

EA075 – Segunda Avaliação

Profa. Ting

14/11/2019

RA: _____ Nome: _____

Questão:	1	2	3	4	5	6	7	Total
Pontos:	5	6	4	2	2	2	8	29
Resultado:								

A nota final seria a soma dos pontos dividida por 2,3.

Foi projetada uma estação meteorológica com as seguintes funcionalidades:

- comunicação Bluetooth,
- tensão de operação 3.3V,
- armazenamento local das amostras,
- monitoramento de temperatura, umidade e pressão atmosférica, e
- controle programável de potência a ser entregue a um circuito de acionamento.

Os componentes selecionados para a sua implementação foram

- 2 módulos de BME280,
- 1 módulo Bluetooth HC-06,
- 1 optoacoplador SFH18A-2,
- 1 cristal de quartzo ECS-160-20-1,
- 2 módulos de memória 24C16 EEPROM (16K *bits*), e
- 1 microcontrolador ATmega328 com o núcleo AVR.

A pinagem de cada componente se encontra detalhada no Apêndice A.

1. Barramentos: O módulo HC-06 é conectado com o módulo de interface USART (porta D) do microcontrolador ATmega328. Os dois módulos de memória EEPROM são ligados em série com a interface TWI (I2C) (porta C). O primeiro módulo tem os seus pinos A0, A1 e A2 aterrados (nível lógico 0) e o segundo tem os seus pinos A1 e A2 aterrados e A0 conectados em 3.3V (nível lógico 1), associando os endereços 000 e 001, respectivamente, aos dois módulos no barramento I2C. Os dois módulos BME280 são conectados com a interface SPI (porta B) do microcontrolador.
 - (a) (2pt) Dados os valores máximos dos parâmetros permitidos a um barramento I2C operando a uma corrente $I_{OL}=3\text{mA}$:

- tempo de subida dos sinais SDA e SCL: $t_r = 0.8473 R_p C_b \leq 1000\text{ns}$;
- carga capacitiva de cada linha do barramento: $C_b \leq 400\text{pF}$;
- tensão de saída no nível lógico 0 dos dispositivos conectados: $V_{OL} \leq 0.4\text{V}$.

Estime a faixa de resistências do resistor *pull-up* R_p do barramento I2C, sabendo que ele está conectado a $V_{DD} = 3.3\text{V}$.

- (b) (1pt) Um barramento *multi-drop* é um barramento em que um mestre comunica com vários escravos univocamente identificáveis, enquanto um barramento *multi-point* é um barramento em que há mais de um mestre conectado, requerendo um mecanismo de arbitragem. É necessário integrar ao barramento I2C um mecanismo de arbitragem? Justifique.

- (c) (1pt) O protocolo USART (*Universal Synchronous/Asynchronous Receiver Transmitter*) suporta tanto a comunicação serial síncrona como a assíncrona. Ele especifica dois fios de comunicação. Sabendo que numa comunicação serial os sinais transmitidos podem incluir, além dos sinais de dados, os pulsos de relógio, quais tipos de sinais são transmitidos nos dois fios quando se conecta o módulo HC-06 à interface USART do microcontrolador ATmega328? Justifique.

- (d) (1pt) Como o protocolo USART, a topologia de comunicação do protocolo SPI é ponto-a-ponto. Como é estabelecida esta comunicação entre o microcontrolador (mestre) e um dos dois módulos BME280 (escravos) no projeto da estação

meteorológica?

2. Microcontroladores: Estão integrados no microcontrolador ATmega328 3 módulos de memória, três temporizadores TCx, um temporizador *watchdog* e três portas de 8 pinos para os módulos se comunicarem com o mundo físico externo.

(a) (1pt) Pelos mapas de endereços mostrados no Apêndice B e pelo diagrama de blocos apresentado na Figura 6, qual é a arquitetura do microcontrolador ATmega328? Von Neumann (Princeton), Harvard, ou Harvard modificada? Justifique.

(b) (2pt) É integrado à estação meteorológica um circuito de controle de potência a ser entregue a uma carga, como um climatizador de ar. A técnica aplicada é baseada em PWM (*Pulse Width Modulation*). No Apêndice C encontram-se as informações acerca a configuração dos registradores do temporizador TC2 do microcontrolador ATmega328 para que os seus canais gerem sinais PWM.

Usando os operadores lógicos *bit* a *bit*, $\&(AND)$, $|(OR)$ e $\sim(NOT)$, da linguagem C, configure o registrador TCCR2A para que o canal TC2B envie ao pino OC2B um sinal PWM não invertido (o nível do sinal é setado em 1 quando o contador TCNT2 do módulo TC2 atinge o valor máximo de contagem).

(c) (1pt) O contador TCNT2 do módulo TC2 é um contador de 8 *bits*. Configurando o canal TC2B para operar no modo *FastPWM* e não-invertido, o sinal passa do nível 1 para o nível 0 quando a contagem iguala o valor setado no registrador OCR2B. Determine o valor a ser setado no registrador OCR2B para que seja produzido um

sinal PWM com 25% *duty cycle* no pino 5 do microcontrolador (OC2B)?

- (d) (2pt) A estação meteorológica projetada será embutida dentro de um objeto bem vedado, de difícil acesso. Qual dos módulos da Figura 6 você ativaria para detectar as operações anormais e, se possível, fazer algumas tomadas de decisão simples automáticas? Justifique, explicando sucintamente o funcionamento do módulo que você ativaria.

3. Programação Concorrente: Foi programado o TC1 para que ele interrompa periodicamente o processador quando ocorre estouro (*overflow*) no contador TCNT1. O fluxo é automaticamente desviado para a rotina de serviço XXXXX que faz a leitura dos dados brutos de temperatura, umidade e pressão capturados pelos módulos BME280. Seguem-se alguns trechos de códigos implementados. Foram usadas as funções de conversão de dados brutos para os processados mostradas no Apêndice E.

```
void armazene_amostras_EEPROM (uint8_t, uint8_t *);
void exhibe_dados_processados (uint32_t, uint32_t, uint32_t);
void inicializacoes_MCU ();

uint32_t adc_T, adc_H, adc_P;
uint32_t temp_em_DegC, umidade_em_Perc, pressao_em_Pa;
uint32_t *vetor_amostras;
uint8_t flag=0;

void XXXXX () {
    le_bme280_dados_brutos (&adc_T, &adc_H, adc_P);

    temp_em_DegC = (uint32_t)(BME280_compensate_T_int32((BME280_S32_t)adc_T));
    umidade_em_Perc = (uint32_t)(bme280_compensate_H_int32((BME280_S32_t)adc_H));
    pressao_em_Pa = (uint32_t)(BME280_compensate_P_int64((BME280_S32_t)adc_P));

    flag = 1;
}

void setup () {
    vetor_amostras = (uint32_t *)malloc(sizeof(uint32_t)*3);
    inicializacoes_MCU ();
}

void loop () {
    if (flag) {
```

```

vetor_amostras[0] = adc_T;
vetor_amostras[1] = adc_H;
vetor_amostras[2] = adc_P;

armazene_amostras_EEPROM (12, (uint8_t *)vetor_amostras);

exibe_dados_processados(temp_em_DegC, umidade_em_Perc, pressao_em_Pa);

    flag = 0;
}
}

main () {
    setup ();

    for(;;) {
        loop ();
    }
    free((char *)vetor_amostras);
}

