



~~程序代写代做 CS 编程辅导~~

Lecture 19



# GPIO

# General Purpose Input/Output

WeChat: cstutorcs  
Assignment Project Exam Help

Email: tutorcs@163.com

Bad joke of the day:  
QQ: 749389476

**Why are assembly programmers always wet?**

<https://tutorcs.com>

**Because they work below C-level**



# Solution to Quiz 5

~~程序代写代做 CS 编程辅导~~

**Task:** Write a subroutine that checks whether a given integer n is prime



```
; Subroutine: is_prime
; Inputs: unsigned int n
; Output: binary value in R13
;           R13 = 1 if n is prime
;           R13 = 0 if n is composite
;
; All other core registers in R4-R15 unchanged
; Subroutine does not access addressed memory locations
```

WeChat: cstutorcs

Assignment Project Exam Help

**Approach:**

Email: tutorcs@163.com

- Check if n is divisible by 2, 3, 4, ..., n-1
- If it is divisible by any of these numbers, the number is composite
- If not divisible, the number is prime
- Works only if n>3, the cases n = 0, 1, 2 need separate logic

QQ: 749389476

<https://tutorcs.com>



# Solution to Quiz 5 v.1

程序代写代做 CS编程辅导

is\_prime\_1:

```
push    R5
push    R12
clr.w   R13
cmp.w   #2, R6
jlo     ret_from_is_prime_1
jne     larger_than_two
        ; if n<2 not prime
        ; if (n>=2 and n!= 2) check n
        ; n=2 is prime
        mov.w   #1, R13
        jmp     ret_from_is_prime_1
```



larger\_than\_two:

```
        mov.w   #2, R5           ; start checking for divisibility by 2
```

check\_divisibility\_1:

```
        call    #is_divisible
        tst.w   R12             ; if divisible, then R12==1, n is composite
        jnz    ret_from_is_prime_1
        inc.w   R5
        cmp.w   R6, R5           ; check divisibility by next integer
        jne    check_divisibility_1
        ; n-1
```

; We land here only if the number is prime
 mov.w #1, R12

ret\_from\_is\_prime\_1:

```
        pop    R12
        pop    R5
        ret
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

**Note:** is\_divisible modifies R12  
Need to push/pop R12!!



# Solution to Quiz 5

~~程序代写代做 CS 编程辅导~~

**What about algorithmic complexity?** How many calls to `is_divisible`?

Depends on the nature of the number:

- If  $n$  is even, then only ~~one~~ call to `is_divisible` is divisible by 2
- If  $n$  is an odd multiple of 3, then ~~two~~ calls: check for divisibility by 2 and 3
- ...
- If  $n$  is prime, then  ~~$n-2$~~  calls to `is_divisible` with 2, 3, ...,  $n-1$

**WeChat: cstutorcs**

**Assignment Project Exam Help**

How can we speed things up?

- No need to check  $n > 2$  if it is an even number. ~~Email: tutorcs@163.com~~ bit.w #BIT0, R6
- For odd  $n$ , it suffices to check divisibility by odd numbers {3, 5, ...,  $n-2$ }
- For  $n > 4$ , no need to check for factors  $> n/2$  check only {3, 5, ...,  $\text{floor}(n/2)$ }
- Actually no need to check for factors  $> \text{sqrt}(n)$

**QQ: 749389476**



# Solution to Quiz 5 v.2

程序代写代做 CS编程辅导

```
; Somewhat improved efficiency
; Test only odd n for division
; Note (n+1)/2 > sqrt(n)
; takes quite some time
is_prime_2:
    push    R5
    push    R7
    push    R12
    ; n is more likely not prime P(0 to prime) ~ 1/ln(n)
    clr.w   R13
    cmp.w   #3, R6
    jeq    found_prime
    cmp.w   #2, R6
    jeq    found_prime
    jlo    ret_from_is_prime_2
    jne    larger_than_two_2
    mov.w   #1, R13
    jmp    ret_from_is_prime_2
larger_than_two_2:
    ; if n>2 is even it is not prime
    bit.w   #BIT0, R6
    jnc    ret_from_is_prime_2
```

Assignment Project Exam Help  
Email: tutorcs@163.com  
QQ: 749389476  
<https://tutorcs.com>





# Solution to Quiz 5 v.2

程序代写代做 CS编程辅导

