Implementation of the CANopen Profile for Battery and Charger

Michel Passemard Atmel, Bob Pickering Minit-Charger, Olaf Pfeiffer Embedded Systems Academy

The growing acceptance of opportunity charging batteries while they remain in the electric vehicle made the development of a standard means of communicating the battery's characteristics to the charger increasingly important.

This paper gives some background information on why such a standard is desirable, and how CANopen provides the necessary framework for the standard. This paper also presents an implementation of the CANopen battery profile highlighting the battery module as well as the charger module.

Introduction

The development of fast charge algorithms has made it practical to return a significant amount of energy to the battery of an electric vehicle in a relatively short period of time, allowing vehicles to be recharged during breaks in activity. Whereas in a conventional charge regime, each vehicle would have multiple batteries and a dedicated charger, fast charge allows a single battery which remains in the vehicle to be recharged as opportunities arise. With the charger no longer dedicated to a single vehicle, it must be able to charge any of the vehicles in the facility. A typical facility has a variety of vehicle types, with batteries of differing voltages and capacities, and possibly different chemistries. The charger must be able to identify the battery in the vehicle to which it is connected in order to appropriately adjust the parameters of the charge algorithm. Battery charger manufacturers have developed a number of proprietary battery modules which communicate the necessary information to their chargers. These solutions, however, restrict the customer to using the modules only with the chargers from the same manufacturer.

With the growing acceptance of opportunity charging both for industrial fork lifts, and for airport ground support equipment, a need was identified for a standard protocol for the communication between the battery module and the charger to allow chargers from different manufacturers to operate with all of the vehicles in a facility. The National Electric Vehicle Infrastructure Working Council (IWC) of the Electric Power Research Institute (EPRI) (Palo Alto, CA) has a mission to coordinate, and collaborate on charging infrastructure issues involving electric transportation, and to make recommendations to standards-making bodies. It struck a committee to develop a recommendation for the battery/charger protocol. The committee, which had representatives from electric utilities, charger manufacturers, battery manufacturers, vehicle manufacturers and users of ground support equipment, first determined the data which needed to be passed between the battery and the charger to allow the charge to take place. It then investigated a variety of alternative communication standards, and decided that CANopen provided the necessary framework on which the protocol could be defined.

CANopen is a widely accepted CAN protocol stack. It offers a framework for multiple application usages. While it could have been seen as a European standard, it became popular in Asia and The Americas too. One of the key benefits of CANopen is the concept of profiles. System manufacturers from the same application environment team-up to develop an agreed upon profile that will be used in their systems, thus insuring inter-operability. Semiconductor vendors can propose standard products that have been proven with such profiles. Furthermore they can offer themselves or through Software vendors a protocol stack that supports the profiles.

This was viewed as the key benefit of adopting CANopen for the battery/charger interface; instead of developing an entire communications protocol, it was only necessary to specify the device profile. The ease with which manufacturer specific data could be incorporated was also important, since none of the manufacturers involved were prepared to divulge all of their trade secrets so that they could be included in the profile. The fact that CANopen is an open standard was also an important consideration.

Given below is a summary of the Device Profiles for Battery Modules, and Chargers, and a description of an implementation in a Minit-Charger battery charger system, using an Atmel microcontroller.

Summary of the CANopen Device Profile for Battery Modules

The CANopen draft proposal 418: Device Profile for Battery Modules defines the communication link between a charger and a battery. It also defines all the necessary information to carry out battery charge.

The profile defines an object dictionary for the battery. This object dictionary includes 21 indexes for a total of 49 entries.

The indexes are:

Standard Mandatory entries
- Device Type
- Error register (temperature sensor)
- Identity object

Profile 418_d

The profile entries are divided into mandatory and optional elements.

