# Interupsi

Yang harus diperhatikan untuk menguanakan interupsi adalah, kita harus tau sumber-sumber interupsi, vektor layanan interupsi dan yang terpenting rutin lyanan interupsi, yaitu subrutin yang akan dikerjakan bila terjadi interupsi .

* 1. **Interrupt Service Routine**.

Interrupt adalah suatu kejadian atau peristiwa yang menyebabkan mikrokontroler berhenti sejenak untuk melayani interrupt tersebut.

Analoginya adalah sebagai berikut, seseorang sedang mengetik laporan, mendadak telephone berdering dan menginterrupsi orang tersebut sehingga menghentikan pekerjaan mengetik dan mengangkat telephone. Setelah pembicaraan telephone yang dalam hal ini adalah merupakan analogi dari Interrupt Service Routine selesai maka orang tersebut kembali meneruskan pekerjaanya mengetik.

Demikian pula pada sistem mikrokontroler yang sedang menjalankan programnya, saat terjadi interrupt, program akan berhenti sesaat, melayani interrupt tersebut dengan menjalankan program yang berada pada alamat yang ditunjuk oleh vektor dari interrupt yang terjadi hingga selesai dan kembali meneruskan program yang terhenti oleh interrupt tadi. Seperti yang terlihat Gambar di bawah, sebuah program yang seharusnya berjalan terus lurus, tiba-tiba terjadi interrupt dan harus melayani interrupt tersebut terlebih dahulu hingga selesai sebelum ia kembali meneruskan pekerjaannya.



AVR menyediakan beberapa sumber interupsi yang berbeda. Tiap-tiap interupsi dan reset memiliki vektor program yang berbeda. Semua interupsi didasari satu bit tunggal yang harus diberi logika tinggi sebagai *Global Interrupt Enable* pada Status Register untuk mengaktifkan interupsi.

Atmega 16 menyediakan 21 macam sumber interupsi yang masing-masing memiliki alamata vektor interupsi seperti pada tabel 19. Setiap interupsi yang aktif akan dilayani segera setelah terjadi permintaan interupsi, tapi jika dalam waktu bersamaan terjadi lebih dari satu interupsi maka perioritas yang akan diselesaikan terlebih dahulu adalah interupsi yang memiliki urut lebih kecil sesuai tabel 19 berikut.



Pada AVR terdapat 3 pin interupsi eksternal, yaitu INT0,INT1,dan INT2. Interupsi eksternal dapat dibangkitkan apabila ada perubahan logika baik transisi naik (rising edge) maupun transisi turun (falling edge) pada pin interupsi. Pengaturan kondisi keadaan yang menyebabkan terjadinya interupsi eksternal diatur oleh 2 buah register I/O yaitu MCUCR dan register MCUCSR

**MCUCR** ( MCU Control Register), mengatur pemicu interupsi dan fungsi MCU secara umum.

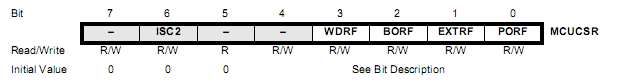


Bit penyusunnya:

Bit ISC11 dan ISC10 bersama-sama menentukan kodisi yang dapat menyebakan interupsi eksternal pada pin INT1. Dan Bit ISC01 dan ISC00 bersama-sama menentukan kodisi yang dapat menyebakan interupsi eksternal pada pin INT0. keadaan selengkapnya terlihat pada table berikut :

|  |  |  |
| --- | --- | --- |
| ISCx1 | ISCx0 | Pemicu interupsi |
| 0 | 0 | Level rendah pada pin INT0 atau INT1 |
| 0 | 1 | Perubahan level pada pin INT0 atau INT1 |
| 1 | 0 | Transisi turun pada pin INT0 atau INT1 |
| 1 | 1 | Transisi naik pada pin INT0 atau INT1 |

**MCUCSR** ( MCU Control and Status Register)



Bit 6 – ISC2 : interrupt sense control INT2

Untuk interupsi INT2 hanya memiliki satu bit ISC, sehingga hanya memiliki 2 kondisi pemicu interupsi yaitu:

‘0’ = interupsi terjadi jika terjadi transisi turun pada pin INT2

‘1’= interupsi terjadi jika terjadi transisi naik pada pin INT2

Untuk sumber interupsi INT2, perubahan/transisi sinyal yang dapat membangkitkan intrupsi harus memiliki lebar pulsa minimal sekitar 50ns.

**GICR**

Pemilihan pengaktifan interupsi eksternal diatur oleh register GICR ( General Interrupt Control Register ) yang terlihat pada gambar berikut :

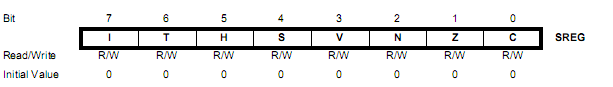


Bit penyusunnya dapat dijelaskan sebagai berikut:

* Bit INT1 adalah bit untuk mengaktifkan interupsi eksternal 1. Apabila bit tersebut diberi logika 1 dan bit I pada SREG (status register) juga satu , maka interupsi eksternal 1 akan aktif.
* Bit INT0 adalah bit untuk mengaktifkan interupsi eksternal 0. Apabila bit tersebut diberi logika 1 dan bit I pada SREG (status register) juga satu , maka interupsi eksternal 0 akan aktif.
* Bit INT2 adalah bit untuk mengaktifkan interupsi eksternal 2. Apabila bit tersebut diberi logika 1 dan bit I pada SREG (status register) juga satu , maka interupsi eksternal 2 akan aktif.

