CYPRESS MICROSYSTEMS

Application Note AN2xxx

PSoC One chip radio controlled clock

Author: Wim Hendrikse
Associated Project: Yes
Associated Part Family: CY8C24xxx, CY8C27xxx
PSoC Designer Version: 4.1

Summary

A LF radio receiver (40-80 kHz) is realized using the analog part of the PSoC device. The digital part decodes the typical one-minute time information codes. No special crystals are needed; the 32,768 kHz PLL-mode brings you the flexibility to tune by setting parameters in software.

Introduction

Popular radio controlled clocks get their precise time information from long-wave transmitters. These transmitters broadcast a time code signal in the 40-80 kHz range. Worldwide there are a number of well-known transmitters for example: DCF 77, Mainflingen/Germany (77.5 kHz); MSF Rugby/UK (60 kHz); WWVB Fort Collins/US (60 kHz); JJY Japan (40 and 60 kHz); HBG Prangins/Switserland (75 kHz).

The precise coding of the time and date information differs between the various transmitters. However, the base is the same. Every second one out of three codes is broadcasted: a zero-bit, a one-bit or a marker-bit. During one second, the time of transmitting the carrier at full strength and the time of transmitting at reduced strength determine which of the three codes is broadcasted. For example for the German transmitter a zero bit is defined as 100 msec reduced (25 %) signal followed by 900 msec full signal. A one bit: 200 msec reduced signal, followed by 800 msec full signal. A marker bit: full signal during 1000 msec. The other transmitters use a comparable (but not the same) definition for the timing within a second to distinguish between the three codes.

During a minute, the complete time and date information is broadcasted. The main purpose of the marker-bits is to indicate the beginning of a new minute. The German and British transmitters use the marker only at the beginning of a new minute. The US and Japanese transmitters use the marker every tenth second and a dual marker at the beginning of a new minute. The information bits (mostly BCD-coded) during a full minute indicate a.o. minutes, hours, day, month, year, summer/wintertime.

A conventional radio clock

The most common radio clock receiver consists of:

- a tuned ferrite core antenna
- a receiver IC
- a microcontroller
- a crystal-based local clock
- a display

In general, the receiver IC consists of a straight through amplifier, an AM detector and automatic gain control. The needed high selectivity of the receiver is obtained by using one or two external crystals tuned at the transmitter frequency.

A microcontroller decodes the binary output information of the receiver IC. As soon as a correct full minute (60 bits) is received, the microcontroller will update the displayed time/day/month/year. The second-counter of the display will be zeroed. As long as there is no new, correct radio signal the microcontroller uses the crystal-based local clock to display the actual time. Most often, this local clock uses a 32.768 watch crystal.

A radio clock the PSoC way

Knowing the PSoC may be configured having amplifiers, filters, a 32.768 kHz crystal based internal clock and also includes a microcontroller that can decode a bit-stream and that can drive a display the challenge of designing a one-chip solution is born.

Generally, the signal measured at a ferrite core antenna tuned at the frequency of the long-wave radio transmitter will be in the microvolt range. Therefore, to process the broadcasted information you need a lot of gain. You also need narrow filtering around the transmitted frequency to get enough selectivity. This filtering can be done by the use of an external special crystal tuned at the transmitted frequency. A different approach is chosen in this design. Using the heterodyne principle, we first mix the incoming signal with a signal of known frequency. The resulting mixed signal contains (a.o.) the difference in frequency between the two input signals. (The theory of heterodyne used in the PSoC-environment is very well described in Cypress MicroSystems Application Note AN 2111.) The basic approach here is mixing the incoming signal having frequency f with a precise signal having almost the same frequency (f + Δ f) or (f - Δ f). The mixed signal is filtered using a band pass filter with centre frequency Δf. By taking Δf around 150 Hz it is no problem to specify a PSoC user module **BPF_2** having a centre frequency of 150 Hz and a bandwidth of 10 Hz. In this way, the output low frequency signal follows the incoming signal and we realize the asked narrow bandwidth. However, how to get that gain, how to get the precise mixing frequency, how to mix your signal and what other problems do we meet?

