Features

- USB-CAN gateway
  - USB CDC class (virtual serial port) provides low level data stream
  - Customized frame-based USB-CAN communication protocol
- Two–node CAN network on one SAM4E-EK
  - CAN0 node connected to PC utility tool with USB-CAN communication
  - CAN1 node data flow displayed on the on-board touch screen
- PC utility tool monitors CAN bus/node status through CAN0 node
- The touch screen acts as data sink to CAN1 node

Description

SAM4E provides two CAN controllers; CAN0 and CAN1, which are fully compliant with CAN 2.0A and 2.0B specification. This demo is developed on SAM4E-EK, using both CAN controllers to establish a simple two-node CAN network on one SAM4E-EK board.

SAM4E-EK acts as the USB-CAN gateway, CAN0 is connected to CAN1 and communicates in CAN protocol and the data transferred/received in CAN0 will be monitored by USB-CAN PC Utility through USB on SAM4E with CDC class (virtual serial port). The PC Utility works together with SAM4E-EK to act as a CAN-bus monitor.

The data transfer in CAN1 is displayed and controlled by the simple UI with the on-board LCD touch screen.

Figure 1. USB-CAN Demo Functional Diagram.
# Table of Contents

1. Overview .................................................................................................................. 3  
   1.1 USB-CAN Firmware ............................................................................................. 3  
   1.1.1 USB-CAN Conversion ................................................................................. 3  
   1.1.2 CAN0 Node .................................................................................................. 3  
   1.1.3 CAN1 Node .................................................................................................. 3  
   1.2 USB-CAN PC Utility .......................................................................................... 3  
2. Demo Setup Procedure ............................................................................................ 6  
   2.1 Programming Procedure ................................................................................... 6  
   2.2 Demo Setup Procedure ..................................................................................... 6  
3. USB-CAN Communication Protocol ......................................................................... 7  
4. Revision History ....................................................................................................... 9
1. **Overview**

This USB-CAN demo implements a simple USB-CAN gateway and CAN-bus monitor based on SAM4E-EK. A simple two-node CAN network is established to demonstrate how the SAM4E CAN module works. Some simple CAN-bus monitor and control functions are provided by this demo. It can also be used as a PC side debugging tool which will ease CAN-bus based application development.

In this USB-CAN demo, both USB-CAN firmware and USB-CAN PC Utility are provided.

The firmware implements USB-CAN data conversion, CAN0 node, CAN1 node, data sink on the touch screen (with CAN1 node).

The PC Utility implements USB-CAN control/configuration, data sending and status monitor.

1.1 **USB-CAN Firmware**

The USB-CAN firmware is developed based on Atmel® Software Framework (ASF), using USB stack with CDC class, CAN driver, LCD driver, Touch Screen driver etc. The firmware includes USB-CAN data conversion, CAN0 node, CAN1 node (including data sink on the touch screen).

1.1.1 **USB-CAN Conversion**

USB-CAN conversion is based on SAM4E USB2.0 Full-Speed USB device controller.

A simple customized USB-CAN communication protocol is designed based on the stream data provided by USB CDC class (virtual serial port). About the USB-CAN communication protocol, refer to Chapter 3 USB-CAN Communication Protocol. The transmission speed (baud rate) is mostly determined by USB2.0 Full-Speed.

1.1.2 **CAN0 Node**

CAN0 node uses CAN0 controller on SAM4E. The node provides the following functions:

- Respond to the command from the PC Utility
- CAN standard frame (CAN 2.0A) transmit and receive
- CAN extended frame (CAN 2.0B) transmit and receive
- CAN0 node and CAN bus error status report

The data transmit on CAN0 node is launched by the PC Utility. The message ID and data are configured by PC Utility. If CAN0 node receives data from CAN1, it reports the received data from CAN1 to the PC Utility.

1.1.3 **CAN1 Node**

CAN1 node uses CAN1 controller on SAM4E. The node provides the following functions:

- CAN standard frame (CAN 2.0A) transmit and receive
- CAN extended frame (CAN 2.0B) transmit and receive

The data transmit on CAN1 node is launched by the UI on the on-board LCD touch screen of SAM4E-EK.

When the button “STD FRM Tx” or “EXT FRM Tx” button is pressed on the screen, the corresponding standard or extended frame will be sent by CAN1. The message ID and data are pre-determined in SAM4E firmware.

When CAN1 node receives data, the receive counter will be updated on the LCD screen.

1.2 **USB-CAN PC Utility**

The PC software runs in Windows® OS with Microsoft® .NET framework 3.5 or up.

