

MANUAL PROGRAMADOR

Manual Operador

Cliente: YouDo
Sr. Fernando

MRD Engenharia
Rua: Arthur Schopenhauer, 586
São José dos Pinhais-PR
CEP: 83030-205
Fone: (41) 3081-4654/(41) 99161-7929/
E-mail: amauri.deus@mrd.net.br



Sumário

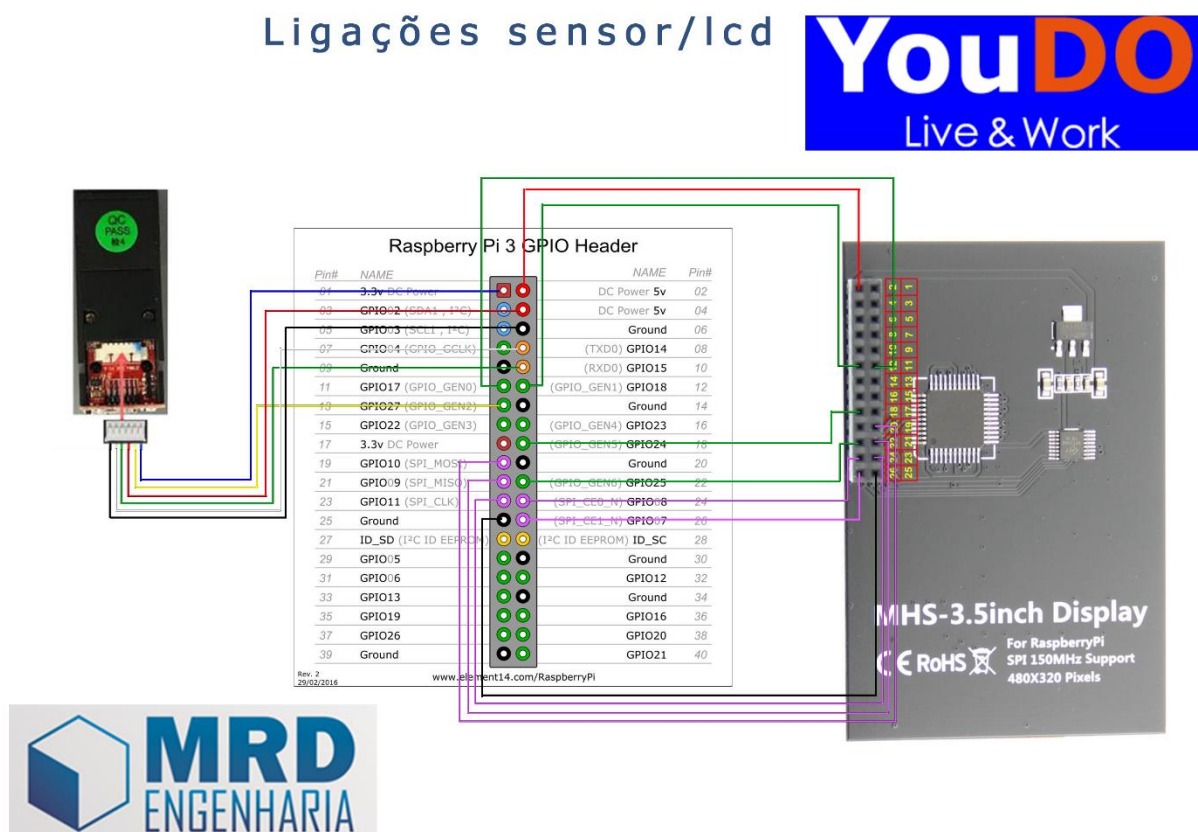
1	Apresentação	1-3
2	Instalação do raspberry pi	2-3
2.1	Ativações e ligações on-board do raspberry pi.....	2-3
2.2	Ativação das portas de comunicação dos raspberry.....	2-4
2.3	Drivers e configuração do display LCD 3.5	2-4
2.4	Autorização das função de criação e edição através do serviço PHP.....	2-5
3	Funções python.	3-6
3.1	Lista das funções:	3-7
4	HTML/JAVASCRIPT/PHP	4-9
4.1	Index.php.....	4-9
4.2	Javascript.....	4-10
5	Configurações do serviço SQL.	5-11
6	Fluxo entre os arquivos.	6-11
7	Restrições sugestões e cuidados.	7-12

1 Apresentação

Este manual tem por fim auxiliar o programador a desenvolver uma aplicação para raspberry pi 3

2 Instalação do raspberry pi

2.1 Ativações e ligações on-board do raspberry pi.



Ligações do LCD 3.5 as ligações do display não podem ser alteradas pois estão na disposição de encaixe do próprio LCD.

