

## SISTEMA DE AUTOMAÇÃO RESIDENCIAL VIA REDE CELULAR USANDO MICROCONTROLADORES E SENSORES

Luis Filipe Campos Cardoso <sup>1</sup>

<sup>1</sup>*Graduando em Engenharia de Redes de Comunicação  
Departamento de Engenharia Elétrica da Faculdade de Tecnologia da Universidade de Brasília,  
Campus Darcy Ribeiro, Asa Norte, 70910-900, Brasília, DF, Brasil*

**Palavras-chave:** Arduino, Shield, GSM, SMS, Sensor, LDR, Transdutor, Corrente Elétrica.

**Resumo:** O presente trabalho apresenta uma aplicação na área de Telecomunicações e Automação. A ideia foi obter uma comunicação entre um aparelho de telefonia móvel e um sistema automatizado, suportado pela plataforma *Arduino* com a utilização da tecnologia GSM (*Global System for Mobile Communications* - Sistema Global para Comunicações Móveis), por meio de SMS (*Short Message Service* - Serviço de Mensagens Curtas). Neste trabalho, foi explorado o ambiente residencial e controle de gastos com energia, utilizando sensores como suporte para elaboração de programas, os quais pudessem oferecer um serviço doméstico.

**Keywords:** Arduino, Shield, GSM, SMS, Sensor, LDR, Transducer, Electric Current.

**Abstract:** This paper presents an application in the field of Telecommunications and Automation. The idea was to create a communication between a mobile device and an automated system, supported by *Arduino* platform using GSM (*Global System for Mobile Communications*) technologies, including SMS (*Short Message Service*). In this study, we explored the residential environment and energy control, using sensors that give support for development of software, which could offer a domestic service.

---

E-mail: luisfcardoso@ieee.org.

## 1 INTRODUÇÃO

Uma preocupação recorrente da tecnologia tem sido o desenvolvimento de sistemas inteligentes e automatizados com o intuito de facilitar a vida das pessoas. E, nesse contexto, há cada vez mais projetos que buscam trazer essa automação para dentro de suas casas.

Com a crescente popularização dos aparelhos celulares, surgiu a ideia de tornar móvel a automação residencial através de uma integração com um celular. Muitos projetos ligados a isso podem surgir com o intuito de solucionar problemas ou de trazer praticidade e comodidade ao cotidiano.



Figura 1: Automação Residencial

O que motivou este trabalho foi à busca de um sistema inteligente que pudesse realizar tarefas dentro de residências, como é o caso do controle de gastos de energia elétrica. Pensou-se que poderia ser bastante prático se algum sistema pudesse detectar – dentro de um nível predeterminado de consumo ideal – se a tensão não corresponde ao nível desejável (se está acima ou abaixo de tal nível). Ou até mesmo, utilizar sensores de luminosidade para informar a presença de luz nos ambientes. Assim, o principal objetivo deste trabalho é utilizar sensores para a construção de redes que possam trazer alguma utilidade no ambiente doméstico.

Foi implementada a tecnologia GSM para a realização do trabalho. Ela pode ser entendida como uma rede celular de segunda geração e foi originalmente proposta para resolver o problema das redes celulares, como uma forma de unificação dos sistemas na Europa.

O sucesso do padrão GSM excedeu as expectativas e ele é atualmente o padrão mais popular para sistemas celulares e equipamentos de comunicação pessoal em todo o mundo [1]. Esse padrão utiliza a FDM (Frequency Division Multiplexing - Multiplexação por Divisão de Frequência) e a TDM (Time Division Multiplexing - Multiplexação por Divisão de Tempo) para a interface de ar [2].

A vinda do GSM possibilitou a inserção de novos serviços, como o SMS. Foi uma ideia proposta para utilização de canais ociosos para transmissão de mensagens de texto de 140 bytes - 160 dígitos de 7 bits, 128 bytes em sua origem - a baixo custo. Esse serviço se tornou muito popular, principalmente entre os jovens.

