文档类型：产品介绍

文档密级：公开

RG-IRT6110 SDK 使用手册
V1.00

锐捷网络股份有限公司

修订记录

序号	版本号	时 间	作 者	修订原因
1	V1.0	2018-5-18	郑闵博	初次建立
2				
3				

目录

1 引言

本文介绍了如何使用 IRT6110 SDK 去二次开发 RG-IRT6110 模组。

2 开发环境说明

2.1 开发环境安装和代码编译

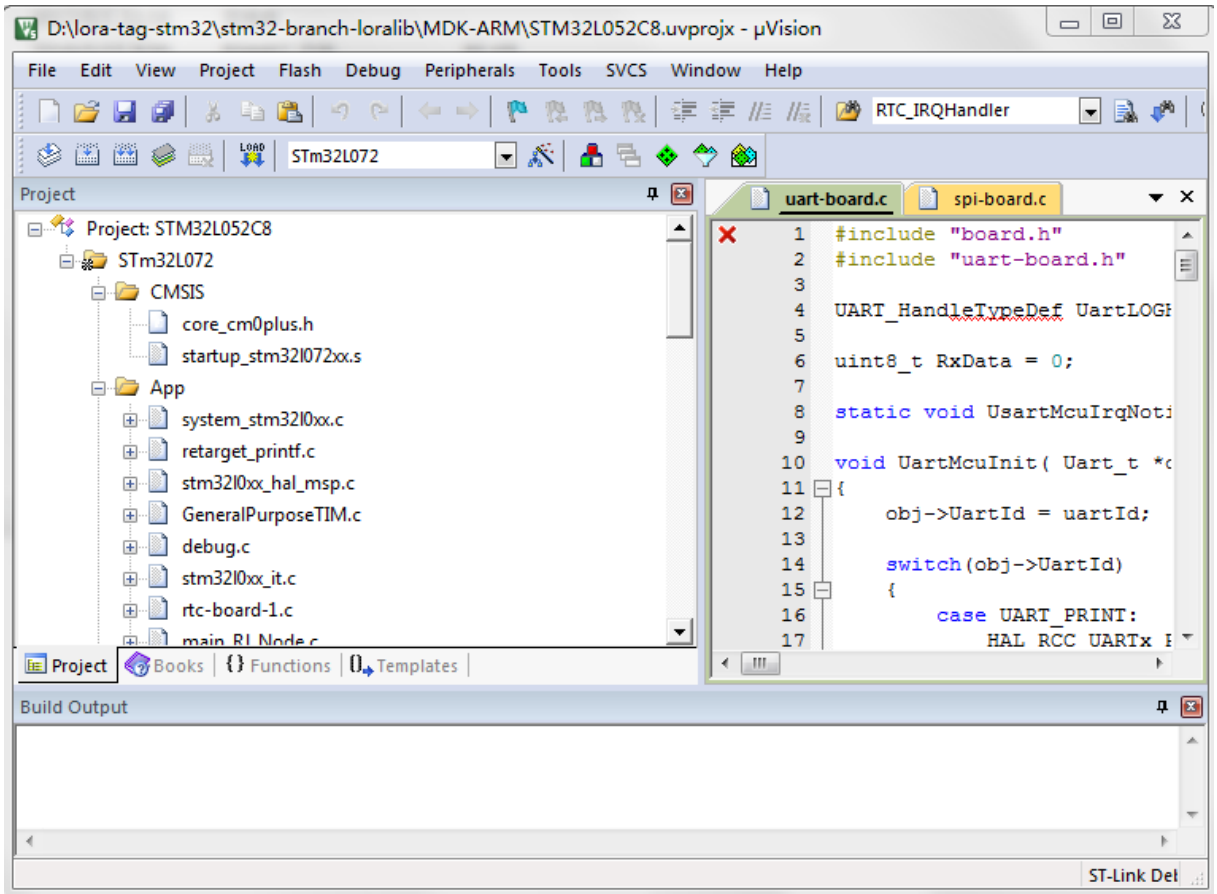
本工程开发采用 KEIL 集成开发环境， MCU 型号为 STM32L072CBTx，集成开发环境为 Keil uVision V5.16a，可以从官网下载获得：

