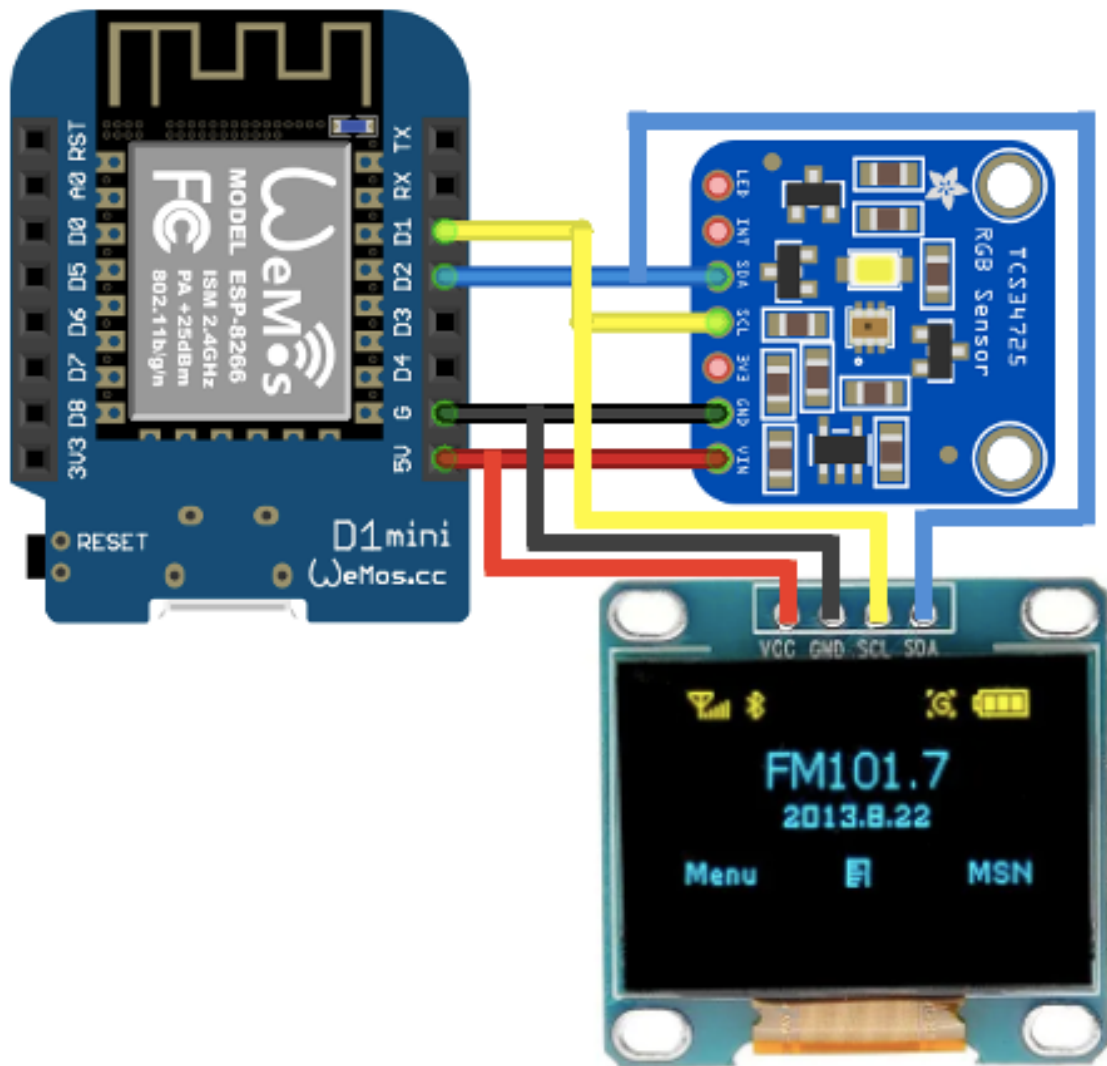


Manual Software Sensor Pendeteksi Warna Kulit Bayi Hiperbilirubin

Software ini merupakan program pengendali mikrokontroller dengan pemrograman Arduino dengan tujuan memanfaatkan sensor warna untuk mendeteksi warna kulit bayi kuning (hiperbilirubin) yang selaras dengan kadar bilirubin darah bayi tersebut. Mikrokontroler membaca data sensor warna, kemudian hasil bacaan akan dikonversikan menjadi perkiraan kadar bilirubin bayi, yang sebelumnya telah dilakukan penelitian untuk mencari persamaan hubungan antara warna kulit bayi dan kadar bilirubin darah. Hasil pembacaan akan ditampilkan di monitor yang tersambung ke mikrokontroller tersebut. Pembacaan dilakukan tiga kali sampai memperoleh angka yang stabil dan menampilkan angka yang valid pada monitor.

