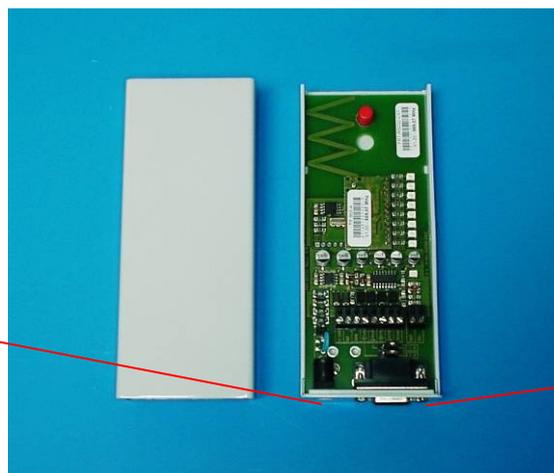
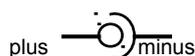


Antennen-Modul
L142 x B58 x H24

Buchse für
Stecker-Netzteil
230Vac/15Vdc



RS232-Schnittstelle
Sub D 9-pol. fem.

Das Antennen-Modul wird für guten Funkempfang möglichst hoch an der Wand montiert, und über das mitgelieferte Steckernetzteil (230Vac/15Vdc) versorgt. Es empfängt die codierten Funk-Rufe der individuellen Tischsender TS-xx und gibt diese anhand des nachfolgenden Protokolls an der RS232-Schnittstelle aus. Im übergeordneten System (PC, Kassensystem etc.) können dann diese Daten übernommen, den Tischnummern zugeordnet und weitere Prozeduren davon abgeleitet werden.



Das Antennen-Modul gibt es in dem o.g. Frequenzbereich passend zu den jeweiligen Funk-Ruf-Knöpfen.

Hinweis: Elektro-Altgeräte sind an Sammelstellen abzugeben, nicht mit dem Hausmüll zu entsorgen.

Datenrahmen Antennenmodul

Automatisch versendete Daten für jeden erkannten Funkrufknopf:
(Daten von dem Antennenmodul zum PC o.ä.)

				Inhalt
0	SOH	Start of Header (01 hex)		0x01 SOH
1	Dummy	Konstante		0x5a Z
2	ADR1low	Adresse Byte 1	} Byte 1 bis 8 sind die individuellen Adress-Daten der Funk-Ruf-Knöpfe	0x30 0 0x30-0x3f
3	ADR1high	Adresse Byte 2		0x30 0 0x30-0x3f
4	ADR2low	Adresse Byte 3		0x30 0 0x30-0x3f
5	ADR2high	Adresse Byte 4		0x30 0 0x30-0x3f
6	ADR3low	Adresse Byte 5		0x30 0 0x30-0x3f
7	ADR3high	Adresse Byte 6		0x30 0 0x30-0x3f
8	ADR4low	Adresse Byte 7		0x30 0 0x30-0x3f
9	ADR4high	Adresse Byte 8		0x30 0 0x30-0x3f
10	ADR5low	Adresse Byte 9	} Diese Bytes <u>nicht</u> benutzen ! (Diese sind für spätere Applikationen vorgesehen)	0x30 0 0x30-0x3f
11	ADR5high	Adresse Byte 10		0x30 0 0x30-0x3f
12	ADR6low	Adresse Byte 11		0x30 0 0x30-0x3f
13	ADR6high	Adresse Byte 12		0x30 0 0x30-0x3f
14	Statuslow	Status Byte 1		0x30 0 0x30-0x3f
15	Statushigh	Status Byte 1		0x30 0 0x30-0x3f
16	ETX	End of Text (03 hex)		0x03 ETX
17	CHK	Checksumme (4 Bytes) :		0xXX 0xXX
18	CHK	Beinhaltet alle Bytes von STX bis ETX		0xXX 0xXX
19	CHK	inklusive dieser 2 Steuerzeichen		0xXX 0xXX
20	CHK	Umgerechnet auf 4 Byte ASCII		0xXX 0xXX
21	EOT	End of Transmission (04 hex)		0x04 EOT

Schnittstelle : RS232 / 9600 Baud / 8 Daten / 1 Stop / Keine Parität
Statusbyte für zukünftige Versionen, um z.B. Low-Battery auszuwerten

Senderoutine der RS232-Antenne:

```

void tx_daten(void){
    unsigned char i;

    tx_anzahl=0;

    buffer[tx_anzahl++]=0x01;// Start of Header
    buffer[tx_anzahl++]=0x5A;// Dummy

    buffer[tx_anzahl++]=LOW_HEX(adr_1_8);// Adresse LowerNibble (HEX)
    buffer[tx_anzahl++]=HIGH_HEX(adr_1_8);// Adresse Higher Nibble (HEX)

    buffer[tx_anzahl++]=LOW_HEX(adr_9_16);// Adresse LowerNibble (HEX)
    buffer[tx_anzahl++]=HIGH_HEX(adr_9_16);// Adresse Higher Nibble (HEX)

    buffer[tx_anzahl++]=LOW_HEX(adr_17_24);// Adresse LowerNibble (HEX)
    buffer[tx_anzahl++]=HIGH_HEX(adr_17_24);// Adresse Higher Nibble (HEX)

    buffer[tx_anzahl++]=LOW_HEX(adr_25_32);// Adresse Lower Nibble (HEX)
    buffer[tx_anzahl++]=HIGH_HEX(adr_25_32);// Adresse Higher Nibble (HEX)

    buffer[tx_anzahl++]=LOW_HEX(adr_33_40);// Adresse Lower Nibble (HEX)
    buffer[tx_anzahl++]=HIGH_HEX(adr_33_40);// Adresse Higher Nibble (HEX)

    buffer[tx_anzahl++]=LOW_HEX(adr_41_48);// Adresse Lower Nibble (HEX)
    buffer[tx_anzahl++]=HIGH_HEX(adr_41_48);// Adresse Higher Nibble (HEX)

    buffer[tx_anzahl++]=LOW_HEX(0); // Statusbyte
    buffer[tx_anzahl++]=HIGH_HEX(0); // Statusbyte

    buffer[tx_anzahl++]=0x03; // End of Text

    for (i=1,tx_summe=0;i<=tx_anzahl-1;i++){
        tx_summe+=(unsigned int)buffer[i];
    }

    sprintf (hex,"%04x",tx_summe);

    buffer[tx_anzahl++]=hex[0];
    buffer[tx_anzahl++]=hex[1];
    buffer[tx_anzahl++]=hex[2];
    buffer[tx_anzahl++]=hex[3];

    buffer[tx_anzahl++]=0x04; // EOT

    for (i=0;i<tx_anzahl;i++){
        putchar(buffer[i]);
    }
}

```

*) = Diese Bytes nicht benutzen ! Diese sind für spätere Applikationen vorgesehen.