Was wird mindestens benötigt?

- PC mit Netzwerkverbindung
- Netzgerät oder el. Last mit Ethernet-Karte IF-E1 oder IF-E2

Was ist zu tun?

1. Netzwerkparameter des Gerätes konfigurieren

Die Standard-IP nach Auslieferung ist 10.0.0.1. Das paßt in den wenigsten Fällen. Sollten Sie jedoch Zugriff auf diese Adresse haben, öffnen Sie sie in einem Browser (vorzugsweise Firefox). Es erscheint die Weboberfläche der Ethernet-Karte. Dort können Sie auf der Seite „CONFIGURATION“ die Netzwerkparameter IP, Subnetzmaske und Gateway Ihren Bedürfnissen anpassen. Nach Klick auf den Knopf „SAVE“ werden die Einstellungen in das Gerät geschrieben, jedoch noch nicht aktiviert. Aktivierung erfolgt durch Ausschalten und erneutes Einschalten des Gerätes. Währenddessen kann man auch gleich den IP-Bereich am PC wechseln.

Achtung! Die Netzwerkparameter gelten für das Gerät und werden im Gerät dauerhaft gespeichert, nicht auf der Karte!

Alternativ können die Netzwerkparameter auch über die an der Karte befindliche USB-Schnittstelle und das auf der beiliegenden CD befindliche Tool „IP-Config“ vorkonfiguriert werden.

2. Neue IP testen

Nach dem erneuten Einschalten führen Sie bitte zunächst einen Test mit dem Browser (Firefox) durch. Geben Sie in die Adressleiste die neue IP ein, die Sie vergeben haben. Es sollte nach einer kurzen Wartezeit die IF-E1 bzw. IF-E2 Webseite angezeigt werden.

3. Zugriff über TCP/IP

Für den Zugriff über TCP/IP in Programmierumgebungen wie Visual Basic, C# usw. sind folgende Dinge zu beachten:

- Port: 80
- Raw Socket
- Timing wichtig, Wartezeiten für das Auslesen für Antworten sind zu berücksichtigen (Anfrageintervall: max. alle 300ms)

3.1 Senden von Befehlen

Die Karte erfordert es, einen Befehl in zwei Packets zu senden.

Das erste Packet enthält einen 32Bit-Wert (big endian), der angibt, wie lang der Befehl in Bytes ist. Wollte man beispielsweise die Fernsteuerung mit „LOCK 1“ aktivieren, so wären das 6 ASCII-Zeichen (Befehlsstrings werden hier ohne Endezeichen 0x00 (EOL) gesendet), also 6 Bytes. Der 32Bit-Wert wäre demnach 0x00000006.

Danach, im zweiten Packet, wird der eigentliche Befehl gesendet. Wie bereits erwähnt, hier nur der Befehlsstring in ASCII-zeichen, ohne Abschlußbyte 0x00. Für den Befehl „LOCK 1“ würden also 0x4C 0x4F 0x43 0x4B 0x20 0x31 gesendet.

Beispiel aus Wireshark, 1. Packet:

```
⊞ Frame 6: 58 bytes on wire (464 bits), 58 bytes captured (464 bits)
⊞ Ethernet II, Src: HewlettP_f2:71:f6 (18:a9:05:f2:71:f6), Dst: 37:35:30:30:30:36 (37:35:30:30:30:36)
⊞ Internet Protocol, Src: 99.0.7.112 (99.0.7.112), Dst: 99.0.7.139 (99.0.7.139)
⊞ Transmission Control Protocol, Src Port: 50311 (50311), Dst Port: http (80), Seq: 1, Ack: 1, Len: 4
⊟ Hypertext Transfer Protocol
  ⊟ Data (4 bytes)
      Data: 00000006
      [Length: 4]

0000  37 35 30 30 30 36 18 a9  05 f2 71 f6 08 00 45 00   750006.. ..q...E.
0010  00 2c 21 af 40 00 80 06  00 00 63 00 07 70 63 00   .,!.@... ..c..pc.
0020  07 8b c4 87 00 50 c9 f1  a5 36 60 5d 00 00 50 18   .....P.. .6`]..P.
0030  fa f0 d5 19 00 00 00 00  00 06                     ........ ..
```

Weiter nächste Seite...

