

# DESENVOLVIMENTO DE DISPLAY UTILIZANDO O MÉTODO DE DEMULTIPLEXAÇÃO E PERSISTÊNCIA DA VISÃO

*DISPLAY DEVELOPMENT USING THE DEMULTIPLEXING AND PERSISTENCE OF VISION METHOD*

*DESARROLLO DE DISPLAY CON EL MÉTODO DE DEMULTIPLEXACIÓN Y PERSISTENCIA DE VISIÓN*

Allan Stuart Virmond<sup>1</sup>

Adriana Dahmer<sup>2</sup>

Daniel Guergolet Baptistone<sup>3</sup>

Ederson Cichaczewski<sup>4</sup>

## Resumo

Este trabalho foi desenvolvido com a metodologia PBL durante o 2º semestre do 2º ano do curso de Engenharia de Computação da Escola Politécnica da UNINTER, e aplica os conceitos de demultiplexação e persistência da visão para desenvolver um letreiro digital com três displays de 7 segmentos de LEDs, a partir de três circuitos integrados demultiplexadores (um circuito integrado para cada display de 7 segmentos) de oito saídas *Texas Instruments CD74HC4051E*, e o *Arduino UNO R3* para controlar o sistema. Como este modelo de Arduino não tem portas suficientes para ativar simultaneamente três displays de sete segmentos sem o método da demultiplexação, recorreu-se à combinação dessas duas técnicas, cuja utilidade em um protótipo se comprovou pelo resultado positivo, através de estratégia de hardware que permitiu controlar três displays com a implementação, no firmware do Arduino, do controle de demultiplexação com a devida taxa de atualização para persistência visual da informação, além da decodificação para 7 segmentos. Porque o letreiro permite exibir apenas três caracteres simultaneamente, implementou-se a exibição de mensagens de forma que os caracteres alternassem sua posição entre os display em um movimento da direita para a esquerda, possibilitando visualizar mensagens com mais caracteres de maneira corrida.

**Palavras-chave:** Arduino; demultiplexador; display de 7 segmentos.

## Abstract

This work was developed with the PBL methodology during the 2nd semester of the 2nd year of the Computer Engineering course at *Escola Politécnica da UNINTER*, and applies the concepts of demultiplexing and persistence of vision to develop a digital sign with three 7-segment LED displays, from three demultiplexer integrated circuits (one integrated circuit for each 7-segment display) of eight outputs Texas Instruments CD74HC4051E, and the Arduino UNO R3 to control the system. As this Arduino model does not have enough ports to simultaneously activate three 7-segment displays without the demultiplexing method, we resorted to the combination of these two techniques, whose usefulness in a prototype was proven by the positive result, through a hardware strategy that allowed controlling three displays with the implementation, in the Arduino firmware, of the demultiplexing control with the proper refresh rate for visual persistence of the information, besides the decoding for 7 segments. Because the sign allows displaying only three characters simultaneously, the message display was implemented in a way that the characters alternate their position between displays in a right-to-left movement, making it possible to visualize messages with more characters in a race.

**Keywords:** Arduino; demultiplexer; Seven-segment Display.

## Resumen

---

<sup>1</sup> Graduando em Engenharia da Computação pela UNINTER. E-mail: allan199215@gmail.com.

<sup>2</sup> Graduanda em Engenharia da Computação pela UNINTER. E-mail: adrianadahmer@yahoo.com.br.

<sup>3</sup> Graduando em Engenharia Elétrica pela UNINTER. E-mail: dgbaptistone@hotmail.com.

<sup>4</sup> Orientador, Mestre em Engenharia Biomédica, Professor da UNINTER. E-mail: ederson.c@uninter.com.

Este trabajo se desarrolló con la metodología PBL, durante el segundo semestre del segundo año del curso de Ingeniería en Computación de la escuela Politécnica de la UNINTER y aplica los conceptos de demultiplexación y persistencia de visión para desarrollar un letrero digital con tres displays de 7 segmentos de LED, a partir de tres circuitos integrados demultiplexores (un circuito integrado para cada display de siete segmentos) de ocho salidas *Texas Instruments CD74HC4051E*, y el *Arduino UNO R3* para controlar el sistema. Como ese modelo de Arduino no tiene entradas suficientes para activar simultáneamente tres displays de siete segmentos sin el método de demultiplexación, se recorrió a la combinación de esas dos técnicas, cuya utilidad en un prototipo fue comprobada con resultado positivo, por medio de estrategia de hardware que permitió controlar tres displays con el uso, en el firmware de Arduino, del control de demultiplexación con la debida tasa de actualización para persistencia visual de la información, además de la decodificación para siete segmentos. Una vez que el letrero permite exhibir apenas tres caracteres simultáneamente, se implementó la exhibición de mensajes de forma que los caracteres alternaran su posición entre los displays, en un movimiento de derecha a izquierda, permitiendo visualizar mensajes con más caracteres de manera corrida.

**Palabras-clave:** Arduino; demultiplexor; display de 7 segmentos.

## 1 Introdução

Letreiros digitais são utilizados em diversas aplicações, como em painéis informativos de lojas, restaurantes, meios de transporte, assim como para contagem de pessoas em um ambiente, e esta aplicação é o objetivo de um projeto desenvolvido por Griz e Lukacheski (2013), utilizando displays de 7 segmentos para apresentar a contagem digital do fluxo de pessoas em uma biblioteca.

