

PC-Link

PC-LINK *Application Note* AN122 – USB to I²C Peripheral Bridge

Oleh: Tim IE

Aplikasi berikut adalah salah satu contoh penggunaan mode Bit Bang yang didukung oleh PC-Link USBer. Dalam aplikasi ini PC-Link USBer akan berfungsi sebagai *USB to I²C converter* untuk menghubungkan DT-I/O I2C Peripheral ke port USB komputer. Fungsi RTC, EEPROM, dan ADDA dari DT-I/O I2C Peripheral dapat diakses dari komputer melalui PC-Link USBer. Aplikasi ini menggunakan unit / komponen siap pakai TaUSBdirect yang akan memudahkan kita dalam memakai prosedur / fungsi DLL dari *driver* D2XX FT232. Program aplikasi ini dikembangkan menggunakan Borland® Delphi™ 7. Aplikasi ini membutuhkan modul dan komponen berikut:

- 1 PC-Link USBer,
- 1 DT-I/O I2C Peripheral,
- 1 PCF8591.

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram AN122

Hubungan antara modul-modul tersebut adalah sebagai berikut:

PC-Link USBer (J2)	DT-I/O I2C Peripheral
GND (pin 1)	GND (J1)
VCC (pin 2)	+5VDC (J1)
TXD (pin 3)	SDA (J4)
RXD (pin 4)	SCL (J4)

* Pin ini tidak mutlak dan dapat diganti pin lain tetapi harus mengubah program

Tabel 1
Hubungan PC-Link USBer dengan DT-I/O I2C Peripheral

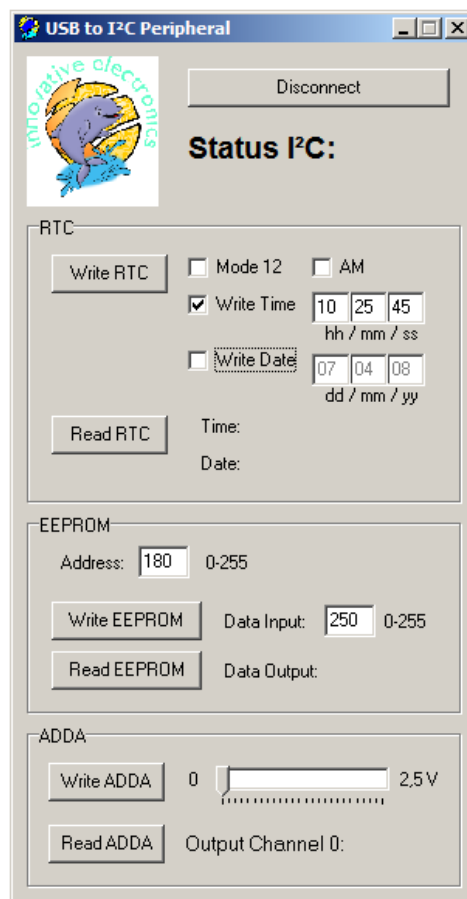
Driver D2XX perlu diinstalasi terlebih dahulu sebelum mencoba program aplikasi **Peripheral.exe**, *driver* ini terdapat pada CD produk PC-Link USBer. Jika pada komputer sudah terlanjur terinstalasi *driver* VCP maka perlu dilakukan proses *uninstall* terlebih dahulu untuk *driver* VCP tersebut.

Proyek **Peripheral.dpr** dikembangkan menggunakan Borland® Delphi™ 7.0 Enterprise® dan menggunakan komponen **TaUSBDirect**. Kode sumber untuk komponen TaUSBDirect terdapat pada AN122.ZIP, cara instalasi komponen ini ke IDE Borland® Delphi™ adalah sebagai berikut:

1. *Extract file* kode sumber TaUSBDirect dari **TaUSBDirect.zip**.
2. Kopikan *file* kode sumber TaUSBDirect ke dalam sub direktori di bawah direktori instalasi Borland® Delphi™. Misalnya: "C:\Program Files\Borland\Delphi\USBer".
3. Tambahkan *path* direktori tersebut ke dalam daftar *Library Path* Borland® Delphi™, yaitu dengan cara:
 - a) Pilih menu "Tools – Environment Options".
 - b) Pada jendela baru "Environment Options" pilih *tab* "Library", lalu pada "Library Path" pilih "...". (*browse*).
 - c) Pada jendela baru "Directories" kolom *edit text* ketikkan *path* direktori untuk kode sumber TaUSBDirect. Misalnya: "\$(DELPHI)\USBer".
 - d) Klik tombol "Add", lalu klik tombol "OK".
 - e) Pada jendela "Environment Options" klik tombol "OK".
4. Pilih menu "File – Open", lalu pilih file **USBPackage.dpk** dan klik "Open".

5. Pada jendela baru "Package - USBPackage.dpk" klik tombol "Compile" lalu klik tombol "Install".
6. Pada *toolbar* "Component Palette" dalam IDE Borland® Delphi™ akan muncul *tab* baru bernama "Astron Logic" yang di dalamnya terdapat komponen "aUSBDirect".
7. Tutup proyek USBPackage.dpk dan simpan.

Lalu hubungkan kedua modul sesuai dengan Tabel 1, pasanglah jumper J3 DT-I/O I2C Peripheral pada pin 1-2 dan pada pin 3-4 untuk memberi *pull up* pada jalur komunikasi SDA dan SCL. Pasanglah juga IC PCF8591 di *socket* U1 DT-I/O I2C Peripheral agar dapat mencoba bagian ADDA dalam program aplikasi ini. Hubungkan PC-Link USBer ke port USB komputer menggunakan kabel USB yang tersedia dalam paket penjualan PC-Link USBer dan berilah catu daya +5VDC pada rangkaian. Setelah semua rangkaian dan sumber tegangan terhubung dengan benar, program Periperol.exe dapat dijalankan. Adapun tampilan program Periperol.exe adalah sebagai berikut:



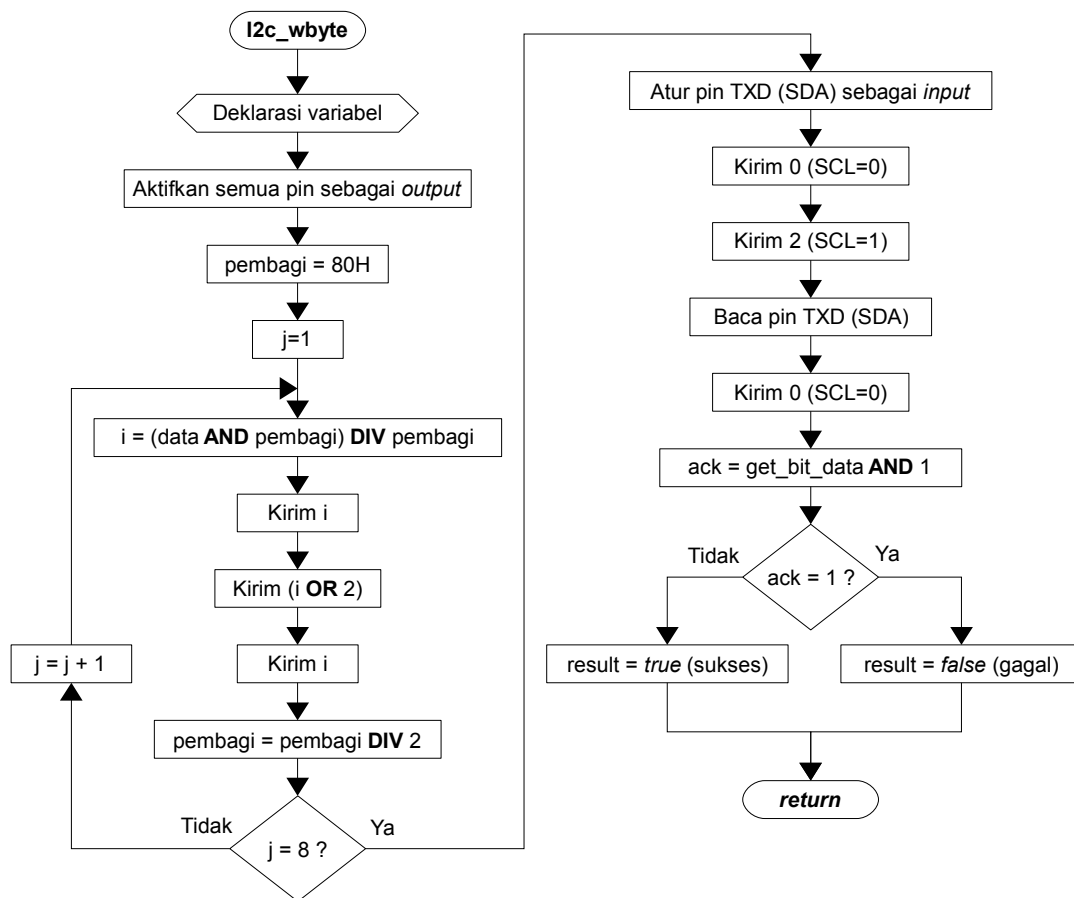
Gambar 2
Tampilan Program Periperol.exe

Fungsi dari masing-masing bagian pada jendela program Periperol.exe dijelaskan sebagai berikut:

- Tombol "Connect" / "Disconnect" berfungsi untuk menghubungkan atau memutus komunikasi antara PC-Link USBer dengan komputer.
- Tombol "Write RTC" berfungsi untuk menuliskan data ke RTC DT-I/O I2C Peripheral. Data yang dituliskan adalah data yang diketikkan pada *edit text* di samping *check box* "Write Time" dan/atau "Write Date".
- *Check box* "Write Time" dan/atau *check box* "Write Date" harus dicentang terlebih dahulu sebelum menekan tombol "Write RTC". Centang *check box* "Write Time" untuk menuliskan data waktu dan/atau centang *check box* "Write Date" untuk menuliskan data tanggal.
- *Check box* "Mode 12" digunakan untuk memilih mode waktu yang akan digunakan RTC. Jika *check box* ini dicentang maka mode waktu yang digunakan adalah mode 12 jam, sedangkan jika tidak dicentang maka mode waktu yang digunakan adalah mode 24 jam.
- *Check box* "AM" digunakan untuk memilih waktu AM atau PM pada mode waktu 12 jam (*check box* "Mode 12" harus dicentang).
- Tombol "Read RTC" berfungsi untuk membaca data dari RTC DT-I/O I2C Peripheral. Hasil pembacaan ditampilkan pada label "Time:" dan "Date:" di sebelah tombol ini.
- *Edit text* "Address" digunakan sebagai tempat untuk memasukkan alamat EEPROM yang akan ditulis atau dibaca.

- Tombol “Write EEPROM” berfungsi untuk menulis data ke EEPROM DT-I/O I2C Peripheral pada alamat yang tertulis dalam *edit text* “Address”.
- *Edit text* “Data Input” digunakan untuk memasukkan nilai data yang akan dituliskan ke EEPROM.
- Tombol “Read EEPROM” berfungsi untuk membaca data dari EEPROM pada alamat yang tertulis dalam *edit text* “Address”. Hasil pembacaan ditampilkan pada label “Data Output.”.
- Tombol “Write ADDA” berfungsi untuk menetapkan tegangan *output* yang dikeluarkan pin A/O PCF8591 pada DT-I/O I2C Peripheral sesuai dengan *track bar* di samping tombol ini.
- Tombol “Read ADDA” berfungsi untuk membaca tegangan *input* yang dimasukkan ke pin AI0 PCF8591 pada DT-I/O I2C Peripheral. Hasil pembacaan ditampilkan pada label “Output Channel 0.”. Pembacaan tegangan *input* analog maupun penulisan tegangan *output* analog dari/ke ADDA (PCF8591) sangat dipengaruhi oleh tegangan referensi yang digunakan oleh ADDA. Nilai tegangan referensi yang digunakan dalam perhitungan program ini adalah 2,5 Volt, oleh karena itu aturlah VR1 pada board DT-I/O I2C Peripheral agar tegangan referensi ADDA adalah 2,5 Volt.
- Label “Status I²C.” berfungsi untuk menunjukkan keadaan dari proses komunikasi I²C, apakah berjalan dengan baik atau tidak.

