

**SISTEM MONITORING DAN PENGENDALI SUHU
TERPUSAT PADA BANYAK INKUBATOR BERBASIS MIKROKONTROLER
DAN PERSONAL COMPUTER (PC)**

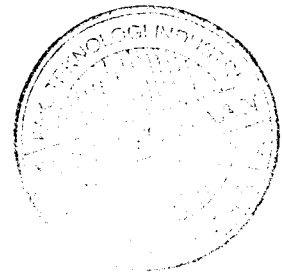
TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat
untuk Melaksanakan Tugas Akhir pada
Jurusan Teknik Elektro



Disusun oleh:

Nama Mhs : Untung Riyono
No. Mhs : 03524085



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2009

LEMBAR PENGESAHAN PEMBIMBING

**SISTEM MONITORING DAN PENGENDALI SUHU TERPUSAT PADA
BANYAK INKUBATOR BERBASIS MIKROKONTROLER
DAN PERSONAL COMPUTER (PC)**

TUGAS AKHIR

Disusun oleh :

Nama : Untung Riyono

No. Mahasiswa : 03 524 085

Yogyakarta, Mei 2009

Pembimbing I



Drs. Abdul Halim

Pembimbing II



Wahyudi Budi P, ST., M.Eng

LEMBAR PENGESAHAN PENGUJI
SISTEM MONITORING DAN PENGENDALI SUHU TERPUSAT PADA
BANYAK INKUBATOR BERBASIS MIKROKONTROLER
DAN PERSONAL COMPUTER (PC)

TUGAS AKHIR

oleh :

Nama : Untung Riyono

No. Mahasiswa : 03 524 085

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, Mei 2009

Tim Penguji,

Drs. Abdul Halim



Ketua

Wahyudi Budi P, ST., M.Eng.

Anggota I

Dwi Ana Ratna Wati, ST., M.Eng.

Anggota II



Mengetahui,

Ketua Jurusan Teknik Elektro

Universitas Islam Indonesia



Gito Yuwono, ST. Msc.

HALAMAN PERSEMBAHAN

Tugas akhir ini Ananda persembahkan Kepada :

Ayahanda dan Ibunda tercinta

Atas segalanya yang telah diberikan kepadaku

*Kakak beserta keluarga besarku yang selalu
memberikan perhatian, semangat, motifasi dan
do'a untukku*

*Dwi Handayani Terima kasih untuk cinta, kasih
sayang, perhatian, dukungan, semangat,
pengorbanan, do'a, dan kesabaranmu yang kau
berikan untukku selama ini.*

MOTTO

“Karena sesungguhnya sesudah kesulitan itu ada kemudahan, sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila kamu telah selesai (urusan dunia), bersungguh-sungguhlah (dalam beribadah), dan hanya kepada Tuhanmulah kamu berharap. ”

(QS. Al-Insyirah : 5-8)

“Sebaik-baik manusia adalah orang yang banyak manfaatnya (kebaikannya) kepada manusia lainnya.”

(H.R. Qadla'ie dari Jabir)

“Allah akan meninggikan orang-orang yang beriman di antaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat.”

(Al-Mujadilah : 11)

“Mulailah dari yang kecil, mulailah dari diri sendiri dan mulailah dari sekarang.”

(AA Gym)

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Wr.Wb.

Alhamdulillah rabbil'alamin, puji syukur senantiasa penulis panjatkan pada Allah SWT atas rahmat dan hidayah-Nya selama menyelesaikan penyusunan tugas akhir ini, Shalawat dan salam pada Nabi Muhammad SAW beserta keluarga dan sahabat-sahabat-Nya.

Tugas akhir dengan judul “ *Sistem Monitoring dan Pengendali Suhu terpusat pada Banyak Inkubator Berbasis Mikrokontroler dan Personal computer (PC)* “ sebagai salah satu syarat untuk memperoleh gelar sarjana Teknik Elektro pada Fakultas Teknologi Industri Universitas Islam Indonesia.

Dalam penyusunan laporan Tugas Akhir ini penulis menyadari bahwa tidak terlepas dari bimbingan, dorongan dan bantuan baik material maupun spiritual dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. ALLAH SWT, yang selalu ada dalam setiap langkah dan dengan izin dan kuasa-Nya selalu memberi kesempatan dan kemudahan untuk dapat menyelesaikan tugas akhir ini.

2. Nabi Muhammad SAW, Nabi akhir zaman serta suri tauladan bagi seluruh umat-Nya.
3. Bapak Tito Yuwono, ST., MSc, selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Bapak Drs. Abdul Halim, selaku Dosen Pembimbing I yang telah memberikan saran-saran, kritik serta bimbingan sehingga penyusun dapat menyelesaikan penyusunan tugas akhir dengan baik.
5. Bapak Wahyudi Budi P, ST., M.Eng, selaku Dosen Pembimbing II yang telah memberikan saran-saran, kritik serta bimbingan sehingga penyusun dapat menyelesaikan penyusunan tugas akhir dengan baik.
6. Seluruh dosen jurusan Teknik Elektro yang telah membagikan ilmunya, karyawan Fakultas Teknologi Industri, kalab Jurusan Teknik Elektro atas tempat, waktu dan ilmu yang telah di berikan.
7. Dwi Handayani, terima kasih atas dukungan, doa dan kesetianmu kepadaku.
8. Mas Irawan, Rahan, Beta, Dodo yang telah berbaik hati meluangkan waktu dan ilmunya.
9. Rekan-rekanku Syarief, Dendy, Adi, Aghel, AdiL, Indrie makasih atas supportnya selama ini.
10. Teman-teman kontrakan Yosh, Irfan, Adhi, Subhan, Bangun, Agung dan semua temen-temen seperjuangan elektro '03 yang telah banyak memberikan bantuan. Serta seluruh rekan-rekan Teknik Elektro Universitas Islam Indonesia yang tidak bisa disebutkan satu-persatu.

Penyusun telah berupaya yang terbaik dalam menyusun tugas akhir ini, namun penyusun menyadari bahwa penulisan laporan ini tidak luput dari kekurangan, maka kritik dan saran yang konstruktif dari semua pihak sangat diperlukan untuk penulisan laporan yang selanjutnya dan penyusun terima dengan sepenuh hati sebagai bahan untuk peningkatan kemampuan dan keterampilan penyusun di lain kesempatan.

Akhirnya penyusun berharap semoga laporan ini dapat bermanfaat dan berguna bagi penyusun dan pembaca serta menjadikan amal ibadah yang diterima di sisi-Nya. Amin...

Wassalamu'alaikum Wr. Wb.

Yogyakarta, Mei 2009

Penulis

ABSTRAK

Inkubator bayi berfungsi menjaga suhu bayi supaya tetap stabil. Bayi prematur pada umumnya perlu diletakkan di inkubator yang mempunyai kontrol suhu, sehingga bayi tetap berada pada suhu yang sesuai seperti saat bayi berada dalam kandungan. Perawatan bayi dalam inkubator yang ada di rumah sakit sekarang masih dilakukan secara manual yaitu membutuhkan campur tangan perawat yang harus mengecek satu per satu suhu inkubator dan menentukan setpoint suhunya sesuai dengan yang diinginkan, hal tersebut dinilai kurang efisien karena suhu inkubator tidak bisa dikendalikan dan dimonitoring secara keseluruhan oleh satu perawat. Padahal perawat juga manusia biasa yang kadang juga bisa lalai, ketiduran, lupa, teledor, yang nantinya dapat membahayakan bayi yang ada dalam inkubator tersebut. Dengan menggunakan mikrokontroler ATmega8535 dirancang suatu sistem yang dapat memonitoring dan mengendalikan suhu pada banyak inkubator yang hasil pengukurannya dapat ditampilkan secara real time pada personal computer (PC). Selain itu alat ini juga dapat mendeteksi keberadaan bayi sehingga sistem bisa ON/OFF secara otomatis. Pada alat ini input suhu bisa disetting antara 32°C-35°C, dengan outputan berupa driver AC untuk heater sebagai pemanas. Sistem pengendaliannya menggunakan metoda ON/OFF sehingga ketika suhu inkubator telah mencapai “setpoint +1°C” heater akan OFF, demikian juga sebaliknya ketika suhu inkubator telah mencapai “setpoint -1°C” heater akan kembali ON. Dari pengujian pengukuran suhu dengan LM35 memiliki error rata-rata sebesar 0,67 %. Penggunaan ADC internal ATmega8535 dapat berfungsi dengan baik terbukti dengan di dapatkan error rata-rata 1,745 %. Pengujian keseluruhan sistem dapat bekerja dengan baik terbukti sistem dapat mengendalikan dan menampilkan suhu baik secara keseluruhan maupun satu per satu dari ketiga inkubator tersebut, meskipun masih terjadi adanya error.

DAFTAR ISI

| | |
|---|------|
| HALAMAN JUDUL | i |
| LEMBAR PENGESAHAN PEMBIMBING | ii |
| LEMBAR PENGESAHAN PENGUJI | iii |
| HALAMAN PERSEMBAHAN..... | iv |
| MOTTO | v |
| KATA PENGANTAR | vi |
| ABSTRAK | ix |
| DAFTAR ISI | x |
| DAFTAR TABEL | xiii |
| DAFTAR GAMBAR | xiv |
| BAB I. PENDAHULUAN | |
| 1.1. Latar Belakang | 1 |
| 1.2. Rumusan Masalah | 2 |
| 1.3. Batasan Masalah | 2 |
| 1.4. Tujuan dan Manfaat | 3 |
| 1.5. Sistematika Penulisan | 3 |
| BAB II. TINJAUAN PUSTAKA | |
| 2.1. Studi Pustaka | 5 |
| 2.2. Mikrokontroler | 6 |
| 2.2.1. Arsitektur Mikrokontroler ATmega8535 | 7 |
| 2.2.2. Fitur ATmega8535 | 8 |

| | |
|--|----|
| 2.2.3. Konfigurasi Pin Atmega8535 | 9 |
| 2.2.4. ADC Internal ATmega8535 | 10 |
| 2.2.5. USART | 12 |
| 2.3. Antar Muka RS-232..... | 13 |
| 2.3.1. Standar RS-232 | 14 |
| 2.3.2. Pengkonversi Level Tegangan TTL ke Level Tegangan RS-232 | 15 |
| 2.4 Sensor Suhu LM35 | 16 |
| 2.5 Cahaya Inframerah | 18 |
| 2.5.1. IC Pembangkit Gelombang | 19 |
| 2.5.2. Penerima Inframerah | 23 |
| 2.6 Borland Delphi 7.0 | 24 |
| 2.6.1. IDE Delphi | 25 |
| 2.6.2. Menu Borland Delphi | 28 |
| 2.6.3. Komunikasi Serial dengan Delphi | 30 |
| BAB III. PERANCANGAN SISTEM | |
| 3.1. Gambaran Umum Sistem | 31 |
| 3.2. Perancangan perangkat Keras | 36 |
| 3.2.1. Rangkaian Mikrokontroler AtMega8535 | 36 |
| 3.2.1.1. Rangkaian Osilator | 36 |
| 3.2.1.2. Rangkaian Power On Reset | 36 |
| 3.2.2. Rangkaian Sensor Suhu..... | 37 |
| 3.2.3. Rangkaian Driver Lampu | 39 |

| | |
|--|----|
| 3.2.4. Rangkaian Pemancar Inframerah | 40 |
| 3.2.4.1. Rangkaian Pembangkit Frekuensi | 40 |
| 3.2.4.2. Rangkaian Penguat | 42 |
| 3.2.5. Rangkaian Penerima Inframerah | 43 |
| 3.2.6. Rangkaian RS232 | 44 |
| 3.2.7. Perancangan Rangkaian ADC | 45 |
| 3.2.8. Rangkaian Catu Daya | 46 |
| 3.3. Perancangan Perangkat Lunak | 47 |
| BAB IV. ANALISIS DAN PEMBAHASAN | |
| 4.1. Pengujian Rangkaian Catu Daya | 50 |
| 4.2. Pengujian Rangkaian Sensor Suhu | 51 |
| 4.3. Pengujian Rangkaian Driver Lampu | 56 |
| 4.4. Pengujian Rangkaian Pemancar Inframerah | 58 |
| 4.5. Pengujian Rangkaian Penerima Inframerah | 62 |
| 4.6. Pengujian ADC Internal ATmega8535 | 63 |
| 4.7. Pengujian Keseluruhan Sistem | 67 |
| BAB V. KESIMPULAN DAN SARAN | |
| 5.1. Kesimpulan | 70 |
| 5.2. Saran | 71 |
| DAFTAR PUSTAKA | |
| LAMPIRAN | |

DAFTAR GAMBAR

| | | |
|---------------|---|----|
| ambar 3.6. R | | |
| ambar 3.7. R | | |
| ambar 3.8. R | Gambar 2.1. Blok Diagram Fungsional ATmega8535 | 7 |
| ambar 3.9. R | Gambar 2.2. Pin ATmega8535 | 9 |
| ambar 3.10. I | Gambar 2.3. Konektor DB-9..... | 14 |
| ambar 3.11. I | Gambar 2.4. Susunan Pin IC MAX232 | 16 |
| ambar 3.12. I | Gambar 2.5. Sensor Suhu LM35 | 17 |
| ambar 4.1. di | Gambar 2.6. Karakteristik Sensor Suhu LM35 | 18 |
| ambar 4.2. G | Gambar 2.7. Blok Diagram IC NE555 | 20 |
| di | Gambar 2.8. Konfigurasi Pin IC NE555..... | 20 |
| ambar 4.3. R | Gambar 2.9. Rangkaian Astable Multivibrator | 21 |
| ambar 4.4. T | Gambar 2.10. Bentuk Gelombang Rangkaian Astable Multivibrator | 22 |
| ambar 4.5. P | Gambar 2.11. Blok Diagram modul TSOP 4838 | 24 |
| ambar 4.6. P | Gambar 2.12. Lembar Kerja Delphi | 25 |
| ambar 4.7. P | Gambar 2.13. Komponen CportLib Tab..... | 30 |
| ambar 4.8. P | Gambar 2.14. Meletakkan Komponen di Form | 30 |
| ambar 4.9. S | Gambar 2.15. Box Dialog Setting Serial | 31 |
| ambar 4.10. I | Gambar 2.16. Dialog dan Program Menerima Data..... | 32 |
| ambar 4.11. I | Gambar 3.1. Diagram Blok Sistem..... | 33 |
| ambar 4.12. I | Gambar 3.2. Rangkaian Osilator | 36 |
| | Gambar 3.3. Rangkaian Power On Reset | 37 |
| | Gambar 3.4. Rangkaian Pengkondisi Sinyal Sensor Suhu | 38 |
| | Gambar 3.5. Rangkaian Driver Pemanas..... | 39 |

DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1. Konfigurasi Pin Serial..... | 15 |
| Tabel 4.1. Tegangan Keluaran Rangkaian Catu Daya .. | 51 |
| Tabel 4.2. Perbandingan Tegangan Hasil Perhitungan Sensor dengan Hasil Pengukuran output sensor LM35 | 53 |
| Tabel 4.3. Perbandingan antara Pengukuran Suhu Menggunakan Termometer dengan Sensor LM35 | 54 |
| Tabel 4.4. Persen Kesalahan Tegangan Output LM35 Antara Hasil Pengukuran dengan Hasil Perhitungan | 56 |
| Tabel 4.5. Pengujian Driver Lampu..... | 57 |
| Tabel 4.6. Hasil Pengujian Driver AC | 58 |
| Tabel 4.7. Pengujian ADC Internal ATmega8525 | 65 |
| Tabel 4.8. Selisih Keluaran ADC Internal ATmega8525..... | 66 |

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi di zaman ini mengalami kemajuan yang sangat pesat sekali seiring dengan perkembangan pola pikir manusia. Suatu ide untuk melakukan penelitian di bidang medis yang perlu dikembangkan untuk memudahkan aktivitas perawatan. Dalam hal ini topik yang diamati adalah perawatan inkubator bayi, inkubator bayi berfungsi untuk menjaga bayi prematur dalam suhu 32°C - 35°C . Dalam penentuan suhu inkubator didasarkan pada berat dan umur bayi prematur tersebut. Bila suhu inkubator tidak sesuai dengan yang ditentukan, maka akibatnya terhadap bayi:

- Bila suhunya kurang dari 32°C bayi akan kedinginan dan bisa terkena penyakit *hipotermia* (kedinginan yang berlebihan).
- Bila suhunya lebih dari 35°C tidak nyaman untuk bayi.

