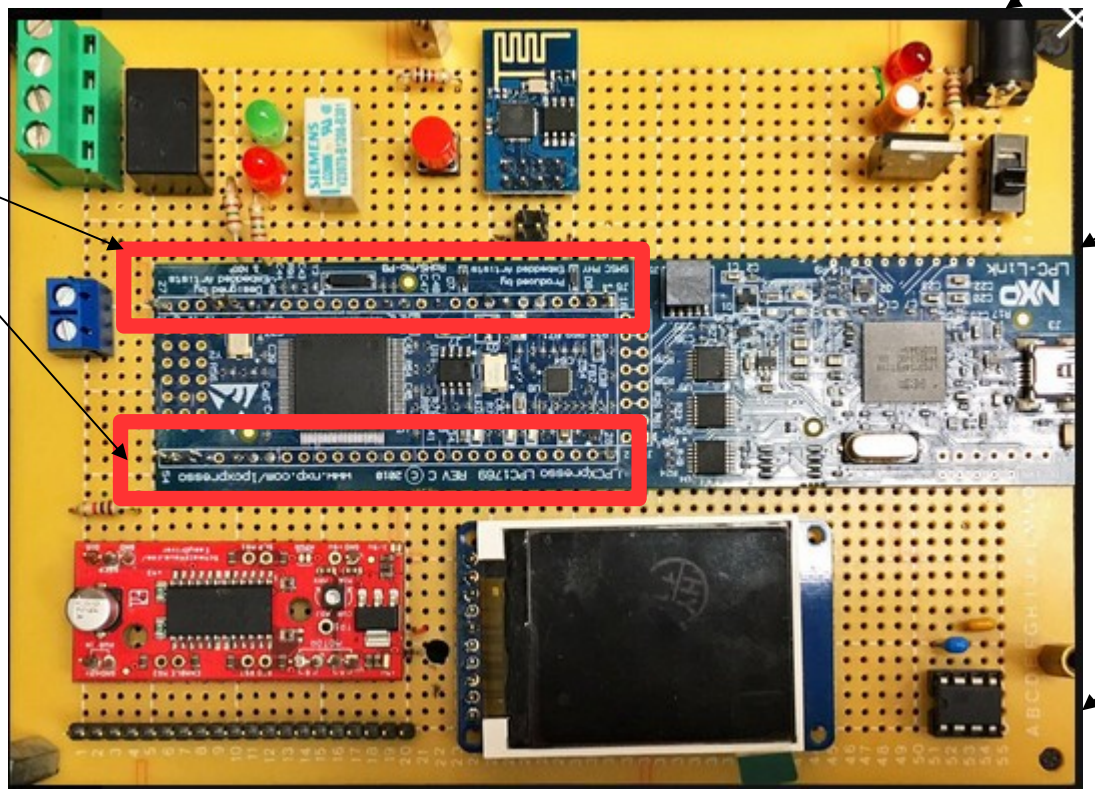
J2/(or 6 for rev. B) Connector with CPU GPIO Pins

