

Erfassung der Geschwindigkeit und des Verfahrweges eines Förderbandes

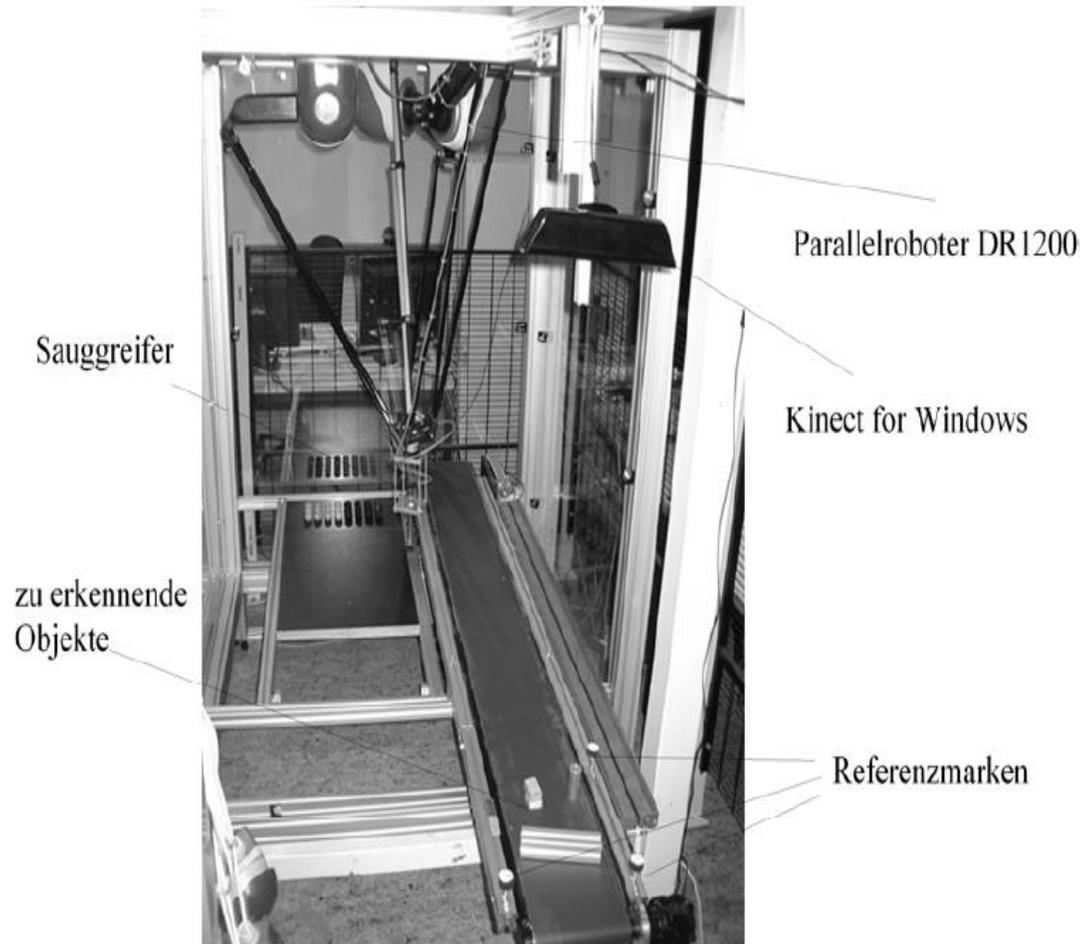
Andrea Molina
Simon Nagelschmidt
David Shenavai

Hochschule Darmstadt
Seminar der Robotik
Prof. Dr.-Ing. W. Weber

Inhalt

- ▶ Problemstellung
- ▶ Signalaufnahme
 - ▶ Auswahl des Sensors
- ▶ Signalbearbeitung
- ▶ Kommunikation zwischen μC und Motion Controller
- ▶ Fazit

Problemstellung



- ▶ Optimierung von Greifvorgang des Roboters DR1200
- ▶ Verfahrensweg von Förderband des DR1200 ist ungenau
 - ▶ Roboter greift neben Objekte

Auswahl des Sensors

Am Motor

- Einfache Montage
- höhere Auflösung
- Die exakte Geschwindigkeit des Bandes wird nicht erfasst

