

PRAKTIKUM DASAR MIKROPROSESOR

UNIT 4

COUNTER UP (7-SEGMENT)

LABORATORIUM DASAR ELEKTRO



ADAM MARDHATILLAH

3332200024

DM-22

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS SULTAN AGENG TIRTAYASA

2021

BAB I

METODOLOGI PRAKTIKUM

1.1 Prosedur Percobaan

Berikut ini adalah prosedur percobaan pada Unit 4 yaitu *Counter Up (7-segment)* sebagai berikut:

Langkah pertama : Pilih *Chip*

1. Dipanggil aplikasi *CooCox Software > CoSmart* dari *Start* Menu untuk membantu konfigurasi pin.
2. Diklik tombol *New chip* di tengah layar.
3. Dipilih **Nuvoton**, lalu pilih NUC140VE3CN.
4. Akan muncul gambar *chip* NUC140VE3CN dengan semua *peripheral*-nya.

Langkah kedua bagian a : Set pin 7-Segment Data

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

Langkah kedua bagian b : Set pin 7-Segment Select

1. Dientang **GPIOC** pada gambar *chip*.
2. Di jendela *Configuration* pilih GPIOC.
3. Di bawahnya, klik **pin 4 s/d 7** agar *Enable*.
4. *Discroll* ke bawah, atur pin *config* 4 s/d 7:
 - Mode Control: Output[1].

Langkah ketiga : Set pin *Push button*

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

Langkah keempat: Set Pin *Buzzer*

1. Di jendela *Configuration* pilih GPIOB.
2. Di bawahnya, klik pada Pin 11 agar *Enable*.
3. *Discroll* ke bawah, atur Pin *Config* 11:
 - Mode Control: *Output*

Langkah kelima: *Generate Project*

1. Disimpan konfigurasi dengan klik *Project > Save* lalu pilih *folder* dan ketik nama *file*.
2. Lalu Diklik *Generate > Generate CoIDE Project* lalu pilih *folder* dan ketik nama *project*, misal praktikum02.
3. *CoSmart* akan membuatkan *file Project* dan membukanya otomatis di *Windows Explorer*.

Langkah keenam : Buka *Project*

1. Klik dua kali *file praktikum02* untuk membuka *Project* dengan *CoIDE*.
2. Terlihat bahwa *file Project* sudah disiapkan, klik dua kali pada *main.c* untuk memulai memprogram[1].

Langkah ketujuh : Pemrograman

Nuvoton telah menyediakan *Board Support Package* (yang sudah di-install) 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 :

1. Diklik kanan di *cmsis_lib\Include\Driver*, lalu pilih *Add Files. Browse* ke *C:\Nuvoton\BSPLibrary\NUC100SeriesBSP\NuvPlatform_Keil\Include\NUC1xx-LB_002*
2. Dipilih file *Seven_Segment.h*

Langkah 2 :

1. Diklik kanan di *cmsis_lib\Src\Driver*, lalu pilih *Add Files. Browse* ke *C:\Nuvoton\BSPLibrary\NUC100SeriesBSP\NuvPlatform_Keil\Src\NUC1xx-LB_002*.
2. Dipilih file *Seven_Segment.c*

LISTING PROGRAM

```
#include "DrvGPIO.h"
#include "DrvSYS.h"
#include "Seven_Segment.h"

int nilai=0;
void Init();
int main(void)
{
    Init();
    close_seven_segment();
    show_seven_segment(0, nilai);
    while(1)
    {
        DrvGPIO_SetBit(E_GPB, 11);
        if(DrvGPIO_GetBit(E_GPB, 15) == 0)
        {
            DrvGPIO_ClrBit(E_GPB, 11);
            nilai=nilai+1;
            close_seven_segment();
            show_seven_segment(0, nilai);
        }
    }
}
```

```
        DrvSYS_Delay(1000000);  
    }  
}  
}
```

Langkah kedelapan : *Build* dan *Download* program

1. Diklik ikon *Build* (F7) di *toolbar* untuk melakukan *compile* program.
2. Dipanggil aplikasi *CooCox Software* > *CoFlash*.
3. Dipilih **Nu-Link** pada menu adapter.
4. Diklik Nuvoton, lalu pilih NUC140VE3CN.
5. Diklik *Command*, lalu *browse data file*.

Klik ikon *Download Code to Flash* di *toolbar* untuk meng-*input* program ke *chip*.
Pastikan *Done* tidak ada *error*[1].

BAB II

TUGAS

2.1 Tugas Unit

1. Apa yang dimasuk *Identifier* (Pengenal) dan fungsinya?
2. Jelaskan bagaimana cara kerja dari *7-segment* pada unit ini!

