UNIVERSIDADE DE SÃO PAULO

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO - POLI USP

PCS3645 - Laboratório Digital II

Gabriel Yugo Nascimento Kishida 11257647
Gustavo Azevedo Corrêa 11257693

Turma 1 - Bancada A4

NEKO NEKO NEE - SEMANA 2

Prof. Edson Midorikawa

SÃO PAULO - SP

2021

SUMÁRIO

1 INTRODUÇÃO 3
1.1 Neko Neko Nee: Desenvolvimento do controle do servo 3
1.2 Funcionalidade 4
1.3 Não Funcionalidade 4
2 SOLUÇÃO TÉCNICA 5
2.1 Descrição Geral 5
3 Projeto 6
3.1 Desenvolvimento do Circuito 6
3.1.1 Definição da máquina de estados 6
3.1.2 Implementação do fluxo de dados 6
3.1.3 Implementação da unidade de controle 7
3.2 Testagem em Simulação 8
3.2.1 Definição dos Testes 9
3.2.2 Código do Testbench 9
3.2.3 Resultado dos Testes 11
3.3 Aplicação da Placa 14
3.3.1 Modificações para a montagem 14
3.3.2 Pinagem do circuito 15
3.3.3 Testes em Bancada 15
4 RESULTADOS 20
4.1 Conclusão 20
4.2 O que já foi desenvolvido 20
4.3 O que resta ser desenvolvido. 20
5 REFERÊNCIAS BIBLIOGRÁFICAS 21

1 INTRODUÇÃO

1.1 Neko Neko Nee: Desenvolvimento do controle do servo

Figura 1: Exemplo de alimentador automático

O objetivo desse laboratório foi criar uma implementação da saída de comida do alimentador para que agisse de forma diferente em cada tamanho de ração definida. Para a realização será projetado um controle de servomotor com os tamanhos das porções como entradas e também o momento de abertura.

1.2 Funcionalidade

O circuito deve somente sair do estado de espera quando detectado a proximidade de um animal

1.3 Não Funcionalidade

O circuito não deve mapear num espectro contínuo de posições, nem criar algum tipo de visualização dos arredores com as medidas (no entanto, isso seria possível reunindo os dados enviados ao computador por meio da comunicação serial).

2 SOLUÇÃO TÉCNICA

2.1 Descrição Geral

Neste experimento da disciplina de Laboratório Digital II, será desenvolvido o projeto da interface de um sensor de distância por ondas ultrassônicas. As especificações fornecidas pelo docente da disciplina e as instruções da apostila serão utilizadas como guia de projeto.

Para um desenvolvimento completo e organizado, o projeto deve ser acompanhado dos seguintes cuidados:

Aferir o circuito fornecido pelo docente em *.vhd*, estudando o funcionamento do circuito e possíveis melhorias a serem aplicada na lógica do sistema. Isto também inclui a documentação do funcionamento no relatório, de forma extensa e compreensiva para auxiliar os estudantes e os leitores a melhor entenderem o desenvolvimento do projeto.

Verificar se o circuito desenvolvido foi aplicado de maneira correta, analisando suas entradas e saídas no arquivo *.vhd*, procurando erros no código e verificando se o mesmo compila no *Quartus Prime*.

Simular o circuito desenvolvido para diversas entradas, estudando as saídas obtidas no *Modelsim* e analisando se os resultados obtidos são condizentes com o que era esperado.

Aplicar o circuito desenvolvido na placa FPGA, testando-o em conjuntura com o suporte à tecnologia *IoT* (Internet of Things) fornecida pelo Laboratório Digital, utilizando o aplicativo **MQTT Dash**. Além disso, utilizará-se os sinais de depuração na placa FPGA para testar o circuito ao vivo.

3 Projeto

3.1 Desenvolvimento do Circuito

3.1.1 Definição da máquina de estados

Foi idealizado uma máquina de estados simples que deixasse o trabalho de diferentes comportamentos de cada tamanho de porção para o fluxo de dados, definindo um comportamento geral de abertura e enceramento da distribuição de comida. O diagrama que descreve esse comportamento pode ser visualizado a seguir:

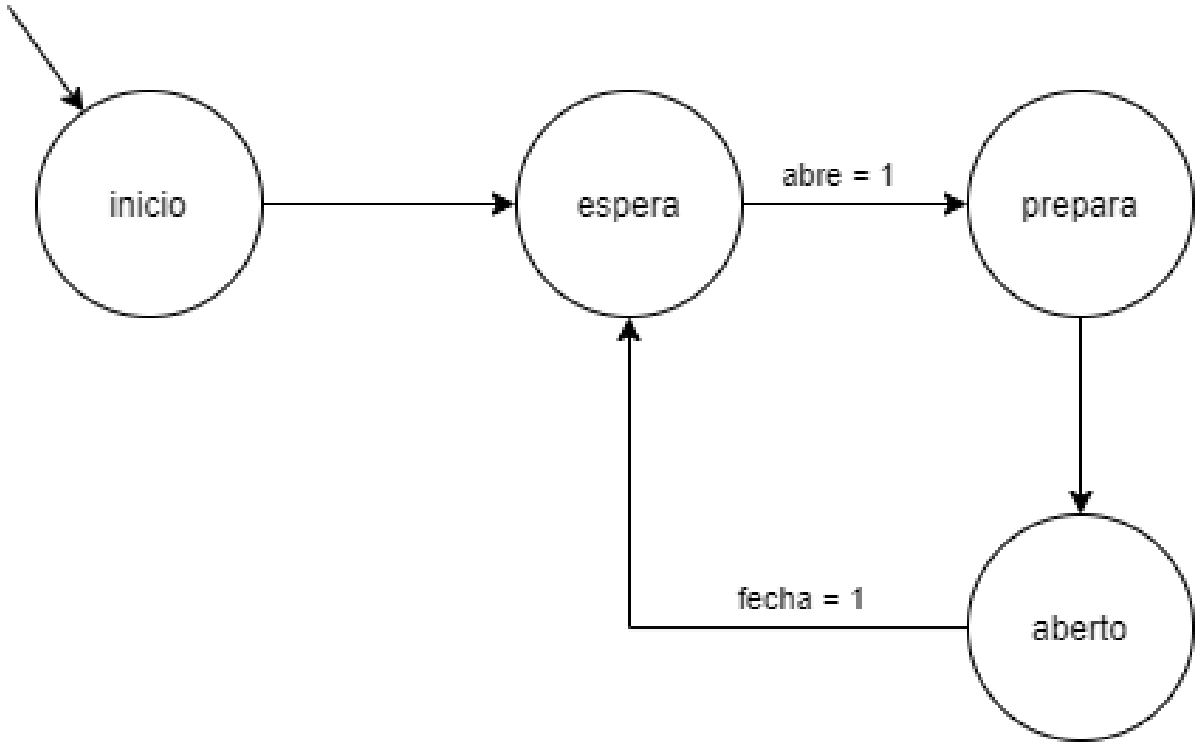

Figura 2: Diagrama de estados do sistema de alimentação

3.1.2 Implementação do fluxo de dados

Foram criados 4 contadores com diferentes valores de contagem máxima afim de emular os diferentes períodos de tempo disponíveis para cada tamanho de porção disponível, esses seriam ligados a um multiplexador que definiria qual das saídas *fim* dos difrentes contadores seria escolhida. Para que não houvessem problemas de mudança da entrada

de seleção no meio do funcionamento, acarretando diversos problemas e questões a serem tratadas, foi utilizado um registrador que guardava a entrada selecionada no período de preparação. Pode-se observar o diagrama de blocos do circuito descrito a seguir:

