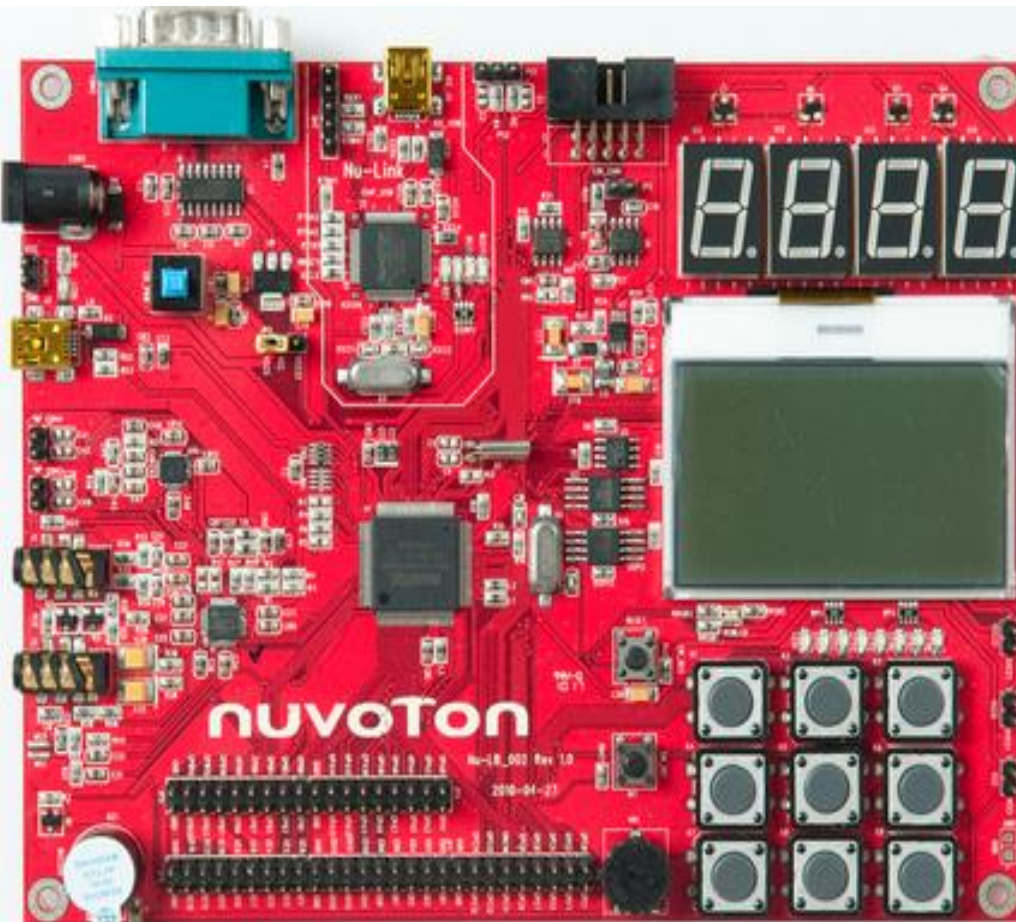


# PEMROGRAMAN MIKROKONTROLER



Menggunakan  
Nuvoton  
Learning Board  
NU-LB 002



OLEH: LUKMAN ROSYIDI

# PRAKTIKUM DASAR PEMROGRAMAN MIKROKONTROLER ARM

Revisi 1.0, Maret 2014.

Copyright © 2014 Lukman Rosyidi.

Ditulis oleh Lukman Rosyidi.

Didukung oleh Nuvoton Technology Corp., salah satu ARM partner.

Diterbitkan secara digital oleh Yayasan Prasimax Bina Teknologi, Depok.

Kontak: [informasi@mikron123.com](mailto:informasi@mikron123.com) | [www.mikron123.com](http://www.mikron123.com)

ARM® adalah trademark milik ARM Holding Limited.

NuMicro™ adalah trademark milik Nuvoton Technology Corporation.

CooCox adalah trademark milik Embest Technology Co., Ltd.

Hak Cipta dilindungi undang-undang.

Dilarang mengambil atau memodifikasi sebagian atau keseluruhan dari karya ini dalam bentuk apapun tanpa seijin penulis dan penerbit.

# PRAKTIKUM DASAR PEMROGRAMAN MIKROKONTROLER ARM

## **Tujuan:**

1. Memahami arsitektur dan fitur mikrokontroler ARM CORTEX M0 NuMicro NUC140
2. Dapat melakukan instalasi software development tools yang diperlukan untuk pemrograman mikrokontroler ARM
3. Dapat menggunakan software development tools mikrokontroler ARM untuk menulis, melakukan kompilasi, dan memasukkan program ke dalam chip mikrokontroler ARM
4. Dapat melakukan pemrograman mikrokontroler ARM untuk obyek input/output dasar

## **Prasyarat:**

Telah memahami dasar algoritma dan pemrograman terutama dengan bahasa C  
Pengetahuan tentang elektronika analog & digital akan sangat membantu

# DAFTAR ISI

PEMBAHASAN		HALAMAN
1	OVERVIEW ARSITEKTUR ARM	5
2	PENGENALAN LEARNING BOARD NUC140	10
3	INSTALASI SOFTWARE & HARDWARE	12
4	MEMULAI PROJECT PEMROGRAMAN	18
5	PEMROGRAMAN LED	39
6	PEMROGRAMAN PUSHBUTTON SWITCH	44
7	PEMROGRAMAN 7-SEGMENT DISPLAY	47
8	PEMROGRAMAN KEYPAD MATRIX	53
9	PEMROGRAMAN BUZZER	59
10	PEMROGRAMAN LCD MATRIX DISPLAY	62
11	PEMROGRAMAN ANALOG VARIABLE RESISTANCE	69
12	REFERENSI	72

# 1. OVERVIEW ARSITEKTUR ARM

Mikrokontroler adalah mikrokomputer kecil dalam satu chip, di dalamnya berisi prosesor, memori dan peripheral I/O yang bisa diprogram.

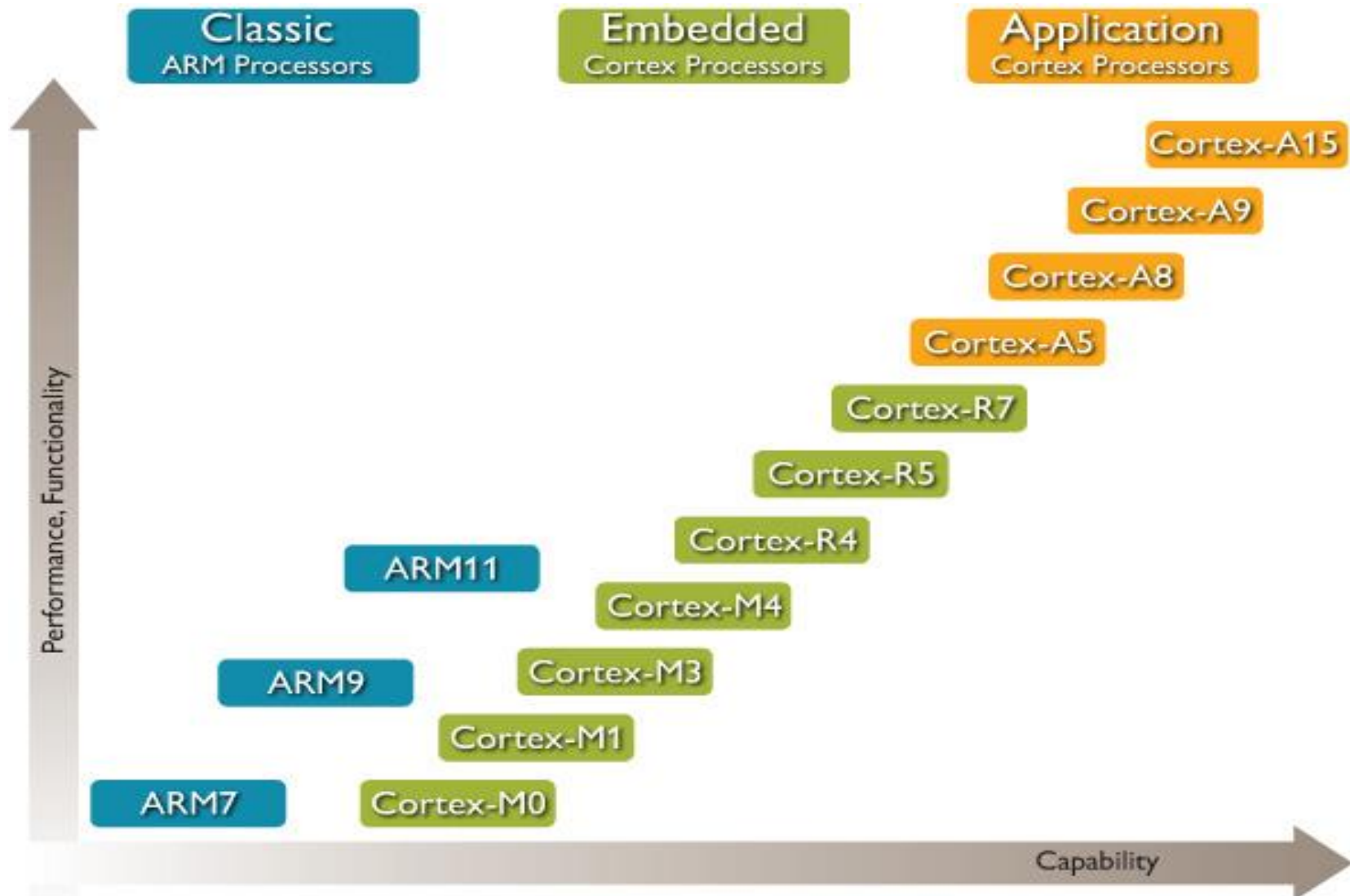
ARM adalah arsitektur prosesor 32 bit yang dibuat oleh ARM Holding dan dilisensikan untuk diproduksi oleh berbagai vendor di dunia termasuk AMD, Atmel, Freescale, Nuvoton, Nvidia, NXP, Samsung, ST Micro, dan TI.

Prosesor ARM digunakan pada perangkat smartphone, tablet, dan embedded system. Kini ARM menjadi arsitektur prosesor yang paling banyak diproduksi di dunia.

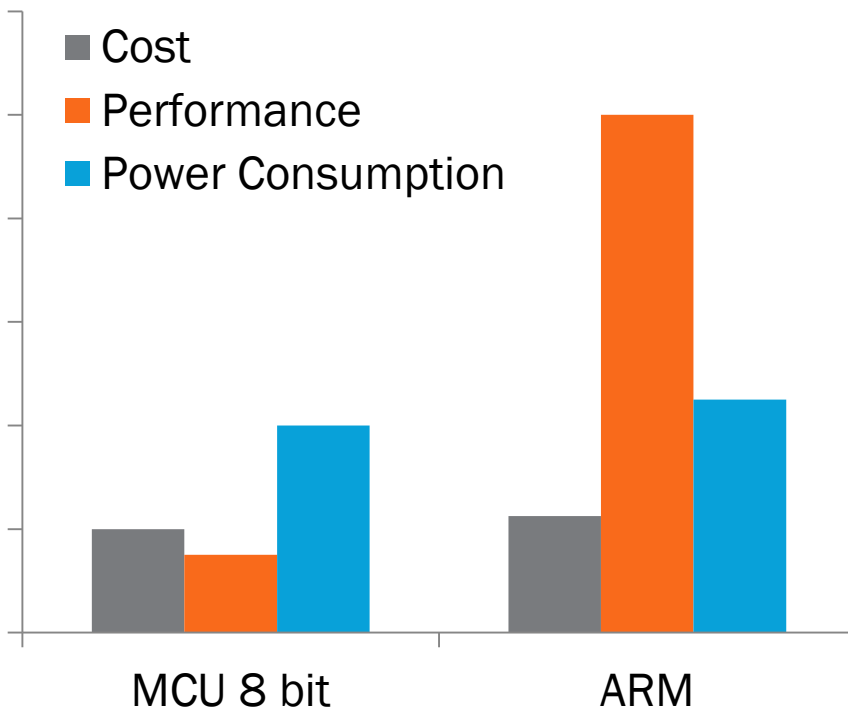
Seri arsitektur ARM terbaru terdiri dari 3 lini kelas penggunaan yaitu:

1. ARM CORTEX A untuk prosesor aplikasi
2. ARM CORTEX R untuk prosesor real time
3. ARM CORTEX M untuk prosesor mikrokontroler

# LINI ARSITEKTUR PROSESOR ARM



# APA KELEBIHAN MIKROKONTROLER ARM ?



Dengan harga yang relatif sama, prosesor ARM memberikan kinerja jauh lebih tinggi dengan konsumsi daya yang rendah.

## Belajar ARM berarti ...

### 1. Learn for the most used architecture

Belajar arsitektur prosesor yang paling banyak dikembangkan & didukung industri saat ini

### 2. Learn for the latest technology

Belajar untuk mempersiapkan diri dengan teknologi dan tools yang terkini dan menjadikannya sebagai suatu keunggulan

### 3. Learn one for all lines

Belajar prosesor dengan arsitektur yang memiliki keuntungan dalam hal kesamaan yang luas dari lini prosesor untuk berbagai kelas penggunaan, dari kelas mikrokontroler hingga prosesor aplikasi

### 4. Learn with ease of support

Belajar dengan banyak dukungan dari pembuat ARM dan para vendor/partner, serta dukungan komunitas yang luas



# NUVOTON MIKROKONTROLER ARM CORTEX M0 NUC140

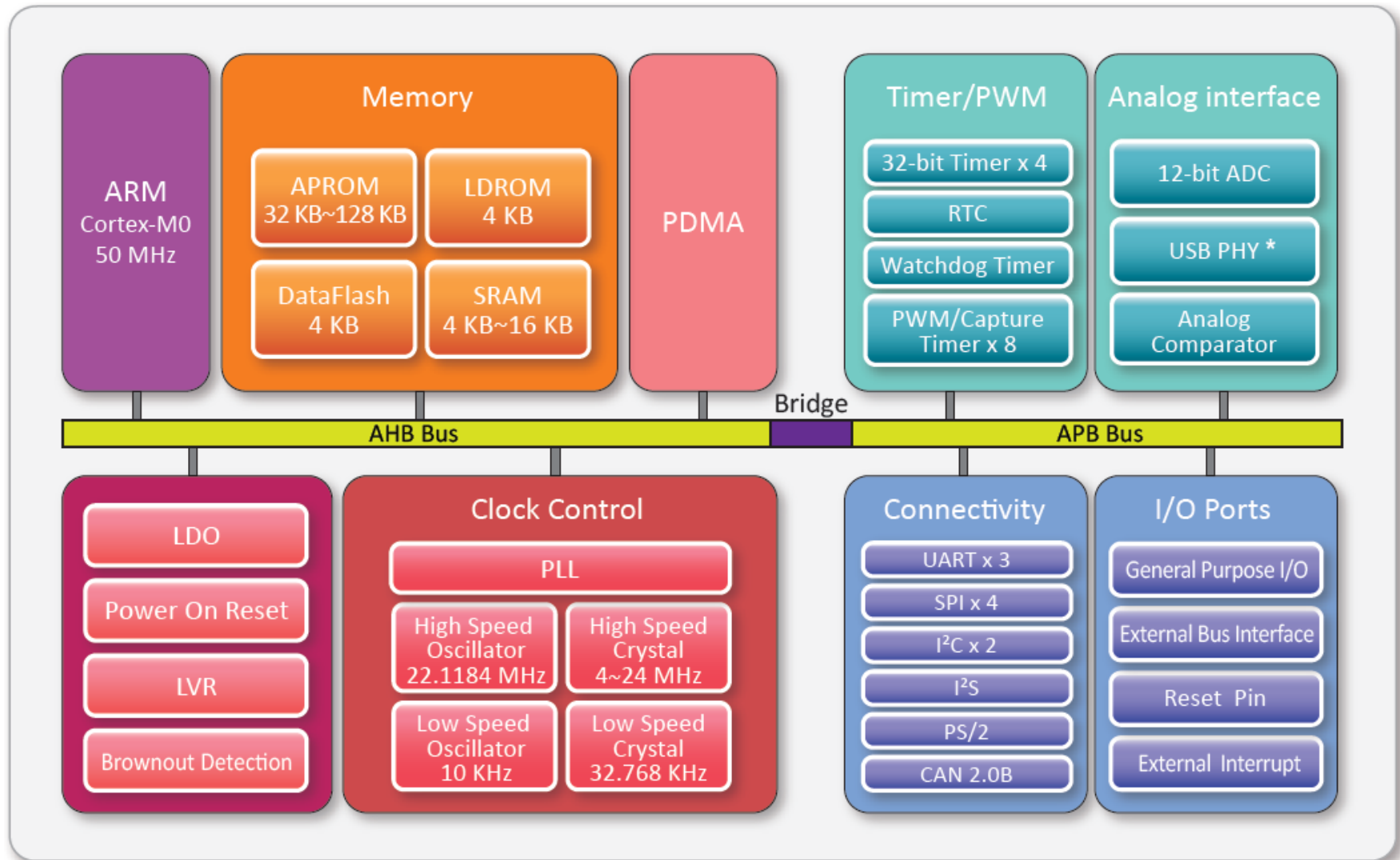


## Spesifikasi:

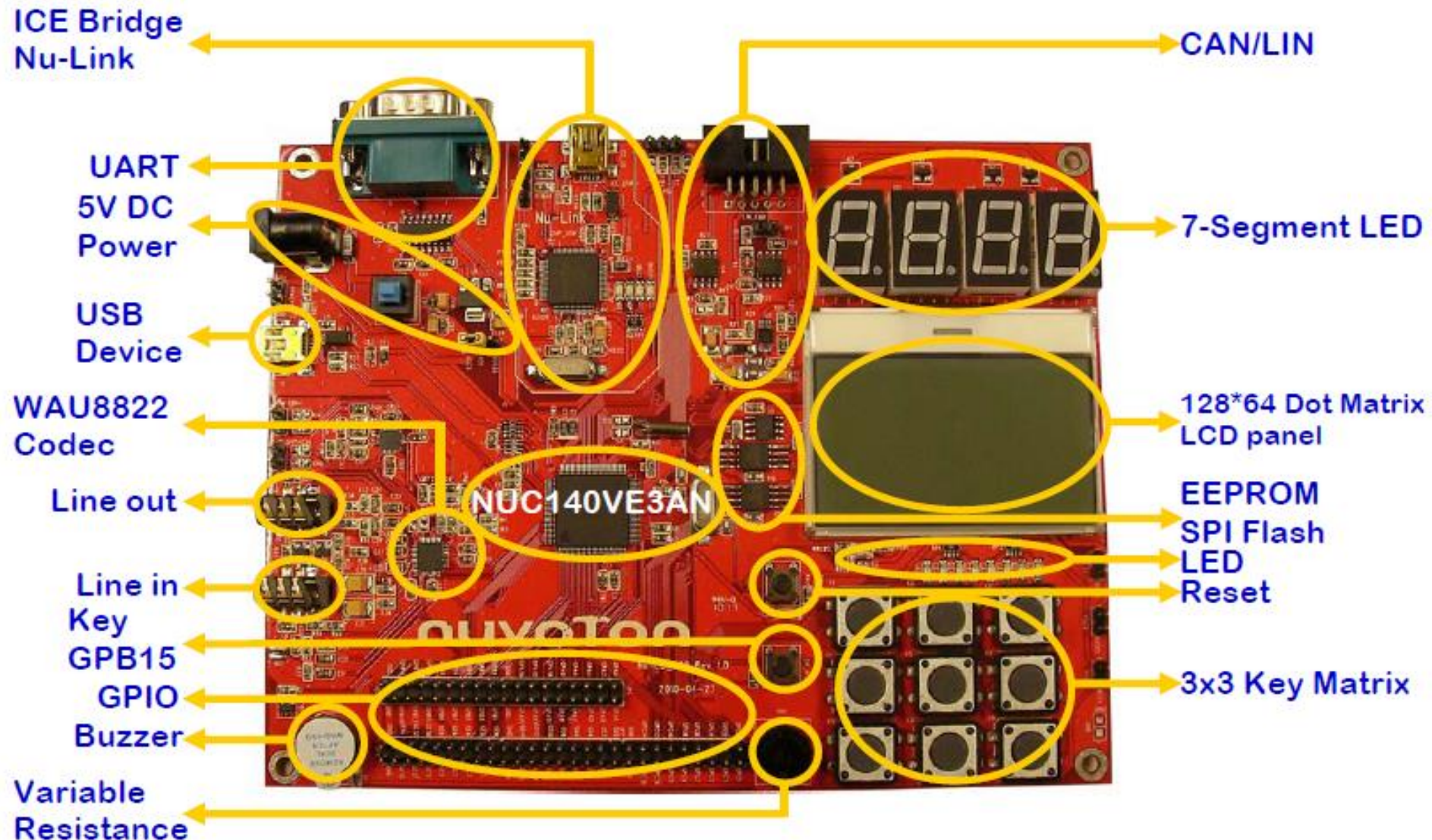
- NuMicro ARM Cortex M0
- Internal clock 22 MHz
- 16KB RAM
- 128KB memori program (APROM)
- 4KB memori data (DataFlash)
- 4KB memori bootloader (LDROM)
- 80 pin General Purpose I/O (GPIO)
- 8 channel ADC 12 bit, 2 analog comparator
- 8 PWM, 4 Timer 32 bit, Real Time Clock
- 3 serial UART mendukung IrDA, LIN, RS485
- 4 SPI, 2 I2C, 1 USB, 2 LIN, 1 CAN, 1 I2S
- Programmable via USB



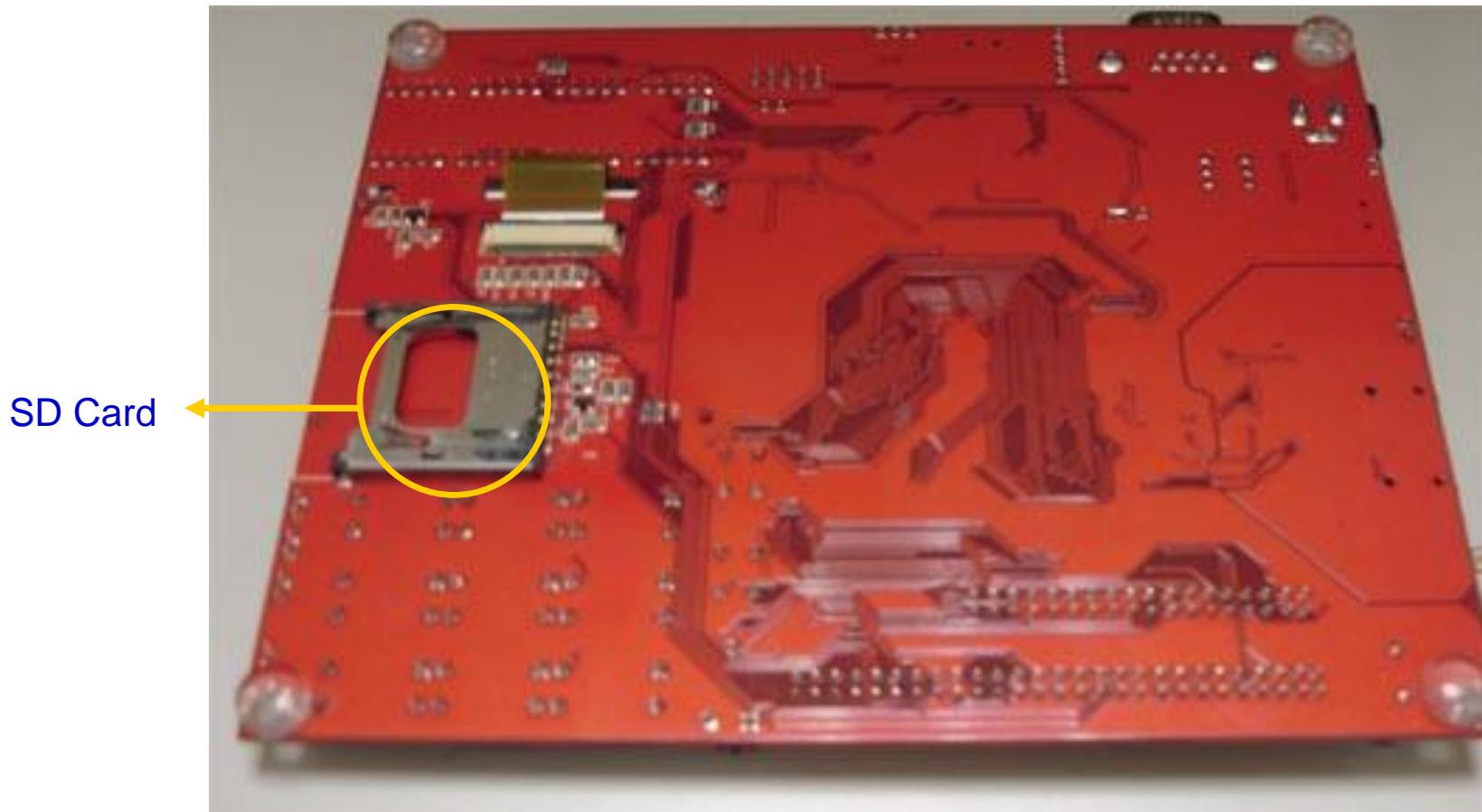
# BLOK DIAGRAM NUMICRO ARM CORTEX M0 NUC140



## 2. PENGENALAN LEARNING BOARD NUC140



# PENGENALAN LEARNING BOARD NUC140



### 3. INSTALASI SOFTWARE & HARDWARE

Software development di lingkungan Windows XP/7:

1. Library Board Support Package : **NUC100SeriesBSP\_CMSIS**  
Download <http://www.nuvoton.com>  
Software downloader ini free, disediakan oleh Nuvoton.
2. Software Compiler: **GCC ARM**  
Download <https://launchpad.net/gcc-arm-embedded>  
Software compiler ini free, tidak ada pembatasan kode atau fitur.
3. Software Editor: **CooCox CoIDE**  
Download <http://www.CooCox.com>  
Software editor ini free, tanpa pembatasan waktu atau fitur.
4. Software Pin Configurator: **CooCox CoSmart**  
Download <http://www.CooCox.com>  
Software configurator ini free, tanpa pembatasan waktu atau fitur.

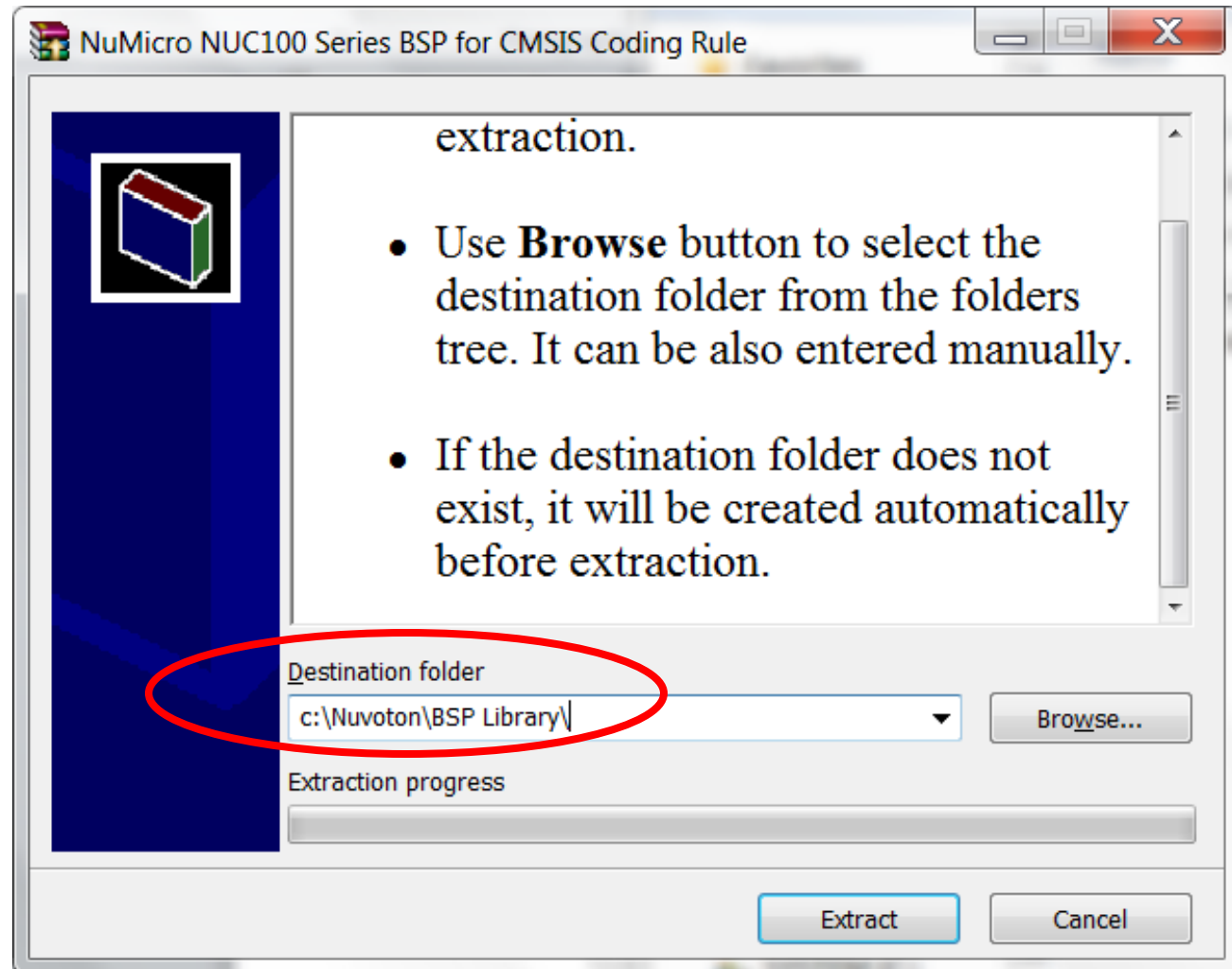
Semua software di atas sudah ada dalam CD yang disediakan PRASIMAX MIKRON.



