

```

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

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

void setup(){
  lcd.begin();          // LCD wird gestartet
  Serial.begin(115200);
}

void loop() {
  while (Serial.available()) {
    int val = Serial.read();
    test = val;
    if(sumdIndex == 0 && val != 0xA8) {continue; }
    if(sumdIndex == 2) {sumdSize = val;}
    if(sumdIndex < SUMD_BUFSIZE) {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
      }
    }

    debug();
  }

  void debug(){
    // lcd.setCursor(0,0);
    // lcd.print(test);
    // delay(1000);
    lcd.setCursor(0,0);
    // lcd.print("      ");
    lcd.setCursor(0,0);
    lcd.print(channel[0]);
    lcd.setCursor(5,0);
    lcd.print(channel[1]);
    lcd.setCursor(10,0);
    lcd.print(channel[2]);
    lcd.setCursor(0,1);
    // lcd.print("      ");
    lcd.setCursor(0,1);
    lcd.print(channel[3]);
    lcd.setCursor(5,1);
    lcd.print(channel[4]);
    lcd.setCursor(10,1);
    lcd.print(channel[5]);

    delay(50);
  }
}

```