



Documentação do módulo RTC DS3231

Autor: Saulo de Tarsio Silva Sousa

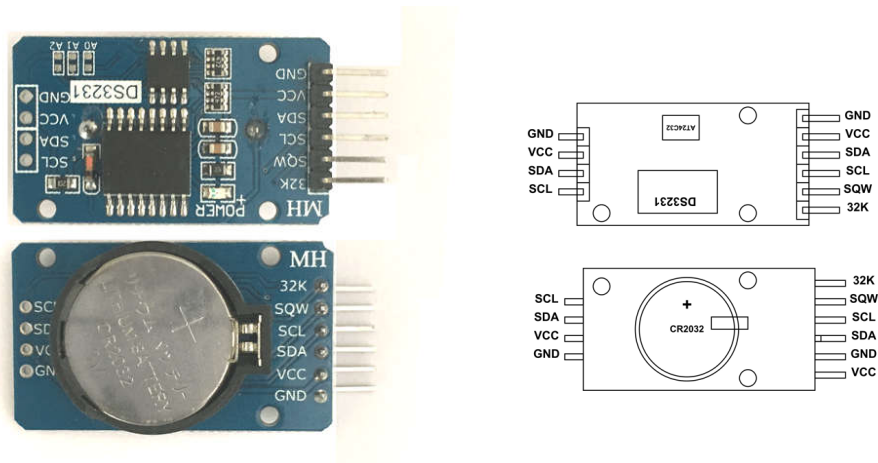
31 de Dezembro de 2019

Cabo Frio, RJ

O módulo RTC DS3231

O RTC DS3231 é um módulo capaz de medir e manter as informações de data e hora de determinado processo. A bateria conectada ao corpo do módulo mantém a energia necessária para que a data e hora não sejam perdidas em caso de falha do módulo ou se a energia de alimentação for cessada. As figuras abaixo mostram como é o módulo fisicamente, bem como os pinos que ele contém.

Figura 1: Módulo RTC DS3231 e seus pinos



Especificações e características do módulo

- Controlador: DS3231;
- Tensão de operação: 3,3V – 5VDC;
- Interface: I2C;
- Computa horas / minutos / segundos / dias da semana / mês / meses / anos (de 2000 a 2099);
- Consome menos de 500nA no modo bateria com oscilador em funcionamento;
- Memória: AT24C32 (capacidade de 32Kb que podem ser usados também como RAM estendida ao microcontrolador);
- Possui circuito de detecção de falha de energia;
- Interface: I2C;
- Bateria compatível: CR2032;
- Sensor de temperatura: ~ 3° celsius de exatidão;
- Temperatura de operação: 0 a 40° celsius.

O Módulo RTC DS3231 e os pinos utilizados

Os pinos do módulo que serão utilizados são:

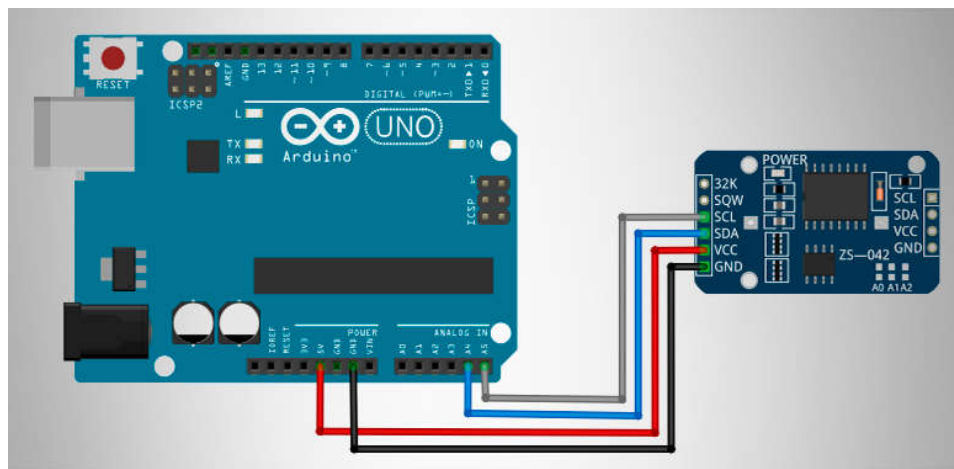
VCC: alimentação positiva com mínimo de 2.3V e máximo de 5.5V;

GND: pino terra do sistema;

SDA (Serial Data): será ligado ao pino SDA do Arduino Uno (A4). Permite a troca de dados entre o mestre e escravo de forma não simultânea;

SCL (Serial Clock Line): sinal de clock gerado pelo mestre para sincronismo de comunicação. Será ligado ao pino SCL do Arduino Uno (A5)

Figura 2: Conexão do módulo RTC DS3231 com a placa Arduino



A biblioteca RTCLib

Existem duas abordagens para se trabalhar com o módulo DS3231: a primeira é utilizando o seu datasheet (<https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>) e criarmos nosso próprio conjunto de funções que manipulam e retornam as informações de tempo que são necessárias ao sistema; a outra abordagem se trata de utilizar uma biblioteca, que é um conjunto de funções pertencentes a terceiros e que manipulam as informações de tempo necessárias. Iremos utilizar a segunda abordagem e para isso,

devemos baixar o conjunto de funções e importa-las à nossa ferramenta de desenvolvimento Arduino Software.

Link: <https://blogmasterwalkershop.com.br/arquivos/libs/RTClib.zip>

Imprimindo a hora e minuto no console do ambiente de desenvolvimento Arduino

O código abaixo mostra como capturar o valor de hora e minuto do módulo RTC DS3231. De posse desses valores, um pequeno texto, com o seguinte formato “HH:MM”, irá ser formado para ser impresso no console da ferramenta de desenvolvimento.

