

FRAMEWORK

INDIGO

EIN FRAMEWORK ZUR INTEGRATION VON MASCHINEN UND IT-SYSTEMEN SOWIE DEREN STEUERUNG MITTELS GRAFISCHER PROZESSMODELLIERUNG MIT DER BPMN-NOTATION.

STABILIS

POWERED BY INDIGO

In welche Richtung gehen wir und warum?

Die Suche nach einer Methode, die den Menschen den Komfort einer verständlichen Kommunikation mit Maschinen gewährleistet, ist eine der Hauptentwicklungsrichtungen der Informatik. Programmiersprachen ähneln immer mehr natürlichen Sprachen, was die Erstellung einer zuverlässigen Software erleichtert und erheblich beschleunigt.

Die Programmiergeschichte begann mit einer Maschinensprache, in der Befehle in Form von Nullen und Einsen (Binärcode) direkt an den Prozessor im Gerät ausgegeben wurden. Dies bedeutete, dass man für jeden Prozessor separate Anweisungen schreiben musste. Dann erschienen unkomplizierte Programmiersprachen, in denen einige Operationen unter Verwendung von Symbol-Kennzeichnungen ausgeführt werden konnten. Dieser Trend wurde in nachfolgenden Sprachen wie C und C++ verstärkt, die zunehmend vom Maschinencode abwichen.

Sie waren viel universeller und ermöglichten die Lösung einer Vielzahl von Problemen. Darüber hinaus haben diese Sprachen, von denen einige bis heute verwendet und entwickelt werden, die Zeit für die Anwendungsentwicklung erheblich verkürzt und die Fehleranzahl reduziert, dank der Automatisierung von Aufgaben, die zuvor von Programmierern ausgeführt wurden. Vor allem aber erlaubten sie es, die Software von der Hardware-schicht zu trennen, wodurch es möglich wurde, sie zu übertragen, d. h. es gab die Möglichkeit, denselben Code auf verschiedenen Hardwareplattformen oder in unterschiedlichen Betriebssystemen zu verwenden.

Wir haben einen weiteren Schritt nach vorne gemacht und das Indigo-Framework entwickelt – eine Technologie, mit der Maschinen sowie IT-Systeme integriert und mithilfe von Graphen gesteuert werden können. Indigo kann nicht nur auf verschiedene Plattformen übertragen werden, sondern was noch wichtiger ist – es erlaubt, beliebige Maschinen, IT-Systeme (z. B. ERP), SPS oder verschiedene Automaten so zu kombinieren, dass einzelne Knotenpunkte Informationen nicht nur frei untereinander austauschen, aber auch ihr Verhalten gegenseitig kontrollieren können.

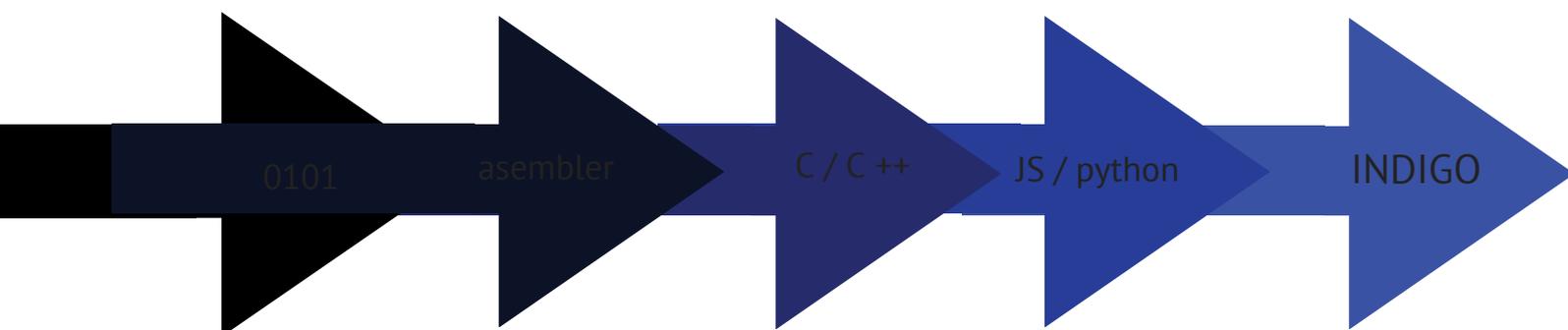
Außerdem besteht das Programmieren in Indigo darin, einzelne Prozesse mit Graphen in BPMN 2.0-Notation zu modellieren. Was ergibt sich daraus? Während in den traditionellen Programmiersprachen das kleinste unabhängige Codefragment tatsächlich eine Liste einfacher Anweisungen ist, die sich seit der Turing-Maschine kaum verändert haben, ist es in unserem Fall ein fertiges Element eines Prozesses, das eine bestimmte Funktion ausführt, z. B. „Fenster mit der Meldung anzeigen“ oder „Befehl an das Gerät senden“.

