circuito con RTC para arduino

Antes de empezar:

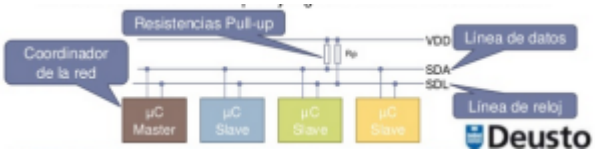
Comunicación I2C y SPI

I2C

I2C (Inter-integrated Circuit) es un bus de comunicaciones serie síncrona muy utilizado en la industria para la comunicación entre controladores y sus periféricos en sistemas integrados.

Consideraciones

- Necesitan resistencias pull-up, ya que los dispositivos sólo pueden forzar al bus a ponerse a '0'.
- la línea de tierra debe ser común a todos los dispositivos (estar unida).
- Es bidireccinal half-Duplex y sigue el modelo maestro-esclavo.



Solo utiliza dos líneas para transmitir datos (SDA) y la señal de reloj (SCL)

Los dispositivos de un bus I2C tienen una dirección única para cada uno, y que pueden clasificarse como maestros o como esclavos.

1. El **maestro es el que inicia la transferencia de datos y genera la señal de reloj**
2. El **esclavo** espera a que un maestro se comunique con el para generar la transferencia.

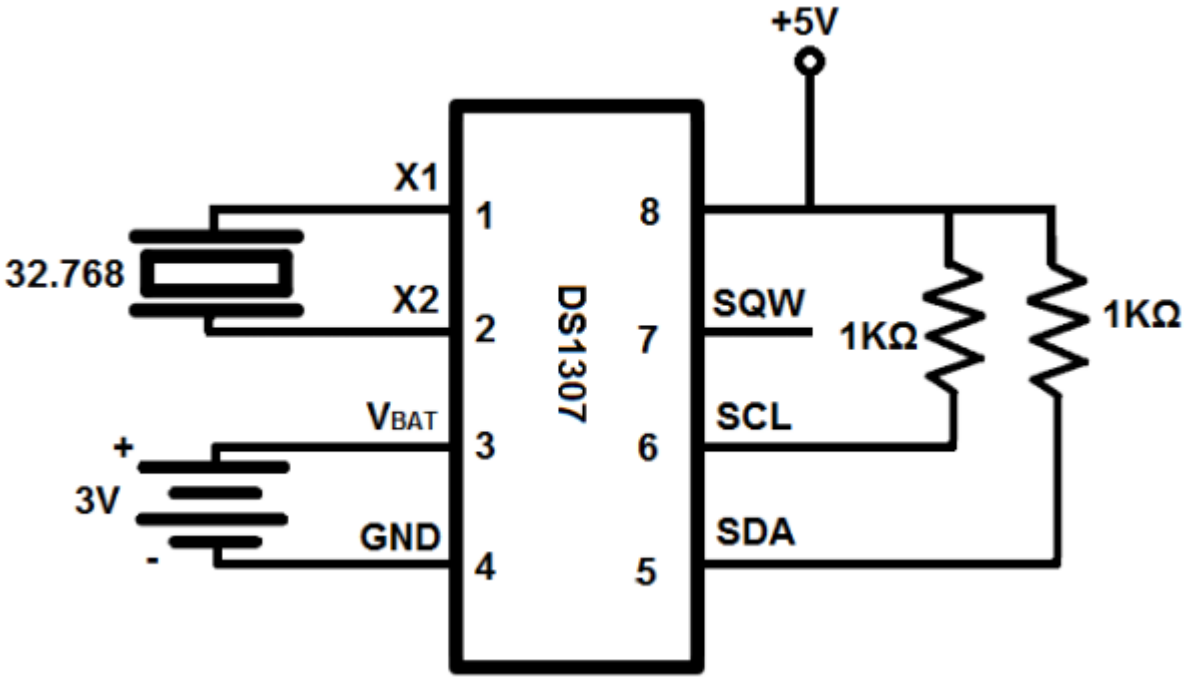
SPI

SPI(Serial Peripherial Interface) es un bus de comunicaciones serie síncrona usado para la transferencia serie síncrona usado para la transferencia de información entre circuitos integrados en equipos electrónicos.