http://www.keil.com/

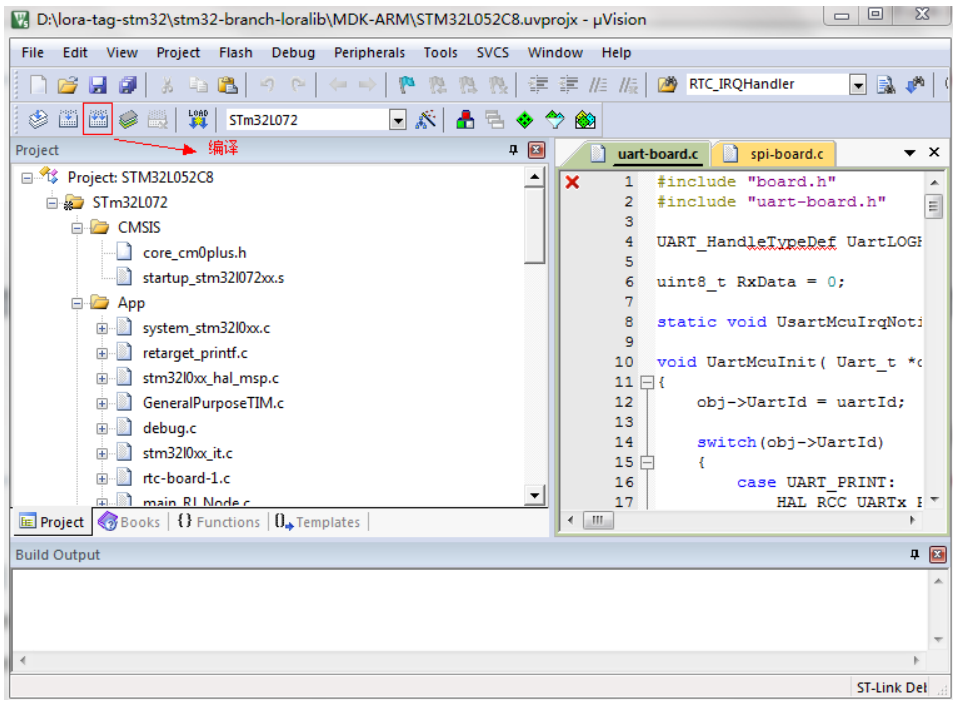
根据提示默认安装即可。

安装完成后，双击工程文件夹下：

MDK-ARM\STM32L052C8.uvprojx 文件就可以打开工程，打开后如下：

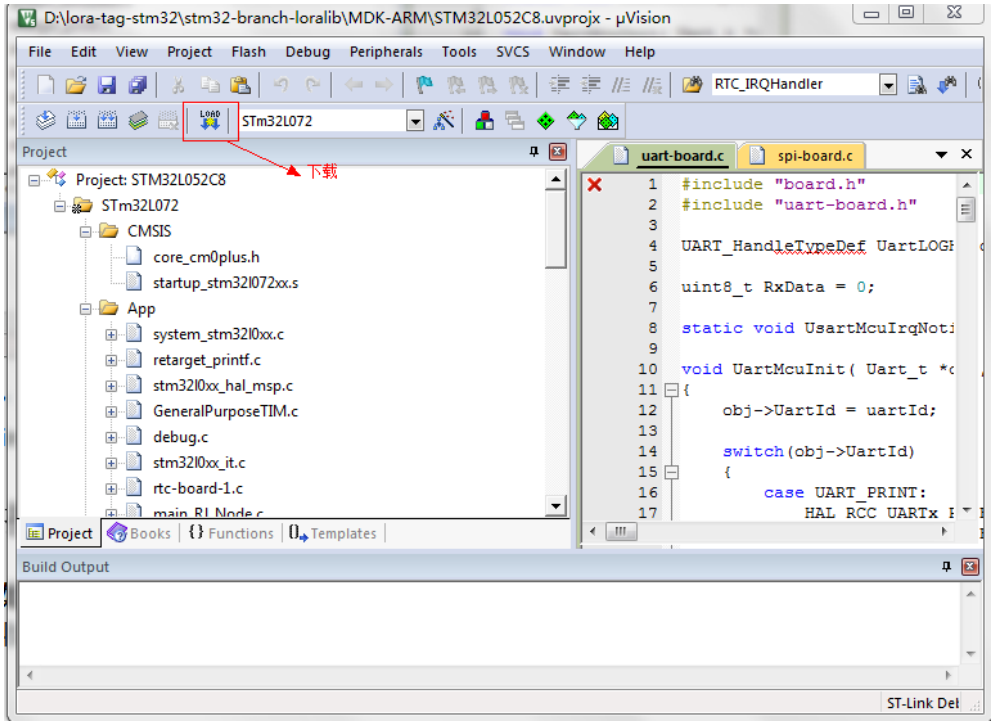


选择 编译按钮



，等待编译完成后，可以在工程文件夹下 MDK-ARM \Objects 看到编出的 bin 文件：STM32L052C8.bin。

可以使用下载功能烧写 bin 程序到模组上



3 SDK 说明

3.1 LORA 初始化

用于注册 LORA 收包处理等接口

```
/*!
 * \brief   LoRaMAC layer initialization
 *
 * \details In addition to the initialization of the LoRaMAC layer, this
 *          function initializes the callback primitives of the MCPS and
 *          MLME services. Every data field of \ref LoRaMacPrimitives_t must be
 *          set to a valid callback function.
 *
 * \param   [IN] events - Pointer to a structure defining the LoRaMAC
 *                        event functions. Refer to \ref LoRaMacPrimitives_t.
 *
 * \param   [IN] events - Pointer to a structure defining the LoRaMAC
 *                        callback functions. Refer to \ref LoRaMacCallback_t.
 *
 * \retval  LoRaMacStatus_t Status of the operation. Possible returns are:
 *          returns are:
 *          \ref LORAMAC_STATUS_OK,
 *          \ref LORAMAC_STATUS_PARAMETER_INVALID.
 */
LoRaMacStatus_t LoRaMacInitialization( LoRaMacPrimitives_t *primitives, LoRaMacCallback_t
*callbacks );
```

3.2.获取 LORA 参数

```
/*!
 * \brief   LoRaMAC MIB-Get
 *
 * \details The mac information base service to get attributes of the LoRaMac
 *          layer.
 *
 *          The following code-snippet shows how to use the API to get the
 *          parameter AdrEnable, defined by the enumeration type
 *          \ref MIB_ADR.
 * \code
 * MibRequestConfirm_t mibReq;
 * mibReq.Type = MIB_ADR;
 *
 * if( LoRaMacMibGetRequestConfirm( &mibReq ) == LORAMAC_STATUS_OK )
 * {
 *   // LoRaMAC updated the parameter mibParam.AdrEnable
 * }
 * \endcode
 *
 * \param   [IN] mibRequest - MIB-GET-Request to perform. Refer to \ref MibRequestConfirm_t.
 *
```

```
 * \retval  LoRaMacStatus_t Status of the operation. Possible returns are:
 *          \ref LORAMAC_STATUS_OK,
 *          \ref LORAMAC_STATUS_SERVICE_UNKNOWN,
 *          \ref LORAMAC_STATUS_PARAMETER_INVALID.
 */
LoRaMacStatus_t LoRaMacMibGetRequestConfirm( MibRequestConfirm_t *mibGet );
```