```
; test divisibility by odd numbers only
mov.w #3, R5 ; R7=n
mov.w R6, R7 ; R7=n+1
inc.w R7 ; unsigned divide by 2 using cleared carry bit
clrc
rrc.w R7 ; R7=(n+1)/2

check_divisibility:
; check if n is divisible by 3, 5, 7, 9, ..., n-1
call #is_divisible
tst.w R12
jnz ret_from_is_prime_2 ; found a factor, n is composite

incd.w R5 ; next odd number
cmp.w R7, R5 ; will stop checking at R7 = (n+1)/2
; this works because both R5 and R6 are odd
jlo check_divisibility

; We land here only if the number is prime
found_prime:
mov.w #1, R13 ; QQ: 749389476

ret_from_is_prime_2:
pop R12
pop R7
pop R5
ret
```

WeChat: cstutorcs  
Assignment Project Exam Help  
Email: tutorcs@163.com  
<https://tutorcs.com>  
~4 times faster for prime numbers



; test divisibility by odd numbers only

mov.w #3, R5

mov.w R6, R7

inc.w R7

clrc

rrc.w R7

; R7=n

; R7=n+1

; unsigned divide by 2 using cleared carry bit

; R7=(n+1)/2

**check\_divisibility:**

; check if n is divisible by 3, 5, 7, 9, ..., n-1

call #is\_divisible

tst.w R12

jnz ret\_from\_is\_prime\_2 ; found a factor, n is composite

incd.w R5

cmp.w R7, R5

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

; We land here only if the number is prime

**found\_prime:**

mov.w #1, R13

QQ: 749389476

**ret\_from\_is\_prime\_2:**

pop R12

pop R7

pop R5

ret



# Learning From Quiz 4

程序代写代做 CS编程辅导

How can we *improve* following code? Logic is correct

```
69 is_divisible:
70     push    R5
71     push    R6
72
73     clr.w   R12
74
75 check:      cmp.w   R5, R6
76     jlo     found_remainder
77
78     sub.w   R5, R6
79     jhs     check
80
81 found_remainder:      Email: tutorcs@163.com
82     tst.w   R6
83     jnz     ret_from_is_divisib
84
85     mov.w   #1, R12
86
87 ret_from_is_divisible: https://tutorcs.com
88     pop     R6
89     pop     R5
90     ret
91
```



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

https://tutorcs.com

No need to push/pop R5  
since it is not modified



# Learning From Quiz 4

程序代写代做 CS编程辅导

How can we *improve* following code?

Logic is correct

```
67 is_divisible:
68     push    R6
69     clr.w   R12
70
71 check_R6:
72     tst.w   R6
73     jz      yes_divisible
74
75     sub.w   R5, R6
76     cmp.w   R5, R6
77     jlo     ret_is_divisible
78     jz      yes_divisible
79     jhs     check_R6
80
81 yes_divisible:
82     add.w   #1, R12
83     jmp     ret_is_divisible
84
85 ret_is_divisible:
86     pop    R6
87     ret
```



WeChat: cstutorcs

Assignment Project Exam Help

No need; sub.w already sets the status bits!!

Email: tutorcs@163.com

QQ: 749389476

#1, R12  
ret\_is\_divisible

<https://tutorcs.com>

R6

No need to jump  
ret\_is\_divisible is the next line already!!!



# Learning From Quiz 4

程序代写代做 CS编程辅导

How can we *improve* following code?

