# BAB V Timer pada Nuvoton NUC140

## Tujuan

1. Praktikan dapat memahami fungsi dan cara kerja timer pada mikrokontroler NUC140.
2. Praktikan memiliki kemampuan untuk membuat program dasar yang memanfaatkan fitur timer pada board Nuvoton NUC140.
3. Praktikan dapat mengaplikasikan penggunaan timer untuk mendukung kinerja sistem berbasis mikrokontroler.
4. Praktikan mengetahui bagaimana timer digunakan dalam berbagai situasi seperti pengukuran waktu dan delay sistem.
5. Praktikan memperoleh pengalaman praktis dalam mengonfigurasi dan memanfaatkan timer melalui pemrograman langsung pada perangkat NUC140.

## Dasar Teori

### ARM Nuvoton NUC140

ARM Nuvoton NUC140 adalah sebuah mikrokontroler berbasis arsitektur ARM Cortex-M0 yang dirancang untuk kebutuhan sistem tertanam dengan efisiensi tinggi. Mikrokontroler ini memiliki kecepatan clock hingga 50 MHz dan mendukung memori Flash sebesar 32 KB hingga 128 KB, serta SRAM antara 4 KB hingga 16 KB. Selain itu, NUC140 dilengkapi dengan berbagai peripheral seperti GPIO (General Purpose Input Output), Timer, Watchdog Timer, Real Time Clock (RTC), UART, SPI, I²C, CAN Bus, PWM, ADC 12-bit, serta USB 2.0 FS. Dengan dukungan ISP (In-System Programming), mikrokontroler ini memudahkan proses pemrograman ulang tanpa perlu perangkat eksternal tambahan. Mikrokontroler ini sangat cocok untuk aplikasi industri, perangkat otomasi, serta sistem pembelajaran mikrokontroler tingkat lanjut.

Sumber : (<https://www.nuvoton.com>)

### Coocox CoIDE

Coocox CoIDE adalah sebuah Integrated Development Environment (IDE) gratis yang dirancang khusus untuk pengembangan mikrokontroler berbasis ARM Cortex-M0, M3, dan M4. IDE ini memiliki fitur seperti project manager, editor kode, debugger, dan compiler ARM GCC yang terintegrasi, sehingga memudahkan praktikan untuk menulis, mengompilasi, dan memprogram mikrokontroler secara efisien. CoIDE juga mendukung drag-and-drop library serta integrasi langsung dengan alat bantu seperti CoSmart, sehingga mempercepat proses pembuatan proyek mikrokontroler. Pengguna dapat melakukan debugging langsung ke board melalui fitur debug yang disediakan.

Sumber : (<https://www.coocox.org/>)

### Coocox CoSmart

CoSmart adalah perangkat lunak pendukung dari Coocox CoIDE yang digunakan untuk melakukan konfigurasi pin mikrokontroler berbasis ARM. Melalui antarmuka grafis, pengguna dapat memilih jenis chip, mengatur pin apa saja yang aktif (Enable), menetapkan fungsinya (GPIO, SPI, Timer, PWM, dll), serta langsung menghasilkan file proyek yang siap digunakan di CoIDE. CoSmart sangat berguna untuk menghindari konflik pin dan mempercepat proses setup awal dalam pengembangan aplikasi mikrokontroler. CoSmart juga memberikan visualisasi yang memudahkan pengguna dalam memahami layout pin pada chip yang digunakan.

Sumber : (<https://www.coocox.org/>)

### Seven Segment

Seven segment display merupakan modul tampilan digital yang terdiri dari tujuh elemen LED (segmen) yang disusun menyerupai angka 8. Setiap segmen dapat dinyalakan atau dimatikan untuk membentuk angka 0 hingga 9, dan beberapa huruf tertentu. Modul ini sangat populer karena cara penggunaannya yang relatif mudah dan sangat efisien untuk menampilkan angka. Dalam konteks pembelajaran mikrokontroler, seven segment digunakan untuk menampilkan hasil perhitungan, waktu, penghitung (counter), atau keluaran dari sistem timer. Terdapat dua jenis konfigurasi umum pada seven segment: common anode dan common cathode, yang menentukan logika pengendaliannya.

Sumber :(<https://www.electronics-tutorials.ws/>)

### LED dan LCD

LED adalah komponen elektronik yang berfungsi sebagai pemancar cahaya ketika diberi arus listrik. Dibandingkan lampu biasa, LED memiliki efisiensi yang tinggi, umur panjang, dan respons yang cepat. Dalam praktik mikrokontroler, LED sering digunakan sebagai indikator status — misalnya untuk menunjukkan sistem aktif, hasil logika keluaran, atau merespon interupsi. LED juga kerap dipakai dalam eksperimen awal sebagai output sederhana yang memperlihatkan bahwa suatu proses berjalan sesuai program. Penggunaan LED menjadi dasar yang penting dalam memahami hubungan antara perangkat lunak dan perangkat keras dalam sistem tertanam.

Sumber : (<https://www.allaboutcircuits.com/>)

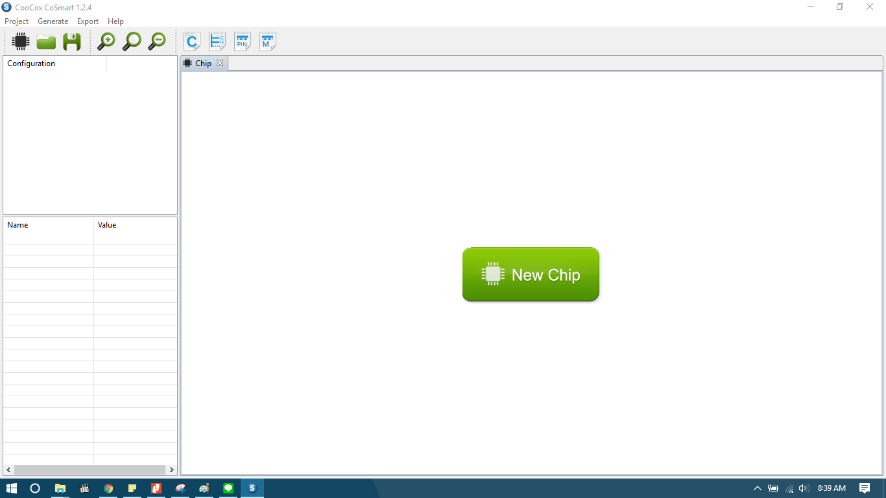
LCD Dot Matriks 128 x 64 merupakan layar grafis berbasis teknologi Liquid Crystal Display yang terdiri dari 128 kolom dan 64 baris piksel. Setiap piksel pada layar ini dapat dikontrol secara individual, memungkinkan pengguna untuk menampilkan teks, grafik, diagram, hingga animasi sederhana. Modul ini biasanya menggunakan pengontrol seperti KS0108 atau ST7920 yang menerima perintah digital dari mikrokontroler. Dalam konteks praktikum mikrokontroler, LCD ini sangat berguna untuk menampilkan status sistem, keluaran sensor, hasil pengolahan data, atau pesan dari user interface. Kemampuan tampilan yang lebih fleksibel dibandingkan seven segment menjadikannya pilihan utama dalam pengembangan prototipe sistem tertanam interaktif.

