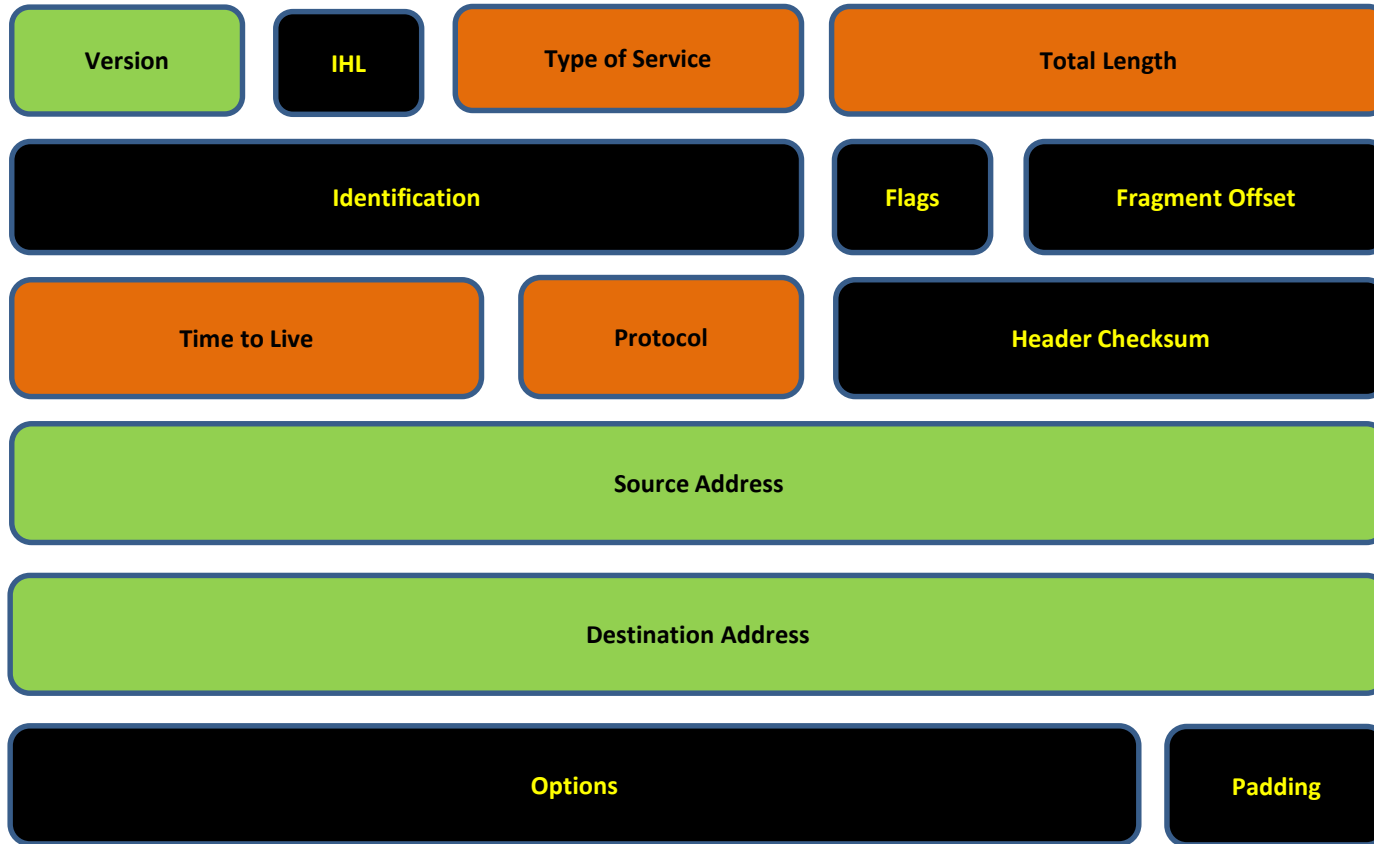


ARM Development Primer



Networking the ATSAME54P20A

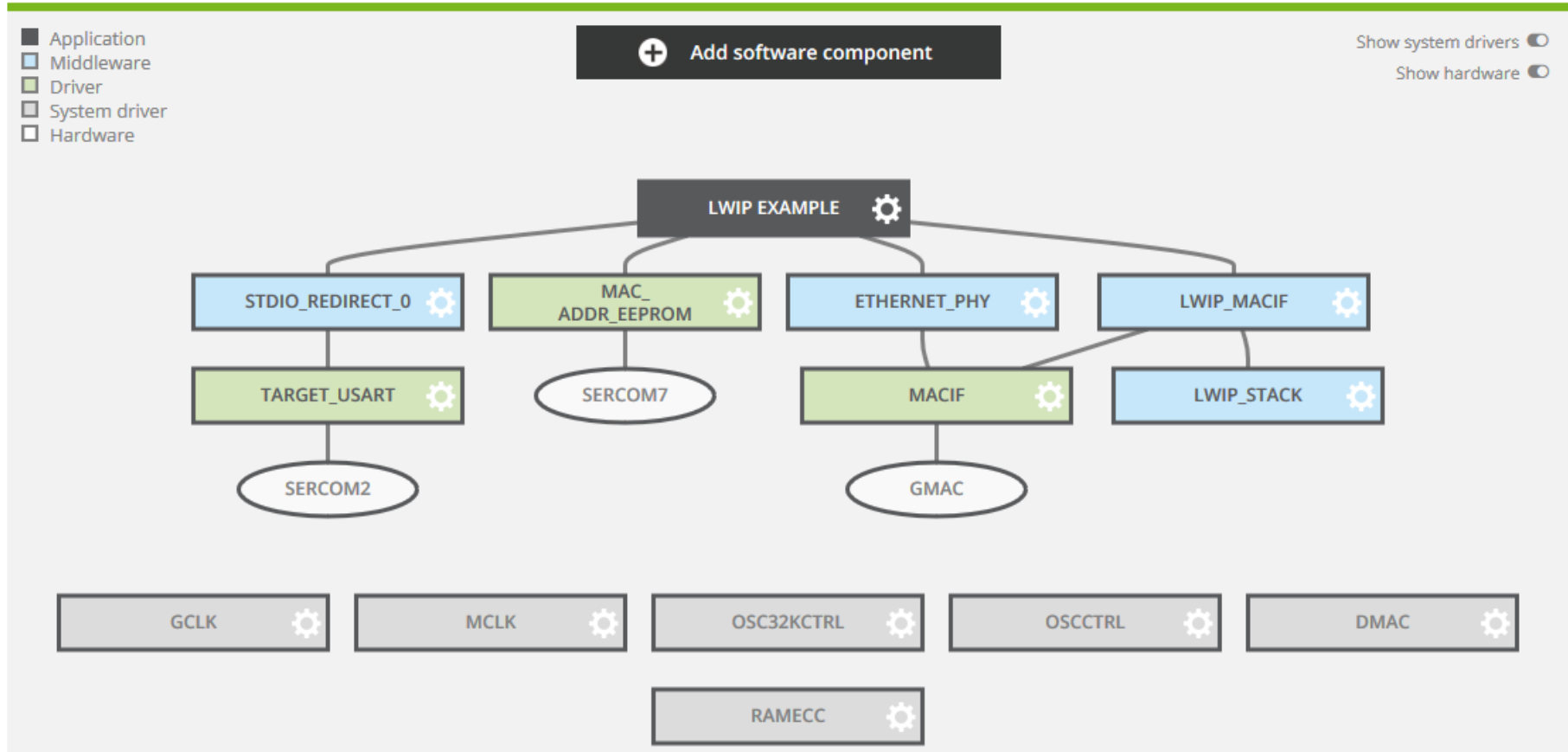
November 17, 2017

FRED EADY

ARM Development Primer

Bringing Up the LwIP Stack

MY SOFTWARE COMPONENTS



ARM Development Primer

