LAPORAN PROJECT UTS

**“ALAT PENGHITUNG LOMBA BALAP LINE TRACER”**



**Disusun oleh :**

**Andi Mei Prasetyo Isworo**

**(1110171035)**

**Program Studi Teknik Elektronika**

**Departemen Elektro**

**Politeknik Elektronika Negeri Surabaya**

**Surabaya**

**2019**

**Daftar Isi**

[**I.** **Pendahuluan** 3](#_Toc23729588)

[**II.** **Tujuan** 3](#_Toc23729589)

[**III.** **Dasar Teori** 3](#_Toc23729590)

[**IV.** **Metodologi Percobaan** 3](#_Toc23729591)

[**A.** **Bahan Percobaan** 3](#_Toc23729592)

[**B.** **Desain Elektrik** 4](#_Toc23729593)

[**C.** **Desain Box** 7](#_Toc23729594)

[**D.** **Hasil Akhir Alat** 8](#_Toc23729595)

[**V.** **Analisa Data** 10](#_Toc23729596)

[**A.** **Program Counter disertai anti bouncing dan timeout pada button digital** 10](#_Toc23729597)

[**B.** **Program Counter disertai anti bouncing dan timeout pada Sensor analog** 10](#_Toc23729598)

[**C.** **Perhitungan Timer 2** 11](#_Toc23729599)

[**D.** **Perhitungan Timer 3** 11](#_Toc23729600)

[**E.** **Tugas yang dijalankan pada Timer 2** 11](#_Toc23729601)

[**F.** **Tugas yang dijalankan pada Timer 3** 11](#_Toc23729602)

[**G.** **Serial debugging dengan komunikasi UART** 11](#_Toc23729603)

[**VI.** **Kesimpulan** 12](#_Toc23729604)

[**VII.** **Link Repository GitHub** 12](#_Toc23729605)

# **Pendahuluan**

Pada dasarnya program yang baik adalah program yang dapat bekerja tidak selalu menggunakan clock tertinggi Processor, program yang bekerja pada clock tertinggi menyebabkan daya yang dibutuhkan juga besar, oleh karena itu dibuatlah program dengan jadwal dan tugas tertentu, pengaturan jdwal ditentukan oleh interrup timer yang telah dikehendaki oleh perancang. Dengan adanya manajemen waktu, kinerja processor tidak terlalu berat dan dapat mengemat penggunaan daya. Hal tersebut merupakan konsep dasar dari system operasi.

# **Tujuan**

* + - * + Mahasiswa dapat memahami algoritma program object counting
        + Mahasiswa dapat memehami dasar Sistem Operasi berbasis timer
        + Mahasiswa dapat membuat program tanpa delay
        + Mahasiswa dapat mengimplemtasikan materi yang telah didapat dengan membuat alat penghitung lap dan waktu pada lomba balap mobil
        + Mahasiswa dapat memahami filosofi System Embedded melalui project yang dibuat.

# **Dasar Teori**

Sistem Operasi merupakan program utama (Sekumpulan program kontrol atau alat pengendali) yang secara terpadu bertindak sebagai penghubung Software Aplikasi yang digunakan oleh user dengan hardware komputer.

**HARDWARE**

**BRAINWARE**

**SOFTWARE**

**Sistem Operasi**

**Aplikasi**

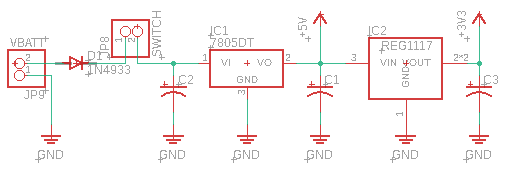
# **Metodologi Percobaan**

## **Bahan Percobaan**

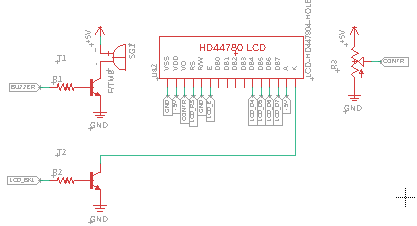
* + - 1. STM32F103C8 (*Blue Pill*) + ST LINK v2 1 buah
      2. Push Button 2 buah
      3. LCD karakter 20x4 1 buah
      4. Modul Dot Matrix 8x8 3 buah
      5. Modul Step Down 1 buah
      6. Photo diode 5mm 1 buah
      7. Infra Red 3mm 1 buah
      8. PCB Dot Matix 1 buah
      9. PCB Polos 1 buah
      10. Kabel jumper secukupnya
      11. Kabel pita secukupnya
      12. Kertas karton 3mm secukupnya
      13. Header Male & Female secukupnya
      14. Baut + Mur secukupnya

