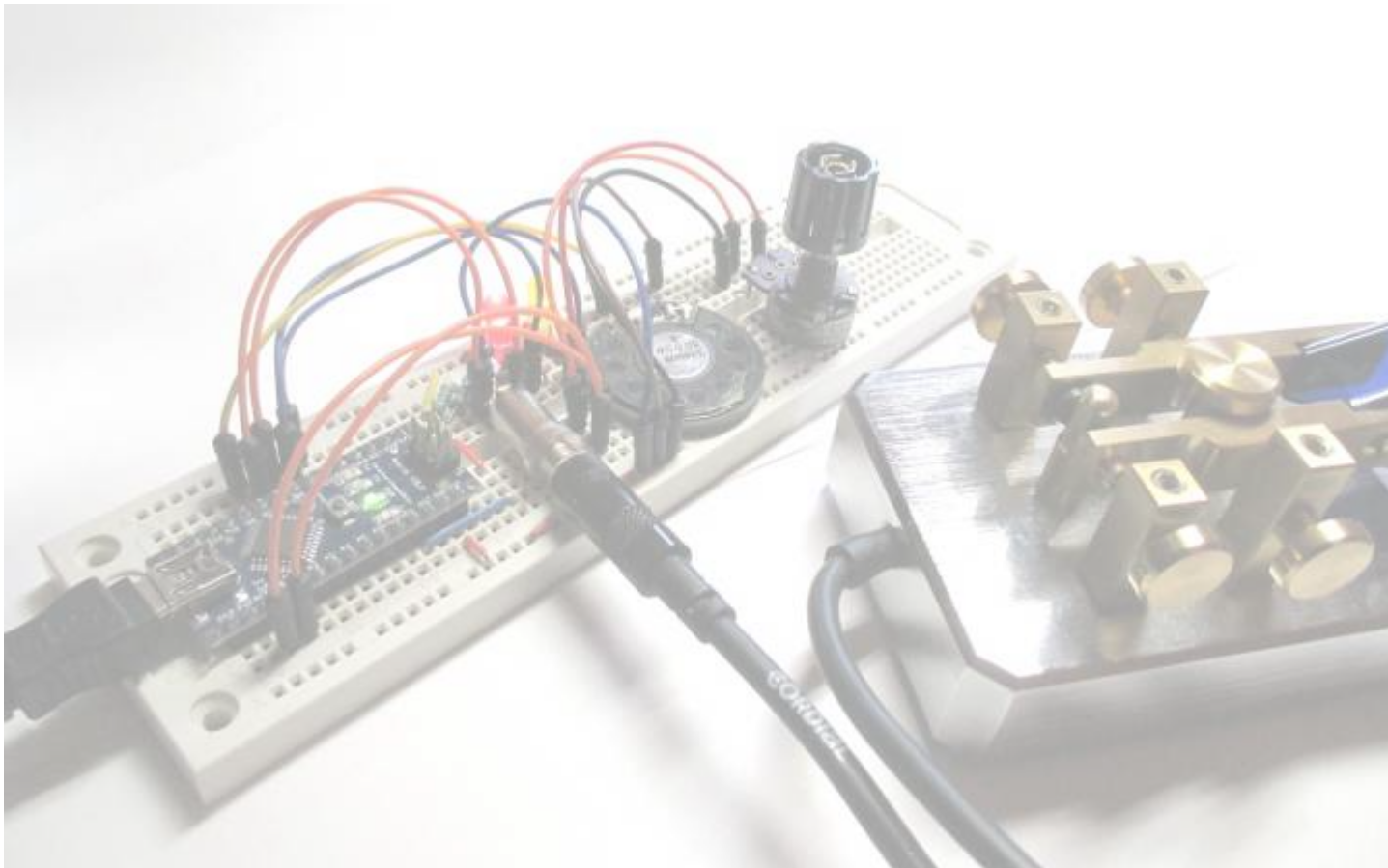
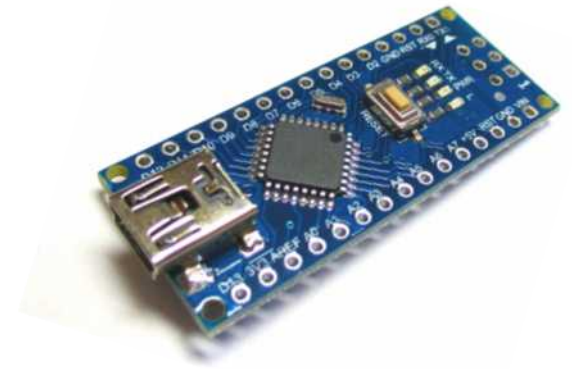
