

```
/* ATB_WTMA_Prg_11.c Created: 01.03.2015 09:00:42 Author: AS */
```

```
#include "main.h" // Angabe der Adresse und Frequenz
#include "i2cmaster.h" // Aufruf twimaster
#include <util/delay.h>
uint8_t da;
uint8_t db;
uint8_t var=0;
uint8_t mask;
uint8_t ret;

// adr1_w 0x82 P22 WTMA mit PCA9536
// adr2_w 0x40 P11 4xT + 4xLED

#define adr1_w 0x82
#define adr2_w 0x40
#define adr2_r 0x41
#define output_Reg 0x01
#define input_Reg 0x02
#define config_Reg 0x03

int8_t abfrage_Bus(uint8_t slave_adresse) // Abfrage und Fehlermeldung
{
    ret = i2c_start(slave_adresse); // Abfrage Slave vorhanden
    i2c_stop();
    if (ret == 0)
    {
        PORTC|=(1<<PC5);
    }
    else // Fehlererkennung
    {
        PORTC &=~(1<<PC5);
    }
}

int8_t PCA9536_Write(uint8_t slave_adresse, uint8_t cfg_Reg, uint8_t data)
{
    i2c_start(slave_adresse);
    i2c_write(cfg_Reg);
    i2c_write(data);
    i2c_stop();
}

int8_t PCF8574_Read(uint8_t slave_adresse1, uint8_t slave_adresse2)
{
    i2c_start(slave_adresse1); // Schreibbefehl für Device 2
    i2c_write(0xff); // Alle Pins des PCF auf 0
    i2c_start(slave_adresse2); // Starte Lesezugriff
    da=i2c_readNak(); // Schreib Leseergebnis in d
    i2c_stop();
}

int8_t PCA9536_Read(uint8_t slave_adresse)
{
    i2c_start(slave_adresse); // Startbefehl für WTMA
    i2c_write(0x00); // setze Lesezugriff
}
```

```

i2c_start(slave_adresse|0x01);           // Starte Lesezugriff
db=i2c_readNak();                       // Schreib Leseergebnis in d
i2c_stop();
}

int main(void)
{
i2c_init();                             // Starte I2C Bus
DDRC=0x20;                              // Anzeige LED am Prozessor
abfrage_Bus(adr1_w);                    // Bus Adresse vorhanden
abfrage_Bus(adr2_w);                    // Bus Adresse vorhanden
PCA9536_Write(adr1_w, config_Reg, 0x08); // PCA9536 auf OUTPUT MODUS umstellen
while(1)
{
PCF8574_Read(adr2_w, adr2_r);           // Hauptschleife
PCA9536_Read(adr1_w);                   // Lesezugriff P11
if(~db & 0x08)                           // Lesezugriff Abfrage WTMA
{
i2c_start(adr2_w);                       // Auswertung Lesezugriff
i2c_write(0x70);                          // weckt Busteilnehmer auf
i2c_stop();                               // Index auf OUTPUT Data
}                                         // Stopt Bus
else
{
i2c_start(adr2_w);                       // weckt Busteilnehmer auf
i2c_write(0xf0);                          // Index auf OUTPUT Data
i2c_stop();                               // Stop Bus
}
var = 0;                                  // Abfrage Taste
for(mask=0x01; mask < 0x07; mask<<=1)
{
if (da & mask)
{
var |=mask;                               // Taste prüfen
}                                         // und kopieren
}
if (~da & 0x08)                           // Taste 4
{
i2c_start(adr2_w);                       // weckt Busteilnehmer auf
i2c_write(0xe0);                          // Index auf OUTPUT Data stellen
i2c_stop();                               // Stopt Bus
}
else
{
i2c_start(adr2_w);                       // weckt Busteilnehmer auf
i2c_write(0xf0);                          // Index auf OUTPUT Data stellen
i2c_stop();                               // Stopt Bus
}
PCA9536_Write(adr1_w, output_Reg, var); // Aufruf Unterprogramm
}
}

```