## **Desain Elektrik**

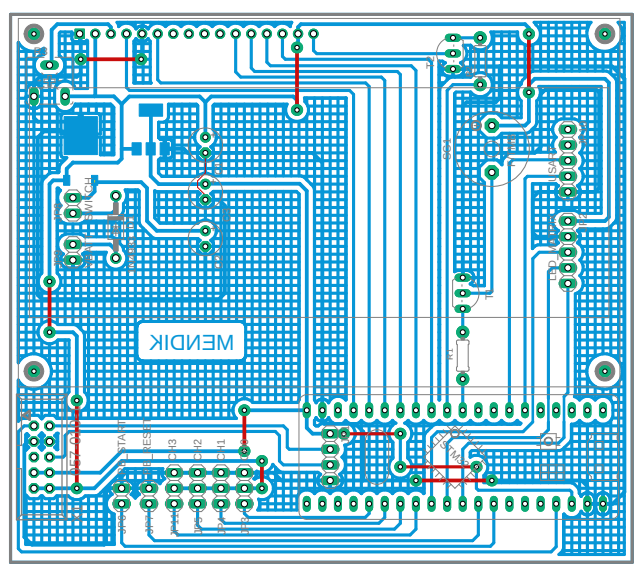
*Gambar 1.1 Konfigurasi Pin yang digunakan*



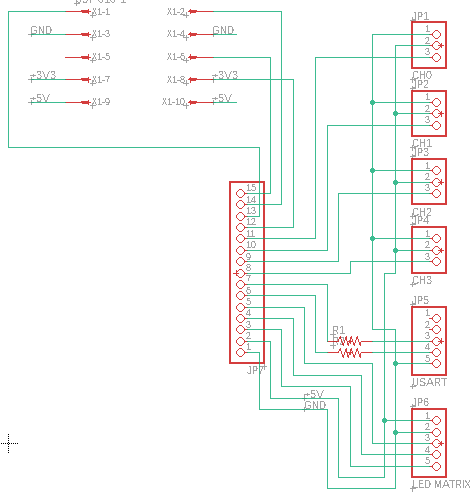
*Gambar 1.2 Rangkaian Regulator Power Supply*



