Project: To Design and build possible exact clock synchronized by time signal from radio sender DCF77

Figure 1: The low frequency T-aerial antennas of the continuously operated DCF77 signal in Mainflingen at night.

1 DCF77[1]

Almost all own or have seen advertisement for "atomic" or "radio controlled" clock. They, in our part of Europe, are synchronized (receive) signal from German radio-station DCF77. The time signal is sent on long waves, on frequency 77.5 kHz from Mainflingen near Frankfurt am Main.

Electronic in these, home clocks, decodes amplitude changes; ideal sketch of such signal shows below.

1.1 Aplitude modulering

The DCF77 signal uses amplitude-shift keying to transmit digitally coded time information by reducing the amplitude of the carrier to 15% of normal for 0.1 or 0.2 seconds at the beginning of each second. A 0.1 second reduction denotes a binary "0"; a 0.2 second reduction denotes a binary "1". As a special case, the last second of every minute is marked with no carrier power reduction.

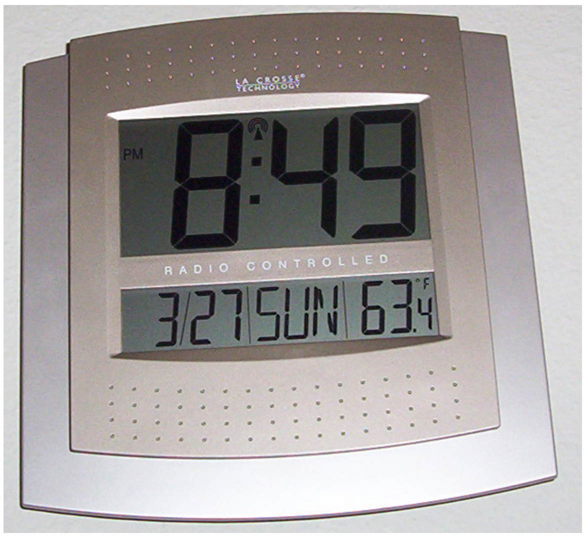


Figure 2: "Atomic clock."

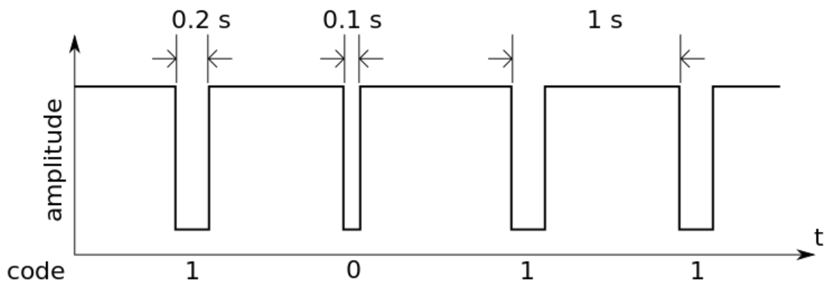


Figure 3: Amplitude modulated signal of DCF77 as a function of time.

This solution is very easy to decode, the problem may be noise from switched-mode power-supplies, washing machine, MP3 players, computers. There is trade-off between low bandwidth (low noise) and sharpness of the received signal (time precision).

1.2 Phase modulation, better time resolution, better signal to noise ratio

One use second independent (engineers say *orthogonal*) kind of modulation, namely *phase modulation*. Interval 0.2 to ca 1.0 second is phase modulated by 512 long, pseudo-random bit sequence. This sequence is clocked with frequency $77.5\ kHz : 120 = 645,8\overline{3}\ Hz$

Our goal is to build receiver-clock that uses this (PM) modulation.

2 General about time

Time is a difficult concept to define, Richard Feynman: "Let us consider first what we mean by time. What is time? It would be nice if we could find a good definition of time. Webster defines "a time" as "a period," and the latter as "a time", which doesn't seem to be very useful. Perhaps we should say: "Time is what happens when nothing else happens". Which also doesn't get us very far. Maybe it is just as well if we face the fact that time is one of the things we probably cannot define (in the dictionary sense), and just say that it is what we already know it to be: it is how long we wait!" [2]

Evolution in time measurements – see[3;4].