## MATERIAIS E MÉTODOS

Na contemporaneidade, houve um grande aumento nos projetos de tecnologias livres, que são fortemente encorajados por comunidades da internet, onde a cooperação mútua é a principal ferramenta para a construção de novos sistemas e a solução de problemas dos mesmos.

Um bom exemplo é o projeto *Arduino* (veja figura 2), criado na Itália em 2005. Tem como

objetivo oferecer uma plataforma de prototipagem eletrônica *open-source* (código aberto), onde a documentação para elaboração da placa eletrônica é disponibilizada para os usuários em seus sítios de projeto. O projeto *Arduino* disponibiliza para download o IDE (*Integrated Development Environment* - Ambiente de Desenvolvimento Integrado) para a programação da placa eletrônica, fornecendo ainda o código fonte deste ambiente. O IDE do *Arduino* pode rodar nas plataformas *Windows* e *Linux*, e ainda, possui uma linguagem de programação baseada no projeto *Wiring* [3, 4]. Além disso, essa plataforma possui um baixo custo.

O *Arduino* utilizado no projeto foi o modelo *Uno R3*. Ele é uma placa de microcontrolador baseado no *ATmega328*. Possui 14 entradas/saídas digitais (dos quais seis dessas podem ser usadas como saídas PWM (*Pulse-Width Modulation* - Modulação por largura de pulso)), 6 entradas analógicas, uma conexão *USB*, um conector de alimentação e um botão de reset. Para comunicação, utiliza o protocolo *ICSP* (*In Circuit Serial Programming* – Programação Serial em Circuito).



Figura 2: *Arduino UNO R3*

O *Arduino UNO* pode ser alimentado através da conexão *USB* ou com uma fonte de alimentação externa, em particular, no trabalho utilizou-se uma fonte de 9V.

Tabela 1: Especificações do *Arduino UNO R3*

<b>Microcontrolador</b>	<i>ATmega328</i>
<b>Tensão de funcionamento</b>	5V
<b>Tensão de entrada (recomendado)</b>	7-12V
<b>Tensão de entrada (limite)</b>	6-20V
<b>Entradas digitais de I / O</b>	14
<b>Entradas analógicas</b>	6
<b>Corrente DC para pino I / O</b>	40 mA
<b>Corrente DC para Pin 3.3V</b>	50 mA
<b>Memória Flash</b>	32 KB ( <i>ATmega328</i> ), dos quais 0,5 KB utilizado para gerenciamento de boot
<b>SRAM</b>	2 KB ( <i>ATmega328</i> )
<b>EEPROM</b>	1 KB ( <i>ATmega328</i> )
<b>Velocidade do relógio</b>	16 MHz

Os pinos de energia são os seguintes:

- 5V - Este pino gera um 5V regulado. A placa pode ser alimentado com energia a partir da tomada CC (Corrente Contínua) de 7 a 12V ou do conector *USB* (*Universal Serial Bus* - Barramento Serial Universal) de 5V.
- 3,3V - Gerado pelo regulador de bordo. Sua corrente máxima é de 50 mA.
- GND - Terra.
- IOREF - Este pino na placa *Arduino* fornece a referência de tensão com que o microcontrolador opera. Um escudo configurado corretamente pode ler a tensão do

pino IOREF e selecionar a fonte de alimentação adequada ou habilitar tradutores de tensão nas saídas para o trabalho com a 5V ou 3,3V.

A IDE do *Arduino* (figura 3) é bem simples. Uma função bem interessante é o *Serial Monitor*. É usado como uma forma de comunicação entre a placa e o computador, depois de compilado e enviado o programa.