```

- (a) (1pt) Na implementação do *software* da estação meteorológica, aplicou as inicializações da tabela de vetores de exceção apresentadas no Apêndice D. Qual deve ser o nome simbólico da rotina de serviço XXXXX? Justifique, indicando o número do vetor de exceção que geraria as interrupções.

- (b) (3pt) Há uma região crítica no programa apresentado. Destaque-a e justifique o seu destaque. Dê uma estratégia para evitar acessos indevidos a recursos compartilhados enquanto se processam as instruções de uma região crítica.

4. Estrutura de Dados: Todas as variáveis declaradas num programa tem os seus espaços de memória alocados antes da execução do programa. As variáveis com inicialização são alocadas no segmento conhecido por “.data” e as sem inicialização, no segmento denominado BSS (*Block Started by Symbol*). Existem ainda dois espaços, pilha e *heap*, que tem seus tamanhos variados durante o tempo de execução. Um esboço desta divisão é mostrado no Apêndice F.

(a) (2pt) Em quais sub-espacos serão alocados os endereços para as variáveis “adc_T”, “pressao_em_Pa”, “vetor_amostras”, “flag” do programa apresentado na questão 3 e as variáveis “var1” e “P” da subrotina chamada BME280_compensate_P_int64, cujo código é apresentado no Apêndice E?

5. Tecnologia de Rede: Existe uma grande variedade de protocolos de comunicação para atender demandas das mais diversas que se possa imaginar. Com ou sem fio. Em redes de área pessoal (PAN), em redes de área pessoal sem fio (WPAN), em redes locais (LAN) ou em redes de área ampla (WAN). O modelo OSI, mostrado no Apêndice G, nos permite ter um entendimento melhor desses protocolos e as relações entre eles. Da mesma forma como USB é o protocolo de comunicação PAN mais encontrado, Bluetooth é o protocolo de comunicação WPAN mais popular. Bluetooth tem baixo consumo de energia, baixas taxas de transmissão (até 3Mbps) e pequeno alcance (até 10m). Figura 11 mostra que ele pode ser aproximadamente mapeado às camadas física e de enlace de dados do modelo OSI. Ele foi aplicado no projeto da estação meteorológica para comunicações da estação com outros dispositivos Bluetooth nas proximidades, como um *smartphone* e um *laptop*.

(a) (1pt) É possível associar ao adaptador HC-06 usado no projeto da estação meteorológica um endereço IP (*Internet Address*)? Justifique.

(b) (1pt) Sensores, conectividade e repositórios de dados são componentes básicos de um sistema IoT (*Internet of Things*). Visando a reduzir o consumo de energia, foi introduzido em 2010 a quarta versão de Bluetooth, BLE (*Bluetooth Low Energy*) ou Bluetooth *smart*. Com esta nova versão de Bluetooth, um dispositivo alimentado somente por uma bateria de célula tipo moeda pode operar por meses ou até anos. Qual(is) camada(s) de protocolo do modelo OSI deve(m) ser agregado(s) ao BLE

para que seus dispositivos participem plenamente da IoT?

6. Circuitos de Alimentação e de Relógio

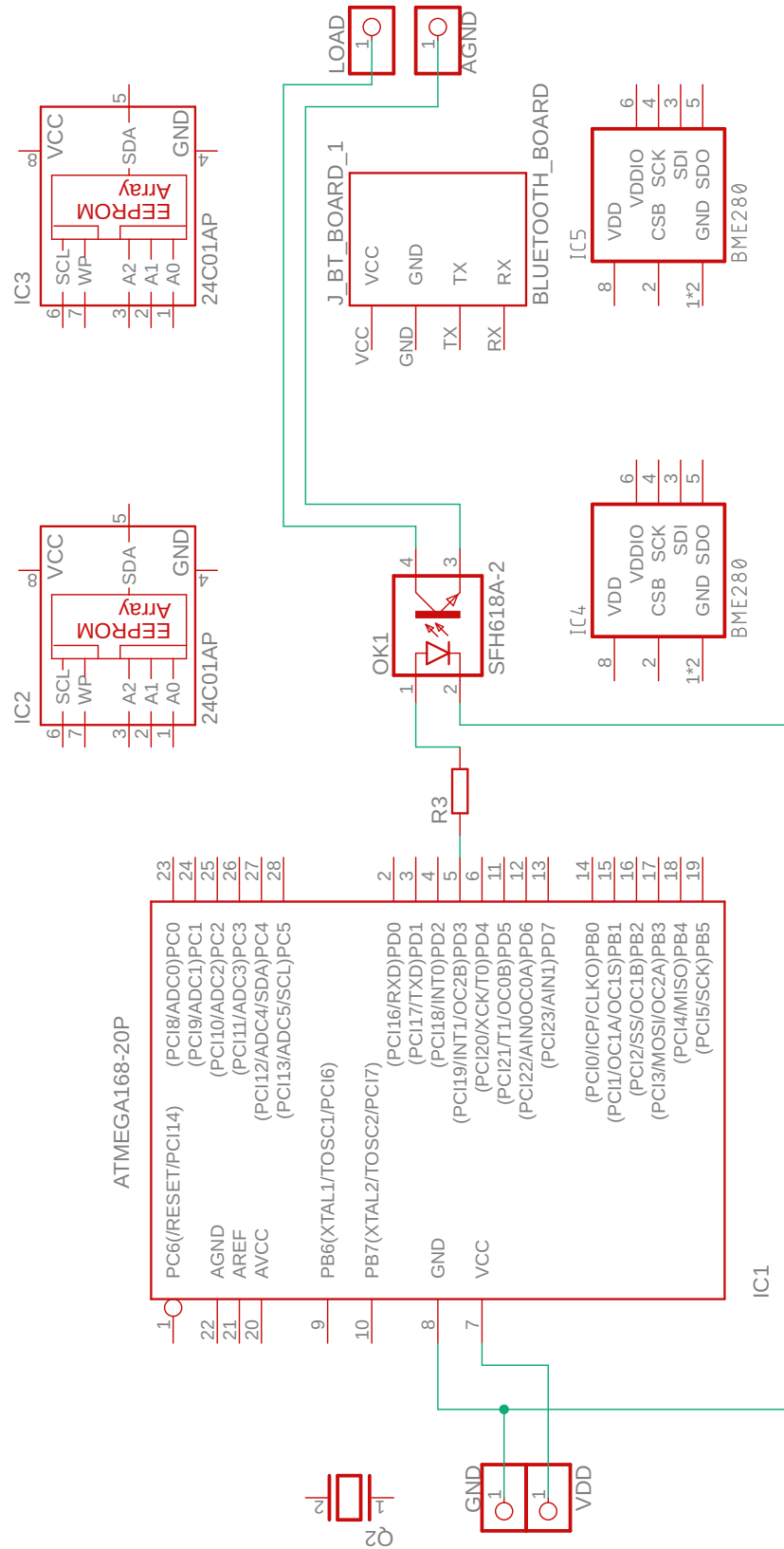
(a) (1pt) Cite uma prática, por *software*, para a redução de consumo de energia da estação meteorológica projetada.

