```
 * EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 * --/COPYRIGHT--*/

#ifndef SRC_LIB_CAENRFIDLIB_LIGHT_H_
#define SRC_LIB_CAENRFIDLIB_LIGHT_H_

#include "CAENRFIDTypes_Light.h"

 /*
   CAENRFID_Connect
   ---------------------------------------------------------------------------
   Parameters:
      [in],[out]  reader     : The reader data structure that identifies the device.
      [in]        PortType   : The type of port used for connecting to the device.
      [in]        PortParams : The port connection parameters.
   ---------------------------------------------------------------------------
   Returns:
      An error code about the execution of the function.
   ---------------------------------------------------------------------------
      Description:
      The function opens a connection to an attached device.
 */
CAENRFIDErrorCodes CAENRFID_Connect(CAENRFIDReader* reader,
                                    CAENRFIDPort PortType,
                                    void* PortParams);

/*
   CAENRFID_Disconnect
   ---------------------------------------------------------------------------
```

```
    Parameters:
        [in]   reader      : The reader data structure that identifies the device.
    ---------------------------------------------------------------------------
    Returns:
        An error code about the execution of the function.
    ---------------------------------------------------------------------------
    Description:
        The function closes an open connection with an attached device.
*/
CAENRFIDErrorCodes CAENRFID_Disconnect(CAENRFIDReader* reader);


/*
    CAENRFID_GetReadPoints.
    ---------------------------------------------------------------------------
    Parameters:
        [in]  reader        : The reader data structure that identifies the device.
        [out] Antennas      : The names of the Antenna avalaible on the reader.
        [out] numAnt        : The number of string in Antennas.
    ---------------------------------------------------------------------------
    Returns:
        An error code about the execution of the function.
    ---------------------------------------------------------------------------
    Description:
        The function permits to know the name of the antennas available on the reader.
        Names are formatted according to CAEN RFID communication protocol.
*/
CAENRFIDErrorCodes CAENRFID_GetReadPoints(CAENRFIDReader* reader,
                                char** Antennas[],
                                int16_t* numAnt);
```

```
/*
   CAENRFID_GetSourceNames.
   ---------------------------------------------------------------------------
   Parameters:
      [in]  Handle        : The reader data structure that identifies the device.
      [out] Sources       : The names of the logical source avalaible on the reader.
      [out] numSrc        : The number of string in logical sources.
   ---------------------------------------------------------------------------
   Returns:
      An error code about the execution of the function.
   ---------------------------------------------------------------------------
   Description:
      The function permits to know the name of the logical sources available on the reader.
      Names are formatted according to CAEN RFID communication protocol.
*/
CAENRFIDErrorCodes CAENRFID_GetSourceNames(CAENRFIDReader* reader,
                                 char** Sources[],
                                 int16_t* numSrc);


/*
   CAENRFID_GetFirmwareRelease.
   ---------------------------------------------------------------------------
   Parameters:
      [in]  reader        : The reader data structure that identifies the device.
      [out] FWRel         : Returns the firmware release of the device.
   ---------------------------------------------------------------------------
   Returns:
      An error code about the execution of the function.
```

```
    ----------------------------------------------------------------------------
    Description:
        Permits to read the firmware release loaded into the device.
*/
CAENRFIDErrorCodes CAENRFID_GetFirmwareRelease(CAENRFIDReader* reader,
                                    char* FWRel);


/*
    CAENRFID_GetReaderInfo
    ----------------------------------------------------------------------------
    Parameters:
        [in]  reader        : The reader data structure that identifies the device.
        [out] Model         :  Returns the model of the reader.
        [out] SerialNum     : Returns the Serial number of the reader.
    ----------------------------------------------------------------------------
    Returns:
        An error code about the execution of the function.
    ----------------------------------------------------------------------------
    Description:
        Permits to read the Model and the Serial number of the Reader.
*/
CAENRFIDErrorCodes CAENRFID_GetReaderInfo(CAENRFIDReader* reader,
                                 char* Model,
                                 char* SerialNum);


/*
    CAENRFID_SetProtocol.
    ----------------------------------------------------------------------------
    Parameters:
```

```
    [in]  reader        : The reader data structure that identifies the device.
