

Dokumentation S5Connector



Stand 20.02.2015

S. Rothenbacher GmbH

Automation Industrieelektronik GmbH

Zeppelinstraße 16
D-89160 Dornstadt, Germany

Phone +49-7348-201208

Fax +49-7348-201382

web www.Rothenbacher-GmbH.de

e-mail Info@Rothenbacher-GmbH.de

Inhaltsverzeichnis

1 Allgemeines	4
1.1 Support.....	4
1.2 Projektunterstützung	4
1.3 Lieferumfang	4
1.4 Einschränkung der Demo Version.....	5
1.5 Lizenzvereinbarungen.....	5
1.6 Kompatibilität Betriebssystem.....	7
1.6.1 Windows X86 und X64 /ARM / MIPS.....	7
1.6.2 Linux / MAC / ARM.....	7
2 Grundlegendes zur Anwendung	8
2.1 Der Namespace.....	8
2.2 Die Basisklasse.....	8
2.3 Die Überladung der Basisklasse.....	9
3 Verbindungsaufbau	10
3.1 S5 Protokoll Kompatibilität.....	10
3.2 S5 AS511 (RS232) Verbindungsaufbau.....	11
3.3 S5/S7 RK512 (RS232) Verbindungsaufbau.....	12
4 AG Funktionen	13
4.1 AG_Read	13
4.2 AG_Write.....	14
4.3 Speicherbereich	16
4.4 Datentypen.....	17
5 PG Funktionen	18
5.1 PG_Start.....	18
5.2 PG_Stop.....	18
6 Fehlerbehandlung	19
6.1 Exception Handling.....	19
6.2 Error List.....	19

1 Allgemeines

Unser Softwarepaket S5Conneector besteht aus einer einzigen .Net Dll (Bibliothek).

Die dll ist für sämtliche Windows Plattformen entwickelt worden. Durch das Linux Mono Projekt kann die dll auch unter Linux verwendet werden.

S7-200[®], S7-300[®], S7-400[®], S7-1200[®], HMI[®], WINCC[®], PCS7[®], STEP[®] und SIMATIC[®] sind eingetragene Warenzeichen der Siemens AG,

1.1 Support

Haben Sie Fragen oder Probleme mit der Installation oder der Anwendung des S7Conectors, so wenden Sie sich bitte an unseren Support. Sie erreichen ihn entweder telefonisch unter (07348) 20 12 08 oder per E-Mail über support@rothenbacher-gmbh.de. Schicken Sie uns Ihre Fragen oder die Problembeschreibung mit der von ihnen verwendeten .net version und Betriebssystemversion.

1.2 Projektunterstützung

Kontaktieren Sie uns, wenn Sie einen personellen Engpass oder einfach Bedarf an kompetenter Projektunterstützung haben. Gerne realisieren wir für Sie Projekte zum Festpreis. Setzen Sie sich im Bedarfsfalle einfach mit uns in Verbindung, wir erstellen Ihnen gerne ein unverbindliches Angebot.

1.3 Lieferumfang

Der S7Connector wird als Zip Datei geliefert. In dieser Zip Datei sind alle Basisprogramme, Beispiele und die Demoversion der DLL enthalten.

S5Conector	
Dll S5Connector	.net Dll zum Einbinden in Visuel Studio 2008,2010,2013

1.4 Einschränkung der Demo Version

In der Demoversion stehen Ihnen sämtliche Funktionen der Vollversion zur Verfügung. Es erscheint nur beim Starten ein kleiner Hinweis, dass es sich um eine Demoversion des S5Connectors handelt.

1.5 Lizenzvereinbarungen

EULA / Endbenutzerlizenzvereinbarung

Diese Endbenutzerlizenzvereinbarung (englisch: End User License Agreement - im folgenden als EULA abgekürzt) enthält die Bedingungen und Konditionen bezüglich der Verwendung dieser SOFTWARE (wie unten definiert). Diese EULA enthält Beschränkungen Ihrer Rechte in Bezug auf diese SOFTWARE. Sie sollten diese EULA sorgfältig lesen und als wichtiges Merkmal dieser SOFTWARE behandeln.

1. Vereinbarung zwischen Ihnen und S.Rothenbacher GmbH

Diese EULA ist eine gesetzlich bindende Vereinbarung zwischen Ihnen und der Firma S.Rothenbacher GmbH. Sie beabsichtigen, gesetzlich an diese EULA in demselben Umfang gebunden zu werden, als ob Sie und die Firma S.Rothenbacher GmbH diese EULA physisch unterschreiben würden. Indem Sie diese SOFTWARE installieren, kopieren oder anderweitig nutzen, erklären Sie sich einverstanden, an die in dieser EULA enthaltenen Bedingungen und Konditionen gebunden zu sein. Wenn Sie nicht mit allen Bedingungen und Konditionen dieser EULA einverstanden sind, dürfen Sie die SOFTWARE nicht installieren oder benutzen.