Mandatory entries
- Battery Status
- Charger status
- Temperature
- Battery parameters (type, capacity, max charge current, number of cells)

Optional Entries
- Battery Serial Number
- Battery Identification Number
- Vehicle Serial Number
- Vehicle Identification Number
- Cumulative total Ah charge
- Ah expended since last charge
- Ah returned during last charge
- Ah since last equalization
- Date of last equalization
- Battery voltage
- Charge current requested
- Charger state of charge
- Battery state of charge
- Water level status

Mandatory entries are the minimum data that is required to allow the charge to be carried out. These specify the battery chemistry, number of cells, and the maximum current which can be carried by the battery cables and intercell connectors. Since all useful fast charge algorithms use some form of temperature compensation, the battery temperature is also a mandatory data element. The mandatory entries were kept to the bare minimum to allow battery module manufacturers the greatest possible flexibility in choosing hardware for implementation.

Battery chemistry is specified by the battery type parameter. The profile defines the following battery types:

- Lead acid battery (flooded and maintenance free)
- Nickel cadmium battery
- Nickel zinc battery
- Nickel Iron battery
- Silver oxide battery
- Nickel hydrogen battery
- Nickel metal hydride battery
- Zinc/Alkaline/Manganese battery

- Lithium ion battery
- Zinc bromine battery
- Metal air battery
- Lithium/Iron sulfide battery
- Sodium beta battery

The optional data entries may be used to support common enhancements to the battery charging process. The serial and identification numbers allow fleet tracking systems to record battery usage and status. Data entries cumulative total Ah charge, Ah since last equalization, and date of last equalization record battery history, which may be necessary to maintain the battery manufacturer's warranty. The charger updates the Ah returned during last charge value as the charge progresses, allowing the battery module to maintain the battery history without requiring it to measure current flow. More sophisticated battery modules may provide battery voltage measurements, and estimates or measurements of the Ah expended since last charge. If the battery module includes charge algorithm software, it may request the level of charge current desired via the charge current requested data entry. Two state of charge entries are defined; one supplied by the charger, and one by the battery module. This allows the unit with the best data to establish the state of charge (for example, the battery module might estimate the state of charge during discharge, while the charger sets it during charge based on the progress through the charge algorithm). Finally, a water level status data entry is defined to allow modules on flooded batteries to pass the status to the charger where warning indications may be displayed or other action taken when the battery needs maintenance.

The profile defines one Transmit Process Data Object (TPDO) and one Receive Process Data Object (RPDO) to transmit battery status, and temperature, and receive charger status.

The profile also defines 2 additional TPDO and 2 additional RPDO to carry optional information.

The profile's object dictionary requires only Expedited Service Data Object (Expedited SDO) for download and upload.

The SDOs are

- Initiate SDO download (charger request download from battery)
- Battery respond with the data to the charger
- Initiate SDO upload (charger upload to battery)
- Acknowledge response from battery

The CANopen draft proposal 419: Device Profile for Battery Charger, provides the counterpart to the battery module profile for the charger. PDOs mirror the PDOs of the battery module profile. The data dictionary defines only those entries which are used in the PDOs. The use of other information from the battery module is left up to the implementer.

Mapping the CANopen Device Profile for battery into the Atmel T89C51CC01

The Atmel Flash + CAN microcontroller is perfectly suited for the CANopen profile for battery and charger, with ample memory and processing power to accommodate the optional entries as well as the mandatory ones.

32Kbytes of flash contains the program. It can be reprogrammed up to 100K times. 32KB is large enough to contain the CANopen profile plus a good size application.

Only 30% of the extended RAM (1024 bytes) will be used to keep the object dictionary leaving a large amount of RAM for the user application; pointers, stack and other variables being located in the 256 bytes regular RAM.

2Kbytes of E2PROM are available to maintain information such as battery S/N type etc... The E2PROM can be written by bytes or block from 2 bytes to 128 bytes in 11mS per cycle. The controller can operate down to 3 volts, thus a simple supply monitoring device can be designed to sustain Vcc >= 3 volts for enough time to save 128 bytes or more into the E2PROM when the power disappear.

The T89C51CC01 CAN peripheral architecture with the unique Message Object Buffer (MOB) is perfectly suited for the CANopen battery profile.

