

Features

- Module LoRa SAMR34 V4.1 (Based on ATSAMR34J18B LoRa IC)
 - Cortex® M0+ CPU and LoRa Transceiver
 - Frequency Range: 862 to 928 MHz
 - 2-Kbit Serial EEPROM + Factory-Programmed EUI-48™ Address
 - On-Board 32MHz TXCO
 - U.FL RF connectors (MHF)
- Isolated RS-485 Interface
- Serial USB and Target USB Connector with ESD Protection
- 3 GPIO LEDs
- Manual Reset Button
- GPIO Button
- Board Supply (USB bus, external VIN 12VDC)

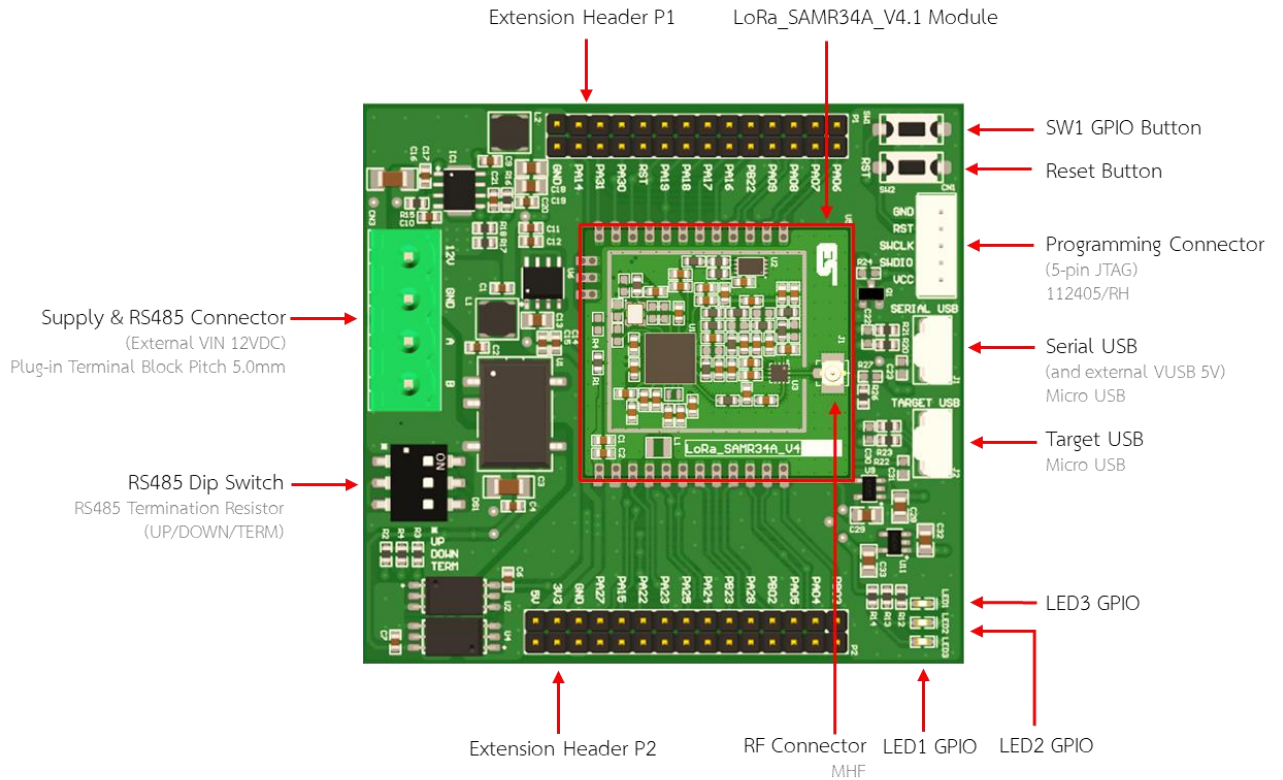
Table of Contents

1 Overview.....	2
1.1 Evaluation Board	2
1.2 Development environment and Tool.....	2
2 Hardware User Guide	3
2.1 Power Supply Diagram.....	3
2.2 Board Connectors	3
2.3 Peripherals	5
3 Example Demo Project.....	6
3.1 Setup	6
3.2 Application Configuration.....	7
3.3 Building/Flashing the Firmware	8
3.4 Demo Application Usage.....	10
4 Example Modbus Demo Project.....	13
4.1 Setup	13
4.2 Application Configuration.....	14
4.3 Demo Application Usage.....	14
5 Setting Project / Properties.....	15
5.1 EUI Address.....	15
5.2 Sleep Mode.....	16
5.3 Modbus.....	17
5.4 LEDs	18
5.5 Watchdog Timer	18
5.6 USB	18
6 Appendix	19
6.1 Board Dimensions	19
6.2 Board Layout.....	20
6.3 Schematic	20
6.4 Optional Accessories.....	23
7 Reference Links.....	23
8 Reversion History.....	23

1 Overview

1.1 Evaluation Board

LoRa SAMR34 EVA MODBUS V1.0 Board's overview



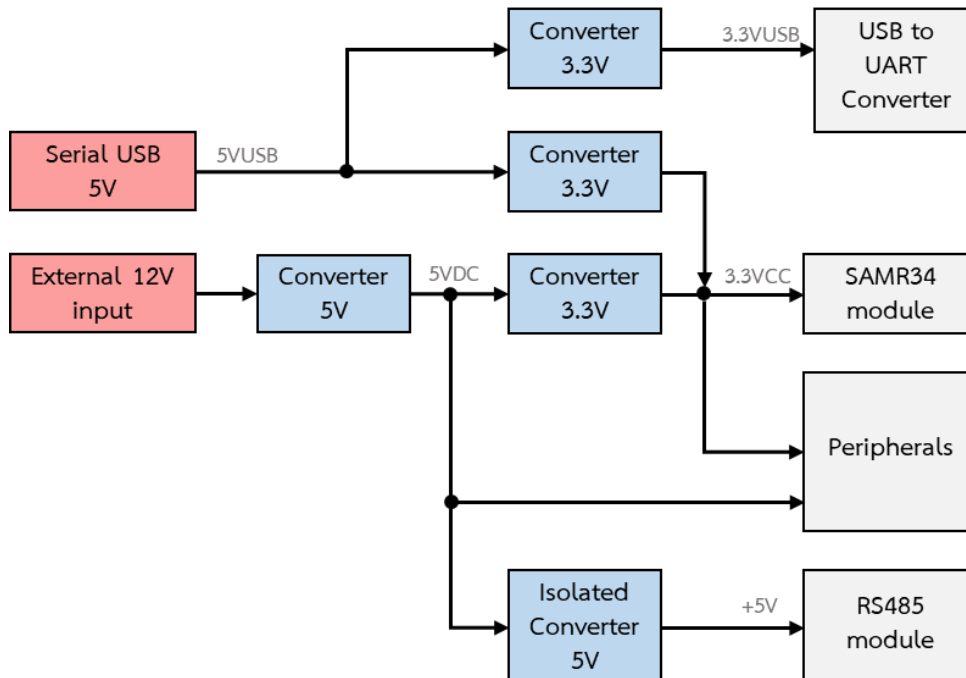
1.2 Development environment and Tool

- Microchip Studio
<https://www.microchip.com/en-us/development-tools-tools-and-software/microchip-studio-for-avr-and-sam-devices>
- ATMEL ICE
https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-ICE_UserGuide.pdf
- Terminal emulator software: Tera Term
<https://ttssh2.osdn.jp/>

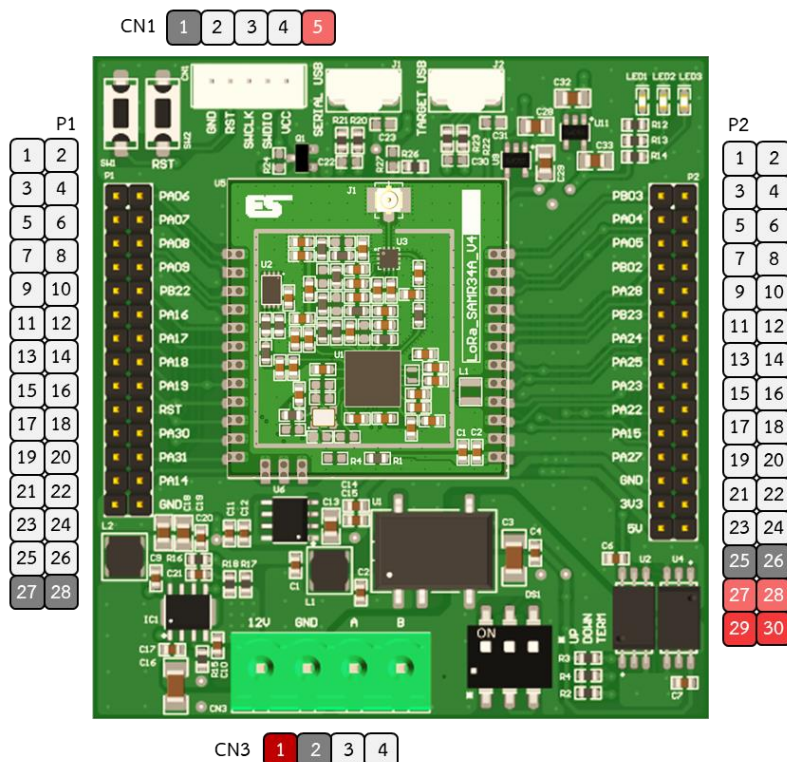
2 Hardware User Guide

2.1 Power Supply Diagram

The LoRa_SAMR34_EVA_MODBUS board can be powered by External 12V or Serial USB 5V



2.2 Board Connectors



Note : For more details on Pin descriptions, refer to [SAM R34/R35 Low Power LoRa Sub-GHz SiP Datasheet](#) (Table 5-1. Port Function Multiplexing)