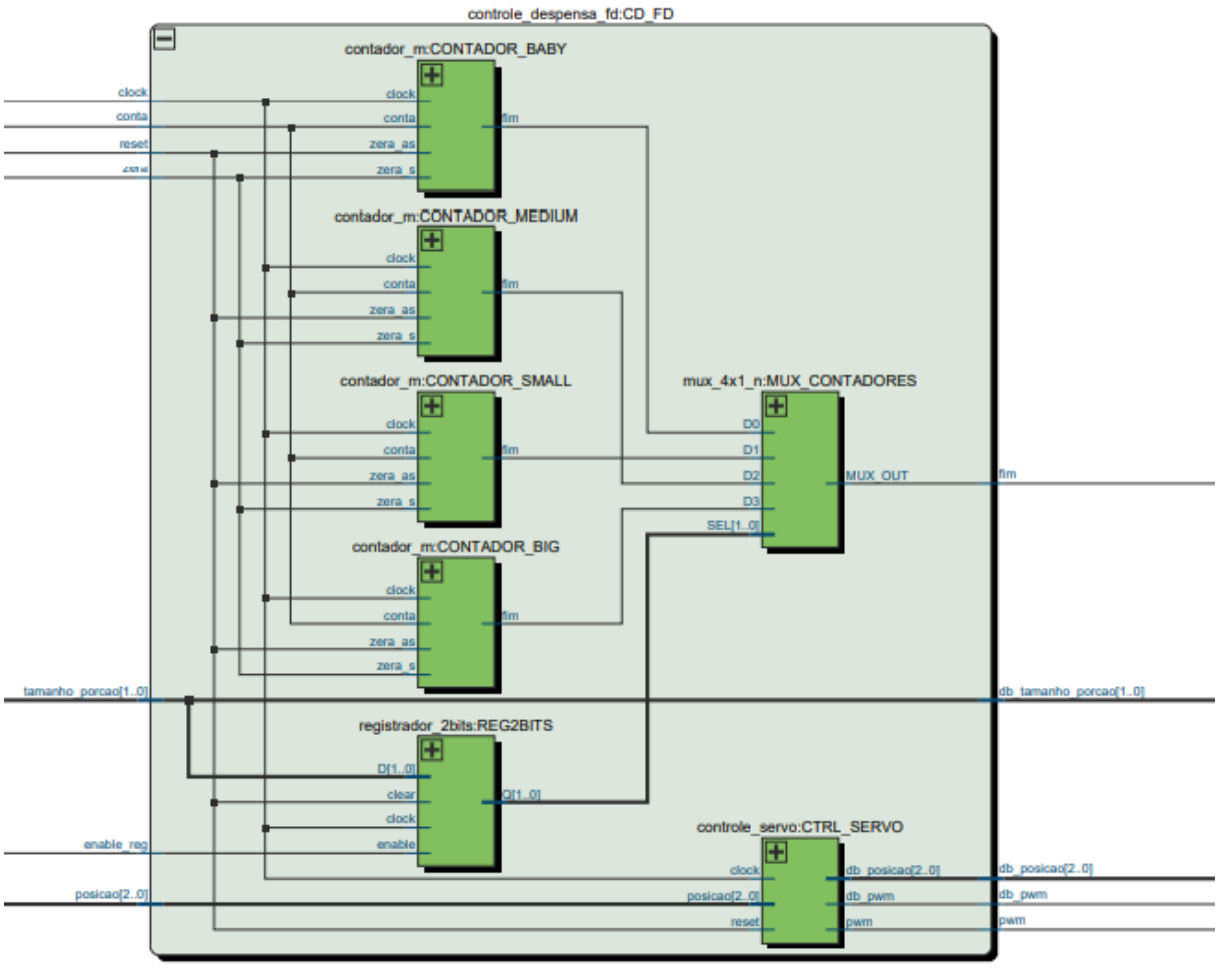

Figura 3: Fluxo de dados do controle do servo

3.1.3 Implementação da unidade de controle

Para a unidade de controle foram considerados os estados definidos na figura 2. O comportamento do circuito em cada estado é descrito da seguinte forma:

- Inicio

  Estado de entrada, é o estado depois do reset.

- Espera

  Estado que aguarda a proximidade do animal, não há nenhuma mudança de sinal. Em caso de ativação do sinal *abre* (quando há a proximidade) passa-se para o estado

prepara.

- Prepara

  Estado em que ocorre a preparação para a abertura, onde é amostrado e registrado o tamanho da porção e os contadores que definem o tempo de abertura são zerados.

- Aberto

  Estado de abertura do alimentador, o alimento é liberado com o servomotor na sua posição máxima mas por um tempo definido pelo tamanho da porção guardado no estado de preparação. Quando o contador escolhido chega ao fim é recebido o sinal *fecha* e passa-se para o estado de espera novamente.

Ao final foi adicionado um detector de borda junto a entrada *abre* da unidade de controle para evitar que o sinal fosse ativado continuamente por animal que ficasse parado na frente do circuito, causando outro diversos problemas no comportamento final. Ademais, foi adicionado um sinal de depuração *db_aberto* para que apurasse de forma mais precisa se o tempo determinado estava correto durante a realização do laboratório. A montagem completa pode ser visualizada a seguir:

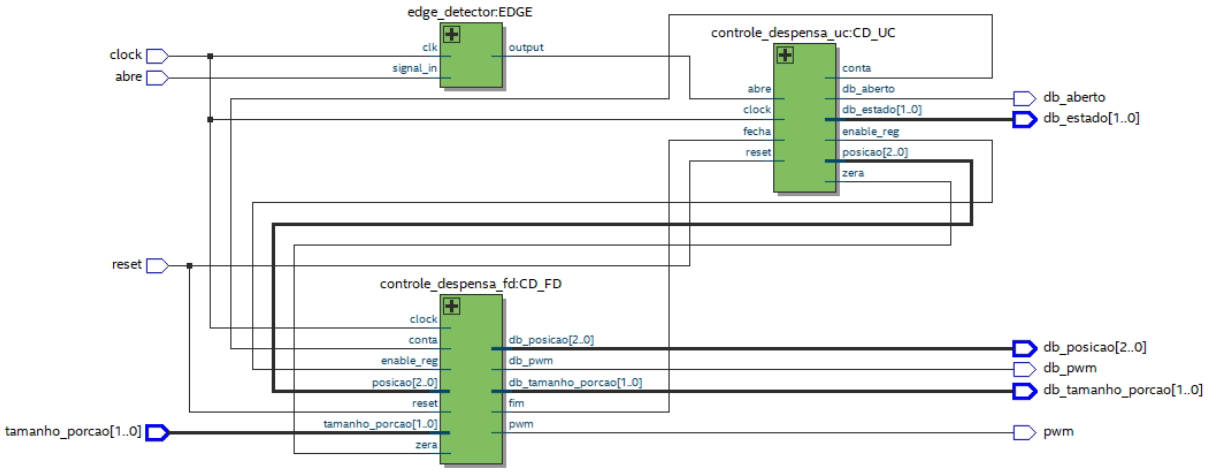

Figura 4: Fluxo de dados do controle do servo

3.2 Testagem em Simulação

Para a testagem do circuito, foi desenvolvido um *testbench* que consiste em avaliar o funcionamento ideal do módulo. Assim, a testagem deve avaliar:

- O tamanho da onda PWM para as duas situações possíveis (aberto e fechado);
- O tempo em que o servomotor fica em aberto para cada tamanho de porção;

3.2.1 Definição dos Testes

Para cobrir todas as situações determinadas, o teste desenvolvido consiste das seguintes etapas:

- Inserir input **"tamanho_porcao"** com "00"e um pulso de **"abre"**;
- Inserir input **"tamanho_porcao"** com "01"e um pulso de **"abre"**;
- Inserir input **"tamanho_porcao"** com "10"e um pulso de **"abre"**;
- Inserir input **"tamanho_porcao"** com "11"e um pulso de **"abre"**;

NOTA: entre cada inserção de input, deve haver uma espera para que o movimento do servomotor seja finalizado, e que a máquina de estados da unidade de controle retorne ao estado de **"espera"**.

3.2.2 Código do Testbench

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3
4  ENTITY controle_despensa_tb IS
5  END ENTITY;
6
7  ARCHITECTURE tb OF controle_despensa_tb IS
8
9      COMPONENT controle_despensa IS
10         PORT (
11             clock : IN STD_LOGIC;
12             reset : IN STD_LOGIC;
13             abre : IN STD_LOGIC;
14             tamanho_porcao : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
15             pwm : OUT STD_LOGIC;
16             db_pwm : OUT STD_LOGIC;
17         db_aberto: OUT STD_LOGIC;
18             db_posicao : OUT STD_LOGIC_VECTOR(2 DOWNTO 0);
19             db_tamanho_porcao : OUT STD_LOGIC_VECTOR(1 DOWNTO 0);
20             db_estado : OUT STD_LOGIC_VECTOR (1 DOWNTO 0)
21         );
22     END COMPONENT;
23
24     SIGNAL clock_in : STD_LOGIC := '0';
25     SIGNAL reset_in : STD_LOGIC := '0';
26     SIGNAL abre_in : STD_LOGIC := '0';
27     SIGNAL tamanho_porcao_in : STD_LOGIC_VECTOR (1 DOWNTO 0) := "00";
28     SIGNAL pwm_out : STD_LOGIC;
```

```
29    SIGNAL db_aberto_out : STD_LOGIC;
30     SIGNAL db_pwm_out : STD_LOGIC;
31     SIGNAL db_posicao_out : STD_LOGIC_VECTOR(2 DOWNTO 0);
32     SIGNAL db_tamanho_porcao_out : STD_LOGIC_VECTOR (1 DOWNTO 0) := "00";
33     SIGNAL db_estado_out : STD_LOGIC_VECTOR (1 DOWNTO 0) := "00";
34
35     SIGNAL keep_simulating : STD_LOGIC := '0';
36     CONSTANT clockPeriod : TIME := 20 ns;
37
38 BEGIN
39
40     clock_in <= (NOT clock_in) AND keep_simulating AFTER clockPeriod/2;
41
42     dut : controle_despensa PORT MAP(
43         clock => clock_in,
44         reset => reset_in,
45         abre => abre_in,
46         tamanho_porcao => tamanho_porcao_in,
47         pwm => pwm_out,
48         db_pwm => db_pwm_out,
49       db_aberto => db_aberto_out,
50         db_posicao => db_posicao_out,
51         db_tamanho_porcao => db_tamanho_porcao_out,
52         db_estado => db_estado_out
53     );
54
55     stimulus : PROCESS IS
56     BEGIN
57
58         ASSERT false REPORT "Inicio da simulacao" & LF & "... Simulacao ate 800 ms. Aguarde o final da
       simulacao..." SEVERITY note;
59         keep_simulating <= '1';
60
61         ---- inicio: reset ----------------
62         reset_in <= '1';
63         WAIT FOR 2 * clockPeriod;
64         reset_in <= '0';
65         WAIT FOR 10 * clockPeriod;
66
67         tamanho_porcao_in <= "00";
68         WAIT FOR 100 ms;
69
70         abre_in <= '1';
71         WAIT FOR 5 * clockPeriod;
72
73         abre_in <= '0';
74         WAIT FOR 101 ms;
75
76         tamanho_porcao_in <= "01";
77         WAIT FOR 100 ms;
78
79         abre_in <= '1';
80         WAIT FOR 5 * clockPeriod;
81
82         abre_in <= '0';
83         WAIT FOR 151 ms;
84
85         tamanho_porcao_in <= "10";
86         WAIT FOR 100 ms;
87
88         abre_in <= '1';
89         WAIT FOR 5 * clockPeriod;
90
91         abre_in <= '0';
92         WAIT FOR 201 ms;
93
```

