

Perancangan RS 232 to RS 485 Converter Sistem Network Multidrop

Irwan Sahli, Irwan Kristanto, Tony Chandra Thali

Fakultas Teknologi Industri, Jurusan Teknik Elektro, Universitas Kristen Petra
e-mail: irwans@petra.ac.id, irwankj@petra.ac.id, tonyct@hotmail.com

Abstrak

Komunikasi serial banyak sekali digunakan dalam interface PC namun serial yang biasa dipakai adalah RS-232 yang hanya dapat berhubungan secara one to one. Dalam paper ini dirancang suatu *Converter* dari RS-232 to RS-485 dimana sistem RS-485 dapat berhubungan secara one to many menggunakan sistem multidrop network. Converter menggunakan 3 komponen utama yaitu IC Max 232, IC Max 491 dan IC LM 555. Dalam makalah ini dijelaskan tentang cara kerja, perencanaan sinyal protokol dan pengujian sinyal yang masuk dan keluar dari konverter. Hasil akhir adalah suatu solusi untuk mengatasi keterbatasan komputer dengan terminal RS232 yang hanya mampu membentuk jaringan one to one, menjadi mampu untuk dikembangkan ke sistem jaringan one to many dengan menggunakan kabel RS485 dengan tetap menggunakan terminal RS232 tanpa harus mengganti terminal RS232 menjadi terminal RS485.

Kata kunci : desain dan konstruksi, RS 232 to RS 485 Converter, multidrop network

Abstract

Serial communication is commonly used for PC interfaces. Nevertheless, the commonly used is RS-232, which can communicate one to one only. An achievement was made as a converter from RS 232 to RS 485 has successfully been constructed, whereas the RS 485 system can communicate one to many using network multidrop system. The converter consists of three main components, i.e., IC Max 232, IC Max 491, and IC LM 555. This paper discusses the converter principal operation, the protocol signal planning, and the testing of the signal coming in and out of the converter. As the final result, the converter is indeed a solution to the weak point of computer with terminal RS232 that can only setup one to one networking. The networking can now be extended to one to many network system by using RS485 cabl, but not necessarily replacing the terminal RS232 with terminal RS485

Keywords : design and construction, RS 232 to RS 485 Converter, multidrop network

Pendahuluan

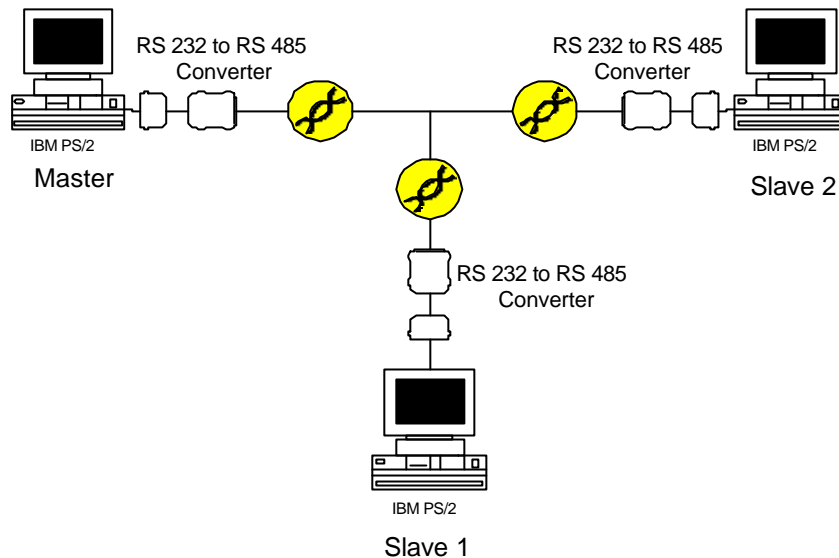
Komputer pada umumnya hanya menyediakan komunikasi secara paralel dan serial, dan komunikasi paralel biasanya digunakan untuk printer sedangkan untuk serial biasanya disediakan dua buah, satu untuk mouse (COM1) dan yang satunya untuk *modem* atau digunakan untuk hubungan antar komputer (COM2).

Karena pada komputer yang menggunakan hubungan serial (RS 232) hanya dapat ber-

hubungan secara *one to one* maka penulis akan menggunakan suatu sistem baru yaitu RS 485 sebagai suatu standar komunikasi serial yang mempunyai kemampuan untuk *multidrop* yaitu sistem dimana sistem ini dapat berhubungan secara *one to many*.

