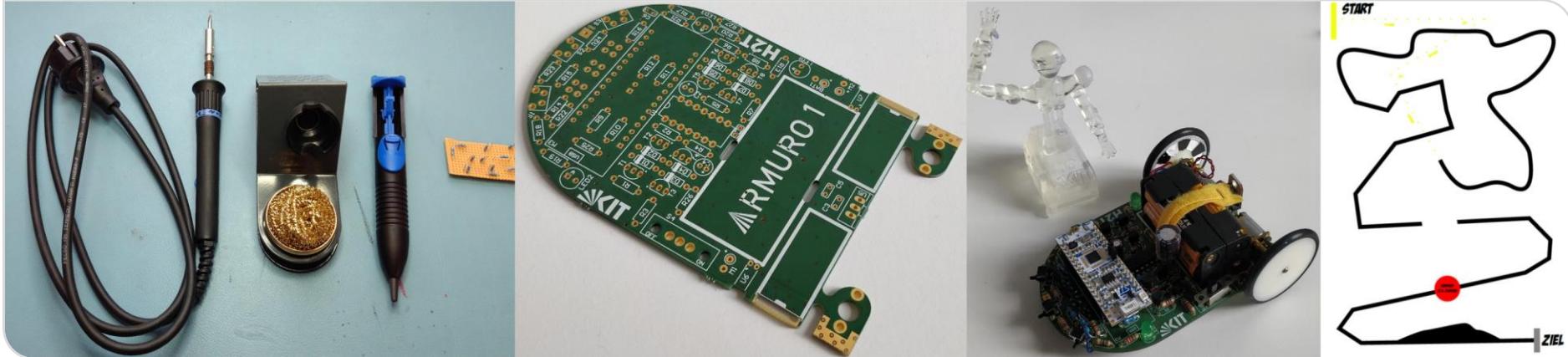


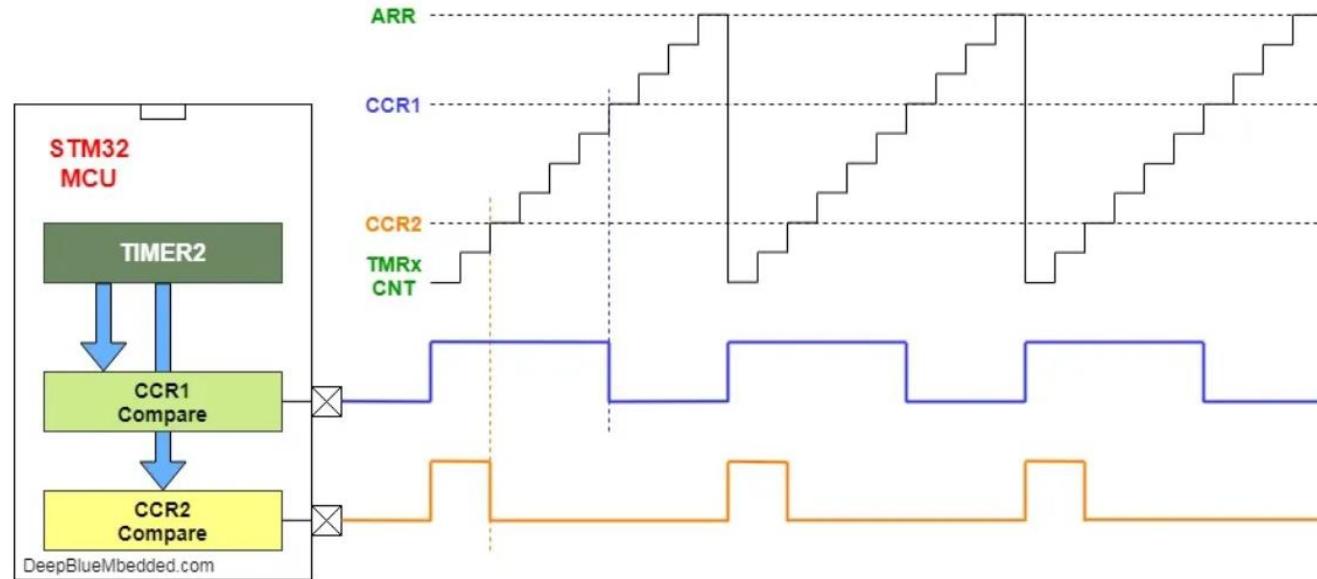
Basispraktikum Mobile Roboter im SoSe 2024