O objetivo do desenvolvimento do letreiro digital aqui proposto é mostrar, em três displays de 7 segmentos, com letras e números, as iniciais do nome da Escola Superior Politécnica Uninter (ESPU) e dois números correspondentes ao ano em que esta pesquisa foi apresentada pela primeira vez (2020) “ESPU20”. Este projeto considera o uso de uma estratégia alternativa ao método trivial de acionamento de displays de 7 segmentos por multiplexação, ao fazer uma varredura por segmentos por meio de demultiplexadores.

Os displays de 7 segmentos utilizados para a demultiplexação com o Arduino Uno são do tipo cátodo comum. Logo, os LEDs são acionados pela aplicação de um nível alto no anodo de cada segmento. O circuito integrado demultiplexador utilizado é o Texas Instruments CD74HC4051E. Resistores realizam a limitação da corrente elétrica que circula por cada um dos segmentos de LEDs.

O programa do Arduino que faz a demultiplexação recebe os valores das letras e dos dois dígitos mostrados no display e aciona os respectivos segmentos, de forma intermitente, porém, a visão humana percebe os dígitos predeterminados exibidos constantemente.

## 2 Fundamentação Teórica

### 2.1 Diodo Emissor de Luz

O crescente uso de displays digitais em várias formas de instrumentação contribuiu para um interesse crescente em dispositivos eletrônicos. Diodos emissores de luz LED são um exemplo de dispositivo baseado na ação de uma junção PN. As luzes de LEDs foram substituídas das lâmpadas incandescentes em várias aplicações ao longo do tempo, o que ocorreu por diversas razões, como o baixo consumo de energia, o tamanho reduzido, a resposta mais rápida a chaveamentos e o ciclo de vida mais longo.

Como o nome indica, o diodo emissor de luz (LED) é aquele que emite luz visível ou invisível (infravermelha) quando energizado. Em qualquer junção p-n polarizada diretamente, existe, dentro da estrutura e principalmente próximo da junção, uma recombinação de lacunas e elétrons. Essa recombinação exige que a energia do elétron livre não ligado seja transferida para outro estado. Em todas as junções p-n semicondutoras, uma parte dessa energia será liberada na forma de calor e outra parte, na forma de fótons. (BOYLESTAD, 2013, p. 36).

Segundo Dutra (2018), enquanto os diodos de silício têm queda de tensão de junção de 0,6 Volts, os LEDs apresentam queda de tensão de cerca de 2 Volts (LEDs comuns). Outro parâmetro a se considerar é a corrente do LED, que deve estar conforme o fornecido pelo fabricante. LEDs comuns têm corrente típica de 20 mili Amperes. Correntes maiores que as recomendadas pelo fabricante podem encurtar a vida útil do componente e queimá-lo.

## 2.2 Microcontroladores e o Arduino

Segundo Dutra (2018), os microcontroladores revolucionaram a eletrônica digital. São chips com alta concentração de circuitos lógicos e microprocessadores, inclusive com memórias do tipo ROM (*Read-Only Memory*) e RAM (*Random Acces Memory*, que é volátil). São a evolução dos microprocessadores, incluem diversos periféricos no mesmo circuito integrado a um custo acessível, que permite a automação de dispositivos e equipamentos. Como exemplo, podemos citar os microcontroladores da empresa Atmel, que podem ser aplicados em plataformas de custo baixo.

O Arduino é como um pequeno computador, um sistema embarcado que interage com seu ambiente, podemos programá-lo para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele. É o que chamamos plataforma de comutação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software (MCROBERTS, 2011). O dispositivo oferece a vantagem, para o usuário, de não exigir conhecimento sobre arquitetura de computadores, além do fato de não necessitar de compiladores, visto ter kit de desenvolvimento próprio (CÁSSIO, [s.d.]).

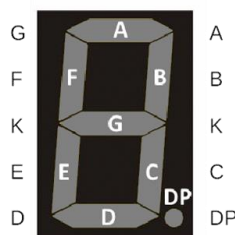
### 2.3 Display de Sete Segmentos

Um display de 7 segmentos recebe este nome por ser formado por sete LEDs, posicionados de tal forma que permita exibição de números de 0 a 9, e das letras de A à F.

O display pode ser do tipo ânodo comum, ou seja, os terminais ânodos de todos os segmentos estão interligados internamente e para os LEDs do display funcionarem o comum deve estar conectado em Vcc, enquanto que o cátodo de cada Led deve estar conectado em GND. O outro modelo é o display de cátodo comum, de forma contrária ao anterior, todos os cátodos estão conectados e interligados ao GND e para funcionar cada LED o ânodo do mesmo deve estar conectado em Vcc. (GRIZ; LUKACHESKI, 2013, p. 18-19).

Os segmentos são nomeados de “A” até “G” e o ponto decimal é chamado *decimal point* (DP), como mostra a Figura 1.

