

BAB 2

LANDASAN TEORI

2.1 *Analog to Digital Converter*

Salah satu komponen penting dalam sistem akuisisi data adalah pengubah sinyal analog menjadi sinyal digital atau disebut juga ADC (*Analog to Digital Converter*). Pengubah ini mengubah sinyal-sinyal analog menjadi sinyal-sinyal digital sehingga dapat diproses oleh komputer.

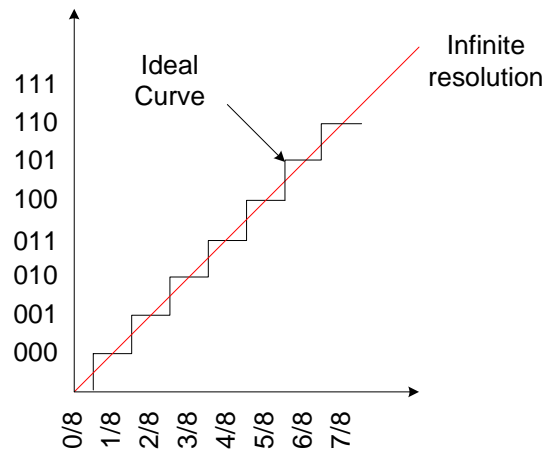
Contoh aplikasi dari ADC ini bisa dilihat misalnya pada *Voltmeter* digital, *sampling* (mengambil contoh dengan interval waktu tertentu) suara dengan komputer sehingga suara dapat disimpan dalam bentuk digital ke dalam media penyimpanan seperti : *disket* dan *compact disk*.

Konsep pengubah sinyal analog menjadi sinyal digital ini adalah *sampling* terhadap sinyal analog yang kemudian mewakilinya dengan bilangan digital dengan batas yang sudah diberikan.

2.1.1 Fungsi Parameter Ideal Konverter Analog ke Digital

Secara teoritis, fungsi transfer ideal untuk konverter analog-ke-digital (ADC, *analog-to-digital converter*) berbentuk garis lurus (Gambar 2.1). Bentuk ideal garis lurus hanya dapat dicapai dengan konverter data beresolusi tak-hingga. Karena tidak mungkin mendapatkan resolusi tak hingga, maka secara praktis fungsi transfer ideal tersebut berbentuk gelombang tangga

seragam. Semakin tinggi resolusi ADC, semakin halus gelombang tangga tersebut.



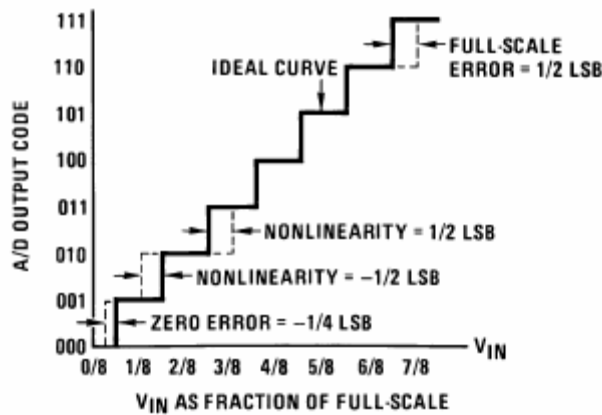
Gambar 2.1 3-Bit A/D Transfer Curve

ADC ideal secara unik dapat merepresentasikan seluruh rentang masukan analog tertentu dengan sejumlah kode keluaran digital. Oleh karena sinyal analog bersifat kontinu sedangkan sinyal digital bersifat diskrit, maka ada proses kuantisasi yang menimbulkan kekeliruan. Apabila jumlah sinyal diskritnya (yang mewakili rentang masukan analog) ditambah, maka lebar undak (*step width*) akan semakin kecil dan fungsi transfer akan mendekati garis lurus ideal.

Lebar satu undak (*step*) didefinisikan sebagai satu LSB (*Least Significant Bit*) dan unit ini digunakan sebagai unit rujukan untuk besaran-besaran lain dalam spesifikasi peranti konversi data. Satu unit LSB itu juga digunakan untuk mengukur resolusi konverter karena dapat menggambarkan jumlah bagian atau unit dalam rentang analog penuh.

Resolusi ADC selalu dinyatakan sebagai jumlah bit dalam sinyal keluaran digitalnya. Misalnya, ADC dengan resolusi n-bit memiliki 2^n kode digital yang mungkin dan berarti juga memiliki 2^n tingkat undak (*step level*). Meskipun demikian, karena undak pertama dan undak terakhir hanya setengah dari lebar penuh, maka rentang skala-penuh (FSR, *full-scale range*) dibagi dalam $(2^n - 1)$ lebar undak. Karenanya,

$$\text{LSB} = \text{FS} / (2^n - 1)$$



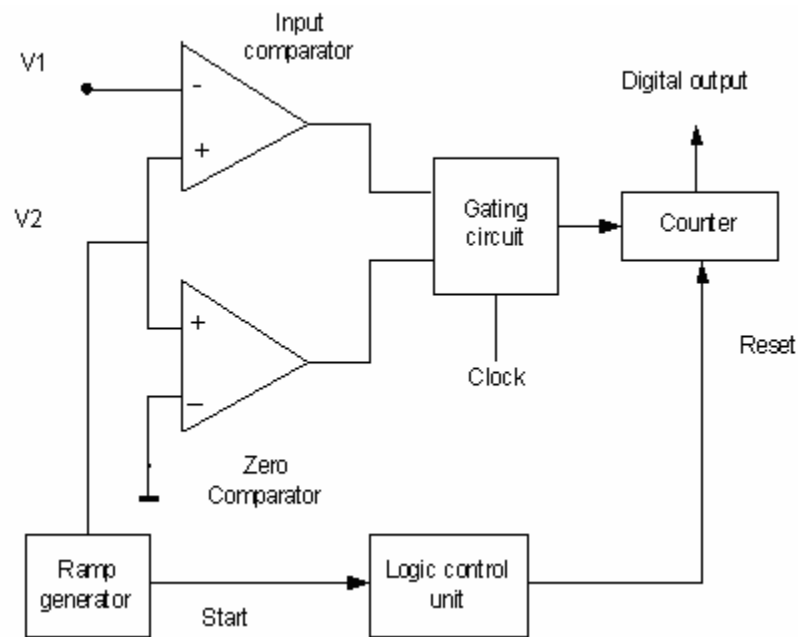
Gambar 2.2 Full Step Mode

2.1.2 Tipe-tipe ADC

Saat ini banyak terdapat ADC dengan kecepatan konversi yang berbeda-beda, interface yang berbeda serta akurasi yang berbeda. Dibawah ini akan dijelaskan tipe-tipe ADC yang sering digunakan yaitu *single slope*, *dual slope*, *Flash*, *Successive Approximation Register* dan *Integrating*.

2.1.1.1 Single slope ADC

Jenis ADC ini keluaran digitalnya dihasilkan dari keluaran tegangan masukan (V_{in}) dengan sinyal *ramp linear*. Cara kerjanya pertama pencacah di *reset* dan sinyal dikirim ke *Ramp Generator*, yang akan bertambah secara linear sesuai dengan waktu dari inisial nilai negatifnya selama tegangan antara lain 0 dan V_{in} . Tegangan keluaran pembanding V_1 dan V_2 akan di-inisialisasi sama dengan nol.



Gambar 2.3 Diagram Blok Single Slope ADC

2.1.1.2 Dual slope ADC

Jenis ADC ini disebut juga *ratiometric* ADC. Cara kerjanya adalah saat keadaan awal S_1 dalam keadaan terbuka, S_2 dihubungkan dan *counter* telah di *reset* rangkaian control. Pada saat $t = t_1$, S_1 dihubungkan dengan masukan *inverting integrator* melalui resistor dan S_2 terbuka, karena V_{in} dan $V_a > 0$, tegangan keluaran *integrator* akan

negatif. Begitu tegangan keluaran ini mencapai *microVolt* dibawah potensial *ground*, keluaran komparator menjadi tinggi. Selama tegangan keluaran dari *integrator* masih negatif, keluaran komparator akan sama dengan satu.