(b) (1pt) Na Figura 6, o módulo “Clock generation” é conectado com um cristal de quartzo e dois capacitores de carga paralelos. Qual é a função primária do cristal e dos capacitores de carga?

7. Esquemático: Síntese dos conceitos e das ideias do projeto.

(a) (8pt) Conecte os pinos dos periféricos com os pinos do microcontrolador, levando em conta que:

1. as memórias EEPROM tem interface I2C.
2. o conversor HC-06 tem interface UART RS232.
3. os valores dos resistores *pull-up* foram calculados.
4. os endereços A0A1A2 de cada módulo de memória são especificados no enunciado desta questão.
5. os dois pontos de alimentação no módulo BME280, conforme mostra a Figura 5, podem ser VDD (3.3V).



Apêndice A: Componentes do projeto de estação meteorológica

1. 1 módulo Bluetooth HC-06, que converte sinais UART RS232 em sinais do protocolo Bluetooth,

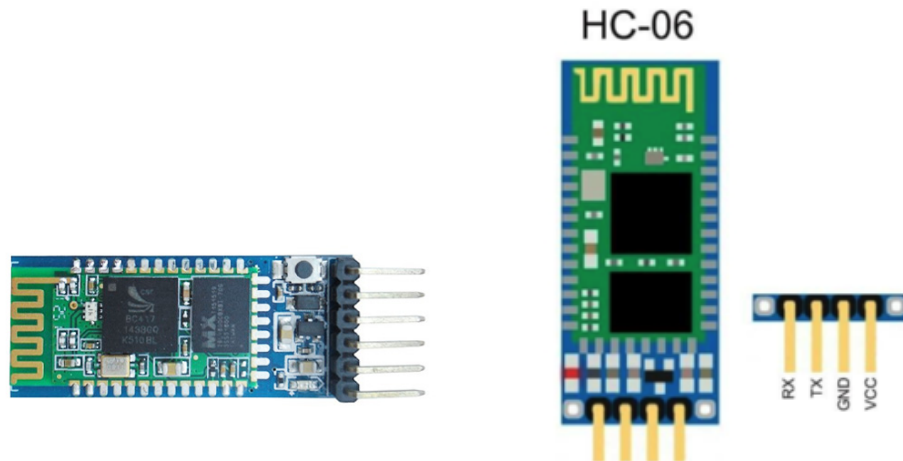


Figura 1: Módulo Bluetooth HC-06.

2. 2 módulos de memória 24C16 EEPROM (16K *bits*) com a interface de comunicação I2C (Nota: WP (*Write Protection* à escrita, ativo alto)),

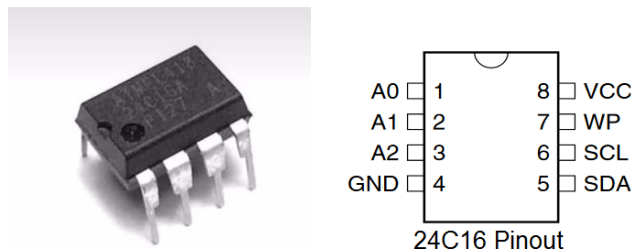


Figura 2: Memória EEPROM 24C16.

3. 2 módulos BME280, com 1 sensor de temperatura, 1 sensor de umidade e 1 sensor de pressão barométrica integrados integrados, providos de interfaces de comunicação SPI e I2C (Nota: SCK (*Serial Clock*), SDI (*Serial Date Input*), SDO (*Serial Data Output*) e CSB (*Chip Select*, ativo baixo)).

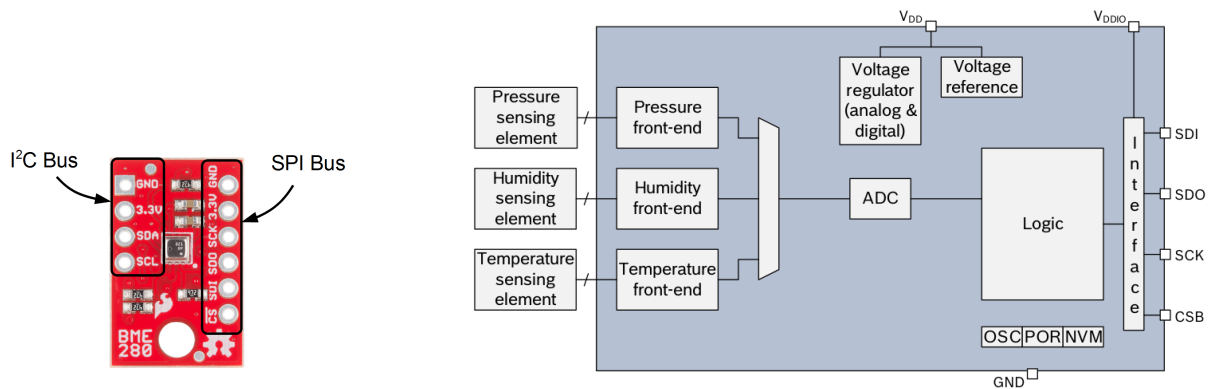


Figura 3: Módulo BME280.

