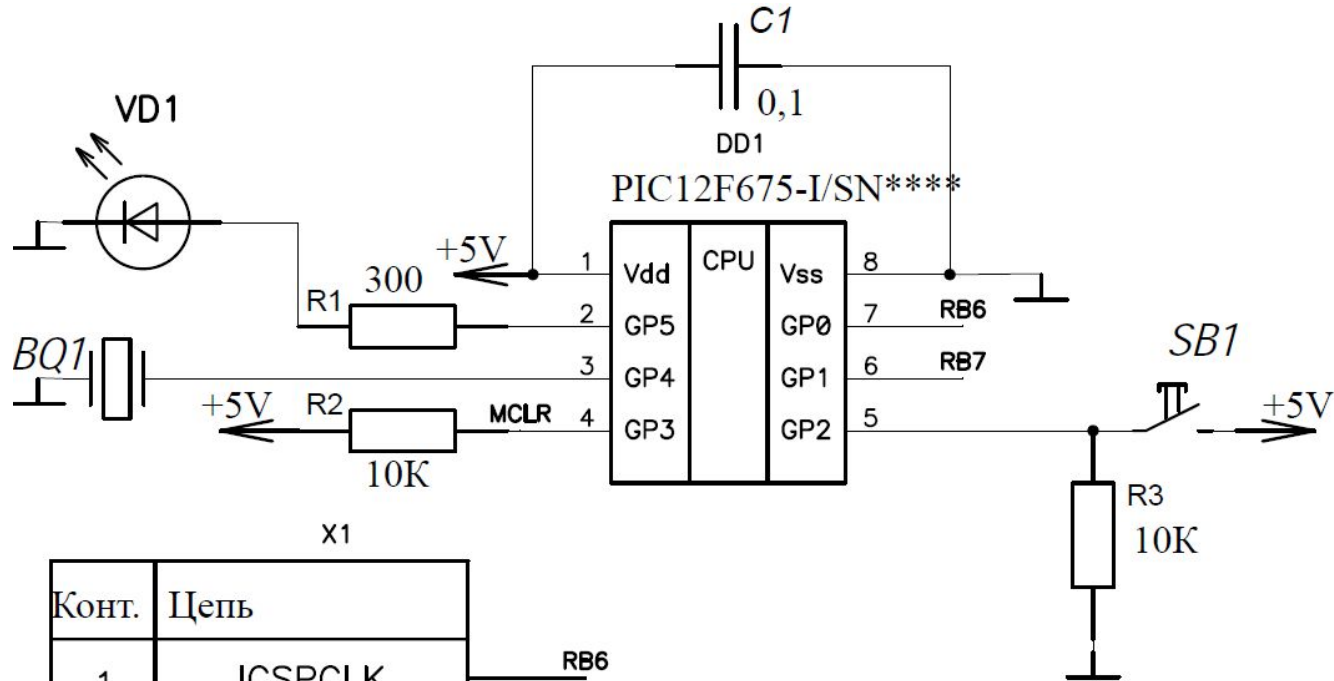


Микроконтроллеры и ПЛИС

Задания

Пример создания программы.



X1

Конт.	Цепь	
1	ICSPCLK	RB6
2	ICSPDAT	RB7
3		
4	GND	
5	+5 V	+5V
6	MCLR	MCLR

Задание.

1. При включении питания звуковой сигнал не генерируется, светодиод светится.
2. При нажатии на кнопку генерируется сигнал частотой 300 Гц, светодиод не светится.
3. При повторном нажатии на кнопку генерация звукового сигнала прекращается, светодиод светится .
4. При нажатии кнопки переход к п. 2.

Анализ задания

1. Порт GP0 не используется, оставляем по умолчанию входом.
2. Порт GP1 не используется, оставляем по умолчанию входом.
3. Порт GP2 вход. При нажатии кнопки SB1 на GP2 подаётся уровень лог. 1.
4. Порт GP3 может быть только входом.
5. Порт GP4 назначаем выходом, для работы пьезоизлучателя BQ1 с частотой 300 Гц необходимо сформировать импульсы на GP4 частотой 300 Гц.
6. Порт GP5 назначаем выходом. Для свечения светодиода на GP5 необходимо подать уровень лог. 1, для выключения светодиода на GP5 необходимо подать уровень лог. 0.

1. Обработка кнопки SB1 на GP2 по прерываниям, прерывание по фронту, подавление дребезга контактов кнопки реализовать программно.
2. Для генерации импульсов частотой 300 Гц на выводе GP4 используем таймер TMR0.
3. Биты конфигурации микроконтроллера: внутренний тактовый RC генератор INTRCCLK. Остальные биты должны быть отключены.

Настройка параметров
периферии микроконтроллера.

Порт ввода-вывода

TABLE 3-1: SUMMARY OF REGISTERS ASSOCIATED WITH GPIO

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOD	Value on all other RESETS
05h	GPIO	—	—	GP5	GP4	GP3	GP2	GP1	GP0	--xx xxxx	--uu uuuu
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	0000 000u
19h	CMCON	—	COU \bar{T}	—	CINV	CIS	CM2	CM1	CM0	-0-0 0000	-0-0 0000
81h	OPTION_REG	\overline{GPPU}	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0	--11 1111	--11 1111
95h	WPU	—	—	WPU5	WPU4	—	WPU2	WPU1	WPU0	--11 -111	--11 -111
96h	IOC	—	—	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	--00 0000	--00 0000
9Fh	ANSEL	—	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0	-000 1111	-000 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by GPIO.

REGISTER 3-1: GPIO — GPIO REGISTER (ADDRESS: 05h)

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	
—	—	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0	
bit 7								bit 0

bit 7-6: **Unimplemented:** Read as '0'

bit 5-0: **GPIO<5:0>:** General Purpose I/O pin.

1 = Port pin is >V_{IH}

0 = Port pin is <V_{IL}

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- При включении питания светодиод должен светиться: GP5=1, остальные выводы порта =0 , GPIO=0b00100000.

