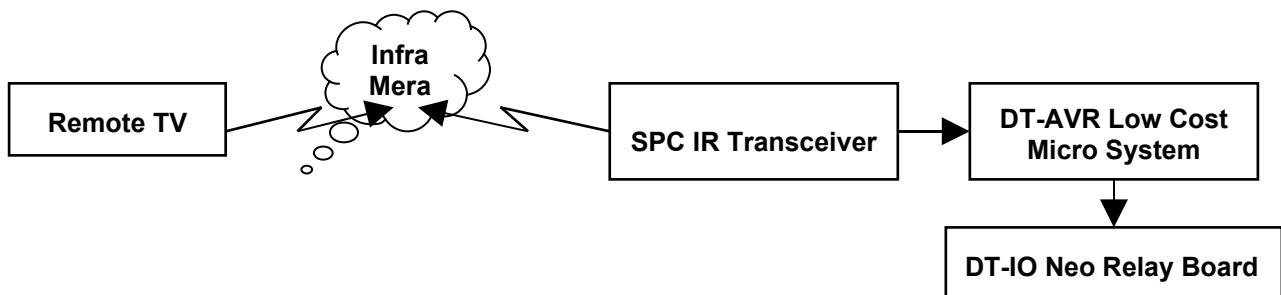


Sekarang ini hampir semua peralatan elektronik rumah tangga ataupun industri telah dilengkapi dengan *remote control* untuk pengoperasian peralatan tersebut. Pada umumnya *remote control* hanya digunakan untuk satu jenis peralatan, misal : *remote TV* hanya dapat digunakan untuk mengatur/mengendalikan TV, *remote AC* hanya dapat digunakan untuk mengatur/mengendalikan AC. Pada aplikasi kali ini akan dibahas perihal penggunaan *remote TV* yang dapat digunakan untuk mengatur/mengendalikan relay. Untuk aplikasi nyata, nantinya relay tersebut dapat dihubungkan dengan lampu ataupun peralatan elektronik yang lain. Sehingga dengan hanya menggunakan *remote TV*, kita dapat mengendalikan TV dan juga dapat mengendalikan lampu ataupun peralatan yang lain. Agar dapat mengendalikan relay, diperlukan proses *login* dengan menekan tombol “*Mute*” selama lebih dari 3 detik sebelum mengendalikan relay tersebut. Dengan fitur ini, nantinya penekanan tombol pada remote TV yang digunakan untuk mengendalikan TV tidak akan mengubah logika relay. Aplikasi ini mampu mengakomodasi *remote TV* Sony, Panasonic dan Philips dengan bantuan modul SPC IR Transceiver. Sedangkan untuk modul Relay digunakan DT-I/O Neo Relay Board dengan bantuan modul mikrokontroler DT-AVR Low Cost Micro System yang akan mengatur logika relay dan menerima data remote TV melalui SPC IR Transceiver. Program di dalam mikrokontroler dikembangkan menggunakan bahasa C dengan bantuan kompilator WINAVR.

Aplikasi kali ini menggunakan modul sebagai berikut :

- SPC IR Transceiver
- DT-AVR Low Cost Micro System
- DT-IO Neo Relay Board

Adapun blok diagram secara keseluruhan adalah sebagai berikut :



Gambar 1
Blok Diagram AN159

Hubungan antara modul-modul tersebut adalah sebagai berikut :

SPC IR Transceiver	DT-AVR Low Cost Micro System
TX (J4 pin 3)	RX (PORTD.0 J13 pin3 / J4 pin2)
VCC (Supply 5VDC)	VCC (J3 5VDC)
GND (Supply GND)	GND (J3 GND)

Tabel 1
Hubungan Antara SCP IR Transceiver Dengan DT-AVR Low Cost Micro System

Perhatikan *jumper* J2, J3 dan J7 pada SPC IR Tansceiver.

