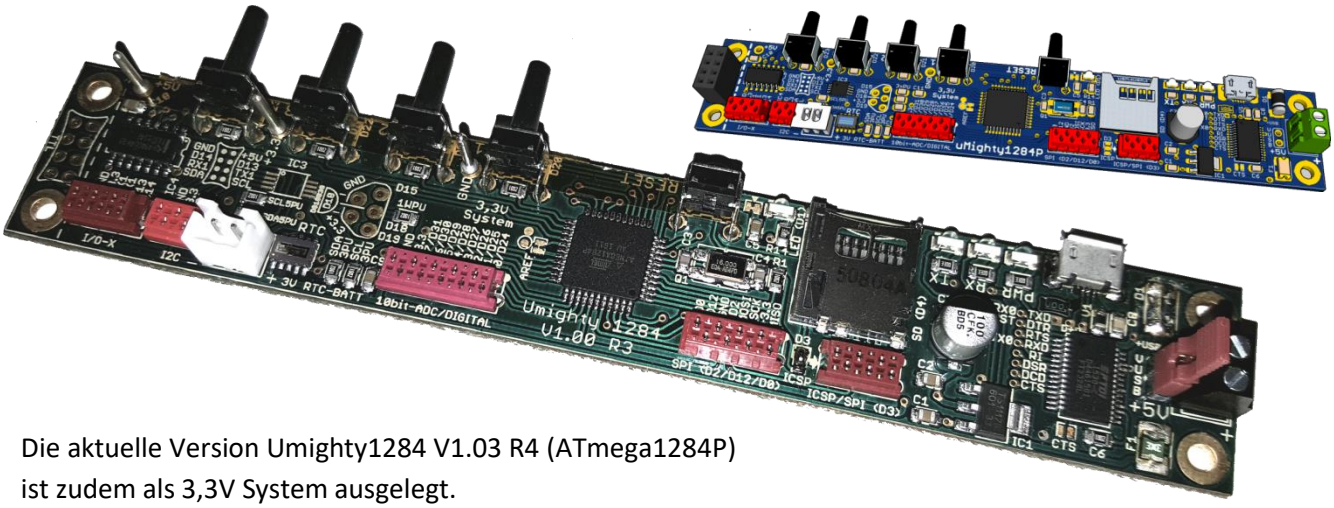
uMIGHTY1284P

Ein kleiner ***ARDUINO*** mit viel Speicher für Datalogging (Wetterstation), LED Stripe Anwendungen u.v.a.m.

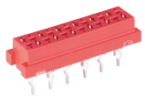
Der Mighty1284 (ATmega1284/ATmega1284P) verfügt im Vergleich zum Arduino UNO über viel mehr Speicher, und hat sogar mehr RAM als ein Arduino Mega2560.



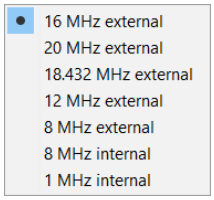
Die aktuelle Version Umighty1284 V1.03 R4 (ATmega1284P)
ist zudem als 3,3V System ausgelegt.

Ein OnBoard verbautes USB-Interface mit FT232RL mit micro USB-Anschluss, 3,3V Spannungsregler, SD-Speicherkarten Slot für micro-SD Karten, eine Batteriegestützte Echtzeituhr (RV3029C2), vier Eingabe- und ein Reset-Taster sowie zwei aktive bidirektionale Level-Shifter für ausgewählte Funktionen zum Anschluss an 5V Peripherie runden die Features dieser Platine ab.

