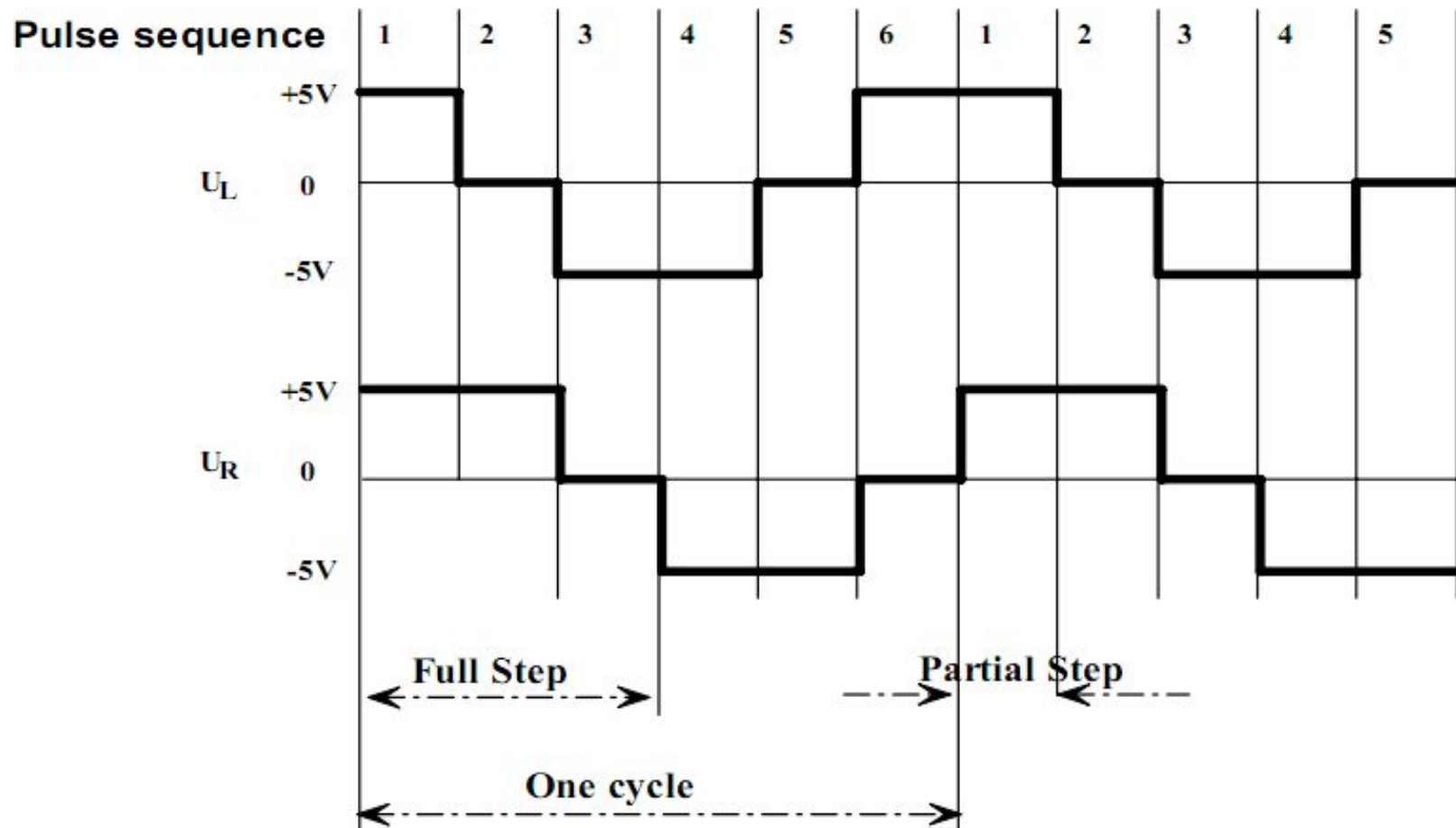


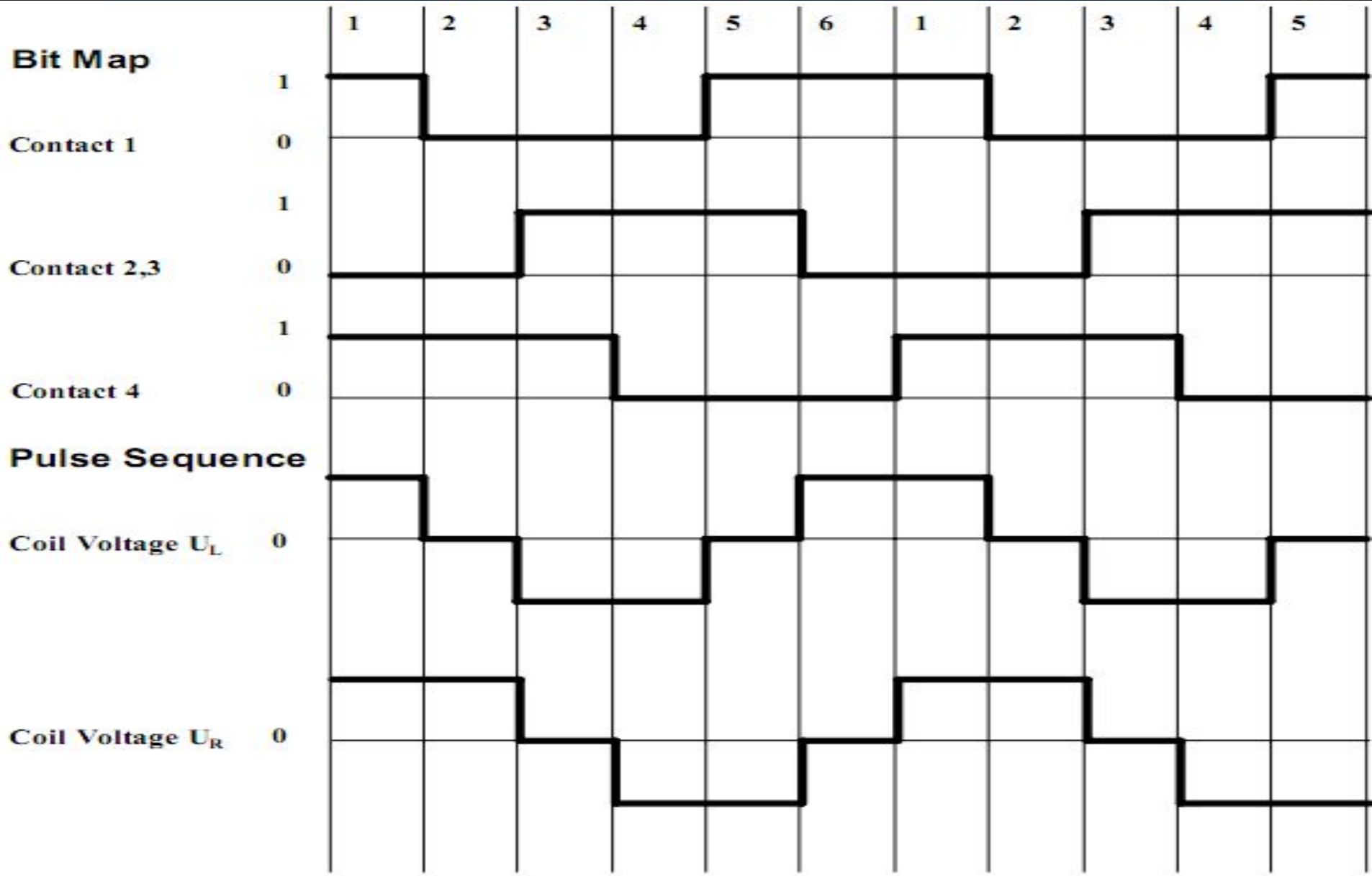
## Stepper Motors



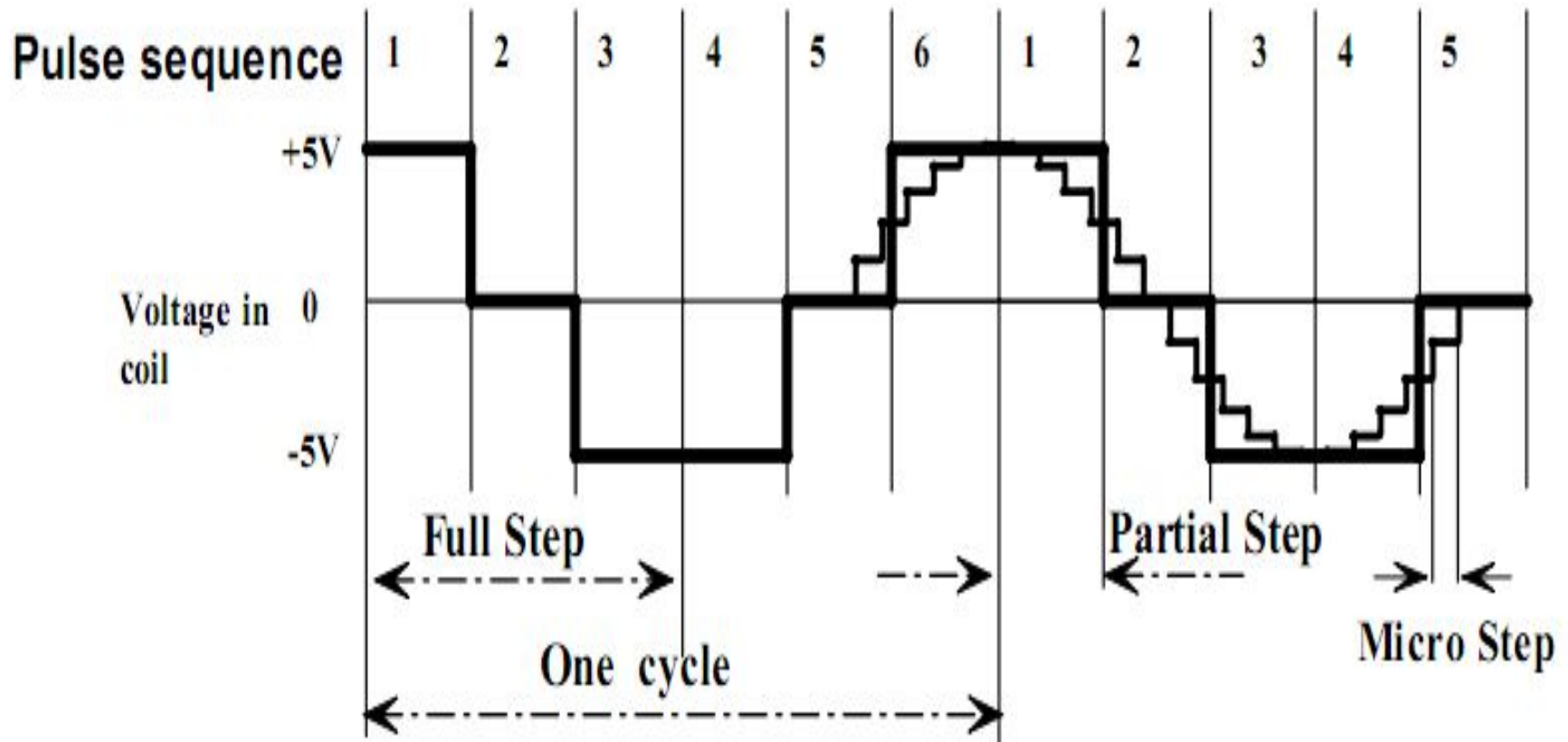
## STEP DEFINITION AND ROTOR MOVEMENT

VID29-XX and VID29-XXP series motor are driven by 2 sequent logic pulse singnals. Its work diagram is as following:

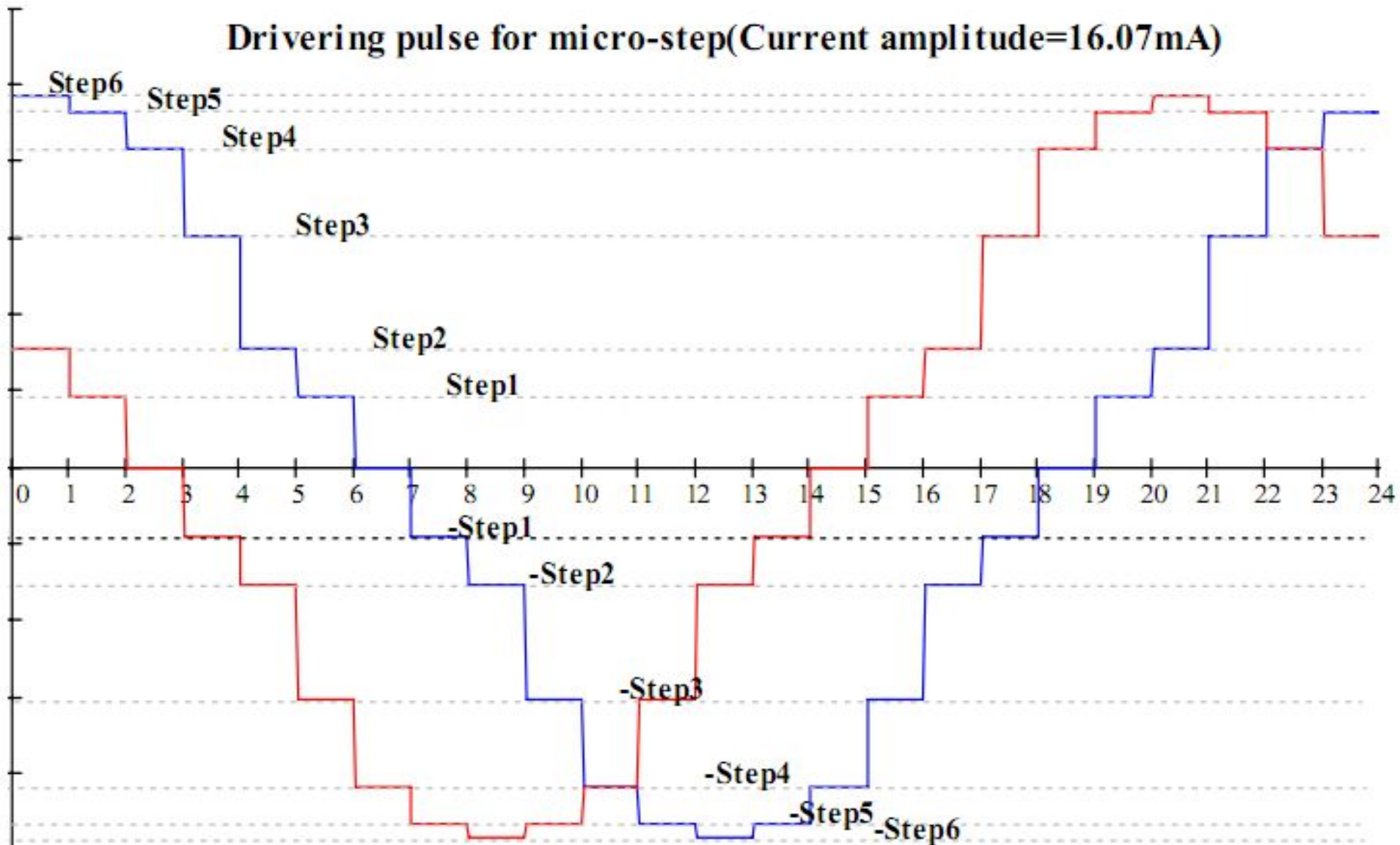








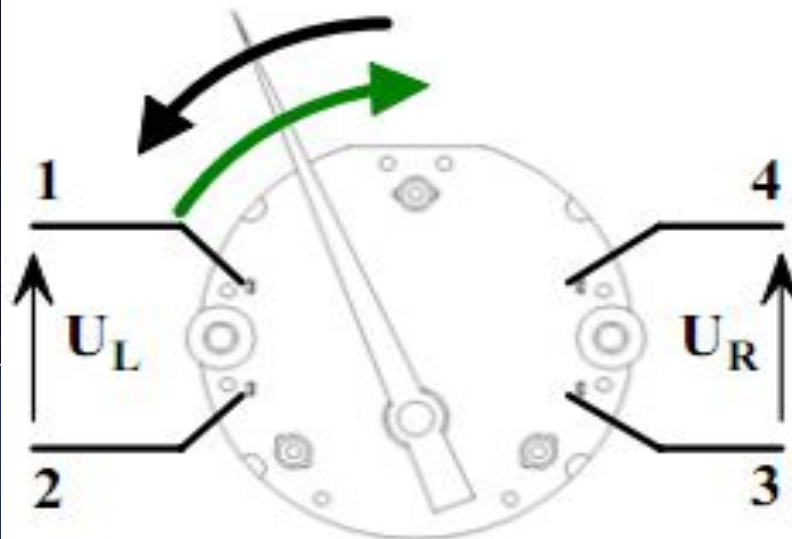
Driving pulse for micro-step (Current amplitude=16.07mA)



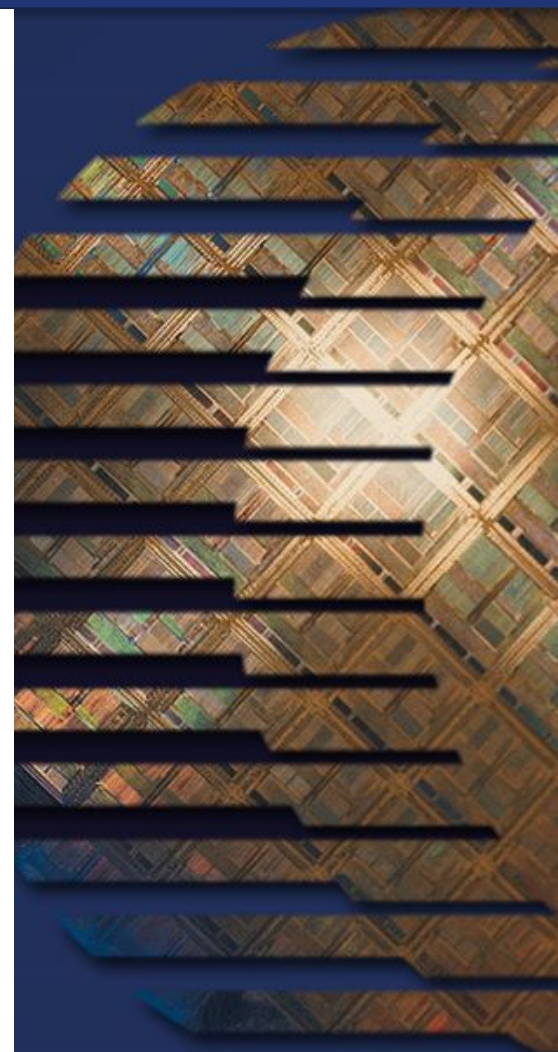
Clockwise (for VID29\_xxP)  
Counterclockwise (for VID29\_xx)



Counterclockwise (for VID29\_xxP)  
Clockwise (for VID29\_xx)



VID29\_xxP





# Lab\_8 Stepper Motor

New Project

Design Other

### Empty Templates

- Empty PSoC 3 Design
- Empty PSoC 4 (CY8C42\*) Design
- Empty PSoC 4 (CY8C41\*) Design
- Empty PSoC 5 Design
- Empty PSoC 5LP Design

### PSoC 3 Starter Designs

- ADC\_DMA\_VDAC
- DelSig\_16Channel
- DelSig\_I2CM
- DelSig\_I2CS
- DelSig\_SPIM
- Filter\_ADC\_VDAC
- HW\_Fan\_Control\_with\_Alert

### PSoC 4 (CY8C42\*) Starter Designs

- 
- 
- 

Creates a PSoC 3, 8 bit, design project.