Jawaban

1. Identifier merupakan sebuah pengenal atau pengidentifikasi yang kita deklarasikan agar kompiler dapat mengenalinya. atau Identifier juga biasa diartikan sebagai nama yang diberikan untuk penamaan objek, Identifier dapat berupa nama variabel, nama konstanta, nama fungsi, nama prosedur maupun nama namespace. namu pada kesempatan ini kita akan batasi pembahasannya pada identifier yang berperan sebagai variabel dan konstanta saja[2].
2. *7-segmenti* pada unit ini bekerja dengan *counter up* dimana pada listing program telah dimasukan kode untuk display awal yaitu angka 0 dan ketika kita tekan *push button* akan bertambah ke angka 1, itu karena pada *listing program* kita menambahkan (`DrvGPIO_GetBit(E_GPB,15)==00`), yang mana jika *push button* ditekan akan menambah nilai pada display yang dapat dilihat juga pada *listing program* yaitu di `nilai=nilai+1;`, itu mengapa jika ditekan maka nilai akan bertambah 1. Namun ada percobaan pada unit yang menyebabkan *counter down* yang terdapat pada percobaan terakhir yaitu percobaan ke 6.

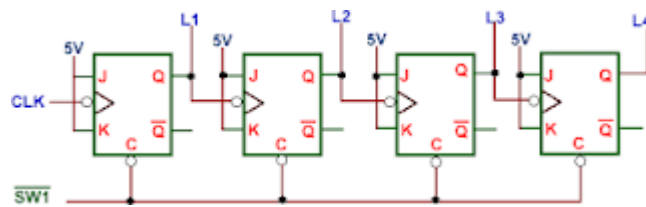
BAB III

ANALISI

3.1 Dasar Teori

3.1.1 Asynchronous Binary Up Counter

Counter ini dapat menghitung bilangan biner dengan urutan dari bawah ke atas. Apabila digunakan 4 buah flip-flop, maka kita dapat melakukan hitungan paling tinggi adalah 1111. Counter yang dapat menghitung sampai 1111 disebut 4 bit binary counter. Oleh karena dapat menghitung dengan cara ke atas, maka disebut pula asynchronous 4 binary up counter. Gambarnya dapat dilihat seperti berikut.



Pada rangkaian di atas, input J dan K dari seluruh flip-flop dibuat dalam keadaan 1. Sebelum pulsa pertama yang akan dihitung masuk ke input, maka seluruh output counter L4, L3, L2 dan L1 dibuat 0 terlebih dahulu dengan jalan membuat clear dalam keadaan 0 walaupun sesaat.

Pada saat pulsa pertama bergerak dari 1 ke 0, maka output flip-flop A akan berubah dari 0 ke 1, Output B akan tetap karena sinyal yang masuk pada input clock berubah dari 0 ke 1. Flip ke 3 dan 4 juga tidak mengalami perubahan karena belum ada perubahan pada input clocknya. Jadi dapat disimpulkan bahwa sesudah pulsa pertama datang keadaan output L4, L3, L2, L1 adalah 0001.

Selanjutnya apabila pulsa kedua bergerak dari 1 ke 0, output flip-flop 1 akan kembali menjadi 0, akibatnya terjadi perubahan juga pada input clock flip-flop 2 (dari 1 ke 0) sehingga output flip-flop 2 menjadi 1. Sedangkan flip flop 3 dan 4 outputnya

belum mengalami perubahan karena pulsa input clocknya belum mengalami perubahan dari 1 ke 0. jadi sekarang output rangkaian counter ini adalah 0010.

Begitulah seterusnya sampai pulsa ke 15 datang. Keempat output rangkaian counter akan bernilai 1111. Begitu masuk pulsa ke 16 (perubahan dari 1 ke 0) datang maka output dari masing-masing flip-flop akan berubah menjadi 0000 (seperti keadaan awal)[3].

3.2 Analisi Percobaan *Counter Up (7-segment)*

3.2.1 Percobaan 1

No	Percobaan	Keterangan
1	Tekan tombol <i>push button</i> sebanyak 10 kali.	Ketika satu persatu ditekan, buzzer mengeluarkan bunyi tiap ditekan