REGISTER 2-3: INTCON — INTERRUPT CONTROL REGISTER (ADDRESS: 0Bh OR 8Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF
bit 7						bit 0	

bit 7	GIE: Global Interrupt Enable bit 1 = Enables all unmasked interrupts 0 = Disables all interrupts
bit 6	PEIE: Peripheral Interrupt Enable bit 1 = Enables all unmasked peripheral interrupts 0 = Disables all peripheral interrupts
bit 5	T0IE: TMR0 Overflow Interrupt Enable bit 1 = Enables the TMR0 interrupt 0 = Disables the TMR0 interrupt
bit 4	INTE: GP2/INT External Interrupt Enable bit 1 = Enables the GP2/INT external interrupt 0 = Disables the GP2/INT external interrupt
bit 3	GPIE: Port Change Interrupt Enable bit ⁽¹⁾ 1 = Enables the GPIO port change interrupt 0 = Disables the GPIO port change interrupt
bit 2	T0IF: TMR0 Overflow Interrupt Flag bit ⁽²⁾ 1 = TMR0 register has overflowed (must be cleared in software) 0 = TMR0 register did not overflow
bit 1	INTF: GP2/INT External Interrupt Flag bit 1 = The GP2/INT external interrupt occurred (must be cleared in software) 0 = The GP2/INT external interrupt did not occur
bit 0	GPIF: Port Change Interrupt Flag bit 1 = When at least one of the GP5:GP0 pins changed state (must be cleared in software) 0 = None of the GP5:GP0 pins have changed state

1. Используются прерывания:
глобальное разрешение прерываний-да, GIE=1,
разрешение прерываний INT (GP0)-да, INTE=1,
прерывание от TMR0-да, T0IE=1.
2. Все неиспользуемые прерывания запрещаем, флаги прерываний сбрасываем.
3. INTCON=0b10110000

REGISTER 6-1: CMCON — COMPARATOR CONTROL REGISTER (ADDRESS: 19h)

U-0	R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	COUT	—	CINV	CIS	CM2	CM1	CM0	
bit 7								bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6 **COUT:** Comparator Output bit

When CINV = 0:

1 = $V_{IN+} > V_{IN-}$

0 = $V_{IN+} < V_{IN-}$

When CINV = 1:

1 = $V_{IN+} < V_{IN-}$

0 = $V_{IN+} > V_{IN-}$

bit 5 **Unimplemented:** Read as '0'

bit 4 **CINV:** Comparator Output Inversion bit

1 = Output inverted

0 = Output not inverted

bit 3 **CIS:** Comparator Input Switch bit

When CM2:CM0 = 110 or 101:

1 = V_{IN-} connects to $CIN+$

0 = V_{IN-} connects to $CIN-$

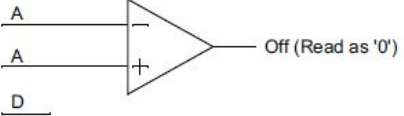
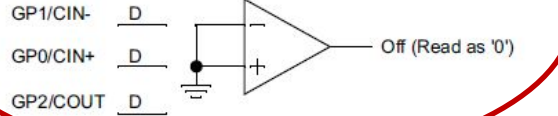
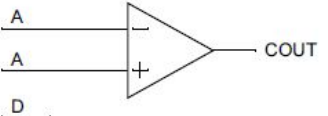
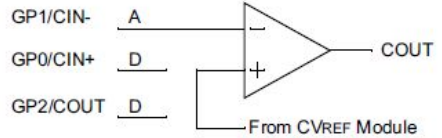
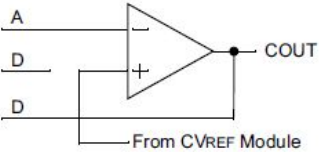
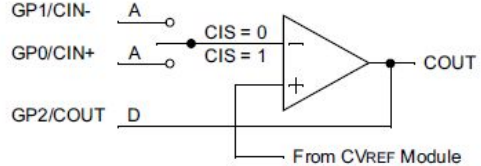
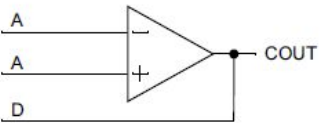
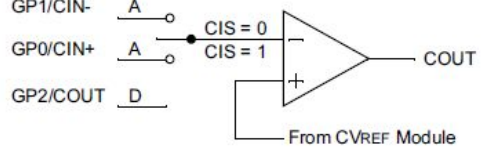
bit 2-0 **CM2:CM0:** Comparator Mode bits

Figure 6-2 shows the Comparator modes and CM2:CM0 bit settings

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Компаратор не используется, все выводы связанные с компаратором должны быть цифровыми. CMCON=0b00000111.

FIGURE 6-2: COMPARATOR I/O OPERATING MODES

<p>Comparator Reset (POR Default Value - low power) CM2:CM0 = 000</p> <p>GP1/CIN- A GP0/CIN+ A GP2/COUT D</p> 	<p>Comparator Off (Lowest power) CM2:CM0 = 111</p> <p>GP1/CIN- D GP0/CIN+ D GP2/COUT D</p> 
<p>Comparator without Output CM2:CM0 = 010</p> <p>GP1/CIN- A GP0/CIN+ A GP2/COUT D</p> 	<p>Comparator w/o Output and with Internal Reference CM2:CM0 = 100</p> <p>GP1/CIN- A GP0/CIN+ D GP2/COUT D</p> 
<p>Comparator with Output and Internal Reference CM2:CM0 = 011</p> <p>GP1/CIN- A GP0/CIN+ D GP2/COUT D</p> 	<p>Multiplexed Input with Internal Reference and Output CM2:CM0 = 101</p> <p>GP1/CIN- A GP0/CIN+ A GP2/COUT D</p> 
<p>Comparator with Output CM2:CM0 = 001</p> <p>GP1/CIN- A GP0/CIN+ A GP2/COUT D</p> 	<p>Multiplexed Input with Internal Reference CM2:CM0 = 110</p> <p>GP1/CIN- A GP0/CIN+ A GP2/COUT D</p> 
<p>A = Analog Input, ports always reads '0' D = Digital Input CIS = Comparator Input Switch (CMCON<3>)</p>	

REGISTER 2-2: OPTION_REG — OPTION REGISTER (ADDRESS: 81h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
GPPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7						bit 0	

- bit 7 **GPPU**: GPIO Pull-up Enable bit
 1 = GPIO pull-ups are disabled
 0 = GPIO pull-ups are enabled by individual port latch values
- bit 6 **INTEDG**: Interrupt Edge Select bit
 1 = Interrupt on rising edge of GP2/INT pin
 0 = Interrupt on falling edge of GP2/INT pin
- bit 5 **T0CS**: TMR0 Clock Source Select bit
 1 = Transition on GP2/T0CKI pin
 0 = Internal instruction cycle clock (CLKOUT)
- bit 4 **T0SE**: TMR0 Source Edge Select bit
 1 = Increment on high-to-low transition on GP2/T0CKI pin
 0 = Increment on low-to-high transition on GP2/T0CKI pin
- bit 3 **PSA**: Prescaler Assignment bit
 1 = Prescaler is assigned to the WDT
 0 = Prescaler is assigned to the TIMER0 module
- bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Подтягивающие резисторы не используются, GPPU=1,
 Прерывание от INT по фронту, INTEDG=1,
 Предделитель подключен к TMR0, PSA=1,
 Коэффициент деления предделителя 1:8, PS0=0, PS1=1, PS2=0.
 OPTION_REG=0b11000010

REGISTER 3-2: TRISIO — GPIO TRISTATE REGISTER (ADDRESS: 85h)

U-0	U-0	R/W-x	R/W-x	R-1	R/W-x	R/W-x	R/W-x	
—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0	
bit 7								bit 0

bit 7-6: **Unimplemented:** Read as '0'

bit 5-0: **TRISIO<5:0>**: General Purpose I/O Tri-State Control bit

1 = GPIO pin configured as an input (tri-stated)

0 = GPIO pin configured as an output.

