

# Микропроцесорна системотехника

## Лабораторно упражнение №4

### “Изследване на таймерен модул на микроконтролер MSP430FR6989”

#### 4.1. Въведение

В практиката се налага микроконтролерните системи да генерират или да отмерват точно определени времеинтервали. Ако програмиста знае тактовата честота на контролера и времето за изпълнение на асемблерните инструкции, то той би могъл да реализира софтуерно времеинтервал. Това обаче почти никога не се прави, защото тактовата честота може да се изменя динамично, изпълнението на програмата не е последователно (обслужват се прекъсвания, които биха удължили софтуерното закъснение), съществуват инструкции с условие, които променят хода на програмата. Затова във всички микроконтролери се вграждат хардуерни модули за отмерване на време, сърцето на които е един брояч. Тези модули се наричат таймери. Те се конфигурират от микропроцесора, чрез запис в съответните им регистри. Броячът се увеличава/намалява автономно, независимо от останалата система. При препълване той се нулира и сигнализира за това на микропроцесорното ядро чрез сигнал за прекъсване.

Когато увеличаването/намаляването на брояча става в интервал от време, определен от извод на микроконтролера се казва, че таймерът работи в режим *Capture*. Това е полезно за **измерване продължителността на даден времеви параметър на сигнал**. Аналогия може да се направи с хронометъра. При него, като натиснете бутона за стартиране започва да отброява и като натиснете бутона за спиране, спира да отброява и вие виждате отброеното време. В случая отброяването започва, когато на съответния вход на микроконтролера нивото се смени и спира, когато нивото се върне до изходното. Това, при какво ниво ще започне и ще спре отброяването се определя от конфигурацията на регистрите.

Когато текущата стойност на брояча се сравнява с друг регистър и при съвпадение на двете стойности се генерира прекъсване се казва, че таймерът работи в режим *Compare*. Това е полезно за получаване на **сигнал с желана от нас продължителност**. Този сигнал може да е вътрешен за контролера и да управлява вътрешни процеси (например, прекъсване работата на процесора) или да бъде изведен на някой извод.

Когато стойността на брояча се сравнява с други два регистъра и автоматично се запуска отначало след достигане на съвпадение се казва, че таймера работи в режим на генерация на **ШИМ** (Широчинно Импулсна Модулация) сигнал, или още режим *PWM*. В последния случай единия регистър **определя периода на ШИМ** (респективно честотата), а другият – **продължителността на импулса/паузата**. Този режим е много полезен при управление на силови електронни устройства (DC/DC конвертори, инвертори, управляеми токоизправители), както и за директно управление на товари чрез използването на мощен ключов елемент (мощни светодиоди, електродвигатели, нагреватели). При по-ниски напрежение, като силов ключов елемент обикновено се използва мощен MOS транзистор, поради почти нулевото му напрежение дрейн-сорс при работа в линейната област. При по-голяма мощност на товара се налага използването на драйвери за управление на MOS транзистора, които да осигурят необходимия гейтов ток за работата му при по-висока честота.

Ако честотата на задаващия генератор за таймерите е кварцово стабилизирана, точността при измерване/задаване на времеинтервалите е много висока. Това предполага приложимостта на микроконтролерите за точно измерване на времевите параметри на сигнали, както и за генерирането на сигнали с много точно зададени времеви параметри.

Таймерите понякога се наричат **Capture/Compare/PWM** или **CCP** модули.

#### 4.2. Таймерен модул на MSP430FR6989

На **фиг. 4.1** е показана блокова схема на таймерния модул. От нея се вижда, че 16-битовият брояч (TBxR) за един таймерен модул е само един, докато CCP регистрите може да достигнат до 7 броя. За конкретния модел (6989) те са 7 (т.е. максимумът). Всеки един CCP регистър е поместен заедно с допълнителна логика в таймерен подмодул. От друга страна микроконтролера има общо три таймерни модула (Timer\_A0, Timer\_A1 и Timer\_B0) с общо 14 CCP регистъра. Тактовата честота, която се подава на брояча идва от един от източниците TBxCLK, ACLK, SMCLK или INCLK. Следват