2. Definition von "SOFTWARE"

Diese EULA regelt die Verwendung von Software-Produkten der Firma S.Rothenbacher GmbH. (beigefügt oder anderweitig verfügbar) durch Sie - einzeln und kollektiv als "SOFTWARE" bezeichnet. Der Begriff "SOFTWARE" beinhaltet, neben dem von der Firma S.Rothenbacher GmbH. gelieferten Umfang:

- 1) Alle Revisionen, UPDATES und/oder UPGRADES hierzu
- 2) Alle Daten, Bilder, ausführbare Dateien, Datenbanken, Datenbanksysteme, Computersoftware oder ähnliche Elemente, die normalerweise mit Computersoftware-Produkten verteilt oder verwendet werden
- 3) Alle damit in Verbindung stehenden Datenträgern, Dokumentationen (beinhaltet physische, elektronische und online verfügbare Dokumente) und gedruckte Materialien.

3. Copyright

Die SOFTWARE gehört S.Rothenbacher GmbH. und/oder ihren Lizenzgebern und wird von Copyrightgesetzen und internationalen Verträgen geschützt. Sie dürfen den Copyright-Hinweis aus keiner Kopie der SOFTWARE entfernen.

4. Einräumung einer Lizenz

Die SOFTWARE wird Ihnen nicht verkauft. Stattdessen wird die SOFTWARE auf einer nicht exklusiven Grundlage an Sie - und nur an Sie - zum Gebrauch unter den Bestimmungen dieser Vereinbarung lizenziert. S.Rothenbacher GmbH. behält alle Titel und Eigentumsrechte an der SOFTWARE sowie alle Rechte, die Ihnen nicht ausdrücklich gewährt werden. Solange Sie keine Lizenz für die SOFTWARE erhalten und installiert haben, läuft die SOFTWARE im Demomodus und ist in ihrer Funktionalität eingeschränkt oder funktioniert eines Tages nach der Installation nicht mehr. Für Details zum Demomodus lesen Sie bitte die der SOFTWARE beigefügten Dokumentation.

5. Verwendung nur an einem Einzelarbeitsplatz

Die SOFTWARE darf nur an einem einzigen Arbeitsplatz genutzt werden. Ein Arbeitsplatz ist definiert durch die Kombination eines physischen oder virtuellen Computers (oder einer Session auf einem Terminal-Server) und einer Person. Sie dürfen die SOFTWARE auf jedem Computer eines Arbeitsplatzes installieren (z.B. Arbeitsstation und Notebook), wenn sichergestellt ist, dass die SOFTWARE zu keiner Zeit von mehr als einer Person verwendet wird.

6. Eine Archivierungskopie

Sie dürfen nur eine einzige Sicherungskopie anfertigen, die ausschließlich zu Archivierungszwecken genutzt werden darf. Diese darf nicht an Dritte weiter gegeben werden.

7. Dekompilierung, Entassemblierung oder Zurückentwicklung

Sie erkennen an, dass die SOFTWARE Betriebsgeheimnisse und andere Eigentumsinformationen der Firma S.Rothenbacher GmbH. und/oder ihren Lizenzgebern enthält. Sie dürfen die SOFTWARE nicht dekompileieren, entassemblieren oder auf irgendeine Art und Weise zurückentwickeln (Reverse Engineering) - noch sich mit anderen Aktivitäten beschäftigen, um die zugrunde liegenden Informationen zu erhalten, die für den Benutzer während der normalen Verwendung der SOFTWARE nicht sichtbar sind.

8. Aktualisierungen (Updates und Upgrades)

Diese Lizenz räumt Ihnen kein Recht auf irgendwelche Erweiterungen, Fehlerbeseitigungen, Programmkorrekturen oder Aktualisierungen der SOFTWARE ein - noch irgendwelche Support-Dienstleistungen. UPDATES (Aktualisierungen) sind definiert als neue Versionen der SOFTWARE, in denen sich die Hauptversionsnummer nicht geändert hat (die Hauptversionsnummer ist die erste Zahl der Versionskennzeichnung des Produktes. Beispiel: Die Hauptversionsnummer von "1.2.3.4" ist 1). Wird die Hauptversionsnummer geändert, dann wird diese neue Version der SOFTWARE als UPGRADE definiert. Verfügbare UPDATES werden kostenlos von der Firma S.Rothenbacher GmbH. über die jeweilige Produkt-Website oder auf der Hauptwebsite (www.rothenbacher-gmbh.de) zum Download zur Verfügung gestellt (Kosten für Ihre Internet-Verbindung und/oder Kosten des Transfers selber sowie jegliche Kosten, die in Verbindung mit dem Erhalt des UPDATES stehen, müssen von Ihnen übernommen werden).

