

Eine tastengesteuerte Ampelanlage

Stellt euch vor, ihr gehörtet zum Verkehrsausschuss einer ländlichen Gemeinde, der ein Problem mit einer schmalen Straße in einem unübersichtlichen Gelände hat: Dort kommt es relativ häufig zu Unfällen, weil viel zu schnell gefahren wird und so bei Gegenverkehr nicht mehr rechtzeitig reagiert werden kann.

Ihr habt schon angeregt, Ampeln zu installieren, aber der Bürgermeister befürchtet, dass es damit zu Staus kommen könnte, weil das Verkehrsaufkommen bei Arbeitsbeginn anders ist als bei Arbeitsende. Die Straße wird richtungsmäßig zu verschiedenen Zeiten unterschiedlich stark befahren.

Ihr erklärt, dass das trotzdem kein Problem sei, weil sich eure Ampeln darauf einstellen könnten. Das kann sich der Bürgermeisten nicht recht vorstellen und ihr plant, ihn mithilfe eines funktionstüchtigen Modells von der Idee zu überzeugen.

Die Aufgabe

Das Ziel besteht darin, an jedem Ende des unübersichtlichen Straßenabschnitts eine Dreifarben-Verkehrsampel aufzustellen, die den Verkehr so überwacht, dass er immer in nur eine Richtung fließen kann. Sensoren an den Enden des Engpasses erkennen, wenn ein Auto vor einer roten Ampel anhält. Sie senden einen Impuls an das Steuergerät, das Umschalten zu veranlassen: Die Ampel,

- die Grün zeigt, springt um über Gelb auf Rot und
- die Rot zeigt, springt um über Rot/Gelb auf Grün.

Das Umschalten darf immer nur von der Seite aus möglich sein, an der die Ampel Rot zeigt. Wenn kein Verkehr herrscht, wartet die Ampel auf die nächste Anforderung.

Hinweise

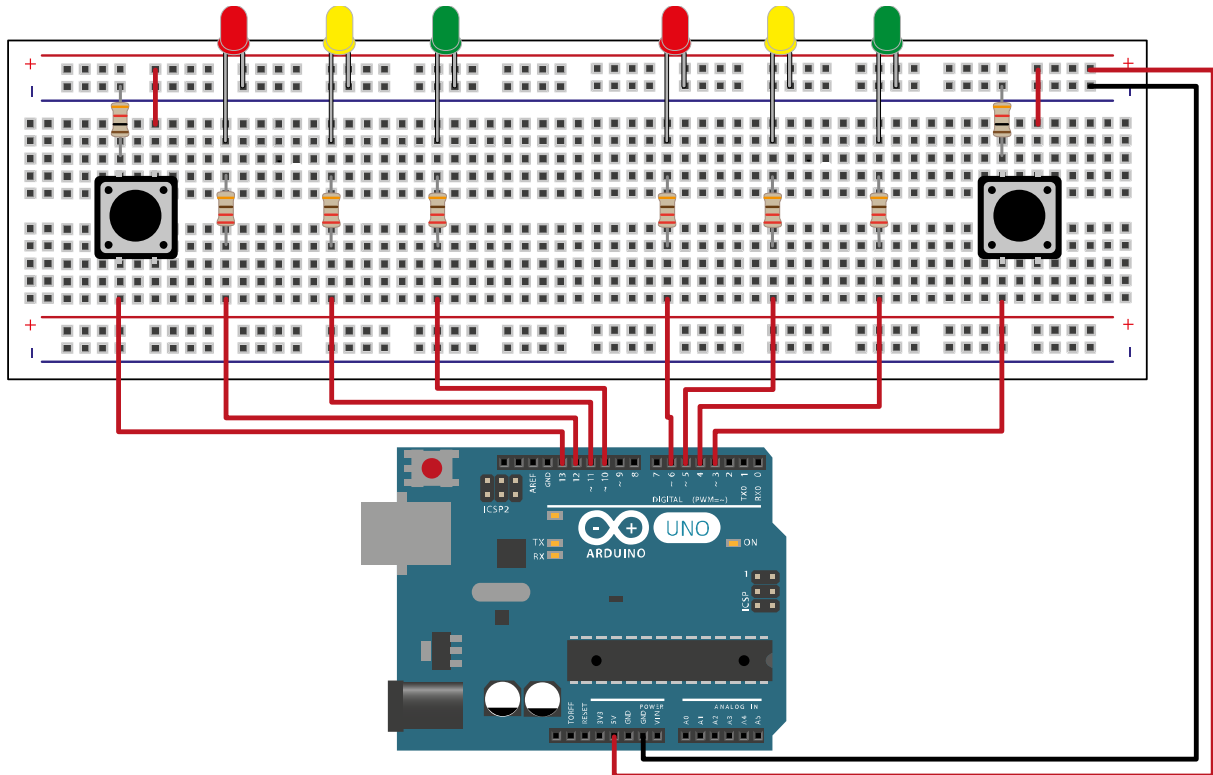
Um die Sensoren an den beiden Enden des Engpasses zu simulieren, bekommt ihr zwei Drucktaster. Zu jeder Ampel gehört eine rote, eine gelbe und eine grüne LED. Zu Beginn soll das System den Verkehr für eine bestimmte Richtung Grün anzeigen.

