

Universidade Federal de São Carlos - UFSCar
Ciência da Computação

Laboratório de Circuitos Digitais
Projeto Final

Turma – A

Mayk Tulio Bezerra, 727953
Prof. Fredy João Valente

São Carlos
12 de Dezembro de 2018

1. Introdução

Inicialmente a ideia é fazer um sistema lógico para controlar a pista de pedágio. Ou seja fazer uma FSM que controle um sistema de pedágio e auxilie na cobrança ajudando a identificar o automóvel. Minha lógica adotada foi o seguinte: o automóvel equipado com a etiqueta RFID, passa pelo primeiro sensor, quando o primeiro sensor desligar, a FSM identificar o automóvel, lembrando que a cobrança acontece depois da identificação, o servidor quando receber a identificação da etiqueta só vai verificar se os pagamentos estão “ok”, caso esteja ok ele vai abrir a cancela normal e identificar o automóvel pela FSM e depois fazer a cobrança, caso não esteja “ok” a cancela não vai abrir ligar a câmera e aguardar um responsável pela plataforma para fazer o liberação.

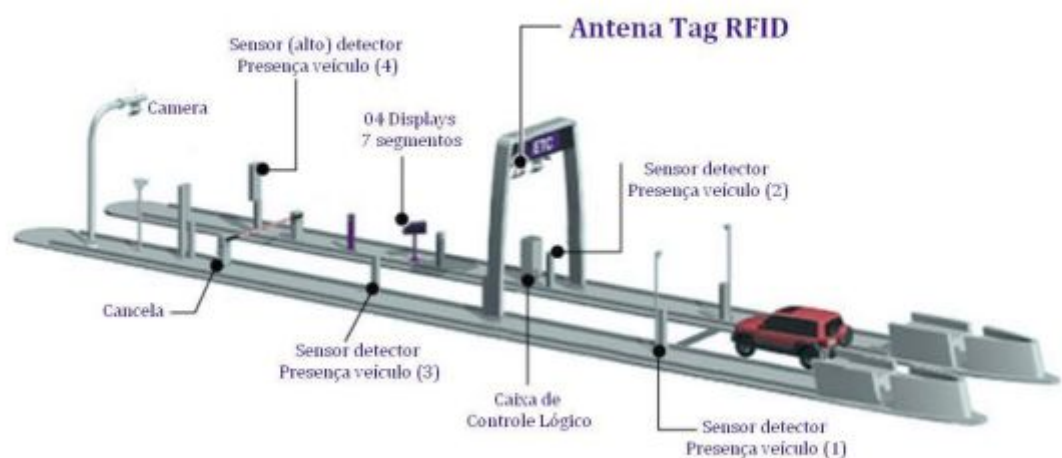


Imagem 1 - Projeto do sistema de pedágio a ser construído

Etiquetas RFID (Radio Frequency Identification), também conhecidas por Identificação por Rádio Frequência, são dispositivos de identificação e rastreamento através de um pequeno sinal de radiofrequência. Sua estrutura básica é bem simples: uma antena, responsável por captar informações, e um microchip, que é capaz de armazenar dados. Ambos são protegidos por um transponder, também conhecido como tag, composto por um material de plástico ou silicone, que pode ter diversos formatos.

Etiquetas RFID: o código fica gravado no chip, possibilita a leitura sem contato com o objeto e pode ser lido através de madeira, plástico, papel tecido, entre outros materiais. O sistema pode ler várias etiquetas simultaneamente, com maior distância e maior velocidade, o que resulta em maior produtividade. A única forma de ler o código sem a antena é tê-lo impresso em sua superfície.

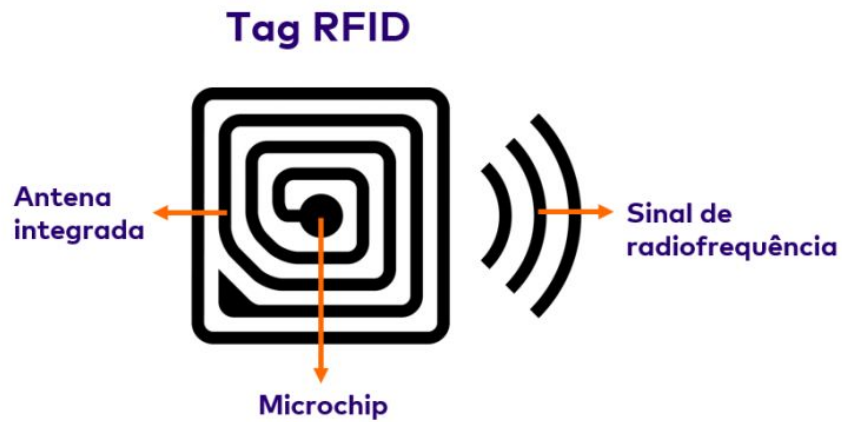


Imagem 2 - TAG RFID

2. Procedimento Experimental

A lógica adotada foi que o sistema de pilha fosse feito computacionalmente, e a FSM ficasse somente responsável pela identificação e liberação dos veículos, a seguir a máquina de estado obtido, com base nesse escopo, foi levado em consideração que o sistema de conferência de pagamento seria uma entrada na FSM. O tipo da FSM adotado foi Mealy em virtude que é um sistema que necessita de uma certa agilidade pois estamos trabalhando com veículos em movimento.

