Sommersemester 2022-  
Wintersemester 2022/23

Praxissemester:

01.08.22 – 28.02.2023

Hochschule Hamm-Lippstadt

Praxissemester

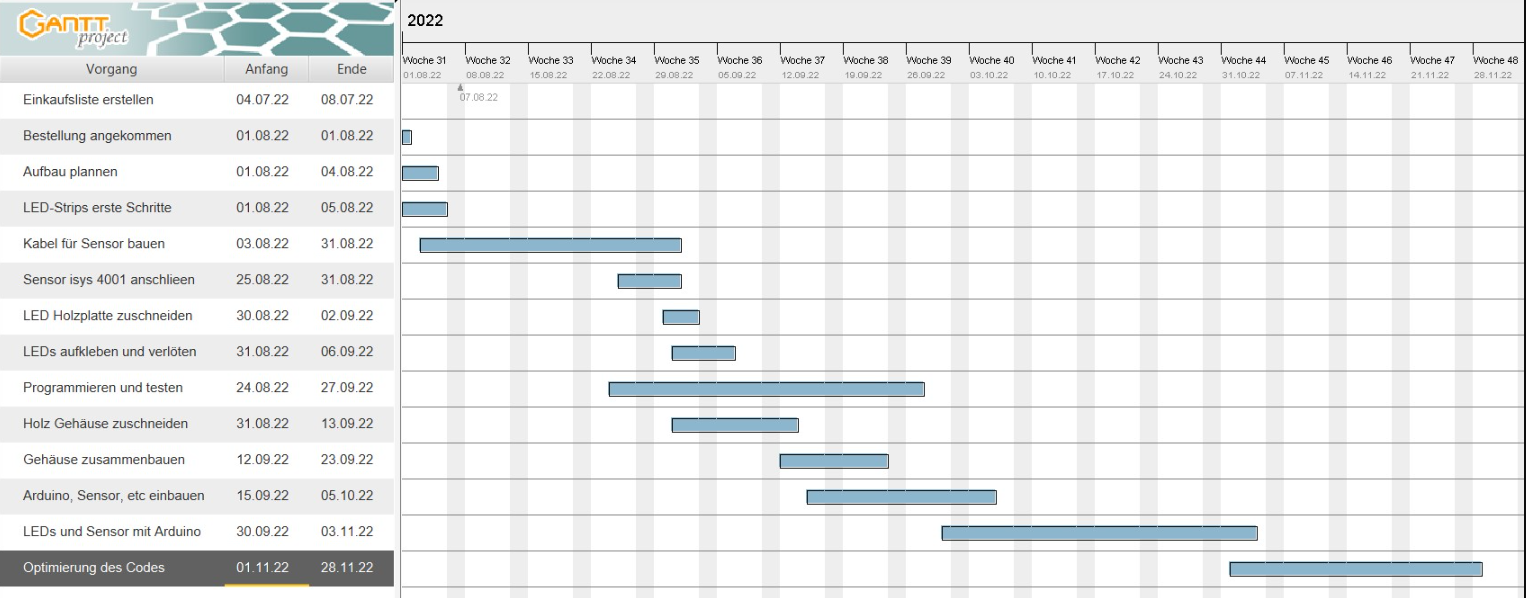
Betreuer/in: Prof. Schneider

* **Projektarbeit-**

Einleitung  
  
Der Schulweg von Kindern ist nicht sehr sicher. Durch Geschwindigkeitsmessungen des ADAC vor 25 Schulen in zehn Bundesländern wurden häufig Fahrer mit zu hohen Geschwindigkeiten gemessen. Die zulässige Höchstgeschwindigkeit von 30 km/h wurde von mehr als 60 Prozent der Autofahrer überschritten. Von insgesamt 43 828 Messungen, waren 26 329 zu schnell unterwegs. Vor einer Grundschule in Hamburg wurde ein Rekordwert von 96 km/h bei vorgeschriebenen 30km/h gemessen.  
  
