

## Teil 1: Erste Messungen mit dem GSR-Sensor

---

Ein Lügendetektor, auch als Polygraph bekannt, misst physiologische Reaktionen, um falsche Aussagen zu erkennen. Ein wichtiger Bestandteil ist der GSR-Sensor (Galvanic Skin Response), der den Hautwiderstand erfasst. Bei Stress, wie etwa beim Lügen, ändert sich die Schweißproduktion, was den elektrischen Widerstand der Haut verringert. Dieser Widerstandswert wird vom Sensor aufgezeichnet und analysiert, um festzustellen, ob der Proband die Wahrheit sagt oder nicht.

- ① Euer GSR-Sensor wird mit mehreren Kabeln verbunden. Verbindet diese Kabel folgendermaßen mit eurem Arduino:

| Kabelfarbe | Pin auf dem Arduino  |
|------------|----------------------|
| schwarz    | GND                  |
| rot        | 5V                   |
| gelb       | A0                   |
| weiß       | wird nicht verbunden |

- ② Ihr habt in dem Verzeichnis XXX eine Codevorlage gegeben. Kopiert diese und fügt sie in eurem Verzeichnis XXX ein.
- ③ In der Codevorlage habt ihr die Funktion `getDurchschnittlichenGSRWert(int n)` gegeben. Schreibt dafür den Programmcode, welcher das arithmetische Mittel von  $n$ -Lesedurchgängen berechnet und diesen zurück gibt. Zwischen den Lesevorgängen soll 5 Millisekunden gewartet werden.
- ④ Ruft jetzt in der `loop()` Funktion `getDurchschnittlichenGSRWert(10)` auf und speichert die zurückgegebene Zahl in einer Variable ab. Gebt diese Variable mit Hilfe `Serial.println()` aus.
- ⑤ Ladet das Programm nun auf den Arduino und nehmt den Widerstand von jedem Gruppenmitglied auf, indem ihr den Sensor an dem Zeigefinger und an dem Mittelfinger anschließt. Die Metallkontakte sollten dabei die Fingerkuppen berühren. Notiert die Werte in der Tabelle.

|             | Name | GSR-Wert |
|-------------|------|----------|
| Mitglied 1: |      |          |
| Mitglied 2: |      |          |
| Mitglied 3: |      |          |

Bei der Messung sollte auffallen, dass für jedes Gruppenmitglied unterschiedliche Werte rauskommen. Dies liegt daran, dass jeder Mensch eine andere Haut hat und somit den Strom unterschiedlich leitet. Für unseren Lügendetektor müssen wir am Anfang für jede Person einen Ruhewert kalibrieren, welcher dann verglichen wird, um eine Lüge festzustellen.

## Teil 2: Programmieren eines einfachen Lügendetektors

---

- ⑥ Schreibt den Code für die Funktion `kalibriereRuhewert()`. Diese soll für  $n=1000$  (Durchgänge) den durchschnittlichen GSR-Wert berechnen und in der globalen Variable `ruhewert` speichern. Rufe die Funktion einmalig in der Funktion `setup()` auf.

**Tipp**

Benutze die Funktion `getDurchschnittlichenGSRWert(int n)`.

Bei einer Lüge steigt der Stress und somit auch die Schweißproduktion. Dies erhöht die elektrische Hautleitfähigkeit und senkt den elektrischen Widerstand. Das bedeutet andererseits aber auch, dass in einer gestressten Situation fälschlicherweise eine Lüge angezeigt werden kann. Wir wollen nun eine grüne LED aufleuchten lassen, wenn die Person ruhig ist, bei leichter Unsicherheit soll eine gelbe LED leuchten und starkes Stresslevel wollen wir als Lüge ansehen, das heißt es soll eine rote LED leuchten.

- ⑦ **Achtung:** Trennt den Arduino vom Strom. Bevor ihr ihn wieder mit dem Strom verbindet, meldet ihr euch und lasst euren Aufbau von dem Lehrer überprüfen.  
Baut auf einem Steckbrett eine LED-Schaltung mit den drei LEDs rot, gelb und grün. Definiert die LEDs als globale Variablen und lege sie in `setup()` als Output-Pins fest.
- ⑧ Schreibt den Code für die Funktion `ueberpruefeLuege(int gsrWert)`. Wenn der übergebene `gsrWert` mehr als 40% unter dem Ruhewert ist, soll die rote LED leuchten. Wenn er mehr als 20% unter dem Ruhewert ist, soll die gelbe LED leuchten. Wenn beides nicht eintrifft, soll die grüne LED leuchten. Ruft die Funktion in der `loop` Funktion auf mit dem zuvor berechneten durchschnittlichen GSR-Wert.
- ⑨ Probiert den Lügendetektor nun aus, indem ihr euch gegenseitig mindestens fünf Fragen stellt. Seid dabei respektvoll miteinander. Wenn jemand eine Frage nicht beantworten will, darf er das sagen. Dies wird von den anderen anerkannt und es wird keiner zu einer Antwort gedrängt. Notiert in der folgenden Tabelle, wie gut der Lügendetektor die Lügen erkennt.