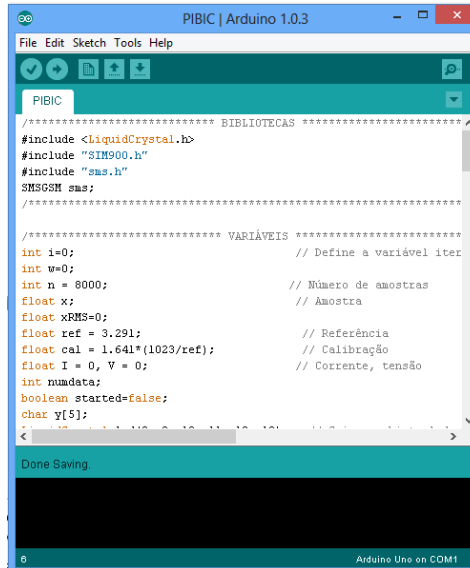


Figura 3: IDE do *Arduino*

As *shields* são módulos expansivos que agregam uma nova função ao microcontrolador. No projeto foi utilizado a *IComsat* versão 1.0 (veja na figura 4) que, por meio do módulo *SIM900 Quad-band*, tem como objetivo estender as funcionalidades do UNO para o uso da tecnologia GSM, para envio e recebimento de SMS, por exemplo. O módulo SIM900 foi fabricado pela empresa Sim Com.

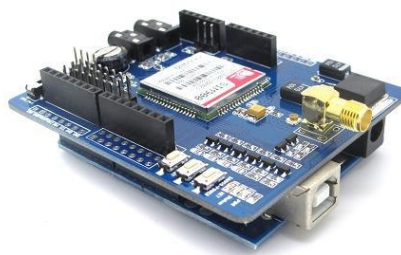


Figura 4: *Arduino UNO R3* com a *Shield IComsat* versão 1.0

Tabela 2: Especificações da *Shield GSM*

<b>Dimensões</b>	77.2mm X 66.0mm X 1.6mm
<b>Indicadores luminosos</b>	PWR, status, net status
<b>Alimentação</b>	9~20V, compatível com <i>Arduino</i>
<b>Protocolo de comunicação</b>	UART
<b>RoHS (Restriction of Certain Hazardous Substances - Restrição de Certas Substâncias Perigosas).</b>	Sim

Para fazer a instalação, é importante escolher a porta do TX (transmissor) e do RX

(receptor) utilizando os *jumpers*, na figura 5 são representados pelo retângulo azul. No trabalho, utilizamos a porta 4 para TX e 5 para RX, conforme a mesma figura.

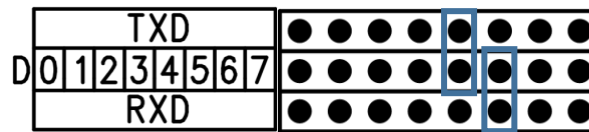


Figura 5: Multiplexador UART para escolha do TX e do RX

O primeiro ensaio realizado foi o envio de mensagem SMS do valor de corrente elétrica coletada por um sensor LDR (*Light Dependent Resistor* – Resistor Dependente de Luz). Como uma forma de ilustração do que foi feito, veja as figuras 6 e 7.

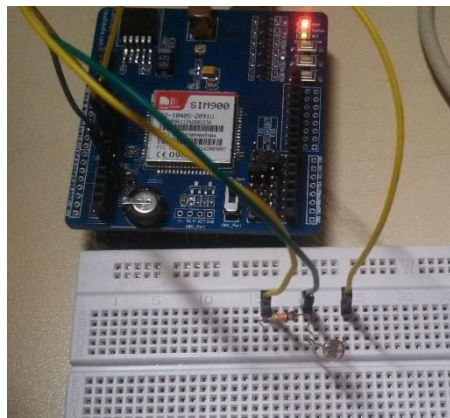


Figura 6: Montagem do LDR

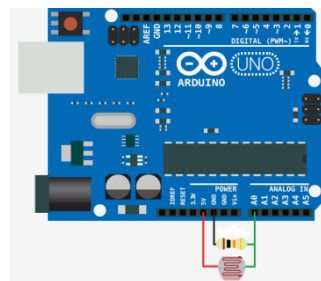


Figura 7: Esquemático de Montagem do LDR

Como foi visto nas figuras 6 e 7, foi utilizado o LDR, que, como o próprio nome sugere, é um resistor cuja resistência se relaciona inversamente com a incidência de luminosidade. Ou seja, uma maior incidência de luz diminui a resistência do sensor.

