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

How They Work

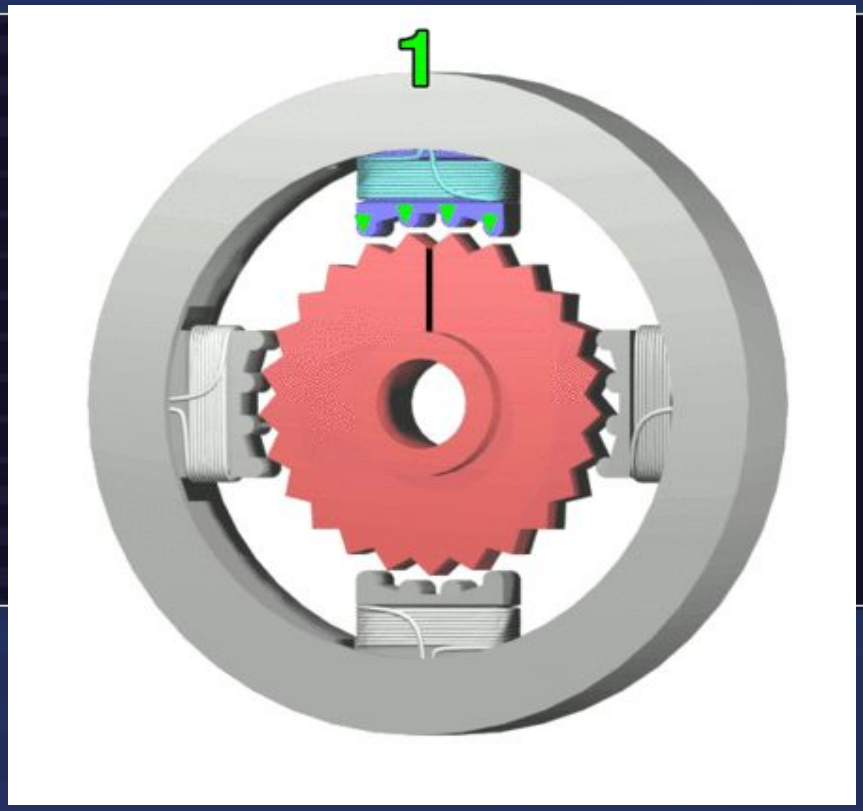
A DC motor uses a commutator to supply continuous direct current to a coil placed between two or more magnets.

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.

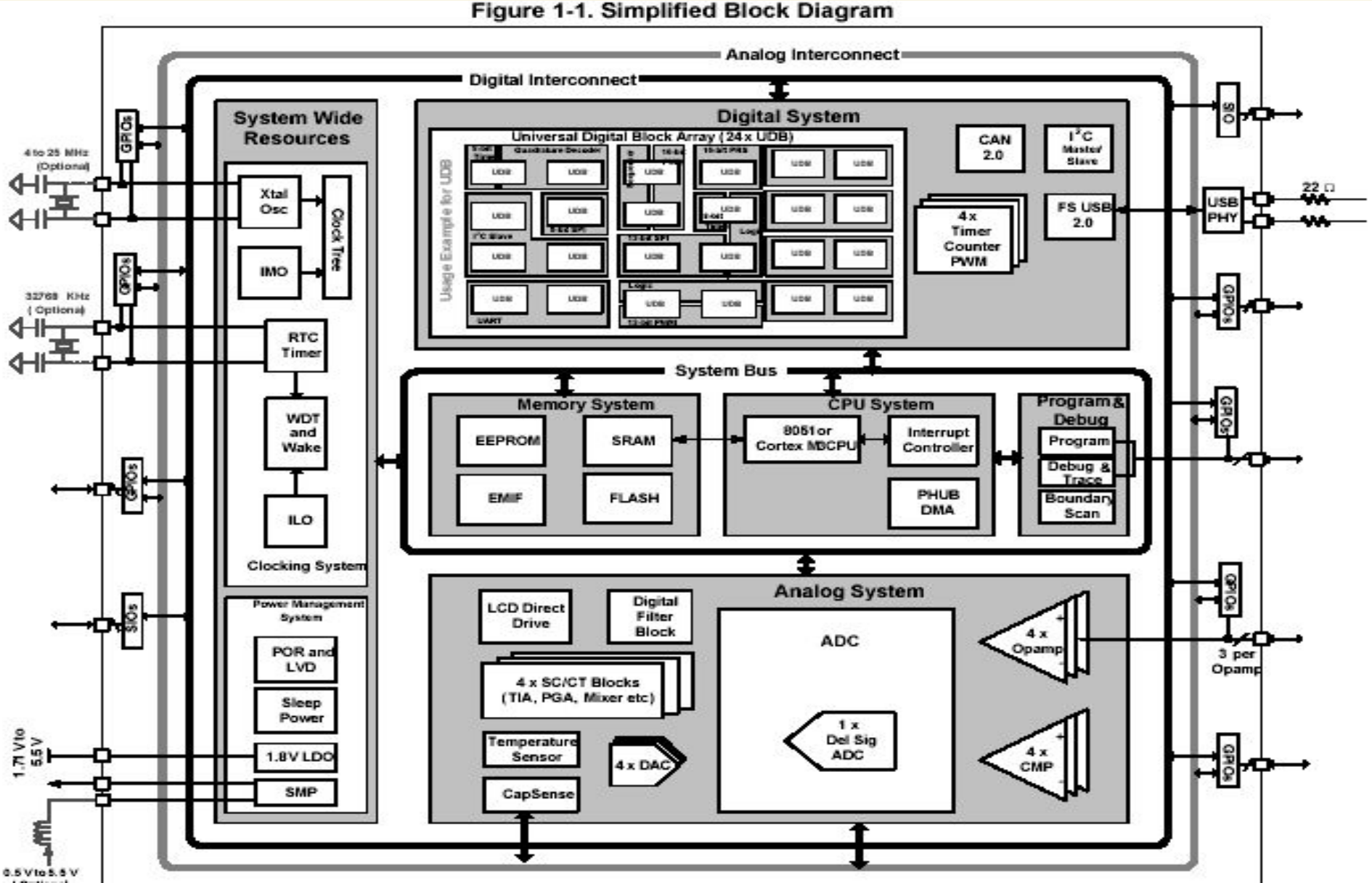
Read more: DC Motor vs. Stepper Motor | eHow.com http://www.ehow.com/facts_6777501_dc-motor-vs_-stepper-motor.html#ixzz2QKmmwrgy

Stepper Motor



Цифрові модулі

Figure 1-1. Simplified Block Diagram



Цифрові модулі

Figure 7-3. Component Catalog

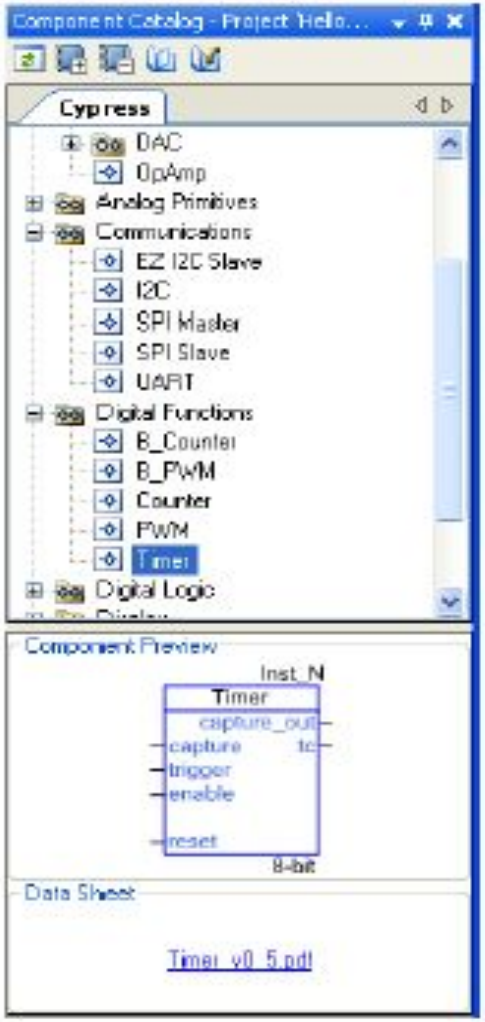
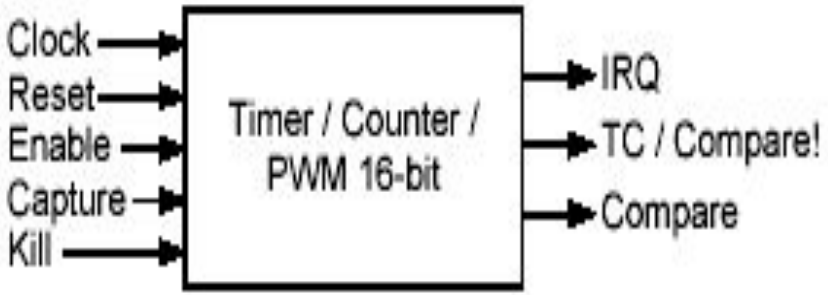
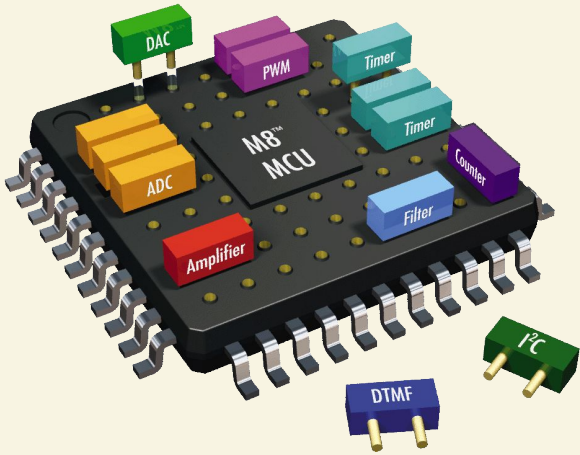


Figure 7-21. Timer/Counter/PWM





PWMs and Counters

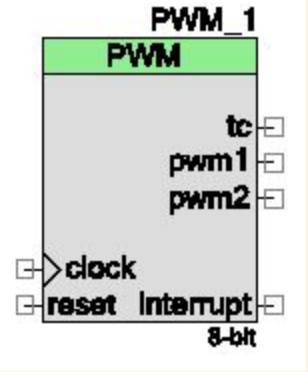
PWMs and **Counter**s share many capabilities but each provides specific capabilities.

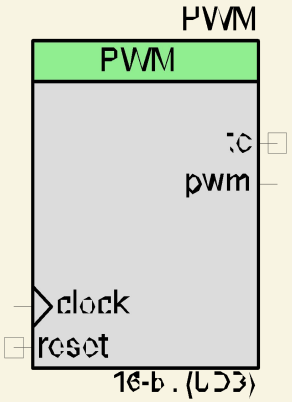
When to Use a PWM

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.

PWM

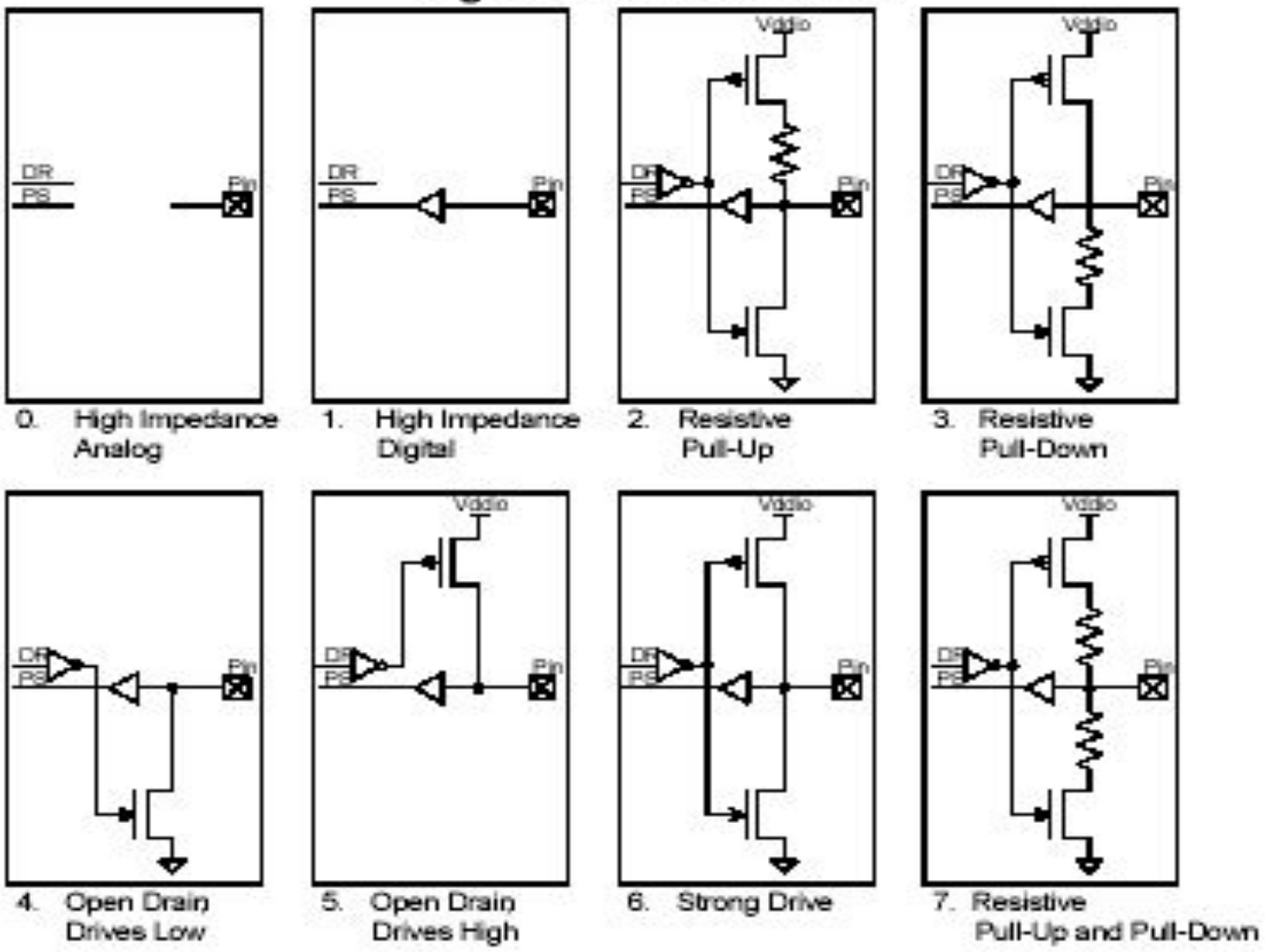
Output	May Be Hidden	Description
tc	N	The terminal count output is '1' when the period counter is equal to zero. In normal operation this output will be '1' for a single cycle where the counter is reloaded with period. If the PWM is stopped with the period counter equal to zero then this signal will 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.