Am Objekt

- Präzise Geschwindigkeitsberechnung des Objekts
- Sehr teuer

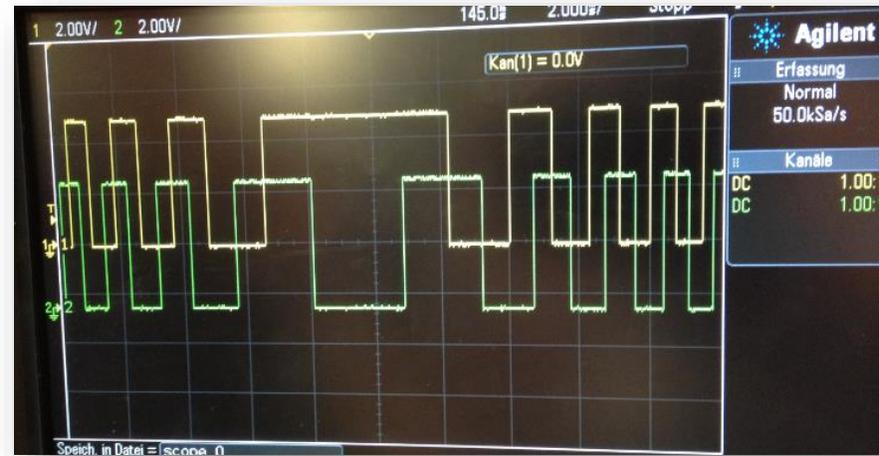
Am Band

- Einfache Montage
- Fast exakte Geschwindigkeit des Bandes
 - (Ungenauigkeiten wegen Schlupf)