J1: PWR Connector

J2 (Rev. D) or J6 (Rev.B) connector with CPU GPIO Pin



LPC1769

SPI I/F Flash

Dimension: 16 x 11 mm or 6.25 x 4.50 inch

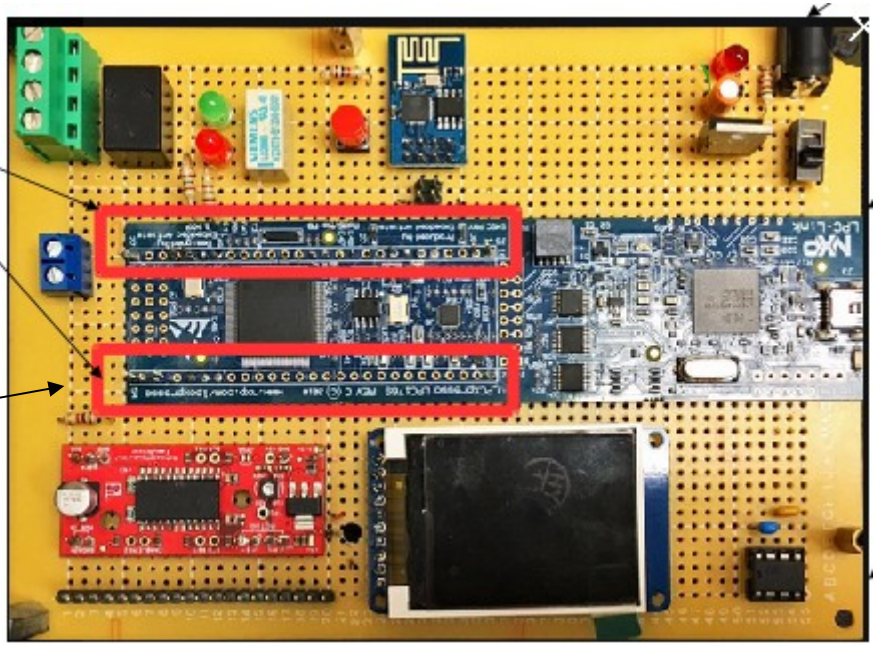
J2 (Rev. D) or J6 (Rev. B) Connector with CPU GPIO Pins

Table 1. J2 Pin Assignment

P1.31	AD0.5	P1.31	J2-20
P0.2		P0.2	J2-21
P0.3		P0.3	J2-22
P0.21		P0.21	J2-23
P0.22		P0.22-RED_LED	J2-24
P0.27		P0.27-I2C_SDA	J2-25
P0.28		P0.28-I2C_SCL	J2-26
P2.13		P2.13	J2-27

Reference: SCH design Rev D.

