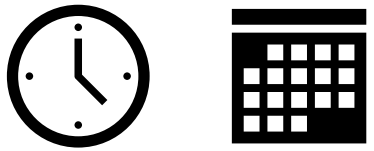
Introduction to Real Time Clocks

TI Precision Labs – Microcontrollers

Presented by Brandon Fisher

Prepared by Evan Lew

What is a real time clock?



A **real time clock** (RTC) is a highly accurate time keeping reference that enables systems to keep track of time and date in human readable formats

Scheduled Actions

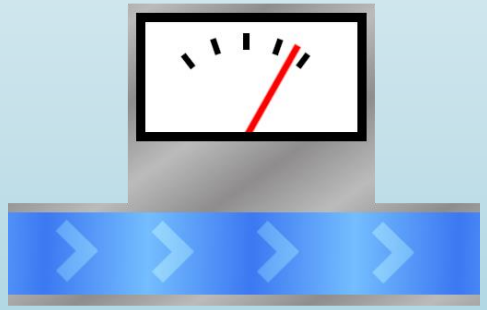
A thermostat changes the temperature setpoint on a schedule

Timestamping

Logging information may need to be saved with an accurate timestamp

Metering

A water meter must record water usage on a monthly interval



Alternatives for timekeeping

Software Delay

- ⊕ Simple
- ⊖ Low accuracy
- ⊖ CPU required

```
main()
{

  initialize_application();

  while(1)
  {
     update_application();
     delay(100000);
  }
}
```

