

Arduino-Grundlagen

Miniprojekt

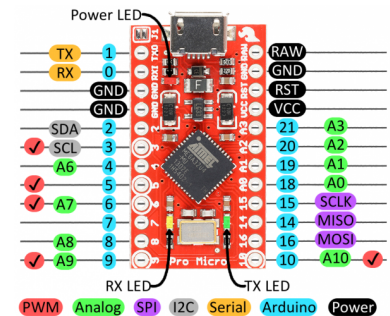
Projektanfrage

a) Hardware

- Arduino mit Schaltung und bespielter Software
- Aufkleber mit Name und Projekttitel auf dem Breadboard

b) PDF-Datei

- Titelblatt
- Aufgabenstellung
- Programmcode
- Fotos des Aufbaus
- Schaltplan (z. B. mit fritzing erstellt)
- Testkonzept (→ Zustandsbasiertes Testen)
- Video-Link auf Ihre Live-Demo
- Quellenangabe
- Eigenständigkeitserklärung



Thema 1: Morsecode Decoder/Encoder

Anforderungen:

Durch eine dunkle Röhre (z.B. die Pappöhre einer Küchenpapierrolle) sollen Morsecodesignale von einem Ende zum anderen Ende übertragen werden. Ein Photoresistor, der an einen Arduino angeschlossen ist, soll am Empfangsende die Lichtsignale empfangen und in Characterzeichen, bzw. einen String übersetzen und ausgeben. Die Lichtsignale an der Senderseite können im ersten Schritt durch das Öffnen und wieder Verschließen der Röhre stattfinden. Es kann auch eine Taschenlampe verwendet werden, um die Lichtsignale zu senden.

Erweiterung: Die Senderseite der Röhre wird durch einen zweiten Arduino erweitert, der mit einer oder mehreren LEDs Morsecode in die Röhre schickt. Am sendenden Arduino soll es möglich sein, Text einzugeben. Dieser soll über die Röhre codiert in Morsecode an den empfangenden Arduino gesendet werden.

Thema 2: Alarmanlage

Anforderungen:

Eine Alarmanlage soll Tonsignale von sich geben, wenn sich jemand der Anlage nähert. Die Anlage soll per Einschalten nach 10 Sekunden für eine bestimmte Dauer scharf gestellt sein (hard-kodiert, z. B. 10 Stunden). Innerhalb der ersten 10 Betriebssekunden kann man alternativ die Dauern in Stunden ändern. Die Alarmanlage bzw. der ausgelöste Alarm kann mit einer 3-Tastenkombination (drei Tasten in einer bestimmten Reihenfolge mit fünf Tastendrücken, z. B. 2x Taste 1, 1x Taste 2, 1x Taste 3, 1x Taste 1) deaktiviert werden.

Weitere Anforderungen:

- langsames Blinken, wenn scharf, schnelles Blinken bei Alarm
- die Empfindlichkeit der Alarmanlage kann über ein Poti eingestellt werden

Erweiterungen:

- Man kann die Uhrzeit und Dauer der Scharfstellung programmieren (nicht im Code hard kodiert sondern per Einstellungsmöglichkeiten mit beliebigem „Interface“).
- Aktivieren/Deaktivieren alternativ mit RFID-Chip
- Einstellungsmöglichkeiten bei mit Tastenfernbedienung

Thema 3: CO₂-Lüfter

Anforderungen:

Eine bereits vorhandene CO₂-Ampel soll „angezapft“ werden. Wenn der Zustand der Ampel hochspringt, soll sich ein Lüfter leicht drehen. Wenn der Zustand der Ampel weiter hochspringt, soll sich ein Lüfter schneller drehen. Springt der Zustand wieder eine Stufe runter, soll sich der Lüfter entsprechend weniger drehen oder anhalten. Die Schaltstufen für mehr oder weniger Lüfterdrehung sind per Hysterese umzusetzen.

Hysterese-Beispiel: bei Überschreiten von 500 ppm schaltet der Lüfter von Stufe 0 auf Stufe 1, aber erst bei Unterschreiten von 300 ppm schaltet der Lüfter von Stufe 1 auf Stufe 0 (Hinweis: die Zahlenwerte sind willkürlich und nicht praxisrelevant. Sie Spannungs- oder Stromstufen passend zu den CO₂-Ampel-Stufen sind im Projekt zunächst zu ermitteln).

Es handelt sich um diese CO₂-Ampel: <https://www.mb-systemtechnik.de/de/CO2-Messung/CO2-Ampel.html> (siehe technische Daten).

Hinweis: Ein Motor kann nicht direkt an einem Arduino betrieben werden!

