

# EP30-PPIO64

## Universelles I/O-Erweiterungsmodul

Mit dem Erweiterungsmodul EP30-PPIO64 wird ein System mit einem Multitaskingcomputer der BASIC-Tiger™ Familie um 64 Ein- und Ausgänge erweitert. Das Modul bietet sechs 8-Bit I/O Ports, dessen Bits einzeln als Eingang oder Ausgang genutzt werden können und einen 16 Bit Port mit synchronem Lese- und Schreibzugriff.

Die I/O Erweiterungsmodule werden über einen 8-Bit Daten- und Adressbus angeschlossen und mit einer individuellen Adresse belegt. Auf diese Weise lassen sich bis zu 16 Module in ein System integrieren. Softwareseitig ist das Ansprechen von externen I/O-Kanälen durch das XPORT-System genauso einfach wie im Falle interner I/O-Kanäle.

Das Modul wird mit einer 5V Spannungsversorgung betrieben, seine Ein- und Ausgänge arbeiten als 5V tolerante LVTTTL Signale(3,3V).

Der maximal entnehmbare Strom bei „Low“ und bei „High“ beträgt mehr als 20 mA, solange ein Gesamtstrom von 200mA nicht überschritten wird.



### Eigenschaften:

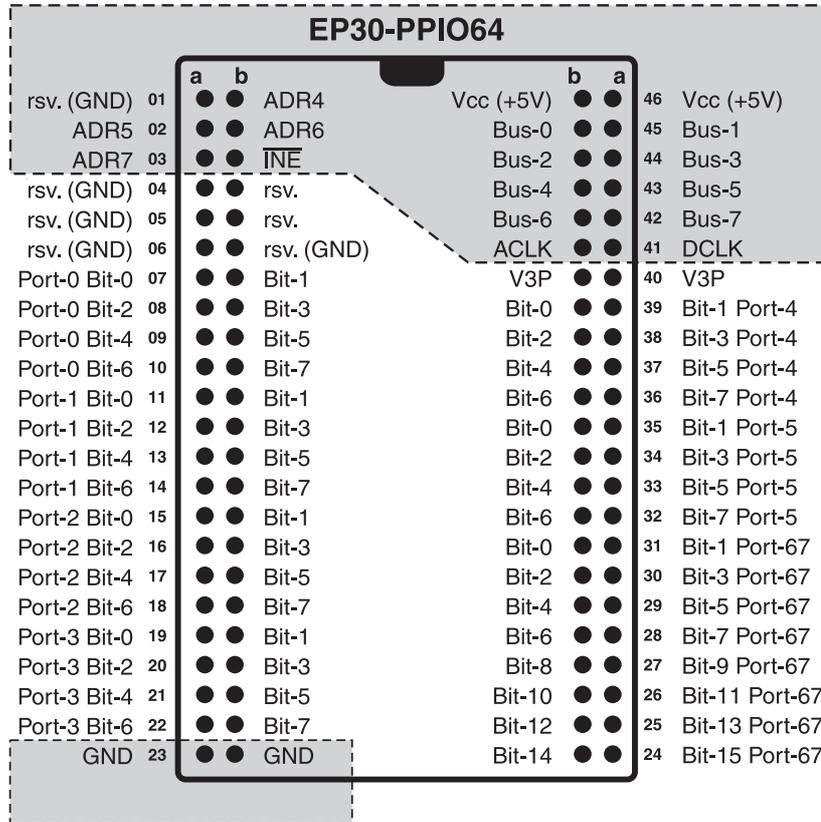
- ◆ sechs 8 Bit IO-Ports, bitweise als Ein- oder Ausgang konfigurierbar.
- ◆ ein 16 Bit IO-Port, als Ein- oder Ausgang konfigurierbar.
- ◆ einstellbare Adressen
- ◆ einfache Programmierung durch Tiger BASIC XPORT-System
- ◆ 5V DC Spannungsversorgung
- ◆ LVTTTL kompatible Ein- und Ausgänge

### Inhalt

◆ Pin-Beschreibung	Seite 2
◆ Anschluß an BASIC-Tiger™	Seite 3
◆ Einstellen der Basisadresse	Seite 4
◆ Programmierung der IOs	Seite 5
◆ Spezifikationen	Seite 7
◆ Modulgröße	Seite 8

# EP30-PPIO64

## Universelles I/O-Erweiterungsmodul



### Signalbeschreibung:

Bezeichnung	Funktion <sup>1</sup>	Bemerkung
GND	PWR	Ground
Vcc	PWR	Versorgungsspannung (+5V)
V3P	PWR	Ausgang des internen Spannungsreglers (+3,3V)
Bus-0...Bus7	I/O	Daten / Adressbus
DCLK	I	Data-Clock (steigende Flanke)
ACLK	I	Adress-Clock (steigende Flanke)
INE	I	Input-Enable (statisch, low aktiv)
ADR <sub>x</sub>	I	einstellbare Basisadresse des Moduls
Port-x Bit-y	I/O	Bit y des PIO-Ports x
Rsv.	I	reserviert, reservierte Pins müssen auf GND gelegt werden!

