

Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
Кафедра проектирования информационно-компьютерных систем

Отчет
По лабораторной работе №2

Исследование устройств динамической индикации

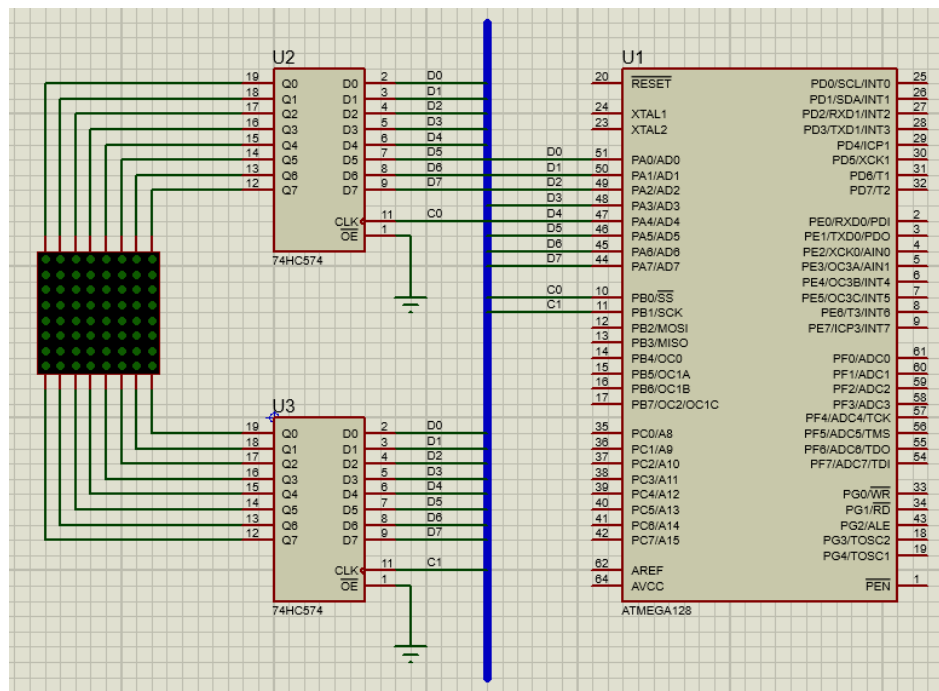
Выполнил:
Студент ФКП,
гр.914302
Воробей Д.А.

Проверил: Ячин Н.С.

Минск 2021

Цель: Исследовать устройство динамической индикации

Принципиальная схема в Proteus



ИСХОДНЫЙ КОД

/main.c/

```
#include <C:\Users\dariavarabei\4\avr\include\avr\sfr_defs.h>
```

```
#include <C:\Users\dariavarabei\4\avr\include\avr\iom128.h>
```

```
#include <C:\Users\dariavarabei\4\avr\include\avr\io.h>
```

```
#define REG_STRING_ENABLE 0x02
```

```
#define REG_COLUMN_ENABLE 0x01
```

```
#define STR_QUANTITY8
```

```
#define DIGIT_PERIOD 0x1000
```

```
static uint8_t aunFont[][STR_QUANTITY] = {
```

```
{
```

```
(uint8_t)~0x00,
```

```
(uint8_t)~0xE0,
```

```
(uint8_t)~0x3E,
```

```
(uint8_t)~0x22,
```

```
(uint8_t)~0x22,
```

```
(uint8_t)~0x3E,
```

```
(uint8_t)~0xE0,
```

```

        (uint8_t)~0x00,
    },
    {
        (uint8_t)~0x00,
        (uint8_t)~0xFC,
        (uint8_t)~0x12,
        (uint8_t)~0x11,
        (uint8_t)~0x11,
        (uint8_t)~0x12,
        (uint8_t)~0xFC,
        (uint8_t)~0x00,
    },
    {
        (uint8_t)~0x00,
        (uint8_t)~0xFF,
        (uint8_t)~0x80,
        (uint8_t)~0xFF,
        (uint8_t)~0xFF,
        (uint8_t)~0x80,
        (uint8_t)~0xFF,
        (uint8_t)~0x00,
    },
    {
        (uint8_t)~0x00,
        (uint8_t)~0xFC,
        (uint8_t)~0x12,
        (uint8_t)~0x11,
        (uint8_t)~0x11,
        (uint8_t)~0x12,
        (uint8_t)~0xFC,
        (uint8_t)~0x00,
    },
};

int main ( void ) {
    uint8_t unDispString = 1;
    uint8_t unFontIndex = 0;
    int32_t nDelayIn, nDelayOut;
    int32_t nDigit = 0;