Um Platz zu sparen wurden zum Anschluss von externen Bauteilen und Geräten anstelle „normaler“ Buchsenleisten (wie am Arduino üblich) sog. **MicroMatch** Steckverbinder verwendet. Diese MicroMatch Steckverbinder können auch durch normale einreihige Stift/Buchsenleisten im Raster 0.1“/2,54mm ersetzt werden. Die jeweils zweite Reihe ist um einen halben Raster (0.05“/1,27mm) versetzt im Abstand von 0.1“/2,54mm zur ersten Reihe definiert. Somit kann man eine Verbindung zum uMighty1284P entweder mit einem Standard Flachbandkabel (AWG28) und den entsprechenden aufgepressten MicroMatch Stecker oder den sog. DUPONT Jumper-Wire Stiftkontakten und anderen Steckbrücken herstellen. Auch eine Kombination dieser Technologien ist denkbar.



Obwohl der ATmega1284P laut Datenblatt mit 3,3V Betriebsspannung eine Maximale Taktfrequenz von 10MHz erlaubt, wurde auf dem uMighty1284P ein 16MHz Quarz verbaut. Trotz dieses **erheblichen übertaktens** konnten bisher keine Probleme festgestellt werden. **Im Extremfall** muss der verbaute 16MHz Quarz gegen einen 12MHz oder 8MHz Quarz gewechselt werden. Die jeweils verwendete Quarzfrequenz kann in der Arduino IDE vor der Kompilierung eingestellt werden, mehr hierzu etwas weiter unten.



Der uMighty1284P wurde **absichtlich lang und schmal** ausgelegt (Größe der Platine 2,54 mm x 151 mm) um diesen in z.B. **Bilderahmen** (IKEA Ribba o.ä.) **einzubauen** und eine Bedienung über die Taster von der (Rück-)Seite aus zu ermöglichen. Somit sind sehr schöne Licht und Deko Objekte z.B. mit LED Streifen (WS2812 o.ä.) sehr einfach realisierbar. Auch ein offline **Datalogging z.B. eine Wetterstation** mit Temperatur, Luftdruck und Luftfeuchte sind mit der Echtzeituhr, dem vorhandenen SD-Speicherkarten Slot sowie entsprechender externer Sensoren (BMP085/DHT22 u.v.a.m.) sehr einfach zu realisieren.

IoT (Internet of Things): Auch an einen Anschluss (I/O-X) des beliebten ESP8266 (z.B. ESP01) wurde gedacht. Mit dem entsprechenden, sehr günstigen Modul, kann der uMighty1284P auch per WLAN in das Heimnetzwerk / Internet eingebunden werden.

Technische Daten im Vergleich

	Arduino UNO	uMighty1284P	Arduino MEGA2560
CPU	ATmega 328	ATmega 1284P	ATmega 2560
Taktfrequenz	**16 MHz**	**Übertaktet mit 16 MHz** (@3,3V sicher mit 10 MHz)	**16 MHz**
Programmspeicher	32 kByte	128 kByte	**256 kByte**
Variablenspeicher (RAM)	2 kByte	**16 kByte**	8 kByte
EEprom (onChip)	1 kByte	**4 kByte**	**4 kByte**
RTC Echtzeituhr (onBoard)	-	**RV 3029 C2** Library für Arduino	-
SD-Card (onBoard)	-	**Micro SD-Card SDHC**	-
3,3V zu 5,0V LevelShifter (onBoard, bidirektional, aktiv)	-	**RXD1, TXD1, SCL, SDA und GPIO 13, 14**	-
Eingabe	-	**4x Taster**	-
LEDs (onBoard)	Power, RXD, TXD, Status	Power, RXD, TXD, **Status (exclusiv)**	Power, RXD, TXD, Status
Serielles Interface für Programmierung (onBoard)	USB-B	Micro USB-B	USB-B
Stromversorgung	über USB oder 7V bis 12V (Buchse)	über USB oder 5V (Schraubklemme)	über USB oder 7V bis 12V (Buchse)

Anmerkung: Den ATmega1284 gibt es auch noch im klassischen IC-Gehäuse PDIP 40, welches in sog. Breadboards oder Lochrasterplatinen ohne Adapter eingesteckt/eingelötet werden kann. Diese Bauform ist sehr einfach zu handhaben.

