

**RANCANG BANGUN SISTEM PENGENDALI SUHU DAN
KELEMBAPAN OTOMATIS MESIN PENETAS TELURAYAM
POLANDIA BERBASIS ARDUINO**



ADHI PUTRA PRATAMA


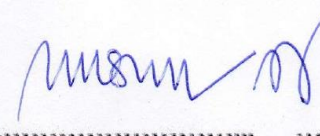
5215122627

**Skripsi Ini Ditulis untuk Memenuhi Sebagian Persyaratan dalam
Memperoleh Gelar Sarjana Pendidikan**


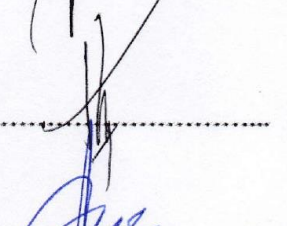
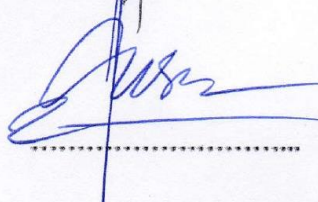
**PROGRAM STUDI PENDIDIKAN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI JAKARTA**

2017

HALAMAN PENGESAHAN

NAMA DOSEN	TANDA TANGAN	TANGGAL
Drs. Jusuf Bintoro, M.T (Dosen Pembimbing 1)		26 - 7 - 2017
Drs. Wisnu Djatmiko, M.T (Dosen Pembimbing 2)		1 - 8 - 2017

PENGESAHAN PANITIA UJIAN SKRIPSI

NAMA DOSEN	TANDA TANGAN	TANGGAL
Dr. Moch. Sukarjo, M.Pd. (Ketua Penguji)		12-7-2017
Drs. Pitoyo Yuliatmojo, MT. (Anggota Penguji)		20-7-2017
Dr. Muhammad Yusro, MT. (Anggota Penguji)		17 Juli 2017

Tanggal Lulus : 11 - 07 - 2017

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis skripsi saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik sarjana, baik di Universitas Negeri Jakarta maupun di perguruan tinggi negeri lain.
2. Karya tulis ini adalah murni gagasan, rumusan dan penelitian saya sendiri dengan arahan dosen pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma yang berlaku di Universitas Negeri Jakarta.

Jakarta, Januari 2017

Yang membuat pernyataan


Adhi Putra Pratama
5215122627

KATA PENGANTAR

Segala puji dan syukur terpanjatkan kehadiran Allah SWT karena telah memberikan rahmat, karunia, hidayah, dan pertolongan-Nya sehingga dapat diselesaikannya skripsi dengan judul “Rancang bangun sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandia berbasis arduino” yang merupakan persyaratan dalam menyelesaikan studi untuk meraih gelar Sarjana Pendidikan di Fakultas Teknik Universitas Negeri Jakarta.

Selama melakukan penelitian ini peneliti ingin mengucapkan terimakasih kepada:

1. Bapak Drs. Jusuf Bintoro, M.T. dan bapak Drs. Wisnu Djatmiko, M.T. selaku dosen pembimbing I selaku dosen pembimbing II yang telah memberikan motivasi dan arahan serta solusi dan pendapat dalam menyelesaikan skripsi.
2. Keluarga tercinta, yang telah memberikan semangat dan dukungan baik moril maupun materi.
3. Teman-teman Jurusan Teknik Elektro dan Elektronika 2012 yang memberikan bantuan serta semangat kepada penulis.

Semoga Allah SWT membalaskebaikansemuapihak yang telahmembantu. Saya menyadaribahwa banyak kekurangan dalam penyusunan skripsiini, namun peneliti berharap semoga penelitian ini dapat bermanfaat bagi semua pihak.

Jakarta, Januari2017

Peneliti

ABSTRAK

ADHI PUTRA PRATAMA. NIM: 5215122627. Rancang Bangun Sistem Pengendali Suhu Dan Kelembapan Otomatis Mesin Penetas Telur Ayam Polandia Berbasis Arduino.Skripsi. Jakarta: Program Studi Pendidikan Teknik Elektronika, Fakultas Teknik, Universitas Negeri Jakarta, Januari 2017.

Penelitian bertujuan untuk membuat sistem pengendali suhu dan kelembapan tempat pengeraman telur ayam polandia secara otomatis. Penelitian dilakukan di Laboratorium Instrumentasi Teknik Elektro, Universitas Negeri Jakarta pada bulan Mei sampai bulan Desember tahun 2016.

Sistem pengendali suhu dan kelembapan mesin penetas telur ayam polandia terdiri dari 3 subsistem utama, yaitu: subsistem input terdiri dari push button sebagai akses untuk mengaktifkan mesin penetas, sensor DHT11 untuk membaca suhu dan kelembapan, subsistem proses memanfaatkan mikrokontroler arduino sebagai kontrol pada alat, dan subsistem output terdapat lampu, *heater, exhaust* untuk mengendalikan suhu dan kelembapan, pompa air untuk memompa air kedalam bak penampungan, dan buzzer untuk mengetahui ketika telur sudah 21 hari didalam mesin penetas.

Sistem yang diusulkan sudah berhasil dibuat dan diuji untuk menetas 4 butir telur ayam polandia, dengan mendapatkan hasil ke 4 telur menetas. Sistem yang dibuat dapat mengendalikan suhu pada 39° C dan kelembapan 55%.

Kata Kunci: Pengendali, pengontrolan, mesin penetas, ayam polandia, arduino, *sensor DHT11*

ABSTRACT

ADHI PUTRA PRATAMA. NIM: 5215122627. Design of Automatic Temperature and Humidity Control System Arduino-based Polish Chicken Egg Hatch Machine. Thesis. Jakarta: Education Program of Electronic Engineering, Faculty of Engineering, Jakarta State University, January 2017.

The objective of this research is to make controlling temperature and humidity of polish chicken eggs incubation automatically. The research was conducted at Instrumentation Laboratory of Faculty of Engineering, Jakarta State University from May to December 2016.

The temperature and humidity control system of the egg hatcher machine of Polish chicken consists of three main subsystems, namely: subsystem input consists of push button as an access to activate the hatcher machine, DHT11 sensor to read temperature and humidity, process subsystem utilize arduino microcontroller as control on tool, and subsystem output includes lamps, heater, exhaust to control temperature and humidity, water pumps to pump water into reservoirs, and buzzers to find out when the egg is 21 days inside the hatchery machine.

Proposed system has been successfully constructed and tested to incubate 4 eggs of polish chicken, by getting the result of 4 eggs hatching. The created system can control the temperature at 39 ° C and 55% moisture.

Keywords: controller, control, hatcher machine, polish chicken, arduino, DHT11 sensor

DAFTAR ISI

HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN	iii
KATA PENGANTAR	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DARTAR TABEL	x
DAFTAR GAMBAR	xi
BAB I PENDAHULUAN	1
Latar Belakang Masalah.....	1
Identifikasi Masalah	3
Pembatasan Masalah	3
Perumusan Masalah.....	3
Tujuan Penelitian.....	4
Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA	5
Kerangka Teori.....	5
Hakekat Merancang.....	5
Sistem Kendali	5
Mesin Penetas	6
Menenal ayam	8
Jenis ayam berdasarkan tujuan pemeliharaan	9
Jenis-jenis ayam hias	11
Telur ayam.....	16
Struktur telur	17
Komposisi telur	18
Pengelolaan telur sebelum ditetaskan.....	19
Proses penetasan telur	20
IDE(<i>Integrated Development Environment</i>)Arduino	21
Sensor	22
Sensor suhu dan kelembapan (DHT11)	22
Sensor ketinggian air	23
Arduino	24

2.1.9.1. Arduino mega2560	25
Relay.....	27
Lampu pijar	28
Pompa DC	30
Kipas angin.....	30
Pemanas air (<i>heater</i>).....	31
<i>Buzzer</i> (Pendering)	32
Kerangka Berfikir.....	33
Blok Diagram	33
Alur Kerja Sistem.....	34
BAB III METODOLOGI PENELITIAN	36
Tempat dan Waktu Penelitian	36
Alat dan Bahan Penelitian	36
Diagram Alir Penelitian	38
Desain Sistem.....	39
Perancangan <i>Hardware</i>	40
Menentukan Sistem Kendali	40
Menentukan sensor suhu DHT11	40
Menentukan Sensor ketinggian air	41
Menentukan Jenis lampu.....	42
Menentukan <i>Heater</i>	43
Menentukan <i>Exhaust</i>	44
Menentukan Pompa DC	45
Pemilihan <i>power supply</i>	46
Perancangan <i>relay 4 chanel</i>	47
Perancangan <i>Software</i>	48
Perancangan Arduino IDE	48
Teknik dan Prosedur Pengumpulan Data	51
Teknik Analisis Data	52
Pengujian suhu diluar dan didalam box mesin penetas	52
Pengujian sumber panas	54
Pengujian <i>relay 4 channel</i>	55
Pengujian Akurasi Sensor	57
Pengujian Mesin Penetas Telur.....	58
BAB IV HASIL PENELITIAN	60
4.1. Deskripsi Hasil Penelitian	60

Analisis Hasil Penelitian	62
Hasil Pengujian Kenaikan Suhu	62
Hasil Pengujian Sumber Panas	64
Hasil Pengujian <i>relay 4 channel</i>	65
Hasil Pengujian akurasi sensor DHT11	68
Hasil Pengujian Mesin Penetas Telur	69
Pembahasan	72
Aplikasi Hasil Penelitian	75
BAB V KESIMPULAN DAN SARAN	76
KESIMPULAN	76
SARAN	76
DAFTAR PUSTAKA	78
LAMPIRAN	79
1. Desain Alat dan Foto	
2. Skema Rangkaian full	
3. Datasheet Arduino Mega	
4. Datasheet Sensor DHT11	
5. Datasheet RTC DS 3231	
6. Program Inkubator Telur Ayam Polandia	
7. Tabel Populasi Ternak Ayam	
RIWAYAT HIDUP	

DAFTAR TABEL

Tabel 2.1. Komposisi Telur.....	18
Tabel 2.2. Spesifikasi Arduino Mega.....	26
Tabel 2.3. Bagian-bagian Lampu Pijar	29
Tabel 3.1. Input Arduino mega	50
Tabel 3.2. Output Arduino mega	50
Tabel 3.3. Pengujian Kenaikan suhu 24 jam	53
Tabel 3.4. Pengujian Sumber Panas	55
Tabel 3.5. Pengujian Tegangan Input Relay Diaktifkan	55
Tabel 3.6. Pengujian Tegangan Input Relay Dinonaktifkan.....	56
Tabel 3.7. Pengujian Tegangan Output Relay Diaktifkan	56
Tabel 3.8. Pengujian Tegangan Output Relay Dinonaktifkan.....	57
Tabel 3.9. Pengujian Sensor DHT11.....	58
Tabel 3.10. Pengujian Mesin Penetas	59
Tabel 4.1. Hasil Pengujian Kenaikan Suhu 24 jam.....	62
Tabel 4.2. Hasil Pengujian Sumber Panas	64
Tabel 4.3. Hasil Pengujian Tegangan Input Relay Diaktifkan	65
Tabel 4.4. Hasil Pengujian Tegangan Input Relay Dinonaktifkan.....	66
Tabel 4.5. Hasil Pengujian Tegangan Output Relay Diaktifkan	66
Tabel 4.6. Hasil Pengujian Tegangan Output Relay Dinonaktifkan.....	67
Tabel 4.7. Hasil Pengujian Sensor DHT11.....	68
Tabel 4.8. Hasil Pengujian Mesin Penetas	69

DAFTAR GAMBAR

Gambar 2.1.Sumber Panas yang Biasa Digunakan Peternak.....	6
Gambar 2.2. <i>Termostat</i> sebagai alat pengatur suhu.....	7
Gambar 2.3. Ruang Penetas dan Wadah Telur.....	7
Gambar 2.4. Ayam kate <i>Cochin</i>	12
Gambar 2.5. Ayam Kapas	13
Gambar 2.6. Ayam Polandia	13
Gambar 2.7. Ayam Batik	14
Gambar 2.8. Ayam Hutan	15
Gambar 2.9. Ayam Pelung	15
Gambar 2.10. Ayam Bekisar	16
Gambar 2.11. Struktur Telur.....	17
Gambar 2.12. Pembentukan Telur	18
Gambar 2.13. IDE Arduino	21
Gambar 2.14.Sensor DHT11	23
Gambar 2.15.Transistor	24
Gambar 2.16. Logo Arduino	25
Gambar 2.17.Arduino Mega 2560	26
Gambar 2.18. Struktur Sederhana Relay	27
Gambar 2.19.Bagian-bagian lampu pijar	28
Gambar 2.20. Pompa Air DC.....	30
Gambar 2.21.Kipas Angin	31
Gambar 2.22.Elemen Pemanas Air	32
Gambar 2.23. <i>Buzzer</i>	32
Gambar 2.24. Blok diagram.....	33
Gambar 2.25. <i>Flowchart</i>	35
Gambar 3.1. Langkah-langkah Perancangan Alat	38
Gambar 3.2. Diagram Blok sistem.....	39
Gambar 3.3. Arduino mega.....	40
Gambar 3.4. Sensor DHT11	41
Gambar 3.5. Pin Modul Suhu DHT11 dengan Arduino	41
Gambar 3.6. Transistor	42
Gambar 3.8. Lampu Pijar	43
Gambar 3.9. Rangkaian arduino dengan Lampu	43

Gambar 3.10. Elemen Pemanas / <i>Heater</i>	44
Gambar 3.11. Rangkaian arduino dengan <i>Heatre</i>	44
Gambar 3.12.Kipas Pendingin / <i>fan</i>	45
Gambar 3.13. Rangkaian arduino dengan kipas.....	45
Gambar 3.14.Pompa air DC.....	46
Gambar 3.15. Rangkaian arduino dengan pompa	46
Gambar 3.16. Rangkaian catu daya	47
Gambar 3.17. Rangkaian <i>Relay 4 Channel</i>	48
Gambar 3.18. Aplikasi arduino IDE.....	49
Gambar 3.19. Program pengujian sensor dht11	54
Gambar 4.1. Hasil rancangan 1	60
Gambar 4.2. Hasil Rancangan 2	61
Gambar 4.3. Hasil Rancangan 3	61
Gambar 4.4. Grafik kenaikan suhu di lingkungan dan didalam box.....	63
Gambar 4.5. Grafik kenaikan suhu menggunakan 4 buah lampupijar.....	64
Gambar 4.6. Rumus perhitungan presentase keberhasilan	74

BAB I

PENDAHULUAN

Latar Belakang Masalah

Perkembangan teknologi diberbagai bidang semakin hari semakin mengalami peningkatan, salah satunya adalah perkembangan teknologi di bidang elektronika. Dalam era globalisasi sekarang tidak lepas dari perkembangan dan teknologi, oleh karena itu kita harus mampu menguasai teknologi dan bersaing dengan negara lain. Kemudahan dan efisiensi waktu serta tenaga menjadi pertimbangan utama manusia dalam melakukan aktifitas. Dari waktu ke waktu manusia dihadapkan pada perkembangan teknologi yang begitu pesat, sehingga membuat pekerjaan manusia semakin mudah.

Dalam bidang peternakan khususnya dalam peternakan ayam, masalah yang dihadapi adalah bagaimana untuk menetasakan telur ayam dalam jumlah yang banyak dan dalam waktu yang bersamaan. Kemampuan induk ayam dalam mengerami telurnya terbatas, ini menjadi masalah yang sangat serius karena kebutuhan daging dan telur ayam di pasar yang sangat banyak. Menurut data statistic yang di ambil oleh Badan Pusat Statistik (BPS.GO.id, 2016) populasi ayam ras petelur di Jakarta tidak dibudidayakan atau dikembangbiakan. Masalahlain yang timbul untuk peternak dan pecinta ayam hias yaitu jenis ayam polandia.

Ayam poland, sering disebut sebagai ayam mahkota atau ayam tanduk, merupakan salah satu ayam yang mempunyai keunikan pada tubuhnya. Ayam polandia mempunyai jambul yang lebat, sehingga terlihat seperti tidak memiliki mata karena tertutup oleh lebatnya bulu jambul. Ayam polandia memiliki aneka

ragam yang indah, menurut para pecinta ayam, ayam polandia kemungkinan berasal dari Polandia. Ayam polandia dipelihara untuk tujuan kesenangan. Sesuai dengan fungsinya sebagai hewan peliharaan, beberapa dipelihara untuk dinikmati keindahan bulu atau bentuknya. Tujuan pemeliharaan ayam polandia ini antara lain untuk menghilangkan kejenuhan dan stress.

Berdasarkan liputan yang ditulis (jitunews.com, 2016), salah satu peternak ayam polandia memberitahukan alasannya memelihara ayam polandia dikarenakan penampilan ayam polandia yang sangat cantik dan lucu, bahkan masih jarang orang yang memiliki dan memeliharanya. Sepasang indukan ayam polandia dijual dengan harga 2,5 juta, prospek usaha ayam polandia bagus kedepannya mengingat begitu banyak orang yang tertarik terhadap penampilan ayam ini, sedangkan jumlah peternak belum begitu banyak. Peternak ayam hias yang biasa mendapatkan penghasilan dari harga penjualan ayam yang lebih mahal dari pada ayam kampung atau ayam potong yang biasa masyarakat konsumsi sehari-hari ini masih memiliki beberapa masalah. Diantaranya ayam hias khususnya jenis ayam polandia tidak bertelur dalam jumlah yang banyak dibanding ayam telur atau ayam kampung, lalu berikutnya indukan ayam polandia tidak mau mengerami telurnya sehingga proses perkembangbiakan terhenti (jitunews, 2016)

Dengan permasalahan yang ada pada proses perkembangbiakan ayam hias khususnya jenis ayam polandia maka Penggunaan teknologi dapat dimaksimalkan pada sistem perkembangbiakan ayam hias jenis ayam polandia sebagai salah satu bentuk inovasi yang mampu membantu proses berkembangbiak yang lebih baik untuk menggantikan induk ayam mengerami telurnya yaitu dengan merancang

sebuah sistem mesin penetas telur yang dapat mengontrol suhu dan kelembapannya secara otomatis.

Identifikasi Masalah

Ditinjau dari latar belakang masalah, maka permasalahan dapat diidentifikasi sebagai berikut :

1. Bagaimana merancang dan membuat program arduino yang dapat mengendalikan suhu dan kelembaban mesin penetas?
2. Bagaimana memantau proses penetasan telur ayam polandia didalam mesin penetas menggunakan arduino?

Pembatasan Masalah

Untuk menjaga agar penelitian lebih terarah dan focus, maka diperlukan adanya pembatasan masalah. Dengan pertimbangan tersebut, maka penelitian ini dibatasi pada:

- 1.. Mikrokontroler yang digunakan adalah Arduino mega.
2. Objek penelitian adalah ayam polandia.
3. Proses pembalikan telur dilakukan dengan cara manual.
4. Perhitungan kelembaban yang belum maksimal (mengacu pada sensor kelembaban yang digunakan).
5. Tidak dibahas masalah listrik padam.

Perumusan Masalah

Berdasarkan latar belakang masalah, identifikasi masalah, dan pembatasan masalah yang telah dijelaskan, maka masalah dalam penelitian ini dapat dirumuskan menjadi “Bagaimana Merancang Sistem kendali Otomatis Mesin

Penetas Telur Ayam Polandia yang dapat Mengendalikan Suhu 38°C - 40°C Serta Kelembapan 50% - 65% ?”

Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk Membuat sistem kendali mesin penetas telur ayam polandia yang dapat mengendalikan suhu serta kelembapan secara otomatis berbasis arduino sehingga nantinya dapat digunakan untuk menetas telur ayam polandia.

Manfaat Penelitian

Penelitian ini bermanfaat untuk :

1. Untuk Mahasiswa

Melatih kemampuan mahasiswa dalam membuat teknologi yang tepat guna untuk proses penetasan telur.

2. Untuk Pengguna

Memudahkan pengguna dalam mengembakbiakan ayam hias jenis ayam polandia.

3. Untuk Institusi Perguruan Tinggi

Sebagai sumber pustaka dalam, ilmu pengetahuan dan teknologi, serta perkembangan khususnya Program Studi Pendidikan Teknik Elektronika konsentrasi instrumen kendali dan umumnya Fakultas Teknik Universitas Negeri Jakarta.

BAB II

TINJAUAN PUSTAKA

Kerangka Teori

Rancang Bangun

Perancangan atau rancang bangun merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dan sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian (Pressman, 2002).

Dengan kata lain, merancang berarti merencanakan suatu hal yang akan dibuat agar menghasilkan suatu rancangan sesuai dengan yang telah direncanakan ataupun yang telah diatur sebelumnya.

Sistem Kendali

Sistem menurut Marimin, Tanjung, & Prabowo, (2006: 1) adalah suatu kesatuan usaha yang terdiri dari bagian-bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks.

Sistem Kendali adalah suatu sistem yang bertujuan untuk mengendalikan suatu proses agar keluaran yang dihasilkan dapat dikontrolkan sehingga tidak terjadi kesalahan terhadap referensi yang ditentukan. Dalam hal ini keluaran yang dikontrolkan adalah kestabilan, ketelitian dan kedinamisannya (Rawung, 2013: 93) Sistem kendali biasanya digunakan dalam alat yang berhubungan dengan

automasi dan pemrograman. Pengendalian dilakukan agar perencanaan system kerja alat dapat berjalan dengan baik sesuai dengan keinginan.

Mesin Penetas

Mesin penetas adalah mesin yang pada umumnya bekerja berdasarkan sistem yang terdiri atas beberapa komponen penyusun, komponen penyusun tersebut terdiri dari sumber panas, pengatur suhu, pengatur kelembapan dan wadah telur. (Wakhid Abdul: 2016)

1. Jenis sumber panas

Embrio di dalam telur membutuhkan suhu lingkungan yang hangat agar dapat berkembang sempurna hingga menetas. Dalam proses penetasan alami, suhu hangat didapatkan melalui pengeraman oleh induk, sementara dalam penetasan menggunakan mesin tetas, suhu hangat diperoleh melalui sumber panas diantaranya lampu minyak, lampu pijar, lampu listrik. Pada gambar 2.1 dapat dilihat contoh sumber pemanas yang biasa digunakan oleh peternak dalam skala rumah tangga.



Gambar 2.1 Sumber panas yang biasa digunakan peternak

2. Pengatur suhu dan kelembapan

Alat pengatur suhu berfungsi untuk mengatur tinggi rendahnya panas yang dihasilkan di dalam ruang penetasan dari sumber panas. Pada gambar 2.2 dapat dilihat pengatur suhu pada mesin tetas semi modern yang sudah bekerja secara otomatis



Gambar 2.2 Termostat sebagai alat pengatur suhu

3. Ruang penetasan dan wadah telur

Bagian dalam dari mesin biasanya berisi ruang penetasan dan wadah untuk menempatkan telur. pada gambar 2.3 dapat dilihat ruang penetasan dan wadah untuk menempatkan telur



Gambar 2.3 Ruang penetasan dan wadah telur

Mengenal Ayam

Menurut sejarahnya, ayam jinak yang dipelihara manusia sekarang adalah berasal dari ayam liar. Proses domestikasi atau penjinakan ayam diperkirakan terjadi seumur dengan adanya manusia di bumi. Keturunan ayam yang telah menjadi jinak kemudian disilang-silangkan oleh manusia. Menurut teorinya ayam liar ini adalah ayam hutan atau *Gallus gallus*.

Hirarki klasifikasi ayam menurut Rose (2001) adalah sebagai berikut.

Kingdom : Animalia
 Subkingdom : Metazoa
 Phylum : Chordata
 Subphylum : Vetebrata
 Kelas : Aves
 Ordo : Galliformes (*Game birds*)
 Famili : Phasiandae (*Peasants*)
 Genus : Gallus
 Spesies : Gallus gallus

Usaha pemeliharaan dan peternakan ayam mulai berkembang pesat di Amerika dan Eropa pada abad ke-19. Melalui penyilangan atau perkawinan antar ayam kemudian diarahkan untuk mendapatkan jenis ayam ternak baru yang memiliki nilai ekonomi tinggi. (Rahayu, Sudaryani, Santosa : 2011)

Ayam berkembang biak dengan cara bertelur, kehidupan ayam dimulai dari sebutir telur yang berasal dari induknya, setelah bertelur induk ayam mengerami dan terus membolak-balik telur telurnya agar tetap hangat pada umumnya ayam mengerami telurnya selama kurang lebih 21 hari. (Ray, Hannah : 2007)

Penetasan telur ayam dapat dilakukan secara ilmiah yaitu dengan dierami oleh induknya dan dapat pula dilakukan dengan mesin tetas. Jika penetasan telur dilakukan pada induknya, jumlah telur yang dapat ditetaskan terbatas. Tetapi, penetasan telur dengan mesin tetas dapat mencapai puluhan bahkan ratusan butir telur dalam sekali penetasan.

Jenis Ayam Berdasarkan Tujuan Pemeliharaan

Ayam yang telah mengalami penyilangan dari bermacam-macam bangsa akan menciptakan jenis ayam baru dengan nilai ekonomi produksi tinggi dan bersifat turun-temurun. Jenis ayam baru tersebut kemudian dikenal dengan nama strain. Pemberian nama strain biasanya dilakukan oleh *breeding farm* pencipta. Di Indonesia dikenal dua istilah, yaitu ayam ras dan ayam buras/kampung. Ayam ras yaitu ayam yang induk atau nenek moyangnya merupakan ayam impor, sedangkan ayam buras atau kampung adalah ayam yang menurut pengertian kita adalah ayam yang induk atau nenek moyangnya berasal dari Indonesia. Jenis ayam berdasarkan tujuan pemeliharaannya seperti berikut ini (Rahayu, Sudaryani, Santosa : 2011).

1. Ayam Bibit Penghasil Anakan Ayam

Ayam bibit berasal dari ayam ras penghasil anak ayam pedaging maupun petelur. Pada pemeliharaan ayam bibit harus dipelihara ayam jantan untuk mengawini ayam betina.

2. Tipe Ayam Petelur

Ayam tipe ini khusus menghasilkan telur dengan cirri-ciri sebagai berikut.

