

## myTWI PortExpander Version 1.04

### Inhalt

Allgemeine Beschreibung.....	3
Eigenschaften.....	3
Prinzipschaltplan.....	3
Technische Daten.....	4
Betriebsdaten.....	4
Schnittstellendaten.....	4
Mechanische Daten.....	4
Pinbelegung.....	4
Schaltplan.....	5
Bestückungsplan.....	5
Bestücktes Board.....	5
Adresskonfiguration.....	6
Anwendungsscript mySmartUSB Terminal.....	7
Programmbeispiel LED-Ansteuerung.....	10
Versuchsaufbau mit Laborkarte.....	14
Allgemeine Sicherheitshinweise.....	14

### Contenu

Description globale.....	3
Caractéristiques.....	3
Schéma de principe.....	3
Caractéristiques techniques.....	4
Conditions de fonctionnement recommandées.....	4
Caractéristiques des ports.....	4
Caractéristiques mécaniques.....	4
Configurations du brochage.....	4
Schéma électrique.....	5
Schéma d'implantation.....	5
Carte assemblée.....	5
Configuration de l'adresse.....	6
Script d'application mySmartUSB Terminal.....	7
Programme d'exemple Contrôle LED.....	10
Démonstration avec carte de prototypage.....	14
Précautions d'utilisation.....	14

Die Informationen in diesem Produkt werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen.

Trotzdem können Fehler nicht vollständig ausgeschlossen werden.

Die Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind die Autoren dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien.

Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

Fast alle Hardware- und Softwarebezeichnungen, die in diesem Dokument erwähnt werden, sind gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden.

© Laser & Co. Solutions GmbH  
Promenadenring 8  
02708 Löbau  
Deutschland

[www.myAVR.de](http://www.myAVR.de)  
[support@myavr.de](mailto:support@myavr.de)

Tel: ++49 (0) 358 470 222  
Fax: ++49 (0) 358 470 233

Malgré le plus grand soin apporté à la rédaction de ce document, les auteurs ne sauraient être tenus responsable de l'exactitude des informations fournies. Les auteurs déclinent toute responsabilité en cas de dommages dus à une quelconque information erronée.

Tous droits réservés. Toute reproduction même partielle, et par tous les moyens que ce soient, électroniques ou mécaniques, incluant la photocopie et le microfilm, est formellement interdite sans la permission du publicitaire.

Toutes les marques et marques déposés utilisées dans ce documents sont et restent la propriété de leur détenteurs.

Devtronic SARL  
7 av de la Marcaissonne  
31400 Toulouse  
France

[www.myAVR.fr](http://www.myAVR.fr)  
[support@myavr.fr](mailto:support@myavr.fr)

## Allgemeine Beschreibung

Die Zusatzplatine (Add-On) „myTWI PortExpander“ ist ein Teil der TWI-Serie der myAVR-Produktfamilie. Damit wird das myAVR-System um eine komfortable Lösung für die Ansteuerung weiterer I/O Ports, beispielsweise für die Verwendung zusätzlicher Peripheriegeräte, erweitert.

Es kann mit weiteren TWI (I<sup>2</sup>C) Add-Ons am myAVR Erweiterungsport angeschlossen werden.

### Eigenschaften

- Universelle TWI-Porterweiterung mit 2 zusätzlichen I/O Ports für insgesamt 16 Pins
- Steckerleiste für den Anschluss an die myAVR Boards MK1, MK2, MK3 sowie mySmartControl
- Buchsenleiste für den Anschluss weiterer Module
- Robust, mit Dokumentationsdruck
- Industriefertigung
- Material: FR4, 1,5 mm; 0,35 µm Cu
- Gebohrt, verzinkt, Lötstopmmaske

## Description globale

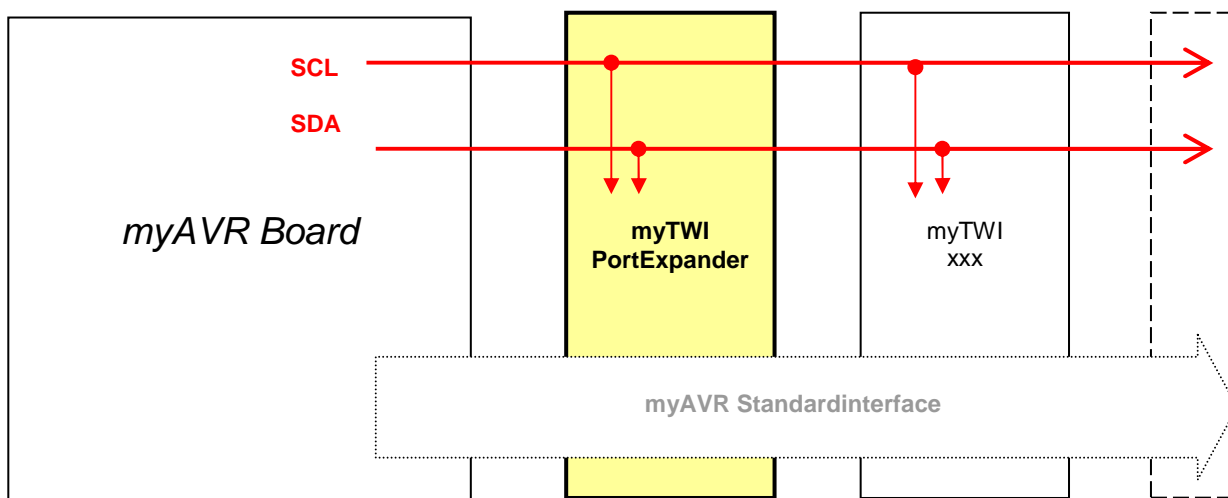
Le module additionnel myTWI Port Expander vient agrandir la série des modules TWI de la gamme myAVR. Ce module offre une solution pratique pour contrôler plus d'entrées/sorties et donc de connecter plus de périphériques ensemble.

Peut s'utiliser avec d'autres modules TWI (I<sup>2</sup>C) connectés sur le port d'extension des cartes myAVR.