2.2.1 Extension Header P1

Extension Header P1		Pins on LoRa_SAMR34 Module	Shared Functionality
Pin No.	Pin Name		
1, 2	PA06	PA06	-
3, 4	PA07	PA07	-
5, 6	PA08	PA08	UART_TX
7, 8	PA09	PA09	UART_RX
9, 10	PB22	PB22	-
11, 12	PA16	PA16	EDBG_I2C SDA ^[1]
13, 14	PA17	PA17	EDBG_I2C SCL ^[1]
15, 16	PA18	PA18	EXT2, USB_VBUS ^[3]
17, 18	PA19	PA19	EXT3, SW0
19, 20	RST	RESET*	RESET*
21, 22	PA30	SWCLK	SWCLK
23, 24	PA31	SWDIO	SWDIO
25, 26	PA14	PA14	TXCO_PWR ^[1]
27, 28	GND	GND	-

^[1] On board LoRa_SAMR34A_V4 module - refer [LoRa_SAMR34A_V4_DataSheet](#)(Schematic)

^[3] Optional

2.2.2 Extension Header P2

Extension Header P2		Pins on LoRa_SAMR34 Module	Shared Functionality
Pin No.	Pin Name		
1, 2	PB03	PB03	-
3, 4	PA04	PA04	UART_TX ^[2]
5, 6	PA05	PA05	UART_RX ^[2]
7, 8	PB02	PB02	-
9, 10	PA28	PA28	-
11, 12	PB23	PB23	-
13, 14	PA24	PA24	USB_DM
15, 16	PA25	PA25	USB_DP
17, 18	PA23	PA23	LED0
19, 20	PA22	PA22	LED1
21, 22	PA15	PA15	LED2
23, 24	PA27	PA27	-
25, 26	GND	GND	-
27, 28	3V3	-	3.3V Supply to peripherals
29, 30	5V	-	5V Supply to peripherals

^[2] RS485 UART pin – change this pin to General purpose I/O pin by remove R38, R39 on LoRa_SAMR34_EVA_MODBUS board

2.2.3 Programming Connector CN1

Pin No.	Pin Name	Description
1	GND	GND
2	RST	RESET*
3	SWCLK	SWCLK
4	SWDIO	SWDIO

2.2.4 Supply and RS485 Connector CN3

Pin No.	Pin Name	Description
1	GND	GND
2	12V	Supply Input 12VDC
3	A	RS485 A
4	B	RS485 B

2.3 Peripherals

2.3.1 LED Connections

Pin Name	Function	Shared Functionality
PA23	Yellow LED0	P2
PA22	Green LED1	P2
PA15	Red LED2	P2

2.3.2 Mechanical Buttons

Pin Name	Function	Shared Functionality
RESET*	RESET	P1, Programing connection
PA28	SW0	P2

2.3.3 Serial USB

Pin Name	Function	Shared Functionality
PA08 ^[4]	S1_UART TX	P1
PA09 ^[4]	S1_UART RX	P1

^[4] change this pin to General purpose I/O pin by remove R36(PA09), R37(PA08) on Evaluation board

2.3.4 Target USB

Pin Name	Function	Shared Functionality
PA25	USB DP	P2
PA24	USB DM	P2
PA18 ^[5]	VBUS EXI (Optional)	P2

^[5] change this pin to General purpose I/O pin by remove R25 on Evaluation board

2.3.5 UART RS485

Pin Name	Function	Shared Functionality
PA04 ^[6]	S0_UART TX	P2, RS485 Connection
PA05 ^[6]	S0_UART RX	P2, RS485 Connection

^[6] change this pin to General purpose I/O pin by remove R38(PA05), R39(PA04) on Evaluation board

3 Example Demo Project

This Application is used to send count payload number and example data payload through the LoRaWAN Network.

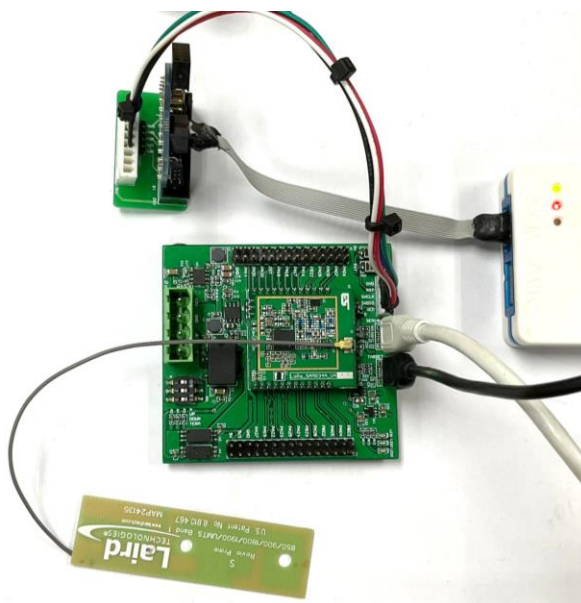
- Establish a LoRaWAN connection automatically between the SAMR34 end node and available Gateways that are within radio range.
- The UART is used to display the menu options, The user input is provided through keyboard.
- Periodically transmit a confirmed data frame (default periodic timer duration 10 seconds).

Define by **DEMO_CONF_APP_PERIODIC_TIMER_PERIOD_MS** in **conf_app.h**.

- Use pushbuttons **SW0** on the board for wake up from sleep mode and show app information via a USB Interface.
- LED1 shows LoRaWAN operating status.

3.1 Setup

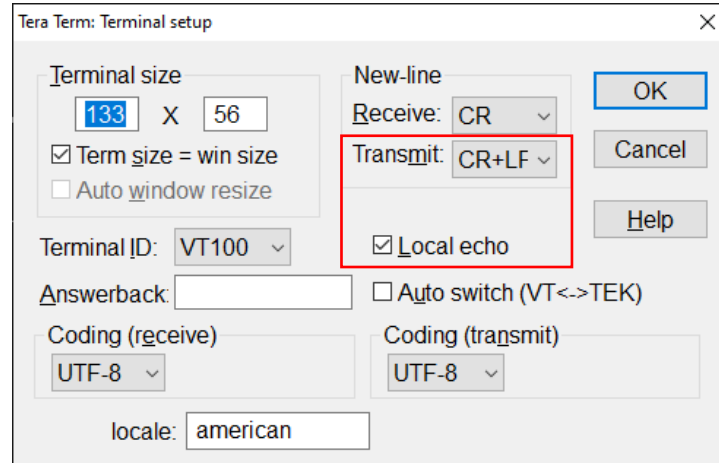
1. Download source code [LORAWAN_P6_MODBUS_DEMO](#), Open **Microchip Studio** and Select File > Open > Project/solution... . Select **..\LORAWAN_P6_MODBUS_DEMO_V1.atsln** and open.
2. The connection shown in figure.
 - Evaluation Board
 - ATMEL-ICE
 - Antenna
 - Micro USB cable (for Serial USB)
 - Micro USB cable (for Target USB) (Optional)



Note : For connector pin and details, see [1.1 Evaluation Board](#) and [2.2 Board Connectors](#)

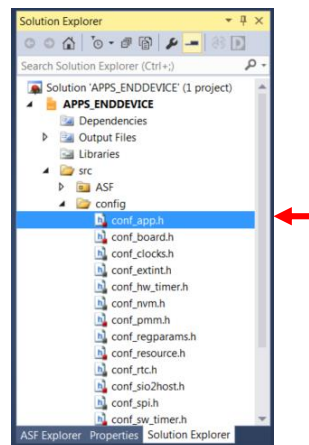
3. Open Terminal emulator software, Setting UART interface are as follows.
 - UART : 9600 bps 8N1
 - USB : 115200 bps 8N1

Note : for Tera Term, Setup > Terminal... set New Line Transmit as CR+LF.



3.2 Application Configuration

3.2.1 Define application configuration parameters in ..\src\config\conf_app.h



3.2.2 Activation Method and Type of Transmission

```
/* Activation method constants */
#define OVER_THE_AIR_ACTIVATION    LORAWAN_OTAA
#define ACTIVATION_BY_PERSONALIZATION    LORAWAN_ABP

/* Message Type constants */
#define UNCONFIRMED                LORAWAN_UNCNF
#define CONFIRMED                  LORAWAN_CNF

#define DEMO_APP_ACTIVATION_TYPE    OVER_THE_AIR_ACTIVATION
// #define DEMO_APP_ACTIVATION_TYPE    ACTIVATION_BY_PERSONALIZATION

/* Select the Type of Transmission - Confirmed(CNF) / Unconfirmed(UNCNF) */
// #define DEMO_APP_TRANSMISSION_TYPE    UNCONFIRMED
#define DEMO_APP_TRANSMISSION_TYPE    CONFIRMED
```

Activation Method

Type of Transmission

3.2.3 Device Class

```
/* Device Class - Class of the device (CLASS_A/CLASS_C) */
#define DEMO_APP_ENDDEVICE_CLASS    CLASS_A
// #define DEMO_APP_ENDDEVICE_CLASS    CLASS_C
```