**Figura 1:** Segmentos e disposição dos contatos do display de sete segmentos



**Fonte:** adaptado de <https://www.filipeflop.com/blog/como-construir-um-relogio-com-arduino/>

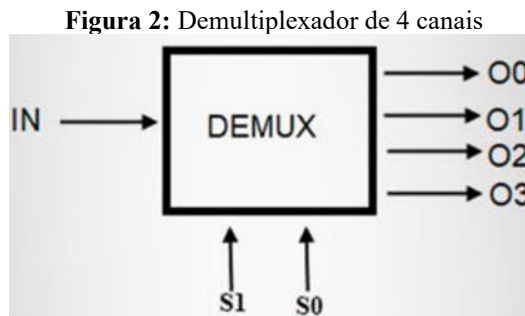
Para montagem do circuito que acionará o display de 7 segmentos é necessário ter um elemento de ativação que permitirá o fluxo de energia através dos LEDs, resistores que limitarão a corrente nos LEDs e garantirão brilho suficiente para visualização, bem como um seletor de segmentos, tarefa do demultiplexador, que definirá os segmentos ativados.

### 2.4 Persistência da Visão

A ideia do método de demultiplexação é dividir as atividades no tempo, utilizando o mesmo meio físico para isso. O método usado neste projeto é a demultiplexação temporal baseada na persistência da visão. Tal método é utilizado, pois, o olho humano é incapaz de perceber a alteração de um sinal luminoso com frequência acima de 24 Hz, ao que chamamos persistência da visão. Isto significa que, se um LED piscar 24 vezes durante um segundo, aos olhos humanos parecerá estar aceso constantemente (LIMA; VILLAÇA, 2012).

### 2.5 Demultiplexador

O demultiplexador é um circuito que recebe uma única entrada e distribui para várias saídas (TOCCI, 2018). Ao ilustrar um demultiplexador (Figura 2) vemos que a operação do dispositivo é multiplicar a saída a partir de uma entrada singular.



Fonte: <http://blog.baudaeletronica.com.br/multiplexadores-e-demultiplexadores/>

No demultiplexador, as variáveis de seleção  $S_0$  e  $S_1$  são controladoras que definirão a qual saída a entrada será direcionada. Seu funcionamento corresponde a *tabela verdade*, como mostra a Tabela 2.

**Tabela 2:** Tabela Verdade de um demultiplexador de 4 canais

$S_1$	$S_0$	O0	O1	O2	O3
0	0	IN	0	0	0
0	1	0	IN	0	0
1	0	0	0	IN	0
1	1	0	0	0	IN

Na Tabela 2 é possível compreender melhor o funcionamento do circuito. Quando ambas variáveis de seleção estiverem em 0 ( $S_0 = 0$  e  $S_1 = 0$ ), a entrada do demultiplexador (IN) será direcionada à saída O0. Quando a variável de seleção for  $S_0 = 1$  e a variável de seleção  $S_1 = 0$ , a entrada do demultiplexador será direcionada à saída O1. Quando a variável de seleção for  $S_0 = 0$  e a variável  $S_1 = 1$ , a entrada do demultiplexador será direcionada à saída O2. Quando ambas as variáveis de seleção estiverem em 1 ( $S_0 = 1$  e  $S_1 = 1$ ), a entrada do demultiplexador será direcionada à saída O3. Vale lembrar que 0 é o nível lógico baixo e 1 é o nível lógico alto, e, quando uma saída for selecionada, será a única a transmitir a informação.

### 3 Metodologia

Depois de compreender o funcionamento dos dispositivos utilizados nesta pesquisa empírica pelo referencial teórico, aplicaram-se os conceitos no desenvolvimento de um

protótipo do letreiro digital. Os materiais e métodos utilizados para implementação do projeto são apresentados nesta seção.

### 3.1 Materiais Utilizados

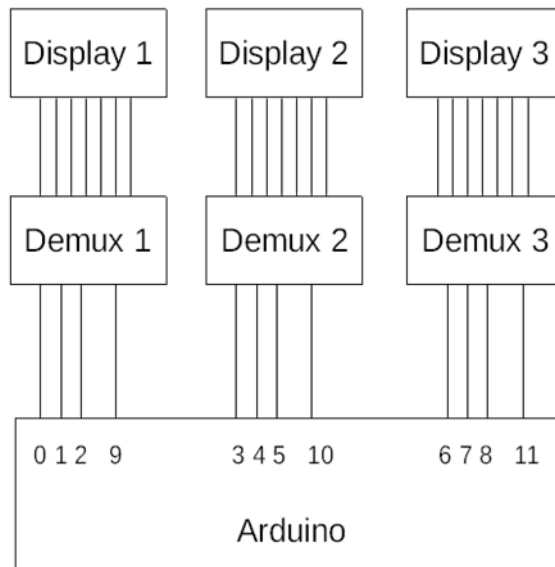
Para os experimentos realizados nesta pesquisa foram utilizados os seguintes componentes eletrônicos:

- Arduíno UNO R3 (1 unidade);
- Resistores de 470 Ohms (21 unidades);
- Displays 7 segmentos D206TK cátodo comum (3 unidades);
- Demultiplexador de 8 saídas Texas Instruments CD74HC4051E (3 unidades);
- Transformador com saída 15 Volts e entrada de corrente alternada em 127 ou 220 Volts;
- Fonte de tensão ajustável baseada no LM317 (1 unidade);
- Protoboard (1 unidade);
- Multímetro digital VC9805;
- Cabos rígidos compatíveis com o tamanho das conexões da protoboard (15 de aproximadamente 30 cm, 15 de aproximadamente 2 cm);
- Software ArduinoIDE;
- Software Autodesk Tinkercad.

