



SH ARC-GATEWAY

Programmierhandbuch
Programming Manual



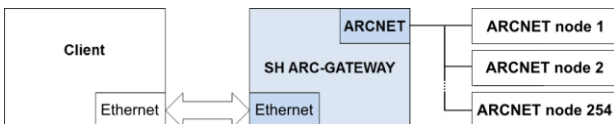
SOHARD
EMBEDDED SYSTEMS

1	Allgemeine Information.....	4
2	Beschreibung des SOHARD-Protokolls.....	4
3	Beispiele zum SOHARD-Protokoll.....	18
4	Beschreibung des AUG-Protokolls.....	20
5	General Information.....	36
6	Description of the SOHARD Protocol.....	36
7	Examples for the SOHARD Protocol.....	50
8	Description of the AUG Protocol.....	52

1 Allgemeine Information

Mit dem SH ARC-GATEWAY kann über das TCP/IP-Protokoll auf daran angeschlossene ARCNET-Netze zugegriffen werden. Das Gerät basiert auf dem Betriebssystem Linux, was die Integration in die IT-Infrastruktur der Kunden und die nachfolgende Administration einfach gestaltet.

Die typische Einsatzumgebung des SH ARC-GATEWAY ist auf folgendem Bild dargestellt:



Die nachfolgenden Kapitel beschreiben die Applikationsprotokolle von SOHARD und der ARCNET User Group e.V. (AUG).

Die Konfiguration des SH ARC-GATEWAY ist im Benutzerhandbuch beschrieben.

Wir behalten uns Änderungen und Erweiterungen vor.

2 Beschreibung des SOHARD-Protokolls

Die Kommunikation mit dem SH ARC-GATEWAY erfolgt über TCP/IP-Sockets. Das Protokoll besteht aus den unten beschriebenen Pakettypen, wobei jedes Datenpaket einen Header aus

- Protokollkennung,
- Paketkennung und
- Paketlänge hat.

Das Feld „Paketlänge“ ist 2 Byte groß und enthält die Länge des Pakets **ohne den Header** in Bytes.

Zahlenwerte, die mehrere Bytes umfassen, werden im Little-Endian-Format abgelegt, d.h. je niederwertiger ein Byte ist desto niedriger ist seine Speicheradresse.

2.1 Paketheader

0 Byte 3			
Protokollkennung	Paketkennung	Paketlänge	
0x53 ('S')	Siehe Beschreibung	Unteres Byte	Oberes Byte

2.2 OP_HELLO (0x01)

Client ↔ Gateway – Gesamtlänge: 10 Bytes

0 Byte 3			
Protokollkennung	Paketkennung	Paketlänge	
Zeichenfolge „SHTCP“			
		Version	

0 Byte 3			
0x53	0x01	0x0006 (6)	
'S'	'H'	'T'	'C'
'P'	0x01		

Beschreibung: Mit OP_HELLO wird dem SH ARC-GATEWAY das Kommunikationsprotokoll mitgeteilt. Diese Paket-Kennung sollte immer als erstes Datenpaket gesendet werden, nachdem der Socket geöffnet wurde. Das SH ARC-GATEWAY schließt sofort den Socket, wenn das erste Datenpaket nicht OP_HELLO ist.

Das SH ARC-GATEWAY antwortet auf OP_HELLO, um dem Client mitzuteilen, dass das Protokoll unterstützt wird und für weitere Anfragen bereit ist.

Wird die Kommunikation mit dem SH ARC-GATEWAY ohne OP_HELLO eingeleitet, bricht das Gateway die Verbindung ggf. mit OP_BYE (0xFF) ab.

2.3 OP_READ_REQUEST (0x02)

Client → Gateway – Gesamtlänge: 4 Bytes

0	Byte		3
Protokollkennung	Paketkennung	Paketlänge	

0	Byte		3
0x53	0x02	0x0000 (0)	

Beschreibung: Mit OP_READ_REQUEST wird dem SH ARC-GATEWAY mitgeteilt, dass der Client ein ARCNET Paket empfangen will (siehe Beispiel 3.1).

Hinweis: Das SH ARC-GATEWAY empfängt erst ARCNET-Pakete (solange es Ressourcen frei hat) vom ARCNET-Netzwerk, nachdem mindestens ein OP_READ_REQUEST bearbeitet oder OP_READ_CONTINUOUS aktiviert wurde.

2.4 OP_READ_ERROR (0x03)

Client ← Gateway – Gesamtlänge: 8 Bytes

0	Byte		3
Protokollkennung	Paketkennung	Paketlänge	
Fehlercode			

0	Byte		3
0x53	0x03	0x0004 (4)	
0x00000000 - 0xFFFFFFFF			

Beschreibung: OP_READ_ERROR wird gesendet, wenn eine Read-Anfrage im ARCNET-Netzwerk durch einen Fehler abgebrochen wurde, oder nach einem Timeout.

2.5 OP_READ_DATA (0x04)

Client ← Gateway – Gesamtlänge: 11 – 518 Bytes

0		Byte		3	
Protokollkennung		Paketkennung		Paketlänge	
SID (Sender-Node-ID)		DID (Empfänger/ Destination- Node-ID)		Reserviert	
Reserviert			Daten 1–508 Byte		

0		Byte		3	
0x53		0x04		0x0007–0x0202 (7–514)	
0x00–0xFF (0–255)		0x01–0xFF (1–255)		–	
–			...		

Beschreibung: OP_READ_DATA liefert die ausgelesenen Daten aus dem ARCNET-Netzwerk.

2.6 OP_WRITE_DATA (0x05)

Client → Gateway – Gesamtlänge: 11 – 518 Bytes

0		Byte		3	
Protokollkennung		Paketkennung		Paketlänge	
SID		DID		Reserviert	
Reserviert				Daten 1–508 Byte	

0		Byte		3	
0x53		0x05		0x0007–0x0202 (7–514)	
0x01–0xFF (1–255)		0x00–0xFF (0–255)		–	
–				...	

Beschreibung: Mit OP_WRITE_DATA liefert der Client dem SH ARC-GATEWAY die Daten, die an den ARCNET Teilnehmer gesendet werden sollen.

2.7 OP_WRITE_RESULT (0x06)

Client ← Gateway – Gesamtlänge: 8 Bytes

0		Byte		3	
Protokollkennung		Paketkennung		Paketlänge	
Ergebnis des OP_WRITE_DATA					

0		Byte		3	
0x53		0x06		0x0004 (4)	
0x00000000 – 0xFFFFFFFF					

Beschreibung: Ein OP_WRITE_RESULT wird immer auf einen OP_WRITE_DATA gesendet, womit der Client über das Ergebnis des Write-Vorgangs informiert wird. Der Wert 0 zeigt den Erfolg an.

2.8 OP_READ_CONTINUOUS (0x07)

Client → Gateway – Gesamtlänge: 5 Bytes

0			Byte			3		
Protokollkennung		Paketkennung		Paketlänge				
Ein-/Ausschalten 0x00: aus 0x01: ein								

0			Byte			3		
0x53		0x07		0x0001 (1)				
0x01/0x00								

Beschreibung: OP_READ_CONTINUOUS wird gesendet, um dauerhaften Empfang ein- und auszuschalten.

Wenn dauerhafter Empfang aktiv ist, sendet der Server die OP_READ_DATA und OP_READ_ERROR ohne Read-Anfrage (OP_READ_REQUEST) von dem Client (siehe Beispiel 3.2).

Hinweis: Das SH ARC-GATEWAY empfängt erst ARCNET-Pakete, solange es Ressourcen frei hat, vom ARCNET-Netzwerk, wenn mindestens ein OP_READ_REQUEST bearbeitet wird oder OP_READ_CONTINUOUS aktiviert ist.

2.9 OP_LIFE_SIGN_REQUEST (0x08)

Client → Gateway – Gesamtlänge: 8 Bytes

0	<i>Byte</i>		3
Protokollkennung	Paketkennung	Paketlänge	
Verzögerung in Sekunden (min. 10s)			

0	<i>Byte</i>		3
0x53	0x08	0x0004 (4)	
Verzögerung			

Beschreibung: OP_LIFE_SIGN_REQUEST wird gesendet, um den Zeitgeber auf der Serverseite mit einer bestimmten Verzögerungszeit zu aktivieren. Der Server sendet dann ein einzelnes OP_LIFE_SIGN, wenn der Zeitgeber abläuft (siehe Beispiel 3.4).

