DR. BOB DAVIDOV

Построение графического интерфейса пользователя (GUI) в среде МатЛАБ для целей управления и наблюдением за состоянием объекта

Цель работы: освоение правил построения графического интерфейса пользователя (Graphical User Interface "GUI") в среде МатЛАБ..

Задача работы: построение интерфейса в среде МатЛАБ для обеспечения связи пользователя с устройством дискретного ввода-вывода МК110-224.8Д.4Р через ОРС сервер.

Приборы и принадлежности: Персональный компьютер, МК110, преобразователь UPort1150 (USB/RS-485), МатЛАБ, ОРС Toolbox, ОРС DA сервер.

ОБЩИЕ СВЕДЕНИЯ

Интерфейс пользователя обеспечивает связь пользователя с программными и аппаратными ресурсами системы. Через интерфейс пользователь вводит заданные значения и принимает параметры системы. Интерфейс должен отображать состояние системы в наиболее удобном для восприятия виде.

Для выполнения работы необходимо убедиться в наличии OPC сервера на рабочем компьютере

InSAT ► MasterOPC Universal Modbus Server ► MasterOPC UniverS



Если раздел невиден (спрятан), то его необходимо высветить командой "Show hidden files and folders":

Folder Options 🛛 🔹 🛛 🖓 🔀
General View File Types Offline Files
Folder views You can apply the view (such as Details or Tiles) that you are using for this folder to all folders. Apply to All Folders Reset All Folders
Advanced settings:
 Files and Folders Automatically search for network folders and printers Display file size information in folder tips Display simple folder view in Explorer's Folders list Display the contents of system folders Display the full path in the address bar Display the full path in the title bar Do not cache thumbnails Hidden files and folders Show hidden files and folders Y Hide extensions for known file types
Restore Defaults
OK Cancel Apply

Убедитесь что драйвер UPort1150 (устройство связи USB порта компьютера с МК110) настроен на работу в режиме RS485.

Примеры интерфейсов, разработанные в среде МатЛАБ представлены на Рис. 1 Рис.2 и Рис.4.



Рис. 1. Интерфейс системы для тестирования магнитного поля объекта.



Рис.2. Магнитоэлектрический датчик с эталонной катушкой интерфейс которого представлен на Рис. 3.

Configuration	HANDSET	Battery
Ambient pressure 1010 mbar		2200 (mAh)
O2 cylinder	T.T.S CEILING TIME CEILING	Av.discharge
02 tank size		SCRB
DIL Coo Mix 1	· · · · · · · · · · · · · · · · · · ·	4.5 Life-Time, (h)
		O2 control
02 40 He 0 N2 60	DEPTH DIVE TIME CNS% DES/ASCENT RATE	Predict Bang 😽
itial DIL pressure 230 bar		O CO2 control
IL tank size 3 litres	0 m · · · · O 0 m/min	- View a Profile
DIL Gas Mix 2		prof_DV_HS 🔽
	02 6 5 % He 0 % N2 3 5 % k 9 9 % 4 9 9 % U 3 0 % k 2 4 ° c	plot
02 32 He 0 N2 68		- Log for Uploading
itial DIL pressure 230 bar	PP02 S4 PPC02 EqND DIL.CYLINDER 02 CYLINDER	denth plot
IL tank size 3 litres		MAT-file name
DIL Gas Mix 3	0.04.00 Z/m Z30÷ Z30	
2 40 He 0 N2 60		save
* al Dil angeneration 230 have		
Ittai DIL pressure 200 par	LEDs Diving with out ob oakell	Log for plotting
	Diving without checks!!!	
Depth Input Sim Mode		Ceil_t plot
O Profile	O mode A/B	Ceil_d plot
O Deco O Accelerate	Manual injectors, I/min FO2 FHe FN2 FCO2 Fres FTotal BL(bar)	O CNS plot
00000	75 Diluent O2 75 64.61 0 35.39 0 0 100 0.99 << >>	O FO2 plot
Idealise the He & N2 sensors		
Idealise the He & N2 sensors Input	Depth 0 Metabolic Rate 10 Respiratory Rate 60 Tidal Volume 20 Gas Mix 1	OFHe plot
o beed) Idealise the He & N2 sensors es/Asc. m/min) o depth	Depth 0 Metabolic Rate 1.0 Respiratory Rate 60 Tidal Volume 2.0 Gas Mix 1 (m) (iitres/min) (iitres/min) 60 Tidal Volume 2.0 Gas Mix 1	OFN2 plot
D Idealise the He & N2 sensors es/Asc. m/min) → 60 → rate	Depth 0 Metabolic Rate 1.0 Respiratory Rate 60 Tidal Volume 2.0 Gas Mix 1 0 + + 6 0 + 100 0.5 + 90 1 •	FN2 plot
Idealise the He & N2 sensors es/Asc. m/min) -15 60 rate	Depth 0 Metabolic Rate 1.0 Respiratory Rate 60 Tidal Volume 2.0 Gas Mix 1 0 + 600 0 + 100 0.5 + 90 1 • pause -	FN2 plot
Idealise the He & N2 sensors Idealise the He & N2 sensors es/Asc. m/min) •	Depth 0 Metabolic Rate 1.0 Respiratory Rate 60 Tidal Volume 2.0 Gas Mix 1 0 + 60 - + 100 0.5 + 90 1 • 0 + 60 - + 100 0.5 + 90 1 • • Pause Pause Reset Office MP 10 0 (bar) 10	FHe plot FN2 plot hum plot temp plot temp plot
Idealise the He & N2 sensors Input Input	Depth 0 Metabolic Rate 1.0 Respiratory Rate 60 Tidal Volume 2.0 Gas Mix 1 0 + + 6 0 + 100 0.5 + 90 1 • Pause Pause Reset O2 reduction Orifice 000 (implication) 10 100 0.5 10 10 0.5 10 10 0.5 10 00	FHe plot FN2 plot hum plot temp plot PPO2 plot
Idealise the He & N2 sensors Idealise the He & N2 sensors Input Input Imput	Depth 0 Metabolic Rate 1.0 Respiratory Rate 60 Tidal Volume 2.0 Gas Mix 1 0 + + 60 + + 100 0.5 + 90 1 • • + 60 • + 100 0.5 + 90 1 • • Run Pause Reset O2 reduction O// (bar) 10 0.5 • Pause O// (bar) 10 0.5 • • 0.0 0.0 IMP 0.0	 PHe plot FN2 plot hum plot temp plot PPO2 plot PPC02 plot

Рис.4.Интерфейс модели глубоководного аппарата.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Задание 1. Разработка графического интерфейса пользователя для связи с устройством дискретного ввода-вывода МК110.

В рамках этой работы необходимо построить интерфейс связи с МК110 который имеет

- восемь окон для отображения состояния счетных входов устройства. В первом окне отображается количество циклов связи с ОРС сервером. В последнем 8-м окне отображается значение счетчика восьмого входа устройства МК110..
- четыре окна для ввода 4-х значений ШИМ в диапазоне от 1 до 1000.
- клавиша Reset для сброса выбранного счетчика.
- клавиша Run/Stop для записи установленного ШИМ или его нулевого значения в 4-й выход устройства.
- клавиши Connect и Disconnect для установки и разрыва связи с ОРС сервером.

После установки связи с ОРС устройством, программа должна каждую секунду отображать состояние восьмого входа устройства МК110. И записывать значение ШИМ в четвертый выход устройства после нажатия клавиши Run.

Пример интерфейса показан на рисунке ниже.



Рис. 5.Интерфейс связи с МК110.

Последовательность выполнения работы:

- 1. Загрузите МатЛАБ. Настройте ее на рабочую папку.
- 2. В командном окне наберите guide и затем нажмите клавишу <Enter>
- В появившемся окне GUI Quick Start выберите закладку Create New GUI > Blank GUI (Default).

- 4. Дайте название проекту (файл fig), например, main_MK110. В рабочей папке дополнительно к этому файлу появится файл main_MK110.m. В этот файл система МатЛАБ автоматически добавляет функции для связи с каждым объектом GUI (кнопки, окна, меню, списки, графики и др.) которые будут перенесены из панели типовых объектов в рабочее пространство GUI.
- 5. Перенесите нужные объекты в поле GUI. С каждым объектом связан набор атрибутов. Наиболее важные из них следующие:
 - Тад поле для ввода имени объекта, это имя автоматически присваивается соответствующей функции m-файла и вводится в поле Callback
 - String надпись в поле объекта, которая отображается при загрузке GUI..