- a) Berbadan langsing dan tegap.
- b) Menghasilkan telur sekitar 200-300 butir per tahun

- c) Mempunyai masa rontok bulu untuk peremajaan sel-sel dalam tubuhnya.
- d) Bila dikawinkan dengan pejantan dapat menghasilkan telur tetas

3. Tipe Ayam Pedaging

Ayam tipe pedaging dapat menghasilkan relative banyak daging dalam waktu singkat. Ciri-ciri ayam tipe pedaging adalah sebagai berikut.

- a) Ukuran badan ayam pedaging relative besar, padat, kompak, dan berdaging penuh sehingga disebut tipe berat.
- b) Jumlah telur relative sedikit.
- c) Bergerak lambat dan tenang.
- d) Biasanya lebih lambat mengalami dewasa kelamin.
- e) Beberapa jenis ayam tipe pedaging mempunyai bulu kaki dan masih suka mengeram. Contoh ayam kampung adalah pelung.(Rahayu, Sudaryani, Santosa : 2011)

4. Tipe Ayam Dwiguna

Ayam yang dapat menghasilkan telur sekaligus daging sering disebut dengan tipe dwiguna. Ciri-ciri ayam tipe dwiguna adalah sebagai berikut.

- a) Ukuran badannya sedang sehingga dapat juga dijadikan sebagai tipe pedaging.
- b) Jumlah telurnya banyak dan besar-besar, kerabangnya berwarna cokelat bercampur hitam
- c) Dalam satu kelompok ayam, ada yang bersifat suka mengeram.
(Rahayu, Sudaryani, Santosa : 2011).

Tipe ayam ini sangat berkembang sebagai petelur unggul dan setelah tua dijual dagingnya. Contoh pada ayam kampung adalah ayam kedu putih dan ayam merawang. (Rahayu, Sudaryani, Santosa : 2011)

5. Ayam untuk Kesenangan

Ayam yang dipelihara untuk kesenangan biasanya dijual dengan harga yang lebih mahal dibandingkan dengan jenis ayam lainnya. Hal ini disebabkan oleh keistimewaan yang dimiliki oleh ayam tersebut, seperti keindahan bentuknya, kekuatannya, dan kemerduan suaranya.

a. Keindahan bentuknya

Ciri-ciri ayam yang dipelihara untuk kesenangan ini adalah sosoknya indah, lucu, menarik, dan biasanya berbentuk mungil. Contoh ayam jenis ini diantaranya adalah ayam kate, kate emas, serama, ayam batik, ayam kate lokal, ayam polandia.

b. Keindahan/ kemerduan suaranya

Beberapa ayam seperti ayam pelung dan bekisar dijuluki sebagai ayam penyanyi karena kemerduan suaranya. Ada juga ayam buras yang terkenal dengan kokokannya yang khas, di antaranya ayam balenggek dan ayam ketawa.

Jenis-Jenis Ayam Hias

Selain untuk tujuan konsumsi (penghasil daging dan telur), ada juga ayam yang dipelihara untuk tujuan kesenangan (hobi). Para hobiis biasa menyebutnya sebagai ayam hias. Oleh karena bersifat untuk kesenangan, pemeliharaan ayam hias hanya dilakukan oleh kalangan terbatas. Ayam hias dapat berasal dari ayam hias lokal maupun ayam hias impor. Ayam hias terdiri dari berbagai macam jenis.

Setiap jenis ayam hias tentunya memiliki keunggulan yang berbeda-beda. Ayam hias yang dikembangkan di Indonesia selama ini berasal dari ayam hias impor dan ayam hias lokal. (Rahayu, Sudaryani, Santosa : 2011)

1. Jenis ayam hias Impor

Ayam-ayam hias impor yang dikenal di Indonesia antara lain berasal dari Vietnam, Jepang, China, Polandia, dan Eropa.

a. Ayam kate *chochin*

Ayam kate *chochin* merupakan ayam kate yang berasal dari Vietnam. Ciri-cirinya memiliki bulu hias pada punggung yang menjurai sampai ke lutut, warnanya putih bergaris coklat kehitaman dengan jengger berbentuk mawar besar merah ayam kate *chochin* dapat dilihat pada gambar 2.4.



Gambar 2.4 Ayam Kate *Cochin*, ayam hias yang berasal dari Vietnam

b. Ayam Kapas

Ayam kapas berasal dari China. Ayam ini memiliki bulu seperti kapas berwarna putih dengan jengger bilah atau mawar. Ayam kapas dapat dilihat pada Gambar 2.5.



Gambar 2.5 Ayam kapas merupakan ayam hias yang berasal dari jepang

c. Ayam Polandia

Merupakan jenis ayam hias impor yang berasal dari Polandia. Bulu ayam berwarna indah dengan jambul yang lebat, sehingga terlihat seperti tidak memiliki mata karena tertutup oleh lebatnya bulu jambul. Ayam Polandia dapat kita lihat seperti gambar 2.6 dibawah ini



Gambar 2.6 Ayam Polandia merupakan ayam hias yang berasal dari Polandia

d. Ayam Batik

Ayam batik berasal dari Eropa. Warna bulunya berukir-ukir seperti batik dengan motif sisik ikan yang jelas dan rapi. Ayam jantan

dewasa berbobot 600 g dan betina 500 g. Ayam batik dapat dilihat seperti gambar 2.7.



Gambar 2.7 Ayam Batik merupakan ayam hias yang berasal dari Eropa

2. Jenis-jenis Ayam Hias Lokal

a. Ayam Hutan

Ayam hutan adalah ayam yang ditangkap di hutan, tetapi saat ini ada yang sudah ditangkarkan, seperti ayam hutan hijau yang berasal dari Jawa dan Madura. Ayam jantan ini memiliki berat sekitar 0,7 kg untuk ayam jantan dewasa dan 0,4 kg untuk ayam betina. Ada juga ayam hutan merah yang berasal dari Sumatera, Sulawesi dan Semenanjung Malaysia. Konon kedua jenis ayam hutan ini merupakan cikal bakal ayam kampung atau buras di Indonesia. Ayam hutan bisa dilihat pada gambar 2.8.



Gambar 2.8 Ayam Hutan

b. Ayam Pelung

Ayam pelung merupakan persilangan dari ayam kampung jantan dengan ayam hutan betina. Persilangan ini dilakukan pertama kali di daerah Jawa Barat (Cianjur) dan sekarang banyak berkembang di daerah Sukabumi. Sosok badannya tegap dengan suara kokoh khas pada ayam jantannya. Selain untuk kemerduan suaranya, ayam ini dapat dijadikan cikal bakal untuk ayam kampung pedaging. Bobot ayam jantan dewasa mencapai 3,5-5,5 kg dan ayam betina berkisar 2,5-3,5 kg. Ayam pelung dapat dilihat pada gambar 2.9.



Gambar 2.9 Ayam Pelung

c. Ayam Bekisar

Ayam bekisar adalah hasil persilangan jantan ayam hutan dengan betina ayam kampung. Warna bulu didominasi oleh warna bulu induk betina; sedangkan postur tubuh, sifat dan suara dipengaruhi oleh induk jantan. Ayam bekisar dipelihara karena kemerduan suaranya. Ayam bekisar bisa kita lihat pada gambar 2.10



Gambar 2.10 Ayam Bekisar

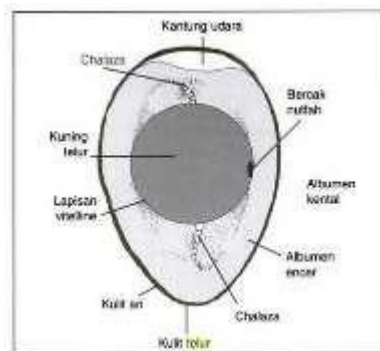
Telur Ayam

Ayam berkembang biak dengan cara bertelur, kehidupan ayam dimulai dari sebutir telur yang berasal dari induknya, setelah bertelur induk ayam mengerami dan terus membolak-balik telur telurnya agar tetap hangat pada umumnya ayam mengerami telurnya selama kurang lebih 21 hari. (Ray, Hannah : 2007)

Penetasan telur ayam dapat dilakukan secara ilmiah yaitu dengan dierami oleh induknya dan dapat pula dilakukan dengan mesin tetas. Jika penetasan telur dilakukan pada induknya, jumlah telur yang dapat ditetaskan terbatas. Tetapi, penetasan telur dengan mesin tetas dapat mencapai puluhan bahkan ratusan butir telur dalam sekali penetasan.

Struktur Telur

Kuning telur dibentuk di dalam tubuh oleh sistem perkembangbiakan ayam betina sewaktu sedang birahi dan siap untuk dikawini ayam jantan yang juga sedang birahi. Semasa di dalam peranakan, sekelompok kuning telur yang bentuknya seperti sekelompok buah anggur ini dimasuki oleh sel telur betina (ovum), tepat berada di tengah-tengahnya. Karenanya, agar terjadi pembuahan, dibutuhkan sel jantan (sperma) yang kuat yang dapat menerobos masuk ke dalam kuning telur sehingga dapat bersatu dengan ovum. (Hartono, T., Isman :2012) Bagian-bagian telur dapat dilihat pada gambar 2.11.

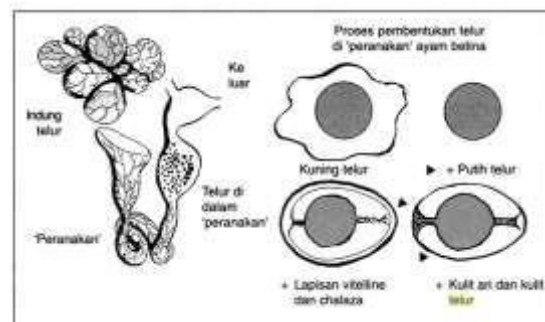


Gambar 2.11 Bagian-bagian Pada Telur

Proses selanjutnya adalah dilapisinya kuning telur ini oleh lapisan yang terbuat dari zat fosfoprotein (vitellin), yang berfungsi sebagai bagian pengaman pertama pembuahan. Pada saat ini dibentuk pula semacam tambang penyeimbang, yang biasa disebut *chalaz*, agar kuning telur dapat tepat berada di bagian ujung atas dan ujung bawah nulation kuning telur.

Kuning telur lalu turun ke bagian tengah ‘peranakan’. Disini dua kali lagi kuning telur dilapisi zat putih telur yang berfungsi sebagai penahan guncangan. Setelah itu, kuning dan putih telur turun ke bagian bawah ‘peranakan’ untuk dilapisi dengan kulit arid an zat kapur yang terlihat sebagai kulit telur. Pada proses akhir ini, kulit ari akan membentuk kantung udara, zat kapur akan semakin mengeras, dan telur keluar melalui dubur ayam betina. (Hartono, T., Isman :2012)

Seluruh proses ini terjadi dalam waktu 24-26 jam. Itulah sebabnya, ayam betina (sebagus apapun kualitasnya) hanya dapat bertelur sebutir setiap pagi. Proses terbentuknya telur pada tubuh ayam betina dapat dilihat pada gambar 2.12 berikut



Gambar 2.12 Pembentukan Telur

Komposisi Telur

Telur pada umumnya memiliki berat sekitar 50-57 gram / butirnya, dan terdiri dari 11 % bagian kulit telur, 58 % bagian putih telur, dan 31 % bagian kuning telur. Komposisi zat yang terkandung di dalam setiap telur dapat dilihat pada table 2.1.

Tabel 2.1 Komposisi Telur

Komposisi	Air	Protein	Lemak	Lainnya
Putih Telur	88%	11%	-	1%
Kuning Telur	48%	18%	33%	1%
Seluruhnya	74%	13%	11%	2%

Pada table diatas jelas terlihat bahwa kandungan protein yang terdapat pada setiap butir telur adalah sekitar 7 gram. Protein sangat dibutuhkan untuk membangun sel tubuh dan memperbaiki sel tubuh yang rusak. (Hartono, T., Isman:2012)

Pengelolaan Telur Sebelum Ditetaskan

Sebelum memasukan telur kedalam mesin tetas ada beberapa hal yang perlu mendapatkan perhatian diantaranya.

1. Telur yang akan ditetaskan harus diyakini sebagai telur yang telah dibuahi. Telur yang dapat ditetaskan hanyalah telur yang dihasilkan oleh ayam betina melalui perkawinan dengan ayam jantan.
2. Telur disimpan dengan baik pada kondisi temperature ideal yaitu antara 5-15°C, sel embrio yang ada didalam telur tidak tumbuh dan tidak mati (dormansi). Pada temperature ruang seperti inilah idealnya sebutir telur disimpan sebelum ditetaskan.
3. Usahakan telur tetap kering. Harus disimpan ditempat yang sejuk dan kering sebelum ditetaskan.
4. Uji cobakan mesin tetas sebelum benar-benar digunakan. Selama 1 hari penuh dimaksudkan untuk mengethui tingkat kestabilan temperature yang stabil yang hanya bervariasi sekitar 0,2-0,3°C.
5. Jangan menyimpan telur lebih dari seminggu. Telur yang disimpan lebih dari seminggu sejak ditetaskan memiliki risiko kegagalan penetasan yang tinggi.

6. Telur yang baik saja yang ditetaskan. Maksudnya, telur yang akan ditetaskan berasal dari ayam betina yang diyakini merupakan ayam betina yang sehat dan terawat dengan baik. (Hartono, T., Isman :2012)

Proses Penetasan Telur

Prinsip teknis agar seseorang sukses dalam usaha penetasan sebenarnya sangat sederhana, yaitu pelaku penetasan harus bias mengkondisikan mesin tetas seperti keadaan alami induk ayam yang sedang mengerami telur. kondisi tersebut meliputi kestabilan suhu dan kelembapan. Ada beberapa hal yang harus diperhatikan diantaranya adalah :

1. Kestabilan Suhu

Menurut (Hartono, Tito dan Isman:2012)Suhu yang diperlukan untuk menetas telur ayam ialah 38°C - 40°C , tidak sampai lima jam ayam akan bergerak atau bergeser lagi. Jika masih terlalu panas, telur yang dierami akan dibalik dengan kepala dan lehernya. Proses pembalikan ini bertujuan untuk meratakan suhu dan melawan gaya gravitasi, sehingga posisi embrio di dalam telur tetap baik.

2. Kelembapan

Kelembapan udara berfungsi untuk mengurangi atau menjaga dalam telur dan merapuhkan kerabang telur. Jika kelembapan tidak optimal, embrio tidak akan mampu memecahkan kerabang yang terlalu keras. Namun kelembapan yang terlalu tinggi juga dapat menyebabkan air masuk melalui pori-pori kerabang lalu terjadi penimbunan cairan di dalam telur.

Pada keadaan alaminya, kelembapan diatur oleh keringat yang dikeluarkan induk ayam. Ayam tidak memiliki kelenjar keringat yang sempurna sehingga

kelembapan yang terjadi menjadi tidak terlalu tinggi. Penetasan telur ayam membutuhkan kelembapan sebesar 50%-65% (Hartono, Tito dan Isman:2012)

3. Ventilasi

Ventilasi memegang peranan yang sangat penting sebagai sumber oksigen embrio untuk bernafas. Ventilasi juga menjadi kunci penyeimbang antara kelembapan dan suhu. Jika ventilasi lancar maka kelembapan bisa berkurang. Namun jika ventilasi terhambat maka suhu dan kelembapan akan meningkat.

IDE (*Integrated Development Environment*) Arduino

IDE (*Integrated Development Environment*) Arduino adalah sebuah *software* untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam memory mikrokontroler. (Syahwil, 2013: 39), contoh IDE arduino dapat dilihat pada Gambar 2.13



Gambar 2.13. (*Integrated Development Environment*) IDE Arduino 1.6.9

Bahasa pemrograman yang digunakan untuk menulis program di dalam Arduino adalah Processing. Processing adalah bahasa pemrograman tingkat tinggi yang dialektanya sangat mirip dengan C++ dan Java, sehingga pengguna yang

sudah terbiasa dengan kedua bahasa tersebut tidak akan menemui kesulitan dengan Processing. IDE Arduino terdiri dari:

- 1) *Editor*, sebuah window yang memungkinkan pengguna untuk menulis dan mengedit program dengan bahasa processing.
- 2) *Compiler*, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Mikrokontroler tidak mampu memahami sebuah kode program (bahasa processing) sehingga modul ini dibutuhkan untuk mengubah kode program menjadi kode biner sebagai bahasa di pahami oleh sebuah mikrokontroler.
- 3) *Uploader*, sebuah modul yang memuat kode biner dari computer ke dalam *memory* papan arduino.

Sensor

Sensor adalah jenis transduser yang digunakan untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi tegangan dan arus listrik. Sensor sering digunakan untuk pendeteksian pada saat melakukan pengukuran atau pengendalian. Beberapa jenis sensor yang banyak digunakan dalam rangkaian elektronik antara lain sensor cahaya, sensor suhu, dan sensor tekanan (Sensor dan Transduser, 2003: 6)

Sensor Suhu dan Kelembapan DHT11

DHT11 adalah sensor digital yang dapat mengukur suhu dan kelembapan udara di sekitarnya. Sensor ini sangat mudah digunakan bersama dengan Arduino. Memiliki tingkat stabilitas yang sangat baik serta fitur kalibrasi yang sangat akurat. Koefisien kalibrasi disimpan dalam OTP program memory, Dalam

sensor ini terdapat sebuah thermistor tipe NTC (*Negative Temperature Coefficient*) untuk mengukur suhu, sebuah sensor kelembaban tipe resistif dan sebuah mikrokontroler 8-bit yang mengolah kedua sensor tersebut dan mengirim hasilnya ke pin output dengan format *single-wire bi-directional* (kabel tunggal dua arah). Setiap DHT11 dikalibrasi secara ketat di laboratorium yang sangat akurat pada kalibrasi kelembaban. Koefisien kalibrasi disimpan sebagai program dalam memori OTP, yang digunakan oleh proses deteksi sinyal internal sensor (uuegar, 2017). Bentuk fisik sensor suhu DHT11 ditunjukkan pada Gambar 2.14.

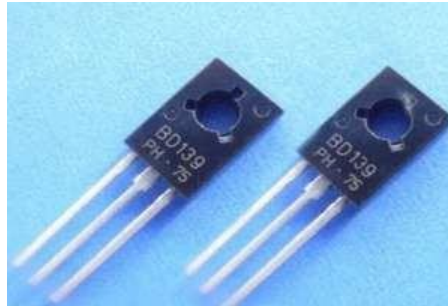


Gambar 2.14 Sensor DHT11

Sensor Ketinggian Air

Transistor adalah komponen semikonduktor yang berfungsi sebagai penguat arus, pemutus dan penyambung (switching) sirkuit, stabilisasi tegangan, atau sebagai pemodulasi sinyal (Istiyanto, 2014: 26).

Cara kerja transistor dapat dianalogikan sebagai keran arus listrik, misalnya pada transistor BJT, arus yang diberikan pada terminal Basis dijadikan pengatur aliran arus antara terminal Emitor dan Kolektor. Untuk bentuk fisik dari transistor dapat dilihat dari **Gambar 2.15**



Gambar 2.15. Transistor

Pada umumnya, transistor memiliki 3 terminal, yaitu Basis (B), Emitor (E) dan Kolektor (C). Transistor dapat dijadikan sebagai sebuah masukan dan dapat bekerja bila terminal pada basis ter-trigger sehingga arus dari kolektor mampu mengalir ke emitor. Dengan prinsip kerja ini pula, transistor dapat dimanfaatkan sebagai sensor ketinggian air.

Arduino

Arduino adalah suatu perangkat prototipe elektronik berbasis mikrokontroler yang fleksibel dan *open-source*, perangkat keras dan perangkat lunaknya mudah digunakan (Andrianto dan Darmawan, 2016: 15). Pembuatan Arduino dimulai pada tahun 2005, dimana sebuah situs perusahaan komputer Olivetti di Ivrea Italia, membuat perangkat untuk mengendalikan proyek desain interaksi siswa supaya lebih murah dibandingkan sistem yang ada pada saat itu. Dilanjutkan pada bulan Mei 2011, dimana sudah lebih dari 300.000 unit Arduino terjual. Logo arduino dapat dilihat pada gambar 2.16

Pendiri dari Arduino itu sendiri adalah Massimo Banzi dan David Cuartielles sebagai *founder*. Awalnya mereka memberi nama proyek itu dengan sebutan Arduin dari Ivrea tetapi seiring dengan perkembangan zaman, nama proyek itu diubah menjadi Arduino yang berarti “teman yang kuat” atau dalam versi bahasa Inggrisnya dikenal dengan sebutan “Hardwin” (Syahwil, 2013 : 61).



Sumber : (Arduino, 2005)

Gambar 2.16 Logo Arduino

2.1.9.1 Arduino Mega2560

Arduino Mega 2560 adalah papan mikrokontroler berbasis Atmega2560. Yang mempunyai 54 pin digital *input/output*, dimana 14 pin dapat digunakan sebagai PWM, 16 analog *input*, 4 UARTs (*hardware serial ports*), 16 MHz *crystal oscillator*, sambungan USB, *power jack*, ICSP *header*, dan tombol *reset*. Board ini juga menggunakan daya yang terhubung ke komputer dengan kabel USB atau daya eksternal dengan adaptor AC-DC atau baterai. Arduino mega *compatible* dengan *shield* yang didesain untuk Arduino Duemilanove or Diecimilia (Syahwil, 2013 : 68). Bentuk serta tampilan dari Arduino Mega 2560 dapat dilihat pada Gambar 2.17.



Gambar 2.17 Arduino Mega 2560
 Sumber: *Data Sheet* Arduino Mega 2560

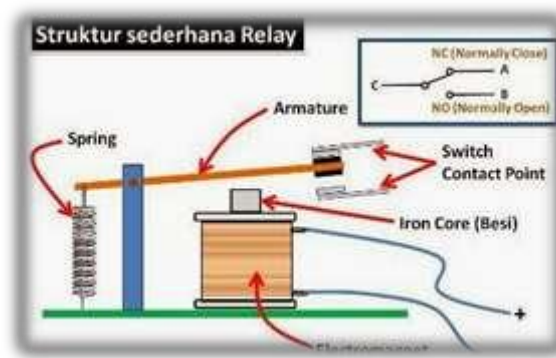
Arduino Mega 2560 memiliki dimensi panjang dan lebar yaitu 4 inch x 2,1 inch (10,16 cm x 5,3 cm), dilengkapi dengan konektor USB dan *jack power*. Arduino Mega 2560 mempunyai 4 *port* serial dan *flash memory* sebesar 256Kb yang secara umum sudah cukup besar untuk kebanyakan program di mikrokontroler.

Tabel 2.2 Spesifikasi Arduino Mega 2560

Mikrokontroler	ATmega2560
Operating Voltage	5V
<i>Input Voltage (recommended)</i>	7-12V
<i>Input Voltage (limits)</i>	6-20V
<i>Digital I/O Pins</i>	54 (of which 15 provide PWM output)
<i>Analog Input Pins</i>	16
<i>DC Current per I/O Pin</i>	40 mA
<i>DC Current for 3.3V Pin</i>	50 mA
<i>Flash Memory</i>	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
<i>Clock Speed</i>	16 MHz

Relay

Dalam suatu sistem kontrol elektronik *relay* menjadi komponen yang sering dipakai, karena relay mudah dalam pengoperasiannya dan dapat dikendalikan dari jarak yang jauh. *Relay* adalah sebuah saklar yang dikendalikan oleh arus (Bishop, 2004 : 55). Dan merupakan suatu piranti yang menggunakan magnet listrik untuk mengoperasikan seperangkat kontak. Struktur *relay* dapat dilihat pada **Gambar 2.18**.



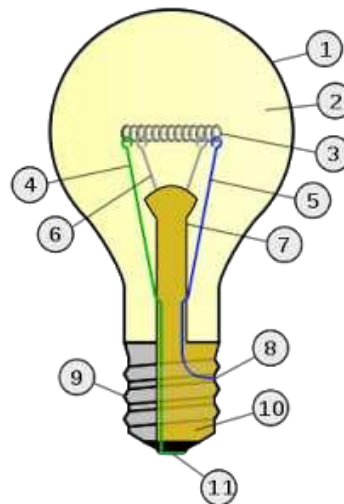
Gambar 2.18 Struktur Sederhana Relay

Prinsip Kerja dari *Relay* adalah memiliki sebuah kumparan tegangan rendah yang dililitkan pada sebuah inti, dimana terdapat sebuah armatur besi yang akan tertarik menuju inti apabila arus mengalir melewati kumparan. Gerakan armatur ini dipakai melalui pengungkit, untuk menutup atau membuka kontak-kontak. Beberapa susunan kontak dapat dipakai, semuanya itu secara listrik terisolasi dari rangkaian kumparan. Pada pokoknya relay digunakan sebagai alat penghubung pada rangkaian (Petruzella, 2001 : 371).

Lampu Pijar

Lampu pijar adalah sumber cahaya buatan yang dihasilkan melalui penyaluran arus listrik melalui filamen yang kemudian memanas dan menghasilkan cahaya. Kaca yang menyelubungi filamen panas tersebut menghalangi udara untuk berhubungan dengannya sehingga filamen tidak akan langsung rusak akibat teroksidasi.

Komponen utama dari lampu pijar dapat dilihat pada **Gambar 2.19** dan **Tabel 2.3** adalah bola lampu yang terbuat dari kaca, filamen yang terbuat dari wolfram, dasar lampu yang terdiri dari filamen, bola lampu, gas pengisi, dan kaki lampu.



Gambar 2.19 Bagian-bagian Lampu Pijar

Tabel 2.3 Bagian- bagian Lampu Pijar

Nomor	Keterangan gambar
1	Bola Lampu
2	Gas bertekanan rendah (argon, neon, nitrogen)
3	Filamen wolfram
4	Kawat penghubung ke kaki tengah
5	Kawat penghubung ke ulir
6	Kawat penyangga
7	Kaca penyangga
8	Kontak listrik di ulir
9	Sekrup ulir
10	Isolator
11	Kontak listrik di kaki tengah

Pada dasarnya filamen pada sebuah lampu pijar adalah sebuah resistor. Saat dialiri arus listrik, filamen tersebut menjadi sangat panas, berkisar antara 2800 Kelvin hingga maksimum 3700 Kelvin. Ini menyebabkan warna cahaya yang dipancarkan oleh lampu pijar biasanya berwarna kuning kemerahan. Pada temperatur yang sangat tinggi itulah filamen mulai menghasilkan cahaya pada panjang gelombang yang kasatmata. Hal ini sejalan dengan teori radiasi benda hitam. Indeks renderasi warna menyatakan apakah warna obyek tampak alami apabila diberi cahaya lampu tersebut dan diberi nilai antara 0 sampai 100. Angka 100 artinya warna benda yang disinari akan terlihat sesuai dengan warna aslinya. Indeks renderasi warna lampu pijar mendekati 100. Efisiensi lampu atau dengan kata lain disebut dengan efikasi luminus adalah nilai yang menunjukkan besar efisiensi pengalihan energi listrik ke cahaya dan dinyatakan dalam satuan lumen per Watt. Kurang lebih 90% daya yang digunakan oleh lampu pijar dilepaskan sebagai radiasi panas dan hanya 10% yang dipancarkan dalam radiasi cahaya kasat mata. (Wikipedia, 2016).

