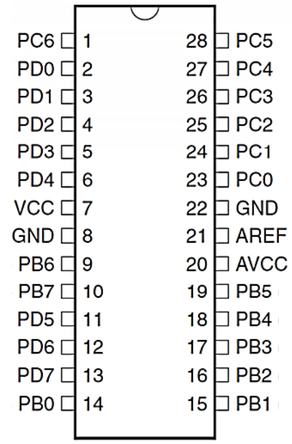
E209
Sistemas Microcontrolados e Microprocessados

Prof. João Magalhães

GPIO



Introdução

Os registradores e referências de bits para cada pino de porta recebe o nome de **Pxn**, a letra "x" representa o nome da porta, como PB (PORTB) ou PD (PORTD), e a letra "n" representa o número do bit, como PB[0-7] (PORTB[0-7]) ou PB[0-7] (PORTB[0-7]).

Vale ressaltar que as portas analógicas são chamadas de PC (PORTC) que passam por um conversor A/D.

Esses nomes estão indicados na pinagem do microcontrolador e são usados na programação.

GPIO

Introdução

Para cada porta digital **três endereços de memória I/O** são alocados, sendo eles:

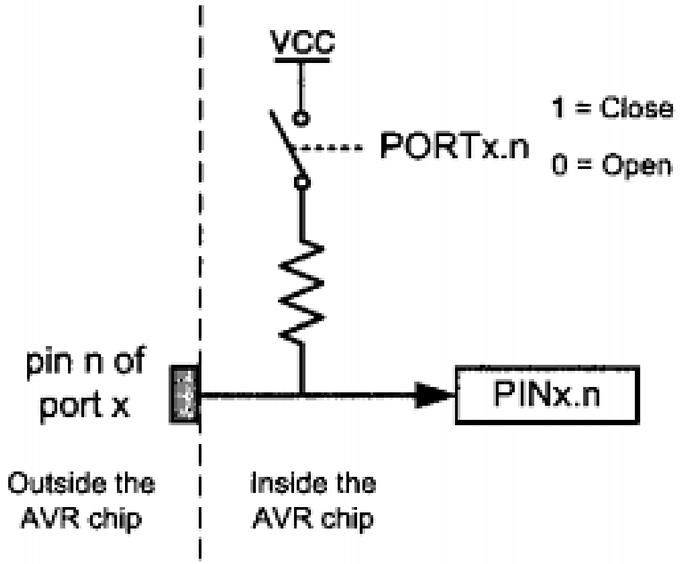
PORTx (leitura/escrita): O registro de dados, responsável em determinar o estado do pino (HIGH/LOW).

DDRx (leitura/escrita): A direção dos dados, responsável pela configuração de entrada ou saída do pino (OUTPUT/INPUT).

PINx (leitura): A entrada da porta, responsável em armazenar o estado do pino, onde a execução de uma função de escrita no PINx, resultará na alteração no valor do PORTx.

GPIO

Registradores



Três bits de registros são setados, sendo eles:

PORTxn: Se PORTxn for setado 1/verdadeiro e o pino estiver configurado como entrada, o resistor de ***pull-up*** interno será ativado. Para desabilitar o resistor pull-up, o PORTxn deve ser escrito como 0/falso ou o pino deve ser configurado como saída.

GPIO

Registradores

DDxn: O bit DDxn no endereço DDRx seleciona a direção desse pino. Se DDxn for escrito como 1/verdadeiro, Pxn será configurado como **saída**. Se DDxn for escrito como 0/falso, Pxn será configurado como **entrada**.

PINxn: Os bits são acessados pelo endereço PINx.

GPIO

Registradores

- **PORTx**: registrador de dados, usado para escrever nos pinos do PORTx.
- **DDRx**: registrador de direção, usado para definir se os pinos do PORTx são entrada ou saída.
- **PINx**: registrador de entrada, usado para ler o conteúdo dos pinos do PORTx.

Bits de controle dos pinos dos PORTs.

DDXn*	PORTXn	I/O	Pull-up	Comentário
0	0	Entrada	Não	Alta impedância (Hi-Z).
0	1	Entrada	Sim	PXn irá fornecer corrente se externamente for colocado em nível lógico 0.
1	0	Saída	Não	Saída em zero (drena corrente).
1	1	Saída	Não	Saída em nível alto (fornece corrente).

