

Prototyping a Wi-Fi ARMED IoT Sensor Node

August 27, 2018 Fred Eady

Presented by:

CONTINUING



DesignNews

ARM Your Sensors AGENDA

ARMed Abstraction Lock and Load Firing UDP Datagrams Day 1 Summary

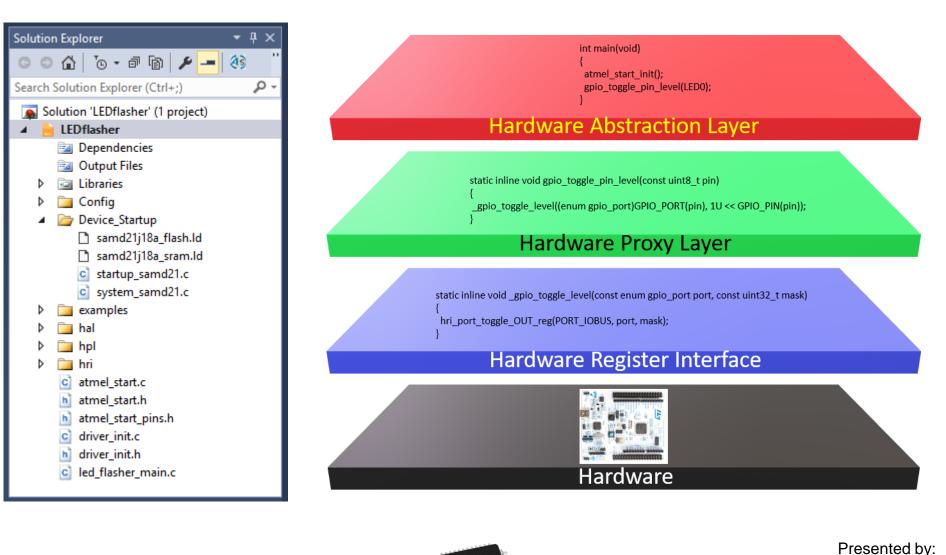


Presented by:

CONTINUING EDUCATION



ARMed Abstraction – Advanced Software Framework (ASF)



DesignNews



CONTINUING EDUCATION

ARMed Abstraction – Advanced Software Framework (ASF)



_usart_transmit(void *const hw, uint8_t *const buf, uint32_t length);

DesignNews

void *hw;

};

struct _usart_device

struct _usart_callbacks usart_cb

usart init(void *const hw);

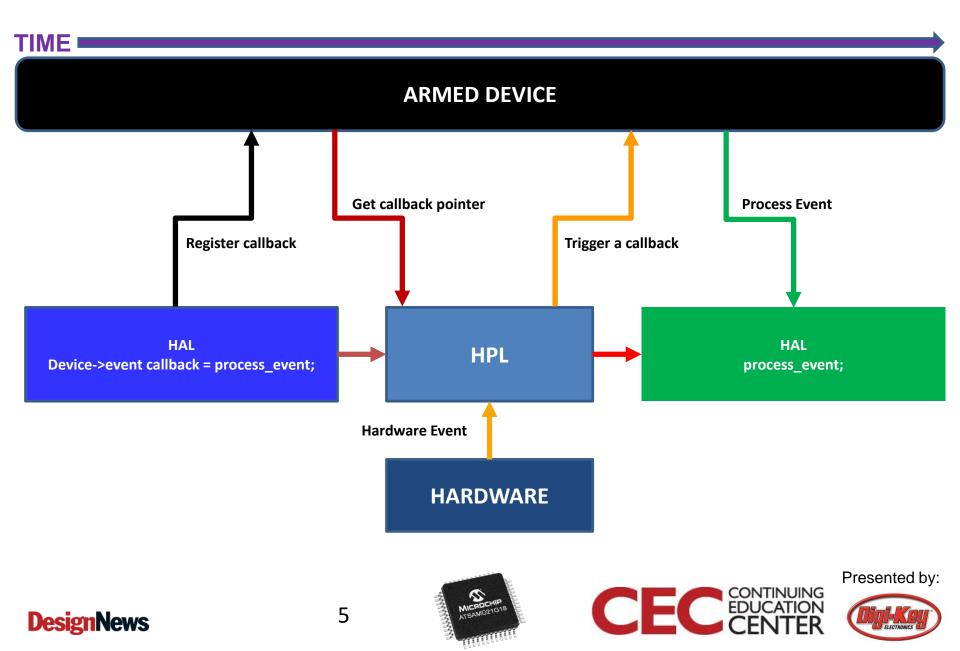
_usart_deinit(void *const hw);