- 1 optoacoplador SFH18A-2 para acoplar opticamente o circuito de controle microcontrolado com o circuito de acionamento de um climatizador de ar, a fim de proteger o microcontrolador dos sinais de cargas ruidosas.

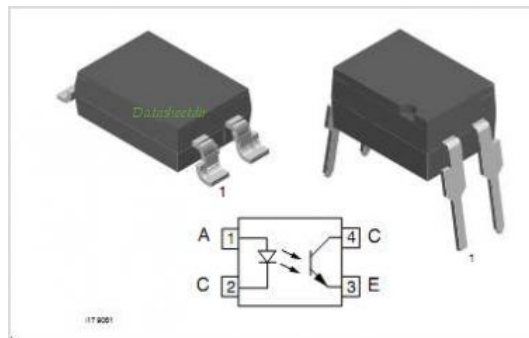



Figura 4: Optoacoplador SFH18A-2

- 1 cristal de quartzo de 16MHz.



PART NUMBERING GUIDE: Example ECS-200-20-1X

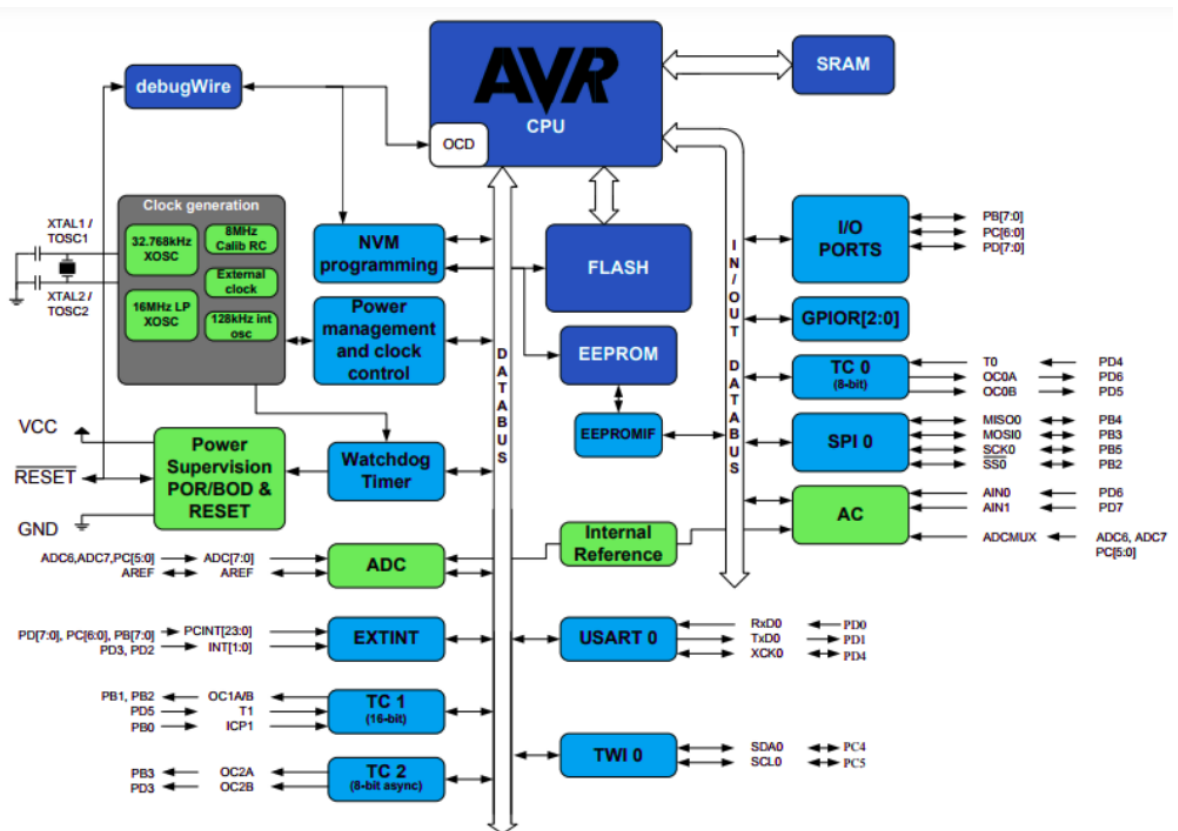
ECS - FREQUENCY ABBREVIATION	LOAD CAPACITANCE	PACKAGE	AVAILABLE OPTIONS			PACKAGING	
			Tolerance	Stability	Temp Range		
ECS	200 = 20.000 MHz See P/N Guide	20 = 20 pF S = Series	1X = HC-49UX	Blank = Std A = ± 25 ppm J = ± 20 ppm R = ± 15 ppm C = ± 10 ppm	Blank = Std D = ± 100 ppm E = ± 50 ppm G = ± 30 ppm H = ± 25 ppm T = ± 20 ppm † W = ± 15 ppm † K = ± 10 ppm †	Blank = Std L = -10 ~ +70°C M = -20 ~ +70°C Y = -30 ~ +85°C N = -40 ~ +85°C P = -40 ~ +105°C S = -40 ~ +125°C U = -55 ~ +125°C	Bulk

* Specify available options in P/N.
† Contact ECS for availability over extended temp range.

Rev.2017

Figura 5: Cristal ECS-160-20-1.

- 1 microcontrolador ATmega328 com o núcleo AVR (*Advanced Virtual RISC*) de 8 bits (Nota: NVM (*Non-Volatile Memory*)=memória não-volátil; TWI (*Two Wire Interface*)=I²C; TC = Temporizador Contador; EXTINT = *External Interrupt*; PCINT = *Pin Change Interrupt*)



Atmega328

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

Figura 6: Microcontrolador ATmega328.

Apêndice B: ATmega328: Mapas de Memória

Estão integrados no microcontrolador ATmega328 3 módulos de memória, sendo o módulo de memória *Flash* tipicamente usado para armazenar as instruções e os módulos EEPROM e SRAM para armazenar os dados.