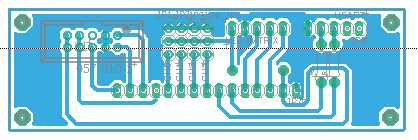
*Gambar 1.3 Rangkaian LCD dan Buzzer*



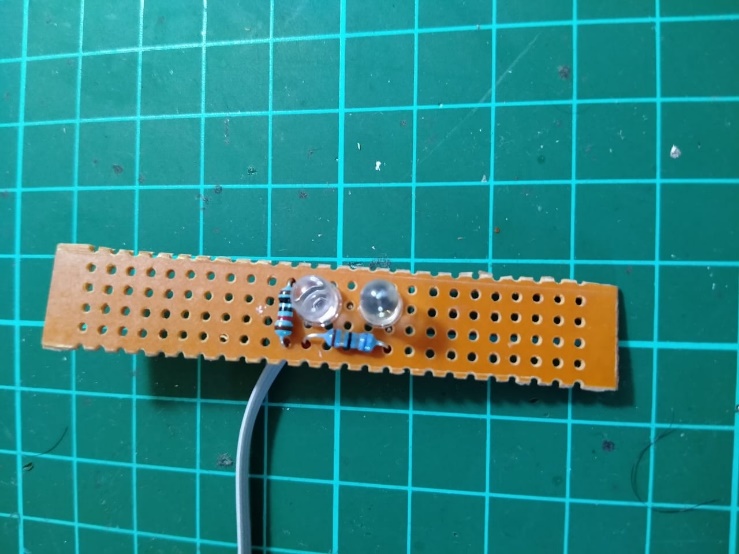
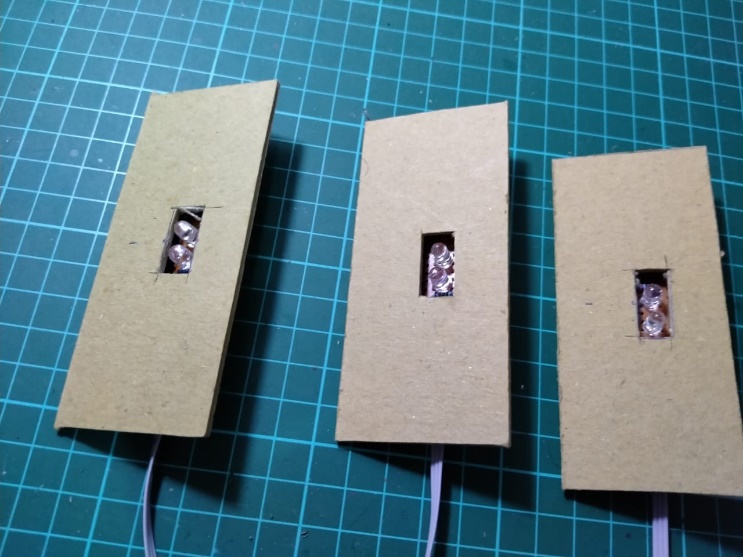
*Gambar 1.4 Layout Main PCB*



