Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute of Geodesy and Photogrammetry
Geosensors and Engineering Geodesy

September 30, 2015 (MF)

Engineering Geodesy, 2015 HS

GeoCOM Example

1. Setting and structure of the communication

Implementing an application by means of the ASCII protocol is based on simple data transfers using a serial line. The programmer is responsible to set up the serial line parameters of the client such that they correspond to the settings of the TS60 instrument. In Matlab 'serial' command could be used to establish the connection between the PC and the total station. The port created using 'serial' command has to be then opened for communication and closed after performing all measurements. Hint: do not close and reopen the port after each command – it will significantly slow performance of your script.

Example:

```
%% Open TS port
obj = instrfind; if ~isempty(obj); delete(obj); end
TPSport = serial('COM7', 'BaudRate', 115200, 'Parity', 'none', 'DataBits', 8, 'StopBits', 1, 'Ter-
minator', 'CR/LF', 'TimeOut', 60);
fopen (TPSport);

%% Communication with the instrument

% Request - send a command to the total station
req='string';
fprintf(TPSport, req);

% Reply - read response data into the sting-variable out
out = fscanf(TPSport);

% Extract the returned values
data = regexp(out, expression, 'names');

% Request | Reply | Extract
% Request | Reply | Extract
…

% Close TPS port
fclose(TPSport);
```

2. Syntax of the ASCII request and response

Remote calls are done by just sending the valid encoded requests and receiving and decoding the replies of them.

ASCII Request:
[<LF>]%R1Q,<RPC>[,<TrId>]:[<P0>][,<P1>,...]<Term>

<LF>	An initial line feed clears the receiver buffer.
%R1Q	GeoCOM request type 1.
<RPC>	Remote Procedure Call identification number in between 0 to 65535.
<TrID>	Optional transaction ID: normally incremented from 1 to 7. Same value in reply.
:	Seperator between protocol header and following parameters.
<P0>,<P1>,…	Parameter 0, Paramter 1
<Term>	Terminator string (default CR/LF, use COM_SetTerminator to change the terminator).

Syntax – ASCII Reply:
%R1P,<RC_COM>[,<TrId>]:<RC>[,<P0>,<P1>, ...]<Term>

%R1P	GeoCOM reply type 1.
<RC_COM>	GeoCOM return code. This value denotes the success of the communication. GRC_OK = 0 means the communication was successful.
<TrId>	Transaction ID – identical to that of request. If the request had no Transaction ID then it will be 0.
:	Seperator between protocol header and following parameters.
<RC>	Return code from the called PRC and denotes the successful completion if it is set to 0.
<P0>,<P1>,…	Parameter 0, Parameter 1,… These parameters will be valid only if <RC> is 0.
<Term>	Terminator string (default CR/LF, use COM_SetTerminator to change the terminator).

3. Implementation in Matlab

1. Look for the function in the manual (as an example a command to get the station coordinates is used)

15.6.11 TMC_GetStation - getting the station coordinates of the instrument

C-Declaration

```
TMC_GetStation(TMC_STATION &Station)
```

VB-Declaration

```
VB_TMC_GetStation(Station As TMC_STATION)
```

ASCII-Request

```
%R1Q,2009:
```

ASCII-Response

```
%R1P,0,0:RC,E0[double],N0[double],H0[double],Hi[double]
```

Remarks

This function is used to get the station coordinates of the instrument.

Parameters

Station	Out	Instrument station co-ordinates [m].

Return-Code Names and Return-Code Values

GRC_OK	0	Execution successful.

See Also

TMC_SetStation

2. Create request variable in Matlab:

```
%% 15.6.11 TMC_GetStation - getting the station coordinates of the instrument
% %R1Q,2009,<TrId>:
% %R1P,0,0:RC,E0[double],N0[double],H0[double],Hi[double]
req='%R1Q,2009,1:';
```

3. Send the command to the total station:

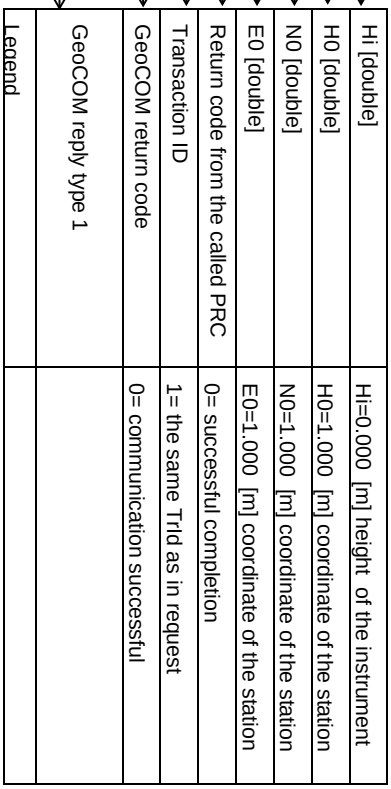
```
fprintf(TPSport, req);
```

4. Read the response from the total station:

```
out = fscanf(TPSport);
```

5. Decode the received string:

out =
‘%R1P,0,1:0,1,1,1,0’




```
% Extract the returned values
data = regexp(out, ':(?<RC>[^,]*),(?<E0>[^,]*),(?<N0>[^,]*),(?<H0>[^,]*)', 'names');

% Check output and store result
if str2num(data.RC) == 0
   Station(1) = str2double(data.E0);
   Station(2) = str2double(data.N0);
   Station(3) = str2double(data.H0);

   fprintf('The coordinates of the TPS are: E0 = %.3f N0 = %.3f H0 = %.3f\n', Sta-
tion(1),Station(2),Station(3));
else
   fprintf('GRC_Return-Code= %.0f; Check the GRC return-code in manual!\n', data.RC)
end
```