### Caractéristiques

- Port d'extension TWI universel ajoutant 16 broches d'entrees/sorties, soit 2 fois plus.
- Connecteur pour relier le module aux cartes myAVR MK1, MK2, MK3 ou encore mySmartControl
- Connecteur permettant la liaison à d'autres modules
- Circuit imprimé robuste, avec texte
- Production Industriel
- Technologie PCB : FR4, 1,5; 0,35µm CU
- Pre-percé, pré-étamé, vernis de protection

## Prinzipschaltplan Schéma de principe



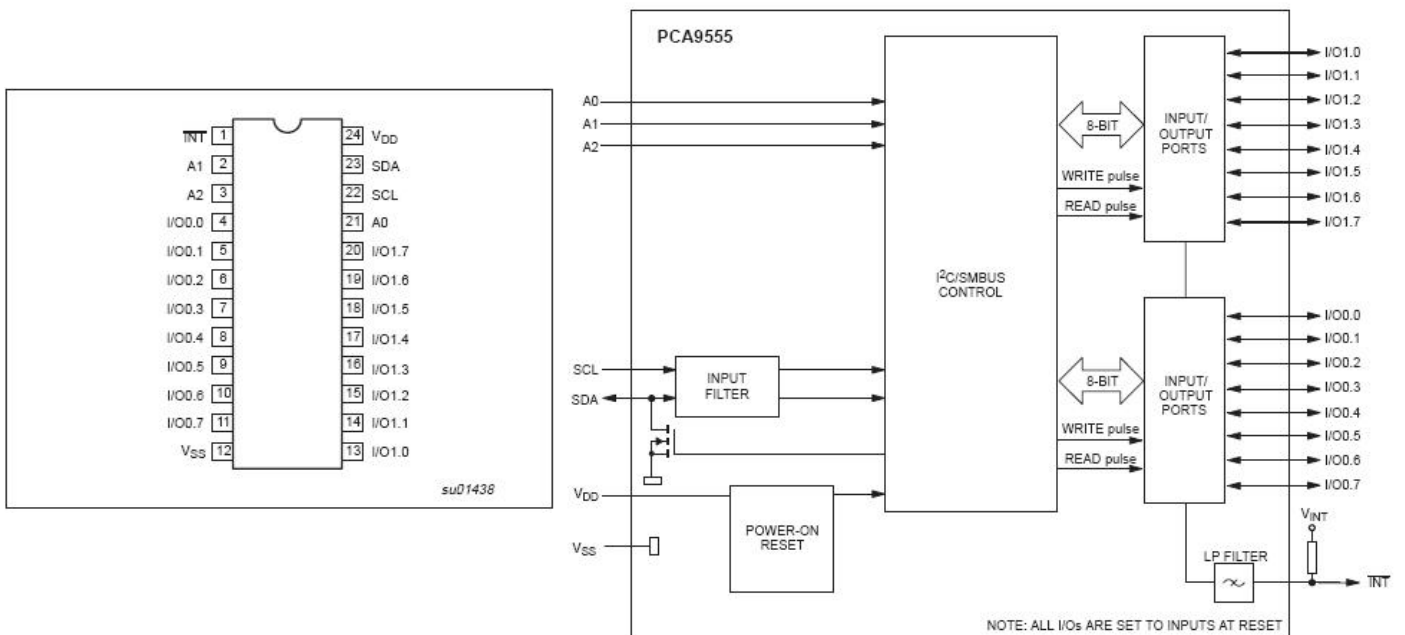
Technische Daten	
<b>Betriebsdaten</b>	
Versorgungsspannung	5V
Betriebsstrom	200mA
Betriebsspannung	2,3V - 5,5 V
Betriebstemperatur	-40 °C – 85 °C
Lagertemperatur	-65 °C – 150 °C
<b>Schnittstellendaten</b>	
Adresse	0b0100aaax a=Adressjumper x=read/write
Kommunikation	TWI (I <sup>2</sup> C)

Caractéristiques techniques	
<b>Conditions de fonctionnement recommandées</b>	
Tension d'alimentation	5V
Courant consommé	200mA
Tension de fonctionnement	2,3V – 5,5V
Température de fonctionnement	-40 °C – 85 °C
Température de fonctionnement	-65 °C – 150 °C
<b>Caractéristiques des ports</b>	
adressage	0b0100aaax a=address jumper x=read/write
Communication	TWI (I <sup>2</sup> C)

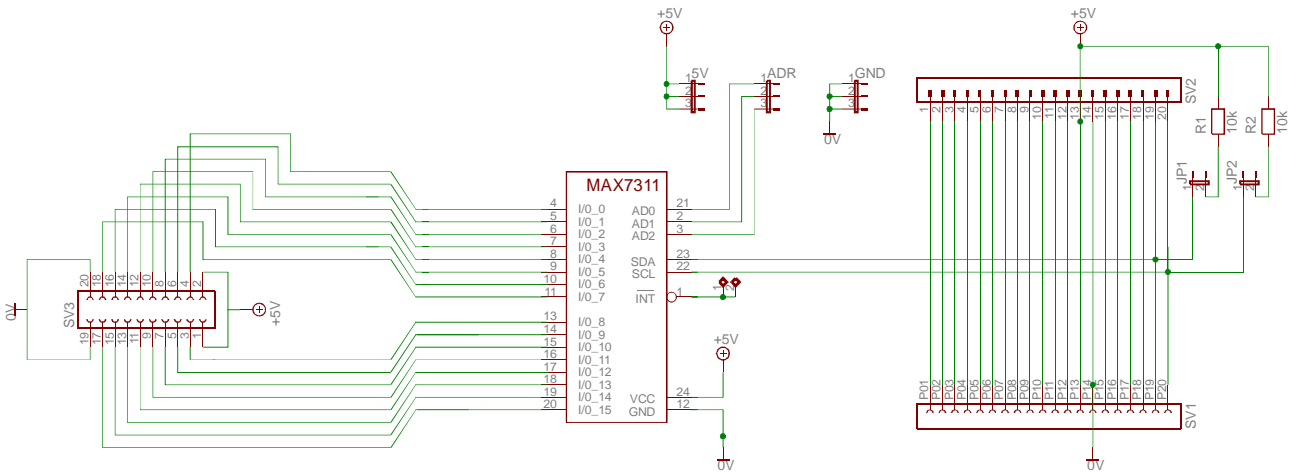
Mechanische Daten	
Abmessungen Platine (L x B x H) in mm	90 x 30 x 1,5
Masse in g	15
Rastermaß in mm	2,54
Leiterplattenmaterial:	FR4; 0,35 µm Cu