Atividade - sensor 1, sensor 2, sensor 3, sensor 4, situação (L ou NL)

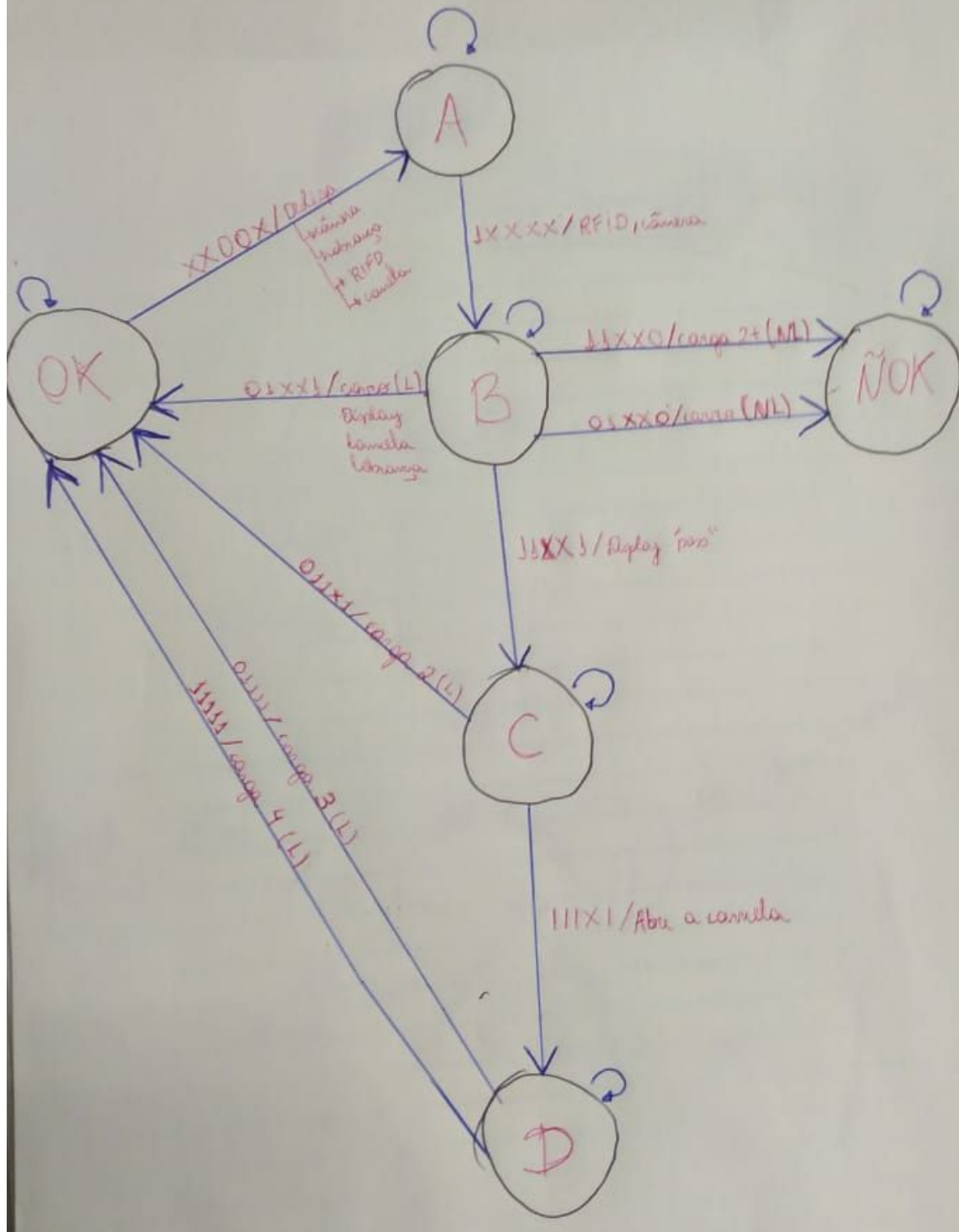


Imagem 3 - FSM do Projeto

Estados e seus significados:

A= Stand By

B= 1º sensor detectado

C= 2º sensor detectado

D= 3º sensor detectado

OK = Automóvel identificado/Liberado(PASS)/Cobrado/

ÑOK = Não Liberado (STOP)/Cancela Fechada/ Aguardando responsável da plataforma

Imagem 4 - Estados e Significados

A seguir as entradas e as saídas utilizadas no sistema:

ENTRADAS		SAÍDAS
SENSOR 1		LIGAR O RFID
SENSOR 2		DESLIGAR RFID
SENSOR 3		DISPLAY "PASS"
SENSOR 4		DISPLAY "STOP"
SITUAÇÃO (L OR NL)		ABRIR CANCELA
		FECHAR CANCELA
		LIGAR CAMERA
		DESLIGAR CÂMERA
		COBRANÇA

Imagem 5 - Entradas e saídas do sistema

```

1  module Mayk_PROJECT (HEX3, HEX2, HEX1, HEX0, SW, LEDR, LEDG, clock);
2
3  input [9:5] SW;
4  input clock;
5
6  output reg [0:6] HEX0;
7  output reg [0:6] HEX1;
8  output reg [0:6] HEX2;
9  output reg [0:6] HEX3;
10 output reg [7:0] LEDG;
11 output reg [7:0] LEDR;
12
13 reg [2:0] estado;
14 reg [4:0] entrada;
15 reg [3:0] automovel;
16 reg camera, RIFD, cancela, cobrar;
17
18 parameter A = 0, B = 1, C = 2, D = 3, OK = 4, NOK = 5;
19
20 /*
21 A = STAND BY
22 B = 1Â° SENSOR DETECTADO
23 C = 2Â° SENSOR DETECTADO
24 D = 3Â° SENSOR DETECTADO
25 OK = VEICULO AUTORIZADO A PASSAGEM
26 NOK = VEICULO NAO AUTORIZADO A PASSAGEM
27 */
28
29 always @(posedge clock) begin
30     entrada[4] = SW[9]; //A
31     entrada[3] = SW[8]; //B
32     entrada[2] = SW[7]; //C
33     entrada[1] = SW[6]; //D
34     entrada[0] = SW[5]; //L or NL(Liberado ou não liberado)
35     LEDG[7] = automovel[3];
36     LEDG[6] = automovel[2];
37     LEDG[5] = automovel[1];
38     LEDG[4] = automovel[0];
39     LEDR[0] = camera;
40     LEDR[1] = RIFD ;
41     LEDR[2] = cancela;
42     LEDR[3] = cobrar;
43

