

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**  
**CENTRO DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**Aplicação de XBees em Automação**

Marcelo Siqueira Besch  
Danilo de Santana Pena

Natal, RN  
Julho de 2011

## Sumário

Introdução.....	3
1. Arquitetura.....	4
1.1 Descrição.....	4
1.2 Módulo Base.....	5
1.3 Microcontrolador.....	6
2. Aplicações.....	8
2.1 Descrição das Aplicações.....	8
2.2 Ambiente de Desenvolvimento.....	12
3. Microcontrolador.....	13
3.1 Descrição.....	13
3.2 Configuração.....	13
3.3 Periféricos.....	14
3.3.1 Timer.....	14
3.3.2 EEPROM.....	14
4. Sensores e Atuadores.....	18
4.1 Acelerômetro.....	18
4.2 Sensor de Força.....	19
4.3 Sensor de Temperatura.....	19
5. XBee.....	21
5.1 Comunicação Serial.....	22
Referências.....	23

## **Introdução**

O projeto visa o desenvolvimento de uma solução para receptividades e ações através de dispositivos eletrônicos que são utilizados para construir um ambiente inteligente. Utilizando um pouco do conceito de computação ubíqua, esta aplicação visa integrar aplicações úteis em objetos do cotidiano.

Este modelo de sistema possui por objetivo trazer conforto e eficiência no uso da automação, considerando que estes dispositivos inteligentes são embarcados e possuem fácil acessibilidade, utilizando-se um computador.

# 1. Arquitetura

## 1.1 Descrição

O projeto conta com aplicações que possuem sensores e atuadores como interface para o meio. Os dados das receptividades dos sensores e das ações dos atuadores serão transmitidos através de um dispositivo de comunicação sem fio, o XBee<sup>1</sup>. Este dispositivo transmite estas informações através de uma interface RF utilizando o padrão ZigBee<sup>8</sup> para o módulo Base, que por sua vez faz todo o gerenciamento e armazenamento destes dados.

No módulo base os dados recebidos são tratados, visto o protocolo utilizado, e armazenado em uma memória para posterior consulta. Para o tratamento e gerenciamento dos dados é utilizado um microcontrolador de 8 bits e mais alguns periféricos externos.

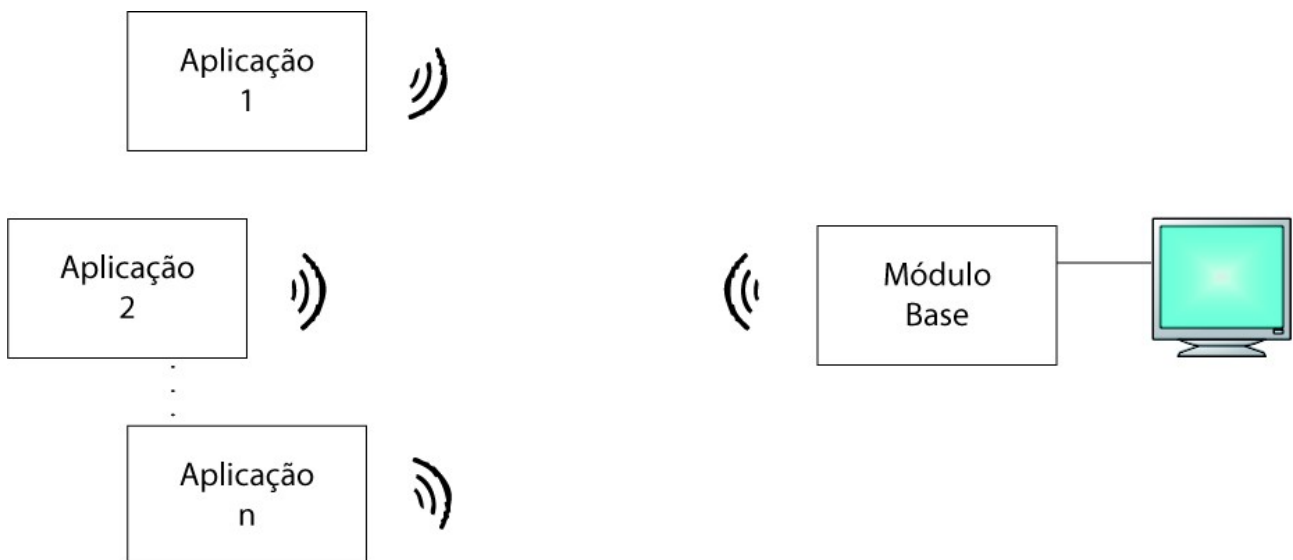


Figura 1: Esquemático.

Uma segunda interface (por fio), utilizada com comunicação UART<sup>9</sup>, é feita entre o módulo base e um computador. Sendo utilizada apenas para configuração das aplicações, do módulo e para consulta dos dados na memória.

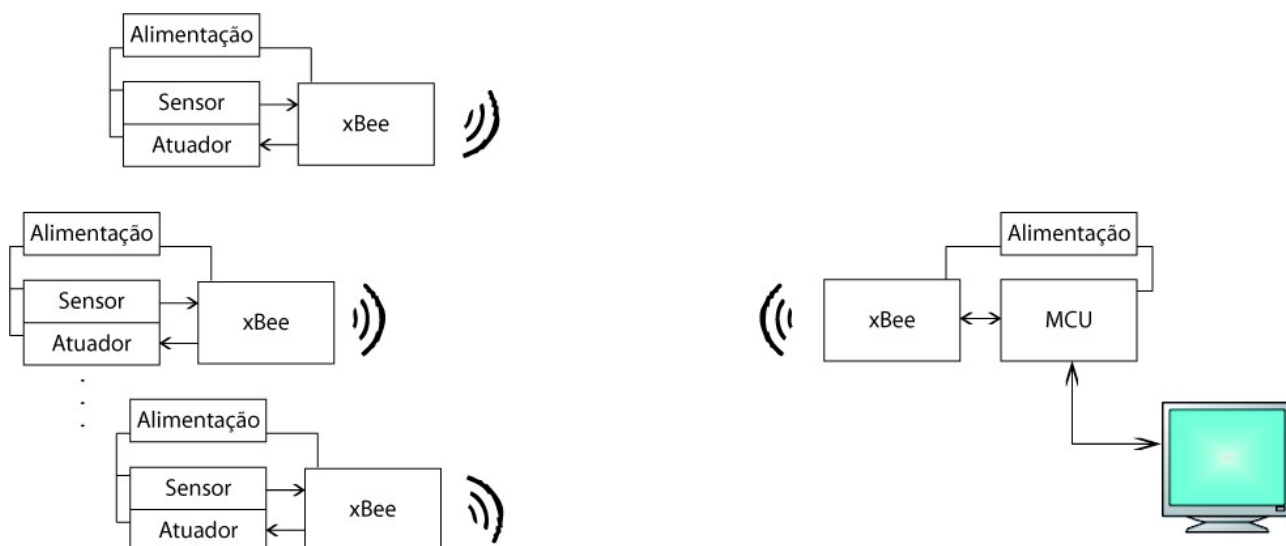


Figura 2: Esquemático detalhado.

## 1.2 Módulo Base

O circuito do módulo base ou módulo central implementa a comunicação do módulo XBee e da FTDI para o microcontrolador AVR ATmega328P. Como o microcontrolador possui apenas um par de pinos RX/TX para a USART, a solução encontrada para esta comunicação foi o uso de um CI multiplexador, que realiza o controle de quem realizará a comunicação com o AVR.