### 3.2 Montagem do Letreiro

Inicialmente, elaborou-se um diagrama de blocos de conexão entre o Arduíno, os demultiplexadores e os displays. A Figura 3 ilustra esse diagrama, como referência para montagem do circuito proposto. Os números representam os terminais de conexão do Arduíno e as linhas representam as trilhas de conexão entre os componentes.

**Figura 3:** Diagrama inicial do projeto

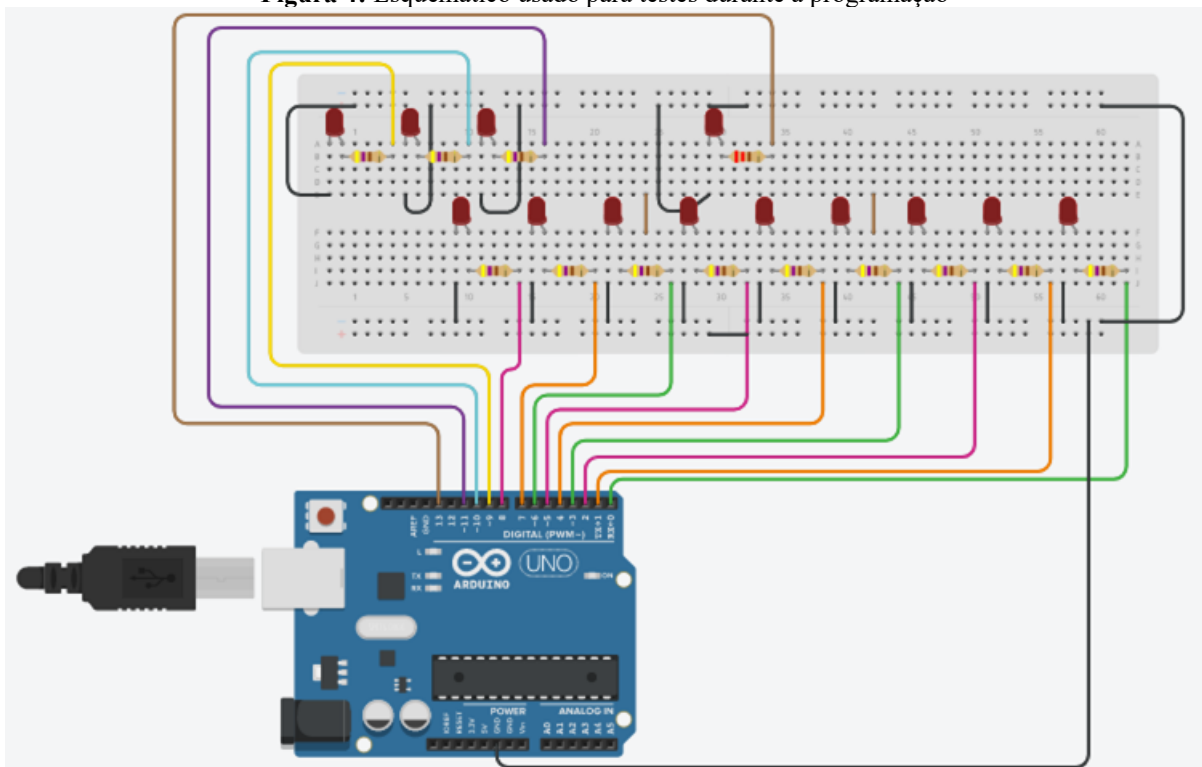


**Fonte:** os autores.

Os números 0, 1, 2, 3, 4, 5, 6, 7, 8 são as conexões da entrada de seleção dos demultiplexadores, e os números 9, 10, 11 são as saídas responsáveis por ativar os displays.

Para os testes iniciais, o circuito ilustrado na Figura 4 apresenta uma montagem simplificada, em que é possível identificar o status de cada saída do Arduino e visualizar a resposta completa do sistema.

**Figura 4:** Esquemático usado para testes durante a programação



**Fonte:** os autores.

No esquemático da Figura 4 é possível ver o estado lógico das entradas de seleção dos demultiplexadores em tempo real, pois, os LEDs logo abaixo, da direita para esquerda, correspondem a cada uma das entradas de seleção separadas por um fio sem função eletrônica, apenas para melhorar a visualização das entradas de seleção do demultiplexador 1, 2 e 3; os LEDs da parte superior representam o segmento a ser ligado na iteração, um para cada display; o LED superior mais isolado dos demais representa o fim do laço principal do Arduino (loop). Todo esquemático foi desenhado por meio do ambiente de simulação *Tinkercad*.

Antes de iniciar a programação foi elaborado um esquemático de como a comutação aconteceria. A programação se iniciou somente após mapeado o comportamento do dispositivo.

Usando como referência a disposição do display de sete segmentos da Figura 1, montou-se uma tabela que corresponde a um único segmento ligado por vez, tendo em vista a escrita “ESP”, que individualmente ligava a partir do segmento A até o segmento G, correspondendo 7 iterações (do 0 ao 6, uma para cada segmento) em cada um dos displays ao mesmo tempo. A Tabela 3 ilustra melhor o procedimento.