Signalaufnahme Inkrementalgeber

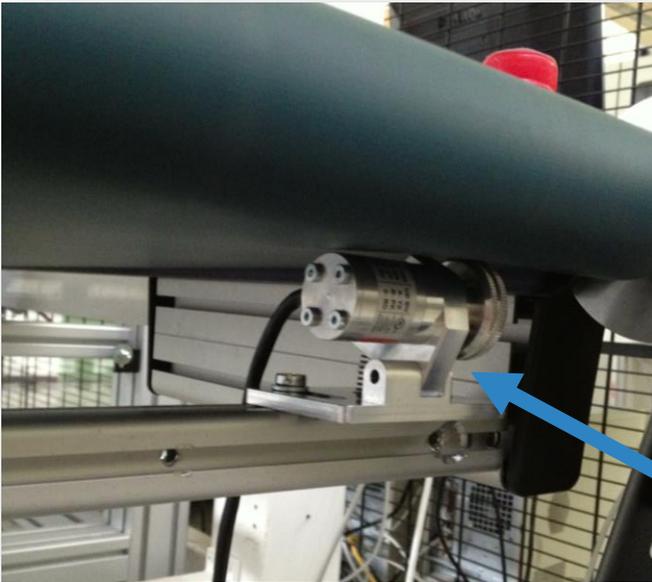
/

Signal des Gebers



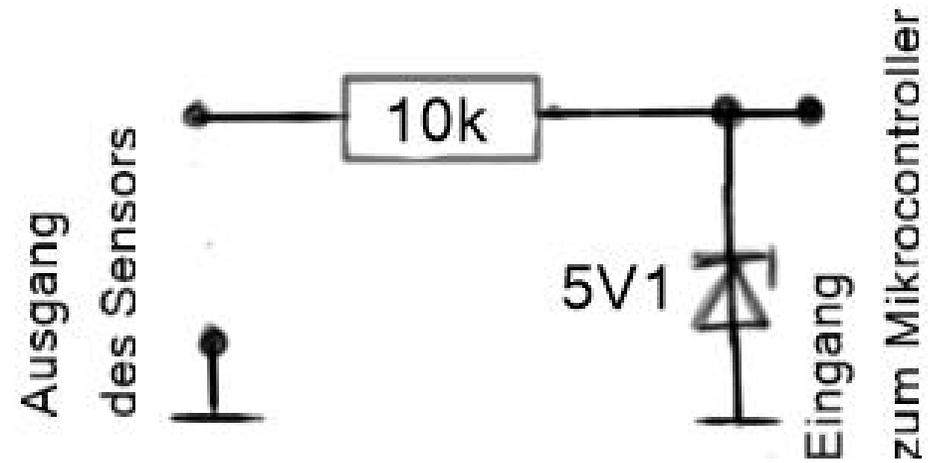
Montage und Energieversorgung

Montage



Sensor

Energieversorgung

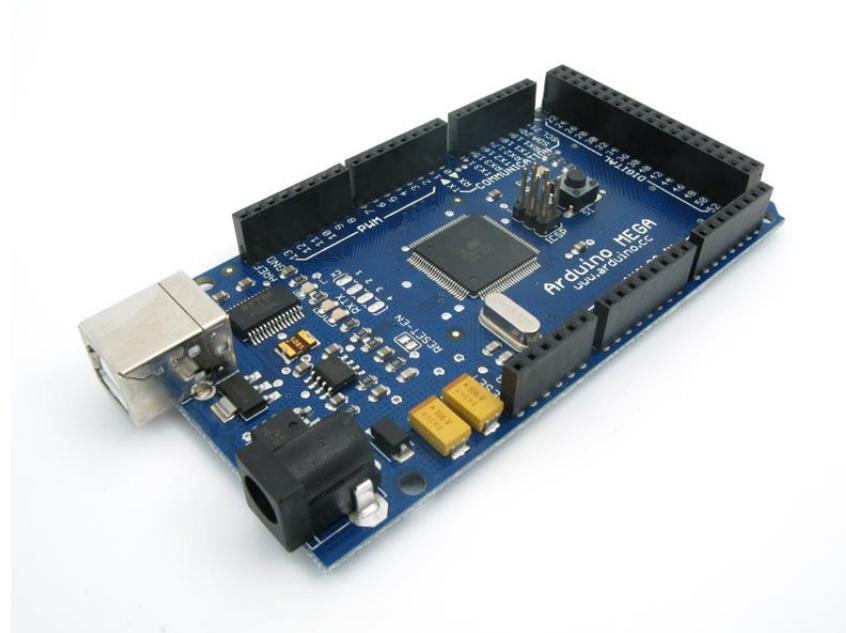


Mikrocontroller

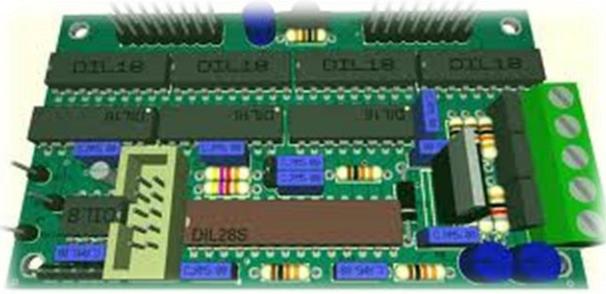
Auswahl

Programmierung