Um den gesamten Prozess in Indigo zu programmieren, genügt es also, den Verlauf mit Graphen zu modellieren. Einzelne Graphkomponenten können Befehle für den Server sein, um Daten zu speichern oder abzulesen oder eine bestimmte Funktion auszuführen, es können auch Befehle für Geräte sein, um zuvor definierte Aktionen auszuführen, oder eine bidirektionale Kommunikation mit IT-Systemen zu initiieren.

Beim Erstellen von Software in Indigo denken somit wir nicht an den Quellcode, wie bei der Programmierung in traditionellen Sprachen, sondern an seine Funktionen, die durch einzelne Elemente des Graphen ausgelöst werden.

Graphen sind unsere Programmiersprache – verständlich nicht nur für Programmierer.

Abb. 1 (Die Entwicklung der Programmiersprachen)



Warum funktioniert das so gut? Also Architektur und Technologien

Die Indigo-Architektur hat einen modularen und transparenten Charakter. Sie wurde in einer Weise konzipiert, die ihre zukünftige Entwicklung berücksichtigt. Sie ermöglicht auch eine horizontale Skalierung, d. h. es ist möglich, die Leistung der Software durch Hinzufügen zusätzlicher Server (virtueller/physischer) zu steigern. Zusätzlich verfügt das System über integrierte Mechanismen zum Sammeln von Informationen, sodass es bereit ist, große Informationsmengen (von vielen Geräten) mit einer sehr hohen Frequenz zu empfangen.

Auf einer höheren Detailebene kann man die Architektur von Indigo als auf verteilten Knotenpunkten basierend beschreiben, die für die Durchführung spezieller Aufgaben im Zusammenhang mit der Steuerung, Integration sowie Erfassung und Verarbeitung der Daten von Geräten und Benutzern verantwortlich sind. Jeder dieser Knotenpunkte ist unabhängig in Aktion und sammelt und verarbeitet nach dem Trennen von dem Netzwerk, zum Beispiel im Falle eines Ausfalls, weiterhin Informationen, um diese unmittelbar nach der Wiederaufnahme der Kommunikation sofort weiterzuleiten.

Ein besonderes Element des Systems ist der zentrale Knotenpunkt, der in vielen Instanzen betriebsfähig ist und eine hohe Skalierbarkeit der Lösung sowie die Bereitstellung von Diensten im SaaS-Modell ermöglicht.

Der Aufbau einzelner Systemebenen und die Programmierung von Prozessen mithilfe von Graphen ermöglichen eine Echtzeit-Änderung der Betriebslogik jedes Systemelements sowie seine Ersetzung gegen ein anderes mit demselben Funktionsumfang.