Caractéristiques mécaniques	
Dimension de la carte (L x l x h) en mm	90 x 30 x 1,5
Poids en g	15
Pas des connecteurs (mm)	2,54
Technologie PCB	FR4; 0,35µm Cu

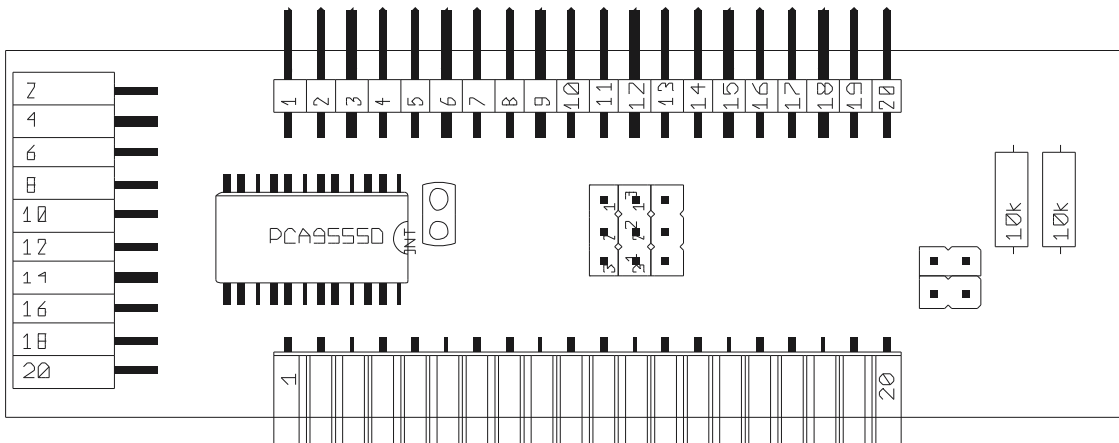
**Pinbelegung Configurations du brochage**



