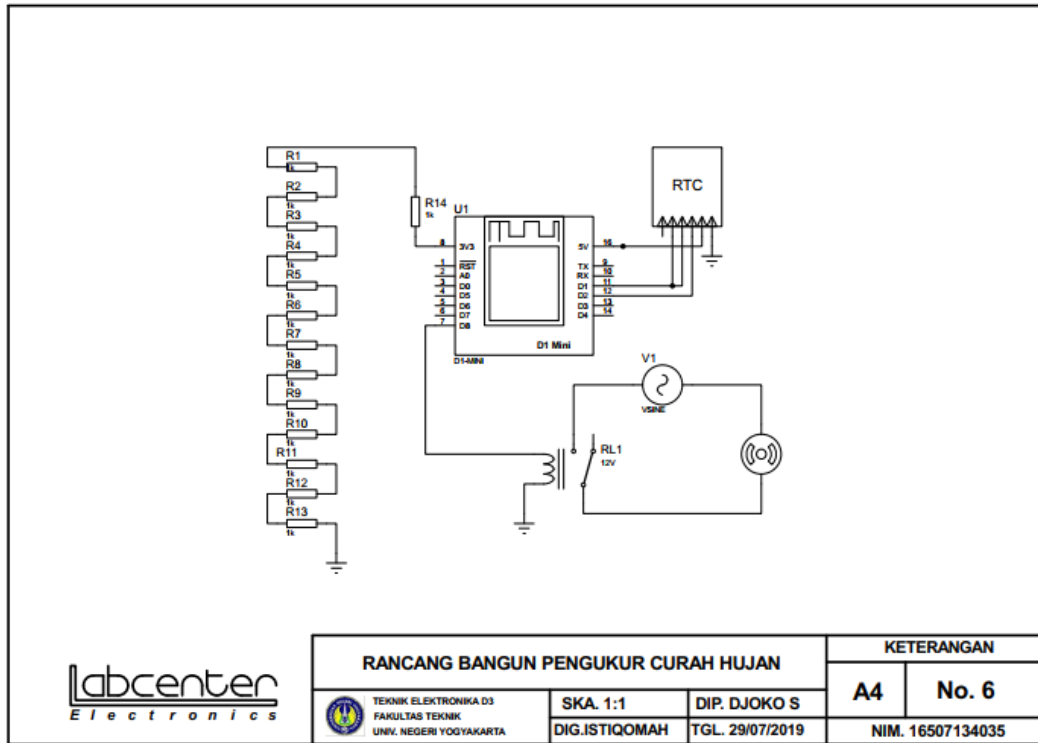
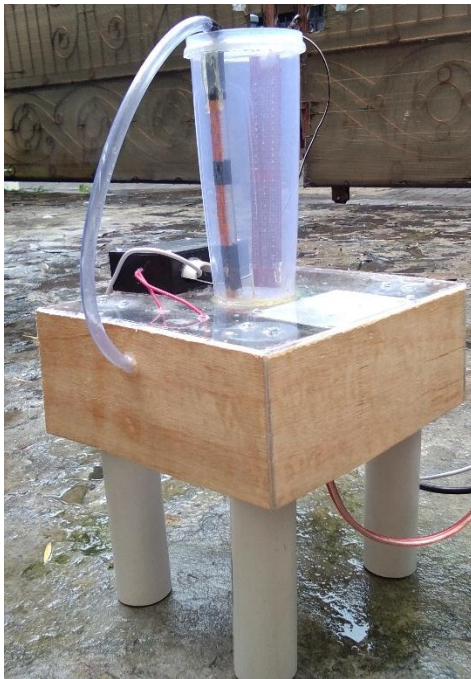


