



netidee prj1177
macchina.io
Endbericht

Günter Obiltschnig
Applied Informatics Software Engineering GmbH

2017-08-04

Inhalt

1	Einleitung.....	3
1.1	macchina.io	3
1.2	Netidee Förderprojekt	4
2	Neue Features und Arbeitspakete.....	4
2.1	Verbesserungen am JavaScript Laufzeitsystem	5
2.2	Task Scheduler Service	5
2.3	Network Environment Service	5
2.4	Device Status Service	6
2.5	Web Applikation zur Konfiguration	6
2.6	Web Applikation zur Protokollierung und Darstellung von Sensordaten	6
2.7	Modbus Protokoll	7
2.8	Bluetooth LE	7
2.9	6LoWPAN	7
2.10	Zusätzliche Hardwareunterstützung	8
2.11	Ubuntu Snap	8
2.12	OpenWRT Paket	8
2.13	Windows Unterstützung	8
2.14	QNX Unterstützung	8
2.15	Dataflow Engine und Editor	8
2.16	OPC-UA	9
3	Kundenprojekte.....	9
3.1	Gebäudeautomatisierung	9
3.2	Fahrzeugtelematik	10
3.3	IoT Integrationsplattform	10
3.4	Energie Management System	10
4	Offene Arbeitspakete.....	10
5	Marketing-Aktivitäten.....	10
5.1	2015-2016	10
5.2	2017	11

1 Einleitung

macchina.io ist ein modularer Softwarebaukasten zur Entwicklung von Edge- und Fog Computing Applikationen für das Internet der Dinge. Das Projekt wurde im Förderjahr 2015 durch Netidee gefördert. Dieser Endbericht gibt einen Überblick über macchina.io, sowie über die im Rahmen des Förderprojekts umgesetzten Arbeitspakete.

1.1 macchina.io

macchina.io ist ein modularer Softwarebaukasten zum schnellen Prototyping, sowie zur Entwicklung von Edge- und Fog Computing Applikationen für das Internet der Dinge. Diese Applikationen laufen auf vernetzten, typischerweise Linux-basierten Geräten, sowie IoT Gateways und kommunizieren einerseits mit lokal angebenen Sensoren, Aktoren und anderen Geräten, sowie andererseits auch mit Cloud Services. Zu diesem Zweck bietet macchina.io eine Vielzahl an Softwaremodulen die flexibel kombiniert werden können. So werden z. B. verschiedene Kommunikationsprotokolle und -technologien aus dem IoT- und Automatisierungsbereich unterstützt, wie z. B. Bluetooth LE, 6LoWPAN, Modbus oder OPC-UA, aber auch Internet Technologien zur Kommunikation mit Cloud Services wie HTTP, REST und MQTT.

macchina.io unterstützt mehrere Programmiersprachen, bzw. Programmiermodelle. Als Programmiersprachen werden derzeit C++ und JavaScript unterstützt. C++ wird vornehmlich für low-level Code (z. B. Geräteanbindungen und Kommunikationsprotokolle) verwendet, während JavaScript für die Entwicklung der Anwendungslogik stark vereinfacht und beschleunigt. Außerdem ist der Kern von macchina.io selbst in C++ implementiert. Dadurch eignet sich macchina.io auch gut für kleinere, Ressourcen-limitierte Systeme, bzw. Systeme mit hohen Anforderungen bezüglich Performance und Effizienz, speziell im Vergleich zu Java-basierten Plattformen wie z. B. Eclipse Kura. Zusätzlich steht eine Dataflow Engine mit einem graphischen, web-basiertem Editor zur Verfügung um einfache Applikationen auch ohne Code realisieren zu können. Unterstützung für weitere Programmiersprachen wie z. B. Python wäre in Zukunft ebenso möglich.

Ein integrierter Web Server mit diversen vorgefertigten Web-Applikationen ermöglicht schnelles Ausprobieren von eigenen Ideen. Umfangreiche Funktionen zum Web-basierten Gerätemanagement (Konfiguration, Installation von neuen Software-Komponenten) sind vorhanden.

Gegenüber anderen verfügbaren IoT Plattformen grenzt sich macchina.io unter anderem durch folgende Eigenschaften ab:

- Fokus auf Edge-, bzw. Fog-Computing. Geräte dienen nicht nur als einfache Datensammler, die Sensordaten an ein Cloud Service übermitteln, sondern sie sollen Daten lokal, am Entstehungsort, verarbeiten, analysieren und filtern können ("Edge Computing"). Dies

kann auch im Verbund mit anderen lokalen Geräten erfolgen ("Fog-Computing").