Programmierung

Der uMighty1284P kann wie ein „normaler“ Arduino über die USB Schnittstelle programmiert werden. Auch eine Programmierung mit einem ISP Programmer ist wie mit den originalen Arduinos möglich.

Download der Arduino IDE

Die Arduino Software anpassen

Bevor der uMighty1284P mit der Arduino Software verwendet/programmiert werden kann, muss diese erst noch erweitert werden.

Hierzu ist über den Boardverwalter die entsprechende Erweiterung zu installieren

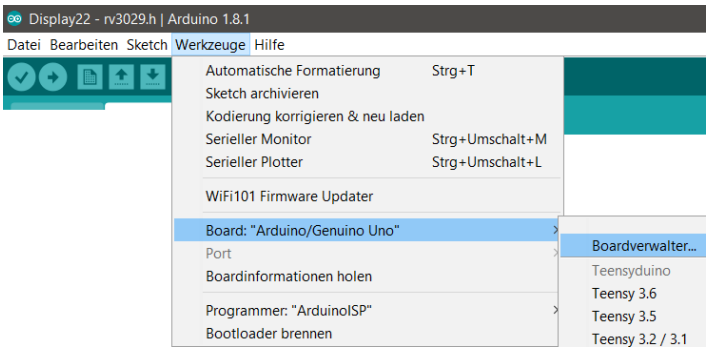
Beispiel getestet mit ARDUINO Version 1.8.1 und aktualisierten Library‘s

Im Menü (oben) unter

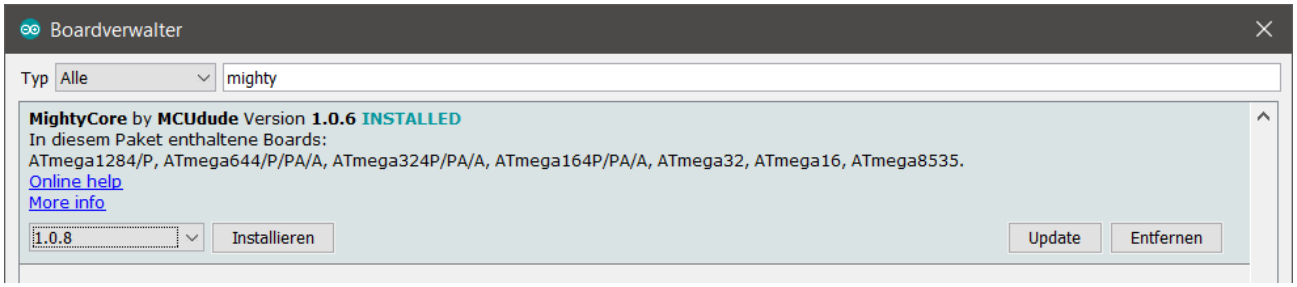
Werkzeuge

Board:

Boardverwalter



Nach „mighty“ suchen und installieren (hier die Version 1.0.8 oder neuer verwenden)



Ab dann können die Sketche für den uMighty1284P kompiliert werden.

Allerdings sind **vor der Kompilation noch einige Einstellungen auszuwählen**.

Optionen für die Kompilierung

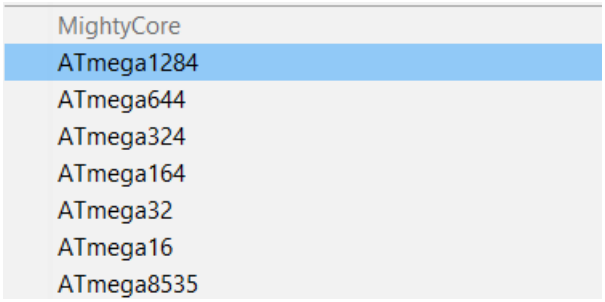
Zunächst muss, sofern nicht bereits geschehen das entsprechende Board ausgewählt werden.

