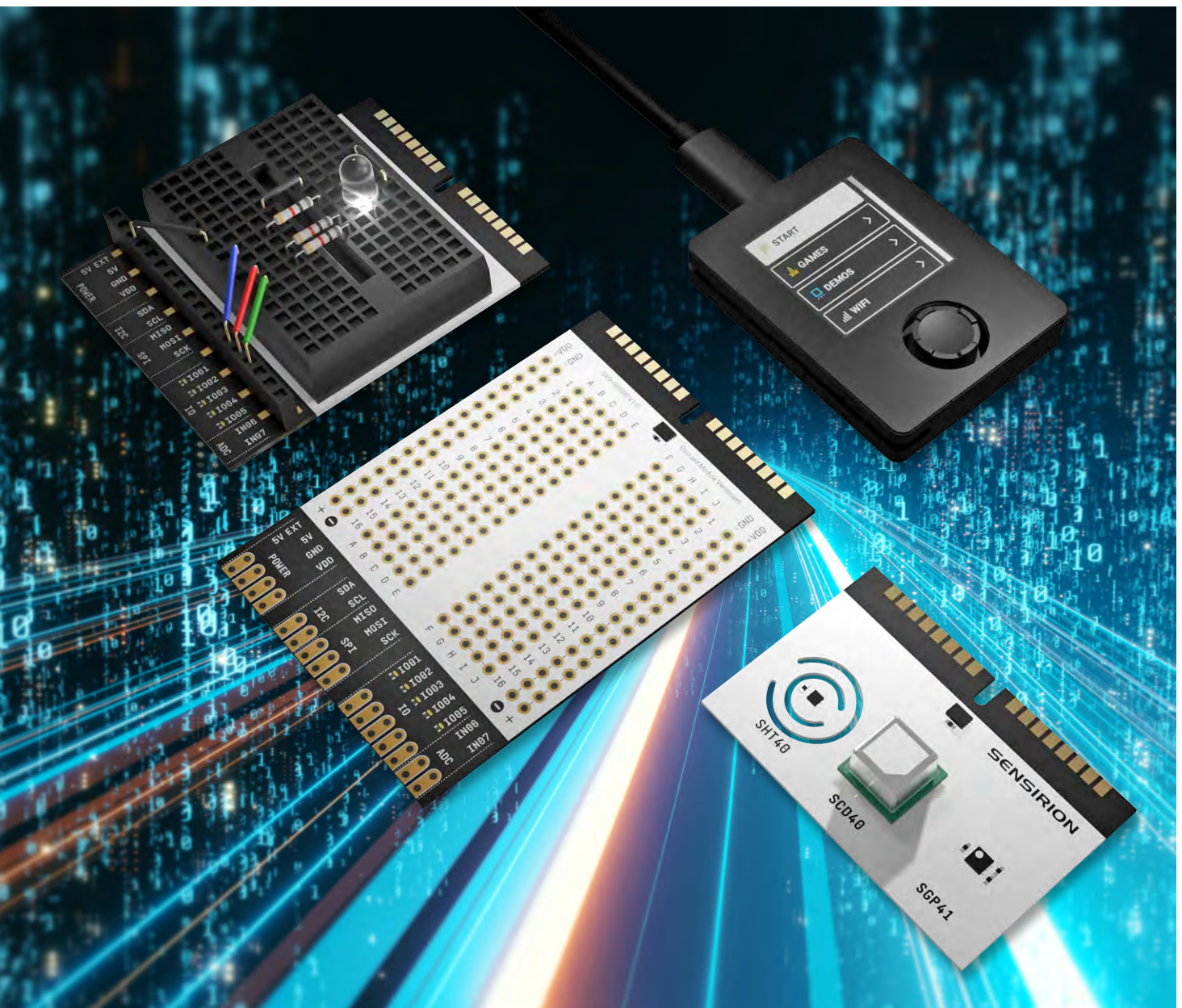


OXON – die universelle Experimentierplattform für den schnellen Lernerfolg

Ein Erfahrungsbericht

von Dr. Peter Tschulik

Seit Kurzem stehen im ELVshop verschiedene Komponenten des Schweizer Start-ups OXON zur Verfügung. Das Versprechen dieses europäischen Herstellers, mit einem einsteigerfreundlichen Komplettsset und einem Elektronikurs auf einfache Weise die Programmierung mittels Python zu erlernen, ließ mich als eingefleischten Arduino C++-Programmierer aufhorchen. Im Folgenden beschreibe ich die Möglichkeiten der Plattform sowie die Erfahrungen meines Einstiegs mit diesem System.



In vielen europäischen Ländern herrscht ein eklatanter Fachkräftemangel in den MINT-Berufen. Ein Grund dafür ist, dass die Lehrpläne, nach denen unsere Kinder unterrichtet werden, irgendwo in der Zeit der ersten industriellen Revolution entstanden sind. Tatsächlich werden unseren Kindern Sprachen in der Form von Lesen und Schreiben vermittelt. Doch was ist mit dem Programmieren, der vielleicht wichtigsten Sprache im 21. Jahrhundert? Vermehrt stelle ich fest, dass große Teile der Jugend zwar immer früher Mobiltelefone und Computer bedienen können, aber kein Verständnis für die grundlegenden Prozesse dahinter aufbringen. Der US-amerikanische Astrophysiker Carl Sagan hat das einmal wie folgt auf den Punkt gebracht: „Wir leben in einer Gesellschaft, die hochgradig von Technologie abhängig ist, in der aber kaum jemand etwas von Technologie versteht!“

Außerdem hört man oft Klagen, dass interessante Elektronikprodukte für den Maker nur aus Fernost erhältlich sind und die Einstiegshürde oftmals recht hoch ist. Ein Beispiel dafür ist die [Firma M5-Stack](#), die eine sehr breite und interessante Palette ESP32-basierter Controller und Erweiterungen mit vielen Funktionen anbietet. Für einen Einstieg ist es einerseits nicht immer leicht, die richtigen Komponenten auszuwählen und andererseits die verschiedenen Librarys, Boards und Erweiterungen in die Arduino IDE einzubinden.

