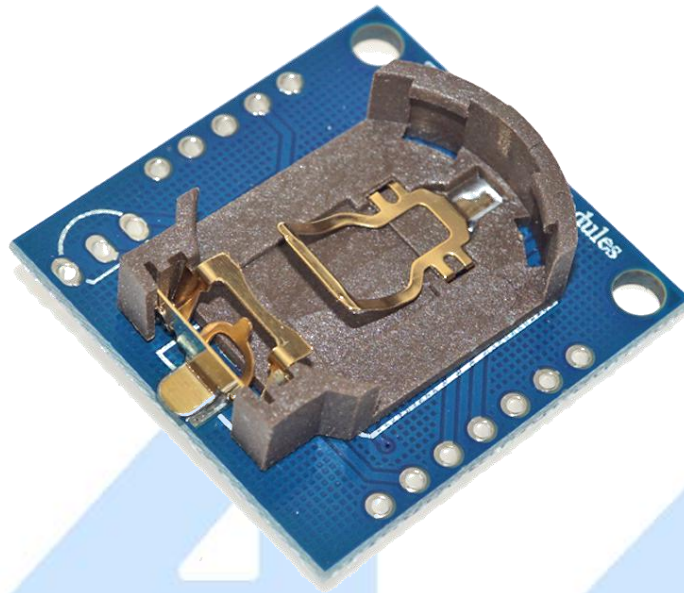


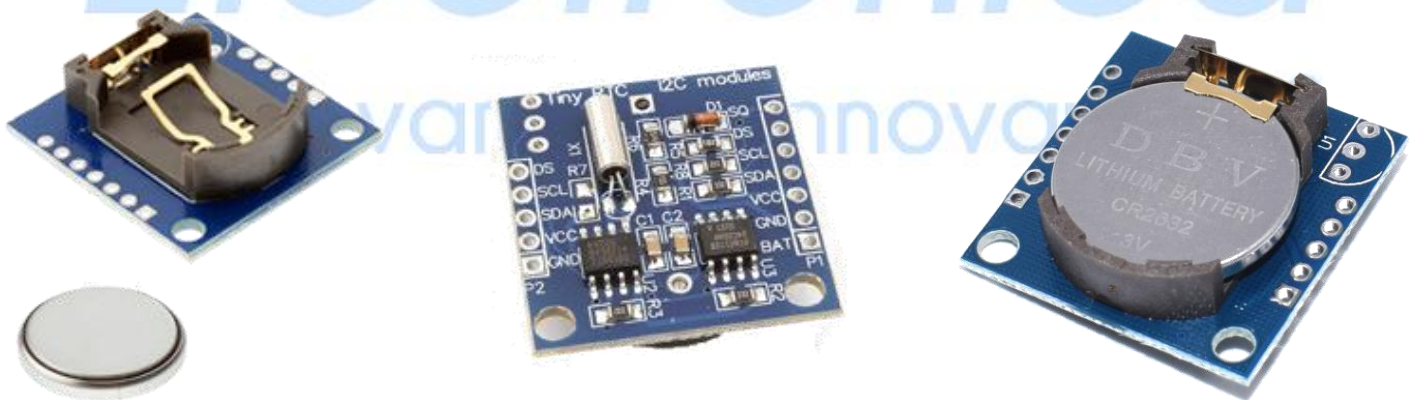
OKY3392: MÓDULO TINY RTC DS1307



Descripción

Módulo RTC (Real Time Clock), contiene un circuito integrado, cuya función es el seguimiento y almacenaje de la hora actual de manera autónoma, los RTC están presentes en casi cualquier dispositivo electrónico que necesite mantener la hora exacta. Estos dispositivos usualmente tienen una fuente de energía alternativa (batería de botón) para que puedan continuar suministrando el tiempo aunque la fuente de energía principal se encuentre apagada o no disponible.

El DS1307 internamente cuenta con un circuito que al detectar fallas de energía, cambia automáticamente a la fuente de respaldo, además utiliza un oscilador de cristal de cuarzo de 32.768kHz que genera una frecuencia estable y divisible mediante hardware digital para lograr medir segundos exactos.



