Ultraschallproblem

- Lösungsansätze -

1. Internen CPU Timer benutzen
2. Zeitbegrenzung bei PulseIn, 600 microsekunden für Reichweite von 10cm, soll Zeit sparen

Statt: pulseIn(echo2Pin, HIGH);

Das hier: pulseIn(echo2Pin, HIGH, 600);

1. Um Zeit zu sparen Umrechnung von Microsekunden in cm zusammenfassen

(Dauer \* 0.034 / 2) wird durch Einführung von MWT zu : ( MWT/5 \* 0.034/2) kürzt sich zu (MWT \* 0.0034

1. Verschachtelte While Schleife für das Überprüfen der Entfernungen

While(…) {

Geradeausfahren()

While(Rechts USS){

Ausweichen()

}

While(Links USS){

Ausweichen()

}

}

(Quelltext stark gekürzt)

1. Damit die Entfernung nicht immer als Bedingung der Schleife überprüft werden muss, wird sie am Ende eines Jeden Durchgangs in einer Variable gemessen und gespeichert und diese immer wieder überprüft (-> sollte Zeit sparen), ergibt statt 4 Messungen nur noch eine Messung

Statt dem hier:

while ( (Rot() > 10 && Blau() > 13) && (USMLinks() > 5 || USMLinks() == 0) && (USMRechts() > 5 || USMRechts() == 0) ) {

 stepper1.setSpeed(-700.0);

 stepper2.setSpeed(700.0);

 stepper1.runSpeed();

 stepper2.runSpeed();

}

Das hier:

float EntfLinks = 10;

float EntfRechts = 10;

while ( (Rot() > 10 && Blau() > 13) && (EntfLinks > 5 || EntfLinks == 0) && (EntfRechts > 5 || EntfRechts == 0) ) {

 stepper1.setSpeed(-700.0);

 stepper2.setSpeed(700.0);

 stepper1.runSpeed();

 stepper2.runSpeed();

 EntfLinks = USMLinks();

 EntfRechts = USMRechts();

 }

(Unterschiede sind Blau markiert)

1. OUT Pin entkabeln, da wir den nicht brauchen
2. Zwei Arduinos miteinander kommunizieren lassen