Berdasarkan pengamatan perawatan bayi dalam inkubator yang ada di rumah sakit sekarang masih dilakukan secara manual yaitu membutuhkan campur tangan perawat yang harus mengecek satu per satu suhu inkubator dan menentukan setpoint suhunya sesuai dengan yang diinginkan, hal tersebut dinilai kurang efisien karena suhu inkubator tidak bisa dikendalikan dan dimonitoring secara keseluruhan oleh satu perawat. Padahal perawat juga manusia biasa yang kadang juga bisa lalai,

ketiduran, lupa, teledor, yang nantinya dapat membahayakan bayi yang ada dalam inkubator tersebut.

Hal tersebut diatas yang mendasari dari pembuatan suatu sistem untuk mengendalikan dan memonitoring suhu secara real time dari banyak inkubator berbasis mikrokontroler sebagai pengolah datanya dan PC sebagai penampilnya, Sehingga perawat bisa mengendalikan dan memonitoring suhu dari banyak inkubator melalui PC (*Personal Computer*) secara real time.

1.2 Rumusan Masalah

Berdasarkan dari uraian latar belakang diatas maka dapat dirumuskan suatu permasalahan sebagai berikut :

1. Bagaimana merancang dan merakit hardware beserta software yang dapat mengendalikan dan memonitoring suhu dari banyak inkubator yang ditampilkan pada PC secara real time?
2. Bagaimana merancang sebuah sistem sehingga inkubator tersebut bisa aktif dan tidak aktif secara otomatis berdasarkan keberadaan bayi dalam inkubator tersebut?

1.3 Batasan masalah

Dalam melaksanakan suatu penelitian diperlukan adanya batasan-batasan agar tidak menyimpang dari yang telah direncanakan sehingga tujuan yang

sebenarnya dapat dicapai. Adapun batasan-batasan yang diperlukan yaitu sebagai berikut :

1. Sistem yang dirancang dapat mengendalikan dan memonitoring suhu inkubator pada range 32°C - 35°C.
2. Miniatur inkubator dirancang dengan menggunakan akrilik untuk 3 bayi (3 inkubator).
3. Sistem yang dirancang dapat mendeteksi keberadaan bayi sehingga dapat ON atau OFF secara otomatis.
4. Hanya membahas pengendalian dan monitoring suhu inkubator.
5. Sistem pengendali suhu yang digunakan yaitu dengan sistem kendali ON/OFF.

1.4 Tujuan dan Manfaat

Tujuan yang akan dicapai dalam penulisan Tugas Akhir ini adalah dapat membangun dan merealisasikan suatu sistem yang dapat memonitoring dan mengendalikan suhu terpusat dari banyak inkubator melalui PC.

1.5 Sistematika penulisan

Sistem penulisan laporan tugas akhir ini di bagi menjadi lima bab, sistematika penulisannya sebagai berikut :

Demikian juga dalam perancangan sebelumnya, hanya dapat mengendalikan suhu dan kelembaban dalam satu ruangan saja dan menampilkan data suhu pada sebuah LCD. Sedangkan pada perancangan yang dibuat dapat mengendalikan dan memonitoring suhu lebih dari satu ruangan dan menampilkannya pada PC, dan juga dapat mendeteksi keberadaan orang yang ada didalam ruangan tersebut sehingga dapat mengaktifkan dan menonaktifkan sistem secara otomatis.

Pada studi pustaka ini, ada beberapa landasan teori dari komponen utama yang digunakan, yaitu

2.2 Mikrokontroler

Perkembangan teknologi telah maju dengan pesat dalam perkembangan dunia elektronika, khususnya dunia mikroelektronika. Penemuan silikon menyebabkan bidang ini mampu memberikan sumbangan yang amat berharga bagi perkembangan teknologi modern. Atmel sebagai salah satu vendor yang mengembangkan dan memasarkan produk mikroelektronika telah menjadi suatu teknologi standar bagi para desainer elektronika masa kini.

Mikrokontroler yang digunakan dalam perancangan tugas akhir ini yaitu menggunakan ATmega8535 yang merupakan generasi AVR (*Alf and Vegard's Risc Processor*) dari vendor ATMEL. Mikrokontroler AVR memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (*16-bits words*) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus clock. Gambar 2.2 memperlihatkan arsitektur yang dimiliki mikrokontroler ATmega8535.

2.2.1 Arsitektur Mikrokontroler ATmega8535

Adapun blok diagram dari Arsitektur ATmega8535 :

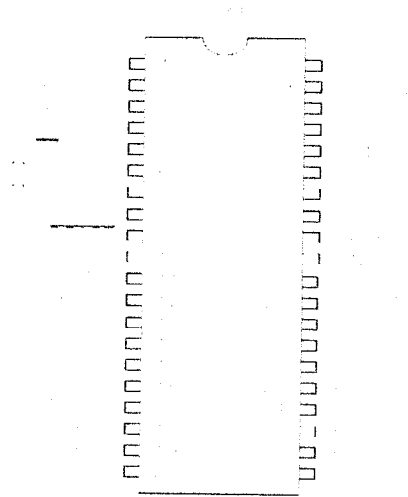


Gambar 2.1 Blok Diagram Fungsional ATmega8535

4. Port komunikasi serial (USART) dengan kecepatan maksimal 2,5 Mbps.
5. Enam pilihan mode *sleep* menghemat penggunaan daya listrik.

2.2.3 Konfigurasi Pin ATmega8535

Konfigurasi pin ATmega8535 adalah sebagai berikut :



Gambar 2.2 Pin ATmega8535

Dari gambar tersebut dapat dijelaskan secara fungsional konfigurasi pin ATmega8535 sebagai berikut :

1. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
2. GND merupakan pin *ground*.
3. Port A (PA0..PA7) merupakan pin I/O dua arah dan pin masukan ADC.
4. Port B (PB0..PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu Timer/Counter, komparator analog, dan SPI.

- e) Tegangan referensi ADC internal 2,56 Volt
- f) Mode konversi *free running* dan *single conversion*
- g) Interrupt on ADC Conversion Complete
- h) 2 kanal input dengan penguatan 10 dan 200 x (differential)

ADC ini menggunakan dua mode, yaitu *single conversion* dan *free running* untuk melakukan konversi sinyal analog yang masuk ke pin input menjadi sinyal digital. Bilangan biner yang mewakili sinyal analog pada suatu harga tertentu dikenal dengan nama *cuplikan* (sample) sedangkan frekuensi untuk mencuplik disebut laju cuplikan (sampling rate).

❖ Resolusi

Resolusi merupakan perubahan yang dapat terjadi pada keluaran digital sebagai hasil perubahan pada input analog. Proses perubahan data analog menjadi data digital tergantung dari jumlah bit pada suatu ADC yang digunakan. Dimana resolusi adalah besaran terkecil (analog) yang masih dapat dikonversikan menjadi satuan digital.

$$\text{Resolusi} = \left(\frac{1}{2^n} \right) \times V_{\text{ref}} \quad (2.1)$$

Dengan n : Banyaknya bit ADC

V_{ref} : Tegangan referensi yang digunakan

❖ Akurasi

Akurasi merupakan spesifikasi yang menunjukkan suatu ADC untuk mengkonversi suatu input analog. Semakin tinggi akurasi yang dimiliki ADC maka

mode *synchronous* harus 3 pin yaitu TXD, RXD dan XCK. Untuk mengatur mode dan prosedur komunikasi USART dilakukan melalui register UCSRA, UCSRB, UCSRC, UBRRH, UBRRL, dan UDR.

2.3 Antar Muka RS-232

Komunikasi atau *transfer* data antara mikrokontroler dengan memori atau piranti lain dapat dilakukan secara serial dan paralel. Transmisi serial berarti mengirimkan data satu bit dalam satu satuan waktu melewati satu jalur, berlawanan dengan pengiriman secara paralel yang mengirimkan sejumlah bit sekaligus dalam satu satuan waktu sehingga memerlukan banyak jalur.

Transmisi data serial hanya memakai sebuah jalur untuk pengiriman data artinya pengiriman data dilakukan dengan mengirimkan satu persatu bit dalam satu satuan waktu. Sedangkan dalam transmisi paralel, beberapa buah bit dikirimkan sekaligus dalam satu satuan waktu. Hal ini akan mempercepat proses pengiriman data atau menaikkan *baud rate* pengiriman.

Keuntungan pengiriman data secara serial adalah jumlah kabel yang digunakan lebih sedikit, jika komunikasi paralel minimal memerlukan delapan kabel plus *ground*, maka komunikasi serial hanya butuh dua kabel plus *ground*, selain itu jangkauan panjang kabel lebih jauh dibandingkan paralel karena port serial mengirimkan logika 1 dengan kisaran tegangan -3 volt hingga -15 volt dan logika 0 sebagai +3 volt hingga +15 volt, sehingga kehilangan daya karena panjangnya kabel bukan masalah utama. Bandingkan hal ini dengan port paralel yang menggunakan

Tabel 2.1 menerangkan kegunaan pin konektor DB-9 dalam proses komunikasi data secara serial.

Tabel 2.1 Konfigurasi Pin Serial

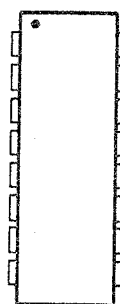
| Pin DB9 | Nama | Kepanjangan |
|----------------|-------------|----------------------------|
| 3 | TD | <i>Transmit Data</i> |
| 2 | RD | <i>Receive Data</i> |
| 7 | RTS | <i>Request To Send</i> |
| 8 | CTS | <i>Clear To Send</i> |
| 6 | DSR | <i>Data Set Ready</i> |
| 5 | SG | <i>Signal Ground</i> |
| 1 | CD | <i>Carrier Detect</i> |
| 4 | DTR | <i>Data Terminal Ready</i> |
| 9 | RI | <i>Ring Indicator</i> |

2.3.2 Pengkonversi Level Tegangan TTL ke Level Tegangan RS-232

Untuk mengkonversi level tegangan TTL ke level tegangan RS-232 digunakan IC MAX 232. Tegangan tingkat RS-232 sangat jauh berbeda dengan tingkat TTL. Jika TTL bekerja dengan tegangan antara 0 sampai 5 volt, dengan tegangan sekitar 0 volt dianggap sebagai logika '0' dan tegangan disekitar 5 volt sebagai logika '1', sedangkan untuk tingkat RS232 tegangan kerjanya antara -15

sampai +15 volt dan cara menerjemahkan logika '0' dan '1' -nya yang sangat berbeda.

Untuk itu diperlukan piranti khusus yang digunakan untuk melakukan konversi tingkat tegangan TTL dan RS-232. IC MAX 232 ini dapat digunakan dalam dua arah konversi tegangan yaitu dari level tegangan RS-232 ke level tegangan TTL atau sebaliknya. Gambar 2.4 memperlihatkan konfigurasi pin dari IC MAX 232.



Gambar 2.4 Susunan Pin IC MAX 232

2.4 Sensor Suhu LM35

Sensor suhu LM35 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. Sensor Suhu LM35 yang dipakai dalam penelitian ini berupa komponen elektronika yang diproduksi oleh *National Semiconductor*. LM35 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, LM35 juga mempunyai keluaran impedansi yang rendah dan linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali khusus serta tidak memerlukan penyetelan lanjutan.

Meskipun tegangan sensor ini dapat mencapai 30 volt akan tetapi yang diberikan kesensor adalah sebesar 5 volt, sehingga dapat digunakan dengan catu daya tunggal dengan ketentuan bahwa LM35 hanya membutuhkan arus sebesar 60 μA hal ini berarti LM35 mempunyai kemampuan menghasilkan panas (*self-heating*) dari sensor yang dapat menyebabkan kesalahan pembacaan yang rendah yaitu kurang dari 0,5 $^{\circ}\text{C}$ pada suhu 25 $^{\circ}\text{C}$.



Gambar 2.5 Sensor Suhu LM35

Gambar diatas menunjukkan bentuk dari LM35 tampak depan dan tampak bawah. 3 pin LM35 menunjukan fungsi masing-masing pin diantaranya, pin 1 berfungsi sebagai sumber tegangan kerja dari LM35, pin 2 atau tengah digunakan sebagai tegangan keluaran atau V_{OUT} dengan jangkauan kerja dari 0 Volt sampai dengan 1,5 Volt dengan tegangan operasi sensor LM35 yang dapat digunakan antar 4 Volt sampai 30 Volt. Keluaran sensor ini akan naik sebesar 10 mV setiap derajat *celcius* sehingga diperoleh persamaan sebagai berikut

$$V_{LM35} = \text{Temp} \times 10 \text{ mV} \quad (2.3)$$

Dengan :

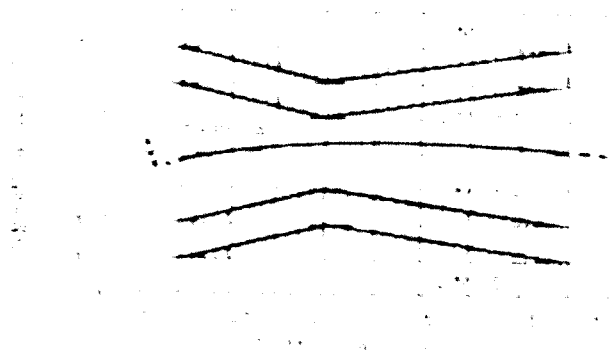
V_{LM35} : Tegangan output dari sensor LM35 (Volt)

Temp : Besarnya suhu yang dibaca sensor ($^{\circ}\text{C}$)

Spesifikasi dari Sensor Suhu LM35:

- Dikalibrasi secara langsung ke celcius
- Faktor skala linier +10 mV/°C
- Jaminan akurasinya 0,5 °C pada suhu +25 °C
- Rata-rata temperaturnya antara -55 °C sampai 150 °C.
- Dioperasikan pada 4 volt sampai 30 volt.
- Arus yang dibawa kurang dari 60 uA
- *Self-heating*-nya yaitu 0,08 °C pada udara tetap.
- Ketidak linieranya $\pm 0,25$ °C.
- Impedansi outputnya rendah yaitu 0,1 Ω dari 1 mA muatan

Karakteristik dari sensor suhu LM35



Gambar 2.6 Karakteristik Sensor LM35

2.5 Cahaya Inframerah

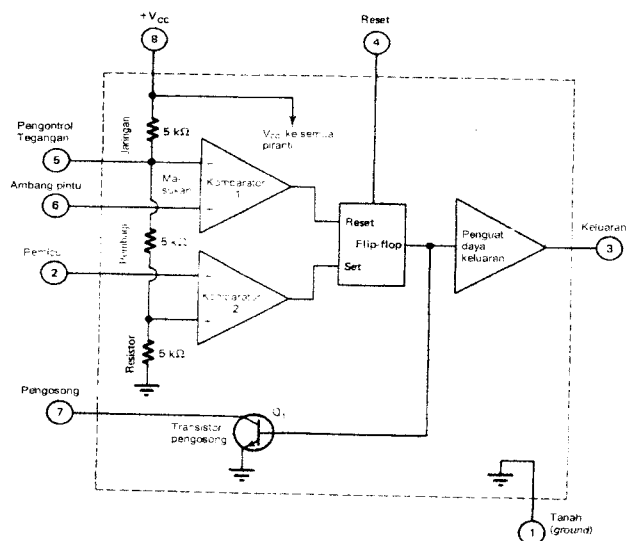
Cahaya inframerah merupakan cahaya yang tidak tampak. Jika di lihat dengan spektroskop cahaya, maka radiasi cahaya inframerah akan nampak pada

spektrum elektromagnet dengan panjang gelombang diatas panjang gelombang cahaya merah yaitu 10^{-3} m sampai $7,8 \times 10^{-7}$ m dengan frekuensi pada range 10^{12} Hz sampai 10^{14} Hz. Dengan panjang gelombang ini maka cahaya inframerah ini akan tidak nampak oleh mata, namun radiasi panas yang ditimbulkannya masih terasa atau dideteksi.