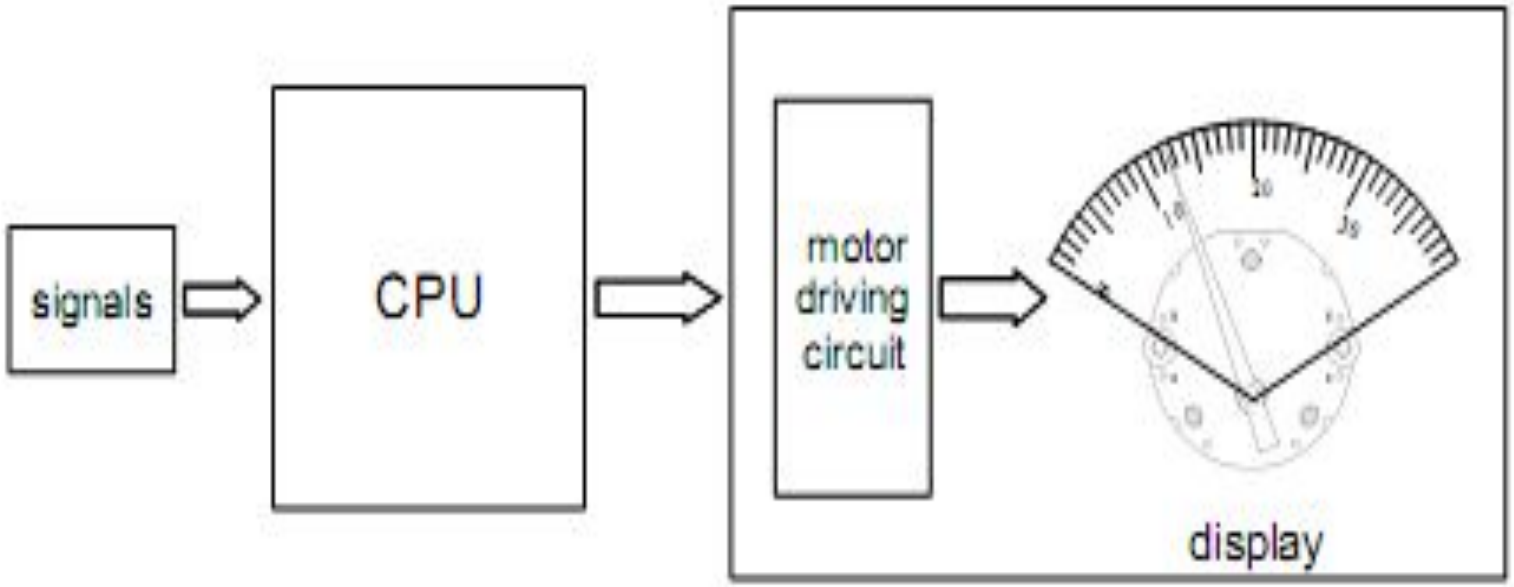
Port's Pins

Figure 6-11. Drive Mode



Stepper Motor

TYPICAL APPLICATION



perfect combination of digital accuracy and analog facility

Lab_8 Stepper Motor

VID29 Series Cluster Stepper Motor

MAIN FEATURES

- High speed rotation: 600°/sec.
- High μ-step resolution: 1/12°.
- Wide working voltage: 3.5~10V.
- Wide working temperature: -40 ~ 105℃.
- Low current consumption.
- Extremely robust construction: Φ30mm X 7.6mm.