**Tabela 3:** Mapa de ciclo de acionamento de segmentos

Iteração	Segmento	Display 1 (E)	Display 2 (S)	Display 3 (P)
0	A	Ligado	Ligado	Ligado
1	B	Desligado	Desligado	Ligado
2	C	Desligado	Ligado	Desligado
3	D	Ligado	Ligado	Desligado
4	E	Ligado	Desligado	Ligado
5	F	Ligado	Ligado	Ligado
6	G	Ligado	Ligado	Ligado

Para desenvolver o algoritmo utilizado nesta pesquisa, informaram-se os vetores cujos índices correspondem a cada um dos padrões responsáveis pela ativação de uma das iterações do padrão. Em seguida, no setup, todas as portas digitais são ativadas em modo de saída. Definiu-se um laço para ligar as saídas conforme foram calibradas, uma etapa por vez de acordo com os vetores que calibram esse padrão. Outro laço mais interno é responsável por diminuir o tempo de comutação entre os padrões para testar a demultiplexação, pois, quanto mais perto de 1, mais perfeita é a demultiplexação, visto que os displays são ligados para parecer um padrão estático em cada um. Os padrões foram configurados de acordo com a Tabela 4, e, da combinação entre a Tabela 3 e a Tabela 4, obtiveram-se os padrões utilizados na programação



do algoritmo, representados na Tabela 5, em que I representa a iteração, S, a entrada de seleção do demultiplexador, A, B e C, os displays 1, 2 e 3, x, as condições de irrelevância, 0, o nível lógico baixo, e 1, o nível lógico alto.

**Tabela 4:** Padrões de configurações de cada segmento

Seleção 2	Seleção 1	Seleção 0	Segmento
0	0	0	A
0	0	1	B
0	1	0	C
0	1	1	D
1	0	0	E
1	0	1	F
1	1	0	G
1	1	1	DP

**Tabela 5:** Padrões utilizados na programação (ESP)

I	S2A	S1A	S0A	S2B	S1B	S0B	S2C	S1C	S0C
0	0	0	0	0	0	0	0	0	0
1	x	x	x	x	x	x	0	0	1
2	x	x	x	0	1	0	x	x	x
3	0	1	1	0	1	1	x	x	x
4	1	0	0	x	x	x	1	0	0
5	1	0	1	1	0	1	1	0	1
6	1	1	0	1	1	0	1	1	0

Para complementar a pesquisa, os padrões foram implementados em três ciclos, o primeiro inicia lentamente a primeira escrita ESP, demonstrando o funcionamento da demultiplexação. Contudo, para exibir os outros caracteres, a multiplexação foi dividida em: ESP (Tabela 5), SPU (Tabela 6), PU2 (Tabela 7), U20 (Tabela 7), correspondendo à escrita ESPU20. Para tanto, foi necessário fazer quatro tabelas ao todo, em que o x corresponde ao estado de irrelevância do padrão, por ser onde a saída que ilumina o segmento naquele ciclo se encontrará desligada.

**Tabela 6:** Padrões utilizados na programação (SPU)

I	S2A	S1A	S0A	S2B	S1B	S0B	S2C	S1C	S0C
0	0	0	0	0	0	0	x	x	x
1	x	x	x	0	0	1	0	0	1
2	0	1	0	x	x	x	0	1	0
3	0	1	1	x	x	x	0	1	1
4	x	x	x	1	0	0	1	0	0
5	1	0	1	1	0	1	1	0	1
6	1	1	0	1	1	0	x	x	x

**Tabela 7:** Padrões utilizados na programação (PU2)

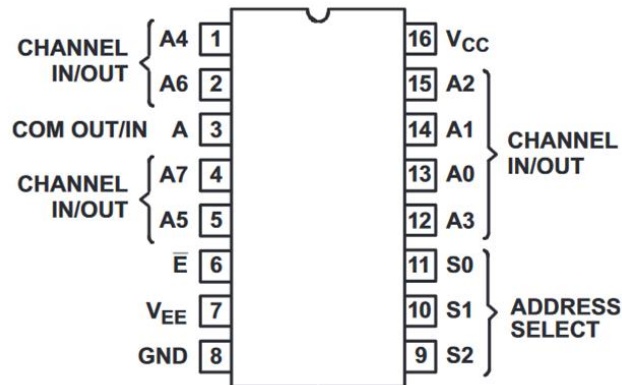
I	S2A	S1A	S0A	S2B	S1B	S0B	S2C	S1C	S0C
0	0	0	0	x	x	x	0	0	0
1	0	0	1	0	0	1	0	0	1
2	x	x	x	0	1	0	x	x	x
3	x	x	x	0	1	1	0	1	1
4	1	0	0	1	0	0	1	0	0
5	1	0	1	1	0	1	x	x	x
6	1	1	0	x	x	x	1	1	0

**Tabela 8:** Padrões utilizados na programação (U20)

I	S2A	S1A	S0A	S2B	S1B	S0B	S2C	S1C	S0C
0	x	x	x	0	0	0	0	0	0
1	0	0	1	0	0	1	0	0	1
2	0	1	0	x	x	x	0	1	0
3	0	1	1	0	1	1	0	1	1
4	1	0	0	1	0	0	1	0	0
5	1	0	1	x	x	x	1	0	1
6	x	x	x	1	1	0	x	x	x

Para implementação do circuito foi necessário consultar a folha de dados (*datasheet*) para identificar a disposição dos terminais do demultiplexador, demonstrado na Figura 5.