3.2.4 Join Parameter

This application can set DevEUI or read form EUI-48 MAC Address. (see in [5.1 EUI Address](#))

```
/* ABP Join Parameters */
#define DEMO_DEVICE_ADDRESS          0xabcd0001
#define DEMO_APPLICATION_SESSION_KEY {0x10, 0x00, 0x20, 0x00, 0x30, 0x00, 0x40, 0x00, 0x50, 0x00, 0x60, 0x00, 0x70, 0x00, 0x80, 0x0A}
#define DEMO_NETWORK_SESSION_KEY    {0x10, 0x00, 0x20, 0x00, 0x30, 0x00, 0x40, 0x00, 0x50, 0x00, 0x60, 0x00, 0x70, 0x00, 0x80, 0x0A}

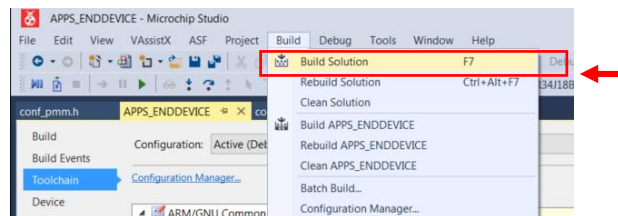
/* OTAA Join Parameters */
#define DEMO_DEVICE_EUI              {0x00, 0x01, 0x00, 0x00, 0x35, 0x8f, 0x00, 0x00}
#define DEMO_APPLICATION_EUI        {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x05}
#define DEMO_APPLICATION_KEY        {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x05}
```

3.2.5 Region Profile

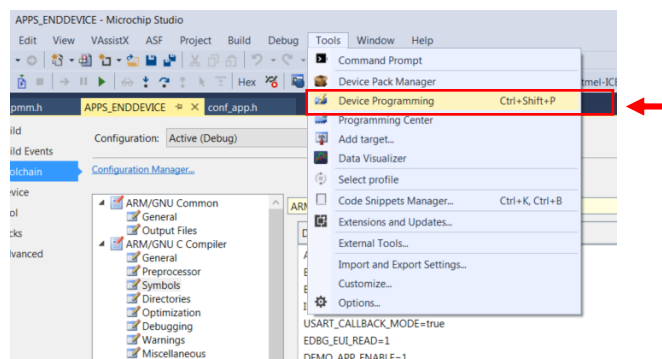
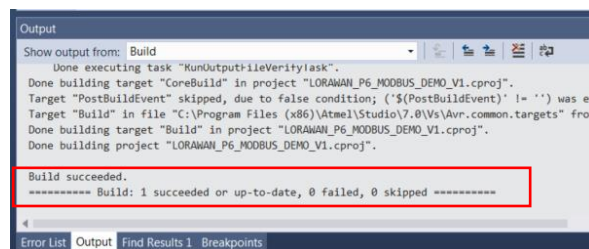
```
/* ISM Band Types */
#define DEMO_APP_ENDDEVICE_BAND      ISM_THAI923 /*Thailand*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_EU868 /* EU 863 - 870MHz ISM Band */
// #define DEMO_APP_ENDDEVICE_BAND  ISM_EU433 /* EU 433MHz ISM Band */
// #define DEMO_APP_ENDDEVICE_BAND  ISM_NA915 /*AU*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_AU915 /*NA*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_KR920 /*KR*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_JPN923 /*Japan*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_BRN923, /*Brunei*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_CMB923 /*Cambodia*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_INS923 /*Indonesia*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_LAOS923 /*Laos*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_NZ923 /*New Zealand*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_SP923 /*Singapore*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_TW923 /*Taiwan*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_THAI923 /*Thailand*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_VTM923 /*Vietnam*/
// #define DEMO_APP_ENDDEVICE_BAND  ISM_IND865 /*India*/
```

3.3 Building/Flashing the Firmware

- Go to Build > Build Solution



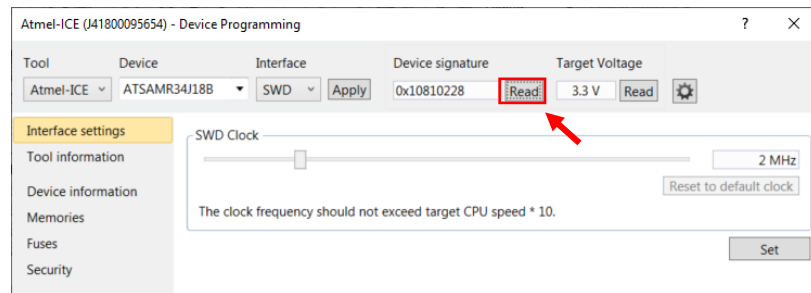
- After Output pane shown **Build succeeded**, go to Tool > Device Programming



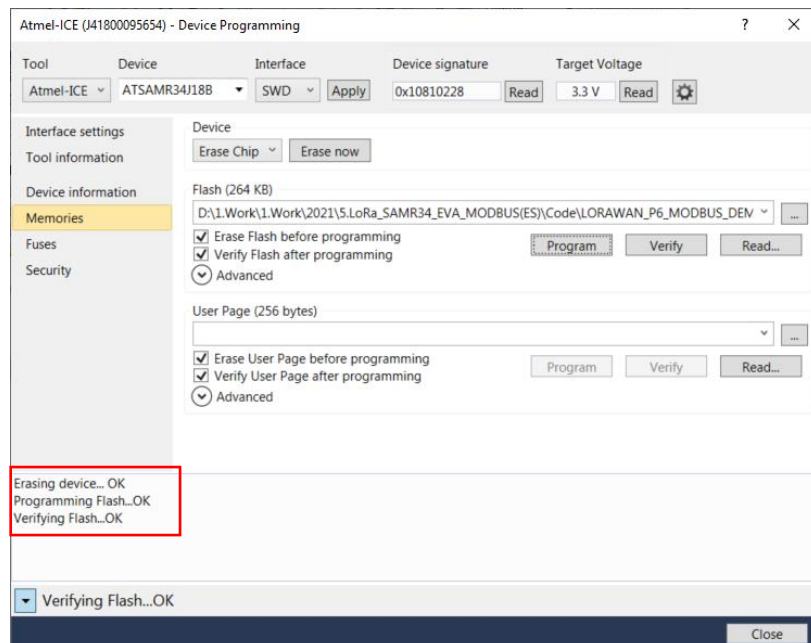
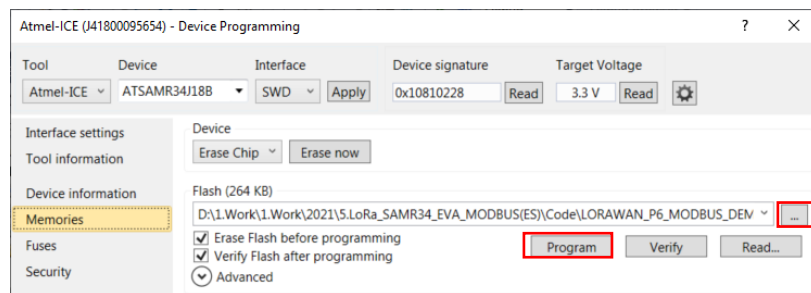
- From the Tool list, select Atmel-ICE



- Select Device ATSAMR34J18B, click Read at Device signature



- Click Memories, In the Flash pane, Browse for the ..\Debug\LORAWAN_P6_MODBUS_DEMO_V1.elf or ..\Debug\LORAWAN_P6_MODBUS_DEMO_V1.hex, then click Program



3.4 Demo Application Usage

Terminal emulator software is shown in the following figure.

```
Last reset causer: External Reset
Microchip LoRaWAN Stack - MLS SDK_1_0_P_6
Board Name - SAMR34_EVA_MODBUS
Code Name - LORAWAN_P6_MODBUS_DEMO_V1.0

*****Join Parameters*****
DevEUI : 0x0000fccc23d227eac
JoinEUI : 0x0000000000000005
AppKey : 0x00000000000000000000000000000005

TxPower Index = 01
Current Data rate = DR5
ADR = OFF

Confirmed Retries : 04

Join Request Sent for AS923

Joining Denied, cntRetries 0
Status : LORAWAN_RADIO_BUSY
Join Attempt was on CH 0
Awaiting next Join attempt

*****
Initial Finished
*****

1. Send Join Request
2. Send Data
3. Start Periodic Data
4. Stop Periodic Data
5. Sleep
6. App Information
7. Reset

Enter your choice:
```

The Menu Option as shown in the following figure.

```
*****
1. Send Join Request
2. Send Data
3. Start Periodic Data
4. Stop Periodic Data
5. Sleep
6. App Information
7. Reset

Enter your choice:
```

Option 1: Sends the join request to the network server.

```
Enter your choice:
1

LORAWAN Join...

Joining Successful, cntRetries 1
Joined on Channel 0

DevAddr: 0x2

*****Application Configuration*****
DevType : CLASS A
ActivationType : OTAA
Transmission Type : CONFIRMED
FPort : 1
TxPower Index : 01

Confirmed Retries : 04
*****
```

Option 2: Sends the example data to the network server.

```
Enter your choice:
2

Send Data...

Success

**** Uplink ****
DR      : 5
Type    : Cnf
FPort   : 1
FCntUp  : 0
Data    : 0x0374657374
Size    : 5
Status  : LORAWAN_SUCCESS
Transmission Success
*****
```

Option 3: Start periodically transmit (default periodic timer duration 10 seconds).