Beispiel aus Wireshark, 2. Packet:

```
⊞ Frame 8: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
⊞ Ethernet II, Src: HewlettP_f2:71:f6 (18:a9:05:f2:71:f6), Dst: 37:35:30:30:30:36 (37:35:30:30:30:36)
⊞ Internet Protocol, Src: 99.0.7.112 (99.0.7.112), Dst: 99.0.7.139 (99.0.7.139)
⊟ Transmission Control Protocol, Src Port: 50433 (50433), Dst Port: http (80), Seq: 5, Ack: 1, Len: 6
    Source port: 50433 (50433)
    Destination port: http (80)
    [Stream index: 0]
    Sequence number: 5    (relative sequence number)
    [Next sequence number: 11    (relative sequence number)]
    Acknowledgement number: 1    (relative ack number)
    Header length: 20 bytes
  ⊞ Flags: 0x18 (PSH, ACK)
    Window size: 64240
  ⊞ Checksum: 0xd51b [validation disabled]
  ⊞ [SEQ/ACK analysis]
    TCP segment data (6 bytes)

0000  37 35 30 30 30 36 18 a9  05 f2 71 f6 08 00 45 00   750006.. ..q...E.
0010  00 2e 7f c9 40 00 80 06  00 00 63 00 07 70 63 00   ....@... ..c..pc.
0020  07 8b c5 01 00 50 26 12  c5 6a 62 2e 00 00 50 18   .....P&. .jb...P.
0030  fa f0 d5 1b 00 00 4c 4f  43 4b 20 31               ......LO CK 1
```

3.2 Empfangen von Antworten

Die Vorgehensweise ist hier wie beim Senden, nur daß man mit dem nächsten Packet einen 32Bit-Wert ausliest, der die Länge der im nächsten Packet folgenden Antwort enthält.

Generell kann bzw. sollte das Abfragen der Länge der Antwort immer erfolgen, auch bei Befehlen die keine Antwort senden. Die Länge der Antwort würde dann mit 0x00000000 zurückgegeben.

Für den Befehl „*IDN?“, der die Geräte-Identifikation zurückliefert, ergäbe sich dann folgende Packetreihenfolge:

1. 32Bit-Wert des Anfragebefehls senden (Länge: 5)
2. Befehl *IDN? senden und etwas warten
3. 32Bit-Wert der Antwortlänge auslesen, z. B. 0x00000037
4. Antwort lesen mit der Länge, die in Schritt 3 ausgelesen wurde

Beispiel aus Wireshark, 3. Packet (Packet 1 und 2 wie oben, nur anderer Befehl):

```
⊞ Frame 10: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
⊞ Ethernet II, Src: 37:35:30:30:30:36 (37:35:30:30:30:36), Dst: HewlettP_f2:71:f6 (18:a9:05:f2:71:f6)
⊞ Internet Protocol, Src: 99.0.7.139 (99.0.7.139), Dst: 99.0.7.112 (99.0.7.112)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 50413 (50413), Seq: 1, Ack: 10, Len: 4
⊟ Hypertext Transfer Protocol
  ⊟ Data (4 bytes)
      Data: 00000037
      [Length: 4]

0000  18 a9 05 f2 71 f6 37 35  30 30 30 36 08 00 45 00   ....q.75 0006..E.
0010  00 2c 00 24 00 00 80 06  65 ad 63 00 07 8b 63 00   .,.$.... e.c...c.
0020  07 70 00 50 c4 ed 55 2b  00 00 d7 d8 6d ad 50 18   .p.P..U+ ....m.P.
0030  16 d0 63 d7 00 00 00 00  00 37 00 00               ..c...... .7..
```

Beispiel aus Wireshark, Antwort-Packet(s) vom Gerät:

```
⊞ Frame 12: 105 bytes on wire (840 bits), 105 bytes captured (840 bits)
⊞ Ethernet II, Src: 37:35:30:30:30:36 (37:35:30:30:30:36), Dst: HewlettP_f2:71:f6 (18:a9:05:f2:71:f6)
⊞ Internet Protocol, Src: 99.0.7.139 (99.0.7.139), Dst: 99.0.7.112 (99.0.7.112)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 50375 (50375), Seq: 5, Ack: 10, Len: 51
⊟ Hypertext Transfer Protocol
  ⊟ Data (51 bytes)
      Data: 3b454120566965727365 6e3b454c20333136302d3630413b...
      [Length: 51]

0000  18 a9 05 f2 71 f6 37 35  30 30 30 36 08 00 45 00   ....q.75 0006..E.
0010  00 5b 00 0d 00 00 80 06  65 95 63 00 07 8b 63 00   .[...... e.c...c.
0020  07 70 00 50 c4 c7 21 73  00 04 33 a8 45 5c 50 10   .p.P..!s ..3.E\P.
0030  16 d0 3f 63 00 00 3b 45  41 20 56 69 65 72 73 65   ..?c..;E A Vierse
0040  6e 3b 45 4c 20 33 31 36  30 2d 36 30 41 3b 30 30   n;EL 316 0-60A;00
0050  30 30 30 30 30 30 30 30  3b 56 34 2e 31 36 20 32   00000000 ;V4.16 2
0060  36 2e 30 37 2e 31 30 3b  56                        6.07.10; V
```

