



iL-TROLL Der superschnelle BASIC-PIC

- + Kostenfreier BASIC-Compiler (Download)
- + echter Compiler (kein langsamer BASIC-Interpreter)
- + Editor (farbige Syntax, ein-/ausklappbare Bereiche)
- + Programmierung mit einem Klick
- + Debugger (u.a. Schritt-für-Schritt, Variablen ändern, Breakpoint, selektive Echtzeit)
- + ausführliche Hilfe und Dokumentation

Eigenschaften:

Quarzfrequenz 20MHz (fest)
 bis zu 5 Million BASIC-Anweisungen pro Sekunde
 Timer 0 (8-Bit) / Timer 1 (16-Bit) / Timer 2 (8-Bit)
 2 x Capture / Compare, PWM-Modul (bis 10-Bit Auflösung)
 I²C (Master / Slave) und SPI (Master)
 USART (serielle Schnittstelle)
 Analog-Eingänge (10-Bit Auflösung; 3 Kanal fest)

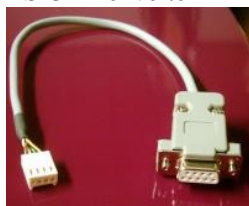
Elektrische Spezifikationen:

Spannungsversorgung: 2,0 V bis 5,5 V (empfohlen 5,0 V)
 Strombedarf: < 2 mA bei 5 V und 4 MHz
 typ. 20 µA bei 3 V und 32 kHz
 typ. < 1 µA in Sleepmodus

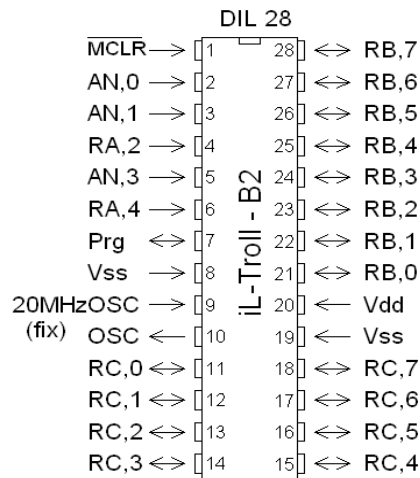
Programmieradapter:

Mit diesem Adapter werden die Datenleitungen normgerecht auf RS232-Pegel gebracht. Ebenso erlaubt er es, den iL-TROLL per Schnittstelle zurück zusetzen

Falls Ihr PC keine serielle Schnittstelle besitzt, kann ein USB-RS232-Konverter verwendet werden.



Fertig aufgebauter Adapter (ab Lager lieferbar)

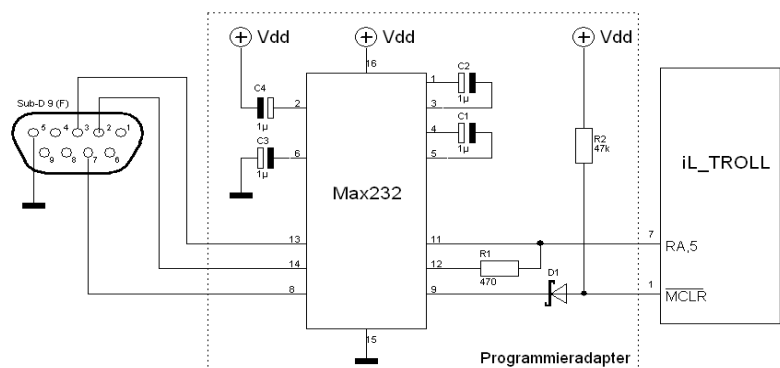


Port RB 8 digitale Ein- bzw. Ausgänge, beliebig wähl
 Port RC 8 digitale Ein- bzw. Ausgänge, beliebig wähl
 Port RA 3 analog-, 2 digitale Eingänge

Datenhaltbarkeit mehr als 40 Jahre
 Sleep-Modus (Stromsparmmodus)

7936 Worte Speicher für Anwenderprogramme
 150 Bytes Datenspeicher Anwendervariablen (RAM)
 iL-TROLL-Betriebssystem 256 Worte Programmspeicher
 iL-TROLL-RAM-Bedarf: max 40 Bytes

max. Gesamtverlustleistung: 1 W
 max. Strom aus Vss: 300 mA
 max. Strom in Vdd: 250 mA
 max. Strom am IO-Pin: 25 mA (gilt für High und Low)
 max. Gesamtstrom pro Port: 200 mA



BASIC-Befehle des iL-TROLLs (Auswahl)