Manipulando Registradores

IDE Arduino x Uso de Registradores

```
pinMode(3, OUTPUT);
pinMode(5, OUTPUT);
pinMode(7, OUTPUT);
```

```
pinMode(PIN_D3, OUTPUT);
pinMode(PIN_D5, OUTPUT);
pinMode(PIN_D7, OUTPUT);
```

Manipulação

```
DDRD = 0b10101000;

or

DDRD = 0xA8;

or

DDRD | = 1<<PD7 | 1<<PD5 | 1<<PD3;
```

Manipulando Registradores

IDE Arduino x Uso de Registradores

DDRD = B11111110;

// configura portas 1 ate 7 como saídas e a porta 0 como entrada

DDRD = DDRD | B11111100;

// esta é uma forma mais segura de configurar os pinos 2 até 7 como saída sem mudar as configurações dos pinos 0 e 1 que são da serial

Manipulando Registradores

IDE Arduino x Uso de Registradores

```
pinMode(0, INPUT);
pinMode(1, INPUT);
digitalWrite(0, HIGH);
digitalWrite(1, HIGH);
```

```
pinMode(PIN_D0, INPUT);
pinMode(PIN_D1, INPUT);
digitalWrite(PIN_D0, HIGH);
digitalWrite(PIN_D1, HIGH);
```

```
DDRD = 0; // all PORTD pins inputs
PORTD = 0b00000011;
or
PORTD = 0x03;

or better yet:
  DDRD & = ~(1<<PD1 | 1<<PD0);
  PORTD | = (1<<PD1 | 1<<PD0);
```

Manipulando Registradores

IDE Arduino x Uso de Registradores

** Para alterar o conteúdo dos registradores DDRx e PORTx é necessário realizar uma operação de escrita **de um byte completo**, mesmo que se deseje alterar apenas um dos bits.*

Manipulando Registradores

IDE Arduino x Uso de Registradores

|	OU lógico bit a bit (usado para ativar bits , colocar em 1)
&	E lógico bit a bit (usado para limpar bits, colocar em 0)
^	OU EXCLUSIVO bit a bit (usado para trocar o estado dos bits)
~	complemento de 1 (1 vira 0, 0 vira 1)

Nr >> x	O número é deslocado x bits para a direita
Nr << x	O número é deslocado x bits para a esquerda

Manipulando Registradores

IDE Arduino x Uso de Registradores

Lógica “e” bit a bit (**&**)

```
x:        10001101
y:        01010111
x & y:    00000101
```

Lógica “ou” bit a bit (**|**)

```
x:        10001101
y:        01010111
x | y:    11011111
```

Deslocamento para a esquerda

```
y = 1010
x = y << 1
Resulta
em: x = 0100
```

Operação “não” (**~**)

```
~0 = 1
~1 = 0
```

Operação “ou-exclusivo” (**^**)

```
0 ^ 0 = 0
0 ^ 1 = 1
1 ^ 0 = 1
1 ^ 1 = 0
```

Deslocamento para a direita

```
y = 1010
x = y >> 1
Resulta
em: x = 0101
```

Operações

- **Ativação de bit, colocar em 1:**

```
#define set_bit(Y,bit_x) (Y|=(1<<bit_x))
```

onde `Y |= (1<<bit_x)` ou `Y = Y | (1<<bit_x)`

Exemplo:

set_bit(PORTD,5)

PORTD = PORTD | (1<<5) ,

```
          0bxxxxxxxx     (PORTD, x pode ser 0 ou 1)
        | 0b00100000     (1<<5 é a máscara)
PORTD = 0bxx1xxxxx       (o bit 5 com certeza será 1)
```

Operações

- **Limpeza de bit, colocar em 0:**

```
#define clr_bit(Y,bit_x) (Y&=~(1<<bit_x))
```

onde `Y &= ~(1<<bit_x)` ou `Y = Y & (~ (1<<bit_x))`

Exemplo:

clr_bit(PORTB,2)

PORTB = PORTB & (~ (1<<2)) ,

```
          0bxxxxxxxx     (PORTB, x pode ser 0 ou 1)
        & 0b11111011     (~(1<<2) é a máscara)
PORTB = 0bxxxxx0xx       (o bit 2 com certeza será 0)
```

Operações

- **Troca o estado lógico de um bit, 0 para 1 ou 1 para 0:**

```
#define cpl_bit(Y,bit_x) (Y^=(1<<bit_x))
```

onde `Y ^ = (1<<bit_x)` ou `Y = Y ^ (1<<bit_x)`

Exemplo:

cpl_bit(PORTC,3)

PORTC = PORTC ^ (1<<3) ,

```
        0bxxxx1xxx     (PORTC, x pode ser 0 ou 1)
      ^ 0b00001000     (1<<3 é a máscara)
PORTC = 0bxxxx0xxx     (o bit 3 será 0 se o bit a ser
                        complementado for 1 e 1 se ele for 0)
```

Operações

- **Leitura de um bit:**

```
#define tst_bit(Y,bit_x) (Y&(1<<bit_x))
```

Exemplo:

tst_bit(PIND,4)

PIND & (1<<4) ,

```
            0bxxxTxxxx     (PIND, x pode ser 0 ou 1)
          & 0b00010000     (1<<4 é a máscara)
resultado = 0b000T0000     (o bit 4 terá o valor T, que será 0 ou 1)
```

Exercícios


KEEP
CALM
AND
LET KARMA
FINISH IT
KEEP
CALM
AND
ACCEPT
MURPHYS LAW

Prof. João Magalhães

Horário de Atendimento:

- Segunda-feira: 17h30
- Quinta-feira: 19h30

E-mail: joao.magalhaes@inatel.br

Celular: (35) 99895-4450

Linkedin: https://www.linkedin.com/in/joaomagalhaespaiva/