XFS4IoT SP-Dev Workgroup

4th April 2023

XFS4IoT SP-Dev Workgroup agenda

- Recap from previous meeting
- What's new in the latest release
- Sigma presentation
- Next meeting

Recap from previous meeting

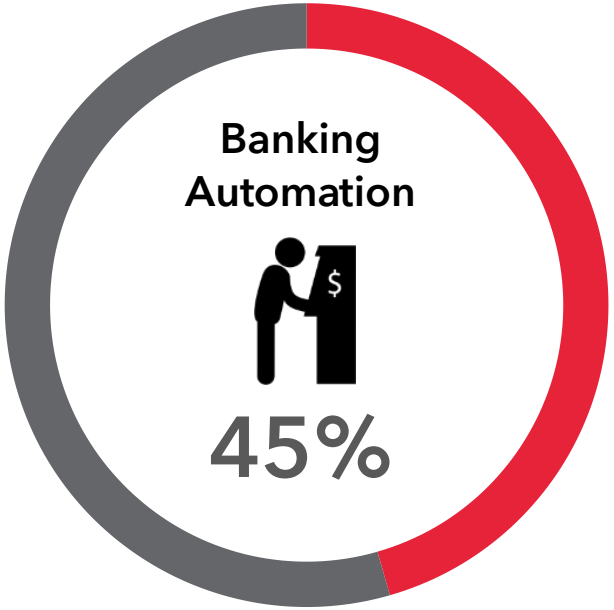
Recap from previous meeting

- Discussed XFS4 committee's progress
  - New mixed media class, moved "check scanning" out of printer class
  - Stricter definition of 'required' versus 'optional' parameters
  - Future releases will include more End-to-End security among other items
- Detailed discussion about End-to-End Security
  - Reviewed what is supported in the current spec and later this year
  - Brief discussion on 'post 2023' plans

SIGMA

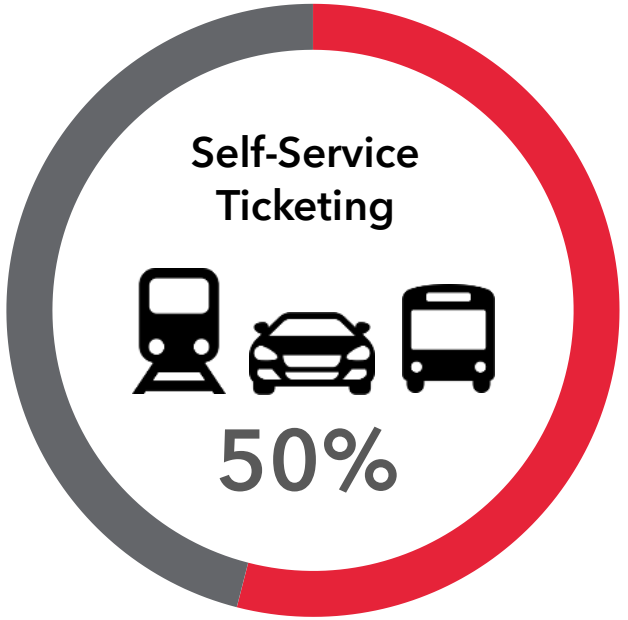
Our job, your solution

Business Area



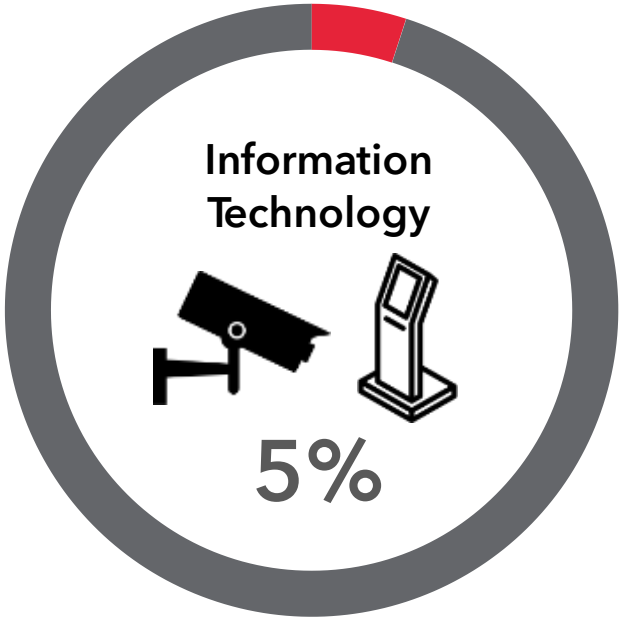
Bank Automation Machines

- Cash Dispensing
- Assisted Self Service
- Cash Recycling
- Teller Cash Recycler
- Banking Kiosk