The chip sports 15 Message Object Buffers with each of them programmable as Transmitter or Receiver. Each Message Object has an 8 byte dedicated message data FIFO, a dedicated Message ID and Message Mask (for receiver).

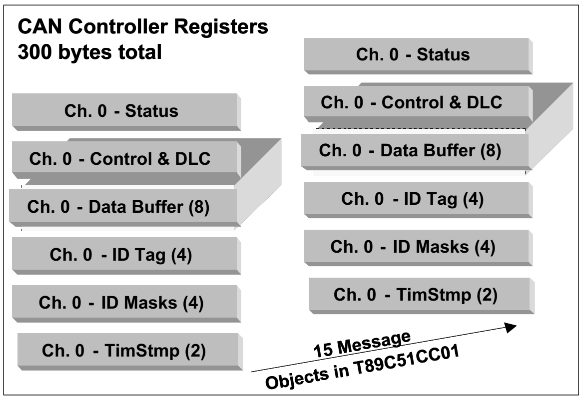
Basic implementation of the CANopen battery profile will use 9 Message Object Buffers:

- NMT (ID=000h)
- Sync (ID=080h)
- Emergency (ID=81f to FFh)
- Time Stamp (ID=100h)
- TPDO (ID=181h to 186h)
- RPDO (ID=201h to 203h)
- TSDO (ID=581h)
- RSDO (ID=601h)
- NMT (ID=701h to 77Fh)

Implementation of the CANopen battery profile with optional TPDOs RPDOs will use 12 Message Object Buffers:

- NMT (ID=000h)
- Sync (ID=080h)
- Emergency (ID=81f to FFh)
- Time Stamp (ID=100h)
- 1^{st} TPDO (ID=181h to 186h) could be replaced by 2^{nd} TPDO (ID=281h to 287h)
- 3^{rd} optional TPDO (ID=381h to 387h)
- 1^{st} RPDO (ID=201h to 203h)
- 2^{nd} optional RPDO (ID=301h to 303h)
- 3^{rd} optional RPDO (ID=401h to 403h)
- TSDO (ID=581h)
- RSDO (ID=601h)
- NMT (ID=701h to 77Fh)

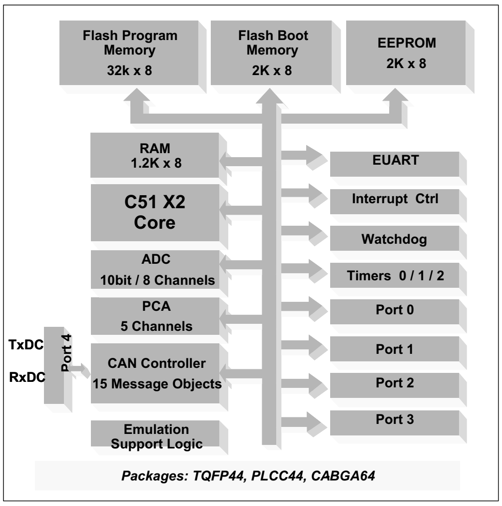
The Message Object Buffer structure is given in figure 2 below



The controller also includes a 16-bit time stamp timer. Each transmit messages and receive messages can be time stamped. This feature can be used for instance to synchronize clock between a charger and a battery

An Analog/Digital converter is necessary in the battery module to measure the temperature. This can be done using the 10-bit precision A/D included in the T89C51CC01. The temperature measurement is then sent with the 1^{st} TPDO. The A/D includes 7 more channels to convert other signals.

The block diagram of the T89C51CC01 is shown on the figure below



Adding the CANopen bootloader to the Device Profile for battery into the Atmel T89C51CC01

The Atmel Flash + CAN microcontroller has a 2KB flash area for the bootloader. It comes factory preprogrammed with a CAN based or a UART based bootloader.

It is possible to include with the Device Profile for battery, the CANopen custom bootloader such as the Embedded System Academy CANopen bootloader.