2.10 OP_LIFE_SIGN (0x09)

Client ← Gateway – Gesamtlänge: 4 Bytes

0	<i>Byte</i>		3
Protokollkennung	Paketkennung	Paketlänge	

0	<i>Byte</i>		3
0x53	0x09	0x0000 (0)	

Beschreibung: OP_LIFE_SIGN wird an den Client gesendet, wenn der Zeitgeber nach einer bestimmten Verzögerung abläuft. Das Verhalten muss mit einer einzelnen Abfrage durch OP_LIFE_SIGN_REQUEST aktiviert werden. (siehe Beispiel 3.4).

2.11 OP_LIFE_SIGN_ACK (0x11)

Client → Gateway – Gesamtlänge: 4 Bytes

0			Byte			3		
Protokollkennung		Paketkennung		Paketlänge				
0			Byte			3		
0x53		0x11		0x0000 (0)				

Beschreibung: mit OP_LIFE_SIGN_ACK quittiert der Client ein OP_LIFE_SIGN (siehe Beispiel 3.4).

2.12 OP_SET_NODE_ID_REQUEST (0x0A)

Client → Gateway – Gesamtlänge: 6 - 260 Bytes

0			Byte			3		
Protokollkennung		Paketkennung		Paketlänge				
Anzahl der Nodes		Node ID 1		[Node ID 2]		...		
0			Byte			3		
0x53		0x0A		0x0002 - 0x0100 (2-256)				
0x01 - 0xFF			

Beschreibung: Mit OP_SET_NODE_ID_REQUEST definiert der Client die Node IDs aus dem ARCNET-Netzwerk, die er abbildet. Es ist auch möglich, dass sich der Client beim Gateway mit mehreren Node-IDs anmeldet.

Beispiele:

- 1) Ein Knoten mit ID 32 ergibt folgendes Paket
0x53,0x0A,0x02,0x00,0x01,0x20
- 2) Die Knoten mit ID 1, 15 und 255 ergeben folgendes Paket
0x53,0x0A,0x04,0x00,0x03,0x01,0x0F,0xFF

2.13 OP_SET_NODE_ID_RESULT (0x0B)

Client ← Gateway – Gesamtlänge: 8 Bytes

0	<i>Byte</i>		3
Protokollkennung	Paketkennung	Paketlänge	
Ergebnis des OP_SET_NODE_ID_RESULT			

0	<i>Byte</i>		3
0x53	0x0B	0x0004 (4)	
0x00000000 - 0xFFFFFFFF			

Beschreibung: OP_SET_NODE_ID_RESULT wird immer als Antwort auf einen OP_SET_NODE_ID_REQUEST gesendet, womit der Client über das Ergebnis der Einstellung der Node-IDs informiert wird.

Der Wert 0 zeigt die erfolgreiche Übertragung an.

2.14 OP_NETMAP_REQUEST (0x0C)

Client → Gateway – Gesamtlänge: 5 Bytes

0		Byte		3	
Protokollkennung		Paketkennung		Paketlänge	
Netmap-Format (siehe unten)					

0		Byte		3	
0x53		0x0C		0x0001 (1)	
0x00/0x01					

Beschreibung: Mit OP_NETMAP_REQUEST kann der Client eine Liste der Knoten abfragen, die aktuell im Netzwerk anwesend sind. Mit „Netmap-Format“ wird das Format des zurückzuliefernden OP_NETMAP-Pakets festgelegt:

Format	Beschreibung
0x00	<p>Länge: 256 Bit (32 Bytes)</p> <p>Netzliste in Form eines Bit-Vektors, wobei ein gesetztes Bit einen vorhandenen Knoten anzeigt. Bit 1 in Byte 0 repräsentiert die Node-ID 1 (Bit 0 in Byte 0 wird nicht verwendet), das MSB des Vektors (Bit 7 in Byte 31) repräsentiert die Node-ID 255.</p> <p>z.B. Netzliste 1, 15 und 255</p> <p>Node 0..7: netmap[0] = 0x02 // ID 1</p> <p>Node 8..15: netmap[1] = 0x80 // ID 15</p> <p>...</p> <p>Node 248..255: netmap[31] = 0x80 // ID 255</p>
0x01	<p>Länge: 1 – 256 Bytes</p> <p>Byte-Array mit den IDs der im Netzwerk vorhandenen Knoten. Das erste Byte enthält die Anzahl der vorhandenen Knoten.</p> <p>z.B. Netzliste 1,15 und 255</p> <p>netmap[0] = 3</p> <p>netmap[1] = 1</p> <p>netmap[2] = 15</p> <p>netmap[3] = 255</p>

2.15 OP_NETMAP (0x0D)

Client ← Gateway – Gesamtlänge: 5 - 260 Bytes

0		Byte		3	
Protokollkennung	Paketkennung	Paketlänge			
Anzahl der Nodes	Node ID	Node ID	...		

0		Byte		3	
0x53	0x0D	0x0001 - 0x0100 (2-256)			
0x00 - 0xFF		

Beschreibung: OP_NETMAP liefert die aktuelle Netzliste in dem mit OP_NETMAP_REQUEST angeforderten Format:

2.16 OP_GET_PARAMETER (0x0E)

Client → Gateway – Gesamtlänge: 5 Bytes

0		Byte		3	
Protokollkennung	Paketkennung	Paketlänge			
Parameter-ID					

0		Byte		3	
0x53	0x0E	0x0001 (1)			
0x00-0xFF					

Beschreibung: Mit OP_GET_PARAMETER kann der Client den Wert eines Parameters des SH ARC-GATEWAY abfragen. Als Antwort auf OP_GET_PARAMETER wird OP_PARAMETER_VALUE gesendet.

2.17 OP_SET_PARAMETER (0x0F)

Client → Gateway – Gesamtlänge: 10 Bytes

0		<i>Byte</i>	3
Protokollkennung	Paketkennung	Paketlänge	
Parameter-ID (siehe Tabelle im nächsten Abschnitt)	Parameterformat derzeit nur 0x01	Wert (32 Bit, unteres Wort)	
Wert (32 Bit, oberes Wort)			

0		<i>Byte</i>	3
0x53	0x0F	0x0006 (6)	
Parameter-ID	0x01	0x00000000 - 0xFFFFFFFF	

Beschreibung: Mit OP_SET_PARAMETER kann der Client den Wert eines Parameters im SH ARC-GATEWAY setzen. Als Reaktion auf OP_SET_PARAMETER wird OP_PARAMETER_VALUE gesendet.

2.18 OP_PARAMETER_VALUE (0x10)

Client ← Gateway – Gesamtlänge: 10 Bytes

0		Byte		3	
Protokollkennung		Paketkennung		Paketlänge	
Parameter-ID		Rückgabety 0x00: Fehlercode 0x01: Parameter		Wert (32 Bit, unteres Wort)	
		Wert (32 Bit, oberes Wort)			

0		Byte		3	
0x53		0x10		0x0006 (6)	
Parameter-ID		0x00/0x01		0x00000000 - 0xFFFFFFFF	

Beschreibung: Mit OP_PARAMETER_VALUE teilt SH ARC-GATEWAY dem Client entweder den Wert des abgefragten Parameters mit oder einen Fehlercode, falls der Parameter nicht gesetzt ist bzw. nicht abgefragt werden kann.

