


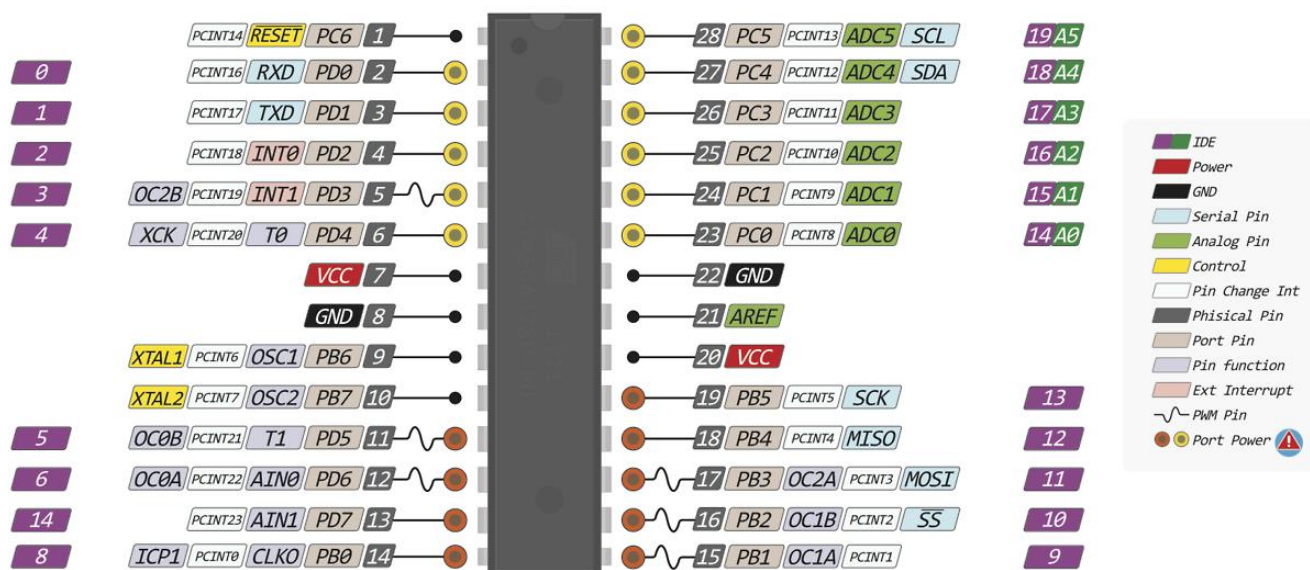
<div> <i>Instituto Nacional de Telecomunicações</i></div>	RELATÓRIO 1 - Teoria		Data:    /    /	
	Disciplina: E209			
	Prof: João Pedro Magalhães de Paula Paiva			
Conteúdo: Microcontroladores AVR				
Tema: GPIO ATmega 328P				
Nome:			Matrícula:	Curso:

## Objetivos

- Apresentar os conceitos da arquitetura do microcontrolador Atmega328P.
- Interpretar as funcionalidades dos registros dos pinos GPIO do microcontrolador.
- Utilizar ferramentas para aplicar os firmwares na prática para resolução dos problemas.
- Aplicar na prática a lógica booleana em conjuntos com os operadores booleanos para filtragem de bits.

## Parte Teórica:

O microcontrolador Atmega328P



O Atmega328P apresenta conjuntos de “portas” ou “portais” identificados por **PB**, **PC** e **PD**. Cada pino do conjunto possui funcionalidades básicas de I/O (**entradas e saída**). Alguns destes pinos possuem funções especiais, as mesmas serão abordadas em relatórios futuros.

## O uso da linguagem C nos microcontroladores:

A linguagem C, inicialmente criada para desenvolvimento de programas de computador, foi aos poucos sendo substituída por outras linguagens que facilitavam o desenvolvimento, mas ganhou uma sobrevida devido ao seu uso nos sistemas embarcados, ou seja, nos microcontroladores.

