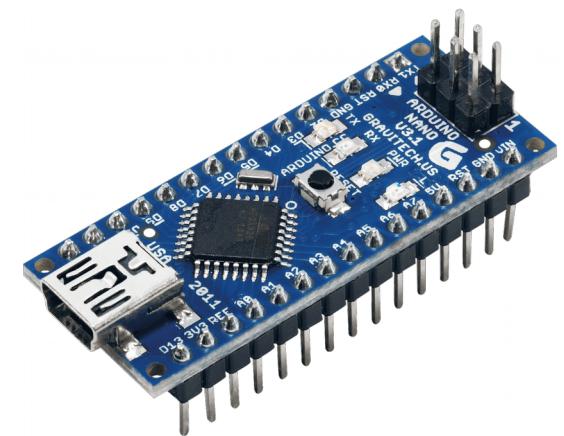
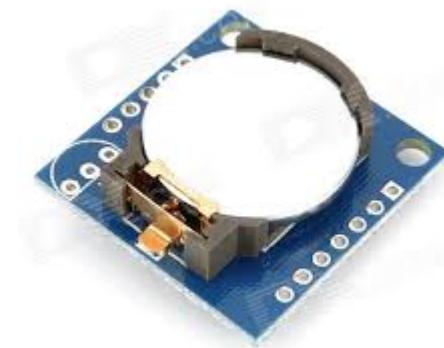




# arduino nano UHR

Mit RTC ds1307 und  
4x 7-segment anzeige tm1637



mark hoffmann



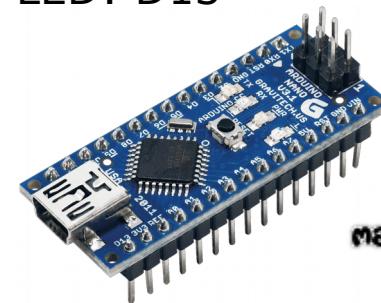
## Inhalt

- Inbetriebnahme Arduino Nano
- Einführung I2C Bus
- Inbetriebnahme Real-Time-Clock DS1307 (I2C)
- Inbetriebnahme 4x 7-Segment Anzeige TM1637
- Kombination RTC mit 7-Segment Anzeige => Uhr



# Arduino Nano

- Microcontroller: Atmel ATmega328P
- Operating Voltage: 5V
- Input Voltage: 5-12V
- Digital I/O Pins: 14 (6 PWM)
- Analog Input Pins: 8
- DC Current per I/O Pin: 40 mA
- Flash Memory: 32 KB (2 KB für Bootloader)
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- UART: D0(RX), D1(TX)
- SPI: D10(SS), D11(MOSI), D12(MISO), D13(SCK)
- I2C: A4(SDA), A5(SCL)
- PWM: D3, D5, D6, D9, D10, D11
- Interrupts: D2, D3
- LED: D13



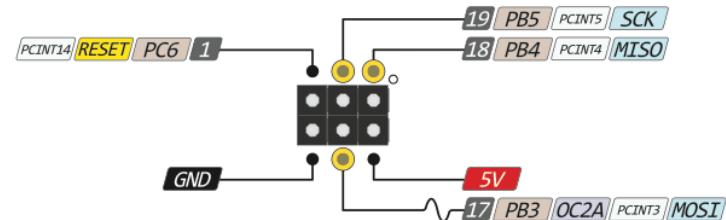
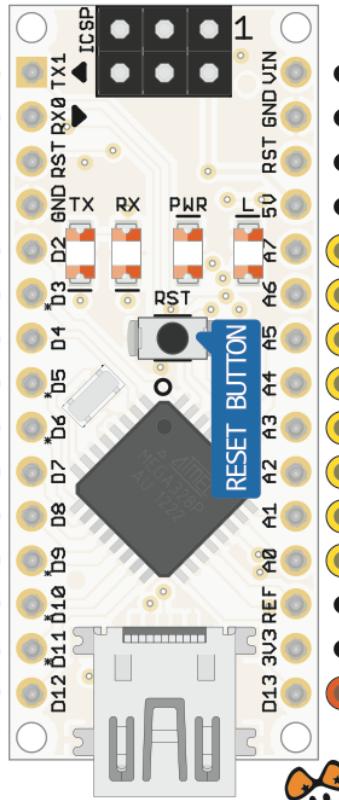
# NANO PINOUT