Note: TRISIO<3> always reads 1.

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

Назначение выводов порта GPIO: GP0 вход, GP1 вход, GP2 вход, GP3 вход, GP4 выход, GP5 выход.

TRISIO=0b00001111

REGISTER 3-3: WPU — WEAK PULL-UP REGISTER (ADDRESS: 95h)

U-0	U-0	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1
—	—	WPU5	WPU4	—	WPU2	WPU1	WPU0
bit 7						bit 0	

- bit 7-6 **Unimplemented:** Read as '0'
- bit 5-4 **WPU<5:4>:** Weak Pull-up Register bit
 1 = Pull-up enabled
 0 = Pull-up disabled
- bit 3 **Unimplemented:** Read as '0'
- bit 2-0 **WPU<2:0>:** Weak Pull-up Register bit
 1 = Pull-up enabled
 0 = Pull-up disabled

- Note 1:** Global $\overline{\text{GPPU}}$ must be enabled for individual pull-ups to be enabled.
Note 2: The weak pull-up device is automatically disabled if the pin is in Output mode (TRISIO = 0).

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Подтягивающие резисторы не используются.

WPU=0

REGISTER 3-4: IOC — INTERRUPT-ON-CHANGE GPIO REGISTER (ADDRESS: 96h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0	
bit 7								bit 0

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **IOC<5:0>:** Interrupt-on-Change GPIO Control bit
 1 = Interrupt-on-change enabled
 0 = Interrupt-on-change disabled

Note 1: Global interrupt enable (GIE) must be enabled for individual interrupts to be recognized.

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

Прерывание по изменению состояния порта не используется.

IOC=0

REGISTER 7-2: ANSEL — ANALOG SELECT REGISTER (ADDRESS: 9Fh)

U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1
—	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'.

bit 6-4 **ADCS<2:0>:** A/D Conversion Clock Select bits

000 = Fosc/2

001 = Fosc/8

010 = Fosc/32

x11 = FRC (clock derived from a dedicated internal oscillator = 500 kHz max)

100 = Fosc/4

101 = Fosc/16

110 = Fosc/64

bit 3-0 **ANS3:ANS0:** Analog Select bits

(Between analog or digital function on pins AN<3:0>, respectively.)

1 = Analog input; pin is assigned as analog input⁽¹⁾

0 = Digital I/O; pin is assigned to port or special function

Note 1: Setting a pin to an analog input automatically disables the digital input circuitry, weak pull-ups, and interrupt-on-change. The corresponding TRISIO bit must be set to Input mode in order to allow external control of the voltage on the pin.

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

- n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

АЦП не используется.

ANSEL=0b00000000

TMR0

TABLE 4-1: REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOD	Value on all other RESETS
01h	TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF	0000 0000	0000 000u
81h	OPTION_REG	$\overline{\text{GPPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISIO	—	—	TRISIO5	TRISIO4	TRISIO3	TRISIO2	TRISIO1	TRISIO0	--11 1111	--11 1111

Legend: — = Unimplemented locations, read as '0', u = unchanged, x = unknown.

Shaded cells are not used by the Timer0 module.

При настройке порта ввода вывода попутно был настроен и таймер. 😊

Если предделитель 1:8, то при тактовой частоте 4 МГц частота прерываний таймера будет равна:

$f_{пр} = ((4000000/4)/256)/8 = 488,3$ Гц. Кривое 😞

Для формирования сигнала частотой 300 Гц нужна частота 600 Гц.

К таймеру можно прибавлять значение во время его работы.

$f_{пр} = (1000000/X)/8 = 600$ Гц. $X = 208$, после каждого прерывания к TMR0 надо прибавлять $256 - 208 = 44$, чтобы частота была равна 600 Гц. 😊

БИТЫ конфигурации

The screenshot displays the MPLAB IDE interface. The 'PIC Memory Views' menu is open, showing options like Program Memory, File Registers, SFRs, Configuration Bits, EE Data Memory, Hardware Stack, and User ID Memory. The 'Configuration Bits' window is also open, showing the configuration statements for the PIC12F675 device.