Рис. 6. Порядок составления GUI

- 6. Для построения рабочего варианта GUI и m-файла необходимо нажать на зеленый треугольник меню (см. **Рис. 6**).
- 7. Для установки и запуска таймера можно использовать следующие функции. rt_timer = timer('TimerFcn', 'do_mode_MK110', 'ExecutionMode', 'FixedRate'); rt_timer.Period = 1; rt_timer = timerfind; rt_timer.TasksToExecute = 36000; get(rt_timer, 'Running') start(rt_timer) stop(rt_timer)

```
8. Для установки связи с ОРС, чтения и записи в устройство MK110 используются

cледующие команды.

da = opcda('localhost', 'InSAT.ModbusOPCServer.DA');

connect(da)

grp_in = addgroup(da);

additem(grp_in, 'COM_port.MK110.Counters.Input8');

r_8 = read (grp_in);

grp_out = addgroup(da);

additem(grp_out, 'COM_port.MK110.Outputs.Output4');

write(grp_out, 500)

write(grp_out, 0)

disconnect(da)
```

Пример фрагментов кода основной программы:

function main_MK110_OpeningFcn(hObject, eventdata, handles, varargin)
model_open(handles)
handles.output = hObject;
guidata(hObject, handles);

% ------

function model_open(handles)
global run_1 run_2 run_3 run_4 sim_state GUIparams counter da
run_1 = false;
in_panel_num = 8;
sim_state = 'stop';
counter = 0;

set(handles.uipanel18, 'BackgroundColor', [160 158 158] ./256)
set(handles.input_number, 'String', num2str(in_panel_num))

% Create new timer object

rt_timer = timer('TimerFcn', 'do_mode_MK110', 'ExecutionMode', 'FixedRate'); rt_timer.Period = 1; % Specifies the delay, in seconds, between executions

da = opcda('localhost', 'InSAT.ModbusOPCServer.DA');

function counter1_1_Callback(hObject, eventdata, handles) function counter1_2_Callback(hObject, eventdata, handles) function counter1_3_Callback(hObject, eventdata, handles) function counter1_4_Callback(hObject, eventdata, handles) function counter1_5_Callback(hObject, eventdata, handles)

function counter8_1_Callback(hObject, eventdata, handles) function counter8_2_Callback(hObject, eventdata, handles) function counter8_3_Callback(hObject, eventdata, handles) function counter8_4_Callback(hObject, eventdata, handles) function counter8_5_Callback(hObject, eventdata, handles)

