

```

1  % APP WASPMOTE MONITORING DATA
2  %CREATED BY RUBÈN ALMANSA WITH MATLAB FOR EXIT RESEARCH GROUP
3
4
5  function varargout = APP(varargin)
6
7  % Seguir els comentaris en CATALÀ, i les funcions que tenen contingut
8  % els que estan en anglés o no hi ha contingut ho fa el propi Matlab per
9  % defecte
10
11
12
13  %Aquest codi el posa l'aplicació i no és pot modificar.
14  % Begin initialization code - DO NOT EDIT
15  gui_Singleton = 1;
16  gui_State = struct('gui_Name',       mfilename, ...
17                    'gui_Singleton',   gui_Singleton, ...
18                    'gui_OpeningFcn',  @APP_OpeningFcn, ...
19                    'gui_OutputFcn',   @APP_OutputFcn, ...
20                    'gui_LayoutFcn',   [], ...
21                    'gui_Callback',    []);
22  if nargin && ischar(varargin{1})
23      gui_State.gui_Callback = str2func(varargin{1});
24  end
25
26  if nargout
27      [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
28  else
29      gui_mainfcn(gui_State, varargin{:});
30  end
31  % End initialization code - DO NOT EDIT
32
33
34  % Part del codi que s'executa abans d'iniciar el programa
35  % Executa el logo i crear el contingut dela desplegable
36  function APP_OpeningFcn(hObject, eventdata, handles, varargin)
37  axes(handles.axes1);
38  I=imread('logo.jpg'); % inserta el logo del grup
39  imshow(I);
40  % mostra el contingut de cada desplegable el popmenu1 és el del waspmote
41  set(handles.popupmenu1,'String',{'EVENTS_DOOR',
42    'EVENTS_WINDOW','SMART_CITIES','GASES','PEOPLE','POWER_CONSUMITION'});
43  set(findall(handles.popupmenu1, '-property', 'enable'), 'enable', 'off');
44  set(handles.Events_Door,'String',{'Temperature_Door',
45    'Luminosity_Door(%)','Vibrations','Door_State','Movements_Door','Humidity'});
46  set(findall(handles.Events_Door, '-property', 'enable'), 'enable', 'off');
47  set(handles.Events_Window,'String',{'Temperature_Window',
48    'Luminosity_Window(%)','Window_State1','Window_State2','Window_State3','Window_State4',
49    'Movements_Window'});
50  set(findall(handles.Events_Window, '-property', 'enable'), 'enable', 'off');
51  set(handles.Smart_Cities,'String',{'Temperature_SC',
52    'Luminosity_SC(%)','Noise_avg','Noise_max','Luxometer'});
53  set(findall(handles.Smart_Cities, '-property', 'enable'), 'enable', 'off');
54  set(handles.Gases,'String',{'Temperature_Gases',
55    'Pressure','Outdoor_temperature','Voltage_CO2','ppm_CO2'});
56  set(findall(handles.Gases, '-property', 'enable'), 'enable', 'off');
57  set(handles.People,'String',{'People_avg','People_instant'});
58  set(findall(handles.People, '-property', 'enable'), 'enable', 'off');
59  set(handles.Power,'String',{'Power_avg'});
60  set(findall(handles.Power, '-property', 'enable'), 'enable', 'off');
61  set(handles.ADDvar, '-property', 'enable'), 'enable', 'off');
62  set(findall(handles.Import, '-property', 'enable'), 'enable', 'off');
63  set(findall(handles.ImportCSV, '-property', 'enable'), 'enable', 'off');
64  % This function has no output args, see OutputFcn.
65  % hObject      handle to figure
66  % eventdata    reserved - to be defined in a future version of MATLAB
67  % handles      structure with handles and user data (see GUIDATA)
68  % varargin     command line arguments to APP (see VARARGIN)
69
70  % Choose default command line output for APP
71  handles.output = hObject;
72
73  % Update handles structure

