

Daten und Informationen zwischen Arduinos senden

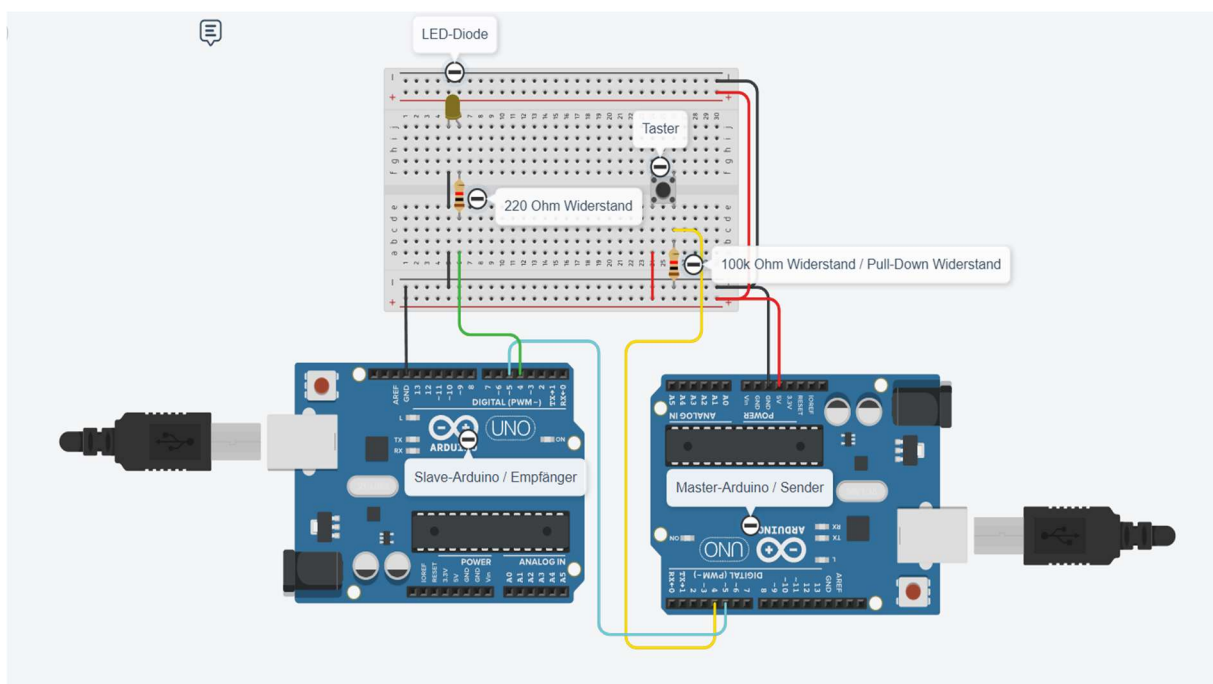
Arbeitsauftrag 1:

Erstelle eine Schaltung mit dem dazugehörigen Programm, um ein Signal von einem Arduino zum anderen zu senden. Dabei wird eine LED eingeschalten.

Materialien:

- 2x Arduino
- 1x LED
- 1x Taster
- 1x 100k Ohm Widerstand
- 1x 220 Ohm Widerstand
- Mehrere Kabel

Aufbau:



Code:

Master / Sender:

```
const int taster = 4;           //Pins für Taster und Übertragungskabel
const int cable = 5;

void setup() {
  pinMode(4, INPUT);           //Pinmodes für Taster und Übertragungskabel
  pinMode(5, OUTPUT);
}

void loop() {
  if (digitalRead(taster) == HIGH) { //Fragt ab, ob der Taster gedrückt ist
    digitalWrite(cable, HIGH);      //wenn ja, Übertragungskabel = HIGH
  }
  else {
    digitalWrite(cable, LOW)        //wenn nein, Übertragungskabel = LOW
  }
}
```