Name:

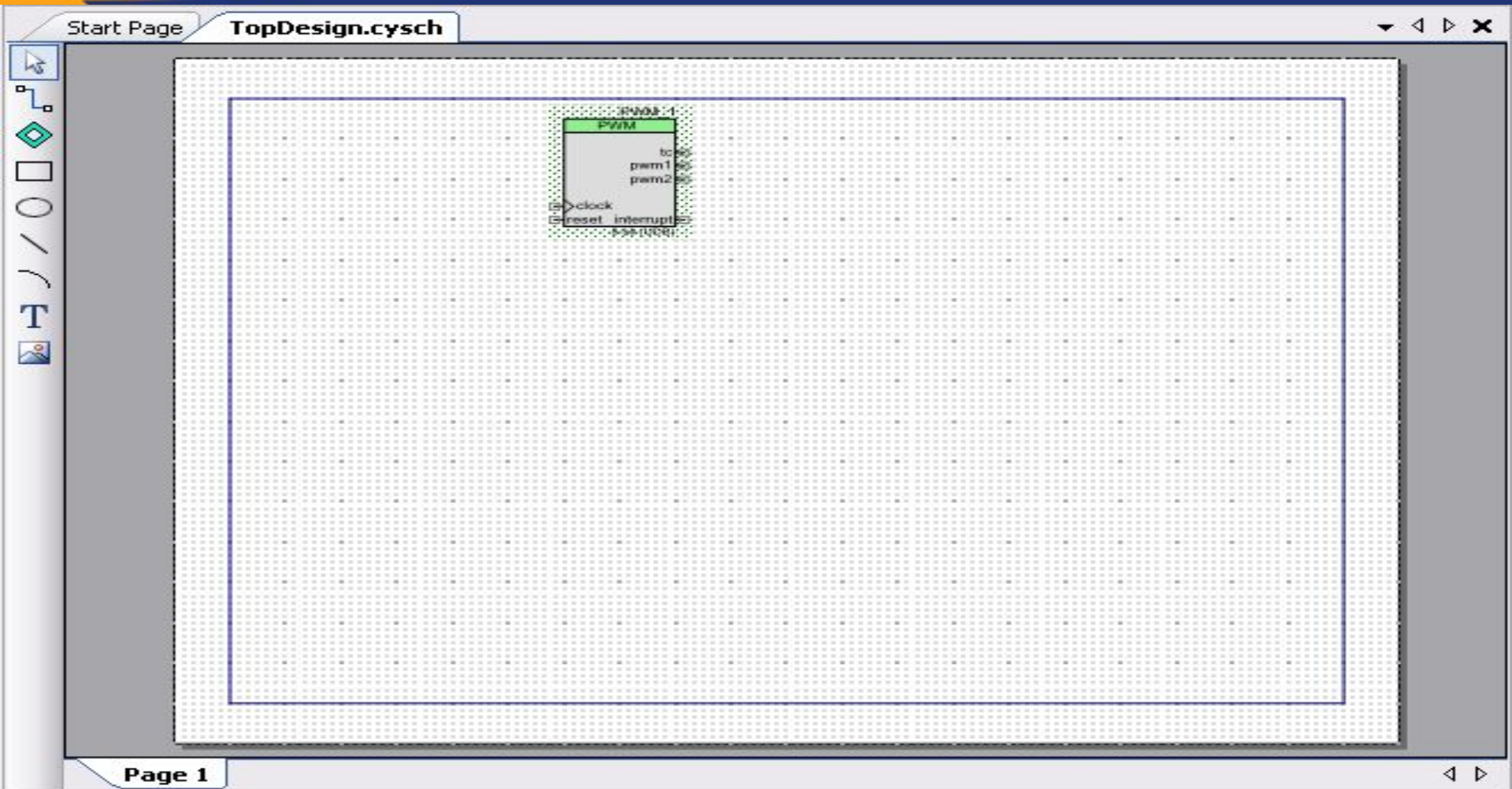
Location:  ...

+ Advanced

OK Cancel



# Lab\_8 Stepper Motor



Page 1

Notice List

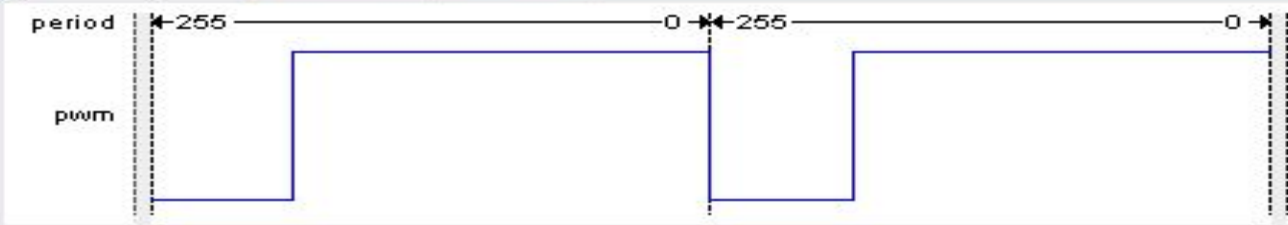
2 Errors 0 Warnings 0 Notes Go To Error View Details

	Description	File	Error Location
+	2 Errors		
+	sdb.M0025:No input on Instance "PWM_1", terminal "clock".	TopDesign.cysch	Signal: Net_2,...
+	sdb.M0050:Terminal "clock" requires connection when it is visible.	TopDesign.cysch	Signal: Net_2,...

### Configure 'PWM'

Name:

Configure   
  Advanced   
  Built-in



Implementation:   
  Fixed Function   
  UDB

Resolution:   
  8-Bit   
  16-Bit

PWM Mode:

Period:   *Period = UNKNOWN SOURCE FREQ*

CMP Value 1:

Description	File	Error Location
<input checked="" type="checkbox"/> sdb.M0025: No input on Instance "PWM_1", terminal "clock".	TopDesign.cysch	Signal: Net_2,...
<input checked="" type="checkbox"/> sdb.M0050: Terminal "clock" requires connection when it is visible.	TopDesign.cysch	Signal: Net_2,...

Start Page | **TopDesign.cysch**

### Configure 'PWM'

Name:

Configure | **Advanced** | Built-in

Enable Mode:

Run Mode:

Trigger Mode:

Kill Mode:

Capture Mode:

Interrupts:

- None
- Interrupt On Terminal Count Event
- Interrupt On Compare 1 Event
- Interrupt On Compare 2 Event
- Interrupt On Kill Event

