

## PEMANFAATAN RASPBERRY Pi DALAM MODEL SISTEM PEREKAM KECEPATAN KENDARAAN RODA

Agus Triyono<sup>1)</sup>, Noor Alam Hadiwijaya<sup>2)</sup>

<sup>1,2)</sup>Jurusan Teknologi Informasi Politeknik Negeri Samarinda

E-mail : [triyono@polnes.ac.id](mailto:triyono@polnes.ac.id)<sup>1)</sup>, [alamhadiwijaya@mail.ugm.ac.id](mailto:alamhadiwijaya@mail.ugm.ac.id)<sup>2)</sup>

### Abstract

*Raspberry Pi is a credit card-sized computer that is very widely used. Linux-based operating system, this device can serve the script and applications. By utilizing script on the Raspberry Pi and use to serial communication can be interfaced with microcontroller AT89S52. It's communication purpose to save the data that may produce by microcontroller such as timestamps and speed and stored to then SD Card as a storage.*

*Keywords: Raspberry pi, black box, speed, accesleration, microcontroller, script, proteus*

### 1. Pendahuluan

Semakin pesatnya produksi kendaraan bermotor dengan terjangkaunya harga kendaraan maka semakin tinggi pula tingkat kecelakaan. Tingginya angka kecelakaan disebabkan ketidakdisiplinan pengendara dalam mengendalikan kecepatan kendaraan bermotor. Dalam Undang-Undang Lalu Lintas dan Angkutan Jalan (LLAJ) No.22 Tahun 1999 Pasal 21 mengamanatkan agar kecepatan kendaraan dibatasi, pada Ayat 3 dituliskan atas pertimbangan keselamatan atau pertimbangan khusus lainnya, pemerintah daerah dapat menetapkan batas kecepatan paling tinggi setempat yang harus dinyatakan dengan rambu lalu lintas.

Diluar negeri pada jalan tertentu dipasang sensor dan kamera untuk menangkap pengemudi yang melebihi batas kecepatan. Untuk Indonesia, sangat sulit diterapkan hal demikian, terutama untuk pengendara bermotor roda dua.

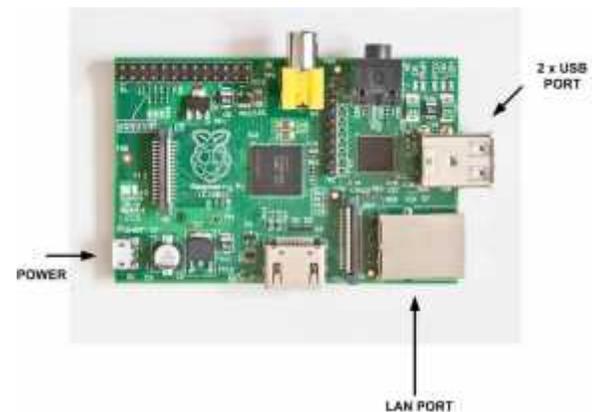
Kendaraan roda dua sangatlah sulit membatasi kecepatan pada daerah tertentu karena sangat fleksibel dalam memacu kecepatan dimanapun dan kapanpun dibandingkan dengan kendaraan roda empat. Hal inilah yang menyebabkan tingginya angka kecelakaan untuk kendaraan roda dua dibanding roda 4.

Untuk keperluan bukti pelanggaran kecepatan pada kendaraan roda dua maka diperlukan sebuah alat yang dapat merekam kecepatan secara simultan, alat ini berfungsi mirip seperti black box pada pesawat terbang. Alat ini juga kemudian dapat juga digunakan untuk keperluan analisa dari sebuah kecelakaan.

### 2. Raspberry Pi

Raspberry Pi adalah komputer dengan ukuran sekecil kartu kredit dan dengan kemampuan komputer.

Menggunakan prosesor ARM11 700 MHz dan RAM 512 MB serta perangkat tambahan seperti LAN, USB, GPIO, HDMI, 3.5 mm Audio Jack, RCA Jack untuk RGB Video dan Micro SD hingga 8 GB<sup>3</sup> dapat digunakan sebagai komputer kecil berbasis sistem operasi linux. Micro SD berperan sebagai HDD (Hard Disk Drive) sehingga data dan operating dapat disimpan.