Após alguns testes do sensor utilizando apenas o *Arduino UNO*, foi inserido a placa *IComSat v1.0*, para que fosse possível fazer um código, a fim de que enviasse o valor de tensão via SMS a um número específico. Foi usado a biblioteca *GSM SHIELD*, disponibilizado no *Google Code*<sup>[5]</sup>, para compilar o código montado. Para ficar mais fácil o entendimento, foi construída uma tabela das principais funções da biblioteca:

Tabela 3: Lista de algumas funções para SMS das bibliotecas do *GSM SHIELD*

Função	Funcionamento
SendSMS (char *number_str, char *message_str)	Envia SMS para um número específico (do tipo <i>char*</i> ) e uma mensagem específica (do tipo <i>char*</i> ).
SendSMS (byte sim_phonebook_position, char *message_str)	Envia SMS para o número gravado na memória da placa e envia uma mensagem específica (do tipo <i>char*</i> ).
GetSMS (byte position, char *phone_number, char *SMS_text, byte max_SMS_len);	Faz a captura de um SMS enviado para a placa e atribui o mesmo em uma posição da memória, grava a mensagem e atribui o valor máximo de bytes da mensagem.
readSMS(char* msg, int msglength, char* number, int nlength);	Lê a mensagem e grava o número e o tamanho do SMS.

Após realizar o primeiro ensaio, foi proposto um novo sensor para implementação do código e o escolhido foi o sensor de corrente, conforme a figura 8, o qual se realizou a última etapa do projeto, onde foi montado um sistema semelhante ao primeiro (figura 9), a fim de que fosse testada a mobilidade do código com outros tipos de sensores. O sensor de corrente elétrica é um dispositivo que capta a intensidade de uma corrente alternada de um sistema em análise e permite que este valor seja, de algum modo, traduzido para um sinal compreensível.

Seu funcionamento é análogo ao dos transformadores de potência, uma vez que, a corrente alternada sensorizada produz um campo elétrico proporcional a sua intensidade. Este campo elétrico induz corrente em uma bobina, no caso em análise, a bobina do sensor. Esta indução proporciona a passagem de corrente na bobina que, em seu terminal final, devido à existência de resistores; fornece uma tensão alternada de valor proporcional ao da corrente sensorizada. Desta forma, o sinal obtido é uma tensão alternada proporcional a corrente medida.



Figura 8: Sensor de Corrente Elétrica *SCT-013*

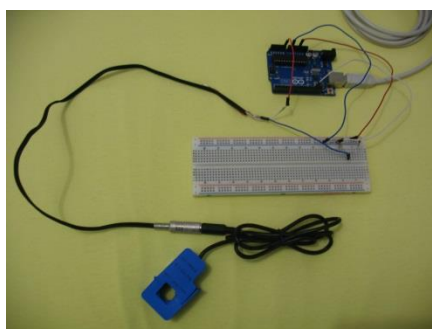


Figura 9: Esquema de Montagem do Sensor de Corrente Elétrica

## REALIZAÇÃO DO PROJETO

No processo de realização do primeiro ensaio, viu-se que com a conversão analógico-digital, o *Arduino* forneceu uma gama de valores entre 0 e 1023, valores de resistência captados pelo sensor LDR, que serão convertidos para valores entre 0V a 5V, sendo quanto maior o número de volts, maior será o valor de tensão. De acordo com o *datasheet* foi elaborada a equação (1) que mostrará o valor de tensão do fio em Volts.

$$\text{Valor de Tensão} = 5 - \left( \frac{5 \cdot (\text{valor do LDR})}{1024} \right) \quad (1)$$

Como visto na figura 10, o gráfico mostra que, de fato, existe uma função inversamente proporcional entre a resistência do sensor e o valor de tensão coletada pelo LDR. O que faz muito sentido quando se observa a equação montada.