LAMPIRAN-LAMPIRAN

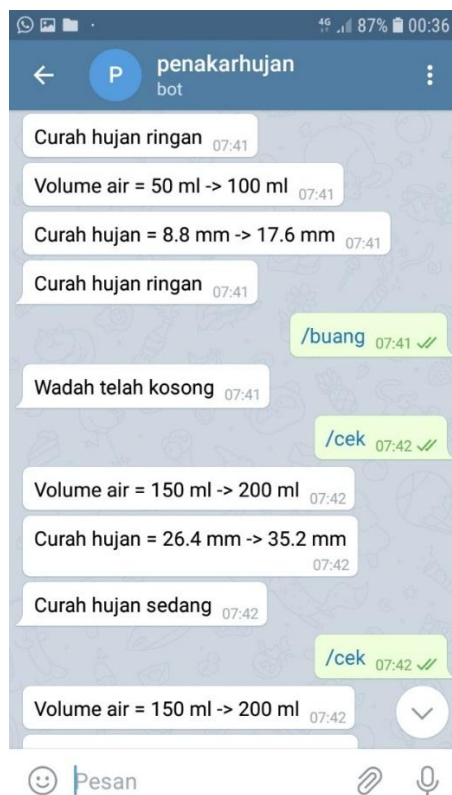
Lampiran 1. Rangkaian Skematik



Lampiran 2. Tampilan Alat



Lampiran 3. Tampilan pada Aplikasi Telegram



Lampiran 4. List Program Arduino IDE

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <ESP8266TelegramBOT.h>
#include <RTClib.h>
RTC_DS3231 rtc;

char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday"};

const int pinAir = A0; // pin sensor
int pinRelay = D7; // relay pada pin D7
int data;
int counter;
int i;
const char* ssid = "AA";
const char* password = "isti12345";

WiFiServer server(80);
#define BOTtoken "823305887:AAHEzWb13tiea6RgmzI8BOmZ3onjCdvYePA"
//token of FlashledBOT
#define BOTname "penakarhujan"
#define BOTusername "penakarhujan_bot"
e; //last time messages' scan has been done
bool Start = false;
/*****
* EchoMessages - function to Echo messages *
*****/
```

```
void curahHujan(){
    if(data < 10 ) {bot.sendMessage(bot.message[i][4], "Tidak ada hujan", "");}
    if(data >= 10 && data < 22) {
        bot.sendMessage(bot.message[i][4], "Volume air = 0 ml -> 50 ml", "");
        bot.sendMessage(bot.message[i][4], "Curah hujan = 0 mm -> 8.8 mm", "");
        bot.sendMessage(bot.message[i][4], "Curah hujan sangat ringan", "");}
    if(data >= 22 && data < 28) {
        bot.sendMessage(bot.message[i][4], "Volume air = 50 ml -> 100 ml", "");
        bot.sendMessage(bot.message[i][4], "Curah hujan = 8.8 mm -> 17.6 mm", "");
        bot.sendMessage(bot.message[i][4], "Curah hujan ringan", "");}
    if(data >= 28 && data < 31) {
        bot.sendMessage(bot.message[i][4], "Volume air = 100 ml -> 150 ml", "");
        bot.sendMessage(bot.message[i][4], "Curah hujan = 17.6 mm -> 26.4 mm", "");
        bot.sendMessage(bot.message[i][4], "Curah hujan ringan", "");}
    if(data >= 31 && data < 36) {
        bot.sendMessage(bot.message[i][4], "Volume air = 150 ml -> 200 ml", "");
        bot.sendMessage(bot.message[i][4], "Curah hujan = 26.4 mm -> 35.2 mm", "");
        bot.sendMessage(bot.message[i][4], "Curah hujan sedang", "");}
    if(data >= 36 && data < 40) {
        bot.sendMessage(bot.message[i][4], "Volume air = 200 ml -> 250 ml", "");
        bot.sendMessage(bot.message[i][4], "Curah hujan = 35.2 mm -> 44.0 mm", "");
        bot.sendMessage(bot.message[i][4], "Curah hujan sedang", "");}
    if(data >= 40 && data < 45) {
        bot.sendMessage(bot.message[i][4], "Volume air = 250 ml -> 300 ml", "");
        bot.sendMessage(bot.message[i][4], "Curah hujan = 44.0 mm -> 52.9 mm", "");
        bot.sendMessage(bot.message[i][4], "Curah hujan sedang", "");}
}
```