Kolloquium 04 am 29.05.2024



Frage 2.1

Wie wird ein PWM-Signal mittels eines Timers im Mikrocontroller erzeugt?



Frage 2.2

Für die Ansteuerung der Motoren werden Timer1 Channel 2N und Channel 3N verwendet. Wofür steht das N?

- Das N steht für „Not“, d.h. das PWM-Signal wird vor der Ausgabe invertiert

Frage 2.3

Schreiben Sie eine Funktion zur Ansteuerung der Motoren

```

int dead_Time;
void Motor_Steering(uint8_t motor, int speed){

    //direction

    if(motor == right){

        HAL_GPIO_WritePin(GPIOB_BASE , GPIO_PIN_1 , TIM_CHANNEL_2);

        if(speed >= 0){ //forward
            dead_Time = 65500 - ((65500 / 100) * speed);
            dead_Time = 65500 - dead_Time;
            TIM1->CCR3 = dead_Time;
            HAL_GPIO_WritePin(GPIOB_BASE , GPIO_PIN_6 , GPIO_PIN_RESET);
            //Break-Lights
            HAL_GPIO_WritePin (LED_SMD_GPIO_Port , LED_SMD_Pin , GPIO_PIN_RESET);
            if(speed == 0){
                HAL_GPIO_WritePin (LED_SMD_GPIO_Port , LED_SMD_Pin , GPIO_PIN_SET);
            }
        }else{ //backwards
            speed = -speed;
            dead_Time = 65500 - ((65500 / 100) * speed);
            TIM1->CCR3 = dead_Time;
            HAL_GPIO_WritePin(GPIOB_BASE , GPIO_PIN_6 , GPIO_PIN_SET);
            HAL_GPIO_WritePin (LED_SMD_GPIO_Port , LED_SMD_Pin , GPIO_PIN_RESET);
        }
    }

    if(motor == left){

        HAL_GPIO_WritePin(GPIOB_BASE , GPIO_PIN_0 , TIM_CHANNEL_3);

        if(speed >= 0){ //forward
            dead_Time = 65500 - ((65500 / 100) * speed);
            dead_Time = 65500 - dead_Time;
            TIM1->CCR2 = dead_Time;
            HAL_GPIO_WritePin(GPIOA_BASE , GPIO_PIN_12 , GPIO_PIN_SET);
            //Break-Lights
            HAL_GPIO_WritePin (LED_SMD_GPIO_Port , LED_SMD_Pin , GPIO_PIN_RESET);
            if(speed == 0){
                HAL_GPIO_WritePin (LED_SMD_GPIO_Port , LED_SMD_Pin , GPIO_PIN_SET);
            }
        }else{ //backwards
            speed = -speed;
            dead_Time = 65500 - ((65500 / 100) * speed);
            dead_Time = 65500 - dead_Time;
            TIM1->CCR2 = dead_Time;
            HAL_GPIO_WritePin(GPIOA_BASE , GPIO_PIN_12 , GPIO_PIN_RESET);
            HAL_GPIO_WritePin (LED_SMD_GPIO_Port , LED_SMD_Pin , GPIO_PIN_RESET);
        }
    }
}
  