Ligações do Fingerprint, é de extrema importância que a alimentação do sensor de toque seja alimentado com 3.3v, ou se for ligado no 5v é necessário regular com um resistor antes da entrada do sensor "fio azul".

O fio amarelo é a do sinal do sensor na programação está sendo usado na programação atual GPIO27 entrada física de numero 13.

A alimentação principal do sensor pode ser tanto 3.3v quanto 5v "fio vermelho".



REV0.1

A comunicação do fingerprint é rx, tx além da ligação ser obrigatória nos pinos 8 e 10 (GPIO14, GPIO15) deve ser liberado a comunicação destes pinos na configuração do raspberry, cap 2.2.

2.2 Ativação das portas de comunicação dos raspberry.

Ativação da comunicação rx tx das GPIO14 – 15:

Edite o arquivo config.txt do raspberry pi, procure ou adicione a linha `enable_uart=0` e mude para `enable_uart=1`.

Agora será necessário parar e desabilitar o uso da serial pelo sistema execute os seguintes comandos:

```
sudo systemctl stop serial-getty@ttyS0.service
sudo systemctl disable serial-getty@ttyS0.service
```

É necessário remover também a inicialização da serial através da `/boot/cmdline.txt`, edite este arquivo removendo a parte

```
console=serial0,115200.
```

Remova também a linha do arquivo `/etc/inittab`

```
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

2.3 Drivers e configuração do display LCD 3.5

Baixe e instale os drivers Elecrow_LCD35 através dos comandos:

```
git clone https://github.com/Elecrow-keen/Elecrow-LCD35.git
cd Elecrow-LCD35
sudo ./Elecrow-LCD35
```

Mudando a tela para posição retrato:

Edite o arquivo `/boot/config.txt` alterando ou adicionando as seguintes linhas:

```
dtoverlay=ads7846,penirq=17,penirq_pull=2,speed=1000000,xohms=60
xohms=60,xmin=200,xmax=3900,ymin=200,ymax=3900
dtoverlay=elecrow35a,speed=22000000,fps=60,rotate=0
```

Calibração do touch da tela LCD3.5:

Edite o arquivo `/etc/X11/xorg.conf.d/99-calibration.conf`:

```
Section "InputClass"
    Identifier "calibration"
    MatchProduct "ADS7846 Touchscreen"
    Option "Calibration" "3917 250 242 3863"
    Option "SwapAxes" "0"
    Option "InvertY" "0"
    Option "InvertX" "1"
```

2.4 Autorização das função de criação e edição através do serviço PHP.

O software de exemplo necessita executar e editar alguns arquivos para isso é necessário dar permissões ao serviço e usuário do serviço http.

Edite o arquivo /etc/sudoers adicionando a seguinte linha:

```
www-data ALL=(ALL) NOPASSWD:ALL
```

Na pasta destinada ao serviços html (por padrão pasta /var/www/html/) execute os seguintes comandos após adicionar os arquivos referentes ao programa:

```
sudo chown www-data:www-data modo.txt  
sudo chmod 644 modo.txt
```

Use estes comandos acima alterando o nome do arquivo “modo.txt” para que o programa php possa fazer alterações nos arquivos desejados.

3 Funções python.

O programa principal arquivo main.py tem por objetivo chamar as funções referentes principalmente ao leitor de digital:

```
1  import sys
2  import functions
3  import os
4
5  argument = sys.argv[1]
6
7  if argument == "-src":
8      functions.waitForFinger()
9  elif argument == "-add":
10     functions.addFinger()
11  elif argument == "-del":
12     functions.removeFinger(sys.argv[2])
13  elif argument == "-sqlSrc":
14     functions.sqlSearch()
15  elif argument == "-sqlAdd":
16     functions.sqlEnroll(sys.argv[2])
17  else:
18     print argument
```

(imagem 01)

O main.py necessita ser executado adicionando 1 ou 2 argumentos dependendo da função que deseja chamar da seguinte forma:

```
main.py [-argumento1 int(argumento2)]
```

Via ssh você pode usar o comando:

```
sudo python main.py [-argument1 int(argument2)]
```

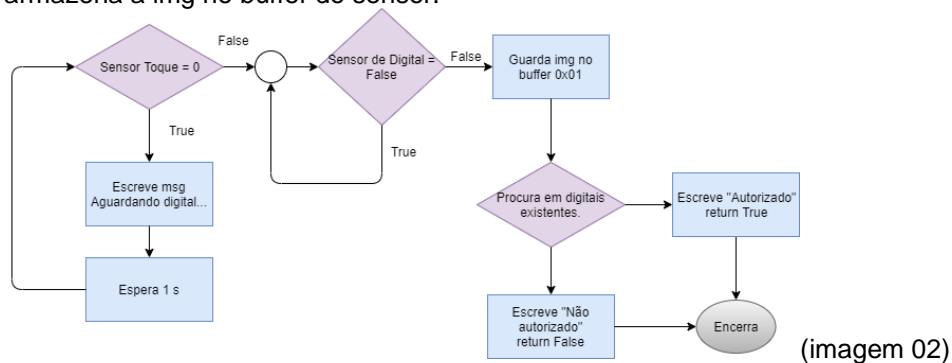
Substituindo "argumento" pela função desejada como por exemplo:

```
sudo python main.py -src
```

3.1 Lista das funções:

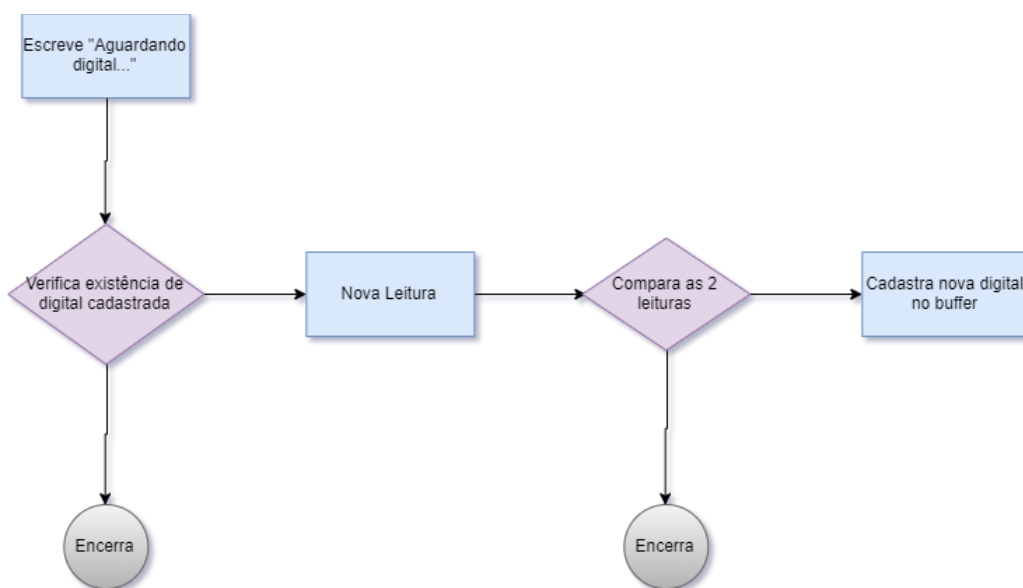
-src:

Esta função é a função de verificação da digital, primeiro ela espera em loop o contato do dedo no sensor de toque, após o contato, espera leitura de digital valida em loop, após armazena a img no buffer do sensor.



-add

Esta função adiciona uma digital ao buffer do sensor. Primeiro ela escreve msg "Aguardando nova digital...", chama a função de leitura de digital, verifica existência no buffer interno, em caso de existência informa digital já cadastrada.

