Vacas Martínez, José Antonio

# Arduino: conviértete en un experto desde cero



Licencia CC by SA

## by @javacasm

### José Antonio Vacas Martínez

### http://elCacharreo.com

## Marzo 2020

# Arduino: Nivel avanzado 3ed

## José Antonio Vacas @javacasm

## Organiza: Darwin Eventur

## BiblioMaker - Facultad de Ciencias

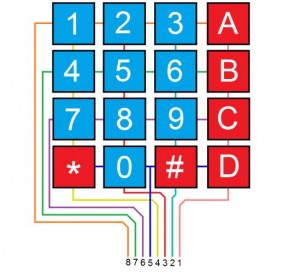
## 23 al 27 de Septiembre de 2019

## http://bit.ly/ArduinoAvanzado19

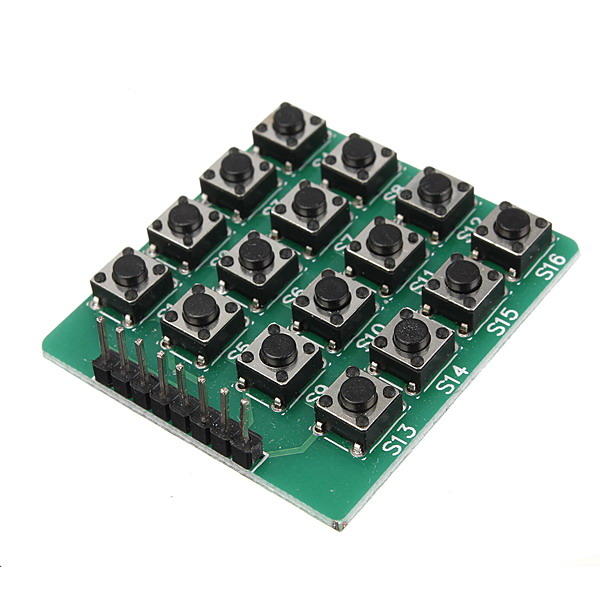


CC

# Uso de teclados (Keypad)

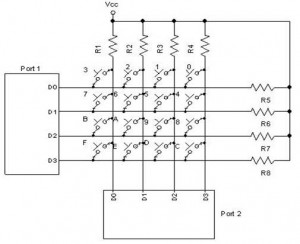


Teclado



Teclado2

Internamente son así



Teclado interno

Conectaremos los pines, por ejemplo de esta forma

![pinout](data:text/html; charset=iso-8859-1;base64,PCFET0NUWVBFIEhUTUwgUFVCTElDICItLy9JRVRGLy9EVEQgSFRNTCAyLjAvL0VOIj4KPGh0bWw+PGhlYWQ+Cjx0aXRsZT40MDMgRm9yYmlkZGVuPC90aXRsZT4KPC9oZWFkPjxib2R5Pgo8aDE+Rm9yYmlkZGVuPC9oMT4KPHA+WW91IGRvbid0IGhhdmUgcGVybWlzc2lvbiB0byBhY2Nlc3MgdGhpcyByZXNvdXJjZS48L3A+CjxwPkFkZGl0aW9uYWxseSwgYSA0MDMgRm9yYmlkZGVuCmVycm9yIHdhcyBlbmNvdW50ZXJlZCB3aGlsZSB0cnlpbmcgdG8gdXNlIGFuIEVycm9yRG9jdW1lbnQgdG8gaGFuZGxlIHRoZSByZXF1ZXN0LjwvcD4KPC9ib2R5PjwvaHRtbD4K)

pinout

Y usaremos la librería [Keypad](https://playground.arduino.cc/Code/Keypad/)

#include<Keypad.h>  
 const byte filas = 4;  
 const byte columnas = 4;  
 byte pinesF[filas] = {9,8,7,6}; // Donde hemos conectado las filas  
 byte pinesC[columnas] = {5,4,3,2}; // Donde hemos conectado las columnas  
   
 char teclas[filas][columnas] = { // Definición de las teclas  
 {'1','2','3','A'},  
 {'4','5','6','B'},  
 {'7','8','9','C'},  
 {'\*','0','#','D'}  
 };  
   
