AN3206

Serialized G3-PLC Device on ATSAM4CMx-DB

Summary

This application note shows how to add G3-PLC communications to the meter demonstration boards ATSAM4CMx-DB. The metrology board works as a host controller connected to a G3-PLC board working as a modem. The communication between host and modem is established by means of a serial port. The Microchip Universal Serial Interface (USI) is used to serialize the components and services described in the G3-PLC specification.

The purpose of this application note is to explain how to integrate the metrology code of the ATSAM4CMx-DB boards with the G3-PLC, USI, IPv6 and DLMS components. An explanation of the tasks to be performed by the MCU is provided, analyzing priorities and critical actions. The application note includes sample code.

Table of Contents

1. Introduction

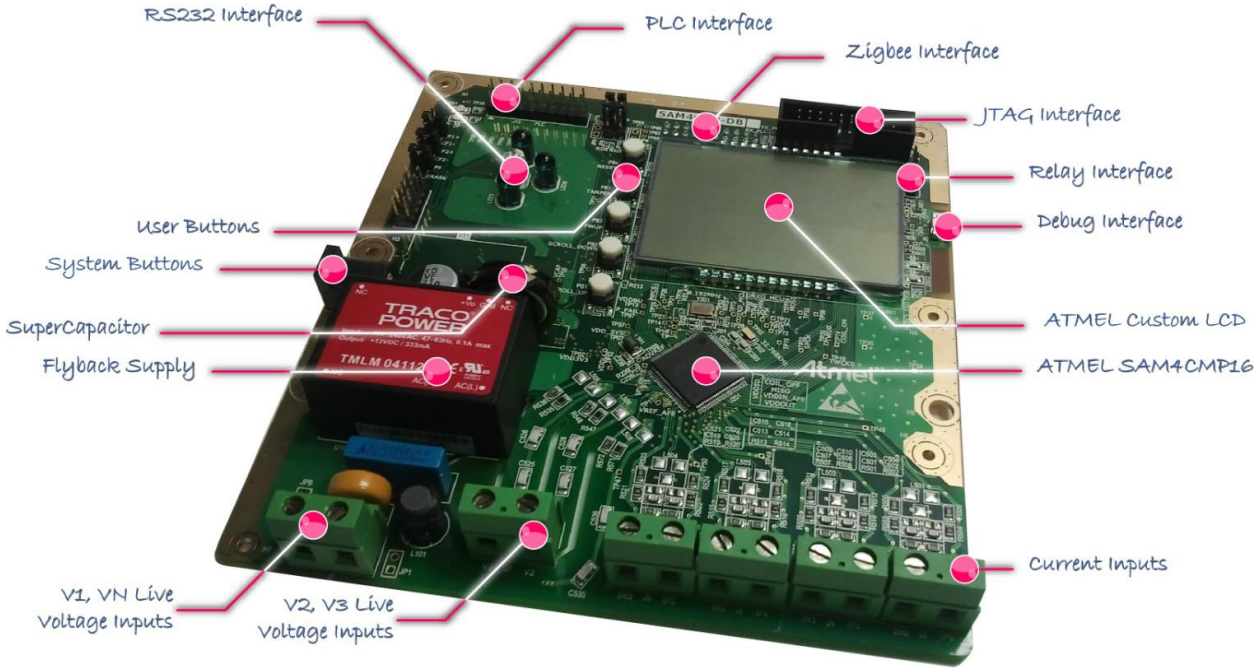
The ATSAM4CMS-DB and ATSAM4CMP-DB kits are meter demonstration boards for the 32-bit ARM® Cortex®-M4 SAM4CMx16C microcontrollers from Microchip:

- ATSAM4CMS-DB—for device SAM4CMS16C
- ATSAM4CMP-DB—for device SAM4CMP16C

Target demonstrations:

- Metrology performance verification
- Dual-Core ARM Cortex-M4 solution for meter firmware integration

Figure 1-1. ATSAM4CMx-DB Board



This application note shows how to add G3-PLC communications to the meter demonstration boards. The metrology board works as a host controller connected to a G3-PLC board employed as a modem. The communication between host and modem is established by means of a serial port. The Microchip USI is used to serialize the components and services described in the G3-PLC specification.

These are the new features added to the meter firmware:

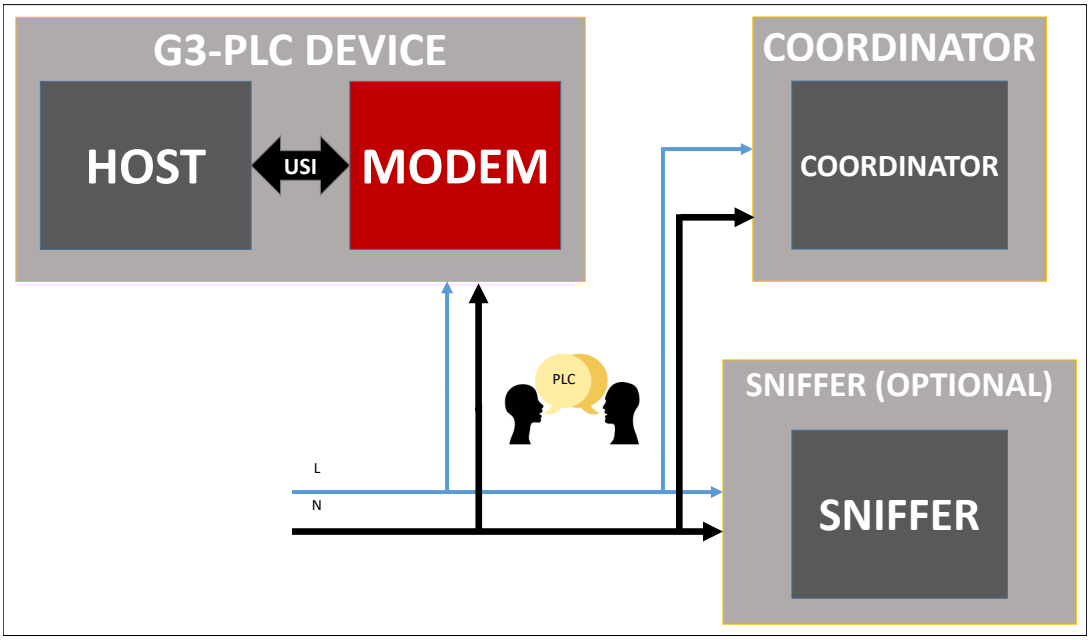
- Communications with the G3-PLC modem
- Registering of the device in a G3-PLC network
- IPv6 stack integration
- An example showing how to return metrology data using DLMS

2. Hardware Requirements

2.1 Overview

A G3-PLC network is composed of a device, a coordinator and an optional sniffer. It's possible to add more devices to the network.