**Figura 5:** Disposição dos terminais do demultiplexador CD74HC4051E



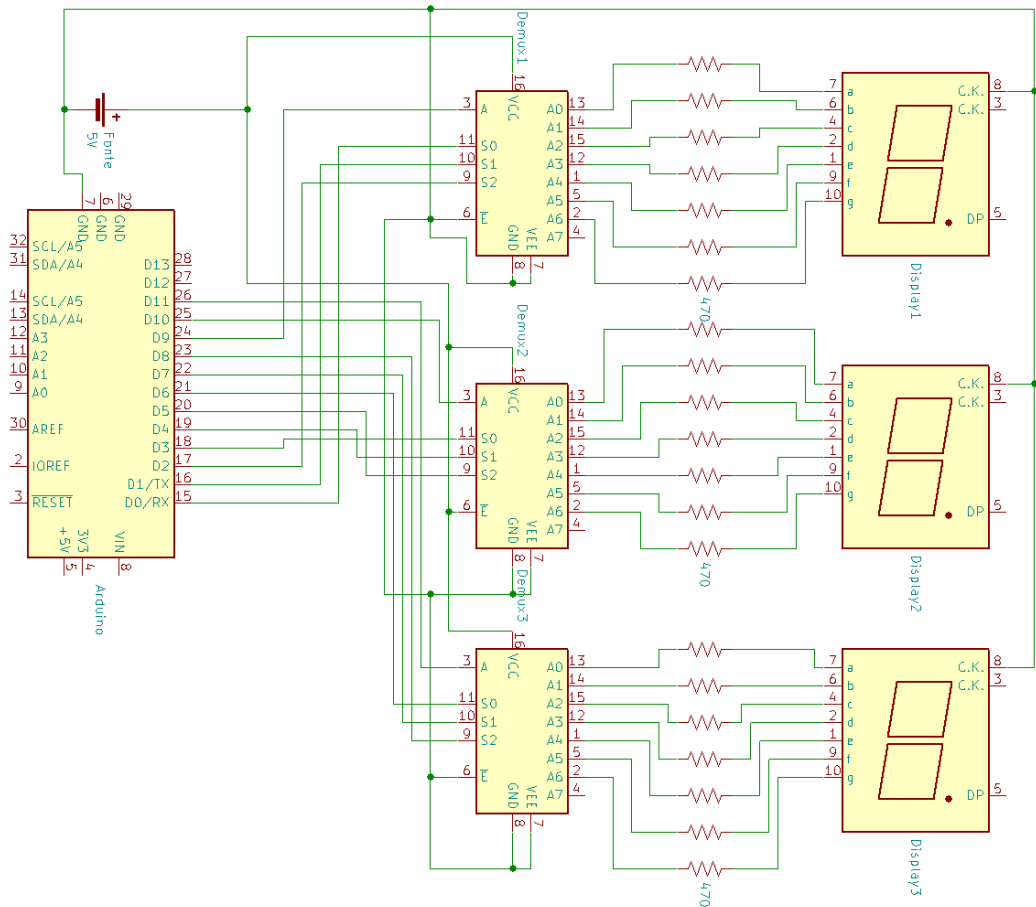
**Fonte:** Texas Instruments (2020).

A Figura 5 mostra a utilização do dispositivo como demultiplexador digital, pois, trata-se de um componente 4 em 1 que pode operar em uma das quatro configurações: multiplexador analógico, multiplexador digital, demultiplexador analógico e demultiplexador digital. CHANNEL IN/OUT, são os canais de saída; ADDRESS SELECT é o seletor de saídas; COM OUT/IN é a entrada distribuída entre as saídas de acordo com a combinação acionada em ADDRESS SELECT, que são os canais de seleção S0, S1 e S2; a entrada E é a habilitação do chip, ou seja, desativa tanto a entrada como todas as saídas; VCC é a entrada de alimentação em 5 Volts; GND é o terminal de referência 0 Volt da fonte e VEE é o referencial adotado como nível lógico baixo.

A representação do diagrama esquemático do circuito utilizado neste trabalho está ilustrada na Figura 6.

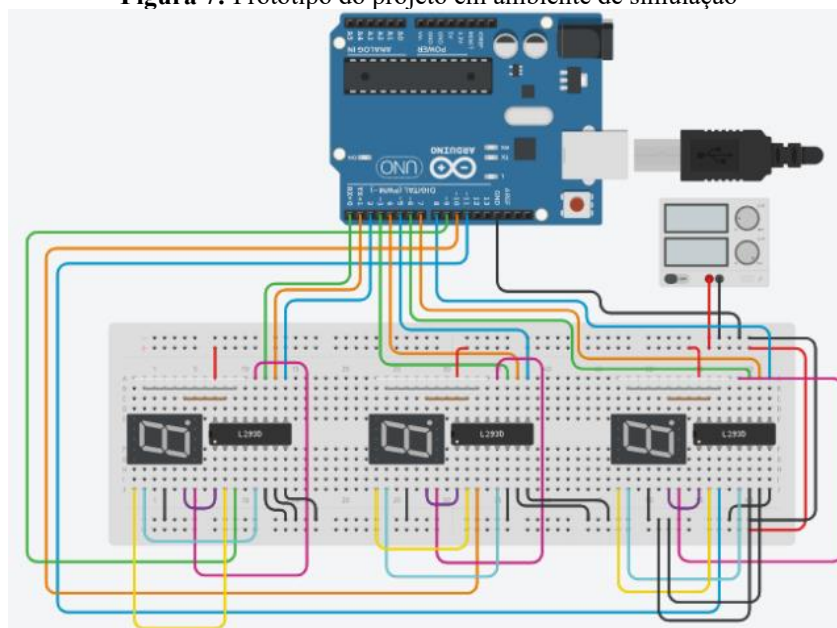
A Figura 7 apresenta a montagem completa do circuito no ambiente de simulação Tinkercad, com o protótipo do projeto.