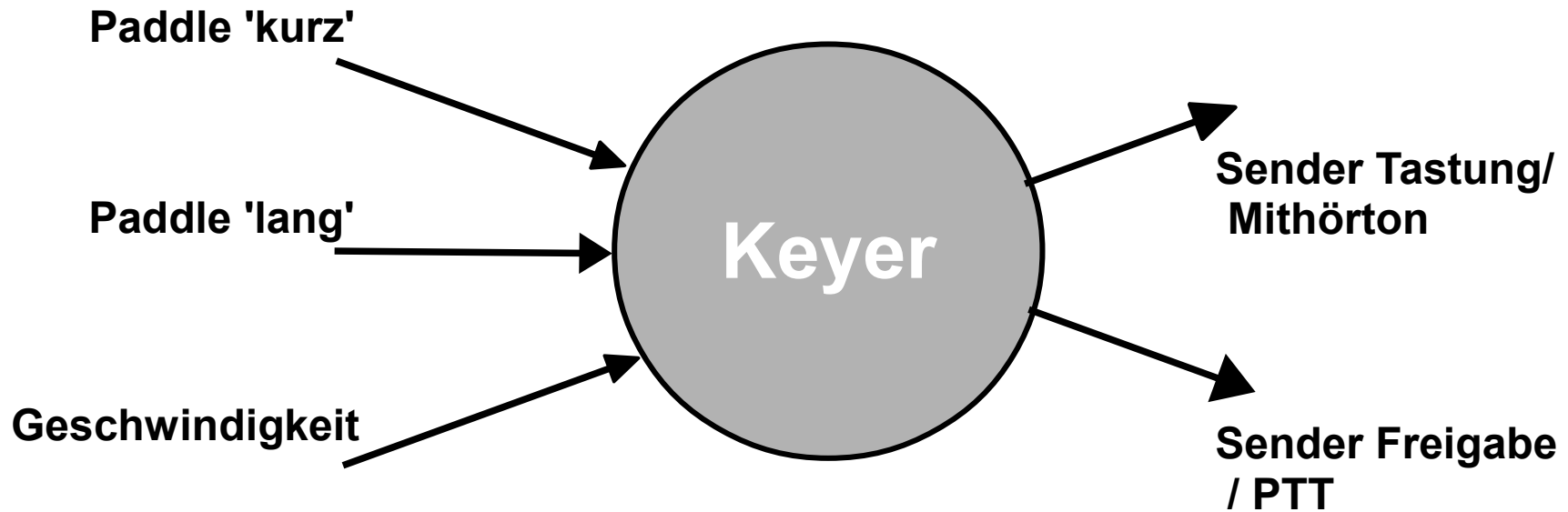
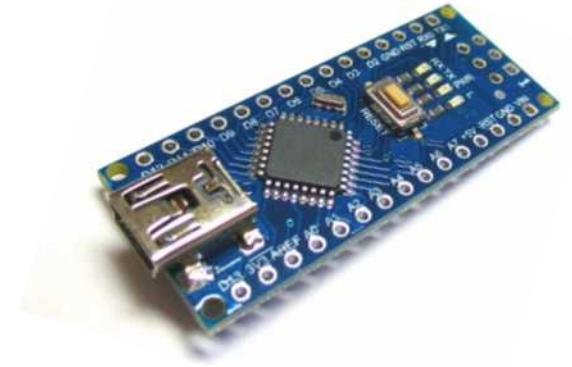


