

Wie benutzt mal das Nextion Display ?

Ich habe mich jetzt nach vielen Jahren ohne dieses Display wieder mal durch diese vielen, leider offenbar schlecht dokumentierten Möglichkeiten gekämpft, habe viele Menschen in Arduino Foren befragt und damit ein Programm gebaut, das wunderbar das tut, was ich brauchen. Und ich möchte jetzt hier die verschiedenen Ansätze, die ich mir für mein Programm herausgesucht habe, hier ein wenig Dokumentieren.

Hier gleich mal zu Beginn meine benutzten Deklarationen.

```
HardwareSerial Nextion(2);
byte Next[3] = { 255, 255, 255 }; // global deklarieren
/*
Wird dann so benutzt:
Nextion.write(Next, 3);
*/
```

Und im Setup:

```
void setup() {
  Serial.begin(115200);
  delay(500);
  Nextion.begin(9600, SERIAL_8N1, 16, 17);
```

16, 17 sind die TX/RX Pins, die eigentlich beim ESP32 Hardwaremässig vorgegeben sind. Trotzdem muss ich diese „HardwareSerial“ Lib benutzen. Die Pins 16, 17 sind bei meinem ESP32 als Serial2 vorgegeben.

```
Nextion.print("dim=50"); // Display wird etwas gedimmt
Nextion.write(Next, 3);
```

Hier oben wird das Display etwas dunkler gestellt. Mit „dim=100“ wird es wiederauf volle Helligkeit gestellt. Ein User im ArduinoCC Forum hat sich beklagt, dass er Probleme hatte, wenn er das Display länger mit „dim=0“ ausgestellt hatte, und es dann, wenn er die Helligkeit wieder hochfuhr, zu flackern begann, was er nicht mehr weg bekam. Also ich bin der Meinung, weil ich es nicht unbedingt auch erleben will, dunkle ich es nur bis auf 10% ab, mache es also nicht völlig dunkel.

```
Nextion.print("Startseite.Meldungen.txt=\"FRAM Speicher ist voll !!\");
Nextion.write(Next, 3);
```

Hier oben wird eine Textmeldung in ein Textfeld ausgegeben. Das Textfeld befindet sich in dem Fall auf der Seite „Startseite“ und heißt „Meldungen“. Das „.txt“ steht für Textfeld. Das Textfeld muss im Nextion Editor `vscode -> global` eingestellt sein. Auch die Seite, auf der es sich befindet habe ich `vscode -> global` eingestellt. Global steht für, von Aussen Erreichbar.

```

Nextion.print("page 0");
Nextion.write(NexT, 3);

Nextion.print("Startseite.Meldungen.txt=\"Messungen wurden
gestartet\"");
Nextion.write(NexT, 3);

Nextion.print("Startseite.MessungStart.txt=\"Messungen
Stopp\"");
Nextion.write(NexT, 3);

```

Hier oben haben wir Drei Anweisungen. Die erste ist „page 0“. Jede dieser drei Anweisungen wird mit dem (NexT, 3) abgeschlossen. Das ist oben in der Deklaration zusammengestellt und besteht aus 3x ff. Ohne diese 3x ff wird keine Anweisung ausgeführt !!

Die zweite Anweisung hier, sendet den Text „Messungen wurden gestartet“ am das Textfeld „Meldungen“ das sich auf der „Startseite“ befindet.

Die dritte Anweisung hier, sendet den Text „Messungen Stopp“ am das Button Textfeld „MessungenStart“ das sich auf der „Startseite“ befindet. Damit ändere ich die Beschriftung des Buttons, die zu Beginn „Messungen Start“ heißt. Wenn ich ihn drücke, ist die Messung gestartet und um die Messungen wieder zu stoppen, muss man wieder auf den Button drücken, der dann ja zu „Messungen Stopp“ umgetauft wurde. So kann man klar zeigen, dass der Button mehrfach Funktionen ausführt.

```

Nextion.printf("Startseite.Datum.txt=\"");
Nextion.print(&local, "%d.%m.%y");
Nextion.write('');
Nextion.write(NexT, 3);
//
Nextion.printf("Startseite.Zeit.txt=\"");
Nextion.print(&local, "%H:%M");
Nextion.write('');
Nextion.write(NexT, 3);
//
Nextion.printf("Startseite.Hysterese.val=");
Nextion.print(Hysterese);
Nextion.write(NexT, 3);
}

```

Hier gebe ich auf die Textfelder für „Datum“ und für „Zeit“, die entsprechenden Text - Daten zur Datums und Zeitanzeige aus.

