

# PROPOSTA DE PROJETO: SISTEMA ELETRÔNICO DE ENVIO DE RESPOSTAS EM PROVAS OBJETIVAS

*Filipe Rodrigues da Silva*

Programa de Engenharia Eletrônica  
Microcontroladores e Microprocessadores  
Faculdade Gama - Universidade de Brasília  
email: filipebsb\_10@hotmail.com

*Jucelino Ferreira de Brito Júnior*

Programa de Engenharia Eletrônica  
Microcontroladores e Microprocessadores  
Faculdade Gama - Universidade de Brasília  
email: jucelino.brito.unb@gmail.com

## RESUMO

Este documento apresenta a proposta de projeto da disciplina de Microcontroladores e Microprocessadores do curso de Engenharia Eletrônica da Universidade de Brasília. O tema a ser abordado foi de categoria livre, assim a dupla decidiu por desenvolver um projeto voltado para a grande área de educação, mais especificamente voltado para sistemas de avaliações em provas objetivas, onde tem-se o intuito de construir um aparelho capaz de substituir o atual modelo de envio de respostas e correção de prova de forma a facilitar a correção, melhorar a segurança, e contribuir para a sustentabilidade com a economia no uso de papel.

## 1. JUSTIFICATIVA

Ao se observar a forma atual dos sistemas de avaliação, percebeu-se que seria possível aperfeiçoar o procedimento de envio das informações de forma a trazer uma série de benefícios, como maior segurança das informações, maior autonomia para deficientes visuais, maior agilidade na correção de provas e disponibilização de resultados e maior economia no uso de papel e material humano.

Em provas e concursos a segurança é, senão o mais importante, um dos mais importantes fatores do processo de avaliação. O fato de as respostas serem registradas em papel exige toda uma logística e pessoal para que não seja comprometida a lisura do processo, além de possibilitar a ocorrência problemas como extravio de cartões resposta e fraudes como troca de cartões respostas e compras de gabaritos. É inegável que o risco inerente ao fator humano existe, ainda mais quando se trabalha com equipes muito grandes onde várias pessoas têm acesso a esses dados. A utilização de um sistema único eletrônico para envio e recebimento desses dados de respostas eliminaria grande parte do risco no sentido de que não seria necessária quase nenhuma interferência humana, além de dispensar o transporte de cartões resposta, pois todas as respostas iriam quase que instantaneamente para uma central de dados.[1][2]

Ao se observar dados referentes a vários concursos passados, observa-se que o tempo médio entre a realização das provas objetivas e os resultados preliminares demoram entre um e dois meses, em alguns casos levando até mais tempo. Isso provoca todo um desgaste de candidatos além de penalizar o órgão ou empresa que necessita o mais rápido possível do resultado para que se possa realizar as contratações de mão de obra. Além disso, em outro aspecto que não o de provas de concursos, a correção de provas por parte de professores dos ensino médio e superior toma uma boa parte do tempo, assim a utilização de um meio eletrônico para realizar esse processo possibilitaria uma correção instantânea e uma economia de tempo enorme.[3]

Referente ao aspecto de acessibilidade, atualmente a entidade que organiza um concurso público precisa atentar para as particularidades do candidato com deficiência visual, já que ele é o hipossuficiente da relação. Assim, não lhe cabe apenas zelar pelo registro confiável e fidedigno do exame, mas também provar que forneceu ao candidato o que estava previsto no edital e o prometido em termos de acessibilidade e condições especiais na hora da inscrição. Um sistema eletrônico de registro e envio de respostas possibilitaria uma maior dependência e autonomia por parte de candidatos com deficiências visuais.[4]

No tocante a sustentabilidade tem-se dados relevantes a serem considerados. Considerando a folha A4 160gsm que pesa 10g, pode-se fazer uma estimativa básica a respeito dos gastos realizados com cartões resposta em concursos pelo Brasil.[5]

Tomando por base o maior exame por provas do Brasil, o ENEM, pode-se observar o quantitativo do número de inscritos dos últimos quatro anos na tabela 1 abaixo. [6]

ANO	Nº INSCRITOS
2016	9.276.328
2015	8.478.096
2014	9.519.827
2013	7.173.574
Total	34.447.825