Gain, Mixing, Filtering

As described in AN 2111 a low pass filter **LPF_2** user module can be combined with a modulator. Just put the right signal on the mixer input of the filter by setting bits in the **Analog Modulator Control** register. At this point the band pass filter **BPF_2** behaves in the same way as the low pass filter **LPF_2**. However, the highest sample frequency of a **BPF_2** with a centre frequency of 100 Hz will be about 20 kHz, far below the input signal of 40 – 80 kHz and that brings you aliasing problems. The use of two cascaded stages of mixers/filters is the solution for this problem.

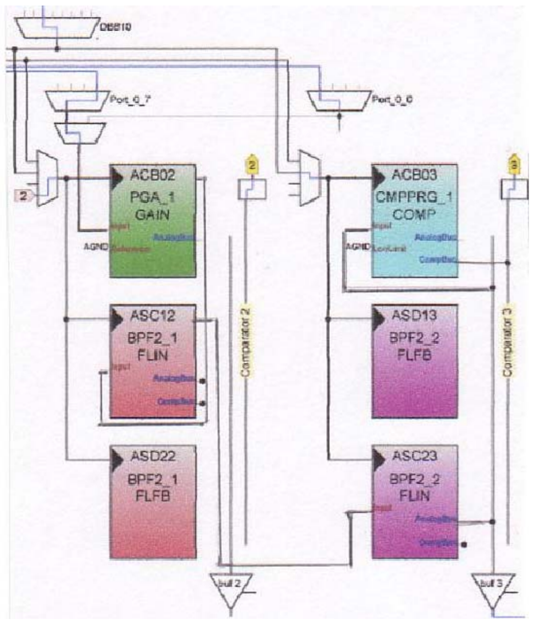

Figure 1: Analog Blocks Column 3 and 4

BPF_2_1 delivers a medium frequency output, **BPF_2_2** the low frequency output (figure 1). Fortunately, there is a bonus: both active filters may contribute to the needed high gain.

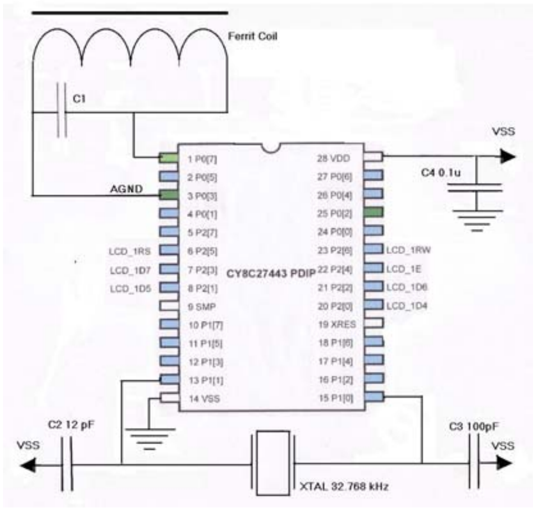

Figure 2: Radio Clock Circuit

The signal coming from the ferrite coil (figure 2) is fed into a **PGA** module. The output of this amplifier forms the input of the medium frequency mixer/filter. In comparison with low pass filters the advantage of band pass filters is the absence of problems at 0 Hz (DC), caused by offset voltages in combination with a high gain.

The mixing frequencies must be highly precise and stable. The solution is the application of the 32.768 kHz external crystal in combination with the PLL mode. The resulting internal PSoC system clock will have a frequency of 23.986176 MHz. Getting the right digital signal for the mixers is just a matter of dividing down that SysClk frequency. As an example, the possible values for a 77.5 KHz receiver are:

Mixing frequency first stage:	85057 Hz
(= SysClk/282)	
Filter center frequency first stage:	7557 Hz
(= 88057 – 77500 Hz)	
Mixing frequency second stage:	7737 Hz
(= SysClk/3100)	
Filter center frequency second stage:	180 Hz
(= 7737 - 7557 Hz)	

In addition, for a 60 kHz receiver a possible choice is:

Mixing frequency first stage:	67377 Hz
(= SysClk/356)	
Filter center frequency first stage:	7377 Hz
(= 67377 – 60000 Hz)	
Mixing frequency second stage:	7543 Hz
(= SysClk/3180)	
Filter center frequency second stage:	166 Hz
(= 7543 – 7377 Hz)	

PWM8 user modules having a half duty cycle generate the mixing signals. To ease the calculation of the mixing frequencies depending on the input radio frequency in combination with the design of the band pass filters the project includes a BPF2_radio_clock Excel worksheet. This is the known Cypress Microsystems BPF2 worksheet with a few extra cells that help you in choosing practical (in relation to dividing numbers) frequencies for filters and mixers.

