



UNIVERSITAS INDONESIA

**DESAIN SISTEM MONITORING SUHU
MENGUNAKAN SENSOR DS 18B20**

SKRIPSI

**AINUR ROFIQ
0706199022**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
DESEMBER 2010**



UNIVERSITAS INDONESIA

**DESAIN SISTEM MONITORING SUHU
MENGUNAKAN SENSOR DS 18B20**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Teknik**

**AINUR ROFIQ
0706199022**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
DESEMBER 2010**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar**

Nama : AINUR ROFIQ

NPM : 0706199022

Tanda Tangan :

Tanggal : 29 Desember 2010

HALAMAN PENGESAHAN

Skrisi ini diajukan oleh

Nama : Ainur Rofiq
NPM : 0706199022
Program Studi : Teknik Elektro
Judul Skripsi : Desain sistem monitoring suhu menggunakan sensor DS 18B20


Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

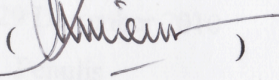
Pembimbing : Aji Nur Widyanto, ST, MT

()

Penguji : Prof. Dr. Ir. Iwa Garniwa, MK, MT

()

Penguji : Ir. Amien Rahardjo, MT

()

Ditetapkan di : Depok

Tanggal : 29 Desember 2010

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Allah SWT, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari tanpa bantuan dan bimbingan dari berbagai pihak dari masa perkuliahan sampai pada penyusunan skripsi ini, saya tidak akan mampu menyelesaikannya. Untuk itu saya mengucapkan terima kasih kepada:

- (1) Aji Nur Widyanto, ST, MT, selaku dosen pembimbing yang telah meluangkan waktu, tenaga dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini.
- (2) Orang tua tercinta, adik-adik, Istri, sahabat dan keluarga besar yang telah memberikan bantuan dukungan material dan moral.
- (3) Anggery laqaiza sofyana atas dukungan material dan moral yang tidak terhingga.
- (4) Teman-teman yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhirnya, saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi semua.

Depok, 29 Desember 2010

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

:

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Ainur Rofiq
NPM : 0706199022
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

Desain Sistem Monitoring Suhu Menggunakan Sensor DS 18B20

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 29 Desember 2010

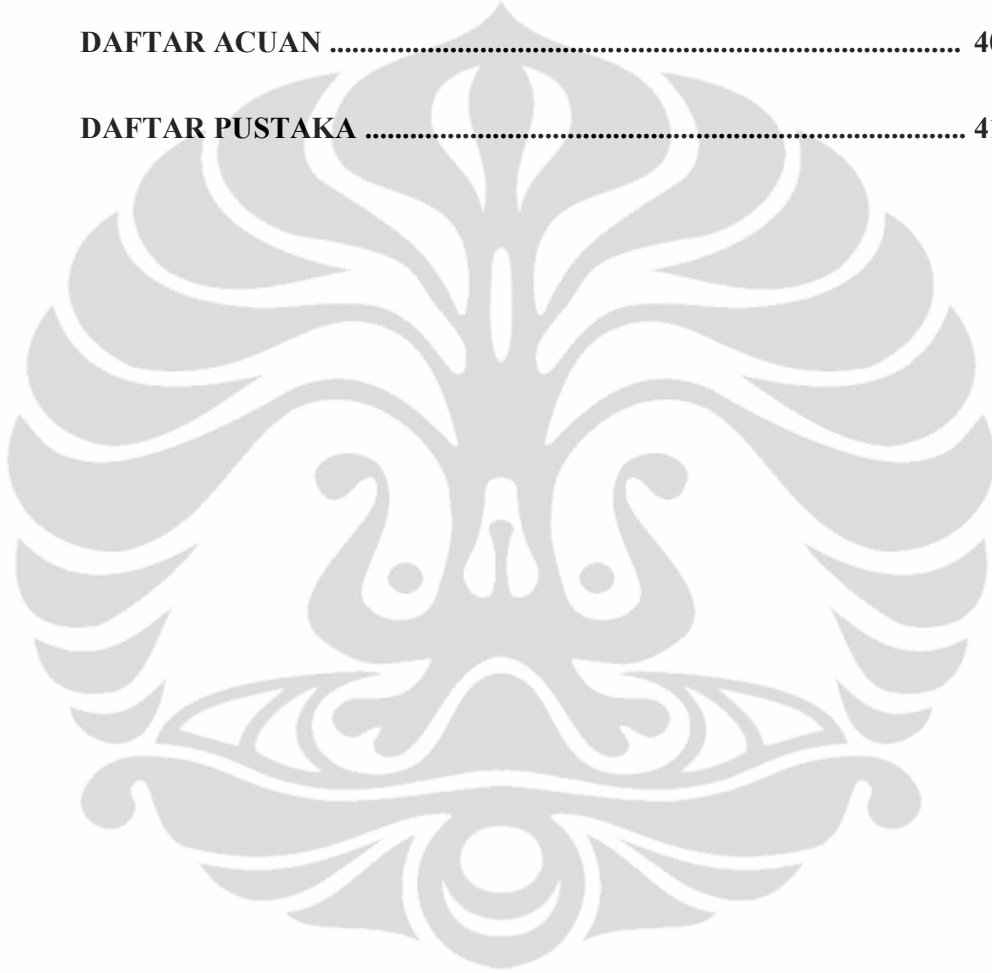
Yang menyatakan

(Ainur Rofiq)

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERNYATAAN ORISINALITAS	ii
HALAMAN PENGESAHAN	iii
UCAPAN TERIMA KASIH	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	x
DAFTAR TABEL	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	1
1.3 Batasan Masalah	1
1.4 Metode Pembahasan	1
1.5 Sistematika Penulisan	2
BAB 2 DASAR TEORI	3
2.1 Sensor Suhu DS 18S20	3
2.2 Mikrokontroler ATMEGA 8535	6
2.2.1 Arsitektur	7
2.2.2 Konfigurasi Pin	9
2.2.3 Sistem Clock	11
2.2.4 Memori	12
2.2.5 Port Sebagai Input/Output Digital	14
2.2.6 <i>Interupt</i>	15
2.2.7 Timer/Counter	17
BAB 3 PERANCANGAN SISTEM	24
3.1 Tujuan Perancangan	24
3.2 Fungsi Alat	24
3.3 Spesifikasi Alat	24
3.4 Langkah-langkah Perancangan	25
3.4.1 Perancangan Perangkat Keras	25
3.4.2 Perancangan Perangkat Lunak	25
3.5 Realisasi	25
3.5.1 Realisasi Perangkat Keras	25
3.5.2 Realisasi Perangkat Lunak	30
BAB 4 PENGUJIAN DAN ANALISA	32
4.1 Pengujian	32
4.1.1 Koneksi Komputer (PC) dengan Mikrokontroler	32

4.1.2	Koneksi Sensor Suhu DS 18B20 dengan Mikrokontroler	34
4.2	Analisa	35
4.2.1	Hasil Pengujian Perbandingan Pengukuran Suhu	35
4.2.2	Hasil Pengujian <i>Time Response</i>	38
BAB 5 KESIMPULAN		39
DAFTAR ACUAN		40
DAFTAR PUSTAKA		41



DAFTAR GAMBAR

Gambar 2.1	Sensor suhu DS 18B20	3
Gambar 2.2	Diagram blok DS18B20	5
Gambar 2.3	Peta memori program	7
Gambar 2.4	Peta memori	8
Gambar 2.5	Konfigurasi pin	9
Gambar 2.6	Sistem <i>clock</i>	12
Gambar 2.7	Peta memori data.....	13
Gambar 2.8	Peta memori program	13
Gambar 2.9	Register MCUCR.....	15
Gambar 2.10	Register GICR	15
Gambar 2.11	Timer/counter 0	17
Gambar 2.12	Timer/counter 1	19
Gambar 2.13	Register TCCR1B	20
Gambar 2.14	Register TCCR2	21
Gambar 2.15	Register TIMSK	22
Gambar 2.16	Register TIFR	23
Gambar 3.1	Blok diagram sistem monitoring suhu	25
Gambar 3.2	Skematik power supply	26
Gambar 3.3	Skematik serial interface	27
Gambar 3.4	Skematik rangkaian sensor suhu DS 18B20	28
Gambar 3.5	Rangkaian Sistem Minimum ATmega 8535	30
Gambar 3.6	Rangkaian LCD (<i>liquid colour display</i>)16 x 2	30
Gambar 3.7	Tampilan sistem monitoring suhu	31
Gambar 4.1	Membuka hyperterminal	32
Gambar 4.2	Memasukkan nama	33
Gambar 4.3	Memilih port komunikasi	33
Gambar 4.4	<i>Setting port</i>	34
Gambar 4.5	Tampilan sensor suhu	34
Gambar 4.6	Grafik Perbandingan hasil pembacaan sensor #1 dengan thermometer digital	36
Gambar 4.7	Grafik Perbandingan hasil pembacaan sensor #2 dengan thermometer digital	37
Gambar 4.8	Perbandingan rata-rata hasil pembacaan sensor #1 & #2 dengan thermometer digital	38
Gambar 4.9	Pengujian <i>response time</i> untuk 500 ms	39
Gambar 4.10	Pengujian <i>response time</i> untuk 5 ms	40

DAFTAR TABEL

Tabel 2.1	Reset dan interrupt	16
Tabel 2.2	Table setting	17
Tabel 2.3	<i>Compare output mode, non-PWM mode</i>	18
Tabel 2.4	<i>Compare output mode, fast PWM mode</i>	18
Tabel 2.5	<i>Compare output mode, phase correct PWM mode</i>	18
Tabel 2.6	<i>Clock select bit description</i>	19
Tabel 2.7	<i>Waveform generation mode bis description</i>	20
Tabel 2.8	<i>Clock select bit description</i>	20
Tabel 2.9	<i>Waveform generation mode bit description</i>	21
Tabel 2.10	<i>Compare output mode, non-PWM mode</i>	21
Tabel 2.11	<i>Compare output mode, fast PWM mode</i>	22
Tabel 2.12	<i>Compare output mode, phase correct PWM mode</i>	22
Tabel 2.13	<i>Clock select bit description</i>	22
Tabel 3.1	Standar subset dari <i>interface circuit</i> untuk aplikasi telekomunikasi	28
Tabel 4.1	Perbandingan hasil pembacaan sensor #1 Dengan thermometer digital	35
Tabel 4.2	Perbandingan hasil pembacaan sensor #1 Dengan thermometer digital	36
Tabel 4.3	Perbandingan hasil pembacaan sensor #1 & 2 Dengan thermometer digital	37

ABSTRAK

Nama : Ainur Rofiq
Program Studi : Teknik Elektro
Judul : Desain Sistem Monitoring Suhu Menggunakan
Sensor DS 18B20

Menejemen energi dewasa ini semakin menjadi pembicaraan yang hangat, mengingat semakin mahalnya harga energi dan kian terbatasnya sumber energi. Suhu merupakan salah satu parameter yang digunakan dalam menejemen energi, penggunaan suhu yang tepat dan sesuai kebutuhan di dalam ruangan, akan dapat membuat terjadinya pemanfaatan energi secara tepat. Untuk itu diperlukan suatu sistem yang dapat memonitor berapa suhu yang ada didalam suatu ruangan. Sistem monitoring suhu ini terdiri dari sensor suhu DS 18S20, mikrokontroller ATmega 8535 LCD (*liquid colour display*) 16 X 2 dan PC (*personal Computer*). Cara kerja dari sistem ini adalah sebagai berikut : perubahan suhu yang terjadi akan dideteksi oleh sensor DS 18S20, dimana data yang ada akan diolah oleh mikrokontroller ATmega 8535 dan ditampilkan ke LCD (*liquid colour display*) 16 X 2 serta PC (*Personal Computer*) untuk memonitor perubahan suhu yang terjadi. Dari hasil pengujian yang telah dilakukan keakuratan pembacaan suhu dari sistem yang dikerjakan ini cukup baik, ini dibuktikan dengan adanya error rata-rata sebesar 0.12 %.

Kata kunci:

Sensor Suhu DS 18B20, Mikrokontroller ATmega 8535, LCD (*liquid colour display*) PC (*personal computer*)

ABSTRACT

Name : Ainur Rofiq
Study Program : Electrical Engineering
Title : Design of temperature monitoring system using DS 18B20 sensor

Energy management lately has becoming very interested issue. Now energy cost increasing and energy source decreasing. One of parameter that used for energy management is temperature, using temperature with wisely will reduced energy wasted. This monitoring system consist of DS 18B20 sensor, ATmega 8535 microcontroller, LCD (liquid colour display) 16 X 2 and personal computer (PC). Temperature change will detected by DS 18B20 sensor and then this temperature data will processing by ATmega 8535 microcontroller. The measurement of temperature can be displayed by LCD (liquid colour display) 16 X 2 and also monitored from personal computer (PC) with Visual Basic 6 help. By seing 0.12 % average error temperature measurement on this system, we can conclude that this sistem has very accurate temperature measurement

Keywords :

Temperature Sensor DS 18B20, ATmega 8535 Microcontroller, Liquid Colour Display (LCD) 16 X 2, Personal Computer (PC)

BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Suhu merupakan salah satu besaran fisis dimana besaran fisis ini dapat kita ukur untuk mengetahui kondisi suhu yang ada dalam suatu ruangan. Hasil monitoring ini nantinya akan sangat berguna sebagai salah satu parameter untuk melakukan efisiensi energi. Karena dengan kita mengetahui nilai dari suhu yang berada dalam ruangan maka kita juga akan dapat menentukan, berapa energi yang harus kita gunakan untuk membuat ruangan tersebut nyaman.

1.2 TUJUAN

Tujuan dari skripsi yang berjudul “Desain Sistem Monitoring Suhu menggunakan Sensor DS 18B20” adalah merancang alat sistem monitoring Suhu untuk memantau suhu yang berada di dalam suatu ruangan.

1.3 BATASAN MASALAH

- Suhu yang dibandingkan hanya pada 2 titik pada suatu ruangan.
- Hasil sistem monitoring akan ditampilkan di komputer dengan menggunakan perangkat lunak Visual Basic 6.
- Log data tidak diberikan karena yang kami lakukan hanya memonitor suhu yang berada dalam ruangan.

1.4 METODE PEMBAHASAN

- Studi Literatur
Penulis melakukan pencarian literatur-literatur sebagai teori penunjang dalam perancangan sistem monitoring suhu, yang nantinya penulis dapat menentukan spesifikasi yang diinginkan dari perangkat lunak dan perangkat keras yang akan direalisasikan.
- Perancangan perangkat keras dan perangkat lunak
Perancangan akan dilakukan berdasarkan spesifikasi yang diinginkan dan ketersediaan komponen yang berada di pasar.

- Realisasi perangkat keras dan Perangkat lunak
Realisasi dilakukan setelah perancangan selesai, dimana realisasi perangkat lunak akan dilakukan setelah perangkat keras selesai dikerjakan.
- Pengukuran dan Pengujian
Pengukuran dan pengujian dilakukan untuk mengetahui apakah perancangan sistem ini sudah sesuai dengan parameter yang diinginkan. Jika tidak sesuai maka akan dilakukan analisa dan perbaikan, sampai sistem yang dirancang dapat bekerja sesuai dengan yang diharapkan oleh penulis.
- Perbaikan dan Penyempurnaan
Penulis melakukan perbaikan terhadap alat yang telah dirancang apabila masih terdapat kesalahan. Jika tidak terdapat kesalahan, maka penulis akan melakukan penyempurnaan alat ini.

1.5 SISTEMATIKA PENULISAN

Bab Satu yang merupakan pendahuluan yang berisi penjelasan mengenai latar belakang, tujuan penelitian, batasan masalah, metoda perancangan dan sistematika penulisan, dan pada bab Dua akan dijelaskan landasan teori mengenai Sensor Suhu DS 18B20, Mikrokontroler ATmega 8535, Program *Visual Basic 6*. Untuk bab Tiga merupakan rancang bangun sistem, menjelaskan mengenai bagaimana sistem bekerja. Prosedur pengambilan data dan pengujian data untuk selanjutnya dilakukan analisa tersaji pada bab Empat. Pengujian dilakukan dengan cara membandingkan nilai suhu dalam ruangan yang terdapat pada thermometer digital dengan nilai pengukuran suhu pada sistem yang telah didesain. Sedangkan bab Lima, Penutup, berisikan kesimpulan dari keseluruhan hasil perancangan yang telah dilakukan.

BAB 2

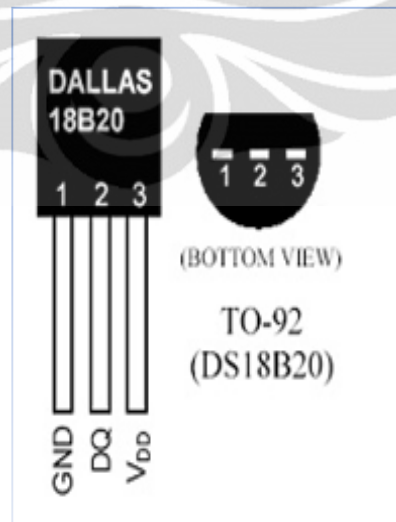
DASAR TEORI

2.1 Sensor Suhu DS 18S20^[2]

Suhu merupakan salah satu parameter yang digunakan dalam manajemen energi. Kebutuhan untuk memenej kebutuhan energi diperlukan ditengah semakin menipis dan mahalnya sumber energi. Untuk itu diperlukan suatu usaha agar dapat menggunakan energi secara bijak.

Sistem monitoring suhu sangat diperlukan dalam melakukan manajemen energy karena sistem monitoring suhu ini dapat digunakan untuk berbagai aktifitas dan kegiatan produksi diantaranya, adalah :control thermostatis di lingkungan industry,control dalam HVAC, mengontrol suhu didalam ruangan dan lain-lain.

Sensor Suhu DS 18B20 dipilih sebagai sensor dalam sistem monitoring suhu dikarenakan selain stabil, aman, handal, serta *user friendly* dalam hal pemasangan, juga dapat digunakan untuk melakukan pengukuran dengan range -55 to +125°C tanpa kalibrasi, dengan menggunakan 3 kaki yang berguna untuk: GND, VCC and 1-wire data. Sensor suhu DS 18B20 menyediakan 9-12 bit untuk pengukuran suhu dalam celcius (°C). Oleh karena itu sensor suhu DS 18B20 sangat cocok digunakan dalam sistem yang akan didesain ini.



Gambar 2.1 Sensor suhu DS 18B20

Spesifikasi teknis:

- DC supply voltage : 3 – 5.5 Volt
- Tegangan dapat diambilkan dari jalur data (VDD)
- Tingkat keakuratan : 0.5 °C dari -10°C sampai 85°C
- Batas temperatur : -55 °C s/d >125 °C
- Output : Digital 1-*wire*
- Resolusi : 9-12 Bit
- Waktu konversi maks : 750 ms.
- Tidak membutuhkan komponen lain untuk melakukan pembacaan suhu

Dari spesifikasi teknis yang disebut diatas, kita bisa melihat bahwa sensor suhu DS 18B20 dapat juga disebut dengan thermometer digital, dikarenakan kemampuan dari sensor suhu ini dalam mendeteksi perubahan suhu dengan keluaran digital.

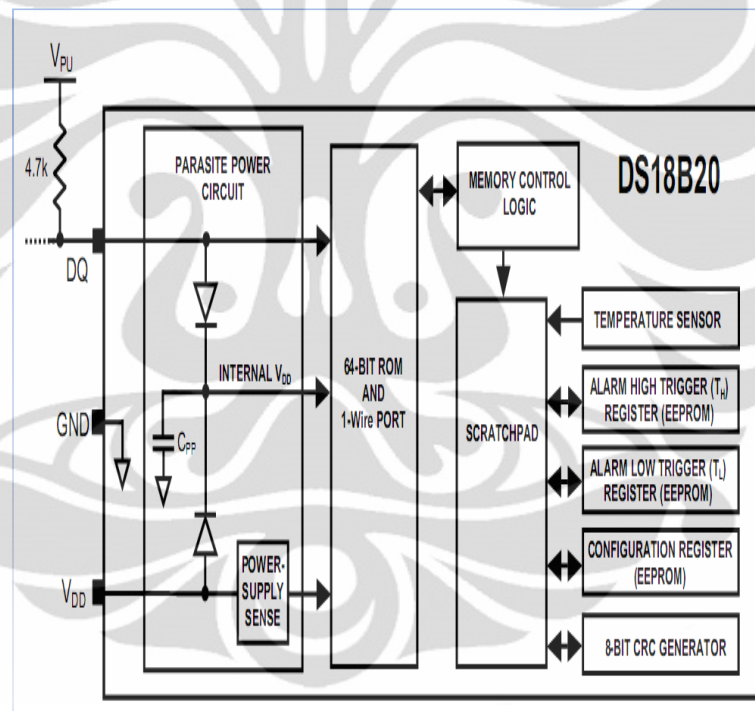
Sensor suhu DS 18B20 menggunakan protocol 1 wire komunikasi yang merupakan pengembangan dari Maxim, dimana protokol ini dipakai hanya dengan satu kontrol sinyal. Pada jalur control atau data dibutuhkan resistor *pull-up* dikarenakan semua sensor yang terhubung dengan jalur akan melalui 3 state atau port open drain yang dalam hal ini ditunjukkan oleh kaki DQ. Dari gambar 2.2 dibawah ini kita bisa melihat bahwa di dalam sensor DS terdapat blok-blok yang berfungsi:

- **64-Bit ROM**
Digunakan untuk menyimpan nomer kode dari sensor suhu DS 18B20, dimana tiap sensor akan memiliki nomer kode yang berbeda-beda yang sangat berguna dalam melakukan pemrograman pada mikrokontroler yang akan digunakan.
- **1-Wire Port**
Digunakan sebagai port untuk protokol komunikasi data bagi sensor suhu yang digunakan.
- **Scratchpad**
Memiliki 2 byte register data yang digunakan sebagai tempat menyimpan keluaran digital dari sensor suhu, 1 byte upper dan lower register alarm (T_H)

dan T_L). 1 byte juga dimiliki oleh register konfigurasi yang bersifat *non volatile* (EEPROM).

- Rangkaian daya parasit

Rangkaian daya parasit ini memungkinkan sensor suhu DS 18B20 mampu bekerja tanpa menggunakan pasokan daya dari luar. Pasokan daya nantinya akan berasal dari jalur data DQ melalui resistor pull-up pada saat DQ dalam posisi *high* atau 1. Akan tetapi sebagai alternatif sensor suhu DS 18B20 juga memungkinkan untuk menerima daya dari luar yang tentunya melalui kaki V_{DD} .



