

DEVAIS INPUT/OUTPUT ANTAR MUKA BERBASIS UNIVERSAL SERIAL BUS (USB)

Agus Pracoyo¹

Jurusan Teknik Elektro¹, Politeknik Negeri Malang
coyo_k5ya@yahoo.co.id

ABSTRACT

Almost all modern computers provide a USB port (Universal Serial Bus) for access to the device input / output external. This phenomenon has forced the designers and developers full-custom devices, input / output is always based on the USB interface technique. A USB system consists of a controller (host PC) and several Device input / output

This research aims to design devices, input / output USB HID class (Human Interface Decive). Design based on the USB protocol version 1.1 and 2.0 as the software on the Device (firmware) and the host, as a controller selected PIC18F4550 microcontroller. The design of devices, input / output USB with analog and digital terminals that can diakes and controlled by the host

Testing is done using software USBTrace. The test results showed that the speed of transactions is achieved is 12 Mbps (full-speed) with 48 MHz MCU clock

Keyword: *computer interfacing, USB, devices I/O.*

ABSTRAK.

Hampir semua komputer modern dan laptop hanya menyediakan terminal USB (Universal Serial Bus) untuk keperluan akses devais input/output eksternal. Fenomena ini memaksa para perancang dan pengembang full-custom devais input/output selalu berbasis pada teknik antar muka USB. Sistem USB terdiri atas pengontrol (PC host) dan beberapa Devais input/output.

Penelitian ini bertujuan untuk merancang devais input/output USB kelas HID (Human Interface Decive). Perancangan berdasarkan protokol USB versi 1.1 dan 2.0 sebagai perangkat lunak pada sisi Devais (firmware) dan sisi host. Sebagai rangkaian pengontrol dipilih Microcontroller PIC18F4550. Hasil rancangan berupa devais input/output USB dengan terminal analog dan digital yang dapat diakes dan dikendalikan oleh host.

Pengujian dilakukan menggunakan perangkat lunak USBTrace. Hasil pengujian menunjukkan bahwa kecepatan transaksi yang dicapai adalah 12 Mbps (full-speed) dengan clock MCU 48 Mhz.

Kata kunci : *antarmuka komputer, USB, devais I/O*

PENDAHULUAN

a) *Universal Serial Bus(USB)*

Hampir semua komputer *modern* dan laptop hanya menyediakan terminal USB (*Universal Serial Bus*) untuk keparluan akses Devais input/output eksternal. Fenomena ini memaksa para perancang dan pengembang *full-custom* Devais *input/output* selalu berbasis pada teknik antar muka USB [3]. USB adalah standar bus serial sebagai perangkat penghubung

antara komputer dengan *peripheral* eksternal. Kelebihan utama antar muka USB adalah *hot plugable*, yang berarti Devais USB dapat di *plug* sementara PC dalam kondisi *running* dan mengkonfigurasi secara otomatis [1][2]

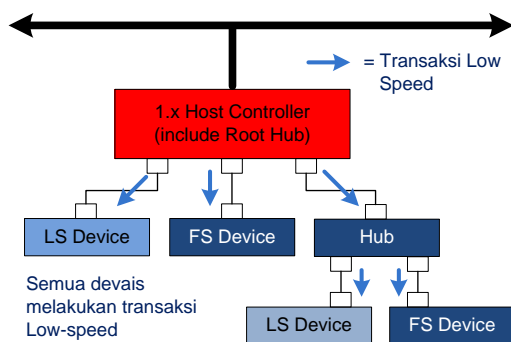
Penelitian ini merancang bangun devais input/output USB kelas HID yang dilengkapi dengan akses *peripheral* digital dan analog. Sebagai rangkaian kontroler dipergunakan *Microcontroller* Microchip PIC18F4550. Pemilihan *Microcontroller*

ini di dasarkan pada kekhususan fiturnya untuk menangani antar muka USB.

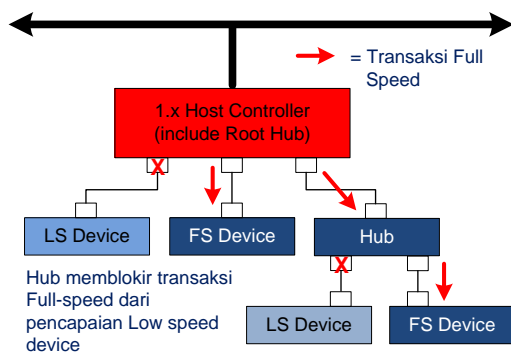
USB adalah suatu sistem *host-centric connectivity* dimana *host* mendikte pemakaian bus USB^[1]. Masing-masing Devais *input/output* pada bus ditandai dengan suatu alamat USB yang unik, dan tidak ada devais *slave* dapat menyatakan suatu *signal* pada *bus* sampai *host* memintanya.