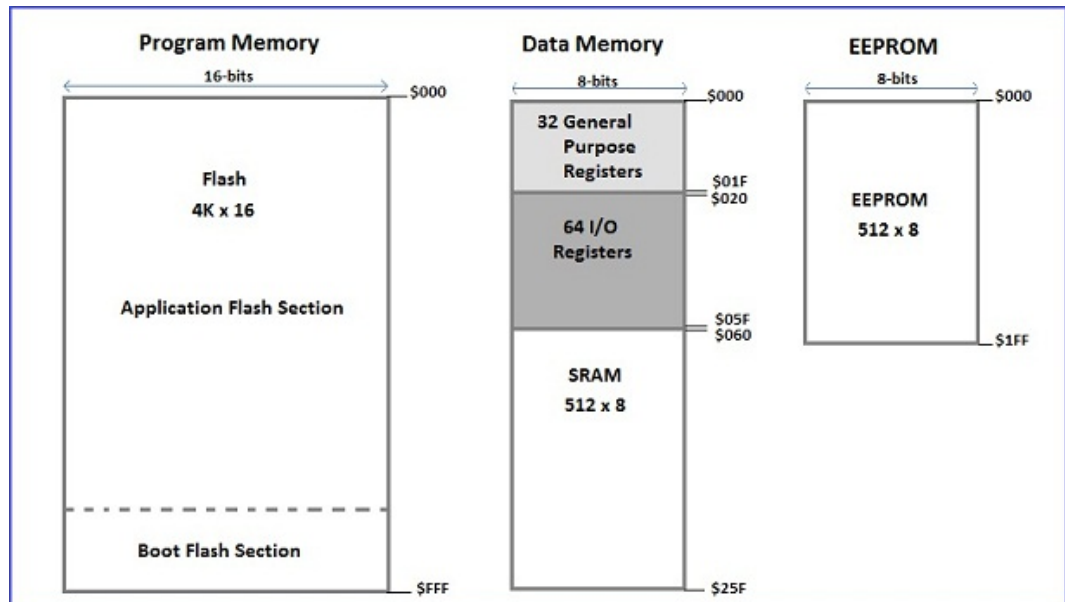


Figura 7: Mapas de memória do microcontrolador ATmega328

Apêndice C: ATmega328: Configuração das Funções dos Pinos

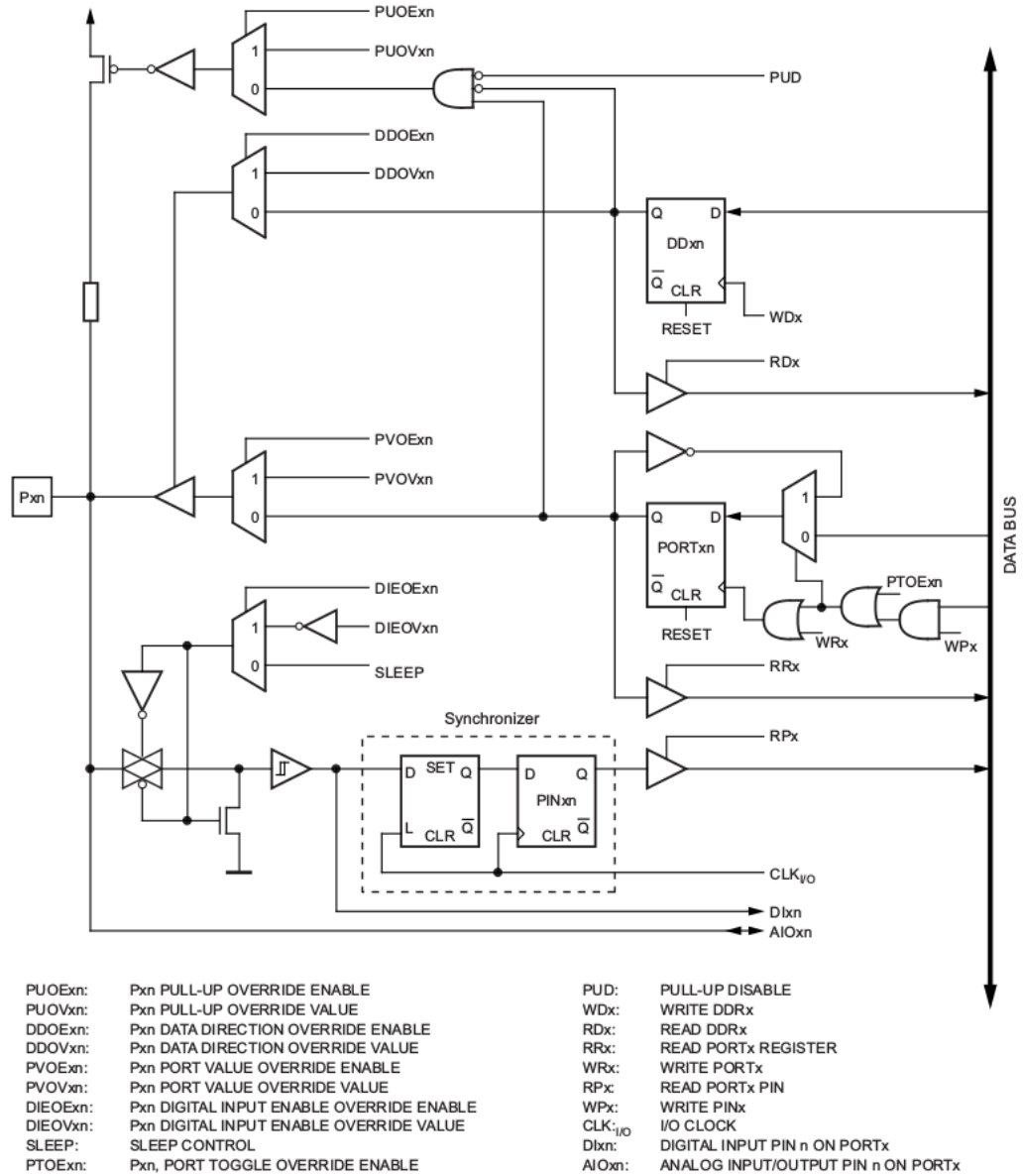
Todos os pinos das portas B, C e D do microcontrolador ATmega328 podem assumir a função GPIO (*General Purpose Input/Output*). Porém, eles podem ser configurados para servir sinais de outros módulos como ilustra a tabela abaixo. Veja as funções alternativas para os pinos da porta D.

