

Лабораторная работа №2

Прерывания. Таймеры

Задание на ЛР2

Запрограммировать кнопки и светодиоды в соответствии с вариантом задания, используя прерывания и таймеры

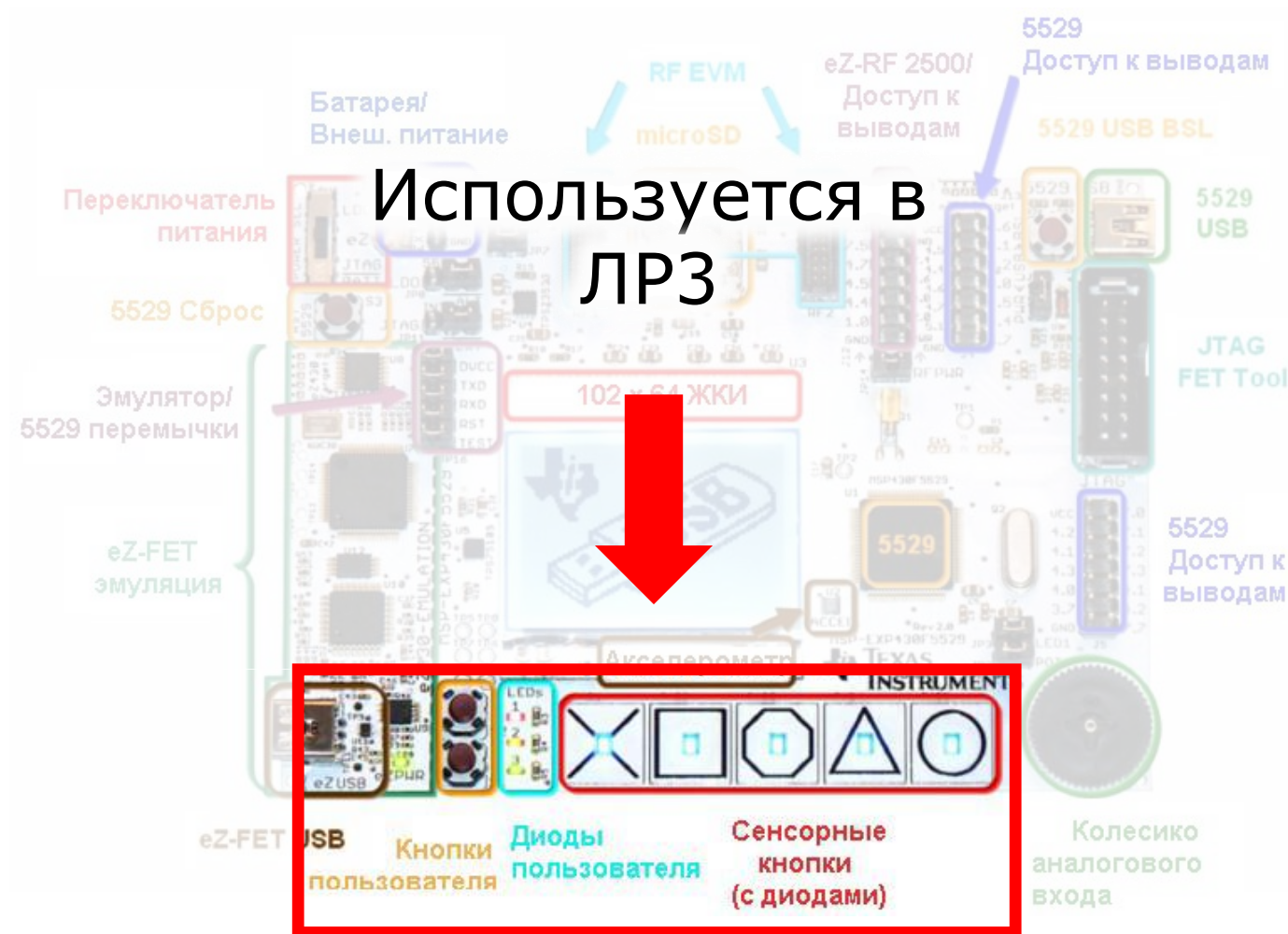
НЕЛЬЗЯ ИСПОЛЬЗОВАТЬ:

Опрос флагов состояния в цикле

Активное ожидание (циклы задержки)

Высокоуровневые библиотеки

Плата MSP-EXP430F5529



Прерывание

Событие, требующее немедленной реакции со стороны процессора (ядра)

Запрос на прерывание – сигнал, по которому управление передается подпрограмме-обработчику прерывания

Вектор прерывания – номер (адрес) обработчика прерывания

Таблица векторов прерываний – таблица в памяти, в которой хранятся адреса функций обработки прерывания

Типы прерываний

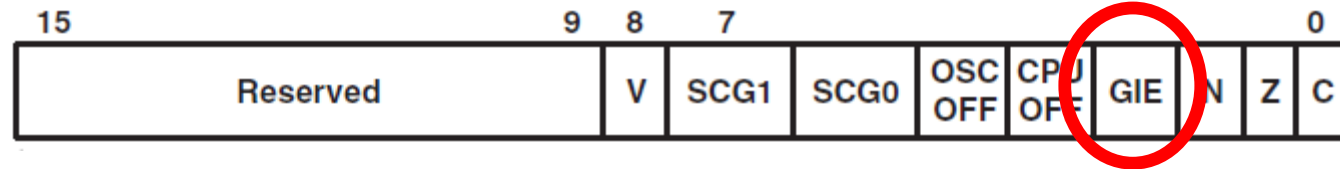
- системные немаскируемые (сигнал RST/NMI в режиме NMI, сигнал от сторожевого таймера, сбой тактового генератора, ошибка доступа Flash памяти);
- пользовательские немаскируемые (сбой напряжения питания от РММ, доступ к несуществующей памяти, события буфера JTAG);
- маскируемые прерывания (генерируются периферийными модулями) – необходимо установить флаг разрешения пользовательских маскируемых прерываний

Правила «хорошего тона»

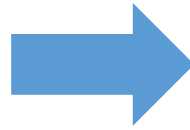
Необходимо переводить контроллер в режим низкого энергопотребления когда он находится в режиме ожидания прерывания (т.е. когда управление отдано функции main)

```
__bis_SR_register(LPM0_bits);
```

Регистр состояния SR



~~$SR \mid= mask$~~



`__bis_SR_register(mask);`

Разрешить пользовательские
маскируемые прерывания!

Обработка прерывания

1. Завершается выполнение текущей инструкции.
2. Счетчик команд PC, указывающий на следующую инструкцию, сохраняется в стеке;
3. Регистр состояния SR сохраняется в стеке.
4. Выбирается прерывание с наивысшим приоритетом.
5. Сбрасывается флаг запроса на прерывания, если данному вектору соответствует единственный источник. Если источников несколько, флаг запроса на прерывание необходимо сбрасывать программно.
6. Все биты регистра состояния SR кроме SCG0 сбрасываются в 0 (т.к. бит GIE = 0 маскируемые прерывания запрещаются)
7. В счетчик команд PC загружается содержимое выбранного вектора прерывания.

Типы регистров (необходимых для ЛР2)

PxIN – чтение данных с вывода;

PxOUT – установка значения выхода;

PxDIR – выбор направления: 0 – вход, 1 – выход;

PxREN – разрешение подтягивающего резистора;

PxSEL – выбор функции вывода: 0 – I/O, 1 – периферия;

PxIES – выбор направления перепада для генерации запроса на прерывание: 0 – по фронту, 1 – по спаду;

PxIE – разрешение прерывания;

PxIV – значение смещения счетчика команд для обработки

PxIFG – флаг прерывания.

Прерывания от портов I/O

Прерывания от ВСЕХ пинов в порте объединены в
ОДИН вектор прерывания

Чтобы определить, какая ножка вызвала прерывание,
необходимо обращение к регистру PxIV (обращение к
данному регистру сбрасывает флаг запроса на
прерывание для активного прерывания с самым
высоким приоритетом)

Если известно, что прерывание может вызвать только
один из пинов порта, можно просто сбросить флаг
запроса на прерывание для этого пина

Работа с прерываниями I/O портов

1. Инициализация порта для работы с кнопкой (см. ЛР1)
2. Разрешение прерывания ($PxIE \neq 0$)
3. Выбор направления перепада для генерации запроса (если необходимо)
4. Сброс флага прерывания (лучше это делать явно всегда в конце инициализации)