Configuration Statements:

```
// 'C' source line config statements
// CONFIG
#pragma config FOSC = INTRCCLK // Oscillator Selection bits (INTOSC)
#pragma config WDTE = ON // Watchdog Timer Enable bit (WDTE)
#pragma config PWRTE = OFF // Power-Up Timer Enable bit (PWRT)
#pragma config MCLRE = ON // GP3/MCLR pin function select (GP3)
#pragma config BOREN = ON // Brown-out Detect Enable bit (BOREN)
#pragma config CP = OFF // Code Protection bit (Program Memory)
#pragma config CPD = OFF // Data Code Protection bit (Data Memory)

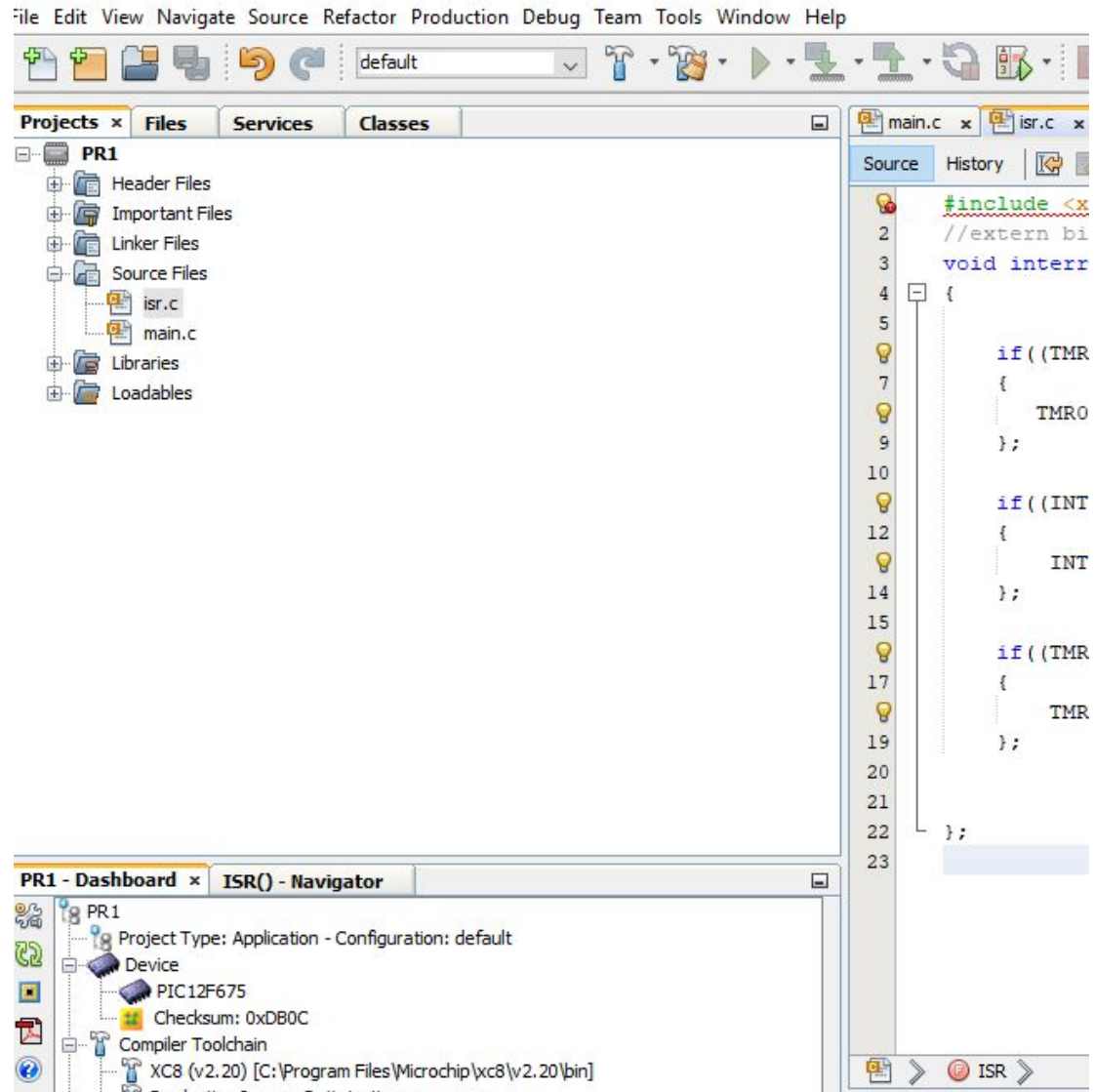
// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

#include <xc.h>
```

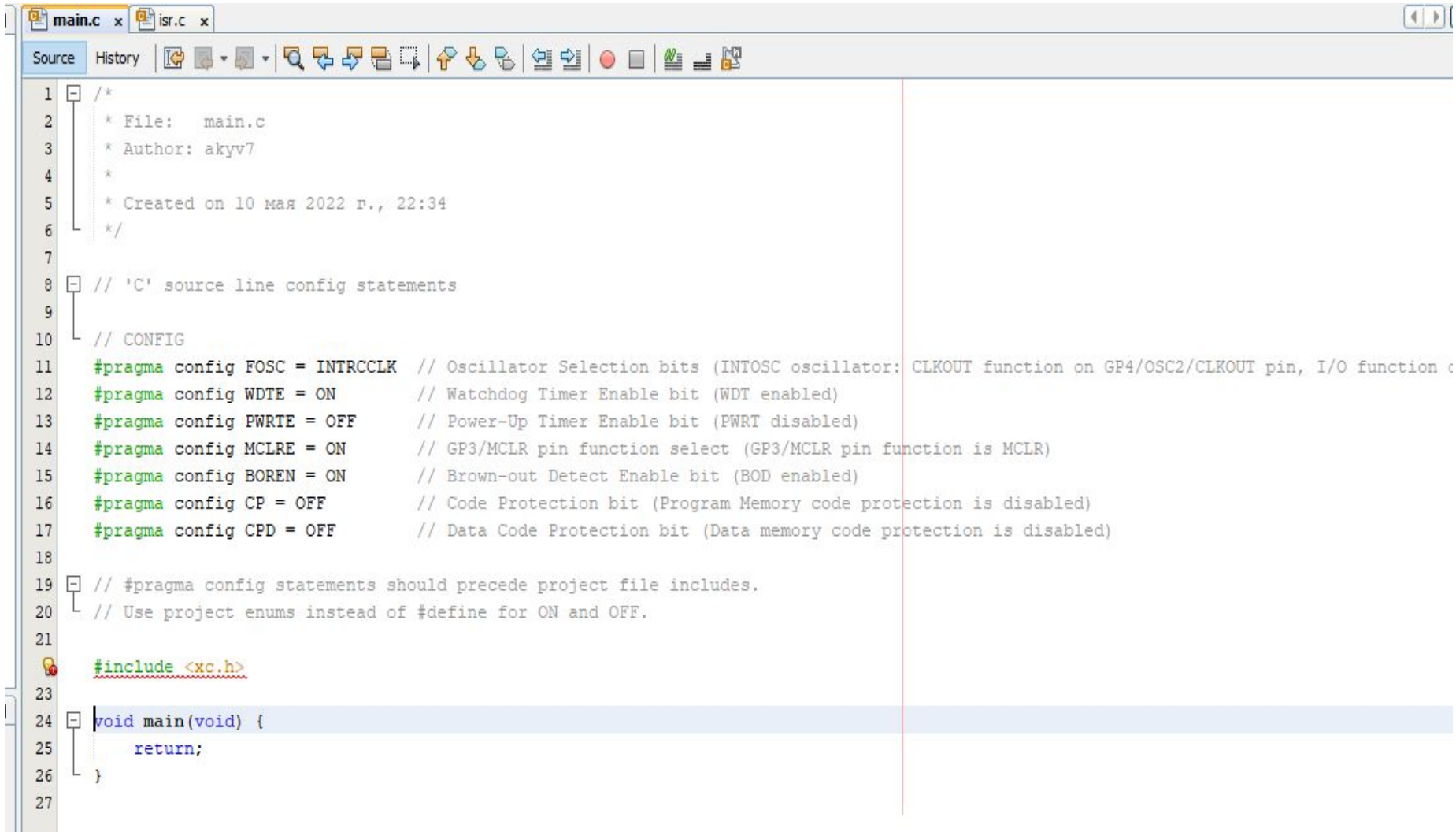
Обработчик прерываний

Создать файл ISR.C с текстом:

```
#include <xc.h>
void interrupt ISR(void)
{
    if((TMR0IF)&&(TMR0IE))
    {
// Здесь обработка прерывания
TMR0IF=0;
    };
    if((INTF)&&(INTE))
    {
// Здесь обработка прерывания
INTF=0;
    };
    if((TMR1IF)&&(TMR1IE))
    {
// Здесь обработка прерывания
TMR1IF=0;
    };
};
```