A CANopen Compliant Bootloader

If a CANopen device has flash memory, it makes sense to also have a CANopen compliant bootloader. This does not only free the serial communication channel from the bootloader task, it also allows using standard CANopen configuration tools as the communication partner providing the new code (hex file) to be loaded into the Flash memory.

When being in boot-mode a CANopen node's only purpose is to accept a hex-file to be loaded into a Flash memory. While being in that mode, the node does not really need to be 100% CANopen compliant. It just needs to provide enough CANopen compatibility that it does not interfere with any other communication on the network and that it provides a fully functional SDO server, so that SDO clients (like Masters, Managers or Configuration Tools) can make read and write accesses to the Object Dictionary in the node.

So the only CANopen features and communication channels that need to be implemented are the SDO server and the SDO request and response channels.

Sometimes it is desirable that the bootloader can be activated without the requirement to physically touch the device (like setting a jumper, switch or button). In CANopen the straight forward method would be to use a selected write sequence to an Object Dictionary entry as an additional command to actually switch the node into the bootloader mode.

Minimal Set of Required OD Entries

OD entry [1000,00]: Device type information, read-only

As there is no device type number standardized for a bootloader, a manufacturer specific value can be used. The Embedded Systems Academy uses 746F6F62h (ASCII representation is “boot”) in their bootloader implementations.

OD entry [1001,00]: Error register, read-only

The bootloader can use this register to signal Flash erase or programming failures. Setting the manufacturer specific error bit can indicate an out-of-range error – a try to program a memory location that is either protected or at which there is no Flash memory.

OD entry [1018,00-04]: Identify Object

The standard Identify Object as specified in CANopen [CiADS301].

OD entry [1F50,XX]: Download program data

This Object Dictionary entry is described in [CiADSP302] and used to directly accept the code programmed into the target memory. Sub-index 0 is used to quantify how many different program or Flash memory areas are available. The following Sub-indexes can each handle the download to one program or memory area. For many applications it is sufficient to implement one area (Sub-index 1).

Although not specified by the standard, the Embedded Systems Academy recommends using standard ASCII hex files as the files containing the program or data. Using a hex file has two benefits: the file contains the target address on where the data needs to be programmed to and the file also contains checksums making the downloading process more secure.

Although the T89C51CC01 flash memory doesn't need to be erased first before being reprogrammed, Embedded System Academy has implemented a specific erase command. For example, an erase could be initiated by sending the value 66726C63h (ASCII representation is "clrf") to the Object Dictionary entry [1F50,01] (or other Sub-indexes to differentiate between different blocks or segments of Flash memory).

OD entry [1F51,XX]: Program Control Object

This Object Dictionary entry is described in [CiADSP302] and used to control the program(s) downloaded to [1F50,XX]. The essential command to implement is "Start program" which requires writing a "1" to the Object Dictionary entry. For example, if a program was downloaded to [1F50,01] it can be started by writing a "1" to [1F51,01].

This Object Dictionary entry could also be used to implement the activation of the bootloader itself. So if the regular CANopen application running on this node supports this entry, it should activate the bootloader upon receiving a "0" (zero =Stop Program).

CANopen Bootloader for 89C51CC0x

The Embedded Systems Academy implemented a CANopen bootloader for the Atmel 89C51CC01 microcontroller family. Code examples can be downloaded from www.esacademy.com/atmel

Michel Passemard
Atmel
BP70602 Nantes 44306 France
Phone +33 2 40 18 19 65
Fax +33 2 40 18 19 60
E-mail michel.passemard@atmel.com
Website www.atmel.com

Bob Pickering
Edison Minit-Charger
2486 Dunwin Drive
Mississauga, Ontario, Canada L5L 1J9
Phone (905)828-7700
Fax (905)828-7716
E-mail bpickering@minit-charger.com
Website www.minit-charger.com

Olaf Pfeiffer
Embedded Systems Academy
50 Airport Parkway
San Jose, CA 95110
(408) 910-7899
opfeiffer@esacademy.com
www.MicroCANopen.com