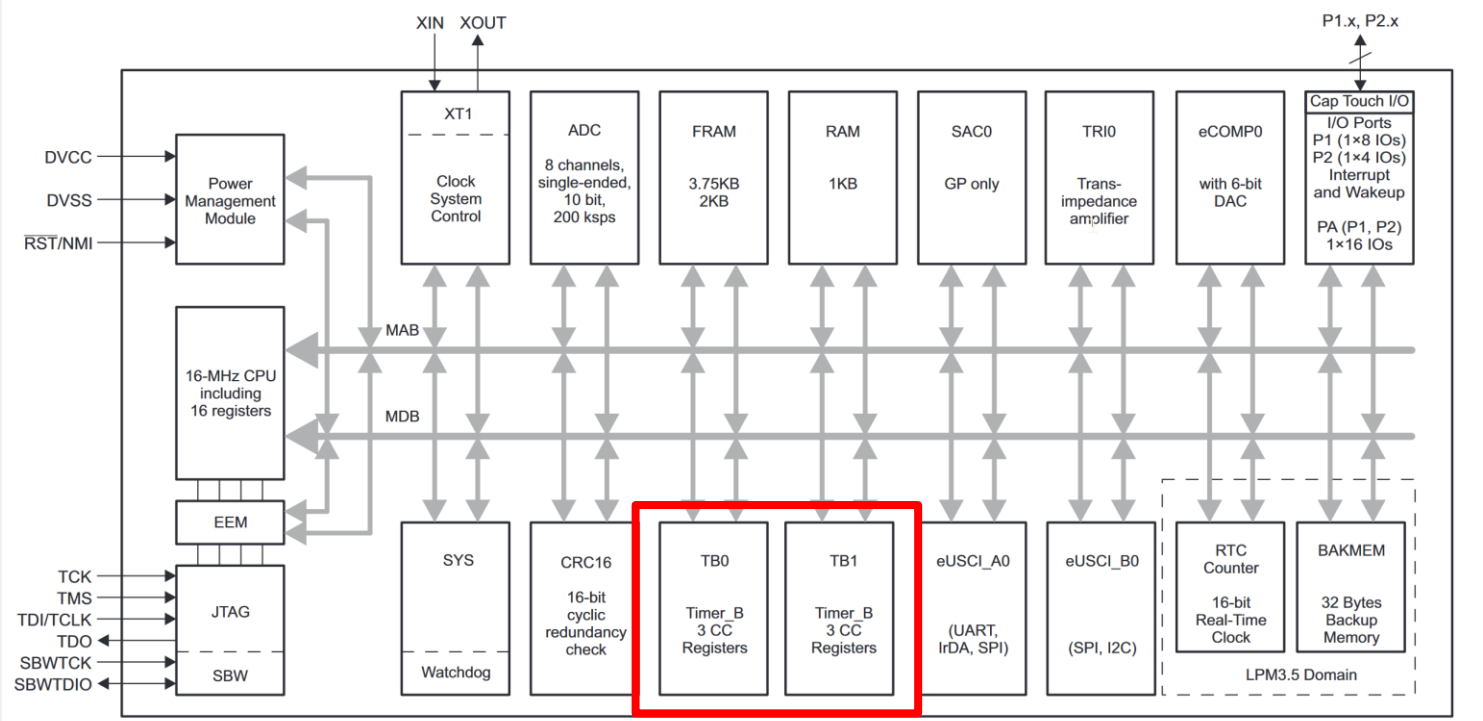
Alternatives for timekeeping

Software Delay

- ⊕ Simple
- ⊖ Low accuracy
- ⊖ CPU required

Hardware Timer

- ⊕ Accurate
- ⊕ No CPU required