**A sintaxe da linguagem é a mesma, seja para PC ou para MCU, mas cabe salientar algumas observações:**

- Deve-se prestar atenção nos tipos de variáveis utilizadas (**char, short, int, long**) devido a limitação de espaço de memória de dados (RAM).
- Normalmente, os firmwares não possuem fim. Dessa forma, utiliza-se estruturas de repetição infinita (**loop-infinito**) no programa: **for(;;);** ou **while(1);**.
- Quando os programas são de baixa complexidade e apresentam lógica simples, pode-se utilizar uma execução sequencial, que possibilita a implementação prática da máquina de estados:
  - o realiza a leitura das entradas e armazena em variáveis,
  - o interpreta os valores das variáveis e executa a lógica desejada,
  - o atualiza as saídas (método denominado **super-loop**).
- É boa prática utilizar recursos que facilitam a alteração do uso dos pinos de GPIO/portais. Normalmente isso é feito utilizando a diretiva “**#define**”. Dessa forma, caso um periférico tenha que ser trocado de pino, fica simples adaptar o programa. Exemplo:

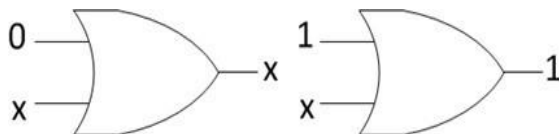
```
#define SENSOR 1<<PB5
```

```
#define BOMBA 1<<PB0
```

## Técnica de mascaramento:

Durante o curso, será muito comum utilizar bits para manipulação dos registros. Porém, a arquitetura do Atmega328P é de 8-bits, ou seja, as variáveis mínimas são de 8-bits (Byte). Para manipular bits, utiliza-se a aritmética binária com a lógica “OU” e “E” da seguinte forma:

- **Lógica OU:** possível fazer com que uma informação **X** seja “1”. Se fizermos a lógica **OU** entre “bit qualquer” e “1”, o resultado sempre será “1”. Se fizermos a lógica **OU** entre “bit qualquer” e “0”, o resultado será o valor do “bit qualquer”.

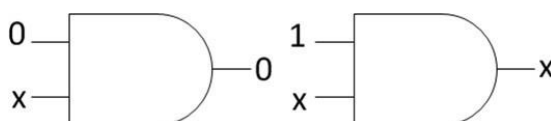


Exemplos:

Escrever “1” no bit 0:  $PORTx = PORTx | (1 \ll Pino0);$

PORTx - bits	7	6	5	4	3	2	1	0
PORTx antes	X	X	X	X	X	X	X	X
Máscara a ser aplicada	0	0	0	0	0	0	0	1
PORTx depois	X	X	X	X	X	X	X	1

- **Lógica E:** possível fazer com que uma informação **X** seja “0” ou mascarar(filtrar) uma informação **X** desejada para ser lida. Se fizermos a lógica **E** entre “bit qualquer” e “0”, o resultado sempre será “0”. Se fizermos a lógica **E** entre “bit qualquer” e o valor “1”, o resultado será o valor do “bit qualquer”.



Exemplos:

Escrever “0” no bit 0:  $PORTx = PORTx \& \sim(1 \ll Pino0);$

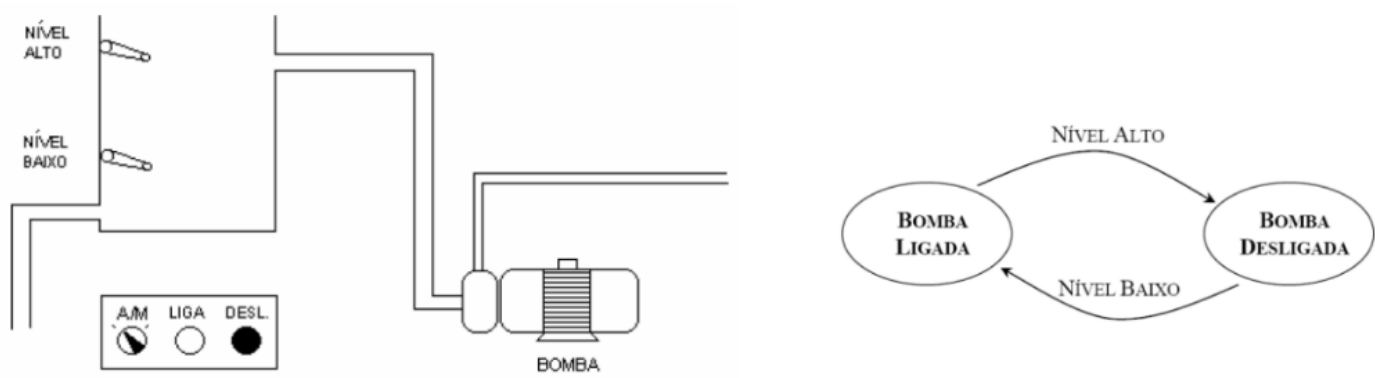
PORTx - bits	7	6	5	4	3	2	1	0
PORTx antes	X	X	X	X	X	X	X	X
Máscara a ser aplicada	1	1	1	1	1	1	1	0
PORTx depois	X	X	X	X	X	X	X	0