```

```

68 guidata(hObject, handles);
69
70 % UIWAIT makes APP wait for user response (see UIRESUME)
71 % uiwait(handles.figure1);
72
73
74 % --- Outputs from this function are returned to the command line.
75 function varargout = APP_OutputFcn(hObject, eventdata, handles)
76 % varargout cell array for returning output args (see VARARGOUT);
77 % hObject     handle to figure
78 % eventdata   reserved - to be defined in a future version of MATLAB
79 % handles     structure with handles and user data (see GUIDATA)
80
81 % Get default command line output from handles structure
82 varargout{1} = handles.output;
83
84
85 %S'executa quan és prem el botó de connect és el primer botó de tots per
86 %començar a funcionar
87 function Connect_Callback(hObject, eventdata, handles)
88 try
89     %Afegim el connector al path de funcions java
90     javaaddpath('mysql-connector-java-5.1.27-bin.jar');
91     %fem un import per tenir-hi accés
92     import java.sql.*;
93
94     javaaddpath('myDB.jar');
95
96     %creem la connexió a la BD
97     a=DataBase();
98     %a.setUrl('10.10.10.1');
99     a.setUrl('84.88.154.120');
100    a.setPort('3306');
101    a.setUser('rootMesh');
102    a.setPass('libelium2016');
103    a.conecta();
104    handles.ConnD = a.getConnection();
105
106    guidata(hObject, handles);
107
108    %Apareix una finestra que diu que esta connectat correcte
109
110
111    if ~isempty(handles.ConnD)
112
113        msgbox('Connected');
114        set(findall(handles.popupmenu1, '-property', 'enable'), 'enable', 'on');
115        set(findall(handles.Connect, '-property', 'enable'), 'enable', 'off');
116
117    else
118        errordlg('Connection failed!','Error');
119    end
120    catch err
121        %Error al Conectar
122        errordlg(err.message,'Error');
123
124    DADES{end} = {};
125    i=0;
126    handles.i=i;
127    handles.DADES=DADES;
128    guidata(hObject, handles);
129 end
130 % hObject     handle to Connect (see GCBO)
131 % eventdata   reserved - to be defined in a future version of MATLAB
132 % handles     structure with handles and user data (see GUIDATA)
133
134
135 % Es basa en casos que segons el que és tria en el desplegable assigna una
136 % posició d'on esta contingut per després anar a buscar on es troba aquell contingut.
137 % i assigna el valor d'aquell contingut en la grafica per a posterior mostrar-ho
138 % tots aquest desplegables en total 6 funcionen igual
139 function Events_Door_Callback(hObject, eventdata, handles)
140 try

```

```

141 strf = get(hObject,'String');
142 valf = get(hObject,'Value');
143
144 id = '4102eccc'; % la id identifica el waspmote en la base de dades.
145
146 switch strf{valf};
147
148     case 'Temperature_Door'
149         pos = '1'; % la "pos" indica la posició de la columna on es troba la
150             dada en BBDD.
151         set(handles.varDATA,'String','°C'); % col·loca la unitat la variable en la
152             gràfica (eix y)
153
154     case 'Luminosity_Door(%)'
155         pos = '2';
156         set(handles.varDATA,'String','%');
157     case 'Vibrations'
158         pos = '3';
159         set(handles.varDATA,'String','mV')
160     case 'Door_State'
161         pos = '4';
162         set(handles.varDATA,'String','')
163     case 'Movements_Door'
164         pos = '5';
165         set(handles.varDATA,'String','mov')
166     case 'Humidity'
167         pos = '6';
168         set(handles.varDATA,'String','%')
169
170 end
171 handles.pos=pos;
172 handles.id=id;
173 guidata(hObject,handles);
174 end
175 % hObject    handle to Events_Door (see GCBO)
176 % eventdata  reserved - to be defined in a future version of MATLAB
177 % handles     structure with handles and user data (see GUIDATA)
178
179 % Hints: contents = cellstr(get(hObject,'String')) returns Events_Door contents as
180 % cell array
181 % contents{get(hObject,'Value')} returns selected item from Events_Door
182
183 % --- Executes during object creation, after setting all properties.
184 function Events_Door_CreateFcn(hObject, eventdata, handles)
185
186 % hObject    handle to Events_Door (see GCBO)
187 % eventdata  reserved - to be defined in a future version of MATLAB
188 % handles     empty - handles not created until after all CreateFcns called
189
190 % Hint: popupmenu controls usually have a white background on Windows.
191 % See ISPC and COMPUTER.
192 if ispc && isequal(get(hObject,'BackgroundColor'),
193     get(0,'defaultUicontrolBackgroundColor'))
194     set(hObject,'BackgroundColor','white');
195 end
196
197 % --- Executes on selection change in Events_Window.
198 function Events_Window_Callback(hObject, eventdata, handles)
199
200 try
201     strf = get(hObject,'String');
202     valf = get(hObject,'Value');
203
204     id = '4102ecb6';
205
206 switch strf{valf};
207
208     case 'Temperature_Window'
209         pos = '1';
210         set(handles.varDATA,'String','°C')
211     case 'Luminosity_Window(%)'
212         pos = '2';
213         set(handles.varDATA,'String','%')