| mögliche Fälle                         | Anzahl |
|----------------------------------------|--------|
| Lüge richtig erkannt                   |        |
| Lüge erkannt, obwohl wahre Aussage     |        |
| Wahrheit richtig erkannt               |        |
| Wahrheit erkannt, obwohl gelogen wurde |        |

Ihr werdet feststellen, dass der Lügendetektor viele Fehler macht. In der Wissenschaft gibt es viele verschiedene Studien, welche verschiedene Ergebnisse zu der Genauigkeit von Lügendetektoren geben. Einige Studien berichten von einer Wahrscheinlichkeit zwischen 93%-98,5%, dass eine Lüge richtig erkannt wird. Andere Studien berichten aber auch, dass der Lügendetektor mit einer Wahrscheinlichkeit von 10%-45% einen Fehler macht, das heißt entweder eine Lüge als Wahrheit interpretiert oder im noch schlimmeren Fall die Wahrheit als Lüge interpretiert.

Diese Studien stammen aber von professionellen Lügendetektoren, welche noch weitere und genauere Sensoren haben. Zudem ist unsere Programm relativ einfach gehalten. Bei einem professionellen Lügendetektor wird zum Beispiel die Kalibrierung so vorgenommen, dass mehrere Fragen gestellt werden, bei denen man die Antwort schon weiß. Dabei werden normale und unangenehme Fragen gemischt, um die jeweilige Stresssituation mit zu berücksichtigen. Unser selbstgebastelter Lügendetektor hat somit eine noch deutlich höhere Fehleranfälligkeit.

## Zusatz - Verbesserung des Lügendetektors

---

- ⑩ Ändert in der Funktion `ueberpruefeLuege(int gsrWert)` die Verzweigung, so dass eine Lüge und eine Unsicherheit schneller oder später erkannt wird (Abweichung vom Ruwert). Wiederholt das Experiment aus Aufgabe 9 und notiert euch die Werte.

| mögliche Fälle                         | Anzahl |
|----------------------------------------|--------|
| Lüge richtig erkannt                   |        |
| Lüge erkannt, obwohl wahre Aussage     |        |
| Wahrheit richtig erkannt               |        |
| Wahrheit erkannt, obwohl gelogen wurde |        |

Vergleicht die Werte mit denen aus Aufgabe 9 und erläutert, welche Vor- und Nachteile die Änderung hat.

---

---

---

---

---

Bei einem Lügendetektor ist vor allem die Abweichung in kurzer Zeit entscheiden. Bei dem jetzigem Programm legen wir den Ruhewert einmal am Anfang fest. Dieser kann sich jedoch mit der Zeit ändern. Es kann sein, dass wir mit der Zeit mehr zur Ruhe kommen und somit der aktuelle GSR-Wert deutlich über dem Ruhewert ist. Wir erkennen mögliche Lügen somit nicht. Andererseits können wir mit der Zeit auch nervöser geworden sein. Der GSR-Wert ist somit dauerhaft unter dem Ruhewert. Leichte Abweichungen können also schon als Lüge gewertet werden.

Eine Lösung, um das Problem zu beseitigen, ist, dass wir immer die letzten 10 Werte betrachten und schauen, wie stark der neue Wert davon abweicht.

⑪ Schreibt eine Funktion `ueberpruefeLuege2()`, welches diese Idee umsetzt.

**Tipps**

- Benutzt eine globale Variable, welche die letzten zehn Werte in einem Array abspeichert.
- Wenn der Wert einmal stark gefallen ist und dann konstant bleibt, schaltet sich wieder die grüne LED an. Die rote LED soll aber erst einmal für ein paar Sekunden leuchten, bevor es wieder auf grün springt.
- Ist die gelbe LED bei dieser Umsetzung noch sinnvoll?