```
if(data >= 45 && data < 50) {  
    bot.sendMessage(bot.message[i][4], "Volume air = 300 ml -> 350 ml", "");  
    bot.sendMessage(bot.message[i][4], "Curah hujan = 52.9 mm -> 61.7 mm", "");  
    bot.sendMessage(bot.message[i][4], "Curah hujan lebat", "");}  
if(data >= 50 && data < 56) {  
    bot.sendMessage(bot.message[i][4], "Volume air = 350 ml -> 400 ml", "");  
    bot.sendMessage(bot.message[i][4], "Curah hujan = 61.7 mm -> 70.5 mm", "");  
    bot.sendMessage(bot.message[i][4], "Curah hujan lebat", "");}  
if(data >= 56 && data < 64) {  
    bot.sendMessage(bot.message[i][4], "Volume air = 400 ml -> 450 ml", "");  
    bot.sendMessage(bot.message[i][4], "Curah hujan = 70.5 mm -> 79.3 mm", "");  
    bot.sendMessage(bot.message[i][4], "Curah hujan lebat", "");}  
if(data >= 64 && data < 77) {  
    bot.sendMessage(bot.message[i][4], "Volume air = 450 ml -> 500 ml", "");  
    bot.sendMessage(bot.message[i][4], "Curah hujan = 79.3 mm -> 88.16 mm", "");  
    bot.sendMessage(bot.message[i][4], "Curah hujan lebat", "");}  
if(data >= 77 && data < 98) {  
    bot.sendMessage(bot.message[i][4], "Volume air = 500 ml -> 550 ml", "");  
    bot.sendMessage(bot.message[i][4], "Curah hujan = 88.16 mm -> 97.0 mm", "");  
    bot.sendMessage(bot.message[i][4], "Curah hujan lebat", "");}  
if(data >= 98 && data < 135) {  
    bot.sendMessage(bot.message[i][4], "Volume air = 550 ml -> 600 ml", "");  
    bot.sendMessage(bot.message[i][4], "Curah hujan = 97.0 mm -> 105.8 mm", "");  
    bot.sendMessage(bot.message[i][4], "Curah hujan sangat lebat", "");}
```

```

}
void Bot_ExecMessages() {
    data = analogRead(pinAir);
    Serial.println(data);

    for (i = 1; i < bot.message[0][0].toInt() + 1; i++){
        bot.message[i][5]=bot.message[i][5].substring(1,bot.message[i][5].length());
        if (bot.message[i][5] == "start") {
            String wellcome = "Selamat datang di SensorBMKGbot, bot sensor BMKG
            untuk mengukur curah hujan";

            String wellcome1 = "/id : to check id device";
            String wellcome2 = "/cek : status all sensor";
            String wellcome3 = "/buang : membuang sisa air";
            bot.sendMessage(bot.message[i][4], wellcome, "");
            bot.sendMessage(bot.message[i][4], wellcome1, "");
            bot.sendMessage(bot.message[i][4], wellcome2, "");
            bot.sendMessage(bot.message[i][4], wellcome3, "");

            Start = true;
        }
        if (bot.message[i][5] == "id") {
            bot.sendMessage(bot.message[i][4], "Istiqomah 19", "");
        }
        if (bot.message[i][5] == "cek"){
            curahHujan();
        }
    }
}

```

```

}
if(data >= 135) {
  counter++;
  if(counter == 1){
    bot.sendMessage(bot.message[i][4], "Volume air = 650 ml -> 700 ml", "");
    bot.sendMessage(bot.message[i][4], "Curah hujan = 114.6 mm -> 123.43 mm",
    "");
    bot.sendMessage(bot.message[i][4], "Curah hujan sangat lebat ", "");
    digitalWrite(pinRelay, LOW);
    delay(20000); //10detik
    digitalWrite(pinRelay, HIGH);
    //bot.sendMessage(bot.message[i][4], "Wadah telah kosong", "");
    bot.sendMessage(bot.message[i][4], "Pembuangan ke-1", "");
  }
  if(counter == 2){
    bot.sendMessage(bot.message[i][4], "Volume air = 1350 ml -> 1400 ml", "");
    bot.sendMessage(bot.message[i][4], "Curah hujan = 238 mm -> 246.87 mm ",
    "");
    bot.sendMessage(bot.message[i][4], "Curah hujan sangat lebat ", "");
    digitalWrite(pinRelay, LOW);
    delay(20000); //10detik
    digitalWrite(pinRelay, HIGH);
    //bot.sendMessage(bot.message[i][4], "Wadah telah kosong", "");
    bot.sendMessage(bot.message[i][4], "Pembuangan ke-2", "");
  }
  if(counter == 3){
    bot.sendMessage(bot.message[i][4], "Volume air = 2050 ml -> 2100 ml", "");
    bot.sendMessage(bot.message[i][4], "Curah hujan = 261.4 mm -> 370.3 mm ",
    "");
    bot.sendMessage(bot.message[i][4], "Curah hujan sangat lebat ", "");
  }
}

```