Cahaya inframerah walaupun mempunyai panjang gelombang yang sangat panjang, tetapi tidak dapat menembus bahan-bahan yang tidak dapat melewatkan cahaya yang nampak sehingga cahaya inframerah tetap mempunyai karakteristik seperti halnya cahaya yang nampak oleh mata

2.5.1 IC Pembangkit Gelombang

IC NE/SE 555 adalah piranti multiguna yang telah secara luas digunakan. Piranti ini dapat digunakan sebagai astable multivibrator. Rangkaian khusus ini dapat dibuat dengan komponen dan daya yang minimal. Rangkaian dapat dengan mudah dibuat dan sangat reliabel. Rangkaian internal IC NE555 biasanya dilihat sebagai blok-blok. Dalam hal ini chip memiliki 2 komparator, sebuah bistable flip-flop, sebuah pembagi resistif, sebuah transistor pengosongan dan sebuah keluaran. Gambar berikut memperlihatkan blok fungsional IC NE555.

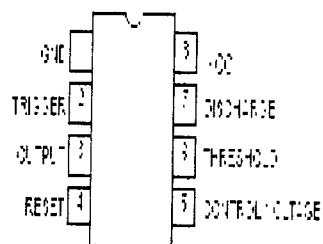


Gambar 2.7 Blok Diagram IC NE555

Rangkaian IC NE555 dikemas dalam bentuk DIP (*Dual in Line Package*) 8 pin.

Secara garis besar memiliki 4 komponen internal yaitu :

- Output drive
- Pengosongan transistor
- Pembanding (komparator)
- Flip-flop pengendali

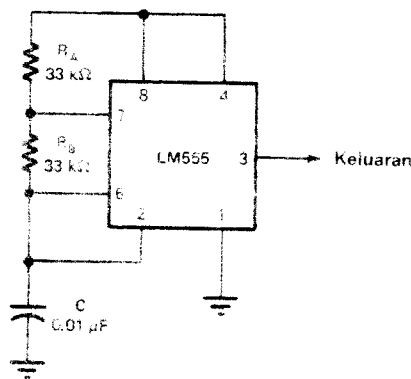


Gambar 2.8 Konfigurasi Pin IC NE555

Fungsi masing-masing pin IC NE555 adalah sebagai berikut :

- Pin 1 : ground (pentanahan)
- Pin 2 : sebagai pemicu (trigger)
- Pin 3 : sebagai keluaran (output)
- Pin 4 : sebagai reset
- Pin 5 : mengendalikan tingkat keluaran tegangan picu dengan tegangan ambang, memodulasi bentuk gelombang keluaran, melewati gangguan riak tegangan yang mungkin muncul dari catu daya bila dihubungkan dengan kapasitor ke ground.
- Pin 6 : terminal ambang
- Pin 7 : sebagai pengosongan muatan yang dihasilkan selama proses
- Pin 8 : sebagai catu daya (VCC)

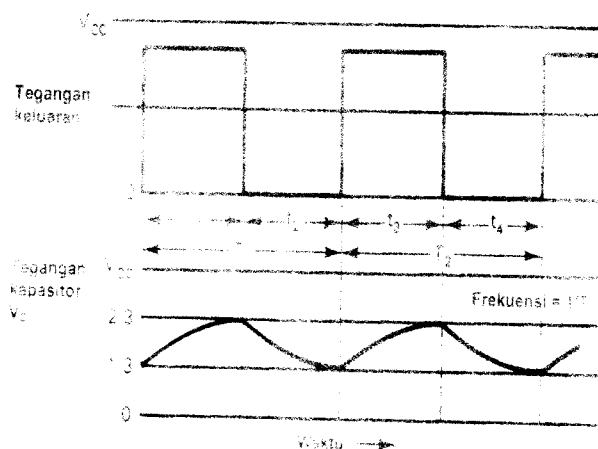
Rangkaian astable dibuat dengan mengubah susunan resistor dan kapasitor luar pada IC 555. Ada dua buah resistor R_a dan R_b serta satu kapasitor eksternal yang diperlukan seperti terlihat pada gambar berikut :



Gambar 2.9 Rangkaian Astable Multivibrator

Saat daya mula-mula diberikan, kapasitor akan terisi melalui R_a dan R_b . Ketika tegangan pada pin 6 ada sedikit kenaikan di atas dua pertiga V_{cc} , maka terjadi perubahan kondisi pada komparator 1. ini akan mereset flip-flop dan keluaran akan bergerak ke positif. Keluaran (pin 3) bergerak ke "tanah" dan basis Q1 berprategangan maju. Q1 mengosongkan C lewat R_b ke "tanah".

Ketika tegangan pada kapasitor C turun sedikit dibawah sepertiga V_{cc} , ini akan memberikan energi ke komparator 2. antara pemicu (pin 2) dan pin 6 masih terhubung bersama. Komparator 2 menyebabkan tegangan positif ke masukan set dari flip-flop dan memberikan keluaran negatif. Keluaran (pin 3) akan bergerak ke harga $+V_{cc}$. Tegangan Q1 berpanjar mundur. Ini akan membuka proses pengosongan (pin 7). C mulai terisi lagi dengan harga berkisar antara sepertiga dan dua pertiga V_{cc} . Perhatikan gelombang yang dihasilkan pada gambar berikut :



Gambar 2.10 Bentuk Gelombang pada Rangkaian Astable Multivibrator

Frekuensi keluaran astable multivibrator dinyatakan sebagai $f = 1/T$. Ini menunjukkan sebagai total waktu yang diperlukan untuk pengisian dan pengosongan kapasitor C. Waktu pengisian ditunjukkan oleh jarak t_1 dan t_3 . jika dinyatakan dalam detik $t_1 = 0,693 (R_A + R_B)C$. Waktu pengosongan diberikan oleh t_2 dan t_4 . Dalam detik $t_2 = 0,693 R_B C$. Dalam satu putaran atau satu periode pengoperasian waktu yang diperlukan adalah sebesar $T = t_1 + t_2$ atau $T = t_3 + t_4$. Dengan menggunakan harga t_1 dan t_2 atau t_3 dan t_4 , maka persamaan frekuensi dapat dinyatakan sebagai :

$$f = \frac{1}{T} = \frac{1,44}{(R_A + 2R_B)C} \quad (2.4)$$

Dengan : f : Frekuensi (Hz)

T : Periode (Sekon)

R : Resistansi (Ohm)

C : Kapasitansi (Farad)

1.5.2 Penerima Inframerah

Modul penerima TSOP 4838 merupakan suatu modul penerima data melalui gelombang inframerah dengan frekuensi carrier sebesar 38 KHz. Modul penerima inframerah terdiri dari PIN diode sebagai photo detektor dan preamplifier sebagai penguat dalam satu kemasan. Dalam modul ini dilengkapi juga dengan bandpass filter yang hanya dapat melewatkan frekuensi tertentu. Modul ini terbungkus oleh plat yang terhubung dengan ground rangkaian untuk melindungi rangkaian dari interferensi noise. Adapun blok diagram dari modul penerima TSOP 4838

BAB I : Pendahuluan

Bab ini menjelaskan tentang latar belakang topik yang diangkat menjadi tugas akhir, definisi masalah dari topik yang diangkat, indikator atau *performance criteria* yang dijadikan sebagai tolok ukur dalam melakukan pengujian dan analisis sistem yang dibuat

BAB II : Tinjauan Pustaka

Bab ini memuat tinjauan pustaka atau literatur yang berhubungan dengan penelitian yang sedang dilakukan dan menjelaskan rencana penelitian yang akan dilakukan.

BAB III : Perancangan Sistem

Bab ini membahas tentang penjabaran metodologi, penjelasan diagram blok atau diagram alir sistem yang dibuat, dan penjabaran mengenai indikator unjuk kerja sistem, tentang bagaimana validasi atau pengujian sistem akan dilakukan.

BAB IV : Pengujian, Analisis dan Pembahasan

Bab ini berisi penjelasan analisis hasil pengujian sistem yang dibuat dibandingkan dengan kriteria hasil pengujian yang telah ditentukan.

BAB V : Penutup

Bab ini berisi kesimpulan dari tugas akhir yang telah selesai dikerjakan berdasarkan analisis dan pembahasan di bab sebelumnya, saran untuk pengembangan dan penelitian lebih lanjut.

BAB II

TINJAUAN PUSTAKA

2.1 Studi Pustaka

Pada perancangan sebelumnya yaitu dari penelitian dengan judul “Sistem Pengkondisi Suhu dan Kelembaban Udara pada Ruang Berbasis Mikrokontroler AT89C51” (Oleh Pandu Mayor Hermawan, 2001), terdapat perbedaan yang signifikan dari perancangan “Sistem Monitoring dan Pengendali Suhu Terpusat pada Banyak Inkubator Berbasis Mikrokontroler dan Personal komputer (PC)” yang dibuat. Pada dasarnya, alat yang dibuat menggunakan sensor suhu LM35 sebagai pengindera suhu sama halnya dengan perancangan sebelumnya. Letak perbedaannya yaitu pada perancangan yang dibuat sekarang ini dapat mengendalikan dan memonitoring suhu dari banyak ruangan (inkubator) kemudian ditampilkannya pada sebuah PC.

Dalam perancangan sebelumnya menggunakan mikrokontroler AT89C51 sebagai pengolah data keseluruhan sistem. Sedangkan alat yang dibuat sekarang menggunakan mikrokontroler tipe ATmega8535 sebagai pengolah datanya. Dalam hal instruksi, terdapat perbedaan, yaitu mikrokontroler AT89C51 mempunyai waktu untuk eksekusi instruksi yang lebih lambat dibandingkan dengan mikrokontroler ATmega8535. Dimana, AT89C51 memerlukan 12 siklus clock untuk melaksanakan satu siklus instruksi, sedangkan ATmega8535 hanya membutuhkan 1 siklus saja dalam melakukan eksekusi intruksi.

Jendela IDE Delphi 7.0 mempunyai perangkat-perangkat yang dapat dipergunakan untuk memudahkan seorang programmer dalam membuat program.

Perangkat-perangkat tersebut seperti terlihat pada gambar 2-10 diantaranya adalah:

1. *Main Window*

Jendela utama ini adalah bagian dari IDE yang mempunyai fungsi yang sama dengan semua fungsi utama dari program aplikasi Windows lainnya. Jendela utama Delphi terbagi menjadi tiga bagian, berupa Main Menu, Toolbar dan Component Palette.

2. *Main Menu*

Menu utama pada Delphi memiliki kegunaan yang sama seperti program aplikasi Windows lainnya. Dengan menggunakan fasilitas menu, dapat memanggil atau menyimpan program. Pada dasarnya semua perintah yang diberikan dapat ditemukan pada bagian menu utama ini.

3. *Toolbar*

Dengan *toolbar* dapat melakukan beberapa operasi pada menu utama dengan sebuah klik tunggal. Setiap tombol pada *toolbar* mempunyai sebuah *tooltip* yang berisi informasi mengenai fungsi dari tombol tersebut.

4. *Component Palette*

Component Palette merupakan bagian yang digunakan untuk meletakkan berbagai komponen yang sesuai dengan kategorinya. Pada bidang ini semua komponen yang merupakan bawaan dari Delphi, baik berupa komponen visual maupun komponen nonvisual. Komponen-komponen tersebut berguna untuk

mendesain user interface (antarmuka pemakai) dari program yang sedang dibuat. Borland Delphi sendiri memungkinkan untuk menambahkan komponen dari luar, baik yang dibuat sendiri ataupun komponen dari pihak ketiga (third party).

5. *Form Designer*

Merupakan sebuah bidang jendela (*window*) yang masih kosong. Pada bidang ini dapat menempatkan komponen-komponen visual dan nonvisual untuk mendesain *user interface* program. Ketika menjalankan Borland Delphi 7, secara otomatis *Form Designer* akan memanggil sebuah form yang bernama Form1.

6. *Object Inspector*

Dengan perangkat ini dapat mengubah *property* dan *event* pada setiap *object* atau komponen. Object atau komponen yang satu dengan yang lain mempunyai *property* dan *event* yang berbeda. Jika menempatkan sebuah komponen pada suatu Form, komponen tersebut akan berisi nilai *default* dari Delphi. Misalnya komponen TButton akan berisi nilai 25 dan 75 untuk *property* height dan width. Nilai-nilai *property* tersebut dapat diubah kemudian. Baik pada saat perancangan *interface* program (*design time*) maupun pada saat program berjalan (*run time*) dengan menggunakan kode program.

7. *Code Editor*

Pada bidang ini dapat menuliskan kode-kode program dan logika program dalam bahasa Delphi untuk mengatur jalannya program. Antara Form Designer dan Code Editor merupakan dua bagian yang berkaitan, tidak bias hanya mendesain

2.6.3 Komunikasi Serial dengan Delphi

Program interface dengan komputer bisa dilakukan dengan banyak cara salah satunya secara serial. Transfer data secara serial berarti juga data dikirim dari devais luar misalnya mikrokontroller ke komputer secara serial dengan standard yang telah ditentukan. Data dikirim per 8 bit dengan bit start dan bit stop bisa juga ditambahkan parity. Delphi yang digunakan disini adalah delphi version 7. Delphi 7 tidak mempunyai package serial sehingga perlu di instal terlebih dahulu. Setelah menginstall serial package maka akan muncul CportLib tab. dengan komponen seperti dibawah ini:



Gambar 2.13 Komponen CportLib Tab

untuk memulai membuat program, kita cukup mendrag dan meletakkan komponen di form akan kita buat, seperti gambar dibawah:



Gambar 2.14 Meletakkan Komponen di Form

langkah selanjutnya, kita buat program untuk memanggil dialog setting dari parameter serial yang akan kita gunakan. Programnya adalah cukup dengan menuliskan syntax

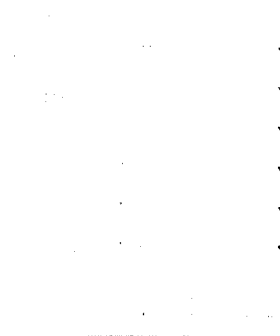
yang berada dibawah ini:

```

procedure TForm1.Button1Click(Sender: TObject);
begin
  Serial1.SerialName := 'COM1';
  Serial1.BaudRate := 9600;
  Serial1.Parity := 'None';
  Serial1.StopBits := 'One';
  Serial1.DataBits := 8;
  Serial1.Open;
end;

```

Jika program diatas dieksekusi, maka akan tampil dialog box setting serial seperti pada gambar dibawah ini:



Gambar 2.15 Box Dialog Setting Serial

Setelah setup serialnya benar, sekarang yang akan kita lakukan adalah membuat program untuk mengirim dan menerima data.

