VIRTUAL COURSE
BUILD YOUR OWN DATA LOGGER

WILDLABS.NET
[ The conservation technology network ]

FREAKLABS
MODULE 3-6

INTERRUPTS

LAB 6C:

WAKE ME UP BEFORE YOU GO-GO
Goal

• Get an understanding of how RTC interrupts work
• Set timer to periodically interrupt our device
• Handle interrupts from real time clock
Why Is This Important?

- The RTC plays a key role in power savings strategy
- Our device will sleep most of the time, and wake up periodically to take measurements using the RTC
- Without an RTC, our device can use only sleep a maximum of 8 seconds max, using watchdog timer
- With an RTC, our device can sleep hours using our RTC timer functions, and days using our RTC alarm functions
What Do I Need to Know?

- For RTC interrupt, use interrupt number 0 and pinNumber 2 (see below)

<table>
<thead>
<tr>
<th>Interrupt Source</th>
<th>Interrupt Number</th>
<th>Pin Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Time Clock</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>PIR Motion Sensor/Aux Intp</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Pushbutton 0</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>
What Do I Need to Know?

• **Timer functions**
  • `rtc.setTimer(count, frequency, isPulsed)`
    • count: value from 0 to 255 to count down
    • frequency
      • `TMR_1Hz`: countdown in seconds
      • `TMR_1MIN`: countdown in minutes
    • isPulsed: output interrupt pulse or level
      • we don’t really care, but use false as a rule of thumb
  
  • `rtc.resetTimer()`
    • good for re-using timer function after it finishes countdown
    • clears timer interrupt and keeps timer enabled
    • resets timer
  
  • `rtc.clearTimer()`
    • clears and disables timer
What Do I Need to Know?

- Alarm functions
  - `rtc.setAlarm(min, hour, day, weekday)`
    - Looks for a match on the RTC
    - 99 is a wildcard tells the function to ignore the parameter.
    - Eg. `rtc.setAlarm(0, 99, 99, 99)` triggers an alarm every hour
What Do I Need to Know?

```javascript
rtc.setAlarm(min, hour, day, weekday)
```

- **Minutes** as per hour (0-59)
- **Hours** as per 24 hour clock (0-23)
- **Day** as per day of the calendar month (1–31)
- **Week Day**
  - Sunday = 0
  - Monday = 1
  - Tuesday = 2
  - Wednesday = 3
  - Thursday = 4
  - Friday = 5
  - Saturday = 6
What Do I Need to Know?

• **Examples**
  
  **Hourly**
  - `rtc.setAlarm(00, 99, 99, 99)` - triggers alarm every hour on the hour
  
  **Daily**
  - `rtc.setAlarm(32, 12, 99, 99)` - triggers alarm day at 12.32pm
  
  **Weekly**
  - `rtc.setAlarm(5, 18, 99, 5)` – trigger alarm every Friday at 6.05pm
  
  **Day of the Month**
  - `rtc.setAlarm(45, 02, 15, 99)` – trigger alarm at 2.45am on the 15th of every month
  
  **Another option**
  - Use a counter for weekly or monthly alarms – it’s easier!
  - Eg. For every two weeks, set an alarm for every week, and have a counter that increments each week. When the counter reaches 2, the interrupt triggers, and counter is reset.
What Do I Need to Know?

- Alarm functions

  - rtc.resetAlarm()
    - clears alarm interrupt
    - keeps alarm enabled
    - needs to be called if same alarm will be reused

  - rtc.clearAlarm()
    - clear alarm interrupt
    - disable alarm
```cpp
#include <Rtc_Pcf8563.h>

Rtc_Pcf8563 rtc;

int intpNumRtc = 0;
volatile int rtcFlag = 0;

void setup()
{
    Serial.begin(57600);
    Serial.println("Module 3, Submodule 6, Lab 6c - RTC Interrupt");

    attachInterrupt(intpNumRtc, isrRtc, FALLING);
    rtc.setTimer(1, TMR_1MIN, false);
}
```
void loop()
{
    if (rtcFlag == 1)
    {
        rtcFlag = 0;
        rtc.resetTimer();
        Serial.print("Timer interrupt: ");
        Serial.print(rtc.formatDate());
        Serial.print(", ");
        Serial.println(rtc.formatTime());
    }
}

void isrRtc()
{
    rtcFlag = 1;
}
Timer interrupt. Time is: 01/21/2021, 13:34:00
Timer interrupt. Time is: 01/21/2021, 13:35:00
Timer interrupt. Time is: 01/21/2021, 13:36:00
Timer interrupt. Time is: 01/21/2021, 13:37:00
Timer interrupt. Time is: 01/21/2021, 13:38:00
Timer interrupt. Time is: 01/21/2021, 13:39:00
Timer interrupt. Time is: 01/21/2021, 13:40:00
Timer interrupt. Time is: 01/21/2021, 13:41:00
Timer interrupt. Time is: 01/21/2021, 13:42:00
Timer interrupt. Time is: 01/21/2021, 13:43:00
Timer interrupt. Time is: 01/21/2021, 13:44:00
Timer interrupt. Time is: 01/21/2021, 13:45:00
Timer interrupt. Time is: 01/21/2021, 13:46:00
Timer interrupt. Time is: 01/21/2021, 13:47:00
Timer interrupt. Time is: 01/21/2021, 13:48:00
Timer interrupt. Time is: 01/21/2021, 13:49:00
Timer interrupt. Time is: 01/21/2021, 13:50:00
Timer interrupt. Time is: 01/21/2021, 13:51:00
Timer interrupt. Time is: 01/21/2021, 13:52:00
Timer interrupt. Time is: 01/21/2021, 13:53:00
Timer interrupt. Time is: 01/21/2021, 13:54:00
Timer interrupt. Time is: 01/21/2021, 13:55:00
Timer interrupt. Time is: 01/21/2021, 13:56:00
COMING UP
Module 4:
Tying It All Together