Pompa DC

Pompa air listrik termasuk kategori pompa sentrifugal (turbo), yaitu mengendalikan daya dari luar yang diberikan kepada poros pompa untuk memutar baling-baling (impeller). Pada pompa air listrik terjadi perubahan energy, yaitu energy listrik diubah menjadi energy kinetic (gerak). Untuk beberapa tipe, terdapat perbedaan dalam hal kerja mesin pompa air, tetapi secara umum prinsip kerja pompa air listrik relatif sama (Sutrisno, 2008: 8).

Secara sistematis cara kerja pompa air listrik digambarkan sebagai berikut: Ketika listrik dinyalakan motor akan bergerak memutar rotor ketika rotor berputar maka air akan terhisap ke dalam input pompa dan dikeluarkan melalui saluran buang. Pompa air DC dapat dilihat pada **Gambar 2.20**



Gambar 2.20 Pompa Air DC

Kipas angin (*exhaust fan*)

Pada umumnya kipas angin dipergunakan untuk menghasilkan angin. Fungsi yang umum adalah untuk pendingin udara, penyegar udara ventilasi (*exhaust fan*),.

Perkembangan kipas angin semakin bervariasi baik dari segi ukuran, penempatan posisi, serta fungsi. Mulai dari kipas angin mini, kipas angin yang digunakan di dalam unit CPU computer seperti kipas untuk mendinginkan *processor* . Kipas angin tersebut berfungsi untuk menjaga suhu udara agar tidak melewati batas suhu yang di tetapkan. (Wikipedia, 2016) kipas angin pada CPU dapat dilihat pada

Gambar 2.21



Gambar 2.21 Kipas angin (*exhaust*)

Pemanas Air (*Heater*)

Pemanas air atau katel uap adalah alat untuk menghasilkan uap air, yang akan digunakan untuk pemanas atau tenaga gerak. Mulai dari yang berbahan bakar batubara, minyak bumi sampai listrik. Besi tuang digunakan untuk bejana pemanas untuk pemanas air. Walaupun suatu pemanas biasanya disebut “pendidih” (boiler), karena tujuannya adalah untuk membuat air panas, bukan uap air, karena dioperasikan pada tekanan rendah dan menghindari pendidihan sebenarnya. Kerapuhan dari besi tuang menjadikannya tidak cocok untuk katel uap tekanan tinggi. (Wikipedia, 2016) gambar elemen pemanas (*Heater*) bias dilihat pada **Gambar 2.22**



Gambar 2.22 Elemen Pemanas Air

Buzzer

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hamper sama dengan *loud speaker*, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi electromagnet, karena kumparan dipasang pada diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menimbulkan suara (Wikipedia, 2016). Gambar *buzzer* dapat dilihat seperti pada **gambar 2.23**



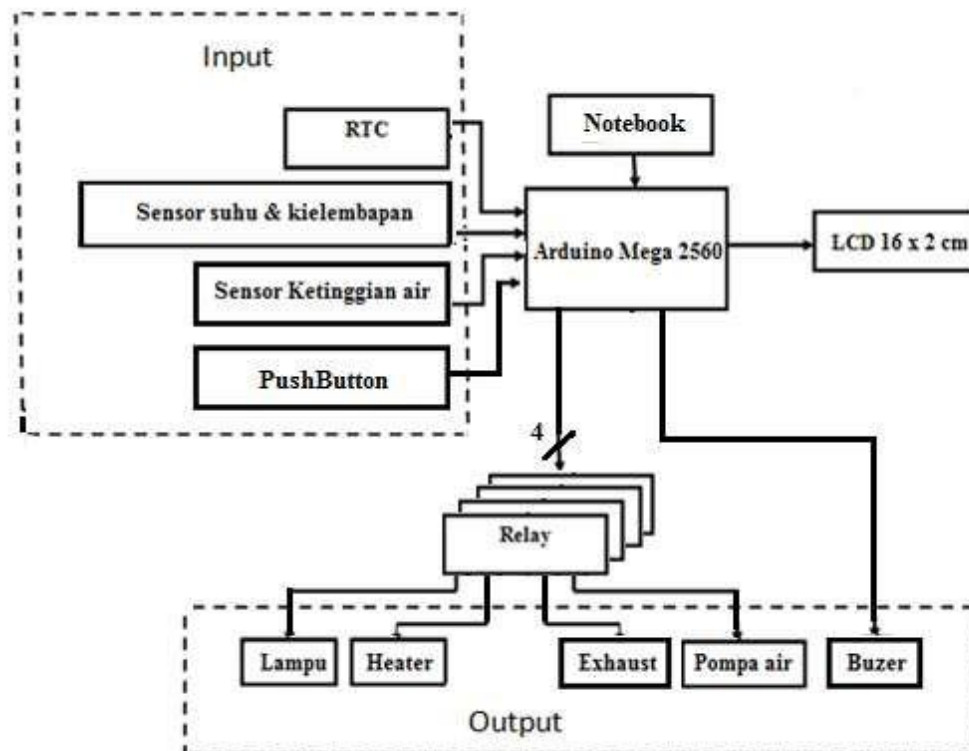
Gambar 2.23 Buzzer

KERANGKA BERFIKIR

Berdasarkan teori-teori yang telah dibahas, maka dapat dirancang sebuah sistem pengendali suhu dan kelembapan otomatis pada mesin penetas telur ayam polandia menggunakan mikrokontroller Arduino Mega sebagai perangkat kontrol kendali.

BLOK DIAGRAM

Blok diagram adalah salah satu tahapan dari proses dalam pembuatan alat. Blok diagram digunakan untuk menentukan komponen penyusunan dari suatu alat yang akan dibuat, sehingga hasil akhirnya sesuai yang diinginkan. Perencanaan dan pembuatan alat pemantauan dan pengendalian jarak jauh pendistribusian bahan bakar minyak menggunakan *web* berbasis arduino uno dapat digambarkan pada blok diagram yang terdapat pada Gambar 2.24.

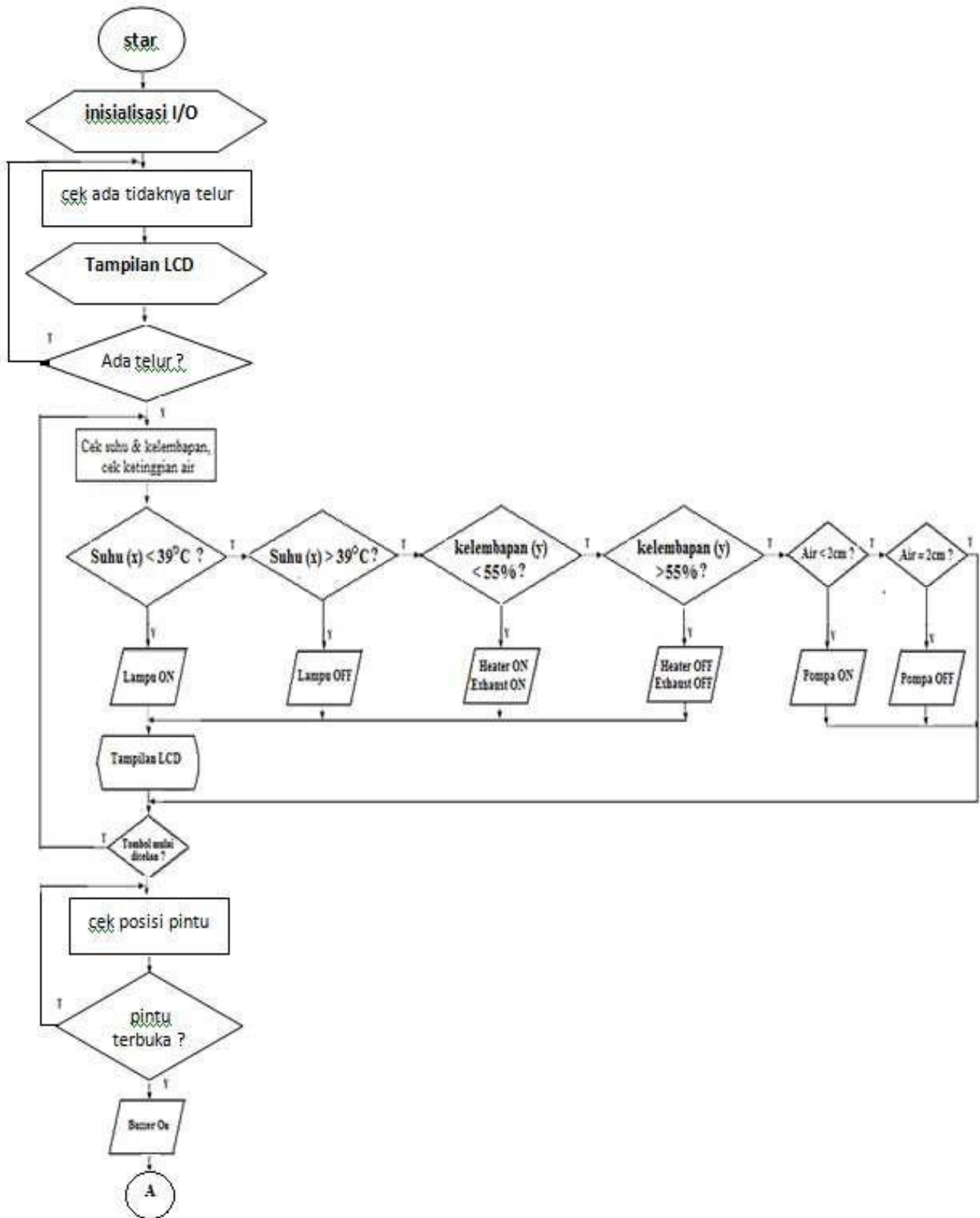


Gambar 2.24 Blok Diagram

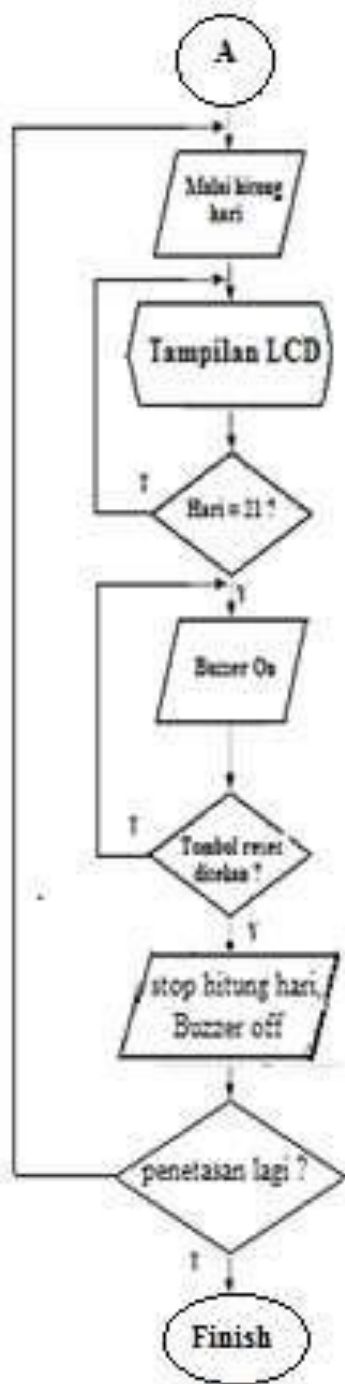
ALUR KERJA SISTEM

Sistem pemantauan dan pengendalian mesin penetas telur ayam polandia ini bekerja pada saat pengguna menghidupkan saklar utama untuk mulai menetas telur. Setelah kondisi saklar on maka sensor suhu dan kelembapan akan mulai membaca dan mengirimkan data untuk di tampilkan pada *interface* LCD 16 x 2, selanjutnya system akan mensinkronisasi data yang telah diterima apakah sesuai dengan perintah. Setelah sesuai maka data tersebut akan diteruskan menuju *output* dengan sistem kendali mikrokontroler dalam hal ini menggunakan komunikasi serial pada mikrokontroler arduino mega yang programnya telah dirancang sebelumnya menggunakan *software* Arduino IDE.

Proses sinkronisasi antara sensor suhu dan kelembapan dengan tampilan *interface* pada LCD 16 x 2 terus berlangsung selama keduanya terhubung ke internet. Sinkronisasi ini memungkinkan sistem untuk terus mengirim data dan melakukan pertukaran data yang berisi perintah untuk mematikan atau menghidupkan lampu, *exhaust*, *heater* dan pompa serta membaca jumlah suhu dan kelembapan yang dikirim. *Output* yang digunakan adalah *relay* sebagai saklar Lampu, *exhaust*, *heater* dan pompa pada sistem. Selain itu terdapat beberapa sensor masukan seperti sensor DHT11 dan sensor ketinggian air yang nantinya digunakan sebagai umpan balik yang menunjukkan tingginya suhu dan kelembapan untuk mengendalikan suhu dan kelembapan di dalam mesin penetas, serta sensor ketinggian air yang digunakan sebagai umpan balik untuk mengaktifkan pompa air. Flowchart kerja sistem dapat dilihat pada gambar 2.25



Gambar 2.25 Flow Chart



BAB III

METODOLOGI PENELITIAN

Tempat dan Waktu Penelitian

Penelitian tugas akhir ini akan dilakukan di Laboratorium Teknik Elektronika, Universitas Negeri Jakarta. Tepatnya di gedung L jurusan Teknik Elektro Lantai 4, dilaksanakan pada bulan Desember 2016 – Juli 2017.

Alat dan Bahan Penelitian

Bahan bahan yang digunakan dalam pembuatan rancang bangun sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandia berbasis arduino, sebagai berikut:

Sistem laptop yang digunakan didalam penelitian ini sebagai berikut:

1. Processor Intel(R) Core i5-2450M CPU @ 2.50GHz
2. RAM 4 GB
3. 32-bit operating system, x64-based processor

Perangkat lunak yang digunakan:

1. Arduino IDE 1.6.7, yang digunakan untuk memprogram *board* arduino mega.
2. Eagle 6.4.0, yang digunakan untuk membuat *layout* dan *schematic* rangkaian PCB.
3. Web Browser, digunakan untuk mengakses web.
4. Sketcup, digunakan untuk membuat desain perancangan alat.
5. Microsoft Word 2013, digunakan untuk menyelesaikan penulisan.:

Alat ukur yang digunakan:

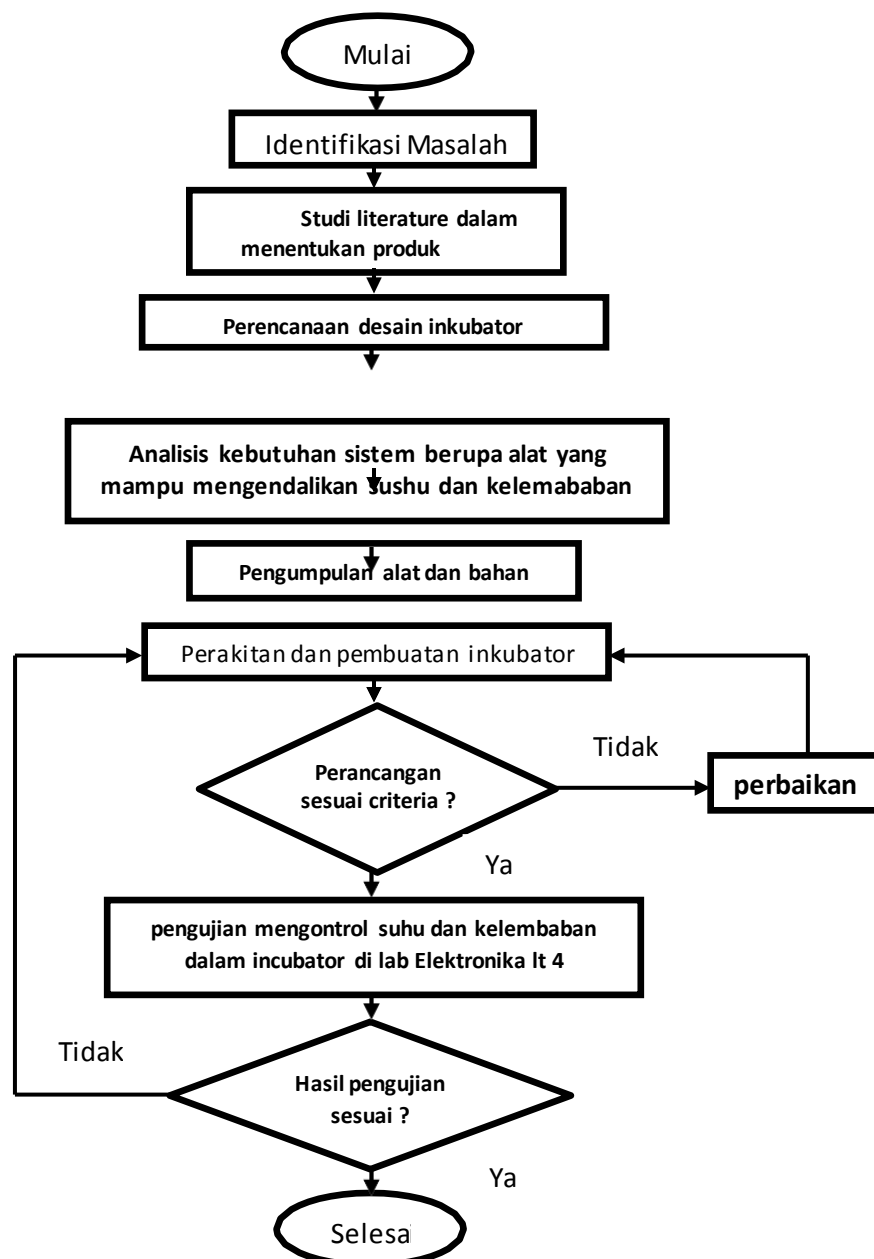
1. Multimeter digital sanwa
2. Thermo Hygro

Bahan bahan yang digunakan dalam pembuatan rancang bangun sistem pengendalian suhu dan kelembapan mesin penetas telur otomatis berbasis arduino, sebagai berikut:

1. Arduino Mega
2. Rangkaian *Driver Relay*
3. Sensor DHT11
4. Sensor Ketinggian Air dengan Transistor
5. Lampu pijar
6. *Exhaust (fan)*
7. *Heater* (Elemen pemanas)
8. Pompa DC
9. *Buzzer*
10. *Push Button* (2)
11. LCD 16 x 2
12. Lampu Indikator
13. Saklar
14. *Power supply/* Catu Daya

Diagram Alir Penelitian

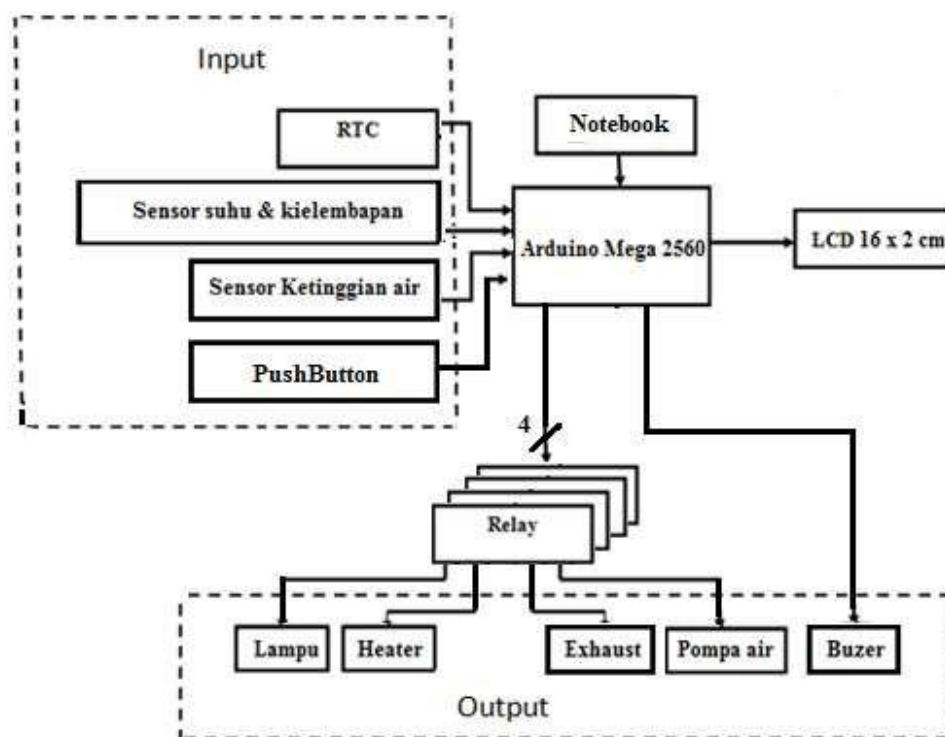
Perancangan penelitian yang akan dilakukan dalam kegiatan penelitian ini dengan metode rekayasa teknik, yaitu dengan studi literatur dan penerapan langsung dilapangan, serta memiliki beberapa langkah penelitian lainnya sehingga pada saat pembuatan alat sudah ditentukan parameter-parameternya. Seperti yang tertera pada gambar 3.1 langkah-langkah perancangan alat dibawah ini.



Gambar 3.1. Langkah-Langkah Prancangan Alat

3.3.1. Desain Sistem

Desain sistem adalah tahapan dimana dilakukan penuangan pikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat permodelan sistem seperti desain sistem dan prinsip kerja alat. Seperti yang tertera pada gambar 3.2 Diagram blok sistem



Gambar 3.2 Diagram Blok Sistem Pengendali Suhu dan Kelembapan Mesin Penetas Telur Ayam Polandia

Pada subsistem alat terdapat sensor suhu (DHT11) untuk membaca suhu dan kelembapan dalam mesin penetas juga terdapat sensor ketinggian air untuk mendeteksi apakah penampang air masih menyediakan air yang cukup untuk menjaga kelembapan dan RTC yang digunakan sebagai pewaktu, yang kemudian akan diproses oleh arduino. Arduino kemudian akan mengolah data yang diterima untuk mengaktivasikan lampu, heater, exhaust, buzzer dan pompa, dan data hasil proses tersebut akan ditampilkan pada tampilan LCD berukuran 16 x 2 cm.

Perancangan *Hardware*

Menentukan Sistem Kendali

Dalam pembuatan dan pengujian sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandi berbasis arduino ini terlebih dahulu harus menentukan sistem kendalinya. Sistem kendali yang digunakan untuk pembuatan alat ini adalah arduino mega yang dapat dilihat pada Gambar 3.3



Gambar 3.3 Arduino Mega

Menentukan Sensor DHT11

Dalam pembuatan dan pengujian sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandi berbasis arduino peneliti menggunakan sebuah Sensor DHT11. Cara memproduksi data pada sensor DHT11 adalah format data bus tunggal digunakan untuk komunikasi dan sinkronisasi antara MCU dan sensor DHT11. Setiap proses komunikasi akan berlangsung sekitar 4ms. Data di transmisikan dalam format : 8bit integral RH data+, 8bit decimal RH data+, 8bit integral T data+, 8bit decimal T data+, 8bit check sum. Status default pin DATA tinggi. Saat komunikasi antara MCU dan DHT11 dimulai, MCU akan menurunkan pin DATA. Ini disebut “Start Signal” dan ini untuk memastikan DHT11 mendeteksi sinyal dari MCU. Kemudian MCU akan menarik pin DATA

untuk menunggu respon DHT11. Setelah DHT11 mendeteksi sinyal awal, ia akan meneruskan pin DATA sebagai “Respon Signal” . Kemudian DHT11 akan menarik pin DATA, dan mempersiapkan pengiriman data. Selama transisi data, setiap bit data diawali dengan tingkat tegangan rendah dan diakhiri dengan tingkat tegangan tinggi. Panjang sinyal level tegangan tinggi menentukan apakah bit “0” atau “1”. (uugear, 2017) dapat dilihat pada gambar 3.4, dan rangkaian sensor DHT11 dengan arduino dapat dilihat pada gambar 3.5



Gambar 3.4 Sensor DHT11

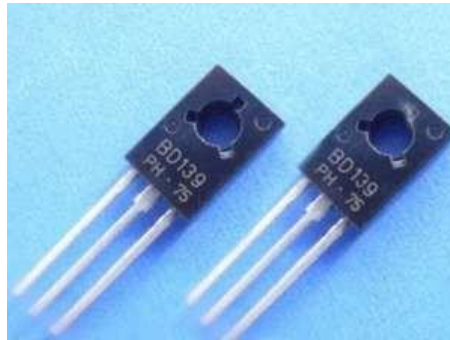


Gambar 3.5 Pin Modul Suhu DHT11 Dengan Arduino

Menentukan Sensor Ketinggian Air

Dalam pembuatan dan pengujian sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandiaberbasis arduinopeneliti menggunakan transistor sebagai komponen untuk dijadikan sensor ketinggian air. Transistor adalah komponen semikonduktor yang berfungsi sebagai penguat arus, pemutus dan penyambung (switching) sirkuit, stabilisasi tegangan, atau sebagai pemodulasi sinyal (Istiyanto, 2014: 26).

Cara kerja transistor dapat dianalogikan sebagai keran arus listrik, misalnya pada transistor BJT, arus yang diberikan pada terminal Basis dijadikan pengatur aliran arus antara terminal Emitor dan Kolektor. Untuk bentuk fisik dari transistor dapat dilihat dari gambar 3.6.



Gambar 3.6. Transistor

Pada umumnya, transistor memiliki 3 terminal, yaitu Basis (B), Emitor (E) dan Kolektor (C). Transistor dapat dijadikan sebagai sebuah masukan dan dapat bekerja bila terminal pada basis ter-trigger sehingga arus dari kolektor mampu mengalir ke emitor. Dengan prinsip kerja ini pula, transistor dapat dimanfaatkan sebagai sensor ketinggian air.

