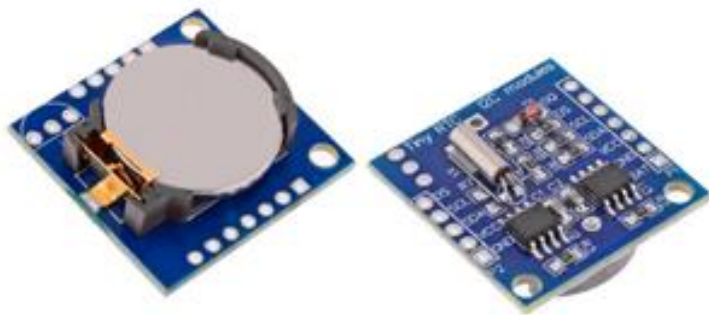


# Como usar o RTC DS1307



DS1307

Por Eng. Roberto Bairros dos Santos

[www.bairrospd.com](http://www.bairrospd.com)

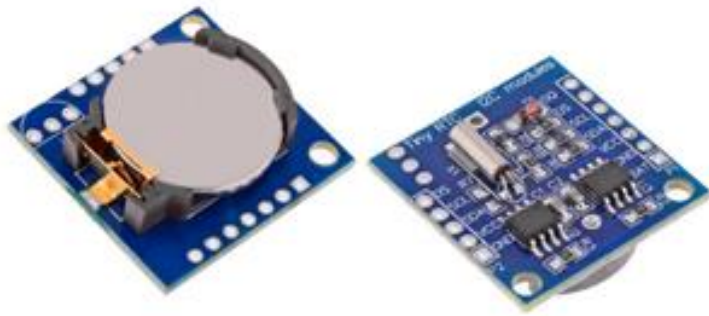
Data: 02/12/2016

## Sumário

Introdução.....	3
O que é um RTC? .....	4
Como é feita a comunicação com o Arduino.....	8
Como ligar no Arduino UNO.....	9
Gerador de onda quadrada.....	10
A biblioteca.....	12
Como usar as funções da biblioteca do RTC. ....	13
Funções “setDOW, setTime e setDate” .....	14
Funções “setQWRate e enableSQW” .....	15
Funções para pegar o tempo e data no formato string.....	16
“getTimeStr” .....	17
“getDateStr” .....	18
“getDOWStr” .....	19
“getMonthStr” .....	20
Variáveis para pegar o tempo e data no formato numérico DECIMAL.....	21
“hour” .....	22
“min” .....	22
“sec” .....	22
“year” .....	22
“mon” .....	22
“dow” .....	22
Programas exemplos:.....	23
Programa exemplo 0.....	24
O programa 1: .....	26
O programa 2: .....	28
Programas exemplos para copiar e colar:.....	30
Exemplo do uso do tempo como variável numérica.....	33
Conclusão: .....	35

## Introdução.

Este tutorial mostra como usar o RTC DS1307 com o Arduino apresentando as suas características e exemplos de aplicações.



**DS1307**

## O que é um RTC?

RTC significa Relógio em Tempo Real.



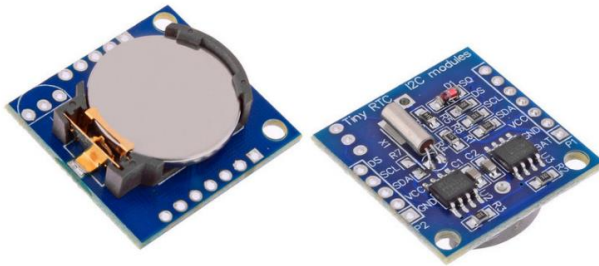
Com um RTC o seu equipamento ficará sabendo a qualquer momento a hora e a data, mesmo depois de você o desligar.

# BATERIA



O RTC é na verdade um circuito integrado com um processador dedicado para contar o tempo, no comercio existem shields onde o circuito integrado já vem montado com todos os periféricos incluindo uma bateria que alimenta o módulo quando a tensão de alimentação é desligada.

**RTC**



**DS1307**



**DS3231**

Um RTC é útil porque todo o processamento de tempo é feito em um componente fora do Arduino simplificando a programação e garantindo a precisão.



**ARDUINO**

```

Sketch
  Blink
  // Turns on an LED on for one second, then off for one second, repeatedly.
  // Most Arduinos have an on-board LED you can control. On the Uno and
  // Leonardo, it is connected to digital pin 13. If you're unsure what
  // pin the on-board LED is connected to on your Arduino model, check
  // the documentation at http://arduino.cc
  // This example code is in the public domain.
  // modified 8 May 2014
  // by Scott Fitzgerald
  //
  // the setup function runs once when you press reset or power the board
  void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
  }
  // the loop function runs over and over again forever
  void loop() {
    digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000);           // wait for a second
    digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
    delay(1000);           // wait for a second
  }
  
```

**Sem RTC**



**ARDUINO**

**RTC**



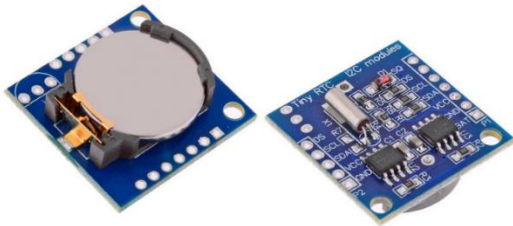
**Com RTC**

O RTC descreve o tempo em horas, minutos, segundo; a data em dia, mês e ano e os dias da semana em inglês (MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SUNDAY, SATURDAY).



