

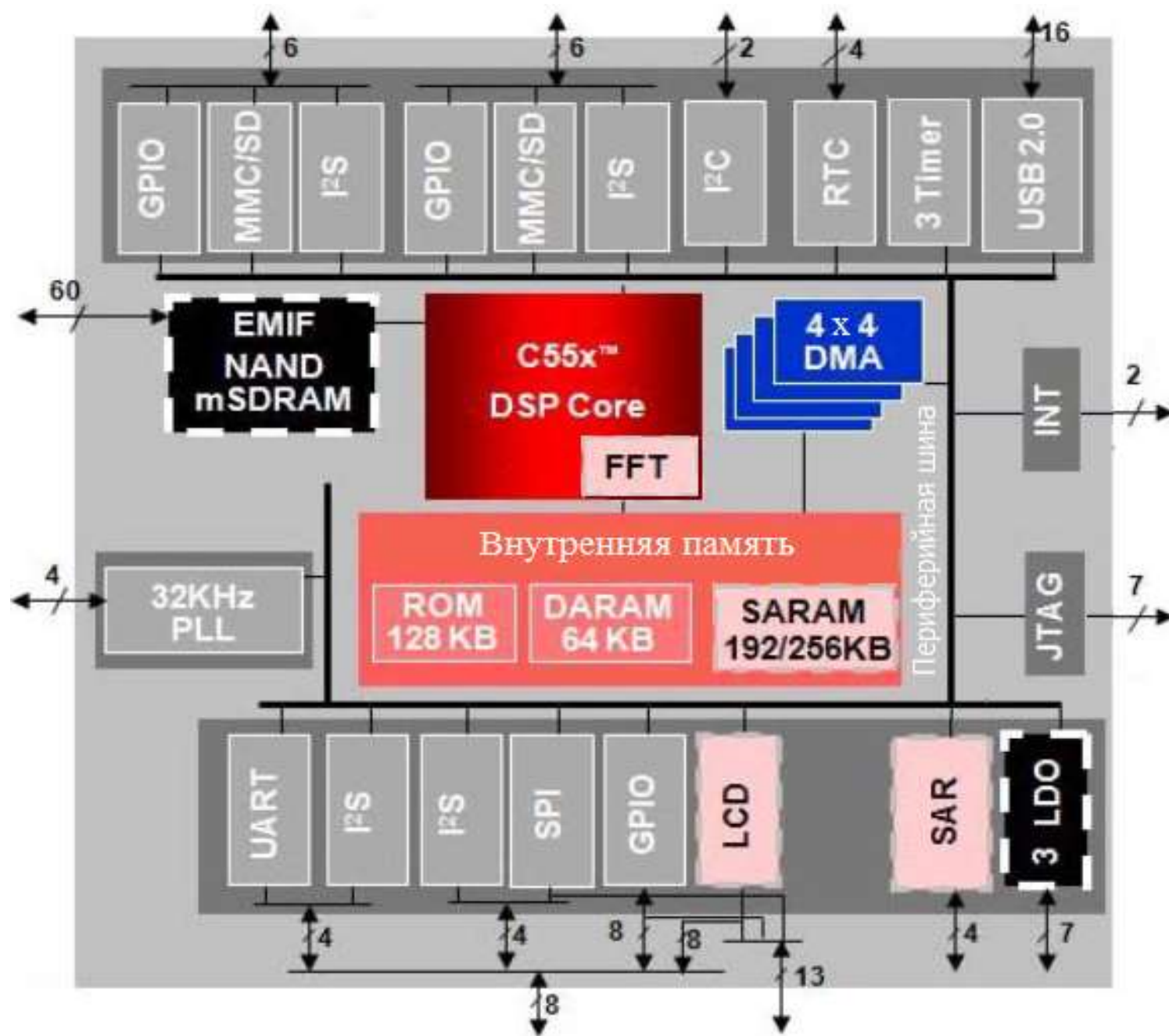


Микропроцессорные устройства обработки сигналов

Лекция L12
«Входы-выходы общего назначения»

