Laboratório de Sistemas Digitais
ENG1004

Gabriel Sadigursky, Thiago Tavolara e Vitor Simon de Souza
Grupo: C
Professor: Marcelo Gotz
Turma: A

22 de janeiro de 2023

Sumário

1 Introdução **2**

2 Codificação dos Dados **3**
2.1 Codificação enviada do Arduíno para Altera DE2 3
2.2 Codificação enviada da Altera DE2 para o Arduíno 3

3 Desenvolvimento **5**
3.1 PWM 5
3.2 Decodificador 8
3.3 Página Web 11
3.4 Chaves da Placa Altera DE2 14
3.5 Codificador 15
3.6 Circuito Principal 17
3.6.1 Bloco S1S0 - Conexão de controle entre Decodificador e Registrador das Chaves 18

4 Conclusão **21**

1 Introdução

Diante da ideia geral do projeto, que é realizar a comunicação entre algum agente e a FPGA apresentada pelo professor, surgiu ao grupo a ideia de utilizar o microcontrolador Arduíno para realizar esta comunicação. Além disso, por conta do agente comunicador prover conexão Wi-Fi, a ideia é que se usufrua dessa utilidade para o aprendizado do grupo.

A proposta do sistema é que, como mostrado na Figura 1, a partir de uma página Web seja possível enviar e receber dados da FPGA Altera DE2. Para tanto, esses comandos deverão ser passados para o Arduíno, que, ao estar conectado com o computador, atuará como um intermediário entre a página Web e a placa Altera DE2 (FPGA). A ideia é que, do lado da página Web, seja possível controlarmos 4 LEDs da FPGA e seus níveis de luminosidade. Além disso, ao ativarmos alguma chave na FPGA, os LEDs deverão ser controlados pela própria DE2, tanto físicos quanto os da página Web.

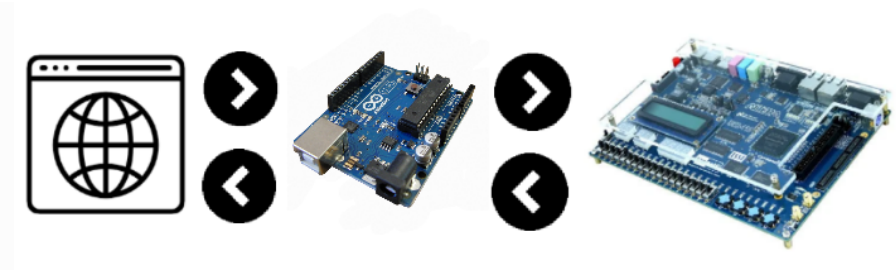

Figura 1: Ilustração básica do sistema.

2 Codificação dos Dados

2.1 Codificação enviada do Arduíno para Altera DE2

Foi montado um protocolo inicial para a codificação dos dados enviados pelo Arduíno e que serão recebidos serialmente pela Altera DE2. O formato da mensagem consiste em um total de 10 bits, compostos por:

- Um bit inicial que indicará o início da mensagem, que valerá sempre 0;
- Dois bits que indicarão o nível de luminosidade do LED 1;
- Dois bits que indicarão o nível de luminosidade do LED 2;
- Dois bits que indicarão o nível de luminosidade do LED 3;
- Dois bits que indicarão o nível de luminosidade do LED 4;
- Um último bit que indicará o final da mensagem, e valerá sempre 1.

Para melhor entendimento, a Tabela 1 exemplifica de maneira gráfica e estruturada como será dado o funcionamento do protocolo.

| 0 | PWM LED 1 | PWM LED 2 | PWM LED 3 | PWM LED 4 | 1 |

Tabela 1: Estrutura das Mensagens

Para uma melhor compreensão dos níveis de luminosidade de cada LED foi criada a Tabela 4, que pode ser visualizada na subseção 3.1.

2.2 Codificação enviada da Altera DE2 para o Arduíno

Assim como foi criada a codificação do Arduíno para a placa Altera DE2, foi necessário criar uma codificação diferente para o retorno da placa para o Arduíno, visto que, com a utilização das chaves da DE2, esse retorno será os estados das chaves. O formato da mensagem de retorno consiste em um total de 10 bits, compostos por:

- Um bit inicial que indicará o início da mensagem, que valerá sempre 0;
- Um bit que indicará o estado da chave 1 (C1);
- Um bit que indicará o estado da chave 2 (C2);
- Um bit que indicará o estado da chave 3 (C3);
- Um bit que indicará o estado da chave 4 (C4);
- Um bit que indicará o estado da chave 5 (C5);
- Um bit que indicará o estado da chave 6 (C6);
- Um bit que indicará o estado da chave 7 (C7);
- Um bit que indicará o estado da chave 8 (C8);
- Um último bit que indicará o final da mensagem, e valerá sempre 1.

Os estados da chaves podem ser ligado ou desligado, como mostra a Tabela 3.

| 0 | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | 1 |

Tabela 2: Estrutura das Mensagens de Retorno

Estado	**Codificação**
Desligado	0
Ligado	1

Tabela 3: Codificação dos estados das chaves

3 Desenvolvimento

A finalidade central do projeto é o controle da Altera DE2 remotamente, adequando uma tecnologia largamente usada com as novas modernidades da indústria 4.0, como a internet das coisas.

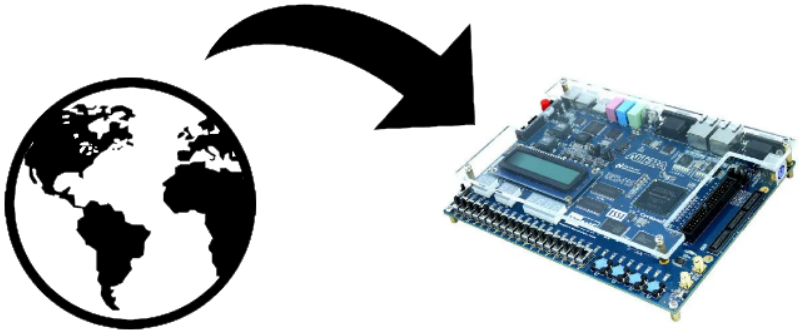

Figura 2: Ilustração controle remoto Altera DE2.

Diante do feedback recebido após a apresentação inicial, foi definido que o projeto consistiria em utilizar a página web para configurar quatro LEDs da FPGA, além de definir um nível de luminosidade específico para cada um. Conforme avançamos no projeto, também decidimos incluir o controle dos LEDs diretamente pela FPGA, de modo que as alterações no estado dos LEDs na FPGA sejam refletidas também na interface do usuário do site.

3.1 PWM

Visando controlar a luminosidade dos LEDs, foi criado uma codificação que pode ser visualizada na Tabela 4.