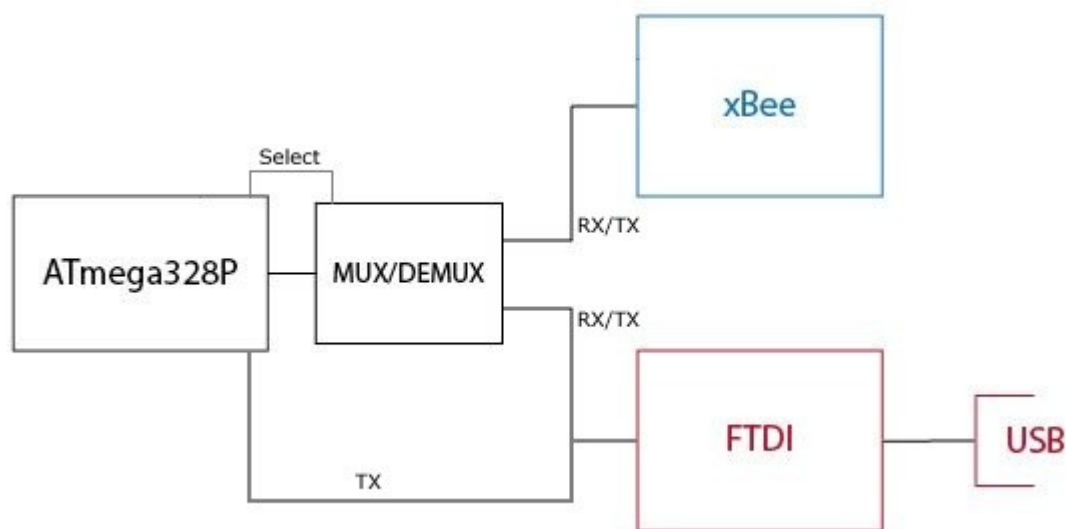


Figura 3: Circuito módulo base.

A prioridade de comunicação é para a FTDI e enquanto esta não precisar se comunicar o micro ficará se comunicando com o XBee. O AVR realiza o controle através de um pino de saída para o MUX 4052, onde o nível lógico 0 seleciona o XBee e o nível lógico 1 seleciona a FTDI. A configuração inicial é o nível lógico 0, porém quando a FTDI realiza a transmissão de algum byte, a via de comunicação TX da FTDI que estava no nível IDLE (nível lógico 1) vai para o nível 0 (start

bit). Esta transição é percebida por uma interrupção externa do AVR (interrupção INT0, borda de descida), que realiza o controle do MUX para que o AVR escute a FTDI.

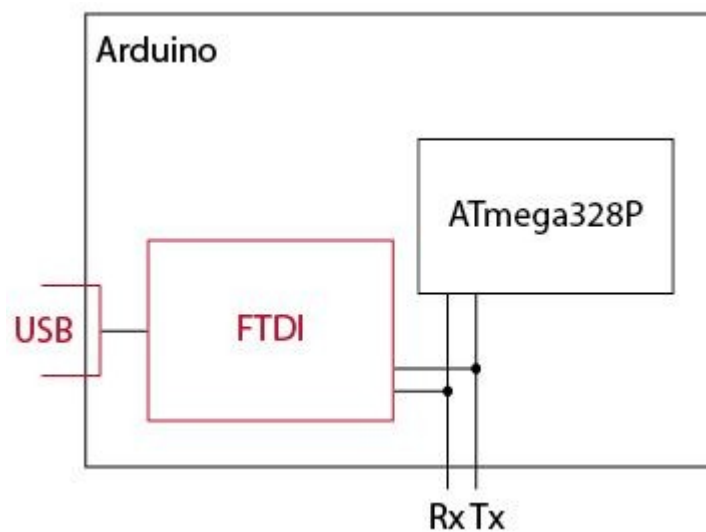


Figura 4: Esquemático FTDI no kit Arduino.

### 1.3 Microcontrolador

O microcontrolador possui uma arquitetura lógica baseada em um modelo de atividades, onde são executadas conforme o algoritmo de software.

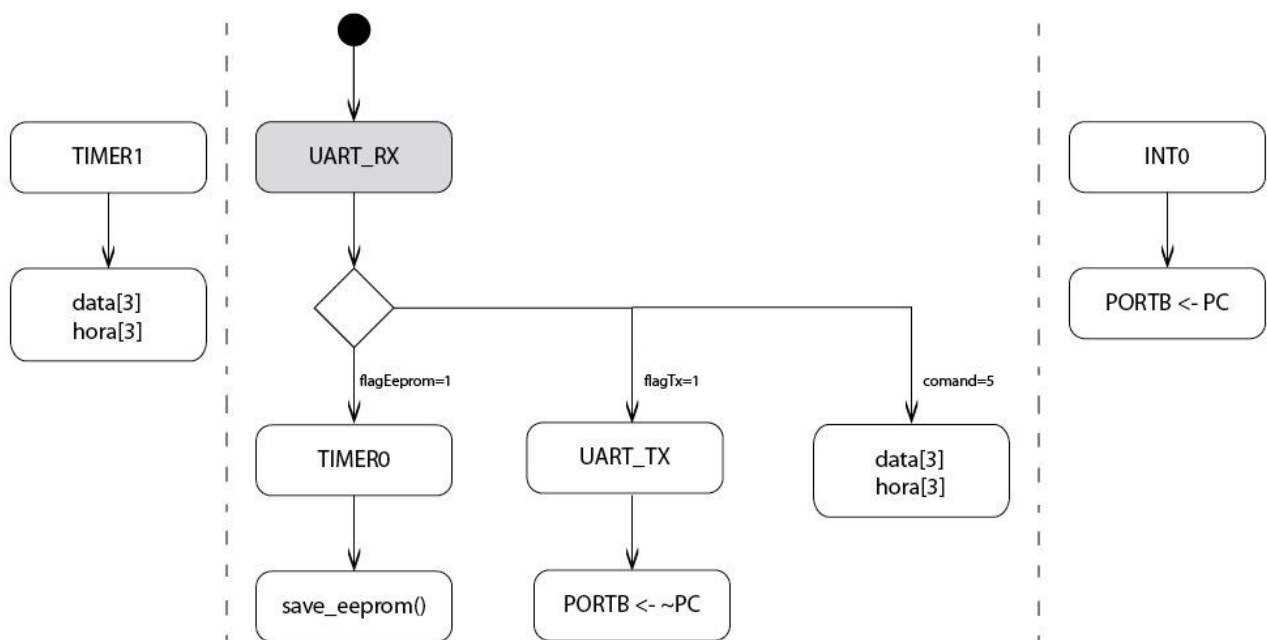


Figura 5: Diagrama de Atividades.

Basicamente são realizadas três operações em simultâneo, operações de interrupção de

Timer1, que realiza atualizações nos vetores data[3] e hora[3], interrupções da UART RX, onde existe uma máquina de estados para tratamento dos bytes recebidos e por fim interrupções de INT0, que é uma interrupção externa ligada ao pino D2 do microcontrolador, ela possui por função monitorar solicitações de comunicação do computador.

Possui como interface externa, pinos RX e TX de comunicação UART, INT0 para interrupção externa e saídas para LED's.

A máquina de estados é inicializada por um estado inicial E0, que verifica quais bytes estão chegando e encaminham para os estados corretos para tratamento dos dados.