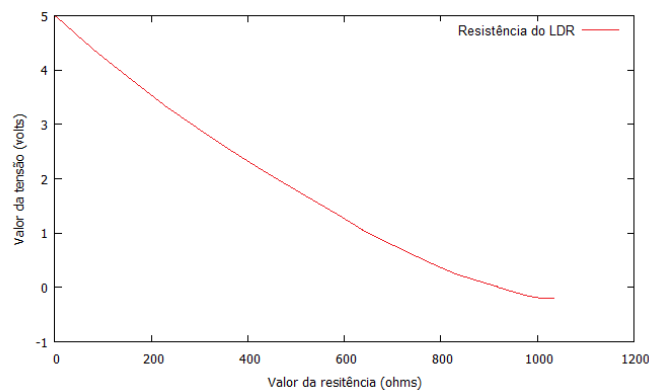


Figura 10: Gráfico do valor de tensão registrada pelo LDR em função do valor da resistência do mesmo

Durante a realização do ensaio, foi encontrado um problema: a conversão do valor de tensão do tipo *float* para *char\**, tipo de variável estabelecido pela biblioteca utilizada, através da função *SendSMS*. Para resolver o problema, foi criado um algoritmo de conversão.

Foi percebido que o tipo *char\** utilizava a tabela ASCII (*American Standard Code for Information Interchange* - Código Padrão Americano para o Intercâmbio de Informação), para representa cada caractere. Portanto, viu-se que era só pegar o número inteiro de cada elemento numérico do tipo *float* e somar 48, visto que o seu resultado era justamente sua representação na tabela ASCII vista na tabela 4.

Tabela 4: Tabela ASCII para Caracteres de 0 a 9 e “.”

Código na tabela ASCII	Caractere
46	.
48	0
49	1
50	2
51	3
52	4
53	5



54	6
55	7
56	8
57	9

De acordo com a tabela 4, foi montado um algoritmo para alocar o código ASCII para cada elemento da *string*, sendo que, de acordo com as características do sensor LDR, a *string* (*char\**) de envio do SMS deve conter cinco posições. Portanto, o algoritmo segue abaixo na tabela 5.

Tabela 5: Algoritmo da resolução do problema da conversão de códigos ASCII em *string* (*char\**)

1. Colete o inteiro do valor de tensão e some ao resultado 48. Atribua o valor encontrado na posição 0 da *string*;
2. Atribua 46 na posição 1 (valor que representa “.” na tabela ASCII);
3. Pegue o valor da tensão e faça a subtração de seu inteiro;
4. Multiplique o valor por 10;
5. Some 48 ao produto e atribua o resultado na posição 2;
6. Pegue novamente o valor da tensão e multiplique por 10;
7. Subtraia o resultado anterior pelo seu inteiro;
8. Multiplique esse valor por 10;
9. Some 48 ao produto e atribua na posição 3;
10. Na posição 4, coloque 0 (código para *nulo*).

Supondo que o valor de tensão seja, por exemplo, 3,58, veja como ficaria a representação do código ASCII na tabela 6, de acordo com o algoritmo anterior.

Tabela 6: Formato do SMS do valor de tensão do sensor usado

	Posições				
<b>Código ASCII</b>	51	46	53	56	0
<b>Visível no SMS</b>	3	.	5	8	

O algoritmo deve se moldar de acordo com os valores de mínimo e máximo e a precisão do sensor. No caso do sensor de luminosidade, o valor mínimo é 0 e o máximo é 5. Usou-se a precisão de duas casas decimais. Em síntese, o código produzido foi escrito na



tabela 7.

Tabela 7: Código utilizado para construir o sistema para o sensor LDR

```
//Bibliotecas
#include "SIM900.h"
#include <SoftwareSerial.h>
#include "sms.h"

//Chamar tipo sms
SMSGSM sms;

boolean started=false; //variável de funcionamento
float ldr = 0; //variável de captura do sensor
float valor_ldr; //valor de tensão do sensor
char x[5]; //string da mensagem SMS

//Iniciar o SIM900
void setup()
{
  Serial.begin(9600);
  Serial.println("GSM Shield Iniciado.");

//Testar o SIM900
  if (gsm.begin(2400))
  {
    Serial.println("/nStatus = PRONTO");
    started=true;
  }
  else Serial.println("/nStatus = OCIOSO");
}

void loop()
{
  valor_ldr = analogRead(ldr); //leitura de captura do sensor
  valor_ldr = 5 -(5*(valor_ldr))/1024; //Equação para encontrar a tensão do LDR

//Algoritmo para alterar float em char* usando tabela ASCII
  x[0] = ((int)valor_ldr) + 48;
  x[1] = 46;
  x[2] = ((valor_ldr) - ((int)valor_ldr))*10 + 48;
  x[3] = (((valor_ldr)*10) - (((int)(valor_ldr*10))))*10 + 48;
  x[4] = 0;

//enviar SMS
  sms.SendSMS("+5561XXXXXXXX", x);

//Imprimir o valor do LDR e o valor da string no "Serial Monitor"
  Serial.println(valor_ldr);
  Serial.println(x);

//Tempo de espera para rodar o programa novamente
  delay(10000);
}
```

```
}
}
```