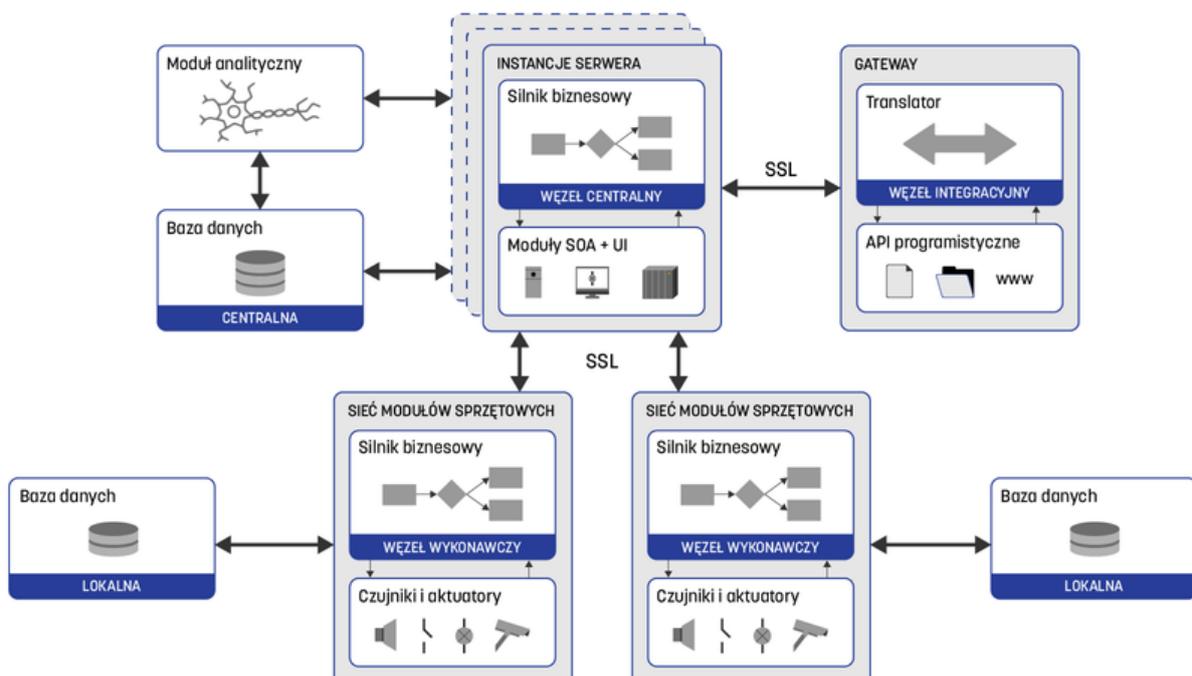
Die Isolierung einzelner Systemebenen und die Programmierung von Prozessen mithilfe von Graphen ermöglichen hingegen eine Echtzeit-Änderung der Betriebslogik jedes Systemelements sowie seine Ersetzung gegen ein anderes mit demselben Funktionsumfang.

Die in dieser Weise konzipierte Architektur erfüllt alle Anforderungen der Industrie 4.0 und ermöglicht die Umrüstung von Fabriken in Echtzeit, die Produktion von Einzelaufträgen in geringer Stückzahl bei gleichzeitiger Gewährleistung von Qualität sowie die Optimierung auf allen Funktionsebenen der Produktion, des Unternehmens und der gesamten Organisation.

Um Benutzern die Nutzung optimaler Hard- und Softwarelösungen verschiedener Hersteller zu ermöglichen und gleichzeitig eine Bindung an nur einen Lieferanten (vendor lock in) zu verhindern, basiert Indigo vollständig auf lizenzfreien OpenSource-Technologien: Das Framework arbeitet mit GNU-Linux-Betriebssystemen und nutzt bewährte, gebräuchliche und entwickelte Lösungen wie PostgreSQL-Datenbankserver, ORM SQLAlchemy, VueJS- und Polymer-Frameworks, nginx-HTTP-Server sowie die Programmiersprachen Python 3.7, JavaScript/TypeScript und C ++.

Die Sicherheit der von Indigo verarbeiteten Daten wird durch die asymmetrische Verschlüsselung der Kommunikation mit SSL/TLS-Protokollen gewährleistet.

Abb. 2 (Architektur der Lösung)



Business-Engine

Das wichtigste Element der Indigo-Technologie ist eine leichte Prozess-Engine, die mit BPMN-Notation - Graphen programmiert wird. Die Engine kann auf Geräten der IoT-Klasse betrieben werden, um deren Verhalten und Kommunikation mit anderen Knotenpunkten und dem Server zu steuern.