Ketika tegangan melewati nol *Volt*, yaitu pada saat t_3 , keluaran komparator menjadi rendah sehingga logika kembali akan menutup gerbangnya dan saat t_3 pulsa-pulsa clock tidak di teruskan lagi ke pencacah. Penunjukan pencacah pada saat t_3 akan proposional dengan tegangan V_{in} , sehingga mewakili V_{in} .

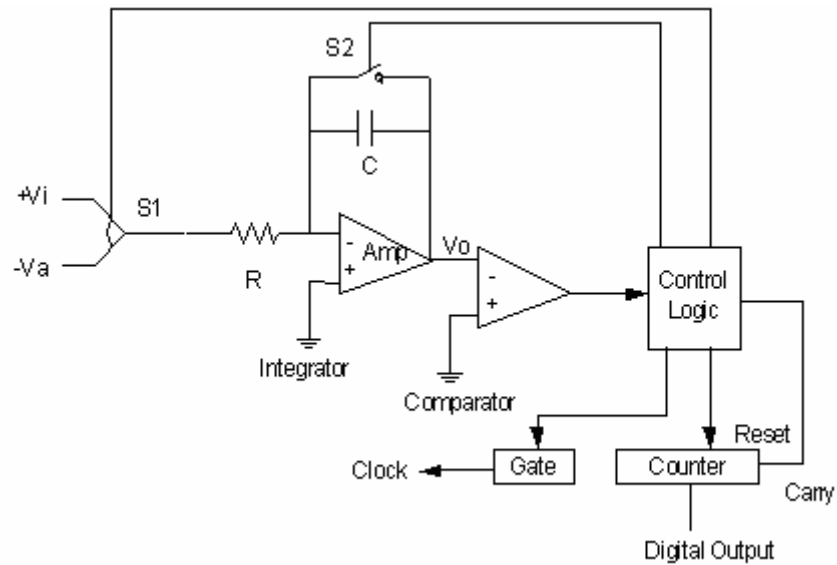
$$V_{in} = (T_f/T_d)V_{ref} = (N_f/N_d)V_{ref}$$

V_{in} : *Charge Time*

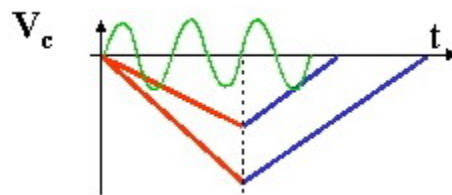
T_1/T_0 : Waktu

N_1/N_0 : Waktu putaran *Clock*

V_{ref} : *Discharge Time*



Gambar 2.4 Diagram blok Dual-Slope ADC

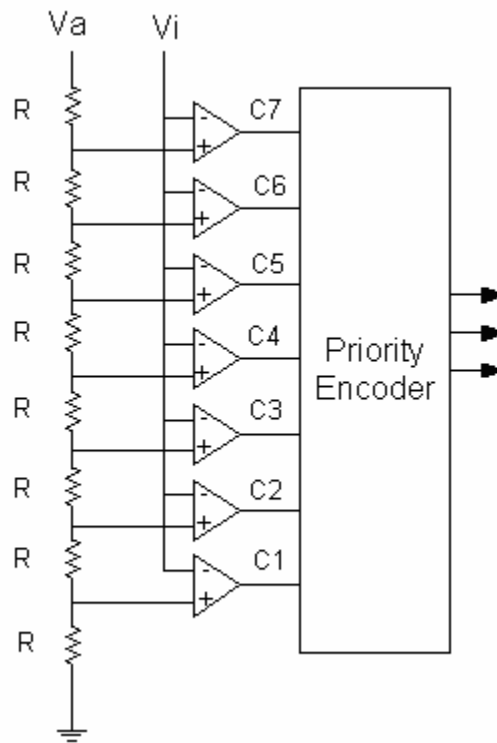


Gambar 2.5 Output Dual-Slope ADC

2.1.1.1 *Flash ADC*

Flash ADC adalah tipe ADC yang memiliki *speed* konversi tercepat. *Flash ADC* menggunakan komparator untuk melakukan konversi sinyal analog ke dalam sinyal digital (4-bit ADC akan memiliki 16 buah komparator dan untuk 8-bit ADC akan memiliki 256 buah komparator). Semua output komparator dihubungkan pada suatu blok logika yang mendeterminasikan output berdasarkan *High/Low*

dari komparator. Kecepatan konversi untuk flash ADC adalah jumlah dari semua waktu yang diperlukan oleh komparator untuk melakukan konversi dan waktu yang diperlukan oleh blok logika untuk mendeterminasikan *output*.



Gambar 2.6 Diagram Blok Flash ADC

2.1.2.4 Successive Approximation Register ADC

SAR (*Successive Approximation Register*) menggunakan komparator dan logika penghitungan untuk melakukan konversi. Cara kerja dari SAR adalah dengan membandingkan sinyal input dengan setengah nilai dari tegangan referensi, dimana apabila keadaan tersebut terbukti kebenarannya maka nilai dari bit MSB (*Most Significant Bit*) adalah set. Kemudian nilai yang dihasilkan akan dikurangi dengan

nilai dari sinyal input, lalu dibandingkan dengan satu per empat dari nilai tegangan referensi. Proses tersebut akan diulangi terus menerus sampai semua bit memiliki nilai set atau reset.

Jenis ini banyak digunakan karena harga yang murah dan waktu konversi SAR jauh lebih pendek dan selalu konstan, tidak bergantung pada nilai sinyal analog yang akan diubah. SAR menggunakan *register control* yang isinya dapat diubah bit demi bit oleh suatu logika kendali. Cara kerja yaitu proses konversi di mulai dengan memberikan pulsa start dan logika kendali akan mereset semua *bit control* sehingga keluaran register semuanya sama dengan nol. Karena $V_0 < V_i$ sehingga keluaran komparator tetap tinggi. Proses diulang sampai semua bit di coba dari MSB ke LSB sehingga proses konversi selesai dan logika kendali akan mengeluarkan sinyal EOC (*End Of Conversion*). Setelah konversi selesai, register control berisi bilangan biner yang ekuivalen dengan nilai analog V_i .

2.1.2.5 Integrating ADC

Cara kerja dari *integrating* ADC atau *dual slope* ADC terletak pada proses charging dan discharging dari kapasitor. Dimana ketika proses *charging* terhadap kapasitor sinyal inputnya akan membawa sinyal noise kedalam sistem ADC. Sedangkan ketika kapasitor melakukan proses *discharging* sistem *counter* akan melakukan penghitungan terhadap output bit ADC. Semakin lama waktu yang

diperlukan untuk proses *discharging* akan meningkatkan output digital ADC. Kelebihan dari *dual slope* ADC adalah memiliki akurasi yang tinggi dan memiliki resolusi yang baik serta menekan frekuensi sinyal *noise* didalam sinyal input. Tetapi dengan menekan sinyal *noise* tersebut akan menurunkan sampel konversi yaitu sekitar 30 sampel per detik. *Dual slope* ADC dapat dioperasikan untuk *sample rate* yang tinggi tetapi semakin meningkat *sample rate* juga akan menurunkan *noise immunity*.

2.2 *Microcontroller*

AT89C52 merupakan salah satu varian dari keluarga besar MCS-51. AT89C52 mempunyai set instruksi dan kemampuan dasar yang kompatibel dengan keluarga MCS-51 lainnya.

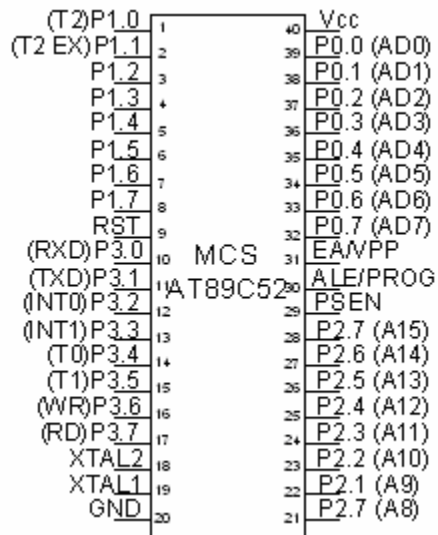
AT89C52 adalah sebuah mikrokontroler yang dikeluarkan oleh Atmel dengan tujuan untuk dengan mudah memadukan arsitektur perangkat keras keluarga mikrokontroler MCS 51, maka AT89C51 terbentuk sebagai sebuah mikrokontroler dengan fasilitas *timer*, port serial 32 kaki I/O, RAM dan *memory* untuk keperluan penyimpanan program.