可获取及设置的参数说明：
以下参数通过设置入参的 mibReq 结构体进行设置

```
/*!
 * LoRa Mac Information Base (MIB)
 *
 * The following table lists the MIB parameters and the related attributes:
 *
 * Attribute                          | Get | Set
 * ---------------------------------- | :-: | :-:
 * \ref MIB_DEVICE_CLASS              | YES | YES
 * \ref MIB_NETWORK_JOINED            | YES | YES
 * \ref MIB_ADR                       | YES | YES
 * \ref MIB_NET_ID                    | YES | YES
 * \ref MIB_DEV_ADDR                  | YES | YES
 * \ref MIB_NWK_SKEY                  | YES | YES
 * \ref MIB_APP_SKEY                  | YES | YES
 * \ref MIB_PUBLIC_NETWORK            | YES | YES
 * \ref MIB_REPEATER_SUPPORT          | YES | YES
 * \ref MIB_CHANNELS                  | YES | NO
 * \ref MIB_RX2_CHANNEL               | YES | YES
 * \ref MIB_CHANNELS_MASK             | YES | YES
 * \ref MIB_CHANNELS_NB_REP           | YES | YES
 * \ref MIB_MAX_RX_WINDOW_DURATION    | YES | YES
 * \ref MIB_RECEIVE_DELAY_1           | YES | YES
 * \ref MIB_RECEIVE_DELAY_2           | YES | YES
 * \ref MIB_JOIN_ACCEPT_DELAY_1       | YES | YES
 * \ref MIB_JOIN_ACCEPT_DELAY_2       | YES | YES
 * \ref MIB_CHANNELS_DATARATE         | YES | YES
 * \ref MIB_CHANNELS_DEFAULT_DATARATE| YES | YES
 * \ref MIB_CHANNELS_TX_POWER         | YES | YES
 * \ref MIB_CHANNELS_DEFAULT_TX_POWER| YES | YES
 * \ref MIB_UPLINK_COUNTER            | YES | YES
 * \ref MIB_DOWNLINK_COUNTER          | YES | YES
 * \ref MIB_MULTICAST_CHANNEL         | YES | NO
 * \ref MIB_NUMBER_TOTAL_SEND         | YES | NO
 * \ref MIB_NUMBER_TOTAL_RECV         | YES | NO
 * \ref MIB_CHANNELS_INDEX            | YES | NO
 *
 * The following table provides links to the function implementations of the
 * related MIB primitives:
 *
 * Primitive        | Function
```

```
 * ---------------- | :--------------------:
 * MIB-Set          | \ref LoRaMacMibSetRequestConfirm
 * MIB-Get          | \ref LoRaMacMibGetRequestConfirm
 */
typedef enum eMib
{
    /*!
     * LoRaWAN device class
     *
     * LoRaWAN Specification V1.0.1
     */
    MIB_DEVICE_CLASS,
    /*!
     * LoRaWAN Network joined attribute
     *
     * LoRaWAN Specification V1.0.1
     */
    MIB_NETWORK_JOINED,
    /*!
     * Adaptive data rate
     *
     * LoRaWAN Specification V1.0.1, chapter 4.3.1.1
     *
     * [true: ADR enabled, false: ADR disabled]
     */
    MIB_ADR,
    /*!
     * Network identifier
     *
     * LoRaWAN Specification V1.0.1, chapter 6.1.1
     */
    MIB_NET_ID,
    /*!
     * End-device address
     *
     * LoRaWAN Specification V1.0.1, chapter 6.1.1
     */
    MIB_DEV_ADDR,
    /*!
     * Network session key
     *
     * LoRaWAN Specification V1.0.1, chapter 6.1.3
     */
    MIB_NWK_SKEY,
    /*!
     * Application session key
     *
     * LoRaWAN Specification V1.0.1, chapter 6.1.4
     */
```

```
MIB_APP_SKEY,
/*!
 * Set the network type to public or private
 *
 * LoRaWAN Specification V1.0.1, chapter 7
 *
 * [true: public network, false: private network]
 */
MIB_PUBLIC_NETWORK,
/*!
 * Support the operation with repeaters
 *
 * LoRaWAN Specification V1.0.1, chapter 7
 *
 * [true: repeater support enabled, false: repeater support disabled]
 */
MIB_REPEATER_SUPPORT,
/*!
 * Communication channels. A get request will return a