Gambar 1. Raspberry Pi Board

### 3. Operating System

Raspberry Pi menggunakan prosesor dengan arsitektur ARM 11 700 MHz yang dibuat oleh perusahaan Broadcom dan mempunyai type prosesor BCM2835 yang dapat menggunakan beberapa sistem operasi antara lain : Raspbian,

Ubuntu Mate, Snappy Ubuntu Core, OSMC, OpenElec, Pinet, atau RISC OS.

Namun yang resmi dan banyak digunakan adalah Raspbian yaitu sebagai *Foundation's Official Supported Operating System*<sup>3</sup>.

Raspbian mempunyai karakteristik hampir sama dengan Linux Debian, namun dalam penggunaannya baik sistem file dan perintah-perintahnya tidak berbeda sama sekali dengan Linux Debian pada komputer berarsitektur x86.

Pengoperasian raspbian ini tidak berbeda dengan linux Debian yaitu menggunakan dua cara antara lain melalui GUI (*Graphical User Interface*) atau Terminal (*Text*). Sistem konfigurasinya pun tidak berbeda dengan Debian.

Dengan demikian Raspberry Pi menawarkan *flexibilitas* pengembangan yang lebih luas dengan mengandalkan *system open source* yang bebas biaya dan dapat dikembangkan oleh siapapun.

#### 4. UART

UART (*Universal Asynchronous Receiver Transmitter*) adalah salah satu komunikasi secara serial. Mengambil *byte* data dan mengirimkan mentransmisikan bit individu secara berurutan<sup>1</sup>.

Pada pemrograman UART pada hampir semua pemroses selalu menggunakan sistem interupsi tertentu untuk menerima bit demi bit dari *port* serial. Dengan menggunakan *Trailing Edge signal*, pemroses dapat diinterupsi untuk menerima bit demi bit dari receiver serial *port*.

Rangkaian biner yang dikirimkan akan membentuk satu *byte* informasi yang diterima penerima sehingga dengan kata lain *serial port* mengirimkan data dalam bentuk per-byte, dengan tanda *start bit* dan *stop bit* serta *baud rate* sebagai parameter konfigurasi.

#### 5. Device File

Pada sistem operasi linux, komunikasi dengan perangkat keras ditulis melalui sistem file tertentu yaitu dibawah directory `/dev`, ada ratusan *device* yang dapat dihubungi melalui *directory* ini.

Ketika USB dipasang maka sistem operasi akan menunjuk secara otomatis bahwa USB dapat dihubungi sebagai `/dev/ttyUSB0` atau `/dev/ttyUSB1`<sup>2</sup>, hal ini tergantung apakah *device driver* sudah terpasang atau belum.

Untuk USB to Serial Converter dengan *device driver* pl2303 pada linux Raspberry Pi

sudah ada didalam kernel, sehingga tidak memerlukan instalasi dan konfigurasi.

#### 6. USB Port Sebagai Converter

*Universal Serial Bus* (USB) dalam hal ini digunakan sebagai converter untuk mengkonversikan protocol RS-232 ke USB sehingga dapat berkomunikasi dua arah. Walaupun Raspberry Pi memiliki *port* khusus untuk serial *port* tanpa melalui konverter, namun hal ini, tidak dilakukan karena masih menggunakan level TTL, hal ini berpengaruh pada jarak transmisi data yang terbatas.

Dengan USB Converter protocol RS-232 dapat diterapkan dengan jarak maksimum transmisi data kurang dari 100 meter. Namun demikian pada mikrokontroler diperlukan IC MAX-232 untuk menyesuaikan protokol, karena mikrokontroler hanya mempunyai level TTL pada serial *port* yang sudah disediakan<sup>5</sup>.



Gambar 2. USB to Serial Converter

Pada Gambar 2 adalah perangkat *usb to serial converter* dengan *driver* yang dipasangkan pada raspberry pi menggunakan *CH-341 driver*.

#### 7. Microcontroller AT89C52

AT89C51 adalah mikrokontroler yang kompatibel dengan intel 8051 yaitu mikrokontroler 8 bit yang dibuat oleh Atmel Corporation.