Pada percobaan ini yaitu percobaan pertama dimana kita diminta untuk menekan tombol *push button* sebanyak 10 kali. Dan didapatkan pada keterangan yaitu Ketika satu persatu ditekan, buzzer mengeluarkan bunyi tiap ditekan dan selain itu *7-segment* mengalami *counter up*, pada dasarnya, percobaan 1 ini kita hanya mengetest program yang ada di *listing program*, itu mengapa kita hanya menekan tombol *push button* yang telah di program. Terdapat juga display pada nuvoton yang menunjukkan angka 0 di bagian ujung, mengapa bisa di paling ujung?, dikarenakan pada *listing program* kita memilih *7 segment* ke-nol, dan juga karena *7 segment* langsung menampilkan yang ada pada identifier nilai yaitu 0, itu mengapa berada paling ujung dikarenakan dipilihnya pada *7 segment 0*, dan kenapa nilai menunjukkan angka 0, karena kita telah mengset *initial value* adalah 0. Lalu ketika ditekan sekali *push button* maka display akan menunjukkan angka 1, itu karena pada *listing program* kita menambahkan (`DrvGPIO_GetBit(E_GPB,15)==00`), yang mana jika *push button* ditekan akan menambah nilai pada display yang dapat dilihat juga pada *listing program* yaitu di `nilai=nilai+1;`, itu mengapa jika ditekan maka nilai akan bertambah 1. Jika ditekan

sepuluh kali angka menampilkan karakter acak, dan dia terus mengacak hingga 256 (dua pangkat 8) kali lalu berulang kembali ke angka 0.

3.2.2 Percobaan 2

2	<i>Listing</i> program <code>#include "Seven_Segment.h"</code> dihilangkan lalu <i>build</i> dan <i>download</i> kemudian operasikan unit. Jika sudah kembalikan seperti semua.	Masih sama seperti percobaan pertama, ketika satu persatu ditekan, buzzer mengeluarkan bunyi tiap ditekan
---	---	---

Pada percobaan ini kita diminta untuk *Listing* program `#include "Seven_Segment.h"` dihilangkan lalu *build* dan *download* kemudian operasikan unit. Jika sudah kembalikan seperti semua. Dan dihasilkan Masih sama seperti percobaan pertama, ketika satu persatu ditekan, buzzer mengeluarkan bunyi tiap ditekan, itu dikarenakan *library* tersebut telah ada pada / tersimpan pada `cmsis_lib`, itu mengapa walaupun *library* tersebut dihapus, itu tidak menyebabkan Program *failed* / tidak akan menyebabkan apa apa.

3.2.3 Percobaan 3

3	<i>Identifier</i> (Pengenal) pada <i>listing</i> program <code>int nilai=0;</code> diubah menjadi <code>int hasil=0;</code> lalu <i>build</i> dan <i>download</i> kemudian operasikan unit. Jika sudah kembalikan seperti semua.	<i>Build failed</i> pada <i>listing program</i> dikarenakan <code>int nilai</code> diganti ke <code>hasil</code>
---	--	--

Pada percobaan ketiga ini kita diminta *Identifier* (Pengenal) pada *listing* program `int nilai=0;` diubah menjadi `int hasil=0;` lalu *build* dan *download* kemudian operasikan unit. Jika sudah kembalikan seperti semua. Dihasilkan *Build failed* pada *listing program* dikarenakan `int nilai` diganti ke `hasil`, ini dikarenakan kita

mendeklarasikan hasil namun yang di panggil pada *listing programnya* adalah nilai, itu menyebabkan *build failed* yang dapat terlihat pada keterangannya yaitu bertulis nilai *undeclared* atau nilai tidak di deklarasikan dikarenakan variabel nilai tidak dimasukan pada integer. Namun walaupun *build failed* nuvoton tetap berjalan, dikarenakan pada software CoFlash akan tetap menjalankan program yang bermasalah namun program yang di *load* adalah program sebelumnya yang tidak bermasalah.

3.2.4 Percobaan 4

4	<i>Identifier</i> (Pengenal) pada <i>listing</i> program <code>int nilai=0;</code> diubah menjadi <code>int nilai=7;</code> lalu <i>build</i> dan <i>download</i> kemudian operasikan unit. Jika sudah kembalikan seperti semua.	Tampilan awal 7 lalu saat ditekan buzzer bersuara dan 7 <i>segment</i> merespon
---	--	---

Pada percobaan empat, yaitu *Identifier* (Pengenal) pada *listing* program `int nilai=0;` diubah menjadi `int nilai=7;` lalu *build* dan *download* kemudian operasikan unit. Jika sudah kembalikan seperti semua. Maka dihasilkan Tampilan awal 7 lalu saat ditekan buzzer bersuara dan 7 *segment* merespon, saat belum diganti 7 tampilan awal adalah 0, namun setelah diganti hasil awal adalah 7, itu berarti program mengirim perintah pada nuvoton untuk memulai perhitungan dari tujuh, yang mana nilai awal sebelumnya adalah 0, saat diganti ke `int nilai=7;` menjadi 7, dan itu menyebabkan ketika *push button* akan naik ke 8 lalu 9. Dan jika kita ganti ke `int nilai=4;` maka perhitungan awalnya adalah 4 dan jika *push button* ditekan maka akan naik terus ke 5 6 dengan alasan yang sama .

