

**SISTEM KONTROL SUHU DAN KELEMBAPAN PADA
BUDIDAYA JAMUR TIRAM BERBASIS IoT**

PROYEK AKHIR



Disusun Oleh :

Mutiara Ramadhana NIRM:0031722

Suwanda Putra NIRM :0031728

POLITEKNIK MENUFAKTUR NEGERI

BANGKA BELITUNG

2020

LEMBAR PENGESAHAN

**SISTEM KONTROL SUHU DAN KELEMBAPAN PADA
BUDIDAYA JAMUR TIRAM BERBASIS IoT**

Oleh:

Mutiara Ramadhana NIRM :0031722

Suwanda Putra NIRM :0031728

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1




Nofriyani, M.Tr.T.

Pembimbing 2



Indra Dwisaputra, MT.

Penguji 1



Yudhi, MT.

Penguji 2



Charlotha, M.Tr.T.

Penguji 3



Zanu Saputra, M.Tr.T.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama mahasiswa 1 : Mutiara Ramadhana NIRM:0031722

Nama mahasiswa 2 : Suwanda Putra NIRM:0031728

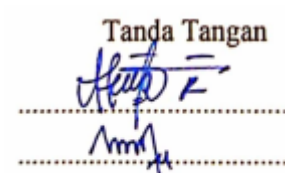
Dengan judul : Sistem Kontrol Suhu dan Kelembapan Pada Budidaya
Jamur Tiram Berbasis IOT

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 26 Agustus 2020

Nama mahasiswa

1. Mutiara Ramadhana
2. Suwanda Putra

Tanda Tangan


ABSTRAK

Jamur tiram adalah jenis jamur yang banyak dibudidayakan, suhu dan kelembapan pada kumbung jamur sangat berpengaruh terhadap hasil panen para petani jamur tiram. Suhu yang ideal adalah 22-26 °C sedangkan untuk kelembapan 80-90% RH. Proyek akhir ini yang dilakukan adalah membuat sistem kontrol suhu dan kelembapan pada budidaya jamur tiram dengan menggunakan kipas exhaust, sprayer, sensor DHT11, yang dikontrol oleh ESP32 dan terhubung dengan NodeMCU ESP32. Perancangan kontrol dengan menggunakan kipas exhaust digunakan untuk mempercepat pengeluaran udara panas pada kumbung jamur serta mempercepat penyebaran partikel air yang dihasilkan oleh sprayer. Peletakan sensor DHT11 di tiap sudut kumbung jamur ditujukan agar pembacaan dari sensor lebih akurat, dengan mengambil nilai rata-rata dari pembacaan sensor. Tujuan pada proyek akhir ini yaitu mampu membuat sistem kontrol suhu dan kelembapan pada budidaya jamur tiram agar tetap dalam keadaan stabil. Hasil pengujian yang didapatkan bahwa penggunaan sistem kontrol suhu dan kelembapan pada jamur tiram dapat meningkatkan efisiensi dalam mengontrol suhu dan kelembapan. Persentase error pembacaan suhu yang dibaca oleh sensor DHT11 sebesar $\pm 0,015\%$ sedangkan untuk kelembapan persentase error sebesar $\pm 0,004\%$. Penurunan dari suhu dan kelembapan awal menjadi suhu dan kelembapan yang diinginkan membutuhkan waktu selama ± 20 menit dengan rentan waktu bertahan sampai ± 1 jam.

Kata kunci: NodeMCU ESP32, DHT11, Blynk, Kipas exhaust, Sprayer.

ABSTRACT

Oyster mushroom is a type of mushroom that is widely cultivated, the temperature and humidity in the mushroom kumbung greatly affect the yields of the oyster mushroom farmers. The ideal temperature is 22-26 °C while for humidity 80-90% RH. This final project is to make a temperature and humidity control system in oyster mushroom cultivation using an exhaust fan, sprayer, DHT11 sensor, which is controlled by ESP32 and connected to the ESP32 NodeMCU. Control design using exhaust fans is used to accelerate the discharge of hot air in the mushroom kumbung and accelerate the spread of water particles generated by the sprayer. Placement of the DHT11 sensor in each corner of the mushroom kumbung is intended so that the reading from the sensor is more accurate, by taking the average value of the sensor readings. The purpose of this final project is to be able to make a temperature and humidity control system in oyster mushroom cultivated buoys to keep it in a stable state. The test results obtained that the use of temperature and humidity control systems on oyster mushrooms can increase efficiency in controlling temperature and humidity. The percentage error in temperature readings by the DHT11 sensor is $\pm 0.015\%$, while for humidity the percentage error is $\pm 0.004\%$. The decrease from the initial temperature and humidity to the desired temperature and humidity takes ± 20 minutes with a vulnerable holding time of up to ± 1 hour.

Keywords: NodeMCU ESP32, DHT11, Blynk, Fan Exhouse, Sprayer.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT. Karena berkat Rahmat dan Hidayah-Nya, sehingga dapat menyelesaikan proyek akhir ini dengan baik. Proyek akhir ini dibuat guna sebagai persyaratan dan kewajiban mahasiswa untuk menyelesaikan kurikulum program pendidikan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung.

Selama proses pelaksanaan proyek akhir, pengambilan data dan penulisan laporan ini kami telah banyak mendapatkan bantuan dari berbagai pihak maka dari itu pada kesempatan ini penulis mengucapkan banyak terima kasih kepada :

1. Allah SWT atas segala berkat dan rahmat-Nya hingga proyek akhir ini selesai.
2. Keluarga tercinta, yang selalu sabar membimbing, mendoakan, dan memberikan motivasi dalam menyelesaikan proyek akhir ini. Terima kasih untuk setiap peluh keringat serta kasih sayang yang tidak ternilai.
3. Bapak I Made Andik Setiawan, M.Eng, Ph.D, selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
4. Ibu Nofriyani, M.Tr.T. Selaku pembimbing I dan Bapak Indra Dwisaputra, MT. Selaku pembimbing II yang telah meluangkan waktu, tenaga, pikiran dalam memberikan pengarahan serta bimbingan dalam pembuatan alat dan penulisan laporan proyek akhir.
5. Seluruh staf pengajar di Politeknik Manufaktur Negeri Bangka Belitung.
6. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak membantu selama menyelesaikan proyek akhir.
7. Pihak-pihak lain yang telah memberikan bantuan secara langsung maupun tidak langsung dalam pembuatan proyek akhir ini yang tidak dapat disebutkan satu per satu.

Proyek akhir ini diharapkan dapat bermanfaat bagi penulis khususnya dan bagi pembaca pada umumnya. Penulis menyadari dalam penyusunan proyek akhir ini

masih banyak terdapat kekurangan, baik dari segi materi, maupun dalam penyampaian materi. Oleh sebab itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak guna memperbaiki kedepannya.

Akhirnya, semoga laporan ini bermanfaat dan berguna sebagaimana yang diharapkan. Aamiin.

Sungailiat, 19 Agustus 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan Proyek Akhir.....	2
1.4 Batasan Masalah.....	2
BAB II DASAR TEORI	3
2.1 IoT (<i>Internet of Things</i>).....	3
2.2 NodeMCU ESP32	3
2.3 DHT11	4
2.4 LCD 16x2.....	5
2.5 Relay Modul.....	6
2.6 <i>Mist Sprayer</i>	6
2.7 Blynk.....	7
BAB III METODE PELAKSANAAN	8

3.1 Flowchart Perancangan dan Pembuatan Alat.....	8
3.2 Survei, Pengolahan dan Analisa data	11
3.2.1 Survei Data.....	11
3.2.1.1 Pengumpulan Data Secara Langsung.....	11
3.2.1.2 Pengumpulan Data Secara Tidak Langsung	11
3.2.2 Pengolahan Data.....	11
3.2.3 Analisa Data	11
3.3 Perancangan <i>Hardware</i> Elektrik Kontrol Budidaya Jamur Tiram Berbasis IoT .	11
3.4 Pembuatan <i>Hardware</i> Elektrik Kontrol Budidaya Jamur Tiram Berbasis IoT	12
3.5 Pengujian <i>Hardware</i> Elektrik Kontrol Budidaya Jamur Tiram Berbasis IoT.....	12
3.6 Perancangan <i>Software</i> Kontrol Budidaya Jamur Tiram Berbasis IoT	12
3.7 Pembuatan <i>Software</i> Kontrol Budidaya Jamur Tiram Berbasis IoT	12
3.8 Perakitan <i>Hardware</i> Elektrik dan <i>Software</i>	13
3.9 Pembuatan Laporan Proyek Akhir	13
BAB IV PEMBAHASAN.....	14
4.1 Deskripsi Alat	14
4.2 Implementasi Pemograman.....	15
4.3 Pengujian Sensor DHT11.....	16
4.3.1 Uji Coba Empat Sensor DHT11.....	18
4.3.2 Uji Coba Akurasi Pembacaan Sensor dengan Pembacaan <i>Hygrometer</i>	20
4.4 Pengujian Rangkaian Relay	21
4.5 Pengujian Rangkaian LCD.....	23
4.6 Pengujian Kipas <i>Exhaust</i>	25
4.7 Rangkaian Pompa Listrik 12VDC	26
4.8 Analisa Data	27

4.8.1 Pengambilan Data dengan <i>Prototype</i>	27
4.8.2 Pengambilan Data pada Kumbung Jamur	28
BAB V KESIMPULAN DAN SARAN.....	30
5.1 Kesimpulan	30
5.2 Saran.....	30
DAFTAR PUSTAKA	31

DAFTAR GAMBAR

Gambar 2.1 NodeMCU ESP32	4
Gambar 2.2 DHT11 Sensor Suhu dan Kelembapan	5
Gambar 2.3 LCD 16x2.....	5
Gambar 2.4 Modul Relay.....	6
Gambar 4.1 Diagram Blok Sistem Kerja Alat	14
Gambar 4.2 Penggunaan <i>Board</i> DOIT ESP32 DEVKIT V1 pada Arduino IDE... 15	15
Gambar 4.3 Tampilan Pembacaan Serial Empat Sensor DHT11	18
Gambar 4.4 Uji Coba Relay	21
Gambar 4.5 <i>Setting</i> Blynk.....	23
Gambar 4.6 Tampilan Blynk.....	23
Gambar 4.7 Rangkaian Uji Coba LCD	24
Gambar 4.8 Tampilan LCD dan Blynk	24
Gambar 4.9 Kipas <i>Exhaust</i> dan Relay.....	25
Gambar 4.10 <i>Schematic</i> Rangkaian Kipas <i>Exhaust</i>	26
Gambar 4.11 Motor Listrik dan Adaptor	27
Gambar 4.12 <i>Prototype</i> Kumbung Jamur dengan Sistem Kontrol Otomatis dan Manual.....	28
Gambar 4.13 (a) <i>Box</i> Kontrol, (b) Kipas <i>Exhaust</i> , (c) Sensor DHT11, (d) <i>Hygrometer</i> , (e) <i>Sprayer</i>	28

DAFTAR TABEL

Tabel 2.1 Spesifikasi NodeMCU ESP32	4
Tabel 4.1 Data Uji Coba Pengaruh Relay Terhadap Suhu dan Kelembapan.....	16
Tabel 4.2 Uji Coba Pembacaan Empat Sensor dan Nilai Rata-Rata Pembacaan Sensor.....	19
Tabel 4.3 Data Perbandingan Hasil Pembacaan Suhu dan Kelembapan dengan Menggunakan Alat Kontrol dan <i>Hygrometer</i>	20
Tabel 4.4 Data Uji Coba Kondisi Relay.....	22
Tabel 4.5 Uji Coba Kontrol Kipas dengan Menggunakan Relay	26
Tabel 4.6 Uji Coba Kontrol Kipas dengan Menggunakan Saklar	26
Tabel 4.7 Uji Coba Kondisi <i>Sprayer</i> dengan Menggunakan Relay.....	27
Tabel 4.8 Uji Coba Kondisi <i>Sprayer</i> dengan Menggunakan Saklar	27
Tabel 4.9 Perbandingan Suhu dan Kelembapan Menggunakan Sistem Kontrol dan <i>Hygrometer</i>	28
Tabel 4.10 Hasil Uji Coba Sistem Kontrol	29

DAFTAR LAMPIRAN

Lampiran 1: Daftar Riwayat Hidup

Lampiran 2: Program Keseluruhan

Lampiran 3: *Data Sheet*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Jamur tiram atau *king oyster mushroom* adalah bahan pangan yang memiliki banyak kandungan gizi yang sangat tinggi, diantaranya yaitu karbohidrat, kalium, kalsium, fosfor, zat besi, vitamin B, B1, dan C (Nunung, 2010). Jamur tiram memiliki banyak peminat sehingga memiliki peluang dunia pasar yang sangat menjanjikan. Tanaman ini dapat tumbuh dengan optimal 22-26°C dengan kelembapan 80%-90% RH (Hadi, 2015) serta PH optimum antara 6-7 (Riyanto, F, 2010). Perkembangan budidaya jamur tiram di Bangka Belitung mengalami peningkatan yang cukup pesat dikarenakan permintaan pasar yang semakin tinggi dan juga banyaknya pelaku usaha makanan dengan menu jamur yang semakin diminati oleh masyarakat.

Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) Provinsi Kepulauan Bangka Belitung menyatakan saat musim kemarau suhu mencapai 24-31°C dan saat musim penghujan 23-30°C. Tingginya permintaan pasar menyebabkan para petani jamur tiram kewalahan dalam memenuhi permintaan pasar. Petani jamur tiram mengeluhkan hasil panen jamur yang belum optimal, dikarenakan petani kesulitan dalam mengontrol suhu dan kelembapan pada kumbung jamur. Selama ini para petani jamur tiram di Bangka Belitung masih menggunakan cara konvensional. Cara yang dilakukan para petani jamur di Bangka Belitung untuk mengontrol suhu dan kelembapan di kumbung jamur mereka adalah dengan melakukan penyiraman secara berkala, sehingga petani harus sering bolak-balik ke kumbung untuk melakukan pengecekan suhu dan kelembapan pada kumbung jamur. Upaya penyiraman yang dilakukan oleh petani jamur merupakan hal yang kurang efisien karena dilakukan dengan cara manual dan juga kurang efektif karena suhu dan kelembapan pada kumbung jamur bersifat fluktuatif karena kondisi setiap saat dapat berubah (Wahyono, 2016), sehingga diperlukan suatu alat yang dapat mengontrol suhu dan kelembapan kumbung jamur agar meningkatkan

kualitas hasil panen jamur tiram yang dihasilkan dan mempermudah petani jamur tiram untuk membudidayakan jamur jenis ini.

1.2 Perumusan Masalah

1. Bagaimana membuat sistem kontrol suhu dan kelembapan pada budidaya jamur tiram berbasis *Internet of Things* (IoT).
2. Bagaimana memantau dan mengontrol suhu dan kelembapan pada kumbung jamur setiap waktu walaupun petani berada dikejauhan (± 20 meter) dari tempat budidaya jamur.
3. Bagaimana meningkatkan kualitas hasil panen jamur tiram.
4. Bagaimana mengurangi persentase gagal panen yang disebabkan oleh ketidaksesuaian suhu dan kelembapan pada kumbung jamur tiram.

1.3 Tujuan Proyek Akhir

1. Membuat Sistem Kontrol Suhu dan Kelembapan Pada Budidaya Jamur Tiram Berbasis *Internet of Things* (IoT).
2. Untuk memantau dan mengontrol suhu dan kelembapan pada kumbung jamur setiap waktu walaupun petani berada dikejauhan (± 20 meter) dari tempat budidaya jamur.

1.4 Batasan Masalah

1. Pengaturan suhu pada kumbung jamur dalam rentan 22-26°C.
2. Pembacaan kelembapan maksimal yang dapat dibaca oleh sensor adalah 90%RH.
3. Jarak jangkauan *monitoring* dari *box* kontrol dapat dipantau dari jarak (± 20 meter).
4. Tidak mengontrol kecepatan kipas, hanya mengontrol putaran kipas.
5. Mode kipas tidak dapat aktif bersamaan dalam satu waktu.

BAB II

DASAR TEORI

Temperatur merupakan salah satu informasi yang sangat penting dalam menentukan kondisi cuaca pada sebuah daerah. Temperatur juga merupakan salah satu kunci penting dalam dunia pertanian atau perkebunan, industri makanan, industri elektronika dan lain-lain. Pengukuran temperatur secara manual dapat dilakukan dengan termometer standar, namun tidak ada alat yang dapat mengatur temperatur tetap dalam kondisi stabil, apalagi temperatur harus dipantau terus menerus, oleh karena itu dibutuhkan alat pengukur temperatur yang disertai dengan sistem kontrolnya, sehingga temperatur dalam sebuah ruangan dapat diukur secara otomatis dan dapat dijaga secara otomatis pula, serta hasil data temperatur dapat ditampilkan pada *display*.

2.1 IoT (*Internet of Things*)

Internet of Things atau yang sering kita sebut IoT adalah sebuah konsep yang memanfaatkan koneksi internet yang tersambung secara terus menerus (L.G, Alexandromeo, 2017). Penggunaan IoT dalam proyek ini digunakan untuk memonitoring kondisi kumbung adalah dengan menggunakan NodeMCU ESP32 yang aktif dengan sambungan Wi-Fi, sehingga hasil yang ditampilkan akan terus *update*.

2.2 NodeMCU ESP32

NodeMCU ESP32 adalah salah satu keluarga mikrokontroler yang dikenalkan dan dikembangkan oleh *Espressif System*. NodeMCU ESP32 ini merupakan penerus dari mikrokontroler ESP8266. Mikrokontroler satu ini *compatible* dengan Arduino IDE. Pada mikrokontroler ini sudah tersedia modul Wi-Fi dan ditambah dengan BLE (*Bluetooth Low Energy*) dalam *chip* sehingga sangat mendukung dan dapat menjadi pilihan bagus untuk membuat

sistem aplikasi *Internet of Things* (Prastyo, 2019). Gambar fisik sensor DHT11 dapat dilihat pada Gambar 2.1.



