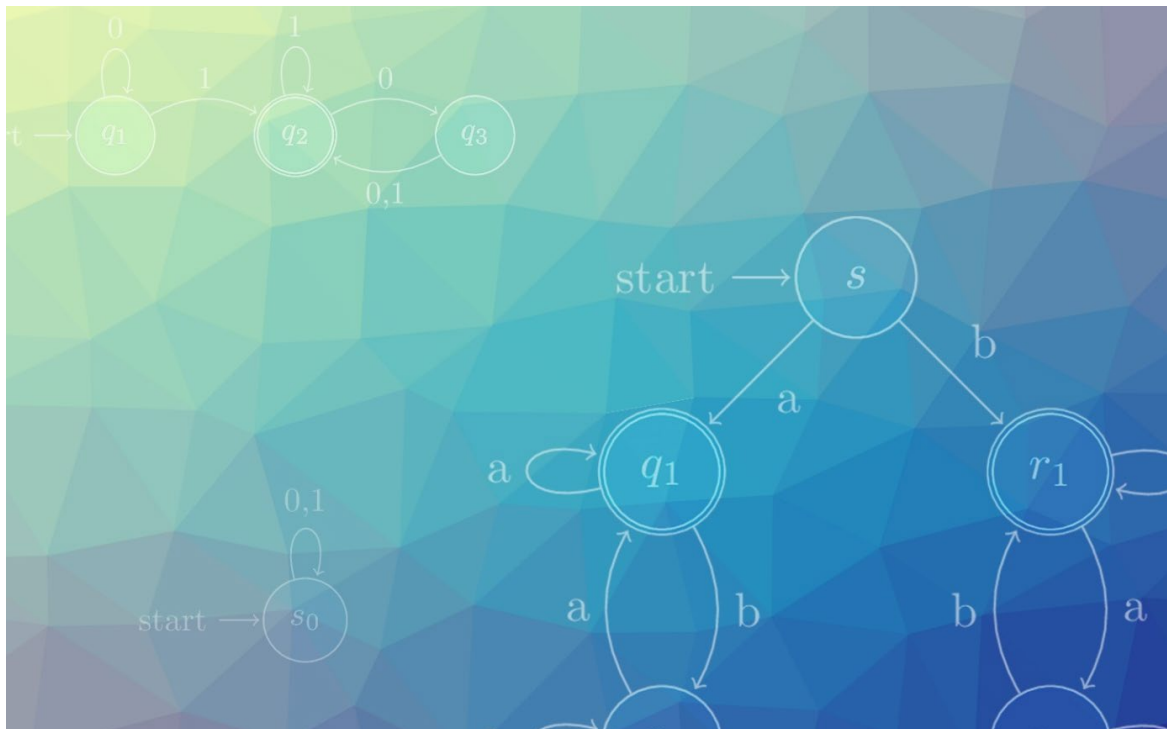


Sistemas Embebidos

Lab. 2 – Timed Automata



Trabalho realizado por:

Diogo Silva, nº190204007

Mussagy Abibo, nº190204013

Enunciado do problema

Neste laboratório, iremos abordar os seguintes temas: Comunicação Wi-Fi, autômato temporizador e a comunicação segura.

Para a realização do mesmo, necessitamos de um sistema composto por um Router Wi-Fi e duas máquinas (um PC e um ESP8266) para comunicar através de dados encriptados por Wi-Fi. O PC dá comandos e faz com que ESP8266 execute. Os comandos seguem uma linguagem pré-definida que deve ser aceite no ESP8266.

A encriptação escolhida por nós foi a Route Cipher, mas, pode ser feita de várias formas, no qual, escolhemos *Side-to-Side*. O funcionamento da mesma é muito simples. Preenche-se a palavra pretendida numa tabela com apenas 3 colunas e depois escreve-se a mesma palavra formada numa linha, como no exemplo abaixo:

Frase: Sistemas Embebidos

Sistemas embebidos [18]

s [0]	t [3]	a [6]	e [9]	e [12]	d [15]
i [1]	e [4]	s [7]	m [10]	b [13]	o [16]
s [2]	m [5]	- [8]	b [11]	i [14]	s [17]

s t a e e d i e s m b o s m b i s
[0][3][6][9][12][15][18][21][24][27][30][33][36][39][42][45][48][51]

A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X

(Side-to-side I)

Exemplo utilizado no Laboratório:

%	D	/0
L	5	
E	&	
%DL5E&	desencriptação ->	%LED5&

Além disso, também escolhemos os comandos corretos, são eles, os que começam com o sinal de percentagem (%), terminados pelo i comercial (&) e, que no seu conteúdo (excluindo os caracteres de único e fim) tenham a palavra “LED seguida de um número, podemos chamá-lo de C, em que o C é o contador do número de vezes que o LED irá acender. Mais tarde, implementámos um delay, e convencionámos que C seria o tempo que o LED estaria aceso.