- ⊖ Missing calendar features



Alternatives for timekeeping

Software Delay

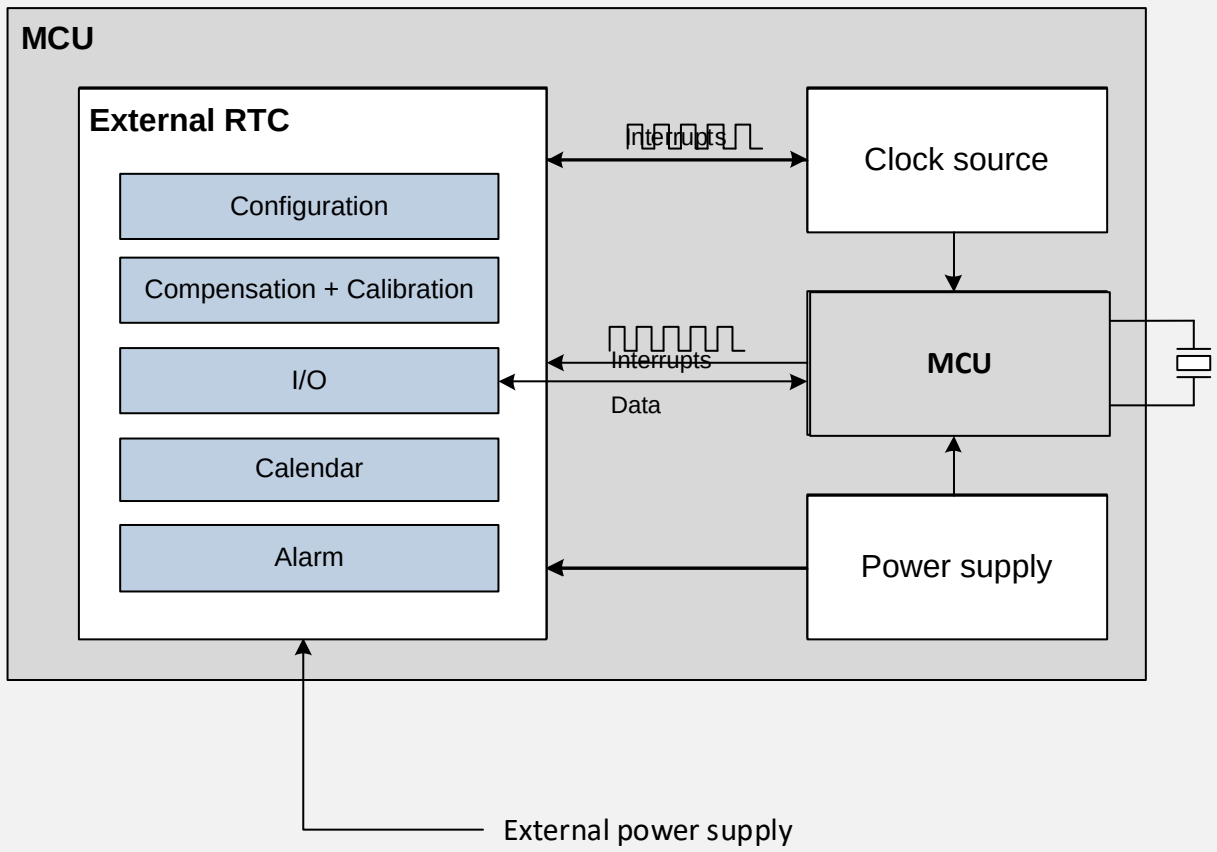
- ⊕ Simple
- ⊖ Low accuracy
- ⊖ CPU required