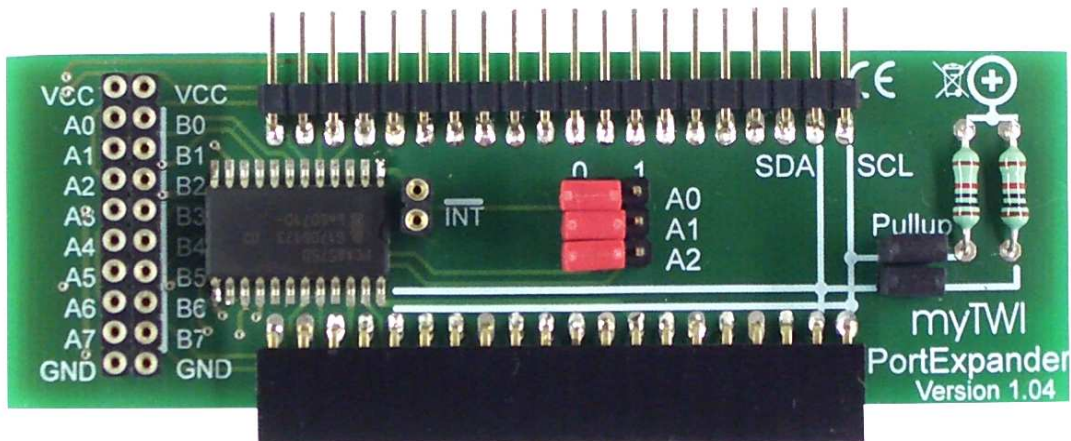
Schaltplan **Schéma électrique**



Bestückungsplan **Schéma d'implantation**



Bestücktes Board **Carte assemblée**

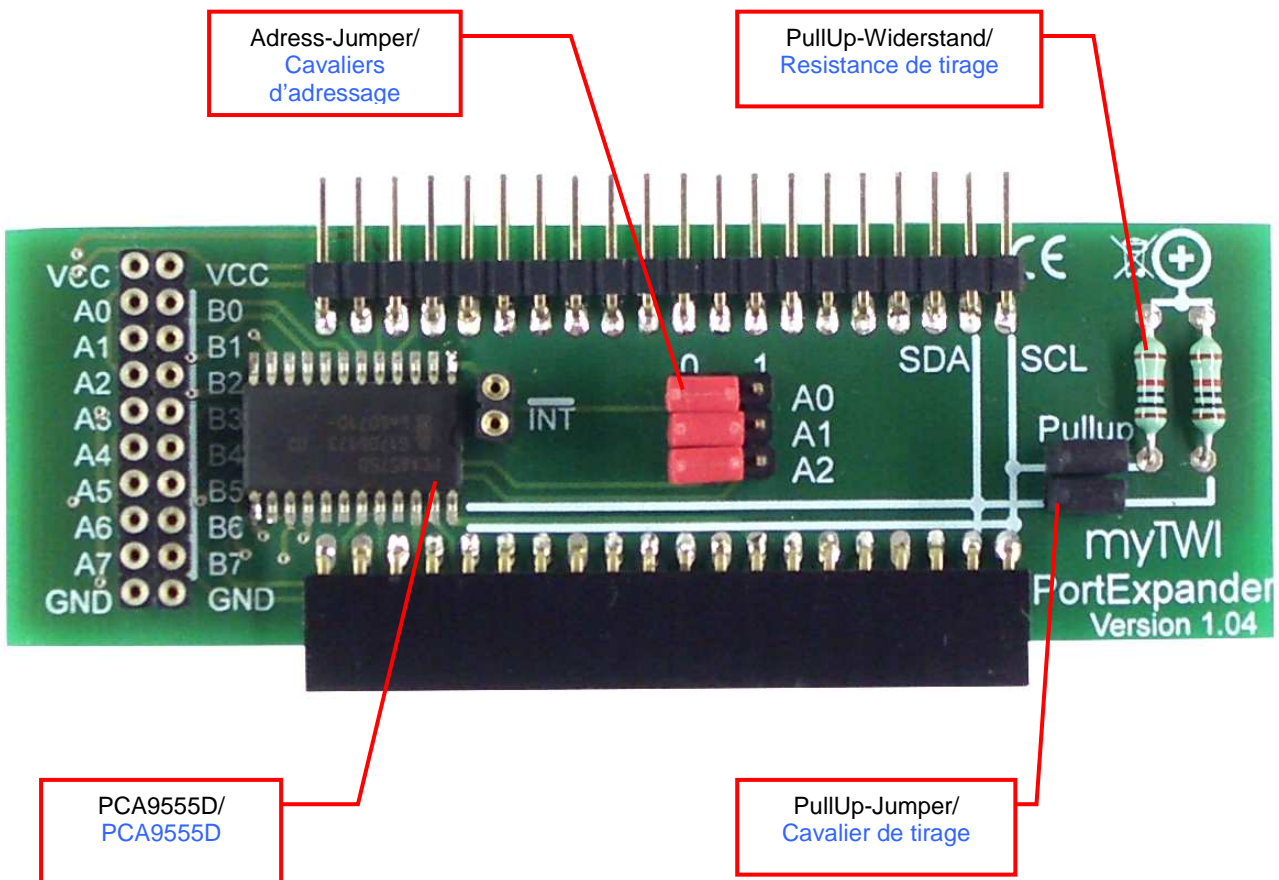


## Adresskonfiguration

Ein myAVR TWI Add-On kann mit weiteren TWI Add-Ons in einem BUS betrieben werden. Ein TWI Gerät bildet aus seiner Geräte-ID und den möglichen Adresspins (A0-A2) seine Geräteadresse im Bus. Somit lassen sich auch mehrere gleiche Geräte in einem BUS betreiben. Auf jedem myTWI Add-On sind die Adresspins per Jumper konfigurierbar. Des weiteren muss der TWI Bus mit PullUp-Widerständen auf High gezogen werden. Dies sollte jeweils nur von einem Add-On erfolgen. Dazu verfügt jedes Add-On über entsprechende PullUp-Widerstände und Jumper, um diese zu aktivieren.

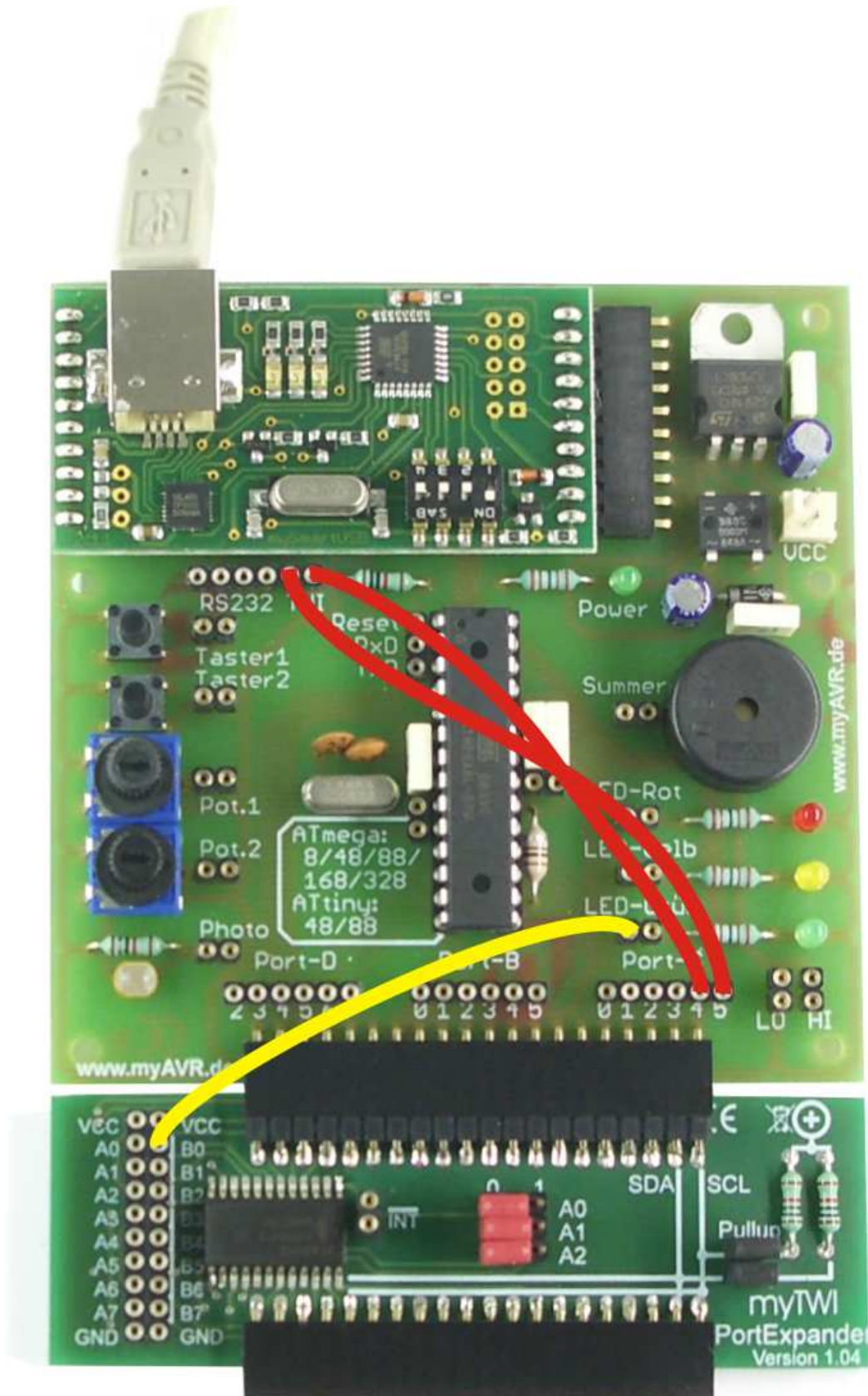
## Configuration de l'adresse

Plusieurs modules d'extensions myAVR TWI peuvent être connectés à un même bus. Chacun d'entre eux se voit attribuer une adresse, définie à partir de son identifiant de module et des valeurs affectées à ses broches d'adresse (A0 à A2). Ces dernières servent à différencier les modules identiques qui sont placés sur un même bus. Elles s'utilisent avec des cavaliers (fournis). Des résistances de pull-up doivent être ajoutées au bus TWI comme le recommande la norme. Cette opération peut être effectuée directement sur l'un des modules d'extension mais ne doit en aucun cas être effectuée en plusieurs points. Des résistances de pull-up activables par cavaliers (fournis) sont implantées sur chaque module d'extension pour faciliter l'exécution de cette opération.