Mempunyai struktur 40 pin, dengan 4 *port* (P0 hingga P3), dengan *memory reprogrammable flash* 4 KB<sup>5</sup>.

Terdapat komunikasi secara serial yaitu pada pin 10 sebagai RXD (Receiver) dan pin 11 sebagai TXD (Transmitter). Kedua pin ini masih menggunakan level TTL sebagai level tegangan komunikasinya.

Membutuhkan unit *converter* untuk dapat berkomunikasi dengan RS-232, maka dibutuhkan

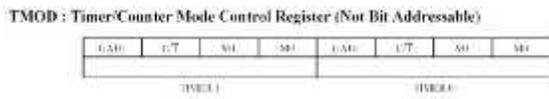
chip MAX-232 sebagai converter dari TTL ke RS-232.

**8. Timer dan Counter**

Disediakan tiga buah *timer* dan *counter* pada mikrokontroler AT89C52, yaitu T0, T1 dan T2. Fungsi *timer / counter* ada 3 yaitu :

1. *Interval Timing*
2. *Event Counting*
3. *Baud Rate Generation*

Timer menggunakan *Special Function Register* (SFR) untuk pengaturannya yaitu : TMOD, TCON, TH0/TL0, TH1/TL1, TH2/TL2.



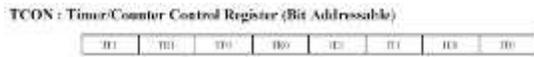
**Gambar 3.** Register TMOD

**TMOD** terdiri dari 8 bit yang masing-masing dapat digunakan untuk membuka mengaktifkan timer/counter, penggunaan timer0 atau timer1, dan mode timer.

*Mode timer* dibagai menjadi 3yaitu :

1. Mode 0 atau 13 bit.
2. Mode 1 atau 16 bit
3. Mode 2 atau 8 bit
4. Mode 3 atau 8 bit hanya timer 0

Keempat mode dapat dikonfigurasi pada SFR TMOD.



**Gambar 4.** Register TCON

**TCON** terdiri dari 8 bit dan addressable. Register untuk mengendalika interrupt timer, timer flag, timer run, dan external interrupt dengan membuat bit-bit yang terdapat dalam register berisi 1 (*set*). Untuk dapat mengaktifka masing-masing timer TR0 atau TR1 harus bernilai = 1.

**9. Kecepatan**

Menurut Lauren iu Dimitriu (2007) dalam journalnya memaparkan prinsip pengukuran kecepatan kendaraan dengan persamaan sebagai berikut :

$$t_m = T/p \tag{1}$$

dengan :  $t_m$  = Waktu yang diukur (*seconds*)

$T$  = Waktu dalam satu periode (*seconds*)  
 $P$  = Banyaknya gerigi (buah)

Sistem akan mengukur  $t_m$  dengan satu waktu tertentu (misalnya 1 Mhz sehingga didapat  $T_{ck} = 1 \mu S$  clock pulse dan dengan 16 bit counter atau  $2^{16} = 65535 \mu S$  maksimum untuk waktu interval untuk  $t_m$ . Hasil dari perhitungan adalah besarnya  $N_m$  dengan :

$$N_m = t_m / T_{ck} \tag{2}$$

dengan:  $N_m$  = Interval (*seconds*)

$T_m$  = Waktu yang diukur (*seconds*)

$T_{ck}$  = Waktu Clocking (*seconds*)

Persamaan diatas dapat ditulis :

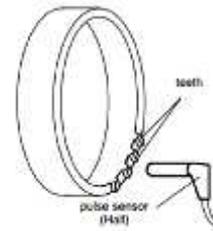
$$t_m = N_m \cdot T_{ck} \tag{3}$$

Persamaan (2) dan (3) didapat :

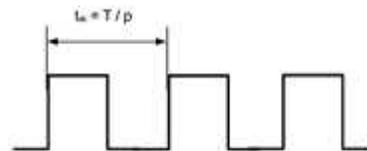
$$N_m \cdot T_{ck} = \frac{T}{P}, T = P \cdot N_m \cdot T_{ck} \tag{4}$$

Perhitungan kecepatan kendaraan dapat dihitung berdasarkan  $N_m$ .

