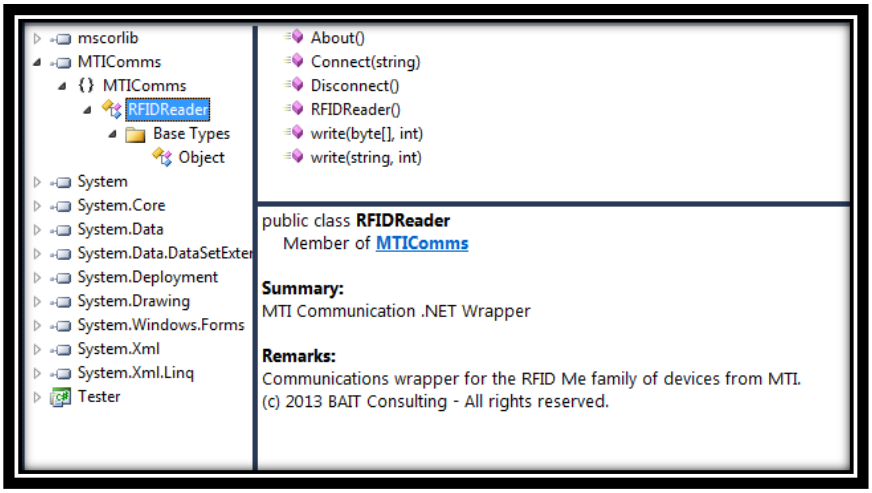
MTI Communication .NET DLL



Spring 2013

BAIT CONSULTING

Introduction

MTI's range of USB based RFID readers all support the USB HID communications standard. However, in .NET, communications can be difficult at best. This DLL will allow for easy communications between your software, the MTI supplied "Transfer.DLL" file and the physical reader.

The setup file will place all files in "c:\Program Files\BAIT Consulting\MTIComms" unless another location is selected at setup. When adding the DLLs in your project, you will need to reference them in this location.

The general process for using the SDK and the hardware are as follows:

1. Plug the reader into the computer
2. In your project:
    a. Add MTIComms.DLL as a project reference.
    b. Add Transfer.DLL as a project resource – make sure it is set to "always copy" under properties.
3. From your program:
    a. Connect to the reader (to ensure it is plugged in)
    b. Initiate a read (to see what tags exist)
    c. Use the results in a business process (to solve your specific problem)
4. When you deploy (via Setup file or manually):
    a. Make sure MTIComms.DLL is included in package.
    b. Make sure Transfer.DLL is included in the package.
    c. Make lots of money!

Functions

Quite simply, the MTIComms.DLL will take the data that you want to send to the reader and will pass it through the "Transfer.dll". The result of the data that was written to the reader will then be passed back in a returned byte array that will have to be interpreted for meaning. If a command fails or is invalid, a byte array with all zeros will be returned.

There is no "read" function as would normally be expected. This is because the MTI readers require a message to be sent to trigger the read. As such, the read functionality has been embedded into the write function.

Please note that the returned byte array in the write function returns a decimal value that has to be converted to hexadecimal to match the MTI documentation!

About

Usage: About()
Description: Reports current SDK version and identified reader.
Parameters: None
Returns: *(String)* "MTI Communication .NET DLL, ver x.x, Current reader set to xxxx".
Comments: None.

Connect

Usage: Connect(string myReader)
Description: Attempts to connect to the indicated RFID reader.
Parameters:
myReader (string): The type of reader that is to be connected to. Currently the only option is "RFIDME". As additional RFID readers are introduced, this list will increase.
Returns: *(String)* "Connected", "Reader not connected", or the specific error message if the connection attempt fails.
Comments: None.

Disconnect

Usage: Disconnect()
Description: Attempts to disconnect from the RFID Me reader.
Parameters: None
Returns: *(String)* "Disconnected" or "Problem disconnecting – " and the specific error message received while trying to disconnect.
Comments: Disconnect is not actually required, but is suggested to help control potential memory leaks.

Write

Usage: Write(string ByteString, int ResponseLength)
Description: Writes the identified byte stream
Parameters:
ByteString(string): The command string, without spaces, to be sent to the reader in a single string.
ResponseLength (int): The number of bytes to return in the response array.
Returns: *(Byte array)* An array of bytes will be returned. The length of the array will be the size that was indicated by the response length parameter.
Comments: **The returned result is a byte array**. This will need to be converted back to hexadecimal to match the MTI documentation.

Write (overload)

Usage: Write(byte[] ByteArray, int ResponseLength)
Description: Writes the identified byte stream
Parameters:
ByteArray(byte array): The command string, in the form of an array of bytes.
ResponseLength (int): The number of bytes to return in the response array.
Returns: *(Byte array)* An array of bytes will be returned. The length of the array will be the size that was indicated by the response length parameter.
Comments: **The returned result is a byte array**. This will need to be converted back to hexadecimal to match the MTI documentation.

For assistance, please email us at info@bait-consulting.com.