Figura 6: Esquema elétrico do projeto



Fonte: os autores.

Figura 7: Protótipo do projeto em ambiente de simulação



Fonte: os autores.

#### 4 Resultados e Discussão

Para configuração do demultiplexador foi necessária a conexão da porta VEE com o referencial negativo do sistema, ativando assim o modo demultiplexador digital. Também foi preciso conectar o terminal E para ativar o recurso de ENABLE, que funciona como um interruptor do circuito quando está em nível lógico alto, desativando todas as saídas e entradas, alternando para o modo de alta impedância. Para desligar esse recurso, conectou-se o terminal E ao referencial 0 Volt do circuito, habilitando por completo a funcionalidade do demultiplexador como dispositivo digital, em que o nível lógico baixo é 0 Volt e o nível lógico alto é 5 Volts, totalmente compatível com o funcionamento do Arduíno.

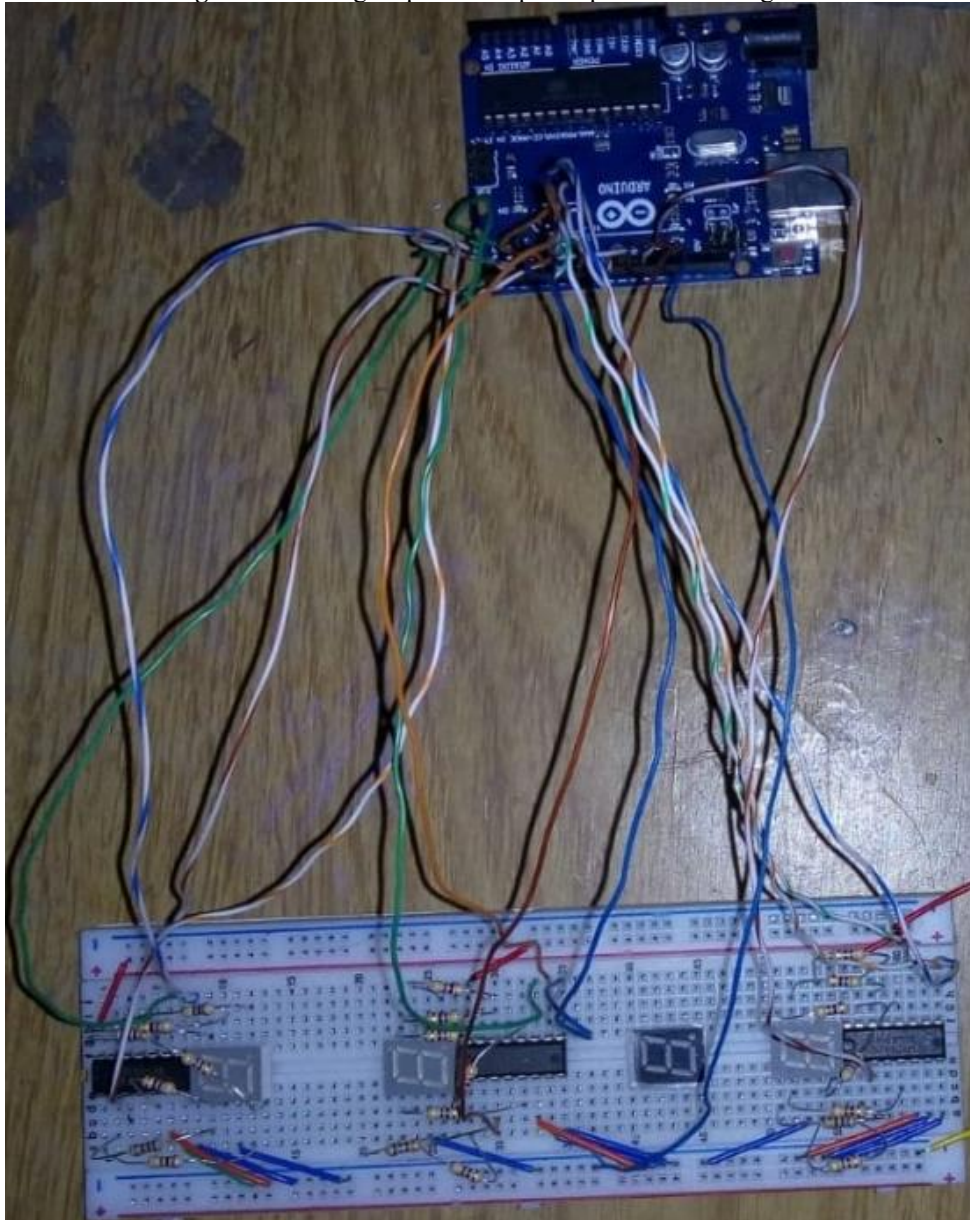
Para identificar os contatos corretos de acionamento dos segmentos dos displays, acionou-se o multímetro em modo teste de continuidade, conectando o catodo (K) no negativo e os ânodos no positivo em cada um de seus terminais que, quando mapeado, possibilitou conectar corretamente as saídas do demultiplexador em cada segmento. A disposição do display de 7 segmentos foi representada na Figura 1, cujas letras em torno representam os terminais que conectam o anodo de cada segmento.