## Langkah 1:

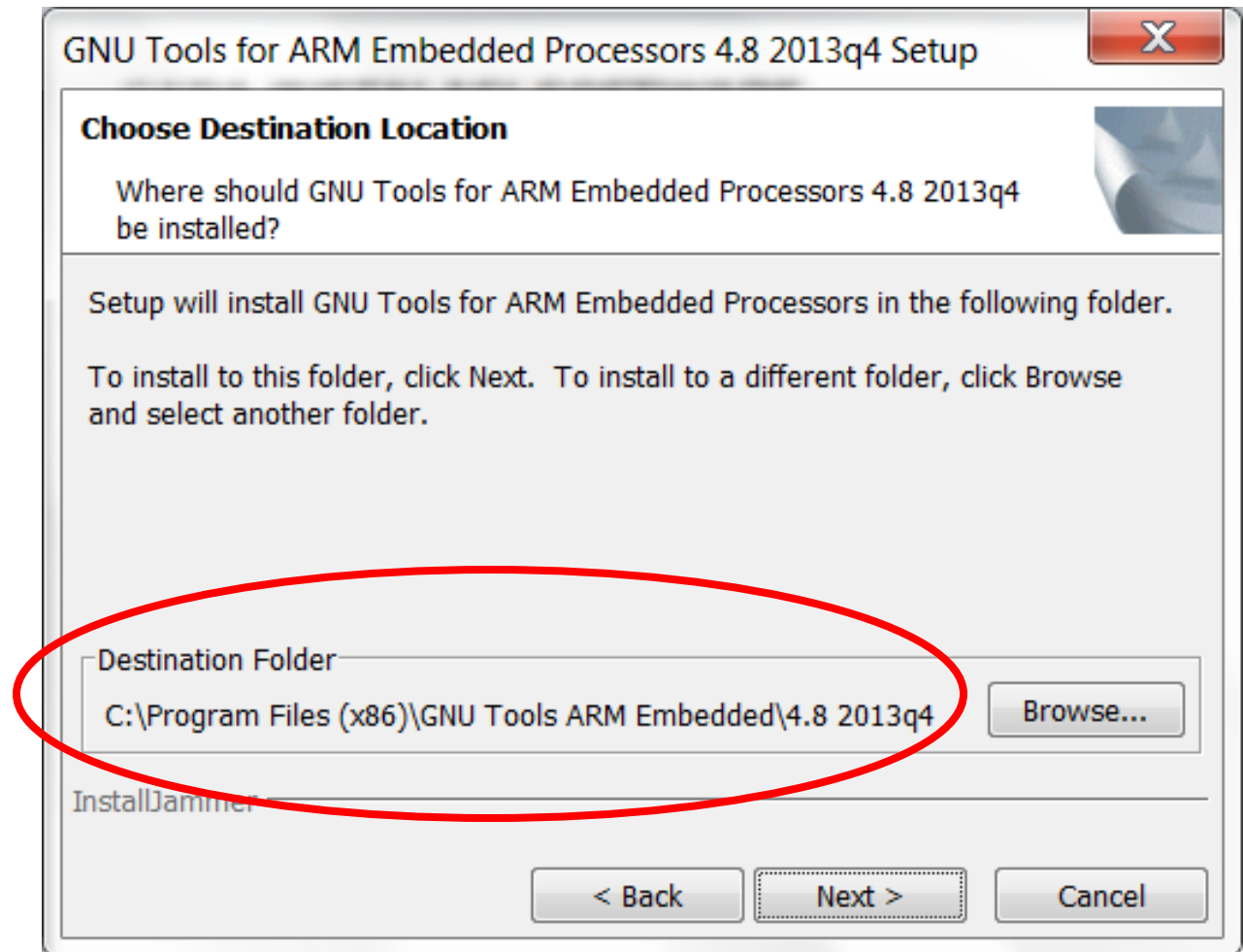
Lakukan instalasi  
NUC100SeriesBSP  
\_CMSIS.exe.

Perhatikan path  
destination folder.  
Defaultnya adalah  
C:\Nuvoton\BSP  
Library\



## Langkah 2:

Lakukan instalasi  
Software Compiler  
GCC ARM.  
Perhatikan lokasi  
path folder tempat  
instalasi software.

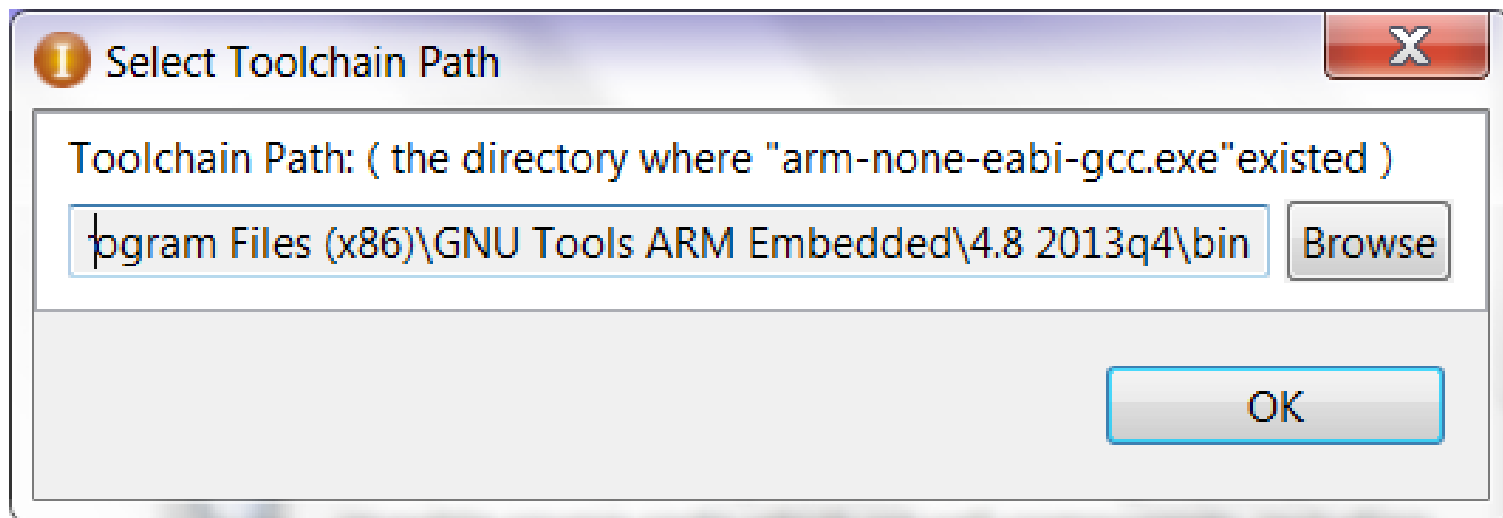


### Langkah 3:

Lakukan instalasi Software Editor ColIDE.

Sesudah selesai, panggil software ColIDE, lakukan setting compiler yang digunakan melalui menu **Project > Select Toolchain Path**.

**Browse** ke folder tempat instalasi software GCC ARM pada sub folder bin.



Tutup ColIDE jika sudah selesai.

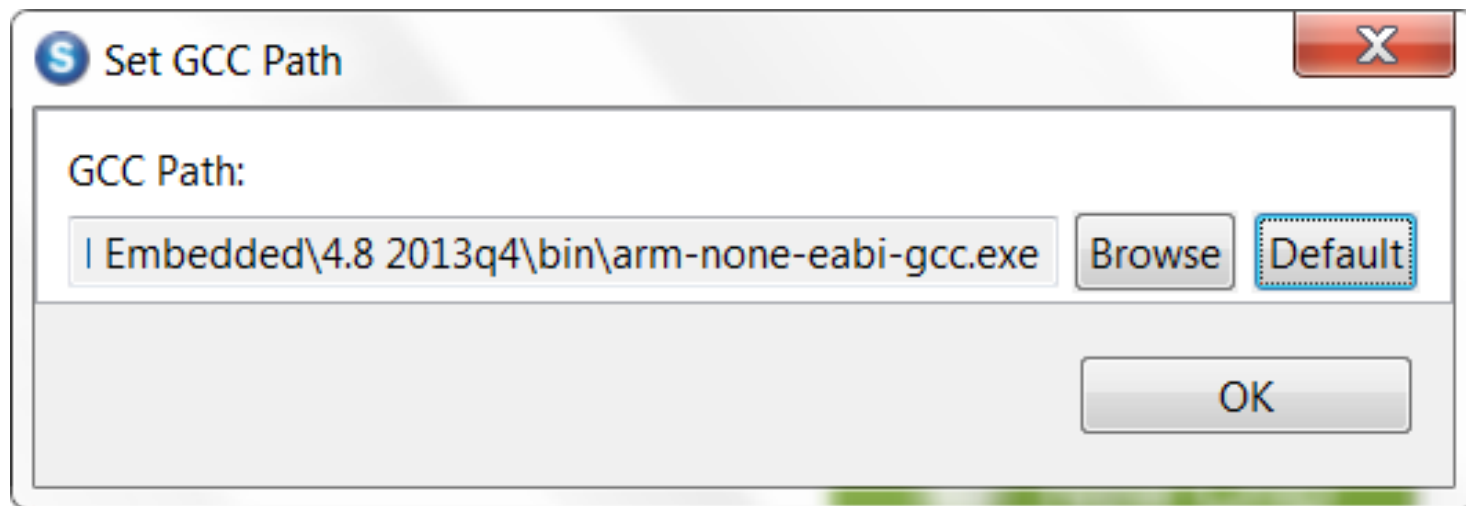


## Langkah 4:

Lakukan instalasi Software Pin Configurator CooCox CoSmart.

Sesudah selesai, panggil software CoSmart, lakukan setting compiler yang digunakan melalui menu **Project > Config GCC Path**.

**Browse** ke lokasi file **arm-none-eabi-gcc.exe** di folder tempat instalasi software GCC ARM pada sub folder bin lalu klik OK.

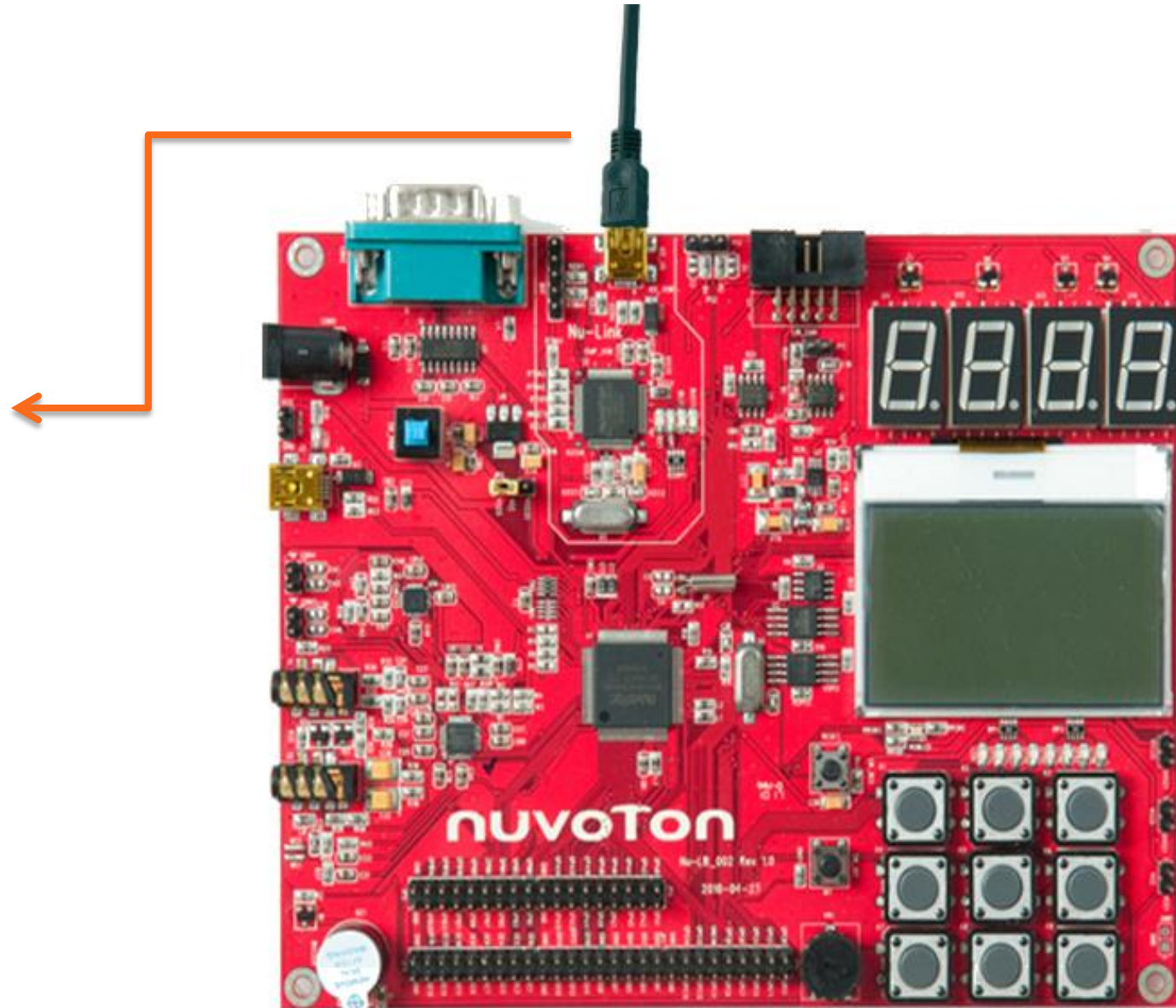


## Langkah 5:

Hubungkan **Kabel USB** ke jalur USB Nu-Link di board sebelah atas.

Lalu hubungkan kabel USB ke komputer.

Board akan mendapat power dari USB. Pastikan LED merah indikator power menyala.



## 4. MEMULAI PROJECT PEMROGRAMAN

Buat daftar koneksi pin yang akan digunakan berdasar skema Learning Board

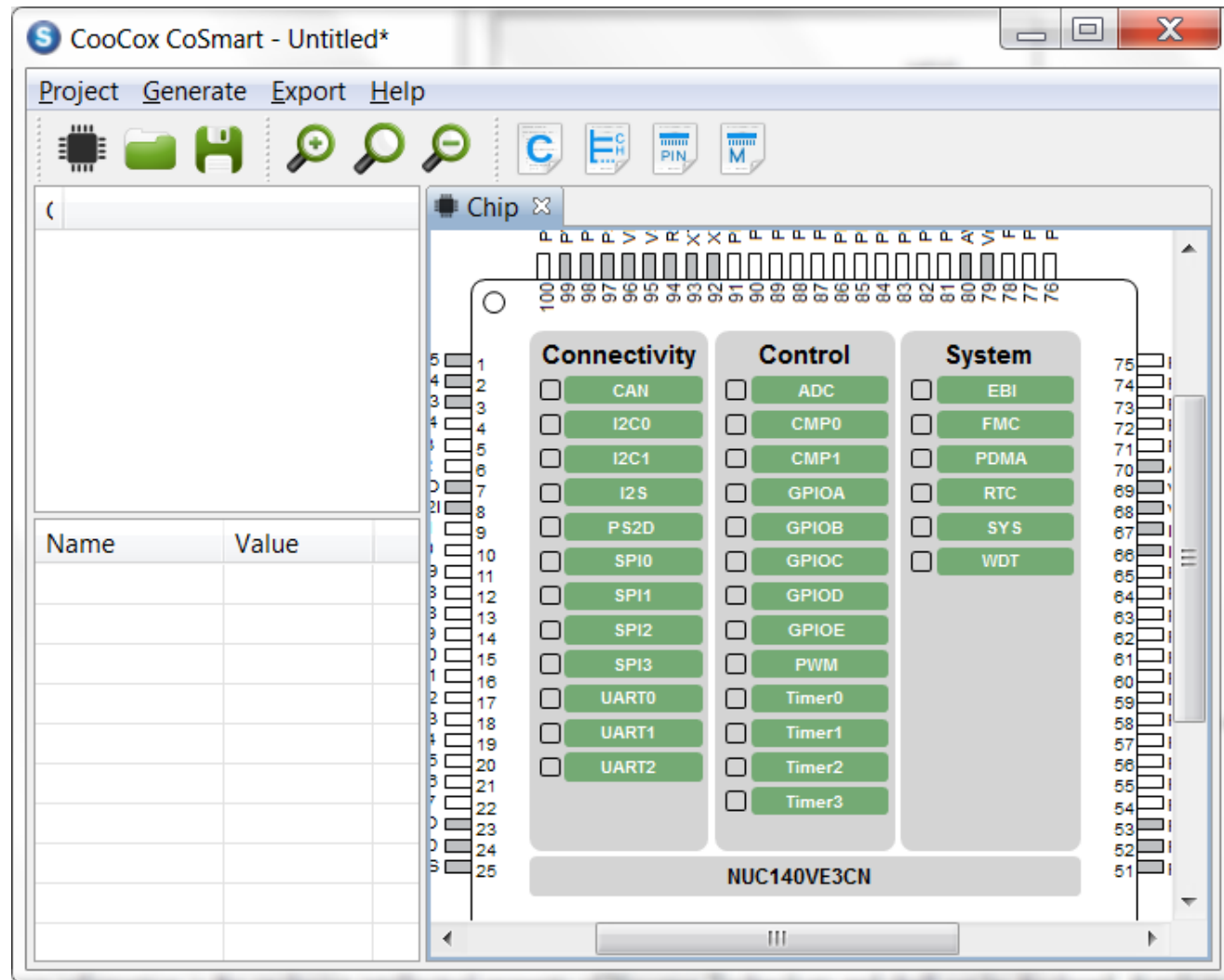
1. LED
2. 7-Segment
3. Pushbutton
4. Keypad Matrix
5. Variable Resistor
6. Buzzer
7. LCD Matrix

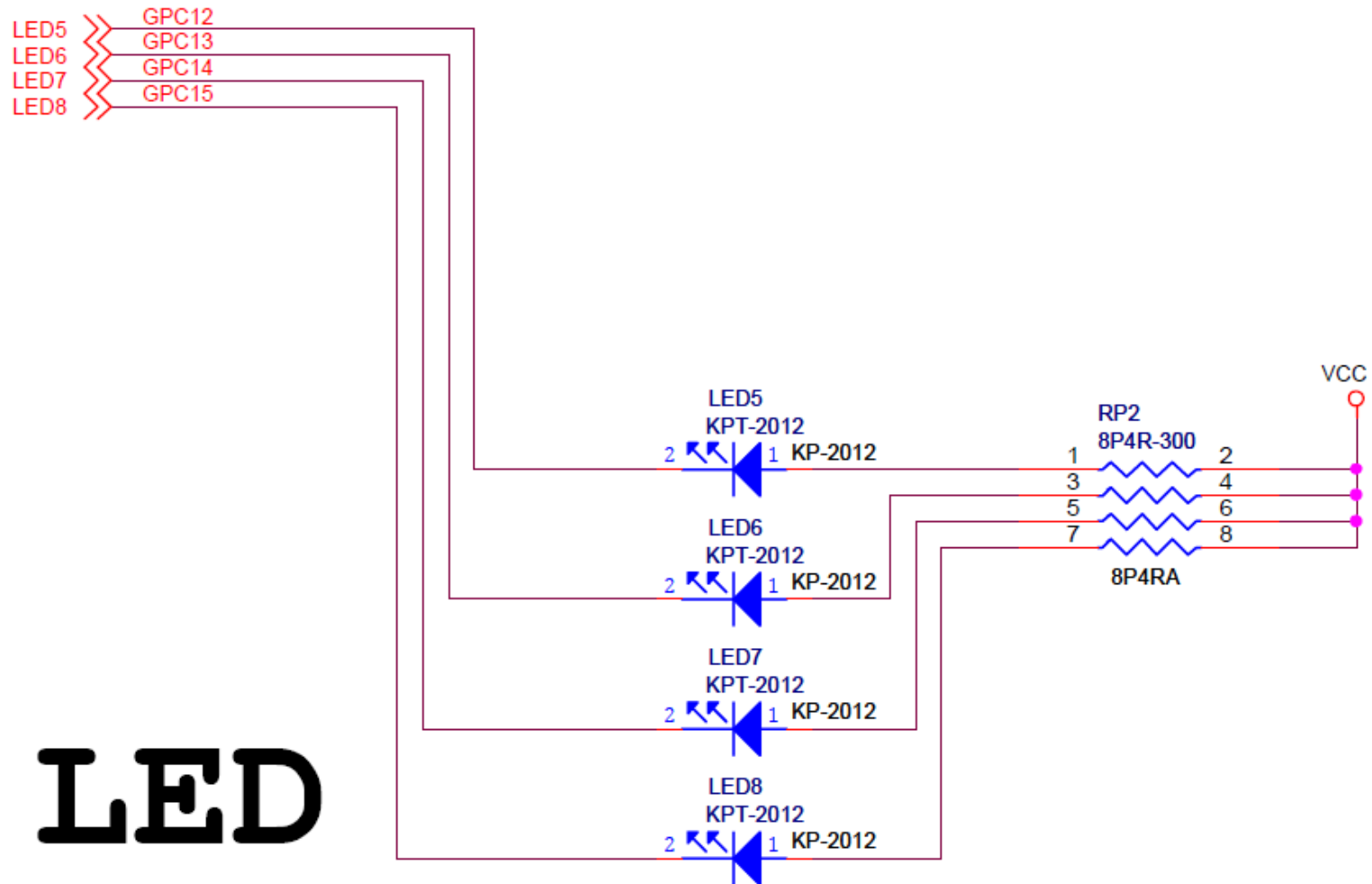
# DAFTAR KONEKSI PIN OBYEK I/O

OBYEK	KONEKSI DI BOARD	PIN	KONFIGURASI
LED	LED5 s/d LED8	GPC12 s/d GPC15	GPIO Output
7-Segment	7-Seg Data	GPE0 s/d GPE7	GPIO Output
	7-Seg Select	GPC4 s/d GPC7	GPIO Output
Pushbutton	SW_INT	GPB15	GPIO Input
Keypad Matrix	Key Column 1 s/d 3	GPA0 s/d GPA2	GPIO Bidirectional
	Key Row 1 s/d 3	GPA3 s/d GPA5	GPIO Bidirectional
Variable Resistor	VR1	GPA7 / ADC7	ADC
Buzzer	BZ1	GPB11	GPIO Output
LCD Matrix	LCD SPI CS LCD SPI CLK LCD Reset LCD SPI MOSI	GPD8 / SPISS3 GPD9 / SPICLK3 GPD10 GPD11 / SPIMOSI3	SPI
	LCD Backlight	GPD14	GPIO Output

## Langkah 1: Pilih Chip

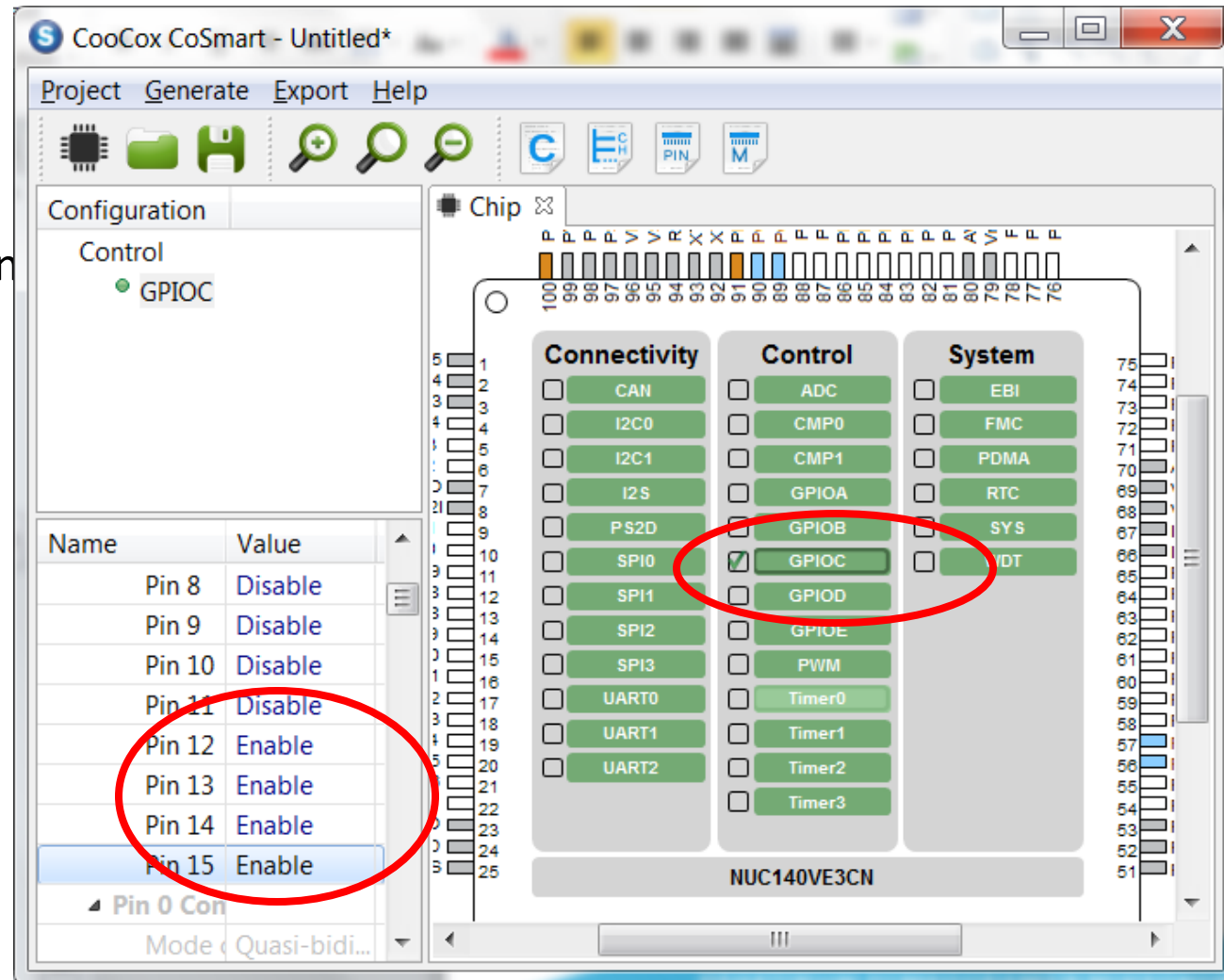
1. Panggil aplikasi **CooCox Software > CoSmart** dari Start Menu untuk membantu konfigurasi pin
2. Klik tombol **New Chip** di tengah layar.
3. Pilih **Nuvoton**, lalu pilih **NUC140VE3CN**
4. Akan muncul gambar chip NUC140VE3CN, dengan semua peripheralnya.