Das Schweizer Start-up OXON, das 2013 von dem umtriebigen Unternehmer, Erfinder und Designer Thomas Garaio gegründet wurde, ist angetreten, die oben beschriebenen Herausforderungen zu lösen. Thomas Garaio ist ein faszinierender Mensch: In seiner Kindheit hatte er im Alter von zwölf Jahren über seinen Vater Kontakt zu den Computern der damaligen Pionierzeit wie Commodore VC-20/C64/Amiga, Apple II und Macintosh. Das Programmieren hat er sich selber beigebracht, zuerst durch Abtippen und Ausprobieren, später durch das Schreiben eigener Programme. In diese Zeit fielen auch viele Experimente mit Elektronikbausätzen u. a. von ELV.

Nach einer kaufmännischen und Managementausbildung arbeitete Thomas Garaio zunächst als Freelancer und gründete dann die Garaio AG, die Software-Engineering für mittlere und große Unternehmen anbietet. Als er im Jahr 2013 die Firma dem Management verkaufte, war diese auf 150 Mitarbeitende gewachsen. Im gleichen Jahr fand die Ausgründung der damaligen Innovationsabteilung in die OXON AG statt. Diese ist eine Innovationsagentur, die Elektronik, Software und Design in einem Haus kombiniert. Die komplette Entwicklung erfolgt zu hundert Prozent in der Schweiz am Standort in Liebfeld, einem Vorort von Bern. Hier verfügt die Firma über einen umfangreichen Gerätepark mit 3D-Druckern, CNC-Fräsen, Bestückungsmaschinen, Lasercutter, Tiefziehmaschine, Spritzgusspresse und mehr.

Da alles aus einer Hand von einem einzigen Team hergestellt wird, können komplexe Produkte in sehr kurzer Zeit gefertigt werden. Die komplette Produktion ist an Auftragsfertiger ausgelagert. Die

OXON AG konzentriert sich auf die Bereiche IoT, Gesundheit und Bildung. Im Bereich IoT wurden vor allem [Smart-Buttons und Sensoren](#) für das LoRaWAN®-Protokoll entwickelt. Im Gesundheitswesen wurde über die Ausgründung Oxomed AG ein [Pfleger- und Notrufsystem](#) entwickelt, das in der Schweiz in Pflegeeinrichtungen verwendet wird.

Auf dem Swiss Economic Forum, das ca. 2000 Vertreter aus Wirtschaft, Politik und Gesellschaft beherbergt, erkannte Thomas Garaio 2017 in einem Gespräch, dass die Wirtschaft zu wenig für die Informatikausbildung in Schulen macht, und so wurde das Kids-Coding-Projekt lanciert. Dafür wurde ein pädagogischer Lerncomputer entwickelt, ein Lehrmittelbehelf geschrieben und eine Schulklasse als Tester gesucht. Das Medieninteresse war so groß, dass sich die OXON AG daraufhin entschieden hat, in diesen Bereich weiterzumachen. Daraus entstand dann die [Oxocard-Reihe](#), die seit 2017 verfügbar ist. Während die älteren Oxocards „nur“ dazu geeignet waren, Programmieren zu lernen oder tolle Schreibtisch-Gadgets zu sein, bindet die neue Oxocard Connect die weite Welt der Elektronik unter dem Motto „Einstecken, ausprobieren, experimentieren“ ein. Man kann, wie man es von Spielkonsolen kennt, fertige oder selbst gelötete Platinen einstecken. Fertige Platinen werden als Cartridges bezeichnet und enthalten außergewöhnliche Sensoren. Mit der Skriptsprache [NanoPy](#) bietet OXON eine - basierend auf Python - einfach zu nutzende Programmiersprache fürs Embedded Computing an.

Im Folgenden beschreibe ich zunächst das im ELVshop erhältliche [Oxocard Connect Innovator Starter-Kit](#) und gehe danach auf die derzeit verfügbaren Cartridges [Time of Flight ToF für Abstandsmessungen](#) und [Luftgütesensor AIR](#) ein (Bild 1).

Unpacking

Das umfangreiche Komplettsset Oxocard Connect Innovator Starter-Kit enthält neben einer Oxocard Connect und einem Breadboard-Cartridge 96 Elektronikbauteile, mit denen man eine Vielzahl elektronischer Schaltungen aufbauen kann. Das Set kommt umweltfreundlich verpackt in einer nett bedruckten Kartonbox. In dieser befinden sich

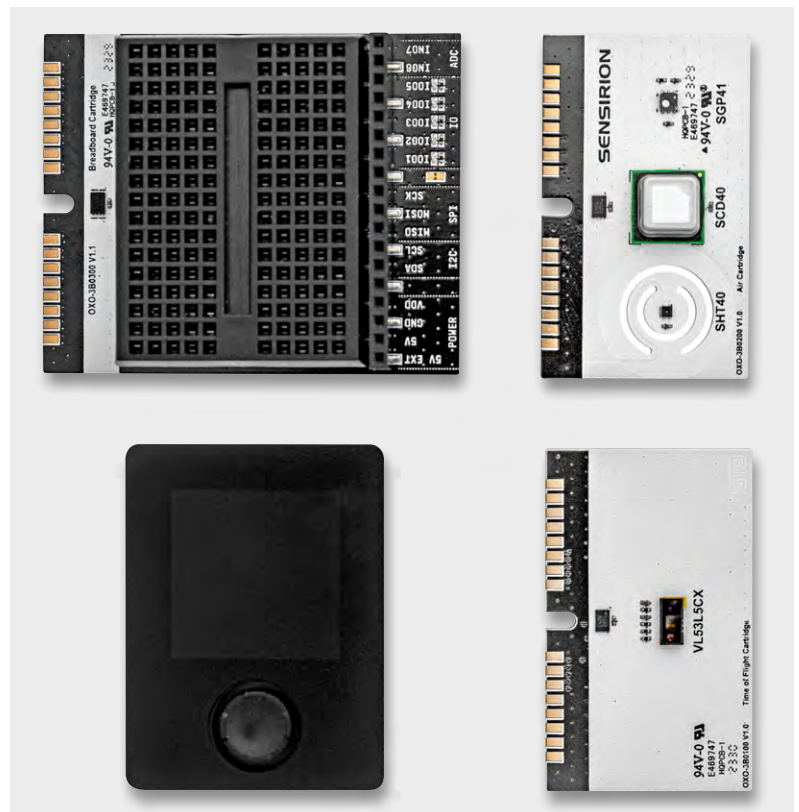


Bild 1: Oxocard Connect, Breadboard-Cartridge, Time of Flight Cartridge und Luftgütesensor AIR Cartridge