Parameterwerte:

Parameter-ID	Access	Beschreibung
0x00	Read Only	Aktueller Modus 0x00: Single-Node 0x01: Multi-Node
0x01	Read Only	Default-Node-ID für Single-Node-Modus
0x02	Read Only	Aktuelle Bitrate: 0x00: 19,5 kBit/s 0x01: 39 kBit/s 0x02: 78 kBit/s 0x03: 156 kBit/s 0x04: 312 kBit/s 0x05: 625 kBit/s 0x06: 1,25 Mbit/s 0x07: 2,5 Mbit/s 0x08: 5 MBit/s 0x09: 10 Mbit/s
0x03	Read Only	Extended Timeout 0x00: 1x Timeout (Standard) 0x01: 4x Timeout 0x02: 8x Timeout 0x03: 16x Timeout
0x04	Read Only	Receive-all: 0x00: deaktiviert 0x01: aktiviert
0x05	Read Only	Applikationsprotokoll (siehe Benutzerhandbuch) 0x00: SOHARD 0x01: AUG
0x06	Read/Write	Anzahl der gepufferten Pakete, die SH ARC-GATEWAY zwischenspeichert, wenn sie der Client nicht mehr ausliest

3 Beispiele zum SOHARD-Protokoll

3.1 Dauerhafte Datenabfrage mit OP_READ_REQUEST

Client		Server
OP_HELLO	=>	
	<=	OP_HELLO
OP_READ_REQUEST	=>	
	<=	OP_READ_DATA (OP_READ_ERROR)
OP_READ_REQUEST	=>	
	<=	OP_READ_DATA (OP_READ_ERROR)

3.2 Dauerhafte Datenabfrage mit OP_READ_CONTINUOUS

Client		Server
OP_HELLO	=>	
	<=	OP_HELLO
OP_READ_CONTINUOUS (0x1)	=>	
	<=	OP_READ_DATA (OP_READ_ERROR)
	<=	OP_READ_DATA (OP_READ_ERROR)

3.3 Datenabfrage mit OP_REQUEST und Datentransfer

Client		Server
OP_HELLO	=>	
	<=	OP_HELLO
OP_READ_REQUEST	=>	
OP_WRITE_DATA	=>	
	<=	OP_READ_DATA (OP_READ_ERROR)
	<=	OP_WRITE_RESULT

Bemerkung: die Reihenfolge von OP_WRITE_RESULT und OP_READ_DATA (OP_READ_ERROR) ist undefiniert.

3.4 Überprüfen der Serververbindung mit OP_LIFE_SIGN_REQUEST

Client		Server
OP_HELLO	=>	
	<=	OP_HELLO
OP_LIFE_SIGN_REQUEST (Time-out > 10 Sekunden)	=>	
	<=	OP_LIFE_SIGN
OP_LIFE_SIGN_ACK	=>	
Durch OP_LIFE_SIGN_REQUEST festgelegte Wartezeit		
	<=	OP_LIFE_SIGN
OP_LIFE_SIGN_ACK	=>	

4 Beschreibung des AUG-Protokolls

Das Feld „Paketlänge“ ist 2 Byte groß und enthält die Länge des Pakets **inklusive Header** in Bytes.

Zahlenwerte, die mehrere Bytes umfassen, werden im Little-Endian-Format abgelegt, d.h. je niederwertiger ein Byte ist desto niedriger ist seine Speicheradresse.

4.1 Paketheader

0	Byte	3
Paketlänge (inklusive Header)		Opcode
Daten ...		

4.2 Sicherer Paketheader

0	Byte	3
Paketlänge (inklusive Header)		Opcode
Sicherheitscode		
Daten ...		

Beschreibung: Sichere Paketheader werden aus Kompatibilitätsgründen verwendet. Der Sicherheitscode ist immer „0000“ (0x30303030).

4.3 OP_CQ (Config Query)

Gesamtlänge: 8 Bytes

0	Byte	3
Paketlänge (inklusive Header)		Opcode: 'C', 'Q' (0x4351)
Sicherheitscode		

Beschreibung: Mit OP_CQ kann ein Konfigurationsbericht angefordert werden. Auf OP_CQ antwortet SH ARC-GATEWAY mit OP_CR:

4.4 OP_CR (Config Report)

Gesamtlänge: >= 19 Bytes

0		Byte		3	
Paketlänge (inklusive Header)			Opcode: 'C', 'R' (0x4352)		
Node-ID des Gateways	Interface-Modul-Kennung	Übertragungsrate	ARCNET-Timeout		
RECON-Timeout	Receive all 0x00: aus 0x01: ein	Timeout Knotenliste in 1/10 s			
Speicher-Größe			Hardware-Version Low High		
Software-Version unteres oberes Byte			Herstellernamen (null-terminierter String) ...		

Beschreibung: Mit OP_CR wird ein Konfigurationsbericht zurückgeliefert.

4.5 OP_CS (Config Set)

Gesamtlänge: 16 Bytes

0		Byte		3	
Paketlänge (inklusive Header)			Opcode: 'C', 'S' (0x4353)		
Sicherheitscode					
Node-ID des Gateways	Interface-Modul-Kennung	Übertragungsrate	ARCNET-Timeout		
RECON-Timeout	Receive-all 0x00: aus 0x01: ein	Timeout Knotenliste in 1/10 s			

Beschreibung: Mit OP_CS wird lediglich die Node-ID des SH ARC-GATEWAY konfiguriert. Übertragungsrate, ARCNET-Time-out und Receive-all-Modus werden im Web-UI oder in den Konfigurationsdateien eingestellt.

4.6 OP_ER (Error Response)

Gesamtlänge: >= 12 Bytes

0	Byte	3
Paketlänge (inklusive Header)		Opcode: 'E', 'R' (0x4552)
Opcode der fehlgeschlagenen Operation		Fehlerursache 0x00: andere 0x01: Gateway 0x02: ARCNET
Fehlercode		
Optional: Fehlertext (null-terminierter String) ...		

Beschreibung: Falls bei Anfragen ein Fehler auftritt, antwortet SH ARC-GATEWAY mit dem Fehlerpaket OP_ER.

4.7 OP_PQ (Promiscuous („Receive-all“) Mode Query)

Gesamtlänge: 8 Bytes

0	Byte	3
Paketlänge (inklusive Header)		Opcode 'P','Q' (0x5051)
Sicherheitscode		

Beschreibung: Mit diesem Opcode wird ermittelt, ob der „Receive-All“-Modus aktiviert ist. Im Receive-all-Modus werden nicht nur die Datenpakete, die an die Node-ID des Geräts gerichtet sind, aufgezeichnet, sondern alle auf dem Bus gesendeten Datenpakete.

Auf OP_PQ antwortet SH ARC-GATEWAY mit OP_PR:

4.8 OP_PR (Promiscuous („Receive-all“) Mode Report)

Gesamtlänge: 5 Bytes

0	Byte	3
Paketlänge (inklusive Header)		Opcode 'P','R' (0x5052)
Receive-all 0x00: aus sonst: ein		

Beschreibung: Mit diesem Paket meldet SH ARC-GATEWAY, ob der Receive-all-Modus aktiviert oder deaktiviert ist.

4.9 OP_PS (Promiscuous („Receive-all“) Mode Set)

Gesamtlänge: 9 Bytes

0	Byte	3
Paketlänge (inklusive Header)		Opcode 'P','S' (0x5053)
Sicherheitscode		
Receive-all 0x00: aus sonst: ein		

Beschreibung: Mit diesem Opcode wird der Receive-all-Modus eingestellt, d.h. nicht nur die Datenpakete, die an die Node-ID des Geräts gerichtet sind, werden aufgezeichnet, sondern alle auf dem Bus gesendeten Datenpakete.

4.10 OP_SQ (Status Query)

Gesamtlänge: 8 Bytes

0	Byte	3
Paketlänge (inklusive Header)	Opcode 'S','Q' (0x5351)	
Sicherheitscode		

Beschreibung: Mit diesem Opcode werden die Statistiken von SH ARC-GATEWAY abgefragt.

Auf OP_SQ antwortet SH ARC-GATEWAY mit OP_SR:

4.11 OP_SR (Status Report)

Gesamtlänge: 24 Bytes

0	Byte	3
Paketlänge (inklusive Header)	Opcode: 'S', 'R' (0x5351)	
Anzahl der empfangenen ARCNET-Datenpakete		
Anzahl der gesendeten ARCNET-Datenpakete		
Anzahl der empfangenen Ethernet-Pakete		
Anzahl der gesendeten Ethernet-Pakete		
Anzahl der angeschlossenen Ethernet-Clients		

Beschreibung: Statistiken des SH ARC-GATEWAY.

4.12 OP_CRESQ (Status Counter Reset Query)

Gesamtlänge: 8 Bytes

0	Byte	3
Paketlänge (inklusive Header)		Opcode OP_CRESQ (0x6002)
Sicherheitscode		

Beschreibung: Mit diesem Opcode werden die Zähler der Statistiken zurückgesetzt.

Auf OP_CRESQ antwortet SH ARC-GATEWAY mit OP_CRESR:

4.13 OP_CRESR (Status Counter Reset Response)

Gesamtlänge: 8 Bytes

0	Byte	3
Paketlänge (inklusive Header)		Opcode OP_CRESR (0x6003)
Sicherheitscode		

Beschreibung: Mit diesem Opcode bestätigt SH ARC-GATEWAY, dass die Statuszähler zurückgesetzt wurden.