Namun karena keterbatasan dari RS-232 seperti keterbatasan panjang komunikasi sepanjang 50 feet (15 meter) dan hanya dapat berkomunikasi secara *one to one*. Maka dari itu dibutuhkan suatu *converter* dari RS 232 ke RS 485 agar dapat memanfaatkan keunggulan dari sistem komunikasi RS 485.

Catatan : Diskusi untuk makalah ini diterima sebelum tanggal 1 Mei 2001. Diskusi yang layak muat akan diterbitkan pada Jurnal Teknik Elektro volume 1 nomor 2 September 2001



Gambar 1. Topologi Sistem

Perencanaan *software* ini mempunyai tujuan untuk memonitori komputer lain sehingga dengan sebuah komputer dapat melihat layar yang ada pada dua komputer lain.

Perancangan Hardware

Dalam paper ini digunakan 3 buah PC (*Personal Computer*) dengan sebuah PC sebagai *Master* (pengendali) dan 2 buah PC sebagai *slave* (dikendalikan) yang dihubungkan dengan *RS 232 to RS 485 Converter* untuk merubah sistem komunikasi dari RS 232 ke sistem komunikasi RS 485. Topologi sistem yang digunakan adalah topologi *multipoint bus (multidrop)*. Kabel yang digunakan adalah 2 twisted pair, untuk transmitter dan receiver dan juga sebuah kabel untuk ground. Dalam sistem *Four Wire Multidrop Network, bus driver (transmitter)* dari *master* dihubungkan ke seluruh *receiver* dari *slave* dan seluruh *transmitter* dari *slave* akan masuk ke *receiver* dari *master*.

Hubungan dari *master* ke *slave* menggunakan hubungan yang bersilangan (*transmitter* dari *master* masuk ke *receiver slave*) sedangkan hubungan antara *slave* menggunakan hubungan yang lurus atau sejajar (*transmitter* dari *slave* masuk ke *transmitter slave* yang lain).

1. RS 232 to RS 485 Converter

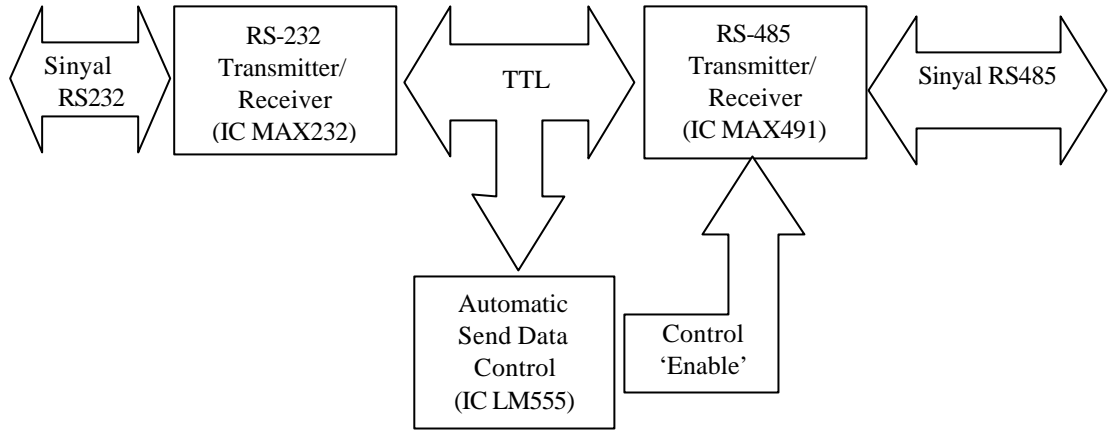
Blok Diagram dari RS 232 TO RS 485 Converter yang di rancang ditunjukkan pada gambar 2. Dari blok diagram tersebut dapat dilihat bahwa sistem terdiri dari tiga komponen utama yakni IC Max 232, IC Max 491 dan IC LM 555.