Elektronische Morsetaste mit ARDUINO

Eine Projektstudie



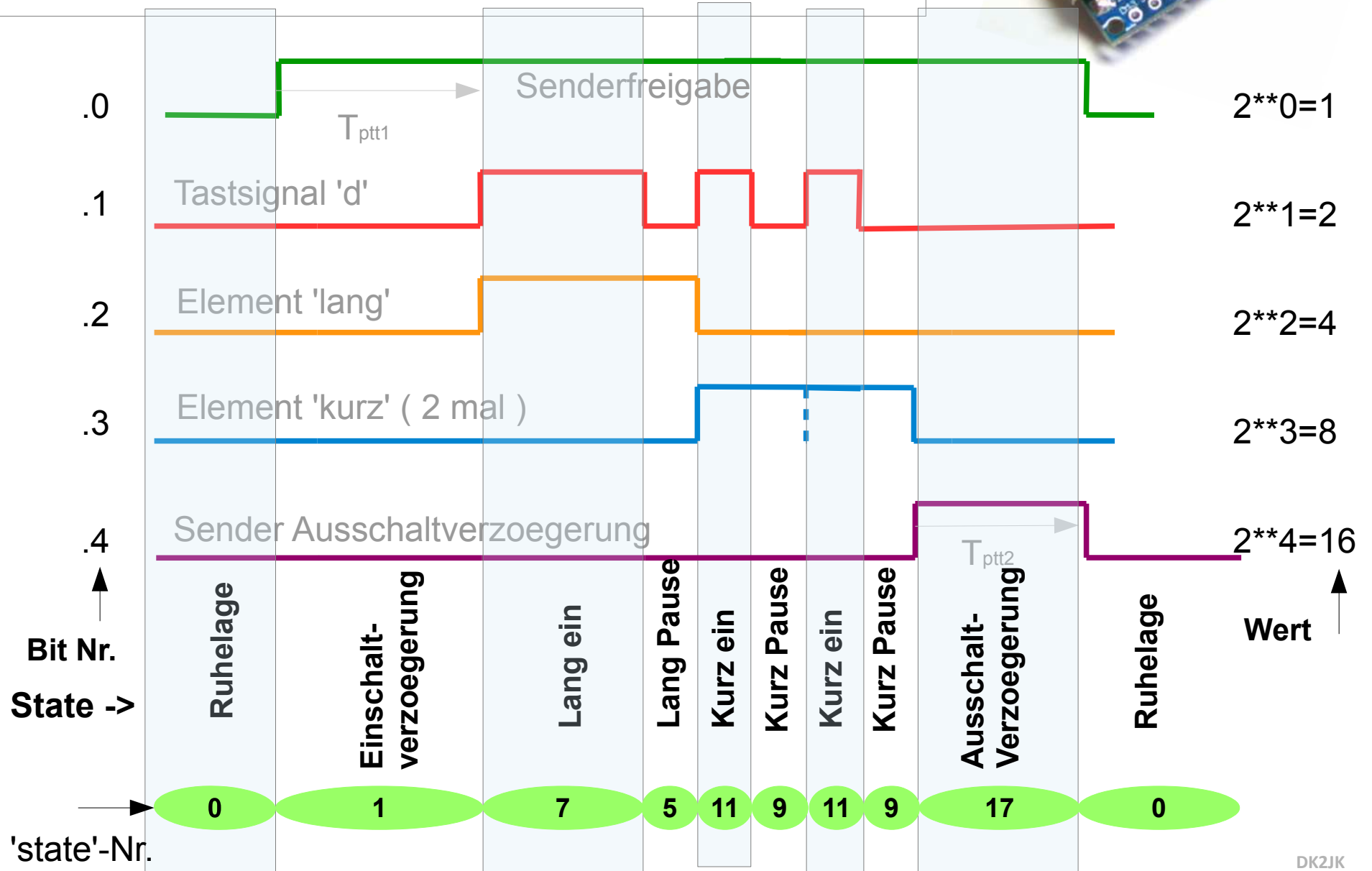
Übersicht



