



Parallax USB RFID Reader

Parallax USB RFID Reader



USB-A to Mini-B Cable

125KHz Tags in different shapes

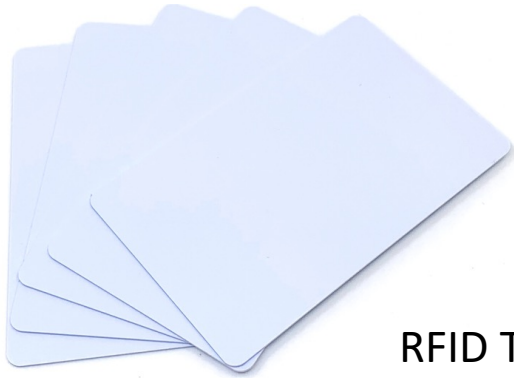
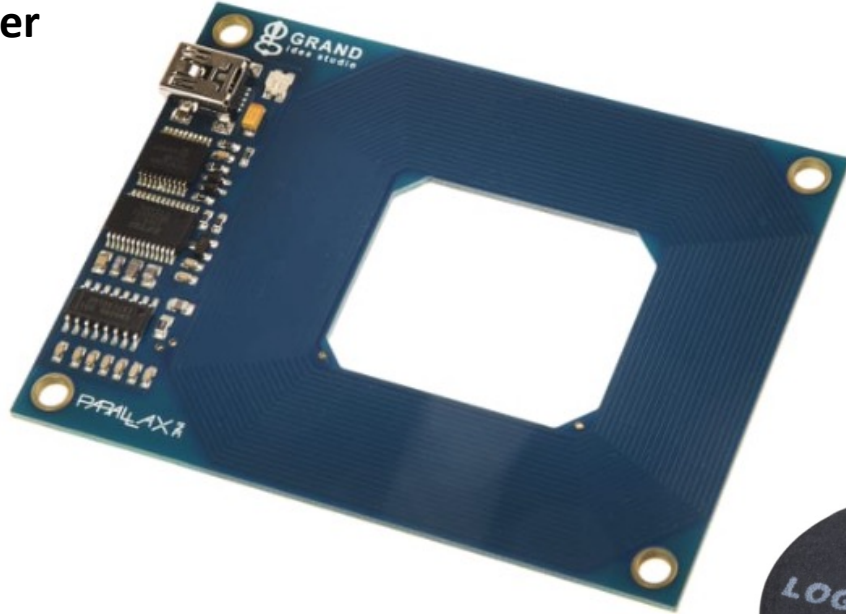
RFID Reader with built-in Antenna

RFID 125KHz

Reads 125kHz Tags with EM4100 protocol

Parallax Inc 28340 RFID Reader

USB RFID Reader



RFID Tags

RFID Tags



<https://www.parallax.com/product/rfid-card-reader-usb/>

<https://no.rs-online.com/web/p/rf-modules/7813061/>

Parallax USB RFID Reader

From Parallax USB RFID Reader Documentation

- It reads passive **125 kHz** RFID transponder tags
- The Parallax RFID Card Reader USB version can be connected directly to any PC, Macintosh, or Linux machine that has a USB port and the appropriate drivers installed. The module is powered from the host computer's USB port and uses an industry-standard **FTDI FT232R** device to provide the USB connectivity
- A visual indication of the state of the RFID Card Reader is given with the on-board LED. When the module is successfully powered-up and is in an idle state, the LED will be **GREEN**. When the module is in an active state searching for or communicating with a valid tag, the LED will be **RED**.
- The RFID Card Reader USB version is activated via the **DTR** line of the USB Virtual COM port. When the DTR line is set HIGH, the module will enter the active state. When the DTR line is set LOW, the module will enter the idle state.
- RFID Tag read distance of approximately 4 inches (**10cm**).

Parallax USB RFID Reader

Communication Protocol:

- The RFID Card Reader USB version transmits the data through the USB Virtual COM Port driver
- All communication is **8 data bits, no parity, 1 stop bit**, and least significant bit first (8N1) at **2400 bps**.
- When the RFID Card Reader is active and a valid RFID transponder tag is placed within range of the activated reader, the tag's unique ID will be transmitted as a **12-byte printable ASCII string** serially to the host in the following format:

Parallax USB RFID Reader

Communication Protocol:

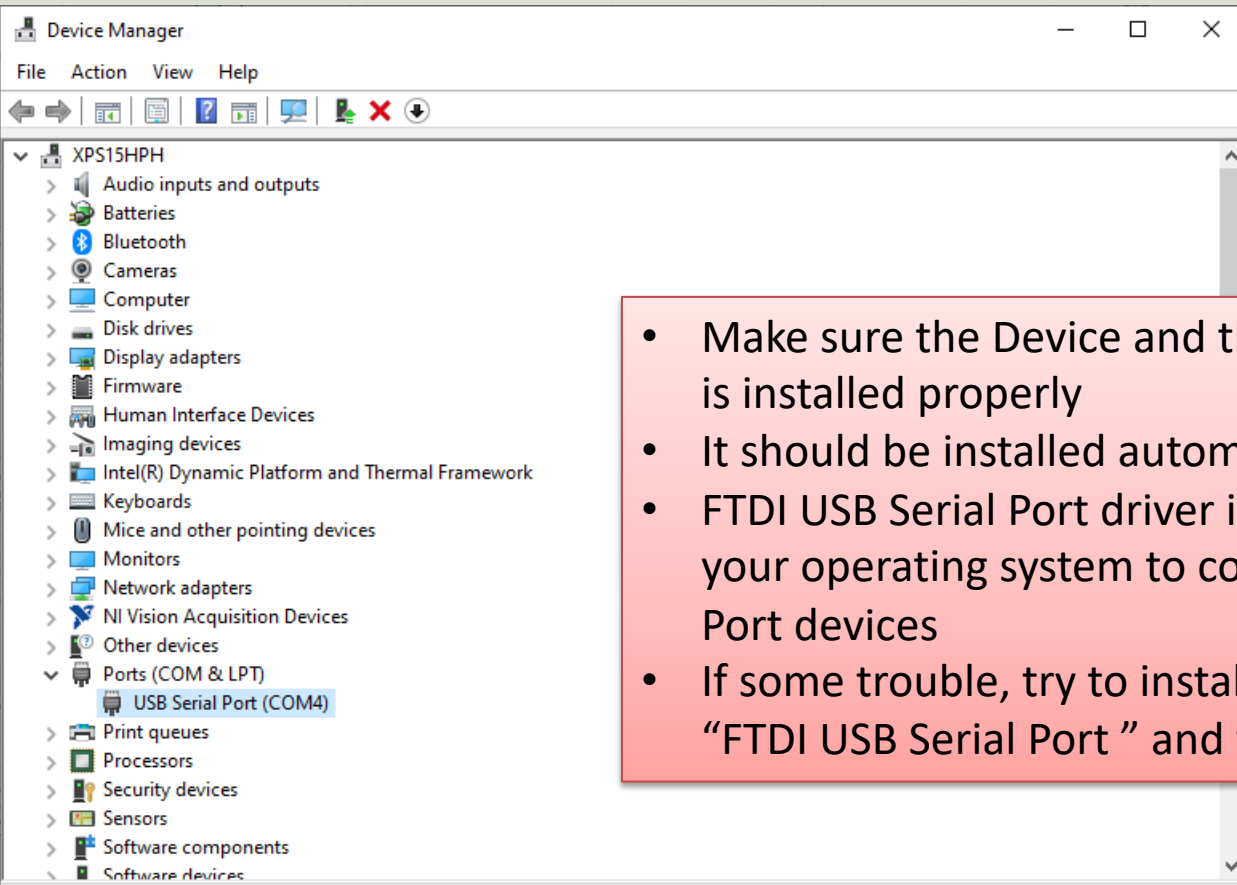
Start Byte (0x0A)	Unique ID Digit 1	Unique ID Digit 2	Unique ID Digit 3	Unique ID Digit 4	Unique ID Digit 5	Unique ID Digit 6	Unique ID Digit 7	Unique ID Digit 8	Unique ID Digit 9	Unique ID Digit 10	Stop Byte (0x0D)
----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	----------------------	-----------------------	---------------------

The **start byte** and **stop byte** are used to easily identify that a correct string has been received from the reader (they correspond to **line feed (\n)** and **carriage return (\r)** characters, respectively).

The **middle ten bytes** are the actual tag's unique ID.

For example, for a tag with a valid ID of 0F0184F07A, the following bytes would be sent: 0x0A, 0x30, 0x46, 0x30, 0x31, 0x38, 0x34, 0x46, 0x30, 0x37, 0x41, 0x0D.

Setup and Configuration



Device Manager

- Make sure the Device and the FTDI USB Serial Port driver is installed properly
- It should be installed automatically by Windows
- FTDI USB Serial Port driver is the software that helps your operating system to communicate with USB Serial Port devices
- If some trouble, try to install the driver manually (Google "FTDI USB Serial Port" and you will find it)



Serial Terminal Software

Hans-Petter Halvorsen

[Table of Contents](#)

Serial Terminal Software

Examples:

- RealTerm
- YAT
- Hterm
 - <https://www.der-hammer.info/pages/terminal.html>
- ..

Parallax USB RFID Reader

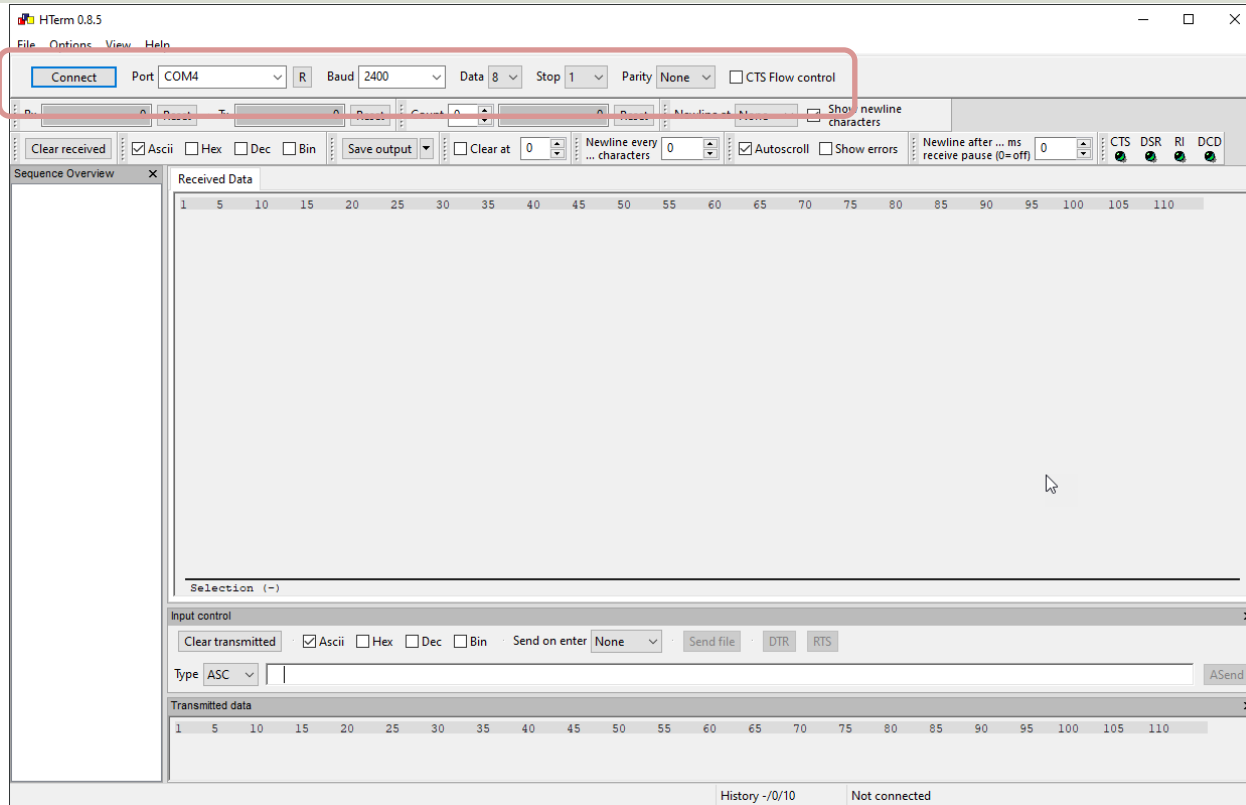
Communication Protocol for Parallax USB RFID Reader:

- Baud rate **2400 bps**
- **8** data bits
- **1** stop bit
- no parity

Parallax USB RFID Reader



HTerm



<https://www.der-hammer.info/pages/terminal.html>

HTerm – Retrieving Tag Id

The screenshot shows the HTerm 0.8.5 interface. The top panel displays connection settings: Port COM4, Baud 2400, Data 8, Stop 1, Parity None, and CTS Flow control. The middle panel shows Rx 444, Tx 0, and various control options like Ascii, Hex, Dec, Bin, Save output, Clear at, Newline every, Autoscroll, Show errors, and Newline after. The bottom panel shows the input control section with a red box around the DTR button. The main window displays received data in hexadecimal format, with a red box around the first 10 lines.

Received Data

1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	
v	0	8	0	0	2	9	C	B	9	B	v													
v	0	8	0	0	2	9	C	0	6	C	v													
v	0	8	0	0	2	9	C	0	6	C	v													
v	0	8	0	0	2	9	6	6	6	3	v													
v	0	8	0	0	2	9	7	F	0	2	v													
v	0	8	0	0	2	9	7	F	0	2	v													
v	0	8	0	0	2	9	6	6	6	3	v													
v	0	8	0	0	2	9	6	6	6	3	v													
v	0	8	0	0	2	9	6	6	6	3	v													
v	0	8	0	0	2	9	6	6	6	3	v													
v	0	8	0	0	2	9	6	6	6	3	v													

Input control

Clear transmitted Ascii Hex Dec Bin Send on enter None Send file **DTR** RTS

Type ASC

Transmitted data

1	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115
---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----

History -/0/10 Connected to COM4 (b:2400 d:8 s:1 p:None)



Code Examples

Parallax USB RFID Reader



Code Examples

Note!

- The examples provided can be considered as a “proof of concept”
- The sample code is very simplified for clarity and doesn't necessarily represent best practices.



Python

Python Example

The image shows a screenshot of the Thonny Python IDE. The main window displays a Python script named `rfid_ex.py` with the following code:

```
1 import serial
2 import time
3
4 ser = serial.Serial('COM4', 2400, timeout=1)
5
6 response = ser.read(12)
7 if response != "":
8     print(response)
9
10 ser.close()
```

Below the code editor is a Shell window showing the execution of the script:

```
Python 3.7.9 (bundled)
>>> %Run rfid_ex.py
b'\n0800296663\r'
>>>
```

On the right side, there is an Assistant panel with a Warnings section:

Warnings
May be ignored if you are happy with your program.

[rfid_ex.py](#)

[Line 2](#) : Unused import time

[Was it helpful or confusing?](#)

Python 3.7.9

Python Example

The screenshot shows the Thonny Python IDE interface. The main editor displays a Python script named `rfid_loop_ex.py` with the following code:

```
1 import serial
2 import time
3
4 ser = serial.Serial('COM4', 2400, timeout=1)
5
6 while True:
7     response = ser.read(12)
8     if response != "":
9         print(response)
10    time.sleep(1)
11
12 ser.close()
```

The Shell window at the bottom shows the output of the script, displaying a series of hexadecimal byte strings:

```
b'\n0800296663\r'
b'\n0800296663\r'
b'\n0800296663\r'
b'\n0800296663\r'
b'\n0800296663\r'
b'\n0800296663\r'
b'\n0800296663\r'
b'\n0800296663\r'
b'\n0800296663\r'
b'\n0800296663\r'
b'\n0800296663\r'
b'\n080029c06c\r'
```

The Assistant panel on the right shows a warning message:

Warnings
May be ignored if you are happy with your program.
[rfid_ex.py](#)
Line 2 : Unused import time
[Was it helpful or confusing?](#)

The status bar at the bottom right indicates the Python version: Python 3.7.9.



LabVIEW

Hans-Petter Halvorsen

[Table of Contents](#)

LabVIEW Example

The screenshot displays the LabVIEW front panel for 'RFID Reader.vi'. The interface includes a menu bar (File, Edit, View, Project, Operate, Tools, Window, Help), a toolbar with various icons, and a search field. The main panel is divided into three sections: 'RFIDTag' with a text box containing '0800297F02', 'Bytes Array' with a row of 12 hexadecimal digit boxes (A, 30, 38, 30, 30, 32, 39, 37, 46, 30, 32, D), and 'Error Information' with a status indicator (green circle), a code field (1073676), a source dropdown (VISA Read in RFID Reader.vi), and a 'Stop' button with a red square icon.

RFID Reader.vi Front Panel

File Edit View Project Operate Tools Window Help

15pt Application Font Search

RFIDTag

0800297F02

Bytes Array

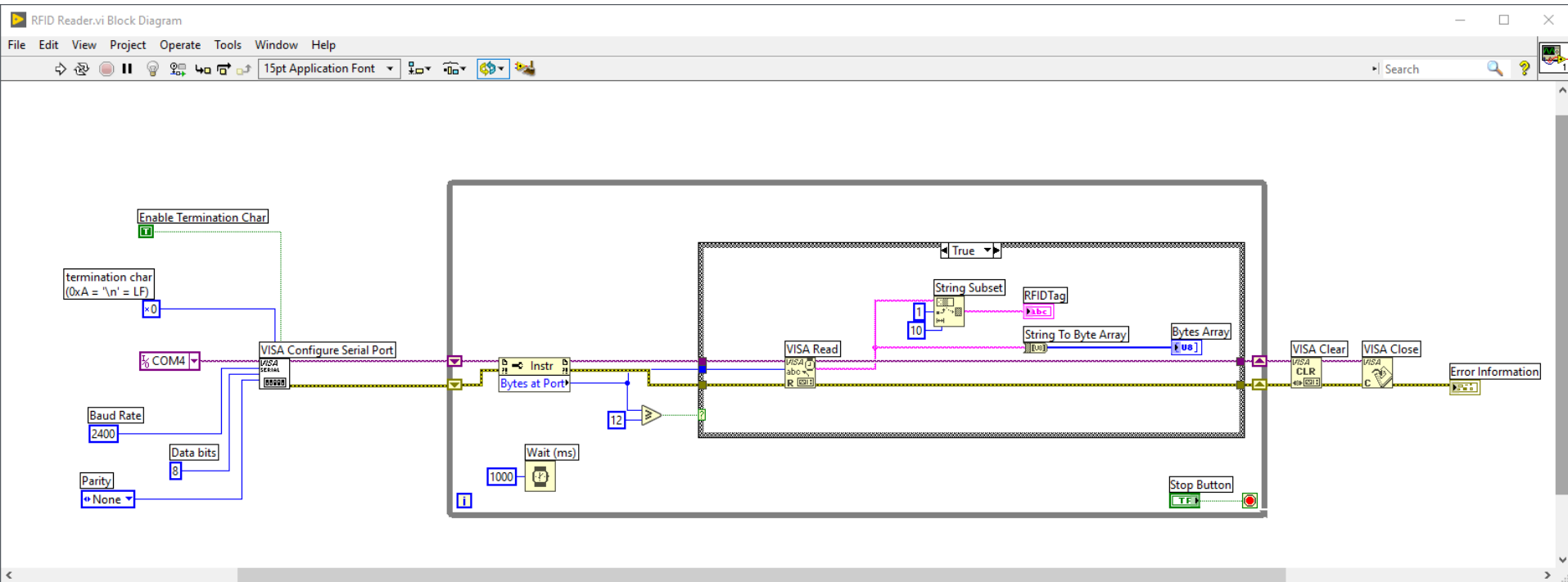
A	30	38	30	30	32	39	37	46	30	32	D
---	----	----	----	----	----	----	----	----	----	----	---

Error Information

status	code
	1073676
source	VISA Read in RFID Reader.vi

Stop

LabVIEW Example





Visual Studio/C#

Read RFID Tag with C#

```
using System.IO.Ports;
```

```
SerialPort port = new System.IO.Ports.SerialPort("COM4", 2400, System.IO.Ports.Parity.None,  
8, System.IO.Ports.StopBits.One);
```

```
port.Open();  
port.DtrEnable = true;
```

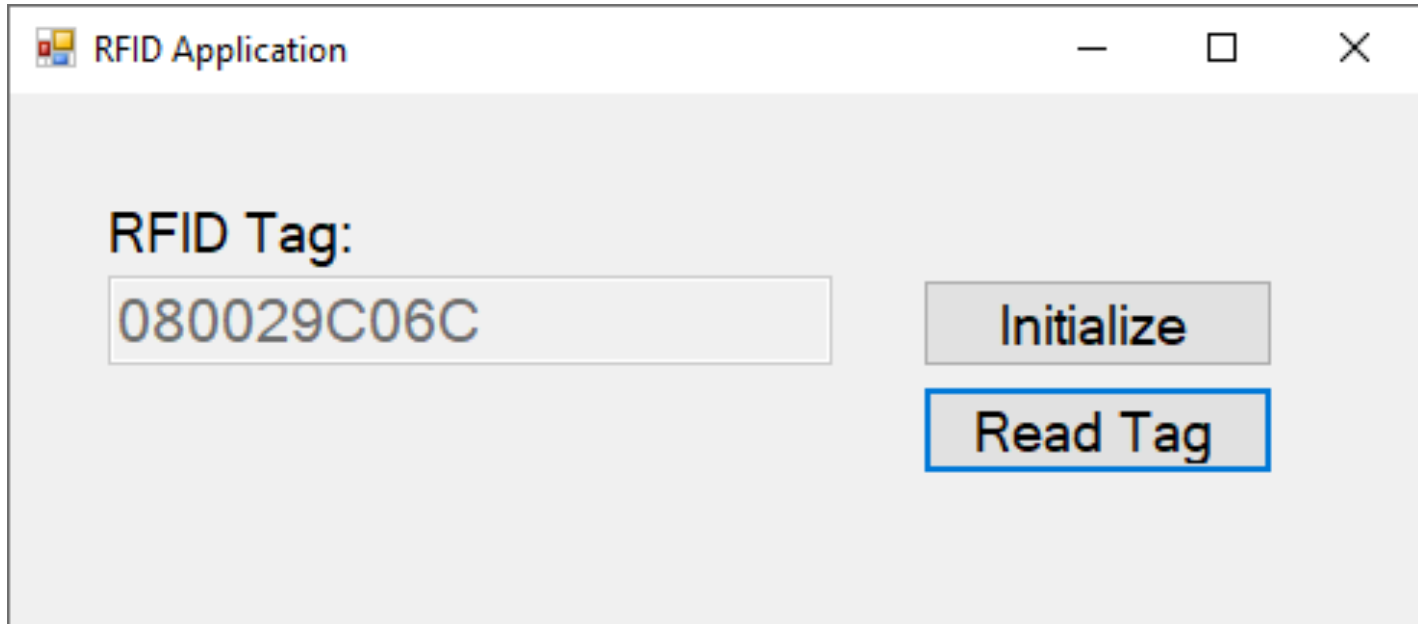
```
int numberBytesToRead = 12;  
byte[] data = new byte[numberBytesToRead];  
port.ReadTimeout = 1000;  
port.Read(data, 0, numberBytesToRead);
```

```
string rfidTag;  
rfidTag = System.Text.Encoding.UTF8.GetString(data, 0, data.Length);
```

```
rfidTag = rfidTag.Replace("\n", "");  
rfidTag = rfidTag.Replace("\r", "");
```

```
port.Close();
```


Visual Studio/C# Example



C# Example

```
using System;
using System.IO.Ports;
using System.Windows.Forms;

namespace ReadRfidApp
{
    public partial class Form1 : Form
    {
        string rfidTag;
        SerialPort port = new System.IO.Ports.SerialPort("COM4", 2400, System.IO.Ports.Parity.None, 8, System.IO.Ports.StopBits.One);

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {}

        private void btnInitialize_Click(object sender, EventArgs e)
        {
            port.Open();
            port.DtrEnable = true;

            txtTagData.Text = "";
        }

        private void btnReadTag_Click(object sender, EventArgs e)
        {
            int numberBytesToRead = 12;
            byte[] data = new byte[numberBytesToRead];
            port.ReadTimeout = 1000;
            port.Read(data, 0, numberBytesToRead);

            rfidTag = System.Text.Encoding.UTF8.GetString(data, 0, data.Length);

            rfidTag = rfidTag.Replace("\n", "");
            rfidTag = rfidTag.Replace("\r", "");

            txtTagData.Text = rfidTag;

            port.Close();
        }
    }
}
```

Resources

- <https://en.wikipedia.org/wiki/Barcode>
- https://en.wikipedia.org/wiki/Radio-frequency_identification
- <https://www.atlasrfidstore.com/rfid-beginners-guide/>
- <https://no.rs-online.com/web/p/rf-modules/1262181/>
- <https://eccel.co.uk/product/oem-micode-usb/>

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