    [in]  Proto         : The tag protocol to be set in the reader.
  ---------------------------------------------------------------------------
  Returns:
    An error code about the execution of the function.
  ---------------------------------------------------------------------------
  Description:
    The function permits to change the tag protocol used by the reader.
*/
CAENRFIDErrorCodes CAENRFID_SetProtocol(CAENRFIDReader* reader,
                                        CAENRFIDProtocol Proto);

/*
  CAENRFID_GetProtocol.
  ---------------------------------------------------------------------------
  Parameters:
    [in]  reader        : The reader data structure that identifies the device.
    [out] Proto         : The tag protocol to be set in the reader.
  ---------------------------------------------------------------------------
  Returns:
    An error code about the execution of the function.
  ---------------------------------------------------------------------------
  Description:
    The function permits to know what tag protocol is used by the reader.
*/
CAENRFIDErrorCodes CAENRFID_GetProtocol(CAENRFIDReader* reader,
                                        CAENRFIDProtocol* Proto);

/*
```

```
    CAENRFID_SetPower.
    --------------------------------------------------------------------------
    Parameters:
      [in]  reader        : The reader data structure that identifies the device.
      [in]  Power         : RF field power expressed in mW.
    --------------------------------------------------------------------------
    Returns:
      An error code about the execution of the function.
    --------------------------------------------------------------------------
    Description:
      The function permits to set the RF field power of the reader.
*/
CAENRFIDErrorCodes CAENRFID_SetPower(CAENRFIDReader* reader,
                                   uint32_t Power);


/*
    CAENRFID_GetPower.
    --------------------------------------------------------------------------
    Parameters:
      [in]  reader        : The reader data structure that identifies the device.
      [out] Power         : The RF field power of the reader expressed in mW.
    --------------------------------------------------------------------------
    Returns:
      An error code about the execution of the function.
    --------------------------------------------------------------------------
    Description:
      The function returns the reader RF field power value.
*/
CAENRFIDErrorCodes CAENRFID_GetPower(CAENRFIDReader* reader,
                              uint32_t* Power);


/*
   CAENRFID_AddReadPoint.
   ------------------------------------------------------------------------------
   Parameters:
      [in]  reader         : The reader data structure that identifies the device.
      [in]  SourceName     : The name of the Logical Source.
      [in]  ReadPoint      : The name of the Read Point.
   ------------------------------------------------------------------------------
   Returns:
      An error code about the execution of the function.
   ------------------------------------------------------------------------------
   Description:
      The function permits to add a read point (antenna) to a logical source.
*/
CAENRFIDErrorCodes CAENRFID_AddReadPoint(CAENRFIDReader* reader,
                                char* SourceName,
                                char* ReadPoint);


/*
   CAENRFID_RemoveReadPoint.
   ------------------------------------------------------------------------------
   Parameters:
      [in]  reader         : The reader data structure that identifies the device.
      [in]  SourceName     : The name of the Logical Source.
      [in]  ReadPoint      : The name of the Read Point.
   ------------------------------------------------------------------------------
   Returns:
    An error code about the execution of the function.
  ---------------------------------------------------------------------------
  Description:
    The function permits to remove a read point (antenna) to a logical source.
*/
CAENRFIDErrorCodes CAENRFID_RemoveReadPoint(CAENRFIDReader* reader,
                                            char* SourceName,
                                            char* ReadPoint);


/*
  CAENRFID_isReadPointPresent.
  ---------------------------------------------------------------------------
  Parameters:
    [in]  reader        : The reader data structure that identifies the device.
    [in]  SourceName    : The name of the Logical Source.
    [in]  ReadPoint     : The name of the Read Point.
    [out] isPresent     : A flag indicating if the read point is associated
                          to the specified source.
  ---------------------------------------------------------------------------
  Returns:
    An error code about the execution of the function.
  ---------------------------------------------------------------------------
  Description:
    The function permits to check if a read point is associated to a
    specified logical source. That is, whether the read point is used within
    a read cycle performed in the source.
*/
CAENRFIDErrorCodes CAENRFID_isReadPointPresent(CAENRFIDReader* reader,
                                               char* ReadPoint,
                                    char* SourceName,
                                    uint16_t* isPresent);


/*
  CAENRFID_GetReadPointStatus.