1.1 IC MAX 232

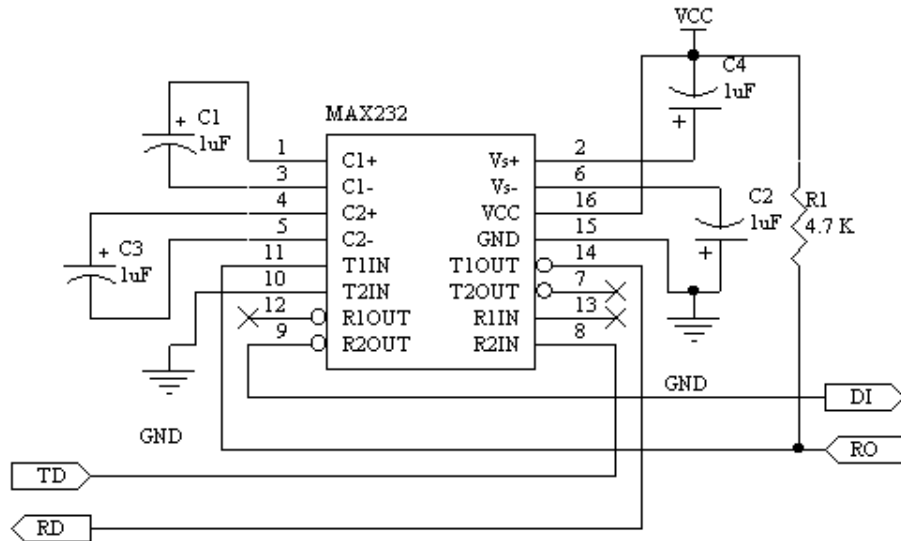
Pada *receiver* dari IC MAX232 diberikan resistor *pull-up*. Resistor ini berfungsi ketika hubungan antara IC MAX232 dan IC MAX491 berada dalam keadaan *transmit*, karena pada saat ini IC LM555 tertrigger dan mengaktifkan *driver enable* dan mematikan *receive enable* pada IC MAX491.

Dengan dimatikannya *receiver enable (RE)*, maka *receiver RS-485* berada dalam kondisi *tri-state*. Hal ini mengakibatkan jalur *receive* mengambang, maka kondisi input yang diterima oleh PC dalam keadaan *idle*-nya, yaitu kondisi logika 1. Sehingga untuk menjaga supaya kondisi logika pada jalur ini tetap 1 (*idle*) digunakan resistor *pull-up* tersebut.

Perancangan RS 232 to RS 485 Converter Sistem Network Multidrop
(Irwan Sahli et al.)



Gambar 2. Blok Diagram RS 232 to RS 485 Converter



Gambar 3. IC MAX232 dalam rangkaian

1.2 IC MAX 491

IC MAX491 ini digunakan untuk merubah tegangan TTL (dari IC MAX232) ke level tegangan pada RS-485 yaitu *differential voltage*. Pada IC ini terdapat pengontrol 'Enable' pada *Transmitter* (DE) yang *active high* dan pada *receiver* (RE) yang *active low*. Sinyal *input* TTL akan masuk ke *Driver Input* (DI) untuk dirubah menjadi *differential voltage*.

Untuk *biasing* digunakan resistor sebesar 4,7 K Ω , resistor sebesar ini digunakan karena cukup memenuhi syarat dalam beberapa kondisi dan juga besaran resistor 4,7 K Ω sering digunakan untuk resistor *pull-up* dan *pull-down*. Apabila

menurut perhitungan sesuai rumus pada sistem yang digunakan maka besarnya resistor sebesar :

$$R_{beban} = (N1/N2/N3)/(T1/T2) \dots \dots \dots (1)$$

$$R_{beban} = (12K\Omega//12K\Omega//12K\Omega)/(120\Omega//120\Omega)$$

$$R_{beban} = 4000\Omega//60\Omega$$

$$R_{beban} = 59\Omega$$

Dimana:
 N1,N2,N3 : Impedansi beban pada tiap node (12K Ω)
 T1,T2 : Termination Resistor (120 Ω)

Untuk menjaga kondisi *tristate* dibawah 200mV (V_{bias}) maka:

$$I_{bias} = V_{bias}/R_{beban} \dots \dots \dots (2)$$

$$I_{bias} = 200mV/59\Omega$$

$$I_{bias} = 3,3 \text{ mA}$$

Untuk membuat arus *bias* (I_{bias}) dari 5V (V_{supply}) maka :

