

Eine tastengesteuerte Ampelanlage

Stellt euch vor, ihr gehörtet zum Verkehrsausschuss einer ländlichen Gemeinde, der mit der Schulwegsicherung befasst ist. Da viele Schulkinder eine stark befahrene Straße überqueren müssen, steht zur Debatte, eine „schlafende Ampel“ aufzustellen. Da nur wenige Ausschussmitglieder sich darunter etwas vorstellen können, schlägt ihr vor, bei der nächsten Sitzung deren Funktion an einem Modell verständlich zu machen.

Aufgabe

Das Modell soll folgende Funktionen erfüllen:

- Die Ampelanlage ist im Normalfall abgeschaltet
- Will eine Person die Straße überqueren, kann sie auf Tastendruck die Anlage in Betrieb nehmen
- Der Tastendruck setzt das Fußgängerlicht auf Rot und – mit einiger Verzögerung – die Ampel für den Straßenverkehr auf Grün
- Die Grünphase bleibt eine Weile bestehen, ehe das Licht über Gelb nach Rot wechselt
- Erst wenn die Verkehrsampel einige Zeit auf Rot steht, wird die Fußgängerampel auf Grün gesetzt.
- Die Grünphase der Ampel ist so bemessen, dass auch gehbehinderte Personen die Straße sicher überqueren können. Danach schaltet die Ampelanlage ab.

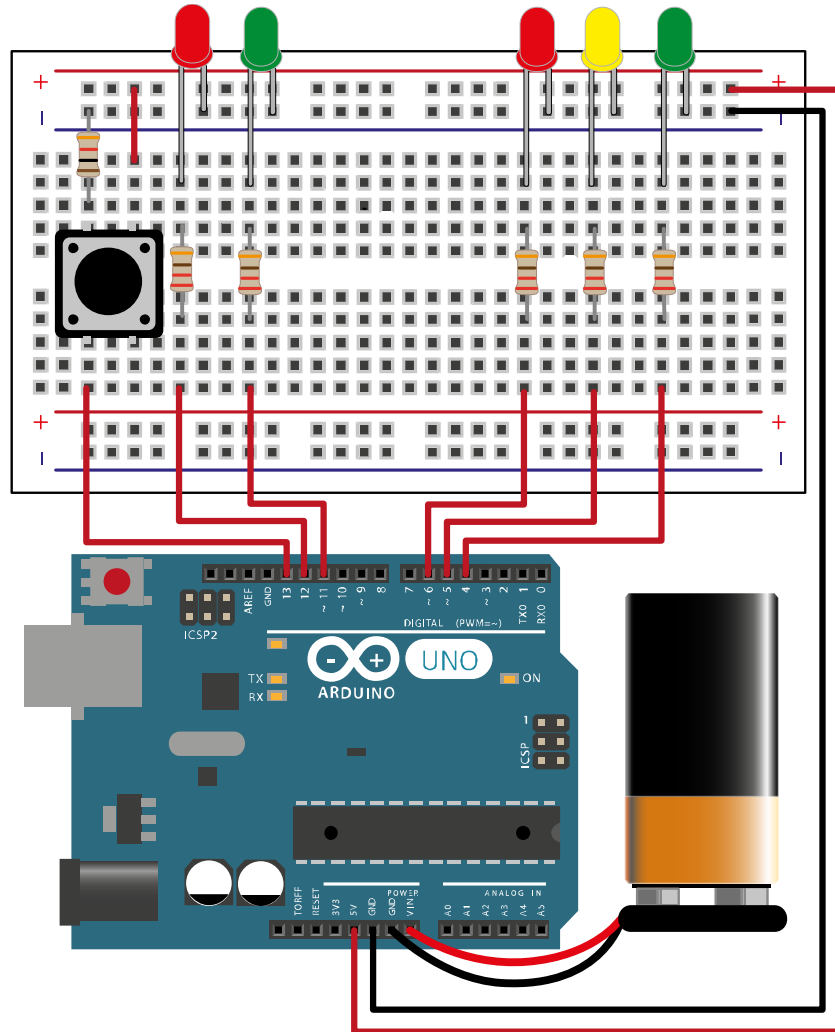
Hinweis

Zur Fußgängerampel gehören eine rote und eine grüne LED und ein Taster, zur Verkehrsampel eine rote, eine gelbe und eine grüne LED. Mit dem Taster wird die Ampelanlage in Betrieb genommen

Benötigte Materialien

- Computer mit der Arduino-Software
- Arduino Mikrocontroller-Board
- USB-Kabel
- zwei rote LEDs
- zwei grüne LEDs
- eine gelbe LED
- fünf 220-Ω-Widerstände für die LEDs
- ein 10-kΩ-Widerstand den Taster
- ein Drucktaster
- eine Steckplatine (Breadboard)
- Steckbrücken

Lösung



Nach der Programmierung kann die USB Verbindung gelöst und die Stromversorgung von einem 9V Block übernommen werden, entweder über einen Clip mit rundem Normstecker oder – wie hier gezeigt – über Vin und GND an der Buchsenleiste.