<http://vykhovanets.ru/course67/>

Состав микропроцессора

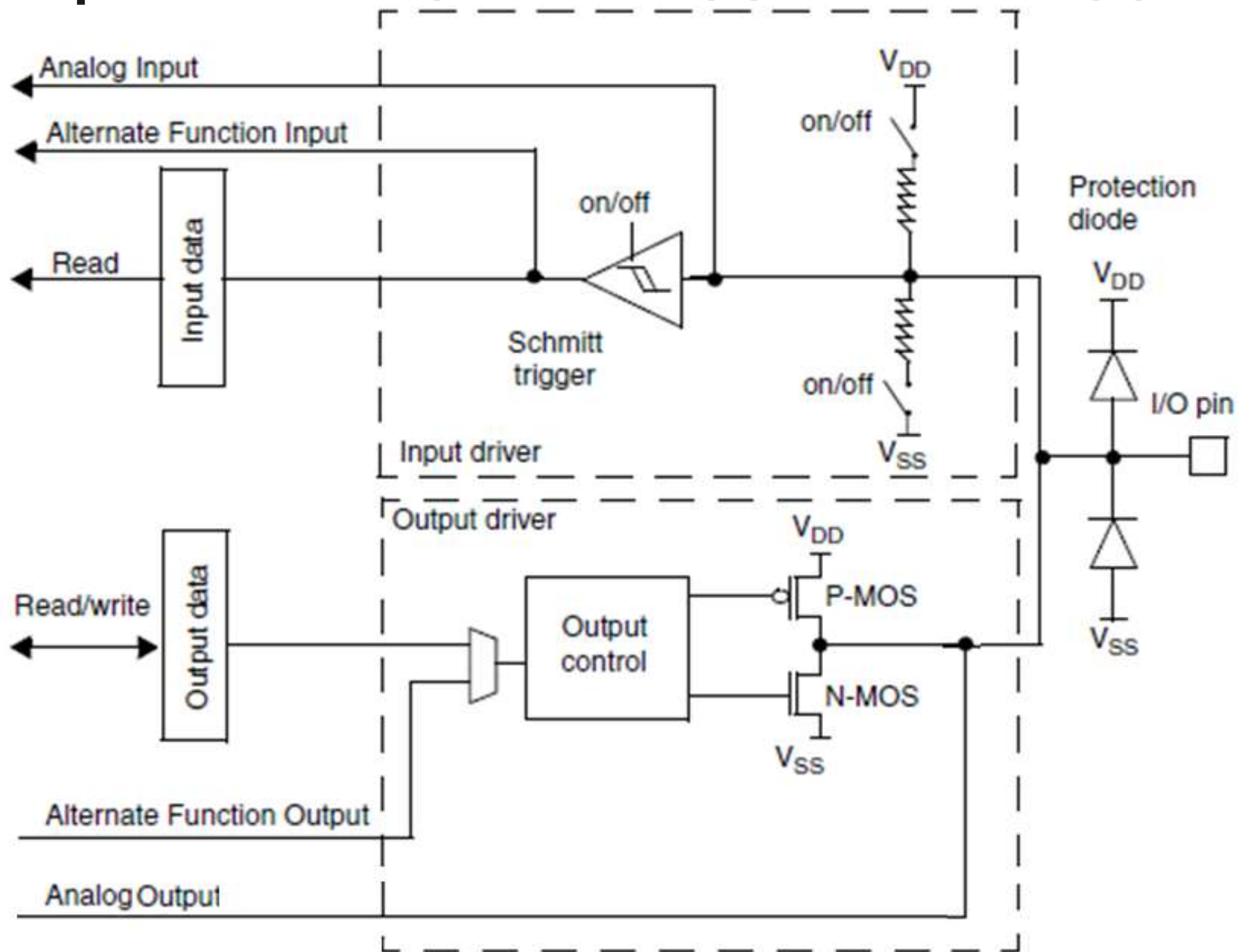


Альтернативные функции

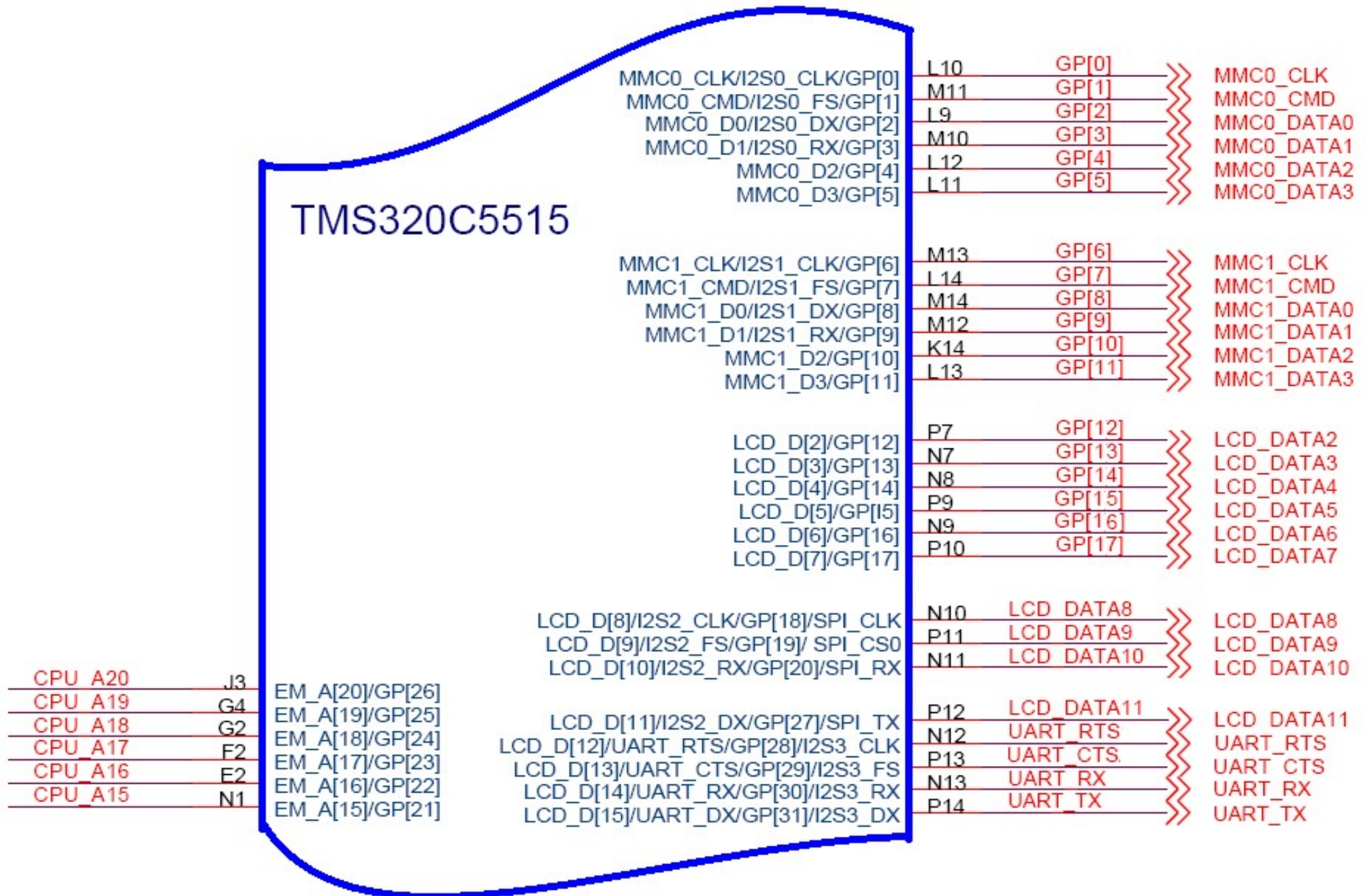
Имя	Вывод	Альтернативная функция
GP[0]	L10	MMC0_CLK, I2S0_CLK
GP[1]	M11	MMC0_CMD, I2S0_FS
GP[2]	L9	MMC0_D0, I2S0_DX
GP[3]	M10	MMC0_D1, I2S0_RX
GP[4]	L12	MMC0_D2
GP[5]	L11	MMC0_D3
GP[6]	M13	MMC1_CLK, I2S1_CLK
GP[7]	L14	MMC1_CMD, I2S1_FS
GP[8]	M14	MMC1_D0, I2S1_DX
GP[9]	M12	MMC1_D1, I2S1_RX
GP[10]	K14	MMC1_D2
GP[11]	L13	MMC1_D3
GP[12]	P7	LCD_D[2]
GP[13]	N7	LCD_D[3]
GP[14]	N8	LCD_D[4]
GP[15]	P9	LCD_D[5]

Имя	Вывод	Альтернативная функция
GP[16]	N9	LCD_D[6]
GP[17]	P10	LCD_D[7]
GP[18]	N10	LCD_D[8], I2S2_CLK, SPI_CLK
GP[19]	P11	LCD_D[9], I2S2_FS, SPI_CS0
GP[20]	N11	LCD_D[10], I2S2_RX, SPI_RX
GP[21]	N1	EM_A[15]
GP[22]	E2	EM_A[16]
GP[23]	G1	EM_A[17]
GP[24]	G2	EM_A[18]
GP[25]	G4	EM_A[19]
GP[26]	J3	EM_A[20]
GP[27]	P12	LCD_D[11], I2S2_DX, SPI_TX
GP[28]	N12	LCD_D[12], I2S3_CLK, UART_RTS
GP[29]	P13	LCD_D[13], I2S3_FS, UART_RTS
GP[30]	N13	LCD_D[14], I2S3_RX, UART_RXD
GP[31]	P14	LCD_D[15], I2S3_DX, UART_TXD

Организация входов-выходов



Использование входов-выходов



Регистр выбора шин

EBSR (1C00h) – External Bus Selection Register (регистр выбора внешних шин)

