**Documentation of extra feature**

As a special feature we implemented an alarm clock. Therefore we implemented a quartz for precise time counting and a speaker for the wake up noise. This setup allows for all kinds of time related applications, but due to the lack of time its currently used for a kitchen clock only.

**Software**

**Microcontroller**

*(The code is in the attachment in the form of a pdf)*

**Change code to fit quartz frequency**

The module “alarm.c” and its headerfile “alarm.h” have two tasks to perform, counting seconds and generate the alarm tone signal for the speaker.

Counting seconds is done with an external quartz (32.768kHz) acting as clock for the asynchronous timer/counter. With prescaler and compare value register the resulting frequency will be 1Hz and the interrupt service routine can be used to count seconds.

Waveform generation is realised with

**Android**

The alarm part of the Android app is yet quite simple. The affiliated activity contains a textView to display the set alarm time and a button to open a time picker dialog and set a new alarm time.

**public class** AlarmActivity **extends** AppCompatActivity **implements** TimePickerDialog.OnTimeSetListener, AlarmRequestTimerListener {  
  
 *// constants* **final private int ALARM\_UPDATE\_TIME\_MS** = 10000; *// = 10s  
  
 // alarm update timer* AlarmRequestTimer **timer**;  
  
 *// GUI instances* TextView **timeText**;  
 Button **timeButton**;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_alarm***);  
  
 *// GUI initialization* **timeText** = (TextView) findViewById(R.id.***timeText***);  
 **timeButton** = (Button) findViewById(R.id.***timeButton***);  
  
 **timeButton**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *// create and open a new time picker dialog* android.support.v4.app.DialogFragment timePicker =  
 **new** TimePickerFragment();  
 timePicker.show(getSupportFragmentManager(), **"timepicker"**);  
 }  
 });  
 }  
  
 @Override  
 **public void** onStart() {  
 **super**.onStart();  
 *// create a new timer to request the current alarm time in the set intervall* **timer** = **new** AlarmRequestTimer(**this**, **ALARM\_UPDATE\_TIME\_MS**);  
 }  
  
 @Override  
 **public void** onStop() {  
 **super**.onStop();  
 *// kill any existing timer on activity stop to avoid left behind threads* **try** {  
 **timer**.kill();  
 } **catch** (Exception e) {  
 *// necessary to avoid errors if the thread doesn’t exist* }  
 }  
  
 */\*\*  
 \* overrides the onTimeSet function of the OnTimeSetListener  
 \** ***@param view*** *\** ***@param hourOfDay*** *\** ***@param minute*** *\*/* @Override  
 **public void** onTimeSet(TimePicker view, **int** hourOfDay, **int** minute) {  
 **int** seconds = (hourOfDay \* 3600) + (minute \* 60);  
 **timeText**.setText(timeToString(hourOfDay, minute));  
 sendAlarmTime(seconds);  
 }

*/\*\*  
 \* sends the moodlight a request to send the current alarm time back  
 \* calls sendAlarmTimeRequest()  
 \*/* **public void** timeElapsed() {  
 sendAlarmTimeRequest();  
 }  
  
 */\*\*  
 \* void sendAlarmTimeRequest(void)  
 \* sends a request to moodlight to get the current value of the alarm timer  
 \*/* **private void** sendAlarmTimeRequest() {  
 **byte**[] buffer = **new byte**[7];  
 buffer[0] = MainActivity.***BT\_ALARM***;  
 buffer[1] = MainActivity.***BT\_REQUEST***;  
 buffer[2] = 0x00;  
 buffer[3] = 0x00;  
 buffer[4] = 0x00;  
 buffer[5] = 0x00;  
 buffer[6] = MainActivity.***BT\_DELIMITER***;  
 MainActivity.*bt*.send(buffer, **false**);  
 }  
  
 */\*\*  
 \* void sendAlarmTime(int)  
 \* sends a new value to the moodlight  
 \** ***@param value*** *\*/* **private void** sendAlarmTime(**int** value){  
 **byte**[] buffer = **new byte**[7];  
 buffer[0] = MainActivity.***BT\_ALARM***;  
 buffer[1] = MainActivity.***BT\_SEND***;  
 buffer[2] = (**byte**) ((value >> 24) & 0xFF);  
 buffer[3] = (**byte**) ((value >> 16) & 0xFF);  
 buffer[4] = (**byte**) ((value >> 8) & 0xFF);  
 buffer[5] = (**byte**) ((value >> 0) & 0xFF);  
 buffer[6] = MainActivity.***BT\_DELIMITER***;  
 MainActivity.*bt*.send(buffer, **false**);  
 }  
  
 */\*\*  
 \* String timeToString(int, int)  
 \* converts to values (hours and minutes) into a String  
 \* of the format "hh : mm";  
 \** ***@param hours*** *\** ***@param minutes*** *\** ***@return*** *\*/* **private** String timeToString(**int** hours, **int** minutes){  
 String text = **""**;  
 **if**(hours < 10){  
 text += **"0"**;  
 }  
 text += hours;  
 text += **" : "**;  
 **if** (minutes < 10){  
 text += **"0"**;  
 }  
 text += minutes;  
 **return** text;  
 }  
}