Und im dritten Teil gebe ich auf ein Numerisches Feld das „Hysterese“ heißt und sich auf der „Startseite“ befindet, den Numerischen Wert für Hysterese im Millivolt aus.

```

Nextion.print("Startseite.LGM.txt=\");
Nextion.print(f, 2);
Nextion.print(F(" V"));
Nextion.print(&tm, " am %d.%m.%y");
Nextion.print(&tm, " um %H:%M:%S");
Nextion.print(" # ");
Nextion.print(Zaehler);
Nextion.write('');
Nextion.write(NexT, 3);

```

Hier werden mehrere Texte in einem langen Textfeld, das ich „LGM“ genannt habe, aneinander gehängt. Den Abschluss bildet „(““““ .

Das ist eigentlich wie das Ergebnis von diesem Serial.print() hier unten:

```

Serial.print(f, 4);
Serial.print(F(" Volt ->"));
Serial.print(&tm, " Datum: %d.%m.%y");
Serial.print(&tm, " Zeit: %H:%M:%S");
Serial.print(F(" ADR = "));
Serial.print(Speicher);
Serial.print(F(" # "));
Serial.println(Zaehler);

```

Nur ohne die Speicher Adresse.

```

Nextion.print("Startseite.Byte.val=");
Nextion.print(Speicher);
Nextion.write(NexT, 3);
//
Nextion.printf("Startseite.Volt2.txt=\"%5.3f\"", ads_mv0);
Nextion.write(NexT, 3);

```

Hier ist ein Numerisches Fenster auf der „Startseite“ das „Byte“ heißt. Und die Zahl die in „Speicher“ abgelegt ist, anzeigt.

Das zweite Fenster ist auf der „Startseite“, es heißt „Volt2“ und zeigt eine Fließkomma Zahl an, der als float formatierter „ads_mv0“ Wert abgelegt ist. Die 3 von „\“%5,3f\“ bestimmt die Darstellung der Kommastellen. Die 5 ist wahrscheinlich die gesammte länge der Zahl, habe ich aber nicht getestet. Das Ergebnis bei dem hier gezeigten war z.B. 32.123

```

    Nextion.printf("PasswortAW.t0.txt=\"\"); // Passwort
löschen
    Nextion.write(NexT, 3);
//
    Nextion.print("page 0");
    Nextion.write(NexT, 3);
//
    Nextion.print("Startseite.Meldungen.txt=\"FRAM wird
geloescht !!\");
    Nextion.write(NexT, 3);

```

Hier wurde im Nextion ein Passwort eingegeben und an den ESP32 ausgegeben. Dass es bei der nächsten PW Eingabe nicht mehr zu sehen ist, lösche ich es, wenn es das richtige war, dass es bei der nächsten PW Abfrage nicht schon im Textfeld steht. Dazu mache ich, indem ich auf der Seite „PasswortAW“ in das Textfenster „t0“, einen leeren Text schicke. Dann schalte ich mit Hilfe von „page 0“ auf die Startseite um und auf die „Startseite“ im Textfenster „Meldungen“ schicke ich die Bestätigung, „Fram wird gelöscht“ wenn das PW richtig war. Das Löschen führe ich natürlich auch aus.

```

    Nextion.print("Startseite.Meldungen.txt=\"FRAM ist
geloescht\");
    Nextion.write(NexT, 3);

```

Wenn das FRAM löschen erledigt ist, was ein paar Sekunden dauert, schicke ich auf die „Startseite“ in das Fenster „Meldungen“ den Text „Fram ist gelöscht“.

