

# Proteus Introduction

April 2011

#2 – kumpulan materi belajar bareng AVR

## LABORATORIUM VIRTUAL KITA

Mungkin kita sudah mengenal PSPICE yaitu sebuah software yang dapat melakukan simulasi rangkaian elektronik. Proteus adalah sebuah software yang memiliki kemampuan seperti PSPICE di atas, namun yang membuatnya istimewa adalah mampu melakukan simulasi terhadap firmware pada embedded system seperti mikrokontroler.

Dengan Proteus terinstal di komputer kita, maka seakan-akan kita telah memiliki sebuah laboratorium elektronik analog maupun digital yang sangat canggih. Memiliki berbagai macam alat ukur, mulai dari oscilloscope, frequency counter, spectrum analyzer, multimeter, dan sebagainya. Memiliki berbagai macam jenis signal generator. Memiliki berbagai macam komponen baik pasif, aktif maupun embedded system (uP) dengan jumlah yang tidak terbatas (tergantung dengan kemampuan komputernya).

Tulisan #2 ini sengaja dibuat untuk memberikan panduan sederhana menjalankan file \*.dsn yang mungkin kita pertukarkan selama belajar AVR ke dalam Proteus yang ada di komputer masing-masing.

## FILE YANG DIPERTUKARKAN VIA EMAIL

Untuk menghemat bandwidth, maka file-file yang akan dipertukarkan adalah \*.bas (firmware), \*.hex (mechine code) dan \*.dsn (file konfigurasi Proteus ISIS). Ketiga file tersebut biasanya besarnya tidak akan lebih dari 400KB, sehingga masih ramah dengan bandwidth masing-masing ... hehehe.

## MENJALANKAN PROTEUS ISIS

Proteus memiliki beberapa modul diantaranya adalah Proteus ARES (untuk mendesain PCB) dan Proteus ISIS (untuk simulasi rangkaian elektronik).

Berikut akan dijelaskan secara sederhana cara membuat sebuah rangkaian berbasis mikrokontroler pada Proteus ISIS dan kemudian menjalankannya.