  ---------------------------------------------------------------------------
  Parameters:
    [in]  reader         : The reader data structure that identifies the device.
    [in]  ReadPoint      : The name of the Read Point.
    [out] Status         : The status of the read point.
  ---------------------------------------------------------------------------
  Returns:
    An error code about the execution of the function.
  ---------------------------------------------------------------------------
  Description:
    The function permits to check the status of a read point.
*/
CAENRFIDErrorCodes CAENRFID_GetReadPointStatus(CAENRFIDReader* reader,
                                    char* ReadPoint,
                                    CAENRFIDReadPointStatus* Status);


/*
  CAENRFID_MatchReadPointImpedance.
  ---------------------------------------------------------------------------
  Parameters:
    [in]  reader         : The reader data structure that identifies the device.
    [in]  ReadPoint      : The name of the ReadPoint to apply impedance automatching.
    [out] value          : The result of the automatching result operation.
  ---------------------------------------------------------------------------
```

```
    Returns:
        An error code about the execution of the function.
    ---------------------------------------------------------------------------
    Description:
        The function matches the antenna impedance passed in ReadPoint parameter.
*/
CAENRFIDErrorCodes CAENRFID_MatchReadPointImpedance(CAENRFIDReader* reader,
                                                    char* ReadPoint,
                                                    float* value);


/*
    CAENRFID_SetSourceConfiguration.
    ---------------------------------------------------------------------------
    Parameters:
        [in]  reader        : The reader data structure that identifies the device.
        [in]  SourceName    : The Name of the Logical Source.
        [in]  Parameter     : The parameter of Logical Source to configure.
        [in]  Value         : The the value of the parameter.
    ---------------------------------------------------------------------------
    Returns:
        An error code about the execution of the function.
    ---------------------------------------------------------------------------
    Description:
        The function permits to configure the Logical Source.
*/
CAENRFIDErrorCodes CAENRFID_SetSourceConfiguration(CAENRFIDReader* reader,
                                                   char* SourceName,
                                                   CAENRFID_SOURCE_Parameter Parameter,
                                                   uint32_t Value);
```

```
/*
   CAENRFID_GetSourceConfiguration.
   ------------------------------------------------------------------------------
   Parameters:
      [in]  reader         : The reader data structure that identifies the device.
      [in]  SourceName     : The Name of the Logical Source.
      [in]  Parameter      : The parameter of Logical Source to configure.
      [out] Value          : The the value of the parameter.
   ------------------------------------------------------------------------------
   Returns:
      An error code about the execution of the function.
   ------------------------------------------------------------------------------
   Description:
      The function permits to get the value of the Logical Source configuration.
*/
CAENRFIDErrorCodes CAENRFID_GetSourceConfiguration(CAENRFIDReader* reader,
                                    char* SourceName,
                                    CAENRFID_SOURCE_Parameter Parameter,
                                    uint32_t* Value);


/*
   CAENRFID_SetRS232.
   ------------------------------------------------------------------------------
   Parameters:
      [in]  reader         : The reader data structure that identifies the device.
      [in]  Baudrate       : The baudrate value.
      [in]  DataBits       : The databit value.
      [in]  StopBits       : The stopbit value.
```

```
    [in]  Parity        : The parity value.
    [in]  FlowControl   : The flowcontrol value.
  ------------------------------------------------------------------------------
  Returns:
    An error code about the execution of the function.
  ------------------------------------------------------------------------------
  Description:
    The function permits to configure the serial communication of the reader.
*/
CAENRFIDErrorCodes CAENRFID_SetRS232(CAENRFIDReader* reader,
                                     uint32_t Baudrate,
                                     uint32_t DataBits,
                                     uint32_t StopBits,
                                     CAENRFID_RS232_Parity Parity,
                                     CAENRFID_RS232_FlowControl FlowControl);

/*
  CAENRFID_SetBitRate.
  ------------------------------------------------------------------------------
  Parameters:
    [in]  reader        : The reader data structure that identifies the device.
    [in]  Bitrate       : link-profile setting.
  ------------------------------------------------------------------------------
  Returns:
    An error code about the execution of the function.