15	14	12	11	10	9	8	
Reserved	PPMODE			SP1MODE		SP0MODE	
7	6	5	4	3	2	1	0
Reserved	Reserved	A20_MODE	A19_MODE	A18_MODE	A17_MODE	A16_MODE	A15_MODE

- **PPMODE** – **Parallel Port Mode** (режим параллельного порта, 21 вывод:
 - 000 – LCD[21 вывод];
 - 001 – SPI[7 выводов], GPIO[29:27, 20:18], UART[4 вывода] и I2S2 [4 вывода];
 - 010 – LCD[13 выводов] и GPIO[31:27, 20:18];
 - 011 – LCD[13 выводов], SPI [4 вывода] и I2S3 [4 вывода];
 - 100 – LCD[13 выводов], I2S2 [4 вывода] и UART [4 вывода];
 - 101 – LCD[13 выводов], SPI[4 вывода] и UART[4 вывода];
 - 110 – SPI[7 выводов], I2S2[4 вывода], I2S3[4 вывода] и GPIO[29:27, 20:18];
 - 111 – зарезервировано)
- **SP1MODE** – **Serial Port 1 Mode** (режим последовательного порта 1, 6 выводов:
 - 00 – MMC/SD1[6 выводов]; 01 – I2S1[4 вывода] and GPIO[11:10];
 - 10 – GPIO[11:6]; 11 – зарезервировано)
- **SP0MODE** – **Serial Port 0 Mode** (режим последовательного порта 1, 6 выводов:
 - 00 – MMC/SD0[6 выводов]; 01 – I2S0[4 вывода], GPIO[5:4];
 - 10 – GPIO[5:0]; 11 – зарезервировано)
- **Ax_MODE** – **Ax Mode** (режим адресных выводов A[20:15]: 0 – интерфейс расширенной памяти, 1 – входы-выходы общего назначения GPIO[26:21])

