





Sistemas Embarcados

O que são os sistema embarcados e tecnologias habilitadoras.



- Graduado em Engenharia Elétrica pela Universidade Federal do Amazonas - UFAM (2006). Atua em pesquisa e desenvolvimento na área de eletrônica e automação industrial. Possui vasta experiência em desenvolvimento de hardware, mais especificamente, placas de circuito impresso, dentre elas: placas de instrumentação analógica e digital utilizadas em sistemas de testes, placas processadoras baseadas em microcontroladores de diversos fabricantes, placas utilizadas em sistemas de comunicação digital com tecnologias cabeadas como RS485, RS422, Ethernet e tecnologias sem fio como Wi-fi 802.11, Zigbee 802.15 dentre outras. Possui experiência em desenvolvimento de firmware em diversas plataformas dentre elas: Intel 8088, 8051, Atmel, Microchip, ESP8266, Arduino.
- <http://lattes.cnpq.br/5067803336101638>

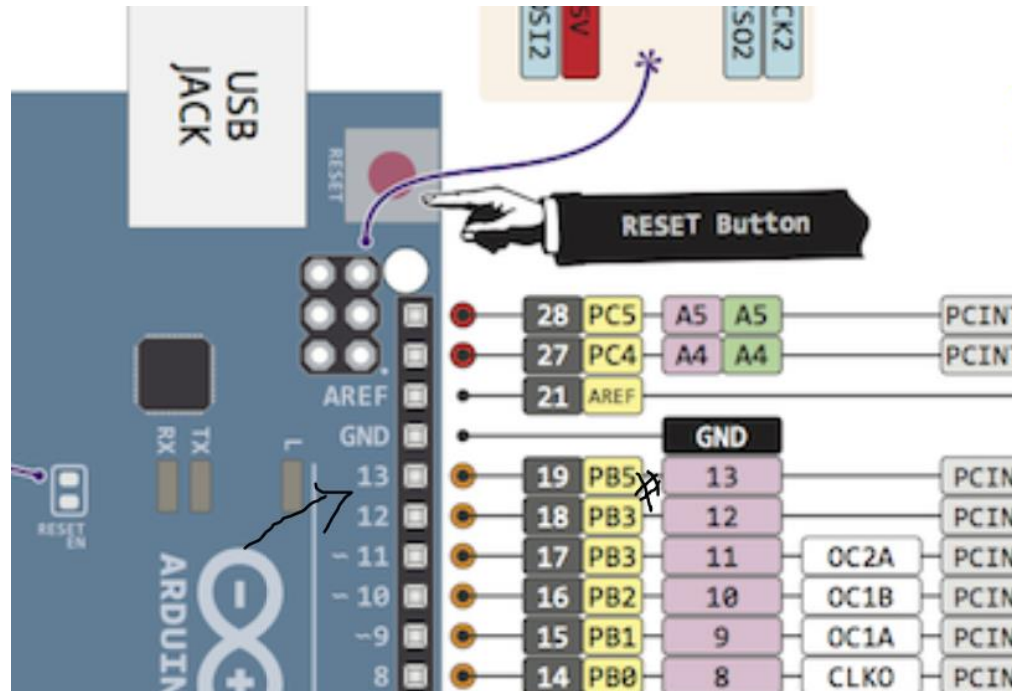


- Criação do primeiro programa (Requisitos);
- Como funcionam as bibliotecas;
- Estudo do **#define**;
- Estudos das variáveis;
- Estudo de tipos;
- Criação de funções
- Estudo da condicional **IF**

Criação do primeiro programa (Requisitos)



- Como primeiro programa criaremos um driver ou biblioteca para escrita nas portas digitais do dispositivo.
- Através de análise visual verificamos que a serigrafia da placa não condiz com a nomenclatura interna do chip;
- Pino D13 = PB5





Criação do primeiro programa (Requisitos)



- Como primeiro programa criaremos uma biblioteca que funcione como uma camada de tradução dos valores que estão na serigrafia para os valores correspondentes do chip.
- Essa biblioteca deverá possuir as seguintes funções:

*/*Configuração de direção do pino*/*

pinMode(PORT, PIN);

*/*Função para escrita no pino*/*

digitalWrite(PORT, PIN);

*/*Função para leitura do pino*/*

digitalRead(PORT, PIN);

Criação do primeiro programa (Requisitos)



- Inicialmente criaremos as funções diretamente no código principal e posteriormente às transportaremos para um novo arquivo que será a biblioteca.

Como funcionam as bibliotecas



- Uma biblioteca ou driver é um conjunto de funções que são disponibilizadas junto com os dispositivos e são utilizada para agilizar o desenvolvimento através da pré-configuração dos dispositivos internos da placa.

```
1  /*****
2  Software criado para estudo da linguagem C
3  Autor: Washington Lisboa
4  *****/
5
6  #define F_CPU 16000000UL
7
8  #include <avr/io.h>
9  #include <util/delay.h>
10 #include "io_ctr.h"
11
12 int main()
```



define NOME VALOR

```
/* *****  
Definições dos pinos digitais  
***** */  
  
#define D0      0  // Port D  
#define D1      1  
#define D2      2  
#define D3      3  
#define D4      4  
#define D5      5  
#define D6      6  
#define D7      7  
#define D8      8  // Porta B  
#define D9      9  
#define D10     10  
#define D11     11  
#define D12     12  
#define D13     13
```



Tipo	Tamanho em Bytes	Faixa Mínima
char	1	-127 a 127
unsigned char	1	0 a 255
signed char	1	-127 a 127
int	4	-2.147.483.648 a 2.147.483.647
unsigned int	4	0 a 4.294.967.295
signed int	4	-2.147.483.648 a 2.147.483.647
short int	2	-32.768 a 32.767
unsigned short int	2	0 a 65.535
signed short int	2	-32.768 a 32.767
long int	4	-2.147.483.648 a 2.147.483.647
signed long int	4	-2.147.483.648 a 2.147.483.647
unsigned long int	4	0 a 4.294.967.295
float	4	Seis dígitos de precisão
double	8	Dez dígitos de precisão
long double	10	Dez dígitos de precisão



- Contexto Global

```
11  
12 | int intMinhaVariavel; // Contexto Global  
13  
14 | int main(){  
15 |     mapPorts();  
16 |     pinMode(D13,OUTPUT);  
17 |     while(1) {
```

- Contexto Local

```
12  
13 | int main(){  
14 |     int intMinhaVariavel; // Contexto Local  
15 |     mapPorts();  
16 |     pinMode(D13,OUTPUT);  
17 |     while(1) {  
18 |         digitalWrite(D13,TRUE);  
19 |         delay_ms(1000);
```



```
tipo nome(parâmetros){  
}
```

```
50  
51  /**  
52  void pinMode(int pin, int dir)  
53  Descrição: Configura os pinos do chip como entrada  
54  ou como saída  
55  ***/  
56  void pinMode(int pin, int dir){  
57      //Configuração do D0  
58      if (pin == D0){  
59          if (dir == INPUT){
```

Condicional IF, ELSE



- A condicional IF serve para direcionar o ponto de execução do programa de acordo com determinadas condições

```
//Configuração do D0
if (pin == D0){
    if (dir == INPUT){
        //Configura como entrada
        DDRD = DDRD & 0b11111110;
    }
    else{
        //Configura como saída
        DDRD = DDRD | 0b00000001;
    }
}
```