9. Beendigung

Die Ihnen erteilte Lizenz ist bis zur Beendigung gültig. Die Beendigung kann zu jeder Zeit durch die Rückgabe der SOFTWARE (inkl. aller Kopien davon) an die Firma S.Rothenbacher GmbH. stattfinden. Außerdem wird die Gültigkeit Ihrer Lizenz automatisch beendet (auch ohne einen Hinweis von der Firma S.Rothenbacher GmbH.), wenn Sie einer Bedingung oder Kondition dieser Vereinbarung zuwiderhandeln. Sie stimmen bei einer solchen Beendigung zu, sämtliche Bestandteile der SOFTWARE (inkl. Kopien davon) an die Firma S.Rothenbacher GmbH. zurückzugeben. Bei Beendigung kann die Firma S.Rothenbacher GmbH. die ihr per Gesetz zustehenden Rechte durchsetzen. Die Bedingungen und Konditionen dieser Vereinbarung, die die Eigentumsrechte der Firma S.Rothenbacher GmbH. schützen, bleiben auch nach Beendigung in Kraft.

10. Haftungsausschluss

Die Firma S.Rothenbacher GmbH. garantiert nicht, dass die in der SOFTWARE enthaltenen Funktionen Ihre Anforderungen erfüllt und/oder dass der Betrieb der SOFTWARE ununterbrochen, fehlerfrei oder frei von arglistigem Code ist ("arglistiger Code" bezeichnet jeglichen Programm-Code, der entwickelt wurde, um andere Computer-Programme und/oder Computer-Daten zu kontaminieren, Computerbetriebsmittel zu verbrauchen, Daten zu ändern / zu löschen / aufzuzeichnen oder zu übermitteln - oder in irgendeiner anderen Art den Normalbetrieb von Computern, Rechnersystemen oder Computernetzwerken zu stören, einschließlich Viren, trojanischen Pferden, Droppern, Würmern, Logik-Bomben und Ähnliches).

Diese SOFTWARE wird WIE SIE IST geliefert – ohne irgendwelche Garantien. S.Rothenbacher GmbH ist nicht verpflichtet, UPDATES, UPGRADES oder technischen Support für diese SOFTWARE bereitzustellen. Die Firma S.Rothenbacher GmbH übernimmt außerdem keine Haftung für die Genauigkeit sämtlicher von S.Rothenbacher GmbH oder von Dritten zur Verfügung gestellten Informationen oder für irgendwelche Schäden, die direkt oder indirekt durch Aktionen oder Unterlassungen aufgrund dieser Informationen entstehen.

Sie übernehmen die volle Verantwortung für die Auswahl der SOFTWARE, um Ihre zukünftigen Ergebnisse zu erreichen, sowie für die Installation, Benutzung und die Ergebnisse, die Sie von der SOFTWARE erhalten. Weiterhin übernehmen Sie das volle Risiko in Bezug auf Qualität und Leistung der SOFTWARE.

Sollte sich die SOFTWARE als fehlerhaft erweisen, übernehmen Sie (und nicht die Firma S.Rothenbacher GmbH oder ihre Distributoren oder Händler) die gesamten Kosten für alle notwendigen Service-, Reparatur- und/oder Korrektur-Leistungen.

In keinem Fall haftet die Firma S.Rothenbacher GmbH oder ihre Lizenzgeber für direkte, indirekte, beiläufige oder besondere Schäden noch für Folgeschäden oder irgendwelche Verluste, entgangene Gewinne, entgangene Einnahmen, entgangene Einsparungen oder für entstandene Datenverluste, die durch oder in Verbindung mit dieser SOFTWARE oder dieser Vereinbarung entstehen, selbst wenn die Firma S.Rothenbacher GmbH oder ihre Lizenzgeber über die Möglichkeit solcher Schäden unterrichtet wurde. In jedem Fall ist die Haftung auf den für die SOFTWARE gezahlten Betrag beschränkt, unabhängig von der Art des Schadenfalls.

11. Schlussklausel

Diese Vereinbarung bindet Sie wie auch Ihre Angestellten, Arbeitgeber, Auftragnehmer und Agenten sowie alle Nachfolger und Bevollmächtigten. Diese Vereinbarung ist die gesamte Vereinbarung zwischen uns und hat Vorrang vor allen anderen Absprachen und Vereinbarungen. Falls eine Bedingung oder Kondition dieser Vereinbarung ungültig ist oder wird, bleiben alle anderen Bedingungen und Konditionen dieser Vereinbarung davon unberührt und behalten ihre Gültigkeit. In einem solchen Fall verpflichten sich beide Parteien, die ungültig Bedingung und/oder Kondition durch eine gültige Bedingung und/oder Kondition zu ersetzen, die in ihrer rechtlichen, wirtschaftlichen und technischen Bedeutung möglichst gleichkommt. Gerichtsstand ist Deutschland.

