

## RANCANG BANGUN DAN PENGUJIAN DEVAIS PERIPHERAL INPUT/OUTPUT (P I/O) BERBASIS UNIVERSAL SERIAL BUS (USB)

Agus Pracoyo<sup>1)</sup>, Tossin Alamsyah<sup>2)</sup>,

<sup>1)</sup> Staf Pengajar Jurusan Teknik Elektro Politeknik Negeri Malang

<sup>2)</sup> Staf Pengajar Jurusan Teknik Elektro Politeknik Negeri Jakarta  
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

**Key words:** Design, Testing, Peripheral Input Output (PI/O) , Universal Serial Bus (USB)

### ABSTRAK.

Teknologi Personal Computer (PC) telah menyediakan terminal USB (Universal Serial Bus) untuk keperluan akses dari dan ke devais input/output eksternal. Fenomena ini memaksa para perancang dan pengembang full-custom untuk merancang devais input/output berbasis antar muka USB. Sistem USB terdiri dari pengontrol yaitu PC sebagai host dan beberapa devais input/output (I/O) sebagai client.

Pada artikel ini dibahas hasil rancangan dan pengujian devais input/output Human Interface Decive USB dengan Microcontroller PIC18F4550. Pengujian menggunakan software Tools USBTrace yang dikembangkan oleh SysNucleus. Hasil pengujian menunjukkan bahwa kecepatan transaksi dapat mencapai 12 Mbps, merupakan kategori full-speed. Kecepatan Transaksi dicapai berdasarkan pada Compiler MikroC, dengan clock pada MCU 48 Mhz dan xtal terpasang 8 Mhz.

**Key words:** Rancang bangun, Pengujian, Peripheral Input Output (PI/O) , Universal Serial Bus (USB)

### PENDAHULUN

Saat ini hampir semua Personal Computer (PC) 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<sup>[3]</sup>.

USB (*Universal Serial Bus*) adalah standar komunikasi serial berbasis bus sebagai perangkat penghubung antara komputer dengan *peripheral I/O* eksternal .

Kelebihan utama *peripheral I/O* USB adalah *hot plugable*, artinya devais USB

dapat di *plugging* saat PC dalam kondisi *running* kemudian akan melakukan konfigurasi secara otomatis.[1]

USB adalah suatu sistem *host-centric connectivity* dimana *host* mendikte pemakaian bus USB. Masing-masing devais *input/output* pada bus ditandai dengan suatu alamat USB yang unik, dan tidak ada sinyal devais *slave* yang aktif pada bus sampai *host* memintanya.

Pada artikel ini dibahas mengenai rancangan dan pembuatan modul devais input/output USB yang menggunakan *Microcontroller* PIC 18F4550 sebagai pengontrolnya. Untuk selanjutnya ditambahkan pula modul *peripheral I/O* digital dan analog sebagai pendukung saat

dilakukan pengujian dengan **tools software USBTrace**

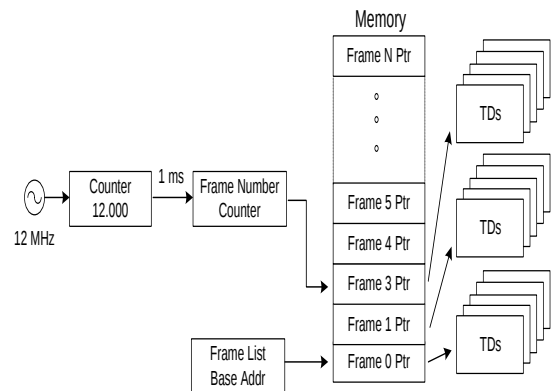
Pemilihan *Microcontroller* Microchip PIC18F4550 di dasarkan pada kekhususan fiturnya dalam menangani pheripheral USB. *Microcontroller* ini memiliki *USB Serial Interface Engine (SIE)* <sup>[6]</sup> yang kompatibel dengan transaksi *low-speed* dan *full-speed* yang memungkinkan dapat berkomunikasi dengan *USB host* versi 2.0, dan secara langsung dapat di-interface-kan ke bus USB dengan memanfaatkan *transveiver* internal.[6]

### 1) Kakateristik transaksi data USB.

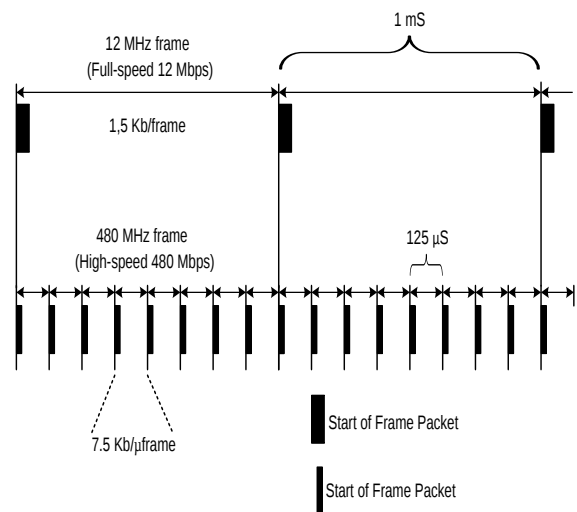
Karakteristik USB versi1.1 atau USB1.1 mendukung kecepatan transaksi pada 1.5Mbps (*low speed*) dan 12Mbps(*full-speed*). Transaksi *low-speed* dapat digunakan untuk mendukung transaksi data pada devais low dan full speed. Artinya devais *full-speed* beroperasi pada transaksi *low-speed*. Tetapi untuk transaksi data full speed hanya dapat dilakukan pada devais *full-speed*, karena devais *low-speed* tidak dapat melakukan transaksi *full-speed*.

