Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 29/11/2026 11:46:41

Annales UMCS Informatica AI 1 (2003) 237-244

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Fast software PWM channel

Jerzy Kotliński*

Institute of Physics, Maria Curie-Skłodowska University,
Pl. M.Curie-Skłodowskiej1,20-031 Lublin, Poland

Abstract

Due to great value of the time constant of the integrator circuit, a hardware defined PWM (Pulse Width Modulation) signal makes it possible to build a Digital to Analog Converter (DAC) characterized by a relatively long response time. An attempt at creating the software defined PWM signal leads to increased response time of the DAC converter with a coefficient several hundred times longer than its hardware equivalent. Reorganization of the PWM signal allows for its software synthesis, in that the response time of the DAC converter is only several times larger than its classical equivalent.

1. Introduction

Some microcontrollers possess an output called PWM output or PWM channel. This output can be used for building a DAC converter which transforms the digital signal into its analog equivalent by using an integrator circuit. The disadvantage of the above mentioned converter is a long response time caused by a large value of the integrator time constant. The attempt of program construction of the PWM signal is greatly limited because it is not possible to set very short time periods indispensable for realization of the PWM signal idea without using full power of the processor. Hence, it follows that construction of the program defined PWM_DAC converter, which would be as fast as its hardware equivalent, is impossible.

2. DAC_PWM converter – standard construction

Fig. 1 presents a block diagram of the DAC converter, which is controlled by the PWM signal. The digital PWM signal is directed to the integrator circuit built of *R* and *C* elements. Low-pass filtration of the digital signal takes place in the *RC* system. The voltage taken from the capacitor through the separator is put

* *E-mail address:* jotkot@tytan.umcs.lublin.pl

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 29/11/2026 11:46:41

out as analog signal DAC (U_a). Figure 1 presents the simplest form of the filtrating system.

In microcontrollers the generator of the PWM signal occurs as an independent unit with a digital comparator [1]. The PWM generator can be realized by using well-expanded counter systems with CCU (Compare/Capture Unit) [2] or PCA (Programmable Counter Array) [3, 4].

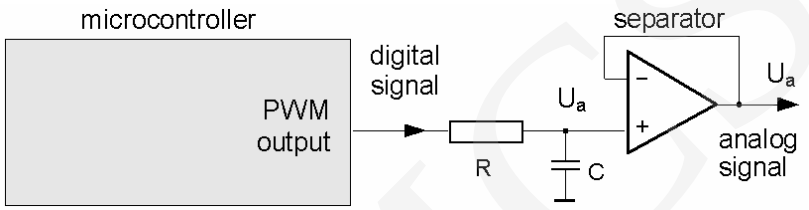


Fig. 1. Block diagram of the DAC_PWM converter

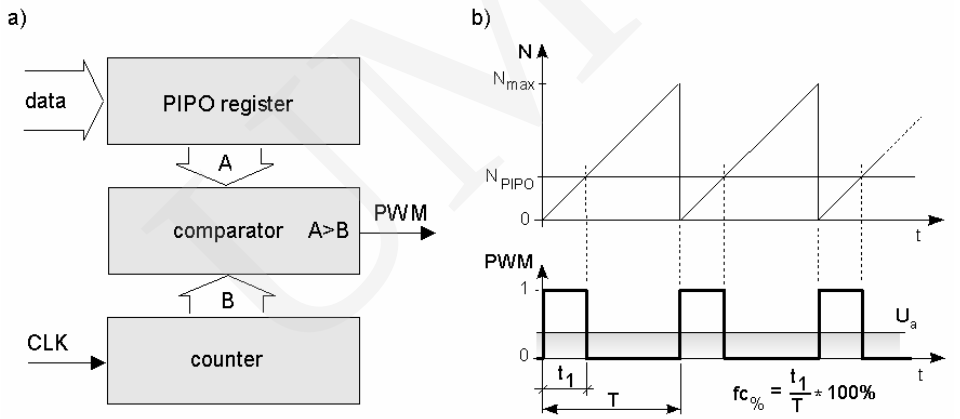


Fig.2. Structure of the PWM signal generator and its operating rule

Fig. 2 shows the hardware system of the digital PWM signal generator and the way of creating this signal. The system is composed of a binary counter, buffer register and comparator whose one of its outputs, e.q. '*A>B*', can be that of PWM signal. The digital data stored in the buffer register PIPO is compared with the content of the binary counter which is changed by the clock CLK signal. For the counter state lower than the PIPO register state the comparator output is set to logical '1' and for other cases to logical '0'. Due to limited binary length of the counter the situation indicated above occurs periodically. This way allows to obtain the rectangular PWM signal whose value of the pulse-duty factor, *fc*, is proportional to the PIPO register content. The repetition time, *T*, of the PWM signal is constant and determined by the counter capacity and the clock CLK frequency.