O módulo mais popular são o DS1307 e o DS3231.

O DS1307 é mais simples não possui o sensor para compensar o erro devido a variação da temperatura, mas vem com a possibilidade de montar o sensor DS18B20.



**DS1307**

O módulo DS3231 já possui o sensor de temperatura instalado e possui uma precisão muito maior.



**DS3231**

Este tutorial mostrará com usar o módulo DS1307, no entanto a biblioteca também funciona no DS3231 você só não terá a informação de temperatura!



**DS1307**



**DS3231**

### Como é feita a comunicação com o Arduino.

A comunicação é feita usando o barramento I2C que é muito simples.

A comunicação I2C usa dois pinos um chamado SDA (Dados) e SCL(Clock).

Você deve verificar a compatibilidade da placa e os pinos a serem usados.

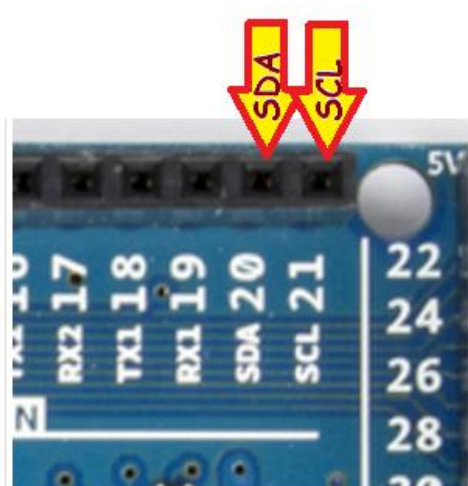
No Arduino UNO você pode usar quaisquer pinos, no exemplo deste tutorial foram usados os pinos A4 e A5.

O fornecedor da biblioteca garante os pinos deste tutorial, outros pinos não foram testados!

Você também pode usar os pinos dedicados a comunicação I2C do Arduino não esquecendo que na placa UNO você deve ajustar os jumpers JP6 e JP8 junto aos pinos analógicos!



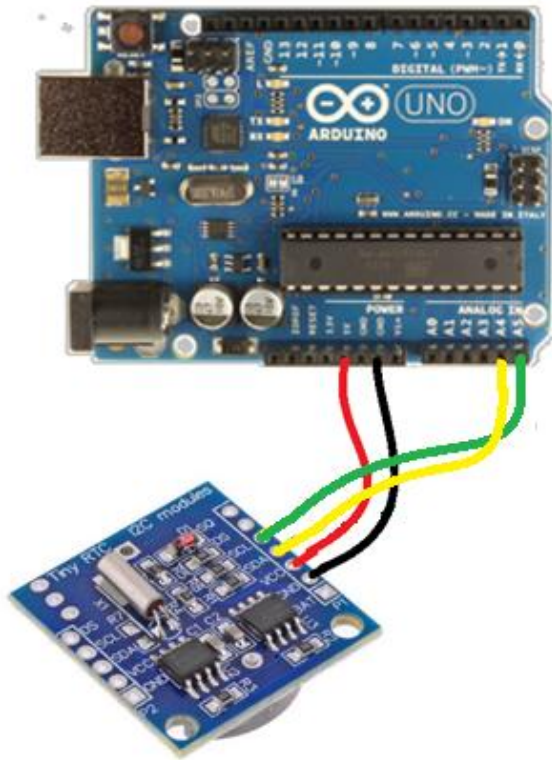
Já para a placa Arduino Mega os pinos são D20 e D21 que já são os pinos dedicados a programação I2C.





Como ligar no Arduino UNO.

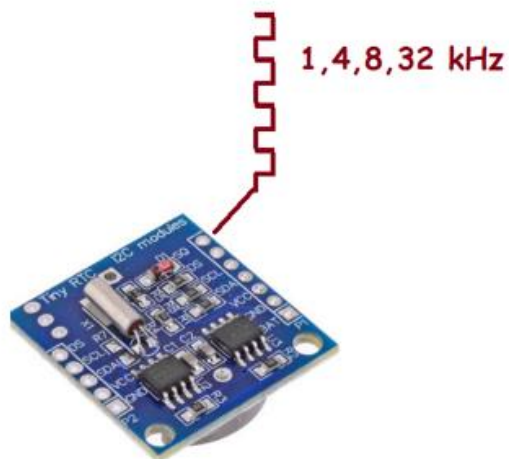
A ligação do módulo RTC DS1307 é simples como mostra a figura, esta é a ligação parametrizada nos programas exemplos!



```
DS1307 rtc(A4, A5);  
           ↑   ↑  
           (SDA, SCL)
```

## Gerador de onda quadrada.

O módulo DS1307 possui um pino de saída chamado de SQ que fornece uma onda quadrada de frequência programável (1kHz, 4kHz, 8kHz e 32kHz).