The PC software includes the following functions:

- USB CDC communication with SAM4E-EK
• Start/Stop control of SAM4E CAN0 node
• CAN bus communication configuration: baud rate, receive filter (SAM4E CAN0 node configuration)
• CAN data send (through SAM4E CAN0 node): single send, repeated send
• Monitor CAN bus connected to SAM4E CAN0

Figure 1-1. USB-CAN PC Utility.

The tool bar in top lists basic configuration for the USB-CAN demo: COM Port for USB virtual serial port (CommPort), CAN baud rate, and CAN0 node receive filter ID.

• For the “CommPort”, the user should select the COM No. corresponding to the USB CDC virtual serial port. The COM No. can be found in “device manager” in Window OS after PC driver is successfully installed for the SAM4E-EK programmed with USB-CAN firmware
• The CAN baud rate has the unit of kbps, represented by the value with “K” or “k” ending. The Default value is 1000k
• For receive filter, the default is 0x00000000, which can receive any message on the CAN bus

In “Send Data” area,
• Frame Type includes Data Frame and Remote Frame for CAN0 node transmit. In current version, only Data Frame is supported
• Frame Format includes Standard Frame and Extended Frame
• Message ID is the message identifier, 32bit hex unsigned integer. For Standard Frame, MessageID is 11 bits, located at bit28-bit18; for extended frame, MesageID is 29-bits, located at bit28-bit0. Message ID is used in message filtering process by receivers, to determine whether the message should be received
• For single send, set “Send Times” to 1 and “Send Interval” to 0. For repeated send, set “Send Times” and “Send Interval” to the expected times and interval
• In Data textbox, the data that will be sent are displayed here. The data should be in hex format with space between bytes as separator. The send data length is fixed to 8, which is also the maximum possible data length for a CAN frame. If the data count in the Data textbox is less than 8, “00” will be appendixes to make 8 bytes
The list view lists all CAN data transaction through USB-CAN. The information includes transmit/receive (T/R), time stamp, message ID, data, frame type, frame format etc. If the data transmit fails, the corresponding failing data line shows in red and the corresponding CAN bus error label on the left also shows in red.

The CAN bus error includes CRC error, Bit-stuffing error, Bit error, Form error, Acknowledgement error. When error is detected, the corresponding label is highlighted in red.

The CAN0 node error mode including Error Active (ERRA), Error Passive (ERRP), WARN, Bus Off (BOFF), located below the CAN bus error labels. The present mode of CAN0 mode is highlighted in red.
2. **Demo Setup Procedure**

This chapter describes how to program the binary file without recompile the source code and basic operation procedure with the demo.

2.1 **Programming Procedure**

1. Update Segger driver to 4.58. (If the Segger driver version is 4.58 or higher, this step can be over passed.)
2. Install SAM-BA® 2.12 or above. Install SAM-BA 2.12 patchsam4e. Copy JLinkARM.dll in the SAM-BA directory.
   - Win 7 C:\Program Files (x86)\Atmel\sam-ba_2.12\drv
   - Win XP C:\Program Files\Atmel\sam-ba_2.12\drv
3. Erase chip using jumper JP7 (labeled ERASE) for 15 seconds:
   a. Unplug the SAM-ICE™.
   b. Unplug the board power cable.
   d. Plug the board power cable.
   e. Wait 15s.
   f. Unplug the board power cable.
   g. Remove jumper JP7.
4. Connect a SAM-ICE between PC and SAM4E-EK.
5. Plug the board power cable.
6. Open a DOS Shell and run prog_flash.bat.
7. After programming, unplug the board power cable.

2.2 **Demo Setup Procedure**

1. Connect CAN0 and CAN1 connectors (J13, J14) by a RJ21 cable.
2. Connect USB cable between J11 and a PC USB port.
3. Power up the SAM4E-EK.
4. Install PC driver for SAM4E-EK as a USB CDC device (virtual serial port).
5. Find the COM No. of the virtual serial port in “Device Manager” of Windows OS.
6. Open the USB_CAN_Utility_vx.x.exe.
7. Select the right COM port corresponding to the USB virtual serial port.
8. Select or input CAN bus baud rate.
9. Press confirm button.
10. Press Start button to enable CAN node.
11. Press Send button to send a frame with default frame property and frame data.
12. Check the LCD to see that receive counter increases and received data shows up.
13. Press “EXT FRM Tx” button in the touch LCD to send an extended frame.
14. Check the PC Utility to see an extended frame is received.
3. **USB-CAN Communication Protocol**

USB-CAN demo uses USB CDC class (virtual serial port) to communicate between PC utility and SAM4E-EK. Based on USB CDC data stream, the communication frame format is designed for data transfer and interpreting. The communication is duplex: PC can send commands/data to EK and vice versa.