Figure 2-1. G3-PLC Network Scheme



2.2 Device (Host + Modem)

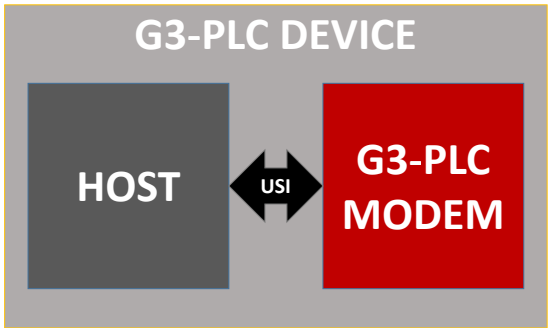
The G3-PLC device is composed of a host and a G3-PLC modem. The host is implemented in the SAM4CMx MCU that is included in the ATSAM4CMx-DB. There are several Microchip demonstration boards which can act as G3-PLC modems, for example:

- ATPL250AMB (ATPL250-EK)
- PL360MB (ATPL360-EK)
- SAM4CP16CMB (ATSAM4CP16C-EK)
- ATPL250ABN (ATPANCOORDINATOR-EK)
- PL360G55CF-EK
- PL360G55CB-EK

The host and modem are connected by a serial port:

HOST	MODEM
TX	RX
RX	TX
GND	GND

Figure 2-2. Serialized G3-PLC Device on ATSAM4CMx-DB



2.3 Coordinator

In order to establish a G3-PLC network, a coordinator is needed. There are several Microchip demonstration boards that can act as G3 PLC coordinator, for example:

- ATPL250AMB (ATPL250-EK)
- PL360MB (ATPL360-EK)
- SAM4CP16CMB (ATSAM4CP16C-EK)
- ATPL250ABN (ATPANCOORDINATOR-EK)
- PL360G55CF-EK
- PL360G55CB-EK

2.4 Sniffer

This is an optional element. An additional G3-PLC board can be programmed as a sniffer, in order to watch the network traffic. The PC software “Microchip PLC Sniffer” is required. There are several Microchip demonstration boards that can act as G3-PLC sniffer, for example:

- ATPL250AMB (ATPL250-EK)
- PL360MB (ATPL360-EK)
- SAM4CP16CMB (ATSAM4CP16C-EK)
- ATPL250ABN (ATPANCOORDINATOR-EK)
- PL360G55CF-EK
- PL360G55CB-EK

3. Firmware Loading

The modem, coordinator and sniffer firmware are included in the standard G3-PLC firmware stack developed by Microchip. Please program the boards with the following firmware:

3.1 Modem

The ADP and MAC serialization is an application example that brings access to the ADP, MAC and Boot Strapping (BS) API through a serial connection. This application is useful to perform intensive tests of the stack or run the upper layers in a separated CPU.

The application allows control of the G3-PLC stack at different levels, by making use of the ADP API (standard access) or directly accessing the MAC API as a shortcut for some tests. Serialization is also available to the provided bootstrap module to control the bootstrap phase on the coordinator side, if needed.

- The example project is located in folder *thirdparty\g3\apps**adp_mac_serialized_app***. Please compile the project and program the board to be used as modem.

3.2 Coordinator

The DLMS application example shows how the G3-PLC API along with an IPv6 stack should be used by a typical DLMS application. The coordinator will perform a continuous data collection to generate data traffic in the G3-PLC network.

The example is located in folder: *thirdparty\g3\apps**dlms_app_coord***. Please compile the project and program the board to be used as coordinator. This example can be configured to include sniffer capabilities.

3.3 Sniffer

The PHY Sniffer is an application example that uses the PHY layer to monitor PLC frames in the G3-PLC network and then sends them via USI serialization to a PC tool (PLC Sniffer Tool), which has to be installed in the user's host PC.

The example configures UART0 at 230400bps by default, but it is possible to change the serial port settings modifying the file *conf_usi.h*.

The example is located in the following paths:

- ATPL250 transceiver: *thirdparty\g3\phy \atpl250\apps**phy_sniffer_tool***
- ATPL360 transceiver: *thirdparty\g3\phy \atpl360\apps**phy_sniffer_tool***

Please compile the project and program the board to be used as sniffer.

3.4 Host

This is the firmware to be developed through this application note.

4. Host Application Baseline Firmware

4.1 Overview

The host application is based on the ATSAM4CMx-DB firmware sample code and with the modifications explained in this document, the board will not only work as a metering board but also as a host controller capable of communicating with a G3-PLC modem. A DLMS example is provided, showing how to send metrology data by DLMS. Several key firmware packages have to be integrated:

- USI Host
- ADP management
- DLMS and IPv6

4.2 SAM4CMx-DB Application Firmware Sample Code

The starting point is the firmware developed for the ATSAM4CMx-DB. The sample code has been developed taking as a baseline the 1.43 version. Please contact SmartMeterSupport@microchip.com to check for Smart Meter Project updates. The project files can be found in the following path: *\SAM4CM_V1.43\Firmware\Application\Source Code*.

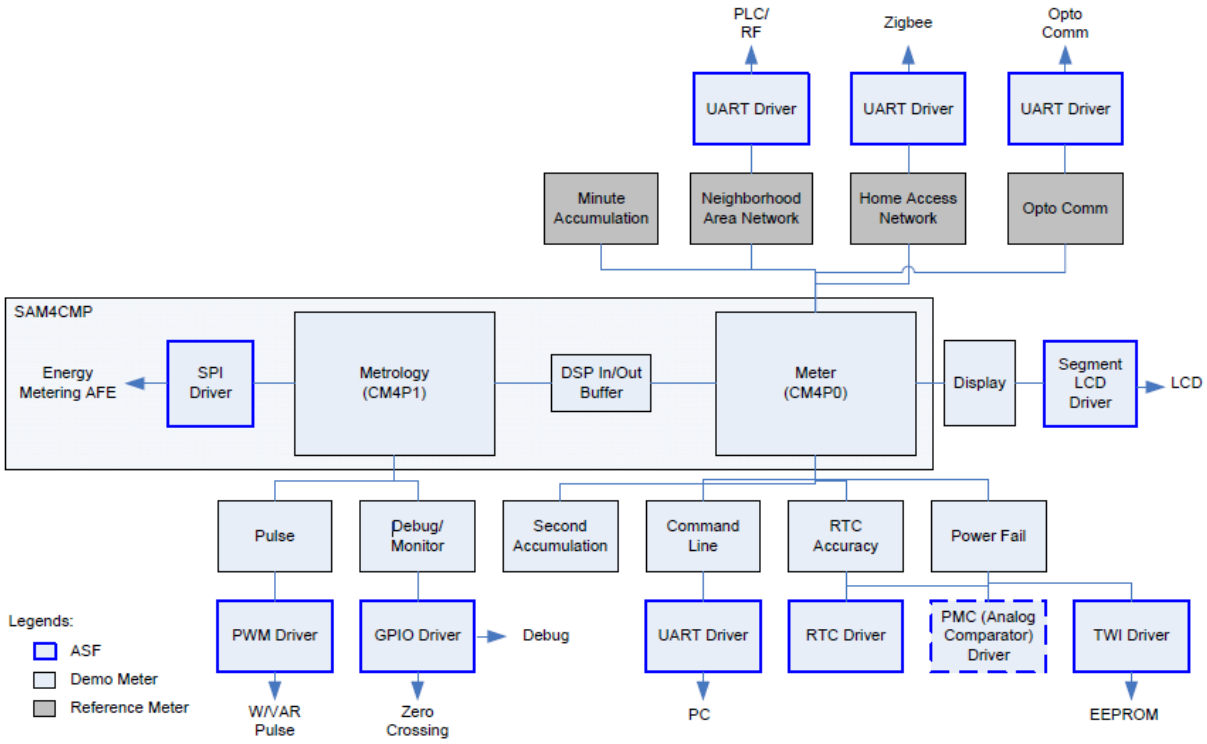
The Smart Meter Project package also contains the following documentation and help files:

Item	Location
Demo Board Hardware User Guide	*SAM4CM_V1.43\Demo Board\ SAM4CM HW user guide.pdf*
Demo Board User Guide	*SAM4CM_V1.43\Demo Board\ SAM4CMx-Metering Demo User Guide.pdf*
Firmware Metrology Datasheet	*SAM4CM_V1.43\Firmware\Metrology\Metrology Datasheet\ Atmel_46005_SE_SAM4CMP16_Metrology_Datasheet_060514.pdf*
Firmware Metrology User Guide	*SAM4CM_V1.43\Firmware\Metrology\Metrology Datasheet\ Atmel_46204_SE_SAM4CMP16_Metrology_User_Guide_060514.pdf*
Firmware Application Developer Guide	*SAM4CM_V1.43\Firmware\Application\Developer guide\ SAM4CMx-Developer Guide.pdf*

The meter demo application implements a simple electric meter, in which Energy Metering AFE collects the data from current sensors and digitizes voltage and current data for further processing.

The meter demo application is an interrupt-driven firmware with event flags to schedule different processing types such as second processing, terminal user command processing, etc. The application runs on top of the Atmel Studio Framework (ASF - device driver, services and components). Therefore, the low level firmware driver handles the hardware setup and controls for both metrology firmware and meter firmware.

Figure 4-1. Meter Demo Firmware Block Diagram



In the metrology processor (CM4P1), metrology firmware collects the momentary interval raw data on every sample interval 16 kHz (62.5 µs). Metrology processor filters and decimates data to 4 kHz (0.25 ms) data rate for use in integration and calculation of all output data quantities updated at a rate by default set to 1 Hz.

The communications should not interfere with the data collection operation, which is the critical task. The processor should collect the momentary data before the new data arrive.

5. Host Application Development

5.1 Step 1: Debug Tips

In order to simplify the debug process, some modifications have been included in the baseline code:

5.1.1 Default Dummy Handler

The default dummy handler has been modified (file *startup_sam4cm.c*) to have more information when an exception handler happens.

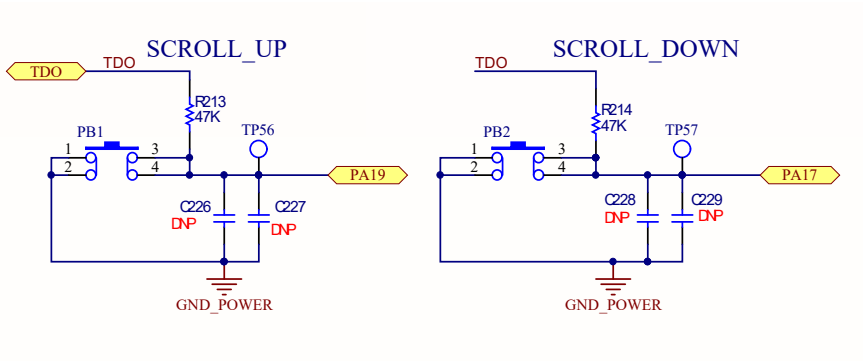
5.1.2 Watchdog Initialization

Watchdog debug halt (WDT_MR_WDDBGHLT) is activated, in order to stop it when the system halts in debug state.

5.1.3 SCROLL UP and SCROLL DOWN Buttons

TDO line is connected to SCROLL UP and DOWN buttons.

Figure 5-1. Scroll Up and Down Schematics



When debugging with JTAG, TDO/RTC_1HZ/RTCOUT0 line activates SCROLL_UP and SCROLL_DOWN buttons' interrupts. To avoid this unexpected activation, SCROLL_UP and DOWN interrupts have been disabled. This behavior is enabled by means of the pre-processor directive `#define SCROLL_DISABLED` (*conf_board.h*).

5.2 Step 2: Adding USI Serialization

5.2.1 Host and Modem Interconnection

The host communicates with the modem through USI, so a free serial port is needed. The ATSAM4CMx-DB only has an available serial port, UART0, attached to PB4 y PB5.

The physical connection between the host and modem serial ports must be done in the following way, using a cable:

HOST	MODEM
TX – PB5	TX
RX – PB4	RX
GND	GND

⚠WARNING Please verify the modem configuration in order to know where to connect the modem wires. The settings are located in the *conf_usi.h* file of the `adp_mac_serialized_app` project: *thirdparty\g3\apps* ***adp_mac_serialized_app***. Example selecting UART1 for USI:

```
#define USI_PORT_1      1
/* Port Communications configuration */
#define NUM_PORTS       1
#define PORT_1 CONF_PORT(UART_TYPE, USI_PORT_1, 230400, TX_UART_BUF0_SIZE, RX_UART_BUF0_SIZE)
```

5.2.2 UART0 Multiplexing

UART0 is used by the terminal command interface. This interface provides access to the Microchip metrology library, meter parameters configuration, measurement results and line voltage status.

The evaluation board includes a multiplexer connected to UART0 and controlled by the microcontroller (PB15 GPIO). There are two possible connections:

- UART0 is redirected to the USB interface J9 (by default). This is the interface used by the terminal. It includes an isolator
- UART0 is redirected to the PLC interface J13

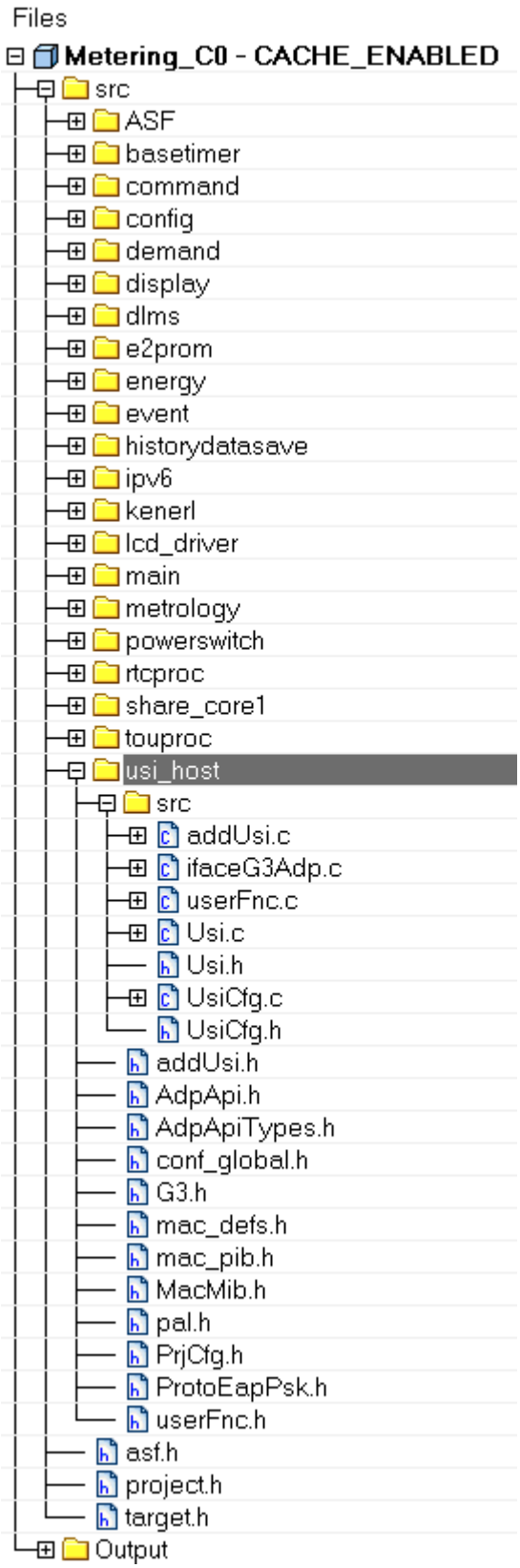
So the sample code includes a simple mechanism to select how the multiplexer is configured once the ATSAM4CMx-DB is switched on:

- The system connects UART0 by default to the PLC interface (USI connection)
- The system connects UART0 to the USB interface if there is a jumper joining pins 3 and 4 of J10 (relay interface)

5.2.3 Adding USI Files

The USI drivers are added to the project (usi_host folder).

Figure 5-2. USI Host Inclusion on IAR Project



Drivers to manage the UART0 and timers used by USI are included.

PrjCfg.h file contains the USI configuration. This file allows defining and configuring the ports to be used. This is the configuration selected for our application:

```
#define NUM_PORTS                                           1
#define PORT_0 CONF_PORT(UART_TYPE, 0, 230400, 512, 512)
#define NUM_PROTOCOLS                                       1
//#define USE_MNGP_PRIME_PORT                               0
//#define USE_PROTOCOL_SNIF_PRIME_PORT                      0
//#define USE_PROTOCOL_PRIME_API                            0
#define USE_PROTOCOL_ADP_G3_PORT                            0
//#define USE_PROTOCOL_COORD_G3_PORT                        0
//#define USE_PROTOCOL_SNIF_G3_PORT                         0
//#define USE_PROTOCOL_MAC_G3_PORT                          0
```

Please ensure that the baud rate of the modem is the same as the host's.

5.3 Step 3: Adding ADP Management Functions

5.3.1 Task System Overview

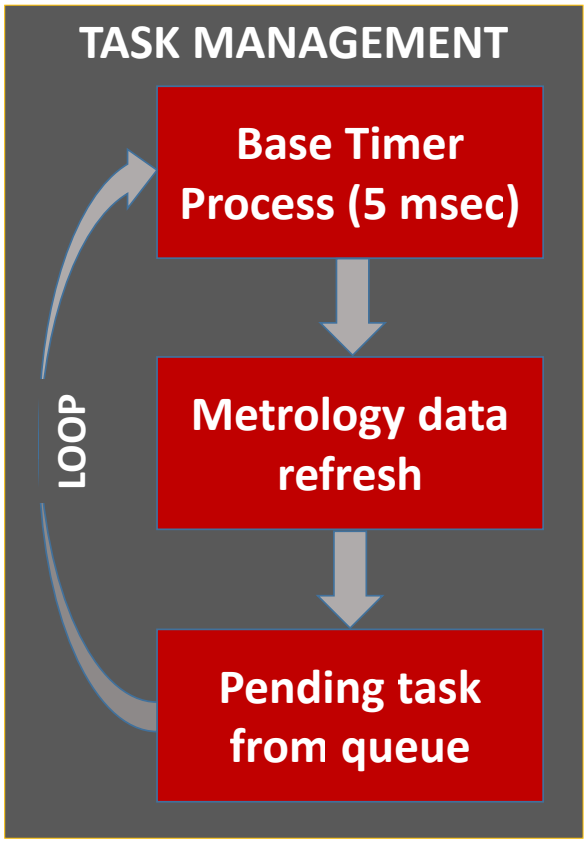
Now it's time to develop and integrate the ADP management task into the baseline standard metrology firmware of ATSAM4CMx-DB. The meter has three operating modes:

- Normal: The meter firmware measures V, I, energy and monitors line voltage input and meter hardware. It processes user commands and updates the LCD display as required. The meter firmware transitions to battery mode if the power failed and battery voltage is higher than the minimum threshold.
- Battery: the meter wakes up and meter firmware turns on the LCD display every 6 seconds when the user pushes the PB2 (configured as WKUP7 Input) or PB3 (configured as WKUP0Input) buttons. Then, it turns off meter power (sleep). When the power is supplied it will reset the MCU.
- Restart: the meter firmware activates an external reset.

In the provided example, the ADP management task is only executed in Normal mode, so it has to be integrated into the Normal mode task system. It's important to understand how the tasks are executed, so let's analyze briefly how the ATSAM4CMx-DB firmware manages the task system in Normal mode:

- Every 5 milliseconds the "BaseTimerProcess" is executed. This procedure periodically loads tasks into the task queue.
- New incoming metrology data from Core 1 is processed as soon as it's obtained. **This is a critical task, because current data has to be processed before new data arrives**. The frequency of arrival of new data depends on the user configuration.
- If there are pending tasks in the task queue, one task is executed. Once executed, the next one will be processed after checking "BaseTimerProcess" event and new metrology data available. The tasks loaded in the queue are related to RTC, TOU, display, demand, serial communications, energy processing, calibration, history data save, harmonic analysis, etc.

Figure 5-3. Task Management Loop



5.3.2 New Tasks Definition

Three new tasks are defined (*task.h* file):

- AdpInit: Initialization of ADP management
- AdpProc: ADP management processing
- UsiProc: USI processing

The new tasks are executed from the task queue.

USI processing is needed because ADP management communicates with the modem by means of USI. “BaseTimerProcess” loads USIProc task into the queue every 5 milliseconds. Since this task is not critical, the frequency of execution can be lowered. The decision of lowering the frequency must be taken by the firmware designer, depending on the resources used by the application firmware.

5.3.3 State Machines and Synchronous Communications

In the host-modem architecture the application layer is located in the host and the ADP and MAC layers are located in the modem. The communication between them is done through the USI, a serial interface, so accessing the information takes more time than it would take if both application and G3-PLC stack were running in the same MCU.

Since the critical task is to collect the metrology data, it’s not possible to wait for a long time for an answer to be received from the modem. In other words, as the host manages critical processes, like the metrology data capture from Core 1, Core 0 cannot be blocked waiting for a response from the modem.

This blocking could happen when a synchronous communication takes place. To avoid such situation the firmware example includes state machines, and splits the tasks into several sub-tasks.

