



**LABORATORIUM PENGUKURAN FISIS
JURUSAN TEKNIK FISIKA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**MODUL III
MIKROKONTROLER
(SISTEM I/O, ADC DAN INTERUPSI EKSTERNAL)**

A. TUJUAN

1. Pengenalan secara umum dasar-dasar mikrokontroler AVR.
2. Mengetahui dan memahami sistem input/output (I/O) dan implementasinya pada mikrokontroler.
3. Mengetahui dan memahami ADC dan implementasinya pada mikrokontroler.
4. Mengetahui dan memahami interupsi eksternal dan implementasinya pada mikrokontroler.

B. DASAR TEORI

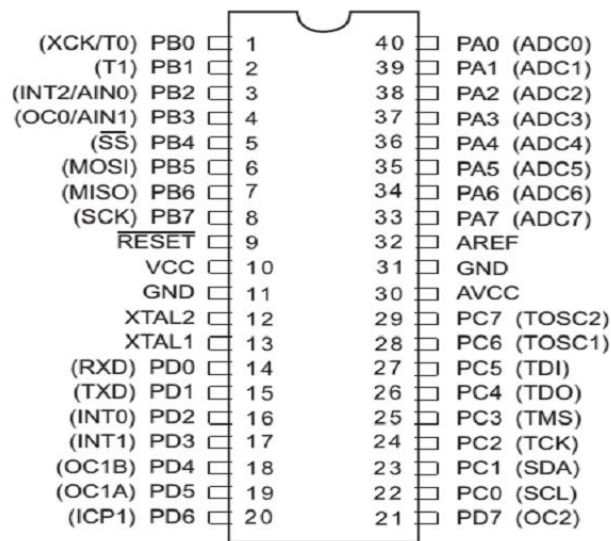
1. Mikrokontroler ATMEGA

Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bias ditulis dan dihapus dengan cara khusus. Mikrokontroler merupakan salah satu perkembangan teknologi yang mengintegrasikan sebuah sistem computer kedalam sebuah chip tunggal (Single Chip), sehingga teknologi ini mampu berfungsi seperti sebuah sistem komputer, salah satunya adalah pada proses pengendalian. Salah satu contoh-contoh chip mikrokontroller yang dipakai dalam praktikum ini adalah ATMEGA 16 dan ATMEGA 8535, dimana chip mikrokontroller ATMEGA 16 ini mempunyai fitur-fitur sebagai berikut:

- 8 Bit Mikrokontroller
- 16 Kbytes Flash
- 512 Bytes EEPROM
- 1 Kbyte internal SRAM
- Clock speed up to 16 MHz
- 8 Channel, 10 Bit ADC
- 3 PWM Channel
- Programmable Serial USART
- External and Internal Interrupt Sources
- 30 pin input/output
- Operating Voltages
 - 2.7V - 5.5V (ATmega16L)
 - 3.5V - 5.5V (ATmega16)