2.2.1 *Features*

- *Compability with MCS-51.*
- *8KB of In-System Programmable (ISP) Falsh memory.*
- *4.0V to 5.0V operating range*

- Fully static Operation : 0Hz to 33 MHz
- 256 x 8 bit internal RAM.
- 32 programable I/O lines
- Three 16-bit Timer/counters
- 8 interups sources.
- Full Duplex UART Serial Channel
- Low-power idle and power-down modes.
- Interrupt Recovery from power-down mode.
- Watchdog Timer.
- Dual Data Pointer.
- Power off flag.

2.1.1 Konfigurasi pin.



Gambar 2.7 konfigurasi pin AT89C52

Fungsi dari pin AT89C52:

- Port 0 (P0.0 s/d P0.7)

Berfungsi sebagai saluran keluaran alamat rendah (*low byte*) dan saluran keluar/masuk data yang berfungsi secara bergantian. Port 0 terdiri dari 8 buah tingkat penyangga *drain* terbuka yang dapat menarik 8 buah masukkan gerbang bertipe LS TTL.

- Port 1 (P1.0 s/d P1.7)

Port 1 merupakan 8 bit jalur I/O yang mempunyai rangkaian *pull up internal*. Tiap pin pada port 1 dapat mengendalikan 4 masukkan gerbang LS TTL.

- Port 2 (P2.0 s/d P2.7)

Port 2 berfungsi sebagai jalur alamat tertinggi A8....A15, mampu mengendalikan 4 masukkan gerbang LS TTL dan mempunyai rangkaian *pull up internal*.

- Port 3 (P3.0 s/d P3.7)

Port 3 mampu mengendalikan 4 masukkan LS TTL dan mempunyai fungsi khusus untuk mengendalikan *control signal* sebagai berikut :

Tabel 2.1 Fungsi Pin-pin Port 3 MCS

PIN	Fungsi
P3.0	RxD : <i>serial port input</i>
P3.1	TxD : <i>serial port output</i>
P3.2	INT0 : eksternal <i>interrupt</i> 0
P3.3	INT1 : eksternal <i>interrupt</i> 1
P3.4	T0 : timer 0 eksternal input
P3.5	T1 : timer 1 eksternal input
P3.6	WR : eksternal data memori <i>write strobe</i>
P3.7	RD : eksternal data memori <i>read strobe</i>

- EA (*External Access Enable*)
Berfungsi untuk memilih program yang akan diambil dari EPROM eksternal atau EEPROM internal IC. Untuk AT89C52 program berada di EEPROM internal maka pin ini harus diberi logika 1 dengan menghubungkannya pada Vcc.
- RST (reset)
Pin *reset* berfungsi untuk mereset prosesor bila diberi logika “1”. *Reset* diperlukan prosesor untuk inisialisasi internal, seperti mengisi register dengan suatu nilai dan melakukan instruksi JUMP ke alamat awal program yaitu 0000.
- ALE (*Address Latch Enable*)
Berfungsi untuk mengeluarkan pulsa untuk menahan (*latch*) selama pengaksesan memori eksternal. Pulsa ini diberikan sebagai pulsa *strobe* pada *Integrated Circuit latch*.
- Vcc
Tegangan catu daya positif yaitu +5 Volt DC.
- GND (*ground*)
Tegangan referensi 0 Volt (masukan rangkaian bumi)
- PSEN (*Program Store Enable*)
Berfungsi untuk mengeluarkan pulsa dalam proses pengambilan data dari ROM/EPROM eksternal. Ini terjadi pada saat pengambilan instuksi (*fetching*) atau mencari data dalam tabel (*look up table*)

- XTAL 1

Pin masukkan ke rangkaian osilator *internal* dalam *Integrated Circuit*.

- XTAL 2

Pin keluaran dari rangkaian osilator *internal Integrated Circuit* .

2.3 Measurement

Measurement biasanya berhubungan dengan penggunaan suatu instrumen sebagai wadah fisik dalam menentukan sebuah kuantitas atau variabel. Dalam banyak kasus instrumen tersebut memudahkan seseorang untuk dapat menentukan nilai dari suatu kuantitas yang tidak di ketahui dimana hal tersebut tidak dapat di ukur oleh mata manusia. Suatu instrument dapat didefinisikan sebagai suatu alat untuk menentukan nilai atau jarak dari kuantitas atau variabel. Instrumen elektronik didasarkan pada prinsip kelistrikan untuk fungsi pengukuran. Suatu instrumen elektronik merupakan suatu alat dengan konstruksi yang relative lebih sederhana seperti pengukuran arus DC. Dengan perkembangan teknologi bagaimanapun permintaan instrumen yang lebih akurat dan detail meningkat, dan menghasilkan perkembangan baru dalam design dan aplikasi dari suatu instrument. Untuk menggunakan instrument ini secara tepat , seseorang perlu mengerti prinsip pengoperasian dan menafsirkan kecocokan untuk aplikasi yang diinginkan.

2.3.1 Akurasi dan Presisi

Cara kerja mesurement memberikan sejumlah batasan yang didefinisikan sebagai berikut :

- *Instrument* : Suatu alat untuk menentukan nilai atau jarak dari suatu kuantitas atau variabel
- Akurasi : Ketepatan suatu nilai dari suatu variabel yang diukur.
- Presisi : Pengukuran dari hasil yang telah ada sebelumnya.
- Sensitifitas : Rasio perubahan output signal atau respon dari *instrument* untuk merubah input atau variabel yang diukur.
- Resolusi : Perubahan terkecil dalam nilai yang diukur.

Akurasi mengacu pada tingkat kebenaran suatu nilai yang hampir mendekati ketepatan pengukurannya. Sedangkan untuk presisi mengacu pada derajat kebenaran nilai dari suatu kelompok pengukuran atau instrumen.

Sebagai contoh dalam membedakan akurasi dan presisi, dua buah *Voltmeter* dengan buatan dan model yang sama di bandingkan dimana keduanya memiliki kesamaan komponen. Dari kedua alat ini dapat di baca suatu nilai dengan presisi yang sama, jika nilai dari suatu seri *resistance* dalam salah satu *Voltmeter* ini berubah maka akan terjadi kesalahan dalam jumlah yang besar, oleh karena itu keakurasian dari kedua *Voltmeter* tersebut mungkin bisa berbeda. Untuk menentukan suatu kesalahan nilai dari pengukuran tersebut maka perlu dibuat standarisasi pengukuran.

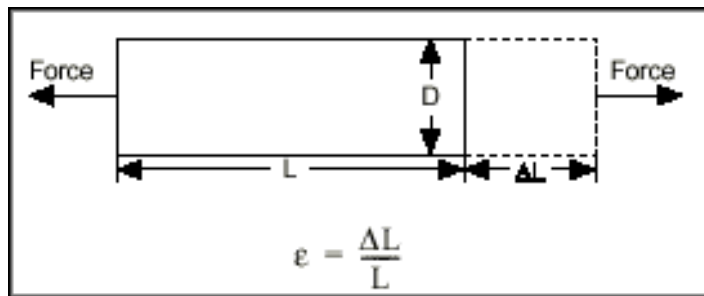
Presisi terdiri dari dua karakteristik yaitu kesamaan dan sejumlah gambaran yang signifikan sehingga dapat dibuat suatu pengukuran. Sebagai contoh sebuah resistor mempunyai tahanan $1.384.572\Omega$ diukur dengan *Ohmmeter* secara konsisten dan dapat digambarkan mempunyai indikasi

sebesar 1.4 MOhm. Tetapi apakah pengamat dapat membaca nilai yang benar dari skala tersebut? Perkiraannya yang terlihat adalah 1.4 MOhm pada skalanya. Hal ini dapat dikatakan benar karena dibaca berdasarkan perkiraan. Meskipun tidak ada deviasi dari nilai yang diamati, kesalahan dapat terjadi karena secara presisi hal tersebut salah. Contoh diatas menggambarkan bahwa kesamaan atau keakuran itu sangat penting. Pengukuran yang baik didasarkan pada hasil keakurasiannya bukan berdasarkan perkiraan.

