**Scope**

The Atmel® | SMART SAMA5D3/D4 series are high-performance, power-efficient embedded MPUs based on the ARM® Cortex®-A5 processor.

This application note takes SAMA5D3 as an example to explain how to implement it as an Ethernet bridge in a PC network to connect two or more LANs, using EMAC/GMAC ports.

**Reference Documents**

<table>
<thead>
<tr>
<th>Title</th>
<th>Literature No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMA5D3 Series Datasheet</td>
<td>11121</td>
</tr>
<tr>
<td>SAMA5D4 Series Datasheet</td>
<td>11238</td>
</tr>
<tr>
<td>SAMA5D3x-EK User Guide</td>
<td>11180</td>
</tr>
<tr>
<td>SAMA5D4-EK User Guide</td>
<td>11294</td>
</tr>
</tbody>
</table>
# Table of Contents

Scope .................................................................................................................. 1

Reference Documents ......................................................................................... 1

Table of Contents ............................................................................................... 2

1. Introduction ..................................................................................................... 3

2. SAMA5D3/D4 Ethernet Port Overview ......................................................... 3

3. Ethernet Bridge Description .......................................................................... 4

4. Software Ethernet Bridge Implementation ..................................................... 5
   4.1 Hardware Connection ................................................................................ 5
   4.2 Software Implementation ......................................................................... 5
      4.2.1 How to Build Linux .......................................................................... 5
      4.2.2 How to Set PC and Bridge .............................................................. 11
      4.2.3 How to Run the Bridge ................................................................. 12
         4.2.3.1 Bridge Acts in Full Duplex Mode (EMAC ↔ GMAC) .......... 12
         4.2.3.2 Bridge Acts in Simplex Mode (GMAC → EMAC) .......... 13
         4.2.3.3 Bridge Acts in Simplex Mode (EMAC → GMAC) .......... 13
      4.2.4 Bandwidth and CPU Usage of Ethernet Bridge ............................. 14

5. Summary ....................................................................................................... 15

Revision History ................................................................................................. 16
1. **Introduction**

The SAMA5D3 connectivity peripherals include one Gigabit Ethernet Media Access Controller (GMAC) and one Ethernet Media Access Controller (EMAC) \(^1\). The SAMA5D4 connectivity peripherals include two 10/100 Ethernet Media Access Controller (GMAC). With these two MACs, SAMA5D3/D4 can be implemented in different network applications.

This application note provides an example of how to set SAMA5D3 as a software Ethernet bridge.

2. **SAMA5D3/D4 Ethernet Port Overview**

The EMAC module of SAMA5D3 implements a 10/100 Mbps Ethernet MAC compatible with the IEEE 802.3 standard using an address checker, statistics and control registers, receive and transmit blocks, and a DMA interface. This EMAC only supports RMII interface to the physical layer and can work in full-duplex and half-duplex mode.

The GMAC module of SAMA5D3 implements a 10/100/1000 Mbps Ethernet MAC compatible with the IEEE 802.3 standard. The GMAC can operate in either half or full duplex mode at all supported speeds. This GMAC can be configured as MII, GMII and RGMII interface.

The GMAC module of SAMA5D4 implements a 10/100 Mbps Ethernet MAC compatible with the IEEE 802.3 standard. The GMAC can operate in either half or full duplex mode at all supported speeds. This GMAC can be configured as MII and RMII interface.

For the detailed information on EMAC and GMAC of the SAMA5D3/D4 series, please refer to the SAMA5D3 Series datasheet and SAMA5D4 Series datasheet available on [www.atmel.com](http://www.atmel.com).

---

\(^1\) The number of available EMAC/GMAC peripherals is device specific. Please refer to the SAMA5D3 Series datasheet for details.
3. **Ethernet Bridge Description**

An Ethernet bridge is used to interconnect LANs using IEEE 802 standards. The standard features of bridges are defined by IEEE 802.1. There are four types of network-bridging technologies:

- Simple
- Multiport
- Learning or Transparent
- Source route

Simple bridge and Multiport bridge are the basic bridges which have ports connected to two (or more) otherwise separate LANs. Packets received on one port may be retransmitted on another port. Unlike a repeater, a bridge will not start retransmission until it has received the complete packet. Consequently, stations on either side of a bridge may be transmitting simultaneously without causing collisions. Bridges, like repeaters, do not modify the contents of a packet in any way. However, unlike repeaters they may, under some circumstances, originate traffic.