Prinsip perhitungan kecepatan kendaraan dengan memanfaatkan diaman D roda sebagai parameter dan dalam satu putaran atau yang disebut dengan  $f$  atau frekuensi.



**Gambar 5.** Sensor Pengukur Kecepatan Roda



**Gambar 6.** Sinyal Keluaran Sensor

$$f = \frac{1}{T} = \frac{1}{P \cdot N_m \cdot T_{ck}} \tag{5}$$

dengan :

$f$  = frekuensi (Hertz)

$P$  = Banyaknya Gerigi Roda

$N_m$  = Interval (detik)

$T_{ck}$  = Waktu Clocking (detik)

Dari formulasi diatas dapat dilakukan perhitungan melalui mikrokontroler sebagai embedded system dan Raspberries Pi sebagai media penyimpanan.

### 10. Perancangan dalam Block Diagram



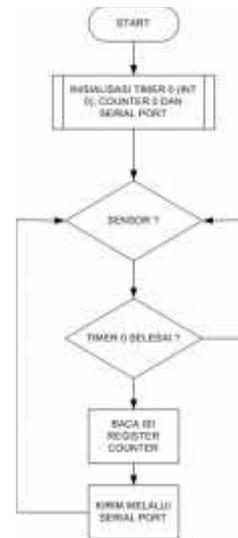
Gambar 7. Block Diagram

Pada rancangan Gambar 4 digunakan simulasi output dari sensor yaitu berupa function generator berupa pulsa-pulsa digital dengan frekuensi bervariasi (linear dengan kecepatan) dan amplitude dengan tegangan +Volts. Sangat tepat pengujian pemrograman dilakukan menggunakan simulasi proteus dengan tingkat keakurasian tinggi.

Modul penghitung kecepatan yaitu berupa rangkaian mikrokontroler yang berfungsi menghitung sinyal dari function generator dalam interval 500 milidetik. Interval 500 milidetik tersebut dihasilkan oleh timer 0. Setial interval 500 milidetik pencuplikan dilakukan.

Raspberry Pi atau sebagai pooler berkomunikasi melalui serial port ke mikrokontroler. Data yang dikirimkan oleh mikrokontroler per 500 milidetik disimpan kedalam SD card raspberry pi.

### 11. Flow Chart Program



Gambar 8. Flow Chart Program Mikrokontroler

Gambar 8 menunjukkan proses awal program adalah proses inisialisasi. Menggunakan timer 0 yang dikonfigurasi untuk 500 milidetik interval waktu untuk pencuplikan hasil penghitungan pulsa. Sedangkan timer 1 digunakan untuk mengatur kecepatan kirim (baud rate) pada serial port.

Timer 0 menghasilkan interupsi sepanjang 500 milidetik, sedangkan kemampuan register TH dan TL untuk menampung pencacah maksimal 65.536, ketika register TH0 dan TL0 diisi dengan 05H, maka cacahan akan overflow 50 mikrosecond. Hal ini tidak mencukupi untuk interval 500 miliseconds, maka diperlukan subroutine delay.

Subroutine delay ini akan menunda selama waktu yang dibutuhkan sehingga 500 miliseconds tercapai ketika interupsi timer terjadi.

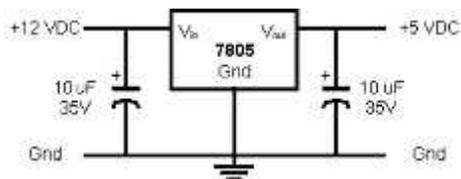
### 12. Power Supply Unit

Tegangan yang akan diberikan ada dua jenis yaitu :

1. Tegangan untuk mikrokontroler
2. Tegangan untuk raspberry pi

Tegangan untuk mikrokontroler akan diberikan ke mikrokontroler sebesar 5 V dan 1

Ampere, tegangan ini cukup untuk men-supply dua komponen aktif berupa integrated circuit AT89C89 dan MAX-232. Pada datasheet tegangan yang dibutuhkan AT89C52 mempunyai  $5\text{ V} \pm 20\%$  dengan arus sebesar  $10\text{ mA}$  [datasheet at89C52], sedangkan MAX-232 mempunyai tegangan  $5\text{ V} \pm 10\%$  dengan arus yang diperlukan sebesar  $10\text{ mA}$  [datasheet max-232], sehingga daya total yang dibutuhkan hanya sebesar  $20\text{ mA}$ .



