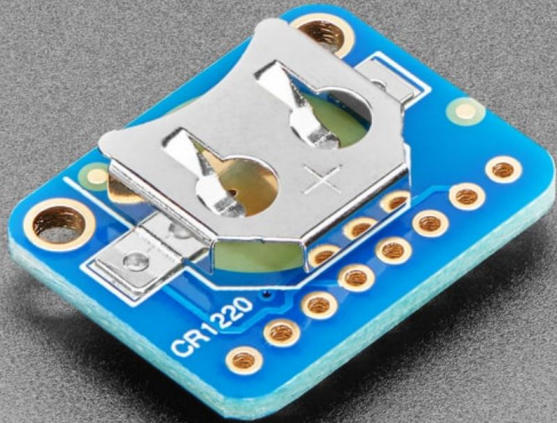


RTC (Real Time Clock)



Group 3: Gus Glasheen, Kyle Bender, Brian Hung, Shrey, Chieh Hsu



Introduction to RTCs.

A real time clock (RTC) is a digital clock is able to keep accurate account of time in the absence of a power supply or when a device is in low power mode.

An RTC is made up of a controller, an oscillator, and a quartz crystal resonator. The fact that it is an all-in-one device produces more reliable performances than discrete components.



Introduction to RTCs.

Registers, or functions of the RTC, are updated periodically and programs its data into RAM memory.

The RTC also includes another low power backup power source, which allows it to maintain continuous time counts in the presence of main power, allowing the user to not have to reset the time and date each time the device is rebooted.



Use of RTCs

Many products today are powered by batteries and have no access to recharging power sources. As a consequence, having all system functions in operation would quickly drain the battery. The way to get around this is to have scheduled intermittent activation to extend the battery life.

When microcontrollers are in a sleep mode, the RTC will maintain accurate timekeeping and alarm functions, drawing an amount of current as low as 0.5 μA .



Use of RTCs

The RTC saves current by always staying on as the lowest current usage device.

Even when the RTC is not used as a power saving device, the continuous timekeeping function is critical for the proper function of today's electronics, medical devices, and industrial products where power savings and backup timekeeping are at premium.

Aside from its power-saving function, the timekeeping function is essential for the proper functioning of various electronics, medical devices, and industrial products.



What makes DS3231 stand out from other RTCs.

Most RTCs utilize an external 32kHz timing crystal that is used for timekeeping with low current draw.

- > Crystals have a slight drift, amplified by temperature changes

For DS3231, the crystal is inside the chip to reduce environmental exposure, and next to it is a temperature sensor, which compensates the crystal's frequency changes by adjusting clock ticks.

The DS3231 can still function even when the main power is lost when a coin cell is plugged into the back

DS3231 Pinouts



Vin- Power pin, since RTC can be powered from 2.3V to 5.5V, no regulator needed for either 3.3V or 5V power

GND- Ground for power

SCL- Connects to the I2C clock line on microcontroller, has a 10K Ω pull up resistor, which ensures the power state is “high”, when not connected with any pins

SDA- Connects to I2C data line on microcontroller, also has a 10K Ω pull up resistor

DS3231 Pinouts



BAT- Allows for the powering of other things from coin cell battery. Also allows for battery backup from separate battery

32K- 32KHz oscillator output, which allows for outputs higher than the supplied voltage, needs a pull up resistor to be able to read from microcontroller

SQW- Optional square wave output, same as above, needing a pull up resistor to be able to read from microcontroller

!RST- An input between DS3231 and microcontroller, also allows for resetting on the device and indicating if main power is lost. Has an internal 50KΩ pull up resistor

Arduino Usage

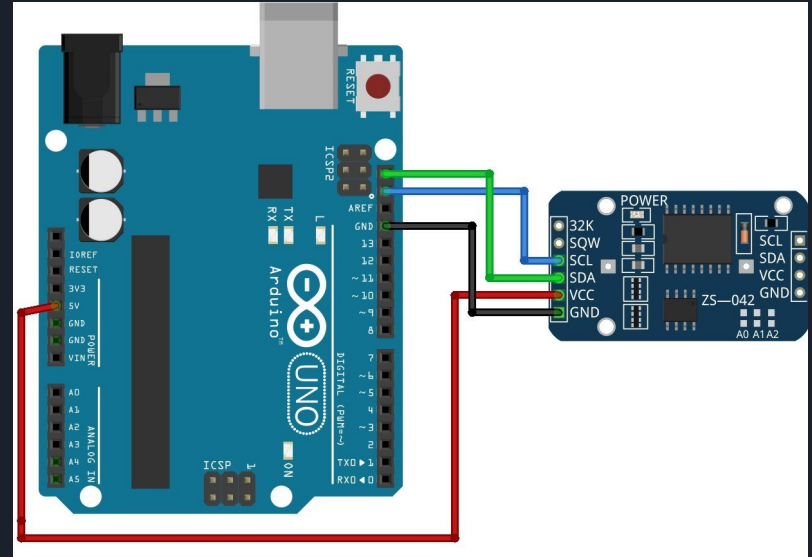
To use the sensor with an Arduino,

Connect Vin (red wire) to the power supply

Connect GND (black wire) to common power/data ground

Connect SCL (Green wire) to the SCL on the Arduino

Connect SDA (Blue wire) to the SDA on the Arduino





Arduino Usage

To program the sensor, we use the RTCLib Library from the Arduino Library manager

We can program the sensor to print the current year, month, day, hour, minute, second through the `now()` function.

Some libraries allow us to use `RTC.year()` or `RTC.hour()` to get the current year or hour.

We can also get a timestamp by calling `unixtime` which counts the number of seconds since midnight, January 1st 1970.



Arduino Usage (Circuit Python)

The DS3231 can also run the CircuitPython library.

We have to install Adafruit CircuitPython DS3231 onto our Arduino, then we have to import the module onto our board.

We can see the current year, month, day, hour, minute, seconds by calling `datetime`.

If time is not set, it defaults to January 1st, 2000.

However, we can set time by calling `datetime` and inputting the values as we want. Such as:

```
import time
```

```
ds3231.datetime = time.struct_time((2017, 1, 1, 0, 0, 0, 6, 1, -1))
```

Which sets time to midnight on January 1st, 2017



Sources

<https://ecsxtal.com/what-is-a-real-time-clock-rtc>

<https://www.adafruit.com/product/3013>

<https://microcontrollerslab.com/ds3231-rtc-module-pinout-interfacing-with-arduino-features/>



Thanks for listening!