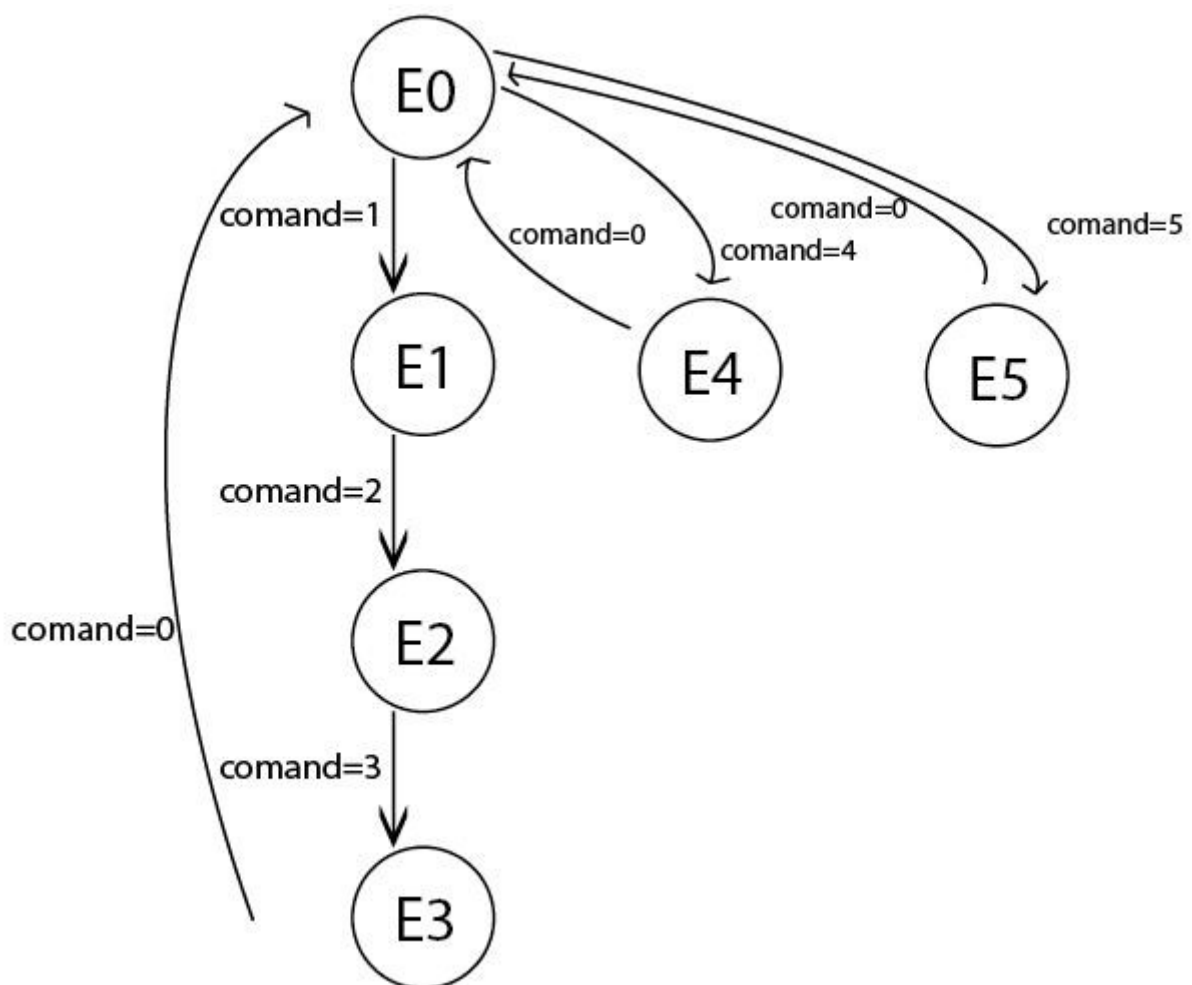


Figura 6: Máquina de estados.

O caminho E1->E2->E3 realiza o tratamento de um frame de bytes do módulo XBee, o estado E4 realiza o tratamento de um comando do computador de solicitação de medições e por fim o estado E5 realiza o tratamento de um comando do computador de sincronismo de data[3] e hora[3].

## 2. Aplicações

As aplicações possuem um conjunto de sensores e/ou atuadores, um módulo XBee e componentes para circuito. Abaixo segue a descrição de algumas aplicações e levantamento de material necessário para a implementação.

### 2.1 Descrição das Aplicações

#### 1. Medição de temperatura

Utiliza um sensor de temperatura (TMP37) para medição da temperatura.

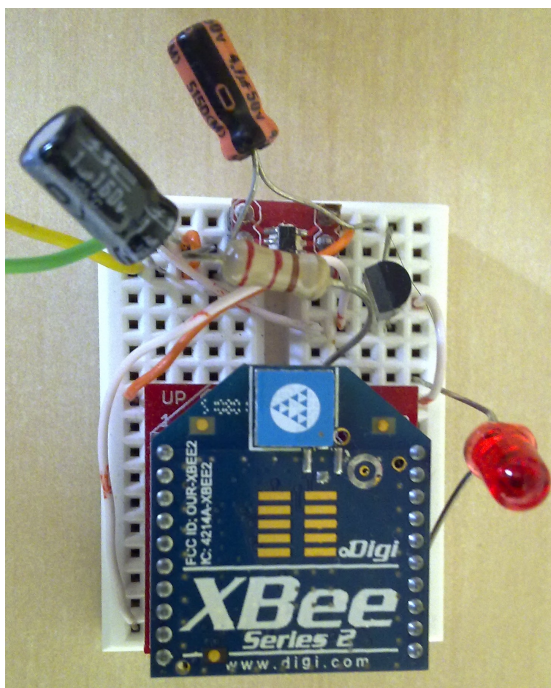


Figura 7: Medição de temperatura.

Materiais:

Qtd.	Descrição	Valor
1	Módulo XBee <sup>2</sup>	R\$ 44,00
1	TMP37 <sup>3</sup>	R\$ 2,85
1	Regulador de tensão TPS76933 <sup>4</sup>	R\$ 1,55
1	LED	-
2	Capacitores	-
1	Bateria 9V	R\$ 8,40

#### 2. Medição de consumo energia elétrica

Utiliza como principal componente o CI ADE7757, que realiza a medição de energia.



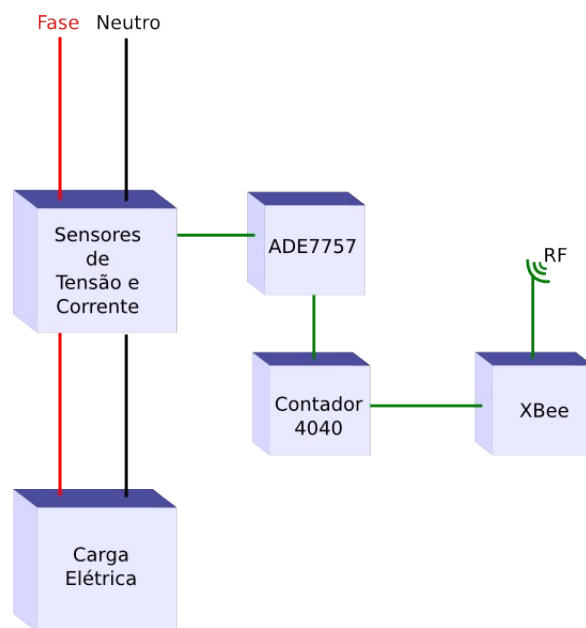


Figura 8: Diagrama da medição de consumo.