USB dapat mendukung kecepatan transaksi 1.5Mbps (*low speed*) dan 12Mbps (*full-speed*). Transaksi *low-speed* ditunjukkan dalam Gambar 1. Transaksi data dapat mencapai ke seluruh devais yang terpasang, baik devais *full speed* atau *low speed*.

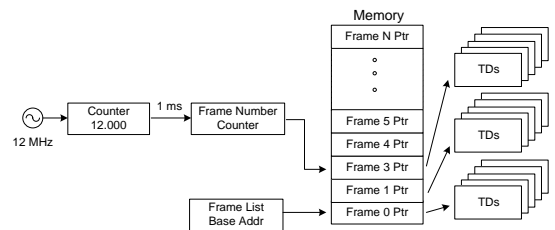
Transaksi *Full-speed* ditunjukkan dalam Gambar 2. Transaksi data hanya dapat dilakukan pada devais *full-speed*, karena devais *low-speed* tidak dapat melakukan transaksi *full-speed*.



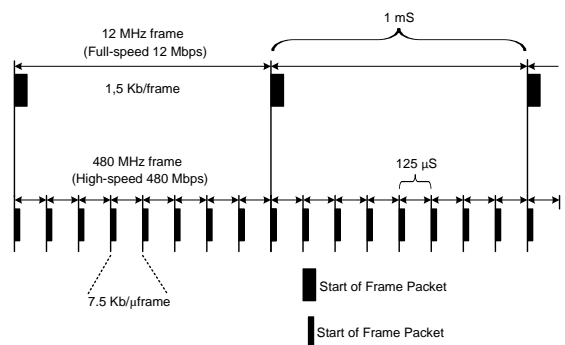
Gambar 1. Transaksi Devais *Low-speed*. [1]



Gambar 2. Transaksi Devais Full-speed. [1] Daftar transaksi pada USB disebut juga *frame*, satu (1) *frame* pada transaksi *full-speed*, dinyatakan oleh interval waktu selama 1ms pada saat *host* mengambil dan mengeksekusi data (*fetch and execute*) pada satu siklus terjadinya transfer *descriptor* dari *memori*. Untuk selanjutnya *controller* akan mengambil masing-masing daftar *frame* di dalam interval 1ms seperti ditunjukkan dalam Gambar 3 dan Gambar 4.

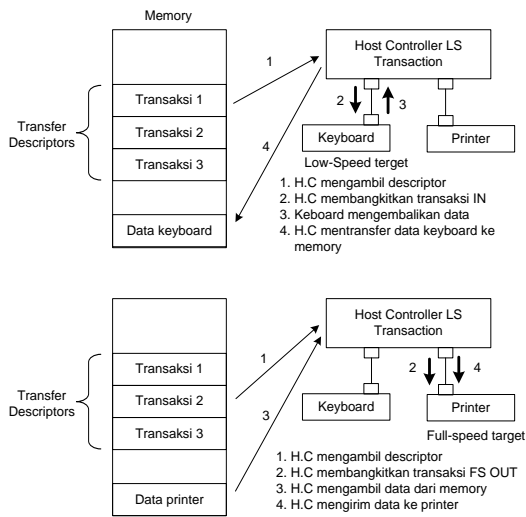


Gambar 3. Pembangkit Frame 1 ms. [1]



Gambar 4. Frame 1ms dan 125 μs. [1]

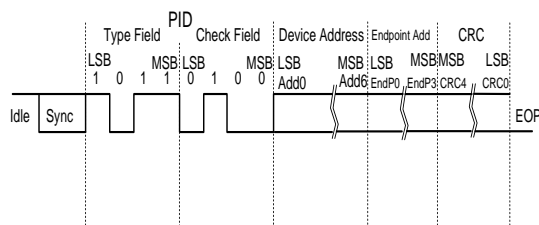
Masing-masing perpindahan *descriptor* menggambarkan suatu transaksi pada USB. USB versi 1.x dapat melakukan transaksi dengan kecepatan *low-speed* 1,5 Mbps dan *full-speed* 12 Mbps. USB versi 2.0 dapat melakukan transaksi dengan kecepatan seperti pada USB versi 1.x ditambah dengan transaksi *high-speed* 480 Mbps. Mekanisme konseptual transaksi pada USB dapat ditunjukkan dalam Gambar 5



