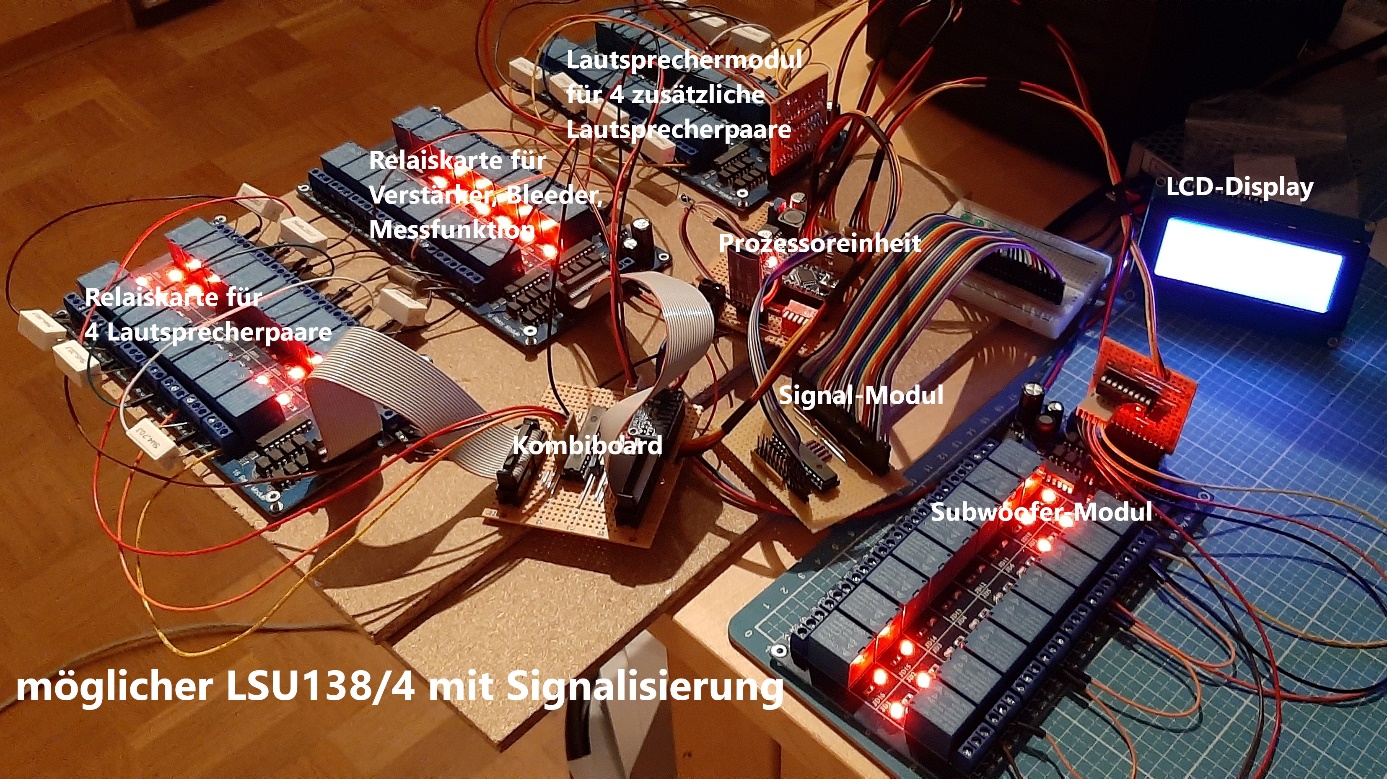
Ein Hallo in die LSU-Fangemeinde ;-)

Wie im Bericht zum LSU026 von Rundmacher bereits angekündigt möchte ich in lockerer Folge die verschiedenen Module zum LSU vorstellen. Hierbei werde ich auch den Schaltplan sowie ein mögliches Platinen Layout für den Nachbau zeigen und auf die Funktion der einzelnen Bauteile in ihrem Kontext eingehen.

Wer den Thread bisher verfolgt hat, wird festgestellt haben, dass sich die Entwicklung des LSU zwischen mir und Rundmacher etwas unterscheidet. Wie man am LSU022 und LSU026 von Rundmacher sieht macht er Nägel mit Köpfen, sprich den Bau eines Komplettsystems.



Ich verfolge einen leicht anderen Ansatz. Ich versuche, eigenständige Module zu entwickeln, die einfach per Kabel an das Steuerungsmodul gesteckt werden und dann automatisch vom Programm erkannt, eingebunden und konfiguriert werden. Die Module können hierbei jeweils auch in eigenen Gehäusen verbaut sein.