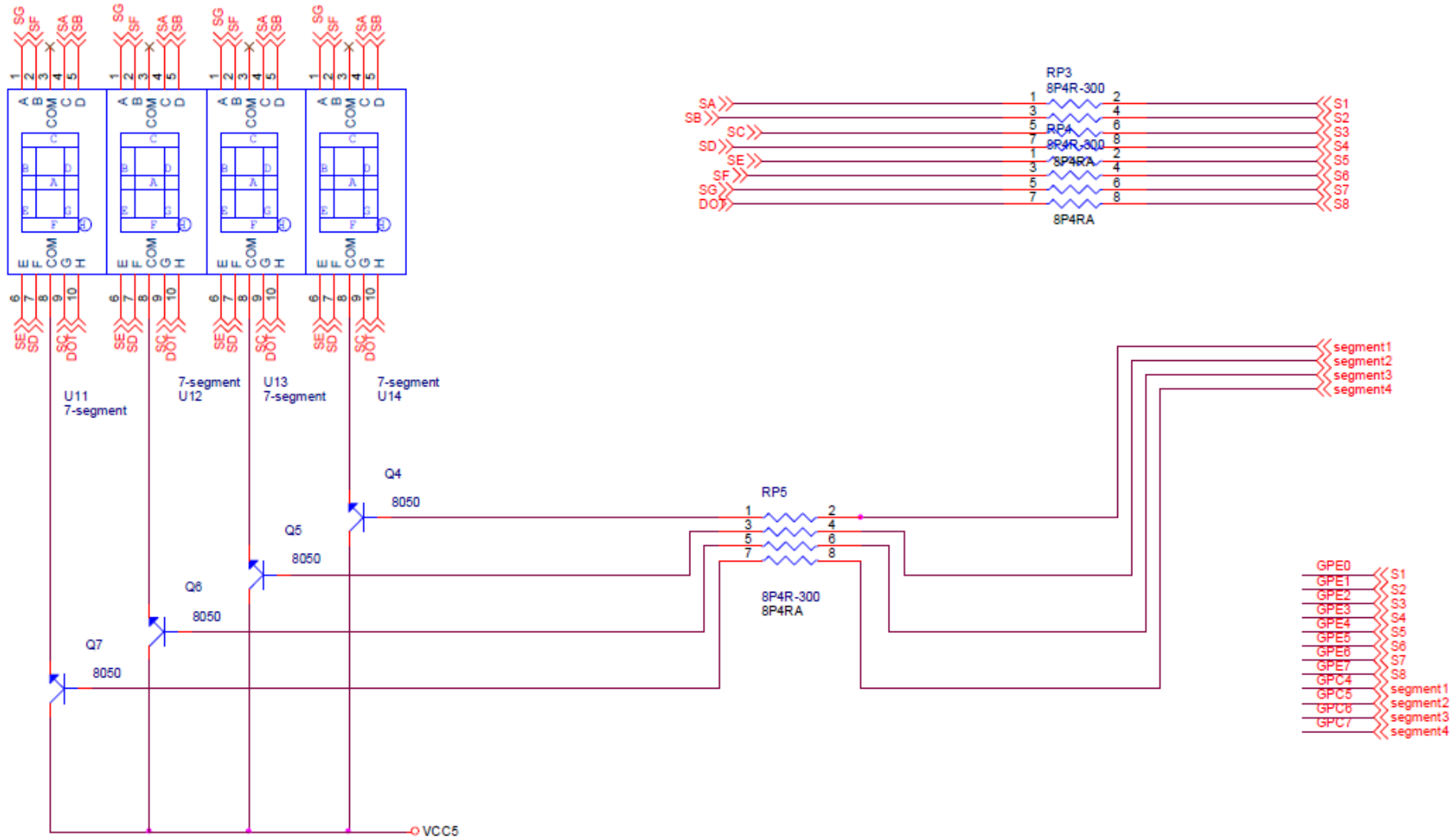
## Langkah 2: Set Pin LED

1. Contreng **GPIOC** pada gambar chip.
2. Di jendela Configuration pilih GPIOC
3. Di bawahnya, klik pada **Pin 12 s/d 15** agar **Enable**
4. Scroll ke bawah, atur Pin Config 12 s/d 15:
  - Mode Control: Output





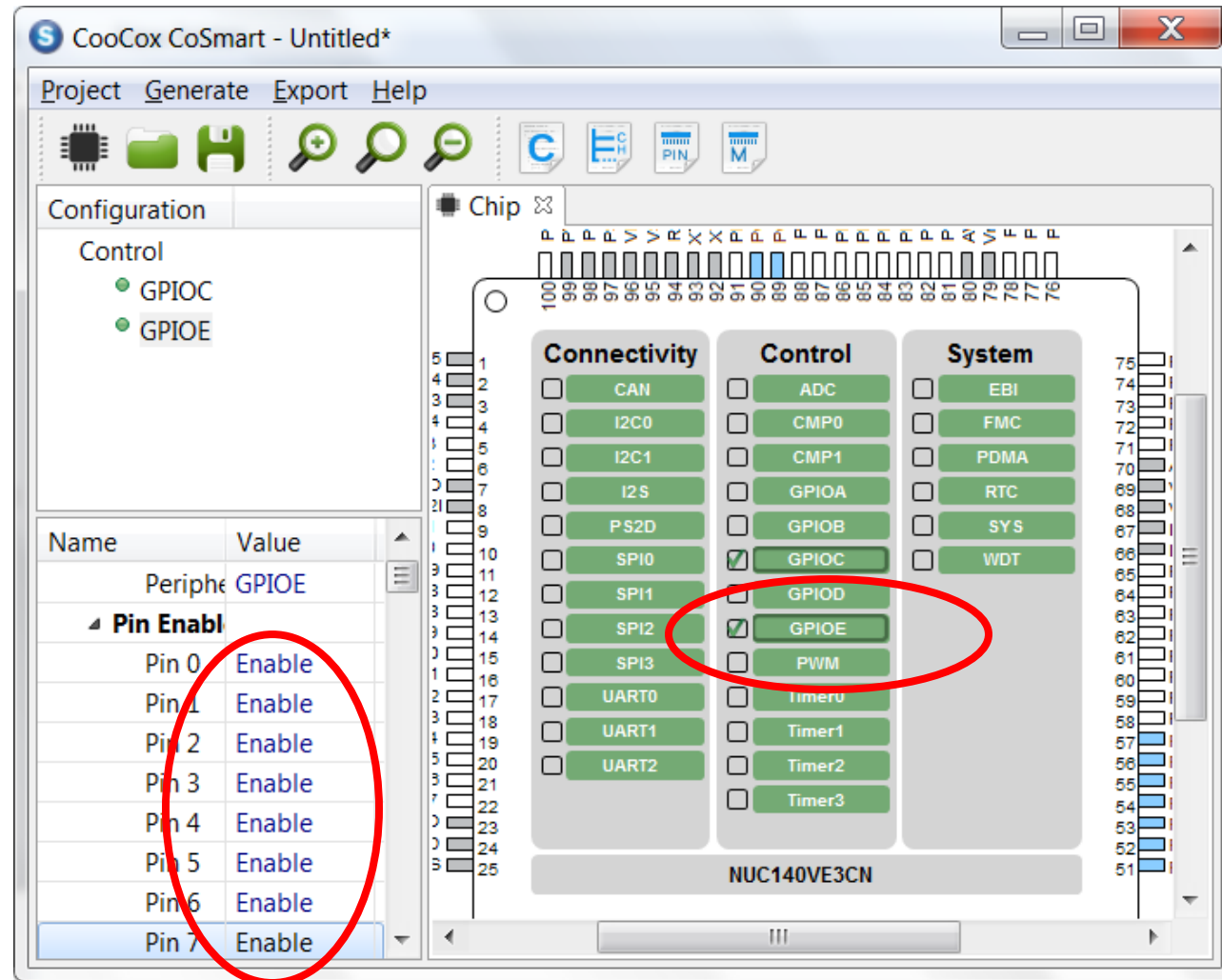
# 7-SEGMENT



## Langkah 3a.

### Set Pin 7-Segment Data

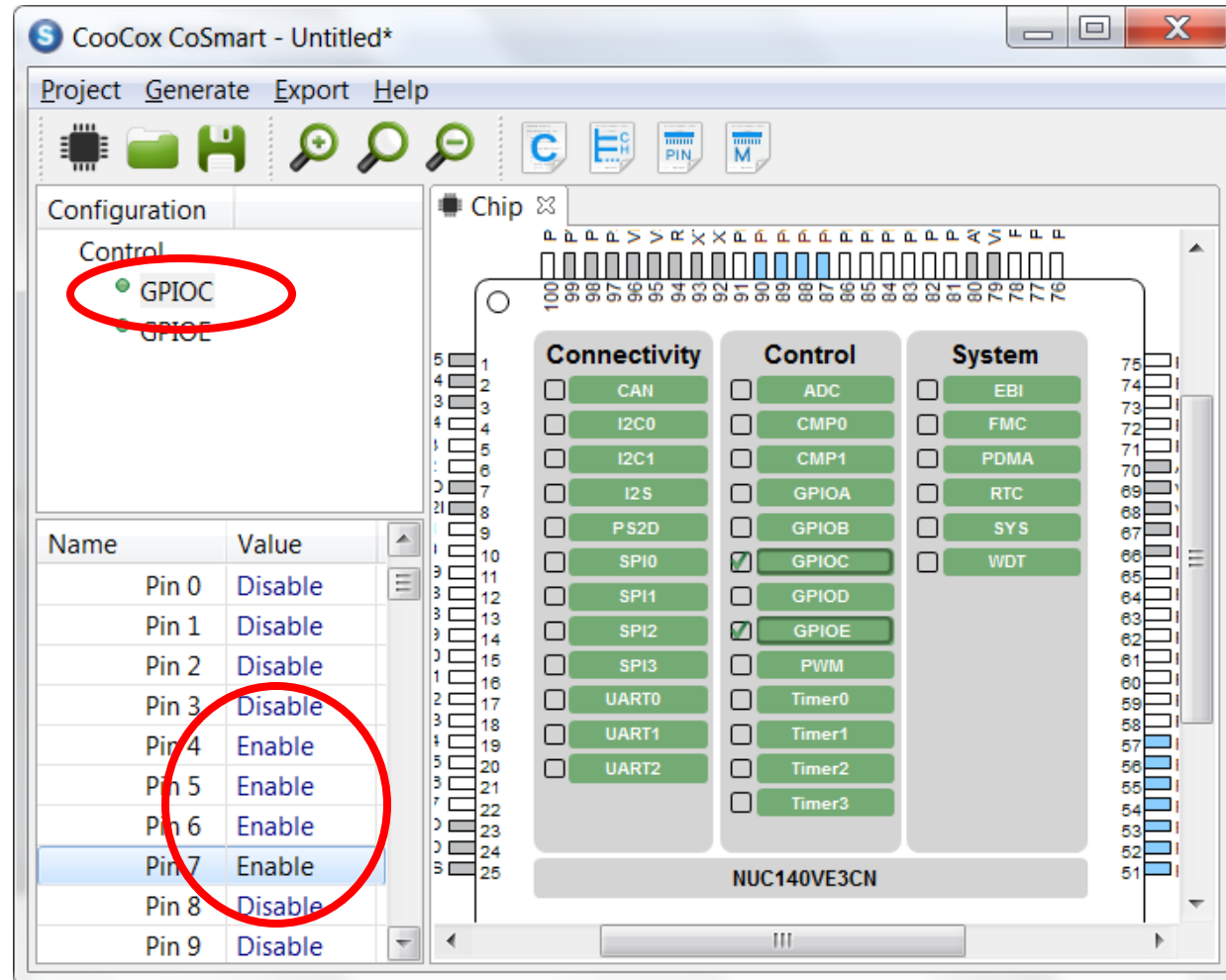
1. Contreng **GPIOE** pada gambar chip.
2. Di jendela Configuration pilih **GPIOE**
3. Di bawahnya, klik pada **Pin 0 s/d 7** agar **Enable**
4. Scroll ke bawah, atur Pin Config 0 s/d 7:
  - **Mode Control: Output**



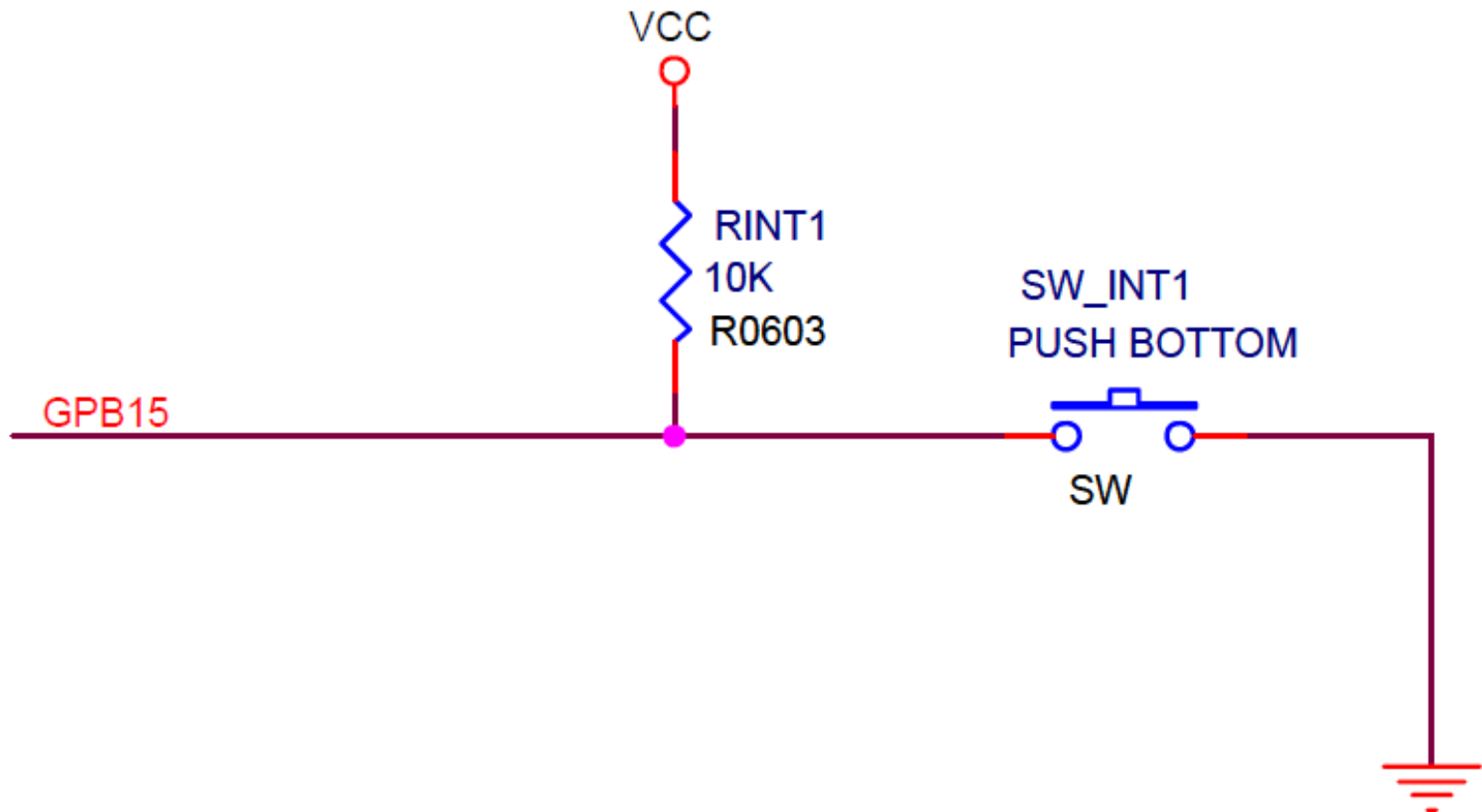
## Langkah 3b. Set Pin 7-Segment Select

1. Di jendela Configuration pilih **GPIOC**
2. Di bawahnya, klik pada **Pin 4 s/d 7** agar **Enable**
3. Scroll ke bawah, atur Pin Config 4 s/d 7:

- **Mode Control: Output**

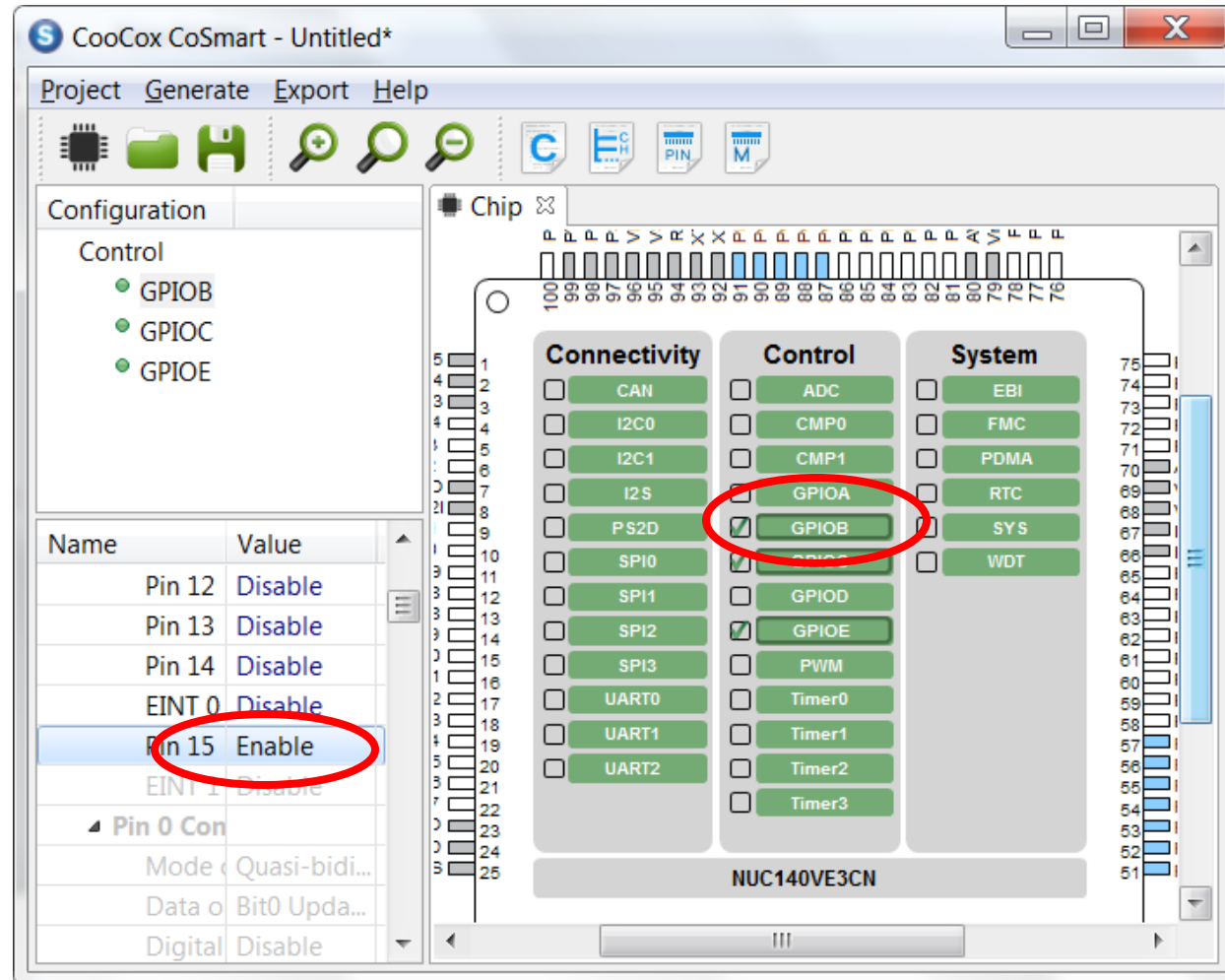


# PUSHBUTTON SWITCH

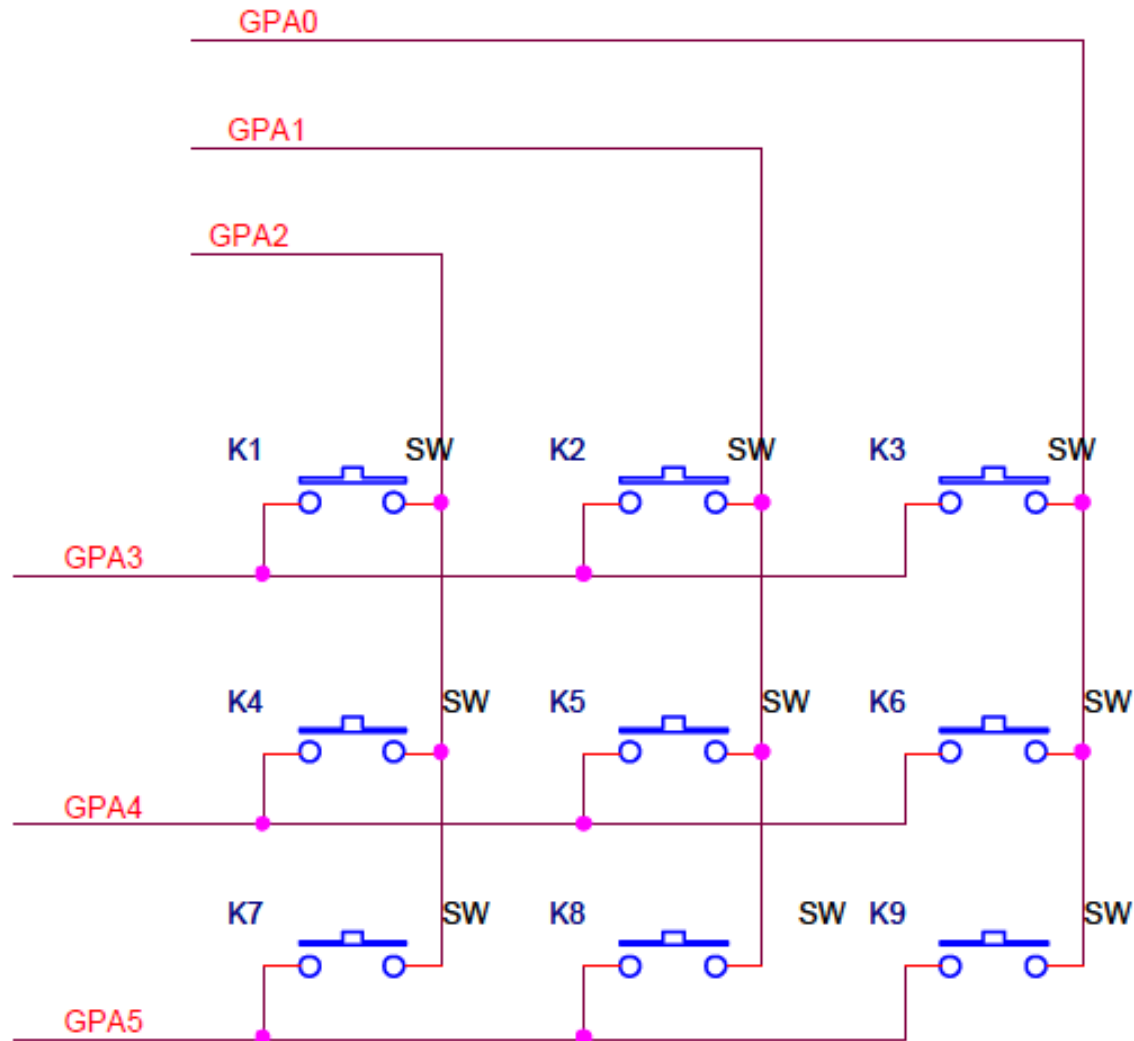


## Langkah 4. Set Pin Pushbutton

1. Contreng **GPIOB** pada gambar chip.
2. Di jendela Configuration pilih **GPIOB**
3. Di bawahnya, klik pada **Pin 15** agar **Enable**
4. Scroll ke bawah, atur Pin Config 15:
  - **Mode Control: Input**
  - **Digital Input: Enable**
  - **Input Signal Deb: Enable**

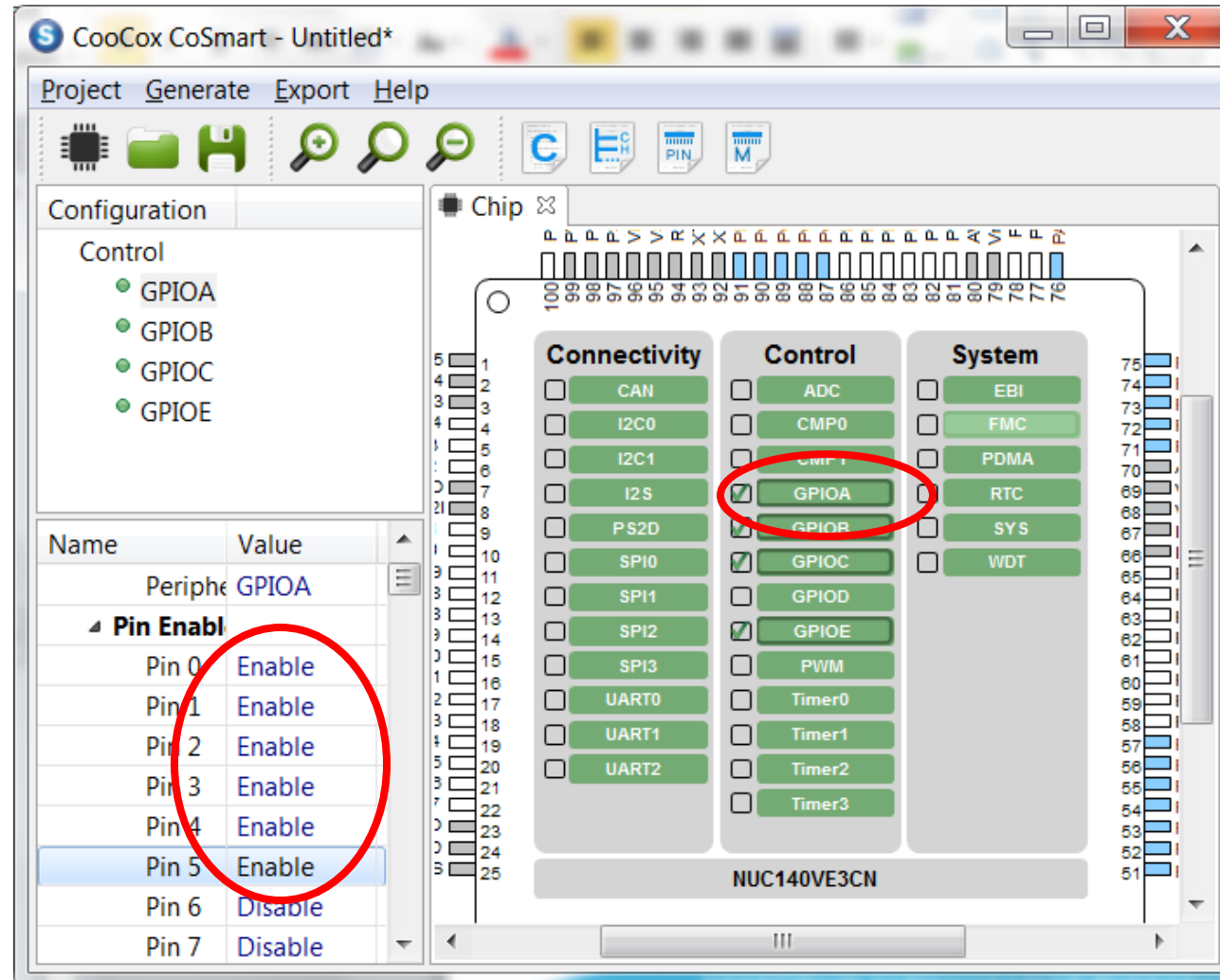


# KEYPAD MATRIX



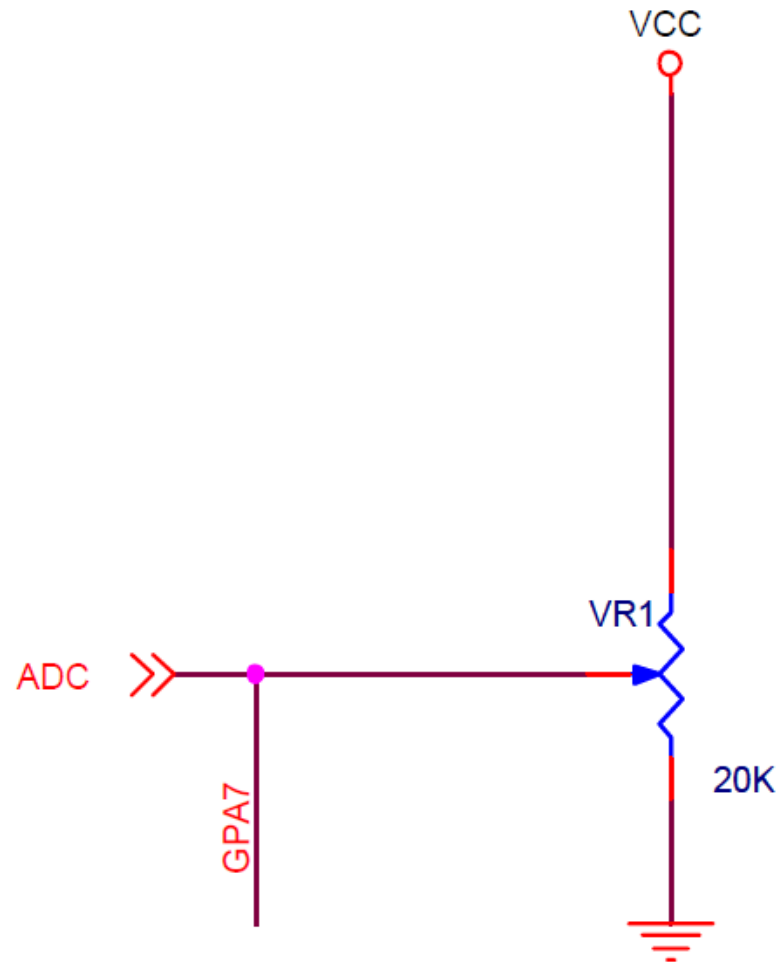
## Langkah 5. Set Pin Keypad

1. Contreng **GPIOA** pada gambar chip.
2. Di jendela Configuration pilih **GPIOA**
3. Di bawahnya, klik pada **Pin 0 s/d 5** agar **Enable**