1  
0

PCINT17 TXD PD1 31  
PCINT16 RXD PD0 30  
PCINT14 RESET PC6 29

GND

PCINT18 INT0 PD2 32  
OC2B PCINT19 INT1 PD3 1  
XCK PCINT20 T0 PD4 2  
OC0B PCINT21 T1 PD5 9  
OC0A PCINT22 AIN0 PD6 10  
PCINT23 AIN1 PD7 11  
ICP1 PCINT0 CLK0 PB0 12  
PCINT1 OC1A PB1 13  
SS PCINT2 OC1B PB2 14  
MOSI PCINT3 OC2 PB3 15  
MISO PCINT4 PB4 16



VIN The input voltage to the board when it is running from external power. Not USB bus power.

GND

29 PC6 RESET PCINT14

5V

22 ADC7

19 ADC6

28 PC5 PCINT13 ADC5 SCL

27 PC4 PCINT12 ADC4 SDA

26 PC3 PCINT11 ADC3

25 PC2 PCINT10 ADC2

24 PC1 PCINT9 ADC1

23 PC0 PCINT8 ADC0

21 AREF

3V3

17 PB5 PCINT5 SCK

A7

A6

19A5

18A4

17A3

16A2

15A1

14A0

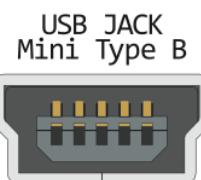
13

- Power
- GND
- Serial Pin
- Analog Pin
- Control
- INT
- Physical Pin
- Port Pin
- Pin function
- Interrupt Pin
- PWM Pin
- Port Power ⚠

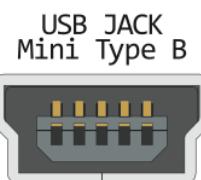
⚠ The power sum for each pin's group should not exceed 100mA

⚠ Absolute MAX per pin 40mA recommended 20mA

🚫 Absolute MAX 200mA for entire package



⚠ Analog exclusively Pins



www.bq.com



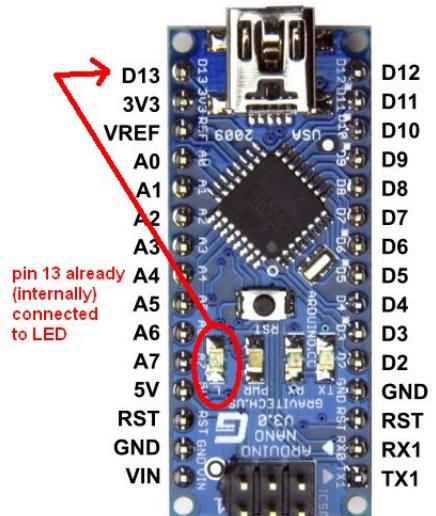
19 AUG 2014

ver 3 rev 1



## Arduino Nano

- „Hello World“ per LED
- On-Board LED (Pin D13)
- Beispiel: Arduino IDE / Beispiele / Basics / Blink





## I2C Bus

- Philips (jetzt NXP) Erfindung (1982)
- Abkürzung: Inter-Integrated Circuit
- Gedacht für geräteinterne Kommunikation zwischen ICs, z. B. in Fernsehgeräten und CD-Playern
- Auch genannt TWI (Two-Wire-Interface)  
lizenzrechtlich bedingt
- 1 Master, mehrere Slaves

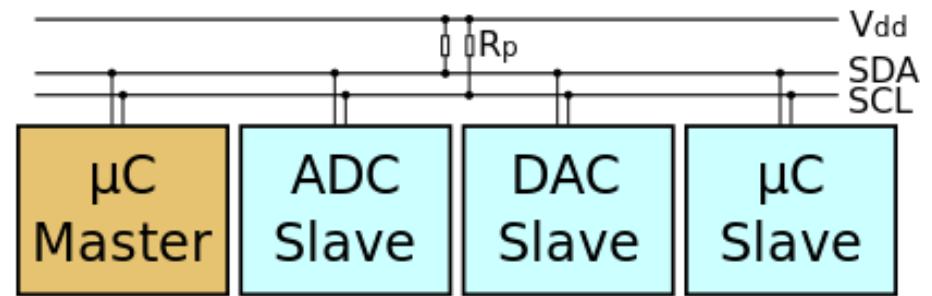


I2C Logo



## I2C Bus

- Benötigt zwei Signalleitungen:  
Takt- (SCL = Signal Clock) und  
Datenleitung (SDA = Signal Data)
- Benötigt Pull-Up Widerstände ( $R_p$ ),  
z.B. 4,7kOhm
- Angeschlossene Geräte haben Open-Collector-Ausgänge, was zusammen mit den Pull-up-Widerständen eine Wired-AND-Schaltung ergibt
- Der High-Pegel soll sein  $0,7 \times VDD$ , der Low-Pegel soll sein  $0,3 \times VDD$
- Positive Logik: ein High-Pegel auf der Datenleitung entspricht einer logischen „1“, der Low-Pegel einer „0“