Flowchart untuk fungsi `I2c_wbyte` dalam program `Periperale.exe` adalah sebagai berikut:



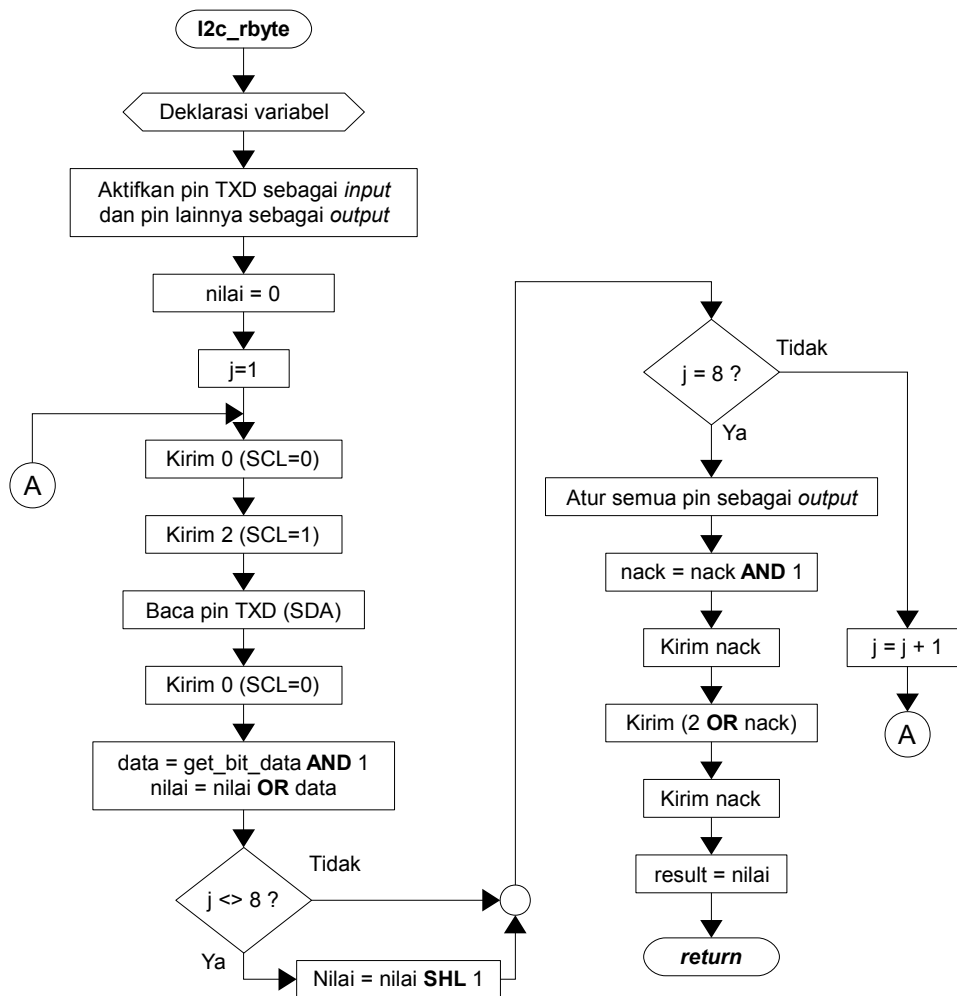
Gambar 3
Flowchart Fungsi “I2c_wbyte”

Fungsi atau rutin `I2c_wbyte` ini berguna untuk mengirimkan 1 byte data melalui komunikasi I²C, cara kerjanya adalah sebagai berikut:

1. Rutin ini memiliki variabel *input* yang diberi nama “data” dan akan berisi 1 byte data untuk dikirimkan melalui komunikasi I²C. Rutin ini juga melakukan deklarasi variabel-variabel lain yang digunakan dalam proses ini, yaitu:
 - pembagi digunakan dalam proses *masking* data.
 - i digunakan untuk menampung hasil *masking* variabel data.
 - j digunakan sebagai indeks dalam proses perulangan.
 - ack digunakan untuk menampung bit tanggapan yang diterima dari divais *slave*.

