if (vendDev == 0)

{

if (neVTomSilose) { inMonitor = true; neVTomSilose = false; goto linkAnotherSilos; }

if (numberSilos != asForm.numberMonitor().ToString() && inMonitor) { inMonitor = false; neVTomSilose = true; continue; }

inMonitor = (asForm.numberMonitor() != "-1") ? true : false;

int leg = thDB.GetPinWire(numberWire);

int countSensThis = thDB.GetCountSensInWire(numberWire);

#region sensorRead

try

{

string[] ResponseBuf = new string[countSensThis];

ushort[] ResponseBufferHoldingRegs = new ushort[countSensThis];

WriteModbusData(0, (ushort)leg);

WriteModbusData(1, 1);

System.Threading.Thread.Sleep(500);

while (ResponseBufferHoldingRegs[1] != 2)

{

ResponseBufferHoldingRegs = ModbusMasterRTU.ReadHoldingRegisters(Address, 0, 2);

System.Threading.Thread.Sleep(200);

}

if (countSensThis>50)

{

ushort[] ResponseBufferHoldingRegs1 = new ushort[25];

ushort[] ResponseBufferHoldingRegs2 = new ushort[25];

ushort[] ResponseBufferHoldingRegs3 = new ushort[countSensThis - 50];

ResponseBufferHoldingRegs1 = ModbusMasterRTU.ReadHoldingRegisters(Address, 2, Convert.ToUInt16(25));

ResponseBufferHoldingRegs2 = ModbusMasterRTU.ReadHoldingRegisters(Address, 27, Convert.ToUInt16(25));

ResponseBufferHoldingRegs3 = ModbusMasterRTU.ReadHoldingRegisters(Address, 52, Convert.ToUInt16(countSensThis - 50));

ResponseBufferHoldingRegs = new ushort[countSensThis];

for (int h = 0; h < 25; h++)

{

ResponseBufferHoldingRegs[h] = ResponseBufferHoldingRegs1[h];

}

for (int h = 25; h < 50; h++)

{

ResponseBufferHoldingRegs[h] = ResponseBufferHoldingRegs2[h-25];

}

for (int h = 50; h < countSensThis; h++)

{

ResponseBufferHoldingRegs[h] = ResponseBufferHoldingRegs2[h - 50];

}

}

else if (countSensThis > 25 && countSensThis<51)

{

ushort[] ResponseBufferHoldingRegs1 = new ushort[25];

ushort[] ResponseBufferHoldingRegs2 = new ushort[countSensThis-25];

ResponseBufferHoldingRegs1 = ModbusMasterRTU.ReadHoldingRegisters(Address, 2, Convert.ToUInt16(25));

ResponseBufferHoldingRegs2 = ModbusMasterRTU.ReadHoldingRegisters(Address, 27, Convert.ToUInt16(countSensThis-25));

ResponseBufferHoldingRegs = new ushort[countSensThis];

for (int h=0;h<25;h++)

{

ResponseBufferHoldingRegs[h] = ResponseBufferHoldingRegs1[h];

}

for (int h = 25; h < countSensThis; h++)

{

ResponseBufferHoldingRegs[h] = ResponseBufferHoldingRegs2[h-25];

}

}

else

{

ResponseBufferHoldingRegs = ModbusMasterRTU.ReadHoldingRegisters(Address, 2, Convert.ToUInt16(countSensThis));

}

WriteModbusData(1, 0);

if (neVTomSilose) { inMonitor = true; neVTomSilose = false; goto linkAnotherSilos; }

if (numberSilos != asForm.numberMonitor().ToString() && inMonitor) { inMonitor = false; neVTomSilose = true; continue; }

inMonitor = (asForm.numberMonitor() != "-1") ? true : false;

for (int i = 0; i < countSensThis; i++)

{

ResponseBuf[i] = ResponseBufferHoldingRegs[i].ToString();

}

for (int i = 0; i < countSensThis; i++)