Karena kita akan melibatkan sebuah mikrokontroler di dalam latihan pengoperasian Proteus ISIS, maka saya sudah menyiapkan firmware sederhana, yaitu mengendalikan LED, 2 detik aktif dan 1 detik padam. Dengan mempergunakan prosedur pembuatan firmware sesuai yang telah dijelaskan pada tulisan #1, copy paste-kan firmware di bawah, kemudian atur environment BASCOM ke ATMEGA-8, kemudian lakukan compiling, maka akan diperoleh file dengan ekstensi \*.hex. File inilah yang akan dilink oleh Proteus ISIS dalam menjalankan simulasi ini.

```

'-----
' 1. Title
'-----
' Project: LED Controller
' Designed by: Cholis Safrudin YD1CHS
' Date: 14th April 2011
'
' Project Description:
' LED (pada pin 14): Nyala 100ms, Mati 50ms
'
' Gear: ATMEGA-8 dengan 8MHz external X'tal
'-----

' 2. Compiler Directives
'-----
$regfile = "m8def.dat"           'memberitahu compiler menggunakan AT-8
$crystal = 8000000              'memberitahu compiler menggunakan 8MHz Xtal
'-----

' 3. Setup Port I/O
'-----
Ddrb.0 = 1                      'setup PORTB.0 (pin 15) sebagai output
Portb.0 = 0                     'set PORTB.0 pada kondisi Low (LED Mati)
'-----


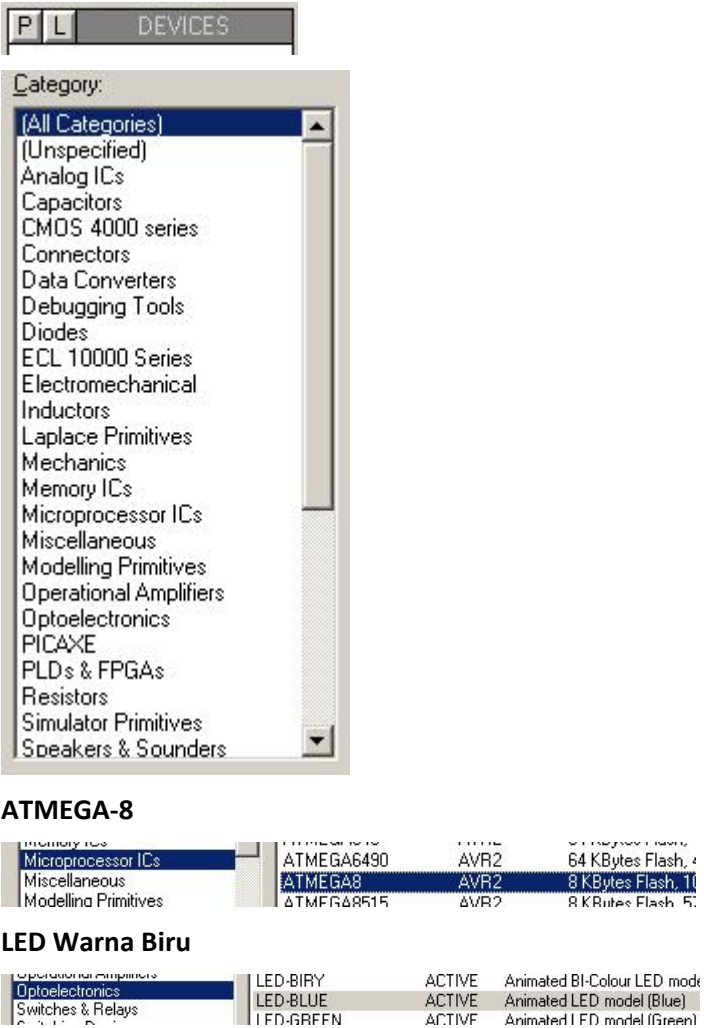
' 4. Program Loop Utama
'-----
Do
  Portb.0 = 1                   'Nyalakan LED
  Waitms 100                   'Tunggu 100 milisecond
  Portb.0 = 0                   'Matikan LED
  Waitms 50                    'Tunggu 50 milisecond
Loop                             'Lakukan hal yang sama seterusnya
'-----

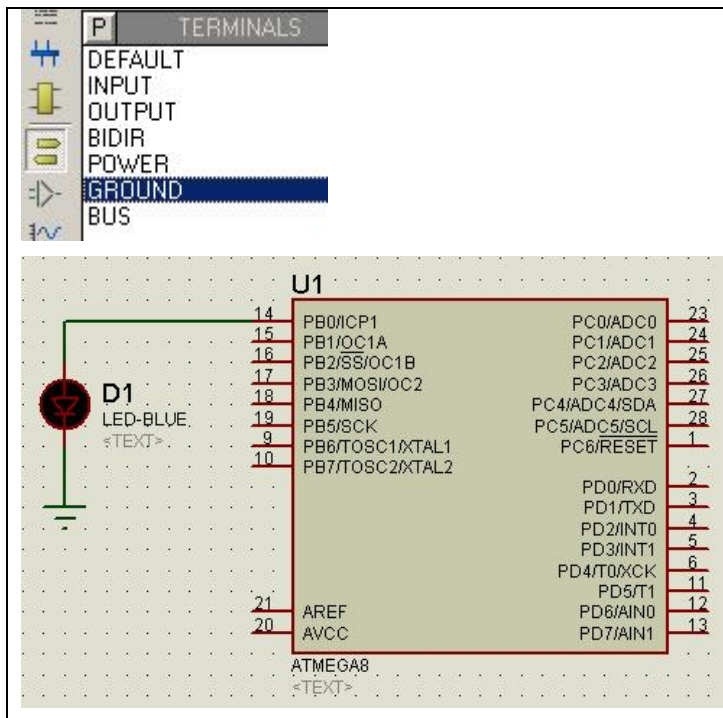
' 5. Akhir dari Program
'-----
End                               'End Program

```

Selanjutnya adalah membuat rangkaian elektroniknya di dalam Proteus ISIS. Untuk memudahkan dalam melakukan debuging nantinya, simpanlah desain Proteus ISIS ini pada direktori dimana \*.hex berada, sehingga bila kita kemudian melakukan editing terhadap \*.bas dan me-recompiling ulang firmware tersebut, maka Proteus ISIS tidak perlu meload ulang \*.hex. Cukup menjalankan tombol “play” maka hasil editing firmware terbaru yang akan dijalankan.