Percebeu-se a característica de persistência da visão, porque foi necessário ajustar a taxa de atualização dos LEDs dos displays de 7 segmentos para visualizar claramente os caracteres exibidos. Quando a taxa não estava corretamente ajustada, notava-se o processo de varredura no acionamento dos segmentos dos displays. O melhor intervalo de acionamento entre cada segmento foi de 1 ms, portanto, 1 kHz, isto significa que cada display é atualizado em uma taxa aproximada de 143 Hz.

A montagem prática do circuito do projeto é apresentada na Figura 8. Percebe-se um quarto display de 7 segmentos na protoboard em cor mais escura, mas este não está conectado ao circuito e não faz parte do letreiro.

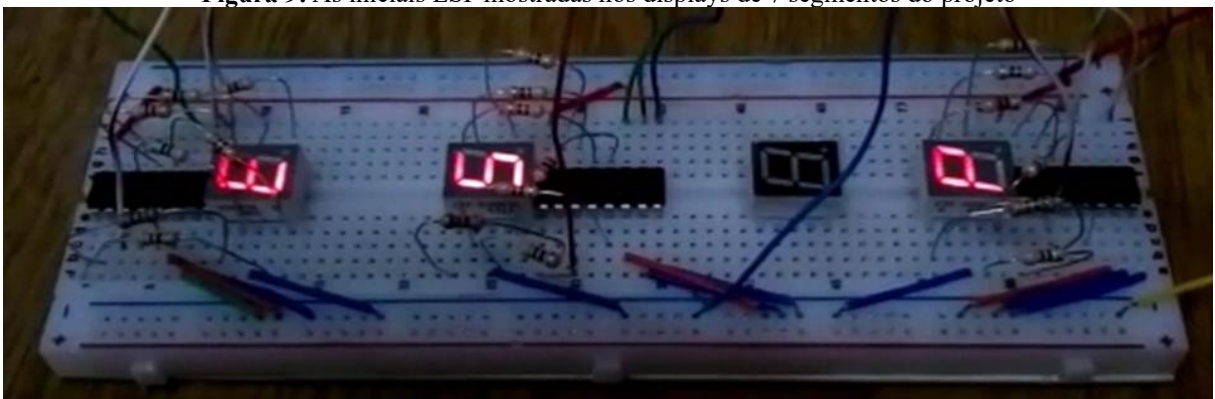
A Figura 9 apresenta em detalhe o circuito do projeto em operação, sendo possível identificar nos displays de 7 segmentos as letras ESP.

**Figura 8:** Montagem prática do protótipo do letreiro digital



**Fonte:** os autores.

**Figura 9:** As iniciais ESP mostradas nos displays de 7 segmentos do projeto



**Fonte:** os autores.

## 5 Considerações finais

Utilizando três displays foi possível implementar a demultiplexação dos segmentos, porque o Arduino Uno R3 só comportaria dois displays sem o ponto decimal preenchendo todas as saídas digitais, sem nenhum processo de multiplexação dele. Entretanto, a demultiplexação e a implementação de um firmware para controlar a demultiplexação e a decodificação para 7 segmentos possibilitaram conectar três displays ao todo. Ademais, foi possível verificar a eficácia da metodologia usada para o acionamento dos segmentos dos displays com persistência da visão.

Como trabalhos futuros para otimizar ainda mais o hardware, pode-se buscar melhorar o método a partir de uma multiplexação para adicionar dígitos ao letreiro.

## Referências

BOYLESTAD, R. L.; NASHESKY, L. **Dispositivos eletrônicos e teoria de circuitos**. São Paulo: Pearson Education do Brasil Ltda, 2013. p. 36-37.

CÁSSIO, H. O que é Arduino? **Protovie**. [s.d.]. Disponível em: <http://protovie.com/o-que-e-arduino>. Acesso em: 21 set. 2021.

DUTRA, E. S. D. S. **Eletrônica**. Porto Alegre: SAGAH, 2018. p. 124.

GRIZ, D. M.; LUKACHESKI, S. P. **Contador Digital De Fluxo De Pessoas**. 2013. Trabalho de Conclusão de Curso (Tecnólogo) — Universidade Tecnológica Federal do Paraná, Medianeira, 2013.

LIMA, C. Borges de; VILLAÇA, M. V. M. **AVR e Arduino: Técnicas de Projeto**. 2. ed. Joinville: Clube dos Autores, 2012.

MCROBERTS, Michael. **Arduino básico**. 1. ed. São Paulo: Novatec, 2011.

MORAES, Fernando. Circuitos Combinacionais – Multiplexador e Demultiplexador. **Baú da Eletrônica**. 27 fev. 2019. Disponível em: <http://blog.baudaeletronica.com.br/multiplexadores-e-demultiplexadores/>. Acesso em: 21 set. 2021.

TEXAS INSTRUMENTS. **CDx4HC405x, CDx4HCT405x High-Speed CMOS Logic Analog Multiplexers and Demultiplexers**. Technical Documents. May, 2019. Disponível em: <https://www.ti.com/lit/ds/symlink/cd74hc4052.pdf>. Acesso em: 21 set. 2021.

TOCCI, R.; WIDMER, N. S. **Sistemas Digitais – Princípios e Aplicações**. 12. ed. São Paulo: Pearson, 2018.