Hardware Timer

- ⊕ Accurate
- ⊕ No CPU required
- ⊖ Missing calendar features

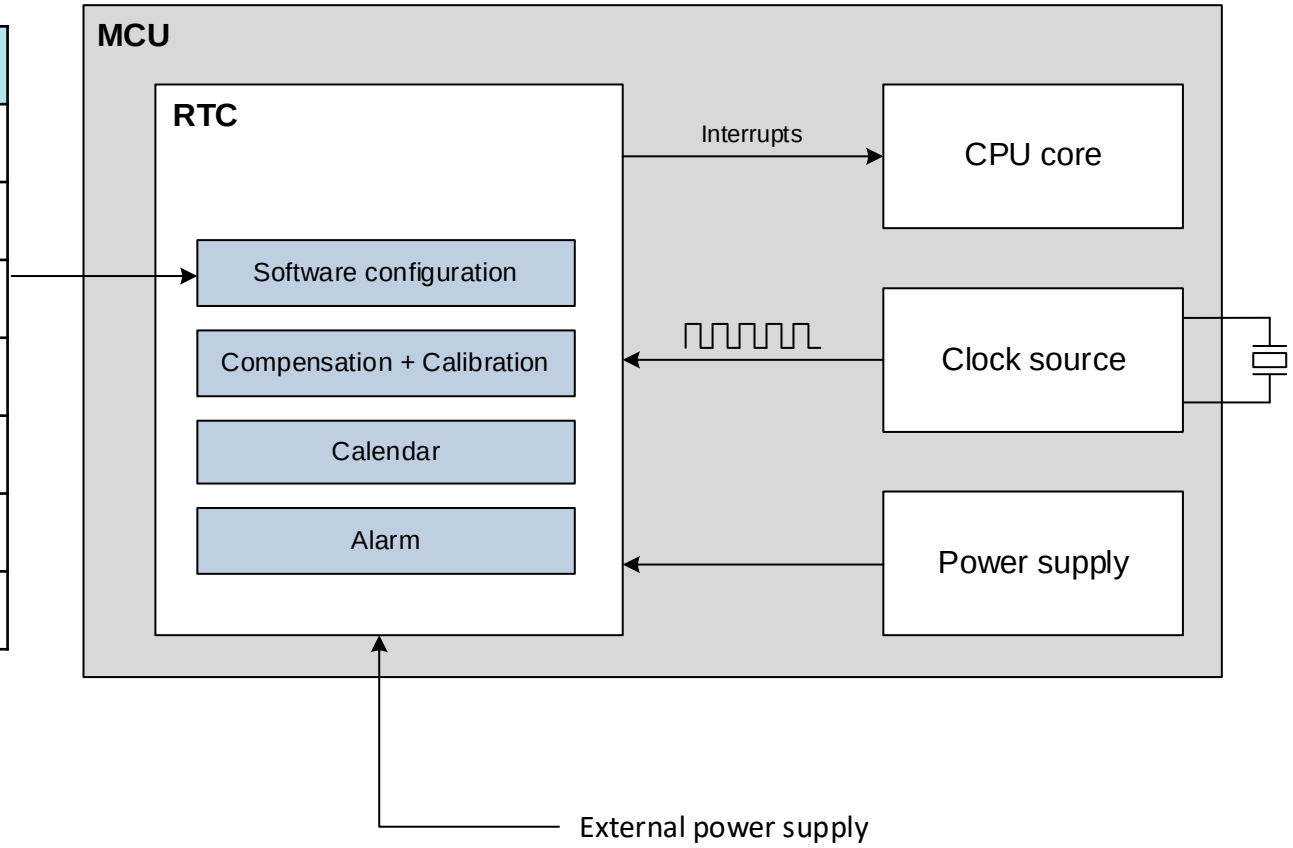
RTC

- ⊕ Convenient date formats
- ⊕ Calendar and alarm features
- ⊕ Calibration and temperature compensation



RTC block diagram

RTC Configuration	Value
Year	0x2022
Month	0x07
Day	0x19
Day of the week	0x02
Hour	0x11
Minute	0x59
Second	0x30



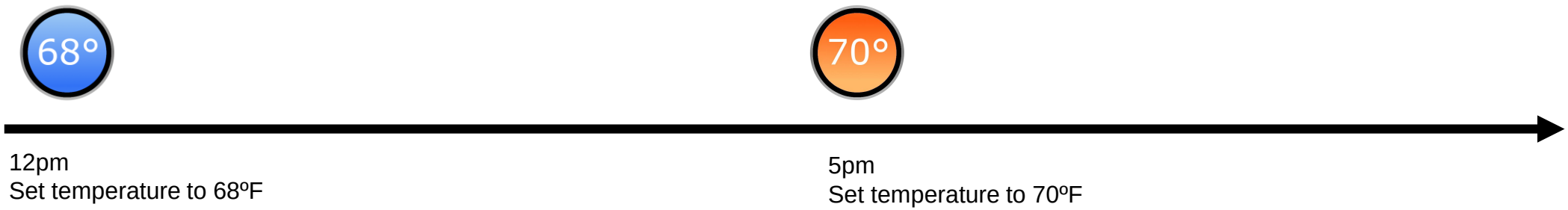
RTC calendar

- RTC calendar functionality allows an application to track the date and time in a human readable format
- Date and time can be stored in convenient formats for processing such as binary coded decimal (BCD).
- Logging example:
  1. Application detects logable event
  2. Read date and time from RTC
  3. Write data and timestamp to nonvolatile memory
  4. Application goes back to sleep

RTC Configuration	Value
Year	0x2022
Month	0x07
Day	0x19
Day of the week	0x02
Hour	0x11
Minute	0x59
Second	0x30

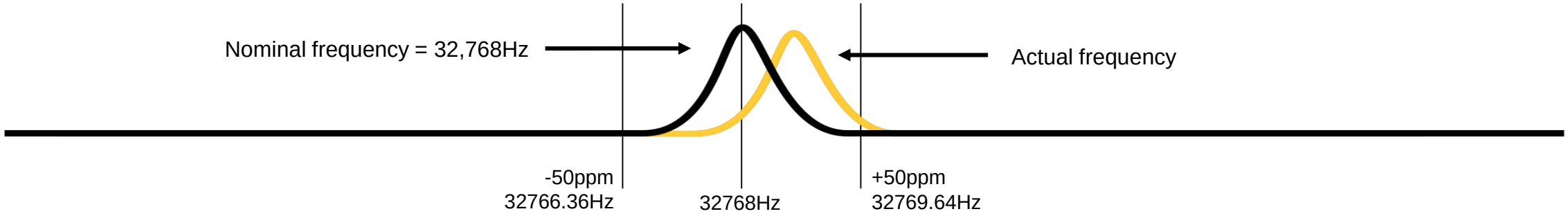
RTC alarm

- RTC alarm functionality allows an application to perform an action at a specific date and time
- RTC can fire an interrupt to wake the MCU from sleep mode
- Thermostat example:
  1. Application is in sleep mode
  2. RTC fires an interrupt at a predefined date and time
  3. Application performs task (*change the set temperature)*
  4. Application goes back to sleep