1. Program Mengirim Data

Untuk membuat program mengirim data maka perlu dipersiapkan sebuah variabel bisa bertipe string atau integer. Untuk memulai proses pengiriman data maka com serial harus dibuka atau dikoneksikan terlebih dahulu dengan delphinya. Adapun program transmit data seperti yang terlihat dibawah

```

procedure TForm1.Button1Click(Sender: TObject);
begin
  Serial1.Send('1234567890');
end;

```

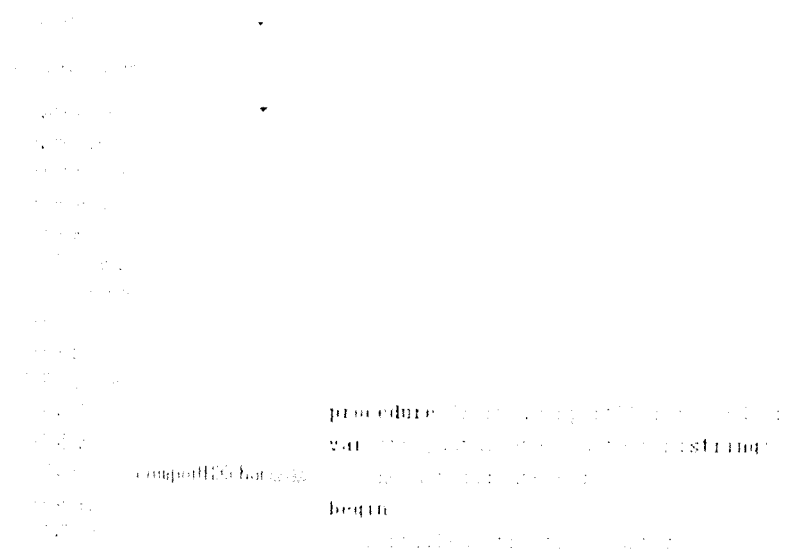
Maksud program :

comport1.Open : open koneksi delphi dengan com serial komputer

comport1.WriteStr(str) : transfer data string (data yang ditransfer bertipe string)

2. Program Menerima Data

Untuk membuat program menerima data dapat dilihat pada gambar dibawah :



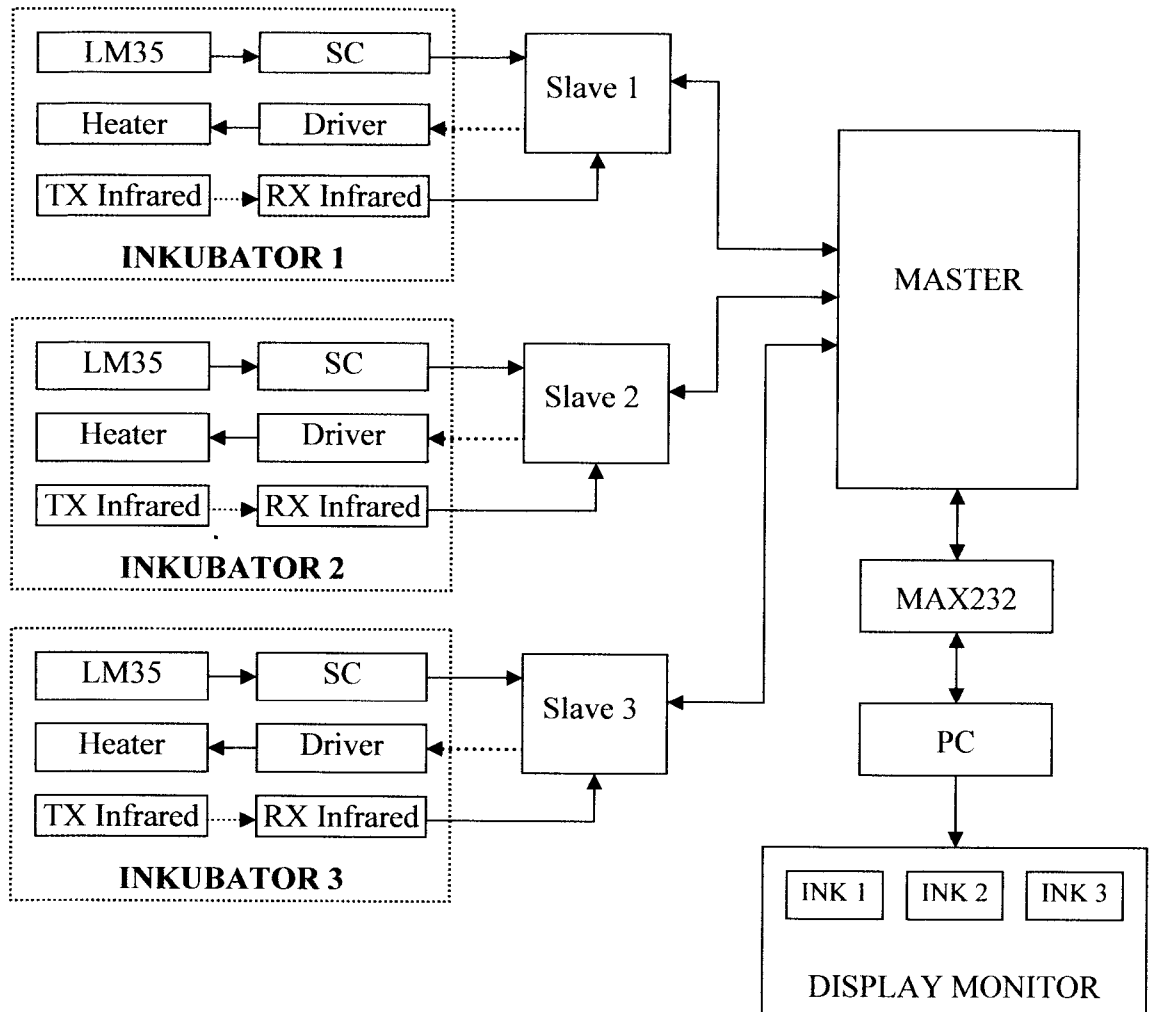
Gambar 2.16 Dialog dan Program Meherima Data

Maksud Program :

comport1.ReadStr(str5,count) menerima data dari luar dan dipindahkan ke variabel str5 yang bertipe string.

BAB III
PERANCANGAN SISTEM

3.1 Gambaran Umum Sistem



Gambar 3.1 Diagram Blok Sistem

Pada prinsipnya sistem yang dibangun adalah sistem monitoring dan pengendalian suhu secara real time dari banyak inkubator dalam suatu ruangan serta dapat juga mendeteksi keberadaan bayi dalam ruangan tersebut. Adapun penampil suhunya menggunakan personal komputer (PC) sehingga bisa memonitoring dari banyak inkubator. Suhu yang dibutuhkan untuk menjaga bayi prematur adalah dalam range 32°C sampai 35°C. Dalam penentuan suhu inkubator di dasarkan pada umur dan berat badan dari bayi prematur tersebut.

Dari diagram blok diatas dapat dijelaskan tentang spesifikasi alat yang dirancang yaitu sebagai berikut:

- Dalam perancangan terdapat tiga buah inkubator yang terbuat dari bahan akrilik yang masing-masing inkubator terdapat sensor suhu, pemanas (heater), dan sensor pemancar infrared serta penerima infrared.
- Sebagai pengolah datanya menggunakan 3 buah mikrokontroler ATmega8535 yang difungsikan sebagai slave dan 1 buah mikrokontroler ATmega8535 yang difungsikan sebagai master. Hal ini dikarenakan ATmega8535 sudah terdapat ADC internal sehingga tidak diperlukan ADC eksternal.
- Sensor suhu berfungsi untuk mendeteksi suhu inkubator, yaitu menggunakan IC LM35. Hal ini dikarenakan IC LM35 memenuhi syarat sebagai sensor suhu yang baik yaitu mempunyai sensitifitas dan linearitas tinggi. Sedangkan jenis yang dipakai pada sistem ini adalah LM 35D dengan jangkauan suhu 0°C sampai +100°C. Selain itu IC LM 35 juga mudah didapatkan di pasaran serta mempunyai karakteristik perubahan tegangan terhadap suhu dalam skala °C,

sehingga tidak diperlukan untai tambahan untuk mengkalibrasi skala output ke °C. Karakteristik perubahan tegangan yang dimiliki IC LM 35 adalah 10 mV/°C.

- Heater berfungsi sebagai pemanas suhu inkubator, yaitu menggunakan bola lampu 40 Watt.
- Pemancar dan penerima inframerah berfungsi sebagai sensor untuk mendeteksi keberadaan orang dalam inkubator tersebut, untuk pemancar inframerah yaitu menggunakan IC NE555 sebagai pembangkit frekuensi 38 KHz. Sedangkan penerima inframerah menggunakan modul penerima inframerah TSOP 4838 yang hanya dapat melewatkan frekuensi 38 KHz.
- Komunikasi data antara mikrokontroler dengan PC menggunakan serial, keuntungannya yaitu jumlah kabel yang digunakan lebih sedikit. Selain itu jangkauan panjang kabel lebih jauh dibanding secara paralel karena port serial mengirim logika 1 dengan kisaran tegangan -3 volt hingga -15 volt dan logika 0 sebagai +3 volt hingga +15 volt, sehingga kehilangan daya karena panjangnya kabel bukan masalah utama. Berbeda dengan port paralel yang menggunakan level TTL yang berkisar dari 0 volt untuk logika 0 dan +5 volt untuk logika 1.

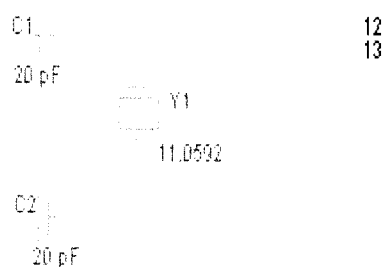
3.2 Perancangan Perangkat Keras

3.2.1 Rangkaian Mikrokontroler ATmega8535

Rangkaian sistem minimum mikrokontroler ATmega8535 terdiri dari rangkaian osilator dan rangkaian *power on reset*.

3.2.1.1 Rangkaian Osilator

Mikrokontroler memiliki osilator internal yang digunakan sebagai sumber detak (*clock*) bagi CPU. Untuk menggunakannya, dihubungkan sebuah resonator kristal atau keramik dengan frekuensi 11,0592 MHz diantara kaki-kaki XTAL1 dan XTAL2 mikrokontroler dengan kapasitor bernilai 20 pF ke *ground*. Rangkaian osilator pada mikrokontroler dapat dilihat pada gambar 3.4.

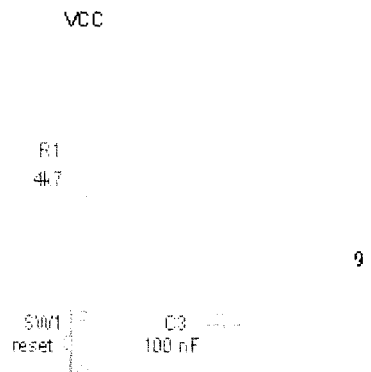


Gambar 3.2 Rangkaian Osilator

3.2.1.2 Rangkaian *Power On Reset*

Mikrokontroler direset pada saat transisi tegangan rendah ke tegangan tinggi dan mikrokontroler mengeksekusi program pada saat reset (RST) dalam keadaan logika rendah oleh karena itu pada pin reset dipasang resistor yang terhubung ke *Vcc* dan kapasitor yang terhubung ke *ground*, dengan demikian mikrokontroler akan

direset setiap kali saklar SW2 ditekan. Gambar 3.5 merupakan rangkaian *power on reset*.



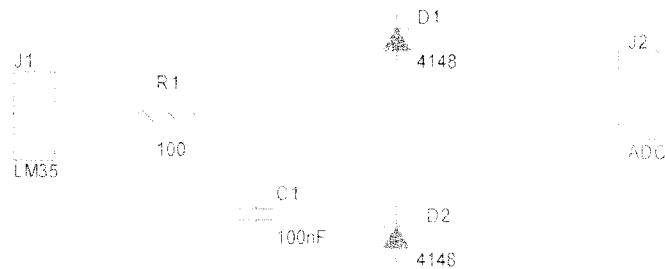
Gambar 3.3 Rangkaian *Power On Reset*

3.2.2 Rangkaian Sensor Suhu

Sensor suhu berfungsi sebagai pengindera suhu dalam inkubator dan mengubah informasi tersebut menjadi tegangan analog. Digunakan sensor LM35 dengan pertimbangan antara lain sederhana rangkaiannya, keluarannya linier terhadap suhu, kepekaan cukup baik, terkalibrasi langsung dalam derajat celcius, serta murah dan mudah didapatkan.

Kepekaan sensor terhadap suhu adalah sebesar 0.01 Volt/°C dengan akurasi sebesar $\pm 2^\circ\text{C}$. Dengan menghubungkan pin Gnd ke tanah, maka batas bawah keluarannya adalah 0 Volt untuk 0°C sehingga keluarannya sebesar 1 Volt pada 100°C. Karena suhu yang dibutuhkan antara 32°C sampai 35°C, maka suhu yang dibahas dibatasi antara 25°C sampai 40°C.

Rangkaian pengkondisi sinyal digunakan untuk menstabilkan sinyal yang diterima dari sensor sehingga dapat diterima dan diproses oleh ADC. Rangkaian ini menggunakan sebuah resistor dan kapasitor serta dua buah diode. Untuk keluaran dari rangkaian pengkondisi sinyal ini terhubung langsung ke pin 40 PA0/ADC0 mikrokontroler yang difungsikan sebagai slave. Berikut ini pada gambar 3.4 adalah rangkaian pengkondisi sinyal sensor suhu :



Gambar 3.4 Rangkaian Pengkondisi Sinyal Sensor Suhu

Untuk menghitung tegangan output dari sensor ini pada saat membaca temperature adalah :

$$V_{out} = \text{Temp} \times 10\text{mV}/^{\circ}\text{C} \quad (3.1)$$

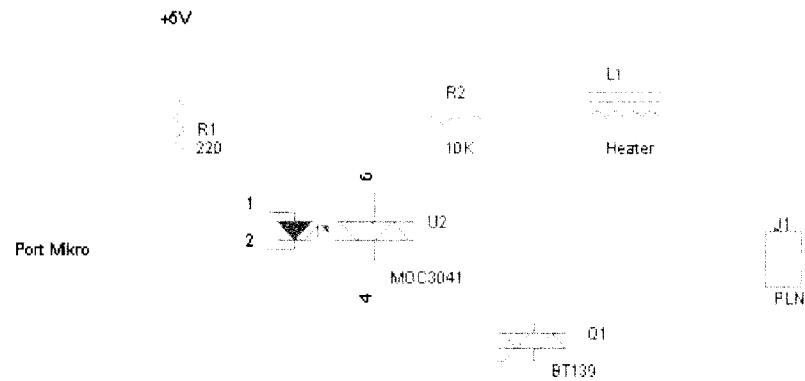
Dengan :

V_{out} : Tegangan output dari sensor LM35 (Volt)

Temp : Besarnya suhu yang dibaca sensor ($^{\circ}\text{C}$)

3.2.3 Rangkaian Driver Lampu

Untuk perancangan driver heater (pemanas) digunakan optotriac karena heater disini memakai supply tegangan 220 VAC.



Gambar 3.5 Rangkaian Driver Pemanas

Tegangan output maksimal dari mikrokontroler adalah 5 Volt sedangkan untuk heater memerlukan supply 220VAC, maka diperlukan rangkaian driver untuk mengendalikannya. Rangkaian driver yang dipakai berupa optotriac MOC 3041 dan triac BT 139, untuk analisa data yang digunakan:

$$V_{in} = 5 \text{ Volt}$$

Data Sheet untuk mengaktifkan MOC 3041:

$$V_F \text{ (tegangan forward dioda)} = 1,5 \text{ Volt}$$

$$I_{FT} \text{ (arus forward Trigger)} = 15 \text{ mA}$$

Maka untuk mengaktifkan optotriac, Resistor yang dipasang:

$$R = \frac{V_m - V_F}{I_{FT}} \quad (3.2)$$

$$R = \frac{5V - 1,5V}{15mA}$$

$$R = \frac{3,5V}{15mA}$$

$$R = 233 \Omega$$

Karena nilai resistor 233Ω tidak ada dipasaran, maka digunakan resistor yang mendekati yaitu 220Ω .

