|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Институт искусственного интеллекта

Кафедра проблем управления

**ЛАБОРАТОРНАЯ РАБОТА №2**

по дисциплине **Микропроцессорная техника в мехатронике и**

**робототехнике**

**Тема лабораторной работы:** «Переключения светодиода по нажатию по

кнопке через прерывание»

|  |  |
| --- | --- |
| **Студенты группы:** КРБО-03-23 | Зенина А. А. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
|  | Грачев А. А. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
|  | Гришаев А. К. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **Преподаватель:** | ассистент Бессонова А. В. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

|  |  |
| --- | --- |
| Работа представлена к защите: | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2025 г. |

Москва 2025

# **1. Цель работы**

Организация переключения светодиода по нажатию кнопки через прерывание.

# **2. Задача работы**

1. Создать проектный файл, в котором подключить к выбранному микроконтроллеру кнопку с подтяжкой либо к питанию, либо к земле, а также светодиод через токоограничивающий резистор.

2. Написать программу, которая по нажатию на кнопку переключает светодиод.

3. Добавить в обработчик прерывания программный способ подавления дребезга.

# **3. Теоретические сведения**

Микроконтроллер общается с внешним миром посредством портов ввода/вывода. Порт представляет собой совокупность выводов микроконтроллера, объединенных в группу. Все линии портов могут программироваться на ввод или вывод данных независимо друг от друга.

Обращение к портам производится через регистры ввода/вывода. Под каждый порт Рх в адресном пространстве ввода/вывода зарезервировано 3 адреса, по которым размещены следующие регистры: PORTx, DDRx и PINx.

DDRx – регистр направления передачи данных. Если бит установлен в 1, то соответствующий вывод порта является выходом, если же сброшен в 0 – входом.

Бит регистра PORTx выполняет двойную функцию. Если вывод функционирует как выход, то этот бит определяет состояние вывода порта. Если бит установлен в 1, на выводе устанавливается напряжение высокого уровня. Если бит сброшен в 0, на выводе устанавливается напряжение низкого уровня. Если же вывод сконфигурирован как вход, то бит порта PORTx определяет состояние внутреннего подтягивающего резистора для данного вывода.

Прерывание прекращает нормальный ход работы программы для выполнения приоритетной задачи, определяемой внутренним или внешним событием МК. При возникновении прерывания после сохранения в стеке текущего значения счетчика команд происходит выполнение команды, расположенной по адресу соответствующего вектора.

Дребезг контактов - это множественные переключения из одного состояния в другое в течении некоторого времени при нажатии кнопки. Дребезг можно подавить как аппаратно (за счёт фильтрации), так и программно. Идея программного подавления дребезга заключается в введении проверки уровня логического сигнала на входе спустя некоторое время после срабатывания прерывания.

# **4. Расчетно-графическая часть**

4.1. Подключения компонентов

Описание подключения:

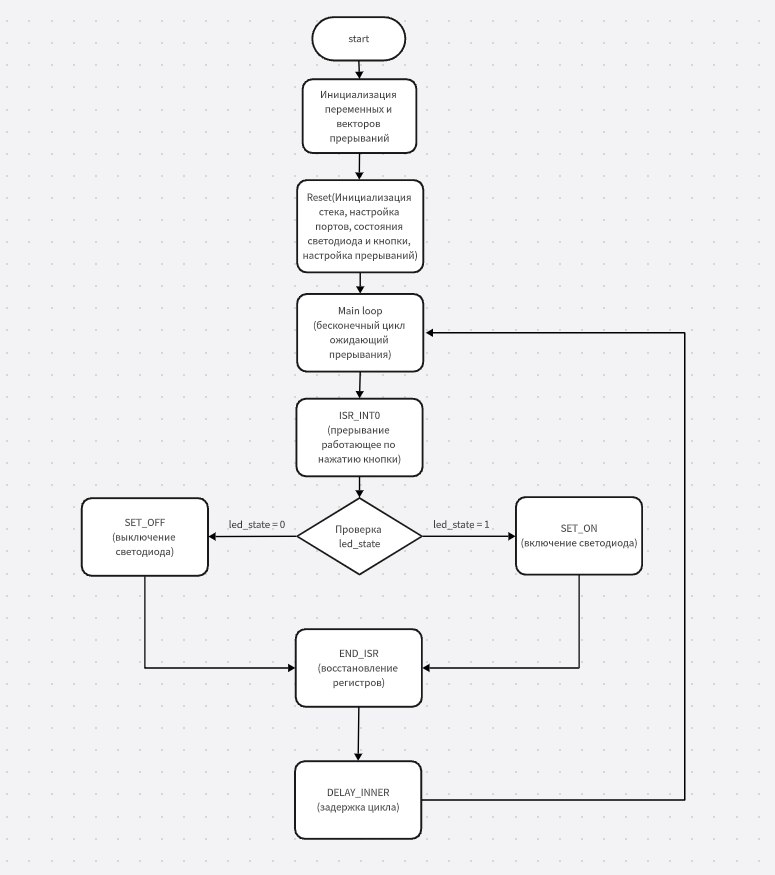
- Светодиод подключен через токоограничивающий резистор 330 Ом к выводу PB0 микроконтроллера

- Кнопка подключена к выводу PD2 (INT0) с использованием внутреннего подтягивающего резистора

- Для зажигания светодиода необходимо сформировать на выводе PB0 сигнал низкого логического уровня

4.2. Алгоритм программы

Блок-схема алгоритма программы представлена на Рисунке 1.



*Рисунок 1 – Блок-схема алгоритма*

Описание алгоритма:

1. Инициализация портов:

- Настройка PB0 как выход для управления светодиодом

- Настройка PD2 как вход для подключения кнопки

- Включение внутреннего подтягивающего резистора для PD2

2. Настройка прерывания INT0:

- Настройка прерывания по спадающему фронту

- Разрешение прерывания INT0

- Глобальное разрешение прерываний

3. Основной цикл программы ожидает прерывания

4. Обработчик прерывания INT0:

- Вызов функции подавления дребезга

- Проверка состояния кнопки после задержки

- Переключение состояния светодиода

- Сброс флага прерывания

4.3. Программное подавление дребезга

Для подавления дребезга контактов реализована функция задержки с последующей проверкой уровня сигнала:

- При срабатывании прерывания производится задержка 10-20 мс

- После задержки проверяется уровень сигнала на пине PD2

- Если уровень соответствует нажатой кнопке, выполняется переключение светодиода

- Это предотвращает множественные срабатывания прерывания при дребезге контактов

Таблица 1 – Настройка условий генерации прерывания INT0

|  |  |  |
| --- | --- | --- |
| Бит ISC01 | Бит ISC00 | Условие срабатывания |
| ISC01 | ISC00 | Условие генерации прерывания |
| 0 | 0 | По низкому уровню на выводе INT0 |
| 0 | 1 | По любому изменению логического уровня |
| 1 | 0 | По спадающему фронту на выводе INT0 |
| 1 | 1 | По нарастающему фронту на выводе INT0 |

# **5. Выводы по работе**

В ходе лабораторной работы была успешно реализована организация переключения светодиода по нажатию кнопки через прерывание. Были изучены принципы работы портов ввода-вывода микроконтроллера и механизм прерываний. Реализовано программное подавление дребезга контактов, что обеспечивает надежную работу системы. Цель работы достигнута.

**ПРИЛОЖЕНИЕ 1**

Код программы.