- Atur posisi *jumper* J2 dan J3 pada posisi 2-3 untuk antamuka UART TTL
- Atur hubungkan pin 1-2 pada J7 dengan menggunakan *jumper* untuk antamuka UART
- Sesuaikan *jumper* pada pin 3-4 dan 5-6 sesuai dengan protokol yang akan digunakan. Penggunaan protokol harus sama antara konfigurasi program dan konfigurasi pin 3-4 dan 5-6 pada J7.

Perhatikan *jumper* J4 pada DT-AVR Low Cost Micro System.

- Atur posisi *jumper* J4 pada posisi 2-3 jika pin menggunakan J3 pin 3 untuk koneksi RX (Lihat Tabel 1).
- Lepas *jumper* J4 jika menggunakan J4 pin 2 untuk koneksi TX (Lihat Tabel 1).

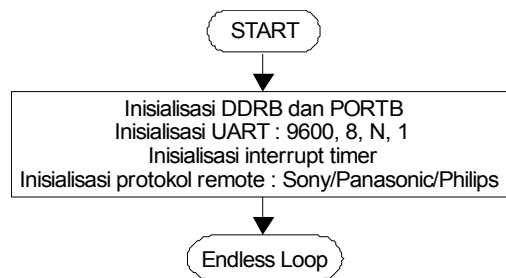
DT-AVR Low Cost Micro System	DT-IO Neo Relay Board
PORTB.0 (J11 pin3)	IN1 (J1 pin3)
PORTB.1 (J11 pin4)	IN2 (J1 pin4)
PORTB.2 (J11 pin5)	IN3 (J1 pin5)
PORTB.3 (J11 pin6)	IN4 (J1 pin6)
PORTB.4 (J11 pin7)	IN5 (J1 pin7)
PORTB.5 (J11 pin8)	IN6 (J1 pin8)
PORTB.6 (J11 pin9)	IN7 (J1 pin9)
PORTB.7 (J11 pin10)	IN8 (J1 pin10)
VCC (J11 pin2)	VCC (J1 pin2)
GND (J11 pin1)	GND (J1 pin1)

Tabel 2
Hubungan antara DT-AVR Low Cost Micro System dengan DT-IO Neo Relay Board

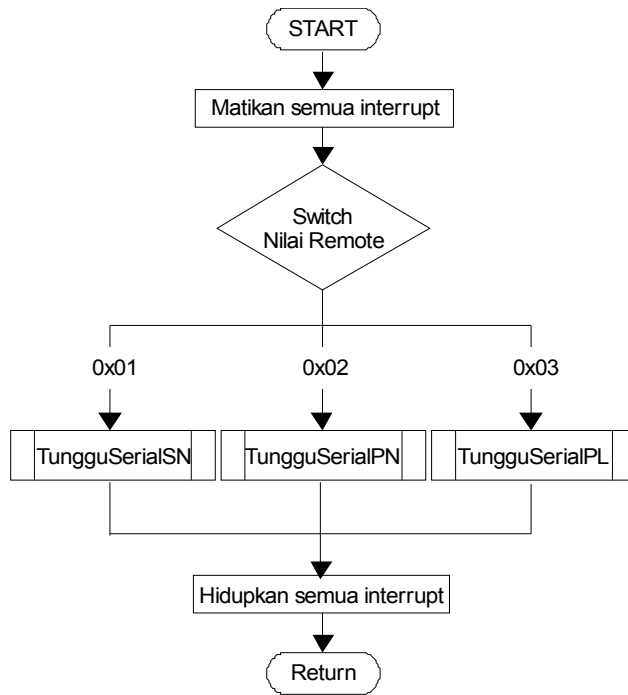
Penggunaan DT-IO Neo Relay Board pada aplikasi ini tidak mutlak dan dapat digantikan dengan modul Relay lain, seperti DT-I/O Relay Board Ver 2.0 atau modul Solid State Relay produksi Innovative Electronics.