- Fokus auf Skalierbarkeit, Effizienz und Performance durch die Verwendung von C++ und eine extrem modulare Architektur.
- Unterstützung von JavaScript. Dadurch kann Anwendungslogik auf einer hohen Abstraktionsebene, auch von Entwicklern die kein C++ beherrschen, entwickelt werden.

macchina.io steht auf Github [<https://github.com/macchina-io/macchina.io>] unter der Apache 2.0 License zur Verfügung.

1.2 Netidee Förderprojekt

Im Rahmen des netidee Förderprojekts konnten von November 2015 bis Juli 2017 zahlreiche Verbesserungen, Optimierungen und Erweiterungen an macchina.io vorgenommen werden. Aufgrund mehrerer parallel laufender Kundenprojekte mit macchina.io und damit einhergehender intensiver Beratungs- und Implementierungstätigkeit, sowie durch die verspätete, bzw. eingeschränkte Verfügbarkeit von eingeplanten Entwicklerressourcen konnten jedoch nicht alle geplanten Arbeitspakete umgesetzt werden, bzw. mussten zum Teil andere vorab nicht geplante Features vorgezogen werden.

2 Neue Features und Arbeitspakete

Folgende neue Features und Verbesserungen konnten im Rahmen des Projektes in macchina.io implementiert werden:

- Verbesserungen am JavaScript Laufzeitsystem
- Task Scheduler Service
- Network Environment Service
- Device Status Service
- Web Applikation zur Konfiguration
- Web Applikation zur Protokollierung und Darstellung von Sensordaten
- Modbus Protokoll
- Bluetooth LE
- 6LoWPAN
- Zusätzliche Hardware- und Sensorunterstützung
- Ubuntu Snap
- OpenWRT Package
- Windows Unterstützung
- Dataflow Engine und Editor
- OPC-UA

2.1 Verbesserungen am JavaScript Laufzeitsystem

Ein relativ großer Aufwand floss in die Verbesserungen am JavaScript Laufzeitsystem. Neben neuen Features wurde vor allem an der Stabilität und Performance, sowie der Modularisierung des JavaScript Laufzeitsystems gearbeitet. Aus AP1 wurde das Buffer Object, sowie die Unterstützung für das Arbeiten mit binären Datenblöcken implementiert. Aus AP11 wurde das Update der V8 Engine auf eine neue Release durchgeführt. Für die 1.0 Release wurde von Version 3.25 auf Version 3.28 gewechselt. In einem separaten Branch wurde auch eine Anpassung an aktuelle Release 5.x implementiert. Nach der Release 1.0 von macchina.io wird die V8 Engine auf eine aktuelle Release aktualisiert. Hierfür waren, bzw. sind jedoch signifikante Änderungen an der V8 Integration in macchina.io erforderlich, da die APIs der V8 Engine in der Zwischenzeit stark verändert wurden. Sehr viel Zeit wurde in Tests, Bugfixes und Optimierung des Speicherverbrauchs und der Stabilität von JavaScript Programmen investiert.

Ein wichtiger Schritt war auch die Modularisierung des JavaScript Laufzeitsystems durch die Erweiterung des bestehenden *CommonJS* Modulsystems um native (in C++ geschriebene) Module. Dadurch konnten die Netzwerk-, Web- und Datenbank-Bindings aus dem Kernsystem entfernt, und als optionale Module bereitgestellt werden. An den bestehenden Bindings wurden zahlreiche Verbesserungen implementiert. Bestehende Programmierschnittstellen wurden nochmals überarbeitet und verbessert.

Schlussendlich wurde auch die Kompatibilität des macchina.io JavaScript Laufzeitsystems mit anderen Laufzeitsystemen (diverse Web Browser, node.js) erhöht, in dem verschiedene JavaScript Objekte und Funktionen, die dort vorhanden sind, ebenfalls implementiert wurden. Dies ermöglicht die einfachere Verwendung von bestehendem JavaScript Code aus anderen Umgebungen in macchina.io (z. B. Test Frameworks).

2.2 Task Scheduler Service

Das Task Scheduler Service aus AP1 wurde vollständig implementiert. Es ermöglicht die zeitgesteuerte Ausführung von JavaScript Code, ähnlich wie der Cron Daemon auf Unix/Linux Systemen.

