

PRAKTIKUM DASAR MIKROPROSESOR

UNIT 2

BELL (PUSH BUTTON DAN BUZZER)

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 1 yaitu BELL (PUSH BUTTON DAN BUZZER) sebagai berikut :

Langkah ke-1: Pilih Chip

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

Langkah ke 2: Set Pin *Pushbutton*

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

Langkah ke 3: Set Pin *Buzzer*

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

Langkah ke-4: Generate Project

1. Menyimpan 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[1].

Langkah ke 5: Buka *Project*

1. Mengklik dua kali *file* praktikum01 untuk membuka *Project* dengan *CoIDE*.
2. Terlihat bahwa *file Project* sudah disiapkan, klik dua kali pada main.c untuk mulai memprogram[1].

Langkah ke-6: Pemrograman

LISTING PROGRAM

```
#include "DrvGPIO.h"
#include "DrvSYS.h"
void Init();
int main(void)
{
    Init();
    while(1)
    {
        if(DrvGPIO_GetBit(E_GPB,15)==0)
        {
            DrvGPIO_ClrBit(E_GPB,11);
        }
        else
        {
            DrvGPIO_SetBit(E_GPB,11);
        }
    }
}
```

Langkah ke-6: *Build* dan *Download* program

1. Mengklik *icon Build* (F7) di toolbar untuk melakukan *compile* program. Pastikan *Build Successful*, tidak ada *error*.
2. Memanggil aplikasi *CooCox Software* > *CoFlash*.
3. Memilih *Nu-Link* pada menu adaptor.

4. Mengklik Nuvoton pilih Nuvoton, lalu pilih NUC140VE3CN.
5. Mengklik *Command*, lalu *browse* data file.
6. Mengklik *icon Download Code to Flash* di *toolbar* untuk meinput program ke chip.
Pastikan *Done*, tidak ada *error[1]*.

BAB II

TUGAS

2.1 Tugas Unit

1. Sebutkan dan jelaskan *port* apa saja yang digunakan dan mengapa *port* tersebut digunakan.
2. Jelaskan fungsi dari *if else* pada program diatas dan mengapa perintah tersebut digunakan.
3. Jelaskan secara singkat bagaimana program di atas bekerja.

Jawaban

1. Port yang digunakan adalah
2. Pada fungsi *if* terdapat perintah GetBit, perintah GetBit tersebut juga ada dalam *library* DrvGPIO.h, GetBit digunakan untuk mengambil logika dari suatu inputan, dan pada inputan tersebut yaitu ketika inputan = 0 maka *buzzer* akan mendapat logika 0 atau menyala, lalu fungsi *else* yaitu selain dari kondisi inputan = 0 atau saat *pushbutton* = 1 maka *buzzer* akan mati atau SetBit.
3. Program dapat bekerja jika saat di *listing program* menuliskan fungsi dengan benar, yang dimana agar *buzzer* dapat menyala, *buzzer* menyala karena pada *listing program* dimana saat ClrBit atau saat kondisi low (*pushbutton* ditekan) maka *buzzer* menyala namun ketika dilepas yaitu saat SetBit kondisi High (*pushbutton* tak ditekan) maka *buzzer* akan mati.

2.2 Tugas Tambahan

1. Apa perbedaan antara kurung sudut kutip pada *file header*?

Jawaban: karena dalam praktiknya, perbedaan terdapat pada lokasi dimana preprocessor mencari file yang disertakan, contoh pada `#include <filename>` pencarian preprocessor dengan cara yang bergantung pada implementasi, biasanya dalam direktori pencarian yang ditentukan sebelumnya oleh kompiler/IDE. Metode ini digunakan untuk memasukan file header perpustakaan standar.

Dan pada `#include "filename"` pencarian preprocessor pertama kali di direktori yang sama dengan file yang berisi arahan, dan kemudian mengikuti jalu pencarian yang digunakan

untuk `#include` bentuk. Metode ini biasanya digunakan untuk mencakup programmer-defined file header.