Relativamente ao temporizador, o autômato apenas pode executar uma instrução após 3 segundos e, tem uma tolerância de 5 segundos, além desse momento é impedida a sua execução.

Exemplos de comandos válidos:

%LED5&

%LED1&

%LED2&

And so on...

Teoria do Autômato

Após reunir os dados referidos anteriormente, utilizamos um programa para enviar uma String encriptada para o endereço IP do ESP8266.

Como já tínhamos feito algo semelhante em Redes de Computadores, decidimos utilizar o programa Hercules, e, por fim, prosseguimos para a programação do ESP8266.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	
1	%	E	5	L	D	&	&		
2	%	E	5	L	D	&	E		replyPacket[6] = replyPacket[1]
3	%	L	5	L	D	&	E		replyPacket[1] = replyPacket[3]
4	%	L	5	D	D	&	E		replyPacket[3] = replyPacket[4]
5	%	L	5	E	D	&	E		replyPacket[3] = replyPacket[6]
6	%	L	5	S	D	&	E		replyPacket[3] = replyPacket[2]
7	%	L	E	5	D	&	E		replyPacket[2] = replyPacket[6]
8	%	L	E	5	D	&	5		replyPacket[6] = replyPacket[3]
9	%	L	E	D	D	&	5		replyPacket[3] = replyPacket[4]
10	%	L	E	D	S	&	6		replyPacket[4] = replyPacket[6]

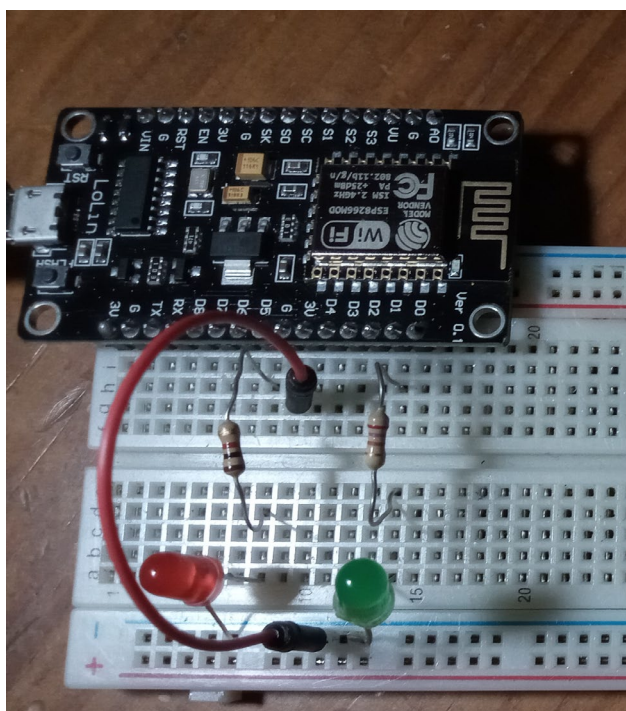
Processo de desencriptação da string segundo Route Cipher

Montagem do Circuito Eletrônico

A montagem do circuito eletrônico que inclui os seguintes componentes:

- 2 Resistências, 1K e 270 Ohm;
- 2 LED's, vermelho e verde;
- Cabo USB;
- Cabos de ligação;
- ESP8266;
- Breadboard

A montagem foi feita de acordo com esquema abaixo:



d) Implementação de temporização

Esta última secção do programa não foi possível ser implementada por completo, uma vez que as dificuldades anteriores ocuparam mais tempo, assim como outros problemas técnicos. No entanto, implementámos temporização utilizando *delay()* de acordo com os comandos recebidos – tempo que o LED acende.

Em conclusão, este trabalho serviu-nos como forma de implementar autómatos com temporizadores, ajudando assim na compreensão desta matéria.

Conclusão

Tivemos mais dificuldades tanto na implementação do problema na programação. Através de uma interpretação estruturada, Conseguimos partir o programa em várias etapas, nomeadamente:

- a) Conexão Wifi
- b) Receção dos caracteres/strings
- c) Descriptação