Periksa kembali semua hubungan antar modul, jika sudah benar berikan catu daya pada DT-AVR Low Cost Micro System sebesar 9-12VDC dan berikan juga catu daya pada DT-IO Neo Relay Board sesuai dengan spesifikasi catu daya *contact* relay yang digunakan. Jika LED indikator pada DT-AVR Low Cost Micro System telah menyala, programlah "receiver.hex" ke dalam mikrokontroler dengan menggunakan DT-HiQ AVR In System Programmer, DT-HiQ AVR USB ISP ataupun programmer lain yang mendukung fitur AVR ISP. Perhatikan juga pengaturan dari *fusebit* ATmega8535 pada DT-AVR Low Cost Micro System. Aturlah fusebit ATmega8535 dengan konfigurasi osilator internal 4MHz atau eksternal *crystal* 4MHz (jika menggunakan *crystal* 4MHz).

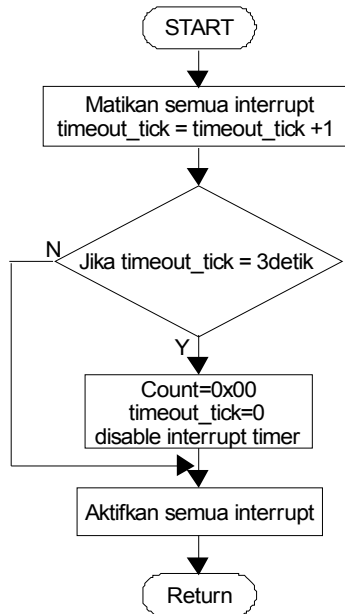
Flowchart program utama receiver.c adalah sebagai berikut :