Bringing Up the LwIP Stack

STDIO_REDIRECT_0

STDIO Redirect, printf to specified io

GENERAL

 User guide

 Rename component

 Remove component

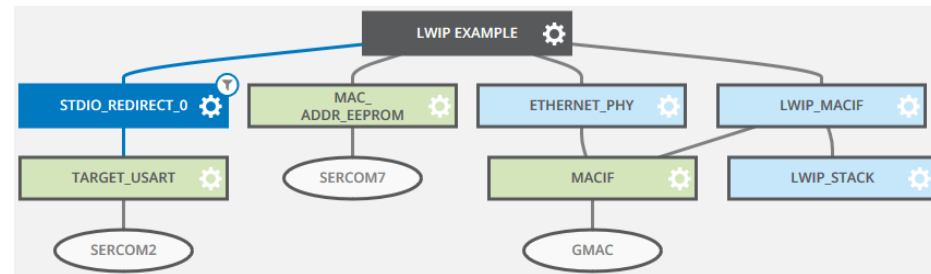
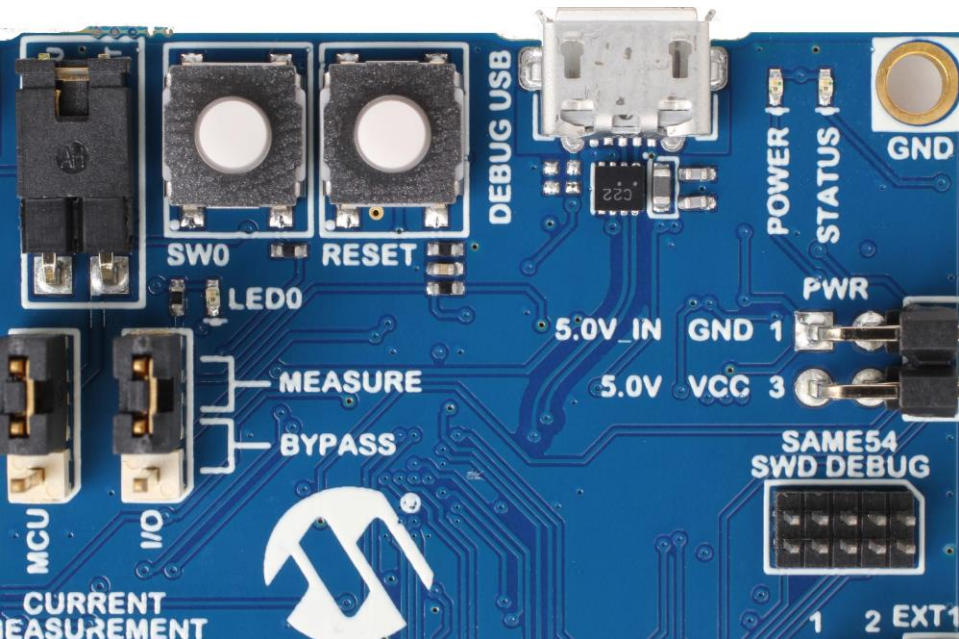
COMPONENT SETTINGS