Die Konstruktion der Engine beschränkt sich nicht auf ausgewählte Einsatzbereiche. Sie ermöglicht das Aufrufen beliebiger Funktionalitäten, das Steuern von Schnittstellen sowie die Durchführung synchroner und geplanter Aufgaben. Die Engine führt ihre Aufgaben auf Basis der in ihr gespeicherten Graphen aus, die einer weiteren Kompilierung unterliegen. Sie unterstützt Berechtigungen, wodurch die Aufgaben direkt von der Benutzerschnittstelle aus aufgerufen werden können.

Da die Indigo-Technologie auf Skriptsprachen basiert, kann die Software auf beliebigen Geräten ausgeführt werden, die mit der x86-, ARM-Architektur und mit Mikrocontrollern betrieben werden. Durch die Integration mit der Electron-Technologie und den geringen Rechenleistungsbedarf kann Indigo auf mobilen Geräten verwendet werden: Tablets, Mobiltelefonen, Handhelds und seine Desktopanwendung auf PCs und Industrie-PCs.

Die externe und interne Kommunikation des Systems verfügt über ein eigenes, auf JSON basierendes leichtes und standardisiertes Protokoll, das die Erfassung und Aufzeichnung von Daten in kohärenter Form abhängig von ihrer Ursprungsquelle ermöglicht (wir nennen es funktionale Kommunikation). Alle Daten können gespeichert werden (jeder Knotenpunkt hat eine eigene Datenbank), sodass der Prozess z. B. nach einem Neustart fortgesetzt werden kann. Derzeit basiert das System auf der PostgreSQL-Datenbank. Dank der Anwendung der SQLAlchemy-Zwischenschicht können jedoch auch jegliche andere, moderne, relationale Datenbanken angeschlossen werden.

Die Engine erlaubt die gleichzeitige synchrone und asynchrone Verwendung mehrerer Prozesse sowie ihre Planung in Cron. Es ist möglich, Meldungen im Offline-Fall in die Warteschlange aufzunehmen und sie nach Wiederherstellung der Verbindung zu senden.

Graphen

Die Indigo-Technologie basiert auf grafischer Programmierung. Jede logische Schicht wird mit Graphen programmiert - von der Steuerung der Hardwaremodule über die Beschreibung der Betriebslogik vieler Begleitmodule des Geräts, den Informationsfluss in der Gerätegruppe bis hin zur Übertragung von Informationen zwischen verschiedenen Gerätegruppen und IT-Systemen, auch an entfernten Standorten.

Die Graphen steuern auch den Betrieb der Schnittstelle, d. h. das Anzeigen und Ausblenden von Fenstern, ihre interne Betriebslogik sowie den angezeigten Inhalt. Das System definiert nur Templates, die die Struktur der Ansicht und der Komponenten bestimmen, aus denen z. B. eine Tabelle, ein Text oder eine Schaltfläche besteht.

Die Graphen definieren die Reaktion auf Meldungen der Schnittstelle, der Hardware und anderer Prozesse, indem sie die nächsten auszuführenden Schritte, die Bedingungen für ihre Ausführung sowie die zu sendenden Nachrichten bestimmen. Mit ihrer Hilfe werden auch die Bedienung von Fehlern und das Verhalten im Offline-Modus definiert.

Die Durchführung von Aktionen am Graphen basiert auf dem Aufrufen von Funktionalitäten aus generischen Bibliotheken – wenig funktionalen wie z. B. Unterstützung von Variablen, Kommunikationsunterstützung über Ports usw. und hoch funktionalen, wie z. B. ein Raumbeschreibungsmodul (z. B. für ein Lager) oder ein Registrierungsmodul für Ausfallzeiten.

Es wird die Gestaltung eines dedizierten WISIWYG-Tools zum Erstellen von Ansichten und Zeichnen sowie zum Debuggen von Diagrammen geplant. Derzeit wird die Open-Source-Software yED verwendet.

Integration mit externen IT-Systemen

Das Indigo-Framework kann auf verschiedene Arten in externe Systeme integriert werden.