```
94          tamanho_porcao_in <= "11";
95          WAIT FOR 100 ms;
96
97          abre_in <= '1';
98          WAIT FOR 5 * clockPeriod;
99
100         abre_in <= '0';
101         WAIT FOR 301 ms;
102
103         ASSERT false REPORT "Fim da simulacao" SEVERITY note;
104         keep_simulating <= '0';
105
106         WAIT;
107     END PROCESS;
108 END ARCHITECTURE;
```

3.2.3 Resultado dos Testes

O seguinte resultado foi obtido pela simulação dos testes no software ModelSim:

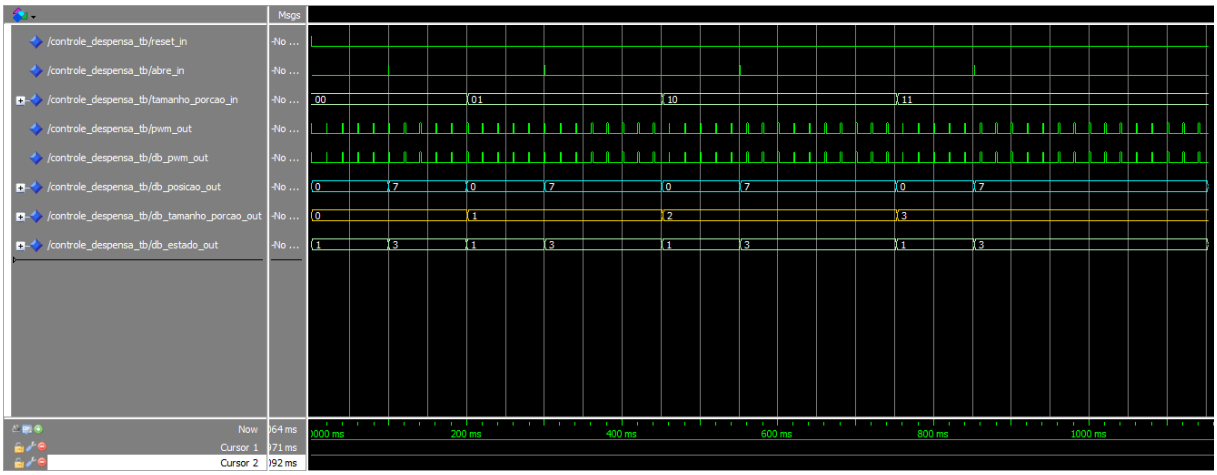

Figura 5: Forma de ondas gerais dos testes realizados

Para dar um melhor enfoque para cada critério a ser avaliado pelo teste, vamos dividi-los e estudá-los um a um.

Largura da PWM

Para o estado de ”espera”, ”prepara”e ”inicial”, a largura da PWM deve ser 1 milissegundo, indicando que o servomotor está na posição de fechado. A onda obtida foi:

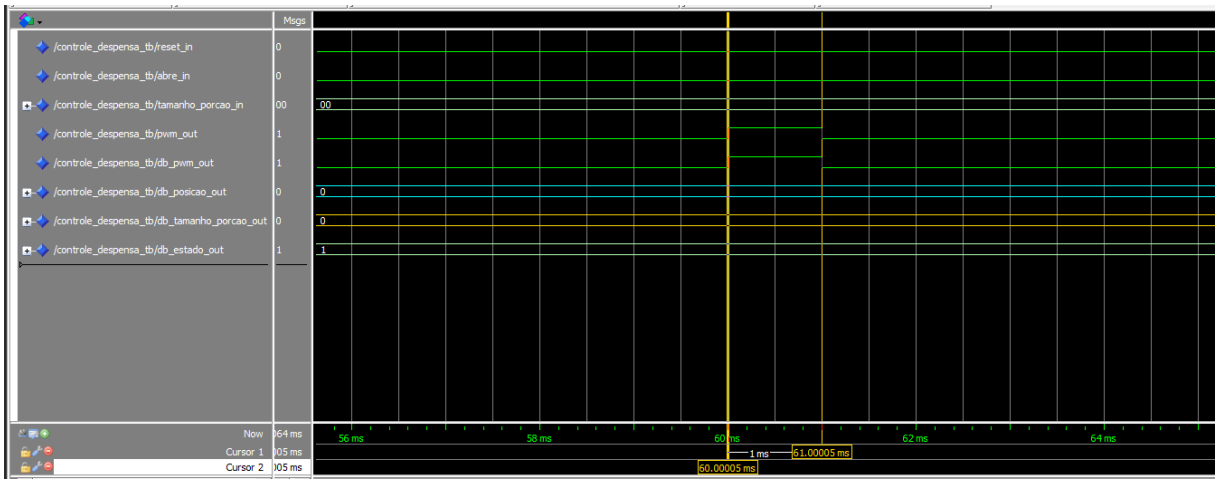

Figura 6: Largura da PWM quando o dispositivo está fechado

Como essa largura está de acordo com o esperado, o circuito passou neste critério.

Já para o estado de ”aberto”, a largura da PWM deve ser de 2 milissegundos, movimentando o servomotor na posição aberta. A onda obtida foi:

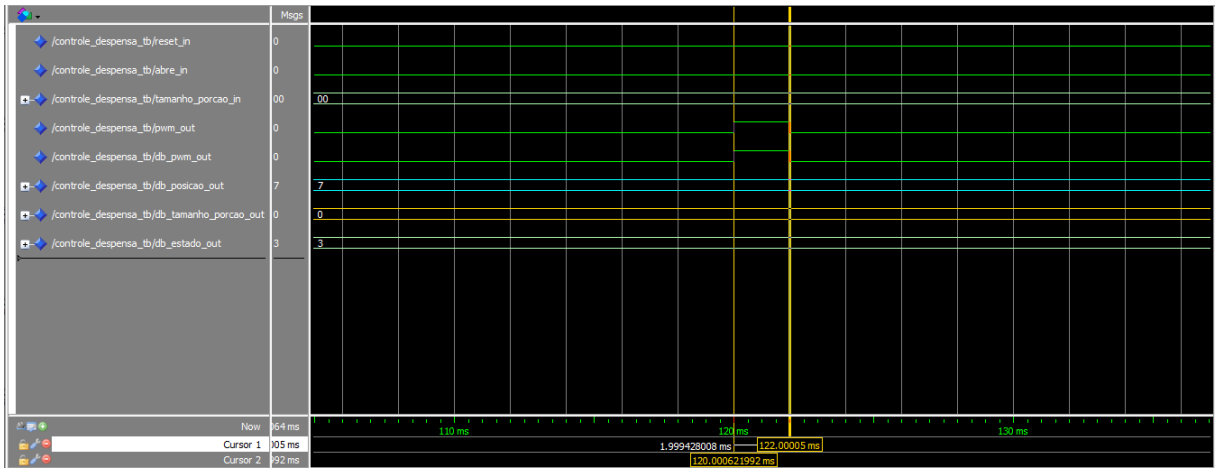

Figura 7: Largura da PWM quando o dispositivo está aberto

Como essa largura está de acordo com o esperado, o circuito passou neste critério.

Tempo em Aberto