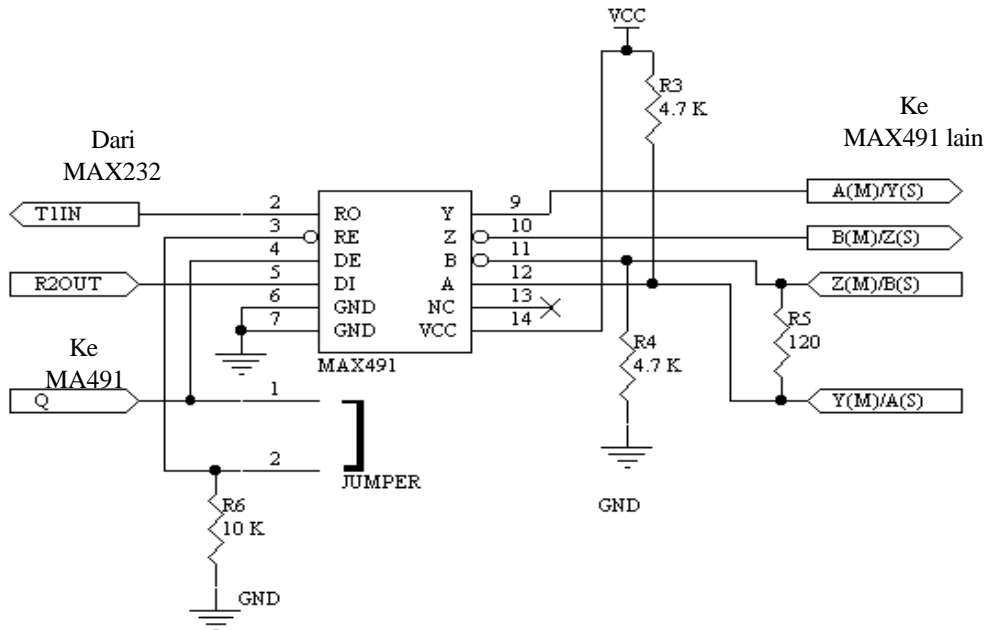
$$R_{biastot} = V_{supply}/I_{bias} \dots \dots \dots (3)$$

$$R_{biastot} = 5V/3,3mA$$

$$R_{biastot} = 1515\Omega$$

Sehingga didapat $R_{bias} = 1515\Omega/2 = 757,5 \Omega \cong 820 \Omega$

Pada Driver Enable dari IC MAX491 ini dihubungkan dari IC LM555 sebagai Automatic Send Data Control.



Gambar 4. IC MAX491 dalam rangkaian

1.3 IC LM555

IC LM555 ini digunakan sebagai *Automatic Send Data Control* yang digunakan untuk mengontrol 'Enable' pada *transmitter* maupun *receiver* dari IC MAX491. Hubungan antara lebar *pulsa* dengan harga resistor R2 dan kapasitor C6 tersebut adalah sebagai berikut :

$$T = 1.1 \times R2 \times C6$$

Dimana T = lamanya waktu *output* berlogic *high*

Sebagai contoh jika rangkaian tersebut didesain untuk 115200 bps (*bit per second*), maka dapat dipakai nilai R2 sebesar 9100 Ω dan nilai C6 sebesar 10 nF. Perhitungan ini dapat dijelaskan sebagai berikut :

$$Bit \text{ rate} = 115200 \text{ bps} \rightarrow \text{waktu untuk 1 pulsa} = 1/115200 = 8,68.10^{-6} \text{ s}$$

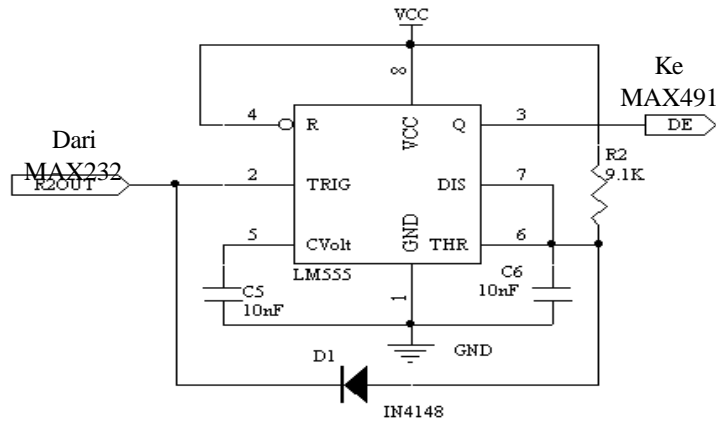
Sedangkan panjang 1 data *asynchronous* dalam perancangan ini adalah 1 *start bit*, 8 *data bit*, 1 *parity bit* dan 1 *stop bit* \rightarrow total 11 bit.

Sehingga waktu untuk mengirimkan 1 karakter adalah : $11 \times 8,68.10^{-6} \text{ s} = 9,55.10^{-5} \text{ s}$

Dengan R=9100 Ω dan C=10 nF, maka diperoleh lebar pulsa :

$$T = 1.1 \times 9100 \times 0.01 \text{ e-6} = 0,0001001 \text{ s}$$

Dioda 1N4148 digunakan untuk membuat IC LM555 menjadi *retriggerable* sehingga waktu pengaktifan IC MAX491 akan diperpanjang. Dari perhitungan dibuat 0,0000046 lebih lama untuk kondisi pada saat semua data bit berlogic '1', karena pengisian kapasitor dimulai pada saat *leading edge*, dan perpanjangan paling lama untuk pengiriman 1 data (11 *bit*).