Ticketing, Tolling and Parking

- Ticket Vending machines
- Self-service payment system
- Input-output parking devices
- OBU Dispensers

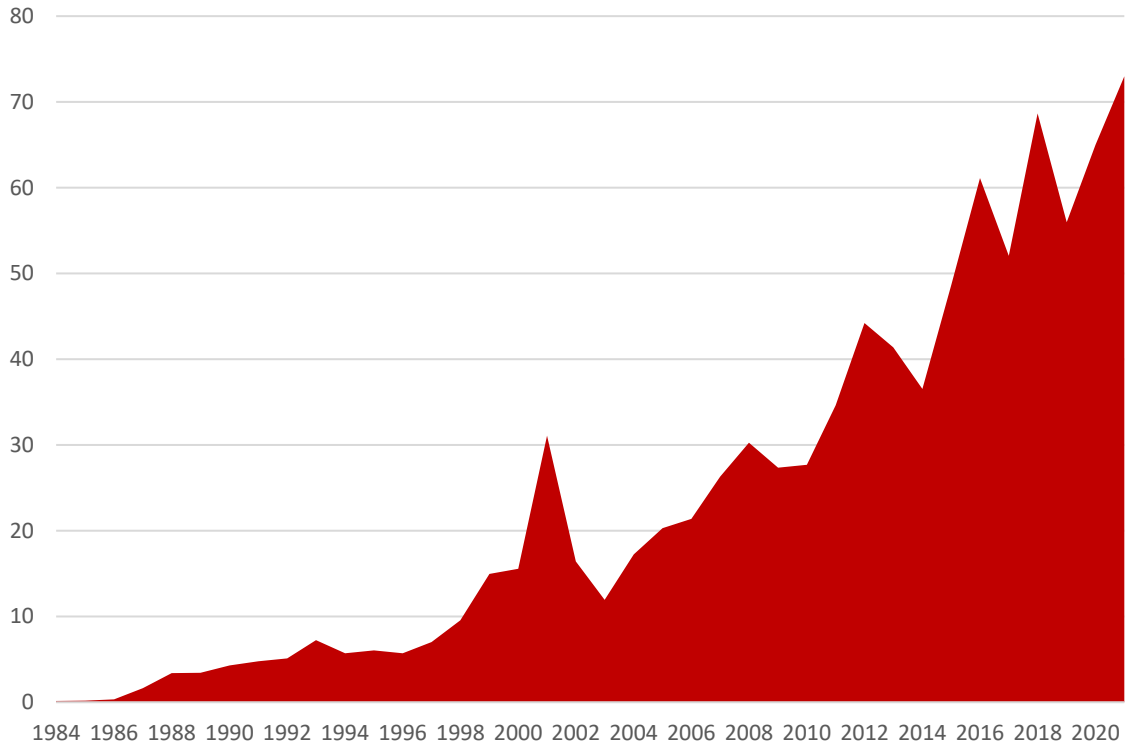


Security

- CCTV
- Access control systems
- Intelligent safe
- Fire detection
- Queue management system