2.4 Strain Gauge

Strain adalah jumlah banyaknya perubahan yang terjadi akibat tekanan yang diterima oleh suatu media. *Strain* juga dapat diartikan sebagai perubahan panjang suatu media akibat dari suatu tekanan. Nilai dari strain dapat berupa *positive* ataupun *negative*.

Karena *strain* tidak memiliki satuan tersendiri maka diekspresikan dengan in/in atau mm/mm.



Gambar 2.8 Keterangan Strain gauge terhadap tekanan

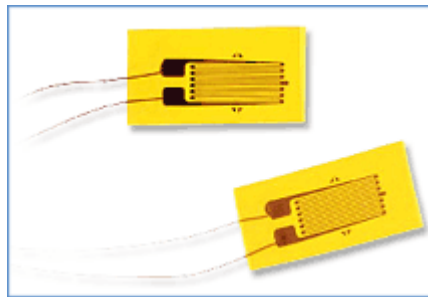
ϵ = Hasil yang di peroleh akibat tegangan atau renggangan.

ΔL = Adalah panjangnya renggangan

L = Panjang *strain gauge*.

2.4.1 Strain gauge

Bonded Foil Strain Gauge pertama kali dikembangkan pada tahun 1938. *Strain gauge* tersebut terbuat dari kabel *filament* dengan ketebalan kurang lebih 0,025mm, yang ditempelkan secara langsung pada permukaan media *epoxy resin*.



Gambar 2.9 *Bonded Foil Strain Gage*

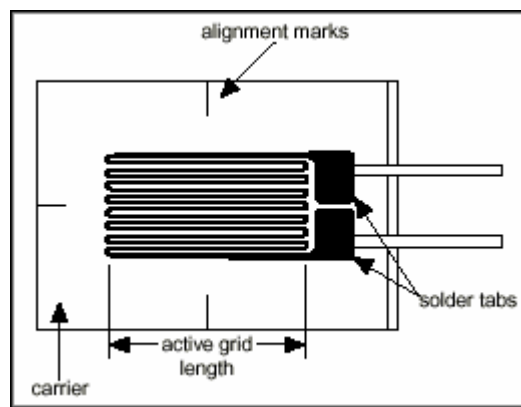
Strain gauge adalah sensor yang mengukur berbagai tekanan yang diterima. *Strain gauge* merubah kekuatan tekanan, ketegangan, berat dan lain-lain, ke dalam bentuk tahanan electric yang dapat diukur.

Ketika terjadi tekanan dari luar pada suatu media tetap, akan menghasilkan tekanan dan ketegangan. Tekanan menunjukkan daya tahan media terhadap suatu tahanan, sedangkan ketegangan menunjukkan perubahan pada media (perenggangan pada media).

Strain gauge adalah salah satu komponen penting dalam teknik pengukuran yang digunakan untuk mengukur kuantiti suatu mekanik. Sesuai dengan namanya, *strain gauge* digunakan untuk mengukur tahanan dan ketegangan yang di bedakan dengan tanda positif atau negatif.

Terdapat banyak cara untuk mengukur dari sebuah *strain*, tetapi yang sering atau umum digunakan adalah *strain gauge*. Dimana *strain gauge* akan mengirimkan hambatan sesuai dengan besarnya tekanan yang diterima oleh *strain gauge*. *Strain gauge* yang sering digunakan adalah *bonded metallic strain gauge*.

Metallic strain gauge terbentuk dari kabel tembaga yang sangat kecil yang tersusun rapi secara *parallel*. Kabel tembaga tersebut ditempelkan pada *carrier* yang digunakan untuk melekatkan *strain gauge* pada suatu media percobaan.



Gambar 2.10 *Bonded Metallic Strain Gauge*

Parameter dasar dari *strain gauge* adalah sensitivitas terhadap *strain*, dimana jumlah *strain* yang diterima diekspresikan dengan *gauge factor*. Sedangkan *gauge factor* itu sendiri adalah perbandingan antara perubahan pada tahanan elektrik yang terjadi terhadap perubahan panjang akibat suatu tekanan.

$$GF = \frac{\frac{\Delta R}{R}}{\frac{\Delta L}{L}} = \frac{\Delta R}{\varepsilon R}$$

GF = Nilai dari Gauge Factor

$\Delta R/R$ = Perubahan daya tahan elektrik

ϵ = Perubahan tekanan atau panjang

2.4.2 Tipe *strain gauge*

A. *Resistance Strain Gauges*

Resistance strain gauge akan aktif bila terdapat perubahan pada konduktor. Ketika media yang digunakan untuk menempelkan strain gauge mengalami perubahan yaitu memanjang dan menyempit akan meningkatkan tahanannya. Perubahan tahanan tersebut di konversikan kedalam tegangan oleh *Wheatstone*. Hasil tegangan yang didapatkan bersifat linier terhadap tekanan yang diterima oleh *strain gauge*.

Resistance strain adalah tipe yang sering digunakan dalam percobaan laboratorium.

B. *Capacitance Strain Gauges*

Capacitance strain gauge dapat digunakan untuk mengukur tekanan.

Dimana persamaan dari *strain gauge*-nya adalah

$$C \propto \frac{ak}{t}$$

Dimana :

C adalah kapasistansi,

a adalah area piringan dari strain gauge,

k adalah konstanta elektrik,

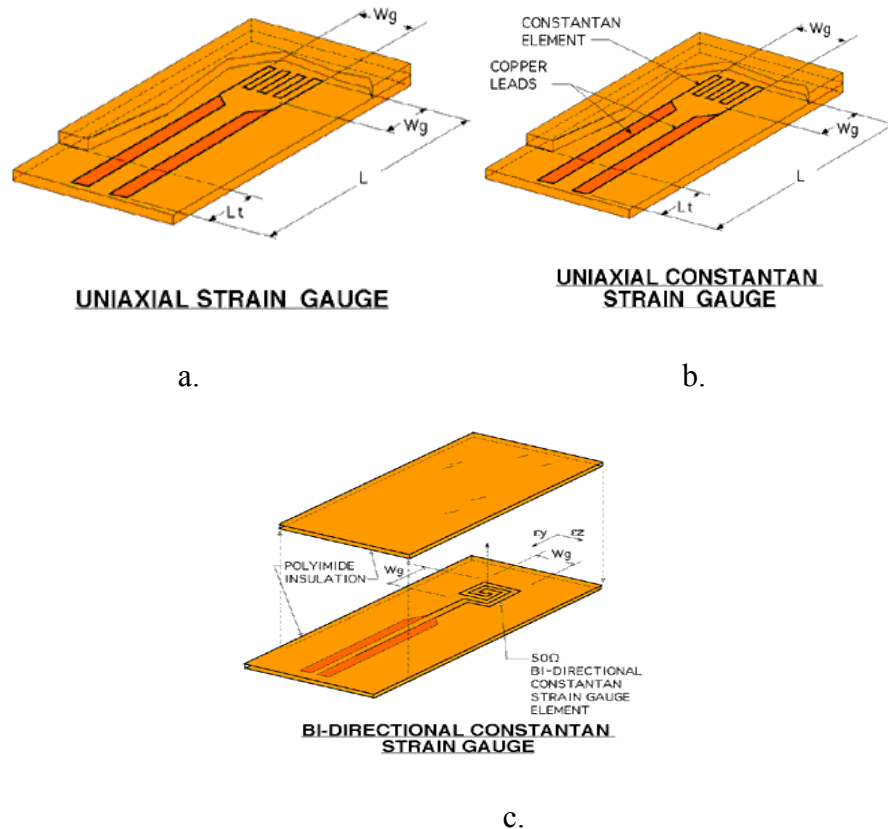
t adalah jarak antara piringan dari strain gauge.