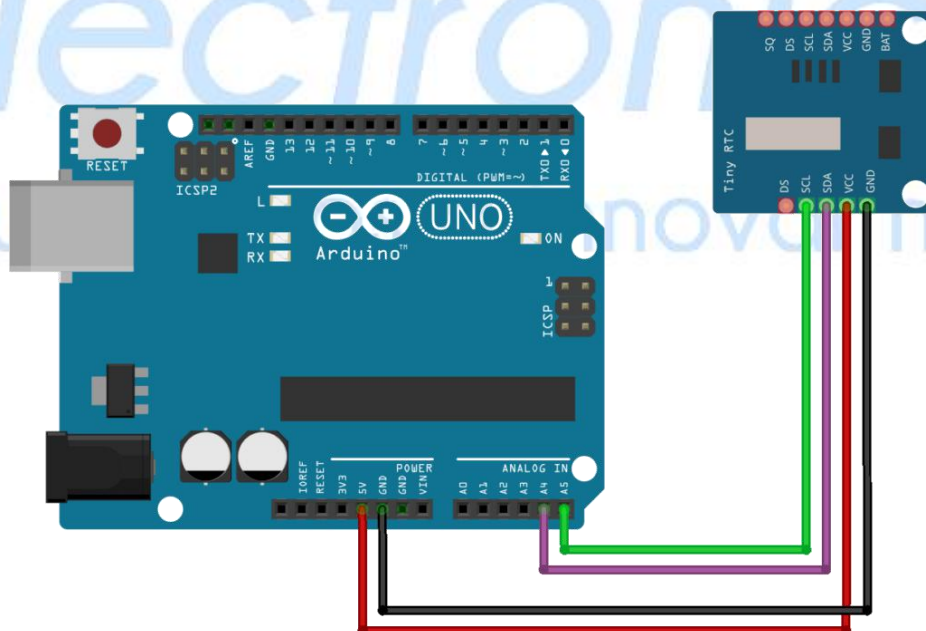
Características

- Compatible con Arduino.
- Interfaz: I2C de bus bidireccional.
- Incluye una memoria EEPROM 12C para almacenamiento de datos.
- El reloj provee: segundos, minutos y horas.
- El calendario provee: día, fecha, mes y año.
- La información del reloj y calendario opera en formato de: 24 o 12 horas y con indicador de AM / PM.
- El módulo consume menos de 500 nA en modo respaldo de batería con el oscilador en ejecución.
- Dispone de 56 bytes de memoria no volátil disponibles para el usuario.

Especificaciones

- Voltaje de entrada: 5VCD.
- Batería externa: Batería de Botón de 3V.
- Tamaño de PCB: 2.6 x 2.9 mm.
- Temperatura de trabajo: -40 ~ 85°C.

Diagrama de conexión



Código de muestra

```
/*  
 * TimeRTCLogger.pde  
 * example code illustrating adding and subtracting Time.  
 *  
 * this sketch logs pin state change events  
 * the time of the event and time since the previous event is calculated and sent to the serial port.  
 */
```

```
#include <TimeLib.h>
```

```
#include <Wire.h>
```

```
#include <DS1307RTC.h> // a basic DS1307 library that returns time as a time_t
```

```
const int nbrInputPins = 6; // monitor 6 digital pins  
const int inputPins[nbrInputPins] = {2,3,4,5,6,7}; // pins to monitor  
boolean state[nbrInputPins]; // the state of the monitored pins  
time_t prevEventTime[nbrInputPins]; // the time of the previous event
```

```
void setup() {  
  Serial.begin(9600);  
  setSyncProvider(RTC.get); // the function to sync the time from the RTC  
  for(int i=0; i < nbrInputPins; i++){  
    pinMode( inputPins[i], INPUT);  
    // uncomment these lines if pull-up resistors are wanted  
    // pinMode( inputPins[i], INPUT_PULLUP);  
    // state[i] = HIGH;  
  }  
}
```

```
void loop()  
{  
  for(int i=0; i < nbrInputPins; i++)  
  {  
    boolean val = digitalRead(inputPins[i]);  
    if(val != state[i])
```

```

{
    time_t duration = 0; // the time since the previous event
    state[i] = val;
    time_t timeNow = now();
    if(prevEventTime[i] > 0)
        // if this was not the first state change, calculate the time from the previous change
        duration = timeNow - prevEventTime[i];
    logEvent(inputPins[i], val, timeNow, duration ); // log the event
    prevEventTime[i] = timeNow; // store the time for this event
}
}
}

```

```

void logEvent( int pin, boolean state, time_t timeNow, time_t duration)

```

```

{
    Serial.print("Pin ");
    Serial.print(pin);
    if( state == HIGH)
        Serial.print(" went High at ");
    else
        Serial.print(" went Low at ");
    showTime(timeNow);
    if(duration > 0){
        // only display duration if greater than 0
        Serial.print(", Duration was ");
        showDuration(duration);
    }
    Serial.println();
}

```

```

void showTime(time_t t){
    // display the given time
    Serial.print(hour(t));
    printDigits(minute(t));
    printDigits(second(t));
    Serial.print(" ");
    Serial.print(day(t));
    Serial.print(" ");
    Serial.print(month(t));
    Serial.print(" ");
    Serial.print(year(t));
}

```

```

void printDigits(int digits){
    // utility function for digital clock display: prints preceding colon and leading 0
    Serial.print(":");
    if(digits < 10)
        Serial.print('0');
    Serial.print(digits);
}

```

```

void showDuration(time_t duration){
// prints the duration in days, hours, minutes and seconds
    if(duration >= SECS_PER_DAY){
        Serial.print(duration / SECS_PER_DAY);
        Serial.print(" day(s) ");
        duration = duration % SECS_PER_DAY;
    }
    if(duration >= SECS_PER_HOUR){
        Serial.print(duration / SECS_PER_HOUR);
        Serial.print(" hour(s) ");
        duration = duration % SECS_PER_HOUR;
    }
}

```

```
if(duration >= SECS_PER_MIN){  
    Serial.print(duration / SECS_PER_MIN);  
    Serial.print(" minute(s) ");  
    duration = duration % SECS_PER_MIN;  
}  
Serial.print(duration);  
Serial.print(" second(s) ");  
}
```



 AG Electrónica <i>¡Que vamos a innovar hoy!</i>		AG Electrónica S.A. de C.V. República del Salvador N° 20 Segundo Piso Teléfono: 5130 - 7210	
ACOTACIÓN: N/A	http://www.agelectronica.com	ESCALA: N/A	REALIZO: DCR
			REV: EAC
TOLERANCIA: N/A	MÓDULO TINY RTC DS1307		
TOLERANCIA: N/A	Fecha: 07/03/2019	No. Parte: OKY3392	