Variant:

DEPENDENCIES

Target IO:

If you can't find your component in the list make sure you have selected the correct variant for this component, and also correct driver and/or mode for the component you wish to select.



ARM Development Primer

Bringing Up the LwIP Stack

TARGET_USART

Universal Asynchronous receiver/transmitter (UART) communication in synchronous/blocking mode

GENERAL

 User guide

 Rename component

 Remove component

COMPONENT SETTINGS

Driver: HAL:Driver:USART_Sync
Mode: UART
Instance: SERCOM2

CLOCKS

Core: Generic clock generator 0 (12 MHz)
Slow: Generic clock generator 3 (32.77 kHz)

SIGNALS

RX: PB24
TX: PB25

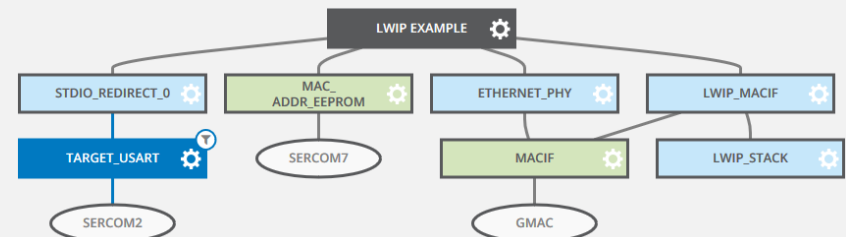
HAL:DRIVER:USART SYNC (UART) CONFIGURATION ON SERCOM2

BASIC CONFIGURATION

Receive buffer enable: ☒
Transmitt buffer enable: ☒
Frame parity: No parity
Character Size: 8 bits
Stop Bit: One stop bit
Baud rate: 9600

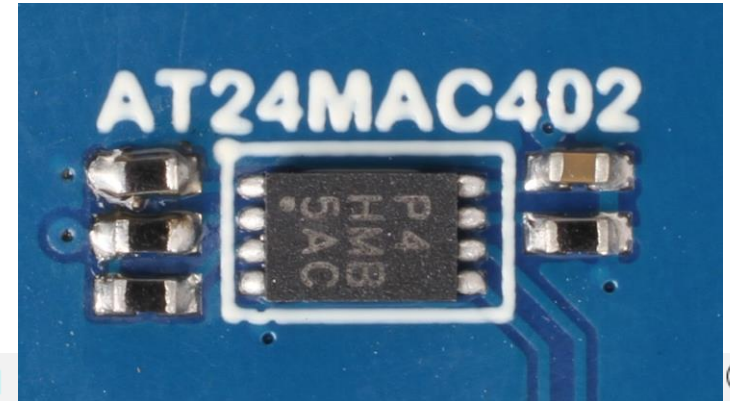
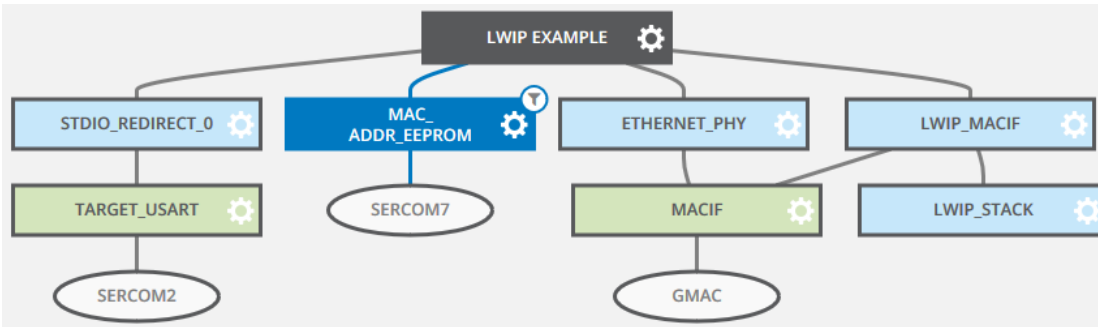
ADVANCED CONFIGURATION