Bild 2: Oxocard Connect mit Hauptmenü



Bild 3: Breadboard-Cartridge mit Bauteilen

in einer weiteren Kartonbox die Oxocard Connect als Herzstück des Experimentierkits, basierend auf einem ESP32 mit 2 MB PSRAM und 8 MB Flash. Zusätzlich bietet die Oxocard Connect einen hochauflösenden TFT-Bildschirm (240 x 240 Pixel), einen Joystick zum Navigieren, einen USB-C-Anschluss sowie einen Expansions-Port zum Anschluss der Cartridges. Entnimmt man dem kleinen Karton die Oxocard Connect, so hält man ein sehr kompaktes und formschönes Gerät in den Händen (Bild 2).

Im Karton befindet sich eine Kurzanleitung. Ein ausführliches [deutschsprachiges Handbuch](#) kann auf der Produktseite unter „Downloads“ heruntergeladen werden.

Das große ESD-Bag beinhaltet dann 96 Bauteile, wobei diese Zahl mit etwas Vorsicht zu genießen ist, da 75 Bauteile davon extra verpackte, abgewinkelte Kabel von verschiedener Farbe und Länge sind. Ebenfalls extra verpackt sind ein PIR-Sensor (Bewegungsmelder) und ein Mikroservo SG92R. Die restlichen Bauteile sind ein Thermistor mit 10 k Ω (Temperatursensor), ein Photoresistor 10 k Ω (Lichtsensor), ein Potentiometer, ein Piezo für akustische Signale, eine RGB-LED, jeweils eine grüne, gelbe und rote LED, zwei Printtaster, ein 10-k Ω -Potentiometer und jeweils drei Widerstände 220 Ω , 330 Ω und 2,2 k Ω .

In einem weiteren ESD-Bag befindet sich das Breadboard-Cartridge, das neben einem Breadboard eine Stiftleiste mit den herausgeführten Signalen des ESP32 sowie einige SMD-LEDs enthält, die die Zustände der Ports anzeigen. Bild 3 zeigt das Breadboard-Cartridge mit einigen Bauteilen.

Schließlich befindet sich im Karton des Oxocard Connect Innovator Starter-Kit noch ein kurzes USB-C-Kabel, ein Lanyard und eine Dankkarte.

Verbindet man die Oxocard Connect mit einem USB-Netzteil, einer Powerbank oder dem PC erscheint nach einer kurzen Animation und einer Begrüßung ein Menü zur Sprachauswahl. Es stehen die Sprachen Englisch, Deutsch und Französisch zur Auswahl. Das Display ist trotz seiner geringen Grö-

ße sehr hell und löst sehr fein auf. Mit dem Joystick navigiert man in den Menüs und bestätigt durch einen Druck in der Mitte des Joysticks. Nach der Auswahl der Sprache kann man die Oxocard Connect gleich mit einem WLAN verbinden. OXON hat es sogar geschafft, eine Bildschirmtastatur auf dem kleinen Display darzustellen, damit man darüber das WLAN-Passwort eingeben kann.

Nach dem Verbinden mit dem WLAN erfolgt eine weitere Abfrage, ob man sich mit dem Editor verbinden möchte. Wenn man das bestätigt, bekommt man im Display einen dreistelligen Code angezeigt. Ruft man an einem PC oder Laptop dann die [Webseite](#) auf, kann man auf der linken Seite den dreistelligen Code eingeben und schon ist der Editor mit der Oxocard Connect über das WLAN verbunden. Es erscheint das in Bild 4 zu sehende Fenster.

Wichtig ist, dass unten links „Verbunden“ steht. Drückt man auf das Blitzsymbol im mittleren Fenster, wird das Python-Skript auf die Oxocard Connect übertragen und sofort gestartet. Das in Bild 4 dargestellte Skript zeichnet viele kleine bunte Kästchen am Display, die laufend die Farbe ändern. Hält man am Joystick die untere Taste gedrückt, kann die Ausführung des Skripts wieder gestoppt werden und man gelangt in das Hauptmenü.

Nach diesem Start fand bei mir gleich ein Firmware-Update statt. Leichter und intuitiver kann ein erster Einstieg in die Mikroprozessorwelt wohl kaum sein!

Die Hard- und Software sind übrigens Open Source und auf [„github“](#) verfügbar.

Das Hauptmenü der Oxocard Connect ist intuitiv und übersichtlich, wie in Bild 2 ersichtlich.

- Der oberste Menüeintrag trägt eine Flagge als Symbol und ist mit „Start“ bezeichnet. Dieser Menüpunkt startet das Skript, das über den NanoPy Editor auf die Oxocard Connect übertragen wurde. Dieses bleibt bis zum Überschreiben durch ein neues Skript auch bei einer Unterbrechung der Stromzufuhr erhalten.
- Der zweite Menüpunkt „Games“ beinhaltet zwei Spiele: „Electron“ und „Logic Gates“.
- Der dritte Menüpunkt „Demos“ beinhaltet 16 Demoskripts.
- Der vierte Menüpunkt „Wifi“ verbindet die Oxocard Connect wie oben beschrieben mit dem WLAN. Neben dem oben beschriebenen Vorgang gibt es auch die Möglichkeit, sich über das Smartphone mit der Oxocard Connect zu verbinden. In diesem Falle spannt die Oxocard Connect einen Hotspot auf, mit dem sich das Smartphone oder ein ähnliches Gerät mit ihr verbinden kann.