LABORATORIUM PENGUKURAN FISIS
JURUSAN TEKNIK FISIKA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER

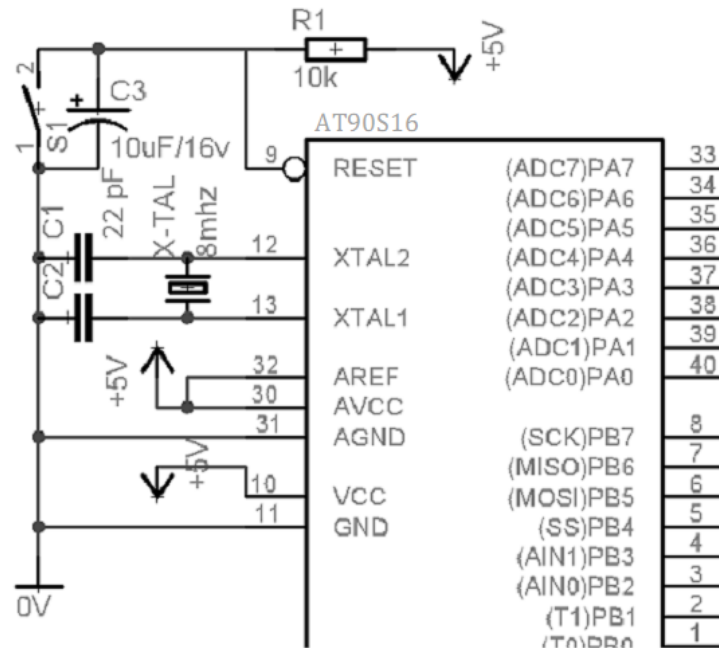


Gambar 3.1 Konfigurasi Pin ATMEGA 16

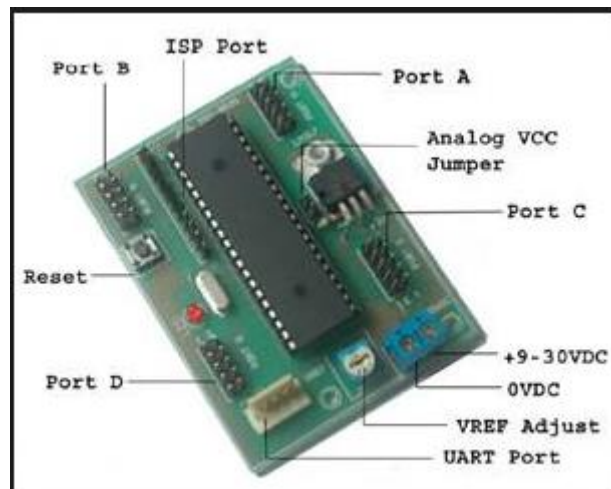
Penjelasan dari masing-masing pin adalah sebagai berikut:

- a. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
- b. GND merupakan pin ground.
- c. *Port A* (PA7...PA0) merupakan terminal masukan analog menuju A/D Converter. Port ini juga berfungsi sebagai port I/O 8 bit dua arah. Jika ADC tidak diaktifkan.
- d. *Port B* (PB7..PB0) merupakan port I/O 8 bit dua arah dengan resistor pull-up internal. Port B juga dapat berfungsi sebagai terminal khusus yaitu timer/counter, komparator analog dan SPI.
- e. *Port C* (PC7..PB0) merupakan port I/O 8 bit dua arah dengan resistor pull-up internal. Port C juga dapat berfungsi sebagai terminal khusus yaitu komparator analog, dan timer Oscilator.
- f. *Port D* (PD7..PD0) adalah merupakan port I/O 8 bit dua arah dengan resistor pull-up internal. Port D juga berfungsi sebagai terminal khusus yaitu komparator analog, interupsi eksternal, dan komunikasi serial.
- g. RESET merupakan pin yang digunakan untuk mereset mikrokontroler.
- h. XTAL1 dan XTAL2 merupakan pin masukan clock eksternal.
- i. AVCC merupakan pin masukan tegangan untuk ADC.
- j. AREF merupakan pin masukan tegangan referensi untuk ADC.

Sebuah mikrokontroler tidak bisa menjalankan fungsi smart device jika tidak terpasang dalam rangkaian minimum sistem yang ditunjukkan pada Gambar 3.2. Rangkaian minimum sistem ini adalah rangkaian dimana sebuah chip mikrokontroler dapat menjalankan perintah yang telah diprogram. Rangkaian minimum system ini terdiri dari pasokan tegangan yang memasok daya ke mikrokontroler, selain itu ada sebuah crystal yang berfungsi untuk memberikan osilasi waktu untuk dapat bekerja dengan variabel waktu secara tepat.



Gambar 3.2 Rangkaian Minimum System ATmega 16



Gambar 3.3 Minimum System ATmega 16

1.1 Sistem I/O (Input/Output)

Port I/O pada mikrokontroler ATmega 16 dapat difungsikan sebagai input ataupun output dengan keluaran *high* atau *low*. Pada dasarnya untuk mengakses I/O (input output) AVR ada beberapa register yang harus disetting terlebih dahulu, yaitu:

a. Data Register (PORTx)

PORTx digunakan untuk mengeluarkan/menulis data ke port mikrokontroler, saat dikonfigurasi sebagai output.

Contoh: $PORTB = 0xff \rightarrow$ maka Port B = 11111111 (biner)

$PORTB = 0xa3 \rightarrow$ maka Port B = 10100011 (biner)



LABORATORIUM PENGUKURAN FISIS
JURUSAN TEKNIK FISIKA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER

b. Data Direction Register (DDRx)

DDRx digunakan untuk mendefinisikan port sebagai input atau output, Jika level logika pada register DDR bernilai 1 maka Port tersebut dikonfigurasi sebagai Output. Sedangkan jika level logika pada register DDR bernilai 0 maka Port tersebut dikonfigurasi sebagai input.

Contoh: DDRA=0xff → maka Port A sebagai Output

DDRA=0x00 → maka Port A sebagai Input

c. Port Input Pin (PINx)

PINx digunakan untuk mengambil/membaca data dari port mikrokontroler, saat dikonfigurasi sebagai input.

1.2 Analog to Digital Converter (ADC)

ADC digunakan sebagai rangkaian yang mengubah sinyal analog menjadi sinyal digital. Dengan menggunakan ADC, kita dapat mengamati sinyal-sinyal dari perubahan-perubahan sinyal analog seperti perubahan temperatur, kepekatan asap, tekanan udara, kecepatan angin, berat benda, kadar asam dan lain-lain yang semuanya dapat diamati melalui sensornya masing-masing. Hal yang paling penting dalam suatu rangkaian ADC adalah resolusi, yaitu besaran analog terkecil yang masih dapat dikonversi menjadi satuan digital.