Gambar 2.1 NodeMCU ESP32

Keterangan tentang spesifikasi ESP32 dapat dilihat pada tabel 2.1 (Rui Santos & Sara Santos)

Tabel 2.1 Spesifikasi NodeMCU ESP32

Number of cores	2 (Dual core)
Wi-Fi	2.4 GHz up to 150 Mbit/s
Bluetooth	BLE (Bluetooth Low Energy) and legacy Bluetooth
Architecture	32 bits
Clock frequency	Up to 240 MHz
RAM	512 KB www.earthbondhon.com
Pins	30
Peripherals	Capacitive touch, ADCs (analog-to-digital converter), DACs (digital-to-analog converter), I ² C (Inter-Integrated Circuit), UART (universal asynchronous receiver/transmitter), CAN 2.0 (Controller Area Network), SPI (Serial Peripheral Interface), I ² S (Integrated Inter-IC Sound), RMII (Reduced Media-Independent Interface), PWM (pulse width modulation), and more.

2.3 DHT11

Sensor Suhu dan Kelembapan DHT11 ini memiliki sensor suhu dan kelembapan kompleks dengan *output* sinyal digital yang dikalibrasi. Dengan menggunakan akuisisi sinyal digital eksklusif. Teknik dan teknologi penginderaan suhu dan kelembapan, memastikan keandalan tinggi dan stabilitas jangka panjang yang sangat baik. Sensor ini mencakup pengukuran kelembapan tipe komponen

resistif dan komponen pengukuran suhu NTC, dan terhubung ke mikrokontroler 8-bit berperforma tinggi, menawarkan kualitas yang sangat baik, respons yang cepat, anti-gangguan kemampuan dan efektivitas biaya. Setiap elemen DHT11 dikalibrasi secara ketat di laboratorium yang sangat akurat kalibrasi kelembapan. Koefisien kalibrasi disimpan sebagai program dalam memori OTP, yang digunakan oleh proses pendeteksian sinyal internal sensor (UK, 2010). Gambar fisik sensor DHT11 dapat dilihat pada gambar 2.2.



Gambar 2.2 DHT11 Sensor Suhu dan Kelembapan

2.4 LCD 16x2

Liquid Crystal Display atau yang biasa disebut LCD adalah suatu jenis media penampil yang menggunakan kristal cair sebagai media penampil utamanya. LCD sudah sering dijumpai dan digunakan diberbagai perangkat, misalnya alat-alat elektronik seperti kalkulator, televisi, ataupun layar komputer. Gambar fisik LCD dapat dilihat pada gambar 2.3.



Gambar 2.3 LCD 16x2

Fungsi LCD sangat penting karena berfungsi untuk menampilkan status kerja pada suatu alat. *Inter Integrated Circuit* atau yang sering disebut I2C merupakan standar komunikasi serial dua arah yang menggunakan dua saluran dan didisain khusus untuk menerima ataupun mengirim data. Sistem yang terdapat pada I2C terdiri dari saluran SCL (*serial clock*) dan SDA (*serial data*) yang mengirim

informasi berupa data antara I2C dengan pengontrolnya. Fungsi LCD sangat penting karena berfungsi untuk menampilkan status kerja pada suatu alat (Sejati, 2019).

2.5 Relay Modul

Relay merupakan komponen elektronika berupa saklar atau *switch electric* yang dioperasikan secara listrik dan terdiri dari 2 bagian utama yaitu Elektromagnet (*coil*) dan mekanikal (seperangkat kontak Saklar/*Switch*) (Lusidah, M., Taufik, M., & A.W.Purwandi, 2018). Relay menggunakan prinsip elektromagnetik untuk menggerakkan saklar sehingga dengan arus listrik yang kecil dapat menghantarkan listrik bertegangan tinggi. Gambar fisik relay modul dapat dilihat pada gambar 2.4.



Gambar 2.4 Modul Relay

2.6 Mist Sprayer

Sprayer adalah alat/mesin yang berfungsi untuk memecah suatu cairan, larutan atau suspensi menjadi butiran cairan (*droplet*) atau *spray*. *Sprayer* memiliki banyak kegunaan yaitu, menyemprotkan *insektisida*, *fungisida*, *herbisida*, pupuk cairan, dan air. *Mist Sprayer* menggunakan mesin pompa air bertekanan tinggi sehingga dapat menghasilkan butiran cairan halus yang bisa menambah kelembapan udara. *Mist Sprayer* memiliki beberapa komponen utama yaitu :

- Pompa air 12VDC berkekuatan 80psi
- *Mist nozzle* 0.3mm
- Selang

Prinsip kerja dari *Mist Sprayer* berasal dari pompa air yang mengalirkan air ke dalam selang yang diujungnya sudah terpasang *nozzle* sehingga air yang

berbentuk cairan bisa berubah menjadi butiran cairan halus untuk meningkatkan kelembapan udara (Agusta, 2019).

2.7 Blynk

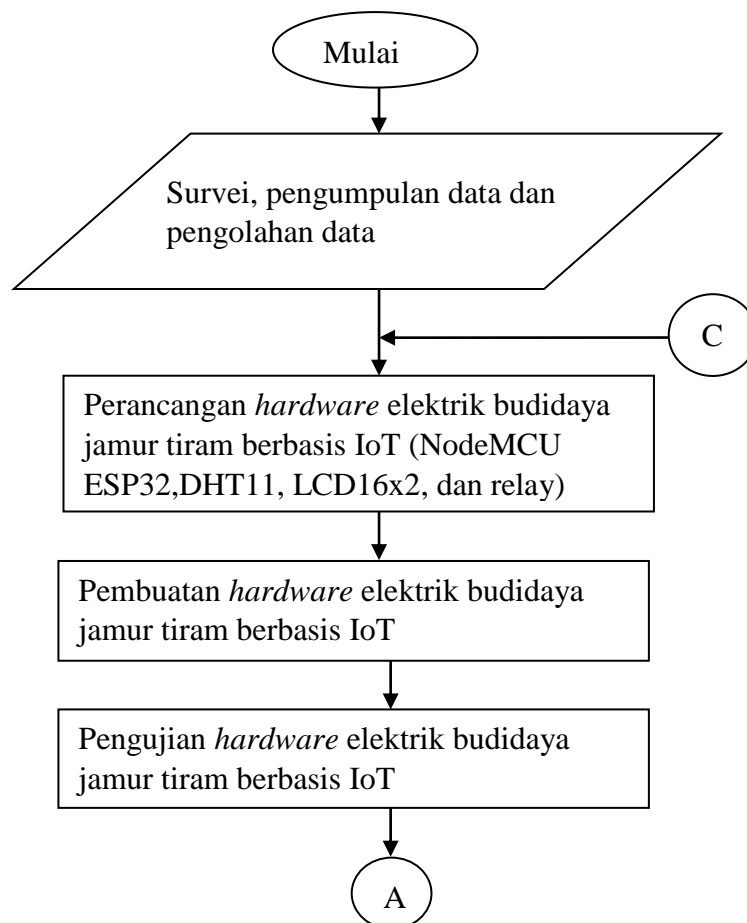
Blynk merupakan sebuah *platform* perangkat lunak yang dapat membantu membangun *interface* untuk mengendalikan atau menjalankan *hardware* dari *iOS* dan perangkat Android. Blynk dirancang untuk dapat mengontrol dan membaca data dari *hardware* yang digunakan. Aplikasi yang digunakan dalam proyek ini adalah Blynk untuk membaca data dari NodeMCU ESP32 .

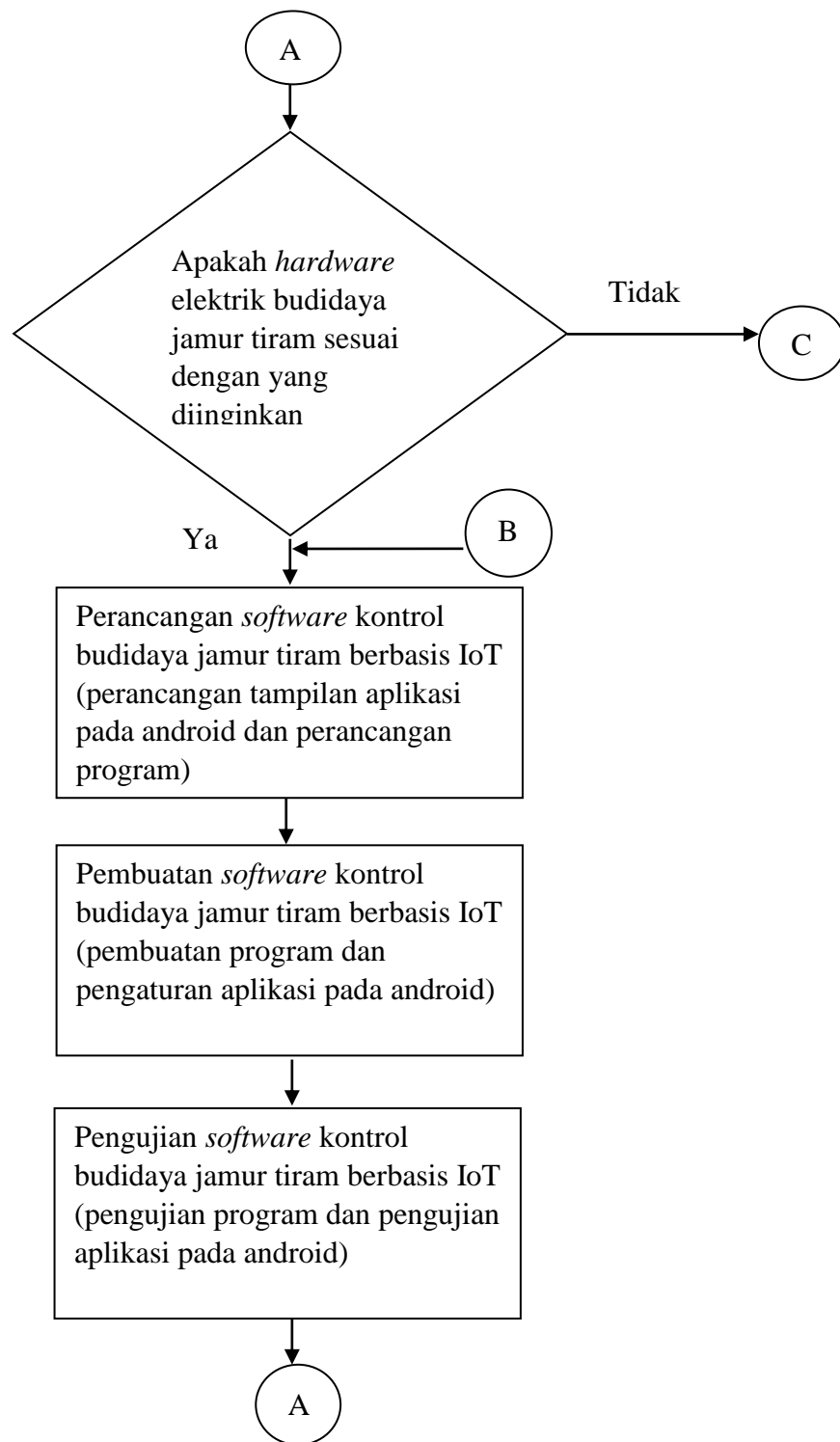
BAB III METODE PELAKSANAAN

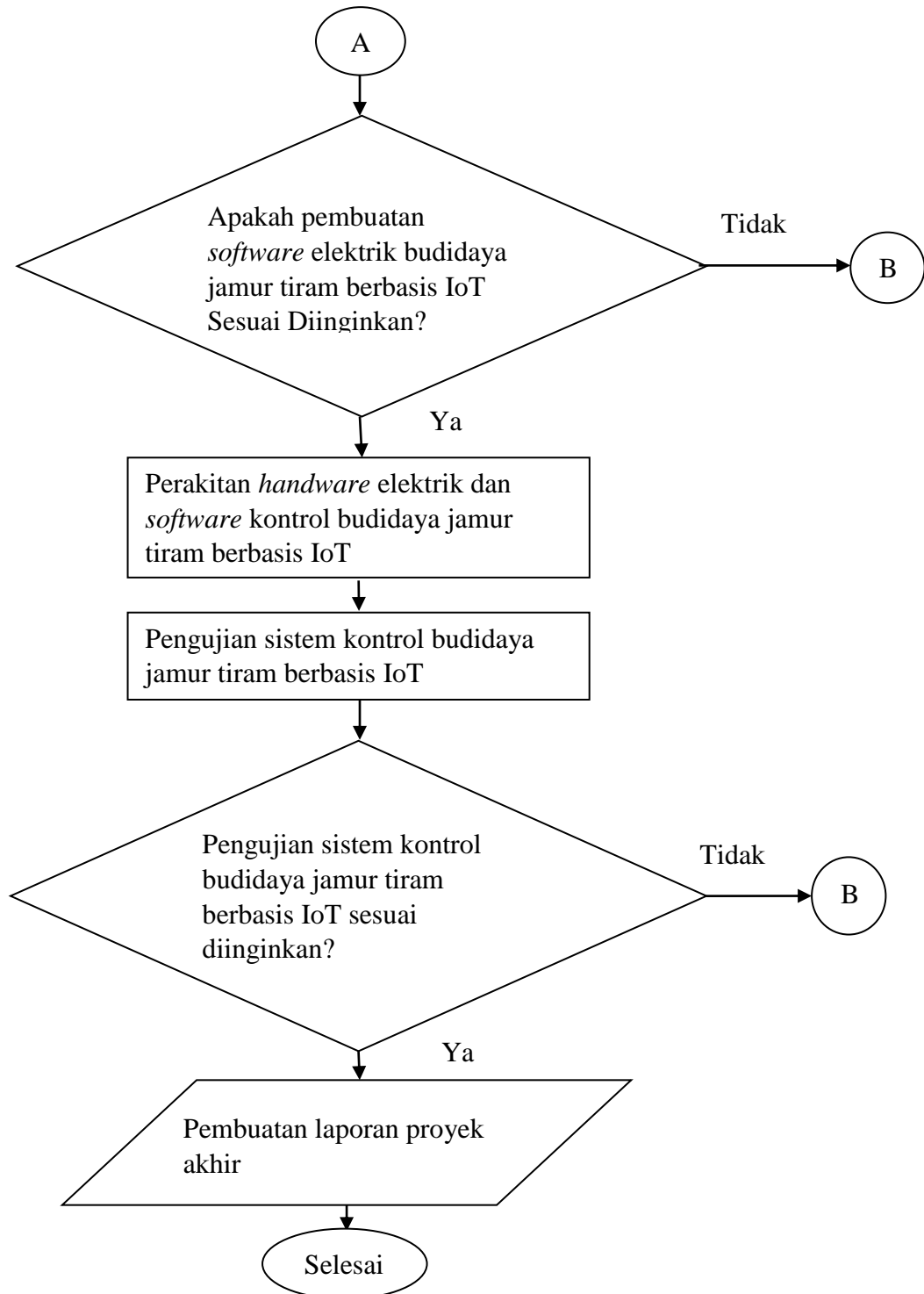
Dalam pembuatan proyek akhir terdapat beberapa tahapan dalam proses pengerjaan yang bertujuan untuk mempermudah dalam pencapaian target penyelesaian. Metode pelaksanaan yang diterapkan seperti berikut :

3.1 Flowchart Perancangan dan Pembuatan Alat

Flowchart perancangan dan pembuatan proyek akhir ini dapat dilihat pada gambar 3.1.







Gambar 3.1 Flowchart Perancangan dan Pembuatan Alat

3.2 Survei, Pengolahan dan Analisa data

3.2.1 Survei Data

Survei data merupakan pengumpulan data-data yang berkaitan dengan proyek akhir. Pengumpulan data ini dibagi menjadi dua yaitu pengumpulan data secara langsung dan tidak langsung.

3.2.1.1 Pengumpulan Data Secara Langsung

Pengumpulan data secara langsung diperoleh dari pembimbing I dan pembimbing II. Pada pengumpulan data ini juga mengumpulkan data seperti survei lapangan untuk pengambilan data dari budidaya jamur tiram berbasis IoT.

3.2.1.2 Pengumpulan Data Secara Tidak Langsung

Pengumpulan data secara tidak langsung didapatkan dari makalah TA sebelumnya yang berjudul “*Monitoring Pemakaian Daya Pada Pembangkit Listrik Tenaga Surya Berbasis Smartphone*” dan jurnal-jurnal yang masih berkaitan dengan proyek akhir ini.

3.2.2 Pengolahan Data

Tahap pengolahan data, setelah data-data terkumpul maka data akan diolah sesuai dengan referensi-referensi yang mengacu pada proyek akhir.

3.2.3 Analisa Data

Analisa data merupakan suatu tahap yang dilakukan untuk mengolah data hasil dari penelitian menjadi sebuah informasi baru yang dapat digunakan dalam membuat kesimpulan.

3.3 Perancangan *Hardware* Elektrik Kontrol Budidaya Jamur Tiram Berbasis IoT

Perancangan *hardware* elektrik kontrol budidaya jamur tiram berbasis IoT dilakukan dengan menentukan komponen elektrik yang akan digunakan yaitu: Sensor DHT11, NodeMCU ESP32, relay, LCD 16x2.

3.4 Pembuatan *Hardware* Elektrik Kontrol Budidaya Jamur Tiram Berbasis IoT

Proses pembuatan *hardware* elektrik kontrol budidaya jamur tiram berbasis IoT dilakukan di kos dengan membeli komponen elektrik yang sudah jadi dan siap digunakan.

3.5 Pengujian *Hardware* Elektrik Kontrol Budidaya Jamur Tiram Berbasis IoT

Pengujian komponen elektrik ini dilakukan untuk mengetahui apakah komponen yang digunakan bekerja sesuai dengan fungsi yang diinginkan. Uji coba komponen elektrik berupa:

- Uji coba, Sensor DHT11, dan Relay
- Uji coba NodeMCU ESP32
- Uji coba pengoneksian ke aplikasi Blynk

3.6 Perancangan *Software* Kontrol Budidaya Jamur Tiram Berbasis IoT

Perancangan *software* dilakukan dengan membuat rancangan berupa aplikasi yang ada pada *smartphone* dan perancangan program dengan menggunakan aplikasi Blynk, untuk memonitoring perubahan suhu dan kelembapan pada tempat budidaya jamur tiram. Perancangan *software* meliputi rancangan tampilan suhu, kelembapan, grafik dan *data logger* pada aplikasi Blynk.

