Appendix H

User Interface Code Listing

The code listing for the user interface in Borland Delphi Version 6 is:

```
unit Unit;

interface

uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Menus, StdCtrls, ComCtrls, ExtCtrls, ComPort, StrUtils, Math;

const
//Data files to store database fields
//Three databases - sensor information, field
//information and received sensor data
cdata='sensor.dat';
cdata2='infield.dat';
cdata3='farmdata.dat';
TEMP='temp.dat';


//Sensor information database details
type
sensordata=record
id:string[4];
farmno:integer;
sensor:string[15];
xcoordinate:integer;
ycoordinate:integer;
end;


//Received sensor data database details
infielddata=record
transmitter:string[4];
data:string[4];
date:string[8];
time:string[8];
end;
```

```
//Field information database details
farmdata=record
farmno:string[4];
farmwidth:integer;
farmlength:integer;
furrowsmon:integer;
end;

TForm1 = class(TForm)
GroupBox1:  TGroupBox;
GroupBox2:  TGroupBox;
MainMenu1:  TMainMenu;
File1:  TMenuItem;
Exit1:  TMenuItem;
Label1:  TLabel;
Label2:  TLabel;
Label3:  TLabel;
edtlength:  TEdit;
edtwidth:  TEdit;
edtfurrows:  TEdit;
Label4:  TLabel;
boxtype:  TComboBox;
Label5:  TLabel;
edtid:  TEdit;
Label6:  TLabel;
edtx:  TEdit;
edty:  TEdit;
Label7:  TLabel;
Label8:  TLabel;
btneditline:  TButton;
ListView1:  TListView;
btnsave:  TButton;
btnedit:  TButton;
btnsaveedit:  TButton;
btndelete:  TButton;
btnadd:  TButton;
btnreturn:  TButton;
```

```
btncancel:  TButton;
btnchangedata:  TButton;
btnok:  TButton;
btnfarmcancel:  TButton;
memstatus:  TMemo;
Image1:  TImage;
Label11:  TLabel;
ComPort1:  TComPort;
btnport:  TButton;
Label10:  TLabel;
cbports:  TComboBox;
btneditOK: TButton;
ListView2:  TListView;
ComboBox1:  TComboBox;
Label9:  TLabel;
ComboBox2:  TComboBox;
ComboBox3:  TComboBox;
Label12:  TLabel;
Label13:  TLabel;
Label14:  TLabel;
procedure Exit1Click(Sender:  TObject);
procedure btneditlineClick(Sender:  TObject);
procedure FormCreate(Sender:  TObject);
procedure btnsaveClick(Sender:  TObject);
procedure btneditClick(Sender:  TObject);
procedure btnsaveeditClick(Sender:  TObject);
procedure btndeleteClick(Sender:  TObject);
procedure btnaddClick(Sender:  TObject);
procedure btnreturnClick(Sender:  TObject);
procedure btncancelClick(Sender:  TObject);
procedure btnchangedataClick(Sender:  TObject);
procedure btnfarmcancelClick(Sender:  TObject);
procedure btnokClick(Sender:  TObject);
procedure FormClose(Sender:  TObject; var Action:  TCloseAction);
procedure ComPort1ReceiveCallBack(Data:  String);
procedure btnportClick(Sender:  TObject);
procedure ListView1ColumnClick(Sender:  TObject; Column:  TListColumn);
```

```
procedure ListView1Compare(Sender:  TObject; Item1, Item2:  TListItem;
Data:  Integer; var Compare:  Integer);
procedure btneditOKClick(Sender:  TObject);
procedure ComboBox2Change(Sender:  TObject);
procedure Image1MouseMove(Sender:  TObject; Shift:  TShiftState; X,
Y: Integer);
procedure Image1Click(Sender:  TObject);
private
( Private declarations )
procedure Deletec1(drop:  String);
procedure Deletec3(drop3:  String);
procedure Refresh(listview1:TListView);
procedure ShowImage(Image1:TImage);
public
( Public declarations )
end;

var
Form1:  TForm1;
// Define fields of sensor information database
c:sensordata;
ci:file of sensordata;
co:file of sensordata;
i:Integer;
ID:string[4];