Die Antwort ist hier aufgeteilt in zwei aufeinanderfolgenden Packets. Die restlichen 4 Bytes sind dann im zweiten.

Minimum requirements

- PC with network access
- Power supply unit or electronic load with Ethernet card IF-E1 or IF-E2

What to do?

1. Configure the network parameters of the device

The default IP of your card is 10.0.0.1. This won't be suitable in most situations. In case you can directly access this IP, open a browser (preferrably Firefox) and enter the IP in the URL box. A webpage should be loaded after a few seconds. Clicking the button „CONFIGURATION" will display the current network parameters of the device. They can be changed to meet your requirements and then saved with the „SAVE" button. This will write the new settings permanently to the device, but not activate them instantly. Activation of the new settings requires to switch the device off and on again. In the meantime you could, for example, change the IP settings on the PC.

Attention! The network parameters are stored inside the device, not inside the card!

Alternatively, those parameters can be preconfigured via the additional USB port and the tool „IP-Config", which is available on the included Tools CD.

2. Test the new IP

After switching the device on, you should test the new IP first. Open a browser (Firefox is recommended) and enter the new IP in the URL box. After a short while, the webpage of IF-E1 resp. IF-E2 should be loaded and displayed.

3. Access via TCP/IP

For the access via TCP/IP in programming environments like Visual Basic, C# etc., following is given:

- Port: 80
- Raw Socket
- Timing is important, you need to put delays before reading responses (request interval: min. 300ms)

3.1 Sending commands

The card requires you to send a command with two packets.

The 1st packet must contain a 32 bit value (big endian), which defines the length of the command. If you want to, for example, activate remote control with „LOCK 1", this would be 6 ASCII characters. Command strings are here transferred without terminating character (0x00, EOL), so it is 6 Bytes. The 32 bit value would then be 0x00000006.

Then, in the 2nd packet, the actual command is sent. Like already stated above, only send the command in ASCII bytes, without EOL. The command „LOCK 1" would thus result in 0x4C 0x4F 0x43 0x4B 0x20 0x31.

Example from Wireshark, 1st packet:

```
⊞ Frame 6: 58 bytes on wire (464 bits), 58 bytes captured (464 bits)
⊞ Ethernet II, Src: HewlettP_f2:71:f6 (18:a9:05:f2:71:f6), Dst: 37:35:30:30:30:36 (37:35:30:30:30:36)
⊞ Internet Protocol, Src: 99.0.7.112 (99.0.7.112), Dst: 99.0.7.139 (99.0.7.139)
⊞ Transmission Control Protocol, Src Port: 50311 (50311), Dst Port: http (80), Seq: 1, Ack: 1, Len: 4
⊟ Hypertext Transfer Protocol
  ⊟ Data (4 bytes)
      Data: 00000006
      [Length: 4]

0000  37 35 30 30 30 36 18 a9  05 f2 71 f6 08 00 45 00   750006.. ..q...E.
0010  00 2c 21 af 40 00 80 06  00 00 63 00 07 70 63 00   .,!.@... ..c..pc.
0020  07 8b c4 87 00 50 c9 f1  a5 36 60 5d 00 00 50 18   .....P.. .6`]..P.
0030  fa f0 d5 19 00 00 00 00  00 06                     ........ ..
```

Continued on next page...

