

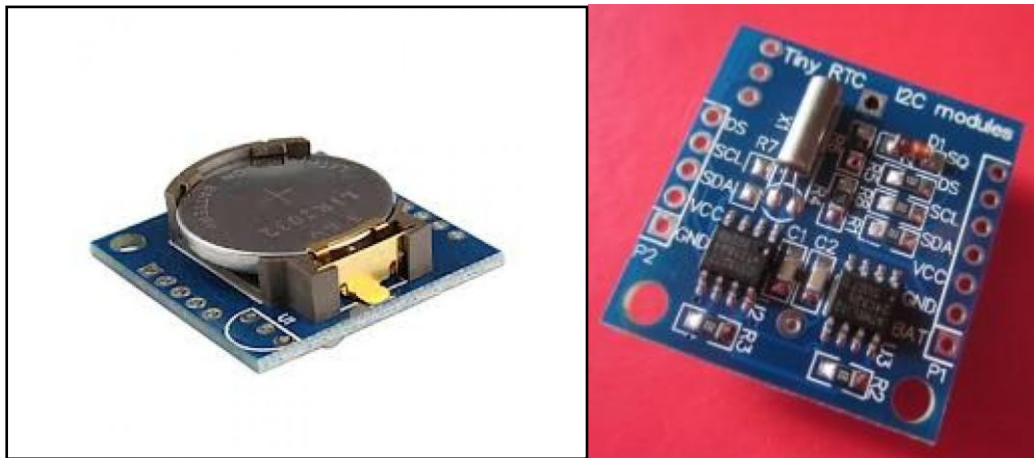
## The steps to set a RTC for the BBB.

As the BeagleBone Black (BBB) does not include a battery-backed Real Time Clock module, we need to connect a RTC module to make its system time correct. We follow below steps to set it working for BBB.

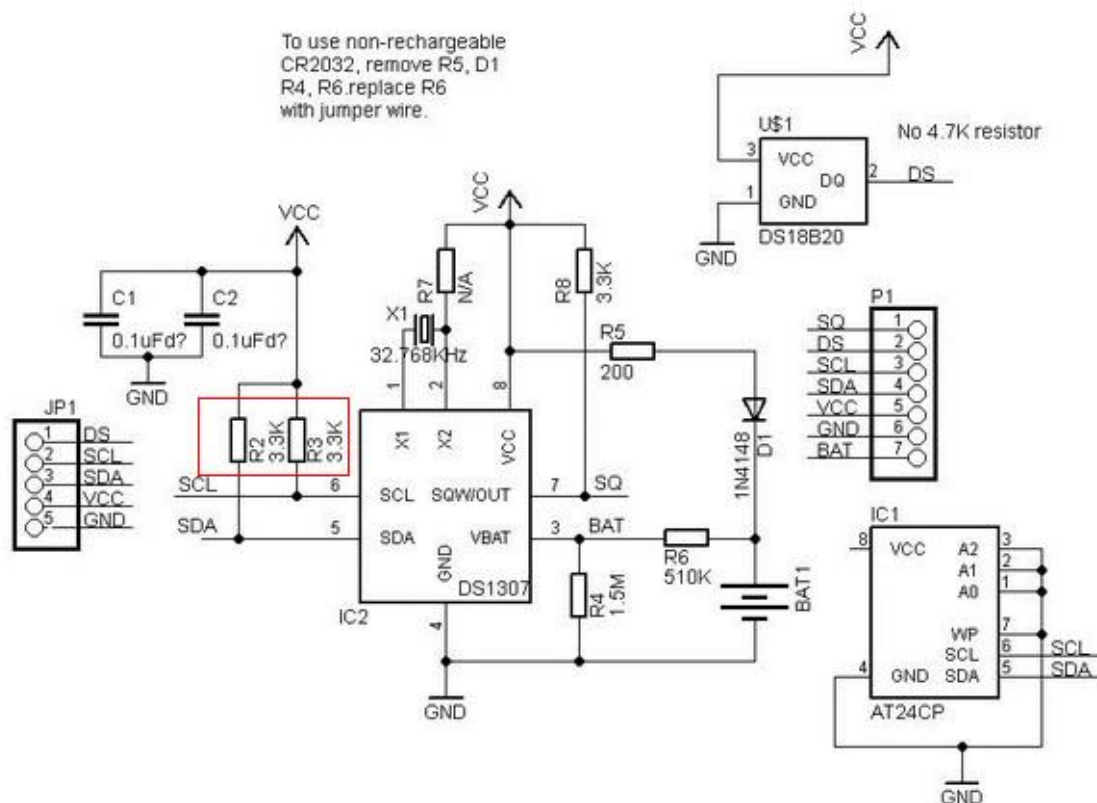
### Step 1: Changed RTC I2C DS1307 Module for BBB:

Currently we use RTC I2C DS1307 Module Including Coin Cell Battery to provide time to BBB. This is the description of RTC I2C DS1307 Module:

Tiny RTC I2C DS1307 Real Time Clock Module ARM PIC for Arduino



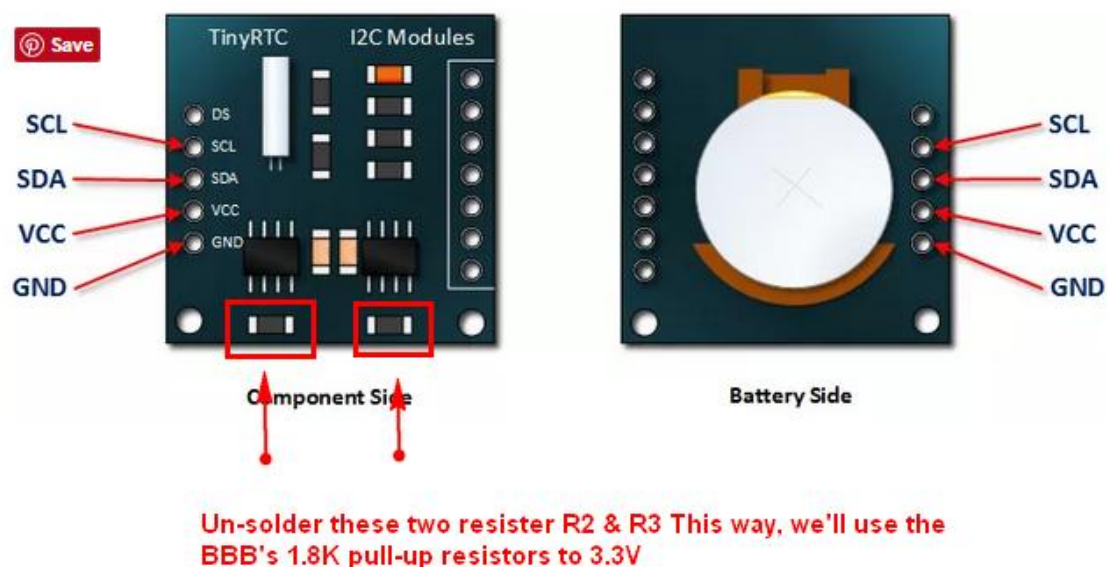
RTC I2C DS1307 circuit diagram



## RTC I2C DS1307 Module Features:

- The DS1307 I2C Real Time Clock chips (RTC)
- I2C EEPROM memory 24C32 32K
- To adopt LIR2032 rechargeable lithium battery, and with the charging circuit
- Solve the problem DS1307 with battery backup cannot read and write.
- Fully charged, can provide the DS1307 timing.
- Compact design, 27mm \* 28mm \* 8.4mm
- Leads to the DS1307 clock pin for the MCU to provide the clock signal.