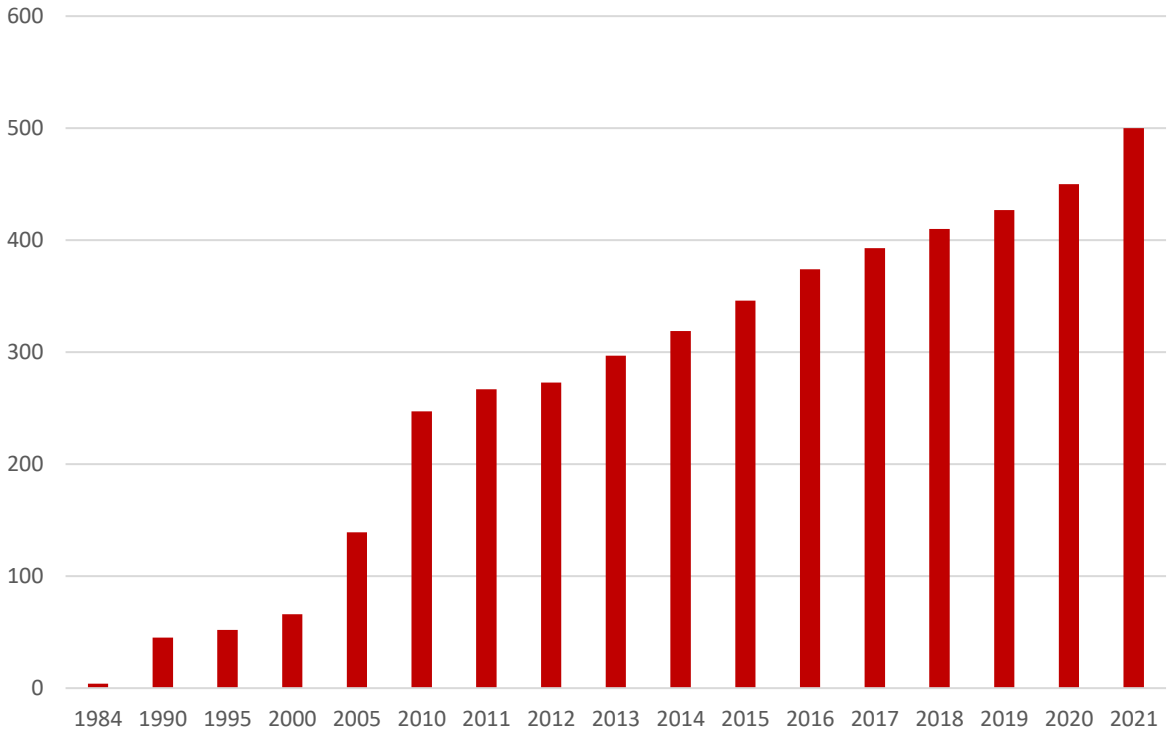
Corporate Growth

SALES €/Mnl



ANNUAL TURNOVER IN 2021: € 74.184.171

EMPLOYEES



EMPLOYEES 2021: 487

EMPLOYEES 2022: 517

Where We Are

Headquarters: Via dell'Industria 19, Monterubbiano (FM) - Italy

MONTERUBBIANO SITE

Building 1
Production Traffic

Building 2
Logistic

Building 3
R&D Laboratory

Building 4
Production Ticketing
Global Service Center

Building 5
Global Service Logistic

Building 6
Executive Offices
Administrative and Commercial Offices

OTHER SITE

COMUNANZA (AP)
Production Banking

MORESCO (FM)
Logistics Center

CONCORDIA S. Secchia (MO)
R&D Laboratory

DORZANO (BI)
Global Service Point

MILANO
Global Service Point

ROMA
Global Service Point

FOREIGN SUBSIDIARIES

SIGMA UK

Kemp House Suite 7000 152 - 160
City Road, LONDON (United Kingdom)

SIGMA Automatique

7 Avenue des Droits de l'Homme
45000, ORLEANS (France)

SIGMA Belgio

Rue Anatole France 115 - 121 - B 1030
BRUXELLES (Belgium)

Banking

TCR

Z1HR

CASH DISPENSING

ST 400EVO

CASH RECYCLING

SD 960EVO

ASSISTED SELF-SERVICE

SR 2000

Banking

MULTI-FUNCTION KIOSKS AND QMS

CRONO

RTS 2020

KIS

Fare Collection Solutions

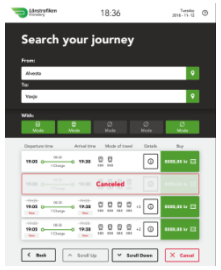
TICKETING AUTOMATION

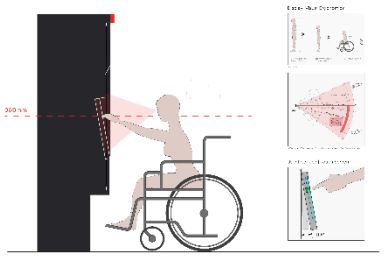
ETS70

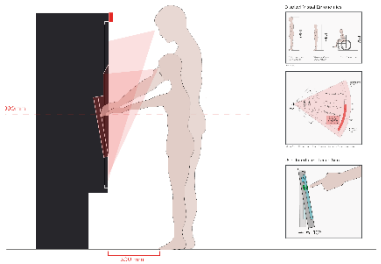
ETS90

KOS

KIS







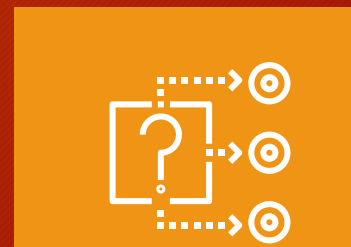
CONCEPT AND DESIGN

SIGMA

Demo XFS4IoT Printer Service

Created using KAL SP-Dev framework

Main Targets



- Implement an XFS4IoT Printer Service
- Run the Printer Service on Single Board Computer