Slave / Empfänger:

```
const int cable = 5;           //Pins für Led und Übertragungskabel
const int LED = 4;

void setup() {
  pinMode(4, OUTPUT);          //Pinmodes für Led und Übertragungskabel
  pinMode(5, INPUT);
}

void loop() {
  if (digitalRead(cable) == HIGH) { //Fragt ab ob der Übertragungskabel HIGH ist
    digitalWrite(LED, HIGH);        //wenn ja, stellt den Pin der Led auf HIGH
  }
  else {
    digitalWrite(LED, LOW);         //wenn nein, stellt den Pin der Led auf LOW
  }
}
```

Funktionsweise:

Dieses Projekt verwendet einen Taster, um ein Signal zu erkennen. Der Master-Arduino misst ständig die Situation des Tasters. Wenn der Taster gedrückt wird, also HIGH ist, sendet der Master via Kabel ein digitales Signal an den Slave-Arduino. Wenn dieser das Signal erhält, schaltet er die LED, welche an ihm angeschlossen ist, ein. Sobald der Taster wieder ausgelassen wird, also LOW ist, wird das Signal unterbrochen und die LED wird wieder ausgeschalten.

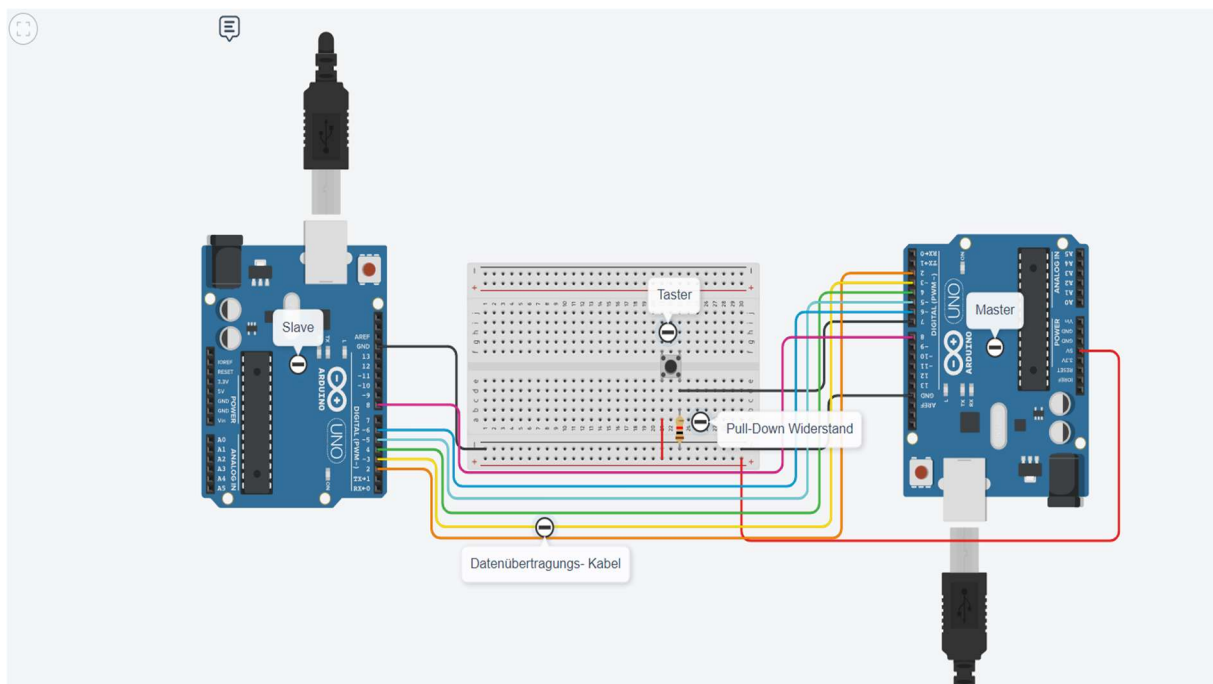
Arbeitsauftrag 2:

Erstelle eine Schaltung mit dem dazugehörigen Programm, um ein Wort von einem Arduino zum anderen zu senden.