```

Imagem 6 - Primeira parte do código

```

44 case(estado)
45 A:begin
46     if(entrada[4] == 1) begin
47         estado = B;
48         camera = 1;
49         RIFD = 1;
50     end
51 end
52 else begin
53     estado = A;
54     HEX3 = 7'b1111111;
55     HEX2 = 7'b1111111;
56     HEX1 = 7'b1111111;
57     HEX0 = 7'b1111111;
58     cobrar = 0;
59     camera = 0;
60     RIFD = 0;
61     cancela = 0;
62     automovel[3]=0;
63     automovel[2]=0;
64     automovel[1]=0;
65     automovel[0]=0;
66 end
67 end
68 end
69
70 B:begin
71     if((entrada[4]==1 && entrada[3]==1 && entrada[0]==1) || (entrada[4]==0 && entrada[3]==1 && entrada[0]==1) ||
72        (entrada[4]==0 && entrada[3]==1 && entrada[0]==0) || (entrada[4]==1 && entrada[3]==1 && entrada[0]==0)) begin
73
74         if(entrada[4]==1 && entrada[3]==1 && entrada[0]==1)begin
75             estado = C;
76             HEX3 = 7'b0011000;
77             HEX2 = 7'b0001000;
78             HEX1 = 7'b0100100;
79             HEX0 = 7'b0100100;
80             //DISPLAY PASS
81         end

```

Imagem 7 - Segunda parte do código


```

82     if(entrada[4]==0 && entrada[3]==1 && entrada[0]==1)begin
83         estado = OK;
84         automovel[3]=1;
85         HEX3 = 7'b0011000;
86         HEX2 = 7'b0001000;
87         HEX1 = 7'b0100100;
88         HEX0 = 7'b0100100;
89         cancela = 1;
90         cobrar = 1;
91
92         //CARRO,DISPLAY PASS,ABRIR CANCELA,COBRAR
93     end
94     if(entrada[4]==0 && entrada[3]==1 && entrada[0]==0) begin
95         estado = NOK;
96         HEX3 = 7'b0100100;
97         HEX2 = 7'b0001111;
98         HEX1 = 7'b0000001;
99         HEX0 = 7'b0011000;
100        //Display STOP
101    end
102    if(entrada[4]==1 && entrada[3]==1 && entrada[0]==0)begin
103        estado = NOK;
104        HEX3 = 7'b0100100;
105        HEX2 = 7'b0001111;
106        HEX1 = 7'b0000001;
107        HEX0 = 7'b0011000;
108        //Display STOP
109    end
110
111    end else begin
112        estado = B;
113    end
114 end
115
116 C:begin
117     if((entrada[4]==1 && entrada[3]==1 && entrada[2]==1 && entrada[0]==1) ||
118        (entrada[4]==0 && entrada[3]==1 && entrada[2]==1 && entrada[0]==1))begin
119
120     if(entrada[4]==1 && entrada[3]==1 && entrada[2]==1 && entrada[0]==1)begin
121         estado = D;
122         cancela = 1;
123         //ABRIR CANCELA
124     end

```

Imagem 8 - Terceira parte do código


```

124   end
125   if(entrada[4]==0 && entrada[3]==1 && entrada[2]==1 && entrada[0]==1)begin
126       estado = OK;
127       automovel[3]=1;
128       automovel[2]=1;
129       HEX3 = 7'b0011000;
130       HEX2 = 7'b0001000;
131       HEX1 = 7'b0100100;
132       HEX0 = 7'b0100100;
133       cancela = 1;
134       cobrar = 1;
135
136       //CARGA 2, DISPLAY "PASS",ABRIR CANCELA, COBRAR
137   end
138
139   end else begin
140       estado = C;
141   end
142
143
144   end
145
146   D:begin
147   if((entrada[4]==0 && entrada[3]==1 && entrada[2]==1 && entrada[1]==1 && entrada[0]==1) ||
148       (entrada[4]==1 && entrada[3]==1 && entrada[2]==1 && entrada[1]==1 && entrada[0]==1))begin
149
150       if(entrada[4]==0 && entrada[3]==1 && entrada[2]==1 && entrada[1]==1 && entrada[0]==1)begin
151           estado=OK;
152           automovel[3]=1;
153           automovel[2]=1;
154           automovel[1]=1;
155           cobrar = 1;
156
157
158           //CARGA 3, COBRAR
159       end
160       if(entrada[4]==1 && entrada[3]==1 && entrada[2]==1 && entrada[1]==1 && entrada[0]==1)begin
161           estado=OK;
162           automovel[3]=1;
163           automovel[2]=1;
164           automovel[1]=1;
165           automovel[0]=1;
166           cobrar = 1;

```

Imagem 9 - Quarta parte do código

```

166         cobrar = 1;
167         //CARGA 4, COBRAR
168     end
169 end else begin
170     estado = D;
171 end
172
173 end
174
175 OK:begin
176     if(entrada[1]==0 && entrada[2]==0)begin
177         estado = A;
178         cobrar = 0;
179         camera = 0;
180         RIFD = 0;
181         cancela = 0;
182
183     end
184 else begin
185     estado = OK;
186 end
187 end
188
189 NOK:begin
190     //SOAR ALARME da placa
191 end
192
193 default: estado = A;
194 endcase
195 end
196 endmodule