Porcentagem	**Codificação(S1/S0)**
0	00
33	01
66	10
100	11

Tabela 4: Codificação da Luminosidade

Para cada LED foi criado um *Duty Cycle* no programa principal, como pode ser visto na Figura 3, padronizado para as porcentagens da Tabela 4. Todavia, dentro do bloco *Duty Cycle*, foi definido pelo grupo que este sistema vai de 0 a 255, 2^8, ou seja, a codificação para as porcentagens de luminosidade dentro deste bloco é de 8 bits, como mostra a Figura 4. Para fazer a correlação desses diferentes tamanhos de códigos, 2 bits para 8 bits, foi utilizado um multiplexador, o qual recebe esses 2 bits (S1S0) que selecionam um determinado número no multiplexador. Como forma de minimizar o erro, esse sistema recebe o *MSG* que define se a mensagem passada pelo Arduíno está correta ou não.

Após a definição do *Duty Cycle*, este passa para o próximo bloco, Figura 5, o qual possui um *PWM* de 8 bits para cada LED. Basicamente, o *Pulse Width Modulation* modula a largura de pulso do *clock* através do *Duty Cycle*, limitando a potência de luminosidade do LED.

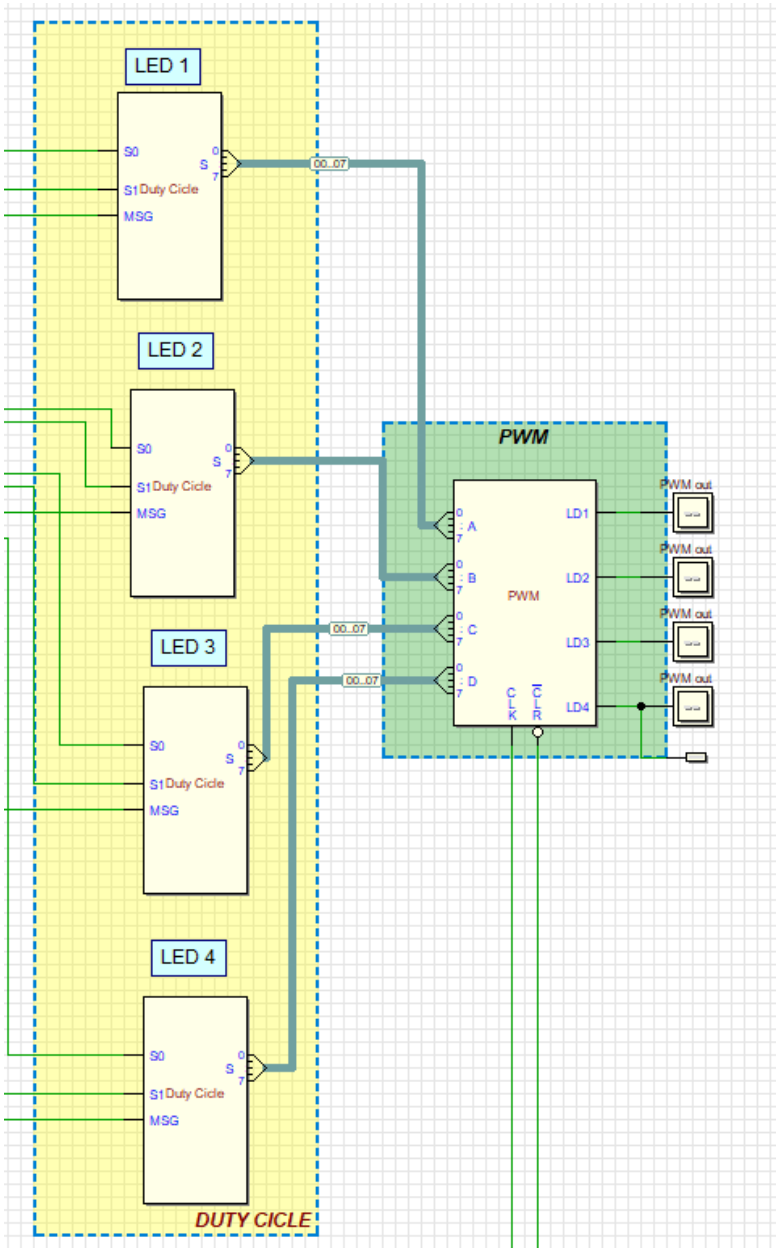

Figura 3: Blocos Duty Cycle e PWM no circuito principal.