Table 13-9. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD7	AIN1 (analog comparator negative input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (analog comparator positive input) OC0A (Timer/Counter0 output compare match A output) PCINT22 (pin change interrupt 22)
PD5	T1 (Timer/Counter 1 external counter input) OC0B (Timer/Counter0 output compare match B output) PCINT21 (pin change interrupt 21)
PD4	XCK (USART external clock input/output) T0 (Timer/Counter 0 external counter input) PCINT20 (pin change interrupt 20)
PD3	INT1 (external interrupt 1 input) OC2B (Timer/Counter2 output compare match B output) PCINT19 (pin change interrupt 19)
PD2	INT0 (external interrupt 0 input) PCINT18 (pin change interrupt 18)
PD1	TXD (USART output pin) PCINT17 (pin change interrupt 17)
PD0	RXD (USART input pin) PCINT16 (pin change interrupt 16)

No microcontrolador ATmega328 estas configurações são feitas através de uma combinação de sinais de controle como mostra o seguinte diagrama de blocos:

Figure 13-5. Alternate Port Functions⁽¹⁾



Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. clk_{I/O}, SLEEP, and PUD are common to all ports. All other signals are unique for each pin.

No projeto de estação meteorológica, o pino 3 da porta D foi selecionado para servir o canal TC2B na transferência dos sinais PWM gerados no modo *FastPWM*. É preciso, portanto, configurar o pino para esta nova função conforme a especificação:

Table 13-11. Overriding Signals for Alternate Functions in PD3..PD0

Signal Name	PD3/OC2B/INT1/ PCINT19	PD2/INT0/ PCINT18	PD1/TXD/ PCINT17	PD0/RXD/ PCINT16
PUE	0	0	TXEN	RXEN
PUO	0	0	0	PORTD0 × $\overline{\text{PUD}}$
DDOE	0	0	TXEN	RXEN
DDOV	0	0	1	0
PVOE	OC2B ENABLE	0	TXEN	0
PVOV	OC2B	0	TXD	0
DIEOE	INT1 ENABLE + PCINT19 × PCIE2	INT0 ENABLE + PCINT18 × PCIE1	PCINT17 × PCIE2	PCINT16 × PCIE2
DIEOV	1	1	1	1
DI	PCINT19 INPUT INT1 INPUT	PCINT18 INPUT INT0 INPUT	PCINT17 INPUT	PCINT16 INPUT RXD
AIO	–	–	–	–

habilitando o canal TC2B e configurando o seu modo de operação através dos registradores de controle TCCR2A e TCCR2B

TCCR2A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
(0xB0)	COM2A1	COM2A0	COM2B1	COM2B0	–	–	WGM21	WGM20	TCCR2A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TCCR2B – Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
(0xB1)	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	TCCR2B
Read/Write	W	W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

de acordo com as especificações do fabricante:

Table 17-6. Compare Output Mode, Fast PWM Mode⁽¹⁾

COM2B1	COM2B0	Description
0	0	Normal port operation, OC2B disconnected.
0	1	Reserved
1	0	Clear OC2B on compare match, set OC2B at BOTTOM, (non-inverting mode).
1	1	Set OC2B on compare match, clear OC2B at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR2B equals TOP and COM2B1 is set. In this case, the compare match is ignored, but the set or clear is done at BOTTOM. See [Section 17.7.4 “Phase Correct PWM Mode” on page 123](#) for more details.

- **Bits 1:0 – WGM21:0: Waveform Generation Mode**

Combined with the WGM22 bit found in the TCCR2B register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see [Table 17-8](#). Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), Clear timer on compare match (CTC) mode, and two types of pulse width modulation (PWM) modes (see [Section 17.7 "Modes of Operation" on page 120](#)).

Table 17-8. Waveform Generation Mode Bit Description

Mode	WGM2	WGM1	WGM0	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on ⁽¹⁾⁽²⁾
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, phase correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, phase correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

- Notes: 1. MAX = 0xFF
 2. BOTTOM = 0x00

Apêndice D: ATmega328: Vetores de Interrupção

Em muitos microcontroladores, o endereço da Tabela de Vetores de Interrupção e os endereços das rotinas de serviço podem ser definidos pelo projetista. Mas a ordem dos endereços das rotinas de serviço dentro da Tabela de Vetores de Interrupção é estabelecida no projeto de um microcontrolador.

Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x0002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCINT0	Pin change interrupt request 0
5	0x0008	PCINT1	Pin change interrupt request 1
6	0x000A	PCINT2	Pin change interrupt request 2
7	0x000C	WDT	Watchdog time-out interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x0012	TIMER2 OVF	Timer/Counter2 overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x0016	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 compare match B
14	0x001A	TIMER1 OVF	Timer/Counter1 overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x0020	TIMER0 OVF	Timer/Counter0 overflow
18	0x0022	SPI, STC	SPI serial transfer complete
19	0x0024	USART, RX	USART Rx complete
20	0x0026	USART, UDRE	USART, data register empty
21	0x0028	USART, TX	USART, Tx complete
22	0x002A	ADC	ADC conversion complete
23	0x002C	EE READY	EEPROM ready
24	0x002E	ANALOG COMP	Analog comparator
25	0x0030	TWI	2-wire serial interface
26	0x0032	SPM READY	Store program memory ready

Figura 8: Tabela de Vetores de Interrupção

Para simplificar o uso do mecanismo de interrupção dos seus microcontroladores, os fabricantes oferecem um ambiente de desenvolvimento integrado (IDE) em que os códigos de definição da tabela de vetores de interrupção são incluídos. Nestes códigos os nomes simbólicos das rotinas de serviço são definidos deixando para o projetista apenas o trabalho de escrever as instruções das rotinas de serviço dos eventos de interesse. O seguinte trecho de código em *assembly* é uma implementação típica da tabela de vetores de exceção para o microcontrolador ATmega328.