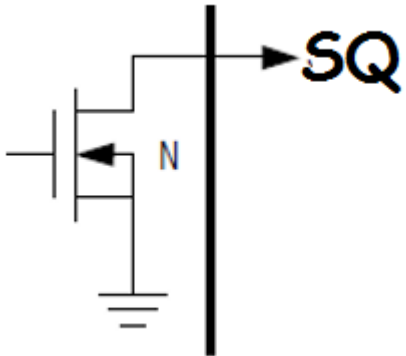


DS1307

A biblioteca possui uma instrução para habilitar o pino gerador de onda quadrada e programar a frequência.

```
//Definicoes do pino SQW/Out|
rtc.setSQWRate(SQW_RATE_1);
rtc.enableSQW(true);
```

A saída deste pino é do tipo DRENO aberto e você deve deixar desligada caso não for usar.



Você pode usar a onda quadrada para gerar um pulso de tempo de precisão, ligando este pino a um pino de interrupção da placa Arduino você pode contar o tempo criando um relógio interno sem usar a comunicação I2C.

O pino de Interrupção para cada placa é mostrado abaixo.

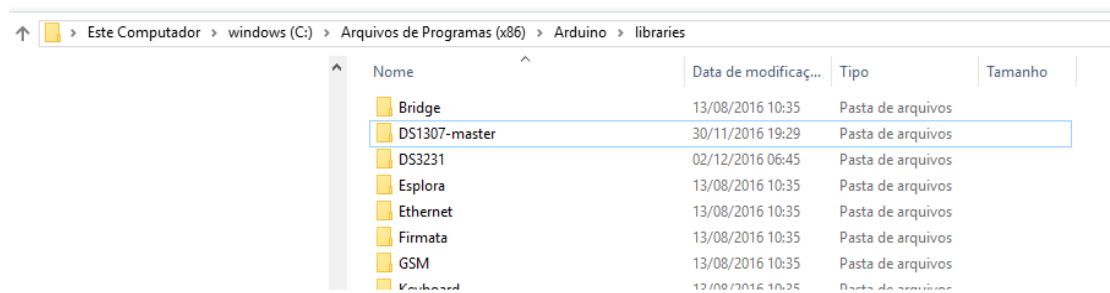
Board	int.0	int.1	int.2	int.3	int.4	int.5
Uno, Ethernet	2	3				
Mega2560	2	3	21	20	19	18
32u4 based (e.g Leonardo, Micro)	3	2	0	1	7	
Due, Zero			(see below)			

## A biblioteca.

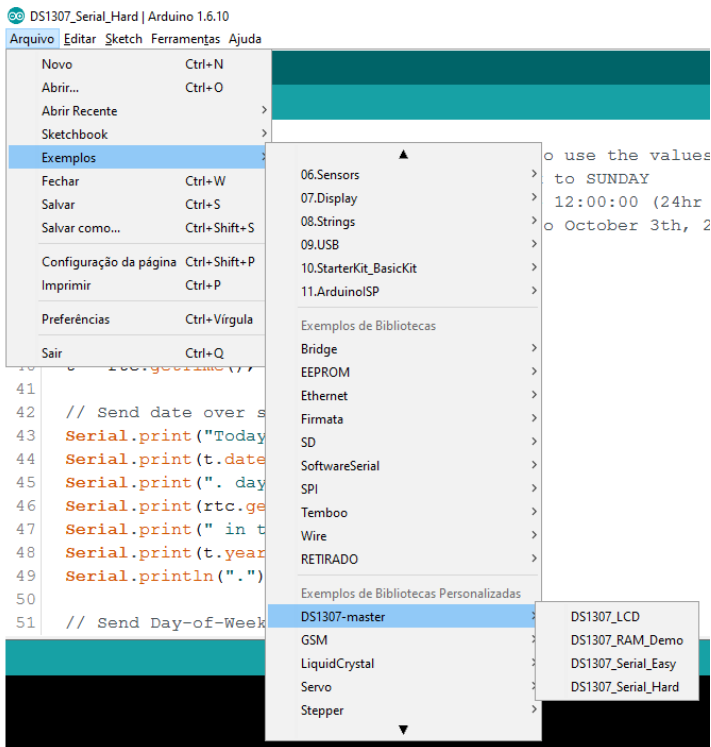
Para usar o RTC você deve baixar a biblioteca RTC DS1307 no link a seguir. Esta biblioteca também poderá ser usada com o RTC DS3231 mas não inclui a medição de temperatura.

<https://github.com/filipeflop/DS1307>

Descompacte o arquivo e copie para a LIBRARIES da IDE do seu Arduino.



A biblioteca possui exemplos que você pode baixar para aprender a usar este módulo, este tutorial é baseado nestes exemplos!



## Como usar as funções da biblioteca do RTC.

Primeiro você deve adicionar a biblioteca ao seu programa usando a instrução include.

```
//Carrega a biblioteca do RTC DS1307  
#include <DS1307.h>
```

Antes de usar as funções da biblioteca você deve configurar o módulo.

Você deve dizer ao compilador onde estará conectado os sinais I2C.

```
//Modulo RTC DS1307 ligado as portas A4 e A5 do Arduino  
DS1307 rtc(A4, A5);  
          ↑   ↑  
        (SDA , SCL )
```

A função "halt" com o parâmetro "false" não desliga o módulo!

```
//Aciona o relógio  
rtc.halt(false);
```

## Funções “setDOW, setTime e setDate” .

A funções “setDOW, setTime e setDate” devem ser usadas para ajustar o tempo e data do módulo RTC e devem ser colocadas dentro do setup().

