

Elektro_{xy}

ATTiny85

Mikrocontroller-Board zum direkten Einstecken in die USB-Buchse

Kurzanleitung

ElektroXY · Bach IT Service UG (haftungsbeschränkt)

Lindenstr. 7 · 53506 Kesseling · www.elektroxy.de

Worum es geht

Dieses Board ist kaum größer als ein Daumennagel und wird direkt in die USB-Buchse gesteckt — ohne Kabel, ohne Adapter. Programmiert wird es mit der Arduino-IDE. Für kleine, fest umrissene Aufgaben ist es unschlagbar günstig und unauffällig: eine Lichtsteuerung, ein Taster, der etwas auslöst, ein Signalgeber, eine Tastatureingabe, die sich der Rechner als USB-Gerät gefallen lässt.

Der Preis dafür ist Beschränkung. Sechs Anschlüsse, davon zwei für USB belegt. Rund sechs Kilobyte Platz für Ihr Programm. Und vor allem: kein serieller Monitor, weil kein USB-Seriell-Wandler verbaut ist. Wer Fehler sucht, kann sich nicht wie gewohnt Werte ausgeben lassen — dazu unten mehr.

Der Ablauf beim Übertragen ist anders als bei jedem anderen Arduino-Board. Lesen Sie den Abschnitt „Programm übertragen“, bevor Sie es zum ersten Mal versuchen — sonst wundern Sie sich über eine Fehlermeldung, die gar keine ist.

Technische Daten

Eigenschaft	Wert
Mikrocontroller	ATtiny85
Taktfrequenz	16,5 MHz, intern erzeugt
Programmspeicher	8 KB, davon rund 6 KB nutzbar
Arbeitsspeicher	512 Byte SRAM
Dauerhafter Speicher	512 Byte EEPROM
Anschlüsse	6, beschriftet mit P0 bis P5
davon frei nutzbar	P0, P1, P2 und mit Einschränkung P3, P4
Analogeingänge	3
PWM-Ausgänge	3
Betriebsspannung	5 V über USB
USB	in Software nachgebildet, kein Wandlerbaustein

Die rund zwei Kilobyte Unterschied beim Programmspeicher belegt der Bootloader — das kleine Programm, das beim Einstecken auf neue Übertragungen wartet. Ohne ihn bräuchten Sie ein separates Programmiergerät.

Anschlüsse

Aufdruck	Funktionen	Hinweis
P0	PWM, I ² C SDA	meist frei; bei manchen Ausführungen die Onboard-LED
P1	PWM	bei den meisten Ausführungen die Onboard-LED
P2	Analog A1, I ² C SCL	frei nutzbar
P3	Analog A3	USB-Datenleitung — nur eingeschränkt nutzbar
P4	Analog A2, PWM	USB-Datenleitung — nur eingeschränkt nutzbar
P5	Analog A0	Reset-Anschluss, im Auslieferungszustand gesperrt

P3 und P4 hängen an den USB-Datenleitungen. Solange das Board in der Buchse steckt, sind sie belegt. Im eigenständigen Betrieb an einem Netzteil lassen sie sich verwenden — nur eben nicht gleichzeitig mit der USB-Verbindung.

Finger weg von P5: Der Anschluss lässt sich nur nutzen, wenn Sie eine Konfigurationseinstellung im Chip ändern. Danach ist das Board über USB nicht mehr programmierbar, und die Änderung lässt sich nur mit einem Hochvolt-Programmiergerät rückgängig machen.

Arduino-IDE einrichten

1. Boardpaket hinzufügen

Der ursprüngliche Hersteller Digistump pflegt sein Paket seit Jahren nicht mehr; mit aktuellen Versionen der Arduino-IDE funktioniert es nicht zuverlässig. Verwenden Sie die gepflegte Abspaltung. Tragen Sie in den Einstellungen unter „Zusätzliche Boardverwalter-URLs“ ein:

```
https://raw.githubusercontent.com/ArminJo/DigistumpArduino/master/package\_digistump\_index.json
```

Danach im Boardverwalter nach „Digistump AVR“ suchen und installieren.

2. Board auswählen

Einstellung	Wert
Board	Digispark (Default - 16.5 MHz)
Port	wird nicht benötigt und bleibt leer

Dass kein Port ausgewählt werden muss, ist richtig so — die Übertragung läuft nicht über eine serielle Schnittstelle, sondern direkt über USB.

3. Treiber unter Windows

Windows braucht einen Treiber für den Bootloader. Das Boardpaket bringt ihn mit; er liegt nach der Installation im Unterordner mit dem Namen „digistump“ und wird über die mitgelieferte Installationsdatei eingerichtet. Unter Linux genügt eine Regel für udev, unter macOS ist nichts nötig.

