

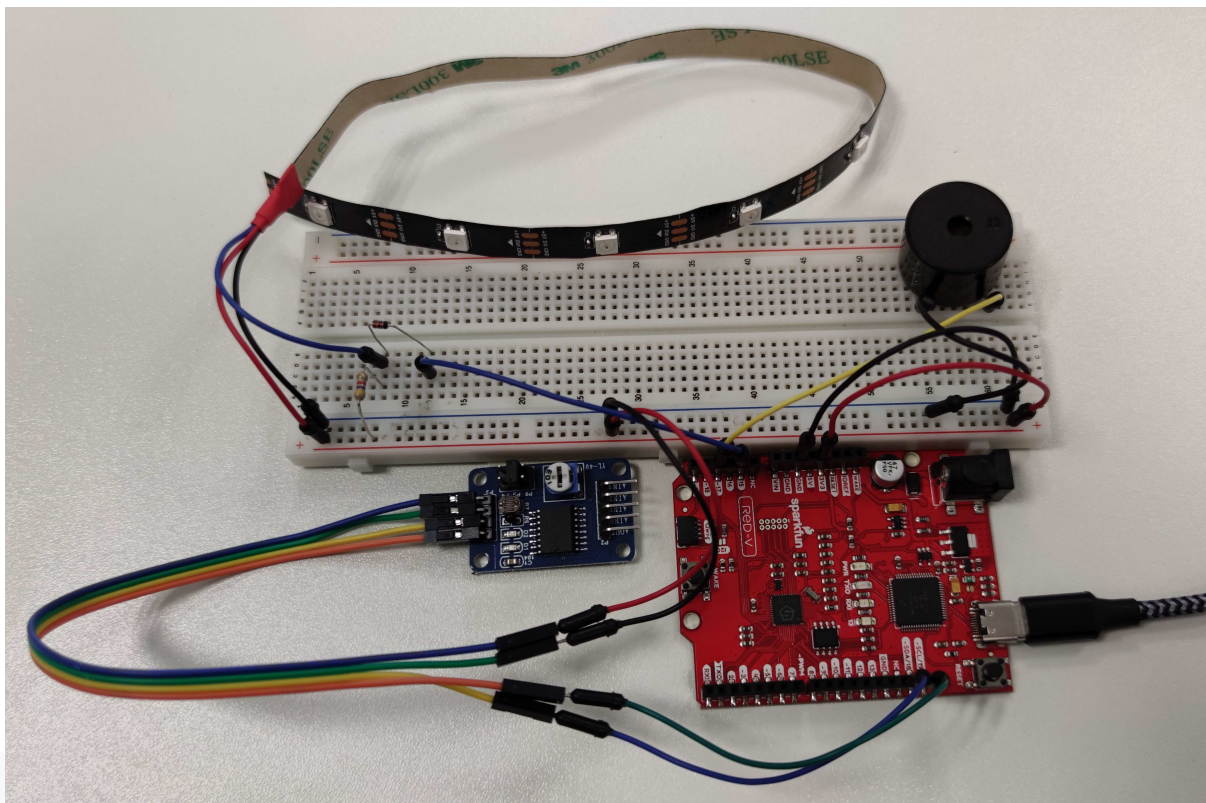
## Praktikumsaufgabe 3 - Weihnachtsbaum

### Aufgabe

In diesem Praktikum implementieren Sie die Steuerung für eine Weihnachtsbaumbeleuchtung auf dem RED-V Board. Nutzen Sie dafür die im Labor vorhandene Hardware oder die *xmas*-Konfiguration im Simulator. Die Abgabe und die Bewertung besteht aus mehreren Anforderungen, die erfüllt werden müssen. Diese sind im Folgenden aufgelistet. Bitte lesen Sie sich alle Informationen gründlich durch.

### Funktionsweise

Die Steuerung für den Weihnachtsbaum umfasst drei Teile (siehe Abbildung 1). Zum einen muss ein LED-Stripe angesprochen werden, um diesen zum Leuchten/Blinken zu bringen. Der zweite Teil umfasst das ansteuern eines Piezo-Buzzers zum Abspielen einer vorgegebenen Melodie. Der letzte Teil besteht aus dem Lesen eines analogen Helligkeitssignales über einen A/D-Wandler, um den LED-Stripe und den Piezo Buzzer nur zu aktivieren, wenn es dunkel genug ist.