C. Photoelectric Strain Gauges

Photoelectric strain gauge digunakan untuk mendeteksi gerak atau perpindahan tempat suatu benda. Cahaya ditembakkan melalui suatu celah variable yang diaktuasikan oleh *extensometer*, langsung pada *photoelectric cell*. Apabila terjadi perubahan pada lebar celah variable, akan meningkatkan jumlah cahaya yang diterima oleh *photoelectric cell*, sehingga *intensity* arus akan berubah ubah.

D. Semiconductor Strain Gauges

Semiconductor strain gauge akan aktif jika terjadi perubahan tahanan pada media akibat dari ketegangan.

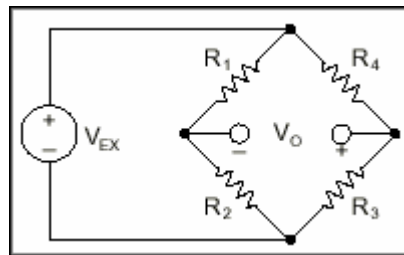


Gambar 2.11 Macam-Macam semikonduktor *Strain Gauge*

2.4.3 Pengukuran *strain gauge*

Tegangan keluaran dari suatu *strain gauge* tidaklah lebih besar dari *millistrain* ($\epsilon \times 10^{-3}$). Tetapi untuk mengukur suatu *strain* diperlukan ketelitian pengukuran terhadap perubahan kecil yang terjadi dalam tahanan.

Untuk dapat melakukan pengukuran terhadap perubahan tahanan yang sangat kecil tersebut *strain gauge* selalu dihubungkan dengan konfigurasi *bridge*. Konfigurasi *bridge* yang umum digunakan pada *strain gauge* adalah *Wheatstone Bridge*.



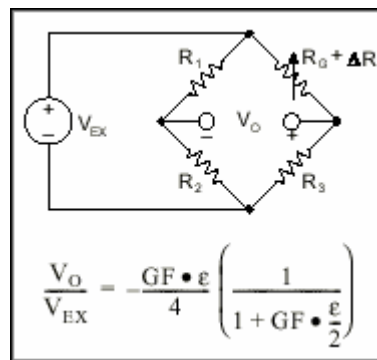
Gambar 2.12 *Wheatstone Bridge*

Tegangan keluaran dari *bridge* adalah V_o , yaitu :

$$V_o = \left[\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right] \cdot V_{EX}$$

Tegangan keluaran V_o akan sama dengan nol apabila nilai $R_1/R_2 = R_4/R_3$. Keadaan dimana tegangan keluaran V_o sama dengan nol adalah keadaan seimbang yaitu pada *strain gauge* tidak terjadi tekanan. Apabila tekanan diberikan pada *strain gauge* maka nilai tegangan keluaran dari V_o akan berubah atau tidak sama dengan nol.

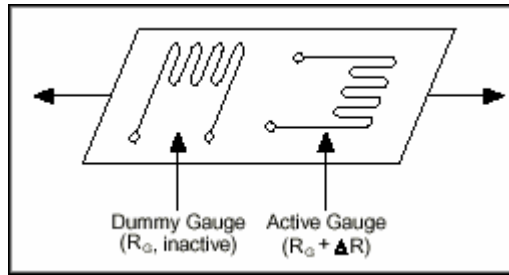
Apabila nilai dari R_4 dirubah pada saat *strain gauge* dalam keadaan aktif, sehingga jika terjadi perubahan hambatan pada *strain gauge* akan menyebabkan ketidakseimbangan pada rangkaian *bridge* dan akan menghasilkan tegangan keluaran yang tidak sama dengan nol. Jika nilai dari hambatan pada *strain gauge* dianggap sebagai R_g , maka nilai tahanan akibat tekanan menjadi $\Delta R = R_G \cdot GF \cdot \tilde{\epsilon}$ Dengan mengasumsikan nilai $R_1 = R_2$ dan nilai $R_3 = R_G$, maka persamaan diatas dapat dituliskan dengan V_o/V_{EX} sebagai fungsi dari *strain* .



Gambar 2.13 *Quarter-Bridge Circuit*

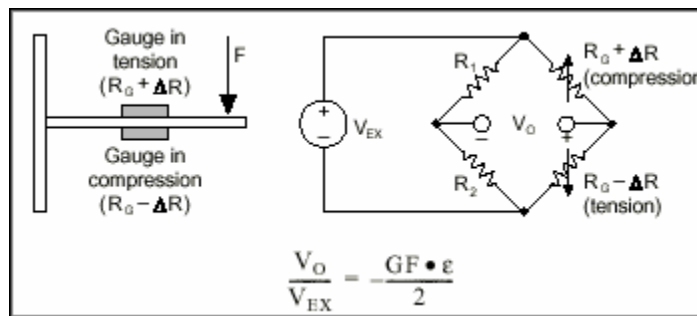
Idealnya *strain gauge* tidak hanya merespon pada tekanan yang diberikan, tetapi juga dapat merespon pada perubahan *temperatur*. Oleh karena hal tersebut perusahaan yang memproduksi *strain gauge* meminimalkan sensitivitas terhadap *temperatur* dengan cara memproduksi *Strain Gauge* dengan menggunakan bahan yang memiliki ketahanan terhadap perubahan suhu yang tinggi.

Dengan menggunakan dua buah *Strain Gauge*, sensitivitas *Strain Gauge* terhadap *temperatur* akan menjadi sangat kecil.

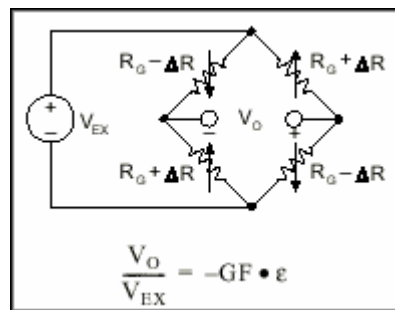


Gambar 2.14 *Active Gauge and Dummy Gauge*

Untuk meningkatkan sensitivitas *strain gauge* dapat dirancang dengan membuat *strain gauge* aktif pada rangkaian konfigurasi *Half-Bridge*. Atau dengan membuat *strain gauge* aktif pada rangkaian konfigurasi *full-bridge*.



a.



b.

Gambar 2.15 a. *Half-Bridge Circuit*, b. *Full-Bridge Circuit*

Persamaan matematika yang didapat dari rangkaian *Wheatstone-bridge* diinisialisasikan sebagai keadaan seimbang dimana tegangan outputnya sama dengan

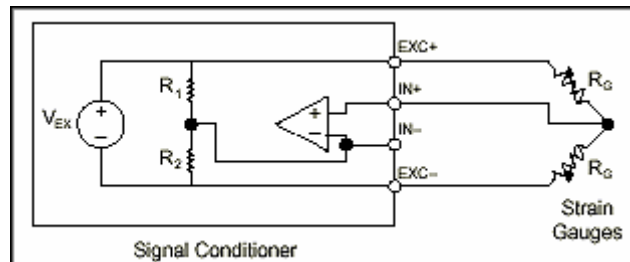
nol, ketika tidak terjadi tekanan pada *strain gauge*. Tetapi didalam kondisi sebenarnya terdapat suatu tegangan offset akibat toleransi resistansi dan *strain* pada *strain gauge*. Tegangan *offset* dapat di setting dengan dua cara yaitu, pertama dengan menggunakan cara *offset-nulling* atau *balancing* rangkaian sehingga tahanan didalam rangkaian *bridge* akan menghasilkan tegangan output nol. Sedangkan alternatifnya adalah dengan mengukur initial output saat tidak terjadi tekanan pada *strain gauge* dan menggunakan software untuk mengatur keadaan saat tidak mengalami tekanan.