Gambar 9. Power Supply Unit

Integrated Circuit LM7805 akan digunakan sebagai supply tegangan dengan arus yang dihasilkan sebesar  $1\text{ Ampere}$  [datasheet LM7805], sehingga arus dibutuhkan oleh modul penghitung kecepatan lebih dari cukup yaitu hanya  $20\text{ mA}$  dan yang tersisa  $980\text{ mA}$ . IC (*integrated circuit*) ini akan panas (*overheat*) jika terjadi *short circuit* atau daya beban mendekati daya maksimum yang dihasilkan LM7805, namun hal ini tidak akan terjadi karena daya beban sebesar  $100\text{ mWatt}$  masih sangat jauh dari yang tersedia sejumlah  $5\text{ Watt}$  atau  $5000\text{ miliWatt}$ .

Raspberry pi memerlukan PSU yang tergantung dari beban, namun beban minimal dari 1 unit raspberry sebesar  $700\text{ mA}$  dengan tegangan  $5\text{ V}$ . [raspberry datasheet] Namun untuk menjaga kestabilan daya raspberry pi akan menggunakan switching power supply  $5\text{ Volt}$  dengan arus hingga  $2\text{ Ampere}$ . PSU ini sudah dalam bentuk modul standar industry, sehingga tidak memerlukan modifikasi komponen.

### 13. Perangkat Lunak

Menggunakan bahasa rakitan keluarga Intel MCS-51, program di *compile* dengan *compiler* ASMX sehingga menjadi bahasa mesin, bahasa mesin berupa Hexa file kemudian di *embedded*kan kedalam mikrokontroler menggunakan *tool programmer*.

Sedangkan modul raspberry menggunakan beberapa aplikasi yaitu menggunakan aplikasi

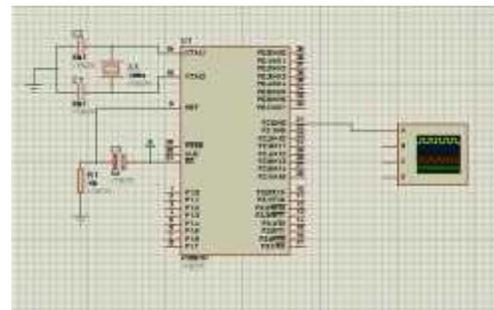
minicom dan scripting untuk dapat merekam data yang dikirim oleh mikrokontroler.

### 14. Pengujian

Menggunakan Proteus untuk menguji rangkaian *hardware* dan *software*. Proteus memiliki keakuratan tinggi dalam mensimulasikan berbagai macam rangkaian elektronik yang berbasis analog maupun digital.

Dilakukan beberapa pengujian pada mikrokontroler dalam hal program yaitu :

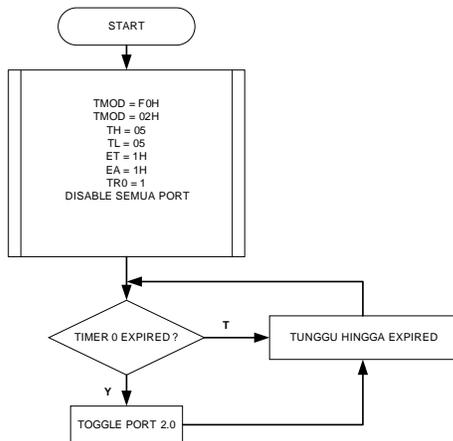
1. Pengujian Timer
2. Pengujian Serial Port
3. Pengujian Sensor ( Function Generator )



Gambar 10. Pengujian Timer Mikrokontroler

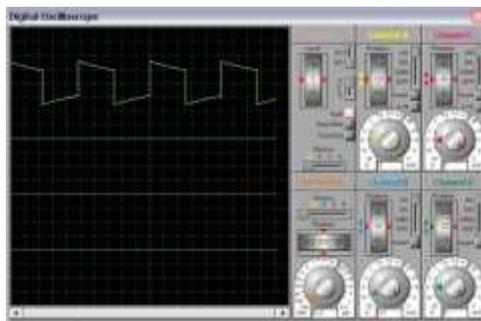
Pengujian Timer dilakukan dengan membuat program dengan keluaran berupa port yang dimonitor dengan oscilloscope (Gambar 10).

