

Teknik Update Program Secara On-line (Over The Air, OTA) Pada Sistem Berbasis Mikrokontroler Menggunakan Modem GPRS

Eru Puspita
Politeknik Elektronika Negeri Surabaya (PENS)
Kampus PENS Keputih Sukolilo Surabaya 60111, Indonesia
eru@eepis-its.edu

Abstrak

Sarana update program secara OTA menjadi kebutuhan saat ini yang akan memudahkan dalam pengembangan suatu program. Update secara OTA memerlukan beberapa komponen utama yang meliputi: koneksi internet, program koneksi dan download ke server OTA, memori penampung program dan bootloader. Pada proyek penelitian ini ingin mengimplementasikan teknik OTA pada sistem sederhana buatan sendiri yang berbasis mikrokontroler AVR ATmega32, modem GPRS dengan kemampuan TCP/IP Stack, memori serial EEPROM AT24C256. Dari hasil pengujian menunjukkan kecepatan komunikasi dari modem ke mikrokontroler pada 38400 bps menunjukkan sistem bekerja dengan baik. Kurang dari itu menyebabkan buffer Modem mengalami overflow dan lebih dari itu menunjukkan buffer serial mikrokontroler mengalami overflow.

Keywords: OTA, GPRS, TCP/IP Stack, Bootloader.

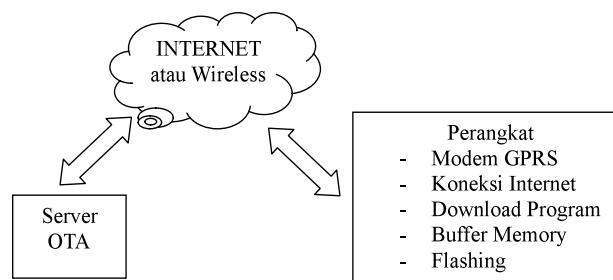
1. Pendahuluan

Dewasa ini banyak sistem kontrol, sistem embedded, sistem monitoring dan sejenisnya yang menerapkan piranti atau peralatan yang dapat deprogram. Dan sering terjadi diperlukan proses untuk memperbaharui program yang sudah ditanam pada piranti atau device tertentu. Salah satu cara yang sudah lama diterapkan pada piranti atau peralatan tertentu adalah dengan menggunakan proses update (memperbaharui program) melalui media nir kable (wireless) jarak jauh, misalkan menggunakan GPRS, wifi atau yang sejenisnya. Update program yang seperti ini sering disebut sebagai Over The Air (OTA).

Penggunaan OTA akan memudahkan proses update kalau piranti atau peralatan yang digunakan berjumlah banyak dan terpisah pada jarak yang jauh yang akan menyulitkan kalau dilakukan update program secara konvensional (langsung diubah pada piranti atau

peralatan yang bersangkutan). Contoh aplikasi yang sering dibahas adalah jaringan sensor, Automatic Meter Reading (AMR), dan sebagainya.

Update program secara OTA akan mudah dilakukan pada sistem yang sudah menyanggah sebagai perangkat cerdas, misalkan smartphone atau smart device lainnya (android [5], blackberry [6], windows smartphone [7], dll). Berbagai fitur yang diperlukan untuk proses update secara OTA sudah dimiliki oleh sistem tersebut. Hal ini tidak dengan perangkat sederhana yang (terutama yang dibuat sendiri) berbasis mikrokontroler dan/atau tidak dilengkapi dengan OS atau firmware khusus.



Gambar 1. Mekanisme umum update OTA

Kelengkapan yang diperlukan untuk melakukan update program secara OTA hampir tidak ada pada sistem yang sederhana yang dibuat sendiri. Kelengkapan yang harus dimiliki untuk bisa melakukan update program secara OTA antara lain: kemampuan koneksi ke internet, kemampuan download data atau program dari server OTA, memori untuk menampung data atau program selama proses download, kemampuan self flashing (pemrograman sendiri).