2. Pertama, rutin ini akan mengaktifkan semua pin PC-Link USBer sebagai *output*.
3. Program memberi nilai awal 80H pada variabel pembagi untuk melakukan *masking* bit 6 – bit 0 dari variabel data. Data akan dikirimkan per bit dimulai dari MSB (bit 7).
4. Program melakukan proses *masking* pada variabel data (menggunakan variabel pembagi) untuk mengambil nilai bit yang akan dikirimkan. Lalu nilai bit data hasil proses ini disimpan pada variabel *i*.
5. Program mengirimkan bit data dari variabel *i* melalui pin TXD (SDA), bersamaan dengan perubahan logika pada pin RXD (SCL) *low-high-low*.
6. Variabel pembagi dibagi (DIV) dengan 2, untuk memproses bit data berikutnya.
7. Langkah nomor 4 – 6 diulang sebanyak 8 kali.
8. Setelah menuliskan data 8 bit, program membuat pin SDA sebagai *input* untuk membaca bit tanggapan (bit ACK) dari divais *slave*.
9. Program memberi pulsa SCL *low-high* dan kemudian program membaca kondisi logika pin SDA (bit ACK).
10. Program memberi logika *low* pada pin SCL untuk mengakhiri proses pembacaan bit ACK.
11. Jika bit ACK = 1 maka fungsi ini akan menghasilkan nilai *false*, yang artinya proses gagal. Tetapi jika bit ACK = 0 maka fungsi ini akan menghasilkan nilai *true*, yang artinya proses penulisan data sukses.

Sedangkan *flowchart* untuk fungsi `I2c_rbyte` dalam program Periperhal.exe adalah sebagai berikut:



Gambar 4
Flowchart Fungsi “I2c_rbyte”

Fungsi atau rutin `I2c_rbyte` ini berguna untuk membaca 1 byte data dari divais *slave* melalui komunikasi I²C, cara kerjanya adalah sebagai berikut:

1. Rutin ini memiliki variabel *input* yang diberi nama “nack” sebagai status *acknowledgement* yang akan dikirimkan ke divais *slave* pada akhir proses pembacaan data. Rutin ini juga melakukan deklarasi variabel-variabel lain yang digunakan dalam proses ini, yaitu:
 - data digunakan untuk menampung sementara nilai bit data yang diterima dari divais *slave*.
 - nilai digunakan untuk menampung keseluruhan 8 bit data dari hasil pembacaan.
 - j digunakan sebagai indeks dalam proses perulangan.

2. Pertama, rutin ini akan mengaktifkan pin TXD (SDA) sebagai *input* dan pin lainnya sebagai *output*.
3. Program memberi nilai awal 0 (nol) pada variabel nilai.
4. Program mengirimkan sinyal SCL *low-high* untuk memulai pembacaan 1 bit data.
5. Program membaca *port* 8-bit PC-Link USBer, lalu mengembalikan pin SCL ke logika *low*.
6. Bit 0 dari hasil pembacaan *port* PC-Link USBer disimpan dalam variabel data, bit 0 ini merupakan kondisi logika pada pin SDA. Proses ini dilakukan menggunakan cara *masking* AND dengan nilai 1.
7. Program menyimpan 1 bit data yang diterima tersebut pada bit 0 variabel nilai, yaitu dengan proses OR. Lalu program menggeser variabel nilai ke kiri 1 bit. Jika perulangan mencapai 8x maka variabel nilai tidak digeser.
8. Proses 4 – 7 diulang hingga 8 kali. Setelah menerima 8 bit data, program membuat semua pin *port* PC-Link USBer sebagai *output*.
9. Program mengirimkan bit *nack* atau *ack* pada pin SDA dan pada saat yang bersamaan program mengirimkan sinyal *low-high-low* pada pin SCL.
10. Data yang tersimpan pada variabel nilai dikeluarkan sebagai nilai fungsi ini.