Materialien:

- 2x Arduino Uno
- 1x Taster
- 1x 100k Ohm Widerstand
- Mehrere Kabel

Aufbau:



Code:

Master / Sender:

```
int L1 = 2; //Pins für Leitungen
int L2 = 3;
int L3 = 4;
int L4 = 5;
int L5 = 6;

char wort[] = "go, touch some grass! "; //Wort eingeben, welches am anderen Arduino ausgegeben wird
int size = 1; //Variabel size, um die Richtige Stelle des Strings auszulesen
int resize = 22; // Anzahl der Buchstaben des Wortes (inc Leerzeichen)

void setup() {
    pinMode (2, OUTPUT); //PinModes für Led's
    pinMode (3, OUTPUT);
    pinMode (4, OUTPUT);
    pinMode (5, OUTPUT);
    pinMode (6, OUTPUT);
    pinMode (8, OUTPUT);
    pinMode (7, INPUT); //PinMode für Taster
}

void loop() {

    if (digitalRead(7) == HIGH) { //Ist der Taster gedrückt? Wenn ja,:
        digitalWrite(8, HIGH); //Synchronisierungskabel auf HIGH

        if (wort[size] == 'a') { //Wenn wort an der Stelle size a ist:
            digitalWrite(L1, HIGH); //Diese Kombination ausgeben
            digitalWrite(L2, HIGH);
            digitalWrite(L3, HIGH);
            digitalWrite(L4, HIGH);
            digitalWrite(L5, LOW);
        }

        if (wort[size] == 'b') { //Wenn wort an der Stelle size b ist:
            digitalWrite(L1, LOW); //Diese Kombination ausgeben
            digitalWrite(L2, LOW);
            digitalWrite(L3, LOW); //Dies für jeden Buchstaben wiederholen
            digitalWrite(L4, LOW); //Dabei immer eine andere Kombination aus
            digitalWrite(L5, HIGH); //HIGH und LOW verwenden
        }

        if (wort[size] == 'c') {
            digitalWrite(L1, LOW);
            digitalWrite(L2, LOW);
            digitalWrite(L3, LOW);
            digitalWrite(L4, HIGH);
            digitalWrite(L5, LOW);
        }
    }
}
```

```

    if (wort[size] == ' ') {
        digitalWrite(L1, HIGH);
        digitalWrite(L2, HIGH);
        digitalWrite(L3, HIGH);
        digitalWrite(L4, HIGH);
        digitalWrite(L5, HIGH);
    }

    size = size + 1;
    if (size == resize) {
        size = 0;
    }
    delay(500);
}
if (digitalRead(7) == LOW) {
    digitalWrite(L1, LOW);
    digitalWrite(L2, LOW);
    digitalWrite(L3, LOW);
    digitalWrite(L4, LOW);
    digitalWrite(L5, LOW);
}

} //Ende des Void-Loop

```

Slave:

```

int L1 = 2; //Pin's für Leitungen
int L2 = 3;
int L3 = 4;
int L4 = 5;
int L5 = 6;
int starter = 8; //Pin des Synchronisierungskabel

void setup()
{
    pinMode(2, INPUT); //PinModes für Leitungen + Synchronisierungskabel
    pinMode(3, INPUT);
    pinMode(4, INPUT);
    pinMode(5, INPUT);
    pinMode(6, INPUT);
    pinMode(8, INPUT);
    Serial.begin(9600); //Frequenz des Seriellen Monitors (9600 Baud)
}

void loop() {
    if (digitalRead(starter) == HIGH) {
        if (digitalRead(L1) == LOW) {
            if (digitalRead(L2) == LOW) {
                if (digitalRead(L3) == LOW) {
                    if (digitalRead(L4) == LOW) {
                        if (digitalRead(L5) == LOW) {
                            //Passiert nichts, da es keine gültige Kombination ist
                        }
                        else if (digitalRead(L5) == HIGH) {
                            //Wenn Leitung 5 hingegen HIGH ist:
                            Serial.println('b'); //Wird b im Seriellen Monitor ausgegeben
                        }
                    }
                    else if (digitalRead(L4) == HIGH) {
                        //Wenn Leitung 4 hingegen HIGH ist:
                        if (digitalRead(L5) == LOW) {
                            //Wenn Leitung 5 LOW ist:
                            Serial.println('c'); //Wird c ausgegeben
                        }
                        else if (digitalRead(L5) == HIGH) {
                            //Wenn Leitung 5 hingegen HIGH ist:
                            Serial.println('d'); //Wird d ausgegeben
                        }
                    }
                }
            }
        }
        //Wiederholen bis alle Kombinationen abgefragt sind (64 Abfragen) +500 delay
    }
    //Ich habe mich dazu entschieden den Rest des Codes nicht
    //zu kommentieren, da es 250 Zeilen mit identischem Code ist.
    //Die Datei des Codes ist im Anhang da sie zu viele Seiten
    //Im Protokoll einnehmen würde.
    else if (digitalRead(L3) == HIGH) {
        if (digitalRead(L4) == LOW) {
            if (digitalRead(L5) == LOW) {
                Serial.println('e');
            }
        }
    }
}