3.2.5 Percobaan 5

5	<i>Listing</i> program <code>nilai=nilai+1;</code> diubah menjadi <code>nilai=nilai+2;</code> lalu <i>build</i> dan <i>download</i> kemudian operasikan unit. Jika sudah kembalikan seperti semua.	Tampilan awal menunjukkan angka 0, dikarenakan <code>nilai=nilai+2</code> maka menyebabkan tiap kita menekan <i>pushbutton</i> akan bertambah dua nilai pada 7 segment
---	--	--

Pada percobaan ini kita diminta pada *Listing* program `nilai=nilai+1;` diubah menjadi `nilai=nilai+2;` lalu *build* dan *download* kemudian operasikan unit. Jika sudah kembalikan seperti semua, dihasilkan Tampilan awal menunjukkan angka 0, dikarenakan `nilai=nilai+2` maka menyebabkan tiap kita menekan *pushbutton* akan bertambah dua nilai pada 7 segment. Contoh nya adalah ketika menekan sekali display akan menunjukkan angka 2, lalu ditekan lagi menunjukkan angka 4 dan kelipatan 2 ketika ditekan seterusnya.

3.2.6 Percobaan 6

6	<i>Listing</i> program <code>int nilai=0;</code> diubah menjadi <code>int nilai=9;</code> dan <i>listing</i> program <code>nilai=nilai-1;</code> lalu <i>build</i> dan <i>download</i> kemudian operasikan unit. Jika sudah kembalikan seperti semua.	Dikarenakan <code>int nilai=9</code> maka angka pada 7 segment dimulai dari 9 dan juga yang awalnya <code>nilai=nilai+1</code> diganti menjadi <code>nilai-1</code> , itu menyebabkan tiap <i>pushbutton</i> maka nilai 9 pada 7 segment dikurangi 1
---	---	--

Pada percobaan ini kita membuat *Listing* program `int nilai=0;` diubah menjadi `int nilai=9;` dan *listing* program `nilai=nilai-1;` lalu *build* dan *download* kemudian operasikan unit. Jika sudah kembalikan seperti semua. Dikarenakan `int nilai=9` maka angka pada 7 segment dimulai dari 9 dan juga yang awalnya `nilai=nilai+1` diganti menjadi `nilai-1`, itu menyebabkan tiap *pushbutton* ditekan akan menghasilkan

nilai 9 pada *7 segment* dikurangi 1, jadi ketika ditekan akan berkurang satu, itu pula menyebabkan *counter down* pada *7 segment* yang dimana akan menghitung mundur, display menunjukkan angka 1 lalu jika ditekan lagi akan menunjukkan karakter acak.

BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan pada praktikum yang telah dilakukan, mengenai *Counter Up (7-Segment)* dapat diambil kesimpulan sebagai berikut:

1. Identifier merupakan sebuah pengenalan atau pengidentifikasi yang kita deklarasikan agar kompiler dapat mengenalinya. atau Identifier juga biasa diartikan sebagai nama yang diberikan untuk penamaan objek.
2. *7 segment* akan terus bertambah jika *push button* terus ditekan, dan akan menunjukkan bentuk acak jika sudah melebihi angka 9 pada *7 segment*
3. `(DrvGPIO_GetBit(E_GPB,15)==00)`, yang mana jika *push button* ditekan akan menambah nilai pada display yang dapat dilihat juga pada *listing program* yaitu di `nilai=nilai+1;`, itu mengapa jika ditekan maka nilai akan bertambah 1.

DAFTAR PUSTAKA

- [1].Asisten Lab Dasar Elektro, " *Counter Up (7-Segment)*," in Modul *Praktikum Dasar Mikroprosesor 2021*, Cilegon, Universitas Sultan Ageng Tirtayasa Fakultas Teknik, 2021 , pp 17 – 20.
- [2].MateriDosen, “*Pengertian, Jeni dan Contoh Identifier dalam C++*” in Website Materi Dosen [terhubung berkala] <https://www.materidosen.com/2017/02/pengertian-jenis-dan-contoh-identifier.html> (diakses pada 14 September 2021 pukul 21.03)
- [3].Hari , “ *Contoh atau rangkaian pencacah*” in Website Uniksharianja 14 May 2015 [terhubung berkala] <https://www.uniksharianja.com/2015/05/counter-atau-rangkaian-pencacah.htm> (Diakses pada 14 September 2021 Pukul 21.50)