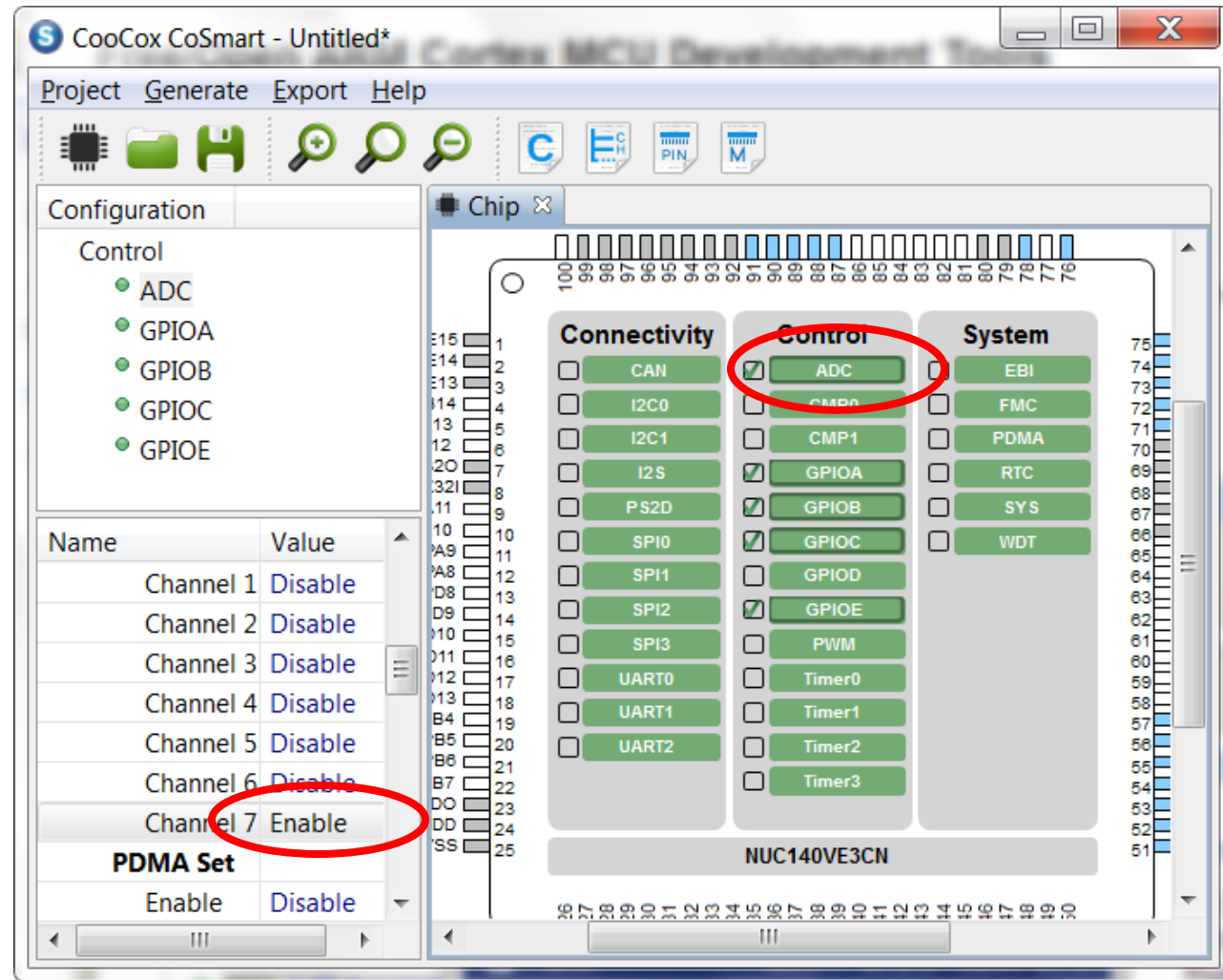


# VARIABEL RESISTOR

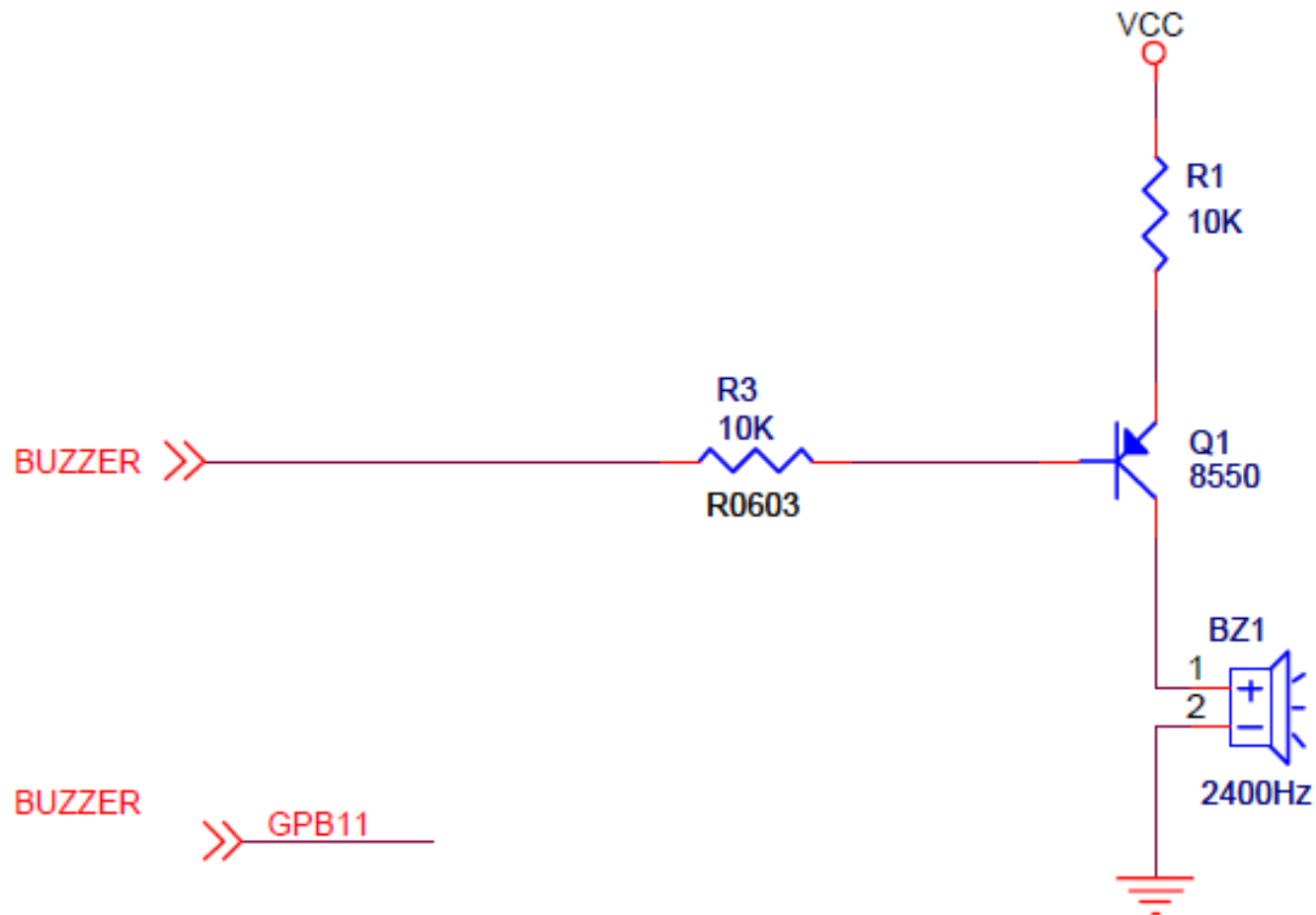


## Langkah 6. Set Pin VR

1. Contreng **ADC** pada gambar chip.
2. Di jendela Configuration pilih **ADC**
3. Klik pada **Channel 7** agar **Enable**
4. Scroll ke atas, klik pada Clock Source, pilih **INTERNAL 22M**.
5. Klik pada Clock Divisor, isi dengan angka **2**.



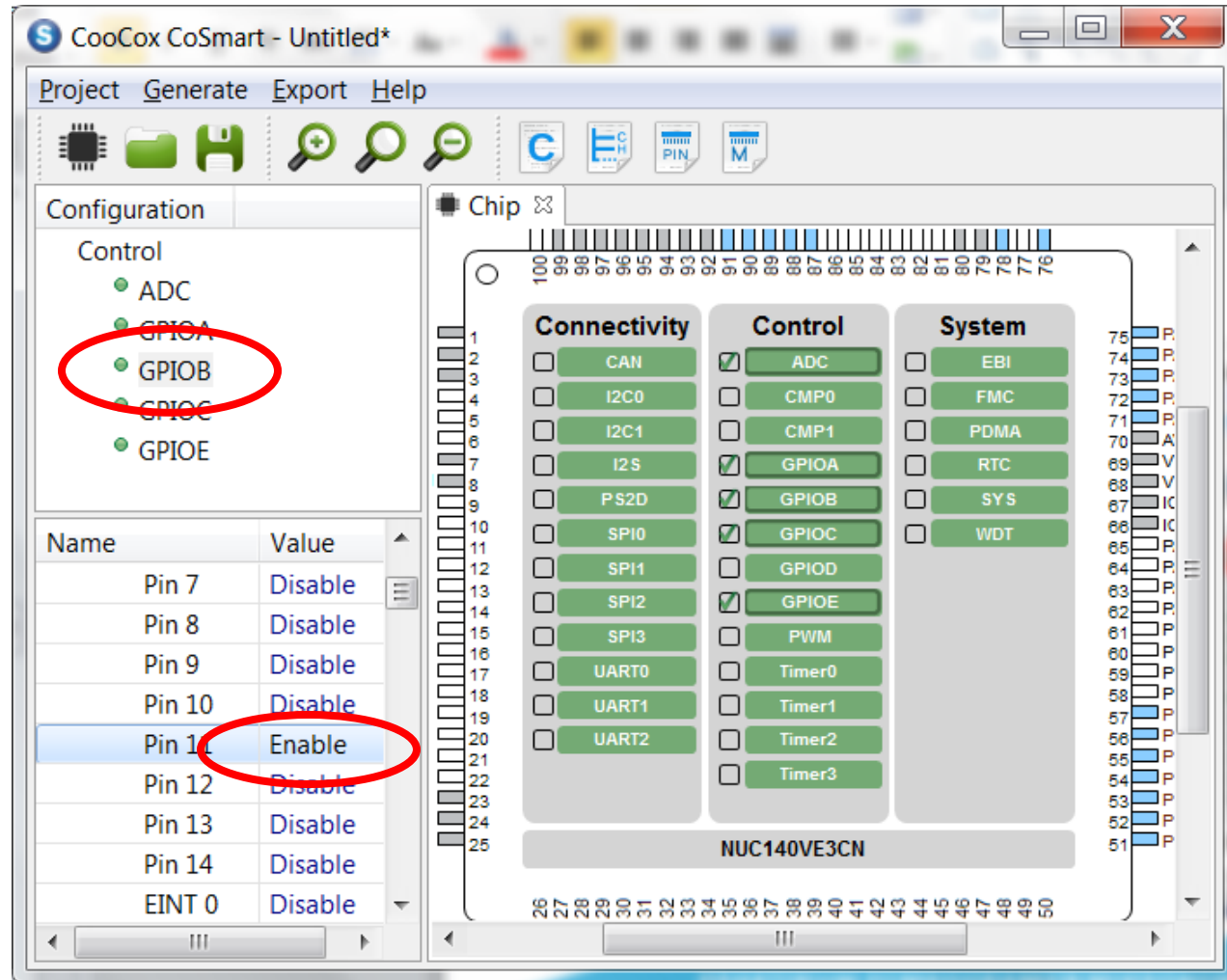
# BUZZER



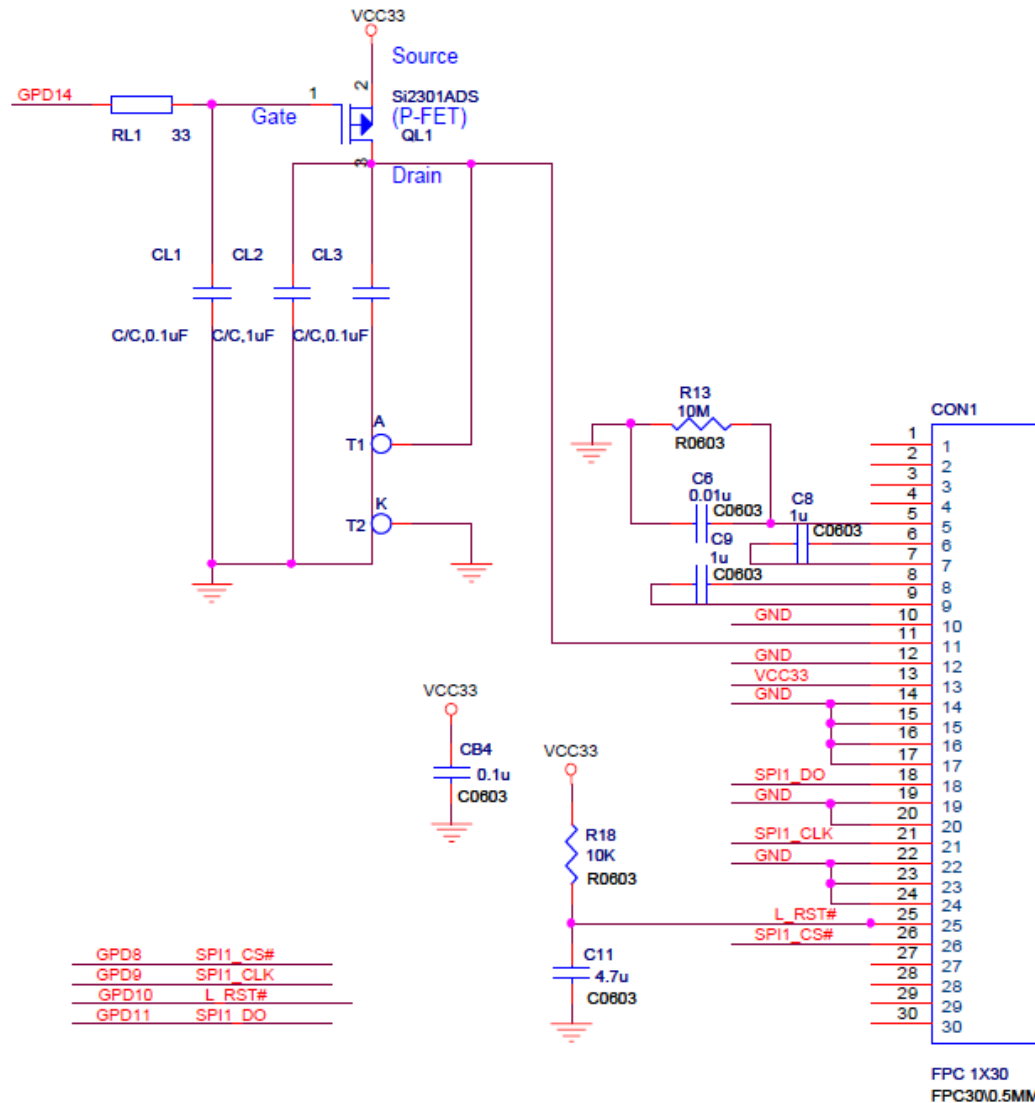
## Step 7. Set Pin Buzzer

1. Di jendela Configuration pilih **GPIOB**
2. Di bawahnya, klik pada **Pin 11** agar **Enable**
3. Scroll ke bawah, atur Pin Config 11:

- **Mode Control: Output**

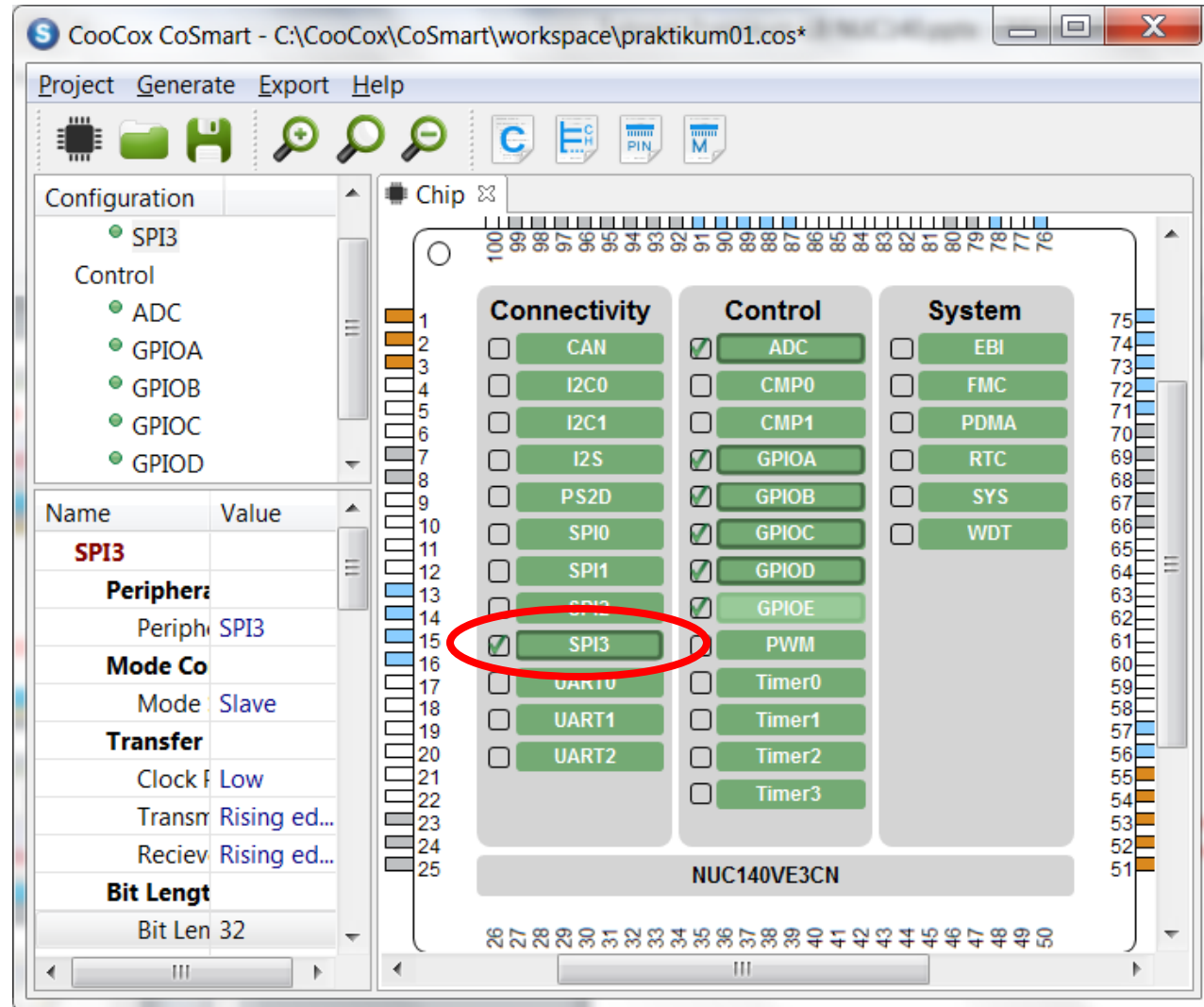


# LCD MATRIX



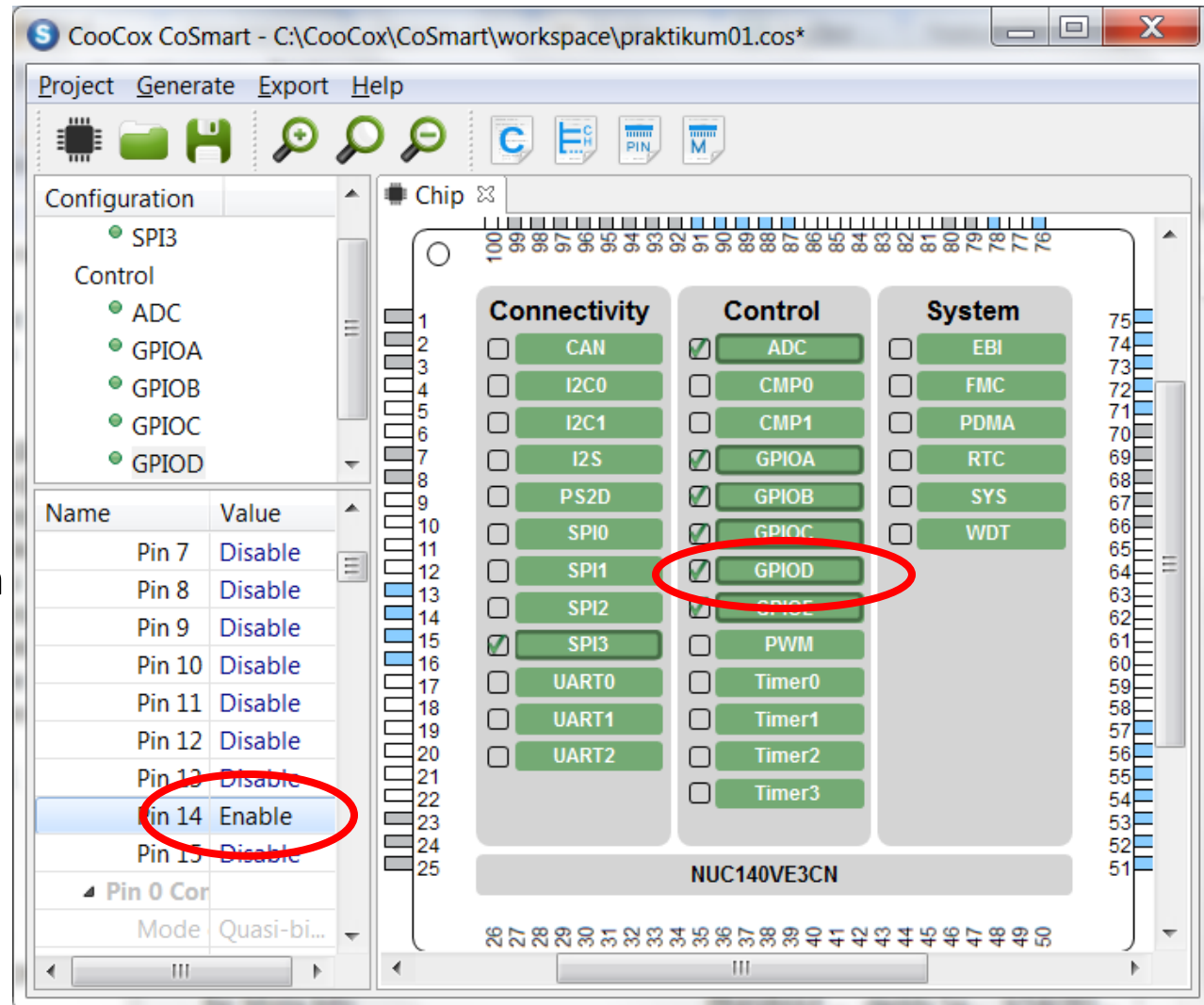
## Step 8a. Set Pin LCD Matrix

Contreng **SPI3** pada gambar chip untuk mengaktifkan fitur komunikasi SPI ke LCD Matrix.



## Step 8b. Set Pin Backlight

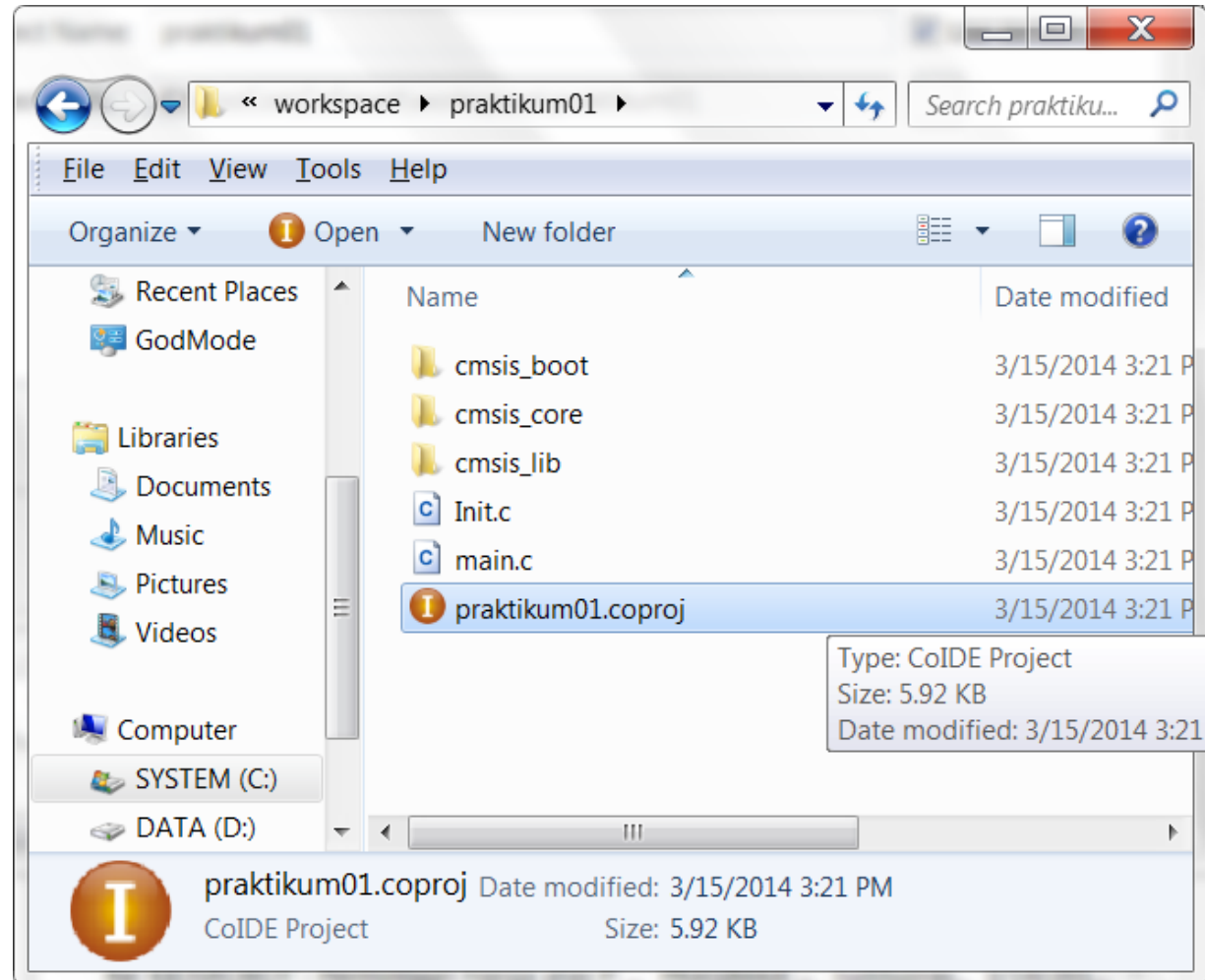
1. Contreng **GPIOD** pada gambar chip.
2. Di jendela Configuration pilih **GPIOD**
3. Di bawahnya, klik pada **Pin 14** agar **Enable**
4. Scroll ke bawah, atur Pin Config 14:
  - **Mode Control: Output**





## Langkah 9: Generate Project

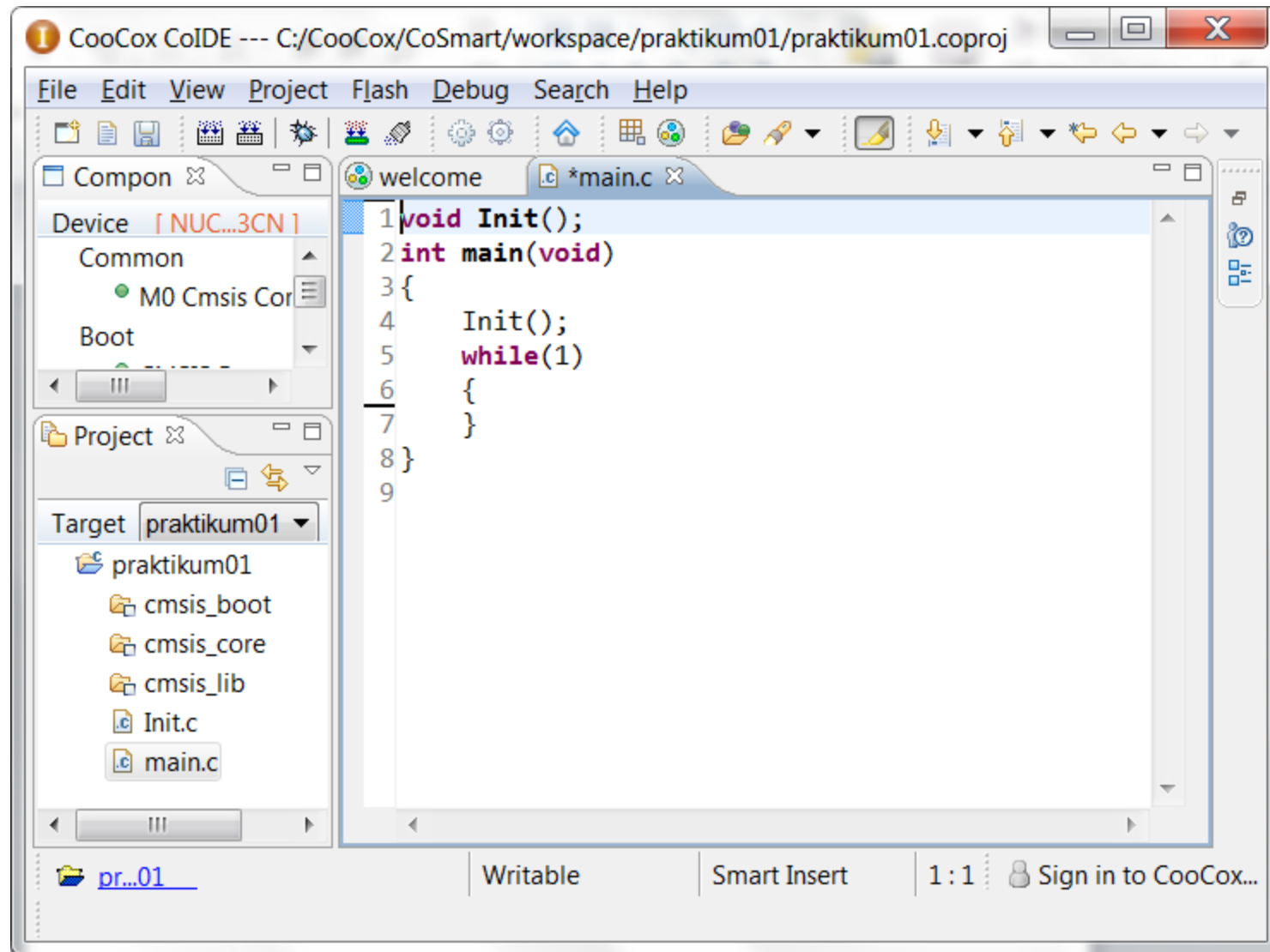
1. Simpan konfigurasi dengan klik **Project > Save** lalu pilih folder dan ketikkan nama file, misal konfigurasi01
2. Lalu klik **Generate > Generate CoIDE Project** lalu pilih folder dan ketikkan nama project, misal praktikum01
3. CoSmart akan membuat file project dan membukanya otomatis di Windows Explorer



## Langkah 10: Buka Project

Klik dua kali file praktikum01 untuk membuka project dengan CoIDE.