Anwendungsscript mySmartUSB Terminal

Script d'application mySmartUSB Terminal

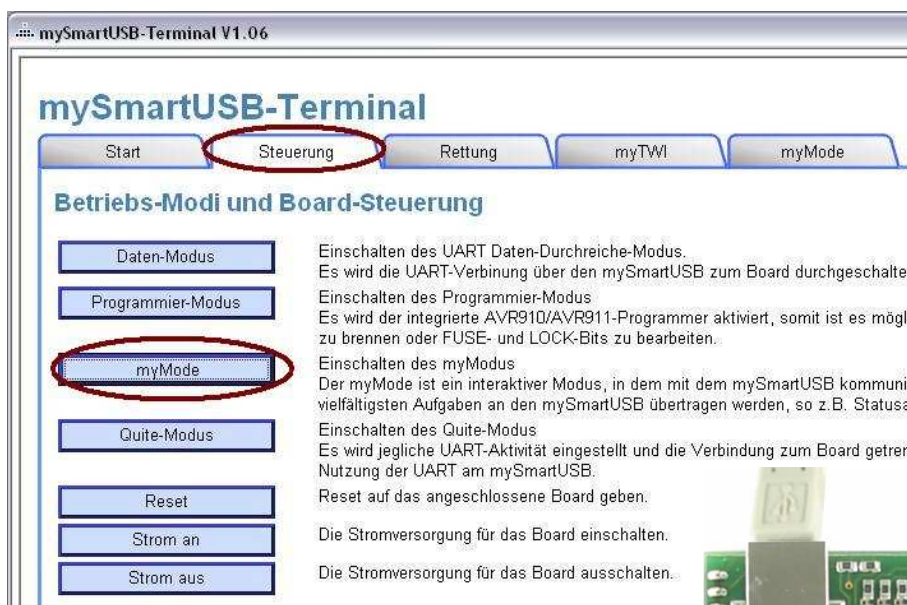


Mit Hilfe dieses Programmbeispiels lassen sich die einzelnen I/O Ports des myTWI PortExpanders als Ein- und Ausgänge schalten. Die korrekte Umsetzung wird durch das Leuchten einer zusätzlich angeschlossenen LED visualisiert. Zur Realisierung wird das Programm „mySmartUSB Terminal“ benötigt, dieses können Sie in der Downloadsektion von unserer Firmenwebseite herunterladen.

1. Stecken Sie das myTWI Add-On an das myAVR Board an. Achten Sie bitte darauf, das die Adresspins in der Grundstellung sind, das heißt A2, A1, A0 auf 0.
2. Verbinden Sie Port TWI-SCL mit Port C.5 sowie Port TWI-SDA mit Port C.4 mit jeweils einem Patchkabel (diese müssen sich überkreuzen). Weiterhin ist ein Anschluss einer LED vom myAVR Board mit Port B.0 des Portexpanders zur Visualisierung notwendig.
3. Verbinden Sie nun das Board mit dem PC.
4. Starten Sie das Programm „mySmartUSB-Terminal“ (Hinweis, die neuste Version wird empfohlen).
5. unter der Registerkarte „Start“ wählen Sie bitte die Schaltfläche „mySmartUSB suchen“ (bei erfolgreicher Suche erscheint rechts daneben die USB-Version in grüner Schrift).
6. Bei erfolgreicher Erkennung wechseln Sie bitte zur Registerkarte „Steuerung“ und wählen den „myMode“ aus. Dies erkennen Sie am Leuchten der mittleren grünen LED auf dem mySmartUSB MK2.

Avec cet exemple, vous pouvez configurer tous les ports d'entrées/sorties du myTWI PortExpander et choisir la direction. Vous pouvez vérifier l'attribution correcte de la direction grace à une LED de debug. Pour réaliser ce programme de test, vous devez utiliser le logiciel "mySmart USB Terminal" que vous pouvez télécharger depuis notre site web.

1. Connecter le module TWI à la carte myAVR Vérifier que les broches d'adresse sont dans leurs positions initial. (A2; A1; A0 set 0)
2. relier les ports TWI SCL au port C.5 et TWI SDA au port C4 avec un pair torsadée. Relier également une led de la carte myAVR au port B0 du myTWI PortExpander pour visualisation.
3. Connecter la carte myAVR à votre ordinateur
4. Démarrer « mySmart USB Terminal » (nous recommandons d'utiliser la dernière version)
5. Dans l'onglet démarrer (Start), choisissez le bouton « mySmartUSB suchen ». Si la recherché a aboutie, un encadré vert apparaît avec la version.
6. Apres cette recherche fructueuse, passer à l'onglet "Steuerung" pour choisir "myMode" Sur le programmeur mySmartUSB MK2, la led centrale doit être allumée.



7. Nun wechseln Sie zur Registerkarte „myMode“ und übertragen den auf der nächsten Seite aufgeführten Quelltext in das Ausführenfenster hinein. Gehen Sie danach auf „Senden“
8. Bei erfolgreicher Ausführung leuchtet die angeschlossene LED und ist somit auf Ausgang geschaltet.

7. Maintenant, vous pouvez ouvrir l'onglet "mymode" et copier le source code suivant dans la fenêtre d'exécution. Ensuite cliquer sur le bouton « Senden ».
8. Une fois l'exécution réussit, la LED connectée en sortie de myTWI PortExpander (port B0) est allumée.



