Through my work I've been dealing with PLCs for the past ~15 years or so.
Mostly Schneider Premium PLC's running PL7, but also controllers running CoDeSys 2.3
Various flavours of vendor modified CoDeSys based programming enviroments has also found their way to my table, for example Schneider SoMachine, Wago E!Cockpit and Beckhoff Twincat 2.
I've also dabbled in Unity, but a decision was made to move away from Schneider to CoDeSys 3.5 based devices when the EOL for the premium platform was announced.
On 3.5 I've found myself messing with devices from Wago(PFC200 controllers with Codesys 3 runtime), Intercontrol and IFM, and of course the Raspberry Pi with a Codesys 3 runtime, but this is mostly for PoCs.
The transition from PL7 to Codesys 3 has been a challenging but a rewarding experience, you're moving away from programming based on a simple schematic understood by an electrician/technician to a full-fledged object-oriented language.
Trying to utilize the advantages of an object-oriented language while retaining the simplicity of a ladder program in PL7 is a difficult task, but I find myself constantly getting new ideas, and developing new simpler methods for problems that wasn't always easy to fix in PL7.
For the moment this blog only contains the instructions for installing PL7 natively on a Windows 10 x64 system – a problem many has encounted.
Schneiders answer has usually been "Upgrade to Unity", while this option is a perfectly valid choice for new deployments, it is not a great idea if you just need a small fix/modification in a running system.
In my spare time I like to dabble with PCs, I am also responsible for a small CS:GO community called uber1337.dk.
My main interest is setting up and maintaining the linux game servers for the community.
The servers are now a meeting ground for young people interested in playing CS:GO.