Hier hole ich mir die Meldungen von Nextion. Das sieht erst mal so aus:

```
void NEXTzuESP() {
    static uint8_t bIndex;
    static uint8_t buffer[bufferSize];
    if (Nextion.available()) {
        if (Abfragezaehler == 0) {
            delay(300);
        }
        uint8_t inChar = Nextion.read();
        buffer[bIndex++] = inChar;
        if ((buffer[0] == 101) || (buffer[0] == 80) || (buffer[0]
== 35) || (buffer[0] == 26) || (buffer[0] == 45) ||
(buffer[0] == 43)) {
            Abfragezaehler = 0;
        }

        Serial.print(Abfragezaehler);
        Serial.print(" Buffer = ");
        Serial.println(buffer[0]);
    }
}
```

`if (Nextion.available())` fragt die Serial Schnittstelle von Nextion Display ab, ob uns das Display was zu sagen hat.

Mit diesen `if ((buffer[0] == 101) || (buffer[0] == 80) || u.s.w.` schaue ich, ob da was ist, was mit interessiert. Wenn ja lösche ich meinen Zähler, dass das erste Byte mit der Zählernummer „0“ beginnt.

Die `Serial.print()` zeigen mir nur zur Kontrolle werend ich mein Programm erstelle, was da so kommt. Das kann dann z.B. so aussehen:

```
0 Buffer = 101
1 Buffer = 0
2 Buffer = 2
3 Buffer = 0
4 Buffer = 255
5 Buffer = 255
6 Buffer = 255
7 Buffer = 49
Messungen Stoppen
0 Buffer = 35
```

Die Erklärung dazu:

Das Byte 0 sagt mit dem „101“ jetzt zum Beispiel, dass da ein Tastenevent ausgelöst wurde.

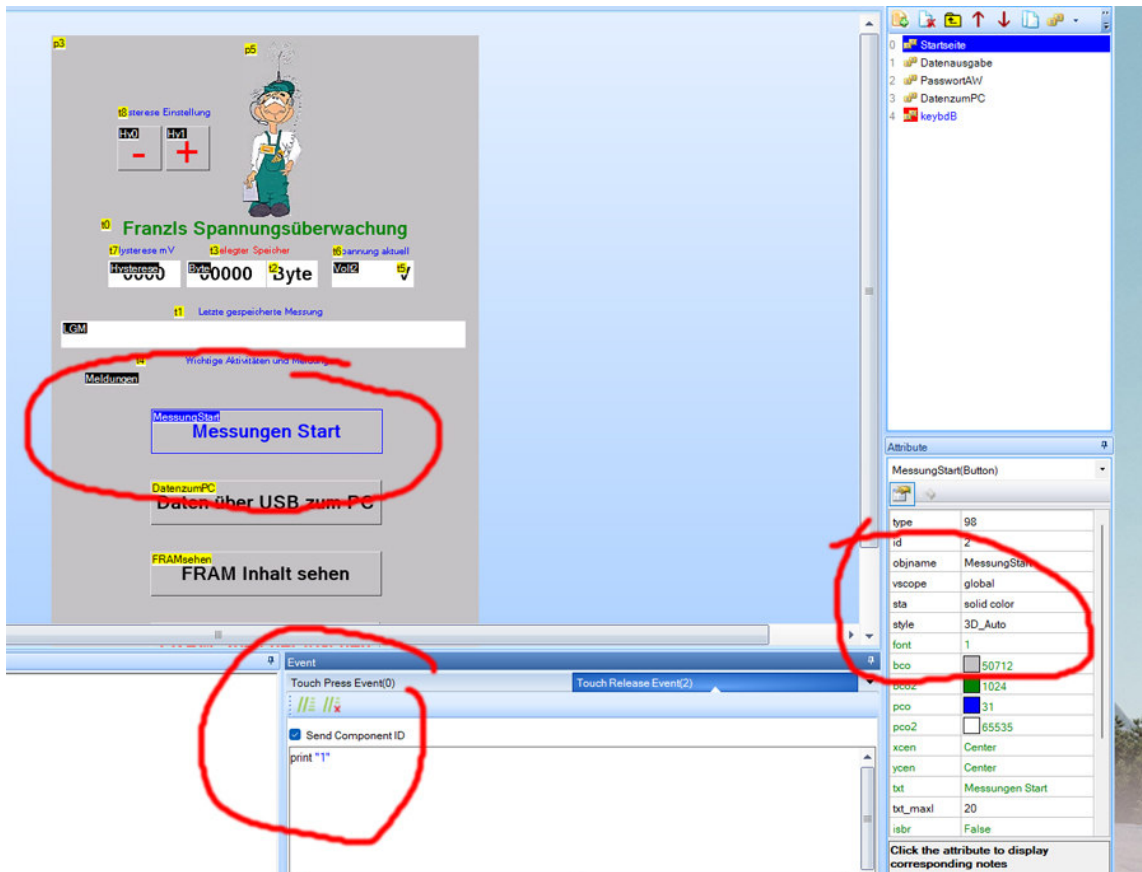
Das Byte 1 sagt, dass es auf der Seite 0, das ist die Startseite, ausgelöst wurde.

