



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CÂMPUS CARAÚBAS
CURSO DE CIÊNCIA E TECNOLOGIA

JUDSON PRAXEDES DA SILVA

**PROTÓTIPO PARA MEDIÇÃO DE ENERGIA ELÉTRICA RESIDENCIAL
UTILIZANDO PLATAFORMA ARDUINO**

CARAÚBAS-RN

2018

JUDSON PRAXEDES DA SILVA

**PROTÓTIPO PARA MEDIÇÃO DE ENERGIA ELÉTRICA RESIDENCIAL
UTILIZANDO PLATAFORMA ARDUINO**

Trabalho de Conclusão de Curso
apresentado à Universidade Federal
Rural do Semi-árido - UFERSA,
Câmpus Caraúbas, para a obtenção do
Título de Bacharel em Ciência e
Tecnologia.

Orientador: Ms. Juan Rafael Filgueira
Guerra- UFERSA.

CARAÚBAS-RN

2018

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

S586p Silva, Judson Praxedes.
Protótipo para medição de energia elétrica
residencial utilizando plataforma Arduino /
Judson Praxedes Silva. - 2018.
48 f. : il.

Orientador: Juan Rafael Filgueira Guerra.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Ciência e
Tecnologia, 2018.

1. Protótipo. 2. Consumo. 3. Arduino. 4. Medição
de energia. 5. Corrente contínua. I. Guerra, Juan
Rafael Filgueira, orient. II. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

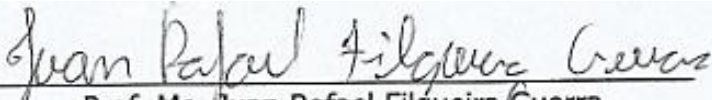
JUDSON PRAXEDES DA SILVA

**PROTÓTIPO PARA MEDIÇÃO DE ENERGIA ELÉTRICA RESIDENCIAL UTILIZANDO
PLATAFORMA ARDUINO**

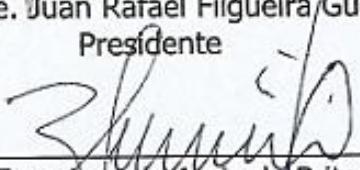
Trabalho de Conclusão de Curso
apresentado à Universidade Federal
Rural do Semi-árido - UFERSA,
Câmpus Caraúbas, para a obtenção do
Título de Bacharel em Ciência e
Tecnologia.

APROVADO EM: 19/04/2018.

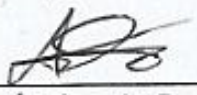
BANCA EXAMINADORA



Prof. Me. Juan Rafael Filgueira Guerra
Presidente



Prof. Dr. Francisco de Assis de Brito Filho
Primeiro Membro



Prof. Me. José Ailton L. Barboza Júnior
Segundo Membro

DEDICATÓRIA

À minha mãe, Rosangela Praxedes, por toda ajuda e apoio dados ao longo de toda minha vida, me possibilitando concluir este curso.

À minha irmã, Nisia Sthefane Praxedes, pelo apoio familiar durante todo este tempo.

Á minha avó, dona Maria de Lourdes Bezerra, pelos cuidados dados a minha pessoa durante todos estes anos.

Ao meu orientador, Juan Rafael Filgueira Guerra, por ter me ajudado quando precisei de orientação, com todo apoio, paciência e conhecimento compartilhado durante a realização deste trabalho.

“Tenho a impressão de ter sido uma criança brincando à beira-mar, divertindo-me em descobrir uma pedrinha mais lisa ou uma concha mais bonita que as outras, enquanto o imenso oceano da verdade continua misterioso diante de meus olhos”

Isaac Newton.

RESUMO

Tendo em vista que o consumo de energia elétrica residencial aumenta a cada ano, tem-se a necessidade de incentivo ao consumidor na tomada de medidas proativas em relação a minimização do desperdício proveniente deste aumento. Com base nisso, o presente trabalho tem como objetivo desenvolver um protótipo capaz de realizar a medição de energia elétrica para corrente contínua. Este utilizará como base a plataforma arduino para seu desenvolvimento, bem como a criação de um algoritmo que permita a interação entre *hardware* e *software*. Por fim, serão realizadas medições de grandezas elétricas (corrente, tensão, potência) visando estimar o consumo individual de um dispositivo em tempo real, comparando-se os resultados obtidos entre diferentes tipos de medidores de energia.

Palavras chave: Consumo; Protótipo; Arduino; Medição de energia; Tempo real.

ABSTRACT

Since residential electricity consumption increases every year, there is a need to encourage the consumer to take proactive measures to minimize the waste resulting from this increase. Based on this, the present work aims to develop a prototype capable of performing the measurement of electrical energy in a single-phase circuit. This will use as base the arduino platform for its development, as well as the creation of an algorithm that allows the interaction between hardware and software. Finally, measurements of electrical quantities (current, voltage, power) will be carried out in order to estimate the individual consumption of a device in real time, comparing the results obtained between different types of energy meters.

Keywords: Consumption; Prototype; Arduino; Measurement of energy; Real time.

LISTA DE FIGURAS

Figura 1 - Placa <i>Serial</i>	12
Figura 2 - Arduino USB.....	13
Figura 3 - Arduino <i>Extreme</i>	13
Figura 4 - Placa utilizada no protótipo	14
Figura 5 - <i>Pinout</i> opto acoplador PC817.....	16
Figura 6 - Disposição dos caracteres.....	17
Figura 7 - Tensão aplicada em um semicondutor.....	18
Figura 8 - Campo magnético externo atuando sobre o material.....	19
Figura 9 - Transdutor de corrente do circuito ACS712.....	20
Figura 10 - <i>Pinout</i> do circuito integrado ACS712.....	20
Figura 11 - Modelo de <i>Home Energy Monitor</i>	21
Figura 12 - Medidor de energia por indução eletromecânica.....	22
Figura 13 - Triângulo das potências.....	24
Figura 14 - Interface de desenvolvimento do Arduino.....	26
Figura 15 - Diagrama do protótipo.....	31
Figura 16 - Esquemático do circuito sensor de tensão	32
Figura 17 - Esquemático do circuito sensor de corrente	33
Figura 18 - Descrição do funcionamento de um conversor A/D.....	36
Figura 19 - Diagrama da montagem do protótipo	37
Figura 20 - Resultado final da montagem	38
Figura 21 - Resultados apresentados no display	39
Figura 22 - Resultado da montagem final dos sensores	41
Figura 23 - Gráfico de tensões do transistor para o PC817.....	41

LISTA DE TABELAS

Tabela 1 - Características elétricas PC817	16
Tabela 2 - Pinout e funções do display	18
Tabela 3 - Características elétricas do sensor ACS712.....	20
Tabela 4 - Resultados para corrente	39
Tabela 5 - Resultados para medidas de tensão	42
Tabela 6 - Estimativas de custos do protótipo.....	43

SUMÁRIO

1.	INTRODUÇÃO GERAL.....	10
1.1	OBJETIVOS	11
1.1.1	GERAL	11
1.1.2	ESPECÍFICOS	11
2.	REFERENCIAL TEÓRICO.....	12
2.1	PLATAFORMA ARDUINO	12
2.2	CONSTRUÇÃO DA PLATAFORMA.....	12
2.2.1	PINOUT E PRINCIPAIS FUNÇÕES.....	14
2.3	SHIELDS APLICADOS AO ARDUINO	15
2.4	OPTO ACOPLADOR.....	15
2.5	DISPLAY LCD.....	16
2.6	EFEITO HALL	18
2.7	SENSOR DE EFEITO HALL.....	19
2.8	MEDIDORES DE ENERGIA	21
2.8.1	INDUÇÃO ELETROMECAÂNICA	21
2.9	POTÊNCIA	22
2.9.1	POTENCIA ATIVA	23
2.9.2	POTÊNCIA REATIVA	23
2.9.3	POTENCIA APARENTE	23
3	ASPECTOS DE PROGRAMAÇÃO VOLTADOS A PLATAFORMA ARDUINO	25
3.1	LINGUAGEM C APLICADA A PLATAFORMA ARDUINO	25
3.2	ARDUINO IDE	26
3.2.1	ESTRUTURA DO CÓDIGO	27
3.2.2	BIBLIOTECAS	27

3.2.3	TIPOS DE VARIÁVEIS E PRINCIPAIS FUNÇÕES	29
4	MATERIAL E MÉTODOS	31
4.1	PROTÓTIPO PROPOSTO	31
4.2	CIRCUITO SENSOR DE TENSÃO	32
4.3	CIRCUITO SENSOR DE CORRENTE	32
4.4	ELABORAÇÃO DO CÓDIGO	33
5.	RESULTADOS E DISCUSSÕES	37
5.1	ESTIMATIVAS DE CUSTOS DO PROJETO	43
6.	CONCLUSÃO	44
	REFERÊNCIAS BIBLIOGRÁFICAS	45

1. INTRODUÇÃO GERAL

A eficiência energética destaca-se cada vez mais no cenário mundial por contribuir com a redução do desperdício de energia elétrica residencial, que por sua vez reduz as consequências de impactos ambientais advindos da necessidade da implantação de grandes projetos de geração de energia. (ANEEL, 2015)

Segundo Copel (2015), com o passar dos anos e o aumento populacional crescente nas regiões urbanas, muitos países, principalmente os mais industrializados, quase que duplicam seu consumo de energia elétrica a cada década. Com base nessa afirmação, observa-se que no Brasil, segundo a Empresa de Pesquisa Energética (EPE), no ano de 2006, o consumo total de energia residencial foi de 84,768 GWh, enquanto que no mesmo período do ano de 2016, registrou-se um consumo de 132,893 GWh, um aumento de 56,77%. Segundo o Ministério do Meio Ambiente (MMA), o desperdício de energia chega a 40 milhões de kW, equivalente a US\$ 2,8 bilhões por ano no Brasil. Os consumidores industriais, residenciais e comerciais desperdiçaram 22 milhões de kW em 2014 no Brasil.

Dentro deste contexto, pode-se destacar a importância do acompanhamento do consumo energético individual que, com o avanço tecnológico, torna-se cada vez mais acessível principalmente ao consumidor residencial. Dentre as tecnologias de medição de energia, pode-se destacar os *Home Energy Monitors*, que permitem comunicação entre diferentes interfaces como computadores e *smartphones*, com medição em tempo real, e transmissão de dados via *wireless*. Partindo disto, pode-se destacar o Arduino como uma ótima interface multiplataforma, usada para auxiliar o programador nos mais diversos tipos de projetos eletrônicos, dentre eles os medidores de energia.

Neste sentido, o presente trabalho visa o desenvolvimento de um protótipo de medição de energia elétrica, que possa auxiliar o acompanhamento do gasto energético, em tempo real, por parte do consumidor residencial, para que o mesmo tenha melhor comportamento quanto ao consumo nos seus utensílios, propiciando assim que sejam realizadas ações assertivas acerca do aumento da eficiência energética.

1.1 OBJETIVOS

1.1.1 Geral

Desenvolver um protótipo capaz de realizar a medição de energia elétrica para corrente contínua, voltado para uso residencial, utilizando a plataforma Arduino.