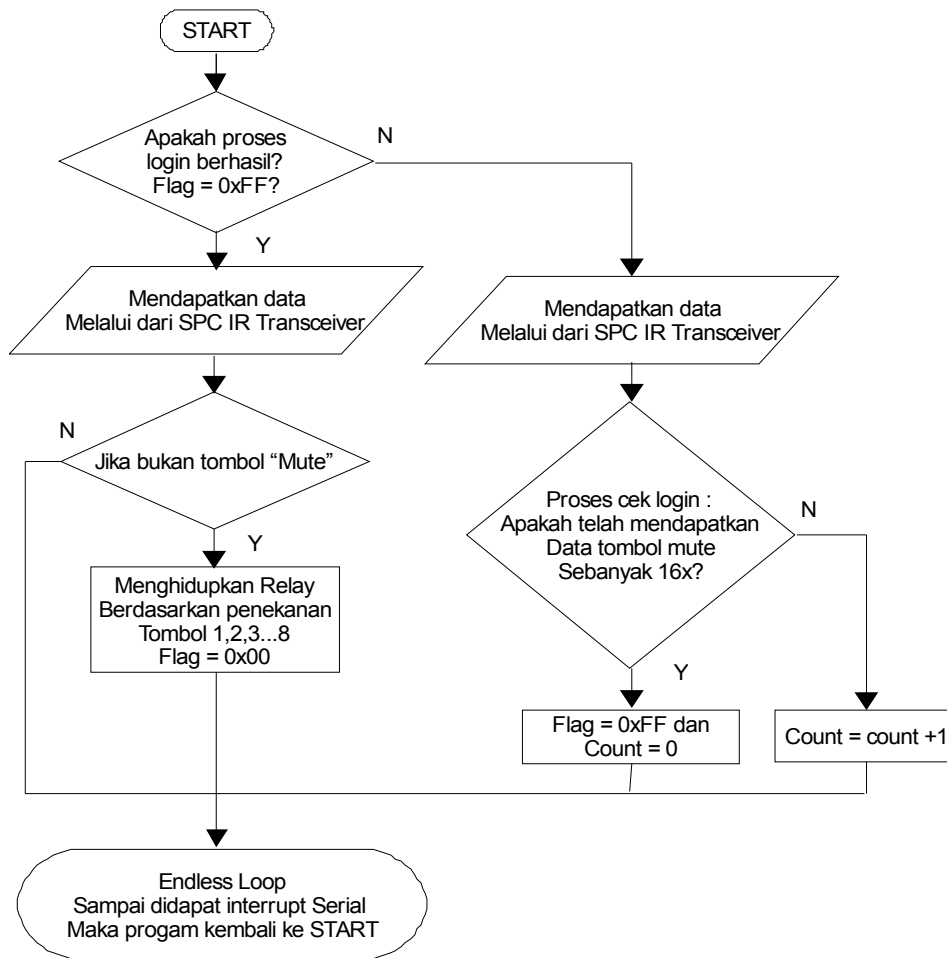
Gambar 2
Blok Diagram Program Utama



Gambar 3
Blok Diagram Rutin *Interrupt* UART



Gambar 4
Blok Diagram Rutin *Interrupt* Timer

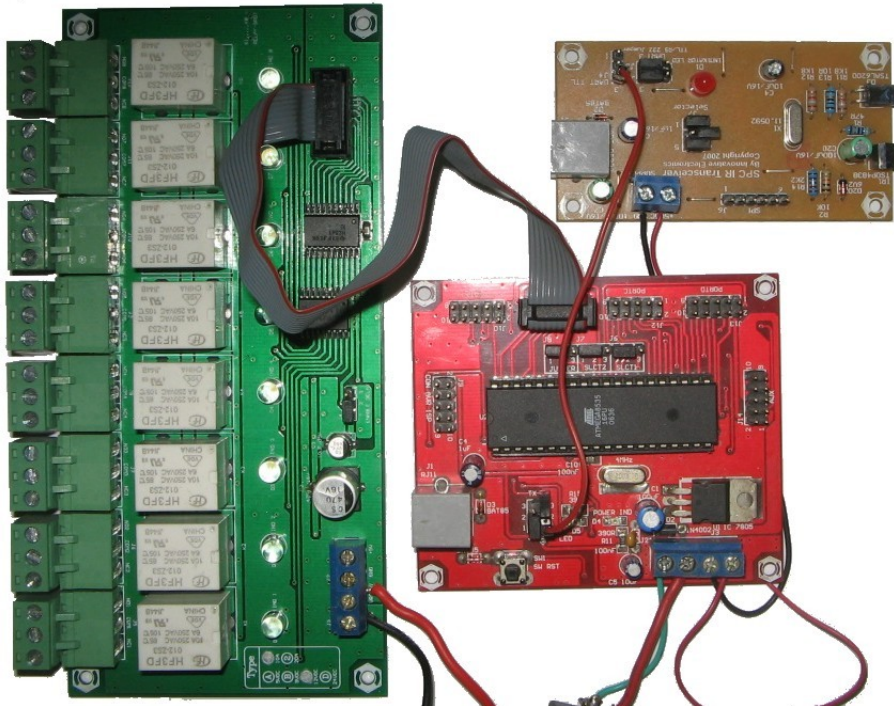


Gambar 5
Blok Diagram Untuk Rutin TungguSerialSN, TungguSerialPN dan TungguSerialPL

Penjelasan singkat program utama dan rutin-rutin yang digunakan adalah sebagai berikut :