Is there anything to take care of at all (precise timing), after all most of us need accuracy of the order of a minute (not to be late for a lesson, train or date with a girl/boy friend)? Well, when we look a little further away from the nose or fiancee and we will consider how a GPS[5], common in use system works; let me cite from Wikipedia: "*The GPS concept is based on time and the known position of GPS specialized satellites. The satellites carry very stable atomic clocks that are synchronized with one another and with the ground clocks. Any drift from true time maintained on the ground is corrected daily. In the same manner, the satellite locations are known with great precision. GPS receivers have clocks as well, but they are less stable and less precise.*

GPS satellites continuously transmit data about their current time and position. A GPS receiver monitors multiple satellites and solves equations to determine the precise position of the receiver and its deviation from true time. At a minimum, four satellites must be in view of the receiver for it to compute four unknown quantities (three position coordinates and clock deviation from satellite time)."

Other devices that need accurate time are, for example, radar, television and computers.

There is no such thing as a singular true time derived from natural phenomena, there is no absolutely accurate clock which could be a reference for others clocks. The time reference is the common agreement and actually in use is *Universal Time Coordinated* or, in french, *Temps Universel Coordonné* (UTC). Logically, previous system – GMT was build on astronomic definition of second, UTC – on the definition based on quantum resonance of Cesium atom (approximately one million more accurately).

UTC uses international net of atomic clocks and data from *International Earth Rotation Service* (IERS).

Worlds most accurate optical clock (2014) wouldn't lose or gain one second in the entire age of the Universe[7] (i.e. 13.6 billion years)!

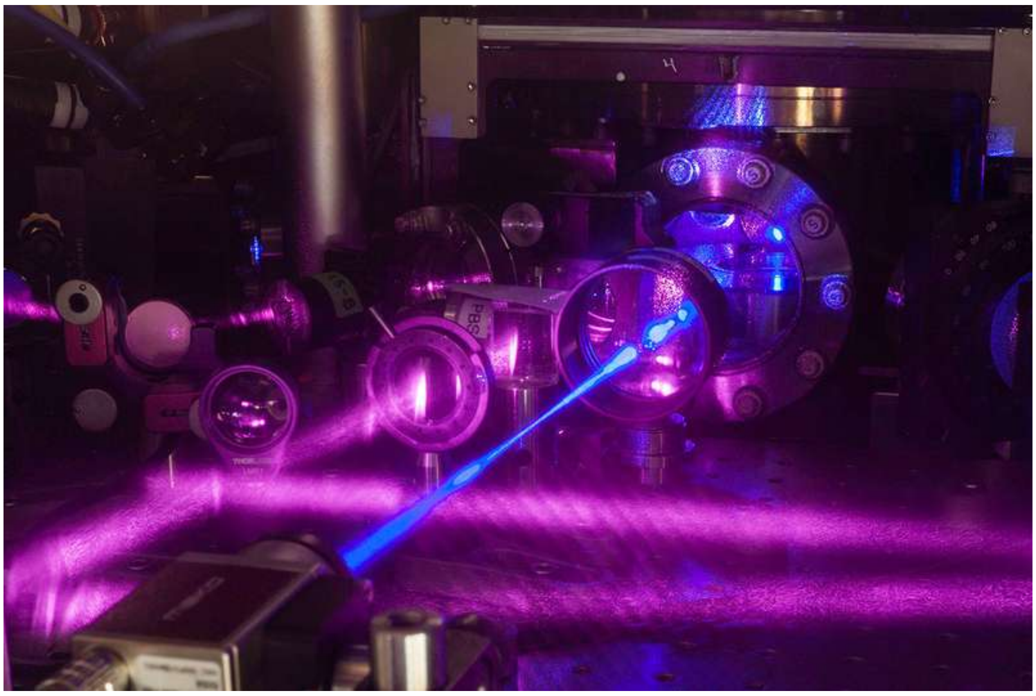


Figure 4: The most precise atomic clock ever made is a cube of quantum gas.