Single Board Computers (SBC)

- **Single board design:** SBCs are self-contained computer systems that are built on a single circuit board. This makes them highly compact and portable.
- **Low power consumption:** SBCs are designed to be highly energy-efficient, consuming much less power than traditional desktop or laptop computers.
- **Low cost:** SBCs are typically much less expensive than traditional computers, making them a cost-effective solution for many applications.
- **Variety of form factors:** SBCs come in a variety of form factors, including the popular Raspberry Pi form factor, as well as other form factors such as Nano-ITX and Pico-ITX.
- **Customizable:** SBCs are highly customizable and can be configured to suit a wide range of applications and use cases.
- **Versatile:** SBCs can be used for a wide range of applications, including embedded systems, robotics, IoT devices, media centers, and more.
- **Accessible:** SBCs are accessible to a wide range of users, from hobbyists and makers to professional developers and engineers.
- **Expandable:** Many SBCs have expansion options, such as GPIO pins, add-on boards, and other interfaces, allowing users to expand their capabilities as needed.

These are just some of the main characteristics of single board computers.

Raspberry Pi 4
(https://www.raspberrypi.com/)

- **Processor:** Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) at 1.5GHz
- **RAM:** 2GB, 4GB, or 8GB LPDDR4-3200 SDRAM (depending on the model)
- **Connectivity:** Gigabit Ethernet, dual-band 802.11ac wireless, Bluetooth 5.0, BLE
- **USB Ports:** 2 USB 3.0 ports and 2 USB 2.0 ports
- **Video Ports:** 2 micro-HDMI ports (up to 4Kp60 supported)
- **Audio Ports:** 3.5 mm audio port, stereo audio and digital surround 7.1 support
- **Mass storage:** microSD card slot
- **GPIO:** 40-pin GPIO header, compatible
- **Power:** USB-C power supply (5V, 3A), Power-over-Ethernet (PoE) support enabled via a PoE HAT (sold separately)
- **Dimensions:** 88 x 58 x 19.5 mm

These are just some of the main specifications of the Raspberry Pi 4. The board also has many other features and capabilities.

Enviroments

Development Environment

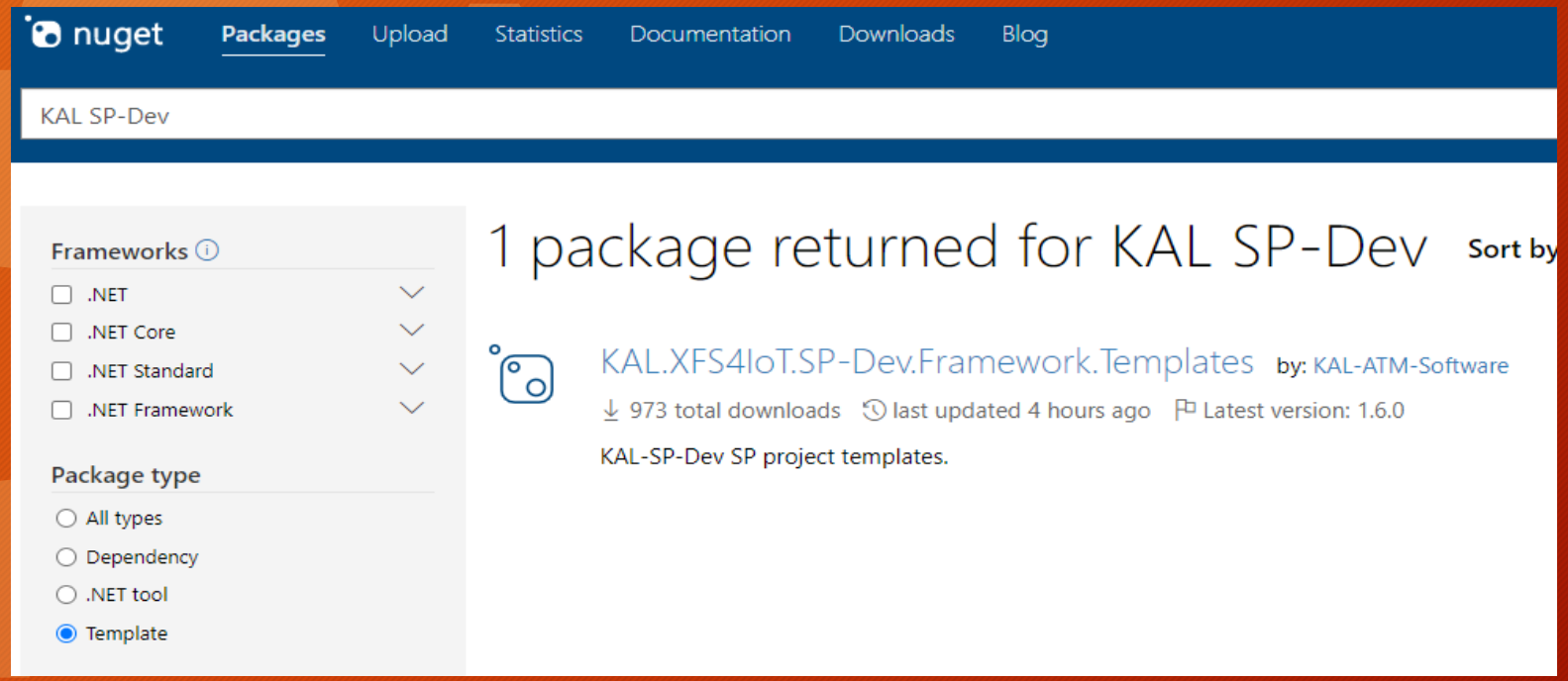
- Microsoft Windows 11
- Microsoft Visual Studio 2022
- KAL SP-Dev

Test Environment

- Raspberry Pi 4
- Raspberry Pi OS
  - Kernel version: 5.15
  - Debian version: 11

SP-Dev Framework

SP-Dev Framework 1.6.0 available on nuget.org



SP-Dev Framework

```
dotnet new install KAL.XFS4IoT.SP-Dev.Framework.Templates::1.6.0
```

```
dotnet new list
```

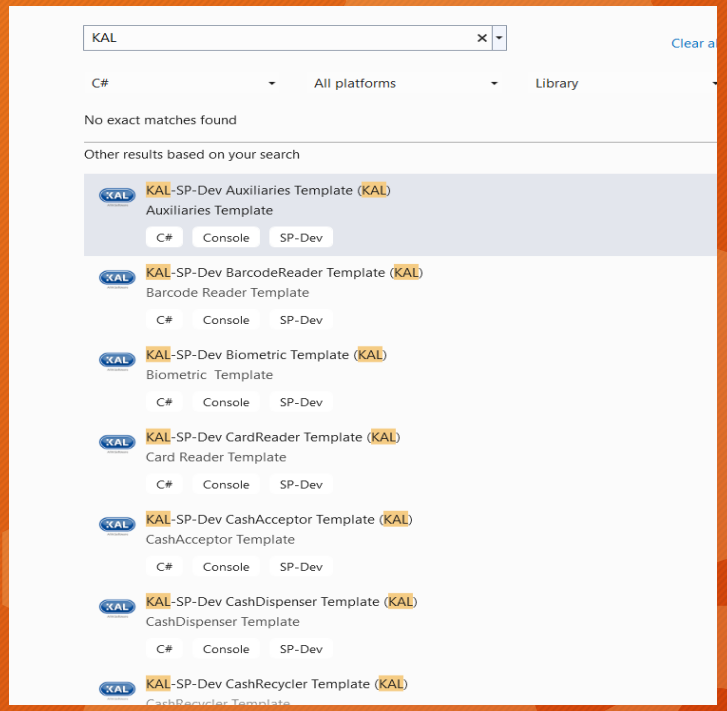
```
dotnet new list --tag SP-Dev
```

```
These templates matched your input: --tag='SP-Dev'

Template Name                          Short Name  Language  Tags
-------------------------------------  ----------  --------  --------------
KAL-SP-Dev Auxiliaries Template        spdev-Ax    [C#]      SP-Dev/Console
KAL-SP-Dev BarcodeReader Template      spdev-Bcr   [C#]      SP-Dev/Console
KAL-SP-Dev Biometric Template          spdev-Bio   [C#]      SP-Dev/Console
KAL-SP-Dev CardReader Template         spdev-Crd   [C#]      SP-Dev/Console
KAL-SP-Dev CashAcceptor Template       spdev-Caa   [C#]      SP-Dev/Console
KAL-SP-Dev CashDispenser Template      spdev-Cad   [C#]      SP-Dev/Console
KAL-SP-Dev CashRecycler Template       spdev-Car   [C#]      SP-Dev/Console
KAL-SP-Dev Lights Template             spdev-Lig   [C#]      SP-Dev/Console
KAL-SP-Dev PinPad Template             spdev-Pin   [C#]      SP-Dev/Console
KAL-SP-Dev Printer Template            spdev-Prn   [C#]      SP-Dev/Console
KAL-SP-Dev TextTerminal Template       spdev-Txt   [C#]      SP-Dev/Console
KAL-SP-Dev VendorApplication Template  spdev-Vda   [C#]      SP-Dev/Console
KAL-SP-Dev VendorMode Template         spdev-Vdm   [C#]      SP-Dev/Console
```

