A 42-kHz ULTRASONIC TRANSMITTER BASED ON A TMS320C6713B PROCESSOR

Victor Cojocaru

Ghitu Institute of Electronic Engineering and Nanotechnologies, Technical University of Moldova, Academiei 3/3 str., Chisinau, MD-2028 Republic of Moldova
E-mail: vikcojocaru@gmail.com

(Received, 20 April, 2023)

https://doi.org/10.53081/mjps.2023.22-1.07

Abstract

The paper describes a software design for a 42-kHz ultrasonic transmitter based on the 32-bit timer of a TMS320C6713B processor.

Keywords: ultrasound, 32-bit timer, pulse trains.

Rezumat

În lucrarea dată este descris designul soft al unui transmițător cu ultrasunete în baza timer-ului de 32 de biți al unui procesor TMS320C6713B.

Cuvinte cheie: ultrasunet, timer de 32 de biți, tren de impulsuri.

1. Introduction

Ultrasounds are acoustic waves with frequencies higher than 20 kHz. They are used, particularly, in medicine and technology. One of these technical fields is robotics, where ultrasound is understood as echolocation. Short pulses modulated with a sinusoidal, chirp, or chaotic signal are used for echolocation. The aim of this study is to develop a software for a 42-kHz ultrasound transmitter consisting of pulse trains with a required length.

2. Pulse train implementation methods

To obtain a required length of pulse trains, it is necessary to calculate the pulse duration $\tau_1$ and pause duration $\tau_2$, with the minimum and maximum detection distances $D_{\text{min}}$ and $D_{\text{max}}$ value, respectively, of a MuRata MA40B8R ultrasound receiver [1, 2], as shown in Figure 1.

With

$$D_{\text{min}} = 0.2m$$

and

$$D_{\text{max}} = 6m$$

the values of
\[
\tau_1 = \frac{2D_{\text{min}}}{V} = \frac{0.4}{350} = 0.001\text{s}
\]

and

\[
\tau_2 = \frac{2D_{\text{max}}}{V} = \frac{12}{350} = 0.034\text{s}
\]

are obtained.

3. Architecture of the 32-bit timer of the TMS320C6713B processor

To obtain pulse trains with a required frequency, it is necessary to use interrupts generated by a 32-bit Timer [4]. The timer operates on the incremental principle in the TMS32C6000 processors. A 32-bit timer can be used for the following issues:

- Counters
- Pulse generation
- CPU interrupts.

The timer has two synchronization modes: from the external clock through a TINT pin and from the internal clock. Through a TOUT pin, it can be used as a clock source, for example, for A/D to start the conversion process. Similarly, signals from a pulse generator are obtained through a TOUT pin. In addition, TINT and TOUT pins can be used as general purposes pins in cases where synchronization or external generation is not required. The timer of a TMS320C6713B processor has three registers [3,4], as shown in Figure 2.
CTL initiates the operating mode of the timer and governs the TOUT pin. BRD contains the number of clock cycles at the input or determines the TOUT signal period. CNT is a 32-bit register that stores the current counter value, while incrementing the values from 0 to N. The timer has two operating modes: a pulse mode and a pulse oscillator mode. The second mode should be used to form pulse trains. The initial setting of a timer requires four basic steps, as shown in Figure 3.

![Figure 3. Initial setting.](image)

4. Structure and organization of the program

To provide the formation of pulse trains, it is primarily necessary to perform required calculations for values of the timer registers, as shown in Fig. 4.

![Figure 4. Formation of pulse trains.](image)

The clock frequency of a TMS320C6713 processor is 225 MHz, while the clock frequency of the timer is 56.25 MHz [3]. The clock pulse duration is calculated using expression (5):

$$\tau_{clock} = \frac{1}{56.25 MHz} = 1.8 \times 10^{-8} \text{ s}.$$  \hspace{1cm} (5)

The duration of a 42-kHz clock pulse is calculated using formula (6):
\[ \tau = \frac{1}{42kHz} = 2.38 \times 10^{-5} \text{ s} \]  

(6)

The calculations for the required value of \( N \) of the BRD register are performed using formulas (7) and (8):

\[ \tau = \tau_{\text{clock}} \times N \]  

(7)

\[ N = \frac{\tau}{\tau_{\text{clock}}} = 1339 \]  

(8)

After calculations for the formation of a 42-kHz meander, it is necessary to calculate the number of pulses \( N_{\text{imp42kHz}} \) required for emission (E) and reception (R), according to formulas (9) and (10), making use of the values of \( \tau_1 \) and \( \tau_2 \) calculated according to formulas (3) and (4), respectively:

\[ N_{\text{imp42kHz}}^E = \frac{\tau_1}{\tau} = 42 \]  

(9)

\[ N_{\text{imp42kHz}}^R = \frac{\tau_1}{\tau} = 14279 \]  

(10)

To obtain pulses at the transmitter, it is necessary to use the ability of the TOUT pin to be used as a general-purpose pin, or a timer pin, depending on the bit value of FUNC and DATOUT of the CTL register (see Table 1).

**Table 1. Structure of the CTL register**

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Value</th>
<th>Value used</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-12</td>
<td>Reserved</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>TSTAT</td>
<td>0/1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>INVNP</td>
<td>0/1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>CLKSRC</td>
<td>0/1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>CP</td>
<td>0/1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>HDL</td>
<td>0/1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>GO</td>
<td>0/1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>Reserved</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>PWID</td>
<td>0/1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>DATIN</td>
<td>0/1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>DATOUT</td>
<td>0/1</td>
<td>1 on emission</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 on reception</td>
</tr>
<tr>
<td>1</td>
<td>INVOUT</td>
<td>0/1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>FUNC</td>
<td>0/1</td>
<td>1 on emission</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 on reception</td>
</tr>
</tbody>
</table>

In the PRD register, enter the required value for a frequency of 42 kHz.
T\_EMISSION=42 is the number of 42-kHz pulses required for emission.
T\_TOTAL=14279 is the number of 42-kHz pulses required for emission.

The program code was described in the Code Composer Studio™ development environment. It includes several facilities to facilitate the use of the peripheral. Figure 5 shows the organizational chart of the program, while in Figure 6 the algorithm of the pulse train formation program is presented.
The Chip Support Library (CSL) provides a C++ interface for setting and controlling chip-peripherals. It is composed of discrete modules, which are constructed and archived in a library of files. Each module is referred to only one peripheral module. In our case, it is the timer. The timer support is in the csl_timer.h file.
Fig. 5. Algorithm of the pulse train formation program.
5. Conclusions

The use of a TMS320C6713B processor at a clock frequency of 225 MHz makes it possible to design transmitters operating at frequencies higher than 42 kHz. The use of the timer simplifies the description of the code, because the frequency can be changed simply by changing the value of the registers.

References