### 2) Frame dan Lebar Jalur (Bandwidth) USB.

*Frame* adalah daftar transaksi atau daftar *descriptor* <sup>[1]</sup>. Pada transaksi *full-speed*, satu(1) frame adalah interval selama 1milisecond yang mana *host* mengambil dan melaksanakan (*fetch and execute*) satu rangkaian *descriptor*. Kemudian *Controller* mengambil masing-masing daftar *frame* di dalam interval 1 millisecond, seperti dapat diperlihatkan pada Gambar 1a yang menunjukkan pembangkit frame 1ms dan Gambar 1b menunjukkan lebar dari frame.



Gambar 1a. Pembangkit Frame 1 ms.[Don Anderson,2001]



Gambar 1b . Frame 1ms dan 125 μs  
Sumber: Don Anderson, 2001

### 3) Transaksi dan class USB.

Transaksi data dalam bentuk pengambilan (*fetching*) dan eksekusi (*executing*) pada *link* struktur data yang disebut dengan *transfer descriptor* dari memori<sup>[1]</sup>. Pada USB versi 1.x transaksi dilakukan dengan kecepatan 1,5 Mbps (*low-speed* ) dan 12 Mbps (*full-speed*). Kemudian untuk USB versi 2.0 disamping dapat bertransaksi pada low speed dan full speed, juga dapat bertransaksi secara *high-speed* sekitar 480 Mbps.

Berdasarkan dari dari kecepatan transaksi maka USB dapat digolongkan kepada beberapa kelas (*class devices*) disesuaikan dengan implementasinya, yang

dibedakan menjadi Sembilan (9) kelas seperti ditunjukkan dalam Tabel 1<sup>[1]</sup>.

Tabel 1. Kelas USB

| Device class | Deskripsi                        | Aplikasi    |
|--------------|----------------------------------|-------------|
| 0x00         | Reserved                         | -           |
| 0x01         | USB audio device                 | Sound card  |
| 0x02         | USB communication dvc            | Modem       |
| 0x03         | USB Human Interface Device (HID) | Kbd mouse   |
| 0x07         | USB printer device               | Printer     |
| 0x08         | USB mass storage dvc             | Flash drive |
| 0x09         | USB hub device                   | Hubs        |
| 0x0B         | USB smart card reader            | Crd reader  |
| 0x0E         | USB video device                 | Webcam      |
| 0xe0         | USB wireless device              | Bluetooth   |

Sumber: Dogan Ibrahim, 2008

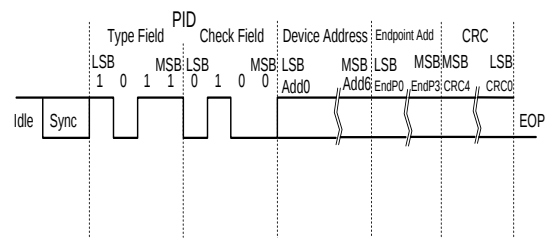
Dalam artikel ini dibahas kelas Human interface device (HID dipilih dalam disain devais input/ output USB, dengan aplikasi yang direncanakan tidak membutuhkan transaksi data dalam jumlah besar.

Devais USB kelas *mass storage* memiliki *firmware* yang kompleks. Jika *firmware* devais membutuhkan untuk membaca atau menulis file dan direktori ke dalam media penyimpanan, *firmware* harus *support* terhadap *file system*. Untuk beberapa aplikasi berbeda dengan kelas USB yang lainnya. Protokol khusus dibutuhkan dalam menangani kelas ini<sup>[2]</sup>.

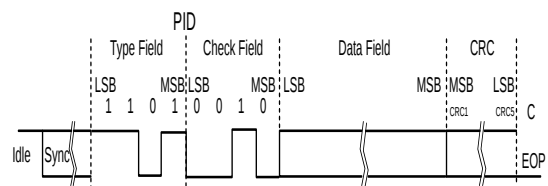
#### 4) Paket Data

Data dipancarkan pada *bus* USB berupa paket-paket. Suatu paket dimulai dengan pola sinkronisasi untuk mensinkronkan antara *clock* penerima dengan data. *Byte-by-byte* data dari paket mengikutinya, dan berakhir dengan satu signal *End Of Packet (EOP)*. *Byte Packet identifier (PID)* mengikuti *sync field* dari tiap paket USB. Panjang PID 8 bit yang terdiri atas *type field* 4 bit dan *check field* 4 bit seperti ditunjukkan dalam Gambar 2a, 2b dan 2c,

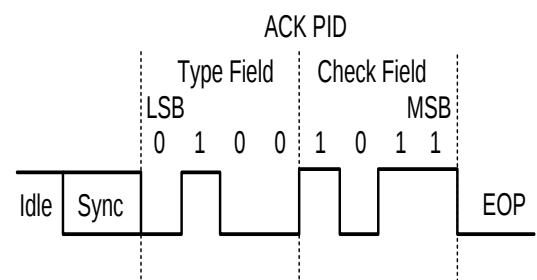
Ada empat bentuk paket, yang berbasis pada PID yaitu: *Token Packets*, *Data Packets*, *Handshake Packets*, dan *Special Packets*.