J2 connector with CPU GPIO Pin

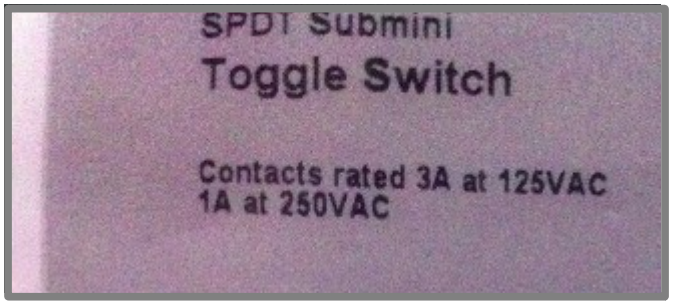


Board: 16 x 11 mm or 6.25 x 4.50 inch

SPDT switch for GPIO input testing

Table 2. J2 Connectivity

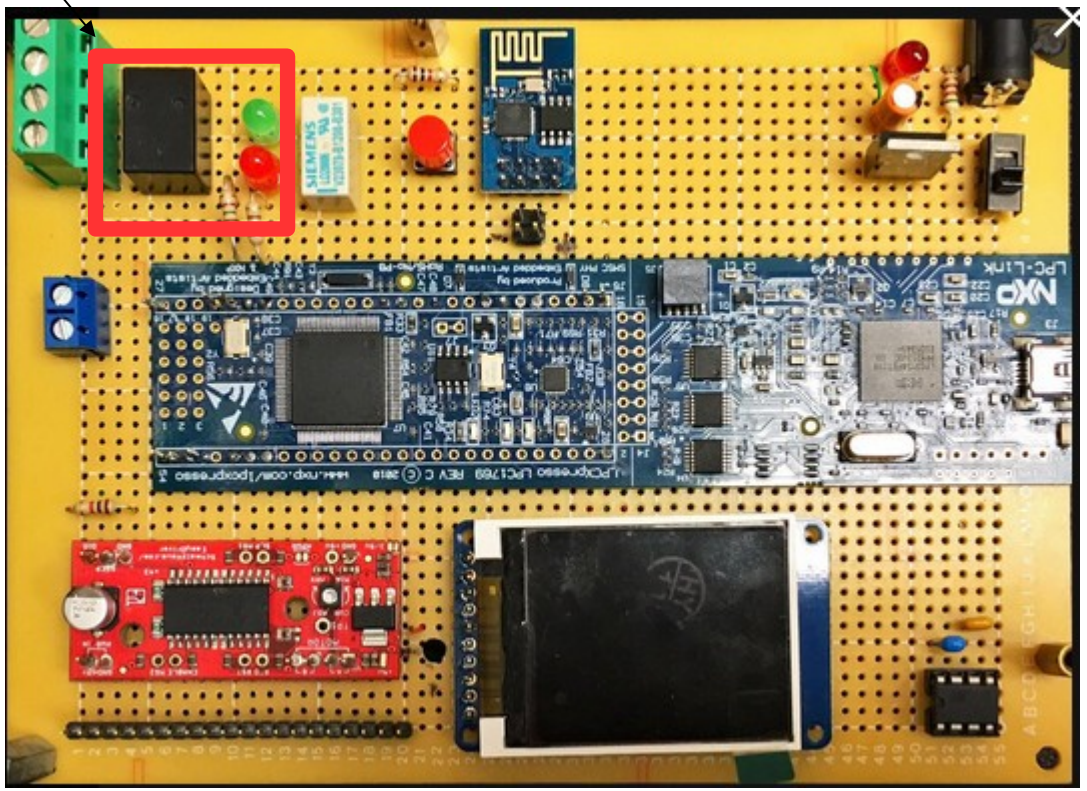
CPU	J2	Description
P0.2	J2-21	GPIO output
P0.3	J2-22	GPIO input

