

Connecting an AnaGate SPI to KMA200

Funktionsweise

- Digital Out 1:
 - Output=HIGH: Data are sent via AnaGate SPI to the KMA200.
 - Output=LOW: Data are read via AnaGate SPI from KMA200.
- Digital Out 2/3: controls the voltage of the KMA200.

Digital out 2	Digital out 3	VDD
0	0	0V
0	1	12V
1	0	3,3V
1	1	NOT ALLOWED

Programming Example (C/C++)

Initialization of the AnaGate SPI

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

nRC = SPIOpenDevice( &hHandle, "10.1.2.197"); // connect to AnaGate SPI
if (nRC == 0 )
{
    // Now program access via normal operation mode or command mode
    // see the following command sequences

    nRC = SPICloseDevice( &hHandle ); // close connection
}
```

Normal operating mode

```
// 500Kbit, 3,3V level, 3.3 voltage, CPHA=1 and CPOL=1
nRC = SPISetGlobals( hHandle, BAUDRATE, 2, 0, 3);

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

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

// now you can read data
nRC = SPIDataReq( hHandle, dummywritbuf, 2, readbuf, 2 );
```

Command-mode

```
// 500Kbit, 3,3V level, 3.3 voltage, CPHA=0 And 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 Hilfspannung, CPHA=1 und CPOL=1
nRC = SPISetGlobals( hHandle, BAUDRATE, 2, 0, 3);

// now we are in command mode and can start

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

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

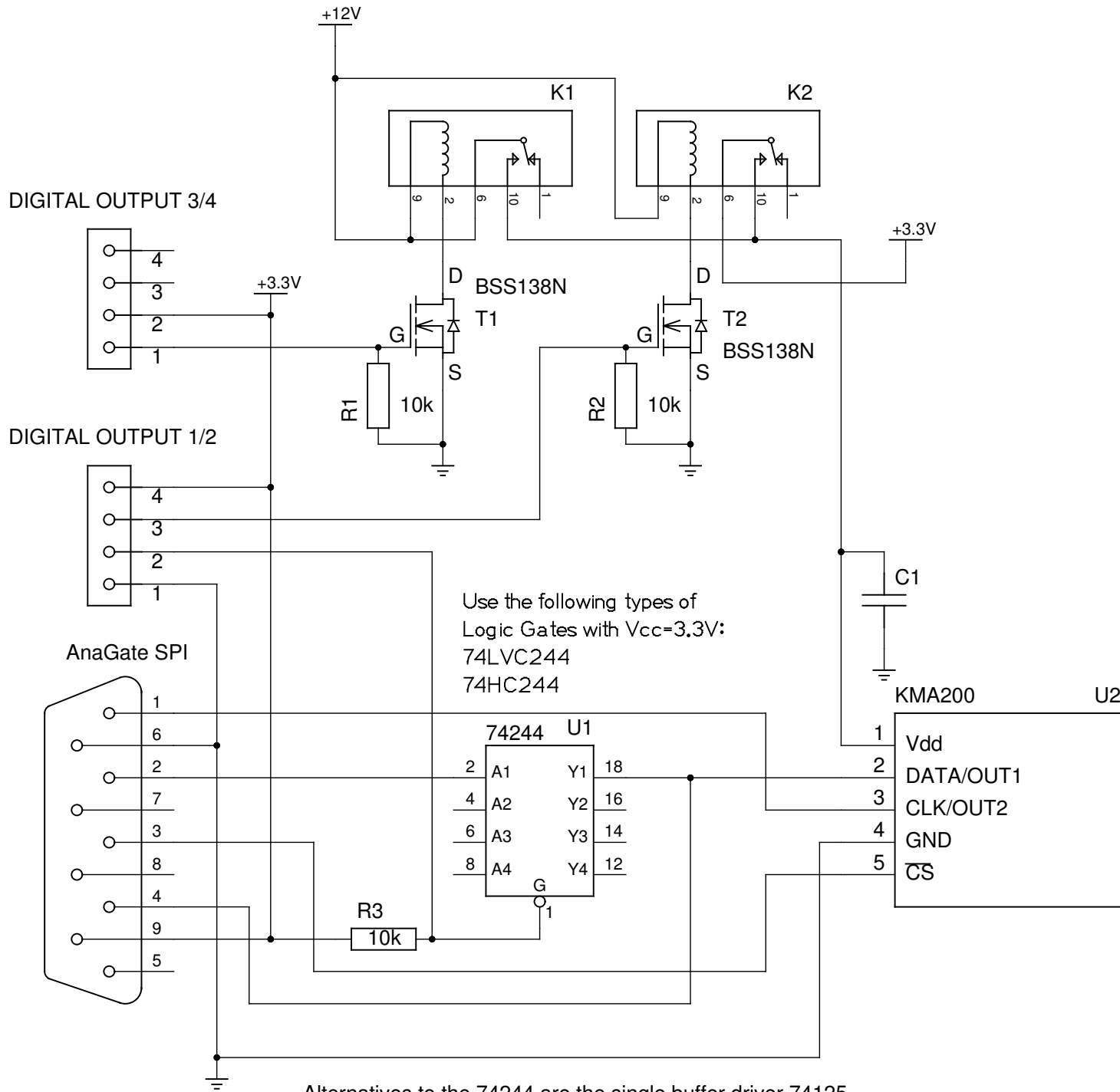
```
char readbuf[2];
nRC = SPIWriteDigital( hHandle, 0x03 ); // set digital 1+2
nRC = SPIDataReq( hHandle, &cmdbyte, 1, dummy, 0 );
nRC = SPIWriteDigital( hHandle, 0x02 ); // set digital 2
nRC = SPIDataReq( hHandle, &dummy, 2, readbuf, 2 );

// C: write programmed data into eeprom
//-----
char cmdbyte2 = 0x30; // set EEPROM
nRC = SPIWriteDigital( hHandle, 0x03 ); // set digital 1+2
nRC = SPIDataReq( hHandle, &cmdbyte2, 1, dummy, 0 );

nRC = SPIWriteDigital( hHandle, 0x04 ); // set digital 3

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