Define by `DEMO_CONF_APP_PERIODIC_TIMER_PERIOD_MS` in `conf_app.h`.

```
Enter your choice:
3

Start periodic data...

Success

**** Uplink ****
DR      : 5
Type    : Cnf
FPort   : 1
FCntUp  : 2
Data    : 0x0574657374
Size    : 5
Status  : LORAWAN_SUCCESS
Transmission Success
*****
```

Option 4: Stop periodically transmit

```
Enter your choice:
4

Stop periodic data...
```

Option 5: Puts the end-device into sleep for 5 sec (PMM Standby mode, refer to [5.2 Sleep Mode](#))

```
Enter your choice:
5

Sleep Mode...

Sleep done: 4990 ms
*****
1. Send Join Request
```

(Wake up by Timer)

```
Enter your choice:
5

Sleep Mode...

Sleep done: 360 ms

Exiting Sleep by BUTTON_0
*****
```

(Wake up by Button)

Option 6: Print app information and Join Parameters

```
Enter your choice:
6

App Information...
*****

Last reset cause: External Reset
Microchip LoRaWAN Stack - MLS_SDK_1_0_P_6
Board Name - SAMR34_EVA_MODBUS
Code Name - LORAWAN_P6_MODBUS_DEMO_V1.0

*****Join Parameters*****

DevEUI : 0x0000fcc23d227eac
AppEUI : 0x0000000000000005
AppKey : 0x00000000000000000000000000000005

*****Application Configuration*****

DevType      : CLASS A
ActivationType : OTAA
Transmission Type : CONFIRMED
FPort        : 1
*****
```

Option 7: Resets the board (soft Reset)

```
Enter your choice:
7

Module Reset...

Last reset cause: System Reset Request
Microchip LoRaWAN Stack - MLS_SDK_1_0_P_6
Board Name - SAMR34_EVA_MODBUS
Code Name - LORAWAN_P6_MODBUS_DEMO_V1.0
```

Note : How to Join to LoRaWAN Network

1. Enter 6 (or 6<CR><LF>) or push a User button for shown app information and Join Parameter via UART or USB interface

```
Enter your choice:
6
App Information...
*****
Last reset cause: External Reset
Microchip LoRaWAN Stack - MLS_SDK_1_0_P_6
Board Name - SAMR34_EVA_MODBUS
Code Name - LORAWAN_P6_MODBUS_DEMO_V1.0

*****Join Parameters*****
DevEUI : 0x0000fcc23d227eac
AppEUI : 0x0000000000000005
AppKey : 0x00000000000000000000000000000005
*****Application Configuration*****
DevType      : CLASS A
ActivationType : OTAA
Transmission Type : CONFIRMED
FPort        : 1
*****
```

(form UART Interface)

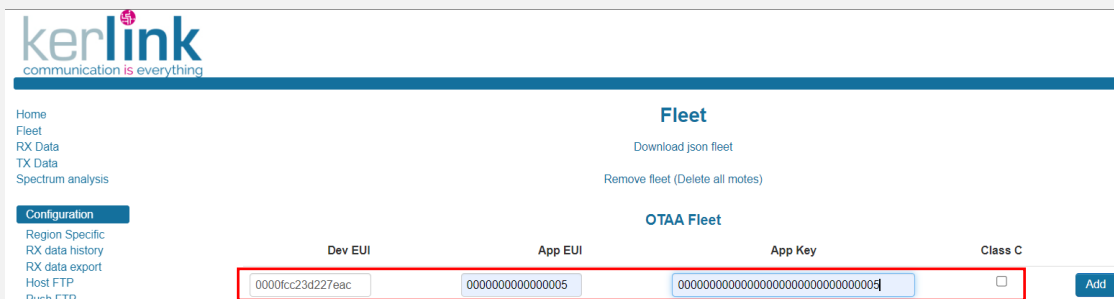
```
USB start...
App Information...
Microchip LoRaWAN Stack - MLS_SDK_1_0_P_6
Board Name - SAMR34_EVA_MODBUS
Code Name - LORAWAN_P6_MODBUS_DEMO_V1.0

*****Join Parameters*****
DevEUI : 0x0000fcc23d227eac
AppEUI : 0x0000000000000005
AppKey : 0x00000000000000000000000000000005
*****Application Configuration*****
DevType      : CLASS A
ActivationType : OTAA
Transmission Type : CONFIRMED
FPort        : 1
*****
```

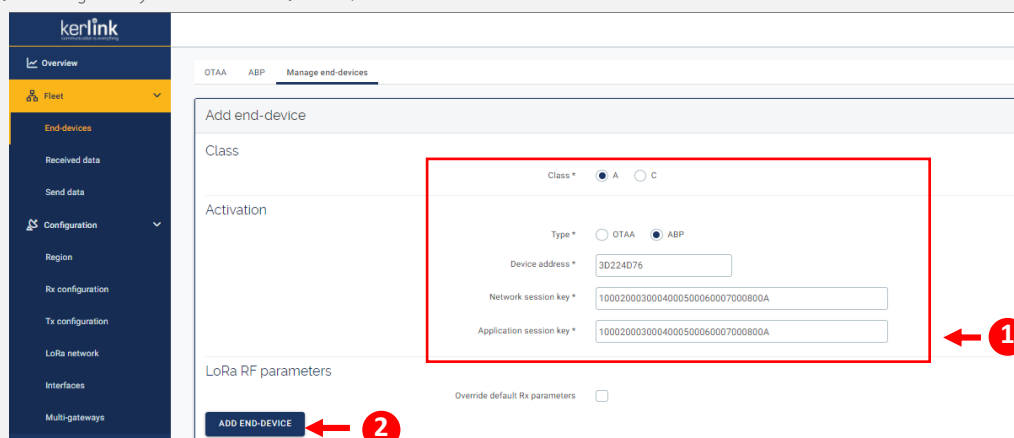
(form USB Interface)

2. Add Join Parameter to LoRaWAN Network

(Example) : LoRa gateway: Wirnet Station 923 (Kerlink)



(Example) : LoRa gateway: PDTIOT-IFE02 (Kerlink)



3. Push Reset Button on Evaluation Board or Enter 1 (or 1<CR><LF>) for send the join request to network server.

4 Example Modbus Demo Project

This Application is same as [3 Example Demo Project](#) and used to be Master for Modbus Application, send request to Slave Device and uplink Response payload through the LoRaWAN Network.

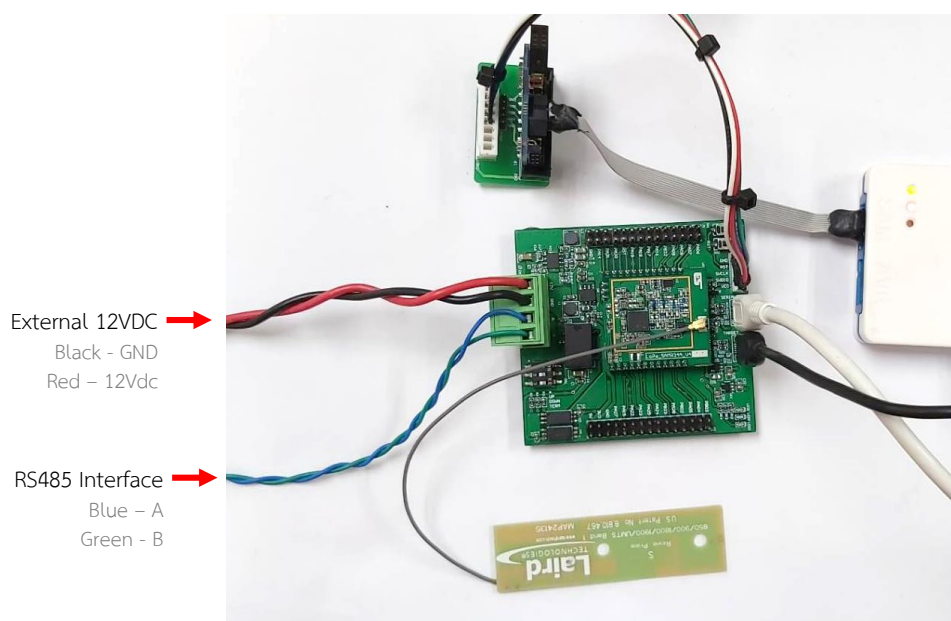
- Periodically scan Modbus Slave Device and transmit a confirmed data frame (default periodic timer duration 10 seconds).

Define `DEMO_CONF_APP_PERIODIC_TIMER_PERIOD_MS` in `conf_app.h`.

- LED2 and LED3 show Modbus Transmitting and Receiving status.

4.1 Setup

Same as [3.1 Setup](#) (from [3 Example Demo Project](#)), plug External 12VDC and Connect RS485 connector to Modbus Slave device.



Note : For connector pin and details, see [1.1 Evaluation Board](#) and [2.2 Board Connectors](#)

4.2 Application Configuration

Define application configuration parameters in “..\src\config\conf_enddevice.h”

define **CONF_MODBUS_ENABLE** for enable modbus option.

```
/* MODBUS */
#define CONF_MODBUS_ENABLE
#ifdef CONF_MODBUS_ENABLE
#define MODBUS_RESPONSE_TIMEOUT (500u) //ms
#define MODBUS_REQUEST_REPEAT (2u)
```

4.3 Demo Application Usage

The Menu Option as shown in the following figure.

```
*****
1. Send Join Request
2. Send Data
3. Scan Modbus
4. Scan and Send Data
5. Start Periodic Data
6. Stop Periodic Data
7. Sleep
8. App Information
9. Reset
Enter your choice:
```

Option 1: Sends the join request to the network server.

Option 2: Sends the example data to the network server.