O segundo ensaio foi realizado de forma semelhante. O objetivo era fazer com que o código pudesse ser implementado em outro tipo de sensor, no caso o sensor de corrente elétrica, com a finalidade de montar um código que pudesse se moldar a diversos tipos de sensores.

Para adequar o sinal fornecido pelo sensor ao sistema do *Arduino*, faz-se necessário o condicionamento do sinal de tensão. Esta etapa de adequação do sinal é realizada através do *software* do *Arduino*, já que, esta forma de condicionamento possui custo benefício mais adequado ao sistema proposto.

Para o condicionamento do sinal, o sistema amostra o sinal de tensão com frequência suficiente para este propósito, respeitando a frequência de *Nyquist*. A partir da amostragem deste sinal submetem-se os dados à formulação da equação 2 de forma a se obter o valor RMS da onda (em Volts) e utilizar este valor de tensão para efetivamente traduzir este em um valor de corrente. De posse do valor de tensão RMS é necessário multiplicar este valor pela constante *k* para se obter o valor da corrente, em Volts, que foi medida, segundo a equação 3.

$$x_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_N^2}{N}} \quad (2)$$

$$I_{RMS} = x_{RMS} \cdot k \quad (3)$$

Sendo que a constante *k* foi obtida através do *datasheet* do sensor e comprovada através de medições, com valor de *k* igual a 15. O próximo passo foi justamente alterar o código para realizar os cálculos das equações encontradas, assim como alterar o algoritmo, visto que os valores máximos de *I* variam entre 0 e 15, com apenas uma casa decimal de precisão. Dessa forma, o código criado foi escrito na tabela 8.

Tabela 8: Código produzido para utilizar o transdutor de corrente

```

/*****BIBLIOTECAS *****/

#include "SIM900.h"

#include <SoftwareSerial.h>

#include "sms.h"

SMSGSM sms;

/***** VARIÁVEIS *****/

```

```
int i=0;                // Define a variável iterativa

int n = 10000;         // Número de amostras

float x;               // Amostra

float xRMS=0;         // Valor máximo da tensão

float cal = 1.6377;    // Calibração

float ref = 3.292;    // Referência

float I = 0, V = 0;    // Corrente, tensão

int numdata;

boolean started=false;

char y[5];

                /***** INICIALIZAÇÃO *****/

void setup()

{

Serial.begin(9600);

Serial.println("GSM Shield Iniciado.");

//Verifica se o programa foi iniciado corretamente

if (gsm.begin(2400))

{

Serial.println("\nStatus = PRONTO");

started=true;

}

}

                /***** PROGRAMA PRINCIPAL *****/

void loop()

{

xRMS = 0;

for (i=0;i<n;i++){

analogReference(EXTERNAL);
```

```
x = analogRead(A0); //Ler o sensor

xRMS = xRMS + sq(x-cal*(1023/ref)); //Calcula uma parte da equação
}

xRMS = floor((sqrt(xRMS/n)*(ref/1023))*1000)/ 1000; //Equação 2

I = xRMS*15.0; //Equação 3

y[0] = ((int)(I/10)) + 48;

y[1] = ((int)(10-I)) + 48;

y[2] = 46;

y[3] = ((I) - ((int)I))*10 + 48;

y[4] = 0;

if(started)
{
sms.SendSMS("+5561XXXXXXXX", y); //Número a ser especificado
}

Serial.println(xRMS,DEC); //Imprime valores

Serial.println(I,DEC);

Serial.println(y);

delay(5000); //Atrasa o retorno ao programa em loop criado

}
```

Com isso, foi possível enviar o nível de tensão em Volts via SMS para um número específico estabelecido no código embarcado na placa e ainda construir um algoritmo que se moldasse ao sensor de forma fácil e rápida. Basta fazer algumas alterações e poder-se-ia utilizar qualquer sensor para algum serviço doméstico.

