

Programmierung ATTiny13a

Benötigte Komponenten:

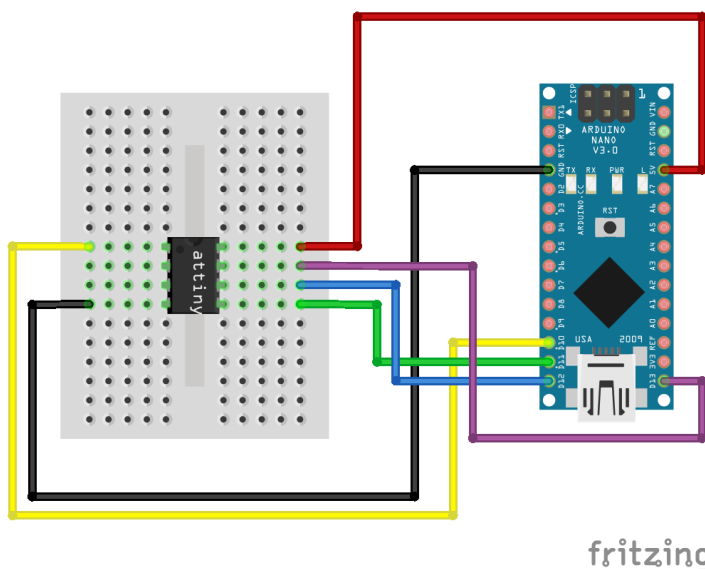
- ATTiny13a
- Arduino (z.B. Arduino Nano)
- 100µF Kondensator
- Steckplatine
- Kabel

Vorbereitung:

1. ArduinoISP auf Arduino laden (Sketch unter „Datei – Beispiele – ArduinoISP“ verfügbar)
2. Nach Upload: „Werkzeuge – Programmer – Arduino as ISP“
3. Jetzt muss die Bibliothek installiert werden: „Datei – Voreinstellungen – Zusätzliche Boardverwalter-URLs“; In die Eingabezeile „https://raw.githubusercontent.com/sleemanj/optiboot/master/dists/package_gogo_diy_attiny_index.json“ eintragen. Danach noch unter „Werkzeuge – Board: ... - Boardverwalter...“ in der Suchzeile „attiny“ eintragen und „DIY ATtiny by James Sleeman“ installieren und das Fenster schließen.
4. Nun noch den Mikrocontroller auswählen: „Werkzeuge – Board: ... - Attiny13“; Nun erscheinen unter „Werkzeuge“ weitere Einstellungen. Diese müssen wie in der folgenden Tabelle eingestellt werden:

Einstellung	Auswahl
Millis, Tone Support:	Millis Available, No Tone
Millis Accuracy:	Better Or Equal 1,666% Error (Default)
Print Support:	Bin, Hex, Dec Supported
Link Time Optimisation (1.6.11+ only):	LTO Enabled
Serial Support:	Half Duplex, Read+Write
Use Bootloader:	No (ISP Programmer Upload)
Processor Speed:	9.6MHz Internal Oscillator
Brown-out Detection Level:	2.7v
Processor Version	Attiny13a

Aufbau:



ATTiny13a	Arduino
VCC	5V
GND	GND
RST	10
SCK	13
MISO	12
MOSI	11

Optional (Arduino)
RST über Kondensator (100µF) auf GND

Mikrocontroller vorbereiten:

Zunächst muss der Bootloader auf den Mikrocontroller geschrieben werden:

„[Werkzeuge – Bootloader brennen](#)“

Sketch auf den Mikrocontroller laden:

Das Hochladen des Sketch funktioniert exakt gleich wie bei einem Arduino:

„[Sketch - Hochladen](#)“