

PRAKTIKUM DASAR MIKROPROSESOR

UNIT 8

WATER LEVEL

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 8 yaitu *Water Level* 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 : Set pin *Push button*

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

Langkah ketiga : Set pin *Buzzer*

1. Dicentang **GPIOB** pada gambar *chip*.
2. Di jendela *Configuration* pilih GPIOB.
3. Di bawahnya, klik **pin 11** agar *Enable*.
4. *Discroll* ke bawah, atur pin *config* 11:

5. Mode Control : Output

Langkah keempat bagian b : Set pin *Backlight*

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

Langkah keempat bagian b : Set pin *Backlight*

1. Centang **GPIOD** pada gambar *chip*.
2. Di jendela *Configuration* pilih GPIOD.
3. Di bawahnya, klik **pin 14** agar *Enable*.
4. *Discroll* ke bawah, atur pin *config* 14:
 - Mode Control : Output

Langkah kelima : Set pin LED

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

Langkah keenam : Set pin VR

1. Centang ADC pada gambar *chip*.
2. Di jendela *Configuration* pilih ADC.
3. Diklik pada *channel 7* agar *Enable*.
4. *Discroll* ke atas, klik pada *Clock Source*, pilih INTERNAL 22M.
5. Diklik pada *Clock Divisor*, isi dengan angka 2.

Langkah ketujuh : Generate Project

1. Simpan 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 kedelapan : Buka Project

1. Diklik 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.

Langkah kesembilan : 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* dan *7-segment* 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. Pilih *file LCD_Driver.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. Pilih *file LCD_Driver.c* dan *Ascii_Table.c*.

LISTING PROGRAM

```

#include "DrvGPIO.h"
#include "DrvSYS.h"
#include "LCD_Driver.h"
#include "DrvADC.h"
int adc,a=0;
void Init();
int main(void)
{
    Init();
    Initial_annel();
    DrvGPIO_ClrBit(E_GPD,14);

    clr_all_annel();
    while(a==0)
    {
        print_lcd(0,"NUVOTON");
        print_lcd(1,"DASAR MIKRO");
        print_lcd(2,"TEKAN PUSH BTN");
        print_lcd(3,"UNTUK MELANJUTKN");
        if(DrvGPIO_GetBit(E_GPB,15)==0)
        {
            a=1;
            clr_all_annel();
        }
    }
    while(a==1)
    {
        if(DrvGPIO_GetBit(E_GPB,15)==1)
        {
            print_lcd(0,"ADC=");
            DrvADC_StartConvert();
            adc=DrvADC_GetConversionData(7);
            Show_Word(0,5,adc/1000+'0');
            Show_Word(0,6,adc%1000/100+'0');
            Show_Word(0,7,adc%100/10+'0');
            Show_Word(0,8,adc%10+'0');
            if(adc<=999)
            {
                DrvGPIO_SetBit(E_GPC,15);
                DrvGPIO_SetBit(E_GPC,14);
                DrvGPIO_SetBit(E_GPC,13);
                DrvGPIO_SetBit(E_GPC,12);
                DrvGPIO_ClrBit(E_GPB,11);
                print_lcd(1,"LOW!!!");
            }
            if(adc>=1000)
            {
                if(adc<=1999)
                {
                    DrvGPIO_ClrBit(E_GPC,15);

```

```

        DrvGPIO_SetBit(E_GPC,14);
        DrvGPIO_SetBit(E_GPC,13);
        DrvGPIO_SetBit(E_GPC,12);
        DrvGPIO_SetBit(E_GPB,11);
        print_lcd(1,"NORMAL");
    }
}
if (adc>=2000)
{
    if (adc<=2999)
    {
        DrvGPIO_ClrBit(E_GPC,15);
        DrvGPIO_ClrBit(E_GPC,14);
        DrvGPIO_SetBit(E_GPC,13);
        DrvGPIO_SetBit(E_GPC,12);
        DrvGPIO_SetBit(E_GPB,11);
        print_lcd(1,"NORMAL");
    }
}
if (adc>=3000)
{
    if (adc<=3999)
    {
        DrvGPIO_ClrBit(E_GPC,15);
        DrvGPIO_ClrBit(E_GPC,14);
        DrvGPIO_ClrBit(E_GPC,13);
        DrvGPIO_SetBit(E_GPC,12);
        DrvGPIO_SetBit(E_GPB,11);
        print_lcd(1,"NORMAL");
    }
}
if (adc>=4000)
{
    DrvGPIO_ClrBit(E_GPC,15);
    DrvGPIO_ClrBit(E_GPC,14);
    DrvGPIO_ClrBit(E_GPC,13);
    DrvGPIO_ClrBit(E_GPC,12);
    DrvGPIO_ClrBit(E_GPB,11);
}

```

Langkah kesepuluh : **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*.