Gambar 2.2 Diagram blok DS18B20

Kemampuan dari sensor suhu DS 18B20 adalah hasil pembacaan dari sensor merupakan digital bukan analog sehingga kondisi yang terjadi adalah kondisi high dan low. Hal ini yang membedakan sensor suhu ini dengan sensor suhu yang lainnya, sebagai contoh LM 35 oleh karena itu sensor suhu DS 18B20 tidak perlu dikalibrasikan lagi seperti halnya sensor LM 35. Keluaran dari sensor

sudah dikalibrasikan dalam bentuk $^{\circ}\text{C}$ yang nantinya akan disimpan dalam register temperatur.

Penggunaan alarm dimungkinkan untuk memberikan batas atas dan batas bawah dari suhu dalam ruangan sesuai dengan yang kita inginkan. Hal ini sangat bermanfaat apabila nantinya sensor suhu akan digunakan sebagai tanda bahaya kebakaran yang tentunya harus bersinergi dengan sistem yang lainya seperti *buzzer*, lapu tanda bahaya dan lain-lain. Pengaturan dari alarm ini dilakukan di T_L (alarm rendah) dan T_H (alarm tinggi) yang berupa register 8 bit.

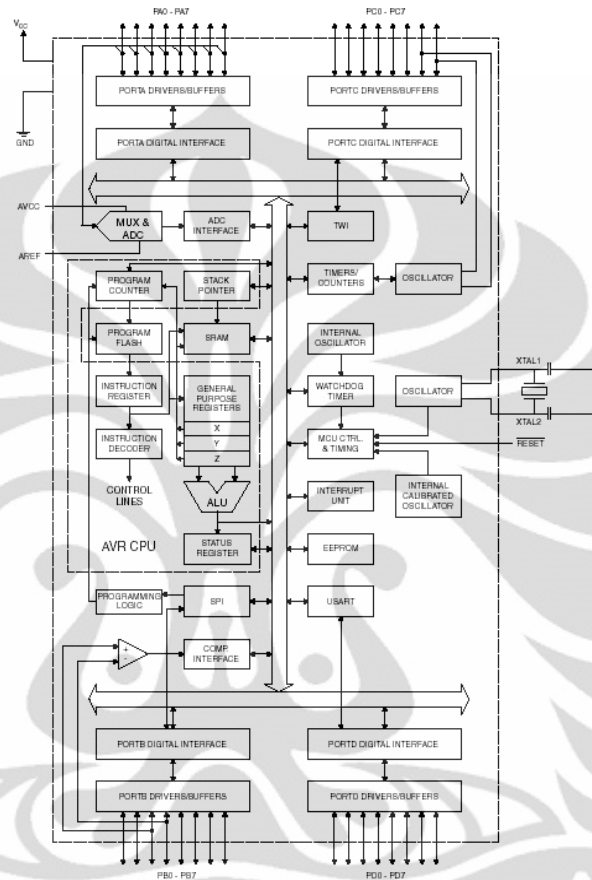
2.2 Mikrokontroler ATMEGA 8535^[1]

Mikrokontroler ATmega 8535 digunakan, karena selain sesuai dengan sensor suhu DS 18B20 juga karena mikrokontroler CMOS 8 bit ini memiliki kelebihan-kelebihan dibanding mikrokontroler yang lain. Kelebihan-kelebihan itu di antaranya:

- Memiliki USART yang terintegrasi dengan *baud rate* maksimum 20 Kbps.
- Mempunyai 8 channel ADC yang terintegrasi dengan resolusi 10 bit dengan metoda pengkonversian menggunakan *Succesive Approximation*.
- Memiliki fungsi *watchdog timer* yang dapat mencegah mikrokontroler dari kesalahan operasi.
- Memiliki 6 fungsi untuk menghemat energi yang digunakan, yaitu: *idle*, *ADC noise reduction*, *power save*, *power down*, *standby*, dan *extended standby mode*.
- Memiliki timer yang dapat difungsikan sebagai PWM (*Pulse Width Modulation*).
- Memiliki kristal internal hingga 8 MHz.
- Memiliki 32 buah *register* untuk *user* dan 64 buah *register* yang digunakan sebagai pengontrol fitur-fitur yang terintegrasi dalam mikrokontroler ini.
- Memiliki timer internal yang dapat difungsikan sebagai real-time timer.
- Pengoperasian memerlukan tegangan rendah, yaitu : 4,0 V s.d. 6,0 V.
- Pemograman dapat dilakukan dengan mudah dengan cara *in system programming*, artinya mikrokontroler ini dapat diganti programnya pada saat *run time*.

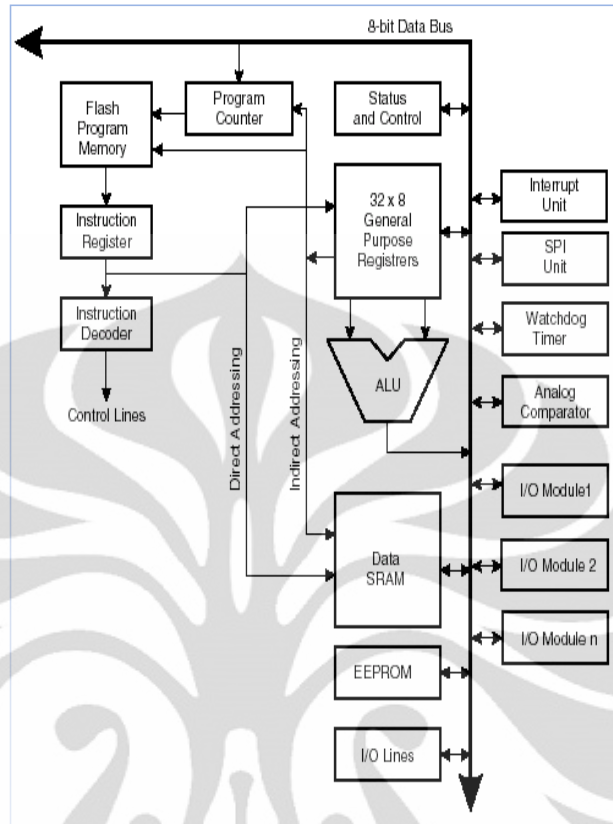
2.2.1 Arsitektur

Arsitektur dari Mikrokontroler ATmega 8535 ditunjukkan seperti gambar dibawah ini.



Gambar 2.3 Peta memori program

Dari gambar di atas terlihat bahwa ada 32 *register* yang digunakan 3 buah 16 bit *register pointer* alamat tidak langsung untuk pengalamatan ruang data. Satu dari 3 *pointer* ini dapat digunakan sebagai *pointer* alamat untuk *look up tables* di dalam memori program *flash*. ALU mendukung operasi aritmatika dan logika diantara register atau diantara konstanta dan sebuah register. Operasi *register* tunggal juga dapat dieksekusi di dalam ALU. Setelah operasi aritmatika, *register status* diperbaharui untuk memberikan informasi hasil operasi.



Gambar 2.4 Peta memori

Aliran program dilakukan oleh lompatan *conditional* dan *unconditional* serta instruksi kolom. Mampu mengamati secara langsung seluruh ruang alamat. Kebanyakan instruksi AVR memiliki sebuah format 16-bit. Setiap alamat memori program terdiri dari instruksi 16 bit atau 32-bit.

Selama *interrupt* dan *subroutines calls*, alamat kembali program *counter* disimpan dalam *stack*. *Stack* ini ditempatkan di dalam SRAM (*Static Random Access Memory*) dan konsekuensinya ukuran *stack* dibatasi oleh besar ukuran SRAM (*Static Random Access Memory*) dan penggunaan SRAM (*Static Random Access Memory*). Setiap program harus menginisialisasikan SP di dalam *routine reset*. Data SRAM (*Static Random Access Memory*) dapat dengan mudah diakses melalui 5 macam *mode* pengalamatan yang didukung dalam arsitektur AVR.

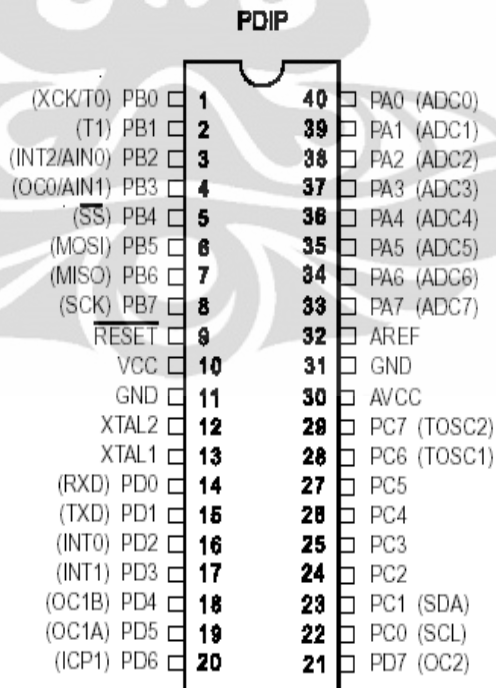
Modul *interrupt* memiliki register kontrol di dalam ruang I/O dengan tambahan bit yang *global interrupt enable* di dalam *register status*. Setiap *interrupt* memiliki *vector* yang terpisah di dalam tabel *interrupt vector*. *Interrupt*

memiliki prioritas yang berhubungan dengan posisi *interrupt vector*-nya. Semakin rendah alamat *interrupt vector*, semakin tinggi prioritasnya. Ruang memori I/O terdiri dari 64 alamat untuk fungsi *peripheral* CPU sebagai *control register*, SPI dan fungsi I.O lainnya. Memori I/O dapat diakses secara langsung atau sebagai lokasi ruang data di dalam *file register* 0 x 20 – 0 x 5F.

Setiap fitur yang terintegrasi dalam mikrokontroler ini mempunyai *register* yang mengontrol setiap fungsi dari masing-masing fitur. Register-register ini tidak dapat digunakan oleh pengguna sebagai tempat penyimpanan data dan termasuk ke dalam I/O *register*. Diantara register-register ini, ada yang bisa dibaca saja (*read only*) atau bisa baca/tulis (*read/write*). Setiap fitur juga memiliki *interrupt* masing-masing dengan prioritas yang sudah ditentukan.

2.2.2 Konfigurasi Pin

Mikrokontroler ATMEGA8535 ini memiliki 4 buah *port* dengan masing-masing berjumlah 8 bit. Adapun konfigurasi pin dari mikrokontroler ini seperti pada Gambar dibawah ini:



Gambar 2.5 Konfigurasi pin

Fungsi masing-masing pin adalah sebagai berikut:

- V_{CC}
Tegangan supply digital.
- GND
Ground.
- Port A (A0..A7)
Port A berfungsi sebagai *input* analog pada A/D Konverter. Port A juga berfungsi sebagai suatu Port I/O 8-bit dua arah, jika A/DKonverter tidak digunakan. Pin - pin Port dapat menyediakanresistor *internal pull-up* (yang dipilih untuk masing-masing bit).Port A *output buffer* mempunyai karakteristik gerakan simetrisdengan keduanya *sink* tinggi dan kemampuan sumber. Ketika pinPA0 ke PA7 digunakan sebagai *input* dan secara *eksternal* ditarikrendah, pin – pin akan memungkinkan arus sumber jika resistor *internal pull-up* diaktifkan. Pin Port A adalah *tri-stated* manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.
- Port B (B0..B7)
Port B adalah suatu Port I/O 8-bit dua arah dengan resistor internal pull-up (yang dipilih untuk beberapa bit). Port B *output buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sin k*tinggi dan kemampuan sumber. Sebagai input, pin port B yangsecara *eksternal* ditarik rendah akan arus sumber jika resistor *pullup* diaktifkan. Pin Port B adalah *tri-stated* manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.
- Port C (C0..C7)
Port C adalah suatu Port I/O 8-bit dua arah dengan resistor internalpull-up (yang dipilih untuk beberapa bit). Port C *output buffer*mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai input, pin port C yang secara *eksternal* ditarik rendah akan arus sumber jika resistor *pullup*diaktifkan. Pin Port C adalah *tri-stated* manakala suatu kondisi reset menjadi aktif, sekalipun waktu habis.

- Port D (D0..D70)

Port ini juga dapat difungsikan sebagai *port* I/O 8-bit dengan resistor *pull-up* internal. Output *buffer port* D memiliki karakteristik kendali simetris di kedua kemampuan *sink* dan *source*. Ketika pin D0-D7 digunakan sebagai input dan ditarik rendah dari luar, maka *port* ini akan menjadi sumber arus jika *pull-up* internal diaktifkan. Pin port D adalah *tri-state* ketika kondisi reset aktif bahkan ketika *clock* tidak jalan sekalipun.
- RESET

Merupakan pin reset on *power* untuk Mikrokontroler ATmega 8535. mikrokontroler akan reset bila diberi logik rendah (logik 0) pada pin ini.
- XTAL1

Merupakan pin untuk input *inverting oscillator amplifier* dan input untuk *internal clock*.
- XTAL2

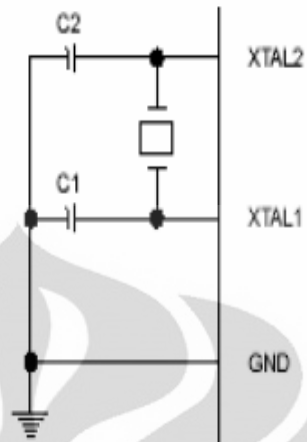
Merupakan output *inverting oscillator amplifier*.
- AVCC

Tegangan *supply* untuk ADC pada *port* A.
- AREF

Tegangan referensi untuk ADC.

2.2.3 Sistem Clock

Mikrokontroler, mempunyai sistem pewaktuan CPU, 12 siklus clock. Artinya setiap 12 siklus yang dihasilkan oleh ceramic resonator maka akan menghasilkan satu siklus mesin. Nilai ini yang akan menjadi acuan waktu operasi CPU. Untuk mendesain sistem mikrokontroler kita memerlukan sistem clock, sistem ini bisa di bangun dari clock eksternal maupun clock internal. Untuk clock internal, kita tinggal memasang komponen seperti di bawah ini.



Notes: C1, C2 = 30 pF \pm 10 pF for Crystals
 = 40 pF \pm 10 pF for Ceramic Resonators

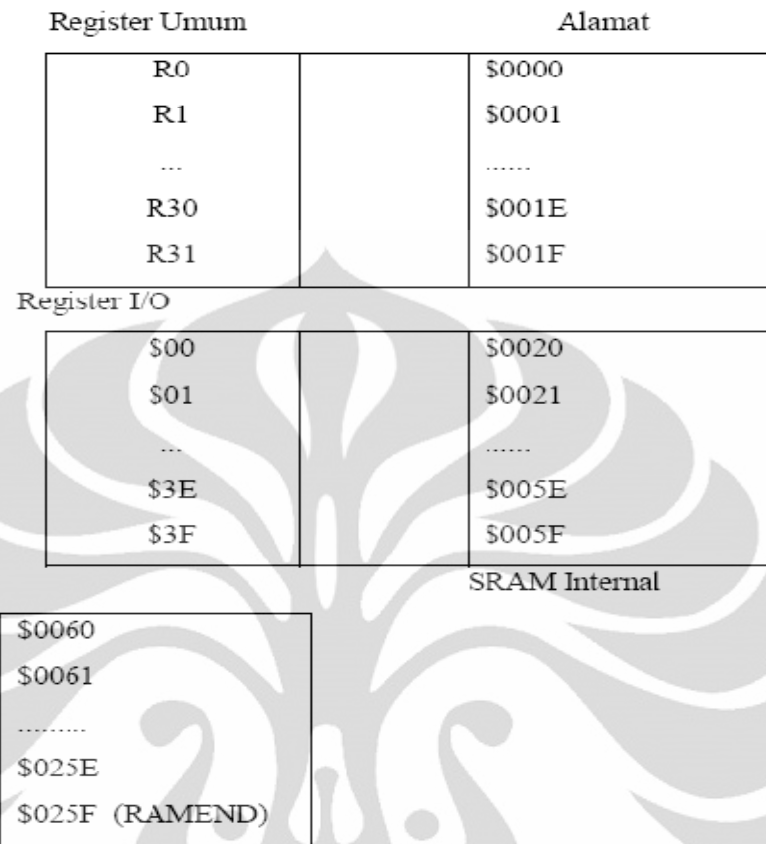
Gambar 2.6 Sistem clock

2.2.4 Memori

ATMega 8535 memiliki ruang pengalamatan memori data dan memori program yang terpisah. Sebagai tambahan, ATMega 8535 memiliki fitur suatu EEPROM Memori untuk penyimpanan data. Dimana ketiga ruang memori tersebut adalah reguler dan linier.

- Memori Data SRAM

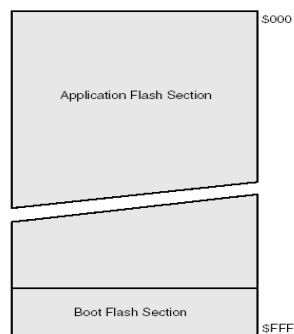
Memori data terbagi menjadi 3 bagian, yaitu 32 register umum, 64 buah register I/O, dan 512 byte SRAM Internal. Register keperluan umum menempati space data pada alamat dibawah, yaitu \$00 sampai \$1F. Sementara itu, register khusus untuk menangani I/O dan kontrol terhadap mikrokontroler menempati 64 alamat berikutnya, yaitu mulai dari \$20 hingga \$5F. Register tersebut merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai peripheral mikrokontroler, seperti kontrol register, timer/counter, fungsi – fungsi I/O, dan sebagainya. Register khusus alamat memori secara lengkap dapat dilihat pada Tabel 2.2. Alamat memori berikutnya digunakan untuk SRAM 512 byte, yaitu pada lokasi \$60 sampai dengan \$25F. Konfigurasi memori data ditunjukkan pada gambar di bawah ini.



Gambar 2.7 Peta memori data

- Memori Program Flash

ATMEGA8535 memiliki *in-system reprogrammable flash* untuk penyimpanan program. Karena instruksi AVR memiliki lebar 16-bit atau 32-bit, maka *flash* diatur sebagai 4K x 16. Untuk keamanan *software* memori *program flash* dibagi ke dalam dua bagian yaitu *boot program* dan aplikasi.



Gambar 2.8 Peta memori program

Memori ini memiliki ketahanan 10.000 kali tulis/hapus. *Program counter* memiliki lebar 12-bit untuk mengamati sampai 4K lokasi memori program. Gambar 2.20 memperlihatkan peta memori program.

- EEPROM

Memori ini memiliki 512 *byte* data EEPROM yang diatur sebagai sebuah ruang data terpisah, dimana setiap *byte* dapat dibaca dan ditulisi dengan ketahanan paling tidak 100.000 siklus baca/tulis.

2.2.5 Port Sebagai Input/Output Digital

ATmega 8535 mempunyai empat buah port yang bernama PortA, PortB, PortC, dan PortD. Keempat port tersebut merupakan jalur bi-directional dengan pilihan internal pull-up. Tiap port mempunyai tiga buah register bit, yaitu DDxn, PORTxn, dan PINxn. Huruf 'x' mewakili nama huruf dari port sedangkan huruf 'n' mewakili nomor bit. Bit DDxn terdapat pada I/O address DDRx, bit PORTxn terdapat pada I/O address PORTx, dan bit PINxn terdapat pada I/O address PINx. Bit DDxn dalam register DDRx (Data Direction Register) menentukan arah pin. Bila DDxn diset 1 maka Px berfungsi sebagai pin output. Bila DDxn diset 0 maka Px berfungsi sebagai pin input. Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin input, maka resistor pull-up akan diaktifkan. Untuk mematikan resistor pull-up, PORTxn harus diset 0 atau pin dikonfigurasi sebagai pin output. Pin port adalah tri-state setelah kondisi reset. Bila PORTxn diset 1 pada saat pin terkonfigurasi sebagai pin output maka pin port akan berlogika 1. Dan bila PORTxn diset 0 pada saat pin terkonfigurasi sebagai pin output maka pin port akan berlogika 0. Saat mengubah kondisi port dari kondisi *tri-state* (DDxn=0, PORTxn=0) ke kondisi *output high* (DDxn=1, PORTxn=1) maka harus ada kondisi peralihan apakah itu kondisi *pull-up enabled* (DDxn=0, PORTxn=1) atau kondisi *output low* (DDxn=1, PORTxn=0). Biasanya, kondisi pull-up enabled dapat diterima sepenuhnya, selama lingkungan impedansi tinggi tidak memperhatikan perbedaan antara sebuah *strong high driver* dengan sebuah pull-up. Jika ini bukan suatu masalah, maka bit PUD pada register SFIO dapat diset 1 untuk mematikan semua *pull-up* dalam semua port. Peralihan dari kondisi *input dengan pull-up* ke kondisi *output low* juga menimbulkan masalah yang sama.

Maka harus menggunakan kondisi tri-state ($DDx_n=0$, $PORTx_n=0$) atau kondisi output high ($DDx_n=1$, $PORTx_n=0$) sebagai kondisi transisi. Lebih detil mengenai port ini dapat dilihat pada manual datasheet dari IC ATmega8535.

2.2.6 Interrupt

Interrupt adalah suatu kejadian atau peristiwa yang menyebabkan mikrokontroler berhenti sejenak untuk melayani interrupt tersebut. Akan tetapi dalam menggunakan perintah ini yang harus diperhatikan adalah, kita harus mengetahui sumber-sumber interupsi, vektor layanan interupsi dan yang terpenting rutin layanan interupsi, yaitu subrutin yang akan dikerjakan bila terjadi interupsi. Pada sistem mikrokontroler yang sedang menjalankan programnya, saat terjadi interrupt, program akan berhenti sesaat, melayani interrupt tersebut dengan menjalankan program yang berada pada alamat yang ditunjuk oleh vektor dari interrupt yang terjadi hingga selesai dan kembali meneruskan program yang terhenti oleh interrupt tadi. Pada AVR terdapat 3 pin interupsi eksternal, yaitu INT0, INT1, dan INT2. Interupsi eksternal dapat dibangkitkan apabila ada perubahan logika atau logika 0 pada pin interupsi. Pengaturan kondisi keadaan yang menyebabkan terjadinya interupsi eksternal diatur oleh register MCUCR (MCU Control Register), yang terlihat seperti gambar ini.

Bit	7	6	5	4	3	2	1	0	
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.9 Register MCUCR

Pemilihan pengaktifan interupsi eksternal diatur oleh register GICR (General Interrupt Control Register) yang terlihat pada gambar berikut.

Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	INT2	-	-	-	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.10 Register GICR

Bit penyusunnya dapat dijelaskan sebagai berikut:

- ❖ Bit INT1 adalah bit untuk mengaktifkan interupsi eksternal 1. Apabila bit tersebut diberi logika 1 dan bit I pada SREG (status register) juga satu , maka interupsi eksternal 1 akan aktif.
- ❖ Bit INT0 adalah bit untuk mengaktifkan interupsi eksternal 0. Apabila bit tersebut diberi logika 1 dan bit I pada SREG (status register) juga satu , maka interupsi eksternal 0 akan aktif.
- ❖ Bit INT2 adalah bit untuk mengaktifkan interupsi eksternal 2. Apabila bit tersebut diberi logika 1 dan bit I pada SREG (status register) juga satu , maka interupsi eksternal 2 akan aktif.

Tabel 2.1 Reset dan interupt

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	0x000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	TIMER2 COMP	Timer/Counter2 Compare Match
5	0x004	TIMER2 OVF	Timer/Counter2 Overflow
6	0x005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	0x006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	0x007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	0x008	TIMER1 OVF	Timer/Counter1 Overflow
10	0x009	TIMER0 OVF	Timer/Counter0 Overflow
11	0x00A	SPI, STC	Serial Transfer Complete
12	0x00B	USART, RXC	USART, Rx Complete
13	0x00C	USART, UDRE	USART Data Register Empty
14	0x00D	USART, TXC	USART, Tx Complete
15	0x00E	ADC	ADC Conversion Complete
16	0x00F	EE_RDY	EEPROM Ready
17	0x010	ANA_COMP	Analog Comparator
18	0x011	TWI	Two-wire Serial Interface
19	0x012	INT2	External Interrupt Request 2
20	0x013	TIMER0 COMP	Timer/Counter0 Compare Match
21	0x014	SPM_RDY	Store Program Memory Ready

2.2.7 Timer/Counter

Timer/Counter pada AT Mega 8535 terdiri dari 3 buah. Yaitu Timer/Counter0 (8bit), Timer/Counter1 (16 bit), dan Timer/Counter2 (8 Bit).

Timer/Counter 0

Pengaturan Timer/Counter0 diatur oleh register TCCR0 yang dapat dilihat pada gambar di bawah ini:

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.11 Timer/counter 0

Bit 7 – FOC0: Force Output Compare

Bit 6,3-WGM01:WGM00 : Waveform generation Unit

Bit mengontrol kenaikan dari konter, sumber nilai maksimum counter, dan tipe dari jenis timer/counter yang dihasilkan, yaitu mode normal,clear timer,mode compare match, dan dua tipe dari PWM. Berikut table setingnya.

Tabel 2.2 Table setting

Mode	WGM01 (CTC0)	WGM00 (PWM0)	Mode of Operation	TOP	Update of OCR0	TOV0 Flag Set on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCRC	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

Bit 5,4 – COM1:COM00 : Compare Match Output Mode

Bit tersebut mengontrol pin OC0 (Output Compare pin). Apabila kedua bit itu 0 atau clear, maka pin OC0 berfungsi sebagai pin biasa. Namun, bila salah satu bit set, maka fungsi pin tergantung pada setting bit pada WGM00 dan WGM01. Berikut daftar table seting bit pada WGM00 dan WGM01.

Tabel 2.3 *Compare output mode, non-PWM mode***Table 40.** Compare Output Mode, non-PWM Mode

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Toggle OC0 on Compare Match
1	0	Clear OC0 on Compare Match
1	1	Set OC0 on Compare Match

Tabel 2.4 *Compare output mode, fast PWM mode***Table 41.** Compare Output Mode, Fast PWM Mode⁽¹⁾

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on Compare Match, set OC0 at TOP (Non-Inverting).
1	1	Set OC0 on Compare Match, clear OC0 at TOP (Inverting)

Tabel 2.5 *Compare output mode, phase correct PWM mode***Table 42.** Compare Output Mode, Phase Correct PWM Mode⁽¹⁾

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on Compare Match when up-counting. Set OC0 on Compare Match when down-counting.
1	1	Set OC0 on Compare Match when up-counting. Clear OC0 on Compare Match when down-counting.

Bit 2,1,0 – CS02,CS01,CS00: Clock select

Ketiga bit tersebut memilih sumber clock yang akan digunakan oleh Timer/Counter

Berikut Tabelnya.

Tabel 2.6 Clock select bit description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/counter stopped).
0	0	1	$clk_{IO}/(No\ prescaling)$
0	1	0	$clk_{IO}/8$ (From prescaler)
0	1	1	$clk_{IO}/64$ (From prescaler)
1	0	0	$clk_{IO}/256$ (From prescaler)
1	0	1	$clk_{IO}/1024$ (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Timer/Counter 1

Timer/Counter1 adalah 16 Bit Timer/Counter yang memungkinkan program pewaktuan lebih akurat.

Pengaturan pada Timer/Counter1 diatur melalui Resgister TCCR1A

Bit	7	6	5	4	3	2	1	0	
	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	W	W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.12 Timer/counter 1

Register COM1A1: 0 dan COM1B:0 mengontrol kondisi Pin Output Compare (OC1A dan OC1B). Jika salah satu atau kedua bit pada register COM1A:0 ditulis menjadi 1 , maka kaki pin OC1A tidak berfungsi normal sebagai I/O. begitu juga pada rekaki OC1B. Fungsi pada pin OC1A dan OC1B tergantung dari seting bit pada register WGM13:0 diset sebagai mode PWM atau mode non PWM.

Bit 3 FOC1A : Force Output Compare untuk chanel A

Bit 2 FOC1B : Force Output Compare untuk chanel B

Bit 1 WGM11:0 : Waveform generation Mode

Dikombinasikan dengan bit WGM13:2 yang terdapat pada register TCCR1B, bit tersebut mengontrol urutan pencacah dari counter, sumber maksimum (TOP) nilai counter, dan tipe gelombang yang dibangkitkan. Mode yang dapat dilakukan

antara lain mode normal, mode clear timer on compare Match (CTC) dan tiga tipe mode PWM. Setingan mode dapat dilihat pada table berikut.

Tabel 2.7 *Waveform generation mode bis description*

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	Reserved	-	-	-
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Bit	7	6	5	4	3	2	1	0	TCCR1B
	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.13 Register TCCR1B

Register TCCR1B digunakan juga untuk mengkonfigurasi/seting Timer/Counter1. Khususnya bit WGM13, WGM12.

Tabel 2.8 *Clock select bit description*

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{IO}/1$ (No prescaling)
0	1	0	$clk_{IO}/8$ (From prescaler)
0	1	1	$clk_{IO}/64$ (From prescaler)
1	0	0	$clk_{IO}/256$ (From prescaler)
1	0	1	$clk_{IO}/1024$ (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Timer/Counter 2

Pengaturan Timer/Counter2 diatur oleh register TCCR2 yang dapat dilihat pada gambar di bawah ini.

Bit	7	6	5	4	3	2	1	0	TCCR2
	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.14 Register TCCR2

Bit 7 – FOC2: Force Output Compare

Bit 6,3-WGM021:WGM20 : Waveform generation Unit

Bit mengontrol kenaikan dari konter, sumber nilai maksimum counter, dan tipe dari jenis timer/counter yang dihasilkan, yaitu mode normal,clear timer,mode compare match, dan dua tipe dari PWM. Berikut table setingnya.

Tabel 2.9 *Waveform generation mode bit description*

Mode	WGM21 (CTC2)	WGM20 (PWM2)	Timer/Counter Mode of Operation	TOP	Update of OCR2	TOV2 Flag Set on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR2	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

Bit 5,4 – COM1:COM00: Compare Match Output Mode

Bit tersebut mengontrol pin OC0 (Output Compare pin). Apabila kedua bit itu 0 atau clear, maka pin OC0 berfungsi sebagai pin biasa. Namun, bila salah satu bit set, maka fungsi pin tergantung pada setting bit pada WGM00 dan WGM01. Berikut daftar table seting bit pada WGM00 dan WGM01.

Tabel 2.10 *Compare output mode, non-PWM mode*

COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected.
0	1	Toggle OC2 on Compare Match.
1	0	Clear OC2 on Compare Match.
1	1	Set OC2 on Compare Match.

Tabel 2.11 Compare output mode, fast PWM mode

COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected.
0	1	Reserved
1	0	Clear OC2 on Compare Match, set OC2 at TOP (Non-Inverting).
1	1	Set OC2 on Compare Match, clear OC2 at TOP (Inverting).

Tabel 2.12 Compare output mode, phase correct PWM mode

COM21	COM20	Description
0	0	Normal port operation, OC2 disconnected.
0	1	Reserved
1	0	Clear OC2 on Compare Match when up-counting. Set OC2 on Compare Match when down-counting.
1	1	Set OC2 on Compare Match when up-counting. Clear OC2 on Compare Match when down-counting.

Bit 2,1,0 – CS22,CS21,CS20 : Clock select

Ketiga bit tersebut memilih sumber klok yang akan digunakan oleh Timer/Counter.

Tabel 2.13 Clock select bit description

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{T2S}/(\text{No prescaling})$
0	1	0	$clk_{T2S}/8$ (From prescaler)
0	1	1	$clk_{T2S}/32$ (From prescaler)
1	0	0	$clk_{T2S}/64$ (From prescaler)
1	0	1	$clk_{T2S}/128$ (From prescaler)
1	1	0	$clk_{T2S}/256$ (From prescaler)
1	1	1	$clk_{T2S}/1024$ (From prescaler)

Register TIMSK dan TIFR

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.15 Register TIMSK

Bit0 – Timer/Counter0 Overflow Interrupt Enable

Jika bit tersebut diberi logika satu dan I SREG juga set, maka bisa dilakukan enable interupsi overflow Timer/Counter0.

Bit1- Timer/Counter0 Output Compere Match Interrupt Enable

Jika bit tersebut diberi logika satu dan I SREG juga set, maka bisa dilakukan enable Interupsi Output Compere Match.

Bit2- Timer/Counter1 Overflow Interrupt Enable

Jika bit tersebut diberi logika satu dan I SREG juga set, maka bisa dilakukan enable interupsi overflow Timer/Counter1.

Bit	7	6	5	4	3	2	1	0	TIFR
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.16 Register TIFR

Bit0 – Timer/Counter0 Overflow Flag

Bit akan bernilai satu jika Timer/Counter0 Overflow. Bit dapat dinolkan lagi dengan memberikan logika satu ke bit Flag ini.

Bit1- Output Comapre Flag 0

Bit akan berniali satu jika nilai pada Timer/Counter0 sama dengan nilai pada OCR0 –Output Comapre.

BAB 3

PERANCANGAN SISTEM

Pada bab ini akan dibahas mengenai tujuan dari perancangan yang dilakukan. Selanjutnya dibahas mengenai spesifikasi dari perangkat keras dan perangkat lunak yang dibuat serta langkah-langkah dari perancangan yang akan dilakukan.

3.1 Tujuan Perancangan

Perancangan bertujuan untuk agar dapat memperoleh alat dengan spesifikasi yang tepat sesuai dengan yang kita inginkan, sehingga kita dapat memperkecil kesalahan, yang berakibat kerugian. Untuk itu diperlukan suatu ketelitian dalam merancang alat yang akan dibuat.

Pemilihan komponen yang tepat yang nantinya akan mendukung alat yang akan kita rancang mutlak diperlukan, akan tetapi apabila sistem ini menggunakan perangkat lunak, maka tujuan, fungsi dan cara kerja perangkat keras harus bersinergi dengan perangkat lunak yang akan kita buat sehingga alat ini nantinya akan menjadi suatu sistem yang saling bersinergi.

3.2 Fungsi Alat

Skripsi yang dibuat membahas mengenai desain sistem monitoring suhu dimana kan memiliki fungsi antara lain:

- Dapat melakukan Monitoring Suhu dalam ruangan dengan menggunakan dua sensor DS 18B20.
- Hasil monitoring suhu tersaji pada LCD 16 X 2 dan monitor komputer untuk memudahkan pengawasan suhu.

3.3 Spesifikasi Alat

- Sensor Suhu DS 18B20.
- Sistem Monitoring Suhu terintegrasi dengan microsoft *hyper terminal* yang dijalankan pada PC dengan menggunakan antarmuka RS 232.
- Tampilan di monitor komputer menggunakan *Visual Basic 6*.

3.4 Langkah–Langkah Perancangan

3.4.1 Perancangan Perangkat Keras

- Desain Perangkat keras
 - Rangkaian antarmuka power supply.
 - Rangkaian antarmuka serial.
 - Rangkaian sensor suhu DS 18B20.
 - Rangkaian sistem minimum ATmega 8535
 - Rangkaian LCD (*liquid colour display*) 16X2.
- Pemasangan komponen
- Pembuatan kotak

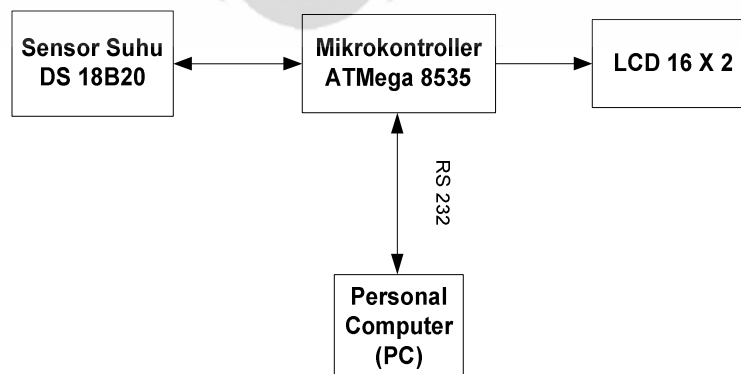
3.4.2 Perancangan Perangkat Lunak

Perangkat lunak yang digunakan adalah visual basic 6 dimana program ini digunakan sebagai tampilan dan merupakan refleksi dari LCD yang ada di perangkat lunak.

3.5 Realisasi

3.5.1 Realisasi Perangkat Keras

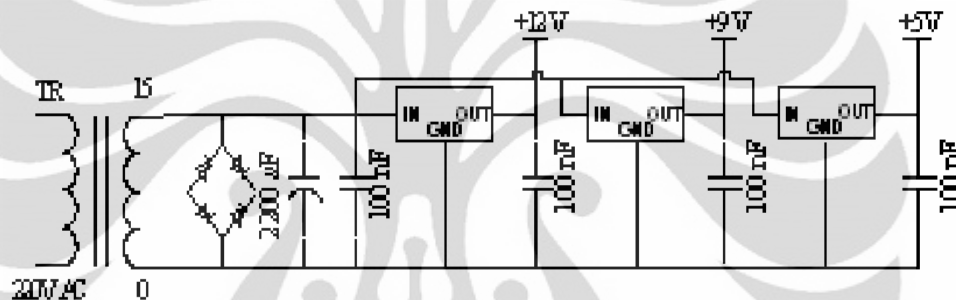
Realisasi dari pengerjaan perangkat keras ini akan berdasarkan dari blok diagram yang telah dibuat agar sistem monitoring suhu ini dapat bekerja dengan baik. Blok diagram sistem secara keseluruhan terdiri dari sensor DS 18B20 yang berjumlah dua buah, kabel RS 232, Mikrokontroler ATmega 8535, LCD 16 X 2, PC (*Personal Computer*).



Gambar 3.1 Blok diagram sistem monitoring suhu

Suhu merupakan salah satu besaran fisis. Suhu dalam suatu ruangan berkisar antara 20 – 25 derajat Celsius ($^{\circ}\text{C}$). Untuk mengetahui kondisi suhu dalam ruangan serta memonitoring perubahan-perubahan yang terjadi didalam ruangan tersebut maka penulis mencoba merancang “ Sistem monitoring suhu”. Dari gambar diatas secara garis besar bisa lihat bahwa perubahan suhu akan dideteksi oleh sensor temperatur DS 18B20 dimana hasil pembacaan dari sensor suhu tersebut akan ditampilkan di PC (*Personal Computer*) dengan bantuan perangkat lunak *Visual Basic 6*.

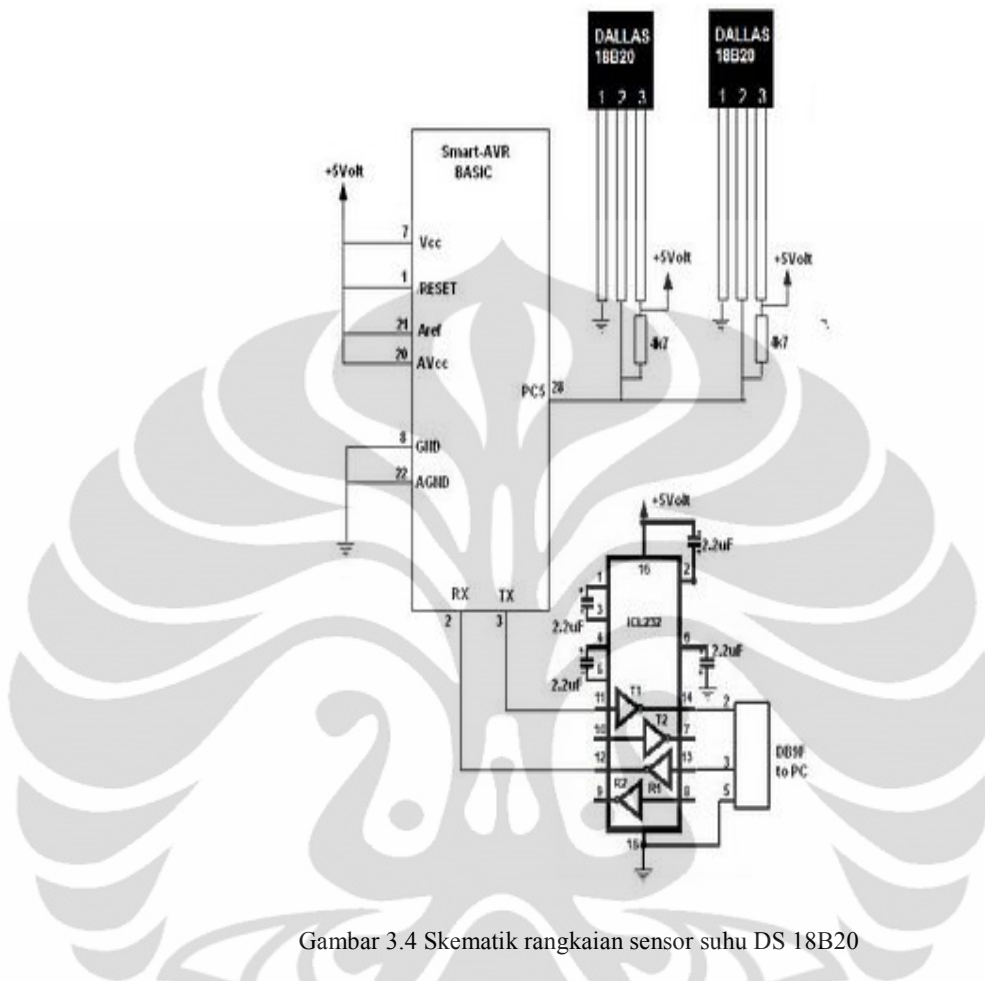
- Rangkaian antarmuka power supply
 - Skematik rangkaian.



Gambar 3.2 Skematik power supply

Pada sistem monitoring suhu ini catu daya yang digunakan memanfaatkan sumber tegangan dari PLN sebesar 220 VAC, Tegangan 220 VAC ini kemudian di turunkan menjadi 15 VAC melalui trafo penurun tegangan. Dioda bridge akan menyearahkan tegangan. Kapasitor berfungsi untuk menghilangkan *noise* pada tegangan DC. masuk ke IC regulator yang fungsinya adalah untuk menstabilkan tegangan. IC regulator ini terdiri dari tiga buah IC, yaitu LM7805, LM7809 dan LM7812 yang menghasilkan tegangan DC sebesar +5V, +9V dan +12V.

- Rangkaian antarmuka serial
 - Skematik rangkaian.