A band pass filter also needs a sample clock. Therefore, you have to generate four clocking signals in total. The band pass filters are configured with an over sample rate as high as possible.

Using the analog column bus as a connecting line, the low frequent sine signal outputted by the second band pass filter is fed into a level detector build by a **CMPPRG** user module. After choosing the right threshold level, the output will show a low frequent square wave during the full strength signal broadcasted by the transmitter and no signal during the reduced strength signal phase of the transmitter. The output of the programmable threshold comparator is connected to the comparator bus. The low frequent comparator bus signal is just used as an interrupt source and is processed by the software. The interrupt signals of the level detector are also used to realize the automatic gain control. How this is done is described in the software part of this application note.

A **RefMux** (column 1) user module drives an output pin with AGND. An alternative is the use of the testmux bit setting in ACB02; this may save an analog continuous time block.

The resulting output, containing the time and date information is displayed on a 2 x 20/40 lines LCD. Driving the display (port 2) is handled by the **LCD** user module.

Software

The crystal based 32.768 kHz internal clock signal is fed into an 8-bit **Timer8** user module having a period set at 256. In this way, the timer generates an interrupt every 1/128 second. The simplest task in the timer interrupt routine is the incrementing of the displayed time and date information after 128 interrupts.

A second interrupt source is the analog comparator bus caused by the output of the programmable threshold comparator. By flipping bits in the **ALT_CRx** register, we are able to generate an interrupt both at the up-going low frequency sine signal and at the down-going sine. In this way, the interval between up-going signal and down-going signal can be determined and is used as a measure for the signal strength (figure 3). By reducing the reference value of the **CMPPRG** module, in case of an up-going signal interrupt and raising the level at the down-going interrupt we also obtain some hysteresis. The CMPRG_1_SetRef() API function can set these levels.

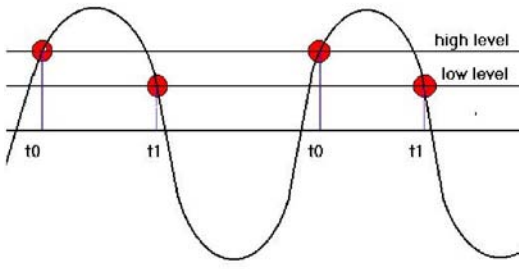

Figure 3: CMPRG interrupt levels

Assuming the time t_1 – t_o is less than 1/128 second (that means the sine frequency is less then 256 Hz) you can do the time measuring by simply capturing the above mentioned **Timer8** module at the two interrupt moments. The resolution is of course 1/32768 second. This time interval is processed in a simple exponential smoothing filter:

new_avg = (1 -α) * old_avg + α * time (1)

Taking α for example, 1/32 results in a simple algorithm with only five right shifts an addition and a subtraction. (1) can be rewritten:

32 * new_avg = 32 * old_avg - old_avg + time (2)

Of course you must store the average and 32 * average. As soon as this smoothed average is below a lower_limit value, the gain will be incremented one-step. The comparable process takes place when the average is above an upper limit: the gain will be decremented. The decision to change the gain by one step will be taken once in a second. The total gain of our circuit is:

gain_PGA * gain_BPF2_1 * gain_BPF2_2 * $4/\pi^2$ (3)

The mixing in the two filters (see AN 2111) causes the loss $4/\pi^2$. The gain of the analog part of our circuit is determined by the setting of the resistor taps in the **PGA** module and by the setting of capacitor C1 (1-31) in the two **BPF2** modules. By using three look-up tables stored in program memory it is possible to change gain in a very flexible and fast way. An index value that may be decremented or incremented depending on the signal strength determines where to find the right settings in the three look-up tables. Actual setting of the gain is easily done by the use of the API-functions **PGA_SetGain()** and **BPF2_SetC1()**. In the design, a relative step size of 3 dB is chosen. That means switching to the next higher gain step results in a plus 3 dB gain. Switching to the next lower step means minus 3 dB.