1. Mithilfe eines Anschlusses, der die Funktionalitäten eines fremden Systems mithilfe unserer funktionalen Kommunikation auslöst. Der Konnektor arbeitet im System nach dem Prinzip des nächsten Knotens.
2. Mithilfe eines Moduls, das im Graphen verwendet wird.
3. Mithilfe der ESB-Integrationsschiene.

Deployment

Indigo unterstützt Continuous Deployment vollständig und ermöglicht ferngesteuertes Laden von Ansichten und Ressourcen sowie das Ersetzen von Graphen. Das System sammelt die Protokolle aller Knotenpunkte auf zentralisierte Weise und in Echtzeit.

Das Framework ist vollständig modular, d. h. das Deployment enthält nur Module, die für den Betrieb erforderlich sind.

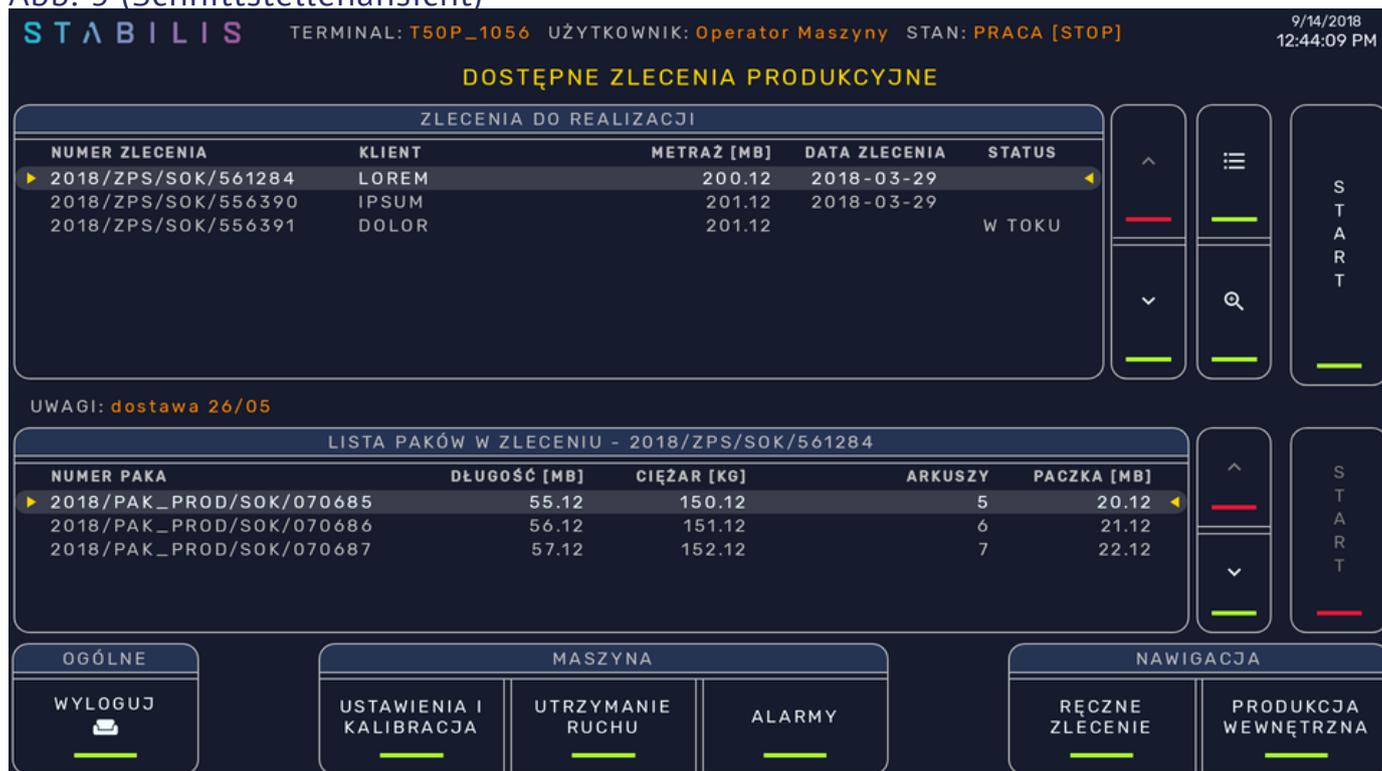
Benutzerschnittstelle, also wie wir mit dem Benutzer kommunizieren

Hohe Flexibilität und Konfigurierbarkeit von Indigo gilt auch für die Benutzerschnittstelle, die wir gemäß den neuesten Trends in der Entwicklung von Webanwendungen entworfen haben.