## I2C Bus

- Der Bustakt wird immer vom Master vorgegeben - für verschiedene Modi ist jeweils ein maximal erlaubter Bustakt vorgegeben
- Auch beliebig langsame Taktraten können verwendet werden, falls diese vom Master-Interface unterstützt werden
- Einige ICs (z. B. ADC) benötigen eine bestimmte minimale Taktfrequenz, um ordnungsgemäß zu funktionieren
- Leitungslänge bis 100m mit 400kHz, Beschränkung Buskapazität max. 400pF

Modus	Maximale Übertragungsrate	Richtung
Standard Mode (Sm)	0,1 Mbit/s	bidirektional
Fast Mode (Fm)	0,4 Mbit/s	bidirektional
Fast Mode Plus (Fm+)	1,0 Mbit/s	bidirektional
High Speed Mode (Hs-mode)	3,4 Mbit/s	bidirektional
Ultra Fast-mode (UFm)	5,0 Mbit/s	unidirektional



## I2C Bus

- Slaves haben eindeutige vom Hersteller festgelegte Adressen von denen die untersten drei Bits (Subadresse) über drei Steuerpins festgelegt werden können - bis zu acht gleichartige ICs an einem Bus möglich
- I2C nutzte ursprünglich Adressraum von 7 Bit, was bis zu 112 Knoten auf einem Bus erlaubt (16 der 128 möglichen Adressen für Sonderzwecke reserviert)
- Adresse z.B. bei RTC DS1307: 0x68



## I2C Bus

- Wegen Adressknappheit wurde eine 10-Bit-Adressierung eingeführt - sie ist abwärtskompatibel zum 7-Bit-Standard durch Nutzung von 4 der 16 reservierten Adressen - beide Adressierungsarten sind gleichzeitig verwendbar, so sind bis zu 1136 Knoten auf einem Bus möglich
- Eine Standard-I2C-Adresse ist das erste vom Master gesendete Byte, wobei die ersten sieben Bit die eigentliche Adresse darstellen und das achte Bit (R/W-Bit) dem Slave mitteilt, ob er Daten vom Master empfangen soll (LOW) oder Daten an den Master zu übertragen hat (HIGH)



## I2C Bus

- Während des Betriebes können Teilnehmer zum Bus hinzugefügt oder entfernt werden (Hot-Plugging)
- I2C wird in VESA Monitordaten-Interface (Display Data Channel, kurz DDC) verwendet
- Der von Intel für die Kommunikation von Mainboard-Komponenten definierte SMBus ist dem I2C Bus sehr ähnlich, die meisten ICs erlauben einen Betrieb an beiden Bussen
- Die bis Ende 2014 verwendete Krankenversichertenkarte war eine I2C-Karte - unter den goldenen Kontaktflächen der Chipkarte befand sich ein einfaches I2C-EEPROM
- NXP Hersteller Datenblatt: <https://is.gd/mikowi>



## I2C Scanner

- Beispielcode: Arduino I2C Scanner
- Vorlage: <https://is.gd/ocisiv>
- Alle I2C Busteilnehmer werden angezeigt

A screenshot of a terminal window titled "/dev/ttyUSB3". The window displays the output of an I2C scan. The text shows the process starting with "Scanning...", then finding two devices at addresses 0x50 and 0x68, and finally concluding with "done".

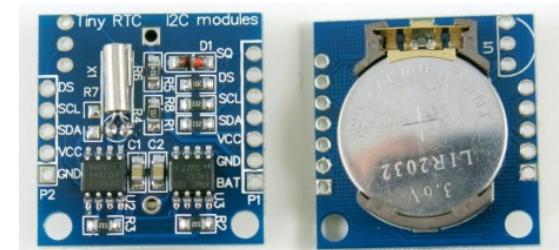
```
Scanning...
I2C device found at address 0x50 !
I2C device found at address 0x68 !
done
```

At the bottom of the window, there are several status indicators and settings: a checked "Autoscroll" checkbox, a "Both NL & CR" dropdown menu, and a "9600 baud" dropdown menu.