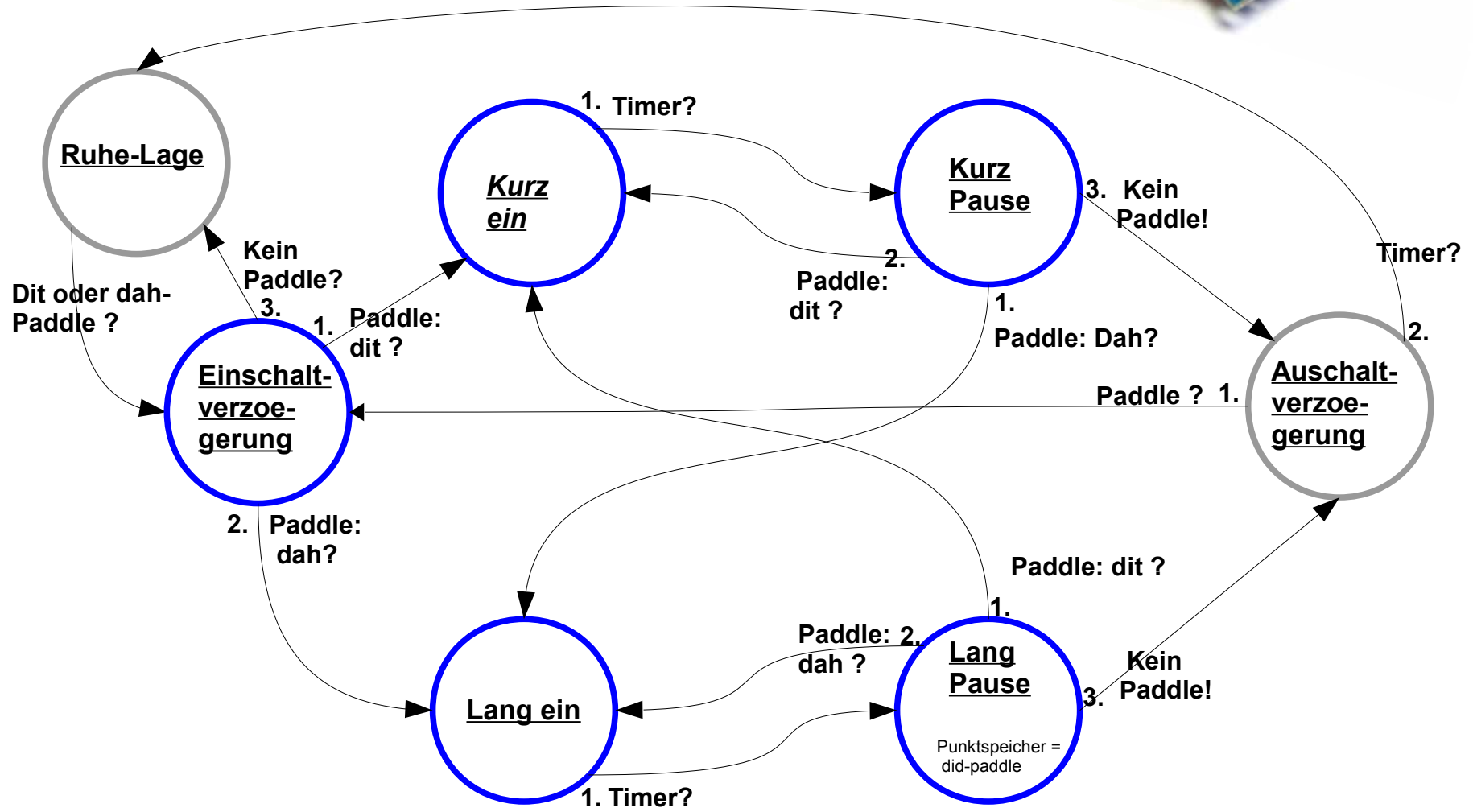
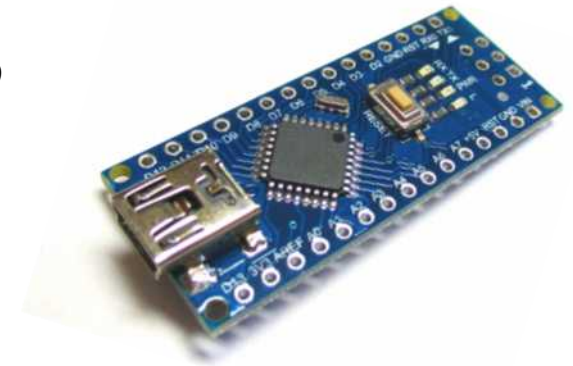
Elektronische Morsetaste mit ARDUINO



