



Das Display sowie die gesamte Schaltung verbinden wir nun. Das Display sollte dabei von der eigentlichen Schaltung etwas abgesetzt werden um eine flexible Verwendung zu gewährleisten.

Das elektronische Namensschild können wir vielseitig verwenden. Wie wäre es z.B mit einem besonderen Tischnamensschild bei der nächsten betrieblichen Schulung? Oder vielleicht auch als neues Büroschild am Eingang des Büros? Als Namensschild bei einem Vortrag als Redner? Möglicherweise aber auch als Namensschild am Briefkasten? Etwa im Briefkastenschlitz hinter dem Namensschlitz:



Beim Einbau hierbei ist jedoch folgende besonderen Punkte zu achten:

1. Alle spannungsführenden oder spannungserzeugenden Teile müssen komplett wasserdicht verbaut werden. Durch Feuchtigkeit kann ansonsten unsere Schaltung zerstört werden bzw. korrodieren, was Ausfälle oder schlimmer, Kurzschlüsse zu Folge haben kann.
2. Das Display muss in der oben gezeigten Einbauweise die Öffnung und Schließung des Kastendeckels mitmachen. Dies bedeutet das hier insbesondere auf sorgfältige und geeignete Kabelverlegung und geeignete Kabelwahl zu achten ist. Das Kabel zwischen Display und Arduino muss dauerhaft die Klappenbewegungen der Postkastenklappe mitmachen. Besser ist ein Einbau an unbeweglichen Stellen wie z.B im Postkastenkorpus.
3. **Es darf an jeder Stelle nur ungefährliche Schutzkleinspannung verwendet werden.**
4. Bei einem Stromausfall erlischt auch die Anzeige. Damit trotzdem noch in diesem Fall klar ist wo die Post hin soll, empfiehlt es sich noch ein Ersatznamensschild anzubringen 😊
5. Die Stromversorgung ist unbedingt mit einer passenden Sicherung abzusichern.

Viele Punkte also, die den Einbau im Briefkasten schwierig machen und deshalb einem Elektrofachinstallateur überlassen werden sollten. Also vielleicht doch besser als Tischnamensschild verwenden 😊

Wir laden folgenden Code auf unseren Arduino hoch:

Code:

```
#include <LiquidCrystal.h>
#include <avr/sleep.h>

//Constants & Definitions
const float Warnvoltage = 4.0;
const float Sleepvoltage = 3.2;
const int Scrollspeed = 500;
const long interval = 1000; // Definition of a Second in Milliseconds

//const byte VoltTestPin = 0; // Test Analogeingang 0
const byte VoltTestPin = 2; // Produktiv Alalogeingang 2

//Real Time Clock
long previousMillis = 0; // will store last time was measured
byte seconds = 0;
byte minute = 0;
byte hour = 0;

// String Defitions
String LetterBoxOwner = " Tobias Kuch "; // Bitte in eigenen Namen ändern !
String StandardScoltext = "Hier den eigenen Text eintragen, der als Scrolltext angezeigt wird.";
//String StandardScoltext = "To bi or not to be – that is the Question !"; // Beispielttext
String BattInfo;

//Program logic
byte CharPointers[200];
float value;
int Voltage;
boolean BatteryEmpty = false;
byte Displaymode = 0;
byte oldDisplaymode = 0;
int Switchstate = 1;
byte DisplATextOneTime = 0;
byte DisplATextOneTimeWASDisp = 0;
int cmpMsgWith = 0;
int MAMsgDC = 0;

// LiquidCrystal(rs, enable, d4, d5, d6, d7);
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

void InitLcdScreen ()
{
  lcd.clear();
  lcd.setCursor(0,0);
  delay(1000);
}

void setup()
{
  for (int a=0; a <200;a++)
  {
    CharPointers[a] = a;
  }
  lcd.begin(16, 2); // start the library
  InitLcdScreen();
  analogReference(DEFAULT);
}

void NormalScrolltext ()
{
  int cmpMsgWith = StandardScoltext.length()- 2;
  Displaymode = 0;
  delay (Scrollspeed);
  lcd.setCursor(0,1);
  for (int a=0; a <17;a++)
  {
    lcd.print(StandardScoltext[CharPointers[a]]);
  }
}
```

```

    for (int a=0; a <cmpMsgWith;a++)
    {
        CharPointers[a] = CharPointers[a] + 1;
        if (CharPointers[a] > cmpMsgWith) {CharPointers[a] = 0; }
    }
}

void BatteryWarningScrolltext ()
{
    BattInfo = "- Achtung! Spannung eingebrochen auf: " + String(Voltage * 3) + " Volt. - " + StandardScrolltext + " ";
    int cmpMsgWith = BattInfo.length()- 2;
    Displaymode = 2;
    delay (Scrollspeed /2 );
    lcd.setCursor(0,1);
    for (int a=0; a <17;a++)
    {
        lcd.print(BattInfo[CharPointers[a]]);
    }

    for (int a=0; a <cmpMsgWith;a++)
    {
        CharPointers[a] = CharPointers[a] + 1;
        if (CharPointers[a] > cmpMsgWith) {CharPointers[a] = 0; }
    }
}

void Powerdown () // Toggle Powerdown, if critical Voltage is reached.
{
    lcd.setCursor(0,1);
    lcd.print("      ");
    lcd.setCursor(0,1);
    lcd.print("Sys in SleepMode ");
    set_sleep_mode(SLEEP_MODE_PWR_DOWN);
    cli();
    sleep_enable();
    sei();
    sleep_cpu();
    //restart
}

boolean BattWarning ()
{
    {
        value = analogRead(VoltTestPin); // read the input pin
        value = value * 0.00488;
        if (value <= Sleepvoltage) {
            Voltage = value;
            Powerdown();
        }
        if (value <= Warnvoltage) {
            Voltage = value;
            return true;
        }
    }
    {
        return false;
    }
}

void runrealTimeClock()
{
    // Real Time Clock & Countdown
    unsigned long currentMillis = millis();
    if(currentMillis - previousMillis > interval)
    {
        previousMillis = currentMillis;
    }
    // Countdown
    // Countup 24 St Hr.
    seconds = seconds+1;
    if (seconds > 59)
    {
        seconds = 0;
        minute=minute+1;
        BatteryEmpty = BattWarning ();
    }
}