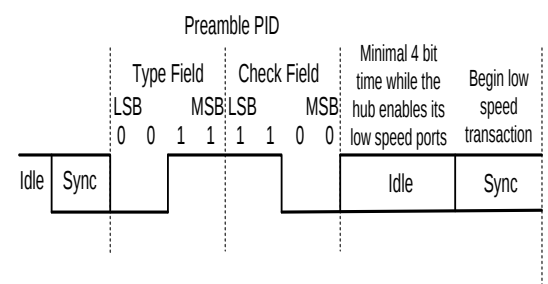
Gambar 2a.. Paket Token [1]



Gambar 2b. Paket Data [1]



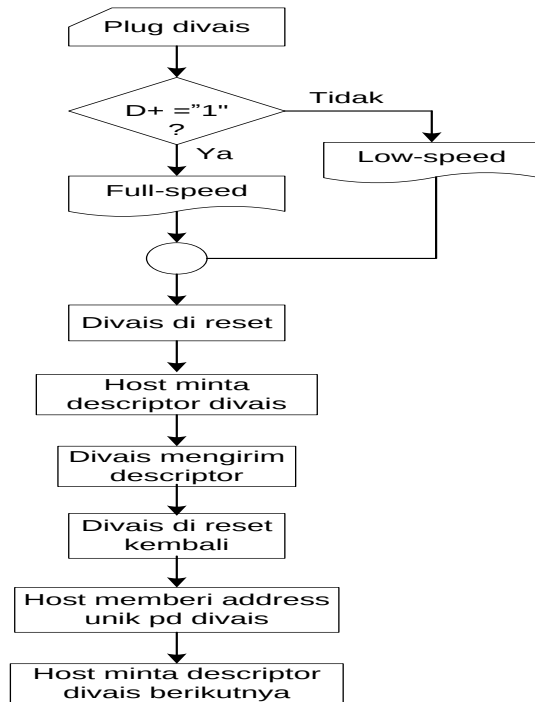
Gambar 2c. Paket Jabat Tangan [1]



Gambar 2c.. Paket Khusus[1]

#### 5) Enumerasi dan Deskriptor.

Enumerasi adalah proses pengenalan devais input/output ketika dipasang (plugged) pada host<sup>[1]</sup>. Langkah-langkah enumerasi ditunjukkan seperti Gambar 3.



**Gambar 3.** Enumerasi

Semua devais USB mempunyai hirarki descriptor yang menguraikan berbagai fitur dari devais antara lain: ID pabrik pembuat, versi devais, versi USB yang mendukung, devais apa, power yang dibutuhkan, nomor dan tipe dari *endpoint*, dan sebagainya. Diskriptor USB yang paling umum adalah:

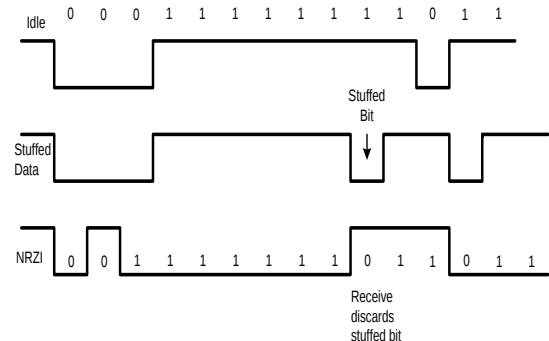
- Device descriptors
- Configuration descriptors
- HID descriptors
- Endpoint descriptors

Penambahan setiap devais I/O USB-HID dilakukan dengan penambahan address pada *Endpoint descriptor*. Setiap devais I/O USB-HID mempunyai address unik hingga 127 devais.

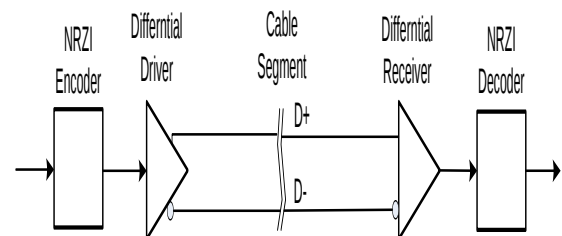
## 6) Sinyal USB

Informasi yang melintasi kabel USB menggunakan *encoding* NRZI (*Non-return to Zero Inverted*) dan signaling diferensial seperti ditunjukkan dalam Gambar 4. Bit 0 diisiikan 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 Stuffed Bit.



**Gambar 4a.** Data NRZI dan Stuffed Bit.[5]



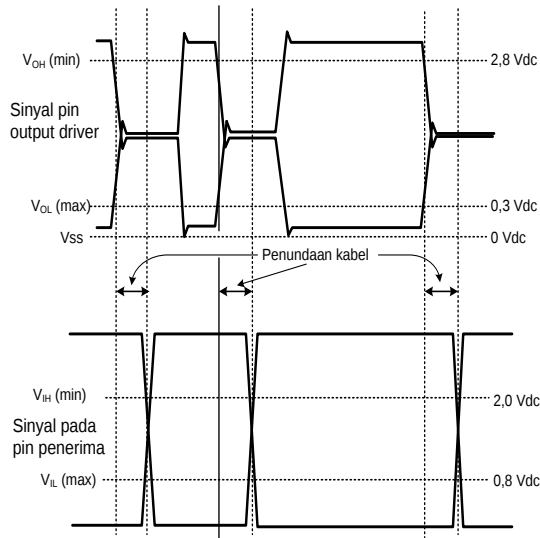
**Gambar 4b.** NRZI Encoding dan Diferensial Signaling.[5]