```
digitalWrite(pinRelay, LOW);
    delay(20000); //10detik
    digitalWrite(pinRelay, HIGH);
    //bot.sendMessage(bot.message[i][4], "Wadah telah kosong", "");
    bot.sendMessage(bot.message[i][4], "Pembuangan ke-3", "");
}
if(counter == 4){
    bot.sendMessage(bot.message[i][4], "Volume air = 2750 ml -> 2800 ml", "");
    bot.sendMessage(bot.message[i][4], "Curah hujan = 484.9 mm -> 493.74 mm ",
    "");
    bot.sendMessage(bot.message[i][4], "Curah hujan sangat lebat ", "");
    digitalWrite(pinRelay, LOW);
    delay(20000); //10detik
    digitalWrite(pinRelay, HIGH);
    //bot.sendMessage(bot.message[i][4], "Wadah telah kosong", "");
    bot.sendMessage(bot.message[i][4], "Pembuangan ke-4", "");
}
if(counter == 5){
    bot.sendMessage(bot.message[i][4], "Volume air = 3450 ml-> 3500", "");
    bot.sendMessage(bot.message[i][4], "Curah hujan = 608.35 mm -> 617.17 mm ",
    "");
    bot.sendMessage(bot.message[i][4], "Curah hujan sangat lebat ", "");
    digitalWrite(pinRelay, LOW);
    delay(20000); //10detik
    digitalWrite(pinRelay, HIGH);
    //bot.sendMessage(bot.message[i][4], "Wadah telah kosong", "");
    bot.sendMessage(bot.message[i][4], "Pembuangan ke-5", "");
}
}
```

```

else if (bot.message[i][5] == "buang"){
    bot.sendMessage(bot.message[i][4], "Wadah telah kosong", "");
    digitalWrite(pinRelay, LOW);
    delay(20000); //20detik
    digitalWrite(pinRelay, HIGH);
    counter = 0;
}
}
bot.message[0][0] = ""; // All messages have been replied - reset new messages
}

void setup() {
    #ifndef ESP8266
        while (!Serial); // for Leonardo/Micro/Zero
    #endif
    Serial.begin(115200);
    delay(3000);
    // attempt to connect to Wifi network:
    Serial.print("Connecting Wifi: ");
    Serial.println(ssid);
    WiFi.begin(ssid,password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    if (! rtc.begin()) {
        Serial.println("Couldn't find RTC");
        while (1);
    }
}

```

```

if (rtc.lostPower() {
    Serial.println("RTC lost power, lets set the time!");
    // following line sets the RTC to the date & time this sketch was compiled
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    // This line sets the RTC with an explicit date & time, for example to set
    // January 21, 2014 at 3am you would call:
    // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
}
Serial.println("");
Serial.println("WiFi connected");
Serial.print("http://");
Serial.print(WiFi.localIP());
bot.begin();    // launch Bot functionalities