Ler a informação contida no bit 3:  $var = PINx \& (1 \ll Pino3);$

PINx - bits	7	6	5	4	3	2	1	0
PINx antes	X	X	X	X	X	X	X	X
Máscara a ser aplicada	0	0	0	0	1	0	0	0
VAR	0	0	0	0	X	0	0	0

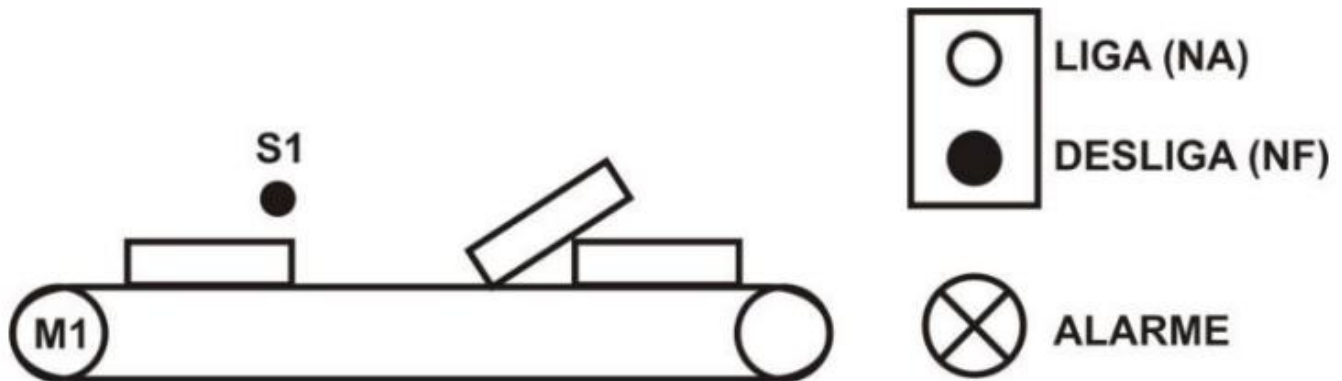
Parte prática:

1. (Fácil) Elabore um firmware para controlar o nível de um tanque, conforme a figura abaixo. O operador poderá selecionar o modo Automático ou Manual, seguindo os seguintes critérios:
- a. Em modo Manual, a bomba poderá ser ligada pressionando o botão LIGA (NA - VCC) e desligada pressionando o botão Desliga (NF - GND). Neste modo, os sensores de nível não tem nenhuma ação.
  - b. Em Automático, a bomba deve ser ligada sempre que o sensor de Nível Baixo estiver desacionado e desligada sempre que o sensor de Nível Alto estiver acionado.
  - c. Não é necessário prever uma condição de defeito.