1.1.2 Específicos

Para atingir o objetivo geral, os seguintes objetivos específicos foram traçados:

- Estudar os parâmetros relacionados ao consumo de energia.
- Estimar fatores relacionados a implementação de medidores de energia.
- Desenvolver algoritmo que permita ao protótipo realizar a medição de corrente e tensão da rede elétrica residencial;
- Realização de testes utilizando *Softwares* e *Protoboard* em laboratório.

2. REFERENCIAL TEÓRICO

2.1 PLATAFORMA ARDUINO

O Arduino é um projeto totalmente aberto de protótipos de eletrônica baseados numa plataforma de hardware e software flexível e de fácil utilização. Tem como objetivo fornecer uma plataforma fácil para prototipação de projetos interativos, utilizando um microcontrolador. Uma placa arduino é composta por microprocessador, memória *Random Access Memory* (RAM), memória *flash* (para guardar o software), temporizadores, contadores, dentre outras funcionalidades. (ARDUINO, 2015)

Seu microcontrolador é fabricado pela *ATMEL Corporation* e é programável utilizando as bibliotecas do Arduino, e também aceita código diretamente programados em C/C++, bem como o ambiente do Arduino que é baseado em *Processing*.

2.2 CONSTRUÇÃO DA PLATAFORMA

As primeiras placas fabricadas para a plataforma Arduino utilizavam comunicação Serial e componentes discretos. Essas placas necessitavam de alimentação externa através de plug *Jack* para seu funcionamento.

Figura 1 - Placa *Serial*



FONTE: Espinola, 2016.

Na fabricação dos hardwares seguintes, passou-se a utilizar a entrada USB no lugar do plug *Jack* na alimentação do circuito. Com isso, surgiu a segunda placa construída para a plataforma, chamada Arduino USB.

Figura 2 - Arduino USB



FONTE: Espinola, 2016.

A terceira lançada foi chamada de *Arduino Extreme*, e dentre as funcionalidades acrescentadas em relação a anterior, pode-se destacar a adição conectores *headers* fêmea, bem como leds, entradas RX e TX, responsáveis pelo tráfego de dados entre a placa e o computador. (SOUZA, 2013)

Figura 3 - Arduino *Extreme*



FONTE: Souza, 2013.

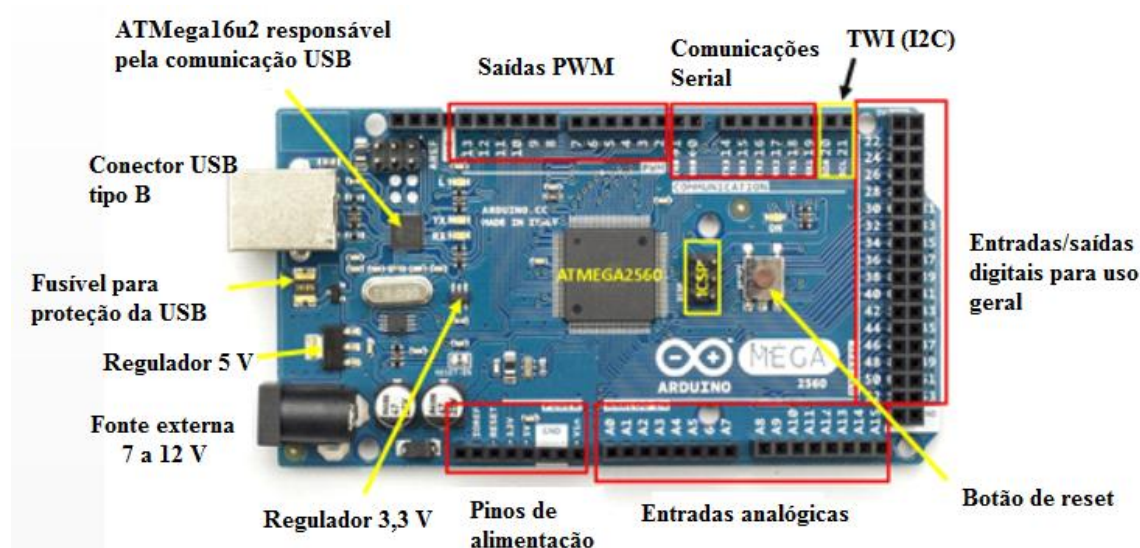
Por conseguinte, foi desenvolvida a Diecimila que é baseada no microcontrolador ATmega168. A mesma possui 14 pinos de saída/entrada digitais (nos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um oscilador de cristal de 16MHz, uma entrada USB, uma entrada de alimentação, e um botão de *reset*.

Por fim, foi lançada a Duemilanove, que apresentou evoluções no hardware e software, pois esta dobrou sua capacidade de memória flash de 16KB para 32KB bem como sua EEPROM pode agora ser desenvolvida para 1KB.

2.2.1 PINOUT E PRINCIPAIS FUNÇÕES

O Arduino Mega 2560 tem seu funcionamento padronizado segundo o microcontrolador ATmega2560. Possui 54 pinos de entrada/saída (no qual 14 podem ser usados como saídas PWM), 16 entradas analógicas, 4 *Universal Asynchronous Receiver Transmitter* (UARTs), um oscilador de cristal de 16 MHz, uma entrada USB, uma entrada Power, uma In Circuit Serial Programming (ICSP) e um botão de Reset. (ARDUINO, 2015)

Figura 4 - Placa utilizada no protótipo



FONTE: Souza, 2014

O hardware pode operar com fornecimento externo de 6 a 20 volts. Em caso de fornecimento abaixo de 5V, pode causar instabilidade. Se estiver usando mais de 12V, o regulador de tensão pode superaquecer e danificar o circuito. O intervalo recomendado é de 7 a 12 volts. (ARDUINO, 2015)

O ATmega2560 possui 256 KB de memória *Flash* para armazenamento (no qual 8 KB são usados para o *bootloader*), 8 KB de *Static Random Access Memory* (SRAM) e 4 KB de *Electrically-Erasable Programmable Read-Only Memory* (EEPROM). Cada um dos 54 pinos digitais podem ser usados como entrada ou saída, através das funções *pinMode()*, *digitalWrite()* e *digitalRead()*. Cada pino pode prover ou receber um máximo de 40 mA e possuem um resistor *pull-up* interno (desconectado por padrão) de 20-50 kOhms. Possui ainda 16 pinos analógicos, no qual cada um fornece 10 bits de resolução (1024 valores diferentes). (ARDUINO, 2015)

2.3 SHIELDS APLICADOS AO ARDUINO

Shields são placas usadas com objetivo de possibilitar diversas funcionalidades ao Arduino. Pode ser utilizado mais de um *shield* ao mesmo tempo, proporcionando a criação de módulos de hardware com várias funções diferentes. Os *Shields* possuem preço acessível, bem como são de fácil utilização. Dentre os *Shields* mais utilizados, pode-se destacar:

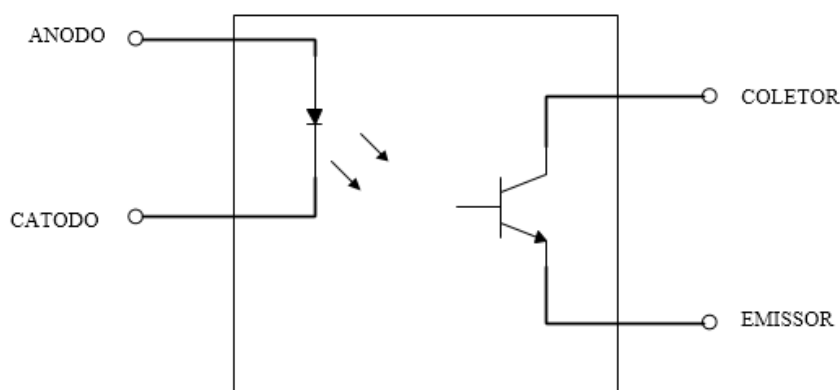
- Medidores de corrente: compostos por toróides e sensores de efeito Hall, permitem a medição de corrente primária através de seu campo magnético;
- *Motor Shield*: Permite o comando de motores de corrente direta e interpretação de *encoders*;
- *Protonshield*: Permite a multiplicação de conexões das portas do Arduino através de uma malha semelhante a uma *protoboard* tradicional;
- *Ethernet*: Permite a comunicação da placa Arduino com a Internet, verificação de posicionamento geográfico do dispositivo, envio e recebimento de comandos por sinais de radio, SMS (*Short Message Service*), pacote de dados 3G, entre outras funções.
- *Display Shield*: Permite a visualização de informações computadas pelos sensores integrados ao Arduino e em alguns modelos existem também botões com funções programáveis para interagir com o sistema de forma simples.

2.4 OPTO ACOPLADOR

Um opto acoplador é um dispositivo que utiliza um diodo foto emissor, bem como um foto transistor responsável por filtrar a tensão que é aplicada no anodo, através da luminosidade emitida pelo diodo.

Desse modo, ambos são combinados em um único dispositivo semicondutor, um foto transmissor e um foto receptor, cuja conexão entre os ambos é óptica.

Figura 5 – Pinout opto acoplador PC817



FONTE: Autor (Adaptado de *Sharp microelectronics*), 2018.

Estes elementos são encapsulados por *Dual In-line Package* (DIP), com o objetivo de isolá-los eletricamente devido a sua sensibilidade. (TESLABEM, 2017)

Tabela 1 - Características elétricas PC817

Parâmetro	Símbolo	Condição	Min.	Med.	Max.	Unidade
Tensão direta	V_F	$I_F = 20\text{mA}$	-	1.2	1.4	V
Tensão direta de pico	V_p	$I_{FM} = 0.5\text{A}$	-	-	3.0	V
Corrente inversa	I_R	$V_R = 4\text{V}$	-	-	10	A
Corrente no coletor-emissor	I_{ce}	$V_{CE} = 50\text{V}$	-	-	100	nA
Tensão de coletor-emissor	V_{ce}	$I_C = 0.1\text{mA}$	5	-	-	V
Tensão de ruptura coletor-emissor	V_{ec}	$I_E = 10\text{mA}$	6	-	-	V
Corrente no coletor	I_C	$I_F = 5\text{mA},$ $V_{CE} = 5\text{V}$	2.5	-	30.0	mA
Tensão de saturação de coletor-emissor	$V_{ce}(\text{sat})$	$I_F = 20\text{mA},$ $I_C = 1\text{mA}$	-	0.1	0.2	V

FONTE: *Sharp microelectronics*, 2018.