Consideraciones

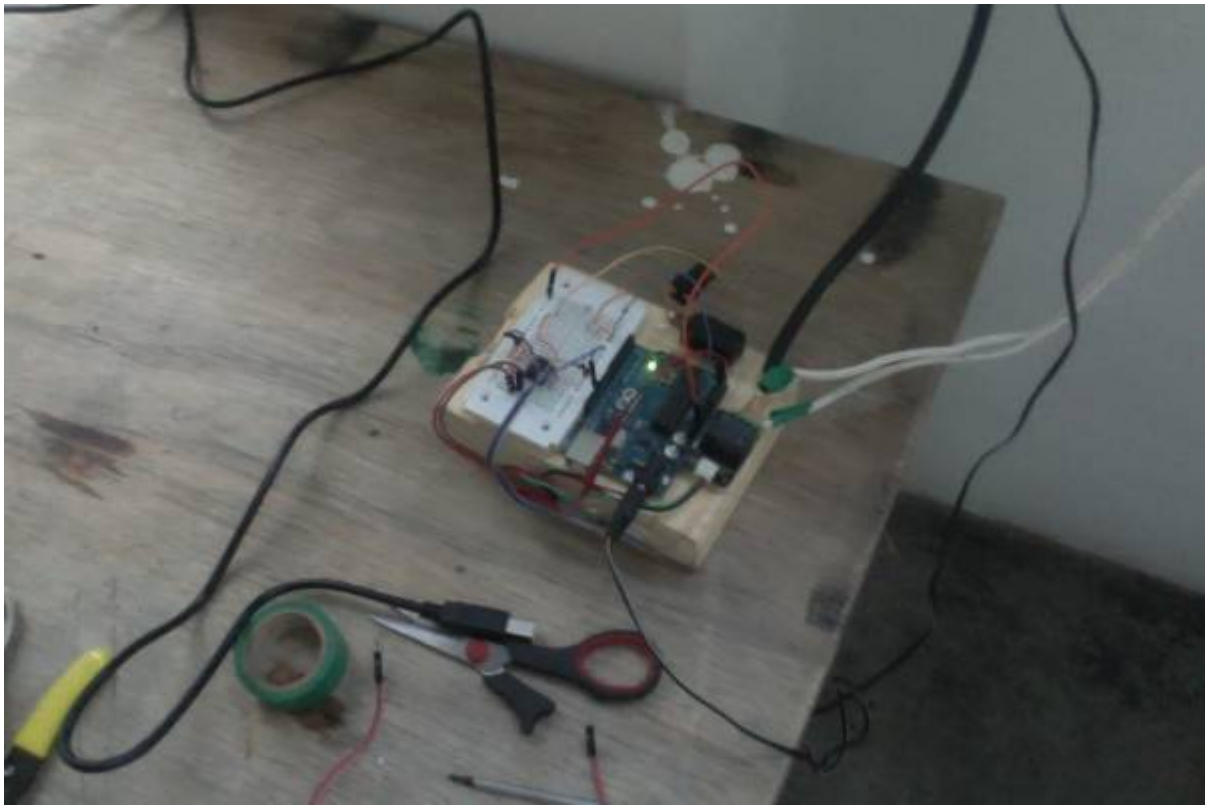
1. Utiliza 3 líneas para transmitir (SDO), recibir (SDI) y para generar la señal de reloj (SCK).
2. la señal de reloj la genera uno de los dispositivos (el que actúa de maestro).
3. la linea de tierra debe ser común a todos los dispositivos (estar unida)
4. Permite comunicación Full-Duplex y utiliza el modelo maestro-esclavo.

Esquema básico del circuito.



RTC o Reloj en Tiempo Real – Es un circuito electrónico especializado cuya función es mantener la hora y fecha actual en un sistema informático (ya sea con microcontrolador u otro tipo de CPU). Se caracteríza por tener un bajo consumo de energía y también normalmente su propia fuente de alimentación auxiliar. Normalmente al recurrir a este tipo de circuitos integrados obtenemos una mejor precisión del tiempo. Un ejemplo de dispositivos que incluyen relojes en tiempo real son las computadoras personales (PC).

Circuito



Codigo actualizado

#include “RTClib.h” #include <Wire.h>

RTC_DS1307 RTC;

void setup() {

```
Serial.begin(57600);
Wire.begin();
RTC.begin();
pinMode(13, OUTPUT);
if(!RTC.isrunning())
{
 // Serial.println("RTC is not runnig");
  RTC.adjust(DateTime(__DATE__, __TIME__));
}
```

}

void loop() {

```
DateTime now = RTC.now();
/*
Serial.print(now.year(), DEC);
Serial.print('/');
Serial.print(now.month(), DEC);
Serial.print('/');
Serial.print(now.day(), DEC);
Serial.print(' ');
Serial.print(' ');
Serial.print(now.hour(), DEC);
Serial.print(':');
```

```
Serial.print(now.minute(), DEC);
Serial.print(':');
Serial.print(now.second(), DEC);
Serial.println();
*/
```

if (now.hour() == 19) {

```
if(now.minute() == 30)
{
  digitalWrite(13, HIGH);
}
if(now.minute() == 35)
{
  digitalWrite(13, LOW);
  delay(1000);
```

```
}
```

}

```
delay(1000);
```

}

Sensor de humedad capacitivo

Conceptos básicos de la humedad del suelo

DIY sensor - http://zerocharactersleft.blogspot.com.co/2011/11/pcb-as-capacitive-soil-moisture-sensor.html

opciones comerciales - http://vegetronix.com/

From:
https://wiki.unloquer.org/ -

Permanent link:
https://wiki.unloquer.org/documentacion-proceso/tecnologicos/rtc?rev=1453222588

Last update: **2016/01/19 16:56**