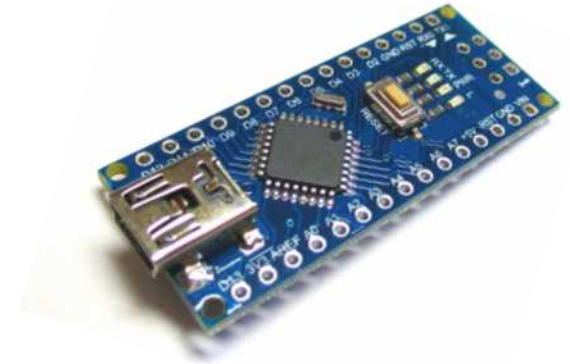
Analyse



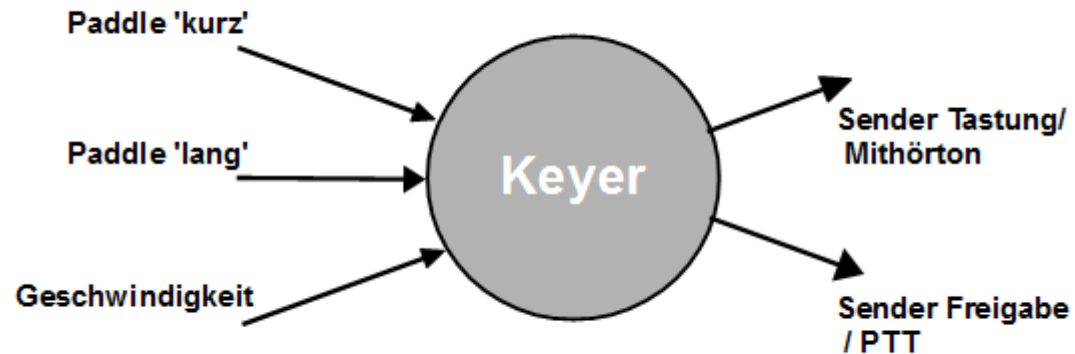
Zustandsdiagramm



Elektronische Morsetaste mit ARDUINO



Programm (1)



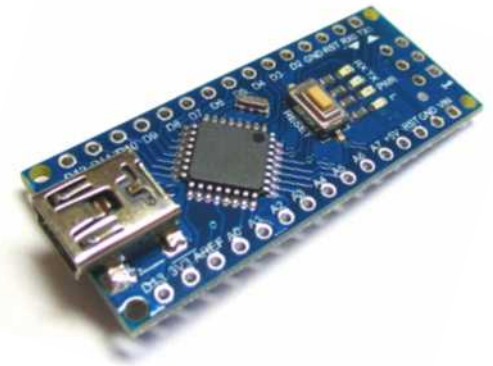
Keyer.h (Headerdatei)

Entwurf

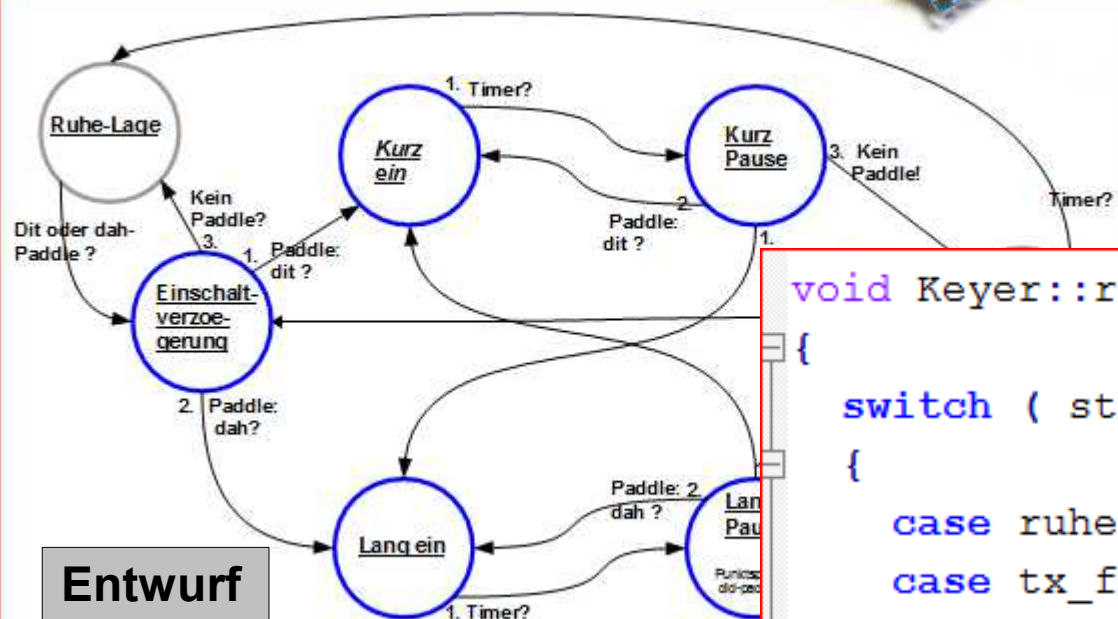
```
inline static bool dit_paddle() ;  
inline static bool dah_paddle();
```

```
public:  
    Keyer( unsigned char wpm = 18,  
           unsigned char delay_start = 50,  
           unsigned char delay_end = 500);  
  
    void run();  
    void update_wpm(unsigned char wpm);  
    bool txEin();  
    bool txFreigabe();
```