Sumber : (<https://www.electronicwings.com/>)

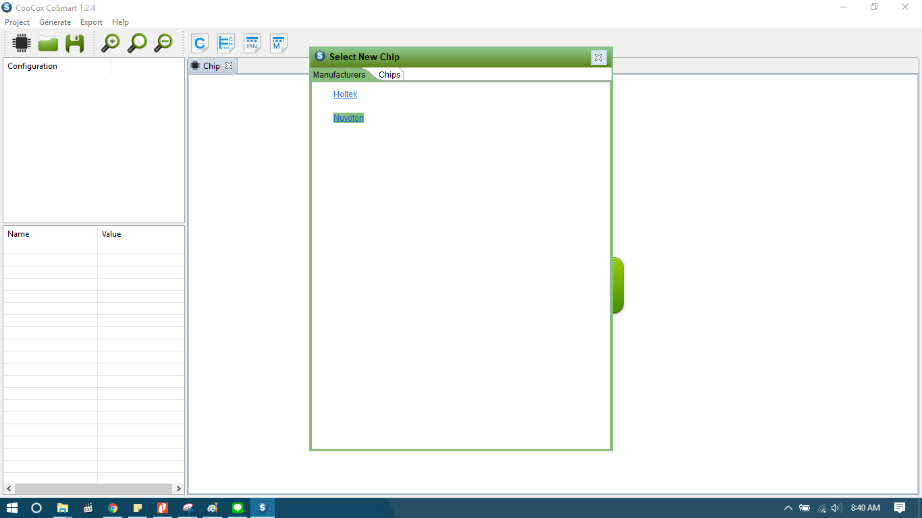
## Langkah Kerja

### Percobaan 1

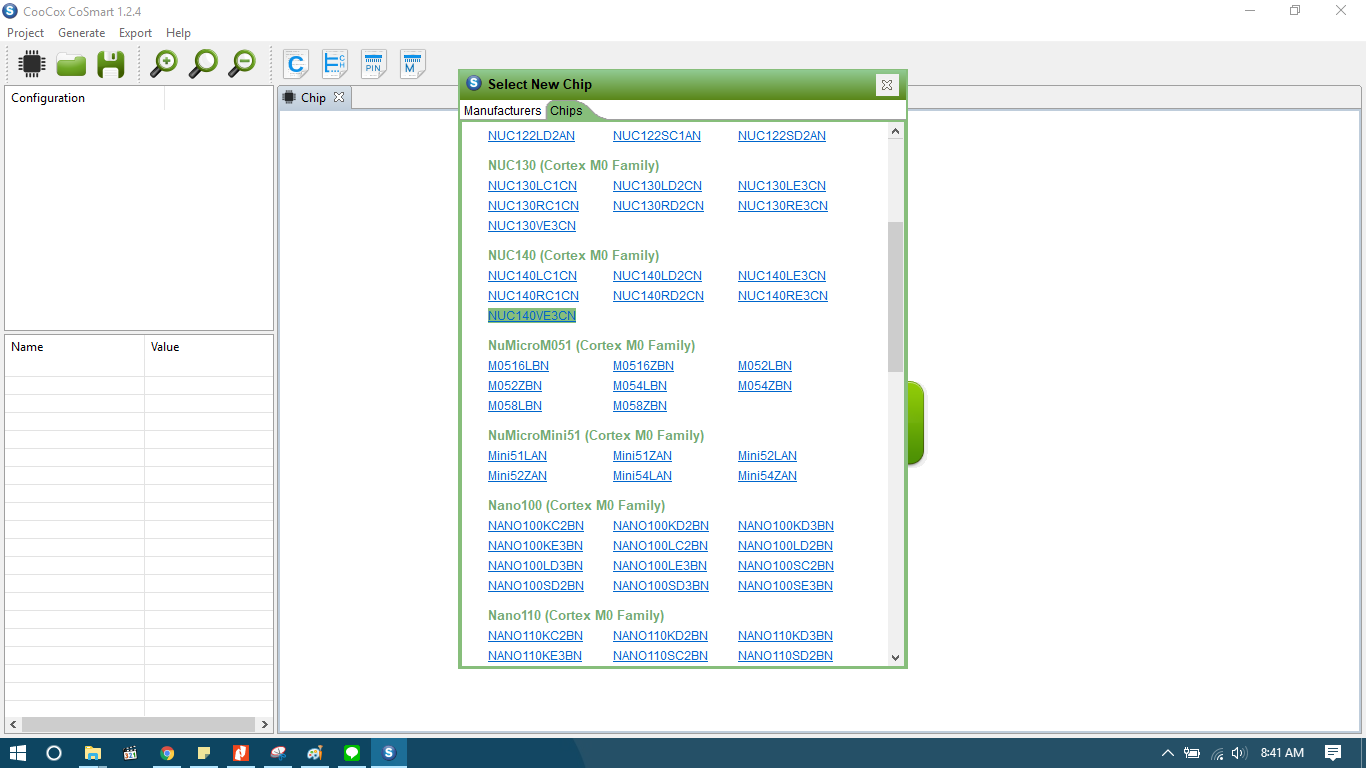
1. Sambungkan kabel USB Nu-Link ke port ICE Bridge pada board Nuvoton NUC140, lalu colokkan ujung satunya ke port USB laptop.
2. Buka aplikasi CoSmart, lalu klik New Chip untuk memulai konfigurasi chip baru.