1.6 Kompatibilität Betriebssystem

1.6.1 Windows X86 und X64 /ARM / MIPS

		32 Bit	64 Bit
Windows 95	.Net Framwork / Mono Framwork	-	-
Windows Me	.Net Framwork / Mono Framwork	-	-
Windows 98	.Net Framwork / Mono Framwork	X	-
Windows NT Workstation 4.0 (alle Versionen - Service Pack 6a erforderlich)	.Net Framwork / Mono Framwork	X	-
Windows 2000 Professional	.Net Framwork / Mono Framwork	X	-
Windows XP Home Edition	.Net Framwork / Mono Framwork	X	X
Windows XP Professional	.Net Framwork / Mono Framwork	X	X
Windows Vista	.Net Framwork / Mono Framwork	X	X
Windows 7	.Net Framwork / Mono Framwork	X	X
Windows 8	.Net Framwork / Mono Framwork	X	X
Windows 2003 Server	.Net Framwork / Mono Framwork	X	X
Windows 2008 Server	.Net Framwork / Mono Framwork	X	X
Windows CE .NET 4.0 – 4.2	.NET Compact Framework	X	-
Windows CE 5.0	.NET Compact Framework	X	-
Windows Embedded CE 6.0	.NET Compact Framework	X	-

1.6.2 Linux / MAC / ARM

		32 Bit	64 Bit
Ubuntu	Mono Framwork	X	X
Red Hat	Mono Framwork	X	X
OpenSuse	Mono Framwork	X	X
Debian	Mono Framwork	X	X
LinuxMint	Mono Framwork	X	X
Knoppix	Mono Framwork	X	X
Raspberry PI - Raspbian	Mono Framwork	X	-
CubieBoard 2	Mono Framwork	X	-
Apple OSX	Mono Framwork	X	X

Und sicher viele mehr

2 Grundlegendes zur Anwendung

2.1 Der Namespace

Alle im folgenden verwendete Funktionen sowie Variablen des S5-Connectors befinden sich im Namespace: „Ro.Communication.Protocol“.

Die Dokumentation bezieht sich darauf das der Namespace dem Projekt bekannt gegeben wurde.

Beispiel für C#

```
using Ro.Communication.Protocol;
```

2.2 Die Basisklasse

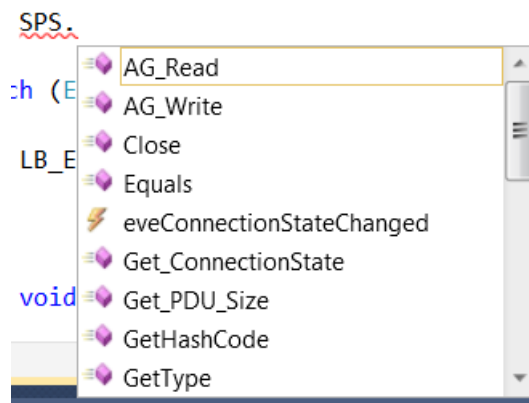
Die Basisklasse „S5.Base“ stellt den Ausgangspunkt der Projektierung dar. Sie beinhaltet alle Funktionsaufrufe um z.B. Daten aus einem DB zu lesen bzw. zu schreiben.

Hierfür wird eine Instanz der Basisklasse benötigt.

Beispiel für C#

```
private S5.Base s5SPS;
```

Der Zugriff auf die Funktionen erfolgt nun über diese Instanz:



2.3 Die Überladung der Basisklasse

Diese Basisklasse wird durch den Verbindungsaufbau über eine der abgeleiteten Klassen „überschrieben“ und nutzt dadurch die physikalischen und logischen Verbindungsparameter, die bei der Verbindung angegeben werden.

Mögliche Verbindungsklassen zur Überladung sind:

- [S5.RK512](#)
- [S5.AS511](#)

Eine genaue Anleitung zu den einzelnen Verbindungsarten finden sie unter „3 Verbindungsaufbau“

3 Verbindungsaufbau

Bevor Sie auf Daten der SPS Steuerung zugreifen können muss die Kommunikation initialisiert werden. Stellen Sie die gewünschte Kommunikation über den Aufruf der Verbindungsaufbau-funktionen(3.2,3.3,) ein. Ist der Verbindungsaufbau erfolgreich verlaufen können Sie mit den Funktionen (4.1,4.2) Daten aus der Steuerung lesen. Die Fehlerbehandlung der einzelnen Funktionen können Sie über die `try{ }Code... catch (Exception E)` abfangen. Eine Liste der möglichen Fehler und ihrer Ursachen entnehmen Sie dem Kapitel (6).