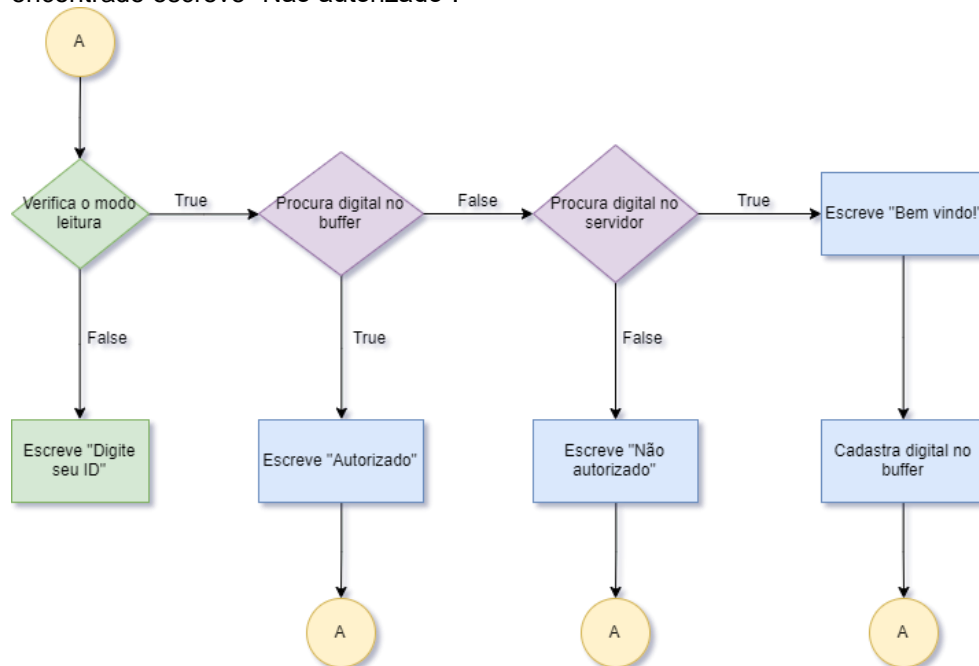


-del [int(argument)]

Esta função remove digital cadastrada no buffer do sensor através de sua posição id, necessário adicionar o argumento de posição da digital cadastrada.

-sqlSrc

Esta função procura digital cadastrada primeiramente no buffer do sensor em caso negativo procura no servidor sql quando encontrado no servidor adiciona no buffer, quando não encontrado escreve "Não autorizado".



(imagem 04)

-sqlEnroll [int(argument)]

Esta função adiciona digital ao cadastro já existente no banco de dados sql através da id do usuário, sendo necessário adição da ID do usuário cadastrado no SQL.

4 HTML/JAVASCRIPT/PHP

4.1 Index.php

A sugestão é que o index da interface do raspberry seja feita em PHP e separada em algumas DIVs de acordo com o exemplo a seguir: (código completo no arquivo index.php).