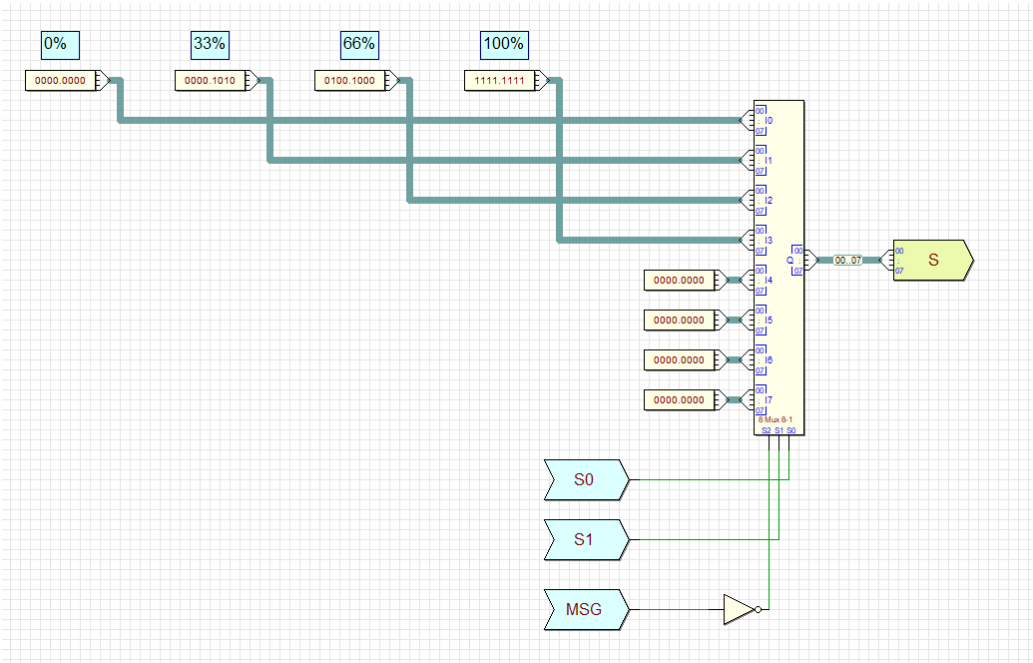

Figura 4: Sistema de funcionamento do bloco Duty Cycle

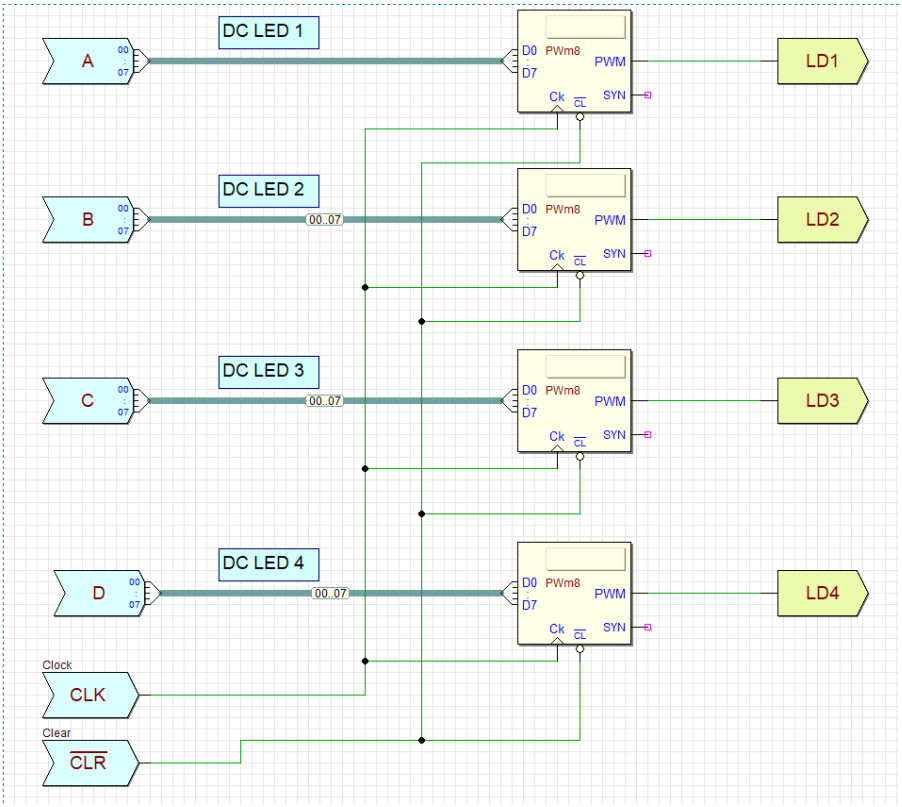

Figura 5: Sistema de funcionamento do bloco PWM

3.2 Decodificador

O decodificador é a parte do circuito lógico que fará a aquisição, registro e validação dos dados enviados pelo Arduino, sendo uma parte crucial para o funcionamento do sistema. Nosso decodificador é composto por uma máquina de estados simples e um circuito lógico.

O circuito lógico completo, mostrado na Figura 6, é composto por quatro módulos:

- Máquina de Estados
- Registrador Série-Paralelo
- Contador
- Baud-rate generator

O módulo da Máquina de Estados, mostrado detalhadamente na Figura 7, possui basicamente dois estados. O primeiro, chamado w(wait), é responsável por aguardar por novas mensagens do Arduíno enquanto confirma que os LEDs podem ser ligados por meio da saída *RDY*.

O estado dois, chamado de fl(first load), é ativo quando uma mensagem nova começa a ser enviada para a FPGA, sendo responsável por atrasar em 52 clocks para começar a realizar o processo de armazenamento da mensagem, de forma que cada bit recebido seja armazenado aproximadamente na metade de seu período, garantindo que o bit recebido esteja correto.

Após isso, no estado três, dc(delay to count), o contador de 104 é utilizado para realizar efetivamente a sincronização do envio dos bits na velocidade de 9600 bits/segundo com o clock de 1MHz da DE2. A cada 104 clocks é realizado o armazenamento de um bit, de forma a atrasar a velocidade de armazenamento do circuito o suficiente para sincronizar com a velocidade de envio do Arduíno.

Além disso, o módulo do Contador é responsável por realizar a contagem de bits da mensagem recebida, mantendo o módulo do Registrador Série-Paralelo ativo durante toda a recepção da mensagem por meio do gerenciamento do estado lb(load bits), que é o responsável por acionar o registrador e o contador no tempo certo para a recepção da mensagem completa.

Posteriormente, a máquina de estados deve aguardar um novo ciclo de 104 clocks por meio do estado sd(stop bit delay), sendo que ao final desse último ciclo, chegaremos no estado ss(stop bit set). Este último estado é responsável por registrar o bit de fim de mensagem, que é utilizado para verificar se a mensagem foi recebida corretamente, onde caso a mensagem seja recebida incorretamente, passaremos a ignorá-la por meio da saída "MSG".