Im oberen Menü unter

Werkzeuge

Board:

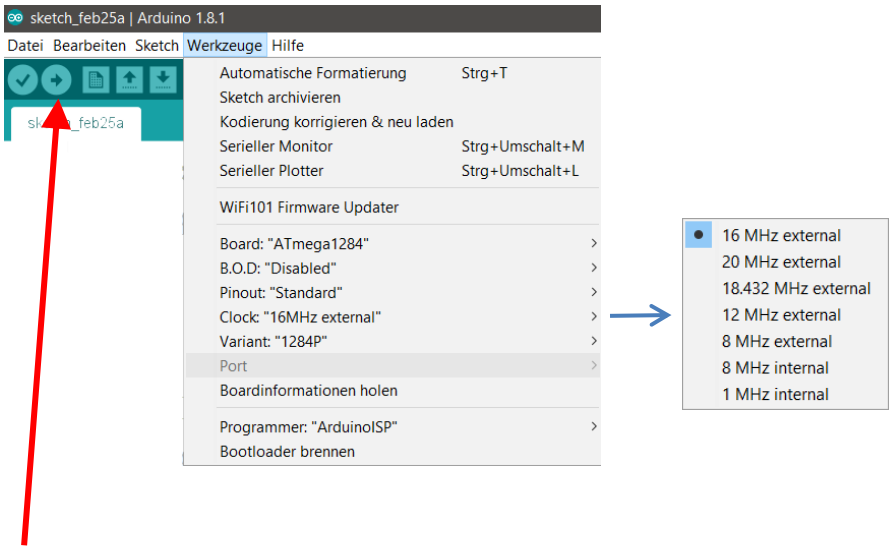
im Abschnitt MightyCore den ATmega1284 a**uswählen**



Erst danach sind im Menü Werkzeuge die weiteren Einträge zu finden:

Die folgenden Einstellungen auswählen:

Board:	„ATmega1284“	
B.O.D:	„Disabled“	(Burn Out Detection)
Pinout:	„Standard“	
Clock:	„16MHz external“	(oder entsprechende Frequenz des Quarzes)
Variant:	„1284P“	
Port:	*je nach erkannten USB Port*	*(hier noch nicht angeschlossen daher grau)*



Ab sofort kann ein Arduino – Sketch wie gewohnt kompiliert und übertragen werden

OnBoard Funktionen / Stecker

LED

Die LED ist beim uMighty1284P an GPIO 1 wie z.B. für den BLINK Sketch verwendet

SPI (Serial Peripheral Interface 0V / 3,3V)

Der SPI-BUS ist ein Hardware SPI
Es sind zwei SPI Anschlüsse auf dem Board und der SD-Karten Sockel vorhanden.

Der 6-polige Anschluss kann auch als **ICSP** Verwendet werden, hierzu den kleinen Lötjumper auf die Position ICSP umlegen. Hiermit kann die Firmware (der Arduino Bootloader) auf den uMighty1284P programmiert werden oder Programme ohne Bootloader mit einem Programmieradapter (ISP) übertragen werden. Die Stecker Belegung ist dann konform mit dem Atmel Standard für ISP Programmierung.
ACHTUNG: VTG entspricht hier +3,3V, also keine externe Spannung (5V) zuführen!

6-poliger SPI / ISP (ICSP) Stecker **SS = 3**

Im Normalfall sollte der Lötjumper in der Position „D3“ gesetzt sein.
Dies ist die Einstellung des Pins für SS/CS (Chip Select) am 6-poligen SPI Stecker

Durch Umsetzen des Lötjumpers D3/ICSP kann dieser Anschluss zu einem ISP/ICSP geändert werden

8-poliger SPI Stecker **SS = 2**

Pin für SS/CS ist hier 2, zusätzlich sind die GPIO 12 und 0 an diesen Stecker verfügbar

SD-Card Sockel SPI **SS = 4**

Die SD Karte wird mit dem SS-Pin 4 angesprochen

SPI Stecker Belegungen (0V / 3,3V)