Заготовка main.c



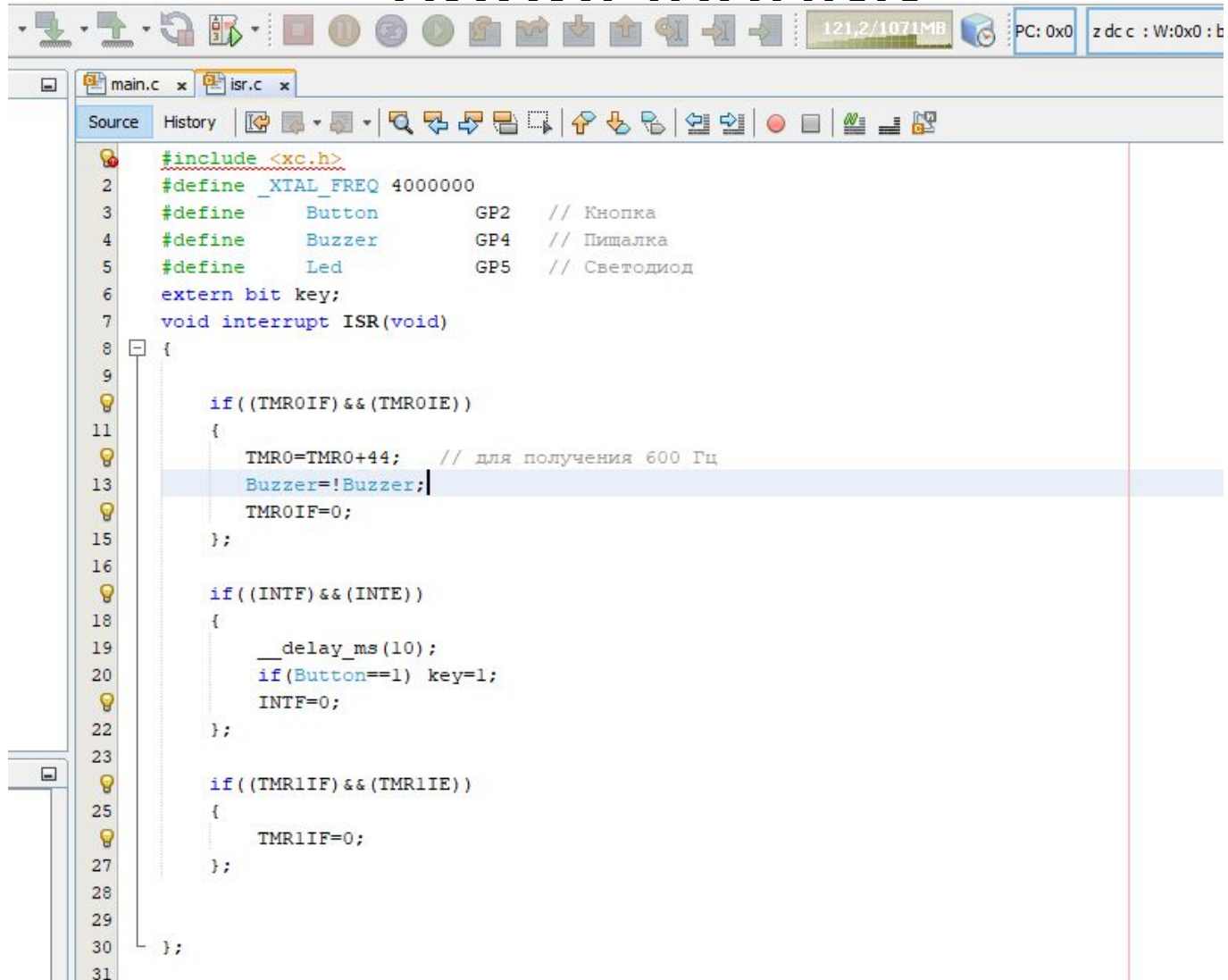
```
1 /*
2  * File:   main.c
3  * Author: akvv7
4  *
5  * Created on 10 мая 2022 г., 22:34
6  */
7
8 // 'C' source line config statements
9
10 // CONFIG
11 #pragma config FOSC = INTRCCLK // Oscillator Selection bits (INTOSC oscillator: CLKOUT function on GP4/OSC2/CLKOUT pin, I/O function on GP4/CLKOUT)
12 #pragma config WDTE = ON      // Watchdog Timer Enable bit (WDT enabled)
13 #pragma config PWRT = OFF     // Power-Up Timer Enable bit (PWRT disabled)
14 #pragma config MCLRE = ON     // GP3/MCLR pin function select (GP3/MCLR pin function is MCLR)
15 #pragma config BOREN = ON     // Brown-out Detect Enable bit (BOD enabled)
16 #pragma config CP = OFF      // Code Protection bit (Program Memory code protection is disabled)
17 #pragma config CPD = OFF     // Data Code Protection bit (Data memory code protection is disabled)
18
19 // #pragma config statements should precede project file includes.
20 // Use project enums instead of #define for ON and OFF.
21
22 #include <xc.h>
23
24 void main(void) {
25     return;
26 }
27
```