2.3 Network Environment Service

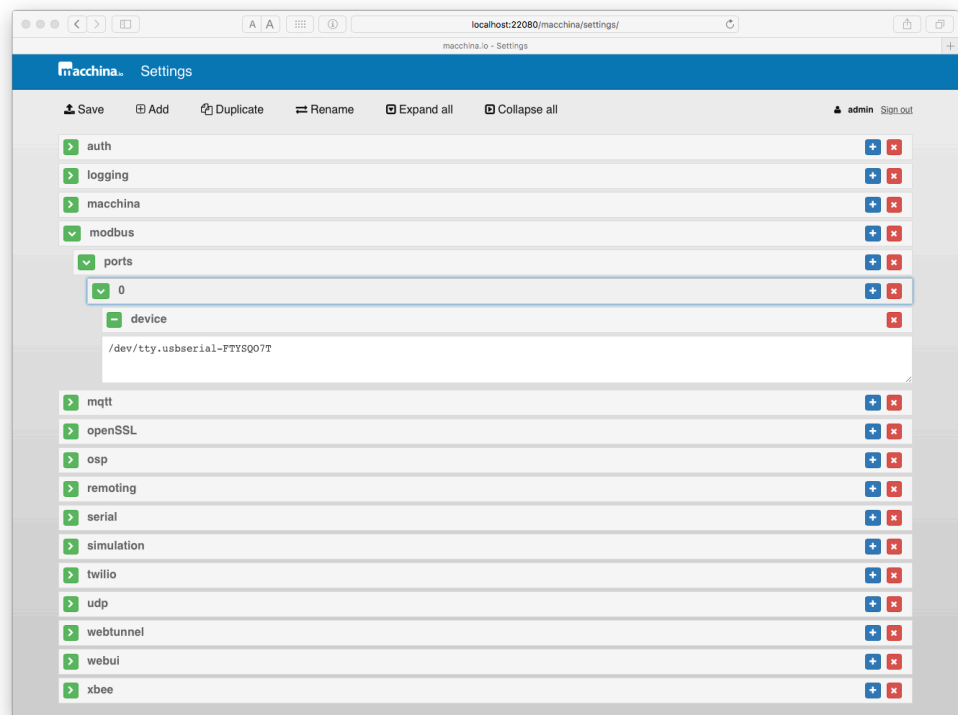
Das ursprünglich nicht vorgesehene Network Environment Service wurde implementiert. Es ermöglicht einerseits die Benachrichtigung wenn sich die Netzwerkumgebung des Geräts ändert (z. B. Wechsel von Ethernet auf WLAN, oder von WLAN auf mobile Datenverbindung), andererseits die Abfrage von verfügbaren Netzwerkschnittstellen und IP Adressen.

2.4 Device Status Service

Das ursprünglich nicht vorgesehene Device Status Service stellt eine zentrale Möglichkeit zur Verfügung, den Zustand eines Gerätes (im Hinblick auf Fehlermeldungen, Warnungen oder sonstige Mitteilungen) zu verwalten und zu überwachen. So kann bei Auftreten eines Fehlers eine Notifizierung des Benutzers erfolgen, z. B. durch eine blinkende LED, oder auch durch das Versenden einer SMS Nachricht. Auch werden alle auftretenden Nachrichten für den Benutzer an zentraler Stelle gespeichert und über eine Web Applikation sichtbar gemacht.

