

# Anleitung Verschaltung AnaGate SPI / KMA200

## Funktionsweise

- Digital Out 1:
  - Ausgang=HIGH: Daten werden vom AnaGate SPI aus zum KMA200 geliefert.
  - Ausgang=LOW: Daten werden vom AnaGate SPI vom KMA200 gelesen.
- Digital Out 2/3: steuert die Versorgungsspannung des KMA200.

Digital out 2	Digital out 3	VDD
0	0	0V
0	1	12V
1	0	3,3V
1	1	NICHT ERLAUBT

## Programmierbeispiel (C/C++)

### Initialisierung AnaGate SPI

```
#define BAUDRATE 500000
int hHandle = 0;
int nRC = 0;

nRC = SPIOpenDevice( &hHandle, "10.1.2.197"); // mit AnaGate SPI verbinden
if (nRC == 0 )
{
    // Zugriff im normalen Betriebsmode bzw. command mode
    // siehe die entsprechenden folgenden Codesequenzen

    nRC = SPICloseDevice( &hHandle ); // Verbindung freigeben
}
```

### Normaler Betriebs-Modus

```
// 500Kbit, 3,3V Pegel, 3.3 Hilfsspannung, CPHA=1 und CPOL=1
nRC = SPISetGlobals( hHandle, BAUDRATE, 2, 0, 3);

// set digital out 2
nRC = SPIWriteDigital( hHandle, 0x02 );

char dummywritbuf[2];
char readbuf[2];

// nun Daten lesen
nRC = SPIDataReq( hHandle, dummywritbuf, 2, readbuf, 2 );
```

### Command-Modus

```
// 500Kbit, 3,3V Pegel, 3.3 Hilfsspannung, CPHA=0 und CPOL=0
nRC = SPISetGlobals( hHandle, BAUDRATE, 2, 0, 0);
nRC = SPIWriteDigital( hHandle, 0x02 ); // set digital out 2

sleep(4); // wait 4ms

// 500Kbit, 3,3V Pegel, 3.3 Hilfsspannung, CPHA=1 und CPOL=1
nRC = SPISetGlobals( hHandle, BAUDRATE, 2, 0, 3);

// jetzt sind wir im Command-Modus und können loslegen

// A: Schreibzugriff
//-----
char cmdbuffer[3];
char dummy[2];
cmdbuffer[0] = 0x01; // sensor offset 2
cmdbuffer[1] = 0x12; //
cmdbuffer[2] = 0x34;
nRC = SPIWriteDigital( hHandle, 0x03 ); // Digital 1+2 setzen
nRC = SPIDataReq( hHandle, cmdbuffer, 3, dummy, 0 );

// B: Lesezugriff
//-----
char cmdbyte = 0x17; // read control 1
```