```

Funktionsweise:

In diesem Projekt kann man alle 26 Buchstaben des Alphabets ausgeben und noch fünf Sonderzeichen (Leerzeichen, Punkt, Komma, Ausrufezeichen und Fragezeichen). Somit kann man fast jeden Satz ausgeben, den man möchte. Jedem Buchstaben und jedem Sonderzeichen wird eine Kombination aus 0 und 1 zugeordnet. Mit diesen Werten werden dann 31 verschiedene 5-Stellige Kombinationen erstellt, welche an den Empfänger per Kabel übermittelt werden. Dieser liest die Signale ein und fragt die Kombination mit insgesamt 64 If - Verzweigungen ab. Dann gibt er den ermittelten Buchstaben im Seriellen Monitor aus. Eine Verzögerung von 500 ms ist eingebaut, um die Übertragung zu garantieren und Fehler beim Erkennen zu vermeiden.

Fehlerquellen:

Viele verschachtelte If-Verzweigungen beim Empfänger, daher ist es wichtig jede Klammer an der richtigen Stelle zu schließen.

Arduinos laufen nicht gleichzeitig, daher ein zusätzliches Synchronisierungskabel.

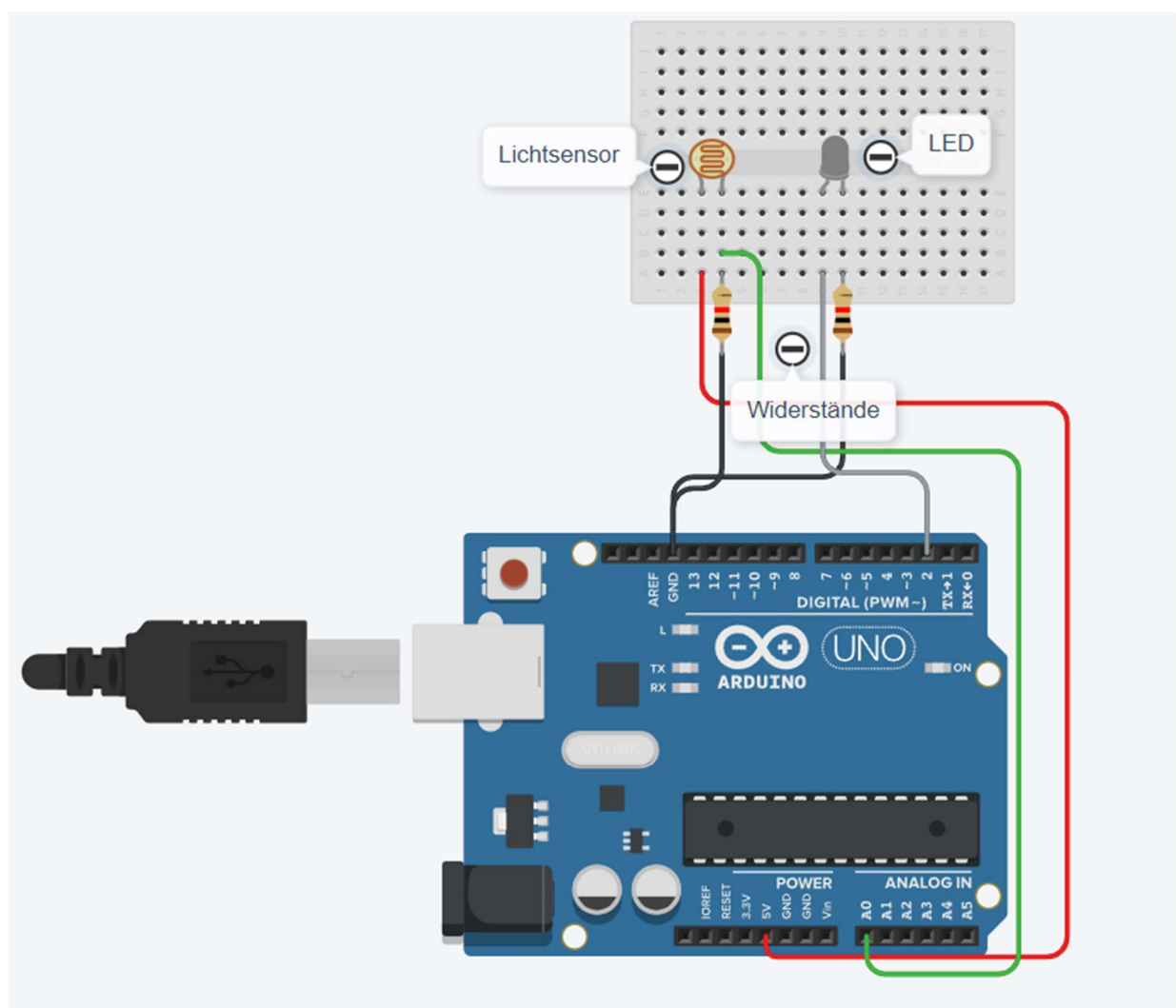
Arbeitsauftrag 3:

Erstelle eine Schaltung mit dem dazugehörigen Programm, um ein Wort kabellos von einem Arduino zum anderen zu senden.

Materialien

- Arduino Uno
- Photowiderstand
- Led
- 220 Ohm Widerstand
- 100k Ohm Widerstand
- Verschiedene Kabel

Aufbau:



Code:

```

const int photocell = A0;           //pin für Lichtsensor
const int l = 2;                     //pin für LED

int slave1 = 0;                      //Speicherorte des Slaves
int slave2 = 0;
int slave3 = 0;
int slave4 = 0;
int slave5 = 0;

const char wort[] = "e";             //Wort eingeben
int size = 0;                        //start[size] des Wortes
const int resize = 1;                //Anzahl der Buchstaben im Wort (inklusive Leerzeichen)

void setup() {
  pinMode(photocell, INPUT);          //pin Photocell
  pinMode(l, OUTPUT);                 //pin LED

  Serial.begin(9600);                 //Serieller Monitor
}

void loop() {                         //void loop()
  if (wort[size] == 'a') {             //wenn wort an der stelle size = a;