6. Diklik 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 Pendahuluan

1. Tugas Pendahuluan dilaksanakan secara lisan melalui gmeet.

2.2 Tugas Unit

1. Apa itu ADC dan pengaplikasiannya pada mikroprosesor?
2. Apa fungsi dari *listing* a=1?

Jawaban

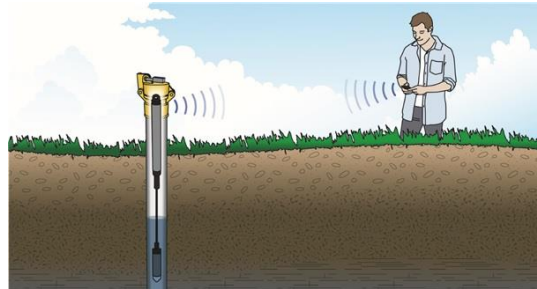
1. ADC atau *Analog to Digital Converter* merupakan alat atau perangkat elektronika yang digunakan untuk mengubah sinyal yaitu dari sinyal analog menjadi sinyal digital
2. a=1 didapat saat *push button* ditekan yang mana akan menjalankan fungsi *while* dibawahnya.

BAB III

ANALISI

3.1 Dasar Teori

Saat ini, ketinggian air dapat diukur secara mudah dengan menggunakan alat modern seperti Water Level. Pengertian Water Level sendiri adalah seperangkat alat yang digunakan untuk mengukur ketinggian air di tempat yang berbeda agar mendapatkan data perbandingan. Water level yang paling sederhana adalah sepasang pipa yang saling terhubung di bagian bawah. Water level sederhana akan mengukur ketinggian air melalui tinggi air di kedua pipa apakah sama atau tidak. Hasil pengukuran dari water level lebih rendah dari menggunakan laser tetapi water level mempunyai akurasi yang tinggi dalam pengukuran jarak jauh. Untuk menghindari kesalahan pengukuran dalam penggunaan water level, suhu pada air haruslah sama.



Gambar 3.2.1 Simulasi Water Level[2].

Water level dapat juga digunakan untuk mengukur tekanan air dengan menggunakan prinsip tekanan Hidrostatik. Air dalam suatu wadah selalu mendapatkan tekanan dari atmosfer dan sebanding dengan level dari air sehingga bisa didapatkan besar tekanan air. Saat ini, sudah ada water level yang lebih modern dimana water level modern dapat mengukur ketinggian dan tekanan air secara bersamaan dengan sensor dan hasil pengukurannya dapat direkam kemudian disimpan dalam bentuk data. Alat tersebut

disebut dengan Sensor Water Level. Salah satu alat sensor water level adalah Water Level HOBO KIT-D-U20-04[2].

3.2 Analisi Percobaan *Water Level*

Terdapat 7 Percobaan pada Unit 8 *Water Level* yang mana percobaan tersebut menghasilkan keterangan yang dapat dilihat pada tabel dibawah ini:

Tabel 3.2.1 Blangko Unit 7

No.	Percobaan	Keterangan
1.	Putar variable resistor searah jarum jam.	<i>Buzzer</i> menyala saat ADC = 0000 dengan keterangan <i>LOW</i> dan lampu LED tidak ada yang menyala, saat berada pada diatas 1000 yaitu pada keadaan <i>NORMAL</i> dan <i>buzzer</i> berhenti bersuara dan LED pada bagian paling kanan menyala pada saat 1000 keatas, LED kedua menyala pada 2000 Keatas, LED ketiga menyala pada 3000 Keatas, dan pada saat diatas 4000 <i>buzzer</i> kembali menyala dengan keterangan <i>HIGH</i> dengan LED menyala keempat-empatnya