Bisher umgesetzt sind folgende Module:

* das Steuerungsmodul mit IR-Empfänger, LCD-Display und Bluetooth-Adapter sowie dem I2C-Bus für den Anschluss der einzelnen Erweiterungsmodule
* das Lautsprecher-Modul für 4 oder 8 Lautsprecherpaare (hier habe ich im Programm eine Deckelung für 2 Lautsprecher-Module und somit maximal 16 Lautsprecherpaare eingebaut)
* das Verstärker-Modul für 2 (3) Verstärker mit den Optionen Bleeder-Zuschaltung oder Anschluss eines 3. Verstärkers und Widerstands-Messfunktion der Lautsprecher sowie der Möglichkeit eine Quelle auf die Verstärker umzuschalten.

Mittlerweile gibt es ein Kombiboard welches ein Lautsprecher- und das Verstärker-Modul kombiniert und somit, mit dem Steuerungsmodul, einen kompletten LSU124 darstellt.

* Subwoofer-Modul für bis zu 8 passive Subwoofer (1:1 Zuschaltung zu den Lautsprecherpaaren später per Konfiguration auch freie Zuordnung auch Mehrfachzuweisung)
* Signal-Modul für die optische Signalgebung welches Lautsprecherpaar/Subwoofer gerade aktiv ist für den Maximalausbau von 16 Lautsprecherpaaren und 8 Subwoofer

In der fortgeschrittenen Entwicklung:

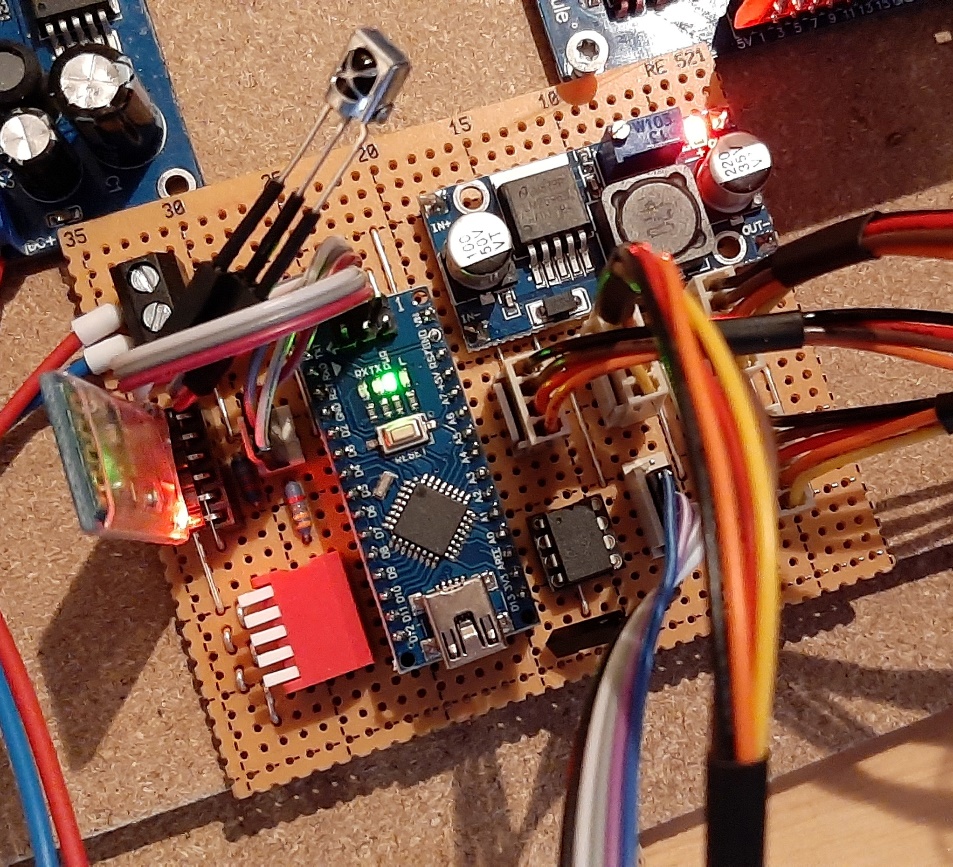
* Konfiguration und Steuerung des LSU über eine Android-App die per Bluetooth gekoppelt ist

Offene und in der Ideenfindung befindliche Module:

* Quellen-Modul für die Umschaltung von RCA, Toslink und Coax Quellen mit den Optionen Integration von Aktivlautsprecher und Lautstärkeanpassung für RCA (diese ist softwaremäßig bereits realisiert mittels PGA2311)
* Evtl. eine Erweiterung für die zusätzliche Bedienung über ein Frontpanel

Da ich kein professioneller Hard-/Softwareentwickler bin, habt bitte ein Nachsehen, wenn es an manchen Stellen am letzte Quäntchen Präzision fehlt. Ich versuche nicht grob fahrlässig zu handeln.

