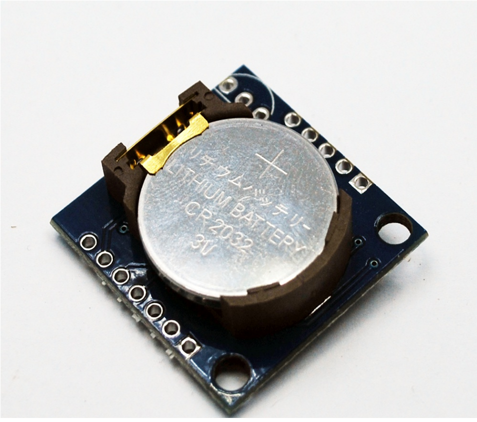
TinyRTC   I2C Module

Technical Manual Rev 1r0

Arduino TinyRTC I2C Real Time Clock with DS1307 I2C real time clock IC and 24C32 with 32KB I2C EEPROM storage. It has a DS18B20 temperature sensor pad connections on board. With LIR2303 rechargeable lithium battery.



FEATURES:

- *Gizduino and Arduino Compatible*
- *Arduino IDE software*

GENERAL SPECIFICATIONS:

- **Input Supply:** ***5V DC***
- **On-board ICs:** ***DS1307 and 24C32***
- **Pad on-board:** ***for DS18B20***
- **Battery*: LIR2303 Rechargeable Lithium***
- **PCB Dimensions: 25mmx28mmx8.4mm**

*DS18B20 Specifications (*optional):*

- **Input Supply:** ***3.0 ~5.5V DC***
- **Features:** ***Waterproof***
- **Temperature Range:** ***-55C to +125C***
- ***Accuracy: (+/-0.5C) from -10C to +85C***
- **Interface: 1 wire**

DS18B20 Temperature sensor
Pads connection

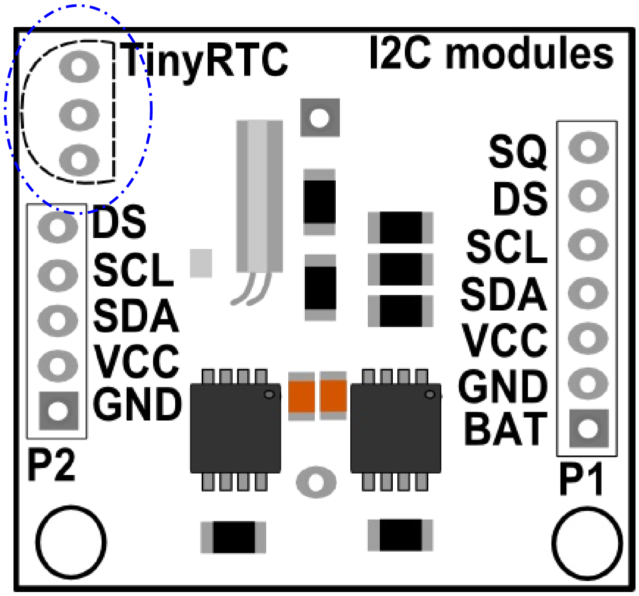


Figure 1. Major parts presentation of TinyRTC I2C modules

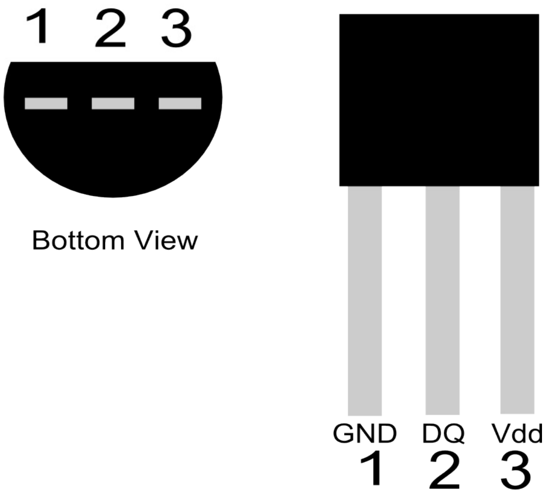


Figure 2. Major parts presentation of DS18B20 Temperature sensor

Table 1. P1 and P2 Pin Descriptions

Reference Pin	Name	Descriptions
BAT	*Battery voltage*	*To monitor the battery voltage/NC*
GND	*Ground*	*Ground*
VCC	*Input Supply*	*+5V power the module and charge the battery*
SDA	*I2C Data*	*I2C data for the RTC*
SCL	*I2C clock*	*I2C clock for the RTC*
DS	*DS18B20*	*Temperature sensor output, 1 wire interface*
SQ	*Square wave*	*Output, Normally not used.*