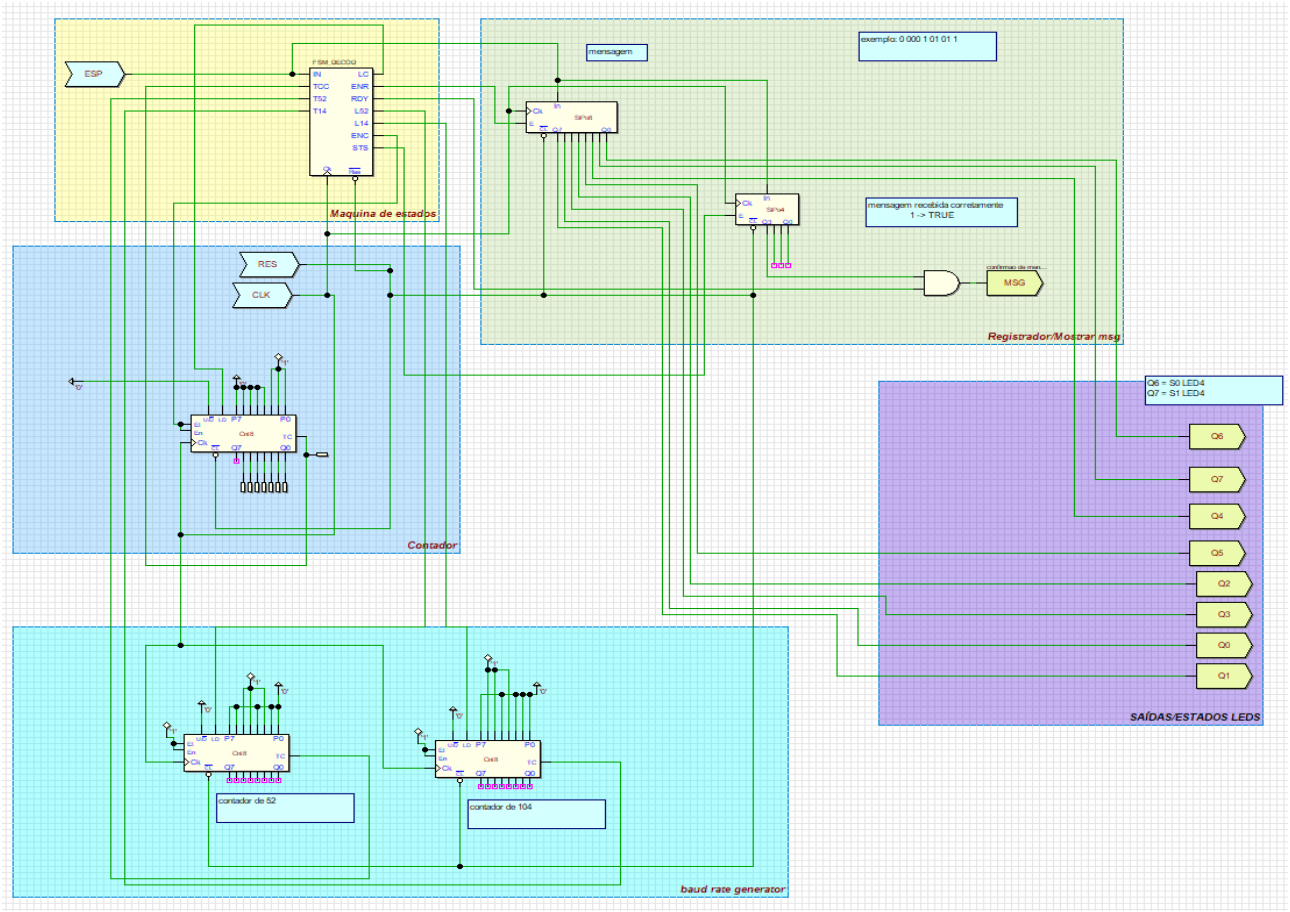

Figura 6: Versão final do decodificador

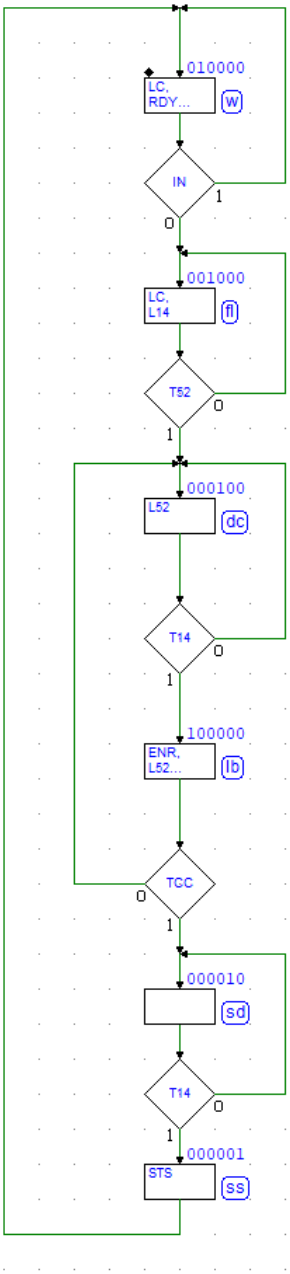

Figura 7: Versão final da máquina de estados do decodificador

3.3 Página Web

O site foi construído usando Node.js, React, Graphql e TypeScript, que são linguagens modernas e populares para o desenvolvimento de soluções Web.

O Front-end é responsável pela interação com o usuário e foi desenvolvido em React, uma biblioteca de JavaScript que permite a criação de interfaces de usuário de forma dinâmica e reativa, utilizando uma sintaxe semelhante a HTML. Na interface do usuário, este pode interagir com uma ilustração da placa DE2 e selecionar os LEDs desejados, que são representados por pequenas caixas de seleção em tons de cinza.

Quando o usuário clica em uma caixa de seleção, a cor da caixa muda gradualmente de cinza para um amarelo intenso, indicando a luminosidade desejada para o LED correspondente na placa física.

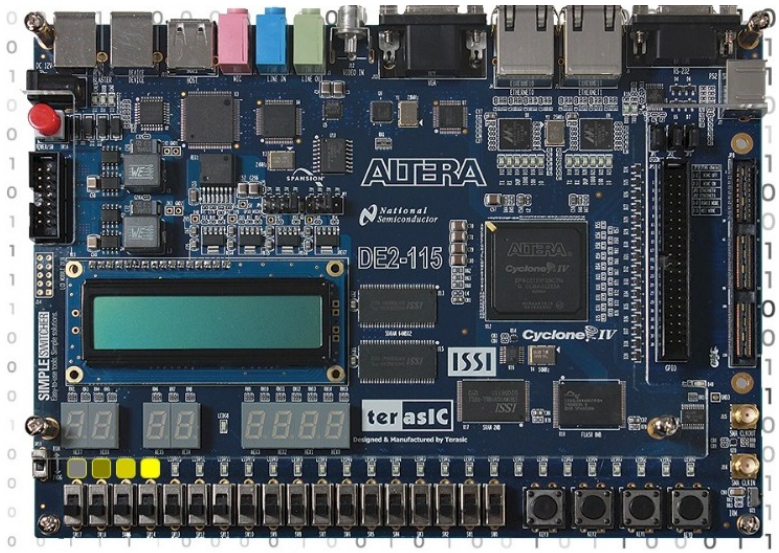

Figura 8: Ilustração da Página Web

O Backend, responsável pela comunicação com o banco de dados e pelo processamento das solicitações do usuário, foi desenvolvido em TypeScript e utiliza o Graphql como linguagem de comunicação entre o banco de dados e a aplicação.

As informações de estado dos LEDs são armazenadas em um banco de dados MongoDB, onde cada cliente representa um LED e contém informações como o ID e o estado atual. Quando o usuário interage com a interface do usuário e altera o estado dos LEDs, o Backend é responsável por atualizar o banco de dados com as informações mais recentes.

O site é programado para atualizar continuamente seus dados em referência ao banco de dados, representado na Figura 9, para garantir a sincronização entre o estado dos LEDs apresentado na interface do usuário e o estado atual dos LEDs na placa física, já que a placa DE2 modifica o estado dos LEDs no banco de dados.