Você deve usar as instruções de ajuste do tempo e data somente uma vez, depois a bateria interna irá manter os dados atualizados.

As instruções para atualizar os dados são mostradas a seguir.

```
//As linhas abaixo ajustam a data e hora do modulo
//e podem ser comentada apos a primeira utilizacao
//Define o dia da semana
//MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY SUNDAY
rtc.setDOW(THURSDAY);
rtc.setTime(19, 37, 0); //Define o horario (h,m,s)
rtc.setDate(30, 11, 2016); //Define o (dia, mes e ano)
```

Depois de você escrever os dados no módulo, você deve comentar estas linhas.

```
//As linhas abaixo ajustam a data e hora do modulo
//e podem ser comentada apos a primeira utilizacao
//Define o dia da semana
//MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY SUNDAY
//rtc.setDOW(THURSDAY);
//rtc.setTime(19, 37, 0); //Define o horario (h,m,s)
//rtc.setDate(30, 11, 2016); //Define o (dia, mes e ano)
//Definicoes do pino SQW/Out
```

## Funções “setSQWRate e enableSQW”.

As funções “**sesSQWRate** e **enableSQW**” servem para definir a frequência do pino gerador de onda quadrada e habilita este pino.

Estas linhas foram colocadas no programa exemplo, mas serão úteis apenas naqueles projetos onde forem usados este pino.

```
//Definicoes do pino SQW/Out
rtc.setSQWRate(SQW_RATE_1);
rtc.enableSQW(true);
```

Os parâmetros para a programação da frequência são mostrados a seguir.

setSQWRate(rate);	
Set the Square Wave output rate.	
Parameters:	rate: SQW_RATE_1 sets a 1Hz rate SQW_RATE_4K sets a 4.096KHz rate SQW_RATE_8K sets a 8.192KHz rate SQW_RATE_32K sets a 32.768KHz rate
Returns:	Nothing
Usage:	rtc.setSQWRate(SQW_RATE_1); // Sets the rate for SQW to 1 Hz

## Funções para pegar o tempo e data no formato string.

As instruções para pegar o tempo e a hora podem ser usadas fora do setup().

Você pode pegar o tempo e a data no formato string que é útil para mostrar na serial ou display, ou no formato de número que é útil para você usar em cálculos de tempo para ligar ou desligar uma saída, cronômetros etc.

### Formato STRIG

```
Tempo : 19:37:00 Data : 30.11.16 Dia:Thursday
```

### Formato DEC

```
Hora : 19 Minutos : 37 Segundos : 0 Ano : 2016 Mes : 11 dia : 4
```

As instruções para pegar valores no formato STRING são mostradas a seguir.



“getTIMEStr”.

A função “getTIMEStr” pega o tempo.

```
getTimeStr([format]);
```

Parâmetros:

```
format: <Optional>  
        FORMAT_LONG  "hh:mm:ss"  (default)  
        FORMAT_SHORT "hh:mm"
```

Exemplo:

```
Serial.print(rtc.getTimeStr());
```

```
Tempo : 01:39:09
```

## “getDateStr”

A função “getDateStr” pega a data no formato.

```
getDateStr([sformat[, eformat[, divider]]]);
```

Parâmetros:

```
sformat: <Optional> *1
         FORMAT_LONG   Year with 4 digits (yyyy) (default)
         FORMAT_SHORT  Year with 2 digits (yy)
eformat: <Optional> *2
         FORMAT_LITTLEENDIAN "dd.mm.yyyy" (default)
         FORMAT_BIGENDIAN   "yyyy.mm.dd"
         FORMAT_MIDDLEENDIAN "mm.dd.yyyy"
divider: <Optional>
         Single character to use as divider. Default is '.'
```

Exemplo:

```
rtc.getDateStr(FORMAT_SHORT)
```

```
Data : 01.12.16
```

“getDOWStr”.

A função “getDOWStr” pega o dia da semana em inglês.

```
getDOWStr([format]);
```

Parâmetros:

```
format: <Optional>
        FORMAT_LONG  Day-of-the-week in English (default)
        FORMAT_SHORT Abbreviated Day-of-the-week in English (3 letters)
```

```
MONDAY: 1
TUESDAY: 2
WEDNESDAY: 3
THURSDAY: 4
FRIDAY: 5
SATURDAY: 6
SUNDAY: 7
```

Exemplo:

```
Serial.print(rtc.getDOWStr(FORMAT_SHORT));
```

```
Dia:Fri
```

`getMonthStr`.

A função `getMonthStr` pega o mês em inglês.

```
getMonthStr([format]);
```

Parâmetros;

```
format: <Optional>  
        FORMAT_LONG   Month in English (default)  
        FORMAT_SHORT  Abbreviated month in English (3 letters)
```

Exemplo:

```
Serial.print(rtc.getMonthStr(FORMAT_SHORT));
```

```
Mes:Dec
```

## Variáveis para pegar o tempo e data no formato numérico DECIMAL.

Se você quiser as informações no formato numérico decimal as instruções são descritas abaixo.

Uma variável numérica trabalha com valores que podem ser usados em cálculos e ainda em instrução do tipo print.