<sup>1</sup> PWR - Power; I - Input; O - Output; I/O - Input oder Output bzw. bidirektionaler Pin

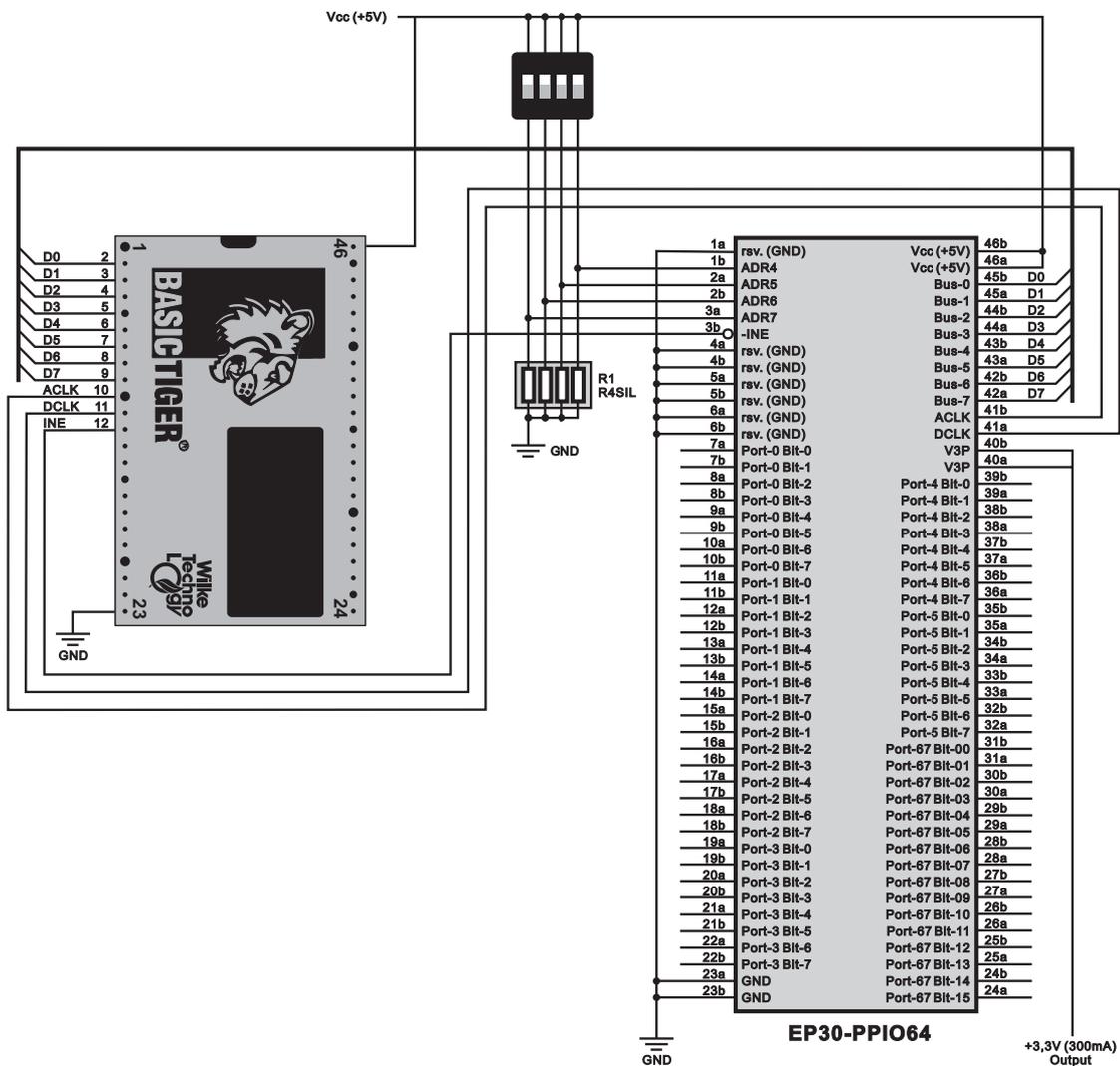
# EP30-PPIO64

## Universelles I/O-Erweiterungsmodul

### Anschluss an BASIC-Tiger™:

Das folgende Schaltbild zeigt die Zusammenschaltung des EP30-PPIO Moduls mit dem BASIC-Tiger™. Die Daten und Steuerleitungen können auch an anderen Pins des BASIC-Tigers angeschlossen werden. Beachten sie hierzu die Dokumentation des XSETUP-Befehls. (siehe Online Dokumentation des Tiger BASICs)

Mit den DIP-Schaltern lässt sich die Basisadresse des Moduls setzen.