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; BASIC .ASM template file for AVR

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.include "C:\VMLAB\include\m8def.inc" ; Подключение файла с определениями для ATmega8

; Define here the variables

; Объявление переменных

.def temp = r16 ; Временная переменная для общих операций

.def led\_state = r17 ; Переменная для хранения состояния светодиода (0-выкл, 1-вкл)

; Векторы прерываний

.org 0x0000 ; Адрес вектора сброса

rjmp RESET ; Переход к основной программе при сбросе

.org INT0addr ; Адрес вектора прерывания INT0

rjmp ISR\_INT0 ; Переход к обработчику прерывания INT0

RESET:

; Инициализация стека

ldi temp, low(RAMEND) ; Загрузка младшего байта конца RAM

out SPL, temp ; Установка младшего байта указателя стека

ldi temp, high(RAMEND) ; Загрузка старшего байта конца RAM

out SPH, temp ; Установка старшего байта указателя стека

; Настройка портов

cbi DDRD, PD2 ; PD2 как вход (кнопка INT0) - очистка бита в регистре направления

sbi DDRB, PB0 ; PB0 как выход (светодиод) - установка бита в регистре направления

; Включить подтягивающий резистор для кнопки

sbi PORTD, PD2 ; Установка бита для включения подтягивающего резистора

; Начальное состояние - светодиод ВЫКЛЮЧЕН

sbi PORTB, PB0 ; Установка высокого уровня на PB0 (светодиод выключен)

ldi led\_state, 0 ; 0 - выключен (инициализация переменной состояния)

; Настройка прерывания INT0 по спадающему фронту

ldi temp, (1 << ISC01) | (0 << ISC00) ; ISC01=1, ISC00=0 - прерывание по спадающему фронту

out MCUCR, temp ; Запись в регистр управления микроконтроллера

; Разрешить прерывание INT0

ldi temp, (1 << INT0) ; Установка бита INT0 для разрешения прерывания

out GICR, temp ; Запись в регистр управления прерываниями

sei ; Разрешение глобальных прерываний (установка флага I в SREG)

MAIN\_LOOP:

rjmp MAIN\_LOOP ; Бесконечный пустой цикл (ожидание прерываний)

; Обработчик прерывания INT0

ISR\_INT0:

push temp ; Сохранение временной переменной в стек

in temp, SREG ; Сохранение регистра состояния в стек

push temp

; Сброс флага прерывания

ldi temp, (1 << INTF0) ; Запись 1 в бит INTF0 для его сброса

out GIFR, temp ; Запись в регистр флагов прерываний

; Защита от дребезга - вызов задержки 20ms

rcall DELAY\_20MS

; Проверяем, что кнопка действительно нажата (низкий уровень)

sbic PIND, PD2 ; Пропустить следующую инструкцию если бит PD2 установлен (кнопка не нажата)

rjmp END\_ISR ; Если кнопка не нажата - выходим из прерывания

; Переключаем состояние светодиода

cpi led\_state, 1 ; Сравнение текущего состояния с 1 (включен)

breq SET\_OFF ; Если равно 1 - переходим к выключению

SET\_ON:

; Включаем светодиод (низкий уровень на PB0)

cbi PORTB, PB0 ; Очистка бита PB0 - установка низкого уровня

ldi led\_state, 1 ; Установка состояния "включен"

rjmp END\_ISR ; Переход к концу обработчика

SET\_OFF:

; Выключаем светодиод (высокий уровень на PB0)

sbi PORTB, PB0 ; Установка бита PB0 - установка высокого уровня

ldi led\_state, 0 ; Установка состояния "выключен"

END\_ISR:

pop temp ; Восстановление регистра состояния из стека

out SREG, temp

pop temp ; Восстановление временной переменной из стека

reti ; Возврат из прерывания

; Задержка ~20ms

DELAY\_20MS:

push r24 ; Сохранение регистров в стек

push r25

ldi r25, 100 ; Загрузка внешнего счетчика (100 итераций)

DELAY\_LOOP:

ldi r24, 200 ; Загрузка внутреннего счетчика (200 итераций)

DELAY\_INNER:

dec r24 ; Декремент внутреннего счетчика

brne DELAY\_INNER ; Продолжать если не ноль

dec r25 ; Декремент внешнего счетчика

brne DELAY\_LOOP ; Продолжать если не ноль

pop r25 ; Восстановление регистров из стека

pop r24

ret ; Возврат из подпрограммы