Lab_8 Stepper Motor

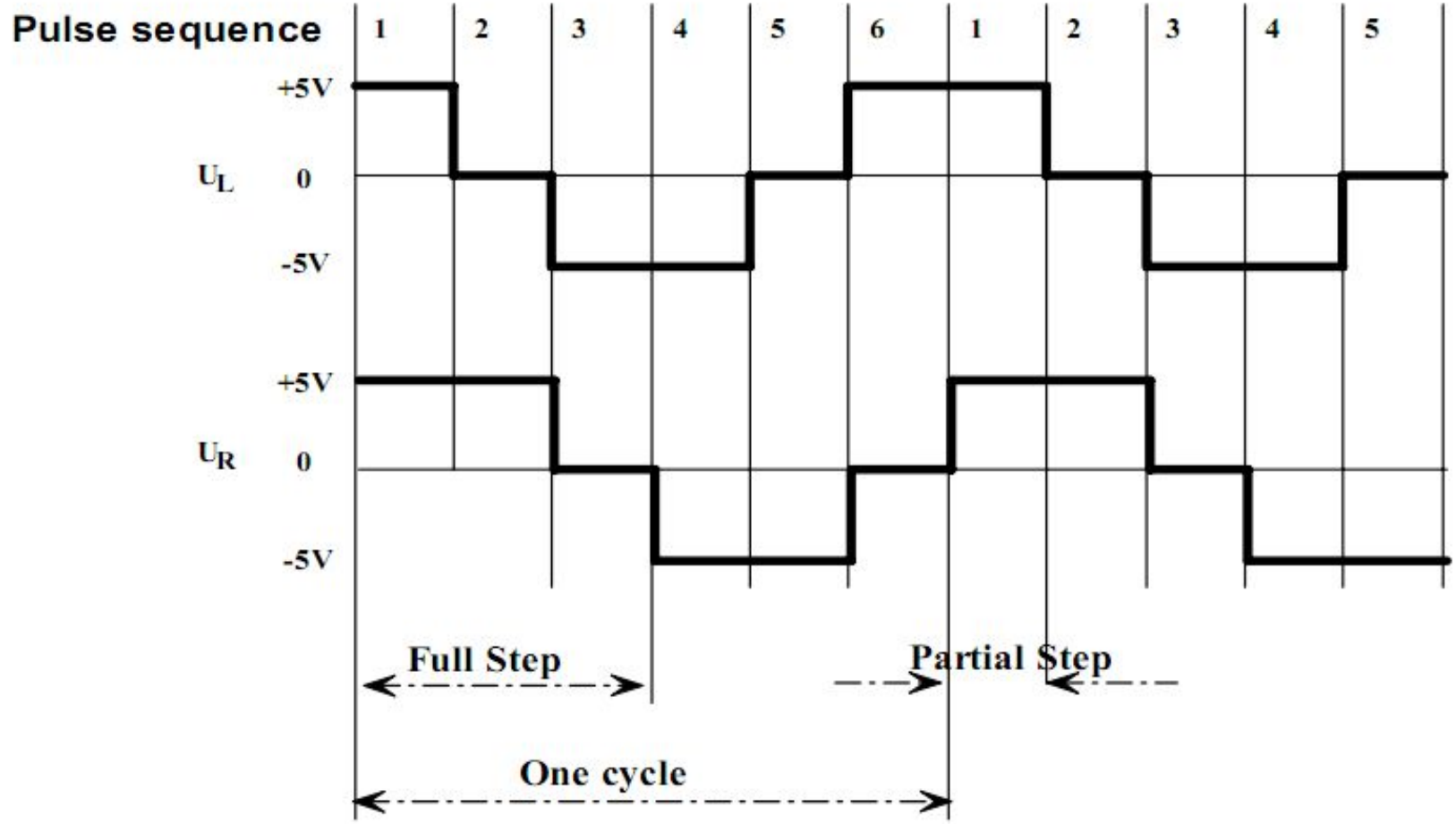
FRAEN AMD's 6405-15xx stepper motors

Lab_8 Stepper Motor

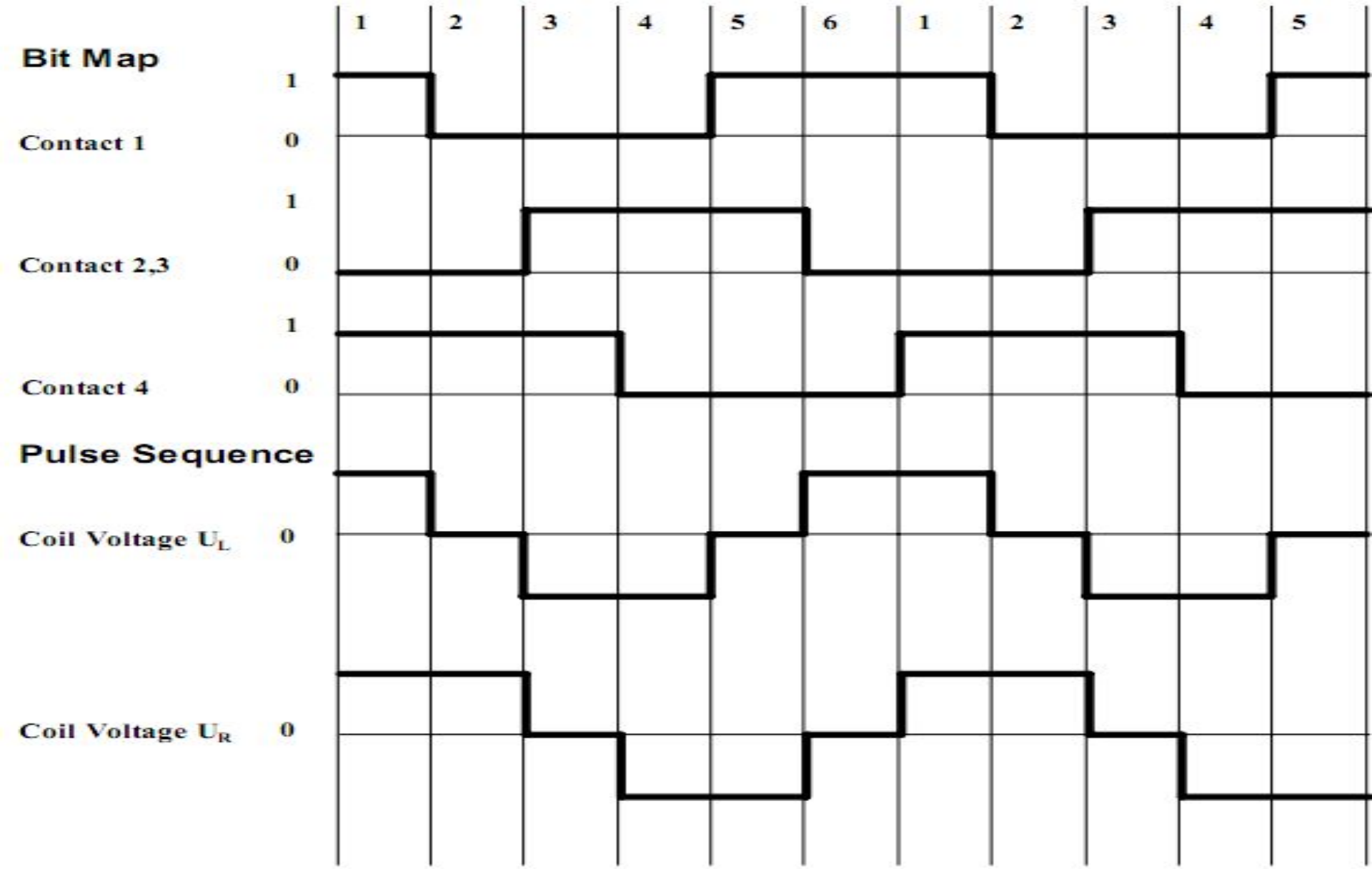
Stepper Motors

STEP DEFINITION AND ROTOR MOVEMENT

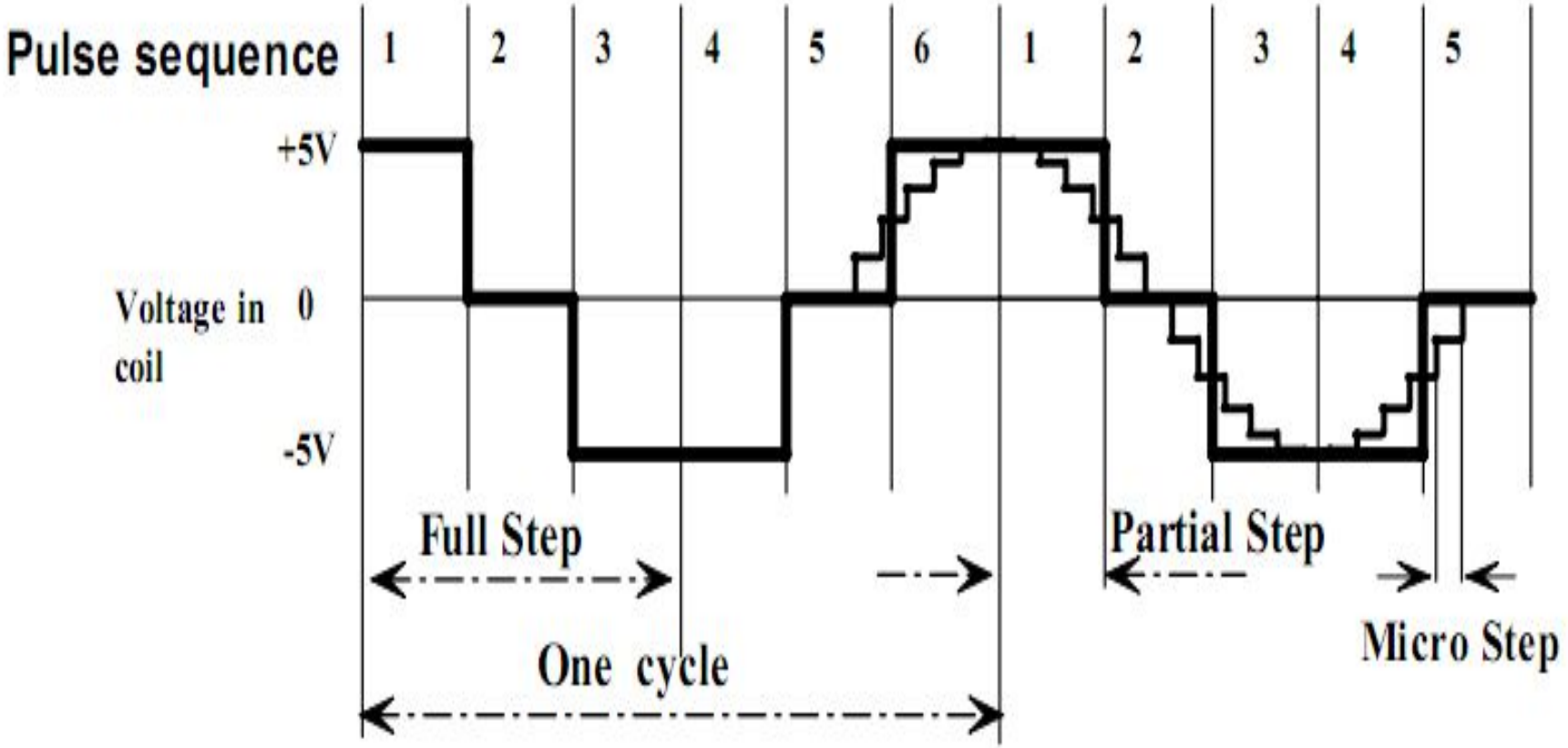
VID29-**XX** and VID29-**XX**P series motor are driven by 2 sequent logic pulse singnals. Its work diagram is as following:



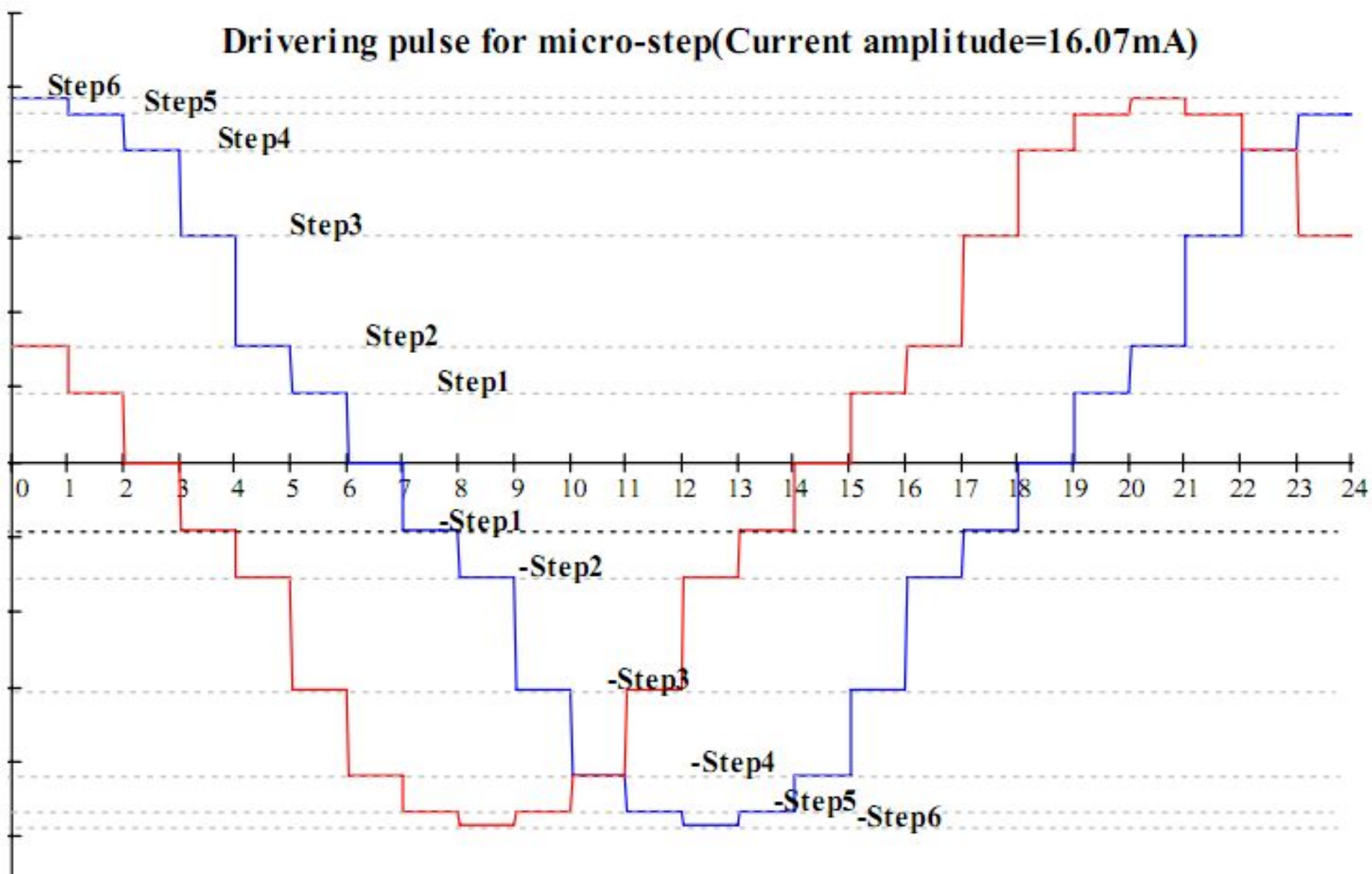
Lab_8 Stepper Motor



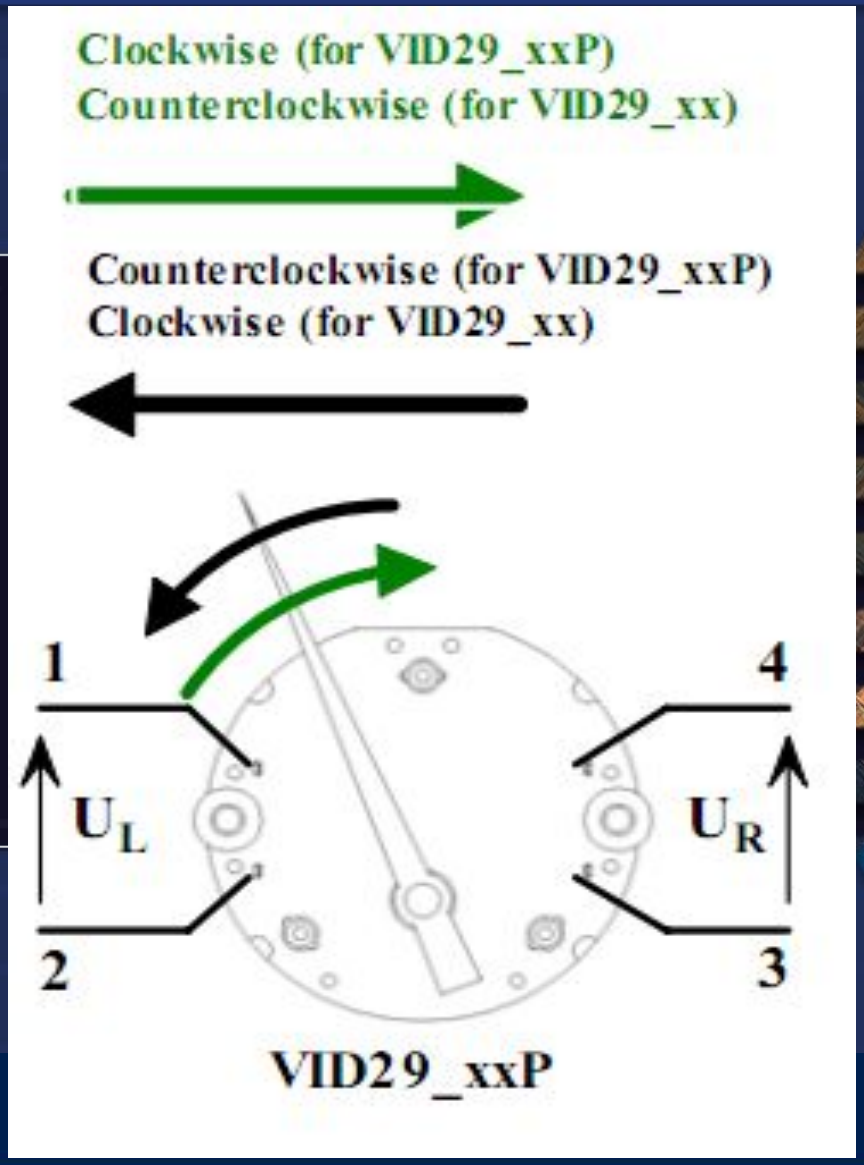
Lab_8 Stepper Motor



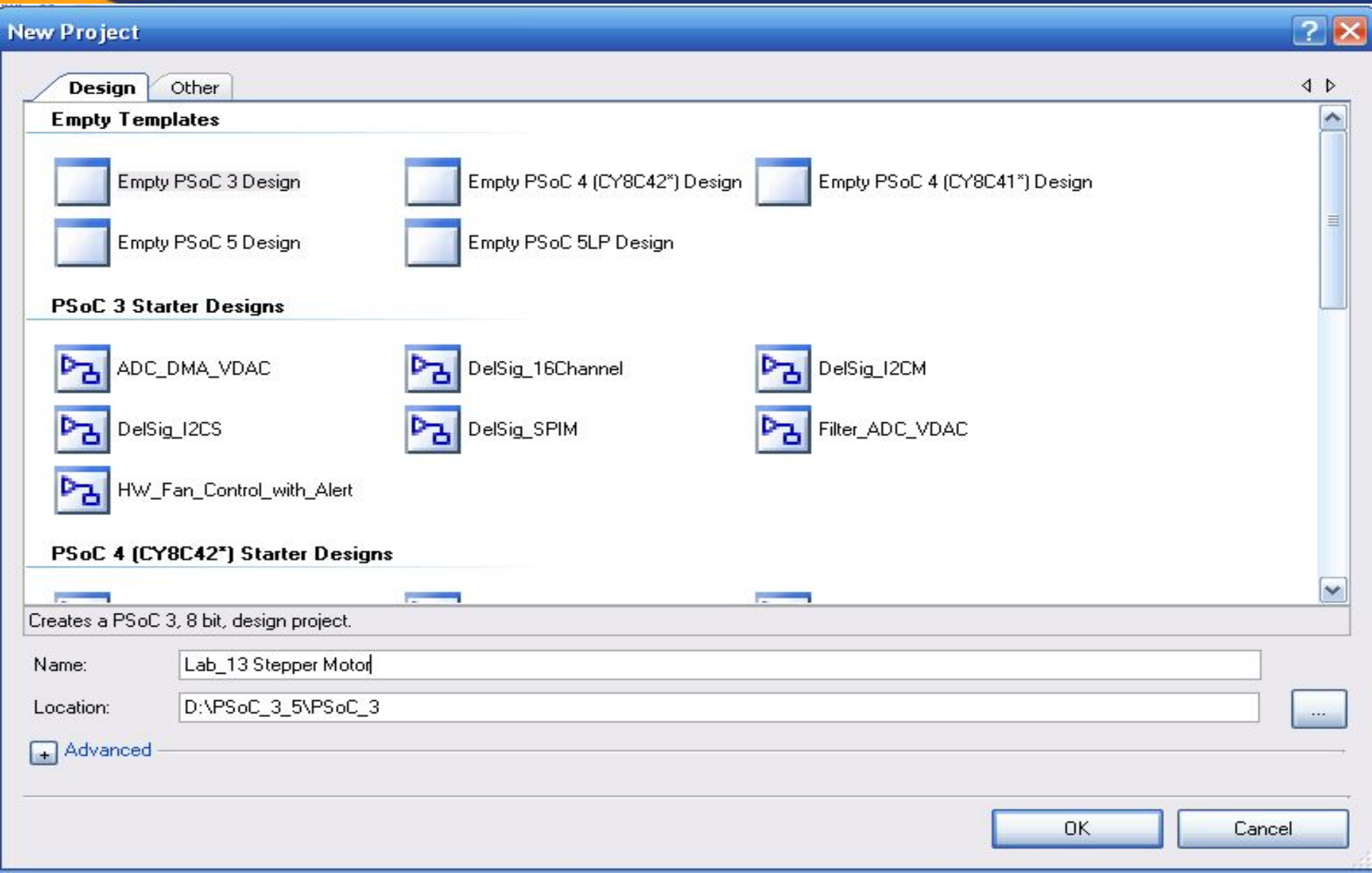
Lab_8 Stepper Motor



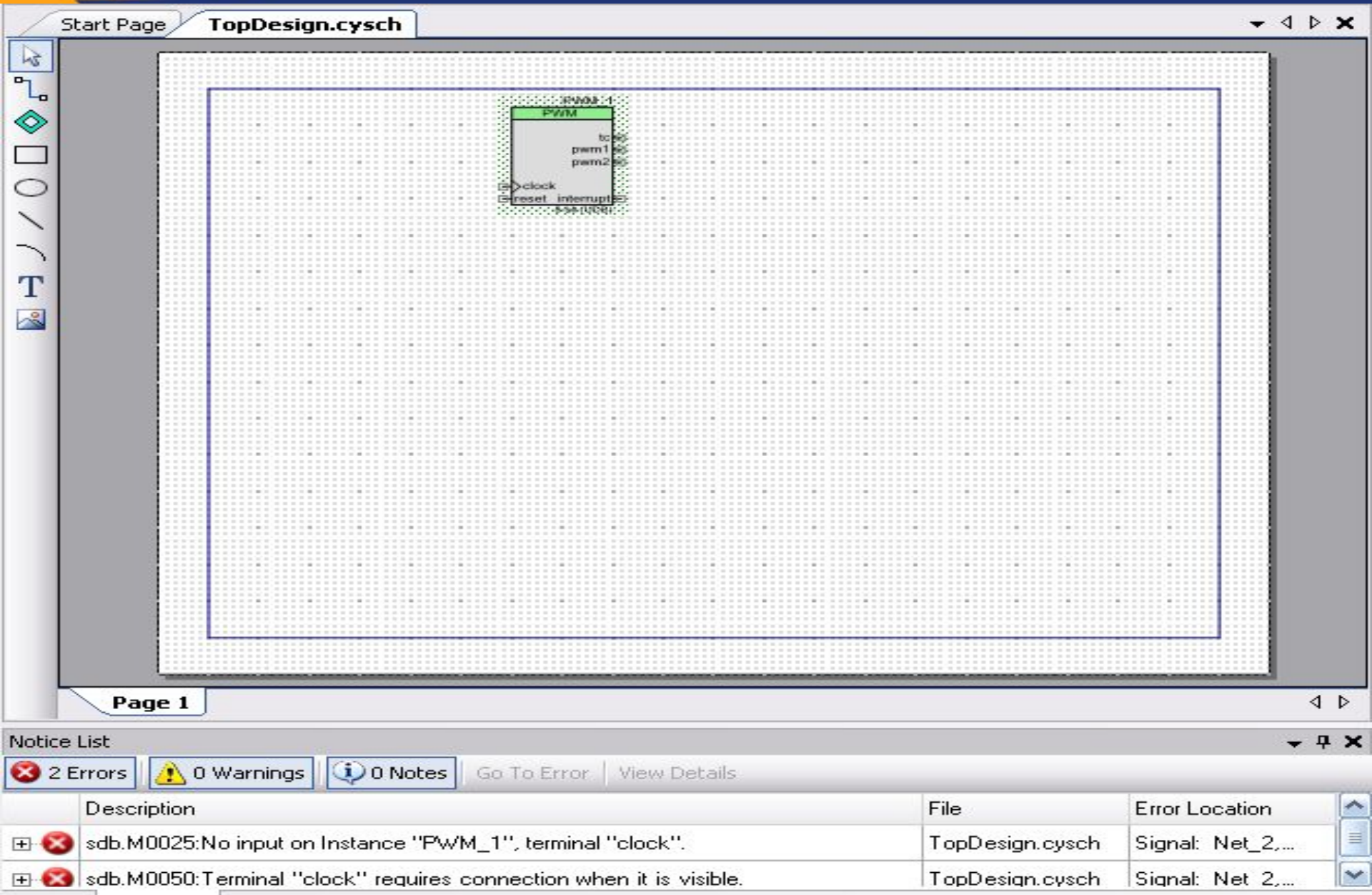
Lab_8 Stepper Motor



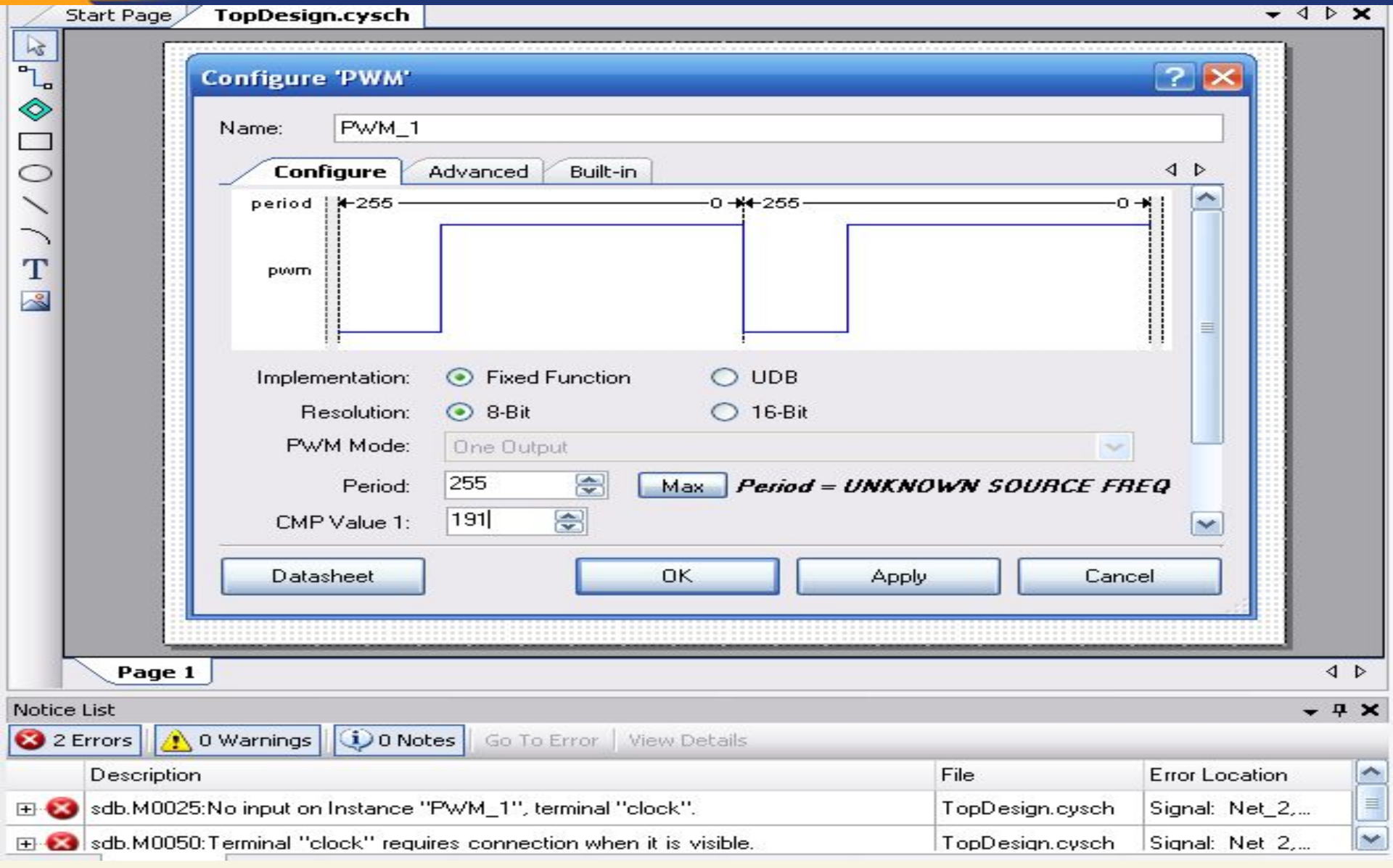
Lab_8 Stepper Motor



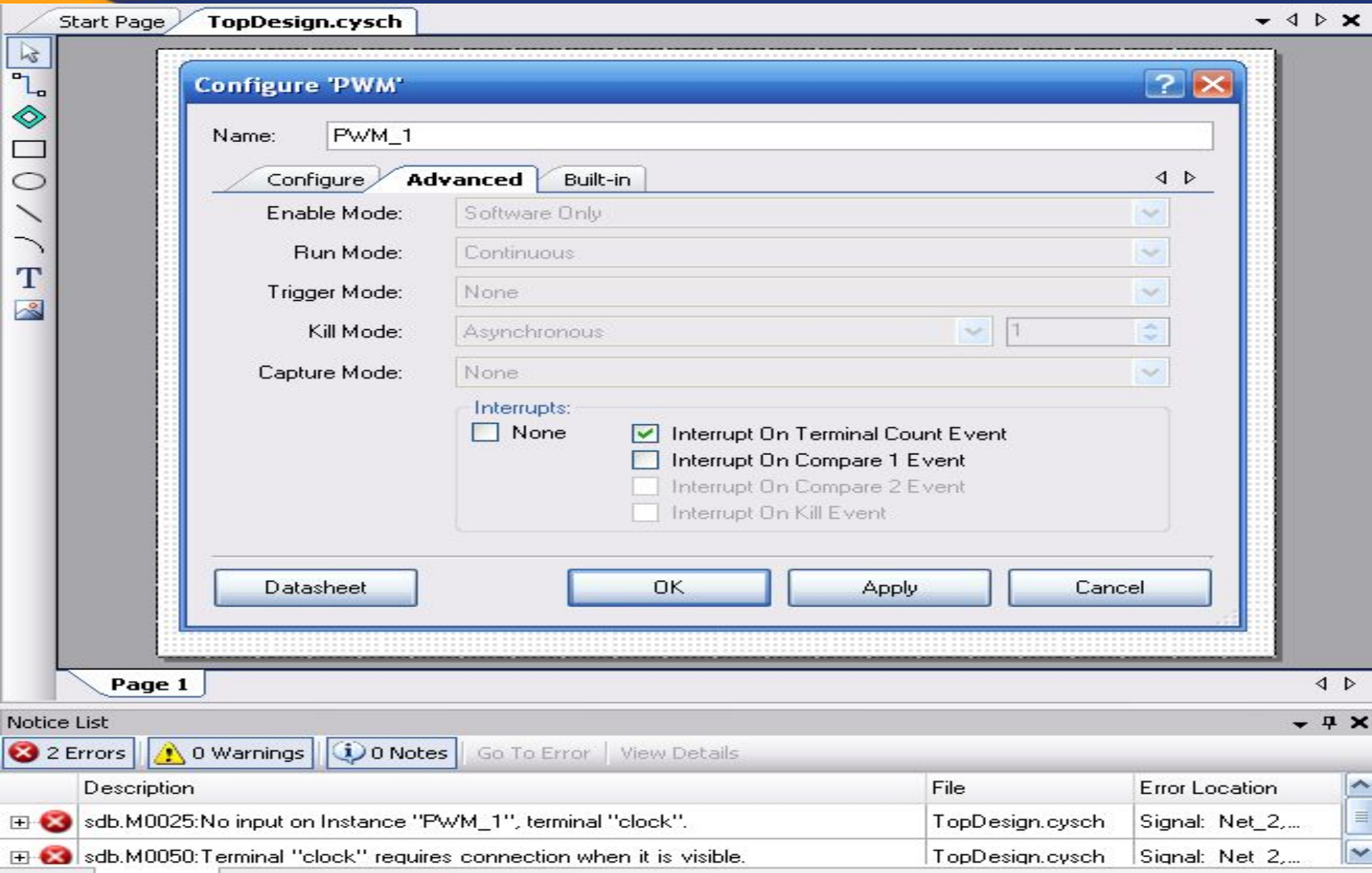
Lab_8 Stepper Motor



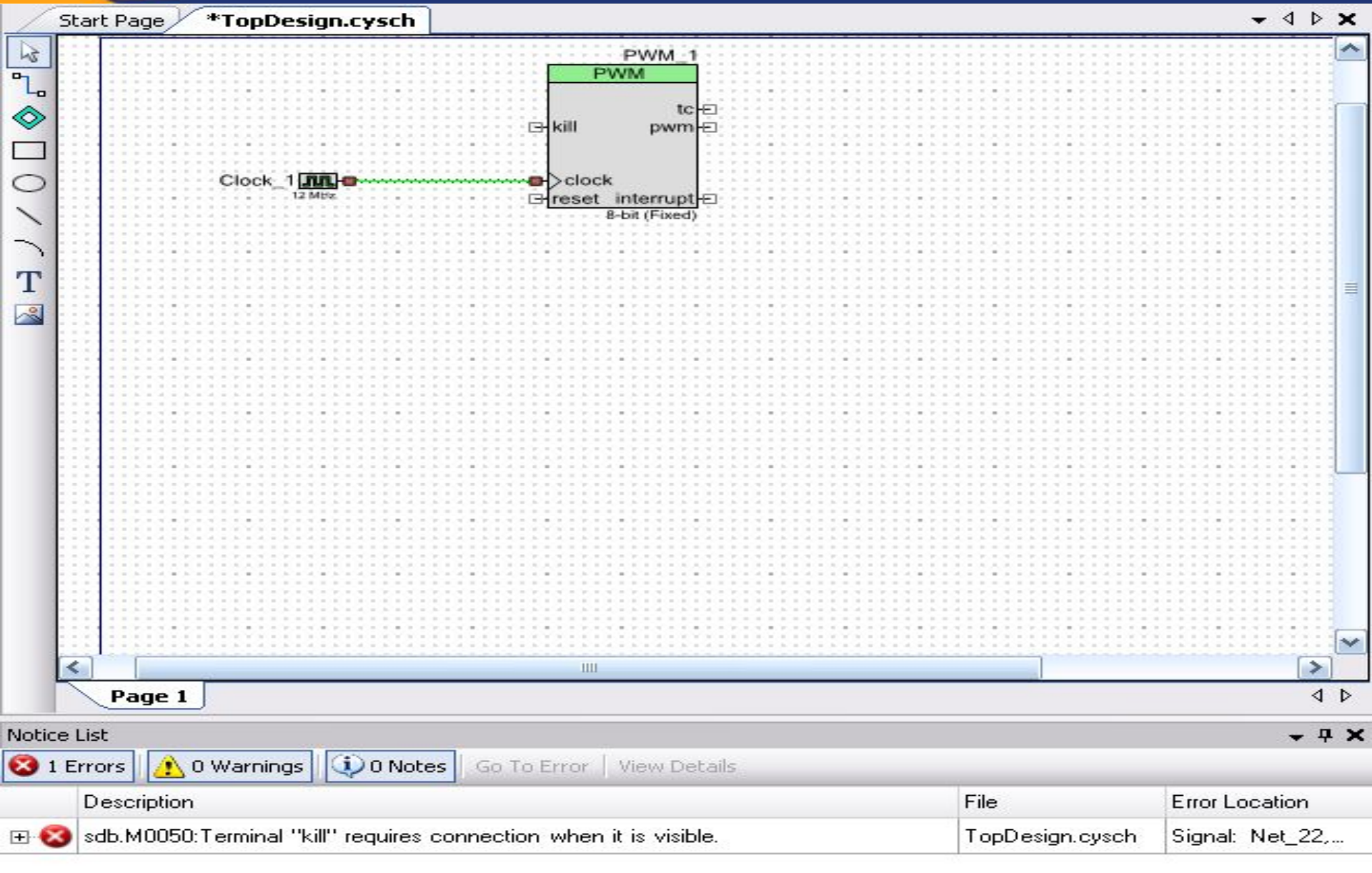
Lab_8 Stepper Motor



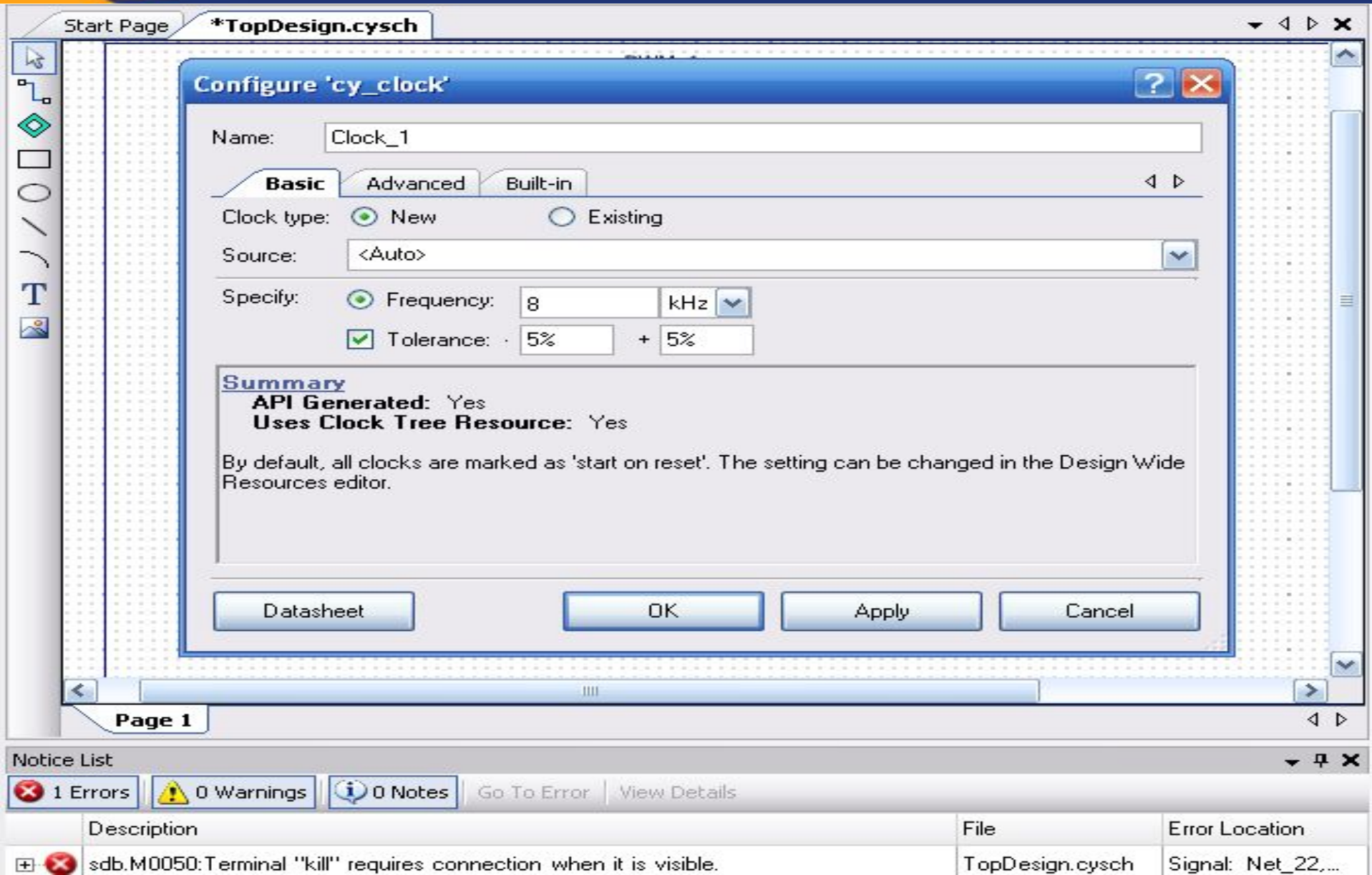
Lab_8 Stepper Motor



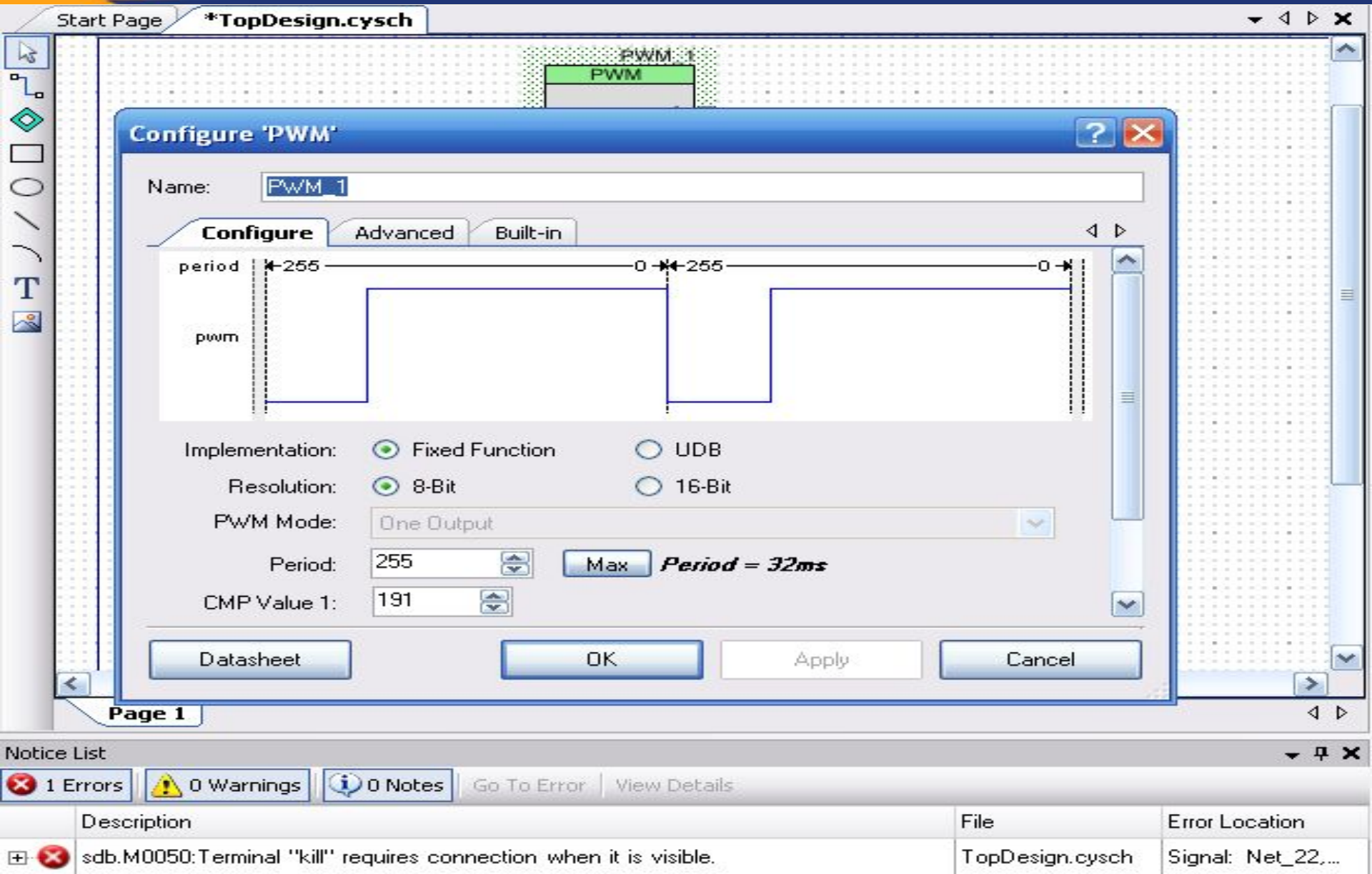
Lab_8 Stepper Motor