```

Jonte Fiegert

Frage 2.3 - Beispiel

Schreiben Sie eine Funktion zur Ansteuerung der Motoren

```

40 void motorControl(char dir, float speed) {
41   switch(dir) {
42     //Vorwärts fahren
43     case 'f':
44       →HAL_GPIO_WritePin(GPIOB, phase2_R_Pin, GPIO_PIN_RESET);
45       →HAL_GPIO_WritePin(GPIOA, phase2_L_Pin, GPIO_PIN_SET);
46       →TIM1->CCR2 = pwm(speed,1);
47       →TIM1->CCR3 = pwm(speed,0);
48       break;
49     //Rückwärts fahren
50     case 'b':
51       →HAL_GPIO_WritePin(GPIOB, phase2_R_Pin, GPIO_PIN_SET);
52       →HAL_GPIO_WritePin(GPIOA, phase2_L_Pin, GPIO_PIN_RESET);
53       →TIM1->CCR2 = pwm(speed,0);
54       →TIM1->CCR3 = pwm(speed,1);
55       break;
56     //Links abbiegen
57     case 'l':
58       →HAL_GPIO_WritePin(GPIOB, phase2_R_Pin, GPIO_PIN_RESET);
59       →HAL_GPIO_WritePin(GPIOA, phase2_L_Pin, GPIO_PIN_RESET);
60       →TIM1->CCR2 = pwm(0,0);
61       →TIM1->CCR3 = pwm(speed,0);
62       break;
63     ....
  
```

```

20 int pwm(float speed, int inv)
21 {
22   if(inv == 0){
23     return 65535*speed;
24   }
25   else if(inv == 1){
26     return 65535*(1-speed);
27   }
28 }
  
```

Frage 3.1

Wie können die Rohwerte des ADCs in Winkel übersetzt werden? Welche Probleme können bei der Auswertung der Rohwerte auftreten? Durch welche Maßnahmen kann die Robustheit der Auswertung erhöht werden?

■ Messen

- Flankenerkennung
- Schwellwertenscheider

■ Rechnen

- Mitzählen der Übergänge
- Umrechnung der Flankenzahl in Winkel mittels hardwarespezifischer Parameter

■ Probleme

- Beeinflussung durch Umgebungslicht
- Ausrichtung/Abstand der Sensoren zum Rad
- Kalibrierung des ADC

■ Maßnahmen

- Richtige Anordnung von Rad und Encoder
- Sorgfältiges und gleichmäßiges Auftragen der Farben des Radencoders

ADC Selbstkalibrierung

- Der interne ADC des Microcontrollers muss zu Beginn während der Initialisierung kalibriert werden!

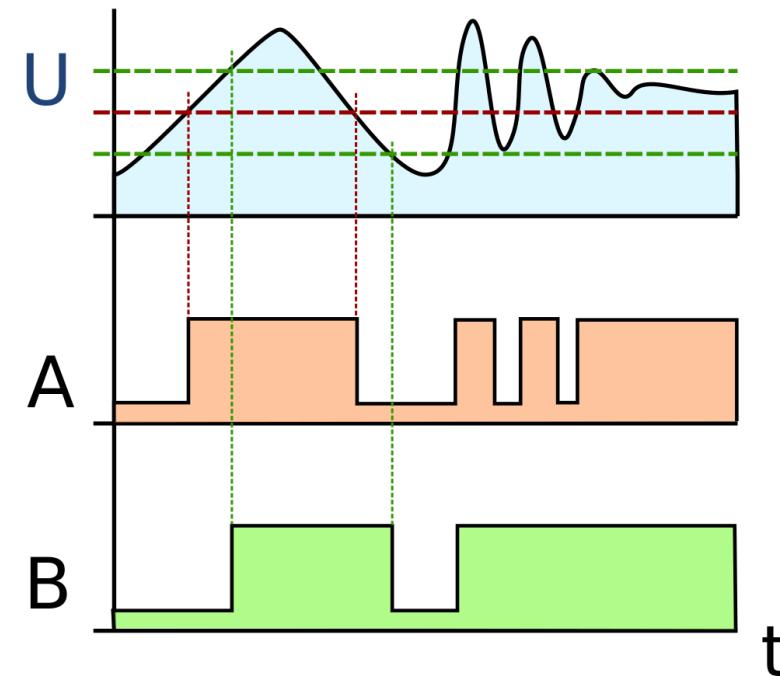
- Microcontroller-Konfiguration erzeugt Offset im ADC
- STM HAL hat eine Selbstkalibrierungsfunktion (stm32l4xx_hal_adc_ex.c):

```
HAL_StatusTypeDef HAL_ADCEx_Calibration_Start(ADC_HandleTypeDef *hadc,  
uint32_t SingleDiff)
```