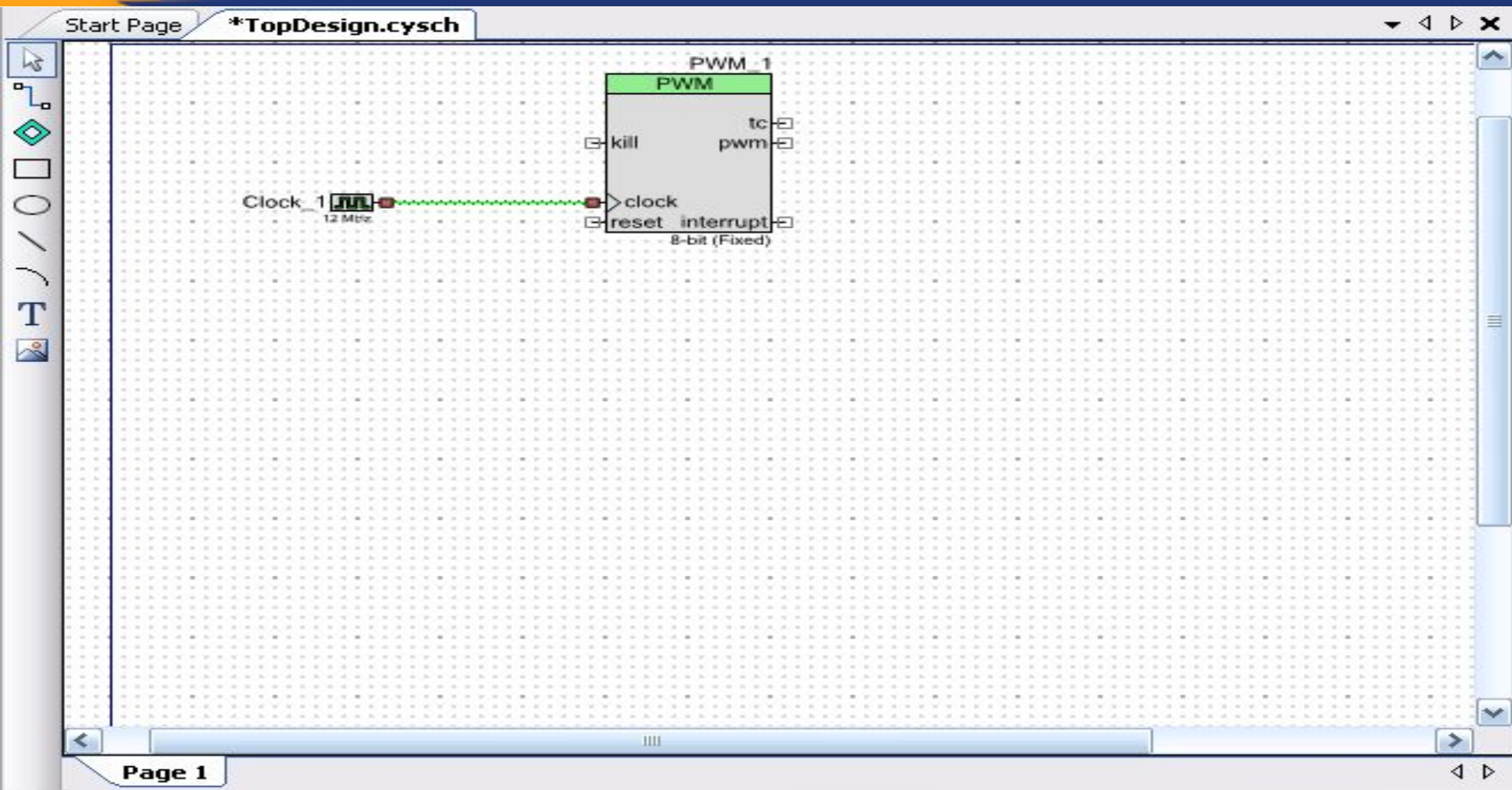
Datasheet | OK | Apply | Cancel

Page 1




Notice List

2 Errors | 0 Warnings | 0 Notes | Go To Error | View Details

Description	File	Error Location
sdb.M0025:No input on Instance "PWM_1", terminal "clock".	TopDesign.cysch	Signal: Net_2,...
sdb.M0050:Terminal "clock" requires connection when it is visible.	TopDesign.cysch	Signal: Net_2,...



Notice List

 1 Errors
  0 Warnings
  0 Notes
 [Go To Error](#) | [View Details](#)

	Description	File	Error Location
	sdb.M0050:Terminal "kill" requires connection when it is visible.	TopDesign.cysch	Signal: Net_22,...



Start Page

\*TopDesign.cysch

## Configure 'cy\_clock'

Name:

**Basic**    Advanced    Built-in

Clock type:  New     Existing

Source:  ▼

Specify:  Frequency:  kHz ▼

Tolerance:  +

### Summary

**API Generated:** Yes  
**Uses Clock Tree Resource:** Yes

By default, all clocks are marked as 'start on reset'. The setting can be changed in the Design Wide Resources editor.

Datasheet




OK


Apply

Cancel

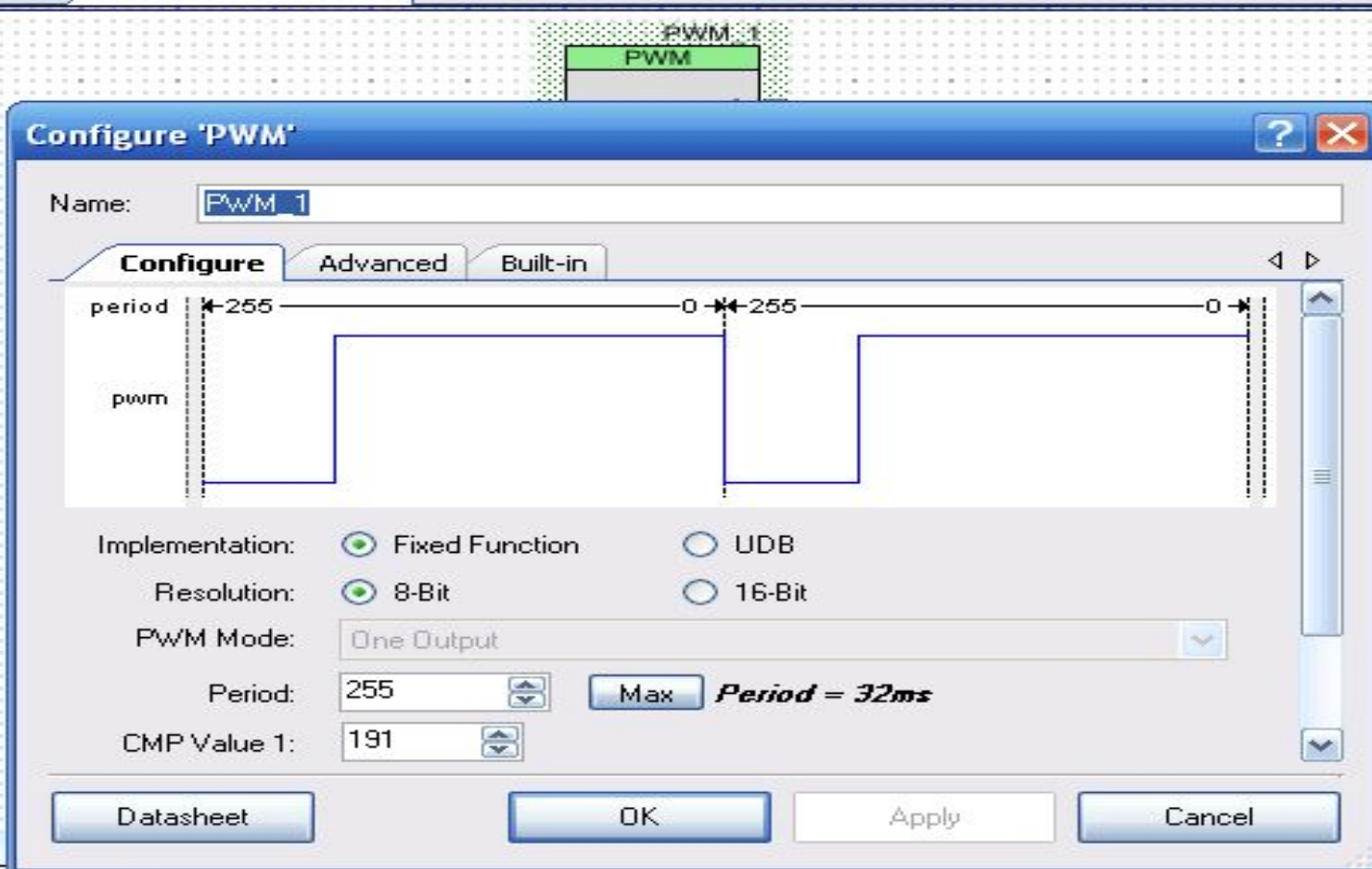
Page 1

Notice List

 1 Errors     0 Warnings     0 Notes    [Go To Error](#)    [View Details](#)

Description	File	Error Location
 sdb.M0050: Terminal "kill" requires connection when it is visible.	TopDesign.cysch	Signal: Net_22,...

Start Page \*TopDesign.cysch



**Configure 'PWM'**

Name:

Configure | Advanced | Built-in

period: 255 0 255 0

pwm

Implementation:  Fixed Function  UDB

Resolution:  8-Bit  16-Bit

PWM Mode: One Output

Period: 255  *Period = 32ms*

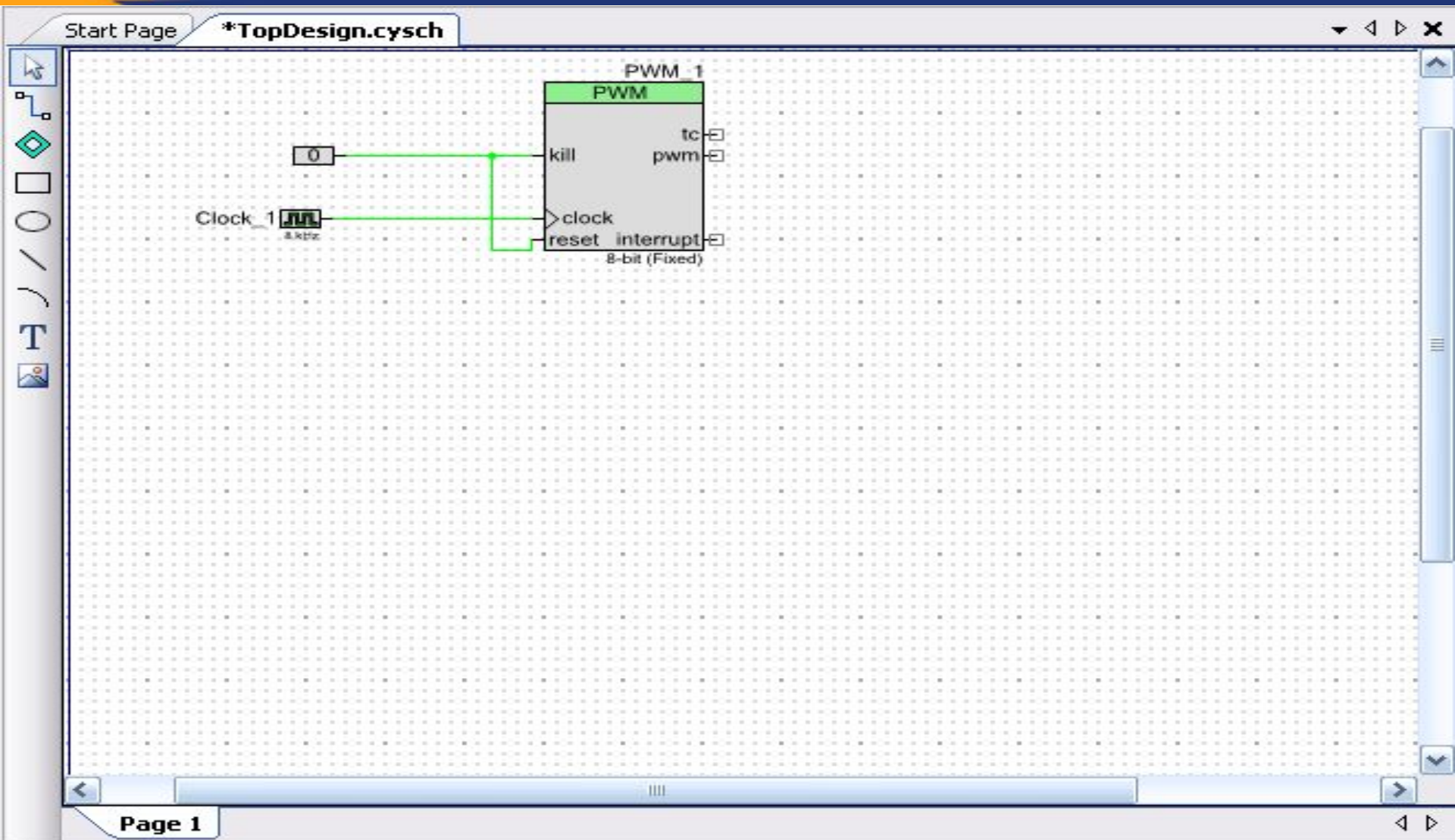
CMP Value 1: 191

Page 1

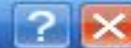
Notice List

1 Errors  0 Warnings  0 Notes

Description	File	Error Location
<input checked="" type="checkbox"/> sdb.M0050:Terminal "kill" requires connection when it is visible.	TopDesign.cysch	Signal: Net_22,...



## Configure 'cy\_constant'



Name: cy\_constant\_1

Basic

Built-in



Parameter	Value
DisplayAsHex	false
Value	0
Width	1

## Parameter Information

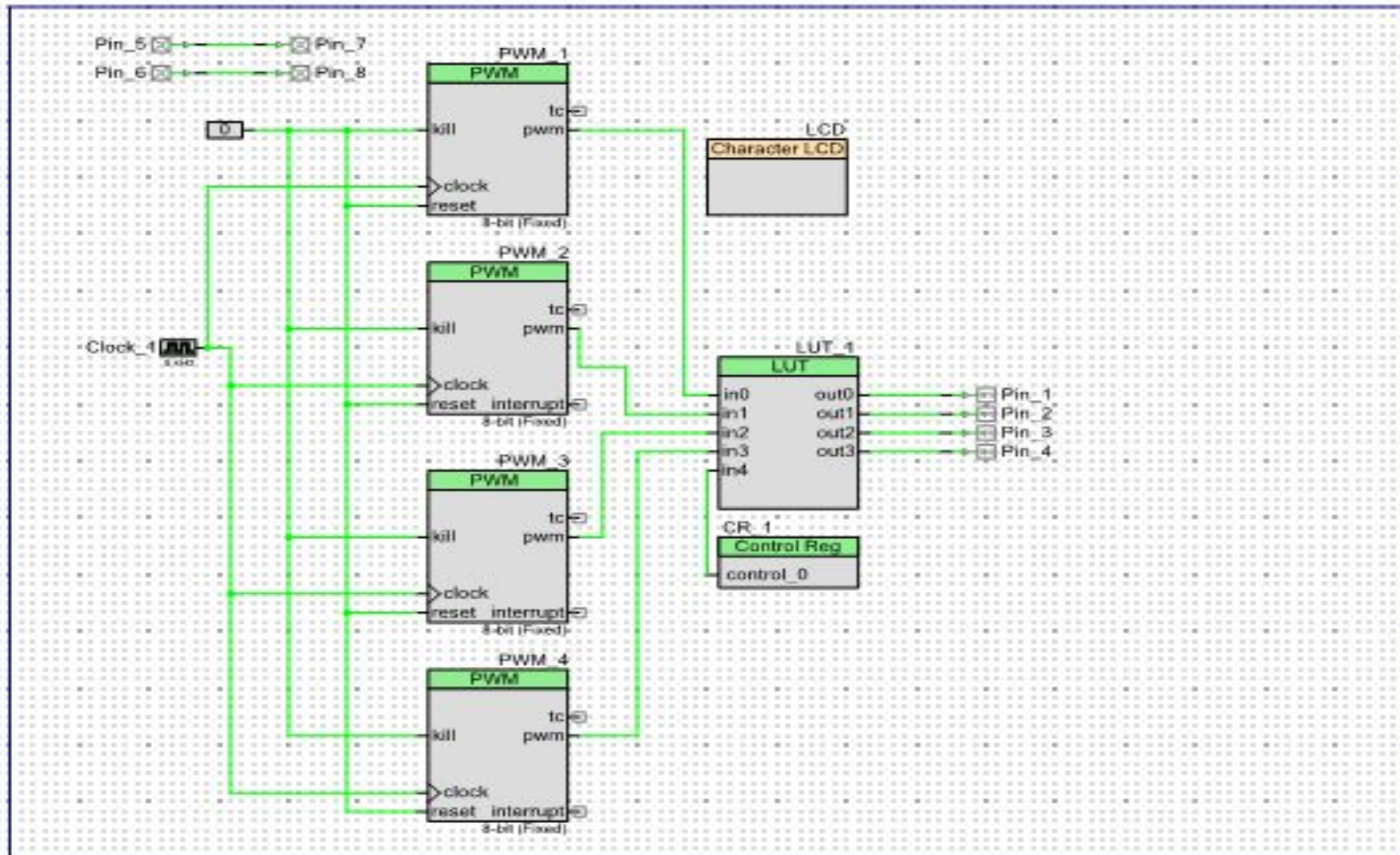
**DisplayAsHex:** Display value as hexadecimal (true) or decimal (false).**Value:****Type:** bool

Datasheet

OK

Apply

Cancel



## Configure 'CyControlReg'

Name:

CR\_1

**Configure**

Built-in

Outputs

1

Display as bus

External reset

Bit

Mode

Initial value

0

Direct

0

Direct

Set all modes

Datasheet

OK

Apply

Cancel



# Configuration Lucup Table

**Configure 'LUT'**

Name:

**Configure** Built-in

Inputs: 5 Outputs: 4  Register Outputs

Input Hex Value	in4	in3	in2	in1	in0		out3	out2	out1	out0	Output Hex Value
0x00	0	0	0	0	0		0	0	0	0	0x00
0x01	0	0	0	0	1		0	0	0	0	0x00
0x02	0	0	0	1	0		0	0	0	0	0x00
0x03	0	0	0	1	1		0	0	0	0	0x00
0x04	0	0	1	0	0		0	0	0	0	0x00

Set All Clear All

Datasheet OK Apply Cancel



# Configuration Lucup Table

Start Page | TopDesign.cysch | Steper\_Motor\_2.cydwr | main.c

## Configure 'LUT'

Name:

**Configure**

Built-in

Inputs

Outputs

5

4

Register Outputs

Input Hex Value	in4	in3	in2	in1	in0		out3	out2	out1	out0	Output Hex Value
0x0E	0	1	1	1	0		0	0	0	0	0x00
0x0F	0	1	1	1	1		0	0	0	0	0x00
0x10	1	0	0	0	0		0	0	0	0	0x00
0x11	1	0	0	0	1		0	0	0	1	0x01
0x12	1	0	0	1	0		0	0	1	0	0x02