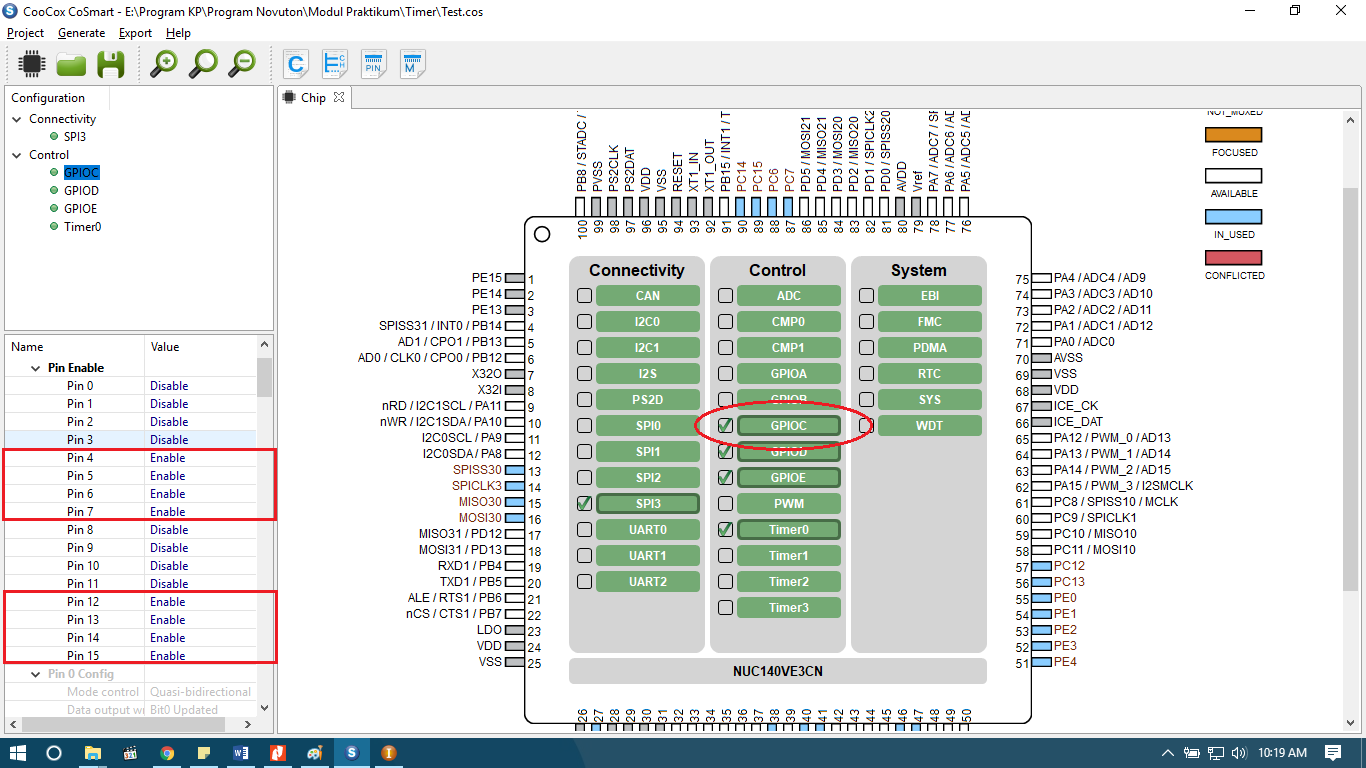
1. Pada pilihan vendor chip, pilih Nuvoton.



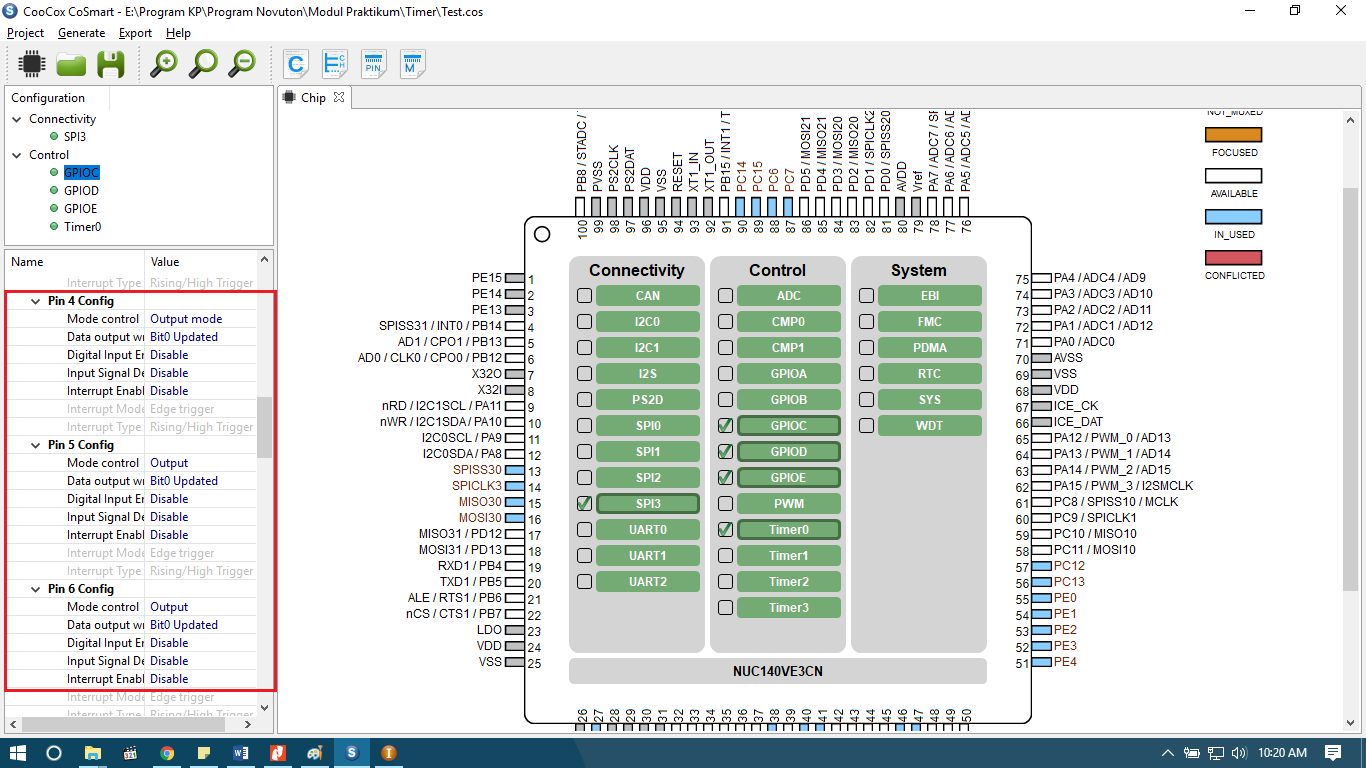
1. Selanjutnya pilih tipe chip NUC140VE3CN pada daftar yang tersedia.