    digitalWrite(l, HIGH);             //Led auf den gewählten Status setzen
    slave1 = slavefunction();           //Funktion slavefunction() aufgerufen und der zurückgegebene Wert in slave1 speichern.
    delay(500);                         //delay von 500 um Fehler zu vermeiden
    digitalWrite(l, HIGH);             //Led auf den gewählten Status setzen
    slave2 = slavefunction();           //Funktion slavefunction() aufgerufen und der zurückgegebene Wert in slave2 speichern.
    delay(500);                         //delay von 500 um Fehler zu vermeiden
    digitalWrite(l, HIGH);             //Led auf den gewählten Status setzen
    slave3 = slavefunction();           //Funktion slavefunction() aufgerufen und der zurückgegebene Wert in slave3 speichern.
    delay(500);                         //delay von 500 um Fehler zu vermeiden
    digitalWrite(l, HIGH);             //Led auf den gewählten Status setzen
    slave4 = slavefunction();           //Funktion slavefunction() aufgerufen und der zurückgegebene Wert in slave4 speichern.
    delay(500);                         //delay von 500 um Fehler zu vermeiden
    digitalWrite(l, LOW);              //Led auf den gewählten Status setzen
    slave5 = slavefunction();           //Funktion slavefunction() aufgerufen und der zurückgegebene Wert in slave5 speichern.
    delay(500);                         //delay von 500 um Fehler zu vermeiden
  }

  //Für jede Kombination wiederholen

  size = size + 1;                     //Size wird um 1 erhöht um nächste stelle des Wortes abzurufen
  if (size == resize) {                 //wenn size = size ist:
    size = 0;                           //size wird auf 0 gesetzt und Programm startet neu
  }
  ask();                               //Funktion ask() wird aufgerufen

  slave1 = 0;                          //variablen werden zurückgesetzt
  slave2 = 0;
  slave3 = 0;
  slave4 = 0;
  slave5 = 0;

}                                       //Ende des Void Loop()