**SREG** (Status Register)

Digunakan untuk menyimpan informasi dan hasil operasi aritmatika terakhir. Data SREG selalu berubah setiap instruksi atau operasi pad ALU dan datanya tidak otomatis tersimpan apabila terjadi instruksi percabangan baik karena interupsi maupun lompatan.

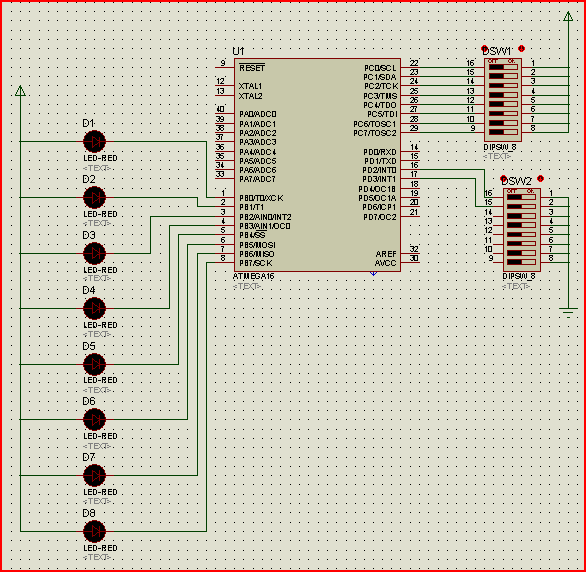


Bit 7 – I : Global interrupt enable

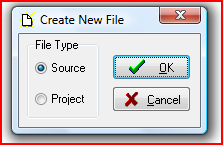
Bit I digunakan untuk mengaktifkan interupsi secara umum (interupsi global). Jika bit I bernilai ‘1’ maka interupsi secara umum aktif, tetapi jika ‘0’ maka tidak satupun interupsi yang aktif.

Langkah Praktikum:

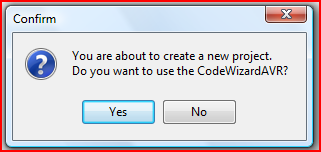
1. Buka gambar modul mikrokontroler ATMega 16 dalam software Proteus ISIS dari folder anda.



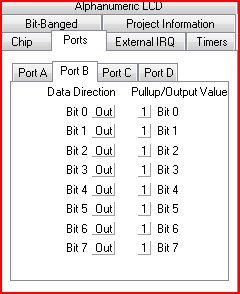
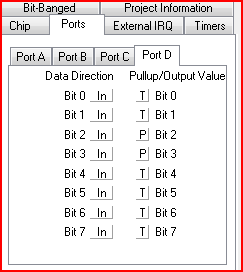
1. Buat Project baru di sofware Codevision dengan langkah2:
   1. Pilih File -🡪 New 🡪 Project

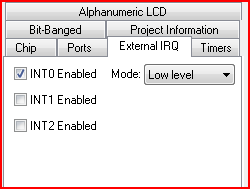
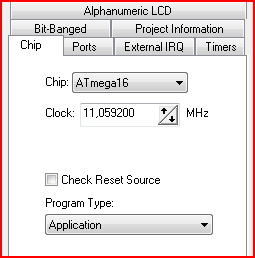


* 1. Ketika muncul jendela yang meminta memilih pengaktifan CodeWizzard pilih YES

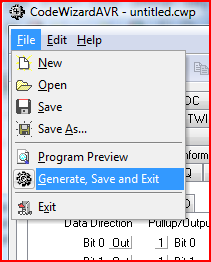


* 1. Sehingga muncul jendela CodeWizardAVR untuk mensetting pemilihan IC mikrokontroler, Nilai Crystal (clock), Setting Port sbg I/O dan pengaktifan interupsi eksternal.

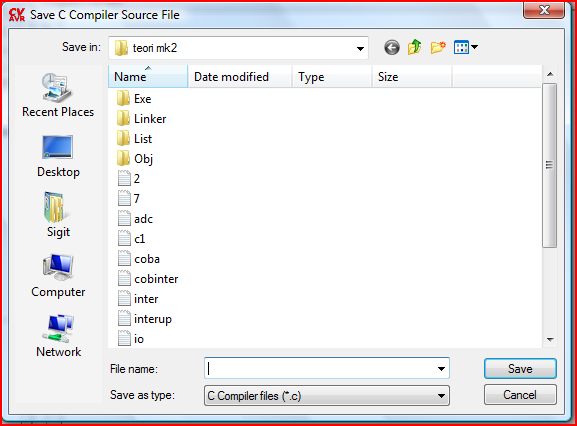
 

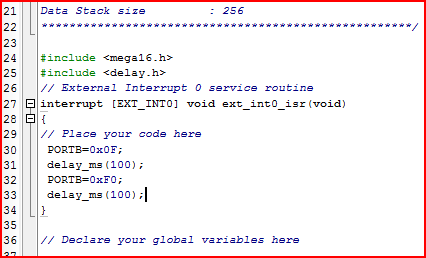
* 1. Masih di jendela CodeWizard, dibagian toolbar pilih File 🡪 Generate, Save and Exit.