5.3.4 ADP Initialization Process

The ADP initialization task is implemented as a state machine. The first time the task is loaded into the queue is at the meter initialization. Then, the system manages the reload mechanism. This mechanism is composed of two reload types:

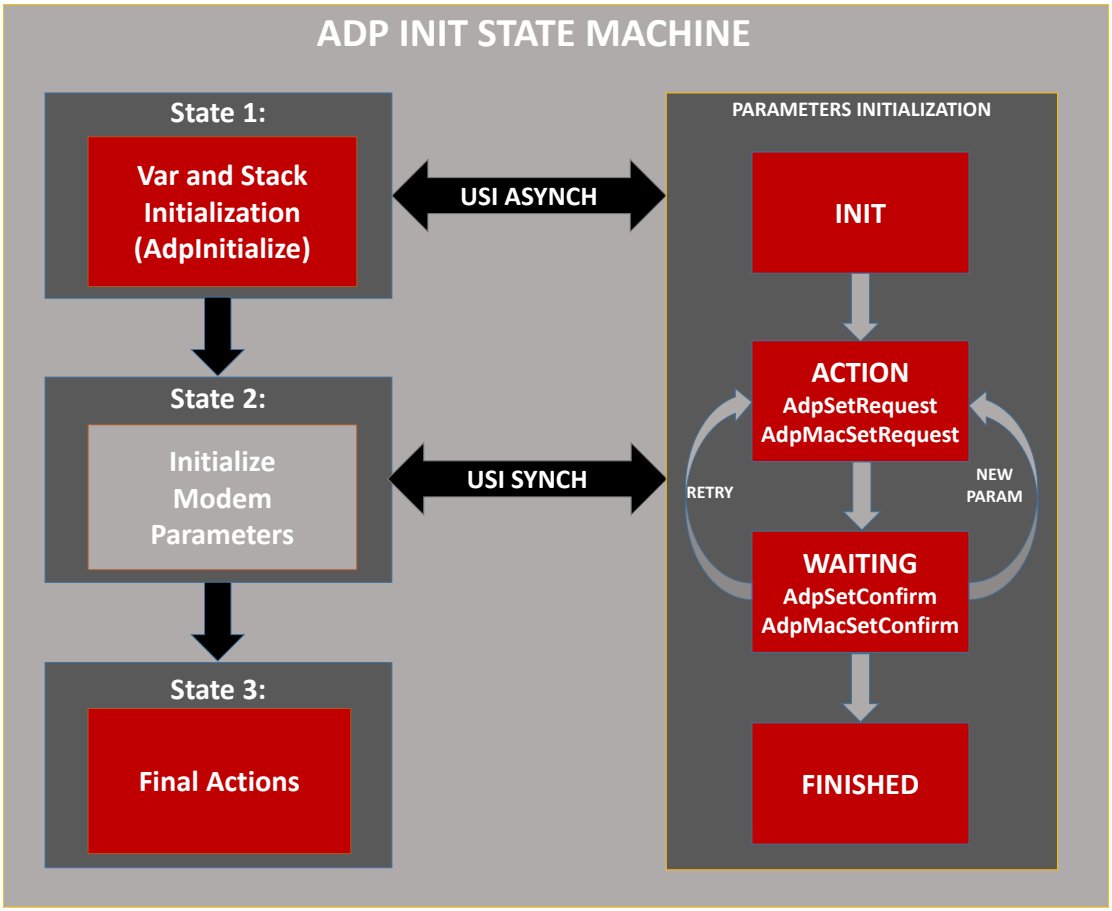
- Direct reload: The task loads itself into the queue. The reloading is accomplished by means of the “PutTaskIntoQue(AdpInit)” statement
- Indirect reload: Another process reloads “AdpInit” task into the queue

Direct reload is useful to fragment the number of actions to be performed consecutively. Since the tasks queue only runs one task per execution cycle, the base timer and metrology processes can be executed more frequently.

Indirect reload is useful to ensure the task is reloaded with a timeout, or after executing other tasks. For example: when the host is waiting for the modem to answer, the reload is done by the base timer process every 5 milliseconds. Additionally, before loading “AdpInit” task, “USIproc” is loaded, because the answer to the host will not arrive until USIproc is executed.

This following diagram shows the state machine implemented in the sample code:

Figure 5-4. ADP Init State Machine



State 1 includes a USI asynchronous communication: “AdpInitialize”.

State 2 calls another state machine, responsible for initializing the modem parameters. When it finishes, state 3 executes final actions (in the sample code, it queues “DlmsInit” task).

The parameters initialization procedure is a good example to understand how to implement a synchronous USI communication. In this case, a set of synchronous communications are done, because each parameter requires a USI command to be initialized. The following states are considered:

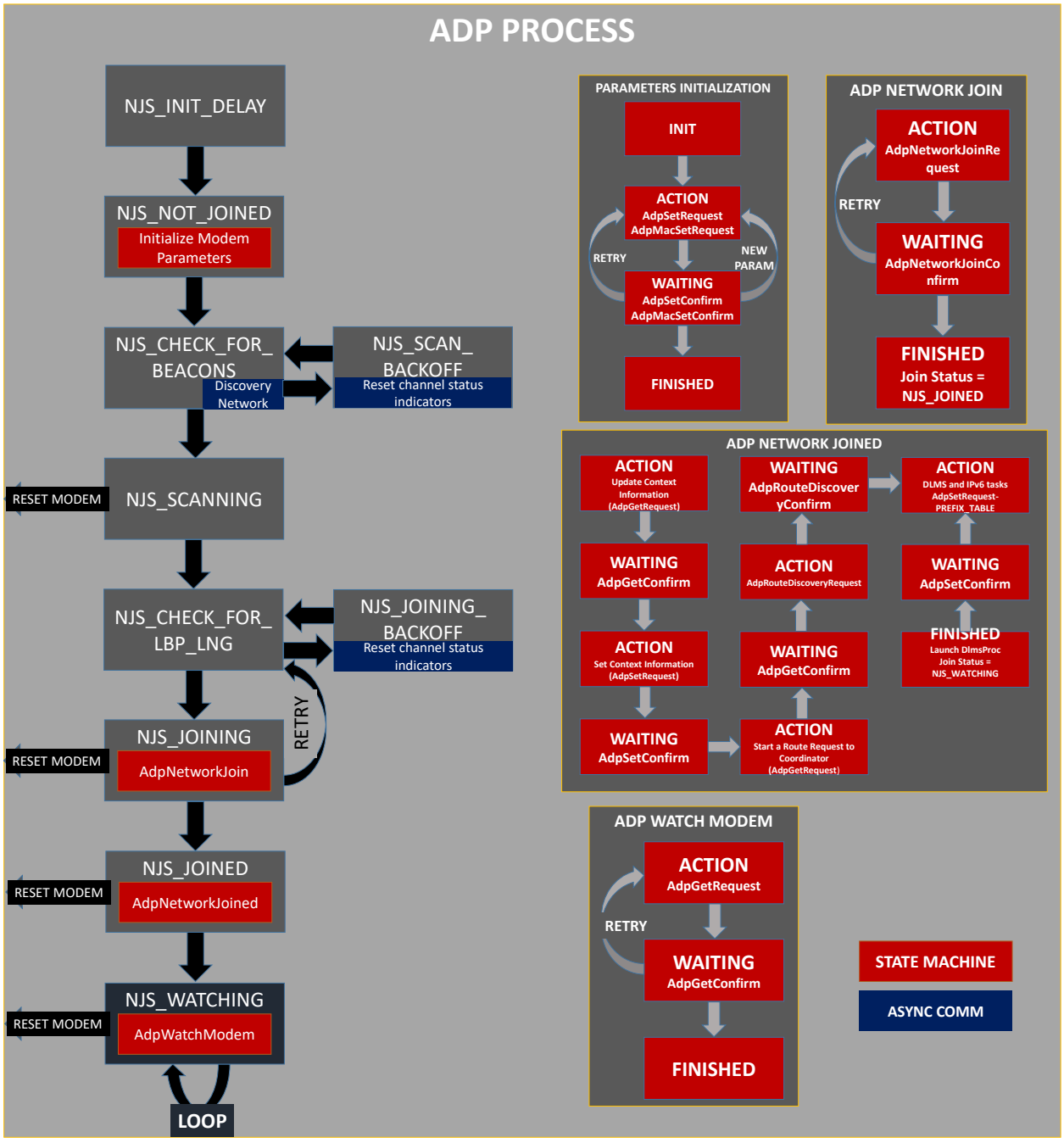
- “Init”: initialization of the variable responsible for managing the number of times a communication is retried, in case it fails