Подтягивающие резисторы

PDINHIBR1 (1C17h) – Pull-Down Inhibit Register 1

15	14	13	GP[11] 12	GP[10] 11	GP[9] 10	GP[8] 9	GP[7] 8
Reserved	S15PD	S14PD	S13PD	S12PD	S11PD	S10PD	
7	6	GP[5] 5	GP[4] 4	GP[3] 3	GP[2] 2	GP[1] 1	GP[0] 0
Reserved	S05PD	S04PD	S03PD	S02PD	S01PD	S00PD	

PDINHIBR2 (1C18h) – Pull-Down Inhibit Register 2

15	14	13	12	11	10	9	8
Reserved	INT1PU	INT0PU	RESETPU	EMU01PU	TDIPU	TMSPU	TCKPU
7	6	GP[26] 5	GP[25] 4	GP[24] 3	GP[23] 2	GP[22] 1	GP[21] 0
Reserved	A20PD	A19PD	A18PD	A17PD	A16PD	A15PD	

PDINHIBR3 (1C19h) – Pull-Down Inhibit Register 3

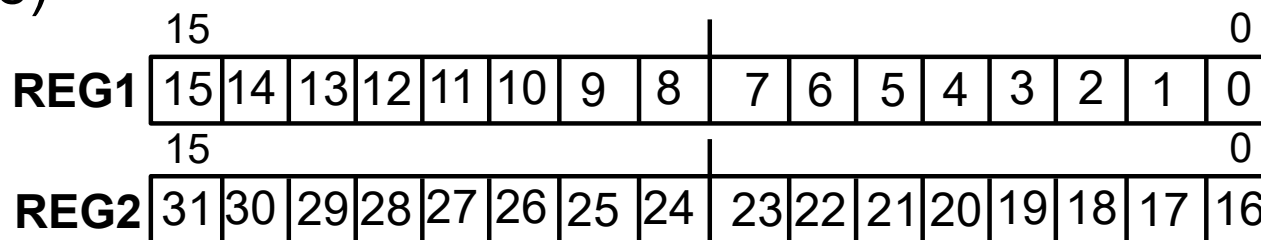
GP[31] 15	GP[30] 14	GP[29] 13	GP[28] 12	GP[27] 11	GP[20] 10	GP[19] 9	GP[18] 8
PD15PD	PD14PD	PD13PD	PD12PD	PD11PD	PD10PD	PD9PD	PD8PD
GP[17] 7	GP[16] 6	GP[15] 5	GP[14] 4	GP[13] 3	GP[12] 2	1	0
PD7PD	PD6PD	PD5PD	PD4PD	PD3PD	PD2PD	Reserved	

- **PU** – **P**ull-**U**p (подтягивание к питанию)
- **PD** – **P**ull-**D**own (подтягивание к земле)
- **S** – **S**erial port (выводы последовательных портов)
- **INTx** – **I**nterrupt x (выводы прерывания микропроцессора)
- **RESET** – **R**eset (вывод сброса микропроцессора)
- **EMU01** – **E**mulator (выводы 0 и 1 прерывания эмулятора)
- **TDI** – **T**est **d**ata input (JTAG, вывод входных данных)
- **TMS** – **T**est **m**ode **s**elect (JTAG, вывод выбора режима)
- **TCK** – **T**est **c**lock (JTAG, вывод тактового сигнала)
- **A** – **A**ddress (адресные выводы EMIF)

Регистры входов-выходов

Адрес в памяти	Обозначение	Описание
1C06h	IODIR1	Регистр направления 1 (0 – ввод данных, 1 – вывод данных)
1C07h	IODIR2	Регистр направления 2 (0 – ввод данных, 1 – вывод данных)
1C08h	IOINDATA1	Регистр входных данных 1 (0 – вход “0”, 1 – вход “1”)
1C09h	IOINDATA2	Регистр входных данных 2 (0 – вход “0”, 1 – вход “1”)
1C0Ah	IODATAOUT1	Регистр выходных данных 1 (0 – выход “0”, 1 – выход “1”)
1C0Bh	IODATAOUT2	Регистр выходных данных 2 (0 – выход “0”, 1 – выход “1”)
1C0Ch	IOINTEDG1	Регистр полярности прерываний 1 (0 – передний фронт, 1 – задний)
1C0Dh	IOINTEDG2	Регистр полярности прерываний 2 (0 – передний фронт, 1 – задний)
1C0Eh	IOINTEN1	Регистр разрешения прерываний 1 (0 – запрещено, 1 – разрешено)
1C0Fh	IOINTEN2	Регистр разрешения прерываний 2 (0 – запрещено, 1 – разрешено)
1C10h	IOINTFLG1	Регистр флагов прерываний 1 (0 – нет события, 1 – событие есть)
1C11h	IOINTFLG2	Регистр флагов прерываний 2 (0 – нет события, 1 – событие есть)