3.7 Pembuatan *Software* Kontrol Budidaya Jamur Tiram Berbasis IoT

Pembuatan *software* meliputi:

- Pembuatan program aplikasi komunikasi antara sistem kontrol ke *smartphone*.
- Pembuatan aplikasi pada *smartphone* dengan menggunakan aplikasi Blynk.
- Pembuatan program keseluruhan pada NodeMCU ESP32

3.8 Perakitan *Hardware* Elektrik dan *Software*

Perakitan *hardware* elektrik dan *software* kontrol budidaya jamur tiram berbasis IoT dilakukan dengan cara merakit keseluruhan pada setiap bagian dari *hardawre* elektrik, *software* dan penginstalasian digabung menjadi satu kesatuan.

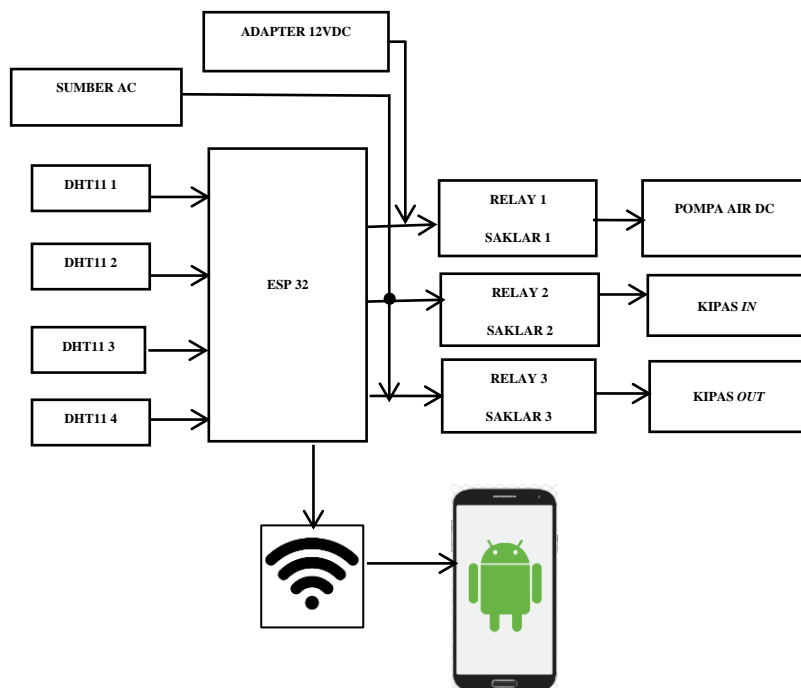
3.9 Pembuatan Laporan Proyek Akhir

Tahap pembuatan laporan merupakan tahap terakhir dalam pembuatan proyek akhir yang bertujuan untuk merangkum keseluruhan yang berhubungan dengan proyek akhir, seperti latar belakang, tujuan, rumusan masalah, batasan masalah, landasan teori, metode pelaksanaan, pembahasan serta kesimpulan dan saran.

BAB IV PEMBAHASAN

4.1 Deskripsi Alat

Sistem kontrol suhu dan kelembapan pada jamur tiram berbasis IoT dapat dilihat pada gambar 4.1. Diagram blok diatas adalah sistem kontrol secara otomatis dan manual dengan menggunakan media *smartphone* dengan aplikasi Blynk koneksi Wi-Fi NodeMCU ESP32. Sensor yang digunakan adalah empat sensor DHT11 yang diletakkan disetiap sudut kumbung. Tampilan pada *box* kontrol dan aplikasi Blynk adalah nilai rata – rata dari pembacaan ke-empat sensor tersebut. Hasil dari pembacaan sensor diproses menggunakan NodeMCU ESP32 untuk mengirim data yang ditampilkan pada *smartphone*.



Gambar 4.1 Diagram Blok Sistem Kerja Alat

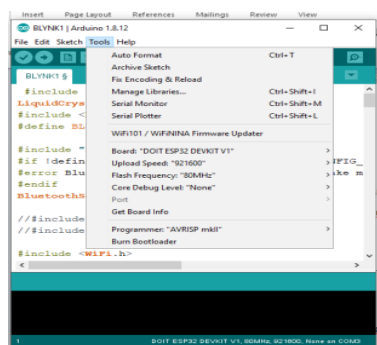
Hasil pembacaan sensor menjadi acuan batas suhu dan kelembapan yang akan dikontrol dengan menggunakan relay sebagai kontak saklar. Relay yang digunakan ada tiga relay yang memiliki tugas yang berbeda, relay pertama

digunakan untuk mengontrol *sprayer*, relay kedua digunakan untuk mengontrol kipas *in* dan relay ketiga digunakan untuk mengontrol kipas *out*. Kipas yang digunakan adalah kipas *exhaust* teggangan 220VAC dengan dua *output* yaitu kipas *in* dan kipas *out*, sedangkan *sprayer* menggunakan motor 12VDC.

Pada *box* kontrol menggunakan *pilot lamp* berwarna jingga yang digunakan sebagai indikator sistem aktif. Penggunaan alat kontrol memiliki tiga mode yaitu mode otomatis, manual dan *stop*. penggunaan mode dapat dipilih dengan mengarahkan tuas *toggle* kearah mode yang diinginkan. Pemilihan mode manual dapat menggunakan saklar untuk mengoperasikan *sprayer* dan kipas, dengan catatan saklar kipas *in* dan *out* tidak bisa digunakan secara bersamaan.

4.2 Implementasi Pemograman

Arduino IDE adalah *software* yang digunakan untuk memprogram NodeMCU ESP32 dengan menggunakan *board* DOIT ESP32 DEVKIT V1. *Software* Arduino IDE digunakan untuk mengkompilasi dan *upload* program atau *sketch* ke dalam NodeMCU ESP32. *Library* yang digunakan adalah *Liquid Crytal* I2C, DHT11, Blynk, Wi-Fi, NodeMCU ESP32. *Platform* Blynk digunakan untuk menjalankan aplikasi IoT. Penggunaan *board* DOIT ESP32 DEVKIT V1 pada *software* arduino IDE dapat dilihat pada gambar 4.2.



Gambar 4.2 Penggunaan *Board* DOIT ESP32 DEVKIT V1 pada Arduino IDE

Aplikasi Blynk sebagai media *monitoring* dengan menampilkan pembacaan suhu dan kelembapan dengan dua mode. Mode pertama menggunakan

gauge sebagai tampilan suhu dan kelembapan yang dibaca oleh sensor, sedangkan mode kedua *superchart* untuk tampilan grafik yang menyimpan data suhu dan kelembapan dari pembacaan *gauge*.

4.3 Pengujian Sensor DHT11

Pada sensor DHT11 dapat mengukur suhu dan kelembapan pada kumbung jamur tiram. Pembacaan dari sensor DHT11 akan ditampilkan pada layar LCD *box* kontrol dan Blynk. Cara kerja sensor DHT11 dalam mengontrol suhu pada program ini yaitu jika suhu pada kumbung terlalu rendah maka akan mengaktifkan kipas *out*, jika suhu terlalu tinggi akan mengaktifkan kipas *in*. Dalam mengontrol kelembapan pada program ini jika sensor DHT11 membaca kelembapan dibawah 80%RH maka akan mengaktifkan kipas *in* dan *sprayer* akan menyala. Saat pembacaan suhu diatas 90%RH maka akan mengaktifkan kipas *out* dan *sprayer* akan mati. Saat suhu dan kelembapan dalam keadaan ideal maka kipas *in*, kipas *out* dan *sprayer* akan dalam keadaan *off*.

Tabel 4.1 Data Uji Coba Pengaruh Relay Terhadap Suhu dan Kelembapan

NO	KONDISI			WAKTU : 5 MENIT						KET			
	R1	R2	R3	SUHU		RH		PERUBAHAN		SUHU		RH	
				AWAL	AKHIR	AWAL	AKHIR	SUHU	RH	NAIK	TURUN	NAIK	TURUN
1	ON	ON	OFF	27.60	27.90	78.90	86.70	+0.30	+7.80	V		V	
2	OFF	ON	OFF	27.90	28.30	86.60	86.00	+0.40	-0.60	V			V
3	ON	OFF	ON	27.10	26.40	85.80	86.90	-0.70	+1.10		V	V	
4	OFF	OFF	ON	26.60	26.30	86.90	86.80	-0.30	-0.10		V		V
5	ON	OFF	OFF	26.40	26.60	86.90	90.00	+0.20	+3.10	V		V	
6	OFF	OFF	OFF	26.60	26.70	86.90	86.80	+0.10	-0.10	V			V

Suhu yang diinginkan yaitu 22°C – 26°C, apabila suhu dibawah 22°C maka, *sprayer* dan kipas akan aktif dengan kondisi *sprayer* akan menyala dan kipas *in* dalam kondisi aktif. Ketika suhu diatas 26°C maka, *sprayer* dan kipas aktif dengan kondisi *sprayer* dan kipas *in* akan mati, sedangkan kipas *out* akan aktif untuk membuang udara panas yang ada didalam kumbung jamur.

Kelembapan yang diinginkan yaitu 80%-90%RH jika kelembapan di bawah 80%RH maka, *sprayer* dan kipas aktif dengan kondisi *sprayer* dan kipas *in* akan

aktif untuk menaikkan kelembapan dalam kumbung sedangkan kipas *out* dalam kondisi mati. Apabila kelembapan diatas 90% maka, *sprayer* dan kipas aktif dengan kondisi *sprayer* dan kipas *in* akan mati dan kipas *out* akan menyala untuk membuang uap air yang ada di dalam kumbung jamur. Setelah pengambilan data pada tabel 4.1. Kita dapat membuat beberapa kondisi untuk mengontrol suhu dan kelembapan pada kumbung jamur agar sesuai dengan yang kita inginkan, Pengaturan suhu dan kelembapan dapat diatur dalam program. Berikut program untuk mengatur batas suhu dan kelembapan yang kita inginkan:

```
//KONTROL KELEMBAPAN KUMBUNG 80-90 RH

if (kelembapanHasil<80.00){//jika kelembapan dibawah 80
digitalWrite ( relay1, LOW);           //RELAY 1 MENYALA
digitalWrite ( relay2, HIGH );         //RELAY 2 MATI
digitalWrite ( relay3, LOW);           // RELAY 3 MENYALA
}

if (kelembapanHasil>=80.00 && kelembapanHasil<=90.00 )
{
//jika kelembapan diantara
80.00-90.00
digitalWrite ( relay1, HIGH);          // RELAY 1 MATI
digitalWrite ( relay2, HIGH );         // RELAY 2 MATI
digitalWrite ( relay3, HIGH);          // RELAY 3 MATI
}

if (kelembapanHasil>90.00){
// jika kelembapan diatas 90.00
digitalWrite ( relay1, HIGH );         // RELAY 1 MATI
digitalWrite ( relay2, LOW );          // RELAY 2 HIDUP
digitalWrite ( relay3, HIGH );         //RELAY 3 MATI
}

//KONTROL SUHU KUMBUNG 22 - 26°C
if ( suhuHasil<22.00 ){ //jika suhu dibawah 22.00
digitalWrite ( relay1, HIGH );         //RELAY 1 MATI
digitalWrite ( relay2, HIGH );         //RELAY 2 MATI
```

```

digitalWrite ( relay3, LOW );           //RELAY 3 HIDUP
}

if (suhuHasil>=22.00 && suhuHasil<=26.00)
{
    //jika suhu diantara 16.00-22.00
    digitalWrite ( relay1, HIGH );     //RELAY 1 MATI
    digitalWrite ( relay2, HIGH );     //RELAY 2 MATI
    digitalWrite ( relay3, HIGH );     //RELAY 3 MATI
}

if (suhuHasil>26.00){                 //jika suhu diatas 26.00
    digitalWrite ( relay1, HIGH );     //RELAY 1 MATI
    digitalWrite ( relay2, LOW );      //RELAY 2 HIDUP
    digitalWrite ( relay3, HIGH );     //RELAY 3 MATI
}

```

4.3.1 Uji Coba Empat Sensor DHT11

Pengujian empat sensor digunakan untuk mendapatkan hasil yang lebih akurat. Hasil dari pembacaan setiap sensor akan dijumlahkan, sehingga hasil yang ditampilkan pada *display box* kontrol dan Blynk adalah hasil rata-rata sensor yang terpasang pada sudut kumbang. Gambar hasil pengambilan data empat sensor dapat dilihat pada gambar gambar 4.3.



Gambar 4.3 Tampilan Pembacaan Serial Empat Sensor DHT11

Hasil pembacaan suhu dan kelembapan yang ditampilkan setiap sensor berbeda, seperti yang dijelaskan pada tabel 4.2.

Tabel 4.2 Uji Coba Pembacaan Empat Sensor dan Nilai Rata-Rata Pembacaan Sensor

No	Pembacaan Sensor								Rata-Rata	
	Suhu 1	Suhu 2	Suhu 3	Suhu 4	RH 1	RH 2	RH 3	RH 4	Suhu	RH
1	32.10	31.70	30.00	30.80	70.00	71.00	74.00	73.00	31.15	72.00
2	32.00	31.80	30.00	30.90	69.00	71.00	74.00	72.00	31.18	71.50
3	32.00	31.80	29.00	31.00	70.00	70.00	74.00	70.00	30.95	71.00
4	32.00	31.80	30.00	31.00	69.00	69.00	72.00	70.00	31.20	70.00
5	31.90	31.80	30.00	30.80	69.00	69.00	72.00	71.00	31.12	70.25

Program untuk pembacaan suhu dan kelembapan dari sensor DHT11 adalah sebagai berikut:

```
float kelembapan1 = dht1.readHumidity();
float kelembapan2 = dht2.readHumidity();
float kelembapan3 = dht3.readHumidity();
float kelembapan4 = dht4.readHumidity();
float kelembapanJumlah =
dht1.readHumidity()+dht2.readHumidity()+
dht3.readHumidity()+dht4.readHumidity();
float kelembapanHasil = kelembapanJumlah/4;

float suhu1 = dht1.readTemperature();
float suhu2 = dht2.readTemperature();
float suhu3 = dht3.readTemperature();
float suhu4 = dht4.readTemperature();
float suhuJumlah =
dht1.readTemperature()+dht2.readTemperature()+
dht3.readTemperature()+dht4.readTemperature();
float suhuHasil = suhuJumlah/4;
delay(50);
```


4.3.2 Uji Coba Akurasi Pembacaan Sensor dengan Pembacaan *Hygrometer*

Pengujian akurasi dilakukan untuk mengetahui akurasi dari pembacaan sensor dan pembacaan *hygrometer*. Berikut adalah tabel hasil pengujian, dapat dilihat pada tabel 4.3.

Tabel 4.3 Data Perbandingan Hasil Pembacaan Suhu dan Kelembapan dengan Menggunakan Alat Kontrol dan *Hygrometer*

No	Waktu	Pembacaan				Selisih		Error	
		Sensor		Higrometer		Suhu	RH	Suhu	RH
		Suhu	RH	Suhu	RH				
1	16.18	31.12	71.25	30.6	71	0.52	0.25	0.016%	0.003%
2	16.19	31.12	71.75	30.6	71	0.52	0.25	0.016%	0.003%
3	16.25	31.13	71.30	30.7	71	0.43	0.30	0.013%	0.004%
4	16.30	31.13	71.60	30.6	71	0.53	0.60	0.017%	0.008%
5	16.40	31.15	70.50	30.5	70	0.65	0.50	0.020%	0.007%
6	16.58	31.10	70.30	30.8	70	0.30	0.30	0.009%	0.004%

Pengujian *error* dilakukan dengan mencari data perbandingan dari pembacaan dua alat yang berbeda yaitu sistem kontrol dan *hygrometer*. Pengujian *error* dari pembacaan dari alat ukur tersebut dijabarkan sebagai berikut:

$$\text{Persentase error} = \frac{\sum \text{selisih data sensor dan hygrometer}}{\text{nilai data sensor}} \times 100\%$$

$$\text{Persentase error suhu 1} = \frac{31.12-30.6}{31.12} \times 100\% = 0.016\%$$

$$\text{Persentase error suhu 2} = \frac{31.12-30.6}{31.12} \times 100\% = 0.016\%$$

$$\text{Persentase error suhu 3} = \frac{31.13-30.7}{31.13} \times 100\% = 0.013\%$$

$$\text{Persentase error suhu 4} = \frac{31.13-30.6}{31.13} \times 100\% = 0.017\%$$

$$\text{Persentase error suhu 5} = \frac{31.15-30.8}{31.15} \times 100\% = 0.020\%$$

$$\text{Persentase error suhu 6} = \frac{31.10-30.8}{31.10} \times 100\% = 0.009\%$$

$$\text{Persentase error suhu rata-rata} = \frac{0.016+0.016+0.013+0.017+0.020+0.009}{6} = 0.015$$

$$\text{Persentase error kelembapan 1} = \frac{71.25-71}{71.25} \times 100\% = 0.003\%$$

$$\text{Persentase error kelembapan 2} = \frac{71.75-71}{71.75} \times 100\% = 0.003\%$$

$$\text{Persentase error kelembapan 3} = \frac{71.30-71}{71.30} \times 100\% = 0.004\%$$

$$\text{Persentase error kelembapan 4} = \frac{71.60-71}{71.60} \times 100\% = 0.008\%$$

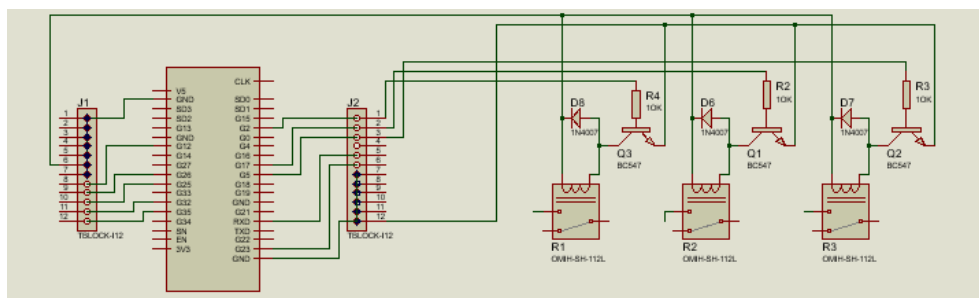
$$\text{Persentase error kelembapan 5} = \frac{70.50-71}{70.50} \times 100\% = 0.007\%$$

$$\text{Persentase error kelembapan 6} = \frac{70.30-70}{70.30} \times 100\% = 0.004\%$$

$$\text{Persentase error kelembapan rata-rata} = \frac{0.003+0.003+0.004+0.008+0.007+0.004}{6} = 0.004\%$$

4.4 Pengujian Rangkaian Relay

Pada alat ini, relay yang digunakan adalah relay *optocoupler* dengan *input*-an 5VDC. Relay digunakan sebagai kontak saklar, yang dapat mengontrol kipas dan *sprayer*. Ketika data pembacaan suhu $<22^{\circ}\text{C}$ maka kontak relay akan aktif untuk menghidupkan kipas *out*, namun jika suhu $>26^{\circ}\text{C}$ maka kontak relay akan mengaktifkan kipas *in*. kontak relay akan mati jika kondisi sudah terpenuhi pada pengujian ini kipas *in* dan kipas *out* tidak boleh aktif secara bersamaan karena motor listrik yang digerakan adalah motor yang sama. Ketika data pembacaan kelembapan adalah $\leq 80\text{RH}$ maka relay akan mengaktifkan *sprayer* dan relay akan mati jika kelembapan $\geq 90\text{RH}$. Pengujian pada alat ini Relay cukup bekerja dengan baik. Gambar dapat dilihat ada gambar 4.4.