Option 3: Sends the Modbus request (Request command order, see in [5.3.1 Modbus request Commands](#))

```
Enter your choice:
3
Scan Modbus...

Modbus id 0x01, addr 0x0000, Qty 0x000a status:00, 25: 0x01031400aa00020003000400050006000700080009000a8b2a
Modbus id 0x01, addr 0x0001, Qty 0x0005 status:00, 15: 0x01030a000200030004000500066f15
Modbus id 0x03, addr 0x0013, Qty 0x000a status:01, 1: 0x00
Modbus id 0x04, addr 0x0001, Qty 0x0005 status:00, 15: 0x04030a000200030002f002e0000b41f
Modbus id 0x05, addr 0x001b, Qty 0x0006 status:00, 5: 0x0583028130
*****
```

Option 4: Sends the Modbus request and uplink modbus response payload to the network sever. (uplink payload, see in [5.3.2 Response Payload](#))

```
Enter your choice:
4
Scan Modbus and Send Data...

Modbus id 0x01, addr 0x0000, Qty 0x000a status:00, 25: 0x01031400aa00020003000400050006000700080009000a8b2a
Modbus id 0x01, addr 0x0001, Qty 0x0005 status:00, 15: 0x01030a000200030004000500066f15
Modbus id 0x03, addr 0x0013, Qty 0x000a status:01, 1: 0x00
Modbus id 0x04, addr 0x0001, Qty 0x0005 status:00, 15: 0x04030a000200030002f002e0000b41f
Modbus id 0x05, addr 0x001b, Qty 0x0006 status:00, 5: 0x0583028130
Success

*** Uplink ***
DR : 5
Type : Cnf
Port : 1
FCntUp : 0
Data : 0x010001031400aa00020003000400050006000700080009000a8b2a
Size : 27
Status : LORAWAN_SUCCESS
Transmission Success
*****
```


Option 5: Start periodically transmit (default periodic timer duration 10 seconds).

Define by **DEMO_CONF_APP_PERIODIC_TIMER_PERIOD_MS** in **conf_app.h**.

```
Enter your choice:
5

Start periodic data...

Modbus id 0x01, addr 0x0000, Qty 0x000a status:00, 25: 0x01031400aa00020003000400050006000700080009000a8b2a
Modbus id 0x01, addr 0x0001, Qty 0x0005 status:00, 15: 0x01030a000200030004000500066f15
Modbus id 0x03, addr 0x0013, Qty 0x000a status:01, 1: 0x00
Modbus id 0x04, addr 0x0001, Qty 0x0005 status:00, 15: 0x04030a000200030002f002a0000b41f
Modbus id 0x05, addr 0x001b, Qty 0x0006 status:00, 5: 0x0583028130
Success

*** Uplink ***
DR : 5
Type : Cnf
FPort : 1
FCntUp : 3
Data : 0x050001031400aa00020003000400050006000700080009000a8b2a
Size : 27
Status : LORAWAN_SUCCESS
Transmission Success
*****
```

Option 6: Stop periodically transmit

Option 7: Puts the end-device into sleep for 5 sec (PMM Standby mode, refer to [5.2 Sleep Mode](#))

Option 8: Print app information and Join Parameters

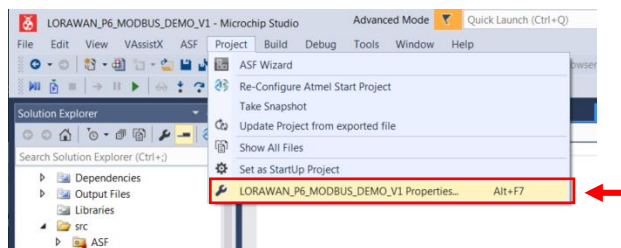
Option 9: Resets the board (soft Reset)

Note : Option 1-2, 5-9 same as [3.5 Demo Application Usage](#)(form [3 Example Demo Project](#))

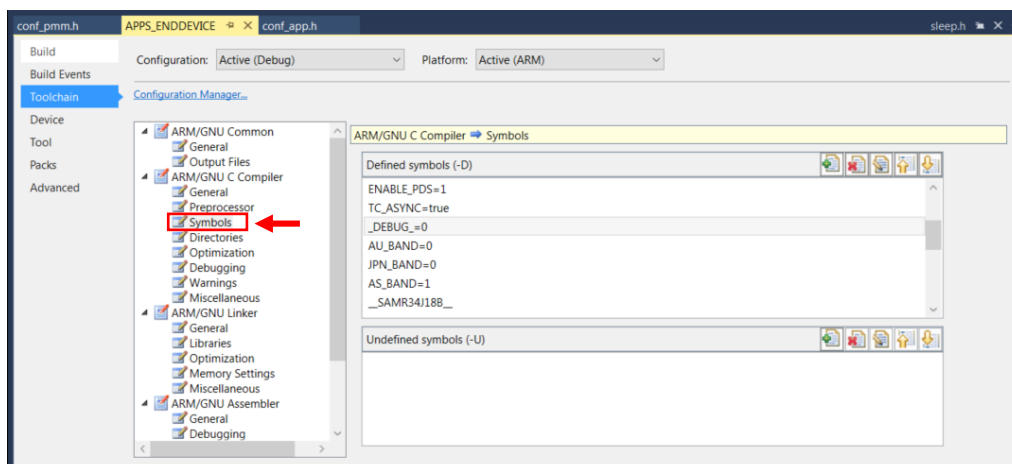
5 Setting Project / Properties

5.1 EUI Address

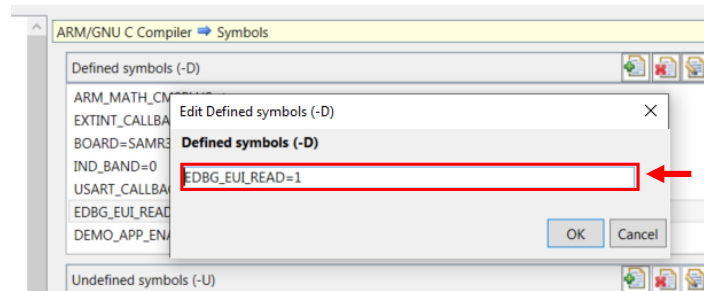
Go to Project > Properties



Select **toolchain**, Go to ARM/GNU C Compiler > Symbols



Double click at `EDBG_EUI_READ`, define `EDBG_EUI_READ` as 1 to read DevEUI from EUI-48 MAC Address and set it.



Otherwise the value `DEMO_DEVICE_EUI` and `DEMO_DEVICE_ADDRESS` configured in `conf_app.h`.

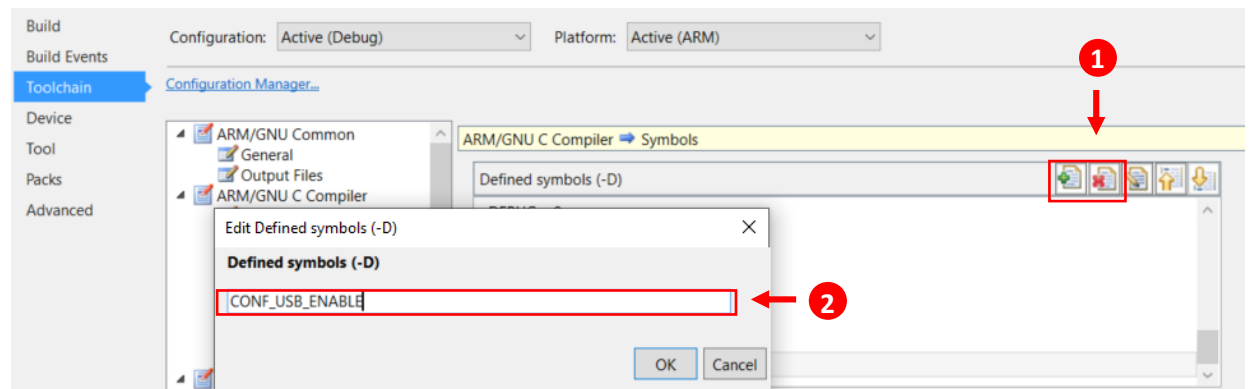
```
/* ABP Join Parameters */
#define DEMO_DEVICE_ADDRESS 0xabcd0001
#define DEMO_APPLICATION_SESSION_KEY {0x10, 0x00, 0x20, 0x00, 0x30, 0x00, 0x40, 0x00, 0x50, 0x00, 0x60, 0x00, 0x70, 0x00, 0x80, 0x0A}
#define DEMO_NETWORK_SESSION_KEY {0x10, 0x00, 0x20, 0x00, 0x30, 0x00, 0x40, 0x00, 0x50, 0x00, 0x60, 0x00, 0x70, 0x00, 0x80, 0x0A}

/* OTAA Join Parameters */
#define DEMO_DEVICE_EUI {0x00, 0x01, 0x00, 0x00, 0x35, 0x8f, 0x00, 0x00}
#define DEMO_APPLICATION_EUI {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x05}
#define DEMO_APPLICATION_KEY {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x05}
```

5.2 Sleep Mode

Go to Project > Properties, select **toolchain**, Go to ARM/GNU C Compiler > Symbols

add or remove `CONF_PMM_ENABLE` macro from Defined symbol for disable/enable Sleep mode function.