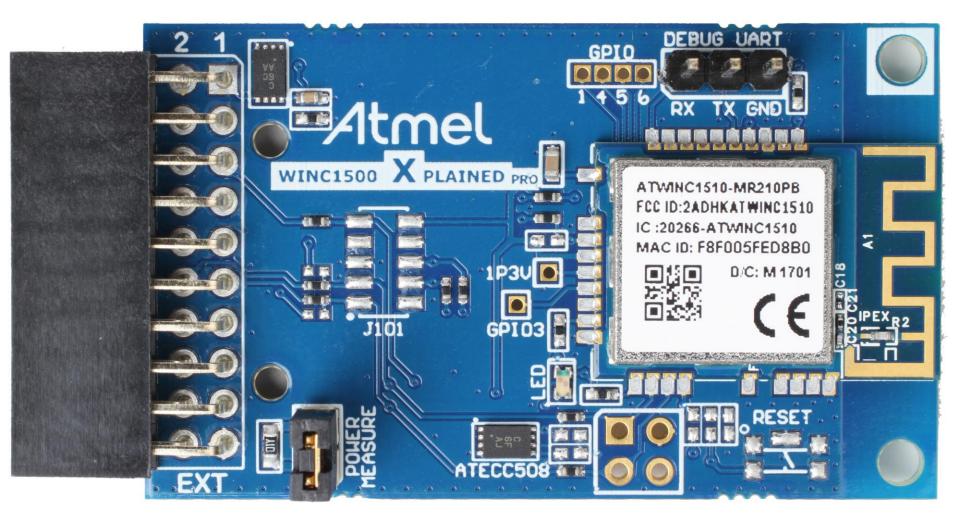






ARMed Abstraction – Advanced Software Framework (ASF)





DesignNews







Basic In	ternet Settings	Local Network	Advanced Routing	Administration				
Router Detail	s Edit			DHCP Server ZEnable	d			
Host name:	armnet			Start IP address:	192	. 168	. 10	. 100
IP address:	192.168.10.1			Maximum number of users:	50		1 to	155
Subnet mask: 255.255.255.0				IP address range:	to	2 .168 2 .168		
				Client lease time:	144	10	Min	utes
				Static DNS 1:	0	0	0	0
				Static DNS 2:	0	0	0	0
				Static DNS 3:	0	0	0	0
				WINS:	0	0	0	0









Lock and Load – Station Mode



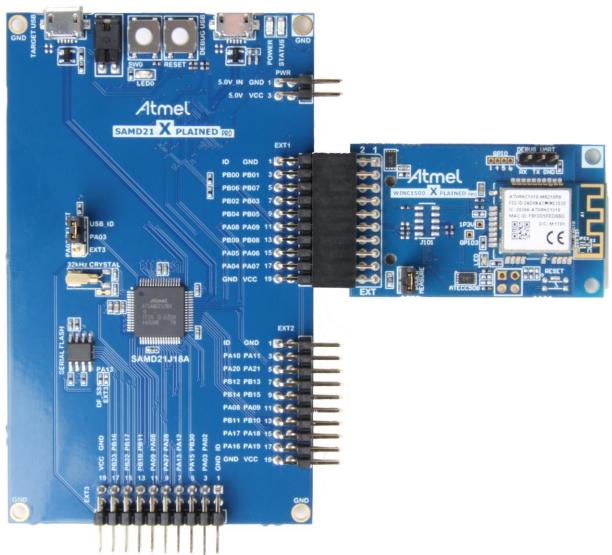
/** Wi-Fi Settings */
#define MAIN_WLAN_SSID "armwifi" /* < Destination SSID */
#define MAIN_WLAN_AUTH M2M_WIFI_SEC_WPA_PSK /* < Security manner */
#define MAIN_WLAN_PSK "armwifipass" /* < Password for Destination SSID */</pre>











