

DT-AVR DT-AVR *Application Note*

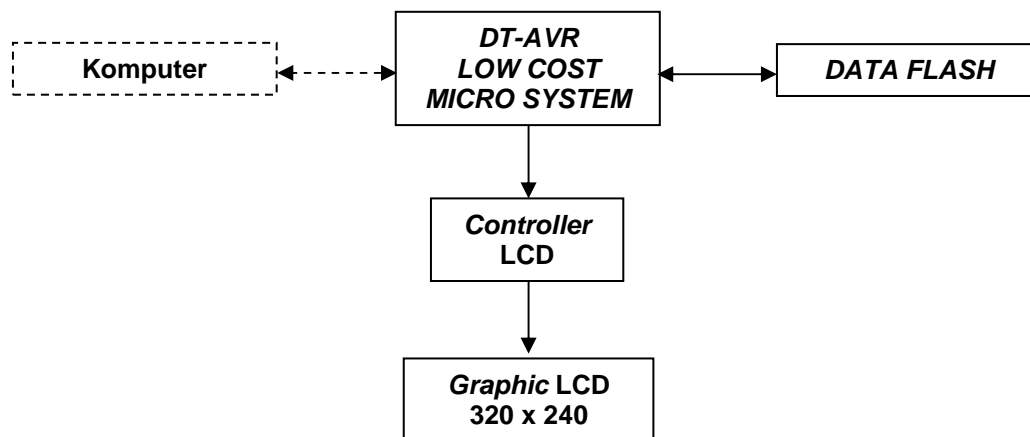
AN155 – Mini Reklame pada Graphic LCD 320 x 240

Oleh : Tim IE

LCD saat ini tidak bisa lepas dari kehidupan manusia pada jaman modern seperti sekarang. Berbagai peralatan elektronika saat ini sudah disertai dengan LCD yang berfungsi sebagai *interface* antara manusia dengan peralatan elektronika tersebut. Pada umumnya layar LCD akan menampilkan menu yang dapat dipilih oleh manusia untuk mengatur dan mengendalikan peralatan tersebut. Selain menampilkan pilihan menu, layar LCD juga dapat berfungsi untuk memberikan informasi kepada manusia, seperti suhu udara, kalender digital, dll. Pada artikel sebelumnya terdapat beberapa aplikasi yang sudah membahas cara koneksi mikrokontroler dengan *graphic LCD*. Pada edisi kali ini akan kembali membahas koneksi antara mikrokontroler dengan menggunakan *graphic LCD* yang memiliki ukuran yang cukup besar yaitu 320 x 240 pixel. Modul mikrokontroler yang digunakan dalam aplikasi ini adalah DT-AVR Low Cost Micro System yang berfungsi sebagai kontroler utama. Mikrokontroler ini nantinya akan memerintahkan LCD controller (SED1375) untuk menampilkan gambar ke *graphic LCD*. Sebagai media penyimpanan gambar, digunakan IC dataflash AT45DB041B yang memiliki antarmuka SPI. Program di dalam mikrokontroler dikembangkan menggunakan bahasa C dengan bantuan **programmer notepad** yang terintegrasi dengan WinAVR. Selain program di dalam mikrokontroler, diperlukan juga program di PC untuk membantu mengisikan gambar ke dalam dataflash. Program tersebut dikembangkan dengan menggunakan Borland Delphi 7. Aplikasi ini memerlukan modul – modul sebagai berikut ini :

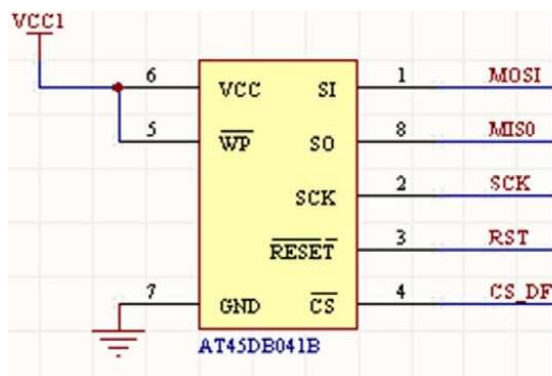
- o 1 DT-AVR Low Cost Micro System (penggantian *crystal* 4MHz menjadi 16MHz)
- o *Touch Screen Panel Graphic LCD* WC320240A-FCI-NUTS
- o *Data flash* AT45DB041B
- o *Embedded Memory LCD Controller* SED1375

Adapun blok diagram secara keseluruhan adalah sebagai berikut:



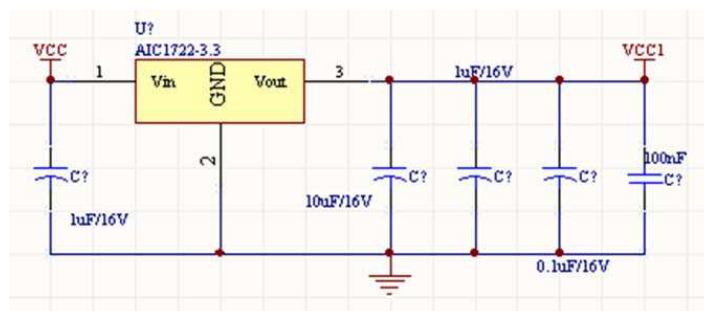
Gambar 1
Blok Diagram *Touch Screen Panel Graphic LCD*

Hubungan antara modul-modul tersebut adalah sebagai berikut:



Gambar 2
Rangkaian IC DataFlash AT45DB041B

IC DataFlash AT45DB041B hanya dapat bekerja pada level catu daya 3.3VDC, oleh sebab itu diperlukan rangkaian catu daya 3,3VDC.

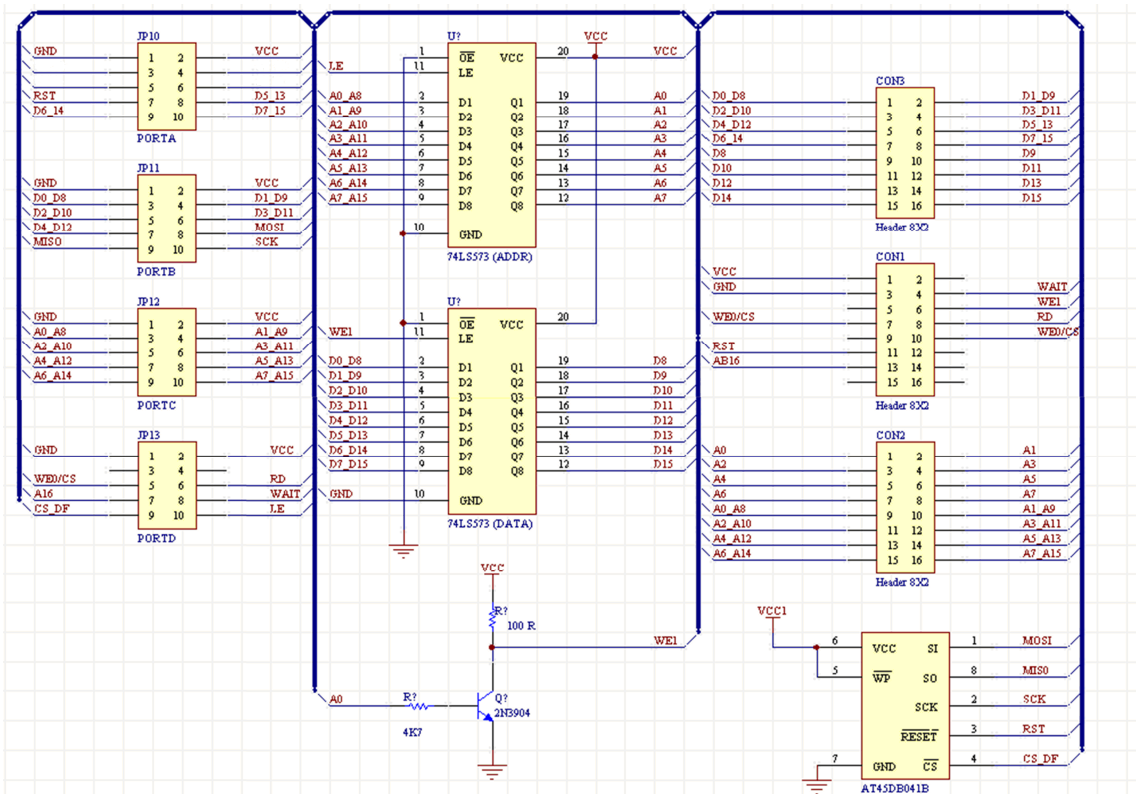


Gambar 3
Rangkaian catu daya 3,3VDC

DataFlash AT45DB041B	DT-AVR Low Cost Micro System
MOSI	PB.5
MISO	PB.6
SCK	PB.7
RST	PA.4*
CS_DF	PD.6*

*) Pin ini tidak mutlak dan dapat diganti pin lain dengan cara mengubah program

Tabel 1
Hubungan DataFlash dengan DT-AVR Low Cost Micro System



Gambar 4
Rangkaian Touch Screen LCD

Kontroler LCD SED1375TM menggunakan lebar jalur data 16-bit, sedangkan mikrokontroler pada DT-AVR Low Cost Micro System memiliki keterbatasan pin I/O untuk dapat mengirimkan data dengan lebar jalur data 16-bit. Oleh sebab itu diperlukan tambahan 2 buah IC 74LS573 untuk membantu mikrokontroler dalam mengirimkan data ke kontroler LCD tersebut. Rangkaian tersebut dapat dilihat pada Gambar 2.