Field:integer; // field were sensor is located
Name:string[15]; // type of sensor
Xcoord:integer; // x coordinate of sensor
Ycoord:integer; // y coordinate of sensor

// Define fields of received sensor data database
c2:infielddata;
ci2:file of infielddata;
co2:file of infielddata;
i2:Integer; // record locations
idv2:string[4]; // sensor identifier
datav2:string[4]; // data
datev2:string[8]; // date
```

```
timev2:string[8]; // time

// Define fields of farm information data database
c3:farmdata;
ci3:file of farmdata;
co3:file of farmdata;
i3:Integer; // record locations
farmid:string[4]; // farm number
farmwidth:integer; // farm width
farmlength:integer; // farm length
nofurrows:integer; // number of furrows

// Other variables
n,t,indicator,farms,farmcount,farmsel,counter:integer;
drop,drop2,drop3:string;
bmp:TBitmap;
cwidth,cheight:integer; // dimensions of canvas for paddock
wide,high,newwide,newhigh,furrows,fields:integer; // number of furrows

blnPortOpened:boolean;

ColumnToSort:  Integer;
LastSortedColumn:integer;
Ascending:boolean;

implementation

($R *.dfm)

procedure TForm1.Exit1Click(Sender:  TObject);
begin
Application.Terminate; // Exit graphic user interface
end;

procedure TForm1.btneditlineClick(Sender:  TObject);
begin
//When edit button of database is clicked, load new window
btneditline.Hide;
btnchangedata.Hide;
```

```
Image1.Hide;
memstatus.Hide;
Label13.Hide;
Label14.Hide;
GroupBox1.Hide;
GroupBox2.Hide;
listview1.Show;
listview2.Hide;
btnadd.Show;
btndelete.Show;
btnedit.Show;
btnreturn.Show;
indicator:=1;
Refresh(listview1);
end;

procedure TForm1.FormCreate(Sender:  TObject);
var
farms,i:integer;
begin
blnPortOpened:=False;
GroupBox1.Hide;
GroupBox2.Hide;
listview1.Hide;
listview2.Hide;
btnAdd.Hide;
btnEdit.Hide;
btnDelete.Hide;
btnReturn.Hide;
Button3.Show;
// Show serial ports available to transfer information base station
cbports.Show;
EnumPorts( cbPorts.Items );
if cbPorts.Items.Count > 0 then
cbPorts.ItemIndex := 0; // show first available port
```

```
Assignfile(ci3,cdata3);
($i-)
Reset(ci3);
($i+)
if ioresult <> 0 then rewrite(ci3);
farms:=filesize(ci3);
closefile(ci3);

Combobox2.Text:=inttostr(1);


for i:=1 to farms do
begin
ComboBox2.Items.Add(inttostr(i));
end;

farmsel:=0;
Label14.Show;
memstatus.Show;
ShowImage(Image1);
end;

procedure TForm1.btnsaveClick(Sender:  TObject);
begin
// If button pressed to save new database entry
btnAdd.Show;
btnEdit.Show;
btnDelete.Show;
btnReturn.Show;
GroupBox1.Hide;
GroupBox2.Hide;
Assignfile(ci,cdata);
($i-)
Reset(ci);
($i+)
if ioresult <> 0 then rewrite(ci);
i:=filesize(ci);
// Save entered values to new database entry
```

```
c.id:=edtid.Text;
c.farmno:=strtoint(Combobox3.Text);
c.sensor:=boxtype.Text;
c.xcoordinate:=strtoint(edtx.Text);
c.ycoordinate:=strtoint(edty.Text);
seek(ci,i);
write(ci,c);
closefile(ci);
edtid.Text:='';
boxtype.Refresh;
edtx.Text:='';
edty.Text:='';
// Refresh display of database entries
Refresh(listview1);
end;

procedure TForm1.btneditClick(Sender:  TObject);
begin
// If edit button of database is pressed, edit the entry of the database selected
btnadd.Hide;
btnedit.Hide;
btndelete.Hide;
btnreturn.Hide;
// For farm sensors
if indicator=1 then
begin
GroupBox1.Hide;
GroupBox2.Show;
btnsave.Hide;
btnsaveedit.Show;
// Obtain each database entry was selected
t:=listview1.Selected.Index;
// Display database entries in editboxes to edit