Set All

Clear All

Datasheet

OK

Apply

Cancel

Page 1

Notice List

0 Errors

0 Warnings

0 Notes

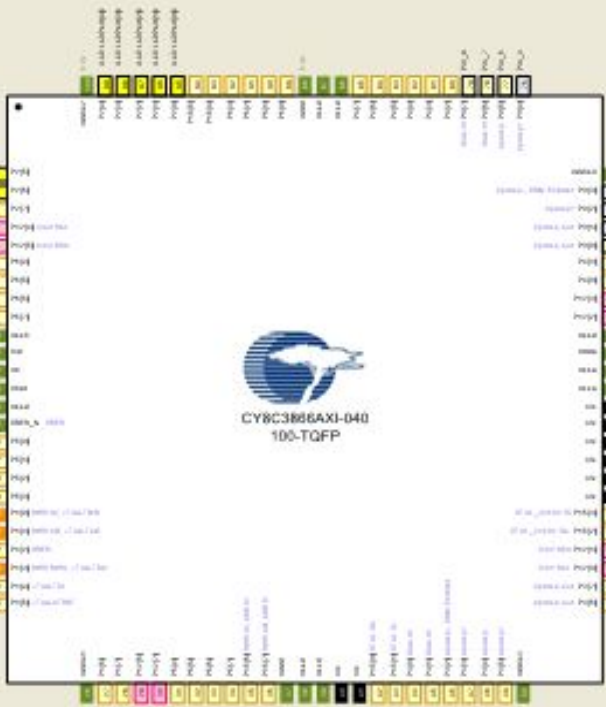
Go To Error

View Details



0x00	0	0	0	0	0	0	0	0	0	0x00
0x01	0	0	0	0	1	0	0	0	0	0x00
0x02	0	0	0	1	0	0	0	0	0	0x00
0x03	0	0	0	1	1	0	0	0	0	0x00
0x04	0	0	1	0	0	0	0	0	0	0x00
0x05	0	0	1	0	1	0	0	0	0	0x00
0x06	0	0	1	1	0	0	0	0	0	0x00
0x07	0	0	1	1	1	0	0	0	0	0x00
0x08	0	1	0	0	0	0	0	0	0	0x00
0x09	0	1	0	0	1	0	0	0	0	0x00
0x0A	0	1	0	1	0	0	0	0	0	0x00
0x0B	0	1	0	1	1	0	0	0	0	0x00
0x0C	0	1	1	0	0	0	0	0	0	0x00
0x0D	0	1	1	0	1	0	0	0	0	0x00
0x0E	0	1	1	1	0	0	0	0	0	0x00
0x0F	0	1	1	1	1	0	0	0	0	0x00
0x10	1	0	0	0	0	0	0	0	0	0x00
0x11	1	0	0	0	1	0	0	0	1	0x01
0x12	1	0	0	1	0	0	0	1	0	0x02
0x13	1	0	0	1	1	0	0	1	1	0x03
0x14	1	0	1	0	0	0	1	0	0	0x04
0x15	1	0	1	0	1	0	1	0	1	0x05
0x16	1	0	1	1	0	0	1	1	0	0x06
0x17	1	0	1	1	1	0	1	1	1	0x07
0x18	1	1	0	0	0	1	0	0	0	0x08
0x19	1	1	0	0	1	1	0	0	1	0x09
0x1A	1	1	0	1	0	1	0	1	0	0x0A
0x1B	1	1	0	1	1	1	0	1	1	0x0B
0x1C	1	1	1	0	0	1	1	0	0	0x0C
0x1D	1	1	1	0	1	1	1	0	1	0x0D
0x1E	1	1	1	1	0	1	1	1	0	0x0E

Start Page | TopDesign.cysch | **Steper\_Motor\_2.cydwr** | main.c



CY8C3866AXI-040  
105-TQFP

Alias	Name	Port	Pin	Loc
	\LCD:LCDPort[6:0]\	P2[6:0]	95..99,1..2	
Pin_1		P0[1] OpAmp:out	72	
Pin_2		P0[2] OpAmp+	73	
Pin_3		P0[3] OpAmp-, DSM: ExtVref	74	
Pin_4		P0[4] OpAmp+	76	
Pin_5		P0[0] OpAmp:out	71	
Pin_6		P0[5] OpAmp-	77	
Pin_7		P0[6] IDAC:HI	78	
Pin_8		P0[7] IDAC:HI	79	

LCD\_LCDPort\_6 - Digital  
LCD\_LCDPort\_5 - Digital

Pins |
 Analog |
 Clocks |
 Interrupts |
 DMA |
 System |
 Directives |
 Flash Security



CYPRESS

# Stepper Motor PSoC@3/5

Start Page TopDesign.cysch Steper\_Motor\_2.cydwr \*main.c

```
1  /* =====
2  *
3  * Copyright YOUR COMPANY, THE YEAR
4  * All Rights Reserved
5  * UNPUBLISHED, LICENSED SOFTWARE.
6  *
7  * CONFIDENTIAL AND PROPRIETARY INFORMATION
8  * WHICH IS THE PROPERTY OF your company.
9  *
10 * =====
11 */
12
13
14 #include <device.h>
15 #include <stdlib.h>
16 #include <string.h>
17
18
19
20
21 int pin_pwm_1; int pin_pwm_2; int pin_pwm_3; int pin_pwm_4; // Пини PWM(1-4)
22 int b_1; int b_2; // Пини кнопок(1-2)
23 int delay_period; // Пауза періода після старту PWM
24 int N; // Кількість мікрорухів для повного оберту
25 int temp; int temp2; int temp3;
26
```

Output

Show output from: All

----- Build Succeeded: 05/17/2013 07:23:43 -----



# Stepper Motor PSoC@3/5

```
Start Page TopDesign.cysch Steper_Motor_2.cydwr *main.c
20
21 int pin_pwm_1; int pin_pwm_2; int pin_pwm_3; int pin_pwm_4; // Піни PWM(1-4)
22 int b_1; int b_2; // Піни кнопок(1-2)
23 int delay_period; // Пауза періода після старта PWM
24 int N; // Кількість мікрорухів для повного оберту
25 int temp; int temp2; int temp3;
26
27 void PWM1(void) {PWM_1_Start(); CyDelay(delay_period);}
28 // pin_pwm_1=LED_PWM_1_Read(); while (pin_pwm_1==1) {pin_pwm_1=LED_PWM_1_Read();};}
29
30 void PWM2(void) {PWM_2_Start(); CyDelay(delay_period);}
31 // pin_pwm_2=LED_PWM_2_Read(); while (pin_pwm_2==1) {pin_pwm_2=LED_PWM_2_Read();};}
32
33 void PWM3(void) {PWM_3_Start(); CyDelay(delay_period);}
34 // pin_pwm_3=LED_PWM_3_Read(); while (pin_pwm_3==1) {pin_pwm_3=LED_PWM_3_Read();};}
35
36 void PWM4(void) {PWM_4_Start(); CyDelay(delay_period);}
37 // pin_pwm_4=LED_PWM_4_Read(); while (pin_pwm_4==1) {pin_pwm_4=LED_PWM_4_Read();};}
38
39 void PWM_ALL_STOP(void) {PWM_1_Stop(); PWM_2_Stop(); PWM_3_Stop(); PWM_4_Stop();}
40
41 void SIDE_1(int NN) {for (temp=0; temp<NN; temp++) {PWM1(); PWM2(); PWM3(); PWM4();} PWM_AL
42 void SIDE_2(int NN) {for (temp=0; temp<NN; temp++) {PWM1(); PWM4(); PWM3(); PWM2();} PWM_AL
43
44 void READ_BUTTONS(void) {CyDelay(10); b_1=Pin_5_Read(); b_2=Pin_6_Read();}
45
```

Output

Show output from: All

----- Build Succeeded: 05/17/2013 07:23:43 -----



# Stepper Motor PSoC@3/5

Start Page | TopDesign.cysch | Steper\_Motor\_2.cydwr | \*main.c

```
35
36 void PWM4(void) {PWM_4_Start(); CyDelay(delay_period);}
37 // pin_pwm_4=LED_PWM_4_Read(); while (pin_pwm_4==1) {pin_pwm_4=LED_PWM_4_Read();}
38
39 void PWM_ALL_STOP(void) {PWM_1_Stop(); PWM_2_Stop(); PWM_3_Stop(); PWM_4_Stop();}
40
41 void SIDE_1(int NN) {for (temp=0; temp<NN; temp++) {PWM1(); PWM2(); PWM3(); PWM4()}
42 void SIDE_2(int NN) {for (temp=0; temp<NN; temp++) {PWM1(); PWM4(); PWM3(); PWM2()}
43
44 void READ_BUTTONS(void) {CyDelay(10); b_1=Pin_5_Read(); b_2=Pin_6_Read();}
45
46 void MOTOR(char motor, char way, char period)
47 {
48     CR_1_Write(0);
49     if (motor=='L') {CR_1_Write (1); }//if (way=='U') {way='D';} else {way='U';}
50
51     if (motor=='L')
52     {
53         if ((way=='U') &&(period=='M')) {SIDE_1(N*1);} // лѝва ввѝрх четвѝртинно
54
55         if ((way=='D') &&(period=='M')) {SIDE_2(N*1);} // лѝва вниѝ четвѝртинно
56     }
57
58 }
59
60
```

Output

Show output from: All

----- Build Succeeded: 05/17/2013 07:23:43 -----



CYPRESS

# Stepper Motor PSoC@3/5

Start Page

TopDesign.cysch

Steper\_Motor\_2.cydwr

\*main.c

```
50
51     if (motor=='L')
52     {
53     if ((way=='U') &&(period=='M')) {SIDE_1(N*1);} // ліва вверх четвєртинно
54
55     if ((way=='D') &&(period=='M')) {SIDE_2(N*1);} // ліва вниз четвєртинно
56     }
57
58 }
59
60 void main()
61 {
62     Clock_1_Enable();
63     delay_period=1;
64     N=20;
65     MOTOR('L','U','M');
66     //MOTOR('L','D','M');
67     /* Place your initialization/startup code here (e.g. MyInst_Start()) */
68
69     /* CyGlobalIntEnable; */ /* Uncomment this line to enable global interrupts. */
70     for(;;)
71     {
72         /* Place your application code here. */
73     }
74 }
75
76 /* [] END OF FILE */
```