Gambar 4.4 Uji Coba Relay

Untuk mengaktifkan relay dapat menggunakan tombol virtual dengan bantuan aplikasi Blynk, program sebagai berikut:

```

void setup()
{
pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
pinMode(relay3, OUTPUT);
pinMode(relay1, HIGH);
pinMode(relay2, HIGH);
pinMode(relay3, HIGH);
Serial.begin(9600);
delay(10);
Serial.println("WiFi connected");
Blynk.begin(auth, ssid, pass);}
void loop()

{
Blynk.run();
}

```

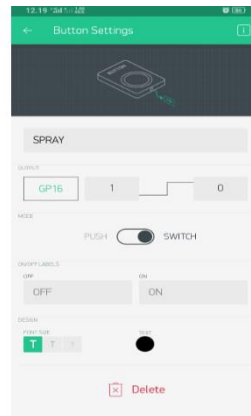
Untuk mengetahui kondisi relay, dapat dilihat pada tabel 4.4

Tabel 4.4 Data Uji Coba Kondisi Relay

No	Relay 1	Relay 2	Relay 3	Kondisi
1	OFF	OFF	OFF	Baik
2	ON	OFF	OFF	Baik
3	OFF	ON	OFF	Baik
4	OFF	OFF	ON	Baik
5	ON	ON	OFF	Baik

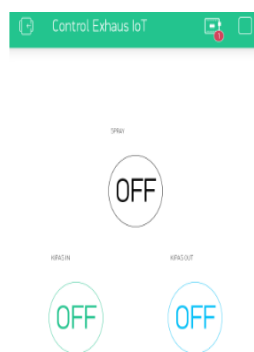
Relay 1 digunakan untuk *sprayer*, relay 2 digunakan untuk kipas *in* dan relay 3 digunakan untuk kipas *out*. Lampu led pada relay digunakan sebagai indikator, saat tombol virtual pada Blynk *on* maka relay akan aktif. Pada saat uji coba relay

1 menggunakan pin 19, relay 2 menggunakan pin 18 dan relay 3 menggunakan pin 15. *Setting-an* tombol dapat dilihat pada Gambar 4.5.



Gambar 4.5 *Setting* Blynk

Untuk tampilan pada *setting-an* tampilan dapat dilihat pada gambar 4.6.

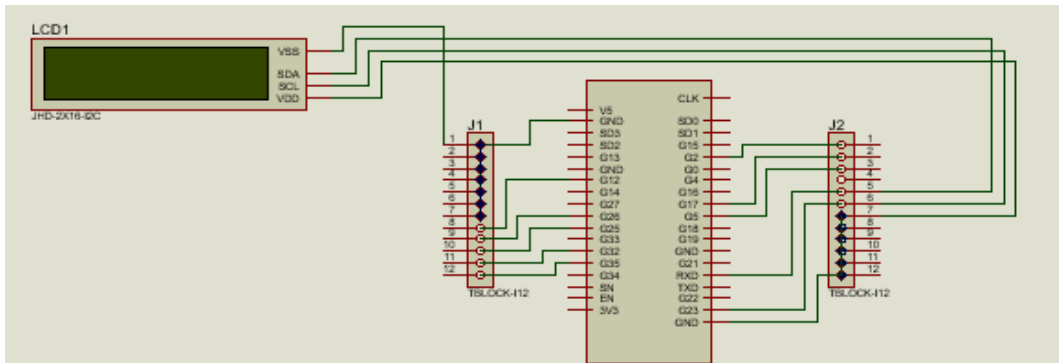


Gambar 4.6 Tampilan Blynk

4.5 Pengujian Rangkaian LCD

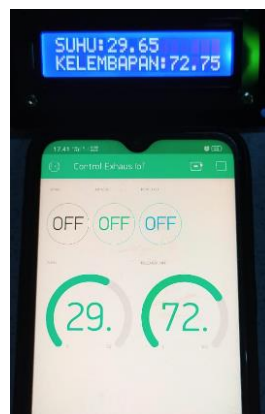
LCD yang digunakan adalah LCD16x2 yang berfungsi sebagai tampilan hasil pembacaan dari sensor DHT11 yang telah dikelola oleh NodeMCU ESP32 yang diletakkan pada *box* kontrol. LCD dihubungkan dengan NodeMCU ESP32, PORT pada LCD perlu dihubungkan dengan pin yang sesuai pada NodeMCU ESP32. PORT pada LCD terdiri dari VCC, SDA, SCL, dan GND. VCC dihubungkan dengan PIN5V pada NodeMCU ESP32, SDA dihubungkan dengan

pin 21, SCL dihubungkan dengan pin 22, dan GND dihubungkan dengan pin GND. Gambar *Schematic* rangkaian dapat dilihat pada gambar 4.7.



Gambar 4.7 Rangkaian Uji Coba LCD

Hasil dari tampilan LCD dan Blynk dapat dilihat pada gambar 4.8.



Gambar 4.8 Tampilan LCD dan Blynk

Berdasarkan keterangan di atas maka kita sudah dapat membuat program untuk menampilkan karakter pada *display* LCD. Adapun program yang diisikan ke mikrokontroller untuk menampilkan karakter pada *display* LCD adalah sebagai berikut:

```
//TAMPILAN LCD BOX
lcd.clear();
lcd.setCursor(0,0);
lcd.print("SUHU:");
lcd.setCursor(5,0);
```

```

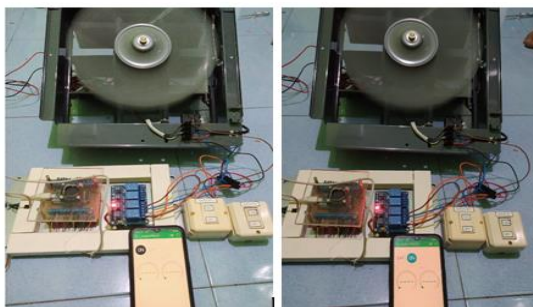
lcd.print(suhuHasil);

lcd.setCursor(0,1);
lcd.print("KELEMBAPAN:");
lcd.setCursor(11,1);
lcd.print(kelembapanHasil);
//TAMPILAN PADA BLYNK
Blynk.virtualWrite(V0,suhuHasil);
Blynk.virtualWrite(V1,kelembapanHasil);
delay(5000);

```

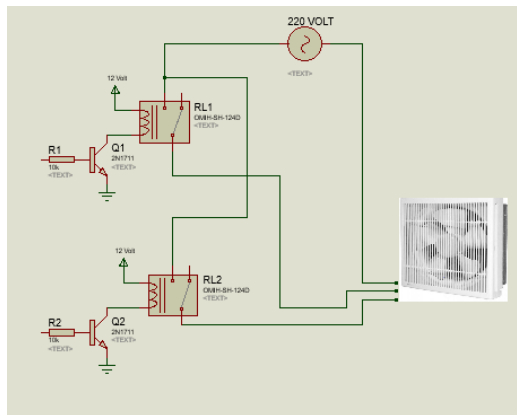
4.6 Pengujian Kipas *Exhaust*

Kipas yang digunakan untuk sistem kontrol ini adalah kipas *exhouse*, kipas ini dipilih karena memiliki dua mode yaitu mode *in* dan mode *out* yang digunakan untuk mempercepat proses penurunan dan kenaikan suhu pada kumbung. Pengujian digunakan dengan menggunakan saklar virtual pada aplikasi Blynk untuk mempermudah saat pengujian kipas *exhaust*. Gambar pengujian dapat dilihat pada gambar 4.9.



Gambar 4.9 Kipas *Exhaust* dan Relay

Penggunaan mode *in* digunakan untuk membantu proses kenaikan kelembapan dan kenaikan suhu pada kumbung, sedangkan penggunaan mode *out* untuk mempercepat penurunan suhu pada kumbung. Gambar *Schematic* rangkaian dapat dilihat pada gambar 4.10.



Gambar 4.10 Schematic Rangkaian Kipas Exhaust

Hasil uji coba arah putaran kipas kontrol otomatis dengan menggunakan relay sebagai saklar dapat dilihat pada tabel 4.5 dan hasil uji coba arah putaran kipas dengan menggunakan saklar untuk kontrol manual dapat dilihat pada tabel 4.6.

Tabel 4.5 Uji Coba Kontrol Kipas dengan Menggunakan Relay

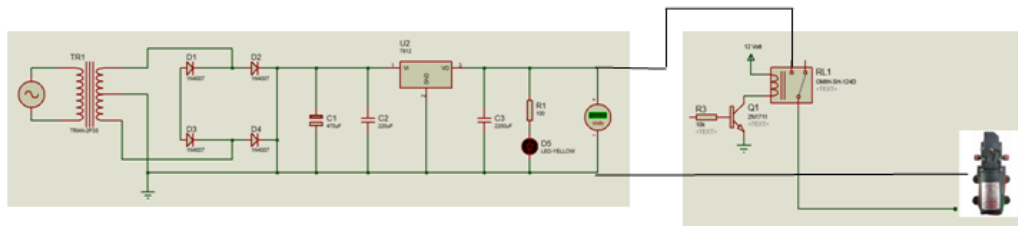
No	Relay <i>in</i>	Relay <i>out</i>	Arah kipas		Kondisi
			<i>In</i>	<i>Out</i>	
1	ON	OFF	V	-	Baik
2	OFF	ON	-	V	Baik
3	OFF	OFF	-	-	Baik

Tabel 4.6 Uji Coba Kontrol Kipas dengan Menggunakan Saklar

No	Saklar <i>in</i>	Saklar <i>out</i>	Arah kipas		Kondisi
			<i>In</i>	<i>Out</i>	
1	ON	OFF	V	-	Baik
2	OFF	ON	-	V	Baik
3	OFF	OFF	-	-	Baik

4.7 Rangkaian Pompa Listrik 12VDC

Pompa 12VDC adalah pompa air yang akan digunakan untuk mengalirkan air pada kumbung jamur tiram secara otomatis. Pompa ini memiliki spesifikasi lingkup tegangan 12 Volt, aliran maksimum 3,3 Liter/ Menit atau 580 Liter/ Jam, dan power 10 Watt



Gambar 4.11 Motor Listrik dan Adaptor

Hasil uji coba *spray* kontrol otomatis dengan menggunakan relay sebagai saklar dapat dilihat pada tabel 4.7 dan hasil uji coba *spray* dengan menggunakan saklar untuk kontrol manual dapat dilihat pada tabel 4.8.

Tabel 4.7 Uji Coba Kondisi *Sprayer* dengan Menggunakan Relay

No	Relay 1	Kondisi	Keterangan
1	ON	Menyala	Baik
2	OFF	Mati	Baik

Tabel 4.8 Uji Coba Kondisi *Sprayer* dengan Menggunakan Saklar

No	Saklar	Kondisi	Keterangan
1	ON	Menyala	Baik
2	OFF	Mati	Baik

4.8 Analisa Data

4.8.1 Pengambilan Data dengan *Prototype*

Pengambilan data dilakukan dengan membuat *prototype* kumbung jamur untuk mendapatkan data perbandingan kumbung dengan menggunakan alat kontrol otomatis dan manual. Pengambilan data menggunakan dua metode, metode pertama yaitu dengan membuat kumbung dalam kondisi manual dengan acuan alat ukur suhu dan kelembapan menggunakan *hygrometer*, sedangkan untuk kumbung yang menggunakan sistem kontrol otomatis menggunakan sensor DHT11 untuk pembacaan suhu dan kelembapannya. Gambar *prototype* yang kami buat dapat dilihat pada gambar 4.12.



Gambar 4.12 *Prototype* Kumbung Jamur dengan Sistem Kontrol Otomatis dan Manual

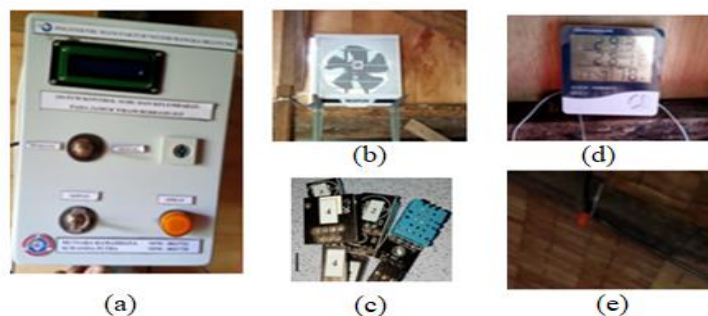
Pengambilan data dilakukan dalam jangka waktu satu minggu , data lebih lengkap dapat dilihat pada tabel 4.9.

Tabel 4.9 Perbandingan Suhu dan Kelembapan Menggunakan Sistem Kontrol dan *Hygrometer*

No	Hari	Tanggal	Waktu	Suhu		Kelembapan	
				Manual	Kontrol	Manual	Kontrol
1	Senin	13 Juli 2020	09.00WIB	30.00	26.00	79	87
2	Selasa	14 Juli 2020	09.05WIB	27.00	25.00	82	89
3	Rabu	15 Juli 2020	09.07WIB	27.00	25.00	83	89
4	Kamis	16 Juli 2020	09.10WIB	27.00	24.50	83	89
5	Jum'at	17 Juli 2020	09.00WIB	27.50	25.00	84	89
6	Sabtu	18 Juli 2020	09.03WIB	27.00	25.50	82	88
7	Minggu	19 Juli 2020	09.12WIB	29.50	26.00	80	89

4.8.2 Pengambilan Data pada Kumbung Jamur

Pengambilan data yang dilakukan saat sistem kontrol telah dipasang pada kumbung jamur, dapat dilihat pada gambar 4.13.



Gambar 4.13 (a) *Box* Kontrol, (b) Kipas *Exhaust*, (c) Sensor DHT11, (d) *Hygrometer*, (e) *Sprayer*

Hasil uji coba sistem kontrol yang telah dipasang di kumbung jamur dengan mengambil beberapa sampel data. Data dapat dilihat pada tabel 4.10.

Tabel 4.10 Hasil Uji Coba Sistem Kontrol

No	Waktu Awal	Manual		Otomatis		Rentan Waktu (Menit)	Manual		Otomatis	
		Suhu °C	RH %	Suhu °C	RH %		Suhu °C	RH %	Suhu °C	RH %
1	09.00	29	77	29.27	77.60	19	26	88	26.45	88.70
2	10.30	29	78	28.93	78.45	20	25	88	25.31	88.35
3	11.40	29	78	28.78	78.90	20	25	88	25.17	88.90

Pada data hasil percobaan dapat disimpulkan bahwa penurunan dari suhu dan kelembapan awal menjadi suhu dan kelembapan yang diinginkan membutuhkan waktu selama ± 20 menit dengan rentan waktu bertahan sampai ± 1 jam.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dari pengujian alat dan analisa serta fungsi alat yang berjudul "Sistem Kontrol Suhu dan Kelembapan pada Budidaya Jamur Tiram Berbasis IoT" dapat diketahui bahwa :

- Suhu yang diatur untuk kumbung budidaya jamur tiram dalam alat kontrol ini adalah dalam *range* 22-26 °C, sedangkan kelembapan berada pada rentang 80-90% RH.
- Persentase *error* pembacaan suhu yang dibaca oleh sensor DHT11 sebesar $\pm 0,015\%$, sedangkan untuk kelembapan persentase *error* sebesar $\pm 0.004\%$ dengan melakukan perbandingan dengan sistem pembacaan manual dengan alat ukur suhu dan kelembapan *hygrometer*.
- Nilai suhu dan kelembapan yang tampil pada aplikasi Blynk dapat di-*monitoring* walaupun petani pada kejauhan ± 100 meter.

5.2 Saran

Dari keseluruhan proyek akhir yang telah dikerjakan ini ada beberapa saran yang disampaikan untuk mengembangkan proyek akhir ini kedepannya, yaitu:

- Fitur dari NodeMCU ESP32 dapat digunakan lagi seperti fitur *bluetooth*, agar *monitoring* dapat dilakukan tanpa perlu akses Wi-Fi.
- Waktu pemasangan alat kontrol dapat dilakukan lebih cepat agar dapat *me-monitoring* hasil dari alat kontrol tersebut.
- Penggunaan sensor dapat menggunakan sensor yang memiliki *range* pembacaan nilai suhu kelembapan, dan akurasi yang lebih baik lagi seperti DHT22.