edtid.Text:=listview1.Items[t].Caption;
combobox3.Text:=listview1.Items[t].SubItems[0];
boxtype.Text:=listview1.Items[t].SubItems[1];
edtx.Text:=listview1.Items[t].SubItems[2];
```

```
edty.Text:=listview1.Items[t].SubItems[3];
end
// For farm data
else if indicator=2 then
begin
GroupBox1.Show;
GroupBox2.Hide;
btnOK.Hide;
btneditOK.Show;
// Obtain each database entry was selected
t:=listview2.Selected.Index;
// Display database entries in editboxes to edit
ComboBox1.Text:=listview2.Items[t].Caption;
edtlength.Text:=listview2.Items[t].SubItems[0];
edtwidth.Text:=listview2.Items[t].SubItems[1];
edtfurrows.Text:=listview2.Items[t].SubItems[2];
end;
end;

procedure TForm1.btnsaveeditClick(Sender:  TObject);
begin
// If entry in database has been edited and now saved
btnadd.Show;
btnedit.Show;
btndelete.Show;
btnReturn.Show;
GroupBox1.Hide;
GroupBox2.Hide;
Assignfile(ci,cdata);
($i-)
Reset(ci);
($i+)
if ioresult <> 0 then rewrite(ci);
// Save over database entry with new values
c.id:=edtid.Text;
c.farmno:=strtoint(combobox3.Text);
c.sensor:=boxtype.Text;
```

```
c.xcoordinate:=strtoint(edtx.Text);
c.ycoordinate:=strtoint(edty.Text);
seek(ci,t);
write(ci,c);
closefile(ci);
Refresh(listview1);
end;

procedure TForm1.btndeleteClick(Sender:  TObject);
begin
GroupBox1.Hide;
GroupBox2.Hide;
// For farm sensors
if indicator=1 then
begin
t:=listview1.Selected.Index;
drop:=listview1.Items[t].Caption;
Deletec1(drop);
Refresh(listview1);
end
// For farm data
else if indicator=2 then
begin
t:=listview2.Selected.Index;
drop3:=listview2.Items[t].Caption;
Deletec3(drop3);
Refresh(listview2);
end;
end;

// procedures I wrote

procedure Tform1.Deletec1(drop:  String);
var
FETCH:Integer;
found:boolean;
begin
Assignfile(ci,cdata);
```

```
Reset(ci);
($i-)
found:=false;
Reset(ci);
FETCH:=-1; // don't skip first file
//(it contains file information) - if we did, FETCH:=0;
while (not eof(ci)) and (not found) do
begin
FETCH:=FETCH+1;
Seek(ci,FETCH);
Read(ci,c);
if (drop=c.id) then
($i+)
begin
c.id:='De';
Seek(ci,FETCH);
Write(ci,c);
found:=true;
end;
end;
Closefile(ci);
Rename(ci,'@@.@@@');
Assignfile(ci,'@@.@@@');
Reset(ci);
Assignfile(co,TEMP);
Rewrite(co);
while not eof(ci) do
begin
($i-)
Read(ci,c);
($i+)
if (c.id<>'De') then Write(co,c);
end;
Closefile(ci);
Closefile(co);
Rename(co,cdata);
Erase(ci);
```

```
end;

procedure Tform1.Deletec3(drop3:  String);
var
FETCH:Integer;
found:boolean;
begin
// To delete a database entry
Assignfile(ci3,cdata3);
Reset(ci3);
($i-)
found:=false;
Reset(ci3);
FETCH:=-1; // don't skip first file
//(it contains file information) - if we did, FETCH:=0;
while (not eof(ci3)) and (not found) do
begin
FETCH:=FETCH+1;
Seek(ci3,FETCH);
Read(ci3,c3);
if (drop3=c3.farmno) then
($i+)
begin
c3.farmno:='De';
Seek(ci3,FETCH);
Write(ci3,c3);
found:=true;
end;
end;
Closefile(ci3);
Rename(ci3,'@@.@@@');
Assignfile(ci3,'@@.@@@');
Reset(ci3);
Assignfile(co3,TEMP);
Rewrite(co3);
while not eof(ci3) do
begin
```

```
($i-)
Read(ci3,c3);
($i+)
if (c3.farmno<>'De') then Write(co3,c3);
end;
Closefile(ci3);
Closefile(co3);
Rename(co3,cdata3);
Erase(ci3);
end;

procedure TForm1.btnaddClick(Sender:  TObject);
begin
// To add database entry
btnadd.Hide;
btnedit.Hide;
btndelete.Hide;
btnreturn.Hide;
// For farm sensors
if indicator = 1 then
begin
btnsaveedit.Hide;
btnsave.Show;
GroupBox1.Hide;
GroupBox2.Show;
edtid.Text:='';