<code>setTime(hour, min, sec);</code>	
Set the time.	
Parameters:	hour: Hour to store in the DS1307 (0-23) min: Minute to store in the DS1307 (0-59) sec: Second to store in the DS1307 (0-59)
Returns:	Nothing
Usage:	<code>rtc.setTime(23, 59, 59); // Set the time to 23:59:59</code>
Notes:	Setting the time will clear the CH (Clock Halt) flag. See the datasheet for more information on the CH flag.

Antes de mais nada você deverá criar uma variável no formato "Time" como descrito a seguir.

O nome da variável pode ser qualquer um, neste caso foi usado o nome "t".

```
Time t;//inicializa o tempo
```

As variáveis são descritas a seguir.

**“hour”**.

Para pegar a hora use a variável **“hour”**.

Exemplo:

```
Serial.print(" Horas:");  
Serial.print(t.hour, DEC);
```

**“min”**.

Para pegar o minuto use a variável **“min”**.

Exemplo:

```
Serial.print(" Minutos:");  
Serial.print(t.min, DEC);
```

**“sec”**.

Para pegar o segundo use a variável **“sec”**.

Exemplo:

```
Serial.print(" Segundos:");  
Serial.println(t.sec, DEC);
```

**“year”**.

Para pegar o ano use a variável **“year”**.

Exemplo:

```
Serial.print(" Ano:");  
Serial.print(t.year, DEC);
```

**“mon”**.

Para pegar o mês use a variável **“mon”**.

Exemplo:

```
Serial.print(" Mes:");  
Serial.print(t.mon, DEC);
```

**“dow”**

Para pegar o dia da semana **“dow”**

Exemplo:

```
Serial.print("Dia:");  
Serial.print(t.date, DEC);
```

### Programas exemplos:

Foram criados três programas exemplos para mostrar como usar a biblioteca do RTC.

Os programas dos exemplos irão usar o monitor da serial para mostrar os valores lidos.

- O programa 0 mostra como atualizar a data e a hora.
- O Programa 1 mostra como pegar os dados no formato string.
- O programa 2 mostra como pegar os dados no formato numérico.

Os programas 1 e 2 não tem as linhas de atualização dos dados.

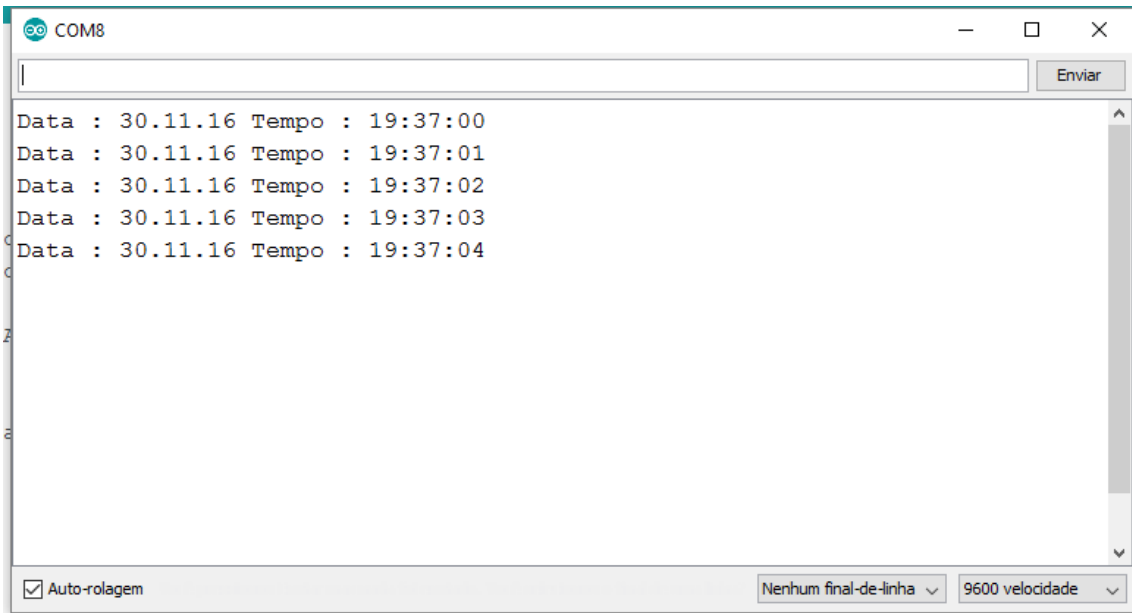
## Programa exemplo 0.

Você deverá usar este programa sempre que precisar atualizar os dados.

```
1 //Programa : Relogio com modulo RTC DS1307
2 //Autor : Professor Roberto Bairros
3 //CARREGA DATA E HORA NO RTC
4 //Carrega a biblioteca do RTC DS1307
5 #include <DS1307.h>
6 //Modulo RTC DS1307 ligado as portas A4 e A5 do Arduino
7 DS1307 rtc(A4, A5);
8
9 void setup() {
10     Serial.begin(9600); //inicia serial
11     //Aciona o relógio
12     rtc.halt(false);
13     //As linhas abaixo ajustam a data e hora do modulo
14     //e podem ser comentada apos a primeira utilizacao
15     //Define o dia da semana
16     //MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY SUNDAY
17     rtc.setDOW(THURSDAY);
18     rtc.setTime(19, 37, 0); //Define o horario (h,m,s)
19     rtc.setDate(30, 11, 2016); //Define o (dia, mes e ano)
20     //Definicoes do pino SQW/Out
21     rtc.setSQWRate(SQW_RATE_1);
22     rtc.enableSQW(true);
23
24 }
25
26 void loop() {
27     //Mostra as informações no Serial Monitor
28     Serial.print("Data : ");
29     Serial.print(rtc.getDateStr(FORMAT_SHORT));
30     Serial.print(" Tempo : ");
31     Serial.println(rtc.getTimeStr());
32     delay (1000);
33 }
```