4.14 OP_WQ (Write Request)

Gesamtlänge: >= 8 Bytes

0		Byte		3	
Paketlänge (inklusive Header)			Opcode 'W','Q' (0x5751)		
Sicherheitscode					
SID (Sender-Node-ID)		DID (Empfänger/ Destination- Node-ID)		Länge des Datenpakets	
ARCNET-Datenpaket mit der angegebenen Länge ...					

Beschreibung: Mit diesem Opcode wird ein ARCNET-Datenpaket gesendet.

Auf OP_WQ antwortet SH ARC-GATEWAY mit OP_WO:

4.15 OP_WO (Write O.K.)

Gesamtlänge: 8 Bytes

0		Byte		3	
Paketlänge (inklusive Header)			Opcode 'W','O' (0x574F)		
Sendestatus 0x00: Senden erfolgreich 0x01: Time-out, EXCNAK oder RECON 0x02: Senden nicht abgeschlossen 0x03: Senden über ARCNET fehlgeschlagen					

Beschreibung: Nachdem SH ARC-GATEWAY ein „Write Request“ erhalten hat, kann es entweder sofort nach dem Absenden der Daten antworten (Sendestatus = 2) oder erst antworten, wenn der Sendevorgang abgeschlossen ist (Sendestatus = 0 oder 1). Näheres siehe unter „Flag Status Query“ oder „Wake On Transmit Complete“.

4.16 OP_DA (Data Advise)

Gesamtlänge: 43 Bytes

0		Byte		3			
Paketlänge (inklusive Header)			Opcode 'D','A' (0x4441)				
Sicherheitscode							
da_map[0]		da_map[1]		da_map[2]		da_map[3]	
da_map[4] ... da_map[27]							
da_map[28]		da_map[29]		da_map[30]		da_map[31]	
da_time: Zeit bis zur Löschung der Anfrage in 1/10 s (0: unbegrenzt)				da_rcvall: Receive-all			

Beschreibung: Mit diesem Opcode kann der Empfang von ARCNET-Paketen bestimmter Knoten eingestellt werden. Frühere OP_DA-Operationen werden überschrieben.

Das Flag „da_rcvall“ ist nur aus Kompatibilitätsgründen vorhanden. Der Receive-all-Modus wird im Web-UI oder in den Konfigurationsdateien eingestellt.

Im Receive-all-Modus empfängt der Client auch Pakete von den Knoten, die in „da_map“ angegeben sind, und nicht nur von den Knoten, die an SH ARC-GATEWAY senden.

Im Array „da_map“ repräsentiert jedes Bit einen Knoten. Wenn das Bit für einen Knoten gesetzt ist, werden Nachrichten dieses Knoten weitergeleitet. Dabei gibt es folgende Entsprechungen:

Bit 7 in da_map[0]: Node-ID 0

Bit 0 in da_map[0]: Node-ID 7

Bit 7 in da_map[31]: Node-ID 248

Bit 0 in da_map[31]: Node-ID 255

4.17 OP_DQ (Data Request)

Gesamtlänge: 10 Bytes

0		Byte		3	
Paketlänge (inklusive Header)			Opcode 'D','Q' (0x4451)		
Sicherheitscode					
ARCNET- Node-ID		Gewünschte Paketanzahl aus dem Speicher			

Beschreibung: Mit dem Opcode kann eine beliebige Anzahl von ARCNET-Paketen des Knotens mit „ARCNET-Node-ID“ abgerufen werden. Bei Einstellung von OP_DA wird dies ignoriert.

Auf OP_DQ antwortet SH ARC-GATEWAY mit OP_DR:

4.18 OP_DR (Data Response)

Gesamtlänge: 13 >= Bytes

0		Byte		3	
Paketlänge (inklusive Header)			Opcode 'D','R' (0x4452)		
Eindeutige ID eines ARCNET-Telegramms					
Immer 0x00		Sender-ID		Empfänger-ID	Paketlänge unteres Byte
Paketlänge oberes Byte		ARCNET-Datenpaket			

Beschreibung: ARCNET-Pakete werden dem Client im obigen Format zugesendet.

4.19 OP_NQ (Nodemap Request)

Gesamtlänge: 8 Bytes

0	Byte	3
Paketlänge (inklusive Header)		Opcode 'N','Q' (0x4E51)
Sicherheitscode		

Beschreibung: Mit diesem Opcode kann die Knotenliste (Nodemap) von SH ARC-GATEWAY angefordert werden.

Auf OP_NQ antwortet SH ARC-GATEWAY mit OP_NR:

4.20 OP_NR (Nodemap Response)

Gesamtlänge: 36 Bytes

0	Byte	3	
Paketlänge (inklusive Header)		Opcode 'N','R' (0x4E52)	
nr_map[0]	nr_map[1]	nr_map[2]	nr_map[3]
nr_map[4]			
...			
nr_map[27]			
nr_map[28]	nr_map[29]	nr_map[30]	nr_map[31]

Beschreibung: Mit dem Opcode wird das 32 Bytes große Array „nr_map“ gesendet, welches Auskunft über die Existenz eines Knotens im ARCNET-Netzwerk gibt. Jedes der 256 Bits repräsentiert ein Bit eines Knotens. Dieses ist vorhanden, wenn das entsprechende Bit gesetzt ist. Bit 1 in nr_map[0] entspricht der Node-ID 1 (Bit 0 in Byte 0 wird nicht verwendet), Bit 7 in nr_map[31] entspricht der Node-ID 255.

4.21 OP_NDQ (TCP NODEDELAY Setting)

Gesamtlänge: 10 Bytes

0	Byte	3
Paketlänge (inklusive Header)		Opcode NDQ (0x6000)
Sicherheitscode		
Aktion 0x00: aus 0x01: an 0x02: Statusabfrage		

Beschreibung: Bei einer TCP-Verbindung können mehrere ARCNET-Pakete von SH ARC-GATEWAY zu einem einzigen TCP-Paket zusammengefasst werden. Der TCP-Stack entscheidet darüber, wann ein solches Paket gesendet wird. Wenn ARCNET-Pakete unmittelbar nach dem Empfang weitergeleitet werden sollen, dann muss die Option „TCP NODELAY“ aktiviert werden.

Auf OP_NDQ antwortet SH ARC-GATEWAY mit OP_NDR:

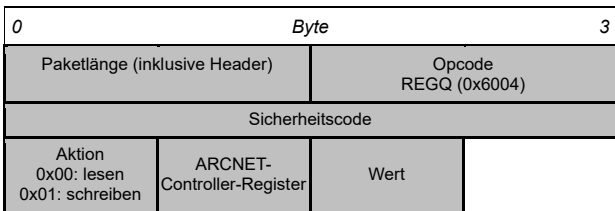
4.22 OP_NDR (TCP NODELAY Response)

Gesamtlänge: 6 Bytes

0	Byte	3
Paketlänge (inklusive Header)		Opcode NDR (0x6001)
Status 0x00: aus 0x01: an		

4.23 OP_REGQ (ARCNET Register Read/Write)

Gesamtlänge: 11 Bytes

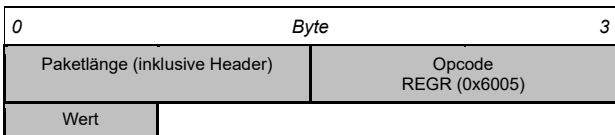


Beschreibung: Mit diesem Opcode können die Register des ARCNET-Controllers im SH ARC-GATEWAY beschrieben und gelesen werden.

Auf OP_REGQ antwortet SH ARC-GATEWAY mit OP_REGR:

4.24 OP_REGR (ARCNET Register Read/Write Response)

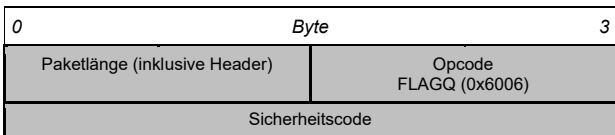
Gesamtlänge: 5 Bytes



Beschreibung: Mit diesem Opcode antwortet SH ARC-GATEWAY auf OP_REGQ.

