

Arduino: Nivel avanzado

José Antonio Vacas @javacasm

Darwin Eventur



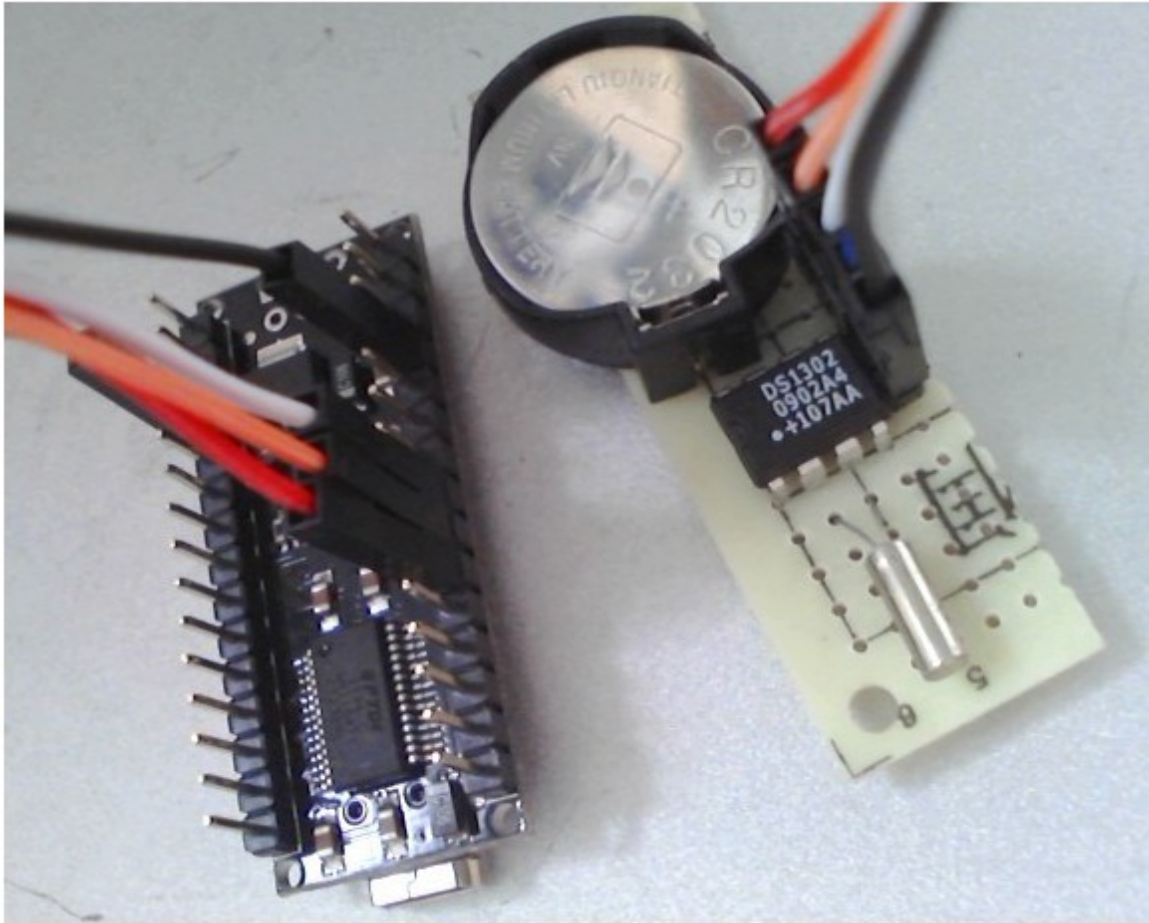
verás el mundo de otra manera

Facultad de ciencias

13 al 17 de Julio de 2015



Módulo con RTC 1302



La [librería DS1302](<http://playground.arduino.cc/uploads/Main/DS1302RTC.zip>:
(<http://playground.arduino.cc/uploads/Main/DS1302RTC.zip>:)

1. Real Time Clock read/write (8 bytes)
2. Battery backed RAM read/write (31 bytes)
3. Power save mode manipulation (start/stop clock)
4. Trickle charger setup
5. Burst mode read/write
6. 24 hour format only (12 hour format is function Time library)