 * pointer which references the first entry of the channel list. The
 * list is of size LORA_MAX_NB_CHANNELS
 *
 * LoRaWAN Specification V1.0.1, chapter 7
 */
MIB_CHANNELS,
/*!
 * Set receive window 2 channel
 *
 * LoRaWAN Specification V1.0.1, chapter 3.3.2
 */
MIB_RX2_CHANNEL,
/*!
 * LoRaWAN channels mask
 *
 * LoRaWAN Specification V1.0.1, chapter 7
 */
MIB_CHANNELS_MASK,
/*!
 * Set the number of repetitions on a channel
 *
 * LoRaWAN Specification V1.0.1, chapter 5.2
 */
MIB_CHANNELS_NB_REP,
/*!
 * Maximum receive window duration in [ms]
 *
 * LoRaWAN Specification V1.0.1, chapter 3.3.3
 */
MIB_MAX_RX_WINDOW_DURATION,
```

```
/*!
 * Receive delay 1 in [ms]
 *
 * LoRaWAN Specification V1.0.1, chapter 7
 */
MIB_RECEIVE_DELAY_1,
/*!
 * Receive delay 2 in [ms]
 *
 * LoRaWAN Specification V1.0.1, chapter 7
 */
MIB_RECEIVE_DELAY_2,
/*!
 * Join accept delay 1 in [ms]
 *
 * LoRaWAN Specification V1.0.1, chapter 7
 */
MIB_JOIN_ACCEPT_DELAY_1,
/*!
 * Join accept delay 2 in [ms]
 *
 * LoRaWAN Specification V1.0.1, chapter 7
 */
MIB_JOIN_ACCEPT_DELAY_2,
/*!
 * Default Data rate of a channel
 *
 * LoRaWAN Specification V1.0.1, chapter 7
 *
 * EU868 - [DR_0, DR_1, DR_2, DR_3, DR_4, DR_5, DR_6, DR_7]
 *
 * US915 - [DR_0, DR_1, DR_2, DR_3, DR_4, DR_8, DR_9, DR_10, DR_11, DR_12, DR_13]
 */
MIB_CHANNELS_DEFAULT_DATARATE,
/*!
 * Data rate of a channel
 *
 * LoRaWAN Specification V1.0.1, chapter 7
 *
 * EU868 - [DR_0, DR_1, DR_2, DR_3, DR_4, DR_5, DR_6, DR_7]
 *
 * US915 - [DR_0, DR_1, DR_2, DR_3, DR_4, DR_8, DR_9, DR_10, DR_11, DR_12, DR_13]
 */
MIB_CHANNELS_DATARATE,
/*!
 * Transmission power of a channel
 *
 * LoRaWAN Specification V1.0.1, chapter 7
 *
```

```
     * EU868 - [TX_POWER_20_DBM, TX_POWER_14_DBM, TX_POWER_11_DBM,
     *           TX_POWER_08_DBM, TX_POWER_05_DBM, TX_POWER_02_DBM]
     *
     * US915 - [TX_POWER_30_DBM, TX_POWER_28_DBM, TX_POWER_26_DBM,
     *           TX_POWER_24_DBM, TX_POWER_22_DBM, TX_POWER_20_DBM,
     *           TX_POWER_18_DBM, TX_POWER_14_DBM, TX_POWER_12_DBM,
     *           TX_POWER_10_DBM]
     */
    MIB_CHANNELS_TX_POWER,
    /*!
     * Transmission power of a channel
     *
     * LoRaWAN Specification V1.0.1, chapter 7
     *
     * EU868 - [TX_POWER_20_DBM, TX_POWER_14_DBM, TX_POWER_11_DBM,
     *           TX_POWER_08_DBM, TX_POWER_05_DBM, TX_POWER_02_DBM]
     *
     * US915 - [TX_POWER_30_DBM, TX_POWER_28_DBM, TX_POWER_26_DBM,
     *           TX_POWER_24_DBM, TX_POWER_22_DBM, TX_POWER_20_DBM,
     *           TX_POWER_18_DBM, TX_POWER_14_DBM, TX_POWER_12_DBM,
     *           TX_POWER_10_DBM]
     */
    MIB_CHANNELS_DEFAULT_TX_POWER,
    /*!
     * LoRaWAN Up-link counter
     *
     * LoRaWAN Specification V1.0.1, chapter 4.3.1.5
     */
    MIB_UPLINK_COUNTER,
    /*!
     * LoRaWAN Down-link counter
     *
     * LoRaWAN Specification V1.0.1, chapter 4.3.1.5
     */
    MIB_DOWNLINK_COUNTER,
    /*!
     * Multicast channels. A get request will return a pointer to the first
     * entry of the multicast channel linked list. If the pointer is equal to
     * NULL, the list is empty.
     */
    MIB_MULTICAST_CHANNEL,
  /*!
     * packets number of this tag totally send
     */
     MIB_NUMBER_TOTAL_SEND,
     MIB_NUMBER_TOTAL_RECV,
     /*!
       * last send channel index
      */
```

```
    MIB_CHANNELS_INDEX,
}Mib_t;
```