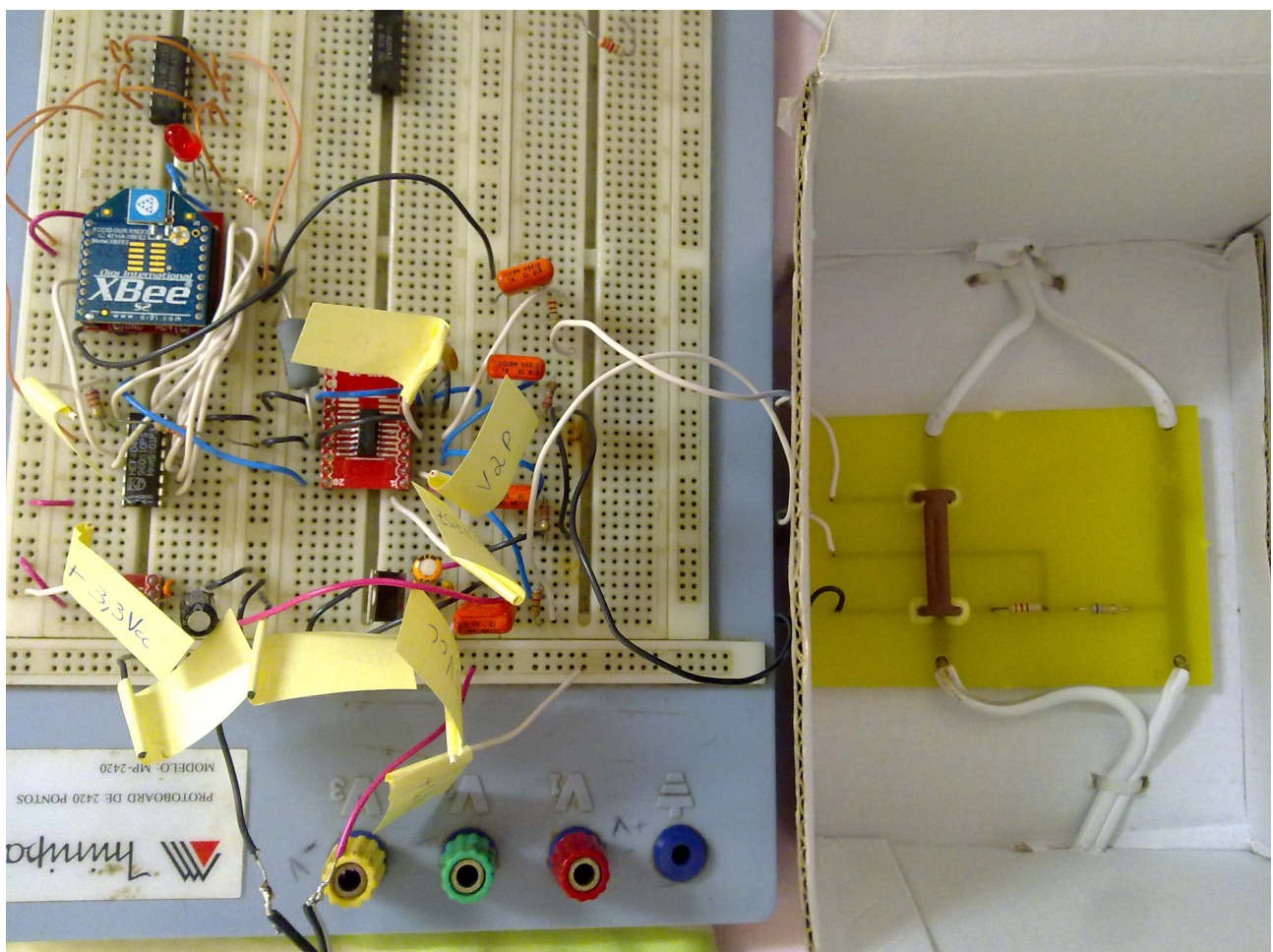
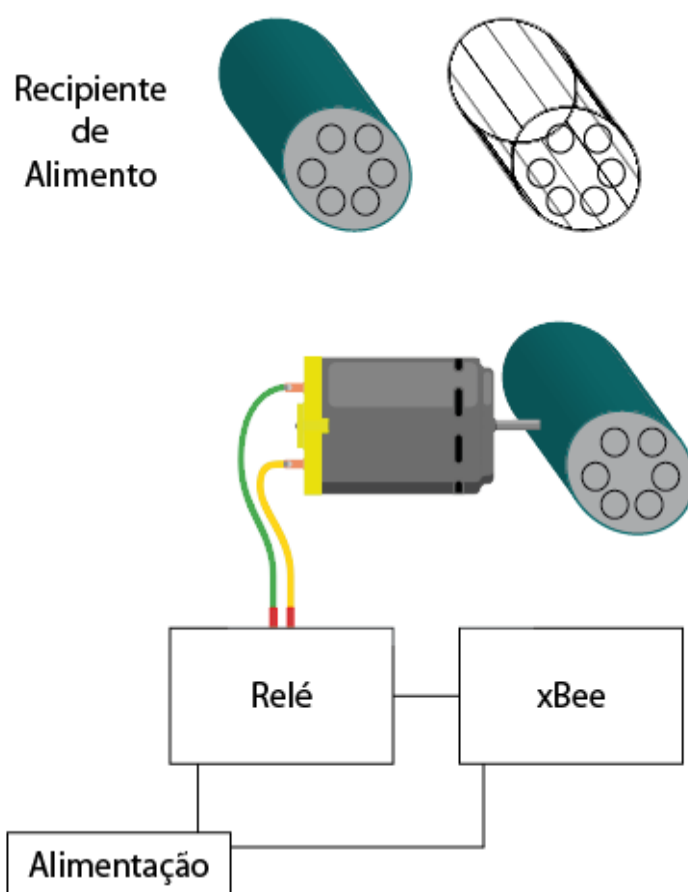


Figura 9: Medição de consumo.

Materiais:

Qtd.	Descrição	Valor
1	Módulo XBee	R\$ 44,00
1	Regulador de tensão TPS76933	R\$ 1,55
1	Medidor de energia ADE7757 <sup>5</sup>	R\$ 6,00
1	Regulador de tensão LM7805	R\$ 1,10
1	CI 4040	R\$ 1,20
1	Resistor de precisão de 5k $\Omega$	-
1	Resistor de precisão de 1m $\Omega$	R\$ 6,80
1	Fonte 220VAC - 12VDC	-
1	Conector macho de tomada	-
1	Conector fêmea de tomada	-
-	Componentes gerais de circuito (resistores, capacitores, ...)	-

### 3. Alimentação de peixes



Materiais:

Qtd.	Descrição	Valor
1	Módulo XBee	R\$ 44,00
1	Regulador de tensão – TPS76933	R\$ 1,55
1	Motor DC	-
1	Relé	R\$ 3,00
1	Recipiente de comida	-

#### 4. Segurança em portas e janelas

A segurança em portas e janelas podem ser implementadas através da simples ideia de contato de um ímã e um sensor hall magnético. Em que separação dos dois será percebida acionando-se um alarme.

Materiais:

Qtd.	Descrição	Valor
1	Módulo XBee	R\$ 44,00
1	Regulador de tensão – TPS76933	R\$ 1,55
1	Sensor Hall	-
1	ímã	-
1	Buzzer	-

#### 5. Iluminação externa

A iluminação externa segue o mesmo princípio que hoje é aplicado em postes e iluminação externa residencial.

Materiais:

Qtd.	Descrição	Valor
1	Módulo XBee	R\$ 44,00
1	Regulador de tensão – TPS76933	R\$ 1,55
1	Photosensor	-

#### 6. Controle de cortinas

O controle de cortinas é realizado através de um comando do usuário que aciona um relé e por si um motor DC realizando a operação de fechar ou abrir a cortina.

Materiais:

Qtd.	Descrição	Valor
------	-----------	-------



1	Módulo XBee	R\$ 44,00
1	Regulador de tensão – TPS76933	R\$ 1,55
1	Relé	R\$ 3,00
1	Motor DC	-

## 2.2 Ambiente de Desenvolvimento

Segue abaixo material necessário para construir o ambiente de desenvolvimento.