Beginnen möchte ich die Dokumentation mit dem Herzstück des LSU, dem Steuerungsmodul.

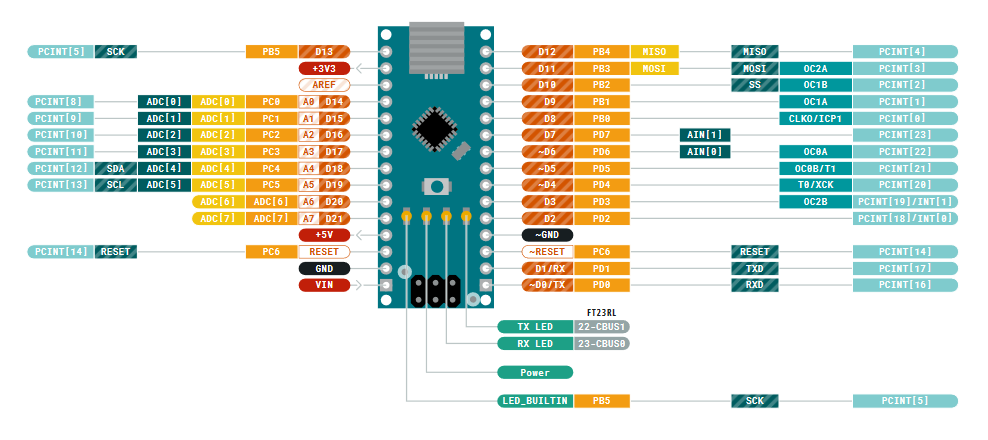


**Der Mikrocontroller:**

Als Schaltzentrale kommt, wie bereits im Ur-LSU, ein Arduino NANO zum Zuge. Dieser beherbergt neben dem eigentlichen Mikrocontroller ATmega328P von Microchip, einen USB-Anschluss für die Programmierung sowie der Kommunikation über eine serielle Schnittstelle, einen Reset-Taster und Spannungswandler zur Erzeugung der Betriebsspannungen von +5V und +3,3V.

Der Vollständigkeit halber hier ein paar technische Daten des ATmega328P.

|  |  |
| --- | --- |
| Microcontroller | ATmega328 |
| Betriebsspannung | 5 V |
| Eingangsspannung | 7-12 V |
| Flash Speicher | 32 KB wobei 2 KB durch den Bootloader belegt sind |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Taktrate | 16 MHz |
| Analog IN Pins | 8 |
| Digital I/O Pins | 22 (davon 6 mit PWM) |
| Schnittstellen | Seriell, SPI, I2C |
| Max. Strom pro I/O Pin | 40 mA |
| Stromverbrauch | 19 mA |
| Platinengröße | 18 x 45 mm |



(Quelle www.arduino.cc)

Wie man an der Pin-Belegung des Arduino NANO sieht, haben die einzelnen Pins unter Umständen mehrerer Funktionen, die über die Programmierung gesteuert werden. So kann der Pin A5 z.B. eine Interrupt-Quelle, ein analog Eingang oder Teil des I2C-Busses sein.

Wo wir gerade den Begriff I2C-Bus haben  Dieser ist ganz zentral für die Funktionalität des LSU. Hierrüber werden sämtliche Module mit der Prozessoreinheit verbunden.

Der I2C-Bus oder auch TWI (Two Wire Interface), im Datenblatt des ATmega328 genannt, ist ein kleiner Steuerbus. Entwickelt wurde er von Philips zur Kommunikation zwischen einer Zentraleinheit und verschiedenen Sensoren oder weiteren Zentraleinheiten auf einer Platine. Es handelt sich hierbei um ein vergleichsweise langsames und einfaches Zweidrahtbussystem, welches den Leitungsaufwand erheblich reduziert und damit die Baugruppenverbindungen übersichtlicher und einfacher gestaltet. Das mit dem „auf einer Platine“ trifft so auch nicht mehr zu, wird der I2C-Bus mittlerweile auch als Hausbus über Leitungslängen von über 100m eingesetzt.