3.3 设置 LORA 参数

```
/*!
 * \brief   LoRaMAC MIB-Set
 *
 * \details The mac information base service to set attributes of the LoRaMac
 *          layer.
 *
 *          The following code-snippet shows how to use the API to set the
 *          parameter AdrEnable, defined by the enumeration type
 *          \ref MIB_ADR.
 *
 * \code
 * MibRequestConfirm_t mibReq;
 * mibReq.Type = MIB_ADR;
 * mibReq.Param.AdrEnable = true;
 *
 * if( LoRaMacMibGetRequestConfirm( &mibReq ) == LORAMAC_STATUS_OK )
 * {
 *   // LoRaMAC updated the parameter
 * }
 * \endcode
 *
 * \param   [IN] mibRequest - MIB-SET-Request to perform. Refer to \ref MibRequestConfirm_t.
 *
 * \retval  LoRaMacStatus_t Status of the operation. Possible returns are:
 *          \ref LORAMAC_STATUS_OK,
 *          \ref LORAMAC_STATUS_BUSY,
 *          \ref LORAMAC_STATUS_SERVICE_UNKNOWN,
 *          \ref LORAMAC_STATUS_PARAMETER_INVALID.
 */
LoRaMacStatus_t LoRaMacMibSetRequestConfirm( MibRequestConfirm_t *mibSet );
```