pinMode(pinAir, INPUT);
pinMode(pinRelay, OUTPUT);
digitalWrite(pinRelay, HIGH);
}
void loop() {
    if (millis() > Bot_lasttime + Bot_mtbs) {
        bot.getUpdates(bot.message[0][1]); // launch API GetUpdates up to xxx message
        Bot_ExecMessages(); // reply to message with Echo
        Bot_lasttime = millis();
    }
    DateTime now = rtc.now();
    if(now.hour() == 7 && now.minute() == 0 && now.second() == 0 ){
        curahHujan();
    }
}
}

```

Lampiran 5. Datasheet Wemos D1 Mini

What is WeMos?

WeMos is a company that develops low cost-effective Internet of Things (IoT) devices for various projects and products.

WeMos D1 Mini series are one of the products developed by the company to enable wireless connectivity, simple data traffic and electronic controlling to electronic projects at the same time.



What's special about WeMos D1 Mini ?



WeMos D1 Mini ESP8266 WiFi Development Board



WeMos D1 Mini PRO ESP8266 WiFi Development Board

- Very small-sized and its footprint is breadboard friendly.
- Built-in antenna on the ESP8266EX chip.
- Uses 802.11b/g/n wireless communication protocol, can be detected and communicated via wireless modem router once powered up.
- Many multilevel-stacking shields designed to fit onto the WeMos D1 Mini board application.
- Uses micro USB Port so you can use any data carrying cable to power up and program the WeMos D1 Mini.
- Arduino IDE compatible, you can use the same Arduino IDE to program this board (add-on extension required and available via Board Manager).
- Built-in USB serial communication module (CH340G) so you can use this board similar as using any other Arduino boards with PC.
