TOADSTOOL

GETTING STARTED

With the **Toadstool *RTC-MCP CAP***

!Crash Bang
prototyping

www.crash-bang.com
@crashbang_proto

open source hardware

Meet the Toadstool Family

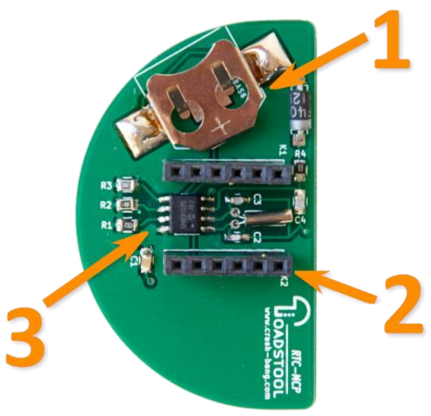
The **Toadstool** ***RTC-MCP*** ***CAP*** is part of a larger group of open source prototyping boards and breakout modules (called *Caps*). This family was designed to let you quickly and easily prototype your microcontroller-based projects. The **Toadstool** ***RTC-MCP*** is one of a selection of *Caps* compatible with the full range of Toadstool prototyping boards, and contains the MCP79400 RTC (real time clock) chip from Microchip[1].

Prototyping Boards vs *Caps*

The **Toadstool** ***RTC-MCP CAP*** is a breakout module – a small module containing specific functionality that connects directly to a Toadstool prototyping board. This module houses an RTC chip. The *Cap* is designed to interface with the Toadstool prototyping boards without obscuring or interfering with the main I/O pins of the prototyping board.

In contrast, Toadstool Prototyping Boards (eg. the Toadstool *Mega328*) are larger boards that house a microcontroller. There are a wide range of prototyping boards, containing microcontrollers from a number of manufacturers. They plug directly into a breadboard to allow for fast and simple prototyping.

Overview of the Toadstool *RTC-MCP*



1. Backup Battery Holder (CR1216, CR1220)
2. *Cap* header: Connect this to a Toadstool prototyping board
3. MCP79400 RTC Chip

Connecting to a Toadstool prototyping board

The ***RTC-MCP*** module connects quickly and easily to any of the Toadstool prototyping boards.

1. Ensure that the parent Toadstool prototyping board is not connected to a power source.
2. Align the semi-circular shape of the *Cap* with the semi-circular shape of the Toadstool prototyping board.

3. Gently, but firmly, insert the header pins into the matching headers on the Toadstool prototyping board.
4. Begin prototyping!

Important: If the module is inserted back-to-front, and power applied to the parent board, both the module and the parent Toadstool prototyping board could be irreparably damaged.

Using the *RTC-MCP*

The MCP79400 is a real time clock module with an I^{2}C serial interface. The module stores date (year, month, day, day of week) and time (hour, minute, second) in 12 or 24-hour format. Leap years are handled up to 2399. Once the date/time is set, and the oscillator started, the application is able to read the time from the module. A CR1216/CR1220 coin cell can be inserted in the battery-holder if backup power is required.

All date and time information is stored in Binary-Coded Decimal (BCD), and needs to be converted to/from decimal.

I^{2}C Addressing

In line with the I^{2}C protocol, the module has an address which the microcontroller must send in order to initiate communication. This allows multiple I^{2}C devices to be daisy-chained, yet separately addressed. The ***MCP79400*** module has a fixed base address of `0b1101111`.

I^{2}C Communication

It is beyond the scope of this Getting Started guide to describe the I^{2}C protocol, or to reiterate the detailed descriptions in the MCP79400 datasheet. However, an overview of basic functionality is useful. Normally, the following steps would be followed in order initialise the module, set the time and read the time:

Initialise the Module:

1. Initialise the I^{2}C interface on the microcontroller

2. Enable the Backup Battery if required (VBATEN bit of RTCWKDAY register)
3. Start the Oscillator (ST bit of RTCSEC register)
4. Check the Oscillator started (OSCRUN bit of RTCWKDAY register)

Set the Time:

1. Stop the Oscillator (ST bit of RTCSEC register)
2. Send a Start Condition
3. Set the time in order: Year, Month, Date, Day of Week, Hour (including 12/24-hour format), Minute, Second. Send a "repeated start" before each element.
4. Start the Oscillator – this can be done when setting the Seconds (ST bit of RTCSEC)
5. Send a Stop Condition
6. Check the Oscillator started (OSCRUN bit of RTCWKDAY register)

Read the Time:

1. Send a Start Condition
2. Retrieve the time in order: Year, Month, Date, Day of Week, Hour, Minute, Second. The MCP79400 uses a buffer to ensure accuracy - send a "repeated start" before each element.
3. Send a Stop Condition

A Sample Project

A sample project has been created to illustrate the use of the ***RTC-MCP*** together with the **Toadstool *Mega328*** (based on the Atmel ATmega328P microcontroller). The project can be downloaded from the Toadstool Github repository[2].

The project sets a hard-coded time on the RTC, and then reads this back every few seconds, displaying it over the UART. The project was written in Atmel Studio 6.2, although the underlying code should be portable to other IDEs using the GCC toolchain.

Where to From Here?

There are many more Toadstool modules in the pipeline to make your prototyping life easier, as well as a host of online resources online to take you further: refer to our website[3] to keep building.

[1] **Microchip's MCP79400 product page:** http://www.microchip.com/MCP79400
[2] **Github Repository:** https://github.com/CrashBangProto
[3] **Toadstool on the Internet:** http://www.crash-bang.com/toadstool