Atmega memiliki resolusi ADC 10 bit (dapat juga menggunakan ADC 8 bit) dengan 8 channel (PA0-PA7) input ADC dan mendukung 16 macam penguat beda. ADC ini bekerja dengan teknik *successive approximation*. Rangkaian internal ADC memiliki catu daya tersendiri yaitu pin AVCC.

Resolusi dari sebuah converter menunjukkan banyaknya nilai diskrit yang dapat dihasilkan pada skala tegangan tertentu. Resolusi biasanya dinyatakan dalam bit (binary digit). Sebagai contoh, sebuah ADC yang mengkodekan sebuah masukan analog menjadi salah satu dari 256 nilai diskrit mempunyai resolusi 8 bit karena antar muka antara mikrokontroler dengan komputer, dilakukan dengan cara memasang rangkaian komunikasi serial, sehingga ada tautan antara mikrokontroler dengan komputer, misal, suatu sensor yang dibaca oleh mikrokontroler, dapat diamati di komputer yaitu data 8 bit, sehingga data yang tertampil dalam komputer dapat terlihat secara real-time. Rumus ADC n-bit:

$$\text{ADC} = (\text{Vin} \times 2^n) / \text{Vref}$$

1.3 Interupsi

Interupsi adalah kondisi dimana mikrokontroler akan berhenti sementara dari program utama untuk melayani perintah-perintah pada interupsi, setelah selesai maka mikrokontroler akan kembali mengerjakan instruksi pada program utama.

Untuk Atmega 16 dan 8535 ada 3 sumber interupsi eksternal yaitu INTO (PD2), INT1 (PD3) dan INT2 (PB2). Ketiga interupsi itu dapat dipicu dengan adanya perubahan level baik transisi naik (rising edge) maupun transisi turun (falling edge). Untuk pengaturan mode dan cara kerjanya interupsi eksternal dilakukan melalui 2 buah register MCUCR dan register MCUCSR. Untuk lebih jelasnya tentang register-register tersebut lebih baik dibaca datasheetnya. Tetapi jika



**LABORATORIUM PENGUKURAN FISIS
JURUSAN TEKNIK FISIKA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

menggunakan compiler CodeVision AVR, kita tidak perlu dituntut untuk tau tentang kedua register tersebut, karena secara otomatis CodeVision AVR akan mensettingnya saat kita menset CodeWizard AVR, dimana penggunaan CodeWizard AVR akan dijelaskan pada bagian berikutnya.

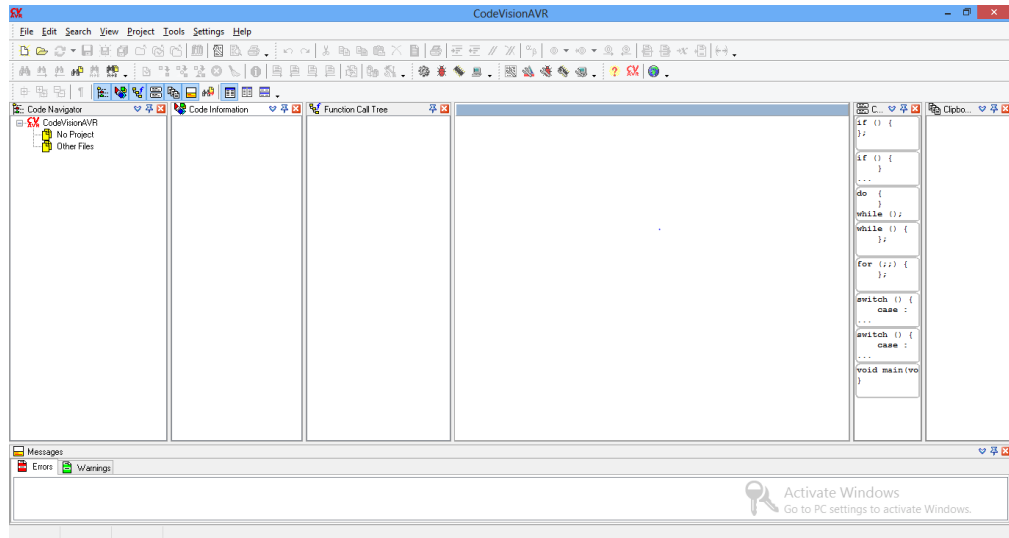
2. CODEVISION AVR

CodeVision AVR compiler (CVAVR) merupakan compiler bahasa C untuk AVR. Compiler ini cukup memadai untuk belajar AVR, karena mudah penggunaannya juga didukung berbagai fitur yang sangat membantu dalam pembuatan software untuk keperluan pemrograman AVR. CVAVR ini dapat mengimplementasikan hampir semua instruksi bahasa C yang sesuai dengan arsitektur AVR, bahkan terdapat beberapa keunggulan tambahan untuk memenuhi keunggulan spesifik AVR.