```
81 not_divisible:  
82     add.w  
83     pop  
84     jmp    main  
85  
86 divisible:  
87     add.w #1, R12  
88     pop    R6  
89     jmp    ret_to_main  
90  
91 ret_to_main:  
92     ret  
93
```



No need:  $x + 0 = x$

WeChat: cstutorcs

No need to jump to  
the next line

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

No need to repeat lines 83-84

divisible:

<https://tutorcs.com>

add.w #1, R12

not\_divisible:

pop R6  
ret



# Learning From Quiz 4

程序代写代做 CS编程辅导

What is the issue with this code?

```
65 is_divisible:  
66     push    clr.w  
67     cmp.w  R6, R5  
68     jlo      test  
69  
70  
71 subtract:  
72     sub.w  R5, R6  
73     jmp    is_divisible  
74  
75  
76 test:  
77     cmp.w  #0, R6  
78     jne    subtract  
79  
80     mov.w  #1, R12  
81  
82 ret_main:  
83     pop    R6  
84     ret  
85 --
```



Pushes to stack with each iteration!!!

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

#0, R6

subtract

QQ: 749389476

<https://tutorcs.com>

R6

Pops only once

(R6) -> PC -> crash



# This is a Mistake

程序代写代做 CS编程辅导

How do we fix it?

```
65 is_divisible:  
66     push  
67     clr.w  
68  
69     cmp.w  
70     jlo  
71  
72 subtract:  
73     sub.w  
74     jmp  
75  
76 test:  
77     cmp.w  
78     jne  
79  
80     mov.w  
81  
82 ret_main:  
83     pop  
84     ret  
85 --
```



WeChat: cstutorcs  
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