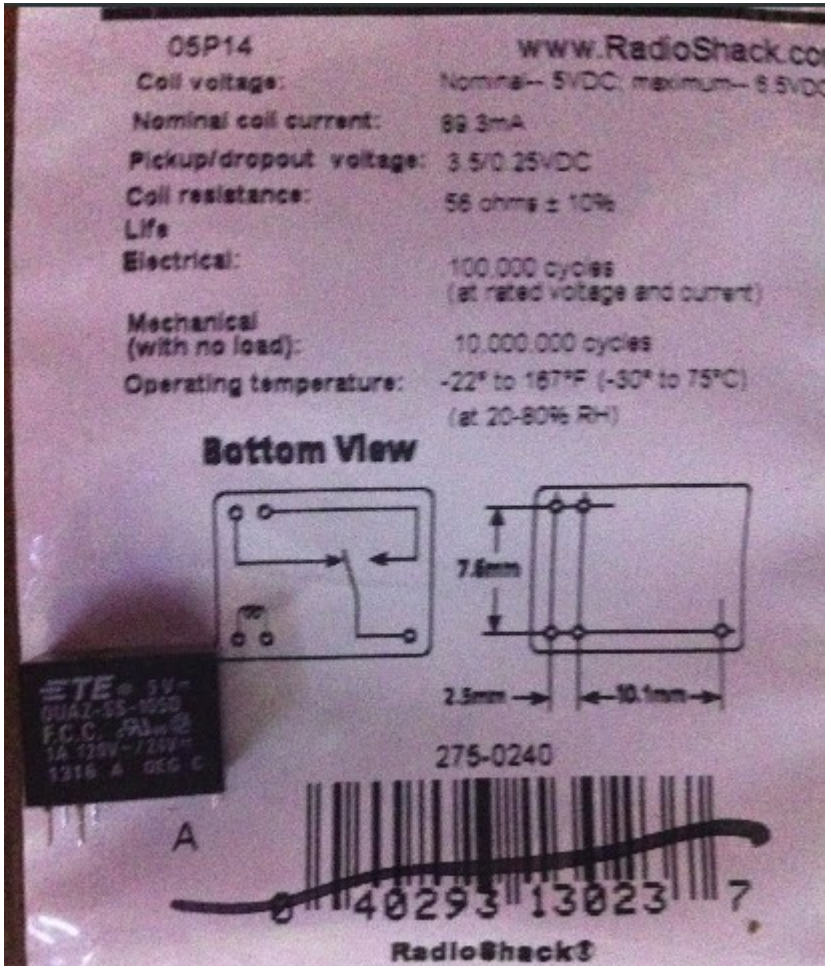


A Single Pole Double Throw (SPDT) switch is a switch that only has a single input and can connect to and switch between 2 outputs.