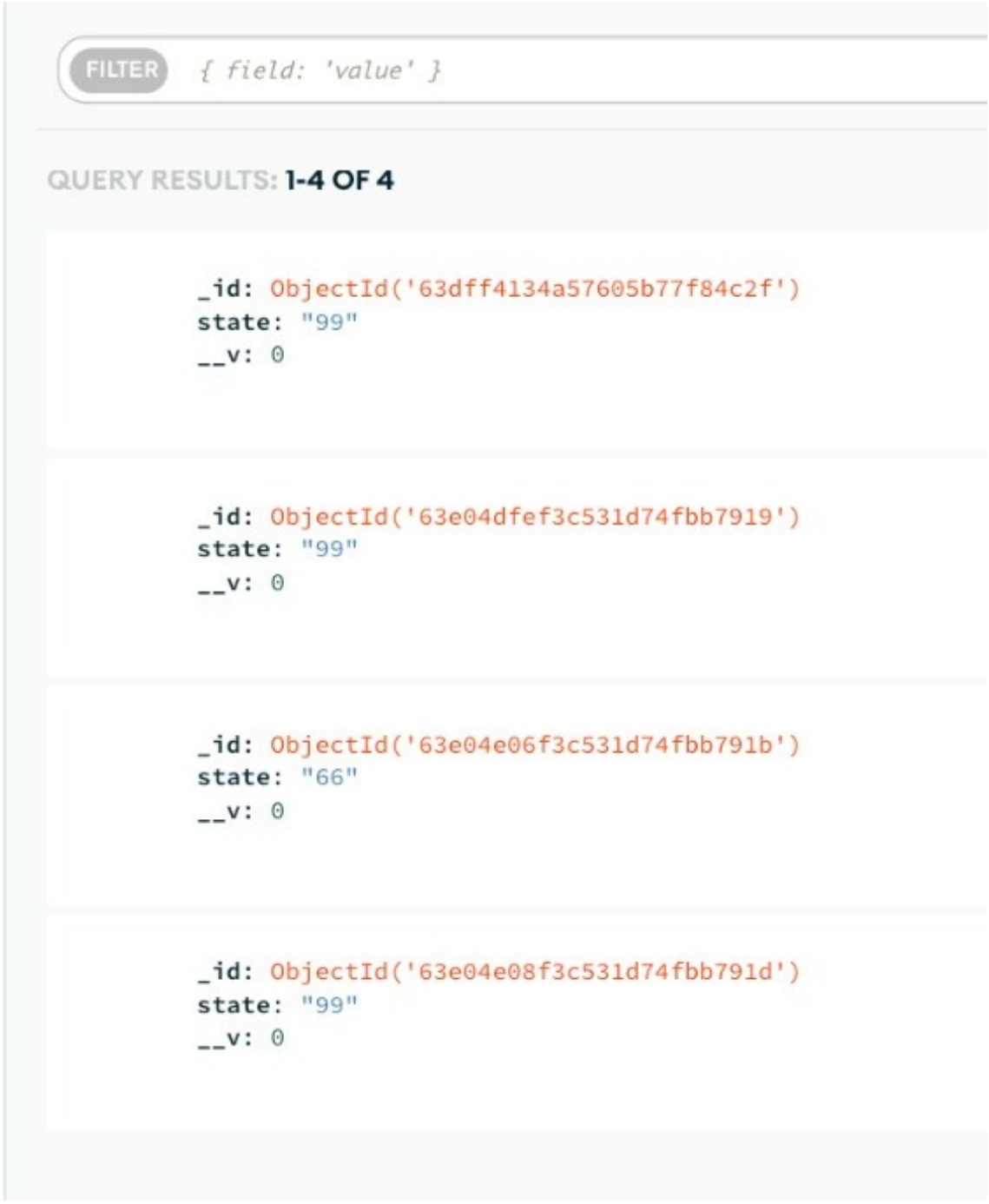

Figura 9: Banco de Dados

Para se comunicar com o Arduíno, o intermédio entre o site e a Altera DE2, utiliza-se uma API feita em Express, uma biblioteca de Node.js para a criação de servidores HTTP. Essa API deve estar instanciada em um computador que esteja ligado com o Arduíno. Através dessa API, o site recebe e manda o estado dos LEDs selecionados pelo usuário na interface do usuário. Esses dados são então comunicados para o Arduíno, por comunicação serial com o computador, que por sua vez, envia o sinal para a placa DE2, por outra porta serial do Arduíno, que altera o estado dos LEDs fisicamente. Assim, o usuário pode interagir com a placa DE2 de forma remota e controlar seus LEDs através do site, utilizando uma interface amigável e intuitiva.

Para garantir que a comunicação entre o site e a placa DE2 seja feita de forma sincronizada, o site utiliza a técnica de *polling*, onde o site faz requisições periódicas para a API e verifica as mudanças no estado dos LEDs. Dessa forma, o site sempre tem acesso ao estado atual dos LEDs na placa física e pode atualizar a interface do usuário com as informações mais recentes.

3.4 Chaves da Placa Altera DE2

Visto que o usuário terá a possibilidade de escolher a luminosidade de cada LED através das chaves da Placa DE2, o circuito possui o bloco *Registradores das Chaves*, o qual utiliza quatro registradores para guardar os estados de cada LED controlados pelas chaves predefinidas, Figura 10.

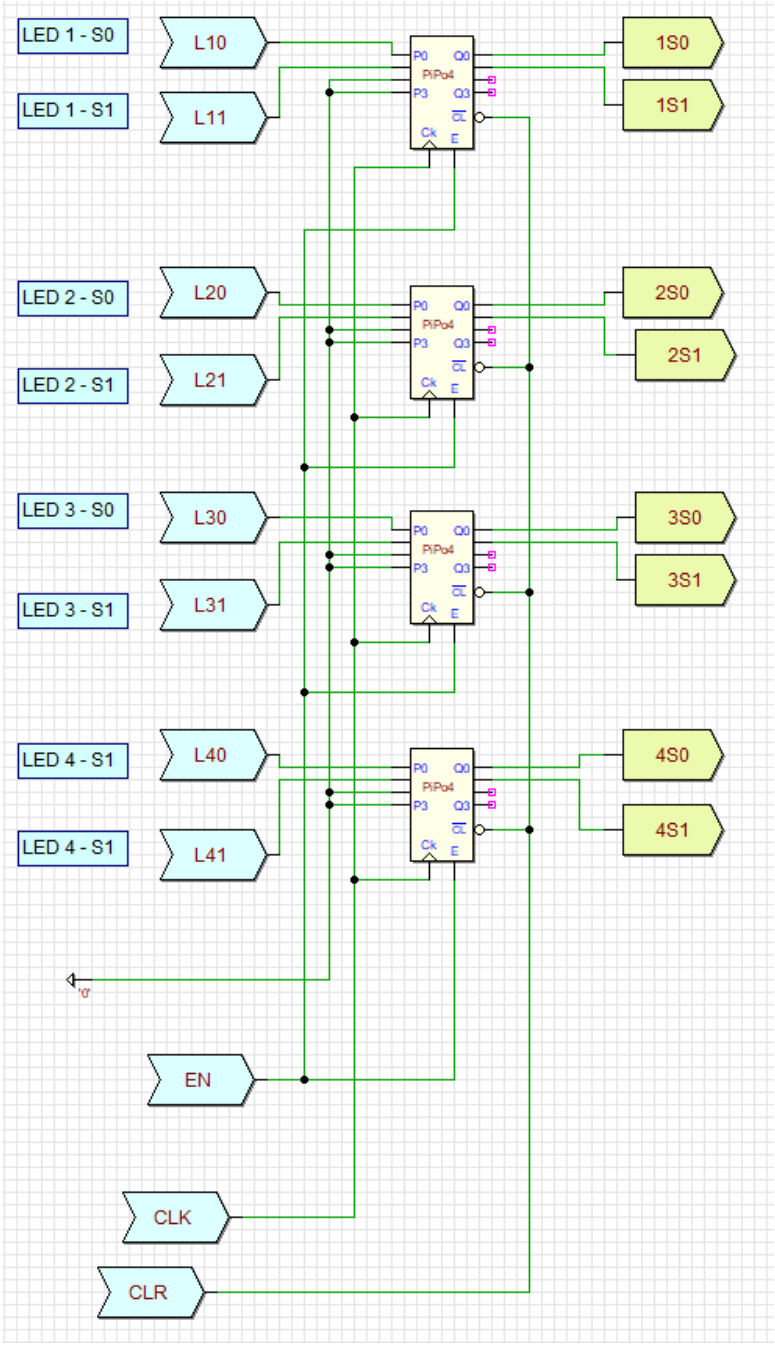

Figura 10: Bloco Registradores das Chaves

3.5 Codificador

O Codificador é o circuito responsável por realizar o envio dos dados da DE2 para o Arduíno, como mostrado na Figura 11, ele é composto por:

- Contador de 10
- Registrador Paralelo-Série
- Contador 104
- Máquina de estados
- Delay

O funcionamento do circuito, a partir de sua máquina de estados mostrada na Figura 12, se dá primeiramente pelo estado *st*(set), onde são carregados os contadores e o registrador.