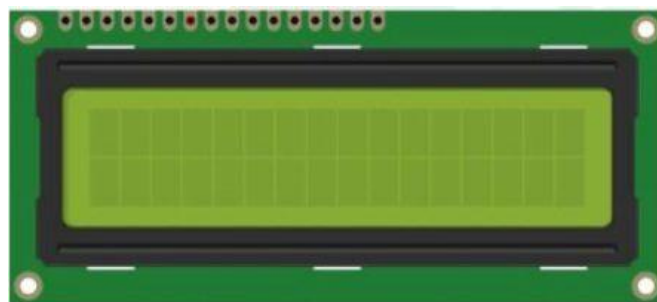
2.5 DISPLAY LCD

Segundo Largura (2017), os *Liquid Crystal Display* (LCDs) são formados por uma fina camada de cristal líquido entre duas placas de vidro, utiliza-se LEDs ou luz fluorescente na parte inferior da estrutura. O cristal líquido, ao receber uma carga elétrica, torna-se ligeiramente opaco, impedindo a passagem da luz através da camada, resultando no surgimento de caracteres ou imagens no *display*. Existem, basicamente, dois tipos de LCDs:

- Caractere: Esses tipos de LCD permitem apenas a escrita de caracteres, números e pequenos símbolos criados pelo usuário. Alguns tamanhos comuns para esse tipo de display são: (8×2, 16×2, 20×4, 24×4, 40×2) caracteres, entre outros.
- Gráficos: Os LCDs gráficos possuem resoluções maiores e permitem a exibição de figuras e imagens. Alguns tamanhos comuns para esse tipo de *display* são: (122×32, 128×64, 240×64, 240×128) caracteres, entre outros.

Nesse protótipo usou-se um LCD 2x16, ou seja, possui 16 colunas e 2 linhas, ou seja, o *display* pode apresentar 32 caracteres em sua tela. Os caracteres são compostos individualmente por uma matriz de 8 linhas por 5 colunas de pixels. Dependendo do estado deste conjunto de pixels, será formado o devido caractere. (HITACHI ELECTRONICS, 2017)

Figura 6 - Disposição dos caracteres



FONTE: Portal VidaDeSilicio, 2017

O display de LCD LM016L exibe alfanuméricos, caracteres e símbolos. Pode ainda ser configurado para conduzir uma exibição de cristal líquido de matriz de pontos sob o controle de microprocessador de 4 ou 8 bits.

A tabela a seguir detalha quais são os pinos do display LCD LM016L e suas respectivas funções.

Tabela 2 - Pinout e funções do display

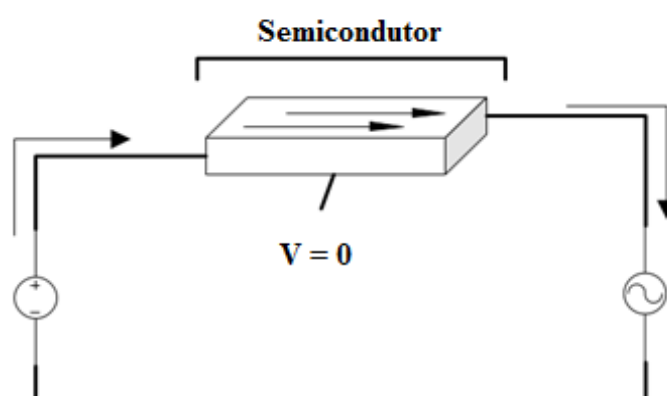
Pino	Função	Descrição
1	Alimentação	GND
2	Alimentação	VCC
3	V0	Pino para ajuste do contraste
4	RS	1 = Dado, 0 = Instrução
5	R/W	1 = Leitura, 0 = Escrita
6	E (Chip select)	1 = Habilita, 0 = Desabilita
7	B0	Pino de dados(bit mais significativo)
8-14	B1-B7	Pino de dados
15	A	Ânodo do LED de backlight
16	K	Cátodo do LED de backlight

Fonte: Pictronics, 2014.

2.6 EFEITO HALL

A sensibilidade elétrica dos materiais semicondutores a variação de temperatura é maior que a de outros materiais. Desta forma, a redução da temperatura os leva a se tornarem condutores. E isto influencia no transporte da corrente por portadores de carga, sejam eles positivos ou negativos, amplamente utilizada na construção de dispositivos que fazem uso de junções do tipo PN. (BRAGA, 2015)

Figura 7 - Tensão aplicada em um semicondutor

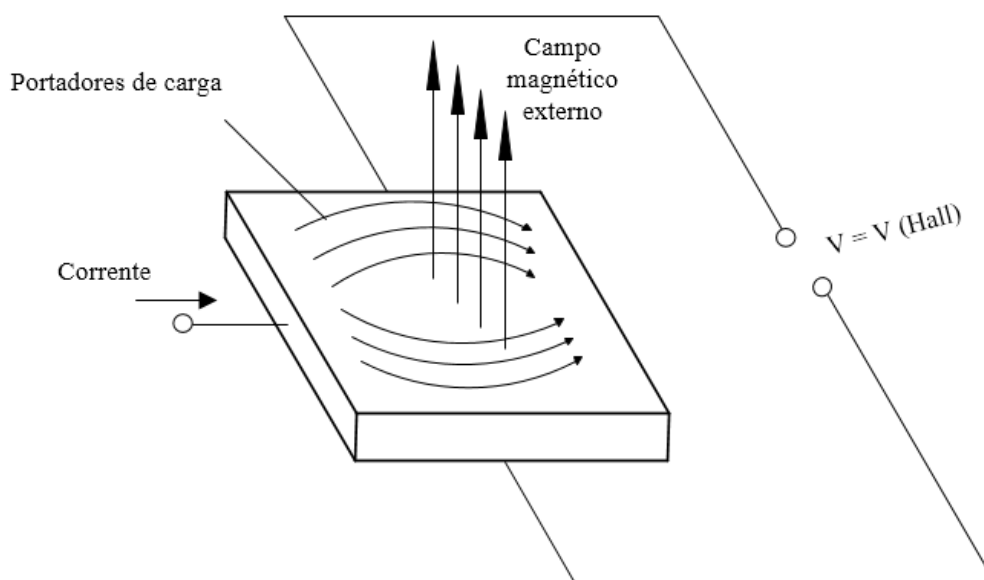


FONTE: Autoria própria, 2018.

Dessa forma, ao aplicar uma tensão no sentido longitudinal a um material semicondutor sem a interferência de campo magnético externos, nota-se que a corrente atravessa o material com os portadores de carga se distribuindo de maneira uniforme. Entretanto, ao aplicar-se um campo magnético externo, os portadores de carga serão

deslocados por uma força perpendicular ao seu deslocamento, modificando sua distribuição, como se pode observar na figura 8. (BRAGA, 2015)

Figura 8 - Campo magnético externo atuando sobre o material



FONTE: Autoria própria, 2018.

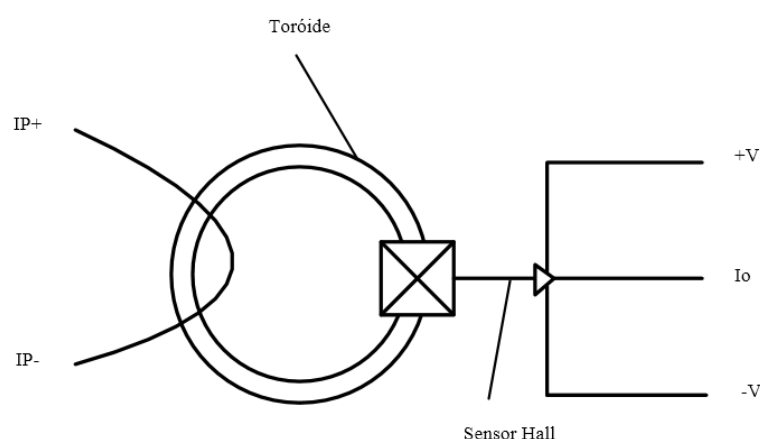
Estes portadores de cargas que sofreram alterações no seu deslocamento devido a presença do campo magnético se acumulam na superfície mais próxima a fonte do campo, na qual detecta-se a presença de tensão. Com isso, tem-se que o aumento da resistência resulta na redução da corrente do circuito. (BRAGA, 2015)

2.7 SENSOR DE EFEITO HALL

Os sensores de efeito Hall utilizam como base os transdutores, que possuem capacidade de converter informação não elétrica (temperatura, pressão) em informação elétrica, como resistência e tensão. Estes transdutores são elaborados através de cristais do tipo piezoelétricos, que quando submetidos a uma pressão, geram um campo elétrico.

O sensor de efeito Hall do tipo invasivo ACS712, fabricado pela empresa *Allegro Instruments*, utiliza como princípio de funcionamento o efeito *Hall*. Desta forma, o dispositivo possui um toróide no qual é aplicada a corrente que flui entre os pinos IP, gerando assim um campo magnético.

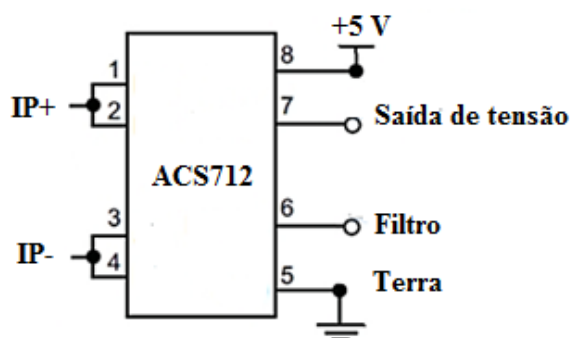
Figura 9 - Transdutor de corrente do circuito ACS712



FONTE: Autoria própria, 2018.

Este campo é detectado pelo transdutor de efeito *Hall*, que o converte em uma tensão proporcional de baixa compensação. Sua precisão é determinada através da proximidade do sinal com o transdutor. (ALLEGRO INSTRUMENTS, 2015)

Figura 10 – Pinout do circuito integrado ACS712.



FONTE: Autor (adaptado de *Allegro Instruments*), 2018.

Tabela 3 - Características elétricas do sensor ACS712

Características	Símbolos	Condições	Mínimo	Médio	Máximo	Unidade
Tensão de alimentação	V_{CC}	-	4.5	5.0	5.5	V
Corrente de alimentação	I_{CC}	$V_{CC} = 5.0\text{ V}$	-	10	13	mA
Capacitancia da carga	C_{CARGA}	V_{IOUT} para GND	-	-	10	nF
Resistencia da carga	R_{CARGA}	V_{IOUT} para GND	4.7	-	-	k Ω
Resistencia primária do conductor	$R_{PRIMARIO}$	$T_A = 25^\circ\text{C}$	-	1.2	-	m Ω

Tensão de saída ($I_p = 0$)	$V_{IOUT(Q)}$	Bidirecional; $I_p = 0$ A, $T_A = 25^\circ\text{C}$	–	$0,5V_{CC}$	–	V
-------------------------------	---------------	--	---	-------------	---	---

Fonte: *Allegro Instruments*, 2018.

2.8 MEDIDORES DE ENERGIA

Dentre os medidores de corrente e tensão comercializáveis voltados a auxiliar o consumidor doméstico a obter dados referentes ao seu consumo em tempo real, pode-se destacar os *Home Energy Monitors*.