Elektronische Morsetaste mit ARDUINO



Programm (2)



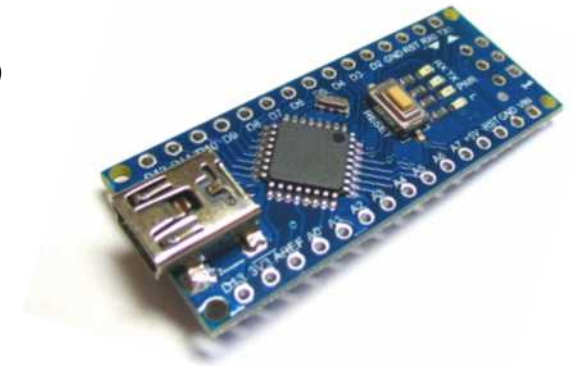
Entwurf

```
void Keyer::run()
{
    switch ( state)
    {
        case ruhelage:           {...}      break;
        case tx_freigabe:        {...}      break;
        case dit_ein:            {...}      break;
        case dit_zwischenraum:   {...}      break;
        case dah_ein:            {...}      break;
        case dah_zwischenraum:   {...}      break;
        case tx_delay:           {...}      break;
    }
}
```

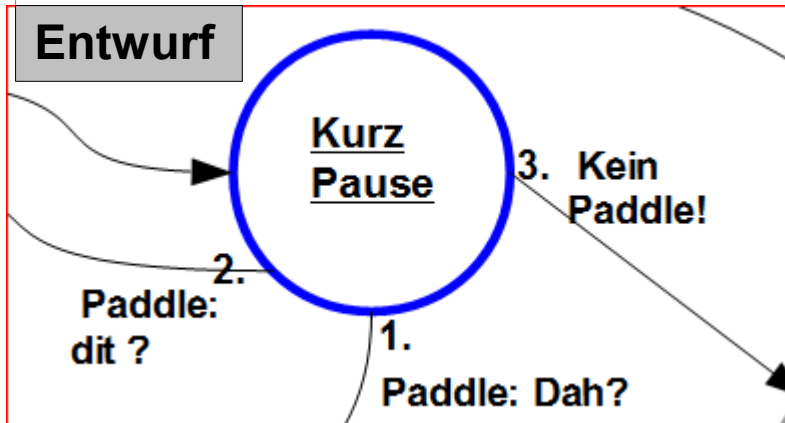
Function Keyer.run()

Programm (3)