- **IO** – **I**nput-**O**utput (входы-выходы)
- **DIR** – **D**irection (направление)
- **IN, OUT** – **I**nput, **O**utput (вход, выход)
- **INT** – **I**nterrupt (прерывание)
- **EDG** – **E**dge (фронт)
- **EN** – **E**nable (разрешение)
- **FLG** – **F**lag (флаг)

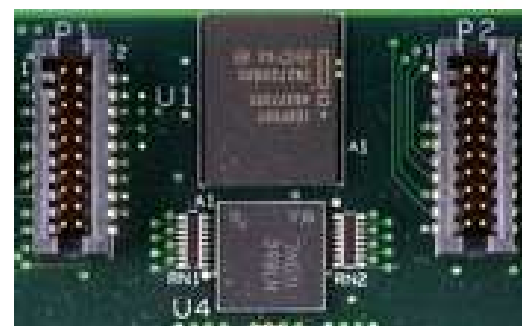
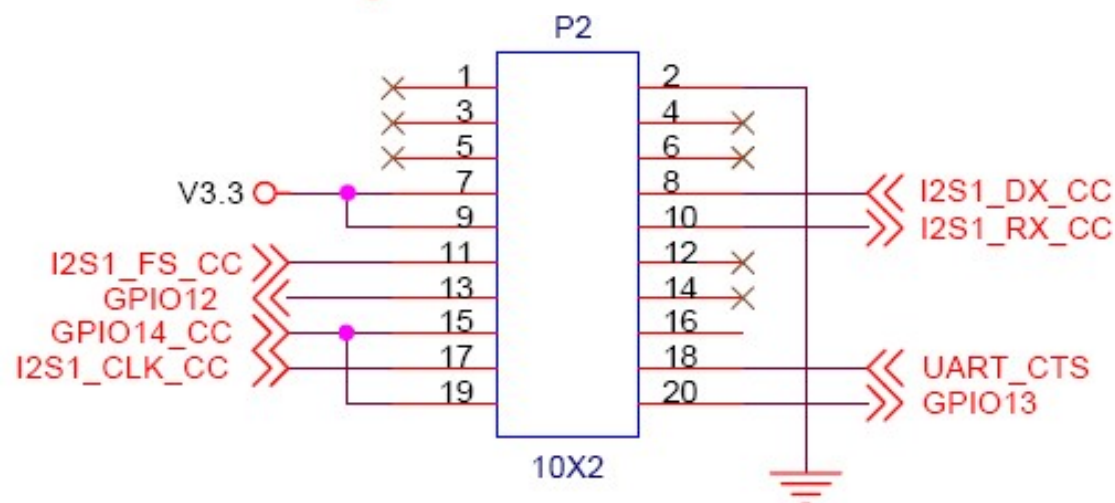
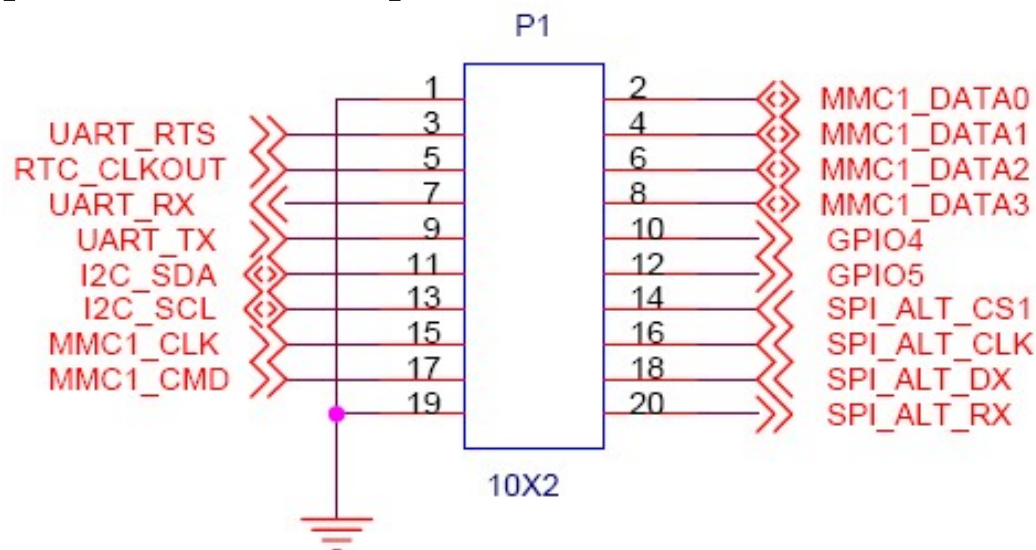
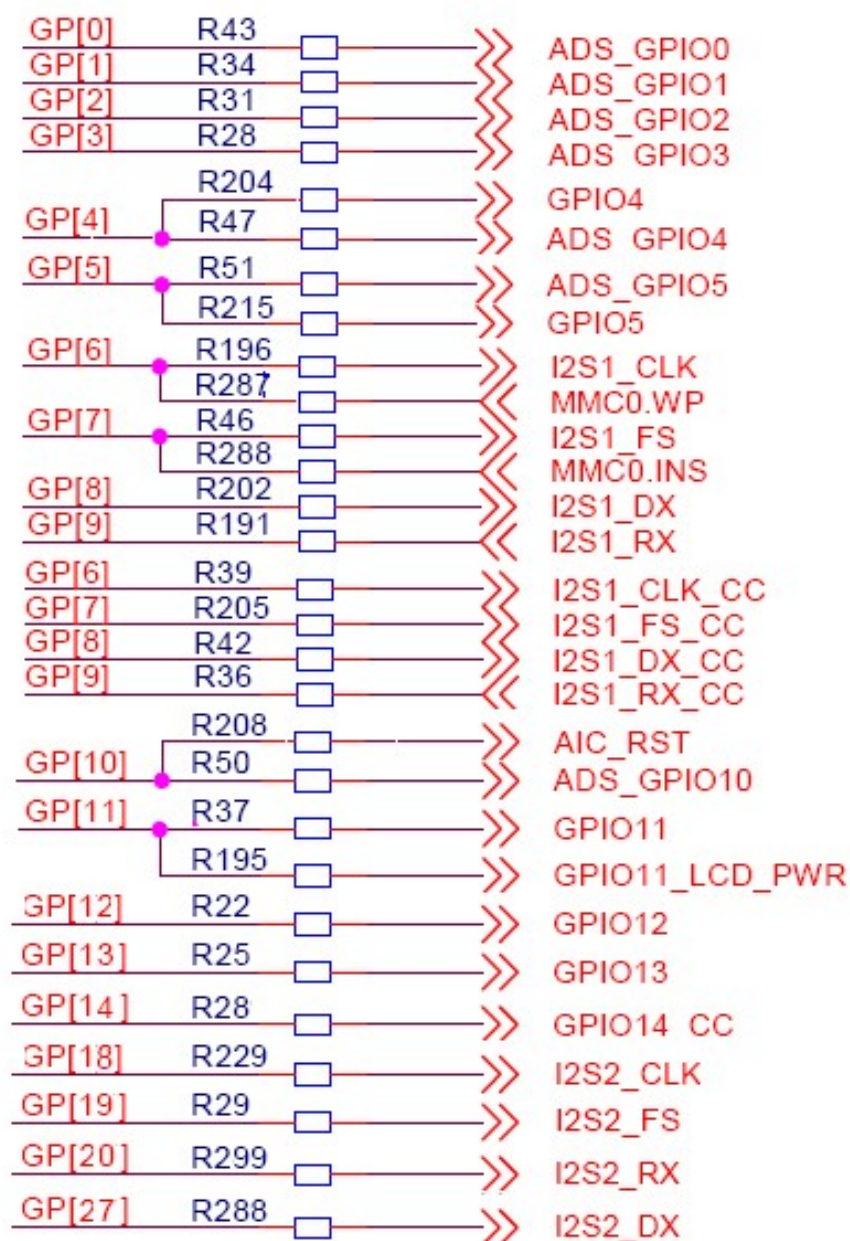