2. Tinjauan Pustaka

Beberapa proyek dan penelitian yang berkaitan dengan OTA dapat dijelaskan secara ringkas berikut ini.

[1] adalah suatu proyek yang berkaitan dengan cara melakukan update OTA pada perangkat jaringan sensor menggunakan 802.15.4 dan ZigBee.

[3] adalah suatu fasilitas OTA yang dilakukan oleh operator wireless CDMA di Alaska.

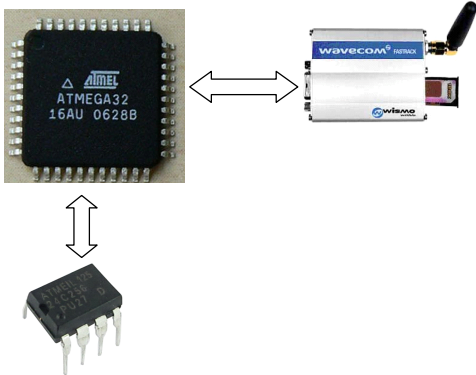
[4] adalah suatu proyek pemrograman OTA untuk jaringan sensor menggunakan mikrokontroler AVR Xmega, modem GPRS dan SD card.

3. Konfigurasi Sistem

Berbagai bentuk konfigurasi dan komponen atau modul dapat digunakan untuk membangun sistem OTA. Pada penelitian ini digunakan mikrokontroler dari keluarga AVR ATmega32 dan modem Wavecom yang dilengkapi dengan TCP/IP Stack. Memori untuk buffer dipilih AT24C256 dengan alasan lebih mudah dan murah dibandingkan misalkan menggunakan SD card atau RAM.

Di sini tidak digunakan perangkat yang besar dan cepat (baik mikrokontroler maupun memori) agar dapat diketahui batas-batas yang dapat digunakan jika hanya menggunakan perangkat yang murah dan sederhana.

Untuk koneksi internet dapat digunakan kartu operator apapun asalkan sesuai dengan modem yang digunakan dan dapat melakukan koneksi ke internet.

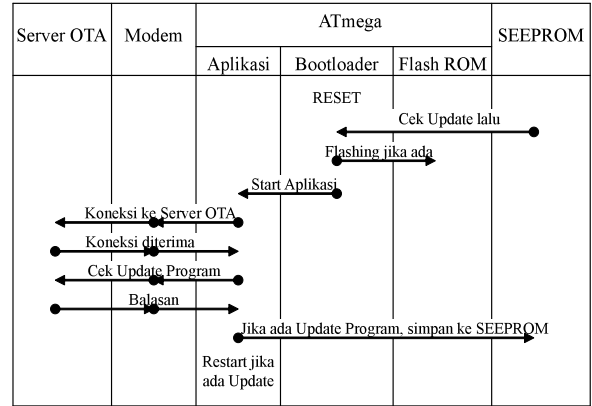


Gambar 2. Blok diagram update OTA.

3.1. Update OTA

Update OTA secara umum berbeda dengan pemrograman OTA. Pemrograman OTA memerlukan bootloader yang secara penuh dapat menjalankan fungsi OTA. Sedangkan update OTA hanya mensyaratkan bootloader harus memiliki fungsi self flashing (programming) dan download OTA dilakukan di program utama. Hal ini menyebabkan mikrokontroler harus memiliki program utama terlebih dulu (yang diisikan secara manual).

Proses update OTA dapat digambarkan dalam bentuk diagram berikut ini.



Gambar 3. Diagram alir update OTA.

Saat pertama modul dijalankan, modul akan menjalankan program yang ada pada bootloader terlebih dulu. Program boot loader akan melakukan pengujian pada serial EEPROM untuk mengetahui jika ada update program yang tersimpan pada serial EEPROM hasil download sebelumnya. Jika ada akan dilakukan prosedur flashing (pemrograman) dari serial EEPROM ke flash ROM dari AVR.