DAFTAR PUSTAKA

- Agusta, A. R. (2019). Implementasi Internet of Things Untuk Menjaga Kelembapan Udara Pada Budidaya Jamur. (95-100).
- Arafat, S. M. (2016, Oktober- Desember). Sistem Pengamanan Pintu Rumah Berbasis Internet Of Things (IoT) Dengan ESP8266 . *Technologia*, 7.
- Hadi, S. a. (2015). Pengatur Suhu dan Kelembapan pada Miniatur Kumbung Jamur Tiram. Diploma thesis, Institut Teknologi Sepuluh Nopember.
- Kadir, A. (2015). Buku Pintar Pemrograman Arduino. Jakarta, Indonesia: Penerbit Mediakom.
- L.G, Alexandromeo. (2017, April 19). Apa itu IOT ? Pengertian, Manfaat, Prinsip, Cara Kerja IOT. Retrieved Juli 9, 2020, from <https://makinrajin.com/blog/apa-itu-iot/>
- Lusidah, M., Taufik, M., & A.W.Purwandi. (2018). Rancang Bangun Alat Pengering Otomatis pada Proses Produksi Rumput. (119).
- Nunung. (2010). Budidaya Jamur Tiram.
- Prastyo, E. A. (2019, Juli 2). www.arduinoindonesia.id. (Arduino Indonesia) Retrieved Juli 10, 2020, from [arduinoindonesia: https://www.arduinoindonesia.id/2019/07/memulai-pemrograman-esp32 menggunakan.html](https://www.arduinoindonesia.id/2019/07/memulai-pemrograman-esp32-menggunakan.html)
- Riyanto, F. (2010). Pembibitan Jamur Tiram (*Pleurotus Ostreatus*) Di Balai Pengembangan Dan Promosi Tanaman Pangan Dan Hortikultura (BPPTPH). *digilib.uns.ac.id*. Ngipiksari Sleman, Yogyakarta.
- Rui Santos & Sara Santos. (n.d.). *ESP32 Web Server with Arduino IDE* .
- Safaat, N. (2012). *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Arduino*. Bandung: Penerbit Informatika.

Sejati, B. S. (2019). Sistem kendali *overhead crane* dengan *wireless control* menggunakan *smartphone* android dan tampilan LCD berbasis arduino. Sistem kendali *overhead crane* dengan *wireless control* menggunakan *smartphone* android dan tampilan LCD berbasis arduino.

UK, D.R. (2010, Juli 30). *Patent No. 2-5*.

Wahyono, R. (2016). Rancang Bangun Sistem Kendali Otomatis Temperatur dan Kelembapan Kumbung Jamur Tiram (*Pleurotus sp*) Berbasis Mikrokontroler.

LAMPIRAN 1

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Mutiara Ramadhana
Tempat, Tanggal Lahir : Koba, 15 Desember 1999
Alamat : Jalan Raya Arung Dalam,
Gg Merpati 2, No. 046,
Koba, Bangka Tengah
No. Hp : 081366696831
Status : Mahasiswi



2. Riwayat Pendidikan

SD Negeri 2 Koba	Lulus 2011
SMP Stania Koba	Lulus 2014
SMA Negeri 1 Koba	Lulus 2017
Politeknik Negeri Bangka Belitung	2017-sekarang

3. Pengalaman Kerja

Praktik Kerja Lapangan di PT. YOKOGAWA <i>Manufacturing</i> Batam (YMB)	02 September 2019-2 Januari 2020
--	----------------------------------

4. Hobi : Doodle Art

Sungailiat, 19 Agustus 2020

Mutiara Ramadhana

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Suwanda Putra
Tempat, Tanggal Lahir : Ciluak, 20 September 1998
Alamat : Jalan Raya Desa Ciluak,
Kecamatan Simpang Katis,
Bangka Tengah
No. Hp : 082175935923
Status : Mahasiswa



2. Riwayat Pendidikan

SD Negeri 4 Simpang Katis	Lulus 2011
SMP Negeri 2 Simpang Katis	Lulus 2014
SMA Negeri 2 Sungai Selan	Lulus 2017
Politeknik Negeri Bangka Belitung	2017-sekarang

3. Pengalaman Kerja

Praktik Kerja Lapangan di PT. TOSO <i>industry</i> Indonesia	02 September 2019-28 Desember 2019
---	---------------------------------------

4. Hobi : Mancing

Sungailiat, 19 Agustus 2020

Suwanda Putra

LAMPIRAN 2

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F,16,2);

#include <DHT.h>

#include <WiFi.h>

#include <WiFiClient.h>

#include <DHT.h>

#define BLYNK_PRINT Serial

#include <BlynkSimpleEsp32.h>

//koneksi internet

char auth[] = " -8b4005w9hJnk7uXPy2k-ynKuV8-rUE4";

char ssid[] = "Mutiara Ramadhana";

char pass[] = "kaktara12";

DHT dht1(26, DHT11); //Sensor 1

DHT dht2(27, DHT11); //Sensor 2

DHT dht3(32, DHT11); //Sensor 3

DHT dht4(33, DHT11); //Sensor 4

int relay1= 19; //kita menggunakan pin 19 untuk kipas
OUT

int relay2= 18; //kita menggunakan pin 18 untuk kipas
IN
```

```
int relay3= 15; //kita menggunakan pin 15 untuk pompa
DC12VDC

void setup() {

  lcd.init();

  lcd.backlight();

  dht1.begin();

  dht2.begin();

  dht3.begin();

  dht4.begin();

  BLYNK_READ(V0);

  BLYNK_READ(V1);

  pinMode(relay1, OUTPUT);

  pinMode(relay2, OUTPUT);

  pinMode(relay3, OUTPUT);

  digitalWrite(relay1, HIGH);

  digitalWrite(relay2, HIGH);

  digitalWrite(relay3, HIGH);

  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
```

```
void loop()

{

float kelembaban1 = dht1.readHumidity();

float kelembaban2 = dht2.readHumidity();

float kelembaban3 = dht3.readHumidity();

float kelembaban4 = dht4.readHumidity();

float suhu1 = dht1.readTemperature();

float suhu2 = dht2.readTemperature();

float suhu3 = dht3.readTemperature();

float suhu4 = dht4.readTemperature();

float kelembabanJumlah =
dht1.readHumidity()+dht2.readHumidity()+dht3.readHumidi
ty()+dht4.readHumidity();

float kelembabanHasil = kelembabanJumlah/4;

float suhuJumlah =
dht1.readTemperature()+dht2.readTemperature()+dht3.read
Temperature()+ dht4.readTemperature();

float suhuHasil = suhuJumlah/4;

delay(50);

Serial.print("RH1: ");
```

```
Serial.println(kelembaban1);

Serial.print(" ");

Serial.print("RH2: ");

Serial.println(kelembaban2);

Serial.print(" ");

Serial.print("RH3: ");

Serial.println(kelembaban3);

Serial.print(" ");

Serial.print("RH4: ");

Serial.println(kelembaban4);

Serial.print(" ");

Serial.print("RH TAMPIL: ");

Serial.println(kelembabanHasil);

Serial.print(" ");

Serial.print("T1: ");

Serial.println(suhu1);

Serial.print(" ");

Serial.print("T2: ");

Serial.println(suhu2);
```

```
Serial.print(" ");  
  
Serial.print("T3: ");  
  
Serial.println(suhu3);  
  
Serial.print(" ");  
  
Serial.print("T4: ");  
  
Serial.println(suhu4);  
  
Serial.print(" ");  
  
  
  
Serial.print("SUHU TAMPIL: ");  
  
Serial.println(suhuHasil);  
  
Serial.print(" ");  
  
  
  
//TAMPILAN LCD BOX  
  
lcd.clear();  
  
lcd.setCursor(0,0);  
  
lcd.print("SUHU:");  
  
lcd.setCursor(5,0);  
  
lcd.print(suhuHasil);  
  
  
  
lcd.setCursor(0,1);  
  
lcd.print("KELEMBAPAN:");
```

```

lcd.setCursor(11,1);

lcd.print(kelembabanHasil);

Blynk.virtualWrite(V0,suhuHasil);

Blynk.virtualWrite(V1,kelembabanHasil);

//KONTROL KELEMBAPAN KUMBUNG 80-90 RH

if (kelembapanHasil<80.00){//jika kelembapan dibawah 80
digitalWrite ( relay1, LOW);           //RELAY 1 MENYALA
digitalWrite ( relay2, HIGH );         //RELAY 2 MATI
digitalWrite ( relay3, LOW);           // RELAY 3 MENYALA
}

if (kelembapanHasil>=80.00 && kelembapanHasil<=90.00 )
{
//jika kelembapan diantara
80.00-90.00
digitalWrite ( relay1, HIGH);          // RELAY 1 MATI
digitalWrite ( relay2, HIGH );         // RELAY 2 MATI
digitalWrite ( relay3, HIGH);          // RELAY 3 MATI
}

if (kelembapanHasil>90.00){
// jika kelembapan diatas 90.00
digitalWrite ( relay1, HIGH );         // RELAY 1 MATI
digitalWrite ( relay2, LOW );          // RELAY 2 HIDUP
digitalWrite ( relay3, HIGH );         //RELAY 3 MATI
}

//KONTROL SUHU KUMBUNG 22 - 26°C
if ( suhuHasil<22.00 ){ //jika suhu dibawah 22.00
digitalWrite ( relay1, HIGH );         //RELAY 1 MATI
digitalWrite ( relay2, HIGH );         //RELAY 2 MATI
}

```

```
digitalWrite ( relay3, LOW );           //RELAY 3 HIDUP
}

if (suhuHasil>=22.00 && suhuHasil<=26.00)
{
    //jika suhu diantara 16.00-22.00
digitalWrite ( relay1, HIGH );         //RELAY 1 MATI
digitalWrite ( relay2, HIGH );         //RELAY 2 MATI
digitalWrite ( relay3, HIGH );         //RELAY 3 MATI
}

if (suhuHasil>26.00){                 //jika suhu diatas 26.00
digitalWrite ( relay1, HIGH );         //RELAY 1 MATI
digitalWrite ( relay2, LOW );          //RELAY 2 HIDUP
digitalWrite ( relay3, HIGH );         //RELAY 3 MATI
}
delay(5000);