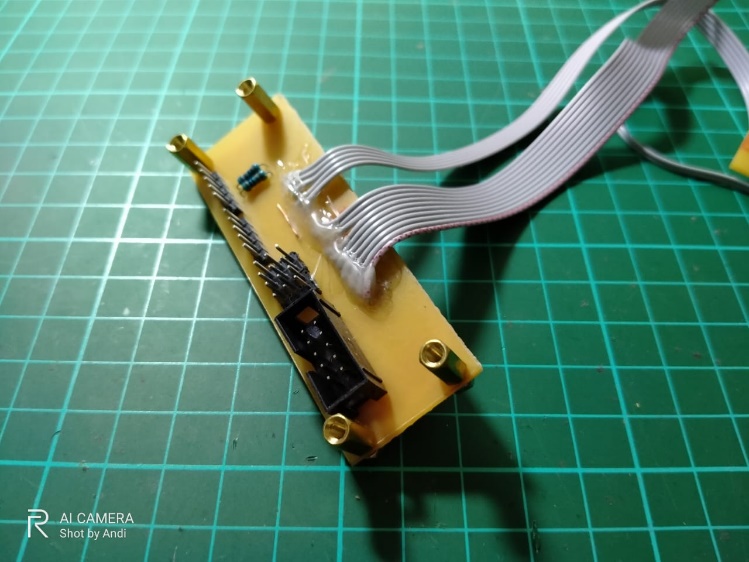
*Gambar 1.5 Rangkaian Panel Depan*

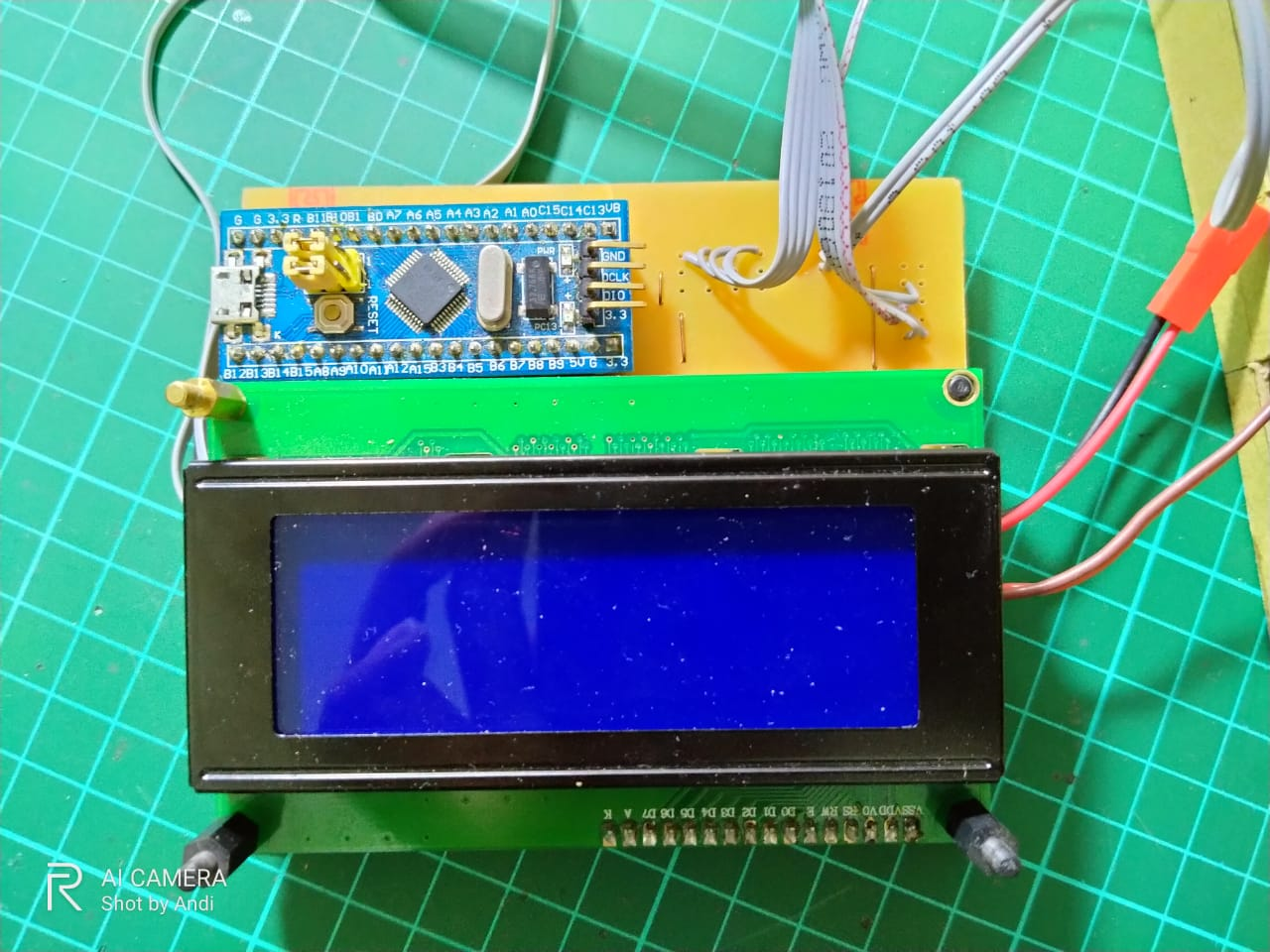


*Gambar 1.6 Layout Panel Depan*

*Gambar 1.7 Board Sensor*

*Gambar 1.8 Pemasangan Sesnsor*

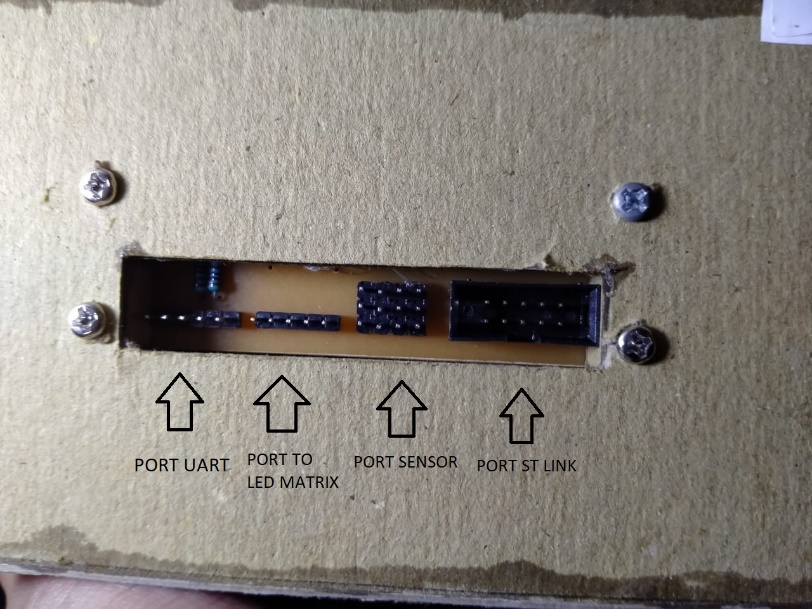
*Gambar 1.9 Board Panel Depan*



*Gambar 1.10 Main Board*

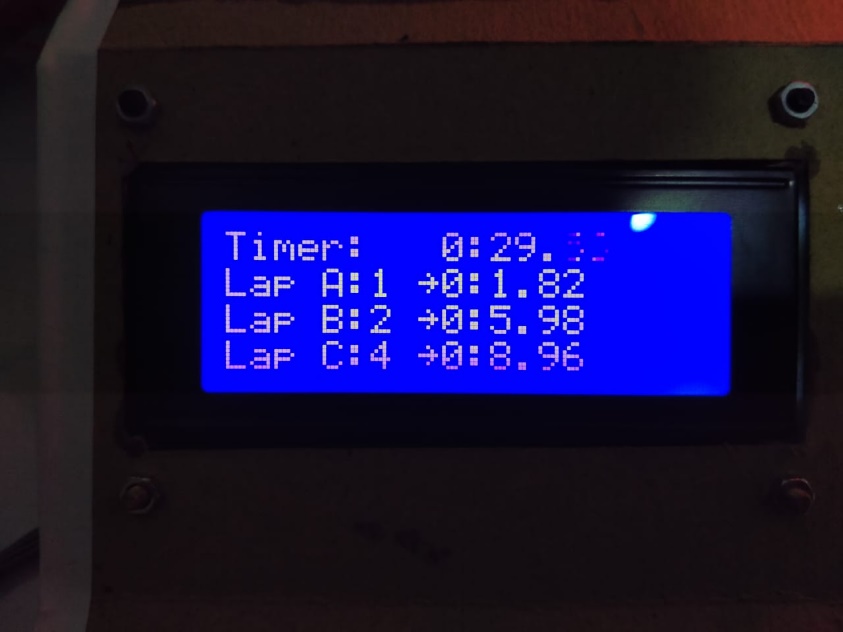
## **Desain Box**

*Gambar 2.1 Boks Monitor*

*Gambar 2.2 Boks Monitor*

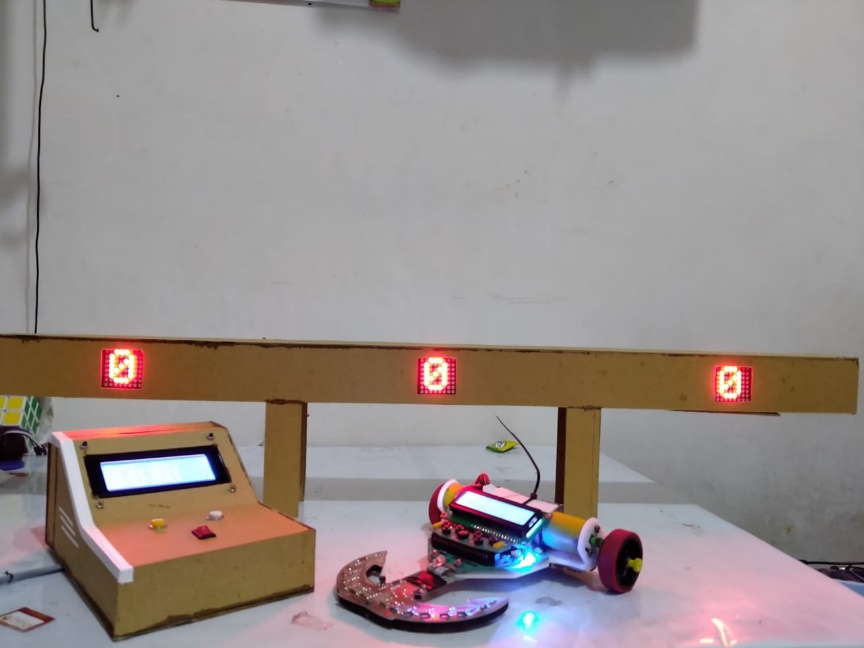
*Gambar 2.3 Port pada panel depan*

## **Hasil Akhir Alat**

*Gambar 2.4 Monitor ketika Dinyalakan*

*Gambar 2.5 Tampilan LCD Monitor*

*Gambar 2.6 Gerbang untuk mendeteksi robot*

*Gambar 2.7 Gerbang dilengkapi dengan display setiap Lap*

*Gambar 2.8 Hasil akhir keseluruhan*

# **Analisa Data**

Pada project ini, dibutuhkan program yang cukup kompleks, penulis mencoba menganalisa beberapa point-point penting dalam program diantaranya.