Gambar 3.4 Skematik rangkaian sensor suhu DS 18B20

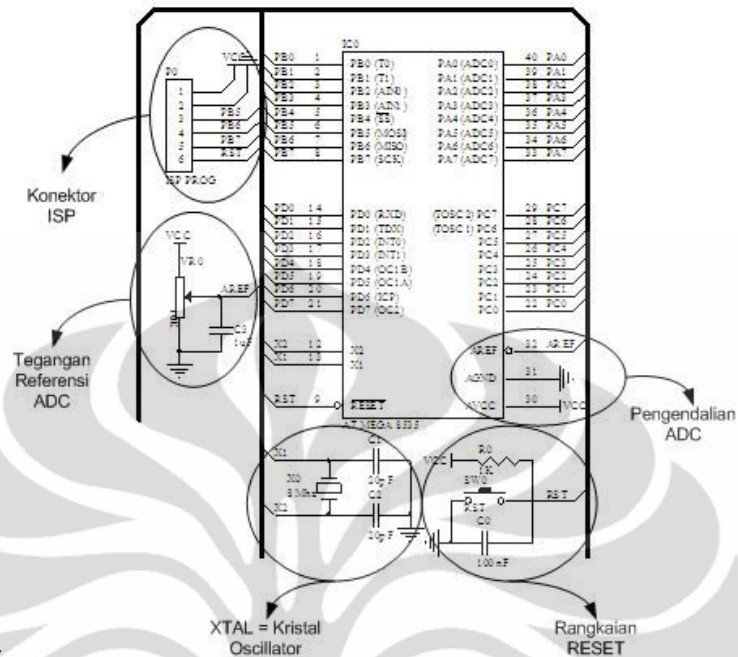
Dari gambar diatas kita bisa melihat bahwa rangkaian dua sensor suhu yang digunakan di paralel satu dengan yang lainnya. Hal ini menjadikan penggunaan sensor DS 18B20 terlihat lebih mudah karena kabel yang berasal dari sensor suhu DS 18B20 yang masuk ke mikrokontroler ATmega 8535 hanya 1 kabel. Inilah yang menjadikan salah satu keunggulan dari sensor DS 18B20 dengan konsep *onewire* nya, dimana berapapun jumlah sensor yang dipasang maka kabel yang digunakan untuk masuk ke sistem hanya berjumlah satu buah. Untuk sistem monitoring suhu nantinya kita akan menggunakan lebih dari satu sensor yaitu 2 sensor suhu DS 18B20. Hal ini sangat diperlukan untuk mendapatkan hasil pengukuran yang lebih akurat daripada hanya menggunakan 1 buah sensor suhu untuk menggunakan dan mengakses dua sensor ini ada beberapa perintah yang digunakan yaitu:

- FN:OWRESET, yang berfungsi untuk mereset 1-wire device.
- FN:OWWRITE, yang berfungsi menulis 1byte data ke 1-wire device.
- FN:OWREAD, yang berfungsi membaca 1byte data ke 1-wire device.
- FN:OWSEARCH, yang berfungsi mencari dan memetakan "64-BIT LASERED ROM CODE" 1-wire device ke dalam internal rom table, dan didalam perintah ini juga menghasilkan informasi berapa banyak device yang tersambung dalam 1-wire bus, informasi ini tersimpan dalam variable owamount.
- FN:OWMATCROM, yang berfungsi untuk mengakses 1-wire yang bersesuaian dengan 64bit ROM yang terkirim, dalam perintah ini harus di tentukan device numbernya dulu dengan memberi nilai dalam variable OWNUMBER

- Rangkaian sistem minimum ATmega 8535

Rangkaian sistem minimum ini merupakan penerima data dari input rangkaian komparator dimana akan dilakukan perhitungan jumlah pulsa. Kabel parallel merupakan suatu penghubung dengan modul SP (*In System Programming*). Pemasangan LCD (liquid colour display) 16 X 2 juga terdapat dalam rangkaian sistem minimum ini. Berikut adalah spesifikasi dari sistem minimum ini:

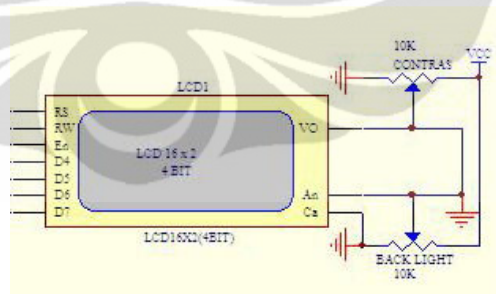
- Tegangan suplai 12 Vdc
- Kristal 8 MHz
- Port input : PA.0
- Port output : PB.0 s.d PB.7.
- IC ATmega 8535



Gambar 3.5 Rangkaian Sistem minimum Atmega 8535

- Rangkaian LCD (*liquid colour display*) 16 X 2

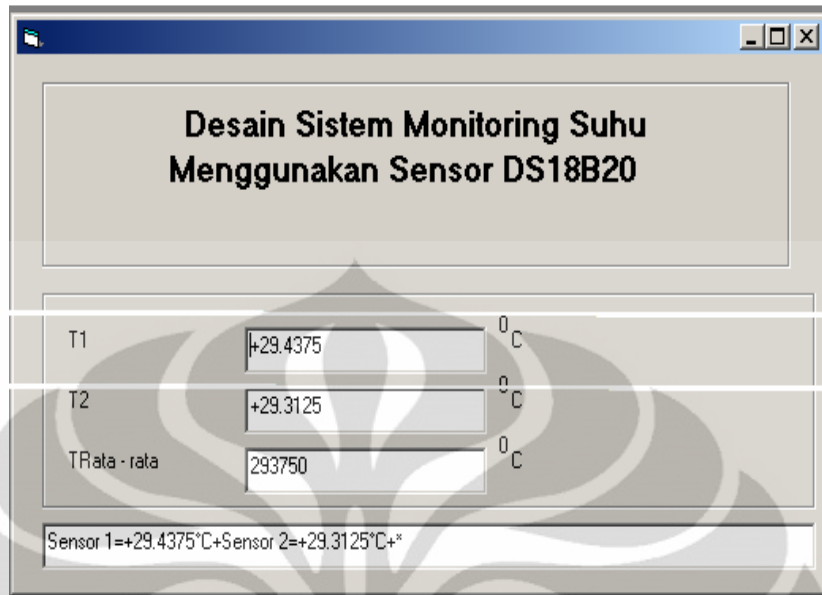
LCD (*liquid colour display*) ini memiliki 16 baris dan 2 kolom dan berguna untuk menampilkan hasil pembacaan dari sistem monitoring suhu.



Gambar 3.6 Rangkaian LCD (*liquid colour display*) 16 X 2

3.5.2 Realisasi Perangkat Lunak

Dalam menampilkan hasil pengukuran dari sistem monitoring suhu ini, maka perangkat lunak yang digunakan adalah *visual basic 6*.



Gambar 3.7 Tampilan sistem monitoring suhu

Pada gambar diatas terdapat 4 kolom yang semuanya adalah data yang diperoleh dari hasil pembacaan sistem monitoring suhu.

- T1 berarti suhu 1 dimana, data ini merupakan hasil dari pembacaan sensor suhu 1.
- T2 berarti suhu 2 dimana, data ini berasal dari pembacaan sensor suhu 2.
- Trata-rata adalah suhu rata-rata dari hasil pembacaan sensor 1 (T1) dan sensor 2 (T2)..
- Kolom paling bawah merupakan tampilan dari *hyper terminal* yang merupakan data mentah yang nantinya akan diolah perangkat lunak *visual basic 6*.

BAB 4

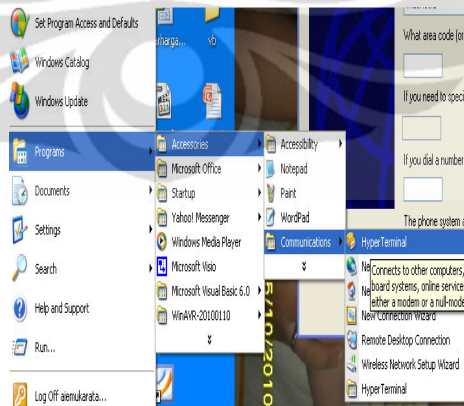
PENGUJIAN DAN ANALISIS

4.1 Pengujian

4.1.1 Koneksi Komputer (PC) dengan Mikrokontroler

Koneksi ini dapat dilihat dilihat pada *hyper terminal*, jika terjadi komunikasi maka set perintah yang telah disimpan pada mikrokontroler sudah dapat dijalankan. Prosedur pengujiannya adalah sebagai berikut :

- Menghubungkan port serial yang ada di mikrokontroler ATmega 8535 dengan port serial komputer (PC).
- Hubungkan modul Mikrokontroler 8535 dengan *power supply* kemudian hidupkan lalu buka aplikasi hyperterminal pada komputer.
- Lakukan pengaturan antara koneksi hyperterminal dengan mikrokontroler ATmega 8535 Setelah pengaturan koneksi selesai maka akan dibutuhkan waktu sampai bisa melakukan komunikasi antara komputer dengan mikrokontroler ATmega 8535. Jika koneksi ini berhasil maka akan terjadi komunikasi antara modul mikrokontroler ATmega 8535 dengan komputer.
- Dibawah ini pengaturan yang dilakukan untuk koneksi hyperterminal untuk mendapatkan koneksi antar mikrokontroler ATmega 8535 dengan personal computer (PC).



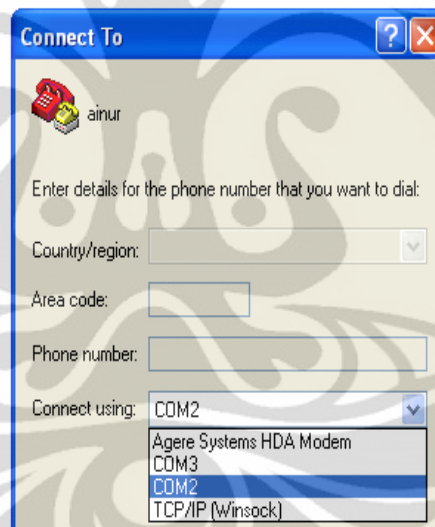
Gambar 4.1 Membuka *hyper terminal*

- Tab dari *hyper terminal* dibuka melalui tab program – accessories – communication.



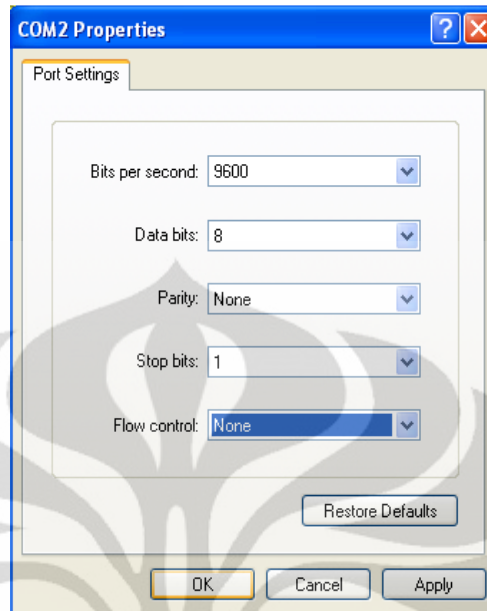
Gambar 4.2 Memasukkan nama

- Setelah itu masukkan nama, hal ini dilakukan untuk memberi identitas buat koneksi yang akan dibuat.



Gambar 4.3 Memilih port komunikasi

- Setelah itu yang harus dilakukan adalah pemilihan port komunikasi. Untuk sistem ini port yang digunakan adalah port komunikasi 2.



Gambar 4.4 *Setting port*

- Setelah port komunikasi 2 dipilih maka harus dilakukan *setting port*. Seperti yang terlihat pada gambar diatas pengaturan yang dilakukan harus sesuai.

4.1.2 Koneksi Sensor Suhu DS 18B20 dengan Mikrokontroller

Koneksi ini dilakukan untuk melakukan pengecekan apakah sensor dapat bekerja dengan baik dan mampu berkomunikasi dengan mikrokontroller melwati port yang ada. Untuk itu diperlukan tampilan dari LCD mengenai pembacaan sensor yang menunjukkan bahwa sensor suhu DS 18B20 bekerja dengan baik.



Gambar 4.5 Tampilan sensor suhu

4.2 Analisa

4.2.1 Hasil Pengujian Perbandingan Pengukuran Suhu

Pengujian ini dilakukan dengan membandingkan hasil pengukuran suhu menggunakan thermometer digital dengan hasil pengukuran menggunakan sistem yang telah dibuat. pengujian ini bertujuan untuk mencari error dalam pembacaan suhu. Thermometer digital digunakan sebagai titik acuan karena sensor yang digunakan dalam sistem monitoring suhu DS 18B20 hasil keluarannya juga digital.

Thermometer digital yang digunakan sebagai titik acuan memiliki spesifikasi sebagai berikut:

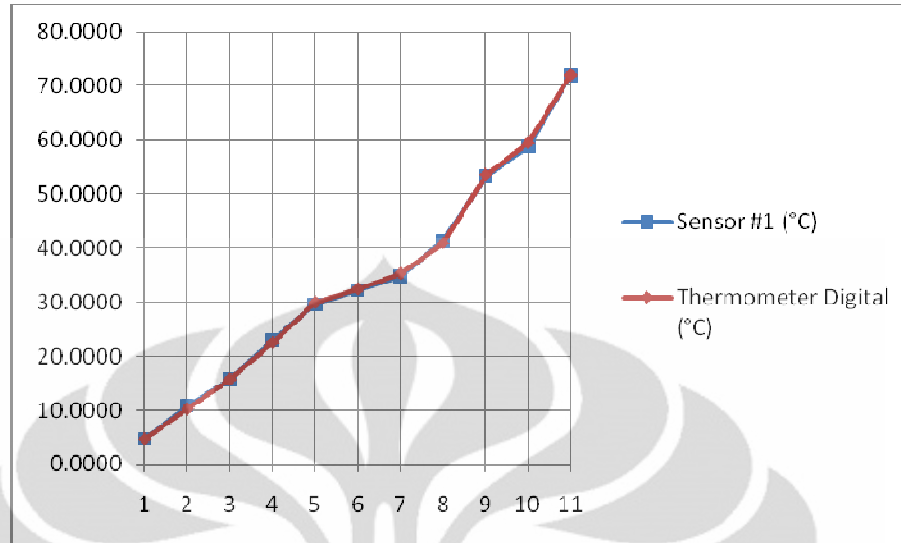
Nama alat : Sanfix
 Distance / spot ratio : 12 : 1
 Resolusi : 0.1 °C
 Rentang suhu : -32 °C – 535 °C
Response Time : 500ms

Tabel 4.1 Perbandingan hasil pembacaan sensor #1 dengan thermometer digital

No.	Sensor #1 (°C)	Thermometer Digital (°C)	%Error
1	4.6255	4.62	0.12
2	10.8125	10.21	5.90
3	15.6173	15.65	0.21
4	23.0159	22.49	2.34
5	29.6252	29.90	0.92
6	32.1832	32.41	0.70
7	34.8255	35.26	1.23
8	41.2581	40.87	0.95
9	53.3709	53.52	0.28
10	58.8622	59.46	1.01
11	72.0425	71.98	0.09

%Error Rata-rata

1.25



Gambar 4.6 Grafik Perbandingan hasil pembacaan sensor #1 dengan thermometer digital

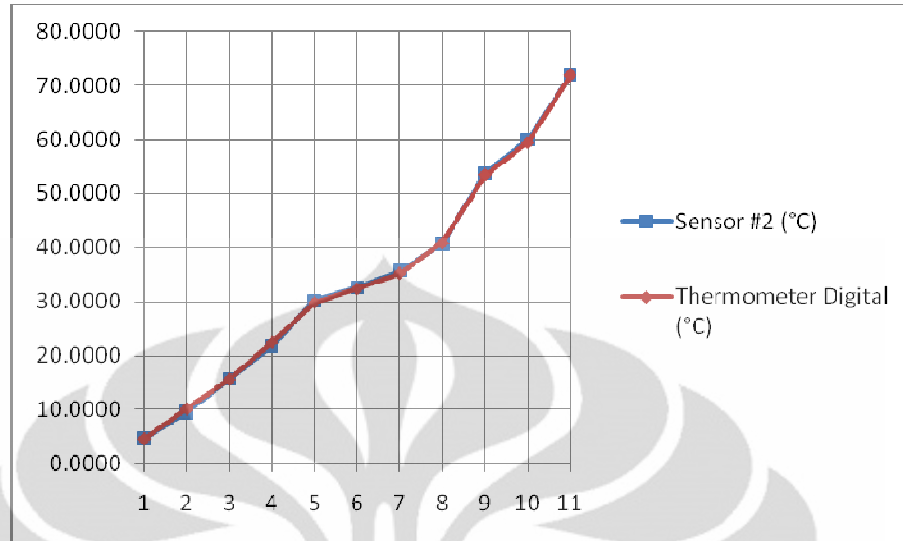
Dari tabel 4.1 terlihat bahwa hasil error rata-rata adalah 1.25 % . Hal ini menunjukkan tingkat kesalahan pembacaan suhu dengan menggunakan sistem monitoring suhu adalah 1.25 % jika dibandingkan dengan pembacaan suhu oleh thermometer digital. Hal ini terjadi karena penempatan sensor suhu DS 18B20 haruslah tepat, agar bisa memperoleh hasil pembacaan yang tepat. error terkecil yang bisa didapat oleh sensor 1 dari sistem terjadi pada kondisi pengukuran ke 10 pada saat suhu 72.0425 °C.

Tabel 4.2 Perbandingan hasil pembacaan sensor #2 dengan thermometer digital

No.	Sensor #2 (°C)	Thermometer Digital (°C)	%Error
1	4.6205	4.62	0.01
2	9.5625	10.21	6.34
3	15.6401	15.65	0.06
4	21.8251	22.49	2.96
5	30.1875	29.90	0.96
6	32.6288	32.41	0.68
7	35.6550	35.26	1.12
8	40.7521	40.87	0.29
9	53.7547	53.52	0.44
10	59.9520	59.46	0.83
11	71.9861	71.98	0.01

%Error Rata-rata

1.24



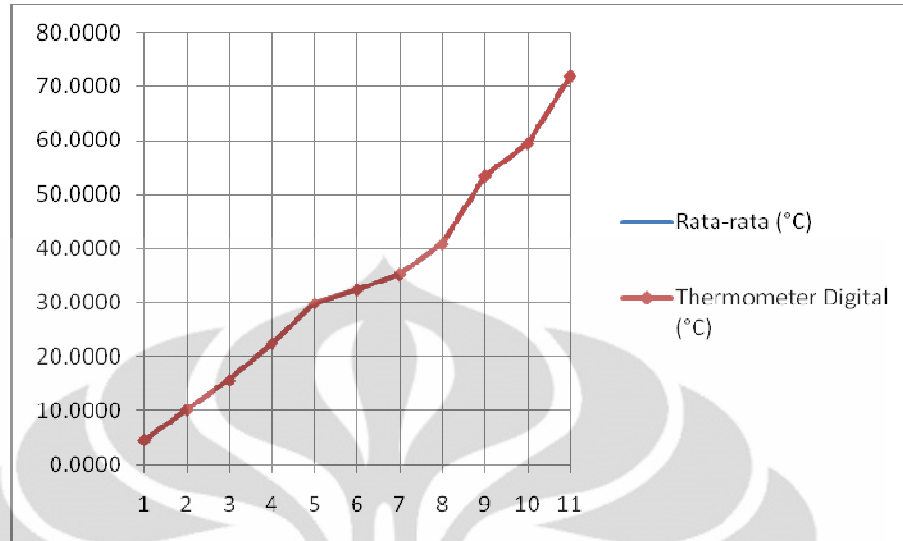
Gambar 4.7 Grafik Perbandingan hasil pembacaan sensor #2 dengan thermometer digital

Dari tabel 4.2 terlihat bahwa hasil error rata-rata adalah 1.24. selisih kesalahan pembacaan sensor 1 dengan sensor 2 tidak terlalu berbeda jauh dimana perbedaan yang terjadi hanya 0.01%. Hal ini menunjukkan bahwa kedua sensor telah bekerja dengan baik.

Tabel 4.3 Perbandingan rata-rata hasil pembacaan sensor #1 & #2 dengan thermometer digital

No.	Sensor #1 (°C)	Sensor #2 (°C)	Rata-rata (°C)	Thermometer Digital (°C)	%Error
1	4.6255	4.6205	4.6230	4.62	0.06
2	10.8125	9.5625	10.1875	10.21	0.22
3	15.6173	15.6401	15.6287	15.65	0.14
4	23.0159	21.8251	22.4205	22.49	0.31
5	29.6252	30.1875	29.9064	29.90	0.02
6	32.1832	32.6288	32.4060	32.41	0.01
7	34.8255	35.6550	35.2403	35.26	0.06
8	41.2581	40.7521	41.0051	40.87	0.33
9	53.3709	53.7547	53.5628	53.52	0.08
10	58.8622	59.9520	59.4071	59.46	0.09
11	72.0425	71.9861	72.0143	71.98	0.05

%Error Rata-rata 0.12



Gambar 4.8 Perbandingan rata-rata hasil pembacaan sensor #1 & #2 dengan thermometer digital

Berdasarkan pengukuran yang telah dilakukan, terlihat bahwa error terbesar yang terjadi terletak pada percobaan ke 7 dengan error mencapai 0.33% hal ini kemungkinan terjadi akibat posisi pembacaan sensor suhu yang kurang tepat sehingga berpengaruh kepada hasil dari pembacaan sensor. Walaupun begitu dari tabel dan grafik di atas kita bisa melihat error total yang terjadi akibat pembacaan suhu sangatlah kecil, hal ini terjadi karena perbedaan pembacaan hanya terjadi pada 1 angka dibelakang koma sehingga menunjukkan sistem monitoring suhu yang telah dikerjakan, dapat berjalan dengan baik. Kinerja yang baik dari sistem monitoring suhu sangat bergantung juga kepada lingkungan dimana sensor itu berada, dimana sensor suhu DS 18B20 harus mendapatkan perlindungan apabila berada dilingkungan yang ekstrim, misalkan panas, hujan, angin yang terlalu berlebihan agar mendapatkan hasil pembacaan yang akurat.

4.2.2 Hasil Pengujian Time Response

Pengujian ini dilakukan dengan membandingkan waktu yang diperlukan sistem monitoring suhu untuk mendeteksi perubahan suhu yang terjadi dan waktu yang diperlukan untuk mendeteksi perubahan suhu tersebut.