Fahrprofil



Auswahl



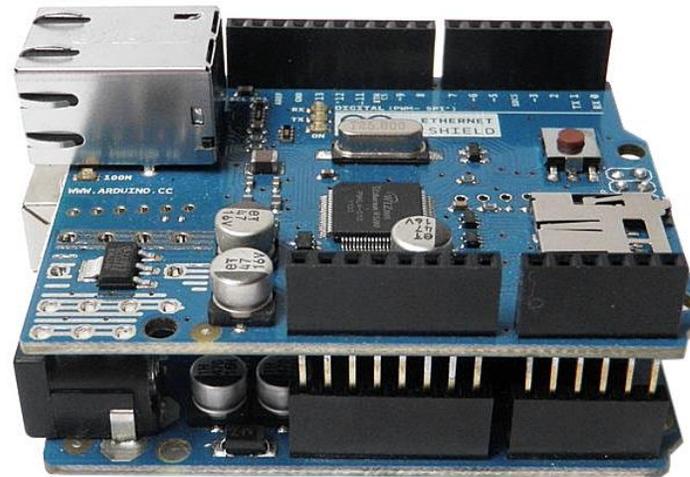
Kriterien

- ▶ Funktionen
- ▶ Geschwindigkeit
- ▶ Preis

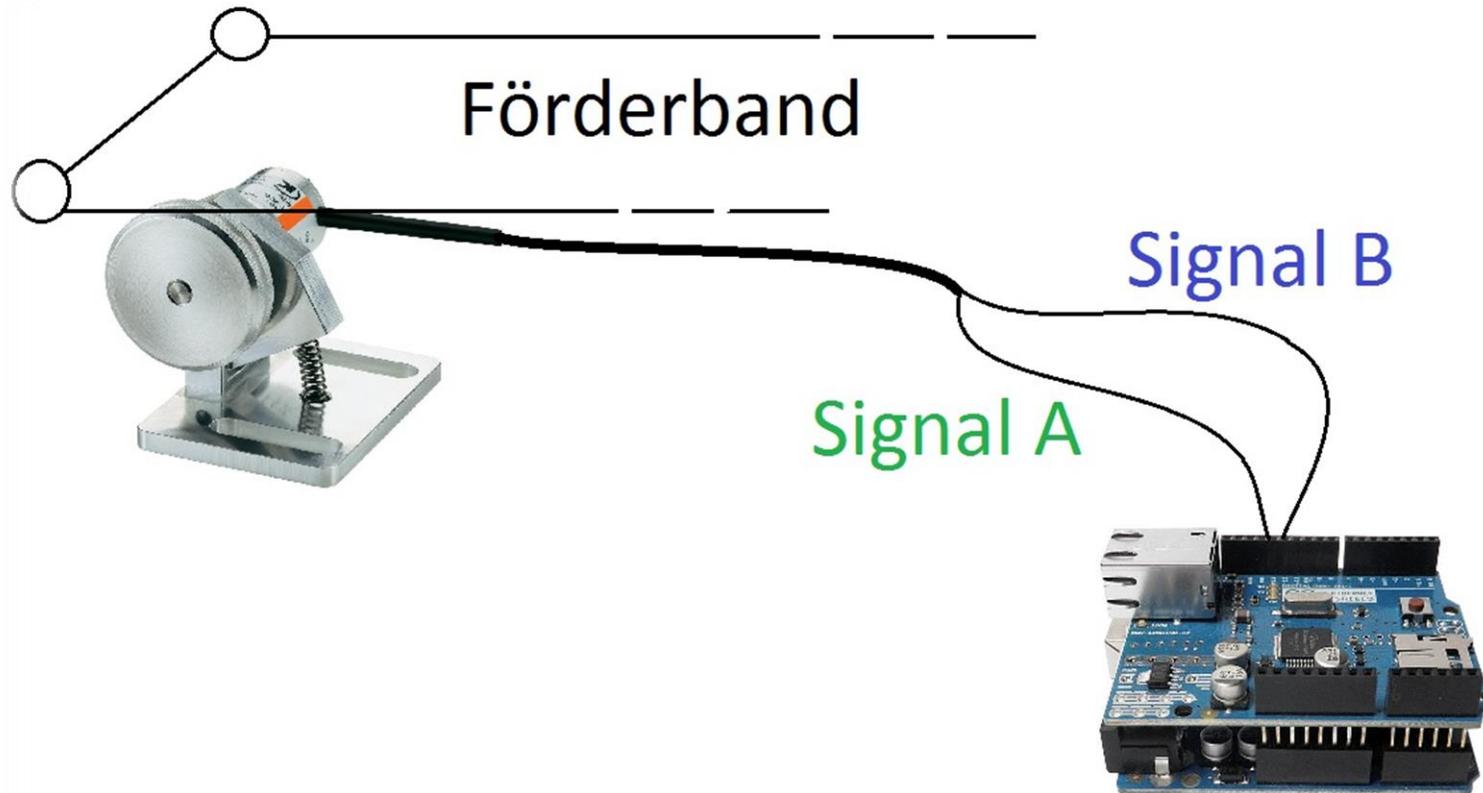


Wahl

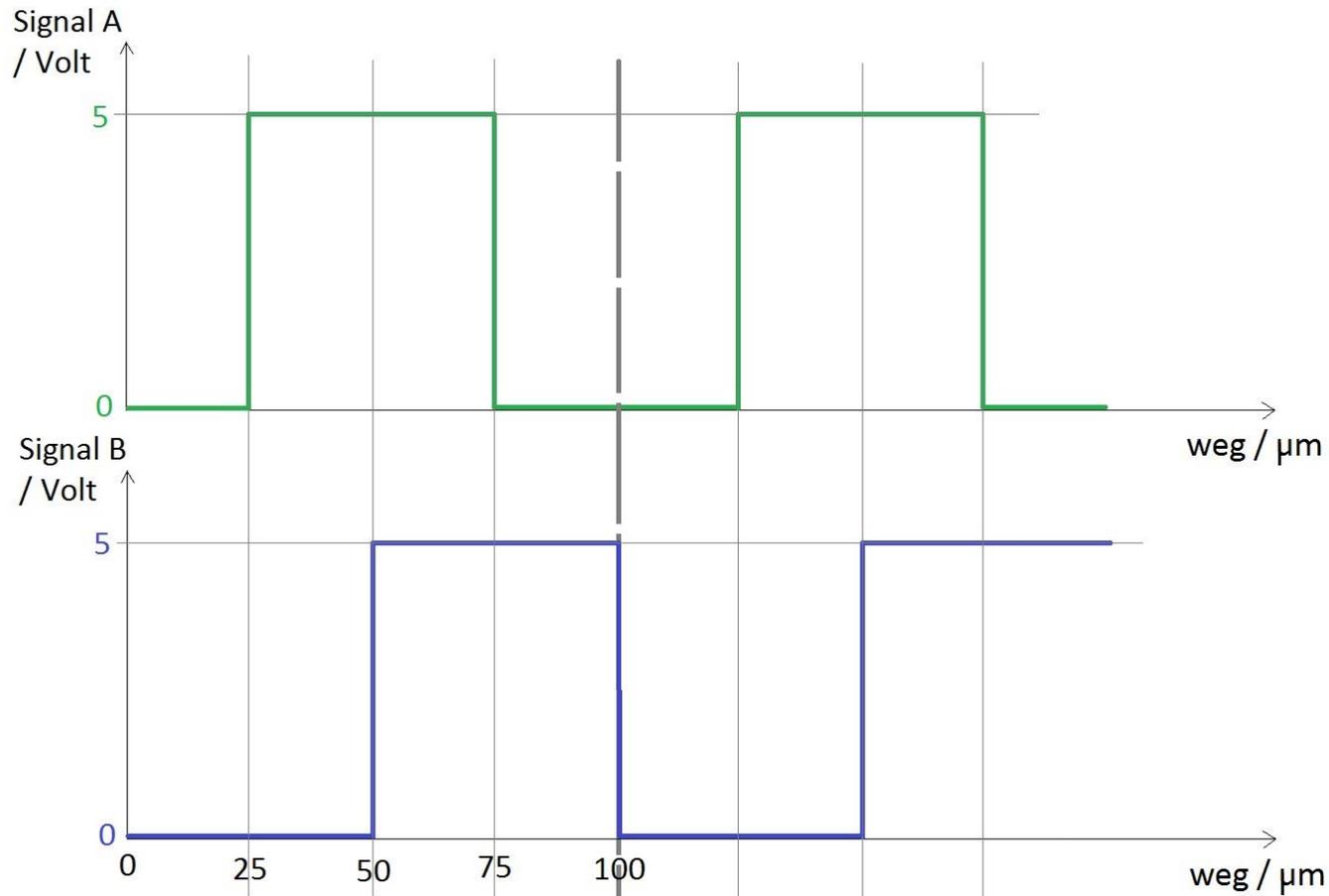
- ▶ Arduino Uno Board
 - ▶ Mikroprozessor: ATmega 328P



Wegmessung



Wegmessung Gray-codiertes Signal



Wegmessung Interrupt Service Routine

```
ISR(INT0_vect)  
{  
    . . .  
    impulse++;  
    . . .  
}
```

Wegmessung Interrupt Service Routine

```
ISR(INT1_vect)
{
    . . .
    impulse++;
    . . .
}
```

Wegmessung Hauptfunktion

```
Int main()  
{  
    ...  
    while(1)  
    {  
        ...  
        weg = impulse*25;  
        ...  
    }  
}
```

