ภาคผนวก ข

ตัวอย่างโปรแกรมใน m-file

ตัวอย่างโปรแกรมใน m-file

```
function varargout = files(varargin)
% FILES MATLAB code for files.fig
%      FILES, by itself, creates a new FILES or raises the existing
%      singleton*.
%
%      H = FILES returns the handle to a new FILES or the handle to
%      the existing singleton*.
%
%      FILES('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in FILES.M with the given input arguments.
%
%      FILES('Property','Value',...) creates a new FILES or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before files_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to files_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help files

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @files_OpeningFcn, ...
                   'gui_OutputFcn',  @files_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before files is made visible.
function files_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to files (see VARARGIN)
% Choose default command line output for files
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes files wait for user response (see UIRESUME)
% uiwait(handles.figure1);

assignin('base', 'stop', 0);
assignin('base', 'reset', 0);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = files_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

clc; % clear screen

% open dialog
[filename, pathname] = uigetfile('*.wmv', 'Select a WMV file');
if isequal(filename,0)
  disp('User selected Cancel'); % not select
else

  assignin('base', 'log_start', datestr(now,'HH:MM:SS'));

  assignin('base', 'stop', 0); % assign value no stop
  file = fullfile(pathname, filename); % file location

     clc;
     vidObj = VideoReader(file); % read video file
     lastFrame = read(vidObj, inf); % get last frame
     nFrames = vidObj.NumberOfFrames; % number of frame video
```

```
c = 0;
s = 0;
axes(handles.axes1); % set current axes
hold on;
for k = 1 : nFrames % for loop frame video

set(handles.text2, 'String', datestr(now,'HH:MM:SS')); % show clock

  stop = evalin('base', 'stop'); % get stop status

  if stop == 0 % if not press stop button

    rgb = read(vidObj,k); % get frame image
    I = rgb2gray(rgb); % convert to grayscale

      % cut image in selected location leg chicken
      A1 = imcrop(I, [437, 234, 10, 20]);
      A2 = imcrop(I, [437, 235, 10, 20]);
      A3 = imcrop(I, [436, 237, 10, 20]);
      A4 = imcrop(I, [438, 237, 10, 20]);
      A5 = imcrop(I, [435, 241, 10, 20]);
      A6 = imcrop(I, [439, 241, 10, 20]);
      A7 = imcrop(I, [435, 245, 10, 20]);
      A8 = imcrop(I, [439, 245, 10, 20]);
      A9 = imcrop(I, [435, 249, 10, 20]);
      A10 = imcrop(I, [439, 249, 10, 20]);

      B1 = imcrop(I, [485, 234, 10, 20]);
      B2 = imcrop(I, [485, 235, 10, 20]);
      B3 = imcrop(I, [486, 237, 10, 20]);
      B4 = imcrop(I, [484, 237, 10, 20]);
      B5 = imcrop(I, [487, 241, 10, 20]);
```

```
B6 = imcrop(I, [483, 241, 10, 20]);
B7 = imcrop(I, [487, 245, 10, 20]);
B8 = imcrop(I, [483, 245, 10, 20]);
B9 = imcrop(I, [487, 249, 10, 20]);
B10 = imcrop(I, [483, 249, 10, 20]);

% filter intensity over 70 only
X1 = find(A1 > 70);
X2 = find(A2 > 70);
X3 = find(A3 > 70);
X4 = find(A4 > 70);
X5 = find(A5 > 70);
X6 = find(A6 > 70);
X7 = find(A7 > 70);
X8 = find(A8 > 70);
X9 = find(A9 > 70);
X10 = find(A10 > 70);

Y1 = find(B1 > 70);
Y2 = find(B2 > 70);
Y3 = find(B3 > 70);
Y4 = find(B4 > 70);
Y5 = find(B5 > 70);
Y6 = find(B6 > 70);
Y7 = find(B7 > 70);
Y8 = find(B8 > 70);
Y9 = find(B9 > 70);
Y10 = find(B10 > 70);

% calculate percentage of index image leg chicken
P1 = (100 * length(X1)) / 256;
P2 = (100 * length(X2)) / 256;
```

```
        P3 = (100 * length(X3)) / 256;
        P4 = (100 * length(X4)) / 256;
        P5 = (100 * length(X5)) / 256;
        P6 = (100 * length(X6)) / 256;
        P7 = (100 * length(X7)) / 256;
        P8 = (100 * length(X8)) / 256;
        P9 = (100 * length(X9)) / 256;
        P10 = (100 * length(X10)) / 256;
        Q1 = (100 * length(Y1)) / 256;
        Q2 = (100 * length(Y2)) / 256;
        Q3 = (100 * length(Y3)) / 256;
        Q4 = (100 * length(Y4)) / 256;
        Q5 = (100 * length(Y5)) / 256;
        Q6 = (100 * length(Y6)) / 256;
        Q7 = (100 * length(Y7)) / 256;
        Q8 = (100 * length(Y8)) / 256;
        Q9 = (100 * length(Y9)) / 256;
        Q10 = (100 * length(Y10)) / 256;

        % check percentage value
        if P1 > 30 && P2 > 30 && P3 > 30 && P4 > 30 && P5 > 30 && P6 > 30 && P7 > 30
&& P8 > 30 && P9 > 30 && P10 > 30 && Q1 > 30 && Q2 > 30 && Q3 > 30 && Q4 > 30 &&
Q5 > 30 && Q6 > 30 && Q7 > 30 && Q8 > 30 && Q9 > 30 && Q10 > 30
            c = c + 1;

            reset = evalin('base', 'reset');

            if reset == 1
               c = 0;
               assignin('base', 'reset', 0);
            end
```

```
              set(handles.text1, 'String', num2str(c)); % show count chicken
           end

      end

      imshow(I); % show frame video on axes

      % plot rectangle leg chicken location
      rectangle('Position',[432, 234, 10, 20], 'EdgeColor', 'w', 'LineWidth',1);
      rectangle('Position',[480, 234, 10, 20], 'EdgeColor', 'w', 'LineWidth',1);

    end

end

 % --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

assignin('base', 'stop', 1);

 % --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

assignin('base', 'reset', 1);

% --- Executes on button press in pushbutton4.
```

```
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

log_end = datestr(now,'HH:MM:SS');
log_start = evalin('base', 'log_start');
count = get(handles.text1, 'String');

fid = fopen('Result.txt','w');
fprintf(fid, ' Start time : %s\n End time : %s\n Count : %c', log_start, log_end, count);
fclose(fid);
```