Pengujian pertama program yang digunakan mikrokontroller ini menggunakan waktu tunda 500 ms. Sedangkan untuk pengujian kedua waktu tunda dirubah menjadi 5 ms.

```

Sensor 1==28.3125 C-Sensor 2==29.6250 C-Sensor 1==28.3125 C-Sensor 2==29.6250
C-Sensor 1==28.3125 C-Sensor 2==29.5625 C-Sensor 1==28.3125 C-Sensor 2==29.56
29.3750 C-Sensor 1==28.3125 C-Sensor 2==29.3125 C-Sensor 1==28.3125 C-Sensor
==29.3125 C-Sensor 1==28.3125 C-Sensor 2==29.2500 C-Sensor 1==28.3125 C-Senso
2==29.2500 C-Sensor 1==28.3125 C-Sensor 2==29.2500 C-Sensor 1==28.3125 C-Sen
C-Sensor 2==29.0625 C-Sensor 1==28.3125 C-Sensor 2==29.0625 C-Sensor 1==28
3125 C-Sensor 2==29.0000 C-Sensor 1==28.3125 C-Sensor 2==29.0000 C-Sens
sor 1==28.3125 C-Sensor 2==28.9375 C-Sensor 1==28.3125 C-Sensor 2==28.9375 C-S
ensor 1==28.3125 C-Sensor 2==28.9375 C-Sensor 1==28.3125 C-Sensor 2==28.8750 C
-Sensor 1==28.3125 C-Sensor 2==28.8750 C-Sensor 1==28.3125 C-Sensor 2==28.8750
C-Sensor 1==28.3125 C-Sensor 2==28.8125 C-Sensor 1==28.3125 C-Sensor 2==28.81
5 C-Sensor 1==28.3125 C-Sensor 2==28.8125 C-Sensor 1==28.3125 C-Sensor 2==28
.7500 C-Sensor 1==28.3125 C-Sensor 2==28.7500 C-Sensor 1==28.3125 C-Sensor 2=
28.6875 C-Sensor 1==28.3125 C-Sensor 2==28.6875 C-Sensor 1==28.3125 C-Sensor
==28.6875 C-Sensor 1==28.3125 C-Sensor 2==28.6875 C-Sensor 1==28.3125 C-Senso
2==28.6250 C-Sensor 1==28.3125 C-Sensor 2==28.6250 C-Sensor 1==28.3125 C-Sen
or 2==28.6250 C-Sensor 1==28.3125 C-Sensor 2==28.6250 C-Sensor 1==28.3125 C-S
ensor 2==28.6250 C-Sensor 1==28.3125 C-Sensor 2==28.5625 C-Sensor 1==28.3125 C
Sensor 2==28.5625 C-Sensor 1==28.3125 C-Sensor 2==28.5625 C-Sensor 1==28.3125
C-Sensor 2==28.5625 C-Sensor 1==28.3125 C-Sensor 2==28.5625 C-Sensor 1==28.31
5 C-Sensor 2==28.5000 C-Sensor 1==28.3125 C-Sensor 2==28.5000 C-Sensor 1==28
.125 C-Sensor 2==28.5000 C-Sensor 1==28.3125 C-Sensor 2==28.5000 C-Sensor 1==2
.3125 C-Sensor 2==28.5000 C-Sensor 1==28.3125 C-Sensor 2==28.4375 C-Sensor 1=
28.3125 C-Sensor 2==28.4375 C-Sensor 1==28.3125 C-Sensor 2==28.4375 C-Sensor
==28.3125 C-Sensor 2==28.4375 C-Sensor 1==28.3125 C-Sensor 2==28.4375 C-Sen
or 1==28.3125 C-Sensor 2==28.3750 C-Sensor 1==28.3125 C-Sensor 2==28.3750 C-S
ensor 1==28.3125 C-Sensor 2==28.3750 C-Sensor 1==28.3125 C-Sensor 2==28.3750 C
Sensor 1==28.3125 C-Sensor 2==28.3750 C-Sensor 1==28.3125 C-Sensor 2==28.3125

```

Gambar 4.9 Pengujian *time response* untuk 500 ms

$T_1 = 28.3125^{\circ}\text{C}$ (merupakan suhu *ambient*)

$T_2 = 29.6250^{\circ}\text{C}$

$$AT = T_2 - T_1$$

$$AT = 29.6250 - 28.3125$$

$$AT = 1.3125^{\circ}\text{C}$$

Untuk mencapai suhu dengan nilai 29.6250°C dibutuhkan waktu 35 detik. Dari data tersebut maka:

$$\begin{aligned} \text{Time Respon} &= \frac{1.4375}{35} \\ &= 0.041^{\circ}\text{C} / \text{det} \end{aligned}$$

Pengujian kedua ini dilakukan dengan merubah waktu tunda program dari 500ms menjadi 5 ms. Hal ini dilakukan dengan tujuan memperbaiki kecepatan respon dari perubahan suhu yang terjadi.

```

1=+29.7500:C+Sensor 2=+30.2500:C**Sensor 1=+29.7500:C+Sensor 2=+30.2500:C**Senso
r 1=+29.7500:C+Sensor 2=+30.1875:C**Sensor 1=+29.7500:C+Sensor 2=+30.1875:C**Sen
sor 1=+29.7500:C+Sensor 2=+30.1875:C**Sensor 1=+29.7500:C+Sensor 2=+30.1875:C**S
ensor 1=+29.7500:C+Sensor 2=+30.1875:C**Sensor 1=+29.7500:C+Sensor 2=+30.1250:C+
**Sensor 1=+29.7500:C+Sensor 2=+30.1250:C**Sensor 1=+29.7500:C+Sensor 2=+30.1250
C**Sensor 1=+29.7500:C+Sensor 2=+30.0625:C0**Sensor 1=+29.7500:C+Sensor 2=+30.06
25:C0**Sensor 1=+29.7500:C+Sensor 2=+30.0625:C0**Sensor 1=+29.7500:C+Sensor 2=+3
0.0625:C0**Sensor 1=+29.7500:C+Sensor 2=+30.0625:C0**Sensor 1=+29.7500:C+Sensor
2=+30.0000:C000**Sensor 1=+29.7500:C+Sensor 2=+30.0000:C000**Sensor 1=+29.7500:C
+Sensor 2=+30.0000:C000**Sensor 1=+29.7500:C+Sensor 2=+30.0000:C000**Sensor 1=+2
9.7500:C+Sensor 2=+30.0000:C000**Sensor 1=+29.7500:C+Sensor 2=+29.9375:C**Sensor
1=+29.7500:C+Sensor 2=+29.9375:C**Sensor 1=+29.7500:C+Sensor 2=+29.9375:C**Sens
or 1=+29.7500:C+Sensor 2=+29.9375:C**Sensor 1=+29.7500:C+Sensor 2=+29.9375:C**Se
nsor 1=+29.7500:C+Sensor 2=+29.8750:C**Sensor 1=+29.7500:C+Sensor 2=+29.8750:C**
Sensor 1=+29.7500:C+Sensor 2=+29.8750:C**Sensor 1=+29.6875:C+Sensor 2=+29.8750:C
**Sensor 1=+29.6875:C+Sensor 2=+29.8750:C**Sensor 1=+29.6875:C+Sensor 2=+29.8750
C**Sensor 1=+29.6875:C+Sensor 2=+29.8750:C**Sensor 1=+29.6875:C+Sensor 2=+29.8750
C**Sensor 1=+29.6875:C+Sensor 2=+29.8125:C**Sensor 1=+29.6875:C+Sensor 2=+29.81
25:C**Sensor 1=+29.6875:C+Sensor 2=+29.8125:C**Sensor 1=+29.6875:C+Sensor 2=+29.
8125:C**Sensor 1=+29.7500:C+Sensor 2=+29.8125:C**Sensor 1=+29.6875:C+Sensor 2=+2
9.8125:C**Sensor 1=+29.6875:C+Sensor 2=+29.8125:C**Sensor 1=+29.6875:C+Sensor 2=
+29.8125:C**Sensor 1=+29.6875:C+Sensor 2=+29.8125:C**Sensor 1=+29.6875:C+Sensor
2=+29.8125:C**Sensor 1=+29.6875:C+Sensor 2=+29.8125:C**Sensor 1=+29.6875:C+Senso
r 2=+29.8125:C**Sensor 1=+29.6875:C+Sensor 2=+29.7500:C**Sensor 1=+29.7500:C+Sen
sor 2=+29.7500:C**

```

Gambar 4.10 Pengujian *time response* untuk 5 ms

$T_1 = 29.7500^{\circ}\text{C}$ (merupakan suhu *ambient*)

$T_2 = 30.2500^{\circ}\text{C}$

$$AT = T_2 - T_1$$

$$AT = 30.2500 - 29.7500$$

$$AT = 0.5^{\circ}\text{C}$$

Waktu yang dibutuhkan oleh sensor 2 (T_2) untuk mencapai suhu dengan nilai 29.7500°C adalah 6 detik. Dari data tersebut maka diperoleh:

$$\begin{aligned} \text{Time Respon} &= \frac{0.5}{6} \\ &= 0.0833^{\circ}\text{C} / \text{det} \end{aligned}$$

Dari data dan perhitungan yang telah dilakukan diatas kita bisa melihat bahwa *time respon* dari pengukuran suhu yang dilakukan oleh sistem dengan waktu

tunda 500ms adalah $0.041 \text{ } ^\circ\text{C} / \text{det}$. Sedangkan time respon dari pengukuran suhu yang dilakukan sistem dengan waktu tunda 5 ms adalah $0.0833 \text{ } ^\circ\text{C}$. Dari pengujian yang telah dilakukan maka dapat dilihat bahwa:

- Penggunaan waktu tunda sebesar 500 ms dan 5 ms dalam pembuatan program mikrokontroler berpengaruh terhadap *time response*.
- Resolusi yang digunakan sebesar 12 bit turut berpengaruh kepada keakuratan data yang diterima oleh sensor suhu DS 18B20 yang berakibat makin lamanya pembacaan waktu yang dilakukan oleh sensor suhu.
- Adanya waktu tunda yang terjadi dalam pembacaan suhu yang dilakukan oleh sensor suhu DS 18B20 berdasarkan data yang tertera pada datasheet sebesar 750 ms.
- Waktu tunda yang terjadi pada waktu terjadi komunikasi antara PC (personal computer) dengan mikrokontroler ATmega 8535 yang dihubungkan melalui RS232.

Hal ini berpengaruh kepada pembacaan sensor suhu dimana pembacaan dari sensor suhu menjadi akurat dengan tingkat ketelitian yang hampir sama dengan thermometer digital.

BAB 5

KESIMPULAN

Setelah melakukan pengujian dan analisis terhadap alat yang telah dirancang maka dapat disimpulkan beberapa poin:

1. Komunikasi antara sensor suhu DS 18B20 dan mikrokontroler ATmega 8535 harus dilakukan dengan sangat baik sehingga terjadi kesesuaian respon diantara kedua sistem ini.
2. Tingkat *error* rata-rata sensor 1 didapatkan nilai 1.25 % jika dibandingkan dengan hasil pengukuran suhu dengan menggunakan thermometer digital
3. Tingkat *error* rata-rata sensor 2 didapatkan nilai 1.24 % jika dibandingkan dengan hasil pengukuran suhu dengan menggunakan thermometer digital.
4. Tingkat error rata-rata dari sistem monitoring suhu adalah 0.12 % jika dibandingkan dengan hasil pengukuran suhu dengan menggunakan thermometer digital
5. Dengan mempercepat waktu tunda yang digunakan oleh program mikrokontroler dari 500 ms menjadi 5 ms, maka *time* respon juga akan menjadi lebih cepat sebesar $0.0423 \text{ }^{\circ}\text{C} / \text{det}$

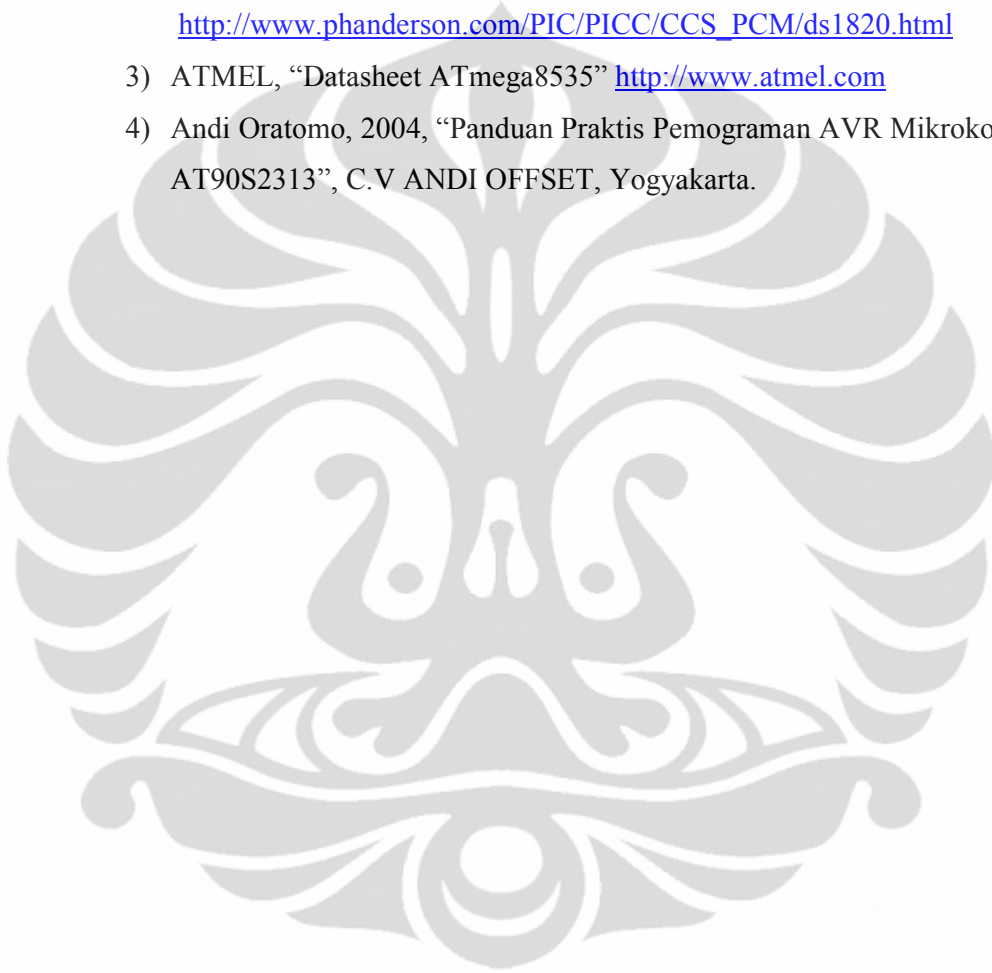
DAFTAR ACUAN

- 1) Andi Oratomo, 2004, “Panduan Praktis Pemograman AVR Mikrokontroler AT90S2313”, C.V ANDI OFFSET, Yogyakarta.
- 2) Datasheet DS 18B20
http://www.phanderson.com/PIC/PICC/CCS_PCM/ds1820.html



DAFTAR PUSTAKA

- 1) Whardana, Lingga., Belajar Sendiri Mikrokontroller AVR Seri ATmega 8535 Simulasi,Hardware, dan Aplikasi,ANDI OFFSET, Jogjakarta, 2006.
- 2) Datasheet DS 18B20
http://www.phanderson.com/PIC/PICC/CCS_PCM/ds1820.html
- 3) ATMEL, “Datasheet ATmega8535” <http://www.atmel.com>
- 4) Andi Oratomo, 2004, “Panduan Praktis Pemograman AVR Mikrokontroler AT90S2313”, C.V ANDI OFFSET, Yogyakarta.





LAMPIRAN

Program Mikrokontroler

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <avr/eeprom.h>
#include <string.h>
#include "uart.h"
#include "onewire.h"
#include "ds18x20.h"
#include "delay.h"
#include "lcd_lib.h"
#define BAUD 9600
// 2400 for 1MHz and 2MHz internal RC
// #define BAUD 2400
#define MAXSENSORS 5
uint8_t gSensorIDs[MAXSENSORS][OW_ROMCODE_SIZE];
uint8_t search_sensors(void);
void uart_put_temp(const uint8_t subzero, uint8_t cel, uint8_t cel_frac_bits);
/*
#ifdef DS18X20_EEPROMSUPPORT
static void eeprom_test(void)
{ uint8_t sp[DS18X20_SP_SIZE], th, tl;
  uart_puts_P( "\rDS18x20 EEPROM support test for fist sensor\r" );
  // DS18X20_recall_E2(&gSensorIDs[0][0]); // already done at power-on
  DS18X20_read_scratchpad( &gSensorIDs[0][0], sp);
  th = sp[DS18X20_TH_REG];
  tl = sp[DS18X20_TL_REG];
  uart_puts_P( "TH/TL from EEPROM sensor 1 : " );
  uart_puti(th);
  uart_puts_P( " / " );
  uart_puti(tl);
  uart_puts_P( "\r" );
  tl++; th++;
  DS18X20_write_scratchpad( &gSensorIDs[0][0], th, tl, DS18B20_12_BIT);
  uart_puts_P( "TH+1 and TL+1 written to scratchpad\r" );
  DS18X20_copy_scratchpad( DS18X20_POWER_PARASITE, &gSensorIDs[0][0] );
  uart_puts_P( "scratchpad copied to DS18x20 EEPROM\r" );
  DS18X20_recall_E2(&gSensorIDs[0][0]);
  uart_puts_P( "DS18x20 EEPROM copied back to scratchpad\r" );
  DS18X20_read_scratchpad( &gSensorIDs[0][0], sp);
  if ( (th == sp[DS18X20_TH_REG]) && (tl == sp[DS18X20_TL_REG]) )
    uart_puts_P( "TH and TL verified\r" );
  else
    uart_puts_P( "verify failed\r" );
  th = sp[DS18X20_TH_REG];
  tl = sp[DS18X20_TL_REG];
}
```

```

    uart_puts_P( "TH/TL from EEPROM sensor 1 now : " );
    uart_puti(th);
    uart_puts_P( " / " );
    uart_puti(tl);
        uart_puts_P( "\r" );
    }
#endif
*/
int main( void )
{ uint8_t nSensors, i;
  uint8_t subzero, cel, cel_frac_bits;
  delay_ms(2000);
  LCDinit();
  LCDclr();
  uart_init((UART_BAUD_SELECT((BAUD),F_OSC)));
#ifdef OW_ONE_BUS
  ow_set_bus(&PIND,&PORTD,&DDRD,PD6);
#endif
  sei();
  LCDsendCommand(0x80+0);
  //LCDstring("TEMP DS18B20");
  LCDsendChar("A");
  delay_ms(1000);
  // dbg uart_puts_P( "\r" );
  // dbg uart_puti((int) OW_CONF_DELAYOFFSET);
  // dbg uart_puts_P( "\r" );
  nSensors = search_sensors();
  //uart_puti((int) nSensors);
  //uart_puts_P( " DS18X20 Sensor(s) available:\r\n" );
  delay_ms(1000);

  // #ifdef DS18X20_VERBOSE
  // for (i=0; i<nSensors; i++)
  // { uart_puts_P("# in Bus :");
  //   uart_puti((int) i+1);
  //   uart_puts_P(" : ");
  //   DS18X20_show_id_uart( &SensorIDs[i][0], OW_ROMCODE_SIZE );
//OW_ROM_CODE_SIZE=8
  //   uart_puts_P( "\r\n" );
  // }
  // #endif

  // for (i=0; i<nSensors; i++)
  // { uart_puts_P("Sensor# ");
  //   uart_puti((int) i+1);
  //   uart_puts_P(" is a ");

```

```

// if ( gSensorIDs[i][0] == DS18S20_ID)
//   uart_puts_P("DS18S20/DS1820");
// else uart_puts_P("DS18B20");
//   uart_puts_P(" which is ");
//
// if ( DS18X20_get_power_status( &gSensorIDs[i][0] ) == DS18X20_POWER_PARASITE
// )
//   uart_puts_P( "parasite" );
// else uart_puts_P( "externally" );
//   uart_puts_P( " powered\r" );
// }
// #ifdef DS18X20_EEPROMSUPPORT
// if (nSensors>0)
// { eeprom_test();
// }
// #endif
if ( nSensors == 1 )
{ uart_puts_P( "\rThere is only one sensor -> Demo of \"read_meas_single\":\r" );
i = gSensorIDs[0][0]; // family-code for conversion-routine
DS18X20_start_meas( DS18X20_POWER_PARASITE, NULL );
delay_ms(DS18B20_TCONV_12BIT);
DS18X20_read_meas_single(i, &subzero, &cel, &cel_frac_bits);
uart_put_temp(subzero, cel, cel_frac_bits);
uart_puts_P("\r");
}

for(;;)
{ // main loop
//   uart_puts_P( "\rConvert_T and Read Sensor by Sensor (reverse order)\r" );
//   for ( i=nSensors; i>0; i-- )
//     { if ( DS18X20_start_meas( DS18X20_POWER_PARASITE, &gSensorIDs[i-1][0] ) ==
DS18X20_OK )
//       { delay_ms(DS18B20_TCONV_12BIT);
//         uart_puts_P("Sensor# ");
//         uart_puti((int) i);
//         uart_puts_P(" = ");

//           if ( DS18X20_read_meas( &gSensorIDs[i-1][0], &subzero, &cel, &cel_frac_bits) ==
DS18X20_OK )
//             { uart_put_temp(subzero, cel, cel_frac_bits);
//               }
//             else uart_puts_P("CRC Error (lost connection?");
//               uart_puts_P("\r");
//             }
//           else uart_puts_P("Start meas. failed (short circuit?");
//             }
}
}

```

```

//uart_puts_P( "\rConvert_T for all Sensors and Read Sensor by Sensor\r\n" );
if ( DS18X20_start_meas( DS18X20_POWER_PARASITE, NULL ) == DS18X20_OK)
{ delay_ms(DS18B20_TCONV_12BIT);
  for ( i=0; i<nSensors; i++ )
  { uart_puts_P("Sensor ");
    uart_puti((int) i+1);
    uart_puts_P("=");
    if ( DS18X20_read_meas( &gSensorIDs[i][0], &subzero, &cel, &cel_frac_bits) ==
DS18X20_OK )
    { uart_put_temp(subzero, cel, cel_frac_bits);
      }
    else uart_puts_P("CRC Error (lost connection?)");
    uart_puts_P("+");
  }
}
else uart_puts_P("Start meas. failed (short circuit?)");

#ifdef DS18X20_VERBOSE
// // all devices:
// uart_puts_P( "\rVerbose output\r\n" );
// DS18X20_start_meas( DS18X20_POWER_PARASITE, NULL );
// delay_ms(DS18B20_TCONV_12BIT);
// DS18X20_read_meas_all_verbose();
// uart_puts_P("\r\n");
#endif
uart_puts_P("*");
delay_ms(5000);
}
}

uint8_t search_sensors(void)
{ uint8_t i;
  uint8_t id[OW_ROMCODE_SIZE];
  uint8_t diff, nSensors;

//uart_puts_P( "\rScanning Bus for DS18X20\r" );

nSensors = 0;

for( diff = OW_SEARCH_FIRST; diff != OW_LAST_DEVICE && nSensors <
MAXSENSORS ; )
{ DS18X20_find_sensor( &diff, &id[0] );

if( diff == OW_PRESENCE_ERR )

```

```

    { uart_puts_P( "No Sensor found\r" );
      break;
    }
    if( diff == OW_DATA_ERR )
    { uart_puts_P( "Bus Error\r" );
      break;
    }
    for (i=0;i<OW_ROMCODE_SIZE;i++)
        gSensorIDs[nSensors][i]=id[i];

    nSensors++;
}
return nSensors;
}

void uart_put_temp(const uint8_t subzero, uint8_t cel, uint8_t cel_frac_bits)
{ uint8_t buffer[sizeof(int)*8+1];
  uint16_t decicelsius;
  uint8_t i, j;

  uart_putc((subzero)?'-':'+');
  uart_puti((int)cel);
  uart_puts_P(".");
  itoa((cel_frac_bits*DS18X20_FRACCONV),buffer,10);
  j=4-strlen(buffer);
  for (i=0;i<j;i++) uart_puts_P("0")
    ;

  uart_puts(buffer);
  uart_puts_P("°C");
  // "rounding"
  //uart_putc((subzero)?'-':'+');
  //decicelsius = DS18X20_temp_to_decicel(subzero, cel, cel_frac_bits);
  //uart_puti( (int)(decicelsius/10) );
  //uart_puts_P(".");
  //uart_putc( (decicelsius%10) + '0');
  //uart_puts_P("°C");
}

```

Program Visual Basic

```
Dim dataakhir As String
Dim sensor1 As String
Dim sensor2 As String
Dim valsensor1 As String
Dim valsensor2 As String
Dim valsensor11 As Single
Dim valsensor21 As Single
Dim valrata As Single
```