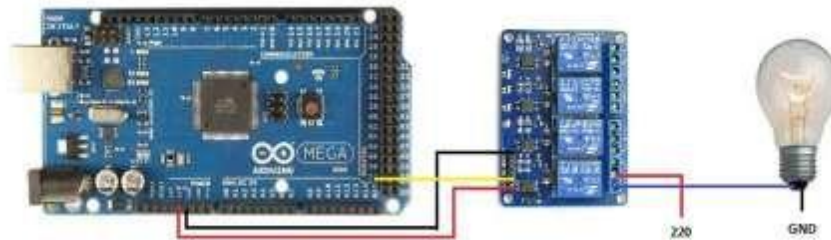
Menentukan Jenis Lampu

Dalam pembuatan dan pengujian sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandi berbasis arduino peneliti menggunakan 4 buah lampu jenis lampu pijar berdaya 5 watt, untuk menyalakannya menggunakan tegangan AC 220 watt. Lampu pijar adalah sumber cahaya buatan yang dihasilkan melalui penyaluran arus listrik melalui filamen yang kemudian memanaskan dan menghasilkan cahaya. Kaca yang menyelubungi filamen panas tersebut menghalangi udara untuk berhubungan dengannya

sehingga filamen tidak akan langsung rusak akibat teroksidasi. Gambar lampu pijar dapat dilihat pada gambar 3.8 dan gambar rangkaian arduino dengan lampu dapat dilihat pada gambar 3.9



Gambar 3.8 Lampu Pijar



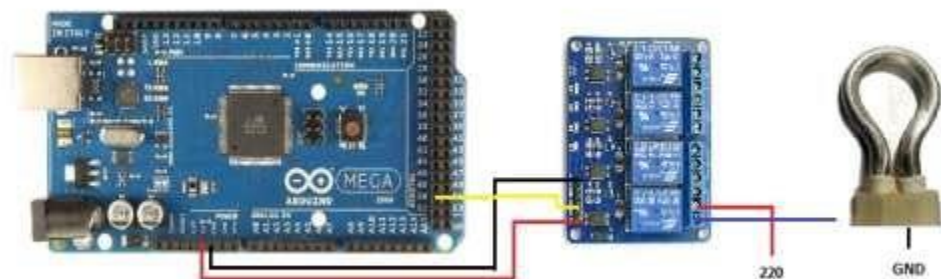
Gambar 3.9 Rangkaian Arduino dengan Lampu

Menentukan *Heater*/ Elemen pemanas

Dalam pembuatan dan pengujian sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandi berbasis arduino peneliti menggunakan sebuah pemanas air / *Heater* dengan tegangan AC 220 volt digunakan sebagai pemanas air untuk meningkatkan kelembapan mesin penetas telur. Gambar *Heater* bias dilihat pada gambar 3.10 di bawah ini. Dan gambar rangkaian *heater* dan arduino bias dilihat pada gambar 3.11



Gambar 3.10 Elemen pemanas / *Heater*



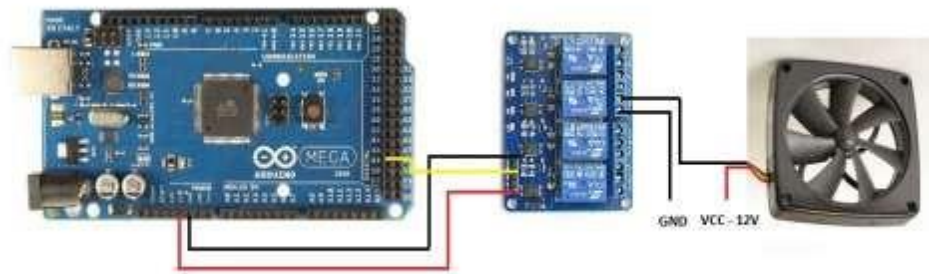
Gambar 3.11 Rangkaian Arduino dengan *Heater*

Menentukan *exhaust* / Kipas angin

Dalam pembuatan dan pengujian sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandia berbasis arduino peneliti menggunakan *exhaust*/ kipas dari pendingin *CPU* sebagai ventilasi pertukaran udara, kipas ini juga digunakan untuk mengurangi kelembapan yang terlalu tinggi didalam mesin penetas telur ayam polandia. Gambar *exhaust* dapat dilihat pada gambar 3.12 dibawah ini, dan rangkaian arduino dengan *exhaust* dapat dilihat pada gambar 3.13



Gambar 3.12 *exhaust* / kipas pendingin CPU



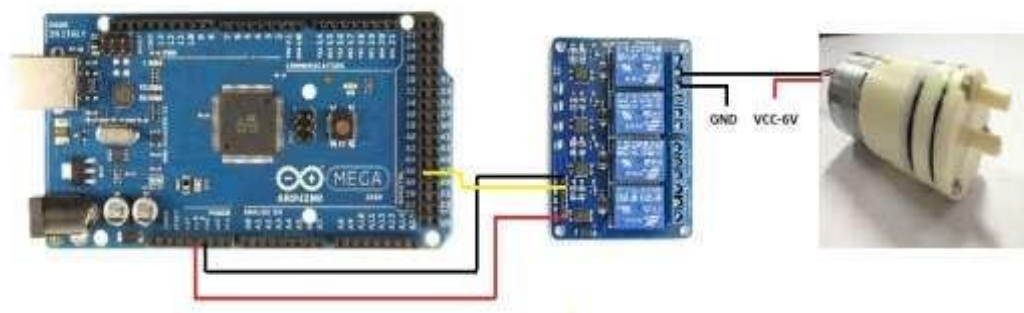
Gambar 3.13 Rangkaian Arduino dengan *exhaust*

Menentukan Pompa DC

Dalam pembuatan dan pengujian sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandiaberbasis arduinopeneliti menggunakan sebuah pompa air DC untuk mengalirkan air dari penampungan diluar mesin masuk kedalam penampungan air di bagian dasar mesin penetas telur. Pompa yang idgunakan pada pembuatan alat ini adalah pompa air dengan tegangan 6 VDC gambar pompa dapat dilihat pada gambar 3.14 dan rangkaian pompa dengan arduino pada gambar 3.15



Gambar 3.14 Pompa DC



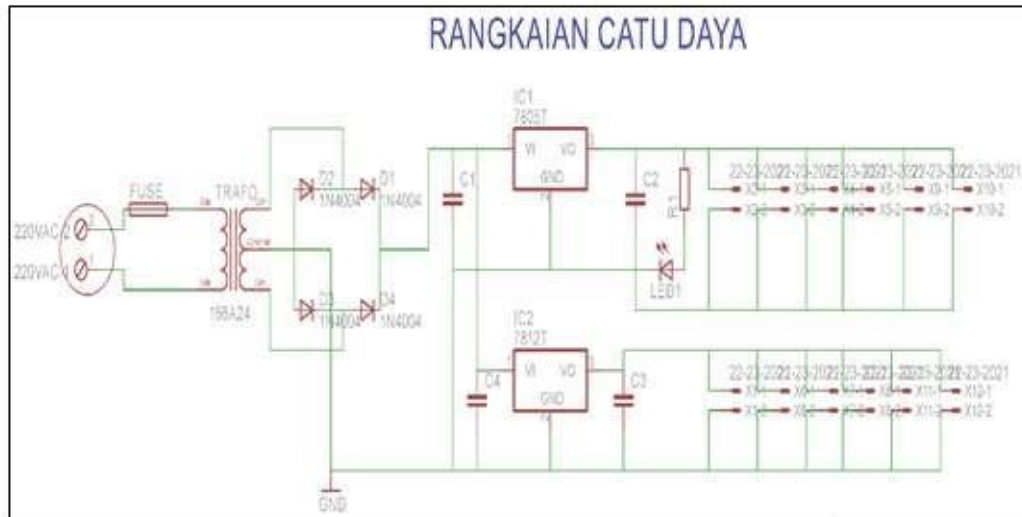
Gambar 3.15 Rangkaian Arduino dengan Pompa DC

Pemilihan *power supply*/ catu daya

Dalam pembuatan dan pengujian sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandia berbasis arduino peneliti menggunakan 3 sumber listrik, yaitu :

1. *Power Supply* dengan tegangan 12 Volt dan arus 3A untuk rangkaian sensor LM358 dan Driver Relay 10 Channel.
2. *Power Supply* router dengan tegangan 5 Volt dan arus 1.0 A.
3. Sumber listrik PLN 220 Volt.

Rangkaian catu daya berfungsi sebagai pemberi daya ke tiap rangkaian agar rangkaian dapat bekerja. Berikut adalah gambar catu daya yang digunakan peneliti tertera pada **Gambar 3.16** di bawah ini.

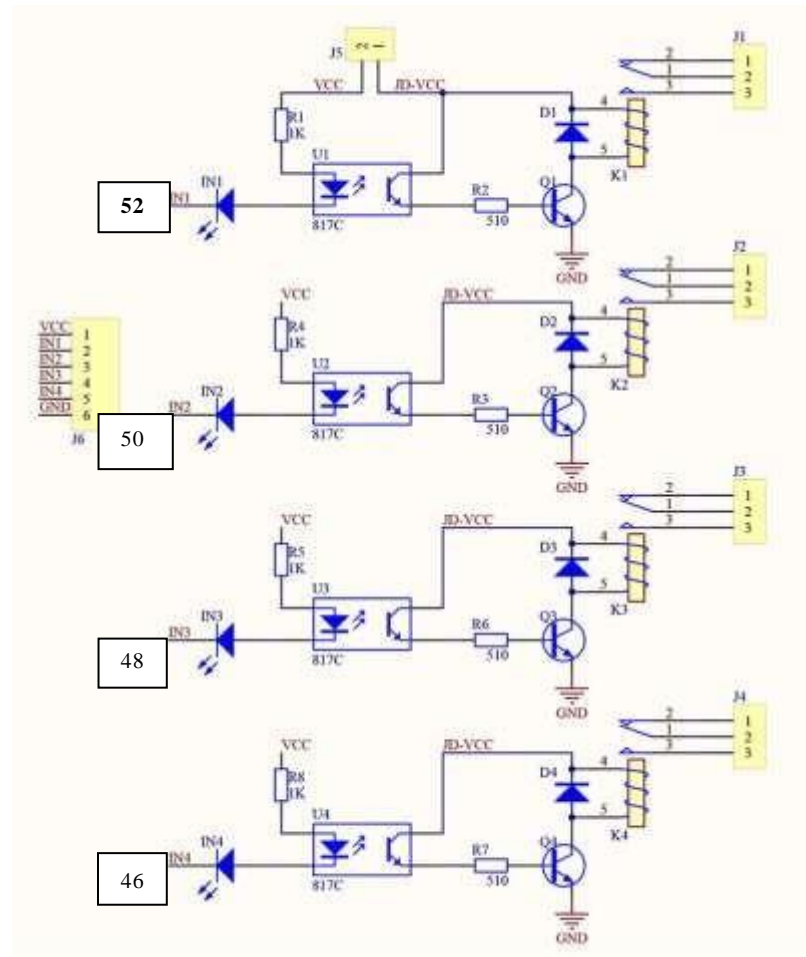


Gambar 3.16 Rangkaian Catu Daya
(Arsip pribadi)

Rangkaian tersebut terdiri dari 2 bagian yaitu, catu daya simetris 12V dan catu daya 5V. Catu daya simetris 12V digunakan untuk memberi daya pada rangkaian driver relay. Sedangkan catu daya 5V digunakan untuk memberi daya ke rangkaian optocoupler, sensor suhu, sensor ketinggian air dan pompa DC

Perancangan Rangkaian *Relay 4 channel*

Rangkaian *relay* berfungsi untuk memungkinkan penggunaan tegangan dan arus yang kecil didapat dari Arduino untuk mengontrol tegangan dan arus yang lebih besar. Rangkaian *relay* ini terdiri dari 4 *channel*, yang masing-masing *channel*-nya digunakan sebagai saklar *on/off* pada lampu, *heater*, *exhaust*, dan pompa air yang dikendalikan oleh Arduino Mega 2560. Rangkaian *relay* dapat dilihat pada Gambar 3.17.

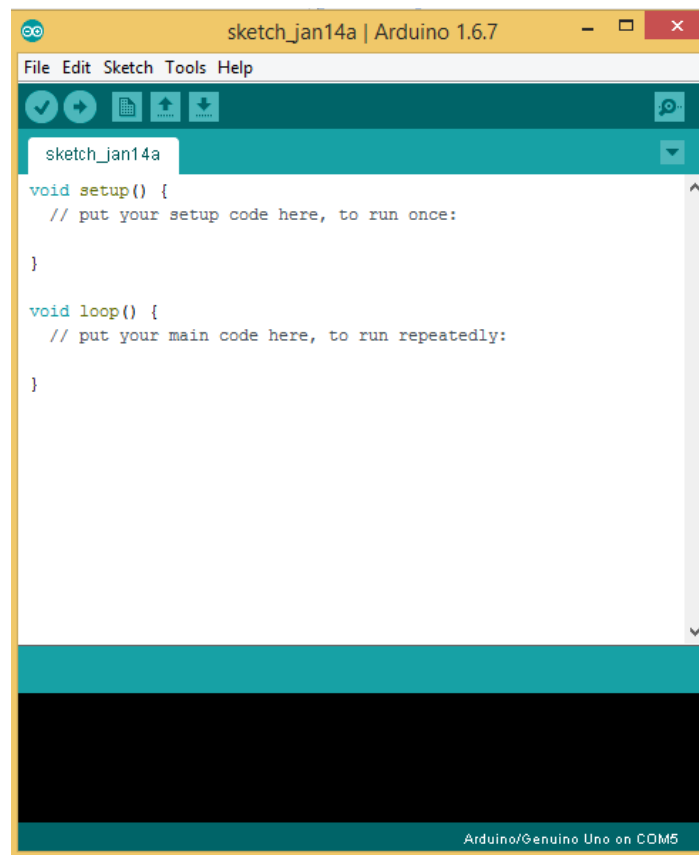


Gambar 3.17 Rangkaian Relay 4 chanel
Gambar diambil dari sunfounder.wiki

Perancangan Software

Perancangan Arduino IDE

Arduino IDE merupakan salah satu bawaan *software* sendiri dari perangkat arduino di mana pemrogramannya menggunakan bahasa C/C++. Arduino IDE yang beroperasi di komputer berfungsi untuk menghasilkan sebuah *file* yang berformat .ino yang akan unggah pada papan arduino. Dalam hal ini arduino IDE digunakan untuk membuat program alat sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandia pada bagian mikrokontroler arduino uno yang nantinya digunakan sebagai pengendali perangkat keras. Berikut ini adalah Gambar 3.18 tampilan awal perangkat lunak arduino IDE 1.6.7:



Gambar 3.18 Aplikasi Arduino IDE 1.6.7

Berdasarkan fungsi dari aplikasi arduino IDE di atas, peneliti menggunakan sebagai pemrograman mikrokontroler arduino mega untuk sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandiaberbasis arduino. Beberapa *input* dan *output* yang dihubungkan menggunakan parameter data untuk memasukan program. Berikut ini adalah parameter data yang digunakan pada arduino uno menggunakan perangkat lunak arduino IDE 1.6.7.

1. *Input* berupa Sensor DHT11 dan Sensor ketinggian air, pin yang digunakan pada arduino mega dapat dilihat pada Tabel 3.1 berikut:

Tabel 3.1. *Input* Arduino Uno

JENIS	PIN PERANGKAT INPUT	PIN ARDUINO
Sensor DHT11	VCC	5V
	GROUND	GND
	DATA	22
Sensor ketinggian air	VCC	5V
	GROUND	GND
	DATA	32
	DATA	34

2. *Output* berupa lampu, heater, exhaust, pompa dan buzzer, pin yang digunakan pada arduino mega dapat dilihat pada Tabel 3.2 berikut:

Tabel 3.2. *Output* Arduino Uno

JENIS	PIN PERANGKAT INPUT	PIN ARDUINO
Lampu	<i>Output 1</i>	52
<i>Heater</i>	<i>Output 2</i>	50
<i>Exhaust</i>	<i>Output 3</i>	48
Pompa	<i>Output 4</i>	46
<i>Buzzer</i>	<i>Output 5</i>	13

Teknik dan Prosedur Pengumpulan Data

Prosedur penelitian yang ditempuh dalam merancang dan membuat sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandiaberbasis arduino terdiri dari beberapa tahapan yaitu :

1. Perakitan modul dimulai dari pemilihan komponen, pemasangan, penyolderan, dan *wiring* kabel.
2. Mengecek kondisi suhu di luar mesin penetas selama 24 jam, dan suhu di dalam mesin penetas selama 24 jam lalu membandingkannya.
3. Mengecek dengan manual apakah *relay* dapat digunakan atau tidak.
4. Mengecek dengan manual apakah lampu dapat digunakan atau tidak.
5. Menguji 1-4 buah lampu sebagai sumber panas untuk mendapatkan suhu maksimal yang dibutuhkan.
6. Mengecek dengan manual apakah *heater* dapat digunakan atau tidak.
7. Mengecek dengan manual apakah *exhaust* dapat digunakan atau tidak.
8. Mengecek dengan manual apakah pompa dapat digunakan atau tidak.
9. Mengecek dengan manual apakah sensor ketinggian air dapat digunakan atau tidak.
10. Mengkalibrasi sensor DHT11 agar dapat digunakan sebagai alat ukur yang presisi.
11. Meletakkan sensor DHT11 sebagai sensor untuk membaca suhu dan kelembapan dengan *input* yang diberikan dari
12. Meletakkan rangkaian sensor ketinggian air sebagai sensor untuk menhidupkan pompa air jika air yang berada didalam penampang air kurang dari batas yang dibutuhkan

13. Pemrograman arduino mega dengan menggunakan *software* arduino. penulisan pemrograman menggunakan *software* arduino IDE. Setelah program selesai dibuat, program siap di upload ke arduino *unoboard*.
14. Membuat tampilan di LCD 16 x 2 sehingga dapat dibuat sesuai dengan kebutuhan peneliti.
15. Pengujian sensor-sensordengan program arduino untuk mengaktifkan aktuator.
16. Selanjutnya uji coba dilakukan dengan memasukan telur ayam polandia untuk di tetaskan di dlaam mesin penetas.
17. Menyimpulkan hasil penelitian dari hasil pengujian ini

Teknik Analisis Data

Teknik analisis data merupakan penelitian yang diperlukan peneliti untuk mendapatkan data dari keseluruhan sistem pendistribusian bahan bakar minak, penguji melakukan penelitian apakah alat tersebut berhasil atau gagal.

Untuk mendapatkan hasil yang diinginkan pada sistem pengendali suhu dan kelembapan otomatis mesin penetas telur ayam polandiaberbasis arduino harus dilakukan pengujian. Berikut adalah kriteria pengujian alat tersebut dengan parameter keberhasilan yang akan dilakukan.

1. Mesin penetas dapat bekerja untuk kemudian mendapatkan tingkat keberhasilan dalam penetasan minimal 80% persen.

Pengujian Suhu diluar dan didalam Box mesin penetas

Pengujian suhu diluar dan di dalam box dilakukan untuk mengetahui grafik perbedaan suhu yang dihasilkan di dalam box selama 24 jam, pengujian ini

dilakukan dengan menggunakan termometer digital, pengujian ini bertujuan untuk mengetahui perbandingan grafik kenaikan suhu yang dihasilkan dilingkungan sekitar dan suhu yang berada di dalam box selama 24 jam. Grafik kenaikan suhu dilingkungan sekitar dan grafik kenaikan suhu didalam box dapat dilihat pada tabel 3.3

Tabel 3.3 Perbedaan suhu lingkungan dan inkubator

Waktu (Jam)	Suhu lingkungan	Suhu di dalam Box
19.00		
20.00		
21.00		
22.00		
23.00		
00.00		
01.00		
02.00		
03.00		
04.00		
05.00		
06.00		
07.00		
08.00		
09.00		
10.00		

11.00		
12.00		
13.00		
14.00		
15.00		
16.00		
17.00		
18.00		

Pengujian sumber panas

Pengujian sumber panas ini dilakukan dengan cara memasang 4 buah lampu pijar berukuran 5 watt pada mesin penetas telur ayam polandia dan melihat suhu panas yang dihasilkan dari 4 buah lampu pijar berukuran 5 watt hingga mesin penetas mendapatkan suhu yang maksimal dari lampu pijar tersebut. Pengujian sumber panas yang dihasilkan 4 buah lampu pijar berukuran 5 watt dapat dilihat pada tabel 3.4 dan program pengujian suhu dengan sensor DHT11 dapat dilihat pada gambar 3.19

```
#include <simpleDHT.h>
int pinDHT11 = 22;
simpleDHT11 dht11;

void setup () {
  Serial.begin (115200);
}

void loop() {
  Serial.println ("sample DHT11...");

  byte temperatur = 0;
  byte humidity = 0;
  if (dht11.read(pinDHT11, &temperature, &humidity, NULL)) {
    Serial.print ("Read DHT11 failed.");
    return;
  }

  serial.print ('sample OK: ');
  serial.print ((int)temperature) ; serial.print("°C, ")
  serial.print ((int)humidity) ; serial.println (" %")

  delay(1000);
}
```

Gambar 3.19 program pengujian suhu dengan sensor DHT11

Tabel 3.4 Pengujian sumber panas

Wktu (menit)	1	2	3	4	5	6	7	8	9	10
Suhu										

Pengujian *Relay4 Channel*

Pengujian *relay* untuk *switch*Lampu, *heater*, *exhaust* dan pompa dilakukan dengan mengukur tegangan *input* dan *output* ketika *relay* aktif dan *relay* tidak aktif. Pengujian tegangan *input relay* diaktifkan tertuang pada Tabel 3.5 dan pengujian tegangan *input relay* dinonaktifkan tertuang pada Tabel 3.6. Sedangkan pengujian tegangan *output relay* diaktifkan tertuang pada Tabel 3.7 dan pengujian tegangan *output relay* dinonaktifkan tertuang pada Tabel 3.8.

Tabel 3.5 Pengujian Tegangan *Input Relay* Diaktifkan

Jenis	Kondisi	Tegangan	Tampilan	Hasil
Lampu	<i>High</i>	4,91 VDC		
<i>Heater</i>	<i>High</i>	4,93 VDC		
<i>Exhaust</i>	<i>High</i>	4,98 VDC		
Pompa	<i>High</i>	4,92 VDC		

Tabel 3.6 Pengujian Tegangan *Input Relay* Dinonaktifkan

Jenis	Kondisi	Tegangan	Tampilan	Hasil
Lampu	<i>Low</i>	0,0 v		
<i>Heater</i>	<i>Low</i>	0,0 v		
<i>Exhaust</i>	<i>Low</i>	0,0 v		
Pompa	<i>Low</i>	0,0 v		

Tabel 3.7 Pengujian Tegangan *Output Relay* Diaktifkan

Jenis	Kondisi	Tegangan	Tampilan	Hasil
Lampu	<i>High</i>	200,3 VAC		
<i>Heater</i>	<i>High</i>	199,6 VAC		
<i>Exhaust</i>	<i>High</i>	8,67 VDC		
Pompa	<i>High</i>	4,98 VDC		

Tabel 3.8 Pengujian Tegangan *Output Relay* Dinonaktifkan

Jenis	Kondisi	Tegangan	Tampilan	Hasil
Lampu	<i>Low</i>	0,015 mV		
<i>Heater</i>	<i>Low</i>	0,086 mV		
<i>Exhaust</i>	<i>Low</i>	0,024 mV		
Pompa	<i>Low</i>	0,00 mV		

Pengujian Akurasi Sensor DHT11

Pengujian sensor DHT11 sebagai *input* dilakukan dengan cara mengukur tingkat akurasi suhuda kelembapan yang dihitung oleh Sensor DHT11 ketika sensor aktif. Pada pengujian sensor DHT11 kali ini peneliti melakukan pengujian dengan cara membandingkan untuk mengetahui tingkat akurasi dari sensor DHT11 yaitu dengan mengaktifkan sensor DHT11 kemudian hasil perhitungannya akan di tampilkan pada LCD 16 x 2, kemudian suhu dan kelembapan yang terdeteksi oleh sensor DHT 11 akan di bandingkan dengan alat yang bernama *Termo Hygro* Untuk pengujian dengan cara diatas dapat dilihat pada Tabel 3.9

Tabel 3.9 Pengujian Sensor DHT11

Pembacaan Sensor DHT11	Pembacaan alat Termo Hygro	Tampilan

Pengujian Mesin Penetas Telur

Pengujian penetasan telur dilakukan dengan memasukan 5 butir telur ayam polandia ke dalam mesin penetas, nantinya dapat terlihat pada hari keberapa telur ayam polandia 1,2,3,4,5 menetas dan keberhasilan penetasan telur ayam polandia ini yang nantinya akan dijadikan acuan sebagai tingkat keberhasilan rancang bangun sistem kendali otomatis mesin penetas telur ayam polandia berbasis arduino. Waktu telur menetas dapat dilihat pada table 3.10

Tabel 3.10 Pengujian mesin penetas

Jadwal Pencatatan	Suhu	Kelembapan	Tampilan
Hari ke 1			
Hari ke 2			
Hari ke 3			
Hari ke 4			
Hari ke 5			
Hari ke 6			
Hari ke 7			
Hari ke 8			
Hari ke 9			
Hari ke 10			
Hari ke 11			
Hari ke 12			
Hari ke 13			
Hari ke 14			
Hari ke 15			
Hari ke 16			
Hari ke 17			
Hari ke 18			
Hari ke 19			
Hari ke 20			
Hari ke 21			

BAB IV

HASIL PENELITIAN

Deskripsi Hasil Penelitian

Berdasarkan blok diagram serta *flowchart* yang telah dirancang pada penjelasan sebelumnya, maka rancang bangun sistem pengendalian suhu dan kelembapan otomatis mesin penetas telur ayam polandia berbasis arduino diimplementasikan oleh peneliti pada gambar 4.1, gambar 4.2 dan gambar 4.3. Pada bab ini akan dilakukan pengujian dan analisa terhadap perangkat sesuai hasil perancangan dan implementasi. Perancangan perangkat diimplementasikan pada instalasi dan konfigurasi komponen. Setelah itu dilakukan pengujian untuk kemudian dilakukan analisa sehingga bias memperbaiki perancangan sistem yang baru. Pengujian yang dilakukan yaitu pengujian secara fungsional.