Размещение переменных и конфигурация периферии

```
16 #pragma config CP = OFF          // Code Protection bit (Program Memory code protection is disabled)
17 #pragma config CPD = OFF         // Data Code Protection bit (Data memory code protection is disabled)
18
19 // #pragma config statements should precede project file includes.
20 // Use project enums instead of #define for ON and OFF.
21
22 #include <xc.h>
23
24 #define _XTAL_FREQ 4000000       // Частота тактового генератора в герцах
25                                 // Для работы встроенной функции по формированию задержек
26 /* Например
27  * __delay_ms(100);             //сформирует задержку 100 миллисекунд
28  * __delay_us(50);             //сформирует задержку 500 микросекунд
29  */
30 #define Button GP2              // Кнопка
31 #define Buzzer GP4              // Пищалка
32 #define Led GP5                 // Светодиод
33
34 //unsigned char counter=0;
35 bit key=0;
36 void main(void) {
37     // Настройка периферии
38     OPTION_REG=0b11000011;      //Прерывания
39     INTCON=0b10110000;          // Прерывания
40     CMCON=0b00000011;          // Коопаратор
41     ANSEL=0b00000000;          // АЦП
42     TRISIO=0b00000111;         // входы и выходы
43     WPU=0;                      // Подтягивающие резисторы
44     IOC=0;                       // Прерывание по изменению состояния порта
45     GPIO=0b00100000;           // Начальная установка состояния порта
46
47 }
```

Доработка обработчика

прерываний



```
1  #include <xc.h>
2  #define _XTAL_FREQ 4000000
3  #define Button GP2 // Кнопка
4  #define Buzzer GP4 // Пищалка
5  #define Led GP5 // Светодиод
6  extern bit key;
7  void interrupt ISR(void)
8  {
9
10     if((TMROIF) && (TMROIE))
11     {
12         TMRO=TMRO+44; // для получения 600 Гц
13         Buzzer=!Buzzer;
14         TMROIF=0;
15     };
16
17     if((INTF) && (INTE))
18     {
19         __delay_ms(10);
20         if(Button==1) key=1;
21         INTF=0;
22     };
23
24     if((TMR1IF) && (TMR1IE))
25     {
26         TMR1IF=0;
27     };
28
29
30 };
31
```

Изменение main.c

```
35 bit key=0;
36 void main(void) {
37     // Настройка периферии
38     OPTION_REG=0b11000011; //Прерывания
39     INTCON=0b10110000; // Прерывания
40     CMCON=0b00000111; // Коипаратор
41     ANSEL=0b00000000; // АЦП
42     TRISIO=0b00001111; // входы и выходы
43     WPU=0; // Подтягивающие резисторы
44     IOC=0; // Прерывание по изменению состояния порта
45     GPIO=0b00100000; // Начальная установка состояния порта
46     IOIE=0; // Запрет прерываний от TMR0
47     do
48     {
49         while(key==0) NOP();
50         Led=0; // потасили
51         TOIE=1; //Пищим
52         key=0; // сброс флага кнопки
53         while(key==0) NOP();
54         Led=1; // включил
55         TOIE=1; //не Пищим
56         key=0; // сброс флага кнопки
57     }while(1);
58
59
60     return;
61 }
```

Сколько это занимает места?

The screenshot displays the MPLAB IDE interface. On the left, the 'PR1 - Dashboard' shows project configuration for a PIC12F675. The 'Memory' section indicates that 11% of the 64 (0x40) bytes of Data memory and 9% of the 1024 (0x400) words of Program memory are used. The 'Output' window on the right shows the execution of a C program with the following code:

```
47 do
48 {
50     while(key==0)
51         Led=0;
52     TOIE=1;
53     key=0;
54     while(key==0)
55         Led=1;
56     TOIE=1;
57     key=0;
58 }while(1);
59
60 return;
61 }
62
```

The 'Output' window also shows the following text:

```
Config Bits Source x PR1 (Clean, B
BUILD SUCCESSFUL (total time
Loading code from C:/Users/a
Loading completed
```


Симуляция проекта, железа нет

The screenshot shows the MPLAB IDE interface. The 'Window' menu is open, and the 'Simulator' option is selected. A sub-menu is visible with the following options:

- Stimulus
- Logic Analyzer
- IOPin
- RegisterTrace

The code editor displays the following assembly code:

```
// Для работы встроенной функции
// сформирует задержку 100 миллисекунд
// сформирует задержку 500 микросекунд
GP2 // Кнопка
GP4 // Пиналка
GP5 // Светодиод

counter=0;

{
ка периферии
=0b11000011; //Прерывания
// Прерывания
// Коопаратор
// АЦП
// входы и выходы
// Подтягивающие резисторы
// Прерывание по изменению
// Начальная установка состо
// Запрет прерываний от TMP
key==0) NOP ();
// погасили
//Пишим
TOIE=1;
key=0; // сброс флага кнопки
while (key==0) NOP ();
Led=1; // включил
TOIE=1; //не Пишим
```

The I/O Pins window is visible at the bottom, showing the pin configuration for the device.

```
CMCON=0b00000111; // КоиPARATOR
ANSEL=0b00000000; // АЦП
TRISIO=0b00001111; // входы и выходы
WPU=0; // Подтягивающие резисторы
IOC=0; // Прерывание по изменению состояния порта
GPIO=0b00100000; // Начальная установка состояния порта
TOIE=0; // Запрет прерываний от TMP0
47 do
48 {
    while (key==0) NOP ();
    Led=0; // погасили
    TOIE=1; //Пишим
    key=0; // сброс флага кнопки
    while (key==0) NOP ();
    Led=1; // включил
    TOIE=1; //не Пишим
```

< main > do >

Variables	I/O Pins x	Output	IO View	SFRs	Configuration Bits	Configuration Bits
Pin	/	Mode				Value
I						
GP1						
GP2						
GP3						
GP4						
GP5						
ICSPCLK						
ICSPDAT						
INT						
MCLR						
OSC1						

The screenshot shows a software development environment with a menu open over a code editor. The menu includes options like 'Xplained', 'Projects', 'Files', 'Classes', 'Favorites', 'Services', 'Dashboard', 'Navigator', 'Action Items', 'Tasks', 'Output', 'Editor', 'Debugging', 'Web', 'IDE Tools', 'PIC Memory Views', 'Simulator', 'Configure Window', 'Reset Windows', 'Close Window', 'Close All Documents', 'Close Other Documents', 'Document Groups', and 'Documents...'. The 'Simulator' sub-menu is open, showing 'Stimulus', 'Logic Analyzer', 'IOPin', and 'RegisterTrace'. The code editor displays assembly-like code with comments in Russian. The status bar at the bottom shows 'main > do >' and several tool windows: 'Variables', 'I/O Pins', 'Output', 'IO View', 'Call Stack', 'Breakpoints', 'SFRs', and 'Configuration'. The 'Output' window is active, showing the text: 'Launching', 'Initializing simulator', 'User program running', and 'User program stopped'.

```
// Для работы встроенной функции по
100); //сформирует задержку 100 миллисекунд
(50); //сформирует задержку 500 микросекунд
cton GP2 // Кнопка
zzer GP4 // Пышалка
1 GP5 // Светодиод
counter=0;
>
> {
> ка периферии
> =0b11000011; //Прерывания
> // Прерывания
> // Кооператор
> // АЦП
> // входы и выходы
> // Подтягивающие резисторы
> // Прерывание по изменению состояния
> // Начальная установка состояния пор
> // Запрет прерываний от TMR0
key==0) NOP ();
led=0; // погасили
TOIE=1; //Пишим
key=0; // сброс флага кнопки
52 while (key==0) NOP ();
54 led=1; // включил
TOIE=1; //не Пишим
```