{

if (neVTomSilose) { inMonitor = true; neVTomSilose = false; goto linkAnotherSilos; }

if (numberSilos != asForm.numberMonitor().ToString() && inMonitor) { inMonitor = false; neVTomSilose = true; continue; }

inMonitor = (asForm.numberMonitor() != "-1") ? true : false;

int tempINTMult10 = int.Parse(ResponseBuf[i])>59000? int.Parse(ResponseBuf[i]) : int.Parse(ResponseBuf[i]) / 10;

string adrSensCompl = numberDevice.ToString() + ' ' + numberWire.ToString() + ' ' + (i + 1).ToString();

if (tempINTMult10 == 3276 || tempINTMult10 == 999 || tempINTMult10 == 85)

{

notFound = true;

}

else

{

if (tempINTMult10 > (red \* 10)) fire = 4;

else if (tempINTMult10 > (yellow \* 10)) fire = 3;

else if (tempINTMult10 > 0) fire = 2;

}

if (tempINTMult10 > 59000)

{

tempINTMult10 = ((65535 - tempINTMult10) \* (-1))/10;

if (tempINTMult10 < 0) fire = 1;

}

if (fire > fireOnWire) fireOnWire = fire;

if (fire > maxfire && fire > 2) maxfire = fire;

else if (fire < minfire) minfire = fire;

if (!notFound)

{

Temp = tempINTMult10 / (float)10;

if (Temp < minTemp) minTemp = Temp;

if (Temp > maxTemp) maxTemp = Temp;

sumTemp += Temp;

countTemp++;

if (inMonitor)

{

asForm.changeCell(numberWire, i, tempINTMult10, false, maxcountsensinsilosNOW - countSensThis, fire);

}

if ((thDB.LastTime(thDB.GetIdSens(adrSensCompl)) < DateTime.Now.AddMinutes(-writeTime)))

asForm.UpdateTempCKomplect(tempINTMult10, i, adrSensCompl);

else if ((fire > 3) && (thDB.LastTime(thDB.GetIdSens(adrSensCompl)) < DateTime.Now.AddMinutes(-(writeTime/4))))

{

asForm.UpdateTempCKomplect(tempINTMult10, i, adrSensCompl);

}

}

else

{

countAlarm++;

alarm = true;

asForm.changeAlarmLabel(numberSilos);

asForm.changeCell(numberWire, i, -1000, false, maxcountsensinsilosNOW - countSensThis,fire);

}

if (notFound)

{

notFound = false;

}

}

}

catch (TimeoutException)

{

asForm.Message("Timeout");

}

catch (Modbus.SlaveException e)

{

asForm.Message("Клиент не может прислать запрошенные данные");

}

catch (System.IO.IOException ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

catch (System.InvalidOperationException ex)

{

asForm.Message(ex.Message);

}

System.Threading.Thread.Sleep(requestInterval);

#endregion

if (asForm.bThreadStop) break;

if (countAlarm == countSensThis)

{

asForm.missedWire(numberWire);

}

}

#endregion

#region vend1

else if (vendDev == 1)

{

if (neVTomSilose) { inMonitor = true; neVTomSilose = false; goto linkAnotherSilos; }

if (numberSilos != asForm.numberMonitor().ToString() && inMonitor) { inMonitor = false; neVTomSilose = true; continue; }

inMonitor = (asForm.numberMonitor() != "-1") ? true : false;

int leg = thDB.GetPinWire(numberWire);

int countSensThis = thDB.GetCountSensInWire(numberWire);

#region sensorRead

try

{

string[] ResponseBuf = new string[countSensThis];

ushort[] ResponseBufferHoldingRegs = new ushort[countSensThis];

ushort newStartReg = 0;

switch (leg)

{

case 1:

newStartReg = 101;

break;

case 2:

newStartReg = 152;

break;

case 3:

newStartReg = 203;

break;

case 4:

newStartReg = 254;

break;

case 5:

newStartReg = 305;

break;

case 6:

newStartReg = 356;

break;

case 7:

newStartReg = 407;

break;

case 8:

newStartReg = 458;

break;