2.	<p>Amati hasil nilai ADC berdasarkan listing dibawah ini</p> <pre>Show_Word(0,5,adc/1000+'0'); Show_Word(0,6,adc%1000/100+'0'); Show_Word(0,7,adc%100/10+'0'); Show_Word(0,8,adc%10+'0');</pre>	<p>Ketika nilai ADC berada pada diatas 1000 dan dibawah 2000, LED yang menyala hanya satu dan berada pada keadaan <i>NORMAL</i></p>
3.	<p>Ubah listing dibawah ini</p> <pre>Show_Word(0,5,adc/1000+'0'); Show_Word(0,6,adc%1000/100+'0'); Show_Word(0,7,adc%100/10+'0'); Show_Word(0,8,adc%10+'0');</pre> <p>Menjadi</p> <pre>Show_Word(0,5,adc/4000+'0'); Show_Word(0,6,adc%4000/400+'0'); Show_Word(0,7,adc%400/40+'0'); Show_Word(0,8,adc%4+'0');</pre> <p>Amati perubahan nilai ADC kemudian buktikan</p>	<p>Pada percobaan ini saat menunjukan angka dibawah 0500 merupakan keadaan <i>NORMAL</i> dengan LED paling kanan menyala, namun saat diatas 0500 LED kedua menyala, diatas 0700 LED ketiga menyala dan saat angka menyentuh diatas 1000 maka <i>buzzer</i> menyala dan LED menyala semua dengan keterangan pada LCD yaitu <i>HIGH</i></p>

3.2.1 Percobaan 1

Pada percobaan pertama kita diminta untuk mutar variable resistor searah jarum jam, dan dihasilkan *Buzzer* menyala saat $ADC = 0000$ dengan keterangan *LOW* dan lampu LED tidak ada yang menyala, saat berada pada diatas 1000 yaitu pada keadaan *NORMAL* dan *buzzer* berhenti bersuara dan LED pada bagian paling kanan menyala pada saat 1000 keatas, LED kedua menyala pada 2000 Keatas, LED ketiga menyala pada 3000 Keatas, dan pada saat diatas 4000 *buzzer* kembali menyala dengan keterangan *HIGH* dengan LED menyala keempat-empatnya, terdapat 2 *while* pada *listing program* yang mana program

akan mengeksekusi `while(a==0)` lebih dulu, karena kita telah mendeklarasikan di awal yaitu `a==0` yang mana program akan mengeksekusi `a==0` terlebih dahulu, lalu terdapat `while(a==1)` yang mana merupakan program yang akan di eksekusi ketika push button ditekan, nuvoton merupakan rangkaian aktif high yang mana jika diberi logika 1 dia akan hidup dan diberi logika 0 dia akan mati. Terdapat juga perintah `StartConvert` yang merupakan perintah untuk memulai konversi dari sinyal analog ke sinyal digital, terdapat pula variable ADC yang mana berguna untuk menyimpan perintah `getconversiondata` pada channel 7, jadi `getconversiondata` adalah perintah saat memutarakan variable resistor.

3.2.2 Percobaan 2

Pada percobaan kedua yaitu amati hasil nilai ADC berdasarkan listing dibawah ini

```
Show_Word(0,5,adc/1000+'0'); lalu ada Show_Word(0,6,adc%1000/100+'0');
Show_Word(0,7,adc%100/10+'0'); lalu Show_Word(0,8,adc%10+'0'); dan
```

didapatkan hasil yaitu Ketika nilai ADC berada pada diatas 1000 dan dibawah 2000, LED yang menyala hanya satu dan berada pada keadaan NORMAL, dibawah ini merupakan penghitungan dengan nilai ADC adalah 1653, perhitungan digunakan dengan tujuan apakah LCD akan menampilkan angka pada ADC sesuai dengan nilai pada ADC itu sendiri:

yaitu nilai ADC dibagi dengan 1000 karena terdapat `Show_Word(0,5,adc/1000+'0');` pada *listing program*:

$$\frac{1653}{1000} = 1 \text{ (Tersisa 653)}$$

Disini dipatkan nilai 1 yang mana akan ditunjukan pada LCD yaitu di kolom ke 5, dan masih tersisa 653 dari pembagian tersebut, terdapat juga alasan mengapa ditaruh pada kolom ke 5, dikarenakan pada kolom 1 hingga kolom 3 telah digunakan untuk “ADC=” yang mana terdapat 3 huruf dan 1 simbol dalam kolom tersebut jadi kolom 1 hingga 4 telah digunakan, lalu dibagi dengan 100 karena terdapat `Show_Word(0,6,adc%1000/100+'0');` pada *listing program*:

$$\frac{653}{100} = 6 \text{ (Tersisa 53)}$$

Disini dipatkan nilai 6 yang mana akan ditunjukkan pada LCD yaitu di kolom ke 6, dan masih tersisa 53 dari pembagian tersebut, lalu dibagi dengan 10 karena terdapat `Show_Word(0,7,adc%100/10+'0')`; pada *listing program*:

$$\frac{53}{10} = 5 \text{ (Tersisa 3)}$$

Disini dipatkan nilai 5 yang mana akan ditunjukkan pada LCD yaitu di kolom ke 7, dan masih tersisa 3 dari pembagian tersebut, lalu kolom ke 8 akan menunjukkan angka pada penyisaan pembagian tersebut yang mana terdapat `Show_Word(0,8,adc%10+'0')`; jadi angka 3 itu akan berada pada kolom ke 8, jadi didapatkan disimpulkan LCD dapat menampilkan nilai yang ada pada ADC.