Terlihat bahwa file project sudah disiapkan, klik dua kali pada main.c untuk mulai memprogram.



## 5. PEMROGRAMAN LED

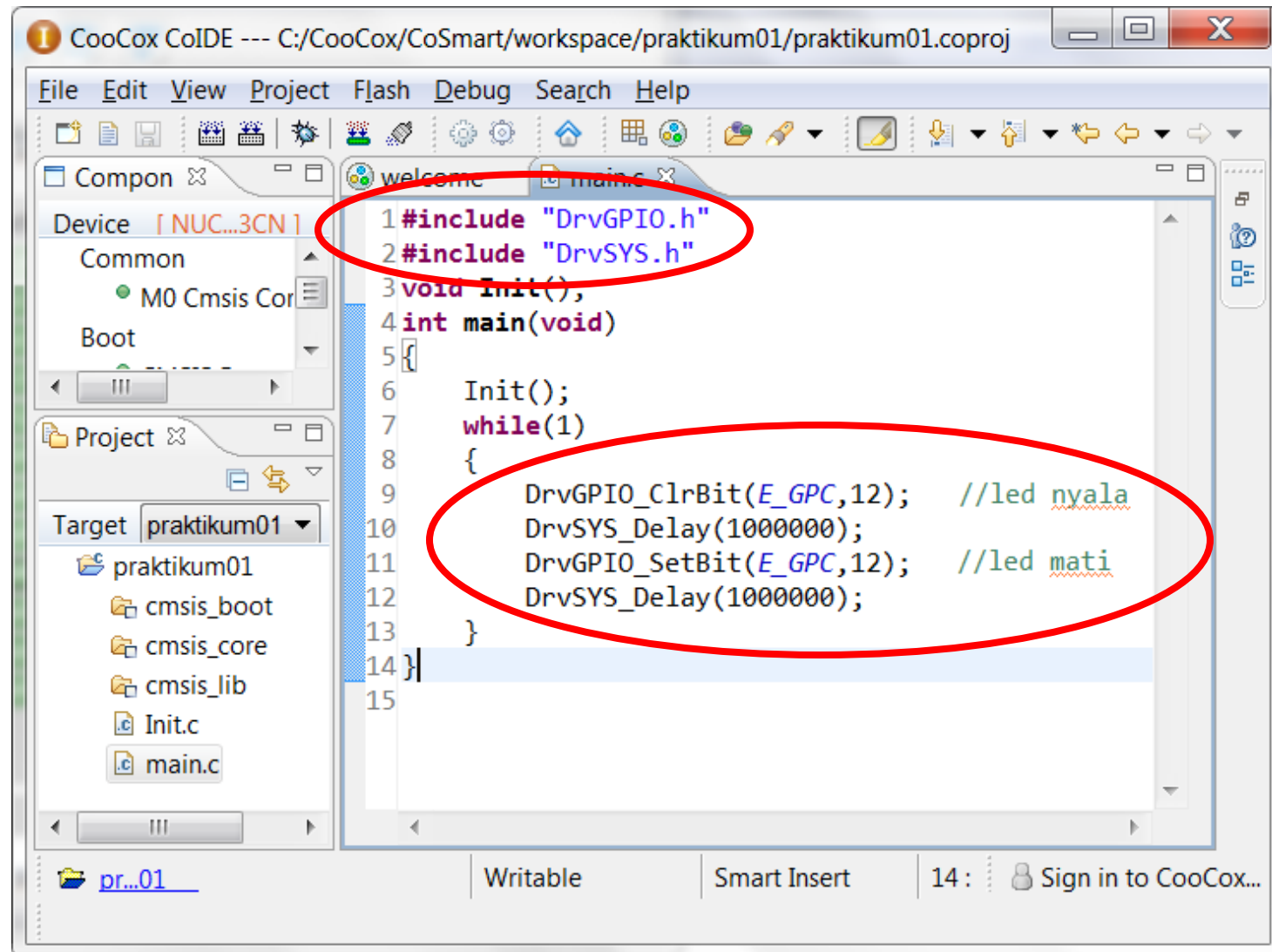
**LED** pada rangkaian hardware adalah digital output yang bersifat **ACTIVE LOW**, yaitu menyala jika diberi logika LOW, dan mati jika diberikan logika HIGH.

- Untuk memberikan logika HIGH gunakan perintah berikut:  
**DrvGPIO\_SetBit>Nama\_port, Nomor\_pin);**
- Untuk memberikan logika LOW gunakan perintah berikut:  
**DrvGPIO\_ClrBit>Nama\_port, Nomor\_pin);**
- Untuk membuat delay bisa menggunakan perintah berikut:  
**DrvSys\_Delay(Lama\_delay\_dalam\_cycle);**
- Untuk bisa menggunakan perintah di atas, perlu memasukkan library  
**#include "DrvGPIO.h"**  
**#include "DrvSYS.h"**

**Nama\_port** diisi E\_GPA / E\_GPB / E\_GPC / E\_GPD / E\_GPE. Dalam hal ini karena LED dihubungkan ke PORT C maka isi dengan **E\_GPC**.

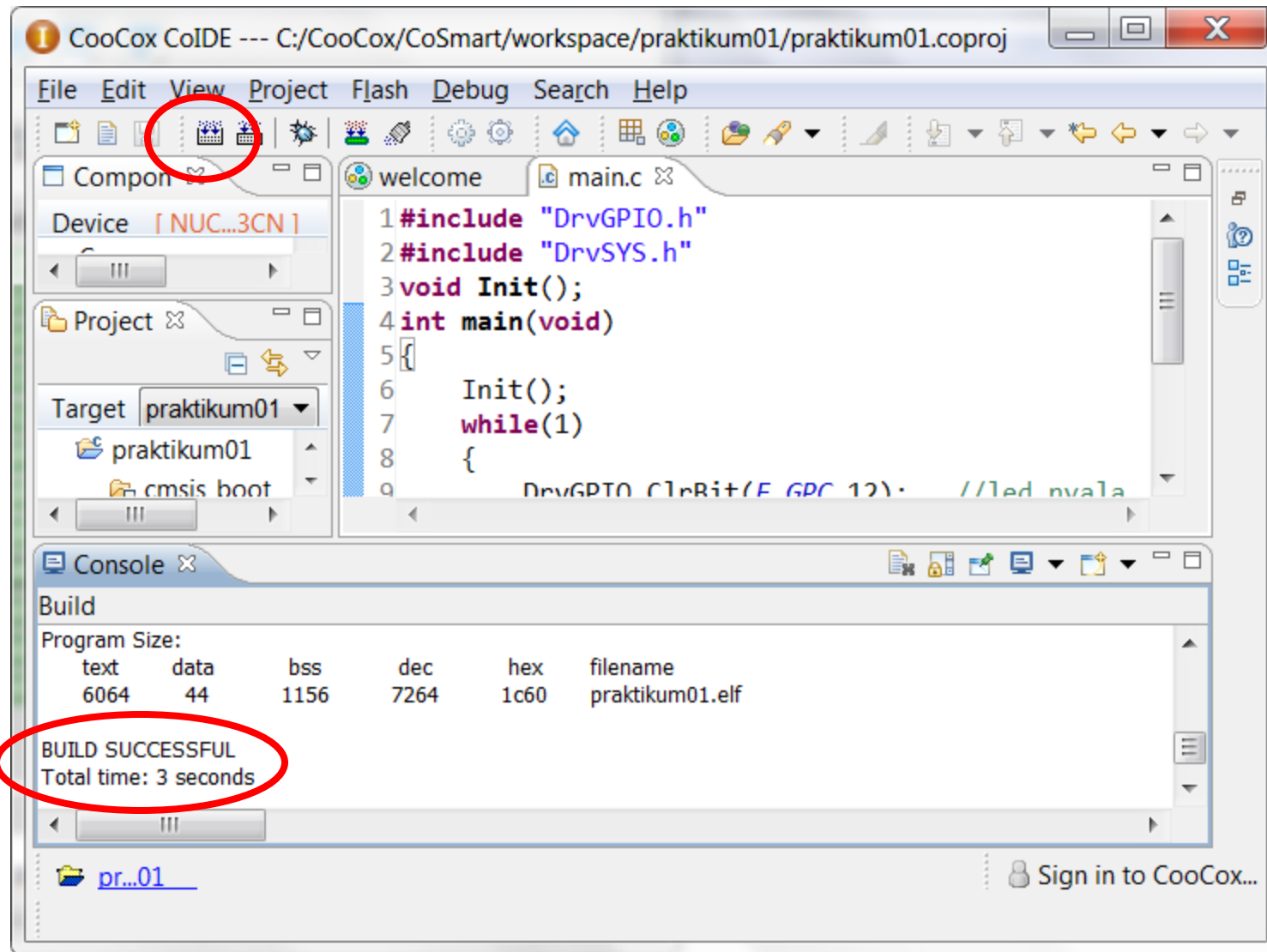
**Nomor\_pin** diisi dengan angka pin kaki mikrokontroler. Dalam hal ini empat buah LED sudah dihubungkan ke pin **12 s/d 15**.

Untuk membuat nyala LED berkedip, maka tambahkan baris program sehingga menjadi seperti gambar di samping.



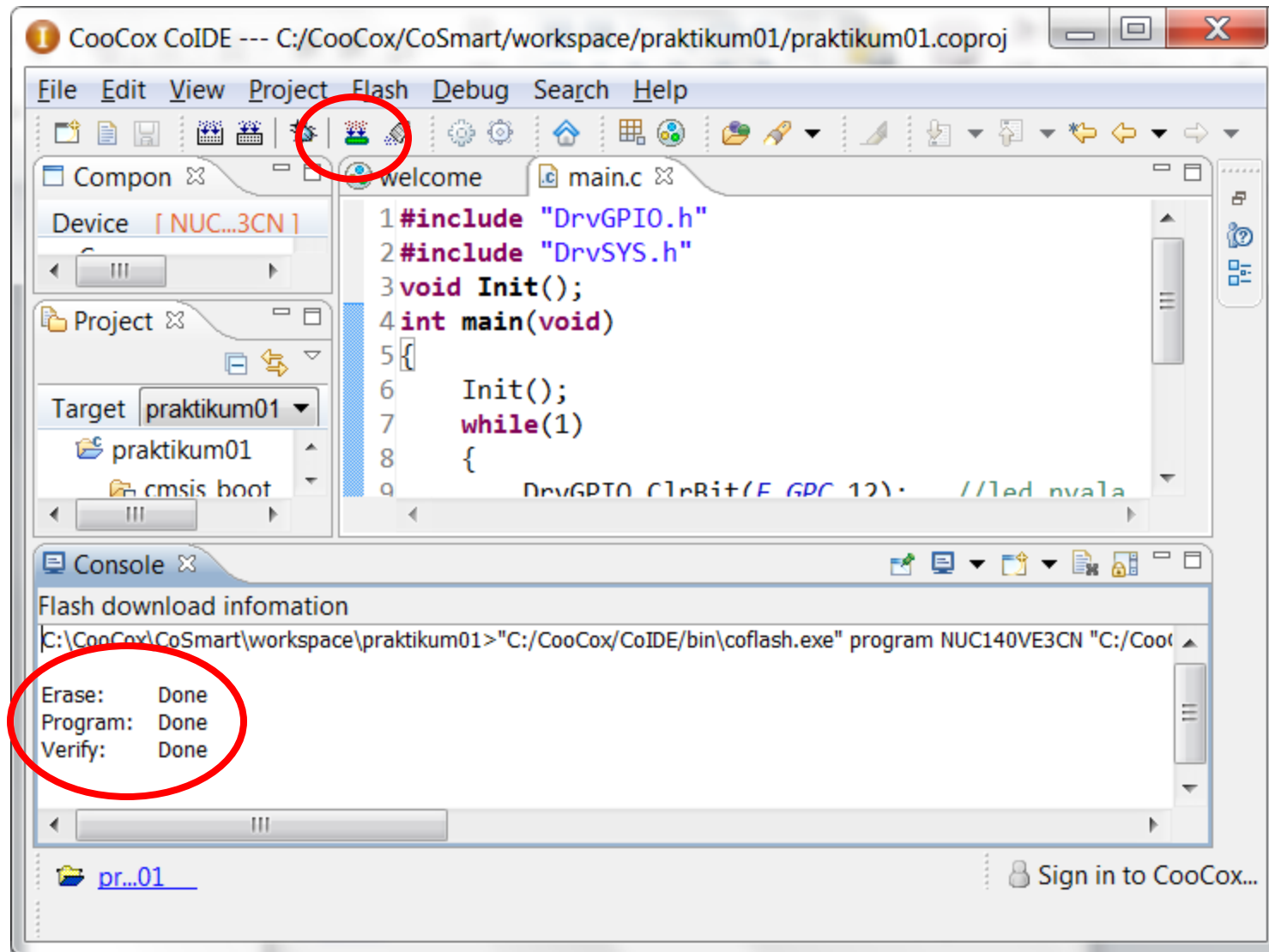
Klik icon  
**Build (F7)** di  
toolbar untuk  
melakukan  
compile  
program.

Pastikan Build  
Successful,  
tidak ada  
error.



Klik icon  
**Download  
Code to Flash**  
di toolbar  
untuk  
memasukkan  
program ke  
chip.

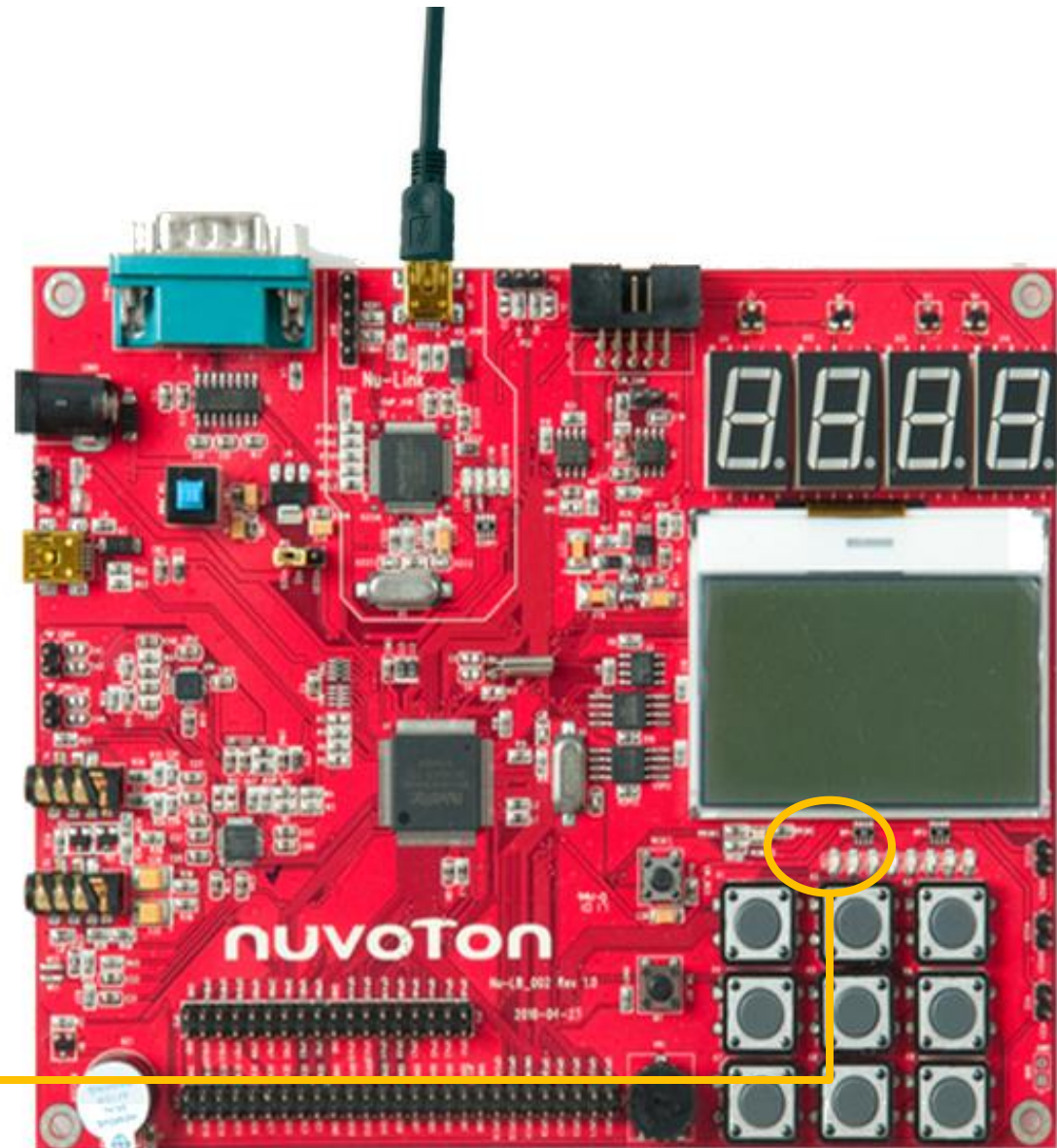
Pastikan Done,  
tidak ada error.





Lihat hasilnya di board.

LED akan menyala berkedip.





## 6. PEMROGRAMAN PUSHBUTTON SWITCH

**Pushbutton** switch pada rangkaian hardware adalah digital input yang bersifat **ACTIVE LOW**, yaitu jika ditekan memberikan logika LOW.

- Untuk mengambil nilai logika input digital gunakan perintah berikut:  
**DrvGPIO\_GetBit>Nama\_port, Nomor\_pin);**

### Keterangan:

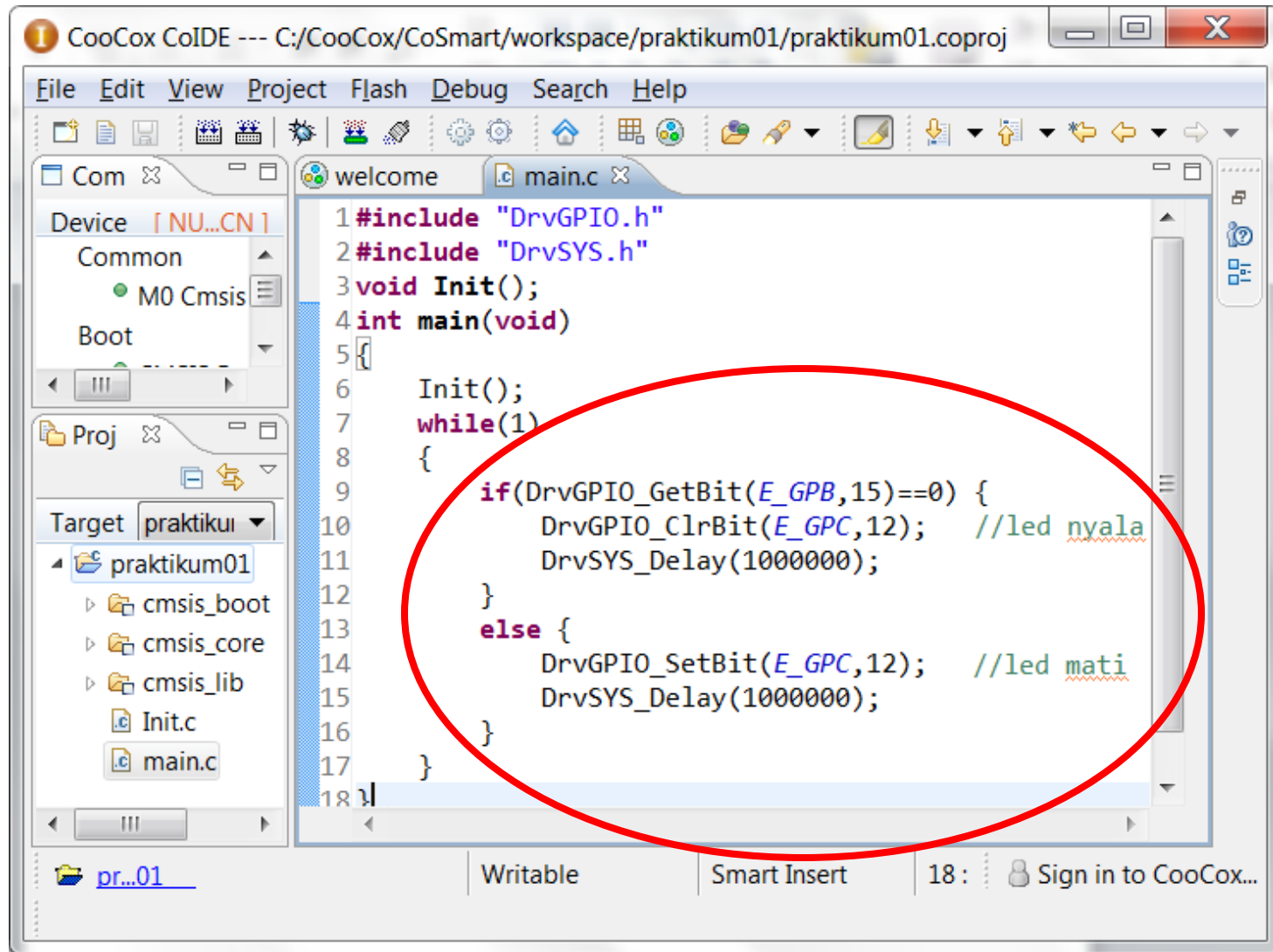
**Nama\_port** diisi E\_GPA / E\_GPB / E\_GPC / E\_GPD / E\_GPE. Dalam hal ini karena Pushbutton switch dihubungkan ke PORT B maka isi dengan **E\_GPB**.

**Nomor\_pin** diisi dengan angka pin kaki mikrokontroler di PORT yang terhubung Pushbutton switch yang ingin dimainkan. Dalam hal ini empat buah Pushbutton switch sudah dihubungkan ke pin **15**.

Fungsi **DrvGPIO\_GetBit** menghasilkan **nilai 0 jika logika LOW** (pushbutton ditekan), atau 1 jika logika HIGH (pushbutton tidak ditekan).

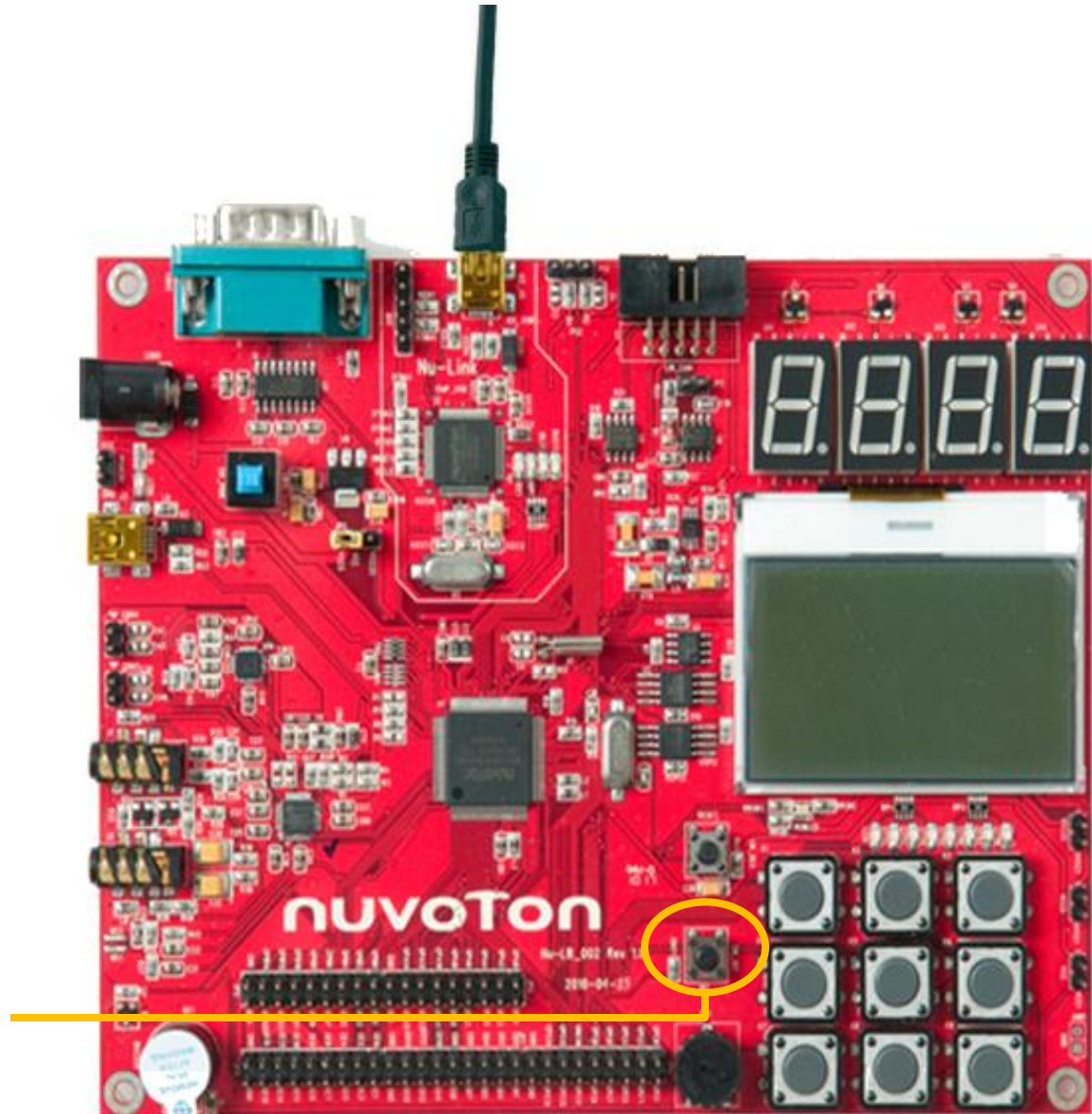
Sehingga untuk **membuat nyala LED** tergantung **penekanan pushbutton**, ubah baris program sehingga menjadi seperti gambar di samping.

Build & Download ke chip.



LED akan menyala hanya ketika Pushbutton ditekan.