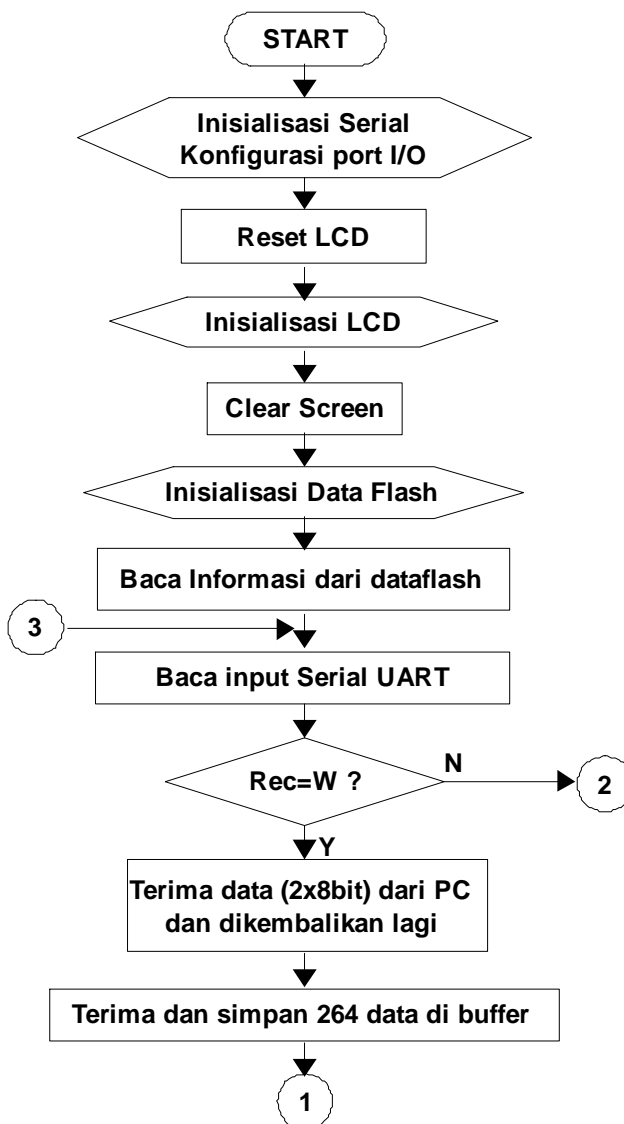
PIN LCD	PIN Kontroler LCD	Simbol
1	1	FRM
2	2	LOAD
3	3	CP
4	4	DISP
5	5	VDD
6	6	VSS
7	7	VEE
8	8	DB7
9	9	DB6
10	10	DB5
11	11	DB4
12	12	DB3
13	13	DB2
14	14	DB1
15	15	DB0

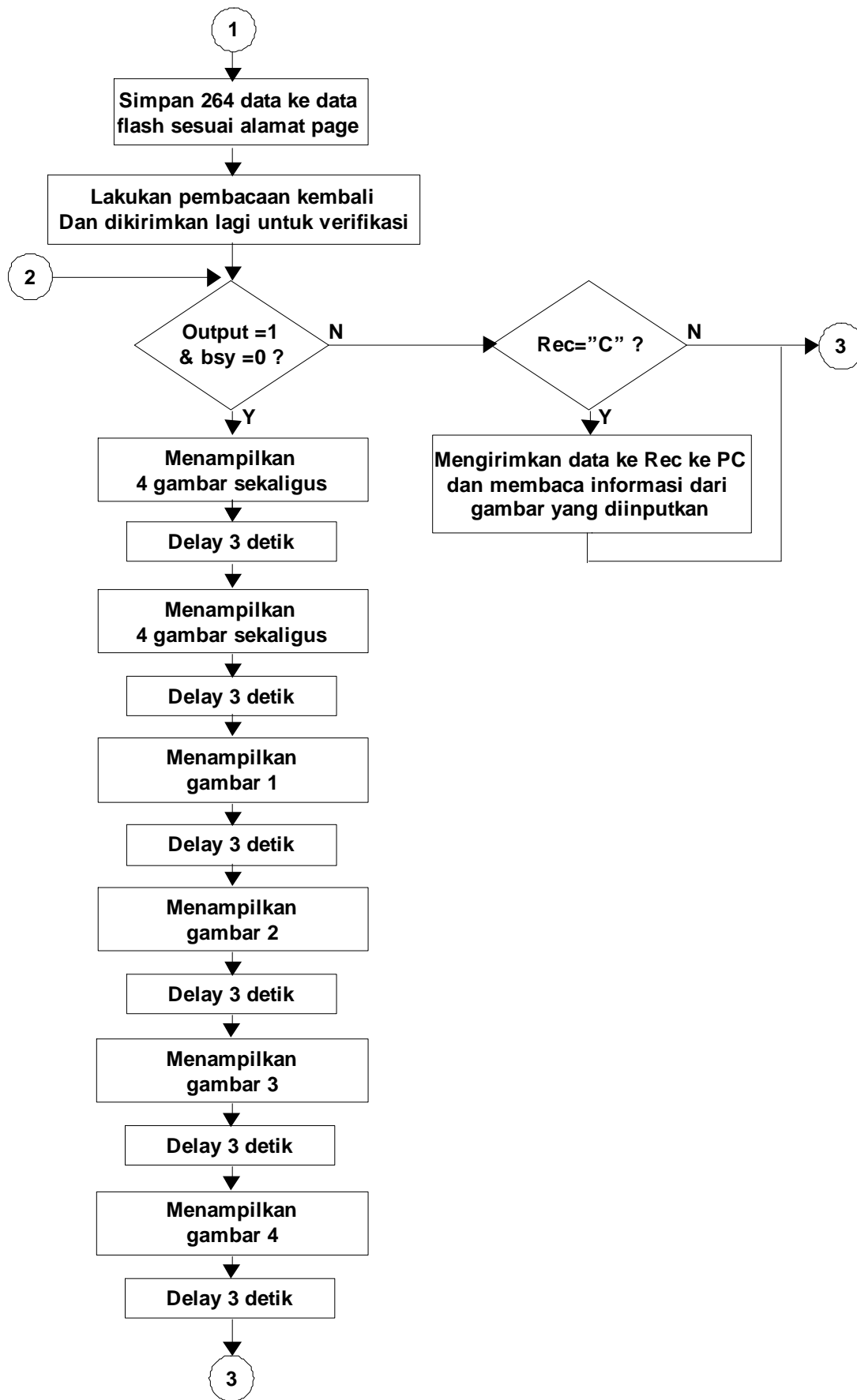
Tabel 2
Hubungan Graphic LCD WC320240A-FCI-NUTS dengan Kontroler LCD SED1375