Ada update program atau tidak, bootloader melanjutkan aliran program ke program aplikasi (program utama dari sistem aplikasi). Setelah program aplikasi menjalankan beberapa inisialisasi, program segera melakukan prosedur pengujian ada atau tidak update program secara OTA.

Prosedur update OTA dimulai dengan melakukan permintaan koneksi terlebih dulu ke server OTA dan akan diulang sampai beberapa kali jika mengalami kegagalan. Jika permintaan berhasil, maka berikutnya dilanjutkan dengan memintah update program yang sesuai dengan model hardware, versi dan sebagainya.

Server OTA akan mencari pada file server file update yang sesuai dengan permintaan dan ada atau tidak akan mengembalikan hasilnya ke aplikasi. Jika program aplikasi mengetahui ada program update yang harus di-download, maka program akan melakukan prosedur pembacaan program update dan disimpan ke serial EEPROM dan setelah selesai akan melakukan prosedur restart sistem.

3.2. Server OTA

Server OTA dapat dibuat dengan berbagai cara. Pada penelitian ini digunakan server web untuk menangani prosedur OTA. Server web digunakan di sini untuk memudahkan penyediaan infrastruktur server OTA (dapat menggunakan server web apapun).

Pada server OTA dipasang program untuk melayani permintaan file update program dalam bentuk script php atau lainnya.

Program pada server OTA berupa program sederhana yang menerima permintaan pencarian file

update program dengan cara menerima parameter berupa model mesin, versi dari mesin dan sebagainya dalam bentuk HTTP in-line parameter (GET parameter).

Setelah menerima permintaan pencarian, maka program akan mencari dalam file server (current directory) file update program yang sesuai dan memiliki versi terakhir. Jika ketemu, file tersebut akan dikembalikan ke mesin peminta. Contoh program server OTA.

```
<?php
$Modul=@$_GET['Modul'];
$Model=@$_GET['Model'];
$Versi=@$_GET['Versi'];
$v=doubleval($Versi); $f="";
foreach (glob($Modul.$Model."v*.hex") as $file){
    $ff=str_replace($Modul.$Model."v","",$file);
    $n=doubleval($ff);
    if($n>$v){
        $v=$n;
        $f=$file;
    }
}
if($f!="") include $f; else echo "*";
?>
```

Gambar 4. Program server OTA menggunakan script PHP

3.3. Modem GPRS Dengan TCP/IP Stack

Modem yang digunakan dapat berbentuk apapun asalkan dapat melakukan koneksi ke internet. Modem yang umum dijumpai untuk perangkat berbasis mikrokontroler adalah modem GPRS dengan interface komunikasi serial. Untuk memudahkan pemrograman dapat digunakan modem GPRS yang memiliki TCP/IP Stack. Contoh koneksi ke server menggunakan AT command yang dilengkapi dengan TCP/IP Stack seperti pada gambar 5.

Langkah-langkah untuk melakukan koneksi dengan server sebagai berikut:

- Buka modul TCP/IP stack pada modem
- Jalankan modul TCP/IP stack
- Persiapkan komunikasi menggunakan GPRS
- Atur konfigurasi internet sesuai dengan kartu yang digunakan, meliputi APN, user name, dan password
- Koneksi menggunakan GPRS
- Koneksi ke server sebagai client
- Buka komunikasi data dengan server (mengubah mode saluran komunikasi dari perintah AT ke mode data)
- Berikan perintah ke server dan terima respon dari server
- Ubah saluran komunikasi ke modem dari mode data ke mode perintah AT kembali
- Tutup komunikasi data dengan server
- Diskonek dari GPRS
- Tutup layanan GPRS
- Hentikan modul TCP/IP stack

```
AT+WOPEN=1 ..... Open TCP/IP Stack
OK
AT+WIPCFG=1 ..... Start TCP/IP Stack
OK
AT+WIPBR=1,6 ..... Open GPRS bearer
OK
AT+WIPBR=2,6,11,"3data" ... Set APN
OK
AT+WIPBR=2,6,0,"3data" ... Set User Name
OK
AT+WIPBR=2,6,1,"3data" ... Set Password
OK
AT+WIPBR=4,6,0 ..... Start GPRS bearer
OK
AT+WIPCREATE=2,1,URL,80 ... Connect ke Server
OK
+WIPREADY: 2,idx
AT+WIPDATA=2,idx,1 ..... Open data
CONNECT
..... komunikasi data
..... client - server
+++ ..... Kembali ke perintah AT
OK
AT+WIPCLOSE=2,idx ..... Close data
OK
AT+WIPBR=5,6 ..... Stop GPRS bearer
OK
AT+WIPBR=0,6 ..... Close GPRS bearer
AT+WIPCFG=0 ..... Stop TCP/IP Stack
OK
```