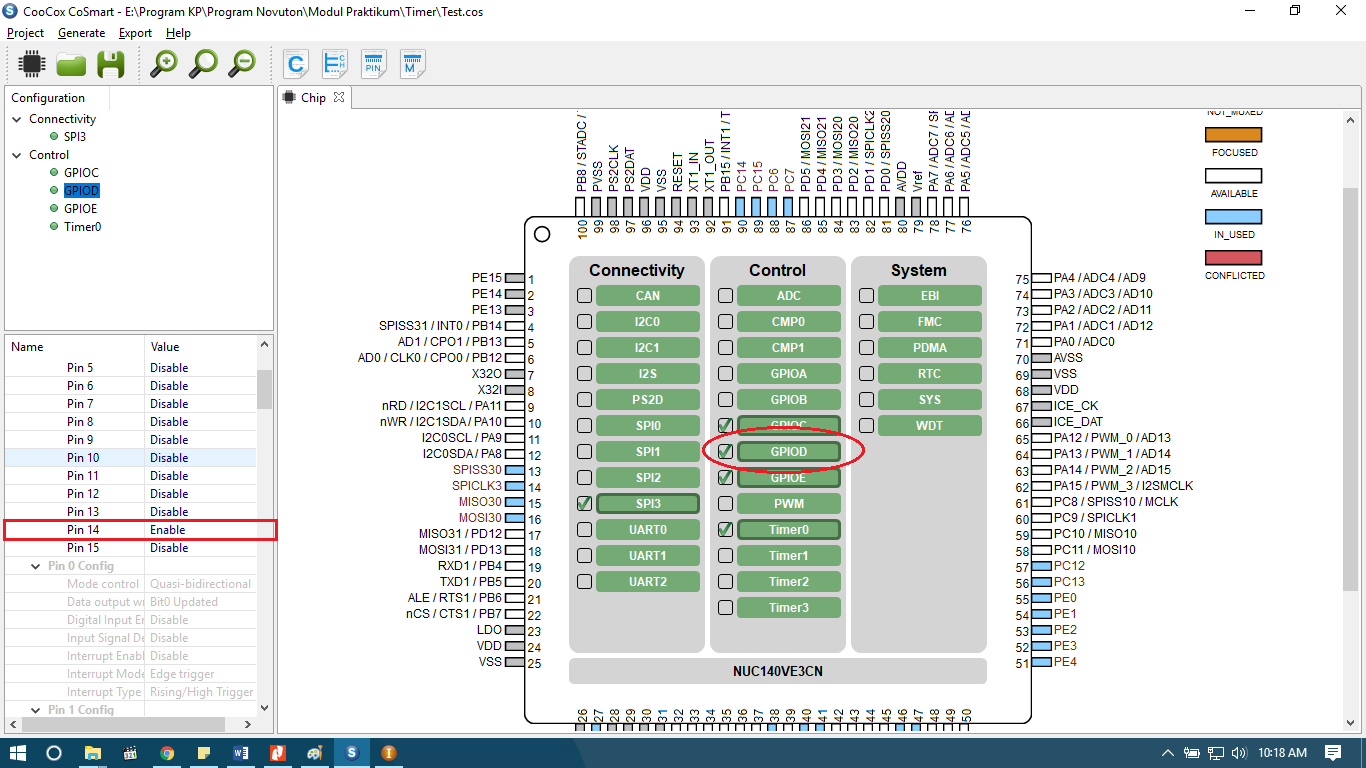
1. Setelah tampilan chip muncul di panel utama, centang komponen yang akan digunakan: GPIOC, GPIOD, GPIOE, SPI3, dan TIMER0.
2. Pada panel kiri, aktifkan (Enable) pin GPIOC 12–15 dan 4–7.



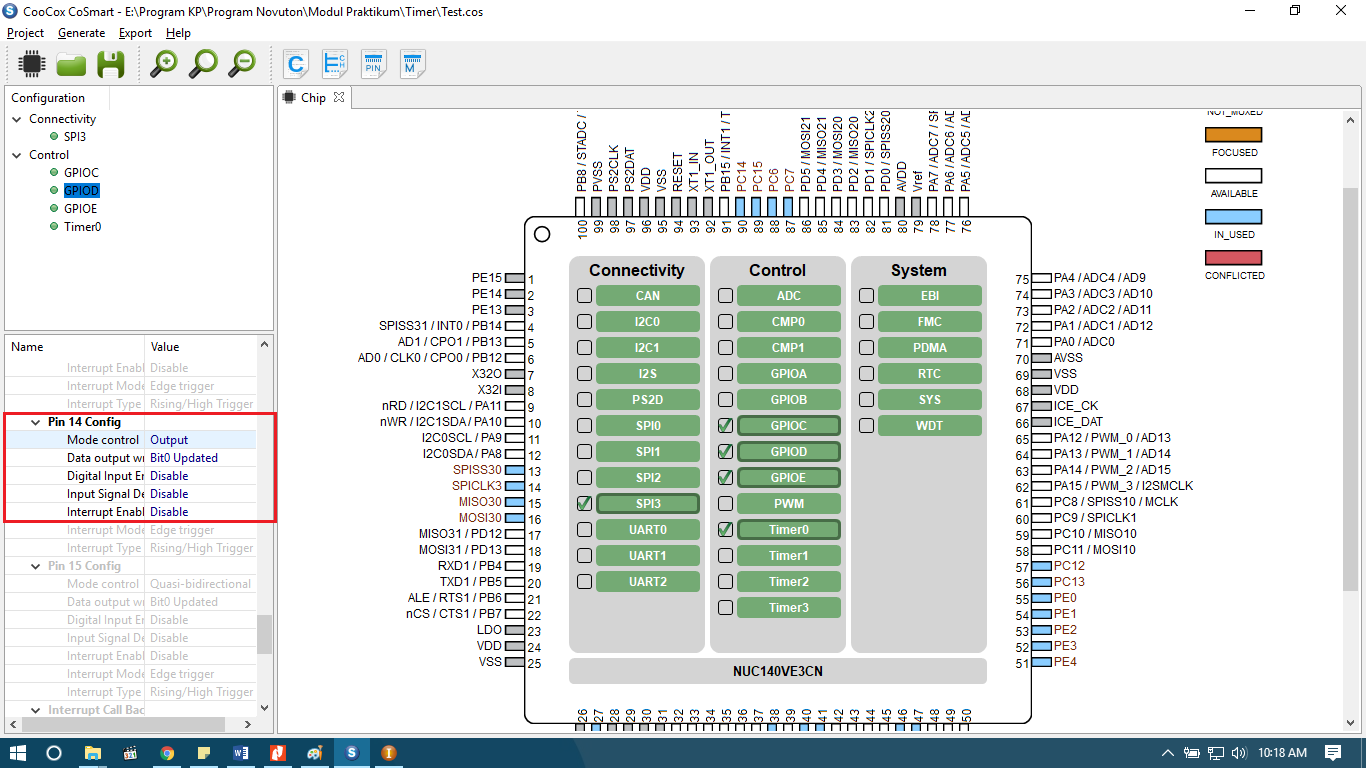
1. Gulir ke bawah, lalu ubah mode pin tersebut menjadi Output.