Karena tegangan jaringan yang digunakan adalah $220V$ dan arus gate maksimum (I_{GTM}) adalah 25 mA , maka total nilai R adalah :

$$R = \frac{V_{jaringan}}{I_{GTM}} = \frac{220}{25 \cdot 10^{-3}} = 8,8 K\Omega$$

Karena nilai resistor $8,8 K\Omega$ tidak ada dipasaran, maka digunakan resistor yang mendekati yaitu $10 K\Omega$.

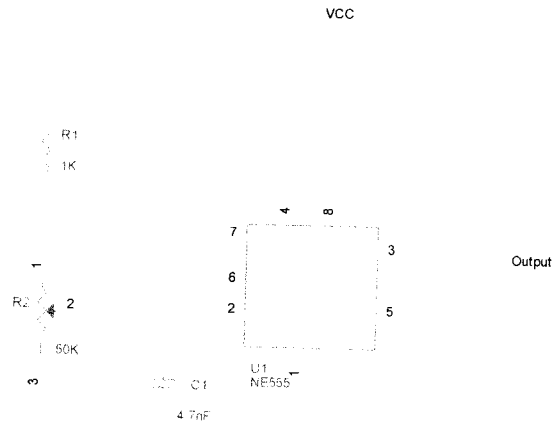
3.2.4 Rangkaian Pemancar Inframerah

Rangkaian pemancar inframerah terdiri dari rangkaian pembangkit frekuensi dan rangkaian penguat.

3.2.4.1 Rangkaian pembangkit frekuensi

Rangkaian pembangkit frekuensi atau osilator digunakan untuk membangkitkan frekuensi inframerah. Frekuensi yang dibangkitkan disetting sesuai

astable multivibrator, IC NE 555 berlaku sebagai osilator RC. Dimana nilai R dan C menentukan bentuk gelombang dan frekuensi keluaran.



Gambar 3.6 Rangkaian Pembangkit Frekuensi

Adapun perancangan pembangkit frekuensi dengan IC NE 555 adalah sebagai berikut :

Frekuensi yang dibangkitkan : 38 KHz

$$f = \frac{1,44}{(R_1 + 2R_2)C_1} \quad (3.3)$$

Dengan menentukan nilai R_1 dan C_1 yaitu :

R_1 : 1K

C_1 : 4,7 nF

Maka nilai R_2 adalah sebagai berikut :

$$R_2 = \frac{1}{2} \left(\frac{1,44}{f.C} - R_1 \right)$$

$$R_2 = \frac{1}{2} \left(\frac{1,44}{38.10^3 \cdot 4,7.10^{-9}} - 1000 \right)$$

$$R_2 = \frac{1}{2} \left(\frac{1,44}{178,6.10^{-6}} - 1000 \right)$$

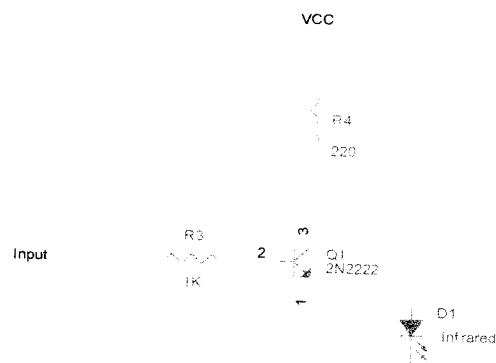
$$R_2 = \frac{1}{2} (8062,7099 - 1000)$$

$$R_2 = \frac{1}{2} (7062,7099)$$

$$R_2 = 3,531 \text{ K}\Omega$$

3.2.4.2 Rangkaian Penguat

Rangkaian ini terdiri dari transistor jenis NPN yaitu 2N2222 dan dua buah resistor $1\text{K}\Omega$ dan 220Ω . Rangkaian ini berfungsi untuk menguatkan sinyal yang keluar dari rangkaian pembangkit frekuensi kemudian dipancarkan melalui infrared.

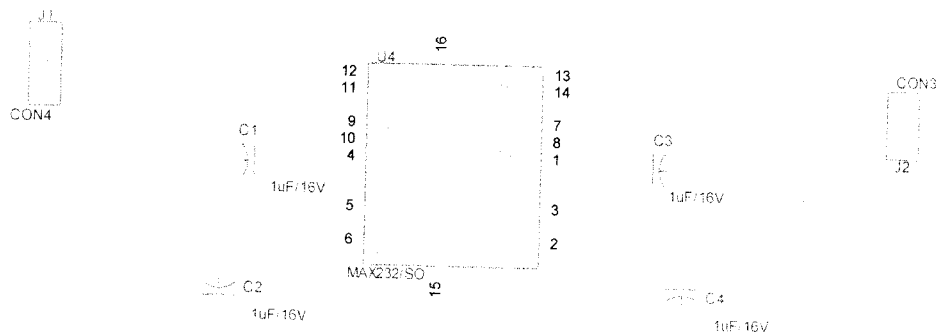


Gambar 3.7 Rangkaian Penguat

tidak mendapat sinyal inframerah (transmitter terhalang benda) maka modul penerima tersebut akan berada pada logika '1' (sekitar 5V).

Dalam rangkaian terdapat diode zener 6,2 V untuk membatasi tegangan yang masuk ke modul tersebut karena sesuai datasheet modul tersebut membutuhkan daya $-0.3 - 6V$. kapasitor digunakan sebagai penstabil tegangan sedangkan LED digunakan sebagai indikator. Transistor 2N3904 digunakan sebagai inverting untuk menentukan output dari rangkaian. Keluaran dari rangkaian akan berlogika '0' (sekitar 0 V) dan LED akan menyala apabila receiver tidak menerima sinyal inframerah (terhalang oleh bayi). Dan sebaliknya keluaran dari rangkaian akan berlogika '1' (sekitar 5V) dan LED akan mati apabila receiver menerima sinyal inframerah (tidak terhalang bayi).

3.2.6 Rangkaian RS232



Gambar 3.9 Rangkaian RS232

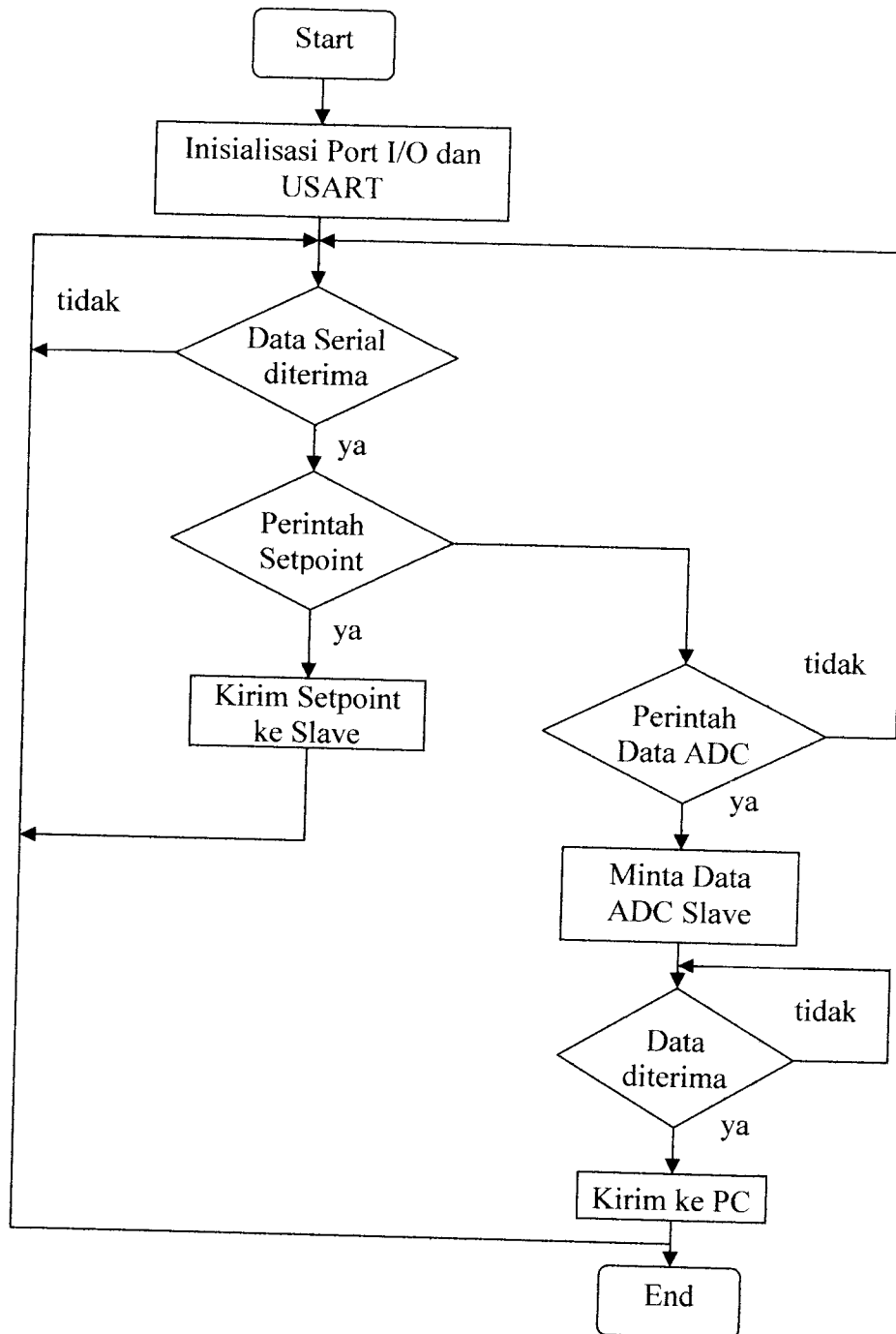
Untuk mengirimkan data dari mikrokontroler ke PC (*Personal Computer*), digunakan port serial RS-232 yang terdapat pada PC, dimana pada port ini terdapat fungsi-fungsi untuk Tx (pengiriman data), Rx (penerimaan data) dan TX/RX (pemilihan mode Tx atau Rx). Untuk melakukan transfer data dari mikrokontroler ke PC digunakan IC MAX232, yang merupakan rangkaian terpadu untuk antarmuka komunikasi serial. Selain secara hardware diperlukan pula inisialisasi secara software guna mendukung unjuk kerja komunikasi serial ini.

Secara software terdapat beberapa register yang harus diinisialisasikan, yaitu: UCSRA, UCSRB, UCSRC, UBRRH, UBRRL, dan UDR. Komunikasi dilakukan secara asinkron dengan jumlah data 8 bit *noparity*, dan menggunakan *baud rate* sebesar 19200 bps. Untuk pengiriman data digunakan fasilitas yang ada pada pengendali mikrokontroler yaitu fasilitas RXD (PD.0), TXD (PD.1) dan GND.

3.2.7 Perancangan Rangkaian ADC

ADC yang digunakan dalam perancangan ini menggunakan ADC internal mikrokontroler ATmega8535, ADC ini lebih mudah penggunaannya daripada ADC eksternal yang memerlukan rangkaian khusus. Input dari ADC ini berada pada port A0/ Pin ADC0. dimana inputnya adalah data dalam bentuk tegangan dari output LM35 yang telah melewati rangkaian pengkondisi sinyal.

Proses inisialisasi ADC meliputi proses penentuan clock, tegangan referensi, format keluaran data, dan mode pembacaan. Register yang perlu di set nilainya adalah ADMUX, ADCSRA, ADCL, ADCH, dan SFIOR. ADMUX



Gambar 3.11 Flowchart Master

BAB IV

ANALISIS DAN PEMBAHASAN

Setelah tahap perancangan dilakukan, tahap berikutnya adalah pengujian dan analisis terhadap tiap-tiap bagian pendukung sistem sebelum dilakukan pengujian terhadap sistem secara keseluruhan. Pengujian dilakukan untuk mengetahui apakah perangkat keras yang telah dirancang dapat bekerja atau berfungsi dengan baik sebagaimana yang diinginkan. Pengujian yang dilakukan terhadap perangkat keras meliputi beberapa blok rangkaian perangkat keras yang telah dirancang dan juga pengujian terhadap gabungan dari beberapa blok rangkaian. Pengujiannya meliputi :

1. Pengujian rangkaian catu daya
2. Pengujian rangkaian sensor suhu
3. Pengujian rangkaian driver lampu
4. Pengujian rangkaian pemancar inframerah
5. Pengujian rangkaian penerima inframerah
6. Pengujian ADC internal Atmega8535
7. Pengujian keseluruhan alat

4.1 Pengujian Rangkaian Catu Daya

Rangkaian catu daya berfungsi sebagai sumber tegangan dan arus catuan untuk keseluruhan rangkaian alat. Dalam desain rangkaian dapat dilihat bahwa rangkaian bekerja menggunakan tegangan 5 VDC yang dibentuk berdasarkan

tegangan keluaran penyearah yang diregulasi menggunakan IC regulator LM7805. Besar tegangan yang dihasilkan dari catu daya diperlihatkan pada tabel 4.1.

Tabel 4.1 Tegangan Keluaran Rangkaian Catu Daya

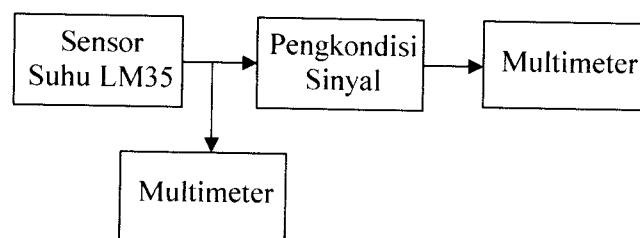
| Karakteristik | Maksimum | Minimum |
|-------------------|----------|----------|
| Tegangan Keluaran | 5,3 Volt | 4,9 Volt |

Tegangan yang dibutuhkan mikrokontroler ATmega8535 adalah antara tegangan 4,0 volt DC - 5,5 volt DC. Jadi dari hasil pengukuran tegangan keluaran catu daya aman dan bisa digunakan untuk operasional mikrokontroler. Selain itu tegangan dari catu daya ini digunakan untuk mensupply tegangan sensor.

4.2 Pengujian Rangkaian Sensor Suhu

Tujuan pengujian ini adalah untuk mengetahui apakah rangkaian sensor LM35 dapat bekerja dengan baik, dan juga untuk mengetahui besarnya tegangan output dari sensor LM35 berdasarkan perubahan temperatur dari ruangan inkubator. Langkah-langkah pengujiannya antara lain :

1. Merangkai rangkaian seperti pada gambar blok dibawah ini:



Gambar 4.1 Diagram Blok Pengujian Sensor Suhu

2. Menghubungkan kaki IC LM35 dengan tegangan DC +5 Volt pada kaki untuk Vcc, kaki ground dengan ground, dan kaki Vout ke Multimeter.
3. Meletakkan sensor LM35 dan termometer pada kotak inkubator.
4. Melakukan pengukuran pada kaki output IC LM 35 dan output pengkondisi sinyal dari suhu 30 °C sampai suhu 40 °C.

Dari pengukuran dan pengamatan apakah sensor suhu bekerja sesuai dengan input atau perubahan temperatur ruangan inkubator. Untuk hasil perhitungan dari sensor suhu dapat dihitung sebagai berikut:

$$V_{LM35} = \text{Temp} \times 10\text{mV} \quad (4.1)$$

Dengan :

V_{LM35} : Tegangan output dari sensor LM35 (Volt)

Temp : Besarnya suhu yang dibaca sensor

10mV : ketetapan tegangan setiap kenaikan 1°C

Berikut merupakan tabel perbandingan tegangan output hasil perhitungan dengan hasil pengukuran output sensor, baik dengan pengkondisi sinyal maupun langsung tanpa pengkondisi sinyal.