Go to `conf_pmm.h` and for setting `CONF_PMM_SLEEPMODE_WHEN_IDLE`

```
/*
 * This macro specifies the sleep mode to use when node is in IDLE
 * mode i.e., when no transaction is happening.
 */
#define CONF_PMM_SLEEPMODE_WHEN_IDLE SLEEP_MODE_STANDBY
// #define CONF_PMM_SLEEPMODE_WHEN_IDLE SLEEP_MODE_BACKUP
```

and go to `conf_app.h` for setting sleep duration. By default, configured in 5000 msec.

```
/* This macro defines the application's default sleep duration in milliseconds */
#define DEMO_CONF_DEFAULT_APP_SLEEP_TIME_MS 5000
```

5.3 Modbus

5.3.1 Modbus request Commands

Go to `..\src\modbus_app.c`, set Slave Id, Function, Start address, Quantity of Registers in `Device_t Slave` (in example limit maximum 10 commands)

In Example, 5 request commands are follows:

Request NO.		1	2	3	4	5
Field Name	Variable Name					
Slave ID	Slave.id	0x01	0x01	0x01	0x01	0x01
Function Code	Slave.func	0x03	0x03	0x03	0x03	0x03
Starting Address	Slave.Addr	0x0000	0x0001	0x0013	0x0001	0x0018
Quantity of input Registers	Salve.data	0x000A	0x0005	0x000A	0x0005	0x0006

```
typedef struct _Device_t{
    const uint8_t id;
    uint8_t status;
    uint8_t func;
    uint16_t addr;
    uint8_t data;
    uint8_t buffer[MODBUS_DATA_BUFFER_MAX];
    uint8_t size;
}Device_t;

Device_t Slave[MODBUS_DEVICE_TOTAL] = { {1, 0, MB_READ_HOLDING_REGISTERS, 0, 10},
                                           {1, 0, MB_READ_HOLDING_REGISTERS, 1, 5},
                                           {3, 0, MB_READ_HOLDING_REGISTERS, 19, 10},
                                           {4, 0, MB_READ_HOLDING_REGISTERS, 1, 5},
                                           {5, 0, MB_READ_HOLDING_REGISTERS, 27, 6},
                                           {0, 0, 0, 0, 0} };
```

5.3.2 Response Payload

In Example, uplink payload formats is follow:

Payload No.	Error Status	Modbus Response
1 Byte	1 Byte	N Bytes

Byte	Value	Description
Payload No.	01	Payload No. 0x01 to 0xff
Error Status	00	Error Status (Binary format) bit 0 : Modbus timeout bit 1 : CRC Failed bit 2: Response Failed bit 3: Request parameter Failed E.g. 0x00 = no error
Message	01031400aa00020003000400050006000700080009000a8b2a	Modbus response for request command no.1

Go to “..\src\enddevice.c”, see in function `static void insert_data_modbus(void)`

```
//-----
static void insert_data_modbus(void){
    uint8_t slave_id = 1;

    if(0<(mbReceivedSize=getSlaveResponse(mbReceivedBuffer+1,slave_id))){
        mbReceivedBuffer[0] = getSlaveStatus(slave_id);
        mbReceivedSize += 1;
    }
    else{
        memset(mbReceivedBuffer,0,3);
        mbReceivedBuffer[1] =0xff;
    }
}
```

5.4 LEDs

Go to ..\src\config\conf_app.h define `DEMO_LED_STATUS` for enable, or undefined for disable led status option.

```
/* This macro enables or disables the LED indications */
#define DEMO_LED_STATUS
```

5.5 Watchdog Timer

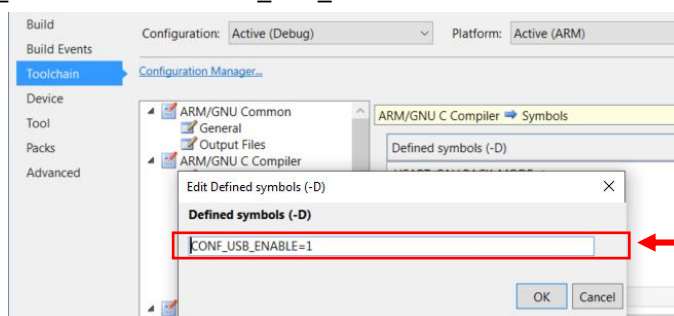
Go to ..\src\config\conf_enddevice.h define `CONF_WDT_ENABLE` for enable, or undefined for disable watchdog function.

```
/* WDT */
#define CONF_WDT_ENABLE
#ifdef CONF_WDT_ENABLE
#define APP_PERIOD_WDT_CLEAR (1000u)
#endif //CONF_WDT_ENABLE
```

5.6 USB

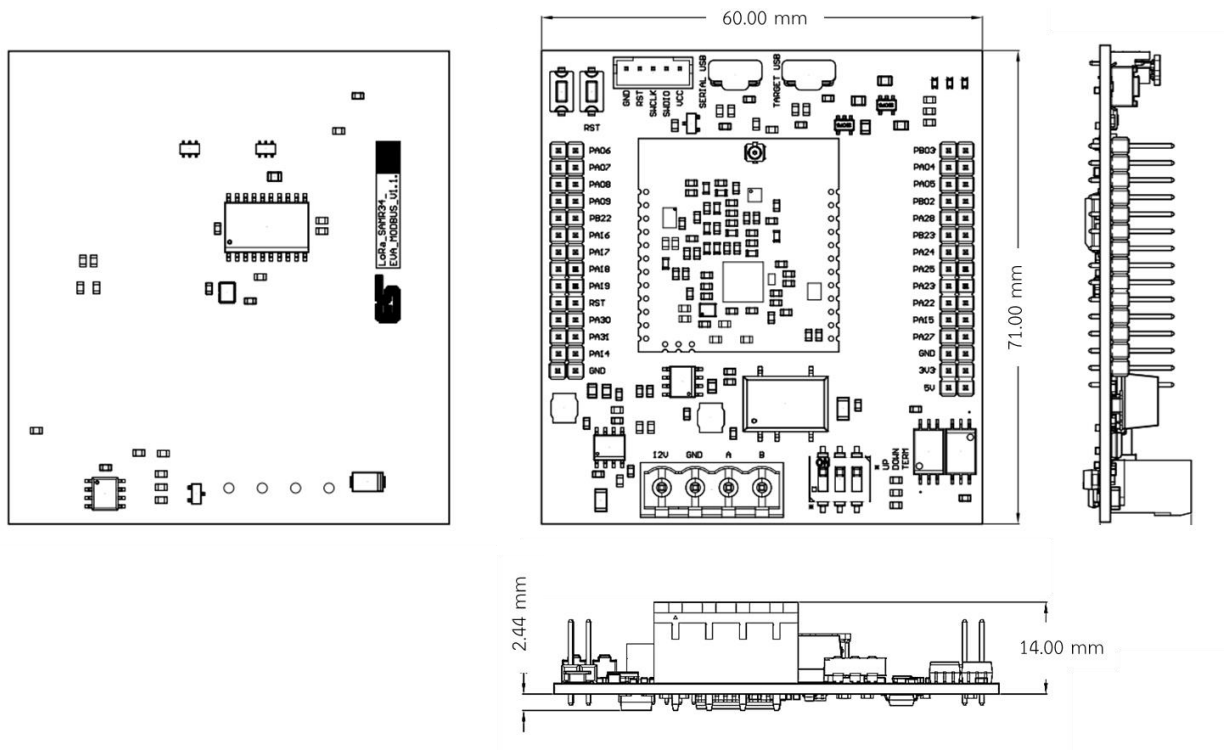
Go to Project > Properties, select **toolchain**, Go to ARM/GNU C Compiler > Symbols

Double click at `CONF_USB_ENABLE`, define `CONF_USB_ENABLE` as 1, for enable USB function.