Siganiling diferensial dan melintasi kabel USB seperti ditunjukkan dalam Gambar 4b. Signal Differential “1” pada saat D+ lebih besar daripada D- dan Differential “0” adalah pada saat D- lebih besar daripada D+ seperti ditunjukkan dalam Gambar 5. Tegangan actual ditetapkan untuk driver differential adalah :

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

Level tegangan yang diterima harus terdeteksi sbb:

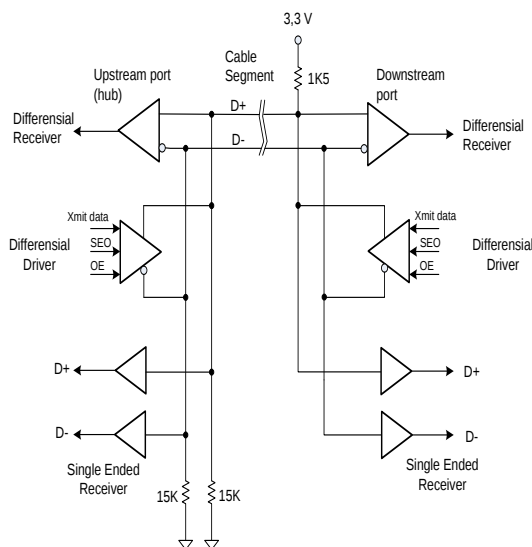
- Differential “1” =  $(D+) - (D-) > 200m$  dan  $D+ > V_{IH(min)}$
- Differential “0” =  $(D-) - (D+) > 200m$  dan  $D- > V_{IH(min)}$



Gambar 5 . Sinyal Diferensial.[5]

## 7) Deteksi Kecepatan

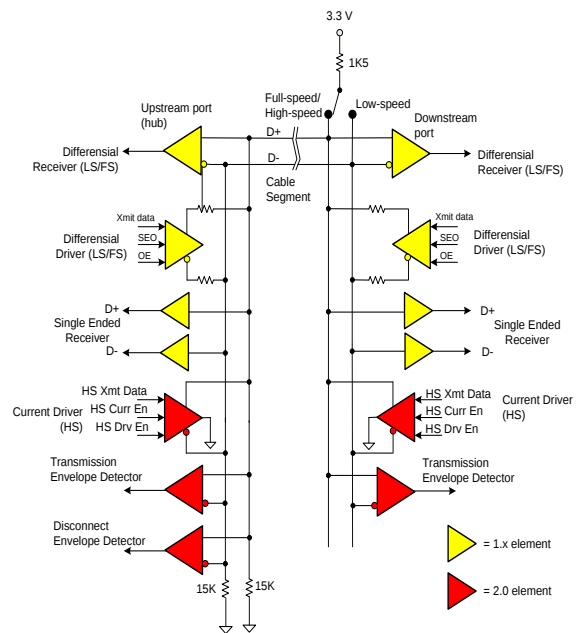
Pada bus *full-speed* , resistor dihubungkan dari D+ ke 3,3V, dan pada bus *low-speed*, resistor dihubungkan dari D- ke 3,3V. Ketika tidak ada devais yang disambungkan, *host* akan melihat kedua jalur data dalam keadaan rendah (*single ended reveiver*) seperti ditunjukkan dalam Gambar 6. Menghubungkan devais ke bus akan menarik salah satu D+ atau D- ke logika tinggi, dan *host* akan mengetahui bahwa devais dipasang (*plugged*) ke dalam bus.



Gambar 6. Interface USB 1.1.[4]

Kecepatan devais ditentukan dengan observasi jalur data mana yang dihubungkan ke resistor *pull-up* seperti ditunjukkan dalam Gambar 6 di atas.

Jika devais *full-speed* dan *low-speed* di *plug* pada USB 2.0 maka bentuk interface ditunjukkan dalam Gambar 7.

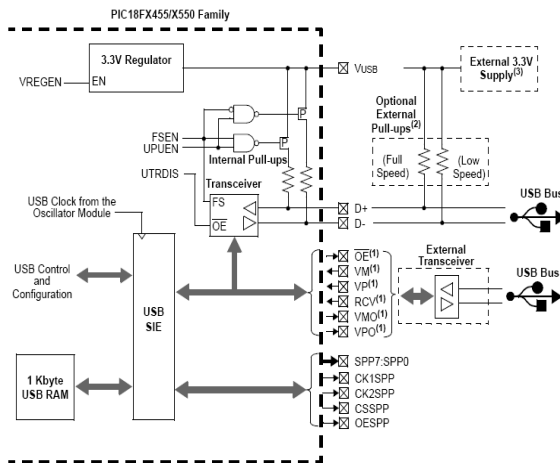


Gambar 7. Interface USB 2.0.[4]

Perbedaan interface USB 1.x dan USB 2.0 adalah ditambahkannya elemen *high-speed*. Sehingga USB 2.0 tetap dapat melayani transaksi *low-speed* dan *full-speed*. Pada saat devais beroperasi pada *high-speed* resistor *pull-up* tetap terpasang pada D+ seperti pada saat beroperasi pada *full-speed*.

