

```

// Alle Signale vom Empfänger Kanal 6 SUMD auslesen und für BLDC Steuerung umsetzen
// 10.09.2019
//

#include <Wire.h> // Wire Bibliothek hochladen;
#include <LiquidCrystal_I2C.h> // LiquidCrystal_I2C //Bibliothek hochladen;
LiquidCrystal_I2C lcd1(0x27, 16, 2); //Displaytyp und Zeichenumfang 16 Zeichen in 2
Zeilen
LiquidCrystal_I2C lcd2(0x26, 16, 2);

int i, channel[6], rcValue[6];
#define SUMD_MAXCHAN 6
#define SUMD_BUFFSIZE SUMD_MAXCHAN*2 + 5
static uint8_t sumdIndex=0;
static uint8_t sumdSize=0;
static uint8_t sumd[SUMD_BUFFSIZE]={0};

const int midpv=1500; // Mittelstellung Joystick vor-rue
const int maxpv=2000; // Vollauschlag Joystick vor
const int minpv=1000; // Vollausschlag Joystick rue
const int midpr=1500; // Mittelstellung Joystick re-li
const int maxpr=2000; // Vollauschlag Joystick rechts
const int minpr=1000; // Vollausschlag Joystick links
const int empf1=2; // Digital Eingang Empfängerkanal 1 auf PIN D2
const int empf2=3; // Digital Eingang Empfängerkanal 2 auf PIN D3
int pulslaenge1=1500; // Pulslänge digital Kanal 1 vor-zurück in mikro-sec
int pulslaenge2=1500; // Pulslänge digital Kanal 2 rechts-links in mikro-sec
int pulslaenge=1500; // Pulslänge Relaiskanäle
const int drehrire=30; // digi-Pin Ausgang für Drehrichtung rechter Motor LOW =
rückwärts
const int drehrili=31; // digi-Pin Ausgang für Drehrichtung linker Motor LOW =
rückwärts
const int bremsre=32; // digi-Pin Ausgang für Bremse rechter Motor LOW = Bremse an
const int bremsli=33; // digi-Pin Ausgang für Bremse linker Motor LOW = Bremse an
int drehrirealt; // alte Drehrichtung vorheriger Durchlauf rechter Motor
int drehrilialt; // alte Drehrichtung vorheriger Durchlauf linker Motor
int zwvor=0; // Zwischenwert vor-rück
int zwre=0; // Zwischenwert Fahrtrichtung rechts
int zwli=0; // Zwischenwert Fahrtrichtung links
int pause = 200; // 2x Umschaltpause in millisec bei Drehrichtungswechsel
const int motrepin=10; // PWM-Pin D10 für rechten Motor
const int motlipin=11; // PWM-Pin D11 für linken Motor
int motre; // rechter Motor Steuerspannung 0-255 (0-5V)
int motli; // linker Motor Steuerspannung 0-255 (0-5V)
boolean relais = (HIGH); // Voreinstellung Relaiskanäle

void setup(){
  lcd1.begin(); // LCD wird gestartet
  lcd2.begin();
  Serial1.begin(115200); // Baudrate für HOTT-Empfänger
  pinMode(empf1, INPUT); // Empfängerkanal D2
  pinMode(empf2, INPUT); // Empfängerkanal D3
  pinMode(10, OUTPUT); // Ausgang rechter Motor
  pinMode(11, OUTPUT); // Ausgang linker Motor
  pinMode(drehrire, OUTPUT); // Drehrichtung rechter Motor schreiben
  pinMode(bremsre, OUTPUT); // Bremse rechter Motor
  pinMode(drehrili, OUTPUT); // Drehrichtung linker Motor schreiben
  pinMode(bremsli, OUTPUT); // Bremse linker Motor
  pinMode(40, OUTPUT); // Testkanal
  pinMode(41, OUTPUT); // Testkanal
  // for (int i=4; i < 10; i++) {pinMode (i, INPUT);} // Relaiskanäle 4-9 lesen
  // for (int i=24; i < 30; i++) {pinMode (i, OUTPUT);} // Relaiskanäle 24-29 schreiben
}

void loop() {
  while (Serial1.available()) {
    int val = Serial1.read();
    if(sumdIndex == 0 && val != 0xA8) {continue; }
    if(sumdIndex == 2) {sumdSize = val;}
    if(sumdIndex < SUMD_BUFFSIZE) {sumd[sumdIndex] = val;}
    sumdIndex++;

    if(sumdIndex == sumdSize*2+5)
    { sumdIndex = 0;
      if (sumdSize > SUMD_MAXCHAN) sumdSize = SUMD_MAXCHAN;
      for (uint8_t b = 0; b < sumdSize; b++){