Sie basiert auf Desktops und Widgets, aus denen wir Management-Panels konfigurieren. Der Aufbau unserer Schnittstelle ermöglicht die Darstellung verschiedener Arten von Informationen in verschiedenen grafischen und funktionalen Layouts, die sich aus den Anforderungen der gegebenen Implementierung ergeben. Die Erstellung dedizierter Bildschirme ist einfach und schnell, da sie auf der Ebene der Schnittstelle selbst, ohne Konfiguration in den Dateien und zusätzliche Programmierung zusammengestellt werden (WYSIWYG). Dadurch kann die Benutzeroberfläche für jeden Benutzer individuell gestaltet werden. Es ist auch möglich, eine einheitliche Schnittstelle für einen nicht homogenen Maschinenpark zu erstellen, der als HMI-Schnittstelle arbeitet.

Abb. 3 (Schnittstellenansicht)

Abb. 3 (Schnittstellenansicht)



Wir verwenden Graphen auch zur Steuerung der Schnittstelle

Indigo unterstützt vollständig die Berechtigungsmechanismen für verbundene Rollen, Internalisierung, Änderungen der Größe von Oberflächenelementen sowie Auflösung. Es ist auch möglich, das Layout der Schnittstelle (auch während des Betriebs) im Rahmen des Farbschemas, des Aussehens von Steuerelementen usw. anzupassen.

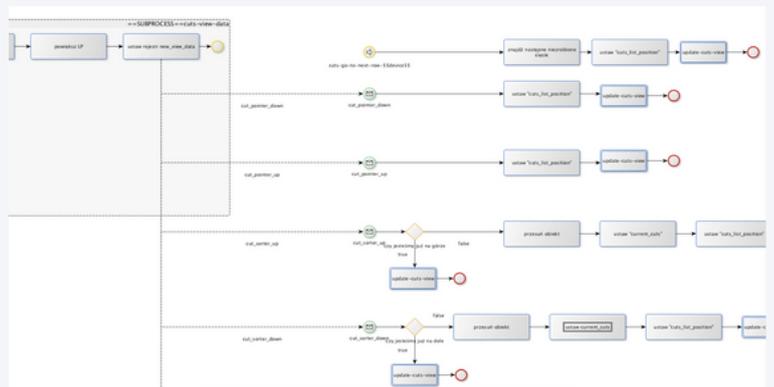
Die Steuerelemente werden automatisch in definierten Sektionen verteilt (die Benutzeroberfläche wurde in HTML5 mithilfe der Flex-Technologie geschrieben).

Wie machen wir das?

Ein Template, die das Verhalten der Ansicht beschreibt
Abb. 4 (Graphik – Code)

```
<template-section id='body_section' type='vertical' scheme='fullscreen'>
  <template-section id='header_section' scheme='header'>
    <info-picture id='stabilis_logo' icon='images/stabilis-logo.png' position='start' left='--separation-big' right='--separation-big' />
    <info-line id='console_info' position='start' left='--separation-big' right='--separation-big' interspace='--separation-big' />
    <date-and-time id='clock' position='end' right='--separation-big' stick='end' />
  </template-section>
  <template-section id='main_section' type='vertical'>
    <template-section id='cuts_section' align='start'>
      <template-section id='table_section' type='vertical' size='stretch'>
        <info-table id='cuts_table' grow='expand' />
        <info-line id='comments' size='--font-big' />
      </template-section>
      <template-section id='input_section' type='vertical' align='end' grow='none' size='stretch'>
        <input-number id='input_1kat' />
        <input-number id='input_2kat' stick='end' />
        <input-number id='input_waste' bottom='--separation-small' />
      </template-section>
    </template-section>
    <template-section id='machine_section' size='stretch' grow='none'>
      <info-box id='machine_info' />
    </template-section>
    <template-section id='actions_section' scheme='commands'>
      <execute-command id='actions_navigate' />
      <execute-command id='actions_cuts' />
      <execute-command id='actions_machine' />
    </template-section>
  </template-section>
</template-section>
```