- **Home Energy Monitors**

Figura 11 - Modelo de *Home Energy Monitor*



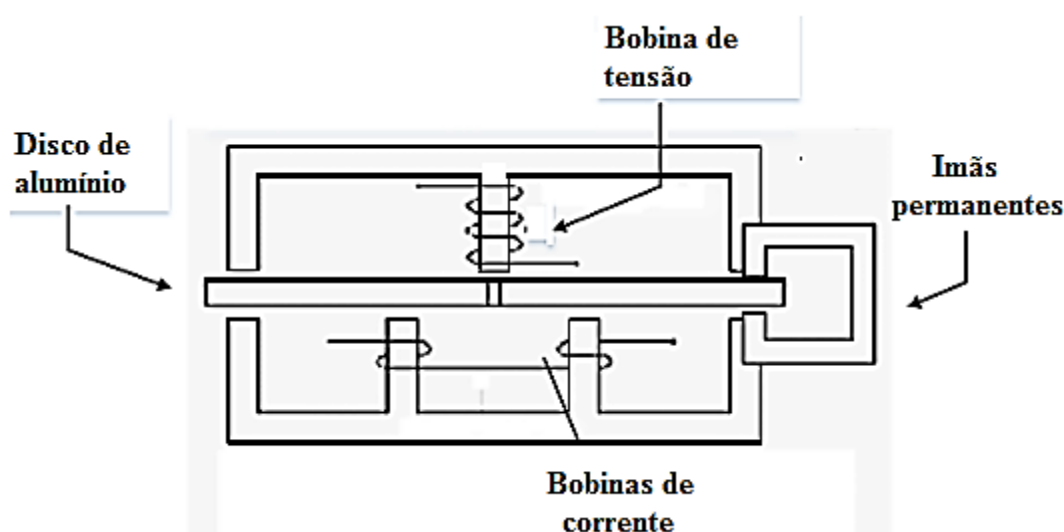
FONTE: Portal *Mitchellinstruments*, 2017.

Estes dispositivos, assim como o desenvolvido através deste projeto, utilizam sensores de medição invasivos, que podem ser conectados na rede monofásica de energia de uma residência.

Suas funcionalidades variam de acordo com o preço de cada dispositivo. Dentre elas, pode-se destacar a comunicação entre diferentes interfaces como computadores e *smartphones*, bem como o armazenamento em banco de dados online, e cálculo de valores como potência ativa e aparente.

2.8.1 INDUÇÃO ELETROMECÂNICA

Figura 12 - Medidor de energia por indução eletromecânica



FONTE: Autor (Adaptado de *Datasheet*), 2018.

Os medidores baseados no princípio de indução eletromecânica utilizam um disco de alumínio movimentado por uma força proveniente de um torque gerado pelo acoplamento magnético entre duas bobinas, no qual são induzidas correntes de Foucault, ou correntes parasitas. O valor aproximado de medição gira em torno de 1 kWh a cada 330 rotações. A velocidade das rotações é proporcional a potência que flui no medidor. (BRAGA, 2015)

2.9 POTÊNCIA

A importância da potência em projetos de medidores de energia se dá através da determinação da taxa de energia consumida em função do tempo. Segundo Nielson (2003), a mesma pode ser definida como uma grandeza física definida pela relação diretamente proporcional entre a corrente e a tensão. No regime estacionário, considera-se que os sinais de corrente e tensão são senoidais, de modo que pode-se fazer uso destes para o cálculo da potência instantânea, como mostrado na Equação 1.

$$P = VI \quad (1)$$

Sabendo-se que P é a potência instantânea dada em Watts, V é a tensão elétrica [V] e I é a corrente elétrica [A].

2.9.1 POTENCIA ATIVA

Segundo Riedel (2009), pode-se definir a potência ativa como a que converte energia elétrica em não elétrica, calculada através do valor médio da potência instantânea em um determinado período de tempo.

$$P = VICos(\theta_v - \theta_i) \quad (2)$$

De modo que P é a potência [W], V é a tensão dada em Volts, I é a corrente [A], θ_v é o ângulo de fase da tensão e θ_i o ângulo de fase da corrente.

2.9.2 POTÊNCIA REATIVA

Segundo Nillson (2003), diferentemente da potência ativa, a potência reativa não pode ser convertida em formas não elétricas de energia, podendo ser armazenada no campo magnético de um indutor ou capacitor. Sua potência é medida em Var (Volt Ampere Reativo), como na Equação 3.

$$Q = VISen(\theta_v - \theta_i) \quad (3)$$

De modo que Q é a potência reativa [Var], V é a tensão dada em Volts, I é a corrente [A], θ_v é o ângulo de fase da tensão e θ_i o ângulo de fase da corrente.

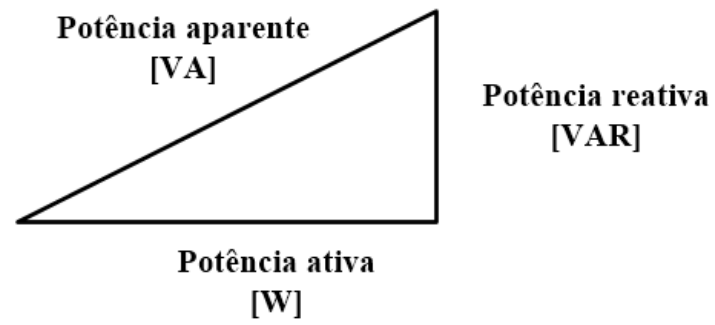
2.9.3 POTENCIA APARENTE

Num triângulo retângulo, também chamado de triângulo das potências, a potência aparente é a soma quadrática das potências ativa e reativa, e seu resultado é dado em Volt-Ampère (VA) (NIELSON, 2003).

$$|S| = \sqrt{P^2 + Q^2} \quad (4)$$

Sabendo-se que S é a potência aparente dada em [VA], P é a potência ativa [W] e Q é a potência reativa [Var].

Figura 13 - Triângulo das potências



FONTE: Autoria própria, 2018.

3 ASPECTOS DE PROGRAMAÇÃO VOLTADOS A PLATAFORMA ARDUINO

Com a evolução tecnológica, cada vez mais foram surgindo diferentes tipos de linguagens de programação voltadas para os mais variados propósitos. Dentre as linguagens usadas em projetos com micro controladores, ou sistemas embarcados, pode-se destacar algumas como a *Assembly*, *C*, *C++*, *Java*, *Python*, *Java Script* ou *Wiring*.

Algumas dessas linguagens são mais voltadas para a facilidade de interação com o hardware, como a *Assembly*. Esta, por sua vez, possui como notação símbolos conhecidos por *mneumônicos*, responsáveis por substituir os comandos interpretados pelas máquinas (binários e hexadecimais), por caracteres facilmente interpretados pelo programador. (ASSEMBLY PROGRESSIVO, 2016)

Já a linguagem *C*, que é a linguagem mais utilizada para o desenvolvimento dos mais variados tipos de algoritmos no mundo, herdou muitas características da linguagem citada no parágrafo anterior. De modo a ser mais flexível e possuir um número maior de bibliotecas e funções nativas, bem como uma padronização que é o seu diferencial.

Como uma linguagem orientada a objeto tem-se a *Java*. Muitos micro controladores a usam devido ao ganho de performance na execução de aplicativos que utilizam esta linguagem. Possui inclusive tecnologia embutida em alguns microprocessadores do tipo ARM. (WALLS, 2014)

A *Python*, criada pelo programador Guido van Rossum, é uma linguagem multi plataforma que possui uma larga gama de bibliotecas e funções disponíveis para atender desde algoritmos simples ao mais complexos. Possui ainda alta legibilidade, facilitando assim a compreensão de algoritmos datados. (PYSCIENCE, 2014)

Por fim, tem-se a linguagem mais utilizada para desenvolvimento de algoritmos na plataforma Arduino, conhecida por *Wiring*. Desenvolvida em *Processing*, tem como objetivo a facilitação de acesso a pessoas com conhecimento limitado em programação, que desejem realizar projetos que não exijam muito processamento de dados por parte do micro controlador.

3.1 LINGUAGEM C APLICADA A PLATAFORMA ARDUINO

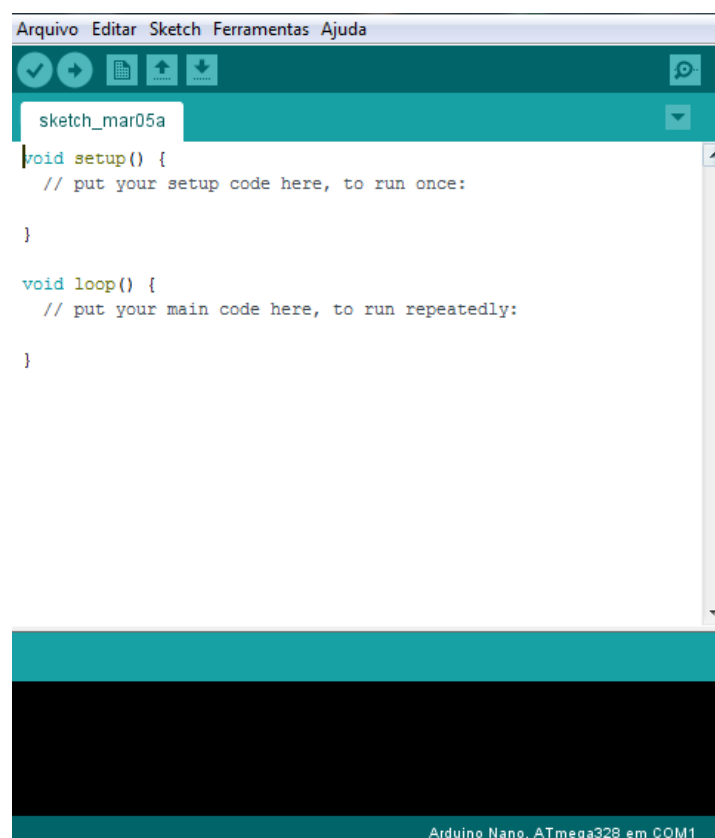
O desenvolvimento dos algoritmos voltados para placas arduino é feito em

Wiring, que consiste em uma plataforma de programação composta por interface de desenvolvimento conhecida como IDE, e uma linguagem de programação baseada em C++, porém com menos padronização e um acervo menor de bibliotecas e funções.(BARRAGÁN, 2017)

Dentre as diferenças encontradas entre a linguagem C que é comumente utilizada e a linguagem *Wiring*, pode-se destacar a restrição no acervo de funções e bibliotecas disponíveis para esta plataforma devido a limitações de hardware, bem como a preferência da demanda deste tipo de dispositivo, que é mais voltado a projetos não tão complexos.

3.2 ARDUINO IDE

Figura 14 - Interface de desenvolvimento do Arduino



FONTE: Arduino, 2017.

Sua interface de desenvolvimento possui características semelhantes a outros compiladores, porém com menos ferramentas mostradas na tela inicial do *software*. O mesmo possui recursos para exportar o binário compilado, bem como procurar e

substituir um texto escrito dentro do código. Possui também no canto inferior um console, responsável por reportar mensagens de erro e outras informações relacionadas ao código.

Algoritmos feitos a partir do IDE são chamados *sketches*, que possuem o mesmo significado de programas em outras linguagens. Estes, por sua vez, possuem um conjunto de funções, instruções e variáveis que dão forma ao algoritmo. Os microcontroladores ATMEL possuem um compilador chamado avr-gcc, que é responsável por transformar o *sketch* (ou programa) em um arquivo objeto. Ao combinar-se este *sketch* com as bibliotecas padrão do Arduino, tem-se um arquivo com códigos compreensíveis ao microcontrolador.(CAETANO, 2016)

3.2.1 ESTRUTURA DO CÓDIGO

A referência para programas Arduino pode ser dividida em três seções principais: Estrutura, Valores (variáveis e constantes), e Funções. Na parte de Estrutura, utiliza-se expressões como *void setup()* e *void loop()* para iniciar o código. Existem ainda outras categorias, como:

- Estruturas de controle, como as funções *if*, *if...else*, *for*, *switch case*, *while*, *do...while*, *break*, *continue*, *return*, e *goto*.