3.1 S5 Protokoll Kompatibilität

Simatic S5-90U	CPU90U / 8-Bit CPU
Simatic S5-95U	CPU95U / 8-Bit CPU
Simatic S5-100U	CPU100 / 8-Bit CPU CPU102 / 8-Bit CPU CPU103 / 8-Bit CPU
Simatic S5-115U	CPU941 / 8-Bit CPU CPU942 / 8-Bit CPU CPU943 / 8-Bit CPU CPU944 / 8-Bit CPU CPU945 / 8-Bit CPU mit 20 bit Adressbereich
Simatic S5-135U	CPU922 / 16-Bit CPU CPU928 / 16-Bit CPU CPU928 / 16-Bit CPU
Simatic S5-135U	CPU921 / 16-Bit CPU (nur über RK512)
Simatic S5-155U	CPU948 / 20-Bit CPU

3.2 S5 AS511 (RS232) Verbindungsaufbau

Die Überladung durch „S5.AS511“ referenziert die Instanz der Basisklasse „S5.Base“ mit einer S5-Kommunikation über AS511. Bei AS511 handelt es sich um Schnittstellen mit TTY-Physik, über die ein Siemens-internes Protokoll abgewickelt wird. Das AS511 Protokoll Arbeitet mit folgender Com Port Einstellung: 9600 Baud, Parität gerade, kein Handshake, 8 Datenbits, 1 Stop Bit

	Beschreibung
Position	AS511(string ComPort, int TimeoutInMS)
Codeauszug C#	<pre> Beispiel für C# try { SPS = new S5Connector.AS511("COM1",1000); } catch (Exception e) { // Exception Handling } </pre>
Beschreibung	Die Funktion dient zum Verbindungsaufbau über AS511. Als Rückgabe erhalten Sie eine Instanz auf den S5connector (Beispiel s5SPS). Diese Verbindung kann nur seriell (ausgenommen USB RS232 Wandler) kommunizieren.
Parameter	<ul style="list-style-type: none"> - ComPort : Adresse der seriellen Schnittstelle des Host PCs z.B. "COM1" - TimeoutInMS : Zeitspanne in Millisekunden in der die SPS eine Anfrage beantworten muss, bevor eine "Exception" ausgelöst wird.

3.3 S5/S7 RK512 (RS232) Verbindungsaufbau

Die Überladung durch „S7.RK512“ referenziert die Instanz der Basisklasse „S5.Base“ mit einer S5-Kommunikation über RK512. Mit der Rechnerkopplung RK 512 können zwischen zwei Punkt-zu-Punkt gekoppelten Kommunikationspartnern Datenausgetauscht werden. Beim RK512 Protokoll werden Datenbytes mit der Übertragungsprozedur 3964 oder 3964R übertragen. Diese fügt den Datenbytes Start- und Endezeichen hinzu und führt bei Fehlern eventuelle Wiederholungen durch.

	Beschreibung
Position	PPI(string ComPort, int TimeoutInMS, int BaudRate, int DataBits, System.IO.Ports.StopBits StopBits, System.IO.Ports.Parity Parity, byte ppiAddressPG, byte ppiAddressAG)
Codeauszug C#	<pre>try { SPS5 = new S5Connector.RK512("COM1", 1000, 38400, 8, System.IO.Ports.StopBits.One, System.IO.Ports.Parity.None,255,255); } catch (Exception e) { // Exception Handling }</pre>
Beschreibung	Die Funktion dient zum Verbindungsaufbau über RK512 . Als Rückgabe erhalten Sie ein e Instanz auf den S5connector (Beispiel s5SPS).Diese Verbindung kann nur seriell kommunizieren.
Parameter	<ul style="list-style-type: none"> - ComPort : Adresse der seriellen Schnittstelle des Host PCs z.B. "COM1" - TimeoutInMS :Zeitspanne in Millisekunden in der die SPS eine Anfrage beantworten muss, bevor eine "Exception" ausgelöst wird. - BaudRate : Geschwindigkeit der seriellen Datenübertragung z.B. 38400 - DataBits : Anzahl Datenbits der seriellen Datenübertragung z.B. 8 - StopBits : Anzahl Stopbits der seriellen Datenübertragung z.B 1 - Parity : Parität der seriellen Datenübertragung - coordination_flag : = 255 - CPU_No : = 255

4 AG Funktionen

4.1 AG_Read

Die Funktion „AG_Read“ liest Daten von einer verbundenen SPS.