}
```


LAMPIRAN 3

ESP32-WROOM-32

Datasheet



Version 2.8
Espressif Systems
Copyright © 2019

About This Document

This document provides the specifications for the ESP32-WROOM-32 module.

Revision History

For revision history of this document, please refer to the [last page](#).

Documentation Change Notification

Espressif provides email notifications to keep customers updated on changes to technical documentation. Please subscribe at www.espressif.com/en/subscribe.

Certification

Download certificates for Espressif products from www.espressif.com/en/certificates.

Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice. THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein. The Wi-Fi Alliance Member logo is a trademark of the Wi-Fi Alliance. The Bluetooth logo is a registered trademark of Bluetooth SIG.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.

Copyright © 2019 Espressif Inc. All rights reserved.

Contents

1 Overview	1
2 Pin Definitions	3
2.1 Pin Layout	3
2.2 Pin Description	3
2.3 Strapping Pins	4
3 Functional Description	6
3.1 CPU and Internal Memory	6
3.2 External Flash and SRAM	6
3.3 Crystal Oscillators	6
3.4 RTC and Low-Power Management	7
4 Peripherals and Sensors	8
5 Electrical Characteristics	9
5.1 Absolute Maximum Ratings	9
5.2 Recommended Operating Conditions	9
5.3 DC Characteristics (3.3 V, 25 °C)	9
5.4 Wi-Fi Radio	10
5.5 BLE Radio	11
5.5.1 Receiver	11
5.5.2 Transmitter	11
5.6 Reflow Profile	12
6 Schematics	13
7 Peripheral Schematics	14
8 Physical Dimensions	16
9 Recommended PCB Land Pattern	17
10 Learning Resources	18
10.1 Must-Read Documents	18
10.2 Must-Have Resources	18
Revision History	19

List of Tables

1	ESP32-WROOM-32 Specifications	1
2	Pin Definitions	3
3	Strapping Pins	5
4	Absolute Maximum Ratings	9
5	Recommended Operating Conditions	9
6	DC Characteristics (3.3 V, 25 °C)	9
7	Wi-Fi Radio Characteristics	10
8	Receiver Characteristics – BLE	11
9	Transmitter Characteristics – BLE	11

List of Figures

1	ESP32-WROOM-32 Pin Layout (Top View)	3
2	Reflow Profile	12
3	ESP32-WROOM-32 Schematics	13
4	ESP32-WROOM-32 Peripheral Schematics	14
5	Discharge Circuit for VDD33 Rail	14
6	Reset Circuit	15
7	Physical Dimensions of ESP32-WROOM-32	16
8	Recommended PCB Land Pattern	17

1. Overview

ESP32-WROOM-32 is a powerful, generic Wi-Fi+BT+BLE MCU module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding, music streaming and MP3 decoding.

At the core of this module is the ESP32-D0WDQ6 chip*. The chip embedded is designed to be scalable and adaptive. There are two CPU cores that can be individually controlled, and the CPU clock frequency is adjustable from 80 MHz to 240 MHz. The user may also power off the CPU and make use of the low-power co-processor to constantly monitor the peripherals for changes or crossing of thresholds. ESP32 integrates a rich set of peripherals, ranging from capacitive touch sensors, Hall sensors, SD card interface, Ethernet, high-speed SPI, UART, I²S and I²C.

Note:

* For details on the part numbers of the ESP32 family of chips, please refer to the document [ESP32 Datasheet](#).

The integration of Bluetooth, Bluetooth LE and Wi-Fi ensures that a wide range of applications can be targeted, and that the module is all-around: using Wi-Fi allows a large physical range and direct connection to the Internet through a Wi-Fi router, while using Bluetooth allows the user to conveniently connect to the phone or broadcast low energy beacons for its detection. The sleep current of the ESP32 chip is less than 5 μ A, making it suitable for battery powered and wearable electronics applications. The module supports a data rate of up to 150 Mbps, and 20 dBm output power at the antenna to ensure the widest physical range. As such the module does offer industry-leading specifications and the best performance for electronic integration, range, power consumption, and connectivity.

The operating system chosen for ESP32 is freeRTOS with LwIP; TLS 1.2 with hardware acceleration is built in as well. Secure (encrypted) over the air (OTA) upgrade is also supported, so that users can upgrade their products even after their release, at minimum cost and effort.

Table 1 provides the specifications of ESP32-WROOM-32.

Table 1: ESP32-WROOM-32 Specifications

Categories	Items	Specifications
Certification	RF certification	FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC
	Wi-Fi certification	Wi-Fi Alliance
	Bluetooth certification	BQB
	Green certification	RoHS/REACH
Test	Reliability	HTOL/HTSL/uHAST/TCT/ESD
Wi-Fi	Protocols	802.11 b/g/n (802.11n up to 150 Mbps)
		A-MPDU and A-MSDU aggregation and 0.4 μ s guard interval support
	Frequency range	2.4 GHz ~ 2.5 GHz
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and BLE specification
	Radio	NZIF receiver with -97 dBm sensitivity
		Class-1, class-2 and class-3 transmitter
		AFH

Categories	Items	Specifications
	Audio	CVSD and SBC
Hardware	Module interfaces	SD card, UART, SPI, SDIO, I ² C, LED PWM, Motor PWM, I ² S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC
	On-chip sensor	Hall sensor
	Integrated crystal	40 MHz crystal
	Integrated SPI flash	4 MB
	Operating voltage/Power supply	2.7 V ~ 3.6 V
	Operating current	Average: 80 mA
	Minimum current delivered by power supply	500 mA
	Recommended operating temperature range	-40 °C ~ +85 °C
	Package size	(18.00±0.10) mm × (25.50±0.10) mm × (3.10±0.10) mm

2. Pin Definitions

2.1 Pin Layout

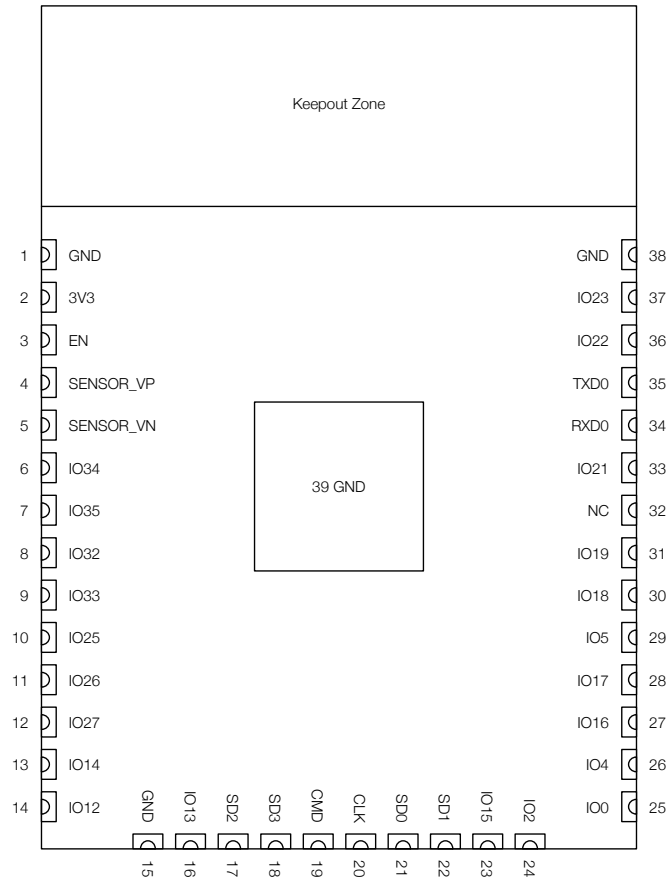


Figure 1: ESP32-WROOM-32 Pin Layout (Top View)

2.2 Pin Description

ESP32-WROOM-32 has 38 pins. See pin definitions in Table 2.

Table 2: Pin Definitions

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply
EN	3	I	Module-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8

Name	No.	Type	Function
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2*	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3*	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD*	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK*	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0*	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SDI/SD1*	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPICLK, HS2_DATA1, SD_DATA1, EMAC_TX_ER
IO16	27	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

Notice:

* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on the module and are not recommended for other uses.

2.3 Strapping Pins

ESP32 has five strapping pins, which can be seen in Chapter 6 Schematics:

- MTDI
- GPIO0
- GPIO2
- MTDO
- GPIO5

Software can read the values of these five bits from register "GPIO_STRAPPING".

During the chip's system reset release (power-on-reset, RTC watchdog reset and brownout reset), the latches of the strapping pins sample the voltage level as strapping bits of "0" or "1", and hold these bits until the chip is powered down or shut down. The strapping bits configure the device's boot mode, the operating voltage of VDD_SDIO and other initial system settings.

Each strapping pin is connected to its internal pull-up/pull-down during the chip reset. Consequently, if a strapping pin is unconnected or the connected external circuit is high-impedance, the internal weak pull-up/pull-down will determine the default input level of the strapping pins.

To change the strapping bit values, users can apply the external pull-down/pull-up resistances, or use the host MCU's GPIOs to control the voltage level of these pins when powering on ESP32.

After reset release, the strapping pins work as normal-function pins.

Refer to Table 3 for a detailed boot-mode configuration by strapping pins.

Table 3: Strapping Pins

Voltage of Internal LDO (VDD_SDIO)					
Pin	Default	3.3 V		1.8 V	
MTDI	Pull-down	0		1	
Bootling Mode					
Pin	Default	SPI Boot		Download Boot	
GPIO0	Pull-up	1		0	
GPIO2	Pull-down	Don't-care		0	
Enabling/Disabling Debugging Log Print over U0TXD During Bootling					
Pin	Default	U0TXD Active		U0TXD Silent	
MTDO	Pull-up	1		0	
Timing of SDIO Slave					
Pin	Default	Falling-edge Sampling Falling-edge Output	Falling-edge Sampling Rising-edge Output	Rising-edge Sampling Falling-edge Output	Rising-edge Sampling Rising-edge Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

Note:

- Firmware can configure register bits to change the settings of "Voltage of Internal LDO (VDD_SDIO)" and "Timing of SDIO Slave" after bootling.
- The module integrates a 3.3 V SPI flash, so the pin MTDI cannot be set to 1 when the module is powered up.

3. Functional Description

This chapter describes the modules and functions integrated in ESP32-WROOM-32.

3.1 CPU and Internal Memory

ESP32-D0WDQ6 contains two low-power Xtensa® 32-bit LX6 microprocessors. The internal memory includes:

- 448 KB of ROM for booting and core functions.
- 520 KB of on-chip SRAM for data and instructions.
- 8 KB of SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.
- 8 KB of SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the co-processor during the Deep-sleep mode.
- 1 Kbit of eFuse: 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including flash-encryption and chip-ID.

3.2 External Flash and SRAM

ESP32 supports multiple external QSPI flash and SRAM chips. More details can be found in Chapter SPI in the [ESP32 Technical Reference Manual](#). ESP32 also supports hardware encryption/decryption based on AES to protect developers' programs and data in flash.

ESP32 can access the external QSPI flash and SRAM through high-speed caches.

- The external flash can be mapped into CPU instruction memory space and read-only memory space simultaneously.
 - When external flash is mapped into CPU instruction memory space, up to 11 MB + 248 KB can be mapped at a time. Note that if more than 3 MB + 248 KB are mapped, cache performance will be reduced due to speculative reads by the CPU.
 - When external flash is mapped into read-only data memory space, up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads are supported.
- External SRAM can be mapped into CPU data memory space. Up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads and writes are supported.

ESP32-WROOM-32 integrates a 4 MB SPI flash, which is connected to GPIO6, GPIO7, GPIO8, GPIO9, GPIO10 and GPIO11. These six pins cannot be used as regular GPIOs.

3.3 Crystal Oscillators

The module uses a 40-MHz crystal oscillator.

3.4 RTC and Low-Power Management

With the use of advanced power-management technologies, ESP32 can switch between different power modes.

For details on ESP32's power consumption in different power modes, please refer to section "RTC and Low-Power Management" in [ESP32 Datasheet](#).

4. Peripherals and Sensors

Please refer to Section Peripherals and Sensors in [ESP32 Datasheet](#).

Note:

External connections can be made to any GPIO except for GPIOs in the range 6-11. These six GPIOs are connected to the module's integrated SPI flash. For details, please see Section 6 Schematics.

5. Electrical Characteristics

5.1 Absolute Maximum Ratings

Stresses beyond the absolute maximum ratings listed in Table 4 below may cause permanent damage to the device. These are stress ratings only, and do not refer to the functional operation of the device that should follow the [recommended operating conditions](#).

Table 4: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
VDD33	Power supply voltage	-0.3	3.6	V
I_{output}^1	Cumulative IO output current	-	1,100	mA
T_{store}	Storage temperature	-40	150	°C