- 11 Digital IO pins, all IO pins except D0 are capable of PWM, Interrupt, I2C and 1-wire interfaces. This board also accepts SPI interface.
- 1 Analog Input pin that can accept up to 3.3V signal.
- 4 MB Flash for WeMos D1 Mini, 16 MB Flash for PRO version.

1

MAIN BOARD

WEMOS D1 MINI ESP8266 WIFI DEVELOPMENT BOARD



WeMos D1 mini is a mini WiFi board based on ESP8266EX. This board is Arduino IDE compatible, therefore it can be programmed using Arduino or its own Lua compiler. It also supports both serial and OTA programming.

This small development board has 4MB Flash memory, based on the specifications of ESP8266EX, has 11 digital IO pins, all IO pins capable of Interrupt / PWM / I2C / 1-wire except D0 pins. It has 1 Analog input (3.3V max) and a Micro USB port to connect with PC for communication and programming.

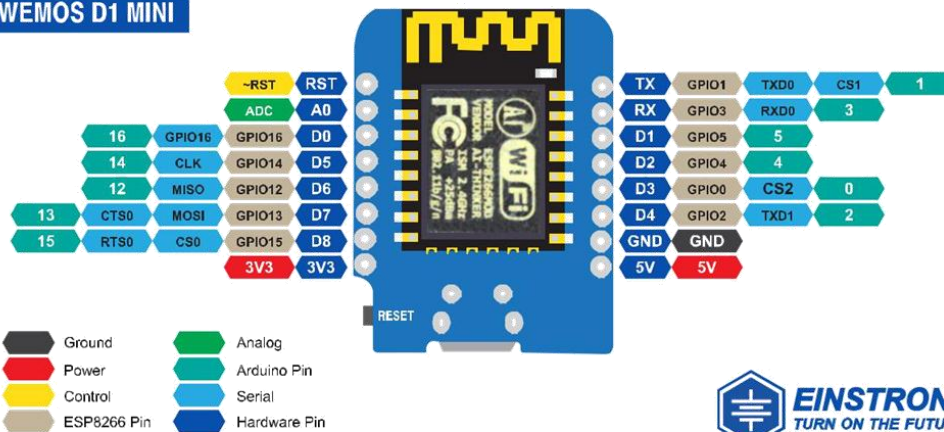
WEMOS D1 MINI PRO ESP8266 WIFI DEVELOPMENT BOARD

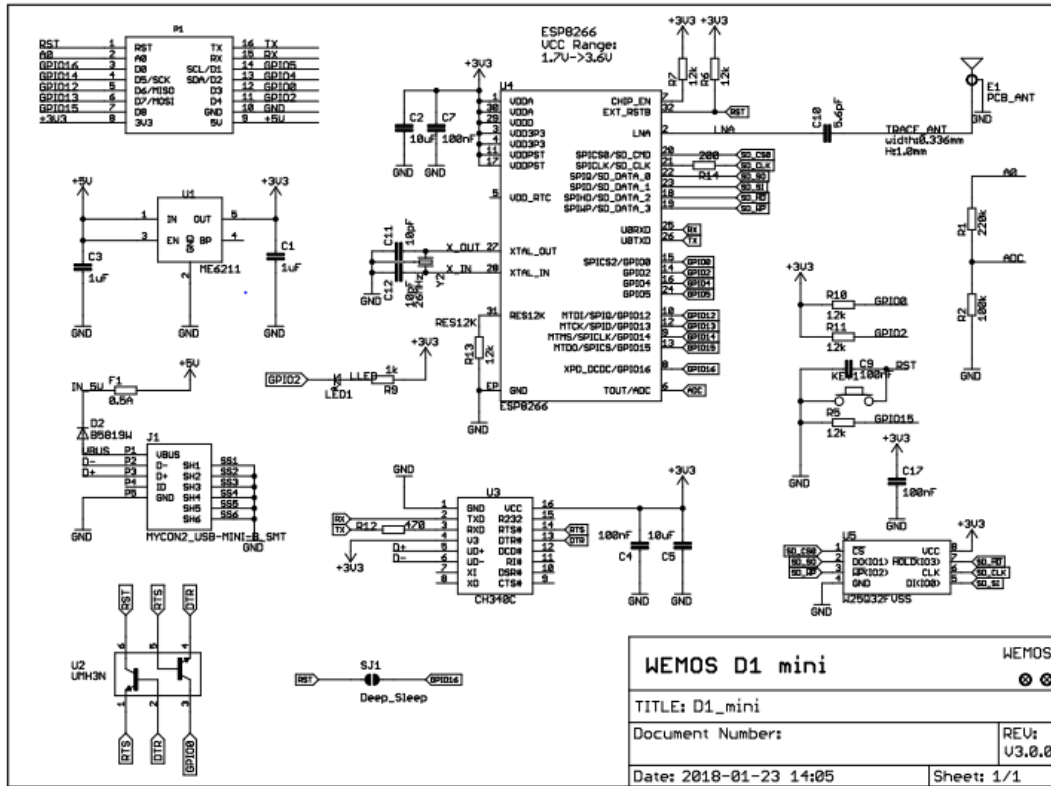
WeMos D1 Mini PRO differs from the previous version WeMos D1 Mini by using the new CP2104 USB-to-UART interface circuit. This board is lighter than the WeMos D1 Mini but shares the same footprint.

This board has 16MB Flash memory, external antenna connector and built-in ceramic antenna, has 11 digital IO pins, all pins capable of Interrupt / PWM / I2C / 1-wire except D0 pins. It has 1 Analog input (3.3V max) and a Micro USB port to connect with PC for programming.



PINOUT DIAGRAM WEMOS D1 MINI





Lampiran 6. Datasheet RTC DS3231

General Description

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

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

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

Applications