```
65 is_divisible:  
66     push  
67     clr.w  
68  
69 compare:  
70     cmp.w  
71     jlo  
72  
73 subtract:  
74     sub.w  
75     jmp  
76  
77 test:  
78     cmp.w  
79     jne  
80  
81     mov.w  
82  
83 ret_main:  
84     pop  
85     ret  
R6
```

# How MCUs are used in the Real-World



~~程序代写代做 CS 编程辅导~~

Not the way we have used them so far: we have only used the CPU and memory (RAM/FRAM) of  to do basic data manipulations

When treated like this, the MCU is a very limited computer:

- No real input or interaction with the user/environment
- We defined (hardcoded) data in RAM/FRAM as input to some logic
- Output is limited too: we peek at registers using CCS to view the input

~~Assignment Project Exam Help~~

An MCU is intended to do much more

- Interact with the user / Environment / other MCUs  
~~Email: tutorcs@163.com~~
- **Input** from buttons, sensors, other MCUs  
~~QQ: 749389476~~
- **Output** via displays, motors (motor drivers), actuators, control circuitry ...

<https://tutorcs.com>

All input / output to the MCU is through **Input/Output Pins**

# The MSP430FR6989 Launchpad



程序代写代做 CS 编程辅导

This part enables interface to a PC and enables debugging