Gambar 5. IC LM555 Dalam Rangkaian

1.4 Cara Kerja Converter

Pada dasarnya *RS232 to RS485 Converter* ini hanya merubah *level* tegangan saja, apabila ada sinyal yang masuk dari PC ke IC MAX232, maka IC ini akan merubah *level* tegangan yang masuk menjadi *level* tegangan TTL. Pada awal pengiriman akan ada *start bit* yang berlogic low (0 volt), sinyal *start bit* ini dihubungkan ke *Driver Input* (DI) dari IC MAX485 dan juga ke *trigger* dari IC LM555. Pada saat yang hampir bersamaan *start bit* yang dikirimkan ini mentrigger IC LM555 sehingga IC LM555 ini akan menghasilkan *output* berlogic high (5 volt) dan mengaktifkan *Driver Enable* (DE) dari IC MAX491 sehingga IC ini akan meneruskan *bit* yang dikirimkan dari IC MAX232. *Output* dari IC LM555 ini akan berlogic high selama pengisian kapasitor hingga mencapai ambang batas *threshold* dan kemudian akan kembali low.

Dengan pengaturan waktu pengisian kapasitor yang sedikit lebih lama dari waktu yang dibutuhkan untuk menyelesaikan 1 pengiriman (11 *bit*) maka seluruh *bit* yang akan dikirim tidak akan terpotong. Setelah IC MAX491 menerima *bit-bit* dari IC MAX232 maka akan dirubah menjadi perbedaan tegangan antara dua *terminal* Y dan Z (*terminal* pada *transmitter*) sesuai dengan *logic bit* yang diterimanya. Begitu pula pada saat IC MAX491 menerima data dari MAX491 lain, IC ini akan merubah perbedaan antara tegangan *terminal* A dan B (*terminal* pada *receiver*) menjadi tegangan TTL. Karena

receiver enable (RE) dari IC MAX491 dihubungkan ke *ground* maka *receiver* akan selalu aktif dan akan menerima langsung *logic* yang diterima dan diteruskan ke IC MAX232 yang selanjutnya oleh IC MAX232 ini akan dirubah menjadi *level* tegangan RS-232 sesuai dengan *logic* yang diterimanya dan dimasukkan ke dalam *serial* pada PC.

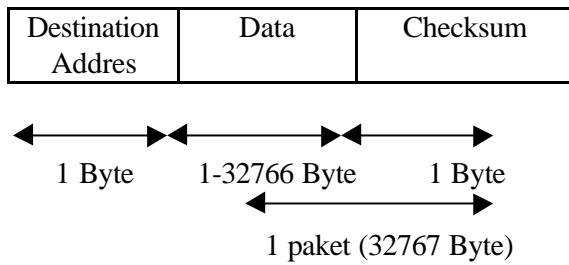
Perancangan Protocol

Perencanaan *software* ini menggunakan bahasa pemrograman Delphi. Komponen yang digunakan dalam membuat program ini adalah TCommPortDriver yang mengatur *serial device* dari komputer.

Software yang digunakan dalam aplikasi ini adalah *software* yang dapat digunakan untuk berkomunikasi dengan 2 komputer dimana komputer *master* dapat memilih layar komputer mana yang akan diambil. Untuk pengambilan gambar dilakukan dengan cara pengambilan seluruh layar dengan perintah *GetWindowDC*.

Maka dari itu harus dibuat suatu *protocol* sehingga masing masing *slave* dapat mengetahui apakah data yang dikirim oleh *master* ditujukan kepadanya atau kepada *slave* lain.

Protocol yang digunakan dalam software adalah protocol dengan susunan sebagai berikut:



Gambar 6. Protocol Software

Destination address adalah alamat dari *slave* yang dituju dalam hal ini untuk *Master* adalah #00, *slave* 1 adalah #01 dan untuk *slave* 2 adalah #02. Khusus untuk destination address, logic 9th bit diset ke '1', dengan demikian dapat diketahui perbedaan antara address dan data. Sehingga apabila didalam data terdapat #00, #01 maupun #02 apabila 9th bit tidak diset '1' maka dianggap sebagai data bukan address. Karena pada *slave* maupun *master* standar penerimaan data akan diset dengan 9th bit pada logic '0' sehingga apabila menerima karakter dengan 9th bit berlogic '1', *slave* akan menganggapnya sebagai error dan setelah itu dilihat data bitnya apakah telah sesuai dengan addressnya.