Mikrokontroler akan mengeluarkan secara periodik sebesar  $500\text{ milidetik}$ . Pulsa tersebut merepresentasikan bahwa timer dalam keadaan aktif karena di trigger oleh program timer. Gambar 4.8 adalah rangkaian mikrokontroler dan oscilloscope dengan keluaran pulsa sebesar  $500\text{ milisecond}$ . Program timer ini dimanfaatkan untuk mencuplik hasil penghitungan yang dilakukan oleh counter secara periodik. Jadi selama  $500\text{ milidetik}$  program akan mengambil hasil dari perhitungan.



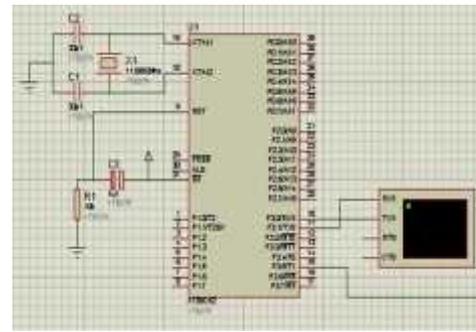
**Gambar 11.** Flow Chart Program Timer

Pada proses inisialisasi Gambar 11 diatas menggunakan *timer 0* dengan mode 2 yaitu 16 bit *counter*, dengan nilai pada *TMOD* = 02H. Memberi nilai *TH* dan *TL* masing-masing 05H akan membuat *register* ini terisi penuh selama 250 mikrodetik. Dengan demikian dibutuhkan pewaktuan yang lebih lama dengan memberi *delay* ketika terjadi interupsi pada *Interrupt Service Routine*.



**Gambar 12.** Hasil Pengujian Timer 0

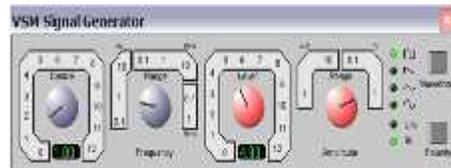
Menggunakan kanal A sebagai masukan dari simulasi *oscilloscope* dengan, *time division* = 200 milidetik, dan *amplitude division* sebesar 2 Volts, gelombang yang dihasilkan sebesar 500 milidetik dan amplitude tegangan sebesar 5 Volts, dihasilkan oleh program timer. Ini membuktikan program berjalan dengan baik.



**Gambar 13.** Pengujian Serial Port

Pengujian program serial port juga dilakukan dengan simulasi proteus untuk membuktikan serial port pada mikrokontroler bekerja dengan baik.

Rangkaian simulasi mikrokontroler pada Gambar 13 menggunakan *Virtual Terminal* yang disediakan oleh proteus. Untuk membuktikan bahwa program serial bekerja dengan baik dilakukan dengan mengintegrasikan dengan program counter. Sinyal yang diumpankan pada mikrokontroler menggunakan function generator yang disediakan juga oleh proteus untuk mengenerate pulsa-pulsa yang merepresentasikan keluaran dari sensor.

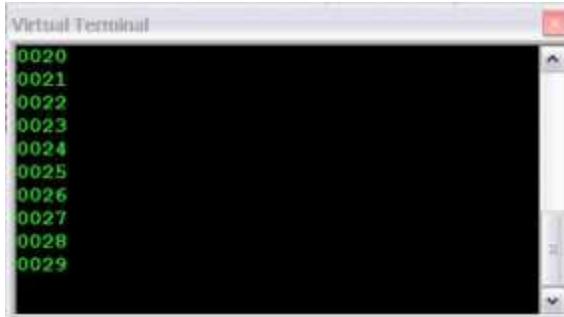


**Gambar 14.** VSM Signal Generator

*Function Generator* dikonfigurasi dengan frekuensi yang bervariasi dan *amplitude* yang tetap yaitu 5 Volts. *Function generator* dihubungkan dengan pin 3.5 pada mikrokontroler yaitu sebagai pin yang berfungsi sebagai penerima sinyal untuk dihitung.