```
23 <table>
24 <tbody>
25 <tr>
26 <td>
27 <div id="exec"><?php shell_exec("sudo /usr/bin/python /var/www/html/main.py -sqlSrc> /dev/null 2>&1 & echo $!") ?></div>
28 <div id="show"><?php include 'msg.txt' ; ?></div>
29 <p>&nbsp;</p>
30 </td>
31 </tr>
32 <tr>
33 <td width="345" height="300"><div id="centro"></div></td>
34 </tr>
35 <tr>
36 <td style="text-align: center;"><a href="#" onclick="trocaDiv();"></a></td>
37 </tr>
38 </tbody>
39 </table>
```

(imagem 05)

São 3 DIVs principais:

exec que será responsável por executar o programa python assim que iniciada ou atualizada.

show que é responsável por mostrar na tela as mensagens do programa python.

centro que quando necessário será alterada para possibilitar o preenchimento de dados.

4.2 Javascript

E adicionada as seguintes funções javascript:

```
40 <script type="text/javascript">
41   var divAtual = 1;
42   function trocaDiv(){
43     if (divAtual == 1) {
44       $.get("adduser.html", function( data ){
45         $('#centro').html(data);
46         $('#button').attr('src', "../imgs/menuButton.png");
47         divAtual = 2;
48       });
49       $.ajax({type: 'post',url:'escrita.php',data: {
50         'modo': 'escrita',
51         'id': 'nada'
52       }});
53     } else if (divAtual == 2) {
54       $('#centro').html('');
55       $('#button').attr('src', "../imgs/addButton.png");
56       $('#exec').load("execMain.php");
57       doRefresh();
58       divAtual = 3;
59       $.ajax({type: 'post',url:'escrita.php',data: {
60         'modo': 'leitura',
61         'id': 'nada'
62       }});
63     } else {
64       $.get("adduser.html", function( data ){
65         $('#centro').html(data);
66         $('#button').attr('src', "../imgs/menuButton.png");
67         doRefresh();
68         divAtual = 2;
69       });
70       $.ajax({type: 'post',url:'escrita.php',data: {
71         'modo': 'escrita',
72         'id': 'nada'
73       }});
74     }
75   };
76   function doRefresh(){
77     $("#show").load("msg.txt");
78   }
79   $(function() {
80     setInterval(doRefresh, 800);
81   });
82 </script>
```

(imagem 06)

Função trocaDiv(), responsável por fazer as transições na DIV Centro pois quando chamada ela altera o centro para um teclado de dígitos numéricos, **alterna o modo de leitura para escrita** e alterna também o botão entre menu e home.

Função doRefresh(): responsável por ler as informações geradas pelo programa python em um intervalo de 800 milissegundos e mostra-las na DIV show.

5 Configurações do serviço SQL.

A aplicação de exemplo usa um serviço interno de sql do raspberry porém o mesmo pode ser alterado mudando as configurações no arquivo sqlConect.py:

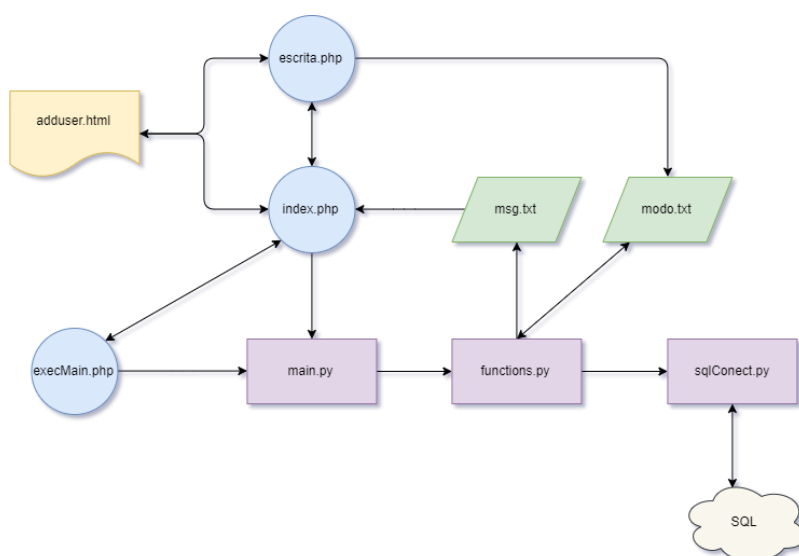
```
3  conf = {
4      "host": "localhost",
5      "user": "phpprincipal",
6      "passwd": "sql12345",
7      "database": "bancoNomesFinger",
8      "tabela": "cadastro",
9      "colHexFinger": "finger",
10     "ident": "id"
11 }
```

(imagem 07)

As variáveis “colHexFinger” referência a coluna onde irá ficar gravada o código HEX da imagem da digital do usuário.

A variável “ident” referência ao índice do usuário que necessita fazer adição do HEX da digital, este campo deve ser preenchido com apenas número pois é utilizado um teclado numérico para preenchimento do mesmo na aplicação exemplo.

6 Fluxo entre os arquivos.



(imagem 08)

7 Restrições sugestões e cuidados.

O serviço php não é totalmente seguro para utilização dos hardwares ligado no Raspberry é possível e sugerido para este escopo utilizar o “Tkinter” que é uma plataforma de desenvolvimento mais indicada para esta aplicação.

Existem outros drivers possíveis para utilização do LCD3.5 porém a maioria deles exigem uma maior performance do LCD assim torna-se necessário alimentações com maior corrente, se caso desejar será necessária uma alimentação externa para o LCD.

A programação desenvolvida como exemplo está executando em loop infinito por se tratar de uma aplicação com interface frontend feita em php se ficar sem alteração por mais de 1hr o serviço pode parar, para correção desta limitação foi desenvolvida dentro da programação uma alteração constante da leitura do display simultaneamente com o sensor de digital, é sugerido fazer o mesmo no software final.