4.25 OP_FLAGQ (Flag Status Query)

Gesamtlänge: 8 Bytes

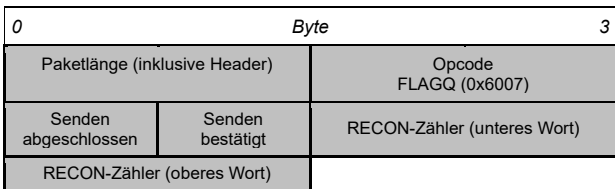


Beschreibung: Der Status einer laufenden ARCNET-Übertragung kann mit diesem Opcode abgefragt werden.

SH ARC-GATEWAY antwortet darauf mit OP_FLAGR:

4.26 OP_FLAGR (Flag Status Response)

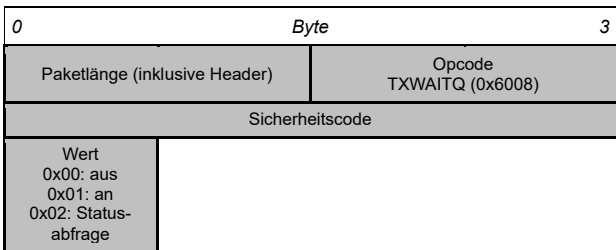
Gesamtlänge: 10 Bytes



Beschreibung: Wenn das Flag „Senden abgeschlossen“ gesetzt ist, das Flag „Senden bestätigt“ jedoch noch nicht, dann wurde ein ARCNET-Paket gesendet aber noch nicht bestätigt (wie es z.B. bei einer Broadcast-Message der Fall ist). „RECON-Zähler“ enthält die Zahl der Rekonfigurationen seit der letzten Abfrage.

4.27 OP_TXWAITQ (Wake On TX Complete)

Gesamtlänge: 9 Bytes

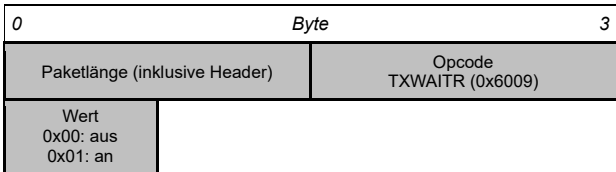


Beschreibung: Mit diesem Opcode kann eingestellt werden, dass SH ARC-GATEWAY erst wieder auf ein „Write Request“ reagiert, wenn der vorherige Sendevorgang abgeschlossen ist.

SH ARC-GATEWAY antwortet darauf mit OP_TXWAITR:

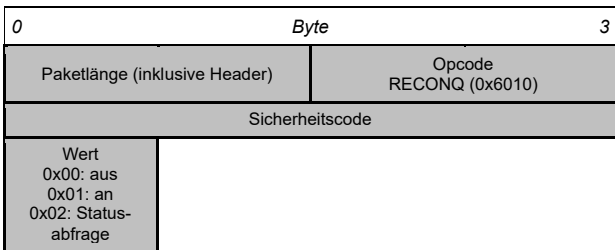
4.28 OP_TXWAITR (Wake On TX Complete Response)

Gesamtlänge: 5 Bytes



4.29 OP_RECONQ (Wake On RECON)

Gesamtlänge: 9 Bytes

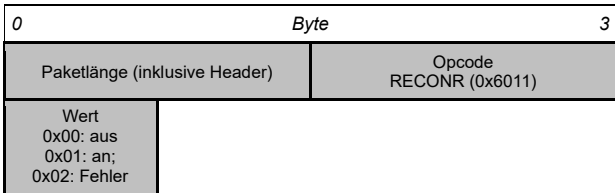


Beschreibung: Mit diesem Opcode kann eingestellt werden, dass SH ARC-GATEWAY eine Nachricht sendet, wenn eine Rekonfiguration erfolgt.

SH ARC-GATEWAY antwortet darauf mit OP_RECONR:

4.30 OP_RECONR (Wake On RECON Response)

Gesamtlänge: 5 Bytes



Beschreibung: Wenn „Wake On Recon“ aktiviert ist, wird bei Auftreten eines RECONs ein Paket-Header OP_RECONM gesendet:

4.31 OP_RECONM (RECON Message)

Gesamtlänge: 4 Bytes

0	<i>Byte</i>	3
Paketlänge (inklusive Header)	Opcode RECONM (0x6012)	

4.32 OP_TXTIMEOQ (Transmit Timeout)

Gesamtlänge: 12 Bytes

0	<i>Byte</i>	3
Paketlänge (inklusive Header)	Opcode TXTIMEOQ (0x6013)	
Sicherheitscode		
Timeout-Zeit in Millisekunden		

Beschreibung: Mit diesem Opcode kann die Timeout-Zeit für ARCNET-Übertragungen eingestellt werden (max. 65535 ms).

SH ARC-GATEWAY antwortet darauf mit OP_TXTIMEOR:

4.33 OP_TXTIMEOR (Transmit Timeout Response)

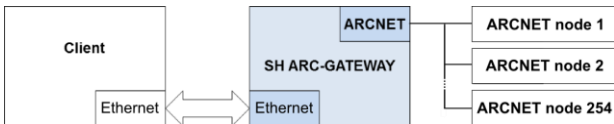
Gesamtlänge: 8 Bytes

0	<i>Byte</i>	3
Paketlänge (inklusive Header)	Opcode TXTIMEOR (0x6014)	
Rückmeldung zur Einstellung des Timeouts 0x00: erfolgreich 0x01: fehlgeschlagen		

5 General Information

SH ARC-GATEWAY provides access to an ARCNET network using the TCP/IP protocol. The device is Linux-based, which simplifies the integration into an existing IT infrastructure and subsequent administration.

The following picture illustrates a typical application environment of SH ARC-GATEWAY:



The following chapters describe the application protocols of SOHARD and of the ARCNET User Group e.V. (AUG).

The configuration of the device is described in the User Manual.

The device and the documentation are subject to change or functional extension without notice.

6 Description of the SOHARD Protocol

Communication with SH ARC-GATEWAY takes place via TCP/IP sockets. The protocol consists of the packet types described below, each data packet comprehending a header which consists of the fields

- protocol identifier,
- packet identifier and
- packet length.

The 2-byte field "packet length" contains the length of the packet **excluding the header**.

Numerical values the size of which is more than one byte are **little endian**.

6.1 Packet header

0				<i>byte</i>		3	
Protocol identifier		Packet identifier		Packet length			
0x53 ('S')		(see description)		Lower byte		Upper byte	

6.2 OP_HELLO (0x01)

Client ↔ Gateway – Total length: 10 bytes

0				<i>byte</i>		3	
Protocol identifier		Packet identifier		Packet length			
Character string "SHTCP"							
				Version			

0				<i>byte</i>		3	
0x53		0x01		0x0006 (6)			
'S'		'H'		'T'		'C'	
'P'		0x01					

Description: The communication protocol is disclosed to SH ARC-GATEWAY by sending OP_HELLO. This packet identifier is the first data packet to be sent after the socket has been opened. Otherwise SH ARC-GATEWAY will close the socket immediately.

SH ARC-GATEWAY replies to OP_HELLO informing the client that the protocol is being supported and that it is ready to receive further requests.

If the communication to SH ARC-GATEWAY is initiated without OP_HELLO, the gateway will close the connection. In some case OP_BYE (0xFF) may be sent before.

6.3 OP_READ_REQUEST (0x02)

Client → Gateway – Total length: 4 bytes

0	<i>byte</i>		3
Protocol identifier	Packet identifier	Packet length	
0x53	0x02	0x0000 (0)	

Description: By sending OP_READ_REQUEST a client informs SH ARC-GATEWAY that it is ready to receive an ARCNET packet (see example 7.1).

Note: The SH ARC-GATEWAY only receives ARCNET packets, as long as it has free resources, from the ARCNET network when at least one OP_READ_REQUEST is in progress or OP_READ_CONTINUOUS is activated.

6.4 OP_READ_ERROR (0x03)

Client ← Gateway – Total length: 8 bytes

0	<i>byte</i>		3
Protocol identifier	Packet identifier	Packet length	
Error code			
0x53	0x03	0x0004 (4)	
0x00000000 – 0xFFFFFFFF			

Description: OP_READ_ERROR is sent when a read request in the ARCNET network was interrupted or after a timeout.