Um operador é um conjunto de um ou mais caracteres que serve para relacionar uma ou mais variáveis ou constantes. Dentre os operadores mais utilizados na plataforma IDE, pode-se destacar:

- Operadores aritméticos: = (atribuição), + (adição), - (subtração), * (multiplicação), / (divisão), % (módulo).
- Operadores Comparativos: == (igual a), != (não igual a), < (menor que), > (maior que), <= (menor ou igual a), >= (maior ou igual a).
- Operadores Booleanos: && (e), || (ou), ! (negação)
- Operadores de Atribuição Composta: ++ (incremento), -- (decremento), += (adição composta), -= (subtração composta), *= (multiplicação composta), /= (divisão composta).

3.2.2 BIBLIOTECAS

As bibliotecas (*libraries*) padrão Arduino formam um conjunto de funções responsáveis por facilitar a comunicação entre o programador e a interface de desenvolvimento. Dentre as principais funções disponibilizadas pela biblioteca *LiquidCrystal*, estão:

- `Begin()`

Inicializa a interface com o LCD, recebendo como parâmetros o número de colunas e linhas do display.

Sintaxe: `lcd.begin(colunas,linhas).`

- `setCursor()`

Posiciona o cursor do LCD nas coordenadas passadas pelo parâmetro, para que o texto possa ser escrito na posição desejada.

Sintaxe: `lcd.setCursor(coluna, linha)`

- `Home()`

Posiciona o cursor no canto superior esquerdo do *display*. Equivale ao `setCursor(0,0)`.

Sintaxe: `lcd.home(coluna, linha).`

- `Print()`

Escreve o conteúdo (*char, int, string, byte ou long*) na tela do LCD, na posição atual do cursor. O parâmetro base é opcional e indica apenas a base em que os números serão impressos (BIN para binário, DEC para decimal, OCT para octal e HEX para hexadecimal).

Sintaxe: `lcd.print(conteúdo, base).`

- `Clear()`

Limpa a tela do LCD e posiciona o cursor na extremidade superior esquerda.

Sintaxe: `lcd.clear().`

- `createChar()`

Cria um caractere customizado para ser utilizado no *display*, até, no máximo, 8

caracteres. O símbolo criado é formado por um *array* de 8 *bytes*, em que os 5 bits menos significativos de cada *byte* determinam os pixels de cada linha.

Sintaxe: `lcd.createChar(numero, caractere)`.

3.2.3 TIPOS DE VARIÁVEIS E PRINCIPAIS FUNÇÕES

As variáveis são regiões da memória responsáveis por armazenar informações atribuídas pelo programador. Esta informação pode ser, por exemplo, um número, caractere ou uma sequência de texto. (CHAVIER, 2016)

No caso dos módulos Arduino que usam processador ATmega, os tipos mais comuns de dados que utiliza-se são:

- *Boolean*: valor verdadeiro (*true*) ou falso (*false*);
- *Char*: um caractere;
- *Byte*: um *byte*, ou sequência de 8 bits;
- *Int*: número inteiro de 16 bits com sinal (-32768 a 32767);
- *Long*: número inteiro de 32 bits com sinal (-2147483648 a 2147483647);
- *Float*: número real de precisão simples (ponto flutuante);
- *Double*: número real de precisão dupla (ponto flutuante);
- *String*: sequência de caracteres;

As funções são uma sequência de comandos possuem um determinado valor de retorno, responsável por executar um valor anteriormente atribuído cada vez que a função é chamada no código.(CHAVIER, 2016)

- Função *pinMode()*

Esta função permite configurar um pino específico para se comportar como um pino de entrada ou de saída. Os pinos do microcontrolador Atmega podem fornecer até 40 mA de corrente para outros componentes ou circuitos.(REIS, 2015)

Sintaxe da função: *pinMode*(pino, modo)

Parâmetros: pino é o número do pino que queremos configurar; modo pode ser: *INPUT*

(entrada) ou *OUTPUT* (saída).

- Função `digitalWrite()`

Esta função escreve um valor *HIGH* (alto) ou *LOW* (baixo) em um pino digital que tenha sido configurado como *OUTPUT* (saída). O valor *HIGH* equivale a uma tensão de 5V e o valor *LOW*, a uma tensão elétrica de 0V. Caso o pino tenha sido configurado como *INPUT* (entrada), a função `digitalWrite()` irá habilitar ou desabilitar o *pullup* interno do pino de entrada.(REIS, 2015)

Sintaxe: `digitalWrite(pino, valor)`

Parâmetros: pino: é o número do pino considerado; valor: *HIGH* ou *LOW*.

- Função `digitalRead()`

Esta função permite ler o valor a partir de um pino digital especificado, sendo os valores possíveis *HIGH* ou *LOW*.

Sintaxe: `digitalRead(pino)`

Parâmetro: pino é o número do pino cujo valor deseja-se ler.

- `Setup()`

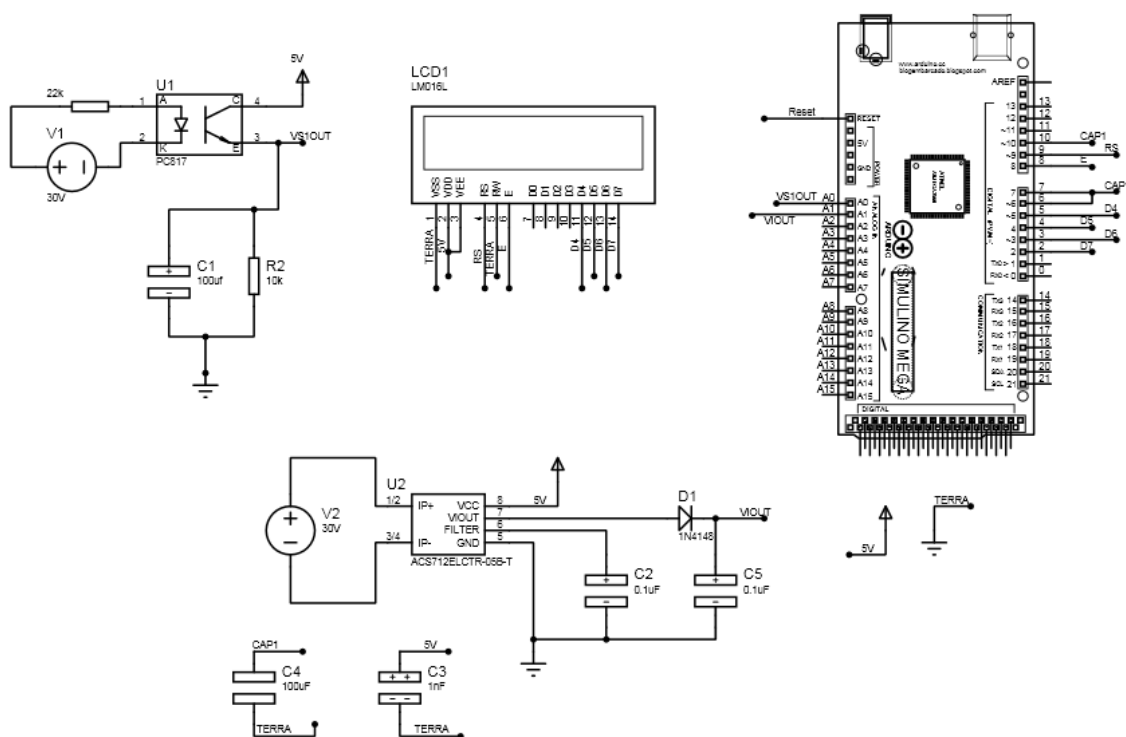
No Arduino a função `setup()` é chamada no momento em que o programa inicia. É utilizada para inicializar variáveis, definir os modos de entrada ou saída dos pinos, indicar bibliotecas, etc.

4 MATERIAL E MÉTODOS

Inicialmente, foi utilizado software para a criação do projeto inicial do circuito de medição de corrente e tensão. Este circuito teve como base as bibliotecas Arduino 4.0 e *LiquidCrystal*. Após a conclusão do projeto inicial, foi desenvolvido o algoritmo específico para o protótipo. Este algoritmo utilizou o software IDE e foi desenvolvido em linguagem C.

4.1 PROTÓTIPO PROPOSTO

Figura 15 - Diagrama do protótipo



FONTE: Autor, 2018.

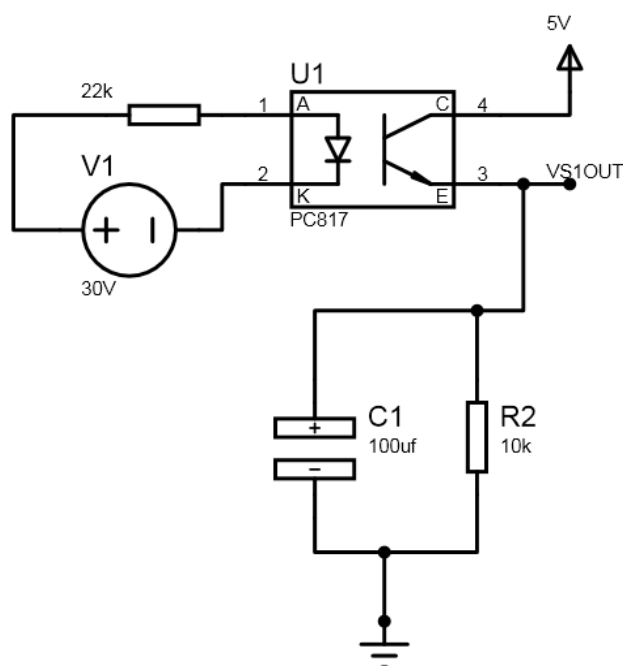
No canto superior esquerdo pode-se observar o circuito sensor de tensão, que utiliza como base um opto acoplador com diodo foto-emissor e um foto-transistor em seu interior.

Na parte inferior está o conjunto sensor de corrente, que utiliza como base o circuito integrado de efeito Hall ACS712, que produz uma tensão proporcional a corrente que flui entre os pinos IP. Na parte superior tem-se um display LCD, responsável por exibir os dados referentes a potência consumida. Já na parte superior

direita está a placa Arduino, que fará a junção dos sensores e executará o algoritmo desenvolvido, permitindo assim o funcionamento do protótipo.

4.2 CIRCUITO SENSOR DE TENSÃO

Figura 16 – Esquemático do circuito sensor de tensão



FONTE: Autor, 2018.

O opto acoplador utilizado no circuito sensor de tensão da figura 17 realiza o isolamento do diodo foto-emissor com o foto-transistor através de um contato luminoso. Isso significa que a passagem de tensão depende da intensidade da luz emitida pelo LED acoplado ao diodo, que por sua vez é determinada pela resistência aplicada ao anodo do diodo.

Neste sensor, usou-se ainda uma fonte de tensão de 30 V. Um resistor de 22 kΩ na entrada do anodo e outro de 10kΩ ligado ao pino analógico de tensão VS1OUT. Por fim, um capacitor de 100uf, também ligado ao VS1OUT.

