DevCon 2025

Connect | Learn | Build

June 4 - 6, 2025

Amsterdam, Netherlands

ZEBRA

DevCon 2025

Connect | Learn | Build

Explore Advanced Windows Development for RFID Readers and POS Integration

Suresh Ramamoorthy
Senior Manager
Software Engineering

Advanced Location
Technologies

ZEBRA

DevCon 2025

Connect | Learn | Build

Agenda

ZEBRA

Agenda

RFID Windows Development

1 SDK/Tools

- Architecture
- Windows
  - RFID .NET SDK
  - 123RFID Desktop
- JPOS
  - RFID Scanner Driver
  - Demo App

2 New Products

- FXP20 POS Reader
- RFID Accessory for ET6xW

3 Technical Resources

4 Best Practices

DevCon 2025

Connect | Learn | Build

SDK/Tools

ZEBRA

RFID System

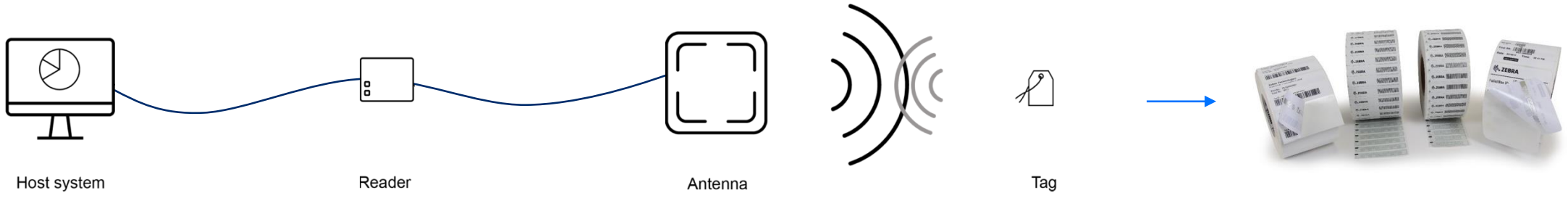
Physical View

DevCon 2025
Connect | Learn | Build

Identify and track objects wirelessly.

Operates at a frequency of 860-930 MHz (Ultra High Frequency).

Read range up to 20 meters



Host system	Reader	Antenna	Tag	
Manages and Controls the reader for reading tags or programming tag. Computer, Server or Cloud based platform	Transmitting the radio frequency to activate the tag for receiving, decoding/encoding the tag	Transmitting the radio frequency to the tag and receiving data from the tag Polarization Linear Tag placement is known Circular Placement of tag is unpredictablec	Attached to the asset / item being tracked Contain Microchip and antenna Types: Passive: Rely energy from reader. Less expensive. Reading range up to 20m Active: Tags have an internal power source (battery) for longer distance. (up to 100m reading range)	RFID Standards (GS1) EPC UHF Gen2 Air interface Protocol Physical and logical requirements for an RFID System https://www.gs1.org/standards/rfid/uhf-air-interface-protocol

Images Source: Avery Dennison, https://rfid.averydennison.com/en/home/explore-rfid/rfid-technology-basics.html

RFID Tag

Logical View

DevCon 2025
Connect | Learn | Build

Memory Bank	USER (optional				
	TID				
	EPC				
	RESERVED				
Target	Sessions				Select Flag
Inventoried Flag	SO	S1	S2	S3	SL Flag
Default	A	A	A	A	Deasserted
Persistence	Indefinite	500ms to 5s	Indefinite	Indefinite	Indefinite
Not Energized	None	500ms to 5s	2s	2s	2s

- **Reserved**:
  - Kill (1st and 2nd word)
  - access passwords (3rd and 4th word)
- **EPC**: Identifies the asset/item beginning from 4th byte or 2nd word
- **TID:** Tag manufacturer specific information (Tag Model, serial number). Perma Locked during tag manufacturing.
- **USER**: Allows user-specific data storage.
- **Tag States**
  - **Sessions**: Tag support four sessions. Tag participate in one and only session during inventory round. Inventoried flag can be A or B.
  - **Select Flag**: SL flag can be asserted or deasserted

Manage Tag Population

C1G2 Standard

DevCon 2025
Connect | Learn | Build

Select
(Pre filter)

Selects the tag population
- Target indicates whether Tag's SL or inventoried flag
- Action indicates whether matching tags
  - modify SL (Assert SL or Deassert SL)or inventoried flag (A or B)
- Memory Bank (EPC, TID or User Memory)
- Start Location, Length, Mask (bit string that tag compares)

Inventory
(Singulation)

Determines which tags to be participated
- Sel - which tags to respond based on SL flag
  0: ALL, 1: ALL, 2: ~SL, 3: SL
- Session - choose the session for the inventory round.
  0: S0, 1: S1, 2: S2, 3: S3
- Target (inventoried flag)
  0: A, 1: B

Q - Tag population in the range of antenna.