## **Program Counter disertai anti bouncing dan timeout pada button digital**

**if**(pushStart){

**if**(timeOut1++ > timeOutVal){

stopwatchEnable=0;

myTask\_ErrorMassage(1, "Button Start Error !");

myUART\_Println("ERROR : Button Start Timeout !");

myTask\_Buzzer(1);

}

**else**

bouncing1 = (bouncing1<<1)|1;

}

**else**{

timeOut1=0;

bouncing1 = bouncing1<<1;

}

**if**(bouncing1==3){

stopwatchEnable = !(stopwatchEnable);

myTask\_DisplayOut(stopwatchEnable);

}

Program tersebut merupakan potongan salah satu button yang digunakan. Program tersebut akan mengubah (toogle) variable stopwatchEnable setiap satu kali tombol ditekan, sekaligus akan mengupdate display, mulai dari LCD, Led Matrix, dan juga UART.

## **Program Counter disertai anti bouncing dan timeout pada Sensor analog**

/\* Sensor active when stopwatch enable \*/

**if**(stopwatchEnable){

/\* Sensor channel 0 \*/

**if**(adcVal[0] < Thrs[0]){

**if**(timeOut3++ > timeOutVal){

stopwatchEnable=0;

myLCD\_clear();

myTask\_ErrorMassage(1, "Sensor CH0 Error !");

myUART\_Println("ERROR : Sensor CH0 Timeout !");

myTask\_Buzzer(1);

}

**else**

bouncing3 = (bouncing3<<1)|1;

}

**else**{

timeOut3=0;

bouncing3 = bouncing3<<1;

}

**if**(bouncing3==3){

lap\_A++;

**if**(lap\_A > n) lap\_A = n;

miliSecond\_A[lap\_A] = miliSecond;

second\_A[lap\_A] = second;

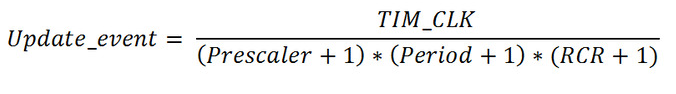
minute\_A[lap\_A] = minute;

myTask\_DisplayOut(stopwatchEnable);

}

Program tersebut merupakan potongan salah asatu sensor yang digunakan, pada dasarnya program tersebut sama dengan yang sebelumnya(digital), yang membedakan adalah program ini diperlukan konversi ADC yang selanjutnya dapat dibandingkan dengan variable Thrs. Thrs adalah variable yang nilainya didapat dari hasil kalibrasi sensor.

## **Perhitungan Timer 2**

Pada timer 2 penulis menginginkan Update\_event dilakukan setiap 10ms sekali untuk kerperluan stopwatch, TIM\_CLK yang telah disetting adalah 72Mhz, berdasarkan persamaan tersebut maka didapatkan nilai Prescaler = 35999 dan Periode = 19, karena auto reload penulis disable maka nilai RCR diabaikan.

## **Perhitungan Timer 3**

Pada timer 3 ini penulis setting lebih cepat yaitu sebesar 1ms untuk keperluan pembacaan sensor, button, dan update display. Berdasarkan persamaan tersebut didapatkan nilai Prescaler = 35999 dan Periode = 1, karean auto reload penulis disable maka nilai RCR diabaikan.

## **Tugas yang dijalankan pada Timer 2**

Pada timer 2 digunakan untuk menjalankan fungsi stopwatch yang mana nilai millisecond di*increment* setiap 10ms, ketika sudah mencapai 100 kali maka nilai Second akan ditambah.