2. Mengapa menggunakan `while(1)` dan bukan `while(0)`!

Jawaban: Perbedaan `while(1)` dan `while(0)` adalah `while(1)` merupakan *infinite loop* yang akan berjalan sampai pernyataan *break* dikeluarkan secara eksplisit. Bukan hanya `while(1)` tetapi bilangan bulat apa pun yang bukan nol akan memberikan efek yang sama seperti `while(1)`. Oleh karena itu, `while(1)`, `while(2)` atau `while(-255)`, semua akan memberikan *infinite loop* saja. Dan pada `while(0)` merupakan kebalikan dari `while(1)`, yaitu berarti kondisi akan selalu salah dan dengan demikian kode dalam `while` tidak akan pernah dieksekusi. Oleh karena itu mengapa digunakan `while(1)` dan bukan `while(0)`.

3. Hubungan GPIOB dengan GPIOC, mengapa saat inisialisasi GPIOB namun GPIOC tetap menyala?

Jawaban: Dikarenakan terletak pada port yang sama yaitu GPIO (*General Purpose Input Output*) yang dimana letak *input* dan *output* yang sama itu mengapa GPIOC ikut menyala.

BAB III

ANALISI

3.1 Dasar Teori

3.1.1 Percabangan *If Else*

Pernyataan *IF* adalah salah satu pernyataan penyeleksian yang memungkinkan kita memanipulasi aliran jalannya program berdasarkan conditional *expression*. Hal ini dapat memungkinkan kita membuat program yang berjalan secara fleksibel sesuai keadaan dari pengguna dan mesin. Penyeleksian *if* adalah pernyataan penyeleksian yang mencari kebenaran dari conditional *expression* yang disebutkan. conditional *expression* harus berupa bilangan Boolean atau operasi yang menghasilkan bilangan Boolean dan menyatakan benar atau salah atas *expression* tersebut. Ketika mesin eksekusi bertemu dengan penyeleksian *if* maka CPU akan memeriksa kebenaran dari conditional *expression* yang disebutkan, jika benar (*true*) maka perintah yang ada di dalamnya akan dijalankan, jika salah (*false*) maka akan memeriksa pernyataan *else if* (jika ada), hal itu dilakukan berulang satu demi satu hingga menemukan kondisi yang bernilai benar (*true*). Jika tidak ditemukan maka akan melakukan perintah pernyataan *else*. Jika *else* tidak ada maka pernyataan *if* secara keseluruhan akan diabaikan[2].

Kondisi setelah keyword *if* atau *else if* adalah tempat dimana anda menaruh sebuah nilai boolean atau ekspresi seperti perbandingan, logika dan lain-lain yang menghasilkan bilangan Boolean. Badan pernyataan *if* akan dieksekusi tergantung pada kebenaran dari kondisi. Jika evaluasi ekspresi bernilai benar (*true*) sesuai kondisi maka program akan melaksanakan perintah (pernyataan) di dalamnya. Jika tidak maka akan diabaikan dan beralih ke pilihan lain atau keluar dari pernyataan penyeleksian *if*. *else if* adalah pilihan alternative dari penyeleksian untuk mencari kondisi yang diinginkan, *else if* memiliki fungsi yang sama seperti pernyataan *if* dan diletakan setelah pernyataan *if*[2].

Else adalah pilihan terakhir yang akan dijalankan jika semua pilihan tidak memiliki nilai benar (*true*) pada kondisi yang ada. *else* merupakan pernyataan opsional untuk digunakan berdasarkan kebutuhan, jika tidak menggunakan pernyataan *else* maka keseluruhan dari pernyataan *if* akan diabaikan[2].

3.2 Analisis Percobaan Bell (Push Button dan Buzzer)

3.2.1 Percobaan 1

Pada percobaan pertama dimana saat menekan tombol *pushbutton*, dapat dilihat pada Lampiran A bahwa *buzzer* menyala karena pada *listing program* dimana saat `ClrBit` atau saat kondisi low (*pushbutton* ditekan) maka *buzzer* menyala namun ketika dilepas yaitu saat `SetBit` kondisi High (*pushbutton* tak ditekan) maka *buzzer* akan mati.

3.2.2 Percobaan 2

Pada percobaan kedua yaitu saat fungsi *if* dihilangkan lalu *build* dan *download* kemudian operasikan unit. Jika sudah kembalikan seperti semula. Build failed karena penyelesaian kondisinya hilang yang dimana jika hanya kondisi *else* saja tidak bisa, dapat dilihat pada Lampiran B.