The most typical and general program setup for the reset and interrupt vector addresses in Atmel® ATmega328P is:

Address	Labels	Code	Comments
0x0000		jmp RESET	; Reset Handler
0x0002		jmp EXT_INT0	; IRQ0 Handler
0x0004		jmp EXT_INT1	; IRQ1 Handler
0x0006		jmp PCINT0	; PCINT0 Handler
0x0008		jmp PCINT1	; PCINT1 Handler
0x000A		jmp PCINT2	; PCINT2 Handler
0x000C		jmp WDT	; Watchdog Timer Handler
0x000E		jmp TIM2_COMPA	; Timer2 Compare A Handler
0x0010		jmp TIM2_COMPB	; Timer2 Compare B Handler
0x0012		jmp TIM2_OVF	; Timer2 Overflow Handler
0x0014		jmp TIM1_CAPT	; Timer1 Capture Handler
0x0016		jmp TIM1_COMPA	; Timer1 Compare A Handler
0x0018		jmp TIM1_COMPB	; Timer1 Compare B Handler
0x001A		jmp TIM1_OVF	; Timer1 Overflow Handler
0x001C		jmp TIM0_COMPA	; Timer0 Compare A Handler
0x001E		jmp TIM0_COMPB	; Timer0 Compare B Handler
0x0020		jmp TIM0_OVF	; Timer0 Overflow Handler
0x0022		jmp SPI_STC	; SPI Transfer Complete Handler
0x0024		jmp USART_RXC	; USART, RX Complete Handler
0x0026		jmp USART_UDRE	; USART, UDR Empty Handler
0x0028		jmp USART_TXC	; USART, TX Complete Handler
0x002A		jmp ADC	; ADC Conversion Complete Handler
0x002C		jmp EE_RDY	; EEPROM Ready Handler
0x002E		jmp ANA_COMP	; Analog Comparator Handler
0x0030		jmp TWI	; 2-wire Serial Interface Handler
0x0032		jmp SPM_RDY	; Store Program Memory Ready Handler
		;	
0x0033	RESET:	ldi r16, high(RAMEND)	; Main program start
0x0034		out SPH, r16	; Set Stack Pointer to top of RAM
0x0035		ldi r16, low(RAMEND)	
0x0036		out SPL, r16	
0x0037		sei	; Enable interrupts
0x0038		<instr> xxx	

Figura 9: Uma implementação em *Assembly*.

Apêndice E: BME280 - Processamento de Sinais Brutos

Para aumentar a precisão dos valores amostrados pelo módulo BME280, o seu fabricante grava uma série de parâmetros na sua memória não-volátil, acessáveis pelos nomes simbólicos mostrados na Tabela 16. Estes parâmetros permitem ajustar as variações nos dados brutos.

Table 16: Compensation parameter storage, naming and data type

Register Address	Register content	Data type
0x88 / 0x89	dig_T1 [7:0] / [15:8]	unsigned short
0x8A / 0x8B	dig_T2 [7:0] / [15:8]	signed short
0x8C / 0x8D	dig_T3 [7:0] / [15:8]	signed short
0x8E / 0x8F	dig_P1 [7:0] / [15:8]	unsigned short
0x90 / 0x91	dig_P2 [7:0] / [15:8]	signed short
0x92 / 0x93	dig_P3 [7:0] / [15:8]	signed short
0x94 / 0x95	dig_P4 [7:0] / [15:8]	signed short
0x96 / 0x97	dig_P5 [7:0] / [15:8]	signed short
0x98 / 0x99	dig_P6 [7:0] / [15:8]	signed short
0x9A / 0x9B	dig_P7 [7:0] / [15:8]	signed short
0x9C / 0x9D	dig_P8 [7:0] / [15:8]	signed short
0x9E / 0x9F	dig_P9 [7:0] / [15:8]	signed short
0xA1	dig_H1 [7:0]	unsigned char
0xE1 / 0xE2	dig_H2 [7:0] / [15:8]	signed short
0xE3	dig_H3 [7:0]	unsigned char
0xE4 / 0xE5[3:0]	dig_H4 [11:4] / [3:0]	signed short
0xE5[7:4] / 0xE6	dig_H5 [3:0] / [11:4]	signed short
0xE7	dig_H6	signed char