Materiais:

Qtd.	Descrição	Valor
1	Módulo XBee	R\$ 44,00
1	XBee Explorer	R\$ 42,40
1	Arduino Duemilanove	R\$ 30,00
1	Protoboard	-
-	Fonte, Sockets, Beckett Boards, componentes	-

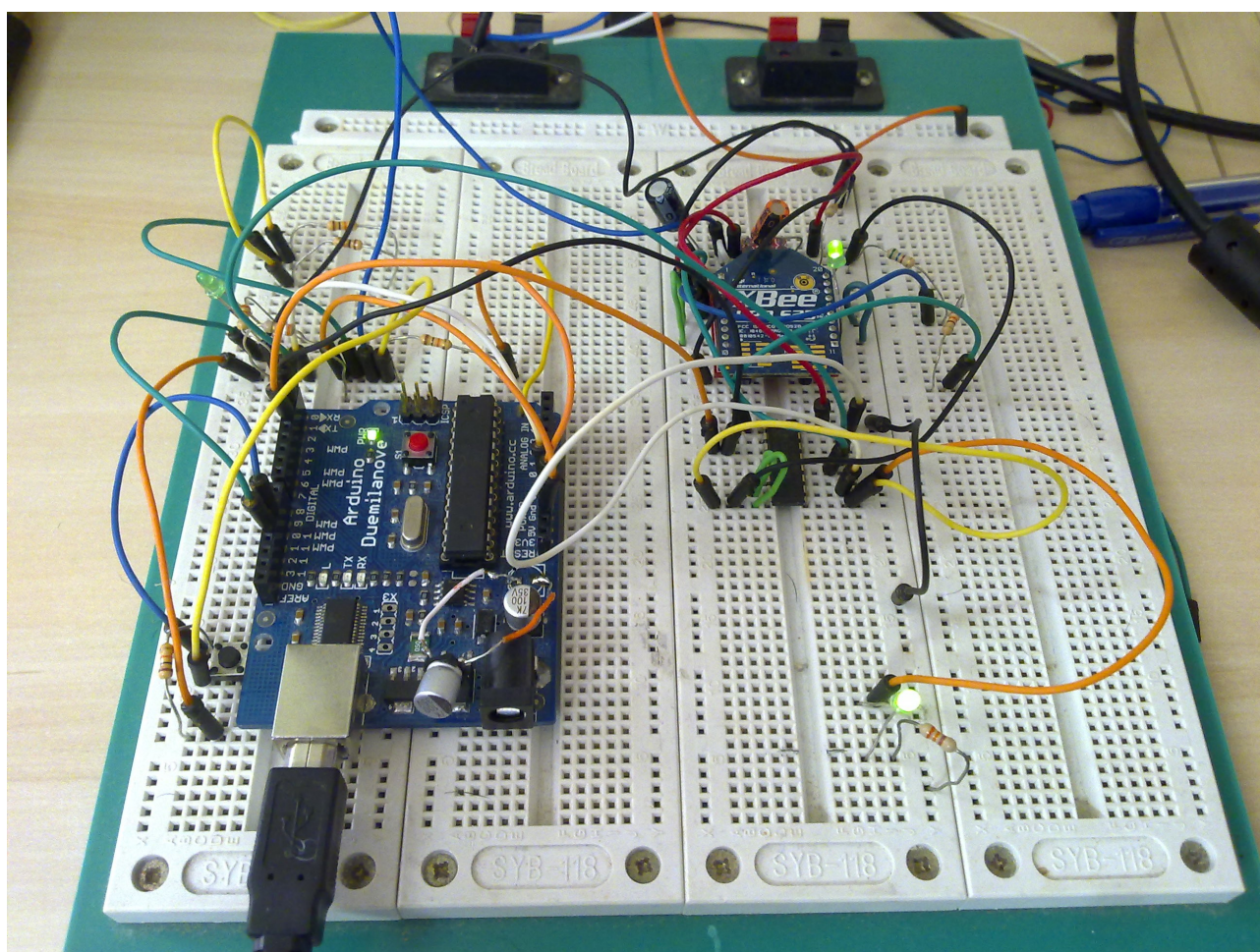


Figura 10: Módulo base.

### 3. Microcontrolador

#### 3.1 Descrição



Figura 11: Kit de desenvolvimento.

O microcontrolador a ser utilizado é um ATmega328P, da família megaAVR da fabricante Atmel. Trata-se de um micro de 8 bits que apesar de possuir uma arquitetura RISC possui um conjunto de 131 instruções. Sua escolha foi baseada em seu rico conjunto de periféricos, possui 32KB de memória Flash, 1KB de EEPROM e 2KB de SRAM. Timers de 8 e 16 bits além de comunicação UART e SPI.

Para que se opere com frequência de 16MHz, que está sendo utilizada como oscilador externo, a alimentação do microcontrolador deve ser de 5V.

A linguagem de programação utilizada é o C e o compilador é o avr-gcc. O ambiente de programação é o AVR Studio 4 e utiliza-se o AVRDUDE para escrita e leitura das memórias do micro. Contamos com ajuda da comunidade AVR Freaks Forum<sup>10</sup>.

#### 3.2 Configuração

Para configuração do micro foi necessário a utilização de registros que definem essas configurações. Os detalhes desses registros são encontradas na documentação do microcontrolador<sup>6</sup>.

Diversos periféricos possuem mais de um bloco para uso, sendo sua representação por números  $n$ . Por exemplo, o periférico UART é referenciado como UART0, ou UARTn para as demais. O mesmo vale para os registros.

Este micro possui 23 pinos de I/O, sendo sua configuração de entrada/saída através do registro  $DDRx$ . Para escrita o  $PORTx$  e para leitura o  $PINx$ .

```
Ex:  
DDRB = (1<<DDB3)|(1<<DDB4)|(1<<DDB5); // Saída  
PORTB = 0; // Zera as saídas
```

### 3.3 Periféricos

#### 3.3.1 Timer

O bloco periférico Timer0 é um timer de 8 bits e o bloco Timer1 é um timer de 16 bits. Eles possuem como principais registros o TIMSKn para habilita-lo e o registro TCCRnx para escolher o prescaler.

#### 3.3.2 EEPROM

A memória EEPROM foi utilizada inicialmente para armazenar os dados das aplicações do módulo base. Porém, levando em conta o baixo custo de memória DataFlash, foi especificado o uso de uma memória externa flash de 16 megabits, o equivalente a 2 megabytes.

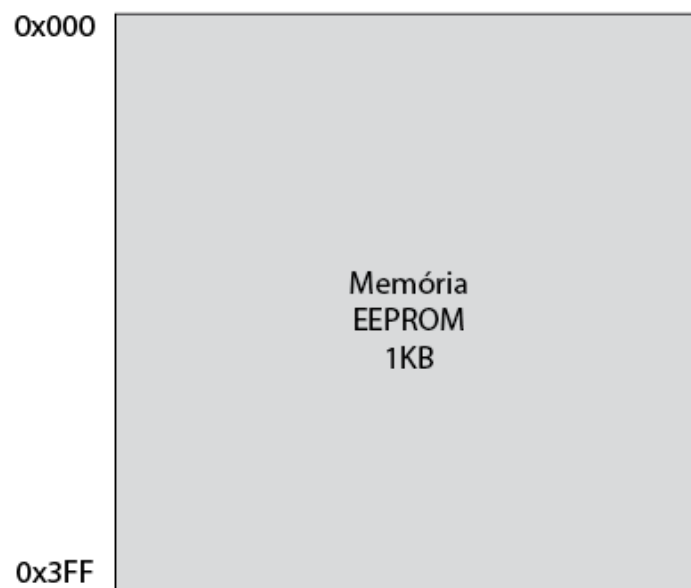


Figura 12: Memória EEPROM.

O endereço 0x000 não é utilizado, porque apesar da memória EEPROM ser uma memória não volátil ao resetar o microcontrolador neste endereço é escrito o valor 0x00. Então perdi-se o valor armazenado.

No valor 0x001 é escrito a variável totalRegTab1 que é o total de registros armazenados na tabela 1. A tabela 1 vai de 0x002 à 0x0FF, ela é responsável por armazenar o ID e o serial de cada

módulo XBee end device (módulo das aplicações).