Master

Arduino NANO

Slave 1

LCD-Display

Slave 2

Modul 1

Slave x

Modul x

…

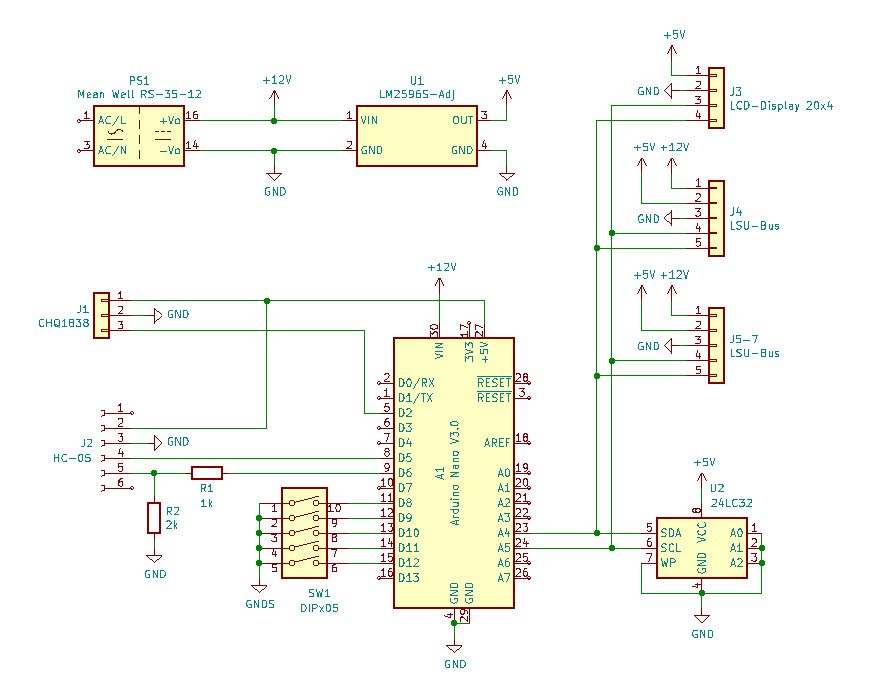
**SCL**

**SDA**

Die Übertragungsgeschwindigkeit des I2C-Busses ist mit 100kHz bis 3,4MHz nicht besonders hoch und wurde auch nicht für die Übertragung von großen Datenmengen konzipiert. Bezogen auf den Arduino sind maximale Datenpakete von 32 Byte möglich.

Der Bus selbst besteht aus einer Datenleitung (SDA) und einer Taktleitung (SCL). Diese sind an den Pins A4 und A5 des Arduino NANO herausgeführt und werden durch 3 weitere Leitungen (+12V, +5V und GND) zum LSU-Bus erweitert.

**Der Schaltplan:**

****



Wie im Schaltplan ersichtlich werden nur eine geringe Anzahl, der vom Arduino NANO, bereitgestellten Pins genutzt. Dies sind im Einzelnen:

|  |  |
| --- | --- |
| Pin | Beschaltung |
| VIN / GND | 12V Betriebsspannung (Mean Well RS-35-12 AC/DC) |
| +5V | eine vom Arduino NANO erzeugte Ausgangsspannung. Diese wird zur Spannungsversorgung des IR-Empfängers und des Bluetooth-Adapters verwendet und darf nicht mit der +5V vom DC-DC StepDown Wandler verbunden werden |
| A4 | Datenleitung SDA vom I2C-Bus |
| A5 | Taktleitung SCL vom I2C-Bus |
| D2 | Signalleitung zum IR-Empfängers |
| D5 | Empfangsleitung Rx vom Bluetooth-Adapter |
| D6 | Sendeleitung Tx vom Bluetooth-Adapter |
| D8 | Dip-Schalter 1: Option Kombiboard |
| D9 | Dip-Schalter 2: Option Lautsprecher 1 und Verstärker 1 an, wenn LSU ausgeschaltet |
| D10 | Dip-Schalter 3: |
| D11 | Dip-Schalter 4: |
| D12 | Dip-Schalter 5: |

**Die Spannungsversorgung:**

Der Arduino NANO selbst kann mit einer Eingangsspannung zwischen 6V und 20V betrieben werden. Da im weiteren Aufbau Relaiskarten mit einer 12V Spannungsversorgung zum Einsatz kommen, bot es sich an diese auch für den Gesamtaufbau des LSU zu nutzen. Hier vertraue ich auf ein Mean Well RS-35-12 AC/DC Netzteil welches mit 3A einen ausreichend großen Ausgangsstrom liefern kann.