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 29/11/2026 11:46:41

3. Time-constant of integrator circuit

As mentioned before, due to filtrating property of the integrator the digital PWM signal is transformed into its analog equivalent. This transformation is presented in Fig. 3. Due to continuous and periodical process of the capacitor reloading, the analog voltage level U_a undergoes slight changes U_D (Fig. 3b) which are called further pulsation or pulsation voltage. Under the equilibrium conditions the voltage value U_a stabilizes on such a level that successive reloading of the capacitor takes in and out the same electrical charge. Pulsation voltage of the analog signal is disadvantageous and should be minimized. This can be achieved by suitable increasing the time constant, $t=RC$, of the integrator.

Because the level of the analog signal U (U_D) is defined by the exponential function, in the case when $t >> T$ any changes of the analog signal are insignificant. Moreover, for such case the values of the exponential function index are also small, which allows presentation of the capacitor reloading by straight lines (Fig. 3b).

To obtain the analog PWM signal satisfying the requirement of the resolving of the DAC converter, it is necessary to reduce the pulsation voltage U_D to a level not exceeding the fixed limit. Good estimation can be achieved on condition, that in the case of realization of n-bit DAC converter, the error caused by the presence of signal pulsation does not exceed 2^{-n} of full conversion range. This provides the possibility for realization of the DAC converter with the accuracy of +/-0,5 bit. For the classical DAC_PWM converter the most disadvantageous case is at 50% value of the pulse-duty factor of the PWM signal (proof in the further part of the paper).

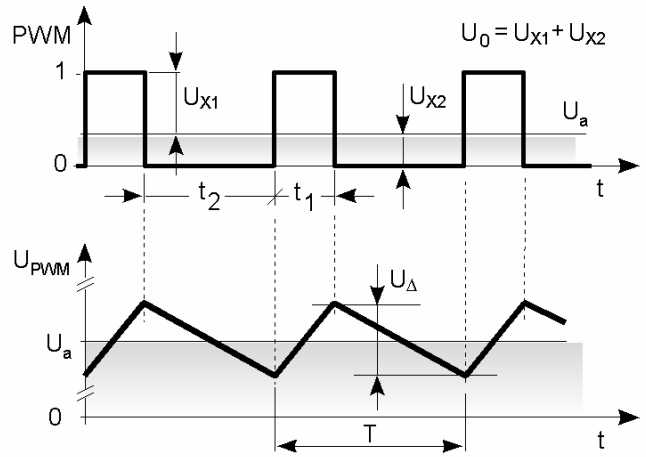


Fig.3. Saw-shape deformation of the analog PWM signal

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 29/11/2026 11:46:41

This corresponds to the analog voltage level situated halfway of the maximum voltage and $U_{x1} = U_{x2} = U_0/2$ and $t_1 = t_2$ (see Fig. 3b). For such case the value of pulsation voltage U_D can be estimated by subtracting two values of voltage U_a for the stipulated moments $t_1 = 0$ and $t_2 = T/2$, in which the voltage U_a achieves its minimal and maximal values. Taking for calculations full exponential form of the capacitor reloading we can obtain:

$$U_{\Delta} = \frac{U_o}{2}\exp(0) - \frac{U_o}{2}\exp\left(\frac{-T/2}{RC}\right),$$

$$U_{\Delta} = \frac{U_o}{2}\cdot\left(1-\exp\left(-\frac{T}{2RC}\right)\right),$$

but for the 8-byte converter:

$$U_{\Delta} = 2^{-8}\cdot U_o = \frac{U_o}{256}:$$

hence:

$$\frac{U_o}{256} = \frac{U_o}{2}\cdot\left(1-\exp\left(-\frac{T}{2RC}\right)\right),$$

$$\exp\left(-\frac{T}{2RC}\right) = 1-\frac{1}{128} = 0,99218,$$

$$\frac{T}{2RC} = -\ln 0,99218 = 0,00785,$$

$$RC = 63,69\cdot T.$$

Therefore, it can be assumed that construction of 8-byte DAC_PWM converter which would be characterized by 1-bit resolving requires application of an integrator system with time constant determined by the dependence: $t_{RC} \geq 64T$.