        rcValue[b] = ((sumd[2*b+3]<<8) | sumd[2*b+4])>>3;
        if (rcValue[b] > 750 && rcValue[b] < 2250){channel[b] = rcValue[b];} //filter
      }
    }
  }
}

```

```

    }
    }

    lcd1.setCursor(0,0);
    lcd1.setCursor(0,0);
    lcd1.print(channel[0]);
    lcd1.setCursor(5,0);
    lcd1.print(channel[1]);
    lcd1.setCursor(10,0);
    lcd1.print(channel[2]);
    lcd1.setCursor(0,1);
    lcd1.setCursor(0,1);
    lcd1.print(channel[3]);
    lcd1.setCursor(5,1);
    lcd1.print(channel[4]);
    lcd1.setCursor(10,1);
    lcd1.print(channel[5]);

    // delay(50);

    pulslaenge1 = channel(0);
    pulslaenge2 = channel(1);

    if (puls1aenge1 != 0) {(zwvor =midpv); {
        if (puls1aenge1 > midpv) { (zwvor = (((puls1aenge1+2)-midpv)/2));} else { (zwvor =
        (((midpv+1)-puls1aenge1)/-2));}}} // Erst Einschaltsituation, dann Vorwärtspulse 0-255
    errechnen (else = negativer Anteil)

    if (puls1aenge2 != 0) {(zwre = 0); {
        if (puls1aenge2 > midpr) { (zwre = (((puls1aenge2+2)-midpr)/2));} else { (zwre =
        ((midpr+1)-puls1aenge2)/-2));}}} // Erst Einschaltsituation, dann Rechtspulse 0-255
    errechnen (else = negativer Anteil)

    //
    *****

    if ((zwvor-zwli) < 0){ motli = abs(zwvor-zwli); if (drehrilialt == 9){
        digitalWrite(bremsli, LOW); delay(pause); digitalWrite(drehrili, LOW);
    digitalWrite(bremsli, HIGH); drehrilialt = 1;}
        else { digitalWrite(drehrili, LOW); drehrilialt = 1;}} // von Vorwärts nach
    Rückwärts mit Umschaltpause linker Motor

    if ((zwvor-zwli) > 0){ motli = abs(zwvor-zwli); if (drehrilialt == 1){
        digitalWrite(bremsli, LOW); delay(pause); digitalWrite(drehrili, HIGH);
    digitalWrite(bremsli, HIGH); drehrilialt = 9;}
        else { digitalWrite(drehrili, HIGH); drehrilialt = 9;}} // von Rückwärts nach
    Vorwärts mit Umschaltpause linker Motor

    if ((zwvor-zwre) < 0){ (motre = abs(zwvor-zwre)); if (drehrirealt == 9){
        digitalWrite(bremsre, LOW); delay(pause); digitalWrite(drehrire, LOW);
    digitalWrite(bremsre, HIGH); drehrirealt = 1;}
        else { digitalWrite(drehrire, LOW); drehrirealt = 1;}} // von Vorwärts nach
    Rückwärts mit Umschaltpause rechter Motor

    if ((zwvor-zwre) > 0){ (motre = abs(zwvor-zwre)); if (drehrirealt == 1){
        digitalWrite(bremsre, LOW); delay(pause); digitalWrite(drehrire, HIGH);
    digitalWrite(bremsre, HIGH); drehrirealt = 9;}
        else { digitalWrite(drehrire, HIGH); drehrirealt = 9;}} // von Rückwärts nach
    Vorwärts mit Umschaltpause rechter Motor

    //
    *****

    //      motre = Steuerspannung 0-5V Motor rechts (Wert 0-255)
    //      motli = Steuerspannung 0-5V Motor links (wert 0-255)
    analogwrite(motrepin, motre); // Steuerspannung rechter Motor 0-5V schreiben
    analogwrite(motlipin, motli); // Steuerspannung linker Motor 0-5V schreiben

    lcd2.setCursor(0,0);
    lcd2.setCursor(0,0);
    lcd2.print(puls1aenge1,10);
    lcd2.setCursor(5,0);
    lcd2.print(puls1aenge2,10);
    lcd2.setCursor(11,0);
    lcd2.print(drehrilialt,10);

```

```
lcd2.setCursor(14,0);  
lcd2.print(drehrireal1t,10);  
lcd2.setCursor(0,1);  
lcd2.setCursor(0,1);  
lcd2.print((zwvor-zwli),10);  
lcd2.setCursor(5,1);  
lcd2.print((zwvor-zwre),10);  
lcd2.setCursor(10,1);  
lcd2.print(motli,10);  
lcd2.setCursor(13,1);  
lcd2.print(motre,10);
```

```
}
```