Gambar 4.1GambarHasil Rancangan1



Gambar 4.2 Gambar Hasil Rancangan2



Gambar 4.3 Gambar Hasil Rancangan3

Analisis Hasil Penelitian

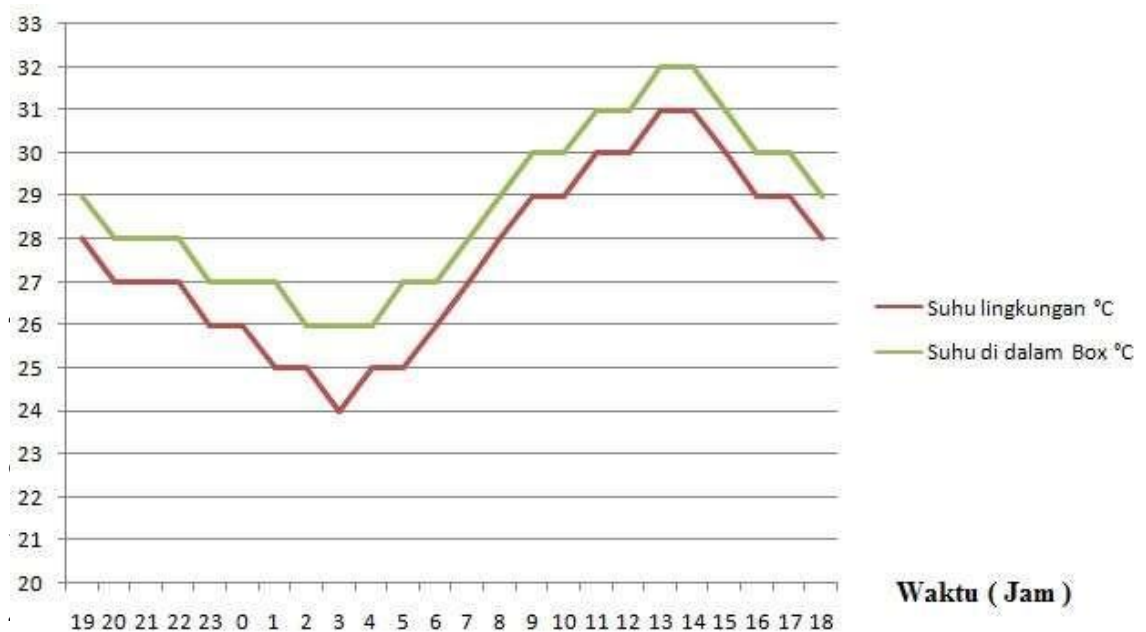
Hasil Pengujian Kenaikan Suhu

Hasil pengujian kenaikan suhu didapat dengan cara mengukur suhu yang berada di lingkungan dan suhu yang berada di dalam mesin penetas selama 24 jam lalu membandingkannya. Hasil pengujian kenaikan suhu tertuang pada tabel 4.1 dan grafik kenaikan suhu tertuang pada gambar 4.4

Tabel 4.1 Pengujian Suhu selama 24 jam

	Suhu lingkungan °C	Suhu di dalam Box °C
19.00	28	29
20.00	27	28
21.00	27	28
22.00	27	28
23.00	26	27
00.00	26	27
01.00	25	27
02.00	25	26
03.00	24	26
04.00	25	26
05.00	25	27
06.00	26	27
07.00	27	28
08.00	28	29

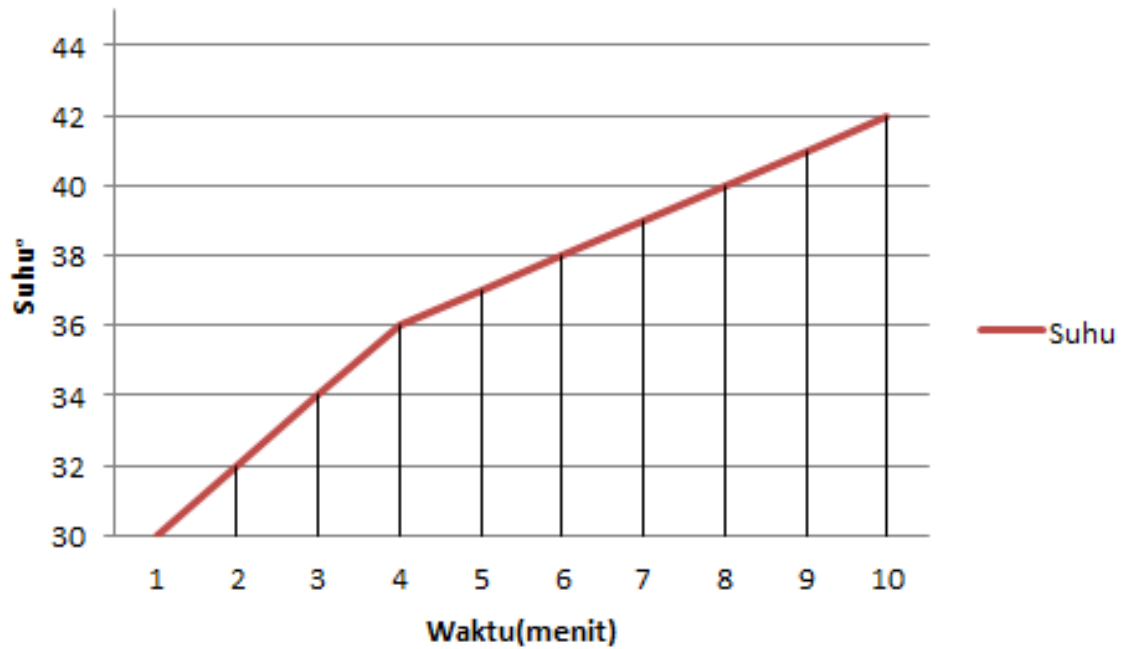
09.00	29	30
10.00	29	30
11.00	30	31
12.00	30	31
13.00	31	32
14.00	31	32
15.00	30	31
16.00	29	30
17.00	29	30
18.00	28	29



Gambar 4.4 Grafik perbandingan suhu lingkungan & suhu didalam box selama 24 jam

Tabel 4.2 Pengujian sumber panas

Wktu (menit)	1	2	3	4	5	6	7	8	9	10
Suhu	30	32	34	36	37	38	39	40	41	42







Gambar 4.5 Grafik kenaikan suhu menggunakan 4 buah lampu pijar

Hasil Pengujian Relay 4 Channel





Hasil pengujian *relay* untuk Lampu, *heater*, *exhaust* dan pompa didapat dengan mengukur tegangan *input* dan *output* yang dihasilkan *relay* aktif dan *relay* tidak aktif. Hasil pengujian tegangan *input relay* diaktifkan tertuang pada Tabel 4.3 dan hasil pengujian tegangan *input relay* dinonaktifkan tertuang pada Tabel 4.4. Sedangkan hasil pengujian tegangan *output relay* diaktifkan tertuang pada

Tabel 4.5 dan hasil pengujian tegangan *output relay* dinonaktifkan tertuang pada Tabel 4.6.


Tabel 4.3 Pengujian Tegangan *Input Relay* Diaktifkan


Jenis	Kondisi	Tegangan	Tampilan	Hasil
Lampu	<i>High</i>	4,91 VDC		Baik
<i>Heater</i>	<i>High</i>	4,93 VDC		Baik
<i>Exhaust</i>	<i>High</i>	4,98 VDC		Baik
Pompa	<i>High</i>	4,92 VDC		Baik

Tabel 4.4 Pengujian Tegangan *Input Relay* Dinonaktifkan


Jenis	Kondisi	Tegangan	Tampilan	Hasil
Lampu	<i>Low</i>	0,0V		Baik
<i>Heater</i>	<i>Low</i>	0,0V		Baik
<i>Exhaust</i>	<i>Low</i>	0,0V		Baik
Pompa	<i>Low</i>	0,0V		Baik

Tabel 4.5 Pengujian Tegangan *Output Relay* Diaktifkan

Jenis	Kondisi	Tegangan	Tampilan	Hasil
Lampu	<i>High</i>	200,3 VAC		Baik

<i>Heater</i>	<i>High</i>	199,6 VAC		Baik
<i>Exhaust</i>	<i>High</i>	8,67 VDC		Baik
Pompa	<i>High</i>	4,98 VDC		Baik



Tabel 4.6 Pengujian Tegangan *Output Relay* Dinonaktifkan

Jenis	Kondisi	Tegangan	Tampilan	Hasil
Lampu	<i>Low</i>	0,015 mV		Baik
<i>Heater</i>	<i>Low</i>	0,086 mV		Baik
<i>Exhaust</i>	<i>Low</i>	0,024 mV		Baik
Pompa	<i>Low</i>	0,00 mV		Baik

Hasil Pengujian Akurasi Sensor DHT11

Hasil pengujian *sensor* DHT11 dilakukan dengan mengukur tingkat akurasi sensor DHT11 ketika mendeteksi Suhu dan kelembapan. Pada pengujian *sensor* DHT11 kali ini peneliti melakukan pengujian yaitu dengan menyalakan lampu pada mesin penetas kemudian hasil suhu dan kelembapan akan di tampilkan pada LCD berukuran 16 x 2, lalu membandingkannya dengan hasil suhu dan kelembapan yang terdeteksi oleh alat *ThermoHygro*. Untuk pengujian akurasi *sensor* DHT11 dapat dilihat pada Tabel 4.7.






Tabel 4.7 Pengujian *sensor* DHT11 dengan *ThermoHygro*


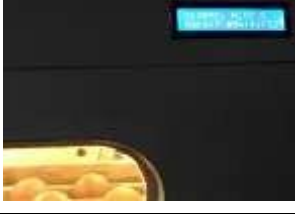




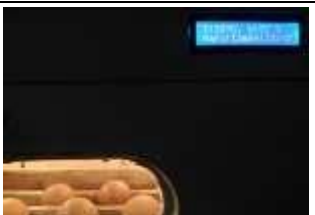
Pembacaan Sensor DHT11	Pembacaan alat ThermoHygro	Tampilan
<p>Suhu 35°C, Kelembapan 64%</p>	<p>Suhu 34,9°C, Kelembapan 64%</p>	
<p>Suhu 37°C, Kelembapan 63%</p>	<p>Suhu 36,9°C, Kelembapan 62%</p>	








Hasil Pengujian Mesin Penetas Telur



Hasil pengujian suhu, kelembapan dan keberhasilan mesin penetas telur yang dilakukan dengan cara memasukan 5 butir telur ayam polandia kedalam mesin penetas yang telah diberi label sebelumnya dengan label telur 1, telur 2, telur 3, telur 4, dan telur 5 untuk ditetaskan terdapat pada Table 4.8.

Tabel 4.8 Pengujian Mesin Penetas Telur

Jadwal Pencatatan	Suhu °C	Kelembapan (%)	Tampilan
Harike 1	38	57%	
Harike 2	38	57%	
Harike 3	38	56%	
Harike 4	38	56%	
Harike 5	39	56%	

Harike 6	39	57%	
Harike 7	39	57%	
Harike 8	39	56%	
Harike 9	40	56%	
Harike 10	40	56%	
Harike 11	38	57%	
Harike 12	38	57%	

Harike 13	39	57%	
Harike 14	39	56%	
Harike 15	39	56%	
Harike 16	39	56%	
Harike 17	39	56%	
Harike 18	39	57%	
Harike 19	39	56%	

Harike 20	39	56%	
Harike 21	40	57%	

Pembahasan

Berdasarkan hasil pengujian secara keseluruhan yang telah dilakukan, diketahui Rancang bangun sistem pengendalian suhu dan kelembapan otomatis mesin penetas telur ayam polandia berbasis arduino berfungsi sesuai dengan perencanaan. Hasil pengujian diketahui melalui pengujian kondisi mesin penetas tanpa sumber pemanas dan dengan sumber pemanas lalu pengujian *relay* lampu, *heater*, *exhaust*, pompa, pengujian sensor suhu, dan pengujian tingkat keberhasilan mesin penetas telur ayam polandia

Pada pengujian kenaikan suhu yang tertuang di Tabel 4.1 diketahui bahwa suhu di dalam box lebih tinggi dari suhu dilingkungan terbuka.

Pada pengujian sumber panas yang tertuang di Tabel 4.2 didapatkan hasil bahwa 4 buah lampu pijar berukuran 5watt bisa menghasilkan panas maksimal hingga 42°C.

Pada pengujian tegangan input saat relay aktif yang tertuang pada Tabel 4.3 dan saat relay tidak aktif yang tertuang pada tabel 4.4 di dapatkan hasil pengujian masing-masing *relay* mempunyai tegangan input sebagai berikut :

1. Tegangan *input relay* saat Lampu aktif adalah 4,91 V
2. Tegangan *input relay* saat *Heater* aktif adalah 4,93 V
3. Tegangan *input relay* saat *Exhaust* aktif adalah 4,98 V
4. Tegangan *input relay* saat Pompa air aktif adalah 4,92 V
5. Tegangan *input relay* saat Lampu tidak aktif adalah 0,0 V
6. Tegangan *input relay* saat *Heater* tidak aktif adalah 0,0 V
7. Tegangan *input relay* saat *Exhaust* tidak aktif adalah 0,0 V
8. Tegangan *input relay* saat Pompa air tidak aktif adalah 0,0 V

Demikian halnya pada pengujian tegangan output saat *relay* aktif yang tertuang pada Tabel 4.5 dan tegangan output pada saat *relay* tidak aktif yang tertuang pada tabel 4.6 didapatkan hasil pengujian masing-masing pada Tabel 4.5 dan Tabel 4.6 hasil pengujian masing-masing *relay* mempunyai tegangan *output* sebagai berikut:

1. Tegangan *output relay* saat Lampu aktif adalah 200,3 VAC
2. Tegangan *output relay* saat *Heater* aktif adalah 199,6 VAC
3. Tegangan *output relay* saat *Exhaust* aktif adalah 8,67 VDC
4. Tegangan *output relay* saat Pompa air aktif adalah 4,98 VDC
5. Tegangan *output relay* saat Lampu tidak aktif adalah 0,015mV
6. Tegangan *output relay* saat *Heater* tidak aktif adalah 0,086mV
7. Tegangan *output relay* saat *Exhaust* tidak aktif adalah 0,024mV
8. Tegangan *output relay* saat Pompa air tidak aktif adalah 0,0mV

Pada Tabel 4.9 hasil pengujian akurasi pembacaan *sensor* suhu DHT11 untuk mengetahui tingkat akurasi dari *sensor* suhu. Setelah dilakukan pengukuran

suhu saat menyalakan sumber pemanas pada mesin penetas telur yang ditampilkan pada LCD 16 x 2cm, tampilan pada LCD 16 x 2cm menampilkan suhu 33°C dan kelembapan 64%. Sedangkan ketika suhu 37°C kelembapan 62%.

Pada table 4.9 hasil pengujian akurasi pembacaan *sensor* suhu DHT11 untuk mengetahui tingkat akurasi dari *sensor*. Setelah dilakukan pengukuran pada kondisi suhu 35°C dan 37°C, kondisi kelembapan 64% dan 63%. Hasil yang didapat ketika suhu 35°C dan kelembapan 64%, pada *Thermohygro* suhu kurang dari 34°C dan kelembapan sama 64% , dan ketika suhu 37°C, pada *Thermohygro* kurang dari 37°C dan kelembapan kurang dari 64%.

Pada Tabel 4.10 hasil pengujian mesin penetas telur untuk mengetahui suhu dan kelembapan serta presentase keberhasilan alat. Setelah dilakukan pengujian mesin penetas telur ayam polandia selama 21 hari dengan menggunakan 5 buah telur, didapatkan hasil bahwa suhu dan kelembapan di dalam mesin penetas stabil. Pengambilan data dilakukan setiap harinya selama masa inkubasi telur setiap puku 04.00 dini hari pada saat suhu di lingkungan berada pada titik terendah, dari Tabel 4.10 dapat diketahui rata-rata suhu di dalam mesin penetas pukul 04.00 adalah 38,85°C dan kelembapan adalah 56,4%. Dari hasil pengujian pada Tabel 4.10 juga diketahui bahwa telur yang berhasil ditetaskan berjumlah 4 butir, sedangkan 1 butir lagi gagal menetas. Dengan rumus presentase keberhasilan sperti pada gambar 4.6 sebagai berikut:

$$\frac{M}{N} \times 100\%$$

Gambar 4.6 Rumus perhitungan Presentase keberhasilan

Maka dapat dihitung presentase keberhasilan mesin penetas telur sebagai berikut:

$$\frac{4}{5} \times 100\% = 80\%$$

Berdasarkan hasil pengujian secara keseluruhan yang telah dilakukan, diketahui bahwa Rancang bangun sistem pengendalian suhu dan kelembapan otomatis mesin penetas telur ayam polandia berbasis arduino bekerja dengan baik dan dinyatakan berhasil menetas telur ayam polandia dengan tingkat keberhasilan mencapai 80%

Aplikasi Hasil Penelitian

Alat ini merupakan salah satu alat yang mempunyai nilai ekonomis, dinamis, efisien, praktis dan daya jual yang tinggi. Alat ini dapat dimanfaatkan bagi pengusaha terutama peternak yang bergerak pada bidang hewan peliharaan khususnya hewan hias. Keberhasilan menetas telur dapat menggantikan peran induk yang enggan mengerami telurnya dan dapat menaikkan tingkat kelahiran unggas khususnya ayam hias untuk di budidayakan yang memiliki nilai jual tinggi dibidang komersil.

BAB V

KESIMPULAN DAN SARAN

KESIMPULAN

Dalam penelitian rancang bangun sistem pengendali suhu dan kelembapan otomatis pada mesin penetas telur ayam polandia berbasis arduino ini sudah sesuai dengan tujuan dari penelitian yang dilakukan, yaitu :

1. Rancang bangun sistem pengendali suhu dan kelembapan otomatis pada mesin penetas telur ayam polandia berbasis arduino ini berhasil mengendalikan suhu 38 – 40 °c dan kelembapan 50 – 65 %.
2. Rancang bangun sistem pengendalian mesin penetas telur otomatis berbasis arduino ini dapat menetas telur ayam polandia dengan tingkat keberhasilan yaitu 80%.

SARAN

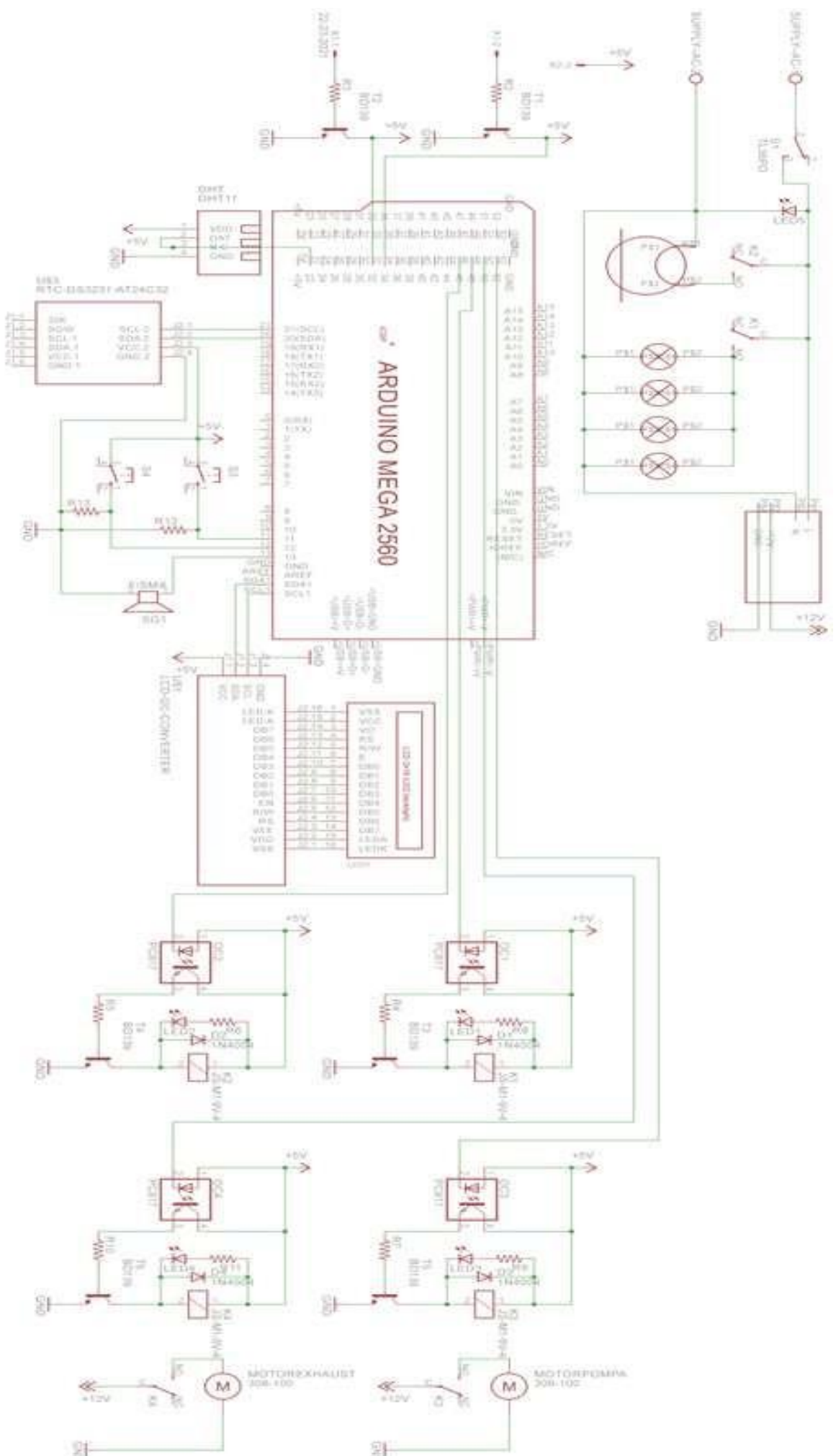
Berdasarkan kesimpulan yang ditarik dari analisis data, maka penulis mencoba untuk memberikan saran dalam pengembangan penelitian ini adalah:

1. Penggunaan sensor DHT11 harus diletakan sejajar dengan tempat peletakan telur.
2. Gunakan sensor suhu dan kelembapan yang lebih presisi dan bisa digunakan jangka panjang dalam waktu yang lama misalnya sensor DHT22.
3. Pemilihan telur yang baik yang akan ditetaskan agar memperoleh tingkat keberhasilan penetasan yang lebih baik.

4. Mengembangkan proses pembalikan telur agar dapat dilakukan secara otomatis

DAFTAR PUSTAKA

- Arduino. (2005). Arduino – introduction. Diambil 28 Desember 2016, dari <http://www.arduino.cc/en/Guide/introduction>.
- Bishop, O. (2004). Dasar-dasar Elektronika. Jakarta: Erlangga.
- [Depdiknas] Departemen Pendidikan Nasional. (2008). Kamus Besar Bahasa Indonesia. Jakarta: Balai Pustaka.
- [FT] Fakultas Teknik. (2015). Buku Panduan Skripsi dan Non Skripsi. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta
- Gerai Cerdas. (2016). Sensor DHT11. Diambil 30 Desember 2016, dari <http://www.geraicerdas.com/sensor/temperature/dht11-sensor-suhu-dan-kelembaban-detail>.
- Hannah, Ray. (2007). Ayam. Solo: Tiga Serangkai Pustaka mandiri
- Hartono, T., Isman. (2012). Kiat Sukses Menetas Telur Ayam. Jakarta: PT Agro Media Pustaka.
- Hastono, Tri. (2009). Thesis Sistem Pengendali dan pengukur suhu pada mesin penetas telur berbasis mikrokontroler AT89S51
- Irfan, M., Malekhi, A., Mulyana, R., Susanto, R. (2011). Jurnal PERANCANGAN SISTEM PENERAM TELUR AYAM OTOMATIS. Diambil 20 Desember 2016 dari http://research-dashboard.binus.ac.id/uploads/paper/document/publication/Journal/Teknik%20Komputer/Vol%2019%20No%202%20Agustus%202011/07_Rudy%20_telur%20ayam-ok.pdf. Jakarta
- Marimin, Tanjung, M., Prabowo, H. (2006). Sistem informasi manajemen sumber daya manusia. Jakarta : PT Gramedia Widiasarana Indonesia.
- Rahayu, I., Sudaryani, T., Santosa, H. (2011). Panduan Lengkap Ayam. Jakarta: Penebar Swadaya.
- Rawung, A. E. (2013). Perencanaan sistem control untuk SMK/MAK kelas X semester 1. Jakarta: Kemendikbud.
- Rose, S.P. (2001). *Principles of poultry science*. CAB International
- Syahwil, M. (2013). Panduan Mudah Simulasi & Praktek Mikrokontroler arduino. (T.A. Prabawati, Ed.) (Edisi 1). Yogyakarta: Andi
- Ugear.com. (2017). <http://www.ugear.com/portfolio/dht11-humidity-temperature-sensor-module/>
- Wakhid, A. (2014). Membuat Sendiri Mesin Tetes Praktis. Jakarta : PT Agromedia Pustaka.