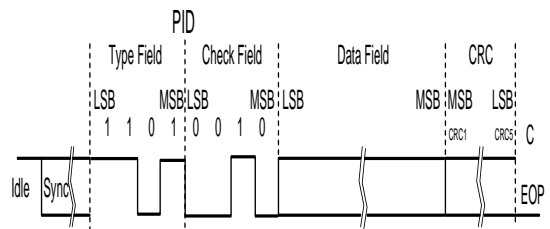
Gambar 5. Konseptual Transaksi pada USB.[1]

Paket data yang dipancarkan oleh bus USB terdiri dari beberapa bagian. Suatu paket data dimulai dengan pola sinkronisasi antara clock penerima dengan data. Untuk selanjutnya *byte-byte* data dari paket akan mengikutinya dan diakhiri oleh signal *End Of Packet (EOP)*.

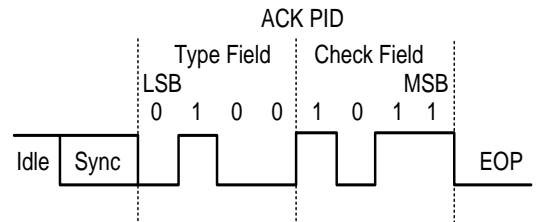
Gambar 6(a) menunjukkan *Token Packets* yang dimulai oleh *synchronisasi* kemudian diikuti oleh *Packet Identifier Data (PID)*. Panjang PID 8 bit yang terdiri atas *type field* 4 bit dan *check field* 4 bit. Ada empat (4) bentuk paket, yang berbasis pada PID yaitu *Token Packets*, *Data Packets*, *Handshake Packets*, dan *Special Packets*, seperti ditunjukkan oleh Gambar 6(a),6(b),6(c) dan 6(d).



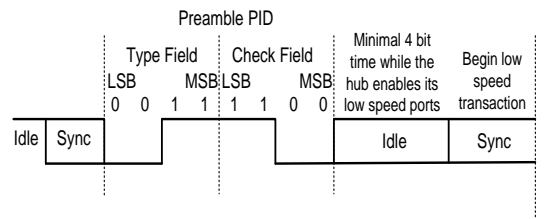
Gambar 6a. *Token Packets*



Gambar6b. *Data Packets*



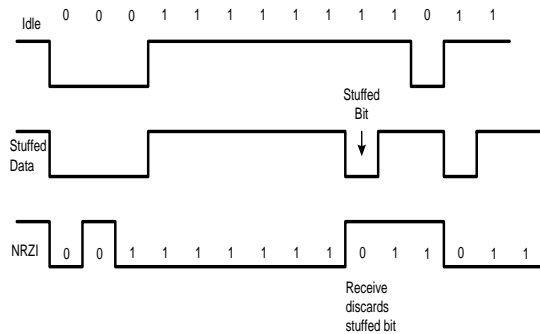
Gambar 6c. *Handshake Packets*



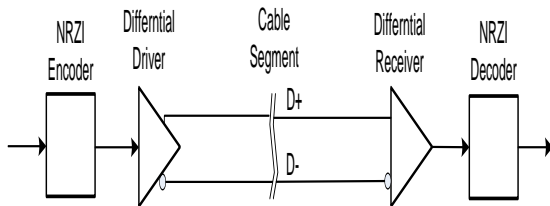
Gambar 6d. *Special Packets*

b) Sinyal pada USB

Informasi yang melintasi *USB cable* berdasarkan *encoding NRZI (Non-return to Zero Inverted)* dengan signaling diferensial seperti ditunjukkan dalam Gambar 7. Pada bit 0 diisikan setelah setiap enam yang berurutan di dalam aliran data untuk pemeliharaan sinkronisasi (*stuffed bit*). Sinyal USB dari sumber dikodekan menjadi NRZI dan dirubah dalam format diferensial dan melintasi kabel USB seperti ditunjukkan dalam Gambar 8.