DesignNews







Lock and Load – Station Mode

```
static void wifi cb(uint8 t u8MsgType, void *pvMsg)
            ł
                switch (u8MsgType) {
                case M2M_WIFI_RESP_CON_STATE_CHANGED: {
                     tstrM2mWifiStateChanged *pstrWifiState = (tstrM2mWifiStateChanged *)pvMsg;
                     if (pstrWifiState->u8CurrState == M2M WIFI CONNECTED) {
                         m2m_wifi_request_dhcp_client();
                     } else if (pstrWifiState->u8CurrState == M2M WIFI DISCONNECTED) {
                         printf("Wi-Fi disconnected\r\n");
                         /* Connect to defined AP. */
                         m2m wifi connect(
                             (char *)MAIN_WLAN_SSID, sizeof(MAIN_WLAN_SSID), MAIN_WLAN_AUTH, (void *)MAIN_WLAN_PSK, M2M_WIFI_CH_ALL);
                     }
                     break;
                }
                case M2M WIFI REQ DHCP CONF: {
                     uint8 t *pu8IPAddress = (uint8 t *)pvMsg;
                     printf("Wi-Fi connected\r\n");
                     printf("Wi-Fi IP is %u.%u.%u.%u\r\n", pu8IPAddress[0], pu8IPAddress[1], pu8IPAddress[2], pu8IPAddress[3]);
                                       TIME 

                     break;
                }
                                                                  ARMED DEVICE
                default: {
                     break;
                                                               Get callback pointer
                                                                                         Process Event
                                                   Register callback
                                                                             Trigger a callback
                }
            }
                                                  HAL
                                                                       HPL
                                          Device->event callback = process_event;
                                                                                         process_event;
                                                               Hardware Event
                                                                    HARDWARE
                                                                                                                               Presented by:
                                                                                                              CONTINUING
                                                                                                              FDUCATION
                                                10
DesignNews
```

```
int main(void)
   tstrWifiInitParam param;
   atmel_start_init();
   /* Initialize Wi-Fi parameters structure. */
   memset((uint8_t *)&param, 0, sizeof(tstrWifiInitParam));
   /* Initialize Wi-Fi driver with data and status callbacks. */
    param.pfAppWifiCb = wifi_cb;
   wifi_init(&param);
   printf("Connecting to %s.\r\n", (char *)MAIN_WLAN_SSID);
   /* Connect to defined AP. */
   m2m_wifi_connect(
        (char *)MAIN WLAN_SSID, sizeof(MAIN_WLAN_SSID), MAIN_WLAN_AUTH, (void *)MAIN_WLAN_PSK, M2M_WIFI_CH_ALL);
   while (1) {
       /* Handle pending events from network controller. */
       while (m2m_wifi_handle_events(NULL) != M2M_SUCCESS) {
    return 0;
```





CONTINUING

EDU





}



```
case M2M_WIFI_REQ_DHCP_CONF: {
    uint8_t *pu8IPAddress = (uint8_t *)pvMsg;
    printf("Wi-Fi connected\r\n");
    printf("Wi-Fi IP is %u.%u.%u.%u\r\n", pu8IPAddress[0], pu8IPAddress[1], pu8IPAddress[2], pu8IPAddress[3]);
    break;
                                                                                                      @ A ~ = 19 # stores
1
```