Um die Autofahrer abzuschrecken, können sogenannte Tempomessgeräte helfen.  
Diese messen die Geschwindigkeit und zeigen diese über eine Anzeige dem vorbeifahrenden Fahrer. Zudem wird oft auch noch mit einem Smiley signalisiert, ob der Fahrer zu schnell fährt oder ob er sich an die vorgegebene Geschwindigkeit hält.

Aufgabenstellung  
  
Die Aufgabe des Projektes war es, ein solches Tempomessgerät mit einem Radarsensor und einer LED-Anzeige, welche die Geschwindigkeit in km/h anzeigt, zu entwickeln.  
  
Anforderungen  
  
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung  
  
Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung  
  
  
  
Projektplan (Justin Frommberger und Jonas Gerken)  
  
  
  
BOM (Justin Frommberger und Jonas Gerken)  
  
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Funktionaler Systementwurf (Justin Frommberger)

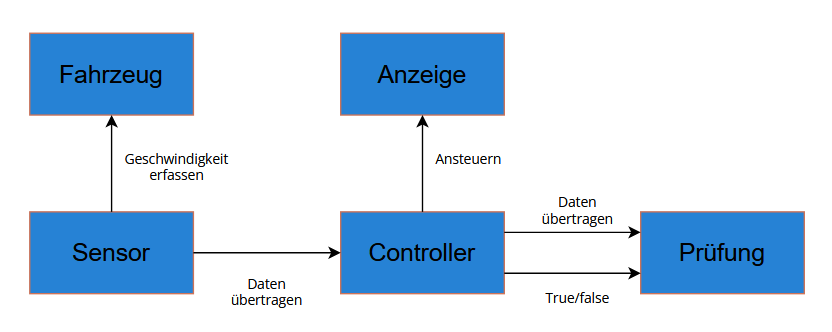


Abb. 1 Funktionaler Systementwurf

In Abb. 1 wird gezeigt, wie die Komponenten verbunden sind.  
Der Sensor erfasst die Geschwindigkeit des Fahrzeugs und übergibt diese dem Microcontroller. Mit dem Mikrocontroller werden die Daten berechnet, die er vom Sensor empfängt. Die berechneten Werten werden dann auf der Segmentanzeige dargestellt.

Technischer Systementwurf (Jonas Gerken)  
  
Ein Bild, das Text, Kasten, Behälter, Bilderrahmen enthält.

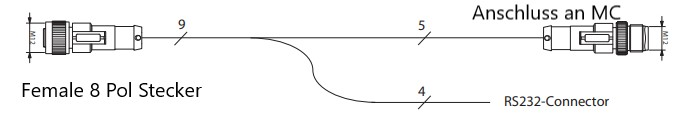
Automatisch generierte Beschreibung  
Abb. 2 Technischer Systementwurf

Die Abbildung 2 zeigt den technischen Systementwurf des Tempomessgerätes als CAD-Modell.

Radarsensor Isys 4001 (Jonas Gerken)

Der Kabelanschluss ist in Abb. 3 grafisch dargestellt.

Dargestellt ist der Female 8-polige Stecker, der mit dem Geschwindigkeitssensor verbunden wird. Das andere Ende des Kabels wird aufgeteilt in eine RS232-Schnittstelle und einem Stecker zur Verbindung mit dem Mikrocontroller der auch die Spannungsversorgung für den Sensor bereitstellt. Die RS232-Schnittstelle wird mit dem Computer verbunden zur Übertragung der Daten.

  
Abb. 3: Sensoranschluss

Um das Kabel anzuschließen, werden zwei Datenkabel mit Abschirmung benötigt. Das Kabel muss eine Abschirmung haben, um Störsignale zu vermeiden.

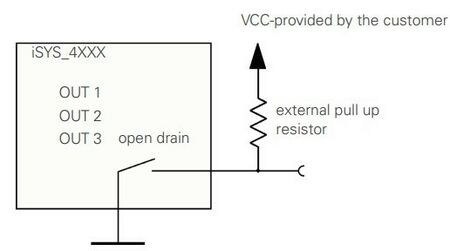
Für den RS232-Connector muss ein Kabel an einen Sub-D Stecke mit lötkontakten gelötet werden (Tabelle 1).

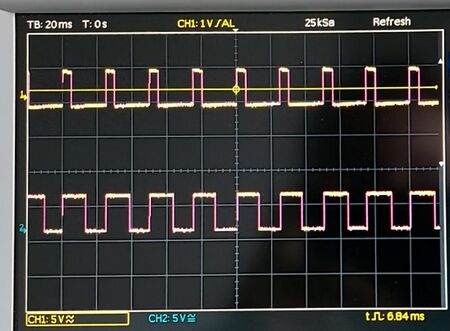
Wichtig ist es, die einzelnen Adern mit einem kleinen Schrumpfschlauch zu versehen, damit die Aderisolierung beim Löten nicht beschädigt wird. Das zweite Kabel ist für die Ausgänge Out1, Out2, Out3, genauso wie die Stromversorgung und der Ground.

Ein Bild, das Text enthält.

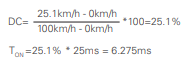
Automatisch generierte Beschreibung

Für das zweite Kabel, welches für die Datenoutputs und die Stromversorgung verwendet wird, sieht die Pin-Belegungen wie folgt aus. (Tabelle 2)  
  
Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung  
  
Da der Sensor mit dem selbst gebauten Kabel nach mehreren Überprüfungen der Verbindungen nicht mit dem GUI-Programm verbunden werden konnte, wurde ein Kabel von der Firma InnoSenT angefragt. Mit diesem Kabel war es dann möglich, den Sensor mit dem GUI zu verbinden und zu konfigurieren.  
  
Sensormessung  
  
Damit ein PWM-Signal vom Sensor gesendet wird, muss zunächst ein Pull-up Widerstand (10kOHM) wie in Abb. 4 gezeigt, zwischen den Outputs geschaltet werden. Bei dem Anschluss an den Arduino muss beachtet werden, dass die Ausgänge OUT 1-3 an die PWM-Schnittstellen angeschlossen werden.  
  
  
Abb. 4 OUT-Anschlüsse/Verkabelung

Das Signal kann nach der Verkabelung über ein Oszilloskop angezeigt werden. (Abb. 5)  
Das obere PWM-Signal ist die Entfernung und das untere die Geschwindigkeit.  
  
  
Abb. 5 PWM-Signal

Der Sensor sendet die PWM-Werte an den Arduino. Mit dem Arduino Befehl pulseIn(pin, HIGH), kann das Signal ausgelesen werden.   
  
Mit den folgenden Formeln (Abb. 6) wird aus einer abgelesenen Geschwindigkeit das DC und folgend das „Ton“ berechnet.  
Der Sensor sendet Ton-Werte an den Arduino und dieser muss in eine Geschwindigkeit umgerechnet werden.   
Dazu muss die obere Formel nach km/h umgestellt werden und die untere Formel nach DC.  
Rechts sind die umgestellten Formeln aufgezeigt, um die Geschwindigkeit zu berechnen.

  
  
  
  
 Abb. 6 Duty Cycle Formel

LED-Seven-Segment Aufbau und Verlötung (Justin Frommberger)  
  
Der Erste Schritt ist, für die LED-Seven-Segmente eine Holzplatte auf 500 mm x 500 mm zuzuschneiden.  
Auf dieser Holzplatte werden dann die LED-Stripes aufgeklebt.

Wir haben uns für die Seven-Segmente anzeige entschieden.

Diese LED-Strips haben wir doppelt nebeneinander geklebt, damit die angezeigte Geschwindigkeit später auf der Straße besser zu lesen ist.

Dazu wurden 8 Strips von einer Länge von 20 LEDs und 12 Strips von einer Länge von 8 LEDs zugeschnitten und angeklebt.



Abb. 7 LEDs auf Holzplatte

Beim Aufkleben ist es wichtig, die LEDs richtig herum aufzukleben.

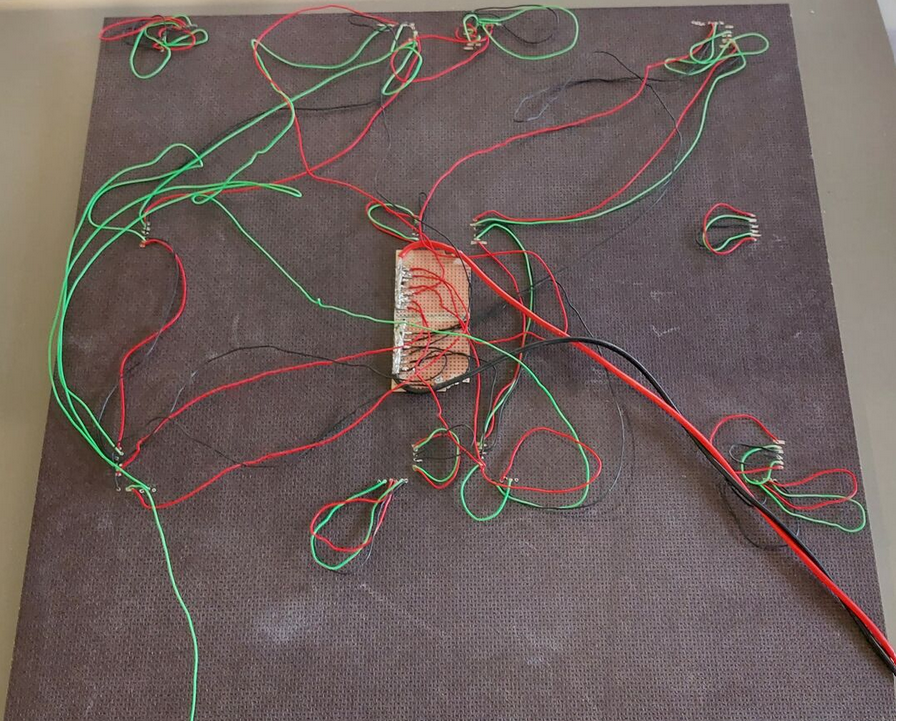
Drauf geachtet werden muss, dass man mit der Kennzeichnung Strips "Di" für Digital In beginnen muss.

Der erste Strip wurde von oben nach unten geklebt, der Stripe rechts daneben von unten nach oben um die Reihenfolge einzuhalten.

So müssen unten später nur 3 kurze Kabel angelötet werden.

Von links außen und rechts außen wurden 50 mm und von dem unteren Rand wurden 15 mm Platz bis zu den LED-Stripes gelassen.

Abb. 7 zeigt die fertige Segmentanzeige.

  
Abb. 8: Verkabelung

Bei einer so großen Anzahl an LEDs ist es notwendig, die LEDs an mehreren Stellen mit der Stromquelle zu verbinden.

Dazu wurden, wie in Abb. 8 zusehen ist ein rotes und ein schwarzes 4 mm² Kabel an eine Europlatine gelötet, welche dann die Hauptstromversorgung darstellt.

Die Seven-Segmentanzeige wurde dafür jeweils in drei Abschnitte unterteilt.  
Das Seven Segment wurde wie folgt aufgeteilt:  
- die linken senkrechten LEDs  
- die rechten senkrechten LEDs  
- die waagerechten LEDs   
  
  
Auf der Holzplatte wurden an jeder Ecke 3 Löcher gebohrt, um die Kabel zur Rückseite durchzuführen und dort zu verdrahten.

Für die Verbindungen zwischen den Strips, wurde ein 0,5 mm² Kabel in zwei verschiedenen Farben: Rot (Strom), Schwarz (Erde), Grün (Daten) zugeschnitten und verlötet.

Das Datenkabel wird nicht in Reihen aufgeteilt, denn alle LEDs müssen zur Programmierung in Reihe angeschlossen werden. So kann später mit Variablen jede einzelne LED angesteuert werden.

Die Abschnitte werden jeweils an die Europlatine angelötet, wie auf dem Bild zu sehen ist. (Abb. 8)

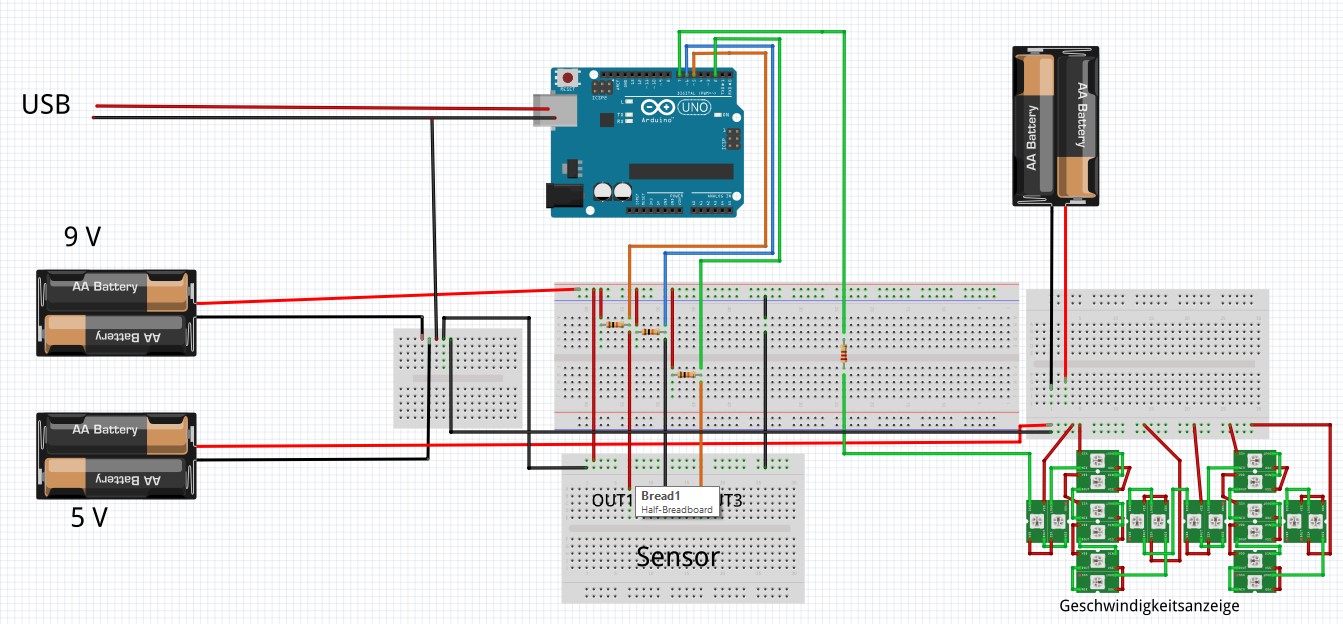
LED Dimmung (Jonas Gerken)  
  
Um die Autofahrer mit den LED‘s nicht zu blenden, haben wir uns überlegt, eine Abdeckung mit dem 3D-Drucker zu entwerfen um das Licht zu dimmen. Die einzelnen Bauteile für die Abdeckung wurden mit SolidWorks gezeichnet und in eine STL Datei umgewandelt. Die Einzelteile die mit dem 3D-Drucker gedruckt wurden, haben wir anschließend auf die Holzplatte über die LED‘s festgeklebt.

In Abb. 9 ist dann das Ergebnis der fertigen Abdeckung zusehen.  
  