Resultado:

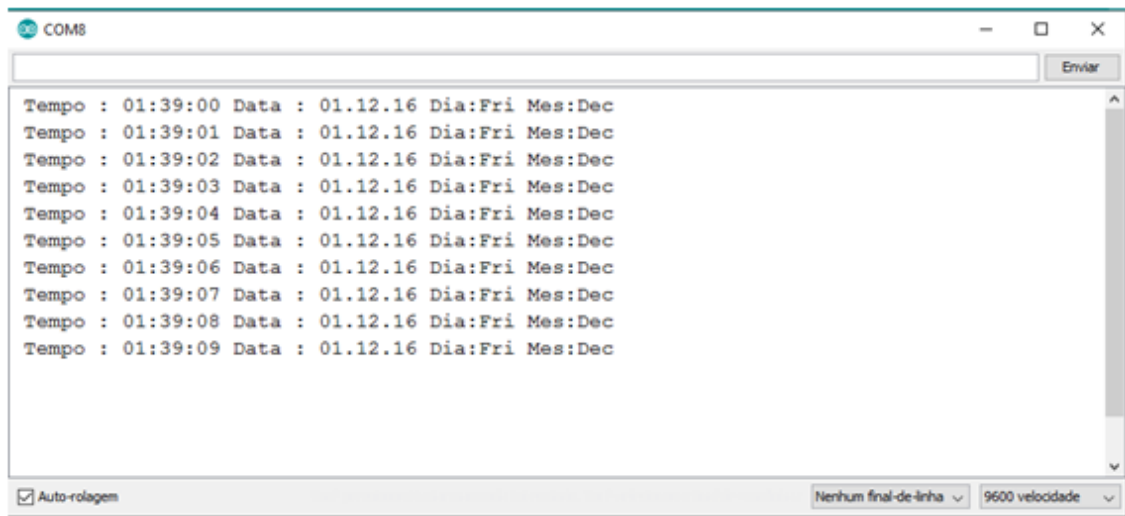


## O programa 1:

Este exemplo pega os dados no formato string.

```
1 //Programa : Relogio com modulo RTC DS1307
2 //Autor : Professor Roberto Bairros
3 //pega string
4 //Carrega a biblioteca do RTC DS1307
5 #include <DS1307.h>
6 //Modulo RTC DS1307 ligado as portas A4 e A5 do Arduino
7 DS1307 rtc(A4, A5);
8
9 void setup() {
10     Serial.begin(9600); //inicia serial
11     //Aciona o relógio
12     rtc.halt(false);
13     //VOCÊ DEVE AJUSTAR A DATA E O TEMPO PREVIAMENTE
14     //Definicoes do pino SQW/Out
15     rtc.setSQWRate(SQW_RATE_1);
16     rtc.enableSQW(true);
17 }
18
19 void loop() {
20     //Mostra as informações no Serial Monitor
21     Serial.print(" Tempo : ");
22     Serial.print(rtc.getTimeStr());
23     Serial.print(" Data : ");
24     Serial.print(rtc.getDateStr(FORMAT_SHORT));
25     Serial.print(" Dia:");
26     Serial.print(rtc.getDOWStr(FORMAT_SHORT));
27     Serial.print(" Mes:");
28     Serial.println(rtc.getMonthStr(FORMAT_SHORT));
29     //Aguarda 1 segundo e repete o processo
30     delay (1000);
31 }
32
```

Resultado.



The image shows a screenshot of a Windows terminal window titled "COM8". The window contains a list of ten lines of text, each representing a timestamp and date. The text is as follows:

```
Tempo : 01:39:00 Data : 01.12.16 Dia:Fri Mes:Dec  
Tempo : 01:39:01 Data : 01.12.16 Dia:Fri Mes:Dec  
Tempo : 01:39:02 Data : 01.12.16 Dia:Fri Mes:Dec  
Tempo : 01:39:03 Data : 01.12.16 Dia:Fri Mes:Dec  
Tempo : 01:39:04 Data : 01.12.16 Dia:Fri Mes:Dec  
Tempo : 01:39:05 Data : 01.12.16 Dia:Fri Mes:Dec  
Tempo : 01:39:06 Data : 01.12.16 Dia:Fri Mes:Dec  
Tempo : 01:39:07 Data : 01.12.16 Dia:Fri Mes:Dec  
Tempo : 01:39:08 Data : 01.12.16 Dia:Fri Mes:Dec  
Tempo : 01:39:09 Data : 01.12.16 Dia:Fri Mes:Dec
```