Prosedur pembuatan rangkaian elektronik ke Proteus ISIS, meload \*.hex dan menjalankan simulasi adalah sebagai berikut.



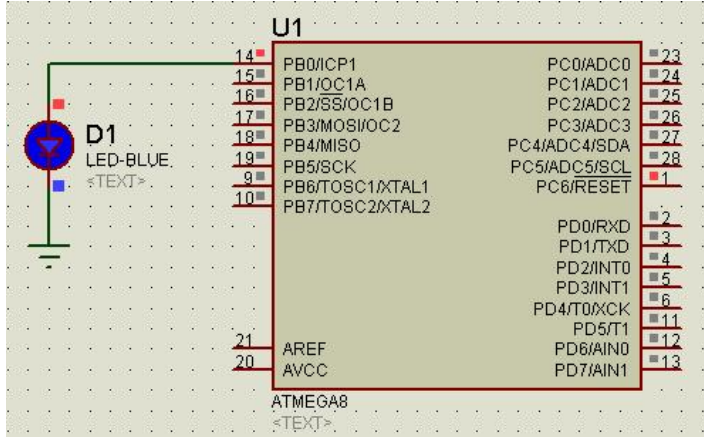
Ilustrasi Grafis	Ilustrasi Deskriptif																								
	<p><b>Berikut adalah Prosedur Membuat Desain Baru</b></p> <ol style="list-style-type: none"> <li>1. Langkah pertama, buka aplikasi Proteus ISIS</li> <li>2. Buat desain baru</li> <li>3. Simpan desain tersebut pada direktori dan nama sesuai dengan keinginan kita</li> </ol>																								
 <p><b>ATMEGA-8</b></p> <table border="1"> <tr> <td>Microprocessor ICs</td> <td>ATMEGA6490</td> <td>AVR2</td> <td>64 KBytes Flash, 4</td> </tr> <tr> <td>Miscellaneous</td> <td>ATMEGA8</td> <td>AVR2</td> <td>8 KBytes Flash, 1</td> </tr> <tr> <td>Modelling Primitives</td> <td>ATMEGA8515</td> <td>AVR2</td> <td>8 KBytes Flash, 5</td> </tr> </table> <p><b>LED Warna Biru</b></p> <table border="1"> <tr> <td>Optoelectronics</td> <td>LED-BIRY</td> <td>ACTIVE</td> <td>Animated BI-Colour LED model</td> </tr> <tr> <td>Switches &amp; Relays</td> <td>LED-BLUE</td> <td>ACTIVE</td> <td>Animated LED model (Blue)</td> </tr> <tr> <td></td> <td>LED-GREEN</td> <td>ACTIVE</td> <td>Animated LED model (Green)</td> </tr> </table> <p><b>Grounding</b></p>	Microprocessor ICs	ATMEGA6490	AVR2	64 KBytes Flash, 4	Miscellaneous	ATMEGA8	AVR2	8 KBytes Flash, 1	Modelling Primitives	ATMEGA8515	AVR2	8 KBytes Flash, 5	Optoelectronics	LED-BIRY	ACTIVE	Animated BI-Colour LED model	Switches & Relays	LED-BLUE	ACTIVE	Animated LED model (Blue)		LED-GREEN	ACTIVE	Animated LED model (Green)	<ol style="list-style-type: none"> <li>4. Untuk memasukan komponen yang akan digunakan di dalam desain, click icon <b>P</b></li> <li>5. Maka akan muncul dialog box DEVICE, di sebelah kiri muncul daftar komponen yang tersedia di library dan bisa digunakan. Tidak semua komponen bisa digunakan untuk simulasi, pilihlah komponen yang memiliki keterangan bisa digunakan untuk simulasi</li> <li>6. Kita membutuhkan 2 buah komponen dan 1 buah terminal ground. Kita tidak perlu menambahkan catuan DC atau memasang Xtal pada uP, karena Proteus secara otomatis mengenalinya dan menganggap telah terpasang</li> </ol>
Microprocessor ICs	ATMEGA6490	AVR2	64 KBytes Flash, 4																						
Miscellaneous	ATMEGA8	AVR2	8 KBytes Flash, 1																						
Modelling Primitives	ATMEGA8515	AVR2	8 KBytes Flash, 5																						
Optoelectronics	LED-BIRY	ACTIVE	Animated BI-Colour LED model																						
Switches & Relays	LED-BLUE	ACTIVE	Animated LED model (Blue)																						
	LED-GREEN	ACTIVE	Animated LED model (Green)																						