## CONCLUSÃO

Diante do trabalho realizado, foi percebido que a Automação é uma área de grande

potencial para o futuro, pois ela consegue trazer uma maior praticidade e controle para a vida cotidiana. Além disso, com o avanço da eletrônica, muitos produtos tornaram-se mais barato, fazendo com que tecnologias nessa área se tornem cada vez mais acessíveis à população em geral.

Muitas tarefas podem ser implementadas usando microcontroladores, dentre elas se destacam: o controle do uso de energia e a percepção de luminosidade dos ambientes (visto no projeto). Sendo assim, uma sugestão para continuação do trabalho seria a melhoria do código para fazer um *menu* de escolha de serviços que o *Arduino* ficaria encarregado; um painel de LCD (Liquid Crystal Display - Display De Cristal Líquido) – figura 11- e a adaptação de uma caixa para lacrar a placa, figura 12. Além disso, poder-se-ia elaborar parâmetros que pudessem auxiliar no controle de gastos de energia de acordo com o estilo de casa e de vida dos indivíduos que residissem nesse lar e ainda explorar eletrodomésticos que pudessem realizar tarefas remotamente ou de forma automatizada.



Figura 11: LCD *Black Light*



Figura 12: Caixa para lacra o sistema

## AGRADECIMENTO

Gostaria de agradecer ao meu orientador Ugo Dias pelos conselhos e motivações para a realização do projeto, e também às graduandas Kim Moraes Mota e Wanessa de Alvarenga Silva por auxiliarem na realização do segundo ensaio. Um agradecimento especial ao meu grupo de pesquisa, Fábio Vinícius e Bárbara de Luca, pela dedicação e união. Além disso, gostaria de agradecer a todos os funcionários e professores do laboratório LEMOM pelo apoio técnico e científico. E ainda agradeço à Capes pelo oferecimento do Programa Jovens Talentos para a Ciência que possibilitou a realização desse projeto financeiramente e institucionalmente. Não deixaria de lembrar dos meus pais, Luis e Catia, pelo apoio incondicional na minha trajetória acadêmica e escolar.

## REFERÊNCIAS

[1] Braghetto, L. F.; Silva, S. C.; Brisqui, M. L.; Costa, P. D. (S.D.). *Redes GSM e GPRS. UNICAMP, 2003.*

- [2] Kurose, J. F.; Ross, K. W. *Redes de Computadores e a Internet*. Pearson Prentice Hall, 2010.
- [3] Projeto Wiring. Disponível em <http://www.wiring.org.co/>. Acesso em 01/02/2014
- [4] Pinto, M. D.; Elia, M. D.; Sampaio, F. F. *Formação de Professores em Robótica Educacional com Hardware Livre Arduino no Contexto Um Computador por Aluno*. Anais do WIE, 2012.
- [5] Biblioteca GSM SHIELD. Disponível em <http://code.google.com/p/gsm-shield-arduino/downloads/list>. Acesso em 01/02/2014.
- [6] Cláudio César Silva De Freitas; R. F. (S.D.). *Automação residencial: cenário atual e perspectivas futuras*. UNICAMP, 2012.
- [7] Barreto Francisco Dos Santos, E. C.; J. M. (S.D.). *Automação Residencial: Sistemas Microcontrolados com Comunicação Wireless via GSM*. VII CONNEPI - Congresso Norte Nordeste de Pesquisa e Inovação, Palmas - TO, 2012;
- [8] MAIA, G. M. *Acionamento remoto de portões eletrônicos via celular através de microcontrolador*. Repositório Institucional do UniCEUB, 2012.