Programmiercode mit Erklärungen

```
// Anschlüssen und der Abfrage sinnstiftende Namen geben
int taster = 13;
int tasterstatus = 0;
int einsRot = 12;
int einsGruen = 11;
int zweiRot = 6;
int zweiGelb = 5;
int zweiGruen = 4;

void setup() {
  // Einrichten der Anschlüsse
  pinMode(taster, INPUT);
  pinMode(einsRot, OUTPUT);
  pinMode(einsGruen, OUTPUT);
  pinMode(zweiRot, OUTPUT);
  pinMode(zweiGelb, OUTPUT);
  pinMode(zweiGruen, OUTPUT);
}

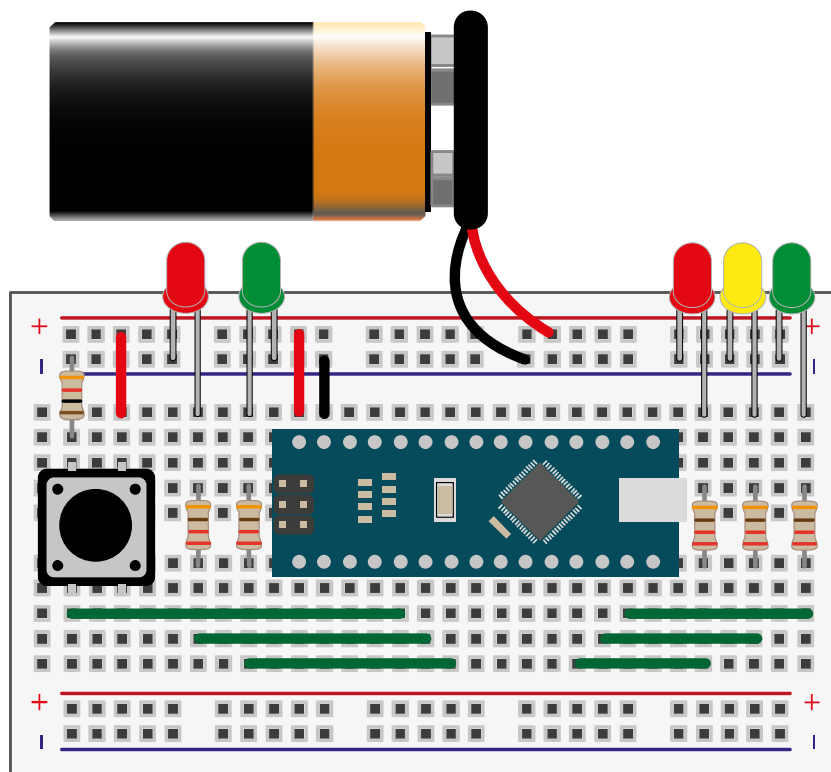
void loop() {
```

```

tasterstatus = digitalRead(taster);
if (tasterstatus == HIGH) { // Grün wird angefordert.
    digitalWrite(einsRot, HIGH); //Fußgängerampel schaltet auf Rot
    delay(3000); // 3 Sekunden wird gewartet, ...
    digitalWrite(zweiGruen, HIGH); // ... ehe die Verkehrsampel auf Grün schaltet.
    delay(5000); // 5 Sekunden Grün
    digitalWrite(zweiGruen, LOW);
    digitalWrite(zweiGelb, HIGH);
    delay(3000); // 3 Sekunden Gelb
    digitalWrite(zweiGelb, LOW);
    digitalWrite(zweiRot, HIGH);
    delay(5000); // 5 Sekunden warten ...
    digitalWrite(einsRot, LOW);
    digitalWrite(einsGruen, HIGH); // ... bis die Fußgängerampel auf Grün springt
    delay(15000); // 15 Sekunden Grün ...
    digitalWrite(einsGruen, LOW); // ... dann schaltet die Anlage ab
    digitalWrite(zweiRot, LOW);
}
else {
    digitalWrite(einsRot, LOW);
    digitalWrite(einsGruen, LOW);
    digitalWrite(zweiRot, LOW);
    digitalWrite(zweiGelb, LOW);
    digitalWrite(zweiGruen, LOW);
}
}

```

Verwendet man statt des Arduino UNO einen Nano, kann das Ampel-Modell wesentlich kompakter realisiert werden.



Anders als beim UNO sind hier bei der Fußgängerampel der Taster mit D2, die rote LED mit D3 und die grüne mit D4 verbunden. Bei der 3-Lichter-Ampel wird die rote LED über D9, die gelbe über D10 und die grüne über D11 gesteuert. Der Programmcode muss dementsprechend angepasst werden. Nach der Programmierung kann auch hier ein 9V Block die Stromversorgung übernehmen: Plus an VIN, Minus an GND.