Das byte 2 sagt, dass es der Button id2 ist, der gedrückt wurde.

Das byte 3 sagt, dass der Button wieder losgelassen wurde, und damit den Event auslöste.

Die bytes 4-6 bringen die Abschluss der Nachricht 3xff

Das Byte 7 sagt, dass da noch eine Nachricht ist, die ich bestimmt habe. Das ist die „49“, das steht für die „1“, was der erste Button der vier Buttons auf der Startseite ist.

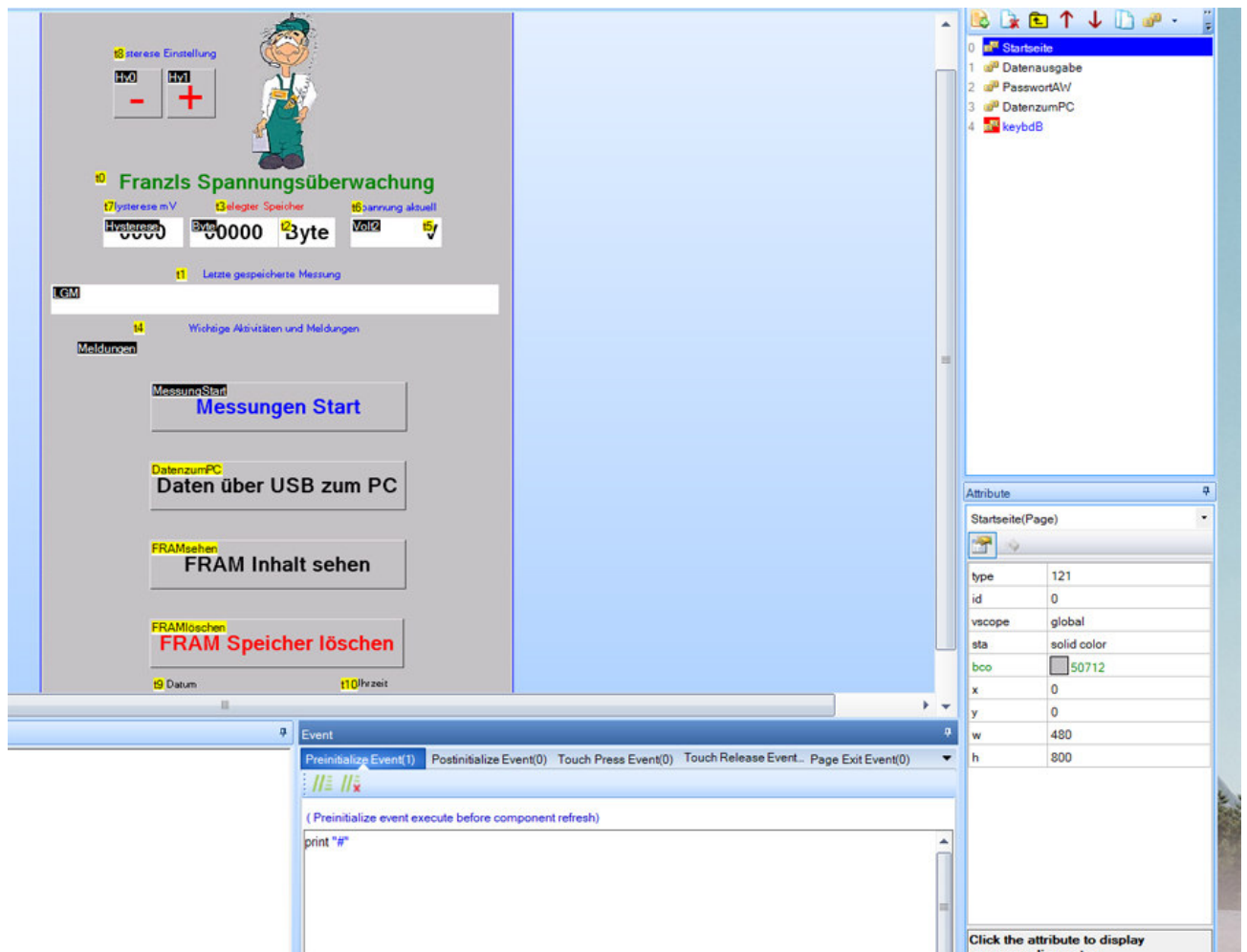


Also es wurde der Button Messungen Start/Stopp gedrückt. Dabei wird die page 0 angesprochen.

Wenn diese angesprochen wird, will ich aus Programmatischen Gründen auch eine Nachricht haben.

Das ist Bufferzähler „0“ die „35“. Das sagt mir, dass die page 0 also die Startseite neu aufgebaut wurde.

Das ist der Event unten mittig im Bild zu sehen, (print „#“) Was von meinem Programm wieder eine Reaktion fordert. u.s.w.



So ist dann die weitere Auswertung der Nachrichten im ESP32 Programm. Also die Nachrichten die mich interessieren. Ein Beispiel:

```

//#####
//##### Rückmeldung vom Nextion auswerten #####
//#####
//
void NEXTzuESP() {
  static uint8_t bIndex;
  static uint8_t buffer[bufferSize];
  if (Nextion.available()) {
    if (Abfragezaehler == 0) {
      delay(300);
    }
    uint8_t inChar = Nextion.read();
    buffer[bIndex++] = inChar;
    if ((buffer[0] == 101) || (buffer[0] == 80) || (buffer[0] == 35) || (buffer[0] == 26) || (buffer[0] == 45)
    || (buffer[0] == 43)) {
      Abfragezaehler = 0;
    }
    Serial.print(Abfragezaehler);
    Serial.print(" Buffer = ");
    Serial.println(buffer[0]);
  }
}

```

```

//----- Menüseite wurde aufgerufen -----