Keterangan koneksi detail dan keterangan lebih lengkap, dapat dilihat pada datasheet masing-masing modul tersebut. Periksa kembali hubungan antar modul tersebut sesuai dengan datasheet dan periksa ulang jalur GND dari masing- masing modul yang terhubung. Pastikan jalur GND tersebut saling terhubung satu sama lain. Gunakan eksternal programmer seperti DT-HiQ AVR In-System Programmer, DT-HiQ AVR USB ISP ataupun divais lain yang mendukung koneksi ISP AVR 10-pin untuk mengisikan kode program AVR_LCD_320x240.hex ke dalam mikrokontroler DT-AVR Low Cost Micro System. **Ketika mengisikan kode program ke dalam mikrokontroler DT-AVR Low Cost Micro System dengan menggunakan ISP, semua koneksi antara DT-AVR Low Cost Micro System ke IC DataFlash AT45DB041B harus dilepas!** Hal ini dikarenakan metode programming ISP menggunakan antarmuka SPI dan IC AT45DB041B juga menggunakan antarmuka yang sama. Setelah kode program sudah dimasukan ke dalam mikrokontroler, masukan data gambar yang akan ditampilkan di LCD ke dalam DataFlash dengan menggunakan bantuan program **BMPFlashProgrammer.exe**. Konfigurasi DT-AVR Low Cost Micro System agar dapat dihubungkan dengan **BMPFlashProgrammer.exe** adalah sebagai berikut :

- o Pastikan koneksi antara DT-AVR Low Cost Micro System dengan DataFlash sudah benar
- o Pastikan eksternal programmer (DT-HiQ AVR In System Programmer, DT-HiQ AVR USB ISP atau programmer yang lain tidak terhubung dengan DT-AVR Low Cost Micro System)
- o Konfigurasikan jumper J4 dan J5 pada DT-AVR Low Cost Micro System dalam posisi 1-2
- o Hubungkan DT-AVR Low Cost Micro System ke PC dengan menggunakan kabel serial yang tersedia pada paket pembelian DT-AVR Low Cost Micro System
- o Setelah pengisian data selesai, Reset DT-AVR Low Cost Micro System untuk melihat tampilan di LCD

Flowchart dari program AVR_LCD_320x240.c adalah sebagai berikut:





Gambar 5
Flowchart main program AVR_LCD_320x240

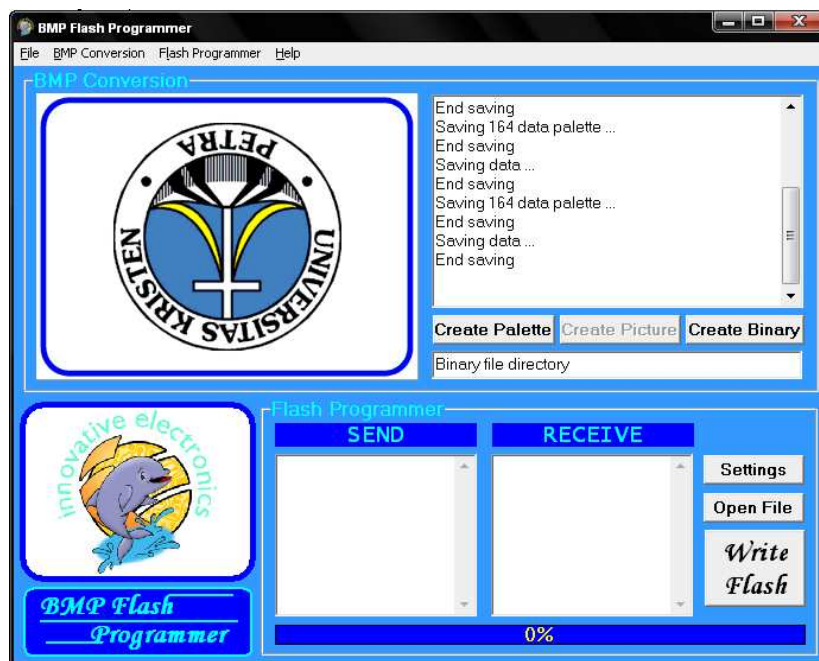
Program AVR_LCD_320x240.c akan diproses sebagai berikut:

1. Inisialisasi serial UART 8 bit, 1 stop bit, none parity dan baud rate 38400 bps dan konfigurasi port I/O
2. Kemudian reset LCD dan inisialisasi LCD. Setelah terinisialisasi, maka layar LCD akan dibersihkan dengan cara memberi 1 jenis warna.
3. Pengaturan jalur komunikasi SPI untuk dataflash
4. Setelah jalur komunikasi selesai di atur, kemudian membaca informasi file gambar yang ada di dalam dataflash, seperti jumlah gambar, jumlah *palette*, alamat *page* dan *byte* dari data pertama *palette*, halaman dan *byte* dari setiap gambar serta ukuran dari setiap gambar. Jika telah selesai membaca informasi, maka akan menghasilkan nilai "1" yang diisi pada variabel "output" yang menandakan bahwa informasi telah didapat.
5. Kemudian variabel *bsy*=0. Variabel ini berfungsi untuk mendakan sistem komunikasi serial UART. Jika *bsy* bernilai "1" maka sedang antara mikrokontroler dan PC sedang melakukan komunikasi serial, Jika "0" maka tidak ada komunikasi antara PC dengan mikrokontroler melalui jalur komunikasi serial UART.
6. Cek data inputan dari komunikasi serial UART yang disimpan pada variabel "Rec". Jika Rec berisi karakter "W" berarti PC (program **BMPFlashProgrammer.exe**) akan melakukan pengisian data ke dataflash. Apabila bukan karakter "W", maka program akan lompat ke langkah no 13.
7. Setelah masuk ke bagian pembacaan data serial UART, maka variabel Rec diisi dengan nilai "0" dan *bsy* diisi dengan nilai 1. Pengisian variabel Rec = 0 berfungsi agar tidak terjadi pengisian data ke dalam dataflash setelah mengisi data ke dalam dataflash.
8. Program menerima pengiriman data *page* yang merupakan *page*/halaman dari dataflash dan disimpan di dalam variabel array "address[0]" dan "address[1]". Dikarenakan jumlah halaman dari dataflash adalah 2048 (memerlukan 11 bit data) dan sistem komunikasi serial adalah 8 bit, maka data yang diterima sebanyak 2 x 8 bit. Setelah di dapat kedua data tersebut, maka digabungkan menjadi satu kesatuan di dalam variabel *page*. Kemudian data di variabel *page* dipecah kembali menjadi 8 bit dan dikirimkan ke serial UART untuk proses verifikasi oleh program **BMPFlashProgrammer.exe**.
9. Setelah proses verifikasi *page* selesai, maka **BMPFlashProgrammer.exe** akan mengirimkan data sebanyak 264 data dari **page pertama** dataflash ke dalam mikrokontroler melalui komunikasi serial UART. Jumlah 264 data merupakan jumlah data yang disimpan pada setiap halaman dataflash. Hasil pembacaan 264 data oleh mikrokontroler di simpan dalam variabel array "*buffer[]*"
10. Setelah selesai, maka mikrokontroler akan mengisi data ke dalam dataflash dengan mengacu pada variabel *page* dan *buffer[]* yang didapat dari **BMPFlashProgrammer.exe** (sesuai dengan langkah no 8 dan 9).
11. Data di dalam dataflash di baca kembali oleh mikrokontroler mengacu pada isi dari variabel *page* sebelumnya. Setelah di dapat hasil pembacaan, kemudian data tersebut dikirimkan lagi ke PC melalui komunikasi serial untuk proses verifikasi data.
12. Jika data verifikasi benar, maka program di dalam **BMPFlashProgrammer.exe** akan secara otomatis menambah nilai dari variabel *page*. Kemudian program diulangi seperti pada langkah no 6. Hal ini terdapat perbedaan pada langkah no 9, **BMPFlashProgrammer.exe** tidak lagi mengirimkan data sebanyak 264 data dari **page pertama**, melainkan mengirimkan data sebanyak 264 data dari **page berikutnya**. Proses pengulangan ini berakhir ketika semua data gambar yang akan dimasukkan ke dalam dataflash sudah selesai.
13. Jika tidak menerima karakter "W" dari input serial UART, maka di cek apakah variabel *output*=1 dan *bsy*=0. Jika benar, maka program akan membaca semua data gambar di dalam dataflash. Pada program ini telah dikonfigurasi untuk 4 buah gambar yang berbeda. Sehingga ketika selesai membaca data gambar, maka program di dalam mikrokontroler akan membagi layar LCD menjadi 4 buah bagian, dan mengecilkan data gambar serta menampilkan 4 buah gambar tersebut menjadi 4 bagian kecil. Jika tidak, program akan lompat ke langkah no 15.
14. Setelah 3 detik menampilkan gambar tersebut, maka program akan menampilkan gambar pertama ke layar LCD dengan konfigurasi layar penuh (full screen). Kemudian program ditunda 3 detik untuk menampilkan gambar tersebut. Setelah 3 detik, kemudian program akan menampilkan gambar ke 2, 3 dan ke 4 dengan jeda waktu selama 3 detik. Kemudian program akan menuju ke langkah no 16.
15. Periksa isi dari variabel Rec. Jika berisi karakter "C" maka program akan mengirimkan kembali karakter tersebut ke PC yang berarti semua pengiriman data dari PC ke mikrokontroler untuk mengisi data ke dataflash telah selesai. Kemudian membaca kembali informasi data gambar di dalam dataflash. Jika tidak program akan langsung menuju ke langkah no 6.
16. Setelah selesai, program akan terus berulang ke langkah 6.