Gambar 7. Data NRZI dan Stuffed Bit.[1]



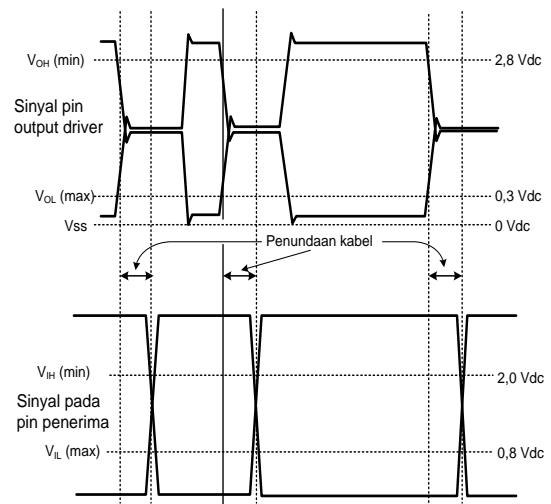
Gambar 8. NRZI Encoding dan Diferensial Signaling.[1][5]

Mengacu pada Gambar 8 di atas signal Diferensial “1” pada saat D+ lebih besar daripada D- dan Diferensial “0” adalah pada saat D- lebih besar daripada D+ seperti ditunjukkan dalam Gambar 9. Tegangan aktual ditetapkan untuk driver diferensial adalah [2][3]:

- a) Diferensial “1” = $D+ > V_{OH}(min)$ dan $D- < V_{OL}(max)$
- b) Diferensial “0” = $D- > V_{OH}(min)$ dan $D+ < V_{OL}(max)$

Level tegangan yang diterima harus terdeteksi sbb:

- a) Diferensial “1” = $(D+) - (D-) > 200m$ dan $D+ > V_{IH}(min)$
- b) Diferensial “0” = $(D-) - (D+) > 200m$ dan $D- > V_{IH}(min)$

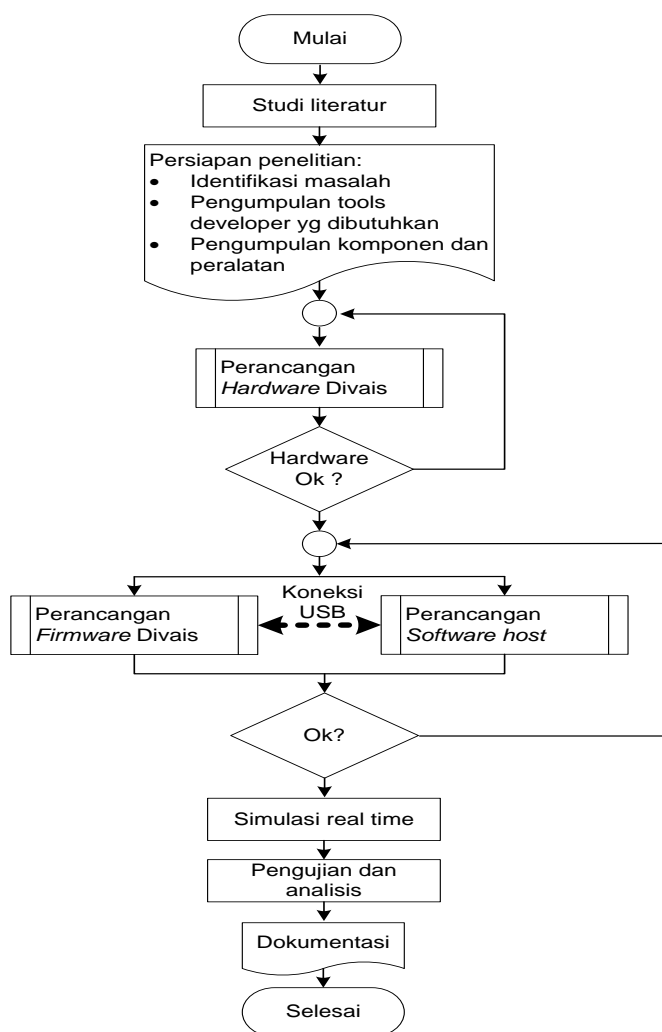


Gambar 9. Sinyal Diferensial.[1]