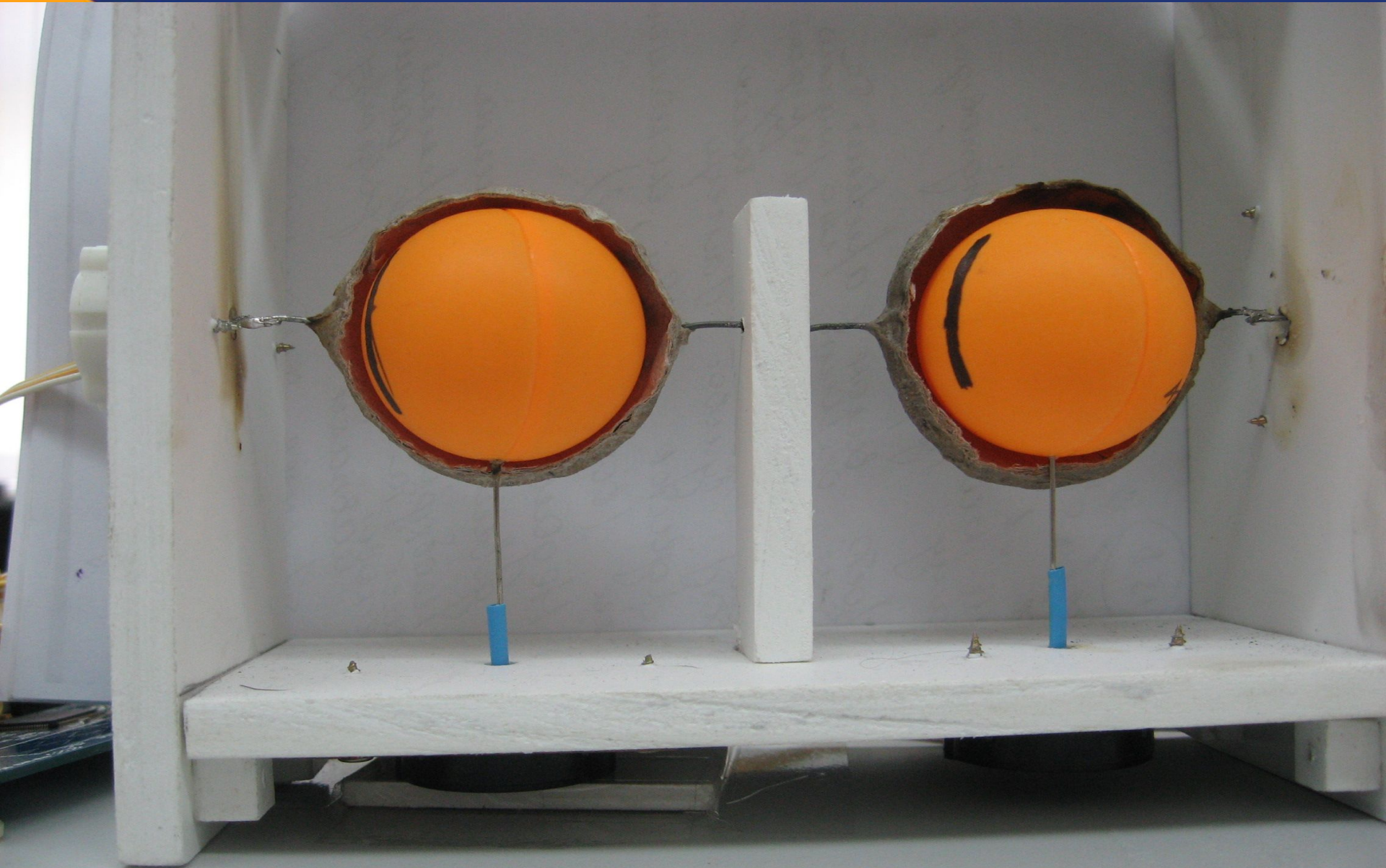
Output

Show output from: All

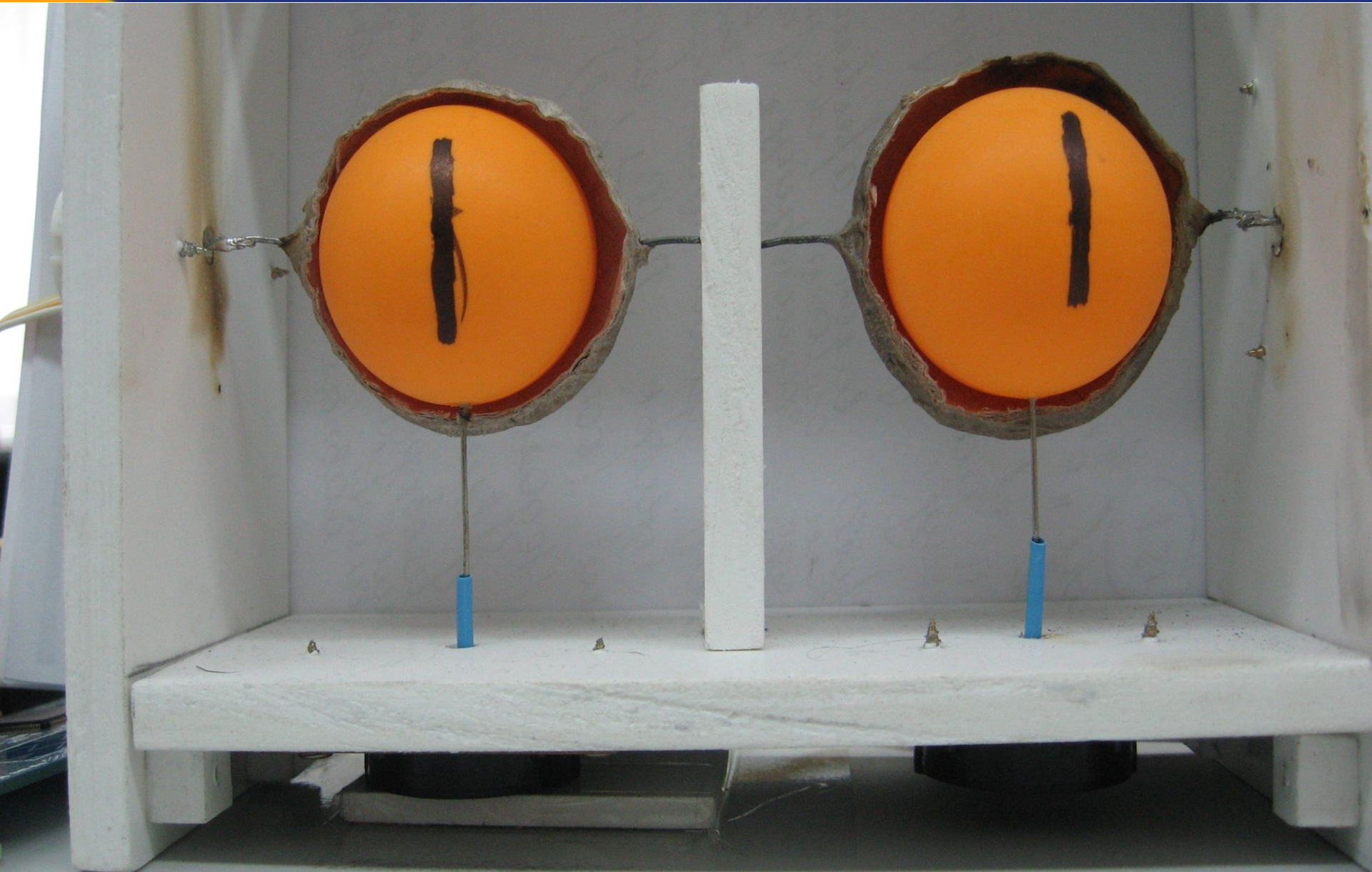
----- Build Succeeded: 05/17/2013 07:23:43 -----

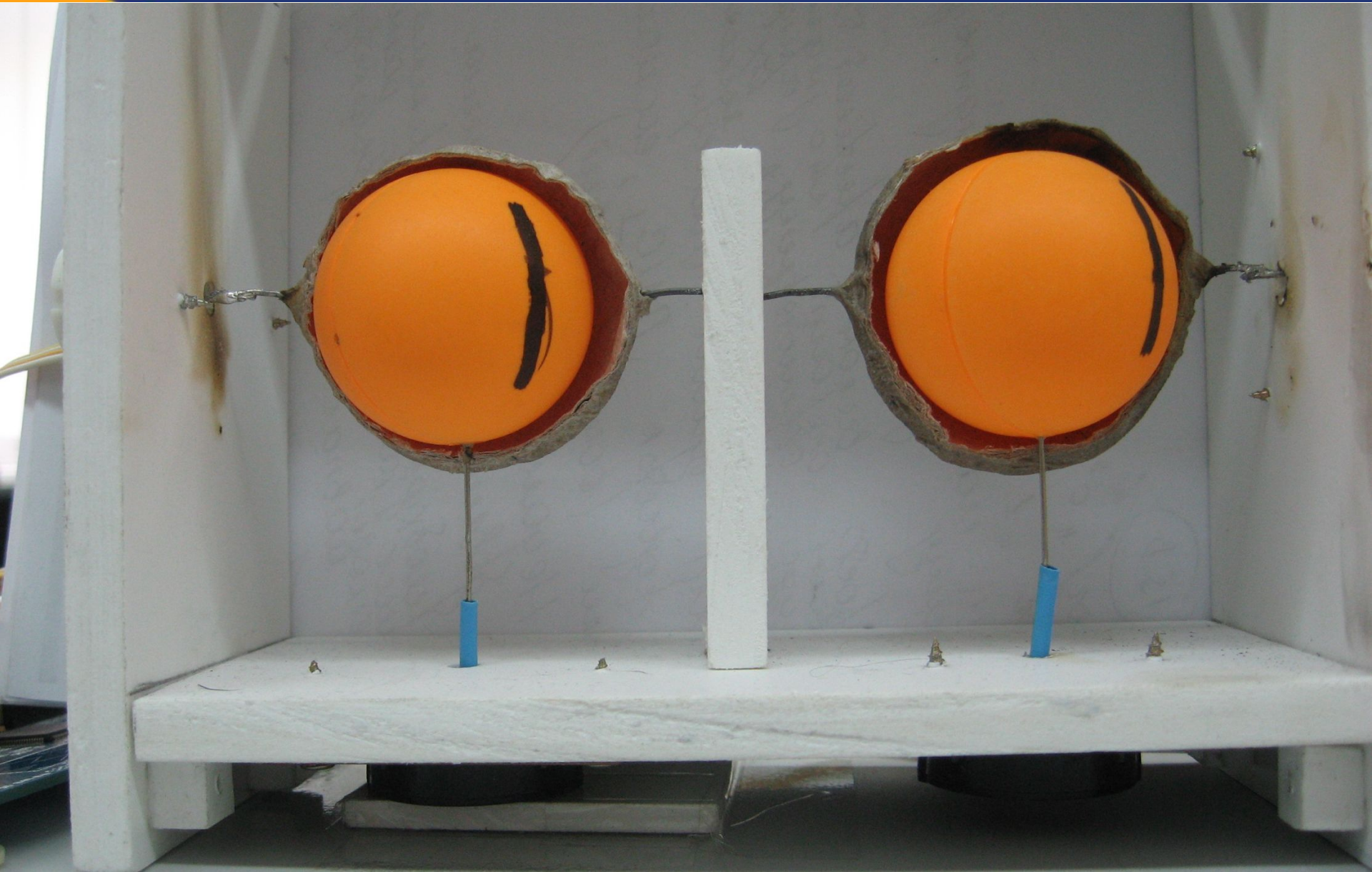
## **РОБОТ-АВТОМАТ НА ПЛАТФОРМІ PSoC@3 / 5**