- Der fünfte Menüpunkt „Pairing“ verbindet die Oxocard Connect wie oben beschrieben mit dem NanoPy Editor.
- Der sechste Menüpunkt „Info“ zeigt im ersten Screen die HW- und FW-Version an, im zweiten Screen einen QR-Code, der auf die Homepage von [Oxocard](https://www.oxocard.com) verweist, und im letzten Screen die Open-Source-Lizenzvereinbarung.
- Der siebte Menüpunkt „HW TEST“ checkt die Funktion der Tasten, des Displays und des WLANs.
- Der achte Menüpunkt „Einstellung“ bietet zwei Untermenüs zur Einstellung der Sprache und der Rotation des Bildschirms (Standardwert 90°).
- Der neunte Menüpunkt „Zurücksetzen“ bietet zwei Untermenüs: „Reset Scripts“ dient zum Löschen des vom NanoPy übertragenen Skripts. In diesem Fall muss man sich wie oben beschrieben auch neu mit dem NanoPy Editor verbinden. Der zweite Untermenüpunkt „Factory Reset“ setzt alle Einstellungen inklusive der WLAN Verbindung zurück.
- Der letzte Menüpunkt „Ausschalten“ schaltet die Oxocard Connect aus. Ein Wiedereinschalten erfolgt mit einem Druck auf die mittlere Joystick-Taste.

Erste Schritte in der Softwareumgebung

OXON verspricht mit dem Kauf einen kostenlosen unbegrenzten Zugriff zum Editor von [NanoPy.io](https://nanopy.io) mit einer Vielzahl von Skripts, die man per Knopfdruck auf die Oxocard Connect übertragen kann. Außerdem findet man einen Elektronikurs mit 15 Experimenten, der Schritt für Schritt zeigen soll, wie man LEDs ansteuert, einen Servo anschließt, mit einem Piezo akustische Signale erzeugt und vieles mehr.

Das vorinstallierte Betriebssystem bietet diverse Spiele und Demoskripts wie oben bereits beschrieben. Die Web-basierte NanoPy-Scripting-Umgebung bietet über 100 Beispiele inklusive diverser Spiele. Die Übertragung des Skripts kann dabei wie weiter oben beschrieben über WLAN oder USB erfolgen. Umfangreiche Bibliotheken stehen zur Verfügung, ebenso MQTT, HTTP und HTTPS-Kommunikation und eine 2D- und 3D-Grafik-Engine.

Beginnen wir mit der NanoPy-Oberfläche, wie sie in **Bild 4** dargestellt ist. Der Bildschirm ist in drei Teile geteilt.