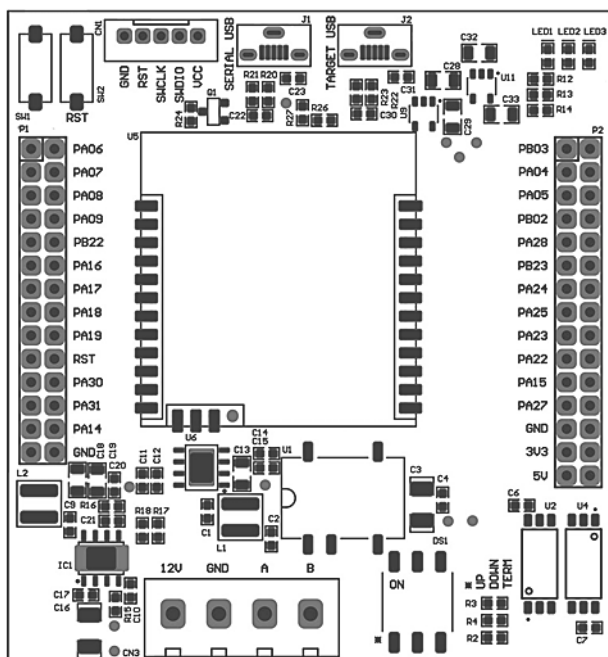
6 Appendix

6.1 Board Dimensions

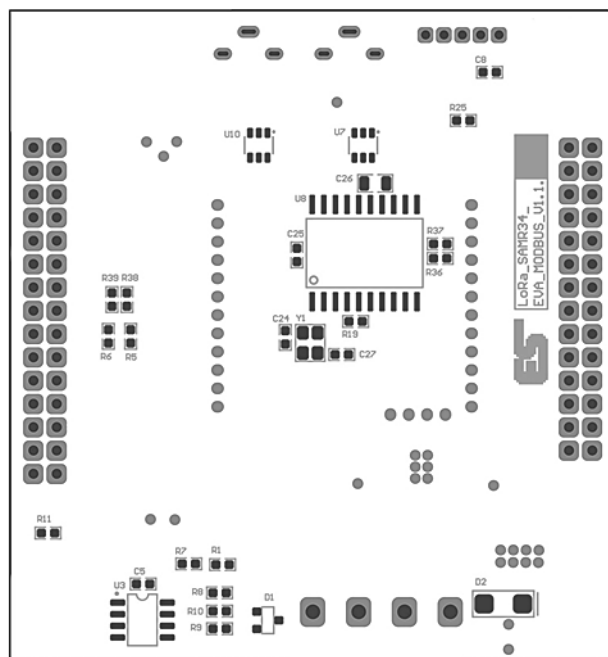


6.2 Board Layout

LoRa_SAMR34A_EVA_MODBUS Layout



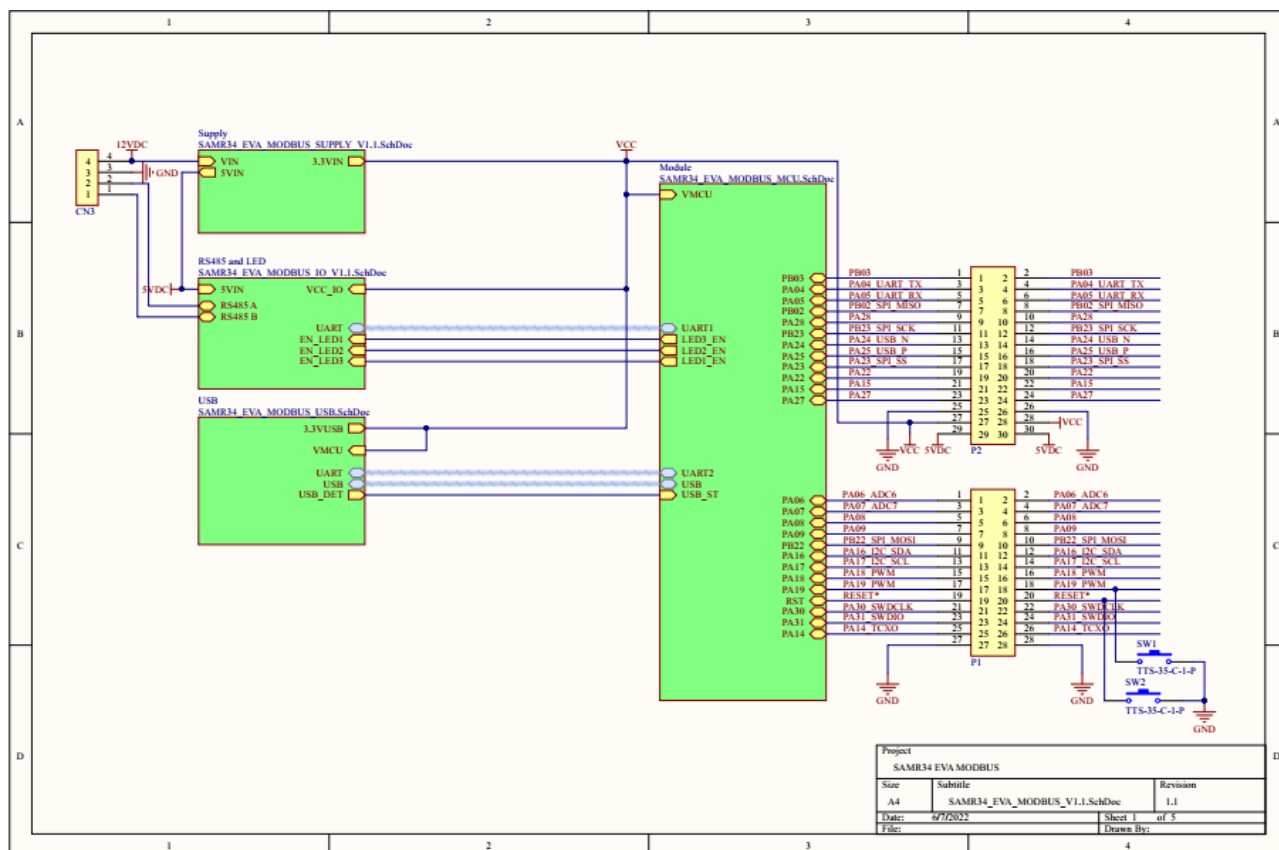
(Top)

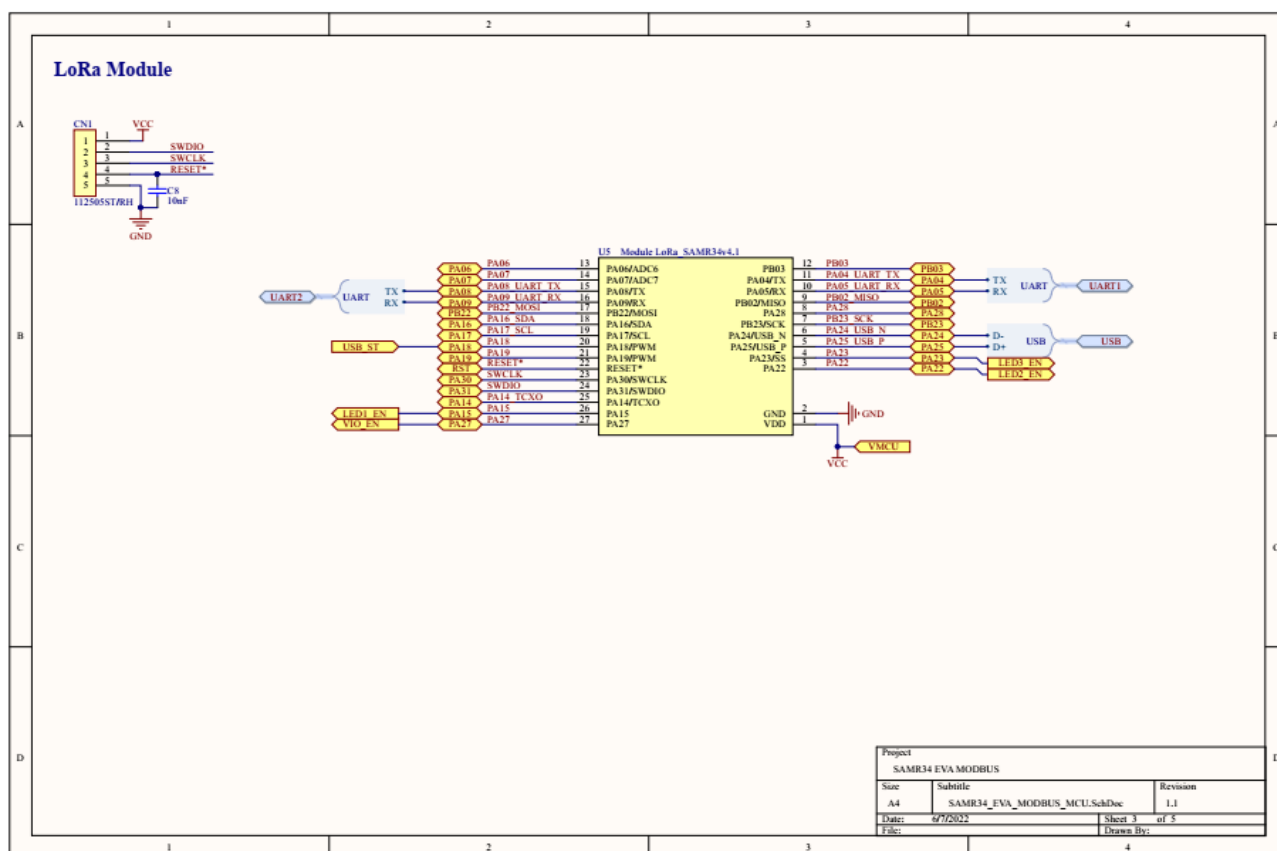
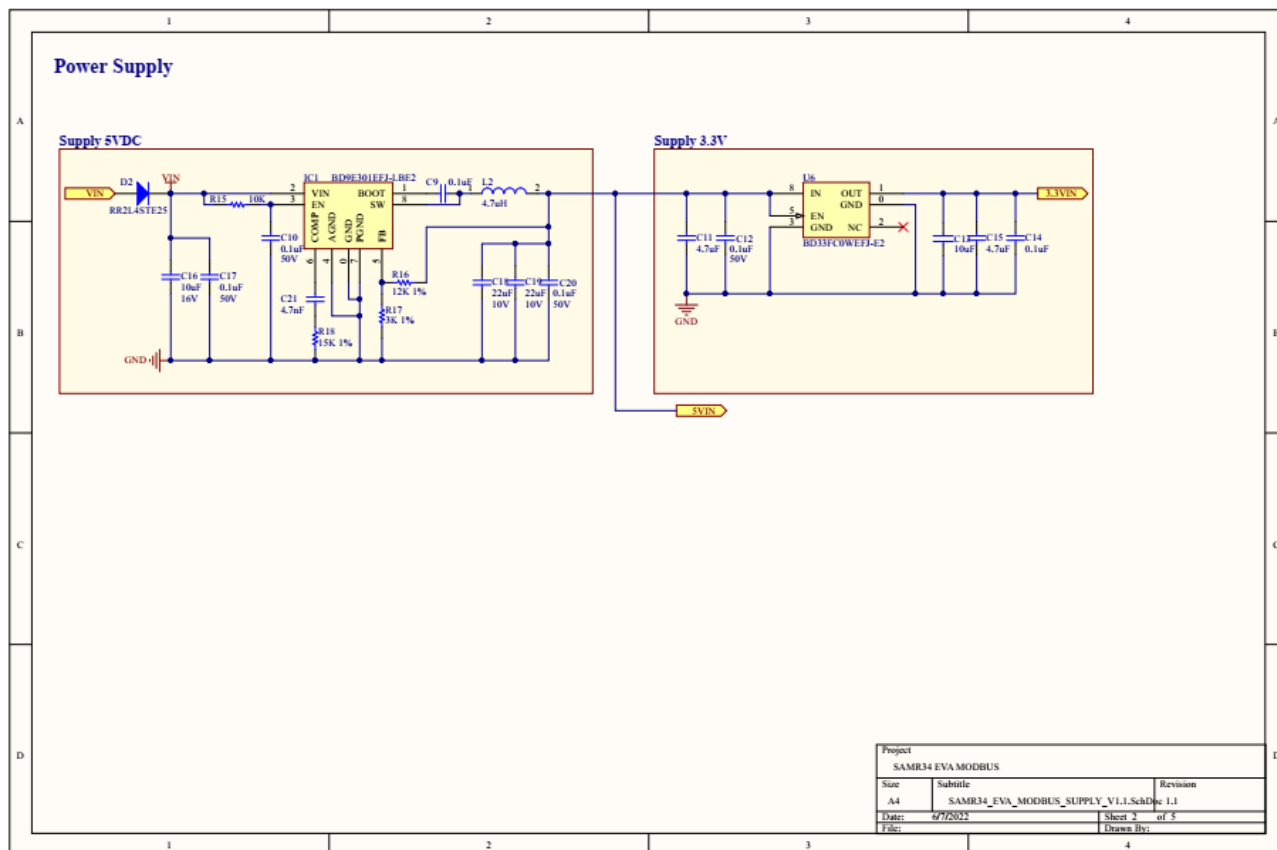