Wenn sich ein Fahrzeug der Brücke nähert (was durch Drücken der Taste simuliert wird) und die Ampel rot ist, schaltet das System die Ampel auf der gegenüberliegenden Seite von Grün über Gelb auf Rot um. Dann wartet sie einen festgelegten Zeitraum, damit Fahrzeuge, die sich bereits auf dem Engpass befinden, den Bereich verlassen können. Danach blinkt das gelbe Licht auf der Seite des wartenden Fahrzeugs, um dem Fahrer zu signalisieren, dass er sich vorausschauend auf den Start vorbereiten soll. Wenn dann die Ampel auf Grün schaltet und dem Fahrer keine weiteren Fahrzeuge entgegenkommen, hat er freie Fahrt. Die Ampel zeigt so lange Grün, bis sich ein Fahrzeug von der anderen Seite nähert und sich der Vorgang wiederholt.

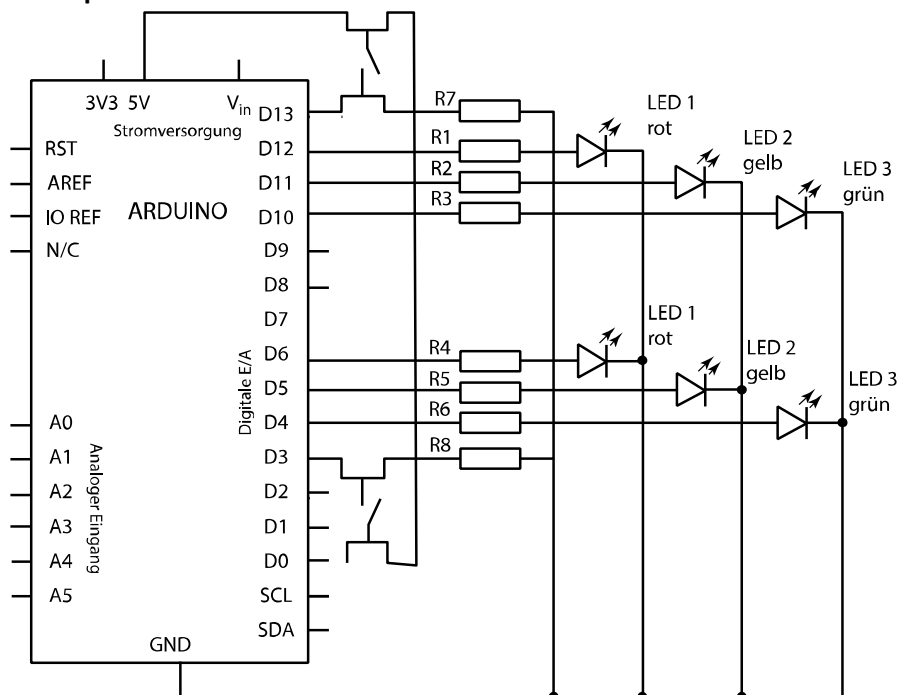
Benötigte Materialien

- Computer mit der Arduino-Software
- Arduino UNO
- USB-Kabel
- zwei rote LEDs
- zwei gelbe LEDs
- zwei grüne LEDs
- sechs 560-Ω-Widerstände
- zwei 10-kΩ-Widerstände
- zwei Drucktaster
- eine Steckplatine (Breadboard)
- Steckbrücken

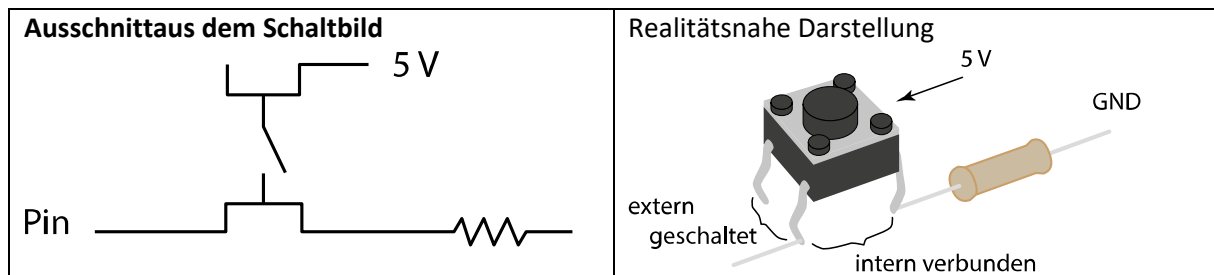
Lösung



Schaltplan



Widerstände $R_1 - R_6$: 560 Ohm, R_7 und R_8 : 10 Kiloohm



Programmiercode mit Erklärungen

// Zunächst werden die Pins festgelegt, mit denen die Taster und LEDs verbunden sind:

```
#define ersterKnopf 3
#define zweiterKnopf 13
#define zweiRot 6
#define zweiGelb 5
#define zweiGruen 4
#define einsRot 12
#define einsGelb 11
#define einsGruen 10
```