* 1. Kemudian beri nama file yang anda inginkan (jangan ada spasi) sebanyak 3 x dg nama file yg sama.



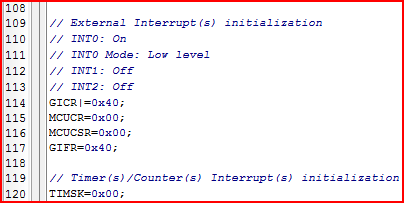
1. Tambahkan perintah berikut



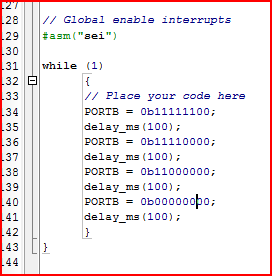
#include <delay.h> untuk mengaktifkan subrutin / header delay.

Dibagian external Interupt 0 service routine adalah program subrutin interupsi INT0 yang kita pilih di codewizard. Perintah tersebut akan menghidupkan led yg tersambung dengan PORTB menyala 4 led bergantian jika kita mengaktifkan interupsi INT0 dengan memberi logika 0 ke PORTD.2 (menggeser saklar DIPSwitch).

1. Cek hasil setting inisialisasi interupsi INT0 dengan codewizard berikut :



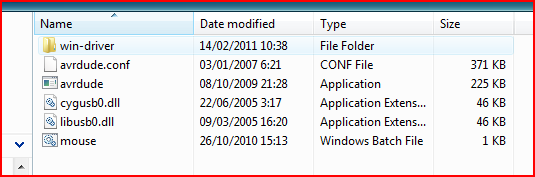
1. Tuliskan perintah di program utama sebagai berikut :



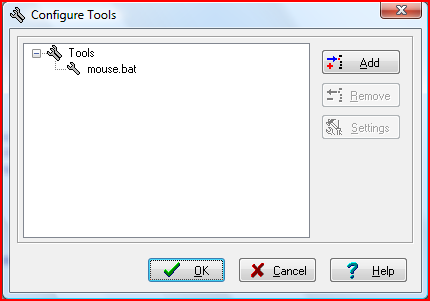
#asm(“sei”) merupakan perintah yg otomatis ditambahkan oleh codewizard ketika kita mengaktifkan interupsi eksternal, jika perintah ini tidak ada maka interupsi eksternal tidak akan aktif. Arti perintah tersebut adalah sei = set interupt enable.

Program utama akan menyalakan led yang tersambung di PORTB dua led demi dua led sampai semua menyala.

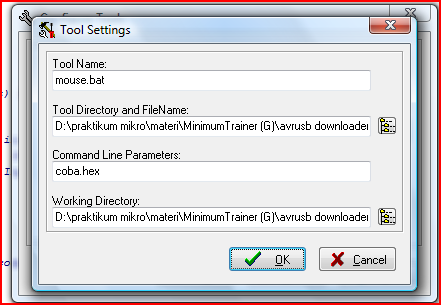
1. Compile dan Build / Make project tersebut.
2. Simulasikan project di Proteus dengan double klik IC ATMega16 dan cari file hex di folder kerja anda dan play simulasi. Hasil eksekusi berupa nyala led 2 demi 2 led sampai menyala semua. Ketika kita menggeser saklar di PORTD.2 yang akan mengaktifkan interupsi INT0 maka akan menyala flip-flop 4 led.
3. Untuk mendownload ke modul mikro lakukan langkah berikut :
   1. Untuk mendownload ke mikrokontroler ikuti langkah berikut : Copy-kan file yg berkstensi hex ke folder tempat menyimpan program AVRdude dan mouse.bat



* 1. Dalam jendela CodeVisionAVR, klik Tools 🡪 Configure sehingga muncul jendela :



* 1. Klik mouse.bat kemudian klik Settings sehingga muncul jendela berikut, pada Command Line Parameters ketikkan nama file hex yg ingin dijalankan. Klik OK



* 1. Kemudian Pada toolbar CodeVisionAVR klik Tools 🡪 mouse.bat . Tunggu sampai file berhasil didownload ke mikrokontroler. Sebelum langkah ini dilakukan pastikan bahwa modul mikrokontroler sudah tersambung ke PC dan driver usb sudah terinstall.
  2. Atau edit mouse.bat melalui software notepad, edit bagian nama file dengan mengganti %1 dengan nama file project anda. Simpan, dan klik dua kali mouse.bat untuk proses download ke modut ATMega16.