```

Imagem 10 - Parte final do código

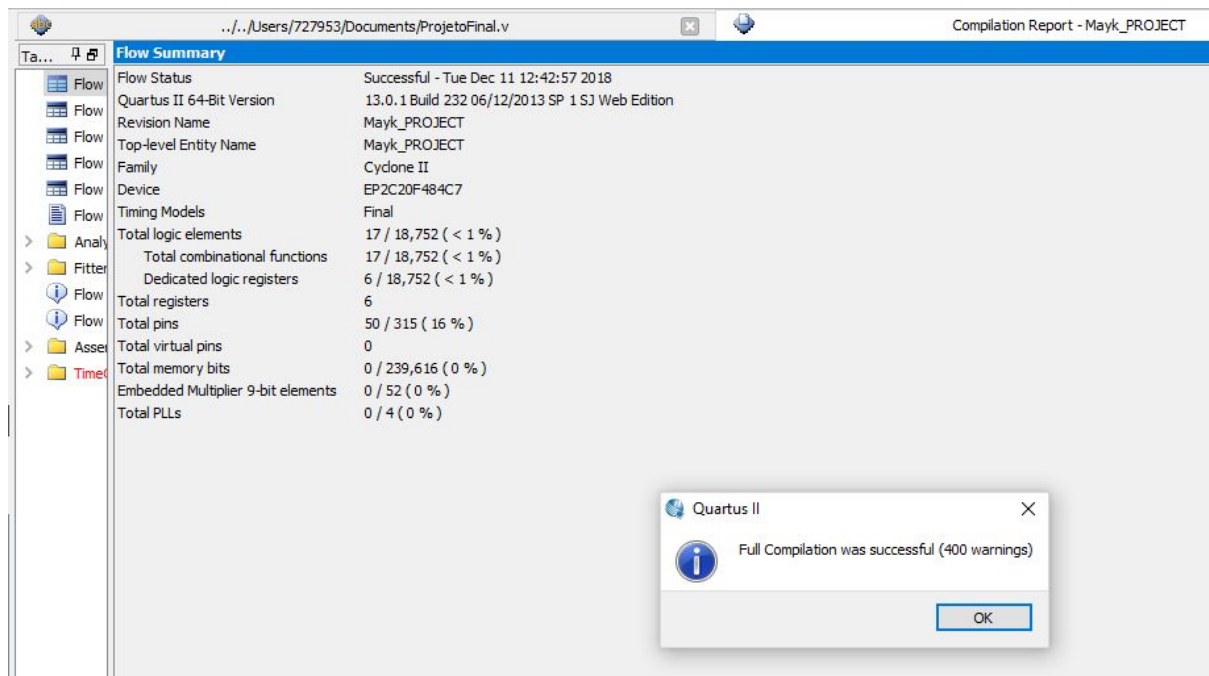


Imagem 11 - Compilação do Projeto

3. Apresentação dos Resultados

3.1 Caso teste 1 (Carro)

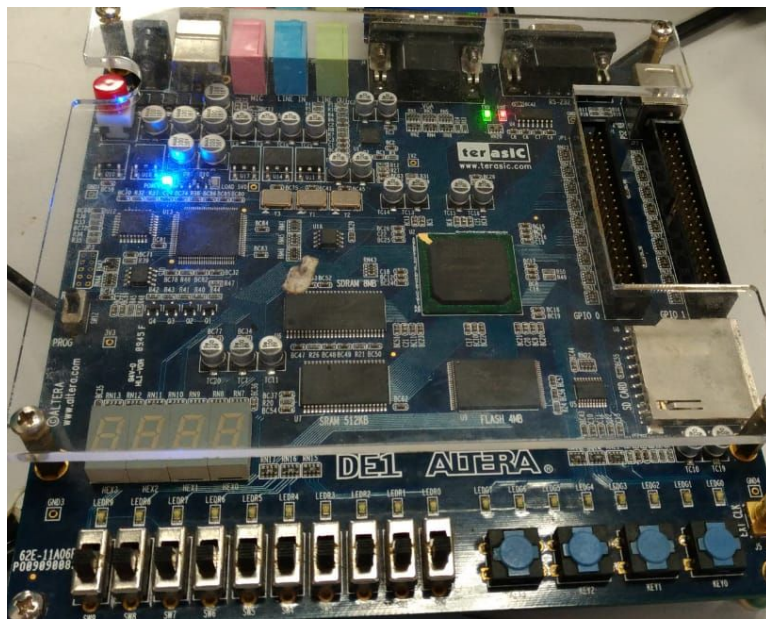


Imagem 12 - Stand By do sistema

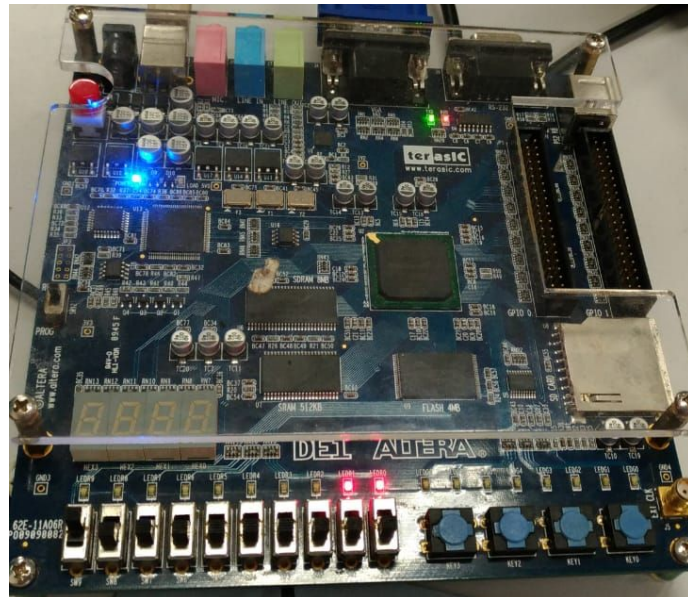


Imagem 13 - Carro passa pelo sensor 1 e liga camera e RFID

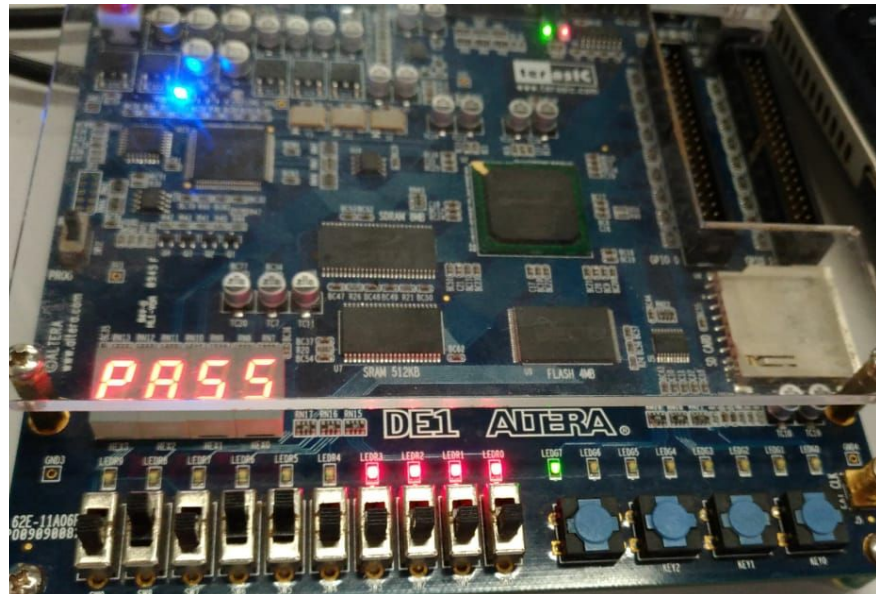


Imagem 14 - Carro passa pelo Sensor 2 é identificado, cancela abre e ele é cobrado