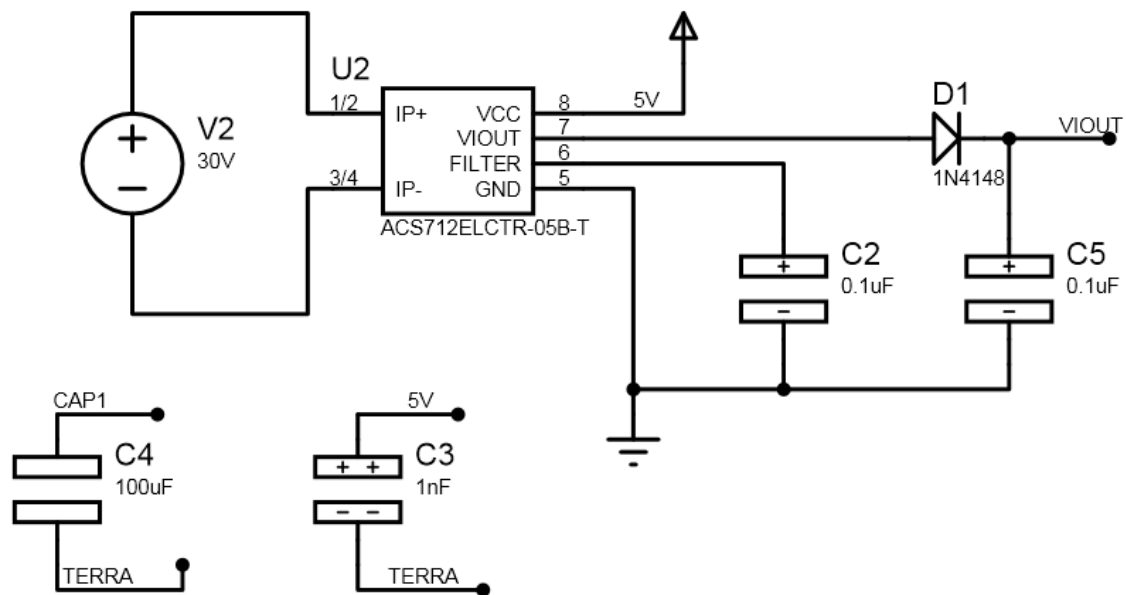
4.3 CIRCUITO SENSOR DE CORRENTE

O conjunto mostrado na figura 17 representa o sensor de corrente do protótipo, que possui o circuito integrado ACS712, que gera uma tensão proporcional a corrente

que flui entre os pinos IP.

Este circuito possui ainda um diodo retificador 1N4148, de 200mA e tensão reversa de 100V, ligado em paralelo com um capacitor de 0.1uF, cujo funcionamento pode ser descrito como um circuito retificador que permite a leitura de corrente alternada através dos pinos IP.

Figura 17 – Esquemático do circuito sensor de corrente



FONTE: Autor, 2018.

Ambos estão ligados ao pino analógico de entrada do arduino VIOU, que é o valor da tensão proporcional a corrente primária medida. Três capacitores de 1nF, 0.1uF e 100uF, ligados aos pinos 5V (ou VCC), FILTER (ACS712) e ao pino digital CAP1, respectivamente. Tomou-se como base uma tensão de 30 V durante a realização das medições.

4.4 ELABORAÇÃO DO CÓDIGO

A elaboração do algoritmo foi inteiramente desenvolvida através do software IDE, baseada em linguagem C. Inicialmente definiu-se os pinos 14 e 15 como entrada (INPUT). Após isso, usou-se a função `Serial.begin(9600)` para iniciar a porta serial e definiu-se a biblioteca *LiquidCrystal* como base para a interação entre o circuito e o LCD de 16 pinos.

Lista 1

```
// Protótipo para Medição de corrente e tensão
// Utilizando PC817 e ACS712

/ Adicionando as bibliotecas
#include <LiquidCrystal.h> //Carrega a biblioteca LCD

LiquidCrystal lcd(11, 8, 5, 4, 3, 2); //Define os pinos que serão
ligados ao LCD
```

Atribuiu-se também as variáveis Long e Float para determinar parâmetros como a tensão e a corrente nas entradas analógicas do arduino.

Lista 2

```
#define vs1Pin 14 // Pino 14 - Leitura da Tensão
#define viPin 15 // Pino 15 - Leitura da corrente

long valorTensao;
long volts;
long valorCorrente;
float amper;
long amper_aux;
float potencia;
float consumo;
float consumo_aux;
```

A partir disto, foram atribuídas estruturas de controle e operadores visando o cálculo da potência em função do valor da tensão V e da corrente I. O valor da corrente foi encontrado através da atribuição de uma variável absoluta, onde observou-se a necessidade de criação de uma relação que permitisse uma aproximação maior dos valores de medição. Após encontrar o valor da potência em Watts, converteu-se este valor para KWh, dividindo-o por 1000.

Lista 3

```
valorTensao = analogRead(vs1Pin); // Atribui a variável
valorTensao o valor do pino vs1Pin
volts = valorTensao * 5 / 1024;

valorCorrente = analogRead(viPin);
amper = (valorCorrente * 2.5 / 511 - 2.48) * 30 / 2.5;

potencia = amper * volts;
potencia = potencia / 1000;
```

Após isso, multiplicou-se o valor da potencia encontrada pela tarifa em questão, (que pode variar conforme a região) e achou-se o valor atribuído à variável consumo_aux.

Lista 4

```
consumo_aux = potencia * 0.3907;  
consumo = consumo + consumo_aux;
```

Por fim, configurou-se as funções de amostragem do display LCD, utilizando-se funções como lcd.print(), ldc.clear() e lcd.setCursor().

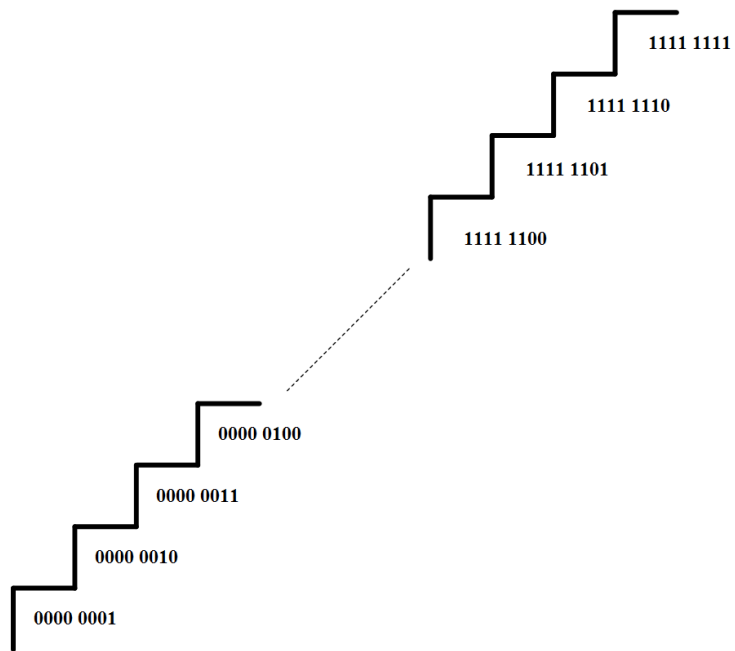
4.5 CONVERSOR ANALÓGICO DIGITAL

Durante a criação do algoritmo, observou-se a necessidade de implementação de uma relação entre a tensão e o número de bits no qual o hardware (arduino Mega) seria capaz de oferecer em cada pino. Para isso, utilizou-se o princípio de funcionamento de um conversor analógico-digital, já disponibilizado na própria placa arduino.

Para descrever seu funcionamento, pode-se utilizar a analogia da escada, na qual cada degrau representa o número de bits suportados por um conjunto de números binários, sejam eles decimais, hexadecimais, ou qualquer outro. Dentro desse contexto, existe uma relação de proporcionalidade entre o número de bits e o número de degraus, onde quanto maior o número de bits, maior o tamanho da escada e, consequentemente, maior a escala.

No caso do arduino Mega2560, o mesmo oferece 10 bits por pino, ou seja, este possui capacidade binária de 1024 bits por pino. Desta forma, tendo-se como base uma tensão de 5 V para o circuito sensor de tensão, estabeleceu-se a relação da equação 5 entre tensão e o número de bits disponibilizados por pino, cujo resultado serviria de utilização para o cálculo dos valores referentes as tensões medidas.

Figura 18 - Descrição do funcionamento de um conversor A/D

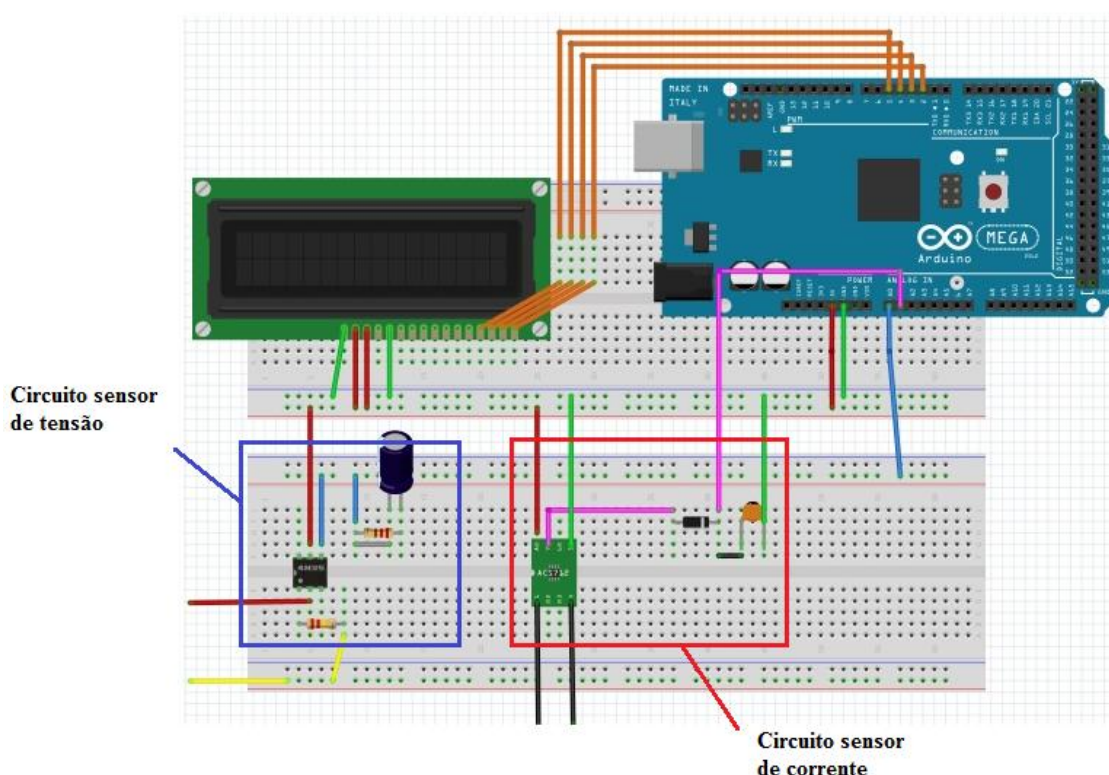


FONTE: Autoria própria (2018)

5. RESULTADOS E DISCUSSÕES

O protótipo final foi montado em uma *protoboard* dupla, dividindo-se entre o medidor de energia e a placa Arduino juntamente com o LCD. A figura 20 mostra, esquematicamente, como foi organizada a montagem do protótipo na *protoboard*, visando facilitar a compreensão do leitor acerca das ligações entre os dispositivos.