Tabela 1- Inscritos ENEM

Considerando que foram impressos cartões resposta em igual número de total de inscritos, foram utilizados 344.478,250 kg de papel. Para se ter uma ideia desse número, para produzir 50 kg de papel é necessária uma árvore inteira, logo somente para a impressão de cartões resposta das últimas quatro edições do ENEM foram necessárias aproximadamente 7 (sete) mil árvores. Além disso para cada folha A4 comum são necessários 10 litros de água, o que nesse exemplo contabiliza um gasto total de 344.478.825 litros de água.[7][8]

Todos esses dados foram apresentados levando em conta apenas quatro edições de uma prova, mas deve-se enfatizar que todos os anos ocorrem milhares de concursos por todo país, além de provas em instituições de educação de ensino médio e superior, vestibulares, e muitas outras provas que poderiam utilizar um meio mais sustentável para o registro de suas respostas e correção de provas.

## **2. OBJETIVOS**

- Utilização de teclado matricial para o envio de respostas através da MSP430;
- Utilizar um método de comunicação sem fio entre MSP430;
- Desenvolvimento de uma estratégia de segurança dos dados enviados;
- Recebimento dos dados transmitidos por uma segunda MSP430 e organização dos dados em uma tabela com a respectiva identificação do remetente das informações.

## **3. REQUISITOS**

- Teclado Alfanumérico matricial;
- Tela de cristal líquido para visualização dos dados inseridos;
- Sistema de comunicação sem fio entre MSPs;
- Um sistema de alimentação por baterias, para que não seja necessário utilizar uma fonte de energia externa;
- Recebimento dos dados e organização em uma tabela para apresentação.

## **4. BENEFÍCIOS**

O projeto possibilitará a melhora no processo de transmissão e recebimento de dados apresentando uma nova técnica de troca de informações, visando agilizar o processo de recebimento e correção de provas, dar maior praticidade, segurança e economia de recursos, além do cunho sustentável no que se refere a diminuição do uso de papel.

## **5. ESTADO DA ARTE**

Embora tenha sido realizado pesquisas via internet, não foi encontrada nenhuma fonte contendo informações que dessem suporte ao tema proposto.

## **6. INTRODUÇÃO**

A implementação do projeto na etapa inicial consistiu na utilização de duas placas MSP430 da família G2553 e de dois módulos wifi NRF24L01.

A família de microcontroladores de baixo consumo da Texas Instruments MSP430 é constituída por vários dispositivos com diferentes conjuntos de periféricos direcionados para várias aplicações. A arquitetura, combinada com cinco modos de baixo consumo de energia, é otimizada para aumentar a vida da bateria em aplicações de medição portáteis. O dispositivo possui uma CPU RISC de 16 bits, registradores de 16 bits e geradores constantes que contribuem para a máxima eficiência do código. O oscilador controlado digitalmente (DCO) permite o despertar dos modos de baixa potência para o modo ativo em menos de 1  $\mu$ s. [9]

A série MSP430G2353 é um microcontrolador de sinal misto de potência ultra baixa com temporizadores embutidos de 16 bits, até 24 pinos ativados capacitivo de E / S, um comparador analógico versátil e capacidade de comunicação embutida usando o serial universal Interface de comunicação. Além disso, os membros da família MSP430G2553 têm um conversor de 10 bits analógico-digital (A / D). [10]

## **7. DESENVOLVIMENTO**

Em cada microcontrolador foi ligado um módulo NRF24L01. Este módulo utiliza o chip NRF24L01 fabricado pela Nordic. Trabalha na frequência de 2.4 Ghz e possui uma antena embutida, o que faz desse um dos módulos mais compactos do mercado. Possui um conector de 10 pinos, com os pinos muito próximos uns dos outros.