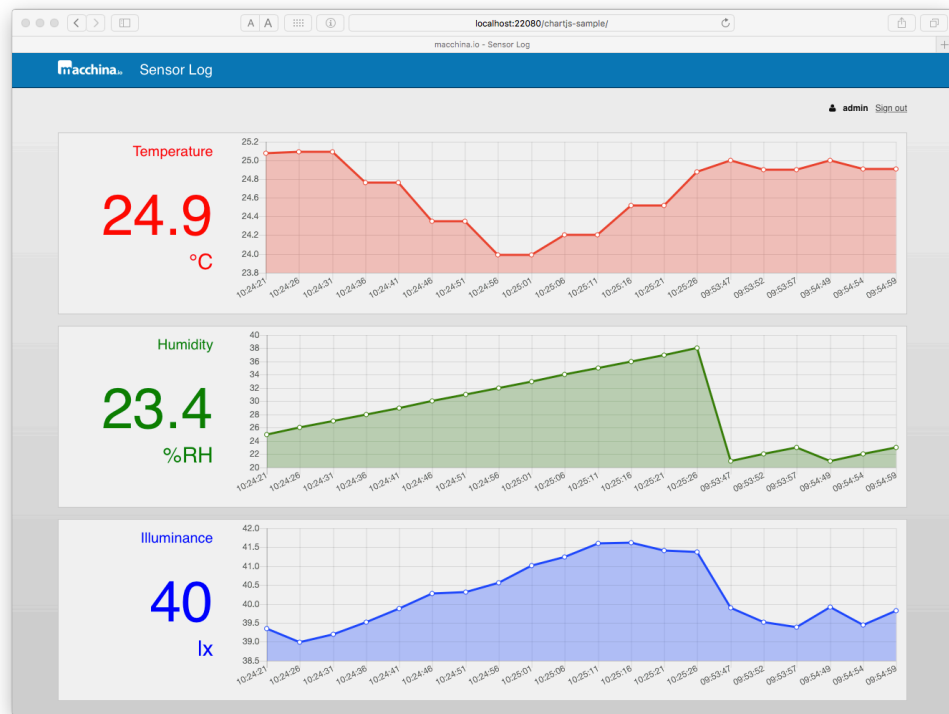
2.5 Web Applikation zur Konfiguration

Es wurde eine Web Applikation zur Konfiguration eines macchina.io-basierten Gerätes implementiert. Aller Konfigurationsparameter werden hierarchisch dargestellt und können bearbeitet und gespeichert werden.



2.6 Web Applikation zur Protokollierung und Darstellung von Sensordaten

Diese Applikation ermöglicht es, Daten von bestimmten Sensoren regelmäßig auszulesen und in einer Datenbank zu protokollieren. Weiters können die aufgezeichneten Daten in Form von Kurven dargestellt werden.



2.7 Modbus Protokoll

Das in AP2 vorgesehend Modbus Protokoll zur Anbindung von externen Geräten wurde implementiert und mit verschiedenen Geräten getestet und ist auch schon produktiv bei einem Kundenprojekt im Einsatz.

2.8 Bluetooth LE

Die Unterstützung für das Bluetooth LE Protokoll basierend auf dem Linux Bluez Stack wurde fertig gestellt. Tests mit diverser Hardware (TI Sensor Tag, Bosch XDK, Pulsmessgurte, Beacons) wurden durchgeführt.

2.9 6LoWPAN

Unterstützung für das 6LoWPAN Protokoll für drahtlose Sensornetzwerke wurde implementiert (AP4). In diesem Zusammenhang wurde macchina.io auf das Phytex Wega (IoT Enablement Kit), sowie das Creator Ci40 Board portiert (wobei auf letzterem die JavaScript Engine noch nicht funktioniert). Im Rahmen von Tests mit Funksensoren wurde ein Treiber für ein Bosch Multi-Sensormodul (Temperatur, Luftfeuchtigkeit, Luftdruck) für das 6LoWPAN Click Board portiert und an das LetMeCreateIoT Open Source Projekt contributed.

2.10 Zusätzliche Hardwareunterstützung

Unterstützung für weitere Sensor- und Gerätetypen wurde implementiert (AP6). Weiters wurde macchina.io auf diversen Zielsystemen getestet (Raspberry Pi 3, Beaglebone Black, Udo, Phytex, Moxa, kundenspezifische Hardware).

2.11 Ubuntu Snap

Auf Anfrage und mit Unterstützung des Ubuntu Core Teams wurde ein Snap (eine Deployment Package) für Ubuntu Core erstellt. Dies ermöglicht die einfache Installation von macchina.io auf einem Ubuntu Core basierendem Gerät.

2.12 OpenWRT Paket

Im Rahmen der Portierung auf das Creator Ci40 Board wurde ein OpenWRT Paket für macchina.io erstellt.

2.13 Windows Unterstützung

Das Kernsystem von macchina.io (inkl. JavaScript Engine) wurde unter neben Linux und OS X auch unter Windows lauffähig gemacht.

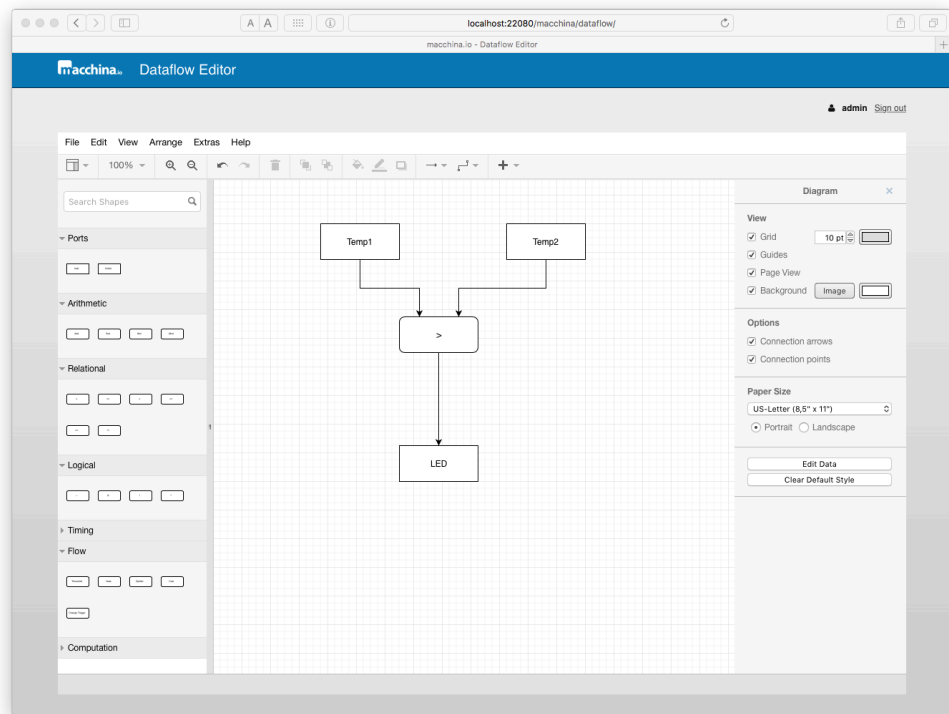
2.14 QNX Unterstützung

Im Rahmen eines Kundenauftrags wurde der Kern von macchina.io (ohne JavaScript Engine) auf das Echtzeitbetriebssystem QNX portiert.

2.15 Dataflow Engine und Editor

Die Dataflow Engine ermöglicht die Realisierung von einfachen ereignisgesteuerten Abläufen (z. B. Lesen von Sensordaten, Berechnungen, Ausführen von Aktionen) ohne ein C++ oder JavaScript Programm dafür schreiben zu müssen. Vordefinierte Verarbeitungsblöcke können über eine XML Datei miteinander verbunden werden. Weiters steht ein Web-basierter, graphischer Editor zur Verfügung um Datenflüsse zu beschreiben.

Ein Prototyp der Dataflow Engine ist vorhanden. In den nächsten Schritten erfolgt die Entwicklung zur Produktreife, sowie die Entwicklung des Web-basierten Editors, wobei hier die *mxgraph* JavaScript Library als Basis dient.



2.16 OPC-UA

Der Industrie 4.0, bzw. Industrial IoT (IIoT) Bereich ist ein wichtiger Zielmarkt für macchina.io. Immer mehr stellt sich heraus, dass sich OPC-UA zu einem der wichtigsten Protokolle in diesem Bereich entwickelt. Gleichzeitig stehen mittlerweile auch gute Open Source Implementierungen dieser Technologie zur Verfügung. Ziel ist es daher, auch OPC-UA Unterstützung in macchina.io anzubieten. Hierzu wurde die relativ neue, u.a. vom Fraunhofer Institut entwickelte *open62541* Open Source Library in macchina.io integriert. Der so entwickelte OPC-UA Client in macchina.io kann von C++ und JavaScript aus zur Kommunikation mit Systemen der Automatisierungstechnik verwendet werden.

3 Kundenprojekte

Im Laufe des Projekts konnten mehrere Kundenprojekte mit macchina.io durchgeführt werden. Dadurch konnte wertvolles Feedback für die weitere Entwicklung von macchina.io gewonnen werden. Im Rahmen dieser Projekte wurden die Kunden von Applied Informatics beim Einsatz von macchina.io unterstützt. Auch wurden kundenspezifische Erweiterungen implementiert.

3.1 Gebäudeautomatisierung

Mit einem international führenden Hersteller von Gebäudeautomatisierungssystemen wurde basierend auf macchina.io Software für ein Gerät entwickelt, welches Techniker beim Test und bei der Inbetriebnahme von

Gebäudeautomatisierungssystemen unterstützt. Dazu wurde macchina.io um Unterstützung für ein spezielles Netzwerkprotokoll aus dem Gebäudeautomatisierungsbereich erweitert (kundenspezifisch, nicht open source).

3.2 Fahrzeugtelematik

Ein Komponenten-Zulieferer aus dem Automobil-Bereich entwickelt mit macchina.io als Basis ein Telematik-System, welches e-Call Funktionalität mit Telematik-Diensten kombiniert. Dazu steht eine mobile Internet-Anbindung zur Verfügung. Das System wird ab 2018 in verschiedenen Fahrzeugenmodellen mehrerer bekannter europäischer Automobilhersteller eingesetzt und kann einfach um neue Dienste (z. B. Fleet Management, Mobile App Integration) erweitert werden.

3.3 IoT Integrationsplattform

macchina.io wird als Basis der IoT Software-Integrationsplattform eines internationalen Technologieherstellers (u. a. von Präzisionsmessgeräten) verwendet.

3.4 Energie Management System

Ein Startup aus der Steiermark verwendet macchina.io als Basis eines innovativen Energie Management Systems für Photovoltaik-Anlagen. Dieses Projekt wurde beim Constantinus Wettbewerb mit einem 3. Platz ausgezeichnet und schaffte es beim Staatspreis Innovation unter die 6 besten Beratungsprojekte österreichweit.

4 Offene Arbeitspakete

Folgende Arbeitspakete, bzw. Tätigkeiten wurden vorerst aufgeschoben:

- Die Implementierung des CoAP Protokolles wurde aus Zeitgründen (und mangels Nachfrage von Kunden) nicht fertiggestellt.

5 Marketeing-Aktivitäten

5.1 2015-2016

Folgende Marketing Aktivitäten wurden im Rahmen des Projekts durchgeführt:

- Teilnahme am ESE (Embedded Software Engineering) Kongress 2015 in Sindelfingen als Aussteller (gemeinsam mit Partnerfirma) und Vortrag zum Thema IoT (30.11. – 03.12.2015).

- Teilnahme am Austria IoT Day in Wien am 04.11.2015 als Aussteller und mit macchina.io Workshop.
- Teilnahme als Referent an der Meeting C++ Konferenz in Berlin mit Vortrag zum Thema macchina.io (04.12. – 05.12.2015).
- macchina.io Workshop bei IoT Vienna am 28.01.2016.
- Teilnahme an der building IoT Konferenz in Köln als Aussteller und Referent (10.05. – 12.05.2016).
- macchina.io Workshop bei IoT Ljubljana am 08.09.2016.
- Teilnahme als Referent an der IoT With The Best online Konferenz mit Vortrag zu macchina.io (29.10. – 30.10.2016).
- Teilnahme als Referent am JS Kongress in München mit Vortrag zu macchina.io (28.11. – 29.11.2016).

Im November 2016 wurde die Unterstützung eines externen Beraters in Anspruch genommen. Ziele waren einerseits eine Reflexion des Projektes mit Hilfe externer Sicht (Wahrnehmung), sowie die zukünftige Positionierung und Weiterentwicklung des Projekts.

Beim Innovations- und Forschungspreis des Landes Kärnten 2016 schaffte es macchina.io unter die drei nominierten Projekte in der Kategorie Kleinstbetriebe.

5.2 2017

- Teilnahme Staatspreis Digital Solutions (leider ohne Nominierung)
- Vortrag bei buildingIoT London (27.-29. März) [diese Konferenz wurde leider abgesagt]
- Teilnahme am Global IoT Day Event der IoT Austria (9. April)
- Vorträge bei lokalen IoT Meetups in Österreich und im benachbarten Ausland
- Teilnahme als Vortragender und Aussteller/Sponsor beim Internet of Things Kongress am 13. und 14. September in München
- Teilnahme als Vortragender bei der Internet-of-Things – Vom Sensor bis zur Cloud Konferenz am 19. Oktober in München
- Teilnahme als Vortragender beim Automotive Software Kongress am 27. September in Landshut
- Kooperation mit dem Silkroad4.0 Projekt als Unterstützer und Technologie-Provider. In diesem Rahmen wird macchina.io auch bei diversen Unternehmen präsentiert.