Na tabela 2 também temos uma variável totalReg que também é o total de registro da tabela 2. Lembrando, que neste caso estamos limitados a apenas 1 byte. A tabela 2 vai de 0x102 à 0x3FF, sendo utilizado apenas 255 registros, dos 765 possíveis (devido ao totalReg limitado).

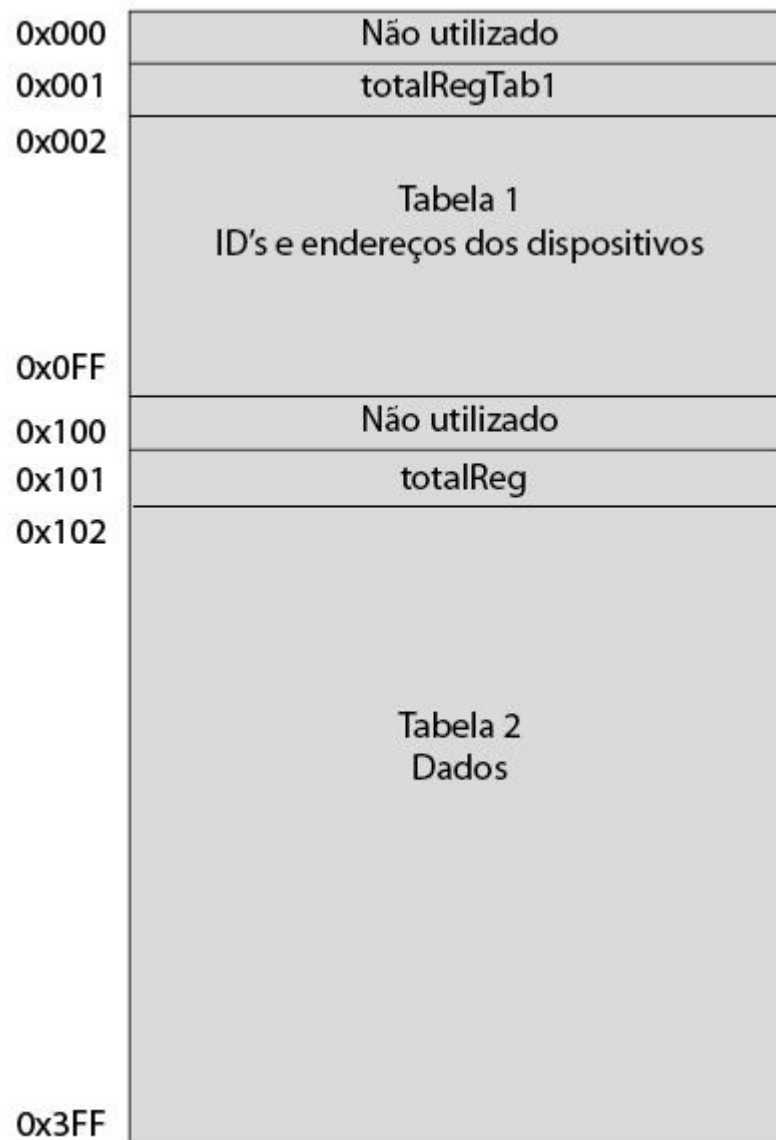


Figura 13: Organização da memória.

A tabela 1 é construída de registros de 8 bytes, sendo o primeiro o símbolo "\$", que corresponde ao valor hexadecimal 24. Em seguida vem 2 bytes que é um ID associado ao serial do módulo XBee. Depois vem os 4 bytes de endereço serial do módulo, onde cada módulo possui o seu serial fixo (sua origem é da fábrica). Por fim, a quebra de linha que corresponde ao valor 0x0A.

Cada vez que um dispositivo end device é conectado a rede o módulo base designa um ID associado ao seu serial. Este ID será referenciado no software do computador para que o usuário saiba quais medições correspondem a quais aplicações.

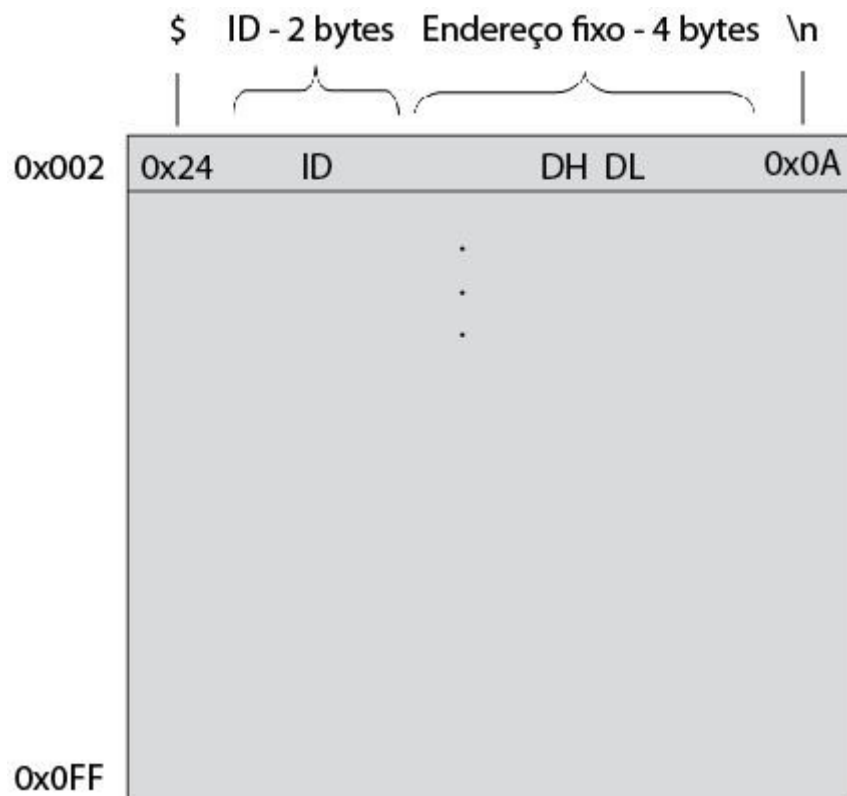


Figura 14: Tabela 1.

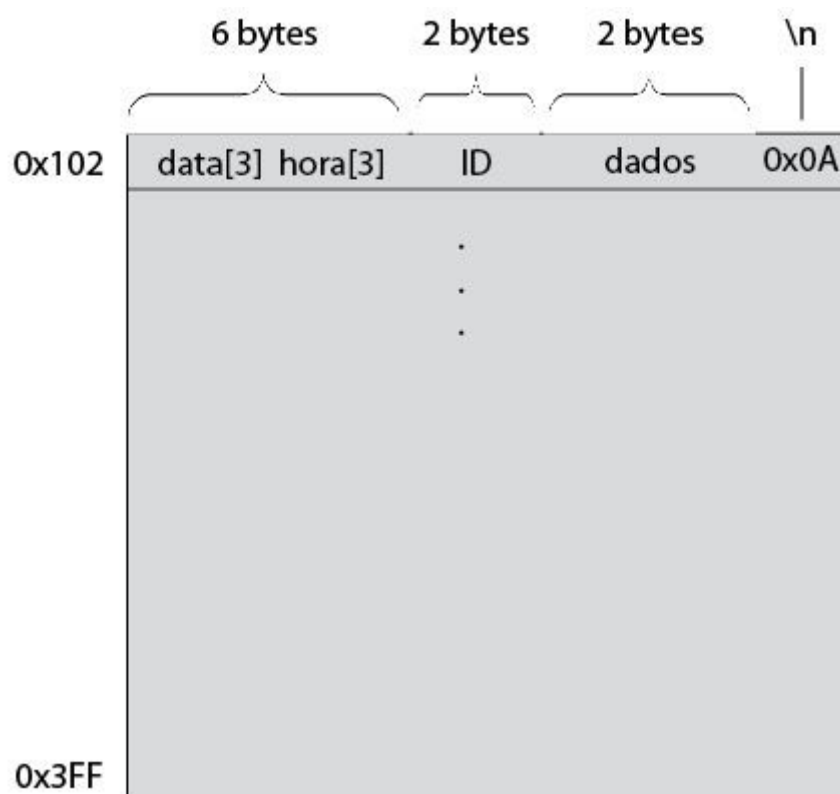


Figura 15: Tabela 2.