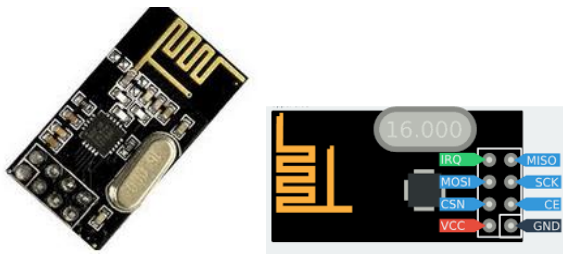


Figura 1: Módulo Wifi nRF24L01 da Nordic Semicondutores

A conexão entre a MSP430 e o módulo NRF24L01 foi realizada utilizando jumpers macho-fêmea e fêmea-fêmea. A figura 2 e a tabela 1 apresentam as conexões entre os componentes.

nRF24L01	MSP430G2353
GND	GND
VCC	VCC
SCK	P1.5
MOSI	P1.7
MISO	P1.6
CE	P2.0
CSN	P2.1
IRQ	P2.2

Tabela 2: Conexões entre os pinos do módulo wifi e da MSP430G2353

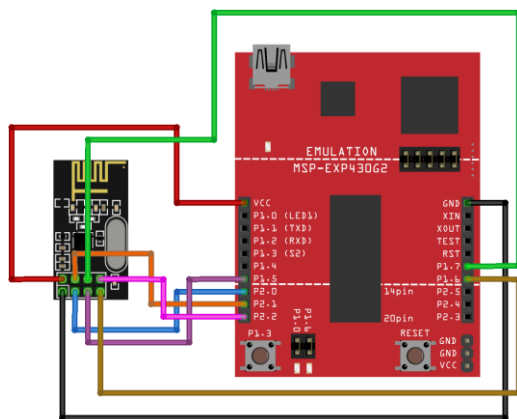


Figura 2: Desenho ilustrativo das conexões realizadas

O objetivo dessa etapa inicial consistia em obter uma forma consistente de comunicação entre as duas placas MSP, pois isso se trata da espinha dorsal do projeto. Assim caso fosse obtido êxito nessa etapa, o projeto poderia ter continuidade.

A comunicação wireless entre as placas foi verificada através de dois códigos programados na interface ENERGIA. O primeiro código se refere ao transmissor e o segundo código se refere ao receptor.

A ideia do teste foi realizar uma tarefa que pudesse demonstrar a viabilidade do projeto como um todo. Para isso, foi montado um circuito externo com dois botões (push-buttons) e dois resistores de pull-down.

Basicamente o circuito funciona da seguinte forma:

- Um botão representa a informação que se deseja enviar;
- Cada vez que o este botão da informação é pressionado, um contador é incrementado;
- Esse incremento é exibido na porta serial do circuito transmissor;
- Ao se atingir o número que se deseja (a informação a ser enviada), pressiona-se o segundo botão;
- Esse segundo botão é utilizado exclusivamente para enviar a informação desejada. Ou seja, nenhuma informação será enviada enquanto esse botão não for pressionado;
- Ao se enviar a informação, o contador é zerado, podendo assim ser definida nova informação para ser enviada;
- Pode-se observar a informação recebida no receptor através da porta serial desse.

Para a concluir este projeto seriam necessários mais dois componentes: um display LCD QAPSS 1602, figura 1, e um teclado matricial, figura 2.

Mas foi encontrado um empecilho neste ponto. Para a conexão do teclado matricial são necessários 8 pinos das portas do MSP, para a conexão do LCD são necessários 6 pinos além dos 6 pinos necessários para conexão do módulo Wifi. A MSP utilizada possui 14 pinos disponíveis, sendo necessário no mínimo 28 pinos.

Para solucionar o problema encontrado, foi utilizada uma segunda MSP no módulo transmissor, onde a primeira MSP

recebe os dados do teclado matricial e envia esses dados para a segunda MSP via UART que mostrará os valores digitados no LCD e fará o envio dos dados para o modulo receptor.



Figura 3 - Microcontrolador MSP430 e Display LCD.

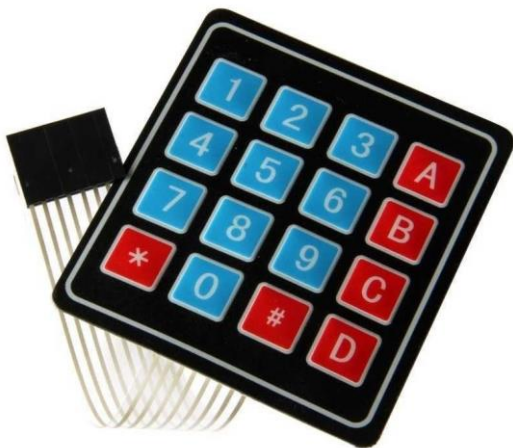


Figura 2 - Teclado Matricial 4x4.