3.4 发送 JOIN 请求

```
/*!
 * Initiates the Over-the-Air activation
 *
 * \param [IN] devEui Pointer to the device EUI array ( 8 bytes )
 * \param [IN] appEui Pointer to the application EUI array ( 8 bytes )
 * \param [IN] appKey Pointer to the application AES128 key array ( 16 bytes )
 *
 * \retval status [0: OK, 1: Tx error, 2: Already joined a network]
 */
uint8_t LoRaMacJoinReq( uint8_t *devEui, uint8_t *appEui, uint8_t *appKey );
```

3.5 发送不需要 ACK 的数据消息

```
/*!
 * LoRaMAC layer send frame
 *
 * \param [IN] fPort       MAC payload port (must be > 0)
 * \param [IN] fBuffer     MAC data buffer to be sent
 * \param [IN] fBufferSize MAC data buffer size
 *
 * \retval status          [0: OK, 1: Busy, 2: No network joined,
 *                          3: Length or port error, 4: Unknown MAC command
 *                          5: Unable to find a free channel
 *                          6: Device switched off]
 */
uint8_t LoRaMacSendFrame( uint8_t fPort, void *fBuffer, uint16_t fBufferSize );
```

3.6 发送需要 ACK 的数据消息

```
/*!
 * LoRaMAC layer send frame
 *
 * \param [IN] fPort       MAC payload port (must be > 0)
 * \param [IN] fBuffer     MAC data buffer to be sent
 * \param [IN] fBufferSize MAC data buffer size
 * \param [IN] fBufferSize MAC data buffer size
 * \param [IN] nbRetries   Number of retries to receive the acknowledgement
 *
 * \retval status          [0: OK, 1: Busy, 2: No network joined,
 *                          3: Length or port error, 4: Unknown MAC command
 *                          5: Unable to find a free channel
 *                          6: Device switched off]
 */
uint8_t LoRaMacSendConfirmedFrame( uint8_t fPort, void *fBuffer, uint16_t fBufferSize, uint8_t
nbRetries );
```

3.7 LORA LIB 占用的资源

NRST_SX	PB2
V1_CTRL	PA8
V2_CTRL	PC13
DIO0	PA6
DIO1	PA7
DIO2	PB0
DIO3	PB1
SPI_SCK	PB13
SPI_MISO	PB14
SPI_MOSI	PB15
SPI_NSS	PB12

LORA 单元需要使用上述引脚

存储资源占用：

```
Total RO  Size (Code + RO Data)                46768 (  45.67kB)
```

```
Total RW  Size (RW Data + ZI Data)              8632 (   8.43kB)
Total ROM Size (Code + RO Data + RW Data)       47052 (  45.95kB)
```

3.8 时钟频率

时钟频率为 32MHz