```

```

210     case 'Window_State1'
211         pos = '3';
212         set(handles.varDATA,'String',' ')
213     case 'Window_State2'
214         pos = '4';
215         set(handles.varDATA,'String',' ')
216     case 'Window_State3'
217         pos = '5';
218         set(handles.varDATA,'String',' ')
219     case 'Window_State4'
220         pos = '6';
221         set(handles.varDATA,'String',' ')
222     case 'Movements_Window'
223         pos = '7';
224         set(handles.varDATA,'String','mov')
225
226 end
227 handles.pos=pos; % extreu la variable pos per fer utilitzada en
228 handles.id=id;
229 guidata(hObject,handles);
230 end
231 % hObject     handle to Events_Window (see GCBO)
232 % eventdata   reserved - to be defined in a future version of MATLAB
233 % handles     structure with handles and user data (see GUIDATA)
234
235 % Hints: contents = cellstr(get(hObject,'String')) returns Events_Window contents as
236 %         cell array
237 %         contents{get(hObject,'Value')} returns selected item from Events_Window
238
239 % --- Executes during object creation, after setting all properties.
240 function Events_Window_CreateFcn(hObject, eventdata, handles)
241 % hObject     handle to Events_Window (see GCBO)
242 % eventdata   reserved - to be defined in a future version of MATLAB
243 % handles     empty - handles not created until after all CreateFcns called
244
245 % Hint: popupmenu controls usually have a white background on Windows.
246 %       See ISPC and COMPUTER.
247 if ispc && isequal(get(hObject,'BackgroundColor'),
248 get(0,'defaultUicontrolBackgroundColor'))
249     set(hObject,'BackgroundColor','white');
250 end
251
252 % --- Executes on selection change in Smart_Cities.
253 function Smart_Cities_Callback(hObject, eventdata, handles)
254
255 % hObject     handle to Smart_Cities (see GCBO)
256 % eventdata   reserved - to be defined in a future version of MATLAB
257 % handles     structure with handles and user data (see GUIDATA)try
258 strf = get(hObject,'String');
259 valf = get(hObject,'Value');
260
261 id = '4102ecc0';
262
263 switch strf{valf};
264
265     case 'Temperature_SC'
266         pos = '1';
267         set(handles.varDATA,'String','°C')
268     case 'Luminosity_SC(%)'
269         pos = '2';
270         set(handles.varDATA,'String','%')
271     case 'Noise_avg'
272         pos = '3';
273         set(handles.varDATA,'String','dBA')
274     case 'Noise_max'
275         pos = '4';
276         set(handles.varDATA,'String','dBA')
277     case 'Luxometer'
278         pos = '5';
279         set(handles.varDATA,'String','Lux')
280 end

```

```

281 handles.pos=pos;
282 handles.id=id;
283 guidata(hObject,handles);
284
285
286 % Hints: contents = cellstr(get(hObject,'String')) returns Smart_Cities contents as
cell array
287 %         contents{get(hObject,'Value')} returns selected item from Smart_Cities
288
289
290 % --- Executes during object creation, after setting all properties.
291 function Smart_Cities_CreateFcn(hObject, eventdata, handles)
292
293 % hObject    handle to Smart_Cities (see GCBO)
294 % eventdata  reserved - to be defined in a future version of MATLAB
295 % handles    empty - handles not created until after all CreateFcns called
296
297 % Hint: popupmenu controls usually have a white background on Windows.
298 %         See ISPC and COMPUTER.
299 if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
300     set(hObject,'BackgroundColor','white');
301 end
302
303
304 % --- Executes on selection change in Gases.
305 function Gases_Callback(hObject, eventdata, handles)
306 % hObject    handle to Gases (see GCBO)
307 % eventdata  reserved - to be defined in a future version of MATLAB
308 % handles    structure with handles and user data (see GUIDATA)try
309 strf = get(hObject,'String');
310 valf = get(hObject,'Value');
311
312 id = '40fc719a';
313
314 switch strf{valf};
315
316     case 'Temperature_Gases'
317         pos = '1';
318         set(handles.varDATA,'String','°C')
319     case 'Pressure'
320         pos = '2';
321         set(handles.varDATA,'String','Kpa')
322     case 'Outdoor_temperature'
323         pos = '3';
324         set(handles.varDATA,'String','°C')
325     case 'Voltage_CO2'
326         pos = '4';
327         set(handles.varDATA,'String','V')
328     case 'ppm_CO2'
329         pos = '5';
330         set(handles.varDATA,'String','ppm')
331 end
332 handles.pos=pos;
333 handles.id=id;
334 guidata(hObject,handles);
335
336 % Hints: contents = cellstr(get(hObject,'String')) returns Gases contents as cell
array
337 %         contents{get(hObject,'Value')} returns selected item from Gases
338
339
340 % --- Executes during object creation, after setting all properties.
341 function Gases_CreateFcn(hObject, eventdata, handles)
342 % hObject    handle to Gases (see GCBO)
343 % eventdata  reserved - to be defined in a future version of MATLAB
344 % handles    empty - handles not created until after all CreateFcns called
345
346 % Hint: popupmenu controls usually have a white background on Windows.
347 %         See ISPC and COMPUTER.
348 if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
349     set(hObject,'BackgroundColor','white');