Access

Access commands
- Read – Reads memory bank
- Write – Writes memory bank
  0-Reserved, 1-EPC, 2-TID, 3-User
- Kill – Permanently disable the tag (Kill password)
- Lock – Access/Kill Password (Prevent read/write),
  Individual memory banks (prevent write),
  Permalock lock status password or memory bank

Action	Matching	Non-Matching
0	assert SL or inventoried -> A	deassert SL or inventoried -> B
1	Assert SL or inventoried -> A	do nothing
2	do nothing	deassert SL or inventoried -> B
3	negate SL or (A->B, B->A)	do nothing
4	assert SL or inventoried -> B	assert SL or inventoried -> A
5	assert SL or inventoried -> B	do nothing
6	do nothing	assert SL or inventoried -> A
7	do nothing	negate SL or A -> B, B->A)

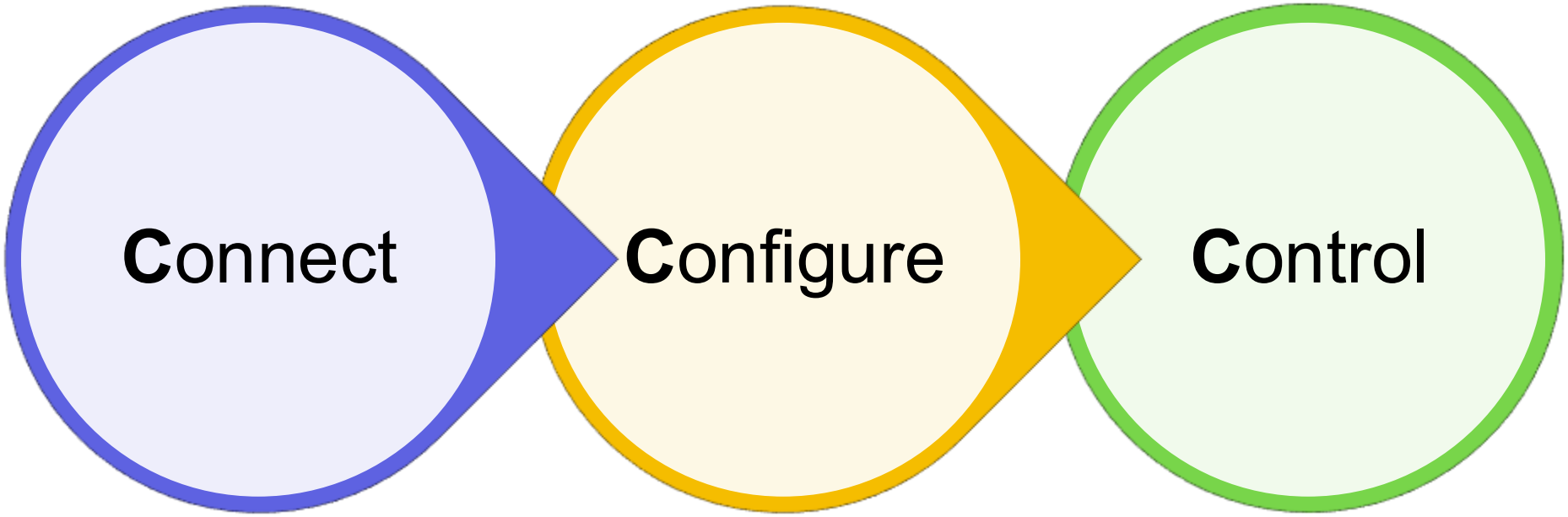
Select Tags Matching with EPC "3074"
Target: S0, Action: 4
Mem Bank: EPC
Start Location: 32
Length: 16
Mask: "3074"

Inventory (Singulation)
Sel: 0
Session S0
Target: B
Q: 8

3C Framework

Advance RFID Integration Quickly

DevCon 2025
Connect | Learn | Build



- Discover available readers
- Establish Connection

- Setup Region (Persistent)
- Customize the reader to meet specific needs (Transient)
  - Setup Event Streaming
  - Settings for optimal functionality
  - User preferences

- To achieve optimal performance
  - Control
    - Start/Stop operation
  - Manage
    - Software Update
    - Reboot
    - Factory Reset
  - Monitor Heath Status

RFID API

Structure

DevCon 2025
Connect | Learn | Build

Generic Reader

- Reading Tags (Inventory)
- Programming Tags (Tag Access)
    - Read, Write, Lock, Kill

Reader Management

- Managing Reader
- Region Setting
- Software Update
- Factory Reset
- Reboot
- Configuration

DevCon 2025

Connect | Learn | Build

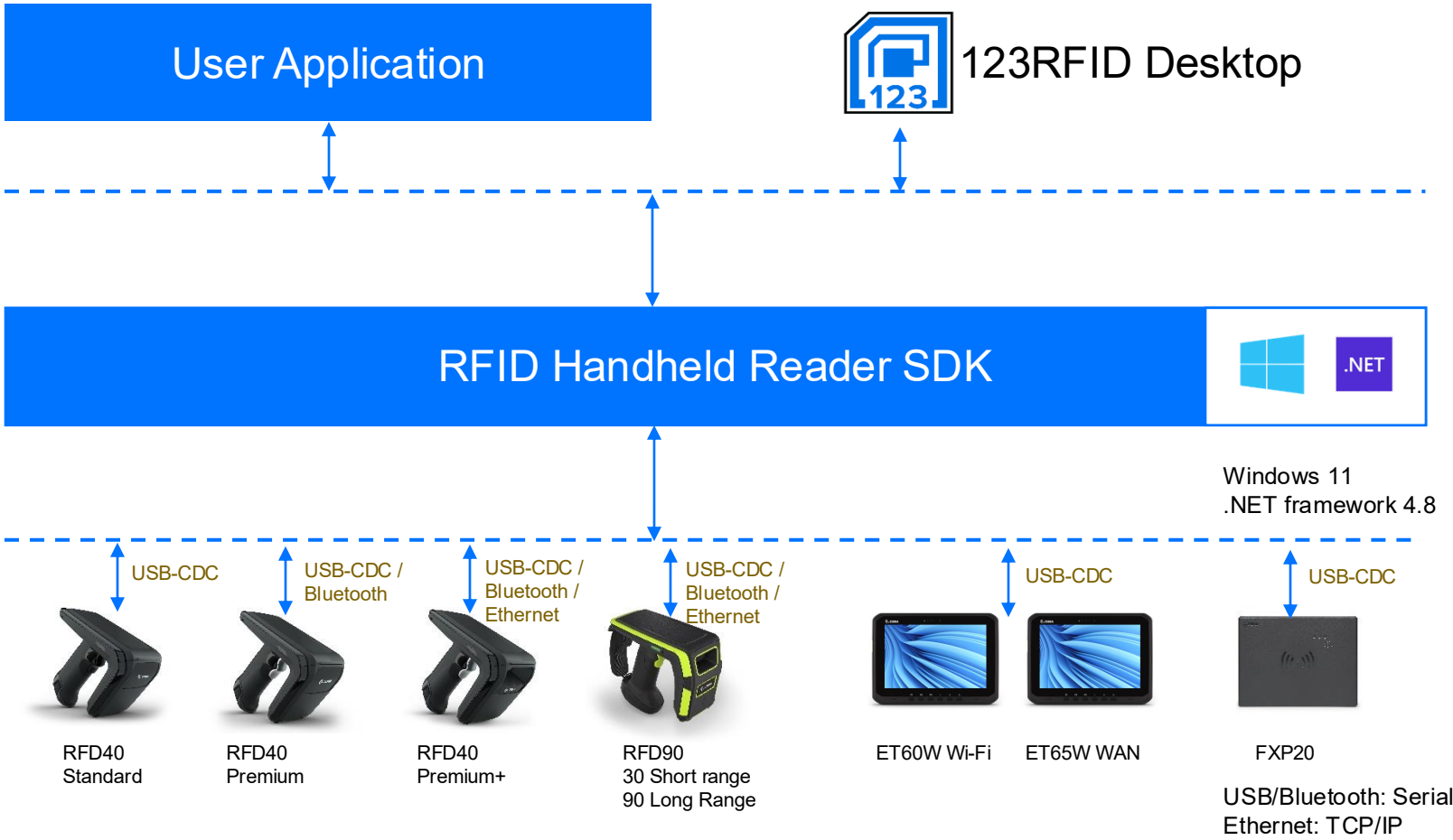
Windows SDK/Tool

ZEBRA

RFID Windows SDK

Architecture

DevCon 2025
Connect | Learn | Build



RFID Handheld Reader SDK

Windows

DevCon 2025
Connect | Learn | Build

- RFID Handheld Reader SDK for Windows
  - Same SDK that supports handheld RFID readers (RFD40/90, RFD8500, ET6xW, FXP20)
- Tools
  - 123RFID Desktop - GUI based Demo Tool
- Prerequisites
  - Host Platform: Microsoft Windows 11 64-bit and above
  - Microsoft .NET Framework: 4.8 for Windows
  - Visual Studio
- Release Package
  - Demo App: Windows.RFID.DemoApp_3.0.33.zip
  - Demo Source: Windows.Desktop.DemoApp.Src_v3.0.33.zip
  - RFID SDK: Windows.RFID.SDK_3.0.33.zip. Includes Libraries and config files are to be added in the project.

RFID .NET SDK

Generic Reader

DevCon 2025
Connect | Learn | Build

Connect	Configure	Control

Discover & Connect
- Create Reader Instance
  - Transport: USB
  - ReaderType.FXP
- **Connect**()
	Settings
	- Antenna (Power, Pre-filter, Singulation, RF Mode)
	  Configurations.Antennas[index].**Configuration** (getter/setter)
	  TransmitPowerIndex
	  RFMode
	  SingulationControl
	Setup Event Streaming using .NET event handler (+=)
	- Read Notification
	- Status Notification
	Profiles (ET6xW)
	- Configurations.**GetRFIDProfiles()**
	- Configurations.**SetRFIDProfile**(Profile)
		Reading Tags
		- Read tags (Start/Stop Conditions)
		  - Inventory.**Perform**()
		- Continuous Read
		  - Inventory.**Perform**() , Inventory.**Stop**()
		Tag Queue
		- Inventory.**GetReadTags**(Number of tags)
		Programming Tag (Set Tag ID, data, password as properties)
		  Write a Tag
		  - AccessOperations.TagWrite.**Write**()
		  Lock a Tag
		  - AccessOperation.TagLock.**Lock**()
		  Kill a Tag
		  - AccessOperation.TagKill.**Kill**()
		  Events
		  - TagDataReceived, InventoryStarted, InventoryStopped
		  Disconnect
		  - **Disconnect**()

RFID .NET SDK

Reader Management

DevCon 2025
Connect | Learn | Build

Connect	Configure	Control

Discover & Connect
- Create Reader Instance
  - Transport: USB
  - ReaderType.FXP
- **Connect**()
	Region (Persistent)
	- Configurations.**RegulatoryConfig** (getter/setter)
	  - **Region**
	  - **Hopping**
	  - **Channels**
	Setup Event Streaming
	• **N/A**
		Monitor
		- **N/A**
		
		Software Update
		- **SoftwareUpdate.Update** (firmwareFileName)
		- Poll Update progress status
		  - SoftwareUpdate.UpdateStatus.**Percentage**
		  - SoftwareUpdate.UpdateStatus.**UpdateInfo**
		Factory Reset
		- **ResetFactoryDefaults**()
		Disconnect
		- **Disconnect**()

Read Tags Demo

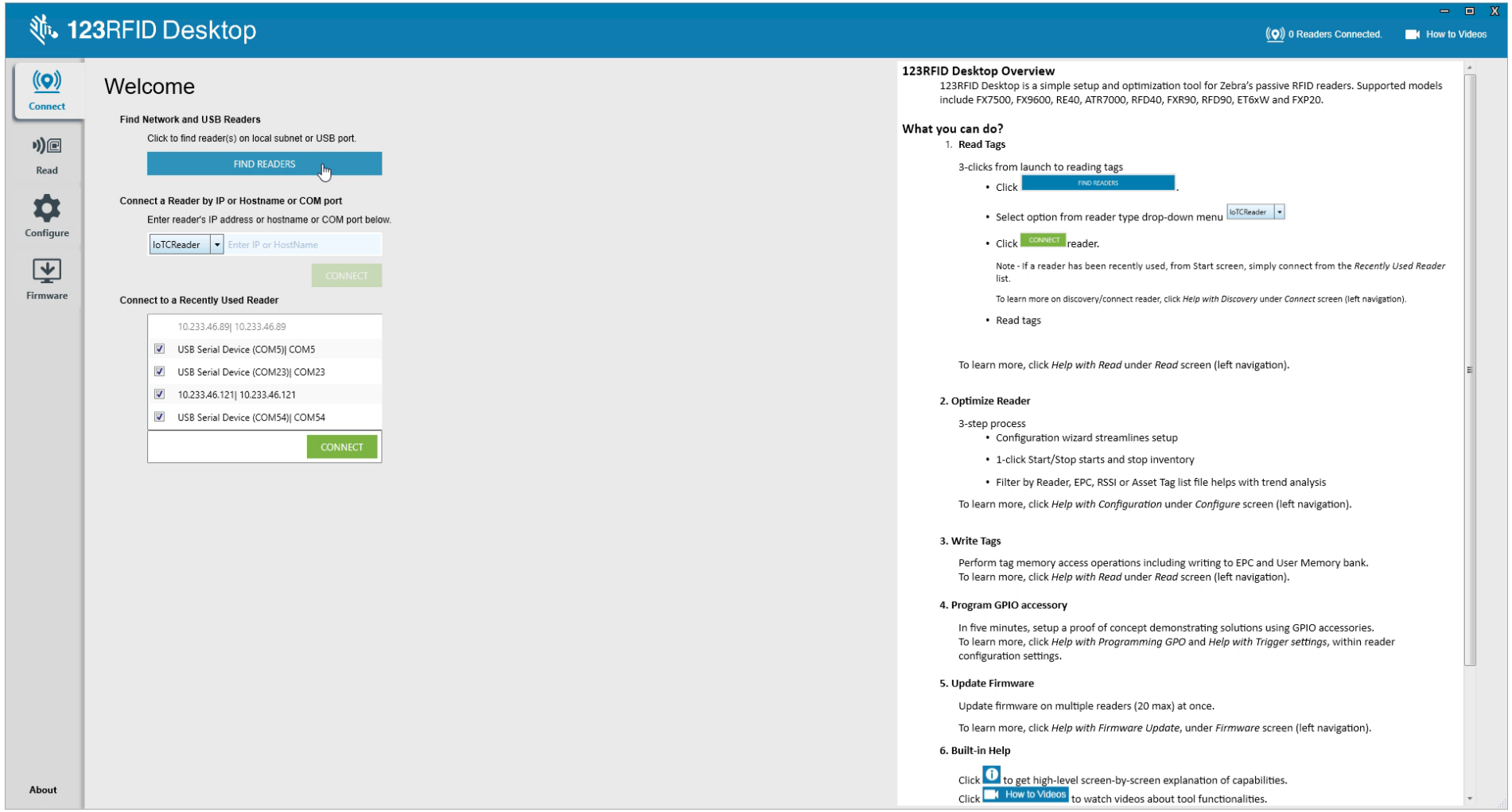

```
using Symbol.RFID.SDK.Domain.Reader;
using Symbol.RFID.SDK;
using System.Collections.Generic;
using System;
using System.Linq;
using System.Threading;
//CONNECT: Discover and connect to the reader
Console.WriteLine("Finding readers....");

//Initialize reader management and reader info list.
var readerInfoList = new List<IRfidReaderInfo>();

var readerManager = RfidSdk.ReaderManagementServicesFactory.Create(ReaderCommunicationMode.USB);
//get available readers list
readerInfoList = readerManager.GetReaders(ReaderSearchOptions.AllReaders);

foreach (IRfidReaderInfo readerInfo in readerInfoList.OrderBy(o => o.FriendlyName))
{
    //create reader instance based on reader information.
    var reader = RfidSdk.RfidReaderFactory.Create(readerInfo);
    Console.WriteLine("Found reader:-"+ reader.FriendlyName);
    readerList.Add(reader);
}
if(readerList!=null && readerList.Count>0)
{
    var fxp20Reader=readerList.FirstOrDefault(ee=>ee.ReaderType==ReaderType.FXP);
    if(fxp20Reader!=null)
    {
           Console.WriteLine("Connecting FXP20 reader-" + fxp20Reader.FriendlyName);
           fxp20Reader.Connect();
           Console.WriteLine("Connected-"+ fxp20Reader.FriendlyName+ "\n");
    }
}
```

```
// CONFIG: Setup Events
fxp20Reader.Inventory.AttachTagDataWithTagDataReceivedEvent = false;

//register for events
fxp20Reader.Inventory.InventoryStarted += Inventory_InventoryStarted;
fxp20Reader.Inventory.InventoryStopped += Inventory_InventoryStopped;

// CONTROL: Read tags for 2s
Console.WriteLine("Performing Inventory for 2 sec...");
//perform Inventory for 2 sec
fxp20Reader.Inventory.Perform();
Thread.Sleep(2000);
//stop Inventory
fxp20Reader.Inventory.Stop();

// read tags
var tagDataArr = fxp20Reader.Inventory.GetReadTags(1000);
foreach (ITagData dataReceived in tagDataArr)
    Console.WriteLine("Tag ID:"+ dataReceived.EPCId);

// Disconnect reader
fxp20Reader.Disconnect();

// Event Notifications
private static void Inventory_InventoryStarted(object sender, EventArgs e)
    Console.WriteLine("Event: Inventory started.\n");

private static void Inventory_InventoryStopped(object sender, EventArgs e)
    Console.WriteLine("Event: Inventory stopped.\n");
```

123RFID Desktop

Demo

DevCon 2025
Connect | Learn | Build



DevCon 2025

Connect | Learn | Build

JPOS Driver for FXP20

ZEBRA

POS Standards

Overview

DevCon 2025
Connect | Learn | Build

- What is JPOS?
  - JavaPOS (JPOS) is a free, open-source software framework
  - Provides a standard interface for point of sale (POS) devices (including RFID readers).
  - JPOS is based on the International Organization for Standardization transaction card originated messages standard (ISO-8583)

  Please see http://www.jpos.org/ for more info

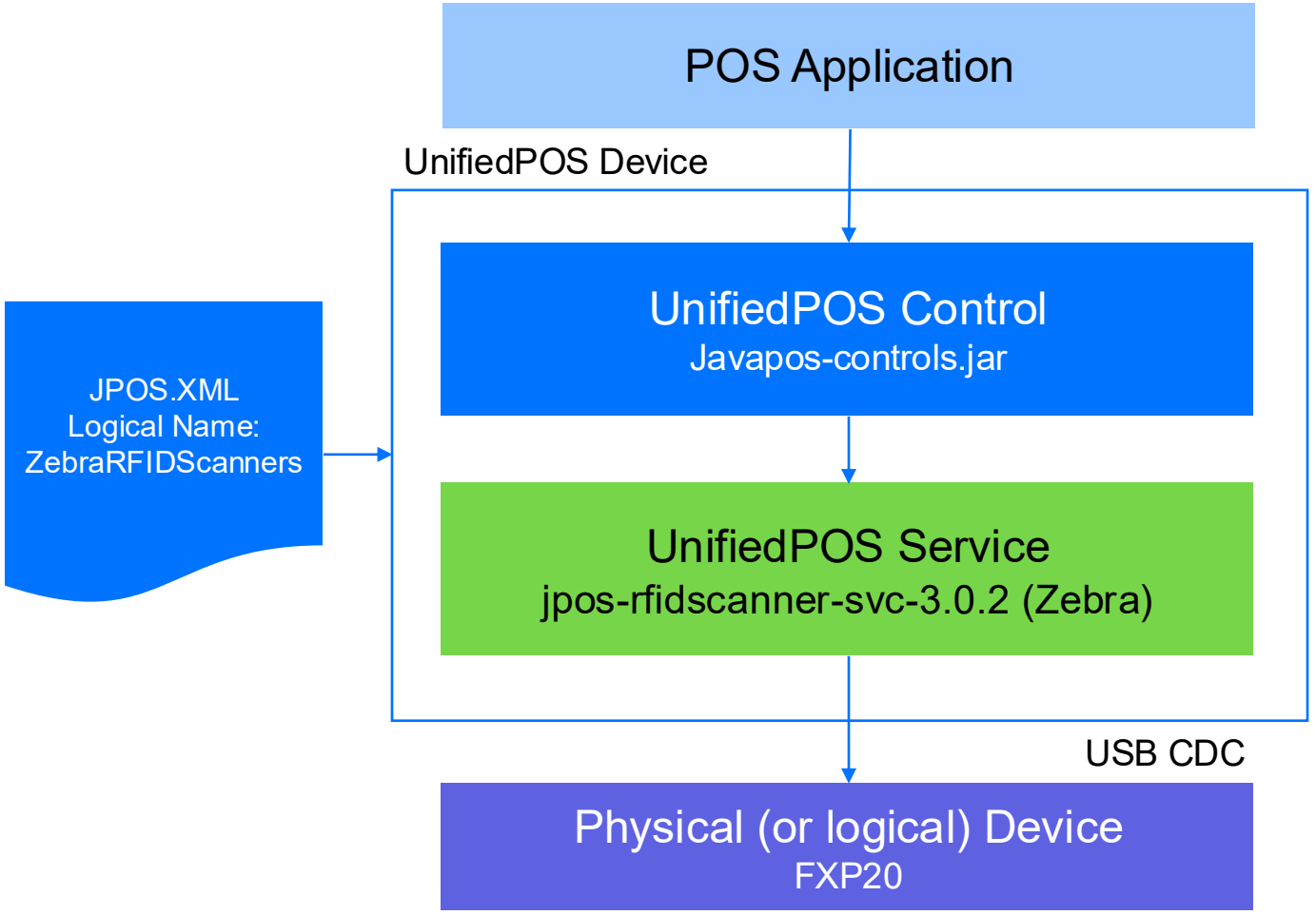
- What is UPOS?
  - UPOS (Unified Point of Service) is a standard specification used in retail and point-of-sale (POS) systems that defines a common programming interface for various types of POS devices.
  - UPOS is part of the broader JavaPOS and OPOS standards, which are often used in retail environments to interface with hardware like barcode scanners, receipt printers, cash drawers, and RFID readers.

  Please see https://www.omg.org/retail/unified-pos.htm for more info
- Provides consistent framework that is Platform-independent and vendor-neutral for POS devices

JPOS/UnifiedPOS

Architecture

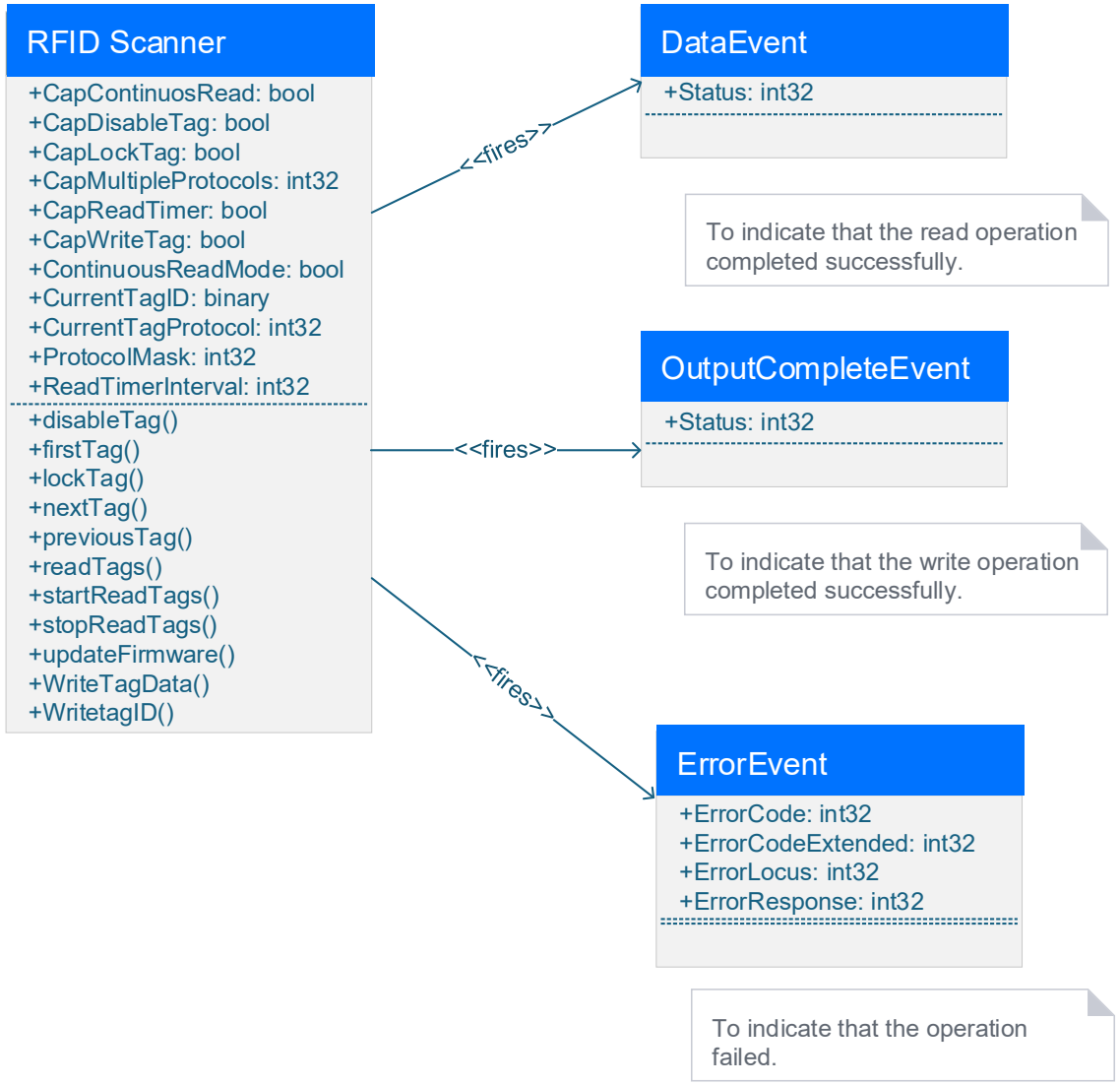
DevCon 2025
Connect | Learn | Build



JPOS RFID Scanner

Capabilities

- Reading Tags (Continuous / timer based)
  - Reads TagID and UserData
  - Reads Partial UserData
- Writing a Tag
  - Writes TagID
  - Writes UserData / Partial UserData
- Locking a Tag
- Disables (kills) RFID tag



JPOS RFID Driver

Windows

DevCon 2025
Connect | Learn | Build

- JPOS RFID Driver
  - Develop applications using the JPOS RFID Scanner Interface for FXP20 RFID POS Reader
- Demo Tools
  - JPOS Sample App
- Prerequisites
  - Host Platform: Microsoft Windows 11 64-bit and above
  - Java: Oracle JDK 1.8 or above
  - IDE: NetBeans IDE 8.2 or above [64-bit]

- Release Package
  - Extract the zip file on host system. The following are included in the java folder:
  - Lib: Service Class library and other JPOS dependent libraries
    - Zebra: jpos-rfidscanner-svc-3.0.2.jar
    - UPOS: jcl.jar, javapos-controls.jar, JavaCoreLogger.jar, jpos_trace.properties
  - jpos-rfidscanner-test.jar: Contains sample app to demonstrate
  - jpos.xml: Jpos configuration file contains the logical device name
  - javapostest.bat: Batch file to run the sample app (Windows)