Enable: ☐



ARM Development Primer

Bringing Up the LwIP Stack



MAC_ADDR_EEPROM

I2C Master driver in synchronous/blocking master mode

GENERAL

User guide

Rename component

Remove component

COMPONENT SETTINGS

Driver: HAL:Driver:I2C_Master_Sync
Mode: I2C Master Standard/Fast-mode
Instance: SERCOM7

CLOCKS

Core: Generic clock generator 0 (12 MHz)
Slow: Generic clock generator 3 (32.77 kHz)

SIGNALS

SCL: PD09
SDA: PD08

HAL:DRIVER:I2C MASTER SYNC (I2C MASTER STANDARD/FAST-MODE) CONFIGURATION ON SERCOM7

BASIC

I2C Bus clock speed (Hz): 100000

ADVANCED

Enable: ☐

ARM Development Primer

Bringing Up the LwIP Stack

ETHERNET_PHY

Generic 10Base-T and 100Base-TX PHY Driver



GENERAL

User guide

Rename component

Remove component

COMPONENT SETTINGS

Variant:

DEPENDENCIES

Communication IO:

If you can't find your component in the list make sure you have selected the correct variant for this component, and also correct driver and/or mode for the component you wish to select.

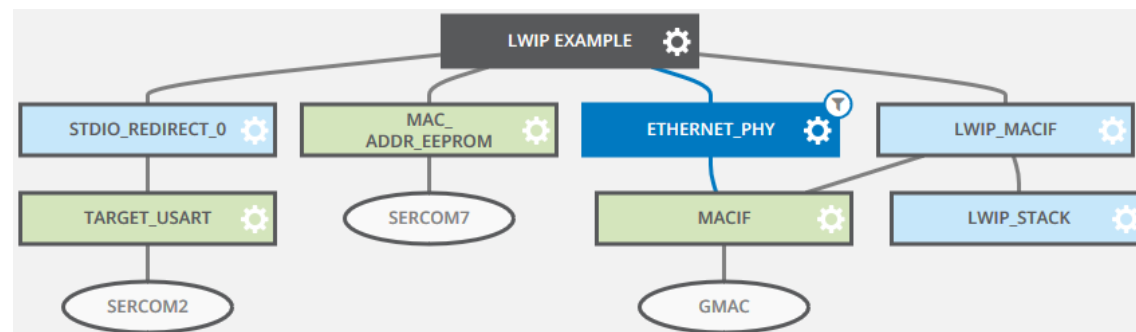
GENERIC ETHERNET PHY DRIVER CONFIGURATION

BASIC CONFIGURATION

PHY Address:

CONTROL REGISTER (REGISTER 0) SETTINGS

Enable: ☐



ARM Development Primer

Bringing Up the LwIP Stack

MACIF

Ethernet MAC in asynchronous mode

GENERAL

- User guide
- Rename component
- Remove component

COMPONENT SETTINGS

Driver: HAL:Driver:GMAC_Async

SIGNALS

GCOL:	☐ PC21
GCRS:	☐ PA16
GMDC:	PC11
GMDIO:	PC12
GRX/0:	☒ PA13
GRX/1:	☒ PA12
GRX/2:	☐ PC15
GRX/3:	☐ PC14
GRXCK:	☐ PC18
GRXDV:	☒ PC20
GRXER:	☒ PA15
GTX/0:	☒ PA18
GTX/1:	☒ PA19
GTX/2:	☐ PC16
GTX/3:	☐ PC17
GTXCK:	☒ PA14
GTXEN:	☒ PA17
GTXER:	☐ PC19

Diagram

```
graph TD;
    LE[LWIP EXAMPLE] --> SRD[STDIO_REDIRECT_0];
    LE --> MAE[MAC_ADDR_EEPROM];
    LE --> EP[ETHERNET_PHY];
    LE --> LMACIF[LWIP_MACIF];
    SRD --> TU[TARGET_USART];
    TU --> S2[SERCOM2];
    MAE --> S7[SERCOM7];
    EP --> MACIF[MACIF];
    MACIF --> GMAC[GMAC];
    LMACIF --> LSTACK[LWIP_STACK];
```

ARM Development Primer

Bringing Up the LwIP Stack

HAL:DRIVER:GMAC ASYNC (GMAC) CONFIGURATION ON GMAC

NETWORK CONTROL CONFIGURATION

Enable LoopBack Local: ☐

Management Port Enable: ☒

Enable write for Static Register: ☐

Enable Back pressure: ☐

Enable PFC Priority-based Pause Reception: ☐

Enable PFC Priority-based Pause Frame: ☐

MII CONFIGURATION

Enable: ☒

MII Mode:

PHY Clause Operation:

DMA CONFIGURATION

Enable: ☒

Fixed Burst Length for DMA Data Operations:

Endian Swap Mode Enable for Management Descriptor Accesses: ☐

Endian Swap Mode Enable for Packet Data Accesses: ☐

Receiver Packet Buffer Memory Size Select:

Transmitter Packet Buffer Memory Size Select:

Transmitter Checksum Generation Offload Enable: ☐

DMA Receive Buffer Size:

DMA Discard Received Packets: ☐

NETWORK CONFIGURATION

100Mbps Speed: ☒

Enable Full Duplex: ☒

Discard Non-VLAN Frames: ☐

Enable Jumbo Frame: ☐

Copy All Frames: ☐

No broadcast: ☐

Multicast Hash Enable: ☐

Unicast Hash Enable: ☐

1536 Maximum Frame Size: ☒

Retry Test: ☐

Pause Enable: ☐

Receive Buffer Offset:

Length Field Error Frame Discard: ☐

Remove FCS: ☐

MDC Clock Division:

Disable Copy of Pause Frames: ☐

Receive Checksum Offload Enable: ☐

Enable Frames Received in Half Duplex: ☐

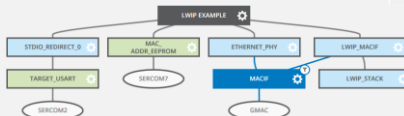
Ignore RX FCS: ☐

IP Stretch Enable: ☐

Receive Bad Preamble: ☐

Ignore IPG GRXER: ☐





STACKED VLAN PROCESSING

Enable: ☐

ARM Development Primer

Bringing Up the LwIP Stack

LWIP_MACIF

Ethernet MAC interface used by LwIP TCP/IP Stack.



GENERAL

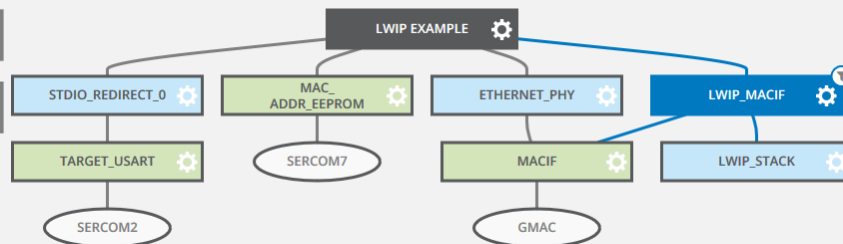
User guide

Rename component

Remove component

COMPONENT SETTINGS

Variant:



DEPENDENCIES

TCP/IP Stack:

Communication IO:

If you can't find your component in the list make sure you have selected the correct variant for this component, and also correct driver and/or mode for the component you wish to select.

LWIP MAC INTERFACE CONFIGURATION

BASIC CONFIGURATION

Enable DHCP: ☐

Enable Broadcast: ☒

Enable ARP: ☒

Enable IGMP: ☐

Enable Point to Point: ☐

MTU Size:

Host Name:

Host Name descriptive abbreviation:

ENABLE STATIC IP ADDRESS

Enable: ☒

IP Address:

Netmask Address:

Gateway Address:



ARM Development Primer

Bringing Up the LwIP Stack

LWIP_STACK

LwIP is a small independent implementation of the TCP/IP protocol suite.

GENERAL

User guide

Rename component

Remove component

COMPONENT SETTINGS

Variant: Ethernet:IPStack:LwIP

DEPENDENCIES

If you can't find your component in the list make sure you have selected the correct variant for this component, and also correct driver and/or mode for the component you wish to select.