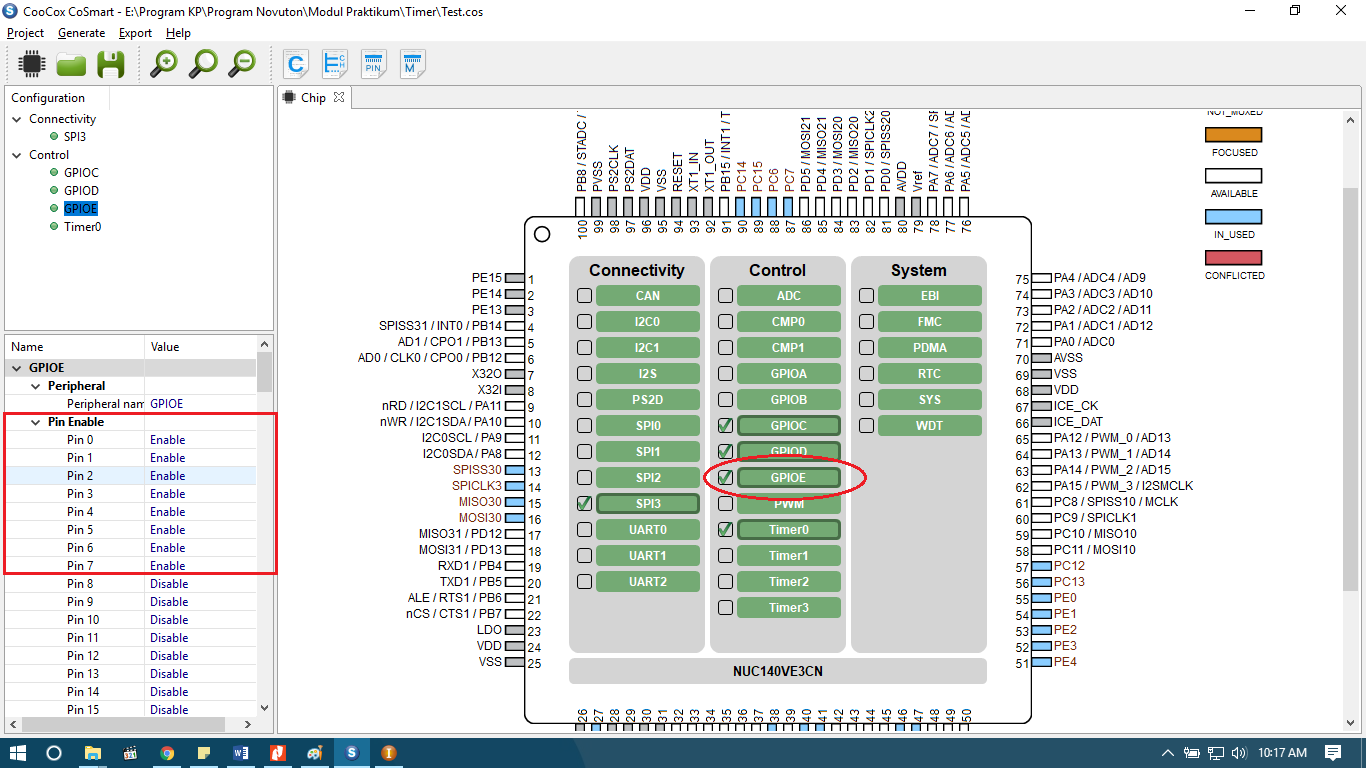
1. Kembali ke panel kiri, ubah pengaturan dari GPIOC menjadi GPIOD, kemudian aktifkan pin 14.



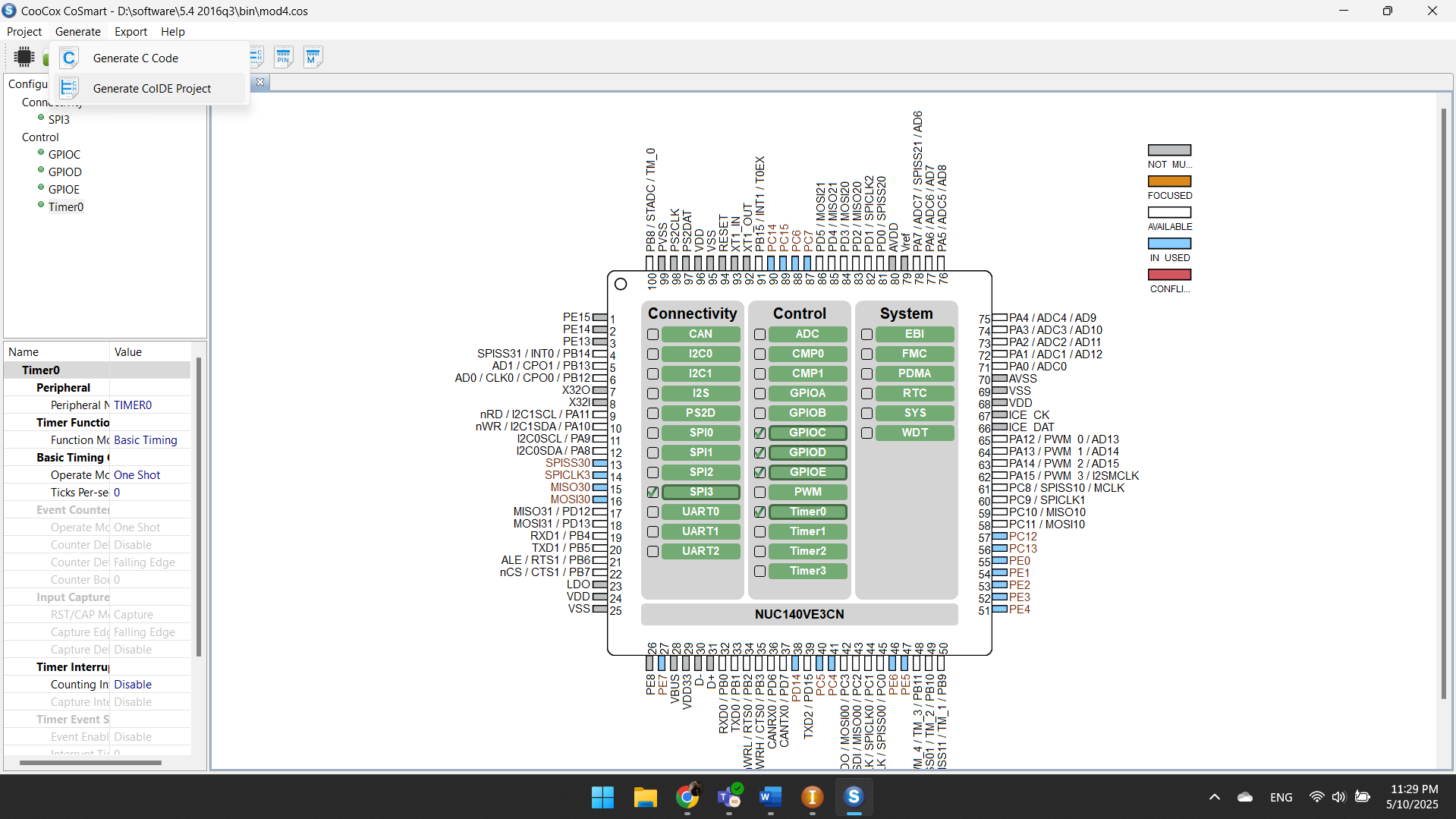
1. Gulir ke bawah dan ubah mode pin GPIOD.14 menjadi Output.