Dies ergibt folgende Scriptsequenz in der myMode Konsole:

[La suite illustre la séquence dans la console myMode.](#)

Den Controller auf dem myAVR Board deaktivieren durch Schaltung der Resetleitung nach Restart  
In das TWI (I<sup>2</sup>C) Menü wechseln  
mySmartUSB als Master initialisieren  
TWI START  
TWI WRITE Adresse des PCA9555D  
TWI WRITE Kommandoregister 6=CONFIG  
TWI WRITE I/O-Konfiguration Port A+B  
TWI START  
TWI WRITE Adresse des PCA9555D  
TWI WRITE Kommandoregister 2=OUTPUT  
TWI WRITE Wertzuweisung Port A+B  
TWI STOP  
END

```
m:main
mode mh
pwr 0
rst 1
pwr 1
m:twi
ima
sta
sla 0x40 w
wr 6
wr 0x00 0x00
sta
sla 0x40 w
wr 2
wr 0x00 0x01
sto
end
```

Desactive le microcontrôleur de myAVR Board et effectue un reset

passé en mode TWI (I<sup>2</sup>C).

Initialise le mySmartUSB en master

TWI start

TWI WRITE adresse de PCA9555D

TWI WRITE command register 6=CONFIG

TWI WRITE I/O configuration port A+B

TWI START

TWI WRITE adresse du PCA9555D

TWI WRITE command register 2=OUTPUT

TWI WRITE value assignment port A+B

TWI STOP

END

## Programmbeispiel LED-Ansteuerung

Zielstellung dieses Anwendungsprogramms ist eine einfache Tasterabfrage, sowie eine daraus resultierende LED-Ansteuerung mit dem myAVR Workpad PLUS zu realisieren.

Notwendige Einstellungen:

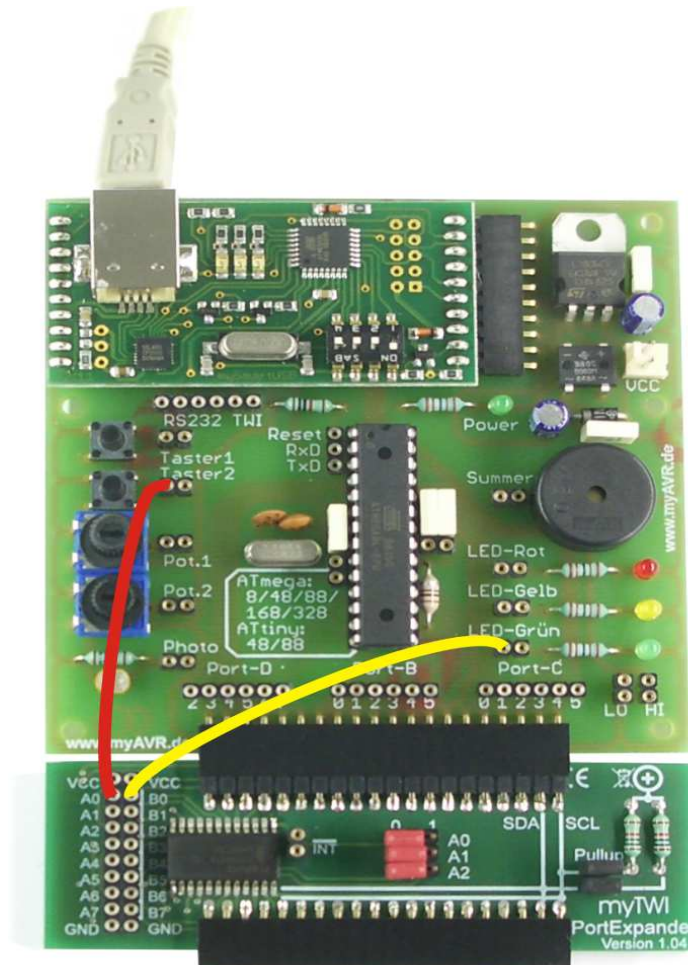
1. Stecken Sie das myTWI Add-On an das myAVR Board MK2 USB an. Achten Sie bitte darauf, dass die Adresspins in der Grundstellung sind (A2, A1, A0 auf 0).
2. Verbinden Sie einen Taster des myAVR Boards mit dem Port A.0 des PortExpanders und eine LED mit dem Port B.0 mithilfe eines Patchkabels.
3. Starten Sie das myAVR Workpad.
4. Klicken Sie im Begrüßungs Menü auf „neue Datei“ und wählen bitte „Grundgerüst“ unter der Programmiersprache C/C++ aus.
5. Führen Sie nun die Hardwareerkennung unter der Registerkarte „Einstellung“ durch und speichern Sie diese.
6. Wechseln Sie nun zurück in die Entwicklungsumgebung und wählen unter „Vorlagen“: 9.TWI > *twilnitMaster* aus.

## Programme d'exemple Contrôle LED

Le but de ce programme est un simple test permettant de contrôler une LED et utilise l'environnement myAVR WorkPad PLUS.

Configuration nécessaire:

1. Connecter le module TWI à la carte myAVR MK2 Vérifier que les cavaliers d'adressage sont dans leurs positions initiales.
2. Connecter un bouton poussoir de la carte myAVR au port A0 du PortExpander ainsi qu'une LED au port B0.
3. Démarrer myAVR WorkPad
4. Cliquer sur le menu "neue Datei" et sélectionner "Grundgerüst » pour le langage C/C++
5. Lancer la détection du matériel dans le menu "Einstellungen" et sauver cette configuration.
6. Retourner sous votre environnement de développement et sélectionner "Vorlagen" : 9 TWI *twilnitMaster*



Das Beispielscript umfasst folgende Komponente:

- Die Initialisierung des TWI-Master
- Auslösen der Start-Sequenz
- Die Adressierung des PCA955D im TWI-Bus sowie die Konfiguration der beiden I/O Ports (Port A Eingang, Port B Ausgang)
- Realisierung der eigentlichen Mainloop mit Abfrage der aktuellen Pinbelegung am Eingang und schalten der angeschlossenen LED bei Tasterbetätigung.

Um dies realisieren zu können, sind weitere Definitionen im Programmkopf vorzunehmen, so muss das TWCR-Register sowie die Busgeschwindigkeit wie im unten aufgeführten Beispiel deklariert werden.