Sicherheitshinweis:  
Bei Verwendung des obigen Netzteils liegen die Anschlüsse für die Netzspannung offen. Generell erfolgen Arbeiten mit Netzspannung auf eigene Gefahr. Bei unsachgemäßem Umgang oder Konstruktionsfehlern drohen lebensgefährliche Stromschläge. Für eventuelle Schäden oder Verletzungen übernehme ich keine Verantwortung. Die Arbeiten erfolgen auf eigene Gefahr.   
Wer den LSU aufbaut, sollte ausgebildete Elektrofachkraft sein und wissen was er/sie tut.

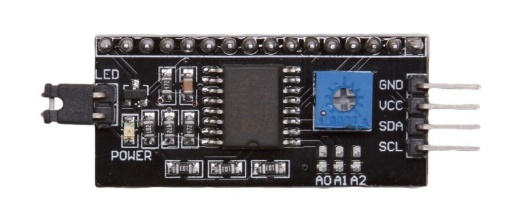
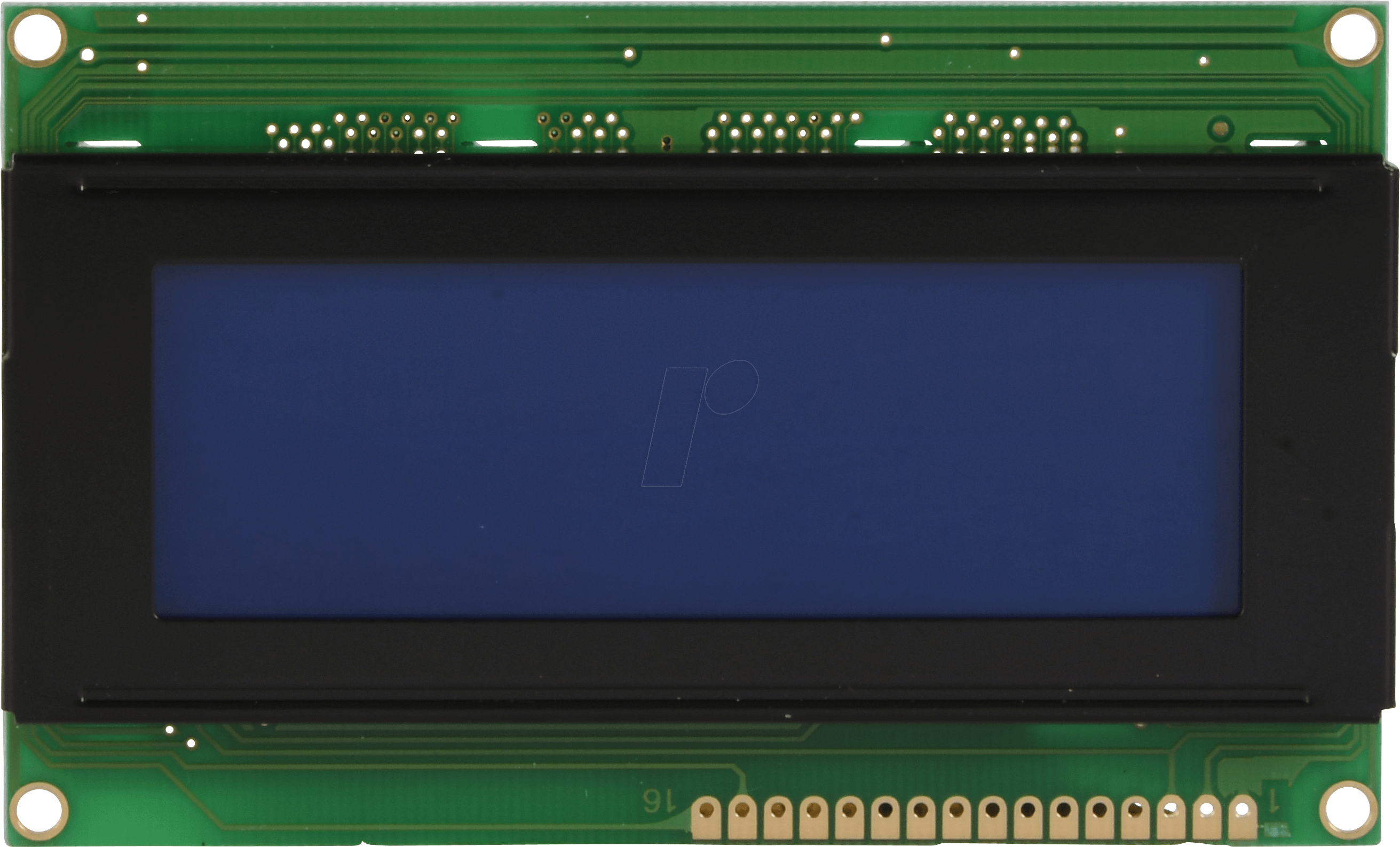
An dieser Stelle kann auch ein leistungsmäßig gleichwertiges Steckernetzteil verwenden werden.