As partes referentes a multiplexação do teclado e ao display LCD foram desenvolvidas separadamente para serem integradas posteriormente ao modulo de transmissão.

A multiplexação do teclado foi feita utilizando os pinos da porta P1 da seguinte forma:

PORTA MSP430G2553	TECLADO MATRICIAL
P1.0	Linha 1
P1.1	Linha 2
P1.2	Linha 3
P1.3	Linha 4
P1.4	Coluna 1
P1.5	Coluna 2
P1.6	Coluna 3
P1.7	Coluna 4

Tabela 3: Conexões teclado matricial

```
1
2 #include "msp430.h"
3
4 unsigned char leitura(void)
5 {
6     char letra = 0xFF;
7
8     P1OUT = BIT0; //linha1 = 1
9
10    if((P1IN&BIT4)!=0) letra = '1';
11    if((P1IN&BIT5)!=0) letra = '2';
12    if((P1IN&BIT6)!=0) letra = '3';
13    if((P1IN&BIT7)!=0) letra = 'A';
14
15    P1OUT = BIT1; //linha2 = 1
16
17    if((P1IN&BIT4)!=0) letra = '4';
18    if((P1IN&BIT5)!=0) letra = '5';
19    if((P1IN&BIT6)!=0) letra = '6';
20    if((P1IN&BIT7)!=0) letra = 'B';
21
22    P1OUT = BIT2; //linha3 = 1
23
24    if((P1IN&BIT4)!=0) letra = '7';
25    if((P1IN&BIT5)!=0) letra = '8';
26    if((P1IN&BIT6)!=0) letra = '9';
27    if((P1IN&BIT7)!=0) letra = 'C';
28
29    P1OUT = BIT3; //linha4 = 1
30
31    if((P1IN&BIT4)!=0) letra = '*';
32    if((P1IN&BIT5)!=0) letra = '0';
33    if((P1IN&BIT6)!=0) letra = '#';
34    if((P1IN&BIT7)!=0) letra = 'D';
35
36    return letra;
37 }
```

Figura 4 – Código referente a multiplexação do teclado

```

39 unsigned char espera_leitura(void)
40 {
41     /
42     unsigned char tecla_p = 0xFF;
43     //Esperar até que uma tecla seja pressionada
44     do {
45         tecla_p = leitura();
46     }
47     while (tecla_p == 0xFF);
48
49     return tecla_p;
50 }

```

Figura 5 – Código do loop para leitura do teclado

```

54 void main(void)
55 {
56     WDTCTL = WDTPW + WDTHOLD;
57
58     unsigned char tecla_pressionada;
59     PIDIR = BIT0+BIT1+BIT2+BIT3; //configuração de entradas e saídas
60     while(1){
61         tecla_pressionada = espera_leitura();
62         if (tecla_pressionada == '1')
63             P1OUT = BIT0;
64     }
65 }

```

Figura 6 – Código principal para teste. Caso a tecla 1 fosse pressionada o LED ligado a porta P1.0 acendia.

O Display Lcd16x2 QAPASS é um objeto retangular que imprime caracteres em uma tela de lcd. Ele possui as nomeações da pinagem na placa. Isso ajuda a identificar qual é o pino.

8.

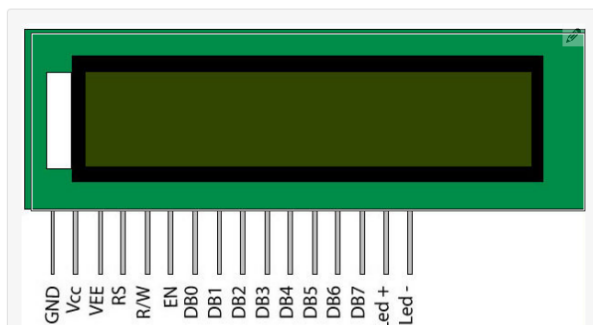


Figura 7 - Pinagem do Display Lcd QAPASS 1602.