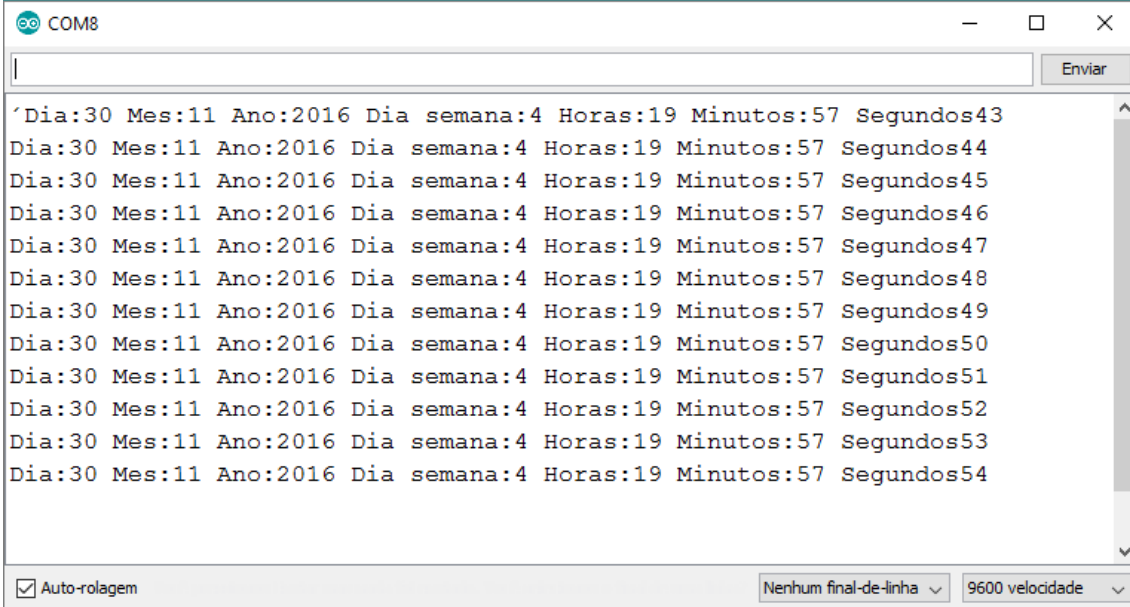
At the bottom of the window, there are three controls: a checked checkbox labeled "Auto-rolagem", a dropdown menu set to "Nenhum final-de-linha", and another dropdown menu set to "9600 velocidade".

## O programa 2:

Este exemplo pega os dados no formato numérico.

```
1 //Programa : Relogio com modulo RTC DS1307
2 //Autor : Professor Roberto Bairros
3 //pega string
4 //Carrega a biblioteca do RTC DS1307
5 #include <DS1307.h>
6 //Modulo RTC DS1307 ligado as portas A4 e A5 do Arduino
7 DS1307 rtc(A4, A5);
8 //Declara a variável t do tipo Time
9 Time t;
10 void setup() {
11     Serial.begin(9600); //inicia serial
12     //Aciona o relógio
13     rtc.halt(false);
14     //VOCÊ DEVE AJUSTAR A DATA E O TEMPO PREVIAMENTE
15     //Definicoes do pino SQW/Out
16     rtc.setSQWRate(SQW_RATE_1);
17     rtc.enableSQW(true);
18 }
19
20 void loop() {
21     // pega o tempo e a data
22     t = rtc.getTime();
23     //manda valores numericos DEC
24     Serial.print("Dia:");
25     Serial.print(t.date, DEC);
26     Serial.print(" Mes:");
27     Serial.print(t.mon, DEC);
28     Serial.print(" Ano:");
29     Serial.print(t.year, DEC);
30     Serial.print(" Dia semana:");
31     Serial.print(t.dow, DEC);
32     Serial.print(" Horas:");
33     Serial.print(t.hour, DEC);
34     Serial.print(" Minutos:");
35     Serial.print(t.min, DEC);
36     Serial.print(" Segundos");
37     Serial.println(t.sec, DEC);
38     delay (1000);
39 }
```

O resultado:



The screenshot shows a terminal window titled 'COM8'. The window contains a list of 13 lines of text, each representing a timestamp in the format: 'Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos43'. The seconds value increases by 1 for each subsequent line, from 43 to 54. At the bottom of the window, there are three controls: a checked checkbox for 'Auto-rolagem', a dropdown menu set to 'Nenhum final-de-linha', and another dropdown menu set to '9600 velocidade'.

```
'Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos43
Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos44
Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos45
Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos46
Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos47
Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos48
Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos49
Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos50
Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos51
Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos52
Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos53
Dia:30 Mes:11 Ano:2016 Dia semana:4 Horas:19 Minutos:57 Segundos54
```

Auto-rolagem    Nenhum final-de-linha    9600 velocidade

## Programas exemplos para copiar e colar:

### Programa 0.

```
//Programa : Relogio com modulo RTC DS1307
//Autor : Professor Roberto Bairros
//CARREGA DATA E HORA NO RTC
//Carrega a biblioteca do RTC DS1307
#include <DS1307.h>
//Modulo RTC DS1307 ligado as portas A4 e A5 do Arduino
DS1307 rtc(A4, A5);

void setup() {
  Serial.begin(9600);//inicia serial
  //Aciona o relógio
  rtc.halt(false);
  //As linhas abaixo ajustam a data e hora do modulo
  //e podem ser comentada apos a primeira utilizacao
  //Define o dia da semana
  //MONDAY TUESDAY WEDNESDAY THURSDAY FRIDAY SATURDAY SUNDAY
  rtc.setDOW(THURSDAY);
  rtc.setTime(19, 37, 0);//Define o horario (h,m,s)
  rtc.setDate(30, 11, 2016);//Define o (dia, mes e ano)
  //Definicoes do pino SQW/Out
  rtc.setSQWRate(SQW_RATE_1);
  rtc.enableSQW(true);
}

void loop() {
  //Mostra as informações no Serial Monitor
  Serial.print("Data : ");
  Serial.print(rtc.getDateStr(FORMAT_SHORT));
  Serial.print(" Tempo : ");
  Serial.println(rtc.getTimeStr());
  delay (1000);
}
```