Die 12V werden direkt mit den Anschlüssen VIN und GND des Arduino NANO verbunden. Dieser erzeugt für den IR-Empfänger und Bluetooth-Adapter die benötigte Betriebsspannung von +5V.



Um den Arduino NANO nicht mit dem LSU-Bus zu belasten erhält dieser, über einen DC-DC StepDown Spannungsregler (LM2596), eine eigene +5V Spannungsversorgung. Die Ausgangsspannung ist im Vorfeld über das Spindelpotentiometer auf +5V einzustellen.

**Das LCD-Display:**

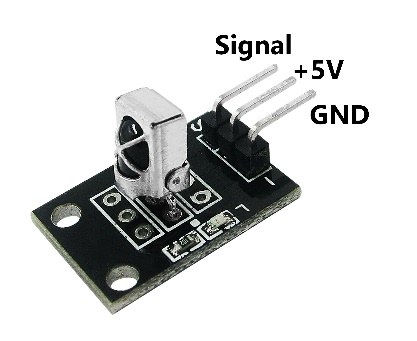
****

Mit 4 Zeilen a 20 Zeichen bietet das Display genug Raum für die Anzeige der gerade aktiven Komponenten sowie Fehler- und Statusmeldungen. Normalerweise werden diese LCD-Displays mittels SPI- oder Parallel-Schnittstelle angesteuert. Findige Entwickler bzw. Programmierer haben aber eine Adaption an die I2C-Schnittstelle entwickelt. Dies ist mit einer der Gründe warum ich den I2C-Bus nutze. Meist muss man die I2C-Controller-Platine noch selbst auf das LCD-Display löten, was aber unproblematisch ist.

Achtung! Beim Verbinden mit dem Steuermodul ist auf die Reihenfolge der Anschlüsse zu achten (siehe Schaltung/Platine).

Kleiner Hinweis, sollte nach dem Anschließen das Display dunkel bleiben, liegt es unter Umständen an der Kontrasteinstellung. Diese kann über das blaue Potentiometer eingestellt werden.

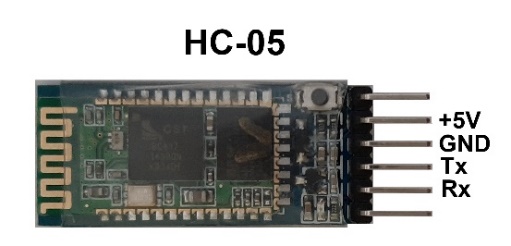
**Der IR-Empfänger:**

****

In vielen Starter-Sets für Arduino wird ein IR-Empfänger mitgeliefert. Diese kleinen Platinen, sind mit dem eigentlichen IR-Empfänger und einer 3 poligen Steckerleiste, für die Nutzung auf einem Steckbrett, ausgestattet. Da diese Platinen meist ungünstig für eine Montage in einem Gehäuse sind benutze ich einen Standard IR-Empfänger vom Typ CHQ1838. In der Stückliste werde ich das IR-Fertig-Modul angeben.

Zu beachten ist, dass die Pinbelegung des CHQ1838 von den IR-Fertig-Modulen abweicht.

Bei der Fernbedienung selbst, lehne ich mich an Rundmacher an und verwende innerhalb des Programms die IR-Codes der Codeliste 2393. Um die Codes für eine andere Fernbedienung zu ermitteln, werde ich ein separates Programm zur Verfügung stellen.

**Der Bluetooth-Adapter:**

Hier verwende ich ein herkömmliches, aus der Arduino-Welt bekanntes, HC-05 Fertig-Board. Alternativ kann man auch ein HC-06 Board einsetzen. Im Normalfall sind diese Plug&Play und werden mit 4 Leitungen (+5V, GND, Tx und Rx) mit dem Arduino NANO verbunden. Da die Tx und Rx Leitungen aber nur Signale mit 3,3V Pegel verarbeiten können muss das Signal vom Arduino an Rx noch mit einem Spannungsteiler auf die 3,3V reduziert werden. Die Tx Leitung kann 1:1 an den Arduino geführt werden, da der NANO auch 3,3V Signale direkt verarbeiten kann.