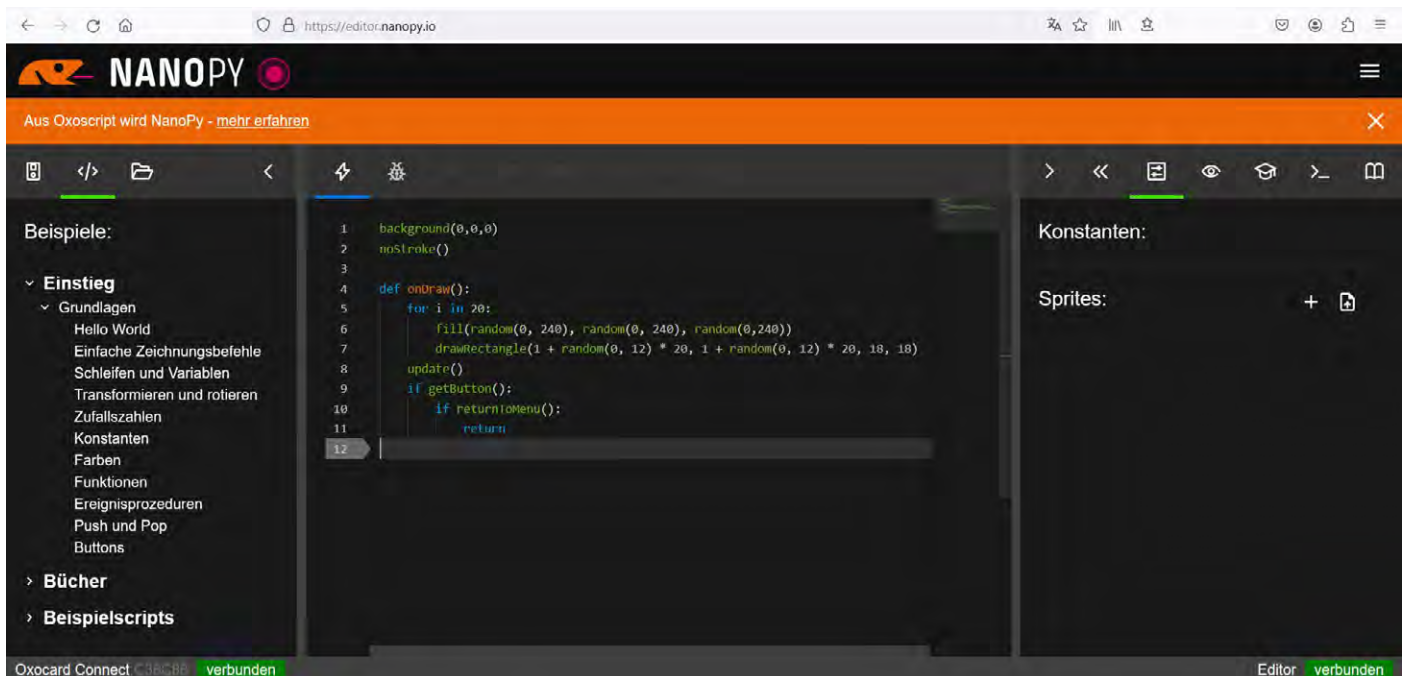


Bild 4: NanoPy-Oberfläche

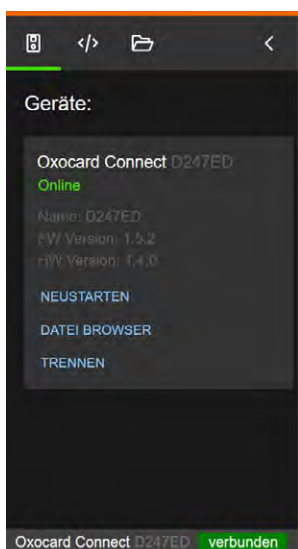


Bild 5: Dateibereich der NanoPy-Oberfläche

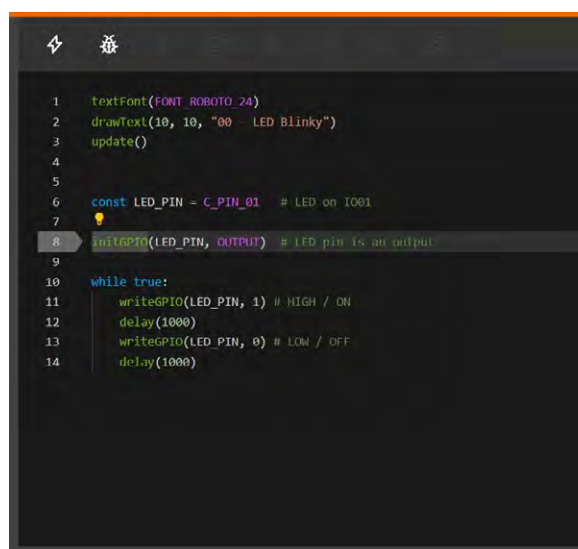


Bild 6: Programmereich der NanoPy-Oberfläche

Der linke Teil enthält im oberen Navigationsbereich drei Schaltflächen, wie **Bild 5** zeigt. Die linke Schaltfläche „Meine Geräte“ zeigt an, ob die Oxocard Connect mit dem Editor verbunden ist und listet die HW- und FW-Version auf. Die mittlere Schaltfläche „Beispiele“ zeigt alle Beispielskripts an. Auf diese werde ich nach der Beschreibung des Editors genauer eingehen. Mit der rechten Schaltfläche „Meine Skripts“ können eigene Skripts gespeichert und aufgerufen werden. Der Pfeil „<“ verbirgt das Menü. Im mittleren Bereich wird der Code angezeigt (**Bild 6**).

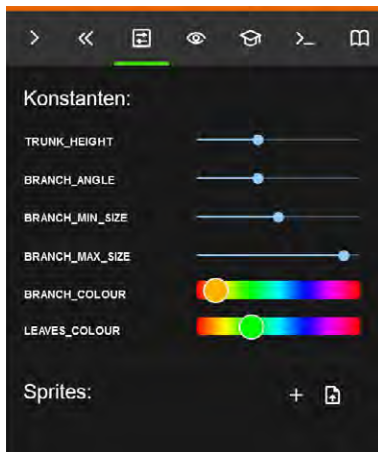


Bild 7: Informationsbereich der NanoPy-Oberfläche

Die Farbdarstellung der verschiedenen Skriptelemente wie Befehle, Schleifen, Kommentare usw. erleichtert die Lesbarkeit der Beispielskripte. Klickt man auf einen Befehl, so erscheint wie in Bild 6 gezeigt eine gelbe Lampe. Klickt man auf diese Lampe, so kann man sich eine ausführliche Dokumentation des Befehls ansehen, was für Anfänger ungemein hilfreich ist.

Mit der linken Schaltfläche „Code ausführen“ wird das Skript auf die Oxocard Connect übertragen und dort sofort ausgeführt. Mit der Schaltfläche „Code debuggen“ erscheinen die weiter rechts gelegenen ausgegrauten Schaltflächen und das Skript springt auf die erste Zeile.

Die Schaltfläche „Debug ausführen“ startet das Skript. Die Schaltfläche „Beobachten“ bewirkt, dass das Skript Schritt für Schritt ausgeführt wird. Die Schrittgeschwindigkeit ist dabei rechts oben einstellbar. Die nächste Schaltfläche „Pause“ pausiert die Ausführung. Beim Druck auf die Schaltfläche „Schritt“ wird der jeweils nächste Programmschritt ausgeführt. So kann man das Ausführungstempo selbst bestimmen und genau sehen, was welcher Programmschritt bewirkt.

Die Schaltfläche „Bildschirmfoto“ erzeugt einen Screenshot des LCD-Displays des Oxocard Connect. Im rechten Bereich werden weitere Informationen dargestellt, wie Bild 7 zeigt. Mit der Schaltfläche „>“ kann das Fenster ausgeblendet werden, die Schaltfläche „<<“ hingegen vergrößert das Fenster. Die Schaltfläche „Konstanten“ zeigt die Werte der Konstanten an. Außerdem können hier „Sprites“, d. h. kleine Bilder, erstellt und in den Code eingefügt werden. Unter „Variablen“ werden „Globale Variablen“, „Lokale Variablen“ und „Objekte“ angezeigt. Die Schaltfläche „Tutorial“ zeigt, wo verfügbar, Lerninhalte für das jeweilige Skript an. Ich gehe ein wenig später noch genauer darauf ein. Die Schaltfläche „Terminal“ blendet ein Terminalfenster ein. Die Schaltfläche „Dokumentation“ bietet umfangreiche Informationen zu programmtechnischen Fragen und eine Referenz zu allen Befehlen.

Weitere umfangreiche Informationen über die NanoPy zugrunde liegende Sprache Python, eine Übersicht und Einführung in die Programmierung sowie die komplette Sprachreferenz findet man bei [NanoPy.io](https://nanopy.io).

Zusammenfassend kann gesagt werden, dass dieser Online-Editor vom Funktionsumfang, der einfachen Bedienung und der grafischen Aufmachung anderen Programmieroberflächen meilenweit voraus ist und gerade für Anfänger schnell intuitiv erlernbar ist.