```

```

    if (minute > 59)
    {
        minute = 0;
        if (MatC > 1) { MatC --; } //Anzeige der des Nachricht Symbols
        hour = hour + 1;
    }
    if (hour > 23)
    {
        hour = 0;
    }
}
}
}

void ChangeDisplayMode()
{
    if ( Displaymode > oldDisplaymode )
    {
        oldDisplaymode = Displaymode;
        for (int a=0; a <200;a++)
        {
            CharPointers[a] = a;
        }
    }
    if ( Displaymode < oldDisplaymode )
    {
        oldDisplaymode = Displaymode;
        for (int a=0; a <200;a++)
        {
            CharPointers[a] = a;
        }
    }
}

void loop()
{
    runrealTimeClock();
    if (!BatteryEmpty)
    {
        NormalScrolltext ();
    }
    if (BatteryEmpty)
    {
        BatteryWarningScrolltext ();
    }
    ChangeDisplayMode();
}

```

Nach Hochladen des Codes fängt unser Display direkt an, in der ersten Zeile den Namen zu zeigen und in der zweiten den vorgegebenen Scroll Text der im Code in der Zeile

```
String StandardScrolltext = "Hier den eigenen Text eintragen, der als Scrolltext angezeigt wird.";
```

definiert wurde, anzuzeigen. Zusätzlich dazu wird die Versorgungsspannung über den Pin A2 überwacht und bei unterschreiten eines bestimmten Spannungslevels die Warnung: „Achtung! Spannung eingebrochen auf: x Volt.“ im Display als Scrolltext Erweiterung ausgegeben. Dies soll in einem möglichen Szenario mit Batteriebetrieb auf eine schwache Batterie hinweisen. Wenn sich die Batterie dann noch weiter entlädt, geht der Prozessor in diesem Fall dauerhaft in einen Schlafmodus und die Meldung: „Sys in SleepMode“ wird angezeigt. Da das Display aber noch aktiv ist, entlädt sich die Batterie aber dennoch weiterhin. Dies ist also kein Schutz vor einer Tiefentladung, sondern nur ein Hinweis

Falls die Schaltung ohne Batterie betrieben wird, kann der Code Teil ohne weitere Beeinträchtigung der Funktion auch entfernt werden.

Viel Spaß beim Nachbauen uns bis zum nächsten Mal.