Elektronische Morsetaste mit ARDUINO



Entwurf

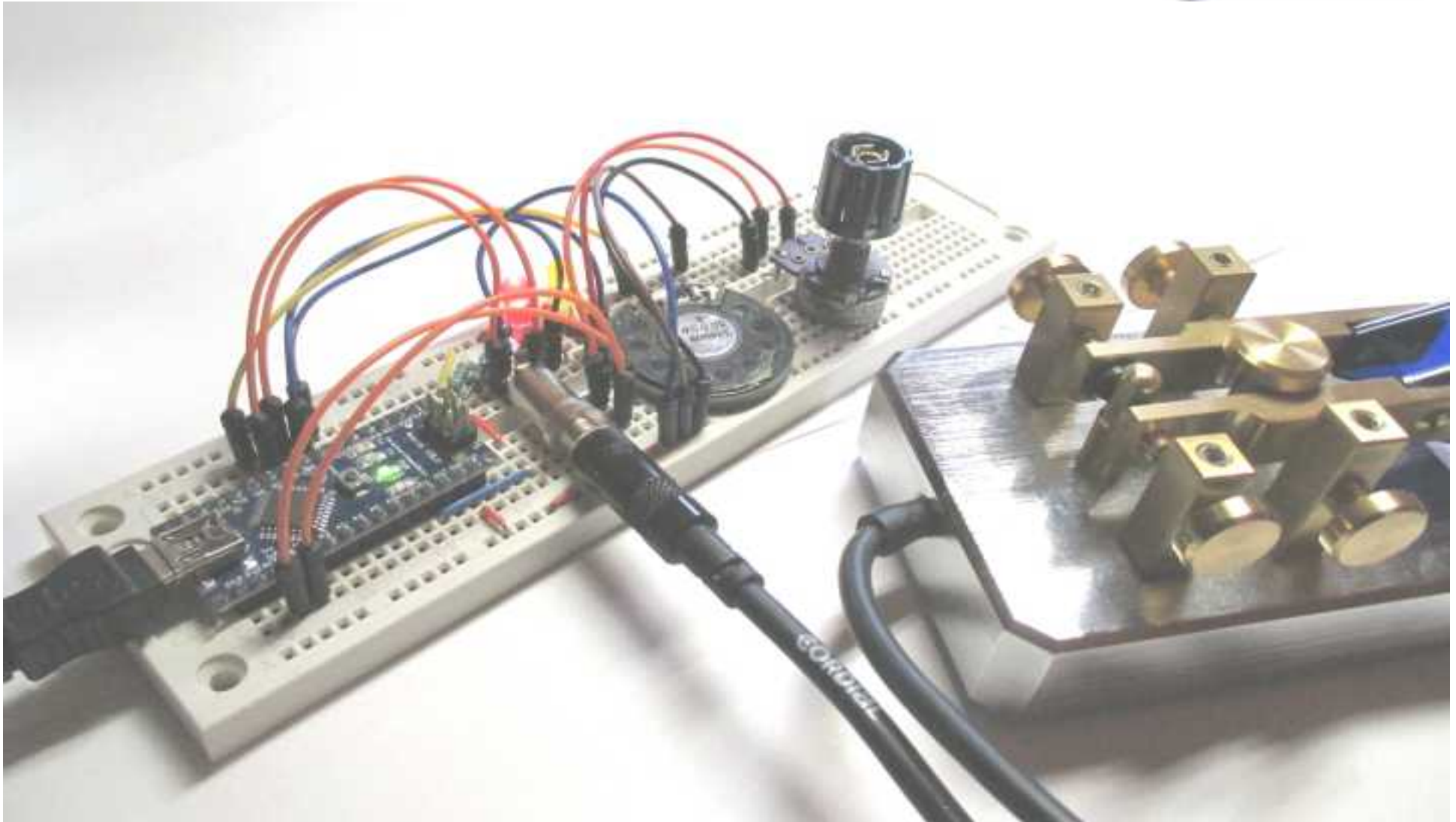
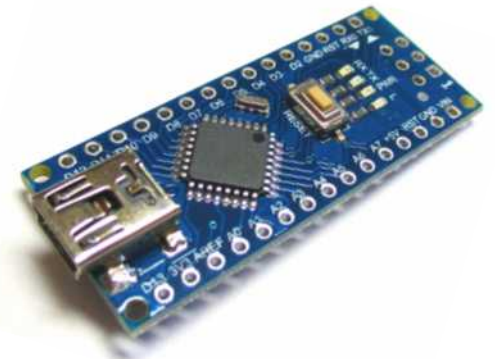