Обработчик прерывания I/O портов

```
#pragma vector = PORTx_VECTOR  
__interrupt void ISR(void)  
{...}
```

x – номер порта (1 для S1 и 2 для S2)

ISR – название функции обработчика прерывания

Особенности

PxIFG не сбрасывается автоматически при завершении обработчика прерывания

Запись в регистры PxOUT, PxDIR и PxREN может быть причиной установки значений в регистре PxIFG

Не запрещается изменять направление перепада для генерации прерывания внутри обработчика (если это необходимо)

Необходимо помнить о необходимости разрешения пользовательских маскируемых прерываний

Сторожевой таймер

Сторожевой или интервальный режим

Доступ к регистру управления защищен паролем

Возможность выбора источника тактовых импульсов

Возможность деления частоты

Возможность отключения для уменьшения
энергопотребления

Счетчик таймера не имеет программного доступа (т.е. временными интервалами можно управлять только меняя источник тактирования и делитель частоты)

Режимы сторожевого таймера

Сторожевой

Генерирует сигнал сброса
PUC

Может использоваться для
определения
программных сбоев

Интервальный

Генерирует сигнал
запроса на прерывание

Может использоваться для
генерации интервалов
времени (обработка
исключительных ситуаций
ложится на программиста)

Источники тактирования для сторожевого таймера

SMCLK – 1 МГц

ACLK – 32кГц

VLOCLK – 10кГц

X_CLK – 32кГц (или 10 кГц)

Обработчик прерывания сторожевого таймера в интервальном режиме

```
#pragma vector = WDT_VECTOR  
__interrupt void ISR(void)  
{...}
```

ISR – название функции обработчика прерывания

Конфигурирование сторожевого таймера

- Регистр WDTCTL:
 - WDTPW – пароль доступа
 - WDT HOLD – остановка таймера
 - WDT SSEL – выбор источника тактирования
 - WDT MSEL – выбор режима
 - WDT CNTCL – очистка счетчика
 - WDT IS – выбор делителя (интервала)
- Регистр IE1:
 - WDT IE – разрешение прерываний
- Регистр IFG1:
 - WDT IFG – флаг прерывания

Настройка интервалов таймера

```
/* WDT is clocked by fSMCLK (assumed 1MHz) */
#define WDT_MDLY_32      (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2)           /* 32ms interval (default) */
#define WDT_MDLY_8       (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTIS0)    /* 8ms      " */
#define WDT_MDLY_0_5     (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTIS1)    /* 0.5ms    " */
#define WDT_MDLY_0_064   (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTIS1+WDTIS0) /* 0.064ms  " */
/* WDT is clocked by fACLK (assumed 32KHz) */
#define WDT_ADLY_1000     (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTISSEL0) /* 1000ms   " */
#define WDT_ADLY_250      (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTISSEL0+WDTIS0) /* 250ms    " */
#define WDT_ADLY_16       (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTISSEL0+WDTIS1) /* 16ms     " */
#define WDT_ADLY_1_9      (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTISSEL0+WDTIS1+WDTIS0) /* 1.9ms    " */
/* WDT is clocked by fSMCLK (assumed 1MHz) */
#define WDT_MRST_32       (WDTPW+WDTCNTCL+WDTIS2)                   /* 32ms interval (default) */
#define WDT_MRST_8        (WDTPW+WDTCNTCL+WDTIS2+WDTIS0)           /* 8ms      " */
#define WDT_MRST_0_5      (WDTPW+WDTCNTCL+WDTIS2+WDTIS1)           /* 0.5ms    " */
#define WDT_MRST_0_064    (WDTPW+WDTCNTCL+WDTIS2+WDTIS1+WDTIS0)    /* 0.064ms  " */
/* WDT is clocked by fACLK (assumed 32KHz) */
#define WDT_ARST_1000     (WDTPW+WDTCNTCL+WDTISSEL0+WDTIS2)        /* 1000ms   " */
#define WDT_ARST_250      (WDTPW+WDTCNTCL+WDTISSEL0+WDTIS2+WDTIS0) /* 250ms    " */
#define WDT_ARST_16       (WDTPW+WDTCNTCL+WDTISSEL0+WDTIS2+WDTIS1) /* 16ms     " */
#define WDT_ARST_1_9      (WDTPW+WDTCNTCL+WDTISSEL0+WDTIS2+WDTIS1+WDTIS0) /* 1.9ms    " */
```