## 8) Mikrokontroler PIC18F4550

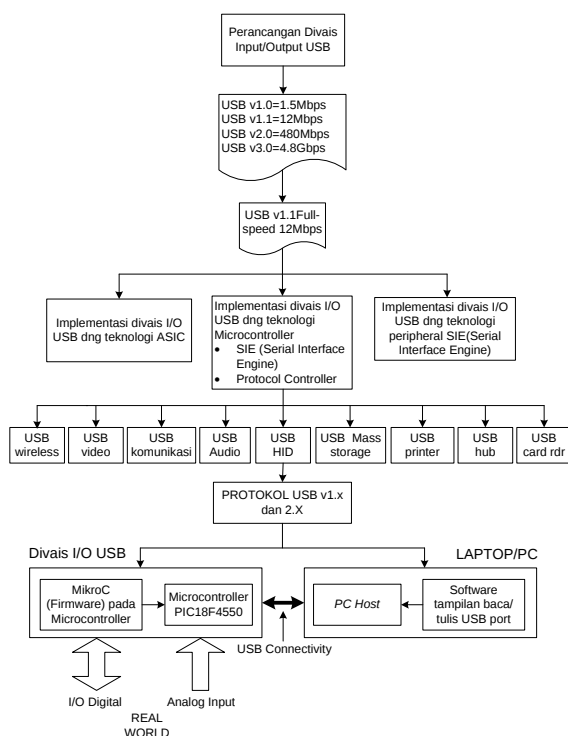
Rangkaian utama pada devais input/output didesain menggunakan mikrokontroler PIC18F4550 produksi Microchip. Mikrokontroler ini dirancang khusus untuk menangani teknik antar muka USB. Diagram blok PIC18F4550 ditunjukkan dalam Gambar 8.



Gambar 8. Diagram Blok PIC18F4550.[3]

## METODOLOGI

Pola alur kegiatan kerangka penelitian yang menghasilkan model desain USB HID dan pengujiannya dapat digambarkan sesuai flowchart berikut.



Gambar 9. Flowchart alur Kerangka Penelitian

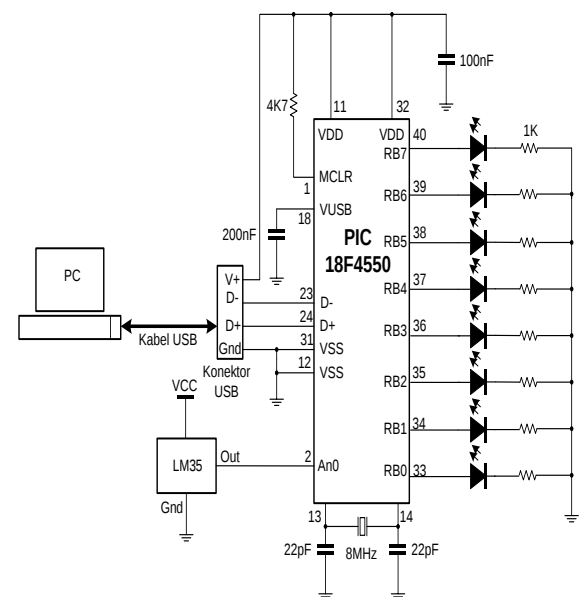
Sesuai dengan tujuan penelitian, maka dirancang dan dibangun devais input/output USB-HID dengan kecepatan transaksi 12 Mbps (*full-speed*). Implementasi devais ini menggunakan teknologi *Microcontroller*

Microchip 18F4550. Sebagai peripheral ditambahkan input/output digital dan input analog berupa sensor suhu LM35. *Firmware* dibangun menggunakan *compiler* MikroC pro v3.2, dan *software* pada *host* dibangun menggunakan Visual BASIC. Unjuk kerja devais USB diuji menggunakan perangkat lunak USBTrace.

## HASIL DAN PEMBAHASAN

### 1) Perancangan *Hardware*

Rangkaian devais input/output USB-HID ditunjukkan dalam Gambar 10. *Hardware* terdiri atas dua blok, *controller* dan *peri-pheral I/O*. *Peripheral* merupakan contoh bentuk aplikasi pada saat pengujian. *controller* merupakan pengontrol antar muka USB antara devais dengan *host PC*.



Gambar 10. Rangkaian Devais Input/Output  
USB



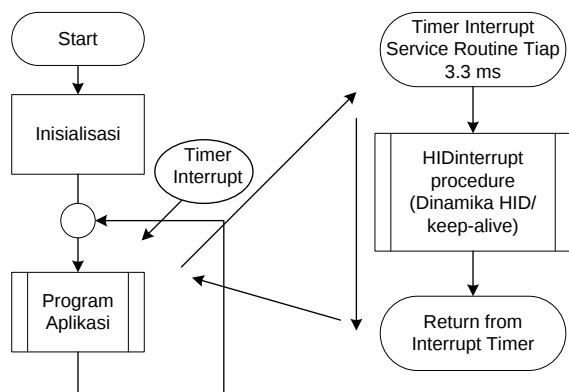
Gambar 11. Foto Devais I/O USB

## 2) Perancangan Software

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

### (a) *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 11.



Gambar 11. Diagram Alir *Firmware*

Ada dua *firmware* aplikasi yang dirancang di dalam penelitian ini yaitu:

- Aplikasi input/output digital
- Aplikasi input analog

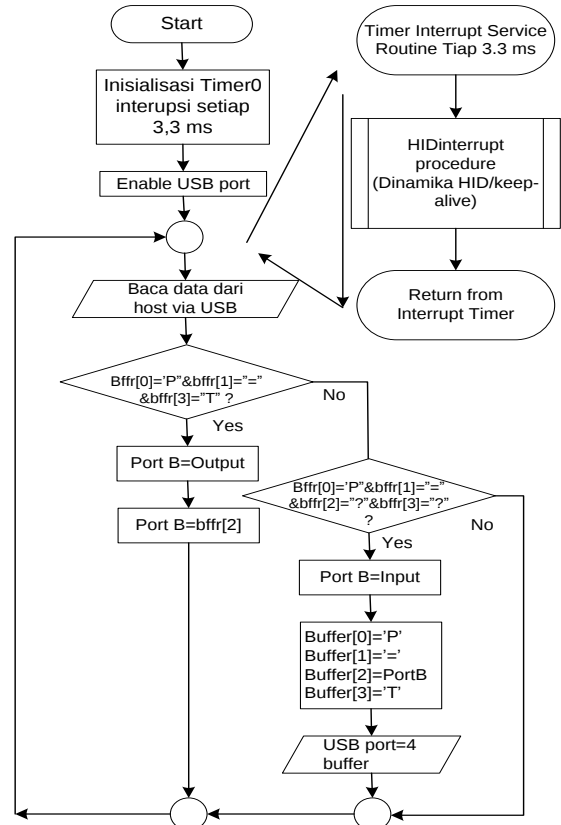
### (b) *Firmware* Aplikasi *Input/Output* Digital

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*:

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

Sehingga *flowchart* *firmware* ditunjukkan dalam Gambar 12.



Gambar 12. Diagram Alir *Firmware* Aplikasi I/O Digital

### (a) Pengujian dan Simulasi.

Pengujian hasil perancangan dan implementasi devais *Input/Out* USB-HID bertujuan untuk mengetahui unjuk kerja dan performa hasil perancangan.

- **Pengujian Descriptor**

Pengujian *firmware descriptor* merupakan pengujian tahap awal. Jika devais USB telah diisi *descriptor* dan di *plug* ke *port* USB *host*, maka oleh Windows dikenali dengan tampilan berturut-turut seperti ditunjukkan dalam Gambar 13 yang menunjukkan keberhasilan pemrograman *descriptor*, karena devais telah berhasil dikenali oleh Windows dan di konfigurasi.



Gambar 13. Pengenalan Widows Pada Devais

- **Pengujian Aplikasi *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”.

- **Pengujian Fungsi Devais**

Data yang dikirim diamati pada LED port B. Data pada *port* B dikirim kembali ke *host* dengan cara *click command button* “Click untuk terima”. *Form* transaksi ditunjukkan dalam Gambar 20. Data yang dikirim balik ke *host* diamati pada *text box* terima. Pengujian ini dilakukan 10X seperti ditunjukkan dalam Tabel 2.

Tabel 2. Hasil Pengujian Transaksi *Input/Output* Digital

| No | Data | Data biner yang diterima oleh LED |      |      |      |      |      |      |      | Data |
|----|------|-----------------------------------|------|------|------|------|------|------|------|------|
|    | Krm  | PB.7                              | PB.6 | PB.5 | PB.4 | PB.3 | PB.2 | PB.1 | PB.0 | Trm  |
| 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    |

Dari hasil pengujian dalam Tabel 2 sebanyak 10x tidak ditemukan kesalahan transaksi. Keberhasilan mutlak 100%.

- **Pengujian Dengan Tools *USBTrace***

*Software* ini merupakan analiser protokol USB yang dikembangkan oleh *SysNucleus* yang dijalankan pada *host* PC. *Software* memonitor *port* USB pada *host* PC dan mendisplaykan semua transaksi pada *bus* USB.

Hasil pengujian transaksi menggunakan *USBTrace* ditunjukkan secara berturut-turut dalam Gambar14a, 14b dan 14c.

| Seq | Type  | Time      | Request                  | I/O | En... | Device Object     | IRP        | Status               | Buffer Snippet | Buffer Size |
|-----|-------|-----------|--------------------------|-----|-------|-------------------|------------|----------------------|----------------|-------------|
| 0   | START | 0.000000  | START OF LOG             |     |       |                   |            |                      |                |             |
| 1   | URB   | 24.389058 | BULK_OR_INTERRUPT_TRA... | OUT | 1     | \Device\USBPDO-6  | 0xFF6CF... | STATUS_SUCCESS       | 50 3D 05 54    | 4           |
| 2   | URB   | 24.389070 | BULK_OR_INTERRUPT_TRA... | OUT | 1     | \Device\USBPDO-6  | 0xFF6CF... | STATUS_SUCCESS       | 50 3D 05 54    | 4           |
| 3   | URB   | 24.389145 | BULK_OR_INTERRUPT_TRA... | IN  | 1     | \Device\HID000... | 0xFF6CF... | STATUS_PENDING       |                | 0           |
| 4   | URB   | 24.390369 | BULK_OR_INTERRUPT_TRA... | IN  | 1     | \Device\HID000... | 0xFF6CF... | STATUS_SUCCESS       |                | 0           |
| 5   | URB   | 31.131975 | BULK_OR_INTERRUPT_TRA... | OUT | 1     | \Device\USBPDO-6  | 0xFF6CF... | STATUS_SUCCESS       | 50 3D 3F 3F    | 4           |
| 6   | URB   | 31.131991 | BULK_OR_INTERRUPT_TRA... | OUT | 1     | \Device\HID000... | 0xFF6CF... | STATUS_SUCCESS       | 50 3D 3F 3F    | 4           |
| 7   | URB   | 31.132065 | BULK_OR_INTERRUPT_TRA... | IN  | 1     | \Device\HID000... | 0xFF6CF... | STATUS_PENDING       |                | 0           |
| 8   | URB   | 31.133382 | BULK_OR_INTERRUPT_TRA... | OUT | 81    | \Device\USBPDO-6  | 0x82133... | STATUS_NOT_SUPPOR... |                | 0           |
| 9   | URB   | 31.133392 | BULK_OR_INTERRUPT_TRA... | OUT | 81    | \Device\HID000... | 0x82133... | STATUS_NOT_SUPPOR... |                | 0           |
| 10  | URB   | 31.133466 | BULK_OR_INTERRUPT_TRA... | IN  | 81    | \Device\HID000... | 0x82133... | STATUS_PENDING       |                | 0           |
| 11  | URB   | 31.133490 | BULK_OR_INTERRUPT_TRA... | IN  | 1     | \Device\HID000... | 0xFF6CF... | STATUS_SUCCESS       |                | 0           |