 Keypad teclado = Keypad(makeKeymap(teclas), pinesF, pinesC, filas, columnas);  
   
 char tecla;  
  
 void setup() {  
 Serial.begin(9600);  
 }  
   
 void loop() {  
 tecla = teclado.getKey();  
 if (tecla != 0)  
 Serial.print(tecla);  
 }

## Más detalles

[Teclados matriciales por Prometec](https://www.prometec.net/teclados-matriciales/) [Teclados matriciales por Luis Llamos](https://www.luisllamas.es/arduino-teclado-matricial/)

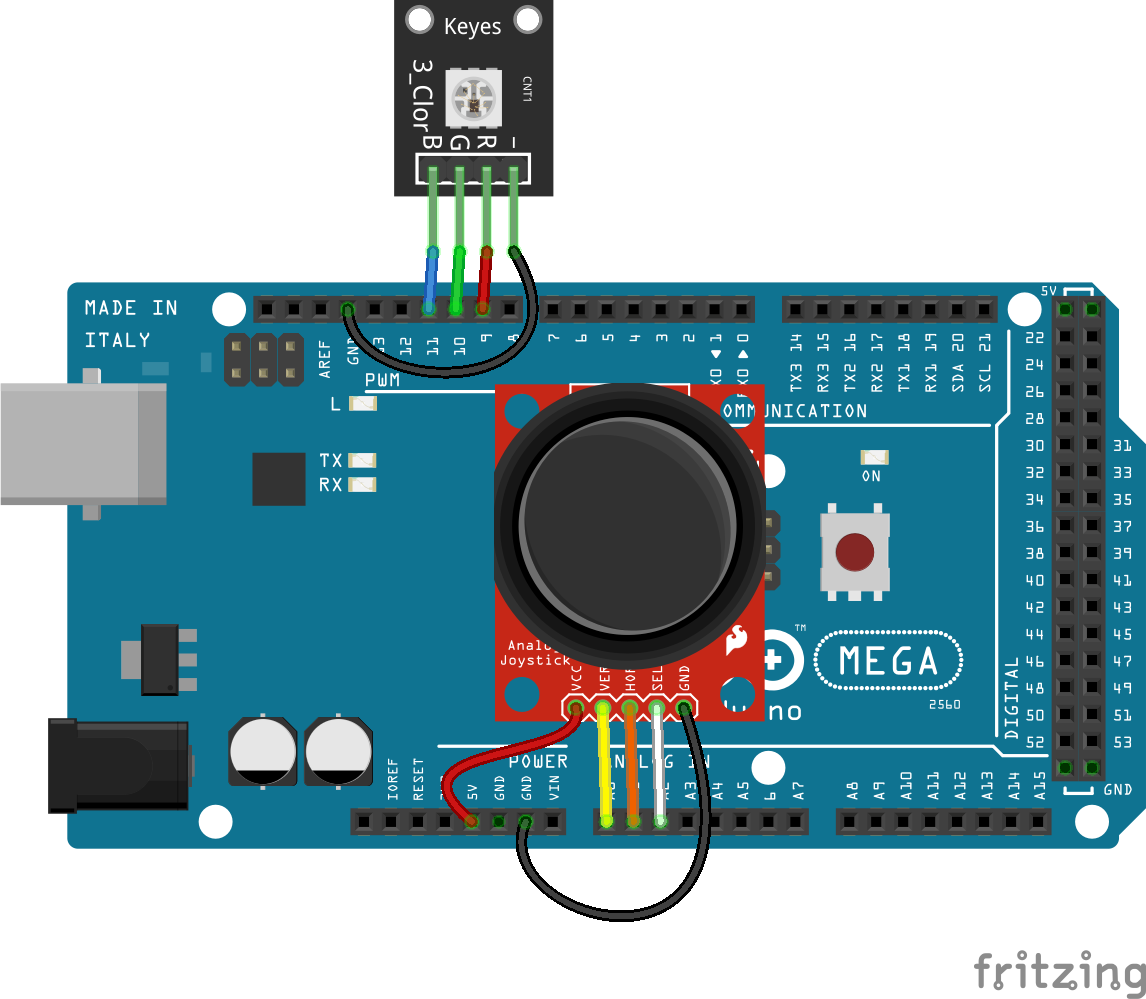
## Joystick XY



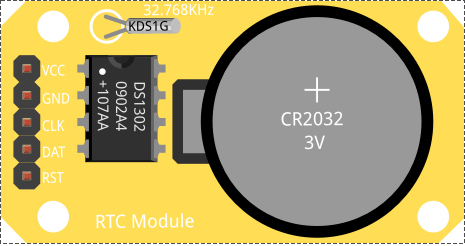
Módulo joystick de doble eje y pulsador

Este módulo joystick de doble eje (x e y) y pulsador está formado por 2 potenciómetros