Gambar 5. Perintah AT untuk Koneksi ke Server

Contoh perintah update OTA ke server OTA menggunakan perintah HTTP tampak seperti gambar 6.

```
GET
/~eru/Update/UVCom/M3201/index.php?Modul=UVCom&
Model=M3201&Versi=01.00 HTTP/1.0<LF>
Host: ~eru/Update<LF>
Connection: Keep-Alive<LF>
Accept: */*<LF>
Accept-Language: en-us<LF>
<LF>
<EOF>
```

Gambar 6. HTTP Request untuk update OTA

Yang harus diperhatikan dari modem GPRS ini adalah kecepatan komunikasi serial yang digunakan antara modem dengan mikrokontroler. Kecepatan koneksi modem dengan internet tergantung dari kecepatan paket internet yang digunakan dan kelas modem. Sebagai contoh, umumnya modem GPRS yang digunakan bekerja pada kelas 10 yang memiliki kecepatan sekitar 32 kbps s/d 48 kbps. Kecepatan komunikasi serial antara modem dengan mikrokontroler antara 300 bps s/d 115200 bps.

Kecepatan komunikasi serial antara modem dengan mikrokontroler harus sesuai dengan kecepatan internet (GPRS) dari modem dan kecepatan penyimpanan data ke serial EEPROM.

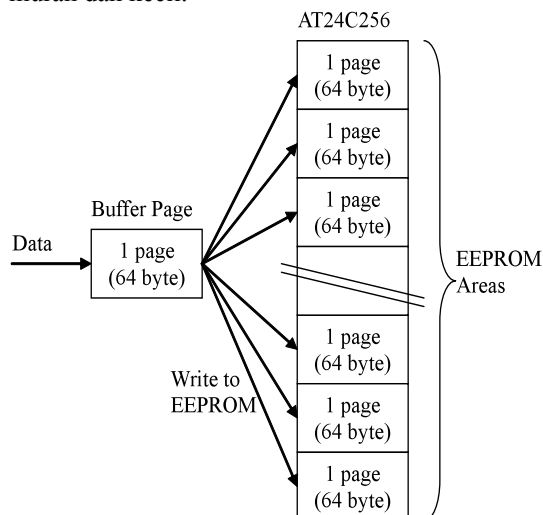
Penggunaan kecepatan serial yang lebih rendah dari kecepatan GPRS menyebabkan modem mengalami

buffer overflow yang berakibat banyak data yang hilang. Kecepatan komunikasi serial yang lebih tinggi dari kecepatan GPRS menyebabkan kesulitan melakukan penyimpanan data ke serial EEPROM yang lambat.

3.4. Memori Serial EEPROM

Selama proses download update program, data-data program harus disimpan ke memori tertentu yang bersifat permanen. Hal ini dikarenakan proses flashing (pemrograman flash ROM pada mikrokontroler) harus dilakukan oleh program bootloader (tidak dapat dilakukan oleh program aplikasi) yang akan dilakukan setelah proses restart dari mikrokontroler.

Memori yang digunakan harus bersifat permanen selama proses restart (bisa menggunakan RAM, asalkan dapat dipastikan data tidak hilang selama restart), memiliki ukuran sama dengan ukuran flash ROM (program yang akan di-download) dan bersifat cepat saat dilakukan penyimpanan data ke memori. Pada penelitian ini digunakan serial EEPROM dengan alasan mudah, murah dan kecil.