As soon as the analog column interrupt, caused by the sine signal that equals the signal level set in the programmable comparator module, takes place a software timer is set to zero. The same timer is incremented by the 1/128 second timer interrupt routine. In this way, we know within the timer interrupt routine how long ago the last sine high signal was received. This information is used to determine the state of the receiver. The two states are “signal” (1) or “silence” (0) corresponding with a full strength signal or a reduced strength signal broadcasted by the transmitter. Every 1/128 second the information, “a signal received during the previous 1/ 128” or “no signal received during the previous 1/128 second” is determined and bit-wise shifted into an 8-bit history-register. The decision to change state is based on the last three (or more if you want) bits in this register. For example, the state is “signal” and three zero-bits have been shifted in. This can be interpreted as a no-signal time long enough to change the state into “silence”. This mechanism prevents a state change because of a received short spike or dip. This simple history buffer gives you a flexible and fast way to decide when a change of state must take place.

The moments of a state change are stored in units of 1/128 second and processed to deliver the “silence-time” and the “signal-time” that are transmitted during a one-second interval.

The next step in the processing of the incoming signal involves matching each pair of “signal-time” and “silence-time” with predefined time masks. As is understandable we obtain four possible results: 0-bit match, 1-bit match, marker-bit match or no-match (error).

The consecutive correct bits may be displayed as an indication of the receiving of a correct radio signal. As soon as an end-of-minute marker is detected, a counter is zeroed and the following received correct bits are stored in the corresponding time and date variables. At the next received end of-minute marker these variables, containing the complete time and date information are copied to the display. At this moment, the second counter of the internal clock is set to zero.

Building your PSoC radio clock

The number of external components is reduced to the bare minimum (figure 2, 4).

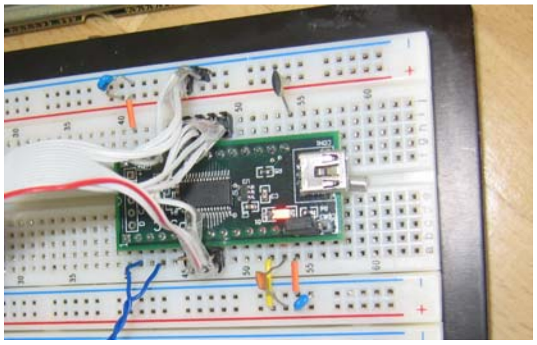

Figure 4. Invention Board as a radio clock

This makes the building to a relatively simple task. Two parts need some attention: the 32.768 kHz crystal and the ferrite coil.

To realize the crystal based PLL mode of the internal clock of the PSoC you are advised to follow the rules published in application note AN 2027. The prototype was build using the “PSoC Invention Board”. Although on this board the sensible crystal pins P1(0) and P1(1) are also permanently connected with the USB interface IC the PLL-mode did function without any problems, even without connecting P1(3) to Vcc as described in AN 2027. The only point of attention is the power supply. Using an external source will be more reliable then the use of the USB cable connected to your computer as power source. Of course, during the programming of the PSoC-chip you can use the USB connection as power source. The PLL mode showed to be absolutely jitter-free in the prototype, using a power supply in the 4.0 – 5.2 V range. However, in the prototype the high-gain analog part preferred a somewhat lower power level. Voltages in the 5.0-5.2 V range seem to generate now and than somewhat more disturbance.

The ferrite coil + capacitor can be bought as a tuned set. 60 kHz and 77.5 kHz are well known values. Of course, you can configure your own tuned set based on a separate ferrite coil and C. Prototypes were build using both pre-tuned sets and self-made sets (figure 5).