In einem Metallgehäuse wird der Empfang wahrscheinlich nicht gegeben sein. Hier muss man ggf. eine Öffnung mit einer Kunststoffabdeckung vorsehen und die Leitungen entsprechend verlängern.   
Ich habe bisher leider kein Bluetooth-Modul, mit einem Anschluss für eine externe Antenne, gefunden.

**Der EEProm-Speicher:**

Dieser „nichtflüchtige Speicher“ ist erst vor kurzem hinzugekommen als mir klar wurde, dass ich evtl. aus Speicherplatzgründen, auf den Arduino NANO Every umsteigen muss. Da dieser aber nur noch einen EEProm-Speicher von 256 Byte statt der 1024 Byte vom ursprünglichen NANO besitzt, hielt dieser separate EEProm-Speicher-Baustein, mit einer Größe von 4048 Byte, Einzug. Ziel ist es später die gesamte Konfiguration (Einstellungen, Bezeichnungen, Favoriten usw.) hier zu speichern um diese nach dem Einschalten wieder herstellen zu können. Zurzeit wird er noch nicht genutzt aber den Steckplatz kann man ja schon mal vorsehen. ;-)

**Die Platine:**

Durch die doch recht überschaubare Anzahl von verwendeten Pin’s und der Nutzung des I2C-Busses, gestaltet sich die Platine schön übersichtlich. Viele Leitungen verlaufen parallel und die Pin-Belegung des Arduino NANO begünstigt eine anschlussmäßig nahe Platzierung der Bauteile. Daher erfolgt der Aufbau auch auf einer einfachen Streifenrasterplatine (Rastermaß 2,54mm).

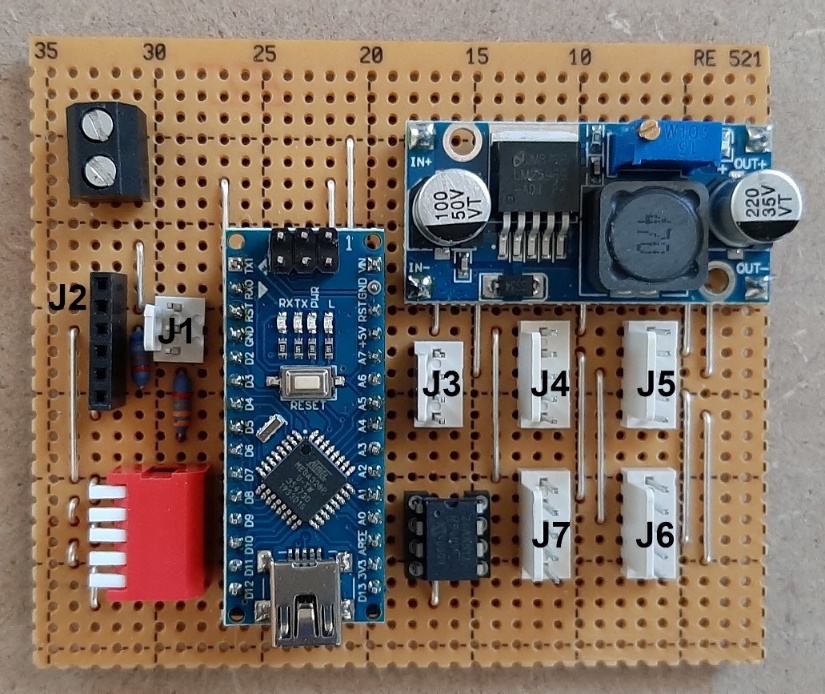
­­­

Abbildung entspricht nicht dem endgültigen Layout, da EEProm nachträglich eingesetzt wurde