Na tabela 2, os registros possuem 11 bytes, sendo os 6 primeiros os vetores data[3] e hora[3] que são a data e hora que foram recebido cada medição. Em seguida o ID associado ao dispositivo de origem, depois 2 bytes de dados que são informações de medidas digitais e por fim um byte fixo de valor 0x0A.

Os dois bytes de dados serão tratados no software do computador, sendo implementado suas devidas conversões.

Baseado no tempo de amostragem e da quantidade de aplicações é possível saber quanto tempo o usuário pode ficar sem consultar os dados do módulo base.

Vejamos, considerando a memória DataFlash de 16 megabits (2048 kbytes, 2097152 bytes). São um total de 11 bytes por registro.

$$\frac{2097152}{11} = 190650 \text{ registros}$$

Se utilizarmos um tempo de amostragem de 5 minutos, e caso seja usado 2 aplicações, teremos 288 registros em um dia para cada aplicação e 576 para as duas. Assim:

$$\frac{190650}{576} = 330 \text{ dias}$$

Serão necessários 330 dias para que se consiga ocupar toda a memória flash. Antes dos 330 dias o usuário deve realizar uma consulta no módulo base através de um comando de solicitação de medições com um computador, assim a memória será limpada e será liberado novamente os registros.

## 4. Sensores e Atuadores

Neste capítulo será descrito as principais características de sensores e atuadores que podem ser utilizados nas aplicações.

### 4.1 Acelerômetro

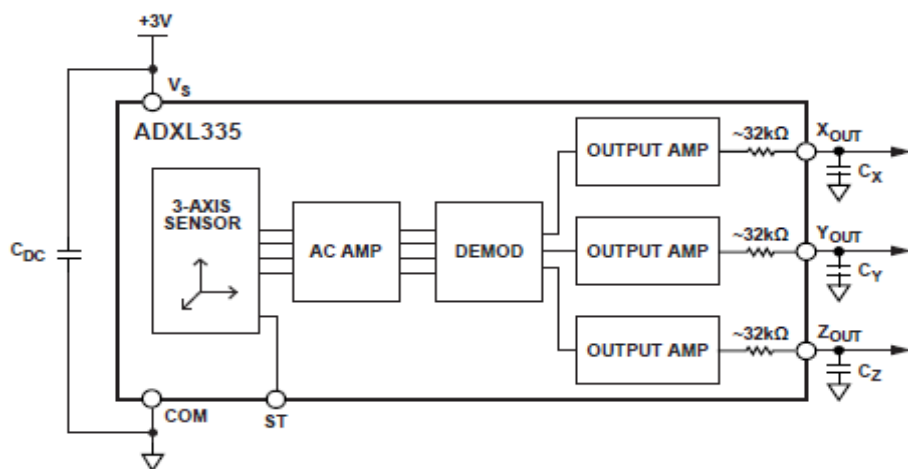
Modelo: ADXL335

Fabricante: Analog Device

Alimentação: 1.8 a 3.6V

O ADXL335 é pequeno, fino, de baixa potência (350uA), com 3 sinais de saída de tensão para 3-eixos. Este acelerômetro mede para um mínimo de 3g.

O usuário seleciona a largura de banda usando os capacitores  $C_x$ ,  $C_y$  e  $C_z$  para os pinos  $X_{out}$ ,  $Y_{out}$  e  $Z_{out}$ . A largura de banda tem a faixa de 0.5 a 1600Hz para X e Y, e 0.5 a 550Hz para o eixo Z.



Considerando condições de operação de  $V_s = 3V$  e  $T = 25^\circ C$ . Ele mede  $\pm 3.6g$ . Possui sensibilidade de 300mV/g (variação de apenas  $\pm 0.01\%/^\circ C$ ). O nível 0g está posicionado em 1.5V na saída.

Para maioria das aplicações um simples capacitor de 0.1uF em  $C_{dc}$  é suficiente para eliminar ruídos. O dispositivo possui um limite de banda nos pinos  $X_{out}$ ,  $Y_{out}$  e  $Z_{out}$ , sendo necessário adicionar capacitores para implementar um filtro. A equação:

$$F = 5uF/C$$

O valor mínimo recomendado para todos os pinos é de 0.0047uF.

Segue tabela com larguras de banda para diferentes valores de capacitores:

Bandwidth (Hz)	Capacitor ( $\mu$ F)
1	4.7
10	0.47
50	0.10
100	0.05
200	0.027
500	0.01

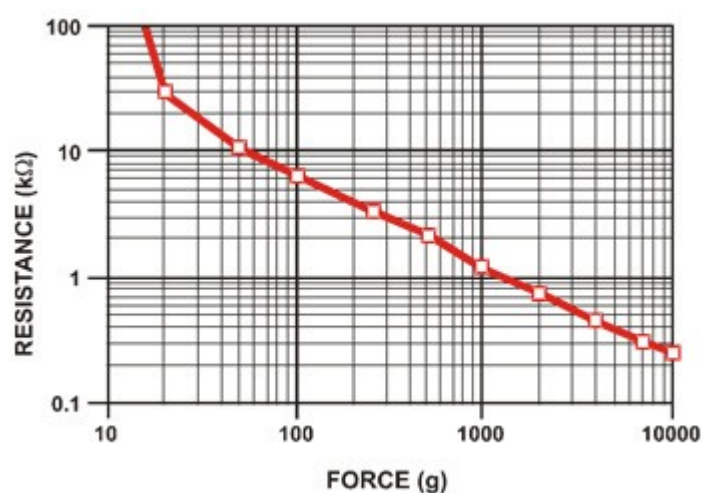
## 4.2 Sensor de Força

Também chamado de FSR (Force Sensitive Resistor) detecta pressão física, força e peso. São basicamente um resistor (em ohms  $\Omega$ ) que muda de valor dependendo de quanto foi pressionado. São de baixo custo, fácil de usar e de baixa precisão.

A sua faixa de resistência varia de infinito ou circuito aberto para nenhuma pressão, 100K $\Omega$  para pouca pressão e 200K $\Omega$  para máxima pressão.

Sua faixa de força varia de 0 a 100 Newtons. Sua alimentação não é especificada, basta apenas 1mA de corrente (com pullup/down).

Gráfico abaixo mostra sua variação de resistência por força.



## 4.3 Sensor de Temperatura

Modelo: TMP36

Fabricante: Analog Device

Alimentação: 2.7 a 5.5V

Possui um fator de escala de  $10\text{mV}/^{\circ}\text{C}$ . Especificado para  $-40^{\circ}\text{C}$  a  $+125^{\circ}\text{C}$ , possui acurácia de  $\pm 2^{\circ}\text{C}$ .