3 UTC (Tems Universel Coordonné)[6]

Atomic clock are more accurate than time steamed from astronomical observations, but most of us comes from the Earth and want to have time corresponding to the Earth's rotation around its axis and around the Sun. But the Earth's rotation is irregular, due to the Moon and the oceans. UTC stays within about 1 second of mean solar time at 0° longitude by occasional inserting *leap* second. The number of seconds in a minute is usually 60, but with an occasional leap second, it may be 61 or 59 instead. Thus, in the UTC time scale, the second and all smaller time units (millisecond, microsecond, etc.) are of constant duration, but the minute and all larger time units (hour, day, week, etc.) are of variable duration. Decisions to introduce a leap second are announced at least six months in advance in "Bulletin C" produced by the International Earth Rotation and Reference Systems Service. The leap seconds cannot be predicted far in advance due to the unpredictable rate of rotation of the Earth.

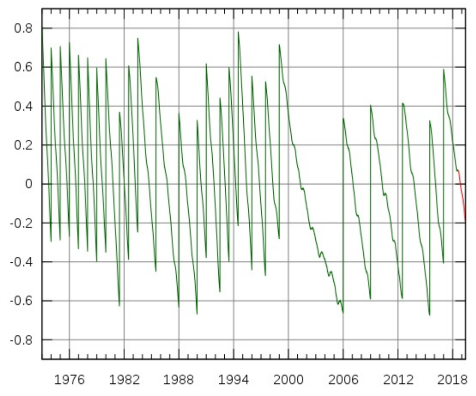
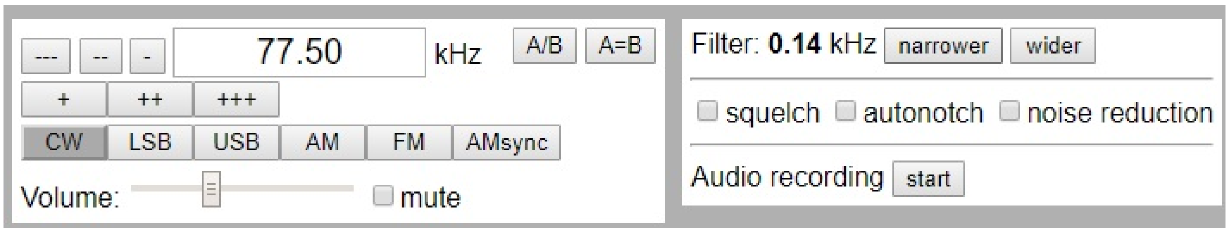


Figure 5: Graph showing the difference DUT1 between UT1 and UTC (in seconds). Vertical segments correspond to leap seconds.

4 Decoders, Solutions

4.1 ”Warm-up” Solution, No Hardware

How to get signal from DCF77 without receiver? Possible solution is to go to the site Wide-band WebSDR, the site of Dutch radio amateur club at the University of Twente. They have build a short-wave receiver which can be tuned by multiple users simultaneously. This time I need only AM modulation, so I have chosen CW (Continuous Wave), $f = 77.5\ kHz$ and filter $140\ Hz$.