## Real-Time-Clock DS1307 (I2C)

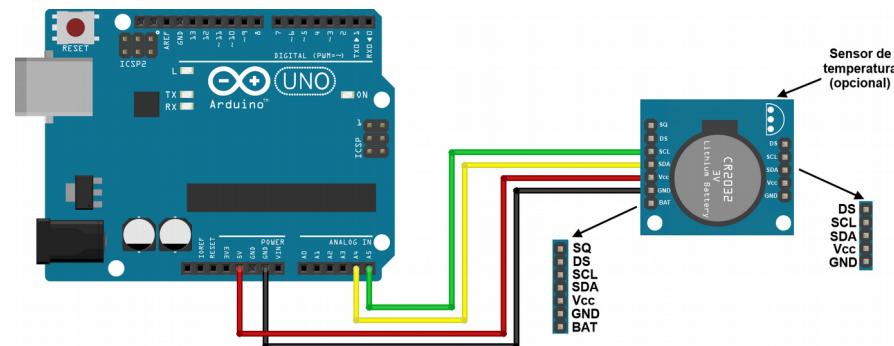
- Batteriegepuffert: LIR2032
- Abweichung / Genauigkeit:  $\pm 1,5$  Sek./Tag
- Quarz: 32,768kHz
- SRAM: 56 Bytes nichtflüchtiger SRAM
- Verbrauch: 500nA bei laufendem Quarz
- Verzicht auf Pull-Up Widerstände möglich





## Real-Time-Clock DS1307 (I2C)

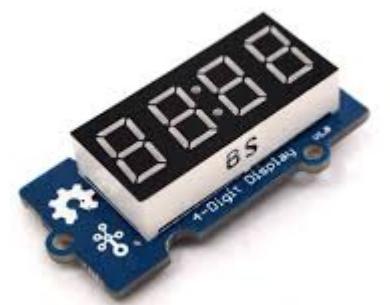
- Beispielcode: Stellen und Lesen der Uhrzeit
- Datenblatt: <https://is.gd/cenonu>
- Vorlage: <https://is.gd/axogir>
- Library: <https://is.gd/opojov>





## 4x 7-Segment Anzeige TM1637

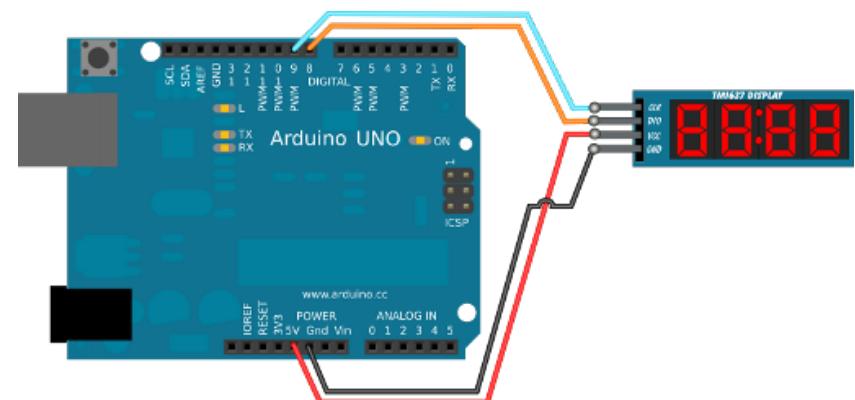
- Reduzierter I2C Bus:  
keine Busteilnehmeradresse
- Verzicht auf Pull-Up  
Widerstände möglich