Esquema do TMP36 modelo T-3:

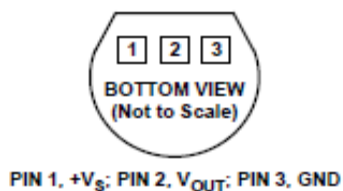
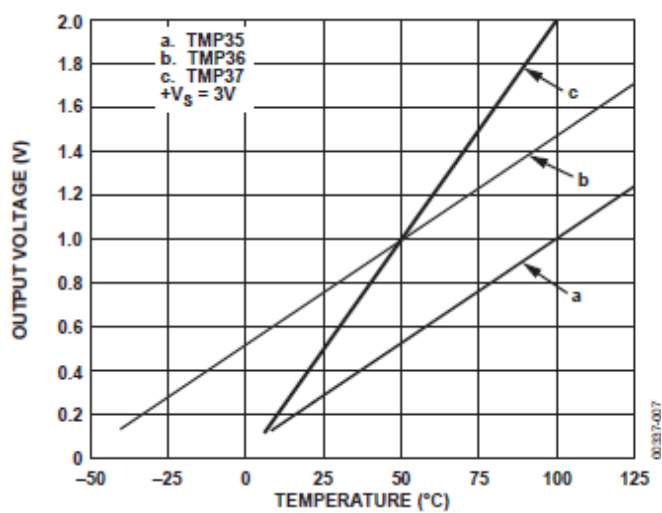


Gráfico de variação da saída com a temperatura:



## 5. XBee

Os módulos Xbees utilizam a tecnologia ZigBee Wireless que é um padrão que define um protocolo de comunicação e que possui por objetivo baixo custo, baixa transmissão de dados, uma curta faixa de rede wireless e com proposta de ser utilizado com boa vida útil no uso de bateria.

Eles operam com frequência de 2.4GHz. Possui alcance de 40m, baixa potência com alimentação de 3.3V (40mA) em operação. A taxa de transmissão de dados em RF é de 250000 bps, a taxa de dados serial é de 1200 a 230400 bps.

Pinos:

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / $\overline{\text{CONFIG}}$	Input	UART Data In
4	DIO12	Either	Digital I/O 12
5	$\overline{\text{RESET}}$	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI / DIO10	Either	PWM Output 0 / RX Signal Strength Indicator / Digital IO
7	DIO11	Either	Digital I/O 11
8	[reserved]	-	Do not connect
9	$\overline{\text{DTR}}$ / SLEEP_RQ / DIO8	Either	Pin Sleep Control Line or Digital IO 8
10	GND	-	Ground
11	DIO4	Either	Digital I/O 4
12	$\overline{\text{CTS}}$ / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7. CTS, if enabled, is an output.
13	ON / $\overline{\text{SLEEP}}$	Output	Module Status Indicator or Digital I/O 9
14	VREF	Input	Not used on this module. For compatibility with other XBee modules, we recommend connecting this pin to a voltage reference if Analog sampling is desired. Otherwise, connect to GND.
15	Associate / DIO5	Either	Associated Indicator, Digital I/O 5
16	$\overline{\text{RTS}}$ / DIO6	Either	Request-to-Send Flow Control, Digital I/O 6. RTS, if enabled, is an input.
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0 / Commissioning Button	Either	Analog Input 0, Digital IO 0, or Commissioning Button

Pinos de entrada não utilizados podem ser conectador em pull-up resistores (30K) com o comando de software PR.

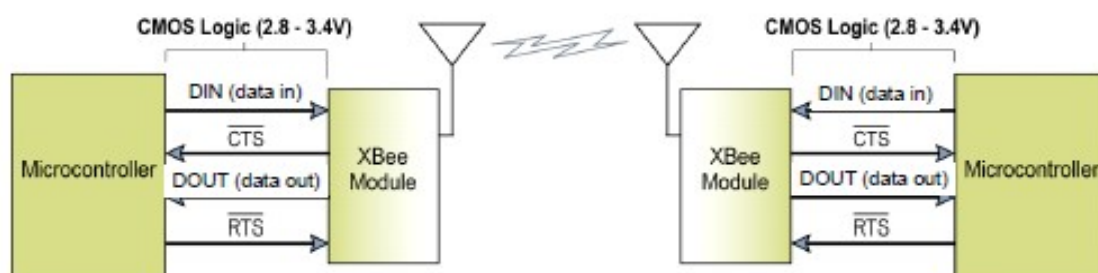
Os pinos Associated Led (pino 15) e o Commissioning Button (pino 20) diz respeito a saída de um LED e entrada de um button e funcionam dependendo do estado do modulo na rede, sem precisar usar os comandos via UART para usar algumas funções.

Mais detalhes podem ser encontrados no manual dos módulos XBee<sup>7</sup>.

## 5.1 Comunicação Serial

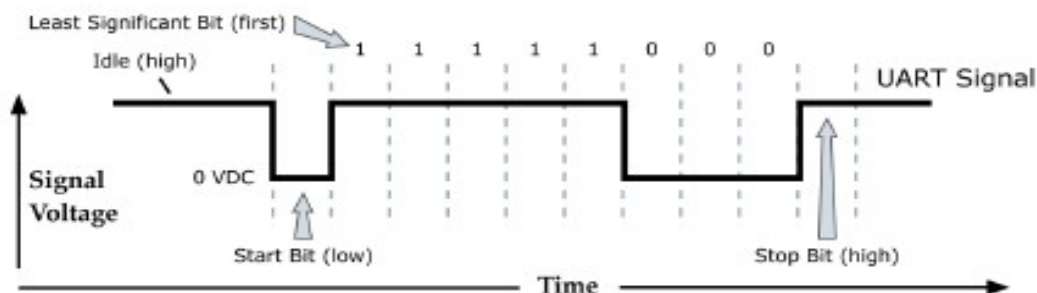
Os módulos RF possuem uma interface para com o host através de uma porta serial assíncrona. Esta comunicação utiliza a UART podendo utilizar alguns padrões que usem a UART como USB, RS-232.

O fluxo de dados da comunicação serial:



Os dados do módulo UART utiliza a entrada DIN (pino 3) como um sinal assíncrono serial. Cada byte consiste de um start bit (low), 8 bits de dado e por fim um stop bit (high).

Abaixo segue a transmissão do valor 0x1F (número decimal 31):



Para que haja comunicação se faz necessário configurar duas UART de modo compatíveis, o baud rate, paridade, start bits, stop bits e bits de dados.

O XBee suporta dois tipos de Protocolos de Interface Serial, o transparente e API (Application Programming Interface).

- Operação Transparente

Nesta operação todos os dados recebidos pela UART através do DIN é enviado via RF e quando o dado é recebido via RF este é automaticamente enviado pela DOUT na UART.

- Operação API

Nesta operação os módulos podem interagir com os recursos da rede. Todos os dados que entram e saem do módulo está contido em frames que definem as operações ou eventos dentro do módulo.

## Referências

- [1] - XBee® ZB ZigBee® RF Modules - <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/zigbee-mesh-module/XBee-zb-module.jsp>
- [2] - XBee 1mW Chip Antenna - <http://www.sparkfun.com/products/8664>
- [3] - Digikey TMP37FT9Z-ND - <http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=TMP37FT9Z-ND>
- [4] - Digikey TPS76933DBVR - <http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=296-11039-1-ND>
- [5] - Digikey ADE7757ARNZ-ND - <http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=ADE7757ARNZ-ND>
- [6] - Atmel ATmega328P - [http://www.atmel.com/dyn/products/product\\_card.asp?part\\_id=4198](http://www.atmel.com/dyn/products/product_card.asp?part_id=4198)
- [7] - Digi Product Manual: XBee / XBee-PRO ZB OEM RF Modules - [http://ftp1.digi.com/support/documentation/90000976\\_G.pdf](http://ftp1.digi.com/support/documentation/90000976_G.pdf)
- [8] - Shahin Farahani, ZigBee Wireless Networks and Transceivers, 2008.
- [9] - Jerry Luecke, USART Serial Communications, Analog and Digital Circuits for Electronic Control Systems Applications, 2005.
- [10] - AVR Freaks - <http://www.avrfreaks.net/>