Nach dieser Vorstellung des NanoPy Editors will ich mich nun mit den Skriptbeispielen beschäftigen. Zuerst klickt man im linken Fenster auf die Schaltfläche „Beispiele“ und wählt dort unter Grundlagen das erste Beispiel „Hello World“ aus (Bild 8).

Überträgt man das Skript auf die Oxocard Connect, wird „Hello World“ am Display ausgegeben. Das zugehörige Tutorial hat sechs Schritte, wobei zuerst die Funktion des Skripts, danach der Aufbau, Abänderungsmöglichkeiten und schließlich weitere Inspirationen angezeigt werden. Bei Änderungen im mittleren Programmfenster wird das Skript automatisch unter einem neuen Namen unter „Meine Scripts“ gespeichert. Wenn man nacheinander die elf Beispiele unter dem Punkt Einführung durchgeht, erlernt man sehr schnell die wichtigsten Befehle und den Aufbau der Programmiersprache Python. Die Beispiele decken einfache und komplexe Zeichnungsbefehle, Schleifen, Wenn-dann-Bedingungen, Variablen und Konstanten, Zufallszahlen, Farben, Funktionen und die Abfrage der Taster ab. Kein Beispielskript benötigt ein externes Bauteil oder das Breadboard-Cartridge. Ich habe noch nie einen einfacheren und intuitiveren Einstieg in die Mikrocontrollerprogrammierung gesehen – ein dickes Lob an die Entwickler bei OXON!

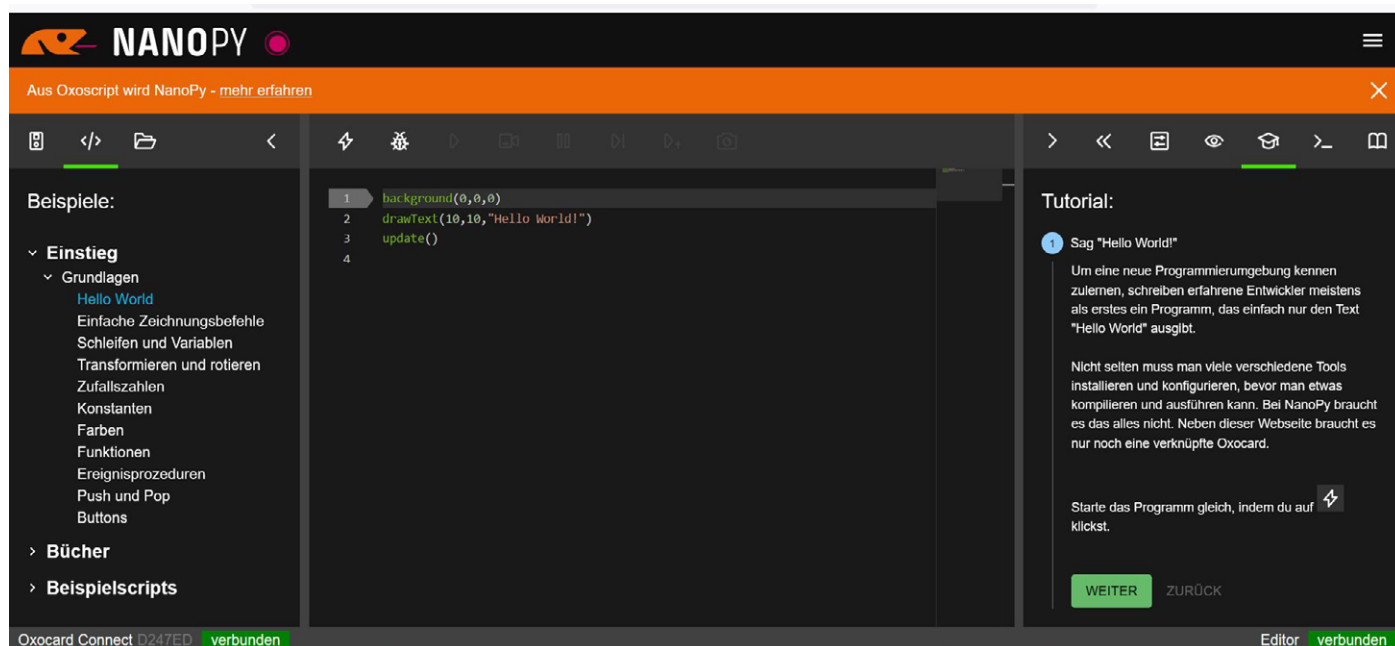


Bild 8: Das einfachste Skriptbeispiel mit eingblendetem Tutorial

Die nächste Gruppe von Beispielen bezieht sich auf das Buch „Informatikmagie“. Dieses Begleitbuch, das von Thomas Garaio verfasst wurde, ist leider nicht im ELVshop verfügbar, aber beispielsweise bei Amazon, Weltbild oder Heise für rund 40 € zu kaufen. Ohne dieses Buch sind die Beispiele wohl nur für bereits versiertere Programmierer nützlich, da keine Informationen im Tutorial verfügbar sind. Die Nummern vor den Beispielen beziehen sich anscheinend auf die Kapitelnummern im Buch.

Bevor ich die anderen Beispielrubriken beschreibe, möchte ich kurz auf den Elektronikurs eingehen, der sich unter Connect → Elektronikurs befindet und der das Breadboard-Cartridge mit den beigefügten Elektronikbauteilen nutzt.

