

INITIALISIERUNGSBEISPIELE

Initialisation example: 8-Bit											
RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	HEX	Description
0	0	0	0	1	1	1	0	0	1	\$39	Function Set, western european character set, 8-Bit
0	0	0	0	0	0	1	0	0	0	\$08	Display off
0	0	0	0	0	0	0	1	1	0	\$06	Entry mode set, increment cursor by 1 not shifting display
0	0	0	0	0	1	0	1	1	1	\$17	Character mode and internal power on (have to turn on internal power to get the best brightness)
0	0	0	0	0	0	0	0	0	1	\$01	Clear display
0	0	0	0	0	0	0	0	1	0	\$02	Return home
0	0	0	0	0	0	1	1	0	0	\$0C	Display on

void initDisplay (*void*)

```
{
  RS_DD=1; //RS Pin as output
  RW_DD=1; //RW Pin as output
  EN_DD=1; //EN Pin as output
  WriteIns(0x39); //function set european character set
  WriteIns(0x08); //display off
  WriteIns(0x06); //entry mode set increment cursor by 1 not shifting display
  WriteIns(0x17); //Character mode and internal power on
  WriteIns(0x01); //clear display
  WriteIns(0x02); //return home
  WriteIns(0x0C); //display on
}
```

void WriteIns(*char* instruction)

```
{
  CheckBusy();
  DATA_PORT_DD=0xFF; //Dataport as Output
  RS = 0;
  RW = 0;
  DATA_PORT = instruction; //set Data on Outputport
  EN = 1; //set Enable to high
  Wait(10); //wait 1us (stabilize Outputport)
  EN = 0; //reset Enable to low
}
```

Initialisation example: 4-Bit											
RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	HEX	Description
0	0	0	0	1	0	1	0	0	0	\$28	Function Set, English/Japanese char set, 4-Bit Note: Western European charset not available
0	0	0	0	0	0	1	0	0	0	\$08	Display off
0	0	0	0	0	0	0	1	1	0	\$06	Entry mode set, increment cursor by 1 not shifting display
0	0	0	0	0	1	0	1	1	1	\$17	Character mode and internal power on (have to turn on internal power to get the best brightness)
0	0	0	0	0	0	0	0	0	1	\$01	Clear display
0	0	0	0	0	0	0	0	1	0	\$02	Return home
0	0	0	0	0	0	1	1	0	0	\$0C	Display on

void initDisplay(*void*)

```
{
  RS_DD=1; //RS-Pin as Output
  EN_DD=1; //EN-Pin as Output
  RW_DD=1; //RW-Pin as Output
  RS = 0; //RS-Pin to low
  RW = 0; //RW-Pin to low
  EN = 0; //EN-Pin to low
  send_nibble(0x03); //Be sure to
  send_nibble(0x03); //be in
  send_nibble(0x03); //8-Bit-Mode
  send_nibble(0x02); //Switch to 4 Bit
  Wait(50); //Wait 5us
  WriteIns(0x28); //4-Bit-Mode
  WriteIns(0x08); //display off
  WriteIns(0x06); //entry mode set increment cursor by 1 not shifting display
  WriteIns(0x17); //Character mode and internal power on
  WriteIns(0x01); //clear display
  WriteIns(0x02); //return home
  WriteIns(0x0C); //display on
}
```

void send_nibble (*char* data)

```
{
  DATA_PORT = data; //output data
  EN=1;
  Wait(10); //wait 1us (stabilize output)
  EN=0;
  Wait(10); //wait 1us (stabilize output)
}
```

void WriteIns(*char* instruction)

```
{
  CheckBusy();
  DATA_PORT_DD=0x0F; //Dataport as Output
  RS = 0;
  RW = 0;
  send_nibble((instruction&0xF0)>>4); //Highbyte
  send_nibble(instruction&0x0F); //Lowbyte
}
```