6.5 OP_READ_DATA (0x04)

Client ← Gateway – Total length: 11 – 518 bytes

0			<i>byte</i>			3		
Protocol identifier		Packet identifier		Packet length				
SID (Sender node ID)		DID (Destination node ID)		Reserved				
Reserved				Data 1–508 bytes				

0			<i>byte</i>			3		
0x53		0x04		0x0007–0x0202 (7–514)				
0–255		1–255		–				
–		–		...				

Description: OP_READ_DATA contains data from the ARCNET network for the client.

6.6 OP_WRITE_DATA (0x05)

Client → Gateway – Total length: 11 – 518 bytes

0			<i>byte</i>			3		
Protocol identifier		Packet identifier		Packet length				
SID		DID		Reserved				
Reserved				Data 1–508 bytes				

0			<i>byte</i>			3		
0x53		0x05		0x0007–0x0202 (7–514)				
1–255		0–255		–				
–		–		...				

Description: OP_WRITE_DATA contains data from the client that is to be sent to an ARCNET node by SH ARC-GATEWAY.

6.7 OP_WRITE_RESULT (0x06)

Client ← Gateway – Total length: 8 bytes

0			<i>byte</i>			3		
Protocol identifier		Packet identifier		Packet length				
Return value of OP_WRITE_DATA								

0			<i>byte</i>			3		
0x53		0x06		0x0004 (4)				
0x00000000 – 0xFFFFFFFF								

Description: OP_WRITE_RESULT is always sent following OP_WRITE_DATA. It informs the client of the return value of the write command.

Value 0 indicates a successful transmission.

6.8 OP_READ_CONTINUOUS (0x07)

Client → Gateway – Total length: 5 bytes

0			<i>byte</i>			3		
Protocol identifier		Packet identifier		Packet length				
0x00: disable 0x01: enable								

0			<i>byte</i>			3		
0x53		0x07		0x0001 (1)				
0x01/0x00								

Description: OP_READ_CONTINUOUS is sent to enable or the disable the continuous reception. When continuous reception is enabled the server sends OP_READ_DATA and OP_READ_ERROR without OP_READ_REQUEST from the Client (see example 7.2).

Note: The SH ARC-GATEWAY only receives ARCNET packets, as long as it has free resources, from the ARCNET network when at least one OP_READ_REQUEST is in progress or OP_READ_CONTINUOUS is activated.

6.9 OP_LIFE_SIGN_REQUEST (0x08)

Client → Gateway – Total length: 8 bytes

0	<i>byte</i>		3
Protocol identifier	Packet identifier	Packet length	
Heartbeat delay in seconds (10s min.)			

0	<i>byte</i>		3
0x53	0x08	0x0004 (4)	
Delay			

Description: OP_LIFE_SIGN_REQUEST is sent to enable the heartbeat timer on server side with a specific delay. Now the server sends an OP_LIFE_SIGN every time the timer has expired (see example 7.4).

6.10 OP_LIFE_SIGN (0x09)

Client ← Gateway – Total length: 4 bytes

0	<i>byte</i>		3
Protocol identifier	Packet identifier	Packet length	

0	<i>byte</i>		3
0x53	0x09	0x0000 (0)	

Description: OP_LIFE_SIGN is sent to the client when the timer expires after a specific delay. This behavior has to be enabled by a single use of OP_LIFE_SIGN_REQUEST. (see example 7.4).

6.11 OP_LIFE_SIGN_ACK (0x11)

Client → Gateway – Total length: 4 Bytes

0	<i>Byte</i>		3
Protocol identifier	Packet identifier	Packet length	

0	<i>Byte</i>		3
0x53	0x11	0x0000 (0)	

Description: OP_LIFE_SIGN_ACK is send by the Client to acknowledge an OP_LIFE_SIGN (see example 7.4).

6.12 OP_SET_NODE_ID_REQUEST (0x0A)

Client → Gateway – Total length: 6 - 260 Bytes

0	<i>byte</i>			3
Protocol identifier	Packet identifier	Packet length		
Count of Nodes	Node ID 1	[Node ID 2]	...	

0	<i>byte</i>			3
0x53	0x0A	0x0002 – 0x0100 (2-256)		
0x01 – 0xFF	

Description: OP_SET_NODE_ID_REQUEST allows the client to set one or multiple node IDs within the ARCNET network. These IDs are used for the data exchange.

Examples:

- 1) Request packet for one node with ID 32 will look like:
0x53,0x0A,0x02,0x00,0x01,0x20
- 2) Request packet for nodes with ID 1, 15 and 255 will look like
0x53,0x0A,0x04,0x00,0x03,0x01,0x0F,0xFF

6.13 OP_SET_NODE_ID_RESULT (0x0B)

Client ← Gateway – Total length: 8 Bytes

0	<i>byte</i>		3
Protocol identifier	Packet identifier	Packet length	
Result of OP_SET_NODE_ID_RESULT			

0	<i>byte</i>		3
0x53	0x0B	0x0004 (4)	
0x00000000 - 0xFFFFFFFF			

Description: OP_SET_NODE_ID_RESULT returns the result of OP_SET_NODE_ID_REQUEST.

The requested node IDs are set if the value is equal to 0, otherwise the requested IDs are not set.

6.14 OP_NETMAP_REQUEST (0x0C)

Client → Gateway – Total length: 5 Bytes

<i>0</i>		<i>byte</i>	<i>3</i>
Protocol identifier	Packet identifier	Packet length	
Format of Netmap (see below)			

<i>0</i>		<i>byte</i>	<i>3</i>
0x53	0x0C	0x0001 (1)	
0x00/0x01			

Description: OP_NETMAP_REQUEST allows the client to get the active node IDs within the ARCNET network. "Format of Netmap" defines how the list of the present nodes will be encoded.

Format	Description
0x00	<p>Length: 256 Bit (32 Bytes)</p> <p>List of the nodes in form of a bit array where a set bit represents a present node. Bit 1 in Byte 0 represents the Node ID 1, the MSB of the array (i.e. bit 7 of Byte 31) the ID 255.</p> <p>For example: node IDs 1,15 and 255</p> <p>Node 0..7: netmap[0] = 0x02 // ID 1</p> <p>Node 8..15: netmap[1] = 0x80 // ID 15</p> <p>...</p> <p>Node 248..255: netmap[31] = 0x80 // ID 255</p>
0x01	<p>Length: 1 – 256 Bytes</p> <p>Byte array with the list of nodes present in the ARCNET network. The first byte contains the number of nodes.</p> <p>For example: node IDs 1, 15 and 255</p> <p>netmap[0] = 3</p> <p>netmap[1] = 1</p> <p>netmap[2] = 15</p> <p>netmap[3] = 255</p>

6.15 OP_NETMAP (0x0D)

Client → Gateway – Total length: 5 - 260 Bytes

0				<i>byte</i>				3			
Protocol identifier			Packet identifier			Packet length					
number of Nodes			[Node ID 1]			[Node ID 2]			...		

0				<i>byte</i>				3			
0x53			0x0D			0x0001 - 0x0100 (1-256)					
0x00 - 0xFF				

Description: OP_NETMAP returns a list of the nodes that are currently present within the ARCNET network using the format requested by OP_NETMAP_REQUEST.

6.16 OP_GET_PARAMETER (0x0E)

Total length: 5 Bytes

0				<i>byte</i>				3			
Protocol identifier			Packet identifier			Packet length					
Parameter ID											

0				<i>byte</i>				3			
0x53			0x0E			0x0001 (1)					
0x00-0xFF											

Description: OP_GET_PARAMETER allows the client to request the value of SH ARC-GATEWAY parameters. The response to this opcode will be OP_PARAMETER_VALUE.

6.17 OP_SET_PARAMETER (0x0F)

Client → Gateway – Total length: 10 Bytes

0		<i>byte</i>	3
Protocol identifier	Packet identifier	Packet length	
Parameter ID (see table in the next section)	Format currently 0x01 only	Value (32 Bit, lower word)	
Value (32 Bit, upper word)			

0		<i>byte</i>	3
0x53	0x0F	0x0006 (6)	
Parameter ID	0x01	0x00000000 - 0xFFFFFFFF	

Description: OP_SET_PARAMETER allows the client to change the value of SH ARC-GATEWAY parameters. As result of the request the client will receive OP_PARAMETER_VALUE.