Gambar 6
Tampilan Menu Utama pada LCD

Program “BMPFlashProgrammer.exe” yang dikembangkan dengan Borland Delphi 7. Program ini berfungsi untuk membuat file binary gambar dan mentransfer data tersebut ke data flash. Tampilan program tersebut dapat dilihat pada gambar berikut ini.



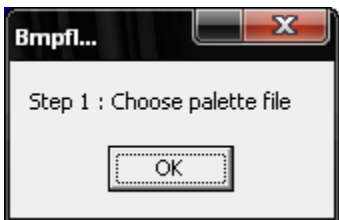
Gambar 7. Tampilan “BMPFlashProgrammer.exe”

Program ini mempunyai terbagi menjadi 2 bagian utama, bagian pertama adalah **BMP Conversion** dan bagian kedua adalah **Flash Programmer**.

1. **BMP Conversion** berfungsi untuk mengubah file gambar menjadi file binary sebelum dikirim ke data flash.
2. **Flash Programmer** berfungsi untuk mentransfer data melalui DT-AVR Low Cost Micro System ke data flash. Data ini ditransfer secara serial dan juga terdapat fitur verifikasi atau pengecekan kembali.

BMP Conversion mempunyai 3 tombol, yaitu **Create palette**, **Create Picture** dan **Create Binary**. Selain itu juga terdapat **BMP picture window**, **memo** dan **textbox**.

1. "BMP picture window" berfungsi untuk menampilkan gambar yang ingin dikonversi dan simpan.
2. "Memo" berfungsi untuk menampilkan informasi dan "textbox" berfungsi untuk menampilkan directory file binary yang telah dibuat dengan tombol "Create Binary".
3. Tombol **Create Palette** berfungsi untuk mengkonversi file BMP menjadi palette file ".pal". Tombol ini akan menjalankan rutin untuk memeriksa data warna (RGB) dari tiap pixel dalam gambar dan membandingkannya dengan kumpulan data warna (RGB palette). Jika data RGB tersebut belum ada, maka program akan menambahkan data tersebut. Palette ini dibutuhkan karena setiap data RGB menghabiskan 3 byte data (18 bit). Untuk menghemat memori, maka dibuat palette sehingga data gambar hanya berupa alamat dari palette yang ukurannya hanya 1 byte per pixel. Ukuran dan kedalaman warna harus disesuaikan dengan LCD. Ukuran gambarnya maksimum 320x240 pixel dan kedalaman warnanya 256 warna. Untuk mengubah ukuran pixel, dapat menggunakan "ACDSee" atau program yang sejenis. Sedangkan untuk mengubah kedalaman warna, dapat menggunakan "Microsoft Paint" yang merupakan program bawaan "Microsoft Windows". Caranya dengan melakukan *save as file* gambar menjadi tipe 256 Color Bitmap.
4. Tombol **Create Picture** berfungsi membuat picture file dari gambar BMP. Tombol ini hanya akan aktif bila tombol **Create palette** sudah di-click. Hal ini mencegah kesalahan pembuatan picture file karena seperti dijelaskan sebelumnya, proses **Create Picture** ini akan menghasilkan data gambar yang mana data warna untuk masing-masing pixel adalah alamat atau indeks dari data RGB dalam data palette, jadi bila data palette tidak ada, maka program tidak bisa membuat picture file-nya.
5. Tombol **Create Binary** akan menjalankan rutin program yang paling kompleks. Rutin program ini berfungsi untuk membuat file binary yang merupakan gabungan dari berbagai data. Oleh karena itu, ditambahkan pop up window sehingga pengguna program ini tahu langkah yang harus dilakukan selanjutnya.