### Пример за работата на таймера

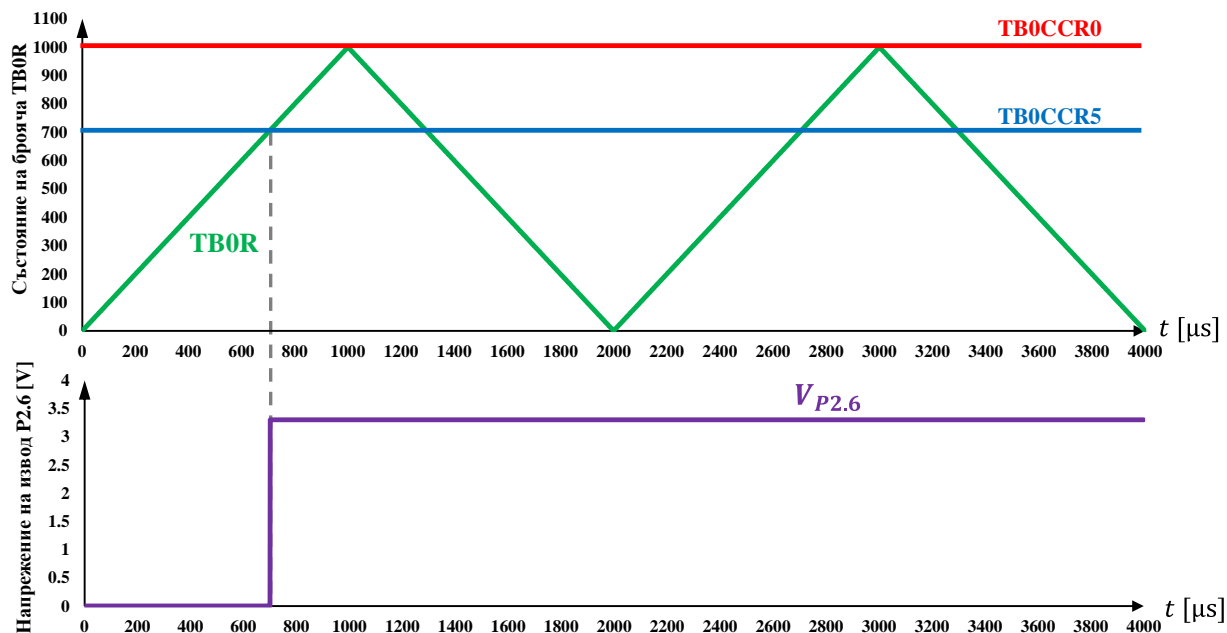
Нека разгледаме един пример. Нека броячът да е в реверсивен режим (Up Mode). Броячът се тактува с **тактов сигнал с честота 1 MHz (период 1  $\mu$ s)**. Нека да управляваме субмодул 5, който управлява нивото на извод P2.6 на микроконтролера. Нека сме задали следните стойности за TBxCCRn регистрите:

- TB0CCR0 = 1000;
- TB0CCR5 = 700;

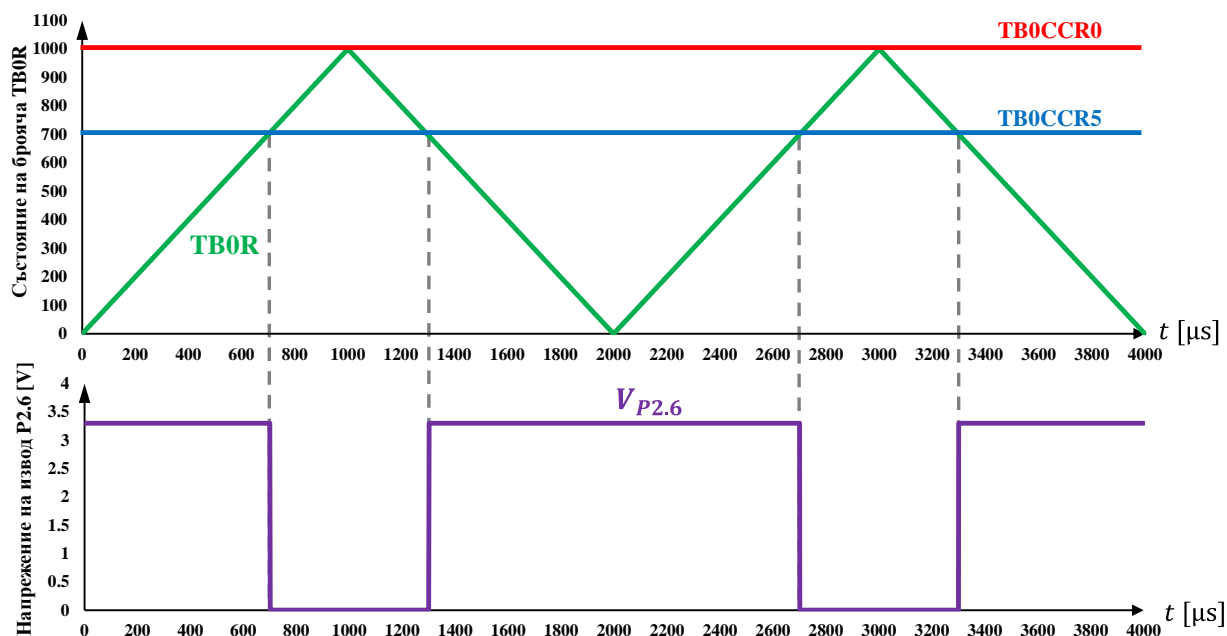
При тези настройки, броячът непрекъснато ще брои от 0 до 1000, като при достигане на състояние 1000 започва от 0.