	ICSP / SPI	SPI	SD-Card
MISO	1 – GPIO 6	1 – GPIO 6	7 – GPIO 6
+3,3V	2 – +3,3V	2 – +3,3V	4 – +3,3V
SCK	3 – GPIO 7	3 – GPIO 7	5 – GPIO 7
MOSI	4 – GPIO 5	4 – GPIO 5	3 – GPIO 5
➔ SS (CS) / RST	5 – GPIO **3** / **RESET** ❶	5 – GPIO **2**	2 – GPIO **4**
GND	6 – GND	6 – GND	6 – GND
Extra 1	–	7 – GPIO 12	–
Extra 2	–	8 – GPIO 0	–

❶ GPIO **3** / **RESET** ABHÄNGIG vom Lötjumper neben dem Stecker ICSP / SPI

1W Bus (one Wire 0V / 3,3V)

In der Mitte der Platine sind drei zusätzliche Ports Pins + Versorgungsanschlüsse verfügbar, welche als 1W-Bus verwendet werden können, D15, D18 und D19.

An diesen Pins ist jeweils ein diskreter PullUp Widerstand (5,1k) vorhanden.

10 – Bit ADC / Digital (0V – max. 3,3V)

Stecker Belegung

	analog	digital
Pin	**ADC**	alternativ **GPIO**
1	A0	24
2	A1	25
3	A2	26
4	A3	27
5	A4	28
6	A5	29
7	A6	30
8	A7	31
9	+3,3V	+3,3V
10	GND	GND

I2C (auch als IIC / I²C / TWI oder 2W-Bus bekannt 0V / 3,3V)

Der I2C Anschluss teilt sich die gleichen Signale über den Level Shifter am TTL Anschluss

Pin	Signal	GPIO	Zu beachten
1	SCL	16	auch am TTL Anschluss
2	SDA	17	auch am TTL Anschluss
3	+3,3V		
4	GND		

Echtzeituhr I2C (onBoard)

I2C RTC RV 3029 C2 Adresse HEX = **0x56**

Eine passende Library (rv3029.h) und ein entsprechendes Beispiel ist u.a. bei **Elektor** (Link zum Beitrag) zu finden

I/O-X (0V / 3,3V)

Der I/O-X Anschluss teilt sich die gleichen Signale über den Level Shifter am TTL Anschluss

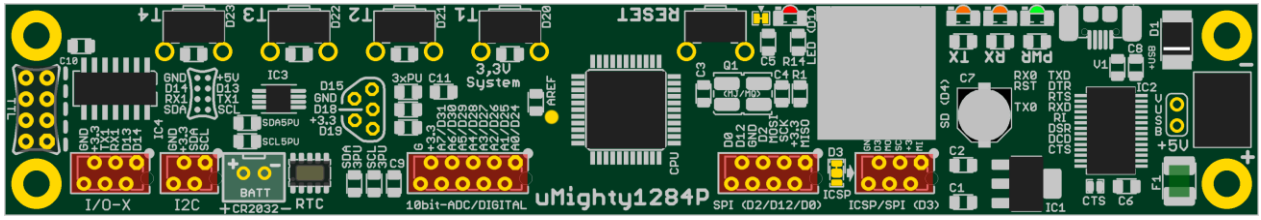
Pin	Signal	GPIO	Zu beachten
1		14	auch am TTL Anschluss
2		13	auch am TTL Anschluss
3	RXD1	10	auch am TTL Anschluss
4	TXD1	11	auch am TTL Anschluss
5	+3,3V		
6	GND		

TTL I/O (5V Pegel) (0V / 5V)

Der TTL Anschluss teilt sich die gleichen Signale über den Level Shifter an **I/O-X** und **I2C** Anschlüssen

Pin	Signal	über Level Shifter an GPIO
1	SCL	16
2	SDA	17
3	TXD1	11
4	RXD1	10
5	13	13
6	14	14
7	+5V	
8	GND	

Platinen Pinout



Anmerkungen