```



```
int slavefunction() {                                //Funktion slavefunction()
  if (analogRead(photocell) > 1000) {                //Wenn der Wert des Lichtsensors > 1000 ist:
    return 1;                                         //Wird 1 zurückgeben
  }
  if (analogRead(photocell) < 1000) {                //Wenn der Wert < 1000 ist:
    return 0;                                         //Wird 0 zurückgeben
  }
}

void ask() {                                          //Funktion ask()
  if (slave1 == 0) {                                  //wenn slave1 = 0;
    if (slave2 == 0) {                                //wenn slave2 = 0;
      if (slave3 == 0) {                              //wenn slave3 = 0;
        if (slave4 == 0) {                            //wenn slave4 = 0;
          if (slave5 == 0) {                          //wenn slave5 = 0;
            //wird nichts ausgegeben da es keine Kombination ist
          }
          else if (slave5 == 1) {                      //wenn slave5 = 1;
            digitalWrite('b');                       //Wird b ausgegeben
            return;                                    //und zum Loop zurückgekehrt
          }
        }
      }
    }
    else if (slave4 == 1) {                            //wenn slave4 = 1;
      if (slave5 == 0) {                              //wenn slave5 = 0;
        digitalWrite('c');                           //wird c ausgegeben
        return;                                       //und zum Loop zurückgekehrt
      }
      else if (slave5 == 1) {                          //Die selben abfragen bis alle 32 Abgefragt sind
        digitalWrite('d');
        return;
      }
    }
  }
}
```

Funktionsweise:

Da ich nur einen Arduino zur Verfügung hatte habe ich die Programme zusammengefügt. Dabei übernehmen die Funktionen slavefunction() und ask() die Arbeit des Empfängers. Das Ergebnis ändert sich dabei nicht.

Eine Led wird in einer bestimmten Reihenfolge und einem delay von 500 ms zu blinken gebracht. Ein Photosensor liest die Werte in slavefunction() ein und weist ihr einen Wert zu (1 oder 0). Im void ask() wird diese Kombination dann abgefragt und ausgewertet. Die Ergebnisse werden dann im Seriellen Monitor ausgegeben. Dieses Programm hat eine sehr hohe Fehleranfälligkeit, da der Lichtsensor leicht vom Umgebungslicht beeinflusst werden kann.

Neue Befehle und Bauteile:

Funktionen: Mit ihnen lässt sich ein Projekt besser gliedern und Copy-Pasta vermeiden. Ein Beispiel im Code wäre: `slavefunktion()` im Arbeitsauftrag 3.

Arrays: Man kann zb einen bestimmten Buchstaben eines Wortes auslesen oder verändern. Man benutzt den befehl auch um PinModes vieler Pins festzulegen. Ein Beispiel im Code wäre: `wort[]` in Arbeitsauftrag 2 und 3.

Photosensor: Dieses Bauteil verändert seinen Widerstand abhängig davon wie viel Licht auf ihn fällt. Im Arbeitsauftrag 3 verwendet.