Gambar 7. Penulisan ke EEPROM

Jenis AVR yang digunakan pada penelitian ini adalah AVR ATmega32 dengan kapasitas flash ROM 32k Byte, sehingga diperlukan serial EEPROM jenis AT24C256 yang memiliki kapasitas 256k bit atau 32k Byte. Kecepatan penyimpanan dari EEPROM cukup rendah, sekitar 10 ms per penyimpanan. Kecepatan dapat ditingkatkan dengan melakukan penyimpanan untuk setiap satu halaman (1 page) yang terdiri dari 64 Byte.

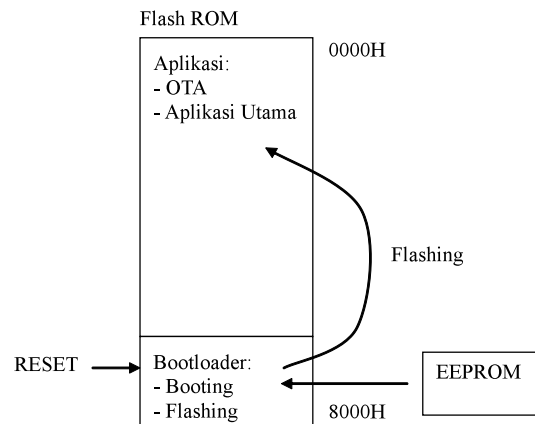
Kecepatan penyimpanan dapat dihitung berdasarkan kecepatan simpan dan kecepatan transfer untuk data 64 byte. Kecepatan transfer sekitar $4 \times 10 \text{ bit} \times t_{\text{clock}} \times 64$, yaitu sekitar 51,2 ms, dengan asumsi t_{clock} sekitar 20 us (dari perkiraan eksekusi program). 10 bit adalah panjang rata-rata untuk mengirimkan 1 byte komunikasi. Sedangkan 4 byte adalah total byte yang harus

dikirimkan untuk setiap 1 byte data yang akan disimpan. Sehingga waktu total penyimpanan untuk 64 byte sekitar 51,2 ms + 10 ms (untuk menulis ke EEPROM) atau 61,2 ms. Untuk tiap byte dapat diperkirakan sekitar 61,2 ms / 64 byte, yaitu sekitar 0,96 ms/byte atau 1045 byte/detik.

3.5. Bootloader

Suatu kontroler atau Prosesor secara umum tidak diperbolehkan melakukan update program secara langsung menggunakan program aplikasi itu sendiri. Hal ini dengan tujuan untuk menghindari update program yang tidak dikehendaki selama program berjalan yang dapat merusak program. Selain itu, tidak mungkin melakukan update program pada suatu program yang sedang berjalan.

Cara umum yang digunakan adalah dengan menggunakan program lain yang sifatnya bukan program utama dan program ini hanya berjalan saat pertama saja untuk keperluan tertentu. Program ini sering disebut sebagai bootloader.



Gambar 8. Bootloader

Salah satu fungsi umum dari bootloader adalah melakukan pemrograman flash ROM (flashing) untuk program aplikasi (program utama). Program bootloader itu sendiri tidak dapat deprogram ulang kecuali menggunakan alat pemrogram (programmer). Ukuran dari boot loader biasanya kecil, cukup untuk proses start-up (booting), membaca data dari EEPROM serial dan flashing. Contoh program flashing seperti pada gambar 9.

```

if (rdWord(0x0000) != 0xFFFF && rdWord(0x0002) != 0xFFFF)
{
    tunggu(1000);
    puts("\r\nFLASHING ... \r\n");
    alamat=0x0000;
    while(1) {
        data=rdWord(alamat);
        if (data==0xFFFF)
            if (rdWord(alamat+2) == 0xFFFF)
                break;
    }
}