**Headers** with access to selected pins connect I/O devices e.g., sensors, motors, logic analyzer...



Only I/O we will use

Push Button S1

Red LED

Push Button S2

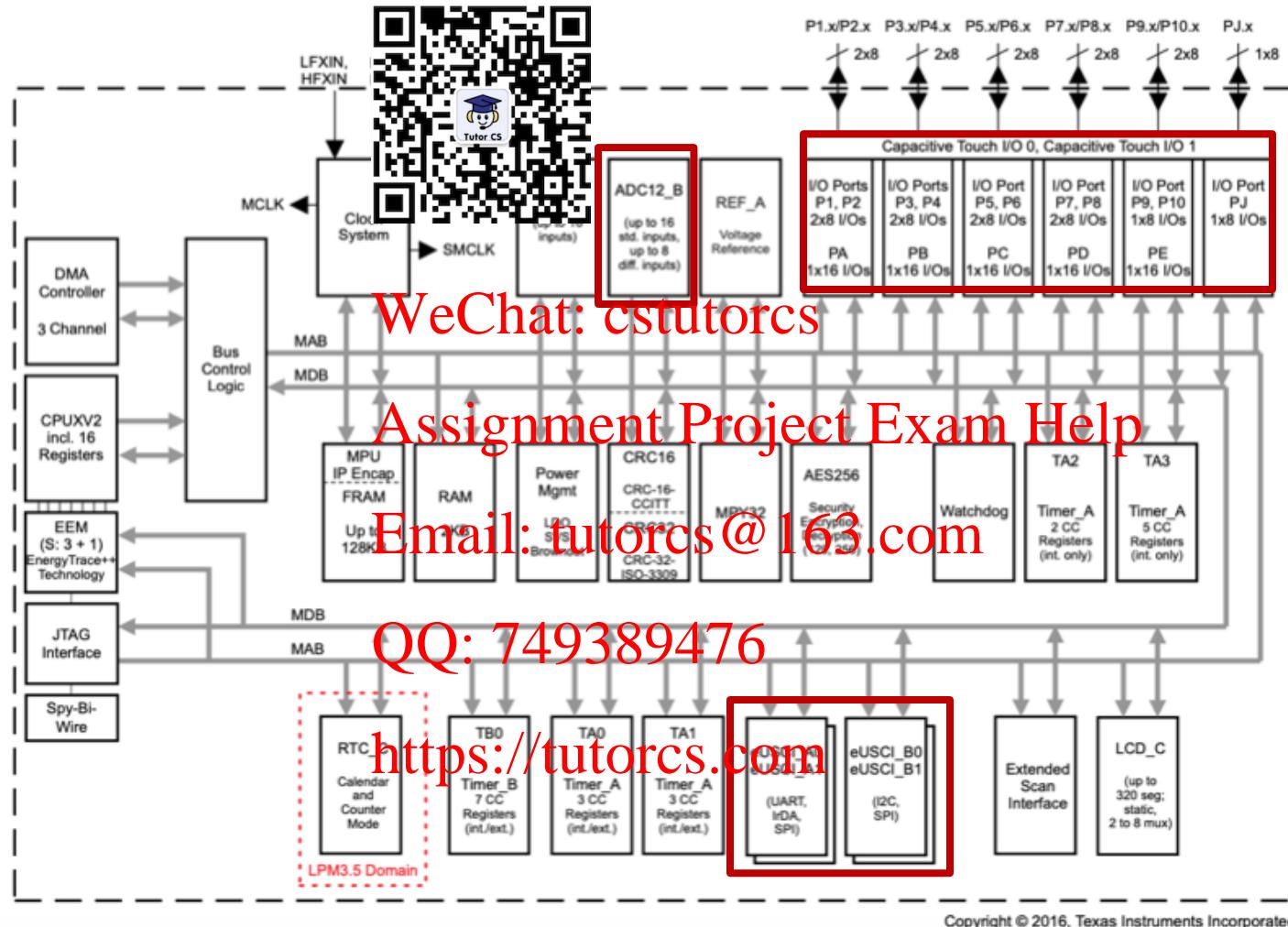
Green LED



# MSP430FR6989 I/O Options

程序代写代做 CS 编程辅导

Ports for GPIO



# I/O Through Standard Protocols



~~程序代写代做 CS 编程辅导~~

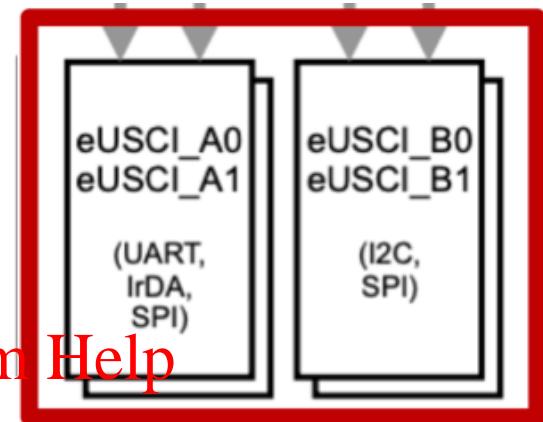
The enhanced Universal Serial Communication Interfaces

(eUSCI\_A and eUSCI\_B)



Several standard protocols for I/O

- **Serial Peripheral Interface (SPI)**
- **Inter Integrated Circuit (I<sup>2</sup>C, I2C, IIC)**
- **Universal Asynchronous Receiver Transmitter (UART)**



How to use?

- Dedicated pins for I/O
- Registers for configuration
- Devices that use these protocols  
<https://tutorcs.com>
- e.g., SPI sensors, I<sup>2</sup>C sensors, I<sup>2</sup>C motor drivers etc.

Email: [tutorcs@163.com](mailto:tutorcs@163.com)

QQ: 749389476 All the details in [slau367p.pdf](#)  
Posted to Carmen under Resources



# GPIO Ports P1 – P10

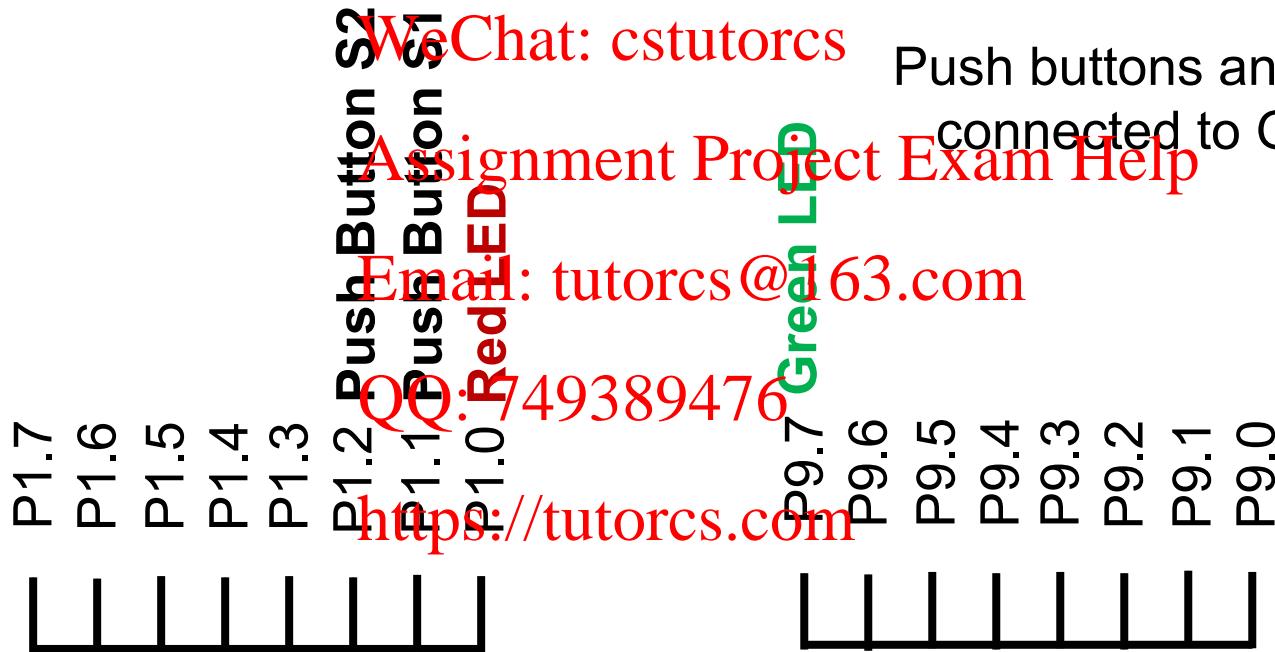
~~程序代写代做 CS 编程辅导~~

Our MCU has 10 **General Purpose Input Output (GPIO) Ports P1 – P10**

TI refers to these as **Digital I/O Ports**