  ------------------------------------------------------------------------------
  Description:
    The function permits to choose the modulation (the bitrate of the
    transmission and reception).
```

```
*/
CAENRFIDErrorCodes CAENRFID_SetBitrate(CAENRFIDReader* reader,
                                   CAENRFID_Bitrate Bitrate);


/*
   CAENRFID_GetBitRate.
   ---------------------------------------------------------------------------
   Parameters:
      [in]  reader        : The reader data structure that identifies the device.
      [out] Bitrate        : Modulation setting.
   ---------------------------------------------------------------------------
   Returns:
      An error code about the execution of the function.
   ---------------------------------------------------------------------------
   Description:
      The function permits to retrieve the modulation (the bitrate of the
      transmission and reception).
*/
CAENRFIDErrorCodes CAENRFID_GetBitrate(CAENRFIDReader* reader,
                                   CAENRFID_Bitrate* Bitrate);


/*
   CAENRFID_SetFHSSMode.
   ---------------------------------------------------------------------------
   Parameters:
      [in]  reader        : The reader data structure that identifies the device.
      [in]  FHSSMode       : A zero value to disable FHSS, non-zero value
                      to enable FHSS.
   ---------------------------------------------------------------------------
```

```
    Returns:
        An error code about the execution of the function.
    ----------------------------------------------------------------------------
    Description:
        The function permits to enable/disable the frequency hopping mode of
        the reader.
*/
CAENRFIDErrorCodes CAENRFID_SetFHSSMode(CAENRFIDReader* reader,
                                        uint16_t FHSSMode);

/*
    CAENRFID_GetFHSSMode.
    ----------------------------------------------------------------------------
    Parameters:
        [in]  reader       : The reader data structure that identifies the device.
        [out] FHSSMode     : A zero value if the FHSS is disabled, non-zero value
                             if it is enabled.
    ----------------------------------------------------------------------------
    Returns:
        An error code about the execution of the function.
    ----------------------------------------------------------------------------
    Description:
        The function indicates the activation of the frequency hopping mode of
        the reader.
*/
CAENRFIDErrorCodes CAENRFID_GetFHSSMode(CAENRFIDReader* reader,
                                        uint16_t* FHSSMode);

/*
```

```
   CAENRFID_SetRFChannel.
   ---------------------------------------------------------------------------
   Parameters:
      [in]  reader        : The reader data structure that identifies the device.
      [in]  RFChannel      : The RF Channel
   ---------------------------------------------------------------------------
   Returns:
      An error code about the execution of the function.
   ---------------------------------------------------------------------------
   Description:
      The function permits to set the RF Channel.
*/
CAENRFIDErrorCodes CAENRFID_SetRFChannel(CAENRFIDReader* reader,
                                         uint16_t RFChannel);


/*
   CAENRFID_GetRFChannel.
   ---------------------------------------------------------------------------
   Parameters:
      [in]  reader        : The reader data structure that identifies the device.
      [out] RFChannel      : The RF Channel
   ---------------------------------------------------------------------------
   Returns:
      An error code about the execution of the function.
   ---------------------------------------------------------------------------
   Description:
      The function permits to retrieve the RF Channel.
*/
CAENRFIDErrorCodes CAENRFID_GetRFChannel(CAENRFIDReader* reader,
                                    uint16_t* RFChannel);


/*
  CAENRFID_SetRFRegulation.
  ------------------------------------------------------------------------------
  Parameters:
    [in]  reader        : The reader data structure that identifies the device.
    [in]  RFRegulation   : The RF Regulation
  ------------------------------------------------------------------------------
  Returns:
    An error code about the execution of the function.
  ------------------------------------------------------------------------------
  Description:
    The function permits to set the RF Regulation.
*/
CAENRFIDErrorCodes CAENRFID_SetRFRegulation(CAENRFIDReader* reader,
                                    CAENRFIDRFRegulations RFRegulation);


/*
  CAENRFID_GetRFRegulation.
  ------------------------------------------------------------------------------
  Parameters:
    [in]  reader        : The reader data structure that identifies the device.
    [out] RFRegulation   : The RF Regulation
  ------------------------------------------------------------------------------
  Returns:
    An error code about the execution of the function.