O LCD possui uma voltagem de trabalho na ordem de 5V. Assim, a primeira observação importante se dá quando varia a tensão de entrada. Com isso se obtém diferentes valores de tensão da rede de divisão de tensão. E assim, pode se alterar o contraste para atender às suas necessidades visuais. Para que não ocorra uma tensão que queime a entrada do componente é necessário inserir um resistor que possa equilibrar essa tensão.

9.

Os pinos RS, R / W, E são os sinais de controle do LCD. DB0 para DB7 são as linhas de dados. Onde se envia a

palavra de comando, bem como os dados a serem escritos neste caminho de dados.

O RS significa seleção do registro. Os dois registradores em LCD são o registro de dados e o registro de palavra / código de comando. Para dizer ao LCD que os bits no databus são para qual registrador, usa-se o sinal de controle RS através do pino RS. Quando torna-se este pino alto, seleciona-se o registro de dados, onde enviará os valores ASCII para serem exibidos na tela. Quando RS baixo, seleciona-se o registro de palavras do comando onde se enviará todos os dados para configurar e inicializar o LCD. Com isso pode-se especificar que quando RS = 1 implica em Registro de dados. Quando RS = 0 há Código de comando no Registrador.

A nomeação R / W significa ler ou escrever. A leitura é o sinal alto ativo e a gravação está ativa baixa. Assim, quando você deseja ler no LCD, você faz o sinal neste pino alto, e quando quiser escrever, você faz o sinal no pino, baixo. Assim, R / W igual a 1 há uma operação de leitura. E quando há R / W = 0 há operação de gravação.

A letra E significa habilitação. Este é um sinal de disparo de borda que é usado enquanto escreve ou lê dados do microcontrolador para LCD, respectivamente. Quando E é igual ao limite alto a baixo há uma operação de Escrita. Quando E é igual baixo para alto ou borda positiva há leitura.

O processador interno do LCD leva tempo para trancar e fazer os ajustes necessários de acordo com a palavra de comando. Enquanto o processador interno do LCD estiver ocupado, esse sinalizador é definido. Portanto, um deve verificar o status deste sinalizador antes de enviar a próxima palavra ou dados do comando. D7 é o pino da bandeira ocupada. É preciso configurar o pino de porta conectado ao pino D7 como entrada ao verificar a condição da bandeira. Juntamente com isso, precisa-se fazer RS igual 0 e R / W igual 1, uma vez que esta operação é lida e a bandeira de ocupado é fornecida pelo modo de registro de código de comando.

Pode-se dar atrasos também para o LCD para terminar o trabalho. Mas esta é uma maneira melhor se tiver pinos de porta suficientes. Porque para ler o status da bandeira ocupada, você precisa do sinal R / W e, portanto, um pino de porta.

A lista de instruções do LCD pode ser observada na figura a seguir.

LCD Command Codes	
Code (Hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning to 1st line
C0	Force cursor to beginning to 2nd line
38	2 lines and 5x7 matrix

Figura 8. Lista de comandos do LCD QAPASS.

O hardware de conexão que foi utilizado foi o modo de 4bit (dados entre D7-D4 apenas) para a comunicação com o display LCD, pois não se tem pinos suficientes para o MSP430 para comunicação de 8 bits.

Segue abaixo uma tabela que relaciona a pinagem do microcontrolador MSP430 com o display LCD QAPASS 1602.

Pinagem MSP430	Pinagem Display LCD
TP1	Vcc (+5V)
TP3	Vss (Gnd)
P2.0	D4
P2.1	D5
P2.2	D6
P2.3	D7
P2.4	Enable(N)
P2.5	RS
Gnd	Rw
Gnd	Vee/Vdd
Gnd	K (LED -)
Vcc	A (LED +)

Tabela 4: Pinagem correspondentes entre MSP430 e LCD.

Para teste do funcionamento do display e do Código foram feitas as ligações descritas na tabela acima, e foi impressa no display a mensagem “Gabarito Eletrônico”, fazendo uma referência ao nome do projeto. Essa imagem pode ser vista abaixo.