if ((Abfragezaehler == 0) && (buffer[0] == 35)) { // page 0 Start
  Serial.println(F("Seite Null wurde aufgerufen"));
  if (MeIO == 1) {
    Nextion.print("Startseite.MessungStart.txt=\\"Messungen Stopp\\"");
    Nextion.write(NexT, 3);
  }
}
// ----- Hysterese + Abfragen -----
if ((Abfragezaehler == 0) && (buffer[0] == 43)) { // Hysterese plus ?
  if (Hysterese < Hysteresemax) {
    Hysterese = Hysterese + 50;
    Nextion.printf("Startseite.Hysterese.val=");
    Nextion.print(Hysterese);
    Nextion.write(NexT, 3);
  }
}
// ----- Hysterese - Abfragen -----
if ((Abfragezaehler == 0) && (buffer[0] == 45)) { // Hysterese minus ?
  if (Hysterese > Hysteresemin) {
    Hysterese = Hysterese - 50;
    Nextion.printf("Startseite.Hysterese.val=");
    Nextion.print(Hysterese);
    Nextion.write(NexT, 3);
  }
}
// -----FRAM löschen Taste + PW Abfrage -----
if ((Abfragezaehler == 0) && (buffer[0] == 80)) { // P? für Passwort
  PW = PW + 1;
}
if ((Abfragezaehler == 1) && (buffer[0] == 51)) { // 3?
  PW = PW + 1;
}
if ((Abfragezaehler == 2) && (buffer[0] == 54)) { // 6?
  PW = PW + 1;
}
if ((Abfragezaehler == 3) && (buffer[0] == 57)) { // 9?
  PW = PW + 1;
}
if ((Abfragezaehler == 4) && (buffer[0] == 57)) { // 9?
  PW = PW + 1;
}
if ((Abfragezaehler == 4) && (PW == 5)) {
  Serial.println("Passwort ist korrekt");
  Nextion.printf("PasswortAW.t0.txt=\\""); // Passwort löschen
  Nextion.write(NexT, 3);
  Nextion.print("page 0");
  Nextion.write(NexT, 3);
  Nextion.print("Startseite.Meldungen.txt=\\"FRAM wird geloescht !!\\"");
  Nextion.write(NexT, 3);
  PW = 0;
  FRAM_Loeschen();
}