Para viabilizar a testagem e reduzir o tempo de simulação, reduzimos os tempos em aberto em 10 vezes (para reduzir o esforço necessário pelo computador);

Para a configuração **”tamanho_porcao”** em ”00”, o dispositivo deve ficar aberto por 1 segundo (100 ms na simulação). O resultado obtido foi:

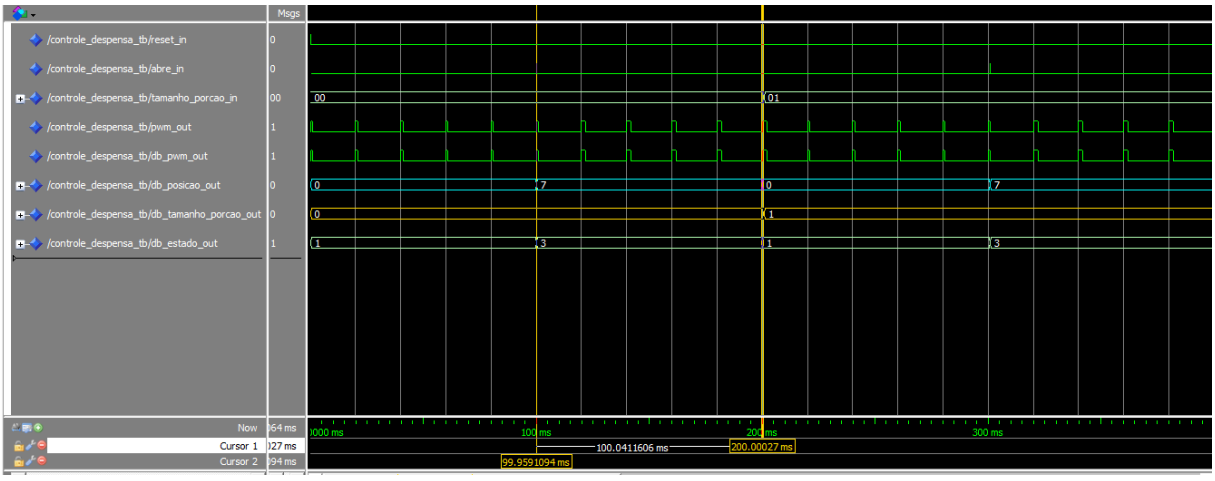

Figura 8: Tempo em aberto para ”tamanho_porcao”em ”00”

Para a configuração **”tamanho_porcao”** em ”01”, o dispositivo deve ficar aberto por 1,5 segundos (150 ms na simulação). O resultado obtido foi:

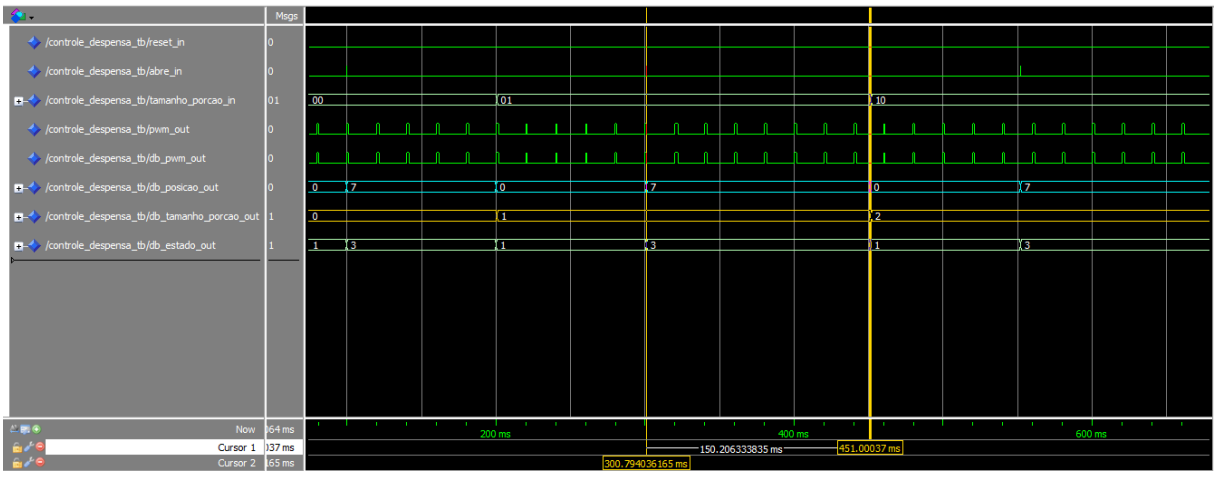

Figura 9: Tempo em aberto para ”tamanho_porcao”em ”01”

Para a configuração **”tamanho_porcao”** em ”10”, o dispositivo deve ficar aberto por 2 segundos (200 ms na simulação). O resultado obtido foi:

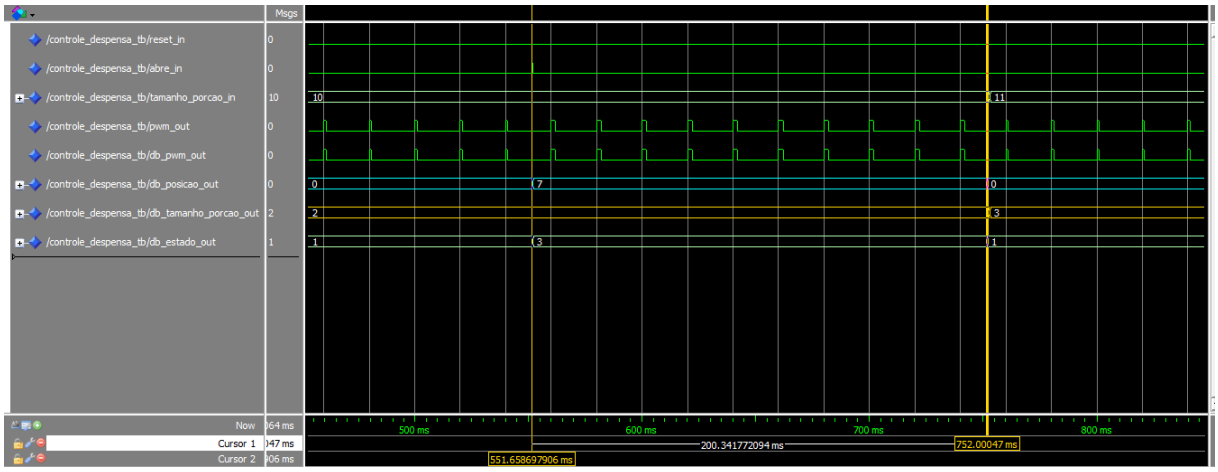

Figura 10: Tempo em aberto para ”tamanho_porcao”em ”10”

Para a configuração **”tamanho_porcao”** em ”11”, o dispositivo deve ficar aberto por 3 segundos (300 ms na simulação). O resultado obtido foi:

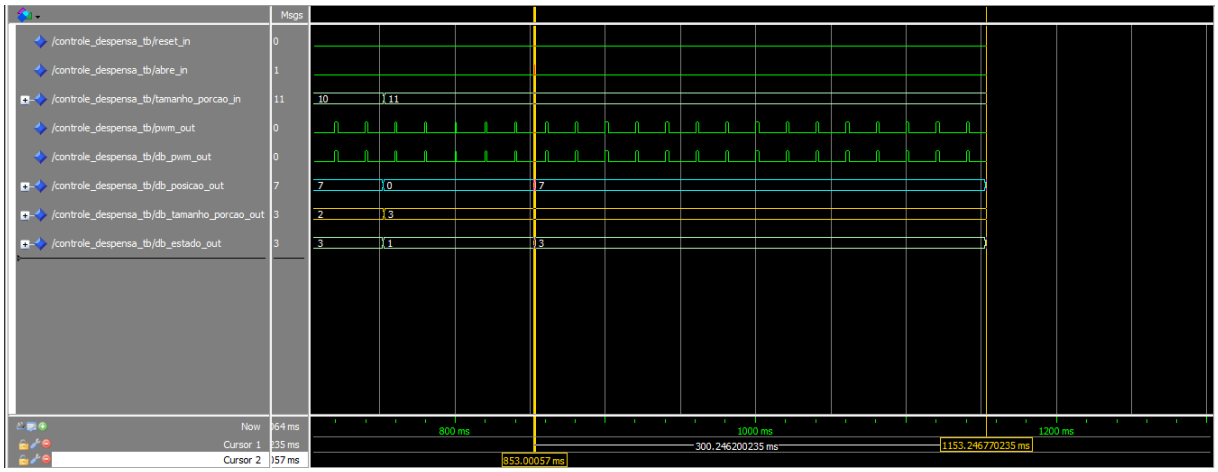

Figura 11: Tempo em aberto para ”tamanho_porcao”em ”11”

Como todos os testes tiveram resultados satisfatórios, o resultado da testagem em simulação pode ser considerada um sucesso.

3.3 Aplicação da Placa

3.3.1 Modificações para a montagem

Para a montagem na bancada, adicionou-se um simples sinal de depuração: **”db_aberto”**, que sinaliza quando o circuito está no estado de ”aberto”.