	Beschreibung
Position	<code>public virtual object AG_Read(Int16 DBNr, Int32 StartAdr, Int16 Count, S7DataTypes RetType, S5MemmoryTypes MemArea)</code>
Codeauszug C#	<pre>try { short DBNr = Convert.ToInt16(TB_DBNr_Read.Text); short Count = Convert.ToInt16(TB_Count_Read.Text); int Start = Convert.ToInt32(TB_StartAddress_Read.Text); object Result = SPS.AG_Read(DBNr, Start, Count, Read_Selected_DataType, Read_Selected_MemmoryType); LB_Result.Items.Clear(); for (int i = 0; i < (Result as Array).Length; i++) LB_Result.Items.Add("VAL" + i.ToString("0000") + " = " + (Result as Array).GetValue(i).ToString()); } catch (Exception E) { // Exception Handling } }</pre>
Beschreibung	(Datablock, Flag, Input, Output, Counter, Timer, Periphery, InstanceDatablock, LocalData, Localdataofpreviousblock) (CHAR, BYTE, WORD, INT, REAL, DWORD, BOOL, S5TIME, TIMEIEC32, COUNTER)
Parameter	<ul style="list-style-type: none"> - DBNr : DB Adresse, von dem gelesen werden soll. Beim Lesen von anderen Speicherbereichen (Merkerbereich, E/A Bereich usw) muss die Adresse immer "0" sein. - StartAdr : Byteadresse, wo der Leseauftrag startet. - Count : Anzahl Daten, die gelesen werden sollen. - RetType : Datenart, die gelesen werden soll.4.4 - MemArea : Speicherbereich, von dem gelesen werden soll.4.3

4.2 AG_Write

Die Funktion „AG_Write“ schreibt Daten in eine verbundene SPS.

	Beschreibung
Position	<pre> public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, string Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, byte[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, UInt16[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, Int16[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, UInt32[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, float[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, S5Time[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, TimeIEC32[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdr, MemmoryTypes MemArea, Counter[] Values) public virtual bool AG_Write(Int16 DBNr, Int32 StartAdrInBits, MemmoryTypes MemArea, bool Value) </pre>
Codeauszug C#	<pre> try { short DBNr = Convert.ToInt16(TB_DBNr_Read.Text); short Count = Convert.ToInt16(TB_Count_Read.Text); int Start = Convert.ToInt32(TB_StartAddress_Read.Text); switch (CB_DataType_Read.Text) { case "BYTE": SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, new byte[] { Convert.ToByte(TB_Value.Text) }); break; case "WORD": case "DATE": SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, new UInt16[] { Convert.ToUInt16(TB_Value.Text) }); break; case "INT": SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, new Int16[] { Convert.ToInt16(TB_Value.Text) }); break; case "REAL": SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, new float[] { float.Parse(TB_Value.Text) }); break; case "DWORD": case "TIME_OF_DAY": SPS.AG_Write(DBNr, Start, Read_Selected_MemmoryType, new UInt32[] { Convert.ToUInt32(TB_Value.Text) }); break; } } </pre>

	<pre> case "S5TIME": S7Connector.S5Time[] s5Times = new S7Connector.S5Time[2]; s5Times[0] = new S7Connector.S5Time(new TimeSpan(0, 0, 0, 10, 0)); s5Times[1] = new S7Connector.S5Time(new TimeSpan(0, 0, 0, 11, 0)); SPS.AG_Write(DBNr, Start, Read_Selected_MememoryType, s5Times); break; case "TIMEIEC32": S7Connector.TimeIEC32[] s7Times = new S7Connector.TimeIEC32[1]; s7Times[0] = new S7Connector.TimeIEC32(new TimeSpan(0, 0, 2, 1, 100)); SPS.AG_Write(DBNr, Start, Read_Selected_MememoryType, s7Times); break; case "COUNTER": S7Connector.Counter[] s7Conter= new S7Connector.Counter[1]; s7Conter[0] = new S7Connector.Counter(Convert.ToInt16(TB_Value.Text)); SPS.AG_Write(DBNr, Start, Read_Selected_MememoryType, s7Conter); break; case "BOOL": bool Val = false; if (TB_Value.Text.ToLower() == "true") Val = true; SPS.AG_Write(DBNr, Start, Read_Selected_MememoryType, Val); break; } </pre>
Beschreibung	
Parameter	<ul style="list-style-type: none"> - DBNr : DB Adresse, die beschrieben werden soll. Beim Beschreiben von anderen Speicherbereichen (Merkerbereich, E/A Bereich) muss die Adresse immer "0" sein. - StartAdr : Byteadresse, wo der Schreibauftrag startet. - MemArea : Speicherbereich, von dem gelesen werden soll.4.3 - Value(s) : Werte, die geschrieben werden sollen. Beim Schreiben von Boolschen Werten, wird nur ein Wert berücksichtigt. Der Datentyp wird durch die Überlagerung des Parameters automatisch erkannt.