A learning bridge examines the source field of every packet it sees on each port and builds up a picture of which addresses are connected to which ports. This means that it will not retransmit a packet if it knows that the destination address is connected to the same port as the bridge saw the packet on.

Source route bridging is used on token ring networks. It is a hybrid of source routing and transparent bridging, which allows source routing and transparent bridging to coexist on the same bridged network. The operation of the bridge is simpler and many of the bridging functions are performed by the end systems, particularly the sources.

Note: The description of Ethernet Bridge types is extracted from http://www.wikipedia.org/. Please refer to it for more details.

*Figure 3-1* illustrates how an Ethernet bridge connects two LANs.

*Figure 3-1. Ethernet Bridge Diagram*
4. **Software Ethernet Bridge Implementation**

This section explains how to implement a simple bridge with SAMA5D3 by software.

4.1 **Hardware Connection**

In this example, a SAMA5D35 Evaluation Kit (SAMA5D35-EK) serves as the bridge, which is connected between two PCs (see **Figure 4-1**). (Currently, the network cards in most PCs are Gigabit, which can auto-negotiate with EMAC and GMAC according to the type of Ethernet port they connect to).

**Figure 4-1. Connection Diagram Between SAMA5D35-EK and PC**

4.2 **Software Implementation**

After the hardware connection is done, configure each part to make sure that they all work in correct modes.

4.2.1 **How to Build Linux**

1. Rebuild Linux kernel to make sure the Ethernet Bridge function is enabled.\(^1\)

```bash
//Get the source code from server
$ git clone git://github.com/linux4sam/linux-at91.git
Cloning into 'linux-at91'...
remote: Counting objects: 3875625, done.
remote: Compressing objects: 100% (624675/624675), done.
remote: Total 3875625 (delta 3220631), reused 3875625 (delta 3220631)
Resolving deltas: 100% (3220631/3220631), done.
$ cd linux-at91/
//Checkout from the github repository
$ git checkout origin/linux-3.10-at91 -b <your branch>
//Configure and build the Linux kernel
$ export CROSS_COMPILE=arm-linux-gnueabi-
$ export ARCH=arm
$ make clean
$ make sama5_defconfig
$ make menuconfig
//In the following interface, select the needed components to make sure
//the network driver is included.
```

---

1. Both ETH0 and ETH1 need to be enabled for the Ethernet Bridge function. However, for SAMA5D4, ETH1 is disabled by default, due to pin conflict with LCD power.

To enable ETH1:

* Before rebuilding Linux kernel, first disable LCD node and enable ETH1 node in the corresponding dts file.
* Close JP2 jumper on SAMA5D4-EK board to disable LCD power and enable ETH1.
//Enter Menu “Networking support -> Networking options”, select “Network packet filtering framework (Netfilter)".
//Open the submenu "IP: Netfilter Configuration", select all the options.
Back to Menu “Networking options”, select “802.1d Ethernet Bridging”.

Save the changes and Exit the menu, build the image.

$ make uImage
$ make dtbs
2. Rebuild the rootfs to add bridge utility support.

   //Get the buildroot source code
   $ git clone git://github.com/linux4sam/buildroot-at91.git
   Cloning into 'buildroot-at91'...
   remote: Counting objects: 117708, done.
   remote: Total 117708 (delta 0), reused 0 (delta 0)
   Receiving objects: 100% (117708/117708), 33.74 MiB | 687 KiB/s, done.
   Resolving deltas: 100% (79958/79958), done.
   $ cd buildroot-at91/
   //Checkout from the github repository
   $ git checkout origin/buildroot-2012.11.1-at91 -b <your branch>
   //Configure and build rootfs
   # make sama5d3ekdemo_defconfig
   # make menuconfig
   //Add support for bridge-utils: Package Selection for the target->
   //Networking applications->bridge-utils
   //"brctl" file will be generated in
   //[work directory of buildroot]/output/build/bridge-utils-1.5 directory.
$ make

3. Download the Linux image to SAMA5D35-EK.
4.2.2 How to Set PC and Bridge

Note: Here, Linux is the default OS installed on the PC.