Befehl	Funktion	Befehl	Funktion
ADDELAY	Zeit zum Laden des Haltekapazitors beim AD-Wandler	INTEND	markiert das Ende der Interrupt-Service-Routine
ADINP	Analogwerte einlesen (8-Bit)	INTPROC	markiert den Anfang der Interrupt-Service-Routine
BINTOASC	konvertiert 8- oder 16-Bit Variable in ASCII-Zeichenkette (wie CONASC)	LDCLEAR	LCD-Anzeige löschen
BINTOBCD	konvertiert 8- oder 16-Bit Variable in eine BCD-gepackte Zahl	LCDEDELAY	zur Anpassung des Timings
BINTODEC	konvertiert 8- oder 16-Bit Variable in Zahlenkette (wie CONDEC)	LCDINIT	initialisiert eine LCD-Anzeige mit HD44780 Controller (4-Bit)
BITPOS	ordnet einer Zahl zwischen 0 und 7 deren Bitposition zu	LCDTYPE	verschiedene Anschlussvarianten für LCDs (z. B. auch über 74LS164)
CLOCK	Echtzeituhr mittels internem Timer TMR0	LCDWRITE	schreibt Daten auf die LCD
CLOCKI	Echtzeituhr mittels internem Timer TMR1	LET	Anweisung für eine Zuweisung (obligatorisch)
CLRWDT	setzt den internen Watchdog zurück (in Verbindung mit SWDTUSR und WDT_ON)	LOCATE	positioniert den Cursor an einer beliebigen Stelle auf der LCD
CONASC	konvertiert 8- oder 16-Bit Variable in ASCII-Zeichenkette	LOFREQ	erzeugt Frequenzen bis ca. 2 kHz
CONDEC	konvertiert 8- oder 16-Bit Variablen in Zahlenkette	LOOKDN	Tabellen lesen
CURSOFF	schaltet Cursor auf der LCD aus	LOOKUP	Tabellenindex lesen
CURSON	schaltet Cursor auf der LCD ein	LOW	I/O-Pin wird Ausgang und auf Low gesetzt
DATA	definiert ein Datenfeld für Konstanten	ON...GOSUB	mehrfach bedingte Verzweigung
DEC	erniedrigt den Inhalt einer Variablen um 1	ON...GOTO	mehrfach bedingte Verzweigung
DELAY	kurze Verzögerung (Auflösung 100 us)	OUTP	I/O-Pin wird als Ausgang definiert
DOZE	Strom sparender Kurzschlaf	OUTPUT	8-Bits werden auf Port ausgegeben
DTMFOUT	erzeugt DTMF-Signale (Tonwahl beim Telefon)	PRINT	zum Debuggen
EEDATA	Diese Daten werden vom Programmiergerät in das interne EEPROM geschrieben	PULSIN	misst die Pulsdauer
END	Programmende, Controller geht in Stromsparmodus	PULSOUT	gibt einen Puls auf einem I/O-Pin aus
ERR	abfragt derdie internen Error-Variablen, z. B. Timeout, Overflow ab	PWM	erzeugt ein pulsweitenmoduliertes Signal, z. B. für Analogausgabe
FREQIN	zählt die Anzahl der Impulse innerhalb einer bestimmten Zeit	RANDOM	erzeugt eine Zufallszahl
FOR TO NEXT	FOR-NEXT-Schleifen, bis zu 16 Schachtelungsebenen	RCTIME	misst die Lade-/Entladezeit eines Kondensators, z. B. für Widerstandsmessungen
GOSUB	Sprung in ein Unterprogramm, bis zu 4 Ebenen	READ	lesen der Daten aus dem EEPROM (nur für PICs mit eingebauten Daten-EEPROM)
GOTO	Sprung zu einer Programmmarke, keine Zeilennummern!	READATA	liest einen Datenwert aus dem Datenfeld
HIGH	I/O-Pin wird Ausgang und auf High gesetzt	REM	leitet eine Kommentarzeile ein
I2CDELAY	verlangsamt die I2C-Kommunikation, z. B. bei langen Leitungen	RES	Pegel des I/O-Pins auf Low setzen
I2CHARDS	aktiviert das I2C-Hardwaremodul des PICs als SLAVE	RESTORE	setzt den Lesezeiger für READATA an eine bestimmte Position
I2CINIT	initialisiert 2 Pins für die I2C-Kommunikation	RETURN	Ende des Unterprogramms
I2CREAD	lesen aus einem I2C-Baustein	REVERS	I/O-Pin wird Ausgang und die aktuelle Information invertiert
I2CRD	lesen aus einem I2C-Baustein ohne Generierung von START- bzw. STOP-Bedingung	SERIN	liest Daten seriell ein
I2CSLAVE	Controller wird zu einem universellen SLAVE-Baustein	SEROUT	gibt Daten seriell aus
I2CST	generiert eine START-Bedingung	SET	Pegel des I/O-Pins wird auf High gesetzt
I2CSP	generiert eine STOP-Bedingung	SETBAUD	ändern der Baudrate bei Interrupt gesteuertem UART-Empfang
I2CWRITE	schreiben in einen I2C-Baustein	SLEEP	Controller wird in den SLEEP-Modus versetzt
I2CWR	schreiben in einen I2C-Baustein ohne Generierung von START- bzw. STOP-Bedingung	SOUND	erzeugt Frequenzen zwischen 1 kHz und 20 kHz
IF THEN ELSE	bedingte Verzweigungen	SPLAY	Verbesserter Ersatz für SOUND und LOFRQ
INC	erhöht den Inhalt einer Variablen um 1	SWAP	vertauscht die beiden Halbbytes in einer Variablen
INKEY	4x4 Tastaturmatrix abfragen, an gleichen Port wie LCD möglich (4x3)	TOGGLE	Pegel des I/O-Pins wird invertiert
INP	I/O-Pin wird als Eingangspin definiert	TRIS	initialisiert das Datenrichtungsregister
INPUT	gesamter Port (8-Bit) wird eingelesen	TXDDELAY	Pause zwischen zwei Zeichen beim Senden mittels UART
INTERRUPT	erlaubt das Einklinken benutzerdefinierter Interrupts	WAIT	verzögert den Programmablauf (kein Stromsparmodus)
		WRITE	schreibt Daten in das EEPROM (nur für PICs mit eingebautem Daten-EEPROM)

Weitere Informationen finden Sie im Internet z.B. Handbuch

Dieses Datenblatt wurde nach bestem Wissen erstellt. Trotzdem kann keinerlei Haftung übernommen werden.