3.2.3 Percobaan 3

Pada percobaan ketiga yaitu saat fungsi *else* dihilangkan lalu *build* dan *download* kemudian operasikan unit. Jika sudah kembalikan seperti semula. Build Success namun *buzzer* tidak menyala dikarenakan program bersifat sekuensial, ketika *else* dihilangkan otomatis akan mengambil logika terakhir yaitu logika 1 yang menyebabkan *buzzer* tidak menyala, dapat dilihat pada Lampiran C.

3.2.4 Percobaan 4

Pada percobaan keempat fungsi *else* dihilangkan lalu *build* dan *download* kemudian operasikan unit. Jika sudah kembalikan seperti semula. sama seperti percobaan 3, build successful namun *buzzer* tidak menyala, karena penyelesaian kondisi dihapus maka dia akan bekerja sekuensial, karena sekuensial otomatis dicari perintah yang baru yang dikerjakan oleh Nuvoton yaitu perintah terakhir, maka *buzzer* akan mati walaupun *pushbutton* ditekan, dapat dilihat pada Lampiran D

3.2.5 Percobaan 5

Pada percobaan kelima dimana angka 15 pada *listing program*
`DrvGPIO_GetBit(E_GPB, 15);` . Diganti menjadi 14 lalu *build* dan *download* kemudian

operasikan unit. Lakukan juga dengan angka 16. Jika sudah kembalikan seperti semula. saat diganti menjadi pin 14 *buzzer* mati, dan saat pin 16 *buzzer* tetap menyala, dikarenakan pin 15 untuk *pushbutton* dan jika kita ganti ke 14 maka itu bukan pin untuk *pushbutton*, dan kondisi takan terpenuhi karena *pushbutton* tak ditekan maka langsung ke *else* yang mana *buzzer* akan mati, dan pada pin 16 *buzzer* tetap menyala, pada GPIO tidak ada pin 16, karena tak adanya pin 16 otomatis kondisinya menjadi benar, 16 bernilai 0 yang mana perintah yang ada pada kondisi tersebut akan diproses, itu mengapa *buzzer* akan terus menyala yang dapat dilihat pada Lampiran E.

3.2.6 Percobaan 6

Pada percobaan keenam ini yang mana saat Angka 11 pada *listing* program `DrvGPIO_ClrBit(E_GPB,11); DrvGPIO_SetBit(E_GPB,11);` . Diganti menjadi 10 lalu *build* dan *download* kemudian operasikan unit. Lakukan juga dengan angka 12. Jika sudah kembalikan seperti semula. saat angka 11 diganti ke 10 dan 12 *buzzer* tidak menyala, dikarenakan pin pada *buzzer* merupakan pin 11 dan jika diganti ke 10 dan 12 itu bukan lagi pin *buzzer* dan menyebabkan *buzzer* tak menyala, dapat dilihat pada Lampiran F.

3.2.7 Percobaan 7

Dipercobaan terakhir ini yaitu percobaan ketujuh, dilakukan dengan *menginput* dan *listing* program `DrvGPIO_ClrBit(E_GPC,14);` di dalam fungsi *else* lalu *build* dan *download* kemudian operasikan unit. Hasil yang didapatkan adalah *buzzer* dan LED menyala, dikarenakan program menggunakan port C yang dimana menyebabkan LED menyala, tidak seperti percobaan ke 5 yang mana menggunakan port B, dapat dilihat pada Lampiran G.

BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan pada praktikum yang telah dilakukan, mengenai Bell (Push Button dan Buzzer) dapat diambil kesimpulan sebagai berikut:

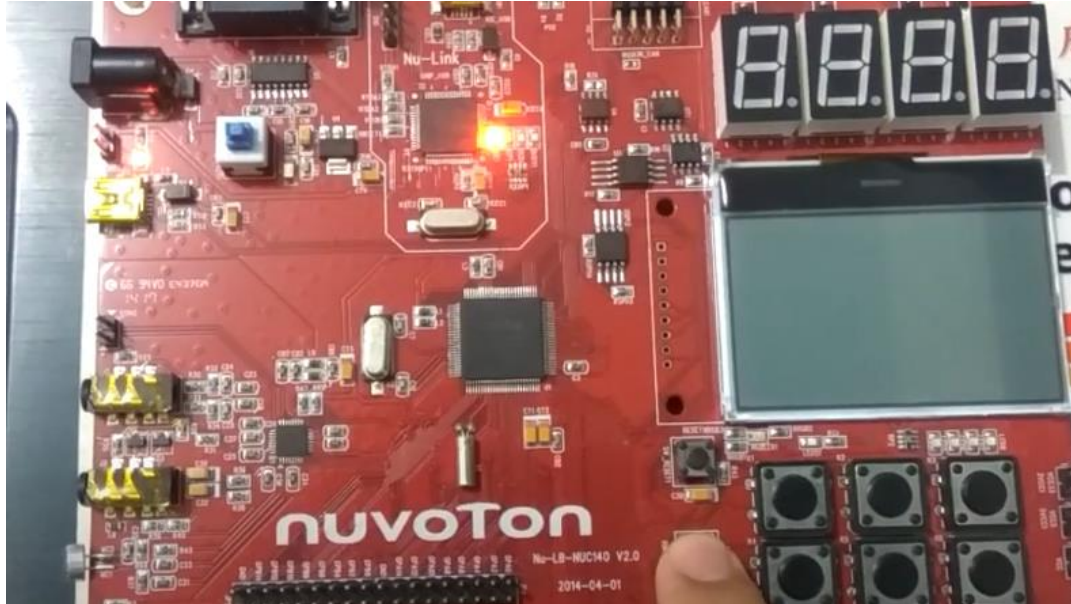
1. Terdapat port yang digunakan pada praktikum ini. Port yang digunakan adalah GPIO (*General Purpose Input Output*), dikarenakan semua komponen yang ada di Nuvoton dianggap sebagai input/output ada di port GPIO.
2. Fungsi *if* dan *else* disini adalah fungsi *if* terdapat perintah *GetBit*, perintah *GetBit* tersebut juga ada dalam *library* *DrvGPIO.h*, *GetBit* digunakan untuk mengambil logika dari suatu inputan, dan pada inputan tersebut yaitu ketika inputan = 0 maka *buzzer* akan mendapat logika 0 atau menyala, lalu fungsi *else* yaitu selain dari kondisi inputan = 0 atau saat *pushbutton* = 1 maka *buzzer* akan mati atau *SetBit*.
3. Pada Program *BELL* (*Push* dan *Buzzer*) ini kita dapat mengetahui pada *listing program* maupun Nuvoton cara agar dapat membuat *buzzer* dan LED menyala, yang mana telah diterangkan pada percobaan satu yaitu saat menekan tombol *pushbutton*, dapat dilihat pada Lampiran A bahwa *buzzer* menyala karena pada *listing program* dimana saat *ClrBit* atau saat kondisi low (*pushbutton* ditekan) maka *buzzer* menyala namun ketika dilepas yaitu saat *SetBit* kondisi High (*pushbutton* tak ditekan) maka *buzzer* akan mati.