- "Action": the USI command is launched. A timeout is defined
- "Waiting":
  - If the answer has been received, the machine switches to "Action" state, in order to launch the following USI communication. If there are no commands remaining, the machine goes to "Finished" state
  - Is the answer has not been received and the timeout has expired, a retry is performed. If the number of retries have been completed, the modem is reset
- "Finished"

5.3.5 ADP Process

The following diagram shows the execution flow for the ADP Process.

Figure 5-5. ADP Process Tasks



The ADP Process manages the G3-PLC network joining mechanism via the Network Join State (NJS). The red boxes on the left denote state machines, and the blue ones asynchronous communications.

The main state machine, located at the left, is responsible for joining to the network. The steps to accomplish it are common to the G3-PLC stack Microchip examples. However, there are some new features added due to the host-modem configuration, which means that the device (composed of a host and a modem) is built in two independent electronic boards:

- There are several “RESET MODEM” transitions, which take place when communication with the modem fails, most of the times after several retries. This behavior has been included to prevent errors which could happen, for example, if the modem switches off temporarily and the host remains powered. In this case, switching to “NJS_INIT_DELAY” state is not enough and the application must launch “AdpInit” task in order to properly initialize the modem.
- “NJS _WATCHING” has been created to show how to include a simple procedure to check if the communication with the modem is working properly. The host sends a command to the modem. If the modem does not answer, the application must launch “AdpInit” task. “AdpWatchModem” is launched periodically, at low frequency. This state is useful when an abnormal situation happens to the modem (for example, if it's temporarily switched off). The frequency depends on the application. This state could be removed depending on the application features (for example, it will not be needed in case the device receives frequent or periodical communications).

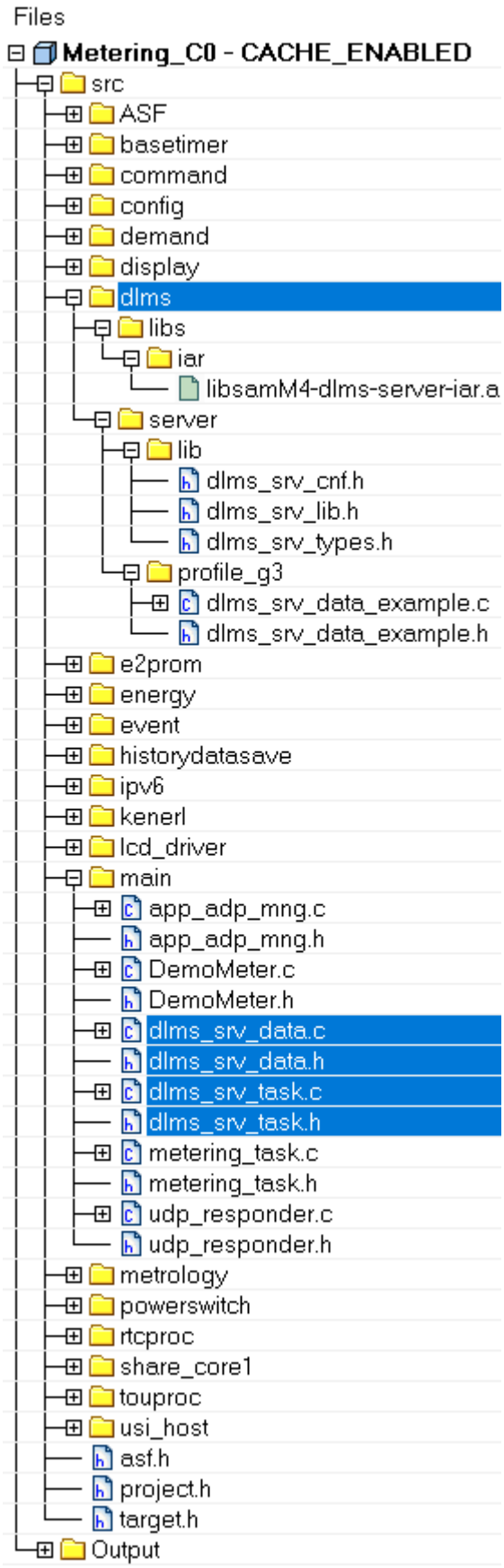
The main state machine calls several auxiliary state machines, as can be seen in the diagram.

5.4 Step 4: Adding DLMS Lite

5.4.1 DLMS Lite Files and Library

The “dlms” folder contains the DLMS Lite files and library. Additionally, some files have been added to the “main” folder.

Figure 5-6. DLMS Inclusion on IAR Project



The Microchip DLMS Lite library includes basic functionality for complete G3-PLC protocol stack testing. Please take into account that this library does not include a fully-featured DLMS implementation.

DLMS server library has two separate parts:

- A compiled library core to perform common DLMS operations: association management, headers processing, block reassembly, interface classes' common implementation, etc.
- The specific profile implementation module: *dlms_srv_data.c* and *dlms_srv_data.h*. This module is intended to be connected to the application, responsible for processing all the information related to the collected data.

Header files contained in the *dlms/server/lib* folder are related to the library core and should not be modified by the user.

The user application is responsible for association configuration, library initialization and OBIS objects configuration.

The project has its own dlms_srv_data files. These files contains the specific routine for each OBIS object. The request response payload is generated here. Within the library, OBIS code functions implementation examples are provided. By default *dlms_srv_data.c* functions wrap *dlms_srv_data_example.c* functions present in the library. OBIS implementation examples allow basic DLMS testing. Example functions return dummy data in a similar way a regular meter would do.

The following sections show how to modify the project to link the metrology data with DLMS objects, to return measured data.

5.4.2 IPv6

DLMS Lite is linked to the IPv6 convergence sublayer based on CycloneTCP from Oryx Embedded (http://www.oryx-embedded.com). All the required files are included in the "ipv6" folder.