Чрез битове 5, 6 и 7 на регистър TB0CCTL5 можем да зададем различни варианти по отношение това, как ще се променя нивото на извод P2.6 при съвпадение със стойностите, записани в TB0CCR0 и TB0CCR5:

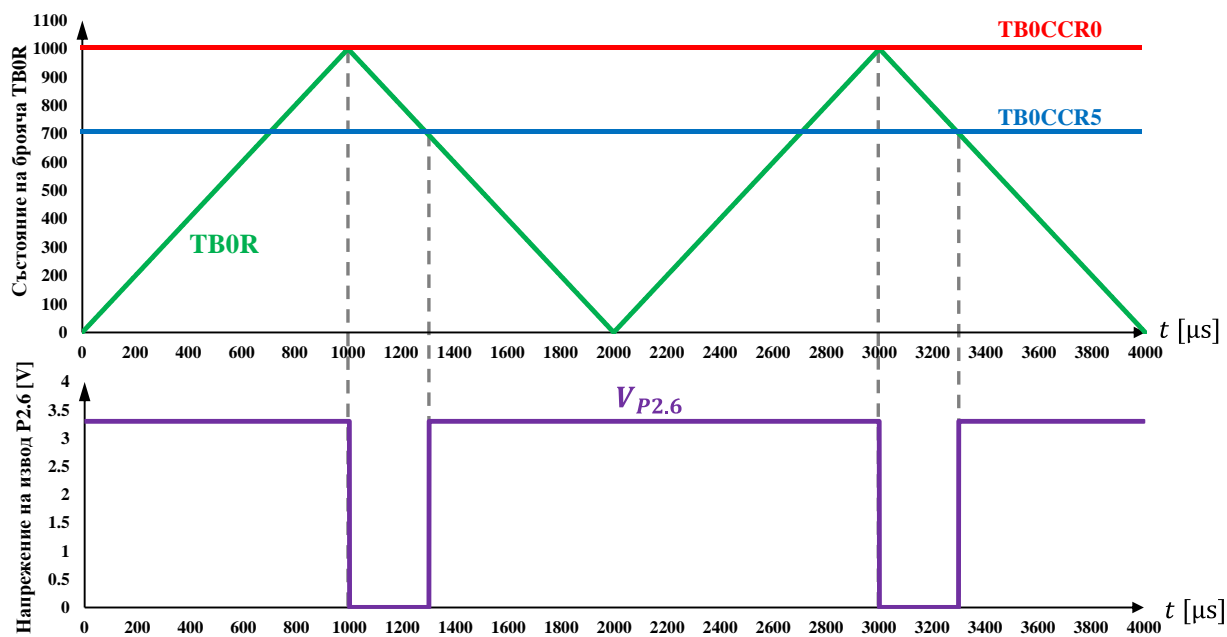
- **Output Mode 1: Set** (фиг. 4.1) - докато състоянието на брояча е по-малко от 700, извод P2.6 е в лог. 0. Когато достигне 700, извод P2.6 преминава в лог. 1 и си стои в това състояние, докато не се промени стойността на TB0CCR5. На извод P2.6 нивото ще бъде лог. 0 в момента на промяна на стойността на TB0CCR5 и след изтичане на **700 периода** на тактовия сигнал ще стане **лог. 1** и си стои в това състояние. При период на тактовия сигнал 1  $\mu$ s, тези 700 периода ще изтекат за 700  $\mu$ s. При този режим, работата е както на чакащ мултивибратор, изработващ единичен импулс с ниско ниво с продължителност 700  $\mu$ s.
- **Output Mode 2: Toggle/Reset** (фиг. 4.2) - когато състоянието на брояча е по-малко от 700, извод P2.6 е в лог. 1. Когато състоянието е между 700 и 1000, извод P2.6 е в лог. 0. На извод P2.6 нивото ще бъде **лог. 0** за  $(1000 - 700) \times 2 = \mathbf{600}$  **периода** и **лог. 1** за  $700 \times 2 = \mathbf{1400}$  **периода**. Т.е. ще имаме сигнал с период 2000  $\mu$ s (честота 500 Hz) и време на импулса 1400  $\mu$ s. Коефициентът на запълване е  $\frac{1400}{2000} \times 100 = 70 \%$ .
- **Output Mode 3: Set/Reset** (фиг. 4.3) - когато състоянието на брояча е по-малко от 1000 и брои „нагоре“, извод P2.6 е в **лог. 1**. Когато състоянието е между 1000 и 700 и брои „надолу“, извод P2.6 е в **лог. 0**. На извод P2.6 нивото ще бъде лог. 0 за  $(1000 - 700) = 300$  периода и **лог. 1** за  $700 \times 2 + (1000 - 700) = 1700$  периода. Т.е. ще имаме сигнал с период 2000  $\mu$ s (честота 500 Hz) и време на импулса 1700  $\mu$ s. Коефициентът на запълване е  $\frac{1700}{2000} \times 100 = 85 \%$ .
- **Output Mode 4: Toggle** (фиг. 4.4) - когато състоянието на брояча е по-малко от 700, извод P2.6 е в **лог. 0**. Когато състоянието е между 700 и 1000, извод P2.6 е в **лог. 1**. На извод P2.6 нивото ще бъде **лог. 1** за  $(1000 - 700) \times 2 = \mathbf{600}$  **периода** и **лог. 0** за  $700 \times 2 = \mathbf{1400}$  **периода**. Т.е. ще имаме сигнал с период 2000  $\mu$ s (честота 500 Hz) и време на импулса 600  $\mu$ s. Коефициентът на запълване е  $\frac{600}{2000} \times 100 = 30 \%$ .
- **Output Mode 6: Toggle/Set** - същото като при Output Mode 4: Toggle.
- **Output Mode 7: Reset/Set** (фиг. 4.5) - когато състоянието на брояча е по-малко от 1000 и брои „нагоре“, извод P2.6 е в **лог. 0**. Когато състоянието е между 1000 и 700 и брои „надолу“, извод P2.6 е в **лог. 1**. На извод P2.6 нивото ще бъде **лог. 1** за  $(1000 - 700) = \mathbf{300}$  **периода** и **лог. 0** за  $700 \times 2 + (1000 - 700) = \mathbf{1700}$  **периода**. Т.е. ще имаме сигнал с период 2000  $\mu$ s (честота 500 Hz) и време на импулса 300  $\mu$ s. Коефициентът на запълване е  $\frac{300}{2000} \times 100 = 15 \%$ .