CodeVisionAVR juga mempunyai Automatic Program Generator bernama CodeWizardAVR, yang mengizinkan Anda untuk menulis, dalam hitungan menit, semua instruksi yang diperlukan untuk membuat fungsi-fungsi berikut:

- Set-up akses memori eksternal
- Identifikasi sumber reset untuk chip
- Inisialisasi port input/output
- Inisialisasi interupsi eksternal
- Inisialisasi Timer/Counter
- Inisialisasi Watchdog-Timer
- Inisialisasi UART (USART) dan komunikasi serial berbasis buffer yang digerakkan oleh interupsi
- Inisialisasi Pembanding Analog
- Inisialisasi ADC
- Inisialisasi Antarmuka SPI
- Inisialisasi Antarmuka Two-Wire
- Inisialisasi Antarmuka CAN
- Inisialisasi Bus I2C, Sensor Suhu LM75, Thermometer/Thermostat DS1621 dan Real-Time Clock PCF8563, PCF8583, DS1302, dan DS1307
- Inisialisasi Bus 1-Wire dan Sensor Suhu DS1820, DS18S20
- Inisialisasi modul LCD

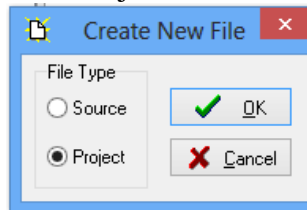
CodeVisionAVR merupakan hak cipta dari Pavel Haiduc, HP InfoTech s.r.l.



Gambar 3.3 Tampilan Awal CodeVision AVR

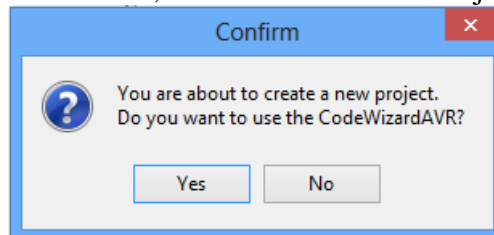
Untuk memulai membuat program pada CVAVR langkah-langkah yang harus dilakukan yaitu:

1. Klik File > New, maka akan muncul jendela berikut :



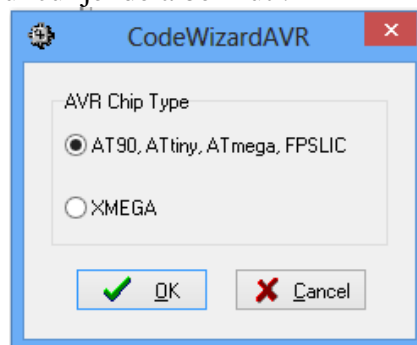
Gambar 3.5 Jendela Create New File

2. Pilih Project, kemudian klik OK, kemudian akan muncul jendela berikut :



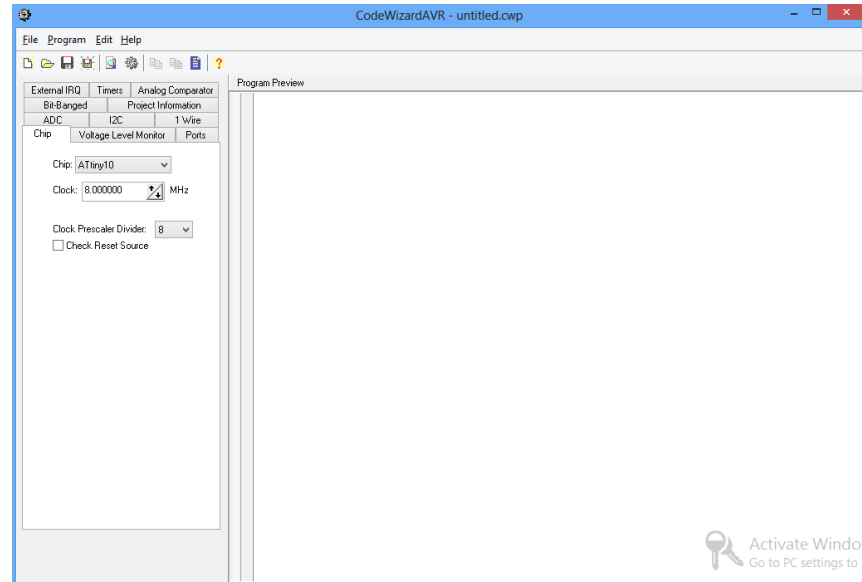
Gambar 3.6 Jendela Confirm

3. Klik Yes, maka akan muncul jendela berikut :



Gambar 3.6 Jendela konfirmasi chip

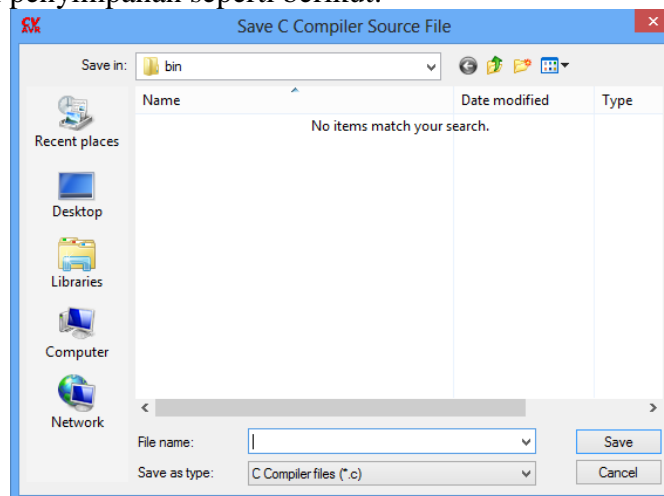
4. Pilih AT90, ATtiny, ATmega, FPSLIC, Klik OK, maka akan muncul jendela berikut :



Gambar 3.7 Tampilan CodeWizard AVR

Pada CodeWizard tersebut kita dapat melakukan inisiasi dari parameter-parameter mikrokontroler seperti jenis chip mikrokontroler, frekwensi Kristal/clock, maupun inisiasi dari settingan register-register dari mikrokontroler yang ingin kita akses, misalnya ADC, timer, counter, interupsi, dll, serta dapat juga digunakan untuk mengkondisikan port I/O mikrokontroler untuk LCD.

5. Setelah muncul form tersebut, silahkan atur tab-tab pada form tersebut sesuai dengan kebutuhan (nilai clock dibuat = 12.0000Mhz), dan setelah selesai Klik pada menu Program > "Generate Program Save and Exit" dan kemudian muncul form penyimpanan seperti berikut:



Gambar 3.8 Jendela save



**LABORATORIUM PENGUKURAN FISIS
JURUSAN TEKNIK FISIKA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

6. Simpan ketiga file tersebut (file dengan format *.c, *.pjr, dan *.cwp) dengan nama dan folder yang sama, setelah itu akan muncul source code dasar, seperti berikut :

```
2 This program was produced by the
3 CodeWizardAVR V2.05.3 Standard
4 Automatic Program Generator
5 © Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
6 http://www.hpinfotech.com
7
8 Project :
9 Version :
10 Date   : 1/21/2015
11 Author : Dionisius
12 Company :
13 Comments:
14
15
16 Chip type           : ATmega16
17 Program type        : Application
18 AVR Core Clock frequency: 12,000,000 MHz
19 Memory model        : Small
20 External RAM size    : 0
21 Data Stack size     : 256
22 *****/
23
24 #include <mega16.h>
25
26 #include <delay.h>
27
```

Gambar 3.9 Tampilan Source Code Dasar

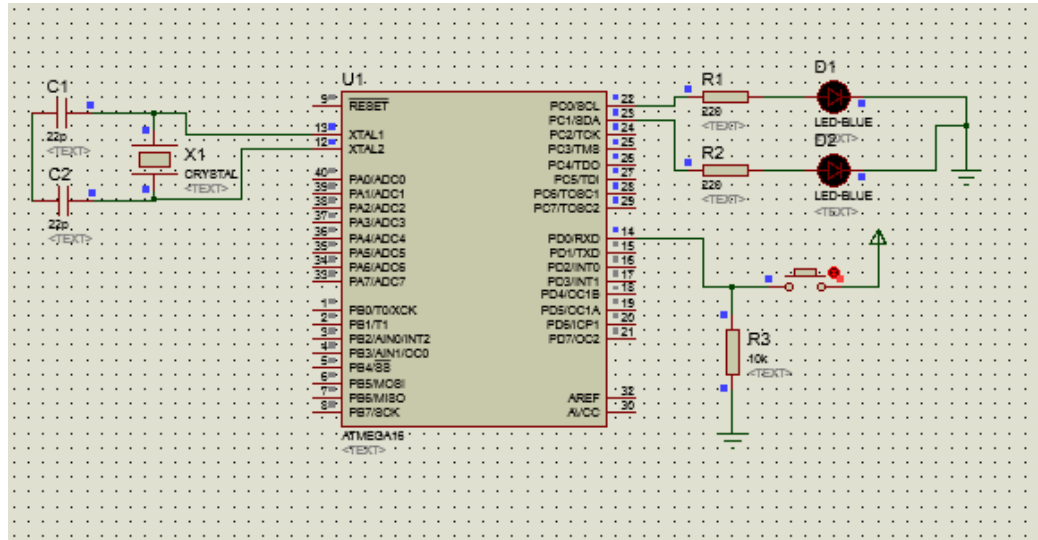
C. PERALATAN

1. Breadboard
2. Minimum Sistem ATmega 16
3. Downloader.
3. LED.
5. Power Supply 1 buah.
6. Kabel jumper.
7. LCD.
8. Potentiometer/trimpot.
9. Push Button,
10. Resistor.