% -----

```
function left button Callback(hObject, eventdata, handles)
NewStrVal = get(handles.input_number, 'String');
in_panel_num = str2num(NewStrVal);
in_panel_num = in_panel_num - 1;
if in_panel_num < 1
  in panel num = 8:
end
switch in_panel_num
  case 1
     set(handles.uipanel11, 'BackgroundColor', [160 158 158] ./256)
    set(handles.uipanel12, 'BackgroundColor', [236 233 216]./255)
  case 8
     set(handles.uipanel18, 'BackgroundColor', [160 158 158] ./256)
     set(handles.uipanel11, 'BackgroundColor', [236 233 216]./255)
end
set(handles.input number, 'String', num2str(in panel num))
% -----
function reset button Callback(hObject, eventdata, handles)
global counter da
NewStrVal = get(handles.input number, 'String');
in_panel_num = str2num(NewStrVal);
switch in panel num
  case 1
     counter = 0:
     set(handles.counter1 1, 'string', '0');
     set(handles.counter1 2, 'string', '0');
     set(handles.counter1_3, 'string', '0');
     set(handles.counter1_4, 'string', '0');
     set(handles.counter1 5, 'string', '0');
  case 8
     grp in = addgroup(da);
     additem(grp_in, 'COM_port.MK110.Counters.Input8');
     if strcmp(da.status,'connected')
       write(grp in, 0) % reset counter 8
     end
end
% -
function output 4 Callback(hObject, eventdata, handles)
  NewStrVal = get(handles.output_4,'String');
  NewVal = str2num(NewStrVal);
  if NewVal > 1000
    NewVal = 1000;
```

```
end
  if NewVal < 0
    NewVal = 0;
  end
  NewStrVal = num2str(NewVal);
  OutStrVal = '0000';
  k = length(NewStrVal);
  for i = 1:k
    OutStrVal (5-i) = NewStrVal (k+1-i);
  end
  set(handles.output_4,'String',OutStrVal)
%_____
function run_button_4_Callback(hObject, eventdata, handles)
global run_4 da gpr_out
grp out = addgroup(da);
additem(grp_out, 'COM_port.MK110.Outputs.Output4');
if~run 4
  set(handles.run_button_4, 'BackgroundColor', [0.835294 0.815686 0.784314])
  %wavplay(param.DING sound.y,param.DING sound.Fs)
  NewStrVal = get(handles.output_4,'String');
  NewVal = str2num(NewStrVal);
  set(handles.run_button_4,'String','Run 4')
  write(grp_out, 0) % stop PWM 4
  run_4 = true;
else
  set(handles.run_button_4, 'BackgroundColor', [0.9608 0.9569 0.9529])
  set(handles.run_button_4,'String','Stop 4')
  NewStrVal = get(handles.output_4,'String');
  NewVal = str2num(NewStrVal);
  write(grp out, NewVal) % start PWM 4
  run_4 = false;
end
% ------
function OPC connect button Callback(hObject, eventdata, handles)
global sim state GUIparams da grp out
if strcmp(sim_state,'stop')
  % RT sim mode
  rt timer = timerfind;
  rt_timer.TasksToExecute = 36000; % Specifies max possible number of clocks
  GUlparams = handles:
  if strcmp(get(rt_timer, 'Running'),'off')
```

```
start(rt timer) % start timer that run do mode MK110
  end
  set(handles.OPC_disconnect_button, 'BackgroundColor', [0.835294 0.815686 0.784314])
  set(handles.OPC_connect_button, 'BackgroundColor', [0.9608 0.9569 0.9529])
  sim state = 'run';
  connect(da);
end
% ----
        -----
function OPC disconnect_button_Callback(hObject, eventdata, handles)
global sim_state da
if strcmp(sim_state,'run')
  % stop RT mode
  rt timer = timerfind;
% if strcmp(get(rt_timer, 'Running'),'on')
    stop(rt_timer);
% end
  set(handles.OPC connect button, 'BackgroundColor', [0.835294 0.815686 0.784314])
  set(handles.OPC_disconnect_button, 'BackgroundColor', [0.9608 0.9569 0.9529])
% wavplay(param.DING sound.y,param.DING sound.Fs)
  sim_state = 'stop';
  disconnect(da)
end
Пример кода подпрограммы вызываемой таймером:
function do mode MK110
global GUIparams counter da
handles = GUIparams;
grp in = addgroup(da);
additem(grp in, 'COM port.MK110.Counters.Input8');
connect(da)
r = read (grp in);
str_counter = num2str (r_8.Value);
k = length(str counter);
for i = 1:k
  switch i
     case 1
       set(handles.counter8 1, 'string', str counter (k));
     case 2
       set(handles.counter8 2, 'string', str counter (k-1));
     case 3
       set(handles.counter8 3, 'string', str counter (k-2));
     case 4
       set(handles.counter8 4, 'string', str counter (k-3));
```

```
case 5
    set(handles.counter8_5, 'string', str_counter (k-4));
end
end
```

```
GUIparams = handles;
```

контрольные вопросы

- 1. Назовите возможные варианты применение графического интерфейса пользователя GUI?
- 2. Каким образом задается требуемая частота связи GUI с устройством МК 100?
- 3. Перечислите основные компоненты канала связи GUI устройство МК.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1. Нер система МатЛАБ
- 2. Инструкция по эксплуатации МК110-224.8Д.4Р
- 3. Инструкция по установки драйвера преобразователя. UPort1150.
- 4. Инструкция по установке ModBUS OPC сервера компании ИнСАТ.