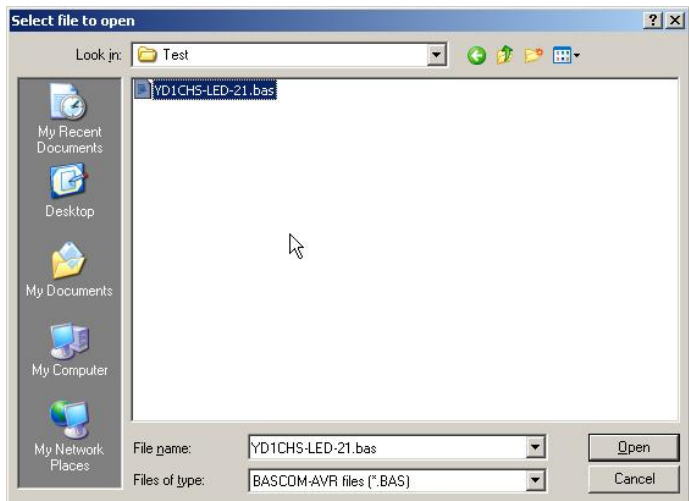
7. Susun seluruh komponen tersebut dan sambungkan dengan konektor sehingga membentuk rangkaian seperti pada gambar di samping
8. Simpan desain yang sudah terbentuk

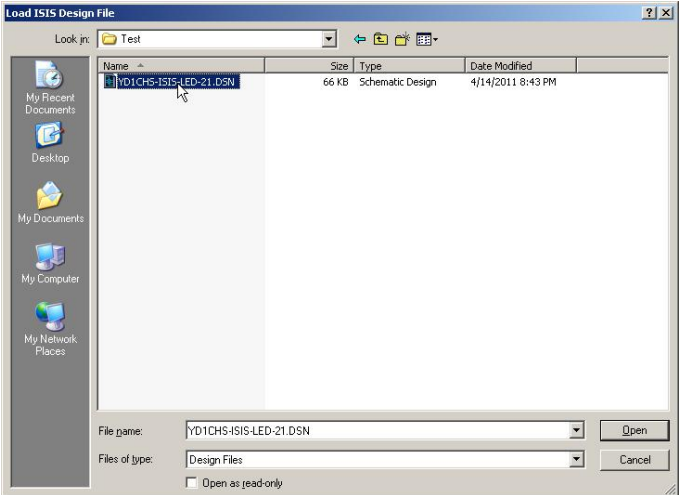
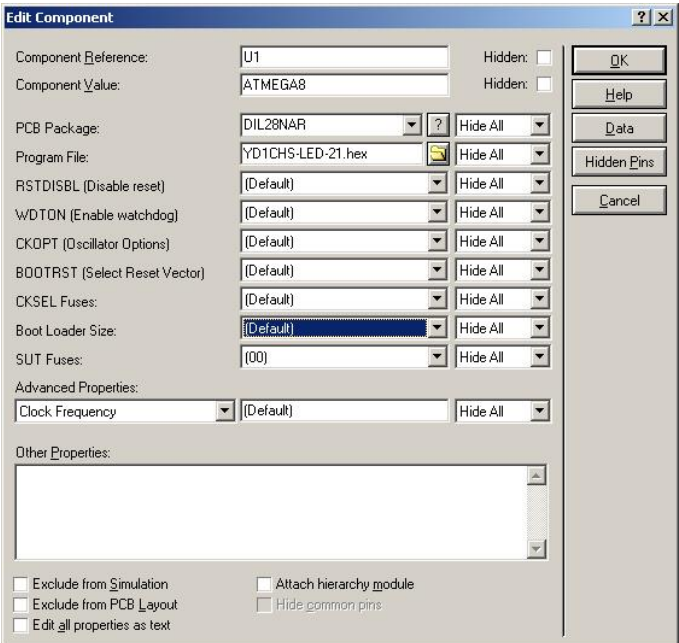
Selanjutnya adalah prosedur untuk meload \*.hex ke dalam desain Proteus ISIS dan menjalankan simulasi.