(or PA – PJ)

- Each port has **8 pins**
- Pins are labeled as **Px<sub>y</sub>** where **x** is the port number, **y** is the pin number



Pins of Port 1

Pins of Port 9

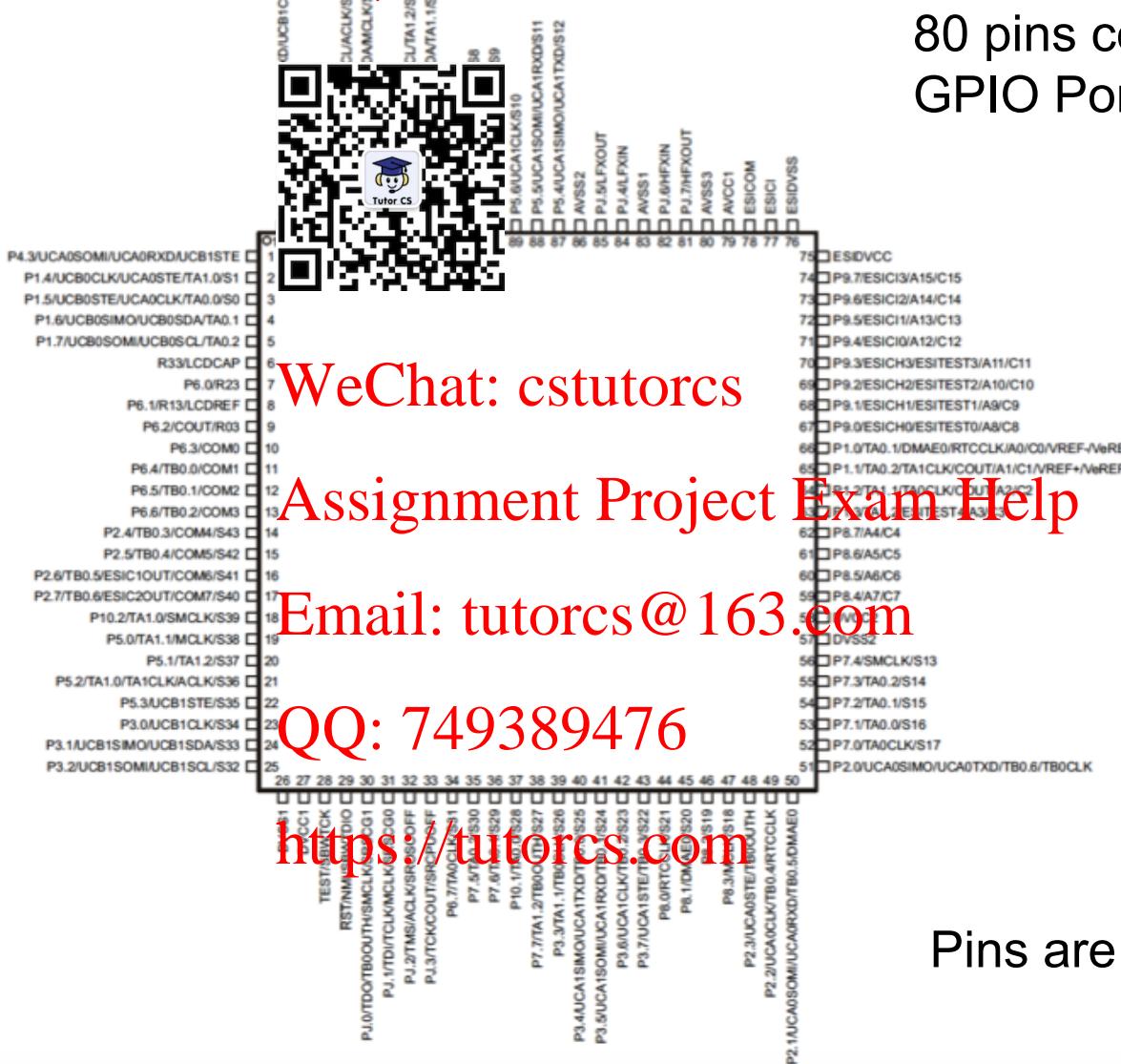


# MSP430FR6989IPZ Pinout

~~程序代写代做 CS 编程辅导~~

slas789d.pdf

80 pins connected to  
GPIO Ports



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Pins are multiplexed



# MSP430FR6989IPZ Pinout

程序代写代做 CS 编程辅导

Pins are **multiplexed**  
many are connected to  
multiple peripherals and need  
to be configured by selecting  
one functionality



WeChat: cstutorcs

Green LED  
Assignment Project Exam Help

Email: tutorcs@163.com

QC: 7798976

<https://tutorcs.com>

Red LED  
Push Buttons S1 & S2

P10.0/SIMCLK/S4	P4.7/UCB1SOMI/UCB1SLC/T1A12/S5
P4.6/UCB1SIMO/UCB1SDA/T1A11/S6	P4.5/UCB1CLK/T1A1.0/S7
P4.4/UCB1STE/T1A1CLK/S8	P5.7/UCA1SOMI/UCA1RXD/S11
P5.6/UCA1CLK/S10	P5.5/UCA1SIMO/UCA1TXD/S12
P5.7/UCA1STE/TBCLK/S9	P5.4/UCA1FXIN
AVSS2	PJ.5/LFXOUT
PJ.6/HFXIN	AVSS1
PJ.7/HFXOUT	PJ.4/LFXIN
AVSS3	AVSS2
ESIC1	AVCC1
ESIDVSS	ESICOM

75	ESIDVCC
74	P9.7/ESICI3/A15/C15
73	P9.6/ESICH3/ESITEST3/A11/C11
72	P9.5/ESICI1/A13/C13
71	P9.4/ESICI0/A12/C12
70	P9.3/ESICH3/ESITEST3/A11/C11