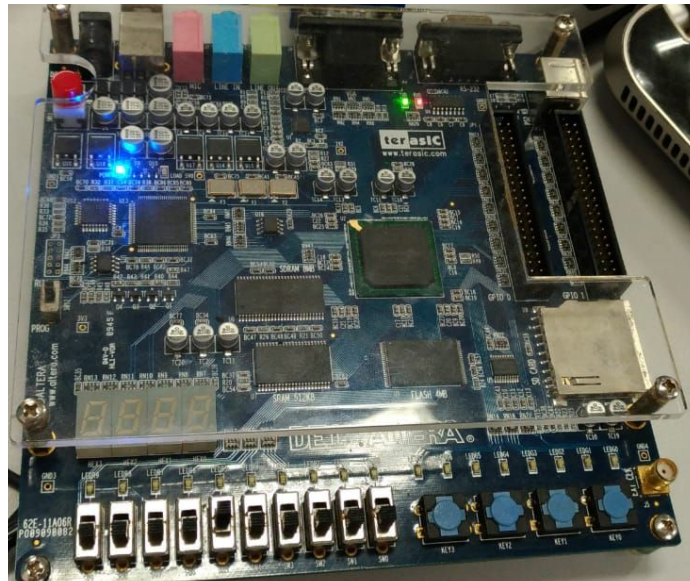


Imagem 15 - Quando o 4º sensor e o 3º sensor desligar ele volta para o estado Stand by