Data dapat berupa *service* (perintah) maupun data gambar yang dikirim. Untuk pengiriman data file dalam bentuk paket sebesar 32 Kbyte termasuk didalamnya *Checksum*. Sedangkan *Checksum* adalah suatu perhitungan untuk mengetahui apakah data yang diterima telah sama dengan data yang dikirim. Perhitungan *Checksum* ini adalah sebagai berikut:

Setiap data yang akan dikirim akan diambil besaran desimalnya kemudian dijumlahkan dengan besaran desimal data selanjutnya sampai semua data terjumlah, dalam hal ini hasilnya dalam bentuk integer.

Setelah didapat hasil penjumlahan seluruh data, Hasil tersebut akan dibagi dengan 256 dan diambil sisa dari pembagian tersebut. Hasil pembagian tersebut akan berada dalam desimal 0-255 yang kemudian akan dikonversikan lagi kedalam kode ASCIInya dan dikirim pada akhir dari data.

Sedangkan pada penerima juga akan melakukan hal yang sama yaitu menjumlahkan besaran desimal dari semua data yang masuk dan menyisakan data yang terakhir untuk dicocokkan dengan hasil perhitungan *checksum* yang didapat.

Handshaking yang dipakai adalah sebagai berikut:

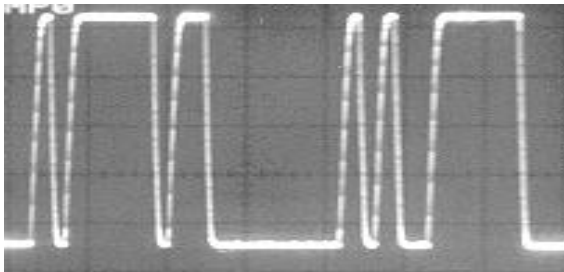
- Untuk pengecekan keadaan, aktif atau tidaknya *slave* digunakan #49 untuk *slave* 1 dan #50 untuk *slave* 2 dengan 9th bit diset '1'. Apabila *slave* menerima karakter ini dengan 9th bit diset '1' maka *slave* akan mengirimkan kembali karakter tersebut dengan 9th bit diset '1'.
- Untuk permintaan besar file dari *slave* digunakan karakter #62. Setelah *slave* menerima addressnya maka *slave* tersebut akan aktif dan akan mengecek penerimaan data. Apabila data yang diterima adalah karakter #62 maka *slave* akan mengcapture layar dan akan mengirim address master #00 dengan 9th bit diset '1' dan data yang berisi besar file ini dikirim pada awal pengiriman file. Besar file ini digunakan untuk mengetahui apakah besar file total yang terima oleh *master* sudah terpenuhi (sesuai dengan besar file yang dikirim).
- Untuk permintaan data dari *slave* digunakan karakter #60. Apabila *slave* menerima addressnya setelah itu akan menerima karakter #60 maka *slave* akan mengirim address master #00 dengan 9th bit diset '1' dan isi dari file ke *master* sebesar 32 KByte, dan counter besar file yang dikirim akan dijumlahkan sebesar 32 Kbyte. Apabila data file yang diterima oleh *master* sesuai dengan data yang dikirim (*checksum* sesuai dan jumlah data sesuai) maka *master* akan meminta lagi data file selanjutnya kepada *slave* dengan perintah #60 lagi setelah mengirimkan address *slave* yang dituju.
- Untuk permintaan data perulangan dari *slave* digunakan karakter #61. Apabila terdapat kesalahan penerimaan data atau besarnya paket yang diterima tidak sama maka *master*

akan meminta *slave* untuk mengirimkan data yang tadi dikirim setelah sebelumnya mengirim *address* dari *slave* yang dituju.

Pengujian

1. Sinyal Input pada IC MAX 232

Sinyal input pada IC MAX 232 diambilkan dari sinyal *output* pada serial PC yang berupa sinyal ASCII dengan urutan sesuai dengan pengiriman secara *asynchronous* yaitu *start bit*, *data bit* (dimulai dari *bit* terkecil), *parity bit* (bila ada), dan *stop bit*.