**URB\_FUNCTION\_BULK\_OR\_INTERRUPT\_TRANSFER**  
 Length : 0x48  
 USBD Status : USBD\_STATUS\_SUCCESS (0x0)  
 EndpointAddress : 0x1  
 PipeHandle : 0x861CD9A4  
 TransferFlags : 0x2 ( USBD\_TRANSFER\_DIRECTION\_OUT USBD\_SHORT\_TRANSFER\_OK )  
 TransferBufferLength : 0x4  
 TransferBuffer : 0xED905248  
 TransferBufferMDL : 0x0  
 UrbLink : 0x0  
 Data : 50 3D 05 54

Gambar 14a.. Hasil *Capture* Transaksi 1

| Seq | Type  | Time      | Request                  | I/O | En... | Device Object     | IRP        | Status               | Buffer Snippet | Buffer Size |
|-----|-------|-----------|--------------------------|-----|-------|-------------------|------------|----------------------|----------------|-------------|
| 0   | START | 0.000000  | START OF LOG             |     |       |                   |            |                      |                |             |
| 1   | URB   | 24.389058 | BULK_OR_INTERRUPT_TRA... | OUT | 1     | \Device\USBPDO-6  | 0xFF6CF... | STATUS_SUCCESS       | 50 3D 05 54    | 4           |
| 2   | URB   | 24.389070 | BULK_OR_INTERRUPT_TRA... | OUT | 1     | \Device\HID000... | 0xFF6CF... | STATUS_SUCCESS       | 50 3D 05 54    | 4           |
| 3   | URB   | 24.389145 | BULK_OR_INTERRUPT_TRA... | IN  | 1     | \Device\HID000... | 0xFF6CF... | STATUS_PENDING       |                | 0           |
| 4   | URB   | 24.390369 | BULK_OR_INTERRUPT_TRA... | IN  | 1     | \Device\HID000... | 0xFF6CF... | STATUS_SUCCESS       |                | 0           |
| 5   | URB   | 31.131975 | BULK_OR_INTERRUPT_TRA... | OUT | 1     | \Device\USBPDO-6  | 0xFF6CF... | STATUS_SUCCESS       | 50 3D 3F 3F    | 4           |
| 6   | URB   | 31.131991 | BULK_OR_INTERRUPT_TRA... | OUT | 1     | \Device\HID000... | 0xFF6CF... | STATUS_SUCCESS       |                | 0           |
| 7   | URB   | 31.132065 | BULK_OR_INTERRUPT_TRA... | IN  | 1     | \Device\HID000... | 0xFF6CF... | STATUS_PENDING       |                | 0           |
| 8   | URB   | 31.133382 | BULK_OR_INTERRUPT_TRA... | OUT | 81    | \Device\USBPDO-6  | 0x82133... | STATUS_NOT_SUPPOR... |                | 0           |
| 9   | URB   | 31.133392 | BULK_OR_INTERRUPT_TRA... | OUT | 81    | \Device\HID000... | 0x82133... | STATUS_NOT_SUPPOR... |                | 0           |
| 10  | URB   | 31.133466 | BULK_OR_INTERRUPT_TRA... | IN  | 81    | \Device\HID000... | 0x82133... | STATUS_PENDING       |                | 0           |
| 11  | URB   | 31.133490 | BULK_OR_INTERRUPT_TRA... | IN  | 1     | \Device\HID000... | 0xFF6CF... | STATUS_SUCCESS       |                | 0           |

**URB\_FUNCTION\_BULK\_OR\_INTERRUPT\_TRANSFER**  
 Length : 0x48  
 USBD Status : USBD\_STATUS\_SUCCESS (0x0)  
 EndpointAddress : 0x1  
 PipeHandle : 0x861CD9A4  
 TransferFlags : 0x2 ( USBD\_TRANSFER\_DIRECTION\_OUT USBD\_SHORT\_TRANSFER\_OK )  
 TransferBufferLength : 0x4  
 TransferBuffer : 0xF7EB9260  
 TransferBufferMDL : 0x0  
 UrbLink : 0x0  
 Data : 50 3D 3F 3F