boxtype.ItemIndex:=-1;
edtx.Text:='';
edty.Text:='';
end
// For farm data
else if indicator = 2 then
begin
btneditOK.Hide;
btnOK.Show;
GroupBox1.Show;
GroupBox2.Hide;
```

```
ComboBox1.Text:='';
edtlength.Text:='';
edtwidth.Text:='';
edtfurrows.Text:='';
end

else if indicator = 3 then
begin
btneditOK.Hide;
btnOK.Show;
GroupBox1.Show;
GroupBox2.Hide;
ComboBox1.Text:='';
edtlength.Text:='';
edtwidth.Text:='';
edtfurrows.Text:='';
end;
end;

procedure TForm1.btnreturnClick(Sender:  TObject);
begin
// To return from window to edit or add
// database entry to graphical display
Label13.Hide;
btneditline.Show;
btnchangedata.Show;
indicator:=0;
Image1.Show;
GroupBox1.Hide;
GroupBox2.Hide;
btnadd.Hide;
btndelete.Hide;
btnedit.Hide;
btnreturn.Hide;
listview1.Hide;
listview2.Hide;
indicator:=1;
```

```
//Obtain database entries to draw new
//graphical display of farm information
Refresh(listview1);
indicator:=2;
Refresh(listview2);

Assignfile(ci3,cdata3);
($i-)
Reset(ci3);
($i+)
if ioresult <> 0 then rewrite(ci3);
farms:=filesize(ci3);
closefile(ci3);

Combobox2.Clear;

for i:=1 to farms do
begin
ComboBox2.Items.Add(inttostr(i));
end;
Combobox2.Text:=inttostr(1);

// Update graphic display
Label14.Show;
memstatus.Show;
ShowImage(Image1);
end;

procedure TForm1.btncancelClick(Sender:  TObject);
begin
btnadd.Show;
btnedit.Show;
btndelete.Show;
btnReturn.Show;
GroupBox1.Hide;
GroupBox2.Hide;
end;
```

```
procedure TForm1.btnchangedataClick(Sender:  TObject);
begin
// Change database addressed
btneditline.Hide;
btnchangedata.Hide;
Image1.Hide;
Label13.Hide;
Label14.Hide;
memstatus.Hide;
GroupBox1.Hide;
GroupBox2.Hide;
listview1.Hide;
listview2.Show;
btnadd.Show;
btndelete.Show;
btnedit.Show;
btnreturn.Show;
indicator:=2;
Refresh(listview2);
end;

procedure TForm1.btnfarmcancelClick(Sender:  TObject);
begin
//If change to database entry cancelled
btnadd.Show;
btnedit.Show;
btndelete.Show;
btnReturn.Show;
GroupBox1.Hide;
GroupBox2.Hide;
end;

procedure TForm1.btnokClick(Sender:  TObject);
var
i:  integer;
begin
// If change in database entry confirmed update database entries
btnAdd.Show;
```

```
btnEdit.Show;
btnDelete.Show;
btnReturn.Show;
GroupBox1.Hide;
GroupBox2.Hide;


Assignfile(ci3,cdata3);
// ($i-)
Reset(ci3);
// ($i+)
if ioresult <> 0 then rewrite(ci3);
i:=filesize(ci3);
c3.farmno:=Combobox1.Text;
c3.farmlength:=strtoint(edtlength.Text);
c3.farmwidth:=strtoint(edtwidth.Text);
c3.furrowsmon:=strtoint(edtfurrows.Text);
seek(ci3,i);
write(ci3,c3);
closefile(ci3);
Combobox1.Text:='';
edtlength.Text:='';
edtwidth.Text:='';
edtfurrows.Text:='';
indicator:=2;
Refresh(listview2);
end;

procedure TForm1.FormClose(Sender:  TObject; var Action:  TCloseAction);
begin
bmp.Free;
ComPort1.Close; // on form close
ComPort1.Free; // on form destroy
end;

procedure TForm1.ComPort1ReceiveCallBack(Data:  String);
var
t,counter:integer;
```

```
rstring,rid,rdatau,rdatal,rday,rmonth,ryear,rhr,rmin,rsec:string;
begin
Label13.Hide;
if counter=1 then sleep(3000);
sleep(5000); // pause for 5 seconds
memstatus.Lines.Add(Data); // display incoming serial data
rstring:=Data;

// ------------------------------------------------------------------------
// Identify sensor identification, data, date and time parts of message
// ------------------------------------------------------------------------
t:=Pos('-',rstring); // find location of delimiter, '-'