**void** **myTask\_Stopwatch**(**void**){

**if**(stopwatchEnable){

miliSecond++;

**if**(miliSecond>99){

miliSecond=0;

second++;

**if**(second>59){

second=0;

minute++;

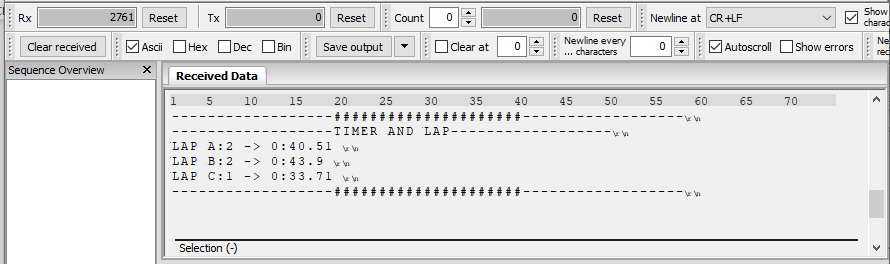
}

}

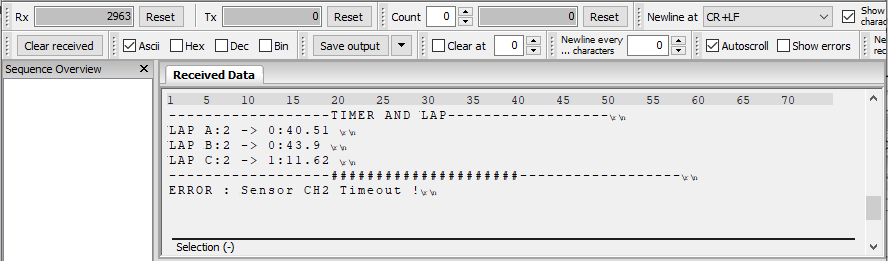
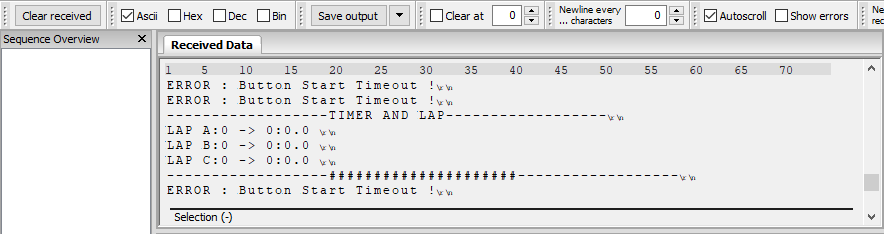
## **Tugas yang dijalankan pada Timer 3**

