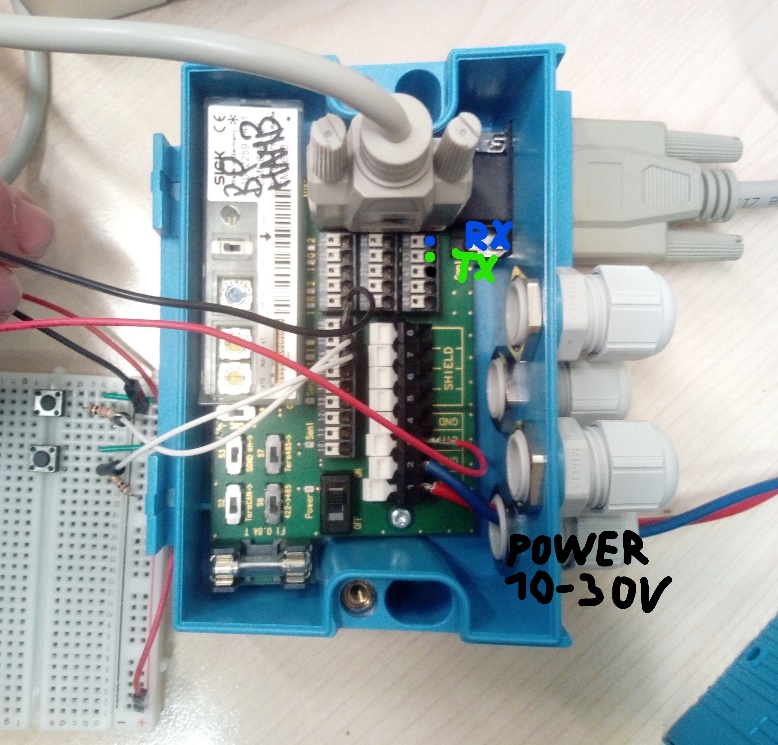
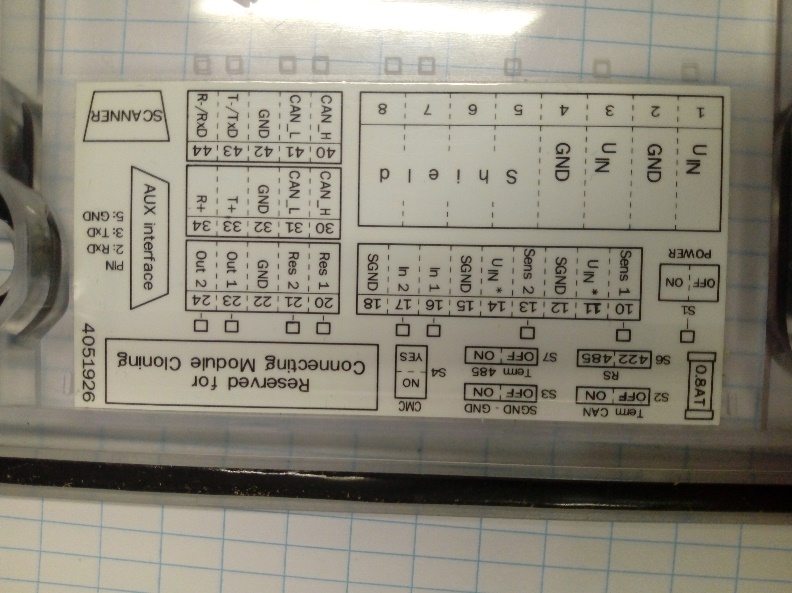
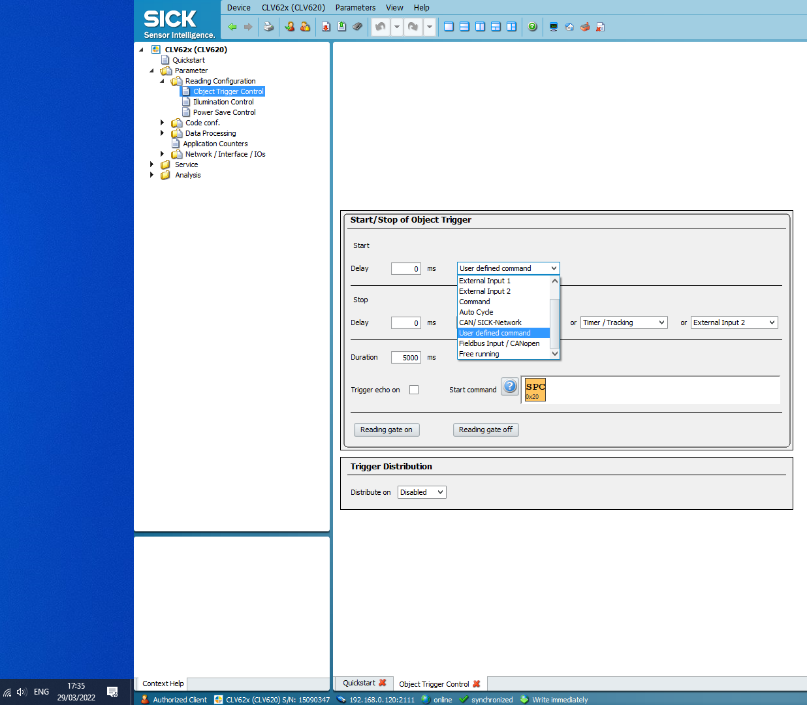
**SOPAS SERIAL**   
For the CLV260-001 Barcode scanner

In this document I will explain how to connect and read data for your scanner using serial communication.