1. Selanjutnya ubah pengaturan ke GPIOE, lalu aktifkan pin 0 sampai 7, dan ubah mode semua pin tersebut menjadi Output.



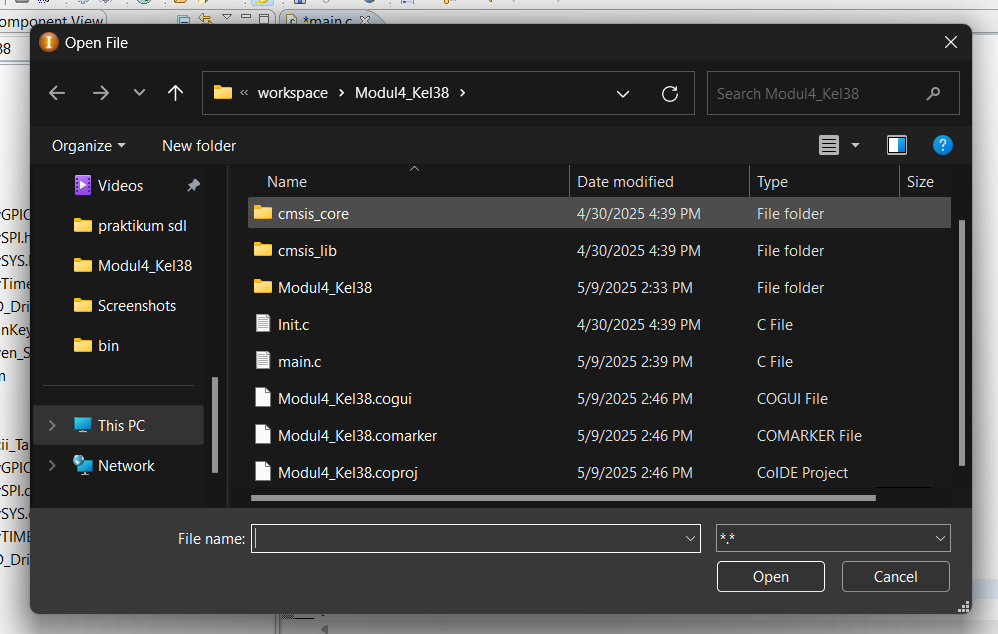
1. Pastikan seluruh pengaturan pin sudah sesuai.
2. Klik Generate, lalu pilih Generate CoIDE Project untuk membuat file proyek yang kompatibel dengan CoIDE.



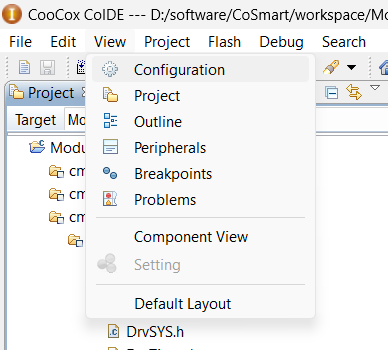
1. Masukkan nama proyek dan pilih lokasi penyimpanan, lalu klik Finish.



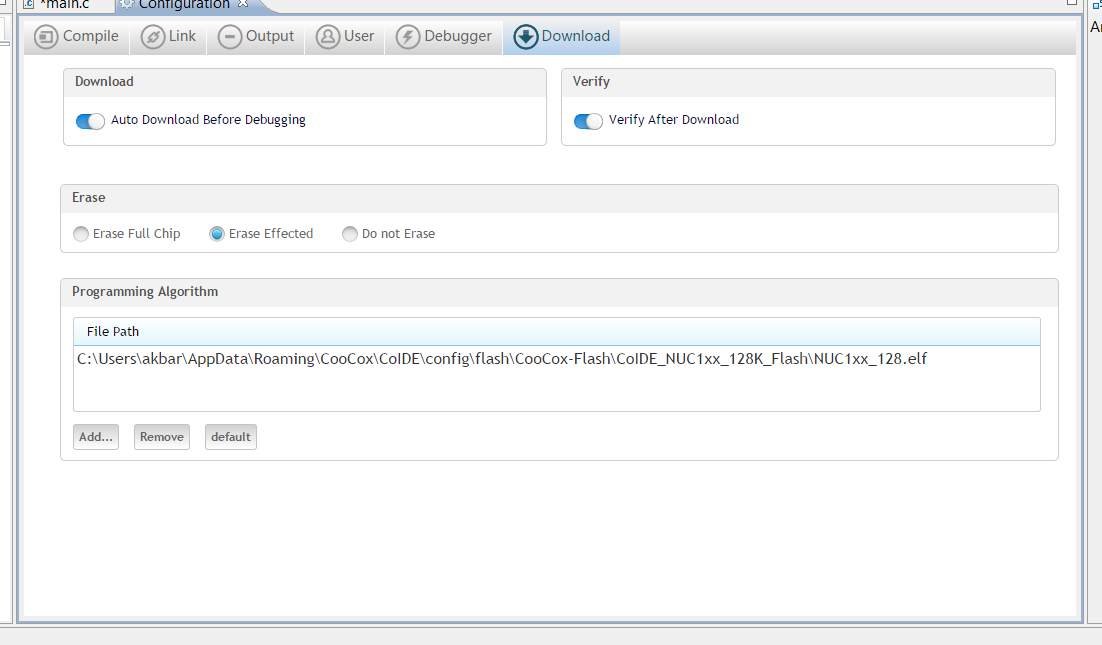
1. Buka aplikasi Coocox CoIDE, lalu pilih menu Project > Open Project, kemudian cari dan buka proyek yang telah dibuat sebelumnya dari CoSmart.