Programa 1:

```
//Programa : Relogio com modulo RTC DS1307
//Autor : Professor Roberto Bairros
//pega string
//Carrega a biblioteca do RTC DS1307
#include <DS1307.h>
//Modulo RTC DS1307 ligado as portas A4 e A5 do Arduino
DS1307 rtc(A4, A5);

void setup() {
  Serial.begin(9600);//inicia serial
  //Aciona o relógio
  rtc.halt(false);
  //VOCÊ DEVE AJUSTAR A DATA E O TEMPO PREVIAMENTE
  //Definicoes do pino SQW/Out
  rtc.setSQWRate(SQW_RATE_1);
  rtc.enableSQW(true);
}

void loop() {
  //Mostra as informações no Serial Monitor
  Serial.print(" Tempo : ");
  Serial.print(rtc.getTimeStr());
  Serial.print(" Data : ");
  Serial.print(rtc.getDateStr(FORMAT_SHORT));
  Serial.print(" Dia:");
  Serial.print(rtc.getDOWStr(FORMAT_SHORT));
  Serial.print(" Mes:");
  Serial.println(rtc.getMonthStr(FORMAT_SHORT));
  //Aguarda 1 segundo e repete o processo
  delay (1000);
}
```

Programa 2:

```
//Programa : Relogio com modulo RTC DS1307
//Autor : Professor Roberto Bairros
//pega string
//Carrega a biblioteca do RTC DS1307
#include <DS1307.h>
//Modulo RTC DS1307 ligado as portas A4 e A5 do Arduino
DS1307 rtc(A4, A5);
//Declara a variável t do tipo Time
Time t;
void setup() {
  Serial.begin(9600); //inicia serial
  //Aciona o relógio
  rtc.halt(false);
  //VOCÊ DEVE AJUSTAR A DATA E O TEMPO PREVIAMENTE
  //Definições do pino SQW/Out
  rtc.setSQWRate(SQW_RATE_1);
  rtc.enableSQW(true);
}

void loop() {
  // pega o tempo e a data
  t = rtc.getTime();
  //manda valores numéricos DEC
  Serial.print("Dia:");
  Serial.print(t.date, DEC);
  Serial.print(" Mes:");
  Serial.print(t.mon, DEC);
  Serial.print(" Ano:");
  Serial.print(t.year, DEC);
  Serial.print(" Dia semana:");
  Serial.print(t.dow, DEC);
  Serial.print(" Horas:");
  Serial.print(t.hour, DEC);
  Serial.print(" Minutos:");
  Serial.print(t.min, DEC);
  Serial.print(" Segundos");
  Serial.println(t.sec, DEC);
  delay (1000);
}
```



### Exemplo do uso do tempo como variável numérica.

O programa abaixo usa a hora no formato numérico para desligar o LED da meia noite até a quatro horas da manhã. Este exemplo poderia ser usado para ligar o sinal amarelo de uma sinaleira durante a noite.

```
26 void loop()
27 {
28   t = rtc.getTime();//pega o tempo
29   Serial.print("\nHora : ");
30   Serial.print(t.hour, DEC);
31   if (t.hour>=0 && t.hour<4){
32     //desliga led
33     digitalWrite(13, LOW);
34   }
35   else {
36     //liga led
37     digitalWrite(13, HIGH);
38   }
```

Programa 3:

```
//Programa : Relogio com modulo RTC DS1307
//Autor : Professor Roberto Bairros
//Carrega a biblioteca do RTC DS1307
#include <DS1307.h>
Time t;//inicializa o tempo
//Modulo RTC DS1307 ligado as portas A4 e A5 do Arduino
DS1307 rtc(A4, A5);
void setup()
{
  //Aciona o relógio
  rtc.halt(false);
  //VOCÊ DEVE AJUSTAR A DATA E O TEMPO PREVIAMENTE
  //Definicoes do pino SQW/Out
  rtc.setSQWRate(SQW_RATE_1);
  rtc.enableSQW(true);
  pinMode(13, OUTPUT);
  Serial.begin(9600);//inicia a serial
}

void loop()
{
  t = rtc.getTime();//pega o tempo
  Serial.print("\nHora : ");
  Serial.print(t.hour, DEC);
  if (t.hour>=0 && t.hour<4){
    //desliga led
    digitalWrite(13, LOW);
  }
  else {
    //liga led
    digitalWrite(13, HIGH);
  }
  //Aguarda 1 segundo e repete o processo
  delay (1000);
}
```

### Conclusão:

O módulo RTC DS1307 é muito simples de usar e você poderá aplica-lo em temporizadores, minuteiras, relógios, despertadores e muito mais.

Sites: [www.bairrospd.com](http://www.bairrospd.com)

SEO: [www.bairrospd.com](http://www.bairrospd.com), eletrônica, tutorial, Arduino, RTC, DS1307