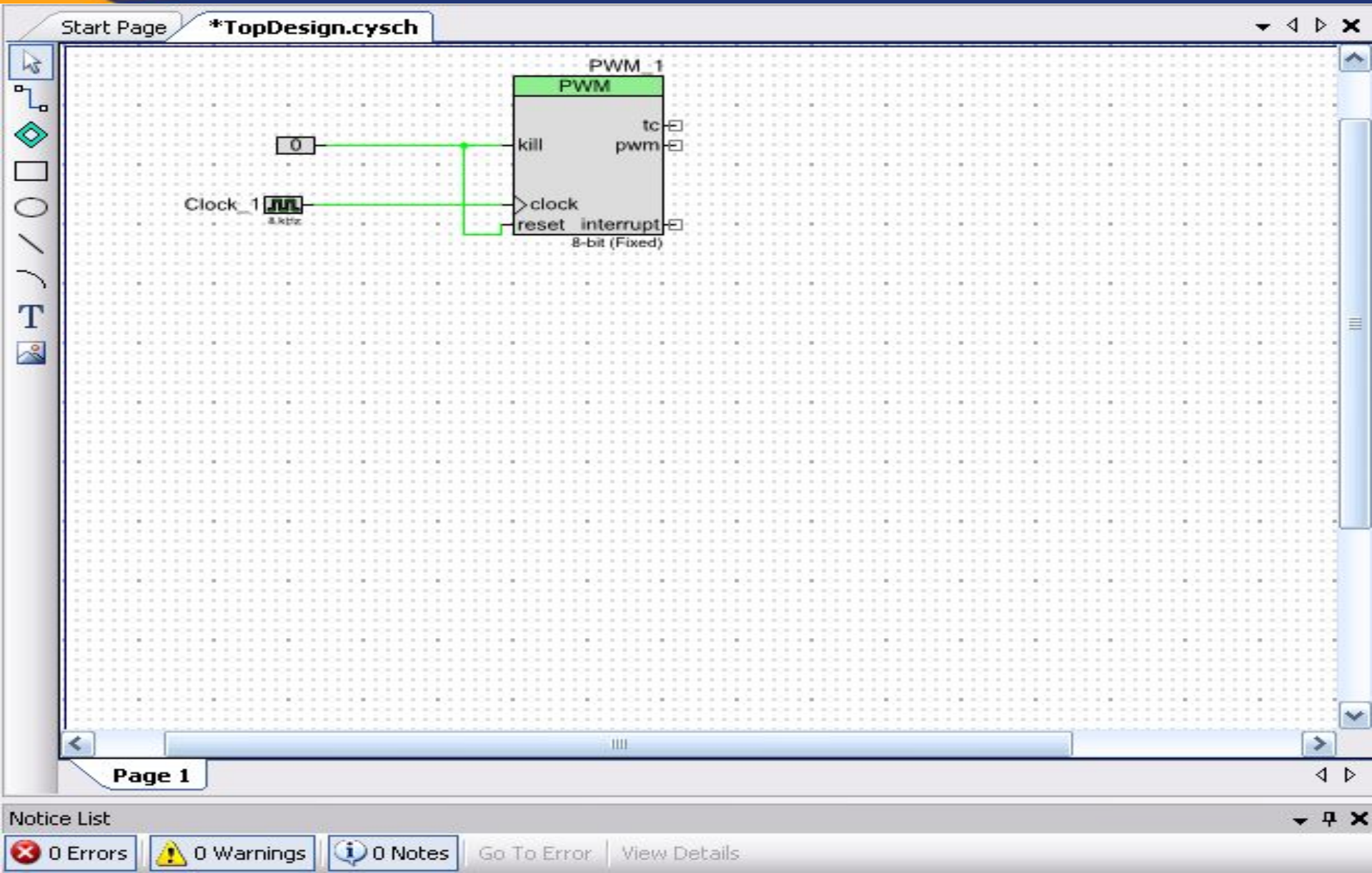
Lab_8 Stepper Motor



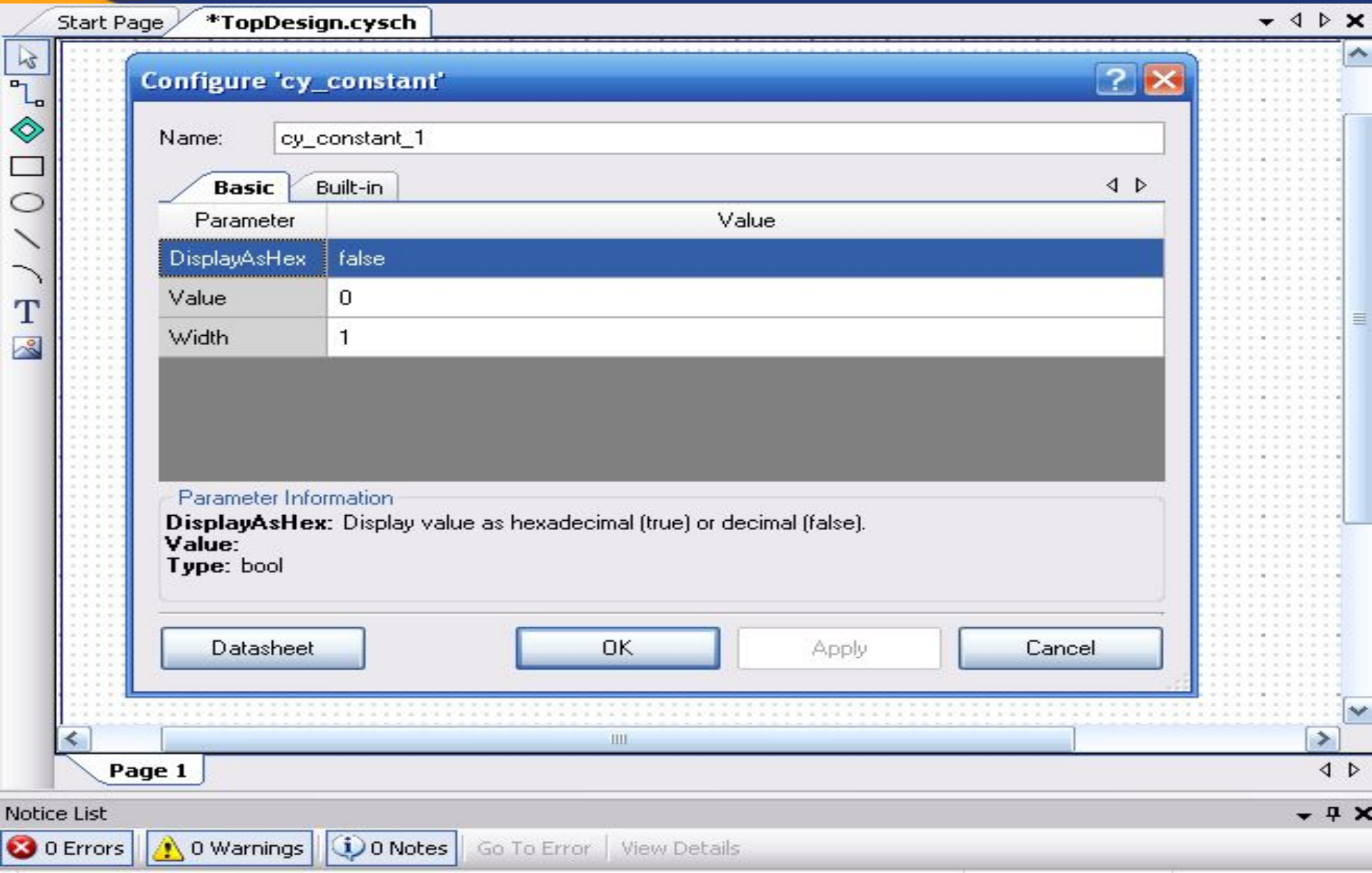
Lab_8 Stepper Motor



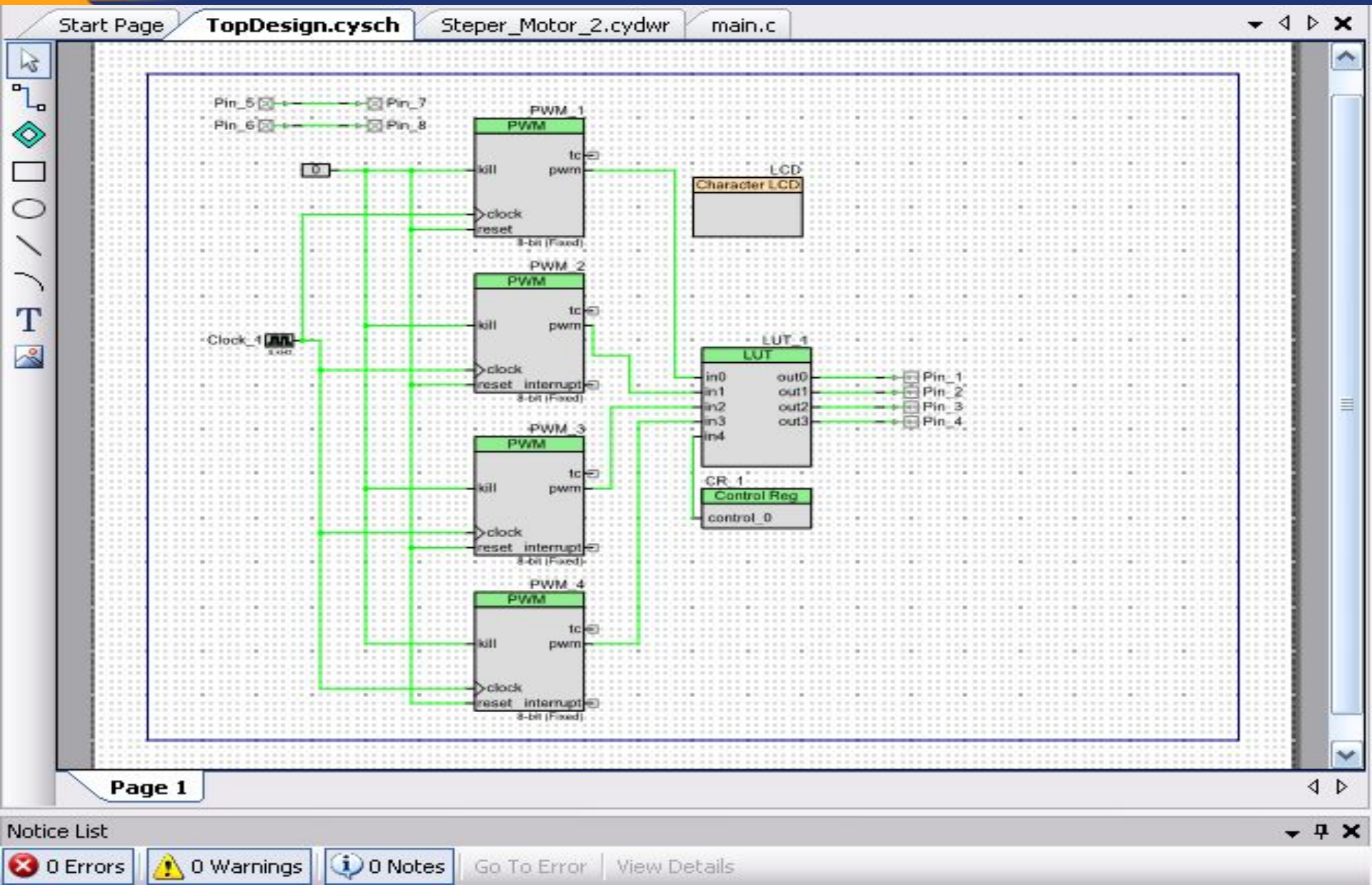
Lab_8 Stepper Motor



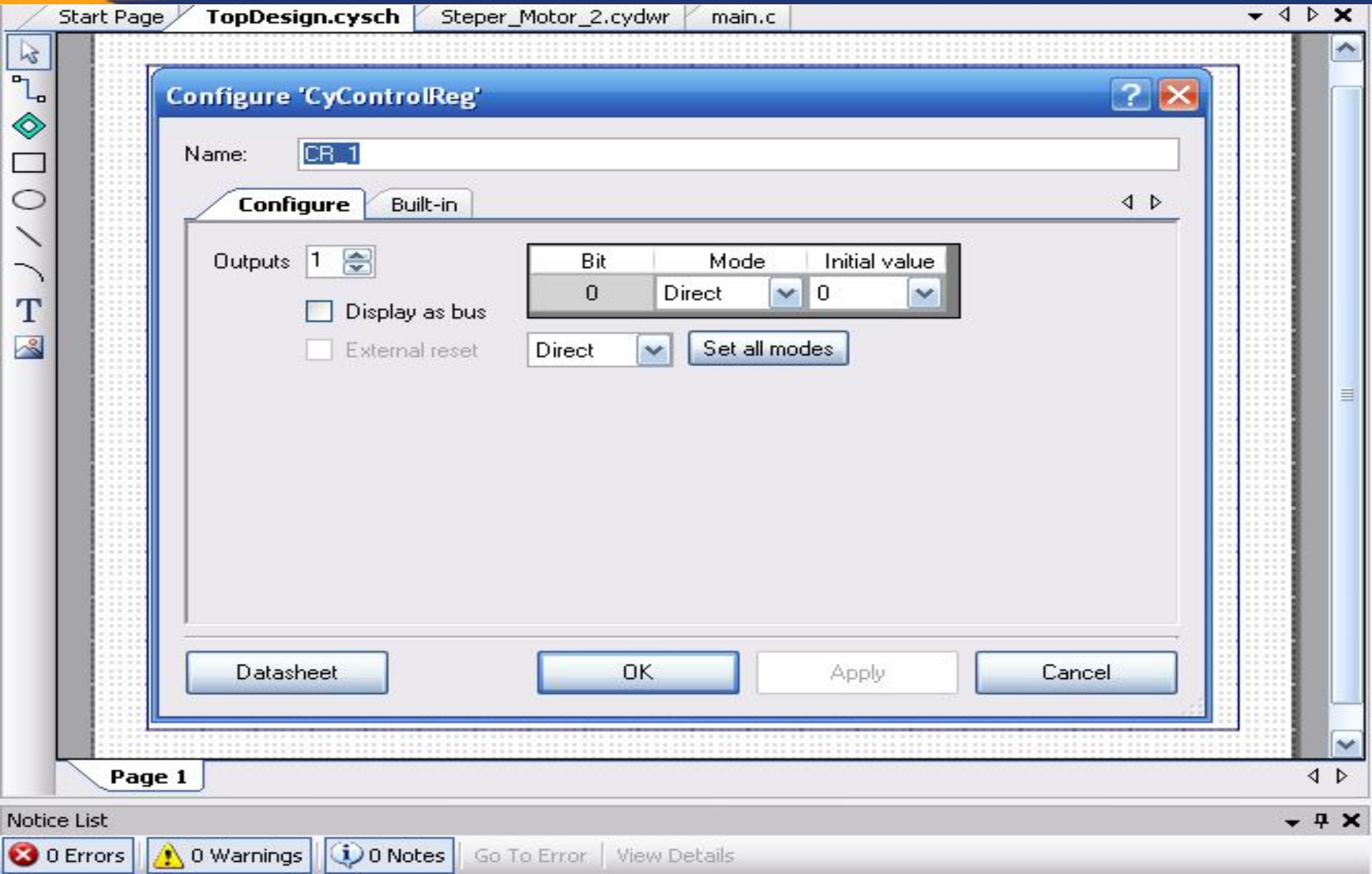
Lab_8 Stepper Motor



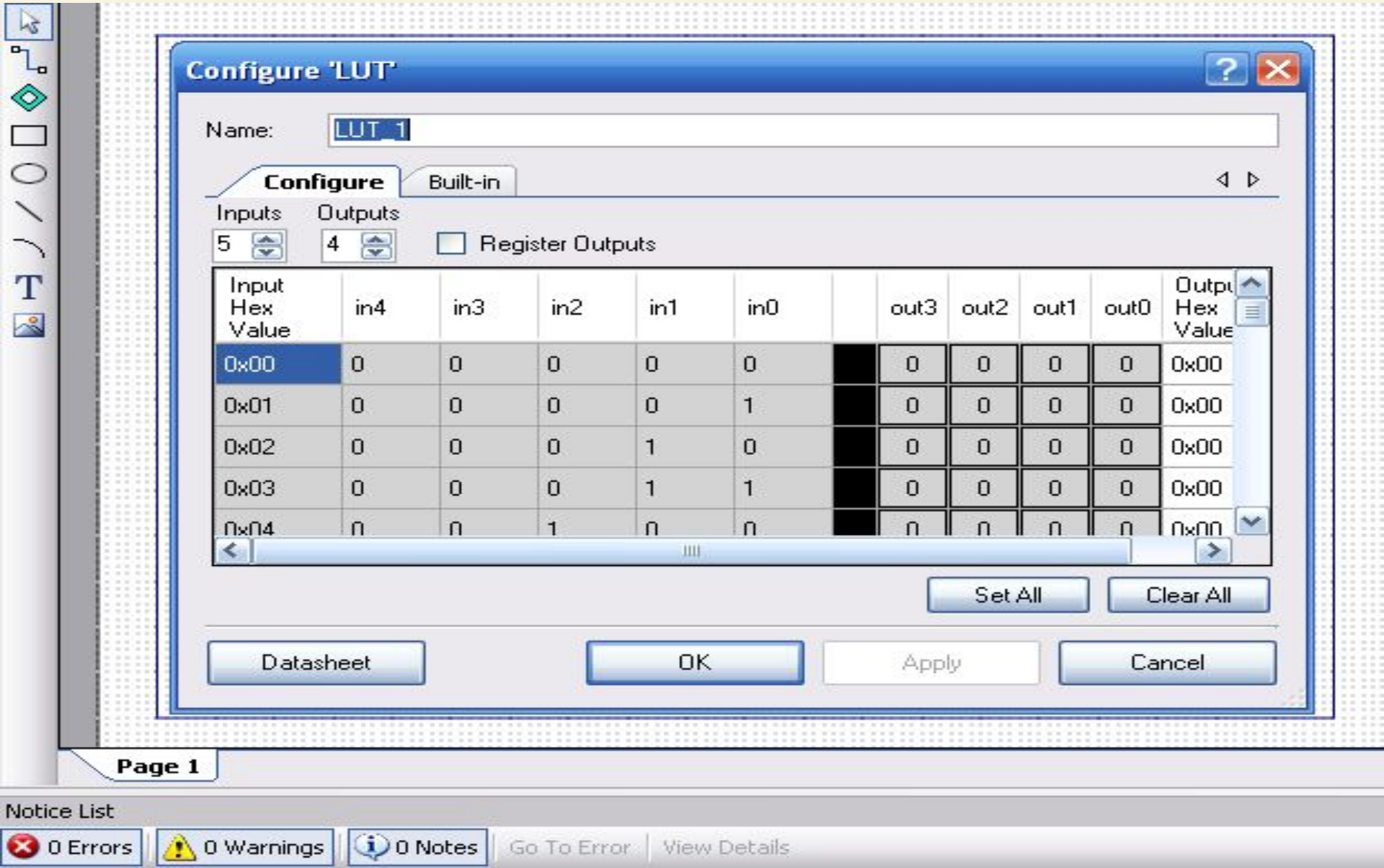
Lab_8 Stepper Motor



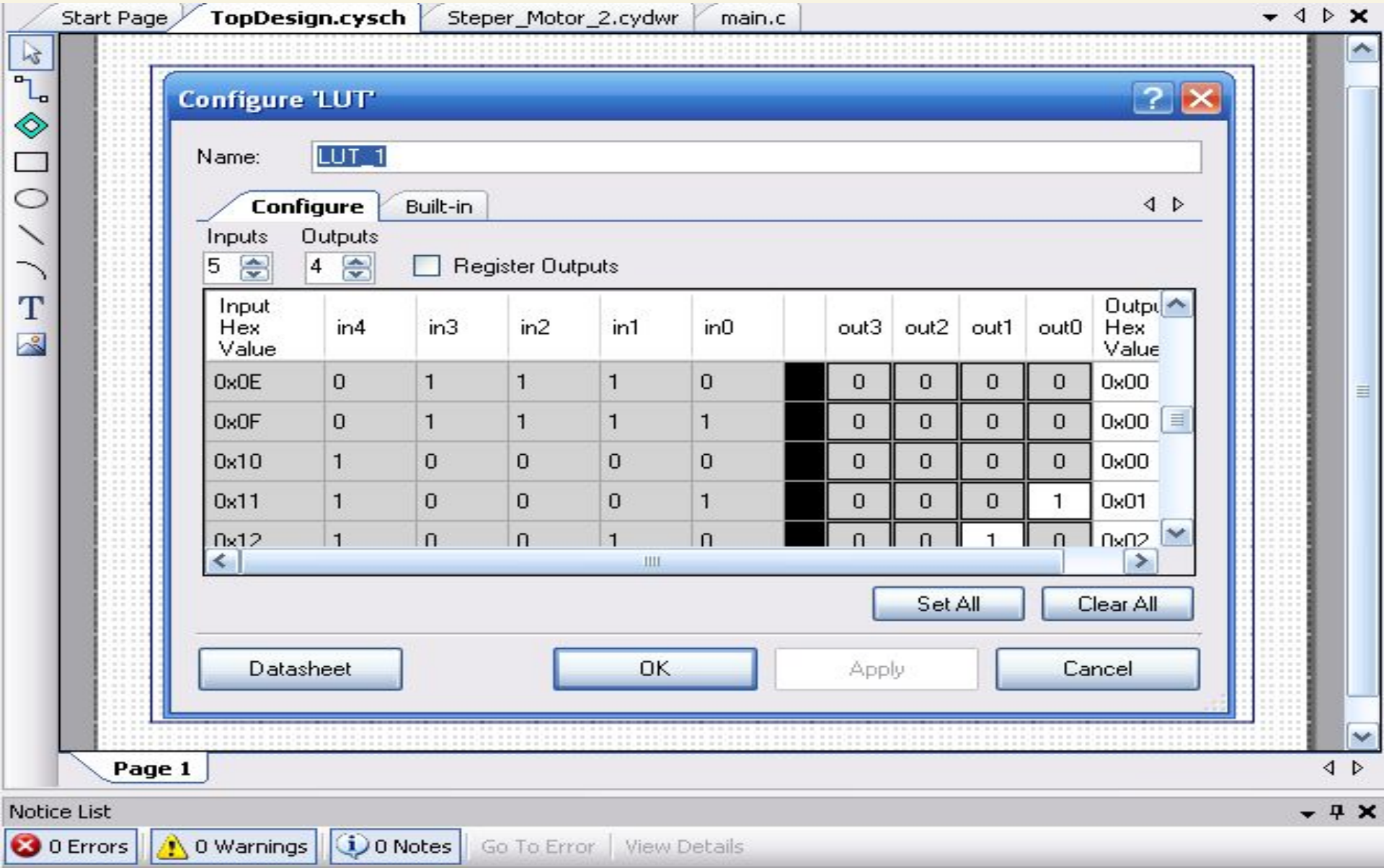
Lab_8 Stepper Motor



Configuration Lucup Table

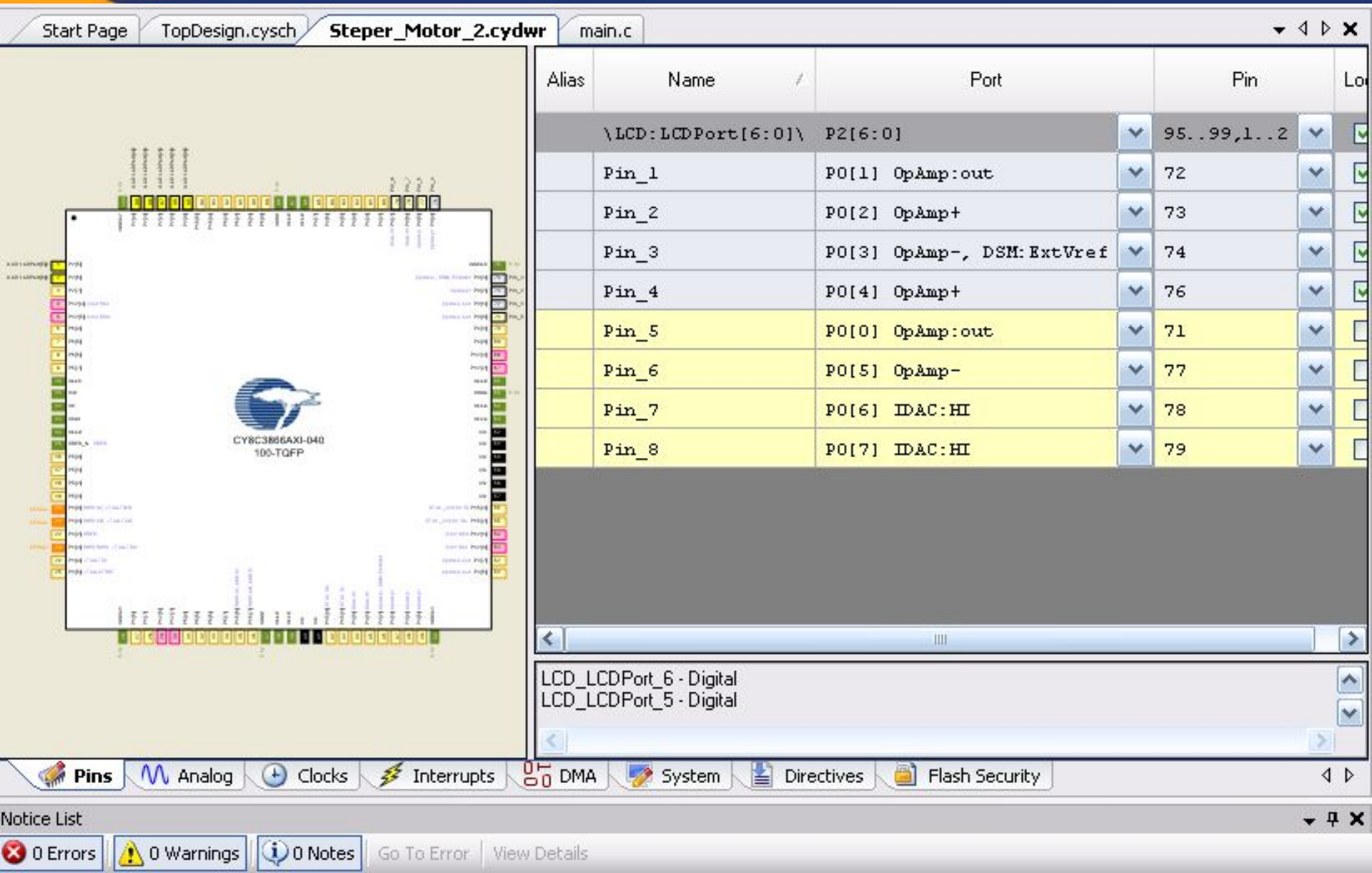


Configuration Lucup Table



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

Lab_8 Stepper Motor



Stepper Motor PSoC@3/5

Start Page | TopDesign.cysch | Steper_Motor_2.cydwr | *main.c

```
/* ========================================
 *
 * Copyright YOUR COMPANY, THE YEAR
 * All Rights Reserved
 * UNPUBLISHED, LICENSED SOFTWARE.
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION
 * WHICH IS THE PROPERTY OF your company.
 *
 * ========================================
*/


#include <device.h>
#include <stdlib.h>
#include <string.h>




int pin_pwm_1; int pin_pwm_2; int pin_pwm_3; int pin_pwm_4; // Піни PWM(1-4)
int b_1; int b_2; // Піни кнопок(1-2)
int delay_period; // Пауза періода після старта PWM
int N; // Кількість мікрорухів для повного оберту
int temp; int temp2; int temp3;
```

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);} // ліва вверх четвертинно
54
55      if ((way=='D')&&(period=='M')) {SIDE_2(N*1);} // ліва вниз четвертинно
56          }
57
58  }
59
60
```

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

```
50
51      if (motor=='L')
52          {
53      if ((way=='U')&&(period=='M')) {SIDE_1(N*1);} // ліва вверх четвертинно
54
55      if ((way=='D')&&(period=='M')) {SIDE_2(N*1);} // ліва вниз четвертинно
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