Figura 3: Código base para uso do módulo RTC DS3231

```
ds3231_sketch
#include "RTClib.h" //Importar a biblioteca para a aplicação
RTC_DS3231 rtc;    //Objeto que contém as funções de manipulação das informações

void setup () {
  Serial.begin(9600); //Inicializar comunicação serial
  delay(3000);       //Espere 3 segundos
  /**
   * Se o módulo não inicializou corretamente, uma mensagem é enviada
   * ao console e a o arduino fica preso em um loop infinito.
   */
  if (! rtc.begin()) {
    Serial.println("Módulo RTC DS3231 não pode ser encontrado!");
    while (1);
  }

  //Caso o módulo desligue, há o ajuste da data
  if (rtc.lostPower()) {
    //Ajuste de data
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
}

void loop () {
  DateTime now = rtc.now(); //Uma instância do momento atual
  if(now.hour() < 10){
    Serial.print("0");
  }
  Serial.print(now.hour(), DEC); //Imprime a hora
  Serial.print(":");

  if(now.minute() < 10){
    Serial.print("0");
  }
  Serial.println(now.minute(), DEC); //Imprime o minuto
  delay(1000); //Espere 1 segundo
}
```

Passos

1. Importe a biblioteca e declare um objeto do tipo RTCLib para começar a manipulação do módulo:

```
#include "RTCLib.h"    //Importar a biblioteca para a aplicação
RTC_DS3231 rtc;       //Objeto que contém as funções de manipulação das informações
```

2. Dentro do método *setup*, inicie o canal de comunicação via *Serial* com o valor 9600 para 'baud rate' e espere 3 segundos:

```
Serial.begin(9600); //Inicializar comunicação serial
delay(3000);        //Espere 3 segundos
```

3. Ainda dentro do método *setup*, inicie o módulo RTC DS3231. A condição abaixo testa se o módulo inicializou corretamente. Se caso haja falhas na inicialização do módulo, uma mensagem de erro é enviada ao console e o Arduino ficará preso em um loop infinito.

```
/**
 * Se o módulo não inicializou corretamente, uma mensagem é enviada
 * ao console e a o arduino fica preso em um loop infinito.
 */
if (! rtc.begin()) {
    Serial.println("Módulo RTC DS3231 não pode ser encontrado!");
    while (1);
}
```

4. Ainda dentro do método *setup*, adicione a funcionalidade de ajuste “automático” de data e hora do sistema caso o módulo falhe e venha a desligar.

```
//Caso o módulo desligue, há o ajuste da data
if (rtc.lostPower()) {
    //Ajuste de data
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}
```

5. Dentro do método *loop*, crie uma instância das informações capturadas do módulo RTC DS3231. Logo após, iremos acessar alguns parâmetros do objeto

instanciado, como a hora e minuto medidos. Os *prints* presentes no código apenas imprimem a mensagem da seguinte forma:

HH:MM

Por exemplo: 13:22, 14:55, 22:45, 02:02, 05:09, ...

Espere por 1 segundo até requisitar novamente as informações de hora do módulo.

```
DateTime now = rtc.now(); //Uma instância do momento atual
if(now.hour() < 10){
    Serial.print("0");
}
Serial.print(now.hour(), DEC); //Imprime a hora
Serial.print(":");

if(now.minute() < 10){
    Serial.print("0");
}
Serial.println(now.minute(), DEC); //Imprime o minuto
delay(1000); //Espere 1 segundo
```

6. Carregue o sketch no Arduino com o módulo RTC DS3231 devidamente conectado. Abra o console do Arduino Software. A mensagem do tipo “HH:MM” estará sendo impressa no console.

Observações:

- A biblioteca RTCLib contém muitas funções que fazem a manipulação das informações de data e hora e isso facilita o desenvolvimento das aplicações. Abaixo serão listadas algumas dessas funções e pode-se listar o objetivo de cada uma.

- `rtc.hour()` -> para obter a hora atual
- `rtc.minute()` -> para obter minuto atual
- `rtc.second()` -> para obter o segundo atual
- `rtc.year()` -> para obter o ano atual
- `rtc.day()` -> para obter o dia atual
- `rtc.month()` -> para obter o mês atual

Essas são só algumas das mais diversas funções que a biblioteca RTCLib apresenta. Para ver mais sobre as funções que estão contidas na biblioteca, acesse a documentação no seguinte endereço:

Link: <https://github.com/adafruit/RTCLib>

- O módulo RTC DS3231 permite que acoplemos sensores de temperaturas a ele e assim recuperar o valor medido por meio da biblioteca RTCLib. Como o protótipo será construído com o sensor de temperatura DHT11/DHT22 separadamente, não veremos as funções que manipulam a temperatura.

Conclusões

O módulo RTC DS3231 se mostra uma excelente abordagem para dar a uma aplicação a capacidade de medir e manter a informação de tempo. Foi visto que trata-se de um módulo que utiliza I2C como protocolo de comunicação. Para isso, surge a necessidade de utilizar os pinos com as funções SCL e SDA do Arduino e RTC DS3231.

Após realizar toda a montagem elétrica, partiu-se para a programação do módulo. Para isso, foi necessário a utilização de uma biblioteca que apresenta funções que retornam as informações de data e hora requisitadas. A biblioteca RTCLib foi devidamente importada à ferramenta de desenvolvimento Arduino Software e um código foi feito para imprimir um texto com o seguinte formato: “HH:MM”.

Como o protótipo visa recuperar apenas o valor da hora e minuto atuais, as outras funções da biblioteca não serão analisadas neste escopo. Um estudo mais aprofundado sobre a biblioteca deve ser feito acessando o link: <https://github.com/adafruit/RTCLib>.