# EP30-PPIO64



## Universelles I/O-Erweiterungsmodul

### Einstellen der Basisadresse:

Das EP30-PIO64-Modul belegt 15 aufeinander folgende Adressen des XPORT Systems. Die niedrigste der belegten Adressen wird Basisadresse genannt. Man kann die Basisadresse auf durch 16 teilbare Werte festlegen, indem man die Leitungen ADR7...ADR4 auf low oder high Pegel legt:

Basisadresse	ADR7	ADR6	ADR5	ADR4
00h	low	low	low	low
10h	low	low	low	high
20h	low	low	high	low
30h	low	low	high	high
40h	low	high	low	low
50h	low	high	low	high
60h	low	high	high	low
70h	low	high	high	high
80h	high	low	low	low
90h	high	low	low	high
A0h	high	low	high	low
B0h	high	low	high	high
C0h	high	high	low	low
D0h	high	high	low	high
E0h	high	high	high	low
F0h	high	high	high	high

### Programmierung:

Der PIO-Erweiterungsbaustein wird mit dem XPort I/O-Erweiterungssystem angesprochen und programmiert. Beachten Sie die Dokumentation der BASIC-Funktionen XSETUP, XIN und XOUT der Tiger-BASIC Online Hilfe.

#### 8 Bit Ports: (Port0...Port5)

Jedem Port ist ein Daten- und ein Steuerregister zugeordnet. Die Adresse eines Portregisters errechnet sich aus Port-Nummer + Basisadresse. Die Adresse des zugehörigen Steuerregisters ist gleich der Adresse des Portregisters plus 8, sie errechnet sich also aus Basisadresse + Portnummer + 8.

	<b>Port-Adr.</b>	<b>Steuer-Adr.</b>
Port 0	Basisadresse + 00h	Basisadresse + 08h
Port 1	Basisadresse + 01h	Basisadresse + 09h
Port 2	Basisadresse + 02h	Basisadresse + 0Ah
Port 3	Basisadresse + 03h	Basisadresse + 0Bh
Port 4	Basisadresse + 04h	Basisadresse + 0Ch
Port 5	Basisadresse + 05h	Basisadresse + 0Dh

#### Umschalten zwischen Ein- und Ausgang eines 8 Bit Ports

Jedes Bit eines 8 Bit Ports kann individuell als Ein- oder Ausgang programmiert werden. Nach dem Einschalten der Versorgungsspannung sind alle Bits als Eingang konfiguriert.

Um ein Bit eines 8 Bit Ports auf Ausgang zu schalten, muss das Bit des Steuerregisters auf 0 gesetzt werden. Durch setzen auf 1 wird das Portbit wieder auf Eingang geschaltet und der Pin wird hochohmig.

Die Anweisung

XOUT (Basisadresse + Portnummer + 8, 1111000b)

schaltet die Bits 0, 1 und 2 des durch „Portnummer“ angegebenen Ports auf Ausgang und die anderen Bits des Ports auf Eingang.

Beachten Sie, dass die Steuerregister nur beschrieben werden können. Lesen mit XIN liefert undefinierte Werte.

#### Lesen eines 8 Bit Ports

Durch einen Aufruf von XIN kann der Zustand der Portbits gelesen werden:

Wert = XIN (Basisadresse + Portnummer)

Die Bits der Variable Wert erhalten eine 1 für jedes Portbit, an dem High-Pegel anliegt, oder 0 bei Low-Pegel.

#### Setzen eines 8 Bit Ports

Durch eine XOUT Anweisung können die auf Ausgang geschalteten Bits auf High- oder Low-Pegel gelegt werden. Ein gesetztes Bit setzt das Portbit auf High-Pegel, eine 0 auf Low-Pegel. Auf Eingang geschaltete Pins bleiben hochohmig.

Beispiel:

XOUT (Basisadresse + Portnummer + 8, 11000000b)

XOUT (Basisadresse + Portnummer, 11110000b)

Die erste Anweisung setzt die Bits 0 bis 5 des durch „Portnummer“ angegebenen Ports auf Ausgang, die Bits 6 und 7 auf Eingang. Die zweite Anweisung setzt die Bits 0 bis 3 auf Low Pegel, die Bits 4 und 5 auf High Pegel. Die Bits 6 und 7 bleiben hochohmig, da sie auf Eingang geschaltet sind.