```
Private Sub Form_Load()
With MSComm1
```

```
    'make sure the serial port is not open (by this program)
    If MSComm1.PortOpen Then MSComm1.PortOpen = False
    'set the active serial port
    MSComm1.CommPort = 2
    'set the badurate,parity,databits,stopbits for the connection
    MSComm1.Settings = "9600,N,8,1"
    'set the DRT and RTS flags
    MSComm1.DTREnable = True
    MSComm1.RTSEnable = True
    'enable the oncomm event for every received character
    MSComm1.RThreshold = 1
    'disable the oncomm event for send characters
    MSComm1.SThreshold = 0
    'open the serial port
    MSComm1.PortOpen = True
End With 'MSComm1
```

```
End Sub
```

```
Private Sub MSComm1_OnComm()
'On Error GoTo serialportterima
```

```
=====
' Proses Penerimaan Data Serial
=====
```

```
Select Case MSComm1.CommEvent
```

```
Case comEvSend:
```

```
Case comEvReceive:
```

```
    dataakhir = dataakhir & MSComm1.Input
    If Right(dataakhir, 1) = "*" Then
    Text4.Text = ""
    Text4.Text = dataakhir
```

```
    sensor1 = Mid(dataakhir, 10, 8)
    sensor2 = Mid(dataakhir, 30, 8)
```

```

Text1.Text = sensor1
Text2.Text = sensor2

valsensor1 = Mid(sensor1, 2, 7)
valsensor2 = Mid(sensor2, 2, 7)

Text5.Text = valsensor1
Text6.Text = valsensor2

valsensor11 = CSng(valsensor1)
valsensor21 = CSng(valsensor2)

valrata = (valsensor11 + valsensor21) / 2
Text3.Text = CStr(valrata)

'If dataakhir = "Dataready*" Then
'MsgBox "IC Siap Di Test"
'sh_status.FillColor = vbGreen
'dataakhir = ""
'Exit Sub
'End If

'status_data_mikon = Left(dataakhir, 1)

' If status_data_mikon = "U" Then
' j = 3
'   For i = 1 To 16
'     datastatuspin(i) = Mid(dataakhir, j, 1)
'     j = j + 2
'   Next i
'   Call update_status_pin_on_mikon

