

Vielen Dank für Ihr Interesse, wie Telefonfachleute für das Programmieren einer Telefonanlage unterstützt wurden.



Verstanden beim  
ersten Lesen

**„Aaah!“**

**Franz Pils.**  
Partner für  
Funktionierendes

## Mit Vorher/Nachher-Beispiel.

Was es für Sie zu vergleichen und zu beurteilen gibt:

1. Den Textaufbau
2. Die Strukturierung und Differenzierung der Texte
3. Die Vollständigkeit der Unterstützung
4. Die Textlänge

**MMC: 113****VIEW MEMO NUMBER****DESCRIPTION:**

Allows a station user the ability to view a memo that the user has left. A memo can be left by entering a memo via the dial keypad using the table below. MMC 112 Alarm Reminder can be programmed to remind you to read the memo. A memo of up and including fourteen characters can be entered. Using the dial keypad, press "6" two times to enter the letter "N," and press "3" two times for an "e." Pressing the "A" key will change the letters from upper case to lower case.

NOTE: When the character you want appears on the same dial pad key as the previous character, press the UP key to move the cursor to the right.

COUNT	1	2	3	4	5
DIAL 0	Q	Z	.	)	0
DIAL 1	space	?	,	!	1
DIAL 2	A	B	C	@	2
DIAL 3	D	E	F	#	3
DIAL 4	G	H	I	\$	4
DIAL 5	J	K	L	%	5
DIAL 6	M	N	O	^	6
DIAL 7	P	R	S	&	7
DIAL 8	T	U	V	*	8
DIAL 9	W	X	Y	(	9
DIAL *	:	=	[	]	*

The # key can be used for the following special characters: #, space, &, !, :, ?, ., %, \$, -, <, >, /, =, [ ], @, ^, (, ), \_ , +, {, }, |, ;, \, " and ~.

**PROGRAM KEYS**

UP & DOWN	Used to scroll through options
KEYPAD	Used to enter selections
SOFT KEYS	Move cursor left and right
SPK	Used to store data and advance to next MMC
HOLD	Used to clear previous entry

**ACTION****DISPLAY**

- Press TRSF 113  
Display shows
- Press RIGHT soft key to move cursor and add memo via dial key pad with above list  
Press RIGHT soft key to return to step 2
- Press TRSF to store and exit  
OR  
Press SPK to store and advance to next MMC

```
[201] VIEW MEMO
1:
```

```
[201] VIEW MEMO
1:NEED BREAD
```

# Programmier-Code

# 113

## MEMO

Individuelle  
(nebenstellenspezifische)  
Memos eingeben

DCS COMPACT  
Oktober 1998

**Funktionsbeschreibung**

- Memo eingeben; maximale Länge 13 Zeichen.
- 3 Speicherplätze verfügbar.
- Funktion 112 (Terminerinnerung) erinnert an das Lesen einer Nachricht.

**Standardeinstellungen**

Keine.

zu Kapitel 3.1 siehe nächste Seite

**Programmierung (Prg.)**

Start in Kapitel 3.1 beschrieben.

- Funktionsspezifische Prg. einleiten:

Programmier-Code **113** eintippen.

```
[201] MEMO LESEN
1:
```

- Nebenstelle bestimmen:

Nummer eintippen (z. B. 205);

– oder

mit Tasten + / - aussuchen; Auswahl bestätigen mit **rechtem Softkey**.

```
[205] MEMO LESEN
1:
```

- Texteingabe:

Je Zeichen eine Wähltaste **1x** bis **6x** drücken. Anleitung mit ausführlichem Beispiel unter Kapitel 3.4.

	1x	2x	3x	4x	5x	6x
Wähltaste <b>0</b>	Q	Z	.	)	0	
Wähltaste <b>1</b>	Leerzeichen	?	,	!	1	
Wähltaste <b>2</b>	A	B	C	@	2	
Wähltaste <b>3</b>	D	E	F	#	3	
Wähltaste <b>4</b>	G	H	I	\$	4	
Wähltaste <b>5</b>	J	K	L	%	5	
Wähltaste <b>6</b>	M	N	O	^	6	
Wähltaste <b>7</b>	P	Q	R	S	&	7
Wähltaste <b>8</b>	T	U	V	*	8	
Wähltaste <b>9</b>	W	X	Y	Z	(	9
Wähltaste <b>*</b>	:	=	[	]	*	

Wähltaste # enthält zusätzliche Sonderzeichen.