6.18 OP_PARAMETER_VALUE (0x10)

Client ← Gateway – Total length: 10 Bytes

0		<i>byte</i>		3	
Protocol identifier		Packet identifier		Packet length	
Parameter ID		Format		Value (32 Bit, lower word)	
Value (32 Bit, upper word)					

0		<i>byte</i>		3	
0x53		0x10		0x0006 (6)	
Parameter ID		Return value type 0x00: error code 0x01: parameter		0x00000000 - 0xFFFFFFFF	

Description: OP_PARAMETER_VALUE is used by SH ARC-GATEWAY for reporting the value of the parameters or an error code if a parameter is not set or cannot be read.

Parameter values:

Parameter ID	Access	Description
0x00	Read Only	Current mode 0x00: Single-Node 0x01: Multi-Node
0x01	Read Only	Default Node ID for Single-Node mode
0x02	Read Only	Current bit rate: 0x00: 19.5 kBit/s 0x01: 39 kBit/s 0x02: 78 kBit/s 0x03: 156 kBit/s 0x04: 312 kBit/s 0x05: 625 kBit/s 0x06: 1.25 Mbit/s 0x07: 2.5 Mbit/s 0x08: 5 MBit/s

		0x09: 10 Mbit/s
0x03	Read Only	Extended Timeout 0x00: 4x Timeout (standard) 0x01: 4x Timeout 0x02: 8x Timeout 0x03: 16x Timeout
0x04	Read Only	Receive-all: 0x00: disabled 0x01: enabled
0x05	Read Only	Application protocol (see User Manual) 0x00: SOHARD 0x01: AUG
0x06	Read/Write	Limit of cached TCP/IP packets that SH ARC-GATEWAY caches if remote client does not read the sent data.

7 Examples for the SOHARD Protocol

7.1 Permanent data retrieval using OP_READ_REQUEST

Client		Server
OP_HELLO	=>	
	<=	OP_HELLO
OP_READ_REQUEST	=>	
	<=	OP_READ_DATA (OP_READ_ERROR)
OP_READ_REQUEST	=>	
	<=	OP_READ_DATA (OP_READ_ERROR)

7.2 Permanent data retrieval using OP_READ_CONTINUOUS

Client		Server
OP_HELLO	=>	
	<=	OP_HELLO
OP_READ_CONTINUOUS (0x1)	=>	
	<=	OP_READ_DATA (OP_READ_ERROR)
	<=	OP_READ_DATA (OP_READ_ERROR)

7.3 Data retrieval using OP_REQUEST and data transfer.

Client		Server
OP_HELLO	=>	
	<=	OP_HELLO
OP_READ_REQUEST	=>	
OP_WRITE_DATA	=>	
	<=	OP_READ_DATA (OP_READ_ERROR)
	<=	OP_WRITE_RESULT

Note: the sequence of OP_WRITE_RESULT and OP_READ_DATA (OP_READ_ERROR) is undefined.

7.4 Verify server connection using OP_LIFE_SIGN_REQUEST.

Client		Server
OP_HELLO	=>	
	<=	OP_HELLO
OP_LIFE_SIGN_REQUEST (Time-out > 10 seconds)	=>	
	<=	OP_LIFE_SIGN
OP_LIFE_SIGN_ACK	=>	
Delay set by OP_LIFE_SIGN_REQUEST		
	<=	OP_LIFE_SIGN
OP_LIFE_SIGN_ACK	=>	

8 Description of the AUG Protocol

The 2 Bytes field “packet length” contains the length of the packet including the header in bytes.

Numerical values the size of which is more than one byte are little endian.

8.1 Packet Header

0	Byte	3
Packet length (incl. header)		Opcode
Data ...		

8.2 Secure Packet Header

0	Byte	3
Packet length (incl. header)		Opcode
Security code		
Data ...		

Description: secure packet headers are used for compatibility reasons. The security code always is “0000” (0x30303030).

8.3 OP_CQ (Config Query)

Total length: 8 Bytes

0	Byte	3
Packet length (incl. header)		Opcode: 'C', 'Q' (0x4351)
Security code		

Description: OP_CQ asks for a configuration report. In reply SH ARC-GATEWAY sends an OP_CR:

8.4 OP_CR (Config Report)

Total length: >= 19 Bytes

0		Byte		3	
Packet length (incl. header)		Opcode: 'C', 'R' (0x4352)			
Node-ID of the gateway	Interface module identifier	Transmission rate	ARCNET time-out		
RECON time-out	Receive-all 0x00: disabled 0x01: enabled	Timeout of node list in 1/10 s			
Memory size		Hardware version Low High			
Software version Low high byte		Manufacturer name (zero-terminated string) ...			

Description: OP_CR contains a configuration report.

8.5 OP_CS (Config Set)

Total length: 16 bytes

0		Byte		3	
Packet length (incl. header)		Opcode: 'C', 'S' (0x4353)			
Security code					
Node-ID of the gateway	Interface module identifier	Transmission rate	ARCNET time-out		
RECON time-out	Receive-all 0x00: disabled 0x01: enabled	Timeout of node list in 1/10 s			

Description: only the node ID of the present SH ARC-GATEWAY is configured using OP_CS. The remaining settings are made using the web UI or the configuration files.

8.6 OP_ER (Error Response)

Total length: >= 12 Bytes

0		Byte		3	
Packet length (incl. header)				Opcode: 'E', 'R' (0x4552)	
Opcode of the failed operation				Cause of failure 0x00: others 0x01: Gateway 0x02: ARCNET	
Error code					
Optional: error text (zero-terminated string) ...					

Description: in case of the failure of a request SH ARC-GATEWAY sends an OP_ER.

8.7 OP_PQ (Promiscuous (“Receive-all”) Mode Query)

Total length: 8 Bytes

0		Byte		3	
Packet length (incl. header)				Opcode 'P','Q' (0x5051)	
Security code					

Description: this Opcode asks whether the Receive-all mode is enabled. In Receive-all mode **all** data packets sent on the bus are being recorded, not only those directed to the node ID of the present device. The reply of the SH ARC-GATEWAY to OP_PQ is an OP_PR:

8.8 OP_PR (Promiscuous (“Receive-all”) Mode Report)

Total length: 5 Bytes

0	Byte	3
Packet length (incl. header)		Opcode 'P','R' (0x5052)
Receive-all 0x00: disabled 0x01: enabled		

Description: by means of OP_PR SH ARC-GATEWAY reports the state of the Receive-all mode.

8.9 OP_PS (Promiscuous (“Receive-all”) Mode Set)

Total length: 9 Bytes

0	Byte	3
Packet length (incl. header)		Opcode 'P','S' (0x5053)
Security code		
Receive-all 0x00: enabled else: disabled		

Description: by means of this Opcode the Receive-all mode is being enabled. In Receive-all mode **all** data packets sent on the bus are being recorded, not only those directed to the node ID of the present device.

8.10 OP_SQ (Status Query)

Total length: 8 Bytes

0	Byte	3
Packet length (incl. header)	Opcode 'S','Q' (0x5351)	
Security code		

Description: this Opcode asks for the statistics of the present SH ARC-GATEWAY.

It is being acknowledged by SH ARC-GATEWAY with OP_SR:

8.11 OP_SR (Status Report)

Total length: 24 Bytes

0	Byte	3
Packet length (incl. header)	Opcode: 'S', 'R' (0x5351)	
Number of received ARCNET data packets		
Number of sent ARCNET data packets		
Number of received Ethernet data packets		
Number of sent Ethernet data packets		
Number of connected Ethernet clients		

Description: statistics of the present SH ARC-GATEWAY.

8.12 OP_CRESQ (Status Counter Reset Query)

Total length: 8 Bytes

0	Byte	3
Packet length (incl. header)	Opcode OP_CRESQ (0x6002)	
Security code		

Description: this Opcode resets the counters for the statistics.

It is being acknowledged by the SH ARC-GATEWAY with OP_CRESR:

8.13 OP_CRESR (Status Counter Reset Response)

Total length: 8 Bytes

0	Byte	3
Packet length (incl. header)	Opcode OP_CRESR (0x6003)	
Security code		

Description: with this Opcode the SH ARC-GATEWAY confirms the reset of the counters.

8.14 OP_WQ (Write Request)

Total length: >= 8 Bytes

0		Byte		3	
Packet length (incl. header)			Opcode 'W','Q' (0x5751)		
Security code					
SID (Sender node ID)		DID (Destination node ID)		Length of the data packet	
ARCNET data packet of given length ...					

Description: by means of this Opcode an ARCNET data packet is being transmitted.