Serial Port Control Panel		SANSUNG
EDBG Virtual COM Port (COM11)		SI CO
Baud rate Parity Stop bits 115200 None 1 bit		Land
Terminal 2 (APP)(INFO)Chip ID 1503a0		
(APP)(INFO)DriverVerInfo: 0x13301354		Constant of the local division of the local
(APP)(INFO)Firmware ver : 19.5. (APP)(INFO)Firmware Build Oct 4		Show widget on the homepage
(APP)(INFO)Firmware Min driver ve	avices and their status.	
(APP)(INFO)Driver ver: 19.5.4 MyNetwork Gue	st Network	+ Add a Device C
(APP)(INFO)Driver built at Aug 12		
Connecting to armwifi. Wi-Fi connected Wi-Fi IP is 192.168.10.100	B SHEDLAPTOP Online	WINC-d8-b0 Online
~~		Presented by:
DesignNews 12		EC CONTINUING EDUCATION CENTER

ARM Your Sensors Firing UDP Datagrams

UDP Test Tool 3.0	- 🗆 X
File Edit Clear Help	₽ A
Remote	Host
- IP Address/Name Port	Local IP Address Port Port
localhost 12345 Send	192.168.10.115/8088 Bind 8088
Edit/Send Data: 21	Edit/Received Data: 0
Enter data to send	~
ASCII O Hex Line Feed Carriage Return	Display data as: OASCII O Binary O Decimal O Hex
Auto Send Elaps Time	Packet Timer
Send every 1 sec. 00:00:00 Reset	0.0 S O Average O Last Pkt. Reset
- Sent HEX Data Log	Received HEX Data Log
/** Wi-Fi Settings */ #define MAIN_WLAN_SSID "armwifi" #define MAIN_WLAN_AUTH M2M_WIFI_SEC_WPA_ #define MAIN_WLAN_PSK "armwifipass" #define MAIN_WIFI_M2M_SERVER_IP ØxFFFFFF #define MAIN_WIFI_M2M_SERVER_PORT (8088) #define MAIN_WIFI_M2M_REPORT_INTERVAL (1 #define MAIN_WIFI_M2M_BUFFER_SIZE 1460	_PSK /**< Security manner */ /**< Password for Destination SSID * FFF /* 255.255.255.255 */)
/** UDP MAX packet count */	
#define MAIN_WIFI_M2M_PACKET_COUNT 10	
otal Bytes Sent: 0 Total Bytes Received: 2035	8/12/2018 5:21:45 PM









ARM Your Sensors Firing UDP Datagrams

```
int main(void)
    tstrWifiInitParam param;
    int8 t
                       ret;
    struct sockaddr_in addr;
    pktData = 0;
    /* Initialize the board. */
    atmel_start_init();
    /* Initialize the BSP. */
    nm_bsp_init();
    /* Initialize socket address structure. */
    addr.sin family
                        = AF INET;
                      = _htons(MAIN_WIFI_M2M_SERVER_PORT);
    addr.sin port
    addr.sin_addr.s_addr = _htonl(MAIN_WIFI_M2M_SERVER_IP);
    /* Initialize Wi-Fi parameters structure. */
   memset((uint8_t *)&param, 0, sizeof(tstrWifiInitParam));
    /* Initialize Wi-Fi driver with data and status callbacks. */
    param.pfAppWifiCb = wifi_cb;
   wifi_init(&param);
    /* Initialize socket module */
    socketInit();
```

{







ARM Your Sensors Firing UDP Datagrams

```
/* Connect to router. */
m2m wifi connect(
   (char *)MAIN_WLAN_SSID, sizeof(MAIN_WLAN_SSID), MAIN_WLAN_AUTH, (char *)MAIN_WLAN_PSK, M2M_WIFI_CH_ALL);
while (1) {
   if (packetCnt == MAIN_WIFI_M2M_PACKET_COUNT) {
        close(tx socket);
        tx socket = -1;
        break;
    }
   m2m_wifi_handle_events(NULL);
    if (wifi_connected == M2M_WIFI_CONNECTED) {
        /* Create socket for Tx UDP */
        if (tx_socket < 0) {</pre>
            if ((tx_socket = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {</pre>
                printf("main : failed to create TX UDP client socket error!\r\n");
                continue;
            }
        }
        udpPkt[0] = 0x01;
        udpPkt[1] = pktData++;
        ret = sendto(tx socket,
                     &udpPkt,
                     sizeof(udpPkt),
                     0,
                     (struct sockaddr *)&addr,
                     sizeof(addr));
        if (ret == M2M_SUCCESS) {
            packetCnt += 1;
        }
    }
}
                                                                                                              Presented by:
                                                                                             CONTINUING
                                                                                             EDU
```