Geschwindigkeitsmessung

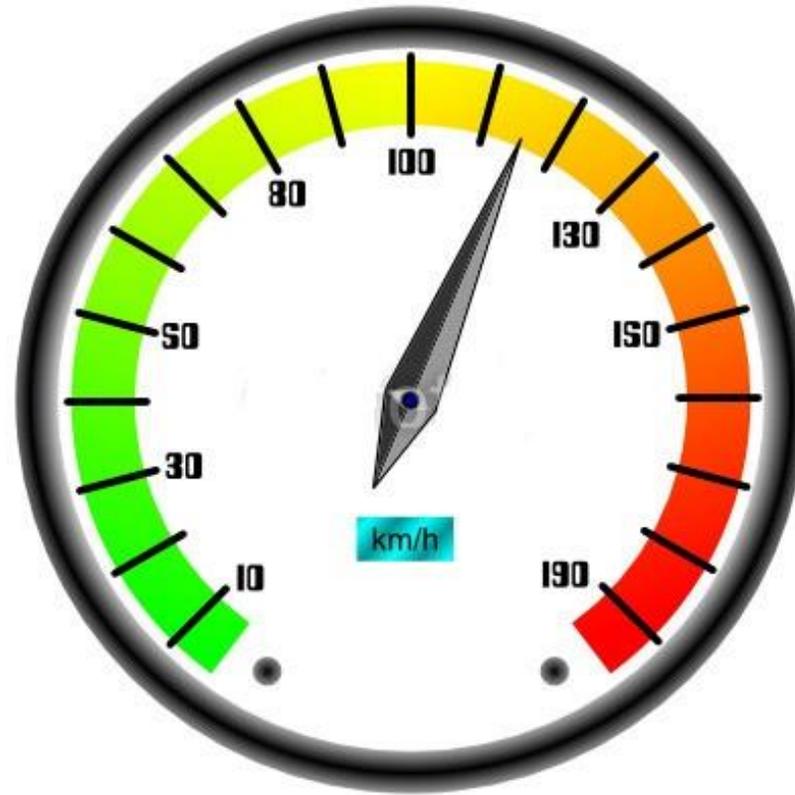
Analytisch

$$v(t) = ds/dt$$



Diskret

$$v(t) = \Delta s / \Delta t$$



Geschwindigkeitsmessung ISR des Timers

```
ISR(TIMERO_COMPA_vect)
```

```
{
```

```
    zeit++;
```

```
    . . .
```

```
}
```

Geschwindigkeitsmessung

Hauptfunktion

```
int main()
{
    ...
    while(1)
    {
        while(zeit%100 . . .)
        {
            asm("sleep"); //zum Stromsparen Mikrocontroller ausschalten
        }
        ...
        weg = impulse*25;

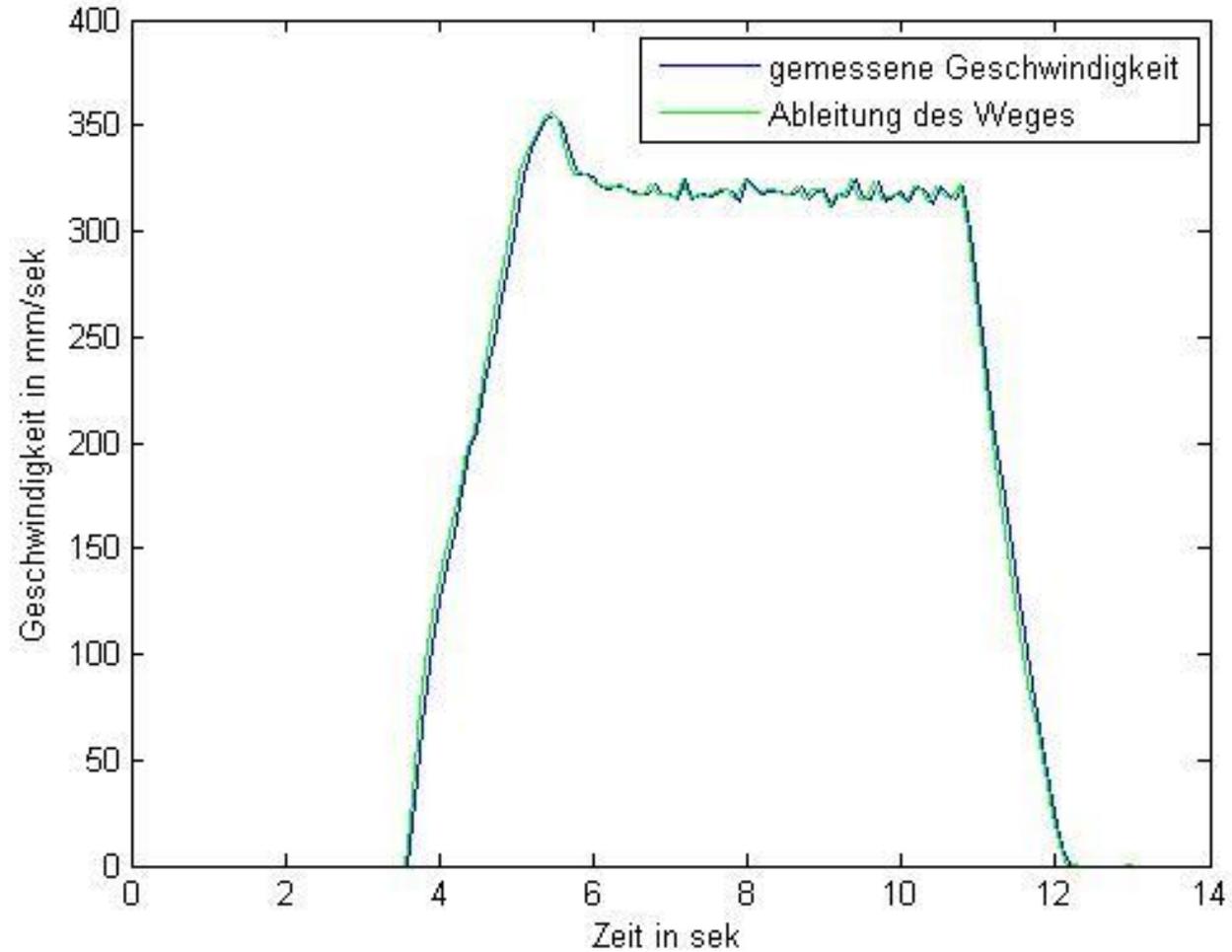
        //Geschwindigkeitsberechnung
        delta_s = weg-s0;
        delta_t = zeit-t0;
        v=delta_s/delta_t;
        s0 = weg;
        t0 = zeit;
        . . . //Senden der Daten an den Motion Controller per TCP/IP
    }
}
```