RFID Reader

Generic Reader

DevCon 2025
Connect | Learn | Build

Connect	Configure	Control
Discover & Connect - **Open**() - **Claim**()	Settings - Read Timer Interval Setup Event Streaming - **addDataListener** - **addOutputCompleteListener** - **addErrorListener**	Reading Tags - **readTags** (filter, timeout, password) - Continuous Read - **StartReadTags**() , **StopReadTags**() Tag Queue - **firstTag**(), **nextTag**(), **PreviousTag**() Write a Tag - **writeTagID**(), **writeTagData**() Lock a Tag - **lockTag** (tagID, password) Kill a Tag - **disableTag** (tagID, password) Events - DataEvent, ErrorEvent, OutputCompleteEvent Disconnect - **Release()**, **Close()**

RFID Reader

Reader Management

DevCon 2025
Connect | Learn | Build

Connect	Configure	Control
Discover & Connect - **Open**() - **Claim**()	Setup Event Streaming • **addStatusUpdateListener** for firmware update JPOS.XML - Default Transmit Power	Monitor - **checkHealth** Software Update - **updateFirmware** (firmwareFileName) Disconnect - **Release**(), **Close**()

Read Tags Demo

```java
import jpos.JposException;
import jpos.RFIDScanner;
import jpos.events.*;

// Read Tags Demo Sample Code
public class JavaPosReadTagsDemo {

    static String logicalName = "ZebraRFIDScanners";
    static RFIDScanner reader = null;
    static private DataListener dataListener;

    public static void main(String[] args) throws JposException {
        // Create Reader Instance
        reader = new RFIDScanner();

        // 1. CONNECT
        // Establish connection to the reader (Discover and connect automatically)
        reader.open(logicalName);
        reader.claim(1000);

        // 2. CONFIGURE
        // Enable Device
        reader.setDeviceEnabled(true);

        // Enable Data Event
        reader.addDataListener(dataEventListener);

        // 3. CONTROL
        // Reading tags for 5 Seconds
        reader.readTags(RFID_RT_ID, new byte[0], new byte[0], 0, 0, 5000, new byte[0]);

        // Wait for a seconds
        try{
            Thread.sleep(1000);
        } catch ( InterruptedException jposException) {
            jposException.printStackTrace();
        }
        try{
            // Fetch first tag if read
            reader.firstTag();

            for(int index = 0; index < reader.getTagCount()-1;index++){
                String tagID = new String(reader.getCurrentTagID());
                System.out.println("Tag ID "+tagID);
                // move iterator to next if tag enqueued
                reader.nextTag();

            }
        } catch (JposException e) {
            e.printStackTrace();
        }
        // Dispose Reader
        reader.release();
        reader.close();

    }

    static DataListener dataEventListener = new DataListener() {
        @Override
        public void dataOccurred(DataEvent dataEvent) {
            System.out.println("dataOccurred STATUS : " + dataEvent.getStatus());
        }
    };
}
```

DevCon 2025
Connect | Learn | Build

JPOS Sample App

Demo: Run JavaPOSTest.bat

DevCon 2025
Connect | Learn | Build

DevCon 2025

Connect | Learn | Build

FXP20 POS Reader

ZEBRA

FXP20 Fixed RFID Reader

Specification

DevCon 2025
Connect | Learn | Build

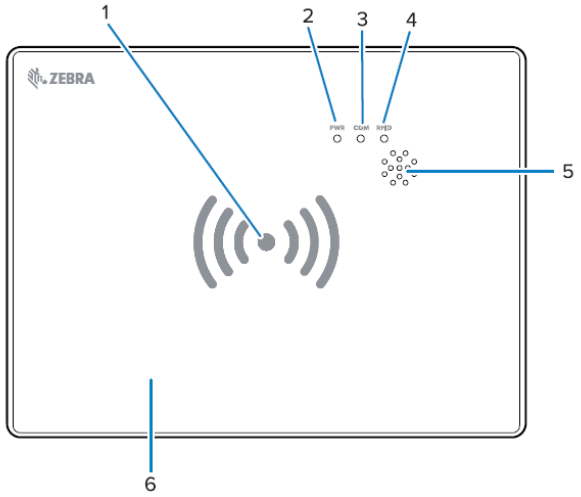
Power	12VDC, 2A with lockable connector
Internal worldwide Antenna	1 dBic – Circular Polarized
RFID Power	Scalable from -10 to 27 dBm
External Antenna	3x SMA connector
Visual Status Indicators	3 LED indicator multicolor lights (green, blue, orange)
Audible Status Indicators	Buzzer adjustable from 0 to 95dB (4 levels)
Connectivity	1 x USB Host Type-B Lockable
GPIO port (input)	1 – 12 volts/50 ma