Tabel 4.2 Perbandingan Tegangan Hasil Perhitungan Sensor dengan Hasil Pengukuran Output Sensor LM35

| Pengujian ke- | Suhu Termometer (°C) | Hasil Perhitungan (mV) | Pengukuran Tegangan Output Sensor (mV) | |
|---------------|----------------------|------------------------|--|---------------------------|
| | | | Tanpa Pengkondisi sinyal | Dengan Pengkondisi sinyal |
| 1 | 31 | 310 | 311 | 312 |
| 2 | 32 | 320 | 321 | 323 |
| 3 | 33 | 330 | 333 | 333 |
| 4 | 34 | 340 | 342 | 342 |
| 5 | 35 | 350 | 353 | 352 |
| 6 | 36 | 360 | 365 | 363 |
| 7 | 37 | 370 | 373 | 372 |
| 8 | 38 | 380 | 384 | 383 |
| 9 | 39 | 390 | 395 | 394 |
| 10 | 40 | 400 | 403 | 402 |

Setiap kenaikan tegangan $10 \text{ mV} \approx 1 \text{ }^\circ\text{C}$, maka dari hasil pengukuran sensor suhu LM35 dapat dihitung:

$$\text{Pengukuran Suhu}(LM35) = \frac{V_{out \text{ pengukuran}}}{10 \text{ mV}} \quad (4.2)$$

Dengan : $V_{out \text{ pengukuran}}$: Tegangan output sensor hasil pengukuran (Volt)

Berikut merupakan tabel perbandingan antara pengukuran suhu menggunakan termometer dibandingkan dengan pengukuran suhu menggunakan sensor LM35.

Tabel 4.3 Perbandingan antara Pengukuran Suhu Menggunakan Termometer dengan Sensor LM35

| Pengujian ke- | Pengukuran Suhu (°C) | |
|---------------|----------------------|-------------|
| | Thermometer | Sensor LM35 |
| 1 | 31 | 31,2 |
| 2 | 32 | 32,3 |
| 3 | 33 | 33,3 |
| 4 | 34 | 34,2 |
| 5 | 35 | 35,2 |
| 6 | 36 | 36,3 |
| 7 | 37 | 37,2 |
| 8 | 38 | 38,3 |
| 9 | 39 | 39,4 |
| 10 | 40 | 40,2 |

Dari hasil pengujian sensor suhu menggunakan termometer dan menggunakan LM35 dengan 10 data pengujian, didapat grafik hubungan antara N (jumlah pengujian) dan suhu (°C) sebagai berikut :

masukan logika pada rangkaian driver. Hasil pengujiannya dapat dilihat pada tabel dibawah ini.

Tabel 4.5 Pengujian Driver Lampu

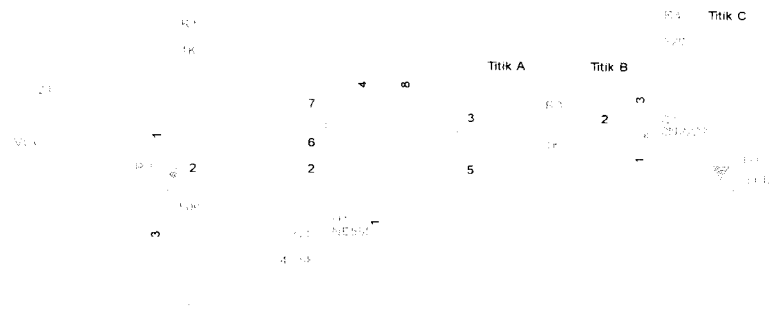
| Logika | IC MOC 3041 | Triac | Lampu |
|------------|-------------|-------|-------|
| 0 (0 Volt) | ON | ON | NYALA |
| 1 (5Volt) | OFF | OFF | MATI |

Ketika rangkaian driver di beri masukan logika 0 (0Volt) kaki IC MOC 3041 menjadi aktif (ON) sehingga kaki 4 dan 6 terhubung. Keluaran IC MOC 3041 pada kaki 4 memicu gate pada triac (kaki 3). Saat gate terpicu maka kaki 2 dan kaki 1 pada triac menghantar arus AC (ON) sehingga dapat mengaktifkan bola lampu.

Pengukuran tegangan rangkaian driver lampu yaitu dengan merangkai rangkaian dan melakukan pengukuran seperti pada gambar di bawah ini :

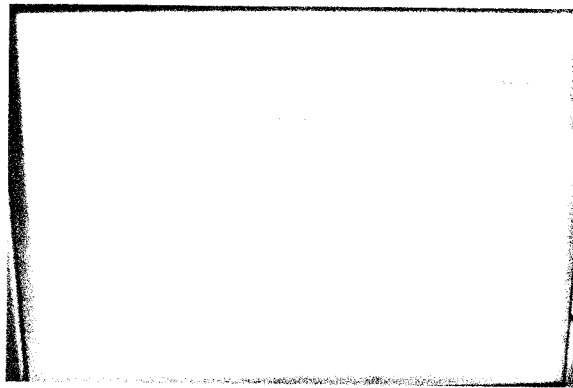


Gambar 4.3 Rangkaian Pengujian Driver AC



Gambar 4.4 Titik Pengukuran Rangkaian Pemancar Inframerah

Hasil pengujian pada titik A



Gambar 4.5 Pengujian pada Titik A

Pengamatan sinyal diatas dilakukan pada kondisi Volt/div = 2,0 Volt, Time/div = 10 uS sehingga pada gambar diatas dapat diketahui :

$$V_{pp} = \text{Volt/div} \times \text{tinggi pulsa}$$

$$= 2 \text{ Volt} \times 1,8$$

$$= 3,6 \text{ Volt}$$

$$T = \text{Time/div} \times \text{lebar pulsa}$$

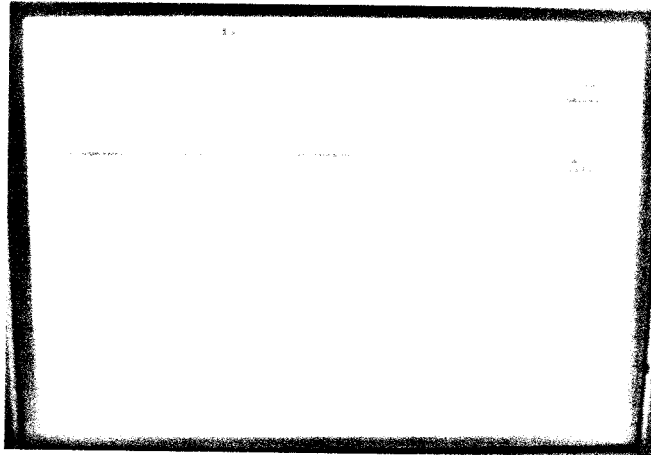
$$= 10 \text{ uS} \times 2,6 = 26 \text{ us}$$

Maka frekuensinya adalah sebesar :

$$\text{Frek} = \frac{1}{T} = \frac{1}{26\text{us}}$$

$$= 38,46 \text{ KHz}$$

Hasil pengujian pada titik B



Gambar 4.6 Pengujian pada Titik B

Pengamatan sinyal diatas dilakukan pada kondisi Volt/div = 500 mVolt, Time/div = 10 uS sehingga pada gambar diatas dapat diketahui :

$$V_{pp} = \text{Volt/div} \times \text{tinggi pulsa}$$

$$= 500 \text{ mVolt} \times 1,5$$

$$= 750 \text{ mVolt}$$

$$T = \text{Time/div} \times \text{lebar pulsa}$$

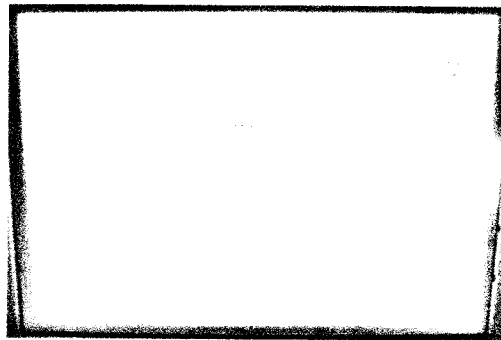
$$= 10 \text{ uS} \times 2,6 = 26 \text{ us}$$

Maka frekuensinya adalah sebesar :

$$\text{Frek} = \frac{1}{T} = \frac{1}{26\text{us}}$$

$$= 38,46 \text{ KHz}$$

Hasil pengujian pada titik C



Gambar 4.7 Pengujian pada Titik C

Pengamatan sinyal diatas dilakukan pada kondisi Volt/div = 1,00 Volt, Time/div = 10 uS sehingga pada gambar diatas dapat diketahui :

$$V_{pp} = \text{Volt/div} \times \text{tinggi pulsa}$$

$$= 1,00 \text{ Volt} \times 1,2$$

$$= 1,2 \text{ Volt}$$

$$T = \text{Time/div} \times \text{lebar pulsa}$$

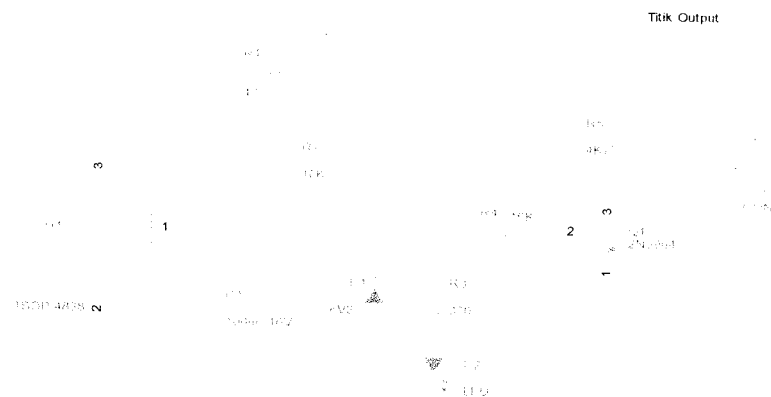
$$= 10 \text{ uS} \times 2,6 = 26 \text{ us}$$

Maka frekuensinya adalah sebesar :

$$\begin{aligned}
 \text{Frek} &= \frac{1}{T} = \frac{1}{26\mu\text{s}} \\
 &= 38,46 \text{ KHz}
 \end{aligned}$$

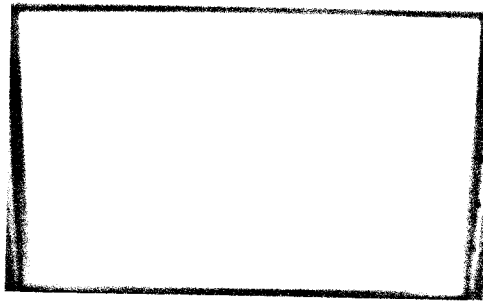
4.5 Pengujian Rangkaian Penerima Inframerah

Pengamatan rangkaian penerima inframerah dilakukan dengan mengamati sinyal keluaran dari rangkaian tersebut, yaitu ketika ada sinyal inframerah yang diterima dan ketika tidak ada sinyal inframerah.



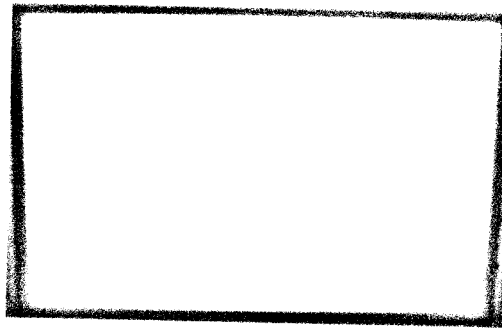
Gambar 4.8 Pengamatan Rangkaian Penerima Inframerah

Pengamatan keluaran rangkaian penerima inframerah ketika tidak ada benda yang menghalangi sinyal inframerah dari pemancar tersebut. terlihat bentuk gelombang sinyalnya yaitu berupa garis lurus dengan logika '1' (5 Volt). Gambar pengamatan sinyalnya adalah :



Gambar 4.9 Sinyal Penerima Inframerah Ketika
Tidak ada Penghalang

Pengamatan keluaran ranmerah dari pemancar tersebut, terlihat bentuk gelombang sinyalnya yaitu berupa garis lurus dengan logika '0' (0 Volt). Gambar pengamatan sinyalnya adalah :

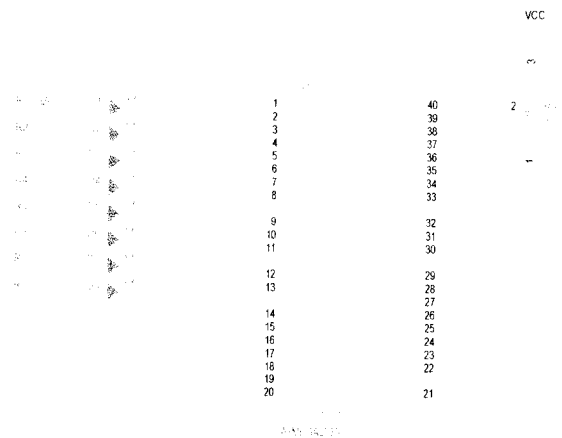


Gambar 4.10 Sinyal Penerima Inframerah ketika
Ada Penghalang

4.6 Pengujian ADC internal ATmega8535

Tujuan dari pengujian ADC internal yaitu untuk mengetahui respon dari ADC yang mana masukannya berasal dari tegangan analog sedangkan keluarannya adalah berupa sinyal digital 8 bit yang di hubungkan ke LED.