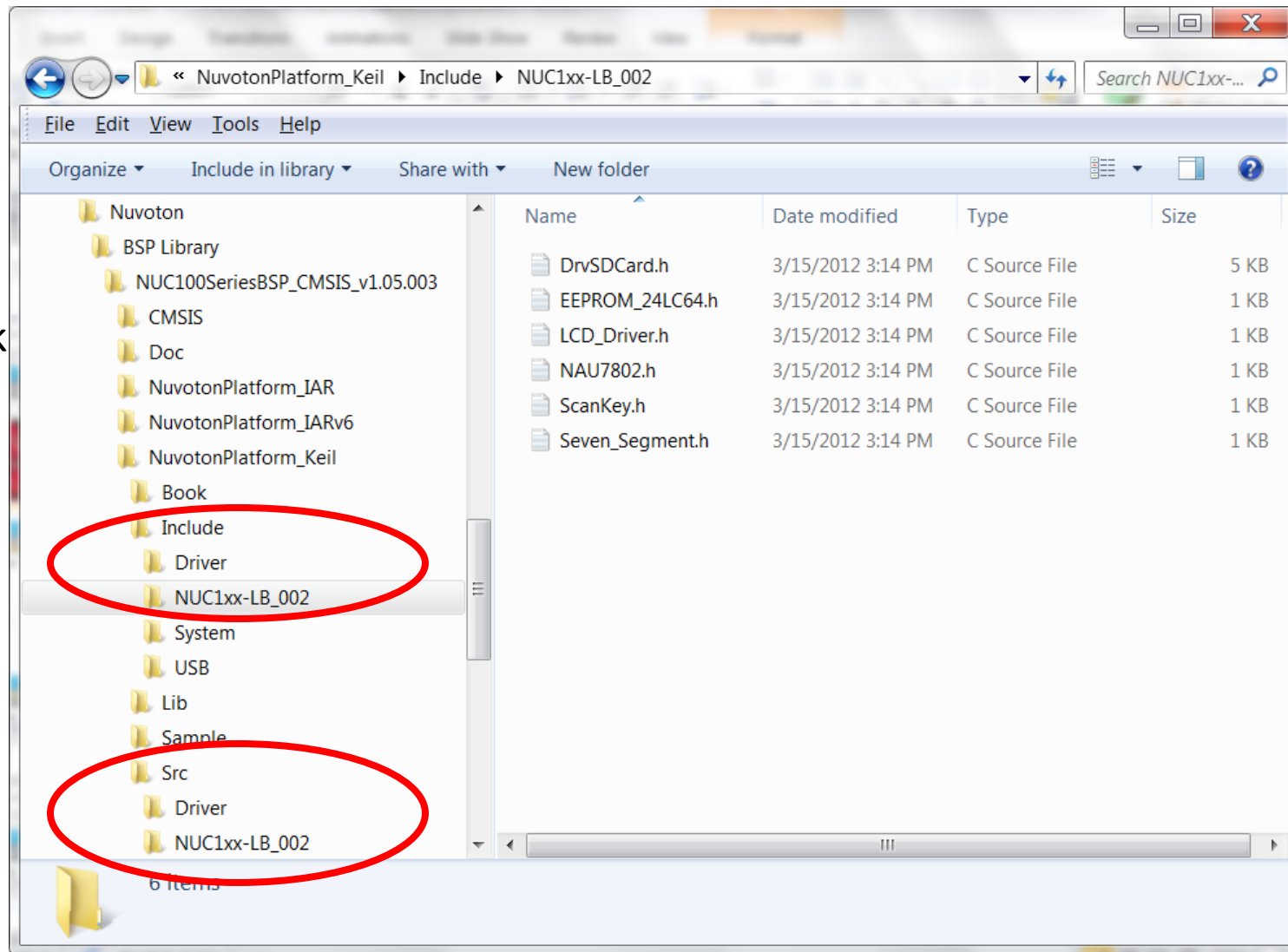
Tes dengan menekan tombol Pushbuton yang bawah.



## 7. PEMROGRAMAN 7-SEGMENT DISPLAY

Nuvoton telah menyediakan **Board Support Package** (yang sudah di-instal) yang berisi banyak library yang dapat mempermudah kita memprogram object-object di Learning Board.

Untuk 7-segment kita bisa coba pakai library dari BSP.

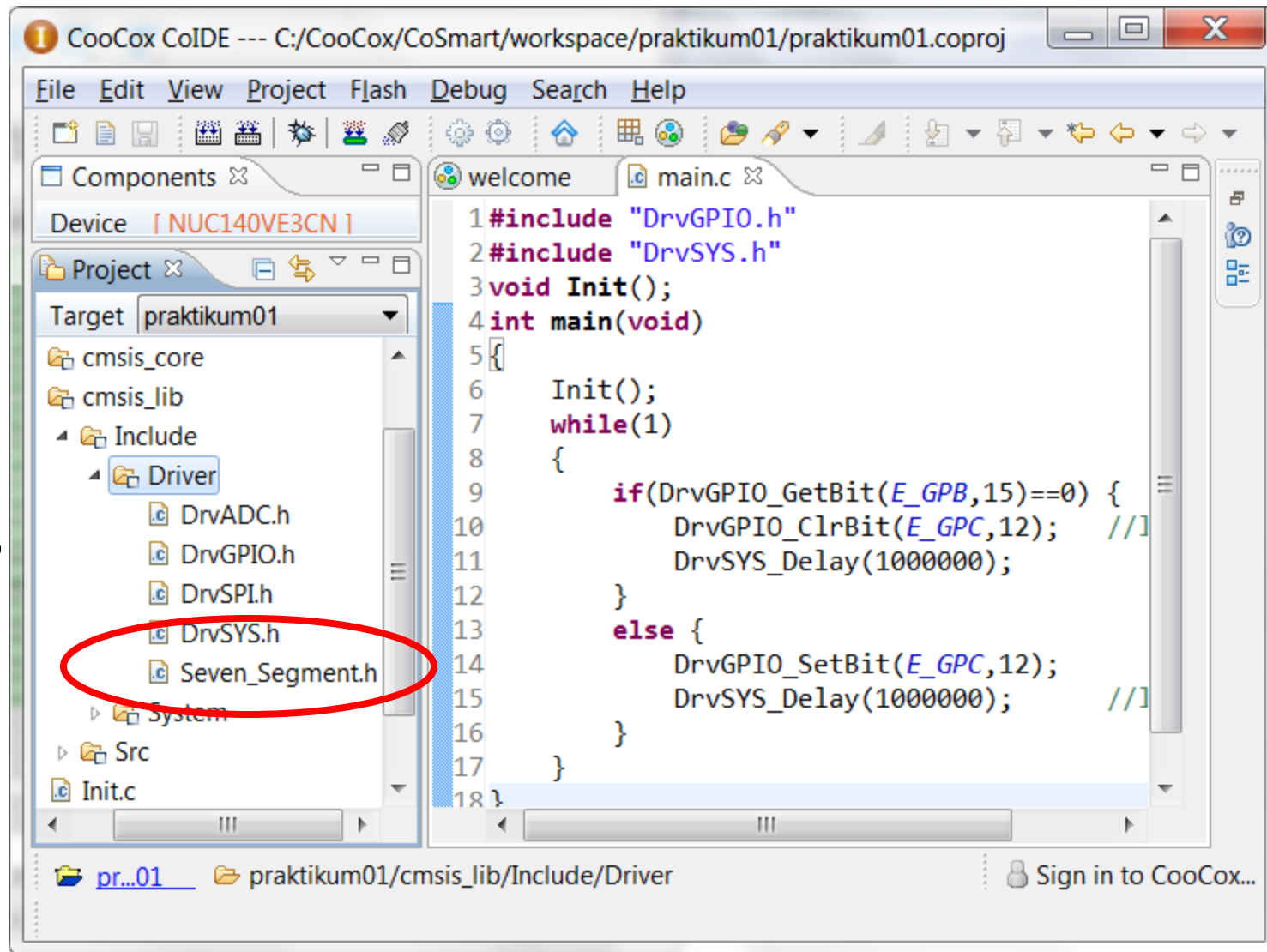


## Langkah 1:

Klik kanan di  
cmsis\_lib\Include  
\Driver, lalu pilih  
**Add Files.**

Browse ke  
C:\Nuvoton  
\BSP Library  
\NUC100SeriesBSP  
\NuvPlatform\_Keil  
\Include  
\NUC1xx-LB\_002

Pilih file  
**Seven\_Segment.h**

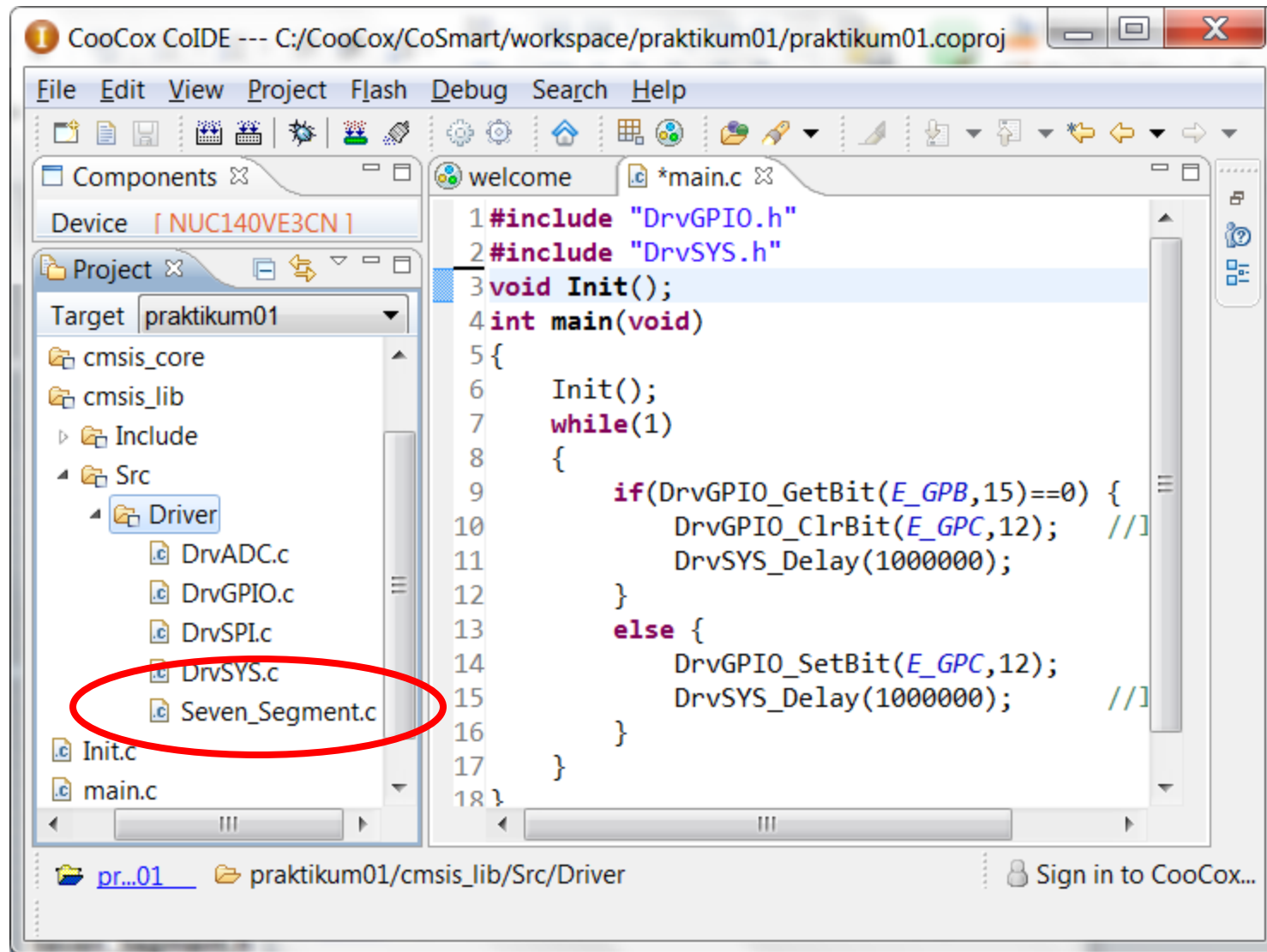


## Langkah 2:

Klik kanan di  
cmsis\_lib\Src  
\Driver, lalu pilih  
**Add Files.**

Browse ke  
C:\Nuvoton  
\BSP Library  
\NUC100SeriesBSP  
\NuvPlatform\_Keil  
\Src  
\NUC1xx-LB\_002

Pilih file  
**Seven\_Segment.c**







7-segment display di Learning Board menggunakan metode scanning. Untuk memunculkan angka di salah satu 7-segment maka berikan data selagi salah satu 7-segment yang dituju diaktifkan, sementara 7-segment lain tidak diaktifkan.

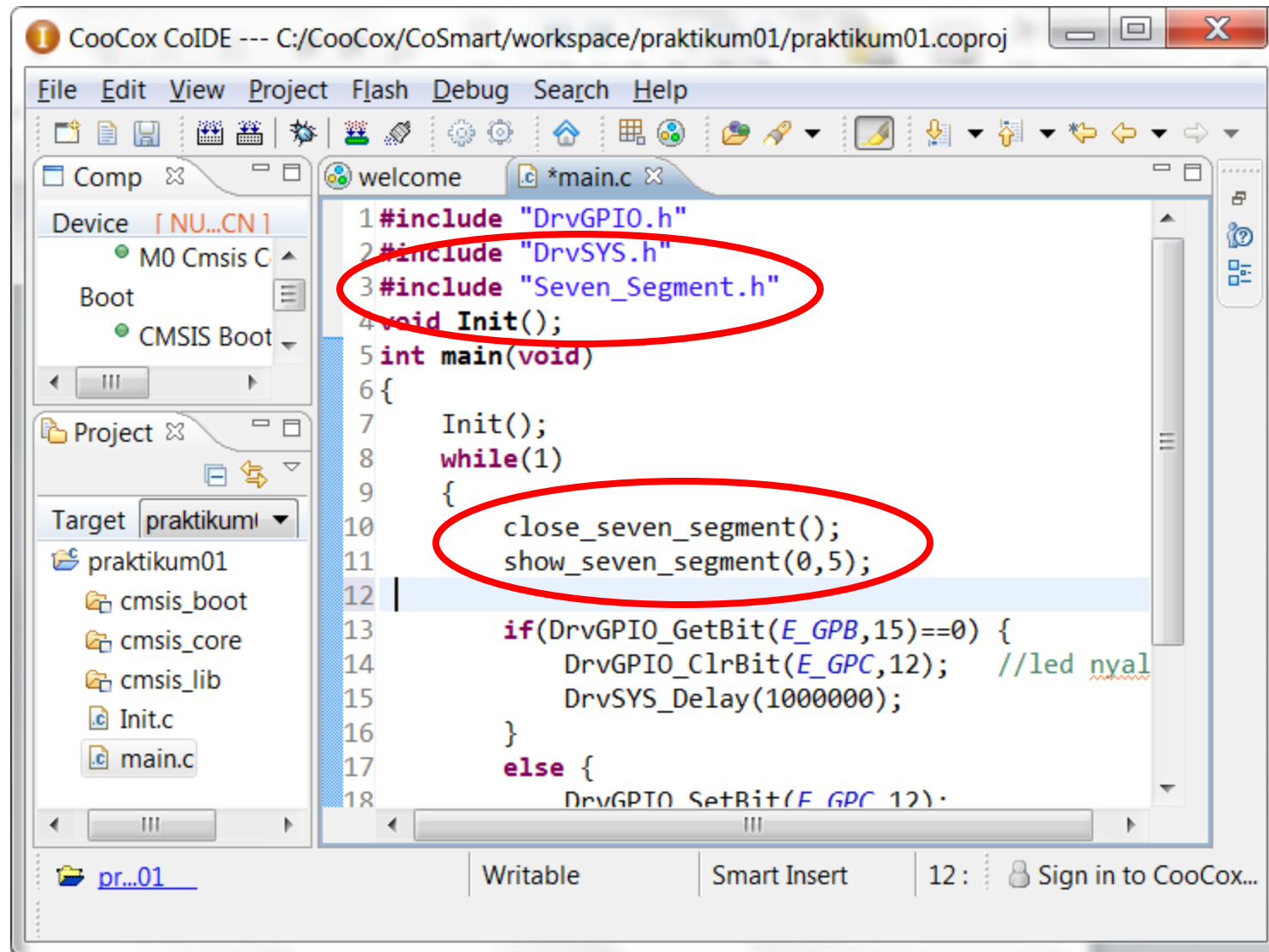
- Untuk menonaktifkan semua 7-segment bisa menggunakan perintah berikut:  
**close\_seven\_segment();**
- Untuk mengaktifkan dan memberi data salah satu 7-segment, gunakan perintah berikut:  
**show\_seven\_segment(Nomor\_7segment,Data\_angka);**
- Untuk bisa menggunakan perintah di atas, perlu memasukkan library  
**#include "Seven\_Segment.h"**

### Keterangan:

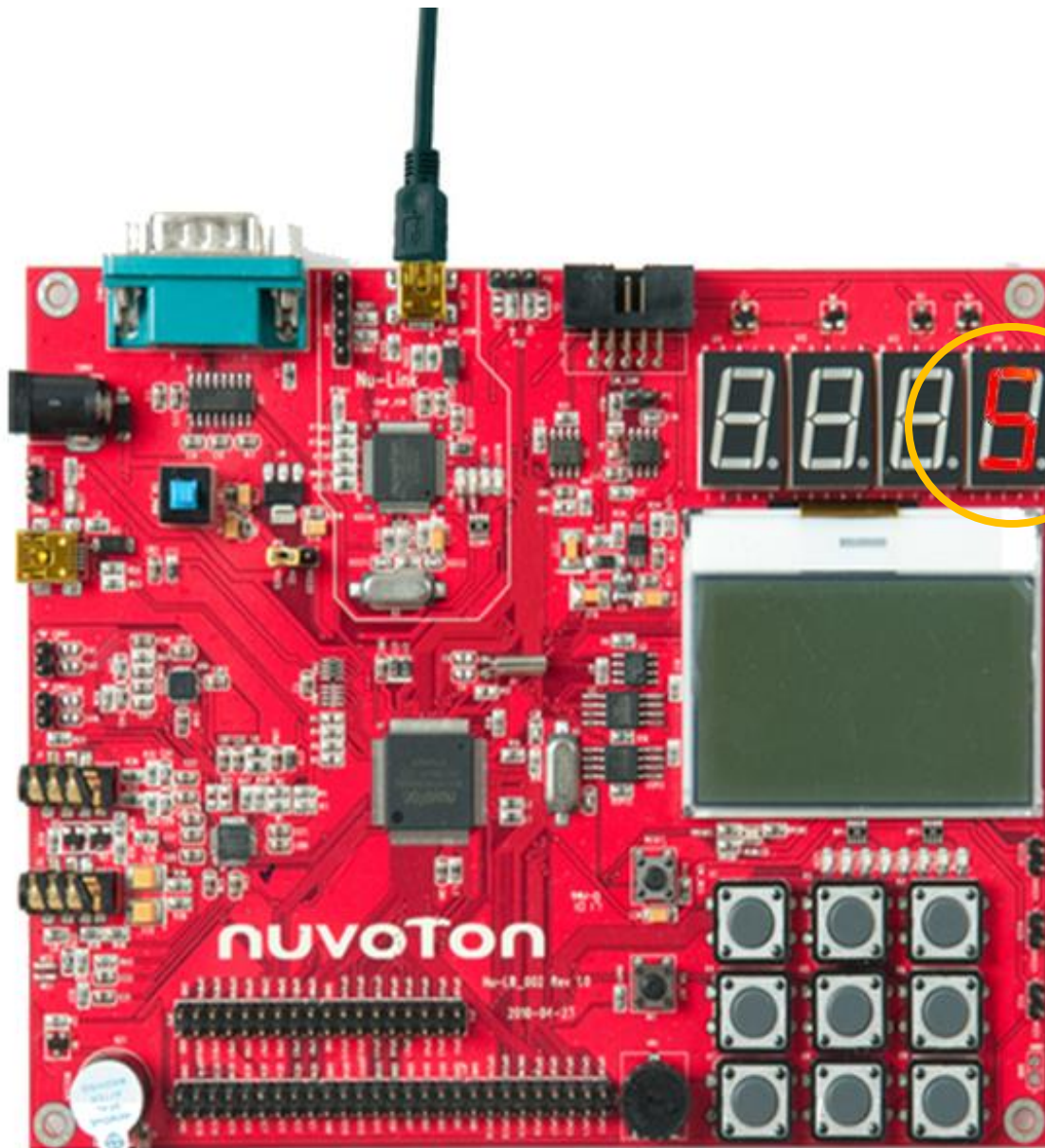
Nomor\_7segment diisi dengan nomor urut 0-3, dihitung dari paling kanan.  
Data\_angka diisi dengan angka 0-9.

Untuk menampilkan angka 5 di 7-segment paling kanan, tambahkan baris program sehingga menjadi seperti gambar di samping.

Build & Download ke chip. Pastikan tidak ada error.





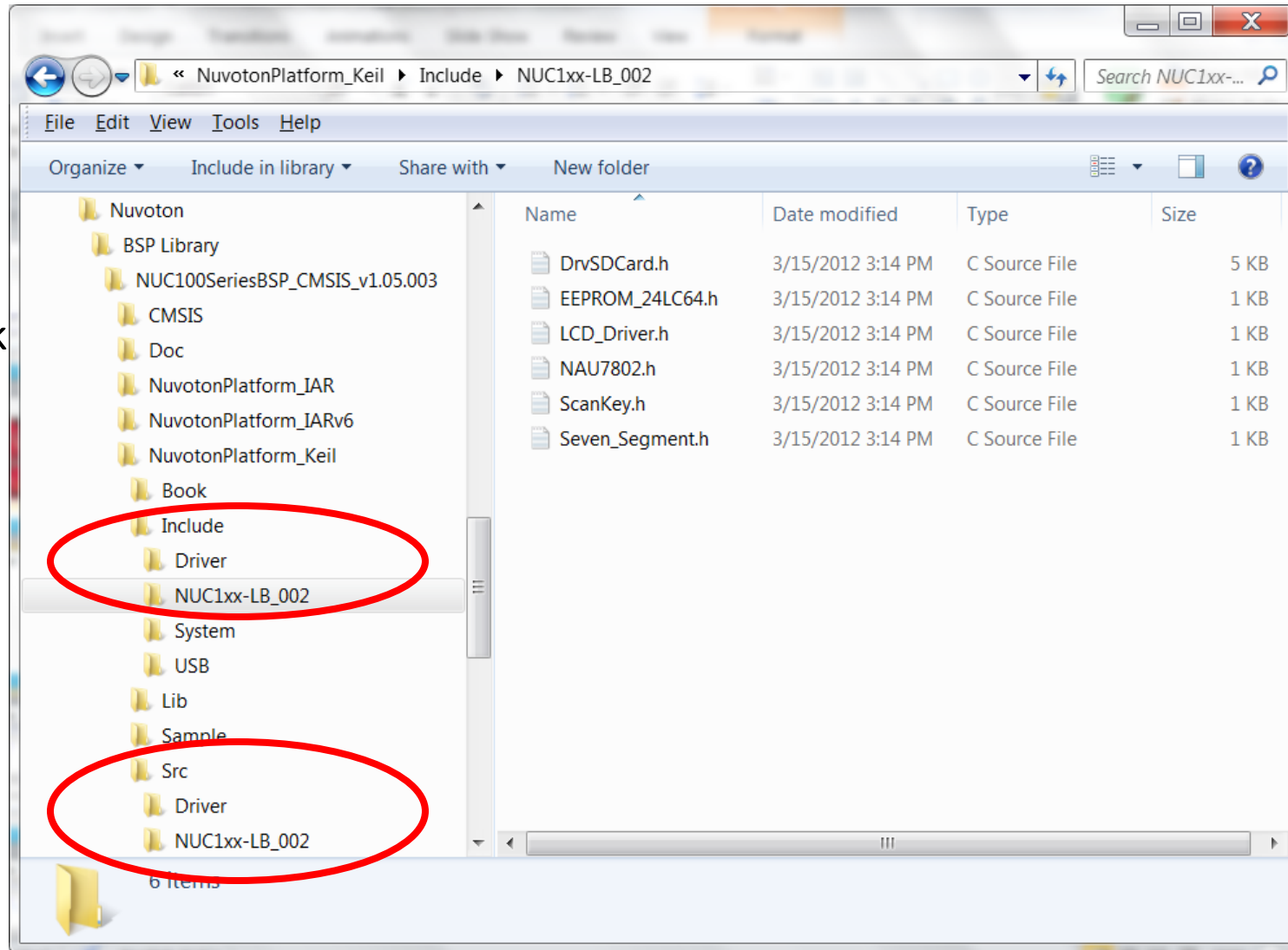


7-segment display paling kanan akan menampilkan angka 5.

## 8. PEMROGRAMAN KEYPAD MATRIX

Nuvoton telah menyediakan **Board Support Package** (yang sudah di-instal) yang berisi banyak library yang dapat mempermudah kita memprogram object-object di Learning Board.

Untuk keypad kita bisa coba pakai library dari BSP.

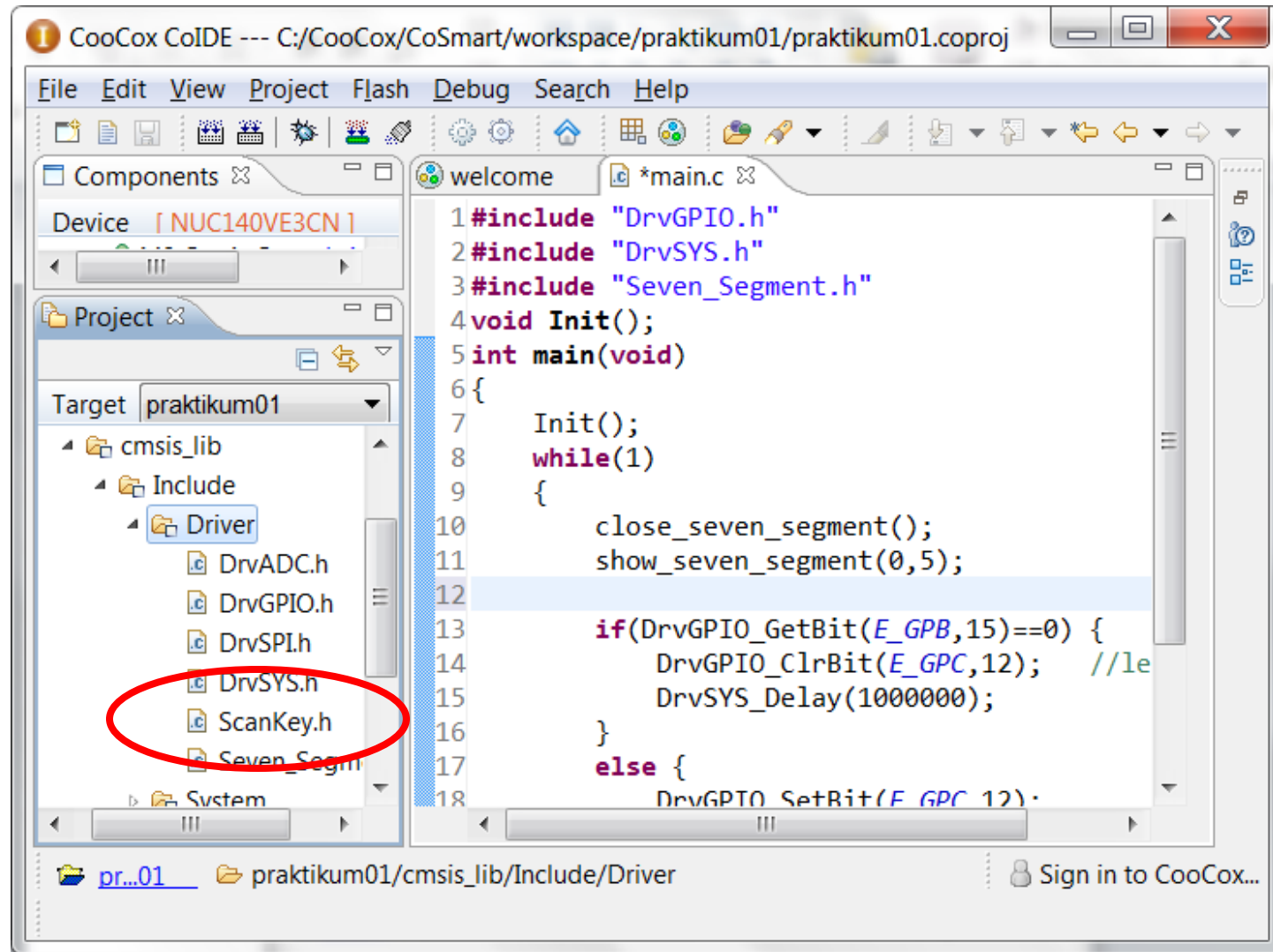


**Langkah 1:**

Klik kanan di  
 cmsis\_lib\Include  
 \Driver, lalu pilih  
**Add Files.**

Browse ke  
 C:\Nuvoton  
 \BSP Library  
 \NUC100SeriesBSP  
 \NuvPlatform\_Keil  
 \Include  
 \NUC1xx-LB\_002

Pilih file **ScanKey.h**

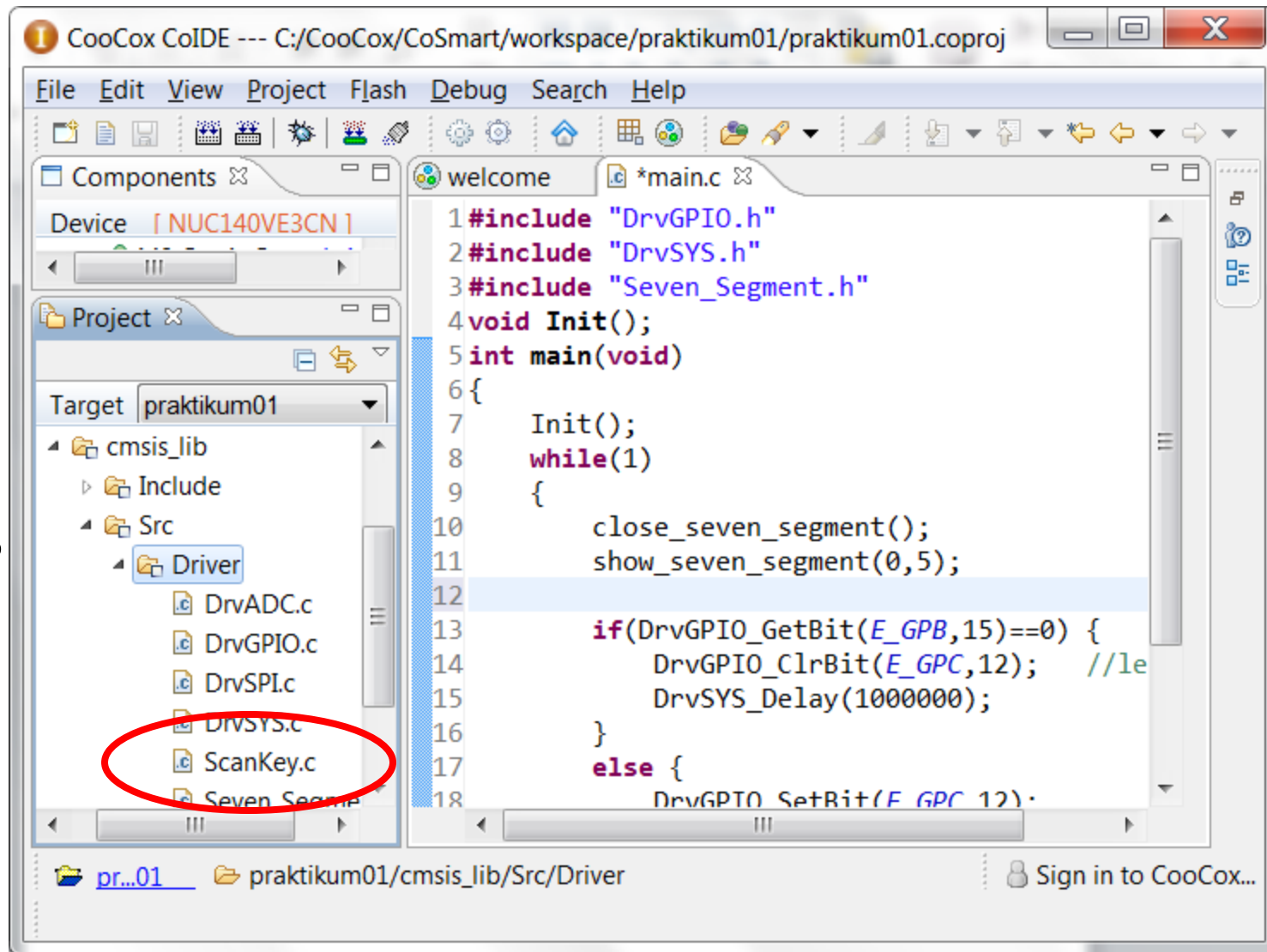


**Langkah 2:**

Klik kanan di  
 cmsis\_lib\Src  
 \Driver, lalu pilih  
**Add Files.**

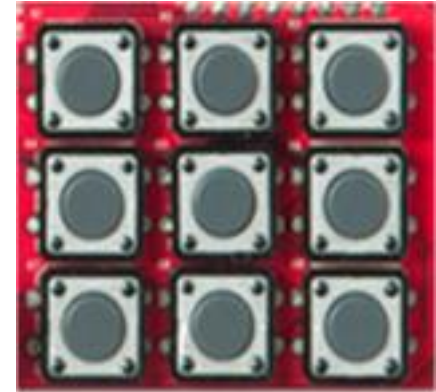
Browse ke  
 C:\Nuvoton  
 \BSP Library  
 \NUC100SeriesBSP  
 \NuvPlatform\_Keil  
 \Src  
 \NUC1xx-LB\_002

Pilih file **ScanKey.c**



Keypad matrix di Learning Board menggunakan metode scanning. Proses scanning dengan cara membaca logic di Row, ketika Column diberikan logic LOW.