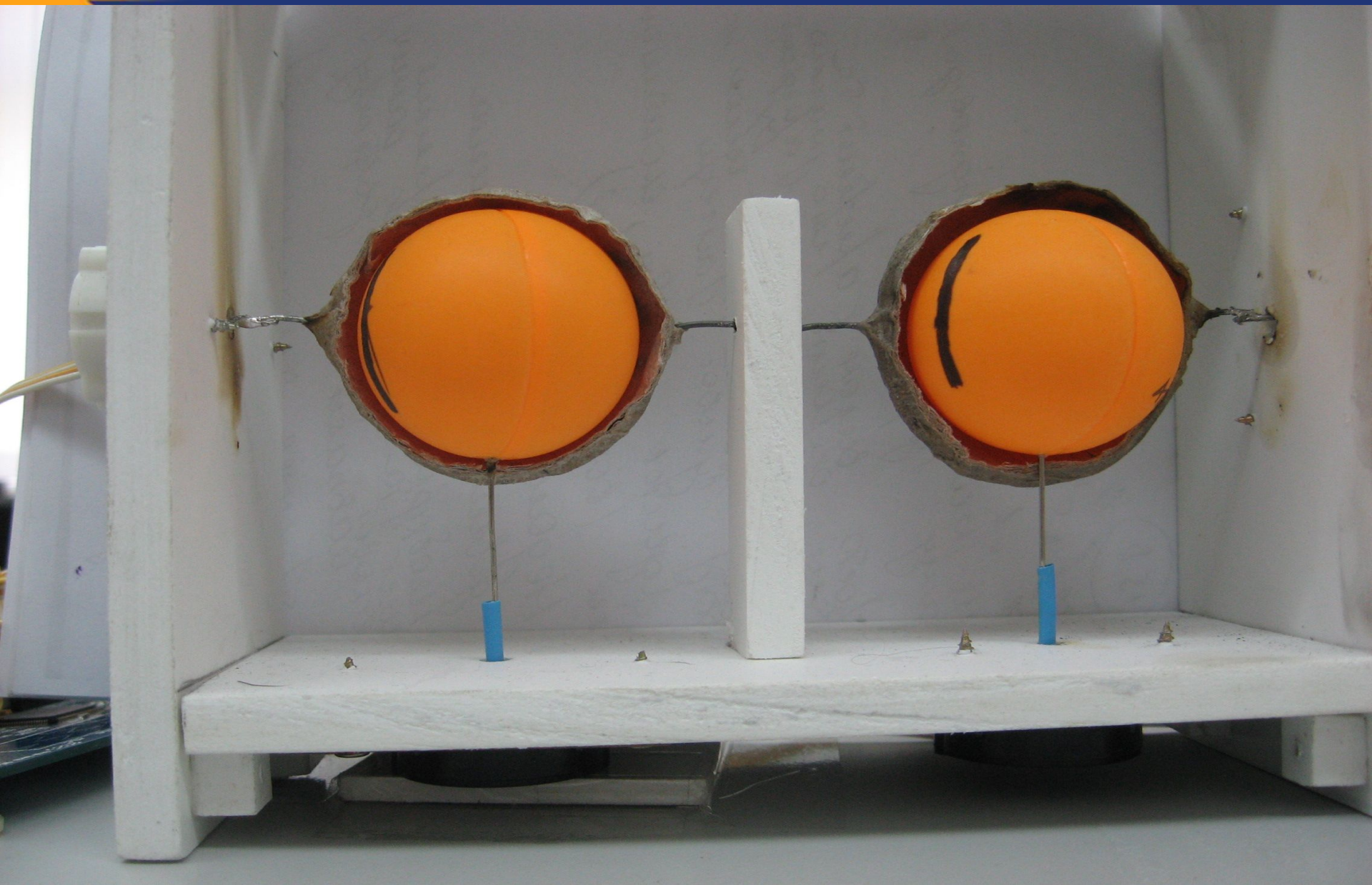
Show output from: All

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

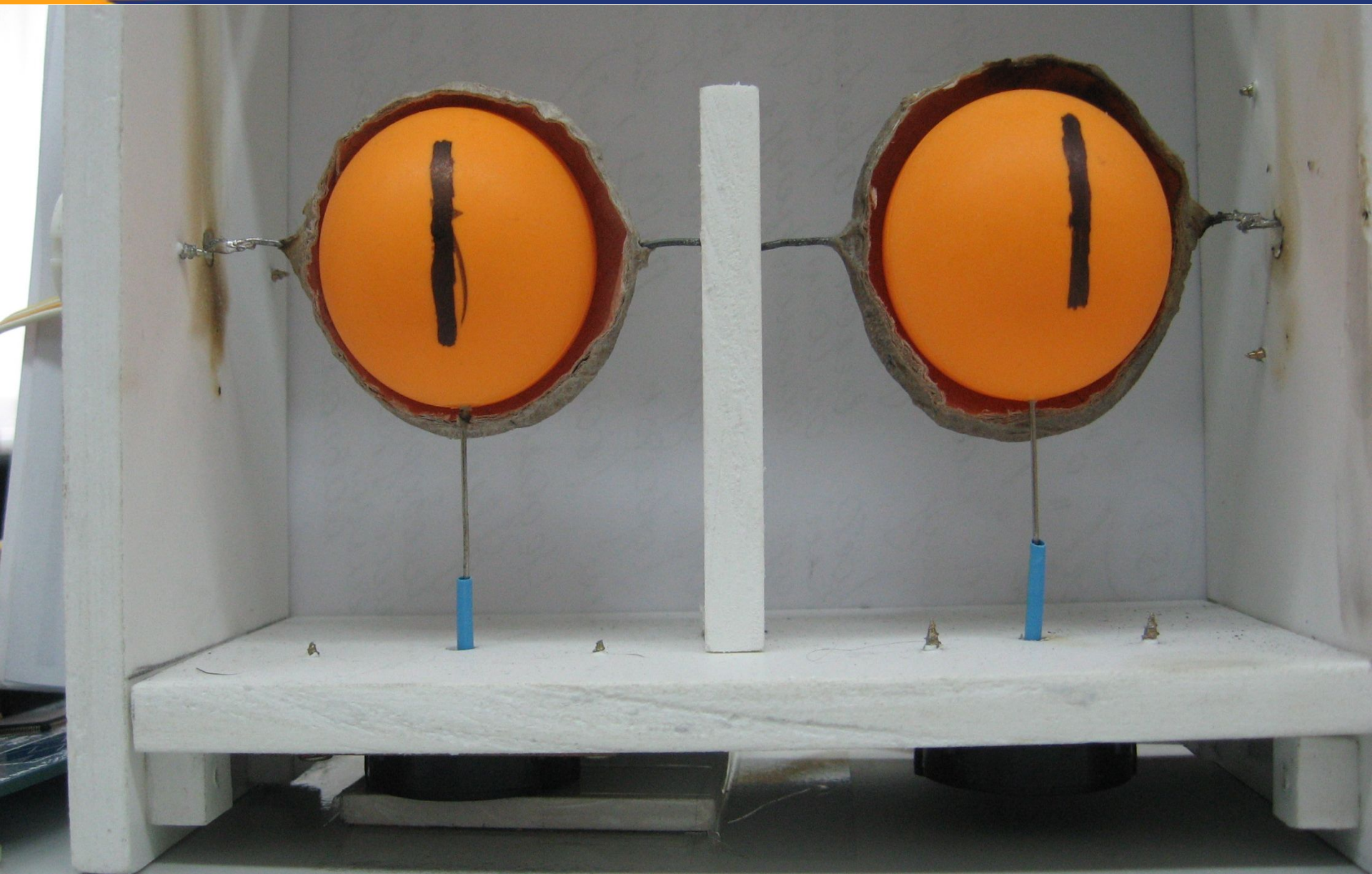
РОБОТ-АВТОМАТ НА ПЛАТФОРМІ PSoC@3 /5

Робот-автомат виконує встановлення та утримання в заданому положенні кута повороту роторів чотирьох мікро-потужних 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.

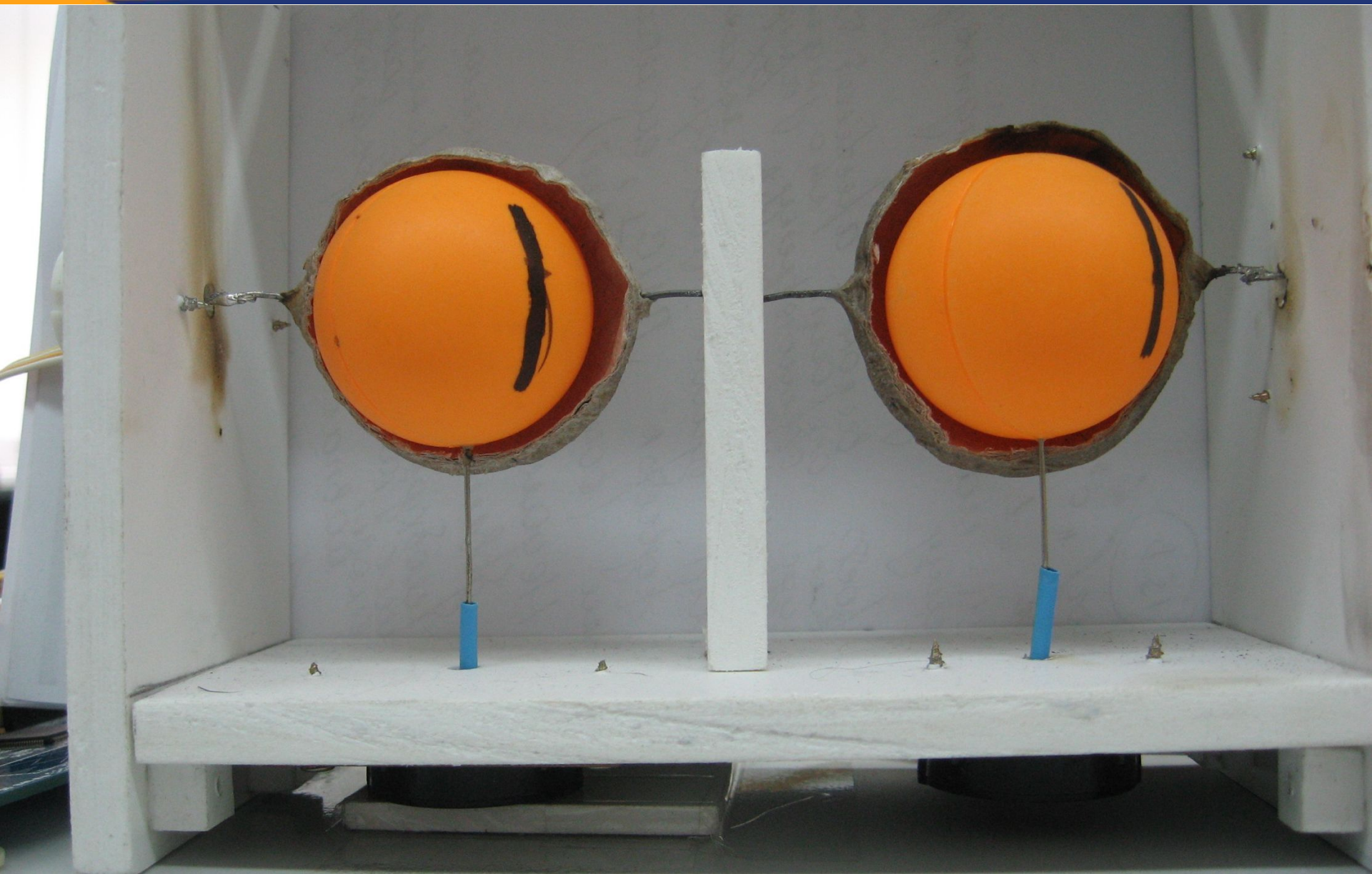
Stepper Motor PSoC@3/5



Stepper Motor PSoC@3/5



Stepper Motor PSoC@3/5



Stepper Motor PSoC@3/5

Stepper Motor PSoC@3/5

Рекомендована література :

***Основна*:**

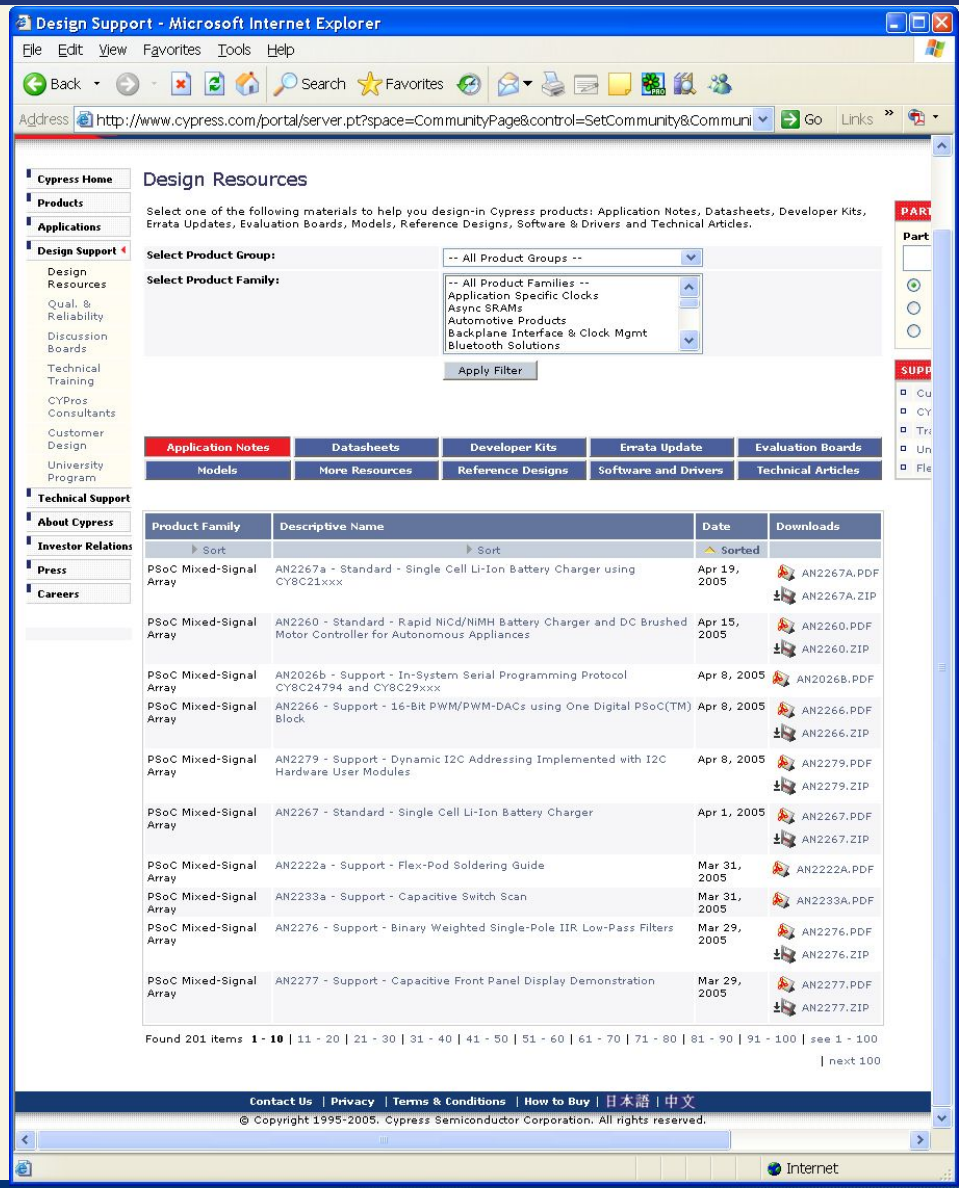
1. AN56551 PSoC® 3 Stepper Motor Control with Precision Microstepping. 2011 . - 15 с.
1. PSoC® 3 Architecture TRM (Technical Reference Manual) – 379 с.

***Додаткова*:**

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

Приклади застосування МК PSOC

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



Мікропроцесорна техніка

2019 р.

(лекція 8, кінець)

Благітко Б.Я.