**Figure 3-1. USB-CAN Communication Frame Format.**

<table>
<thead>
<tr>
<th>F_ID</th>
<th>CMD</th>
<th>Data0</th>
<th>...</th>
<th>DataN</th>
<th>CRC</th>
</tr>
</thead>
</table>

F_ID: Frame Identifier, 1 byte. It has fixed value, 0xFF.

CMD: Command code, 1 byte. Its MSB bit (bit 7) stands for transfer direction: 0 means PC to SAM4E-EK; 1 means SAM4E-EK to PC. The command set is listed in later section.

Data0 – N: Data area, (N+1) bytes. For different command, the N is possibly different. The parameters necessary to the command can be placed in this data area. In order to simplify the SAM4E command parsing process, the commands from PC to SAM4E-EK have the same data length. Note the (N+1)-byte data length does not includes F_ID, CMD and CRC field.

CRC: CRC8 from F_ID to Data7. CRC8 generating polynomial is X^8+X^5+X^4+1. CRC check is not implemented in the current version.

**Table 3-1. USB-CAN Command Set.**

<table>
<thead>
<tr>
<th>Name</th>
<th>CMD Code</th>
<th>Data Length (N+1)*</th>
<th>Data</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Start CAN</td>
<td>0x01</td>
<td>13</td>
<td>Data0-1: baud rate</td>
<td>PC -&gt; SAM4E-EK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data0: High byte</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data1: Low byte</td>
<td></td>
</tr>
<tr>
<td>2 Stop CAN</td>
<td>0x02</td>
<td>13</td>
<td></td>
<td>PC -&gt; SAM4E-EK</td>
</tr>
<tr>
<td>3 Baud Rate setting</td>
<td>0x03</td>
<td>13</td>
<td>Data0-4: Message ID (high byte to low byte)</td>
<td>PC -&gt; SAM4E-EK</td>
</tr>
<tr>
<td>4 Receive Filter ID setting</td>
<td>0x04</td>
<td>13</td>
<td>Data0-[bit7]: frame format 0 standard frame, 1 extended frame</td>
<td>PC -&gt; SAM4E-EK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data0-[bit6]: frame type 0 data frame,(transmit mailbox) 1 remote frame (producer mailbox)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data0- [bit5-0],Data1-3: Message ID Data4-11: CAN send data</td>
<td></td>
</tr>
<tr>
<td>5 Send data</td>
<td>0x05</td>
<td>13</td>
<td></td>
<td>PC -&gt; SAM4E-EK</td>
</tr>
<tr>
<td>6 Status request</td>
<td>0x06</td>
<td>13</td>
<td></td>
<td>PC -&gt; SAM4E-EK</td>
</tr>
<tr>
<td>7 Status data report</td>
<td>0x81</td>
<td>12</td>
<td>Data0-3: active mailbox status (CAN_MSR register) Data4-7: CAN0 status code (CAN_SR register) Data8-11: Error counter (CAN_ECR register)</td>
<td>SAM4E-EK -&gt; PC</td>
</tr>
<tr>
<td>8 Data sent notification</td>
<td>0x82</td>
<td>12</td>
<td>Data0-3: active mailbox status (CAN_MSR register) Data4-7: CAN0 status code (CAN_SR register) Data8-11: Error counter (CAN_ECR register)</td>
<td>SAM4E-EK -&gt; PC</td>
</tr>
<tr>
<td>Name</td>
<td>CMD Code</td>
<td>Data Length (N+1)*</td>
<td>Data</td>
<td>Comment</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------</td>
<td>-------------------</td>
<td>----------------------------------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Data received report</td>
<td>0x83</td>
<td>14</td>
<td>Data0-[bit7]: frame format</td>
<td>SAM4E-EK -&gt; PC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 standard frame, 1 extended frame</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data0-[bit6]: frame type</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 data frame (transmitt mailbox), 1 remote frame (producer mailbox)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data0-[bit5-0],Data1-3: Message ID</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data4-11: CAN received data</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Data12-13: Time Stamp</td>
<td></td>
</tr>
</tbody>
</table>

Note: *The data length does not include F_ID, CMD and CRC.
## 4. Revision History

<table>
<thead>
<tr>
<th>Doc. Rev.</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>32208A</td>
<td>07/2013</td>
<td>Initial document release</td>
</tr>
</tbody>
</table>