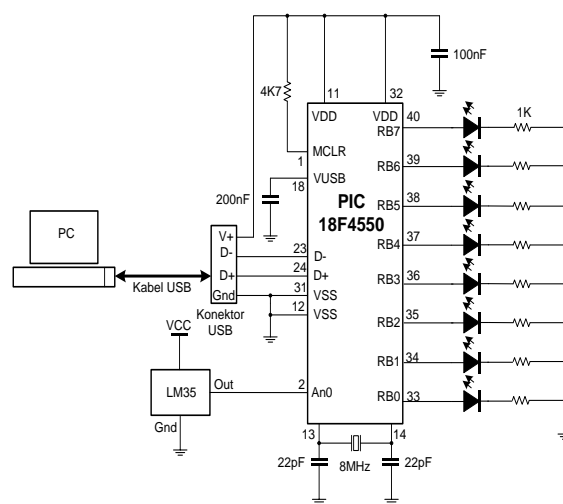
METODOLOGI

Penelitian dilakukan di Laboratorium Microprocessor Politeknik Negeri Malang. Bahan dan alat yang dipergunakan dalam penelitian ini antara lain , a) Personal Computer yang dilengkapi dengan tools developer: MikroC, Visual BASIC, USBTrace, program down-loader, b) Microcontroller PIC18F4550 dan perlengkapannya serta beberapa alat ukur seperti dan pendukungnya seperti logic probe, oscilloscope, AVO meter dan power supply

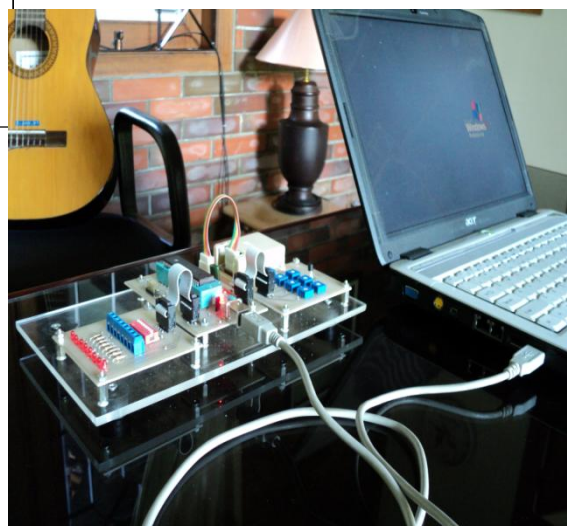
Tahapan kegiatan penelitian dilakukan seperti ditunjukkan oleh flowchart, yang disajikan pada Gambar 10.



Gambar 10. Tahapan Kegiatan Penelitian



Gambar 11a. Rangkaian Devais Input/Output USB.[7]



Gambar 11b. Foto Devais I/O USB

ANALISA DAN PEMBAHASAN.

a) Hasil Perancangan Hardware

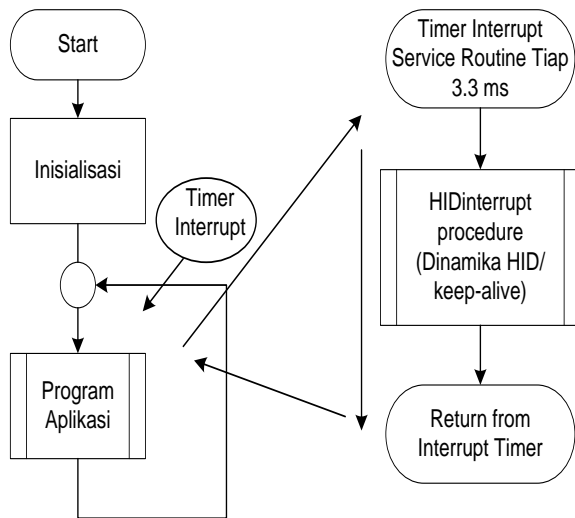
Rangkaian devais input/output USB-HID dan hasil rancangannya ditunjukkan oleh Gambar 11(a) dan 11(b). Rancangan *hardware* terdiri atas dua bagian yaitu bagian *controller* dan *peri-pheral I/O*. *Peripheral* merupakan gambaran dari bentuk aplikasi pada saat dilakukan pengujian. *controller* merupakan pengontrol antar muka USB antara devais dengan *host PC*.

b) Hasil Perancangan Software.

Software dalam sistem ini terdiri atas dua bagian, yaitu *firmware* dalam *Microcontroller* dan *software* dalam *host PC*.