Este sinal foi criado para, na bancada, utilizar a ferramenta Scope do Waveforms

para realizar a medição precisa do tempo em que o circuito ficou em aberto, verificando se o funcionamento é adequado.

3.3.2 Pinagem do circuito

A pinagem do circuito foi realizada de acordo com a seguinte tabela:

Sinal	Tipo	Significado	Pino
clock	Input	CLOCK_50	M9
reset	Input	MQTT/E0	N16
abre	Input	MQTT/E1	B16
tamanho_porcao	Input	MQTT/E2, E3	M16,C16
pwm	Output	Servomotor	K16
db_aberto	Output	Scope CH2	F12
db_pwm	Output	Scope CH1	F15
db_tamanho_porcao	Output	LEDR2-3	Y3,W2
db_posicao	Output	LEDR7-9	L1,L2,U1
db_estado	Output	LEDR0-1	AA1,AA2

Figura 12: Pinagem do circuito para a aplicação

3.3.3 Testes em Bancada

Para verificar o circuito em bancada, realizamos as seguintes atividades:

- Testagem da largura da PWM por meio da ferramenta Scope;
- Testagem da largura do pulso ”db_aberto”para avaliar o tempo em aberto;
- Testagem para sinais erráticos (”abre”quando o circuito já está aberto e mudança do sinal de seleção do tamanho da porção);

Testagem da Largura da PWM

Utilizando a ferramenta Scope para fazer a medição da onda PWM e o aplicativo MQTT Dash para enviar os sinais, obtemos:

PWM com o circuito em aberto

A largura da PWM neste modo deve ser de 2ms:

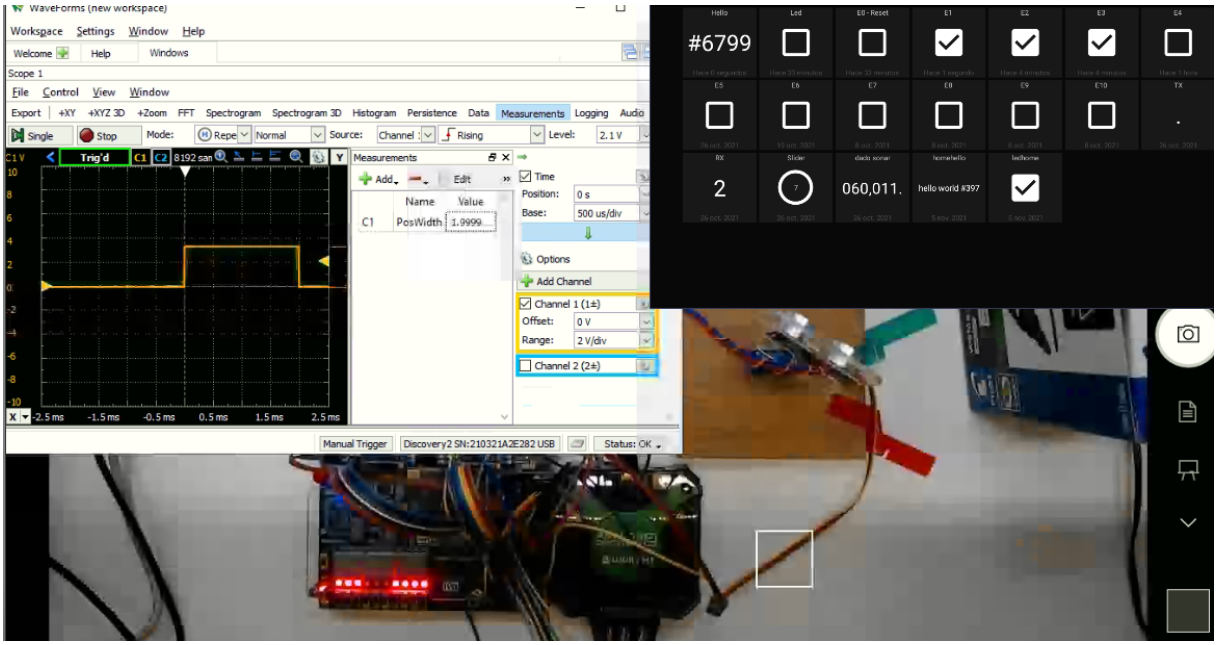

Figura 13: Avaliação da PWM em aberto por meio do Scope

PWM com o circuito em fechado

A largura da PWM neste modo deve ser de 1ms:

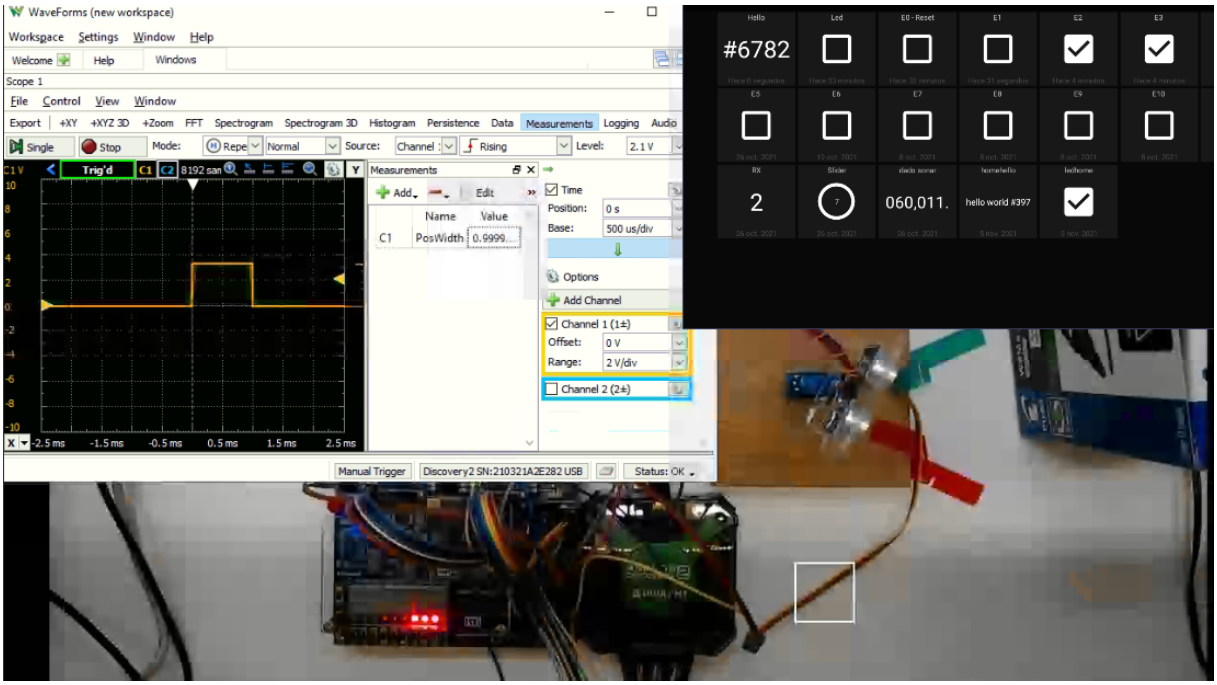