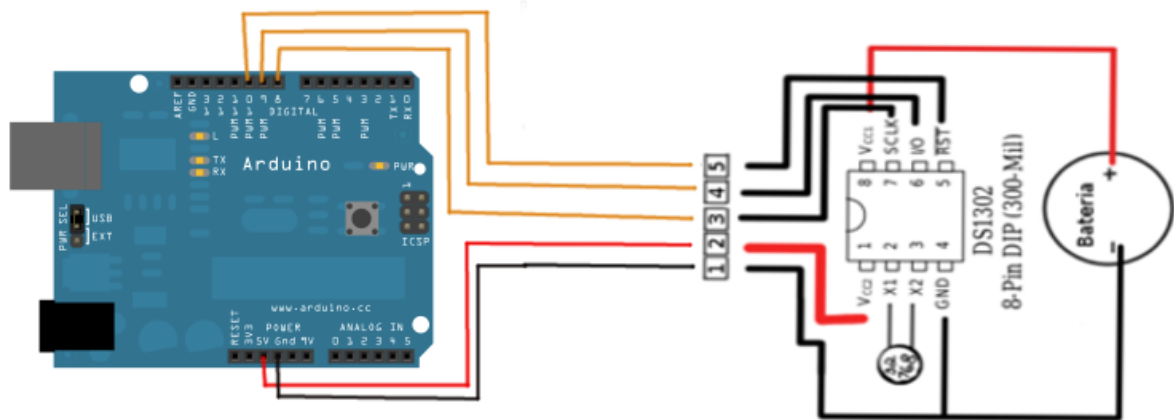
Whether used with the DS1302, the user is responsible for ensuring reads and writes do not exceed the device's address space (0x80-0x90 for DS1302 clock data and 0xC0-0xFC RAM data); no bounds checking is done by this library.

El DS1302 usa un 3-wire interface (The "Chip Enable" pin was called "/Reset):

- bidirectional data.
- clock
- chip select

No es I2C, ni OneWire, ni SPI.

Montaje



Más detalles (<http://playground.arduino.cc/Main/DS1302RTC>)

Otro tutorial (<http://www.instructables.com/id/Real-Time-Clock-DS1302/>)

Código

```
// Timur Maksimov 2014
//
// A quick demo of how to use DS1302-library to make a quick
// clock using a DS1302 and a 16x2 LCD.
//
// I assume you know how to connect the DS1302 and LCD.
// DS1302: CE pin    -> Arduino Digital 27
//              I/O pin  -> Arduino Digital 29
//              SCLK pin  -> Arduino Digital 31
//              VCC pin   -> Arduino Digital 33
//              GND pin   -> Arduino Digital 35
//
// LCD:      DB7          -> Arduino Digital 7
//              DB6          -> Arduino Digital 6
//              DB5          -> Arduino Digital 5
//              DB4          -> Arduino Digital 4
//              E            -> Arduino Digital 9
//              RS           -> Arduino Digital 8
```

```

#include <LiquidCrystal.h>
#include <DS1302RTC.h>
#include <Time.h>

// Init the DS1302
// Set pins: CE, IO, CLK
DS1302RTC RTC(27, 29, 31);

// Optional connection for RTC module
#define DS1302_GND_PIN 33
#define DS1302_VCC_PIN 35

// Init the LCD
// initialize the library with the numbers of the interface pins
// lcd(RS, E, d4, d5, d6, d7)
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

void setup()
{
    // Setup LCD to 16x2 characters
    lcd.begin(16, 2);

    // Activate RTC module
    digitalWrite(DS1302_GND_PIN, LOW);
    pinMode(DS1302_GND_PIN, OUTPUT);

    digitalWrite(DS1302_VCC_PIN, HIGH);
    pinMode(DS1302_VCC_PIN, OUTPUT);

    lcd.print("RTC activated");

    delay(500);

    // Check clock oscillation
    lcd.clear();
    if (RTC.haltRTC())
        lcd.print("Clock stopped!");
    else
        lcd.print("Clock working.");

    // Check write-protection
    lcd.setCursor(0,1);
    if (RTC.writeEN())
        lcd.print("Write allowed.");
    else
        lcd.print("Write protected.");

    delay ( 2000 );

    // Setup Time library
    lcd.clear();
    lcd.print("RTC Sync");

```

```

    setSyncProvider(RTC.get()); // the function to get the time from the RTC
    if(timeStatus() == timeSet)
        lcd.print(" Ok!");
    else
        lcd.print(" FAIL!");

    delay ( 2000 );

    lcd.clear();
}

void loop()
{

    // Display time centered on the upper line
    lcd.setCursor(3, 0);
    print2digits(hour());
    lcd.print(" ");
    print2digits(minute());
    lcd.print(" ");
    print2digits(second());

    // Display abbreviated Day-of-Week in the lower left corner
    lcd.setCursor(0, 1);
    lcd.print(dayShortStr(weekday()));

    // Display date in the lower right corner
    lcd.setCursor(5, 1);
    lcd.print(" ");
    print2digits(day());
    lcd.print("/");
    print2digits(month());
    lcd.print("/");
    lcd.print(year());

    // Warning!
    if(timeStatus() != timeSet) {
        lcd.setCursor(0, 1);
        lcd.print(F("RTC ERROR: SYNC!"));
    }

    delay ( 1000 ); // Wait approx 1 sec
}

void print2digits(int number) {
    // Output leading zero
    if (number >= 0 && number < 10) {
        lcd.write('0');
    }
    lcd.print(number);
}

```