main > do >

Variables I/O Pins Output x IO View Call Stack Breakpoints SFRs Configuration

Config Bits Source x PR1 (Build, Load, ...) x Debugger Console x Simulator x

Launching
Initializing simulator
User program running
User program stopped

```

Source History
// компаратор
ANSEL = 0b00000111; // АЦП
TRISIO = 0b00001111; // входы и выходы
WPU = 0; // Подтягивающие резисторы
IOC = 0; // Прерывание по изменению состояния порта
GPIO = 0b00100000; // Начальная установка состояния порта
TOIE = 0; // Запрет прерываний от TMP0
47 do
48 {
49     while(key==0) NOP();
50     Led=0; // погасили
51     TOIE=1; //Пишим
52     key=0; // сброс флага кнопки
53     while(key==0) NOP();
54     Led=1; // включил
55     TOIE=1; //не Пишим
56     key=0; // сброс флага кнопки
57 }while(1);
58

```

Stimulus x

Asynchronous Pin/Register Actions Advanced Pin/Register Clock Stimulus Register Injection

Fire	Pin	Action	Value	Units	Comments
	GP2	Toggle			Optional.com

Asynchronous Stimulus Toggle GP2 fired.

Variables	I/O Pins x	Output	IO View	Call Stack	Breakpoints	SFRs	Configuration Bits	Configuration Bits
Pin			/	Mode			Value	C
GP5				Dout			1	GI

Задания

Задание 1.

При нажатии кнопки генерируется звуковой сигнал азбукой Морзе «73».

Задание 2.

При включении питания звуковой сигнал не генерируется. При нажатии на кнопку генерируется сигнал частотой 100 Гц. При повторном нажатии на кнопку генерация звукового сигнала прекращается. При следующем нажатии на кнопку генерируется сигнал частотой 100 Гц.

Задание 3.

Нажатие на кнопку сопровождается звуковым сигналом, обозначающим факт нажатия кнопки и включается либо выключается светодиод.

Задание 4.

При включении питания звуковой сигнал не генерируется.

При нажатии на кнопку SB2 генерируется сигнал частотой 100 Гц.

При нажатии на кнопку SB3 генерируется сигнал частотой 500 Гц.

При нажатии на кнопку SB4 генерируется сигнал частотой 1000 Гц.

Задание 5.

При включении питания генерируется звуковой сигнал в зависимости от угла поворота переменного резистора R8.

При крайнем левом положении частота 50 Гц, при крайнем правом положении частота 500 Гц, частота меняется плавно при повороте вала резистора R8.

Задание 6.

При включении питания генерируется звуковой сигнал:

Две секунды генерируется сигнал частотой 100 Гц, далее две секунды генерируется сигнал частотой 1000 Гц, далее две секунды генерируется сигнал частотой 2000 Гц.

Повторить все с начала.

Задание 7.

При нажатии на кнопку включить все светодиоды, при повторном нажатии на кнопку светодиоды выключить.

Задание 8.

Используя три светодиода реализовать светофор. При нажатии на кнопку перейти в режим мигающего желтого, при повторном нажатии перейти к нормальной работе светофора.

Задание 9.

При включении питания светодиод выключен. При нажатии на кнопку светодиод мигает с частотой 5 Гц. При повторном нажатии на кнопку светодиод выключается. При следующем нажатии на кнопку светодиод мигает с частотой 5 Гц.

Задание 10.

При включении питания светодиод выключен. При нажатии на кнопку светодиод мигает с частотой 5 Гц. При повторном нажатии на кнопку светодиод выключается. При следующем нажатии на кнопку светодиод мигает с частотой 5 Гц.

Задание 11.

При включении питания светодиод плавно включается в течении 1..2 сек. При нажатии на кнопку светодиод плавно включается в течении 1..2 сек.

Задание 12.

При нажатии кнопки генерируется световой сигнал азбукой Морзе «SOS».

Задание 13.

При включении питания световой сигнал не генерируется.

При нажатии на кнопку SB4 светодиод мигает с частотой 1 Гц.

При нажатии на кнопку SB5 светодиод мигает с частотой 5 Гц.

При нажатии на кнопку SB6 светодиод мигает с частотой 10 Гц.

Задание 14.

При включении питания яркость свечения светодиода зависит от угла поворота переменного резистора R8.

При крайнем левом положении светодиод не светится, при крайнем правом положении светодиод светится с максимальной яркостью, яркость свечения меняется плавно при повороте вала резистора R8.

Задание 15.

При включении питания четыре светодиода отображают угол поворота переменного резистора R8 в диапазоне от 0 до 15.

При крайнем левом положении вала резистора R8 отображается 0, при крайнем правом положении вала резистора R8 отображается 15.

Задание 16.

При нажатии кнопки происходит счет количества нажатий от 0 до 15, при переполнении счетчика нажатий кнопки счет начинается сначала. Вывести состояние счетчика на светодиоды. Каждое нажатие кнопки сопровождается звуковым сигналом.

Задание 17.

При первом включении питания макета светодиоды выключены. При нажатии на первую происходит счет количества нажатий от 0 до 3, при переполнении счетчика нажатий кнопки счет начинается сначала. Вывести состояние счетчика на светодиоды. При нажатии на вторую кнопку происходит запись содержимого счетчика нажатий в ЭСППЗУ. При включении питания данные из ЭСППЗУ считываются и данные отображаются светодиодами.

Задание 18.

Автомат световых эффектов «бегущий свет», скорость переключения регулируется резистором R8.