```

```

350 end
351
352
353 % --- Executes on selection change in People.
354 function People_Callback(hObject, eventdata, handles)
355 % hObject      handle to People (see GCBO)
356 % eventdata    reserved - to be defined in a future version of MATLAB
357 % handles      structure with handles and user data (see GUIDATA)try
358 strf = get(hObject,'String');
359 valf = get(hObject,'Value');
360
361 id = '40f6993b';
362
363 switch strf{valf};
364
365     case 'People_avg'
366         pos = '1';
367         set(handles.varDATA,'String','n°pax')
368     case 'People_instant'
369         pos = '2';
370         set(handles.varDATA,'String','n°pax')
371 end
372 handles.pos=pos;
373 handles.id=id;
374 guidata(hObject,handles);
375
376 % Hints: contents = cellstr(get(hObject,'String')) returns People contents as cell
377 %          array
378 %          contents{get(hObject,'Value')} returns selected item from People
379
380 % --- Executes during object creation, after setting all properties.
381 function People_CreateFcn(hObject, eventdata, handles)
382 % hObject      handle to People (see GCBO)
383 % eventdata    reserved - to be defined in a future version of MATLAB
384 % handles      empty - handles not created until after all CreateFcns called
385
386 % Hint: popupmenu controls usually have a white background on Windows.
387 %       See ISPC and COMPUTER.
388 if ispc && isequal(get(hObject,'BackgroundColor'),
389 get(0,'defaultUicontrolBackgroundColor'))
390     set(hObject,'BackgroundColor','white');
391 end
392
393 % --- Executes on selection change in Power.
394 function Power_Callback(hObject, eventdata, handles)
395 % hObject      handle to Power (see GCBO)
396 % eventdata    reserved - to be defined in a future version of MATLAB
397 % handles      structure with handles and user data (see GUIDATA)
398 strf = get(hObject,'String');
399 valf = get(hObject,'Value');
400
401 id = '409eaae2';
402
403 switch strf{valf};
404
405     case 'Power_avg'
406         pos = '1';
407         set(handles.varDATA,'String','Wh')
408
409 end
410 handles.pos=pos;
411 handles.id=id;
412 guidata(hObject,handles);
413
414 % Hints: contents = cellstr(get(hObject,'String')) returns Power contents as cell
415 %          array
416 %          contents{get(hObject,'Value')} returns selected item from Power
417
418 % --- Executes during object creation, after setting all properties.
419 function Power_CreateFcn(hObject, eventdata, handles)

```

```

420 % hObject      handle to Power (see GCBO)
421 % eventdata    reserved - to be defined in a future version of MATLAB
422 % handles      empty - handles not created until after all CreateFcns called
423
424 % Hint: popupmenu controls usually have a white background on Windows.
425 %      See ISPC and COMPUTER.
426 if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
427     set(hObject,'BackgroundColor','white');
428 end
429
430
431 % --- Executes on button press in Clear.
432 function Clear_Callback(hObject, eventdata, handles)
433 % hObject      handle to Clear (see GCBO)
434 % eventdata    reserved - to be defined in a future version of MATLAB
435 % handles      structure with handles and user data (see GUIDATA)
436
437
438 % --- Executes on button press in Import.
439 function Import_Callback(hObject, eventdata, handles)
440 % hObject      handle to Import (see GCBO)
441 % eventdata    reserved - to be defined in a future version of MATLAB
442 % handles      structure with handles and user data (see GUIDATA)
443 nom = get(handles.edit3,'string');
444 if isempty(nom)
445     errordlg('Enter a variable name !','Error');
446 end
447 DADES=handles.DADES;
448 assignin('base',nom,DADES);
449
450
451 % quan s'ha tirat el waspmote i la dada a continuació polsem OK sempre i
452 % quan la data hagi estat introduïda
453 function OK_Callback(hObject, eventdata, handles)
454 % hObject      handle to OK (see GCBO)
455 % eventdata    reserved - to be defined in a future version of MATLAB
456 % handles      structure with handles and user data (see GUIDATA)
457 try
458
459     query='use MeshliumEXIT;';
460     executeme = handles.ConnD.prepareStatement(query, 10001);
461     resu = executeme.execute();
462     executeme.close();
463     %nom = get(handles.edit3,'string');
464     %if isempty(nom)
465     %    errordlg('Enter a variable name !','Error');
466     %end
467     id=handles.id;
468     pos=handles.pos;
469     % aquesta comanda és la més important i la que s'executata en funció
470     % dels desplegable que hagim triat. "query"
471     query = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",%s),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "%s" AND timestamp < "%s" AND timestamp
>"%s";',pos,id,get(handles.edit2,'string'),get(handles.edit1,'string'));
472     if isempty(query)
473         errordlg('Query is empty! write it first!','Error');
474     else
475         %executem la comanda
476
477         %creem una base de dades
478         executeme = handles.ConnD.prepareStatement(query, 10001);
479         resu = executeme.execute();
480         executeme.close();
481         if resu == 1
482             executeme = handles.ConnD.prepareStatement(query, 10001);
483             resu2 = executeme.executeQuery();
484             vari = {};
485             if resu2.next == 1
486                 %primera fila la llegim a saco
487
488                 i=1;

