

João Pedro dos Reis  
Gustavo Deitos Bernardini

## Proposta de projeto

Relatório da disciplina Projeto em Eletrônica II  
Professor Eduardo Luiz Ortiz Batista  
Universidade Federal de Santa Catarina

Florianópolis, Setembro de 2015

## Sumário

1. INTRODUÇÃO .....	3
2. DESENVOLVIMENTO .....	4
2.1 <i>HARDWARE</i> .....	4
2.2 CRONOGRAMA .....	7
3. UPDATE FINAL .....	9
3.1 TRANSMISSÃO SEM FIO .....	9
3.2 TECLADO .....	10
3.3 <i>DISPLAY</i> .....	10
3.4 FLUXOGRAMA DE OPERAÇÃO .....	10
3.5 ANEXO DE FOTOS FINAIS .....	11
4. CONCLUSÕES .....	12

## **1. INTRODUÇÃO**

O projeto consiste na implementação de dois dispositivos iguais de envio/recepção de dados digitais sem fio. Os quais serão interpretadas como simples texto ou sinais de controle dependendo da sintaxe enviada, podendo ser usados para comunicação ou para controle de outros dispositivos

A transmissão será realizada via protocolo serial e modulado com uma modulação AM na faixa UHF (300MHz ~ 3GHz).

Ambos os dispositivos serão controlados por um microcontrolador, que além de interpretar os dados recebidos/enviados, também gerenciará os blocos de interface (antenas, display, teclado, etc).

## 2. DESENVOLVIMENTO

### 2.1 HARDWARE

Para a construção do sistema serão necessários sistemas iguais, os quais os materiais definidos foram:

- uma antena receptora e transmissora AM 433MHz;
- um micro controlador Arduino Nano v3 atmega 328;
- um teclado 4x4, matricial passivo;
- um display 16x2

A escolha do micro controlador se deve à facilidade de implementação do sistema, ou seja, a plataforma Arduino é uma boa alternativa para prototipação de projetos.

Os demais componentes são componentes típicos para seus propósitos, fácil de achar, fácil de usar e de baixo custo.

2xArduino Nano v3	R\$40.55
2xLCD 16x2	R\$33.68
2xTeclado 4x4	R\$26.10
2xTransmissor/Receptor AM 433MHz	R\$34.20
<b>TOTAL</b>	<b>R\$134.53</b>

Tabela 1 - Preços prototipação

R\$/Hora de um engenheiro	R\$55.00/Hora
Horas/semana dedicadas	3Horas/semana
Total de semanas	14
Total de profissionais	2
<b>TOTAL</b>	<b>R\$4620.00</b>

Tabela 2 - Preços mão de obra

O sistema construído é formado por 3 etapas:

Primeiro, o usuário, utilizando o teclado insere a mensagem que deseja enviar, podendo ela ser uma mensagem de controle ou uma mensagem de texto.

A mensagem definida pelo usuário é interpretada pelo uC, e em tempo real aparece na tela LCD, quando o usuário envia a mensagem, o uC passa os caracteres para o modulo da antena, via serial, e a antena modula e transmite o sinal via AM.

A antena receptora recebe a mensagem, demodula e, via serial, passa para o uC, que decodifica a mensagem e, caso seja texto ele escreve no LCD, caso seja comando, ele faz o que foi requisitado.

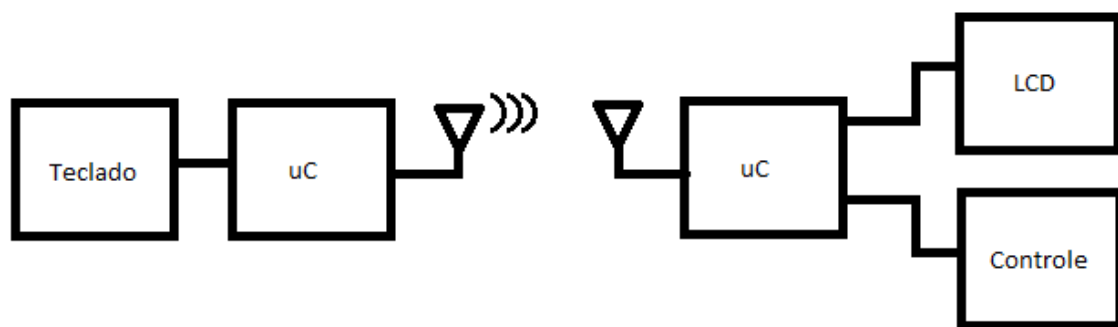


Figura 1 - Diagrama de blocos

O teclado utilizado é um teclado passivo, de 8 pinos, organizado matricialmente:

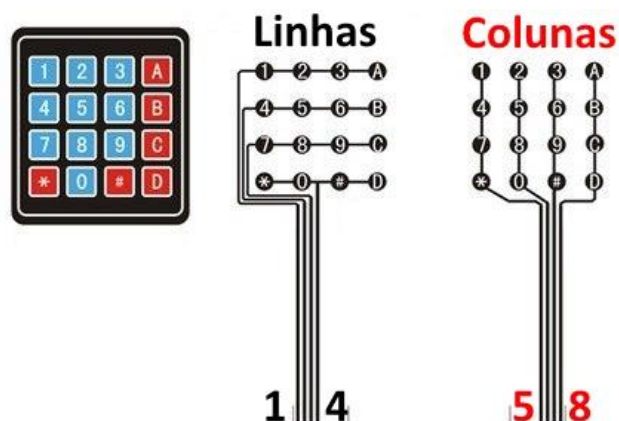


Figura 2 - Teclado 4x4 matricial

Para sua utilização, foi conectado ele em 8 pinos do Arduino sendo 4 pinos conectados como entrada e os outro 4 como saída, dessa forma, enviando um pulso de cada vez em cada uma das linhas e verificando a saída toda vez, é possível identificar qual botão está sendo apertado.

O LCD utilizado, segue uma lógica de comunicação paralela, possui 16 pinos de interface sendo eles:

Número	Pino	Função
1	GND	0V
2	VCC	5V
3	VO	Ajuste contraste
4	RS	Controle 1
5	Read/Write	Controle 2
6	Enable	Insere novos comandos

7	DB0	Data
8	DB1	Data
9	DB2	Data
10	DB3	Data
11	DB4	Data
12	DB5	Data
13	DB6	Data
14	DB7	Data
15	BLA	Contraste da letra +
16	BLK	Contraste da letra -

Tabela 3 - Pinos Teclado

Os pinos utilizados conectados foram os pinos Rs, Read/Write, Enable e DB0-7, totalizando 8 pinos.

O teclado funciona de maneira fácil, os dados são inseridos nas portas DB0-7 e, quando o pino Enable sobe, de acordo com as entradas nos Controles 1 e 2 ele interpreta o que deseja ser feito, conforme o *datasheet* informa.

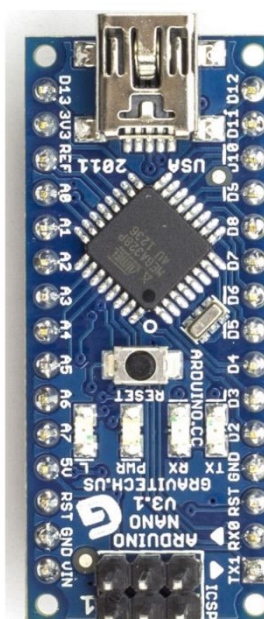
O modulo de antena utilizado são produzidos em pares, no qual um recebe a mensagem via serial, modula AM 433MHz e transmite, o outro recebe a mensagem, demodula e transmite via serial.

	Receptor	Transmissor
Modelo	MX-05V	MX-FS-03V
Alcance limite	/	20-200 metros (Tensão)
Tensão de operação	5 V	3,5-12 V
Modo de operação	AM	AM
Taxa de transferência	4 KB/s	4 KB/s
Potência (max)	20 mW	10 mW
Sensibilidades	-105dB	/
Frequência	315 MHz	315 MHz

Tabela 4 - Dados Antena

Por último, o micro controlador Arduino nano possui um Atmega328 que é mapeado da seguinte forma:

Microcontroller : ATmega168/ ATmega328  
 Operating Voltage : 5 V  
 Input Voltage : 7-12 V  
 Digital IO/PWM : 14 /6  
 Analog In/Out : 8  
 DC Current per I/O Pin : 40 mA  
 Flash Memory : 16 KB/32 KB  
 SRAM : 1 KB/2 KB  
 EEPROM : 512 bytes/1 KB  
 Clock Speed : 16 MHz



Arduino	Microcontroller
1(TX)	- PD1(TXD)
0(RX)	- PD0(RXD)
D2	- PD2(INT0)
D3	- PD3(INT1)
D4	- PD4
D5	- PD5
D6	- PD6
D7	- PD7
D8	- PB0
D9	- PB1
D10	- PB2(SS')
D11	- PB3(MOSI)
D12	- PB4(MISO)
D13	- PB5(SCK)
A0	- PC0
A1	- PC1
A2	- PC2
A3	- PC3
A4	- PC4(SDA)
A5	- PC5(SCL)
A6	- ADC6
A7	- ADC7