Ilustrasi Grafis	Ilustrasi Deskriptif
	<p><b>Berikut adalah Prosedur Meload *.hex ke dalam Desain Proteus ISIS dan Menjalankan Simulasi</b></p> <ol style="list-style-type: none"> <li>1. Langkah pertama, pada desain yang telah terdapat komponennya, lakukan double click pada gambar ATMEGA-8 maka akan terbuka dialog box Edit Componen seperti gambar di samping</li> <li>2. Pada pilihan "Program File" lakukan browser kemudian arahkan kepada file *.hex yang telah dibuat tadi. Disini saya memakai contoh file *.hex dengan nama YD1CHS-LED-21.hex</li> <li>3. Pilihan yang lain diset ke "Default"</li> <li>4. Click menu "OK" untuk menyimpan loading ini dan kembali ke window desain utama</li> <li>5. Desain telah siap disimulasikan</li> </ol>

 <p><b>Icon Play/ Simulasi</b></p>  	<ol style="list-style-type: none"> <li>Untuk menjalankan simulasi, pada kiri bawah window Proteus ISIS terdapat jajaran icon seperti gambar disamping</li> <li>Untuk menjalankan simulasi, click icon Play/ Simulasi dan amati bagaimana rangkaian disimulasikan</li> <li>Untuk menghentikan simulasi, tinggal click icon STOP</li> </ol>
---	---

**CARA ME-LOAD FILE \*.HEX DAN \*.DSN HASIL SHARING EMAIL**

Ilustrasi Grafis	Ilustrasi Deskriptif
<p>Misal ada 3 file yang dishare, yaitu:</p> <ul style="list-style-type: none"> <li>- YD1CHS-LED-21.bas</li> <li>- YD1CHS-LED-21.hex</li> <li>- YD1CHS-ISIS-LED-21.DSN</li> </ul>	<ol style="list-style-type: none"> <li>1. Buat sebuah folder di sebarang lokasi dan beri nama folder tersebut, disini saya membuat sebuah folder "test"</li> <li>2. Paste ketiga file di samping ke dalam folder tersebut</li> </ol>
	<ol style="list-style-type: none"> <li>3. Bila anda menginginkan melakukan editing terhadap file *.bas, tinggal buka BASCOM kemudian arahkan ke lokasi file *.bas dimaksud, lalu "open". Ketika anda mengcompile file ini maka *.hex yang baru akan ditimpa selama nama file *.bas sama dengan nama file *.hex</li> </ol>

Ilustrasi Grafis	Ilustrasi Deskriptif
	<p>5. Buka aplikasi Proteus ISIS, lalu buka file *.dsn maka akan muncul gambar desain</p>
	<p>6. Double click gambar mikrokontroler pada desain untuk membuka dialog box “Edit Component”</p> <p>7. Kemudian load *.hex dengan melakukan browsing ke file tersebut</p> <p>8. Click menu OK untuk menyimpan dan menutup dialog “Edit Component” dan kembali ke window desain utama</p> <p>9. Simpan Desain dan kemudian Jalankan simulasi</p>

## PENUTUP

Mudah-mudahan petunjuk sederhana ini dapat membantu mempercepat penguasaan terhadap Proteus ISIS, dan kita bisa segera membahas project-project sederhana, menengah maupun advanced. Oh yaa, bersama posting ini saya sertakan juga 3 file \*.bas, \*.hex dan \*.dsn.

Have a nice day ... de yd1chs