For standard construction of the DAC_PWM converter the time T of the single cycle PWM is usually in the range *100-500ms*. For *T=200ms* the time constant of the integrator t_{RC} should be *13-15ms*. Any 8-bit data exchange from the value 0 to 255 becomes stabilized after 6-7 periods t_{RC}, i.e. after about *80-100ms*.

4. DAC_PWM converter – modified construction

The classical construction of the digital PWM signal consists in grouping the logical states '0' and '1' into neighboring single and continuous blocks. This leads to time maximize of the capacitor reloading, and thus to put the pulsation level U_{Δ} on the maximal level.

Breaking blocs, e.q. logical '1' into smaller, indivisible elements, causes that capacitor loading and unloading become the shortest possible. This way directly leads to stabilize the pulsation voltage on a minimal level. Both situations are

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 29/11/2026 11:46:41

given in Fig. 4a and 4b. Figure 4a presents a classical diagram of the capacitance reloading, and Fig. 4b a modified one. The decomposition of a classical block, e.g. '1', should lead to such location of '1' that their time distance from one another is equal with the accuracy determined by the individual duration time of the state '0' or '1'.

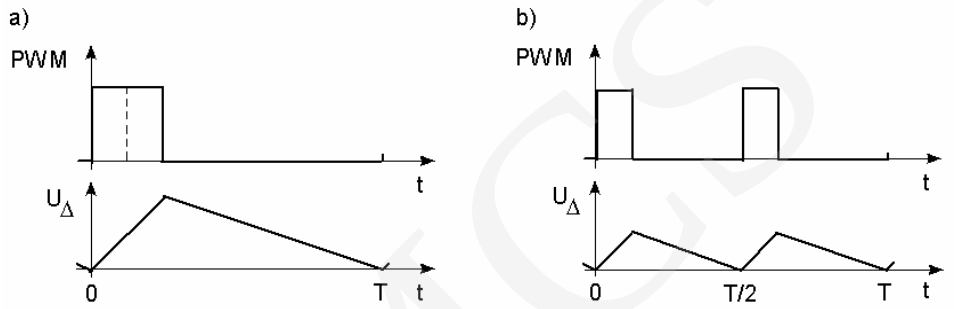


Fig. 4. Effect of digital PWM signal shape on the pulsation amplitude U_D: a) classical system; b) modified system

Figure 5 presents a relative value of the pulsation voltage U_D for classical and modified cases. The horizontal axis shows the value of the pulse-duty factor (*fc*) of the PWM signal. The vertical one presents the voltage U_D of 100% value for the most disadvantageous cases. A logarithmic scale was introduced in Fig.5a due to a large span of U_D. Figure 5b presents a part of Fig. 5a with a linear vertical scale. The results presented in the figures for classical and modified cases were calculated for the same parameters of the integrator, which are characteristic of one-bit resolving of the DAC converter.

As follows from Fig. 5, in the case of classical construction of the PWM signal the pulsation voltage increases to reach the maximal value at 50% of the pulse-duty factor. In the case of modified construction of the PWM signal the pulsation voltage achieves its maximal value for the last but one point on the axis *X*, amounting 1.56% of the maximal value. For 50% of the pulse-duty factor the pulsation voltage drops to 0.8%.

Analyzing the above presented results it can be seen that the ratio of maximal pulsation voltage U_D for the classical and modified cases is 100:1,56, i.e. about 64:1. This is the same value by which the time constant of the integrator should be lengthened to obtain one-bit resolving in the case of 8-bit DAC_PWM converter. It means that conversion of classical PWM signal shape to its modified equivalent allows using the same integrator time constant comparable to the repetition time *T* of the standard PWM signal! Thus a fast DAC_PWM converter can be obtained. Hardware preparation of such a converter would require great changes in hardware of the microcontroller PWM channel.