2 potenciómetros y switch

 ## Módulo con RTC 1302

Un RTC es un Reloj de Tiempo Real, es decir un reloj capaz de medir el paso del tiempo, normalmente dotado de una pila para que no se pierda la fecha y que se puede comunicar con Arduino



La [librería DS1302](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)

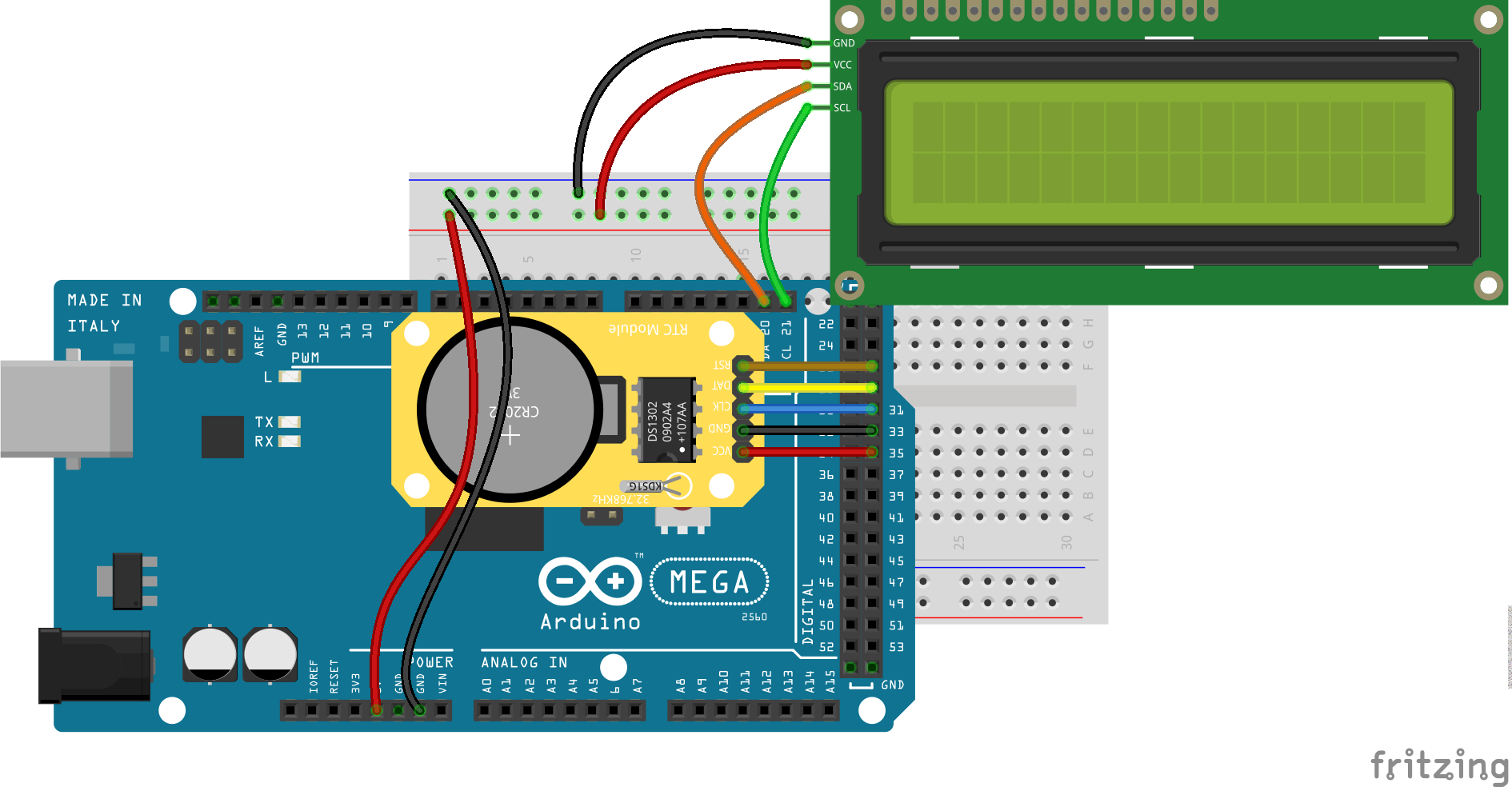
Whether used with the DS1302, the user is responsible for ensuring reads and writes do not exceed the device's address space (0x80-0x90 for DS1302 clock data and 0xC0-0xFC RAM data); no bounds checking is done by this library.