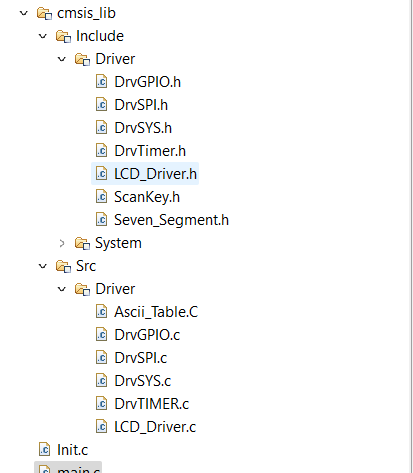
1. Setelah proyek terbuka, masuk ke menu View > Configuration.



1. Pada bagian Download, ubah opsi Erase dari *Erase Effected* menjadi Erase Full Chip.



1. Tambahkan library Seven Segment dan LCD ke dalam proyek. Pastikan semua file dengan ekstensi .c dan .h dimasukkan.



1. Di panel proyek sebelah kiri, buka file Main.c, lalu masukkan source code program yang telah disediakan dalam modul.

|  |
| --- |
| #include "DrvSYS.h"  #include "DrvGPIO.h"  #include "DrvTimer.h"  #include "LCD\_Driver.h"  #include "Seven\_Segment.h"  void delay\_loop(void)  {  uint32\_t i,j;  for(i=0;i<2;i++)  {  for(j=0;j<10000;j++);  }  }  void Timer\_initial(void)  {  SYSCLK->CLKSEL1.TMR0\_S = 0;  SYSCLK->APBCLK.TMR0\_EN = 1;  TIMER0->TCSR.MODE=1;  TIMER0->TCSR.PRESCALE=0;  TIMER0->TCMPR = 2400000;  TIMER0->TCSR.IE = 1;  TIMER0->TISR.TIF = 1;  NVIC\_EnableIRQ(TMR0\_IRQn);  TIMER0->TCSR.CRST = 1;  TIMER0->TCSR.CEN = 1;  TIMER0->TCSR.TDR\_EN = 1;  }  int adc;  int tegangan;  void Init();  int main(void)  {  Init();  int i=0, j=0;  UNLOCKREG();  DrvSYS\_SetOscCtrl(E\_SYS\_XTL12M, 1);  SysTimerDelay(5000);  DrvSYS\_SelectHCLKSource(0);  LOCKREG();  DrvSYS\_SetClockDivider(E\_SYS\_HCLK\_DIV, 0);  Initial\_pannel();  clr\_all\_pannal();  print\_lcd(0, “Florencia Irena”);  print\_lcd(1, “Timer Interrupt”);  print\_lcd(2, “Kelompok XX”);  print\_lcd(3, “Sistem Komputer”);  DrvGPIO\_ClrBit(E\_GPD, 14);  Timer\_initial();  while(1)  {  tegangan=((float)adc/4096)\*5000;  for(i=0;i<9;i++)  {  for(j=0;j<4;j++)  {  show\_seven\_segment(j, i);  delay\_loop();  }  }  }  } |

1. Setelah kode selesai dimasukkan, klik menu Project > Build untuk memproses dan menyusun program.
2. Jika proses build berhasil (Build Success), pilih menu Flash > Program Download untuk mengunggah program ke mikrokontroler.
3. Amati hasil eksekusi program pada board Nuvoton NUC140, termasuk tampilan di LCD, Seven Segment, dan respon terhadap penekanan Switch Button.

## Hasil Percobaan dan Analisis Percobaan

### Percobaan 1





Analisis:

Percobaan ini bertujuan untuk mengimplementasikan Timer pada mikrokontroler Nuvoton NUC140 menggunakan Coocox CoIDE dan CoSmart. Hasilnya menunjukkan bahwa Timer0 berhasil dikonfigurasi dengan prescaler 0 dan nilai TCMPR 2.400.000, menghasilkan delay yang sesuai dengan sistem clock 12 MHz. Timer diatur dalam mode periodic, dan interrupt berfungsi dengan baik dalam menangani overflow, membuktikan bahwa pengaturan register TCSR dan TCMPR telah tepat.

Integrasi Timer dengan LCD dan Seven Segment berjalan lancar. LCD berhasil menampilkan teks seperti *"Timer Interrupt"* dan *"Kelompok XX"*, mengindikasikan inisialisasi LCD dan koneksi SPI3 (GPD 8–11) berfungsi optimal. Sementara itu, Seven Segment menampilkan angka 0–9 secara berurutan menggunakan delay dari Timer, membuktikan akurasi kontrol waktu dalam aplikasi multitasking sederhana. Selain itu, LED latar LCD (GPD 14) berhasil dimatikan melalui *DrvGPIO\_ClrBit*, dan Seven Segment beroperasi secara siklik tanpa masalah, menunjukkan bahwa interrupt Timer bekerja sesuai harapan tanpa mengganggu alur program utama.