RTC clock source considerations

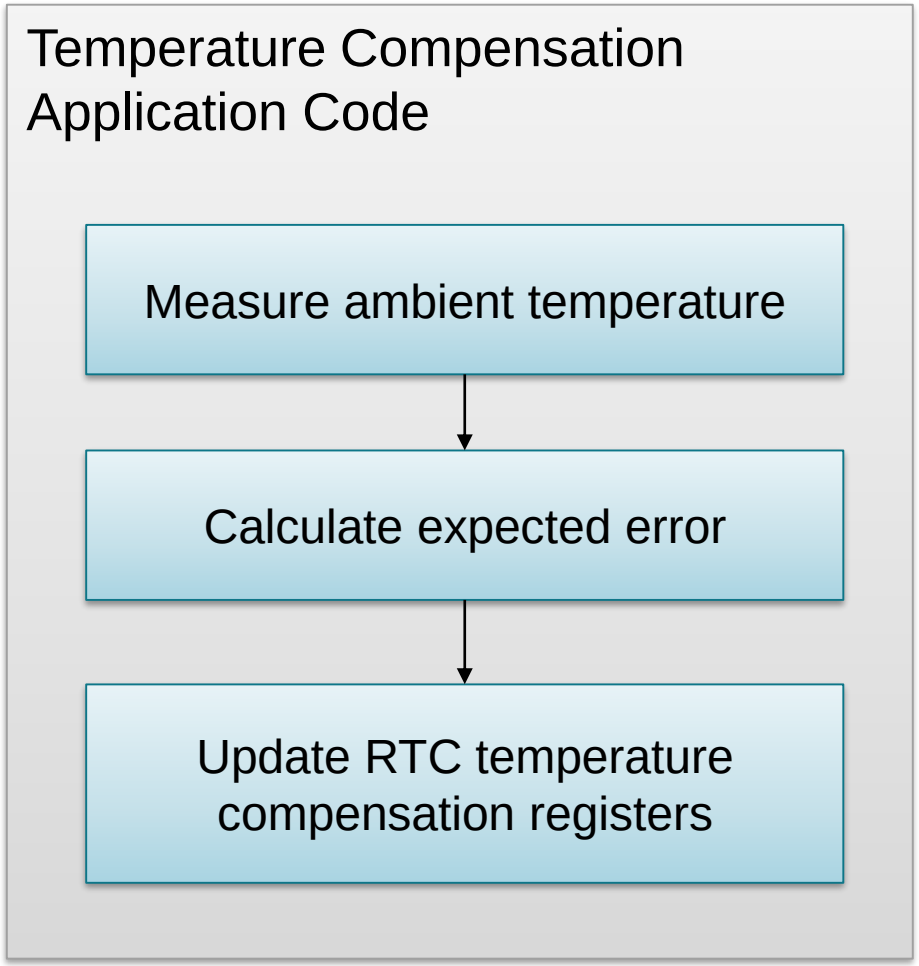
- RTC accuracy is determined directly by the accuracy of it's clock source.
- Crystal oscillators have offset error. Example: ±50ppm 32,768Hz crystal.
  - Worse case error over one day: $(seconds\ in\ a\ day) \times \frac{50ppm}{10^6} \approx 4.3\frac{seconds}{day}$



- Offset calibration helps reduce error
  1. Measure units RTC clock error against a high precision source
  2. Update RTC registers with measured offset

RTC temperature compensation

- Crystals drift with changes in temperatures
- Temperature compensation can help reduce error