Frage 3.1

Wie funktioniert ein Schmitt Trigger?

- Zwei Schwellwerte statt einem
- Im „Totbereich“ zwischen Schwellwerten wird der letzte Zustand gehalten (B im Bild)
- Verhindert Rauschen des Digitalsignals bei Rauschen des Analogsignals um den Schwellwert (A im Bild)
- Kann z.B. über Zustandsautomat abgebildet werden



Frage 3.2

Wie kann der Armuro die Drehzahl eines Rades bestimmen?

Nennen Sie zwei Verfahren zur Drehzahlmessung mit diskreten Winkelgebern. Wo liegen die jeweiligen Vor- und Nachteile?

- Bestimmung der Frequenz der hell/dunkel Übergänge
- Zählen der Übergänge in einer festen Zeit T (windowing)
 - Vorteil: Einfach
 - Nachteil: hohes Quantisierungsrauschen für kleine T, niedrige Update Frequenz für große T
- Messen der Periodendauer pro Segment
 - Vorteil: Sehr schnelle Antwort auf Geschwindigkeitsänderungen
 - Nachteil: hohe Samplefrequenz notwendig, sonst hohes Diskretisierungsrauschen

Frage 3.3 & 3.4

Warum können die Radencoder die Drehrichtung der Räder nicht bestimmen?

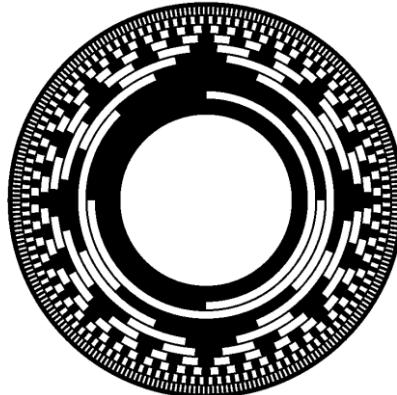
Wie müssten die Radencoder erweitert werden, so dass die Drehrichtung bestimmt werden kann?

- Aufgrund des Designs der Radencoder ist deren Drehrichtung prinzipiell nicht bestimmbar
- Um dieses Problem zu umgehen kann die Drehrichtung der Motoren hinzugezogen werden (falls bekannt)

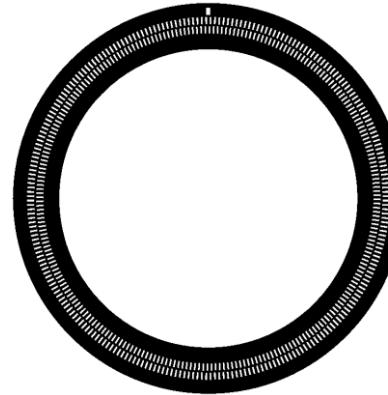
- Hinzufügen eines zweiten Sensors pro Encoder mit 90° Phasenversatz
- Stichwort: Quadraturencoder