  ------------------------------------------------------------------------------
  Description:
        The function permits to retrieve the RF Regulation.
*/
CAENRFIDErrorCodes CAENRFID_GetRFRegulation(CAENRFIDReader* reader,
                                      CAENRFIDRFRegulations* RFRegulation);


/*
    CAENRFID_SetIO.
    ----------------------------------------------------------------------------
    Parameters:
        [in]  reader        : The reader data structure that identifies the device.
        [in]  IORegister     : The IO Register value.
    ----------------------------------------------------------------------------
    Returns:
        An error code about the execution of the function.
    ----------------------------------------------------------------------------
    Description:
        The function permits to write the IO register.
*/
CAENRFIDErrorCodes CAENRFID_SetIO(CAENRFIDReader* reader,
                              uint32_t IORegister);


/*
    CAENRFID_GetIO.
    ----------------------------------------------------------------------------
    Parameters:
        [in]  reader        : The reader data structure that identifies the device.
        [out] IORegister     : The current IO Register.
    ----------------------------------------------------------------------------
    Returns:
        An error code about the execution of the function.
    ----------------------------------------------------------------------------
    Description:
        The function permits to read the IO register.
*/
CAENRFIDErrorCodes CAENRFID_GetIO(CAENRFIDReader* reader,
                                  uint32_t* IORegister);

/*
    CAENRFID_SetIODirection.
    ----------------------------------------------------------------------------
    Parameters:
        [in]  reader        : The reader data structure that identifies the device.
        [in]  IODirection    : The IO Direction value.
    ----------------------------------------------------------------------------
    Returns:
        An error code about the execution of the function.
    ----------------------------------------------------------------------------
    Description:
        The function permits to set the IO Direction for the IORegister.
*/
CAENRFIDErrorCodes CAENRFID_SetIODirection(CAENRFIDReader* reader,
                                           uint32_t IODirection);

/*
    CAENRFID_GetIODirection.
    ----------------------------------------------------------------------------
    Parameters:
        [in]  reader        : The reader data structure that identifies the device.
        [out] IODirection    : The current IO Direction.
    ---------------------------------------------------------------------------
    Returns:
        An error code about the execution of the function.
    ---------------------------------------------------------------------------
    Description:
        The function permits to read the IO direction of IORegister.
*/
CAENRFIDErrorCodes CAENRFID_GetIODirection(CAENRFIDReader* reader,
                                           uint32_t* IODirection);


/*
    CAENRFID_ReadTagData_EPC_C1G2.
    ---------------------------------------------------------------------------
    Parameters:
        [in]  reader        : The reader data structure that identifies the device.
        [in]  Tag           : The tag to be read.
        [in]  Bank          : The memory Bank of EPC C1G2 Tag
        [in]  ByteAddress    : The byte address of the memory to read.
        [in]  ByteLength     : The number of bytes to read.
        [out] Data          : The data read from the tag's memory.
        [in]  AccessPassword : The tag Access password. If 0, no password is used.
    ---------------------------------------------------------------------------
    Returns:
        An error code about the execution of the function.
    ---------------------------------------------------------------------------
    Description:
        This function allows to read ByteLength bytes from the bank memory,
        specified by Bank, of a specific tag identified by the
        Tag at the address specified by ByteAddress.
*/
CAENRFIDErrorCodes CAENRFID_ReadTagData_EPC_C1G2(CAENRFIDReader* reader,
                                                 CAENRFIDTag* Tag,
                                                 uint16_t Bank,
                                                 uint16_t ByteAddress,
                                                 uint16_t ByteLength,
                                                 uint8_t* Data,
                                                 uint32_t AccessPassword);


/*
    CAENRFID_WriteTagData_EPC_C1G2.
    ---------------------------------------------------------------------------
    Parameters:
        [in]  reader         : The reader data structure that identifies the device.
        [in]  Tag            : The tag to be written.
        [in]  Bank           : The memory Bank of EPC C1G2 Tag
        [in]  ByteAddress    : The byte address of the memory to write.
        [in]  ByteLength     : The number of bytes to write.
        [in]  Data           : The data to write in the tag's memory.