```

```

489         while(true)
490             try
491                 vari ={vari{:}, resu2.getObject(i)};
492                 if isjava(resu2.getObject(i))
493                     switch resu2.getObject(i).class
494                         case 'java.sql.Timestamp'
495                             vari{end} = char(resu2.getObject(i).toString);
496
497                         otherwise
498                             vari{end} = resu2.getObject(i).doubleValue;
499
500                     end
501                 end
502                 i=i+1;
503             catch
504                 break;
505             end
506         end
507         j=2;
508         while resu2.next == 1% mou a la seguent columna
509             for i=1:size(vari,2)
510                 if isjava(resu2.getObject(i))
511
512                     switch resu2.getObject(i).class
513                         case 'java.sql.Timestamp'
514                             vari{j,i} = char(resu2.getObject(i).toString);
515
516                         otherwise
517                             vari{j,i} = resu2.getObject(i).doubleValue;
518
519                     end
520
521                 else
522                     vari{j,i}=resu2.getObject(i);
523                 end
524             end
525             j=j+1;
526         end
527     end
528     % assignin('base',nom,vari);
529     handles.vari=vari;
530     guidata(hObject,handles);
531     executeme.close();
532 end
533 end
534
535 catch err
536     % Error al executar comanda SQL
537     errordlg(err.message,'Error');
538 end
539 % la comanda executada és guarda en en "vari" i a posterior la trenquem
540 % perquè aquesta pugui ser visualitzada.
541 set(findall(handles.ADDvar, '-property', 'enable'), 'enable', 'on');
542 time=vari(:,2);
543 Val_Sel=(vari(:,1));
544 handles.time=time;
545 handles.Val_Sel= Val_Sel;
546 guidata(hObject,handles);
547 plot(handles.axes2,datetime(vari(:,2),'yy-mm-dd HH:MM:SS.FFF'),str2double(vari(:,1)));
548 datetick(handles.axes2,'x','yy-mm-dd HH:MM');
549 set(handles.axes2,'XTickLabelRotation',45);
550
551
552
553 % Aquesta funció activa i desactiva cada submenú, depenen del waspmote que triem els
554 % altres no serà possible editar-lo, així no podrem seleccionar un waspmote
555 % amb una dada que no li correpon o li correspon a un altre waspmote.
556 %
557 function popupmenu1_Callback(hObject, eventdata, handles)
558 str = get(hObject,'String');
559 val = get(hObject,'Value');
560 switch str{val};
561

```



```

562     case 'EVENTS_DOOR'
563         set(findall(handles.Events_Door, '-property', 'enable'), 'enable', 'on');
564         set(findall(handles.Events_Window, '-property', 'enable'), 'enable', 'off');
565         set(findall(handles.Smart_Cities, '-property', 'enable'), 'enable', 'off');
566         set(findall(handles.Gases, '-property', 'enable'), 'enable', 'off');
567         set(findall(handles.People, '-property', 'enable'), 'enable', 'off');
568         set(findall(handles.Power, '-property', 'enable'), 'enable', 'off');
569     case 'EVENTS_WINDOW'
570         set(findall(handles.Events_Window, '-property', 'enable'), 'enable', 'on');
571         set(findall(handles.Events_Door, '-property', 'enable'), 'enable', 'off');
572         set(findall(handles.Smart_Cities, '-property', 'enable'), 'enable', 'off');
573         set(findall(handles.Gases, '-property', 'enable'), 'enable', 'off');
574         set(findall(handles.People, '-property', 'enable'), 'enable', 'off');
575         set(findall(handles.Power, '-property', 'enable'), 'enable', 'off');
576
577     case 'SMART_CITIES'
578         set(findall(handles.Smart_Cities, '-property', 'enable'), 'enable', 'on');
579         set(findall(handles.Events_Door, '-property', 'enable'), 'enable', 'off');
580         set(findall(handles.Events_Window, '-property', 'enable'), 'enable', 'off');
581         set(findall(handles.Gases, '-property', 'enable'), 'enable', 'off');
582         set(findall(handles.People, '-property', 'enable'), 'enable', 'off');
583         set(findall(handles.Power, '-property', 'enable'), 'enable', 'off');
584     case 'GASES'
585         set(findall(handles.Gases, '-property', 'enable'), 'enable', 'on');
586         set(findall(handles.Events_Door, '-property', 'enable'), 'enable', 'off');
587         set(findall(handles.Events_Window, '-property', 'enable'), 'enable', 'off');
588         set(findall(handles.Smart_Cities, '-property', 'enable'), 'enable', 'off');
589         set(findall(handles.People, '-property', 'enable'), 'enable', 'off');
590         set(findall(handles.Power, '-property', 'enable'), 'enable', 'off');
591
592     case 'PEOPLE'
593         set(findall(handles.People, '-property', 'enable'), 'enable', 'on');
594         set(findall(handles.Events_Door, '-property', 'enable'), 'enable', 'off');
595         set(findall(handles.Events_Window, '-property', 'enable'), 'enable', 'off');
596         set(findall(handles.Smart_Cities, '-property', 'enable'), 'enable', 'off');
597         set(findall(handles.Gases, '-property', 'enable'), 'enable', 'off');
598         set(findall(handles.Power, '-property', 'enable'), 'enable', 'off');
599     case 'POWER_CONSUMPTION'
600         set(findall(handles.Power, '-property', 'enable'), 'enable', 'on');
601         set(findall(handles.Events_Door, '-property', 'enable'), 'enable', 'off');
602         set(findall(handles.Events_Window, '-property', 'enable'), 'enable', 'off');
603         set(findall(handles.Smart_Cities, '-property', 'enable'), 'enable', 'off');
604         set(findall(handles.Gases, '-property', 'enable'), 'enable', 'off');
605         set(findall(handles.People, '-property', 'enable'), 'enable', 'off');
606 end
607 % hObject      handle to popupmenu1 (see GCBO)
608 % eventdata    reserved - to be defined in a future version of MATLAB
609 % handles      structure with handles and user data (see GUIDATA)
610
611 % Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as
cell array
612 %      contents{get(hObject,'Value')} returns selected item from popupmenu1
613
614
615 % --- Executes during object creation, after setting all properties.
616 function popupmenu1_CreateFcn(hObject, eventdata, handles)
617 % hObject      handle to popupmenu1 (see GCBO)
618 % eventdata    reserved - to be defined in a future version of MATLAB
619 % handles      empty - handles not created until after all CreateFcns called
620
621 % Hint: popupmenu controls usually have a white background on Windows.
622 %      See ISPC and COMPUTER.
623 if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
624     set(hObject,'BackgroundColor','white');
625 end
626
627
628
629 function edit1_Callback(hObject, eventdata, handles)
630 % hObject      handle to edit1 (see GCBO)
631 % eventdata    reserved - to be defined in a future version of MATLAB
632 % handles      structure with handles and user data (see GUIDATA)