Create the SP Project

Choose the KAL-SP Dev Printer template



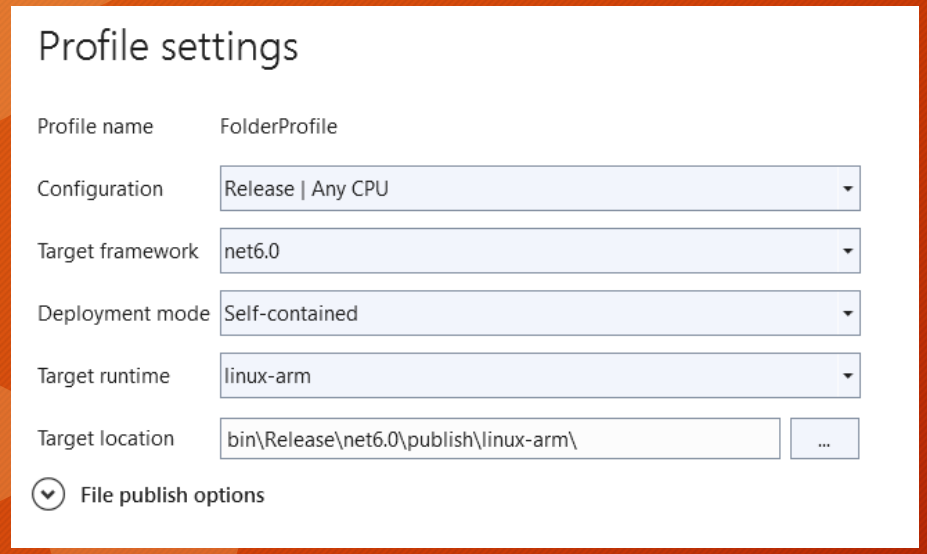

```
dotnet new spdev-Prn -n PrinterXfs4IoT
```

Methods and Properties Implemented

Features	Methods\Properties
Capabilities	PrinterCapabilities
Status	PrinterStatus
Reset	ResetDeviceAsync
Send Raw Data	RawPrintAsync
Print using forms	FormRules • DotsPerInchTopX / DotsPerInchBottomX • DotsPerInchTopY / DotsPerInchBottomY • DotsPerMMTopX / DotsPerMMBottomX • DotsPerMMTopY / DotsPerMMBottomY • DotsPerRowTop / DotsPerRowBottom • DotsPerColumnTop / DotsPerColumnBottom PageSize MediaSpecs ExecutePrintTasksAsync
Present Ticket	ControlMediaAsync
Send Events	RunAsync

Publish the project as portable for linux-arm

the self-contained package for Linux ARM contain all the libraries necessary to run the application on the specific platform, without the need to separately install .NET




```
dotnet publish -c Release -r linux-arm --self-contained true -p:PublishReadyToRun=true
-p:PublishSingleFile=true -p:TargetFramework=net6.0
```