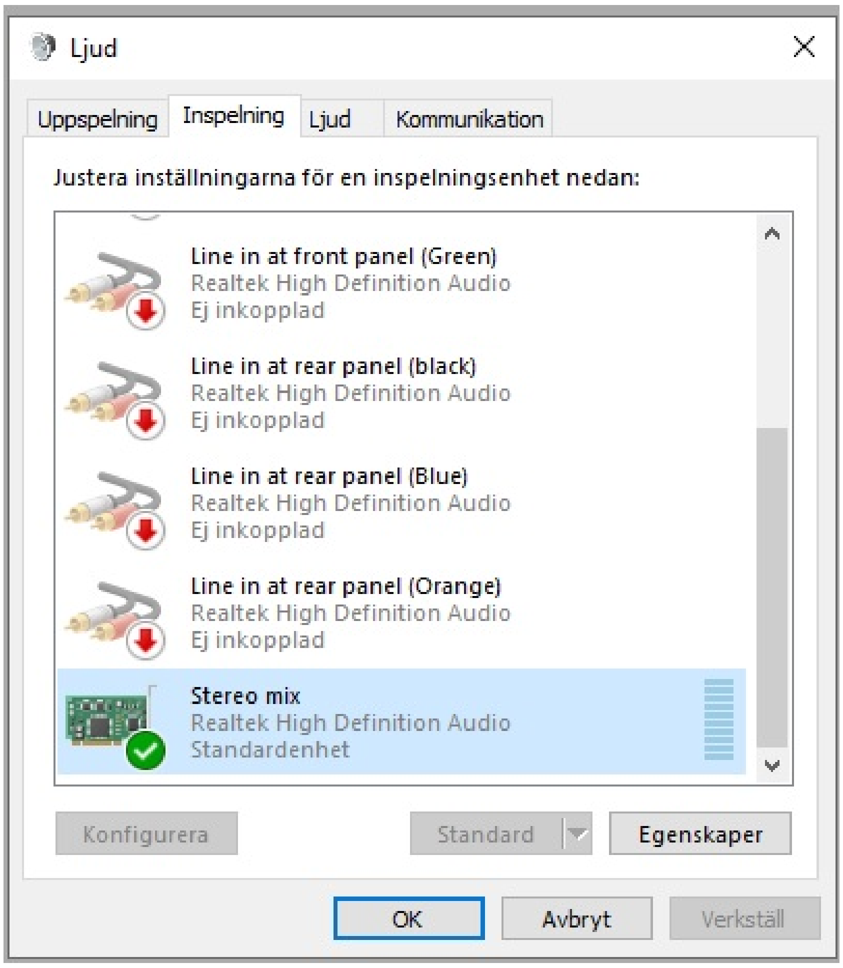


77.50 kHz

Filter: **0.14** kHz

Figure 6: Parameters to receive DCF77 via computer’s sound card.

Then we must choose stereo mixer as input device and set sound volume level rather high:



Ljud

Justera inställningarna för en inspelningsenhet nedan:

Line in at front panel (Green)
Realtek High Definition Audio
Ej inkopplad

Line in at rear panel (black)
Realtek High Definition Audio
Ej inkopplad

Line in at rear panel (Blue)
Realtek High Definition Audio
Ej inkopplad

Line in at rear panel (Orange)
Realtek High Definition Audio
Ej inkopplad

Stereo mix
Realtek High Definition Audio
Standardenhet

Figure 7: Stereo mixer as input sound device (Windows 10).

The next issue, which has taken some time for me since I have never dealt with sound before was to capture the sound. I have used the Bass audio library. Whole program to decode time signal and show result was written in C++ in Visual Studio 2017 (free). Well, this program is almost exact copy of Udo Klein’s[8] program for a variant of *Arduino*[9] module.

Practical remark I have succeed with this solution only when the signal from the Wide-band WebSDR site gets a ton ($f \approx 700\ Hz$). I don't know if this ton is produced as standard beep for Morse code in CW mode or is result of mixing down $f = 77.5\ kHz$? The sound card's highest sampling frequency is 192 kHz, so it is, theoretically, possibly to sample $f = 77.5\ kHz$ and decode phase modulation as well. I could not do it.