LWIP CONFIGURATION

BASIC CONFIGURATION

Enable DHCP: ☒

NO RTOS: ☒

The size of the heap memory: 4000

Enables TCP: ☒

Enables UDP: ☒

Enables ICMP: ☒

Enables AUTOIP: ☐

Enables SNMP: ☐

Enables IGMP: ☐

Enables SLIP interface: ☐

Enables DNS: ☐

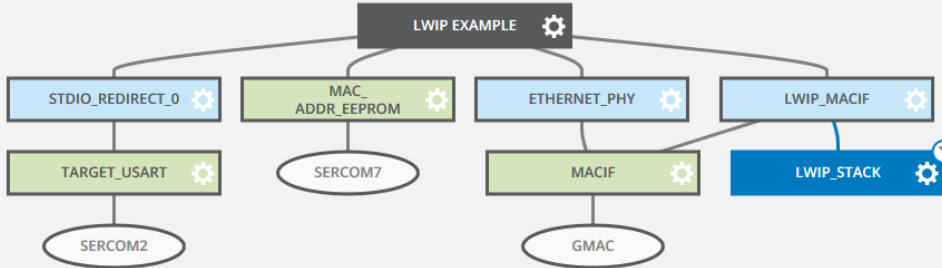
Enable PPP: ☐

Enable PPPoE: ☐

Enable PPPoS: ☐

ADVANCED CONFIGURATION

Enable: ☐

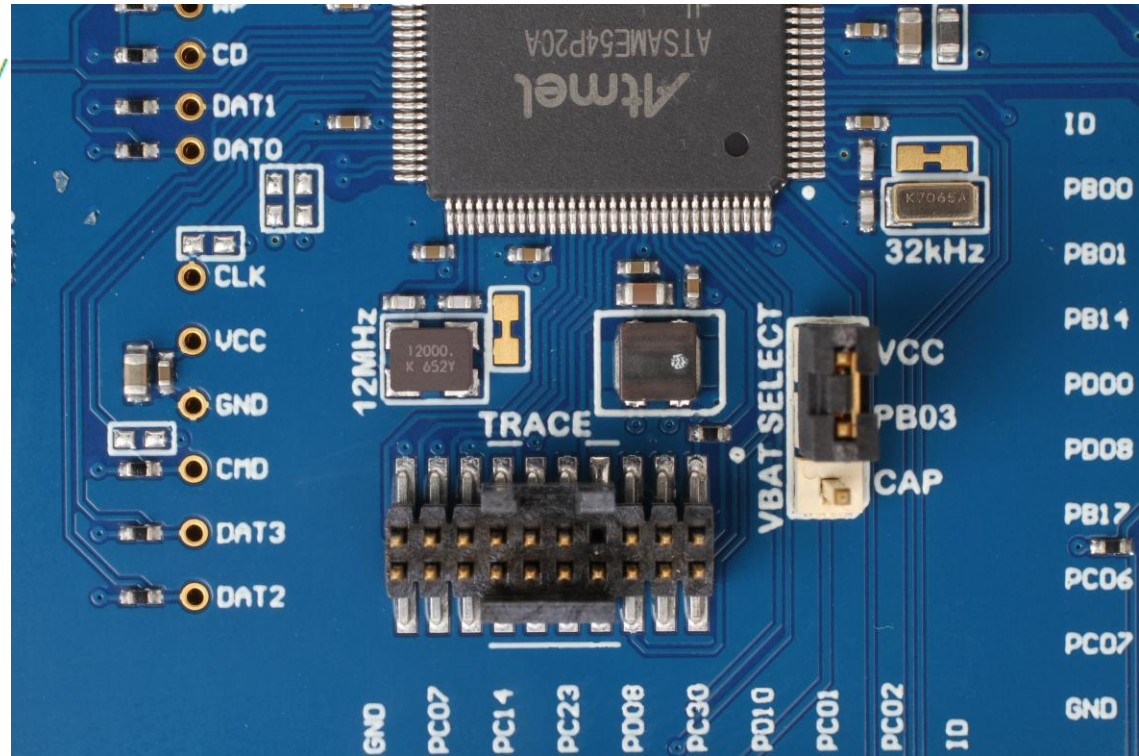


```
graph TD; LWIP_EXAMPLE[LWIP EXAMPLE] --- STDIO_REDIRECT_0[STDIO_REDIRECT_0]; LWIP_EXAMPLE --- MAC_ADDR_EEPROM[MAC_ADDR_EEPROM]; LWIP_EXAMPLE --- ETHERNET_PHY[ETHERNET_PHY]; LWIP_EXAMPLE --- LWIP_MACIF[LWIP_MACIF]; STDIO_REDIRECT_0 --- TARGET_USART[TARGET_USART]; TARGET_USART --- SERCOM2((SERCOM2)); MAC_ADDR_EEPROM --- SERCOM7((SERCOM7)); ETHERNET_PHY --- MACIF[MACIF]; MACIF --- GMAC((GMAC)); LWIP_MACIF --- LWIP_STACK[LWIP_STACK];
```

ARM Development Primer

Bringing Up the LwIP Stack

```
/* Saved total time in mS since timer was enabled */  
volatile static u32_t systick_timems;  
volatile static bool  rcv_flag = false;  
static bool          link_up  = false;  
  
u32_t sys_now(void)  
{  
    return systick_timems;  
}  
  
void SysTick_Handler(void)  
{  
    systick_timems++;  
}  
  
void systick_enable(void)  
{  
    systick_timems = 0;  
    SysTick_Config((CONF_CPU_FREQUENCY) / 1000);  
}
```