Вектор прерываний

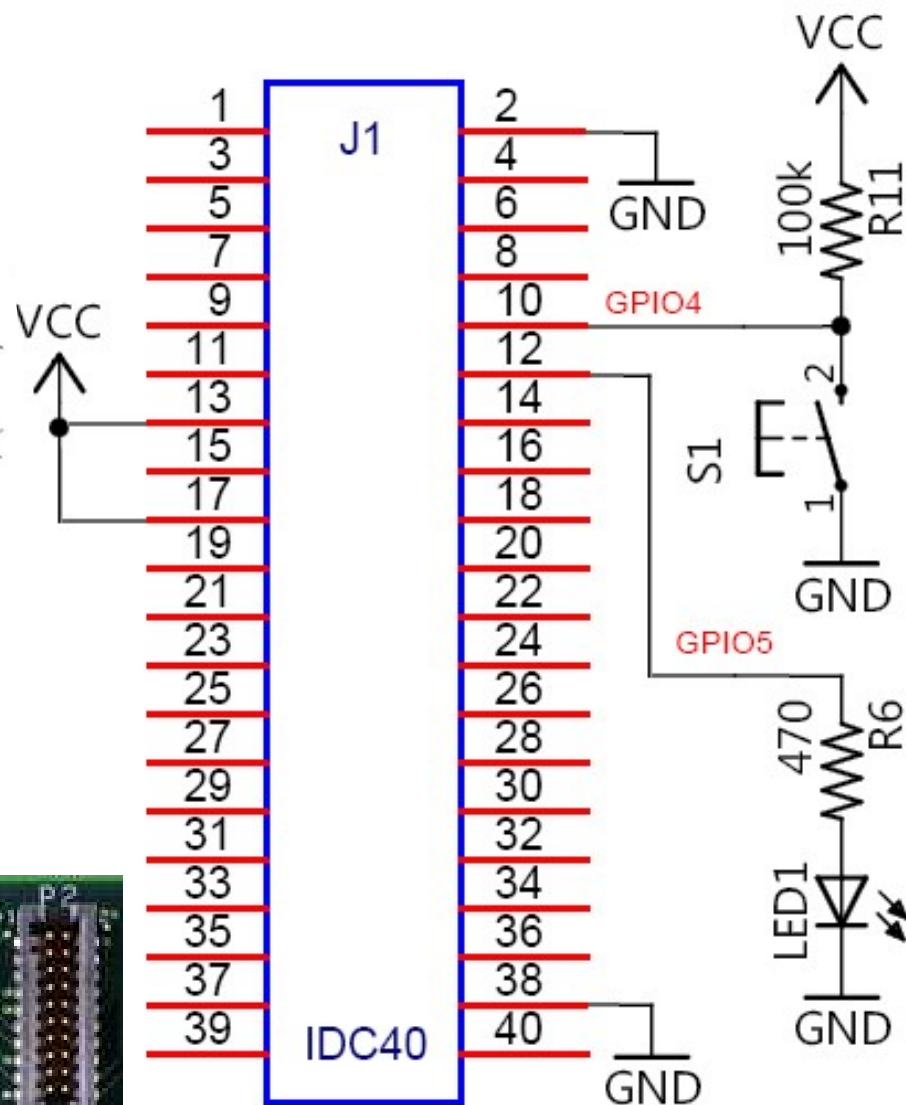
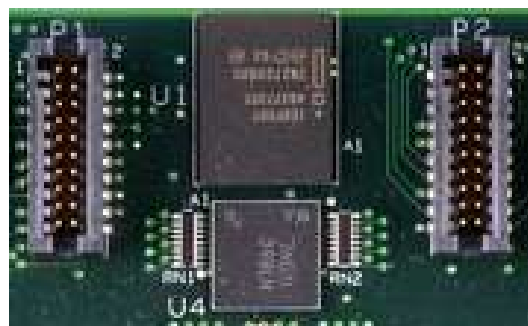
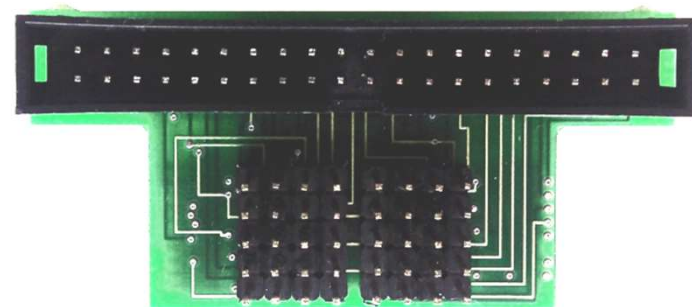
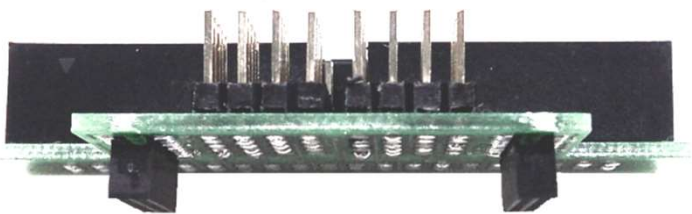
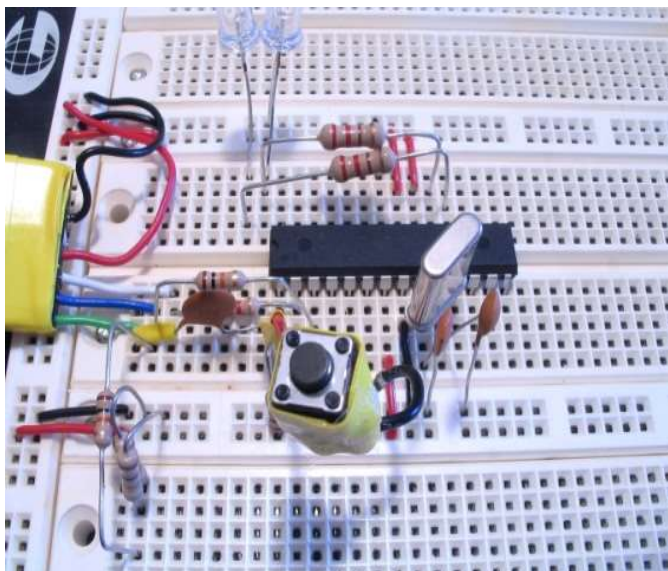
Вектор		Прерывание	Приоритет	Адрес
00	RESET	Сброса и инициализации	00	IVPD:00h
04	TINT	Агрегированное таймера	06	IVPD:20h
...
08	DMA	Прямого доступа к памяти	11	IVPD:40h
...
13	SAR	Агрегированное АЦП	18	IVPD:68h
...
18	RTC	Часов реального времени	12	IVPH:90h
...
21	GPIO	Портов ввода-вывода	20	IVPH:A8h