Figura 9 - Montagem do Display com a MSP430.

O Código do display segue nas figuras abaixo.

```

4 // uC Port definitions
5 #define lcd_port      P2OUT
6 #define lcd_port_dir  P2DIR
7
8 // LCD Registers masks based on pin to which it is connected
9 #define LCD_EN        BIT4
10 #define LCD_RS        BIT5
11
12 void lcd_reset()
13 {
14     lcd_port_dir = 0xff;    // output mode
15     lcd_port = 0xff;
16     delay_cycles(20000);
17     lcd_port = 0x03+LCD_EN;
18     lcd_port = 0x03;
19     delay_cycles(10000);
20     lcd_port = 0x03+LCD_EN;
21     lcd_port = 0x03;
22     delay_cycles(1000);
23     lcd_port = 0x03+LCD_EN;
24     lcd_port = 0x03;
25     delay_cycles(1000);
26     lcd_port = 0x02+LCD_EN;
27     lcd_port = 0x02;
28     delay_cycles(1000);
29 }

```

Figura 10 – Código Display Parte 1

A primeira parte do código foi para definir a porta P2 como saída. A segunda parte foi realizada as máscaras para o Lcd nos pinos nos quais ele está conectado.

Logo em seguida , foram realizadas 5 funções para o envio de dados do MSP430 para o LCD. As quatro funções void lcd\_reset( ), void lcd\_cmd( char cmd), void lcd\_init( ) void lcd\_data( unsigned char dat) e void display\_line( char \*line) que realizam o modo de saída do display, o atraso no envio de dados, o reset de chamada do Lcd , o endereçamento dos dados enviados pelo MSP430 e um incremento nos dados em que se quer enviar para a tela.



```

31 void lcd_cmd (char cmd)
32 {
33     // Send upper nibble
34     lcd_port = ((cmd >> 4) & 0x0F) | LCD_EN;
35     lcd_port = ((cmd >> 4) & 0x0F);
36
37     // Send lower nibble
38     lcd_port = (cmd & 0x0F) | LCD_EN;
39     lcd_port = (cmd & 0x0F);
40
41     _delay_cycles(4000);
42 }
43
44 void lcd_init ()
45 {
46     lcd_reset();           // Call LCD reset
47     lcd_cmd(0x28);         // 4-bit mode - 2 line - 5x7 font.
48     lcd_cmd(0x0C);         // Display no cursor - no blink.
49     lcd_cmd(0x06);         // Automatic Increment - No Display shift.
50     lcd_cmd(0x80);         // Address DDRAM with 0 offset 80h.
51     lcd_cmd(0x01);         // Clear screen
52 }
53
54
55 void lcd_data (unsigned char dat)
56 {
57     // Send upper nibble
58     lcd_port = (((dat >> 4) & 0x0F) | LCD_EN | LCD_RS);
59     lcd_port = (((dat >> 4) & 0x0F) | LCD_RS);
60
61     // Send lower nibble
62     lcd_port = ((dat & 0x0F) | LCD_EN | LCD_RS);
63     lcd_port = ((dat & 0x0F) | LCD_RS);
64
65     _delay_cycles(4000); // a small delay may result in missing char display
66 }

```

Figura 11 – Código Display Parte 2

```

68 void display_line(char *line)
69 {
70     while (*line)
71         lcd_data(*line++);
72 }

```

Figura 12 – Código Display Parte 3

Por fim o Código da figura 12 é a função que envia os dados a serem impressos no display.

## 10. RESULTADOS

Na etapa inicial os resultados obtidos satisfatórios. O objetivo foi alcançado. Ocorreram alguns problemas no início na configuração do código para o envio dos dados, mas foi solucionado. Nessa etapa foi concluído o módulo referente a transmissão e recepção dos dados de uma MSP para outra utilizando o protocolo SPI.

Neste segundo ponto, o objetivo era completar os módulos restantes para a junção de todos e unificação do projeto final. Tais módulos restantes eram o teclado matricial e o display LCD. Conforme observado no desenvolvimento acima, obteve-se sucesso na implementação de ambos os módulos separadamente. Ocorreram alguns problemas de início como a dificuldade para implementação do um código para o display de LCD e a multiplexação do teclado matricial, mas os problemas foram resolvidos, pelo menos em partes.