Jika tombol ditekan maka Column dan Row yang bersesuaian menjadi terhubung, sehingga Row ikut menjadi LOW.



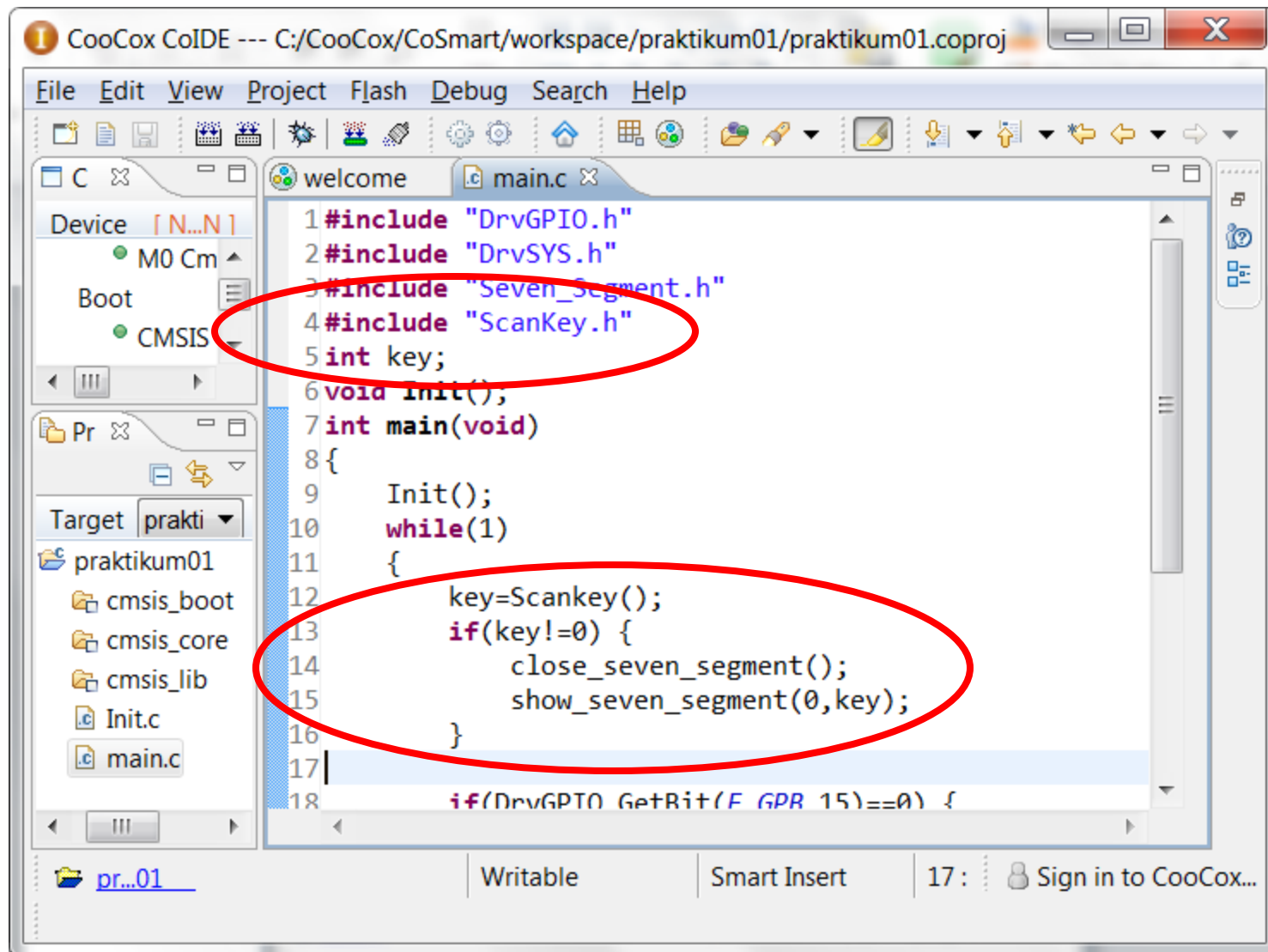
- Untuk membaca penekanan keypad, bisa menggunakan perintah berikut:  
**Scankey();**
- Untuk bisa menggunakan perintah di atas, perlu memasukkan library  
**#include "ScanKey.h"**
- Untuk menampung hasil pembacaan keypad, ada baiknya membuat suatu variabel, misal bernama Key bertipe data integer:  
**int key;**

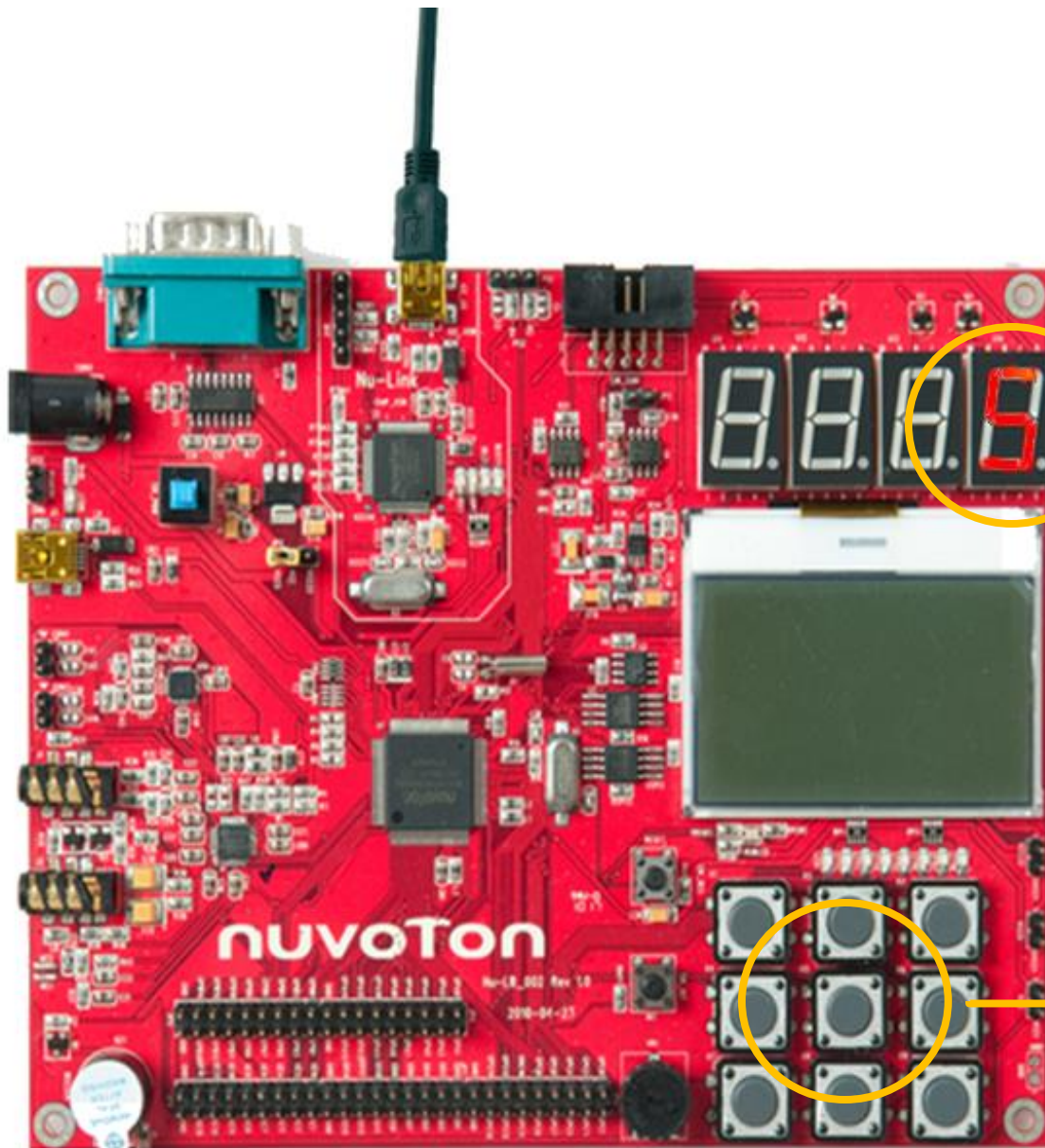
Hasil pembacaan adalah 0 jika tidak ada yang ditekan, atau berupa angka 1 s/d 9 jika ada yang ditekan.



Untuk menampilkan apa yang ditekan di keypad ke 7-segment, ubah baris program sehingga menjadi seperti gambar di samping.

Build & Download ke chip. Pastikan tidak ada error.





7-segment display paling kanan berubah sesuai penekanan tombol keypad

## 9. PEMROGRAMAN BUZZER

**BUZZER** pada rangkaian hardware adalah digital output yang bersifat **ACTIVE LOW**, yaitu berbunyi jika diberi logika LOW, dan mati jika diberikan logika HIGH.

- Untuk memberikan logika HIGH gunakan perintah berikut:  
**DrvGPIO\_SetBit>Nama\_port, Nomor\_pin);**
- Untuk memberikan logika LOW gunakan perintah berikut:  
**DrvGPIO\_ClrBit>Nama\_port, Nomor\_pin);**
- Untuk membuat delay bisa menggunakan perintah berikut:  
**DrvSys\_Delay(Lama\_delay\_dalam\_cycle);**
- Untuk bisa menggunakan perintah di atas, perlu memasukkan library  
**#include "DrvGPIO.h"**  
**#include "DrvSYS.h"**

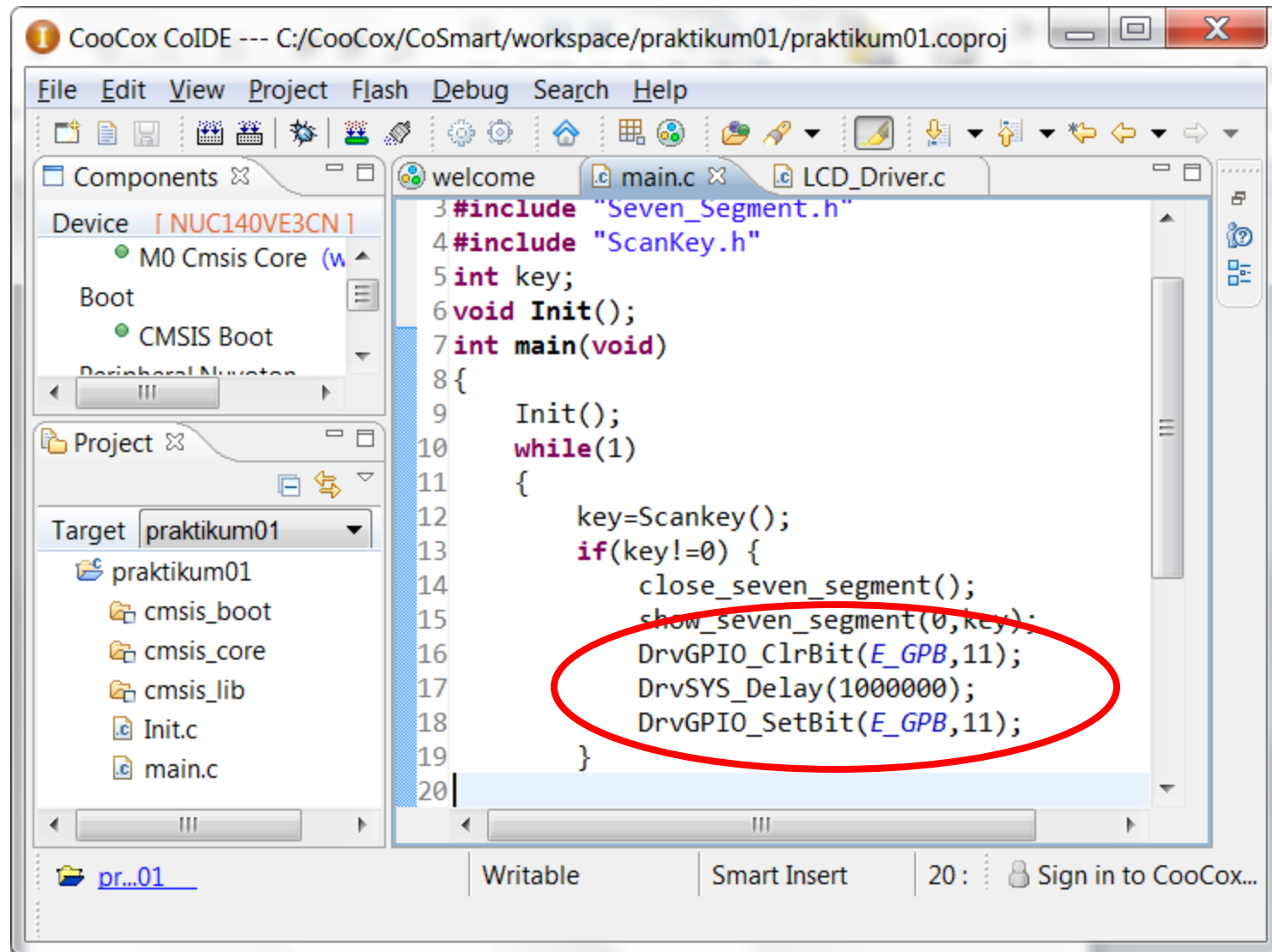
**Nama\_port** diisi E\_GPA / E\_GPB / E\_GPC / E\_GPD / E\_GPE. Dalam hal ini karena BUZZER dihubungkan ke PORT B maka isi dengan **E\_GPB**.

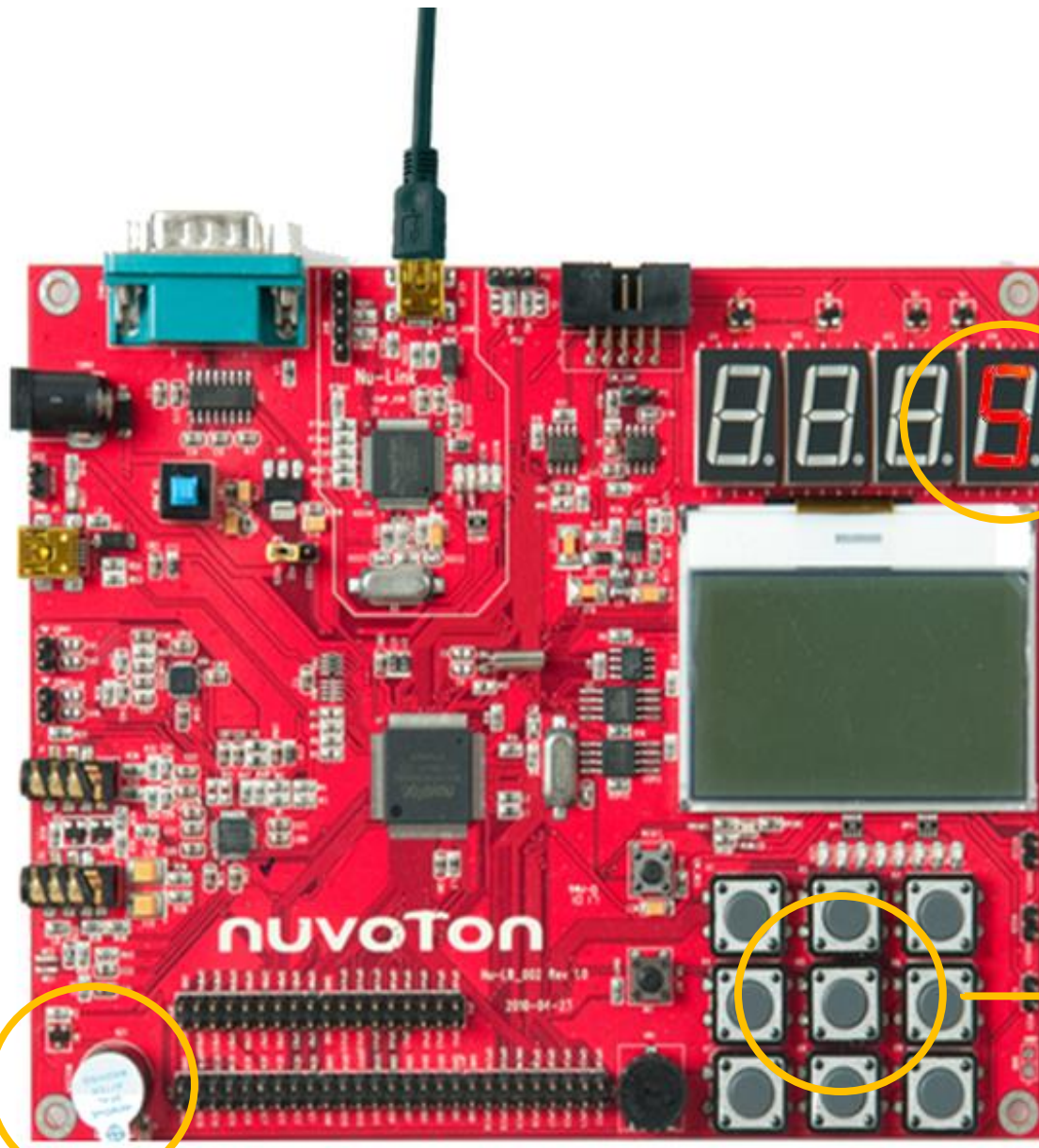
**Nomor\_pin** diisi dengan angka pin kaki mikrokontroler. Dalam hal ini BUZZER dihubungkan ke pin **11**.



Untuk **BUZZER** berbunyi ketika tombol keypad ditekan, maka tambahkan baris program sehingga menjadi seperti gambar di samping.

Build & Download ke chip. Pastikan tidak ada error.



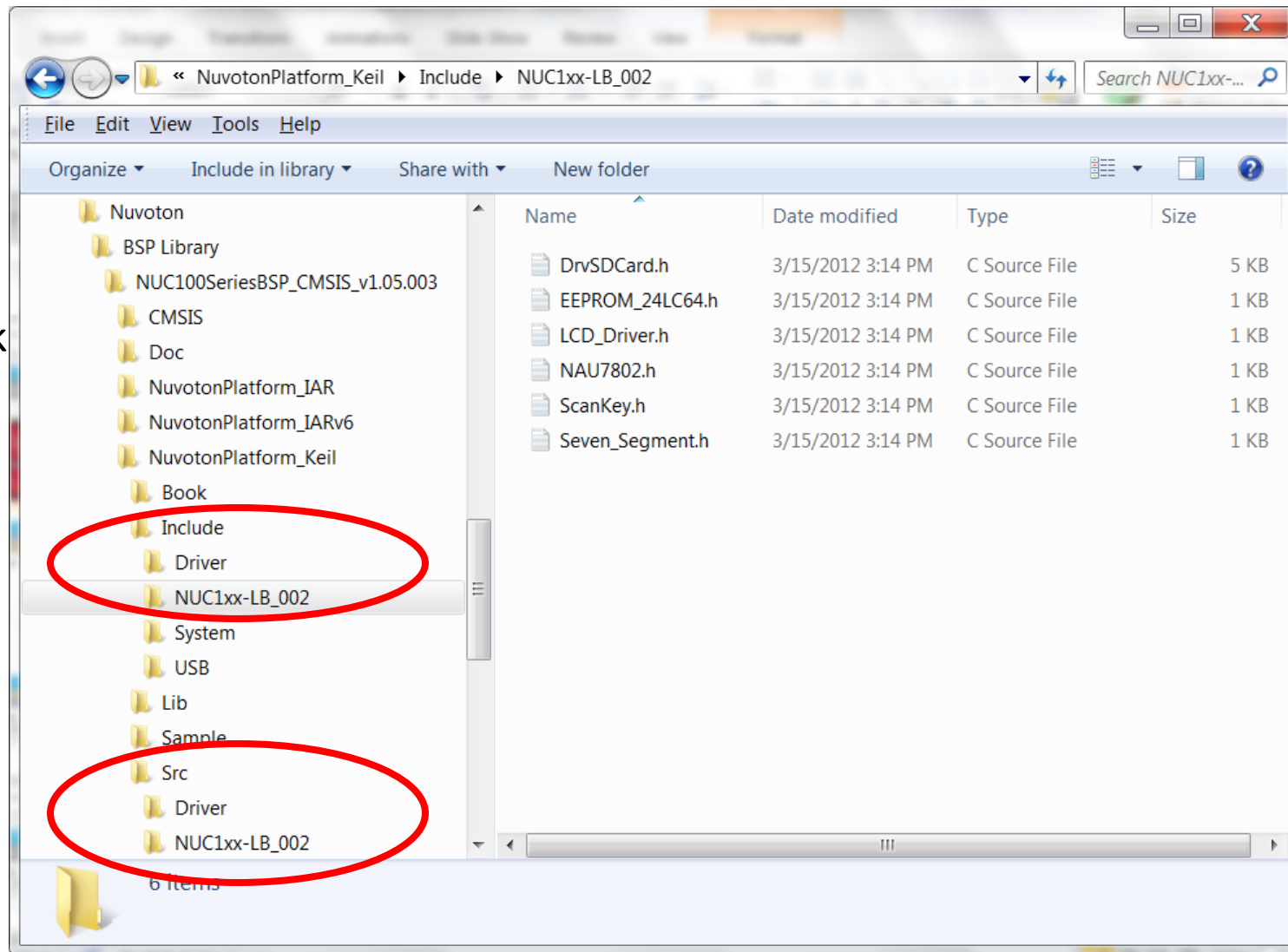


BUZZER ikut memberikan indikator bunyi setiap kali penekanan tombol keypad

# 11. PEMROGRAMAN LCD MATRIX DISPLAY

Nuvoton telah menyediakan **Board Support Package** (yang sudah di-instal) yang berisi banyak library yang dapat mempermudah kita memprogram object-object di Learning Board.

Untuk LCD Matrix Display bisa coba pakai library dari BSP.