Weiterhin ist für den korrekten Ablauf die Initialisierung von eigenen Unterfunktionen notwendig. Konkret wird folgendes zusätzlich zu den Funktionen aus der Bibliothek benötigt:

- `void twiInitPCA()`  
zur Initialisierung des PCA
- `void outPorts(uint8_t data, uint8_t ackn)`  
zur Übermittlung der gewünschten Pinbelegung (=data).
- `uint8_t readPorts (uint8_t portNr)`  
zum Auslesen des gewünschten Ports (=PortNr) und Rückgabe der aktuellen Pinbelegung (data)

Am Ende sollte Ihr Programmkopf in dieser Form aussehen:

```
//-----
// Titel      : Beispielpogramm Portexpander
//-----
// Funktion   : Ansteuerung einer LED via Tastereingabe
// Schaltung  : ...
//-----
// Prozessor  : ATmega8
// Takt       : 3.6864 MHz
// Sprache    : C
// Datum     : 24.09.09
// Version    : 1.0
// Autor      : Christian Müller
//-----
#define F_CPU 3686400
#include <avr\io.h>
#define BAUD 9600
//-----
// TWI-Funktionssammlung
#ifndef TWI_CLOCK
#define TWI_CLOCK 100000 // Geschwindigkeit des TWI-Busses
#endif
// TWCR - Control-Register-Bits
#define _TWINT 0b10000000
#define _TWEA 0b01000000
#define _TWSTA 0b00100000
#define _TWSTO 0b00010000
#define _TWWC 0b00001000
#define _TWWEN 0b00000100
#define _TWIE 0b00000001

void twiInitMaster(uint8_t twiAdr);
void twiStart();
void twiStop();
void twiWriteByte(uint8_t data, uint8_t ackn);
uint8_t twiReadByte(uint8_t ackn);
void twiInitPCA();
void outPorts(uint8_t portNr, uint8_t data);
uint8_t readPorts(uint8_t portNr);
```

Cet exemple comprend les fonctions suivantes

- Initialiser le TWI-maître
- Exécute la sequence de démarrage
- Adresse le PCA955D via un Bus TWI et configure ces ports en entrées/sorties (port a input, port B output)
- Exécute une boucle principale pour échantillonner les entrées et piloter la LED connecter à l'aide de l'entrée bouton poussoir.

Pour réaliser ces fonctionnalités, quelques définitions préalables sont requises. Ainsi les registres TWCR et la vitesse de communication doivent être définis comme dans l'exemple ci-dessous.

De plus, les fonctions définies doivent d'abord être déclarées.

- `void twiInitPCA()` to initialisation the PCA
- `void outPorts(uint8_t data, uint8_t ackn)`  
to transfer of the required pin assignment (=data)
- `uint8_t readPorts (uint8_t portNr)`  
to read-out the ports (portNr) and return the state of the pin assignment

Finalement, le début du programme doit ressembler à cela:

Folgend die eigentliche Mainloop mit den eingefügten Bibliotheksfunktionen:

Es wird der TWI-Master mit Geräte-ID 0x40 initialisiert und die I/O Belegung des PCA9555D konfiguriert. Innerhalb der Endlosschleife wird die Pinbelegung am Port 0 abgefragt und mit dem Wert 0xFE (Bit 1-7 auf logisch 1, Bit 0 auf logisch 0 = Taster gedrückt) verglichen. Sollte der Taster geschlossen sein, werden am Port 1 alle Ausgänge auf High geschaltet (LED leuchtet), sonst bleiben sie auf Low.

Maintenant, regardons l'écriture de la boucle principale :

Sont déclarés un maître TWI avec l'adresse 0x40 et la configuration des entrées/sorties du PCS 9555D. Dans cette boucle infinie, l'état du port 0 est lu et comparé avec 0xFE (masque logique pour tester le bit 0). Si le bouton poussoir est activé, toutes les sorties passent à l'état 1 et la LED s'allume, sinon les sorties restent à l'état 0.

```
//-----
// main
//-----
int main ()
{
    uint8_t data;
    twiInitMaster(0x40);
    twiInitPCA();
    do
    {
        data=readPorts(0);    // Abfrage Pins an PortA
        if(data == 0xFE)      // Taster gedrückt
        {
            outPorts(1,0xFF); // alle Pins an PortB geschalten
        }
        else
        {
            outPorts(1,0x00); // kein Pin an PortB geschalten
        }
        waitMs(100);         // Warte 100ms
    }
    while (true);
}

//-----
// twiInitMaster
//-----
void twiInitMaster(uint8_t twiAdr)
{
    // Clock
    TWBR=((F_CPU/TWI_CLOCK)-16)*2;
    // TWI-Status-Register (Vorteiler)
    TWSR=0;
    // Bus-Addr
    // TWAR=twiAdr;
    // Enable
    TWCR=_TWINT|_TWEN;
}

//-----
// Start TWI (ohne Interrupt)
//-----
void twiStart()
{
    uint8_t x = TWCR;
    x &= _TWEN|_TWIE;    // nur Beibehalten von Enable und InterruptJ/N
    TWCR = x|_TWINT|_TWSTA;
    // warten bis fertig
    while( !(TWCR & _TWINT))
    {}
}

//-----
// Stopp TWI (ohne Interrupt)
//-----
void twiStop()
{
    uint8_t x=TWCR;
    x &= _TWEN|_TWIE;    // nur Beibehalten von Enable und InterruptJ/N
    TWCR = x|_TWINT|_TWSTO;
}

//-----
// Write Byte per TWI (ohne Interrupt)
// PE: data = zu sendende Daten
// ackn = wenn !=0 wird Acknowledge (=TWEA) gesetzt
//-----
```

```

void twiWriteByte(uint8_t data, uint8_t ackn)
{
    TWDR=data;        // Daten bereitlegen
    // Befehl zusammenstellen
    uint8_t x=TWCR;
    x&=_TWEN|_TWIE;   // nur Beibehalten von Enable und InterruptJ/N
    x|=_TWINT;
    if(ackn)
        x|=_TWEA;    // evt. TWEA setzen, für Datenanforderung
    TWCR=x;           // senden
    // warten bis fertig
    while( !(TWCR & _TWINT))
    {}
}

//-----
// Read Byte per TWI (ohne Interrupt)
// PE: ackn = wenn !=0 wird Acknowledge (=TWEA) gesetzt
// PA: Data
//-----
uint8_t twiReadByte(uint8_t ackn)
{
    // Befehl zusammenstellen
    uint8_t x=TWCR;
    x&=_TWEN|_TWIE;   // nur Beibehalten von Enable und InterruptJ/N
    x|=_TWINT;
    if(ackn)
        x|=_TWEA;    // evt. TWEA setzen, für Datenanforderung
    TWCR=x;           // senden
    // warten bis fertig
    while( !(TWCR & _TWINT))
    {}
    return TWDR;
}

//-----
// Funktionen für Portexpander
//-----
// twiInitPCA
//-----
void twiInitPCA()
{
    // I/O Definition (A Eingang, B Ausgang)
    twiStart();
    twiWriteByte(0x40,0); // TWI-Adresse
    twiWriteByte(0x06,0); // Adressierung des Kommandoregister = Config 0
    twiWriteByte(0xFF,0); // ConfigDaten A = Eingang
    twiWriteByte(0x00,0); // ConfigDaten B = Ausgang
    twiStop();
}

//-----
// readPorts
//-----
uint8_t readPorts(uint8_t port)
{
    twiStart();
    twiWriteByte(0x40,1); // TWI-Adresse
    twiWriteByte(port,1); // Kommandoregister 0/1 -> Input P0/P1
    twiStart();
    twiWriteByte(0x41,1); // TWI-Adresse und Read
    return twiReadByte(0);
}

//-----
// outPorts
//-----
void outPorts(uint8_t port, uint8_t data)
{
    twiStart();
    twiWriteByte(0x40,1); // TWI-Adresse
    twiWriteByte(0x02+port,1); // Kommandoregister 2/3 -> Output P0/P1
    twiWriteByte(data,0); //
}

//-----

