Tak, první zajímavá změna – nastavení acq time nezávislé na base freq:

public void AcqTimeSetting(double acqtime)

{

int MessageID = 0x1;

Byte[] senddata = new Byte[8];

for (int i = 0; i < senddata.Length; i++)

senddata[i] = 0;

senddata[6] = (byte)MessageID;

float tmp = (float)acqtime;

byte[] tmp\_array = BitConverter.GetBytes(tmp);

senddata[0] = tmp\_array[0];

senddata[1] = tmp\_array[1];

senddata[2] = tmp\_array[2];

senddata[3] = tmp\_array[3];

udpClient.Send(senddata, senddata.Length);

if (!WaitResponse(MessageID))

MessageBox.Show("NACK - " + MessageID.ToString());

}

Parsování výstupu:

public PixelValue PixelWrite(UInt64 pixel, UInt32 time\_offset)

{

PixelValue tmp\_value = new PixelValue();

switch (DetectorMode)

{

// ToT/10 & ToA/18

case 0 : tmp\_value.count\_value = 1;

tmp\_value.ToT\_value = (int)((pixel >> 18) & 0x3FF);

tmp\_value.ToA\_value = (int)(pixel & 0x3FFFF);

tmp\_value.ToA\_offset = 0;

tmp\_value.coordinateY = (int)((pixel) >> 28 & 0xFF);

tmp\_value.coordinateX = (int)((pixel >> 36) & 0xFF);

break;

// ToT/14 & ToA/14

case 1: tmp\_value.count\_value = 1;

tmp\_value.ToT\_value = (int)((pixel >> 14) & 0x3FFF);

tmp\_value.ToA\_value = (int)(pixel & 0x3FFF);

tmp\_value.ToA\_offset = 0;

tmp\_value.coordinateY = (int)((pixel) >> 28 & 0xFF);

tmp\_value.coordinateX = (int)((pixel >> 36) & 0xFF);

break;

// ToT/10 & Count/4

case 2: tmp\_value.count\_value = (int)(pixel & 0xF);

tmp\_value.ToT\_value = (int)((pixel >> 18) & 0x3FF);

tmp\_value.ToA\_value = 0;

tmp\_value.ToA\_offset = 0;

tmp\_value.coordinateY = (int)((pixel) >> 28 & 0xFF);

tmp\_value.coordinateX = (int)((pixel >> 36) & 0xFF);

break;

// ToT/14

case 3: tmp\_value.count\_value = 1;

tmp\_value.ToT\_value = (int)(pixel & 0x3FFF);

tmp\_value.ToA\_value = 0;

tmp\_value.ToA\_offset = 0;

tmp\_value.coordinateY = (int)((pixel) >> 28 & 0xFF);

tmp\_value.coordinateX = (int)((pixel >> 36) & 0xFF);

break;

// ToA/10

case 4: tmp\_value.count\_value = 1;

tmp\_value.ToT\_value = 0;

tmp\_value.ToA\_value = (int)(pixel & 0x3FF);;

tmp\_value.ToA\_offset = 0;

tmp\_value.coordinateY = (int)((pixel) >> 28 & 0xFF);

tmp\_value.coordinateX = (int)((pixel >> 36) & 0xFF);

break;

// ToA/14

case 5: tmp\_value.count\_value = 1;

tmp\_value.ToT\_value = 0;

tmp\_value.ToA\_value = (int)(pixel & 0x3FFF);

tmp\_value.ToA\_offset = 0;

tmp\_value.coordinateY = (int)((pixel) >> 28 & 0xFF);

tmp\_value.coordinateX = (int)((pixel >> 36) & 0xFF);

break;

// Count/10

case 6: tmp\_value.count\_value = (int)(pixel & 0x3FF);

tmp\_value.ToT\_value = 0;

tmp\_value.ToA\_value = 0;

tmp\_value.ToA\_offset = 0;

tmp\_value.coordinateY = (int)((pixel) >> 28 & 0xFF);

tmp\_value.coordinateX = (int)((pixel >> 36) & 0xFF);

break;

// Count/14

case 7: tmp\_value.count\_value = (int)(pixel & 0x3FFF);

tmp\_value.ToT\_value = 0;

tmp\_value.ToA\_value = 0;

tmp\_value.ToA\_offset = 0;

tmp\_value.coordinateY = (int)((pixel) >> 28 & 0xFF);

tmp\_value.coordinateX = (int)((pixel >> 36) & 0xFF);

break;

// iToT/10 & ToA/18

case 8: tmp\_value.count\_value = 1;

tmp\_value.ToT\_value = (int)((pixel >> 18) & 0x3FF);

tmp\_value.ToA\_value = (int)(pixel & 0x3FFFF);

tmp\_value.ToA\_offset = 0;

tmp\_value.coordinateY = (int)((pixel) >> 28 & 0xFF);

tmp\_value.coordinateX = (int)((pixel >> 36) & 0xFF);

break;

// iToT/14 & ToA/14

case 9: tmp\_value.count\_value = 1;

tmp\_value.ToT\_value = (int)((pixel >> 14) & 0x3FFF);

tmp\_value.ToA\_value = (int)(pixel & 0x3FFF);

tmp\_value.ToA\_offset = 0;

tmp\_value.coordinateY = (int)((pixel) >> 28 & 0xFF);

tmp\_value.coordinateX = (int)((pixel >> 36) & 0xFF);

break;

// iToT/10 & Count/4

case 10: tmp\_value.count\_value = (int)(pixel & 0xF);

tmp\_value.ToT\_value = (int)((pixel >> 18) & 0x3FF);

tmp\_value.ToA\_value = 0;

tmp\_value.ToA\_offset = 0;

tmp\_value.coordinateY = (int)((pixel) >> 28 & 0xFF);

tmp\_value.coordinateX = (int)((pixel >> 36) & 0xFF);

break;

// iToT/14

case 11: tmp\_value.count\_value = 1;

tmp\_value.ToT\_value = (int)(pixel & 0x3FFF);

tmp\_value.ToA\_value = 0;

tmp\_value.ToA\_offset = 0;

tmp\_value.coordinateY = (int)((pixel) >> 28 & 0xFF);

tmp\_value.coordinateX = (int)((pixel >> 36) & 0xFF);

break;

}

return tmp\_value;

}

Mody jsou indexovány takto:

0: ToT10/ToA18

1: ToT14/ToA14

2: ContToT10

3: ContToT14

4: ContToA10

5: ContToA14

6: ContCount10

7: ContCount14

8: iToT10/ToA18

9: iToT14/ToA14

10: ContiToT10

11: ContiToT14