Университет ИТМО

Информационно-управляющие системы

Лабораторная работа №2

Выполнили: Шумеев А., Мохнаткин Д.

Р3418

Г. Санкт-Петербург,

2017г.

1. Задание

Контроллер SDK-1.1 на линейку светодиодов циклически отображает приведенную ниже

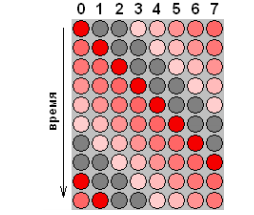
анимацию или выводит количество замыканий входа T0 (счетный вход таймера 0 на рис.

5). Подсчет количества замыканий входа должен быть реализован с помощью таймера-

счетчика 0. Смена режима отображения производится по поступлению сигнала внешнего

прерывания INT0 (рис. 5). В анимации должны использоваться не менее 6 градаций

яркости свечения светодиодов (например, 0%, 20%, 40%, 60%, 80% и 100%).



1. Модель взаимодействия

Прикладная программа

Показ значения счетчика

Режим анимации

Драйвер светодиодов

TH0

Обработчик прерываний Таймера 1

Обработчик прерывания INT0

Драйвер Таймер 2 (отсчет времени)

1. Исходный текст программы

System\_timer.c

#include "system\_timer.h"

#include "led.h"

u16 cnt=0;

u8 levelo;

const u8 LEVEL\_ON=0xF0;

const u8 LEVEL\_OFF=0x0F;

time cur\_ms;

void T2\_ISR( void ) \_\_interrupt ( 2 );

void initialize\_system\_timer() {

cur\_ms=0;

cnt=0;

SetVector( 0x202B, (void \*)T2\_ISR );

TH2=(-1000)>>8;0;

TL2=(-1000)&0xFF;

RCAP2H=(-1000)>>8;

RCAP2L=(-1000)&0xFF;

ET2=1; TR2=1;

}

time get\_ms(void){

return cur\_ms;

}

time get\_ms\_after(time t0){

return cur\_ms-t0;

}

void delay\_ms(time t){

time now=get\_ms();

while( get\_ms\_after(now)<t){}

}

void T2\_ISR( void ) \_\_interrupt ( 2 ){

cur\_ms++;

}

counter.c

#include "counter.h"

void initialize\_counter() {

TMOD|=0b00000100;

TH0=TL0=0;

TR0=1;

}

u8 get\_count() {

return TL0;}

animation.c

#include "animation.h"

#include "led.h"

#include "system\_timer.h"

u8 power=0;

u8 cur = 0x01;

u8 counter;

u8 level;

//величивать или уменьшать ¤ркость

u8 direction;

//врем¤ следующего кадра

time next\_frame;

const u8 LEVEL\_ON=0xF0;

const u8 LEVEL\_OFF=0x0F;

const time FRAME\_DELAY=9000;

void T1\_ISR( void ) \_\_interrupt ( 1 );

void initialize\_animation() {

TMOD|=0b00010000;

SetVector( 0x201B, (void \*)T1\_ISR );

}

void enable\_animation(){

direction=UP;

next\_frame=get\_ms()+FRAME\_DELAY;

cur=1;

level=LEVEL\_ON;

counter = 0;

TH1 = 0xF8;

TL1= 0xFF;

ET1=1; //включаем прерывание от таймера 1

TR1=1; //разрешаем счет таймера 1

}

void disable\_animation(){

ET1=0;

TR1=0;

}

u8 powers[8] ={0,0,6,5,4,3,2,1};

void movePower() {

u8 last = powers[7];

u8 i;

for(i=7; i > 0; i--)

{

powers[i] = powers[i-1];

}

powers[0] = last;

}

void set\_power(u8 k, u8 power){

powers[k]=power;

}

u8 get\_power() {

return power;

}

void updateLeds()

{

u8 i =0;

u8 lights = 0;

u8 mask;

for(i=0; i<8; i++)

{

if(counter % powers[i] == 0) {

lights = lights | (1 << i);

}

}

leds(lights);

}

void T1\_ISR( void ) \_\_interrupt ( 1 ){

TR1=0;

if(get\_ms()>=next\_frame)

{

movePower();

counter = 0;

next\_frame=get\_ms()+FRAME\_DELAY;

}

counter++;

updateLeds();

TH1 = 0xF8;

TL1= 0xFF;

TR1=1;

}

Mode\_selector.c

u8 mode;

void INT0\_ISR( void ) \_\_interrupt ( 5 ); //1

void initialize\_mode\_selector() {

IT1=1;//INT0 (1)

EX1=1;

SetVector( 0x2013, (void \*)INT0\_ISR ); // 1

}

void INT0\_ISR( void ) \_\_interrupt ( 5 ){

if( mode==MODE\_COUNTER ) {

mode=MODE\_ANIMATION;

}else{

mode=MODE\_COUNTER;

}

}

Main.c

#include "aduc812.h"

#include "led.h"

#include "animation.h"

#include "mode\_selector.h"

#include "counter.h"

#include "system\_timer.h"

void delay ( unsigned long ms );

void main( void ) {

initialize\_system\_timer();

initialize\_mode\_selector();

initialize\_animation();

initialize\_counter();

mode=MODE\_COUNTER;

EA=1;

while(1){

if( mode==MODE\_ANIMATION ){

enable\_animation();

while( mode==MODE\_ANIMATION ){

delay\_ms(1);

}

disable\_animation();

}

if( mode==MODE\_COUNTER ){

while( mode==MODE\_COUNTER ){

leds( get\_count() );

delay\_ms(1);

}

}

}

}

1. Вывод

В ходе лабораторной работы была изучена модель прерываний учебного стенда и возможность их использования в своей программе. Также были изучены принципы работы таймеров-счетчиков.

Благодаря возможности установки своих обработчиков на различные виды прерывания, настройки их приоритетов и возможности выбора режима работы таймер и счетчика, можно создавать достаточно гибкие системы управления устройствами и разнообразным поведением.