Inti dari aplikasi ini adalah penggunaan mode Bit Bang yang didukung oleh PC-Link USBer dengan menggunakan *driver* D2XX. Dalam mode Bit Bang, jalur I/O PC-Link USBer dapat berfungsi sebagai 8 bit *bi-directional* I/O. Di bawah ini akan dijelaskan cara mengakses jalur I/O PC-Link USBer pada mode Bit Bang menggunakan bantuan komponen TaUSBDirect yang terpasang pada Borland® Delphi™. Pembacaan dan penulisan data dari/ke jalur 8-bit I/O ini hanya dapat dilakukan per byte.

- Mengubah fungsi pin 1 (TXD) menjadi *input* dan pin 2 (RXD) menjadi *output*.

Sintaksis:

```
usb.SetBitMode($FE,$03);
```

Penjelasan:

\$FE = 1111 1110

- satu berarti *output*
- nol berarti *input*
- bit 0 = TXD ; bit 1 = RXD ; bit 2 = RTS ; bit 3 = CTS; dst sesuai dengan urutan pin J2 pada PC-Link USBer

\$03 = 0000 0011

- satu berarti aktif
- nol berarti tidak aktif

- Cara untuk menuliskan logika pada pin yang sudah difungsikan sebagai *output*, misalkan semua pin difungsikan sebagai *output*.

Sintaksis:

```
FT_Out_Buffer[0]:=$AA;
```

```
usb.Write_USB_Device_Buffer(1);
```

Penjelasan:

\$AA=1010 1010

- satu berarti berlogika *high*
- nol berarti berlogika *low*
- **\$AA** artinya pin TXD=*low*, pin RXD=*high*, dst
- penentuan logika pin ini harus dilakukan langsung 1 byte
- pin yang sebelumnya difungsikan sebagai *input* akan tetap berlogika *high*

- Cara untuk membaca pin yang sudah difungsikan sebagai *input*.

Sintaksis:

```
usb.GetBitMode;
```

```
data:=get_bit_data;
```

Penjelasan:

- sebelumnya variabel data ditentukan sebagai tipe byte atau integer
- hasil pembacaan terdapat pada variabel data, bit 0 = logika pin TXD, bit 1 = logika pin RXD, dst.
- logika pada pin yang ditetapkan sebagai *output* tidak akan berubah setelah eksekusi perintah ini.

Catatan: Komponen TaUSBdirect memiliki nama default “aUSBDirect1”, nama ini dapat diganti pada *properties* di jendela Borland® Delphi™. Pada contoh di atas nama komponen TaUSBdirect diganti dengan nama “usb”.

Emulasi komunikasi I²C menggunakan PC-Link USBer dengan mode Bit Bang ini hanya dapat bekerja normal pada divais *slave* yang memiliki antarmuka I²C *hardware*. Jika divais I²C *slave* menggunakan antarmuka I²C emulasi *software* maka aplikasi ini hanya dapat menuliskan data ke divais tersebut. Hal ini disebabkan karena kecepatan komunikasi I²C emulasi ini sangat rendah dan hampir tidak mencukupi untuk standar komunikasi I²C.

Fungsi `I2c_wbyte` dan `I2c_rbyte` yang terdapat dalam *listing* program AN ini dapat digunakan untuk berkomunikasi dengan divais I²C lainnya. Fungsi `I2c_rbyte` tidak dapat digunakan untuk membaca data dari divais I²C *slave* yang memiliki *port* I²C emulasi *software*, seperti SPC Keymatic.

Listing program terdapat pada **AN122.ZIP**.

Selamat berinovasi!

All trademarks, trade names, company names, and product names are the property of their respective owners.
All softwares are copyright by their respective software publishers and/or creators.