```

```

633
634 % Hints: get(hObject,'String') returns contents of edit1 as text
635 %         str2double(get(hObject,'String')) returns contents of edit1 as a double
636
637
638 % --- Executes during object creation, after setting all properties.
639 function edit1_CreateFcn(hObject, eventdata, handles)
640 % hObject      handle to edit1 (see GCBO)
641 % eventdata    reserved - to be defined in a future version of MATLAB
642 % handles      empty - handles not created until after all CreateFcns called
643
644 % Hint: edit controls usually have a white background on Windows.
645 %         See ISPC and COMPUTER.
646 if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
647     set(hObject,'BackgroundColor','white');
648 end
649
650
651
652 function edit2_Callback(hObject, eventdata, handles)
653 % hObject      handle to edit2 (see GCBO)
654 % eventdata    reserved - to be defined in a future version of MATLAB
655 % handles      structure with handles and user data (see GUIDATA)
656
657 % Hints: get(hObject,'String') returns contents of edit2 as text
658 %         str2double(get(hObject,'String')) returns contents of edit2 as a double
659
660
661 % --- Executes during object creation, after setting all properties.
662 function edit2_CreateFcn(hObject, eventdata, handles)
663 % hObject      handle to edit2 (see GCBO)
664 % eventdata    reserved - to be defined in a future version of MATLAB
665 % handles      empty - handles not created until after all CreateFcns called
666
667 % Hint: edit controls usually have a white background on Windows.
668 %         See ISPC and COMPUTER.
669 if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
670     set(hObject,'BackgroundColor','white');
671 end
672
673
674
675 function edit3_Callback(hObject, eventdata, handles)
676 % hObject      handle to edit3 (see GCBO)
677 % eventdata    reserved - to be defined in a future version of MATLAB
678 % handles      structure with handles and user data (see GUIDATA)
679
680 % Hints: get(hObject,'String') returns contents of edit3 as text
681 %         str2double(get(hObject,'String')) returns contents of edit3 as a double
682
683
684 % --- Executes during object creation, after setting all properties.
685 function edit3_CreateFcn(hObject, eventdata, handles)
686 % hObject      handle to edit3 (see GCBO)
687 % eventdata    reserved - to be defined in a future version of MATLAB
688 % handles      empty - handles not created until after all CreateFcns called
689
690 % Hint: edit controls usually have a white background on Windows.
691 %         See ISPC and COMPUTER.
692 if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
693     set(hObject,'BackgroundColor','white');
694 end
695
696
697 % --- Executes during object creation, after setting all properties.
698 function axes2_CreateFcn(hObject, eventdata, handles)
699 % hObject      handle to axes2 (see GCBO)
700 % eventdata    reserved - to be defined in a future version of MATLAB
701 % handles      empty - handles not created until after all CreateFcns called
702