If you don’t have the Scanner connecter with SOPAS engineering tool , its explained in SOPAS 101.

For those who are ready to go you will need a serial to USB cable and a crossed DB9 cable.  
So my setup is the following  
****

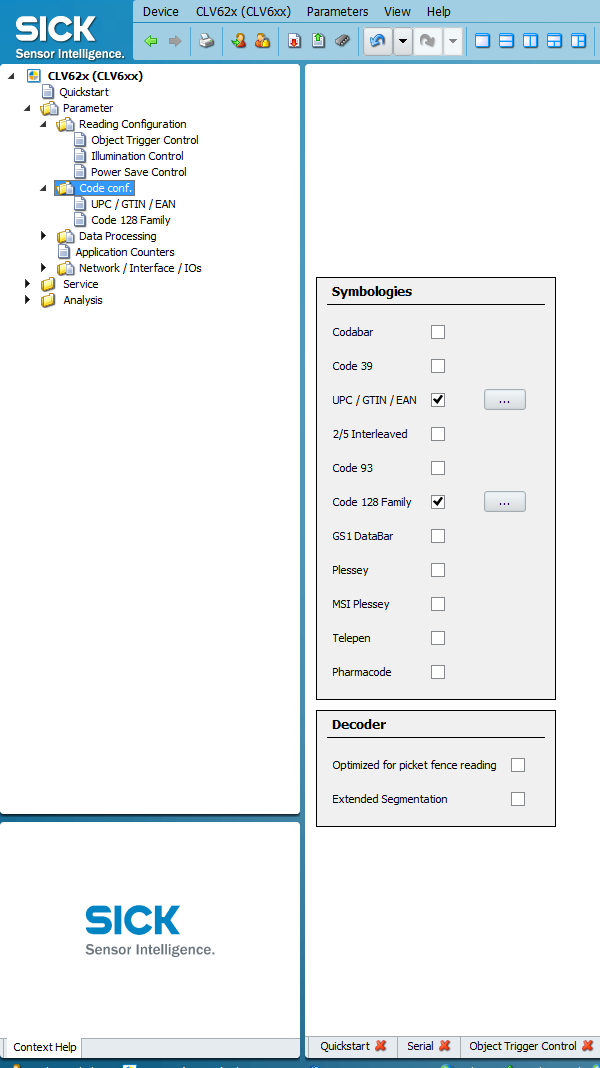
USB to female DB9 should do the job as well , of course if our pc has a DB9 port no USB needed.  
  
This is for the AUX port on the control module , if you want to use the Tx-Rx pins on this module this is possible as well , this way you can control the baud rate and parity. (**you CAN’T use Arduino** for this , Arduino uses TTL serial communication, we use RS232).

Now for the setting.  
First Object Trigger Control  
In other words when does the scanner have to start and stop scanning.

External Input 1,2 These pins can be found on the control module on pin 16 and 17 , they are used by putting a high voltage on them.  
  
Auto cycle you just set up de delay and it will run forever.

The last important one is User defined command. Simply put ,what input do I have to read to start scanning

I will use a space character (0x20) to begin scanning and set stop to timer 5sec or good read.



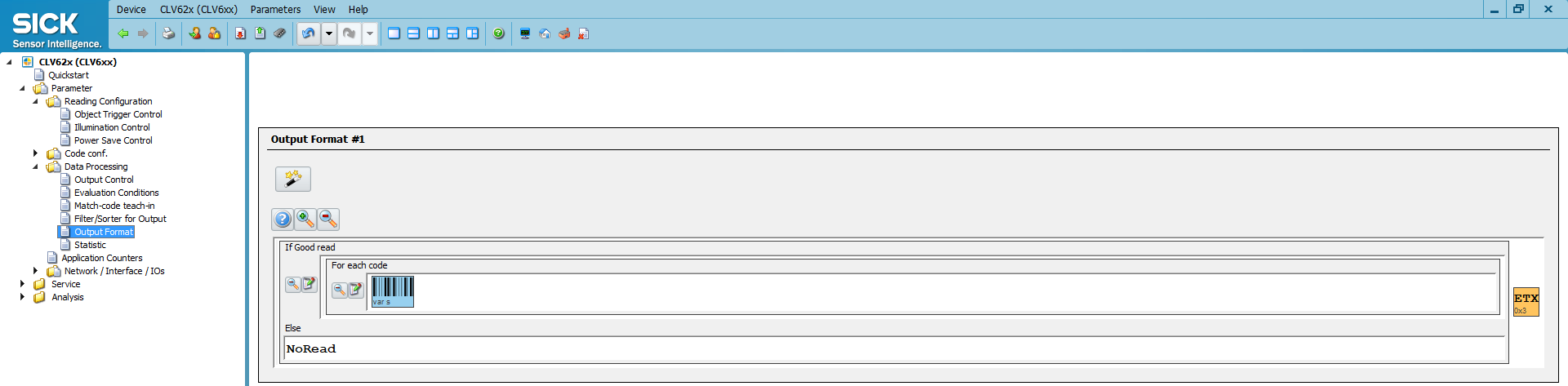
Code Configuration:

If you are in Europe the most important one is EAN this one is found on almost all food and household items.

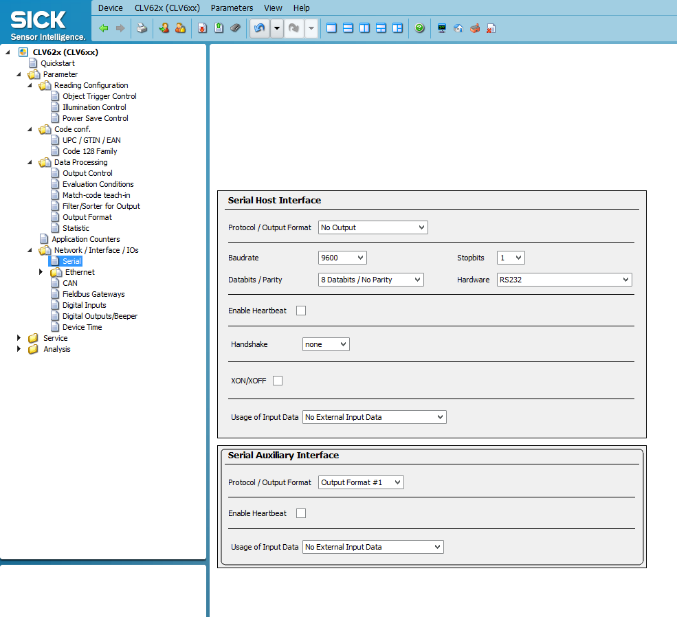
One witch I can find on electronic device’s is C128.

If you cant determent what type of barcode you have go the QuickStart and select everything once you get a successful scan you can see what type you have.  
  
The reason I do not select everything is to make it run faster and less chance to scan a barcode wrong.

Next is a output format , What does the scanner need to send after a good or no read.



I will be removing STX the start byte so it just sends the barcode and a stop byte.  
You can send all kinds of information like , the type , the length , by pressing the green plus sing.

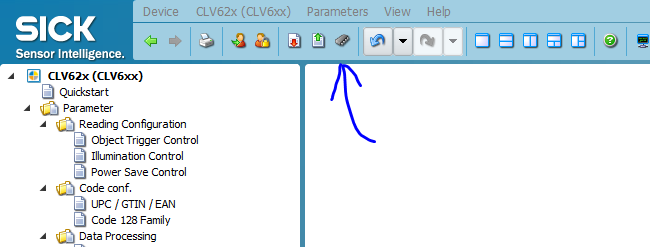
The most important one of all Network > Serial .

Serial Host Interface, this used when using the TX and RX pins on the control module.  
Make sure the settings are the same as the device/application on the other side.

At Output format you can select the format you previously made.

Serial Auxiliary Interface is for the AUX port on the control module , This port has fixed settings :  
Baudrate: 56000 Stopbits: 1

Databits : 8 Parity : none Hardware:RS232



If you are satisfied whit the setting you can put them one your scanner so when you power the scanner again it will have these settings.

Now for the software I will show you 2 ways , simple serial terminal and a wpf C# program.

Both can be found in the GitHub repository.

For the terminal select the correct COM port and baud rate, after connecting it should print the result in HEX DEC or ASCII what suits you the best.  
Afbeelding met tekst

Automatisch gegenereerde beschrijving

If you have a start scanning command you can sent it in this bottom line.  
  
Now for the WPF, it comes down to the same principals. The setup.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

The DataReceive Intrupt , (indata != 3) is to clarify it’s the end of the incoming data since the last ting the scanner send is a stop byte.

Afbeelding met tekst

Automatisch gegenereerde beschrijving



Next is the button ,rather complex for no reason.  
MySerial.Write (“ “) sins the space character is my start scanning signal.   
And while (SerailData == “NULL”) so its in a loop until new data has come in.  
The Thread.Sleep and Dispatcher are to update the screen and allow the application to do something else.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

This is the end , for questions or correction make a issue request in github.