Figura 3 - Mapeamento Portas Arduino nano AtMega328

O Arduino conta com 14 portas digitais (Tx, Rx e D2-D13) e 8 portas analógicas (A0-A7). No projeto, ele foi conectado da seguinte maneira:

Porta Arduino	Porta Módulos
Tx(D1)	Antena transmissora
Rx(D0)	Antena receptora
D2~D9	Display DB0~DB7
D10	Display Read/Write
D11	Display RS
D12	Display Enable
A0~A3	Teclado In
A4~A7	Teclado Out

Tabela 5 - Mapeamento Arduino/Módulos

O mapeamento das portas analógicas para entrada digitais é feito sem problemas, apenas as portas A6 e A7 não contam com resistor de *pullup* e servem apenas de entrada, pois por construção são entradas ADC, portanto foi adicionado 2 resistores de *pullup* nas portas A6 e A7.

## 2.2CRONOGRAMA

26-Aug	Integração Arduino/antena	Ok
2-Sep	Integração Arduino/antena	Ok
9-Sep	Debugs e problemas	Ok
16-Sep	Integração Arduino/display	Ok

23-Sep	Integração Arduino/display – Entrega do relatório 1	Ok
30-Sep	Integração Arduino/teclado	Ok
7-Oct	Desenvolvimento de software teclado/display	Ok
14-Oct	Desenvolvimento de software antena/display	Ok
21-Oct	Debugs e problemas	Ok
28-Oct	Debugs e problemas	Ok
4-Nov	Debugs e problemas - Entrega do relatório 2	Ok
11-Nov	Prototipação	Ok
18-Nov	Prototipação	Ok
25-Nov	Apresentação - Entrega do relatório 3	Ok

Tabela 6 - Cronograma

Houve problemas imprevistos com a instalação do *driver* do Arduino no computador, necessitando o uso da primeira semana (2 de setembro) para resolver, assim, nos atrasando.

Durante a segunda semana (9 de setembro), foi começado o estudo da comunicação Arduino/Arduino, feita via serial e testado primeira versão (com fio, conectando a porta transmissora Tx de um na porta receptora Rx do outro). Nessa semana, foi transmitido uma mensagem, que futuramente será interpretada (via *software*) como comandos ou texto, que será mostrado no *display*.

Na terceira semana (16 setembro) foi estabelecido a conexão entre Arduino/antena antena/Arduino e reportado a necessidade do uso de uma antena externa devido à baixa potência de transmissão. Na comunicação foi observado, muito ruído na recepção do sinal (um pouco devido à falta de antena externa), uma das soluções que provavelmente serão usadas será a proteção na transmissão com redundância nas mensagens enviadas, afim de diferenciar mensagem de ruído.

Nas semanas do dia 23 e 30 de outubro foram construídos programas para a comunicação entre Arduino/teclado e nos dias 7 e 14 de novembro foi construído para Arduino/*display*. Foram criados duas classes (classe Tela – referente ao display e classe Teclado, referente ao teclado), nas quais continham funções para interagir com os periféricos.

A classe Teclado implementa as seguintes funções:

Teclado() -> Seu construtor, define uma matriz 4x4 numerada de 1 a 16, referente aos seus botões.

check() -> Varre as entradas de A0 a A3 com um pulso baixo e olha sua saída A4 a A7 em cada interação, até encontrar um pulso baixo e saber assim, qual botão está apertado (ou se nenhum está apertado).



A classe Tela implementa as seguintes funções:

Tela() -> seu construtor, garante pino Enable em zero, chama a função Reset() e a função displayOnOff().

enable() -> levanta o pino Enable, espera 10ms e o abaixa.

displayOnOff(bool displayOn, bool cursorOn, bool cursorBlink) -> seta configuração do display de acordo com as booleanas entradas.

reset() -> limpa a tela e reseta o cursor do display.

setData(String frase) -> Seta string de até 16 caracteres no display.

setByte(byte data) -> Seta um caractere na tela de acordo com o byte de entrada, seguindo a tabela ascii.

Nas semanas do dia 21 e 28 de novembro, foi corrigido alguns problemas e analisado alternativas para melhorar a comunicação com a antena.

### **3.UPDATE FINAL**

#### **3.1 TRANSMISSÃO SEM FIO**

Depois de muito tempo sem solução para nosso problema de transmissão sem fio, a qual apresentava muito ruído a ponto de impossibilitar uma comunicação com distâncias acima de 5cm, foi chegado à causa e a uma amenização deste problema.

O problema surge da má qualidade dos módulos de transmissão AM, simples assim, porém isto estava sendo descartado por nós devido à propaganda de desempenho irreal feita por revendedores.