As the RTC I2C DS1307 Module was designed for the Arduino, when building the kit, we need to remove the pull up resistors (R2 and R3 which in the red box shown in the circuit diagram). we force the RTC to communicate at 3.3V instead of 5V, which is better for the BBB. The resistor needs to be unsoldered is shown below:



## Step 2: Connect the RTC to BBB for verification.

After removed the pull up resistor of the RTC chip, the next step is connect to BBB for detection. The wiring work is simple:

1. Connect **VCC** on the RTC I2C DS1307 to the **P9\_5** (VCC 5V) or **P9\_7** (SYS 5V) pin on the BBB. **NOTE: The P9\_5 VCC 5V pin will only be powered if a 5V adapter is plugged in to the barrel jack. If powering over USB use the P9\_7 (SYS 5V) pin instead!**
2. Connect **GND** on the breakout board to the **P9\_1** (GND) pin on the BBB.
3. Connect **SDA** on the breakout board to the **P9\_20** pin of the BBB
4. Connect **SCL** on the breakout board to the **P9\_19** pin of the BBB

The wire connection is shown below:

[illegible]

### Step 3: Synchronize RTC Time with


After finished wired the RTC chip module wired up and verified that we can see the module with i2cdetect, we can set up the module.

Now, execute the following to add the RTC chip to new device list:

```
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
```

After hooked the address to the BBB new device list, we can run the program to check the current time of the DS1307 module:

```
hwclock -r -f /dev/rtc1
```



```
root@beaglebone:~# hwclock -r -f /dev/rtc1
Sat Jan 1 00:03:24 2000 0.000000 seconds
root@beaglebone:~#
```

If this is the first time the module has been used it will report back Jan 1 2000, so we will need to set the time. The quickest way to set the time on the BeagleBone Black is to execute the following (The BBB need to connect to the internet):

```
/usr/bin/ntpdate -b -s -u pool.ntp.org
```

Now that the system time is set correctly, we can execute the following to write the system time to the DS1307:

```
hwclock -w -f /dev/rtc1
```

We can also verify whether it was set correctly by executing the following command to read the date and time from the DS1307 RTC:

```
hwclock -r -f /dev/rtc1
```

Its time should be set to the current time now. As shown below:

A terminal window titled 'tmp - ssh - 86x27' with tabs for 'bash', 'ssh', and 'ssh'. The prompt is 'root@beaglebone:~#'. The command 'hwclock -r -f /dev/rtc1' has been executed, resulting in the output 'Wed Jul 17 18:49:12 2013 0.000000 seconds'. The prompt is now 'root@beaglebone:~#'.

```
root@beaglebone:~# hwclock -r -f /dev/rtc1
Wed Jul 17 18:49:12 2013 0.000000 seconds
root@beaglebone:~#
```

#### Step 4: Create service that will run each time when BBB boot up.

To start, create a directory and script that will be executed:

```
mkdir /usr/share/rtc_ds1307
```

```
nano /usr/share/rtc_ds1307/clock_init.sh
```

Now, with the nano text editor open, copy the following into the clock\_init.sh script:

```
#!/bin/bash
sleep 15
echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device
hwclock -s -f /dev/rtc1
hwclock -w
```

Next, we'll create a service that will get started on boot, and execute the script we just created:

```
nano /lib/systemd/system/rtc-ds1307.service
```

Copy the following contents into that file, and save it:

```
[Unit]
Description=DS1307 RTC Service

[Service]
Type=simple
WorkingDirectory=/usr/share/rtc_ds1307
ExecStart=/bin/bash clock_init.sh
SyslogIdentifier=rtc_ds1307

[Install]
WantedBy=multi-user.target
```

After saving the file, we'll need to actually enable the service so it starts each time as the system boots:

```
systemctl enable rtc-ds1307.service
```

The way to manually start and stop the service:

```
systemctl start rtc-ds1307.service
```

```
systemctl stop rtc-ds1307.service
```

After reboot the BBB, the RTC I2C DS1307 Module can work normally now.

Reference:

<https://learn.adafruit.com/adding-a-real-time-clock-to-beaglebone-black/wiring-the-rtc>