DesignNews



Firing UDP Datagrams

ile Edit Clear Help				
Remote	n r Host			
IP Address/Name Port	Local IP Address Port			
localhost 12345 Send	192.168.10.115/8088 Bind 8088			
Edit/Send Data: 21	Edit/Received Data: 5			
Enter data to send	01 09			
ASCII O Hex Line Feed Carriage Return Auto Send Elaps Time	Display data as: OASCII O Binary O Decimal O Hex			
Auto Send Elaps Time Send every 1 sec. 00:00:00 Reset	0.0 S O Average O Last Pkt. Reset			
Sent HEX Data Log	Received HEX Data Log			
	{192.168.10.100} 01 00 {192.168.10.100} 01 01 {192.168.10.100} 01 01 {192.168.10.100} 01 02 {192.168.10.100} 01 03 {192.168.10.100} 01 04 {192.168.10.100} 01 05 {192.168.10.100} 01 05 {192.168.10.100} 01 07 {192.168.10.100} 01 08 {192.168.10.100} 01 09			
	{192.168.10.100} 01 00 {192.168.10.100} 01 01 {192.168.10.100} 01 01 {192.168.10.100} 01 02			
- Display	{192.168.10.100} 01 00 {192.168.10.100} 01 01			









Day 1 Summary

🖄 UDP Test Tool 3.0 × _ 2 File Edit Clear Help Remote Host Local IP Address IP Address/Name Port Port -Send 192.168.10.115/8088 Bind localhost 12345 8088 Edit/Send Data: 21 Edit/Received Data: 5 01 09 Enter data to send ... ASCII
 Hex Line Feed Carriage Return Display data as: O ASCII O Binary O Decimal O Hex Auto Send Packet Timer Elaps Time 00:00:00 Reset 0.0 S O Average
 Last Pkt. Reset Send every 1 sec. Sent HEX Data Log Received HEX Data Log {192.168.10.100} 01 00 {192.168.10.100} 01 01 {192 168 10 100} 01 02 {192.168.10.100} 01 03 {192.168.10.100} 01 04 {192,168,10,100} 01 05 {192.168.10.100} 01 06 {192 168 10 100} 01 07 {192,168,10,100} 01 08 {192,168,10,100} 01 09 {192.168.10.100} 01 00 {192.168.10.100} 01 01 {192 168 10 100} 01 02 {192.168.10.100} 01 03 Display Display Sound Clear Log Clear Log Time Date Time Date Enabled Total Bytes Sent: 0 Total Bytes Received: 2035 8/12/2018 5:20:43 PM

/* Connect to router. */
m2m_wifi_connect(

(char *)MAIN_WLAN_SSID, sizeof(MAIN_WLAN_SSID), MAIN_WLAN_AUTH, (char *)MAIN_WLAN_PSK, M2M_WIFI_CH_ALL);

```
while (1) {
```

}

```
if (packetCnt == MAIN_WIFI_M2M_PACKET_COUNT) {
    close(tx_socket);
    tx_socket = -1;
    break;
}
m2m wifi handle events(NULL);
if (wifi_connected == M2M_WIFI_CONNECTED) {
    /* Create socket for Tx UDP */
    if (tx socket < 0) {</pre>
        if ((tx_socket = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {</pre>
            printf("main : failed to create TX UDP client socket error!\r\n");
            continue;
       }
    }
    udpPkt[0] = 0x01;
    udpPkt[1] = pktData++;
```

(struct sockaddr *)&addr, sizeof(addr));

if (ret == M2M_SUCCESS) {
 packetCnt += 1;

DesignNews