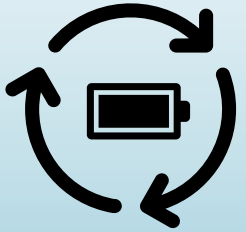


RTC power sources

RTC always needs power

Option 1 – System provides constant power to RTC

- \+ Can be sufficiently low-power
- − Fails if system looses power

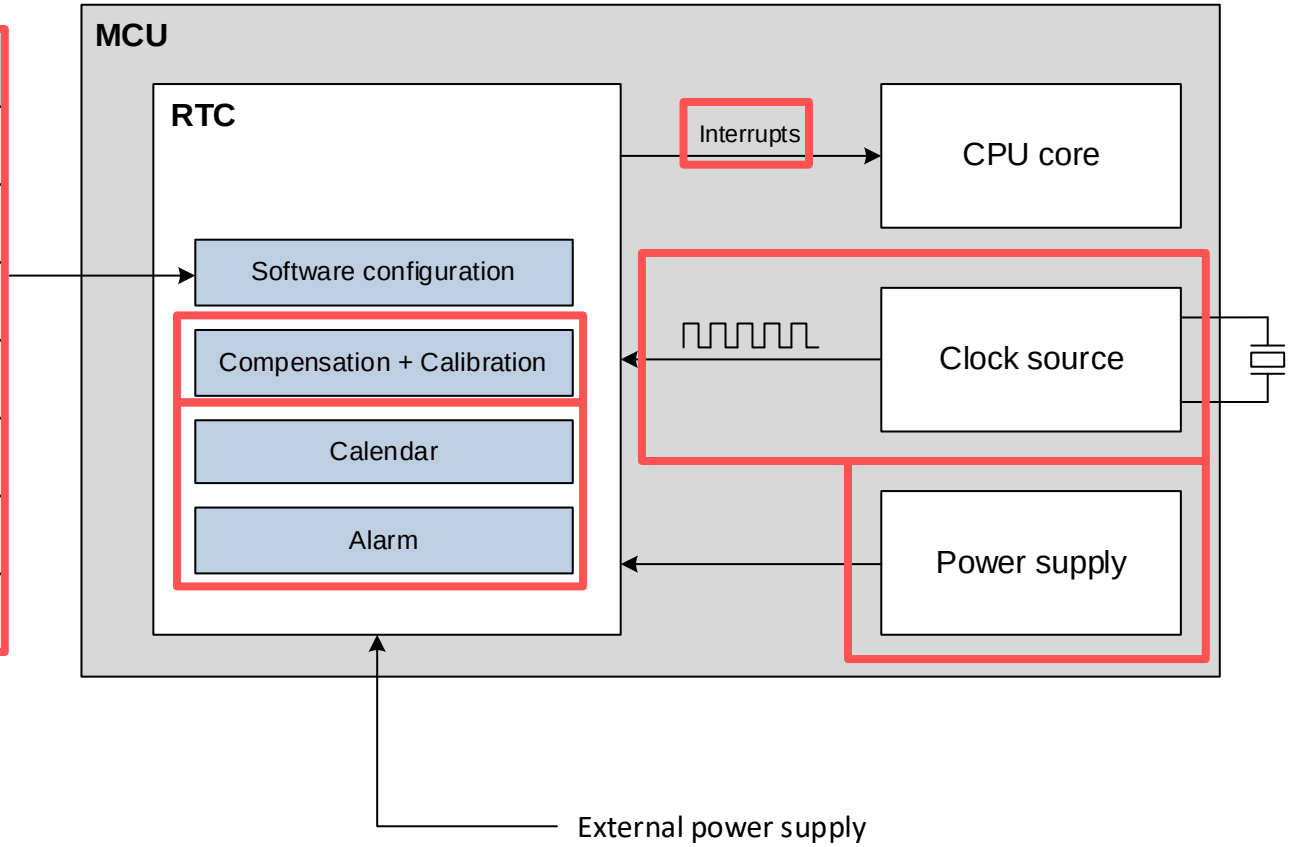


Option 2 – RTC can switch to backup power supply

- \+ Preserves time in unstable power environments
- − Requires additional circuitry

RTC review

RTC Configuration	Value
Year	0x2022
Month	0x07
Day	0x19
Day of the week	0x02
Hour	0x11
Minute	0x59
Second	0x30



To find more TI microcontroller technical resources and search products, visit **ti.com/microcontrollers**.