5.4.3 New Tasks Definition

The following tasks are defined (*task.h* file):

- "DlmsInit": DLMS Initialization
- "DlmsProc": DLMS processing
- "NetInit": IPv6 Initialization
- "NetProc": IPv6 processing

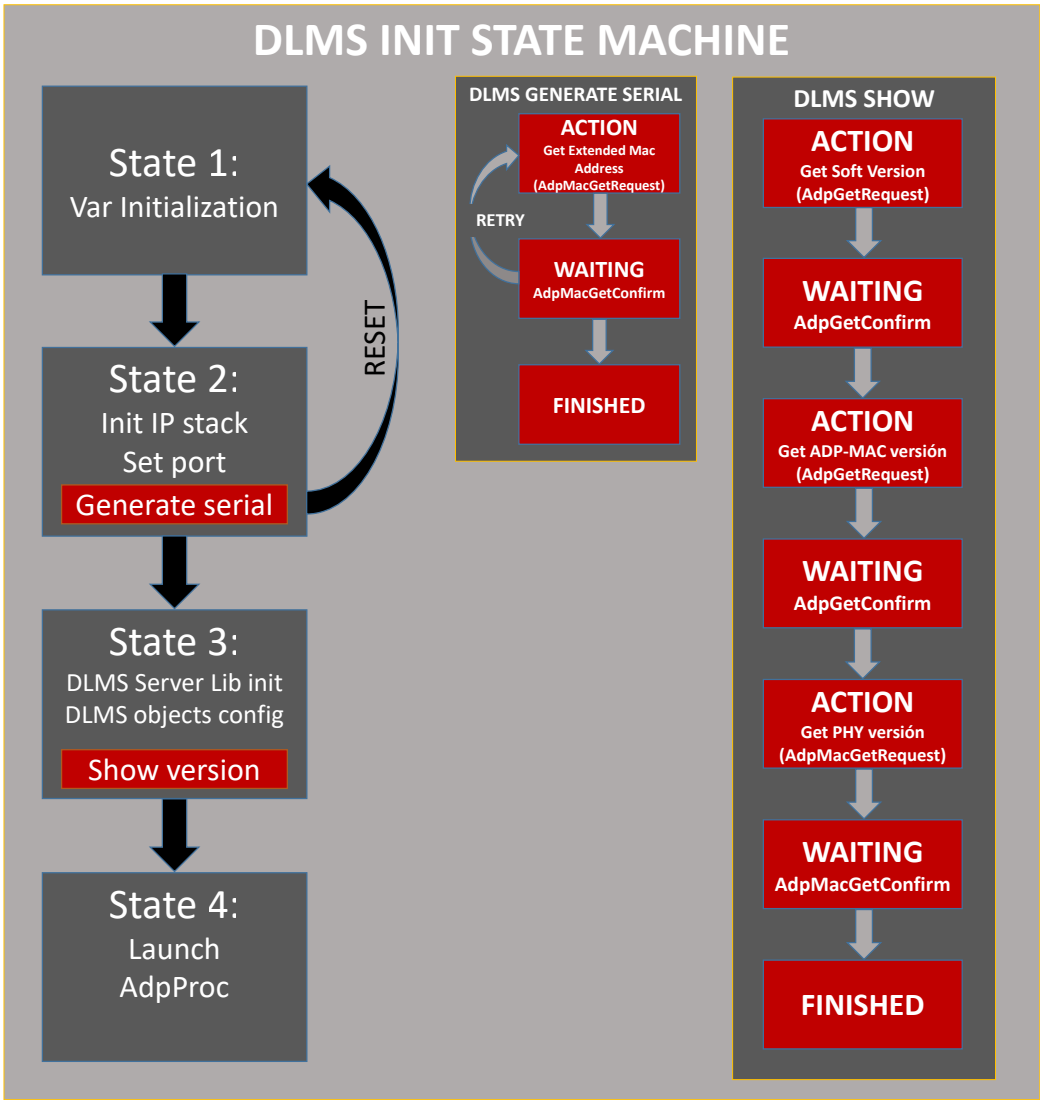
All these tasks are executed from the queue.

DlmsInit is launched by AdpInit once it finishes. DlmsProc is launched by AdpProc once it reaches NJS_WATCHING state; then, it is periodically launched every 5 milliseconds. NetInt is launched when the host initializes. Finally, NetProc is launched every 5 milliseconds.

5.4.4 DLMS Init

The initialization of DLMS Lite services is done by means of a state machine, as shown in the following diagram:

Figure 5-7. DLMS Init State Machine



Each OBIS code is declared in state 3. The declaration joins each OBIS code with its callback function.

There are two auxiliary state machines:

- "DLMS Generate Serial", which communicates with the modem in order to obtain the MAC manufacturer extended address.
- "Show version", responsible for getting from the mode the software, ADP/MAC, and PHY versions.

5.4.5 DLMS Process

The DLMS Process does not include synchronous communications and does not require a high processing time, so a state machine is not needed. The processing does not have strict time requirements but the user application must guarantee that it is called at least once any time a packet is received. Otherwise, the second packet will replace the previous one because reception is not buffered. Only one packet is received each time the DLMS Process function is called.

5.4.6 DLMS Object Implementation Example

This chapter shows how to implement a DLMS object, allowing the application to return data acquired from the metrology library.

As mentioned before, the DLMS default example code returns dummy data. This section shows how to implement the DLMS object 0-0:21.0.5.255.

This electricity related object is used to report instantaneous values of energy registers. The values to be returned are defined in the following table:

DLMS object 0-0:21.0.5.255			
Logical name	0000150005FF		
Capture objects	8,0-0:1.0.0.255,2	Clock	For single-phase meters, only one phase is active, so the values related to other phases (L2 and L3) have to be zero
	3,1-0:32.7.0.255,2	Voltage L1	
	3,1-0:31.7.0.255,2	Current L1	
	3,1-0:52.7.0.255,2	Voltage L2	
	3,1-0:51.7.0.255,2	Current L2	
	3,1-0:72.7.0.255,2	Voltage L3	
	3,1-0:71.7.0.255,2	Current L3	
	3,1-0:90.7.0.255,2	Current (sum over all three phases)	
	3,1-0:1.7.0.255,2	Active Power P+	
	3,1-0:2.7.0.255,2	Active Power P-	
	3,1-0:3.7.0.255,2	Reactive Power Q+	
	3,1-0:4.7.0.255,2	Reactive Power Q-	
	3,1-0:13.7.0.255,2	Power factor	

Step 1: Object Configuration

The object is configured in the DLMS initialization phase with `void dlms_init (void)` procedure (file *dlms_srv_task.c* file) in the sample code:

```
// "Eletricity related objects --> instantaneous values --> Instanteneous values"
dlms_srv_conf_obis(0, 0, 21, 0, 5, 255, 7, obis_0_0_21_0_5_255_cb, &x_new_obis);
```

Step 2: Callback Function Modification

The callback function (`obis_0_0_21_0_5_255_cb` in this example) is located in *dlms_srv_data.c* file. This function is linked to `obis_0_0_21_0_5_255_example_cb`, placed in *dlms_srv_data_example.c*. As it has been explained before, by default *dlms_srv_data.c* functions wrap *dlms_srv_data_example.c* functions present in the library. So now it's time to modify the default `obis_0_0_21_0_5_255_example_cb` function in order to include our customized code to return measured data instead of dummy values.