Programm übertragen

Und hier der Punkt, der alle beim ersten Mal stolpern lässt. Bei anderen Boards stecken Sie das Board an und drücken dann auf Hochladen. Hier ist es umgekehrt:

1. Board aus der USB-Buchse ziehen.
2. In der Arduino-IDE auf Hochladen klicken.
3. Warten, bis unten die Aufforderung erscheint, das Board einzustecken.
4. Board einstecken. Die Übertragung läuft dann von selbst.

Sie haben dafür rund 60 Sekunden Zeit. Danach bricht der Vorgang ab und Sie fangen von vorn an. Der Bootloader lauscht nämlich nur in den ersten Sekunden nach dem Einstecken — danach startet er Ihr Programm.

Genau deshalb sollten Sie es nicht überstürzen, wenn Ihr Programm die LED sofort ansteuert: Wirkt das Board wie tot, weil Ihr Programm läuft, ziehen und stecken Sie es einfach neu, während die IDE wartet.

Erstes Programm

Die Onboard-LED sitzt je nach Ausführung an P1 oder an P0. Probieren Sie im Zweifel beide — mehr als zwei Versuche braucht es nicht:

```
#define LED_PIN 1      // bei manchen Ausfuehrungen 0

void setup() {
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_PIN, HIGH);
  delay(200);
  digitalWrite(LED_PIN, LOW);
  delay(1800);
}
```

Die lange Pause ist Absicht: So bleibt genug Zeit, um zu erkennen, ob das Board Ihr Programm ausführt oder noch im Bootloader wartet.

Ohne seriellen Monitor auskommen

Serial.print gibt es hier nicht. Zum Eingrenzen von Fehlern haben sich drei Wege bewährt:

- Die LED als Signalgeber verwenden — unterschiedlich viele Blinkzeichen an verschiedenen Stellen im Programm verraten, wie weit es kommt.
- Die Bibliothek DigiCDC verwenden. Sie meldet das Board zusätzlich als serielle Schnittstelle an. Das kostet Programmspeicher und macht das Zeitverhalten unsauber, reicht zur Fehlersuche aber aus.
- Das Programm zuerst auf einem größeren Board entwickeln und erst zum Schluss auf den ATtiny übertragen. Für alles, was über Blinken hinausgeht, ist das der bequemste Weg.

Womit Sie rechnen müssen

Thema	Einschränkung
Programmgröße	rund 6 KB; viele Bibliotheken passen nicht hinein
Arbeitsspeicher	512 Byte — lange Texte und große Felder sprengen ihn
delay und millis	laufen etwas ungenau, weil der Takt intern erzeugt wird
USB im Betrieb	belegt P3 und P4 und stiehlt Rechenzeit
Genauigkeit der Zeit	für eine Uhr ungeeignet; dafür ein RTC-Modul ergänzen
Analogeingänge	nur an P2, P3 und P4 — und die beiden letzten kollidieren mit USB

Das ist keine Kritik am Baustein, sondern seine Auslegung: Er soll wenige Dinge zuverlässig und billig erledigen. Für alles Größere gibt es passendere Boards.

Fehlersuche

Beobachtung	Wahrscheinliche Ursache	Abhilfe
„Micronucleus device not	Treiber fehlt oder Board zu spät	Treiber installieren, Board erst nach der

Beobachtung	Wahrscheinliche Ursache	Abhilfe
found“	eingesteckt	Aufforderung einstecken
IDE wartet und bricht nach 60 Sekunden ab	Board wurde gar nicht eingesteckt	Vorgang wiederholen und rechtzeitig einstecken
Board wird gar nicht erkannt	USB-Verlängerung oder passiver Hub dazwischen	Direkt in eine Buchse am Rechner stecken
Erste Übertragung geht, danach keine mehr	Programm blockiert den Bootloader	Board ziehen, Hochladen starten, dann einstecken
LED blinkt nicht	LED sitzt am anderen Anschluss	Im Sketch von P1 auf P0 wechseln
Programm passt nicht in den Speicher	Bibliothek zu groß für 6 KB	Schlankere Bibliothek suchen oder selbst ansteuern
Board verhält sich sprunghaft	Zu wenig Arbeitsspeicher, Stapel überschrieben	Felder verkleinern, Texte ins Flash legen
Zeiten stimmen nicht genau	Interner Taktgeber schwankt mit der Temperatur	Für genaue Zeiten ein RTC-Modul verwenden
Analogmessung an P3 oder P4 unbrauchbar	USB-Verbindung stört die Messung	Board eigenständig mit 5 V versorgen, ohne USB

Angaben ohne Gewähr. Alle Werte beziehen sich auf den Mikrocontroller ATtiny85 und die übliche Beschaltung der USB-Boards mit Micronucleus-Bootloader; die Bestückung kann je nach Ausführung abweichen.