ARM Development Primer

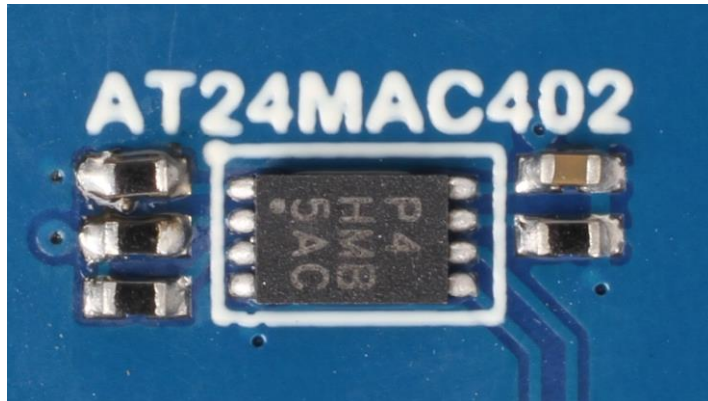
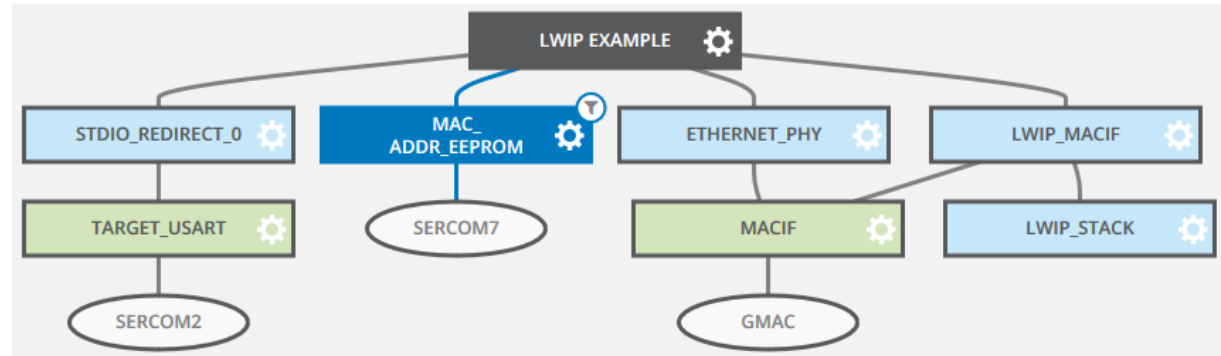
Bringing Up the LwIP Stack

```
#ifndef _LWIP_DEMO_CONFIG_H_INCLUDED
#define _LWIP_DEMO_CONFIG_H_INCLUDED

// <h> Basic Configuration

// <o> AT24MAC Address <0x0-0xFF>
// <i> AT24MAC address
// <id> at24mac_slave_address
#ifndef CONF_AT24MAC_ADDRESS
#define CONF_AT24MAC_ADDRESS 0x5e
#endif
// </h>

#endif /* _LWIP_DEMO_CONFIG_H_INCLUDED */
```

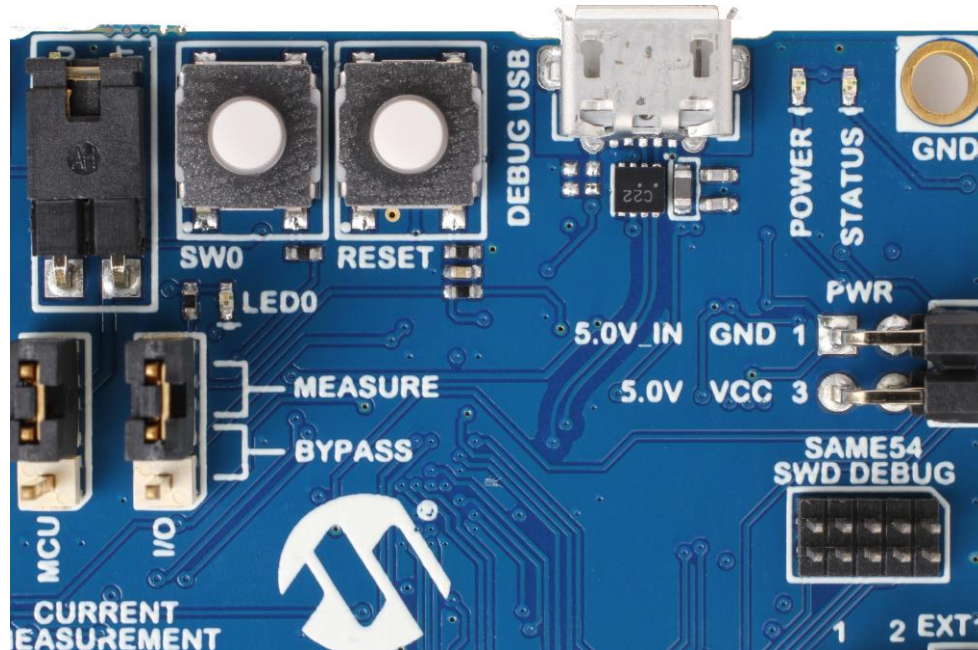


```
static void read_macaddress(u8_t *mac)
{
    #if CONF_AT24MAC_ADDRESS != 0
        uint8_t addr = 0x9A;

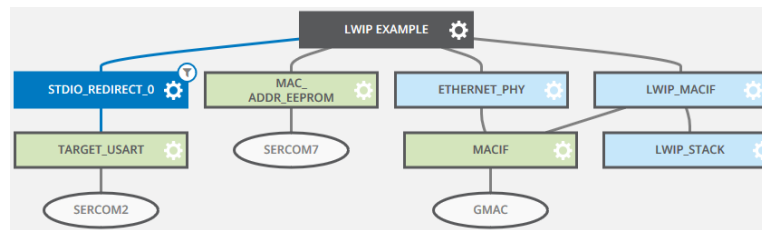
        i2c_m_sync_enable(&MAC_ADDR_EEPROM);
        i2c_m_sync_set_slaveaddr(&MAC_ADDR_EEPROM, CONF_AT24MAC_ADDRESS, I2C_M_SEVEN);
        io_write(&(MAC_ADDR_EEPROM.io), &addr, 1);
        io_read(&(MAC_ADDR_EEPROM.io), mac, 6);
    #else
        /* set mac to 0x11 if no EEPROM mounted */
        memset(mac, 0x11, 6);
    #endif
}
```