  
Abb. 9 LED-Abdeckung  
  
  
Gehäuse (Justin Frommberger)  
  
Für das Gehäuse wurden drei 11 mm dicke Holzplatten auf 250 mm \* 500 mm zugeschnitten. Eine von den Holzplatten wird mit drei Nuten versehen, die Nuten sind zur befestigen der LED-Platte, der Plexiglasscheibe und der Rückwand.

Mit der zweiten und dritten Platte wird die Außenwand befestigt. Die Platte für die Rückwand wird aus einer 11 mm dicken Holzplatte auf 500 mm \* 500 mm zugeschnitten. Zudem muss auch die Plexiglasscheibe auf 500 mm \* 500 mm zugeschnitten werden, um in die dafür vorgesehene Halterung zu passen. Am Ende schnitten wir die letzte Platte auf 500 mm \* 500 mm und klebten auf diese Platte, die 7 Segment LED Pixelstreifen.

Wir montierten alle Teile des Gehäuses zusammen und befestigten diese zusätzlich mit Schrauben für einen besseren Halt.

Der fertige Aufbau ist in Abb. 10 zusehen.  
  
  
Abb. 10 Gehäuse

Elektronik (Jonas Gerken)  
  
Die Abb. 11 zeigt die Elektrische Schaltung.  
  
  
Abb. 11: Schaltung

Stromquellen:

* Radarsensor: 5 bis 30 Volt (in Abb. 7 links)
* LED-Geschwindigkeitsanzeige: 5 Volt (in Abb. 7 rechts)

Arduino Uno:

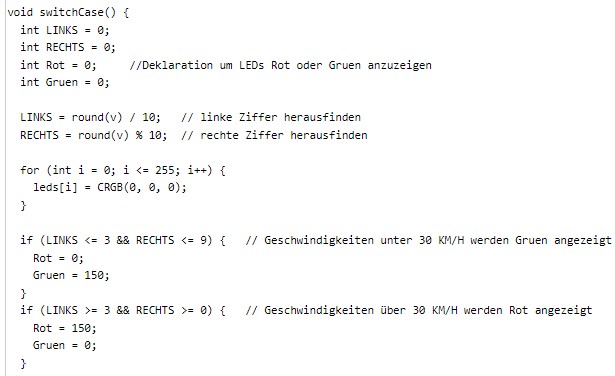
* Digital Pin 3, 5, 6: OUT1, OUT2, OUT3 des Radarsensors
* Digital Pin 7: Digital der Geschwindigkeitsanzeige

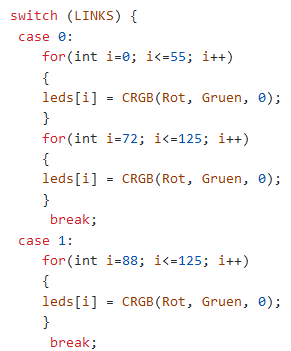
Radarsensor:

* OUT1, OUT2, und OUT3 mit einem PULL-UP Widerstand verbunden

LED-Geschwindigkeitsanzeige:

* Senkrechten LED-Streifen jeweils links und rechts mit der Stromquelle verbunden
* die mittleren LED-Streifen jeweils links und rechts mit der Stromquelle verbunden

Programmierung  
  
**Farbe der LEDs** (Justin Frommberger)  
Abb. 12 LED-Programmierung (Farbe)

**Zahlen Anzeige** (Justin Frommberger)   
  
  
Abb. 13 Zahlen Anzeige

Um die Zahlen auf dem Display anzuzeigen, wurde ein Switch Case programmiert, um die benötigten LEDs einzuschalten. Wie im Beispiel Case 0, werden alle LEDS von 0 bis 55 und von 72 bis 125 eingeschaltet. **Sensor** (Jonas Gerken)  
Ein Bild, das Text enthält.