Arduino MEGA 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical Specifications

Page 2

How to use Arduino
Programming Enviroment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Enviromental Policies
half sqm of green via Impatto Zero®

Page 7



RADIOSPARES

RADIONICS



Technical Specification

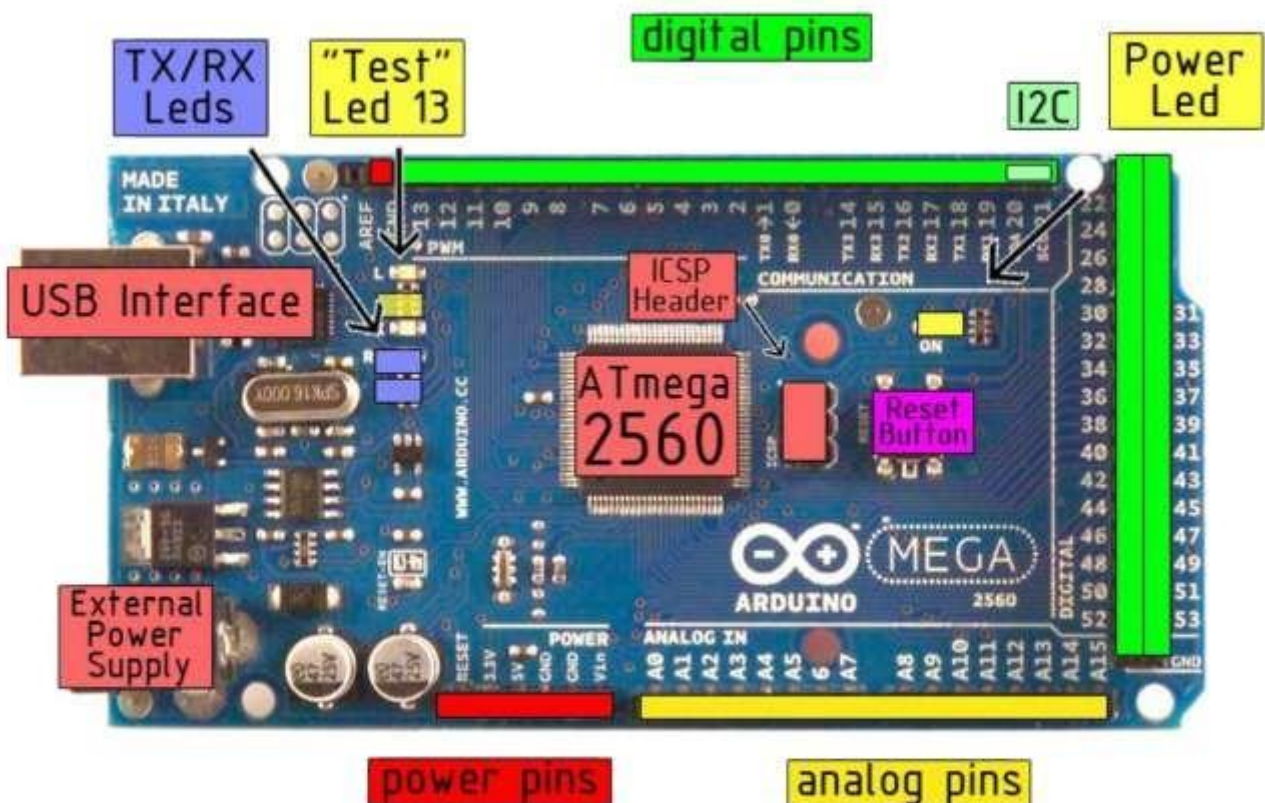


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



RADIOSPARES

RADIONICS



The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



RADIOSPARES

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto -reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**



radiospares

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



Blinking Led!



RADIOSPARES

RADIONICS



Terms & Conditions



1. Warranties

The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



radiospares

RADIONICS



Digital-output relative humidity & temperature sensor/module – DHT11



Resistive-type humidity and temperature module/sensor

1. Feature & Application:

* Full range temperature compensated * Relative humidity and temperature measurement *
Calibrated digital signal *Outstanding long-term stability *Extra components not needed *
Long transmission distance * Low power consumption *4 pins packaged and fully
interchangeable

2. Description:

DHT11 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

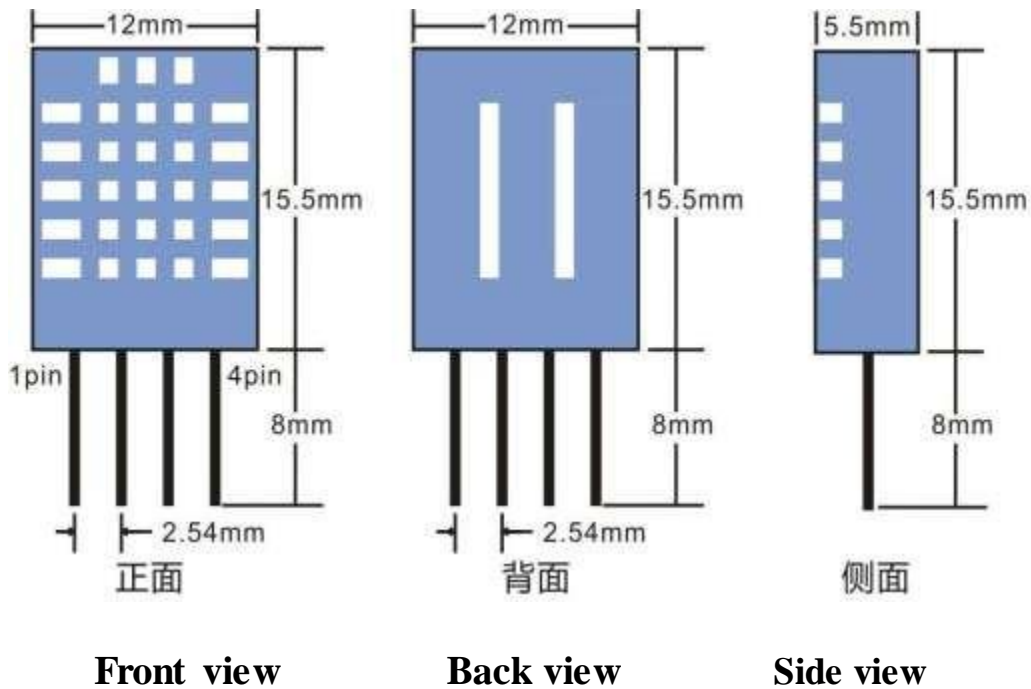
Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in OTP memory.

Small size & low consumption & long transmission distance(20m) enable DHT11 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

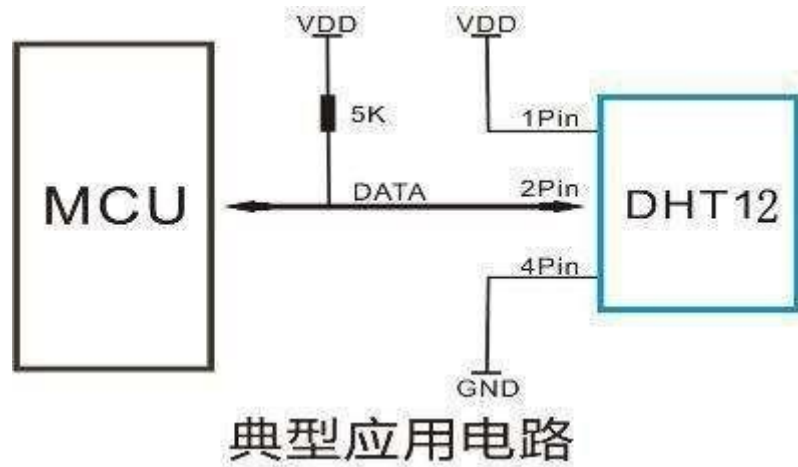
3. Technical Specification:

Model	DHT11	
Power supply	3-5.5V DC	
Output signal	digital signal via single-bus	
Sensing element	Polymer resistor	
Measuring range	humidity 20-90%RH; temperature 0-50 Celsius	
Accuracy	humidity +-4%RH (Max +-5%RH); temperature +-2.0Celsius	
Resolution or sensitivity	humidity 1%RH;	temperature 0.1Celsius
Repeatability	humidity +-1%RH;	temperature +-1Celsius
Humidity hysteresis	+-1%RH	
Long-term Stability	+-0.5%RH/year	
Sensing period	Average: 2s	
Interchangeability	fully interchangeable	
Dimensions	size 12*15.5*5.5mm	

4. Dimensions: (unit----mm)



5. Typical application



3Pin=NULL, MCU=Microcomputer or single-chip

6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3-5.5V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

(2) Communication and signal

Single-bus data is used for communication between MCU and DHT11.

7. Electrical Characteristics:

Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3	5	5.5	V
Current supply	Measuring	0.5		2.5	mA
	Stand-by	100	Null	150	uA
	Average	0.2	Null	1	mA

General Description

The DS3231 is a low-cost, extremely accurate I²C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I²C bidirectional bus.

A precision temperature-compensated voltage reference and comparator circuit monitors the status of VCC to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the RST pin is monitored as a pushbutton input for generating a µP reset.

Benefits and Features

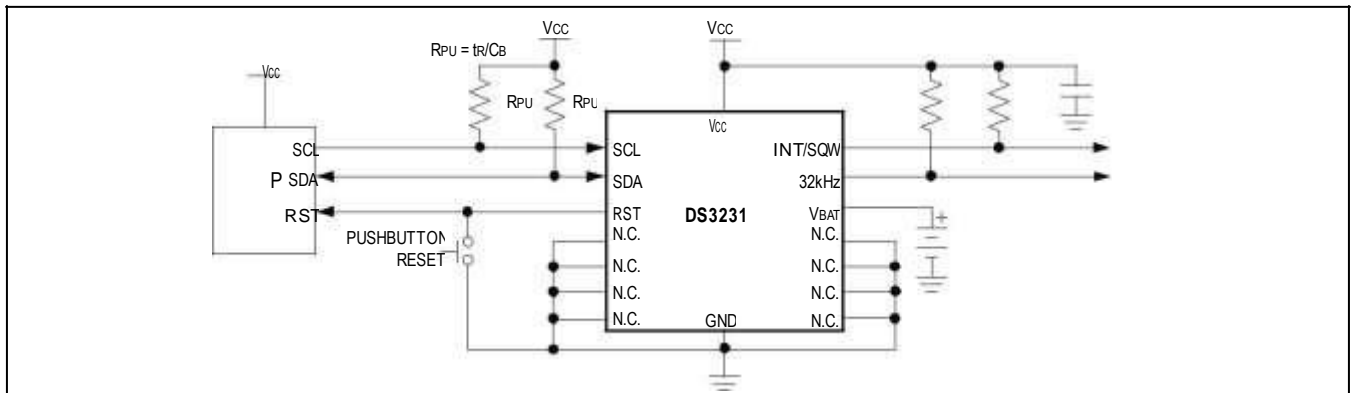
- Highly Accurate RTC Completely Manages All Timekeeping Functions
 - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100
 - Accuracy ±2ppm from 0°C to +40°C
 - Accuracy ±3.5ppm from -40°C to +85°C
 - Digital Temp Sensor Output: ±3°C Accuracy
 - Register for Aging Trim
 - RST Output/Pushbutton Reset Debounce Input
 - Two Time-of-Day Alarms
 - Programmable Square-Wave Output Signal
- Simple Serial Interface Connects to Most Microcontrollers
 - Fast (400kHz) I²C Interface
- Battery-Backup Input for Continuous Timekeeping
 - Low Power Operation Extends Battery-Backup Run Time
 - 3.3V Operation
- Operating Temperature Ranges: Commercial (0°C to +70°C) and Industrial (-40°C to +85°C)
- Underwriters Laboratories[®] (UL) Recognized

Applications

- Servers
- Telematics
- Utility Power Meters
- GPS

Ordering Information and Pin Configuration appear at end of data sheet.

Typical Operating Circuit



Underwriters Laboratories is a registered certification mark of Underwriters Laboratories Inc.



Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground.....-0.3V to +6.0V

Junction-to-Ambient Thermal Resistance (θ_{JA}) (Note 1).73°C/W

Junction-to-Case Thermal Resistance (θ_{JC}) (Note 1).....23°C/W

Operating Temperature Range

DS3231S.....0°C to +70°C

DS3231SN..... -40°C to +85°C

Junction Temperature..... +125°C

Storage Temperature Range..... -40°C to +85°C

Lead Temperature (soldering, 10s)..... +260°C

Soldering Temperature (reflow, 2 times max)..... +260°C

(see the *Handling, PCB Layout, and Assembly* section)

Note 1: Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to www.maximintegrated.com/thermal-tutorial.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Recommended Operating Conditions

(TA = TMIN to TMAX, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		2.3	3.3	5.5	V
	V_{BAT}		2.3	3.0	5.5	V
Logic 1 Input SDA, SCL	V_{IH}		$0.7 \times V_{CC}$		$V_{CC} + 0.3$	V
Logic 0 Input SDA, SCL	V_{IL}		-0.3		$0.3 \times V_{CC}$	V

Electrical Characteristics

(VCC = 2.3V to 5.5V, VCC = Active Supply (see Table 1), TA = TMIN to TMAX, unless otherwise noted.) (Typical values are at VCC = 3.3V, VBAT = 3.0V, and TA = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Supply Current	I_{CCA}	(Notes 4, 5)	VCC = 3.63V		200	μA
			VCC = 5.5V		300	
Standby Supply Current	I_{CCS}	\overline{I}^2C bus inactive, 32kHz output on, SQW output off (Note 5)	VCC = 3.63V		110	μA
			VCC = 5.5V		170	
Temperature Conversion Current	$I_{CCSCONV}$	\overline{I}^2C bus inactive, 32kHz output on, SQW output off	VCC = 3.63V		575	μA
			VCC = 5.5V		650	
Power-Fail Voltage	V_{PF}		2.45	2.575	2.70	V
Logic 0 Output, 32kHz, INT/SQW, SDA	V_{OL}	IOL = 3mA			0.4	V
Logic 0 Output, RST	V_{OL}	IOL = 1mA			0.4	V
Output Leakage Current 32kHz, INT/SQW, SDA	I_{LO}	Output high impedance	-1	0	+1	μA
Input Leakage SCL	I_{LI}		-1		+1	μA
RST Pin I/O Leakage	I_{OL}	RST high impedance (Note 6)	-200		+10	μA
VBAT Leakage Current (VCC Active)	I_{BATLKG}			25	100	nA

Electrical Characteristics (continued)

(VCC = 2.3V to 5.5V, VCC = Active Supply (see Table 1), TA = TMIN to TMAX, unless otherwise noted.) (Typical values are at VCC = 3.3V, VBAT = 3.0V, and TA = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Frequency	f_{OUT}	VCC = 3.3V or VBAT = 3.3V		32.768		kHz
Frequency Stability vs. Temperature (Commercial)	$\Delta f/f_{OUT}$	VCC = 3.3V or VBAT = 3.3V, aging offset = 00h	0°C to +40°C		±2	ppm
			>40°C to +70°C		±3.5	
Frequency Stability vs. Temperature (Industrial)	$\Delta f/f_{OUT}$	VCC = 3.3V or VBAT = 3.3V, aging offset = 00h	-40°C to <0°C		±3.5	ppm
			0°C to +40°C		±2	
			>40°C to +85°C		±3.5	
Frequency Stability vs. Voltage	$\Delta f/V$			1		ppm/V
Trim Register Frequency Sensitivity per LSB	$\Delta f/LSB$	Specified at:	-40°C	0.7		ppm
			+25°C	0.1		
			+70°C	0.4		
			+85°C	0.8		
Temperature Accuracy	Temp	VCC = 3.3V or VBAT = 3.3V	-3		+3	°C
Crystal Aging	$\Delta f/f_O$	After reflow, not production tested	First year	±1.0		ppm
			0–10 years	±5.0		

Electrical Characteristics

(VCC = 0V, VBAT = 2.3V to 5.5V, TA = TMIN to TMAX, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Battery Current	I_{BATA}	EOSC = 0, BBSQW = 0, SCL = 400kHz (Note 5)	VBAT = 3.63V		70	µA
			VBAT = 5.5V		150	
Timekeeping Battery Current	I_{BATT}	EOSC = 0, BBSQW = 0, EN32kHz = 1, SCL = SDA = 0V or SCL = SDA = VBAT (Note 5)	VBAT = 3.63V	0.84	3.0	µA
			VBAT = 5.5V	1.0	3.5	
Temperature Conversion Current	I_{BATTC}	EOSC = 0, BBSQW = 0, SCL = SDA = 0V or SCL = SDA = VBAT	VBAT = 3.63V		575	µA
			VBAT = 5.5V		650	
Data-Retention Current	I_{BATDR}	EOSC = 1, SCL = SDA = 0V, +25°C			100	nA

AC Electrical Characteristics

(VCC = VCC(MIN) to VCC(MAX) or VBAT = VBAT(MIN) to VBAT(MAX), VBAT > VCC, TA = TMIN to TMAX, unless otherwise noted.) (Note 2)

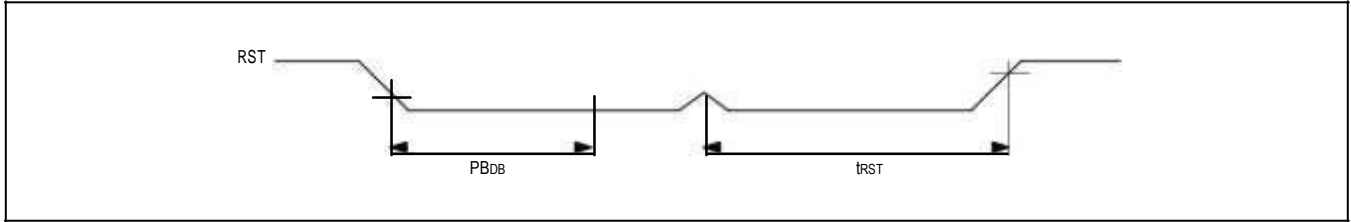
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f SCL	Fast mode	100		400	kHz
		Standard mode	0		100	
Bus Free Time Between STOP and START Conditions	t BUF	Fast mode	1.3			μs
		Standard mode	4.7			
Hold Time (Repeated) START Condition (Note 7)	t HD:STA	Fast mode	0.6			μs
		Standard mode	4.0			
Low Period of SCL Clock	t LOW	Fast mode	1.3			μs
		Standard mode	4.7			
High Period of SCL Clock	t HIGH	Fast mode	0.6			μs
		Standard mode	4.0			
Data Hold Time (Notes 8, 9)	t HD:DAT	Fast mode	0		0.9	μs
		Standard mode	0		0.9	
Data Setup Time (Note 10)	t SU:DAT	Fast mode	100			ns
		Standard mode	250			
START Setup Time	t SU:STA	Fast mode	0.6			μs
		Standard mode	4.7			
Rise Time of Both SDA and SCL Signals (Note 11)	tR	Fast mode	20 +		300	ns
		Standard mode	0.1CB		1000	
Fall Time of Both SDA and SCL Signals (Note 11)	tF	Fast mode	20 +		300	ns
		Standard mode	0.1CB		300	
Setup Time for STOP Condition	t SU:STO	Fast mode	0.6			μs
		Standard mode	4.7			
Capacitive Load for Each Bus Line	CB	(Note 11)			400	pF
Capacitance for SDA, SCL	C I/O			10		pF
Pulse Width of Spikes That Must Be Suppressed by the Input Filter	t SP			30		ns
Pushbutton Debounce	PBDB			250		ms
Reset Active Time	t RST			250		ms
Oscillator Stop Flag (OSF) Delay	t OSF	(Note 12)		100		ms
Temperature Conversion Time	t CONV			125	200	ms

Power-Switch Characteristics

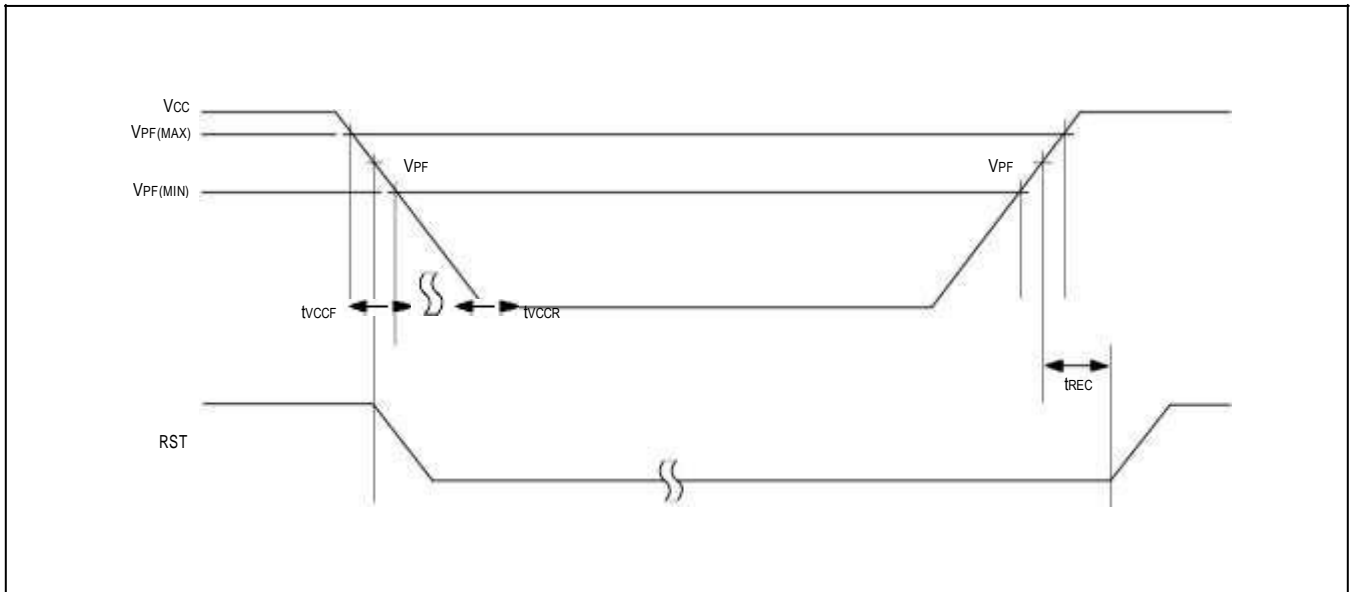
(TA = TMIN to TMAX)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
VCC Fall Time; VPF(MAX) to VPF(MIN)	t VCCF		300			μs
VCC Rise Time; VPF(MIN) to VPF(MAX)	t VCCR		0			μs
Recovery at Power-Up	t REC	(Note 13)		250	300	ms

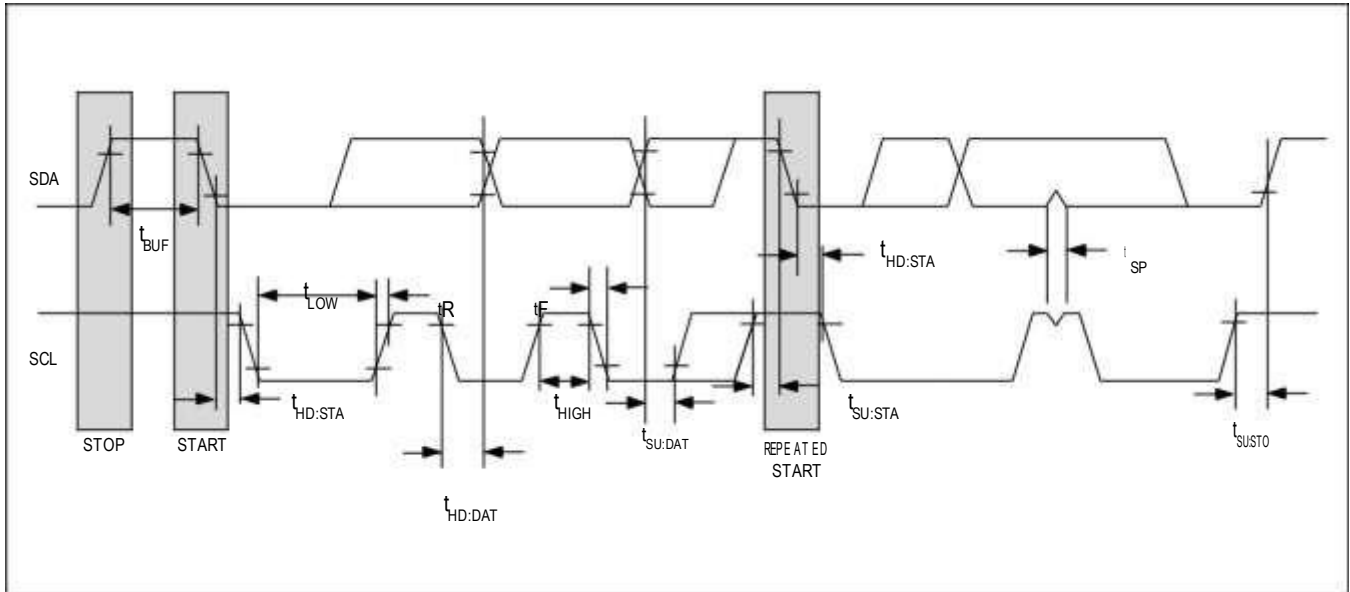
Pushbutton Reset Timing



Power-Switch Timing



Data Transfer on I2C Serial Bus



WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

Note 2: Limits at -40°C are guaranteed by design and not production tested.

Note 3: All voltages are referenced to ground.

Note 4: ICCA—SCL clocking at max frequency = 400kHz.

Note 5: Current is the averaged input current, which includes the temperature conversion current.

Note 6: The RST pin has an internal 50kΩ (nominal) pullup resistor to VCC.

Note 7: After this period, the first clock pulse is generated.

Note 8: A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the $V_{IH}(\text{MIN})$ of the SCL signal) to bridge the undefined region of the falling edge of SCL.

Note 9: The maximum $t_{HD:DAT}$ needs only to be met if the device does not stretch the low period (t_{LOW}) of the SCL signal.

Note 10: A fast-mode device can be used in a standard-mode system, but the requirement $t_{SU:DAT} \geq 250\text{ns}$ must then be met. This is automatically the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line $t_{R}(\text{MAX}) + t_{SU:DAT} = 1000 + 250 = 1250\text{ns}$ before the SCL line is released.