Delete(rstring,1,t); // delete start of message bytes

for counter := 1 to 8 do
begin
t:=Pos('-',rstring); // find location of delimiter, '-'

if counter = 1 then
rid := copy(rstring,0,t-1) // separate id
else if counter = 2 then
begin
rdatau := copy(rstring,0,t-2); // separate datau (decimal)
rdatal := copy(rstring,t-1,t-2); // separate datal (decimal)
end
else if counter = 3 then
rday := copy(rstring,0,t-1) // separate day
else if counter = 4 then
rmonth := copy(rstring,0,t-1) // separate month
else if counter = 5 then
ryear := copy(rstring,0,t-1) // separate year
else if counter = 6 then
rhr := copy(rstring,0,t-1) // separate hour
else if counter = 7 then
rmin := copy(rstring,0,t-1) // separate minute
else if counter = 8 then
rsec := copy(rstring,0,t-1); // separate second
```

```
Delete(rstring,1,t); // delete string section just separated
end;

t:=Pos('-',rstring); // find location of delimiter, '-'
Delete(rstring,t,t+2); // delete end of message bytes
// ----------------------------------------------------------------------------

// ----------------------------------------------------------------------------
// Interpret data
// ----------------------------------------------------------------------------
// Decide where the sensors will be positioned

// ----------------------------------------------------------------------------

// Put into comma separated values (csv) file format
Assignfile(ci2,cdata2);
($i-)
Reset(ci2);
($i+)
if ioresult <> 0 then rewrite(ci2);
i2 := filesize(ci2);
c2.transmitter := rid;
c2.data := rdatau + '.'  + rdatal;
c2.date := rday + '-' + rmonth + '-' + ryear;
c2.time := rhr + ':'  + rmin + ':'  + rsec;

seek(ci2,i2);
write(ci2,c2);
closefile(ci2);

//If number of memo lines exceeds 500, do not show these lines
if memstatus.Lines.Count > 500 then
memstatus.Clear;
Label14.Show;
memstatus.Show;
ShowImage(Image1);
end;
```

```
procedure TForm1.btnportClick(Sender:  TObject);
begin
cbPorts.Enabled := true;
ComPort1.Port := cbPorts.Items[cbPorts.ItemIndex];

// Close port
if blnPortOpened=True then
begin
blnPortOpened:=False;
ComPort1.Close;
end
else
// Open port
begin
counter:=1;
blnPortOpened:=True;
ComPort1.Open
end;
end;

procedure Tform1.Refresh(listview1:TListView);
begin
// Refreshes the display of the database
// For farm sensors
if indicator=1 then
begin
listview1.Items.Clear;
Assignfile(ci,cdata);
($i-)
Reset(ci);
($i+)
while not eof(ci) do
begin
read(ci,c);
listview1.Items.Add;
n:=listview1.Items.Count-1;
// Insert database entries into database displayed in user interface
listview1.Items[n].Caption:=c.id;
```

```
listview1.Items[n].SubItems.Add(inttostr(c.farmno));
listview1.Items[n].SubItems.Add(c.sensor);
listview1.Items[n].SubItems.Add(inttostr(c.xcoordinate));
listview1.Items[n].SubItems.Add(inttostr(c.ycoordinate));
end;
Closefile(ci);
end
// For farm data
else if indicator=2 then
begin
listview2.Items.Clear;
Assignfile(ci3,cdata3);
($i-)
Reset(ci3);
($i+)
while not eof(ci3) do
begin
read(ci3,c3);
listview2.Items.Add;
n:=listview2.Items.Count-1;
// Insert database entries into database displayed in user interface
listview2.Items[n].Caption:=c3.farmno;
listview2.Items[n].SubItems.Add(inttostr(c3.farmlength));
listview2.Items[n].SubItems.Add(inttostr(c3.farmwidth));
listview2.Items[n].SubItems.Add(inttostr(c3.furrowsmon));
end;
Closefile(ci3);
end
else if indicator=3 then
begin