3.2 Caso teste 2 (Carreta Tamanho 4)

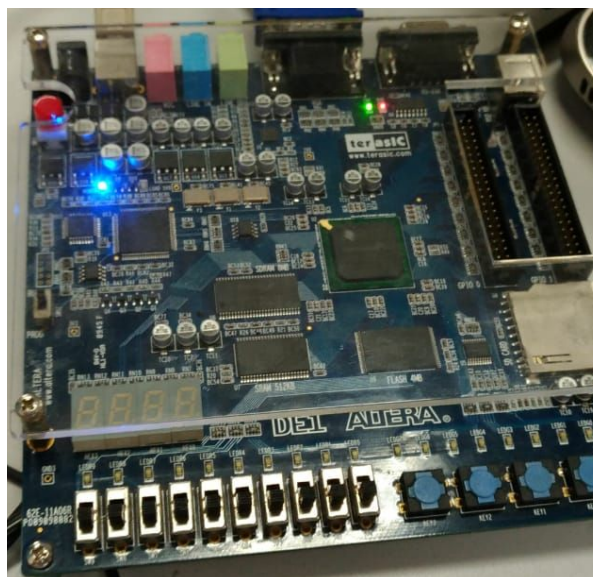


Imagem 16 - Stand By do sistema

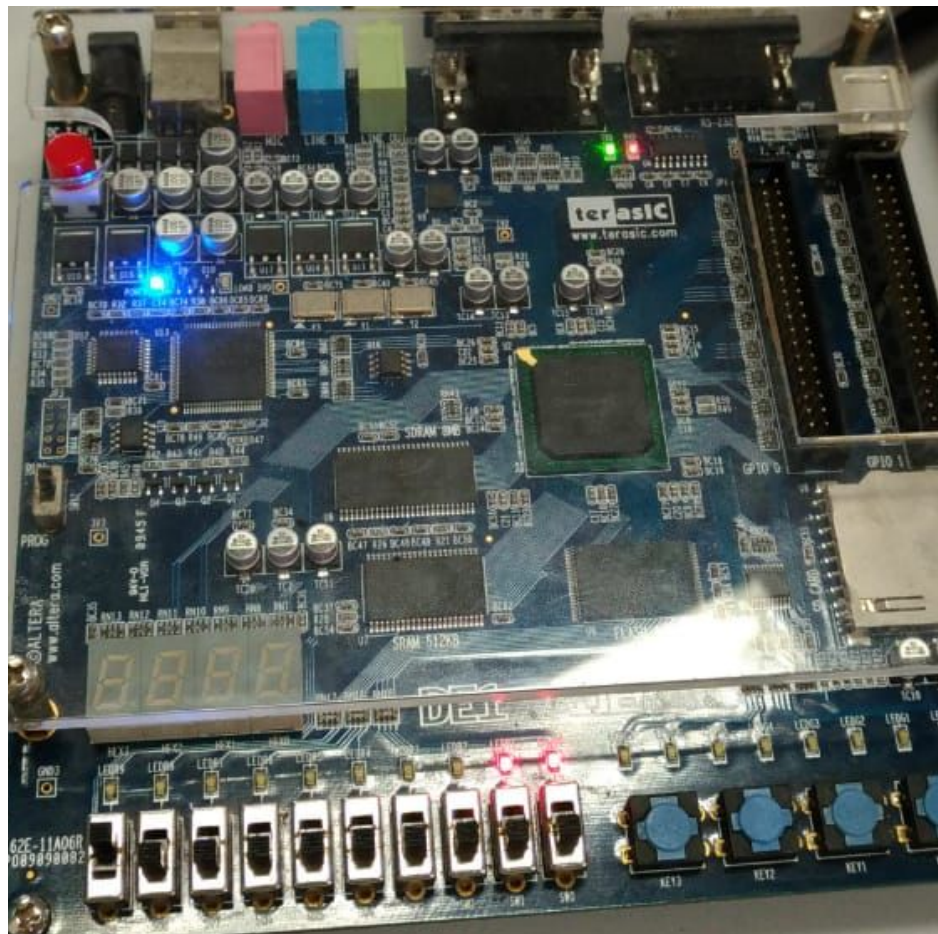


Imagem 17 - Carreta passa pelo sensor 1, liga camera e RFID

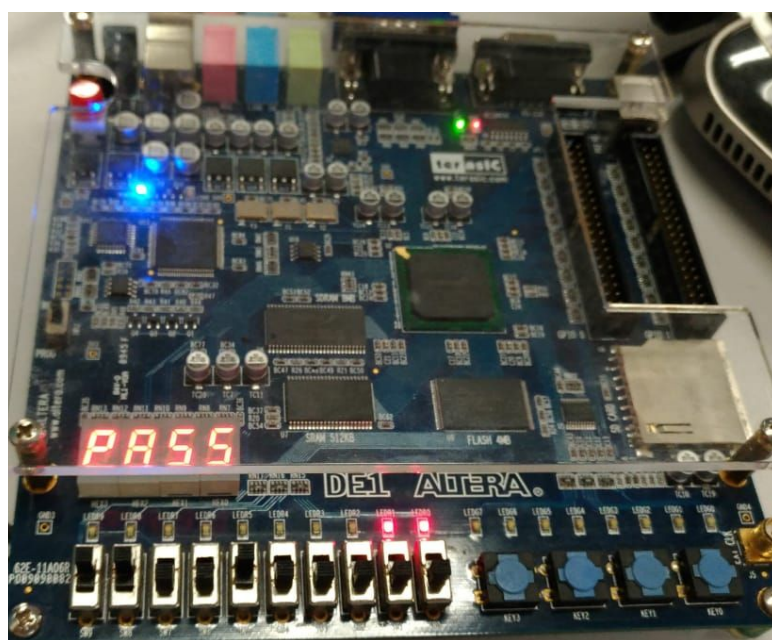


Imagem 18 - Carreta passa pelo sensor 2, mostra display PASS

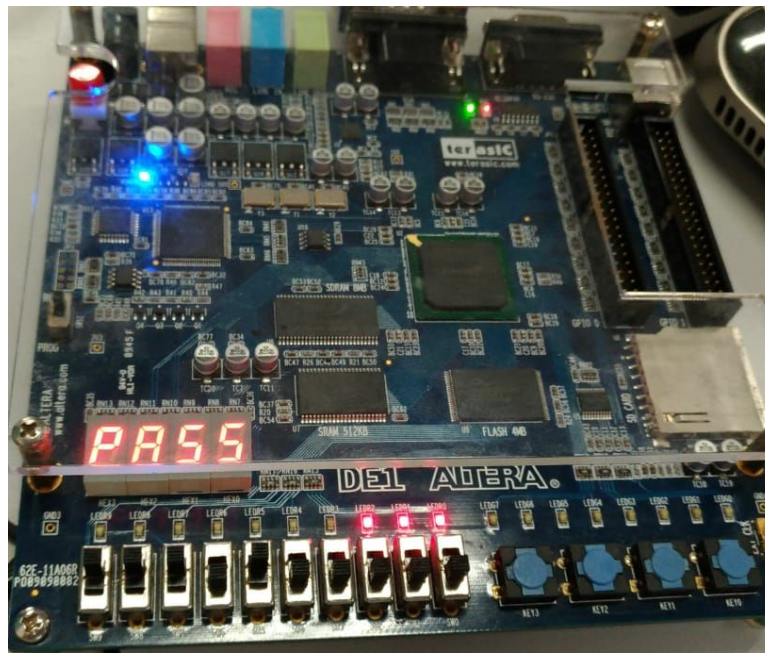


Imagem 19 - Carreta passa pelo sensor 3, abre cancela

