///

/// Creation of the frames and animation

/// A frame is a 2d-boolarray in which true values mean that an LED is on and false that it is off

/// A animation is required for the "video" mode and consists of a list of frames/2d-boolarrays

///

int row = 16;

int col = row;

bool[,] Frame = new bool[col, row], Frame2 = new bool[col, row];

for (int i = 0; i< 16; i++)

{

for (int j = 0; j< 16; j++)

{

if (i > j) Frame[i, j] = true;

else Frame2[i, j] = true;

}

}

List<bool[,]> Animation = new List<bool[,]>();

Animation.Add(Frame);

Animation.Add(new bool[col, row]); // Blank frame

Animation.Add(Frame2);

Animation.Add(new bool[col, row]); // Blank frame

Animation.Add(new bool[col, row]); // Blank frame

Animation.Add(new bool[col, row]); // Blank frame

///

/// Creation of a SMILEUSBDevice object

///

SMILEUSBDevice sMILEUSBDevice = new SMILEUSBDevice();

///

/// There are two ways to connect to the device

/// The first option is to connect to a special comport, the second option is to connect to a device automatically

///

///

/// Returns a list of comports on which a SMILEDevice is connected to the computer

///

List<string> ComportList = sMILEUSBDevice.GetComPorts();

///

/// Connect to a specific comport.

/// The input should be the comport as a string.

/// The function returns either true if the connection was successful or false if the connection failed

///

bool ConnectStatus = sMILEUSBDevice.Connect(ComportList[0]);

if (ConnectStatus == true)

{

///

/// Sends a Frame/2d-Boolarray to the device and sets the corresponding Leds

/// If a frame with a mismatched dimension is passed, the frame is scaled to the dimension required for the device

///

sMILEUSBDevice.SendFrame(Frame);

///

/// Disconnects the connection to the device so that it can be accessed by other processes

///

sMILEUSBDevice.Disconnect();

}

///

/// Automatically connects to a fitting comport

/// Input value for autoconnect should be "null"

/// Return: see above

///

ConnectStatus = sMILEUSBDevice.Connect(null); // Automatically connects

if (ConnectStatus == true)

{

///

/// Sets the global voltage of a 8x8 device to the input value. This function has no effect for 16x16 devices.

/// Input value is the voltage in [mV]

///

sMILEUSBDevice.SetVoltage(3300);

sMILEUSBDevice.SendFrame(Frame2); // See above

sMILEUSBDevice.SetVoltage(2800); // See above

///

/// Uploads an animation to the device

/// The input is the animation/framelist and the refreshrate of a frame in [µs]

///

sMILEUSBDevice.UploadAnimation(Animation, 10000);

///

/// Wait 10s so the animation can be seen

///

System.Threading.Thread.Sleep(10000);

///

/// Stops the uploaded animation

///

sMILEUSBDevice.StopAnimation();

sMILEUSBDevice.Disconnect(); // See above

sMILEUSBDevice.Dispose();

}