1. Setting on PC-1:
   ```
   //Configure IP address: 192.168.3.2
   $ sudo ifconfig eth0 down
   $ sudo ifconfig eth0 192.168.3.2
   $ sudo ifconfig eth0 up
   ```

2. Setting on PC-2:
   ```
   //Configure IP address: 192.168.3.3
   $ sudo ifconfig eth1 down
   $ sudo ifconfig eth1 192.168.3.3
   $ sudo ifconfig eth1 up
   ```

3. Build and configure the bridge (SAMA5D35-EK):
   ```
   //Create a bridge named mybridge
   # brctl addbr mybridge
   //Include eth0 and eth1 into mybridge
   # brctl addif mybridge eth0
   # brctl addif mybridge eth1
   //Set the network modes
   # ifconfig eth0 0.0.0.0 promisc
   # ifconfig eth1 0.0.0.0 promisc
   # ifconfig mybridge 192.168.3.4 netmask 255.255.255.0
   # ifconfig mybridge up
   //Show mybridge
   # ifconfig mybridge
   mybridge  Link encap:Ethernet  HWaddr FE:A4:3A:79:3E:5C
     inet addr:192.168.3.4  Bcast:192.168.3.255  Mask:255.255.255.0
     inet6 addr: fe80::fca4:3aff:fe79:3e5c/64 Scope:Link
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:0 errors:0 dropped:0 overruns:0 frame:0
     TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:0
     RX bytes:0 (0.0 B)  TX bytes:468 (468.0 B)
   ```

After setting up the bridge on SAMA5D35-EK, test if mybridge works correctly by using the following commands.

```bash
// ping PC-2 IP address on PC-1
$ ping 192.168.3.3
PING 192.168.3.3 (192.168.3.3) 56(84) bytes of data.
64 bytes from 192.168.3.3: icmp_seq=1 ttl=64 time=2.31 ms
64 bytes from 192.168.3.3: icmp_seq=2 ttl=64 time=0.606 ms
64 bytes from 192.168.3.3: icmp_seq=3 ttl=64 time=0.585 ms
... 
```

```bash
// ping PC-1 IP address on PC-2
$ ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data.
64 bytes from 192.168.3.2: icmp_seq=1 ttl=64 time=0.546 ms
64 bytes from 192.168.3.2: icmp_seq=2 ttl=64 time=0.477 ms
64 bytes from 192.168.3.2: icmp_seq=3 ttl=64 time=0.613 ms
... 
```

The above text messages show that the bridge on SAMA5D35-EK has been successfully built and configured. Now it looks like PC-1 and PC-2 exist in the same network.
4.2.3 How to Run the Bridge

When acting as the bridge, the SAMA5D35-EK needs to receive packages from EMAC/GMAC and send the package to the other GMAC/EMAC, which increases the load on the CPU. Perform the following tests to identify the CPU load and the bridge transfer speed by setting SAMA5D35 in three working modes:

- Bridge acts in Full Duplex mode (Server ↔ Client or EMAC ↔ GMAC)
- Bridge acts in Simplex mode (Server → Client or EMAC → GMAC)
- Bridge acts in Simplex mode (Client → Server or GMAC → EMAC)

The commands vary in the different working modes as detailed in the following sections.

4.2.3.1 Bridge Acts in Full Duplex Mode (EMAC ↔ GMAC)

PC-1:

//Set the working mode as UDP and work as Server
$ iperf -u -s

PC-2:

//Set the working mode as UDP, work as Client and transfer in full duplex mode
$ iperf -u -c 192.168.3.2 -t 120 -i 1 -b 100M -d

Transfer report on server command line:

------------------------------------------------------------
Client connecting to 192.168.3.3, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 110 KByte (default)
------------------------------------------------------------
[  5] local 192.168.3.2 port 5001 connected with 192.168.3.3 port 57594
[  3] local 192.168.3.2 port 57673 connected with 192.168.3.3 port 5001
[ ID] Interval Transfer Bandwidth
[  3]  0.0-120.0 sec  2.10 GBytes  100 Mbits/sec
[  3] Sent 1537212 datagrams
[  5]  0.0-120.1 sec  1.99 GBytes  95.1 Mbits/sec  0.150 ms 82205/1538420 (5.3%)
[  5]  0.0-120.1 sec  1 datagrams received out-of-order
[  3] Server Report:
[  3]  0.0-120.0 sec  1.99 GBytes  95.0 Mbits/sec  0.031 ms 82606/1537211 (5.4%)
[  3]  0.0-120.0 sec  44 datagrams received out-of-order

On the bridge (SAMA5D35-EK), use the following command to capture the information on CPU usage:

//Capture the log file of CPU usage
top -d 1 -n 100 | grep idle > XXX.log
4.2.3.2 Bridge Acts in Simplex Mode (GMAC → EMAC)