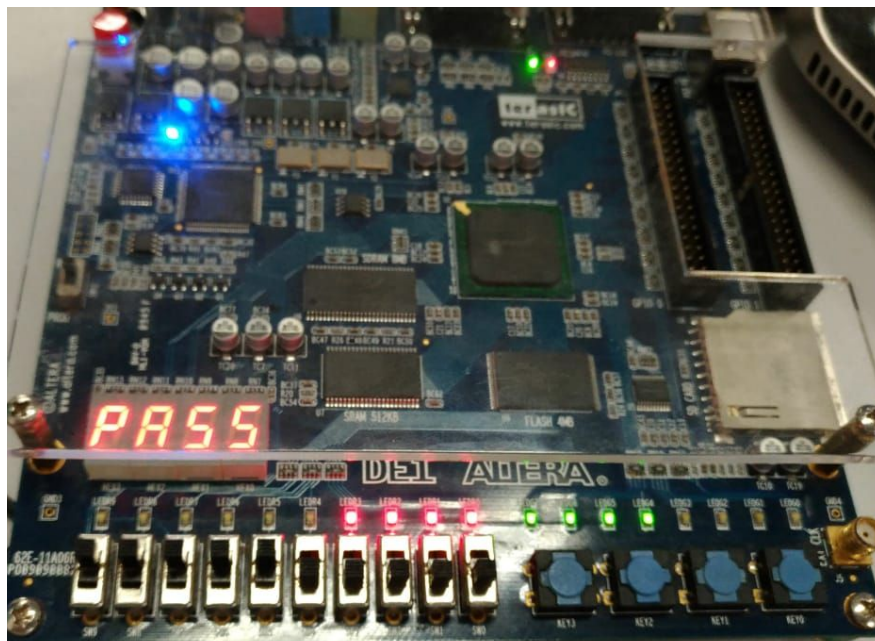


Imagem 20 - Carreta passa pelo sensor 4 é identificada e cobrada

3.3 Caso teste 3 (Carreta Tamanho 2)

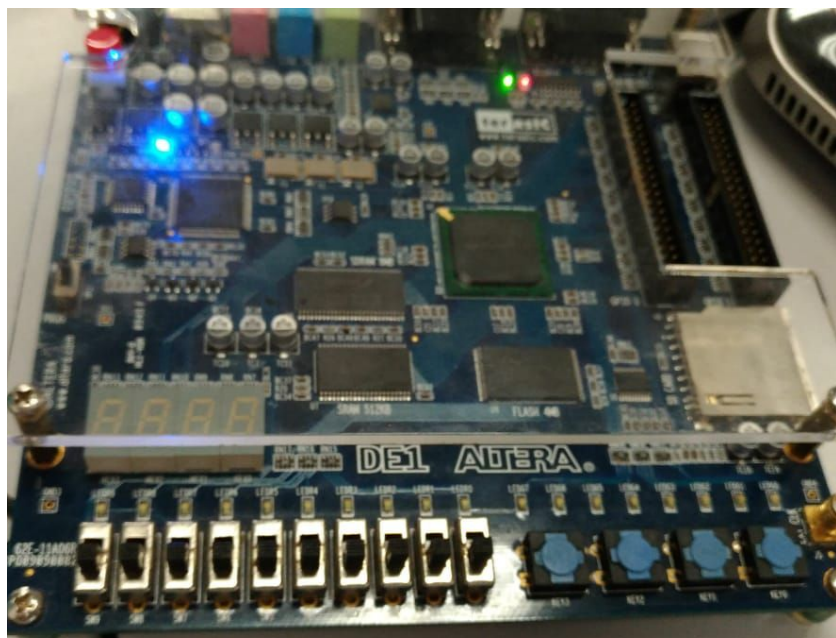


Imagem 21 - Stand by do sistema

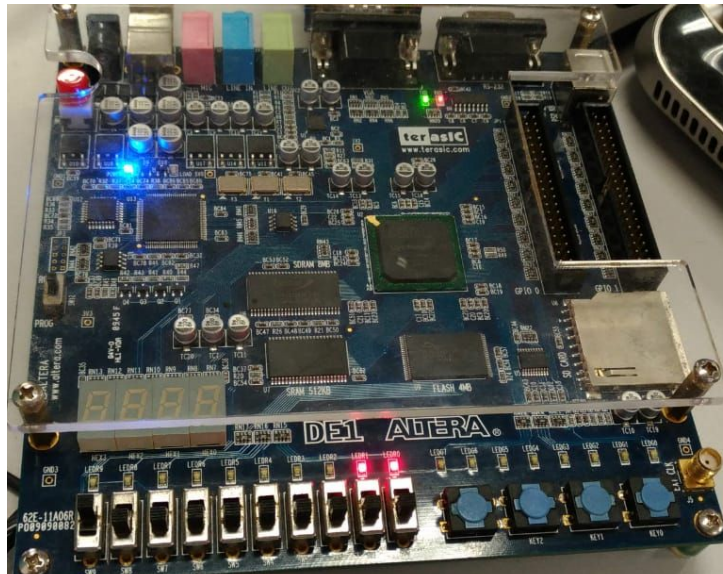


Imagem 22 - Carreta passa pelo sensor 1, liga camera e RFID

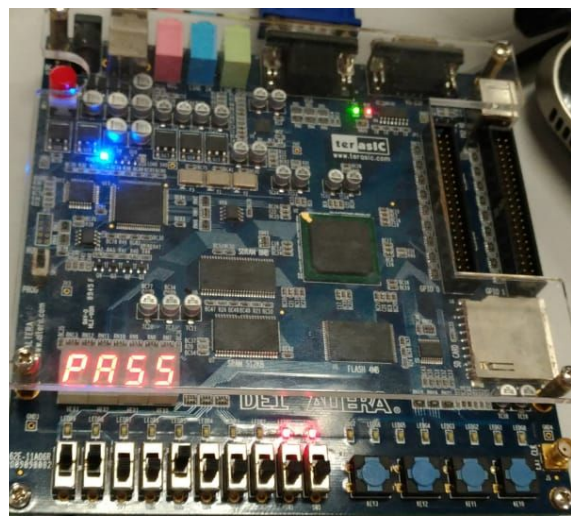


Imagem 23 - Carreta passa pelo sensor 2, liga display “PASS”

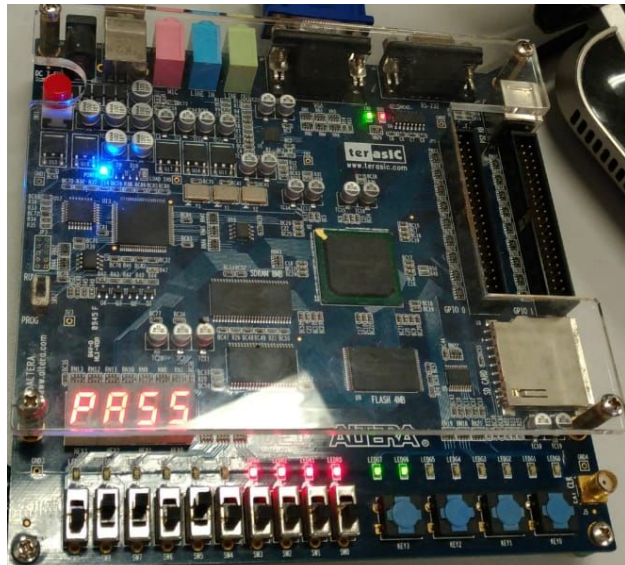


Imagem 24 - Carreta passa pelo sensor 3, identificada, cancela aberta e cobrado

3.4 Caso teste 4 (CASO NL)

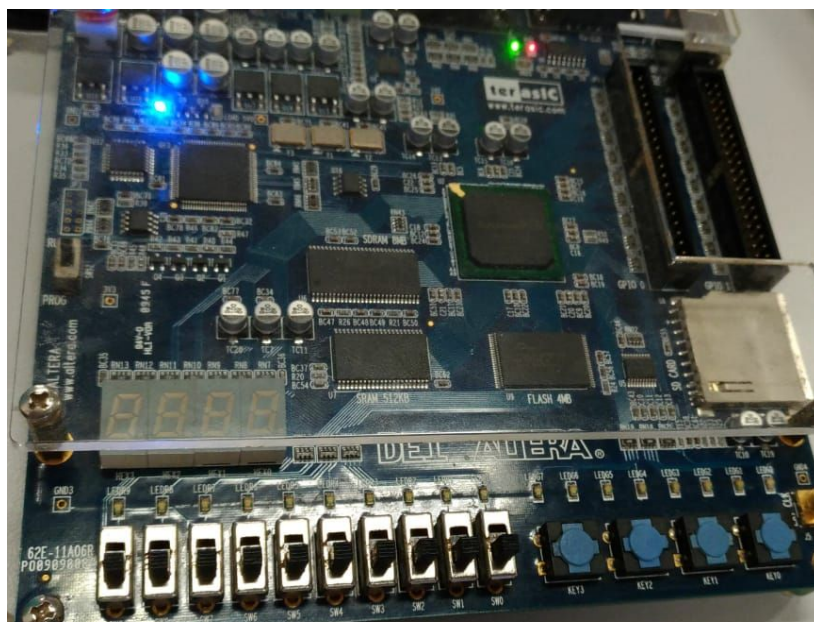


Imagem 25 - Stand By do sistema

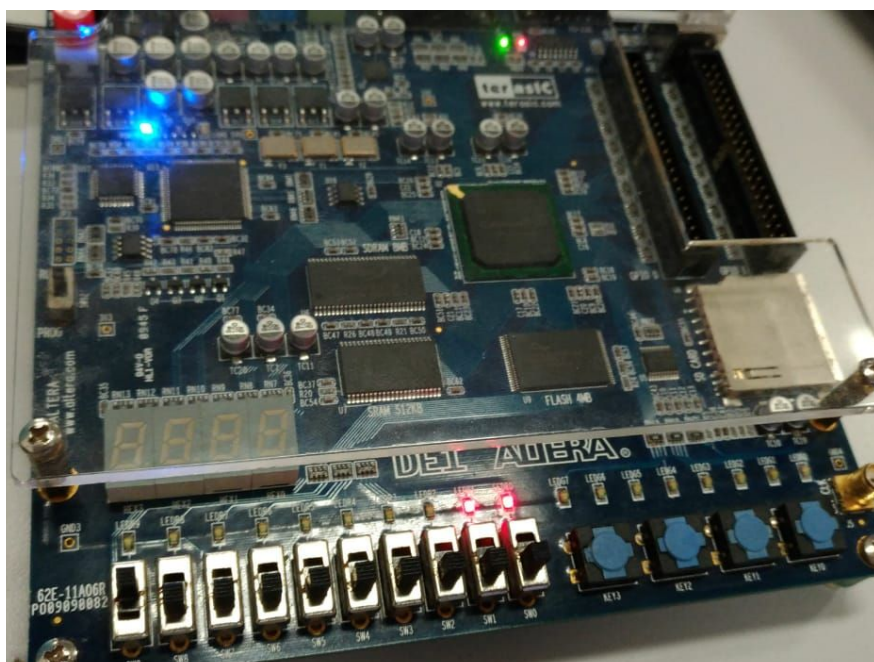


Imagem 26 - Veículo passa pelo sensor 1, liga RFID e câmera

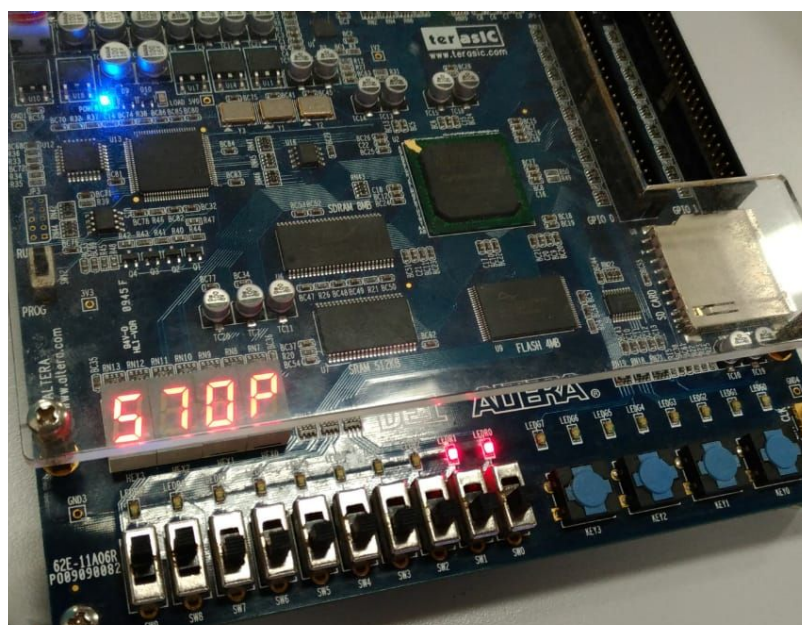


Imagem 27 - Veículo passa pelo sensor 2, identifica inadimplência, soa o alarme, cancela não abre e display mostra STOP

4. Conclusão

Podemos perceber que para implementar um sistema de um simples pedágio de verdade é bem mais complexo do que foi implementado aqui, aqui neste último projeto pude utilizar todo o conhecimento adquirido ao longo do curso tanto na parte teórica quanto no que foi praticado no laboratório, para tornar um sistema completo ainda falta alguns ajustes. A máquina funcionou muito bem com as condições impostas.

A maior dificuldade assim como nos outros trabalhos foi em montar a FSM e como seria o escopo, e em quais condições a nossa máquina trabalharia, isso ocupou o trabalho de 80% do nosso tempo, a parte da programação da FSM foi bem mais fácil pois segue um determinado padrão quanto a implementação das FSM. Achei muito interessante o RFID não o conhecia, mas creio que seja esse sistema que foi implementado na BCo, onde cada livro tem um código, evitando assim perdas com o uso de entradas de mochila na BCo. Gostaria de agradecer pelo semestre, foi um salto bem grande no quesito aprendizado.