' ElseIf status_data_mikon = "T" Then
' result_box.Text = dataakhir
' Call cek_hasilttest_with_database
' End If
dataakhir = ""
End If
End Select
Exit Sub
serialportterima:
MsgBox ("Serial Port Terima Error")
End Sub

```


DS18B20

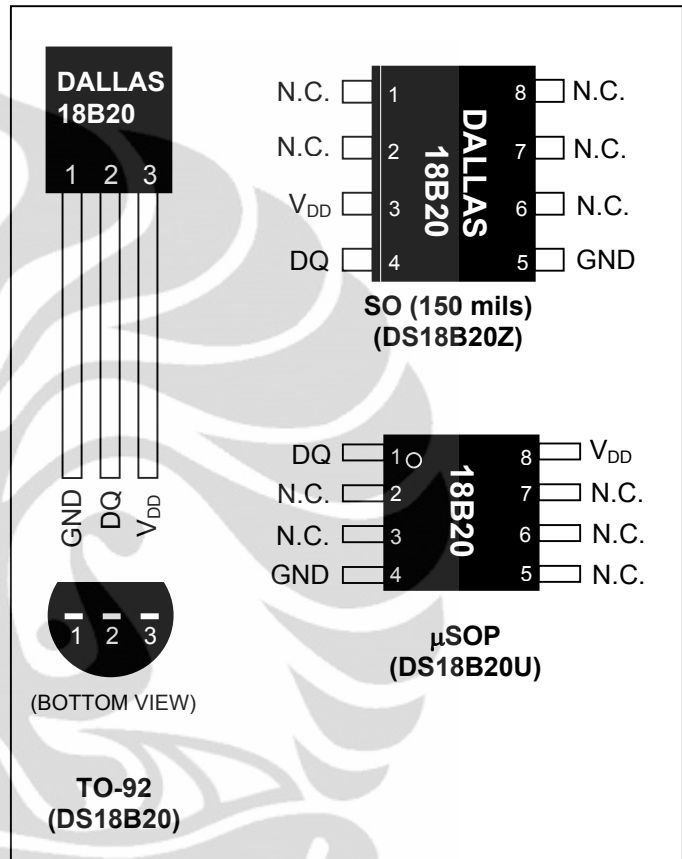
Programmable Resolution 1-Wire Digital Thermometer

www.maxim-ic.com

FEATURES

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Can Be Powered from Data Line; Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C Accuracy from -10°C to +85°C
- Thermometer Resolution is User Selectable from 9 to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)
- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Available in 8-Pin SO (150 mils), 8-Pin μ SOP, and 3-Pin TO-92 Packages
- Software Compatible with the DS1822
- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System

PIN CONFIGURATIONS



DESCRIPTION

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to +125°C and is accurate to ±0.5°C over the range of -10°C to +85°C. In addition, the DS18B20 can derive power directly from the data line (“parasite power”), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

ORDERING INFORMATION

PART	TEMP RANGE	PIN-PACKAGE	TOP MARK
DS18B20	-55°C to +125°C	3 TO-92	18B20
DS18B20+	-55°C to +125°C	3 TO-92	18B20
DS18B20/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20-SL/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20-SL+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20U	-55°C to +125°C	8 μ SOP	18B20
DS18B20U+	-55°C to +125°C	8 μ SOP	18B20
DS18B20U/T&R	-55°C to +125°C	8 μ SOP (3000 Piece)	18B20
DS18B20U+T&R	-55°C to +125°C	8 μ SOP (3000 Piece)	18B20
DS18B20Z	-55°C to +125°C	8 SO	DS18B20
DS18B20Z+	-55°C to +125°C	8 SO	DS18B20
DS18B20Z/T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20
DS18B20Z+T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20

+Denotes a lead-free package. A "+" will appear on the top mark of lead-free packages.

T&R = Tape and reel.

*TO-92 packages in tape and reel can be ordered with straight or formed leads. Choose "SL" for straight leads. Bulk TO-92 orders are straight leads only.

PIN DESCRIPTION

PIN			NAME	FUNCTION
SO	μ SOP	TO-92		
1, 2, 6, 7, 8	2, 3, 5, 6, 7	—	N.C.	No Connection
3	8	3	V _{DD}	Optional V _{DD} . V _{DD} must be grounded for operation in parasite power mode.
4	1	2	DQ	Data Input/Output. Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see the <i>Powering the DS18B20</i> section.)
5	4	1	GND	Ground

OVERVIEW

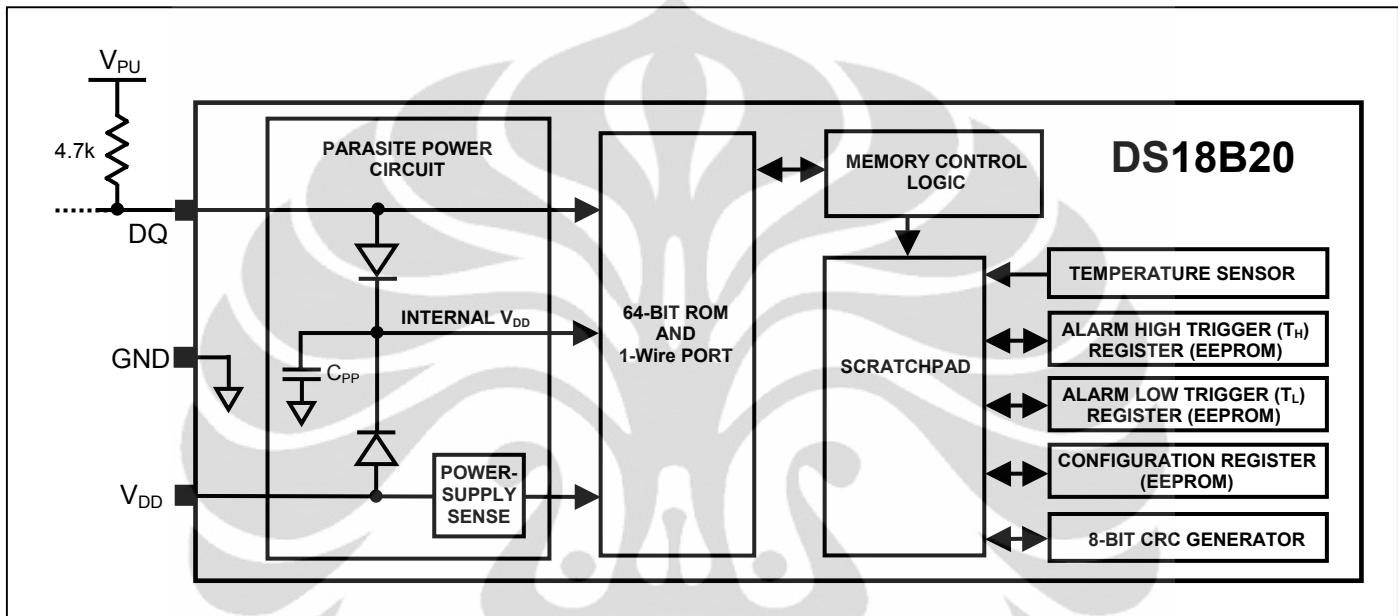
Figure 1 shows a block diagram of the DS18B20, and pin descriptions are given in the *Pin Description* table. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T_H and T_L) and the 1-byte configuration register. The configuration register allows the user to set the resolution of the temperature-to-digital conversion to 9, 10, 11, or 12 bits. The T_H, T_L, and configuration registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18B20 uses Maxim's exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18B20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one

bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and “time slots,” is covered in the *1-Wire Bus System* section.

Another feature of the DS18B20 is the ability to operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor via the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C_{PP}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as “parasite power.” As an alternative, the DS18B20 may also be powered by an external supply on V_{DD} .

Figure 1. DS18B20 Block Diagram



OPERATION—MEASURING TEMPERATURE

The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the temperature sensor is user-configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5°C , 0.25°C , 0.125°C , and 0.0625°C , respectively. The default resolution at power-up is 12-bit. The DS18B20 powers up in a low-power idle state. To initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its idle state. If the DS18B20 is powered by an external supply, the master can issue “read time slots” (see the *1-Wire Bus System* section) after the Convert T command and the DS18B20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the DS18B20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the *Powering the DS18B20* section.

The DS18B20 output temperature data is calibrated in degrees Celsius; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two’s complement number in the temperature register (see Figure 2). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers $S = 0$ and for negative numbers $S = 1$. If the DS18B20 is configured for 12-bit resolution, all bits in the temperature register will contain valid data. For 11-bit resolution, bit 0 is undefined. For 10-bit resolution, bits 1 and 0 are undefined, and for 9-bit resolution bits 2, 1, and 0 are undefined. Table 1 gives examples of digital output data and the corresponding temperature reading for 12-bit resolution conversions.

Figure 2. Temperature Register Format

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	2^6	2^5	2^4

S = SIGN

Table 1. Temperature/Data Relationship

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

*The power-on reset value of the temperature register is +85°C.

OPERATION—ALARM SIGNALING

After the DS18B20 performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte T_H and T_L registers (see Figure 3). The sign bit (S) indicates if the value is positive or negative: for positive numbers $S = 0$ and for negative numbers $S = 1$. The T_H and T_L registers are nonvolatile (EEPROM) so they will retain data when the device is powered down. T_H and T_L can be accessed through bytes 2 and 3 of the scratchpad as explained in the *Memory* section.

Figure 3. T_H and T_L Register Format

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
S	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Only bits 11 through 4 of the temperature register are used in the T_H and T_L comparison since T_H and T_L are 8-bit registers. If the measured temperature is lower than or equal to T_L or higher than or equal to T_H , an alarm condition exists and an alarm flag is set inside the DS18B20. This flag is updated after every temperature measurement; therefore, if the alarm condition goes away, the flag will be turned off after the next temperature conversion.

The master device can check the alarm flag status of all DS18B20s on the bus by issuing an Alarm Search [ECh] command. Any DS18B20s with a set alarm flag will respond to the command, so the master can determine exactly which DS18B20s have experienced an alarm condition. If an alarm condition exists and the T_H or T_L settings have changed, another temperature conversion should be done to validate the alarm condition.

POWERING THE DS18B20

The DS18B20 can be powered by an external supply on the V_{DD} pin, or it can operate in “parasite power” mode, which allows the DS18B20 to function without a local external supply. Parasite power is very useful for applications that require remote temperature sensing or that are very space constrained. Figure 1 shows the DS18B20’s parasite-power control circuitry, which “steals” power from the 1-Wire bus via the DQ pin when the bus is high. The stolen charge powers the DS18B20 while the bus is high, and some of the charge is stored on the parasite power capacitor (C_{PP}) to provide power when the bus is low. When the DS18B20 is used in parasite power mode, the V_{DD} pin must be connected to ground.

In parasite power mode, the 1-Wire bus and C_{PP} can provide sufficient current to the DS18B20 for most operations as long as the specified timing and voltage requirements are met (see the *DC Electrical Characteristics* and *AC Electrical Characteristics*). However, when the DS18B20 is performing temperature conversions or copying data from the scratchpad memory to EEPROM, the operating current can be as high as 1.5mA. This current can cause an unacceptable voltage drop across the weak 1-Wire pullup resistor and is more current than can be supplied by C_{PP} . To assure that the DS18B20 has sufficient supply current, it is necessary to provide a strong pullup on the 1-Wire bus whenever temperature conversions are taking place or data is being copied from the scratchpad to EEPROM. This can be accomplished by using a MOSFET to pull the bus directly to the rail as shown in Figure 4. The 1-Wire bus must be switched to the strong pullup within 10 μ s (max) after a Convert T [44h] or Copy Scratchpad [48h] command is issued, and the bus must be held high by the pullup for the duration of the conversion (t_{CONV}) or data transfer ($t_{WR} = 10$ ms). No other activity can take place on the 1-Wire bus while the pullup is enabled.

The DS18B20 can also be powered by the conventional method of connecting an external power supply to the V_{DD} pin, as shown in Figure 5. The advantage of this method is that the MOSFET pullup is not required, and the 1-Wire bus is free to carry other traffic during the temperature conversion time.

The use of parasite power is not recommended for temperatures above +100°C since the DS18B20 may not be able to sustain communications due to the higher leakage currents that can exist at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that the DS18B20 be powered by an external power supply.

In some situations the bus master may not know whether the DS18B20s on the bus are parasite powered or powered by external supplies. The master needs this information to determine if the strong bus pullup should be used during temperature conversions. To get this information, the master can issue a Skip ROM [CCh] command followed by a Read Power Supply [B4h] command followed by a “read time slot”. During the read time slot, parasite powered DS18B20s will pull the bus low, and externally powered DS18B20s will let the bus remain high. If the bus is pulled low, the master knows that it must supply the strong pullup on the 1-Wire bus during temperature conversions.

Figure 4. Supplying the Parasite-Powered DS18B20 During Temperature Conversions

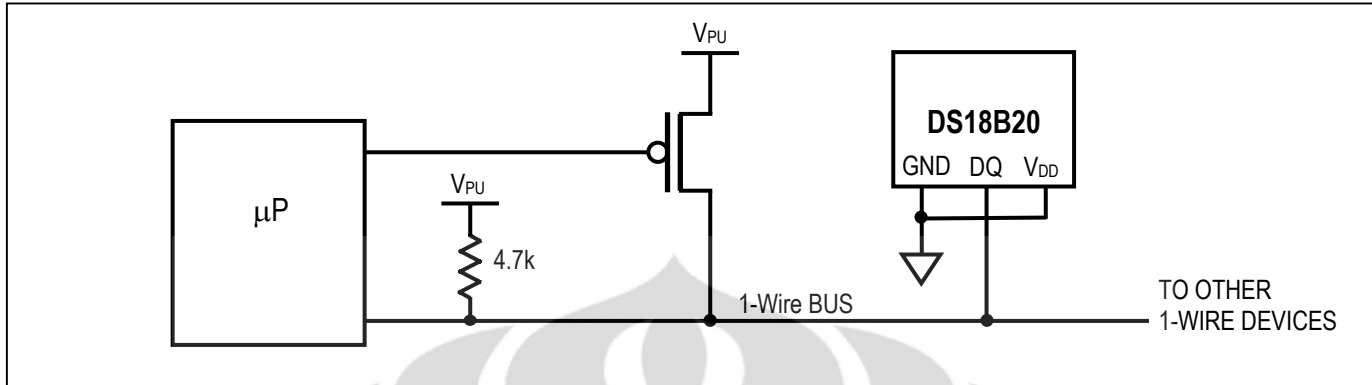
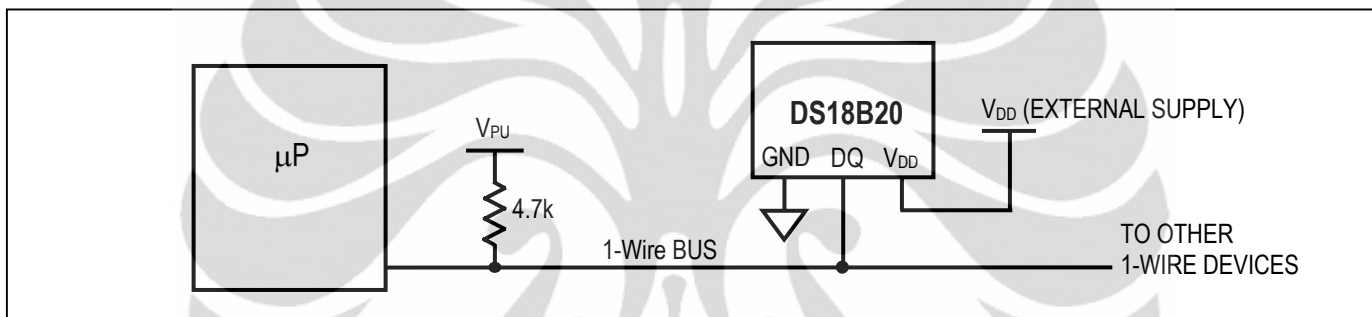


Figure 5. Powering the DS18B20 with an External Supply



64-BIT LASERED ROM CODE

Each DS18B20 contains a unique 64-bit code (see Figure 6) stored in ROM. The least significant 8 bits of the ROM code contain the DS18B20's 1-Wire family code: 28h. The next 48 bits contain a unique serial number. The most significant 8 bits contain a cyclic redundancy check (CRC) byte that is calculated from the first 56 bits of the ROM code. A detailed explanation of the CRC bits is provided in the *CRC Generation* section. The 64-bit ROM code and associated ROM function control logic allow the DS18B20 to operate as a 1-Wire device using the protocol detailed in the *1-Wire Bus System* section.

Figure 6. 64-Bit Lasered ROM Code

8-BIT CRC		48-BIT SERIAL NUMBER		8-BIT FAMILY CODE (28h)	
MSB	LSB	MSB	LSB	MSB	LSB

MEMORY

The DS18B20's memory is organized as shown in Figure 7. The memory consists of an SRAM scratchpad with nonvolatile EEPROM storage for the high and low alarm trigger registers (T_H and T_L) and configuration register. Note that if the DS18B20 alarm function is not used, the T_H and T_L registers can serve as general-purpose memory. All memory commands are described in detail in the *DS18B20 Function Commands* section.

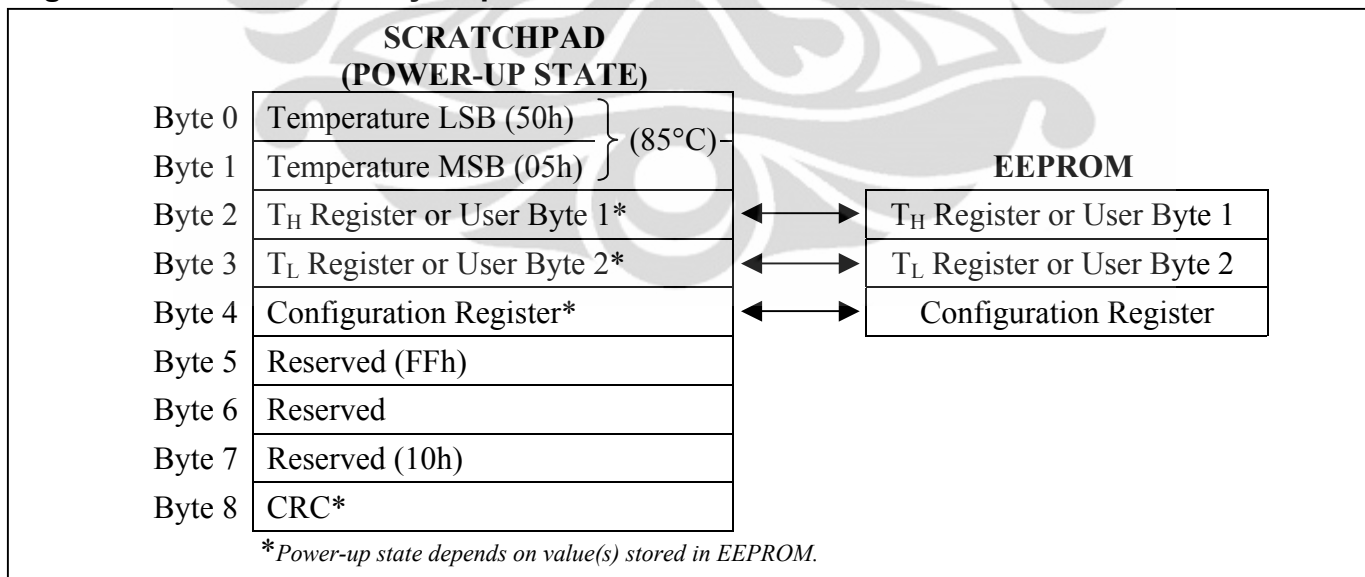
Byte 0 and byte 1 of the scratchpad contain the LSB and the MSB of the temperature register, respectively. These bytes are read-only. Bytes 2 and 3 provide access to T_H and T_L registers. Byte 4 contains the configuration register data, which is explained in detail in the *Configuration Register* section. Bytes 5, 6, and 7 are reserved for internal use by the device and cannot be overwritten.

Byte 8 of the scratchpad is read-only and contains the CRC code for bytes 0 through 7 of the scratchpad. The DS18B20 generates this CRC using the method described in the *CRC Generation* section.

Data is written to bytes 2, 3, and 4 of the scratchpad using the Write Scratchpad [4Eh] command; the data must be transmitted to the DS18B20 starting with the least significant bit of byte 2. To verify data integrity, the scratchpad can be read (using the Read Scratchpad [BEh] command) after the data is written. When reading the scratchpad, data is transferred over the 1-Wire bus starting with the least significant bit of byte 0. To transfer the T_H , T_L and configuration data from the scratchpad to EEPROM, the master must issue the Copy Scratchpad [48h] command.

Data in the EEPROM registers is retained when the device is powered down; at power-up the EEPROM data is reloaded into the corresponding scratchpad locations. Data can also be reloaded from EEPROM to the scratchpad at any time using the Recall E^2 [B8h] command. The master can issue read time slots following the Recall E^2 command and the DS18B20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done.

Figure 7. DS18B20 Memory Map



CONFIGURATION REGISTER

Byte 4 of the scratchpad memory contains the configuration register, which is organized as illustrated in Figure 8. The user can set the conversion resolution of the DS18B20 using the R0 and R1 bits in this register as shown in Table 2. The power-up default of these bits is R0 = 1 and R1 = 1 (12-bit resolution). Note that there is a direct tradeoff between resolution and conversion time. Bit 7 and bits 0 to 4 in the configuration register are reserved for internal use by the device and cannot be overwritten.

Figure 8. Configuration Register

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0	R1	R0	1	1	1	1	1

Table 2. Thermometer Resolution Configuration

R1	R0	RESOLUTION (BITS)	MAX CONVERSION TIME	
0	0	9	93.75ms	($t_{\text{CONV}}/8$)
0	1	10	187.5ms	($t_{\text{CONV}}/4$)
1	0	11	375ms	($t_{\text{CONV}}/2$)
1	1	12	750ms	(t_{CONV})

CRC GENERATION

CRC bytes are provided as part of the DS18B20's 64-bit ROM code and in the 9th byte of the scratchpad memory. The ROM code CRC is calculated from the first 56 bits of the ROM code and is contained in the most significant byte of the ROM. The scratchpad CRC is calculated from the data stored in the scratchpad, and therefore it changes when the data in the scratchpad changes. The CRCs provide the bus master with a method of data validation when data is read from the DS18B20. To verify that data has been read correctly, the bus master must re-calculate the CRC from the received data and then compare this value to either the ROM code CRC (for ROM reads) or to the scratchpad CRC (for scratchpad reads). If the calculated CRC matches the read CRC, the data has been received error free. The comparison of CRC values and the decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS18B20 that prevents a command sequence from proceeding if the DS18B20 CRC (ROM or scratchpad) does not match the value generated by the bus master.

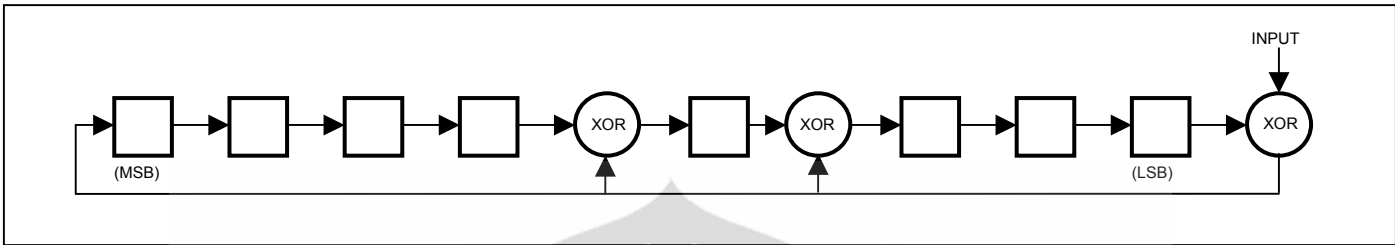
The equivalent polynomial function of the CRC (ROM or scratchpad) is:

$$\text{CRC} = X^8 + X^5 + X^4 + 1$$

The bus master can re-calculate the CRC and compare it to the CRC values from the DS18B20 using the polynomial generator shown in Figure 9. This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0. Starting with the least significant bit of the ROM code or the least significant bit of byte 0 in the scratchpad, one bit at a time should be shifted into the shift register. After shifting in the 56th bit from the ROM or the most significant bit of byte 7 from the scratchpad, the polynomial generator will contain the re-calculated CRC. Next, the 8-bit ROM code or scratchpad CRC from the DS18B20 must be shifted into the circuit. At this point, if the re-calculated CRC was correct, the shift register will contain all 0s. Additional information about the Maxim 1-Wire cyclic redundancy check

is available in *Application Note 27: Understanding and Using Cyclic Redundancy Checks with Maxim iButton Products*.

Figure 9. CRC Generator



1-WIRE BUS SYSTEM

The 1-Wire bus system uses a single bus master to control one or more slave devices. The DS18B20 is always a slave. When there is only one slave on the bus, the system is referred to as a “single-drop” system; the system is “multidrop” if there are multiple slaves on the bus.

All data and commands are transmitted least significant bit first over the 1-Wire bus.

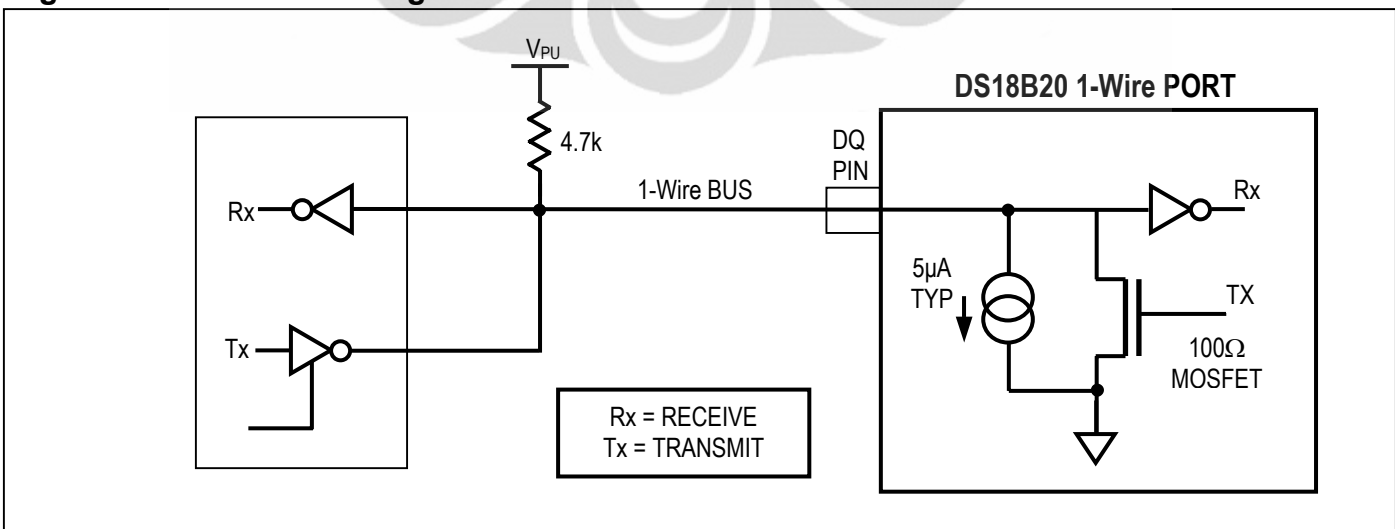
The following discussion of the 1-Wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing).

HARDWARE CONFIGURATION

The 1-Wire bus has by definition only a single data line. Each device (master or slave) interfaces to the data line via an open-drain or 3-state port. This allows each device to “release” the data line when the device is not transmitting data so the bus is available for use by another device. The 1-Wire port of the DS18B20 (the DQ pin) is open drain with an internal circuit equivalent to that shown in Figure 10.

The 1-Wire bus requires an external pullup resistor of approximately 5k Ω ; thus, the idle state for the 1-Wire bus is high. If for any reason a transaction needs to be suspended, the bus **MUST** be left in the idle state if the transaction is to resume. Infinite recovery time can occur between bits so long as the 1-Wire bus is in the inactive (high) state during the recovery period. If the bus is held low for more than 480 μ s, all components on the bus will be reset.

Figure 10. Hardware Configuration



TRANSACTION SEQUENCE

The transaction sequence for accessing the DS18B20 is as follows:

Step 1. Initialization

Step 2. ROM Command (followed by any required data exchange)

Step 3. DS18B20 Function Command (followed by any required data exchange)

It is very important to follow this sequence every time the DS18B20 is accessed, as the DS18B20 will not respond if any steps in the sequence are missing or out of order. Exceptions to this rule are the Search ROM [F0h] and Alarm Search [ECh] commands. After issuing either of these ROM commands, the master must return to Step 1 in the sequence.

INITIALIZATION

All transactions on the 1-Wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s). The presence pulse lets the bus master know that slave devices (such as the DS18B20) are on the bus and are ready to operate. Timing for the reset and presence pulses is detailed in the *1-Wire Signaling* section.

ROM COMMANDS

After the bus master has detected a presence pulse, it can issue a ROM command. These commands operate on the unique 64-bit ROM codes of each slave device and allow the master to single out a specific device if many are present on the 1-Wire bus. These commands also allow the master to determine how many and what types of devices are present on the bus or if any device has experienced an alarm condition. There are five ROM commands, and each command is 8 bits long. The master device must issue an appropriate ROM command before issuing a DS18B20 function command. A flowchart for operation of the ROM commands is shown in Figure 11.