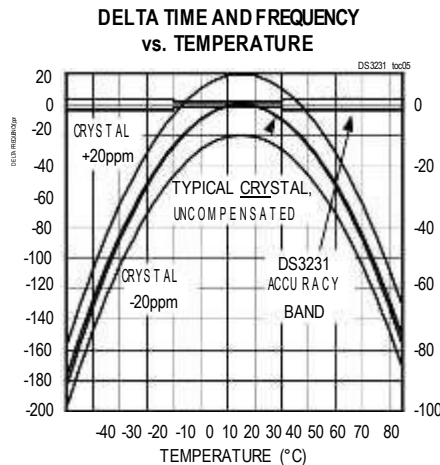
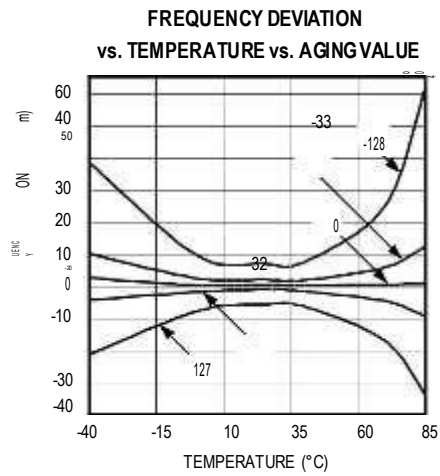
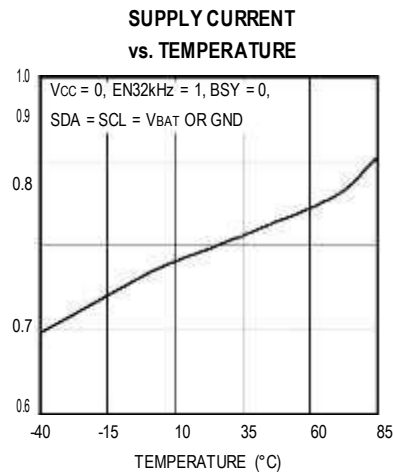
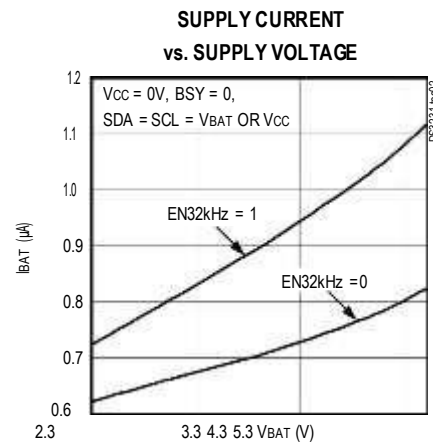
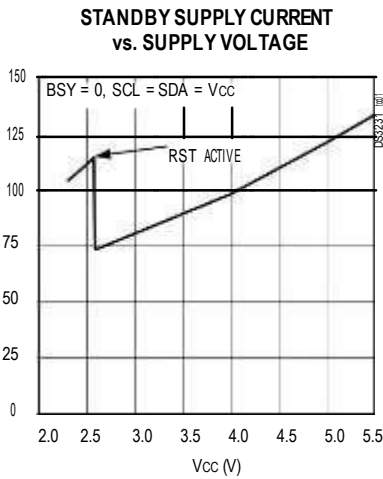
Note 11: CB—total capacitance of one bus line in pF.

Note 12: The parameter t_{OSF} is the period of time the oscillator must be stopped for the OSF flag to be set over the voltage range of $0.0\text{V} \leq V_{CC} \leq V_{CC}(\text{MAX})$ and $2.3\text{V} \leq V_{BAT} \leq 3.4\text{V}$.

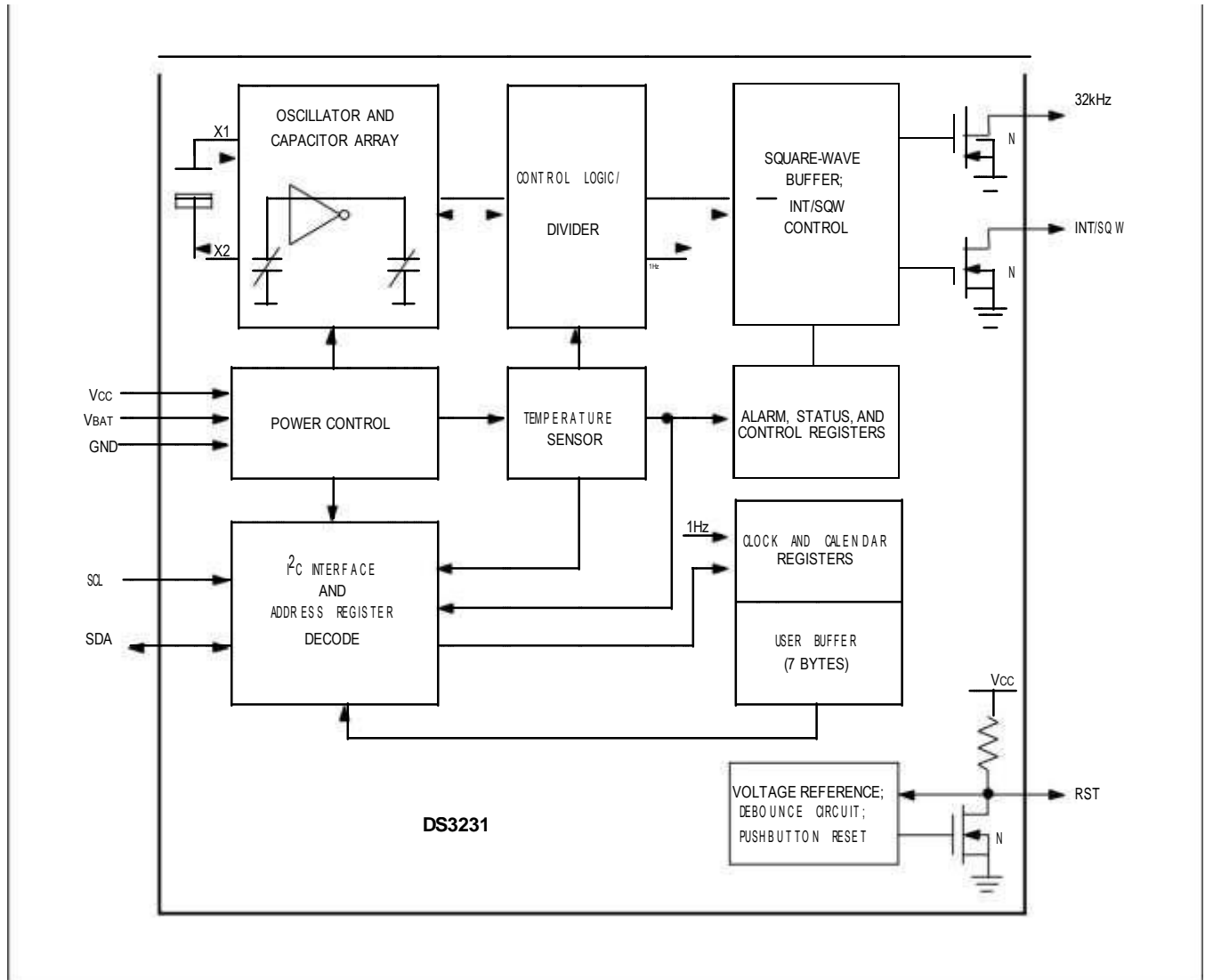
Note 13: This delay applies only if the oscillator is enabled and running. If the EOSC bit is a 1, t_{REC} is bypassed and RST immediately goes high. The state of RST does not affect the I²C interface, RTC, or TCXO.

Typical Operating Characteristics

(VCC = +3.3V, TA = +25°C, unless otherwise noted.)



Block Diagram



Pin Description

PIN	NAME	FUNCTION
1	32kHz	32kHz Output. This open-drain pin requires an external pullup resistor. When enabled, the output operates on either power supply. It may be left open if not used.
2	V _{CC}	DC Power Pin for Primary Power Supply. This pin should be decoupled using a 0.1µF to 1.0µF capacitor. If not used, connect to ground.
3	INT/SQW	Active-Low Interrupt or Square-Wave Output. This open-drain pin requires an external pullup resistor connected to a supply at 5.5V or less. This multifunction pin is determined by the state of the INTCN bit in the Control Register (0Eh). When INTCN is set to logic 0, this pin outputs a square wave and its frequency is determined by RS2 and RS1 bits. When INTCN is set to logic 1, then a match between the timekeeping registers and either of the alarm registers activates the INT/SQW pin (if the alarm is enabled). Because the INTCN bit is set to logic 1 when power is first applied, the pin defaults to an interrupt output with alarms disabled. The pullup voltage can be up to 5.5V, regardless of the voltage on V _{CC} . If not used, this pin can be left unconnected.
4	RST	Active-Low Reset. This pin is an open-drain input/output. It indicates the status of V _{CC} relative to the VPF specification. As V _{CC} falls below VPF, the RST pin is driven low. When V _{CC} exceeds VPF, for t _{RST} , the RST pin is pulled high by the internal pullup resistor. The active-low, open-drain output is combined with a debounced pushbutton input function. This pin can be activated by a pushbutton reset request. It has an internal 50kΩ nominal value pullup resistor to V _{CC} . No external pullup resistors should be connected. If the oscillator is disabled, t _{REC} is bypassed and RST immediately goes high.
5–12	N.C.	No Connection. Must be connected to ground.
13	GND	Ground
14	V _{BAT}	Backup Power-Supply Input. When using the device with the V _{BAT} input as the primary power source, this pin should be decoupled using a 0.1µF to 1.0µF low-leakage capacitor. When using the device with the V _{BAT} input as the backup power source, the capacitor is not required. If V _{BAT} is not used, connect to ground. The device is UL recognized to ensure against reverse charging when used with a primary lithium battery. Go to www.maximintegrated.com/qa/info/ul .
15	SDA	Serial Data Input/Output. This pin is the data input/output for the I ² C serial interface. This open-drain pin requires an external pullup resistor. The pullup voltage can be up to 5.5V, regardless of the voltage on V _{CC} .
16	SCL	Serial Clock Input. This pin is the clock input for the I ² C serial interface and is used to synchronize data movement on the serial interface. Up to 5.5V can be used for this pin, regardless of the voltage on V _{CC} .

Detailed Description

The DS3231 is a serial RTC driven by a temperature-compensated 32kHz crystal oscillator. The TCXO provides a stable and accurate reference clock, and maintains the RTC to within ±2 minutes per year accuracy from -40°C to +85°C. The TCXO frequency output is available at the 32kHz pin. The RTC is a low-power clock/calendar with two programmable time-of-day alarms and a programmable square-wave output. The INT/SQW provides either an interrupt signal due to alarm conditions or a square-wave output. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap

year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. The internal registers are accessible through an I²C bus interface.

A temperature-compensated voltage reference and comparator circuit monitors the level of V_{CC} to detect power failures and to automatically switch to the backup supply when necessary. The RST pin provides an external pushbutton function and acts as an indicator of a power-fail event.

Operation

The block diagram shows the main elements of the DS3231. The eight blocks can be grouped into four functional groups: TCXO, power control, pushbutton function, and RTC. Their operations are described separately in the following sections.

32kHz TCXO

The temperature sensor, oscillator, and control logic form the TCXO. The controller reads the output of the on-chip temperature sensor and uses a lookup table to determine the capacitance required, adds the aging correction in AGE register, and then sets the capacitance selection registers. New values, including changes to the AGE register, are loaded only when a change in the temperature value occurs, or when a user-initiated temperature conversion is completed. Temperature conversion occurs on initial application of VCC and once every 64 seconds afterwards.

Power Control

This function is provided by a temperature-compensated voltage reference and a comparator circuit that monitors the VCC level. When VCC is greater than VPF, the part is powered by VCC. When VCC is less than VPF but greater than VBAT, the DS3231 is powered by VCC. If VCC is less than VPF and is less than VBAT, the device is powered by VBAT. See Table 1.

Table 1. Power Control

SUPPLY CONDITION	ACTIVE SUPPLY
$V_{CC} < V_{PF}, V_{CC} < V_{BAT}$	V_{BAT}
$V_{CC} < V_{PF}, V_{CC} > V_{BAT}$	V_{CC}
$V_{CC} > V_{PF}, V_{CC} < V_{BAT}$	V_{CC}
$V_{CC} > V_{PF}, V_{CC} > V_{BAT}$	V_{CC}

To preserve the battery, the first time VBAT is applied to the device, the oscillator will not start up until VCC exceeds VPF, or until a valid I²C address is written to the part. Typical oscillator startup time is less than one second. Approximately 2 seconds after VCC is applied, or a valid I²C address is written, the device makes a temperature measurement and applies the calculated correction to the oscillator. Once the oscillator is running, it continues to run as long as a valid power source is

available (VCC or VBAT), and the device continues to measure the temperature and correct the oscillator frequency every 64 seconds.

On the first application of power (VCC) or when a valid I²C address is written to the part (VBAT), the time and date registers are reset to 01/01/00 01 00:00:00 (DD/MM/YY DOW HH:MM:SS).

VBAT Operation

There are several modes of operation that affect the amount of VBAT current that is drawn. While the device

is powered by VBAT and the serial interface is active, active battery current, IBATA, is drawn. When the serial interface is inactive, timekeeping current (IBATT), which includes the averaged temperature conversion current, IBATTC, is used (refer to Application Note 3644: *Power Considerations for Accurate Real-Time Clocks*

for details). Temperature conversion current, IBATTC, is specified since the system must be able to support the periodic higher current pulse and still maintain a valid voltage level. Data retention current, IBATTD, is the current drawn by the part when the oscillator is stopped (EOSC = 1). This mode can be used to minimize battery requirements for times when maintaining time and date information is not necessary, e.g., while the end system is waiting to be shipped to a customer.

Pushbutton Reset Function

The DS3231 provides for a pushbutton switch to be connected to the RST output pin. When the DS3231 is not in a reset cycle, it continuously monitors the RST signal for a low going edge. If an edge transition is detected, the DS3231 debounces the switch by pulling the RST low.

After the internal timer has expired (PBDB), the DS3231 continues to monitor the RST line. If the line is still low, the DS3231 continuously monitors the line looking for a rising edge. Upon detecting release, the DS3231 forces

the RST pin low and holds it low for tRST.

RST is also used to indicate a power-fail condition. When VCC is lower than VPF, an internal power-fail signal is generated, which forces the RST pin low. When VCC returns to a level above VPF, the RST pin is held low for approximately 250ms (tREC) to allow the power supply to stabilize. If the oscillator is not running (see the *Power Control* section) when VCC is applied, tREC is bypassed and RST immediately goes high. Assertion of the RST output, whether by pushbutton or power-fail detection, does not affect the internal operation of the DS3231.

Real-Time Clock

With the clock source from the TCXO, the RTC provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator.

The clock provides two programmable time-of-day alarms and a programmable square-wave output. The INT/SQW pin either generates an interrupt due to alarm condition or outputs a square-wave signal and the selection is controlled by the bit INTCN.

ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE
00h	0	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12/24	AM/PM	10 Hour	Hour				Hours	1–12 + AM/PM 00–23
			20 Hour							
03h	0	0	0	0	0	Day			Day	1–7
04h	0	0	10 Date		Date				Date	01–31
05h	Century	0	0	10 Month	Month				Month/ Century	01–12 + Century
06h	10 Year				Year				Year	00–99
07h	A1M1	10 Seconds			Seconds				Alarm 1 Seconds	00–59
08h	A1M2	10 Minutes			Minutes				Alarm 1 Minutes	00–59
09h	A1M3	12/24	AM/PM	10 Hour	Hour				Alarm 1 Hours	1–12 + AM/PM 00–23
			20 Hour							
0Ah	A1M4	DY/DT	10 Date		Day				Alarm 1 Day	1–7
					Date				Alarm 1 Date	1–31
0Bh	A2M2	10 Minutes			Minutes				Alarm 2 Minutes	00–59
0Ch	A2M3	12/24	AM/PM	10 Hour	Hour				Alarm 2 Hours	1–12 + AM/PM 00–23
			20 Hour							
0Dh	A2M4	DY/DT	10 Date		Day				Alarm 2 Day	1–7
					Date				Alarm 2 Date	1–31
0Eh	EOSC	BBSQW	CONV	RS2	RS1	INTCN	A2IE	A1IE	Control	—
0Fh	OSF	0	0	0	EN32kHz	BSY	A2F	A1F	Control/Status	—
10h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	Aging Offset	—
11h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	MSB of Temp	—
12h	DATA	DATA	0	0	0	0	0	0	LSB of Temp	—

Figure 1. Timekeeping Registers

Note: Unless otherwise specified, the registers' state is not defined when power is first applied.

Address Map

Figure 1 shows the address map for the DS3231 timekeeping registers. During a multibyte access, when the address pointer reaches the end of the register space (12h), it wraps around to location 00h. On an I²C START or address pointer incrementing to location 00h, the current time is transferred to a second set of registers. The time information is read from these secondary registers, while the clock may continue to run. This eliminates the need to reread the registers in case the main registers update during a read.

I²C Interface

The I²C interface is accessible whenever either VCC or VBAT is at a valid level. If a microcontroller connected

to the DS3231 resets because of a loss of VCC or other event, it is possible that the microcontroller and DS3231 I²C communications could become unsynchronized, e.g., the microcontroller resets while reading data from the DS3231. When the microcontroller resets, the DS3231 I²C interface may be placed into a known state by toggling SCL until SDA is observed to be at a high level. At that point the microcontroller should pull SDA low while SCL is high, generating a START condition.

Clock and Calendar

The time and calendar information is obtained by reading the appropriate register bytes. Figure 1 illustrates the RTC registers. The time and calendar data are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the binary-coded

decimal (BCD) format. The DS3231 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic-high being PM. In the 24-hour mode, bit 5 is the 20-hour bit (20–23 hours). The century bit (bit 7 of the month register) is toggled when the years register overflows from 99 to 00.

The day-of-week register increments at midnight. Values that correspond to the day of week are user-defined but must be sequential (i.e., if 1 equals Sunday, then 2 equals Monday, and so on). Illogical time and date entries result in undefined operation.

When reading or writing the time and date registers, secondary (user) buffers are used to prevent errors when the internal registers update. When reading the time and date registers, the user buffers are synchronized to the internal registers on any START and when the register pointer rolls over to zero. The time information is read from these secondary registers, while the clock continues to run. This eliminates the need to reread the registers in case the main registers update during a read.

The countdown chain is reset whenever the seconds register is written. Write transfers occur on the acknowledge from the DS3231. Once the countdown chain is reset, to avoid rollover issues the remaining time and date registers must be written within 1 second. The 1Hz square-wave output, if enabled, transitions high 500ms after the seconds data transfer, provided the oscillator is already running.

Alarms

The DS3231 contains two time-of-day/date alarms. Alarm 1 can be set by writing to registers 07h to 0Ah. Alarm 2 can be set by writing to registers 0Bh to 0Dh. The alarms can be programmed (by the alarm enable and INTCN bits of the control register) to activate the INT/SQW output on an alarm match condition. Bit 7 of each of the time-of-day/date alarm registers are mask bits (Table 2). When all the mask bits for each alarm are logic 0, an alarm only occurs when the values in the timekeeping registers match the corresponding values stored in the time-of-day/date alarm registers. The alarms can also be programmed to repeat every second, minute, hour, day, or date. Table 2 shows the possible settings. Configurations not listed in the table will result in illogical operation.

The DY/DT bits (bit 6 of the alarm day/date registers) control whether the alarm value stored in bits 0 to 5 of that register reflects the day of the week or the date of the month. If DY/DT is written to logic 0, the alarm will be the result of a match with date of the month. If DY/DT is written to logic 1, the alarm will be the result of a match with day of the week.

When the RTC register values match alarm register settings, the corresponding Alarm Flag 'A1F' or 'A2F' bit is set to logic 1. If the corresponding Alarm Interrupt Enable 'A1IE' or 'A2IE' is also set to logic 1 and the INTCN bit is set to logic 1, the alarm condition will activate the INT/SQW signal. The match is tested on the once-per-second update of the time and date registers.

Table 2. Alarm Mask Bits

DY/DT	ALARM 1 REGISTER MASK BITS (BIT 7)				ALARM RATE
	A1M4	A1M3	A1M2	A1M1	
X	1	1	1	1	Alarm once per second
X	1	1	1	0	Alarm when seconds match
X	1	1	0	0	Alarm when minutes and seconds match
X	1	0	0	0	Alarm when hours, minutes, and seconds match
0	0	0	0	0	Alarm when date, hours, minutes, and seconds match
1	0	0	0	0	Alarm when day, hours, minutes, and seconds match
DY/DT	ALARM 2 REGISTER MASK BITS (BIT 7)			ALARM RATE	
	A2M4	A2M3	A2M2		
X	1	1	1	Alarm once per minute (00 seconds of every minute)	
X	1	1	0	Alarm when minutes match	
X	1	0	0	Alarm when hours and minutes match	
0	0	0	0	Alarm when date, hours, and minutes match	
1	0	0	0	Alarm when day, hours, and minutes match	

Control Register (0Eh)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	EOSC	BBSQW	CONV	RS2	RS1	INTCN	A2IE	A1IE
POR:	0	0	0	1	1	1	0	0

Special-Purpose Registers

The DS3231 has two additional registers (control and status) that control the real-time clock, alarms, and square-wave output.

Control Register (0Eh)

Bit 7: Enable Oscillator (EOSC). When set to logic 0, the oscillator is started. When set to logic 1, the oscillator is stopped when the DS3231 switches to VBAT. This bit is clear (logic 0) when power is first applied. When the

DS3231 is powered by VCC, the oscillator is always on regardless of the status of the EOSC bit. When EOSC is disabled, all register data is static.

Bit 6: Battery-Backed Square-Wave Enable (BBSQW).

When set to logic 1 with INTCN = 0 and VCC < VPF, this bit enables the square wave. When BBSQW is logic 0, the INT/SQW pin goes high impedance when VCC < VPF. This bit is disabled (logic 0) when power is first applied.

Bit 5: Convert Temperature (CONV). Setting this bit to 1 forces the temperature sensor to convert the temperature into digital code and execute the TCXO algorithm to update the capacitance array to the oscillator. This can only happen when a conversion is not already in progress. The user should check the status bit BSY before forcing the controller to start a new TCXO execution. A user-initiated temperature conversion does not affect the internal 64-second update cycle.

A user-initiated temperature conversion does not affect the BSY bit for approximately 2ms. The CONV bit remains at a 1 from the time it is written until the conversion is finished, at which time both CONV and BSY go to 0. The CONV bit should be used when monitoring the status of a user-initiated conversion.

Bits 4 and 3: Rate Select (RS2 and RS1). These bits control the frequency of the square-wave output when

the square wave has been enabled. The following table shows the square-wave frequencies that can be selected with the RS bits. These bits are both set to logic 1 (8.192kHz) when power is first applied.

SQUARE-WAVE OUTPUT FREQUENCY

RS2	RS1	SQUARE-WAVE OUTPUT FREQUENCY
0	0	1Hz
0	1	1.024kHz
1	0	4.096kHz
1	1	8.192kHz

Bit 2: Interrupt Control (INTCN). This bit controls the INT/SQW signal. When the INTCN bit is set to logic 0, a square wave is output on the INT/SQW pin. When the INTCN bit is set to logic 1, then a match between the time-keeping registers and either of the alarm registers activates the INT/SQW output (if the alarm is also enabled). The corresponding alarm flag is always set regardless of the state of the INTCN bit. The INTCN bit is set to logic 1 when power is first applied.

Bit 1: Alarm 2 Interrupt Enable (A2IE). When set to logic 1, this bit permits the alarm 2 flag (A2F) bit in the status register to assert INT/SQW (when INTCN = 1). When the A2IE bit is set to logic 0 or INTCN is set to logic 0, the A2F bit does not initiate an interrupt signal. The A2IE bit is disabled (logic 0) when power is first applied.

Bit 0: Alarm 1 Interrupt Enable (A1IE). When set to logic 1, this bit permits the alarm 1 flag (A1F) bit in the status register to assert INT/SQW (when INTCN = 1). When the A1IE bit is set to logic 0 or INTCN is set to logic 0, the A1F bit does not initiate the INT/SQW signal. The A1IE bit is disabled (logic 0) when power is first applied.

Status Register (0Fh)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	OSF	0	0	0	EN32kHz	BSY	A2F	A1F
POR:	1	0	0	0	1	X	X	X

Status Register (0Fh)

Bit 7: Oscillator Stop Flag (OSF). A logic 1 in this bit indicates that the oscillator either is stopped or was stopped for some period and may be used to judge the validity of the timekeeping data. This bit is set to logic 1 any time that the oscillator stops. The following are examples of conditions that can cause the OSF bit to be set:

- 1) The first time power is applied.
- 2) The voltages present on both VCC and VBAT are insufficient to support oscillation.
- 3) The EOSC bit is turned off in battery-backed mode.
- 4) External influences on the crystal (i.e., noise, leakage, etc.).

This bit remains at logic 1 until written to logic 0.

Bit 3: Enable 32kHz Output (EN32kHz). This bit controls the status of the 32kHz pin. When set to logic 1, the 32kHz pin is enabled and outputs a 32.768kHz square-wave signal. When set to logic 0, the 32kHz pin goes to a high-impedance state. The initial power-up state of this bit is logic 1, and a 32.768kHz square-wave signal appears at the 32kHz pin after a power source is applied to the DS3231 (if the oscillator is running).

Bit 2: Busy (BSY). This bit indicates the device is busy executing TCXO functions. It goes to logic 1 when the conversion signal to the temperature sensor is asserted and then is cleared when the device is in the 1-minute idle state.

Bit 1: Alarm 2 Flag (A2F). A logic 1 in the alarm 2 flag bit indicates that the time matched the alarm 2 registers. If the A2IE bit is logic 1 and the INTCN bit is set to logic 1, the INT/SQW pin is also asserted. A2F is cleared when written to logic 0. This bit can only be written to logic 0. Attempting to write to logic 1 leaves the value unchanged.

Bit 0: Alarm 1 Flag (A1F). A logic 1 in the alarm 1 flag bit indicates that the time matched the alarm 1 registers. If the

A1IE bit is logic 1 and the INTCN bit is set to logic 1, the INT/SQW pin is also asserted. A1F is cleared when written to logic 0. This bit can only be written to logic 0. Attempting to write to logic 1 leaves the value unchanged.

Aging Offset

The aging offset register takes a user-provided value to add to or subtract from the codes in the capacitance array registers. The code is encoded in two's complement, with bit 7 representing the sign bit. One LSB represents one small capacitor to be switched in or out of the capacitance array at the crystal pins. The aging offset register capacitance value is added or subtracted from the capacitance value that the device calculates for each temperature compensation. The offset register is added to the capacitance array during a normal temperature conversion, if the temperature changes from the previous conversion, or during a manual user conversion (setting the CONV bit). To see the effects of the aging register on the 32kHz output frequency immediately, a manual conversion should be started after each aging register change.

Positive aging values add capacitance to the array, slowing the oscillator frequency. Negative values remove capacitance from the array, increasing the oscillator frequency.

The change in ppm per LSB is different at different temperatures. The frequency vs. temperature curve is shifted by the values used in this register. At +25°C, one LSB typically provides about 0.1ppm change in frequency.

Use of the aging register is not needed to achieve the accuracy as defined in the EC tables, but could be used to help compensate for aging at a given temperature. See the *Typical Operating Characteristics* section for a graph showing the effect of the register on accuracy over temperature.