Example from Wireshark, 2nd packet:

```
⊞ Frame 8: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
⊞ Ethernet II, Src: HewlettP_f2:71:f6 (18:a9:05:f2:71:f6), Dst: 37:35:30:30:30:36 (37:35:30:30:30:36)
⊞ Internet Protocol, Src: 99.0.7.112 (99.0.7.112), Dst: 99.0.7.139 (99.0.7.139)
⊟ Transmission Control Protocol, Src Port: 50433 (50433), Dst Port: http (80), Seq: 5, Ack: 1, Len: 6
    Source port: 50433 (50433)
    Destination port: http (80)
    [Stream index: 0]
    Sequence number: 5    (relative sequence number)
    [Next sequence number: 11    (relative sequence number)]
    Acknowledgement number: 1    (relative ack number)
    Header length: 20 bytes
  ⊞ Flags: 0x18 (PSH, ACK)
    Window size: 64240
  ⊞ Checksum: 0xd51b [validation disabled]
  ⊞ [SEQ/ACK analysis]
    TCP segment data (6 bytes)

0000  37 35 30 30 30 36 18 a9  05 f2 71 f6 08 00 45 00   750006.. ..q...E.
0010  00 2e 7f c9 40 00 80 06  00 00 63 00 07 70 63 00   ....@... ..c..pc.
0020  07 8b c5 01 00 50 26 12  c5 6a 62 2e 00 00 50 18   .....P&. .jb...P.
0030  fa f0 d5 1b 00 00 4c 4f  43 4b 20 31               ......LO CK 1
```

3.2 Reading responses

Generally, the procedure here is the same as with sending commands. The difference is just that you read a 32 bit value with the next packet which tells you the length of the following response string in the subsequent packet(s).

It is recommended to query the length always after sending a command. In case there is no response to the last command, the length of the response would be returned as 0x00000000.

For the command „*IDN?“, which returns the device identification, following packet order would be required:

1. Send 32 bit value with the command length (here: 5)
2. Send command *IDN? and wait a little
3. Read 32 bit value with the response string length, for example 0x00000037
4. Read the response string by requesting that number of bytes, which was read in step 3.

Example from Wireshark, 3rd packet (packet 1 and 2 like above, just a different command):

```
⊞ Frame 10: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
⊞ Ethernet II, Src: 37:35:30:30:30:36 (37:35:30:30:30:36), Dst: HewlettP_f2:71:f6 (18:a9:05:f2:71:f6)
⊞ Internet Protocol, Src: 99.0.7.139 (99.0.7.139), Dst: 99.0.7.112 (99.0.7.112)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 50413 (50413), Seq: 1, Ack: 10, Len: 4
⊟ Hypertext Transfer Protocol
  ⊟ Data (4 bytes)
      Data: 00000037
      [Length: 4]

0000  18 a9 05 f2 71 f6 37 35  30 30 30 36 08 00 45 00   ....q.75 0006..E.
0010  00 2c 00 24 00 00 80 06  65 ad 63 00 07 8b 63 00   .,.$.... e.c...c.
0020  07 70 00 50 c4 ed 55 2b  00 00 d7 d8 6d ad 50 18   .p.P..U+ ....m.P.
0030  16 d0 63 d7 00 00 00 00  00 37 00 00               ..c...... .7..
```

Example from Wireshark, response packet(s) from device:

```
⊞ Frame 12: 105 bytes on wire (840 bits), 105 bytes captured (840 bits)
⊞ Ethernet II, Src: 37:35:30:30:30:36 (37:35:30:30:30:36), Dst: HewlettP_f2:71:f6 (18:a9:05:f2:71:f6)
⊞ Internet Protocol, Src: 99.0.7.139 (99.0.7.139), Dst: 99.0.7.112 (99.0.7.112)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 50375 (50375), Seq: 5, Ack: 10, Len: 51
⊟ Hypertext Transfer Protocol
  ⊟ Data (51 bytes)
      Data: 3b4541205669657273656e3b454c20333136302d3630413b...
      [Length: 51]

0000  18 a9 05 f2 71 f6 37 35  30 30 30 36 08 00 45 00   ....q.75 0006..E.
0010  00 5b 00 0d 00 00 80 06  65 95 63 00 07 8b 63 00   .[...... e.c...c.
0020  07 70 00 50 c4 c7 21 73  00 04 33 a8 45 5c 50 10   .p.P..!s ..3.E\P.
0030  16 d0 3f 63 00 00 3b 45  41 20 56 69 65 72 73 65   ..?c..;E A Vierse
0040  6e 3b 45 4c 20 33 31 36  30 2d 36 30 41 3b 30 30   n;EL 316 0-60A;00
0050  30 30 30 30 30 30 30 30  3b 56 34 2e 31 36 20 32   00000000 ;V4.16 2
0060  36 2e 30 37 2e 31 30 3b  56                        6.07.10; V
```

The response may be splitted into two subsequent packets. In the example, the remaining 4 bytes are in the second one.