D. PROSEDUR PERCOBAAN

Digital Input/Output (I/O)

1. Buat rangkaian dengan LED, resistor dan push button seperti pada gambar dibawah, lalu hubungkan dengan minimum sistem atmega sesuai dengan dengan port yang sesuai dengan gambar dibawah.



Gambar 3.10 Rangkaian LED dan push button

2. Buka CodeVision AVR, buat project baru
3. Pada CodeWizard, pada tab Chip, pilih Atmega 16, dan isikan frekwensi clock sebesar 12.000000 Mhz

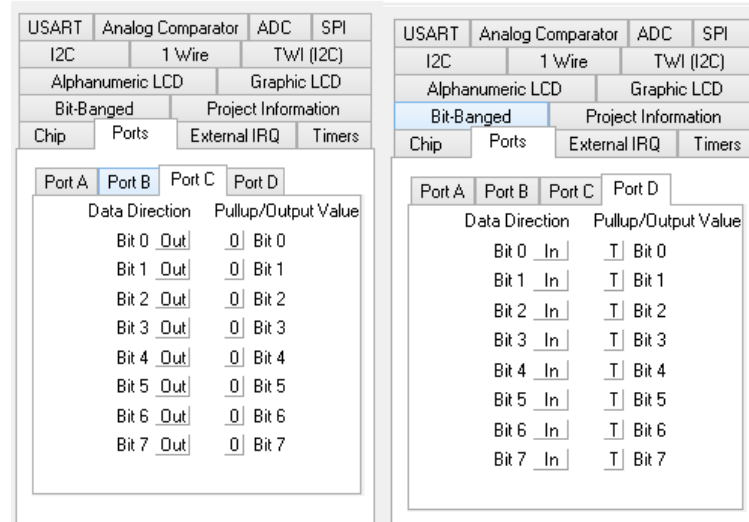
USART	Analog Comparator	ADC	SPI
I2C	1 Wire	TWI (I2C)	
Alphanumeric LCD		Graphic LCD	
Bit-Banged		Project Information	
Chip	Ports	External IRQ	Timers
Chip: ATmega16			
Clock: 12.000000 MHz			
<input type="checkbox"/> Check Reset Source			
Program Type: Application			

Gambar 3.11 Tab Chip



LABORATORIUM PENGUKURAN FISIS
JURUSAN TEKNIK FISIKA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER

4. Pilih tab port dan inisiasikan port C sebagai output, dan port D sebagai input.



Gambar 3.12 Pengkondisian Port C dan Port D sebagai Input dan Output

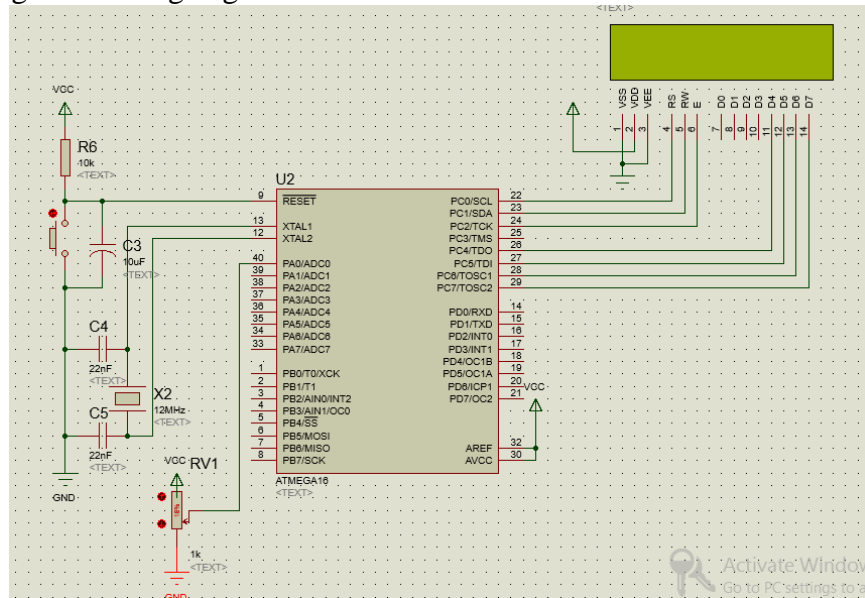
5. Generate program, isikan script dibawah ini pada program utama

```
while (1){  
    if (PIND.0==1)  
    {PORTC.0=1;  
    PORTC.1=1;  
    }  
    else  
    {PORTC.0=0;  
    PORTC.1=0;}  
}
```