Thema 4: Ampelanlage

Anforderungen:

Eine Ampelanlage besteht aus einer Fahrzeugampel (rot, gelb, grün) und einer Fußgängerampel:

- Beim Start der Anlage ist die Fahrzeugampel grün, die Fußgängerampel rot.
- Per Taster kann ein Fußgänger eine Grün-Phase für die Fußgänger und eine Rot-Phase für Fahrzeuge auslösen.
- Vor der Rot-Phase für Fahrzeuge hat die Fahrzeugampel eine Gelb-Phase.
- Die Gelb-Phase dauert zwei Sekunden.
- Eine Rot-Phase für Fahrzeuge dauert immer 15 Sekunden.
- Nach der Rot-Phase für Fahrzeuge erfolgt die Rot-Gelb-Phase (drei Sekunden) und danach wieder die Grün-Phase.
- Die Grün-Phase für Fußgänger beginnt sofort mit der Rot-Phase für Fahrzeuge.
- Die Rot-Phase für Fußgänger beginnt drei Sekunden vor der Rot-Gelb-Phase für Fahrzeuge.
- Die Gelb-Phase für Fahrzeuge beginnt zehn Sekunden nach Drücken des Fußgänger-Tasters, wenn die Grün-Phase der Fahrzeuge mindestens 30 Sekunden andauerte
- War die Grün-Phase für Fahrzeuge weniger als 30 Sekunden, beginnt die Gelb-Phase nach Drücken des Fußgänger-Tasters und wenn die Grün-Phase mindestens 30 Sekunden andauerte hatte.
- Ein Näherungssensor soll prüfen, ob sich Fahrzeuge über eine bestimmte Linie in der Fahrzeug-Rot-Phase bewegen. In dem Fall soll eine Lampe aufblitzen.

Erweiterung 1: Die Ampel soll für Sehbenachteiligte erweitert werden. Wenn die Ampel für Fußgänger auf rot steht, soll alle zwei Sekunden ein Ton („tick“) erfolgen, wenn die Fußgänger grün haben, soll jede halbe Sekunde ein Ton erfolgen. Für Wartungszwecke kann man mit einem Taster den Ton aus- bzw. einschalten.

Erweiterung 2 (sehr schwer): Beim Aufblitzen der Lampe soll ein Foto des die Linie überschreitenden Fahrzeuges irgendwie aufgenommen und irgendwo gespeichert werden

Fach: LF7

Thema: Arduino-Grundlagen

Thema 5: Küchenuhr / Timer

Eine Küchenuhr soll eingestellt werden und nach dem Starten sekundenweise runterzählen. Die Anzeige ist daher immer „hh:mm:ss“.

Beispiel: „01:20:12“ = noch eine Stunde, 20 Minuten und 12 Sekunden.

Die maximal einstellbare Dauer soll 11 Stunden und 59 Minuten sein.

Nach Ablauf der Zeit soll die Küchenuhr einen „dezenten“ Alarm abspielen (z. B. den Alexa-Alarmton). Man kann den Alarm stoppen oder auf Snooze drücken. Snooze verlängert die Zeit nochmals um 5 Minuten, um danach wieder in den Alarmzustand zu kommen.

Eine Küchenuhr soll ein Display und vier Tasten haben, deren Funktionen mehrfach belegt sind.

Die Küchenuhr kann sich in verschiedenen Zuständen befinden. Andere als die genannten Funktionalitäten (=Tastendrucke) dürfen in den Zuständen zu keiner Reaktion führen. Die Funktionalitäten ergeben sich aus der Zustandstabelle.

Tab. 1: Zustandstabelle

Zustand	Taste	Drückdauer	Reaktion
aus	plus	kurz	Zeiteinstellung: Zeit zählt um eine Minute hoch
aus	plus	lang	Zeiteinstellung: Zeit zählt um ein zehn Minuten hoch
aus	minus	kurz	Zeiteinstellung: Zeit zählt um eine Minute runter
aus	minus	lang	Zeiteinstellung: Zeit zählt um ein zehn Minuten runter
aus	Start/Stop	kurz	Zeit läuft im Sekundentakt runter, Zustand wird „tickend“
aus	Start/Stop	lang	Zeiteinstellung: Reset auf „00:00“
tickend	Start/Stop	kurz	Uhr wird angehalten (nicht zurückgesetzt), Zustand wird „wartend“
tickend	-	-	Zeit läuft im Sekundentakt runter. Wenn Restzeit 00:00:00, dann wird der Zustand „Alarm“
wartend	Start/Stop	kurz	Zeit läuft weiter runter im Sekundentakt
wartend	Start/Stop	lang	Anzeige geht auf „00:00“, Zustand geht auf „aus“
Alarm	Start/Stop	kurz	Anzeige geht auf „00:00“, Zustand geht auf „aus“
Alarm	Snooze	kurz	Anzeige geht auf „05:00:00“, Zustand geht auf „tickend“