Gambar 14b. Hasil *Capture* Transaksi 2

| Seq | Type  | Time      | Request                  | I/O | En... | Device Object     | IRP        | Status               | Buffer Snippet | Buffer Size |
|-----|-------|-----------|--------------------------|-----|-------|-------------------|------------|----------------------|----------------|-------------|
| 0   | START | 0.000000  | START OF LOG             |     |       |                   |            |                      |                |             |
| 1   | URB   | 24.389058 | BULK_OR_INTERRUPT_TRA... | OUT | 1     | \Device\USBPDO-6  | 0xFF6CF... | STATUS_SUCCESS       | 50 3D 05 54    | 4           |
| 2   | URB   | 24.389070 | BULK_OR_INTERRUPT_TRA... | OUT | 1     | \Device\HID000... | 0xFF6CF... | STATUS_SUCCESS       | 50 3D 05 54    | 4           |
| 3   | URB   | 24.389145 | BULK_OR_INTERRUPT_TRA... | IN  | 1     | \Device\HID000... | 0xFF6CF... | STATUS_PENDING       |                | 0           |
| 4   | URB   | 24.390369 | BULK_OR_INTERRUPT_TRA... | IN  | 1     | \Device\HID000... | 0xFF6CF... | STATUS_SUCCESS       |                | 0           |
| 5   | URB   | 31.131975 | BULK_OR_INTERRUPT_TRA... | OUT | 1     | \Device\USBPDO-6  | 0xFF6CF... | STATUS_SUCCESS       | 50 3D 3F 3F    | 4           |
| 6   | URB   | 31.131991 | BULK_OR_INTERRUPT_TRA... | OUT | 1     | \Device\HID000... | 0xFF6CF... | STATUS_SUCCESS       | 50 3D 3F 3F    | 4           |
| 7   | URB   | 31.132065 | BULK_OR_INTERRUPT_TRA... | IN  | 1     | \Device\HID000... | 0xFF6CF... | STATUS_PENDING       |                | 0           |
| 8   | URB   | 31.133382 | BULK_OR_INTERRUPT_TRA... | OUT | 81    | \Device\USBPDO-6  | 0x82133... | STATUS_NOT_SUPPOR... |                | 0           |
| 9   | URB   | 31.133392 | BULK_OR_INTERRUPT_TRA... | OUT | 81    | \Device\HID000... | 0x82133... | STATUS_NOT_SUPPOR... |                | 0           |
| 10  | URB   | 31.133466 | BULK_OR_INTERRUPT_TRA... | IN  | 81    | \Device\HID000... | 0x82133... | STATUS_PENDING       |                | 0           |
| 11  | URB   | 31.133490 | BULK_OR_INTERRUPT_TRA... | IN  | 1     | \Device\HID000... | 0xFF6CF... | STATUS_SUCCESS       |                | 0           |

Gambar 14c. Hasil *Capture* Seluruh Transaksi

Dari gambar pengujian diatas (14a,b dan c) hasil *capture* dua *command* transaksi “Kirim” dan “Terima”. Kedua *command* di atas juga dicatat oleh *USBTrace* dalam bentuk yang lain misalnya tabel.

#### • . Pengujian *Descriptor* dan *Speed*

Di dalam proses Enumerasi disebut-kan bahwa pada saat devais USB-HID di *plug* pada *port* USB *host*, maka pertama kali *host* melakukan *reset* pada devais USB-

HID. Kemudian devais USB-HID secara berturut mengirimkan transaksi Deskriptornya ke *host* PC. Seluruh peristiwa ini juga di *capture* dan dicatat oleh *USBTrace*. Sebagian hasil transaksi. Kecepatan transaksi menu-njukkan performa dari devais USB-HID yang dirancang. Dalam Gambar 15

| Parameter                 | Value                                  |
|---------------------------|----------------------------------------|
| Hardware ID               | USB\VID_1234&PID_0001&Rev_0001         |
| Setup Class               | HIDClass                               |
| Class GUID                | {745A17A0-74D3-11D0-B6FE-00A0C90F57DA} |
| PDO Name                  | \Device\USBPDO-6                       |
| Service Name              | HidUSB                                 |
| Parameter                 | Value                                  |
| Connection Information    |                                        |
| ConnectionIndex           | 0x2                                    |
| CurrentConfigurationValue | 0x1                                    |
| LowSpeed                  | TRUE                                   |
| DevicesHub                | TRUE                                   |
| DeviceAddress             | 0x1                                    |

Gambar 15. Diagram Hasil Pengujian USB

transaksi *low-speed* dan transaksi *full-speed* transaksi adalah *low-speed* 1,5 Mbps. Sedangkan di dalam Tabel 7 ditunjukkan bahwa “*Low-speed*” = *FALSE*” yang berarti pencapaian kecepatan transaksi adalah full-speed 12 Mbps

## KESIMPULAN

Dari hasil pengujian dan analisis dapat disimpulkan hal-hal sebagai berikut:

Menurut hasil pengukuran yang dilakukan menggunakan USBTrace, devais Input/Output USB-HID hasil rancangan dapat beroperasi pada USB 2.0 dengan kecepatan transaksi maksimal 12 Mbps (*full-speed*). Kemudian untuk mendapatkan kecepatan transaksi 1,5 Mbps (*low-speed*), dibutuhkan sumber clock eksternal sebesar 6 Mhz, hal ini dapat dibangkitkan dengan memasang xtal 24 Mhz.

Dalam Implementasinya maka setiap Program Descriptor harus disertakan dalam setiap program

## DAFTAR ACUAN..

- [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, Microchip.
- [7.] Mikroelektronika, 2006, *MikroC User's Manual*