[illegible]

Внешние расширения



Экспериментальная схема



Заголовочный файл

/ Структура регистровой памяти входов-выходов общего назначения */*

```
typedef struct {  
    volatile unsigned int  IODIR1;           /* Регистр направлений 1, +0 */  
    volatile unsigned int  IODIR2;           /* Регистр направлений 2, +1 */  
    volatile unsigned int  IOINDATA1;        /* Регистр входных данных 1, +2 */  
    volatile unsigned int  IOINDATA2;        /* Регистр входных данных 2, +3 */  
    volatile unsigned int  IODATAOUT1;       /* Регистр выходных данных 1, +4 */  
    volatile unsigned int  IODATAOUT2;       /* Регистр выходных данных 2, +5 */  
    volatile unsigned int  IOINTEDG1;        /* Регистр полярности 1, +6 */  
    volatile unsigned int  IOINTEDG2;        /* Регистр полярности 2, +7 */  
    volatile unsigned int  IOINTEN1;         /* Регистр разрешений 1, +8 */  
    volatile unsigned int  IOINTEN2;         /* Регистр разрешений 2, +9 */  
    volatile unsigned int  IOINTFLG1;        /* Регистр флагов 1, +10 */  
    volatile unsigned int  IOINTFLG2;        /* Регистр флагов 1, +11 */  
} GPIORegs;
```

/ Регистровая память входов-выходов общего назначения */*

```
#define GPIO      ((ioport GPIORegs*)0x1C06h)
```

/ Регистр разрешения и флагов прерываний */*

```
#define IER1      (*(volatile unsigned int *)0x000045h)
```

```
#define IFR1      (*(volatile unsigned int *)0x000046h)
```

Ввод-вывод по опросу

```
void main() {  
    // Локальные данные  
    int blink = 0, cnt = 0;  
    // Начальные установки  
    GPIO->IODIR1 &= ~0x10;  
    GPIO->IODIR1 |= 0x20;  
    // Бесконечный цикл  
    while( 1 ) {  
        // Вывод данных  
        if( blink == 1 ) {  
            GPIO->IODATAOUT1 ^= 0x20;  
            cnt=15000; while(cnt-- > 0);  
        }  
        // Ввод данных  
        if( !(GPIO->IOINDATA1 & 0x10) ) {  
            cnt=15000; while(cnt-- > 0);  
            while( !(GPIO->IOINDATA1 & 0x10) );  
            cnt=15000; while(cnt-- > 0);  
            if( blink == 1 ){  
                GPIO->IODATAOUT1 &= ~ 0x20;  
                blink = 0;  
            } else  
                blink = 1;  
        }  
    }  
}
```

// Флаг режима индикации и счетчик

// GPIO4 – направление ввода (кнопка)
// GPIO5 – направление вывода(светодиод)

// Если режим индикации 1
// Инверсия выхода GPIO5
// Задержка

// Если кнопка нажата
// Задержка
// Ожидание отпускания кнопки
// Задержка
// Если режим индикации
// Выключить светодиод
// Снять режим индикации
// Иначе
// Установка режима индикации

Ввод-вывод через прерывания

```
interrupt void GPIO_ISR (void) {
    int cnt = 0;                                /* Счетчик */
    /* Ветвление по источникам прерываний */
    if( GPIO->IOINTFLG1 & 0x10 ) {                /* Прерывание от кнопки (GPIO4) */
        GPIO->IODATAOUT1 ^= ~ 0x20;              /* Инверсия выхода светодиода (GPIO5) */
        cnt=15000; while(cnt-- > 0);             /* Задержка */
        while( !(GPIO->IOINDATA1&0x10));          /* Ожидание отпускания кнопки */
        cnt=15000; while(cnt-- > 0);             /* Задержка */
        GPIO->IOINTFLG1 &= ~0x10;                 /* Сброс флага прерывания от GPIO4 */
    }                                              /* else if ... else ... */
    IER1 |= 0x20;                                /* Разрешение прерываний от GPIO */
}

void main() {
    /* Локальные данные */
    long isr = (long)&GPIO_ISR;
    /* Начальные установки */
    GPIO->IODIR1 &= ~0x10;                        /* GPIO4 – направление ввода (кнопка) */
    GPIO->IODIR1 |= 0x20;                         /* GPIO5 – направл. вывода (светодиод) */
    /* Настройка прерываний */
    ISR_plug( 21, isr );                         /* Установка вектора прерываний */
    GPIO->IOINTEDG1 |= 0x10;                      /* GPIO4 – по заднему фронту */
    GPIO->IOINTEN1 |= 0x10;                      /* GPIO4 – разрешение прерываний */
    IER1 |= 0x20;                                /* Разрешение прерываний от GPIO */
    /* Бесконечный цикл */
    while( 1 );
}
```