Optische Positionencoder

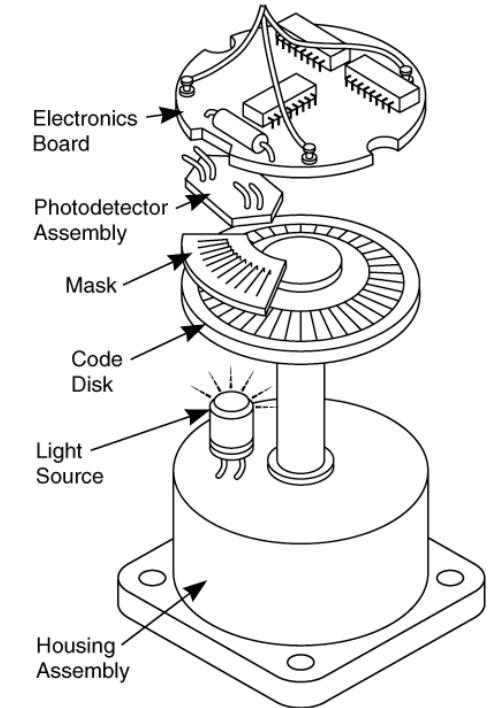
- Encoderscheibe
- I.d.R. Durchlicht-Lichtquelle
- Photodetektorarray



Absolutencoderscheibe
(z.B. mittels. Gray Code)



Relativencoderscheibe



Frage 4.1

Implementieren Sie eine Funktion, die Sensoren ausliest und die Rohwerte auf der seriellen Schnittstelle ausgibt.

```
typedef struct {
    uint32_t pin;
    char *purpose;
} ADC_Channel_t;

ADC_Channel_t adc_channels[] = {
    {GPIO_PIN_0, "Line Sensor Middle"}, // PA0
    {GPIO_PIN_1, "Encoder Left"}, // PA1
    {GPIO_PIN_3, "Line Sensor Right"}, // PA3
    {GPIO_PIN_4, "Battery"}, // PA4
    {GPIO_PIN_5, "Encoder Right"}, // PA5
    {GPIO_PIN_7, "Line Sensor Left"} // PA7
};

uint32_t adc_values[6];

void print_Sensors(ADC_HandleTypeDef *hadc, UART_HandleTypeDef *huart) {
    HAL_ADC_Start_DMA(hadc, (uint32_t*) adc_values, 6);
    for (int i = 0; i < 6; i++) {
        char uart_data[1000];
        sprintf(uart_data, "ADC[%d] %d %s\r\n", i, adc_values[i], adc_channels[i].purpose);
        HAL_UART_Transmit(huart, (uint8_t *)uart_data, strlen(uart_data), HAL_MAX_DELAY);
    }
}
```

Tobias Kirschner

Frage 4.3

Was ist bei der Wahl der Ausleserate der Encoder zu beachten?
Welche Effekte hat eine zu niedrige Samplefrequenz?

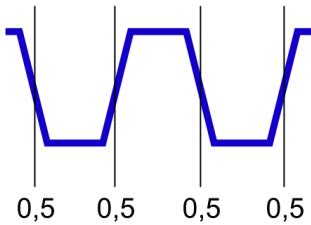
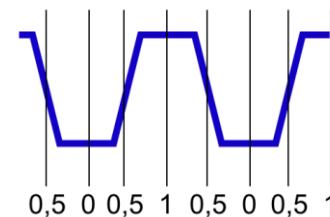
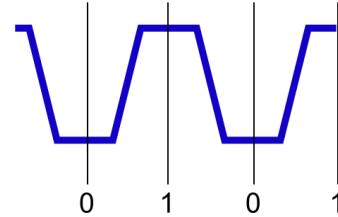
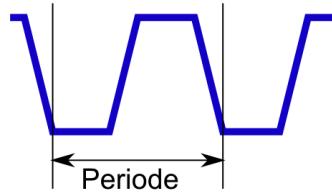
- Die Sensoren müssen schnell genug ausgelesen werden, so dass keine Flanken verpasst werden
- Die Länge der Filter muss angepasst werden
- Aliasing
- Es kann passieren, dass ein Übergang verpasst wird

Frage 4.4

Stellen Sie eine Formel zur Berechnung der minimalen Samplefrequenz auf, um alle Flanken zu detektieren.

$$F = 2NM$$

$$\text{z.B. } F \geq 4NM$$



Fragen?