Gambar 8
Langkah pertama rutin **Create Binary** dengan memilih *palette file*

Langkah	Keterangan
1	Memilih palette file
2	Memilih file gambar
3	Memilih gambar selanjutnya. Jika semua gambar telah selesai dimasukkan, pilih <i>cancel</i>
4	Masukan informasi nama file dan simpan
5	Ketikan nama binary file dan simpan

Tabel 3
Urutan proses **Create Binary**

Pada step 1 dan 2 program akan menghitung jumlah file yang akan dimasukkan ke dalam file informasi. Step 3 akan menyimpan hasil-hasil perhitungan itu ke dalam file informasi. Step 4 akan menggabungkan semua data menjadi binary file dengan nama yang diketikkan. *Directory file binary* ini akan ditampilkan pada "textbox".

No.	Data	Penjelasan	n	16bit data
0	65	A		
1	68	D		
2	65	A		
3	4	Total Picture		
4	12	Total palette*3 LSB		492
5	30	Total palette*3 MSB		
6	0	Palette page LSB		0
7	0	Palette page MSB		
8	6	Palette byte LSB		38
9	2	Palette byte MSB		
10	2	Picture page LSB	0	2
11	0	Picture page MSB	0	
12	2	Picture byte LSB	0	2
13	0	Picture byte MSB	0	
14	0	Picture width LSB	0	320
15	20	Picture width MSB	0	
16	240	Picture height	0	240
17	4	Picture page LSB	1	292
18	18	Picture page MSB	1	
19	2	Picture byte LSB	1	242
20	15	Picture byte MSB	1	
21	0	Picture width LSB	1	320
22	20	Picture width MSB	1	
23	240	Picture height	1	240
24	7	Picture page LSB	2	583
25	36	Picture page MSB	2	
26	10	Picture byte LSB	2	218
27	13	Picture byte MSB	2	
28	0	Picture width LSB	2	320
29	20	Picture width MSB	2	
30	240	Picture height	2	240
31	10	Picture page LSB	3	874
32	54	Picture page MSB	3	
33	2	Picture byte LSB	3	194
34	12	Picture byte MSB	3	
35	0	Picture width LSB	3	320
36	20	Picture width MSB	3	
37	240	Picture height	3	240

Tabel 4
Contoh Data Yang Terdapat Pada File Informasi

Tabel 4 merupakan isi dari file informasi yang dibuat oleh **BMPFlashProgrammer.exe**. Data yang terdapat pada file informasi tergantung dari jumlah gambar dan jumlah palette dan juga jumlah file informasi sendiri. Isi dari data file informasi akan berbeda dengan hasil yang didapat. Hal ini terjadi dikarenakan gambar yang diinputkan berbeda. Data *field* pada file informasi ini umumnya berupa data 16 bit kecuali *height* dari gambar. Karena data yang dikirim harus 1 *byte* (8 bit) maka data tersebut dibagi menjadi 2 *byte* (MSB dan LSB).

Total picture menunjukkan jumlah gambar yang dipakai. Pada tabel di atas tertulis Total palette*3 karena total palette menunjukkan banyaknya warna RGB yang digunakan. Karena 1 palette file RGB membutuhkan 3 byte data maka total palette harus dikali 3 agar jumlah *byte*-nya tepat. *Page* dan *byte* baik pada *palette* maupun *picture* menunjukkan alamat dari data pertama file *palette* dan *picture*. *Picture width* dan *height* adalah lebar dan tinggi dari gambar. Untuk mencari *picture page* dan *picture byte* digunakan rumus berikut ini.