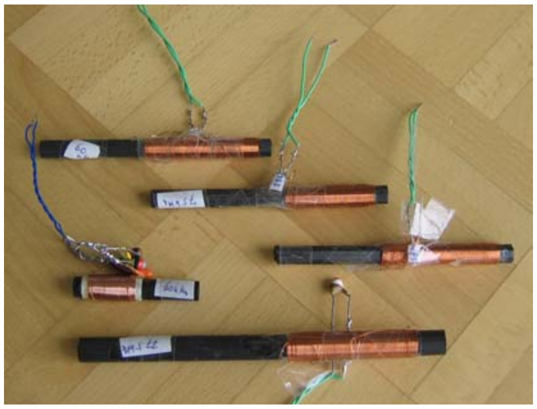

Figure 5. 60, 75 and 77.5 tuned ferrite antennas

If you prefer a self-made antenna than the simple design rules for a ferrite coil are:

$$1/LC = (2\pi * f)^2 \qquad (4)$$

$$L = A * n^2 \text{ nH} \quad (n = \text{number of turns}) \qquad (5)$$

The preferred L is 5 – 10 mH, C about 1000 pF. Try to obtain a Q (= bandwidth / f) of the LC combination of 50 - 100. You can measure A, using a known number of turns. You may expect al value around 30-60. Use 0.25 mm wire. Choose a number of turns of 250-400. Don't forget this number of turns will already result in a C of around 100 pF. Use a good quality ceramic capacitor. Start with a coil with a high number of turns and choose a C that results in a tuning frequency below the wanted frequency. Measure the tuning frequency of your filter using a low C probe. The next step is reducing the number of turns of L to get nearer the frequency of the radio transmitter. Raising the frequency x % roughly means decreasing the number of turns of the coil x %, as can be seen from (4) + (5). Repeat this step a few times to get an accurate filter frequency.

The ferrite core antenna is sensitive to external electrical and magnetically distortions. So try to place it away from digital lines and the LCD display. Start testing without a functioning digital decoding part and without the LCD display. Take a fixed gain (f.e. PGA = 48, C1 =2, C2 =1, C3 =1, C4 =2 in both filters) and connect the analog bus of the second band pass to its pin. Look at the signal on this pin with an oscilloscope. If you receive a radio signal at your location, you will see a sine, flashing at a rate of one per second. Now you can enable the digital decoding software parts and look again at the signal to see if your own digital signals are not the source of distortion. Other rules are of course: no power supplies, PC etcetera close to your ferrite coil antenna. Put Cx close to the coil and use shielded or twisted wire as a connection to the input of the receiver. The ferrite bar must be placed horizontal its side must point into the direction of the location of the transmitting station.

To design your own radio clock you will need the details of the coding of the information by the transmitters. In general the transmitters are (semi-)governmental and the coding scheme is public and can easily be found on the internet. In addition, most datasheets of conventional radio clock receiver IC's have this coding information.

In the included project, you can configure the main.c file to receive one of four different transmitter signals: DCF77, WWVB, MSF and HBG. In normal mode, the LCD layout shows (figure 6):

13:05:55
SUN 28 MAR 2004

An option that can be set in main.c results in some extra system information on the display in hexadecimal bytes, on the first line:

0D = silence time in units of 1/128 second
73 = signal time in units of 1/128 second
00 = received bit
01 = correct code
37 = number of bits since last marker

second line:

3B = smoothed signal strength ($t_1 – t_o$, see figure 3, at lf filter center frequency 165 Hz, units 1/32768 second)
10 = amplifier step (82 dB)

The prototype was tested at a location N 52^0 E 04^0 receiving DCF77 (400 km), HBG (650 km), and MSF (450 km).

More Challenges?

The included project shows that the design focuses on the basic functions of a radio clock. Not all available bits in the signal transmitted are decoded. However, it is not difficult to add in the software things like “announcement of change-over to summer time”, or “leap second”. Also, remember the transmitters broadcast a time that may differ a fixed number of hours with your local time. (time zone) This offset time may be added.

Both the use of ram and flash memory is less then about 25 %. Besides the display pins only two pins, connected to the antenna coil and the pins connected to the crystal are used. This means there may be plenty of room for your specific application functions in combination with the radio clock receiver and decoder. An example: to test the design in WWVB-mode the same PSoC Invention Board functioned as a simulated transmitter by putting a 60 kHz signal (SysClk /400) on pin 18 (port 1-6) using a fourth PWM_8 module. This signal was switched on/off using a 60-byte table with the information of a chosen time in WWVB-code.

Looking at the use of resources in this design, we see that we need only half of the available digital blocks and half of the analog blocks. (Assuming you save the RefMux block by using the testmux in the PGA block to generate AGND.) This means the smaller PSoC CYC8C24xxx, having four digital blocks and two analog column blocks might be used.