2.5 Signal Conditioning for strain gauge

Terdapat beberapa ketentuan untuk menentukan keakuratan dari pengukuran *strain gauge* :

- *Bridge completion*

Bridge Completion – jika menggunakan *full-bridge strain gauge sensor* dengan empat aktif *gauge*, memerlukan tahanan referensi. Sinyal dari *strain gauge* lebih tepat digunakan pada *half-bridge* yang memiliki presisi tahanan referensi yang tinggi.



Gambar 2.16 Connection of Half-Bridge Strain Gauge Circuit

- *Excitation*

Excitation – sinyal *strain gauge* memberikan tegangan konstan untuk mengaktifkan rangkaian *bridge*. Karena tidak ada standar level tegangan, maka level tegangan *excitation* berkisar antara 3 Volt sampai dengan 10 Volt. Sementara tegangan *excitation* yang tinggi akan membangkitkan tegangan output yang tinggi, dan juga dapat mengakibatkan *error* yang sangat tinggi akibat *self-heating*.

- *Remote sensing*

Remote sensing – jika letak dari rangkaian *strain gauge* jauh dari *signal conditioner* dan sumber *excitation*, kemungkinan *error* yang terjadi adalah menurunnya tegangan yang diakibatkan oleh tahanan pada kabel yang menghubungkan *excitation* dengan *bridge*. *Remote sensing* digunakan untuk mengatasi hal tersebut, dimana *remote sensing* dihubungkan diantara *excitation* dan *bridge*.

- *Amplification*

Amplification – keluaran *strain gauge* dan *bridge* relative kecil. Berdasarkan data yang ada sebagian besar *strain gauge* menghasilkan keluaran kurang dari 10 mV/V (10 mV output per tegangan dari *excitation*). Dengan nilai *excitation* sebesar 10 V maka sinyal outputnya akan sebesar 100 mV. Sehingga diperlukan *amplification* untuk meningkatkan level sinyal keluaran *strain gauge*, dengan begitu dapat meningkatkan resolusi pengukuran dan meningkatkan perbandingan antara sinyal dan *noise*.

- *Filtering*

Filtering – *Low Pass filter* dikonjungsikan dengan *strain gauge*, sehingga dapat menekan frekuensi tinggi dari *noise* yang dihasilkan oleh peralatan disekitarnya.

- *Offset*

Offset Nulling – saat merancang *bridge* untuk mendapatkan tegangan keluaran sama dengan nol pada saat tidak terjadi tekanan diperlukan *Offset Nulling*. Dimana terdapat dua cara untuk *offset nulling* yaitu, pertama dengan menggunakan cara *offset-nulling* atau *balancing* rangkaian sehingga tahanan didalam rangkaian *bridge* akan menghasilkan tegangan output nol. Sedangkan alternatifnya adalah dengan mengukur initial output saat tidak terjadi tekanan pada *strain gauge* dan menggunakan software untuk mengatur keadaan saat tidak mengalami tekanan.

- *Shunt calibration*

Shunt Calibration – *Shunt calibration* mensimulasikan input dari tekanan yang diterima dengan cara merubah tahanan pada *bridge*. Sehingga keluaran dari *bridge* dapat diukur dan dibandingkan dengan tegangan yang harapkan. Hasil yang didapatkan digunakan untuk membenarkan nilai *span error* dari seluruh pengukuran.

2.6 Komunikasi Serial

Komputer memiliki *port* paralel dan serial untuk berhubungan dengan peralatan lain di luar komputer. Kedua *port* telah memiliki alamat, nomor interupsi

dan *buffer* yang telah siap dipakai. Port paralel melakukan komunikasi secara paralel yaitu mengirimkan dan menerima data secara paralel, beberapa bit sekaligus. Komunikasi serial dilakukan melalui *port* serial (*communication port*, disingkat COM), menyalurkan data secara berurutan beberapa bit. Standar komunikasi serial yang dipakai adalah standar RS-232C dengan menggunakan IC 8250 pada PC dan IC MAX 232 pada minimum sistem. Kedua IC tersebut berfungsi sebagai pengubah level sinyal TTL ke sinyal RS-232 atau sebaliknya.

2.6.1 RS-232C

RS-232C adalah standar komunikasi serial yang digunakan pada sistem ini. Komunikasi serial menggunakan *port* serial yang lebih rumit pemakaiannya. Komunikasi serial ini dilakukan dengan 3 kawat masing-masing untuk mengirim, menerima dan *ground*. Bila melakukan hubungan jarak jauh, komunikasi serial ini cocok digunakan dibandingkan dengan komunikasi paralel karena menghemat kawat penghubung. Untuk keperluan pengiriman data serial, *interface* RS-232C menggunakan IC MAX 232 karena adanya perbedaan level tegangan sinyal RS-232C dan sinyal TTL pada UART 8250.

Penghubung pada standar RS-232 ada 2 macam yaitu jenis 25 pin (DB-25P) dan jenis 9 pin penghubung DIN (DE-9P). Untuk petunjuk sinyal dan nama-nama pin pada RS-232C (untuk kedua jenis penghubung) dapat dilihat pada table 2.1 berikut ini :

Tabel 2.2 Keterangan Pin-pin serial DB9 & DB25

No.pin untuk 9 pin	No.pin untuk 25 pin	Istilah umum	Istilah RS-232	Deskripsi	Petunjuk sinyal pada DCE
	1		AA	Ground pelindung	-
3	2	TXD	BA	Transmit data	IN
2	3	RXD	BB	Received data	OUT
7	4	RTS	CA	Request to send	IN
8	5	CTS	CB	Clear to send	OUT
6	6	DSR	CC	Data set ready signal ground	OUT
5	7	GND	AB	(Common return)	-
1	8	CD	CF	Received line signal detector	OUT
	9		-	(Menerima pengujian data set)	-
	10		-	(Menerima pengujian data set)	-
	11			Tidak digunakan	-
	12		SCF	Secondary received line signal	OUT
	13		SCB	detector	OUT
	14		SBA	Secondary clear to send	IN
	15		DB	Secondary transmitted data	OUT
				Transmission signal element timing (DCEsource)	
	16		SBB	Secondary received data	OUT
	17		DD	Received signal element timing (DCE	OUT
	18			source)	-
	19		SCA	Tidak digunakan	IN
4	20	DTR	CD	Secondary request to send	IN
				Data terminal ready	
	21		CG	Signal quality detector	OUT
9	22	RI	CE	Ring indicator	OUT

	23		CH/CI	Dta signal rate selector(DTE/DCE	IN/OUT
	24		DA	source)	IN
	25			Transmit signal element timing (DTE	-
				source)	
				Tidak digunakan	

2.6.2 IC UART 8250

IC 8250 adalah UART (*Universal Asynchronous Receiver Transmitter*) digunakan untuk membangkitkan semua sinyal pengendali RS-232C yang menyediakan fungsi-fungsi diantaranya :

- Menyediakan *double buffering*, pencuplikan data masukan serial tidak perlu benar-benar sinkron.
- Keluaran dan masukan dapat menggunakan sinyal *clock* yang berbeda.
- Menyediakan fungsi modulasi dan demodulasi (*modem*). Sinyal ini termasuk CTS, RTS, DSR, DTR, RI dan CD.
- Menyediakan interupsi untuk data masukan dan keluaran.
- Dapat mendeteksi adanya sinyal kesalahan pengiriman data, termasuk *parity, overrun* dan *frame error*
- Dapat mendeteksi adanya *line break signal*.
- Format data yang dikirim dan sinyal perantara pengendali (*control interface signal*) dapat diprogram.

2.6.2.1 Register – register

Terdapat sepuluh register yang berbeda di dalam IC 8250 yang digunakan untuk mewujudkan operasi-operasi diatas. Kesepuluh register terpilih berdasarkan alamat yang diberikan dan nilai bit DLAB (*Divisor Latch Access bit*) dari *Line Control Register* seperti terlihat pada table 2.2 berikut ini :

Tabel 2.3 Register-register IC 8250

Alamat	DLAB	R/W	Register yang dipilih
XF8	0	W	TX data register
XF8	0	R	RX data register
XF8	1	W	LSB of divisor latch
XF9	1	R	MSB of divisor latch
XF9	0	W	Interrupt enable register
XFA	X	R	Interrupt identification register
XFB	X	R/W	Line control register
XFC	X	W	Modem control register
XFD	X	R	Line status register
XFE	X	R	Modem status register
XFF	X	-	Tidak digunakan

Keterangan :

- Nilai x pada alamat register ditentukan oleh COM yang digunakan. Bila COM 1 yang digunakan maka $x = 3$, tetapi jika COM 2 yang digunakan maka $x = 2$
- Nilai x pada bit DLAB artinya nilai bit DLAB bias 0 atau 1 tidak mempengaruhi register yang dipilih.