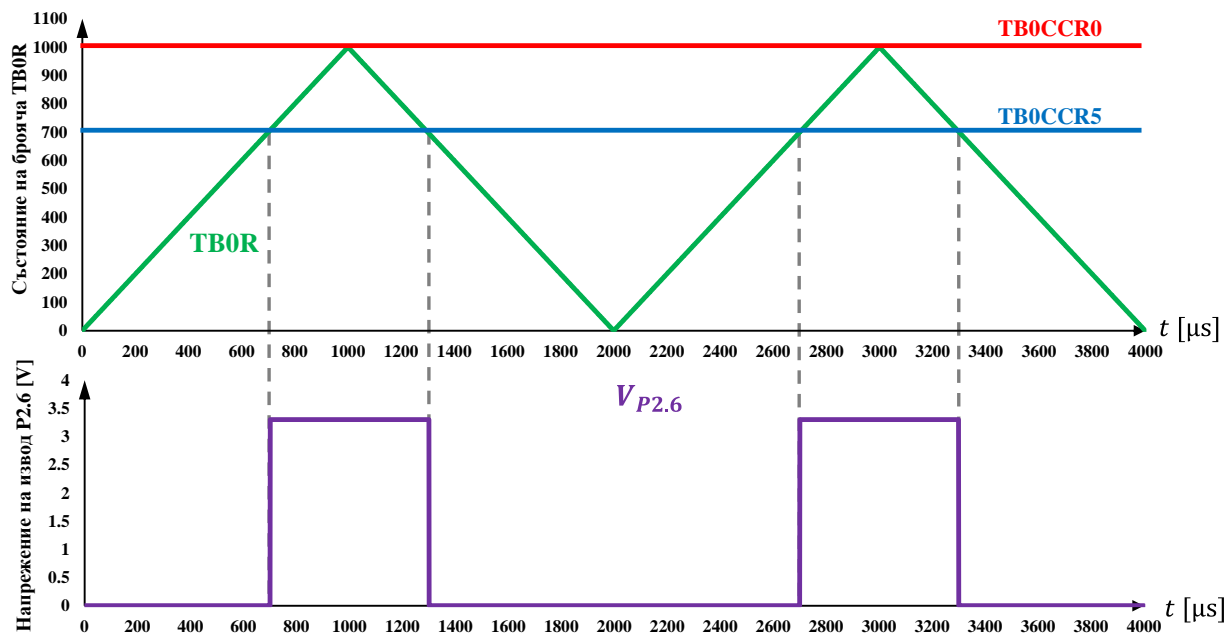
Фиг. 4.1. Output Mode 1: Set



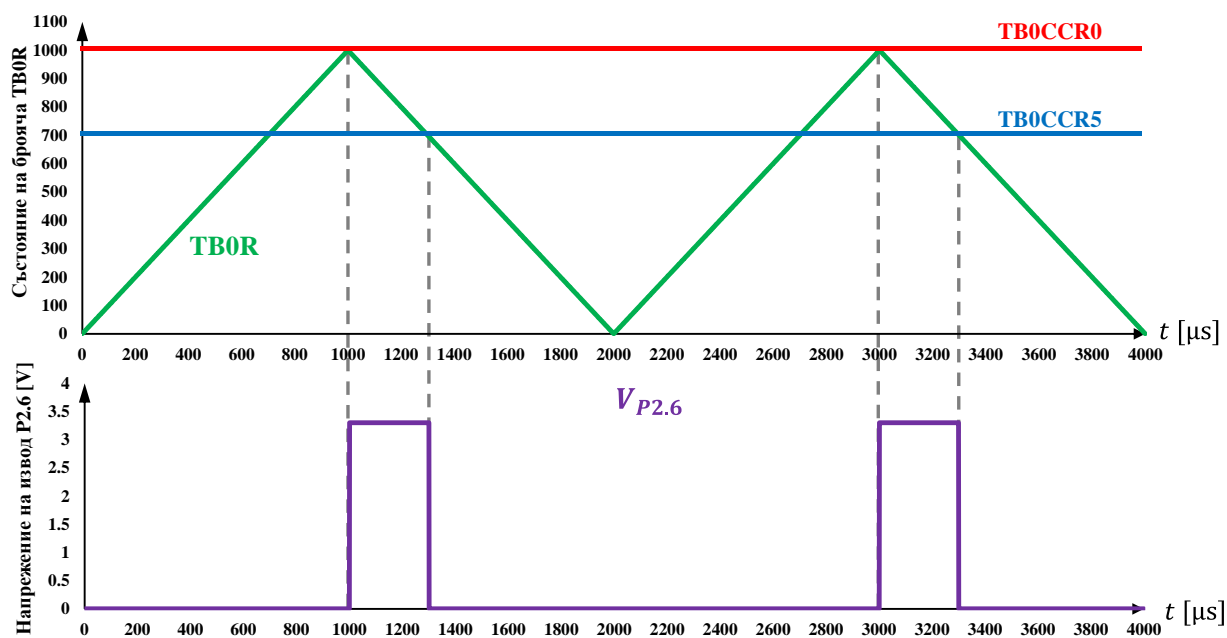
Фиг. 4.2. Output Mode 2: Toggle/Reset



Фиг. 4.3. Output Mode 3: Set/Reset



Фиг. 4.4. Output Mode 4: Toggle



Фиг. 4.5. Output Mode 7: Reset/Set

## ЗАДАЧИ ЗА ИЗПЪЛНЕНИЕ

1. Да се създаде нов проект с име **Lab\_4\_1** в папка **/Desktop/MSHT\_GR\_XX\_N**, да се копира програмата, чрез която мига светодиод D1 от демо платката с помощта на таймерния модул. Задайте 100 милисекунди времеинтервал. Нека тактовата честота на DCO генератора е 8 MHz. Нека тактовата честота на ядрото и всички тактови сигнали на микроконтролера да се конфигурират на 8 MHz (ACLK = VLOCLK, SMCLK = MCLK = DCO).

2. Да се създаде нов проект с име **Lab\_4\_2** в папка **/Desktop/MSHT\_GR\_XX\_N** и да се въведе програмата на C, чрез която се генерира PWM сигнал на извод P2.6 с честота 1 kHz и коефициент на запълване 50 %. Да се свърже осцилоскоп на този извод.

3. Да се зададе коефициент на запълване 5 %, 25 % и 75 %. Да се снимат осцилограмите.

4. Да се промени честотата на PWM сигнала.

5. Да се създаде нов проект с име **Lab\_4\_3** в папка **/Desktop/MSHT\_GR\_XX\_N**, да се копира програмата на C, чрез която се генерират два PWM сигнала в противофаза на P2.6 и P2.7. Да се анализира работата и да се снимат осцилограмите.

### Използвана литература:

1. User guide на MSP430FR6989