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

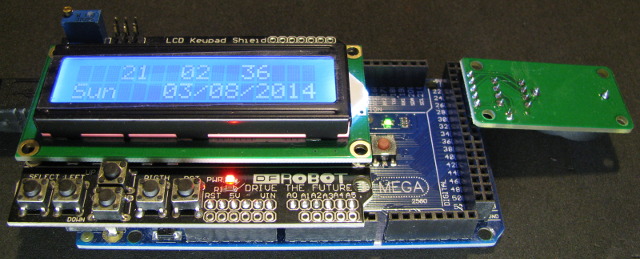
* bidirectional data.
* clock
* chip select

No es I2C, ni OneWire, ni SPI.

## Montaje



montaje



Montaje visual

[Más detalles](http://playground.arduino.cc/Main/DS1302RTC)

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

## Código

// Basado en Timur Maksiomv 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 I2C:   
// SDA -> Arduino Digital 20  
// SCL -> Arduino Digital 21  
// GND -> ND  
// Vcc -> 5V  
  
// Includes del LCD  
#include <Wire.h>   
#include <LiquidCrystal\_I2C.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  
LiquidCrystal\_I2C lcd(0x27,16,2);   
  
void showMessage(String texto){  
 Serial.print(texto);  
 lcd.setCursor(0,0);  
 lcd.print(texto);  
}  
  
  
void setup()  
{  
 lcd.init();   
 lcd.backlight();  
  
 // Activate RTC module  
 digitalWrite(DS1302\_GND\_PIN, LOW);  
 pinMode(DS1302\_GND\_PIN, OUTPUT);  
  
 digitalWrite(DS1302\_VCC\_PIN, HIGH);  
 pinMode(DS1302\_VCC\_PIN, OUTPUT);  
  
 showMessage("RTC activated");  
  
 delay(500);  
  
 // Check clock oscillation   
 lcd.clear();  
 if (RTC.haltRTC())  
 showMessage("Clock stopped!");  
 else  
 showMessage("Clock working.");  
  
 // Check write-protection  
 if (RTC.writeEN())  
 showMessage("Write allowed.");  
 else  
 showMessage("Write protected.");  
  
 delay ( 2000 );  
  
 // Setup Time library   
 showMessage("RTC Sync");  
 setSyncProvider(RTC.get); // the function to get the time from the RTC  
 if(timeStatus() == timeSet)  
 showMessage(" Ok!");  
 else  
 showMessage(" 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);  
}