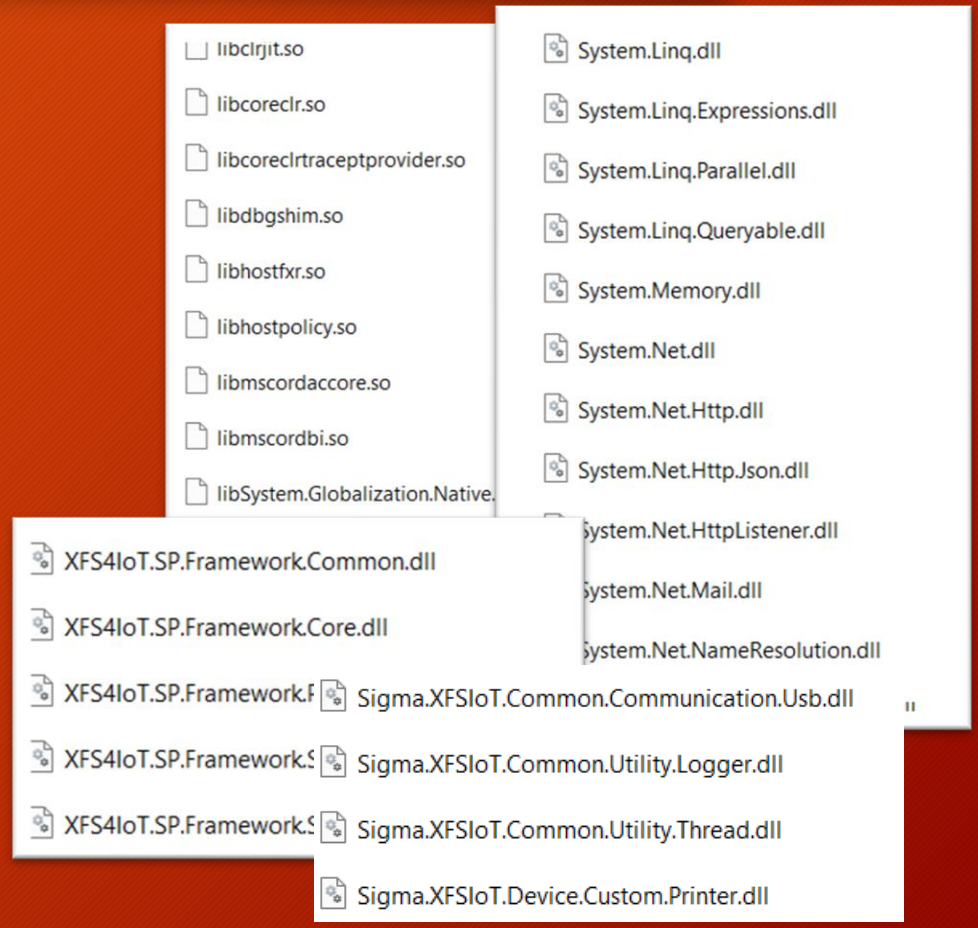
Publish the project as portable for linux-arm

The output folder contains:

NET Core runtime includes a set of libraries that implement the core functionalities of the framework

The SP-Dev Framework core assemblies

The custom assemblies and\or libraries implemented for driving the device



Running on Raspberry Pi

Copy the Publish folder in a Linux directory

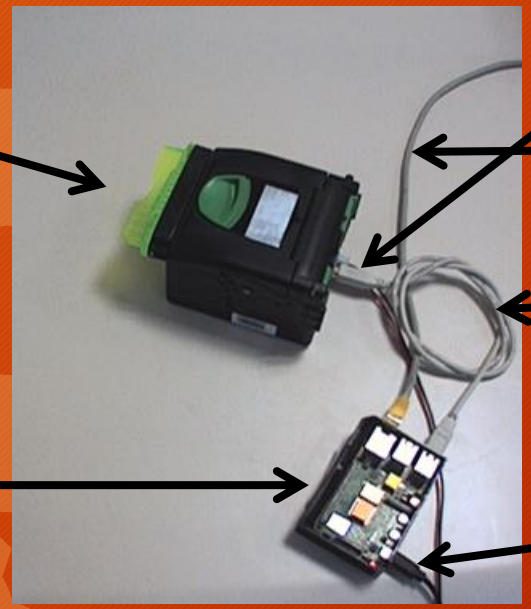
Do runnable the published file (chmod +x DemoPrinterXfs4IoT)

Configure the ip address:

- “DemoPrinterXfs4IoT.dll.config” file
- linux firewall

Configure the runnable file on linux boot

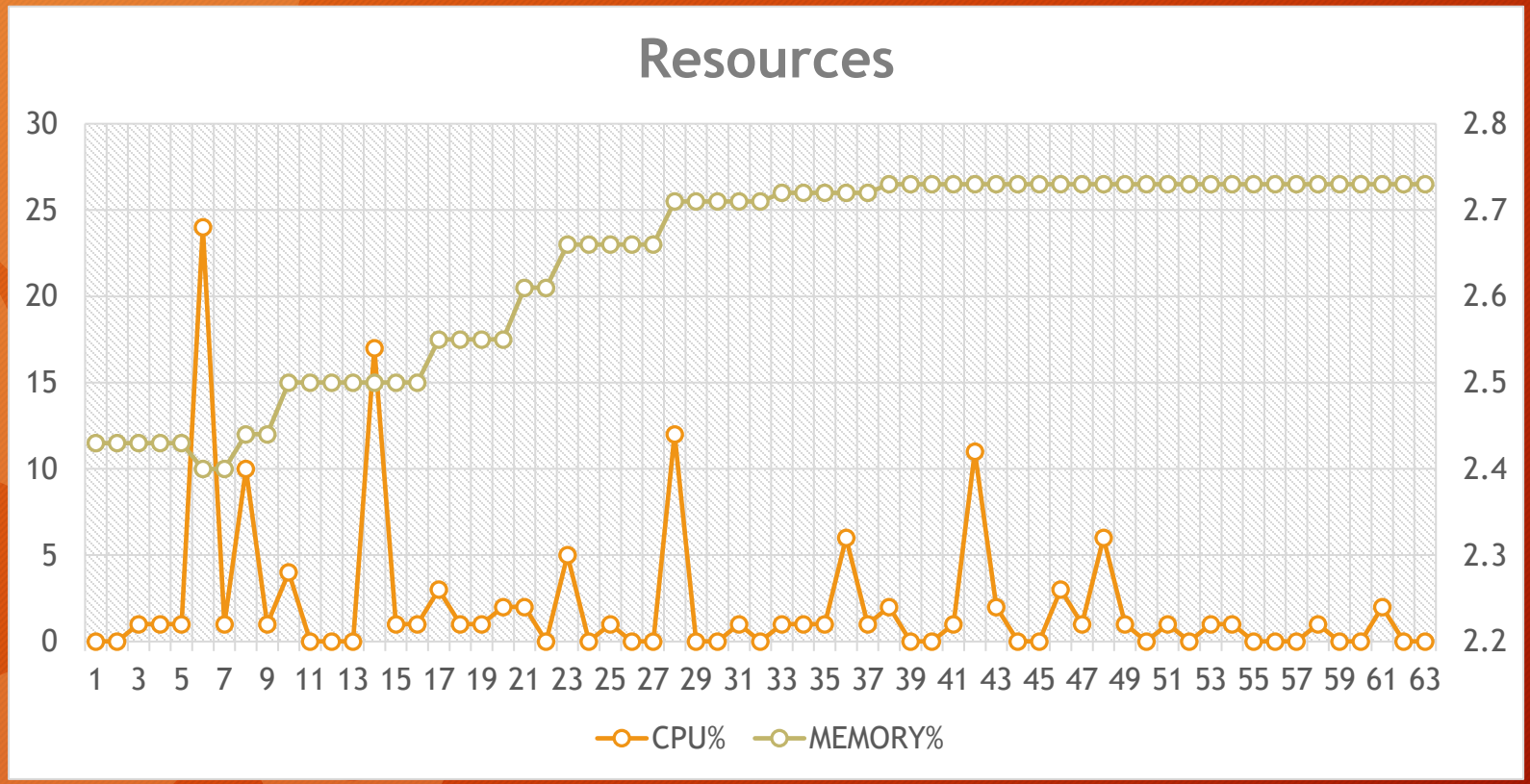
Demo Video


Printer Device
Raspberry Pi 4
Device power cable
Network cable
USB cable
Raspberry power cable

Demo Video

External Video

Resources


Resources
30
25
20
15
10
5
0
2.8
2.7
2.6
2.5
2.4
2.3
2.2
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63
CPU%
MEMORY%

Conclusion

Fast software and easy implementation of XFS4Iot service provider

Platform portability (The self-contained .NET C# solution can be easily ported to different platforms)

Easy deployment

Scalability

Low hardware resource usage

Cost-effective

What's new in the latest release v1.6.2

Multiple Pull Requests Merged

- #9 - Fix valid fonts check and SendAsync()
  — @RotoGiacomo
- #10 - Some bugs in PrintForm handle
  — @LinuxTvT
- #11 - Fix bug: Command not dequeue to handle
  — @LinuxTvT

- Multiple ways to contribute!
- Raise pull requests with changes
- Raise GitHub Issues with questions, bugs reports or suggestions

Framework Change Details

- Updated to ensure only a single message can be sent to the client at a time
- Reduced memory allocations when receiving messages from the client
- Updated Command Queue to always handle any queued commands

Framework Change Details

- Fix check for valid fonts in Print Form handler
- Save printer Form and Media information using persistent storage implementation
- Fix various issues with printer form field assignment
- Correct override for VendorMode CommandPostProcessing implementation

Next call

Zoom

- First Tuesday of each month at 1300 UK time

Next call: 2 May 2023 at 1300 UK

Calls are 30 mins long

We will continue to use Zoom

(Interpretation in Japanese, Chinese and Spanish is available using Zoom's interpretation feature)