

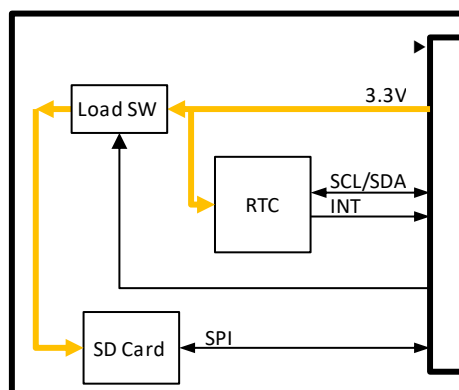
## AZ02 RTC&microSD

### 1. 概要

NXP の Real-Time Clock PCF8523TK および microSD カードスロットを搭載したリーフである。RTC はアラームやタイマーにより MCU に割り込み信号を出力することが可能である。microSD カードは SPI でアクセスする。

### 2. リーフ仕様

#### 2-1. ブロック図



#### 2-1. 電源仕様

Symbol	Parameter	Condition	Min.	Typ.	Max.
Vdd	Power Supply Voltage	—	2.7V	3.3V	3.6V
Idd	Operating current	Active	-	3.4uA	-
		Sleep	-	1uA	-

※microSD の消費電流は除く

#### 2-2. 主要部品

リーフ実装の主要部品は次の通り。

部品番号	部品名	型番	ベンダー名	備考
IC460	Real-Time Clock	PCF8523TK	NXP	—
IS260	micro SD Card slot	105162-0101	molex	—

#### 2-3. 外観

リーフ外観と実装部品は次の通り。

表面	裏面
<p>micro SD Card slot (IS260)</p>	

## 2-4. ピンアウト

Name	Function
SCL	I2C 通信クロック
SDA	I2C 通信データ
D6	SDPON : MicroSD カード電源オン H:電源オン L : 電源オフ
D10	CS : SPI チップセレクト
D13	SCLK : SPI クロック
D11	MOSI : SPI マスター出力データ
D12	MISO : SPI マスター入力データ
3V3	3.3V 入力
GND	GND

## 3. Real-Time Clock (PCF8523TK)仕様

### 3-1. 概要

項目	内容
Type	Real-Time Clock (RTC) and calendar
Interrupt	Alarm and timer
使用水晶振動子	FC-12M 32.768000kHz 7.0 +20.0-20.0
Interfaces	I2C

### 3-2. 電気的特性

#### 3-2-1. 最大定格

Parameter	Value
Operating Temperature	-40°C to +105°C
Maximum Operation Voltage	Vin 6.5V

#### 3-2-2. 定格

Symbol	Parameter	Condition	Min.	Typ.	Max.
Vdd	Operating voltage	power management function active	1.8V	-	5.5V
Iq	Quiescent current	I2C-bus active; fscl=1000kHz	-	-	200uA
		I2C-bus inactive; interrupts disabled	-	150nA	500nA

### 3-3. データシートリンク先

[https://www.nxp.com/jp/products/analog/signal-chain/real-time-clocks/ic-real-time-clocks-rtc/real-time-clock-rtc-and-calendar:PCF8523?lang=jp&lang\\_cd=jp&](https://www.nxp.com/jp/products/analog/signal-chain/real-time-clocks/ic-real-time-clocks-rtc/real-time-clock-rtc-and-calendar:PCF8523?lang=jp&lang_cd=jp&)

## 4. 主な関数とライブラリ

### 4-1. RTC の制御

include file:RTCLib.h

<https://github.com/adafruit/RTCLib>

関数	概要
RTC_PCF8523 rtc	PCF8523 用のインスタンスを作成 【パラメータ】 rtc:インスタンス名 (rtc)
rtc.begin()	RTC との通信を初期化。 【パラメータ】 なし 【戻り値】 true: 成功 false:失敗
rtc.initialized()	RTC を初期化。 【パラメータ】 なし 【戻り値】 true: 成功 false:失敗
rtc.adjust(DateTime (date, time))	RTC に日時を設定。 【パラメータ】 DateTime(日時) year(16byte) month(8byte) day (8byte) hour (8byte) minute (8byte) second(8byte) dayOfWeek(8byte) 【戻り値】 なし
rtc.now()	RTC から現在の日時を読み込む。 【パラメータ】 なし 【戻り値】 DateTime(日時) year(16byte) month(8byte) day (8byte) hour (8byte) minute (8byte) second(8byte) dayOfWeek(8byte)

## 5. microSD 仕様

### 5-1. 概要

microSD はコネクタのみで IC 等は実装せず MCU と直接接続される。

### 5-2. SD カードの制御

include:SD.h(Arduino IDE Standard Libraries)

spi.h(Arduino IDE Standard Libraries)

関数	概要
SD.begin()	ライブラリと SD カードを初期化し、SPI バスとチップセレクト(SS)ピンを使えるようにする。SPI で使うピンのデフォルトはピン 10。 【構文】 SD.begin(clock,cspin) 【パラメータ】 clock : SPI のクロックを設定する Leaf の推奨値は 2500000 cs핀: SD カードのチップセレクト端子に接続するピン。デフォルトは SPI の SS 端子 【戻り値】 true: 成功 false: 失敗
SD.exists()	指定したファイルが存在しているか否かを調べる。 【構文】 SD.exists(filename) 【パラメータ】 filename: 存在を確認したいファイルの名前。パスを指定可能(スラッシュ '/' 区切り) 【戻り値】 true: 存在している false: 存在しない
SD.mkdir()	ディレクトリを作成。 【構文】 SD.mkdir(filename) 【パラメータ】 filename: 作成したいディレクトリの名前。パスを指定可能(スラッシュ '/' 区切り) 【戻り値】 true: 成功 false: 失敗

SD.open()	<p>ファイルを開く。対象となるファイルが存在しないとき、書き込みモードならば、そのファイルを新規に作成。</p> <p>【構文】</p> <pre>SD.open(filename) SD.open(filename, mode)</pre> <p>【パラメータ】</p> <p>filename: 開きたいファイルの名前。パスを指定可能(スラッシュ '/' 区切り)</p> <p>mode (オプション): モードを指定します。デフォルトは FILE_READ</p> <p>FILE_READ: 読み込み(read)</p> <p>FILE_WRITE: 読み込みと書き込み</p> <p>【戻り値】</p> <p>開いたファイルを参照する File オブジェクトを返す。</p> <p>開くことができなかった場合は false を返す。</p>
SD.remove()	<p>ファイルを削除。</p> <p>【構文】</p> <pre>SD.remove(filename)</pre> <p>【パラメータ】</p> <p>filename: 削除したいファイルの名前。パスを指定可能(スラッシュ '/' 区切り)</p> <p>【戻り値】</p> <p>true: 成功</p> <p>false: 失敗</p> <p>ファイルが存在しなかったときの戻り値は不定。</p>

### 5-3. 省電力制御

SD カードは、ロードスイッチにより電源をオフできる回路を実装する。

## 6. 変更履歴

Rev A1.0: 2020 年 1 月初版