4.3 Speicherbereich

Beim Laden in die CPU werden die Bausteine des S5-Programms in einen Ladespeicher transferiert. Das Betriebssystem generiert daraus ein ausführbares Programm im Arbeitsspeicher (RAM) der CPU. Daneben gibt es den Systempeicher (RAM), der in bestimmte Bereiche unterteilt ist:

	S7-Notation		
Datablock Datenbausteine	DB	Datenbausteine werden im Step5 erstellt in Größe und Datenstruktur. Sie können entweder als (Globale DB) oder sie sind einem bestimmten FB oder SFB zugeordnet (Instanz-DB).	
Input Eingänge	I E	Speicherbereich, der von der SPS jeweils vor Beginn oder am Ende des Zyklus mit den Daten der Ein/Ausgabebaugruppen beschrieben wird.	
Output Ausgänge	Q A		
Flag Merker	F M	Anwender - Speicherbereich	
Timer	T		
Counter, Zähler	C Z		
LocalData Lokaldaten	L	Speicherbereich für temporäre Daten eines Codebausteins. Der Aufbau des jeweiligen Lokaldaten-Stacks ergibt sich aus der zugehörigen Deklarationstabelle	
Periphery Eingangspanperipherie	PE	Speicherbereich, der direkt mit den Daten der Ein-/Ausgabebaugruppen in Verbindung steht. Durch Zugriff auf Daten des Peripheriebereichs können die Ein/Ausgabebaugruppen unabhängig vom OB1-Zyklus erreicht werden.	
Periphery Ausgangspanperipherie	PA		

Auf alle Speicherbereiche kann durch den S5Connector lesend und schreibend zugegriffen werden. Die folgende Tabelle zeigt die Speicheraufteilung. Die Menge der verfügbaren Operanten (d.h. die Größe des Systemspeichers) ist abhängig von der jeweiligen CPU.

4.4 Datentypen

Jeder elementare Datentyp verfügt über einen zugeordneten Speicherplatz mit fester Länge. Der Datentyp BOOL zum Beispiel hat nur ein Bit, ein Byte (BYTE) besteht aus 8 Bits, ein Wort (WORD) sind 2 Bytes (bzw. 16 Bits), ein Doppelwort (DWORD) hat 4 Bytes (bzw. 32 Bits). Die folgende Tabelle zeigt alle vorhandenen elementaren Datentypen:

S7 Datentype	C# Datentype	Beschreibung	Bits	Wertebereichs
BOOL	bool	Einzelnes Bit	1	TRUE FALSE
BYTE	byte	Festpunktzahl	8	0 bis 255
WORD	uint16	Festpunktzahl	16	0 bis 65535
DWORD	uint32	Festpunktzahl	32	
INT	int16	Festpunktzahl	16	-32768 bis 32767
DINT	int32	Festpunktzahl	32	-2147483648 bis 2147483647
REAL	float	Gleitpunktzahl	32	1.0 1.238 1.2E-2 -5E12 100.0 -1000.0 -123E-18
S5TIME	S5Time	Zeitwert Simatic 16	16	S5T#0ms bis S5TIME#2h46m30s
TIME	TimeIEC32	Zeitwert IEC 32	32	-24d20h31m23s647ms bis 24d20h31m23s647ms
CHAR	string	ASCII-Zeichen	8	
DATE			16	
TIME_OF_DAY			32	

5 PG Funktionen

Die PG Funktion (Programmiererätfunktionen) stehen nur bei Verbindungen zu S7 300 & S7 400 zur Verfügung. Bei einer Connection zu einer S7 200 oder S 1200, S7 1500 wird eine Exception ausgelöst.

5.1 PG_Start

	Beschreibung
Position	<code>public virtual bool PG_Start()</code>
Codeauszug C#	<code>SPS.PG_Start();</code>
Beschreibung	Die Funktion „PG_Start“ sendet an die verbundene SPS einen Startbefehl. Wenn die Stellung des Schüsselschalters „START“ ist und kein unbehandelter Fehler ansteht, wechselt die SPS in die Betriebsart „RUN“. Im Programm angesteuerte Ausgänge sind geschaltet.
Parameter	

5.2 PG_Stop

	Beschreibung
Position	<code>public virtual bool PG_Stop()</code>
Codeauszug C#	<code>SPS.PG_Stop();</code>
Beschreibung	Die Funktion „PG_Stop“ sendet an die verbundene SPS einen Stoppbefehl. Die SPS wechselt in die Betriebsart „STOP“. Der Programmbetrieb ist unterbrochen. Alle Ausgänge sind spannungsfrei geschaltet oder werden auf Default-Werte gesetzt.
Parameter	

6 Fehlerbehandlung

6.1 Exception Handling

Kommt es während der Kommunikation oder durch die Kommunikation zu einem Programm Fehler so können diese mittels `try{ }Code... catch (Exception E)` abgefangen werden. Sollte es durch einen Fehler zu einem Verbindungsabbruch kommen oder ist der Fehler ein Verbindungsabbruch. So wird der S7Connector beim nächsten Lese oder Schreibversuch selbständig versuchen die Verbindung wieder aufzubauen.