Knowing we used only half of the digital and half of the analog blocks in the PSoC CYC8C27xxx perhaps more challenging to you is the idea of building two radio receivers in this one PSoC CYC8C27xxx. Of course, you need two separate tuned ferrite coil antennas, but that is all. In an area where you can receive two different transmitters this might be a useful option. One small technical restriction of the CYC8C27xxx is the number of possible connecting busses between the digital blocks and the analog column mixer inputs. There are only three possible connections (GOE[0], GOE[1] and Broadcast Bus). Therefore, you will have to use an identical mixing frequency presumably in the first filter in both radio receivers. The other resources, like processor speed and memory space will not be a restriction to design such a radio clock that receives and decodes the signal of two transmitters at the same time.

Conclusion

The result of the use of the PSoC architecture to build a radio clock receiver is a remarkable simple design. In comparison with a conventional radio clock receiver, we note that:

- No special crystals tuned at the transmitter frequency are needed. All narrow filtering is done on the base of the 32768 Hz watch crystal.
- No special chips are needed; the general-purpose PSoC microcontroller fulfills all functions.
- Just one chip is needed, less passive components are needed.
- The PSoC offers many possibilities to add specific functions to the designed radio clock.

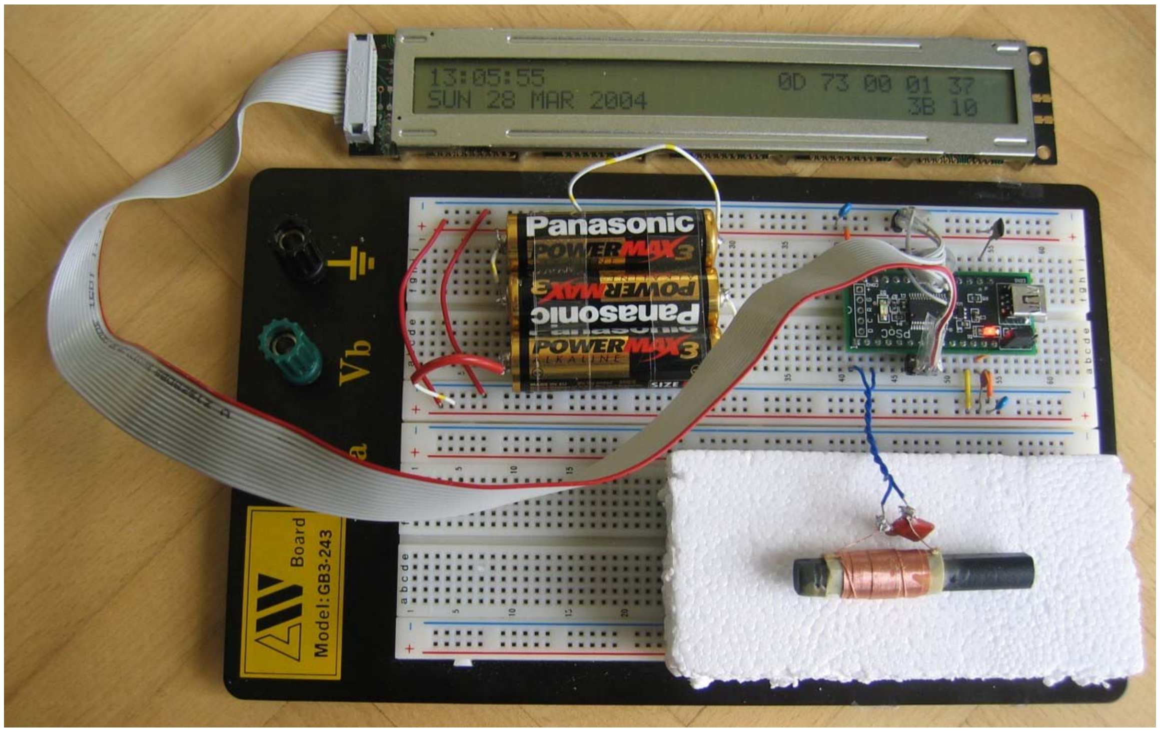

Figure 6: Receiver, receiving the DCF77 signal, the display shows optional system information

About the Author

Name:	Wim Hendrikse
Title:	
Background:	Electronic engineer
Contact:	