```

```

wrFlash(alamat,data);
alamat+=2;
}
wrFlash(0xFFFF,0);
wrWord(0x0000,0xFFFF);
wrWord(0x0002,0xFFFF);
puts("FINISH\r\n");
}

```

Gambar 9. Contoh program flashing

4. Pengujian

Sistem update program menggunakan OTA ini dapat melibatkan banyak aspek yang perlu diuji coba, misalkan proses koneksi dari device ke server OTA, proses permintaan update ke server OTA, proses download dari server OTA ke device, buffering data update ke EEPROM, proses flashing ke mikrokontroler, penggunaan operator yang berbeda, dan lokasi penempatan device (di dalam atau di luar ruang, jarak terhadap BTS, dan sebagainya).

Pada paper ini hanya dibahas pengujian dari proses download sampai dengan buffering data update program ke EEPROM. Pemilihan pengujian ini karena berkaitan dengan keberhasilan proses download data update program.

Pengujian dilakukan dengan menggunakan data uji berupa program dalam bentuk file hex dengan ukuran data 11566 byte dan ukuran file 32589 byte (ukuran file hex berbeda dengan ukuran data asli). Ukuran ini mewakili 50% kapasitas flash memori dari mikrokontroler. Kartu operator yang digunakan adalah Tri. Perbandingan antara data byte dan ukuran file sekitar 2,8 byte file per 1 byte data.

Tabel 1. Pengujian berdasarkan kecepatan.

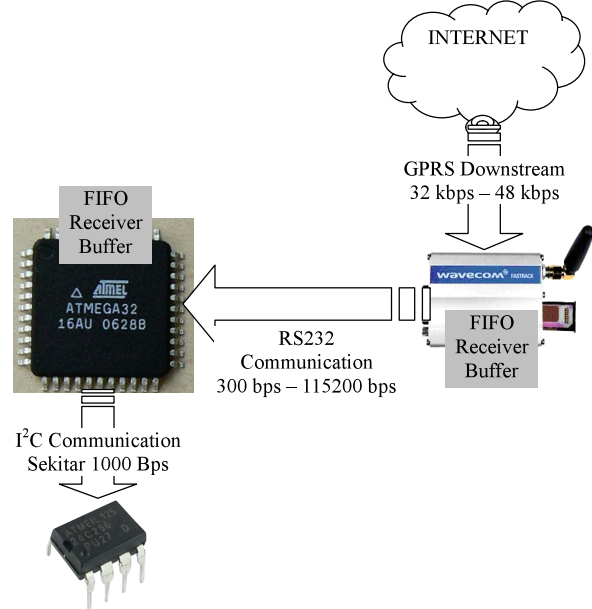
Kecepatan Serial (bps)	Buffer RAM (Byte)	Hasil
9600	-	MODEM ERROR ¹⁾
19200	-	MODEM ERROR ¹⁾
38400	178	OK ²⁾
57600	1042	MIKRO ERROR ³⁾
115200	17105	MIKRO ERROR ³⁾

- 1) Buffer receiver Modem overflow
- 2) Mekanisme OTA berjalan dengan baik
- 3) Buffer receiver Mikrokontroler overflow

Dengan asumsi kecepatan GPRS kelas 10 sekitar 32 kbps s/d 48 kbps, maka pada pengujian dengan kecepatan komunikasi serial lebih kecil dari 38400 bps menyebabkan data yang dikirimkan dari modem ke mikrokontroler mengalami banyak sekali data yang hilang (data tidak lengkap), disebabkan oleh buffer modem tidak dapat menampung semua data yang

terlambat dikirimkan ke mikrokontroler (MODEM ERROR¹⁾).

Dengan kecepatan lebih dari 38400 bps, modem tidak lagi mengalami masalah dengan data yang ditransfer yang dapat dilihat dari lengkapnya data yang ditransfer ke mikrokontroler. Namun Masalahnya justru pada proses penyimpanan data ke serial EEPROM.