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 29/11/2026 11:46:41

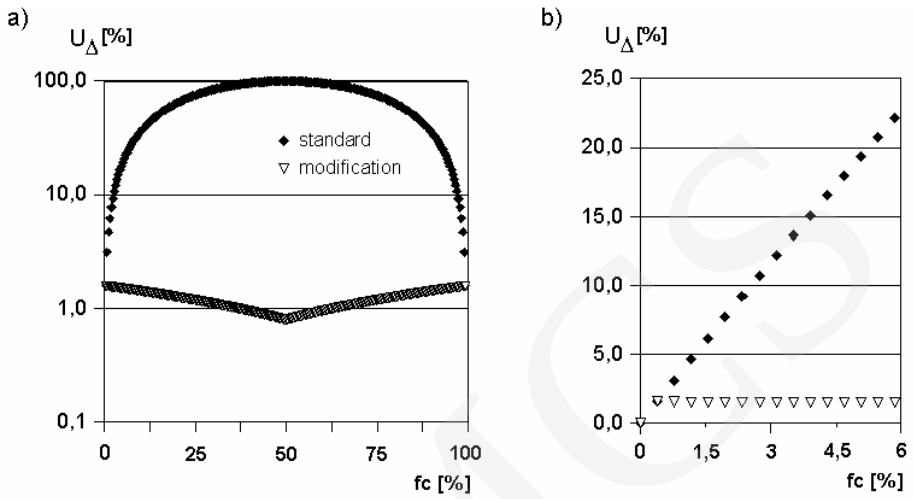


Fig. 5. Relative amplitude of U_D pulsations for classical and modified cases

There is another way to interpret the above results. If the time constant of the integrator is not changed than the conversion of the classical PWM signal to its modified equivalent allows formal lengthening of the repetition time T of the PWM signal with the same level of U_D pulsations. For the 8-bit converter, the repetition time T can be close to the time constant of the integrator, which was used for the classical PWM signal. In this case all 256 bits of the modified PWM signal ought to be emitted by PWM output at the time equal to the time constant of the RC integrator. For the instance given earlier, in which the repetition time T of the classical PWM signal was *200ms*, the value of the time constant of the integrator was about *14ms*. This causes changes of the modified PWM signal every *55ms* (*256*55ms =14ms*). It is the time, which, in the case of fast microcontrollers, allows generation of the PWM signal by using microcontroller interrupts from the timer/counter circuit. In the case of popular microcontrollers the time constant of the integrator should be increased several times. It causes that the software defined PWM channel is only a few times slower than its hardware equivalent.

5. Constructing the filling code

The above presented construction of the modified PWM signal consists in breaking the standard PWM signal into basic elements and regrouping them according to the modified method. Thus it can be considered that a new signal with a specific content is formed. In the further part of the paper this special bit construction of the modified PWM signal is called the *'PWM Filling Code*' or *'filling code*'. On the account of its construction the filling code must include 2^n bits of the PWM signal; the quantity n determines the bit resolving of the PWM

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 29/11/2026 11:46:41

channel which is 2^{-n}. The bits of the PWM signal must be successively directed to the PWM output and emitted in the time T which is connected with the time constant of the integrator by the dependence: $T \approx t_{RC}$.

Direct software operation of the PWM channel consists in cyclic taking up the filling code elements, e.q., from RAM or ROM memory, and taking them out through one of the microcontroller outputs. The particular bits of the filling code should be taken out periodically at time of interrupt servicing.

Indirect software operation of the PWM channel consists in assigning an adequate filling code to the DAC converter data and putting it, e.q., in the RAM memory. Owing to the simplicity of action the best way to carry out such operation would be rewriting the code from previously prepared tables placed in ROM memory. Due to a large bit width of the filling code (256 bit for 8-bit DAC converter) the table size must be very large: e.q. the 8kB area (256*256 bit) for 8-bit converter.

The size of ROM table can be reduced to 137 bytes by using the construction procedure the idea of which is presented in Fig. 6.