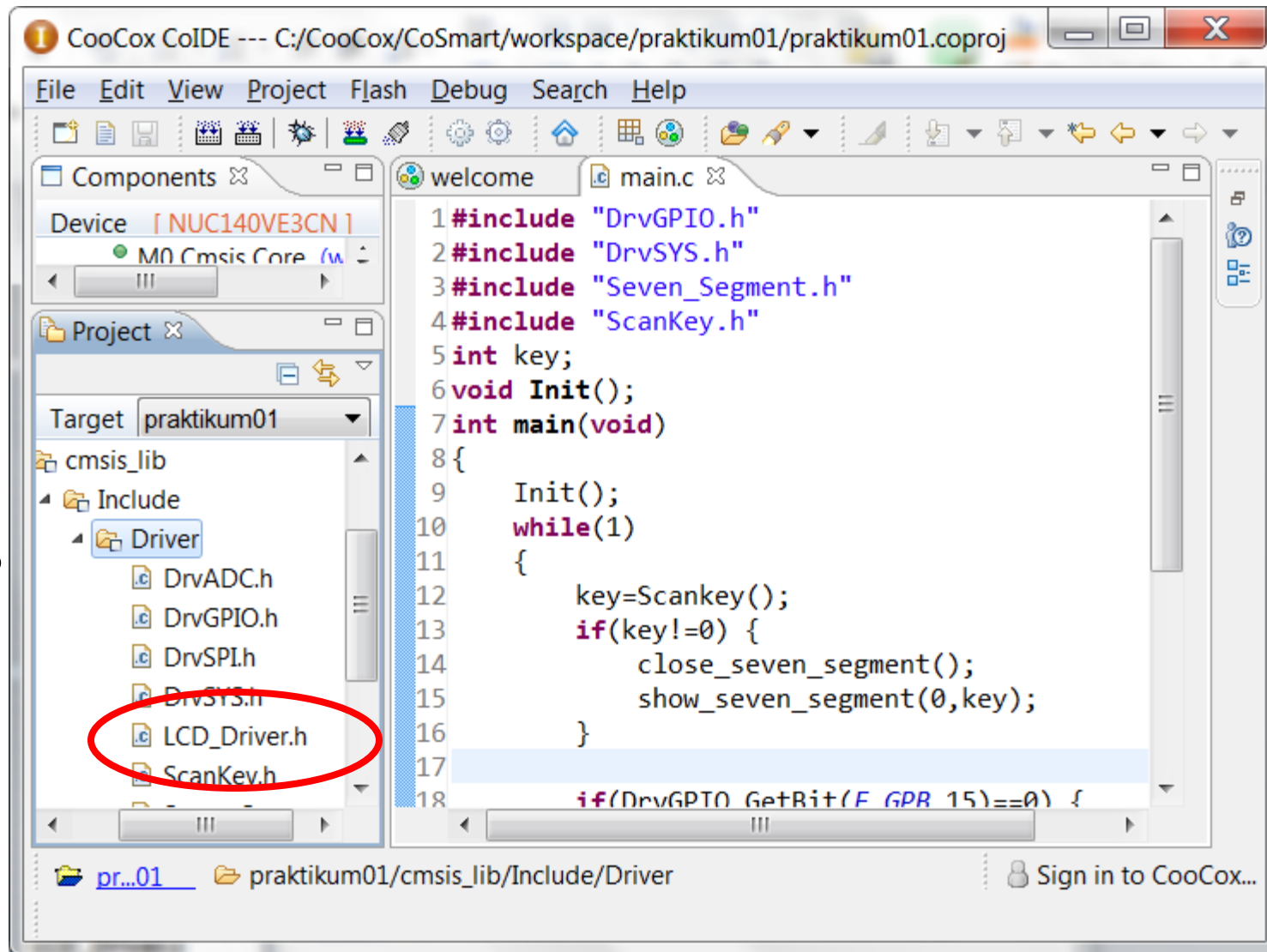


**Langkah 1:**

Klik kanan di  
 cmsis\_lib\Include  
 \Driver, lalu pilih  
**Add Files.**

Browse ke  
 C:\Nuvoton  
 \BSP Library  
 \NUC100SeriesBSP  
 \NuvPlatform\_Keil  
 \Include  
 \NUC1xx-LB\_002

Pilih file  
**LCD\_Driver.h**

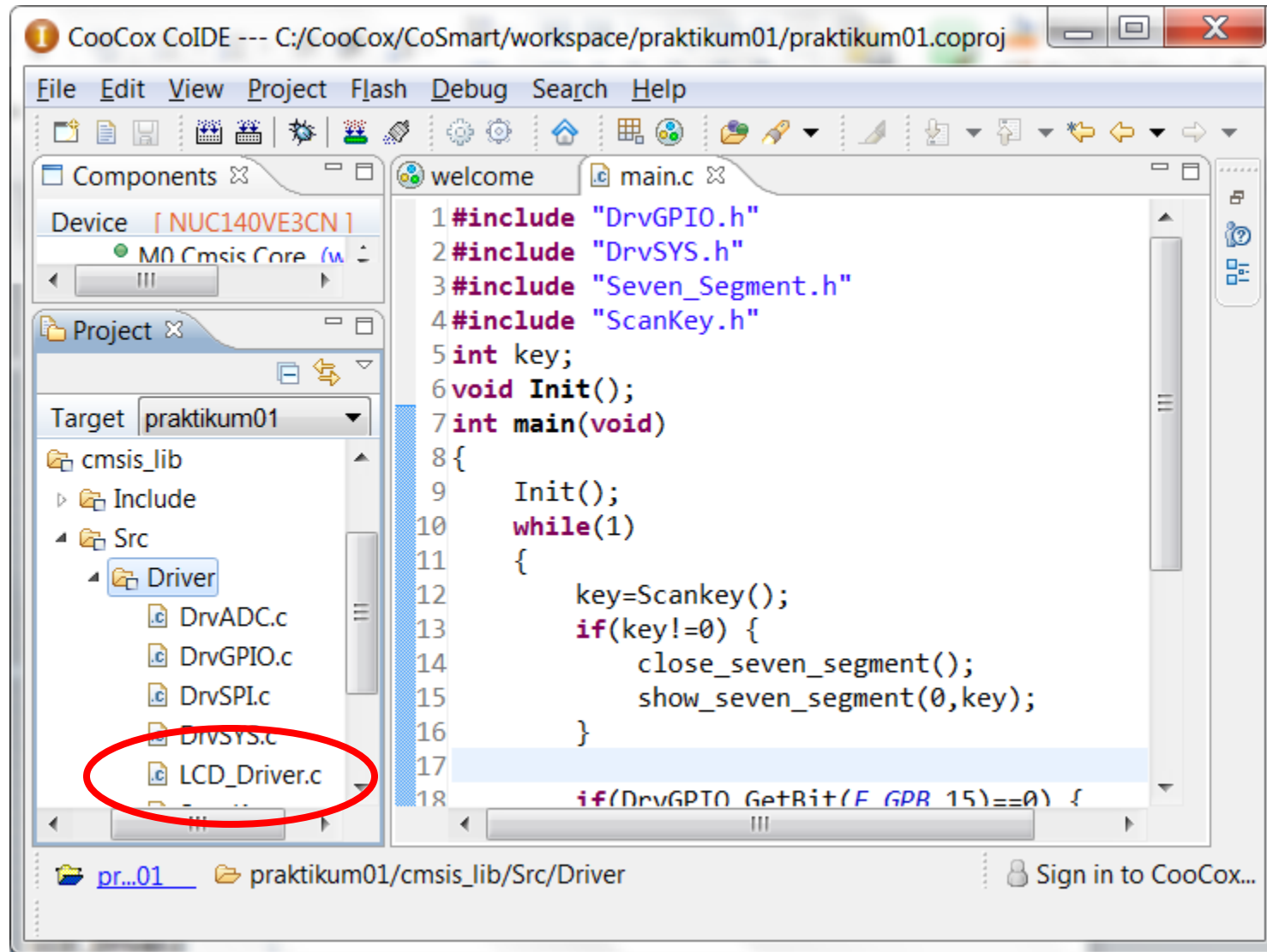


**Langkah 2:**

Klik kanan di  
 cmsis\_lib\Src  
 \Driver, lalu pilih  
**Add Files.**

Browse ke  
 C:\Nuvoton  
 \BSP Library  
 \NUC100SeriesBSP  
 \NuvPlatform\_Keil  
 \Src  
 \NUC1xx-LB\_002

Pilih file  
**LCD\_Driver.c**



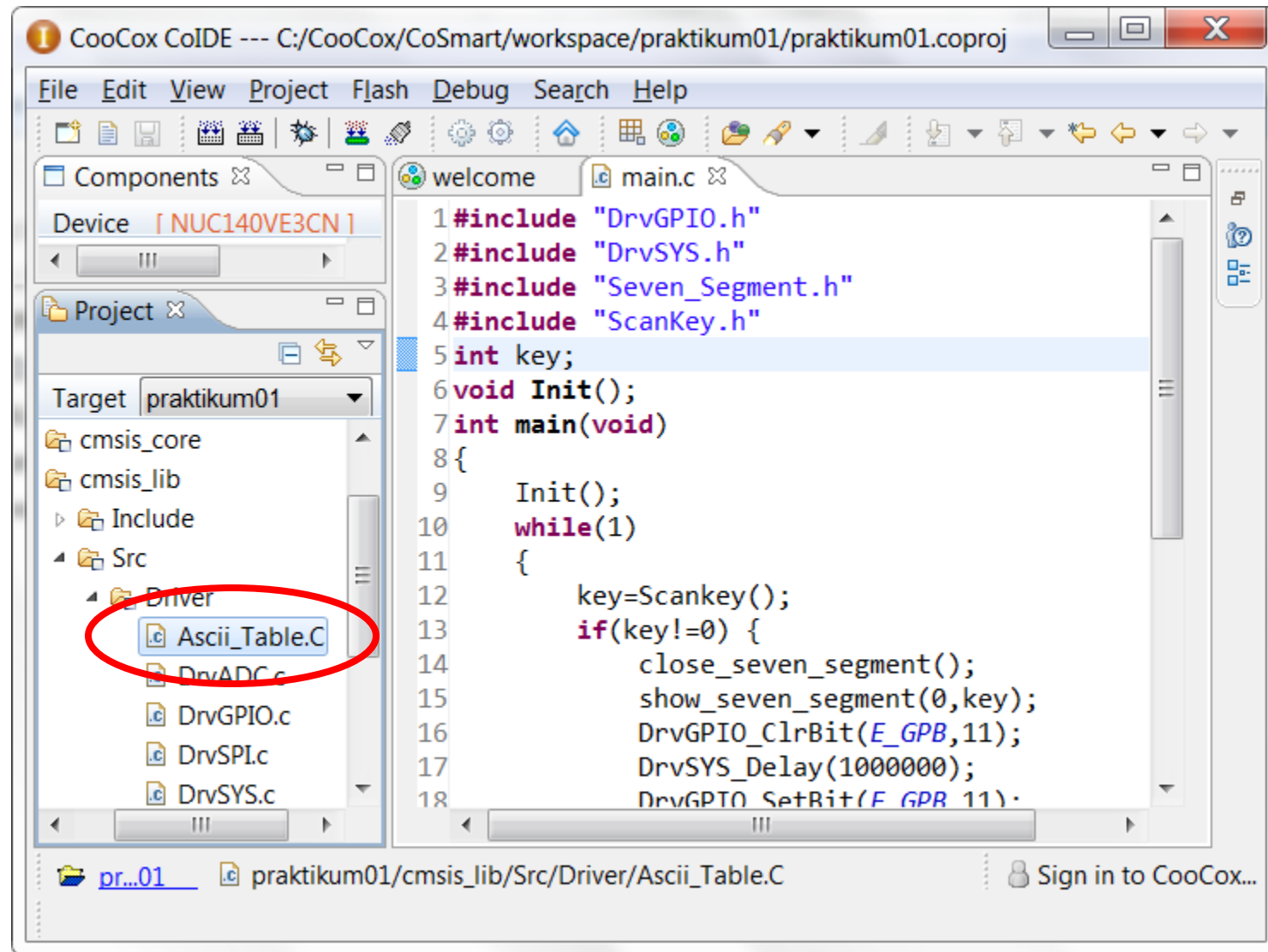


**Langkah 3:**

Klik kanan di  
 cmsis\_lib\Src  
 \Driver, lalu pilih  
**Add Files.**

Browse ke  
 C:\Nuvoton  
 \BSP Library  
 \NUC100SeriesBSP  
 \NuvPlatform\_Keil  
 \Src  
 \NUC1xx-LB\_002

Pilih file  
**Ascii\_Table.c**



**LCD Matrix Display** pada rangkaian hardware dikontrol secara SPI, dan memiliki **pin kontrol backlight di port D pin 14** yang bersifat **ACTIVE LOW**, yaitu menyala jika diberi logika LOW, dan mati jika diberikan logika HIGH.

- Untuk menulis karakter di LCD, bisa menggunakan perintah berikut:  
**print\_lcd(Nomor\_baris,Tulisan);**  
**Show\_Word(Nomor\_baris,Nomor\_kolom,Huruf);**
- Untuk bisa menggunakan perintah di atas, perlu memasukkan library  
**#include "LCD\_Driver.h"**
- Driver LCD\_Driver membutuhkan file Ascii\_Table.c untuk menggambar huruf
- Saat init pertama kali, perlu dipanggil perintah berikut:  
**Initialize\_panel();**
- Untuk membersihkan display, bisa menggunakan perintah berikut:  
**clr\_all\_pannel();**

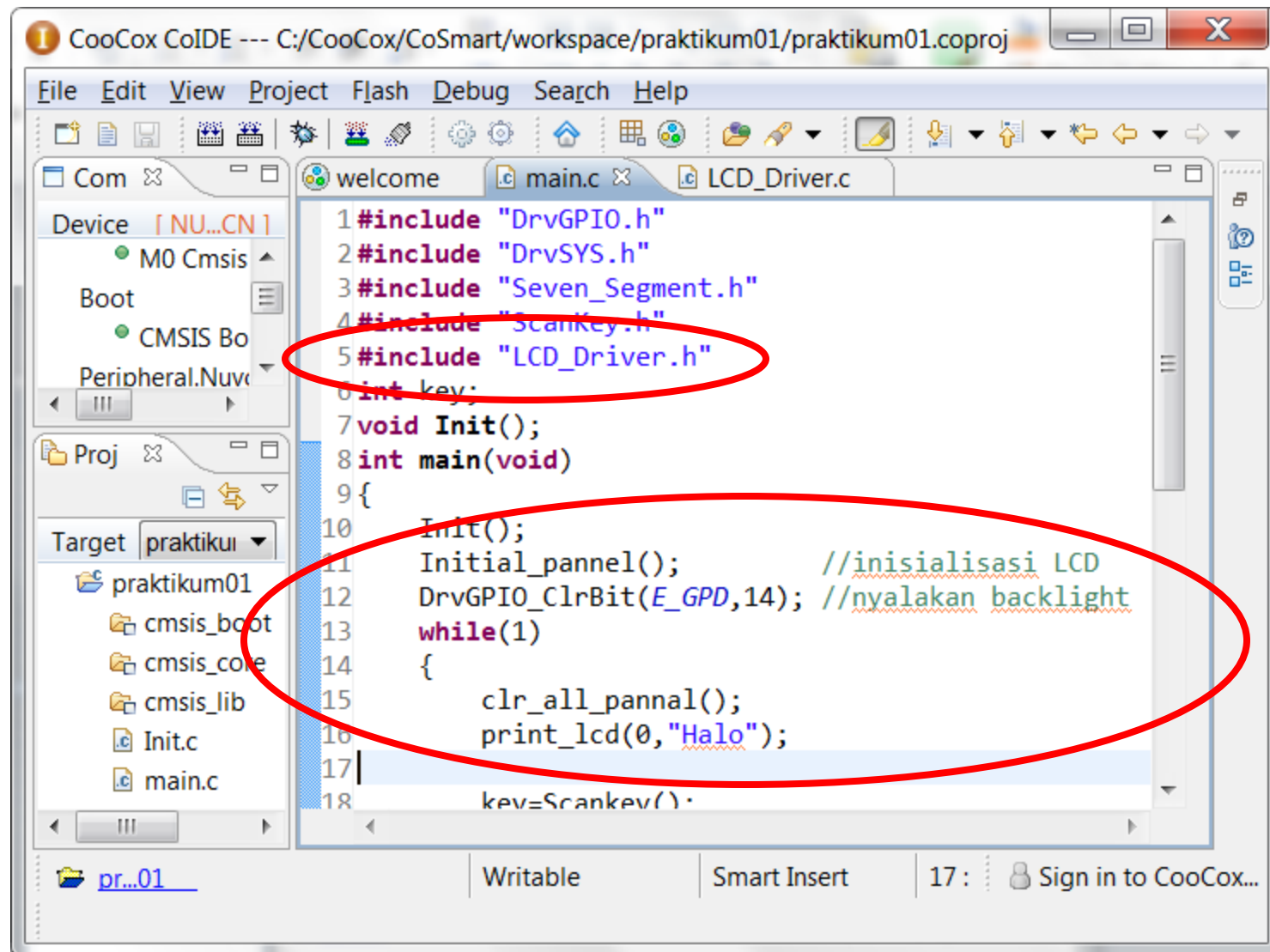
**Keterangan:**

**Nomor\_baris** diisi angka 0-3, dihitung dari paling atas.

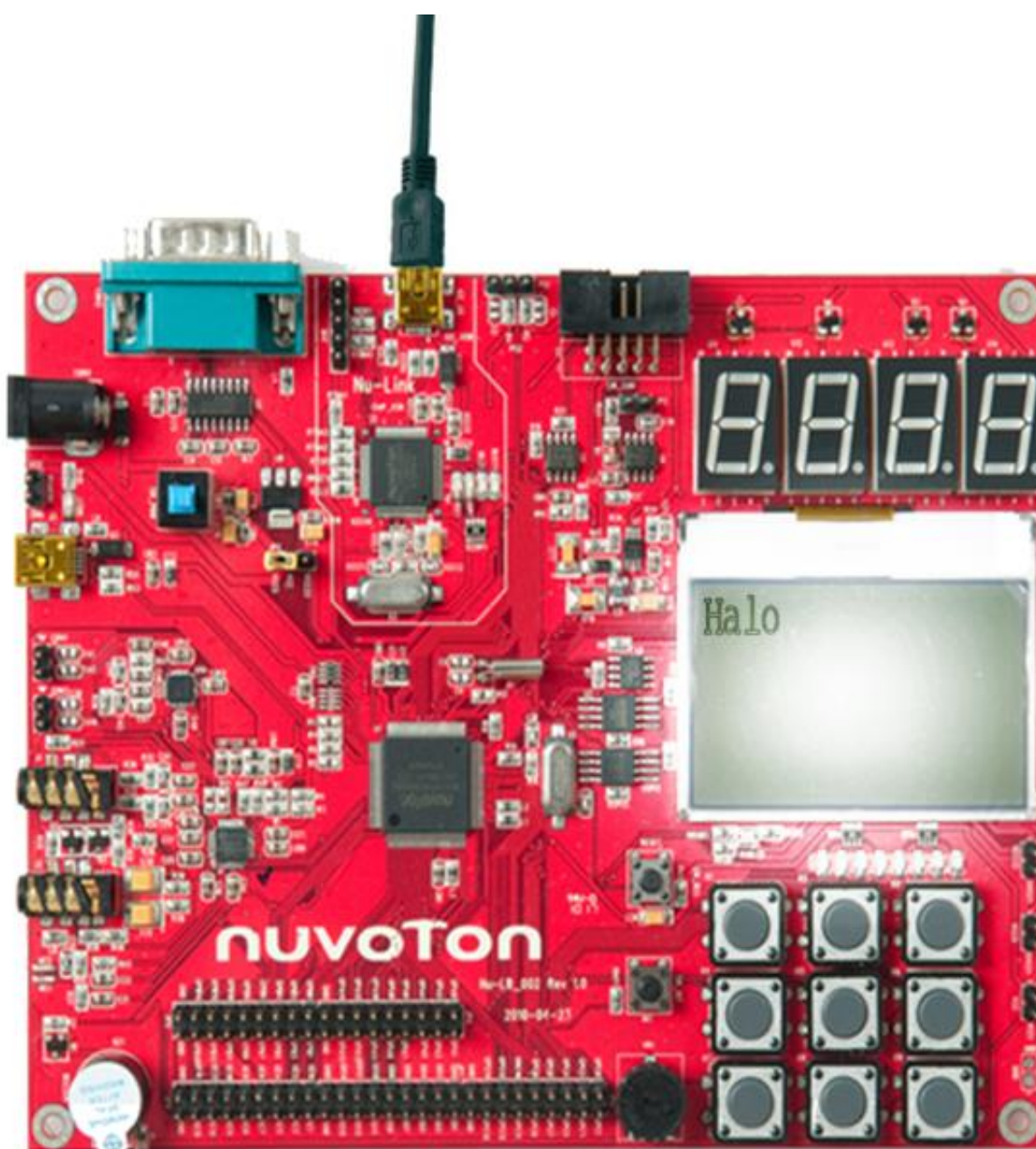
**Nomor\_kolom** diisi angka 0-15, dihitung dari paling kiri.

Untuk menampilkan tulisan "Halo" di baris pertama, tambahkan baris program sehingga menjadi seperti gambar di samping.

Build & Download ke chip. Pastikan tidak ada error.







LCD Matrix menampilkan tulisan “Halo” di baris pertama.

## 12. PEMROGRAMAN ANALOG VARIABEL RESISTANCE

**Variabel Resistor** pada rangkaian hardware yang memberikan masukan **sinyal analog**. ARM NUC140 memiliki 8 channel pengkonversi sinyal analog ke nilai digital (ADC) dengan ketelitian 12 bit, yang dapat membaca level sinyal analog menjadi angka dalam range 0-4095. Clock ADC dapat menggunakan internal 22MHz dengan divisor 2.

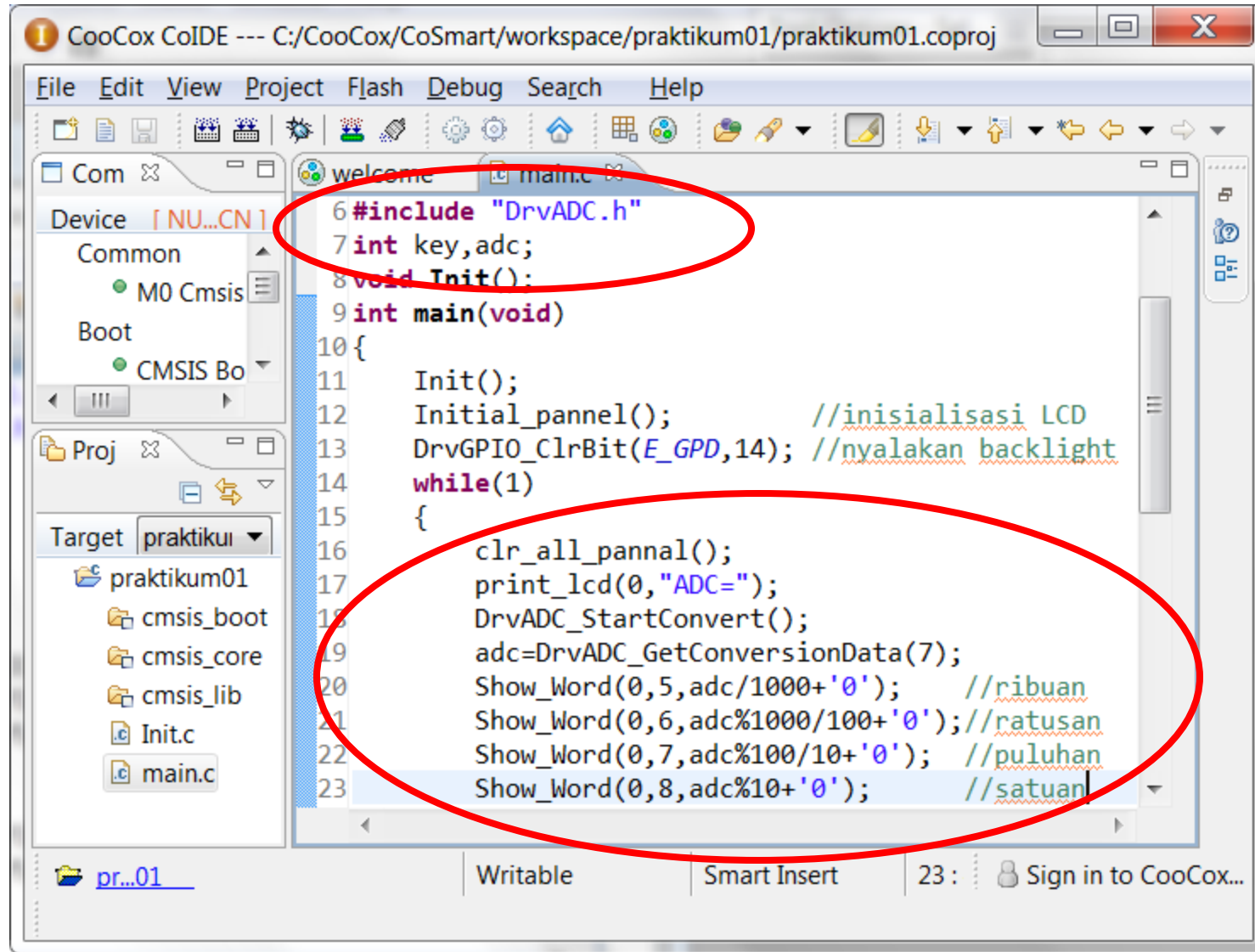
- Untuk membaca sinyal analog, bisa menggunakan perintah berikut:  
**DrvADC\_StartConvert();**  
**DrvADC\_GetConversionData(Nomor\_channel);**
- Untuk bisa menggunakan perintah di atas, perlu memasukkan library **#include "DrvADC.h"**
- Untuk menampung hasil pembacaan ADC, ada baiknya membuat suatu variabel, misal bernama adc bertipe data integer.

### Keterangan:

Nomor\_channel diisi angka 0-7, sesuai dengan pin tempat sinyal analog terhubung. Dalam hal ini, Variabel Resistor ada di channel **7**.

Untuk membaca sinyal analog dan menampilkan ke LCD matrix, ubah baris program sehingga menjadi seperti gambar di samping.

Build & Download ke chip. Pastikan tidak ada error.

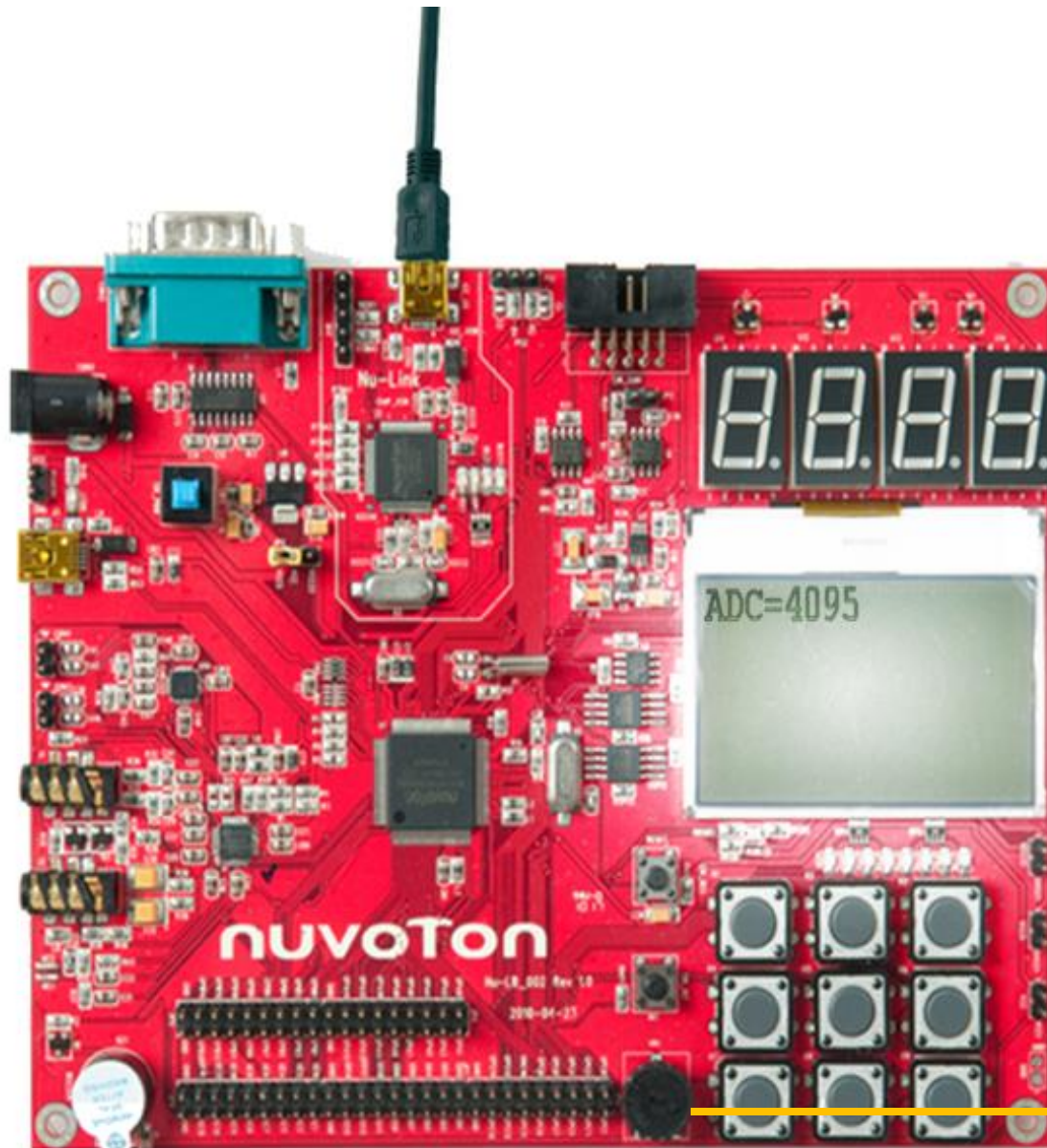


```

CooCox CoIDE --- C:/CooCox/CoSmart/workspace/praktikum01/praktikum01.coproj
File Edit View Project Flash Debug Search Help

Device [NU...CN1]
  Common
    M0 Cmsis
  Boot
    CMSIS Bo
  Proj
  Target praktikui
    praktikum01
      cmsis_boot
      cmsis_core
      cmsis_lib
      Init.c
      main.c

6 #include "DrvADC.h"
7 int key, adc;
8 void Init():
9 int main(void)
10 {
11     Init();
12     Initial_pannel(); //inisialisasi LCD
13     DrvGPIO_ClrBit(E_GPD,14); //nyalakan backlight
14     while(1)
15     {
16         clr_all_pannal();
17         print_lcd(0,"ADC=");
18         DrvADC_StartConvert();
19         adc=DrvADC_GetConversionData(7);
20         Show_Word(0,5,adc/1000+'0'); //ribuan
21         Show_Word(0,6,adc%1000/100+'0'); //ratusan
22         Show_Word(0,7,adc%100/10+'0'); //puluhan
23         Show_Word(0,8,adc%10+'0'); //satuan
  
```



LCD Matrix menampilkan hasil angka konversi ADC dari putaran Variabel Resistance.



# REFERENSI

- [1] Nuvoton. *NuMicro Training Course*. Nuvoton Technology Corporation, 2012.
- [2] Yiu, Joseph. *The Definitive Guide to the ARM Cortex-M0*. Elsevier Inc, 2011.
- [3] ARM. *Cortex-M0 Technical Reference Manual*. ARM Holding Limited, 2009.
- [4] Nuvoton. *NuMicro NUC140 Data Sheet EN V3.02*. Nuvoton Technology Corporation, 2012.
- [5] Nuvoton. *Nu-LB-NUC140 User's Guide v2.0*. Nuvoton Technology Corporation, 2010.
- [6] CooCox. 2013. *CoIDE User Guide*. [http://CooCox.org/CooCox\\_CoIDE.htm](http://CooCox.org/CooCox_CoIDE.htm). 1 Januari 2014.

**Terima kasih telah mempelajari.**

**Silakan belajar lebih lanjut & berkreasi dengan Mikrokontroler ARM.**



**PRASIMAX MIKRON**

**Lab & Training Center**

**Jl. Margonda Raya No.494D Lantai 3 Depok 16424**

**Email: [informasi@mikron123.com](mailto:informasi@mikron123.com) | Web: [www.mikron123.com](http://www.mikron123.com)**

**Telpon: 021-78880672 | SMS Hotline: 08121092300**