6. Compile program tersebut dengan meng klik “compile the project”, Setelah itu masukkan file “*.hex” yang berada pada folder “bin” ke mikrokontroller dengan menggunakan software khazama.
7. Amati perubahan pada LED
8. Ulangi langkah-langkah tersebut dengan mengganti masukan program dari (PIND.0==1) menjadi (PIND.0==0)
9. Amati perubahan yang terjadi dan simpulkan

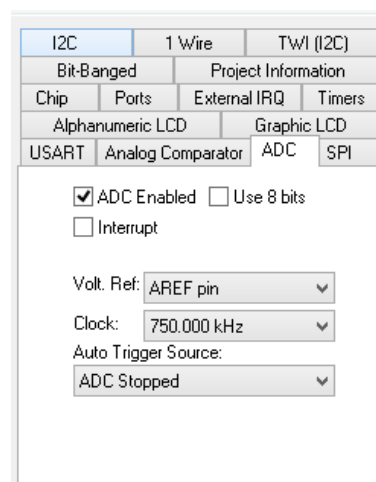
Analog to Digital Converter (ADC)

1. Buat rangkaian dengan LCD dan Potentiometer/trimpot seperti pada gambar dibawah, lalu hubungkan dengan minimum sistem atmega sesuai dengan dengan port yang sesuai dengan gambar dibawah.



Gambar 3.13 Rangkaian Percobaan ADC

2. Buka CodeVision AVR, buat project baru
3. Pada CodeWizard, pada tab Chip, pilih Atmega 16, dan isikan frekwensi clock sebesar 12.000000 Mhz
4. Pilih tab ADC, beri tanda centang pada ADC enabled seperti pada gambar dibawah ini

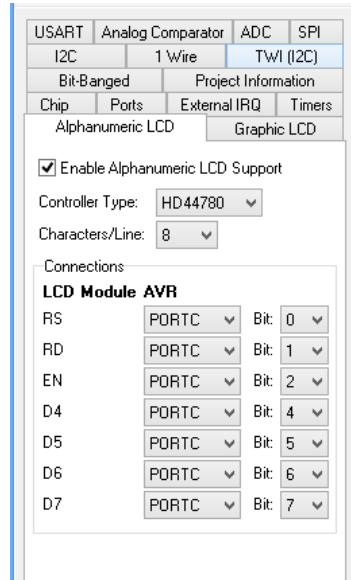


Gambar 3.13 Setting Tab ADC



LABORATORIUM PENGUKURAN FISIS
JURUSAN TEKNIK FISIKA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER

5. Pilih Tab Alphanumeric LCD, setting port c sebagai port output untuk LCD seperti pada gambar berikut.



Gambar 3.15 Setting Tab LCD

6. Generate program, tambahkan library `#include <stdlib.h>` dan Deklarasikan `int ADC` serta `char temp[6];` seperti pada gambar dibawah ini

```
int ADC;  
char temp[16];  
  
#include <mega16.h>  
#include <stdlib.h>  
#include <delay.h>
```

Gambar 3.16 Penambahan library

7. Tambahkan script dibawah ini pada program utama.

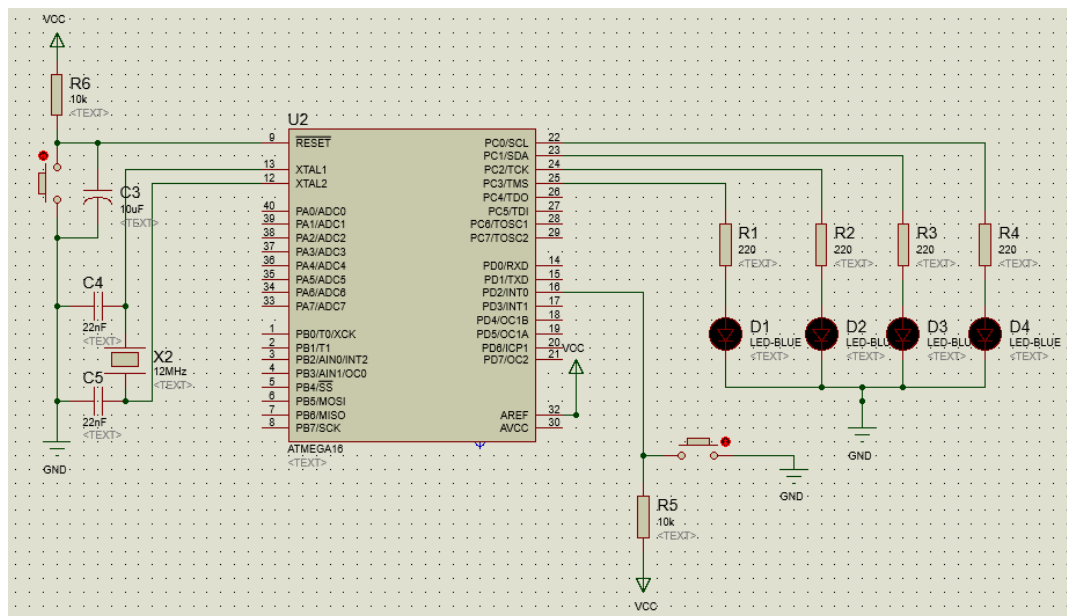
```
while (1)
{
    ADC=read_adc(0);
    lcd_clear( );
    lcd_gotoxy(0,0);
    lcd_putsf("Nilai ADC");
    itoa(ADC,temp);
    lcd_gotoxy(0,1) ;
    lcd_puts(temp);
    delay_ms(500);
}
```