69	P9.2/ESICH2/ESITEST2/A10/C10
68	P9.1/ESICH1/ESITEST1/A9/C9
67	P9.0/ESICH0/ESITEST0/A8/C8
66	P1.0/TA1.1/MAE0/R0/CLK/A10/C1/VREF-/VeREF-
65	P1.1/TA0.2/TA1CLK/COUT/A1/C1/VREF+/VeREF+
64	P1.2/TA1.1/TA0CLK/COUT/A2/C2
63	P1.3/TA1.2/ESITEST4/A3/C3
62	P8.7/A4/C4
61	P8.6/A5/C5
60	P8.5/A6/C6
59	P8.4/A7/C7
58	DVCC2
57	DVSS2

# Registers Controlling GPIO Ports



程序代写代做 CS编程辅导

Each port is configured and controlled by a set of **8-bit registers**

**Px.y** is controlled by **bit y** in register corresponding to **port x**

e.g.

P1.7  
P1.6  
P1.5  
P1.4  
P1.3



e.g., output HIGH on p1.0

Pins



→ set BIT0 in P1OUT  
How?

Bits

Assignment Project Exam Help

bis.b #BIT0, &P1OUT

8-bit

P1SEL0 / P1SEL1 Email: tutorcs@163.com

How not?

Registers

P1SELC

P1DIR

P1OUT P1IN

P1REN

P1IFG

P1IE

P1IES

QQ: 749389476

mov.b #BIT0, &P1OUT

https://tutorcs.com

ALWAYS use bit operations

NEVER use move

# Addressing GPIO Port Registers



~~程序代写代做 CS编程辅导~~

**bic.b #BIT0, &P1OUT**



**byte**

**BIT0, &P1OUT**

**immediate  
mode**

**absolute  
mode**

**(rather than  
symbolic mode)**

WeChat: cstutorcs

All addresses are defined in the header file msp430fr69891.h

**Assignment Project Exam Help**

