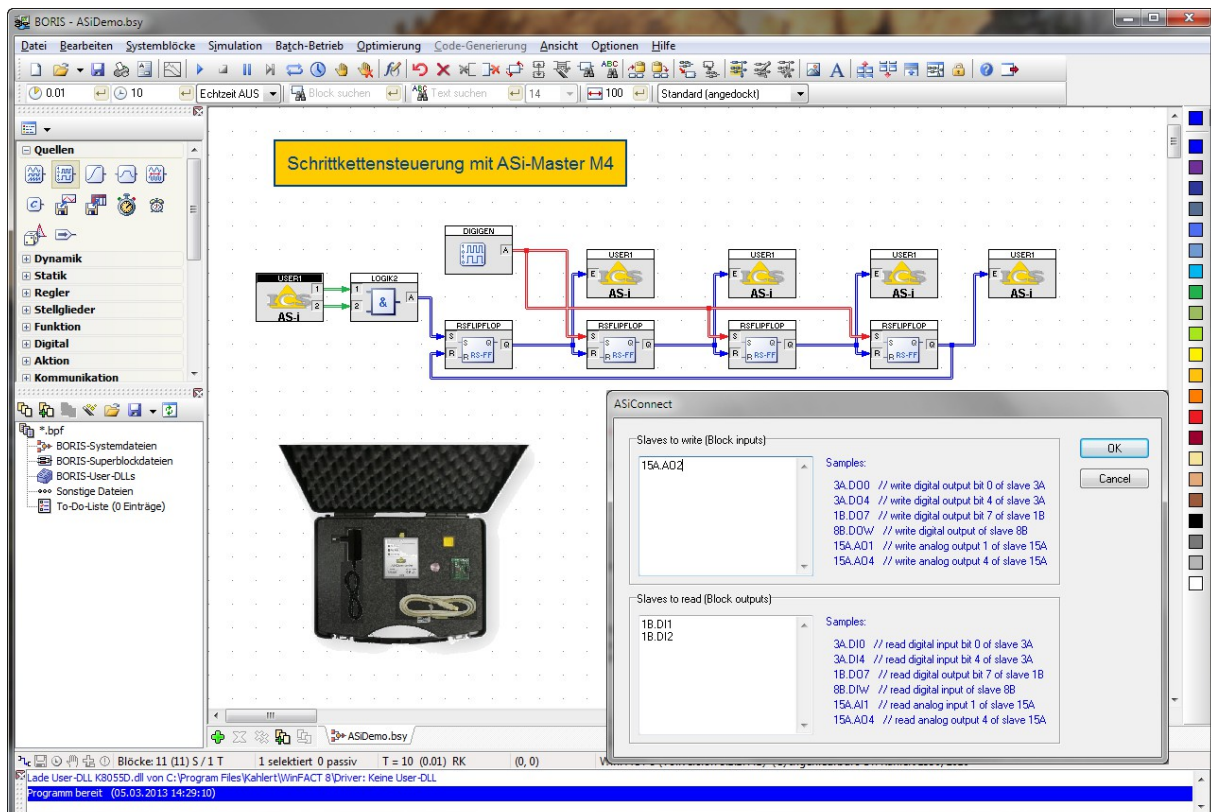


ASiConnect in einem Simulationssystem

Das Blockorientierte Simulationssystem BORIS des Ingenieurbüros Dr. Kahlert in Hamm ist ein insbesondere in der Steuer- und Regelungstechnik weit verbreitetes Werkzeug zum Entwurf und zur Simulation von Automatisierungs- und Regelungssystemen. Einsatz findet es einerseits in der industriellen Forschung und Entwicklung, andererseits aber auch im Rahmen der Ausbildung an Hochschulen und Berufskollegs.

Auf Basis der ASiConnectC-DLL der Fa. ICS wurde ein ASiConnect-Treiber entwickelt, mit dem BORIS nun an ein AS-Interface-Netzwerk angebunden werden kann. Die Schnittstelle zu den Sensor- und Aktuator-Slaves bildet dabei der M-4 Master von ICS, der speziell für Ausbildungszwecke auch in einem preisgünstigen Starter-Kit erhältlich ist. Der ASiConnect-Treiber erlaubt das Lesen und Schreiben sowohl der digitalen als auch der analogen Slaves. Die Konfiguration der Slaves ist dabei in flexibler Weise über einen entsprechenden Parameterdialog möglich. BORIS kann somit einerseits beispielsweise zur Messdatenaufnahme, -auswertung und -weiterverarbeitung genutzt werden, kann aber auch für komplexe Steuerungs- und Regelungsstrategien herangezogen werden. Das Steuerprogramm wird dabei komplett in der komfortablen Entwicklungsumgebung von BORIS aufgebaut, getestet und anschließend im Echtzeitbetrieb gestartet. Der ASi-Master ist dann lediglich für den Datenaustausch zwischen BORIS und den ASi-Slaves zuständig. Das System wird bereits sehr erfolgreich beispielsweise im Rahmen der Fortbildung von Automatisierungstechnikern am Berufsförderungswerk in Dortmund eingesetzt.



The screenshot displays the BORIS simulation environment. The main workspace shows a ladder logic diagram titled "Schrittkettensteuerung mit ASi-Master M4". The diagram includes several blocks: "AS-i" (Master), "LOGN-2" (AND gate), "RSFLIPFLOP" (RS flip-flop), and "USER1 AS-i" (Slave). The diagram is connected to a physical ASiMaster M4 device shown in a carrying case. An "ASiConnect" configuration dialog is open, showing the following settings:

Slaves to write (Block inputs)	Slaves to read (Block outputs)
15A.A02	1B.D11 1B.D12
Samples: 3A.D00 // write digital output bit 0 of slave 3A 3A.D04 // write digital output bit 4 of slave 3A 1B.D07 // write digital output bit 7 of slave 1B 8B.D0W // write digital output of slave 8B 15A.A01 // write analog output 1 of slave 15A 15A.A04 // write analog output 4 of slave 15A	Samples: 3A.D10 // read digital input bit 0 of slave 3A 3A.D14 // read digital input bit 4 of slave 3A 1B.D07 // read digital output bit 7 of slave 1B 8B.D1W // read digital input of slave 8B 15A.A11 // read analog input 1 of slave 15A 15A.A04 // read analog output 4 of slave 15A