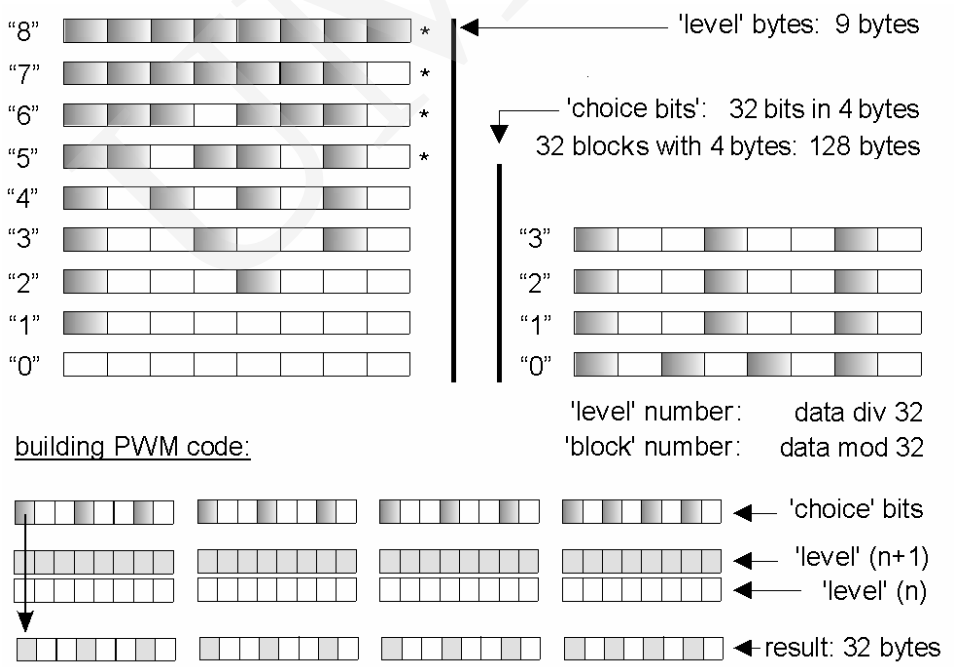


Fig. 6. Constructing the filling code for 8-bit DAC_PWM converter

To create 256 bits of the filling code, 32 bytes (256 bits) of RAM memory are reserved as the operating area. As can be seen, division of the 8-bit data for DAC by factor 32 gives 8 total parts called further 'level' number n (0..7) and 32

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 29/11/2026 11:46:41

fractional parts called further 'block number'. Each 'level' ***n*** is assigned into one pattern byte with a map of specially located logical '1'-s. In Fig.6 each darkening of the area in the 'level' pattern byte means logical '1'. The eight-component table of byte pattern is completed by extra byte with 8 bits of logical '1'-s. The table of the 9 'level' pattern bytes can be placed in ROM memory. As can be seen, further construction of the filling code consists in determination of 'level' and introducing 32 bytes (256 bits) of the pattern bytes into the RAM memory corresponding to a 'level' (n) and the 'higher level' (n+1).

As a result of division there are 32 fractional parts, called 'block numbers', that is 32 specially selected ways of taking the pattern bytes from the level (n) or (n+1) and 32 ways of its description. Each of those ways is the same on each ‘level’ (n)! This can also be tabulated as 32 blocks of 4 bytes with 32 'choice bits'. In this case the next ROM table size should be 128 bytes. Each 'choice bit' with the logical state '0' can indicate the (n) 'level' and each '1' the (n+1) one.

Summing up, construction of 256 bit filling code consists thus in dividing the byte of data for 8-bit DAC by factor 32 and determining the 'level' and 'block number'. Each 'block number' indicates 4 table bytes with 32 'choice' bits. Each 'choice' bit indicates the 'level' (n) or (n+1) of the byte pattern. Successive taking up of the byte pattern indicated by 'choice' bits '0' or '1' and putting it in the RAM area causes formation of the filling code.

6. Conclusions

The paper presents a new construction of the PWM signal which enables several dozen decrease of the time constant of the integrator of the DAC_PWM converter.

The new construction of the PWM signal enables also software possibility to generate the PWM signal and build the DAC_PWM converter with the response time differing not much from the classical one.

Due to the special construction of the digital code of the modified PWM signal the name *'PWM Filling Code*' is proposed.

The way of software forming the filling code whose construction requires only 137 bytes of tabularized data is discussed in the paper.

References

[1] Starecki K., *Mikrokontrolery jednoukładowe rodziny 51*, NOZOMI, Warszawa, (1996), in Polish.
[2] ‘8-bit CMOS Single-Chip Microcontroller SAB80C515/SAB80C535’; Data Sheet, Siemens.
[3] ‘CHMOS Single-Chip 8-bit Microcontroller 8xC51GB’; Data Sheet, Intel.
[4] ‘CMOS Single-Chip 8-bit Microcontroller 80C575/83C575/87C575’; Data Sheet, Philips Semiconductor.

PDF created with FinePrint pdfFactory Pro trial version www.pdffactory.com