```

```

if ((Abfragezaehler == 4) && (PW < 5)) { // Ist es nicht die FRAM - löschne Taste
  PW = 0;
}
//-----
// ----- Daten zum PC Abfrage -----
if ((Abfragezaehler == 0) && (buffer[0] == 101)) { // Tasten Event ?
  PC = PC + 1;
}
if ((Abfragezaehler == 2) && (buffer[0] == 3)) { // Daten zum PC ?
  PC = PC + 1;
}
if ((Abfragezaehler == 7) && (buffer[0] == 50)) { // Taste 2 ?
  PC = PC + 1;
}
if ((Abfragezaehler == 7) && (PC < 3)) { // Ist es nicht die Daten zum PC Taste
  PC = 0;
}
if ((Abfragezaehler == 7) && (PC == 3)) {
  PC = 0;
  Serial.println("Daten zum PC übertragen");
  Nextion.print("page 0");
  Nextion.write(NexT, 3);
  Nextion.print("Startseite.Meldungen.txt=\"Daten wurden zum PC uebertragen\"");
  Nextion.write(NexT, 3);
  DatenzumPC();

// ----- Messungen Starten Abfrage -----
if ((Abfragezaehler == 0) && (buffer[0] == 101)) { // Tasten Event ?
  ME = ME + 1;
}
if ((Abfragezaehler == 2) && (buffer[0] == 2)) { // Taste Messungen IO ?
  ME = ME + 1;
}
if ((Abfragezaehler == 7) && (buffer[0] == 49)) { // Taste 1 ?
  ME = ME + 1;
}
if ((Abfragezaehler == 7) && (ME < 3)) { // ist es nicht die ME – Taste
  ME = 0;
}
if ((Abfragezaehler == 7) && (ME == 3) && (MeIO == 0)) {
  ME = 0;
  Serial.println("Messungen Starten");
  Nextion.print("page 0");
  Nextion.write(NexT, 3);
  Nextion.print("Startseite.Meldungen.txt=\"Messungen wurden gestartet\"");
  Nextion.write(NexT, 3);
  Nextion.print("Startseite.MessungStart.txt=\"Messungen Stopp\""); Nextion.write(NexT, 3);
  MeIO = 1;
}
if ((Abfragezaehler == 7) && (ME == 3) && (MeIO == 1)) {
  ME = 0;
  Serial.println("Messungen Stoppen");
  Nextion.print("page 0");
}

```

```

Nextion.write(NexT, 3);
Nextion.print("Startseite.Meldungen.txt=\"Messungen wurden gestoppt\");
Nextion.write(NexT, 3);
Nextion.print("Startseite.MessungStart.txt=\"Messungen Start\");
Nextion.write(NexT, 3);
MeIO = 0;
}
//----- Kommunikations Fehler ESP32 -> Nextion -----
if ((Abfragezaehler == 0) && (buffer[0] == 26)) { // Fehler ?
    KF = KF + 1;
}
if ((Abfragezaehler == 3) && (buffer[0] == 255)) {
    KF = KF + 1;
}
if ((Abfragezaehler == 3) && (KF == 2)) {
    Serial.println("Kommunikations Fehler"); // Kommunikationsfehler ESP -> Next
    KF = 0;
}
if ((Abfragezaehler == 3) && (KF < 2)) {
    KF = 0;
}
}
//-----
Abfragezaehler++; // Der Abfrage zählt immer die Bytes der einzelnen Nachrichten
bIndex = 0;
}

```