O estado *wt*(wait) é o estado utilizado para realizar o delay de 104 clocks entre o envio de cada bit, de forma a sincronizar o clock da DE2 com a velocidade de recepção de 9600 bits/segundo do Arduíno.

No estado *ct*(count ten), fazemos a contagem dos bits enviados, que, incluso o *start-bit* e o *stop-bit*, contabiliza um total de 10 bits. A cada bit enviado é somado 1 ao contador de 10, de forma que, quando totalizamos 10 bits, a máquina de estados fica transitando entre os estados *dc*(delay to count) e *tr*(timer) até que se passem 255^2 clocks. Após isso, a máquina de estados é resetada e começa a enviar a mensagem novamente.

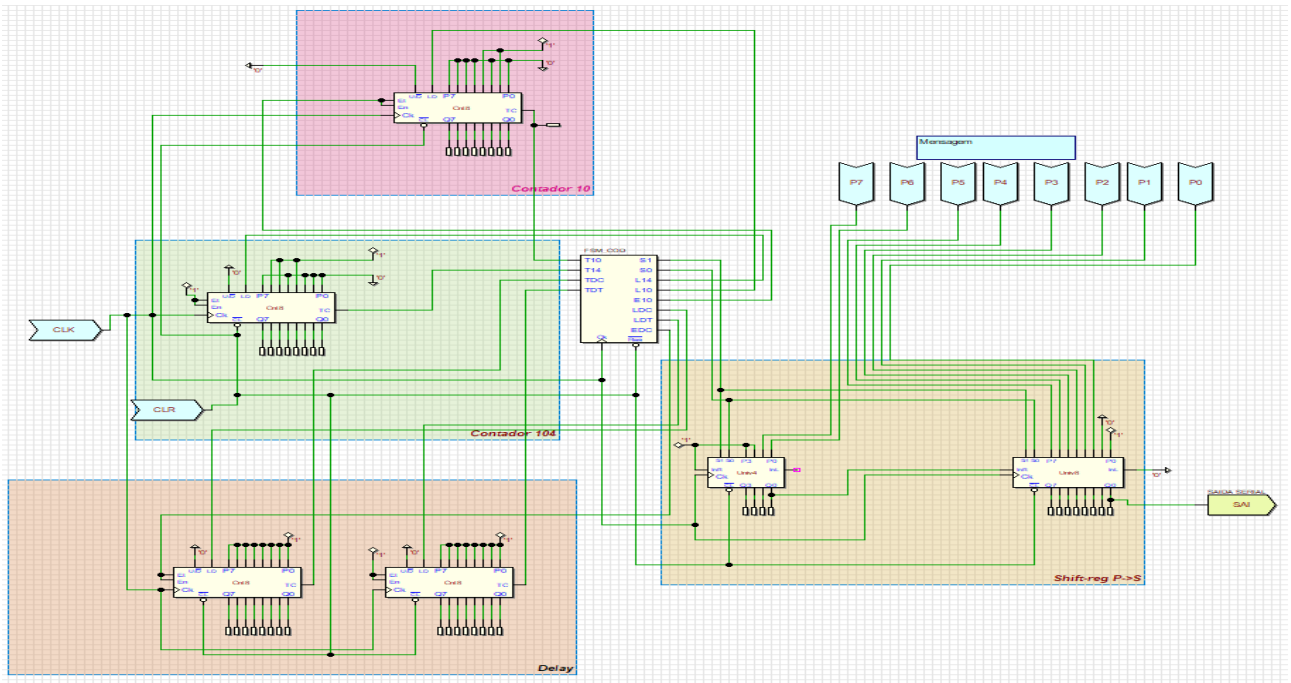

Figura 11: Codificador

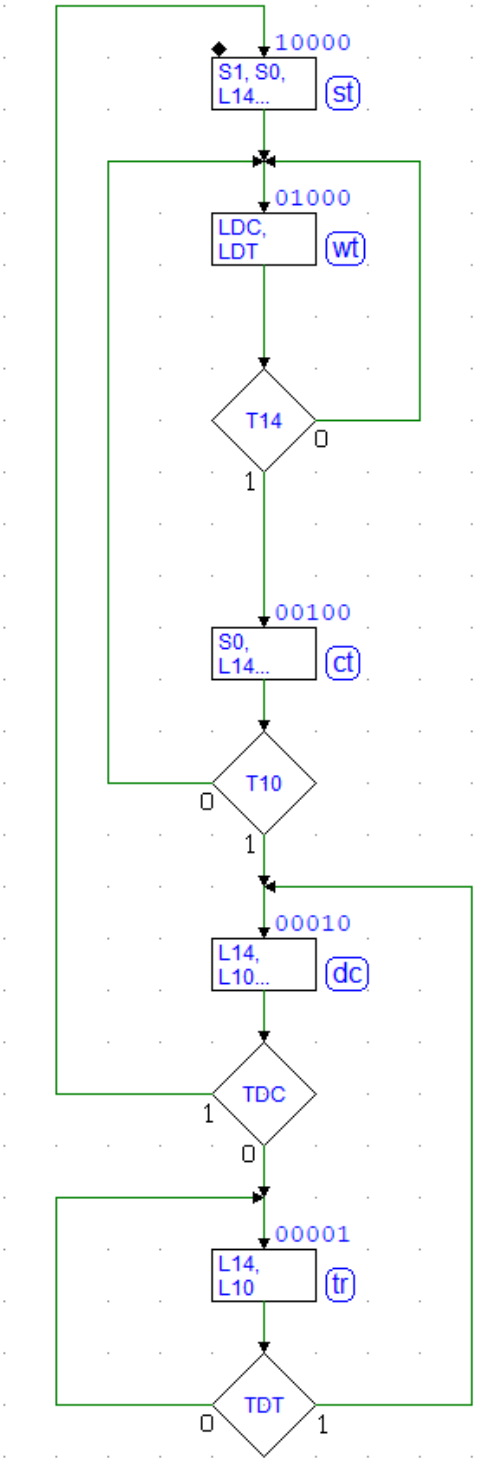

Figura 12: Máquina de Estados do Codificador

3.6 Circuito Principal

Na Figura 13, onde se encontra o circuito principal (junção de todos os blocos), há uma separação por cores para uma melhor identificação:

- Roxo: Decodificador;
- Cinza: Registradores dos estados dos LEDs nas Chaves da Placa DE2;
- Laranja: Chaves da Placa DE2 e *Enable*;
- Azul: Bloco de conexão entre o Decodificador e o Registrador das Chaves;
- Amarelo: Bloco *Duty Cycle* para cada LED;
- Verde: Bloco *PWM*;
- Rosa: Codificador;

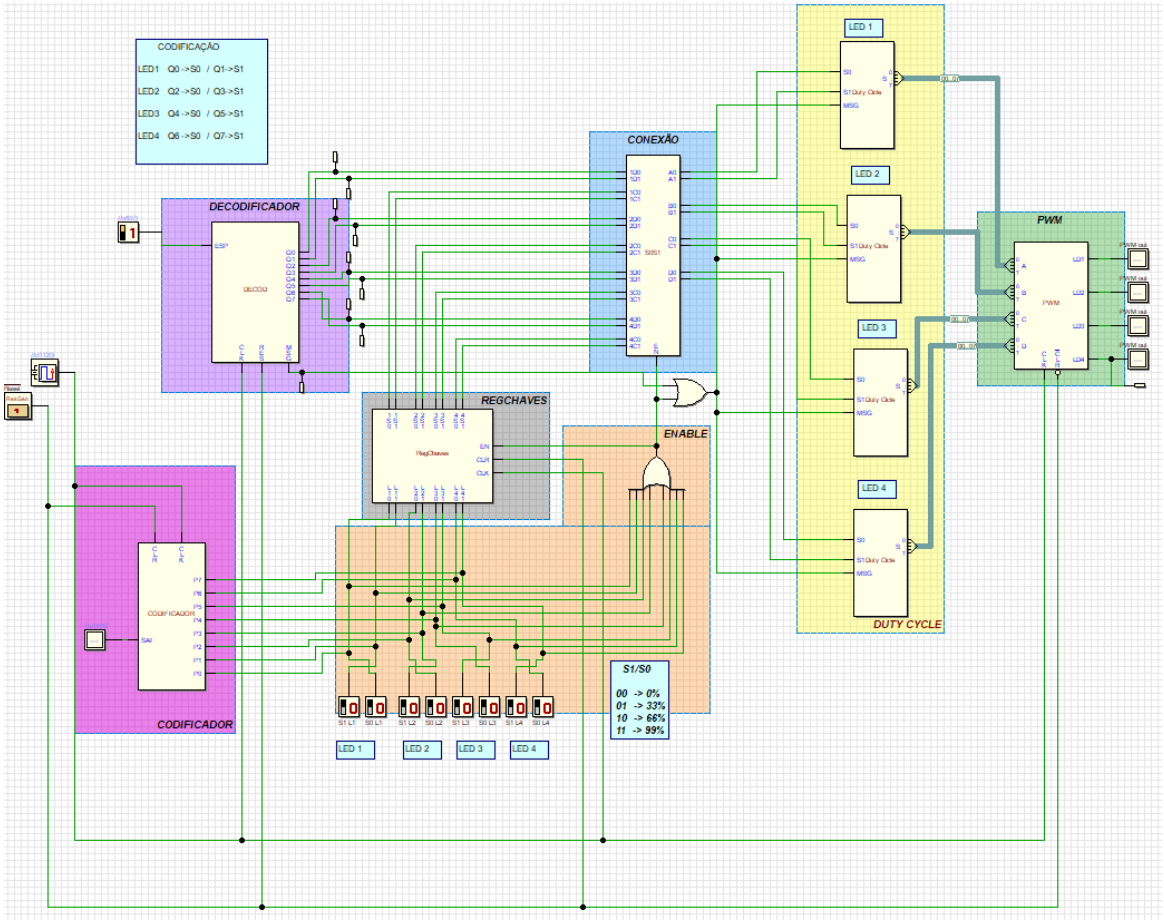

Figura 13: Programa Principal

3.6.1 Bloco S1S0 - Conexão de controle entre Decodificador e Registrador das Chaves

A função desse bloco do circuito é selecionar uma das duas opções disponíveis: o código proveniente do Arduíno ou o código proveniente das chaves da Placa DE2. Quando as chaves da placa DE2 estão zeradas, os multiplexadores dentro do bloco selecionam a opção *I0* (bits originários do Arduíno), Figura 14. Caso as chaves não estejam mais zeradas, os multiplexadores selecionam a opção *I1* (bits passados pelas chaves da Placa DE2), Figura 15.