SEARCH ROM [F0h]

When a system is initially powered up, the master must identify the ROM codes of all slave devices on the bus, which allows the master to determine the number of slaves and their device types. The master learns the ROM codes through a process of elimination that requires the master to perform a Search ROM cycle (i.e., Search ROM command followed by data exchange) as many times as necessary to identify all of the slave devices. If there is only one slave on the bus, the simpler Read ROM command (see below) can be used in place of the Search ROM process. For a detailed explanation of the Search ROM procedure, refer to the *iButton*[®] *Book of Standards* at www.maxim-ic.com/ibuttonbook. After every Search ROM cycle, the bus master must return to Step 1 (Initialization) in the transaction sequence.

READ ROM [33h]

This command can only be used when there is one slave on the bus. It allows the bus master to read the slave's 64-bit ROM code without using the Search ROM procedure. If this command is used when there is more than one slave present on the bus, a data collision will occur when all the slaves attempt to respond at the same time.

MATCH ROM [55h]

The match ROM command followed by a 64-bit ROM code sequence allows the bus master to address a specific slave device on a multidrop or single-drop bus. Only the slave that exactly matches the 64-bit ROM code sequence will respond to the function command issued by the master; all other slaves on the bus will wait for a reset pulse.

SKIP ROM [CCh]

The master can use this command to address all devices on the bus simultaneously without sending out any ROM code information. For example, the master can make all DS18B20s on the bus perform simultaneous temperature conversions by issuing a Skip ROM command followed by a Convert T [44h] command.

Note that the Read Scratchpad [BEh] command can follow the Skip ROM command only if there is a single slave device on the bus. In this case, time is saved by allowing the master to read from the slave without sending the device's 64-bit ROM code. A Skip ROM command followed by a Read Scratchpad command will cause a data collision on the bus if there is more than one slave since multiple devices will attempt to transmit data simultaneously.

ALARM SEARCH [ECh]

The operation of this command is identical to the operation of the Search ROM command except that only slaves with a set alarm flag will respond. This command allows the master device to determine if any DS18B20s experienced an alarm condition during the most recent temperature conversion. After every Alarm Search cycle (i.e., Alarm Search command followed by data exchange), the bus master must return to Step 1 (Initialization) in the transaction sequence. See the *Operation—Alarm Signaling* section for an explanation of alarm flag operation.

DS18B20 FUNCTION COMMANDS

After the bus master has used a ROM command to address the DS18B20 with which it wishes to communicate, the master can issue one of the DS18B20 function commands. These commands allow the master to write to and read from the DS18B20's scratchpad memory, initiate temperature conversions and determine the power supply mode. The DS18B20 function commands, which are described below, are summarized in Table 3 and illustrated by the flowchart in Figure 12.

CONVERT T [44h]

This command initiates a single temperature conversion. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its low-power idle state. If the device is being used in parasite power mode, within 10 μ s (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for the duration of the conversion (t_{CONV}) as described in the *Powering the DS18B20* section. If the DS18B20 is powered by an external supply, the master can issue read time slots after the Convert T command and the DS18B20 will respond by transmitting a 0 while the temperature conversion is in progress and a 1 when the conversion is done. In parasite power mode this notification technique cannot be used since the bus is pulled high by the strong pullup during the conversion.

WRITE SCRATCHPAD [4Eh]

This command allows the master to write 3 bytes of data to the DS18B20's scratchpad. The first data byte is written into the T_H register (byte 2 of the scratchpad), the second byte is written into the T_L register (byte 3), and the third byte is written into the configuration register (byte 4). Data must be transmitted least significant bit first. All three bytes **MUST** be written before the master issues a reset, or the data may be corrupted.

READ SCRATCHPAD [BEh]

This command allows the master to read the contents of the scratchpad. The data transfer starts with the least significant bit of byte 0 and continues through the scratchpad until the 9th byte (byte 8 – CRC) is read. The master may issue a reset to terminate reading at any time if only part of the scratchpad data is needed.

COPY SCRATCHPAD [48h]

This command copies the contents of the scratchpad T_H , T_L and configuration registers (bytes 2, 3 and 4) to EEPROM. If the device is being used in parasite power mode, within 10 μ s (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for at least 10ms as described in the *Powering the DS18B20* section.

RECALL E² [B8h]

This command recalls the alarm trigger values (T_H and T_L) and configuration data from EEPROM and places the data in bytes 2, 3, and 4, respectively, in the scratchpad memory. The master device can issue read time slots following the Recall E² command and the DS18B20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done. The recall operation happens automatically at power-up, so valid data is available in the scratchpad as soon as power is applied to the device.

READ POWER SUPPLY [B4h]

The master device issues this command followed by a read time slot to determine if any DS18B20s on the bus are using parasite power. During the read time slot, parasite powered DS18B20s will pull the bus low, and externally powered DS18B20s will let the bus remain high. See the *Powering the DS18B20* section for usage information for this command.

Table 3. DS18B20 Function Command Set

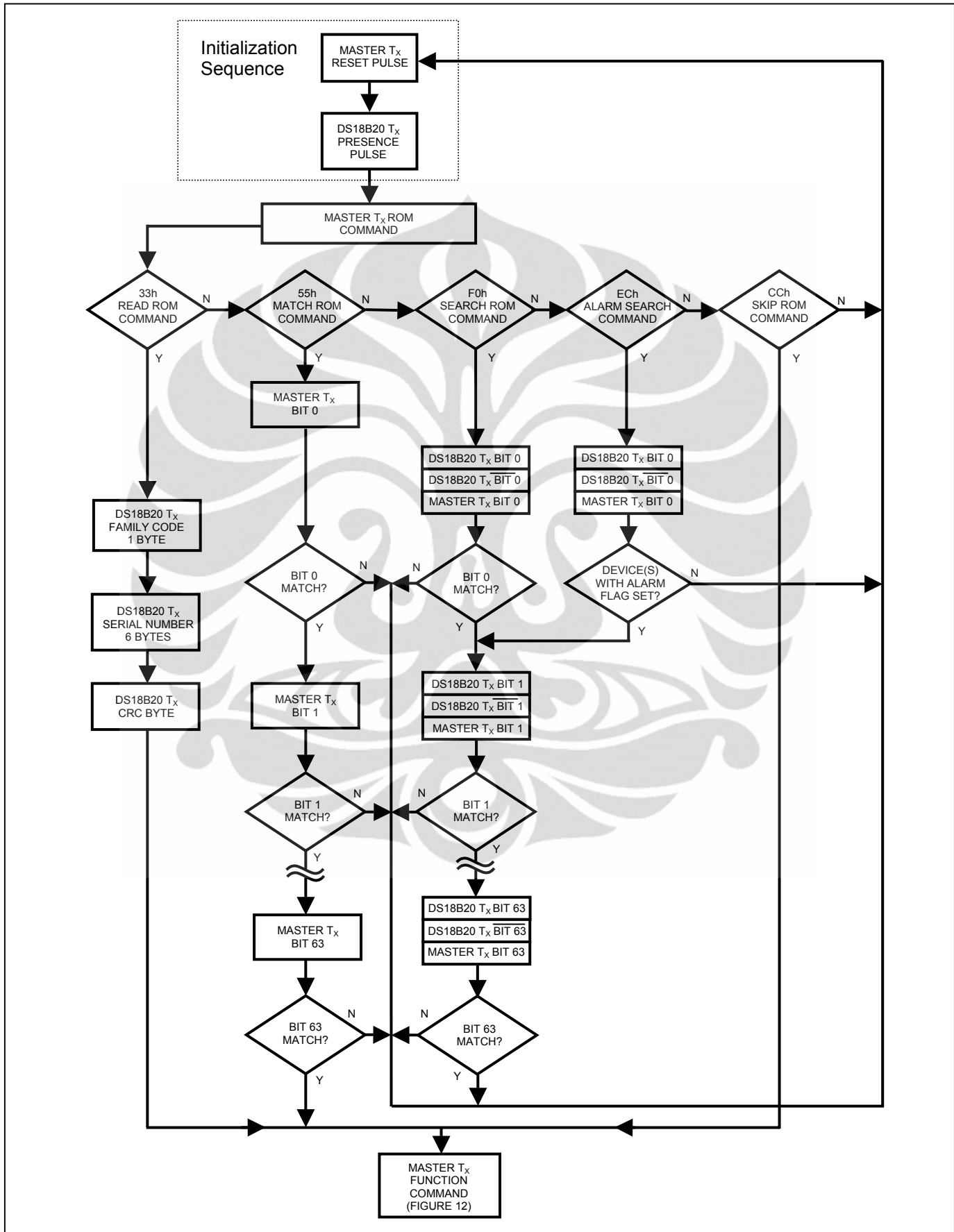
COMMAND	DESCRIPTION	PROTOCOL	1-Wire BUS ACTIVITY AFTER COMMAND IS ISSUED	NOTES
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiates temperature conversion.	44h	DS18B20 transmits conversion status to master (not applicable for parasite-powered DS18B20s).	1
MEMORY COMMANDS				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18B20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2, 3, and 4 (T_H , T_L , and configuration registers).	4Eh	Master transmits 3 data bytes to DS18B20.	3
Copy Scratchpad	Copies T_H , T_L , and configuration register data from the scratchpad to EEPROM.	48h	None	1
Recall E ²	Recalls T_H , T_L , and configuration register data from EEPROM to the scratchpad.	B8h	DS18B20 transmits recall status to master.	
Read Power Supply	Signals DS18B20 power supply mode to the master.	B4h	DS18B20 transmits supply status to master.	

Note 1: For parasite-powered DS18B20s, the master must enable a strong pullup on the 1-Wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.

Note 2: The master can interrupt the transmission of data at any time by issuing a reset.

Note 3: All three bytes must be written before a reset is issued.

Figure 11. ROM Commands Flowchart



1-WIRE SIGNALING

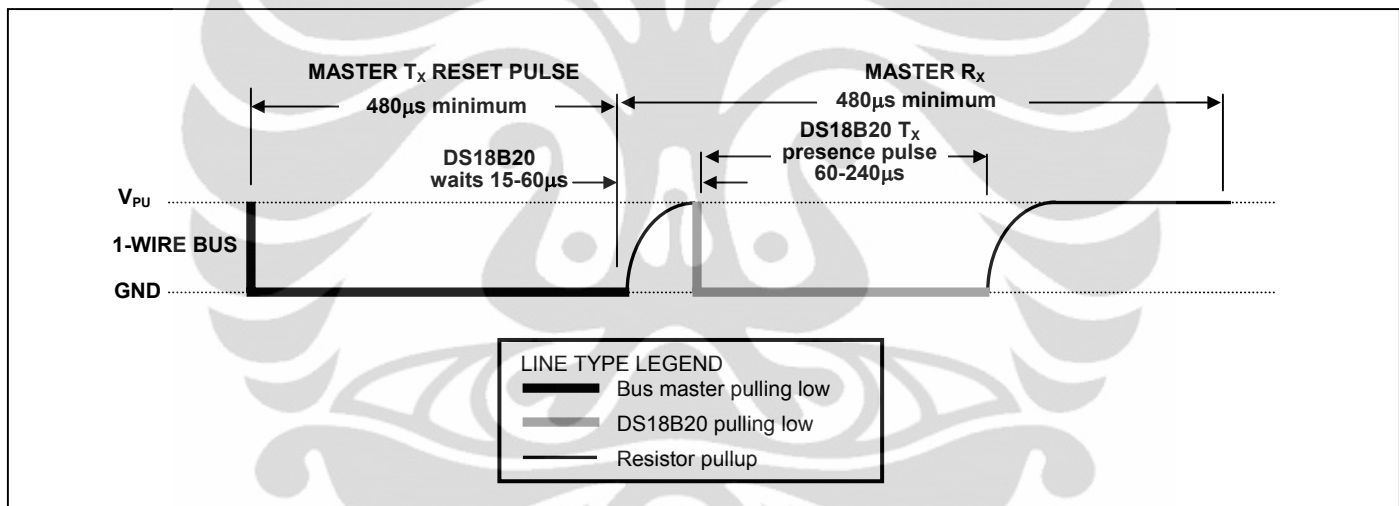
The DS18B20 uses a strict 1-Wire communication protocol to ensure data integrity. Several signal types are defined by this protocol: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. The bus master initiates all these signals, with the exception of the presence pulse.

INITIALIZATION PROCEDURE—RESET AND PRESENCE PULSES

All communication with the DS18B20 begins with an initialization sequence that consists of a reset pulse from the master followed by a presence pulse from the DS18B20. This is illustrated in Figure 13. When the DS18B20 sends the presence pulse in response to the reset, it is indicating to the master that it is on the bus and ready to operate.

During the initialization sequence the bus master transmits (T_X) the reset pulse by pulling the 1-Wire bus low for a minimum of $480\mu\text{s}$. The bus master then releases the bus and goes into receive mode (R_X). When the bus is released, the $5\text{k}\Omega$ pullup resistor pulls the 1-Wire bus high. When the DS18B20 detects this rising edge, it waits $15\mu\text{s}$ to $60\mu\text{s}$ and then transmits a presence pulse by pulling the 1-Wire bus low for $60\mu\text{s}$ to $240\mu\text{s}$.

Figure 13. Initialization Timing



READ/WRITE TIME SLOTS

The bus master writes data to the DS18B20 during write time slots and reads data from the DS18B20 during read time slots. One bit of data is transmitted over the 1-Wire bus per time slot.

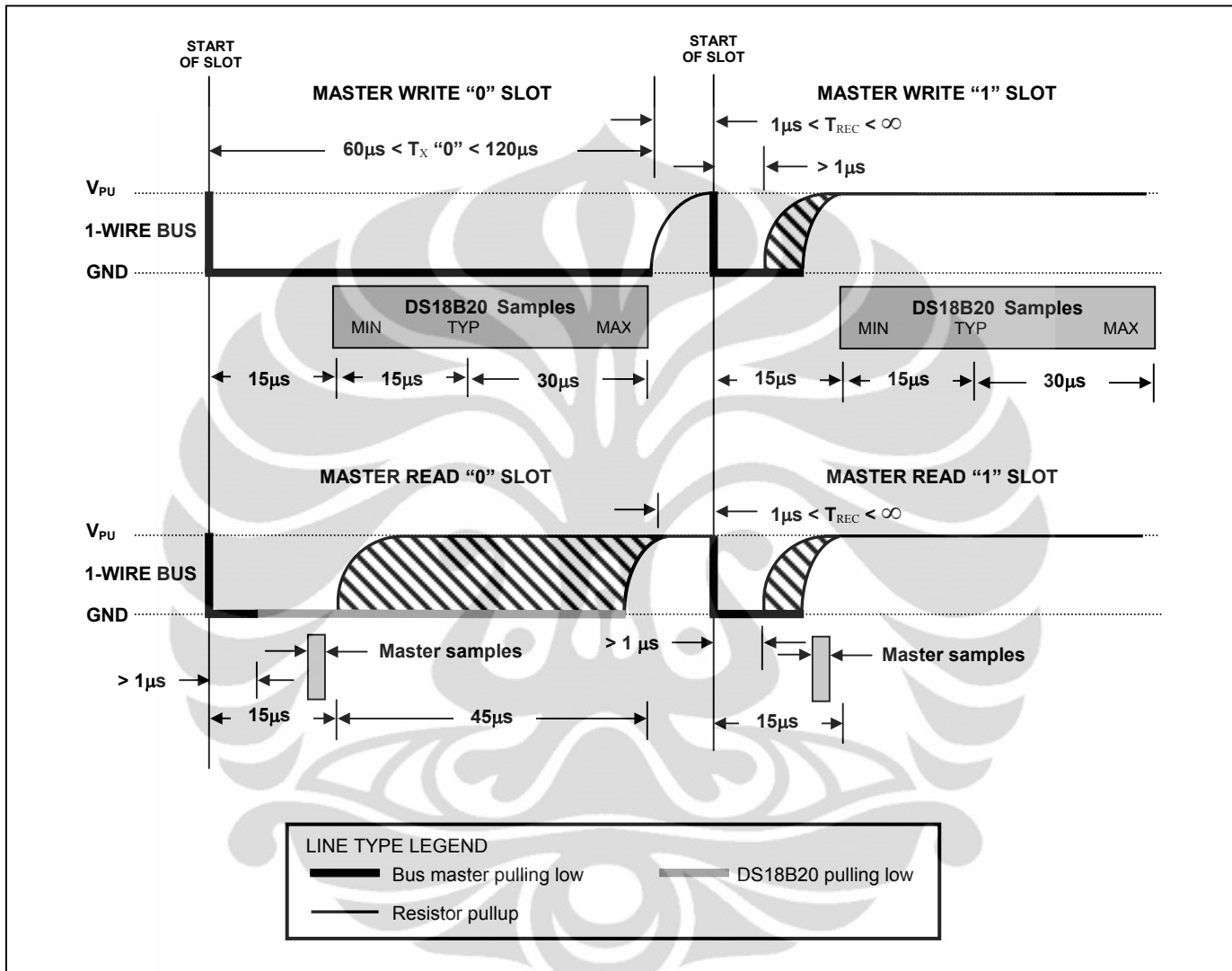
WRITE TIME SLOTS

There are two types of write time slots: “Write 1” time slots and “Write 0” time slots. The bus master uses a Write 1 time slot to write a logic 1 to the DS18B20 and a Write 0 time slot to write a logic 0 to the DS18B20. All write time slots must be a minimum of $60\mu\text{s}$ in duration with a minimum of a $1\mu\text{s}$ recovery time between individual write slots. Both types of write time slots are initiated by the master pulling the 1-Wire bus low (see Figure 14).

To generate a Write 1 time slot, after pulling the 1-Wire bus low, the bus master must release the 1-Wire bus within $15\mu\text{s}$. When the bus is released, the $5\text{k}\Omega$ pullup resistor will pull the bus high. To generate a Write 0 time slot, after pulling the 1-Wire bus low, the bus master must continue to hold the bus low for the duration of the time slot (at least $60\mu\text{s}$).

The DS18B20 samples the 1-Wire bus during a window that lasts from $15\mu\text{s}$ to $60\mu\text{s}$ after the master initiates the write time slot. If the bus is high during the sampling window, a 1 is written to the DS18B20. If the line is low, a 0 is written to the DS18B20.

Figure 14. Read/Write Time Slot Timing Diagram



READ TIME SLOTS

The DS18B20 can only transmit data to the master when the master issues read time slots. Therefore, the master must generate read time slots immediately after issuing a Read Scratchpad [BEh] or Read Power Supply [B4h] command, so that the DS18B20 can provide the requested data. In addition, the master can generate read time slots after issuing Convert T [44h] or Recall E² [B8h] commands to find out the status of the operation as explained in the *DS18B20 Function Commands* section.

All read time slots must be a minimum of $60\mu\text{s}$ in duration with a minimum of a $1\mu\text{s}$ recovery time between slots. A read time slot is initiated by the master device pulling the 1-Wire bus low for a minimum of $1\mu\text{s}$ and then releasing the bus (see Figure 14). After the master initiates the read time slot, the DS18B20 will begin transmitting a 1 or 0 on bus. The DS18B20 transmits a 1 by leaving the bus high and transmits a 0 by pulling the bus low. When transmitting a 0, the DS18B20 will release the bus by the end of the time slot, and the bus will be pulled back to its high idle state by the pullup resistor. Output

data from the DS18B20 is valid for $15\mu\text{s}$ after the falling edge that initiated the read time slot. Therefore, the master must release the bus and then sample the bus state within $15\mu\text{s}$ from the start of the slot.

Figure 15 illustrates that the sum of T_{INIT} , T_{RC} , and T_{SAMPLE} must be less than $15\mu\text{s}$ for a read time slot. Figure 16 shows that system timing margin is maximized by keeping T_{INIT} and T_{RC} as short as possible and by locating the master sample time during read time slots towards the end of the $15\mu\text{s}$ period.

Figure 15. Detailed Master Read 1 Timing

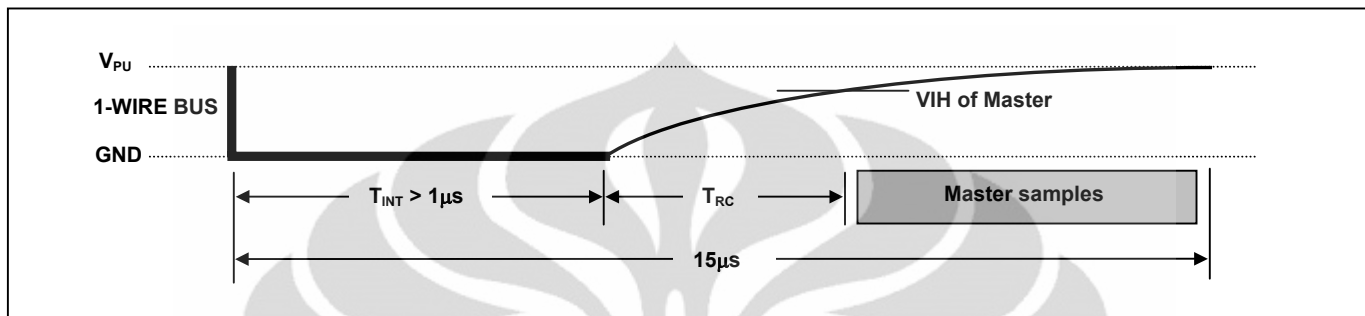
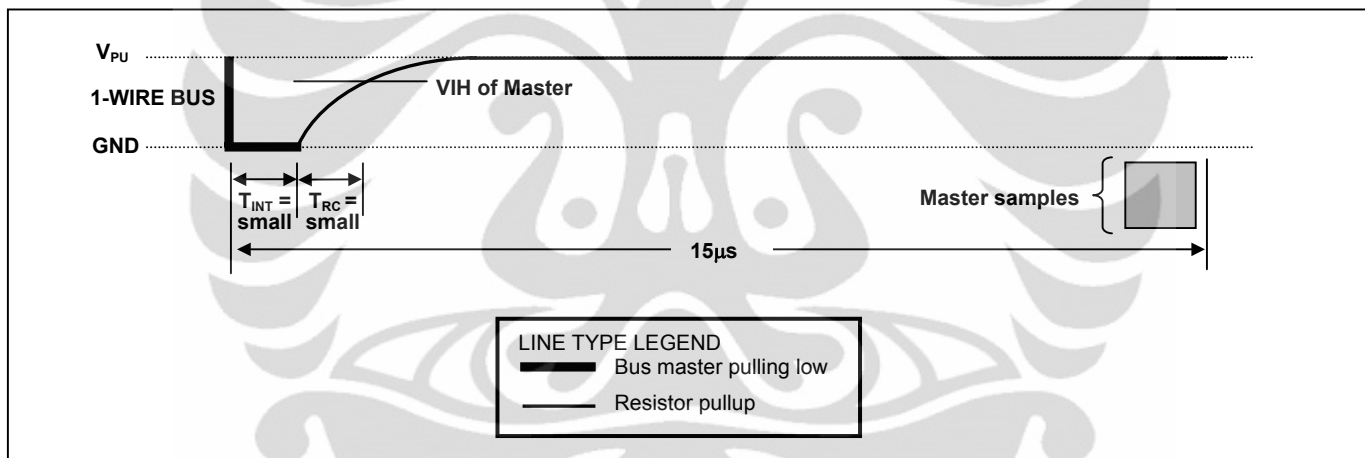


Figure 16. Recommended Master Read 1 Timing



RELATED APPLICATION NOTES

The following application notes can be applied to the DS18B20 and are available on our website at www.maxim-ic.com.

Application Note 27: Understanding and Using Cyclic Redundancy Checks with Maxim iButton Products

Application Note 122: Using Dallas' 1-Wire ICs in 1-Cell Li-Ion Battery Packs with Low-Side N-Channel Safety FETs Master

Application Note 126: 1-Wire Communication Through Software

Application Note 162: Interfacing the DS18x20/DS1822 1-Wire Temperature Sensor in a Microcontroller Environment

Application Note 208: Curve Fitting the Error of a Bandgap-Based Digital Temperature Sensor

Application Note 2420: 1-Wire Communication with a Microchip PICmicro Microcontroller

Application Note 3754: Single-Wire Serial Bus Carries Isolated Power and Data

Sample 1-Wire subroutines that can be used in conjunction with *Application Note 74: Reading and Writing iButtons via Serial Interfaces* can be downloaded from the Maxim website.

DS18B20 OPERATION EXAMPLE 1

In this example there are multiple DS18B20s on the bus and they are using parasite power. The bus master initiates a temperature conversion in a specific DS18B20 and then reads its scratchpad and recalculates the CRC to verify the data.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20s respond with presence pulse.
Tx	55h	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends DS18B20 ROM code.
Tx	44h	Master issues Convert T command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for the duration of the conversion (t_{CONV}).
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20s respond with presence pulse.
Tx	55h	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends DS18B20 ROM code.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.

DS18B20 OPERATION EXAMPLE 2

In this example there is only one DS18B20 on the bus and it is using parasite power. The master writes to the T_H , T_L , and configuration registers in the DS18B20 scratchpad and then reads the scratchpad and recalculates the CRC to verify the data. The master then copies the scratchpad contents to EEPROM.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	4Eh	Master issues Write Scratchpad command.
Tx	3 data bytes	Master sends three data bytes to scratchpad (T_H , T_L , and config).
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	48h	Master issues Copy Scratchpad command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for at least 10ms while copy operation is in progress.

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin Relative to Ground.....	-0.5V to +6.0V
Operating Temperature Range.....	-55°C to +125°C
Storage Temperature Range	-55°C to +125°C
Solder Temperature	Refer to the IPC/JEDEC J-STD-020 Specification.

These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

DC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD}=3.0V$ to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{DD}	Local Power	+3.0		+5.5	V	1
Pullup Supply Voltage	V_{PU}	Parasite Power	+3.0		+5.5	V	1,2
		Local Power	+3.0		V_{DD}		
Thermometer Error	t_{ERR}	-10°C to +85°C			±0.5	°C	3
		-55°C to +125°C			±2		
Input Logic-Low	V_{IL}		-0.3		+0.8	V	1,4,5
Input Logic-High	V_{IH}	Local Power	+2.2		The lower of 5.5 or $V_{DD} + 0.3$	V	1, 6
		Parasite Power	+3.0				
Sink Current	I_L	$V_{IO} = 0.4V$	4.0			mA	1
Standby Current	I_{DDs}			750	1000	nA	7,8
Active Current	I_{DD}	$V_{DD} = 5V$		1	1.5	mA	9
DQ Input Current	I_{DQ}			5		µA	10
Drift				±0.2		°C	11

NOTES:

- All voltages are referenced to ground.
- The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PU} . In order to meet the V_{IH} spec of the DS18B20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: $V_{PU_ACTUAL} = V_{PU_IDEAL} + V_{TRANSISTOR}$.
- See typical performance curve in Figure 17.
- Logic-low voltages are specified at a sink current of 4mA.
- To guarantee a presence pulse under low voltage parasite power conditions, V_{ILMAX} may have to be reduced to as low as 0.5V.
- Logic-high voltages are specified at a source current of 1mA.
- Standby current specified up to +70°C. Standby current typically is 3µA at +125°C.
- To minimize I_{DDs} , DQ should be within the following ranges: $GND \leq DQ \leq GND + 0.3V$ or $V_{DD} - 0.3V \leq DQ \leq V_{DD}$.
- Active current refers to supply current during active temperature conversions or EEPROM writes.
- DQ line is high ("high-Z" state).
- Drift data is based on a 1000-hour stress test at +125°C with $V_{DD} = 5.5V$.

AC ELECTRICAL CHARACTERISTICS—NV MEMORY(-55°C to +100°C; $V_{DD} = 3.0V$ to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
NV Write Cycle Time	t_{WR}			2	10	ms
EEPROM Writes	N_{EEWR}	-55°C to +55°C	50k			writes
EEPROM Data Retention	t_{EEDR}	-55°C to +55°C	10			years

AC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD} = 3.0V$ to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	t_{CONV}	9-bit resolution			93.75	ms	1
		10-bit resolution			187.5		
		11-bit resolution			375		
		12-bit resolution			750		
Time to Strong Pullup On	t_{SPON}	Start Convert T Command Issued			10	μs	
Time Slot	t_{SLOT}		60		120	μs	1
Recovery Time	t_{REC}		1			μs	1
Write 0 Low Time	t_{LOW0}		60		120	μs	1
Write 1 Low Time	t_{LOW1}		1		15	μs	1
Read Data Valid	t_{RDV}				15	μs	1
Reset Time High	t_{RSTH}		480			μs	1
Reset Time Low	t_{RSTL}		480			μs	1,2
Presence-Detect High	t_{PDHIGH}		15		60	μs	1
Presence-Detect Low	t_{PDLow}		60		240	μs	1
Capacitance	$C_{IN/OUT}$				25	pF	

NOTES:

- 1) See the timing diagrams in Figure 18.
- 2) Under parasite power, if $t_{RSTL} > 960\mu s$, a power-on reset may occur.

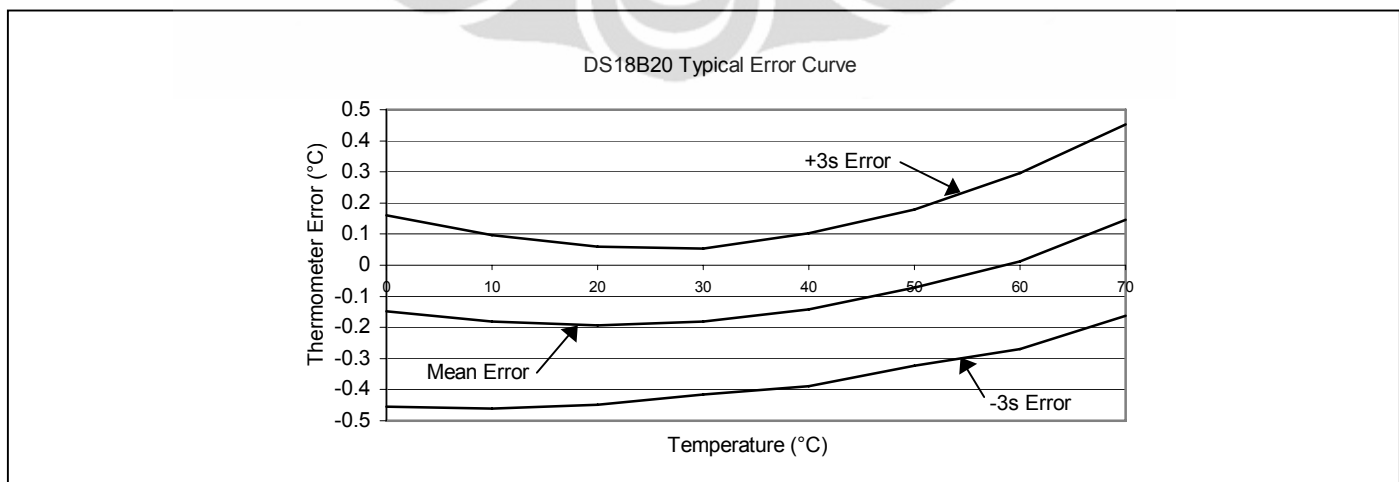
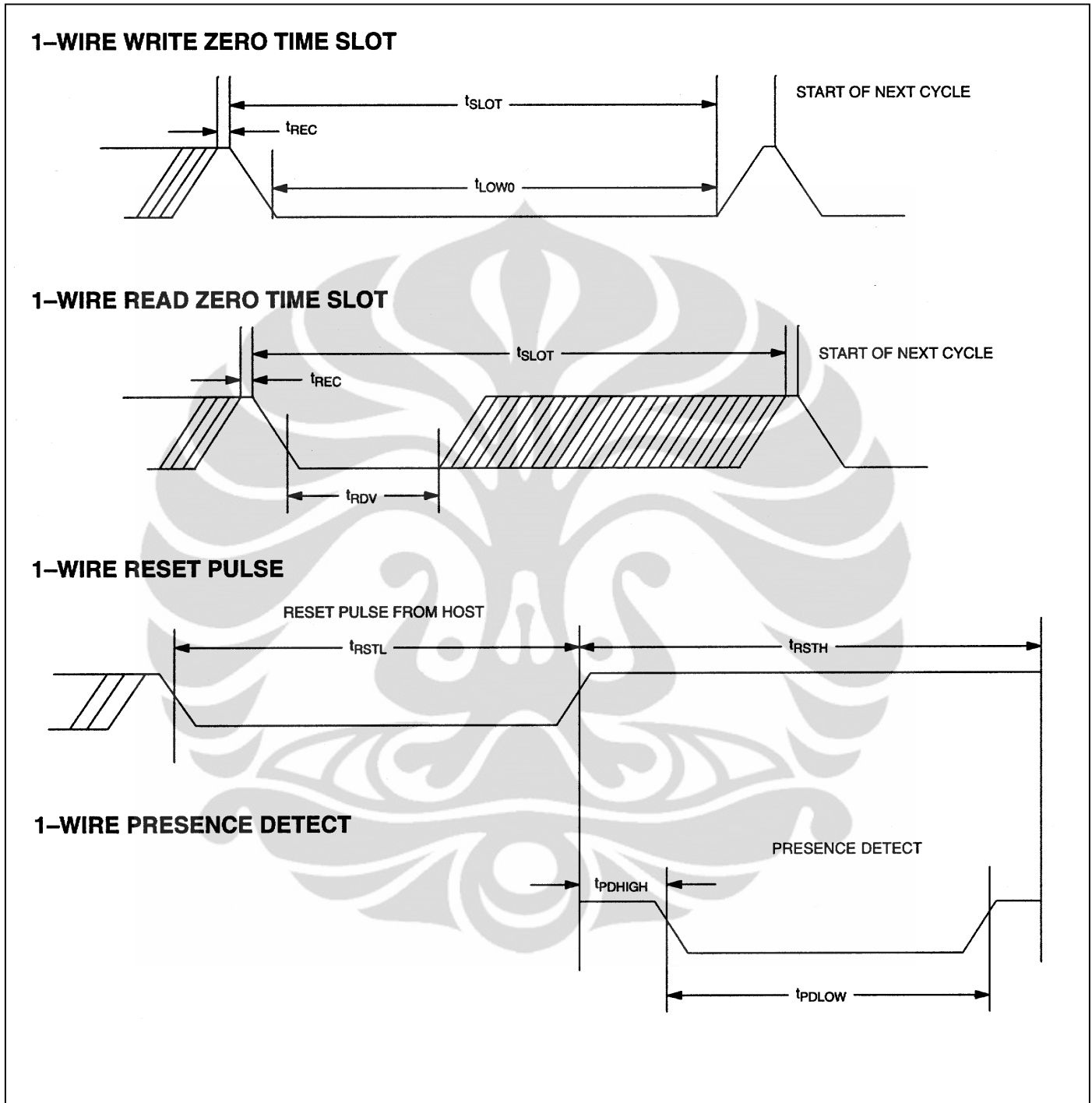
Figure 17. Typical Performance Curve

Figure 18. Timing Diagrams



REVISION HISTORY

REVISION DATE	DESCRIPTION	PAGES CHANGED
030107	In the <i>Absolute Maximum Ratings</i> section, removed the reflow oven temperature value of +220°C. Reference to JEDEC specification for reflow remains.	19
101207	In the <i>Operation—Alarm Signaling</i> section, added “or equal to” in the description for a TH alarm condition	5
	In the <i>Memory</i> section, removed incorrect text describing memory.	7
	In the <i>Configuration Register</i> section, removed incorrect text describing configuration register.	8
042208	In the Ordering Information table, added TO-92 straight-lead packages and included a note that the TO-92 package in tape and reel can be ordered with either formed or straight leads.	2