GPIO (GPP) Output with SSR

SSR: Solid State Relay

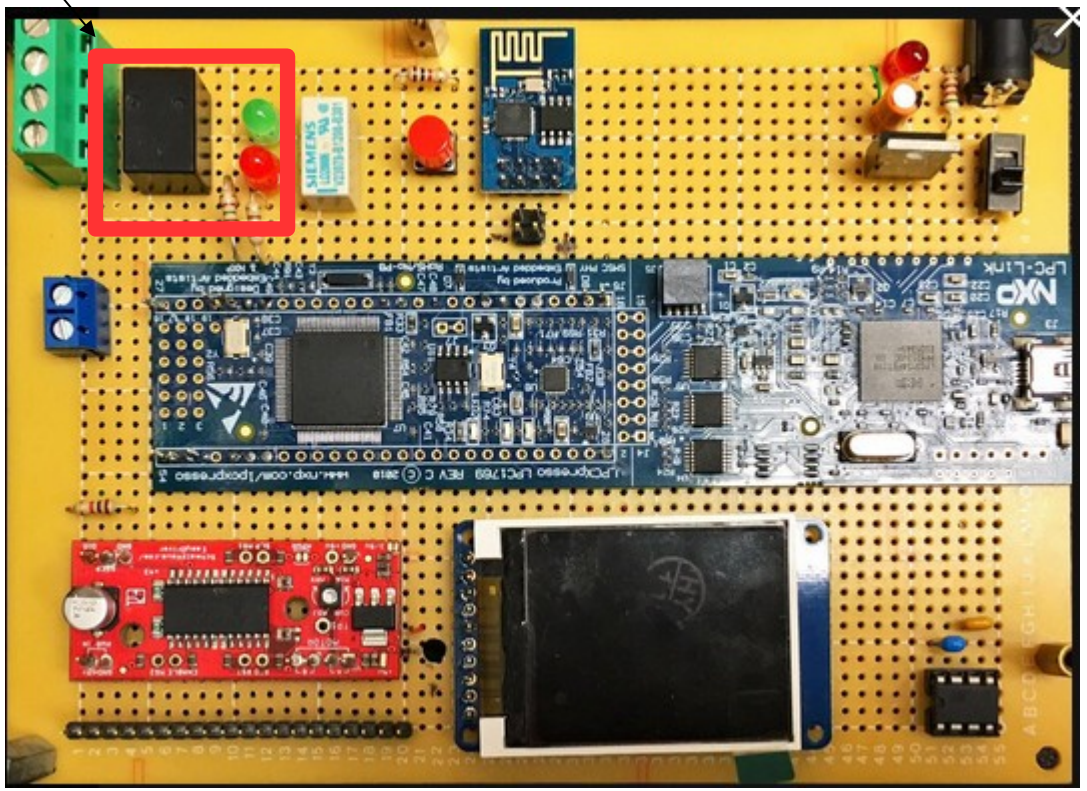




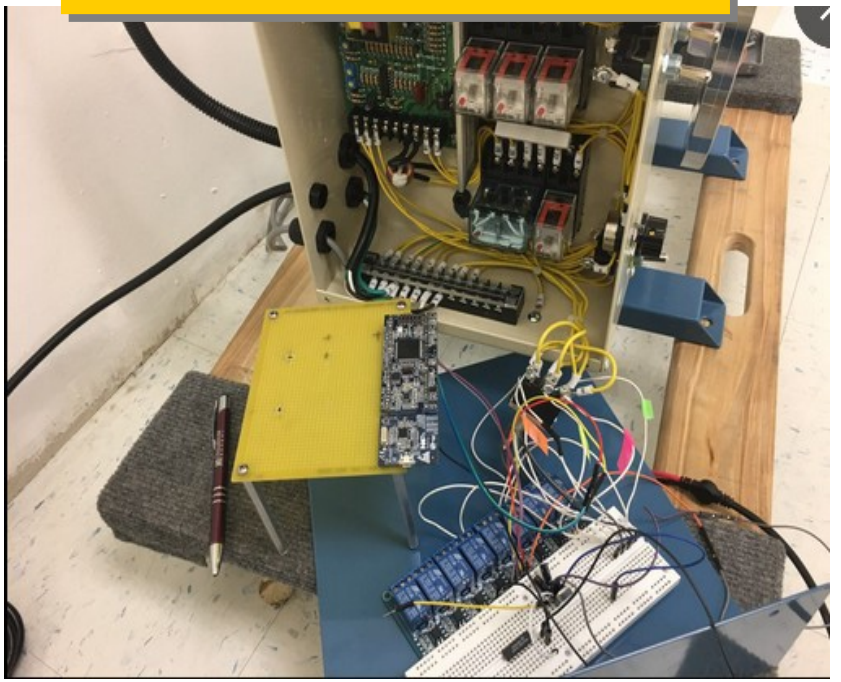
SSR: Solid State Relay

GPIO (GPP) Output SSR Application

SSR: Solid State Relay



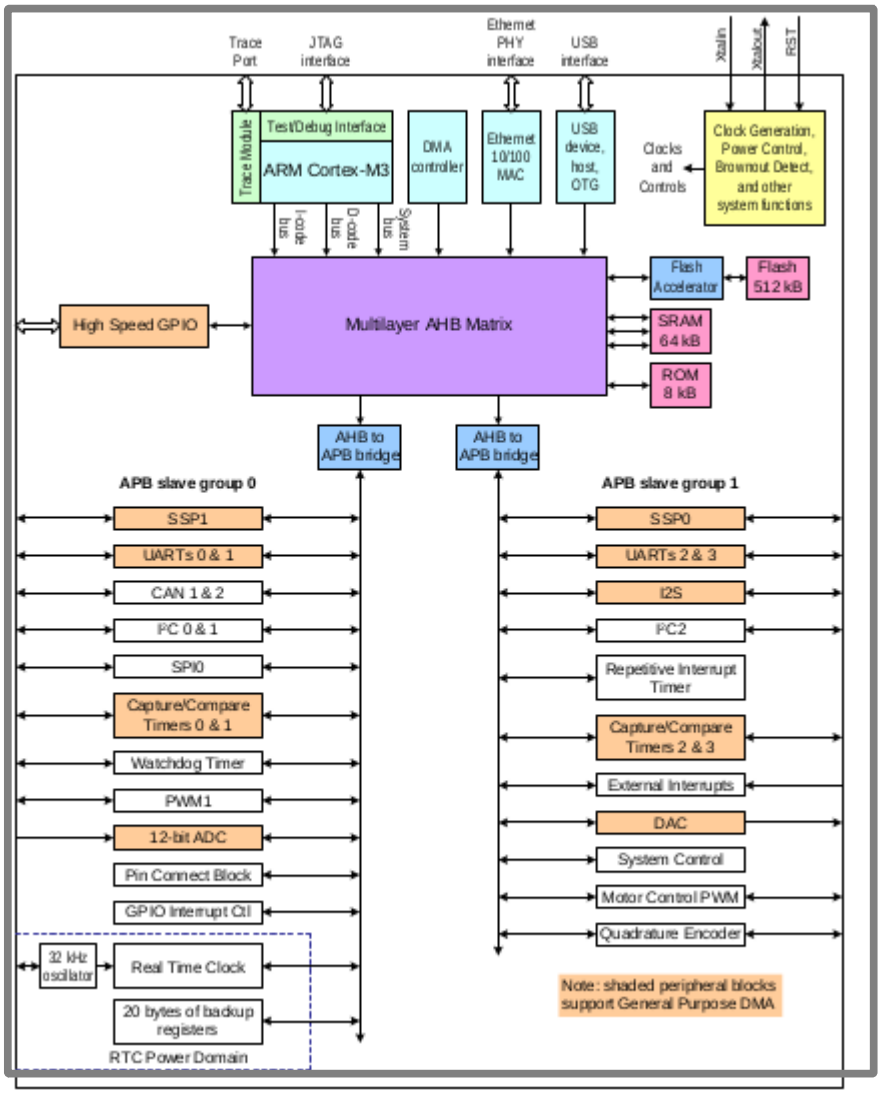
SMT-POS Positioner



GPIO (GPP) Controller and SPRs

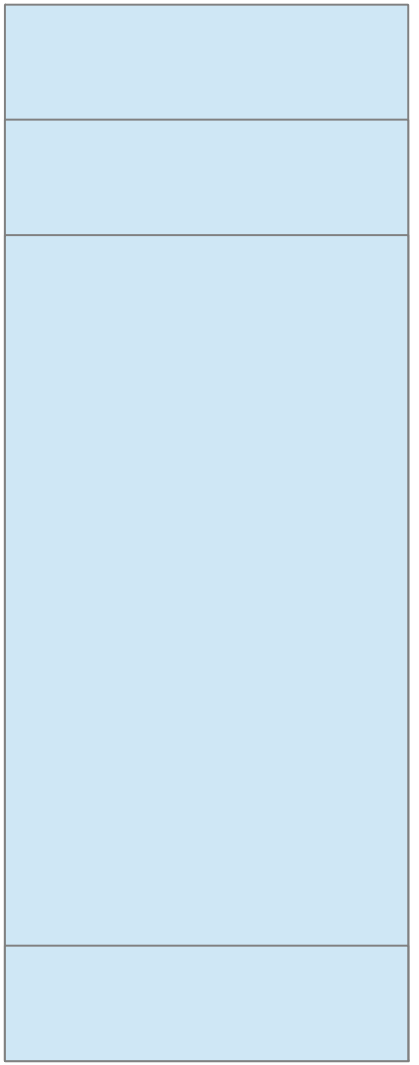
1. Definition of SPRs: Special purpose registers are those to perform init and config functions.
2. 32 bit each with unique address.
3. In IDE (software, e.g., eXpresso in this class), *.h file with something looks like the following
#define SPR 0x2000_0000
map the CPU architecture to the arm gcc compiler

```
LPC_GPIO0->FIODIR

LPC_GPIO0->FIOSET

LPC_GPIO0->FIOCLR
```



CPU
Ref: pp. 9 datasheet

Memory Map



Pwr up address
0x0000_0000

GPIO (GPP) SPRs

GPIO SPRs

Reference: Chapter 9: LPC176x/5x General Purpose Input/Output (GPIO) Rev. 3.1 — 2 April 2014 User manual

```
LPC_GPIO0->FIODIR

LPC_GPIO0->FIOSET

LPC_GPIO0->FIOCLR
```

```
#include "../../../tools/redlib/include/stdint.h"
#include "../../../tools/redlib/include/stdio.h"

#ifdef __USE_CMSIS      //board init & config stuff
 /* CMSIS: the Cortex Microcontroller Software Interface Standard
                        */
#include "LPC17xx.h"
```

Background:

From CPU datasheet, GPIOs are configured using the following registers:
1. Power: always enabled.
2. Pins: See Section 8.3 for GPIO pins and their modes.
3. Wake-up: GPIO ports 0 and 2 can be used for wake-up if needed, see (Section 4.8.8).
4. Interrupts: Enable GPIO interrupts in IO0/2IntEnR (Table 115) or IO0/2IntEnF (Table 117). Interrupts are enabled in the NVIC using the appropriate Interrupt Set Enable register.