Firmware dalam Microcontroller

Firmware dibangun menggunakan *compiler* MikroC pro 3.2. Ada dua bagian *firmware*, *descriptor* dan program aplikasi. Setiap program aplikasi harus menyertakan program *descriptor* pada saat melakukan kompilasi. Konsep penalaran dalam perancangan *firmware* ditunjukkan dalam Gambar 12.



Gambar 12. Diagram Alir Firmware

Ada dua *firmware* aplikasi yang dirancang di dalam penelitian ini yaitu input/output digital dan input analog

Firmware Aplikasi Input/Output Digital

Di dalam aplikasi ini dirancang sebuah *firmware* yang disertai *descriptor* sedemikian rupa sehingga terjadi transaksi dua arah, input dan output.

Host mengirimkan dua *command* ke *Microcontroller* yaitu $Command\ P = nT$ permintaan kepada *Microcontroller* untuk mengirim byte data n ke port B dan $Command\ P = ??$ permintaan kepada *Microcontroller* untuk membaca data port B dan mengirim-kembali ke *host*. *Host* kemudian mendisplaykannya ke layar. *Microcontroller* mengirim-kembali data dengan format yang sama $P = nT$.

Bentuk format transaksi yang dapat ditunjukkan dalam Gambar 13..

Gambar 13. . Form Transaksi Input/Output Digital

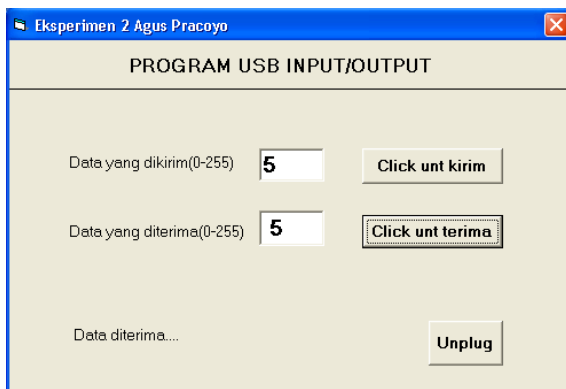
Pengujian dilakukan dengan mengirim data ke devais lewat *text box* dan *click command button* “Click untuk kirim” dan membaca data dari devais dengan *click command button* “Click untuk baca”. Data yang dikirim diamati pada LED port B, pada rangkaian hardware untyuk selanjutnya data pada port B dikirim kembali ke host dengan cara *click command button* “Click untuk terima”. Data yang dikirim balik ke *host* diamati pada *text box* terima. Dari pengujian yang dilakukan dengan melakukan 10x pengiriman hasilnya tidak ditemukan kesalahan transaksi. Hasil pengujian untuk input/output digital ditunjukkan pada Tabel 1 berikut.

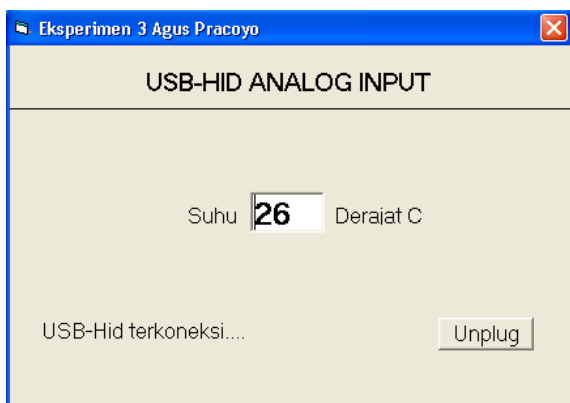
Tabel 1. Hasil Pengujian Transaksi Input/Output Digital

No	Data Krm	Data biner yang diterima oleh LED								Data Trm
		PB.7	PB.6	PB.5	PB.4	PB.3	PB.2	PB.1	PB.0	
1	255	1	1	1	1	1	1	1	1	255
2	254	1	1	1	1	1	1	1	0	254
3	253	1	1	1	1	1	1	0	1	253
4	252	1	1	1	1	1	1	0	0	252
5	15	0	0	0	0	1	1	1	1	15
6	14	0	0	0	0	1	1	1	0	14
7	10	0	0	0	0	1	0	1	0	10
8	9	0	0	0	0	1	0	0	1	9
9	8	0	0	0	0	1	0	0	0	8
10	0	0	0	0	0	0	0	0	0	9

a. Firmware Aplikasi Input Analog

Di dalam aplikasi ini *firmware* yang dirancang disertai *descriptor* yang bekerja satu arah. Input. Aplikasi ini digunakan untuk membaca data berupa tegangan analog, misalnya berasal dari sensor suhu. Bentuk *format* transaksi seperti ditunjukkan oleh Gambar 14. Dalam aplikasi ini *firmware* digunakan untuk membaca keluaran sensor LM35, berupa tegangan dan selanjutnya dikonversikan dalam bentuk digital serta memprosesnya menjadi besaran suhu untuk kemudian dikirimkan ke *host* via port USB.