Table 2. I2C wire in different boards

Board	I2C/TWI pins
UNO	*(D18) A4 (SDA), (D19) A5 (SCL)*
GIZDUINO PLUS	*D25 (SDA), D24 (SCL)*
GIZDUINO X	*(D18) A4 (SDA), (D19) A5 (SCL)*
MEGA2560	*D20 (SDA), D21 (SCL)*
LEONARDO	*D2 (SDA), D3 (SCL)*
DUE	*D20 (SDA), D21 (SCL), SDA1, SCL1*

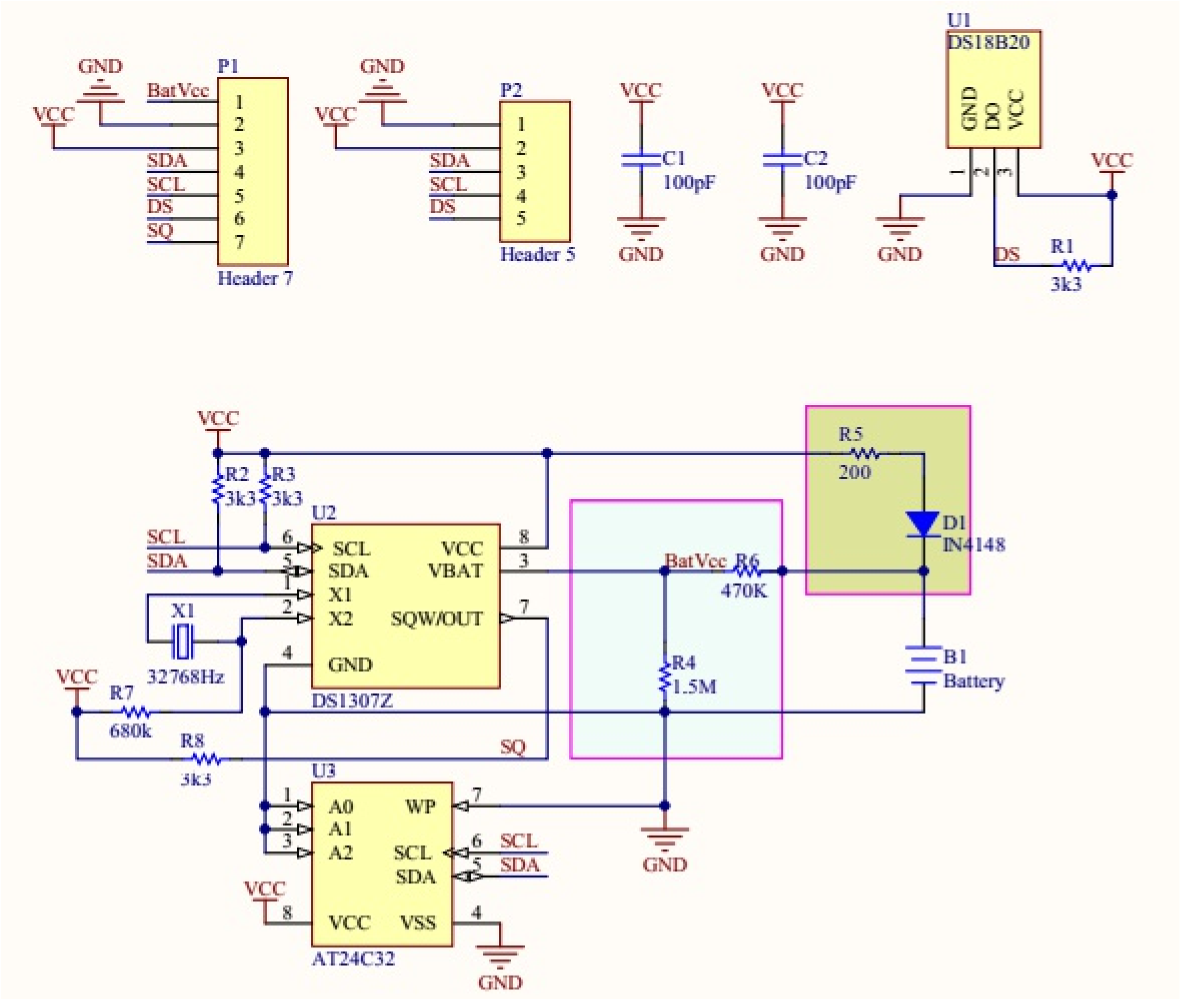


Figure 3. Schematic Diagram of TinyRTC I2C modules

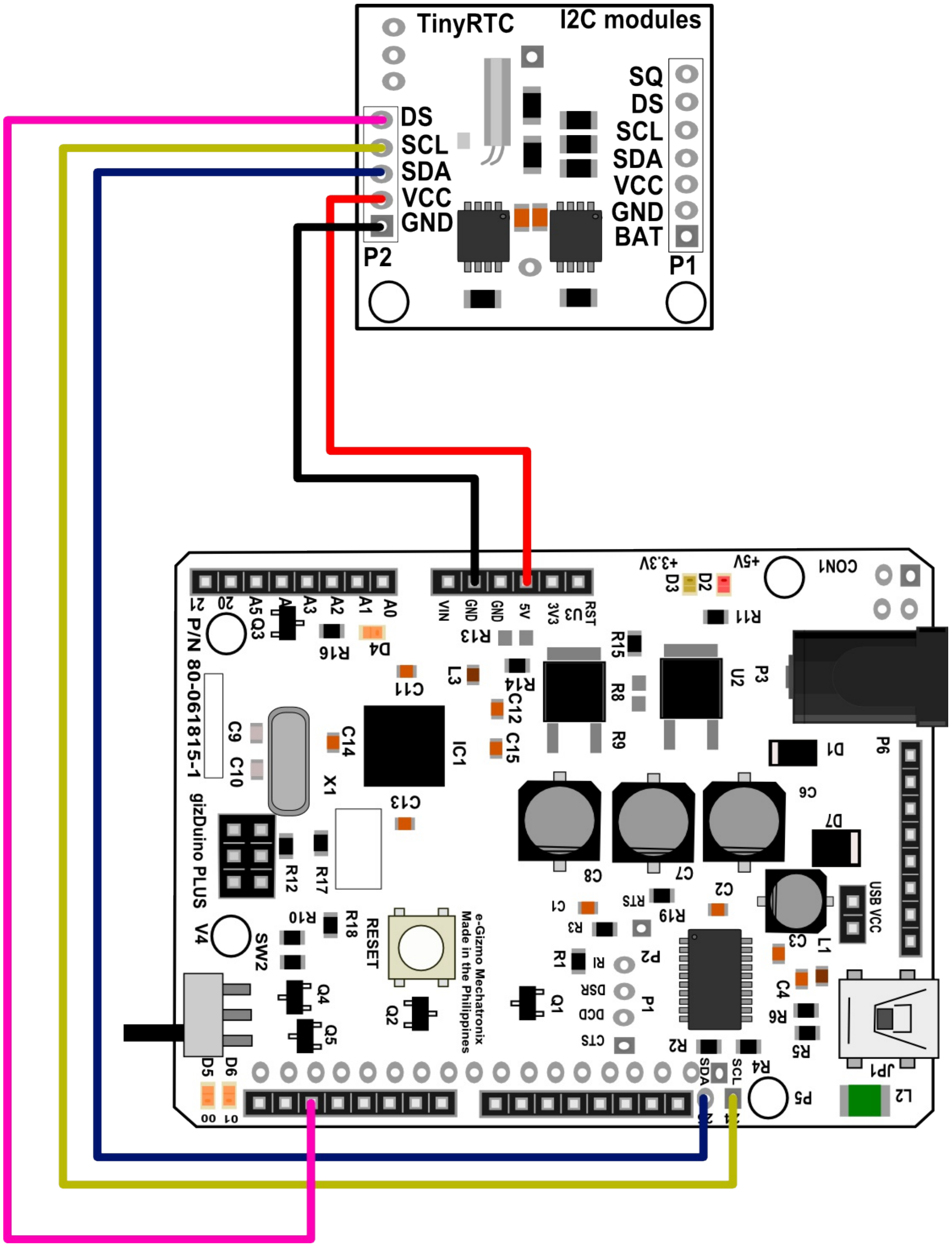


Figure 4. Sample Applications of TinyRTC I2C modules to Gizduino PLUS ATmega644P.

Add the two libraries:

1. Wire Library
2. RTClib Library

My Documents>Arduino>libaries>1,2

Open the Sample/Example code from RTClib. "ds1307.ino"

```
// Date and time functions using a DS1307 RTC connected via I2C and Wire lib
#include <Wire.h>
#include "RTClib.h"

#if defined(ARDUINO_ARCH_SAMD)
// for Zero, output on USB Serial console, remove line below if using programming port to program the Zero!
   #define Serial SerialUSB
#endif

RTC_DS1307 rtc;

char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

void setup () {

#ifndef ESP8266
  while (!Serial); // for Leonardo/Micro/Zero
#endif

  Serial.begin(57600);
  if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }

  if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running!");
     // following line sets the RTC to the date & time this sketch was compiled
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
     // This line sets the RTC with an explicit date & time, for example to set
    // January 21, 2014 at 3am you would call:
    // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
  }
}

void loop () {
    DateTime now = rtc.now();

    Serial.print(now.year(), DEC);
    Serial.print('/');
    Serial.print(now.month(), DEC);
    Serial.print('/');
    Serial.print(now.day(), DEC);
    Serial.print(" (");

    Serial.print(") ");
    Serial.print(now.hour(), DEC);
    Serial.print(':');
    Serial.print(now.minute(), DEC);
    Serial.print(':');
    Serial.print(now.second(), DEC);
    Serial.println();

}
```

For DS18B20 Temperature sensor Sample code.

Add the two libraries:

1. Dallas Temperature Library
2. OneWire Library (latest)

My Documents>Arduino>libaries>1,2

Open the Sample/Example code from OneWire. "DS18x20_Temperature.pde"