Figura 19 - Diagrama da montagem do protótipo



FONTE: Autoria própria

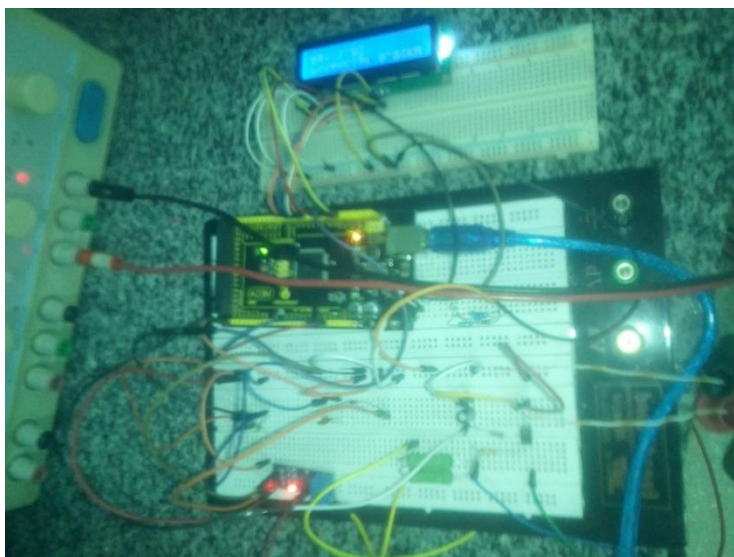
Na figura 20, utilizou-se um opto acoplador versão 4N35, onde na montagem real usou-se um opto acoplador do tipo PC817. Isso se deve as limitações dos circuitos impostas pelo software utilizado durante a realização deste desenho.

Nesta mesma figura, pode-se observar o sensor de tensão no canto inferior esquerdo, composto por um opto acoplador PC817, um resistor de 22 k Ω ligado ao pino 1 em série com uma fonte de 30 V, cuja representação não pôde ser feita no desenho também devido a limitações de software. No pino 4, ligou-se um resistor de 10 k Ω em

paralelo com um capacitor de 100uf. Observa-se que todo o circuito foi alimentado através do pino 5V do arduino.

Já no circuito responsável pela medição de corrente da esquematização, utilizou-se uma fonte de tensão de 30 V nas entradas dos pinos IP. Na figura 21, observa-se o resultado real da montagem descrita pela esquematização da figura anterior.

Figura 20 - Resultado final da montagem



FONTE: Autoria própria

Após a compilação do algoritmo no IDE e posterior carregamento no arduino, observou-se a necessidade de transformar-se a variável de corrente em variável absoluta, pois considera-se que o fornecimento de energia realizado pela fonte é contínuo. Diante disso, foi necessária a criação de uma relação que possibilite uma maior aproximação dos valores obtidos na medição de corrente pelo protótipo.

Após a análise do *datasheet* do circuito ACS712, pode-se notar as seguintes proporcionalidades entre valores de tensão e corrente: durante o estado de saturação, o circuito apresentava uma tensão de saída de 5V proporcionais a uma corrente primária de entrada de 30A entre os pinos IP. Dessa forma, obteve-se uma tensão de saída igual a 2.48V para uma corrente de entrada de 0A. Partindo disso, pode-se criar a seguinte relação:

$$y = \left(\frac{x - 2.48}{2.52} \right) \cdot 30 \quad (5)$$

Onde a variável x representa os valores atribuídos no código referentes aos dados de corrente medidos no protótipo.

$$X = \frac{\text{valorCorrente}(2.5)}{511} - 2.48 \quad (6)$$

Substituindo-se a equação 6 na equação 5, obtém-se o seguinte resultado:

$$Y = \frac{Vc(2.5) - 2.48(1+511)}{(511)(2.52)} \cdot 30 \quad (7)$$

Onde o Y representa a variável absoluta, obtida através da relação entre os valores de corrente mínima e de saturação, ambas medidas durante os testes realizados. A figura 22 mostra como se deu a apresentação dos resultados das medições iniciais, tendo como base uma aplicação de tensão de 30V.

Figura 21 - Resultados apresentados no display



FONTE: Autoria própria

A tabela 5 mostra os resultados obtidos mediante a aplicação de uma corrente através de uma fonte, comparando-se os valores aplicados com os medidos através do protótipo, para valores entre 0.1A e 1.0A.

Tabela 4 - Resultados para corrente

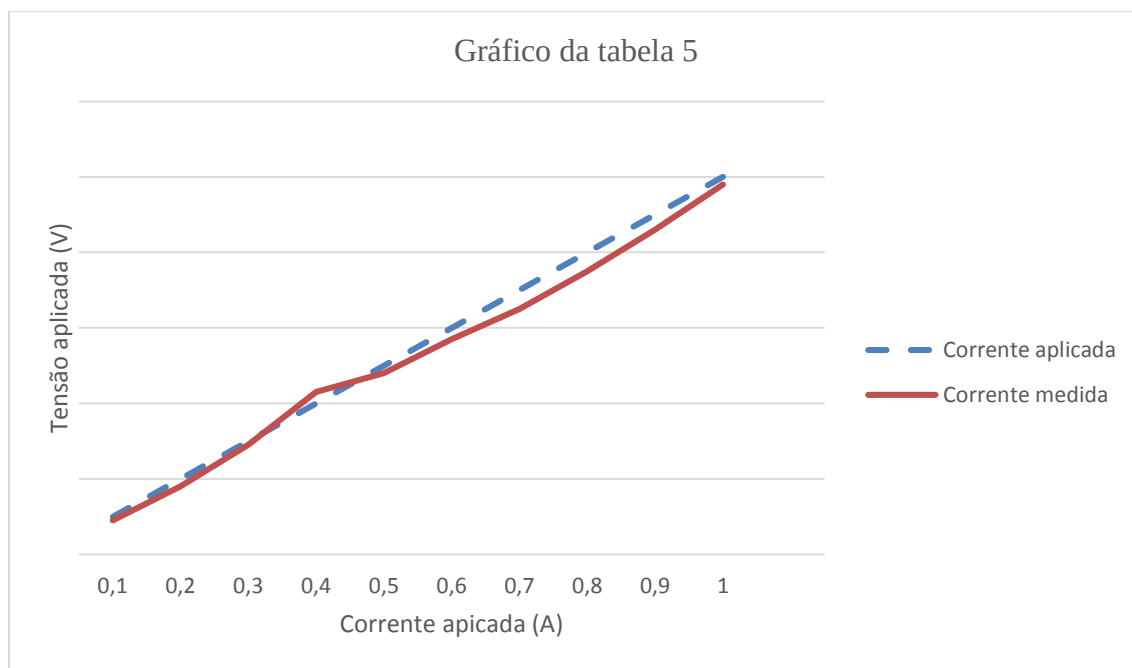
Corrente aplicada	Medição do sensor de corrente	Erro
-------------------	-------------------------------	------

(A)	(A)	(%)
0.1	0.09	10.0
0.2	0.18	10.0
0.3	0.29	3.33
0.4	0.46	10.0
0.5	0.48	4.00
0.6	0.57	5.00
0.7	0.65	7.14
0.8	0.75	6.25
0.9	0.86	4.44
1.0	0.98	2.00

FONTE: Autoria própria

O cálculo do erro da tabela 5 foi baseado na seguinte fórmula:

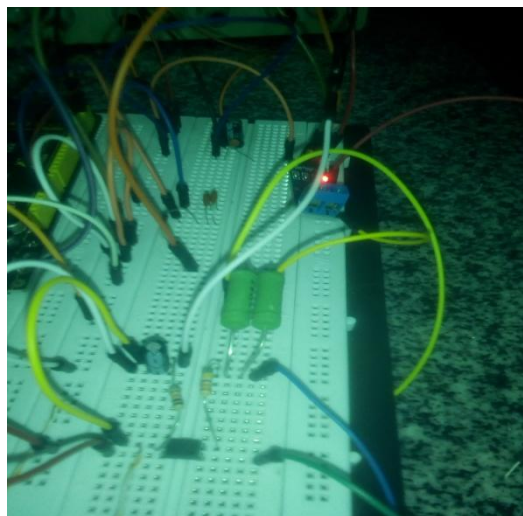
$$|erro\%| = \left(\frac{Valor\ fonte. - Valor\ sensor.}{Valor\ fonte.} \right) \times 100 \quad (8)$$



FONTE: Autoria própria

Com base nos resultados obtidos, calculou-se uma média de erro percentual igual a 6.216%, que se encontra dentro de uma margem aceitável. O menor erro obtido foi de 2 pontos percentuais para corrente de 1 A, enquanto que o maior foi de 10%. Portanto, conclui-se que a diferença entre os valores tende a diminuir conforme a corrente aplicada aumentar.

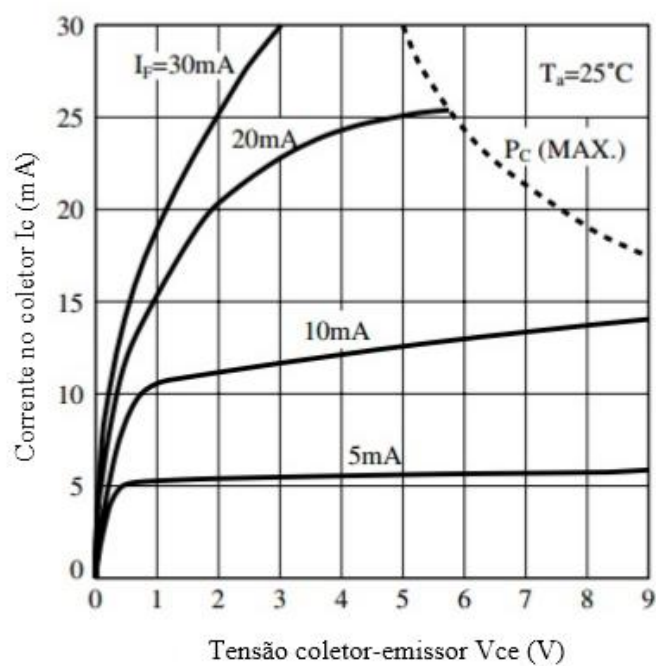
Figura 22 – Resultado da montagem final dos sensores



FONTE: Autoria própria

Em termos das medições referentes a tensões, utilizou-se como base do devido sensor um opto acoplador versão PC817, cujo funcionamento consiste em um foto transistor no qual a tensão do emissor varia com a intensidade de luz fornecida por um LED. A figura 23 mostra o gráfico de tensão Coletor-emissor por Corrente no coletor.

Figura 23 - Gráfico de tensões do transistor para o PC817



FONTE: Sharp microeletronics, 2015

Com base no gráfico da figura 23, foram feitas medidas de tensão buscando-se identificar a região onde a medida de tensão apresenta comportamento próximo ao linear. Com isso, variou-se a resistência de entrada do opto acoplador de modo a fazer com que o mesmo funcione próximo a região ôhmica, tendo como base uma tensão de 30 V.