listview2.Items.Clear;
Assignfile(ci2,cdata2);
$i-
Reset(ci2);
$i+
while not eof(ci2) do
begin
```

```
read(ci2,c2);
listview2.Items.Add;
n:=listview2.Items.Count-1;
// Insert database entries into database displayed in user interface
listview2.Items[n].Caption:=c2.transmitter;
listview2.Items[n].SubItems.Add(c2.data);
listview2.Items[n].SubItems.Add(c2.date);
listview2.Items[n].SubItems.Add(c2.time);
end;
Closefile(ci2);
end;
end;

procedure TForm1.btneditOKClick(Sender:  TObject);
var
t:integer;
begin
// If database entry was edited
btnadd.Show;
btnedit.Show;
btndelete.Show;
btnReturn.Show;
GroupBox1.Hide;
GroupBox2.Hide;
Assignfile(ci3,cdata3);
($i-)
Reset(ci3);
($i+)
if ioresult <> 0 then rewrite(ci3);
// Write editted values entered in window to database
c3.farmno:=Combobox1.Text;
c3.farmlength:=strtoint(edtlength.Text);
c3.farmwidth:=strtoint(edtwidth.Text);
c3.furrowsmon:=strtoint(edtfurrows.Text);
t:=listview2.Selected.Index;
seek(ci3,t);
write(ci3,c3);
```

```
closefile(ci3);
indicator:=2;
// Refresh database display
Refresh(listview2);
end;

procedure Tform1.ShowImage(Image1:TImage);
var
f,tt,s,x,y,ypos,noitems,state,progress,length:integer;
info,sense,fheight:string;
begin
// Update the graphical display of sensors and farm information
Image1.Canvas.Pen.Color:=clMaroon;//$000080FF;
Image1.Canvas.Brush.Color:=clMaroon;//$000080FF;
cwidth:=550;
cheight:=650;
Image1.Canvas.Rectangle(0,0,cwidth,700);

// Get farm coordinates for each field
indicator:=2;
Refresh(listview2);
high:=strtoint(listview2.Items[farmsel].SubItems[0]);
wide:=strtoint(listview2.Items[farmsel].SubItems[1]);
furrows:=strtoint(listview2.Items[farmsel].SubItems[2]);

// Find number of field from data
Assignfile(ci3,cdata3);
($i-)
Reset(ci3);
($i+)
if ioresult <> 0 then rewrite(ci3);
n:=filesize(ci3);
closefile(ci3);

// Draw field to fit in canvas in correct ratios
Image1.Canvas.Brush.Color:=$000080FF;
Image1.Canvas.Brush.Color:=clGreen;
newhigh:=cheight;
```

```
newwide:=cwidth;
Image1.Canvas.Rectangle(0,0,newwide,newhigh);

for tt:=1 to furrows do
begin
// Draw furrows on paddock as black lines
Image1.Canvas.Brush.Color:=clBlack;
Image1.Canvas.Pen.Color:=clBlack;
Image1.Canvas.Rectangle(round((tt-0.05)*newwide/...
(furrows+1)),0,round((tt+0.05)*newwide/(furrows+1)),newhigh);

// Label furrows on paddock
Image1.Canvas.Brush.Color:=$000080FF;
Image1.Canvas.Pen.Color:=clBlack;
Image1.Canvas.TextOut(round((tt-0.05)*newwide/(furrows+1)),newhigh+5,inttostr(tt));
end;

// Determine location (furrow number) of sensor
indicator:=1;
Refresh(listview1);
f:=1; // initialise furrow counter
Image1.Canvas.Brush.Color:=$000080FF;
Image1.Canvas.Pen.Color:=clBlack;
Image1.Canvas.TextOut(round((f-0.05)*newwide/(furrows+1)),newhigh+5,inttostr(f));
for t := 0 to 3 do //listview1.Items.Count-1 do
begin
if listview1.Items[t].SubItems[0]=inttostr(farmsel+1) then
begin
x:=strtoint(listview1.Items[t].SubItems[2]); // x-coordinate of sensor
y:=strtoint(listview1.Items[t].SubItems[3]); // y-coordinate of sensor

// Display sensor information on field
Image1.Canvas.Brush.Color:=clLtGray;
Image1.Canvas.Pen.Color:=clBlack;

if t>1 then
begin
// If two sensors have the same x-coordinate
```

```
// sensors are located in the same furrow
if strtoint(listview1.Items[t].SubItems[2])<>...
strtoint(listview1.Items[t-1].SubItems[2]) then
begin
if strtoint(listview1.Items[t].SubItems[0])=...