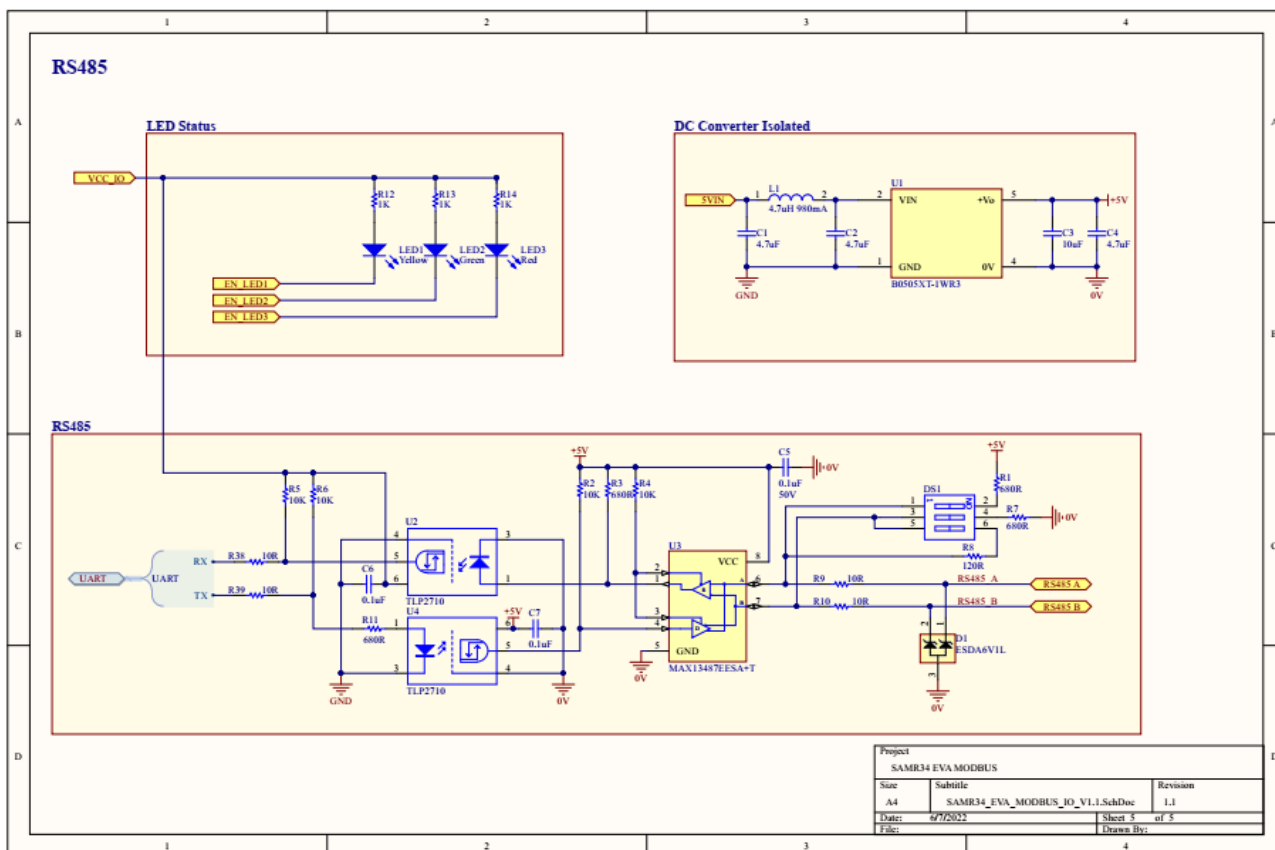
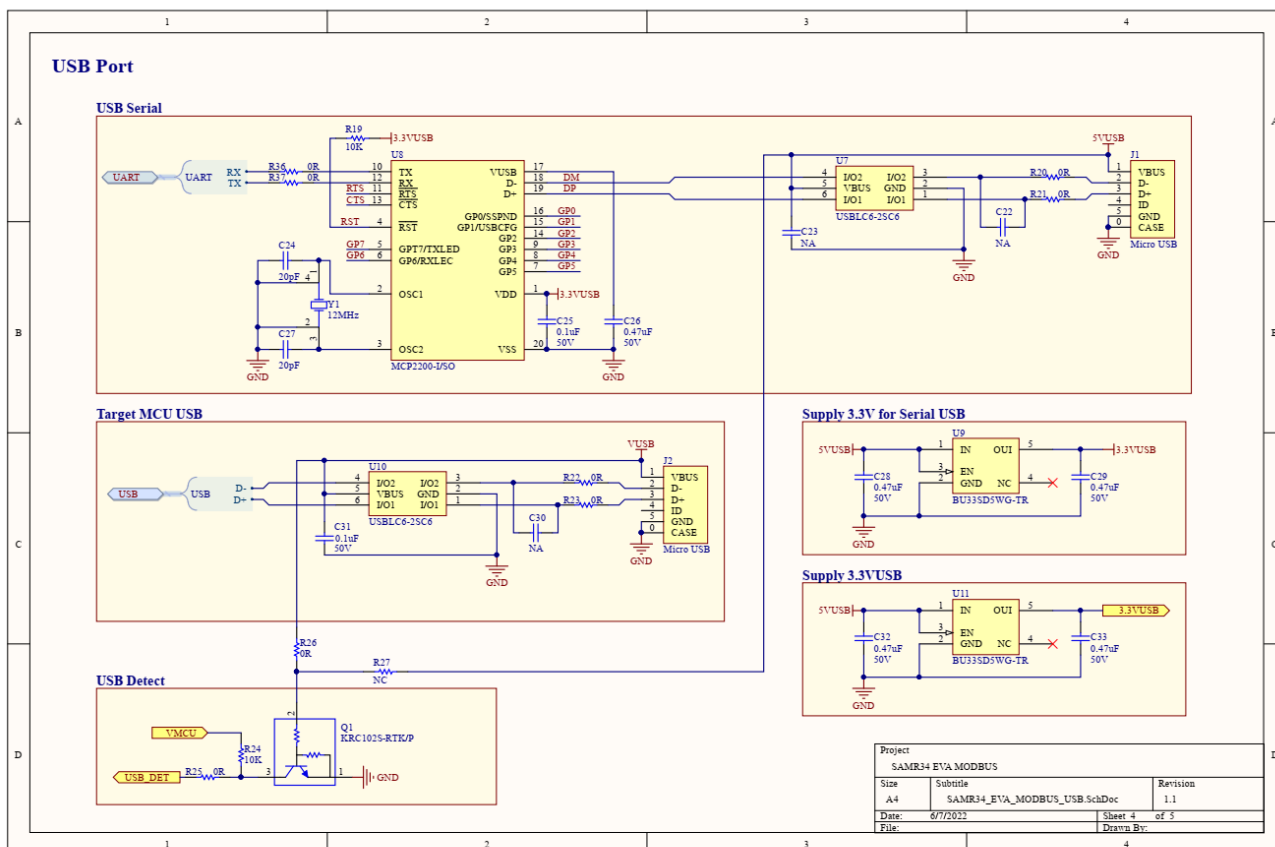
(Bot)

Note : For more details on LoRa_SAMR34A_V4 module, refer [LoRa_SAMR34A_V4.1 module](#)

6.3 Schematic

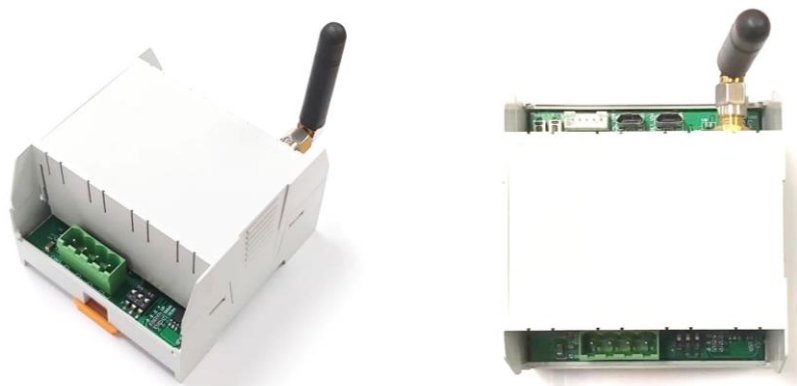






6.4 Optional Accessories

Evaluation Board Modbus RS485 to LoRaWAN (Cover Standard Din Rail)



P/N	Description
0271113MCF-100	RF Cable convert SMA Female Bulkhead to MCF Male
BY-GSM-01 48 MM.	GSM Antenna, Frequency 850-1900/900-1800/2100 MHz Antenna Length 48mm, Connector SMA Right Angle Male
2201819	Electronic Housing, Mounting Base Housing, Material ABS,Size 75.4x70.2x46 mm DIN Rail Application
2201830	Electronic Housing,Upper Part Housing Cover =aterial ABS,Size 75.4x70.2 mm(WxL) DIN Rail Application

7 Reference Links

- [LoRa_SAMR34A_V4.1 module](#)
- [LoRa_SAMR34_EVA\(ES\)](#) – Documentation page and source code
- [SAM R34/R35 Low Power LoRa Sub-GHz SiP Datasheet](#)
- [Microchip RF tools for LoRa](#) - RF tool can be used for detailed testing and RF certification
- [SAM R34 MLS Getting Started Guide](#)

8 Reversion History

Revision	Description	Date
1.0	Initial version	06-07-2022