#define pausenZeit 500 // Hier wird der Zeittakt festgesetzt, mit der die gelbe LED blinken soll.

boolean richtungEins = true; /* Der Befehl "boolean" hält zwei Möglichkeiten vor, hier die beiden Richtungen des Verkehrsflusses, von der nur eine zur gleichen Zeit "wahr" sein darf, um Unfälle zu vermeiden. */

int flussZeit = 10000; // Zeitrahmen, der es einer bestimmten Anzahl Autos ermöglicht, den Engpass zu durchfahren.

int Wartezeit = 2000; /* Das Umspringen der Ampel darf nicht zu schnell erfolgen, damit sich ein Fahrer darauf einstellen kann. */

```
void setup() {
  // Im Folgenden werden die digitalen E/A-Pins eingerichtet.
```

```
  pinMode(ersterKnopf, INPUT);
  pinMode(zweiterKnopf, INPUT);
  pinMode(zweiRot, OUTPUT);
  pinMode(zweiGelb, OUTPUT);
  pinMode(zweiGruen, OUTPUT);
  pinMode(einsRot, OUTPUT);
  pinMode(einsGelb, OUTPUT);
  pinMode(einsGruen, OUTPUT);
```

// Hier wird der Anfangszustand der festgelegt.

```
digitalWrite(zweiRot, LOW);
digitalWrite(zweiGelb, LOW);
digitalWrite(zweiGruen, HIGH);
digitalWrite(einsRot, HIGH);
```

```

digitalWrite(einsGelb, LOW);
digitalWrite(einsGruen, LOW);
}
void loop() {

if ( digitalRead(ersterKnopf) == HIGH ) { // Grün wird angefordert.
if ( richtungEins != true ) { // Das gelingt nur, wenn die Ampel Rot zeigt.
richtungEins = true; // Die vorherige Verkehrsrichtung ("richtungEins = false") wird gewechselt.
delay(flussZeit); // Eine bestimmte Zeit wird gewartet, ...
digitalWrite(einsGruen, LOW); // ... ehe die Gegenampel von Grün über Gelb auf Rot umschaltet.
digitalWrite(einsGelb, HIGH);
delay(Wartezeit);
digitalWrite(einsGelb, LOW);
digitalWrite(einsRot, HIGH);
delay(Wartezeit);
for ( int a = 0; a < 5; a++ ) { // Die Blinkzeit wird festgelegt ...
digitalWrite(zweiGelb, LOW); // ... und die gelbe LED beginnt entsprechend zu blinken.
delay(pausenZeit);
digitalWrite(zweiGelb, HIGH);
delay(pausenZeit);
}
digitalWrite(zweiGelb, LOW);
digitalWrite(zweiRot, LOW);
digitalWrite(zweiGruen, HIGH); /* Dann springt die Ampel auf Grün um. Somit zeigt eine Ampel
Grün, die andere Rot. */
}
}
if ( digitalRead(zweiterKnopf) == HIGH ) { // Dieser Zustand wird erst wieder geändert, wenn von
der Rot-Seite Grün angefordert wird. Knopf gedrückt wird, ...
if ( richtungEins == true ) { //Das gelingt natürlich nur, wenn auch der Gegenverkehr freie Fahrt
hat ...
richtungEins = false; // ... und die vorherige Verkehrsrichtung ("richtungEins = true")dadurch
gewechselt wird.
delay(flussZeit); // Eine bestimmte Zeit wird gewartet, ...
digitalWrite(zweiGruen, LOW); // ... ehe die Gegenampel von Grün über Gelb auf Rot umschaltet.
digitalWrite(zweiGelb, HIGH);
delay(Wartezeit); // Dauer der Gelb- ...
digitalWrite(zweiGelb, LOW);
digitalWrite(zweiRot, HIGH);
delay(Wartezeit); // ... und der Rotphase.
for ( int a = 0; a < 5; a++ ) { // Die Blinkzeit wird erneut festgelegt, ...
digitalWrite(einsGelb, LOW); // ... die gelbe LED beginnt entsprechend zu blinken ...
delay(pausenZeit);
digitalWrite(einsGelb, HIGH);
delay(pausenZeit);
}
}
}
}

```

```
digitalWrite(einsGelb, LOW);  
digitalWrite(einsRot, LOW);  
digitalWrite(einsGruen, HIGH); // und die hiesige Ampel schaltet auf Grün um.  
}  
}  
}
```