DAFTAR PUSTAKA

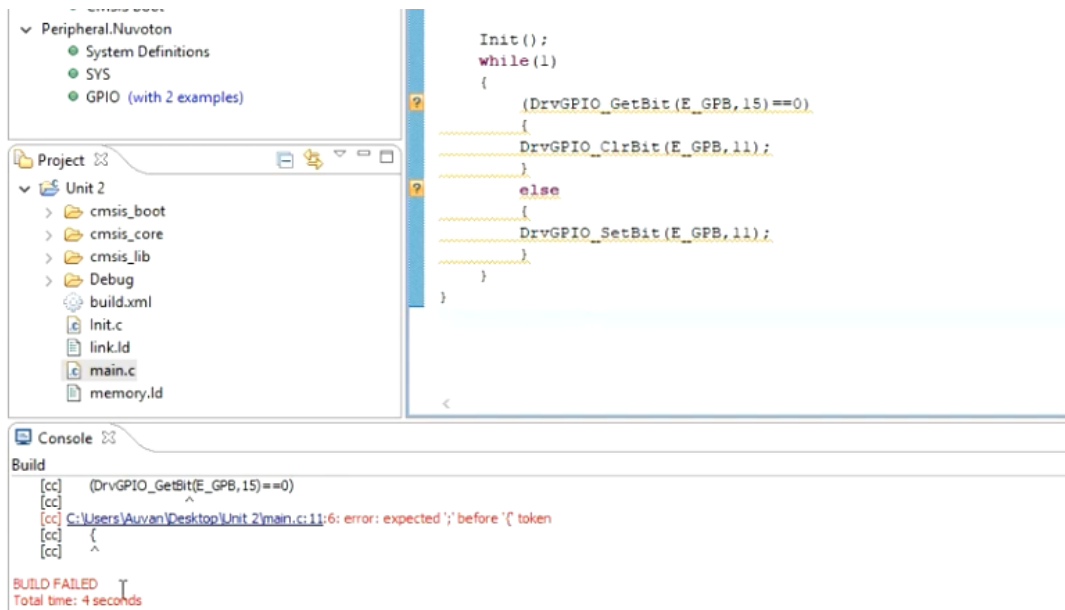
- [1].Asisten Lab Dasar Elektro, "*BELL (Push dan Buzzer)*," in Modul *Praktikum Dasar Mikroprosesor 2021*, Cilegon, Universitas Sultan Ageng Tirtayasa Fakultas Teknik, 2021 , pp 15 - 16
- [2].Fajar, "Penjelasan Pernyataan Penyeleksian IF" in Website Belajar C++, 9 November, 2018
[terhubung berkala] <https://www.belajarcpp.com/tutorial/cpp/if/> (diakses pada 11-09-21 pukul 12.12)

LAMPIRAN

Lampiran A. Percobaan 1



Lampiran B. Percobaan 2



Lampiran C. Percobaan 3



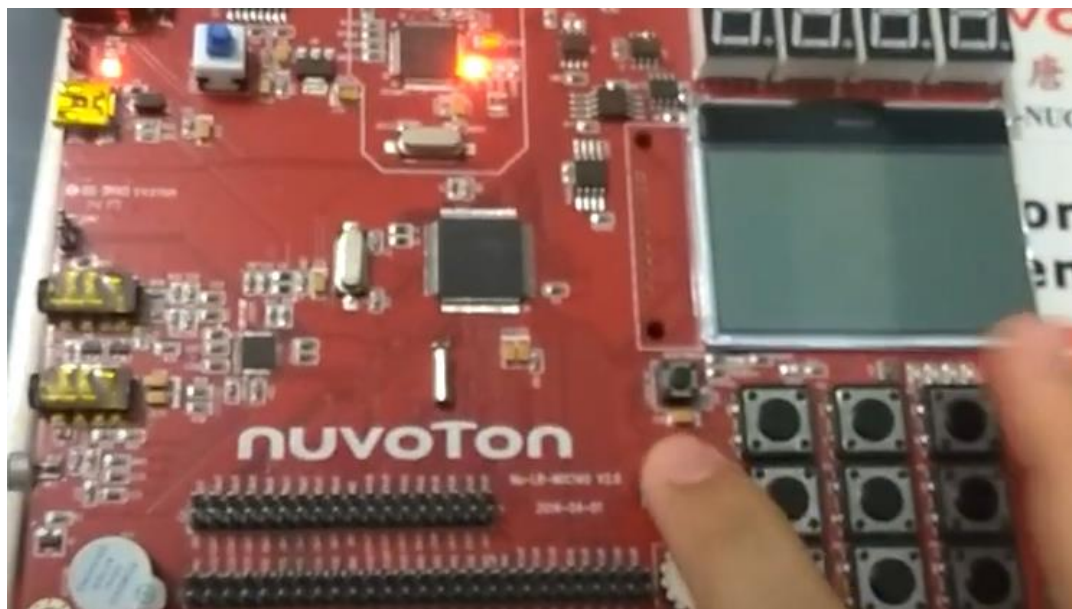
Lampiran D. Percobaan 4



Lampiran E. Percobaan 5



Lampiran F. Percobaan 6



Lampiran G. Percobaan 7