A fim de amenizar o problema, fizemos uso de uma biblioteca de Arduino criada com o intuito de possibilitar a comunicação sem fio mesmo com módulos AM de baixo custo e desempenho, a “VirtualWire”. Os desenvolvedores da biblioteca alegam que esse módulos de transmissão sem fio de baixo custo necessitam dos dados especialmente formatados, com padrões de sincronização, balanço igual de bits de 0 e 1, e verificação de erros. A “VirtualWire” implementa estes “requisitos”, e com a utilização dela, de uma antena improvisada a partir de um fio e de uma alimentação de 9V, chegamos a garantia de transmissões até 15m, o que foi um aumento considerável.

### 3.2 TECLADO

Para o teclado, definiu-se uma lógica semelhante aos teclados de celulares antigos, da forma que cada tecla numérica tem um grupo de letras que são selecionadas com o clique repetido da tecla. A tecla “A” tem a função de mover o cursor para a próxima posição e a tecla “B” move o cursor para a posição passada, sendo usada como *backspace*. A tecla “D” envia o que estiver escrito no display.

### 3.3 DISPLAY

O display já estava funcionando de forma satisfatória, para estas semanas finais apenas implementamos a função *backspace* e garantimos que seja mostrada sempre as atualizações da mensagem a ser enviada quase que em tempo real.

### 3.4 FLUXOGRAMA DE OPERAÇÃO

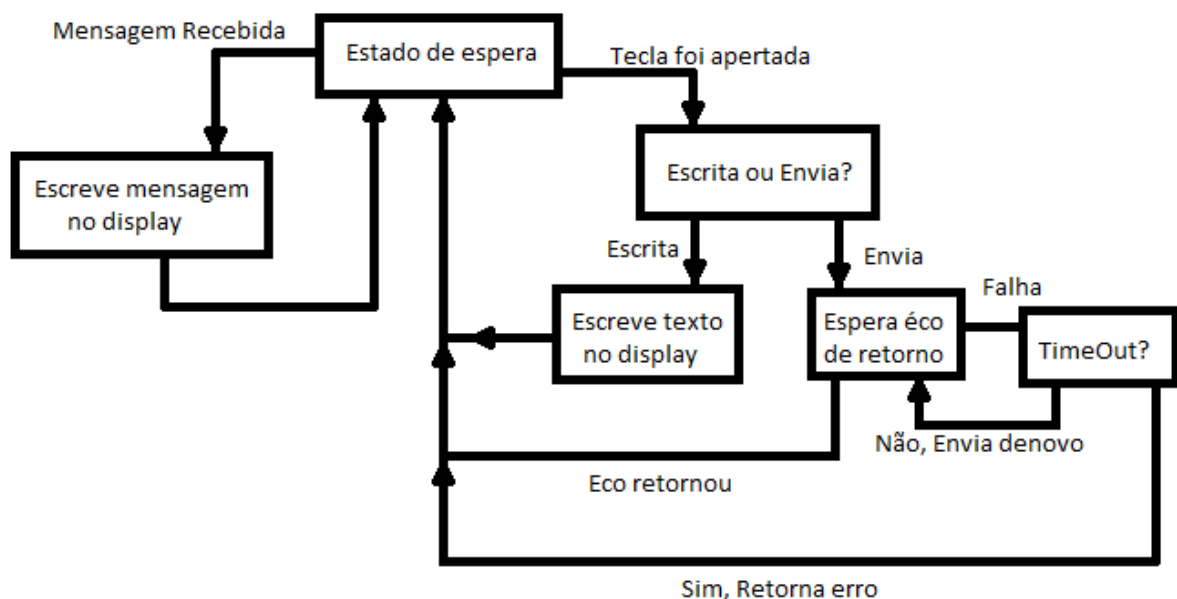


Figura 4- Diagrama de Operação

O fluxograma de operação se encontra no diagrama acima. No envio da mensagem, o transmissor espera uma mensagem igual a transmitida (eco), caso ele não receba, ele tenta a retransmissão até um tempo limite (*timeout*), por fim, se não receber, avisa na tela que a mensagem não foi enviada. O objetivo desse eco é a confiabilidade do envio da mensagem, pois com o aumento da distância, menor é a taxa de recebimento de dados e com isso, essa confirmação torna mais confiável a transmissão.