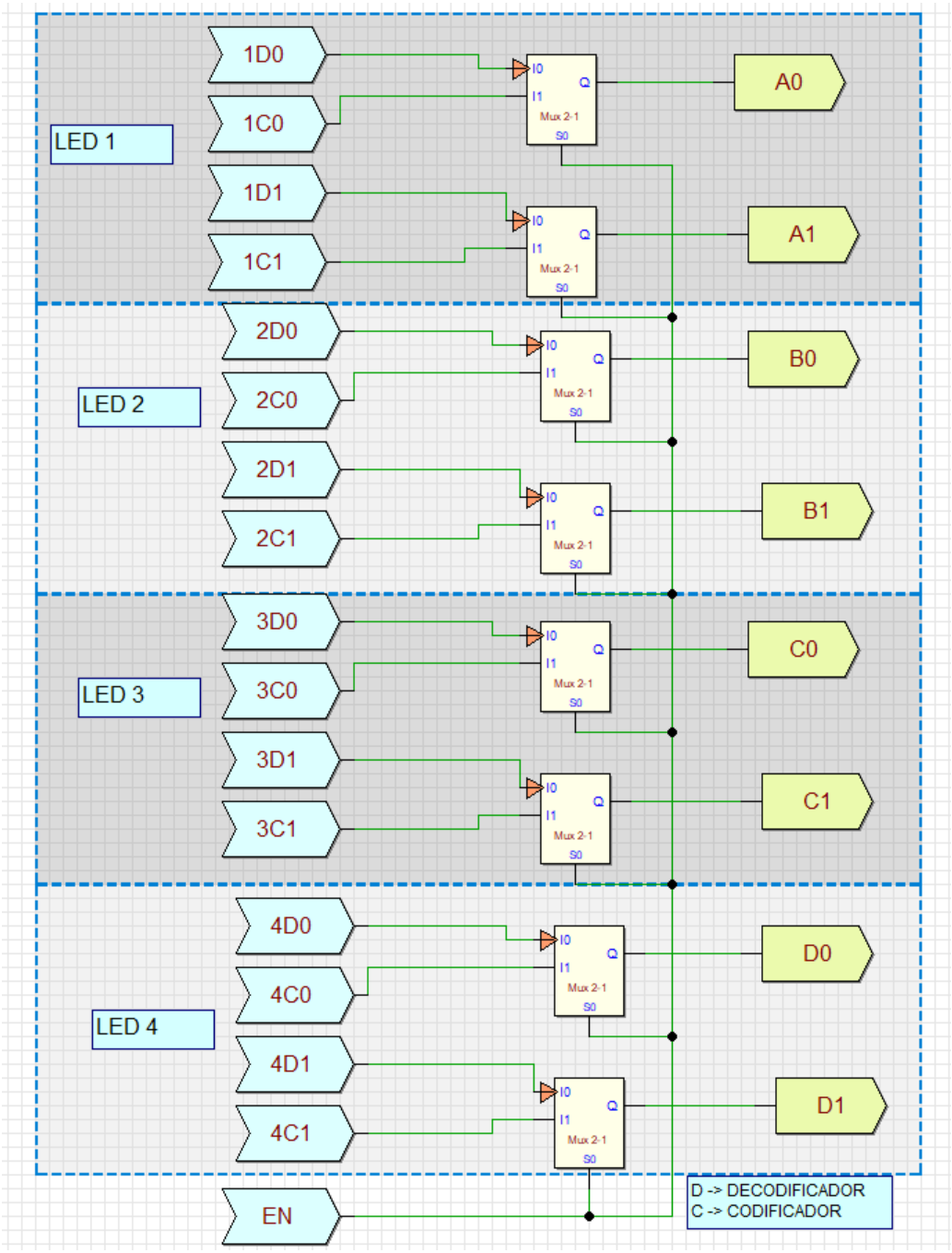

Figura 14: Bloco S1S0, quando o código é originário do Arduíno

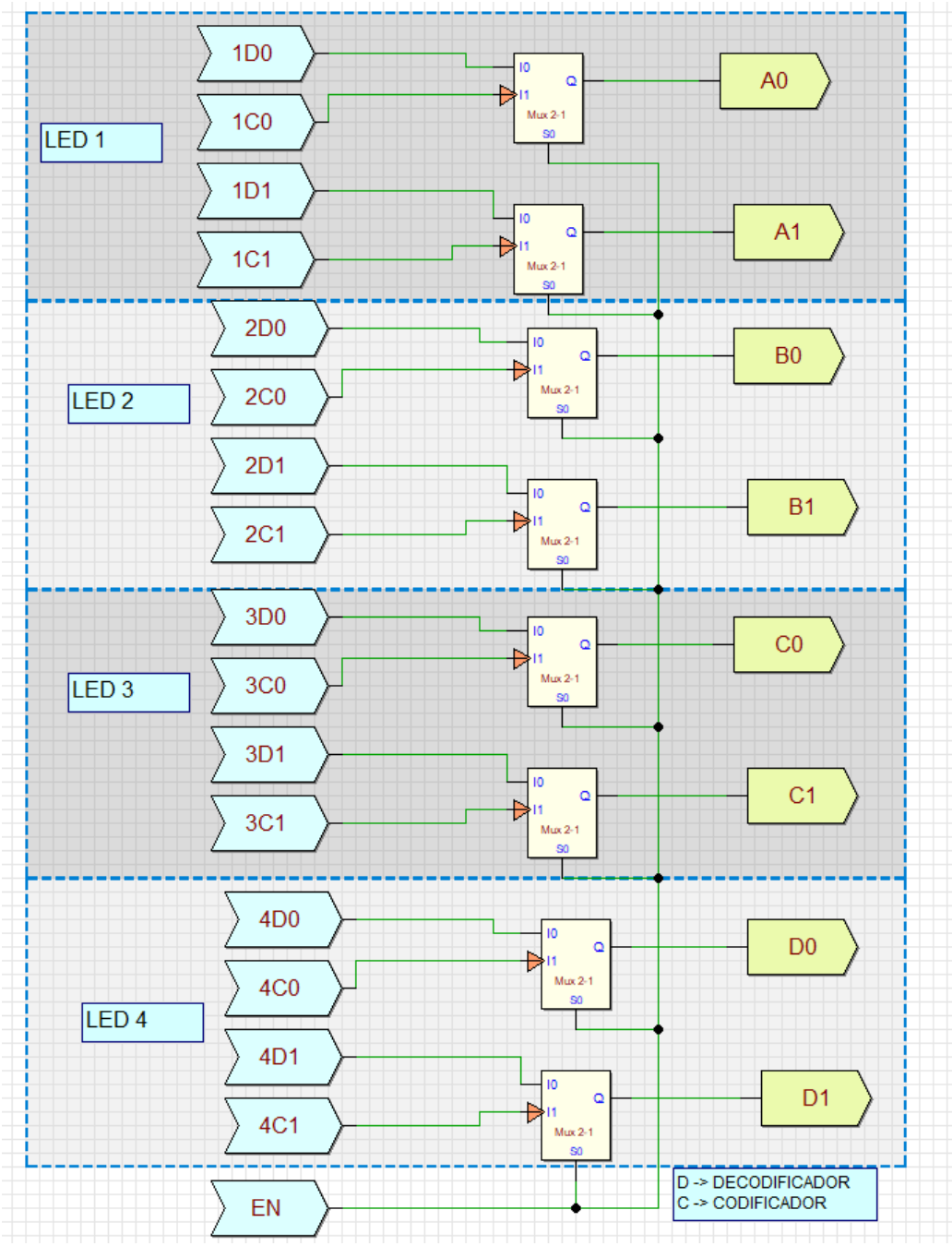

Figura 15: Bloco S1S0, quando o código é originário da Placa DE2

4 Conclusão

O projeto apresentou a proposta de realizar a comunicação entre um agente e uma FPGA por meio do microcontrolador Arduíno, utilizando conexão Wi-Fi para o aprendizado do grupo. O objetivo principal foi permitir o controle remoto da FPGA por meio de uma página web, possibilitando o controle de quatro LEDs da FPGA e seus níveis de luminosidade. Além disso, foi implementado o controle direto dos LEDs pela FPGA, de modo que as alterações no estado dos LEDs na FPGA sejam refletidas também na interface do usuário do site.

Para alcançar esses objetivos, foram desenvolvidas codificações para a comunicação serial entre o Arduíno e a FPGA, permitindo o envio e recebimento de informações sobre os níveis de luminosidade dos LEDs e o estado das chaves. Foi implementado ainda o uso do PWM para controlar a luminosidade dos LEDs.

O projeto apresentou uma abordagem interessante para a comunicação entre diferentes dispositivos, utilizando tecnologias modernas e de fácil acesso. Além disso, permitiu o aprendizado sobre a programação de microcontroladores e a interação com a FPGA.

Como possíveis melhorias para o projeto, poderiam ser implementadas novas funcionalidades na página web, como a adição de botões para controlar as chaves da FPGA, ou a implementação de um sistema de monitoramento de temperatura ou outras variáveis. Essas melhorias poderiam tornar o sistema ainda mais completo e funcional.

Em resumo, o projeto apresentou uma solução interessante para o controle remoto de dispositivos por meio da internet, com potencial para futuras aplicações em diferentes áreas, como a indústria, a automação residencial, entre outras.