```

```

703 % Hint: place code in OpeningFcn to populate axes2
704
705
706 % --- Executes on button press in ImportCSV.
707 function ImportCSV_Callback(hObject, eventdata, handles)
708 % hObject    handle to ImportCSV (see GCBO)
709 % eventdata  reserved - to be defined in a future version of MATLAB
710 % handles    structure with handles and user data (see GUIDATA)
711
712
713 %Aquesta funció escriu en el "box" les dades que hem graficat i ens
714 %interessen exportar-les a matlab.
715 function ADDvar_Callback(hObject, eventdata, handles)
716 % hObject    handle to ADDvar (see GCBO)
717 % eventdata  reserved - to be defined in a future version of MATLAB
718 % handles    structure with handles and user data (see GUIDATA)
719 id=handles.id;
720
721 switch id
722
723     case '4102eccc'
724
725         val_box = get(handles.Events_Door,'String');
726         val_box2 = get(handles.Events_Door,'Value');
727     case '4102ecb6'
728
729         val_box = get(handles.Events_Window,'String');
730         val_box2 = get(handles.Events_Window,'Value');
731     case '4102ecc0'
732
733         val_box = get(handles.Smart_Cities,'String');
734         val_box2 = get(handles.Smart_Cities,'Value');
735     case '40fc719a'
736
737         val_box = get(handles.Gases,'String');
738         val_box2 = get(handles.Gases,'Value');
739     case '40f6993b'
740
741         val_box = get(handles.People,'String');
742         val_box2 = get(handles.People,'Value');
743     case '409eaae2'
744
745         val_box = get(handles.Events_Power,'String');
746         val_box2 = get(handles.Events_Power,'Value');
747
748 end
749 a=get(handles.listbox1,'String');
750 if ischar(a)
751 a={a};
752 end
753 set(handles.listbox1,'String',{a{:} val_box{val_box2}});
754
755 Val_Sel=handles.Val_Sel;
756 time=handles.time;
757 DADES = [time Val_Sel];
758 handles.DADES=DADES;
759 guidata(hObject,handles);
760 set(findall(handles.Import, '-property', 'enable'), 'enable', 'on');
761 set(findall(handles.ImportCSV, '-property', 'enable'), 'enable', 'on');
762
763
764
765 % --- Executes on selection change in listbox1.
766 function listbox1_Callback(hObject, eventdata, handles)
767 % hObject    handle to listbox1 (see GCBO)
768 % eventdata  reserved - to be defined in a future version of MATLAB
769 % handles    structure with handles and user data (see GUIDATA)
770
771 % Hints: contents = cellstr(get(hObject,'String')) returns listbox1 contents as cell
772 %        array
773 %        contents{get(hObject,'Value')} returns selected item from listbox1
774

```

```

775 % --- Executes during object creation, after setting all properties.
776 function listbox1_CreateFcn(hObject, eventdata, handles)
777 % hObject    handle to listbox1 (see GCBO)
778 % eventdata  reserved - to be defined in a future version of MATLAB
779 % handles    empty - handles not created until after all CreateFcns called
780
781 % Hint: listbox controls usually have a white background on Windows.
782 % See ISPC and COMPUTER.
783 if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
784     set(hObject,'BackgroundColor','white');
785 end
786
787
788 % Quan és tenca l'aplicació tanquem la connexió amb la BBDD
789 function figure1_CloseRequestFcn(hObject, eventdata, handles)
790
791 try
792     a=handles.a;
793     a.destroy();
794 end
795
796 % hObject    handle to figure1 (see GCBO)
797 % eventdata  reserved - to be defined in a future version of MATLAB
798 % handles    structure with handles and user data (see GUIDATA)
799
800 % Hint: delete(hObject) closes the figure
801 delete(hObject);
802
803
804 % --- Executes on button press in pushbutton7.
805 function pushbutton7_Callback(hObject, eventdata, handles)
806 % hObject    handle to pushbutton7 (see GCBO)
807 % eventdata  reserved - to be defined in a future version of MATLAB
808 % handles    structure with handles and user data (see GUIDATA)
809
810
811 % --- Executes during object creation, after setting all properties.
812 function ADDvar_CreateFcn(hObject, eventdata, handles)
813 % hObject    handle to ADDvar (see GCBO)
814 % eventdata  reserved - to be defined in a future version of MATLAB
815 % handles    empty - handles not created until after all CreateFcns called
816
817
818 % --- Executes on button press in CLEAR.
819 function CLEAR_Callback(hObject, eventdata, handles)
820 % hObject    handle to CLEAR (see GCBO)
821 % eventdata  reserved - to be defined in a future version of MATLAB
822 % handles    structure with handles and user data (see GUIDATA)
823 set(handles.listbox1,'String',[]);
824
825
826 % Aquesta funció permet descarregar totes les dades al mateix temps a
827 % matlab crear la connexió i a continuació executa cada linea i les
828 % descarrega.
829 function ALLDATA_Callback(hObject, eventdata, handles)
830 % hObject    handle to ALLDATA (see GCBO)
831 % eventdata  reserved - to be defined in a future version of MATLAB
832 % handles    structure with handles and user data (see GUIDATA)
833 try
834     cod=1;
835     query='use MeshliumEXIT;';
836     executeme = handles.ConnD.prepareStatement(query, 10001);
837     resu = executeme.execute();
838     executeme.close();
839     %nom = get(handles.edit3,'string');
840     %if isempty(nom)
841     %     errordlg('Enter a variable name !','Error');
842     %end
843
844 while(cod<27)
845
846     switch cod