        [in]  AccessPassword : The tag Access password. If 0, no password is used.
    ---------------------------------------------------------------------------
    Returns:
        An error code about the execution of the function.
    ---------------------------------------------------------------------------
    Description:
        This function allows to write ByteLength bytes to the bank memory,
        specified by Bank, of a specific tag identified by the
        Tag at the address specified by ByteAddress.
```

```
*/
CAENRFIDErrorCodes CAENRFID_WriteTagData_EPC_C1G2(CAENRFIDReader* reader,
                              CAENRFIDTag* Tag,
                              uint16_t Bank,
                              uint16_t ByteAddress,
                              uint16_t ByteLength,
                              uint8_t* Data,
                              uint32_t AccessPassword);


/*
  CAENRFID_LockTag_EPC_C1G2.
  ------------------------------------------------------------------------
  Parameters:
    [in]  reader        : The reader data structure that identifies the device.
    [in]  Tag           : The tag.
    [in]  Payload       : The payload of the tag's memory.
    [in]  AccessPassword : The tag Access Password. If 0, no password is used.
  ------------------------------------------------------------------------
  Returns:
    An error code about the execution of the function.
  ------------------------------------------------------------------------
  Description:
    This function allows to secure lock the memory of a
    specific tag identified by the Tag.
*/
CAENRFIDErrorCodes CAENRFID_LockTag_EPC_C1G2(CAENRFIDReader* reader,
                           CAENRFIDTag *Tag,
                           uint32_t Payload,
                           uint32_t AccessPassword);
```

```
/*
   CAENRFID_KillTag_EPC_C1G2.
   -----------------------------------------------------------------------------
   Parameters:
      [in]  reader        : The reader data structure that identifies the device.
      [in]  Tag           : The tag to kill.
      [in]  Password      : The tag Kill Password.
   -----------------------------------------------------------------------------
   Returns:
      An error code about the execution of the function.
   -----------------------------------------------------------------------------
   Description:
      The function permits to kill an EPC Class 1 Gen 2 tag.
*/
CAENRFIDErrorCodes CAENRFID_KillTag_EPC_C1G2(CAENRFIDReader* reader,
                                  CAENRFIDTag *Tag,
                                  uint32_t Password);


/*
   CAENRFID_ProgramID_EPC_C1G2.
   -----------------------------------------------------------------------------
   Parameters:
      [in]  reader        : The reader data structure that identifies the device.
      [in]  Tag           : Contains the tag EPC C1G2 ID to be programmed.
      [in]  nsi           : The NSI value for the EPC C1G2.
      [in]  AccessPassword : The tag Access Password. If 0, no password is used.
   -----------------------------------------------------------------------------
   Returns:
      An error code about the execution of the function.
    ---------------------------------------------------------------------------
    Description:
      The function permits to program the EPC Code of an EPC Class 1 Gen 2 tag.
*/
CAENRFIDErrorCodes CAENRFID_ProgramID_EPC_C1G2(CAENRFIDReader* reader,
                                 CAENRFIDTag *Tag,
                                 uint16_t nsi,
                                 uint32_t AccessPassword);


/*
    CAENRFID_CustomCommand_EPC_C1G2.
    ---------------------------------------------------------------------------
    Parameters:
      [in]  reader         : The reader data structure that identifies the device.
      [in]  Tag            : The tag.
      [in]  SubCmd         : The subcommand.
      [in]  TxLen          : The number of bytes to send to the tag.
      [in]  Data           : The data to be sent to the tag.
      [in]  RxLen          : The number of bytes to be received from the tag.
      [in]  AccessPassword : The tag Access Password. If 0, no password is used.
      [out] TRData         : The data received from the tag.
    ---------------------------------------------------------------------------
    Returns:
      An error code about the execution of the function.
    ---------------------------------------------------------------------------
    Description:
      Special function to implement custom commands.
*/
```

```
CAENRFIDErrorCodes CAENRFID_CustomCommand_EPC_C1G2(CAENRFIDReader* reader,
                                        CAENRFIDTag *Tag,
                                        uint8_t SubCmd,
                                        uint16_t TxLen,
                                        uint8_t *Data,
                                        uint16_t RxLen,
                                        uint32_t AccessPassword,
                                        uint8_t *TRData);


/*
   CAENRFID_InventoryTag.