Gambar 6. Kecepatan Komunikasi

Kecepatan simpan ke serial EEPROM dari uraian sebelumnya diperkirakan 1800 byte/detik. Jika 1 byte data identik dengan 2,8 byte file, maka kecepatan simpan serial EEPROM identik dengan kecepatan data file sekitar $1045 \times 2,8 = 2928$ byte data file per detik, atau sekitar 30 kbps.

Jika kecepatan komunikasi serial yang digunakan untuk transfer data file dari modem ke mikrokontroler lebih dari 30 kbps, maka mikrokontroler akan memerlukan buffer memori dengan ukuran tertentu agar data dari modem dapat disimpan dengan aman ke serial EEPROM. Dari hasil percobaan, ternyata pada kecepatan di atas 38400 bps, buffer memori yang diperlukan lebih dari sekian ratus byte (bahkan mencapai 1 kByte lebih pada kecepatan 57600). Ukuran memori ini tidak dapat dipenuhi jika menggunakan ATmega32 (dengan mudah dipenuhi jika menggunakan ATmega yang lebih besar) dengan RAM 1 kByte saja. Dari percobaan selalu didapatkan data yang berhasil disimpan ke EEPROM menjadi salah (MIKRO ERROR³⁾). Dengan demikian kecepatan yang sesuai hanya sekitar 38400 bps saja, batas dari modem dan mikrokontroler.

Hasil pengujian tersebut tentu saja sangat bergantung dengan jenis data pakatnya (CSD, GPRS, MTS, HSDPA, HSUPA atau lainnya), tergantung merek atau jenis modem (dalam hal ini besarnya buffer receiver

modem), jenis mikrokontroler (kapasitas RAM yang dapat digunakan sebagai buffer receiver) dan jenis IC buffer data update program (jenis EEPROM, RAM atau sejenisnya akan mempengaruhi kecepatan penyimpanan data).

Untuk mendapatkan kecepatan komunikasi dari modem ke mikro pada kecepatan yang lebih tinggi dapat dicapai dengan mengganti mikrokontroler dengan jenis yang lebih besar yang memiliki RAM untuk buffer serial yang lebih besar. Dapat juga dilakukan dengan mengganti EEPROM menjadi memori dengan kecepatan penyimpanan yang lebih tinggi. Misalkan menggunakan NFRAM atau EEPROM dengan page buffer yang lebih besar (EEPROM dengan kapasitas yang lebih besar).

4. Kesimpulan

Pada perangkat yang berbasis mikrokontroler, update secara OTA memerlukan perhatian khusus, terutama pada masalah:

- Kecepatan downstream dari paket Internet
- Kecepatan komunikasi serial dari modem ke mikrokontroler
- Kapasitas buffer memory dan kecepatan penyimpanan data ke memory

Dari hasil percobaan, dengan kecepatan internet sekitar 33 kbps s/d 48 bps, diperlukan kecepatan komunikasi serial 38400 bps untuk menyimpan pada serial EEPROM 32 kByte.

Daftar Pustaka

- [1] Gascón, David; Bielsa, Alberto; Genicio, Félix; Yarza, Marcos, "Over the Air Programming with 802.15.4 and ZigBee – OTA", Retrieved 2012
- [2] ____, "Over the Air Programming with Waspote-OTA", Libelium.com. Retrieved 2012-02-02.
- [3] ____, Alaska DigiTel Buys OTA Programming Solution from Interop Technologies, Tmcnet.com. 2008-04-01. Retrieved 2012-02-02.
- [4] Zetter, "Over the air programming", <http://www.avrfreaks.net/index.php?name=PNphpB2&file=viewtopic&t=105935#819295>, Apr 20, 2011
- [5] Michael F. Oryl Jr., "What is OTA", www.mobileburn.com, Retrieved 2012
- [6] ____, "Over-the-air updates for smartphones", uk.blackberry.com, 2012, Retrieved 2012
- [7] ____, "OTA Firmware Update", msdn.microsoft.com, 2010, Retrieved 2012
- [8] ____, "Over-the-air programming", en.wikipedia.org, 2012, Retrieved 2012