1. The module worked properly after a 24-hour test in ambient temperature at 25 °C, and the IOs in three domains (VDD3P3_RTC, VDD3P3_CPU, VDD_SDIO) output high logic level to ground. Please note that pins occupied by flash and/or PSRAM in the VDD_SDIO power domain were excluded from the test.
2. Please see Appendix IO_MUX of [ESP32 Datasheet](#) for IO's power domain.

5.2 Recommended Operating Conditions

Table 5: Recommended Operating Conditions

Symbol	Parameter	Min	Typical	Max	Unit
VDD33	Power supply voltage	2.7	3.3	3.6	V
I_{VDD}	Current delivered by external power supply	0.5	-	-	A
T	Operating temperature	-40	-	85	°C

5.3 DC Characteristics (3.3 V, 25 °C)

Table 6: DC Characteristics (3.3 V, 25 °C)

Symbol	Parameter		Min	Typ	Max	Unit
C_{IN}	Pin capacitance		-	2	-	pF
V_{IH}	High-level input voltage		$0.75 \times VDD^1$	-	$VDD^1 + 0.3$	V
V_{IL}	Low-level input voltage		-0.3	-	$0.25 \times VDD^1$	V
I_{IH}	High-level input current		-	-	50	nA
I_{IL}	Low-level input current		-	-	50	nA
V_{OH}	High-level output voltage		$0.8 \times VDD^1$	-	-	V
V_{OL}	Low-level output voltage		-	-	$0.1 \times VDD^1$	V
I_{OH}	High-level source current ($VDD^1 = 3.3$ V, $V_{OH} \geq 2.64$ V, output drive strength set to the maximum)	VDD3P3_CPU power domain ^{1, 2}	-	40	-	mA
		VDD3P3_RTC power domain ^{1, 2}	-	40	-	mA
		VDD_SDIO power domain ^{1, 3}	-	20	-	mA

Symbol	Parameter	Min	Typ	Max	Unit
I_{OL}	Low-level sink current ($V_{DD}^1 = 3.3\text{ V}$, $V_{OL} = 0.495\text{ V}$, output drive strength set to the maximum)	-	28	-	mA
R_{PU}	Resistance of internal pull-up resistor	-	45	-	$k\Omega$
R_{PD}	Resistance of internal pull-down resistor	-	45	-	$k\Omega$
V_{IL_nRST}	Low-level input voltage of CHIP_PU to power off the chip	-	-	0.6	V

Notes:

1. Please see Appendix IO_MUX of [ESP32 Datasheet](#) for IO's power domain. VDD is the I/O voltage for a particular power domain of pins.
2. For VDD3P3_CPU and VDD3P3_RTC power domain, per-pin current sourced in the same domain is gradually reduced from around 40 mA to around 29 mA, $V_{OH} \geq 2.64\text{ V}$, as the number of current-source pins increases.
3. Pins occupied by flash and/or PSRAM in the VDD_SDIO power domain were excluded from the test.

5.4 Wi-Fi Radio

Table 7: Wi-Fi Radio Characteristics

Parameter	Condition	Min	Typical	Max	Unit
Input frequency	-	2412	-	2484	MHz
Output impedance*	-	-	*	-	Ω
TX power	11n, MCS7	12	13	14	dBm
	11b mode	17.5	18.5	20	dBm
Sensitivity	11b, 1 Mbps	-	-98	-	dBm
	11b, 11 Mbps	-	-89	-	dBm
	11g, 6 Mbps	-	-92	-	dBm
	11g, 54 Mbps	-	-74	-	dBm
	11n, HT20, MCS0	-	-91	-	dBm
	11n, HT20, MCS7	-	-71	-	dBm
	11n, HT40, MCS0	-	-89	-	dBm
	11n, HT40, MCS7	-	-69	-	dBm
Adjacent channel rejection	11g, 6 Mbps	-	31	-	dB
	11g, 54 Mbps	-	14	-	dB
	11n, HT20, MCS0	-	31	-	dB
	11n, HT20, MCS7	-	13	-	dB

*For the modules that use IPEX antennas, the output impedance is 50 Ω . For other modules without IPEX antennas, users do not need to concern about the output impedance.

5.5 BLE Radio

5.5.1 Receiver

Table 8: Receiver Characteristics – BLE

Parameter	Conditions	Min	Typ	Max	Unit
Sensitivity @30.8% PER	-	-	-97	-	dBm
Maximum received signal @30.8% PER	-	0	-	-	dBm
Co-channel C/I	-	-	+10	-	dB
Adjacent channel selectivity C/I	$F = F_0 + 1 \text{ MHz}$	-	-5	-	dB
	$F = F_0 - 1 \text{ MHz}$	-	-5	-	dB
	$F = F_0 + 2 \text{ MHz}$	-	-25	-	dB
	$F = F_0 - 2 \text{ MHz}$	-	-35	-	dB
	$F = F_0 + 3 \text{ MHz}$	-	-25	-	dB
	$F = F_0 - 3 \text{ MHz}$	-	-45	-	dB
Out-of-band blocking performance	30 MHz ~ 2000 MHz	-10	-	-	dBm
	2000 MHz ~ 2400 MHz	-27	-	-	dBm
	2500 MHz ~ 3000 MHz	-27	-	-	dBm
	3000 MHz ~ 12.5 GHz	-10	-	-	dBm
Intermodulation	-	-36	-	-	dBm

5.5.2 Transmitter

Table 9: Transmitter Characteristics – BLE

Parameter	Conditions	Min	Typ	Max	Unit
RF transmit power	-	-	0	-	dBm
Gain control step	-	-	3	-	dBm
RF power control range	-	-12	-	+9	dBm
Adjacent channel transmit power	$F = F_0 \pm 2 \text{ MHz}$	-	-52	-	dBm
	$F = F_0 \pm 3 \text{ MHz}$	-	-58	-	dBm
	$F = F_0 \pm > 3 \text{ MHz}$	-	-60	-	dBm
Δf_{1avg}	-	-	-	265	kHz
Δf_{2max}	-	247	-	-	kHz
$\Delta f_{2avg}/\Delta f_{1avg}$	-	-	-0.92	-	-
ICFT	-	-	-10	-	kHz
Drift rate	-	-	0.7	-	kHz/50 μ s
Drift	-	-	2	-	kHz

5.6 Reflow Profile

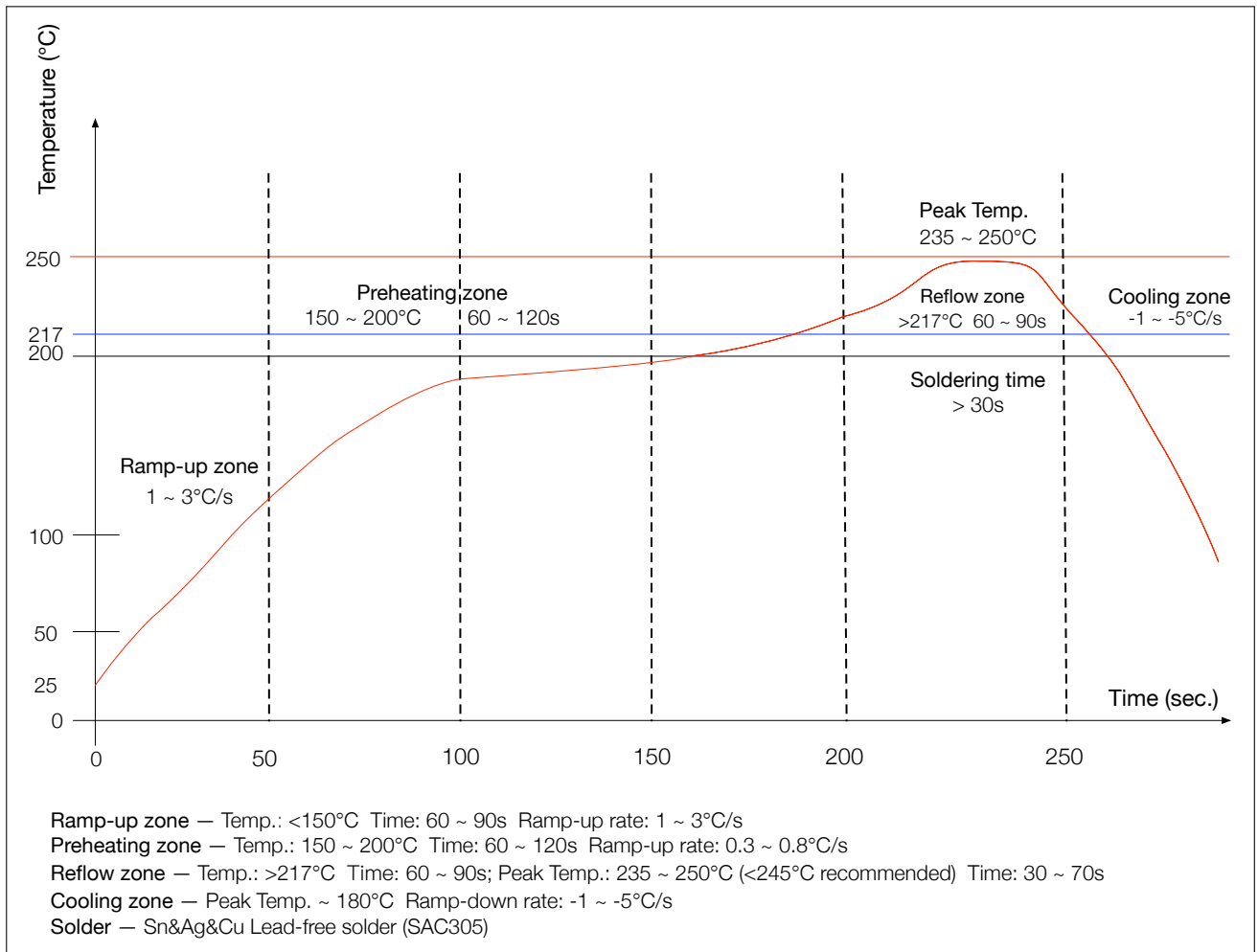


Figure 2: Reflow Profile

6. Schematics

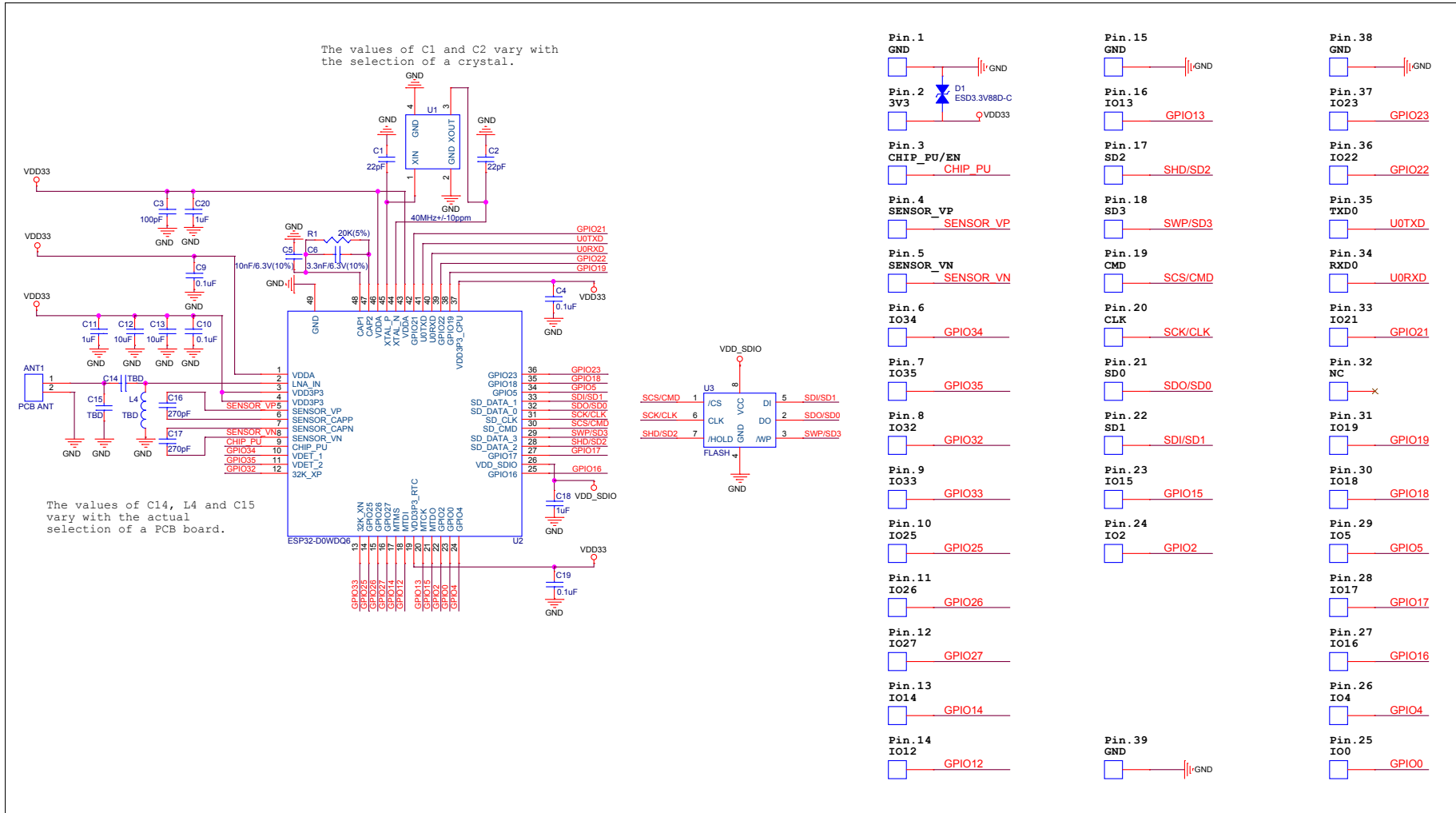


Figure 3: ESP32-WROOM-32 Schematics

7. Peripheral Schematics

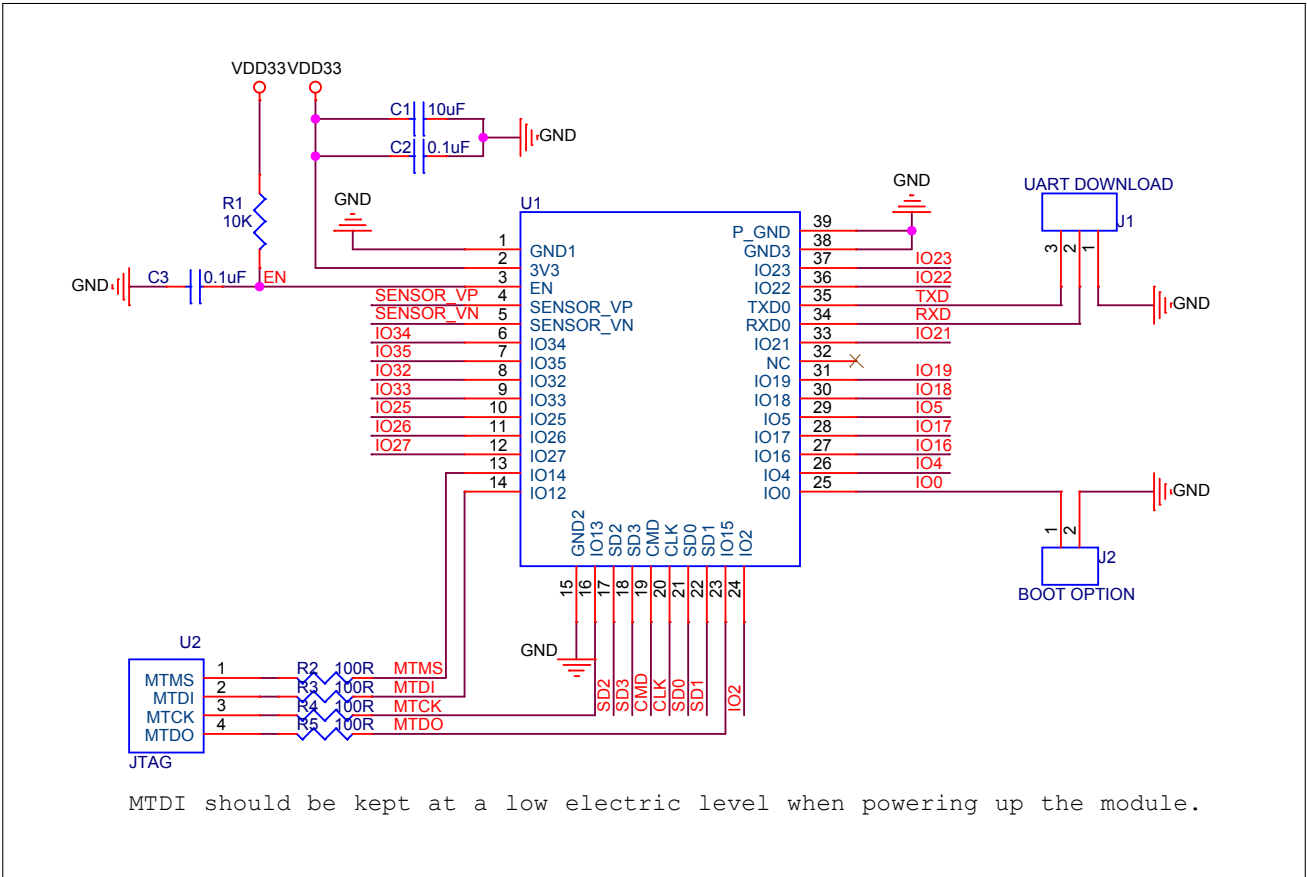


Figure 4: ESP32-WROOM-32 Peripheral Schematics

Note:

Soldering Pad 39 to the Ground of the base board is not necessary for a satisfactory thermal performance. If users do want to solder it, they need to ensure that the correct quantity of soldering paste is applied.

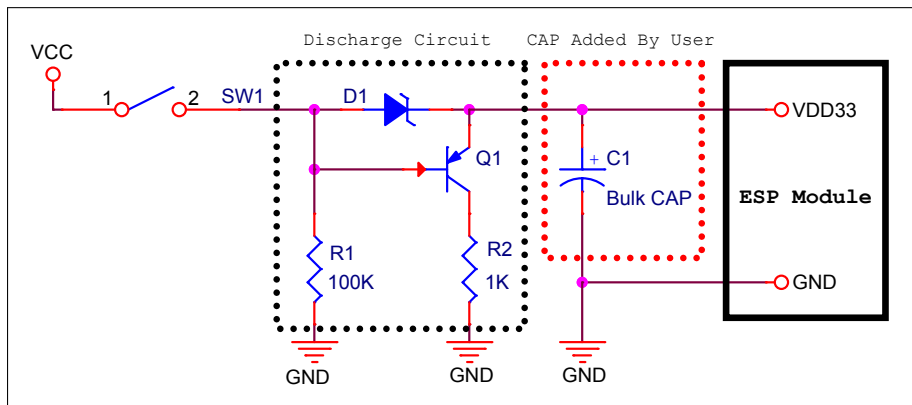
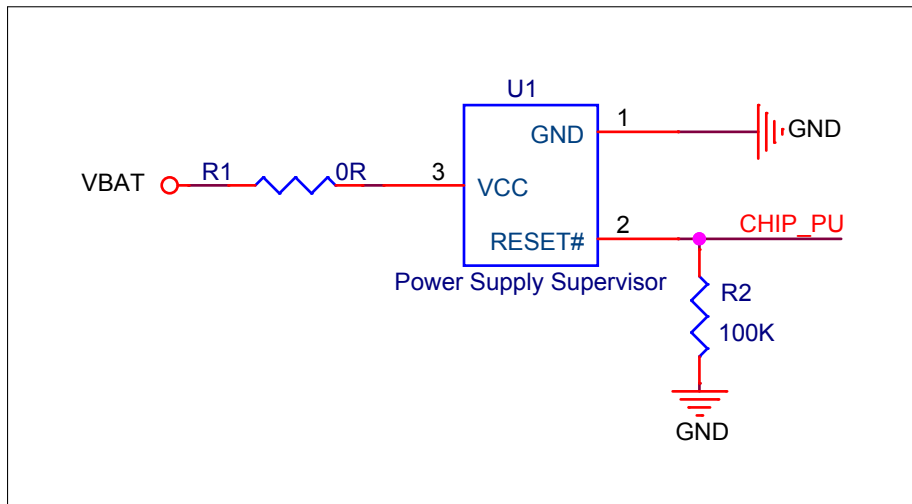


Figure 5: Discharge Circuit for VDD33 Rail

Note:

The discharge circuit can be applied in scenarios where ESP32 is powered on and off repeatedly by switching the power rails, and there is a large capacitor on the VDD33 rail. For details, please refer to Section **Power Scheme** in [ESP32 Datasheet](#).

**Figure 6: Reset Circuit****Note:**

When battery is used as the power supply for ESP32 series of chips and modules, a supply voltage supervisor is recommended to avoid boot failure due to low voltage. Users are recommended to pull CHIP_PU low if the power supply for ESP32 is below 2.3 V.

8. Physical Dimensions

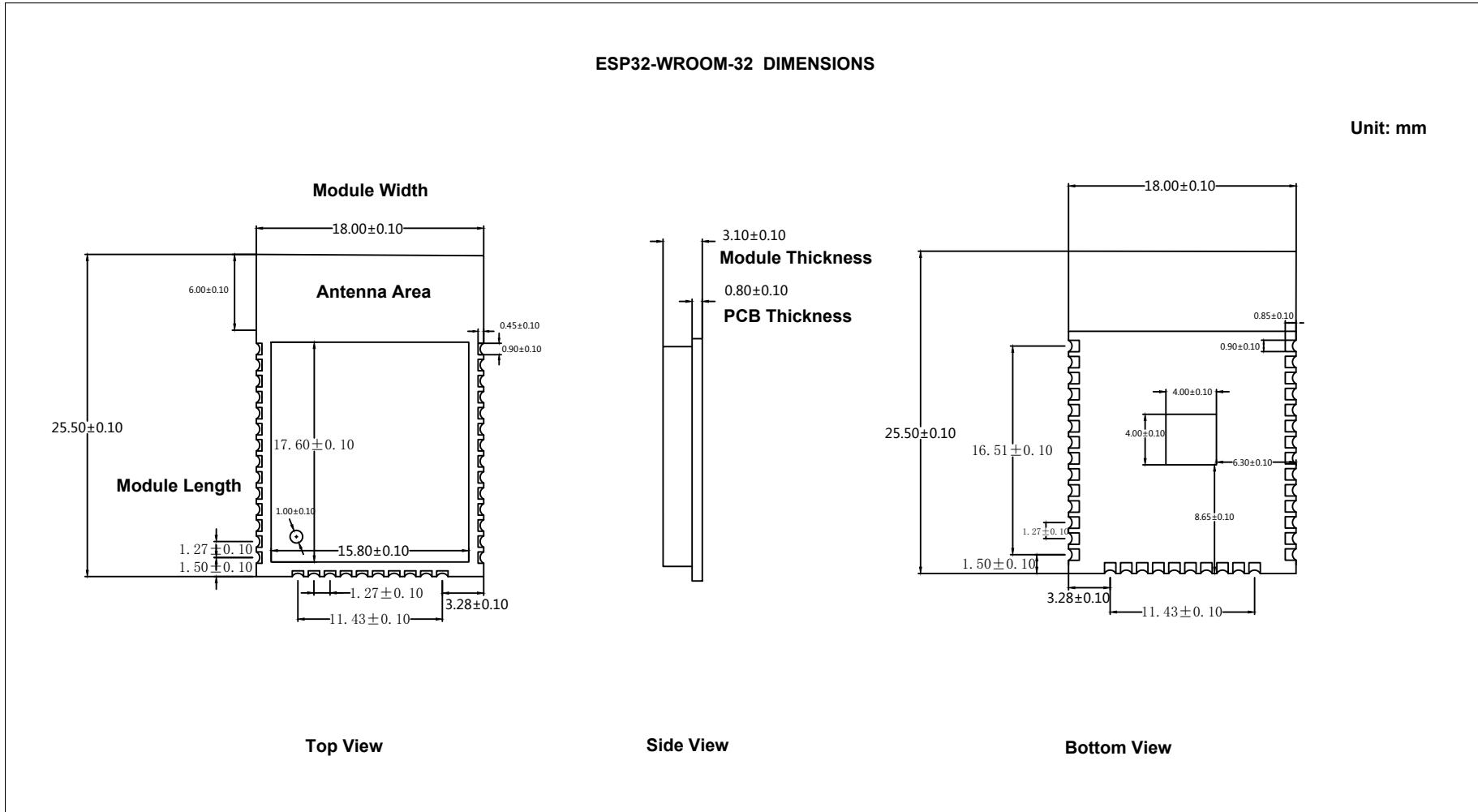
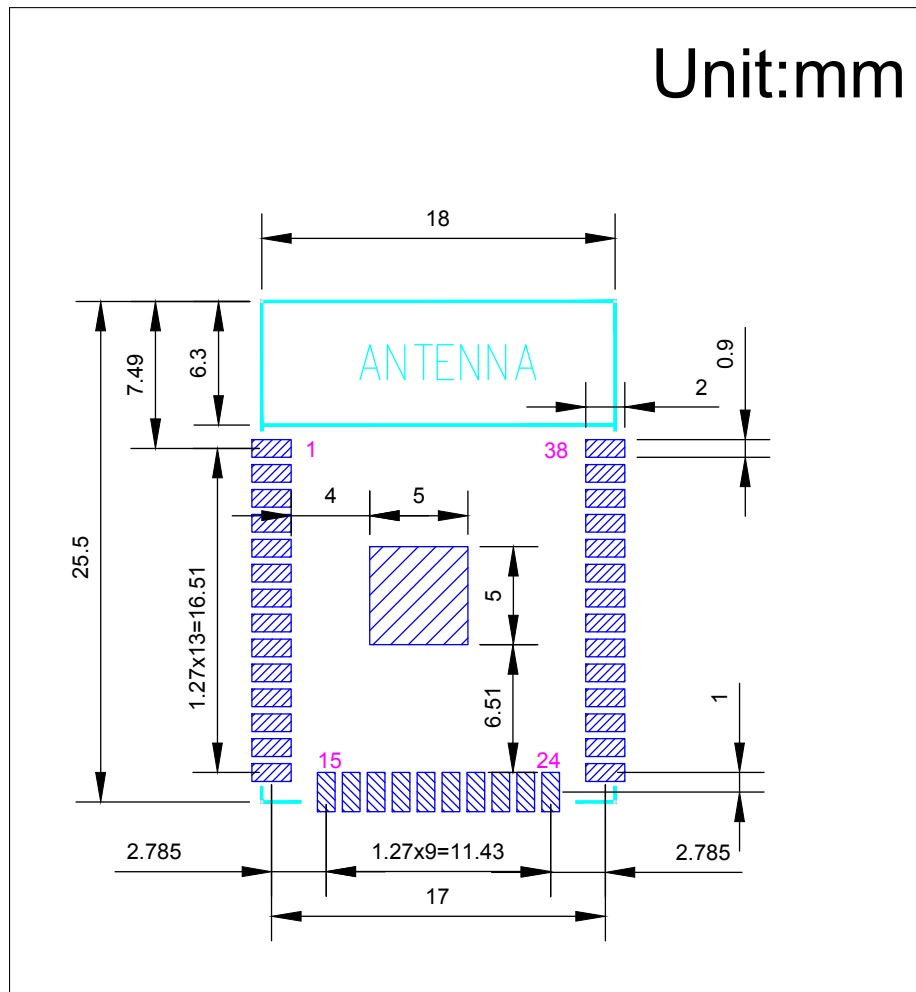


Figure 7: Physical Dimensions of ESP32-WROOM-32

9. Recommended PCB Land Pattern



10. Learning Resources

10.1 Must-Read Documents

The following link provides documents related to ESP32.

- [ESP32 Datasheet](#)
This document provides an introduction to the specifications of the ESP32 hardware, including overview, pin definitions, functional description, peripheral interface, electrical characteristics, etc.
- [ESP-IDF Programming Guide](#)
It hosts extensive documentation for ESP-IDF ranging from hardware guides to API reference.
- [ESP32 Technical Reference Manual](#)
The manual provides detailed information on how to use the ESP32 memory and peripherals.
- [ESP32 Hardware Resources](#)
The zip files include the schematics, PCB layout, Gerber and BOM list of ESP32 modules and development boards.
- [ESP32 Hardware Design Guidelines](#)
The guidelines outline recommended design practices when developing standalone or add-on systems based on the ESP32 series of products, including the ESP32 chip, the ESP32 modules and development boards.
- [ESP32 AT Instruction Set and Examples](#)
This document introduces the ESP32 AT commands, explains how to use them, and provides examples of several common AT commands.
- [Espressif Products Ordering Information](#)

10.2 Must-Have Resources

Here are the ESP32-related must-have resources.

- [ESP32 BBS](#)
This is an Engineer-to-Engineer (E2E) Community for ESP32 where you can post questions, share knowledge, explore ideas, and help solve problems with fellow engineers.
- [ESP32 GitHub](#)
ESP32 development projects are freely distributed under Espressif's MIT license on GitHub. It is established to help developers get started with ESP32 and foster innovation and the growth of general knowledge about the hardware and software surrounding ESP32 devices.
- [ESP32 Tools](#)
This is a webpage where users can download ESP32 Flash Download Tools and the zip file "ESP32 Certification and Test".
- [ESP-IDF](#)
This webpage links users to the official IoT development framework for ESP32.
- [ESP32 Resources](#)
This webpage provides the links to all available ESP32 documents, SDK and tools.

Revision History

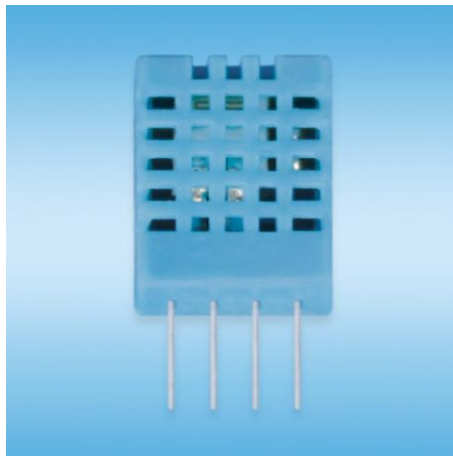
Date	Version	Release notes
2019.01	V2.8	Changed the RF power control range in Table 9 from $-12 \sim +12$ to $-12 \sim +9$ dBm.
2018.10	V2.7	Added "Cumulative IO output current" entry to Table 4: Absolute Maximum Ratings; Added more parameters to Table 6: DC Characteristics.
2018.08	V2.6	<ul style="list-style-type: none"> Added reliability test items the module has passed in Table 1: ESP32-WROOM-32 Specifications, and removed software-specific information; Updated section 3.4: RTC and Low-Power Management; Changed the module's dimensions from (18 ± 0.2) mm x (25.5 ± 0.2) mm x (3.1 ± 0.15) mm to (18.00 ± 0.10) mm x (25.50 ± 0.10) mm x (3.10 ± 0.10) mm; Updated Figure 8: Physical Dimensions; Updated Table 7: Wi-Fi Radio.
2018.06	V2.5	<ul style="list-style-type: none"> Changed the module name to ESP32-WROOM-32; Deleted Temperature Sensor in Table 1: ESP32-WROOM-32 Specifications; Updated Chapter 3: Functional Description; Added Chapter 8: Recommended PCB Land Pattern; Changes to electrical characteristics: <ul style="list-style-type: none"> Updated Table 4: Absolute Maximum Ratings; Added Table 5: Recommended Operating Conditions; Added Table 6: DC Characteristics; Updated the values of "Gain control step", "Adjacent channel transmit power" in Table 9: Transmitter Characteristics - BLE.
2018.03	V2.4	Updated Table 1 in Chapter 1.
2018.01	V2.3	Deleted information on LNA pre-amplifier; Updated section 3.4 RTC and Low-Power Management; Added reset circuit in Chapter 7 and a note to it.
2017.10	V2.2	Updated the description of the chip's system reset in Section 2.3 Strapping Pins; Deleted "Association sleep pattern" in Table "Power Consumption by Power Modes" and added notes to Active sleep and Modem-sleep; Updated the note to Figure 4 Peripheral Schematics; Added discharge circuit for VDD33 rail in Chapter 7 and a note to it.
2017.09	V2.1	Updated operating voltage/power supply range updated to $2.7 \sim 3.6$ V; Updated Chapter 7.
2017.08	V2.0	Changed the sensitivity of NZIF receiver to -97 dBm in Table 1; Updated the dimensions of the module; Updated Table "Power Consumption by Power Modes" Power Consumption by Power Modes, and added two notes to it; Updated Table 4, 7, 8, 9; Added Chapter 8; Added the link to certification download .
2017.06	V1.9	Added a note to Section 2.1 Pin Layout; Updated Section 3.3 Crystal Oscillators; Updated Figure 3 ESP-WROOM-32 Schematics;

Date	Version	Release notes
		Added Documentation Change Notification.
2017.05	V1.8	Updated Figure 1 Top and Side View of ESP32-WROOM-32 (ESP-WROOM-32).
2017.04	V1.7	Added the module's dimensional tolerance; Changed the input impedance value of 50Ω in Table 7 Wi-Fi Radio Characteristics to output impedance value of $30+j10\Omega$.
2017.04	V1.6	Added Figure 2 Reflow Profile.
2017.03	V1.5	Updated Section 2.2 Pin Description; Updated Section 3.2 External Flash and SRAM; Updated Section 4 Peripherals and Sensors Description.
2017.03	V1.4	Updated Chapter 1 Preface; Updated Chapter 2 Pin Definitions; Updated Chapter 3 Functional Description; Updated Table Recommended Operating Conditions; Updated Table 7 Wi-Fi Radio Characteristics; Updated Section 5.6 Reflow Profile; Added Chapter 10 Learning Resources.
2016.12	V1.3	Updated Section 2.1 Pin Layout.
2016.11	V1.2	Added Figure 7 Peripheral Schematics.
2016.11	V1.1	Updated Chapter 6 Schematics.
2016.08	V1.0	First release.

AOSONG

Temperature and humidity module

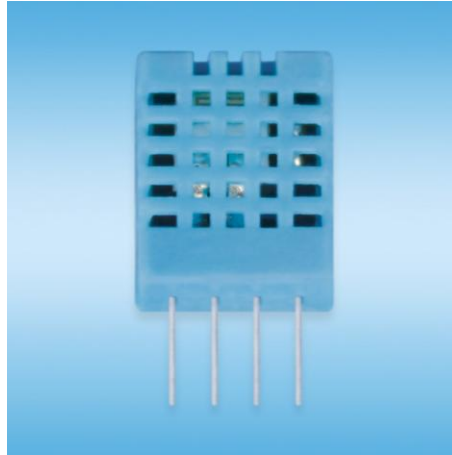
DHT11 Product Manual



www.aosong.com

1、 Product Overview

DHT11 digital temperature and humidity sensor is a composite Sensor contains a calibrated digital signal output of the temperature and humidity. Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long-term stability. The sensor includes a resistive sense of wet components and an NTC temperature measurement devices, and connected with a high-performance 8-bit microcontroller.



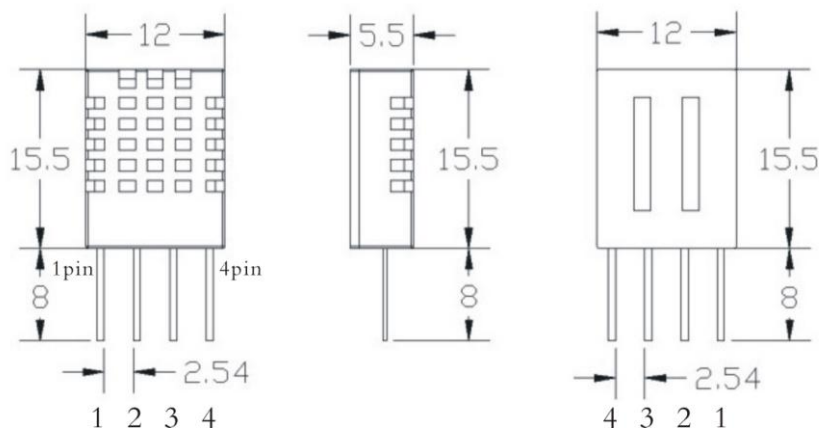
2、 Applications