3.2.3 Percobaan 3

Pada percobaan terakhir ini kita diminta untuk mengubah listing dibawah ini

```
Show_Word(0,5,adc/1000+'0');
Show_Word(0,6,adc%1000/100+'0');
Show_Word(0,7,adc%100/10+'0');
Show_Word(0,8,adc%10+'0');
```

Menjadi

```
Show_Word(0,5,adc/4000+'0');
Show_Word(0,6,adc%4000/400+'0');
Show_Word(0,7,adc%400/40+'0');
Show_Word(0,8,adc%4+'0');
```

Amati perubahan nilai ADC kemudian buktikan hasilnya, didapatkan: Pada percobaan ini saat menunjukkan angka dibawah 0500 merupakan keadaan *NORMAL* dengan LED paling

kanan menyala, namun saat diatas 0500 LED kedua menyala, diatas 0700 LED ketiga menyala dan saat angka menyentuh diatas 1000 maka *buzzer* menyala dan LED menyala semua dengan keterangan pada LCD yaitu *HIGH*, pada percobaan ke-3 juga dilakukan pengujian dengan perhitungan ADC adalah 1653:

yaitu nilai ADC dibagi dengan 4000 karena terdapat `Show_Word(0,5,adc/4000+'0');` pada *listing program*:

$$\frac{1653}{4000} = 0 \text{ (Tersisa 1653)}$$

Disini dipatkan nilai 0 yang mana akan ditunjukkan pada LCD yaitu di kolom ke 5, dan masih tersisa 1653 dari pembagian tersebut, lalu dibagi dengan 400 karena terdapat `Show_Word(0,6,adc%4000/400+'0');` pada *listing program*:

$$\frac{1653}{400} = 4 \text{ (Tersisa 53)}$$

Disini dipatkan nilai 4 yang mana akan ditunjukkan pada LCD yaitu di kolom ke 6, dan masih tersisa 53 dari pembagian tersebut, lalu dibagi dengan 10 karena terdapat `Show_Word(0,7,adc%400/40+'0');` pada *listing program*:

$$\frac{53}{40} = 1 \text{ (Tersisa 13)}$$

Disini dipatkan nilai 5 yang mana akan ditunjukkan pada LCD yaitu di kolom ke 7, dan masih tersisa 13 dari pembagian tersebut, lalu kolom ke 8 akan menunjukkan angka pada penyisaan pembagian tersebut yang mana terdapat `Show_Word(0,8,adc%4+'0');` yaitu hasil sisa dibagi 4, maka

$$\frac{13}{4} = 3 \text{ (Tersisa 1)}$$

Dan pada kolom ke 8 akan menampilkan angka 1, jadi dihasilkan 0413, jadi itu menjadi alasan mengapa nilai yang dihasilkan pada LCD tidak sesuai dengan ketentuan yang ada pada *listing program*, yang mana pada *listing program* bahwa nilai dibawah 1000 merupakan nilai low namun disini masuk pada nilai

normal dengan LCD menyala 1, karena program mengira bahwa nomor yang ditunjukan pada LCD merupakan angka pada ADC, padahal angka tersebut bukan merupakan nilai asli pada ADC.

BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan pada praktikum yang telah dilakukan, mengenai *Water Level* dapat diambil kesimpulan sebagai berikut:

1. Pada *Water Level* ini kita dapat mengetahui bahwa Nilai yang muncul paa LCD merupakan nilai asli pada ADC dengan pembagian yang terdapat pada *listing program*.
2. Penghitungan ADC dapat dilakukan dengan pembagian bertahap pada angka dinilai ADC tersebut, yaitu pembagian 1000 merupakan tahap pertama, lalu pembagian 100 untuk tahap kedua, Pembagian 10 tahap ketiga, dan hasil yang sisa pembagian tersebut pada tahap keempat.

DAFTAR PUSTAKA

- [1].Asisten Lab Dasar Elektro, "*Water Level*" in Modul *Praktikum Dasar Mikroprosesor 2021*, Cilegon, Universitas Sultan Ageng Tirtayasa Fakultas Teknik, 2021 , pp 59 – 63.
- [2].Taharica, "*Pengertian Senor Water Level dan Cara Kerja*", in Website Loggerindo[terhubung berkala] <https://www.loggerindo.com/pengertian-sensor-water-level-dan-cara-kerja-109> (diakses pada 29 September 2021 pukul 01.34)