Programm Fragment

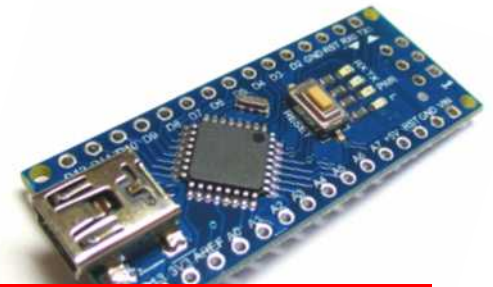
```
case dit_zwischenraum:
  /* sender ist aus fuer eine zeiteinheit
   * dadurch, dass jetzt dah-taste zuerst abgefragt wird,
   * kommen beim druecken beider tasten wechselnd dits und dahs heraus
   */
  if ( timer.event() )
  {
    if (dah_paddle()) {          // ereignis 1: dah-taste gedrueckt
      timer.interval( 3 * t_dit );
      state = dah_ein;
    }
    else if (dit_paddle()) { // ereignis 2: dit-taste gedrueckt
      timer.interval(t_dit);
      state = dit_ein;
    }
    else {                      // ereignis 3: keine taste gedrueckt
      timer.interval(t_tx_delay_end);
      state = tx_delay;
    }
  }
  break;
```


Testaufbau

Elektronische Morsetaste mit ARDUINO



Elektronische Morsetaste mit ARDUINO



Hauptprogramm

Hauptprogramm:

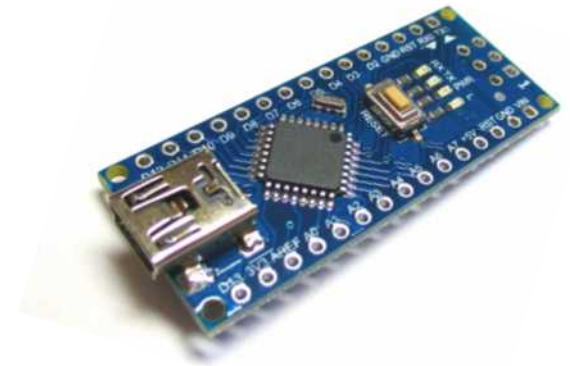
```
void loop() {  
  /////////////// keyer ///////////////  
  keyer.run(); // keyer aktionen  
  digitalWrite(PIN_TX_ENABLE, keyer.txFreigabe()); // tx freigabe schalten  
  digitalWrite(PIN_TX_KEY, keyer.txEin()); // tx schalten  
  if( keyer.txEin() ) { tone(PIN_SIDETONE, 650) ; }  
  else { noTone(PIN_SIDETONE); } // mithoerton  
}
```

Verstellung der
Geschwindigkeit:

```
int x = analogRead( A0);  
int wpm = map ( x, 0,1024, 10,30);  
keyer.update( wpm);
```

Option Decoder:

```
// decoder auf serielle schnittstelle  
decoder.decode(keyer.txEin()); // decoder arbeiten lassen  
if ( decoder.available() ) // hat er was erkannt ?  
{ char x = decoder.read(); // dann lesen  
  Serial.print(x); // und auf UART ausgeben  
}
```



Referenzen

- [1] <http://www.qsl.net/dk5ke/squeezeen.html>
„Die vollautomatische Squeeze-Taste“
- [2] Ted Theroux, N9BQ : „A Digital CMOS IAMBIC Keyer“ Jun 82 QST
(Hier wird ein Hardware- State- Counter verwendet)
- [3] <http://dk2jk.darc.de/arduino/keyer/oldies/>
„CMOS Keyer 1982“
- [4] <http://m0xpd.blogspot.co.uk/2013/02/keyerduino.html>
„KEYERduino“
- [5] <http://m0xpd.blogspot.co.uk/p/occams-microcontroller.html>
„Occams Microcontroller“ SPRAT 156, Autumn 2013
- [6] <http://www.rkrdesignsllc.com/products/transceivers-receivers/506-rebel-open-source-qrp-transceiver/>
„506 Rebel - Open Source Arduino-based CW QRP Transceiver“
- [7] <http://dk2jk.darc.de/arduino/keyer/>
„Elektronische Morsetaste mit ARDUINO“ (dieses Dokument)