$\Delta s = s(t) - s_0(t)$

$\Delta t = t - t_0$

$v = \frac{\Delta s}{\Delta t}$

Geschwindigkeitsprofil

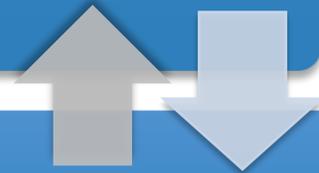


Kommunikation

Mikrocontroller

Ethernet - TCP

Motion Control



Kommunikation

- ▶ TCP / IP → Familie von Netzwerkprotollen
- ▶ TCP → Protokoll / Vereinbarung wie der Datenausch stattfindet (Syntax / Semantik)
- ▶ Identifizierung über IPv4-Adresse und Port
- ▶ Vorteile TCP:
 - ▶ Gesicherte Verbindung
 - ▶ Datenverlust wird erkannt und automatisch behoben
- ▶ Nachteil TCP:
 - ▶ Langsamer

Arduino Ethernet Shield



Netzwerk Aufbau

- ▶ Zum Netzwerk gehören:
 - ▶ Quelle-IP-Adresse und Port
 - ▶ Ziel-IP-Adresse und Port
- ▶ Allgemeiner Verbindungsaufbau im Netzwerk
 - ▶ Client sendet Verbindungsanfrage an Server
 - ▶ Server antwortet mit Bestätigung
 - ▶ Client bestätigt anschließend nochmals
 - ▶ Verbindung hergestellt → Datenaustausch kann beginnen

Informationsaustausch

- ▶ Informationsaustausch über Zeichenketten (String)
- ▶ Zeichenketten werden ausgewertet / Schlüsselwörter finden
- ▶ Konkreter Ablauf zwischen Client (Motion Control) und Server (Mikroprozessor):
 - ▶ Verbindungsaufbau
 - ▶ ...warten... → Bildaufnahme / Bildverarbeitung / Berechnungen
 - ▶ Befehl → Rücksetzen der Messwerte
 - ▶ Bestätigung → Messwerte zurückgesetzt
 - ▶ Befehl → Aktuellen Messwerte erfragen
 - ▶ Bestätigung → Aktuellen Messwerte senden
 - ▶ ...warten... → Start der Robotersoftware

Fazit und Ausblick

- ▶ Erfassung der Wegstrecke
- ▶ Berechnung der Geschwindigkeit
- ▶ Senden und Empfangen von Information über TCP / IP
- ▶ Grundvoraussetzung gelegt für spätere Anwendungen

Vielen Dank für Ihre
Aufmerksamkeit!