A integração entre o teclado e o display de LCD funcionou parcialmente. O que foi conseguido é que o que é digitado no teclado é mostrado no display, mas ainda está ocorrendo alguns erros na leitura do teclado, pois em alguns momentos aparecem caracteres no display mesmo sem ser pressionada alguma tecla. Para a conclusão do projeto resta sanar este erro e resolver o que fazer a respeito da pinagem do MSP.

O problema quanto a quantidade de pinos do MSP continua, pois o módulo transmissor exige bem mais pinos do que ela comporta, assim foram pensadas algumas soluções, entre elas a utilização de duas placas MSP no módulo transmissor e uma comunicação UART entre elas, ou a utilização de um barramento SPI – SÉRIE para o display LCD ocupar menos pinos.

## 11. CONCLUSÕES

De fato é possível dar continuidade ao projeto. O principal ponto era a comunicação entre as duas placas MSP430, e isso foi obtido com êxito. Mas agora vêm uma série de outros desafios para que o projeto funcione corretamente conforme proposto.

A mensagem que deverá ser enviada consistirá de uma matriz de duas colunas e linhas conforme o número de questões, dessa forma será necessário utilizar uma estratégia para alocação dessas informações em posições específicas de um vetor, pois o módulo wifi pode transmitir apenas mensagens na forma de um vetor.

A implementação dos códigos do teclado matricial e do display LCD na MSP foi um grande avanço no projeto, mesmo que tenha funcionado de forma modular (separadamente), bastando agora juntar os dois códigos e buscar uma harmonia entre eles para que funcione corretamente.

Outro avanço importante no projeto é o fato de passar a ser utilizado o Code Composer ao invés do Energia. Isso deixou o projeto mais robusto e profissional.

## 12. REFERÊNCIAS

- [1] Exército vai reaplicar provas de concurso após extravio de cartões-resposta. Disponível em: <  
<http://extra.globo.com/emprego/concursos/exercito-vai-reaplicar-provas-de-concurso-apos-extravio-de-cartoes-resposta-14092967.html>>.

- [2] Extravio de malote com cartões resposta anula concurso público do santa. Disponível em: <<http://www.jornalboavista.com.br/site/noticia/30484/extravio-de-malote-com-cartoes-respostas-anula-concurso-publico-do-santa>>.
- [3] Professores gastam 894 mil dias só com correção de provas. Disponível em: <<http://jornalggn.com.br/forapauta/professores-gastam-894-mil-dias-so-com-correcao-de-provas>>.
- [4] O atendimento diferencial no ENEM. Disponível em: <[http://download.inep.gov.br/educacao\\_basica/enem/nota\\_tecnica/2012/atendimento\\_diferenciado\\_enem\\_2012.pdf](http://download.inep.gov.br/educacao_basica/enem/nota_tecnica/2012/atendimento_diferenciado_enem_2012.pdf)>.
- [5] Pesos de papel. Disponível em: <<http://www.tamanhosdepapel.com/pesos-de-papel.htm>>.
- [6] Comitê de Estatísticas Sociais: Exame Nacional do Ensino Médio – ENEM. Disponível em: <<http://ces.ibge.gov.br/base-de-dados/metadados/inep/exame-nacional-do-ensino-medio-enem.html>>.
- [7] Produção de folha de papel A4 necessita de 10 litros de água. Disponível em: <<http://www.painelflorestal.com.br/noticias/celulose-e-papel/producao-de-folha-de-papel-a4-necessita-de-10-litros-de-agua>>.
- [8] Coleta Seletiva. Disponível em: <<http://mundodaterrazul.blogspot.com.br/2009/11/coleta-seletiva.html>>.
- [9] MSP430x2xx Family User's Guide. Disponível em <[www.ti.com/lit/ug/slau144j/slau144j.pdf](http://www.ti.com/lit/ug/slau144j/slau144j.pdf)>.
- [10] Mixed Signal Microcontroller – Datasheet. Disponível em <<http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=msp430g2513&fileType=pdf>>.