Experimente mit dem Breadboard-Cartridge

Die Breadboard-Cartridge steckt man seitlich in die Oxocard Connect, wobei eine Verpolung des 16-poligen Steckers unmöglich ist, da er beidseitig eingesteckt werden kann. Außerdem können Cartridges jederzeit unter Spannung ein- und ausgesteckt werden. Auf dem Breadboard-Cartridge selbst sind folgende Signale herausgeführt (Bild 9):

- Abschnitt Power:
 - 5V EXT ist wohl ein Eingang, mit dem die Oxocard Connect mit 5 Volt versorgt werden kann, wenn dies nicht über USB erfolgt.
 - 5V ist die 5-Volt-Spannungsversorgung über USB mit dem dazugehörigen GND.
 - VDD ist der Ausgang des 3-V-Spannungsreglers.
- Abschnitt I²C:
 - Hier ist der I²C-Bus verfügbar (SDA und SCL).
- Abschnitt SPI:
 - Hier sind die SPI-Signale verfügbar (MISO, MOSI und SCK).
- Abschnitt IO:
 - Hier sind fünf IO-Signale des ESP32 herausgeführt (IO01 = ESP GPIO 20, IO02 = ESP GPIO 25, IO03 = ESP GPIO 26, IO04 = ESP GPIO 33, IO05 = ESP GPIO35). Alle fünf Ein- und Ausgänge verfügen über grüne LEDs, die bei einem High am Ausgang aufleuchten.
- Abschnitt ADC:
 - Hier sind zwei Analog-digital-Umsetzer, die als Eingänge herausgeführt sind. (IN06 = ESP GPI34, IN07 = ESP GPI35).

Außerdem befindet sich, wie auf allen Cartridges auch, ein serielles EEPROM mit 512 kbit (64 k x 8) auf der Platine. Beim Einstecken einer Cartridge wird diese automatisch erkannt.

Das erste Beispiel „00 – LED Blinky“ lässt eine LED an IO01 im Ein-Sekunden-Takt blinken. Die Verdrahtung der LED sowie die Funktion des Skripts wird wieder anschaulich und leicht verständlich in einem Tutorial erklärt. Interessant ist, dass das Skript weiterhin in der Oxocard Connect gespeichert wird und nicht auf dem EEPROM der Cartridge. In diesem sind wohl Testskripts gespeichert, die vor allem die LEDs in unterschiedlicher Form blinken lassen und deren Ausführung beim Einstecken der Cartridge angeboten werden.

Das zweite Beispiel „01-LED und Button“ erweitert das erste Beispiel um einen Taster, der gedrückt werden muss, damit die LED aufleuchtet. Als kleiner Kritikpunkt ist anzumerken, dass die beigefügten Taster wegen der sehr kurzen Beinchen nicht optimal im Breadboard stecken und es leicht zu Kontaktproblemen kommen kann.

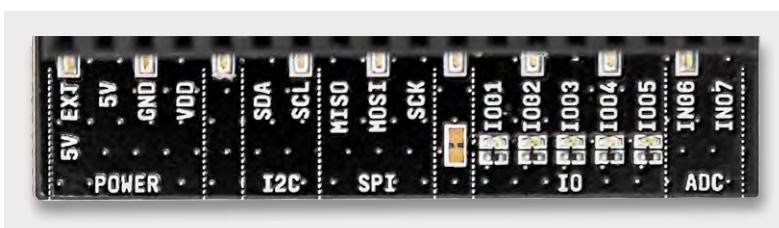


Bild 9: Herausgeführte Signale am Breadboard

Weitere Beispiele sind eine Ampel mit und ohne Fußgängertaster, eine Helligkeitssteuerung für eine LED mittels Pulsweitenmodulation, ein LED-Dimmer mit Tastensteuerung, die Ansteuerung der RGB-LED, die Ansteuerung des Servos sowie eine Scheibenwischersteuerung mit dem Servo, das Einlesen eines Potentiometers über einen Analogeingang, die Steuerung des Servos über ein Potentiometer, die Temperaturmessung mithilfe des NTC, die Messung des Umgebungslichts mittels LDR, eine Lichtsteuerung mittels LDR, die Tonerzeugung mittels Piezo-Signalgeber und eine Alarmanlage mit Bewegungsmelder und Piezo-Signalgeber.

Auch für diese 16 Beispiele gilt, dass sie für Anfänger leicht durchschaubar und aufbaubar sind und einen schnellen Lernerfolg versprechen. Das letzte Skript erlaubt es, eigene Skripts auf das EEPROM der Cartridge zu spielen und es von dort zu starten.

Weitere vielfältige Beispielskripts finden sich in der Rubrik „Beispielskripts“, die weitere 2D- und besonders beeindruckende 3D-Grafikbeispiele, Skripts der Oxocard Connect Community, Beispiele für verfügbare Funktionen, weitere Spiele, Rezepte für die Einbindung von ChatGPT, Neopixel-LEDs und Shelly-Hausautomatisierungsprodukte und vieles mehr enthält.

Die Rubrik „Galaxy“ enthält weitere Spiele und Grafikdemos, die Rubrik „Artwork“ weitere atemberaubende Grafiken und die Rubrik „Science“ Wissenschaftsanwendungen. Im Gegensatz zu den gerade genannten Rubriken, die keine Tutorials bieten, enthält die Rubrik „Connect“ neben dem weiter oben beschriebenen Elektronikurs noch weitere gut dokumentierte Demos und Spiele. Die letzte Rubrik „Cartridges“ enthält weitere Beispielskripts zu den drei derzeit verfügbaren Cartridges Breadboard-Cartridge, Time of Flight Cartridge und Luftgütesensor AIR Cartridge. Für viele abwechslungsreiche Stunden ist gesorgt!

Zuletzt möchte ich noch kurz auf die beiden Cartridges Time of Flight und Luftgütesensor AIR eingehen.

OXON-Steckkarte Time of Flight Cartridge für Abstandsmessungen

Bild 10 zeigt die Cartridge Time of Flight. „Time of Flight“ (oder kurz ToF) ist ein Verfahren zur Distanzmessung mittels Laser.