Aging Offset (10h)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	Sign	Data	Data	Data	Data	Data	Data	Data
POR:	0	0	0	0	0	0	0	0

Temperature Register (Upper Byte) (11h)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	Sign	Data	Data	Data	Data	Data	Data	Data
POR:	0	0	0	0	0	0	0	0

Temperature Register (Lower Byte) (12h)

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NAME:	Data	Data	0	0	0	0	0	0
POR:	0	0	0	0	0	0	0	0

Temperature Registers (11h–12h)

Temperature is represented as a 10-bit code with a resolution of 0.25°C and is accessible at location 11h and 12h. The temperature is encoded in two's complement format. The upper 8 bits, the integer portion, are at location 11h and the lower 2 bits, the fractional portion, are in the upper nibble at location 12h. For example, 00011001 01b = +25.25°C. Upon power reset, the registers are set to a default temperature of 0°C and the controller starts a temperature conversion. The

temperature is read on initial application of VCC or I²C access on VBAT and once every 64 seconds afterwards. The temperature registers are updated after each user-initiated conversion and on every 64-second conversion. The temperature registers are read-only.

I²C Serial Data Bus

The DS3231 supports a bidirectional I²C bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data is defined as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP

conditions. The DS3231 operates as a slave on the I²C bus. Connections to the bus are made through the SCL input and open-drain SDA I/O lines. Within the bus specifications, a standard mode (100kHz maximum clock rate) and a fast mode (400kHz maximum clock rate) are defined. The DS3231 works in both modes.

The following bus protocol has been defined (Figure 2):

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is high. Changes in the data

line while the clock line is high are interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain high.

START data transfer: A change in the state of the data line from high to low, while the clock line is high, defines a START condition.

STOP data transfer: A change in the state of the data line from low to high, while the clock line is high, defines a STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the high period of the clock signal. The data on the line must be changed during the low period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between the START and the STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse, which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable low during the high period of the acknowledge-related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generat-

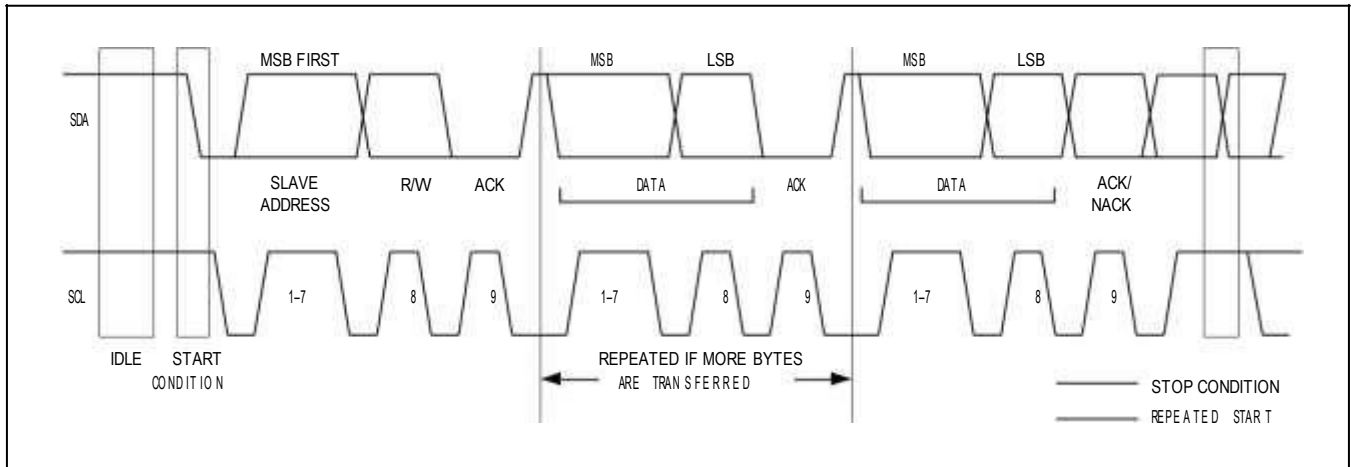


Figure 2. I²C Data Transfer Overview

ing an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line high to enable the master to generate the STOP condition.

Figures 3 and 4 detail how data transfer is accomplished on the I²C bus. Depending upon the state of the R/W bit, two types of data transfer are possible:

Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master

is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.

Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows a number of data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the

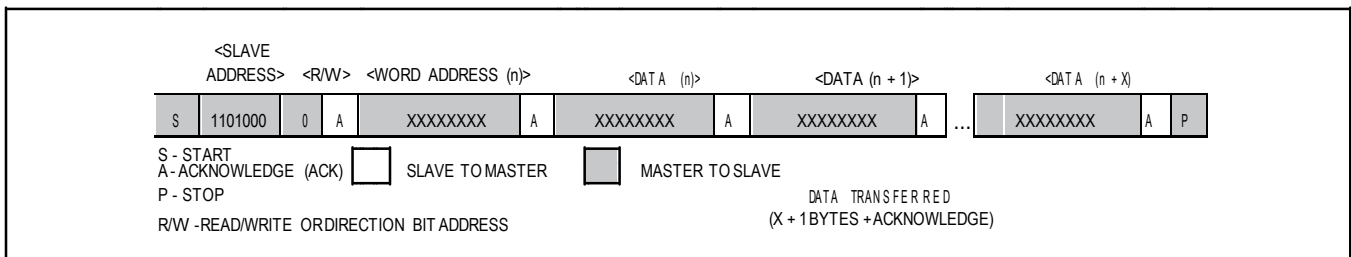


Figure 3. Data Write—Slave Receiver Mode

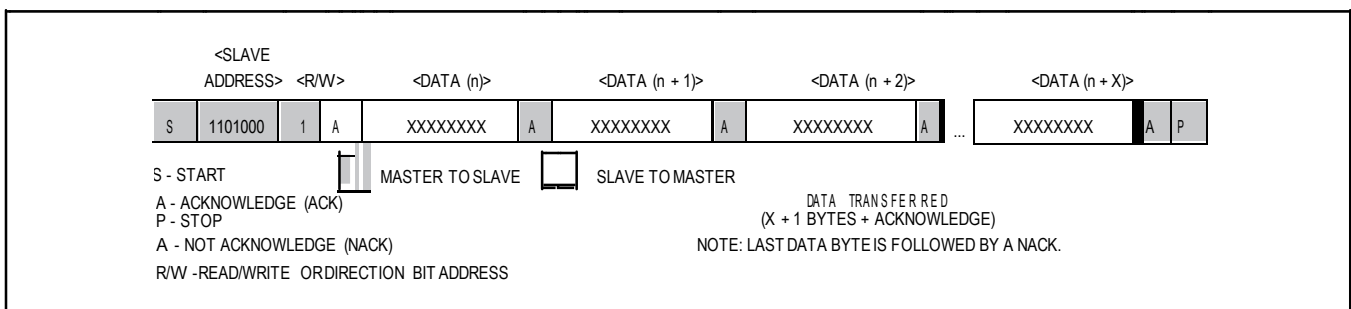


Figure 4. Data Read—Slave Transmitter Mode

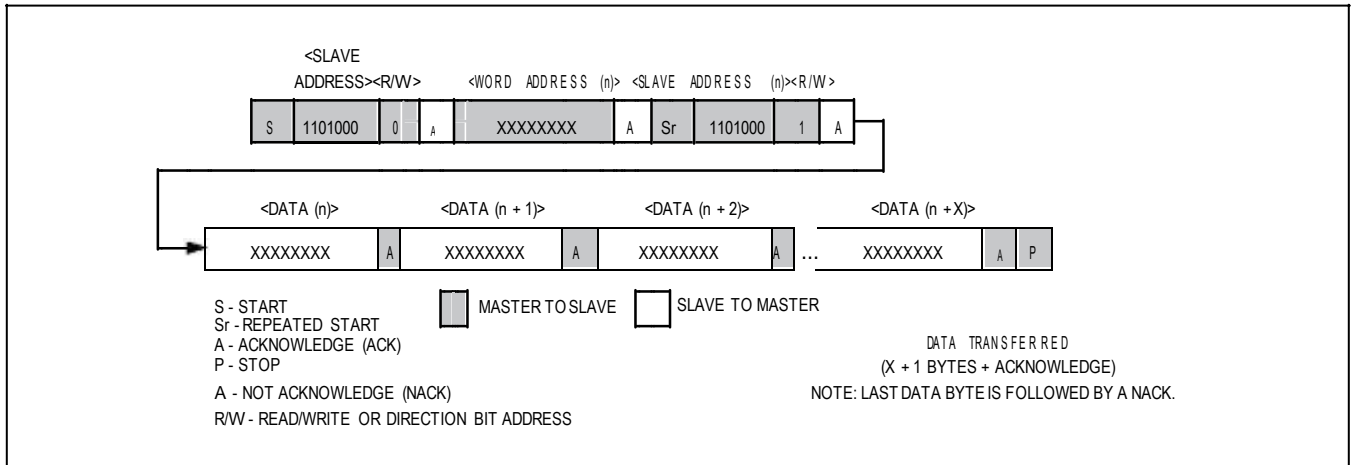


Figure 5. Data Write/Read (Write Pointer, Then Read)—Slave Receive and Transmit

last byte. At the end of the last received byte, a not acknowledge is returned.

The master device generates all the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS3231 can operate in the following two modes:

Slave receiver mode (DS3231 write mode): Serial data and clock are received through SDA and SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit. The slave address byte is the first byte received after the master generates the START condition. The slave address byte contains the 7-bit DS3231 address, which is 1101000, followed by the direction bit (R/W), which is 0 for a write. After receiving and decoding the slave address byte, the DS3231 outputs an acknowledge on SDA. After the DS3231 acknowledges the slave address + write bit, the master transmits a word address to the DS3231. This sets the register pointer on the DS3231, with the DS3231 acknowledging the

transfer. The master may then transmit zero or more bytes of data, with the DS3231 acknowledging each byte received. The register pointer increments after each data byte is transferred. The master generates a STOP condition to terminate the data write.

Slave transmitter mode (DS3231 read mode): The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit indicates that the transfer direction is reversed. Serial data is transmitted on SDA by the DS3231 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit. The slave address byte is the first byte received after the master generates a START condition. The slave address byte contains the 7-bit DS3231 address, which is 1101000, followed by the direction bit (R/W), which is 1 for a read. After receiving and decoding the slave address byte, the DS3231 outputs an acknowledge on SDA. The DS3231 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode, the first address that is read is the last one stored in the register pointer. The DS3231 must receive a not acknowledge to end a read.

Handling, PCB Layout, and Assembly

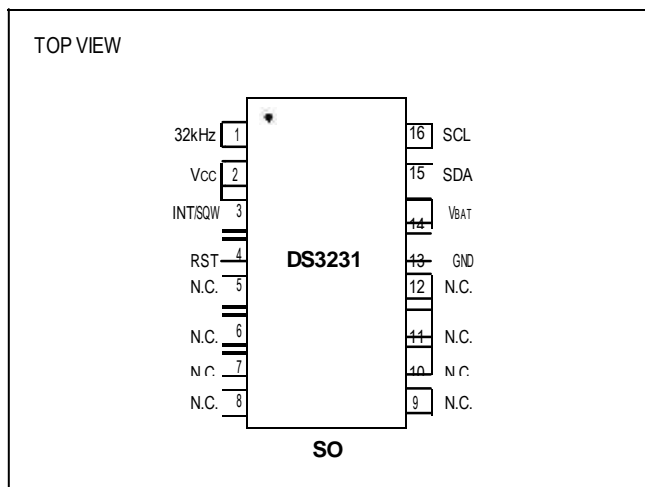
The DS3231 package contains a quartz tuning -fork crystal. Pick-and-place equipment can be used, but precautions should be taken to ensure that excessiveshocksareavoided.Ultrasoniccleaningshouldbe avoided to prevent damage to the crystal.

Avoid running signal traces under the package, unless a ground plane is placed between the package and the

signal line. All N.C. (no connect) pins must be connected to ground.

Moisture-sensitive packages are shipped from the factory dry packed. Handling instructions listed on the package label must be followed to prevent damage during reflow. Refer to the IPC/JEDEC J-STD-020 standard for moisture-sensitive device (MSD) classifications and reflow profiles. Exposure to reflow is limited to 2 times maximum.

Pin Configuration



Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
DS3231S#	0°C to +70°C	16 SO
DS3231SN#	-40°C to +85°C	16 SO

#Denotes an RoHS-compliant device that may include lead (Pb) that is exempt under RoHS requirements. The lead finish is JESD97 category e3, and is compatible with both lead-based and lead-free soldering processes. A “#” anywhere on the top mark denotes an RoHS-compliant device.

Package Information

For the latest package outline information and land patterns (footprints), go to www.maximintegrated.com/packages. Note that a “+”, “#”, or “-” in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

PACKAGE TYPE	PACKAGE CODE	OUTLINE NO.	LAND PATTERN NO.
16 SO	W16#H2	21-0042	90-0107

Chip Information

SUBSTRATE CONNECTED TO GROUND PROCESS: CMOS

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	1/05	Initial release.	—
1	2/05	Changed Digital Temp Sensor Output from $\pm 2^{\circ}\text{C}$ to $\pm 3^{\circ}\text{C}$.	1, 3
		Updated <i>Typical Operating Circuit</i> .	1
		Changed TA = -40°C to $+85^{\circ}\text{C}$ to TA = TMIN to TMAX.	2, 3, 4
		Updated <i>Block Diagram</i> .	8
2	6/05	Added “UL Recognized” to Features; added lead-free packages and removed S from top mark info in <i>Ordering Information</i> table; added ground connections to the N.C. pin in the <i>Typical Operating Circuit</i> .	1
		Added “noncondensing” to operating temperature range; changed VPF MIN from 2.35V to 2.45V.	2
		Added aging offset specification.	3
		Relabeled TOC4.	7
		Added arrow showing input on X1 in the <i>Block Diagram</i> .	8
		Updated pin descriptions for VCC and VBAT.	9
		Added the I ² C Interface section.	10
		<i>Figure 1</i> : Added sign bit to aging and temperature registers; added MSB and LSB.	11
		Corrected title for rate select bits frequency table.	13
		Added note that frequency stability over temperature spec is with aging offset register = 00h; changed bit 7 from Data to Sign (Crystal Aging Offset Register).	14
		Changed bit 7 from Data to Sign (Temperature Register); correct pin definitions in I ² C <i>Serial Data Bus</i> section.	15
Modified the <i>Handling</i> , <i>PC Board Layout</i> , and <i>Assembly</i> section to refer to J-STD-020 for reflow profiles for lead-free and leaded packages.	17		
3	11/05	Changed lead-free packages to RoHS-compliant packages.	1
4	10/06	Changed RST and UL bullets in <i>Features</i> .	1
		Changed EC condition “VCC > VBAT” to “VCC = Active Supply (see Table 1).”	2, 3
		Modified Note 12 to correct tREC operation.	6
		Added various conditions text to TOCs 1, 2, and 3.	7
		Added text to pin descriptions for 32kHz, VCC, and RST.	9
		Table 1: Changed column heading “Powered By” to “Active Supply”; changed “applied” to “exceeds VPF” in the <i>Power Control</i> section.	10
		Indicated BBSQW applies to both SQW and interrupts; simplified temp convert description (bit 5); added “output” to INT/SQW (bit 2).	13
Changed the <i>Crystal Aging</i> section to the <i>Aging Offset</i> section; changed “this bit indicates” to “this bit controls” for the enable 32kHz output bit.	14		
5	4/08	Added Warning note to EC table notes; updated Note 12.	6
		Updated the <i>Typical Operating Characteristics</i> graphs.	7
		In the <i>Power Control</i> section, added information about the POR state of the time and date registers; in the <i>Real-Time Clock</i> section, added to the description of the RST function.	10
		In <i>Figure 1</i> , corrected the months date range for 04h from 00–31 to 01–31.	11

Revision History (continued)

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
6	10/08	Updated the <i>Typical Operating Circuit</i> .	1
		Removed the VPU parameter from the <i>Recommended DC Operating Conditions</i> table and added verbiage about the pullup to the <i>Pin Description</i> table for INT/SQW, SDA, and SCL.	2, 9
		Added the Delta Time and Frequency vs. Temperature graph in the <i>Typical Operating Characteristics</i> section.	7
		Updated the <i>Block Diagram</i> .	8
		Added the <i>VBAT Operation</i> section, improved some sections of text for the <i>32kHz TCXO</i> and <i>Pushbutton Reset Function</i> sections.	10
		Added the register bit POR values to the register tables.	13, 14, 15
		Updated the <i>Aging Offset</i> and <i>Temperature Registers (11h–12h)</i> sections.	14, 15
		Updated the I ² C timing diagrams (Figures 3, 4, and 5).	16, 17
7	3/10	Removed the “S” from the top mark in the <i>Ordering Information</i> table and the <i>Pin Configuration</i> to match the packaging engineering marking specification.	1, 18
8	7/10	Updated the <i>Typical Operating Circuit</i> ; removed the “Top Mark” column from the <i>Ordering Information</i> ; in the <i>Absolute Maximum Ratings</i> section, added the theta-JA and theta-JC thermal resistances and Note 1, and changed the soldering temperature to +260°C (lead(Pb)-free) and +240°C (leaded); updated the functional description of the VBAT pin in the <i>Pin Description</i> ; changed the timekeeping registers 02h, 09h, and 0Ch to “20 Hour” in Bit 5 of Figure 1; updated the BBSQW bit description in the <i>Control Register (0Eh)</i> section; added the land pattern no. to the <i>Package Information</i> table.	1, 2, 3, 4, 6, 9, 11, 12, 13, 18
9	1/13	Updated <i>Absolute Maximum Ratings</i> , and last paragraph in <i>Power Control</i> section	2, 10
10	3/15	Revised <i>Benefits and Features</i> section.	1

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim Integrated's website at www.maximintegrated.com.

Maxim Integrated cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim Integrated product. No circuit patent licenses are implied. Maxim Integrated reserves the right to change the circuitry and specifications without notice at any time. The parametric values (min and max limits) shown in the Electrical Characteristics table are guaranteed. Other parametric values quoted in this data sheet are provided for guidance.

```
//Program Inkubator Telur Ayam Polandia
```

```
#include <DS3231.h>
```

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
#include <EEPROM.h>
```

```
#include <SimpleDHT.h>
```

```
##define Serial SerialUSB
```

```
DS3231 rtc(SDA, SCL); //inisialisasi DS3231 dengan hardware interface
```

```
Time t; //Inisialisasi struktur data waktu
```

```
LiquidCrystal_I2C lcd(0x3F,16,2); //inisialisasi alamat I2C pada LCD
```

```
byte telur[8] = {B00000,B00100,B01110,B11011,B11101,B11011,B01110,B00000};
```

```
byte derajat[8] = {B00110,B01001,B01001,B00110,B00000,B00000,B00000,B00000};
```

```
int pinDHT11 = 22;
```

```
int detik,menit,jam,h;
```

```
int hari;
```

```
int tgl,bln,thn;
```

```
int buzzer = 13; //Pin 13 dihubungkan ke buzzer
```

```
int tombol1,tombol2;
```

```
int lock = 0;
```

```
int exhaust = 48, pompa = 46;
```

```
int heater = 50, lampu = 52;
```

```
int pompa_in1, pompa_in2;
```

```
boolean start = false;
```

```
boolean setop = true;
```

```
SimpleDHT11 dht11;
```

```

void setup(){
  lampu_off();
  lcd.init();
  lcd.backlight();          //Menyalakan Backlight LCD
  lcd.createChar(0, telur);
  lcd.createChar(1, derajat);
  Serial.begin(115200);     //Pengaturan koneksi serial
  rtc.begin();
  pinMode(2,INPUT),(3,INPUT);
  pinMode(exhaust,OUTPUT);
  pinMode(pompa,OUTPUT);
  pinMode(heater,OUTPUT);
  pinMode(lampu,OUTPUT);
  pinMode(32,INPUT),(34,INPUT); //Masukan Sensor untuk Pompa
  pinMode(13,OUTPUT);       //Mengatur pin untuk buzzer sebagai output
  diam();                   //buzzer diam
  lcd.clear();              //Menghapus text pada LCD
  delay(100);
  hari = EEPROM.read(h);
}

void loop(){

  byte temperature = 0;
  byte humidity = 0;

  tombol1 = digitalRead(2);
  tombol2 = digitalRead(3);
  pompa_in1 = digitalRead(32);
  pompa_in2 = digitalRead(34);

```

```

t = rtc.getTime();          //mengambil data waktu dari RTC
detik = t.sec;
menit = t.min;
jam = t.hour;
tgl = t.date;
bln = t.mon;
thn = t.year-2000;

if (h > 21){
    h = 0;
}

//=====Pilih salah satu=====
//tanggal(); //menampilkan Tanggal
waktu(); //Menampilkan Waktu
Serial.print(jam);Serial.print(":");Serial.print(menit);Serial.print(":");Serial.println(detik);

if(tombol2 == HIGH){
    hari = 1;
}

if(tombol1 == HIGH){
    if(start==false){
        start = true;
    }

    else if(start==true){
        start = false;
    }
}
}

```

```

if(start==false){
    lcd.setCursor(7,1);
    lcd.print(" ");
    delay(500);
    if (jam == 0 && menit == 0 && detik == 0){
        delay(1000);
        lcd.setCursor(10,1);
        lcd.print("00:00:00");
    }
}

else if(start==true){
    lcd.setCursor(7,1);
    lcd.write(0);
    delay(500);
    if (jam == 0 && menit == 0 && detik == 0){ //pengaturan hari dihitung dengan waktu 00:00:00
        delay(700);
        lcd.setCursor(10,1);
        lcd.print("00:00:00");
        hari = hari+1;
    }
}

if (hari<10){
    lcd.setCursor(6,1);
    lcd.print(" ");
}

lcd.setCursor(0,1);
lcd.print("Hari:");
lcd.print(hari);

```



```
//=====Program Humidity=====
```

```
if (dht11.read(pinDHT11, &temperature, &humidity, NULL)) {  
    return;  
}
```

```
lcd.setCursor(0,0);
```

```
lcd.print("T:");
```

```
lcd.print((int)temperature); lcd.write(1);lcd.print("C, ");
```

```
lcd.setCursor(8,0);
```

```
lcd.print("H:");
```

```
lcd.print((int)humidity); lcd.println(" % ");
```

```
Serial.print("T:");Serial.print(temperature);Serial.print("C");Serial.print("  
");Serial.print("H:");Serial.print(humidity);Serial.println("% ");
```

```
Serial.println(" ");
```

```
EEPROM.write(h,hari);
```

```
if (hari == 21){
```

```
    bunyi();
```

```
    // hari = 0;
```

```
}
```

```
else {
```

```
    diam();
```

```
}
```

```
//=====Pembacaan Suhu Untuk Lampu=====
```

```
if(temperature > 39){
```

```
    lampu_off();
```

```
}
```

```
else {
```

```
    lampu_on();
```

```
}
```

```
//=====Pembacaan Ketinggian Air Untuk Pompa=====
```

```
if (pompa_in1 == HIGH && pompa_in2 == HIGH){
```

```
    pompa_on();
```

```
}
```

```
if (pompa_in1 == LOW && pompa_in2 == LOW){
```

```
    pompa_off();
```

```
}
```

```
//=====Pembacaan Kelembaban Untuk Heater & hexos=====
```

```
if (humidity < 55){
```

```
    heater_on();
```

```
}
```

```
else{
```

```
    exhaust_on();
```

```
}
```

```
if (humidity > 55){
```

```
    heater_off();
```

```
}
```

```
else {
```

```
    exhaust_off();
```

```
}
```

```
}
```

```
//END PROGRAM
```

```
//=====Kondisi Buzzer=====
```

```
void bunyi(){  
    digitalWrite(buzzer,HIGH);  
}
```

```
void diam(){  
    digitalWrite(buzzer,LOW);  
}
```

```
//=====Tampilan Tanggal-Bulan-Tahun=====
```

```
void tanggal(){  
    if(tgl < 10){  
        lcd.setCursor(8,1);  
        lcd.print("0");  
        lcd.print(tgl);  
    }  
  
    if(tgl >= 10){  
        lcd.setCursor(8,1);  
        lcd.print(tgl);  
    }  
  
    lcd.setCursor(10,1);  
    lcd.print("-");  
  
    if(bln<10){
```

```
lcd.setCursor(11,1);  
lcd.print("0");  
lcd.print(bln);  
}
```

```
if(bln>=10){  
    lcd.setCursor(11,1);  
    lcd.print(bln);  
}
```

```
lcd.print("-");  
lcd.print(thn);  
}
```

```
//=====
```

```
//=====Tampilan Jam=====
```

```
void waktu(){  
    if(jam < 10){  
        lcd.setCursor(8,1);  
        lcd.print("0");  
        lcd.print(jam);  
    }
```

```
    if(jam >= 10){  
        lcd.setCursor(8,1);  
        lcd.print(jam);  
    }
```

```
lcd.setCursor(10,1);  
lcd.print(":");
```

```
if(menit<10){  
    lcd.setCursor(11,1);  
    lcd.print("0");  
    lcd.print(menit);  
}
```

```
if(menit>=10){  
    lcd.setCursor(11,1);  
    lcd.print(menit);  
}
```

```
lcd.print(":");
```

```
if(detik<10){  
    lcd.setCursor(14,1);  
    lcd.print("0");  
    lcd.print(detik);  
}
```

```
if(detik>=10){  
    lcd.setCursor(14,1);  
    lcd.print(detik);  
}  
}
```

```
void lampu_on(){  
    digitalWrite(lampu,LOW);  
    Serial.println("Lampu Nyala");  
}
```

```
void lampu_off(){
```

```
digitalWrite(lampu,HIGH);
Serial.println("Lampu Mati");
}

void exhaust_on(){
digitalWrite(exhaust,LOW);
Serial.println("exhaust Nyala");
}

void exhaust_off(){
digitalWrite(exhaust,HIGH);
Serial.println("Exhaust mati");
}

void heater_on(){
digitalWrite(heater,LOW);
Serial.println("Heater Nyala");
}

void heater_off(){
digitalWrite(heater,HIGH);
Serial.println("Heater mati");
}

void pompa_on(){
digitalWrite(pompa,LOW);
Serial.println("Pompa Nyala");
}

void pompa_off(){
digitalWrite(pompa,HIGH);
```

```
Serial.println("Pompa mati");
```

```
}
```