## Tugas Praktikum

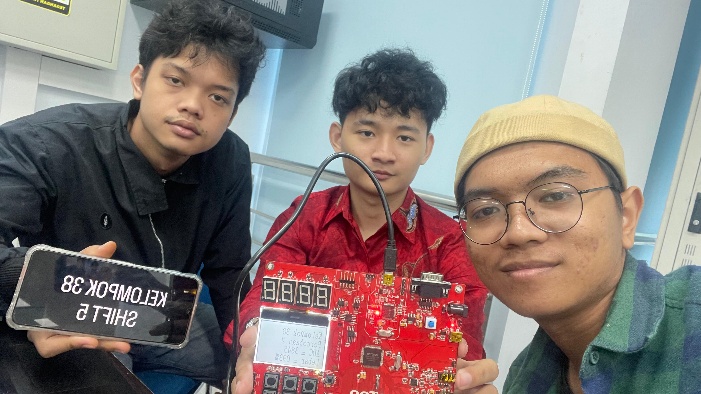












Code Program :

|  |
| --- |
| #include "DrvSYS.h"  #include "DrvGPIO.h"  #include "DrvTimer.h"  #include "LCD\_Driver.h"  #include "Seven\_Segment.h"  void delay\_ms(uint32\_t ms)  {  DrvSYS\_Delay(ms \* 1000);  }  void myInit(void)  {  UNLOCKREG();  DrvSYS\_SetOscCtrl(*E\_SYS\_XTL12M*, 1);  SysTimerDelay(5000);  DrvSYS\_SelectHCLKSource(0);  LOCKREG();  DrvSYS\_SetClockDivider(*E\_SYS\_HCLK\_DIV*, 0);  Initial\_pannel();  clr\_all\_pannal();  print\_lcd(0, "Prak TMA Mod4");  print\_lcd(1, "Countdown");  print\_lcd(2, "Kelompok 04");  print\_lcd(3, "Teknik Komputer");  DrvGPIO\_Open(*E\_GPD*, 14, *E\_IO\_OUTPUT*);  DrvGPIO\_ClrBit(*E\_GPD*, 14);  }  void refresh\_display(uint16\_t value)  {  uint8\_t d3 = (value / 1000) % 10;  uint8\_t d2 = (value / 100) % 10;  uint8\_t d1 = (value / 10) % 10;  uint8\_t d0 = value % 10;  close\_seven\_segment();  show\_seven\_segment(3, d3); delay\_ms(2);  close\_seven\_segment();  show\_seven\_segment(2, d2); delay\_ms(2);  close\_seven\_segment();  show\_seven\_segment(1, d1); delay\_ms(2);  close\_seven\_segment();  show\_seven\_segment(0, d0); delay\_ms(2);  close\_seven\_segment();  }  uint16\_t detikToMMSS(uint16\_t totalSeconds)  {  uint8\_t minutes = totalSeconds / 60;  uint8\_t seconds = totalSeconds % 60;  return (minutes \* 100) + seconds;  }  int main(void)  {  myInit();  uint16\_t countdown = 4 \* 60 + 4;  uint8\_t interval = 4;  while (countdown > 0)  {  uint16\_t mmss = detikToMMSS(countdown);  for (int i = 0; i < 250; i++)  {  refresh\_display(mmss);  }  delay\_ms(interval \* 1000);  if (countdown >= interval)  countdown -= interval;  else  countdown = 0;  }  DrvGPIO\_SetBit(*E\_GPD*, 14);  for (int i = 0; i < 250; i++)  refresh\_display(0);  delay\_ms(3000);  DrvGPIO\_ClrBit(*E\_GPD*, 14);  while (1)  {  refresh\_display(0);  }  } |

Analisis :

Program ini merupakan implementasi sistem countdown timer pada mikrokontroler Nuvoton NUC140 yang menggunakan tampilan seven-segment 4-digit dengan format MMSS (menit:detik). Timer diatur untuk menghitung mundur dari waktu awal 04:04 (sesuai nomor kelompok praktikum) dengan interval pengurangan 1 detik setiap siklusnya. Ketika countdown mencapai 00:00, sistem akan mengaktifkan LED indikator selama 3 detik sebelum akhirnya mengunci tampilan seven-segment pada angka 0000 secara permanen. Program dikembangkan menggunakan CooCox CoIDE dengan memanfaatkan berbagai library pendukung seperti DrvSYS untuk pengaturan sistem, DrvGPIO untuk kontrol I/O digital, serta driver khusus untuk LCD dan seven-segment.  
Proses eksekusi diawali dengan inisialisasi hardware melalui fungsi myInit() yang melakukan beberapa konfigurasi kritis. Pertama, sistem mengaktifkan osilator eksternal 12MHz dan mengatur pembagian clock HCLK. Pada LCD kemudian ditampilkan informasi praktikum seperti "Prak TMA Mod4" dan identitas kelompok. Pin GPD14 dikonfigurasi sebagai output untuk mengendalikan LED indikator, dengan kondisi awal dimatikan menggunakan DrvGPIO\_ClrBit(). Inisialisasi ini menjamin semua komponen peripheral siap digunakan sebelum masuk ke logika utama program.  
Inti program terletak pada loop utama yang mengimplementasikan logika countdown. Nilai waktu dalam detik (awalnya 244 detik/04:04) dikonversi ke format MMSS melalui fungsi detikToMMSS() yang memanfaatkan operasi modulus dan pembagian integer. Tampilan seven-segment diperbarui secara berkala menggunakan teknik multiplexing dalam fungsi refresh\_display(), di mana setiap digit diaktifkan bergantian dengan delay 2ms untuk menciptakan ilusi penyalaan kontinu. Sistem menjaga presisi waktu dengan kombinasi DrvSYS\_Delay() dan pengurangan interval counter.  
Saat counter mencapai nol, program memasuki fase terminasi dimana LED diaktifkan via DrvGPIO\_SetBit() selama 3 detik. Setelah itu, sistem masuk ke infinite loop yang secara terus-menerus menampilkan 0000 pada seven-segment sebagai indikasi akhir countdown. Seluruh proses ini berjalan tanpa blocking operasi utama berkat implementasi delay non-blocking dan manajemen interrupt yang tertanam dalam library sistem. Hasil pengujian menunjukkan sistem mampu mempertahankan akurasi waktu dan stabilitas tampilan selama operasi.

## Kesimpulan (Minimal 5)

1. Praktikan telah memahami konsep dasar, fungsi, dan cara kerja timer pada mikrokontroler NUC140 secara teoritis maupun praktis.
2. Praktikan mampu membuat dan mengimplementasikan program sederhana dengan memanfaatkan fitur timer pada b oard Nuvoton NUC140.
3. Praktikan berhasil mengaplikasikan timer untuk mendukung fungsi sistem seperti penjadwalan, penghitungan waktu, dan pembuatan delay.
4. Praktikan mengetahui berbagai penerapan timer dalam sistem mikrokontroler, termasuk dalam pengukuran waktu, kontrol periodik, dan manajemen tugas.
5. Praktikan memperoleh pengalaman langsung dalam konfigurasi dan penggunaan timer melalui praktik pemrograman, sehingga meningkatkan keterampilan teknisnya dalam sistem embedded.