```
#define P1IN          (PAIN_L)      /* Port 1 Input */
#define P1OUT         (PAOUT_L)     /* Port 1 Output */
#define P1DIR         (PDIR_L)      /* Port 1 Direction */
#define P1REN         (PAREN_L)     /* Port 1 Resistor Enable */
#define P1SEL0        (PASEL0_L)    /* Port 1 Selection 0 */
#define P1SEL1        (PASEL1_L)    /* Port 1 Selection 1 */
#define P1SELC        (PASELC_L)    /* Port 1 Complement Selection */
#define P1IES          (PAIES_L)     /* Port 1 Interrupt Edge Select */
#define P1IE           (PAIE_L)      /* Port 1 Interrupt Enable */
#define P1IFG          (PAIFG_L)     /* Port 1 Interrupt Flag */
```

**QQ: 749389476**

**<https://tutorcs.com>**

Registers are replicated for all 10 ports: P2IN, ..., P3IN, ..., P10IN,...

# Configuring Px.y: PxSEL0/ PxSEL1



程序代写代做 CS编程辅导

## Function Select Registers: PxSEL0, PxSEL1

Pins are multiplexed



66 P1.0/TA0.1/DMAE0/RTCCLK/A0/C0/VREF-/VeREF-  
65 P1.1/TA0.2/TA1CLK/COUT/A1/C1/VREF+/VeREF+

PxSEL0 and PxSEL1 determine the pin function

(PxSELC is a helper register to complement between 00 and 11)

WeChat: cstutorcs

Table 12-2. I/O Function Selection

Assignment Project Exam Help

PxSEL1	PxSEL0	I/O Function
0	0	General purpose I/O is selected
0	1	Primary module function is selected
1	0	Secondary module function is selected
1	1	Tertiary module function is selected

<https://tutorcs.com>

Default values are **PxSEL0.y = 0** and **PxSEL1.y = 0**

⇒ the default function for each pin **Px.y** is **GPIO / Digital I/O**



# Configuring Px.y: PxDIR

程序代写代做 CS编程辅导

## Direction Register: PxDIR

Selects the **direction** of the corresponding I/O pin: i.e., input or output



**PxDIR.y = 0:** Pin Px.y is switched to **input** direction (Default)

**PxDIR.y = 1:** Pin Px.y is switched to **output** direction

WeChat: cstutorcs

Assignment Project Exam Help

## Shorthand notation:

**PxDIR.y** refers to bit  $y \in \{0, 1, \dots, 7\}$  of register controlling port  $x \in \{1, \dots, 10\}$

**Px.y** refers to pin  $y \in \{0, 1, \dots, 7\}$  of port  $x \in \{1, \dots, 10\}$

QQ: 749389476

<https://tutorcs.com>



# Configuring Px.y: PxIN

程序代写代做 CS编程辅导

## Input Register: PxIN

Bit PxIN.y reflects the value of the input signal at pin Px.y



**PxIN.y = 0:** Input at pin Px.y is LOW

**PxIN.y = 1:** Input at pin Px.y is HIGH

WeChat: cstutorcs

**Note: PxIN is a read-only register**

Assignment Project Exam Help

You cannot write to it, attempting to write only results in increased current consumption while the write attempt is active

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

# Configuring Px.y: PxOUT – Role 1



程序代写代做 CS 编程辅导

## Output Register: PxOUT

Bit PxOUT.y is the value of output signal at pin Px.y when the pin is configured as output connection, output direction



PxOUT.y = 0: Output at pin Px.y is LOW

PxOUT.y = 1: Output at pin Px.y is HIGH

## How to write to output?

The red LED is connected to Pin 0.

P1DIR.0 = 1 selects the pin as output

Assignment Project Exam Help

First set the desired output value, then change the direction

Otherwise, the initial output may be random

bis.b  
bis.b

#BIT0, &P1DIR  
#BIT0, &P1OUT

Option 1

bis.b  
bis.b

#BIT0, &P1OUT  
#BIT0, &P1DIR

Option 2

<https://tutorcs.com>