Some screen photos of the saved signal:

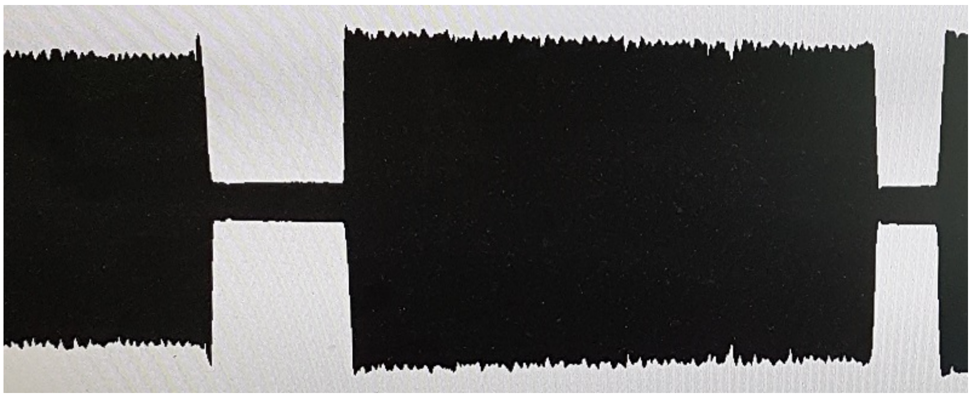


Figure 8: Time resolution $\approx 2\ s$, "1" and "0".

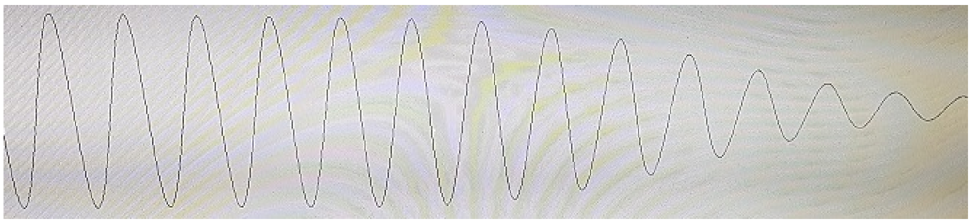
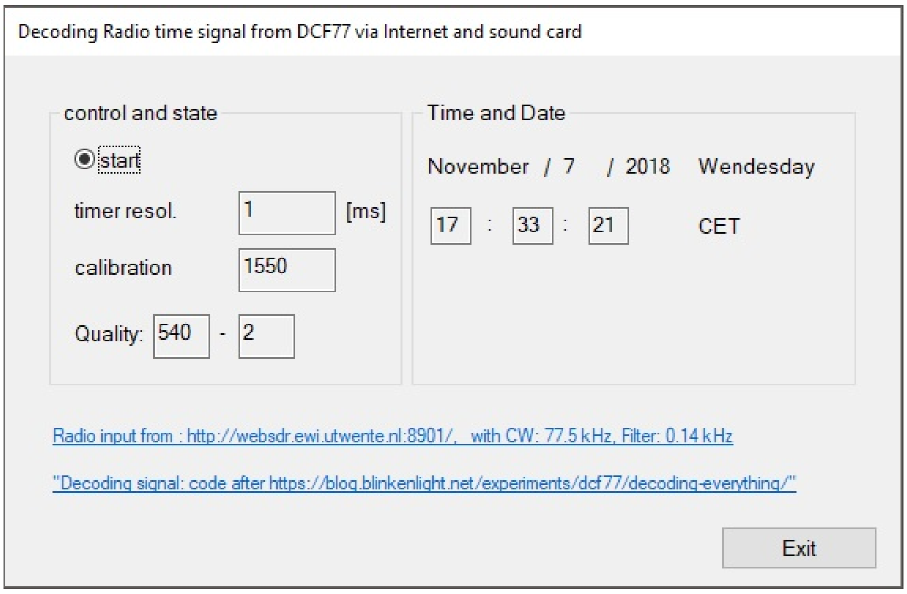


Figure 9: Time resolution $\approx 10\ ms$

And the product, output from the program:



Decoding Radio time signal from DCF77 via Internet and sound card

control and state

(●) start

timer resol. 1 [ms]

calibration 1550

Quality: 540 - 2

Time and Date

November / 7 / 2018 Wendesday

17 : 33 : 21 CET

Radio input from : http://websdr.ewi.utwente.nl:8901/, with CW: 77.5 kHz, Filter: 0.14 kHz

"Decoding signal: code after https://blog.blinkenlight.net/experiments/dcf77/decoding-everything/"

Exit

Figure 10: Useless but working radio controlled clock

The program fragments:

Input for time routine.