2.6.2.2 Konfigurasi pin

Terdapat 40 pin pada IC 8250 yang memiliki fungsi-fungsi tertentu.

- D0...D7 (I/O) : Untuk melakukan komunikasi mikroprosesor menggunakan I/O port. Bentuk data yang ditransmisikan termasuk data, *control word* dan *status word*. Keluaran dari data ini adalah *tri-state*.
- RCLK (Receiver Clock, masukan) : Adalah clock pencuplikan data masukan serial. Frekuensinya harus 16 kali *baud rate* data masukan. Secara umum, BOUD OUT dan keluaran data serial menyediakan hal ini.
- SIN (input serial, masukan) : Menerima masukan data serial dari luar.
- SOUT (output serial, keluaran) : Mengeluarkan data serial ke perangkat luar
- CS0, CS1, $\overline{CS2}$ (*chip select*, masukan) : Jika CS0 = '1', CS1 = '1' dan $\overline{CS2}$ = '0', maka 8250 terpilih. Sinyal \overline{ADS} digunakan untuk me-latch sinyal pemilih dan alamat A0...A2.
- *Baud Out* (keluaran) : Mengeluarkan sebuah *clock* dengan frekuensi 16 kali dari output serial baud. Frekuensinya sama dengan rasio dari frekuensi osilasi masukan dibagi dengan *Divisor*

Latch. Jika pin ini dihubungkan dengan pin RCLK, maka *baud rate* dari data masukan dan keluaran sama.

- XTAL1, XTAL2 (masukan) : Digunakan untuk membangkitkan *baud rate*. Menyediakan *baud rate* 9600, 4800, 2400 dan sebagainya yang biasa digunakan pada RS-232 interface. Frekuensi yang lebih baik adalah 1,8432 Mhz, karena frekuensi tersebut dibagi dengan rangkaian pembagi internal, *baud rate* yang lebih teliti dibangkitkan. Frekuensi maksimumnya 3,1 Mhz.
- DOSTR, $\overline{\text{DOSTR}}$ (*Data Output Strobe*, masukan) : Jika 8250 terpilih, data atau *control word* dapat ditulis ke IC tersebut. Jika DOSTR = '1' atau $\overline{\text{DOSTR}} = '0'$, maka DOSTR ($\overline{\text{DOSTR}}$) sama dengan WRITE ($\overline{\text{WRITE}}$).
- DISTR, $\overline{\text{DISTR}}$ (*Data Input Strobe* , masukan) : Data atau *control word* 8250 dapat dibaca jika DISTR = '1' atau $\overline{\text{DISTR}} = '0'$.
- DDIS (*Drive disable*, keluaran) : Jika mikroprosesor membaca 8250, keluaran DDIS = '0' untuk men-drive penerima eksternal, sedangkan selama penulisan, DDIS tidak berpengaruh. DDIS hanya digunakan untuk men-drive data keluaran bus selama pembacaan.
- CSOUT (*Chip Select Out*, masukan) : Jika 8250 terpilih , pin CSOUT = '1'.

- \overline{ADS} (*Address Strobe* , masukan) : Jika $\overline{ADS} = '0'$ maka A0,A1,A2,CS0,CS1 dan $\overline{CS2}$ akan latch. \overline{ADS} juga mengatur apakah 8250 dapat digunakan atau tidak.
- A0, A1, A2 (*Address* , masukan) : Digunakan untuk memilih 10 register internal 8250.
- INTRPT (*Interrupt* , keluaran) : Jika INTRPT = '1' berarti 8250 menginterupsi mikroprosesor. 4 kondisi yang dapat membangkitkan sinyal INTRPT adalah : sinyal kesalahan di *receiver*, *receiver data ready*, register transmit *holding kosong* dan status modem.
- $\overline{OUT2}$ (keluaran) : Status $\overline{OUT2}$ dapat diprogram pemakai. Bit 3 ($\overline{OUT2}$) modem *control register* dapat di-set. Status $\overline{OUT2}$ adalah kebalikan dari bit 3. jika sistem direset, maka sinyal berada pada kondisi high.
- \overline{RTS} (*Request to Send* , keluaran) : Jika 8250 siap untuk menerima data, $\overline{RTS} = '0'$ untuk meminta modem penerimaan data. Level logika ini dapat di-set dengan bit 1 (\overline{RTS}) *modem control register*. Jika $\overline{RTS} = '1'$, maka keluaran $\overline{RTS} = '0'$. Jika sistem direset, maka $\overline{RTS} = '1'$.
- \overline{DTR} (*Data Terminal Ready*) : Ketika 8250 dapat menerima data dari modem, $\overline{DTR} = '0'$, guna memberitahukan modem untuk mengirim data. Level logika dapat di-set dengan bit 0 \overline{DTR} *modem*

control register. Jika $\overline{DTR} = '1'$,maka keluaran $\overline{DTR} = '0'$. Jika sistem direset , maka $\overline{DTR} = '1'$.

- $\overline{OUT1}$ (keluaran) : Status $\overline{OUT1}$ dapat deprogram oleh pemakai. Bila bit 2 *modem control register* di-set maka logika $\overline{OUT1}$ adalah kebalikan dari bit 2. Jika sistem direset, maka $\overline{OUT1} = '1'$.
- MR (*Master Reset*) : Jika MR = '1' maka semua register dan internal control logic di-clear.

Tabel 2.4 efek master reset (MR)

Register/sinyal	Reset control	Reset status
<i>Interrupt enable register</i>	<i>Master reset</i>	Semua bit low (bit 0-3 dipaksa, bit 4-7 selamanya)
<i>Interrupt identifier register</i>	<i>Master reset</i>	Bit 0 = '1', bit 1 & 2 = '0', bit 3 = '0' selamanya
<i>Line control register</i>	<i>Master reset</i>	Semua bit low
<i>Modem control register</i>	<i>Master reset</i>	Semua bit low
<i>Line status register</i>	<i>Master reset</i>	Bit 5 & 6 = '0' yang lain = '1'
<i>Modem status register</i>	<i>Master reset</i>	Bit 0-3 = '0' , bit 4-7 untuk sinyal input
SOUT	<i>Master reset</i>	<i>high</i>
INTR (RCVR error)	<i>Read LSR / MR</i>	<i>low</i>
INTR (RCVR data ready)	<i>Read LBR / MR</i>	<i>low</i>
INTR (RCVR data ready)	<i>Read IIR/Write THR/MR</i>	<i>low</i>
INTR (<i>modem status</i>)	<i>Read MSR/MR</i>	<i>low</i>

<i>change</i>)		
OUT2	<i>Master reset</i>	<i>high</i>
RTS	<i>Master reset</i>	<i>high</i>
DTR	<i>Master reset</i>	<i>high</i>
OUT1	<i>Master reset</i>	<i>high</i>

- $\overline{\text{CTS}}$ (*Clear to send*, masukan) : Setelah 8250 mengeluarkan sinyal $\overline{\text{RTS}}$, modem mengembalikan $\overline{\text{CTS}}$ ke logika '0' jika modem mengizinkan 8250 mengirimkan data. Dengan membaca bit 4 (CTS) modem status register, kita dapat mengetahui kondisi CTS (sejak pembacaan yang terakhir). Jika interupsi modem status di-*enable*-kan, 8250 akan mengeluarkan suatu sinyal interupsi ke sistem selama perubahan kondisi CTS.
- $\overline{\text{DSR}}$ (*Data Set Ready*, masukan) : Jika modem siap untuk mengirim data ke 8250 , modem memberikan logika '0' ke $\overline{\text{DSR}}$ untuk memberitahukan 8250. 8250 akan mengembalikan $\overline{\text{DTR}}$ ke logika '0' jika modem diijinkan. Dengan membaca bit 5 ($\overline{\text{DSR}}$) *modem status register*, kita dapat mengetahui status $\overline{\text{DSR}}$ sejak pembacaan terakhir. Jika interupsi untuk pembacaan logikanya di-*enable*-kan, 8250 akan mengeluarkan suatu interupsi ke CPU selama perubahan level logika $\overline{\text{DSR}}$.
- RLSD (*Receive Line Signal Detect*, masukan) : Jika $\overline{\text{RLSD}} = '0'$, data carier telah terdeteksi. Dengan membaca bit 7 *modem status register*, dapat mengetahui status $\overline{\text{RLSD}}$. Bit 3 *modem status*

register menunjukkan perubahan \overline{RLSD} sejak pembacaan terakhir. Jika perubahan level logika interupsi di-*enable*-kan, 8250 akan mengeluarkan suatu interupsi ke CPU selama perubahan level logika \overline{RLSD} .