Jalur yang menghubungkan antara pin 3.5 pada mikrokontroler ke *function generator* di-tap dengan *oscilloscope* untuk memastikan sinyal yang keluar dari function generator keluar sesuai dengan kriteria yaitu 5 volts *amplitude* dengan frekuensi yang bervariasi. Untuk pengujian serial port ini hasil yang diharapkan adalah nilai-nilai yang keluar pada virtual terminal yang dihasilkan

setiap ada kejadian trailing edge. Nilai count akan naik ketika port 3.5 bertransisi dari nilai +5 Volts ke 0 Volts.



**Gambar 14.** Hasil Pengujian Counter dan Serial Port

Nilai-nilai yang muncul adalah nilai sekuensial, karena belum bekerjanya integrasi program timer dan program *counter*. Terdapat konflik pada program, yaitu pendefinisian function yang mencuplik sinyal dari pin 3.5 untuk dihitung jumlah dengan interval tertentu dengan sistem interupsi yang men-generate 500 milidetik, sehingga diharapkan didapat hasil yang tidak sekuensial melainkan nilai-nilai yang bervariasi pada *virtual terminal*.

## 15. Kesimpulan

Pada rangkaian pengujian melalui simulasi proteus, semua proses uji berhasil dengan baik dengan nilai yang diharapkan.

Sistem belum dapat diintegrasikan karena belum menyatunya program timer interval dan program counter sehingga data yang ditampilkan melalui serial port data sekuensial.

## Daftar Pustaka

Adam Osborne, *An Introduction to Microcomputers Volume 1: Basic Concepts*, Osborne-McGraw Hill Berkeley California USA, 1980 ISBN 0-931988-34-9 pp. 116-126

**Datasheet AT89C52** (1999), “8 Bit Microcontroller with 8 K Bytes Flash”, Atmel Corporation

**Datasheet LM7805** (2014), “LM78XX / LM78XXA 3 Terminal 1 A Positif Voltage Regulator”, Fairchild Corporation

**Datasheet MAX232** (September 2014), “MAX232x Dual EIA-232 Drivers/Receivers”, Texas Instruments

*EIA standard RS-232-C: Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange*. Washington: Electronic Industries Association. Engineering Dept. 1969. [OCLC 38637094](#).

**Heath, Steve (2003).** *Embedded systems design*. EDN series for design engineers (2 ed.). Newnes. p. 2. [ISBN 978-0-7506-5546-0](#).

<http://www.veritecheng.com/eaton-vorad-collision-warning-system/>, 28 Agustus 2015

[http://mech-ing.com/journal/Archive/4-5-2007/2.Tehnologii/2.4\\_18.Dimitriu.tm07.pdf](http://mech-ing.com/journal/Archive/4-5-2007/2.Tehnologii/2.4_18.Dimitriu.tm07.pdf), 2 September 2015

<http://www.jtle.net/uploadfile/2014/0604/20140604031610638.pdf>, 2 September 2015

[http://www.ijirset.com/upload/2015/icmeet/76\\_43\\_MEAE520\\_NEW.pdf](http://www.ijirset.com/upload/2015/icmeet/76_43_MEAE520_NEW.pdf), 2 September 2015

[http://en.wikipedia.org/wiki/Embedded\\_system#cite\\_note-micro\\_controller-6](http://en.wikipedia.org/wiki/Embedded_system#cite_note-micro_controller-6), 3 September 2015

<http://www.immersa-lab.com/jenis-jenis-mikrokontroler.htm>, 30 Oktober 2015

<http://retired.beyondlogic.org/serial/serial1.htm#40>, 02 November 2015

**Michael Barr; Anthony J. Massa** (2006). "Introduction". *Programming embedded*

*systems: with C and GNU development  
tools*. O'Reilly. pp. 1-2.[ISBN 978-0-596-  
00983-0](#).