Rangkaian pengujian ADC internal adalah sebagai berikut:



Gambar 4.11 Rangkaian Pengujian ADC Internal

Untuk perhitungan dari output ADC dapat dicari dengan menggunakan rumus sebagai berikut :

$$\text{Resolusi} : \left(\frac{1}{2^n}\right)V_{ref} \quad (4.4)$$

Dalam perancangan, banyaknya bit ADC yang di gunakan adalah 8 bit dan tegangan referensi yang digunakan adalah 2,56 Volt sehingga resolusinya :

$$\text{Resolusi} : \left(\frac{1}{2^8}\right) \times 2,56$$

$$: \left(\frac{1}{256}\right) \times 2,56$$

$$: 0,01$$

Keluaran ADC dapat dicari dengan menggunakan rumus berikut:

$$\text{NilaiADC} : \frac{V_{in}}{\text{Resolusi}} \quad (4.5)$$

Berikut merupakan tabel perbandingan tegangan output antara hasil perhitungan dan hasil pengukuran dari keluaran ADC Internal ATmega8535 :

Tabel 4.7 Pengujian ADC Internal ATmega8535

| Vin (Volt) | Keluaran ADC Internal ATmega8535 | | | | | |
|---------------|----------------------------------|---------|-------------|-------------------|---------|-------------|
| | Perhitungan | | | Pengukuran | | |
| | Biner (D7..D0) | Desimal | Hexadesimal | Biner (D0..D0) | Desimal | Hexadesimal |
| 0 | 00000000 | 0 | 0 | 00000000 | 0 | 0 |
| 0,25 | 00011001 | 25 | 19 | 00011000 | 24 | 18 |
| 0,50 | 00110010 | 50 | 32 | 00110000 | 48 | 30 |
| 0,75 | 01001011 | 75 | 4B | 01001100 | 76 | 4C |
| 1,00 | 01100100 | 100 | 64 | 01100111 | 103 | 67 |
| 1,25 | 01111101 | 125 | 7D | 10000000 | 128 | 80 |
| 1,50 | 10010110 | 150 | 96 | 10011000 | 152 | 98 |
| 1,75 | 10101111 | 175 | AF | 10110000 | 176 | B0 |
| 2,00 | 11001000 | 200 | C8 | 11001100 | 204 | CC |
| 2,25 | 11100001 | 225 | E1 | 11100011 | 227 | E3 |
| 2,50 | 11111010 | 250 | FA | 11111111 | 255 | FF |

Dari tabel 4.7 dapat dicari selisih antara hasil perhitungan dengan hasil pengukuran keluaran ADC. Selisih dapat dicari dengan rumus sebagai berikut:

$$\text{Selisih} = |\text{Pengukuran} - \text{Perhitungan}| \quad (4.6)$$

Sedangkan persentase kesalahan (%error) dapat dicari dengan menggunakan persamaan sebagai berikut:

$$\text{Error rata-rata} = \frac{\text{JumlahError}}{\text{JumlahPerhitungan}} \times 100\% \quad (4.7)$$

Dari rumus diatas dapat diketahui selisih antara hasil perhitungan dengan hasil pengukuran keluaran ADC ditunjukkan pada tabel berikut:

Tabel 4.8 Selisih Keluaran ADC Internal

| Keluaran ADC Internal ATmega8535 | | | | | | Selisih (Desimal) |
|----------------------------------|---------|-------------|-------------------|---------|-------------|----------------------|
| Perhitungan | | | Pengukuran | | | |
| Biner (D7..D0) | Desimal | Hexadesimal | Biner (D0..D0) | Desimal | Hexadesimal | |
| 00000000 | 0 | 0 | 00000000 | 0 | 0 | 0 |
| 00011001 | 25 | 19 | 00011000 | 24 | 18 | 1 |
| 00110010 | 50 | 32 | 00110000 | 48 | 30 | 2 |
| 01001011 | 75 | 4B | 01001100 | 76 | 4C | 1 |
| 01100100 | 100 | 64 | 01100111 | 103 | 67 | 3 |
| 01111101 | 125 | 7D | 10000000 | 128 | 80 | 3 |
| 10010110 | 150 | 96 | 10011000 | 152 | 98 | 2 |
| 10101111 | 175 | AF | 10110000 | 176 | B0 | 1 |
| 11001000 | 200 | C8 | 11001100 | 204 | CC | 4 |
| 11100001 | 225 | E1 | 11100011 | 227 | E3 | 2 |
| 11111010 | 250 | FA | 11111111 | 255 | FF | 5 |

Dari tabel diatas dapat dicari persentase kesalahan (%error) dengan menggunakan persamaan sebagai berikut:

$$\text{Error rata-rata} = \frac{\text{JumlahError}}{\text{JumlahPerhitungan}} \times 100\%$$

Jika dari tabel diketahui:

$$\sum \text{Perhitungan (desimal)} : 1375$$

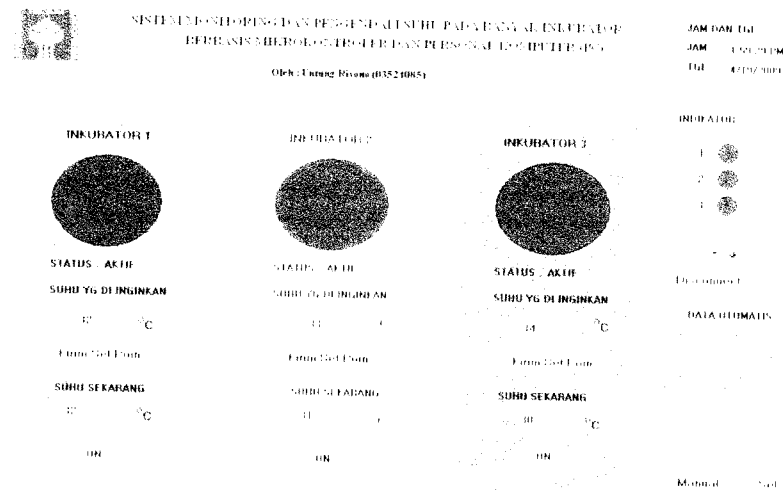
$$\sum \text{Error (desimal)} : 24$$

Sehingga kesalahan rata-ratanya:

$$\begin{aligned} \text{Error rata-rata} &= \frac{\text{JumlahError}}{\text{JumlahPerhitungan}} \times 100\% \\ &= \frac{24}{1375} \times 100\% \\ &= 1,745 \% \end{aligned}$$

4.7 Pengujian keseluruhan sistem

Pengujian sistem secara keseluruhan dilakukan untuk mengamati kondisi suhu secara *realtime* dari tiga buah inkubator, Mengamati tanggapan sistem terhadap *setpoint* yang telah diberikan, dan mengamati tanggapan sistem untuk mendeteksi keberadaan bayi dalam inkubator tersebut. Penampil dari sistem tersebut dapat dilihat pada gambar berikut.



Gambar 4.12 Penampil Sistem Monitoring dan Pengendalian Suhu dari banyak Inkubator

Sistem dapat menampilkan suhu baik secara keseluruhan maupun satu per satu dari inkubator tersebut. Untuk menampilkan suhu secara keseluruhan yaitu dengan menggunakan tombol "otomatis", sehingga suhu yang di tampilkan dapat *up to date*. Sedangkan untuk menampilkan suhu dari tiap-tiap inkubator dengan menggunakan tombol "manual". Untuk mendeteksi keberadaan bayi pemanas akan OFF jika tidak ada bayi dan akan ON ketika ada bayi.

Dalam sistem pengendalian pengujiannya dilakukan dengan memasukkan *setpoint* untuk menentukan besarnya suhu yang di inginkan dari masing-masing inkubator tersebut. Pengendalian dalam sistem ini yaitu pengendalian ON/OFF, sehingga pemanas (lampu) akan OFF apabila suhu inkubator telah mencapai "setpoint + 1". Demikian pula sebaliknya pemanas (lampu) akan ON apabila

suhu inkubator telah mencapai “setpoint -1”. Untuk memasukan setpoint tampilan suhu harus secara “manual” hal ini untuk menghindari terjadinya error.

Perancangan sistem monitoring dan pengendalian suhu dari banyak inkubator ini sudah sesuai dengan yang diinginkan. Dari parameter-parameter yang diuji dan diukur pada alat diatas dapat disimpulkan bahwa sistem ini telah bekerja dengan optimal meskipun masih terjadi error. Hal ini dibuktikan dengan hasil pengujian dan pengukuran diatas. Dari hasil pengujian dan pengukuran ini, perancangan telah memenuhi ide awal atau perencanaan.

BAB V

KESIMPULAN

Berdasarkan dari proses perancangan, perakitan, pengamatan dan hasil uji sistem yang dibuat, maka dapat diambil beberapa kesimpulan dan saran untuk kemajuan, perbaikan dan pengembangan dari aplikasi sistem.

5.1 Kesimpulan

Berdasarkan studi dan penelitian yang dilakukan, maka dapat disimpulkan beberapa hal antara lain :

1. Penggunaan sensor LM35 sebagai media pendeteksi suhu sangat efektif dan efisien karena dapat menghasilkan pengukuran sistem yang linear.
2. Pengukuran suhu dengan LM35 pada penelitian ini memiliki error rata-rata sebesar 0,67 %.
3. Adanya rangkaian transmitter dan receiver infrared untuk mendeteksi keberadaan bayi dalam inkubator sehingga menjadi lebih efisien dalam hal pemakaian daya listrik.
4. Dari pengujian ADC internal ATmega8535 dapat dikatakan sistem ini berfungsi dengan baik, terbukti dengan didapatkan error rata-rata antara perhitungan dan pengukuran sebesar 1,745 %.
5. Dalam pengujian setpoint sistem ini telah berfungsi dengan baik tetapi masih terdapat error, terbukti pemanas akan OFF ketika suhu inkubator telah mencapai “setpoint +1” dan akan ON ketika suhu inkubator telah mencapai “setpoint – 1”.

5.2 Saran

Beberapa saran yang dapat dipertimbangkan untuk pengembangan penelitian selanjutnya antara lain :

1. Untuk pengembangan selanjutnya sebaiknya digunakan sensor yang lebih baik dalam membaca suhu seperti termokopel yang mempunyai keluaran lebih linear dibandingkan dengan sensor LM35.
2. Untuk pengembangan selanjutnya sebaiknya software pada PC dilengkapi dengan database pasien dan data suhu bisa disimpan.
3. Untuk pengembangan selanjutnya sebaiknya dapat ditambahkan kamera sehingga dapat memonitoring gambar dari inkubator secara keseluruhan.

```

.....
This program was produced by the
CodewizardAVR V1.24.5d Professional
Automatic Program Generator
© Copyright 1998-2006 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hp.infotech.com

```

```

Project :
Version :
Date : 11/06/2008
Author : F40G
Company : F40G
Comments:

```

```

.....
Chip type : ATmega8535
Program type : Application
Clock frequency : 11.0592MHz
Memory model : Small
External SRAM size : 0
Data Stack size : 128
.....

```

```

    <mega8535.h>
    <delay.h>
    <stdio.h>

    RXB8 1
    TXB8 0
    UPE 2
    OVR 3
    FE 4
    UDRE 5
    RXC 7

    FRAMING_ERROR (1<<FE)
    PARITY_ERROR (1<<UPE)
    DATA_OVERRUN (1<<OVR)
    DATA_REGISTER_EMPTY (1<<UDRE)
    RX_COMPLETE (1<<RXC)
char data_rx[5], i, puluhan, ratusan, satuan, chanel;

unsigned char SETPOINT1;
char Parse_Perintah(void);
void Eksekusi_Perintah(void);

void kirim( unsigned char data )
{
    while (!(UCSRA & 0x20));
    UDR = data;
}
// USART Receiver buffer
    RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];
    RX_BUFFER_SIZE<256
unsigned char rx_wr_index, rx_rd_index, rx_counter;

unsigned int rx_wr_index, rx_rd_index, rx_counter;

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
    char status, data;
    status=UCSRA;
    data=UDR;
    if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
    {
        rx_buffer[rx_wr_index]=data;
        // Kirim(data);
        i++;
        if (i<=4){
            data_rx[i]=data;
        }
        if (i==4)
        { i=0;
            Eksekusi_Perintah(); }

        if (++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
        if (++rx_counter == RX_BUFFER_SIZE)

```

```

    {
        rx_counter=0;
        rx_buffer_overflow=1;
    };
};
}

    _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
    _ALTERNATE_GETCHAR_
used+
char getchar(void)
{
    char data;
    while (rx_counter==0);
    data=rx_buffer[rx_rd_index];
    if (++rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
    (" ")
--rx_counter;
    (" ")
return data;
}
    used-

void konversi(unsigned int nilai){
    ratusan = (nilai%1000)/100;
    puluhan=(nilai%100)/10;
    satuan=(nilai%10);
    if (chanel == 1)
    { kirim("1"); }
    else if (chanel==2)
    { kirim("2"); }
    else if (chanel==3)
    { kirim("3"); }
    kirim(ratusan+0x30);
    kirim(puluhan+0x30);
    kirim(satuan+0x30);
    kirim(0);
}

void Tulis_SetPoint(unsigned char k, unsigned char j){
    delay_ms(10);
    PORTC=0xff;
    if (k==1){
        PORTC.0=0;
        PORTC.1=1;
        PORTC.2=1;
        PORTC.3=1;
        PORTC.4=1;
        PORTC.5=1;
        // kirim('1');
    }
    else if
    (k==2){
        PORTC.0=1;
        PORTC.1=1;
        PORTC.2=0;
        PORTC.3=1;
        PORTC.4=1;
        PORTC.5=1;
        // kirim('2');
    }
    else if
    (k==3){
        PORTC.0=1;
        PORTC.1=1;
        PORTC.2=1;
        PORTC.3=1;
        PORTC.4=0;
        PORTC.5=1;
        // kirim('3');
    }
    DDRA=0xff;
    delay_ms(10);
    PORTA=j;
    delay_ms(100);
    PORTC=0xff;
    delay_ms(10);
    DDRA=0x00;
    PORTA=0x00;
}

char Baca_Suhu(unsigned char k){

```



```

delay_ms(10);
DDRA=0x00;
PORTC=0xff;
if (k==1){
    PORTC.0=1;
    PORTC.1=0;
}
else if
    (k==2){
    PORTC.2=1;
    PORTC.3=0;
}
else if
    (k==3){
    PORTC.4=1;
    PORTC.5=0;
}
delay_ms(100);
//delay_ms(100);
return PINA;
}

// Declare your global variables here

void main(void)
{
    i=0;
    PORTA=0x00;
    DDRA=0x00;
    PORTB=0xff;
    DDRB=0xff;

    DDRC=0xFF;
    PORTC=0xFF;
    PORTD=0x00;
    DDRD=0x00;
    // USART initialization
    // Communication Parameters: 8 Data, 1 Stop, No Parity
    // USART Receiver: On
    // USART Transmitter: On
    // USART Mode: Asynchronous
    // USART Baud rate: 9600
    UCSRA=0x00;
    UCSRB=0x98;
    UCSRC=0x86;
    UBRRH=0x00;
    UBRRL=0x47;
    ACSR=0x80;
    SFIOR=0x00;
    ("...")

    kirim("1");
    kirim("2");
    kirim("3");

while (1)
    {
    // kirim(1+0x30);
    // kirim(2+0x30);
    // kirim(3+0x30);
    // kirim(0);
    // delay_ms(1000);

    };
}

void Eksekusi_Perintah(void){ // Instruksi PC yang harus dilakukan master
unsigned char k,s;
    k=Parse_Perintah();
    if (k==1)
        { SETPOINT1=data_rx[4];
        Tulis_SetPoint(1,SETPOINT1);
        }

    else if (k==2)
        {SETPOINT1=data_rx[4];
        Tulis_SetPoint(2,SETPOINT1);
        }

    else if (k==3)
        {SETPOINT1=data_rx[4];
        Tulis_SetPoint(3,SETPOINT1);
        }

    else if (k==4)
        {s=Baca_Suhu(1);
        chanel = 1;

```

```
        konversi(s);
    }
    else if (k==5)
    {s=Baca_Suhu(2);
    chanel = 2;
    konversi(s);
    }
    else if (k==6)
    {s=Baca_Suhu(3);
    chanel = 3;
    konversi(s);
    }
}

char Parse_Perintah(){ //terjemahkan perintah
if ((data_rx[1]=='1')&&(data_rx[2]=='1')&&(data_rx[3]=='1'))
    return 1;
else if ((data_rx[1]=='1')&&(data_rx[2]=='1')&&(data_rx[3]=='2'))
    return 2;
else if ((data_rx[1]=='1')&&(data_rx[2]=='1')&&(data_rx[3]=='3'))
    return 3;
else if ((data_rx[1]=='1')&&(data_rx[2]=='2')&&(data_rx[3]=='1'))
    return 4;
else if ((data_rx[1]=='1')&&(data_rx[2]=='2')&&(data_rx[3]=='2'))
    return 5;
else if ((data_rx[1]=='1')&&(data_rx[2]=='2')&&(data_rx[3]=='3'))
    return 6;
}
```

```

DDRB=0x00;
InitADC();
do{
    PORTB = read_adc();
    if (!(PIND.2)&(PIND.3)){
        DDRB=0x00;
        PORTB=0x00;

        delay_ms(20);
        SET_POINT=PINB;
        PORTC=SET_POINT;
    } //SET POINT
    else if (!(PIND.3)&(PIND.2)){
        nilai=read_adc();
        DDRB=0xff;
        PORTB=nilai;
        delay_ms(100);
        DDRB=0x00;
    }

    // PORTB=read_adc();
    if(!PINC.0){
        // DDRB=0xff;
        //delay ms(10);
        nilai=read_adc();
        //PORTC=nilai;
        PORTD.4=0;
        if(nilai>=SET_POINT+1) {
            PORTC.1=1; }//matikan lampu }
        else if (nilai<=SET_POINT-1) {
            PORTC.1=0;} //hidupkan_lampu; }
        }
    else{
        PORTD.4=1;
        PORTC.1=1;}//matikan lampu }
        delay_ms(100);
        // PORTC.1=0;
    }while (1);
}