Figura 14: Avaliação da PWM em fechado por meio do Scope

Testagem do tempo em aberto

Novamente, utilizando a ferramenta Scope para medir a largura do pulso e controlando os sinais por MQTT Dash:

Tempo em aberto com ”tamanho_porcao” em ”00”

O tempo em aberto neste modo deve ser 1 segundo:

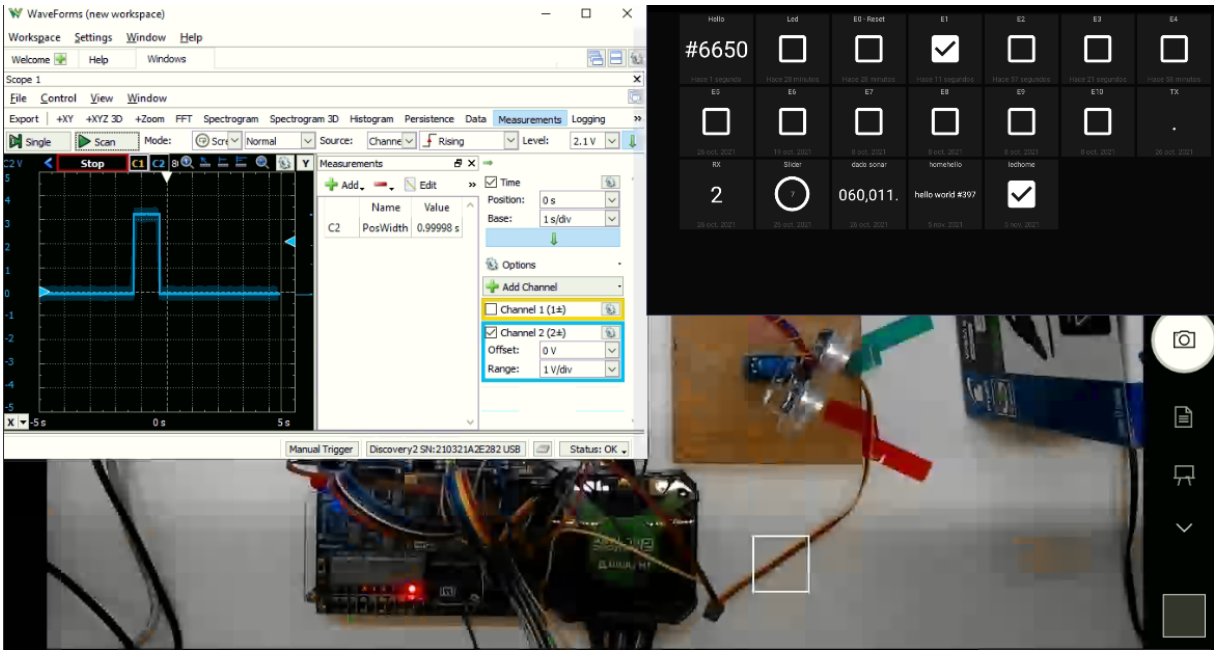

Figura 15: Tempo em aberto com ”tamanho_porcao”em ”00”

Tempo em aberto com ”tamanho_porcao” em ”01”

O tempo em aberto neste modo deve ser 1,5 segundos:

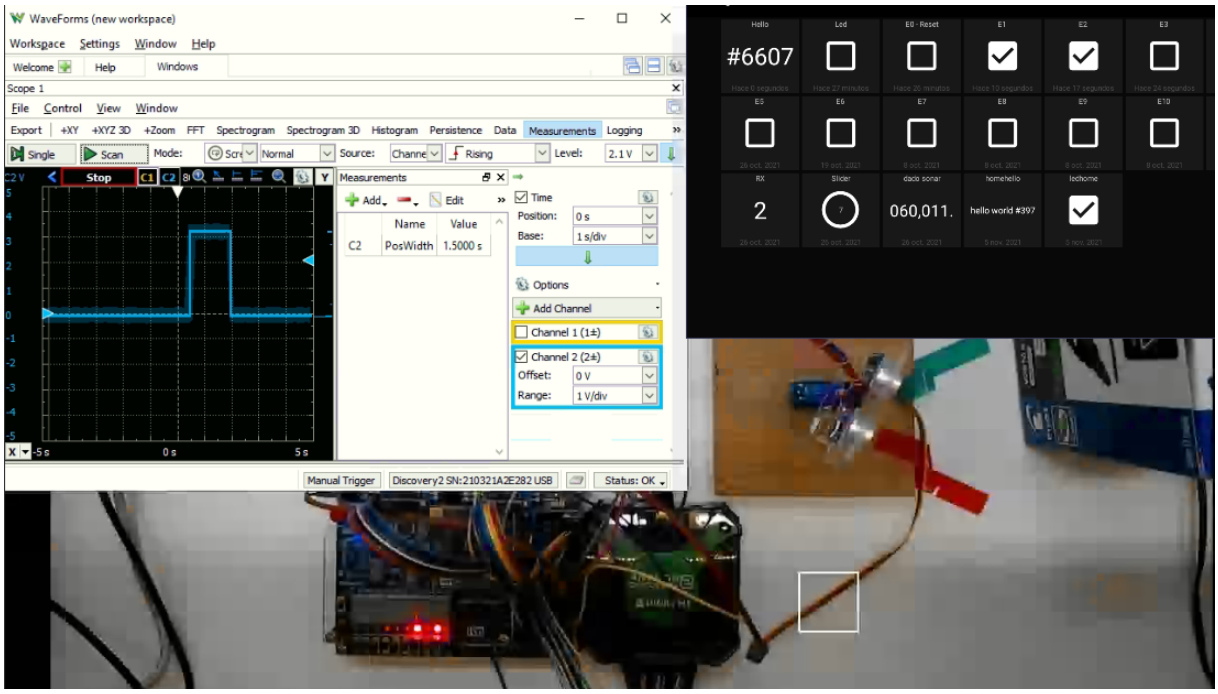

Figura 16: Tempo em aberto com ”tamanho_porcao”em ”01”

Tempo em aberto com ”tamanho_porcao” em ”10”

O tempo em aberto neste modo deve ser 2 segundos:

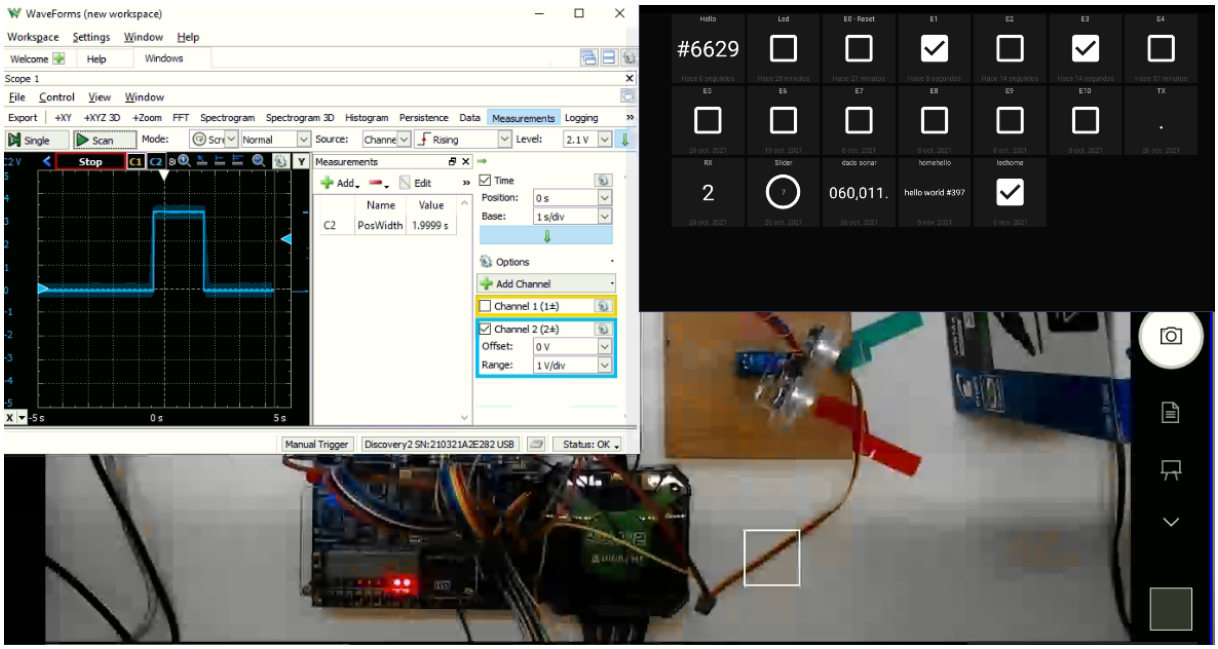

Figura 17: Tempo em aberto com ”tamanho_porcao”em ”10”

Tempo em aberto com ”tamanho_porcao” em ”10”

O tempo em aberto neste modo deve ser 3 segundos:

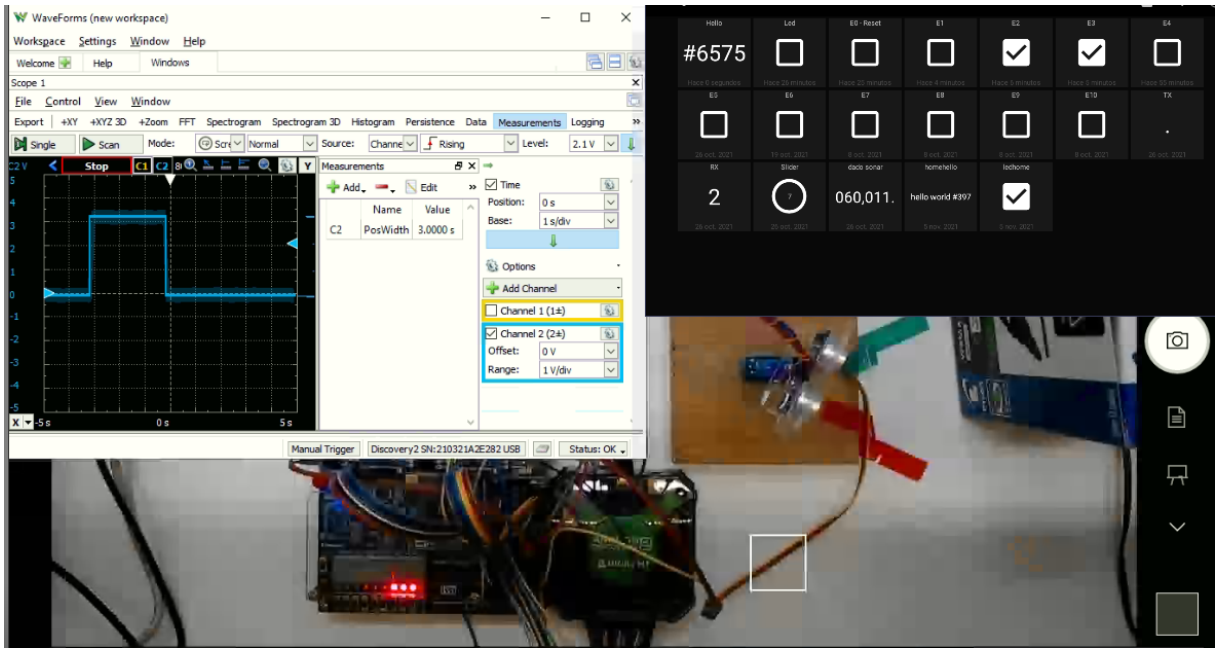

Figura 18: Tempo em aberto com ”tamanho_porcao”em ”11”

Como os testes em bancada foram todos bem sucedidos, conclui-se que a aplicação do circuito durante o laboratório digital foi um sucesso.

4 RESULTADOS

4.1 Conclusão

Durante este período de projeto, foi desenvolvido o módulo de movimentação do servomotor, responsável por controlar a despensa do alimento do pet. O dispositivo desenvolvido é capaz de fornecer 4 tamanhos diferentes de porções para o animal, possibilitando que o proprietário que decida o tamanho da refeição fornecida (caso o animal necessite de uma alimentação mais reduzida ou robusta).