strtoint(listview1.Items[t-1].SubItems[0]) then
begin
f:=f+1;
Image1.Canvas.Brush.Color:=clLtGray;
Image1.Canvas.Pen.Color:=clBlack;
Image1.Canvas.TextOut(round((f-0.05)*newwide/(furrows+1)),newhigh+5,inttostr(f));
end;
end;
end;

indicator:=2;
Refresh(listview2);
length:=strtoint(listview2.Items[farmsel].SubItems[0]);

// Read in sensor data from .dat file to display on canvas
indicator:=3;
Refresh(listview2);
state:=strtoint(listview2.Items[t].SubItems[0]);

// Update furrow water progress
indicator:=1;
Refresh(listview1);
if listview1.Items[t].SubItems[1]='Advance meter' then
begin
// Show advance front progress down field
Image1.Canvas.Brush.Color:=clBlue;
Image1.Canvas.Pen.Color:=clBlue;
progress:=round(state*cheight*strtoint(listview1.Items[t].SubItems[3])/length);
Image1.Canvas.Rectangle(round((f-0.05)*newwide/(furrows+1)),0,
round((f+0.05)*newwide/(furrows+1)),round(progress));
// Assign colour for drawing sensor
// yellow for advance, red for inflow/outflow, green for other
Image1.Canvas.Brush.Color:=clYellow;
```

```
end
else if (listview1.Items[t].SubItems[0]='Inflow meter') or
(listview1.Items[t].SubItems[0]='Outflow meter') then
Image1.Canvas.Brush.Color:=clRed
else
Image1.Canvas.Brush.Color:=clGreen;
Image1.Canvas.Brush.Color:=clYellow;

// Draw sensors on paddock
for s:=0 to listview1.Items.Count-1 do
begin
Image1.Canvas.Pen.Color:=clBlack;
y:=strtoint(listview1.Items[s].SubItems[3]); // y-coordinate of sensor
Image1.Canvas.Ellipse(round((f-0.05)*newwide/(furrows+1)),
round((y/high*cheight-0.1*newwide/(furrows+1))),
round((f+0.05)*newwide/(furrows+1)),
round(y/high*cheight)); // Draw circle with black outline
end;
end;
for s:=0 to listview1.Items.Count-1 do
begin
Image1.Canvas.Pen.Color:=clBlack;
if listview1.Items[s].SubItems[1]='Advance meter' then // for advance meter
Image1.Canvas.Brush.Color:=clYellow
else if (listview1.Items[s].SubItems[1]='Inflow meter')
or (listview1.Items[s].SubItems[0]='Inflow meter') then // for inflow or outflow
meter
Image1.Canvas.Brush.Color:=clRed
else
Image1.Canvas.Brush.Color:=clGreen;

y:=strtoint(listview1.Items[s].SubItems[3]); // y-coordinate of sensor
Image1.Canvas.Ellipse(round((f-0.05)*newwide/(furrows+1)),
round(0.95*(y/high*cheight-0.1*newwide/(furrows+1))),
round((f+0.05)*newwide/(furrows+1)),
round(0.95*y/high*cheight)); // Draw circle with black outline // Draw yellow circle
with black outline
Image1.Canvas.Ellipse(round((f-0.05)*newwide/(...
```

```
furrows+1)),round(0.95*(y/wide*cwidth-0.1*newwide/(furrows+1))
),...
round((f+0.05)*newwide/(furrows+1)),round(0.95*y/wide*cwidth))
end;
Image1.Show;
end;

procedure TForm1.Image1MouseMove(Sender:  TObject; Shift:  TShiftState; X,
Y: Integer);
var
nn,xmouse,ymouse,xpos,ypos,xdisp,ydisp,ff:integer;
sense,info:string;
begin
Label13.Show;
Label14.Show;
memstatus.Show;
ShowImage(Image1);

// Find location of mouse
xmouse:=mouse.CursorPos.X;
ymouse:=mouse.CursorPos.Y;

indicator:=1;
Refresh(listview1);
ff:=1; // initialise furrow counter

for nn:=0 to listview1.Items.Count-1 do // For each of the sensors
begin
indicator:=1;
Refresh(listview1);
xpos:=strtoint(listview1.Items[nn].SubItems[2]); // x-coordinate of sensor
ypos:=strtoint(listview1.Items[nn].SubItems[3]); // y-coordinate of sensor

// Identify sensor that mouse is passing over
if nn>1 then
begin
// If two sensors have the same x-coordinate
// sensors are located in the same furrow
```

```
if xpos<>strtoint(listview1.Items[nn-1].SubItems[2]) then
begin
if strtoint(listview1.Items[nn].SubItems[0])=...
strtoint(listview1.Items[nn-1].SubItems[0]) then
begin
ff:=ff+1;
end;
end;
end;

indicator:=2;
Refresh(listview2);
high:=strtoint(listview2.Items[farmsel].SubItems[0]); // Length of field
wide:=strtoint(listview2.Items[farmsel].SubItems[1]); // Width of field
furrows:=strtoint(listview2.Items[farmsel].SubItems[2]);// No of furrows
xpos:=round(ff*newwide/(furrows+1)+270); // Transformed x-coordinate of sensor
ypos:=round(ypos*cheight/high+80); // Transformed y-coordinate of sensor

// Define pen and brush colours ready to write sensor text on canvas
Image1.Canvas.Brush.Color:=clWhite;
Image1.Canvas.Pen.Color:=clBlack;

// If the mouse is dragged over a point near a sensor then display info
if (xmouse>0.95*xpos) and (xmouse<1.05*xpos) and
(ymouse>0.95*ypos) and (ymouse<1.05*ypos) then
begin
// Display sensor information on field
xdisp:=xpos-300;
ydisp:=ypos-105;

// Read in sensor data from .dat file to display on canvas
indicator:=3;
Refresh(listview2);
sense:=listview2.Items[nn].Caption;
info:=listview2.Items[nn].SubItems[0];
Image1.Canvas.TextOut(xdisp,ydisp,sense+',+info);
end;
end;
```

```
end;

procedure TForm1.ComboBox2Change(Sender:  TObject);
begin
farmsel:=Combobox2.ItemIndex;
Label14.Show;
Label13.Hide;
memstatus.Show;
ShowImage(Image1);
end;

procedure TForm1.Image1Click(Sender:  TObject);
var
xmouse,ymouse,ff,nn,xpos,ypos,xdisp,ydisp:integer;
sense:string;
begin
Label13.Show;
Label13.Color:=clYellow;
Label14.Show;
memstatus.Show;
ShowImage(Image1);
// Find location of mouse
xmouse:=mouse.CursorPos.X;
ymouse:=mouse.CursorPos.Y;

indicator:=1;
Refresh(listview1);
ff:=1; // initialise furrow counter

for nn:=0 to listview1.Items.Count-1 do // For each of the sensors
begin
indicator:=1;
Refresh(listview1);
xpos:=strtoint(listview1.Items[nn].SubItems[2]); // x-coordinate of sensor
ypos:=strtoint(listview1.Items[nn].SubItems[3]); // y-coordinate of sensor
xdisp:=xpos;
ydisp:=ypos;
```

```
// Identify sensor that mouse is passing over
if nn>1 then
begin
// If two sensors have the same x-coordinate
// sensors are located in the same furrow
if xpos<>strtoint(listview1.Items[nn-1].SubItems[2]) then
begin
if strtoint(listview1.Items[nn].SubItems[0])=...
strtoint(listview1.Items[nn-1].SubItems[0]) then
begin
ff:=ff+1;
end;
end;
end;

indicator:=2;
Refresh(listview2);
high:=strtoint(listview2.Items[farmsel].SubItems[0]); // Length of field
wide:=strtoint(listview2.Items[farmsel].SubItems[1]); // Width of field
furrows:=strtoint(listview2.Items[farmsel].SubItems[2]);// No of furrows
xpos:=round(ff*newwide/(furrows+1)+270); // Transformed x-coordinate of sensor
ypos:=round(ypos*cheight/high+80); // Transformed y-coordinate of sensor

// Define pen and brush colours ready to write sensor text on canvas
Image1.Canvas.Brush.Color:=clLtGray;
Image1.Canvas.Pen.Color:=clBlack;

// If the mouse is dragged over a point near a sensor then display info
if (xmouse>0.95*xpos) and (xmouse<1.05*xpos) and
(ymouse>0.95*ypos) and (ymouse<1.05*ypos) then
begin
// Read in sensor data from .dat file to display on canvas
indicator:=3;
Refresh(listview2);
sense:=listview2.Items[nn].Caption;
xdisp:=xpos+10;
ydisp:=ypos-50;
Label13.Caption:='Data incoming...';
```

```
Label13.Left:=xdisp;
Label13.Top:=ydisp;
// ComPort1.send();
end;
end;
end;

end.
```