- \overline{RI} (*ring indicator*, masukan) : Jika $\overline{RI} = '0'$, modem telah menerima suatu bunyi. Dengan membaca bit 6 modem *status register*, kita dapat mengetahui status \overline{RI} . Bit 2 (*TERI*) *modem status register* menunjukkan perubahan RI sejak pembacaan terakhir. Jika perubahan level logika interupsi di-*enable*-kan, 8250 akan mengeluarkan suatu interupsi ke CPU selama perubahan level logika RI.

2.7 Visual Basic 6.0

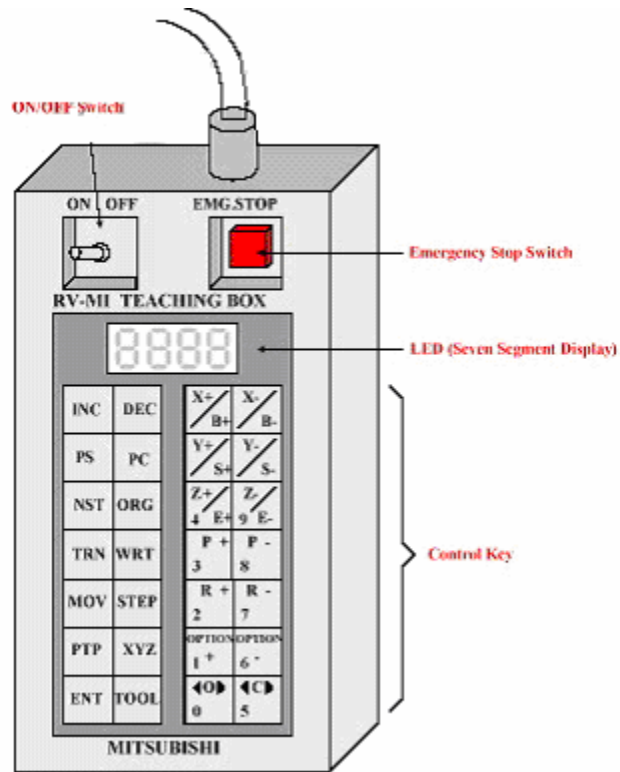
Visual Basic adalah suatu *development tools* untuk membangun aplikasi dalam lingkungan Windows. Dalam pengembangan aplikasi, *Visual Basic* menggunakan pendekatan *Visual* untuk merancang *user interface* dalam bentuk *form*, sedangkan untuk kodingnya menggunakan dialek bahasa *Basic* yang cenderung mudah dipelajari. *Visual Basic* telah menjadi *tools* yang terkenal bagi para pemula maupun *developer*.

Dalam lingkungan *Windows User Interface* sangat memegang peranan penting , karena dalam pemakaian aplikasi yang kita buat, pemakai senantiasa berinteraksi dengan *User interface* tanpa menyadari bahwa di belakangnya berjalan instruksi-instruksi program yang mendukung tampilan dan proses yang dilakukan.

Pada pemrograman *Visual*, pengembangan aplikasi dimulai dengan pembentukan *user interface*, kemudian mengatur *property* dari objek-objek yang digunakan dalam *user interface*, dan baru dilakukan penulisan kode program untuk menangani kejadian-kejadian (*event*). Tahap pengembangan aplikasi demikian dikenal dengan istilah pengembangan aplikasi dengan pendekatan *bottom up*.

2.8 Teaching Box

Alat ini Gambar 2.15 (*Teaching Box*) digunakan pada Robot RV-M1 sebagai alat pengendali dari lengan Robot tersebut. *Teaching Box* selain untuk menggerakkan lengan Robot secara manual, digunakan untuk menyimpan posisi sementara. Setelah itu, baru di-*Load* ke Komputer, dan disimpan dengan ekstensi *.POS. Untuk menggerakkan lengan Robot dari PC, bias digunakan *terminal mode*. Pilihan ini biasa digunakan untuk mentransfer instruksi satu per satu. Berikut konfigurasi tombol dari *Teaching Box* :



Gambar 2.17 *Teaching Box*

- *ON/OFF (Power Switch)* : Untuk mengaktifkan atau non-aktifkan tombol-tombol dalam *teaching box*.
- *EMG.Stop (Emergency stop switch)*: Tombol *push button* yang digunakan untuk menghentikan lengan Robot dalam keadaan darurat.
- INC (+ENT) : Menggerakkan lengan Robot dari posisi lengan Robot yang terakhir (saat ini) ke posisi berikutnya.
- DEC (+ENT) : Menggerakkan Robot dari posisi lengan Robot saat ini ke posisi sebelumnya.
- PS (+ *Number* +ENT) : Menentukan kordinat dari posisi saat ini ke posisi dengan nomor tertentu.
- PC (+ *Number* +ENT) : Menghapus isi dari suatu posisi dengan angka tertentu.

- NST (+ ENT) : Mengembalikan lengan Robot ke posisi *origin*.
- ORG (+ENT) : Menggerakkan lengan Robot ke posisi referensi dalam sistem koordinat kartesian.
- MOV (+ *Number* + ENT) : Menggerakkan lengan Robot ke posisi yang telah ditentukan.
- ENT : Menyelesaikan setiap *entry* tombol untuk menghasilkan operasi yang bersangkutan.
- X+ / B+ : Menggerakkan lengan Robot dalam X-axis positif.
- X- / B- : Menggerakkan lengan Robot dalam X-axis negatif.
- Y+ / S+ : Menggerakkan lengan Robot dalam Y-axis positif.
- Y- / S- : Menggerakkan lengan Robot dalam Y-axis negatif.
- Z+ / E+4 : Menggerakkan lengan Robot dalam Z-axis positif, juga berfungsi sebagai angka *numeric* “4”.
- Z- / E-9 : Menggerakkan lengan Robot dalam Z-axis negatif, juga berfungsi sebagai angka *numeric* “9”.
- P+ / 3 : Merubah posisi pergelangan lengan Robot ke arah naik, juga berfungsi sebagai angka *numeric* “3”.
- P- / 8 : Merubah posisi pergelangan lengan Robot ke arah menurun, juga berfungsi sebagai angka *numeric* “8”.
- R+ / 2 : Memutar pergelangan lengan dalam arah positif (*clockwise*) searah jarum jam, juga berfungsi sebagai angka *numeric* “2”.
- R- / 7 : Memutar pergelangan lengan dalam arah negatif (*counter clockwise*) berlawanan arah jarum jam, juga berfungsi sebagai angka *numeric* “7”.

- Option + / 1 : Menggerakkan axis dalam arah positif, juga berfungsi sebagai angka *numeric* “1”.
- Option- / 6 : Menggerakkan axis dalam arah negatif, juga berfungsi sebagai angka *numeric* “6”.
- <O> / 0 : Membuka genggam tangan, juga berfungsi sebagai angka *numeric* “0”.
- <C> / 5 : Menggenggam, juga berfungsi sebagai angka *numeric* “5”.