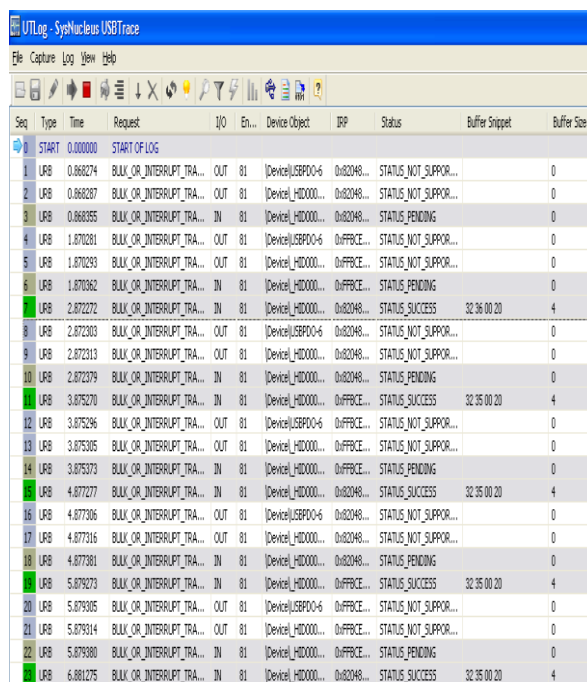
Gambar 14. . Form Transaksi Input Analog.

Pengujian pada bagian ini dilakukan dengan pembacaan sensor LM35 secara terus-menerus dengan menggunakan USB-Trace.[6] Hasilnya ditunjukkan pada Tabel 2 berikut.

Tabel 2. Hasil Pencatatan Pembacaan Suhu oleh *USBTrace*.

Offset	Hex	Ascii
00000000	32 36 00 20	26.
Offset	Hex [0003]	Ascii
00000000	32 35 00 20	25.
Offset	Hex [0003]	Ascii
00000000	32 35 00 20	25.
Offset	Hex [0003]	Ascii
00000000	32 35 00 20	25.
Offset	Hex [0003]	Ascii
00000000	32 35 00 20	25.

Hasil *capture* oleh *USBTrace* ditunjukkan dalam Gambar 15. Dalam pengujian ini dapat dilihat bahwa data ke dua dari 4 byte



Gambar 15. Hasil Capture Pembacaan

KESIMPULAN

Dari hasil pengujian dan analisis dapat disimpulkan bahwa model USB yang dirancang diuji dengan menggunakan USB Trace memiliki kecepatan transaksi kecepatan transaksi maksimal 12 Mbps (*full-speed*) dan beroperasi pada USB 2.0 memiliki kecepatan transaksi maksimal 12 Mbps (*full-speed*). Dalam implementasinya *program descriptor* harus selalu disertakan dalam setiap program aplikasi yang dibuat melalui template yang dibangkitkan oleh Easy-HID.

DAFTAR PUSTAKA

- [1] Anderson, D., 2001. *USB System Architecture (USB 2.0)*, Mind Share Inc.
- [2] Axelson,J., 2006. *USB Mass Storage Designing and Programming Devices and Embedded Hosts*, Lakeview Research LLC & Madison, WI
- [3] Chih-Yuan Huang, Tei-Wei Kuo, dan Ai-Chun Pang, 2004. *QoS Support for USB 2.0 Periodic and Sporadic Device*

- Request*. International Journal IEEE
1052-8725/04
- [4] Hyde,J., 2001. *USB Design by Example
A Practical Guide to Building I/O
Devices*, Intel University Press.
- [5] Ibrahim,Dogan, 2005. *Advance PIC
Micocontroller Projects in C (From
USB to ZIGBEE with The PIC 18F
Series)*.Newnes.
- [6] Microchip, 2004. *High-Performance
Enhanced Flash USB Microcontrollers
with nanoWatt Technology* ,
PIC18F4550,Michrochip.
- [7] Mikroelektronika, 2006, *MikroC
User's Manual*