Bild 10: Cartridge Time of Flight



Bild 11: Tiefendiagramm mit Darstellung der sich annähernden Hand

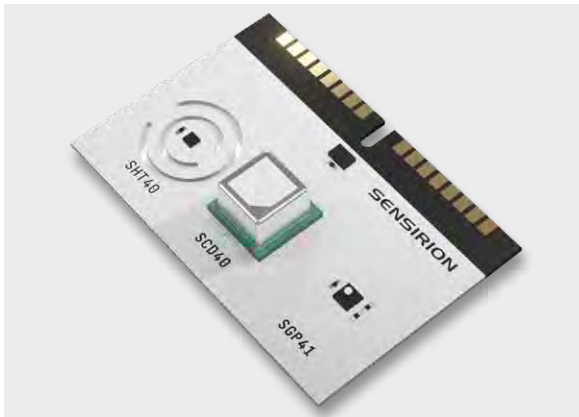
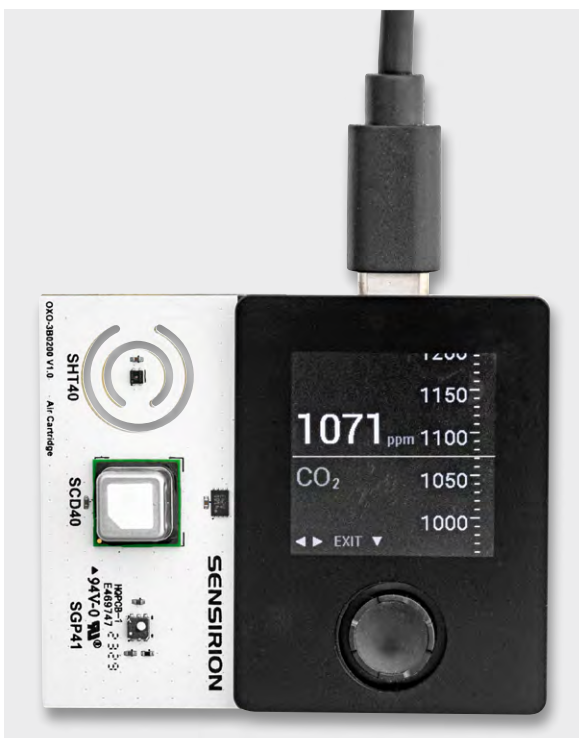


Bild 12: Luftgütesensor AIR Cartridge

Bild 13: CO₂-Gehalt in der Luft mit cooler Skalenanzeige

Der verwendete Sensor VL53L5CX von ST Microelectronics ist ein 8x8-Multizonen-Sensor, mit dem eine 3D-Abtastung mit insgesamt 64 Pixeln möglich ist. Das Sichtfeld beträgt ca. 65 Grad, und Tiefenbilder können mit einer Distanz von bis zu 4 m erzeugt werden. Damit lassen sich Distanzen messen, Handgesten erkennen oder sogar Objekte detektieren! Der Sensor funktioniert unter verschiedenen Lichtbedingungen. Das Grundprinzip: Licht wird ausgesendet, reflektiert und dann erfasst, um Informationen über die Umgebung zu erhalten.

Beim Einschub der Cartridge öffnet sich ein Skript mit einer beeindruckenden 3D-Darstellung eines Tiefendiagramms (Bild 11).

Alternativ kann man in der NanoPy-Entwicklungsumgebung unter Cartridges → ToF → Abstandssensor ein Skript aufspielen, das den Abstand zu einem Objekt misst. Man könnte sich damit zum Beispiel einen eigenen Abstandswarner fürs Auto bauen.

OXON-Steckkarte Luftgütesensor AIR Cartridge

Bild 12 zeigt die Cartridge Luftgütesensor AIR. Diese Cartridge bietet einen MOx-basierten Gassensor (SGP41), der flüchtige Kohlenstoffverbindungen (VOC) und Stickoxide (NOx) messen kann. Ein weiterer Sensor auf der Platine ist der SCD40, ein photoakustischer Sensor, der zuverlässig den CO₂-Wert im Wertebereich von 400 ppm bis 2000 ppm eruiieren kann. Schließlich befindet sich noch mit dem SHT40 ein kalibrierter Temperatur- und Feuchtesensor mit an Bord.

Beim Einschub der Cartridge öffnet sich ein Skript, das die einzelnen Werte in einer coolen Skalenanzeige darstellt (Bild 13).

Alternativ kann man in der NanoPy-Entwicklungsumgebung unter Cartridges → Air → Sensorwerte ein Skript aufspielen, mit dem man von den drei Sensoren alle sieben Werte abfragen und anzeigen kann. Unter Cartridges → Air → IAQ-Smiley befindet sich ein weiteres Skript, das auf dem Display eine nett animierte CO₂-Ampel anzeigt.

Fazit

Mit dem Oxocard Connect Innovator Starter-Kit ist es mir gelungen, ganz leicht in die Python-Programmierung einzusteigen. Das gut durchdachte System mit den Cartridges macht es Anwendern einfach, auch komplexe Sensoren in kürzester Zeit einzubinden. Die ausgezeichneten Tutorials und der integrierte Elektronikurs eignen sich hervorragend für Schulen und in der Bildung, um junge Menschen für Mikrocontroller und deren Programmierung zu begeistern.

An der Programmierung mit NanoPy hat mich vor allem der kompakte Pythoncode und der umfangreiche Befehlssatz fasziniert, der auch komplexe 3D-Darstellungen beherrscht, verschlüsselte HTTPS-Verbindungen aufbaut sowie MQTT-Kommunikation erlaubt. Damit können auch komplexe IoT-Anwendungen rasch realisiert werden.

Nur ganz wenige Verbesserungsmöglichkeiten sind mir aufgefallen: Manche Bauteile wie z. B. die Taster sind nicht für das Breadboard-Cartridge optimiert. Außerdem ist es mir nicht gelungen, Skripts ohne den Umweg über das Hauptmenü zu starten, wie es für die Cartridges sehr wohl funktioniert. Und schlussendlich haben mir komplexere Beispiele für Anwendungen mit HTTP, HTTPS und MQTT gefehlt. Schön wäre auch, wenn das Buch „Informatikmagie“ dem Oxocard Connect Innovator Starter-Kit beiliegen würde oder es per Download als PDF verfügbar wäre. Vielleicht wird es auch einmal möglich, dass andere ESP32-Standardboards in den Genuss der NanoPy-Programmierung kommen.

Für Einsteiger und Umsteiger ist das Angebot von OXON uneingeschränkt zu empfehlen!