

Koden

Kod för bilen

```
/*
 * bil.c
 *
 * Created: 2023-04-30 15:17:59
 * Author : Samuel
 */

#include <avr/io.h> // IO definitions

uint8_t direction;

void init(){
    // pb3, pb4, pb6, pb7 pointing out.
    DDRB |= (1 << DDRB3) | (1 << DDRB4) | (1 << DDRB6) | (1 << DDRB7);
    // Setting baud rate to 9600 Baud.
    UBRR0 = 6;
    // turning on receiver 0 (rx0).
    UCSR0B |= (1 << RXEN0);
}

uint8_t receive(){
    // wait for data
    while(!(UCSR0A & (1<<RXC0)));
    // return data.
    return UDR0;
}

void move(uint8_t data){
    if(data == 1){ /* fram */
        PORTB |= (0 << PINB3) | (0 << PINB4) | (1 << PINB6) | (1 <<
PINB7);
    }
    else if(data == 2){ /* hoger */
```

```

        PORTB |= (1 << PINB3) | (0 << PINB4) | (0 << PINB6) | (0 <<
PINB7);
    }
    else if(data == 3){ /* bak */
        PORTB |= (1 << PINB3) | (1 << PINB4) | (0 << PINB6) | (0 <<
PINB7);
    }
    else if(data == 4){ /* vanster */
        PORTB |= (0 << PINB3) | (1 << PINB4) | (0 << PINB6) | (0 <<
PINB7);
    }

    else PORTB = 0x00;
}

int main(void)
{
    init();
    while (1)
    {
        direction = receive();
        move(direction);
    }
}

```

Kod för styrenheten

```

/*
 * REMOTE.c
 *
 * Created: 2023-04-29 15:17:59
 * Author : Samuel
 */
#define F_CPU 8000000UL // defining cpu speed for delay function
#include <avr/io.h> // IO definitions
#include <util/delay.h> // including delay functions

uint16_t adc_value_x;
uint16_t adc_value_y;
uint8_t direction = 0;

```

```

void init(){
    ADCSRA |= (1 << ADEN); /* Enable ADC */
    DDRA = 0x0; /* PORTDN INPUT MODE */
    ADMUX |= ((1 << REFS0) | (0 << REFS1)); // AVcc as reference
voltage.

    TCCR1C |= (1 << CS02) | (1 << CS00);
    TCNT0 = 0x00;

    UBRR0 = 6; /* Set baud rate to 9600 BAUD */
    UCSR0B |= (1 << TXEN0); // enable transceiver 0.
}

uint16_t adc_read(uint8_t channel){
    ADMUX = (ADMUX & 0xF0) | (channel & 0x0F); // setting adc channel
    ADCSRA |= (1 << ADSC); // starting conversion
    while ( (ADCSRA & 0b01000000) != 0){} // wait for conversion
    return ADC; // returning conversion
}

void transmit(uint8_t data){
    /* Wait for empty transmit buffer */
    while(!(UCSR0A & (1 << UDRE0))){}
    /* send data */
    UDR0 = data;
}

int main(void)
{
    init();

    while (1)
    {
        adc_value_x = adc_read(0);
        adc_value_y = adc_read(1);
        // calculate direction
        if(adc_value_x > 1000 && adc_value_y < 1000){
            direction = 1; // Forward
        }
        else if(adc_value_y > 1000 && adc_value_x < 1000){
            direction = 2; // left
        }
        else if(adc_value_x < 5 && adc_value_y > 10){
            direction = 3; // Back
        }
        else if(adc_value_y < 5 && adc_value_x > 10){

```

```
        direction = 4; // right
    }
    else direction = 0; // Stand still

    _delay_ms(500); // delay for timing
    transmit(direction);
}
}
```