

ARD 2 **Arduino Compatibles** Controllers, Shields, Modules & Sensors

DS3231 High-Precision RTC Module **ARD2-2093**

- **Track time even when module is unpowered**
- **Based on the DS3231 chip**
- **Leap year compensation up till 2099**
- **Programmable Square Wave Output**

Description

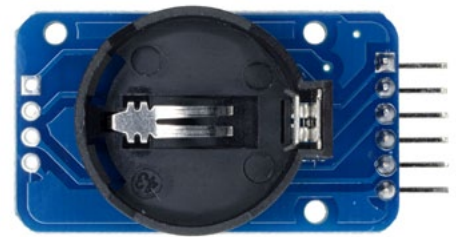
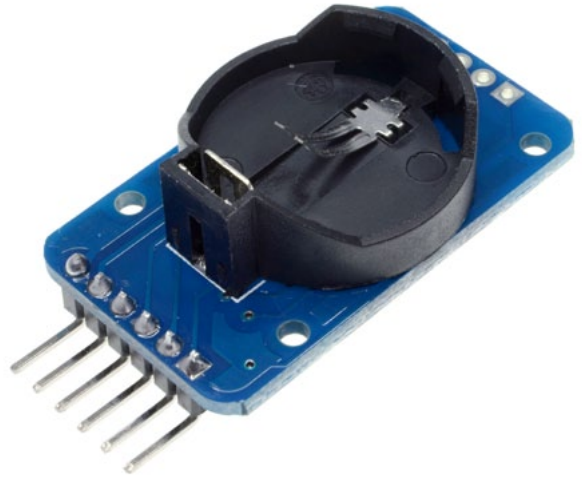
The DS3231 is a low-cost, extremely accurate I2C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal 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 data. 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 works in either the 24-hour or 12-hour format with an AM/PM indicator. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I2C bidirectional bus.

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

Specifications

Operating Voltage	5V
Main Chip	DS3231
Interface Type	Serial I/O interface
Dimensions	37mm x 22mm x 15mm
Operating Temperature	0°C – +70°C
TTL Compatible	V _{cc} =5V
Weight	55g



Pinout

Module	Arduino	Function
VCC	5V	Power Supply
GND	GND	Ground Connection
SCL	A5	Serial Clock
SDA	A4	Data I/O
SQW		Programmable Square Wave Output (optional)
32K		32kHz Output (optional)

Test Code

```
#include "Wire.h"
#define DS3231_I2C_ADDRESS 0x68
// Convert normal_decimal numbers to binary coded decimal
byte decToBcd(byte val)
{
    return( (val/10*16) + (val%10) );
}
// Convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
    return( (val/16*10) + (val%16) );
}
void setup()
{
    Wire.begin();
    Serial.begin(9600);
    // set the initial time here:
    // DS3231 seconds, minutes, hours, day, date, month, year
    // setDS3231time(30,42,21,4,26,11,14);
}
void setDS3231time(byte second, byte minute, byte hour, byte dayOfWeek, byte
dayOfMonth, byte month, byte year)
{
    // sets time and date data to DS3231
    Wire.beginTransmission(DS3231_I2C_ADDRESS);
    Wire.write(0); // set next input to start at the seconds register
    Wire.write(decToBcd(second)); // set seconds
    Wire.write(decToBcd(minute)); // set minutes
    Wire.write(decToBcd(hour)); // set hours
    Wire.write(decToBcd(dayOfWeek)); // set day of week (1=Sunday, 7=Saturday)
    Wire.write(decToBcd(dayOfMonth)); // set date (1 to 31)
    Wire.write(decToBcd(month)); // set month
    Wire.write(decToBcd(year)); // set year (0 to 99)
    Wire.endTransmission();
}
void readDS3231time(byte *second,
byte *minute,
byte *hour,
byte *dayOfWeek,
byte *dayOfMonth,
byte *month,
byte *year)
{
    Wire.beginTransmission(DS3231_I2C_ADDRESS);
    Wire.write(0); // set DS3231 register pointer to 00h
    Wire.endTransmission();
    Wire.requestFrom(DS3231_I2C_ADDRESS, 7);
    // request seven bytes of data from DS3231 starting from register 00h
```

Test Code

```
*second = bcdToDec(Wire.read() & 0x7f);
*minute = bcdToDec(Wire.read());
*hour = bcdToDec(Wire.read() & 0x3f);
*dayOfWeek = bcdToDec(Wire.read());
*dayOfMonth = bcdToDec(Wire.read());
*month = bcdToDec(Wire.read());
*year = bcdToDec(Wire.read());
}
void displayTime()
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  // retrieve data from DS3231
  readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month,
  &year);
  // send it to the serial monitor
  Serial.print(hour, DEC);
  // convert the byte variable to a decimal number when displayed
  Serial.print(":");
  if (minute<10)
  {
    Serial.print("0");
  }
  Serial.print(minute, DEC);
  Serial.print(":");
  if (second<10)
  {
    Serial.print("0");
  }
  Serial.print(second, DEC);
  Serial.print(" ");
  Serial.print(dayOfMonth, DEC);
  Serial.print("/");
  Serial.print(month, DEC);
  Serial.print("/");
  Serial.print(year, DEC);
  Serial.print(" Day of week: ");
  switch(dayOfWeek){
  case 1:
    Serial.println("Sunday");
    break;
  case 2:
    Serial.println("Monday");
    break;
  case 3:
    Serial.println("Tuesday");
    break;
  case 4:
    Serial.println("Wednesday");
    break;
  case 5:
    Serial.println("Thursday");
    break;
  case 6:
    Serial.println("Friday");
    break;
  case 7:
    Serial.println("Saturday");
    break;
  }
}
void loop()
{
  displayTime(); // display the real-time clock data on the Serial Monitor,
  delay(1000); // every second
}
```