O fabricante Bosch recomenda que utilize a API disponível pelos fabricantes do módulo para processar os dados brutos. Ele fornece, porém, as fórmulas de compensação dos sinais para os usuários implementarem o seu procedimento “*at the user’s risk*”. Essas fórmulas são transcritas em seguintes funções/rotinas/subrotinas:

```

// Returns temperature in DegC, resolution is 0.01 DegC. Output value of "5123" equals 51.23 DegC.
// t_fine carries fine temperature as global value
BME280_S32_t t_fine;
BME280_S32_t BME280_compensate_T_int32(BME280_S32_t adc_T)
{
    BME280_S32_t var1, var2, T;
    var1 = (((adc_T>>3) - ((BME280_S32_t)dig_T1<<1))) * ((BME280_S32_t)dig_T2) >> 11;
    var2 = (((((adc_T>>4) - ((BME280_S32_t)dig_T1)) * ((adc_T>>4) - ((BME280_S32_t)dig_T1))) >> 12) *
        ((BME280_S32_t)dig_T3)) >> 14;
    t_fine = var1 + var2;
    T = (t_fine * 5 + 128) >> 8;
    return T;
}

// Returns pressure in Pa as unsigned 32 bit integer in Q24.8 format (24 integer bits and 8 fractional bits).
// Output value of "24674867" represents 24674867/256 = 96386.2 Pa = 963.862 hPa
BME280_U32_t BME280_compensate_P_int64(BME280_S32_t adc_P)
{
    BME280_S64_t var1, var2, p;
    var1 = ((BME280_S64_t)t_fine) - 128000;
    var2 = var1 * var1 * (BME280_S64_t)dig_P6;
    var2 = var2 + ((var1*(BME280_S64_t)dig_P5)<<17);
    var2 = var2 + (((BME280_S64_t)dig_P4)<<35);
    var1 = ((var1 * var1 * (BME280_S64_t)dig_P3)>>8) + ((var1 * (BME280_S64_t)dig_P2)<<12);
    var1 = (((((BME280_S64_t)1)<<47)+var1))*((BME280_S64_t)dig_P1)>>33;
    if (var1 == 0)
    {
        return 0; // avoid exception caused by division by zero
    }
    p = 1048576-adc_P;
    p = ((p<<31)-var2)*3125/var1;
    var1 = (((BME280_S64_t)dig_P9) * (p>>13) * (p>>13)) >> 25;
    var2 = (((BME280_S64_t)dig_P8) * p) >> 19;
    p = ((p + var1 + var2) >> 8) + (((BME280_S64_t)dig_P7)<<4);
    return (BME280_U32_t)p;
}

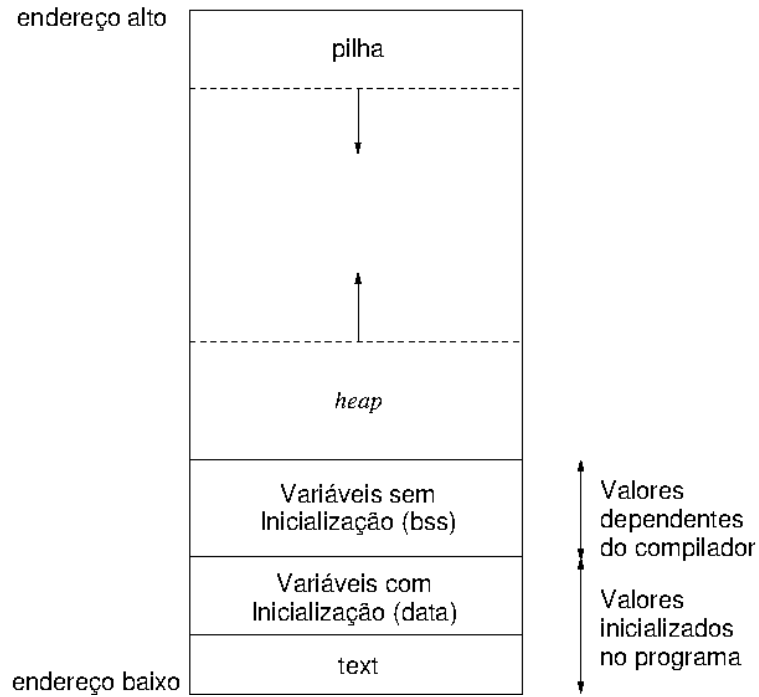
// Returns humidity in %RH as unsigned 32 bit integer in Q22.10 format (22 integer and 10 fractional bits).
// Output value of "47445" represents 47445/1024 = 46.333 %RH
BME280_U32_t bme280_compensate_H_int32(BME280_S32_t adc_H)
{
    BME280_S32_t v_xl_u32r;

    v_xl_u32r = (t_fine - ((BME280_S32_t)76800));
    v_xl_u32r = (((((adc_H << 14) - ((BME280_S32_t)dig_H4) << 20) - ((BME280_S32_t)dig_H5) * v_xl_u32r) +
        ((BME280_S32_t)16384)) >> 15) * ((((((v_xl_u32r * ((BME280_S32_t)dig_H6)) >> 10) * ((v_xl_u32r *
        ((BME280_S32_t)dig_H3)) >> 11) + ((BME280_S32_t)32768))) >> 10) + ((BME280_S32_t)2097152)) *
        ((BME280_S32_t)dig_H2) + 8192) >> 14);
    v_xl_u32r = (v_xl_u32r - (((v_xl_u32r >> 15) * (v_xl_u32r >> 15)) >> 7) * ((BME280_S32_t)dig_H1)) >> 4);
    v_xl_u32r = (v_xl_u32r < 0 ? 0 : v_xl_u32r);
    v_xl_u32r = (v_xl_u32r > 419430400 ? 419430400 : v_xl_u32r);
    return (BME280_U32_t)(v_xl_u32r>>12);
}

```

Apêndice F: Organização da Memória de Dados

Sob o ponto de vista da volatilidade dos dados, um programa pode conter variáveis locais e globais, variáveis alocadas estaticamente e dinamicamente. Para gerenciá-los, é comum dividir o espaço de memória volátil em 4 segmentos como ilustra a figura abaixo.



Apêndice G: Modelo OSI

O modelo OSI é um modelo “teórico” de referência. Ele é hoje muito aplicado para contextualizar um protocolo de comunicação e analisar as suas relações com outros protocolos.

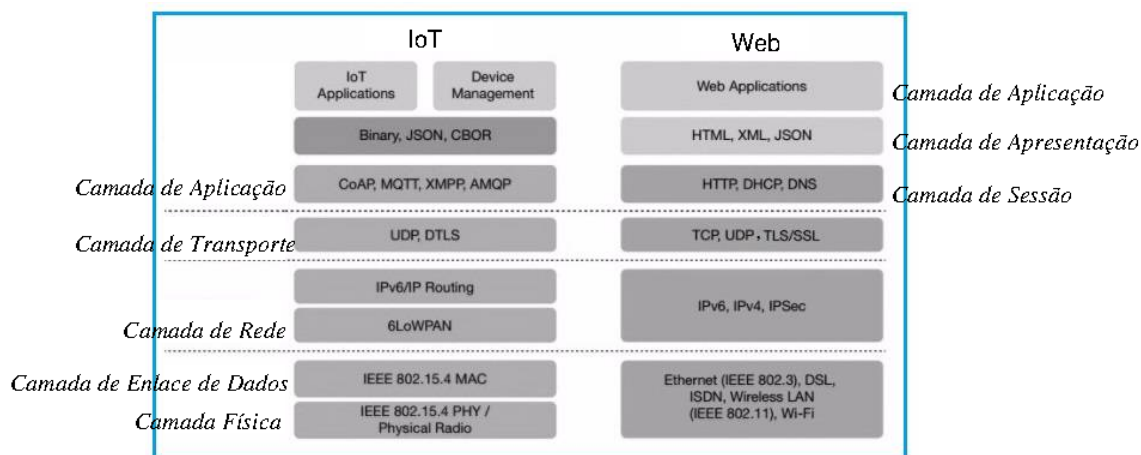


Figura 10: Tabela comparativa entre IoT e serviços de *Web* em termos de camadas de OSI.

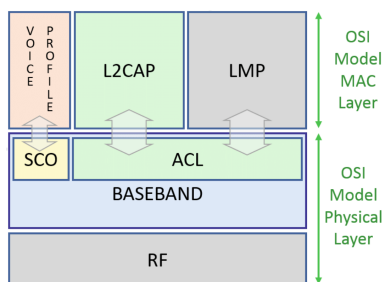
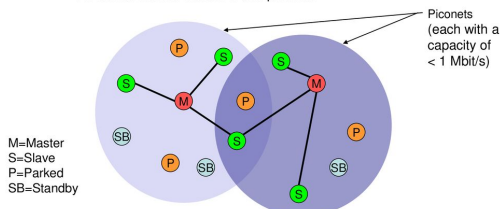


Figura 11: Bluetooth mapeado no modelo OSI.

Scatternet

- Linking of multiple co-located piconets through the sharing of common master or slave devices
 - A device can be slave in one piconet and master of another
 - No device can be master of two piconets



14

Figura 12: Redes de Bluetooth: Piconets com 1 mestre e até 7 escravos.