```

**Beachte:**

Die konkreten Porteinstellungen sind von der Rechnerkonfiguration abhängig. Besonders der USB Programmierer mySmartUSB kann auf unterschiedlichen virtuellen COM Ports angemeldet werden. Es ist zu empfehlen, die COM Einstellung des mySmartUSB auf COM3 oder COM4 zu legen, da manche Werkzeuge wie das AVR Studio maximal COM4 zulässt. Die Zuweisung des COM Port erfolgt über den Gerätemanager.

**Notice:Avertissement :**

La définition des ports dépend de la configuration de votre ordinateur. Surtout le programmeur USB muSmartUSB qui peut être attribué à différent port virtuel. Nous recommandons d'utiliser mySmartUSB en port COM 3 ou 4, car certains outils (comme AVR Studio) ne supportent pas plus de 4 ports. Vous pouvez choisir le numéro du port COM dans l'HyperTerminal ou Minicom.

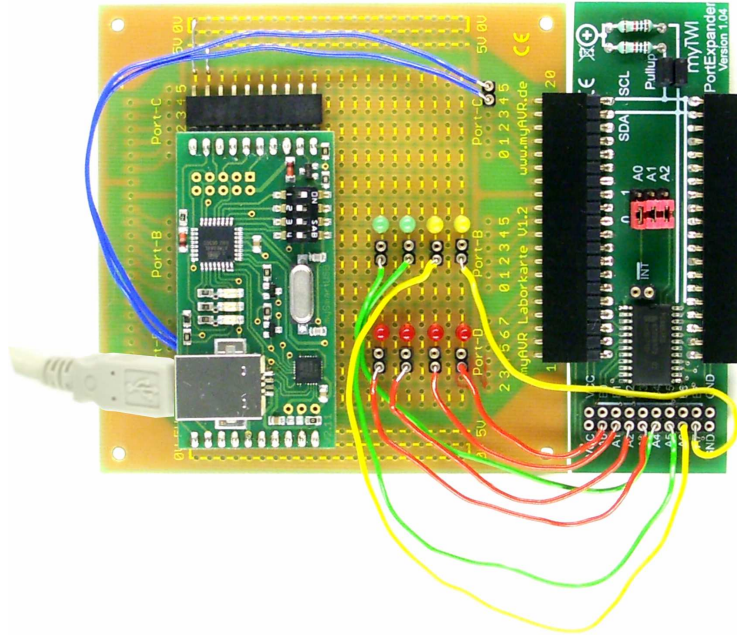
**Versuchsaufbau mit Laborkarte****Démonstration avec carte de prototypage**

Abbildung: Beispielprogrammrealisierung verschiedener LED-Ansteuerungen durch mySmartUSB MK2 mit Laborkarte und myTWI PortExpander

Illustration: Démonstration d'un control de LEDs à l'aide de mySmartUSB MK2, d'une carte de prototypage myAVR et du module myTWI PortExpander.

**Allgemeine Sicherheitshinweise**

Grundsätzlich ist der myTWI PortExpander nur zum Einsatz unter Lern- und Laborbedingungen konzipiert. Er ist nicht vorgesehen und nicht dimensioniert zur Steuerung realer Anlagen. Bei vorschriftsmäßigem Anschluss und Betrieb treten keine lebensgefährlichen Spannungen auf. Beachten Sie trotzdem die Vorschriften, die beim Betrieb elektrischer Geräte und Anlagen Gültigkeit haben. Wir versichern, dass die Leiterplatte durch den Hersteller getestet wurde. Für fehlerhaften und/oder vorschriftswidrigen Einsatz des Boards übernehmen wir keine Garantie.

**Précautions d'utilisation**

Le module myTWI PortExpander est conçu pour une utilisation scolaire et expérimentale uniquement. Il a été dimensionné en ce sens et ne doit donc en aucun cas être utilisé pour le contrôle de systèmes industriels. Aucune tension dangereuse n'est à craindre en cas d'utilisation appropriée. Veuillez néanmoins à respecter les règles élémentaires de sécurité relatives à la manipulation d'équipements électroniques basse tension. Nous assurons que le PCB a été testé par le fabricant. Nous ne pourrions être tenus responsables en cas d'utilisation inappropriée et/ou contraire aux règles de sécurité.

Die aktuellsten Dokumente zum myTWI PortExpander finden Sie unter [www.myAVR.de](http://www.myAVR.de) im Downloadbereich.

Vous trouverez les dernières informations relatives à myTWI PortExpander sur notre site web [www.myAVR.fr](http://www.myAVR.fr), onglet « Téléchargements »

Abbildungen können vom Inhalt abweichen. Änderungen im Sinne des technischen Fortschrittes behält sich der Hersteller vor. Photos non contractuelles. Le fabricant se réserve le droit d'amélioration technique