Ein Fragment des Programms in Form eines Graphen, der das Verhalten der Ansicht beschreibt
Abb. 5 (Grafik – Graph)



Ein beispielhafter Systembildschirm, der für die Maschinensteuerung und die Parameterdarstellung verantwortlich ist. Das Layout und das Verhalten wurden vollständig anhand eines Templates und eines Prozessgraphen beschrieben.
Abb. 6 (Schnittstellenansicht)

LP	PAK	DLUGOŚĆ [MB]	ILOŚĆ [SZT]	I KAT. [SZT]	II KAT. [MB]	ODPAD [MB]	STATUS
1	P03	8500	15	15	0	0	OK
2	P03	7000	12	12	0	14	OK
3	P03	3500	43	43	0	0	OK
4	P03	8000	3	3	0	8	OK
5	P03	10000	21	8	0	0	W trakcie
6	P03	1000	70	0	0	0	
7	P03	1500	10	0	0	0	
8	P03	1250	1	0	0	0	
9	P03	1750	30	0	0	0	

UWAGI: TYLKO PIERWSZY GATUNEK

BIEŻĄCE PARAMETRY PRODUKCJI

KRAJ: 6484/127/01/18 DLUGOŚĆ AKTUALNA: -2 mm PAUZA: 5 sek.
DOSTEPNE: 350 m.b. WYKORZYSTANE: 80 m.b. POZOSTANIE: 120 m.b.

WYKONAJ DZIAŁANIE

ZLECENIA INSTRUKCJA UŻYCIA DODAJ CIĘCIE USUŃ CIĘCIE SORTUJ WG DŁUGOŚCI USTAWIENIA I KALIBRACJA ROZLICZ KRAJ

Technologische Vorteile

Die größten Vorteile der Indigo-Technologie:

- Deutliche Beschleunigung der Softwareentwicklung (von einer bis zwei Größenordnungen)
- Stabilität
- Verringerung der Anzahl von Logik-, Regressions- und Integrationsfehlern
- signifikante Reduzierung der Kosten für sich während des Projekts ändernde Anforderungen
- Möglichkeit der evolutionären Einführung weiterer Funktionalitäten im Einklang mit agilen Softwareentwicklungsmethoden
- Möglichkeit der Projektumsetzung durch Personal mit geringeren Kompetenzen (Implementierung anstatt Programmierer)
- Erleichterung und Beschleunigung des Analyseprozesses mit dem Kunden
- Möglichkeit, sehr komplexe Probleme zu implementieren.

Einzigartige Eigenschaften und Zusammenfassung

Was das Indigo-Framework auszeichnet, ist die mit Graphen programmierte bidirektionale Kommunikation mit Maschinen oder IT-Systemen.

Wir können nicht nur Daten sammeln, sondern auch auf sie reagieren, indem wir Rückbefehle für Maschinen oder Rückmeldungen für IT-Systeme (z. B. ERP) senden.

Dadurch, dass jeder Prozess, der in einem beliebigen System oder Maschinennetzwerk ausgeführt wird, mit einem Graphen programmiert wird, ist es möglich, die Betriebslogik des Systems, das mit unserer Software ausgeführt wird, nahezu in Echtzeit zu ändern. Es reicht aus, den Graphen auszutauschen, ohne Hunderte weiterer Codezeilen schreiben zu müssen.

Dank der Graphen erhöhen wir außerdem die Passgenauigkeit und Zuverlässigkeit unserer Software, da das Entwerfen von Funktionalitäten für den Kunden verständlich ist und sogar in der Implementierungsphase modifiziert werden kann, wenn sich manchmal herausstellt, dass bestimmte Prozesse in der Praxis anders gestaltet werden sollten als vorher angenommen.

Dank dieser Herangehensweise zeichnet sich die Indigo-Software durch ihre außergewöhnliche Flexibilität aus, die sich durch die Möglichkeit einer schnellen Einführung von Änderungen, die Offenheit für die Integration mit neuen Technologien und die Möglichkeit des Hinzufügens neuer Funktionen durch die Konfiguration unseres Frameworks auszeichnet, ohne dass eine zusätzliche Software erstellt werden muss.

Kurz gesagt, wir setzen die Postulate des Rapid Application Development (RAD) vollständig um.

Wir sind überzeugt, dass nur eine solche Herangehensweise es ermöglicht, vollständig über Industrie 4.0 sprechen zu können. Im Gegensatz zu IIoT, das nur als Sammeln und Analyse der Daten von Geräten verstanden wird, bezeichnen wir unsere Technologie Control overIIoT – ColloT.

Wo kann die Indigo-Technologie eingesetzt werden?

Die Flexibilität von Indigo gibt die Möglichkeit, das Framework in verschiedenen Bereichen einzusetzen. Unsere Technologie wird sich überall dort bewähren, wo große Datenmengen von vielen Geräten erfasst sowie analysiert werden und wo auf die gesammelten Daten reagiert werden muss. Unabhängig davon, ob es sich um die Integration von Maschinen in der Fabrik, um Gebäudeautomationssysteme oder verschiedene Hardwaremodule innerhalb eines Geräts (z. B. Vendomat) handelt, passen wir uns immer an die individuellen Bedürfnisse an.

Anwendungsbereiche:

- Datenerfassung, Analyse und Steuerung der Industrieautomatik.
- Intelligente Transportsysteme (Datenerfassung, Steuerung, Managementzentrum)..
- Entwerfen von anspruchsvollen Servicemaschinen und deren Vernetzung (Verkaufsautomaten, Car-/Bikesharing usw.).
- Gebäudemanagementsysteme (Integration von Gebäudesystemen, Steuerung der Gebäudeautomation).
- Ausführung von Logistikprozessen (Mauterhebung, Verkauf, Zählung, Servicezentren, Prozessoptimierung).
- Treue- und Zugangssysteme (Skipässe, Zutritts- und Durchgangskontrolle, Treue- und Einkaufskarten).
- Metering (Strom, Gas, Wasser, andere).

Industrie 4.0

Vielfalt ist unserer Meinung nach eine der wichtigsten Inspirationen der Gegenwart. Vielfalt stimuliert am stärksten die Schaffung neuer Technologien und die Veränderung der Herangehensweise an die Methode der Softwareentwicklung.

Im technologischen Bereich sind dies Maschinen und Geräte sowie Informationssysteme. Die Integration einer vielfältigen Infrastruktur erlaubt eine umfassende Analyse und Datenverwaltung, was die Optimierung sowie Schaffung neuer Lösungen und Einkommensquellen ermöglicht. Vereinheitlichung ist der Haupttrend der Neuzeit und das grundlegende Postulat der Industrie 4.0.

Die globale digitale Transformation dringt immer gewagter in den Industriesektor ein. Die wachsende Nachfrage nach Werkzeugen zur Optimierung von Produktions- und Geschäftsprozessen hat uns dazu inspiriert, das erste Produkt auf Basis unseres Frameworks zu entwickeln - einer Software, die eine schnelle Erstellung dedizierter Systeme ermöglicht, welche die Integration von Maschinen und IT-Systemen erlauben, um Lean-Manufacturing-Praktiken zu implementieren, also eines informationsbasierten modernen Managements des Produktionsunternehmens. Wir haben sie Stabilis genannt.

Stabilis ist eine Reihe von Produkten zur Optimierung der Produktion, u. a. im Bereich Just in Time, TPM, TQM und zur Unterstützung von Kaizen, die die Produktivität steigern und den Personalbedarf reduzieren.

S T A B I L I S

P O W E R E D B Y I N D I G O