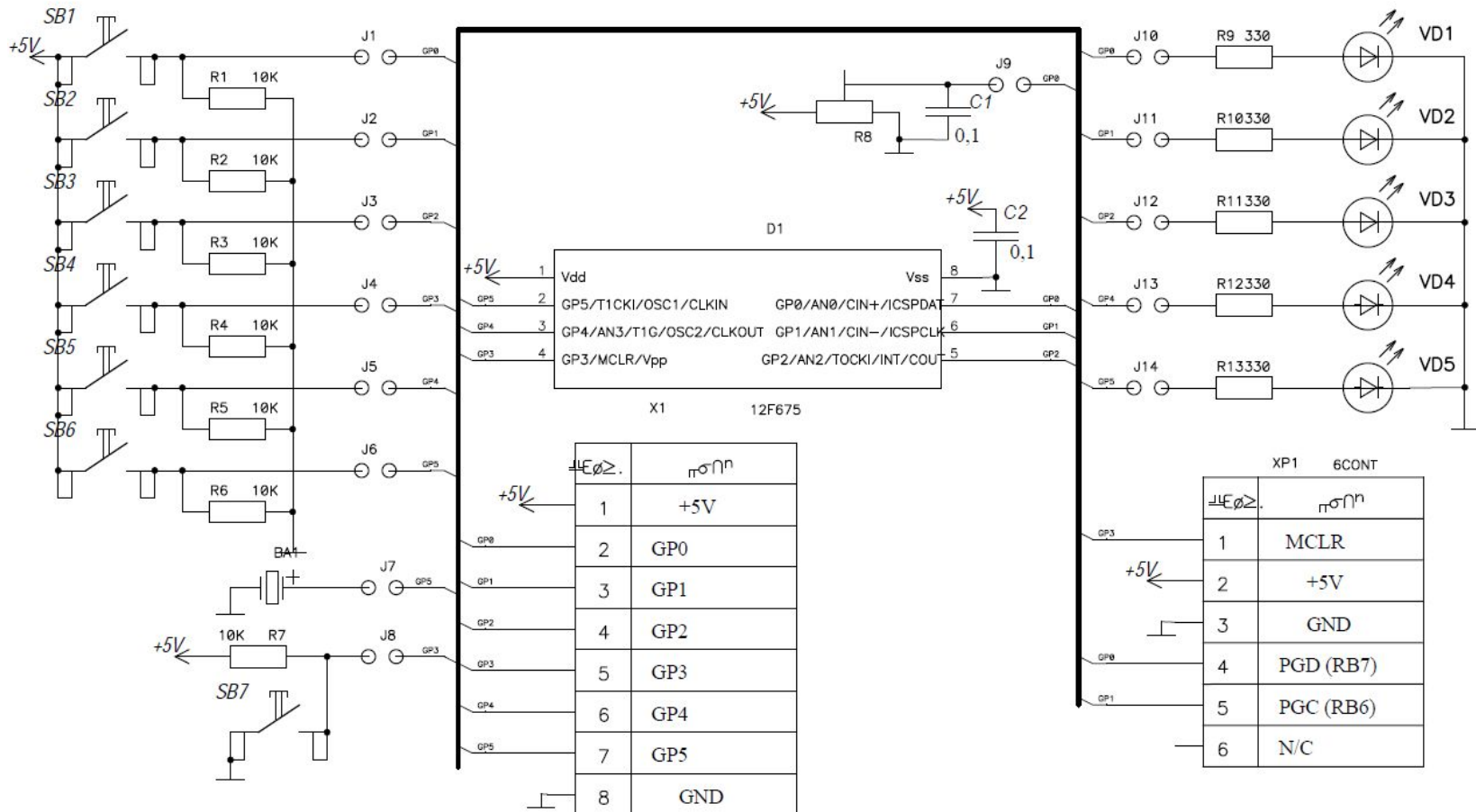
Задание 19.

Автомат световых эффектов «бегущая тень», скорость переключения регулируется кнопкой.

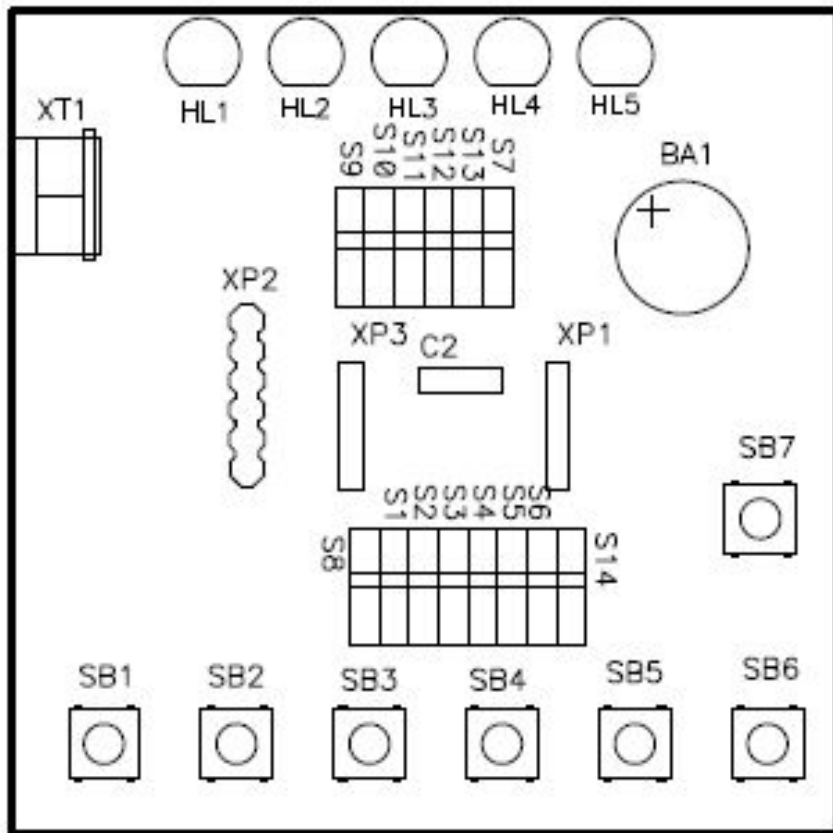
Задание 20.

Генератор случайных чисел в диапазоне от 2 до 12 (сумма чисел, которые могут выпасть при бросании двух кубиков). Случайное число генерировать при нажатии кнопки, результат отобразить светодиодами.

Схема макета



Плата макета



- HL1...HL5 светодиоды
- S1...S14 переключатели выбора элементов.
- SB1...SB6 кнопки.
- SB7 кнопка Reset.
- BA1 пьезоизлучатель.
- XP2 разъем для подключения программатора.
- XT1 разъем подключения внешнего питания.
- XP3, XP1 разъемы, подключенные параллельно выводам микроконтроллера.