Engineering Geodesy I HS 2010
Exercise 2a: Application & Compiling

Goals:

- extend your own automatic deformation monitoring system
- create a graphical interface for your application
- compile it and create a stand alone application
- perform real time tracking

Tasks:

1) Type “`guide`” in the Matlab command line. Create Blank GUI (default).

2) Click on “Pushbutton” and then use the mouse to create a window for a button.

3) Double click on the newly created button “Push Button” → the property inspector opens up.

4) Change the property “String” from “Push Button” into “Angle Request”. Change also the FontSize to e.g. 12. Close property Inspector

5) Adjust window size with the mouse.

6) Save Figure as “`monitoring.fig`” (this creates also monitoring.m)

7) open monitoring.m and add the name of your application `angle_request.m`, (e.g. http://www.geometh-data.ethz.ch/student/eg1/2010/02_deformation/angle_request.txt) at the end of the file (after “`function pushbutton1_Callback(hObject, eventdata, handles)`”), i.e. type “`angle_request;`”. This line will be carried out each time the push button is pressed.

8) For preventing error messages at start-off, write a try-catch block around line 73:

```
try
varargout{1} = handles.output;
catch
end
```

9) Run monitoring.m

10) Add 2 functions to your application:

a) open_port.m that defines and opens the serial com-port

b) close_port.m for closing the com-port

Note that angle_request.m does not require the handling of the com-port anymore!

Note that the serial object has to be defined as a global variable in all files, type:
`global s;`

11) Add find_prism.m that lets the instrument find the next target.

12) type "`mcc –m monitoring.m`" to evoke the matlab compiler. Option "-m" creates a stand alone application: monitoring.exe, c-files (monitoring_main.c, monitoring_mcc_component_data.c) and the monitoring.mcr file.

13) Start monitoring.exe on any computer (include the mcr-files).

14) The figure properties can be changed during runtime directly in Matlab, e.g. type `set(gco,'String','done');` behind a function call for a callback in monitoring.m.

15) Create a new m-file `small_menu.m` with the following contents and run it:

```
function varargout = small_menu(varargin)

mea=uimenu('Label','&Measure');
ang=uimenu(mea,'Label','Angle Request','Callback','angle_request;');
return;
```

Add the functions: open_port, close_port, find_prism and coordinate request.

16) Download `tracking.zip` from the EG1 Website, extract the files and try to figure out what the Tracking_R.m file does.
Note:
If you use a 360° reflector, set at the instrument the prism settings: 5 Konfig, 2 Inst. Set., 1 EDM, → EDM type: Leica 360°. Alternatively, `send(serPort, '17008:3')` would set it automatically (3 = Leica 360º Prism).
A "+" at the top row indicates that the instrument is in lock-mode, a circle around it shows that the prism is locked.