### 3.5 ANEXO DE FOTOS FINAIS

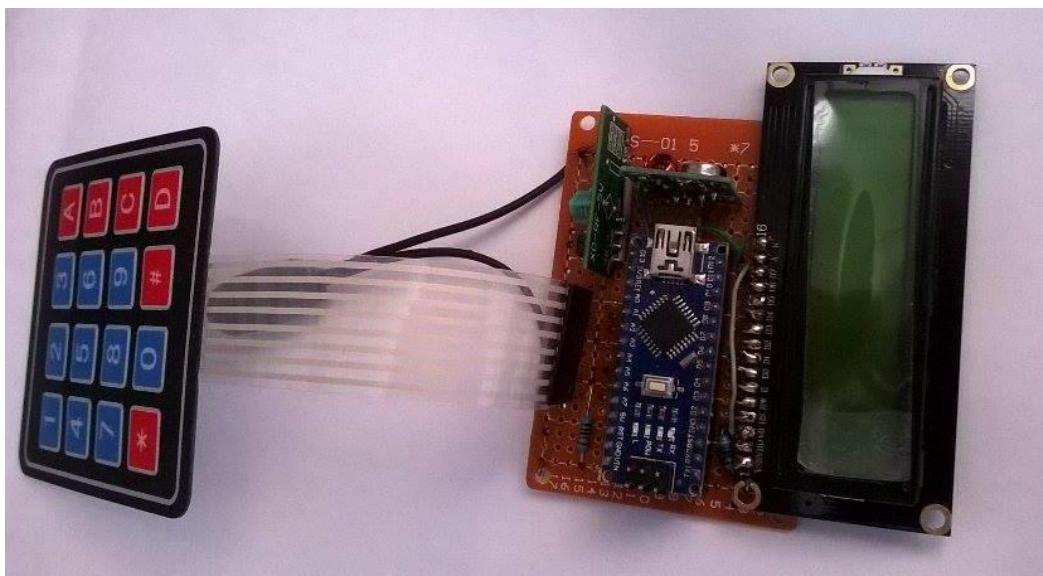


Figura 5 - Modulo 1 (PCB)

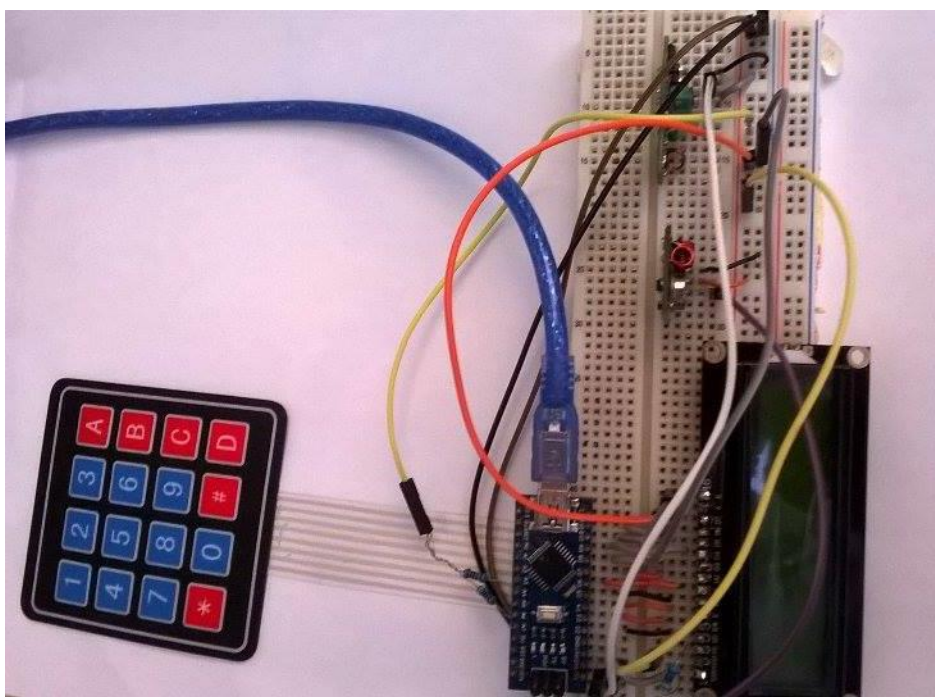


Figura 6 - Modulo 2 (PROTOBOARD)

#### **4. CONCLUSÕES**

Em relação à experiência de desenvolver o projeto, foi uma atividade bastante produtiva, pois foi possível aplicar na prática muitos conceitos teóricos e treinar nossa habilidade de resolução de problemas. Outro ponto a se comentar foi que o projeto chegou a um tamanho maior do que os projetos feitos no passado e a boa organização foi chave para conseguir realiza-lo.

Ao longo do semestre, sempre foi buscado estar em dia com o cronograma proposto, aconteceram imprevistos atrasando o calendário mesmo com algumas folgas consideradas, porém a meta foi cumprida.