**Abbildung 1:** Weihnachtsbaumsteuerung mit LED-Stripe, Helligkeitssensor und Piezo-Buzzer

## LED-Stripe

Der LED-Stripe besteht aus mehreren RGB-LEDs. Auf dem LED-Stripe sollen drei Farben leuchten. Dieses Triple soll sich entlang des Stripes wiederholen. Die drei Farben des Triple sind rot, grün und gelb. Um das Blinken zu erzeugen, soll diese Farbkette bei jedem Update um einen weiter geschoben werden. Wenn also LED 1, 2 & 3 initial die Farben 1, 2 & 3 haben, dann sollen die LED nach einem Update die Farben 3, 1 & 2 haben. Das Weiterschieben der Farben soll abhängig von der Melodie des Buzzers sein. Beim Spielen einer neuen Note soll ein Weiterschieben ausgeführt werden.

Der LED-Stripe ist ein WSB2812b mit 10 LEDs. Zum Setzen der LEDs wird ein eigenes Protokoll verwendet, bei dem ein Pin für bestimmte Intervalle auf high oder low gesetzt werden muss. Diese Intervalle sind dem Datenblatt zu entnehmen. Um diese Zeitintervalle einhalten zu können, muss dieses sogenannte Bit-Banging in Assembler umgesetzt werden. Nutzen Sie hierfür Inline Assembler in C, um die zeitkritischen Stellen zu implementieren.

## Piezo-Buzzer

Das Lied, welches der Piezo-Buzzer spielen soll, ist in der Header-Datei song.h im ILIAS gegeben. Dort ist das Lied in Form eines Arrays von Noten gespeichert. Jede Note ist gegeben als eine Frequenz. Ein zweites Array gibt zusätzlich an, wie lange jede Note erklingen soll. Um den Ton zu erzeugen, muss an dem Pin des Buzzers die richtige Frequenz für diese Dauer anliegen.

## Helligkeitssensor

Als Helligkeitssensor wird ein Photoresistor verwendet. An diesem liegt eine Spannung abhängig von der Stärke des Lichts, das auf ihn einfällt, an. Da das RED-V Board keine analogen Eingänge besitzt, wird der PCF8591 A/D-Wandler zwischen dem Fototransistor und dem Board eingefügt. Das Board kommuniziert über eine I2C Schnittstelle mit dem A/D-Wandler, um die Helligkeit auslesen zu können. In dem Datenblatt des A/D-Wandlers ist beschrieben, wie dieser angesprochen werden muss, um den analogen Input auszulesen. Die Pins A1, A2 und A3 sind auf low gezogen. Ermitteln Sie anhand des Datenblattes die korrekte Adresse des A/D-Wandlers und stellen sie diese in der Konfiguration in der Simulation ein. Der Helligkeitswert ist gegeben in Form eines Bytes. Das Licht soll angeschaltet und die Melodie abgespielt werden, wenn der Schwellwert 170 überschritten wird.

## Technische Umsetzung

- Benutzen Sie Inline Assembler zur Umsetzung des Bit-Bangings des LED-Stripes.

- Setzen Sie die MISRA-Normen bei der Implementierung um und evaluieren Sie Ihren Code, wie in der Vorlesung gezeigt. Verbessern Sie ihren Code so weit wie möglich, um möglichst keine Warnungen zu bekommen.
- Der abzugebende Code für das Spiel **muss** kompilierbar und in der Simulation sowie auf dem RED-V Board lauffähig sein. Kommentieren Sie notfalls fehlerhafte Codeabschnitte bzw. Funktionalitäten aus. Die Lauffähigkeit hat Vorrang.

### **Pinbelegung**

- LED-Stripe => 15
- Piezo-Buzzer => 17
- A/D-Wandler => SCL & SDA
- Fototransistor => AIN0 vom A/D-Wandler

### **Abgabe für die Bewertung (Abgabe im Ilias als ZIP)**

1. Der implementierte Code der sowohl die Funktionsweise als auch die technische Umsetzung erfüllt.