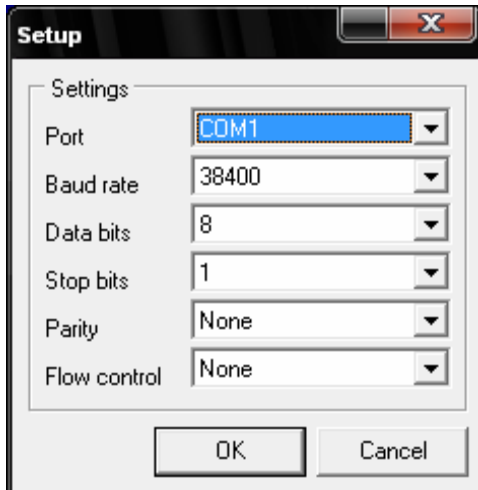
Picture page ke-n = (total info + total palette*3 + (total picture *n)) div 264

Picture byte ke-n = (total info + total palette*3 + (total picture*n)) mod 264

Note : n = 0, 1, 2, 3 (Lihat tabel 4)

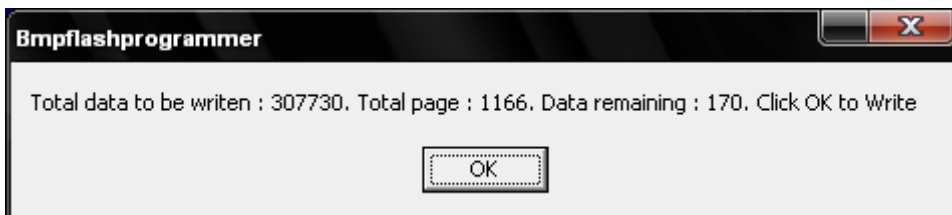
Urutan data yang terdapat dalam binary file yaitu *file informasi*, *file palette*, dan *file picture*. Jumlah *file picture* tergantung banyaknya gambar yang digunakan. Bagian **Flash programmer** terdapat 3 tombol yaitu **Settings**, **Open File**, dan **Write Flash**. Selain itu terdapat **Send memo**, **Receive memo** dan **loading bar**.

1. **Send memo** akan menampilkan data yang dikirimkan.
2. **Receive memo** akan menampilkan data yang diterima. Di "memo" akan terlihat huruf "W" dan "C". "W" berarti Write dan "C" berarti Complete. Setelah huruf "W" akan terlihat page kemudian diikuti dengan 264 data. Page dan data ini ditampilkan dalam hexadesimal. Data pada "Send" dan "Receive memo" selalu sama jika komunikasi serial berlangsung baik. Jika pengiriman gagal, maka akan terlihat perbedaan data pada send dan receive file. Pada bagian bawah terdapat Loading bar yang agar dapat mengetahui berapa persentase data yang telah dikirimkan.
3. Tombol **Settings** dipakai untuk mengatur konfigurasi dari komunikasi serial port. *Setting* default-nya COM1, baud rate 38400, data bit 8, stop bit 1, no parity dan no flow control. COM dapat diatur sesuai komputer yang digunakan. Sedangkan setting lainnya disesuaikan dengan konfigurasi pada serial AVR.



Gambar 9
Pengaturan COM

4. Tombol "Open File" berfungsi membuka file binary yang ingin dikirimkan ke data flash. Directory file yang dibuka akan muncul di "textbox BMP Conversion". Tombol "Write Flash" berfungsi untuk mengirim data binary sesuai directory file secara serial ke data flash melalui perantara AVR Low Cost Micro System sebagai writer-nya. Sebelum data mulai dikirimkan, maka akan muncul informasi data yang akan ditulis, Total Page dan remaining (sisa data pada page terakhir) seperti pada Gambar 10



Gambar 10
Informasi Pengiriman Data

Pengiriman data ini bersifat 2 arah. Data yang dikirimkan akan ditampilkan di **Send memo** dan data yang diterima akan ditampilkan di **Receive memo**. Data yang dikirimkan selalu dibandingkan, setelah dibandingkan baru kemudian ditampilkan di **Send** dan **Receive memo**. Bila pengiriman gagal, maka di setiap "memo" akan tertampil 2 data yang berbeda. Misalnya data ke [263] berbeda, maka akan ditampilkan tulisan [263] 3B >< 20 di **send memo** dan di **receive memo** akan ditampilkan tulisan [263] 20 >< 3B. Bila pengiriman gagal juga akan muncul **pop up window** dengan tulisan "**WriteFailed**". Bila pengiriman berhasil 100%, maka akan keluar tulisan "**Write Successful**". Bila muncul tulisan "**Check your hardware**" berarti *hardware* harus diperiksa koneksi kabel dan catu daya yang digunakan. Bila "**Check your hardware**" muncul lagi berarti harus me-reset hardware dengan mematikan dan menyalakan catu daya. Untuk memudahkan penggunaan, BMP Flash Programmer telah menyediakan "**Help Menu**" yang dapat diakses dengan memilih **Help** atau menekan tombol **shortcut F1**.

Listing program AVR_LCD_320x240.c dan BMPFlashProgrammer.exe terdapat pada **AN155.ZIP**

Selamat berinovasi!

Programmer Notepad is is Copyright © 2002-2010 Simon Steele, All Rights Reserved
All trademarks, trade names, company names, and product names are the property of their respective owners.