**Робот-автомат виконує встановлення та утримання в заданому положенні кута повороту роторів чотирьох мікропотужних крокових електродвигунів постійного струму в залежності від заданих режимів роботи та керується двома імпульсними сигналами, які подаються на два електромагніти зміщені в просторі на один градус.**











## **Рекомендована література :**

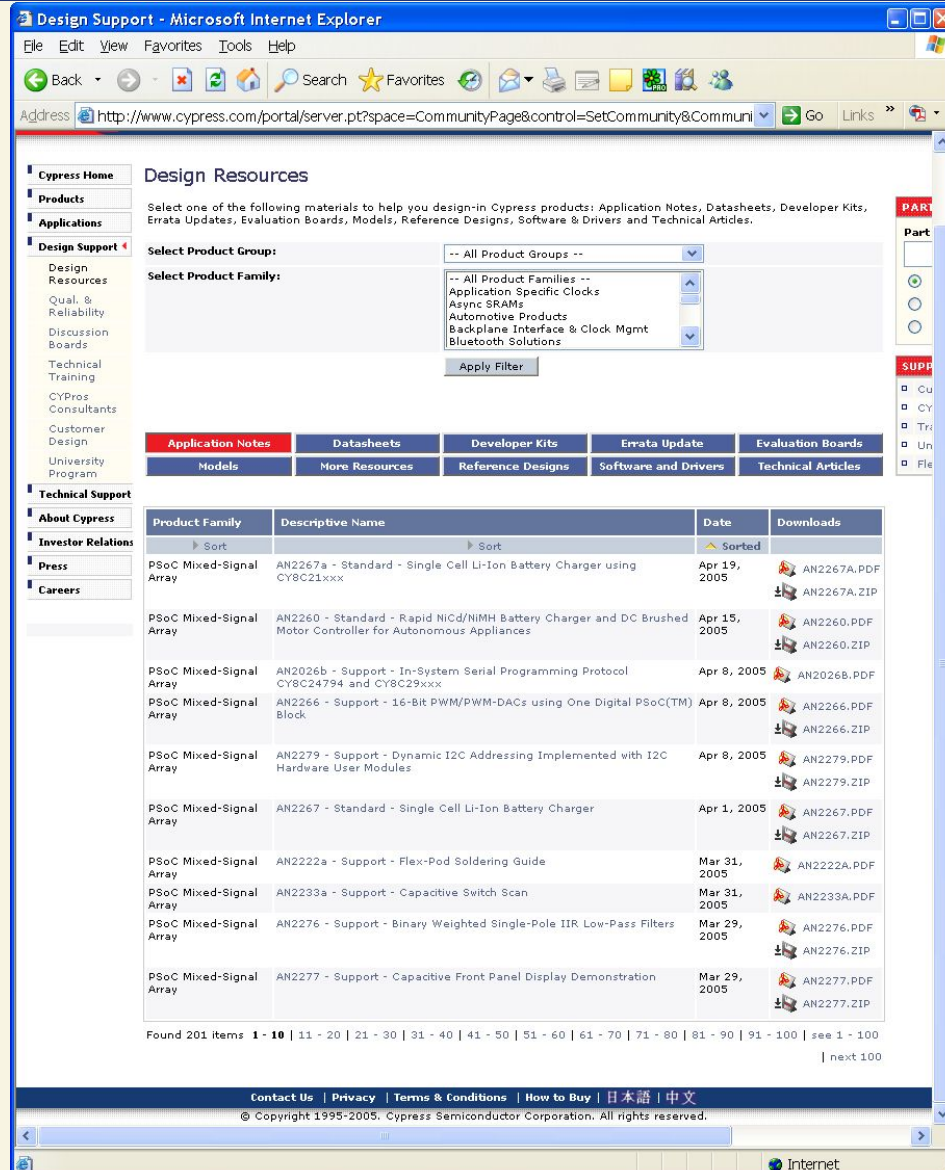
### ***Основна:***

1. AN56551 PSoC® 3 Stepper Motor Control with Precision Microstepping. 2011 . - 15 с.
1. PSoC® 3 Architecture TRM (Technical Reference Manual) – 379 с.

### ***Додаткова:***

1. PSoC® Creator™ Component Datasheet. PWM. DAC8 – 6 с.

На сайті фірми Cypress знаходиться більше **200** Application Notes і Reference Designs, які ілюструють області застосування мікроконтролерів PSoC.



Design Support - Microsoft Internet Explorer

Address: <http://www.cypress.com/portal/server.pt?space=CommunityPage&control=SetCommunity&Communi>

**Design Resources**

Select one of the following materials to help you design-in Cypress products: Application Notes, Datasheets, Developer Kits, Errata Updates, Evaluation Boards, Models, Reference Designs, Software & Drivers and Technical Articles.

Select Product Group: -- All Product Groups --

Select Product Family: -- All Product Families --  
 Application Specific Clocks  
 Async SRAMs  
 Automotive Products  
 Backplane Interface & Clock Mgmt  
 Bluetooth Solutions

Apply Filter

Application Notes	Datasheets	Developer Kits	Errata Update	Evaluation Boards
Models	More Resources	Reference Designs	Software and Drivers	Technical Articles

Product Family	Descriptive Name	Date	Downloads
PSoC Mixed-Signal Array	AN2267a - Standard - Single Cell Li-Ion Battery Charger using CY8C21xxx	Apr 19, 2005	AN2267A.PDF AN2267A.ZIP
PSoC Mixed-Signal Array	AN2260 - Standard - Rapid NiCd/NiMH Battery Charger and DC Brushed Motor Controller for Autonomous Appliances	Apr 15, 2005	AN2260.PDF AN2260.ZIP
PSoC Mixed-Signal Array	AN2026b - Support - In-System Serial Programming Protocol CY8C24794 and CY8C29xxx	Apr 8, 2005	AN2026B.PDF
PSoC Mixed-Signal Array	AN2266 - Support - 16-Bit PWM/PWM-DACs using One Digital PSoC(TM) Block	Apr 8, 2005	AN2266.PDF AN2266.ZIP
PSoC Mixed-Signal Array	AN2279 - Support - Dynamic I2C Addressing Implemented with I2C Hardware User Modules	Apr 8, 2005	AN2279.PDF AN2279.ZIP
PSoC Mixed-Signal Array	AN2267 - Standard - Single Cell Li-Ion Battery Charger	Apr 1, 2005	AN2267.PDF AN2267.ZIP
PSoC Mixed-Signal Array	AN2222a - Support - Flex-Pod Soldering Guide	Mar 31, 2005	AN2222A.PDF
PSoC Mixed-Signal Array	AN2233a - Support - Capacitive Switch Scan	Mar 31, 2005	AN2233A.PDF
PSoC Mixed-Signal Array	AN2276 - Support - Binary Weighted Single-Pole IIR Low-Pass Filters	Mar 29, 2005	AN2276.PDF AN2276.ZIP
PSoC Mixed-Signal Array	AN2277 - Support - Capacitive Front Panel Display Demonstration	Mar 29, 2005	AN2277.PDF AN2277.ZIP

Found 201 items 1 - 10 | 11 - 20 | 21 - 30 | 31 - 40 | 41 - 50 | 51 - 60 | 61 - 70 | 71 - 80 | 81 - 90 | 91 - 100 | see 1 - 100 | next 100

Contact Us | Privacy | Terms & Conditions | How to Buy | 日本語 | 中文

© Copyright 1995-2005. Cypress Semiconductor Corporation. All rights reserved.

# Мікропроцесорн а техніка 2019 р. (лекція 8, кінець) Благітко Б.Я.