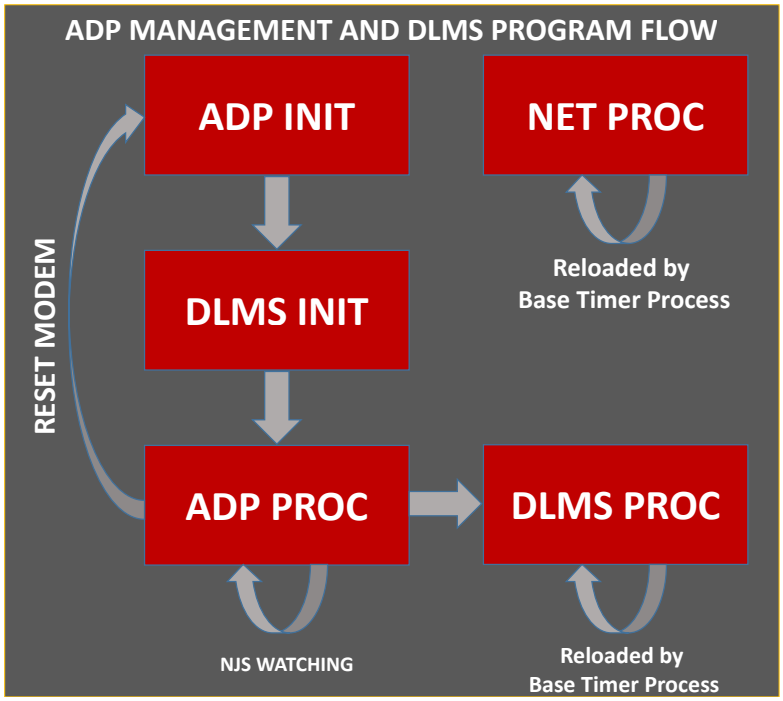
The sample code mixes dummy data (clock) and measured data. The instantaneous values returned are linked to `sx_meter_info` structure, which contains the data acquired by the metrology.

```
// Instantaneous Voltage L1
puc_resp_data[uc_resp_data_idx++] = DT_LONG_UNSIGNED;
puc_resp_data[uc_resp_data_idx++] = (sx_meter_info.Inst_V_rms_L1 >> 8) & 0xFF;
puc_resp_data[uc_resp_data_idx++] = sx_meter_info.Inst_V_rms_L1 & 0xFF;
```

5.5 Added Task Overview

The following diagram shows the new tasks flow.

Figure 5-8. ADP Management and DLMS Program Flow



6. Testing the Firmware

The Microchip G3-PLC stack release includes an example for the coordinator (`dlms_app_coord`), which performs a continuous data collection requesting 4 DLMS objects cyclically. The example is located in the following path: *thirdparty\g3\apps**dlms_app_coord***.

The DLMS objects requested by the coordinator are the following ones:

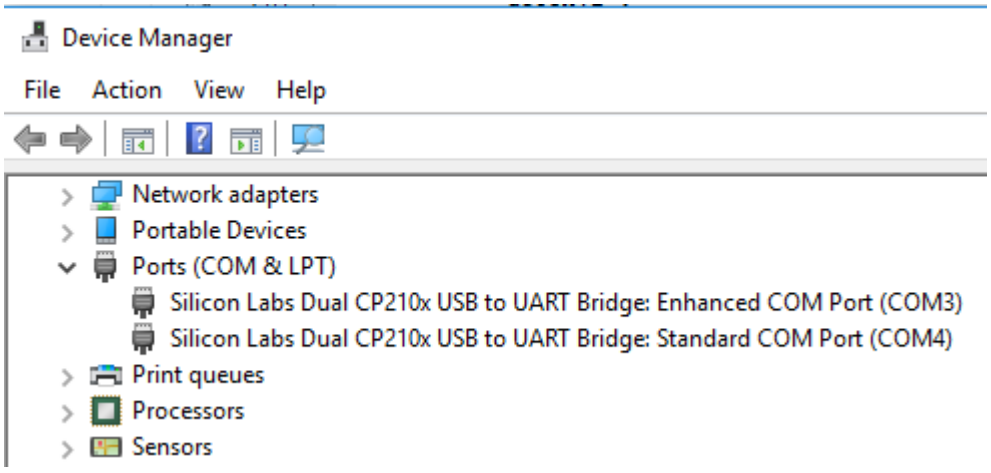
- {{0, 0, 1, 0, 0, 255}, 8, IC08_TIME} --> Clock
- {{1, 0, 99, 1, 0, 255}, 7, IC07_BUFFER} --> Load profiles
- {{0, 0, 29, 1, 0, 255}, 91, IC91_MAC_NEIGHBOUR_TABLE}
- {{0, 0, 29, 2, 0, 255}, 92, IC92_ADP_ROUTING_TABLE}

The coordinator has a serial console that reports the DLMS communication state. To access to the console please open a serial terminal configured in the following way:

- 921600 bps
- 1 stop bit
- No parity
- Flow control: none

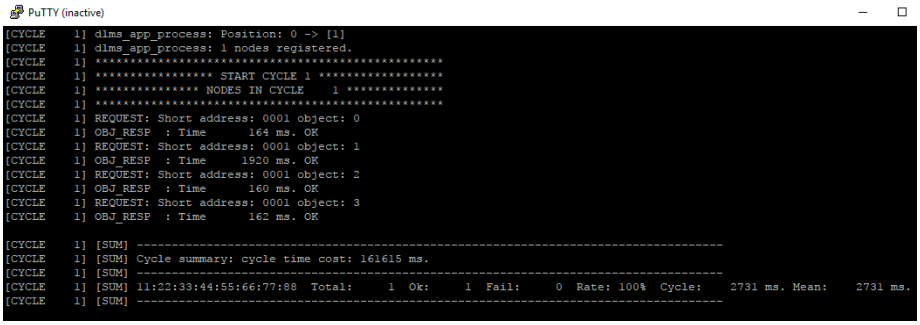
The serial port number can be obtained by means of the Windows “Device Manager”, “Ports” section. Use the “Standard COM port”, not the “Enhanced” one:

Figure 6-1. Serial Port on Device Manager



This is the information provided by the coordinator:

Figure 6-2. Console Running Application



It's possible to modify the coordinator project to retrieve the DLMS object implemented in the previous section. The easiest way is by replacing the callback function in one of the requested objects by the modified function `obis_0_0_21_0_5_255_example_cb` as follows:

```
data_access_result_t obis_0_0_1_0_0_255_cb(assoc_info_t *px_assoc_info, uint8_t uc_attr)
{
//return obis_0_0_1_0_0_255_example_cb(px_assoc_info, uc_attr);  // Enable this line to allow
the normal execution
return obis_0_0_21_0_5_255_example_cb(px_assoc_info, uc_attr);   // Enable this line to debug
the DLMS object
}
```

To verify that the function is called and returns the expected values, a breakpoint can be set in the callback function while debugging. If the execution stops there and the values are meaningful, then all the processes regarding metrology, G3-PLC network management, DLMS and IPv6 are working properly.

Figure 6-3. Debugging Application on IAR

7. Revision History

7.1 Rev A - 08/2019

Document	Initial document release.

The Microchip Website

Microchip provides online support via our website at http://www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to http://www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with

your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-5224-4946-1

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit http://www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: http://www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820