FXP20 Fixed RFID Reader

External Interfaces

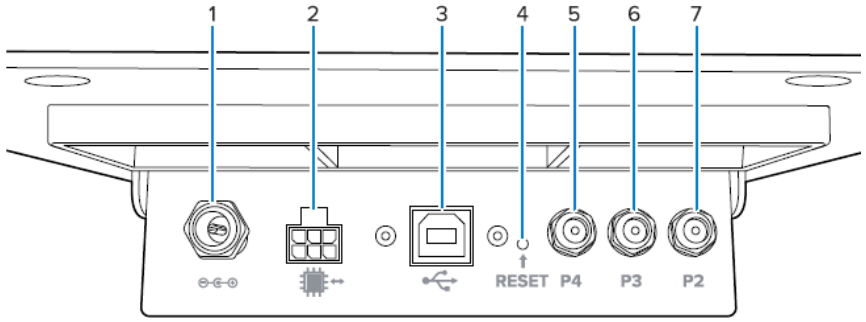
DevCon 2025
Connect | Learn | Build

- Front View

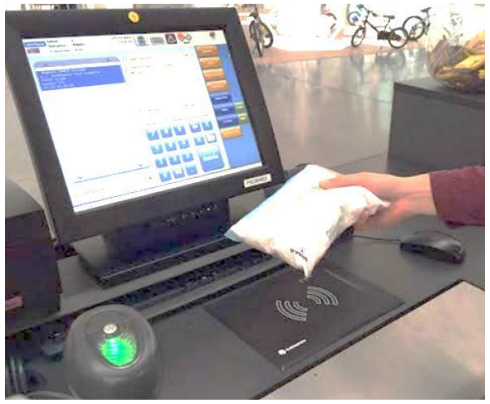


#	Description
1	RFID Reading Area
2	Power LED
3	COM LED
4	RFID LED
5	Internal buzzer
6	ABS UL94V0 case

- Side View



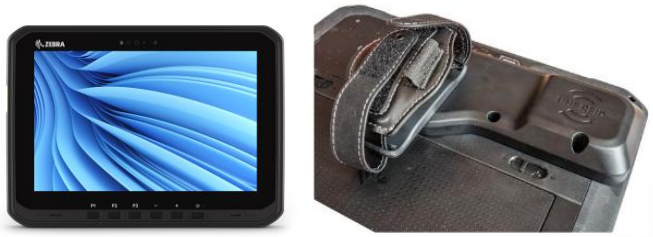
#	Description
1	DC Power connector
2	GPI connector
3	USB connector
4	Reset Button
5	Antenna Port 2
6	Antenna Port 3
7	Antenna Port 4



DevCon 2025

Connect | Learn | Build

ET6xW Tablet



ZEBRA

ET6xW Tablet

Specification

DevCon 2025
Connect | Learn | Build

- SKU: XBK-ET6X-RFID-X-01 (X➔N – North America, E-EMEA, A-AU/NZ)
- Easy to install. Attaches to the back side of the ET6x
- Hand Strap
- EPC Class1 Gen2 UHF RFID protocol
- North America: 902-928 MHz
- EU: 865-867 MHz
- Read range: > 1 meter (3 feet)
- Multiple tags reading: min 10 tags per second
- Circular polarized antenna• Max. USB 5V/750mA
- RF max power: 24 dBM
- IP66 sealed
- Operating Temperature: -20 to +60 C
- Drop rating: 4 feet on concrete

Smart Inventory

Profiles

DevCon 2025
Connect | Learn | Build

Profile	Description	Transmit power	Duty Cycle	Session	Tag Population	Link Profile
Optimal Battery	For Best battery life	20dBm	15% RF ON: 100ms Period: 667ms	S0	3	0
Balanced Performance (Default)	Maintains Balance between performance and battery life	20dBm	25% RF ON: 150ms Period: 600ms	S0	3	0
Max range	Reads as many tags as fast as possible in longer range	24dBm	30% RF ON: 150ms Period: 500ms	S0	10	2
Fastest Read	Reads as many tags as fast as possible in shorter range	20dBm	35% RF ON: 175ms Period: 500ms	S0	10	2
Cycle Count	Reads unique tags	24dBm	30% RF ON: 150ms Period: 500ms	S2	50	0

Link Profile table
0: M=4/240, PIE=2, Tari=25us
1: M=2/320, PIE=2, Tari=18.8us
2: M=4/320, PIE=2, Tari=18.8us

DevCon 2025

Connect | Learn | Build

Technical Resources

ZEBRA

Resources

SDK/Tools/Documentation

DevCon 2025
Connect | Learn | Build

- SDK/Tool
  - RFID Reader SDK for Windows (.NET) – includes Demo App (Exe, Source), SDK
    - https://www.zebra.com/us/en/support-downloads/software/rfid-software/zebra-rfid-sdk-for-windows.html
  - RFID Reader Tech Docs
    - https://techdocs.zebra.com/dcs/rfid/
  - 123RFID Desktop v3.0.0.33 or above
    - https://www.zebra.com/us/en/support-downloads/software/rfid-software/123rfid.html
  - JPOS RFID Scanner Driver (Java)
- FXP20 POS RFID Reader Product Info
  - https://www.zebra.com/us/en/support-downloads/rfid/rfid-readers/fxp20.html
  - Quick Reference Guide
  - Integration Guide
- Contact Zebra for the best and quickest way to get support for any questions or issues:
  - https://www.zebra.com/us/en/about-zebra/contact-zebra/contact-tech-support.html

RFID

Best Practices

DevCon 2025
Connect | Learn | Build

- JPOS RFID
  - JPOS does not support Region Settings, write a Tag Access or Kill Password.
  - 123RFID Desktop or .NET SDK can be used to set the region
- Read Range
  - Adjust transmit power to avoid reading the tags in nearby RFID reader
- Events Handling
  - No Blocking functions in events handler / callback functions
  - Enqueue the tags and release immediately
- Reliable Encoding tag
  - Tags must be closer the antenna and transmit power can be maximum
- Optimal Performance
  - 123RFID Desktop tool can be used to verify and identify the right settings

SDK/Tool

Upcoming features (FXP20)

DevCon 2025
Connect | Learn | Build

- Windows SDK/JPOS
  - Beeper
  - External Antenna
  - GPI
- JPOS
  - Region setting
    - For now, Region must be configured using 123RFID Desktop or Windows RFID SDK.

Questions?

DevCon 2025

Connect | Learn | Build

Thank You