### 16 Bit Port: (Port67)

Dem 16 Bit Port sind zwei Daten- und ein Steuerregister zugeordnet. Die Adressen der Portregister sind 6 + Basisadresse und 7 + Basisadresse. Die Adresse des zugehörigen Steuerregisters ist Basisadresse 0EHex.

	<b>Port-Adr.</b>	<b>Steuer-Adr.</b>
Port 6	Basisadresse + 06h	Basisadresse + 0Eh
Port 7	Basisadresse + 07h	-

### Umschalten zwischen Ein- und Ausgang des 16 Bit Ports

Bei Port67 können alle Bits gleichzeitig auf Ein- oder Ausgang programmiert werden. Nach dem Einschalten der Versorgungsspannung sind alle Bits als Eingang konfiguriert.

Um den 16 Bit Port auf Ausgang zu schalten, muss das Bit 0 des Steuerregisters auf 0 gesetzt werden. Durch setzen auf 1 wird der Port wieder auf Eingang geschaltet und alle Pins des Ports werden hochohmig.

Die Anweisung

XOUT (Basisadresse + 0Eh, 00000000b)

schaltet alle Bits des Port67 auf Ausgang.

Die Anweisung

XOUT (Basisadresse + 0Eh, 00000001b)

schaltet alle Bits des Port67 auf Eingang.

Beachten Sie, dass das Steuerregister nur beschrieben werden kann. Lesen mit XIN liefert undefinierte Werte.

### Lesen des 16 Bit Ports

Der Port67 besitzt einen 16 Bit Lesebuffer. Dadurch ist es möglich alle 16 Bits des Ports gleichzeitig zu lesen. Wird Bit 1 des Steuerregisters von 1 auf 0 geändert, werden die Logikzustände der 16 Bits in den Lesebuffer übernommen. Anschließend kann durch Lesen des Port 6 das Low-Byte und durch Lesen des Port 7 das High-Byte des Lesebuffers erfaßt werden.

Beispiel:

XOUT (Basisadresse + 0Eh, 11b)

‘ Schalte Port67 auf Eingang und

‘ setze Bit 1 des Steuerregisters

XOUT (Basisadresse + 0Eh, 01b)

‘ lösche Bit 1 des Steuerregisters

‘ Logikzustände werden dadurch jetzt gepuffert

low = XIN (Basisadresse + 06h)

‘ lesen des Low-Bytes

high = XIN (Basisadresse + 07h)

‘ lesen des High-Bytes

Wert = low + 256 \* high

### Setzen des 16 Bit Ports

Der Port67 besitzt einen 16 Bit Ausgangspuffer. Dadurch ist es möglich alle 16 Bits des Ports gleichzeitig zu setzen. Durch eine XOUT Anweisung auf Port6 wird das Low-Byte des Puffers gesetzt, durch eine XOUT Anweisung auf Port7 das High-Byte. Wird Bit 1 des Steuerregisters von 0 auf 1 geändert, wird der Wert des Puffers übernommen und alle 16 Bits gleichzeitig auf High oder Low gesetzt.

Beispiel:

XOUT (Basisadresse + 06h, lowByte) ‘ setzen des Low-Bytes des Puffers

XOUT (Basisadresse + 07h, highByte) ‘ setzen des High-Bytes des Puffers

XOUT (Basisadresse + 0Eh, 00b) ‘ Schalte Port67 auf Ausgang

XOUT (Basisadresse + 0Eh, 10b) ‘ setze Bit 1 des Steuerregisters,

‘ gepufferte Zustände werden dadurch ausgegeben

# EP30-PPIO64



## Universelles I/O-Erweiterungsmodul

### Maximalwerte (Überschreiten der Maximalwerte kann zur Zerstörung des Moduls führen)

Versorgungsspannung VCC	5.5V
Eingangsspannung auf allen Eingängen	5.5V
maximaler Strom pro Ausgang	100mA
Betriebstemperatur:	min. 0°C max. 50°C
Gesamtstrom aller Ausgänge incl. V3P	200mA
reservierte Pins sind mit GND zu verbinden	

### Spezifikationen

Parameter	Bedingungen	Wert
Versorgungsspannung		5V DC +-5%
Versorgungsstrom:	alle I/Os auf Eingang programmiert	typ.: 10mA max.: 50mA
Input low voltage		0V...0,8V
Input high voltage		2,0V...5.5V
Output high current	Voh=2.4V	20mA
Output low current	Vol=0.4V	20mA

# EP30-PPIO64

Universelles I/O-Erweiterungsmodul



## Modulgröße