Gambar 7. Sinyal output dari serial PC (#65 dan #13) dengan parity space

Pada pengujian ini terdapat penambahan sebuah karakter yaitu #13 (Enter), karena pada pengujian ini hanya mengirimkan sebuah karakter yang diakhiri dengan penekanan enter, sehingga hasil keluaran dari PC juga terdapat penambahan karakter #13. *Oscilloscope* diset pada Time/Div 0,2ms dengan perbesaran 5X (40 μ s) dan Volt/Div 5 Volt.

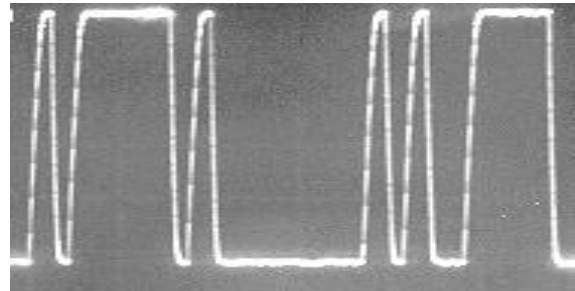
Pada gambar 7 terlihat bahwa sinyal dari serial memiliki *level* tegangan sebesar $4,35 \times 5 = 23,5$ Volt antara *logic* '0' dan *logic* '1', sehingga tegangan antara *logic* '0' terhadap referensi adalah 11,75 volt demikian pula untuk *logic* '1' mempunyai tegangan -11,75 volt. Karakter yang dikirim pada gambar 7 adalah karakter 'A' (#65) dengan *parity space* (selalu '0'). Dari gambar terlihat bahwa sinyal yang didapat berkebalikan dengan kode ASCIInya, tampak disini mempunyai urutan *bit* sebagai berikut :

- Untuk *Disconnect*, *master* akan mengirimkan #100 dengan 9^{th} bit diset '1'. Apabila *slave* menerima karakter ini maka *slave* akan menghapus data yang ada di *buffer*.

0001011111011000000010100111110000

Namun keluaran dari serial PC mempunyai *logic* '1' dengan range -3 sampai -12 V dan *logic* '0' dengan range 3 sampai 12 volt, sehingga ASCIInya merupakan kebalikan dari sinyal yang didapat yaitu :

1110100000100111111101011000001111 dan didapat *start bit* (*bit* 1), *data bit* (*bit* 2-9), *parity*



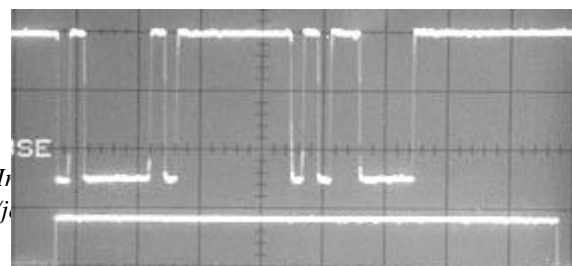
bit (*bit* 10) dan *stop bit* (*bit* 11). Untuk *data bit*, *bit* 2 merupakan *bit* yang paling kecil (*Left Significant Bit*) dan *bit* 9 merupakan *bit* terbesar (*Most Significant Bit*). Sehingga didapat ASCIInya adalah desimal 65 dan 13 dengan 9^{th} *bit* '0', sesuai dengan data yang dikirim.

Gambar 8. Sinyal output dari serial PC (#65 dan #13) dengan parity mark

Pada gambar 8 ini hampir sama dengan gambar 7 hanya *parity bit*nya diset pada *parity mark* (selalu '1'). Dari gambar didapat hasil 1110100000101111111101011000011111. Hasil yang didapat sesuai yaitu ASCII dengan desimal 65 dan 13 dengan 9^{th} *bit* '1'.

2. Sinyal Output dari IC MAX232

Sinyal yang dihasilkan dari IC MAX 232 ini adalah sinyal TTL yaitu sinyal dengan tegangan 5 Volt sebagai *logic* '1' dan sinyal dengan tegangan 0 Volt untuk *logic* '0'. Bentuk sinyal secara keseluruhan hampir sama dengan *output*



dari *serial* PC hanya saja terdapat perbedaan pada *level* tegangan. Bentuk sinyal yang didapat sesuai dengan binari dari data yang dikirim. Bentuk sinyalnya dapat dilihat pada gambar 9 (sinyal atas). *Oscilloscope* diset pada Time/Div 0,2ms dengan perbesaran 5X (40 μ s) dan Volt/Div 2 Volt.