   ---------------------------------------------------------------------------
   Parameters:
      [in]  reader         : The reader data structure that identifies the device.
      [in]  SourceName     : The name that identifies the Logical Source.
      [in]  Bank           : The bank where apply the mask.
      [in]  MaskBitAddress : The position in bit from where starting to compare the mask
                        to the choosen bank.
      [in]  MaskBitLength  : The length in bit of the significative part of the Mask.
      [in]  Mask           : The array containing the Mask.
      [in]  MaskLen        : The number of bytes passed in Mask.
      [in]  flag           : A bitmask that indicates the mode of inventory and the
                        retrieving of RSSI, TID, XPC and PC values.
      [out] TagList        : A List containing the tags read (only for
                        an inventory of type NOT framed and NOT continuous).
      [out] Size           : Returns the number of tags in the list (only for
                        an inventory of type NOT framed and NOT continuous).
   ---------------------------------------------------------------------------
   Returns:
      An error code about the execution of the function.
   ---------------------------------------------------------------------------
   Description:
      This function is used to perform a simple inventory round or to start a cycle
      of continuous plus framed inventory rounds. The inventory is performed on
      all antennas belonging to the logical source SourceName and will apply
      the mask parameters to the founded tags, returning only those tags whose
      relative bank matches the mask.

      If the continuous and framed mode of inventory is NOT selected in the flag
      parameter, the TagList list contains the IDs of the detected tags, together
      with other information  related to the single ID such as the antenna on which
      the ID was detected and the format of the ID itself (see CAENRFIDTag struct
      for the details).

      If the continuos and framed mode of inventory is selected in flag, the
      reader performs a number of inventory rounds dependant on the value
      of the CONFIG_READCYCLE parameter of the logical source SourceName (see
      function CAENRFID_SetSourceConfiguration).
      In this case TagList and Size parameters are ignored, user should call the
      CAENRFID_GetFramedTag function to retrieve tags detected during an inventory
      of such type, as well as to detect its ending.
*/
CAENRFIDErrorCodes CAENRFID_InventoryTag(CAENRFIDReader* reader,
                                         char* SourceName,
                                         uint16_t Bank,
                                         uint16_t MaskBitAddress,
                                         uint16_t MaskBitLength,
                                         uint8_t* Mask,
                               uint16_t MaskLen,
                               uint16_t flag,
                               CAENRFIDTagList** TagList,
                               uint16_t* Size);

/*
  CAENRFID_GetFramedTag.
  ------------------------------------------------------------------------------
  Parameters:
    [in]  reader         : The reader data structure that identifies the device.
    [out] has_tag        : Contains information about whether a tag was detected
                      and received from the reader or not.
    [out] Tag            : The detected tag, if present.
    [out] has_result_code : Contains information about whether the inventory
                      rounds ended and a result code was returned by the
                      reader or not.
  ------------------------------------------------------------------------------
  Returns:
    An error code about the execution of the function.
  ------------------------------------------------------------------------------
  Description:
    This function is used to receive a tag Tag detected by the reader during
    a framed plus continuous inventory. The function also detects the ending
    of such an inventory.
    The framed plus continuous inventory must be started using the
    CAENRFID_InventoryTag function.
*/
CAENRFIDErrorCodes CAENRFID_GetFramedTag(CAENRFIDReader* reader,
                               bool* has_tag,
                                        CAENRFIDTag* Tag,
                                        bool* has_result_code);


/*
    CAENRFID_InventoryAbort.
    --------------------------------------------------------------------------
    Parameters:
        [in]  reader        : The reader data structure that identifies the device.
    --------------------------------------------------------------------------
    Returns:
        An error code about the execution of the function.
    --------------------------------------------------------------------------
    Description:
        This function tries to stop an ongoing framed plus continuous inventory.
        To detect the ending of such inventory the user should use the
        CAENRFID_GetFramedTag function.
*/
CAENRFIDErrorCodes CAENRFID_InventoryAbort(CAENRFIDReader* reader);


#endif /* SRC_LIB_CAENRFIDLIB_LIGHT_H_ */
```