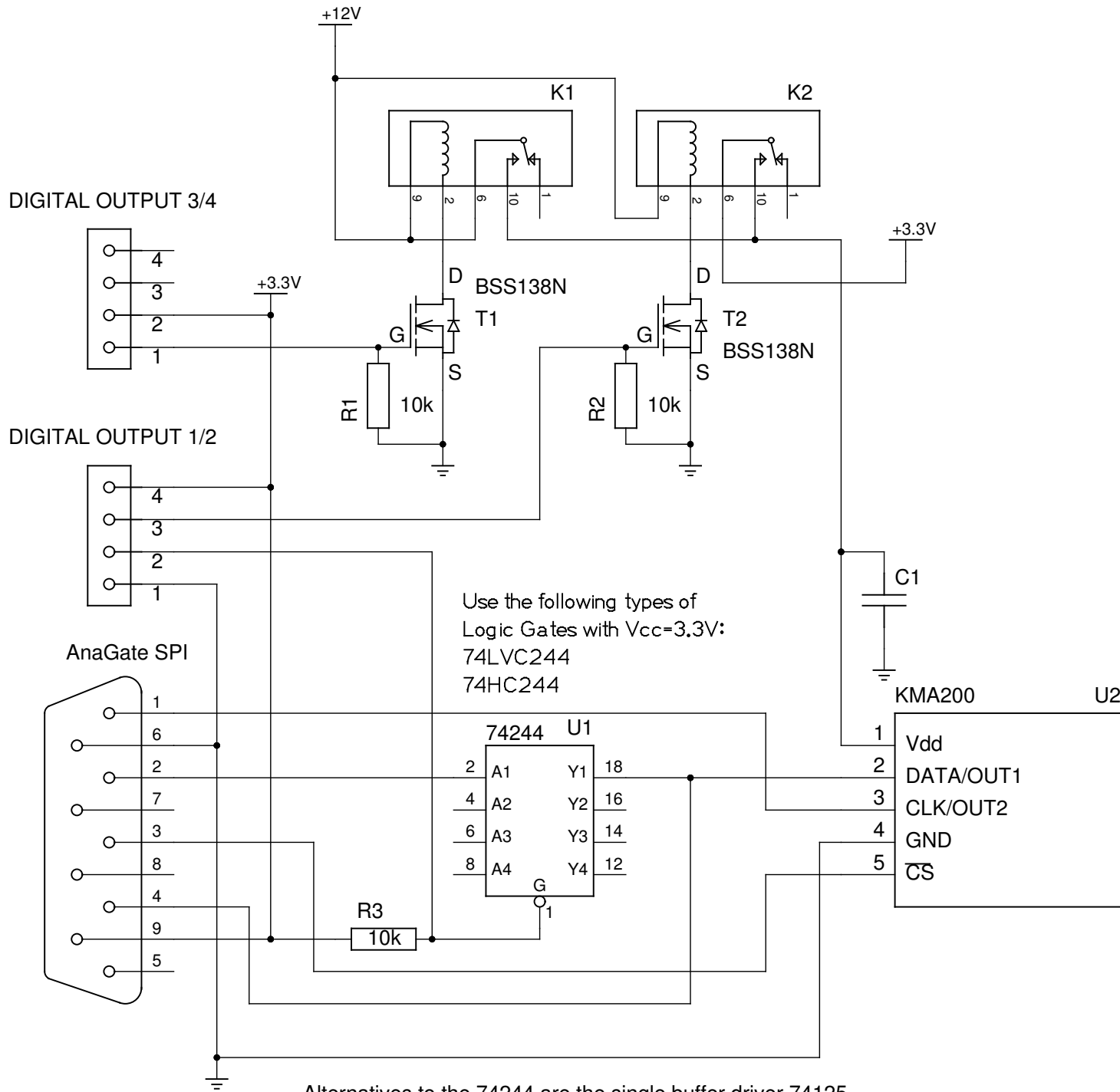
```
char readbuf[2];
nRC = SPIWriteDigital( hHandle, 0x03 ); // Digital 1+2 setzen
nRC = SPIDataReq( hHandle, &cmdbyte, 1, dummy, 0 );
nRC = SPIWriteDigital( hHandle, 0x02 ); // Digital 2 setzen
nRC = SPIDataReq( hHandle, &dummy, 2, readbuf, 2 );

// C: Programmierte Daten ins EEROM schreiben
//-----
char cmdbyte2 = 0x30; // set EEPROM
nRC = SPIWriteDigital( hHandle, 0x03 ); // Digital 1+2 setzen
nRC = SPIDataReq( hHandle, &cmdbyte2, 1, dummy, 0 );

nRC = SPIWriteDigital( hHandle, 0x04 ); // Digital 3 setzen

sleep(?); // wait ? ms

nRC = SPIWriteDigital( hHandle, 0x00); // power off device
```



Use the following types of  
Logic Gates with Vcc=3.3V:  
74LVC244  
74HC244

Alternatives to the 74244 are the single buffer driver 74125