Nas semanas seguintes, será desenvolvido os módulos restantes para o funcionamento completo do alimentador Neko Neko Nee:

4.2 O que já foi desenvolvido

- Dispositivo de controle para a despensa de comida;

4.3 O que resta ser desenvolvido

- Dispositivo de controle do módulo de distância;
- Máquina de estados geral do circuito;
- Módulo para comunicação serial;
- Módulo de recepção da comunicação serial e armazenamento dos dados;
- Módulo timer;

5 REFERÊNCIAS BIBLIOGRÁFICAS

(1) ALMEIDA, F.V. de; SATO, L.M.; MIDORIKAWA, E.T. Tutorial para criação de circuitos digitais em VHDL no Quartus Prime 16.1. Apostila de Laboratório Digital. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Edição de 2017.

(2) ALMEIDA, F.V. de; SATO, L.M.; MIDORIKAWA, E.T. Tutorial para criação de circuitos digitais hierárquicos em VHDL no Quartus Prime 16.1. Apostila de Laboratório Digital. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Edição de 2017.

(3) ALTERA / Intel. DE0-CV User Manual. 2015.

(4) ALTERA / Intel. Quartus Prime Introduction Using VHDL Designs. 2016.

(5) ALTERA / Intel. Quartus Prime Introduction to Simulation of VHDL Designs. 2016.

(6) D'AMORE, R. VHDL - descrição e síntese de circuitos digitais. 2a edição, LTC, 2012.

(7) WAKERLY, John F. Digital Design Principles & Practices. 4th edition, Prentice Hall, 2006.