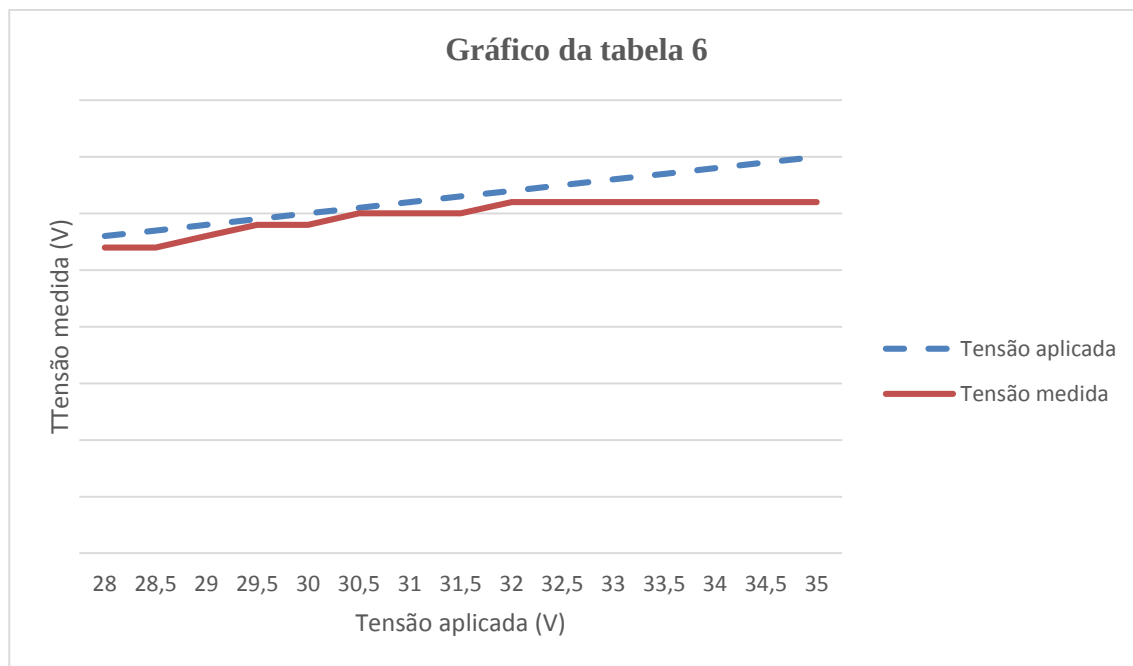
A tabela 6 mostra os valores medidos para tensões onde observou-se a região de linearidade citada anteriormente, comparando-se os resultados através de seu erro percentual.

Tabela 5 - Resultados para medidas de tensão

Fonte de tensão (V)	Medição do sensor de tensão (V)	Erro (%)
28	27	3.57
28,5	27	5.26
29	28	3.45
29,5	29	1.69
30	29	3.33
30,5	30	1.64
31	30	3.22
31,5	30	4.76
32	31	3.12
32,5	31	4.61
33	31	6.06
33,5	31	7.46
34	31	8.82
34,5	31	10.14
35	31	11.42

FONTE: Autoria própria

Com base nos resultados obtidos na tabela 5, observou-se que as maiores diferenças obtidas se deram a partir da região de saturação, ou seja, quando aplicou-se tensões superiores a 31V. Os menores erros foram medidos próximos a região da tensão tomada como base, que variou entre 29,5 V e 30,5 V. Portanto, pode-se concluir que quanto mais próximo o valor de tensão aplicado no opto acoplador for de seu valor base, mais próximo este valor estará de sua região ôhmica, realizando portanto medições mais exatas.



Fonte: Autoria própria

5.1 ESTIMATIVAS DE CUSTOS DO PROJETO

A tabela 5 mostra uma estimativa do somatório dos preços dos componentes do protótipo, sem a inclusão de valores referentes a frete, cujo preço varia conforme a região. Os valores foram encontrados em lojas online e variam com o número de itens comprados.

Tabela 6 - Estimativas de custos do protótipo

Material	Preço por unidades	Unidades	Total
Arduino UNO/Mega	R\$ 45,00	1	R\$ 45,00
ACS712	R\$ 25,00	1	R\$ 25,00
LCD LM016	R\$ 15,00	1	R\$ 15,00
Optoacoplador 4n25	R\$ 2,00	1	R\$ 2,00
Outros	R\$ 15,00	-	R\$ 15,00
			R\$ 102,00

FONTE: Autoria própria

6. CONCLUSÃO

Com base nos resultados obtidos, pode-se considerar que o objetivo básico do trabalho foi alcançado, de modo a obter sucesso na construção de um medidor de energia para corrente contínua em tempo real a um custo acessível. Tendo em vista que a ideia inicial era auxiliar o consumidor residencial a obter dados simultâneos acerca do seu gasto energético, pode-se destacar a importância do incentivo ao mesmo na tomada de medidas voltadas a minimização do desperdício de energia e, conseqüentemente, no aumento da eficiência energética.

Durante os ensaios de medições, foram obtidos erros percentuais menores para os valores de tensão. Isso se dá devido as análises através dos gráficos disponibilizados no *datasheet* do dispositivo opto acoplador, onde pôde-se observar a região ôhmica das tensões do transistor, ou seja, onde as medições de tensão apresentariam comportamento mais próximo ao linear. Já no ensaio de corrente, foi necessária a criação de uma variável absoluta, que permitisse uma melhor relação de proporcionalidade entre os valores aplicados e os medidos.

Contudo, deve-se atentar para o fato de que, devido a limitações nas características elétricas dos dispositivos usados como base para os sensores, estes foram projetados exclusivamente para dispositivos de baixa e média potência, não tendo suporte para grandes equipamentos industriais. Entretanto, pode-se facilmente adaptar o protótipo para uso industrial de modo a substituir os sensores por outros com características elétricas que permitam medições em dispositivos de alta potência.

Dentre as melhorias propostas para o aprimoramento deste protótipo, pode-se destacar a criação de um banco de dados voltado para o armazenamento em plataforma online dos dados referentes as medições feitas no protótipo, seja através da criação de um software, ou mesmo de um aplicativo que seria responsável pela comunicação entre o *shield* e a devida plataforma. Essas informações poderiam ser organizadas na forma de gráficos ou planilhas, à escolha do usuário.

REFERÊNCIAS BIBLIOGRÁFICAS

AGARWAL, T. **Introduction on energy meters.** Disponível em: <<https://www.edgefx.in/introduction-on-energy-meter-different-types-of-energy-meters/>> Acesso em: 29 dez. 2017

ALLEGRO INSTRUMENTS. **Datasheet: Sensor ACS712**, guia de usuário. Publicação eletrônica. 2017.

ANEEL. Agência Nacional de Energia Elétrica. **Relatório de Gestão.** 2015. Disponível em <<http://www.aneel.gov.br/arquivos/pdf>>. Acesso em: 01 abr. 2018.

ANEEL. Agência Nacional de Energia Elétrica. **Acompanhamento da Expansão da Oferta de Geração de Energia Elétrica.** disponível em: < http://www.aneel.gov.br/informacoes-tecnicas/asset_publisher/CegkWaVJWF5E/content/id/14687789> Acesso em: 14 mar. 2018.

ARDUINO. **Boards.** Disponível em : <<https://www.arduino.cc/en/Main/Boards>>. Acesso em: 20 fev. 2018.

ARDUINOECIA. **Como usar o sensor de corrente ACS712.** Disponível em: <<https://www.arduinoecia.com.br/2016/04/como-usar-o-sensor-de-corrente-ac712.html>>. Acesso em 06 jan. 2018.

ASSEMBLY PROGRESSIVO. **Introdução ao Estudo do Assembly.** Disponível em :<<http://www.assemblyprogressivo.net/p/introducao-ao-estudo-do-assembly.html>> Acesso em: 05 fev. 2018.

BARRAGÁN, H. (2017). **Introdução a linguagem Wiring**, disponível em: <<http://wiring.org.co/about.html>>. Acesso em: 16 mar. 2018.

BRAGA, N. C. (2015). **Como funcionam os sensores de Efeito Hall (ART1050).** <<http://www.newtoncbraga.com.br/index.php/como-funciona/6640-como-funcionam-os-sensores-de-efeito-hall-art1050>>. Acesso em: 28 fev. 2018.

CAETANO, P. (2016). **Linguagem Arduino.** disponível em: <<https://professorcaetano.wordpress.com/microcontrolador/linguagem-arduino/>>.

Acesso em: 11 mar. 2018.

CHAVIER, L. F. (2016). **Programação para Arduino**. Disponível em: <<https://www.circuitar.com.br/tutoriais/programacao-para-arduino-primeiros-passos/#varivel>>. Acesso em: 12 mar. 2018.

COPEL. Companhia Paranaense de Energia. **História da energia elétrica**. 2015. Disponível em <<http://www.copel.com/hpcopel/root/index.jsp>> Curitiba, 2015. Acesso em: 01 abr. 2018.

EPE. Empresa de Pesquisa Energética. **Resenha Mensal do mercado de energia elétrica**. Disponível em: <<http://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/resenha-mensal-do-mercado-de-energia-eletrica>>. Acesso em: 03 mar. 2018.

LARGURA, R. **Display LCD 16x2 com Arduino**. Disponível em: <<https://portal.vidadesilicio.com.br/display-lcd-16x2-com-arduino/>>. Acesso em: 16 fev. 2018.

LIMA, C. B. **Os Poderosos Microcontroladores Atmel**. Florianópolis-SC, 2009.

MMA. Ministério do Meio Ambiente. **Eficiência Energética e Conservação de Energia**. Disponível em <[>](http://www.mma.gov.br/clima/energia/eficienciaenergetica)>. Acesso em: 13 jan. 2018.

REIS, F. d. **Conhecendo as funções pinMode, digitalRead e digitalWrite**. Disponível em: <<http://www.bosontreinamentos.com.br/eletronica/arduino/arduino-conhecendo-as-funcoes-pinmode-digitalread-e-digitalwrite/>>. Acesso em: 14 mar. 2018.

RIBEIRO, T. (2013). **Sensor de Corrente por Efeito Hall**. Disponível em: <<https://nerduino.blogspot.com.br/2013/09/sensor-de-corrente-por-efeito-hall.html>>. Acesso em: 15 mar. 2018.

SICA, C. **Sistemas Automáticos com Microcontroladores 8031/8051**. São Paulo, 2006.

SOARES, M. J. (2016). **Os Microcontroladores Avr Atmel**. Disponível em Arnerobotics: <http://www.arnerobotics.com.br/eletronica/Microcontroladores_AVR_ba

sico.htm>. Acesso em: 16 mar. 2018.

TESLABEM. (2017). **Optoacoplador 4n25**. disponível em:
<<http://teslabem.com/optoacoplador-4n25.html>>. Acesso em: 16 mar. 2018.

WALLS, C. (2014). **Programação em Sistemas Embarcados**. Disponível em:
<https://www.eetimes.com/author.asp?section_id=36&doc_id=1323907>. Acesso em:
16 mar. 2018.