Особенности

После сброса устройства есть около 32мс (по умолчанию) на конфигурацию сторожевого таймера

Модуль WDT должен быть приостановлен перед сменой источника тактирования для предотвращения установки некорректного интервала

Интервал WDT должен быть изменен совместно с $WDTCNTCL = 1$ в одной команде, чтобы избежать немедленной генерации сигнала сброса PUC

В интервальном режиме флаг WDTIFG сбрасывается автоматически после обработки прерывания (но может быть сброшен программно)

Таймер А

Доступно 4 режима счета

Возможность выбора источника тактовых импульсов

Возможность деления частоты

Конфигурируемые регистры захвата/сравнения

Возможность множественного захвата/сравнения

Конфигурируемый выход счетчика

Генерация прерывания при переполнении

Регистр вектора прерываний

Обработчик прерывания таймера

```
#pragma vector = TIMER $x$ _A $n$ _VECTOR  
__interrupt void  $ISR$ (void)  
{...}
```

x – номер таймера Ax

n – номер регистра захвата/сравнения (0 для TAxCSCR0, 1 для всех остальных + флага TAIFG)

ISR – название функции обработчика прерывания

Регистр векторов прерываний TAxIV

Используется для определения источника прерывания

Прерывания распределены по приоритетам (от наивысшего CCR1 до наименьшего CCR4)

Прерывание с наивысшим приоритетом генерирует число в TAxIV, которое может быть добавлено к РС для перехода к необходимому обработчику

При обращении к TAxIV происходит автоматический сброс флага с наивысшим приоритетом

Конфигурирование таймера A

- TAxCTL – регистр управления таймером
- TAxCCTLn – регистр управления захватом/сравнением
- TAxCCRn – регистры захвата/сравнения
- TAxIV – регистр векторов прерывания
- TAxR – регистр счетчика таймера

Конфигурирование регистра управления TAхCTL (для LP2)

- TASSEL – выбор источника тактирования (В LP2 - ACLK или SMCLK)
- ID – делитель частоты тактового сигнала
- MC – режим счетчика
- TACLR – сброс таймера (сбрасывает TAхR, IDх и MCх)
- TAIE – разрешение прерываний таймера
- TAIFG – флаг прерывания таймера

Конфигурирование регистра захвата/сравнения TAxCSSTLn (для ЛР2)

- CAP – выбор режима захвата/сравнения
- CCIE – разрешение прерываний захвата/сравнения
- CCIFG – флаг прерывания захвата/сравнения

Особенности

Рекомендуется останавливать таймер перед изменением режима его работы

TAxCCR0 можно менять во время работы таймера (счет вверх).
Если новое значение меньше текущего – таймер обнуляется
(может занять один дополнительный такт)

TAIFG устанавливает при переполнении счетчика (т.е. переходе из TAxCCR0 в 0 или FFFFh в 0 в зависимости от режима)

Флаг TAxCCR0 CCIFG сбрасывается автоматически

Остальные флаги необходимо сбрасывать программно
(обращение к TAxIV)

Вектор TIMERx_A1_VECTOR отвечает за обработку прерываний от всех регистров захвата/сравнения (определение источника осуществляется по TAxIV)

Вопросы на защите

1. Прерывания (терминология, порядок обработки, особенности обработки для портов,
2. Сторожевой таймер (функции, назначение, режимы)
3. Таймеры А и В (функции, назначение, режимы счета, режимы захвата и сравнения, режимы выхода)
4. Часы реального времени

Список обязательных вопросов

1. Что такое прерывание?
2. Порядок обработки прерывания в MSP430?
3. Что такое таймер?
4. Отличие сторожевого и интервального режима WDT.
5. Характеристика режимов счета ТА
6. Для чего нужны регистры захвата/сравнения и выход таймера?
7. Что представляет собой RTC и для чего он нужен?

Вопросы?