Servers Utility Power Meters
Telematics GPS

Pin Configuration appears at end of data sheet.

Features

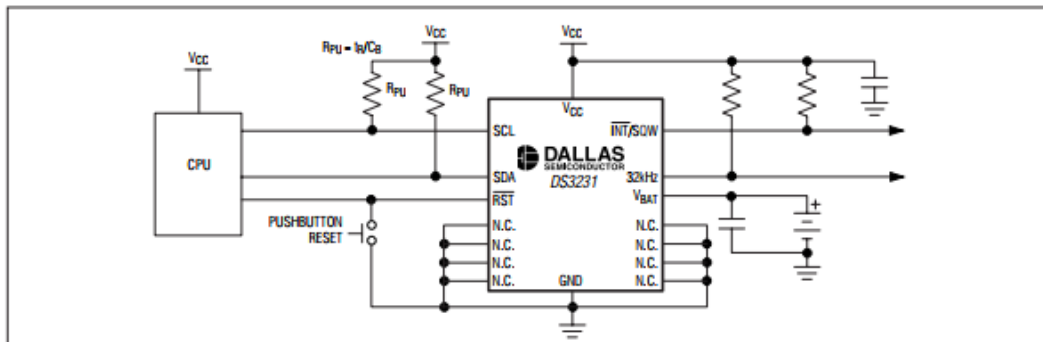
- ◆ Accuracy ±2ppm from 0°C to +40°C
- ◆ Accuracy ±3.5ppm from -40°C to +85°C
- ◆ Battery Backup Input for Continuous Timekeeping
- ◆ Operating Temperature Ranges
Commercial: 0°C to +70°C
Industrial: -40°C to +85°C
- ◆ Low-Power Consumption
- ◆ Real-Time Clock Counts Seconds, Minutes, Hours, Day, Date, Month, and Year with Leap Year Compensation Valid Up to 2100
- ◆ Two Time-of-Day Alarms
- ◆ Programmable Square-Wave Output
- ◆ Fast (400kHz) I²C Interface
- ◆ 3.3V Operation
- ◆ Digital Temp Sensor Output: ±3°C Accuracy
- ◆ Register for Aging Trim
- ◆ RST Input/Output
- ◆ UL Recognized

Ordering Information

| PART | TEMP RANGE | PIN-PACKAGE | TOP MARK |
|-----------|----------------|-------------|----------|
| DS3231S | 0°C to +70°C | 16 SO | DS3231 |
| DS3231SN | -40°C to +85°C | 16 SO | DS3231N |
| DS3231S+ | 0°C to +70°C | 16 SO | DS3231+ |
| DS3231SN+ | -40°C to +85°C | 16 SO | DS3231N+ |

+Denotes lead-free

Typical Operating Circuit

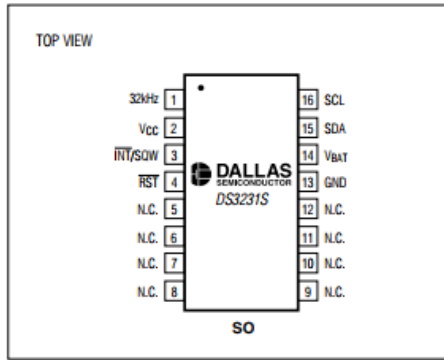


Purchase of I²C components from Maxim Integrated Products, Inc., or one of its sublicensed Associated Companies, conveys a license under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

Extremely Accurate I²C-Integrated RTC/TCXO/Crystal

DS3231

Pin Configuration



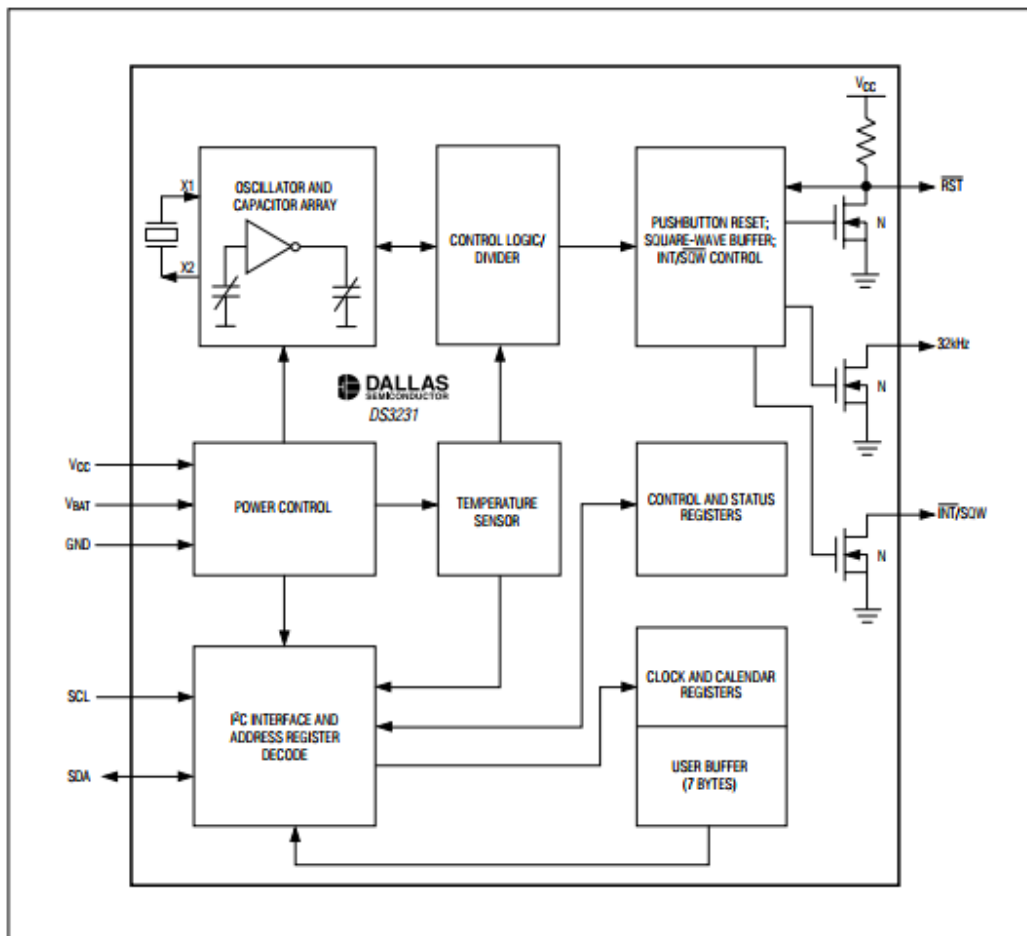
Chip Information

TRANSISTOR COUNT: 33,000
 SUBSTRATE CONNECTED TO GROUND
 PROCESS: CMOS

Thermal Information

Theta-JA: +73°C/W
 Theta-JC: +23°C/W

Block Diagram



Pin Description

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