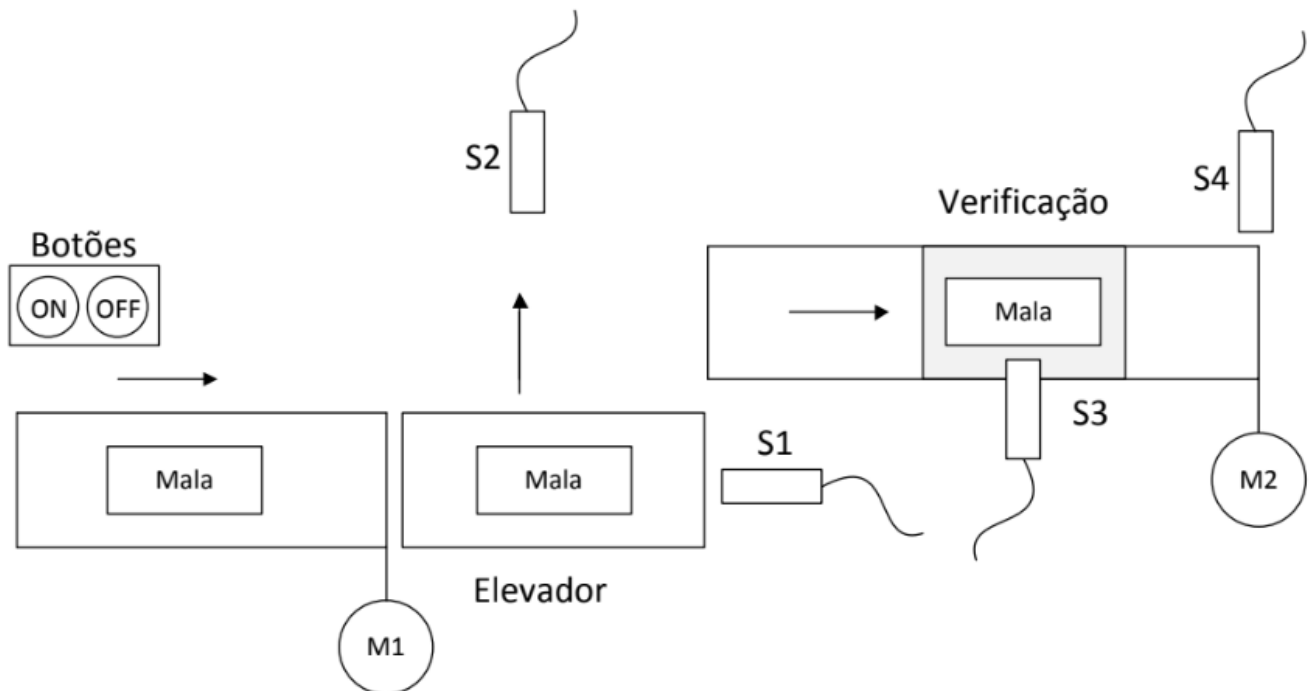
Entradas / Saídas	GPIO Utilizado
Chave Seletora Automático / Manual	PD3
Botão Liga	PC0
Botão Desliga	PC1
Sensor de Nível Baixo	PD4
Sensor de Nível Alto	PD5
Bomba	PB0

2. (Médio) Em uma esteira de transporte, foi instalado um sistema de verificação de peças posicionadas de forma errada. Elabore um firmware para controlar o sistema, seguindo os passos apresentados abaixo.
- Ao pressionar o botão LIGA (NA) a esteira entra em movimento (MOTOR = TRUE);
  - Ao pressionar o botão DESLIGA (NF) a esteira para seu movimento (MOTOR = LOW);
  - Caso aconteça o amontoamento de peças (S1 = HIGH), a esteira deverá parar imediatamente e alarme deverá ser ligado (ALARME = HIGH);
  - Enquanto as peças estiverem amontoadas, a esteira não poderá ser ligada;
  - Para desligar o ALARME, as peças deverão estar desamontoadas (S1 = LOW) e o botão LIGA deve ser pressionado.



Entradas / Saídas	GPIO Utilizado
Motor 1	PD2
Sensor 1	PB3
Botão Liga	PB0
Botão Desliga	PB1
Alarme	PD6

3. (Difícil) Em um sistema de controle de bagagens, a separação e movimentação das bagagens ocorre automaticamente de acordo com o destino do passageiro. Por motivos de segurança, algumas malas devem passar pelo Raio X localizado em um local separado das esteiras de transporte. Elaborar um firmware para controlar o sistema de transporte abaixo.
- Para iniciar o transporte da mala até o local de verificação, o operador deve pressionar o botão LIGA para o Motor 1 (M1) transportar a mala até o elevador.
  - Assim que a mala aciona o Sensor 1, o Motor 1 é desligado e após 3 segundos o Motor do Elevador é ligado.
  - Ao chegar no andar superior, a mala aciona o Sensor 2, desligando o Motor do elevador e acionando o Motor 2 (M2) para transportar a mala até o local de verificação.
  - Assim que a mala acionar o Sensor 3, o Motor 2 é desligado por 3 segundos e ligado novamente para transportar a mala até o carrinho.
  - Ao acionar o Sensor 4, o processo é finalizado e o sistema todo é desligado.
  - Enquanto a bagagem estiver no Raio X, uma lâmpada indicativa deverá ficar piscando com frequência de 2 Hz.
  - Sempre que o botão DESLIGA for pressionado, todo o sistema deve ser desligado.



Entradas / Saídas	GPIO Utilizado
Motor 1	PD0
Motor 2	PD1
Motor Elevador	PD2
Sensor 1	PB0
Sensor 2	PB1
Sensor 3	PB2
Sensor 4	PB3
Botão Liga	PC0
Botão Desliga	PC1
Lâmpada	PD3