9.4 Pin description

P0[30:0] [1] ; Type: Input/Output; Description: General purpose in/output.

From SCH pdf doc make connection from this GPP pin to physical connector pin out, e.g., J2-21, J2-22 etc.

GPIO SPRs Description

```
LPC_GPIO0->FIODIR

LPC_GPIO0->FIOSET

LPC_GPIO0->FIOCLR
```

Table 102. GPIO register map, from CPU data sheet pp133

FIODIR Fast GPIO Port Direction control register, controls the direction of each port pin

FIOSET Fast Port Output Set register using FIOMASK. This register
R/W
controls the state of output pins. Writing 1s produces highs at the corresponding port pins. Writing 0s has no effect. Reading this register returns the current contents of the port output register. Only bits enabled by 0 in FIOMASK can be altered.

FIOCLR Fast Port Output Clear register using FIOMASK. This register WO controls the state of output pins. Writing 1s produces lows at the corresponding port pins. Writing 0s has no effect. Only bits enabled by 0 in FIOMASK can be altered.

FIOxDIR Description

```
LPC_GPIO0->FIODIR

LPC_GPIO0->FIOSET

LPC_GPIO0->FIOCLR
```

Reference: CPU Datasheet, 9.5.1 GPIO port Direction register FIOxDIR (FIO0DIR to FIO4DIR- 0x2009 C000 to 0x2009 C080)

Table 104. Fast GPIO port Direction register FIO0DIR to FIO4DIR - addresses 0x2009 C000 to 0x2009 C080) bit description

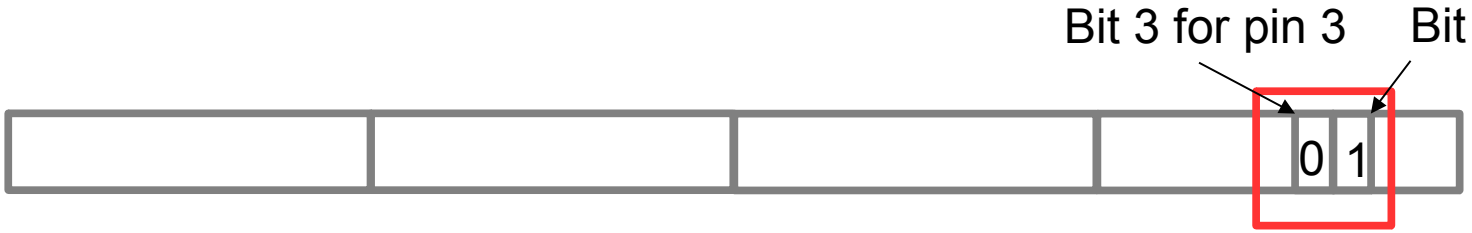
FIO0DIR
(Compiler: LPC_GPIO0->FIODIR)

Bit	Symbol	Value	Description
31:0	FIO0DIR		Fast GPIO Direction PORTx control bits. Bit 0 in FIOxDIR controls pin Px.0, bit 31 in FIOxDIR controls pin Px.31.
	FIO1DIR		
	FIO2DIR	0	Controlled pin is input.
	FIO3DIR		
	FIO4DIR	1	Controlled pin is output.