```

```

DDRA = 0xFF;
DDRB = 0x03;
PORTA = 0;
PORTB = 0;
while (1) {
    nDelayOut = DIGIT_PERIOD;
    while (--nDelayOut > 0) {
        PORTA = unDispString;
        PORTB |= REG_STRING_ENABLE;
        PORTB &= (~REG_STRING_ENABLE);
        unDispString <<= 1;
        if (!unDispString) {
            unDispString = 1;
        }
        PORTA = aunFont[nDigit][unFontIndex++];
        PORTB |= REG_COLUMN_ENABLE;
        PORTB &= (~REG_COLUMN_ENABLE);
        unFontIndex &= (STR_QUANTITY - 1);

        for (nDelayIn = 0; nDelayIn < 100; nDelayIn++) {
        }

    }
    nDigit++;
    nDigit %= (sizeof(aunFont) / STR_QUANTITY / sizeof(uint8_t));
}
return 0;
}

```

/makefile/

TARGET=LED_Display

Define commands

CC=avr-gcc

OBJCOPY=avr-objcopy

SIZE = avr-size

REMOVE= rm -f

COPY= cp -f

MOVE= mv -f

REMOVEDIR = rmdir

MAKEDIR= mkdir

DEBUG=-gdwarf-2

OPTIM=-O0

M_CPU=atmega128

LINKER_FLAGS= -I. -Wl,-Map,\$(TARGET).map -mmcu=\$(M_CPU)

CFLAGS= -I. \

-D _AVR_IO_H_ \

-mmcu=\$(M_CPU) -Wall \$(OPTIM) \$(DEBUG) \

-std=gnu99 \

-funsigned-char -funsigned-bitfields -fpack-struct \

-fshort-enums -MMD -MP

SOURCE= main.c

OBJS = \$(SOURCE:.c=.o)

all: begin Debug \$(TARGET).bin size finished move end

\$(TARGET).bin : \$(TARGET).elf

\$(OBJCOPY) -O ihex -R .eeprom -R .fuse -R .lock -R .signature \$(TARGET).elf \$(TARGET).hex

\$(TARGET).elf : \$(OBJS)

\$(CC) \$(LINKER_FLAGS) -o \$(TARGET).elf \$(OBJS)

\$(OBJS) : %.o : %.c

\$(CC) \$(CFLAGS) -c -o \$\$@ \$<

size:

\$(SIZE) --format=berkeley \$(TARGET).elf

Define Messages

English

MSG_ERRORS_NONE = Errors: none

MSG_MOVE = ----- Moving output files into Debug folder -----

MSG_BEGIN = ----- begin -----

MSG_END = ----- end -----

begin:

@echo

@echo \$(MSG_BEGIN)

finished:

```

        @echo $(MSG_ERRORS_NONE)

move: Debug

        @echo
        @echo $(MSG_MOVE)
        @echo
        $(MOVE) *.d Debug
        $(MOVE) $(OBJS) Debug
        $(MOVE) $(TARGET).elf Debug
        $(MOVE) $(TARGET).hex Debug
        $(MOVE) $(TARGET).map Debug

end:

        @echo $(MSG_END)
        @echo

Debug:

        $(MAKEDIR) Debug

clean : begin clean_list end

clean_list :

        touch Makefile
        @echo $(MSG_CLEANING)
        $(REMOVE) *.d
        $(REMOVE) $(OBJS)
        $(REMOVE) $(TARGET).hex
        $(REMOVE) $(TARGET).bin
        $(REMOVE) $(TARGET).elf
        $(REMOVE) $(TARGET).map
        $(REMOVE) Debug/*
        $(REMOVEDIR) Debug

```

Успешная сборка проекта:

```

make all

----- begin -----
avr-gcc -I. -D _AVR_IO_H_ -mmcu=atmega128 -Wall -O0 -gdwarf-2 -std=gnu99 -funsigned-char -funsigned-bitfields -fpack-struct -fshort-enun
avr-gcc -I. -Wl,-Map,LED_Display.map -mmcu=atmega128 -o LED_Display.elf main.o
avr-objcopy -O ihex -R .eeprom -R .fuse -R .lock -R .signature LED_Display.elf LED_Display.hex
avr-size --format=berkeley LED_Display.elf
   text    data     bss     dec      hex filename
   686      32        0     718      2ce LED_Display.elf
Errors: none

```

Вывод: с помощью управления портами микроконтроллера ATMEGA128 в режиме ввода-вывода общего назначения на светодиодный дисплей размером 8x8 выведено имя студента.