PC-1:
//Set the working mode as UDP and work as Server
$ iperf -u -s

PC-2:
//Set the working mode as UDP, work as Client and transfer in half duplex mode
$ iperf -u -c 192.168.3.2 -t 120 -i 1 -b 100M

Transfer report on server command line:
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:   110 KByte (default)
------------------------------------------------------------
[  4] local 192.168.3.2 port 5001 connected with 192.168.3.3 port 51201
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[  4]  0.0-120.1 sec  1.33 GBytes  95.2 Mbits/sec  0.147 ms 52908/1025641 (5.2%)
[  4]  0.0-120.1 sec  206 datagrams received out-of-order

On the bridge (SAMA5D35-EK), use the following command to capture the information on CPU usage:
//Capture the log file of CPU usage
top -d 1 -n 100 | grep idle > XXX.log

4.2.3.3 Bridge Acts in Simplex Mode (EMAC → GMAC)

PC-2:
//Set the working mode as UDP and work as Server
$ iperf -u -s

PC-1:
//Set the working mode as UDP, work as Client and transfer in half duplex mode
$ iperf -u -c 192.168.3.3 -t 120 -i 1 -b 100M

Transfer report on server command line:
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size:   110 KByte (default)
------------------------------------------------------------
[  4] local 192.168.3.3 port 5001 connected with 192.168.3.2 port 58165
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[  4]  0.0-120.2 sec  1.33 GBytes  95.3 Mbits/sec  14.498 ms 50315/1024735 (4.9%)
[  4]  0.0-120.2 sec  206 datagrams received out-of-order

On the bridge (SAMA5D35-EK), use the following command to capture the information on CPU usage:
//Capture the log file of CPU usage
top -d 1 -n 100 | grep idle > XXX.log
4.2.4 Bandwidth and CPU Usage of Ethernet Bridge

Table 4-1 shows the transfer speed and CPU usage of the software Ethernet Bridge during data transfer with different bandwidth settings. The corresponding relation between the CPU Usage and the transfer speed is illustrated in Figure 4-2.

Table 4-1. Transfer Speed and CPU Usage of Software Bridge in Three Modes

<table>
<thead>
<tr>
<th>iPerf Bandwidth Setting (Mbps)</th>
<th>EMAC ↔ GMAC</th>
<th></th>
<th>EMAC → GMAC</th>
<th></th>
<th>GMAC → EMAC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E2G Speed</td>
<td>G2E Speed</td>
<td>CPU Usage (%)</td>
<td>E2G Speed</td>
<td>CPU Usage (%)</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>11</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
<td>30</td>
<td>29</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
<td>40</td>
<td>38</td>
<td>40</td>
<td>22</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>50</td>
<td>47</td>
<td>50</td>
<td>26</td>
</tr>
<tr>
<td>60</td>
<td>59.9</td>
<td>59.9</td>
<td>57</td>
<td>60</td>
<td>31</td>
</tr>
<tr>
<td>70</td>
<td>70</td>
<td>69.9</td>
<td>66</td>
<td>70</td>
<td>37</td>
</tr>
<tr>
<td>80</td>
<td>79.9</td>
<td>79.7</td>
<td>77</td>
<td>80</td>
<td>42</td>
</tr>
<tr>
<td>90</td>
<td>90.9</td>
<td>90.5</td>
<td>89</td>
<td>90.4</td>
<td>47</td>
</tr>
<tr>
<td>100</td>
<td>95.1</td>
<td>95</td>
<td>94</td>
<td>95.3</td>
<td>50</td>
</tr>
</tbody>
</table>

Figure 4-2. CPU Usage vs. Transfer Speed
5. **Summary**

This software Ethernet Bridge solution introduced in this application note provides an alternate option to customers to realize an Ethernet Bridge function with SAMA5D3/D4 products, especially when without hardware Ethernet Bridge.

During the operation of the software bridge function, SAMA5D3/D4 CPU involvement is required. From the above analysis of the relation between the CPU usage and different transfer speeds, we arrive at the conclusion that with the increase of the transmission bandwidth, the CPU usage also increases with linearity. Based on the application requirements and spare CPU bandwidth, customers can decide the speed of the Ethernet Bridge in their applications.
## Revision History

<table>
<thead>
<tr>
<th>Doc. Rev.</th>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>30-Jan-2015</td>
<td>First release</td>
</tr>
</tbody>
</table>