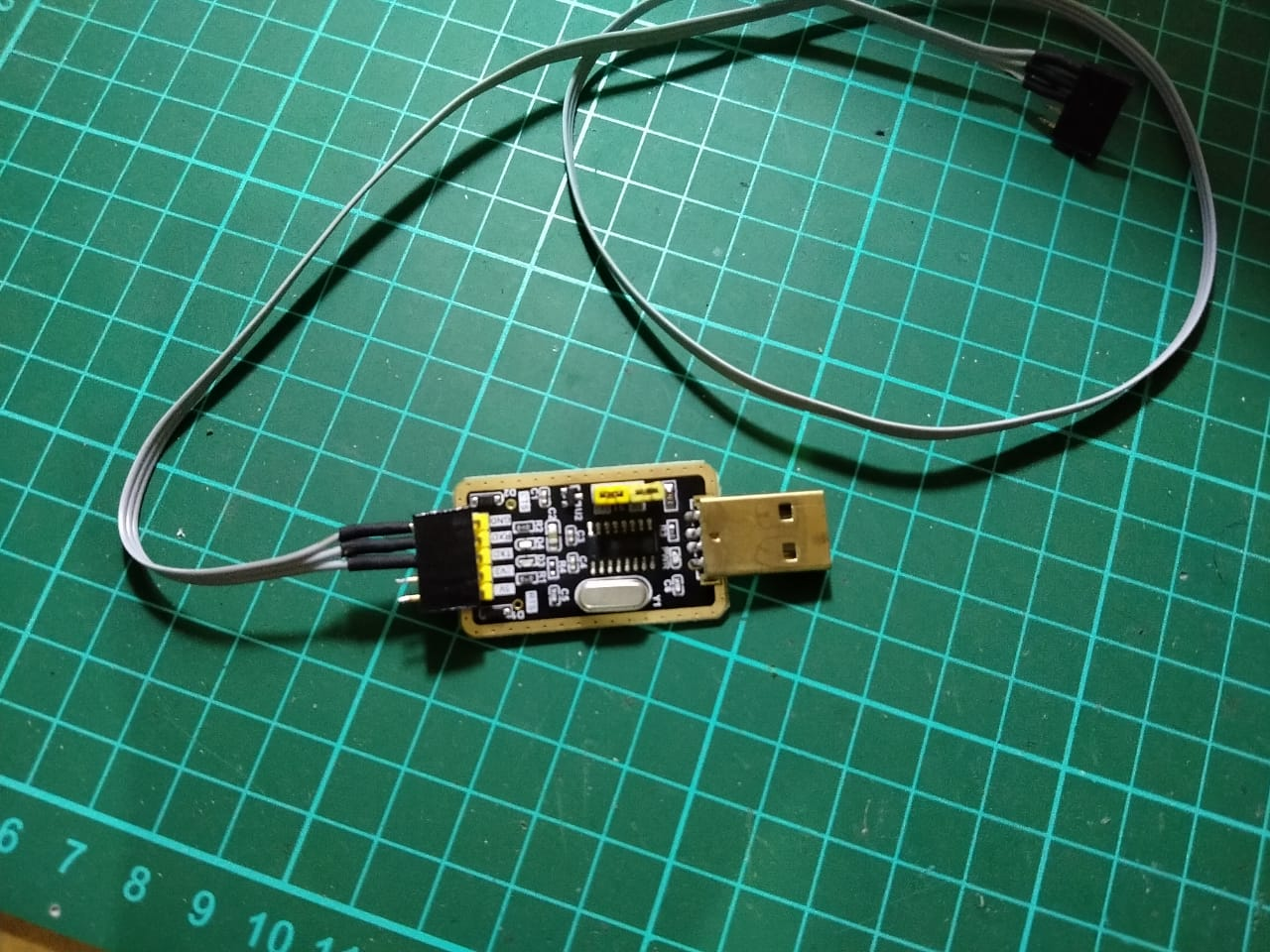
Pada timer 3 ini menangani banyak tugas diantaranya untuk membaca sensor analog, maupun digital, kalibrasi sensor, mengolah data Lap dan timer pada setipa lap, mengupdate display LCD, LedDot Matrix, dan juga untuk keperluan mengirim data pada UART

## **Serial debugging dengan komunikasi UART**

*Gambar 3.1 Tampilan UART debugger saat pertama kali dihidupkan*

*Gambar 3.2 Tampilan UART debugger saat nilai lap diupdate*

*Gambar 3.3 Tampilan UART debugger saat sensor analog timeout*

*Gambar 3.4 Tampilan UART debugger saat button digital timeout*

*Gambar 3.5 Modul USB to TTL*

Selain informasi lap, time, dan error ditampilkan pada LCD, informasi tersebut juga dikirimkan ke PC melalui komunikasi serial (UART). Penulis menggunakan USB to TTL untuk menjembetani antara mikrokontroler dan PC, termial yang digunakan adal ah HTerm.

# **Kesimpulan**

Berdasarkan hasil percobaan dan analisa yang telah penulis lakukan maka dapat ditarik kesimpulan diantaranya.

* Pembagian beberapa tugas berdasarkan waktu merupakan konsep dasar dari system operasi.
* Tugas yang dikerjaakan dapat diatur agar tidak berkerja pada frekuensi tertinggi dari clock internal guna menghemat penggunaan daya.
* Suatu tugas hendaknya tidak menggunakan delay yang membuat kacau jadwal.
* Suatu tugas tidak boleh terlalu lama, hal tersebut akan menggangu jadwal tugas yang lain.
* Algoritma antibouncing juga sangat diperlukan guna menghindari sensor mendeteksi *spike.*
* Mekanisme TimeOut juga diperlukan agar mempermudah developer maupun user untuk menganalisa kerusakan suatu system

# **Link Repository GitHub**

*https://github.com/AndiMei/UTS\_RaceCounter*