Example: FIO0DIR Init & Config,

Set P0.2 output, “1”
P0.3 input, “0”

0x4




```
LPC_GPIO0->FIODIR = 0x4;
```

*.h Mapping the CPU Architecture

```
LPC_GPIO0->FIODIR

LPC_GPIO0->FIOSET

LPC_GPIO0->FIOCLR
```

Example: LPC1769.h
Maps compiler to architecture

```
/*--General Purpose Input/Output (GPIO) --*/
typedef struct
{
  union {
    __IO uint32_t FIODIR;
    struct {
      __IO uint16_t FIODIRL;
      __IO uint16_t FIODIRH;
    };
    struct {
      __IO uint8_t  FIODIR0;
      __IO uint8_t  FIODIR1;
      __IO uint8_t  FIODIR2;
      __IO uint8_t  FIODIR3;
    };
  };
//…
//…
} LPC_GPIO_TypeDef;
```

*.h Mapping the Memory Map

```
LPC_GPIO0->FIODIR

LPC_GPIO0->FIOSET

LPC_GPIO0->FIOCLR
```

Example: LPC1769.h
Maps compiler to memory map

1

Peripheral Memory Map

```
/*****************************************************/
/*      Peripheral memory map                        */
/*****************************************************/
/* Base addresses                                    */
#define LPC_FLASH_BASE        (0x00000000UL)
#define LPC_RAM_BASE          (0x10000000UL)
#define LPC_GPIO_BASE         (0x2009C000UL)
#define LPC_APB0_BASE         (0x40000000UL)
#define LPC_APB1_BASE         (0x40080000UL)
#define LPC_AHB_BASE          (0x50000000UL)
#define LPC_CM3_BASE          (0xE0000000UL)
```

2

```
/* GPIOs */
#define LPC_GPIO0_BASE        (LPC_GPIO_BASE + 0x00000)
#define LPC_GPIO1_BASE        (LPC_GPIO_BASE + 0x00020)
#define LPC_GPIO2_BASE        (LPC_GPIO_BASE + 0x00040)
#define LPC_GPIO3_BASE        (LPC_GPIO_BASE + 0x00060)
#define LPC_GPIO4_BASE        (LPC_GPIO_BASE + 0x00080)
```

3

```
/************ Peripheral declaration ***************/
#define LPC_SC                ((LPC_SC_TypeDef       *) LPC_SC_BASE       )
#define LPC_GPIO0             ((LPC_GPIO_TypeDef     *) LPC_GPIO0_BASE    )
#define LPC_GPIO1             ((LPC_GPIO_TypeDef     *) LPC_GPIO1_BASE    )
#define LPC_GPIO2             ((LPC_GPIO_TypeDef     *) LPC_GPIO2_BASE    )
#define LPC_GPIO3             ((LPC_GPIO_TypeDef     *) LPC_GPIO3_BASE    )
#define LPC_GPIO4             ((LPC_GPIO_TypeDef     *) LPC_GPIO4_BASE    )
```

GPIO Init and Config

```
int main(void)
{
    // Force the counter to be
placed into memory
    volatile static int i = 0 ;
    //Set pin 0.2 as output
GPIOinitOut(0,2); //port number,
pin number
//Set pin 0.3 as output
GPIOinitOut(0,3);

setGPIO(0, 3);
clearGPIO(0, 2); //port number,
pin number
```

```
void GPIOinitOut(uint8_t portNum,
uint32_t pinNum)
{
if (portNum == 0)
{
LPC_GPIO0->FIODIR |= (1 <<
pinNum);
}
else if (portNum == 1)
{
LPC_GPIO1->FIODIR |= (1 <<
pinNum);
```

```
void clearGPIO(uint8_t portNum, uint32_t
pinNum)
{
if (portNum == 0)
{
LPC_GPIO0->FIOCLR = (1 << pinNum);
printf("Pin 0.%d has been cleared.\n",
pinNum);
}
```

```
void setGPIO(uint8_t portNum,
uint32_t pinNum)
{
if (portNum == 0)
{
LPC_GPIO0->FIOSET = (1 <<
pinNum);   //1 as output
printf("Pin 0.%d has been
set.\n",pinNum);
}
```