8. Compile program tersebut dengan meng klik “compile the project”, Setelah itu masukkan file “*.hex” yang berada pada folder “bin” ke mikrokontroller dengan menggunakan software khazama.

9. Putar potentiometer/trimpot, amati perubahan yang terjadi dan simpulkan.

Interupsi Eksternal

1. Buat rangkaian dengan LED dan push button seperti pada gambar dibawah, lalu hubungkan dengan minimum sistem atmega sesuai dengan dengan port yang sesuai dengan gambar dibawah.



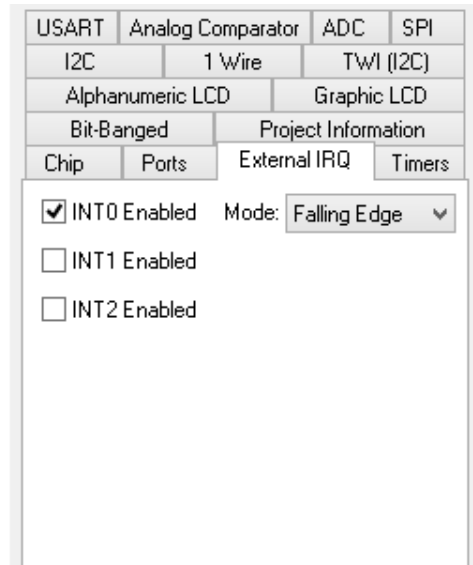
Gambar 3.17 Rangkaian Percobaan Interupsi Eksternal

2. Buka CodeVision AVR, buat project baru



LABORATORIUM PENGUKURAN FISIS
JURUSAN TEKNIK FISIKA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER

3. Pada CodeWizard, pada tab Chip, pilih Atmega 16, dan isikan frekwensi clock sebesar 12.000000 Mhz
4. Pilih tab Port dan kondisikan port C sebagai output.
5. Pilih tab External IRQ, beri tanda centang pada INT0 Enabled yang berarti mengaktifkan interupsi eksternal INT0 serta pilih mode Falling edge seperti pada gambar dibawah



Gambar 3.18 Setting Tab External IRQ

6. Generate program, tambahkan library #include <delay.h>
7. Tambahkan script berikut pada bagian interrupt

```
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    PORTC=0XFF;
    delay_ms(1000);
    PORTC=0X00;
    delay_ms(1000);
    PORTC=0XFF;
    delay_ms(1000);
    PORTC=0X00;
    delay_ms(1000);
    PORTC=0XFF;
    delay_ms(1000);
}
```



LABORATORIUM PENGUKURAN FISIS
JURUSAN TEKNIK FISIKA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER

8. Tambahkan script berikut pada program utama

```
while (1)
{
    PORTC=0X0F;
    delay_ms(500);
    PORTC=0X07;
    delay_ms(500);
    PORTC=0X03;
    delay_ms(500);
    PORTC=0X01;
    delay_ms(500);
    PORTC=0X00;
    delay_ms(500);
}
```

9. Compile program tersebut dengan meng klik “compile the project”, Setelah itu masukkan file “*.hex” yang berada pada folder “bin” ke mikrokontroller dengan menggunakan software khazama.
10. Saat rangkaian di jalankan maka LED akan menyala berjalan, dan kemudian saat terjadi interupsi, yaitu saat push button ditekan maka LED akan berkedip sebanyak 3 kali. Setelah interupsi selesai dijalankan maka program utama akan kembali dijalankan. Ubah tipe interupsi menjadi rising edge dan analisa perubahan nya.