Software ini dapat dijalankan menggunakan aplikasi Arduino (<https://www.arduino.cc>) dengan perangkat mikrokontroller Wemos D1 mini, sensor warna dan layar OLED, seperti susunan dibawah ini.



1. Setting perbedaan pembacaan alat yang akan ditoleransi ketika pembacaan berulang. Jika toleransi pembacaan berulang dianggap sama dalam perbedaan 2%, maka setting diisi 0.02. Dalam hal ini, pembacaan akan dikatakan stabil jika selisih pembacaan terakhir dan rata-rata tiga kali pembacaan terakhir kurang dari 2%.

```
/* set the % Different for stable reading, 0.1 for 10%, 0.5 for 50% etc*/
float diff = 0.02;
```

2. Setting kalibrasi perbedaan alat yang dirangkai. Beda waktu perangkaian dan komponen, ternyata memberikan sedikit perbedaan pembacaan, maka perlu dilakukan kalibrasi terlebih dahulu, dengan membaca kertas putih, dibandingkan dengan perlengkapan yang sudah dirangkai sebelumnya. Rasio perbedaannya akan menjadi konstanta pembagi hasil pembacaan alat yang baru.

```
/* Calibration value of each TCS32725 */  
float cR= 1.00, cG = 1.00, cB = 1.00, cC= 1.00 ;
```

3. Deklarasi mikrokontroler untuk melibatkan library eksternal yang dipakai, yaitu library untuk sensor warna, layar OLED dan koneksi I2C.

```
#include <Wire.h>  
#include "Adafruit_TCS34725.h"  
#include "SSD1306.h"
```

4. Catatan koneksi sensor, display dan mikrokontroler, seperti pada gambar di atas

```
/* wire connection  
Connect SCL to analog D1  
Connect SDA to analog D2  
Connect VDD to 3.3V DC / 5 V DC  
Connect GROUND to common ground */
```

5. Setting dan memulai perangkat sensor warna dan display OLED.

```
/* Initialise with default values (int time = 2.4ms, gain = 1x) */  
// Adafruit_TCS34725 tcs = Adafruit_TCS34725();
```

```
/* Initialise with specific int time and gain values */
```

```
Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_700MS,  
TCS34725_GAIN_1X);  
SSD1306 display(0x3c, 5, 4);
```

```
void setup(void)  
{
```

6. Setting display layar OLED dalam bentuk landscape

```
display.init();  
display.flipScreenVertically();
```

```
/* Check if the color sensor was connect properly and ready to use */
```

7. Cek apakah sensor warna terkoneksi dengan baik

```
if (tcs.begin()) {  
    display.drawString(0,0,"Found sensor");  
    display.display();  
} else {  
    display.drawString(0,0,"Sensor Not Found..");  
    display.display();  
    while (1);  
}  
  
}
```

8. Pembacaan warna secara berulang oleh mikrokontroller, dibaca sebanyak tiga kali dan dibandingkan, jika pembacaan sudah stabil, maka ditampilkan "OK" pada display. Jika pembacaan belum stabil, maka ditampilkan "Reading...".

```
void loop(void) {  
  
    uint16_t r, g, b, c, colorTemp, lux, r1, r2, r3, g1, g2, g3, b1, b2, b3, c1, c2, c3 ;  
    float avg, rf, gf, bf, cf;  
    /* First reading */  
    tcs.getRawData(&r, &g, &b, &c);  
    r1=r;  
    delay(500);  
    /* 2nd reading */  
    tcs.getRawData(&r, &g, &b, &c);  
    r2=r;  
    delay(500);  
    /* 3rd reading */  
    tcs.getRawData(&r, &g, &b, &c);  
    r3=r;  
    g3=g;  
    b3=b;  
    c3=c;  
    rf=r3/cR;  
    gf=g3/cG;  
    bf=b3/cB;  
    cf=c3/cC;  
    /*calculation the average */  
    avg=(r1+r2+r3)/3;  
    /*display result */  
    display.clear();  
}
```

```
display.setFont(ArialMT_Plain_16);
display.setTextAlignment(TEXT_ALIGN_LEFT);
display.drawString(0, 0, "R: " + String(rf, 0));
display.drawString(64, 0, "G: " + String(gf, 0));
display.drawString(0, 23, "B: " + String(bf, 0));
display.drawString(64, 23, "C: " + String(cf, 0));
display.setTextAlignment(TEXT_ALIGN_CENTER);
/* Test the last reading compare to the average reading,
the deviation should be less then % tolerance of the average,
example the difference tolerance of each reading is 2% from the average
If the not much deviation, then OK, of still large deviation, then reading and continue loop */

if ((abs(r3-avg)) < (diff*avg) )
{
display.drawString(64, 46, "OK ");
}
else
{
display.drawString(64, 46, "Reading...");
}
display.display();
}
```