```

coba3;

face

dows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
logs, StdCtrls, ExtCtrls, jpeg, CPortCtl, CPort, strutils;

```
rm1 = class(TForm)
anel1: TPanel;
anel2: TPanel;
abel1: TLabel;
abel2: TLabel;
abel3: TLabel;
'imer1: TTimer;
'anel3: TPanel;
'anel4: TPanel;
'anel5: TPanel;
Label4: TLabel;
Shape1: TShape;
Shape2: TShape;
Shape3: TShape;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Edit1: TEdit;
Label10: TLabel;
Edit3: TEdit;
Button1: TButton;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Edit4: TEdit;
Label17: TLabel;
Edit6: TEdit;
Button3: TButton;
Label18: TLabel;
Label19: TLabel;
Label20: TLabel;
Label21: TLabel;
Edit7: TEdit;
Label24: TLabel;
Edit9: TEdit;
Button5: TButton;
Label25: TLabel;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
Label29: TLabel;
Label30: TLabel;
Label31: TLabel;
Label32: TLabel;
Label35: TLabel;
Label36: TLabel;
Label39: TLabel;
Label40: TLabel;
Timer2: TTimer;
Panel7: TPanel;
Image1: TImage;
GroupBox1: TGroupBox;
Label45: TLabel;
Label46: TLabel;
Label47: TLabel;
Shape4: TShape;
Shape5: TShape;
Shape6: TShape;
GroupBox3: TGroupBox;
Label48: TLabel;
Label49: TLabel;
Edit12: TEdit;
Edit13: TEdit;
ComComboBox1: TComComboBox;
```

```

ComPort1: TComPort;
Button7: TButton;
ComLed1: TComLed;
Level1: TBevel;
Memo1: TMemo;
Button8: TButton;
Button9: TButton;
Button10: TButton;
Button11: TButton;
Button2: TButton;
Label8: TLabel;
Label9: TLabel;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure ComPort1RxChar(Sender: TObject; Count: Integer);
procedure Button9Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;

```

```

Form1: TForm1;
Panel: integer;

```

Implementation

```

*.dfm}

```

```

procedure TForm1.Button1Click(Sender: TObject);
var
s : string;
begin
s := 'ad11';
ComPort1.WriteString(s);
end;

```

```

procedure TForm1.Button2Click(Sender: TObject);
begin
if (button2.Caption = 'Otomatis') then
begin
button2.Enabled := true;
button2.Caption := 'Manual';
Label9.Caption := 'DATA OTOMATIS';
end;
end;

```

```

else
begin
button2.Enabled := False;
button2.Caption := 'Otomatis';
Label9.Caption := 'DATA MANUAL';
end;
end;

```

```

;

procedure TForm1.Button3Click(Sender: TObject);
var
s : string;
begin
s := 'ad21';
ComPort1.WriteString(s);
end;

```

```

procedure TForm1.Button5Click(Sender: TObject);
var
s : string;
begin

```

```
'ad31';  
rt1.WriteString(s);
```

```
procedure TForm1.Timer1Timer(Sender: TObject);  
begin  
  Label2.Text := TimeToStr (Now);  
  Label3.Text := DateToStr (Now);
```

```
procedure TForm1.Button7Click(Sender: TObject);  
begin  
  ComPort1.Connected then  
begin  
  ComPort1.Close;  
  shape1.Brush.Color := clred;  
  shape2.Brush.Color := clred;  
  shape3.Brush.Color := clred;  
  label6.Caption := 'NON AKTIF';  
  label13.Caption := 'NON AKTIF';  
  label20.Caption := 'NON AKTIF';  
  button7.Caption := '&Connect';  
end  
se  
gin  
  ComPort1.Open;  
  shape1.Brush.Color := clBlue;  
  shape2.Brush.Color := clBlue;  
  shape3.Brush.Color := clBlue;  
  label6.Caption := 'AKTIF';  
  label13.Caption := 'AKTIF';  
  label20.Caption := 'AKTIF';  
  button7.Caption := '&Disconnect';  
end;
```

```
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
```

```
var: String;  
  nunjuk, nilai : integer;  
  adc, data : ansistring;  
in  
  ComPort1.ReadStr(Str, Count);  
  memol.Text := Memol.Text + Str;  
  adc := ansimidstr(str,1,1);  
  data := ansimidstr(str,2,3);  
  nilai := strtoint(data);  
  if ( adc = '1') then  
    dit3.Text := inttostr(nilai)  
  else if ( adc = '2') then  
    dit6.Text := inttostr(nilai)  
  else if ( adc = '3') then  
    dit9.Text := inttostr(nilai);  
  
  if nilai < 30 then  
begin  
  shape6.Brush.Color := clgreen;  
end  
  else if ((nilai >= 30) and (nilai <35)) then  
begin  
  shape6.Brush.Color := clyellow;  
end  
  else  
begin  
  shape6.Brush.Color := clred;  
end;  
end;
```

```
procedure TForm1.Button8Click(Sender: TObject);  
begin  
  ComPort1.ShowSetupDialog;
```

```
procedure TForm1.Button9Click(Sender: TObject);
var s: string;
```

```
Memor1.Text := Memor1.Text + 'aku';
s := 'sp1'+chr(strtoint(edit1.Text));
Label1.WriteString(s);
Label2.WriteString(chr(strtoint(edit1.Text)));
```

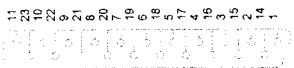
```
procedure TForm1.Button10Click(Sender: TObject);
var s: string;
begin
s := 'sp2'+chr(strtoint(edit4.Text));
Label1.WriteString(s);
```

```
procedure TForm1.Button11Click(Sender: TObject);
var s: string;
begin
s := 'sp3'+chr(strtoint(edit7.Text));
Label1.WriteString(s);
```

```
procedure TForm1.Timer2Timer(Sender: TObject);
begin
if (chanel >2) then
begin
chanel := 0;
Label4.Caption := inttostr(chanel);
```

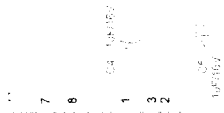
```
if (chanel=0) then
Label1.Click;
if (chanel =1) then
Label3.Click;
if (chanel =2) then
Label5.Click;
chanel := chanel + 1;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
chanel:=0;
end;
```

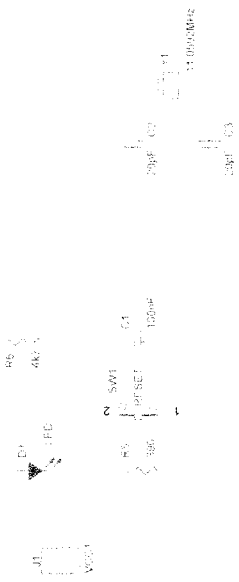
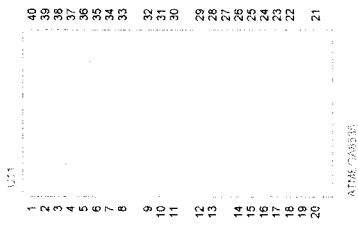


CB29
BGA160

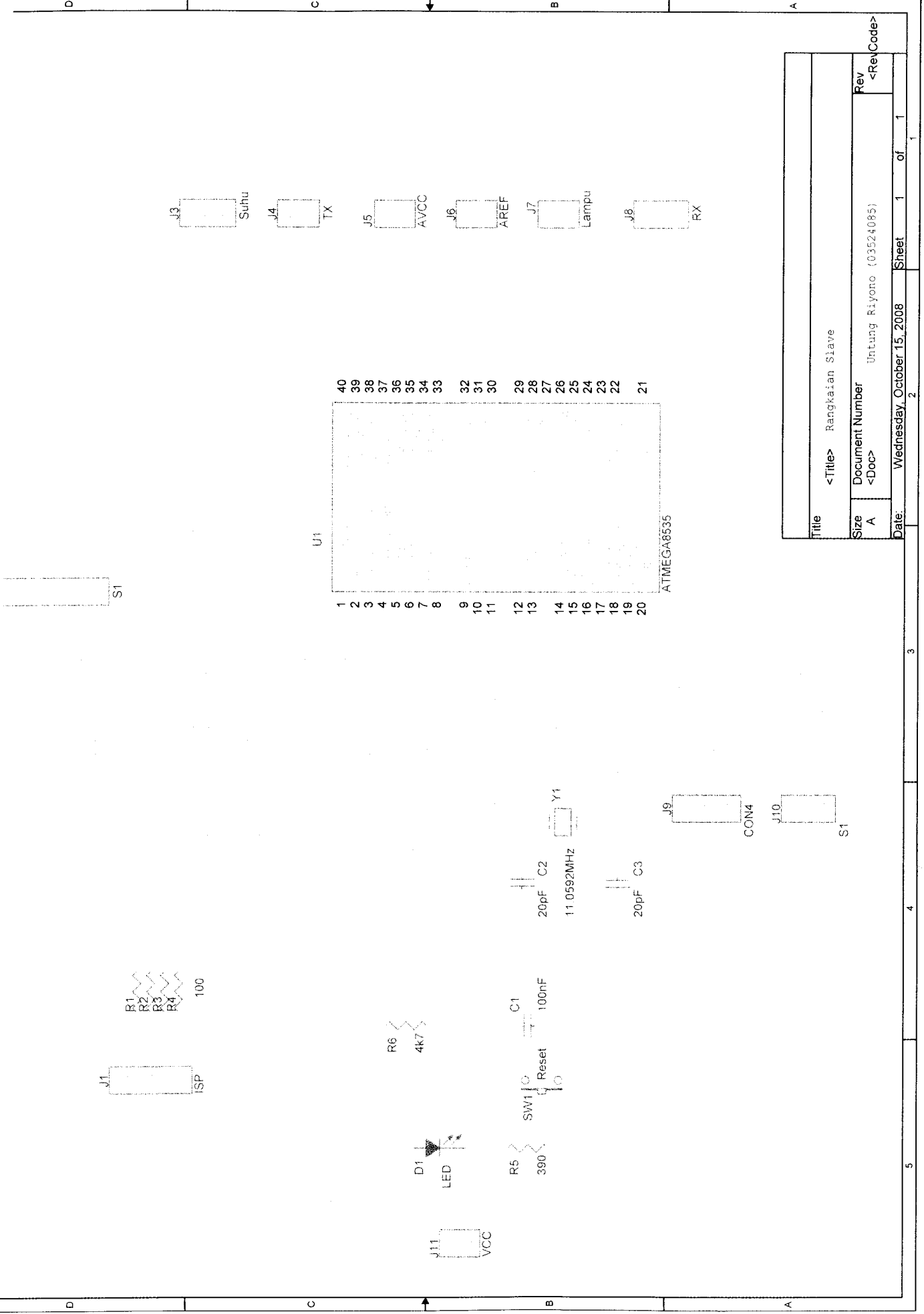
MAX232



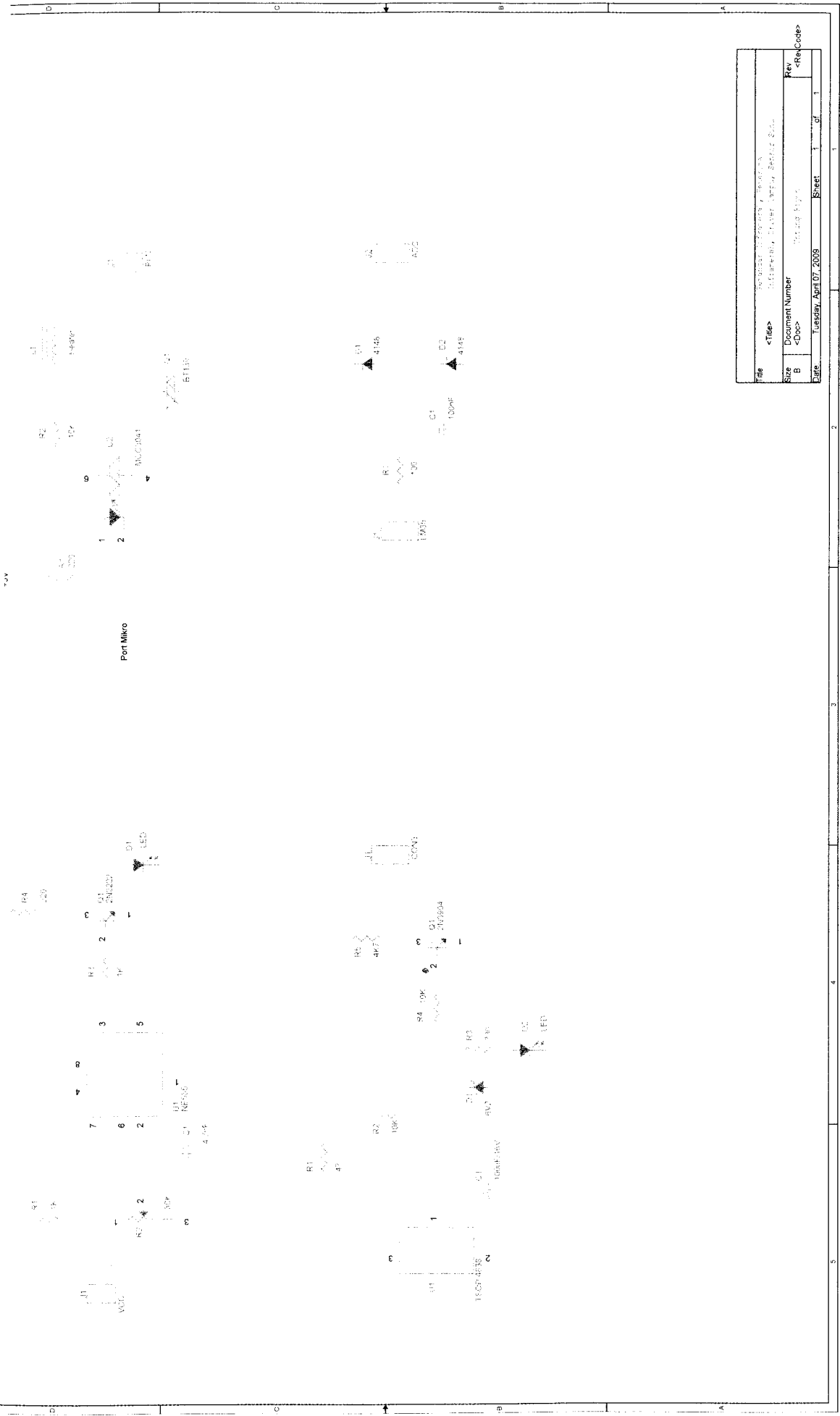
Serial
Send



| | | |
|-----------------|--------------|-----|
| Title | ATME64B33F | |
| Size | B | Doc |
| Document Number | 64B33F | Rev |
| Date | Sheet 1 of 1 | Rev |



| | | | |
|-------|-----------------------------|-------------------------|--------------------------|
| Title | | <Title> Rangkaian Slave | |
| Size | A | Document Number | <Doc> |
| Rev | <RevCode> | Author | Untung Riyono (03524085) |
| Date: | Wednesday, October 15, 2008 | Sheet | 1 of 1 |



| | | | |
|--------|--------------------------------------|-----------------|------------|
| Title | Mikrocontroller Board / Revision 1.0 | | |
| Author | John Doe | | |
| Size | 8 | Document Number | 100-1000 |
| <Doc> | | Rev | <Rev Code> |
| Date | Tuesday, April 07, 2009 | Sheet | 1 of 1 |