HVAC, dehumidifier, testing and inspection equipment, consumer goods, automotive, automatic control, data loggers, weather stations, home appliances, humidity regulator, medical and other humidity measurement and control.

3、 Features

Low cost, long-term stability, relative humidity and temperature measurement, excellent quality, fast response, strong anti-interference ability, long distance signal transmission, digital signal output, and precise calibration.

4、 Dimensions (unit: mm)



5、Product parameters

Relative humidity

Resolution: 16Bit

Repeatability: $\pm 1\%$ RH

Accuracy: At 25°C $\pm 5\%$ RH

Interchangeability: fully interchangeable

Response time: 1 / e (63%) of 25°C 6s

1m / s air 6s

Hysteresis: $< \pm 0.3\%$ RH

Long-term stability: $< \pm 0.5\%$ RH / yr in

Temperature

Resolution: 16Bit

Repeatability: $\pm 0.2^\circ\text{C}$

Range: At 25°C $\pm 2^\circ\text{C}$

Response time: 1 / e (63%) 10S

Electrical Characteristics

Power supply: DC 3.5 ~ 5.5V

Supply Current: measurement 0.3mA standby 60 μ A

Sampling period: more than 2 seconds

Pin Description

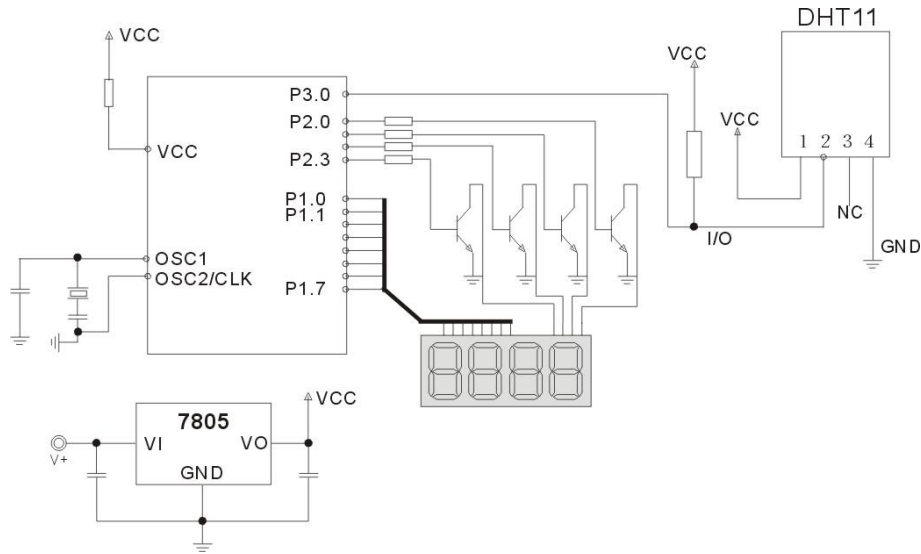
1, the VDD power supply 3.5 ~ 5.5V DC

2 DATA serial data, a single bus

3, NC, empty pin

4, GND ground, the negative power

6、 Typical circuit



Microprocessor and DHT11 of connection typical application circuit as shown above, DATA pull the microprocessor I / O ports are connected.

1. Typical application circuit recommended in the short cable length of 20 meters on the 5.1K pull-up resistor, the resistance of greater than 20 meters under the pull-up resistor on the lower of the actual situation.
2. When using a 3.5V voltage supply cable length shall not be greater than 20cm. Otherwise, the line voltage drop will cause the sensor power supply shortage, caused by measurement error.
3. Each read out the temperature and humidity values are the results of the last measurement For real-time data, sequential read twice, but is not recommended to repeatedly read the sensors, each read sensor interval is greater than 5 seconds can be obtained accurate data.

7、 Serial communication instructions (single-wire bi-directional)

©Single bus Description

DHT11 uses a simplified single-bus communication. Single bus that only one data line, the system of data exchange, control by a single bus to complete. Device (master or slave) through an open-drain or tri-state port connected to the data line to allow the device does not send data to release the bus, while other devices use the bus; single bus usually require an external one about 5.1k Ω pull-up resistor, so that when the bus is idle, its status is high. Because they are the master-slave structure, and only when the host calls the slave, the slave can answer, the host access devices must strictly follow the single-bus sequence, if the chaotic sequence, the device will not respond to the host.

©Single bus to transfer data defined

DATA For communication and synchronization between the microprocessor and DHT11, single-bus data format, a transmission of 40 data, the high first-out.

Data format:

The 8bit humidity integer data + 8bit the Humidity decimal data +8 bit temperature integer data + 8bit fractional temperature data +8 bit parity bit.

©Parity bit data definition

“8bit humidity integer data + 8bit humidity decimal data +8 bit temperature integer data + 8bit temperature fractional data” 8bit checksum is equal to the results of the last eight.

Example 1: 40 data is received:

<u>0011 0101</u>	<u>0000 0000</u>	<u>0001 1000</u>	<u>0000 0000</u>	<u>0100 1101</u>
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

Calculate:

$$0011\ 0101 + 0000\ 0000 + 0001\ 1000 + 0000\ 0000 = 0100\ 1101$$

Received data is correct;

Humidity: 0011 0101=35H=53%RH

Temperature: 0001 1000=18H=24°C

Example 2: 40 data is received:

<u>0011 0101</u>	<u>0000 0000</u>	<u>0001 1000</u>	<u>0000 0000</u>	<u>0100 1001</u>
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

Calculate:

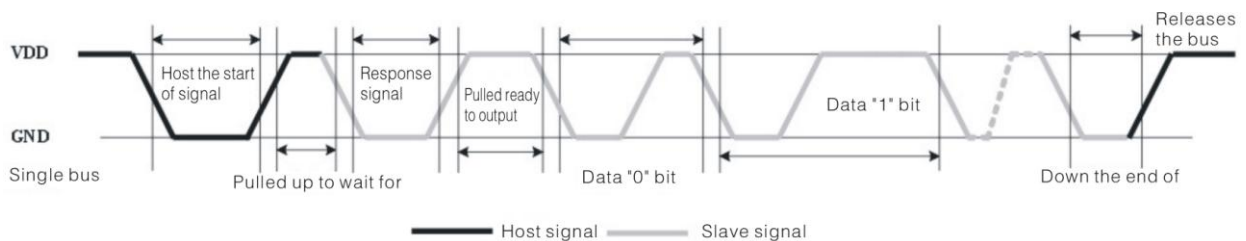
$$0011\ 0101 + 0000\ 0000 + 0001\ 1000 + 0000\ 0000 = 0100\ 1101$$

$$01001101 \neq 0100\ 1001$$

The received data is not correct, give up, to re-receive data.

©Data Timing Diagram

User host (MCU) to send a signal, DHT11 converted from low-power mode to high-speed mode, until the host began to signal the end of the DHT11 send a response signal to send 40bit data, and trigger a letter collection. The signal is sent as shown.



Data Timing Diagram

Note: The host reads the temperature and humidity data from DHT11 always the last measured value, such as twice the measured interval of time is very long, continuous read twice to the second value of real-time temperature and humidity values.

©Peripherals read steps

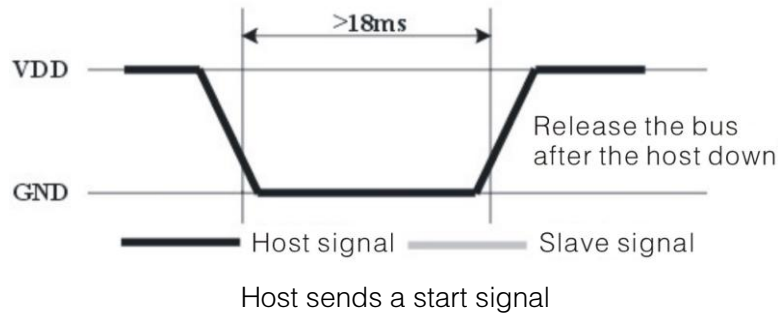
Communication between the master and slave can be done through the following steps (peripherals (such as microprocessors) read DHT11 the data of steps).

Step 1:

After power on DHT11 (DHT11 on after power to wait 1S across the unstable state during this period can not send any instruction), the test environment temperature and humidity data, and record the data, while DHT11 the DATA data lines pulled by pull-up resistor has been to maintain high; the DHT11 the DATA pin is in input state, the moment of detection of external signals.

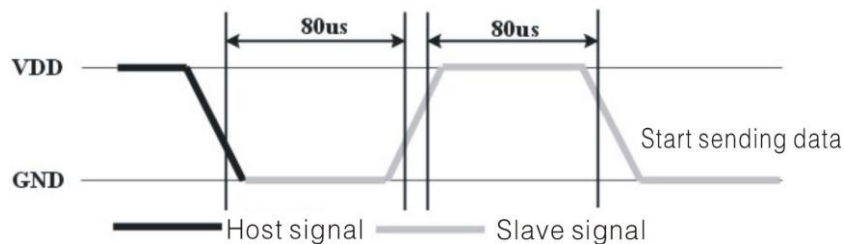
Step 2:

Microprocessor I / O set to output at the same time output low, and low hold time can not be less than 18ms, then the microprocessor I / O is set to input state, due to the pull-up resistor, a microprocessor/ O DHT11 the dATA data lines also will be high, waiting DHT11 to answer signal, send the signal as shown:



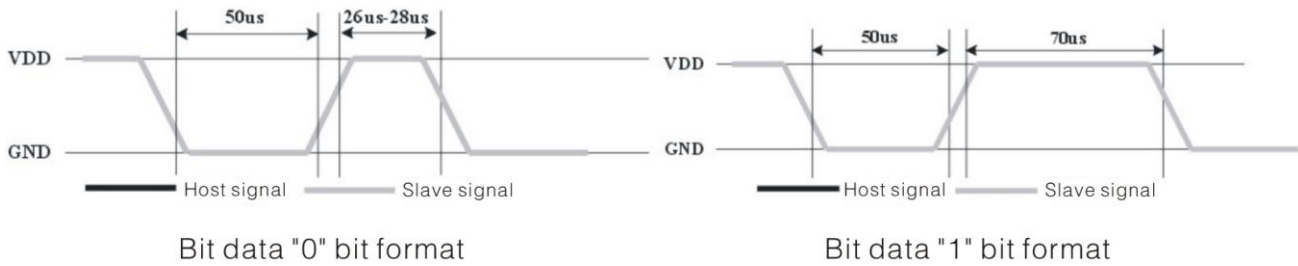
Step 3:

DATA pin is detected to an external signal of DHT11 low, waiting for external signal low end the delay DHT11 DATA pin in the output state, the output low of 80 microseconds as the response signal, followed by the output of 80 micro-seconds of high notification peripheral is ready to receive data, the microprocessor I / O at this time in the input state is detected the I / O low (DHT11 response signal), wait 80 microseconds highdata receiving and sending signals as shown:



Step 4:

Output by DHT11 the DATA pin 40, the microprocessor receives 40 data bits of data "0" format: the low level of 50 microseconds and 26-28 microseconds according to the changes in the I / O levellevel, bit data "1" format: the high level of low plus, 50 microseconds to 70 microseconds. Bit data "0", "1" signal format as shown:



End signal:

Continue to output the low 50 microseconds after DHT11 the DATA pin output 40 data, and changed the input state, along with pull-up resistor goes high. But DHT11 internal re-test environmental temperature and humidity data, and record the data, waiting for the arrival of the external signal.

8、Application of information

1. Work and storage conditions

Outside the sensor the proposed scope of work may lead to temporary drift of the signal up to 300%RH. Return to normal working conditions, sensor calibration status will slowly toward recovery. To speed up the recovery process may refer to "resume processing". Prolonged use of non-normal operating conditions, will accelerate the aging of the product.

Avoid placing the components on the long-term condensation and dry environment, as well as the following environment.

A, salt spray

B, acidic or oxidizing gases such as sulfur dioxide, hydrochloric acid

Recommended storage environment

Temperature: 10 ~ 40 °C Humidity: 60% RH or less

2. The impact of exposure to chemicals

The capacitive humidity sensor has a layer by chemical vapor interference, the proliferation of chemicals in the sensing layer may lead to drift and decreased sensitivity of the measured values. In a pure environment, contaminants will slowly be released. Resume processing as described below will accelerate this process. The high concentration of chemical pollution (such as ethanol) will lead to the complete damage of the sensitive layer of the sensor.

3. The temperature influence

Relative humidity of the gas to a large extent dependent on temperature. Therefore, in the measurement of humidity, should be to ensure that the work of the humidity sensor at the same temperature. With the release of heat of electronic components share a printed circuit board, the installation should be as far as possible the sensor away from the electronic components and mounted below the heat source, while maintaining good ventilation of the enclosure. To reduce the thermal conductivity sensor and printed circuit board copper plating should be the smallest possible, and leaving a gap between the two.

4. Light impact

Prolonged exposure to sunlight or strong ultraviolet radiation, and degrade performance.

5. Resume processing

Placed under extreme working conditions or chemical vapor sensor, which allows it to return to the status of calibration by the following handler. Maintain two hours in the humidity conditions of 45°C and <10% RH (dry); followed by 20–30°C and > 70% RH humidity conditions to maintain more than five hours.

6. Wiring precautions

The quality of the signal wire will affect the quality of the voltage output, it is recommended to use high quality shielded cable.

7. Welding information

Manual welding, in the maximum temperature of 300°C under the conditions of contact time shall be less than 3 seconds.

8. Product upgrades

Details, please the consultation Aosong electronics department.

9、 The license agreement

Without the prior written permission of the copyright holder, shall not in any form or by any means, electronic or mechanical (including photocopying), copy any part of this manual, nor shall its contents be communicated to a third party. The contents are subject to change without notice.

The Company and third parties have ownership of the software, the user may use only signed a contract or software license.

10、 Warnings and personal injury

This product is not applied to the safety or emergency stop devices, as well as the failure of the product may result in injury to any other application, unless a particular purpose or use authorized. Installation, handling, use or maintenance of the product refer to product data sheets and application notes. Failure to comply with this recommendation may result in death and serious personal injury. The Company will bear all damages resulting personal injury or death, and waive any claims that the resulting subsidiary company managers and employees and agents, distributors, etc. that may arise, including: a variety of costs, compensation costs, attorneys' fees, and so on.

11、 Quality Assurance

The company and its direct purchaser of the product quality guarantee period of three months (from the date of delivery). Publishes the technical specifications of the product data sheet shall prevail. Within the warranty period, the product was confirmed that the quality is really defective, the company will provide free repair or replacement. The user must satisfy the following conditions:

- ① The product is found defective within 14 days written notice to the Company;
- ② The product shall be paid by mail back to the company;
- ③ The product should be within the warranty period.

The Company is only responsible for those used in the occasion of the technical condition of the product defective product. Without any guarantee, warranty or written statement of its products used in special applications. Company for its products applied to the reliability of the product or circuit does not make any commitment.