Weitere Memo-Zuordnungen:

Zurück zu Schritt 2. mit **rechtem Softkey**.

- Prg. abspeichern bzw. abschließen:

Taste **FREISPR.** bzw. **ÜBERGABE**.

**Hinweis:**

Für einfaches Lesen der Memos ist eine Funktionstaste programmierbar.

# 3 Programmieranweisungen, spezielle Tastenfunktionen

---

## 3.1 Einleiten der funktionspezifischen Programmierung

---

Das Programmieren der TK-Anlage DCS-Compact wird rasch zur einfachen Routine, wenn Sie sich erst einmal das Prinzipielle eingeprägt haben und sich strikt an die vorgegebenen Abläufe halten:

1. Programmieren ist nur bei aufgelegtem Hörer möglich.
2. Die Tasten **ÜBERGABE** und **FREISPR.** haben eine Art Schlüsselfunktion.

Den eigentlichen Programmierablauf können Sie in drei Phasen untergliedern. Wir veranschaulichen das am Beispiel eines Türschlüssels, der dreimal im Schloß herumzudrehen ist.

---

### Erste „Schlüsselumdrehung“:

---

Taste **ÜBERGABE** drücken. Die folgenden Eingaben:

- Ziffern **800** (Programmier-Code des Servicetechnikers).
- 4stelliger Zugangs-Code (Passwort).
- Ziffer **1**.

---

### Zweite „Schlüsselumdrehung“:

---

Taste **FREISPR.** drücken. Die folgende Eingabe:

- 3stelliger Programmier-Code. Er leitet Ihre funktionspezifische Programmierung ein.

Beachten Sie: Wir beginnen die ausführliche Erläuterung aller Programmierungen stets **mit dieser zweiten „Schlüsselumdrehung“**, das heißt

- mit der Eingabe des funktionspezifischen Programmier-Codes nach dem Drücken der Taste **FREISPR.**

---

### Dritte und letzte „Schlüsselumdrehung“:

---

- Taste **ÜBERGABE** nach der Programmierung erneut drücken. Damit schließen Sie die Programmierung ab.

Alternativ können Sie auch Taste **FREISPR.** drücken, wenn Sie lediglich abspeichern wollen und dann Ihre Programmierung fortsetzen möchten.

Beachten Sie bitte auch, spätestens 30 Sekunden nach dem Starten des Programmier-Modus mit Ihrer Eingabe zu beginnen. Vermeiden Sie Pausen zwischen zwei Tastendrücken, die länger sind als 30 Sekunden. Anderenfalls wird der Programmier-Modus – aus Sicherheitsgründen – automatisch beendet.

Diese maximalen Pausenzeiten zwischen zwei Eingaben sind eine Standardeinstellung. Wenn Sie wollen, können Sie stattdessen beliebig lange Pausenzeiten einstellen: zwischen 10 und 255 Sekunden. Erläuterungen hierzu unter Programmier-Code **500**, Option **31**.

## Beachten Sie bitte:

Mit dem Hinweis auf Kap. 3.1 (siehe gegenüberliegende Seite) erfüllte ich meinen Anspruch,

- stets in sich geschlossene Arbeitsvorgänge darzustellen, auch wenn es „nur“ die Funktionalität eines Programmier-Codes betrifft.

Außerdem: Korrektes Einleiten einer funktionspezifischen Programmierung hat Schlüsselbedeutung. Unkorrekt eingeleitete Programmierungen bedeuten Fehler und Zeitverluste.

In der Vorlage hatte der Technische Redakteur mit der Eingabe des funktionspezifischen Codes begonnen – ohne Bezug zu nehmen auf die vorhergehenden Startschritte der funktionspezifischen Programmierung.