```

```

847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887

```

```

case 1
    query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",1),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102eccc" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
    TemperatureDoor = MeshQuery(query2);
    TemperatureDoor = ['TemperatureDoor (°C)'; TemperatureDoor];

case 2
    query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",2),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102eccc" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
    LuminosityDoor = MeshQuery(query2);
    LuminosityDoor = ['LuminosityDoor(%)'; LuminosityDoor];

case 3
    query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",3),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102eccc" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
    Vibrations = MeshQuery(query2);
    Vibrations= ['Vibrations (mV)';Vibrations];

case 4
    query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",4),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102eccc" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
    DoorState=MeshQuery(query2);
    DoorState=['DoorState';DoorState];

case 5
    query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",5),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102eccc" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
    MovementsDoor=MeshQuery(query2);
    MovementsDoor = ['MovementsDoor';MovementsDoor];

case 6
    query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",6),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102eccc" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
    Humidity=MeshQuery(query2);
    Humidity =['Humidity(%)';Humidity];

case 7
    query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",1),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102ecb6" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
    TemperatureWindow=MeshQuery(query2);
    TemperatureWindow=['TemperatureWindow (°C)';TemperatureWindow];

case 8
    query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",2),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102ecb6" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
    LuminosityWindow=MeshQuery(query2);
    LuminosityWindow=['LuminosityWindow(%)';LuminosityWindow];

```

```

888     case 9
889         query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",3),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102ecb6" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
890         WindowState1=MeshQuery(query2);
891         WindowState1=['WindowState1';WindowState1];
892
893     case 10
894         query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",4),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102ecb6" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
895         WindowState2=MeshQuery(query2);
896         WindowState2=['WindowState1';WindowState2];
897
898     case 11
899         query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",5),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102ecb6" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
900         WindowState3=MeshQuery(query2);
901         WindowState3=['WindowState1';WindowState3];
902     case 12
903         query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",6),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102ecb6" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
904         WindowState4=MeshQuery(query2);
905         WindowState4=['WindowState1';WindowState4];
906     case 13
907         query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",7),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102ecb6" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
908         MovementsWindow=MeshQuery(query2);
909         MovementsWindow=['MovementsWindow';MovementsWindow];
910     case 14
911         query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",1),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102ecc0" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
912         TemperatureSC=MeshQuery(query2);
913         TemperatureSC=['TemperatureSC(°C)';TemperatureSC];
914     case 15
915         query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",2),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102ecc0" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
916         LuminositySC=MeshQuery(query2);
917         LuminositySC=['LuminositySC(%)';LuminositySC];
918     case 16
919         query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",3),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102ecc0" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
920         NoiseAVG=MeshQuery(query2);
921         NoiseAVG=['NoiseAVG(dBA)';NoiseAVG];
922     case 17
923         query = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",4),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102ecc0" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
924         NoiseMax=MeshQuery(query2);

```



```

925 NoiseMax=['NoiseMax(dBA) ' ;NoiseMax];
926
927 case 18
928 query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",5),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "4102ecc0" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
929 Lux=MeshQuery(query2);
930 Lux=['Lux';Lux];
931 case 19
932 query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",1),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "40fc719a" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
933 TemperatureGases=MeshQuery(query2);
934 TemperatureGases=['TemperatureGases(°C) ' ;TemperatureGases];
935 case 20
936 query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",2),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "40fc719a" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
937 Pressure=MeshQuery(query2);
938 Pressure=['Pressure(kPa) ' ;Pressure];
939 case 21
940 query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",3),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "40fc719a" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
941 OutDoorTemperature=MeshQuery(query2);
942 OutDoorTemperature=['OutDoorTemperature(°C) ' ;OutDoorTemperature];
943 case 22
944 query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",5),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "40fc719a" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
945 CO2=MeshQuery(query2);
946 CO2=['CO2(ppm) ' ;CO2];
947 case 23
948 query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",1),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "40f6993b" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
949 PeopleAVG=MeshQuery(query2);
950 PeopleAVG=['PeopleAVG';PeopleAVG];
951 assignin('base','Allw',PeopleAVG);
952 case 24
953 query2 = sprintf('SELECT
SUBSTRING_INDEX(SUBSTRING_INDEX(value,";",1),"",-1),timestamp FROM
SensorsMesh WHERE id_wasp = "409eaae2" AND timestamp < "%s" AND
timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
954 Consumption=MeshQuery(query2);
955 Consumption=['Consumition(Wh) ' ;Consumition];
956 case 25
957 query2 = sprintf('SELECT timestamp FROM SensorsMesh WHERE
id_wasp="4102ecc0" AND timestamp < "%s" AND timestamp
>"%s";',get(handles.edit2,'string'),get(handles.edit1,'string'));
958 Time=MeshQuery(query2);
959 Time=['Timestamp';Time];
960 end
961
962 cod=cod+1;
963 end
964 % és crea una sola matriu amb totes les dades Això esta en fase de proves
965 AllDatas=[TemperatureDoor LuminosityDoor Vibrations DoorState MovementsDoor
Humidity TemperatureWindow LuminosityWindow WindowState1 WindowState2 WindowState3
WindowState4 MovementsWindow TemperatureSC LuminositySC NoiseAVG NoiseMax Lux

```

```
966 TemperatureGases Pressure OutDoorTemperature CO2 PeopleAVG Consumption Time];  
967 assignin('base','All',AllDatas);  
968     end
```