Initiate Bass library, start sampling, set help timer to update screen:

```
if (!BASS_RecordInit(-1))
   {
   Error(m_hWnd, "Can't initialize device");
   return FALSE;
   }
                                                                     // 0 == 16-bit sampl.
if (!(Sound::recChannel = BASS_RecordStart(SAMPL_FREQ, MONO, 0, &Sound::DuffRecording, 0)))
   {
   Error(m_hWnd, "Can't start recording");
   return FALSE;
   }

// update screen
SetTimer(IDT_TIMER_SCR,       // timer identifier
           500,               // 0.5-second interval
           (TIMERPROC)NULL);  // no timer callback, see OnTimer(UINT_PTR nIDEvent)
```

Windows Timer calls a routine for each 10 ms, best guaranteed by Windows resolution is $1ms$ (timer resol on the figure). This method queries timer device to determine its resolution, sets and starts our ”main” timer:

```
int setTimer()
    {
    //https://docs.microsoft.com/sv-se/windows/desktop/api/timeapi/nf-timeapi-timegetdevcaps
    if (timeGetDevCaps(&tc, sizeof(TIMECAPS)) != TIMERR_NOERROR)
        {
        // Error; application can't continue.
        return -1;
        }

    wTimerRes = min(max(tc.wPeriodMin, TARGET_RESOLUTION), tc.wPeriodMax);

    if (timeBeginPeriod(wTimerRes) != TIMERR_NOERROR)
        return -2;

    if (!(TimerID = timeSetEvent(10,
        wTimerRes,
        AnalyzeSample,
        NULL,
        TIME_PERIODIC
        )))
        return -3;

    return 0;
    }
```

AnalyzeSample is the callback function. The sampled signal is clipped, rectified (absolute value is taken) and averaging; the average value is qualify as 1 (if it is low) or 0 otherwise. Then the value goes to decoding part, mainly copy of Udo Klein's[8] code, which is very well explained.

4.1.1 Constant Delay

The receiver position is (57.64103, 18.316413) and the DCF77 position is (50.01556, 9.01083). The distance (big circle) $\approx 1043\ km$. If radio-waves propagates as ground waves with speed nearly $c = 300000\ km/s$, this gives us **3.5** ms delay (in fact at this distance, ground and atmospheric paths contribute with approximately same strength, se PTB-Mitteilungen 2009, 3[10].

4.2 Solution 1, AM decoder, with receiving module from HKW[11], real time clock DS3231[12] and Arduino Due utilizing FreeRTOS[13]

4.2.1 Design

DS3231 is low-cost and has accuracy 2 ppm with an integrated temperature-compensated crystal oscillator, battery-backup. We use 32 kHz rectangle output from the module (well this is, exactly $2^{15} = 32768\ Hz$) as interrupt signal. DS3231 uses I^2C interface, I use fast mode: 400 kHz.

References

[1] DCF77, (Wikipedia)

[2] The Feynman Lectures on Physics, Vol. I, Ch. 5-2 (online)

[3] A walk through Time, *A NIST Physics Laboratory Presentation*

[4] The Science of Timekeeping, *Hewlett Packard™* (pdf)

[5] Global Positioning System (Wikipedia)

[6] Coordinated Universal Time (Wikipedia)

[7] The most precise atomic clock ever made, *New Scientist*, 2017

[8] Udo Klein's C++ DCF77 code, Blinkelight

[9] Arduino's site

[10] `https://www.ptb.de/cms/fileadmin/internet/publikationen/ptb_mitteilungen/mitt2009/Heft3/PTB-Mitteilungen_2009_Heft_3_en.pdf`

[11] https://www.hkw-elektronik.de

[12] DS3231.pdf

[13] FreeRTOS port for Arduino Due