Automatisch generierte BeschreibungAbb.14 Sensor Programmierung

Ergebnis (Justin Frommberger und Jonas Gerken)  
  
Fast alle Anforderungen des Projektes konnten erfüllt werden. Wir unterschätzten, dass die LED‘s eine Menge an Strom benötigen (aktueller Stand 10A), so mussten wir bei unserem Projekt LED‘s einsparen und beschlossen das Smiley Gesicht nicht einzubauen.

Um trotzdem die gewünschte Aufgabe zu erfüllen, überlegten wir uns, dass wir die nicht überschrittene Geschwindigkeit Grün anzeigen lassen.

Die zu hohe Geschwindigkeit stellen wir in roter Farbe dar. Zudem haben wir uns für ein kleineres Format des Schildes entschieden, um unser Projekt besser transportieren zu können. Auch haben wir uns gegen einen Schriftzug vom km/h entschieden, um die Zahlen möglichst groß und sichtbar für die Fahrer zu gestalten. Leider haben wir festgestellt, dass der vorhandene Radarsensor Isys 4001 eine Störung hat und keine konstanten Werte zurückliefert, dieses führt bei der Anzeige zu ungenauen Werten.

Alle anderen oben aufgelisteten Anforderungen konnten wir erfüllen.

Zusammenfassung (Justin Frommberger und Jonas Gerken)  
  
Lessons learned  
  
Das Team hat die folgenden Kompetenzen erlernt:

* Projektplanung und Zeitmanagement
* CAD-Konstruktion für 3D Druck
* 3D Druck
* Löten
* Ansteuerung von WS2812 LEDs
* Bau eines Kabels

Besondere Herausforderungen  
  
Die erste Herausforderung bei unserem Projekt „Speed-Tempomessgerät“, stellte sich direkt am Anfang bei der Materialbeschaffung. Das Problem war es, das alle Materialien, die für das Projekt benötigt werden zu beschaffen. Wir stellten fest, dass bei dem vorhandenen Radarsensor der Hochschule das Kabel zur Strom- und Datenversorgung fehlte. Weil das Kabel kein Standardkabel war, informierten wir uns bei dem Hersteller des Sensors wie das Kabel aufgebaut ist und ließen uns das dazugehörige Datenblatt schicken. Um das Kabel selbst anzufertigen, bestellten wir dafür einen RS232 Sub-D Stecker und einen M12 8 poligen Stecker. Das Herstellen des Kabels konnte nach langem ausprobieren und telefonischem Austausch mit der Herstellerfirma nicht abgeschlossen werden. Folgend fragten wir die Firma, ob sie uns ein Kabel zuschicken könnten. Freundlicherweise ließen sie uns ein Kabel kostenlos zukommen, um unser Problem zu lösen.

Die nächste Herausforderung war es, herauszufinden, wie groß die Netzteile seien müssen, um unser komplettes Projekt mit Strom zu versorgen. So recherchierten wir in den Datenblättern des Sensors und der RGBs, wie groß die Netzteile sein müssen. Später stellte sich heraus beim Anschließen der RGB-Kette, dass die LED‘s viel mehr Strom in Reihenschaltung benötigten als wir berechnet hatten (über 20A). Folgend beschlossen wir unsere Schaltung um zulöten, wir setzten die Stromversorgung mittig und teilten diese in 4 Reihenschaltungen auf. Wir verringerten den benötigten Stromverbrauch auf ungefähr 10A. Zusätzlich verzichteten wir auf weitere RGBs für das Smiley, um Strom zu sparen und ein transportables Netzteil nutzen zu können.

Eine weitere Herausforderung im Projekt war die Ansteuerung der einzelnen RBGs, da diese teilweise zusammengeschaltet sind und beim Ansteuern einer RBG zwei Stück eingeschaltet werden. Nach mehrfachen neu Löten konnte der Fehler nicht gefunden werden.

Daraufhin haben wir uns eine neue Schaltung (Abb.8) entwickelt, mit der wir unsere Schaltung umsetzen konnten.