}

ResponseBufferHoldingRegs = ModbusMasterRTU.ReadHoldingRegisters(Address, newStartReg, Convert.ToUInt16(countSensThis));

for (int i = 0; i < countSensThis; i++)

{

ResponseBuf[i] = ResponseBufferHoldingRegs[i].ToString();

}

for (int i = 0; i < countSensThis; i++)

{

int tempINTMult10 = int.Parse(ResponseBuf[i]);

string adrSensCompl = numberDevice.ToString() + ' ' + numberWire.ToString() + ' ' + (i + 1).ToString();

if (tempINTMult10 == 32767 || tempINTMult10 == 85)

{

notFound = true;

}

else

{

if (tempINTMult10 > (red \* 10)) fire = 4;

else if (tempINTMult10 > (yellow \* 10)) fire = 3;

else if (tempINTMult10 > 0) fire = 2;

}

if (tempINTMult10 > 59000)

{

tempINTMult10 = (65535 - tempINTMult10) \* (-1);

if (tempINTMult10 < 0) fire = 1;

}

if (fire > fireOnWire) fireOnWire = fire;

if (fire > maxfire && fire > 2) maxfire = fire;

else if (fire < minfire) minfire = fire;

if (!notFound)

{

Temp = tempINTMult10 / (float)10;

if (Temp < minTemp) minTemp = Temp;

if (Temp > maxTemp) maxTemp = Temp;

sumTemp += Temp;

countTemp++;

if (inMonitor)

{

asForm.changeCell(numberWire, i, tempINTMult10, true, 0,fire);

}

if ((thDB.LastTime(thDB.GetIdSens(adrSensCompl)) < DateTime.Now.AddMinutes(-writeTime)))

asForm.UpdateTempCKomplect(tempINTMult10, i, adrSensCompl);

else if ((fire > 3) && (thDB.LastTime(thDB.GetIdSens(adrSensCompl)) < DateTime.Now.AddMinutes(-(writeTime / 4))))

{

asForm.UpdateTempCKomplect(tempINTMult10, i, adrSensCompl);

}

}

else

{

alarm = true;

asForm.changeAlarmLabel(numberSilos);

asForm.changeCell(numberWire, i, -1000, true, 0,fire);

}

if (notFound)

{

notFound = false;

}

}

}

catch (TimeoutException)

{

asForm.Message("Timeout");

}

catch (Modbus.SlaveException)

{

asForm.Message("Клиент не может прислать запрошенные данные");

}

catch (System.IO.IOException ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

catch (System.InvalidOperationException ex)

{

asForm.Message(ex.Message);

}

System.Threading.Thread.Sleep(requestInterval);

#endregion

if (asForm.bThreadStop) break;

}

#endregion

}

asForm.changeLabelOnCircle(numberWire, fireOnWire);

}

if (asForm.bThreadStop) break;

}

asForm.changeImage(maxfire!=0?maxfire:minfire, numberSilos);

asForm.changeLabelOnImage(numberSilos, maxTemp, minTemp, minTemp == 1000 ? 0 : (float)sumTemp / (float)countTemp);

if (numberSilos != asForm.numberMonitor().ToString() && inMonitor) { inMonitor = false; neVTomSilose = true; continue; }

inMonitor = (asForm.numberMonitor() != "-1") ? true : false;

if (inMonitor)

{

asForm.changeLabelTemp(maxTemp, minTemp, minTemp == 1000 ? 0 : (float)sumTemp / (float)countTemp);

}

if (asForm.bThreadStop) break;

if (!alarm) asForm.noVisibleAlarmLabel(numberSilos);

}

if (asForm.bThreadStop) break;

}

}

catch (System.Threading.ThreadAbortException abortException)

{

System.Windows.Forms.MessageBox.Show((string)abortException.ExceptionState);

}

}

catch(Exception e)

{

thDB.InsertLogErr(e.ToString(), DateTime.Now);

System.Threading.Thread.Sleep(1000);

}

}

**}**

}