OP_WQ is being acknowledged by the SH ARC-GATEWAY with OP_WO:

8.15 OP_WO (Write O.K.)

Total length: 8 Bytes

0		Byte		3	
Packet length (incl. header)			Opcode 'W','O' (0x574F)		
Transmission state 0x00: successful 0x01: time-out, EXCNAK or RECON 0x02: incomplete 0x03: ARCNET transmission failed					

Description: having received an OP_WO SH ARC-GATEWAY may either response immediately (transmission state = 2) or response only after the transmission is completed (transmission state = 1 or 2).

For further information please see „Flag Status Query“ or „Wake On Transmit Complete“.

8.16 OP_DA (Data Advise)

Total length: 43 Bytes

0		Byte		3			
Packet length (incl. header)			Opcode 'D','A' (0x4441)				
Security code							
da_map[0]		da_map[1]		da_map[2]		da_map[3]	
da_map[4] ... da_map[27]							
da_map[28]		da_map[29]		da_map[30]		da_map[31]	
da_time: time until request is erased in 1/10 s (0: unlimited)				da_rcvall: Receive-all			

Description: this Opcode enables the reception of ARCNET packets from specific nodes. Previous settings are being erased.

The flag "da_rcvall" is present only for compatibility reasons. The Receive-all mode is set using the web UI or the configuration files.

In Receive-all mode the client records the data packets from all nodes given by "da_map".

In the array "da_map" every bit represents one ARCNET node. If the bit for a node is set, all data packets sent by this are being recorded. The bits are being mapped as follows:

- Bit 7 in da_map[0]: node ID 0
- Bit 0 in da_map[0]: node ID 7
- Bit 7 in da_map[31]: node ID 248
- Bit 0 in da_map[31]: node ID 255

8.17 OP_DQ (Data Request)

Total length: 10 Bytes

0	<i>Byte</i>		3
Packet length (incl. header)		Opcode 'D','Q' (0x4451)	
Security code			
ARCNET node ID	Required number of packets		

Description: Using this Opcode an arbitrary number of ARCNET packets sent to the node with the specified node ID can be retrieved.

If OP_DA has been used to enable data recording from selected nodes, OP_DQ will be ignored.

OP_DQ is being acknowledged by SH ARC-GATEWAY with OP_DR:

8.18 OP_DR (Data Response)

Total length: 13 >= Bytes

0	<i>Byte</i>			3
Packet length (incl. header)		Opcode 'D','R' (0x4452)		
Unambiguous ID of an ARCNET telegram				
Must be 0x00	SID	DID	Packet length lower byte	
Packet length upper byte	ARCNET data packet			

Description: ARCNET packets are being relayed in the given format.

8.19 OP_NQ (Nodemap Request)

Total length: 8 Bytes

0	Byte	3
Packet length (incl. header)		Opcode 'N','Q' (0x4E51)
Security code		

Description: this Opcode asks for the current node list (Nodemap) of the SH ARC-GATEWAY.

It is being acknowledged by the SH ARC-GATEWAY with OP_NR:

8.20 OP_NR (Nodemap Response)

Total length: 36 Bytes

Packet length (incl. header)		Opcode 'N','R' (0x4E52)	
nr_map[0]	nr_map[1]	nr_map[2]	nr_map[3]
nr_map[4] ... nr_map[27]			
nr_map[28]	nr_map[29]	nr_map[30]	nr_map[31]

Description: this Opcode contains the 32 Bytes Array „nr_map“, each bit of which informs about the presence of one specific node in the ARCNET network. If a node is present the respective bit will be set. Bit 1 in nr_map[0] correspond to the node ID 1, bit 7 in nr_map[31] to node ID 255.

8.21 OP_NDQ (TCP NODEDELAY Setting)

Total length: 10 Bytes

0	Byte	3
Packet length (incl. header)		Opcode NDQ (0x6000)
Security code		
Operation 0x00: off 0x01: on 0x02: status request		

Description: with TCP connections multiple ARCNET packets may be bundled to one single TCP packet. The TCP stack determines when such a packet will be sent.

For ARCNET packets to be forwarded immediately after reception the option "TCP NODEDELAY" must be enabled.

OP_NDQ is being acknowledged by the SH ARC-GATEWAY with OP_NDR:

8.22 OP_NDR (TCP NODEDELAY Response)

Total length: 6 Bytes

0	Byte	3
Packet length (incl. header)		Opcode NDR (0x6001)
State 0x00: off 0x01: on		

8.23 OP_REGQ (ARCNET Register Read/Write)

Total length: 11 Bytes

0	Byte		3
Packet length (incl. header)		Opcode REGQ (0x6004)	
Security code			
Operation 0x00: read; 0x01: write	ARCNET controller register	Value	

Description: by means of this Opcode the registers of the ARCNET controller of the SH ARC-GATEWAY may be written and read.

OP_REGQ is being acknowledged by SH ARC-GATEWAY with OP_REGR:

8.24 OP_REGR (ARCNET Register Read/Write Response)

Total length: 5 Bytes

0	Byte		3
Packet length (incl. header)		Opcode REGR (0x6005)	
Value			

8.25 OP_FLAGQ (Flag Status Query)

Total length: 8 Bytes

0		<i>Byte</i>		3	
Packet length (incl. header)			Opcode FLAGQ (0x6006)		
Security code					

Description: this Opcode asks for the state of a current ARCNET transmission and is being acknowledged by the SH ARC-GATEWAY with OP_FLAGR:

8.26 OP_FLAGR (Flag Status Response)

Total length: 10 Bytes

0		<i>Byte</i>		3	
Packet length (incl. header)			Opcode FLAGQ (0x6007)		
Transmission completed	Transmission confirmed	RECON counter (lower word)			
RECON counter (upper word)					

Description: if the flag "Transmission completed" is set and the flag "Transmission confirmed" is not, possibly a broadcast has been sent on the bus. RECON counter contains the number of reconfigurations occurred since the last request.

8.27 OP_TXWAITQ (Wake On TX Complete)

Total length: 9 Bytes

0	<i>Byte</i>	3
Packet length (incl. header)		Opcode TXWAITQ (0x6008)
Security code		
Value 0x00: off 0x01: on 0x02: status request)		

Description: this Opcode causes the SH ARC-GATEWAY to only react to a “Write Request”, when the previous transmission has been completed. It is being acknowledged by SH ARC-GATEWAY with OP_TXWAITR:

8.28 OP_TXWAITR (Wake On TX Complete Response)

Total length: 5 Bytes

0	<i>Byte</i>	3
Packet length (incl. header)		Opcode TXWAITR (0x6009)
Value 0x00: off 0x01: on		

8.29 OP_RECONQ (Wake On RECON)

Total length: 9 Bytes

0	<i>Byte</i>	3
Packet length (incl. header)		Opcode RECONQ (0x6010)
Security code		
Value 0x00: off 0x01: on 0x02: status request		

Description: this Opcode causes the SH ARC-GATEWAY to send a message when a reconfiguration occurs.

It is being acknowledged by the SH ARC-GATEWAY with OP_RECONR:

8.30 OP_RECONR (Wake On RECON Response)

Total length: 5 Bytes

0	<i>Byte</i>	3
Packet length (incl. header)		Opcode RECONR (0x6011)
Value 0x00: off 0x01: on; 0x02: error		

Description: with "Wake On Recon" being enabled SH ARC-GATEWAY will send an OP_RECONM when a reconfiguration occurs:

8.31 OP_RECONM (RECON Message)

Total length: 4 Bytes

0	<i>Byte</i>	3
Packet length (incl. header)	Opcode RECONM (0x6012)	

8.32 OP_TXTIMEOQ (Transmit Timeout)

Total length: 12 Bytes

0	<i>Byte</i>	3
Packet length (incl. header)	Opcode TXTIMEOQ (0x6013)	
Security code		
Timeout time in milliseconds		

Description: this Opcode sets the time-out time for ARCNET transmissions (65535 ms max.).

It is being acknowledged by the SH ARC-GATEWAY with OP_TXTIMEOR:

8.33 OP_TXTIMEOR (Transmit Timeout Response)

Total length: 8 Bytes

0	<i>Byte</i>	3
Packet length (incl. header)	Opcode TXTIMEOR (0x6014)	
Feedback 0x00: successful 0x01: failed		

Subject to technical changes and printing errors excepted

Release: November 2020