Arbeitsplan 17.11.26:

Wir können nun einen Zweipunkt-Regler bauen!

1) Schreibe ein Programm, dass den Servo folgendermaßen auslenkt:
   wenn (if) Abstand < Mitte, Servo auf Horizontal – 5°,
   wenn (if) Abstand > Mitte, Servo auf Horizontal + 5°.
2) Speiche dein Programm unter „Zweipunktregelung“
3) Optimiere die Parameter deines Systems so, um den Ball möglichst nah an der Mitte der Wippe zu halten. Schaffst du es, den Ball zum Stoppen zu bringen? Welche Schwierigkeiten ergeben sich dabei?
4) Optimiere deine Regelung, indem du sie zu einer Mehrpunktregelung (3 oder mehr) erweiterst. Bist du zufrieden mit den Ergebnissen?

Unser Weg zum P-Regler:

Auf unserem Arbeitsblatt haben wir die Gleichung für eine Ursprungsgerade kennengelernt:

$$y(t) = k_P \cdot e(t)$$

| | |
|---|---|
| y: | Stellgröße |
| $k_P$: | Proportionalitätsfaktor |
| e: | Regeldifferenz |

1) Schreibe dir ein Programm, das dir die Regeldifferenz berechnet und diese in eine Variable (float?) speichert.
   Die Stellgröße soll nun mit der obigen Gleichung berechnet werden.
   Beachte: Diese ist positiv oder negativ um die Neutralstellung.
   Beachte auch: Begrenze (constrain) den Servo so, dass er nichts beschädigen kann.
   Hinweis: Ein guter Anfang-Proportionalitätsfaktor ist der Wert 0,2.

2) Variierte den Wert von kp, bist du nun mit dem Ergebnis zufrieden?
   (Drehe einen kurzen Film deiner Wippe, auch wenn du den Ball ein bisschen auslenkst)

**Einschub:** Spätestens hier lohnt es sich, im Programm etwas Ordnung zu schaffen und etwas über Unterprogramme zu lernen:

1) Benenne, falls nötig, alle Variablen so um, dass sie auch jemand anderes versteht und kommentiere deinen Code (// Wo wird was berechnet oder getan?)
2) Speichere deinen Code unter einem geeigneten Namen ab.
   (Z.B.: P-Regler-Kommentiert, dein Lehrer freut sich sehr über einen schönen Code, den er geschickt bekommt)
3) Bearbeite das Arbeitsblatt zu Unterprogrammen, dieses findest du in Onenote und itslearning)