## 4x 7-Segment Anzeige TM1637

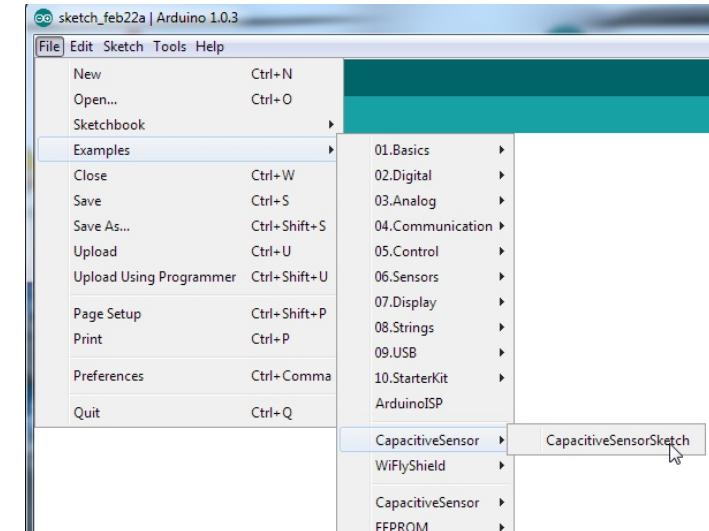
- Beispielcode: Zahl 2017 anzeigen
  - Datenblatt: <https://is.gd/nenine>
  - Vorlage: <https://is.gd/zepubo>
  - Library: <https://is.gd/dalome>
- Quellcode: <https://is.gd/ukunux>





# Arduino Libraries

- Entpacken in arduino/libraries Unterordner:  
<https://github.com/jcw/rtclib>  
<https://github.com/avishorp/TM1637>
- Code Beispiele erscheinen in  
Arduino IDE / Datei / Beispiele

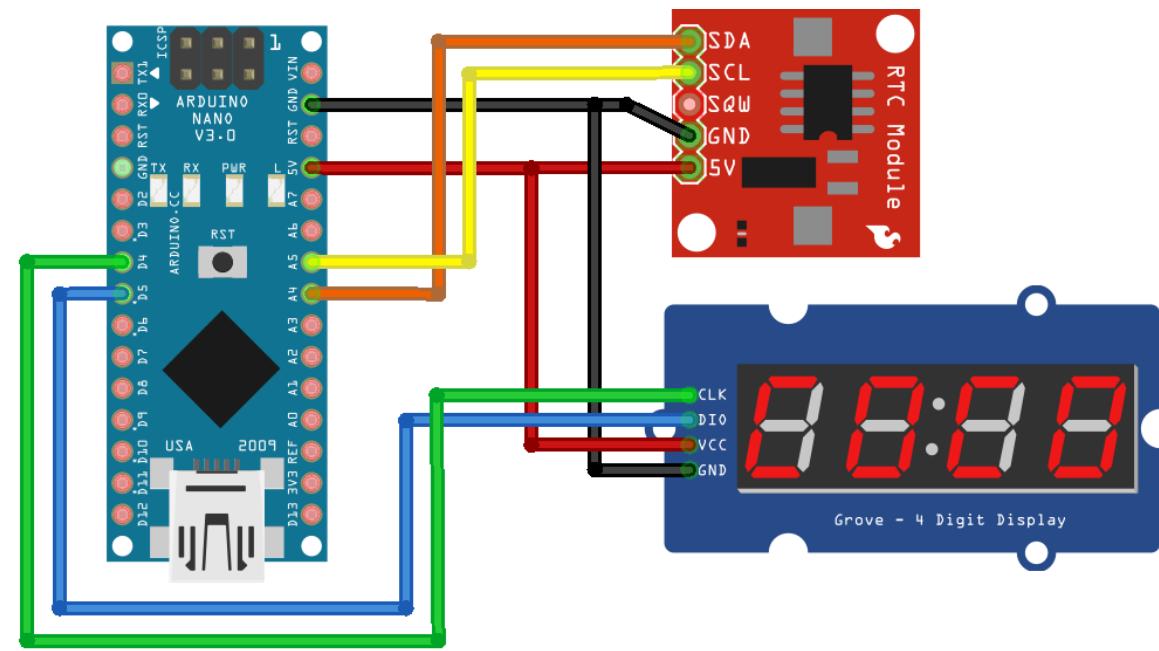




## Arduino Nano Uhr

- TM1637 frei belegbar  
(Pin D4 CLK, D5 DIO)
- RTC DS1307 fest I2C  
(Pin A4 SDA, A5 SCL)

Quellcode:  
<https://is.gd/bihivi>





## Arduino Nano Uhr

- Ziel: Zeitanzeige mit Doppelpunkt, sekündlich blinkend
- Filmzitat BTTF1: „Zeit? Ich habe genug Zeit, ich habe eine Zeitmaschine ...“





**Vielen Dank für eure  
Aufmerksamkeit !**

Noch Fragen?