1. Langkah pertama adalah melakukan inialisasi DDRB dan PORTB agar dapat digunakan sebagai output.
2. Kemudian inialisasi UART 9600, 8, N, 1, inialisasi *Timer* untuk fungsi *timeout* dan inialisasi protokol *remote* yang akan digunakan.
 - Jika Remote yang digunakan adalah *Remote TV Sony*, maka pada *listing* program harus disesuaikan dengan jenis protokol Sony. **Jangan lupa juga ubah konfigurasi jumper pada SPC IR Transceiver.**
Remote = Sony; //Pilihan Menu Protokol
 - Jika Remote yang digunakan adalah *Remote TV Panasonis*, maka pada *listing* program harus disesuaikan dengan jenis protokol Panasonic. **Jangan lupa juga ubah konfigurasi jumper pada SPC IR Transceiver.**
Remote = Panasonic; //Pilihan Menu Protokol
 - Jika Remote yang digunakan adalah *Remote TV Philips*, maka pada *listing* program harus disesuaikan dengan jenis protokol Philips. **Jangan lupa juga ubah konfigurasi jumper pada SPC IR Transceiver.**
Remote = Philips; //Pilihan Menu Protokol
3. Program akan terus *looping* tanpa mengerjakan tugas apapun apapun.
4. Program akan mengerjakan tugas ketika SPC IR Transceiver mengirimkan data serial UART ke mikrokontroler. Proses ini menggunakan fitur *interrupt* serial yang dimiliki oleh ATmega8535. (Blok diagram lihat gambar 3).
5. Ketika *event interrupt* dijalankan, program akan memeriksa nilai dari "*Remote*". Jika nilai dari "*Remote*" adalah Sony, maka program akan akan menjalankan rutin TungguSerialSN. Jika bernilai Panasonic, maka akan menjalankan TungguSerialPN. Hal tersebut juga berlaku untuk Philips, maka akan menjalankan TungguSerialPL.

6. Secara singkat, urutan logika dari TungguSerialSN, TungguSerialPN dan TungguSerialPL dapat dilihat pada Gambar 5.
7. Pada saat rutin tersebut dijalankan, program akan memeriksa nilai dari variabel flag. Jika bernilai 0xFF, maka proses *login* berhasil. Jika tidak, maka proses *login* belum berhasil. Pada saat dideteksi *login* belum berhasil, maka program akan secara otomatis mengambil data serial yang dikirim oleh SPC IR Transceiver. Kemudian data tersebut diperiksa, apakah *remote* TV dan tombol yang ditekan adalah tombol "Mute". Jika benar, maka program akan mengaktifkan *timer* untuk *timeout*. Setelah itu diperiksa apakah mikrokontroler telah mendapatkan data tombol "mute" sebanyak 16 kali melalui variabel count. Waktu yang diperlukan untuk mendapatkan data tombol "mute" sebanyak 16 kali adalah < 3 detik. Jika belum mendapatkan data tombol "mute" sebanyak 16 kali, maka program akan menambahkan nilai variabel count = count +1. Jika sudah mendapatkan 16 kali data tombol "mute", maka program akan mengubah nilai dari variabel flag menjadi 0xFF. Hal ini berarti proses *login* berhasil.
8. Jika proses login berhasil dan mikrokontroler mendapatkan data dari SPC IR Transceiver, maka data tersebut akan diperiksa, apabila penekanan tombol "mute" maka diabaikan. Jika tombol yang ditekan adalah tombol 1,2,3...8, maka program akan menyalakan relay ke 1, 2, 3...8 sesuai dengan tombol yang ditekan.
9. Fungsi *timer* yang digunakan pada aplikasi ini berguna sebagai fungsi *timeout*. Jika mendapatkan data tombol "mute" pertama untuk proses *login*, maka *timer* akan aktif. *Timer* tersebut akan bekerja selama kurang lebih 3 detik. Jika dalam 3 detik (sejak pertama mendapatkan data tombol "mute") belum mendapatkan 16 kali data tombol "mute", maka *interrupt timer* akan secara otomatis me-reset nilai dari count, yaitu count=0. Hal ini berarti pengguna harus kembali mengulangi proses *login* dengan cara menekan tombol "mute" terus menerus selama lebih dari 3 detik.



Aplikasi program ini terdapat dalam **AN159.ZIP**

Selamat berinovasi!

*All trademarks, company names, product names and trade names are the property of their respective owners.
All softwares are copyright by their respective creators and/or software publishers.*