6.2 Error List

Error = "hardware fault";
Error = "object access not allowed: occurs when access to timer and counter data type is set to signed integer and not BCD";
Error = "Not context";
Error = "address out of range: occurs when requesting an address within a data block that does not exist or is out of range";
Error = "address out of range";
Error = "write data size mismatch";
Error = "object does not exist: occurs when trying to request a data block that does not exist";
Error = "communication link not available";
Error = "negative acknowledge / time out error";
Error = "data does not exist or is locked";
Error = "unknown error";
Error = "wrong interface specified";
Error = "too many interfaces";
Error = "interface already initialized";
Error = "interface already initialized with another connection";
Error = "interface not initialized; this may be due to an invalid MPI address (local or remote ID) or the PLC is not communicating on the MPI network";
Error = "can't set handle";
Error = "data segment isn't locked";
Error = "data field incorrect";
Error = "block size is too small";
Error = "block boundary exceeded";
Error = "wrong MPI baud rate selected";
Error = "highest MPI address is wrong";
Error = "address already exists";
Error = "not connected to MPI network";
Error = "-";
Error = "hardware error";
Error = "hardware error";
Error = "communication link unknown";
Error = "communication link not available";
Error = "MPI communication in progress";
Error = "MPI connection down; this may be due to an invalid MPI address (local or remote ID) or the PLC is not communicating on the MPI network";
Error = "interface is busy";
Error = "not permitted in this mode";
Error = "hardware error";
Error = "access to object not permitted";
Error = "Not context";
Error = "address invalid. This may be due to a memory address that is not valid for the PLC";
Error = "data type not supported";
Error = "data type not consistent";
Error = "object doesn't exist. This may be due to a data block that doesn't exist in the PLC";
Error = "not enough memory on CPU";
Error = "maybe CPU already in RUN or already in STOP";

Error = "serious error";
Error = "wrong PDU (response data) size";
Error = "Not address";
Error = "Step7: variant of command is illegal.";
Error = "Step7: status for this command is illegal.";
Error = "Step7: function is not allowed in the current protection level.";
Error = "syntax error: block name";
Error = "syntax error: function parameter";
Error = "syntax error: block type";
Error = "no linked data block in CPU";
Error = "object already exists";
Error = "object already exists";
Error = "data block in EPROM";
Error = "block doesn't exist";
Error = "no block available";
Error = "block number too large";
Error = "coordination rules were violated";
Error = "protection level too low";
Error = "protection violation while processing F-blocks; F-blocks can only be processed after password input";
Error = "invalid SSL ID";
Error = "invalid SSL index";
Error = "information doesn't exist";
Error = "diagnosis: DP Error";
Error = "this job does not exist";
Error = "maybe invalid BCD code or Invalid time format";
Error = "wrong ID2, cyclic job handle";
Error = "API function called with an invalid parameter";
Error = "timeout, check RS232 interface";
Error = "Lack of resources in driver or in the library // Ressourcenengpaß im Treiber oder in der Library";
Error = "Configuration error // Konfigurationsfehler";
Error = "Job not currently permitted // Auftrag zur Zeit nicht erlaubt";
Error = "Parameter error";
Error = "Device already/not yet open // Gerät bereits/noch nicht geöffnet.";
Error = "CP not reacting // CP reagiert nicht";
Error = "Error in firmware // Fehler in der Firmware";
Error = "Lack of memory for driver // Speicherengpaß im Treiber";
Error = "No message // Keine Nachricht vorhanden";
Error = "Error accessing application buffer // Fehler bei Zugriff auf Anwendungspuffer";
Error = "Timeout expired // Timeout abgelaufen";
Error = "Maximum number of logons exceeded // Die maximale Anzahl an Anmeldungen ist überschritten";
Error = "Job aborted // Der Auftrag wurde abgebrochen";
Error = "An auxiliary program could not be started // Ein Hilfsprogramm konnte nicht gestartet werden";
Error = "No authorization exists for this function // Keine Autorisierung für diese Funktion vorhanden";
Error = "Initialization not yet completed // Initialisierung noch nicht abgeschlossen";
Error = "Function not implemented // Funktion nicht implementiert";
Error = "CP name does not exist // CP-Name nicht vorhanden";
Error = "CP name does not exist // CP-Name nicht vorhanden";
Error = "CP name not configured // CP-Name nicht konfiguriert";
Error = "Channel name does not exist // Kanalname nicht vorhanden";
Error = "Channel name not configured // Kanalname nicht konfiguriert";
Error = "Error // Unbekannter Fehler "