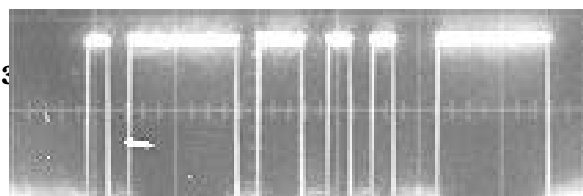
Gambar 9. Sinyal Output IC MAX232 (atas) dan Sinyal Output IC LM555 (bawah)

3. Sinyal Output dari IC LM555

Sinyal *input* dari IC LM555 ini berasal dari *output* dari IC MAX232 dengan *start bit* (0 volt) sebagai *trigger* dari IC LM555. Hasil sinyal *output* dari IC LM555 ini merupakan sinyal *overshoot* dimana sinyal ini akan muncul pada saat terdapat *trigger* dan akan mengeluarkan *output* dengan *logic* '1' (5 volt) selama pelepasan muatan pada kapasitor. Berdasarkan perhitungan maka lama pengaktifan (*output* berlogic '1') adalah 1 ms. IC LM555 pada rangkaian dibuat agar *retriggerable* karena adanya dioda yang terpasang antara kaki 2 dan 6 sehingga lebar pulsa dihitung mulai dari *leading edge* yang terakhir. Tampak pada gambar 9 dari *leading edge* terakhir sampai turunya *output* pada IC LM555 adalah 960 μ s (12x2x40 μ s). *Oscilloscope* diset pada Time/Div 0,2ms dengan perbesaran 5X (40 μ s) dan Volt/Div 5 Volt.

4. Sinyal Output dari IC MAX491

Sinyal *input* dari IC MAX491 adalah sinyal TTL hasil dari *output* IC MAX232. Sedangkan sinyal yang dihasilkan adalah sinyal yang mempunyai bentuk yang sama dengan *input* dari IC MAX491 hanya saja berbeda *level* tegangannya saja. Untuk pengambilan sinyal pada IC MAX491 ini *ground* tidak dihubungkan pada *ground* dari PCB tetapi diambil dari *line A* (*noninverting driver output*) sebagai *output* dari IC MAX491, karena IC MAX491 ini mempunyai *output differensial* yaitu perbedaan tegangan antara *line A* dan *B*. Karena *line A* sebagai *referensi* terhadap *B* maka sinyal akan *high* bila mendapat *input low* demikian pula sebaliknya. Besarnya *level* tegangan sekitar 3,6 volt. *Oscilloscope* diset pada Time/Div 2ms



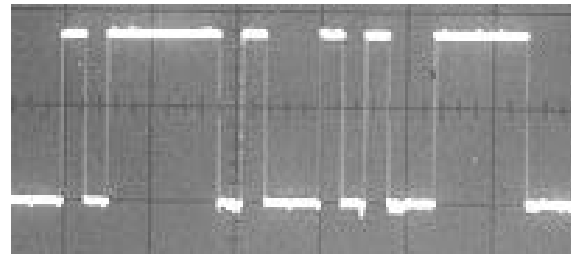
dengan perbesaran 5X (0,4ms) dan Volt/Div 2 Volt.

Gambar 10. Sinyal Output pada IC MAX491 (#65 dan #13) dengan Parity Space

Tampak pada gambar 10 sinyal yang didapat mempunyai bentuk yang sama dengan *output* dari *serial* PC hanya saja sinyal yang didapat lebih *square* demikian pula pada gambar 11 dengan karakter yang sama (A) dengan *parity mark*.

Gambar 11. Sinyal Output pada IC MAX491 (#65 dan #13) dengan Parity Mark

Kesimpulan



- Dengan adanya *RS-232 to RS-485 converter* maka aplikasi serial RS-485 lebih mudah diterapkan pada komputer-komputer biasa yang menggunakan *port* RS-232. Hal ini memberikan kemudahan dan kompatibilitas yang tinggi, sehingga serial RS-485 tergolong praktis dalam pemakaiannya..
- Adanya sistem *four wire multidrop network* memudahkan dalam pembuatan *software* karena tidak diperlukan adanya pengecekan *line* yang digunakan.
- Dengan menggunakan *Automatic Send Data Control* maka tidak diperlukan lagi pengontrolan pengaktifan '*Enable*' melalui *software*.

Daftar Pustaka

- [1]. -----.. "RS-422 and RS-485 Application Note" _____. 1997 B&B Electronics Mfg. Co. Inc.
- [2]. -----.. "Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers". _____ 1996, Maxim Integrated Products.

- [3].-----.. “+5-Powered, Multi-Channel RS-232 Drivers / Receivers.”,_____ 1996, Maxim Integrated Products
- [4]. -----.. “LM555/LM555C Timer Datasheet.” _____, 1997, National Semi-conductor Corp.