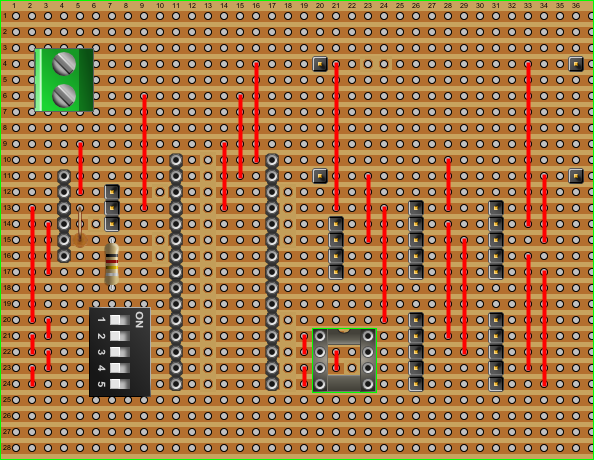
Die weißen Steckplätze (J3 – J7) auf der rechten Seite sind der LSU-Bus mit den Anschlüssen +12V, +5V, GND, SCL und SDA von oben nach unten gesehen. Die Anzahl der Steckplätze richtet sich nach der Anzahl der verwendeten Module. Den letzten Anschluss (neben dem EEProm) würde ich immer an die Gehäuserückseite mit einer Buchse herausführen um ggf. weitere Module anschließen zu können.

Wie man sehen kann, ist der Steckplatz J3 nur ein 4fach Steckplatz. Dieser ist für das LCD-Display vorgesehen, da dieses keine 12V benötigt. Hier direkt der Hinweis, dass die Steckerbelegung nicht 1:1 mit dem LCD-Display übereinstimmt. An meinem LCD-Display ist die Reihenfolge GND, +5V, SDA und SCL.

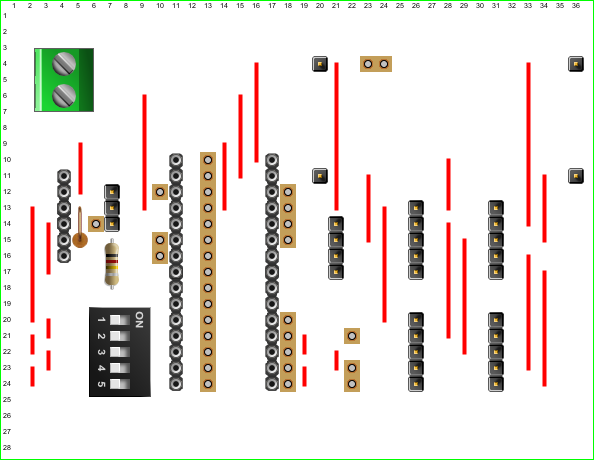
Auf der linken Seite finden sich der Steckplatz J2 für den Bluetooth-Adapter HC-05. Es sind nur die mittleren 4 Pin’s belegt. Die Reihenfolge der Anschlüsse von oben nach unten sind +5V, GND, Tx und Rx. Daneben ist der Steckplatz J1 für den IR-Empfänger mit seinen Anschlüssen +5V, GND und Signal.

Die Dip-Schalter können weggelassen werden und sofern eine der Optionen benötigt wird mit einer Brücke zwischen GND und dem jeweiligen Anschlusspin des Arduino NANO hergestellt werden.

Die Platine als BlackBoard-Grafik. Es fehlt der Arduino Nano und der DC-DC StepDown-Wandler.



Für die bessere Übersicht habe ich im nächsten Bild die Platine und den Sockel für das EEProm ausgeblendet. Die braunen Kästchen mit einem Loch sind die Leiterbahnunterbrechungen damit es nicht zu Fehlfunktionen bzw. Kurzschlüssen kommt.



Vor der Inbetriebnahme der Platine, sind benachbarte Pin’s auf Kurzschlüsse, die Leiterbahn-Unterbrechungen sowie die korrekte Verdrahtung, anhand des Schaltplans, zu prüfen.

**Die Stückliste und Bezugsquelle:**

An dieser Stelle verzichte ich erstmal auf eine Stückliste, da noch kein funktionstüchtiger LSU aufgebaut werden kann und somit auch nicht die Gesamtkosten abzuschätzen sind.

Damit wäre der 1. Teil der LSU-Dokumentation abgeschlossen. Ich hoffe es ist alles soweit verständlich und regt den geneigten Leser zum Nachbau an. Für das Aufspielen der Software wird es einen eigenen Kurzbericht geben.

Im nächsten Teil werde ich das Lautsprechermodul zum Umschalten von 4 bzw. 8 Lautsprechern auf Grundlage der I/0-Expander MCP23008 und -17 vorstellen. Mit diesen Modulen ist dann der Aufbau eines LSU004/08/12 oder in der Maximalausbaustufe LSU0016 möglich.

Wie immer am Ende meiner Posts, Anregungen und Fragen sind wie immer gern gesehen.

Viele Grüße

LianenSchwinger