ARM Development Primer

Bringing Up the LwIP Stack



```
static void print_ipaddress(void)
{
    static char tmp_buff[16];
    printf("IP_ADDR      : %s\r\n", ipaddr_ntoa_r((const ip_addr_t *)&(LWIP_MACIF_desc.ip_addr), tmp_buff, 16));
    printf("NET_MASK      : %s\r\n", ipaddr_ntoa_r((const ip_addr_t *)&(LWIP_MACIF_desc.netmask), tmp_buff, 16));
    printf("GATEWAY_IP    : %s\r\n", ipaddr_ntoa_r((const ip_addr_t *)&(LWIP_MACIF_desc.gw), tmp_buff, 16));
}
```



ARM Development Primer

Bringing Up the LwIP Stack

```
int main(void)
{
    int32_t ret;
    u8_t mac[6];

    atmel_start_init();

    /* Read MacAddress from EEPROM */
    read_macaddress(mac);

    systick_enable();

    printf("\r\nRiding the EDTP Ethernet LAN with Digi-Key and the CEC!\r\n");
    mac_async_register_callback(&MACIF, MAC_ASYNC_RECEIVE_CB, (FUNC_PTR)mac_receive_cb);

    eth_ipstack_init();
    do {
        ret = ethernet_phy_get_link_status(&ETHERNET_PHY_desc, &link_up);
        if (ret == ERR_NONE && link_up) {
            break;
        }
    } while (true);
    printf("Ethernet Connection established\r\n");
    LWIP_MACIF_init(mac);
    netif_set_up(&LWIP_MACIF_desc);

    netif_set_default(&LWIP_MACIF_desc);
    mac_async_enable(&MACIF);

    while (true) {
        if (recv_flag) {
            recv_flag = false;
            ethernetif_mac_input(&LWIP_MACIF_desc);
        }
        /* LWIP timers - ARP, DHCP, TCP, etc. */
        sys_check_timeouts();

        /* Print IP address info */
        if (link_up && LWIP_MACIF_desc.ip_addr.addr) {
            link_up = false;
            print_ipaddress();
        }
    }
}
```



ARM Development Primer

Day 5's Done

The image shows two overlapping windows. The background window is 'Serial Input/Output Monitor' with a menu bar (File, Edit, View, Configuration) and a toolbar with icons for ASCII, HEX, ASCII Send, HEX Send, Line Status, Clear Terminal, Columns, Display, Data Graph, and Ribbon/Classic menu styles. The main text area shows the following output:

```
Riding the EDTP Ethernet LAN with Digi-Key and the CEC!  
Ethernet Connection established  
IP_ADDR      : 192.168.1.253  
NET_MASK     : 255.255.255.0  
GATEWAY_IP   : 192.168.1.1
```

Below the text area is a hex dump showing two lines of data:

```
4E 45 54 5F 4D 41 53 4B 20 20 20 3A 20 32 35 35  
47 41 54 45 57 41 59 5F 49 50 20 3A 20 31 39 32
```

At the bottom of the window are input fields for 'HEX' and 'COM14 8N1 9600', and a status bar showing 'R0 C0 R8 C1'.

The foreground window is a Windows Command Prompt titled 'C:\WINDOWS\system32\cmd.exe'. It displays the output of a ping command:

```
Microsoft Windows [Version 10.0.15063]  
(c) 2017 Microsoft Corporation. All rights reserved.  
  
C:\Users\Fred>ping 192.168.1.253  
  
Pinging 192.168.1.253 with 32 bytes of data:  
Reply from 192.168.1.253: bytes=32 time=38ms TTL=255  
Reply from 192.168.1.253: bytes=32 time=1ms TTL=255  
Reply from 192.168.1.253: bytes=32 time=2ms TTL=255  
Reply from 192.168.1.253: bytes=32 time=1ms TTL=255  
  
Ping statistics for 192.168.1.253:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 1ms, Maximum = 38ms, Average = 10ms  
  
C:\Users\Fred>
```

Presented by: